

CS+ V8.06.00

統合開発環境

User's Manual

ユーザーズマニュアル RX デバッグ・ツール編 対象デバイス RX ファミリ

本資料に記載の全ての情報は発行時点のものであり、ルネサス エレクトロニクスは、 予告なしに、本資料に記載した製品または仕様を変更することがあります。 ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

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。
- 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。
- 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

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。

このマニュアルの使い方

このマニュアルは、RX ファミリ用アプリケーション・システムを開発する際の統合開発環境である CS+ について説明します。

CS+は、RX ファミリの統合開発環境(ソフトウエア開発における、設計、実装、デバッグなどの各開発フェーズに 必要なツールをプラットフォームである IDE に統合)です。統合することで、さまざまなツールを使い分ける必要がな く、本製品のみを使用して開発のすべてを行うことができます。

- 対象者 このマニュアルは、CS+を使用してアプリケーション・システムを開発するユーザを対 象としています。
- 目的 このマニュアルは、CS+の持つソフトウエア機能をユーザに理解していただき、これら のデバイスを使用するシステムのハードウエア、ソフトウエア開発の参照用資料として 役立つことを目的としています。

構成このマニュアルは、大きく分けて次の内容で構成しています。

1. 概 説 2. 機 能 A. ウインドウ・リファレンス B. 入出力機能

読み方 このマニュアルを読むにあたっては、電気、論理回路、マイクロコンピュータに関する 一般知識が必要となります。 凡例 データ表記の重み : <u>左が</u>上位桁、右が下位桁

七1列	アータ表記の重み	<u> </u>	上位桁,石かト位桁
	アクティブ・ロウの表記	XXX	(端子,信号名称に上線)
	注	本文	中についた注の説明
	注意	気を	つけて読んでいただきたい内容
	備考	本文	中の補足説明
	数の表記	10 進	数… XXXX
		16 逍	数… 0xXXXX

ロク
니ク

1.	概	説	10
1.1	概	要要	
1.2	特	長	10
2.	機	能	11
2.1	概	要要	
2.2	ディ	、ッグを始める前の準備	13
2.2	2.1	ホスト・マシンとの接続を確認する	13
:	2.2.1.1	【E1】の場合	13
2	2.2.1.2	【E20】の場合	
2	2.2.1.3	【EZ Emulator】の場合	
2	2.2.1.4	【シミュレータ】の場合	
2.3	ディ	、ッグ・ツールの動作環境設定	
2.3	3.1	使用するデバッグ・ツールを選択する	
2.3	3.2	【E1】の場合	
	2.3.2.1	[接続用設定] タブ	
	2.3.2.2	[デバッグ・ツール設定] タブ	
	2.3.2.3	[ダウンロード・ファイル設定]タブ	
	2.3.2.4	[フック処理設定] タブ	
2.3	3.3	【E20】の場合	
	2.3.3.1	[接続用設定] タブ	
2	2.3.3.2	[デバッグ・ツール設定] タブ	
2	2.3.3.3	[ダウンロード・ファイル設定] タブ	46
2	2.3.3.4	[フック処理設定] タブ	46
2.3	3.4	【EZ Emulator】の場合	47
2	2.3.4.1	[接続用設定] タブ	47
:	2.3.4.2	[デバッグ・ツール設定] タブ	50
2	2.3.4.3	[ダウンロード・ファイル設定] タブ	55
2	2.3.4.4	[フック処理設定] タブ	56
2.3	3.5	【シミュレータ】の場合	57
	2.3.5.1	[接続用設定] タブ	57
	2.3.5.2	[デバッグ・ツール設定]タブ	58
	2.3.5.3	[ダウンロード・ファイル設定] タブ	62
	2.3.5.4	[フック処理設定] タブ	62
2.4	ディ	、ッグ・ツールとの接続/切断	63
2.4	4.1	CS+ にデバッグ・ツールを接続する	63
2.4	4.2	CS+からデバッグ・ツールを切断する	65
2.4	4.3	CS+にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】【E20(JTAG)】	65

2.5	ダウ	ウンロード/アップロード	68
2.5.1		ダウンロードを実行する	68
2.5.2	2	応用的なダウンロード方法	70
2.5	5.2.1	ロード・モジュール・ファイルのダウンロード条件を変更する	71
2.5	5.2.2	ダウンロード・ファイル(*.hex/*.mot/*.bin)を追加する	73
2.5	5.2.3	複数のロード・モジュール・ファイルをダウンロードする	74
2.5	5.2.4	ヘキサ・フォーマット/モトローラ S フォーマット/バイナリ・データ・フォーマットのファイ. でソース・レベル・デバッグを行う76	ル
2.5	5.2.5	外部フラッシュ・メモリヘダウンロードする【E1】【E20】	76
2.5.3	5	アップロードを実行する	79
2.6	プロ	コグラムの表示と変更	80
2.6.1		ソース・ファイルを表示する	80
2.6.2	2	逆アセンブル結果を表示する	81
2.6	6.2.1	表示モードを変更する	81
2.6	5.2.2	表示形式を変更する	82
2.6	5.2.3	指定アドレスへ移動する....................................	82
2.6	5.2.4	シンボル定義箇所へ移動する	83
2.6	6.2.5	逆アセンブル結果の表示内容を保存する	83
2.6.3	5	他の処理と平行してビルドを実行する	84
2.6.4	Ļ	ライン・アセンブルを行う	85
2.6	6.4.1	命令を編集する	85
2.6	6.4.2	命令コードを編集する	86
2.7	PIC	:/PID 機能の使用	87
2.7.1		PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する	87
2.8	オー	-バーレイ・セクションの設定	90
2.8.1		優先セクションを指定する	90
2.9	プロ	コグラムの実行	93
2.9.1		マイクロコントローラ (CPU) をリセットする	93
2.9.2	2	プログラムを実行する	93
2.9	9.2.1	マイクロコントローラ(CPU)をリセットしてから実行する	94
2.9	9.2.2	現在のアドレスから実行する....................................	94
2.9	9.2.3	PC 値を変更してから実行する	95
2.9.3	5	プログラムをステップ実行する.......	95
2.9	9.3.1	関数内にステップ・インする(ステップ・イン実行)	96
2.9	9.3.2	関数をステップ・オーバーする(ステップ・オーバー実行)	96
2.9	9.3.3	関数内でリターンが完了するまで実行する(リターン・アウト実行)	96
2.9.4	ļ	指定ルーチンを実行する【E1】【E20】【EZ Emulator】	97
2.10	プロ	コグラムの停止(ブレーク)	99
2.10.	.1	プログラムの実行を手動で停止する	99
2.10.	.2	任意の場所で停止する(ブレークポイント)	99
2.1	10.2.1	使用するブレークポイントの種別/動作を設定する【E1】【E20】【EZ Emulator】	99
2.1	10.2.2	ブレークポイントを設定する	00
2.1	10.2.3	ハードウエア・ブレークポイントを編集する1	01

2.10.2.4	ブレークポイントを削除する	101
2.10.3	任意の場所で停止する(ブレーク・イベント)【E1】【E20】【EZ Emulator】	101
2.10.3.1	ブレーク・イベント(実行系)を設定する	101
2.10.3.2	ブレーク・イベント(実行系)を編集する	102
2.10.3.3	ブレーク・イベント(実行系)を削除する	102
2.10.4	変数 /l/O レジスタへのアクセスで停止する	103
2.10.4.1	変数 /l/O レジスタへのブレーク・イベント(アクセス系)を設定する	103
2.10.4.2	変数 /l/O レジスタへのブレーク・イベント(アクセス系)を編集する	107
2.10.4.3	変数 /l/O レジスタへのブレーク・イベント(アクセス系)を削除する	107
2.10.5	複数のブレーク・イベントを組み合わせる(組み合わせブレーク)【E1】【E20】【EZ Emulator】.	107
2.10.6	その他のブレーク要因	108
2.11 メモ	リ,レジスタ,変数の表示/変更	110
2.11.1	メモリを表示/変更する	110
2.11.1.1	表示位置を指定する	110
2.11.1.2	値の表示形式を変更する	111
2.11.1.3	メモリの内容を変更する....................................	112
2.11.1.4	プログラム実行中にメモリの内容を表示/変更する	113
2.11.1.5	メモリの内容を検索する....................................	116
2.11.1.6	メモリの内容を一括して変更(初期化)する	117
2.11.1.7	メモリの表示内容を保存する....................................	118
2.11.2	CPU レジスタを表示/変更する	120
2.11.2.1	値の表示形式を変更する	120
2.11.2.2	CPU レジスタの内容を変更する	121
2.11.2.3	CPU レジスタの表示内容を保存する	121
2.11.3	I/O レジスタを表示/変更する	121
2.11.3.1	I/O レジスタを検索する	122
2.11.3.2	I/O レジスタを整理する	122
2.11.3.3	値の表示形式を変更する	122
2.11.3.4	I/O レジスタの内容を変更する	123
2.11.3.5	プログラム実行中に I/O レジスタの内容を表示/変更する	123
2.11.3.6	I/O レジスタの表示内容を保存する	123
2.11.4	グローバル変数/スタティック変数を表示/変更する	123
2.11.5	ローカル変数を表示/変更する	123
2.11.5.1	値の表示形式を変更する	124
2.11.5.2	ローカル変数の内容を変更する	125
2.11.5.3	ローカル変数の表示内容を保存する	125
2.11.6	ウォッチ式を表示/変更する	126
2.11.6.1	ウォッチ式を登録する	126
2.11.6.2	登録したウォッチ式を整理する	127
2.11.6.3	登録したウォッチ式を編集する	128
2.11.6.4	ウォッチ式を削除する	128
2.11.6.5	値の表示形式を変更する	128

2.11.6.6	ウォッチ式の内容を変更する...................................	129
2.11.6.7	プログラム実行中にウォッチ式の内容を表示/変更する	129
2.11.6.8	ウォッチ式をエクスポート/インポートする	129
2.11.6.9	ウォッチ式の表示内容を保存する	131
2.12 スタ	ックからの関数呼び出し情報の表示	132
2.12.1	コール・スタック情報を表示する	132
2.12.1.1	値の表示形式を変更する	132
2.12.1.2	ソース行へジャンプする	133
2.12.1.3	ローカル変数を表示する	133
2.12.1.4	コール・スタック情報の表示内容を保存する	133
2.13 実行	履歴の収集	134
2.13.1	トレース動作の設定をする	134
2.13.1.1	【E1】の場合	134
2.13.1.2	【E20】の場合	136
2.13.1.3	【EZ Emulator】の場合	139
2.13.1.4	【シミュレータ】の場合	140
2.13.2	実行停止までの実行履歴を収集する................................	141
2.13.3	任意区間の実行履歴を収集する	141
2.13.3.1	トレース開始イベント/トレース終了イベントを設定する	141
2.13.3.2	複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】	144
2.13.3.3	プログラムを実行する.......	144
2.13.3.4	トレース開始イベント/トレース終了イベントを編集する	145
2.13.3.5	トレース開始イベント/トレース終了イベントを削除する	145
2.13.4	条件を満たしたときのみの実行履歴を収集する	145
2.13.5	実行履歴の収集を停止/再開する【E20】【シミュレータ】	146
2.13.5.1	実行履歴の収集を一時的に停止する場合	147
2.13.5.2	実行履歴の収集を再開する場合	147
2.13.6	実行履歴を表示する	147
2.13.6.1	表示モードを変更する	147
2.13.6.2	値の表示形式を変更する	148
2.13.6.3	他のパネルと連動させる	148
2.13.7	トレース・メモリをクリアする	149
2.13.8	トレース・データを検索する	149
2.13.8.1	命令レベルで検索する	149
2.13.8.2	ソース・レベルで検索する	151
2.13.9	実行履歴の表示内容を保存する	153
2.14 実行	「時間の計測	155
2.14.1	タイマ計測動作の設定をする【E1】【E20】【EZ Emulator】	155
2.14.2	実行開始から停止までの実行時間を計測する........................	155
2.14.2.1	ステータス・バーでの確認	156
2.14.2.2	イベント パネルでの確認	156
2.14.3	任意区間の実行時間を計測する	157

2.14	1.3.1 タイマ開始イベント/タイマ終了イベントを設定する 1	57
2.14	1.3.2 プログラムを実行する1	60
2.14	I.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】	62
2.14	1.3.4 タイマ開始イベント/タイマ終了イベントを編集する 1	63
2.14	1.3.5 タイマ開始イベント/タイマ終了イベントを削除する 1	63
2.14.4	測定可能時間の範囲1	63
2.15	カバレッジの測定【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72 RX72N グループ】】164	M,
2.15.1	カバレッジ測定の設定をする1	64
2.15.2	カバレッジ測定結果を表示する1	65
2.16	プログラム内へのアクションの設定1	68
2.16.1	printf を挿入する1	68
2.16	6.1.1 Printf イベントを設定する1	68
2.16	1.2 プログラムを実行する	69
2.16	1.3 出力結果を確認する	69
2.16	5.1.4 Printf イベントを編集する1	69
2.16.2	割り込みイベントを挿入する【シミュレータ】1	70
2.16).2.1 割り込みイベントを設定する1	70
2.16	2.2.2 プログラムを実行する	71
2.16	5.2.3 割り込みイベントを編集する1	71
2.17	イベントの管理	72
2.17.1	設定状態(有効/無効)を変更する 1	74
2.17.2	特定のイベント種別のみ表示する1	75
2.17.3	イベントのアドレスにジャンプする 1	75
2.17.4	イベントの詳細設定を編集する1	75
2.17	7.4.1 実行系イベントを編集する1	75
2.17	7.4.2 アクセス系イベントを編集する 1	76
2.17	7.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】	79
2.17.5	イベントを削除する1	81
2.17.6	イベントにコメントを入力する1	82
2.17.7	イベント設定に関する留意事項1	82
2.17	7.7.1 有効イベント数の制限	82
2.17	7.7.2 実行中に設定/削除可能なイベント種別1	84
2.17	7.7.3 その他の注意事項	84
2.18	フック処理を設定する1	86
2.19	消費電流測定【E2】1	89
2.20	デバッグ・コンソールの使用	95
2.21	入力値について	99
2.21.1	入力規約1	99
2.21.2	シンボル名の入力補完機能	203
2.21.3	入力不備箇所に対するアイコン表示2	<u>2</u> 04
2.22	E1/E20/EZ エミュレータとマイクロコントローラの相違点2	205
2.23	その他の使用上の注意事項【E1】【E20】【EZ Emulator】2	206

Α.	ウイン	ィドウ	・リフ	ファレ	ンス	 	 	 	 	 	 		207
A.1	説	明				 	 	 	 	 	 		. 207
В.	入出り	り機能				 	 	 	 	 	 	• • • •	401
B.1	標準之	入出力/	ファイ	ル入出さ	b	 	 	 	 	 	 		. 401
改訂詞	3録					 	 	 	 	 	 	(C - 1

1. 概 説

CS+は、RH850 ファミリ、RX ファミリ、V850 ファミリ、RL78 ファミリ、78K0R マイクロコントローラ、78K0 マ イクロコントローラ用の統合開発環境プラットフォームです。

CS+では,設計/コーディング/ビルド/デバッグ/フラッシュ・プログラミングなど,プログラムの開発における 一連の作業を行うことができます。

本マニュアルは、こうした一連のプログラムの開発工程のうち、デバッグ工程について説明します。

注意 E2 エミュレータ Lite(略称: E2 Lite)を使用する場合は、本マニュアル中で「E1」と記載のある部分を「E2 Lite」と読み替えてください。
 E2 エミュレータ(略称: E2)を使用する場合は、本マニュアル中で「E1」と記載のある部分を「E2」と読み替えてください。

この章では、CS+が提供するデバッグ機能の概要について説明します。

1.1 概 要

CS+が提供するデバッグ機能を使用することにより、RXファミリ用に開発されたプログラムを、効率良くデバッグ することができます。

1.2 特 長

次に、CS+が提供するデバッグ機能の特長を示します。

- 各種デバッグ・ツールとの接続 オンチップ・デバッギング・エミュレータ(E1/E20/EZ Emulator),およびシミュレータと組み合わせて使用する ことにより、より快適な開発環境を実現できます。
- C/C++ ソース・テキストと逆アセンブル・テキストの混合表示 1 つのパネル上で、C/C++ ソース・テキストと逆アセンブル・テキストを混合表示することができます。
- ソース・レベル・デバッグと命令レベル・デバッグ
 C/C++ ソース・プログラムに対して、ソース・レベル・デバッグ、または命令レベル・デバッグを行うことができます。
- データフラッシュ・メモリへの書き込み
 選択しているマイクロコントローラがデータフラッシュ・メモリ内蔵品の場合では、通常のメモリ操作と同様のアクセス方法により、データフラッシュ・メモリの内容を表示/変更することができます(シミュレータを除く)。
- リアルタイム表示更新機能
 プログラムの実行が停止した際に、表示情報を自動的に更新するだけでなく、プログラムが実行中の状態であっても、リアルタイムにメモリ/レジスタ/変数の値を表示更新することができます。
- デバッグ環境の保存/復元
 ブレークポイントやイベントの設定情報、ファイルのダウンロード情報、パネルの表示状態/位置などのデバッグ
 環境を保存することができます。



2. 機 能

この章では、CS+を使用したデバッグの手順、およびデバッグに関する主な機能について説明します。

2.1 概 要

CS+を使用した、プログラムの基本的なデバッグ手順は次のとおりです。

照してください。

編」を参照してください。

- (1) CS+ を起動する Windows[®]の[スタート]メニューから CS+ を起動します。
 備者 "CS+ を起動する"についての詳細は、「CS+ 統合開発環境 ユーザーズマニュアル 起動編」を参
- (2) プロジェクトを設定する プロジェクトの新規作成、または既存のプロジェクトの読み込みを行います。

備考 "プロジェクトを設定する"についての詳細は、「CS+ 統合開発環境 ユーザーズマニュアル 起動

- (3) ロード・モジュールを作成する アクティブ・プロジェクトの設定、および使用するビルド・ツールの設定を行ったのち、ビルドを実行することにより、ロード・モジュールを作成します。
 - 備考 CC-RX を使用して "ロード・モジュールを作成する "場合についての詳細は、「CS+ 統合開発環 境 ユーザーズマニュアル CC-RX ビルド・ツール操作編」を参照してください。
- (4) ホスト・マシンとの接続を確認する
 ホスト・マシンに、使用するデバッグ・ツール(E1/E20/EZ Emulator/ シミュレータ)を接続します。
- (5) 使用するデバッグ・ツールを選択する プロジェクトで使用するデバッグ・ツールを選択します。

備考 プロジェクト内で使用するマイクロコントローラにより,選択可能なデバッグ・ツールは異なり ます。

- (6) デバッグ・ツールの動作環境設定を行う
 (5)において選択したデバッグ・ツールのための動作環境を設定します。
 - 【E1】の場合
 - 【E20】の場合
 - 【EZ Emulator】の場合
 - -【シミュレータ】の場合
- (7) CS+にデバッグ・ツールを接続する CS+とデバッグ・ツールの通信を開始します。
- (8) ダウンロードを実行する

 (3) において作成したロード・モジュールを、デバッグ・ツールへダウンロードします。
- (9) ソース・ファイルを表示する ダウンロードしたロード・モジュールの内容(ソース・ファイル)をエディタパネル、または逆アセンブルパ ネルで表示します。
- (10) プログラムを実行する 目的に応じた実行方法により、プログラムを実行します。 なお、実行したプログラムを任意の箇所で停止する場合は、あらかじめブレークポイント/ブレーク・イベント 注を設定しておきます(「2.10.2 任意の場所で停止する(ブレークポイント)」/「2.10.3 任意の場所で停止す る(ブレーク・イベント)【E1】【E20】【EZ Emulator】」参照)。
 - 注 使用するデバッグ・ツールにイベントを設定することにより実現する機能です。イベントを設定 する際には、「2.17.7 イベント設定に関する留意事項」を参照してください。
- (11) プログラムの実行を手動で停止する 実行したプログラムを停止します。 ただし、(10)においてブレークポイント/ブレーク・イベントを設定している場合では、その条件が満たされる と同時にプログラムの実行は自動的に停止します。

- (12) プログラムの実行結果を確認する プログラムを実行することにより取得した各種情報を確認します。
 - メモリ、レジスタ、変数の表示/変更
 - スタックからの関数呼び出し情報の表示
 - 実行履歴の収集^注
 - 実行時間の計測^注
 - カバレッジの測定【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】
 - 注 使用するデバッグ・ツールにイベントを設定することにより実現する機能です。イベントを設定 する際には、「2.17.7 イベント設定に関する留意事項」を参照してください。

以後,必要に応じて (9) ~ (12)を繰り返すことによりデバッグ作業を進めます。

- なお、この際に、プログラムに変更を加えた場合は、(3) および (8) の操作も繰り返す必要があります。
- 備考 1. 上記のほか,次の機能を利用して,プログラムの実行結果の確認を行うことができます。
 - プログラム内へのアクションの設定
 - フック処理を設定する
 - デバッグ・コンソールの使用
- 備考 2. 取得した各種情報をファイルに保存することができます。
 - 逆アセンブル結果の表示内容を保存する
 - メモリの表示内容を保存する
 - CPU レジスタの表示内容を保存する
 - I/O レジスタの表示内容を保存する
 - ローカル変数の表示内容を保存する
 - ウォッチ式の表示内容を保存する
 - コール・スタック情報の表示内容を保存する
 - 実行履歴の表示内容を保存する
- (13) アップロードを実行する 必要に応じ、プログラム(メモリ内容)を任意のファイル形式(ヘキサ・フォーマット/バイナリ・データ・ フォーマットなど)で保存します。
- (14) CS+からデバッグ・ツールを切断する CS+とデバッグ・ツールとの通信を終了します。
- (15) プロジェクト・ファイルを保存する プロジェクトの設定情報をプロジェクト・ファイルに保存します。
 - 備考 "プロジェクト・ファイルを保存する"についての詳細は,「CS+ 統合開発環境 ユーザーズマニュ アル 起動編」を参照してください。



2.2 デバッグを始める前の準備

この節では、作成したプログラムのデバッグを開始するための準備について説明します。

2.2.1 ホスト・マシンとの接続を確認する

使用するデバッグ・ツールごとに、ホスト・マシンとの接続例を示します。

なお、プロジェクトで選択しているマイクロコントローラと接続可能なデバッグ・ツールの関係は次の表のとおりです。

備考 表内における "(Serial)" または "(JTAG)" は、E1/E20 を FINE 通信方式、または JTAG 通信方式で使用す る場合を意味します。

表 2.1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係

マイクロコントローラ	デバッグ・ツール								
	E1(Serial)	E1(JTAG)	E20(Serial)	E20(JTAG)	EZ Emulator	シミュレータ			
RX610, RX621, RX62N, RX62T, RX62G グループ	_	0	_	0	_	0			
RX630, RX631, RX63N, RX634 グループ	0	0	0	0	_	0			
RX63T グループ	0	0	0	0	0	0			
RX110, RX111, RX113 RX210, RX220 グループ	0	_	0	_	0	0			
RX21A, RX211, RX231, RX23W, RX23T, RX24T, RX24U, RX130, RX140, RX13T,RX23E-A グルー プ	0	_	0	_	_	0			
RX71M, RX66x, RX64M, RX65N, RX651, RX65N- 2M, RX651-2M, RX671, RX72T, RX72M, RX72N グループ	0	0	0	0	_	0			

2.2.1.1 【E1】の場合

2.2.1.2 【E20】の場合

2.2.1.3 【EZ Emulator】の場合

2.2.1.4 【シミュレータ】の場合

2.2.1.1 【E1】の場合

ホスト・マシン, E1, および必要に応じてターゲット・ボードを接続します。

ターゲット・システムとの通信方式として、FINE 通信【E1(Serial)】、または JTAG 通信【E1(JTAG)】を選択することができます(「2.3.1 使用するデバッグ・ツールを選択する」参照)。

接続方法についての詳細は,E1 のユーザーズ・マニュアルを参照してください。







2.2.1.2 【E20】の場合

ホスト・マシン, E20, および必要に応じてターゲット・ボードを接続します。 ターゲット・システムとの通信方式として, FINE 通信【E20(Serial)】, または JTAG 通信【E20 (JTAG)】を選択する ことができます(「2.3.1 使用するデバッグ・ツールを選択する」参照)。 接続方法についての詳細は, E20 のユーザーズ・マニュアルを参照してください。







注 ターゲット・システムとの通信方式として, FINE 通信方式【E20(Serial)】を選択した場合, E20 エ ミュレータに付属の 38 ピン /14 ピン変換アダプタを使用してください。 38 ピン /14 ピン変換アダプタを用いた接続方法についての詳細は, E20 のユーザーズ・マニュアルを 参照してください。

2.2.1.3 【EZ Emulator】の場合

ホスト・マシン, EZ Emulator, および必要に応じてターゲット・ボードを接続します。 接続方法についての詳細は, EZ Emulator のユーザーズ・マニュアルを参照してください。







2.2.1.4 【シミュレータ】の場合

ホスト・マシンのみでデバッグ作業を行うことができます(エミュレータなどの接続は不要)。

図 2.5 ホスト・マシンとデバッグ・ツールとの接続例【シミュレータ】





2.3 デバッグ・ツールの動作環境設定

この節では、各デバッグ・ツールの動作環境の設定方法について説明します。

2.3.1 使用するデバッグ・ツールを選択する

デバッグ・ツールの動作環境設定は、使用するデバッグ・ツールに対応したプロパティパネルで行います。 そのため、まず、プロジェクト内で使用するデバッグ・ツールを選択します(使用するデバッグ・ツールはプロジェ クトごとに選択可)。

使用するデバッグ・ツールの選択/切り替えは、プロジェクト・ツリー パネル上の [*品種名 デバッグ・ツール名*(デバッグ・ツール)] ノードを右クリックすることで表示されるコンテキスト・メニューにより行ってください。

図 2.6	使用するデバッグ・	・ツールの選択/	′切り替え
-------	-----------	----------	-------

ブロジェクト・ツリー			×			
2 🕜 🙎 ᠌						
B- C Sample (70 B R5F563T ▲ CC-RX (8 ▲ RX 7251	ジェク 4ExFL リレド・1	シ) (マイクロコントローラ) ソール)				
in- 🛐 7711		使用するデバッグ・ツ	−Jl/(D)	۲	RX	(E2(2)
	2	プロパティ(P)			RX	(E2 Lite(E)
					RX	(E1(Serial)(L)
					RX	(E1(JTAG)(G)
					RX	(E20(Serial)(R)
					RX	(E20(JTAG)(T)
					RX	(EZ Emulator(Z)
				~	RX	(シミュレータ(S)

- **注意** 表示されるコンテキスト・メニューは、プロジェクトで選択しているマイクロコントローラの種類により異なります(「表 2.1 マイクロコントローラの種類と接続可能なデバッグ・ツールの関係」参照)。
- 備考 1. 【E1】【E20】 E1/E20 とターゲット・システムとの通信方式として, FINE 通信を使用する場合は [*品種名* E1(Serial)] / [*品種名* E20(Serial)] を, JTAG 通信を使用する場合は [*品種名* E1(JTAG)] / [*品種名* E20(JTAG)] を選択してください。
- 備考 2. 【EZ Emulator】 EZ Emulator とターゲット・システムの通信方式は, FINE 通信のみとなります。

すでにプロパティ パネルがオープンしている場合,再び [*品種名 デバッグ・ツール名*(デバッグ・ツール)] ノード をクリックすると,選択したデバッグ・ツールのプロパティ パネルに表示が切り替わります。

プロパティ パネルがオープンしていない場合では,同ノードをダブルクリックすることで,該当するプロパティ パネ ルがオープンします。



2.3.2 【E1】の場合

E1 を使用する場合の動作環境の設定を次のプロパティ パネルで行います。

なお、ターゲット・システムとの通信方式(FINE 通信【E1(Serial)】/JTAG 通信【E1(JTAG)】)により、プロパティパネルの内容は異なります。

図 2.7 動作環境設定【E1(Serial)】(プロパテ	・ィ パネル)
-------------------------------	---------

70757	and a second
🔍 RX EI(Serial) のプロパティ	P
▲ 内藏ROM/RAM	
内蔵R0Mサイズ[K/5イト]	2048
内蔵RAMサイズ[IUT/F]	128
データフラッシュ・メモリ・サイズ[Kパイト]	32
a 0090	
メイン・クロック・ソース	EXTAL
メイン・クロック問題連載【MHz】	
動作[图]波数[MH2]	
内蔵フラッシュ・メモリ書を換え時のクロック操作を許可する	LVC v2
▲ Iミュレータ2の指数	
1211-95/J7/LNo.	
〃 ターゲット・ボードとの接続	
131レークから電源体給をする(最大201mA)	1703
通信方式	FINE
FINE# - L Moos]	210010
4 75521	
IDコードの入力モード	IDコードを18曲32桁で指定
10-5 F	SEFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ワークRAM開始アドレス	MIX 1000
0-08AM#4(X[/54]-)	1824
A CPUDWHTE-F	
モード第子時定	シングルチップモード
レジスク設定	ジングルチッグモード
1777	Littlependian T = 2
4 MS75m21	entre enteners s
h 外部フラッシュ定義ファイル	[4]
▶ 許部75ッシュ定義7ァイル 内蔵RON/RAM	(4)



 ■ RVE EN(JTAG) のグロ/パラィ ■ 戸蔵ROM/RAM 内蔵 ROM/PA(F) 2048 内蔵 RAMUYZ(EUS71) 2048 方法 アクフラックコ・メモリ・サイズ(EUS71) 32 4 00 90 メイン・クロック 男法共動(M1k2) かけ、クロック 男法共動(M1k2) かけ、クロック 男法共動(M1k2) 内蔵 アラッシュ・メモリ書を含え時のクロックは操作を許可する L べ、え エミュレータジの相談 エミュレータジの相談 エミュレータジの相談 エミュレータジの「教養 王ミュレータジの「教養 アラッシュ・メモリ書を含え(最大200 mA) L べ、え エミュレータジの考察法結 をする(最大200 mA) 1 いった アラッシュ D つード での入力モード D つード での入力モード D つード の入力モード D つード の入力モード ビ D つード P FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	70757	and the second s
 ● 四道ROM/RAM ○ 内面 ROM/PAM ○ 内面 RAMサイズ(以ドイト) ○ クロラク > クロラク > か(シャクロック・ソース > タレンクロック周期総計(MHz) 物作取決検測(MHz) 物作取決検測(MHz) (火火 ■ I スュレータシジョンメモジ書を持た時のクロック(倫特を許可する) レベ火 ■ I スュレータシジョンメモジ書を持た時のクロック(倫特を許可する) ■ レベ火 ■ I スュレータシジョンメモジ書を持た時のクロック(倫特を許可する) ■ レベ火 ■ I スュレータシジョン(MHz) ■ I スュレータシジション(根本) ■ レータンがら電源(株価をする(最大200mA)) ■ レベ火 ■ I スュレータンがら電源(株価をする(最大200mA)) ■ レベ火 ■ I スュレータンがら電源(株価をする(最大200mA)) ■ レクガットボードを切除る ■ アラクシュ ■ D コード ■ D コード ■ I 田 コードの(人力モード) ■ ロ コード ■ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ	🔍 RX EI(JTAG) ወታቢ/(ምብ	P -
内蔵FROMサイズ[K/54] 2048 内蔵FRAMサイズ[K/54] 126 アークフラッシュ・メモリ・サイズ[K/54] 22 クロック メイン・クロック思え試験[MHz] わ(電波波数[MHz] わ(電波波数[MHz] わ(電波波数[MHz] わ(電波波数[MHz] た) クロックの思え試験[MHz] た) クロックの思え試験[MHz] た) クロックの思え試験[MHz] た) クロックの思え試験[MHz] た) クロックの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのの思え試験[MHz] た) クロックのにないた た) クロックにないた た た) クロックにないた た) クロックにないた た) クロックにないた た) クロッ	▲ 内藏ROM/RAM	and the second
内蔵PAM9イズ[[0771] 128 データフラッシュ・メモリ・サイズ[[0771] 32 クロック メイン・クロック思決批判(MHz] 物作思決数[[MHz] 内蔵フラッシュ・メモリ書書換え時のクロック目像作を許可する レビス エミュレータンの相談 エミュレータンの相談 エミュレータンの相談 エミュレータンの相談 エミュレータンの相談 エミュレータンのも電源供給をする(最大200mA) レビス ゴロードをの入力モード DDコードを163曲32倍で指定 DDコード フークRAM間結アドレス PEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	内藏R0Mサイズ[KFrFF]	2048
プータフラッシュ・メモリ・サイズ(K/パイト) 82 クロック メイン・クロック(思点数(M+k)) かけて見は数(M+k) EXTAL からこうシシュ・メモリ書き換え時のクロック操作を注すする LK/k エミュレータンの相談 LK/k エミュレータンの相談 LK/k エミュレータンの相談 JTAG リードボードとの接続 JTAG リロードの人力モード Dロードを16歳期32桁で加定 ワードの人力モード MC マークAM間話アドレス MC ワートをAM間話アドレス MC ワートをAM間話アドレス MC ワートをAM間話アドレス MC マークAAM間話アドレス MC マークAAM間話アドレス MC ワートをAM間話アドレス MC マークAAM間話アドレス MC マークAAM間話アドレス MC マングルチョウモード Sングルチョウモード レジス保護 Sングルチョウモード レジス保護 Sングルチョウモード レジス保護 Sングルチョウモード レジス保護 Sングルチョウモード ビンディアン Little-endiarデータ	内蔵RAMサイズ[IUT/F]	128
 クロック メイントクロックアレース メイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロックアレース ドイントクロ・クロック目的を使いたい アント・ボートンの接続 エミュレータンの接続 エミュレータンの接続 エミュレータンの接続 エミュレータンの接続 エミュレータンの接続 ビンパ・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボートンの接続 アント・ボーンの接続 アント・ボーンのための アント・ボーンのための<!--</td--><td>データフラッシュ・メモリ・サイズ[Kパイト]</td><td>32</td>	データフラッシュ・メモリ・サイズ[Kパイト]	32
メイン・クロックソース EXTAL メイン・クロック調洗能剤(M+b) (パンス) 物作電波数[(M+b)] (パンス) 内電ブラッシュ・メモリ書き換え時のクロック操作を許可する (パンス) * エミュレータンの掲載 エミュレータンの掲載 エミュレータンの掲載 (パンス) * エミュレータンの掲載 (パンス) コミュレータンの掲載 (パンス) コミュレータンの掲載 (パンス) コミュレータンの掲載 (パンス) コミュレータンの意味着したのできないます。 (パンス) 調査方式 (パンス) リロードのもの言葉 (根大200mA) (パンス) リロードの人力モード (ワートを16週間のできないます。 リロードの人力モード (ワートを16週間のできないます。 リロードホック・ションの日本 (ロートを16週間のできないます。 コートボ (ロートボ	a 9099	
メイン・クロック間にお針(MHz) 物作間を放映数(MHz) 内蔵フラッシュ・メモリ書を持え時のクロック操作を許可する レマンの相談 Iミュレータンの相談 Iミュレータンの相談 Iミュレータンの相談 Iミュレータンの相談 Iミュレータンの相談 Iミュレータンの相談 Iミュレータンの目前 JTAGクロックトボードをの接続 Iミュレータから電源(朱信をする(最大200mA) IS5 フララシュ IDコードの入力モード IDコードの入力モード IDコードの入力モード IDコード P-クRAM間站アドレス ワークRAM間站アドレス ワークRAM間站アドレス マークRAM間はアドレス マークの副師モード モード端子説定 レジス決決定 エンディアン Littlemendianデータ	メイン・クロック・ソース	EXTAL
 物性間波数[MHz] 内蔵フラッシュ・メモリ書き換え時のクロック目倫性を許可する ■ 12.1レータ200 相数 正名ユレータシの男似 No. ● ターグラト・ボードその接続 エスコレータから電源(株相をする(最大200 mA) いていえ ゴ目ならのの意思(最大200 mA) いていえ コTAGクロック(MHz) 15.5 ● フラッシュ ロロードを163曲32倍で指定 IDコードを163曲32倍で指定 IDコード IDコードを163曲32倍で指定 IDコードを163曲32倍で指定 ロロード ● POLOBÉTE FFF ● クリの動作 FF-FF ● POLOBÉTE FFF ● POLO	メイン・クロック思測度数[MHz]	
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する しくべえ	動作間波数[MH2]	
 エミュレータンの相談 エミュレータンの用数 マーゲット・ボードとの接続 エミュレータングド・ボードとの接続 エミュレータングを書換法論をする(最大200mA) レベンミ ゴ酸ゴカボ、	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	LVC v2
エミュレーラジリアル No. タープラト・ボードとの接続 エミュレージがら電源決結をする(最大200 mA) していえ 油油方式 JTAG JTAGりロック(MHz) 16.5 クラウシュ Dロードの入力モード Dロードの入力モード Dロードを16油の2桁で加定 リード デデドドドドドドドドドドドドドドドドドドドドドド フークRAM間站アドレス FMIC フークRAM間站アドレス FMIC マークの時作モード 1000 モード酸溶子酸定 シングルチップモード レジスタ設定 シングルチップモード エンディアン Littlemendianデータ	▲ Iミュレータとの掲載	
ターゲット・ボードとの接続 I3ュレータがら電源決結をする(根大200mA) いくべる 消除方式 JTAG JTAG90ック(MHz) 18.5 マフラウ3 IDコードの入力モード IDコードの入力モード IDコードを16歳回2桁で銀定 IDコードの入力モード IDコードを16歳回2桁で銀定 マークRAM間站アドレス IDコード コークRAM間站アドレス IDコード コークRAM間站アドレス IDコード マークRAM間站アドレス ID124 4 OPUD動作モード シングルチップモード レジス交換定 シングルチップモード エンディアン Littlemendianデータ	IミュレーダシリアルNo.	
エミュレータから電源決結をする(最大200mA) いくえ 通信方式 JTAG JTAGD(2)(MH2) 155 クラウシュ 155 Dコードを1分通数で加定 Dコードを1分通数で加定 Dコード 1001 ワークRAM間站アドレス 1001 ワークRAM間站アドレス 1001 ワークRAM間站アドレス 1001 ワークRAM間站アドレス 1001 ワークRAM間站アドレス 1001 ワークRAM間はアドレス 1024 CPUD的部件モード シングルチップモード ビンディアン Littlemendianデータ 外部フラッシュ 101	〃 ターゲット・ボードとの接続	
通信方式 JTAG9D99(MHz] JTAG JTAG9D99(MHz] 185 DD=Fを0入力モード DD=Fを1638036行金留定 DD=F DD=F DD=Fを1638036行留定 DD=F DD=F DD=Fを1638036行留定 DD=F DD=F DD=Fを1638036行留定 DD=F DD=F DD=F DD=F DD=F FFFFFFFFFFFFFF	エミュレータから電源供給をする(最大200 nA)	110.12
JTAG9099(MH2) 18.5 * フラウシュ Dコードの入力モード IDコードの入力モード Dコードを16歳回2街で銀定 Dコード FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	通信方式	JTAG
	JTAGDD90(MH2)	16.5
IDコードの入力モード IDコードを18歳02桁で加定 IDコード FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	4 7599Ja	
IDDード FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	IDコードの入力モード	IDコードを16進32行で指定
ワークRAM開始アドレス 1000 ワークRAM開始アドレス 1024 CPU00時作モード 1024 モード増子設定 シングルチョブモード レジス決決定 シングルチョブモード エンディアン Littlemendianデータ 外部フラッシュ 102	IDD-F	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ウークRAMサイズ(ひらし) 1024 CPUの動作モード モード端子設定 シングルチップモード レジス交換定 シングルチップモード エンディアン Littlemendianデータ 外部フラッシュ	ワークRAM開始アドレス	1003
 ▲ CPU00001年モード モード端子設定 シングルチップモード レジスタ設定 シングルチップモード エンディアン Littlemendianデータ ▲ 外部フラッシュ 	ワークRAMサイズ(バイト)	1024
モード第子設定 シングルチップモード レジスタ設定 シングルチップモード エンディアン Littlemendianデータ 4 外部フラッシュ	▲ CPUDMAFEE~F	
レジスタ設定 ジングルチャプモード エンディアン Littlemendianデータ 4 外部フラッシュ	モード端子協定	シングルチップモード
エンディアン Littlemendianデータ ・ 外部フラッシュ	レジス処決定	シングルチッグモード
4 外部フラッシュ	エンディアン	Little-endian 7-2
NAMES & STATES AND A STATES AND	4 外部フラッシュ	
▶ 弁部727921定時771ル 【4】	かがフラッシュ定義ファイル	[4]
A BERON/RAM	AERON/RAM	

図 2.8 動作環境設定【E1(JTAG)】(プロパティ パネル)

プロパティ パネル上の該当するタブを選択し、次の設定を順次行ってください。

2.3.2.1 [接続用設定] タブ
2.3.2.2 [デバッグ・ツール設定] タブ
2.3.2.3 [ダウンロード・ファイル設定] タブ
2.3.2.4 [フック処理設定] タブ

2.3.2.1 [接続用設定] タブ

[接続用設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールとの接続に関する設定を行います。

- (1) [内蔵 ROM/RAM]
- (2) [クロック]
- (3) [エミュレータとの接続]
- (4) [ターゲット・ボードとの接続]
- (5) [E2 拡張インタフェース] 【E2】
- (6) [フラッシュ]
- (7) [CPUの動作モード]
- (8) [外部フラッシュ]
- [内蔵 ROM/RAM]
 このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2.9 [内蔵 ROM/RAM] カテゴリ

⊿ 内蔵ROM/RAM

内蔵ROMサイズ[K/fイト]	2048
内蔵RAMサイズ[Kノiイト]	128
データフラッシュ・メモリ・サイズ[K/トイト]	32



- [内蔵 ROM サイズ [K バイト]] エミュレーションする内蔵 ROM サイズを表示します(単位: K バイト)。 このプロパティ値を変更することはできません。
- [内蔵 RAM サイズ [バイト]] エミュレーションする内蔵 RAM サイズを表示します(単位:バイト)。 このプロパティ値を変更することはできません。
- [データフラッシュ・メモリ・サイズ [K バイト]] ・選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します(単位:Kバイ) ト)。 |選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は, [0]を表示します。 このプロパティ値を変更することはできません。
- [クロック] (2)

このカテゴリでは、クロックに関する設定を行います。

注意 E1と接続中にこのカテゴリ内のプロパティを変更することはできません。

[クロック] カテゴリ【HOCO】 図 2.10

4	<u> </u>		
	メイン・クロック・ソース	HOCO	
	動作問波敬[MHz]	25.0000	
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ	

図 2.11 [クロック] カテゴリ【EXTAL】

9099	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作周波数[MHz]	25.0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

- [メイン・クロック・ソース] メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。

内蔵 HOCO を持たないマイクロコントローラの場合は、[EXTAL] を表示します。

- [メイン・クロック周波数 [MHz]] メイン・クロックの周波数(逓信前)を指定します。 EXTAL 周波数値として 0.0001 ~ 99.9999(単位: MHz)の範囲で数値を直接入力します。 入力値の少数第5位以下は切り捨てられ、指定可能範囲外の入力値は、0.0001(0以下だった時)または 99.9999(100以上だった時)に丸め込みます。 このプロパティは、[メイン・クロック・ソース] プロパティで [EXTAL] を指定した場合のみ表示されます。
- [動作周波数 [MHz]] 動作周波数(ICLK)を指定します。 0.0001~999.999(単位:MHz)の範囲で数値を直接入力します。 指定可能範囲外の入力値は、0.0001(0以下だった時)または 999.999(1000以上だった時)に丸め込みます。
- [内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する] 内蔵フラッシュ・メモリ書き換え時に、デバッガによるクロック操作を許可するか否かを指定します。

注意 **[E1]** [はい]を指定した場合,マイクロコントローラの FlashIF クロック(FCLK)が動作保証範囲 外(LOCO やサブクロック動作中など)でデバッガによる内蔵フラッシュ ROM の書き換えを 実施した時, E1 がクロック・ソースを切り替えます。なお, 書き換え完了後は, 元のクロッ ク・ソースに復帰します。 クロック操作を許可した場合、クロック・ソースを切り替えるため、内蔵フラッシュ・メモリ 書き換え中の周辺クロック動作周波数が変更されます。 クロック操作の許可設定は、プログラム実行やステップ実行などの実行系機能を実施した後の 内蔵フラッシュ ROM を書き換える時に有効となります。なお、デバッグ・ツールの接続直後 や CPU リセット発行時は、クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場 合は、強制的にクロック・ソースを切り替えます。

- [エミュレータとの接続] (3) このカテゴリでは、E1とホスト・マシンとの接続に関する設定を行います。
 - E1と接続中にこのカテゴリ内のプロパティを変更することはできません。 注意

図 2.12 [エミュレータとの接続]カテゴリ

- ▲ エミュレータとの接続 エミュレータシリアルNo.
- [エミュレータシリアル No.] 接続されている E1 のシリアル番号をドロップダウン・リストに全て表示します。 接続する E1 のシリアル番号を指定してください。 なお,ドロップダウン・リストはドロップダウン時に毎回更新します。
- (4) [ターゲット・ボードとの接続] このカテゴリでは, E1とターゲット・ボードとの接続に関する設定を行います。

図 2.13 [ターゲット・ボードとの接続] カテゴリ【E1(Serial)】

\sim	ターゲット・ボードとの接続		
	エミュレータから電源供給をする(最大200mA)	はい	
	供給電圧[V]	3.3V	
	通信方式	FINE	
	FINEボーレート[bps]	2000000	

図 2.14 [ターゲット・ボードとの接続] カテゴリ【E1(JTAG)】

~	ターゲット・ボードとの接続
	エトーレー わから 伊沢原 (世紀) た オス/ 月 ナ (

エミュレータから電源供給をする(最大200mA)	(tt)
供給電圧[V]	3.3V
通信方式	JTAG
JTAGクロック[MHz]	16.5

- [エミュレータから電源供給をする(最大 200mA)] E1からターゲット・システムへ電源を供給するか否かを指定します。 電源を供給する場合は[はい]を選択してください。

注意 E1と接続中にこのプロパティを変更することはできません。

- [電源供給インタフェース]【E2】 このプロパティは、[エミュレータから電源供給をする(最大 200mA)] プロパティにおいて、[はい] を選択 した場合のみ表示されます。 エミュレータからターゲット・ボードに電源を供給するインタフェースを選択します。(デフォルト:[USER I/F])。

- [供給電圧]

E1からターゲット・システムへ供給する電圧をドロップダウン・リストより指定します。 このプロパティは、[エミュレータから電源供給をする(最大 200mA)]プロパティにおいて[はい]を指定 した場合のみ表示されます。 なお、指定可能な電圧値はマイクロコントローラにより異なります。

注意

[E1]

E1 と接続中にこのプロパティを変更することはできません。 【E2 Lite】 このプロパティは変更できません。

- [通信方式]

E1 がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。 プロジェクト・ツリー パネルにて使用するデバッグ・ツールを選択する際に "RX E1(Serial)" を選択した場合, このプロパティには [FINE] が表示されます。また、"RX E1(JTAG)" を選択した場合, [JTAG] が表示されます。 E1 ではこのプロパティ値を変更することはできません。"RX E2 Lite", または "RX E2" を選択した場合, この プロパティ値を変更可能かどうかは選択しているマイクロコントローラの種類と通信方式によって異なりま す。マイクロコントローラの種類と通信方式の関係については、「2.2.1 ホスト・マシンとの接続を確認する」 を参照してください。

なお、デバッグ・ツールの選択についての詳細は、「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。

- [JTAG クロック [MHz]] E1 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (JTAG クロック) をドロップ ダウン・リストより指定します。 このプロパティは, [通信方式] プロパティにおいて [JTAG] の場合のみ表示されます。 ドロップダウン・リストには、次の通信速度が表示されます。

- 16.5(デフォルト), 12.38, 6.188, 3.094, 1.547【E1】
- 6.00(デフォルト), 3.00, 1.50 【E2 Lite】
- 16.5(デフォルト), 12.5, 6.25, 3.125, 1.562 【E2】
- **注意 1.** E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。
- **注意 2.** ターゲット・システム上での JTAG 信号の配線長や配線方法によっては,選択した JTAG クロックで通信できない場合があります。この JTAG クロックを下げることによって,通信が出来るようになる場合があります。
- [FINE ボーレート [bps]]

E1 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (FINE ボーレート) をドロッ プダウン・リストより指定します。

- このプロパティは, [通信方式] プロパティにおいて [FINE] の場合のみ表示されます。
- ドロップダウン・リストには、次の通信速度が表示されます。
 - 2000000(デフォルト), 750000, 500000, 250000 【E1】 【E2】
 - 1500000(デフォルト), 750000, 500000, 250000 【E2 Lite】
- **注意 1.** E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。
- **注意 2.** ターゲット・システム上での FINE 信号の配線長や配線方法によっては,選択した FINE ボーレートで通信できない場合があります。この FINE ボーレートを下げることによって,通信が出来るようになる場合があります。

使用しない

(5) [E2 拡張インタフェース] 【E2】
 E2 拡張インタフェースに関する設定を行います。

図 2.15 [E2 拡張インタフェース] カテゴリ【E2】

- E2拡張インタフェース

 E2拡張インタフェースを使用する
- [E2 拡張インタフェースを使用する] E2 拡張インタフェースを使用するか否かを選択します。 E2 拡張インタフェースを使用する場合, E2 拡張インタフェースの電源を選択してください。
- (6) [フラッシュ]

このカテゴリでは、フラッシュ書き換えに関する設定を行います。

注意 E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.16 [フラッシュ] カテゴリ

4 フラッシュ

۰.	////1	
	IDコードの入力モード	IDコードを16進32桁で指定
	IDコード	HER FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	ワークRAM開始アドレス	HER 1000
	ワークRAMサイズ[パイト]	1024

- [ID コードの入力モード] ID コードの入力モードを指定します。

注意 E1と接続中にこのプロパティを変更することはできません。

- [ID コード]

ID コードプロテクトを解除するための ID コードを指定します。

[ID コードの入力モード] プロパティにおいて, [ID コードを 16 進 32 桁で指定]を指定した場合, ID コード は 32 桁の 16 進数で入力してください。また, [ID コードを 16 文字以内で指定 (ASCII)]を指定した場合, ID コードは 16 文字以内の ASCII 文字列で入力してください。

- **注意 1.** E1 と接続中にこのプロパティを変更することはできません。
- 注意 2. ID コードを 32 桁の 16 進数で入力する場合は、32 ビットのデータ単位に並べて入力してくだ さい。 フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフ ラッシュプログラマの ID コード入力の仕様を確認してください。

- 注意 3. ID コードを ASCII 文字列で入力する場合, 16 文字に満たない部分は "0" で入力されます。
- [ワーク RAM 開始アドレス]

デバッガが使用するワーク RAM の配置アドレスを指定します。なお,アドレス値は4バイト単位の値を指定 してください。(入力値が4バイト単位の値でなかった場合は,自動的に補正されます。) 指定したワーク RAM の配置アドレスから [ワーク RAM サイズ [バイト]] プロパティの表示サイズ分は,デ バッガのファームウェアが使用します。

- **注意 1.** E1 と接続中にこのプロパティを変更することはできません。
- **注意 2.** メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが, ワーク RAM に配置する領域は,以下の用途では使用できません。
 - DMA または DTC 機能の転送元・転送先
 - DTC ベクタテーブル,および転送情報配置アドレス
 - DMAC, DTC 起動要因の割り込みベクタ
- [ワーク RAM サイズ [バイト]] デバッガが使用するワーク RAM のサイズを表示します。
- (7) [CPUの動作モード]

このカテゴリでは、エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

注意 E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.17 [CPU の動作モード] カテゴリ

\sim	CPU0001EE.	- 14

CPUQEDIFT - P	
モード端子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ
起動バンクを変更する	はい
起動バンク	バンクロ
	モード端子設定 レジスタ設定 エンディアン 起動バンクを変更する 起動バンク

- [モード端子設定]

マイクロコントローラのモード端子で設定している動作モードを指定します。

- [USB ブートプログラムの消去を許可する] USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブートモードで起動する際に, ユーザブート領域上の USB ブートプログラムを消去するか否かを指定します。 このプロパティは, [モード端子設定] プロパティにおいて [ユーザブートモード] を指定した場合のみ表示 されます。ただし,ユーザブートモードで起動する際に,ユーザブート領域上の USB ブートプログラムを消 去する必要のないマイクロコントローラの場合は表示されません。
- [レジスタ設定] レジスタで設定する動作モードを指定します。 指定可能な動作モードは、マイクロコントローラによって異なります。
- [エンディアン] プロジェクトのエンディアンを表示します。 プロジェクトからエンディアンの設定情報を取得し、その値を表示します。 デバッグ・ツールと切断中のみ指定することができます。
- [起動バンクを変更する]【RX651-2M, RX65N-2M, RX671, RX72M, RX66N, RX72N グループ】 デバッグ・ツールと接続する際に起動バンクを変更するか否かを指定します。 起動バンクを変更して接続する場合は[はい]を選択してください(デフォルトでは[いいえ]が指定されま す)。
- [起動バンク]【RX65N-2M, RX651-2M, RX671, RX72M, RX66N, RX72N グループ】 デバッグ・ツールと接続する際に変更する起動バンクを指定します。 このプロパティは、[起動バンクを変更する]【RX651-2M, RX65N-2M, RX671, RX72M, RX66N, RX72N グ ループ】プロパティにおいて[はい]を指定した場合のみ表示されます。
- (8) [外部フラッシュ] このカテゴリでは、外部フラッシュに関する設定を行います。

このカテゴリ内の設定は、外部フラッシュ・メモリヘダウンロードする場合に必要となります。各プロパティ内の設定についての詳細は、「2.5.2.5 外部フラッシュ・メモリヘダウンロードする【E1】【E20】」を参照してください。

注意 E1 と接続中にこのカテゴリ内のプロパティを変更することはできません。

2.3.2.2 [デバッグ・ツール設定] タブ

[デバッグ・ツール設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールの基本設定を行います。

- (1) [メモリ]
- (2) [実行中のメモリ・アクセス]
- (3) [レジスタ]
- (4) [E2 拡張インタフェース] 【E2】
- (5) [ブレーク]
- (6) [システム]
- (7) [トレース]
- (8) [タイマ]【RX600, RX700 シリーズ】
- (1) [メモリ]

4

このカテゴリでは、メモリに関する設定を行います。

図 2.18 [メモリ] カテゴリ

・ メモリ			
メモリ・マッピング	[23]		
▷ [0]	内藏RAM領域		
▷ [1]	予約領域		
▷ [2]	L/Oレジスタ領域		
⊳ [3]	予約領域		
⊳ [4]	L/Oレジスタ領域		
▷ [5]	I/Oレジスタ領域		
⊳ [6]	予約領域		
▷ [7]	L/Oレジスタ領域		
⊳ [8]	L/Oレジスタ領域		
⊳ [9]	I/Oレジスタ領域		
▶ [10]	データフラッシュ領域		
▷ [11]	予約領域		
▷ [12]	その他のメモリ領域		
▷ [13]	予約領域		
▶ [14]	L/Oレジスタ領域		
▷ [15]	予約領域		
▷ [16]	1/0レジスタ領域		
▷ [17]	予約領域		
▷ [18]	その他のメモリ領域		
⊳ [19]	予約領域		
▷ [20]	その他のメモリ領域		
▶ [21]	予約領域		
▷ [22]	内藏ROM領域		
メモリ書き込み時にベリファイを行う	はい		

- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。 このパネル上でマッピング値を変更することはできません。 I/O 保護領域を追加、または削除する必要がある場合は、[メモリ・マッピング] プロパティを選択すること で設定欄右端に表示される[...] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行い ます。 設定方法についての詳細は、メモリ・マッピング ダイアログの項を参照してください。 このプロパティでは、メモリ領域の個数のみを表示します。 [メモリ・マッピング] プロパティを展開すると、次のサブ項目が表示されます。 - [メモリ種別]

当該領域のメモリ種別を表示します。

RENESAS

各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM ^{注1} およびデータフラッシュ ^{注2} が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。 ^{注 4}
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6//CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM ^{注 3} , FCU ファーム ^{注 3} , ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。
I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

- 注 1. ダウンロード以外の操作(メモリ パネルからの操作やライン・アセンブル)で内蔵フラッシュ ROM 領域を編集した場合,編集内容は次の実行操作時にフラッシュ ROM に書き込まれます。
- 注 2. 消去された状態のデータフラッシュ ROM の内容を参照すると、マイクロコントローラの仕様どおり不定値が表示されます。なお、デバッガからデータフラッシュ ROM の内容を書き換えた場合は、256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。
- 注 3. FCU-RAM 領域の内容をデバッガから書き換えないでください。また, FCU ファーム領域は デバッガから書き換えできません。
- 注 4. 下記のオプション設定メモリを含む領域は、内蔵 RAM 領域として表示されます。なお、本 領域は、メモリパネルで値を設定することはできません。

【RX71M, RX66x, RX65x, RX64M, RX72T グループ】

- シリアルプログラマコマンド制御レジスタ (SPCC)
- OCD/ シリアルプログラマ ID 設定レジスタ (OSIS)
- エンディアン選択レジスタ (MDE)
- オプション機能選択レジスタ 0(OFS0)
- オプション機能選択レジスタ1 (OFS1)

[RX65N-2M, RX651-2M, RX671, RX72M, RX72N, RX66N]

- シリアルプログラマコマンド制御レジスタ(SPCC)
- OCD/ シリアルプログラマ ID 設定レジスタ (OSIS)
- エンディアン選択レジスタ (MDE)
- オプション機能選択レジスタ0(OFS0)
- オプション機能選択レジスタ1(OFS1)
- バンク選択レジスタ(BANKSEL)
- フラッシュアクセスウィンドウ設定レジスタ(FAW)
- [開始アドレス] 当該領域の開始アドレスを表示します。
- [終了アドレス] 当該領域の終了アドレスを表示します。
- [アドレス幅 [ビット]] 当該領域のアクセス幅を表示します。 [メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。

- [エンディアン] 外部領域および I/O レジスタ領域のエンディアンを表示します。

RENESAS

.

[メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。

- **注意** デバッグ・ツールと接続することにより(「2.4.1 CS+にデバッグ・ツールを接続する」参照), 各メモリ種別ごとの詳細表示を行います。
- [メモリ書き込み時にベリファイを行う] メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。 [はい]を指定した場合、ダウンロードした際、ウォッチパネルやメモリパネルで値を変更した際にベリファ イを行います。
- (2) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセスに関する設定を行います。 このカテゴリ内の設定は、リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新 機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

図 2.19 [実行中のメモリ・アクセス] カテゴリ |)】

実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	いいえ
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

- [実行を一瞬停止してアクセスする]

プログラム実行中に、メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。

アクセスを許可する場合は [はい]を選択してください (デフォルトでは [いいえ] が指定されます)。

- [実行中に表示更新を行う]

プログラム実行中に, ウォッチ パネル/メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい]を選択してください (デフォルト)。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [表示更新間隔 [ms]]

このプロパティは, [実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ表示されます。

プログラム実行中に, ウォッチ パネル/メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数(100 ms 未満の端数切り上げ)を指定してください(デフォルトでは [500] が指定されます)。

なお、[実行中に表示更新を行う] プロパティにおいて指定値を[いいえ] から [はい] へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(3) [レジスタ]

このカテゴリでは、プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.20 [レジスタ] カテゴリ

đ	レジスタ	
	実行中にPC表示を行う	(Itc)
	PC表示更新間隔[ms]	500

- [実行中に PC 表示を行う]

このプロパティは、プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。 [いいえ]を選択した場合、実行中のステータス・バーには"実行中"と表示されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. 【RX100 シリーズ】 プログラム実行中の PC 表示機能をサポートしていないため、非表示となります。

注意 3. 消費電流測定中は、このプロパティは[いいえ]に固定となります。

- [PC 表示更新間隔 [ms]]

このプロパティは, [実行中に PC 表示を行う] プロパティにおいて [はい] を指定した場合にのみ表示されます。

プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。 直接入力により、100 ~ 65500 の整数(100ms 未満は端数切り上げ)を指定してください。(デフォルトでは [500] が指定されます。) なお、[実行中に PC 表示を行う] プロパティにおいて指定値を[いいえ] から[はい] へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(4) [E2 拡張インタフェース] 【E2】
 E2 拡張インタフェースに関する設定を行います。

図 2.21 [E2 拡張インタフェース] カテゴリ

¥	E2拡張インタフェース	
¥	外部 刊ガ入力	[2]
	✓ [0]	Ch0 - 立ち上がりエッジ - プログラムを停止する
	チャネル番号	0
	使用する	はい
	入力信号	立ち上がりエッジ
	外部トリガ入力時の動作	プログラムを停止する
	✓ [1]	Ch1 - 立ち上がりエッジ - プログラムを停止する
	チャネル番号	1
	使用する	(ttu)
	入力信号	立ち上がりエッジ
	外部トリガ入力時の動作	プログラムを停止する
¥	外部 阿ガ出力	[2]
	✓ [0]	Ch0 - プログラム停止 - Highパルス - 1
	チャネル番号	0
	使用する	はい
	出力タイミング	プログラム停止
	出力信号	High/f/JJA
	パルス1幅[us]	1
	✓ [1]	Ch1 - プログラム停止 - Highパルス - 1
	チャネル番号	1
	使用する	はい
	出力タイミング	プログラム(停止
	出力信号	Highパルス
	パルス幅[us]	1

- [外部トリガ入力] 外部トリガ入力に関する設定を行います。 チャネル毎に動作を選択してください。

- チャネル番号 チャネル番号を表示します。

- 使用する このチャネル番号の外部トリガ入力を使用するか否かを指定します。
- 入力信号 入力信号を指定します。
- 外部トリガ入力時の動作 外部トリガ入力時の動作を表示します。
- [外部トリガ出力] 外部トリガ出力に関する設定を行います。 チャネル毎に動作を選択してください。
 - チャネル番号 チャネル番号を表示します。
 - 使用する このチャネル番号の外部トリガ出力を使用するか否かを指定します。
 - 出力タイミング 出力タイミングを表示します。
 - 出力信号



出力信号を表示します。

- パルス幅 [us] パルス幅を指定します。

(5) [ブレーク] このカテゴリでは、ブレーク機能に関する設定を行います。

図 2.22 [ブレーク] カテゴリ

- ブレーク 優先的に使用するブレークポイントの種類 ハードウェア・ブレーク
- [優先的に使用するブレークポイントの種類]

エディタ パネル/逆アセンブル パネルにおいて、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。 優先して使用するブレークポイントを使い切った状態で、ブレークポイントを設定した場合、自動的にもう片方のブレークポイントを設定します。

なお,ブレークポイントについての詳細は,「2.10.2 任意の場所で停止する(ブレークポイント)」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します(デフォルト)。

(6) [システム]

2.7.7.7.1

このカテゴリでは,エミュレーションシステムに関する設定を行います。 実行開始前およびブレーク後の指定ルーチン実行についての詳細は,「2.9.4 指定ルーチンを実行する【E1】 【E20】【EZ Emulator】」を参照してください。

図 2.23 [システム]カテゴリ

2776	
内蔵フログラムROMを書き換えるフログラムをデバッグする	いいえ
内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
実行開始前に指定ルーチンを実行する	はい
実行開始前に実行するルーチン	
ブレーク後に指定ルーチンを実行する	はい
ブレーク後に実行するルーチン	
指定ルーチン実行機能用ワークRAM開始アドレス	HEX 1FDD0
指定ルーチン実行機能用ワークRAMサイズ[)ドイト]	560

- [内蔵プログラム ROM を書き換えるプログラムをデバッグする] 内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグ するか否かを指定します。

注意 E1 と接続中にこのプロパティを変更することはできません。

- [内蔵データフラッシュを書き換えるプログラムをデバッグする] 内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど)をデバッグするか否かを指定します。
- 注意 E1 と接続中にこのプロパティを変更することはできません。
- [実行開始前に指定ルーチンを実行する] ユーザ・プログラム実行前に、指定ルーチンを実行する必要があるか否かを指定します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [実行開始前に実行するルーチン] ユーザ・プログラム実行直前に実行するアドレスを指定します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合 のみ表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に指定ルーチンを実行する] ユーザ・プログラムブレーク後に,指定ルーチンを実行する必要があるか否かを指定します。 **注意** プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に実行するルーチン]

ユーザ・プログラムブレーク直後に実行するアドレスを指定します。 このプロパティは、[ブレーク後に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合のみ表示します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [指定ルーチン実行機能用ワーク RAM 開始アドレス]

指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお、アドレス値は4バイト単位の値を 指定してください。(入力値が4バイト単位の値でなかった場合は、自動的に補正されます。) 指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパ ティの表示サイズ分は、デバッガのファームウェアが使用します。 このプロパティは、[実行開始前に指定ルーチンを実行する] プロパティまたは、[ブレーク後に指定ルーチン を実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. ユーザ・プログラムで使用しない領域を指定してください。

- [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] 指定ルーチン実行機能用のワーク RAM のサイズを表示します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティまたは、[ブレーク後に指定ルーチン を実行する]プロパティのいずれかにおいて [はい]が指定された場合のみ表示します。

(7) [トレース]

このカテゴリでは、トレース機能に関する設定を行います。

図 2.24 [トレース]カテゴリ

å	トレース	
	トレース機能の用途	トレース
	トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
	トレース・データ種別	分岐+データアクセス
	データアクセスのバス・マスタ	CPU
	タイム・スタンプ出力	はい
	トレース・クロック・カウント・ソース[MHz]	
	トレース・クロック・カウント・ソースの分周比	分周なし

- [トレース機能の用途] トレース機能の用途として [トレース]を表示します。 このプロパティ値を変更することはできません。
- [トレース・メモリを使い切った後の動作] トレースの取得モードを指定します。 取得モードは、次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても, トレース・データ の上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら, トレース・データ の書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

- [トレース・データ種別]

トレース取得するデータ種別をドロップダウン・リストより指定します。 指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。 ドロップダウン・リストには、次のデータ種別が表示されます。

- -【RX600, RX700 シリーズ】 分岐 , 分岐+データアクセス , データアクセス
- -【RX100, RX200 シリーズ】 分岐, データアクセス
- [データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グルー プ】

RENESAS

このプロパティは、[トレース・データ種別] プロパティにおいて[分岐+データアクセス]または[データ アクセス]を指定した場合にのみ表示されます。 データアクセスのバス・マスタをドロップダウン・リストより指定します。 ドロップダウン・リストには、次のバス・マスタが表示されます。

- CPU(デフォルト), DMAC/DTC

データアクセスのトレース結果については、指定したバス・マスタからのトレース結果のみをトレースパネル に出力します。

- 注意 1. プログラム実行中にこのプロパティを変更することはできません。
- **注意 2.** データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は、[データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】プロパティは表示されません。その場合、バス・マスタは[CPU] 固定となります。
- [タイム・スタンプ出力]

収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。 このプロパティは、マイクロコントローラのシリーズや[トレース・データ種別]プロパティでの指定値に よって変更可否が次のように切り替わります。

マイクロコントローラ	データ種別	変更可否
RX600, RX700 シリーズ 分岐, 分岐 + データアクセス, データアクセス		プログラム停止時のみ変更可能
RX200 シリーズ	分岐	変更不可 [いいえ] が固定表示
	データアクセス	プログラム停止時のみ変更可能
RX100 シリーズ	分岐, データアクセス	変更不可 [いいえ] が固定表示

- [トレース・クロック・カウント・ソース [MHz]]

このプロパティは, [タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されます。

カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。

直接入力により,0.0001 - 999.999 の間で指定してください。

なお、プロパティが空欄の場合、[接続用設定] タブの [クロック] カテゴリ内の [動作周波数 [MHz]] プロパティの設定値がカウント・ソースとして代用されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. プロパティの設定値は, 収集済みのトレース・データには反映しません。プロパティ設定後に 収集したトレース・データから反映します。

- [トレース・クロック・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M グループ】 このプロパティは, [タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されま す。

タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。

ドロップダウン・リストには、次の分周比が表示されます。

- 分周なし(デフォルト), 16 分周, 256 分周, 4096 分周

[トレース・クロック・カウント・ソース [MHz]] プロパティで指定した周波数をさらに指定した値で分周 (周波数を 1/n 倍に変換)した値が実際のタイム・スタンプの単位(1カウントあたりの周波数)になります。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は、[トレース・クロック・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M グループ】プロパティは表示されません。その場合、[分周なし]固定となります。

(8) [タイマ] 【RX600, RX700 シリーズ】 このカテゴリでは,タイマ機能に関する設定を行います。 図 2.25 [タイマ]カテゴリ **タイマ** カウンタを64ビットで使用する いいえ

- [カウンタを 64 ビットで使用する] 計測カウンタを 32 ビット ×2 本で使用するか, 64 ビット ×1 本で使用するかを指定します。

2.3.2.3 [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、デバッグ・ツールにダウンロードを実行する際の設定を行います。 各カテゴリ内の設定についての詳細は、「2.5.1 ダウンロードを実行する」を参照してください。

2.3.2.4 [フック処理設定] タブ

[フック処理設定]タブでは、デバッグ・ツールにフック処理の設定を行います。 フック処理、および各カテゴリ内の設定についての詳細は、「2.18 フック処理を設定する」を参照してください。



2.3.3 【E20】の場合

E20 を使用する場合の動作環境の設定を次のプロパティ パネルで行います。 なお、ターゲット・システムとの通信方式(FINE 通信【E20(Serial)】/JTAG 通信【E20(JTAG)】)により、プロパ ティ パネルの内容は異なります。

		anna ann an a
🔍 RX E20(S/	eriaŭ መቻር/ለምብ	- a
· 内藏RON	/RAM	
内蔽ROM	#4x0886F1	2048
内蘭RAM	9イズ[10]〒1月	128
9-9799	ウェッメモリ・サイズ[0パイト]	32
4 2092		
メイン・クロ:	カ・ソース	EXTAL
メイン・クロ・	r与图(波敏[MHz]	
動作間波到	8(MHz]	
内蔵 フラッ	シュ・メモリ書き換え時のクロック操作を許可する	LXXX
4 I311-9	2の指統	
I821/-9	5/J7/1/No.	
+ 4-5-5-1-7	ポードとの接続	
エミュレーク	から電源供給をする(最大200 nA)	いいえ
通信方式		FINE
FINEボール	/	2300300
* 75591		
100-F-02	入力モード	10コードを16曲3265で指定
IDコード		FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7-5RAM	間站アドレス	1003
0-bRAM	サイズびらイト】	1024
 CPUIDINF 	FE-F	
モード菓子	的定	シングルチップモード
レジス列始	Ē	シングルチッグモード
エンディアン		Littlemendianデータ
4 外部フラッ	21	
ALL ADDRESS OF	21定義ファイル	(4)



図 2.27	動作環境設定	【E20(JTAG)】	(プロパティ	パネル)

2048 128 82 EXTAL
2048 128 82 EXTAL
2048 128 82 EXTAL
128 32 EXTAL
82 EXTAL
EXTAL
EXTAL
LXXX
11012
JTAG
16.5
IDコードを18連3265で指定
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
HEK 1000
1024
シングルチップモード
シングルチップモード
Littlerendianデータ
(4)

プロパティ パネル上の該当するタブを選択し、次の設定を順次行ってください。

2.3.3.1 [接続用設定] タブ
2.3.3.2 [デバッグ・ツール設定] タブ
2.3.3.3 [ダウンロード・ファイル設定] タブ
2.3.3.4 [フック処理設定] タブ

2.3.3.1 [接続用設定] タブ

[接続用設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールとの接続に関する設定を行います。

- (1) [内蔵 ROM/RAM]
- (2) [クロック]
- (3) [エミュレータとの接続]
- (4) [ターゲット・ボードとの接続]
- (5) [フラッシュ]
- (6) [CPU の動作モード]
- (7) [外部フラッシュ]
- [内蔵 ROM/RAM]
 このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2.28 [内蔵 ROM/RAM] カテゴリ

4	内蔵ROM/RAM	
	内蔵ROMサイズ[Kバイト]	2048
	内蔵RAMサイズ[Kノテイト]	128
	データフラッシュ・メモリ・サイズ[Kバイト]	32

- [内蔵 ROM サイズ [K バイト]]

エミュレーションする内蔵 ROM サイズを表示します(単位: K バイト)。 このプロパティ値を変更することはできません。

- [内蔵 RAM サイズ [バイト]] エミュレーションする内蔵 RAM サイズを表示します(単位:バイト)。 このプロパティ値を変更することはできません。
- [データフラッシュ・メモリ・サイズ [K バイト]]
 選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します(単位:Kバイト)。
 選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は、[0]を表示します。
 このプロパティ値を変更することはできません。

(2) [クロック]

このカテゴリでは、クロックに関する設定を行います。

注意 E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.29 [クロック] カテゴリ【HOCO】

4	クロック		
	メイン・クロック・ソース	HOCO	
	動作問波数[MHz]	25.0000	
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ	

図 2.30 [クロック] カテゴリ【EXTAL】

ø	00 9 0			
	メイン・クロック・ソース	EXTAL		
	メイン・クロック周波数[MHz]	12.5000		
	動作問波数[MHz]	25.0000		
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ		

- [メイン・クロック・ソース] メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。 内蔵 HOCO を持たないマイクロコントローラの場合は, [EXTAL] を表示します。
- [メイン・クロック周波数 [MHz]]
 メイン・クロックの周波数(逓信前)を指定します。
 EXTAL 周波数値として 0.0001 ~ 99.9999 (単位: MHz)の範囲で数値を直接入力します。
 入力値の少数第5位以下は切り捨てられ、指定可能範囲外の入力値は、0.0001(0以下だった時)または
 99.9999(100以上だった時)に丸め込みます。
 このプロパティは、[メイン・クロック・ソース]プロパティで [EXTAL]を指定した場合のみ表示されます。
- [動作周波数 [MHz]] 動作周波数 (ICLK)を指定します。 0.0001 ~ 999.999(単位:MHz)の範囲で数値を直接入力します。 指定可能範囲外の入力値は、0.0001(0以下だった時)または 999.999(1000以上だった時)に丸め込みます。
- [内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する] 内蔵フラッシュ・メモリ書き換え時に、デバッガによるクロック操作を許可するか否かを指定します。

注意 【E20】

[はい]を指定した場合、マイクロコントローラの FlashIF クロック(FCLK)が動作保証範囲 外(LOCO やサブクロック動作中など)でデバッガによる内蔵フラッシュ ROM の書き換えを 実施した時、E20 がクロック・ソースを切り替えます。なお、書き換え完了後は、元のクロッ ク・ソースに復帰します。

クロック操作を許可した場合、クロック・ソースを切り替えるため、内蔵フラッシュ・メモリ 書き換え中の周辺クロック動作周波数が変更されます。

クロック操作の許可設定は、プログラム実行やステップ実行などの実行系機能を実施した後の 内蔵フラッシュ ROM を書き換える時に有効となります。なお、デバッグ・ツールの接続直後 や CPU リセット発行時は、クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場 合は、強制的にクロック・ソースを切り替えます。

- (3) [エミュレータとの接続] このカテゴリでは、E20とホスト・マシンとの接続に関する設定を行います。
 - 注意 E20と接続中にこのカテゴリ内のプロパティを変更することはできません。

(4)

図 2.31	[エミュレータとの接続]カテゴ	IJ		
4 I31/-	タとの接続			
131/5	-95797771 No.			
- [エミュレ 接続されて 接続する [なお,ドロ	ータシリアル No.] Cいる E20 のシリアル番号をドロ E20 のシリアル番号を指定してく コップダウン・リストはドロップグ	ップダウン・リストに全て表示します ださい。 ダウン時に毎回更新します。	- 0	
[ターゲッ このカテゴ!	ト・ボードとの接続] リでは,E20 とターゲット・ボー	ドとの接続に関する設定を行います。		
図 2.32	[ターゲット・ボードとの接続]	カテゴリ【E20(Serial)】		
4 &-577h	・ボードとの接続			
エミュレー 通信方式	タから電源供給をする(最大200mA) た	C)C)Z FINE		
FINE [#]	ν ·ν-ト[bps]	2000000		
図 2.33	[ターゲット・ボードとの接続]	カテゴリ【E20(JTAG)】		
▲ ターゲット	・ボードとの接続			
エミュレー 通信方式	タから電源供給をする(最大200mA)	UNZ JITAG		
JTAG/D	יאל [MHz]	16.5		
 E20は電源供給機能をサポートしていません。 [通信方式] E20がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。 プロジェクト・ツリーパネルにて使用するデバッグ・ツールを選択する際に "RX E20(Serial)" を選択した場合, このプロパティには [FINE] が表示されます。また, "RX E20(JTAG)" を選択した場合, [JTAG] が表示されます。 このプロパティ値を変更することはできません。 なお、デバッグ・ツールの選択についての詳細は、「2.3.1 使用するデバッグ・ツールを選択する」を参照してください。 [JTAG クロック [MHz]] E20 がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (JTAG クロック)をドロップダウン・リストより指定します。 このプロパティは、[通信方式] プロパティにおいて [JTAG] の場合のみ表示されます。 ドロップダウン・リストには、次の通信速度が表示されます。 				
- 16.5(デ	フォルト), 12.38, 6.188, 3.094, 1	.547		
注意 1.	E20 と接続中にこのフロパテ			
注意 2.	ターゲット・システム上での ロックで通信できない場合が 来るようになる場合がありま	JTAG 信号の配線長や配線方法によっ あります。この JTAG クロックを下い す。	っては,選択した JTAG ク げることによって,通信が出	
- [FINE ボ- E20 がタ- プダウン・ このプロ/ ドロップタ	-レート [bps]] -ゲット・システム上のマイクロ: ・リストより指定します。 ペティは、[通信方式] プロパティ ダウン・リストには、次の通信速度	コントローラと通信する際の通信速度 において [FINE] の場合のみ表示され 度が表示されます。	(FINE ボーレート)をドロッ ます。	
- 200000	0(デフォルト), 750000, 500000,	250000		
注意 1.	E20 と接続中にこのプロパテ	ィを変更することはできません。		
注意 2.	ターゲット・システム上での レートで通信できない場合が 出来るようになる場合があり	FINE 信号の配線長や配線方法によっ あります。この FINE ボーレートを下 ます。	っては,選択した FINE ボー 「げることによって,通信が	

RENESAS

(5) [フラッシュ] このカテゴリでは、フラッシュ書き換えに関する設定を行います。

図 2.34 [フラッシュ] カテゴリ

	_	-		×	
4	_	_	-	- 1	
-	-	-		~	1

IDコードの入力モード	IDコードを16進32桁で指定
IDコード	HER FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ワークRAM開始アドレス	HER 1000
ワークRAMサイズ[バイト]	1024

- [ID コードの入力モード]

IDコードの入力モードを指定します。

E20と接続中にこのプロパティを変更することはできません。 注意

- [ID コード]

ID コードプロテクトを解除するための ID コードを指定します。

[ID コードの入力モード] プロパティにおいて, [ID コードを 16 進 32 桁で指定]を指定した場合, ID コード は 32 桁の 16 進数で入力してください。また, [ID コードを 16 文字以内で指定 (ASCII)] を指定した場合, ID コードは 16 文字以内の ASCII 文字列で入力してください。

- E20と接続中にこのプロパティを変更することはできません。 注意 1.
- ID コードを 32 桁の 16 進数で入力する場合は、32 ビットのデータ単位に並べて入力してくだ 注意 2. さい。 フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフ
- 注意 3. ID コードを ASCII 文字列で入力する場合、16 文字に満たない部分は "0" で入力されます。

ラッシュプログラマの ID コード入力の仕様を確認してください。

ID コードを含んだプログラムをダウンロードした場合、ID コードは 注意 4.

- [ワーク RAM 開始アドレス] デバッガが使用するワーク RAM の配置アドレスを指定します。なお,アドレス値は 4 バイト単位の値を指定 してください。(入力値が4バイト単位の値でなかった場合は、自動的に補正されます。) 指定したワーク RAM の配置アドレスから [ワーク RAM サイズ [バイト]] プロパティの表示サイズ分は,デ バッガのファームウェアが使用します。

E20と接続中にこのプロパティを変更することはできません。 注意 1.

注意 2. メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが、 ワーク RAM に配置する領域は、以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル、および転送情報配置アドレス
- DMAC, DTC 起動要因の割り込みベクタ

- [ワーク RAM サイズ [バイト]] デバッガが使用するワーク RAM のサイズを表示します。

(6) [CPU の動作モード]

このカテゴリでは、エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

E20と接続中にこのカテゴリ内のプロパティを変更することはできません。 注意

図 2.35 [CPU の動作モード] カテゴリ

\sim	CPU0001EE-	ĸ
	VEDUX	· ·

~	CPUの動作モード		
	モード端子設定	シングルチップモード	
	レジスタ設定	シングルチップモード	
	エンディアン	Little-endianデータ	
	起動バンクを変更する	はい	
	起動バンク	バンクロ	

- [モード端子設定]

マイクロコントローラのモード端子で設定している動作モードを指定します。
- [USB ブートプログラムの消去を許可する] USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブートモードで起動する際に, USB ブートプログラムを消去するか否かを指定します。 このプロパティは, [モード端子設定]プロパティにおいて [ユーザブートモード]を指定した場合のみ表示 されます。ただし,ユーザブートモードで起動する際に,ユーザブート領域上の USB ブートプログラムを消 去する必要のないマイクロコントローラの場合は表示されません。
- [レジスタ設定] レジスタで設定する動作モードを指定します。 指定可能な動作モードは、マイクロコントローラによって異なります。
- [エンディアン] プロジェクトのエンディアンを表示します。 プロジェクトからエンディアンの設定情報を取得し、その値を表示します。 デバッグ・ツールと切断中のみ指定することができます。
- [起動バンクを変更する]【RX651-2M, RX65N-2M, RX671, RX72M, RX66N, RX72N グループ】 デバッグ・ツールと接続する際に起動バンクを変更するか否かを指定します。 起動バンクを変更して接続する場合は[はい]を選択してください(デフォルトでは[いいえ]が指定されま す)。
- [起動バンク]【RX65N-2M, RX651-2M, RX671, RX72M, RX66N, RX72N グループ】 デバッグ・ツールと接続する際に変更する起動バンクを指定します。 このプロパティは、[起動バンクを変更する]【RX651-2M, RX65N-2M, RX671, RX72M, RX66N, RX72N グ ループ】プロパティにおいて[はい]を指定した場合のみ表示されます。
- (7) [外部フラッシュ] このカテゴリでは、外部フラッシュに関する設定を行います。 このカテゴリ内の設定は、外部フラッシュ・メモリへダウンロードする場合に必要となります。各プロパティ内の設定についての詳細は、「2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】」を参照してください。

注意 E20 と接続中にこのカテゴリ内のプロパティを変更することはできません。

2.3.3.2 [デバッグ・ツール設定] タブ

[デバッグ・ツール設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールの基本設定を行います。

- (1) [メモリ]
- (2) [実行中のメモリ・アクセス]
- (3) [レジスタ]
- (4) [ブレーク]
- (5) [システム]
- (6) [トレース]
- (7) [タイマ] 【RX600, RX700 シリーズ】
- (8) [カバレッジ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】
- (1) [メモリ] このカテゴリでは、メモリに関する設定を行います。



4	メモリ	
4	メモリ・マッピング	[23]
	▷ [0]	内藏RAM領域
	▷ [1]	予約領域
	▷ [2]	I/Oレジスタ領域
	⊳ [3]	予約領域
	⊳ [4]	1/0レジスタ領域
	▷ [5]	I/Oレジスタ領域
	▷ [6]	予約領域
	▷ [7]	1/0レジスタ領域
	⊳ [8]	I/Oレジスタ領域
	▷ [9]	I/Oレジスタ領域
	▷ [10]	データフラッシュ領域
	▷ [11]	予約領域
	▷ [12]	その他のメモリ領域
	▷ [13]	予約領域
	▷ [14]	I/Oレジスタ領域
	▷ [15]	予約領域
	▷ [16]	1/0レジスタ領域
	▷ [17]	予約領域
	▷ [18]	その他のメモリ領域
	▷ [19]	予約領域
	▷ [20]	その他のメモリ領域
	▶ [21]	予約領域
	▷ [22]	内藏ROM領域
	メモリ書き込み時にベリファイを行う	はい

図 2.36 [メモリ] カテゴリ

- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。 このパネル上でマッピング値を変更することはできません。 I/O 保護領域を追加、または削除する必要がある場合は、[メモリ・マッピング]プロパティを選択すること で設定欄右端に表示される […] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行い ます。 設定方法についての詳細は、メモリ・マッピングダイアログの項を参照してください。 このプロパティでは、メモリ領域の個数のみを表示します。

- [メモリ・マッピング]プロパティを展開すると、次のサブ項目が表示されます。
- [メモリ種別]

当該領域のメモリ種別を表示します。 各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM ^{注1} およびデータフラッシュ ^{注2} が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。 ^{注 4}
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6//CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM ^{注 3} , FCU ファーム ^{注 3} , ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。
I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

注 1. ダウンロード以外の操作(メモリパネルからの操作やライン・アセンブル)で内蔵フラッ シュ ROM 領域を編集した場合,編集内容は次の実行操作時にフラッシュ ROM に書き込ま れます。

RENESAS

- 注 2. 消去された状態のデータフラッシュ ROM の内容を参照すると、マイクロコントローラの仕様どおり不定値が表示されます。なお、デバッガからデータフラッシュ ROM の内容を書き換えた場合は、256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。
- 注 3. FCU-RAM 領域の内容をデバッガから書き換えないでください。また, FCU ファーム領域は デバッガから書き換えできません。
- 注 4. 下記のオプション設定メモリを含む領域は、内蔵 RAM 領域として表示されます。なお、本 領域は、メモリパネルで値を設定することはできません。

【RX71M, RX66x, RX65x, RX64M, RX72T グループ】

- シリアルプログラマコマンド制御レジスタ(SPCC)
- OCD/ シリアルプログラマ ID 設定レジスタ (OSIS)
- エンディアン選択レジスタ (MDE)
- オプション機能選択レジスタ0(OFS0)
- オプション機能選択レジスタ1(OFS1)

[RX65N-2M, RX651-2M, RX671, RX72M, RX72N, RX66N]

- シリアルプログラマコマンド制御レジスタ (SPCC)
- OCD/ シリアルプログラマ ID 設定レジスタ (OSIS)
- エンディアン選択レジスタ(MDE)
- オプション機能選択レジスタ0(OFS0)
- オプション機能選択レジスタ1(OFS1)
- バンク選択レジスタ(BANKSEL)
- フラッシュアクセスウィンドウ設定レジスタ(FAW)
- [開始アドレス] 当該領域の開始アドレスを表示します。
- [終了アドレス] 当該領域の終了アドレスを表示します。
- [アドレス幅 [ビット]] 当該領域のアクセス幅を表示します。 [メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。
- [エンディアン] 外部領域および I/O レジスタ領域のエンディアンを表示します。 [メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。
- **注意** デバッグ・ツールと接続することにより(「2.4.1 CS+にデバッグ・ツールを接続する」参照), 各メモリ種別ごとの詳細表示を行います。
- [メモリ書き込み時にベリファイを行う] メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。 [はい]を指定した場合、ダウンロードした際、ウォッチ パネルやメモリ パネルで値を変更した際にベリファ イを行います。
- (2) [実行中のメモリ・アクセス] このカテゴリでは、プログラム実行中におけるメモリ・アクセスに関する設定を行います。 このカテゴリ内の設定は、リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新 機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。
 - 図 2.37 [実行中のメモリ・アクセス] カテゴリ【E20(Serial)】

đ	実行中のメモリ・アクセス		
	実行を一瞬停止してアクセスする	いいえ	
	実行中に表示更新を行う	(\$6)	
	表示更新間隔[ms]	500	

,



図 2.38 [実行中のメモリ・アクセス] カテゴリ【E20(JTAG)【RX600, RX700 シリーズ】】

・ 実行中のメモリ・アクセス	
実行を一瞬停止してアクセスする	はい
実行中に表示更新を行う	はい
表示更新聞隔[ms]	500
リアルタイム表示更新を自動設定する	はい

- [実行を一瞬停止してアクセスする]

プログラム実行中に、メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。

アクセスを許可する場合は [はい]を選択してください (デフォルトでは [いいえ] が指定されます)。



- [実行中に表示更新を行う]

プログラム実行中に, ウォッチ パネル/メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい] を選択してください(デフォルト)。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [表示更新間隔 [ms]]

このプロパティは、[実行中に表示更新を行う]プロパティにおいて[はい]を指定した場合にのみ表示され ます。

プログラム実行中に,ウォッチ パネル/メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の範囲の整数(100 ms 未満の端数切り上げ)を指定してください(デフォル トでは [500] が指定されます)。

なお、[実行中に表示更新を行う] プロパティにおいて指定値を[いいえ]から[はい]へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [リアルタイム表示更新を自動設定する]

RRM 領域を自動的に設定するか否かを指定します。 このプロパティは、[トレース機能の用途]プロパティにおいて[リアルタイム RAM モニタ]を指定した場合 にのみ表示され、[トレース]を指定した場合は[いいえ]の状態で表示されません。

注意 プログラム実行中にこのプロパティを変更することはできません。

(3) [レジスタ]

このカテゴリでは、プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.39 [レジスタ] カテゴリ

đ	レジスタ	
	実行中にPC表示を行う	はい
	PC表示更新間隔[ms]	500

- [実行中に PC 表示を行う]

このプロパティは、プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。 [いいえ]を選択した場合、実行中のステータス・バーには "実行中"と表示されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. 【RX100 シリーズ】

プログラム実行中の PC 表示機能をサポートしていないため、非表示となります。

- [PC 表示更新間隔 [ms]]

このプロパティは, [実行中に PC 表示を行う] プロパティにおいて [はい] を指定した場合にのみ表示されます。

プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。

直接入力により、100 ~ 65500 の整数(100ms 未満は端数切り上げ)を指定してください。(デフォルトでは [500] が指定されます。)

なお、[実行中に PC 表示を行う] プロパティにおいて指定値を [いいえ] から [はい] へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(4) [ブレーク] このカテゴリでは、ブレーク機能に関する設定を行います。

図 2.40 [ブレーク]カテゴリ

▲ ブレーク 優先的に使用するブレークポイントの種類 ハードウエア・ブレーク

- [優先的に使用するブレークポイントの種類] エディタパネル/逆アセンブルパネルにおいて、マウスのワンクリック操作でブレークポイントを設定する 際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。 優先して使用するブレークポイントを使い切った状態で、ブレークポイントを設定した場合、自動的にもう片 方のブレークポイントを設定します。 なお,ブレークポイントについての詳細は,「2.10.2 任意の場所で停止する(ブレークポイント)」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します(デフォルト)。

(5) [システム]

このカテゴリでは,エミュレーションシステムに関する設定を行います。 実行開始前およびブレーク後の指定ルーチン実行についての詳細は,「2.9.4 指定ルーチンを実行する【E1】 【E20】【EZ Emulator】」を参照してください。

図 2.41 [システム]カテゴリ

▲ システム

2777	
内蔵プログラムROMを書き換えるプログラムをデバッグする	いいえ
内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
実行開始前に指定ルーチンを実行する	はい
実行開始前に実行するルーチン	
ブレーク後に指定ルーチンを実行する	はい
ブレーク後に実行するルーチン	
指定ルーチン実行機能用ワークRAM開始アドレス	HEX 1FDD0
指定ルーチン実行機能用ワークRAMサイズ[ノテイト]	560

- [内蔵プログラム ROM を書き換えるプログラムをデバッグする] 内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグ するか否かを指定します。

注意 E20 と接続中にこのプロパティを変更することはできません。

- [内蔵データフラッシュを書き換えるプログラムをデバッグする] 内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど)をデバッグするか否かを指定します。

注意 E20 と接続中にこのプロパティを変更することはできません。

- [実行開始前に指定ルーチンを実行する]

ユーザ・プログラム実行前に、指定ルーチンを実行する必要があるか否かを指定します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [実行開始前に実行するルーチン] ユーザ・プログラム実行直前に実行するアドレスを指定します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合 のみ表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に指定ルーチンを実行する] ユーザ・プログラムブレーク後に,指定ルーチンを実行する必要があるか否かを指定します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に実行するルーチン] ユーザ・プログラムブレーク直後に実行するアドレスを指定します。 このプロパティは、[ブレーク後に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合 のみ表示します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [指定ルーチン実行機能用ワーク RAM 開始アドレス]

指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお、アドレス値は4バイト単位の値を 指定してください。(入力値が4バイト単位の値でなかった場合は、自動的に補正されます。) 指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパ ティの表示サイズ分は、デバッガのファームウェアが使用します。 このプロパティは、[実行開始前に指定ルーチンを実行する] プロパティまたは、[ブレーク後に指定ルーチン を実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. ユーザ・プログラムで使用しない領域を指定してください。

- [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] 指定ルーチン実行機能用のワーク RAM のサイズを表示します。 このプロパティは, [実行開始前に指定ルーチンを実行する] プロパティまたは, [ブレーク後に指定ルーチン を実行する] プロパティのいずれかにおいて [はい] が指定された場合のみ表示します。

(6) [トレース]

このカテゴリでは、トレース機能に関する設定を行います。

図 2.42 [トレース] カテゴリ

✓ トレース

rv-x	
トレース機能の用途	トレース
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・データ種別	分岐+データアクセス(アクセス・データなし)
アクセス・データを取得しない領域の開始アドレス	HEX ()
アクセス・データを取得しない領域の終了アドレス	HEN SFFFF
データアクセスのバス・マスタ	CPU
外部トレース出力	CPU実行優先
トレース・メモリ・サイズ[M/iイト]	1
タイム・スタンプ出力	はい
トレース・クロック・カウント・ソース[MHz]	
トレース・クロック・カウント・ソースの分周比	分周なし
外部トレース出力 トレース・メモリ・サイズ[M/パイト] タイム・スタンプ出力 トレース・クロック・カウント・ソース[MHz] トレース・クロック・カウント・ソースの分周比	CPU実行優先 1 はい 分周なし

- [トレース機能の用途]

トレース機能を利用したリアルタイム RAM モニタ機能 (RRM 機能)として使用するか否かを指定します。 [リアルタイム RAM モニタ]を指定した場合、一部のトレース機能が使用できなくなります。 トレース機能の制限事項についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」 を参照してください。

なお、このプロパティは、プログラム実行停止中のみ変更可能です。

注意 リアルタイム RAM モニタ機能を使用する場合は, E20 とターゲット・ボードを 38 ピンで JTAG 接続している必要があります。それ以外の接続の場合, あるいは, 38 ピン/14 ピン変換ア ダプタを使用している場合は[トレース]を選択してください。

- [トレース・メモリを使い切った後の動作]

トレースの取得モードを指定します。

取得モードは、次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても, トレース・データ の上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら, トレース・データ の書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

注意 【E20(JTAG)【RX600, RX700 シリーズ】】

[トレース機能の用途] プロパティにおいて、[リアルタイム RAM モニタ]を指定した場合、プロパティ値は[トレース・メモリを上書きし実行を続ける]と表示され変更不可状態となります。

- [トレース・データ種別]

トレース取得するデータ種別をドロップダウン・リストより指定します。 指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。 ドロップダウン・リストには、次のデータ種別が表示されます。

- -【RX66x, RX65N-2M, RX651-2M, RX671, RX72M, RX72T, RX72N グループ】 分岐, 分岐+データアクセス, データアクセス, 分岐+データアクセス (アクセス・データなし), データア クセス (アクセス・データなし)
- -【RX600, RX700 シリーズ】 分岐 , 分岐+データアクセス , データアクセス
- -【RX100, RX200 シリーズ】 分岐 , データアクセス

RENESAS

- **注意**【E20(JTAG)【RX600, RX700 シリーズ】】 [トレース機能の用途]プロパティにおいて,[リアルタイム RAM モニタ]を指定した場合,プ ロパティ値は[データアクセス]と表示され変更不可状態となります。
- [アクセス・データを取得しない領域の開始アドレス]【RX65N-2M, RX651-2M, RX66T, RX671, RX72M, RX72T, RX72N, RX66N グループ】 アクセス・データを取得しない領域の開始アドレスを指定します。
- [アクセス・データを取得しない領域の終了アドレス]【RX65N-2M, RX651-2M, RX66T, RX671, RX72M, RX72T, RX72N, RX66N グループ】 アクセス・データを取得しない領域の終了アドレスを指定します。
- [データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グルー

プ】 データアクセスを発生させたバス・マスタを指定します。 このプロパティは、[トレース・データ種別]プロパティにおいて[分岐+データアクセス]または[データ アクセス]を指定した場合にのみ表示されます。

ドロップダウン・リストには、次のバス・マスタが表示されます。

- CPU(デフォルト), DMAC/DTC

[トレース機能の用途] プロパティが [トレース] の場合、データアクセスのトレース結果については、指定 したバス・マスタからのトレース結果のみをトレースパネルに出力します。

- 注意 1. プログラム実行中にこのプロパティを変更することはできません。
- **注意 2.** [トレース機能の用途] プロパティが [リアルタイム RAM モニタ] の場合,パス・マスタは [CPU] 固定となり変更できません。
- 注意 3. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は、[データアクセスのバス・マスタ] 【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】プロパティは表示されません。その場合、バス・マスタは[CPU] 固定となります。

- [外部トレース出力]【E20(JTAG)】

トレース取得データの外部出力方法を指定します。

外部出力方法は、次のドロップダウン・リストより指定します。

CPU 実行優先	トレース出力より CPU 実行を優先します。 出力されるトレース情報にロストが発生する場合があります。	
トレース出力優先	CPU 実行よりトレース出力を優先します。 トレース出力のために CPU 実行が停止するため、リアルタイム性が損なわれま す。	
外部出力しない	トレース情報は出力されません。 マイクロコントローラの内蔵バッファのみを使用します。	

注意 1. [トレース機能の用途] プロパティにおいて, [リアルタイム RAM モニタ]が指定された場合, ドロップダウンリストより [外部出力しない]は指定できません。

注意 2. [CPU 実行優先]または[トレース出力優先]を指定した場合, トレースパネルにトレース・ データが表示された状態でステップ・イン実行すると正しいトレース・データが表示されない 場合があります。

- **注意 3.** 【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】 [外部出力しない]から [CPU 実行優先],または [トレース出力優先] に変更した場合,タイ マ計測結果が初期化されます。
- [トレース・メモリ・サイズ [M バイト]]【E20(JTAG)】 トレース・データを保持するメモリサイズを指定します。 ドロップダウン・リストには、次のメモリサイズが表示されます。
 - 1(デフォルト), 2, 4, 8, 16, 32

注意 [トレース機能の用途] プロパティにおいて,[リアルタイム RAM モニタ]を指定した場合,プ ロパティ値は[1]と表示され変更不可状態となります。

- [タイム・スタンプ出力] 収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。 このプロパティは、[トレース機能の用途]プロパティにおいて、[トレース]を指定した場合のみ選択可能と なります。

マイクロコントローラ データ種別		変更可否	
RX66x, RX65N-2M, RX651-2M, RX671, RX72M, RX72T, RX72N グループ	分岐, 分岐 + データアクセス, データアクセス, 分岐 + データアクセス (アクセス・データなし), データアクセス (アクセス・データなし)	プログラム停止時のみ変更可能	
上記以外の RX600 シ リーズ	分岐, 分岐 + データアクセス, データアクセス	プログラム停止時のみ変更可能	
RX200 シリーズ	分岐	変更不可 [いいえ] が固定表示	
	データアクセス	プログラム停止時のみ変更可能	
RX100 シリーズ	分岐、データアクセス	変更不可 [いいえ]が固定表示	

- [トレース・クロック・カウント・ソース [MHz]]

このプロパティは、[タイム・スタンプ出力]プロパティにおいて[はい]を指定した場合にのみ表示されます。

カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。

直接入力により、0.0001 - 999.999の間で指定してください。 なお、プロパティが空欄の場合、[接続用設定] タブの [クロック] カテゴリ内の [動作周波数 [MHz]] プロ パティの設定値がカウント・ソースとして代用されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. プロパティの設定値は, 収集済みのトレース・データには反映しません。プロパティ設定後に 収集したトレース・データから反映します。

- [トレース・クロック・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】

タイム・スタンプのカウント・ソースの分周比を指定します。

このプロパティは, [タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されます。

ドロップダウン・リストには、次の分周比が表示されます。

- 分周なし(デフォルト), 16 分周, 256 分周, 4096 分周

[トレース・クロック・カウント・ソース [MHz]] プロパティで指定した周波数をさらに指定した値で分周 (周波数を 1/n 倍に変換)した値が実際のタイム・スタンプの単位(1 カウントあたりの周波数)になります。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

- **注意 2.** トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は, [トレース・クロック・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】プロパティは表示されません。その場合, [分周なし] 固定となります。
- (7) [タイマ]【RX600, RX700 シリーズ】このカテゴリでは、タイマ機能に関する設定を行います。

図 2.43 [タイマ]カテゴリ

▲ 9/マ

カウンタを64ビットで使用する いいえ

- [カウンタを 64 ビットで使用する] 計測カウンタを 32 ビット ×2 本で使用するか, 64 ビット ×1 本で使用するかを指定します。
- (8) [カバレッジ] 【RX71M, RX66x, RX65x, RX64M, RX671, RX72M, RX72T, RX72N グループ】
 このカテゴリでは、カバレッジ機能に関する設定を行います。
 カバレッジ機能、およびこのカテゴリ内の設定についての詳細は、「2.15 カバレッジの測定【シミュレータ】
 【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】」を参照してください。



2.3.3.3 [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、デバッグ・ツールにダウンロードを実行する際の設定を行います。 各カテゴリ内の設定についての詳細は、「2.5.1 ダウンロードを実行する」を参照してください。

2.3.3.4 [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。 フック処理、および各カテゴリ内の設定についての詳細は、「2.18 フック処理を設定する」を参照してください。



2.3.4 【EZ Emulator】の場合

EZ Emulator を使用する場合の動作環境の設定を次のプロパティ パネルで行います。

図 2.44 1	動作環境設定	(プロパティ	パネル)
----------	--------	--------	------

- Deleteres as ferral	<u> </u>
* IF 18 FOURT (SOUTH) 内菜 FOURT (SOUTH)	2148
内蔵DAMHとプロロジイト	138
デーカフラッシュ・メチル・サイブ[にパイト]	21
Anah	
14'c/mab-V=1	EXTAL
メイン・クロック IDE#特f Mile1	E. The
新作版》使初[MH2]	
内蔵フラッシュ・メモリまき換え時のりロック操作を許可する	LVCs2
ターゲット・ボードンの視聴	1.716
NCU8hfF@FE[V]	
通信方式	FINE
FINE#-L-b(box)	1000000
1 75901	
10コードの入力モード	IDコードを16道32桁で推定
IDD-F	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ワークRAM開始アドレス	1000
ワークRAMサイズUSTE1	1024
CPUMM/FE~F	
モード導子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ
外部フラッシュ	
外部フラッシュ定義ファイル	[4]

プロパティ パネル上の該当するタブを選択し、次の設定を順次行ってください。

2.3.4.1 [接続用設定] タブ
2.3.4.2 [デバッグ・ツール設定] タブ
2.3.4.3 [ダウンロード・ファイル設定] タブ
2.3.4.4 [フック処理設定] タブ

2.3.4.1 [接続用設定] タブ

[接続用設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールとの接続に関する設定を行います。

- (1) [内蔵 ROM/RAM]
- (2) [クロック]
- (3) [ターゲット・ボードとの接続]
- (4) [フラッシュ]
- (5) [CPU の動作モード]
- (6) [外部フラッシュ]
- [内蔵 ROM/RAM]
 このカテゴリでは、内蔵 ROM/RAM に関する設定を表示します。

図 2.45 [内蔵 ROM/RAM] カテゴリ

4	内蔵ROM/RAM		
	内蔵ROMサイズ[Kパイト]	2048	
	内蔵RAMサイズ[K/テイト]	128	
	データフラッシュ・メモリ・サイズ[K/Ÿイト]	32	



- [内蔵 ROM サイズ [K バイト]] エミュレーションする内蔵 ROM サイズを表示します(単位: K バイト)。 このプロパティ値を変更することはできません。
- [内蔵 RAM サイズ [バイト]] エミュレーションする内蔵 RAM サイズを表示します(単位:バイト)。 このプロパティ値を変更することはできません。
- [データフラッシュ・メモリ・サイズ [K バイト]]
 選択しているマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表示します(単位:Kバイト)。
 選択しているマイクロコントローラがデータフラッシュ・メモリ非搭載品の場合は、[0]を表示します。
 このプロパティ値を変更することはできません。
- (2) [クロック]

このカテゴリでは、クロックに関する設定を行います。

注意 EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.46 [クロック] カテゴリ【HOCO】

4	クロック		
	メイン・クロック・ソース	HOCO	
	動作問波数[MHz]	25.0000	
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ	

図 2.47 [クロック] カテゴリ【EXTAL】

4	<u> / በ </u>	
	メイン・クロック・ソース	EXTAL
	メイン・クロック周波数[MHz]	12.5000
	動作問波数[MHz]	25.0000
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

- [メイン・クロック・ソース] メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。 内蔵 HOCO を持たないマイクロコントローラの場合は, [EXTAL] を表示します。

- [メイン・クロック周波数 [MHz]]
 メイン・クロックの周波数(逓信前)を指定します。
 EXTAL 周波数値として 0.0001 ~ 99.9999(単位:MHz)の範囲で数値を直接入力します。
 入力値の少数第5位以下は切り捨てられ、指定可能範囲外の入力値は、0.0001(0以下だった時)または
 99.9999(100以上だった時)に丸め込みます。
 このプロパティは、[メイン・クロック・ソース]プロパティで [EXTAL]を指定した場合のみ表示されます。
- [動作周波数 [MHz]] 動作周波数 (ICLK)を指定します。 0.0001 ~ 999.999(単位: MHz)の範囲で数値を直接入力します。 指定可能範囲外の入力値は,0.0001(0以下だった時)または999.999(1000以上だった時)に丸め込みます。
- [内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する] 内蔵フラッシュ・メモリ書き換え時に、デバッガによるクロック操作を許可するか否かを指定します。

(3) [ターゲット・ボードとの接続]
 このカテゴリでは、EZ Emulator とターゲット・ボードとの接続に関する設定を行います。

注意
【EZ Emulator】
[はい]を指定した場合、マイクロコントローラの FlashIF クロック(FCLK)が動作保証範囲
外(LOCO やサブクロック動作中など)でデバッガによる内蔵フラッシュ ROM の書き換えを
実施した時、EZ Emulator がクロック・ソースを切り替えます。なお、書き換え完了後は、元
のクロック・ソースに復帰します。
クロック操作を許可した場合、クロック・ソースを切り替えるため、内蔵フラッシュ・メモリ
書き換え中の周辺クロック動作周波数が変更されます。
クロック操作の許可設定は、プログラム実行やステップ実行などの実行系機能を実施した後の
内蔵フラッシュ ROM を書き換える時に有効となります。なお、デバッグ・ツールの接続直後
や CPU リセット発行時は、クロック操作の許可設定に関わらず FCLK が動作保証範囲外の場
合は、強制的にクロック・ソースを切り替えます。

図 2.48 [ターゲット・ボードとの接続] カテゴリ

▲ ターゲット・ボードとの接続

7 771 AFT CA71900A	
MCLI動作電圧[V]	
THE BASE PROVIDENCE AND	
ime方式	FINE
N21877F1	1 21112
FINFボーレート(hoe)	100000
Linch N. Linbal	100000

- [MCU 動作電圧 [V]]

ユーザ基板に実際に供給しているマイクロコントローラの動作電圧値を指定します。 0.0001 ~ 9.9999(単位:V)の範囲の数値を直接入力で指定します。

注意 EZ Emulator と接続中にこのプロパティを変更することはできません。

- [通信方式]

EZ Emulator がターゲット・システム上のマイクロコントローラと通信する際の通信方式を表示します。 このプロパティ値を変更することはできません。 なお、デバッグ・ツールの選択についての詳細は、「2.3.1 使用するデバッグ・ツールを選択する」を参照し てください。

- [FINE ボーレート [bps]] EZ Emulator がターゲット・システム上のマイクロコントローラと通信する際の通信速度 (FINE ボーレート) をドロップダウン・リストより指定します。 ドロップダウン・リストには、次の通信速度が表示されます。

トロップダリン・リストには、火の通信速度が表示されま

- 1000000(デフォルト), 500000, 250000

注意 1. EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

- **注意 2.** ターゲット・システム上での FINE 信号の配線長や配線方法によっては,選択した FINE ボーレートで通信できない場合があります。この FINE ボーレートを下げることによって,通信が出来るようになる場合があります。
- (4) [フラッシュ] このカテゴリでは、フラッシュ書き換えに関する設定を行います。

注意 EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.49 [フラッシュ] カテゴリ

▲ フラゥシュ

-	//// L	
	IDコードの入力モード	IDコードを16進32桁で指定
	IDコード	HEN FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	ワークRAM開始アドレス	HEX 1000
	ワークRAMサイズ[パイト]	1024

- [ID コードの入力モード]

ID コードの入力モードを指定します。

注意 EZ Emulator と接続中にこのプロパティを変更することはできません。

- [ID コード]

ID コードプロテクトを解除するための ID コードを指定します。

プロパティにおいて, [ID コードを 16 進 32 桁で指定] を指定した場合, ID コードは 32 桁の 16 進数で入力し てください。また, [ID コードを 16 文字以内で指定 (ASCII)] を指定した場合, ID コードは 16 文字以内の ASCII 文字列で入力してください。

注意 1. EZ Emulator と接続中にこのプロパティを変更することはできません。

- **注意 2.** ID コードを 32 桁の 16 進数で入力する場合は、32 ビットのデータ単位に並べて入力してくだ さい。 フラッシュプログラスでは ID コードの並び順が CS+ と思たる場合がを以ます。ご使用のフ
 - フラッシュプログラマでは ID コードの並び順が CS+ と異なる場合があります。ご使用のフ ラッシュプログラマの ID コード入力の仕様を確認してください。

注意 3. ID コードを ASCII 文字列で入力する場合, 16 文字に満たない部分は "0" で入力されます。

デバッガが使用するワーク RAM の配置アドレスを指定します。なお、アドレス値は4バイト単位の値を指定 してください。(入力値が4バイト単位の値でなかった場合は、自動的に補正されます。)

^{- [}ワーク RAM 開始アドレス]

指定したワーク RAM の配置アドレスから [ワーク RAM サイズ [バイト]] プロパティの表示サイズ分は, デ バッガのファームウェアが使用します。

- 注意 1. EZ Emulator と接続中にこのプロパティを変更することはできません。
- **注意 2.** メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが, ワーク RAM に配置する領域は,以下の用途では使用できません。
 - DMA または DTC 機能の転送元・転送先
 - DTC ベクタテーブル,および転送情報配置アドレス
 - DMAC, DTC 起動要因の割り込みベクタ
- [ワーク RAM サイズ [バイト]] デバッガが使用するワーク RAM のサイズを表示します。
- (5) [CPU の動作モード]

4

このカテゴリでは、エミュレーションするマイクロコントローラの動作モードに関する設定を行います。

注意 EZ Emulator と接続中にこのカテゴリ内のプロパティを変更することはできません。

図 2.50 [CPU の動作モード] カテゴリ

CPUの動作モード	
モード端子設定	シングルチップモード
レジスタ設定	シングルチップモード
エンディアン	Little-endianデータ

- [モード端子設定] マイクロコントローラのモード端子で設定している動作モードを指定します。 EZ Emulator の場合は、シングルチップモード固定となります。
- [レジスタ設定] レジスタで設定する動作モードを指定します。 EZ Emulator の場合は、シングルチップモード固定となります。
- [エンディアン] プロジェクトのエンディアンを表示します。 プロジェクトからエンディアンの設定情報を取得し、その値を表示します。 デバッグ・ツールと切断中のみ指定することができます。
- (6) [外部フラッシュ] このカテゴリでは、外部フラッシュに関する設定を行います。 EZ Emulator の場合、設定をすることは可能ですが、マイクロコントローラの動作モードがシングルチップモー ド固定となるため、本カテゴリにて設定した内容は反映されません。

2.3.4.2 [デバッグ・ツール設定] タブ

[デバッグ・ツール設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールの基本設定を行います。

- (1) [メモリ]
- (2) [実行中のメモリ・アクセス]
- (3) [レジスタ]
- (4) [ブレーク]
- (5) [システム]
- (6) [トレース]
- (7) [タイマ] 【RX600 シリーズ】



(1) [メモリ]
 このカテゴリでは、メモリに関する設定を行います。

図 2.51 [メモリ] カテゴリ

4	メモリ	
4	メモリ・マッピング	[23]
	▷ [0]	内藏RAM領域
	▷ [1]	予約領域
	▷ [2]	L/Oレジスタ領域
	▷ [3]	予約領域
	▷ [4]	1/0レジスタ領域
	▷ [5]	1/0レジスタ領域
	⊳ [6]	予約領域
	▷ [7]	L/Oレジスタ領域
	⊳ [8]	1/0レジスタ領域
	▷ [9]	I/Oレジスタ領域
	▷ [10]	データフラッシュ領域
	▷ [11]	予約領域
	▷ [12]	その他のメモリ領域
	▷ [13]	予約領域
	▷ [14]	L/Oレジスタ領域
	▷ [15]	予約領域
	▷ [16]	1/0レジスタ領域
	▷ [17]	予約領域
	▷ [18]	その他のメモリ領域
	▷ [19]	予約領域
	▷ [20]	その他のメモリ領域
	▷ [21]	予約領域
	▷ [22]	内藏ROM領域
	メモリ書き込み時にベリファイを行う	はい

- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。 このパネル上でマッピング値を変更することはできません。 I/O 保護領域を追加、または削除する必要がある場合は、プロパティを選択することで設定欄右端に表示される[…] ボタンのクリックによりオープンするメモリ・マッピング ダイアログで行います。 設定方法についての詳細は、メモリ・マッピング ダイアログの項を参照してください。 このプロパティでは、メモリ領域の個数のみを表示します。 [メモリ・マッピング] プロパティを展開すると、次のサブ項目が表示されます。

- [メモリ種別]

当該領域のメモリ種別を表示します。 各メモリ種別は次の領域に該当します。

内蔵 ROM 領域	プログラム ROM ^{注 1} およびデータフラッシュ ^{注 2} が該当します。
内蔵 RAM 領域	内蔵 RAM が該当します。
I/O レジスタ領域	周辺 I/O レジスタが該当します。 エンディアンが異なる領域で分割して表示します。
外部領域 (CS7/CS6//CS0)	外部アドレス空間が該当します。 CS0 - CS7 を区別して表示します。
その他のメモリ領域	FCU-RAM ^{注 3} , FCU ファーム ^{注 3} , ユーザーブートが該当します。
予約領域	上記以外の領域を表示します。
I/O 保護領域	外部領域でデバッガからリードしないアドレス範囲が該当します。 メモリ・マッピング ダイアログで登録します。

注 1. ダウンロード以外の操作(メモリ パネルからの操作やライン・アセンブル)で内蔵フラッシュ ROM 領域を編集した場合,編集内容は次の実行操作時にフラッシュ ROM に書き込まれます。

- 注 2. 消去された状態のデータフラッシュ ROM の内容を参照すると、マイクロコントローラの仕様どおり不定値が表示されます。なお、デバッガからデータフラッシュ ROM の内容を書き換えた場合は、256 バイト単位で書き込みを行います。書き込まれた領域は不定値とはなりません。
- 注 3. FCU-RAM 領域の内容をデバッガから書き換えないでください。また, FCU ファーム領域は デバッガから書き換えできません。
- [開始アドレス] 当該領域の開始アドレスを表示します。
- [終了アドレス] 当該領域の終了アドレスを表示します。
- [アドレス幅 [ビット]] 当該領域のアクセス幅を表示します。 [メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。
- [エンディアン] 外部領域および I/O レジスタ領域のエンディアンを表示します。 [メモリ種別]が外部領域については、デバッグ・ツールと切断中のみ変更可能です。

注意 デバッグ・ツールと接続することにより(「2.4.1 CS+にデバッグ・ツールを接続する」参照), 各メモリ種別ごとの詳細表示を行います。

- [メモリ書き込み時にベリファイを行う] メモリ値の初期化を行う際に、ベリファイを行うか否かをドロップダウン・リストにより指定します。 [はい]を指定した場合、ダウンロードした際、ウォッチ パネルやメモリ パネルで値を変更した際にベリファ イを行います。

(2) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセスに関する設定を行います。 このカテゴリ内の設定は、リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新 機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

図 2.52 [実行中のメモリ・アクセス] カテゴリ

▲ 実行中のメモリ・アクセス

201102027201	
実行を一瞬停止してアクセスする	いいえ
実行中に表示更新を行う	(\$0)
表示更新間隔[ms]	500

- [実行を一瞬停止してアクセスする]
 プログラム実行中に、メモリに対してアクセスを許可するか否かをドロップダウン・リストにより指定します。
 アクセスを許可する場合は [はい]を選択してください (デフォルトでは [いいえ] が指定されます)。

- [実行中に表示更新を行う] プログラム実行中に、ウォッチ パネル/メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。 表示内容の更新を行う場合は [はい]を選択してください (デフォルト)。

注意 プログラム実行中にこのプロパティを変更することはできません。

 - [表示更新間隔 [ms]] このプロパティは、[実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ表示されます。 プログラム実行中に、ウォッチ パネル/メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。
 直接入力により、100 ~ 65500 の範囲の整数(100 ms 未満の端数切り上げ)を指定してください(デフォルトでは [500] が指定されます)。
 なお、[実行中に表示更新を行う] プロパティにおいて指定値を [いいえ] から [はい] へ切り替えた場合、このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(3) [レジスタ]このカテゴリでは、プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.53 [レジスタ] カテゴリ

▲ レジスタ

V2A2		
実行中に取り表示を行う	00	
2011-01030000112	10.01	
PC表示面新期间[me]	500	
LO SCALOC MILLING LINE I	300	

- [実行中に PC 表示を行う]

このプロパティは、プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。 [いいえ]を選択した場合、実行中のステータス・バーには"実行中"と表示されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. 【RX100 シリーズ】

プログラム実行中の PC 表示機能をサポートしていないため、非表示となります。

- [PC 表示更新間隔 [ms]]

このプロパティは, [実行中に PC 表示を行う] プロパティにおいて [はい] を指定した場合にのみ表示されます。

プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。

直接入力により、100 ~ 65500 の整数(100ms 未満は端数切り上げ)を指定してください。(デフォルトでは [500] が指定されます。)

なお、[実行中に PC 表示を行う] プロパティにおいて指定値を [いいえ] から [はい] へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(4) [ブレーク]

このカテゴリでは、ブレーク機能に関する設定を行います。

図 2.54 [ブレーク] カテゴリ

 ブレーク 優先的に使用するブレークポイントの種類
 ハードウエア・ブレーク

- [優先的に使用するブレークポイントの種類]

エディタパネル/逆アセンブルパネルにおいて、マウスのワンクリック操作でブレークポイントを設定する際に使用するブレークポイントの種別を次のドロップダウン・リストより指定します。 優先して使用するブレークポイントを使い切った状態で、ブレークポイントを設定した場合、自動的にもう片方のブレークポイントを設定します。 なお、ブレークポイントについての詳細は、「2.10.2 任意の場所で停止する(ブレークポイント)」を参照してください。

ソフトウエア・ブレーク	ソフトウエア・ブレークポイントを優先的に設定します。
ハードウエア・ブレーク	ハードウエア・ブレークポイントを優先的に設定します(デフォルト)。

(5) [システム]

このカテゴリでは、エミュレーションシステムに関する設定を行います。 実行開始前およびブレーク後の指定ルーチン実行についての詳細は、「2.9.4 指定ルーチンを実行する【E1】 【E20】【EZ Emulator】」を参照してください。

図 2.55 [システム]カテゴリ

å	システム	
	内蔵フログラムROMを書き換えるプログラムをデバッグする	いいえ
	内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
	実行開始前に指定ルーチンを実行する	はい
	実行開始前に実行するルーチン	
	ブレーク後に指定ルーチンを実行する	(tru
	ブレーク後に実行するルーチン	
	指定ルーチン実行機能用ワークRAM開始アドレス	HEX 1FDD0
	指定ルーチン実行機能用ワークRAMサイズ[レテイト]	560



- [内蔵プログラム ROM を書き換えるプログラムをデバッグする] 内蔵プログラム ROM 領域を書き換えるプログラム (ROM P/E モードを使用したプログラムなど) をデバッグ するか否かを指定します。

注意 EZ Emulator と接続中にこのプロパティを変更することはできません。

- [内蔵データフラッシュを書き換えるプログラムをデバッグする] 内蔵データフラッシュ領域を書き換えるプログラム (データフラッシュ P/E モードを使用したプログラムなど)をデバッグするか否かを指定します。

注意 EZ Emulator と接続中にこのプロパティを変更することはできません。

- [実行開始前に指定ルーチンを実行する] ユーザ・プログラム実行前に,指定ルーチンを実行する必要があるか否かを指定します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [実行開始前に実行するルーチン] ユーザ・プログラム実行直前に実行するアドレスを指定します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合 のみ表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に指定ルーチンを実行する]

ユーザ・プログラムブレーク後に、指定ルーチンを実行する必要があるか否かを指定します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [ブレーク後に実行するルーチン] ユーザ・プログラムブレーク直後に実行するアドレスを指定します。 このプロパティは、[ブレーク後に指定ルーチンを実行する]プロパティにおいて、[はい]が指定された場合 のみ表示します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [指定ルーチン実行機能用ワーク RAM 開始アドレス] 指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。なお、アドレス値は4バイト単位の値を 指定してください。(入力値が4バイト単位の値でなかった場合は、自動的に補正されます。) 指定したワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパ ティの表示サイズ分は、デバッガのファームウェアが使用します。 このプロパティは、[実行開始前に指定ルーチンを実行する] プロパティまたは、[ブレーク後に指定ルーチン を実行する] プロパティのいずれかにおいて [はい]が指定された場合のみ表示します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. ユーザ・プログラムで使用しない領域を指定してください。

- [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] 指定ルーチン実行機能用のワーク RAM のサイズを表示します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティまたは、[ブレーク後に指定ルーチン を実行する]プロパティのいずれかにおいて [はい]が指定された場合のみ表示します。
- (6) [トレース]

このカテゴリでは、トレース機能に関する設定を行います。

図 2.56 [トレース]カテゴリ

⊿ トレース

トレース機能の用途	トレース
トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
トレース・データ種別	分岐
タイム・スタンプ出力	an
トレース・クロック・カウント・ソース[MHz]	

- [トレース機能の用途] トレース機能の用途として[トレース]を表示します。 このプロパティ値を変更することはできません。
- [トレース・メモリを使い切った後の動作] トレースの取得モードを指定します。 取得モードは、次のドロップダウン・リストより指定します。

トレース・メモリを上書きし実行を続ける	トレース・メモリがいっぱいになっても, トレース・データ の上書きを続けます。
トレースを停止する	トレース・メモリがいっぱいになったら, トレース・データ の書き込みを中止します。
停止する	トレース・メモリがいっぱいになった時点でブレークします。

- [トレース・データ種別]

トレース取得するデータ種別をドロップダウン・リストより指定します。 指定可能なデータ種別はマイクロコントローラのシリーズにより異なります。 ドロップダウン・リストには、次のデータ種別が表示されます。

-【RX600, RX700 シリーズ】 分岐 , 分岐+データアクセス , データアクセス

-【RX100, RX200 シリーズ】 分岐 , データアクセス

- [タイム・スタンプ出力]

収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。 このプロパティは、マイクロコントローラのシリーズや[トレース・データ種別]プロパティでの指定値に よって変更可否が次のように切り替わります。

マイクロコントローラ	データ種別	変更可否
RX600, RX700 シリーズ	分岐, 分岐 + データアクセス, データアクセス	プログラム停止時のみ変更可能
RX200 シリーズ	分岐	変更不可 [いいえ]が固定表示
	データアクセス	プログラム停止時のみ変更可能
RX100 シリーズ	分岐、データアクセス	変更不可 [いいえ]が固定表示

- [トレース・クロック・カウント・ソース [MHz]]

このプロパティは, [タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されます。

カウント値からタイム・スタンプを算出するためのカウント・ソースを入力します。 直接入力により、0.0001 - 999.999の間で指定してください。 なお、プロパティが空欄の場合、[接続用設定]タブの[クロック]カテゴリ内の[動作周波数 [MHz]]プロ パティの設定値がカウント・ソースとして代用されます。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. プロパティの設定値は, 収集済みのトレース・データには反映しません。プロパティ設定後に 収集したトレース・データから反映します。

(7) [タイマ]【RX600 シリーズ】このカテゴリでは、タイマ機能に関する設定を行います。

図 2.57 [タイマ]カテゴリ

▲ 9/マ

カウンタを64ビットで使用する いいえ

- [カウンタを 64 ビットで使用する] 計測カウンタを 32 ビット ×2 本で使用するか, 64 ビット ×1 本で使用するかを指定します。

2.3.4.3 [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、デバッグ・ツールにダウンロードを実行する際の設定を行います。 各カテゴリ内の設定についての詳細は、「2.5.1 ダウンロードを実行する」を参照してください。

2.3.4.4 [フック処理設定] タブ

[フック処理設定] タブでは, デバッグ・ツールにフック処理の設定を行います。 フック処理, および各カテゴリ内の設定についての詳細は,「2.18 フック処理を設定する」を参照してください。



2.3.5 【シミュレータ】の場合

シミュレータを使用する場合の動作環境の設定を次のプロパティパネルで行います。

図 2.58 動作環境設定【シミュレータ】(プロパティ パネル)

70/777		×		
🔊 RX シミュレータ のプロパティ		P -+		
▲ 内藏RON/RAM				
内蔵R0Mサイズ[K7行下]	2048			
内蔵RAMサイズ[IUS/F]	128			
* エンディアン				
CPUILンディアン	Littlemendianデータ			
▲ クロック				
システムクロック(ICLK)周波数[MH2]	100			
▲ 周辺機能シミュレーション				
▷ 期辺根能シミュレーションモジュール	[0]			
周辺クロックレート	1			
Pi截ROM/RAM				
▲ 「「「「「「」」」」」」」」」「「「」」」」」」」」」」」」」」」」」」」				

プロパティ パネル上の該当するタブを選択し、次の設定を順次行ってください。

2.3.5.1 [接続用設定] タブ
2.3.5.2 [デバッグ・ツール設定] タブ
2.3.5.3 [ダウンロード・ファイル設定] タブ
2.3.5.4 [フック処理設定] タブ

2.3.5.1 [接続用設定] タブ

[接続用設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールとの接続に関する設定を行います。

- (1) [内蔵 ROM/RAM]
- (2) [エンディアン]
- (3) [クロック]
- (4) [周辺機能シミュレーション]
- (1) [内蔵 ROM/RAM] このカテゴリでは,選択しているマイクロコントローラの内蔵 ROM/RAM サイズが表示されます

図 2.59 [内蔵 ROM/RAM] カテゴリ【シミュレータ】

4	内蔵ROM/RAM			
	内蔵 ROMサイズ[K/Ÿイト]	2048		
	内蔵RAMサイズ[Kパイト]	128		

- [内蔵 ROM サイズ [K バイト]] シミュレーションする内蔵 ROM サイズを表示します(単位: K バイト)。 このプロパティ値を変更することはできません。
- [内蔵 RAM サイズ [K バイト]] シミュレーションする内蔵 RAM サイズを表示します(単位: K バイト)。 このプロパティ値を変更することはできません。
- (2) [エンディアン]このカテゴリでは、エンディアンに関する設定を行います。

図 2.60 [エンディアン] カテゴリ【シミュレータ】 **4 エンディアン** CPUエンディアン Little-endianデータ

- [CPU エンディアン] CPU のエンディアンを指定します。 デフォルトは,ビルド・ツールのプロパティで選択されたエンディアンを表示します。
 - 注意 シミュレータと接続中にこのプロパティを変更することはできません。
- (3) [クロック]このカテゴリでは、クロックに関する設定を行います。

図 2.61 [クロック] カテゴリ【シミュレータ】

- クロック システムクロック(ICLK)周波数[MHz] 100
- [システムクロック (ICLK) 周波数 [MHz]] CPU の動作周波数を指定します(単位: MHz)。 使用する周波数に合わせて設定してください。 直接入力により、1~1,000 の範囲の整数を指定してください。

注意 シミュレータと接続中にこのプロパティを変更することはできません。

(4) [周辺機能シミュレーション]このカテゴリでは、周辺機能シミュレーションに関する設定を行います。

図 2.62 [周辺機能シミュレーション] カテゴリ

▲ 周辺機能シミュレーション

山大:(味噌)ノミエレ ノヨノ	
周辺機能シミュレーションモジュール	[3]
⊳ [0]	CMT
▷ [1]	ICU
▷ [2]	MPU
周辺クロックレート	1
	 コン構能シミュレーションモジュール ▷ [0] ▷ [1] ▷ [2] ฉ辺クロックレート

- [周辺機能シミュレーションモジュール] 使用可能な周辺機能シミュレーションモジュール名を表示,および表示された周辺機能シミュレーションモ ジュールを使用するか否かをドロップダウン・リストで指定します。 使用する場合は [はい]を選択してください。(デフォルトでは [いいえ]が指定されます)。 なお,このパネル上で周辺機能シミュレーションモジュール名を変更することはできません。

- [周辺クロックレート] 周辺クロックと内部クロックの比(周辺1クロックが内部クロックいくつに相当するか)をドロップダウン・ リストにより指定します。 ドロップダウン・リストには、次のクロックレートが表示されます。 1(デフォルト), 2, 3, 4, 6, 8, 12, 16, 24, 32, 64

注意 シミュレータと接続中にこのプロパティを変更することはできません。

2.3.5.2 [デバッグ・ツール設定] タブ

[デバッグ・ツール設定] タブでは、次に示すカテゴリごとに、デバッグ・ツールの基本設定を行います。

(1) [メモリ]

- (2) [実行中のメモリ・アクセス]
- (3) [レジスタ]
- (4) [トレース]
- (5) [カバレッジ]
- (6) [ストリーム入出力]
- (7) [実行モード]
- (8) [命令デコードキャッシュ]



(1) [メモリ]
 このカテゴリでは、メモリに関する設定を行います。

図 2.63 [メモリ]カテゴリ【シミュレータ】

⊿ メモリ

メモリ・マッピング	[17]
⊳ [0]	内藏RAM領域
▶ [1]	ノン・マップ領域
▶ [2]	1/0レジスタ領域
⊳ [3]	ノン・マップ領域
▶ [4]	I/Oレジスタ領域
▶ [5]	1/0レジスタ領域
⊳ [6]	ノン・マップ領域
▶ [7]	1/0レジスタ領域
⊳ [8]	L/Oレジスタ領域
⊳ [9]	1/0レジスタ領域
▶ [10]	内藏ROM領域
▷ [11]	ノン・マップ領域
▷ [12]	1/0レジスタ領域
▶ [13]	ノン・マップ領域
▷ [14]	1/0レジスタ領域
▷ [15]	ノン・マップ領域
▷ [16]	内蔵ROM領域

- [メモリ・マッピング]

現在のメモリ・マッピングの状況が、メモリ領域の種別ごとに詳細表示されます。 このパネル上でマッピング値を変更することはできません。メモリ・マッピングを追加する必要がある場合 は、[メモリ・マッピング] プロパティを選択することで設定欄右端に表示される […] ボタンのクリックに よりオープンするメモリ・マッピング ダイアログで行います。 設定方法についての詳細は、メモリ・マッピング ダイアログの項を参照してください。

図 2.64 メモリ・マッピング ダイアログのオープン

4	メモリ		
4	メモリ・マッピング	[17]	
	⊳ [0]	内蔽RAM領域	
	⊳ [1]	ノン・マップ領域	

(2) [実行中のメモリ・アクセス]

このカテゴリでは、プログラム実行中におけるメモリ・アクセスに関する設定を行います。 このカテゴリ内の設定は、リアルタイム表示更新機能を使用する場合に必要となります。リアルタイム表示更新 機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

図 2.65 [実行中のメモリ・アクセス] カテゴリ【シミュレータ】

4	本行	idb/	n-v-a	EII.	マカ	177
-	201	1.473	0.21	C-7 -	ry	EA.

XITTOACY 79CA	
実行中に表示更新を行う	はい
表示更新間隔[ms]	500

- [実行中に表示更新を行う]

プログラム実行中に, ウォッチ パネル/メモリ パネルの表示内容を更新するか否かをドロップダウン・リストにより指定します。

表示内容の更新を行う場合は [はい]を選択してください (デフォルト)。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [表示更新間隔 [ms]]

このプロパティは, [実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ表示します。

プログラム実行中に, ウォッチ パネル/メモリ パネルの表示内容を更新する間隔を 100 ms 単位で指定します。

直接入力により、100 ~ 65500 の整数(100 ms 未満の端数切り上げ)を指定してください(デフォルトでは [500] が指定されます)。 なお、[実行中に表示更新を行う]プロパティにおいて指定値を[いいえ]から[はい]へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(3) [レジスタ] このカテゴリでは、プログラム実行中におけるステータス・バーへの PC 表示に関する設定を行います。

図 2.66 [レジスタ]カテゴリ【シミュレータ】

đ	レジスタ	
	実行中にPC表示を行う	はい
	PC表示更新間隔[ms]	500

- [実行中に PC 表示を行う]

このプロパティは、プログラム実行中にステータス・バーへ PC 値を表示するかどうかを指定します。 [いいえ]を選択した場合、実行中のステータス・バーは"実行中"と表示します。

注意 プログラム実行中にこのプロパティを変更することはできません。

- [PC 表示更新間隔 [ms]]]

このプロパティは, [実行中に表示更新を行う] プロパティにおいて [はい] を指定した場合にのみ表示します。

プログラム実行中にステータス・バーの PC 表示を更新する間隔を 100ms 単位で指定します。

直接入力により、100 ~ 65500 の整数(100ms 未満は端数切り上げ)を指定してください。(デフォルトでは [500] が指定されます。)

なお、[実行中に表示更新を行う] プロパティにおいて指定値を[いいえ]から[はい]へ切り替えた場合、 このプロパティには前回の設定値が表示されます。

注意 プログラム実行中にこのプロパティを変更することはできません。

(4) [トレース]

このカテゴリでは,トレース機能に関する設定を行います。 トレース機能,およびこのカテゴリ内の設定についての詳細は,「2.13 実行履歴の収集」を参照してください。

- (5) [カバレッジ] このカテゴリでは、カバレッジ機能に関する設定を行います。 カバレッジ機能、およびこのカテゴリ内の設定についての詳細は、「2.15 カバレッジの測定【シミュレータ】 【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】」を参照してください。
- (6) [ストリーム入出力] このカテゴリでは、ユーザプログラムからの標準入出力またはファイル入出力を行うためのストリーム入出力に 関する設定を行います。

図 2.67 [ストリーム入出力] カテゴリ

▲ ストリーム入出力

ストリーム入出力モードを選択	シミュレータモード
ストリーム入出力機能を使用する	いいえ
ストリーム入出力用アドレス	HEX 0

- [ストリーム入出力モードを選択]

このプロパティでは、標準入出力を実行するための入出力方式をドロップダウン・リストにより指定します。 エミュレータと同じ低水準インタフェース・ルーチン(アセンブリ言語部分)用のファイルを使用する場合 は、[エミュレータモード]を指定してください(デフォルトでは[シミュレータモード]が指定されていま す)。

備考 ファイル入出力は [シミュレータモード] を指定した場合のみ使用可能です。[エミュレータ モード] を指定した場合は,ファイル入出力は使用できません。

注意 シミュレータと接続中にこのプロパティを変更することはできません。

- [ストリーム入出力機能を使用する] このプロパティは、[ストリーム入出力モードを選択]プロパティにおいて[シミュレータモード]を指定し た場合にのみ表示します。 シミュレータモード使用時のストリーム入出力機能を使用するかどうかをドロップダウン・リストにより指定 します。 ストリーム入出力機能を使用する場合は [はい]を選択してください(デフォルトでは [いいえ] が指定され ます)。

- [ストリーム入出力用アドレス] このプロパティは、[ストリーム入出力モードを選択]プロパティにおいて[シミュレータモード]を指定し た場合にのみ表示します。 シミュレータモード使用時のストリーム入出力の開始位置を指定します。 直接入力によりストリーム入出力用アドレスを指定してください。(デフォルトでは [0x00000000] が指定さ れます。)
- (7) [実行モード]
 - このカテゴリでは、シミュレーションエラー、または例外が発生した場合の動作に関する設定を行います。

図 2.68 [実行モード] カテゴリ

4	実行モード	
	実行モードを選択	停止
	未定義命令例外発生時に停止する	はい
	特権命令例外発生時に停止する	はい
	アクセス例外発生時に停止する	はい
	浮動小数点例外発生時に停止する	はい
	割り込み発生時に停止する	はい
	INT命令実行時に停止する	はい
	BRK命令実行時に停止する	はい

- [実行モードを選択] このプロパティでは、実行モードを指定します。 次のドロップダウン・リストより選択します。

停止	シミュレーションを停止します(デフォルト)。
続行	シミュレーションを続行します(下段のプロパティ項目が有効となります)。

- [未定義命令例外発生時に停止する]

このプロパティは、[実行モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 未定義命令例外発生時に停止するかどうかをドロップダウン・リストにより指定します。 未定義命令例外発生時に停止する場合は、[はい]を選択してください(デフォルト)。

- [特権命令例外発生時に停止する]

このプロパティは、[実行モードを選択] プロパティにおいて [停止] を指定した場合のみ表示されます。 特権命令例外発生時に停止するかどうかをドロップダウン・リストにより指定します。 特権命令例外発生時に停止する場合は、[はい] を選択してください (デフォルト)。

- [アクセス例外発生時に停止する] このプロパティは、[周辺機能シミュレーションモジュール]プロパティに MPU モジュールが存在し、[実行 モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 アクセス例外発生時に停止するかどうかをドロップダウン・リストにより指定します。 アクセス例外発生時に停止する場合は、[はい]を選択してください(デフォルト)。
- [浮動小数点例外発生時に停止する]【FPU 搭載品】 このプロパティは、[実行モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 単精度浮動小数点例外発生時に停止するかどうかをドロップダウン・リストにより指定します。 単精度浮動小数点例外発生時に停止する場合は、[はい]を選択してください(デフォルト)。
- [割り込み発生時に停止する] このプロパティは、[実行モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 割り込み発生時に停止するかどうかをドロップダウン・リストにより指定します。 割り込み発生時に停止する場合は、[はい]を選択してください(デフォルト)。
- [INT 命令実行時に停止する] このプロパティは、[実行モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 INT 命令実行時に停止するかどうかをドロップダウン・リストにより指定します。 INT 命令実行時に停止する場合は、[はい]を選択してください(デフォルト)。
- [BRK 命令実行時に停止する] このプロパティは、[実行モードを選択]プロパティにおいて[停止]を指定した場合のみ表示されます。 BRK 命令実行時に停止するかどうかをドロップダウン・リストにより指定します。

BRK 命令実行時に停止する場合は、[はい]を選択してください(デフォルト)。

(8) [命令デコードキャッシュ] このカテゴリでは、命令実行時にデコード結果を保持し、同一アドレス実行時にデコード結果を利用する機能 (命令デコードキャッシュ)に関する設定を行います。

図 2.69 [命令デコードキャッシュ] カテゴリ

▲ 命令デコードキャッシュ 命令デコード結果をキャッシュしてシミュレーション速度を向上する いいえ

- [命令デコード結果をキャッシュしてシミュレーション速度を向上する] 命令デコードキャッシュ機能を有効にするかどうかをドロップダウン・リストにより指定します。 命令デコードキャッシュ機能を有効にする場合は [はい]を選択してください (デフォルトでは [いいえ] が 指定されます)。

注意 命令デコードキャッシュ機能はデコード結果を再利用するため、自己書き換えコードを使用するプログラムには使用できません。また、プログラムの意図しない動作により命令が書き換わると、正しくエラーを検出できない場合があります。

2.3.5.3 [ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、デバッグ・ツールにダウンロードを実行する際の設定を行います。 各カテゴリ内の設定についての詳細は、「2.5.1 ダウンロードを実行する」を参照してください。

2.3.5.4 [フック処理設定] タブ

[フック処理設定] タブでは、デバッグ・ツールにフック処理の設定を行います。 フック処理、および各カテゴリ内の設定についての詳細は、「2.18 フック処理を設定する」を参照してください。



2.4 デバッグ・ツールとの接続/切断

この節では、CS+とデバッグ・ツールとの接続方法、切断方法、およびホットプラグイン機能を用いた接続方法について説明します。

2.4.1 CS+ にデバッグ・ツールを接続する

[デバッグ]メニュー→[デバッグ・ツールへ接続]を選択することにより、現在のアクティブ・プロジェクトで選択 しているデバッグ・ツールとの通信を開始します。

デバッグ・ツールとの接続に成功すると、メイン・ウインドウのステータス・バーが、次のように変化します。 なお、ステータス・バーに表示される各項目についての詳細は、メイン・ウインドウを参照してください。

図 2.70 デバッグ・ツールとの接続に成功したステータス・バー ▲非接続 ■ BREAK ② 0xffff8000 ■ RX E1(JTAG) ③ 未計測 译 ②

- 備考 デバッグ・ツールバーの ポタンをクリックすることで、デバッグ・ツールと接続したのち、指定 ファイルのダウンロードを実行します(「2.5.1 ダウンロードを実行する」参照)。 また、同ツールバーの ボタンをクリックすることで、プロジェクトのビルドを行い、デバッグ・ ツールと接続したのち、指定ファイルのダウンロードを実行します。
- (1) バージョン情報の表示【E1】【E20】【EZ Emulator】
 デバッグ・ツールとの接続に成功すると、エミュレータ・ファームウェアなどのバージョン情報および供給電圧などのエミュレータ情報を出力パネルに表示します。
 - 図 2.71 接続時の出力 パネル【E1】

出力		×
バージョン情報:」 FFWE20RX600.d11 BfwE20rx600.s」 LEVEL0 LEVEL1 Communi.d11	* . * * . * * . * * * ,] * . * * . * * . * * * ,] * . * * . * * . * * * ,] * . * * . * * . * * * ,]	*
FPGA Communication FPGA エミュレータ情報:J Enulator Board Revision User Vcc [EOF]	**,1 **,1 E1 Rev.**,1 3.33 V,1	E
すべてのメッセージ デバック	・ツール /	* *



このエリアに、デバッグ・ツールの情報が表示されます。



図 2.72 接続時の出力 パネル【E2 Lite】

図 2.73 接続時の出力 パネル【E2】

出力		×
バージョン情報:」 FFME2RX.dll Communi.dll FPGA Communication FPGA エミュレータ情報:」 Enulator Board Revision	*.***.**.***. *.***.***.****. ***.] ***.]	^
User Vcc [EOF]	*.** Y.J	~
<u>すべてのメッセージ</u> *デバッグ・ツール		

図 2.74 接続時の出力 パネル【E20】

出力		×
バージョン情報:」 FFME20FX600.dll BfwE20rx600.s」 LEVEL0 LEVEL1 Communication FPGA Trace Control FPGA エミュレーク情報:」 Enulator Board Revision User Vcc USB Bus Pover User I/F Cable [E0F]	*.**.**.***, *.**.**.***, *.**.**. *.**.**. *.**.**. **, **,	E
		*
すべてのメッセージ デバック	トツール /	-



出力		×
バージョン情報:」 FFMEZCUBERX.dll Firmware BFMOLLEZCUBERM.dll エミュレータ情報:」 Enulator Board Revision [EOF]	*.**.**.***. *.**.**.***. *.**.**.***.J EZ Rev.**.J	×
すべてのメッセージ *デバック	・ツール /	-

注意

[E1] [E20]

エミュレータ・ファームウェアの更新が必要な場合、自動的に更新を行います。そのため、接続 が完了するまでは USB および電源を切断しないでください。

2.4.2 CS+からデバッグ・ツールを切断する

デバッグ・ツールバーの 満 ボタンをクリックすることにより、現在接続しているデバッグ・ツールとの通信を切断します。

- デバッグ・ツールと切断すると、メイン・ウインドウのステータス・バーが、次のように変化します。
- 図 2.76 デバッグ・ツールから切断したステータス・バー ■ BREAK ○ Oxffff8000 ● RX E1(JTAG) ③ 未計測 評 [2] ↓ ↓ このエリアが非表示となります。 "非接続"が表示されます。
- **注意1.** プログラム実行中は、デバッグ・ツールと切断することはできません。切断する場合は、実行中のプロ グラムを停止してから切断してください。

注意 2. 【E1】【E20】【EZ Emulator】
 E1/E20/EZ Emulator の切断は、上記の方法もしくは下記 (1), (2) のいずれかの方法で必ず切断してください。
 (1) [デバッグ]メニューにて [デバッグ・ツールから切断]を選択します。
 (2) Python コンソール パネルにて "debugger.Disconnect()" と入力します。
 強制的にデバッガを終了した場合、次回起動時に正しく起動できません。この状態になった場合は、
 E20 エミュレータは電源スイッチを切り替え、E1/EZ エミュレータは USB インタフェースケーブルを 抜き差しして電源を入れ直してください。なお、EZ エミュレータの場合は、CS+ も再起動してください。

備考 デバッグ・ツールと切断すると、デバッグ・ツールと接続時のみ表示可能なパネル/ダイアログはすべ てクローズします。

2.4.3 CS+ にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】 【E20(JTAG)】

ホットプラグイン接続とは、プログラム実行中にデバッグ・ツールを接続して実行中のプログラムをデバッグすることができる接続方式です。

ホットプラグインの接続手順は、次のとおりです。

- (1) プログラムの実行 エミュレータと接続しない状態で、予めターゲット・システム上のマイクロコントローラにダウンロードしたプ ログラムを実行します。
- (2) デバッグ・ツールの指定
 アクティブ・プロジェクトでホットプラグイン接続可能なデバッグ・ツール(【E1(JTAG)】/【E20(JTAG)】)を
 指定します。

備考 【E1(Serial】【E20(Serial)】【EZ Emulator】 ホットプラグイン接続をサポートしていません。

- (3) CS+にデバッグ・ツールをホットプラグイン接続
 [デバッグ]メニュー→ [ホット・プラグイン]を選択することにより、ホットプラグイン接続の準備を開始します。
- (4) ターゲット・システムの接続 ホットプラグイン接続の準備が完了すると、次のメッセージが表示されます。ターゲット・システムとエミュ レータを接続してから [OK] ボタンをクリックしてください。現在のアクティブ・プロジェクトで選択している デバッグ・ツールとの通信を開始します。
 - 図 2.77 ホットプラグイン接続準備完了メッセージ

+001)
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。
OK キャンセル ヘルプ(日)

注意 【E1(JTAG)】

ホットプラグイン接続する場合,エミュレータとターゲット・システム間に別売の"ホットプラ グアダプタ"(R0E000010ACB00)を接続する必要があります。

(5) ホットプラグイン接続完了

デバッグ・ツールとの接続に成功すると、メイン・ウインドウのステータス・バーが、次のように変化します。 なお、ステータス・バーに表示される各項目についての詳細は、メイン・ウインドウを参照してください。



"RUN" が表示され、プログラム実行中を示します。

- **注意 1.** ホットプラグイン接続完了後、プログラムをブレークするまではトレース機能を使用できません。
- **注意 2.** ホットプラグイン接続完了後,プログラムをブレークするまではリアルタイム RAM モニタ機能を使用 できません。そのため,[トレース]カテゴリの[トレース機能の用途]プロパティにおいて,リアル タイム RAM モニタを指定しないでください。
- **注意 3.** ホットプラグイン接続をした場合、プロジェクトにユーザ情報として設定されていたイベントは全て削除されます(ビルトイン・イベントは除く)。

RENESAS

- **注意 5.** ホットプラグイン接続では、ID コードを確認する為、プログラムを一時的に約 800µs 停止させています。(CPU クロック:100MHz, JTAG クロック:16.5MHz)
- 注意 6. 【RX630, RX631, RX63N, RX63T, RX634 グループ】 マイクロコントローラに書かれたエンディアン選択レジスタ(MDEB, MDES)のエンディアン値とプロジェクトのエンディアン情報を合わせてください。エンディアン選択レジスタの詳細については、マイクロコントローラのハードウエア・マニュアルを参照してください。 また、フラッシュ書き換えプログラムが動作中にホットプラグはできません。



2.5 ダウンロード/アップロード

この節では、デバッグ対象となるプログラム(ロード・モジュール・ファイルなど)を CS+ ヘダウンロードする方法と、デバッグ中のメモリ内容を CS+ からファイルヘアップロードする方法を説明します。

2.5.1 ダウンロードを実行する

デバッグ対象となるロード・モジュール・ファイルのダウンロードを実行します。 次に示す手順に従って、プロパティパネルの[ダウンロード・ファイル設定] タブにおけるダウンロードのための設 定を行ったのち、ダウンロードを実行してください。

(1) [ダウンロード] カテゴリの設定

図 2.79 [ダウンロード] カテゴリ【E1(JTAG)】【E1(Serial)】【E20(JTAG)】【E20(Serial)】【EZ Emulator】

▲ ダウンロード

_	2720 1	
a	ダウンロードするファイル	[1]
	a [0]	DefaultBuild¥Tutorial.abs
	ファイル	DefaultBuild¥Tutorial.abs
	ファイルの種類	ロード・モジュール・ファイル
	オブジェクトをダウンロードする	はい
	シンボルをダウンロードする	はい
	PIC/PIDオフセットを指定する	いいえ
	入力補完機能用の情報を生成する	はい
	ダウンロード後にCPUをリセットする	はい
	ダウンロード前にフラッシュ ROMを消去する	はい
	ダウンロード前にデ ータフラッシュ ROMを消去 する	いいえ
	イベント設定位置の自動変更方法	イベントを保留にする

図 2.80 [ダウンロード] カテゴリ【シミュレータ】

▲ ダウンロード

ダウンロードするファイル	[1]
⊿ [0]	DefaultBuild¥Tutorial.abs
ファイル	DefaultBuild¥Tutorial.abs
ファイルの種類	ロード・モジュール・ファイル
オブジェクトをダウンロードする	(\$6)
シンボルをダウンロードする	はい
PIC/PIDオフセットを指定する	いいえ
入力補完機能用の情報を生成する	(\$0)
ダウンロード後にCPUをリセットする	はい
イベント設定位置の自動変更方法	イベントを保留にする

(a) [ダウンロードするファイル]

ダウンロードの対象となるファイル名,およびダウンロード条件を表示します(プロパティ値の"[]"内の数値は,現在ダウンロードの対象に指定されているファイル数を示す)。 ダウンロードの対象となるファイルは,メイン・プロジェクト/サブプロジェクトでビルド対象に指定してい

タウンロードの対象となるファイルは、メイン・シロシェクド/サウンロシェクドでモルド対象に指定してい るファイルが自動的に決定されます^注。 ただし、ゲウンロードの対象となるファイル、たとびゲウンロード条件は、毛動で恋恵することができます。

ただし、ダウンロードの対象となるファイル、およびダウンロード条件は、手動で変更することができます。 この場合は、「2.5.2 応用的なダウンロード方法」を参照してください。

- 注 外部ビルド・ツール(CS+が提供するビルド・ツール以外のコンパイラ/アセンブラなど)に より作成されたロード・モジュール・ファイルをダウンロードする場合,デバッグ専用プロ ジェクトを作成する必要があります。 デバッグ専用プロジェクトをデバッグの対象とする場合では、ユーザが、プロジェクト・ツ リー上のダウンロード・ファイル・ノードにダウンロードするファイルを追加することで、ダ ウンロードの対象となるファイルがこのプロパティに反映されます。 なお、"外部ビルド・ツールの使用"、および"デバッグ専用プロジェクト"についての詳細は、 「CS+ 統合開発環境 ユーザーズマニュアル プロジェクト操作編」を参照してください。
- (b) [ダウンロード後に CPU をリセットする] ダウンロード完了後に CPU をリセットするか否かをドロップダウン・リストにより指定します。 CPU をリセットする場合は [はい] を選択してください (デフォルト)。

RENESAS

- [ダウンロード前にフラッシュ ROM を消去する]【E1】【E20】【EZ Emulator】 (c) ダウンロード実行前にフラッシュ ROM(プログラム ROM)を消去するか否かをドロップダウン・リストに より指定します。 フラッシュ ROM を消去する場合は [はい]を選択してください。
- (d) 「ダウンロード前にデータフラッシュ ROM を消去する] 【E1】 【E20】 【EZ Emulator】 ダウンロード実行前にデータフラッシュ ROM を消去するか否かをドロップダウン・リストにより指定しま す。 データフラッシュ ROM を消去する場合は「はい〕を選択してください(デフォルトでは「いいえ〕が指定さ れます)。
- (e) [イベント設定位置の自動変更方法]

デバッグ作業を進めることにより、変更を加えたプログラムを再度ダウンロードした場合、現在設定されてい るイベントの設定位置(アドレス)が命令の途中になる場合があります。この場合の対象イベントの扱いをこ のプロパティで指定します。

次のドロップダウン・リストによりどちらかを選択してください。

命令の先頭に移動する	命令の先頭アドレスに対象イベントを再設定します。
イベントを保留にする	対象イベントを保留状態にします(デフォルト)。

ただし、このプロパティでの指定は、デバッグ情報のないイベント設定位置に対してのみ適用されます。デ バッグ情報があるイベント設定位置の場合は、常にソース・テキスト行の先頭に移動します。

[デバッグ情報] カテゴリの設定 (2)

> 図 2.81 [デバッグ情報] カテゴリ

▲ デバッグ情報

CPUリセット後に指定シンボル位置まで実行する	ແບ
指定シンボル	_main
デバッグするオーバーレイ・セクションを設定する	いいえ
メモリ使用量の上限サイズDMバイト]	500

(a) [CPU リセット後に指定シンボル位置まで実行する] CPU リセット後, またはダウンロード完了後([ダウンロード後に CPU をリセットする] プロパティで [は い]を指定している場合のみ)に、プログラムを指定シンボル位置まで実行するか否かをドロップダウン・リ ストにより指定します。

プログラムを指定シンボル位置まで実行する場合は「はい」を選択してください(デフォルト)。

[ダウンロード後に CPU をリセットする] プロパティで [はい] を指定している場合では、こ 備考 のプロパティで[はい]を指定すると、ダウンロード完了後、[指定シンボル]プロパティで指 定した位置のソース・テキストを表示した状態でエディタ パネルが自動的にオープンします。 また, [いいえ]を指定した場合では、リセット番地を表示した状態で同パネルがオープンしま す(リセット番地にソース・テキストが割り付けられていない場合は、逆アセンブル パネルで 該当箇所を表示します)。

(b) [指定シンボル] このプロパティは、[CPU リセット後に指定シンボル位置まで実行する]プロパティにおいて[はい]を指定 した場合のみ表示されます。 CPU リセット後にプログラムを実行して停止する位置を指定します。 0~"アドレス空間の終了アドレス"の範囲のアドレス式を直接入力で指定してください(デフォルトでは [main] が指定されます)。 ただし、指定したアドレス式がアドレスに変換できない場合、プログラムは実行されません。

- 備考 通常、次を指定します。 アセンブラ・ソースの場合:メイン関数に相当する先頭ラベル C ソースの場合: メイン関数名の先頭に付与したシンボル
- (c) [デバッグするオーバーレイ・セクションを設定する] ロード・モジュール内にオーバーレイ・セクション(「2.8 オーバーレイ・セクションの設定」参照)が存在 する場合、[はい]となります。 また、ロード・モジュール内にオーバーレイ・セクションが存在しない場合は、[いいえ]となります(変更 不可)。
- (d) [オーバーレイ・セクション] オーバーレイ・セクションが存在するアドレス・グループを表示します。

なお, このプロパティは, [デバッグするオーバーレイ・セクションを設定する] プロパティにおいて[はい] が設定されている場合のみ表示されます。

(e) [メモリ使用量の上限サイズ [M バイト]] デバッグ情報の読み込みで使用するメモリ量の上限サイズ [M バイト]を指定します。 直接入力により、100 ~ 1000 の範囲の整数を指定してください (デフォルトでは [500] が指定されます)。 メモリ不足が発生する場合、上限サイズを小さくすることで改善される可能性があります。ただし、上限サイ ズを小さくするとデバッグ・ツールの応答性が低下する場合があります。

- **注意** デフォルトの設定では、ダウンロード後に自動的に CPU をリセットし、指定シンボルまで実行 します。この動作が不要な場合は、[ダウンロード後に CPU をリセットする] プロパティ、およ び [CPU リセット後に指定シンボル位置まで実行する] プロパティにおいて [いいえ] を指定し てください。
- (3) ダウンロードの実行 デバッグ・ツールバーの ボタンをクリックします。 なお、デバッグ・ツールと切断時にこの操作が行われた場合は、自動的にデバッグ・ツールと接続したのち、ダウンロードを実行します。
 - 備考 デバッグ作業を進めることにより、変更を加えたプログラムを再度ダウンロードする場合は、メ イン・ウインドウ上の[デバッグ]メニュー→[ビルド&デバッグ・ツールへダウンロード]を 選択することにより、ビルド→ダウンロードを容易に行うことができます。
- (4) ダウンロードをキャンセルする ダウンロードの実行をキャンセルする場合、ダウンロードの進捗状況を表示する処理中表示 ダイアログの [キャンセル]ボタンを押下か、または [Esc] キーを押下してください。

ロード・モジュール・ファイルのダウンロードが成功すると、エディタパネルが自動的にオープンし、ダウンロード したファイルのソース・テキストが表示されます。

- 備考 ダウンロードの実行前/実行後に、I/O レジスタ /CPU レジスタ値を指定した値に自動的に書き換える 処理を設定することができます(「2.18 フック処理を設定する」参照)。
- 2.5.2 応用的なダウンロード方法

ダウンロードの対象となるファイル、およびその際のダウンロード条件は変更することができます。 CS+では、次のファイル形式をダウンロードすることができます。

表 2.2 ダウンロード可能なファイル形式

ファイル形式	拡張子	備考
ロード・モジュール・フォーマット	abs	
インテル・ヘキサ・フォーマット - 標準(オフセットアドレス) - 02(セグメントアドレスレコード) - 04(拡張リニアアドレスコード)	hex	03(スタートセグメントアドレスレ コード), 05(スタートリニアアドレ スレコード)は無視します。
モトローラ S フォーマット - (S0, S1, S9-16 ビット) - (S0, S2, S8-24 ビット) - (S0, S3, S7-32 ビット)	mot	
バイナリ・データ・フォーマット	bin	

ダウンロード・ファイルの変更,およびその際のダウンロード条件の設定は,次のダウンロード・ファイル ダイアロ グにより行います。

ダウンロード・ファイル ダイアログは、プロパティ パネルの [ダウンロード・ファイル設定] タブ上の [ダウンロード] カテゴリ内 [ダウンロードするファイル] プロパティを選択することで欄内右端に表示される […] ボタンをクリックするとオープンします。

RENESAS

図 2.82 ダウンロード・ファイル ダイアログのオープン

4	ダウンロード		\frown
Þ	ダウンロードするファイル	[1]	
	ダウンロード後にCPUをリセットする	(#0)	
	ダウンロード前にフラッシュ ROMを消去する	はい	
	ダウンロード前にデータフラッシュ ROMを消去する	いいえ	
	イベント設定位置の自動変更方法	イベントを保留にする	

図 2.83 応用的なダウンロード方法(ダウンロード・ファイル ダイアログ)

ダウンロード・ファ	イル		
ダウンロード・ファイルー	-覧(E):	ダウンロード・ファイルのプロパティ(P):	
Tutorial.abs	() () () () () () () () () () () () () (▲ ダウンロード・ファイル情報 ファイルの種類 オブジェクトをダウンロードする シンボルをダウンロードする PIC/PIDオフセットを指定する 入力補完線能用の情報を生成する	DefaultBuild#Tutorialabs ロード・モジュール・ファイル はい いいえ はい
Alta (A)	\$(E)	ファイル ダウンロードするファイルを指定します。 のK	キャンセル ヘルプ(H)

[ダウンロード・ファイル一覧] エリア [ダウンロード・ファイルのプロパティ] エリア

ここでは、上記ダウンロード・ファイル ダイアログにおける、次の場合の設定方法を説明します。

2.5.2.1 ロード・モジュール・ファイルのダウンロード条件を変更する
2.5.2.2 ダウンロード・ファイル (*.hex/*.mot/*.bin)を追加する
2.5.2.3 複数のロード・モジュール・ファイルをダウンロードする
2.5.2.4 ヘキサ・フォーマット/モトローラSフォーマット/バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う
2.5.2.5 外部フラッシュ・メモリへダウンロードする【E1】【E20】

2.5.2.1 ロード・モジュール・ファイルのダウンロード条件を変更する

ダウンロードするロード・モジュール・ファイルのダウンロード条件(オブジェクト情報/シンボル情報の読み込み など)を変更する場合は、ダウンロード・ファイル ダイアログにおいて、次の手順の操作を行ってください。

- ロード・モジュール・ファイルの選択
 「ダウンロード・ファイル一覧」エリアにおいて、ダウンロードするロード・モジュール・ファイルを選択します。
- (2) ダウンロード条件の変更
 「ダウンロード・ファイル一覧]エリアでは、選択しているロード・モジュール・ファイルの現在のダウンロード条件が表示されます。
 表示される各項目において、設定の変更を行います。

オブジェクトをダ	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。			
ウンロードする	デフォルト	はい		
	変更方法	ドロップダ	ウン・リストによる選択	
	指定可能值	はい	オブジェクト情報をダウンロードします。	
		いいえ	オブジェクト情報をダウンロードしません。	
シンボルをダウン	指定したファイルからシンボル情報をダウンロードするか否かを指定します ^{注1} 。			
	デフォルト	はい		
	変更方法	ドロップダウン・リストによる選択		
	指定可能值	はい	シンボル情報をダウンロードします。	
		いいえ	シンボル情報をダウンロードしません。	
PIC/PID オフセッ トを指定する	ダウンロード び PID(Posi するか否かを セット ", "PI	ドするロード・モジュールの PIC(Position Independent Code)領域,およ osition Independent Data)領域の位置をロード・モジュール作成時から変更 を指定します。この項目を "はい "に切り替えると,サブ項目に "PIC オフ PID オフセット "項目を表示します。		
	デフォルト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能值	はい	PIC/PID オフセットを指定します ^{注 2} 。	
		いいえ	PIC/PID オフセットを指定しません。	
PIC オフセット	ロード・モジ 例えば, プロ 目に "1000" を デフォルト	ド・モジュール作成時のアドレスからオフセット値を入力します。 ず、プログラム・セクションの開始アドレスが 0x1000 番地からだった場合に本項 "1000" を入力すると、対象のセクションは 0x2000 番地にダウンロードされます。 		
	変更方法	キーボードからの直接入力		
	指定可能值	0x0~0xFFFFFFFFの16准数		
PID オフセット	ロード・モジ す。 例えば,ロー 目に "200" を			
	デフォルト	0		
	変更方法	キーボードからの直接入力		
	指定可能值	0x0 ~ 0xFFFFFFF の 16 進数		
入力補完機能用の 情報を生成する	この項目は, み表示されたフ 機能ったた報 ウォッ択してく ウンロードに	 、ダウンロードするファイルがロード・モジュール・フォーマットの場合の ます。 :ファイルからシンボル情報をダウンロードする際に、シンボル名の入力補完 う報を生成するか否かを指定します。 パネルやメモリパネルなどでシンボル名の入力補完機能を使う場合は"はい" ごください。なお、この情報はシンボルのダウンロード中に生成するため、ダ iに時間が掛かる場合があります。 		
デフォルト はい				
	変更方法	ドロップダウン・リストによる選択		
	指定可能值	はい	入力補完機能用の情報を生成します。	
		いいえ	入力補完機能用の情報を生成しません。	


- 注 1. シンボル情報をダウンロードしない場合、ソース・レベル・デバッグを行うことはできません。
- 注 2. PIC/PID 機能(「2.7 PIC/PID 機能の使用」参照)を使って作成していないロード・モジュールに 対して"はい"を選択した時のデバッグ動作は保証できません。
- (3) [OK] ボタンのクリック このダイアログでの設定を有効とし、ダウンロード条件が変更されます。

2.5.2.2 ダウンロード・ファイル(*.hex/*.mot/*.bin)を追加する

ダウンロードするファイル (ヘキサ・フォーマット (*.hex) /モトローラ S フォーマット (*.mot) /バイナリ・ データ・フォーマット (*.bin))を追加する場合は、ダウンロード・ファイル ダイアログにおいて、次の手順の操作を 行ってください。

- (1) [追加] ボタンのクリック
 [追加] ボタンをクリックすると、[ダウンロード・ファイル一覧] エリアの最終行に空欄の項目("-")が表示されます。
- (2) 追加するダウンロード・ファイルのプロパティ設定
 [ダウンロード・ファイルのプロパティ]エリアにおいて、追加するダウンロード・ファイルの選択とダウンロード条件を設定します。
 表示される各項目において、次の設定を行ってください。
 設定を完了すると、[ダウンロード・ファイル一覧]エリア内の空欄の項目に、ここで指定したファイル名が反

設定を完了すると、「タワンロート・ファイル一覧」エリア内の空禰の項目に、ここで指定したファイル名が反 映されます。

ファイル	追加するダウンロード・ファイル(ヘキサ・フォーマット(*.hex)/モトローラ S フォーマット(*.mot)/バイナリ・データ・フォーマット(*.bin))を指定します(最 大指定文字数:259 文字)。					
	デフォルト	空欄				
	変更方法	キーボードからの直接入力,またはこのプロパティを選択すると右端に 表示される […] ボタンのクリックによりオープンするダウンロードする ファイルを選択 ダイアログによる指定				
	指定可能值					
ファイルの種類	追加するダウンロード・ファイルのファイル形式を指定します。 ここでは, [ヘキサ・ファイル], [S レコード・ファイル]または [バイナリ・データ・ ファイル] を選択します。					
	デフォルト	ロード・モジュール・ファイル				
	変更方法	ドロップダウン・リストによる選択				
	指定可能值	次のいずれか				
		- ロード・モジュール・ファイル				
		- ヘキサ・ファイル				
		- S レコード・ファイル				
		- バイナリ・データ・ファイル				
オフセット	この項目は, S フォーマッ 指定したファ す。	ダウンロードするファイルがヘキサ・フォーマット(*.hex)/モトローラ ト(*.mot)の場合のみ表示されます。 イルのダウンロードを開始するアドレスからのオフセット値を指定しま				
	デフォルト	0				
	変更方法	キーボードからの直接入力				
	指定可能值	0x0 ~ 0xFFFFFFF の 16 進数				

RENESAS

開始アドレス	この項目は, 場合のみ表示 す。	この項目は,ダウンロードするファイルがバイナリ・データ・フォーマット(*.bin)の 湯合のみ表示されます。指定したファイルをダウンロードする開始アドレスを指定しま ⁻ 。			
	デフォルト	0			
	変更方法	キーボードからの直接入力			
	指定可能值	0x0 ~ 0xFFFFFFF の 16 進数			

備考オブジェクト情報,およびシンボル情報をダウンロードするか否かの指定は,ダウンロードするファイルの種類がロード・モジュール・ファイルの場合のみ行うことができます。

(4) [OK] ボタンのクリック このダイアログでの設定を有効とし、指定したファイルがダウンロード・ファイルとして追加されます(プロパ ティパネルの[ダウンロード・ファイル設定]タブ上の[ダウンロード]カテゴリ内に追加したファイル名と ダウンロード条件が表示されます)。

2.5.2.3 複数のロード・モジュール・ファイルをダウンロードする

複数のロード・モジュール・ファイルをダウンロードする場合は、ダウンロード・ファイル ダイアログにおいて、次の手順の操作を行ってください。

- **注意** 複数のロード・モジュール・ファイルから構成されるプログラムをデバッグする際は、配置アドレスが 重ならないよう注意が必要です。
- 備考 複数のロード・モジュール・ファイルをダウンロードした場合, イベント パネルの詳細情報にはロー ド・モジュール名が表示されます。
- (1) [追加] ボタンのクリック
 [追加] ボタンをクリックすると、[ダウンロード・ファイル一覧] エリアの最終行に空欄の項目("-")が表示されます。
- (2) 追加するダウンロード・ファイルのプロパティ設定 [ダウンロード・ファイルのプロパティ]エリアにおいて、追加するロード・モジュール・ファイルの選択とダ ウンロード条件を設定します。 表示される各項目において、次の設定を行ってください。

設定を完了すると、[ダウンロード・ファイル一覧]エリア内の空欄の項目に、ここで指定したファイル名が反 映されます。

ファイル	追加するロード・モジュール・フォーマットのファイルを指定します(最大指定文字 数:259文字)。						
	デフォルト	空欄	空欄				
	変更方法	キーボード; 表示される ファイルを;	キーボードからの直接入力,またはこのプロパティを選択すると右端に 表示される […] ボタンのクリックによりオープンするダウンロードする ファイルを選択 ダイアログによる指定				
	指定可能值	「表 2.2 ダウンロード可能なファイル形式」参照					
ファイルの種類	追加するダウ ここでは, [ロ	るダウンロード・ファイルのファイル形式を指定します。 は, [ロード・モジュール・ファイル]を選択します(デフォルト)。					
オブジェクトをダ	指定したファイルからオブジェクト情報をダウンロードするか否かを指定します。						
ワンロートする	デフォルト	はい					
	変更方法	ドロップダウン・リストによる選択					
	指定可能值	はい	オブジェクト情報をダウンロードします。				
		いいえ	オブジェクト情報をダウンロードしません。				



 ⁽³⁾ ダウンロードの際の実行順序の確認
 [ダウンロード・ファイルー覧]エリアでの表示順序が、ダウンロードの際の実行順序となります。
 順序を変更する場合は [↓] / [↑]ボタンで変更してください。

シンボルをダウン	指定したファ	イルからシン	νボル情報をダウンロードするか否かを指定します ^{注1} 。		
ロートする	デフォルト	はい			
	変更方法	ドロップダ	ウン・リストによる選択		
	指定可能值	はい	シンボル情報をダウンロードします。		
		いいえ	シンボル情報をダウンロードしません。		
PIC/PID オフセッ トを指定する	ダウンロード び PID(Posit するか否かを セット ", "PI	[:] するロード・モジュールの PIC(Position Independent Code)領域,ま ition Independent Data)領域の位置をロード・モジュール作成時から変 :指定します。この項目を " はい " に切り替えると,サブ項目に "PIC オ T ID オフセット " 項目を表示します。			
	デフォルト	いいえ			
	変更方法	ドロップダ	ウン・リストによる選択		
	指定可能值	はい	PIC/PID オフセットを指定します ^{注 2} 。		
		いいえ	PIC/PID オフセットを指定しません。		
PIC オフセット	ロード・モジ 例えば, プロ 目に "1000" を	ュール作成時 グラム・セク E入力すると、	fのアドレスからのオフセット値を入力します。 ションの開始アドレスが 0x1000 番地からだった場合に本項 対象のセクションは 0x2000 番地にダウンロードされます。		
	デフォルト	0			
	変更方法	キーボードからの直接入力			
	指定可能值	0x0 ~ 0xFFFFFFF の 16 進数			
PID オフセット	ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を入す。 例えば,ロード・モジュールの実行時に PID レジスタに 0x200 を設定する場合は, 目に "200" を入力してください。				
	デフォルト	0			
	変更方法	キーボード	からの直接入力		
	指定可能值	0x0 ~ 0xFFFFFFF の 16 進数			

注 1. シンボル情報をダウンロードしない場合、ソース・レベル・デバッグを行うことはできません。

注 2. PIC/PID 機能(「2.7 PIC/PID 機能の使用」参照)を使って作成していないロード・モジュールに 対して"はい"を選択した場合の動作は保証できません。

備考 シンボル情報が不要なロード・モジュール・ファイルの場合では、[シンボルをダウンロードする]項目を[いいえ]に設定することにより、メモリの使用量を軽減することができます(ただし、このファイルをソース・レベルでデバッグすることはできません)。

(3) ダウンロードの際の実行順序の確認 [ダウンロード・ファイルー覧]エリアでの表示順序が、ダウンロードの際の実行順序となります。 順序を変更する場合は[↓] / [↑]ボタンで変更してください。

(4) [OK] ボタンのクリック このダイアログでの設定を有効とし、指定したロード・モジュール・ファイルがダウンロード・ファイルとして 追加されます(プロパティパネルの[ダウンロード・ファイル設定]タブ上の[ダウンロード]カテゴリ内に 追加したファイル名が表示されます)。 2.5.2.4 ヘキサ・フォーマット/モトローラSフォーマット/バイナリ・データ・フォー マットのファイルでソース・レベル・デバッグを行う

へキサ・フォーマット (*.hex), モトローラ S フォーマット (*.mot), またはバイナリ・データ・フォーマット (*.bin)のファイルをダウンロードの対象ファイルと指定している場合でも, 作成元となったロード・モジュール・ファ イルのシンボル情報を併せてダウンロードすることにより, ソース・レベル・デバッグを行うことができます。 この場合は, ダウンロード・ファイル ダイアログにおいて, 次の手順の操作を行ってください。

- (1) [追加] ボタンのクリック
 [追加] ボタンをクリックすると、[ダウンロード・ファイル一覧] エリアの最終行に空欄の項目("-")が表示されます。
- (2) ロード・モジュール・ファイルのプロパティ設定 [ダウンロード・ファイルのプロパティ]エリアにおいて、各項目を次のとおりに指定します。

ファイル	ダウンロードするヘキサ・フォーマット (*.hex), モトローラ S フォーマット (*.mot), またはバイナリ・データ・フォーマット (*.bin)のファイルを作成する元となったロー ド・モジュール・ファイルを指定します。 キーボードからの直接入力, または [] ボタンのクリックによりオープンするダウン ロードするファイルを選択 ダイアログにより指定してください。
ファイルの種類	[ロード・モジュール・ファイル]を指定します(デフォルト)。
オブジェクトをダ ウンロードする	[いいえ]を指定します。
シンボルをダウン ロードする	[はい]を指定します(デフォルト)。
PIC/PID オフセッ トを指定する	[いいえ] を指定します(デフォルト) ^注 。

注 ヘキサ・フォーマット、モトローラ S フォーマット、またはバイナリ・データ・フォーマットの ファイルは PIC 領域、 PID 領域を区別してダウンロードすることができません。このためロー ド・モジュール・ファイルで PIC/PID オフセットを指定してもデバッグ機能と一致しなくなり、 デバッグできません。

このダイアログでの設定を有効とし、指定したロード・モジュール・ファイルがダウンロード・ファイルとして 追加されます(ロード・モジュール・ファイル内に含まれるシンボル情報のみがダウンロードの対象となりま す)。

2.5.2.5 外部フラッシュ・メモリヘダウンロードする【E1】【E20】

ダウンロード・ファイル ダイアログで指定したダウンロード・ファイルを、使用するマイクロコントローラの外部 バスに接続されたフラッシュ・メモリ(外部フラッシュ・メモリ)にダウンロードすることができます。 この場合は、次の手順の操作を行ってください。

(1) プロパティ パネルの設定 プロパティ パネルの [接続用設定] タブ上の [外部フラッシュ]【E1】【E20】カテゴリ内において、外部フ ラッシュ定義ファイル (USD ファイル)の登録を行います。 外部フラッシュ・メモリは、[外部フラッシュ]【E1】【E20】カテゴリに USD ファイルを登録することで認識 され、最大4つの USD ファイルを登録できます。 外部フラッシュ定義ファイルの"+"マークをクリックすると、インデックス([1]~[4])毎に定義ファイル名、 開始/終了アドレスおよびダウンロード前に外部フラッシュ ROM を消去するか否かの選択状況を表示します。

図 2.84 [外部フラッシュ]カテゴリ

1	外部フラッシュ	
	外部フラッシュ定義ファイル	[4]
	⊳ [0]	
	▶ [1]	
	▷ [2]	
	⊳ [3]	

^{(3) [}OK] ボタンのクリック

- [外部フラッシュ定義ファイル]
 - 登録可能な USD ファイルの最大数を表示します。

USD ファイルを登録する場合,サブプロパティとして展開表示される[ファイル]プロパティに USD ファイ ル名を指定します。

ファイル名は直接入力するか(相対パスによる指定の場合はプロジェクト・フォルダ基点), このプロパティ を選択することで設定欄右端に表示される[...] ボタンのクリックによりオープンする 外部フラッシュメモ リダイアログ【E1】【E20】によりファイルを選択することで指定します。

USD ファイルを登録すると、サブプロパティとして [開始アドレス] / [終了アドレス] / [ダウンロード前に外部フラッシュ ROM を消去する] プロパティが追加表示されます。

[開始アドレス]/[終了アドレス]プロパティには、登録した USD ファイルに定義されている外部フラッシュメモリの領域が表示されます(編集不可)。

[ダウンロード前に外部フラッシュ ROM を消去する] プロパティでは、ダウンロード実行前に外部フラッシュ ROM を消去するか否かをドロップダウン・リストにより指定します。

外部フラッシュ ROM を消去する場合は [はい] を指定してください(デフォルトでは [いいえ] が指定されま す)。

また, USD ファイルの登録を解除する場合, 対象となる [ファイル] プロパティのファイル名にキャレットを 移動したのち, [Delete] キーをクリックしてください。

なお, USD ファイルについては,弊社 Web サイトより External Flash Definition Editor のユーザーズ・マニュアルを参照してください。

図 2.85 外部フラッシュメモリ ダイアログのオープン

4	外部フラッシュ		
4	外部フラッシュ定義ファイル	[4]	
	▲ [0]	C#Sample¥usd¥ExFlash.usd	\frown
	774	C:¥Sample¥usd¥ExFlash.usd	
	開始アドレス	HEN 7000000	\smile
	終了アドレス	HEX 7FFFFFF	
	ダウンロード前に外部フラッシュROMを消去する	いいえ	
	▶ [1]		
	▷ [2]		
	▶ [3]		

- 注意 1. USD ファイルのエンディアン情報は、プロパティ パネルの[デバッグ・ツール設定] タブにおける [メモリ] カテゴリ内の [メモリ・マッピング] プロパティに反映されません。そのため、対象となる外部領域のエンディアンを USD ファイルのエンディアン情報に合わせて変更してください。
- **注意 2.** 重複するアドレスの USD ファイルを複数設定した場合,デバッグ・ツールとの接続ができません。
- **注意 3.** 外部フラッシュ・メモリヘダウンロードする場合, ワーク RAM として内蔵 RAM と CS 領域の RAM が使用可能です。SDRAM 領域(SDCS)は使用できません。
- **注意 4.** CS 領域をまたいでデバイスを割り当てる場合は、プロパティ パネルの [ダウンロード前にフ ラッシュ ROM を消去する]を [いいえ] に設定してください。
- **注意 5.** ワーク RAM 領域は、メモリ内容の退避・復帰を行うため、ユーザ・プログラムでも使用できますが、 ワーク RAM に配置する領域は、以下の用途では使用できません。
 - DMA または DTC 機能の転送元・転送先
 - DTC ベクタテーブル,および転送情報配置アドレス
 - DMAC, DTC 起動要因の割り込みベクタ
- 備考 メーカ ID またはデバイス ID が不一致の場合、リードした値と期待値が出力 パネルに表示されます。





出力		×
バージョン情報:」	0.04.00.000.1	*
Bf wE20rx600, s.	2.04.00.003	
LEVELO	1.00.00.000	
LEVEL1 Communic di L	2.04.00.001	
FPGA	0C1	=
Communication FPGA	0E,J	
エミュレーダ情報に Faulator Board Bavision	E1 Rev 02.1	
User Vcc	3.30 YJ	-
エラー(E0204001): ダウンロー	ドに失敗しました。」	
「エラーの直接原因」」		
外部フラッシュROMのメーカIDま	たはデバイス10が一致しません。([0x07000000-0x07ffffff])	
メー力1D=0x0020(期待値:0xffff 0x7026476)(E1012004)), デバイスID=0x225b(期待値:0x227e))(address:0x7000000-	
[E0F]		
		Ψ.
ヽ!すべてのメッセージ! ヘ *ビルド・ツ	ール / *デバッグ・ツール /	-

(2) CPU 動作モードの設定

外部フラッシュ・メモリヘダウンロードする場合、デバッグ・ツールと接続する際にマイクロコントローラの動作モードを設定する必要があります。プロパティパネルの[接続用設定]タブにおける[CPUの動作モード] 【E1】【E20】【EZ Emulator】カテゴリ内の[レジスタ設定]プロパティをプルダウン・リストにて次のように[内蔵 ROM 有効拡張モード]または[内蔵 ROM 無効拡張モード]に変更します。

図 2.87 CPU 動作モードの指定

å	CPUの動作モード		
	モード端子設定	シングルチップモード	
	レジスタ設定	内蔵ROM有効拡張モード	1 4 1
	エンディアン	シングルチップモード	
		内蒂ROM有効拡張モード	
		内蔵ROM無効拡張モード	

- (3) ダウンロードの実行 ダウンロードを実行します(「(3) ダウンロードの実行」参照)。
- **注意 1.** メモリ パネル上で外部フラッシュ・メモリを書き換えることはできません。
- **注意 2.** 外部フラッシュ・メモリヘダウンロード時に外部 RAM をワーク RAM として使用する場合は、CPU と 同じエンディアンに設定してください。
- **注意 3.** セクタ数が 4096 以下のフラッシュ・メモリを登録可能です。 セクタ数が 4096 を越えるフラッシュ・メモリへの書き込みは保証できません。
- 注意 4. ロックコマンドが使用できないフラッシュ・メモリデバイスを使用する場合, USD ファイルの生成時 にロックビット解除を指定してください。この指定により,不要なロックビット状態の確認処理を省略 することができます。

また, プロパティ パネルの [ダウンロード前にフラッシュ ROM を消去する] を [いいえ] に設定して ください。

* ロックコマンドとは、以下のコマンドシーケンスで実行するコマンドを示します。 コマンドシーケンス:

	第1サイクル	第 2 サイクル	第 3 サイクル	第 4 サイクル
アドレス	555H	2AAH	555H	ブロックアドレス
データ	AAH	55H	48H	X1H / X0H

2.5.3 アップロードを実行する

現在接続しているデバッグ・ツールのメモリ内容を,任意のファイルに保存(アップロード)することができます。 アップロードは,[デバッグ]メニュー→[デバッグ・ツールからアップロード…]を選択することによりオープン するデータ保存 ダイアログで行います。

このダイアログにおいて、次の手順で操作を行ってください。

図 2.88	メモリ内容のアップロー	ド(データ保存	¹ ダイアログ)
--------	-------------	---------	---------------------

データ保存 - デバッグ・ツールからアップロード							
ファイル名(N):	(ファイル名を入力してくださ	5(1)					.
ファイルの種類(王): - 保存範囲 アドレス	インテル・ヘキサ・フォーマット ノシンボル(<u>A</u>):	۱× (*)	hex)				•
(保存範囲の関数	台位置を入力してく №▼	- [(保存範囲	目の終	了位置を	く力し	.7(₩ ▼
			(果存(<u>S</u>)		キャンセル		∧ル⊅(H)

- (1) [ファイル名]の指定 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[...]ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより、ファイル を選択することもできます。
- (2) [ファイルの種類]の指定 保存するファイルの形式を次のドロップダウン・リストにより選択します。 選択できるファイルの形式は次のとおりです。

表 2.3 アップロード可能なファイル形式

リスト表示	形式
インテル・ヘキサ・フォーマット (*.hex)	ヘキサ・フォーマット ^注
モトローラSフォーマット (*.mot)	Sレコード・フォーマット
バイナリ・データ (*.bin)	バイナリ・データ・フォーマット

注 インテル・ヘキサ・フォーマットでは常に"拡張リニアアドレスレコード(32ビット・アドレス)"で保存します。

(3) [保存範囲 アドレス / シンボル]の指定 ファイルに保存する範囲を"開始アドレス"と"終了アドレス"で指定します。 それぞれのテキスト・ボックスに 16 進数の数値/アドレス式を直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10 個)。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシ ンボル名を補完することができます。(「2.21.2 シンボル名の入力補完機能」参照)。

(4) [保存] ボタンのクリック 指定したファイルに指定した形式で、メモリ内容をアップロード・データとして保存します。



2.6 プログラムの表示と変更

この節では、デバッグ情報を持ったロード・モジュール・ファイルをデバッグ・ツールにダウンロードした場合のプ ログラムの表示方法、およびその変更方法について説明します。

- ダウンロードしたプログラムの表示は、次の2つのパネルで行うことができます。
- エディタ パネル
- ソース・ファイルの表示/編集を行うほか、ソース・レベル・デバッグ(「2.9.3 プログラムをステップ実行する」 参照)、およびコード・カバレッジ測定結果の表示(「2.15.2 カバレッジ測定結果を表示する」参照)を行います。
- 逆アセンブル パネル

Tutavial a

ダウンロードしたプログラム(メモリ内容)の逆アセンブル結果の表示/編集(ライン・アセンブル)を行うほか、 命令レベル・デバッグ(「2.9.3 プログラムをステップ実行する」参照)、およびコード・カバレッジ測定結果の表示(「2.15.2 カバレッジ測定結果を表示する」参照)を行います。 なお、このパネルでは、逆アセンブル結果の表示とともに、対応するソース・テキストも表示することができます (デフォルト)。

 備考 通常、ソース・レベル・デバッグを行うためには、デバッグ情報を持つロード・モジュール・ファイル をダウンロードする必要がありますが、ヘキサ・フォーマット(*.hex)、モトローラSフォーマット (*.mot)、またはバイナリ・データ・フォーマット(*.bin)のファイルをダウンロード対象として、 ソース・レベル・デバッグを行うことも可能です(「2.5.2.4 ヘキサ・フォーマット/モトローラS フォーマット/バイナリ・データ・フォーマットのファイルでソース・レベル・デバッグを行う」参 照)。

2.6.1 ソース・ファイルを表示する

ソース・ファイルの表示は、次のエディタ パネルで行います。

エディタパネルは、ロード・モジュール・ファイルが正常にダウンロードされると、指定された位置(「2.5.1 ダウン ロードを実行する」参照)のソース・テキストを表示した状態で自動的にオープンします。

手動でエディタ パネルをオープンする場合は、プロジェクト・ツリー パネルにおいてソース・ファイルをダブルク リックしてください。

なお,各エリアの見方,および機能についての詳細は,「CS+ 統合開発環境 ユーザーズマニュアル エディタ編」を参照してください。

図 2.89 ソース・ファイルの表示(エディタ パネル)

Ī	井 アドレス	D		
18	ffff8736	19 14	void tutorial(void)	_
0		1111	⊒{	
1			long a[10];	
2			long j;	
3			int i;	
4			struct Sample far *p_sam;	
5			and have not black and a second	
3	ffff873c		<pre>p_sam= malloc(sizeof(struct Sample));</pre>	
7	ffff8742		init(p_sam);	
3				
3	ffff8748		for(i=0; i<10; i++){	
0	ffff874d	Ĭ	<pre>i = rand();</pre>	
	ffff9759		16(1 (0))	

2.6.2 逆アセンブル結果を表示する

ダウンロードしたプログラム(メモリ内容)の逆アセンブル結果(逆アセンブル・テキスト)の表示は、次の逆アセンブルパネルで行います。

[表示] メニュー→ [逆アセンブル] → [逆アセンブル 1 ~ 4] を選択してください。

逆アセンブル パネルは,最大4個までオープンすることができ,各パネルはタイトルバーの"逆アセンブル1","逆 アセンブル2","逆アセンブル3","逆アセンブル4"の名称で識別されます。

なお,各エリアの見方,および機能についての詳細は,逆アセンブルパネルの項を参照してください。

図 2.90 逆アセンブル結果の表示(逆アセンブル パネル)



備考 ツールバーの [表示]→ 第 ボタンをクリックすることによりオープンするスクロール範囲設定 ダイア ログにより、このパネルの垂直スクロール・バーのスクロール範囲を設定することができます。

ここでは、次の操作方法について説明します。

2.6.2.1 表示モードを変更する
2.6.2.2 表示形式を変更する
2.6.2.3 指定アドレスへ移動する
2.6.2.4 シンボル定義箇所へ移動する
2.6.2.5 逆アセンブル結果の表示内容を保存する

2.6.2.1 表示モードを変更する

ツールバーの ボタン (トグル)をクリックすることにより, 逆アセンブル パネルの表示モードを切り替えることができます。

- 混合表示モード

ソース・テキストと逆アセンブル・テキストを併せて表示する、デフォルトの表示モードです。



逆アセンブル1			×
2 30 🜛 🖇	🦉 辦 🛛 表示(「ソース・テキス		
ffff85a2	670	RTSD	#DCH 🔺
36:	tutorial	0;)
	main:		
🔲 🔶 ffff85a4	390500	BSR.W	tutorial
ffff85a7	2efd	BRA - B	main
40:	void tutorial (void)		
	tutorial: 逆アセン	/ブル・テキスト	
ffff85a9	7100cc	HUU	#-34H.RO.R
47:	p san = &g Sampl	e:	
ffff85ac	fb1208040000	NOV-L	#00000408H
ffff85b2	a301	NOV-L	R1,30H[R0]
48:	init(p_sam);		
ffff85b4	ab01	NOV.L	30H[R0],R1
ffff85b6	057f0000	BSR-A	init
50:	for(i=0; i<10;	i++){	-
ffff85ha	300h00	NOV 1	100H 20H [D
< III			F.

図 2.91 混合表示モード(逆アセンブル パネル)

- 逆アセンブル表示モード

ソース・テキストを非表示にし、逆アセンブル・テキストのみを表示します。

逆アセンブル1			×
2 31 🜛 🕺 8	表示(<u>I</u>)-		
ffff85a0	a009	NOV.L	R1,04H[R0] _
ffff85a2	6703	RTSD	#OCH
➡ ffff85a4	390500	BSR.W	<u>_tutorial</u> _
ffff85a7	2efd	BRA.B	_main
ffff85a9 ffff85ac ffff85b2 ffff85b4 ffff85b6 ffff85ba	ial: 7100cc fb1208040000 a301 ab01 057f0000 3e0b00 2e29	ADD NOV.L NOV.L BSR.A NOV.L	#-34H,R0,R #00000408H R1,30H[R0] 30H[R0],R1 init #00H,2CH[R
ffff85bf	66a1		#DAH,R1
ffff85c1	0681010b		2CH[R0],R1
ffff85c5	9281		P1 28H[P0]

図 2.92 逆アセンブル表示モード(逆アセンブル パネル)

2.6.2.2 表示形式を変更する

逆アセンブル結果の表示形式は、ツールバーの次のボタンにより、自由に変更することができます。

表示		表示形式を変更する次のボタンを表示します。
	范	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場合,一番近いラ ベルからのオフセット値を表示します。
	A DEC	アドレス値を " シンボル + オフセット値 " で表示します(デフォルト)。 ただし,アドレス値にシンボルが定義されている場合は,シンボルのみを表示します。

2.6.2.3 指定アドレスへ移動する

逆アセンブル・テキスト上の指定アドレスへの移動は、コンテキスト・メニューの [移動 ...]を選択することにより オープンする指定位置へ移動 ダイアログで行います。

RENESAS

このダイアログにおいて、次の手順で操作を行ってください。

図 2.93 逆アセンブル結果内のアドレスへ移動(指定位置へ移動 ダイアログ)

指定位置へ移動	×
アドレス/シンボル(A):	•
ОК	キャンセル ヘルプ(日)

(1) [アドレス / シンボル]の指定

キャレットを移動したいアドレスを指定します。 テキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024 文字),またはドロップダウン・リ ストより入力履歴項目を選択します(最大履歴数:10 個)。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシ ンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(2) [OK] ボタンのクリック 指定したアドレスヘキャレットを移動します。

2.6.2.4 シンボル定義箇所へ移動する

シンボルが定義されているアドレスに、キャレット位置を移動することができます。 シンボルを参照している命令にキャレットを移動したのち、ツールバーの また、この操作に続き、ツールバーの デボタンをクリックすると、キャレット移動前のシンボルを参照している命 令にキャレット位置を戻します。

2.6.2.5 逆アセンブル結果の表示内容を保存する

逆アセンブル結果の内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存することができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得し、このパネル上での表示形式に従ったデータで 保存します。

[ファイル] メニュー→ [名前を付けて逆アセンブル・データを保存 …] を選択すると,次のデータ保存 ダイアログ がオープンします (この際,パネル上で範囲選択した状態でこの操作を行うと選択範囲のみの逆アセンブル・データを 保存することができます)。

このダイアログにおいて、次の手順で操作を行ってください。

図 2.94 逆アセンブル・データの保存(データ保存 ダイアログ)

データ保存 - 逆アセンブル・データ				
ファイル名(団):	ル名(N): C+WorkSpace+Tutorial#E1E20+R%600+Tutorial_LittleEnd PF 🔝			
ファイルの種類①	「テキスト・ファイル (*.txt) ・			
保存範囲 アドレス	シンボル(<u A):			
_main	_tutorial+0x18			
	(保存(S) キャンセル ヘルプ(H)			

- (1) [ファイル名]の指定 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択ダイアログにより、ファイル を選択することもできます。
- (2) [ファイルの種類]の指定保存するファイルの形式を次のドロップダウン・リストにより選択します。

RENESAS

選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式(デフォルト)
CSV(カンマ区切り)(*.csv)	CSV 形式 ^注

注 各データを "," で区切り保存します。

(3) [保存範囲 アドレス/シンボル]の指定

ファイルに保存する範囲を"開始アドレス"と"終了アドレス"で指定します。 それぞれのテキスト・ボックスに16進数の数値/アドレス式を直接入力するか、またはドロップダウン・リス トより入力履歴項目を選択します(最大履歴数:10個)。 なお、パネル上で範囲選択している場合は、デフォルトでその選択範囲がテキスト・ボックスに指定されます。 範囲選択していない場合は、現在のパネルの表示範囲が指定されます。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシ ンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(4) [保存] ボタンのクリック 指定したファイルに、指定した形式で逆アセンブル・データを保存します。

図 2.95 逆アセンブル・データ保存の際の出力イメージ

 ラベル(シンボル名)			← ラベル(シンボル)行
: ファイル名	行番号	c ソース	← ソース・テキスト行
・ アドレス オフセット : : :	・ コード :	. 逆アセンブル結果 :	← 逆アセンブル結果行

- 備考 1. [ファイル]メニュー→[逆アセンブル・データを保存]の選択によりパネルの内容を上書き保存 する場合,逆アセンブルパネル(逆アセンブル1~4)はそれぞれ個別に扱われます。 また,保存範囲についても,前回指定したアドレス範囲で保存されます。
- 備考 2. [ファイル]メニュー→[印刷 ...]を選択することにより,現在このパネルで表示している画像イ メージを印刷することができます。

2.6.3 他の処理と平行してビルドを実行する

CS+では、次のタイミングでビルドを自動で開始する機能を提供しています(ラピッド・ビルド機能)。

- (1) デバッグ専用プロジェクト以外の場合
 - プロジェクトに追加しているCソース・ファイル/C++ソース・ファイル/アセンブラ・ソース・ファイル
 /ヘッダ・ファイル/リンク・ディレクティブ・ファイル/シンボル情報ファイル/オブジェクト・モジュール・ファイル/リロケータブル・モジュール・ファイル/ライブラリ・ファイルのいずれかを更新したとき
 - プロジェクトにビルド対象ファイルを追加、または削除したとき
 - オブジェクト・モジュール・ファイルのリンク順を変更したとき
 - ビルド・ツール、およびビルド対象ファイルのプロパティを変更したとき
- (2) デバッグ専用プロジェクトの場合
 - デバッグ専用プロジェクトに追加しているC ソース・ファイル/ C++ ソース・ファイル/アセンブラ・ソー ス・ファイル/ヘッダ・ファイルを編集して保存したとき
 - デバッグ専用プロジェクトにCソース・ファイル/C++ソース・ファイル/アセンブラ・ソース・ファイル
 /ヘッダ・ファイルを追加、または削除したとき
 - デバッグ専用プロジェクトのプロパティを変更したとき

ラピッド・ビルド機能を有効にすることにより、上記の操作と平行してビルドを行うことができます。

なお、データ内に","が含まれている際の不正形式を避けるため、各データを""""(ダブルクォー テーション)で括り出力します。

ラピッド・ビルド機能の有効/無効の設定は、[ビルド]メニュー→ [ラピッド・ビルド]の選択により切り替えます (デフォルトで有効に設定されています)。

- 注意 外部エディタを使用する場合、この機能を有効にするためには、オプションダイアログの [全般 ビルド/デバッグ] カテゴリの [登録されたファイルの変更を監視する] をチェックする必要があります。
- 備考 1. ソース・ファイル編集後、[Ctrl] + [S] キーの押下により、こまめに上書き保存することを推奨します。
- 備考 2. ラピッド・ビルドの有効/無効は、プロジェクト全体(メイン・プロジェクト、およびサブプロジェク ト)に対して設定されます。
- 備考 3. ラピッド・ビルドの実行中に、ラピッド・ビルドを無効に切り替えた場合は、その場でラピッド・ビル ドの実行を中止します。

2.6.4 ライン・アセンブルを行う

逆アセンブル パネルで表示されている命令/命令コードは、編集(ライン・アセンブル)することができます。 ここでは、次の操作方法について説明します。

2.6.4.1 命令を編集する 2.6.4.2 命令コードを編集する

2.6.4.1 命令を編集する

命令を編集する場合は、次の手順で操作を行ってください。

- (1) 編集モードへの切り替え 対象命令をダブルクリックするか、または対象命令にキャレットを移動した状態でコンテキスト・メニューの [命令の編集]を選択すると、編集対象が編集モードに切り替わります。
- (2) 命令の編集 キーボードから直接命令の文字列を編集します。
- (3) メモリへの書き込み 編集終了後、[Enter] キーを押下することにより、変更された命令が自動的にライン・アセンブルされ、コード がメモリに書き込まれます。 ただし、この際に、変更結果が不正な命令となる場合は、編集された文字列が赤色で表示され、メモリへの書き 込みは行いません。

なお、表示されている逆アセンブル結果を別の命令で上書きすることによりメモリに空きが生じた場合、次の例のように自動的に nop 命令でバイト数を補います。

例 1. 3 行目の ADD 命令(6 バイト命令)を BCLR 命令(4 バイト命令)で上書きした場合

編集前	5020	AND	[R2].UB,R0
	99e0	MOV.W	OCH[R6],R0
	700fb98a322a	ADD	#2A328AB9H,R0,R15
	5327	AND	R2,R7
編集後	5020 99e0 f22a45b2 03 03 5327	AND MOV.W BCLR NOP NOP AND	[R2].UB,R0 OCH[R6],R0 #2,0B245H[R2] R2,R7

例 2.

2 行目の MOV.W 命令(2 バイト命令)を BCLR 命令(4 バイト命令)で上書きした場合

編集前	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6].R0
	700fb98a322a	ADD	#2A328AB9H,R0,R15
	5327	AND	R2,R7



編集後	5020	AND	[R2].UB,R0
	99e0	MOV.W	0CH[R6],R0
	f22a45b2	BCLR	#2,0B245H[R2]
	03	NOP	
	5327	AND	R2,R7

2.6.4.2 命令コードを編集する

命令コードを編集する場合は、次の手順で操作を行ってください。

- (1) 編集モードへの切り替え 対象命令コードをダブルクリックするか、または対象命令コードにキャレットを移動した状態で表示されるコン テキスト・メニューの[コードの編集]を選択すると、編集対象が編集モードに切り替わります。
- (2) 命令コードの編集 キーボードから直接命令コードの文字列を編集します。
- (3) メモリへの書き込み 編集終了後、[Enter] キーを押下することにより、命令コードがメモリに書き込まれます。 ただし、この際に、変更結果が不正な命令となる場合は、編集された文字列が赤色で表示され、メモリへの書き 込みは行いません。 命令コードがメモリに書き込まれた場合は、逆アセンブル結果も同時に更新されます。

R20UT4977JJ0100 Rev.1.00 2021.06.01



2.7 PIC/PID 機能の使用

PIC/PID 機能は、一度リンクが完了して配置アドレスが確定した ROM 上のコードやデータを、リンクをやり直すことなく、任意のアドレスに配置して利用できるようにする機能です。

- PIC/PID 機能を利用する場合, プログラムは "マスタ "と "アプリケーション"で構成します。
- PIC/PID 機能では, ROM 上のコードやデータを PIC や PID にしたプログラムを " アプリケーション ", " アプリケー ション " を実行させるのに必要なプログラムを " マスタ " と呼びます。
- この節では, ROM 上のコードやデータを PIC や PID にした " アプリケーション " プログラム (ロード・モジュール) を, 別のアドレスに配置を変更してデバッグする方法について説明します。

- PIC

pic オプションを有効にしてコンパイルすると、PIC 機能が有効になり、コード領域(P セクション)が PIC になり ます。PIC は分岐先アドレスや関数アドレスの取得をすべて PC 相対で行うため、リンク後も任意のアドレスに配 置することができます。

- PID

pid オプションを有効にしてコンパイルすると、PID 機能が有効になり、ROM データ領域(C, C_2, C_1, W, W_2, W_1, およびL セクション)が PID になります。プログラムは PID に対し、その先頭アドレスを示すレジスタ (PID レジスタ)から相対のアクセスでアクセスします。"マスタ"で PID レジスタの設定値を変化させることにより、リンク後も PID を任意のアドレスに移動することができます。

- 備考 1. PIC/PID 機能については、「CC-RX コンパイラ ユーザーズマニュアル」を参照してください。
- 備考 2. ビルド・ツールにおける PIC/PID 機能の設定については,「CS+ 統合開発環境 ユーザーズマニュアル CC-RX ビルド・ツール操作編」を参照してください。

ROM 上のコードやデータを PIC や PID にしたロード・モジュールのアドレスの配置を変更したのち,デバッグを始めるには、次の手順で行ってください。

- (1) ダウンロード・ファイルの追加
 "アプリケーション"のロード・モジュール・ファイルを"マスタ"のダウンロード・ファイルとして追加します(「2.7.1 PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する」参照)。
- (2) ロード・モジュール・ファイルのダウンロード条件の指定
 "アプリケーション"のロード・モジュールに対し、ロード・モジュール作成時のアドレスからのオフセット値 ([PIC オフセット])、および作成時に指定した PID レジスタへ設定するオフセット値([PID オフセット])を指 定します。
- (3) ダウンロードの実行
 "マスタ"および"アプリケーション"のロード・モジュール・ファイルのダウンロードを実行します(「2.5.1 ダウンロードを実行する」参照)。
 新しいアドレスに配置された ROM 上のコードやデータをデバッグできるようになります。

2.7.1 PIC/PID 機能を使用したロード・モジュールのアドレスの配置を変更する

"マスタ"への"アプリケーション"のロード・モジュール・ファイルの追加は、プロパティパネルの[ダウンロード・ファイル設定] タブ上の[ダウンロード] カテゴリ内[ダウンロードするファイル] プロパティで行います。

図 2.96 [ダウンロード]カテゴリ

_▲ ダウンロー	*		
▶ ダウンロー	ドするファイル	[1]	
ダウンロー	ド後にCPUをリセットする	はい	

- [ダウンロードするファイル]

[...] ボタンをクリックすることにより、ダウンロード・ファイル ダイアログをオープンします。

ダウンロード・ファイル ダイアログの [ダウンロード・ファイル一覧] エリアの [追加] ボタンをクリックし, [ダ ウンロード・ファイルのプロパティ] エリアで "アプリケーション"のロード・モジュールの情報を設定します。

- [ファイル] ダウンロードする " アプリケーション " のロード・モジュール・ファイルを指定します。
- [PIC/PID オフセットを指定する] [はい]を選択します。これにより、[PIC オフセット] および [PID オフセット] が表示されます。

- [PIC オフセット]
- ロード・モジュール作成時のアドレスからのオフセット値を指定します。
- [PID オフセット] ロード・モジュール作成時に指定した PID レジスタへ設定するオフセット値を指定します。

図 2.97 ダウンロード・ファイルの追加とダウンロード条件の変更 (ダウンロード・ファイル ダイアログ)

ダウンロード・ファイル	
ダウンロード・ファイル一覧(E): Master abs	ダウンロード・ファイルのプロバティ(P):
Application.abs	7ァイル 7ァイル 7ァイルの種類 ロード・モジュール・ファイル オブジェクトを参ウンロードする はい PICオフセットを指定する はい PICオフセット IIII TO
isto(A) #IFR(E)	A ファイロン A Set A

[PIC オフセット] および [PID オフセット] の値を変更したのち, ロード・モジュールのダウンロードを実行する と,以下の図のように,Pセクションのアドレス,および,外部変数または静的変数のアドレスの配置が変更されます。 上の図は、ダウンロード・ファイル ダイアログの [PIC オフセット] および [PID オフセット] に "4000" を指定した 場合,下の図は、"0" を指定した場合に、ダウンロードを実行した例です。

上の図では, 元のアドレスに対して "4000" が加算されています。



図 2.98 [PIC オフセット]および [PID オフセット]のオフセット値を変更後のダウンロード実行例

WatchTest.c			
122	const char g_acc = 1: 🔨		
123	const signed char g_acsc = 2:		
124	const unsigned char g_acuc = 3: M		
<			
275 ffffd262 🍣	g_ac = g_acc :		
276 ffffd270	g_asc = g_acsc :		
277 ffffd27e	g_auc = g_acuc : 💌		
	>		
🔍 ウォッチ 1	- x		
🗃 🥮 🧞 🖏 🗙 表記 🗤 - 🔤			
ウォッチ式 値 型情報	辋(バイト数) アドレス メモ		
€ s_acc <mark>'r'(0x01)</mark> signe	d char(1) (0xffffd100)		
<			





2.8 オーバーレイ・セクションの設定

CC-RX が使用する最適化リンケージエディタ(OptInk)は、プログラム中に定義した複数のセクションを同じアドレスに割り付けることができる start オプションを提供しています。このように割り付けられたセクションを"オーバーレイ・セクション"と呼びます。

この節では、ロード・モジュール内のオーバーレイ・セクションに対する"オーバーレイ・セクションの選択機能"の設定方法について説明します。

プログラムの実行時、ロード・モジュール内の同じアドレスに割り付けられたセクションはいずれか1つのセクションしか実行されません。

デバッグ・ツールは、同じアドレスに割り付けられたセクションのうち、デバッグ対象とするオーバーレイ・セクション(以降,優先セクションと呼びます)を選択する "オーバーレイ・セクションの選択機能"を提供しています。

プログラムの実行前に優先セクションを選択することにより、非優先セクションのデバッグ情報が隠蔽されるため、 優先セクションをデバッグしやすくなります。

以下は、OptInkの start オプションを使用したオーバーレイ・セクションの定義例になります。

例 Optlnk によるオーバーレイ・セクションの定義

start = Psect01/1000,Psect02/2000,Psect03/3000,ram01:ram02:ram03/5000
rom = Psect01=ram01,Psect02=ram02,Psect03=ram03

rom オプションにより, "PsectXX" セクションと同サイズの "ramXX" セクションを確保し, "PsectXX" セクション内 定義シンボルを "ramXX" セクションのアドレス上でリロケーションします ("XX" は, この例では 01, 02, または 03 に なります)。

備考 ビルド・ツールにおけるオーバーレイ・セクションの設定については, 「CS+ 統合開発環境 ユーザーズ マニュアル CC-RX ビルド・ツール操作編」を参照してください。

図 2.99 オーバーレイ・セクションの配置



2.8.1 優先セクションを指定する

優先セクションの指定は、プロパティ パネルの [ダウンロード・ファイル設定] タブ上の [デバッグ情報] カテゴリ 内で行います。

図 2.100 [デバッグ情報] カテゴリ

	4	デバッグ情報	
		CPUリセット後に指定シンボル位置まで実行する	はい
	_	指定シンボル	main
1		デバッグするオーバーレイ・セクションを設定する	(ttt)
V	⊳	オーバーレイ・セクション	[3]

(1) [デバッグするオーバーレイ・セクションを設定する] ロード・モジュール内にオーバーレイ・セクションが存在する場合に,[はい]となります(変更不可)。

- (2) [オーバーレイ・セクション] オーバーレイ・セクションが存在するアドレス・グループを表示します。
 このプロパティは、[デバッグするオーバーレイ・セクションを設定する] プロパティにおいて [はい] が設定 されている場合のみ表示されます。
 - 図 2.101 [オーバーレイ・セクション] プロパティ

 デバッグ情報	
CPUリセット後に指定シンボル位置まで実行する	はい
指定シンボル	_main
デバッグするオーバーレイ・セクションを設定する	はい
オーバーレイ・セクション	[3]
▲ [0]	0×5000 - 0×5025
ファイル	C:\Project\sample\DefaultBuild\sample.abs
開始アドレス	HEX 5000
終了アドレス	HEN 5025
優先セクション	ram01
▶ [1]	0×5100 - 0×5125
▷ [2]	0x5200 - 0x5225

現在のオーバーレイ・セクションの情報をアドレス・グループごとに詳細表示します。なお、[優先セクション] のみ変更可能です。

- [ファイル] 選択されたアドレス・グループをダウンロードしたロード・モジュール・ファイル名です(変更不可)。
- [開始アドレス] 選択されたアドレス・グループの開始アドレスです(変更不可)。
- [終了アドレス] 選択されたアドレス・グループの終了アドレスです(変更不可)。
- [優先セクション] 選択されたアドレス・グループに定義されたセクションの一覧です。表示された一覧から、デバッグ対象とす るセクション(優先セクション)を選択します。
- **注意 1.** オーバーレイ・セクションに関する設定内容は、プロジェクト・ファイルに保存しません。 ロード・モジュールのダウンロード後は、デバッグ対象のセクションを再設定してください。
- **注意 2.** [優先セクション]項目の変更によって切り替わる情報は、デバッグ情報のみです。 デバッグ・ツールは、対象セクションのデータのコピーは行いません。

次に, [オーバーレイ・セクション] プロパティにおいて, 図 2.99 オーバーレイ・セクションの配置に記載している "ram01" を優先セクションとして選択した場合(左側),および "ram02" へ変更した(右側)場合の例を示します。





図 2.102 [優先セクション]の選択を "ram01" → "ram02" へ変更

2. 機

能



2.9 プログラムの実行

この節では、プログラムの実行方法について説明します。

- なお、この節で説明する主な操作は、プログラムの実行を制御するためのコマンドをまとめたメイン・ウインドウ上のデバッグ・ツールバー、または[デバッグ]メニューより行います。
 - **注意** デバッグ・ツールバー,および [デバッグ] メニューの各項目は,デバッグ・ツールと接続時のみ有効 となります。
 - デバッグ・ツールバー 図 2.103 デバッグ 🚯 🗅, 🐂 🔘 🕟 🕞 🐂 🖙 💭 👘 × 図 2.104 [デバッグ] メニュー デバッグ(D) 🔍 デバッグ・ツールヘダウンロード(D) ※ ビルド&デバッグ・ツールへダウンロード(B) F6 ₩ リビルド&デバッグ・ツールへダウンロード(90) ■ デバッグ・ツールへ接続(C) デバッグ・ツールからアップロード(U)... Shift+F6 ※ デバッグ・ツールから切断(N) 使用するデバッグ・ツール(L) ۲ Shift+F5 (停止⑤) 📡 実行(G) F5 F8 ⑦ ブレークせずに実行(E) S≡ ステップ・イン(1) F11 💭 ステップ・オーバー(O) F10 💁 リターン・アウト(B) Shift+F11 Ctrl+F5 CPUリセット(T) J

 J

- 2.9.1 マイクロコントローラ (CPU) をリセットする

デバッグ・ツールバーの 🛅 ボタンをクリックすることにより、CPU をリセットします。 CPU をリセットすることにより、カレント PC 値をリセット番地に設定します。

- 備考 CPU リセット後に、I/O レジスタ /CPU レジスタの値を指定した値に自動的に書き換える処理を設定することができます(「2.18 フック処理を設定する」参照)。
- 2.9.2 プログラムを実行する

プログラムを実行する操作方法には次の種類があります。 デバッグの目的に応じて,操作方法を選択してください。 なお,実行中のプログラムの停止方法については、「2.10 プログラムの停止(ブレーク)」を参照してください。

2.9.2.1 マイクロコントローラ (CPU) をリセットしてから実行する 2.9.2.2 現在のアドレスから実行する 2.9.2.3 PC 値を変更してから実行する

- 備考 1. プログラムの実行直前に, I/O レジスタ /CPU レジスタ値を指定した値に自動的に書き換える処理を設定することができます(「2.18 フック処理を設定する」参照)。
- 備考 2. プログラム実行中に,ステータス・バーへ任意の更新間隔で PC 値を表示することができます(「(e) カレント PC 位置」参照)。

2.9.2.1 マイクロコントローラ (CPU) をリセットしてから実行する

CPU をリセットしたのち、リセット番地からプログラムの実行を開始します。 操作は、デバッグ・ツールバーの ボタンをクリックします。 この操作によりプログラムの実行を開始した場合、次のいずれかの状態までその実行を続けます。

- 间 ボタンのクリック(「2.10.1 プログラムの実行を手動で停止する」参照)
- PC がブレークポイントに到達(「2.10.2 任意の場所で停止する(ブレークポイント)」参照)
- ブレーク・イベント条件の成立(「2.10.3 任意の場所で停止する(ブレーク・イベント)【E1】【E20】【EZ Emulator】」,「2.10.4 変数 /I/O レジスタへのアクセスで停止する」参照)
- その他のブレーク要因の発生

備考 この操作は, 🔄 ボタンをクリックしたのち, 🕟 ボタンをクリックした場合と同等です。

2.9.2.2 現在のアドレスから実行する

現在のアドレス(カレント PC 値で示されるアドレス)からプログラムの実行を開始する方法には、次の種類があります。

- (1) 通常の実行
 - デバッグ・ツールバーの 🕟 ボタンをクリックします。
 - この操作により実行を開始した場合、次のいずれかの状態までその実行を続けます。
 - 🔳 ボタンのクリック(「2.10.1 プログラムの実行を手動で停止する」参照)
 - PC がブレークポイントに到達(「2.10.2 任意の場所で停止する(ブレークポイント)」参照)
 - ブレーク・イベント条件の成立(「2.10.3 任意の場所で停止する(ブレーク・イベント)【E1】【E20】【EZ Emulator】」,「2.10.4 変数/I/O レジスタへのアクセスで停止する」参照)
 - その他のブレーク要因の発生
- (2) ブレーク関連のイベントを無視した実行
 デバッグ・ツールバーの デボタンをクリックします。
 この操作により実行を開始した場合、次のいずれかの状態までその実行を続けます。
 - 🔳 ボタンのクリック(「2.10.1 プログラムの実行を手動で停止する」参照)
 - その他のブレーク要因の発生

備考 この操作により実行を開始した場合, Printf イベントの発生も無視されます。

(3) キャレット位置までの実行

エディタ パネル/逆アセンブル パネルにおいて、プログラムを停止させたい行/命令にキャレットを移動した のち、コンテキスト・メニューの[ここまで実行]を選択します。

この操作により実行を開始した場合、次のいずれかの状態までその実行を続けます。

- PC がキャレット位置のアドレスに到達
- 间 ボタンのクリック(「2.10.1 プログラムの実行を手動で停止する」参照)
- その他のブレーク要因の発生
- **注意** キャレット位置の行に対応するアドレスが存在しない場合は、下方向の有効な行までプログラム を実行します(有効な行が存在しない場合は、エラーとなります)。
- 備考 この操作により実行を開始した場合, Printf イベントの発生も無視されます。

2.9.2.3 PC 値を変更してから実行する

カレント PC 値を任意の箇所に強制的に変更したのち、プログラムを実行します。

この操作を行うには、まず、エディタ パネル/逆アセンブル パネルにおいて、プログラムの実行を開始したい行/命令にキャレットを移動したのち、コンテキスト・メニューの [PC をここに設定]を選択します(カレント PC 値が現在 キャレットのある行/命令のアドレスに変更されます)。

次に、「2.9.2.2 現在のアドレスから実行する」で示した、いずれかの実行方法を行います。

注意 ソフトウェア・ブレークポイントおよびハードウェア・ブレーク(実行前ブレーク)ポイントで停止後、再度そのアドレスから実行を再開した場合、一度そのアドレスをステップ実行により実行してから実行を継続するので、リアルタイム性はなくなります。なお、このステップ実行で成立したイベントは、その後の実行時にクリアされます。このため、トレースおよびタイマ計測の開始イベントがこのステップ実行で成立した場合、再度イベントが成立するまでトレース取得もタイマ計測も開始しません。また、ブレークおよびトレース開始組み合わせイベントも同様に、ステップ実行で成立したイベントはAND/シーケンシャルの対象になりません。

2.9.3 プログラムをステップ実行する

次のいずれかの操作を行うと、現在のアドレス(カレント PC 値で示されるアドレス)から、ソース・レベル単位 (ソース・テキスト1行分)、または命令レベル単位(1命令分)でプログラムをステップ実行したのち、自動的に停止 します。

プログラムの停止後は逐一各パネルの内容が自動的に更新されるため、ステップ実行は、プログラムの実行遷移を ソース・レベル単位/命令単位でデバッグする場合に有効な実行方法です。

なお、ステップ実行を行う際の実行単位は、次に示すようにエディタ パネルにおける設定に依存します。

- ツールバーの 🛐 ボタンを無効にしている場合(デフォルト) ソース・レベル単位によるステップ実行を行います。 ただし, 逆アセンブル パネルにフォーカスがある場合, またはカレント PC 値で示されるアドレスに行情報が存在 しない場合は, ___
- ツールバーの 🛐 ボタンを有効にしている場合 命令レベル単位によるステップ実行を行います。

備考 🔰 ボタンは、エディタ パネルを混合表示モードに設定している場合のみ有効となります。

ステップ実行には、次の種類があります。

2.9.3.1 関数内にステップ・インする (ステップ・イン実行)

2.9.3.2 関数をステップ・オーバーする (ステップ・オーバー実行)

2.9.3.3 関数内でリターンが完了するまで実行する(リターン・アウト実行)

- **注意 1.** ステップ実行中は,設定されているブレークポイント/ブレーク・イベント/ Printf イベントを発生しません。
- **注意 2.** 【シミュレータ】 ステップ実行中に割り込みハンドラにジャンプすることがあります。
- 注意 3. ソース・レベル単位でステップ実行した場合、実行しないはずの行を実行しているように見える場合があります。
 これはコンパイラの生成したデバッグ情報と生成コードとのずれによる現象であり、コンパイラの生成したコードの実行結果には問題ありません。
 - 例 以下のようなプログラムを記述した場合,生成コード上は (*1) を実行していませんが, (*2) の後 (*1) の位置ヘカレント PC 位置が移動してしまうように見えます。



```
void main(void);
int x, y, z1, z2, z3;
void func(int i)
    if (i == 0) {
        ++x; // <-(*1)
        ++z1;++z2;++z3;
    } else {
        ++y; // <-(*2)
        ++z1;++z2;++z3;
    }
int one = 1;
void main(void)
{
  while (1)
    func(one);
  }
```

なお、本注意は、ビルド・ツールのプロパティ パネルの [コンパイル・オプション] タブ→ [最適化] カテゴリ内にある [最適化レベル] プロパティを [0 (-optimize=0)] に設定すると回避できる場合があ ります。

2.9.3.1 関数内にステップ・インする (ステップ・イン実行)

関数呼び出しの場合,呼び出された関数内の先頭で停止するステップ実行です。 操作は,デバッグ・ツールバーの 😱 ボタンをクリックします。

- 注意 1. デバッグ情報がない関数へのステップ・イン実行はできません。
- **注意 2.** longjmp 関数へのステップ・イン実行は、実行処理が完了せずタイムアウト待ちになることがあります。
- **注意 3.** 【E1】【E20】【EZ Emulator】 ステップ実行中は、割り込みが禁止されます。また、JTAG 通信において、ステップ実行中はリセット が禁止されます。
- 2.9.3.2 関数をステップ・オーバーする (ステップ・オーバー実行)

サブルーチン分岐命令による関数呼び出しの場合、その関数内のソース行/命令すべてを1ステップとみなして実行し、関数から戻った箇所で停止するステップ実行です(サブルーチン分岐命令を実行したときと同じネストになるまで、 ステップ実行します)。

操作は、デバッグ・ツールバーの

なお,サブルーチン分岐命令以外の場合は, 🔤 ボタンのクリックと同じ動作となります。

- **注意 1.** longjmp 関数のステップ・オーバー実行は、実行処理が完了せずタイムアウト待ちになることがあります。
- **注意 2.**【E1】【E20】【EZ Emulator】 関数やサブルーチンの呼び出し箇所をステップ・オーバーする際は割り込み、およびリセットが許可と なります。

2.9.3.3 関数内でリターンが完了するまで実行する(リターン・アウト実行)

現在の関数から,呼び出し元関数に戻った箇所で停止するステップ実行します。

ある関数内において確認が必要なソース行/命令の実行が終了した際などに、この命令によるステップ実行を行うと、 残りの関数内の命令をステップ実行せずに呼び出し元の関数に戻ることができます。 操作は、デバッグ・ツールバーの

注意 1. main 関数内でのリターン・アウト実行は、スタート・アップ・ルーチン内でブレークします。

- 注意 2. longjmp 関数の呼び出し元関数内でリターン・アウト実行すると,ブレークしないことがあります。
- 再帰関数からリターン・アウト実行を行うと、フリーラン状態となります。 注意 3.
- サブルーチン復帰命令以外の命令(ジャンプ命令)で呼び出し元関数に戻る関数については、正しくリ 注意 4. ターン・アウト実行できません。
- 注意 5. [E1] [E20] [EZ Emulator] 関数内やサブルーチン内でリターン・アウトする際は割り込み、およびリセットが許可となります。
- 注意 6. 再帰呼び出しの関数でリターンアウト実行した場合、呼び出し元ではなく呼び出し先関数のリターン命 令を実行したところで停止する場合があります。

2.9.4 指定ルーチンを実行する【E1】 【E20】 【EZ Emulator】

プログラムの実行開始直前および停止直後にプログラムの指定ルーチンを実行することができます。 プログラムの実行や停止に同期してターゲット・システムの制御を行いたい場合に、任意ルーチンを設定します。 なお、実行中のプログラムの停止方法については、「2.10 プログラムの停止(ブレーク)」を参照してください。 ここでは、実行するルーチンを指定する方法について説明します。

指定ルーチンは、プロパティ パネルの [デバッグ・ツール設定] タブにおける [システム] 【E1】 【E20】 【EZ Emulator】カテゴリ内のプロパティにて指定します。

- プログラムの実行開始直前に指定する場合

[実行開始前に指定ルーチンを実行する]プロパティを[はい]を選択して、この操作で表示される[実行開始前に 実行するルーチン]プロパティにおいて実行開始アドレスを入力してください。 指定されたルーチンがプログラム実行直前に実行されます。

- プログラムの停止直後に指定する場合
- [ブレーク後に指定ルーチンを実行する]プロパティを[はい]を選択して、この操作で表示される[ブレーク後に 実行するルーチン]プロパティにおいて実行開始アドレスを入力してください。 指定されたルーチンがプログラム停止直後に実行されます。

なお、[実行開始前に指定ルーチンを実行する]または[ブレーク後に指定ルーチンを実行する]プロパティのいずれ かで[はい]を選択すると, [指定ルーチン実行機能用ワーク RAM 開始アドレス]プロパティが表示されます。指定し たワーク RAM の配置アドレスから [指定ルーチン実行機能用ワーク RAM サイズ [バイト]] プロパティの表示サイズ分 を,指定ルーチン実行機能用ワーク RAM 領域としてデバッガのファームウェアが使用します。 指定ルーチン実行機能を使用する場合は、このワーク RAM 領域を指定してください。

図 2.105 [システム]カテゴリ

4	システム
	中かな やっぷう

2010	
内蔵プログラムROMを書き換えるプログラムをデバッグする	いいえ
内蔵データフラッシュを書き換えるプログラムをデバッグする	いいえ
実行開始前に指定ルーチンを実行する	はい
実行開始前に実行するルーチン	1000
ブレーク後に指定ルーチンを実行する	はい
ブレーク後に実行するルーチン	
指定ルーチン実行機能用ワークRAM開始アドレス	HEND 1600
指定ルーチン実行機能用ワークRAMサイズ[ノテイト]	560

- 注意 1. [実行開始前に実行するルーチン], [ブレーク後に実行するルーチン]および[指定ルーチン実行機能 用ワーク RAM 開始アドレス]プロパティのプロパティ値には、C 言語の場合は関数名、アセンブリ言 語の場合はラベル名も指定することができます。
- 注意 2. 指定ルーチン内でスタックを使用する場合、割り込みスタックを使用してください。
- 注意 3. 指定ルーチン処理の終了は、リターンサブルーチン命令 (RTS) を記述してください。
- 1回の指定ルーチンの処理時間は 100ms 以内に終了させてください。指定ルーチン内でクロック停止 注意 4. 状態を継続させた場合、デバッグ・ツールの制御ができない可能性があります。
- 注意 5. 指定ルーチン実行開始時のレジスタ値は不定です。そのため、指定ルーチン内でレジスタ値の初期設定 をしてください。
- 注意 6. 指定ルーチンの実行直前および停止直後は、デバッグ・ツールの制御で停止状態となります。なお、停 止時間は次のようになります。
 - ユーザ・プログラム実行開始前の指定ルーチン実行後,約100 サイクル(CPU クロック:100MHz の場合、約1µs)でユーザ・プログラムを実行します。

- ユーザ・プログラム停止後から,約 100 サイクル(CPU クロック:100MHz の場合,約 1µs)で指定ルーチンを実行します。 ただし,内蔵 ROM 無効拡張モード時は、ユーザ・プログラム停止後から、CPU クロック:100MHz

にて約 3ms,CPU クロック:32kHz にて約 8ms で指定ルーチンを実行します。また,ホットプラグ イン接続した場合の,最初のユーザ・プログラム停止時も,同様です。

- **注意 7.** 指定ルーチンは開始時にスーパーバイザモードで実行しますので、ユーザモードに切り替えないでくだ さい。
- **注意 8.** 指定ルーチンを実行する場合,指定ルーチンのプログラム領域へのメモリアクセス/ダウンロード/ブ レークポイント設定を行わないでください。
- 注意 9. 指定ルーチン実行機能使用時は、[指定ルーチン実行機能用ワーク RAM 開始アドレス]から 230h
 (560) バイト分の RAM 領域を、エミュレータ専用に使用します。
 この領域を、ユーザ・プログラムおよび指定ルーチン内で使用しないでください。
 この領域は、内蔵 RAM 領域に配置してください。
 なお、[指定ルーチン実行機能用ワーク RAM 開始アドレス]は、4 で割り切れる値にしてください。
- 注意 10. 指定ルーチンで使用する汎用レジスタおよびフラグについては、次の制限があります。

ISP レジスタ	指定ルーチン終了時に、指定ルーチン実行開始時の値に戻してください。
U フラグ	指定ルーチンを実行中はユーザスタックポインタ(USP)の指定は禁止と なります。
Iフラグ	指定ルーチンを実行中は割り込み禁止となります。
PM フラグ	指定ルーチンを実行中はユーザモードへの切り替え禁止となります。

- **注意 11.** 指定ルーチン実行時は、トレース/ブレーク/リアルタイム RAM モニタ/タイマ計測は無効となります。
- 注意 12. 指定ルーチン実行時は、ノンマスカブル割り込みを常に禁止します。
- **注意 13.** 指定ルーチン実行後にプログラムを開始する場合、マイクロコントローラの状態は次のようになります。

汎用レジスタ	プログラム停止時の状態またはユーザがで設定したレジスタ値となりま す。指定ルーチン実行後のレジスタ内容は反映されません。
メモリ	指定ルーチン実行後のメモリアクセスが反映されます。
周辺機能	指定ルーチン実行後のマイクロコントローラ周辺機能動作が継続します。

- **注意 14.** 指定ルーチン内でスタックを使用する場合は,指定ルーチン内で ISP を設定してください。 なお,指定ルーチン開始時の ISP 値は,[指定ルーチン実行機能用ワーク RAM 開始アドレス]に設定 されます。
- **注意 15.** ISP について,指定ルーチン終了時に, ISP 値を指定ルーチン実行開始時の値に戻してください。
- **注意 16.** 指定ルーチン実行機能を使用時は,端子リセットやウォッチドッグリセットをしないでください。
- 注意 17. 指定ルーチン実行機能を使用時は、内蔵 RAM 領域を無効に設定しないでください。
- **注意 18.** 指定ルーチンを実行中は、U フラグに "1(ユーザスタックポインタ(USP)を指定)" を設定しないでください。
- 注意 19. 指定ルーチン実行中は、RTFI(高速割り込みからの復帰)命令を使用しないでください。
- 注意 20. 指定ルーチン内で以下が発生しないようにしてください。 [ブレークポイント、ブレークイベント、トレース開始イベント/トレース終了イベント] 例えば、実行開始前に実行するルーチン内でブレークイベントが発生すると、プログラム実行開始直後 にプログラムが停止します。また、指定ルーチン内でトレース開始イベントが発生すると、指定ルーチ ン実行中のトレースは取得されませんが、その後のプログラム実行開始からトレースが取得されること になり、期待するエミュレータ動作にならない場合があります。
- 注意 21. 指定ルーチン実行機能にて Start ファンクションからユーザプログラムへ移行しない場合、および Stop ファンクションが停止しない場合は、"START/STOP 時の指定ルーチン実行処理でタイムアウトエラー が発生しました。システムリセットを発行しました。"とエラーメッセージが表示され、エミュレータ を初期化してユーザ・プログラムが停止します。システムリセット発行後は、トレース記録も初期化し ます。なお、デバッグを継続して行うことが可能です。

RENESAS

2.10 プログラムの停止(ブレーク)

この節では、実行中のプログラムを停止する方法について説明します。

備考 実行中のプログラムが停止すると、その原因(ブレーク要因)がメイン・ウインドウのステータス・ バーに表示されます。

2.10.1 プログラムの実行を手動で停止する

デバッグ・ツールバーの 🔳 ボタンをクリックすることにより、現在実行中のプログラムを強制的に停止します。

2.10.2 任意の場所で停止する(ブレークポイント)

ブレークポイントを設定することにより,任意の箇所でプログラムの実行を容易に停止させることができます。 ブレークポイントは,マウスのワン・クリックで設定することができます。

ブレークポイントを設定するためには、あらかじめ使用するブレークポイントの種別、および動作の設定を行う必要 があります。

ここでは、次の操作方法について説明します。

2.10.2.1 使用するブレークポイントの種別/動作を設定する【E1】【E20】【EZ Emulator】
 2.10.2.2 ブレークポイントを設定する
 2.10.2.3 ハードウエア・ブレークポイントを編集する
 2.10.2.4 ブレークポイントを削除する

注意 【E1/E20/EZ Emulator【RX630, RX631, RX63N, RX63T, RX210, RX110, RX111, RX113, RX634, RX130, RX13T グループ】] WAIT 命令またはその1命令前に設定した実行系イベントでブレークが発生した場合,エラーが発生する可能性があります。ターゲットプログラムをWAIT 命令またはその1命令前でブレークさせる場合は、ブレークポイントを使用してください。

2.10.2.1 使用するブレークポイントの種別/動作を設定する【E1】【E20】【EZ Emulator】

使用するブレークポイントの種別/動作の設定は、プロパティパネルの[デバッグ・ツール設定]タブ上の[ブレーク]【E1】【E20】【EZ Emulator】カテゴリ内で行います。

図 2.106 [ブレーク] カテゴリ【E1】 【E20】 【EZ Emulator】

▲ ブレーク

🧲 優先的に使用するブレークポイントの種類 📂 ハードウエア・ブレーク

[優先的に使用するブレークポイントの種類] プロパティで、マウスのワンクリック操作で設定するブレークポイントの種別を指定します。

ただし,指定した種類のブレークポイントの設定数が制限を越える場合(「2.17.7.1 有効イベント数の制限」参照), もう一方の種類のブレークポイントが使用されます。

ブレークポイントの用途に合わせて、次のドロップダウン・リストから選択します。

ソフトウエア・ブレーク	指定したアドレスの命令コードを一時的にブレーク用の命令に書き換え,その命令を 実行した際にプログラムを停止させます。 設定すると,ソフトウエア・ブレーク・イベントとして扱われます。
ハードウエア・ブレーク	デバッグ・ツールが, プログラム実行中にブレーク条件を逐次確認し, 条件を満たし た際にプログラムを停止させます(デフォルト) ^注 。 設定すると, ハードウエア・ブレーク・イベントとして扱われます。

注 1. ハードウエア・ブレークは,指定したアドレスの命令実行前にブレークする"実行前ブレーク"であり,デバッグ・ツールの資源を用いて実現している機能です。

注 2. 外部アドレス空間へのソフトウエア・ブレーク・イベントは設定できません。

2.10.2.2 ブレークポイントを設定する

操作は,ソース・テキスト/逆アセンブル・テキストを表示しているエディタ パネル/逆アセンブル パネルで行います。

各パネルのメイン・エリア(エディタ パネル)/イベント・エリア(逆アセンブル パネル)において, ブレークポイントを設定したい箇所をクリックしてください。

ブレークポイントは、クリックした行位置に対応する先頭アドレスの命令に対して設定されます。

ブレークポイントが設定されると、設定した箇所に次のイベント・マークが表示され、ソース・テキスト行/逆アセンブル・テキスト行が強調表示されます。

また,対象アドレスにブレーク・イベント(ソフトウエア・ブレーク・イベント/ハードウエア・ブレーク・イベント)が設定されたとみなされ,イベントパネルで管理されます(「2.17 イベントの管理」参照)。

表 2.4 ブレークポイントのイベント・マーク

種別	イベント・マーク
ソフトウエア・ブレーク (【E1】【E20】【EZ Emulator】)	®
ハードウエア・ブレーク	

図 2.107 ブレークポイントの設定例(逆アセンブル パネルの場合)



- **注意 1.** ブレークポイントはブレーク・イベントとして設定され、イベントとして管理されるため、設定数に制限があります。ブレークポイントの設定に関しては(有効イベント数の制限など)、「2.17.7 イベント設定に関する留意事項」も参照してください。
- **注意 2.** ブレークポイントは、アドレス表示がない行に設定することはできません。
- 注意 3. 【E1】【E20】【EZ Emulator】 ソフトウエア・ブレークポイントは、命令を置き換えることにより実現します。そのため、内蔵 ROM と内蔵 RAM 以外の領域には指定することができません。
- 備考 1. イベントの設定状態によりイベント・マークは異なります(「2.17.1 設定状態(有効/無効)を変更する」参照)。 また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが 設定されていることを示すイベント・マーク(FP)が表示されます。
- 備考 2. 【シミュレータ】 設定できるブレークポイントは, ハードウエア・ブレークポイント固定です。
- 備考 3. 【E1】【E20】【EZ Emulator】 次に示す操作により、「2.10.2.1 使用するブレークポイントの種別/動作を設定する【E1】【E20】【EZ Emulator】」の指定に依存することなく、ハードウエア・ブレークポイント/ソフトウエア・ブレーク ポイントを設定することができます。

種別	操作方法 1	操作方法 2
ハードウエア・ブレーク ポイント	[Ctrl] キー + クリック	コンテキスト・メニューの[ブレークの設 定]→[ハード・ブレークを設定]を選択
ソフトウエア・ブレーク ポイント	[Shift] キー + クリック	コンテキスト・メニューの[ブレークの設 定]→[ソフト・ブレークを設定]を選択

注意

" 操作方法 1" は,逆アセンブル パネルでのみ有効です。

RENESAS

2.10.2.3 ハードウエア・ブレークポイントを編集する

設定したハードウエア・ブレークポイントを編集するには、[表示]メニュー→ [イベント] の選択でオープンするイ ベント パネル上において、編集したいハードウエア・ブレークポイントを選択したのち、コンテキスト・メニューの 「条件の編集…」をクリックしてください。

この操作によりオープンするダイアログで編集します。

ダイアログによる編集の詳細については、「2.17.4.1 実行系イベントを編集する」を参照してください。

2.10.2.4 ブレークポイントを削除する

設定したブレークポイントを削除するには、エディタパネル/逆アセンブルパネル上において、表示されているイベント・マークを再度クリックします(イベント・マークが消失します)。

また, [表示] メニュー→ [イベント] の選択でオープンする イベント パネル上において, 削除したいブレークポイントを選択したのち, ツールバーの ボタンをクリックする操作でも削除することができます。(「2.17.5 イベントを削除する」参照)。

2.10.3 任意の場所で停止する(ブレーク・イベント) 【E1】 【E20】 【EZ Emulator】

ブレーク・イベント(実行系)を設定することにより、任意の箇所でプログラムの実行を停止させることができます。 ブレーク・イベント(実行系)は、指定したアドレスの命令実行後にブレークする"実行後ブレーク"であり、デバッ グ・ツールの資源を用いて実現している機能です。 ここでは、次の操作方法について説明します。

ここでは、火の操作方法について説明します。

2.10.3.1 ブレーク・イベント(実行系)を設定する 2.10.3.2 ブレーク・イベント(実行系)を編集する 2.10.3.3 ブレーク・イベント(実行系)を削除する

2.10.3.1 ブレーク・イベント(実行系)を設定する

操作は,ソース・テキスト/逆アセンブル・テキストを表示しているエディタ パネル/逆アセンブル パネルで行います。

設定したい行にキャレットを移動したのち、コンテキスト・メニューの「ブレークの設定」→「組み合わせブレークを設定」を選択します。

ブレーク・イベント(実行系)は、選択した行位置に対応する先頭アドレスの命令に対して設定されます。

ブレーク・イベント(実行系)が設定されると、設定した箇所にハードウエア・ブレークポイントと同じイベント・マークが表示され、逆アセンブル・テキスト行が強調表示されます(「2.10.2.2 ブレークポイントを設定する」参照)。 また、対象アドレスにブレーク・イベント(実行系)が設定されたとみなされ、イベントパネルで管理されます

(「2.17 イベントの管理」参照)。イベント パネルにおいて、組み合わせブレークの詳細情報内に実行後として登録されます。

注意 ブレーク・イベント(実行系)の設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する留意事項」も参照してください。



- 逆アセンブル1 × 06890104 a009 ADD 10H[R0],R1 R1,04H[R0] NOV.L f85e4 ffff85e8 for(j=i-gap; j>=k; j=j-gap)[38: D4H(RO) .RI TOH(RO) .RI NOV-L ffff85e8 ff85ea 06810104 SUB FF 85 fff85f0 201 💟 ウオッチ1 に登録(R) + 61 37 =]; 1404H_R14 ff85f2 アクション・イベントの登録(A)... fbe: 📢 11118518 e10 ここまで実行(H) [j+gap]){ 381 PCをここに設定(P) .82 . R14 fieß 068 5 移動(G)... ff860 ,R1 ,R15 Ctrl+G ff8608 471 読 シンボルへ移動(Y) 18609 +75H FFFF880b 1 シンボル へ戻る(B) 4 ブレークの設定(8) 💌 ハード・ブレークの設定(H) トレース設定(T) ・ブレークの設定(5) ٠ タイマ設定(1) ▶ 🐘 組み合わせブレークを設定(C) 🛐 命令の編集(D) 読み込み組み合わせブレークを設定(E) コードの編集(C) 書き込み組み合わせブレークを設定(T) A 🚮 読み書き組み合わせブレークを設定(M) 表示(1) ・ ブレーク動作の設定(0) 混合表示(M) 51 ソースヘジャンズト
- 図 2.108 逆アセンブル・テキスト上の行に対するブレーク・イベント(実行系)の設定例【E1】【E20】【EZ Emulator】

図 2.109 イベントパネル上におけるブレーク・イベント(実行系)の設定例【E1】【E20】【EZ Emulator】

1/2/1			×
× 🖲 🖲 🗑 🗑	2 5 8 8	1	
名前	腰骨間線 コー		コメント
🔽 警 無条件トレース		-	
😑 📝 髱 組み合わせブレー	-ク OR		
言羊糸田作青市服			
実行後 sortc#36 ()×ffff85f0		
名前	疑者相称語 マ		
👿 警 Run-Breakタイマ	総実行時間	:722300000 ns	

2.10.3.2 ブレーク・イベント(実行系)を編集する

設定したブレーク・イベント(実行系)を編集するには、[表示]メニュー→[イベント]の選択でオープンするイベ ントパネル上において、組み合わせブレークの詳細情報に表示されたブレーク・イベント(実行系)を選択したのち、 コンテキスト・メニューの「条件の編集…」をクリックしてください。 この操作によりオープンするダイアログで編集します。 ダイアログによる編集の詳細については、「2.17.4.1 実行系イベントを編集する」を参照してください。

2.10.3.3 ブレーク・イベント(実行系)を削除する

設定したブレーク・イベント(実行系)を削除するには、[表示]メニュー→ [イベント]の選択でオープンする イベント パネル上において、組み合わせブレークの詳細情報に表示されたブレーク・イベント(実行系)を選択したのち、ツールバーの ズボタンをクリックする操作で削除することができます。(「2.17.5 イベントを削除する」参照)。

また、ソース・テキスト/逆アセンブラ・テキスト上において、表示されているイベント・マークをクリックするこ とでブレーク・イベント(実行系)を削除することもできます。

2.10.4 変数 /I/O レジスタへのアクセスで停止する

ブレーク・イベント(アクセス系)を設定することにより、任意の変数、または I/O レジスタに対し、指定したアク セスがあった場合にプログラムの実行を停止させることができます。 また、この際に、アクセスした値を限定することもできます。 アクセス系のブレーク・イベントで指定できるアクセス種別は次のとおりです。

表 2.5 変数へのアク	1セス種別
--------------	-------

アクセス種別	説明
リード	指定した変数 ///O レジスタに,リード・アクセスした(読み込みを行った)際に実行 中のプログラムを停止します。
ライト	指定した変数 /l/O レジスタに,ライト・アクセスした(書き込みを行った)際に実行 中のプログラムを停止します。
リード/ライト	指定した変数 /l/O レジスタに,リード・アクセス,またはライト・アクセスした(読 み込み,または書き込みを行った)際に実行中のプログラムを停止します。

注意 1. DMAC(Direct Memory Access Controller)/DTC(Data Transfer Controller)によるアクセスは対象 となりません。

注意 2. 【シミュレータ】 ストリング操作命令、および積和演算命令は、最初と最後のデータアクセスがイベントチェック対象となります。

ここでは、次の操作方法について説明します。

2.10.4.1 変数 ///O レジスタへのブレーク・イベント(アクセス系)を設定する 2.10.4.2 変数 ///O レジスタへのブレーク・イベント(アクセス系)を編集する 2.10.4.3 変数 ///O レジスタへのブレーク・イベント(アクセス系)を削除する

2.10.4.1 変数 /I/O レジスタへのブレーク・イベント(アクセス系)を設定する

変数, または I/O レジスタへのアクセスで、プログラムの実行を停止させるブレーク・イベント(アクセス系)の設定は、次のいずれかの操作により行います。

- **注意 1.** ブレーク・イベント(アクセス系)の設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する留意事項」も参照してください。
- **注意 2.** 【E1】【E20】【EZ Emulator】 ブレーク・イベント(アクセス系)は、イベント パネルにおいて組み合わせブレークの詳細情報内に リード,ライト,リード/ライトとして登録されます。
- (1) ソース・テキスト/逆アセンブル・テキスト上の変数 /I/O レジスタにブレーク・イベント(アクセス系)を設定する場合

操作は, ソース・テキスト/逆アセンブル・テキストを表示しているエディタ パネル/逆アセンブル パネル上 で行います。

ソース・テキスト/逆アセンブル・テキスト上の任意の変数,または I/O レジスタを選択したのち,コンテキスト・メニューより次の操作を行います。ただし,対象となる変数は,グローバル変数/関数内スタティック変数 /ファイル内スタティック変数のみとなります。

なお,この操作を行うことにより,対象変数,または I/O レジスタにブレーク・イベント(アクセス系)が設定 されたとみなされ,イベントパネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法
リード	-【E1】【E20】【EZ Emulator】 [ブレークの設定]→[読み込み組み合わせブレークを設定]を選択したのち, [Enter]キーを押下します。
	-【シミュレータ】 [ブレークの設定]→[読み込みブレークを設定]を選択したのち,[Enter]キーを 押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合,指定した値で読み込 みを行った場合のみブレークします。値を指定しない場合は,値にかかわらず,選択し ている変数に読み込みを行った場合にブレークします。
ライト	-【E1】【E20】【EZ Emulator】 [ブレークの設定]→[書き込み組み合わせブレークを設定]を選択したのち, [Enter]キーを押下します。
	-【シミュレータ】 [ブレークの設定]→[書き込みブレークを設定]を選択したのち,[Enter]キーを 押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合,指定した値で書き込 みを行った場合のみブレークします。値を指定しない場合は,値にかかわらず,選択し ている変数に書き込みを行った場合にブレークします。
リード/ライト	-【E1】【E20】【EZ Emulator】 [ブレークの設定]→[読み書き組み合わせブレークを設定]を選択したのち, [Enter]キーを押下します。
	-【シミュレータ】 [ブレークの設定] → [読み書きブレークを設定]を選択したのち, [Enter] キーを 押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合,指定した値で読み込 み、または書き込みを行った場合のみブレークします。 値を指定しない場合は,値にかかわらず,選択している変数に読み込み,または書き込 みを行った場合にブレークします。

注意 1. カレント・スコープ内の変数が対象となります。

- **注意 2.** ブレーク・イベントは、アドレス表示がない行上の変数 I/O レジスタを選択しても設定することはできません。
- 図 2.110 ソース・テキスト上の変数に対するブレーク・イベント(アクセス系)の設定例【E1】【E20】 【EZ Emulator】



図 2.111 イベントパネル上におけるブレーク・イベント(アクセス系)の設定例【E1】【E20】【EZ Emulator】

1101	×
×®®®®3₽₽ ≤5®	
名前 マ 詳細情報	コメント
📝 🍄 無条件トレース -	
回 図 100 組み合わせブレーク OR	
言羊糸田作青華服	
ライト c_IntBuf 0×1404 - 0×1407 == 0xb	
名前 🗸 詳細情報	コメント
📝 警 Run-Breakタイマ 総実行時間: 722300000 ns	



図 2.113 イベントパネル上におけるブレーク・イベント(アクセス系)の設定例【シミュレータ】



(2) 登録したウォッチ式にブレーク・イベント(アクセス系)を設定する場合 操作は、ウォッチパネル上で行います。 対象となるウォッチ式を選択したのち(複数選択不可)、コンテキスト・メニューより次の操作を行います。

RENESAS

ただし、対象となるウォッチ式は、グローバル変数/関数内スタティック変数/ファイル内スタティック変数 / I/O レジスタのみとなります。

なお、この操作を行うことにより、対象ウォッチ式にブレーク・イベント(アクセス系)が設定されたとみなされ、イベントパネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法
リード	-【E1】【E20】【EZ Emulator】 [アクセス・ブレークの設定]→[読み込み組み合わせブレークを設定]を選択した のち,[Enter]キーを押下します。
	-【シミュレータ】 [アクセス・ブレークの設定]→[読み込みブレークを設定]を選択したのち, [Enter] キーを押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合、指定された値で読み 込みを行った場合のみブレークします。値を指定しない場合は、値に関わらず、選択し ているウォッチ式に読み込みを行った場合にブレークします。
ライト	-【E1】【E20】【EZ Emulator】 [アクセス・ブレークの設定]→[書き込み組み合わせブレークを設定]を選択した のち, [Enter] キーを押下します。
	-【シミュレータ】 [アクセス・ブレークの設定]→[書き込みブレークを設定]を選択したのち, [Enter] キーを押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合、指定された値で書き 込みを行った場合のみブレークします。値を指定しない場合は、値に関わらず、選択し ているウォッチ式に書き込みを行った場合にブレークします。
リード/ライト	-【E1】【E20】【EZ Emulator】 [アクセス・ブレークの設定]→[読み書き組み合わせブレークを設定]を選択した のち,[Enter]キーを押下します。
	-【シミュレータ】 [アクセス・ブレークの設定]→[読み書きブレークを設定]を選択したのち, [Enter] キーを押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合,指定された値で読み 込み,または書き込みを行った場合のみブレークします。値を指定しない場合は,値に 関わらず,選択しているウォッチ式に読み書きを行った場合にブレークします。

備考 カレント・スコープ内のウォッチ式が対象となります。 カレント・スコープ外のウォッチ式を対象とする場合は、スコープ指定したウォッチ式を選択し てください。

図 2.114 ウォッチ式に対するブレーク・イベント(アクセス系)の設定例【E1】【E20】【EZ Emulator】

ウォッチ1			8
📵 🤫 🧞 🗓	1	(表記(N)+ 👦	
ウォッチ式 ・ g_IntBuf	> 1	値 聖情報(パイト数) (0x0000007) int(4)	アドレス メモ 0x00011404
		アクセス・ブレークの設定(B) 🔸	ほみ込み組み合わせブレークを設定(E)
		(T)	🔏 書き込み組み合わせブレークを設定(T) 0xb
		タイマ設定(1)	読み書き組み合わせブレークを設定(M)
		リアルタイム表示更新設定い	
	2	最新の情報に更新(R)	ウォッチ式 "g_IntBuf" 上のコンテキスト・メ
	89	値を強制読み込み(F)	[書き込み組み合わせブレークを設定]内に値
L.	~	新規ウォッチエキ	を入力したのち、[Enter] キーを押下します。
			き込まれた場合にブレークする設定をしていま
			す。





図 2.115 ウォッチ式に対するブレーク・イベント(アクセス系)の設定例【シミュレータ】

2.10.4.2 変数 /I/O レジスタへのブレーク・イベント(アクセス系)を編集する

設定したブレーク・イベント(アクセス系)を編集する場合は,[表示]メニュー→ [イベント]の選択でオープンするイベントパネルにおいて,編集したいブレーク・イベント(アクセス系)を選択したのち,コンテキスト・メニューの「条件の編集...」をクリックしてください。

ここでオープンするダイアログで編集します。 ダイアログによる編集の詳細については、「2.17.4.2 アクセス系イベントを編集する」を参照してください。

2.10.4.3 変数 /I/O レジスタへのブレーク・イベント(アクセス系)を削除する

設定したブレーク・イベント(アクセス系)を削除する場合は、[表示]メニュー→ [イベント]の選択でオープンするイベントパネルにおいて次の操作を行います。なお、使用するデバッグ・ツールにより、イベント名は異なります。

- (1) E1/E20/EZ Emulator の場合 削除したい組み合わせブレークの詳細情報に表示されたブレーク・イベント(アクセス系)を選択したのち、同 パネルのツールバーのズボタンをクリックします(「2.17.5 イベントを削除する」参照)。 また、逆アセンブラ・テキスト上において、表示されているイベント・マークをクリックすることでブレーク・ イベント(アクセス系)を削除することもできます。
- (2) シミュレータの場合
 削除したいアクセス系のハードウェア・ブレークを選択したのち、同パネルのツールバーのズボタンをクリックします(「2.17.5 イベントを削除する」参照)。
- 2.10.5 複数のブレーク・イベントを組み合わせる(組み合わせブレーク)【E1】 【E20】【EZ Emulator】

2つ以上のブレーク・イベント(実行系およびアクセス系)を組み合わせて設定することで、設定されたブレーク・ イベントが組み合わせ条件を満たした時点でプログラムの実行を停止させることができます。

設定可能な組み合わせブレークは1つのみです。そのため、2つ以上のブレーク・イベントを設定した場合、イベントパネル上では1つの組み合わせブレークの詳細情報内にブレーク・イベントが順に追加されます。 組み合わせブレークでは、次の組み合わせ条件が指定できます。

表 2.6	組み合わせブレークに対す	る組み合わせ条件
-------	--------------	----------

組み合わせ条件	説明
OR	設定されたブレーク・イベントの内, いずれかひとつでも条件が成立した時点でブレークし ます。

組み合わせ条件	説明
AND	時間軸に関係なく,設定された全てのブレーク・イベントの条件が成立した時点でブレーク します。
シーケンシャル	指定した順序で設定されたブレーク・イベントの条件が成立するとブレークします。 また、組み合わせブレーク内のブレーク・イベントを1点のみリセットイベント(Rイベン ト)として登録することができます。登録されたブレーク・イベントが成立した場合、その 時点まで成立していた他のブレーク・イベントの条件成立は全てクリアされます。

なお、組み合わせブレークを編集する場合、イベントパネルにおいて組み合わせブレークを選択したのち、コンテキ スト・メニューの「条件の編集…」をクリックしてください。 ここでオープンするダイアログで編集します。

ダイアログによる編集の詳細については、「2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】」を参照してください。

- **注意 1.** 組み合わせブレークの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する 留意事項」も参照してください。
- 注意 2. イベントの組み合わせに関しては、「(1) 組み合わせ条件の編集」の注意も参照してください。
- 図 2.116 イベントパネル上における組み合わせブレーク(シーケンシャル)の設定例【E1】【E20】【EZ Emulator】

1701			×
× ® ® ® 8 9 9 9	3 5 6 6		
名前	部分對清華版	マーコメント	
🔽 警 Run-Break 🖓 🖓 💎	総実行時間:1580	500000 ns	
😑 📝 鼈 組み合わせブレーク	シーケンシャル		=
詳細情報			
実行後 sort.c#36 0xffff	185e8		
ライト g_IntBuf 0×1404 -	- 0x1407 == 0xb		
リード g_CharBuf 0x100	0 - 0x1000 == 0x55		
実行後 sort.c#35 0xffff	185de		J
•			P.

2.10.6 その他のブレーク要因

上記のほか、プログラムの実行が停止する原因(ブレーク要因)には次のものがあります。 なお、ブレーク要因は、プログラム停止時に、メイン・ウインドウのステータス・バーで確認することができます。

表 2.7 その他のブレーク	ク要因
----------------	-----

要因	使用するデバッグ・ツール		
	E1(Serial)/E1(JTAG)/ E20(Serial)/E20(JTAG)/EZ Emulator	E2	シミュレータ
トレース・メモリを使い切った ^{注 1}	0	0	0
テンポラリ・ブレークの発生	0	0	0
実行の失敗、または不明な原因	0	0	_
WAIT 命令の実行	—	-	0
未定義命令例外の発生	—	_	O ^{注2}
特権命令例外の発生	_	_	O ^{注2}


要因	使用するデバッグ・ツール			
	E1(Serial)/E1(JTAG)/ E20(Serial)/E20(JTAG)/EZ Emulator	E2	シミュレータ	
アクセス例外の発生	_		O ^{注2}	
浮動小数点例外の発生	-	_	O ^{注 2, 3}	
割り込みの発生	—	-	O ^{注2}	
INT 命令例外の発生	_		O ^{注 2}	
BRK 命令例外の発生	—	-	O ^{注2}	
周辺機能シミュレーションによるエラーの発生	—		0	
不正なメモリ・アクセス	-	_	O ^{注 2}	
ストリーム入出力エラーの発生	—	-	0	
カバレッジ・メモリの確保に失敗	—	-	0	
トレース・メモリの確保に失敗	—	-	0	
E2 拡張機能	-	0	—	
記録メモリを使い切った	-	0	—	

注 1. プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内 [トレース・メモリ を使い切った後の動作] プロパティの設定に依存

- 注 2. 【シミュレータ】 プロパティ パネルの [デバッグ・ツール設定] タブ上の [実行モード]【シミュレータ】カテゴリ内各 プロパティの設定に依存
- 注 3. 【シミュレータ】 単精度浮動小数点による例外が対象



2.11 メモリ、レジスタ、変数の表示/変更

この節では、メモリ、レジスタ、および変数の内容を表示/変更する方法について説明します。

2.11.1 メモリを表示/変更する

メモリの内容の表示,および値の変更は,次のメモリパネルで行います。 [表示] メニュー→ [メモリ] → [メモリ 1 ~ 4]を選択してください。 メモリパネルは,最大4個までオープンすることができ,各パネルはタイトルバーの"メモリ 1","メモリ 2","メモ リ3","メモリ 4"の名称で識別されます。

なお,各エリアの見方,および機能についての詳細は、メモリパネルの項を参照してください。



ここでは、次の操作方法について説明します。

2.11.1.1 表示位置を指定する
2.11.1.2 値の表示形式を変更する
2.11.1.3 メモリの内容を変更する
2.11.1.4 プログラム実行中にメモリの内容を表示/変更する
2.11.1.5 メモリの内容を検索する
2.11.1.6 メモリの内容を一括して変更(初期化)する
2.11.1.7 メモリの表示内容を保存する

2.11.1.1 表示位置を指定する

表示位置指定エリアにアドレス式を指定することにより、メモリ値の表示開始位置を指定することができます(デフォルトでは, 0x0番地より表示を開始します)。

備考 コンテキスト・メニューの [表示アドレス・オフセット値を設定 …]を選択することでオープンするア ドレス・オフセット設定 ダイアログにより、メモリ値の表示開始アドレスにオフセット値を設定する ことができます。

図 2.118 表示位置指定エリア(メモリ パネル)

🔄 停止時に移動	移動
	4

(1) アドレス式の指定 表示したいメモリ値のアドレスとなるアドレス式をテキスト・ボックスに直接入力します。最大 1024 文字まで の入力式を指定することができ、その計算結果を表示開始位置アドレスとして扱います。

- ただし、マイクロコントローラのアドレス空間よりも大きいアドレス式を指定することはできません。
- 備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- 備考 2. 指定したアドレス式がシンボルを表現し、サイズが判明する場合では、そのシンボルの先頭アド レスから終了アドレスまで選択状態で表示します。
- (2) アドレス式の自動/手動評価の指定

表示開始位置を変更するタイミングは、[停止時に移動]チェック・ボックスの指定、および [移動] ボタンに より決定します。

[停止時に移動]	1	プログラム停止後, 自動的にアドレス式の評価を行い, その計算結果のアドレスに キャレットが移動します。
		プログラム停止後, アドレス式の評価を自動的に行いません。 この場合, [移動] ボタンをクリックすることにより, アドレス式の評価を行いま す。
[移動]		[停止時に移動] チェック・ボックスのチェックをしなかった場合, このボタンをク リックすることによりアドレス式の評価を行い, その計算結果のアドレスにキャ レットが移動します。

2.11.1.2 値の表示形式を変更する

ツールバーの次のボタンにより、このパネルのアドレス・エリア/メモリ値エリア/文字列エリアの表示形式を変更 することができます。

表記	メモリ値の表示形式を変更する次のボタンを表示します。
🔙 16 進数	メモリ値を 16 進数で表示します(デフォルト)。
🔜 符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
🔛 符号無し 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を8進数で表示します。
1 進数	メモリ値を2進数で表示します。
サイズ表記	メモリ値のサイズの表示形式を変更する次のボタンを表示します。
4 ビット	メモリ値を4ビット幅で表示します。
1バイト	メモリ値を8ビット幅で表示します(デフォルト)。
1182バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
114 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8 バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
エンコード	文字列のエンコードを変更する次のボタンを表示します。



	ASCII	文字列を ASCII コードで表示します(デフォルト)。
	Shift_JIS	文字列を Shift_JIS コードで表示します。
	EUC-JP	文字列を EUC-JP コードで表示します。
	UTF-8	文字列を UTF-8 コードで表示します。
ſ	UTF-16	文字列を UTF-16 コードで表示します。
	Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
	Float	文字列を単精度浮動小数点数値 ^注 で表示します。
	Double	文字列を倍精度浮動小数点数値で表示します。
	Float Complex	文字列を単精度浮動小数点数の複素数で表示します。
	Double Complex	文字列を倍精度浮動小数点数の複素数で表示します。
	Float Imaginary	文字列を単精度浮動小数点数の虚数で表示します。
	Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表	 示	表示形式を変更する次のボタンを表示します。
	注 スクロール範囲を設 定 …	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
	表示桁数を設定	メモリ値エリアの表示桁数を設定するため,表示桁数設定 ダイアログがオープンし ます。
	表示アドレス・オフセッ ト値を設定	アドレス・エリアに表示するアドレスのオフセット値を設定するため、アドレス・オフセット設定 ダイアログをオープンします。

注 浮動小数点数値表示についての詳細は、メモリパネルの項を参照してください。

2.11.1.3 メモリの内容を変更する

メモリの値は編集することができます。

メモリ値エリア/文字列エリアにおいて、対象メモリ値にキャレットを移動したのち、直接キーボードより編集しま す。

メモリ値を編集すると変更箇所の表示色が変化し、この状態で[Enter]キーを押下することにより、変更した値が ターゲット・メモリに書き込まれます([Enter]キーの押下前に[Esc]キーを押下すると編集をキャンセルします)。 ただし、変更の際に入力可能な文字列は、現在指定されている表示進数で扱うことができる文字列に限ります。また、

アンドレージャンクロードなく子列は、現在指定されている表示進致で扱うことができる文子列に限ります。ま 文字列エリアでの変更は、文字コードとして "ASCII" が指定されている場合のみ可能です。

なお,メモリの値の編集は、プログラム実行中の状態でも行うことができます。設定方法についての詳細は, 「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

値を変更する際において、留意する必要がある例を次に示します。

- 例 1. 表示ビット幅の最大値を越えた場合
- 10 進数 8 ビット表示において,表示値 "105" の "1" を編集して "3" を入力した場合,変更値は最大値で ある "127" となります。
- 例 2. 数値の途中に "-" を入力した場合 符号あり 10 進数 16 ビット表示において,表示値 "32768" を "32-68" と編集した場合, "3" と "2" が空 白に変わり,変更値は "-68" となります。
- 例 3. 数値の途中に空白記号(スペース)を入力した場合 10 進数 16 ビット表示において、表示値 "32767" を "32 67" と編集した場合、 "3" と "2" が空白に変わ り、変更値は "67" となります。
- 例 4. 同一の値を入力した場合 現在のメモリ値と同一の値を指定した場合でも,指定した値をメモリに書き込みます。

2.11.1.4 プログラム実行中にメモリの内容を表示/変更する

メモリ パネル/ウォッチ パネルでは、プログラムの実行中に、リアルタイムにメモリ/ウォッチ式の内容を表示更新、および書き換えることができるリアルタイム表示更新機能を備えています。

- このリアルタイム表示更新機能を有効化することにより、プログラムが停止している状態時だけでなく、実行中の状態であっても、メモリ/ウォッチ式の値の表示/変更を行うことができます。
 - 注意 1. 【E20(JTAG)【RX600 シリーズ】】 トレース機能/リアルタイム RAM モニタ機能(RRM 機能)は一部排他使用の機能です。このため、 以降の設定を行う前に、優先的に使用する機能を指定することを目的として、プロパティ パネルの [デバッグ・ツール設定] タブにおいて次の設定を完了させておく必要があります。
 - [トレース] カテゴリ→ [トレース機能の用途] プロパティ→ [リアルタイム RAM モニタ]
 - **注意 2.** 【E20(JTAG)【RX600 シリーズ】】 プログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合、それ以 降のリアルタイム RAM モニタの動作は保証できません。(正しい値を表示できない可能性がありま す。)
 - 注意 3. 【E20】
 リアルタイム RAM モニタ機能を使用する場合は、E20 とターゲット・ボードを 38 ピンで接続している必要があります。
 14 ピンで接続している場合は、リアルタイム RAM モニタ機能は使用できません。
 - 備考 リアルタイム表示更新機能は、デバッグ・ツールの RRM(Real-time RAM Monitor)機能,擬似 RRM 機能,DMM(Dynamic Memory Modification)機能により実現されます。 擬似 RRM 機能では、プログラムの実行を一瞬ブレークさせ、ソフトウエア・エミュレーションによる 読み込み/書き込みを行います。

リアルタイム表示更新機能を使用した読み込み/書き込みが可能となる対象領域は、使用するデバッグ・ツールとプロパティ パネルの設定内容により異なります。

次に示すプロパティ パネルの [デバッグ・ツール設定] タブ上の [実行中のメモリ・アクセス] カテゴリ,および [トレース] カテゴリ内の各プロパティの設定内容に基づき,リアルタイム表示更新機能の使用目的に応じ,表 2.8/表 2.9/表 2.10 の設定を行ってください。

設定記号	[トレース] カテゴリ内の プロパティ	設定値	備考
A	トレース機能の用途	リアルタイム RAM モニタ	E20(JTAG)【RX600, RX700 シ リーズ】

設定記号	[実行中のメモリ・アクセス] カ テゴリ内の プロパティ	設定値	備考
В	実行を一瞬停止してアクセスする 【E1】【E20】【EZ Emulator】	[いいえ](デフォルト)	RRM/DMM 機能の有効化
	リアルタイム表示更新を自動設定 する	[はい](デフォルト)	E20(JTAG)【RX600, RX700 シ リーズ】
С	実行を一瞬停止してアクセスする 【E1】【E20】【EZ Emulator】	[はい]	擬似 RRM 機能の有効化
D	実行中に表示更新を行う	[はい](デフォルト)	
	表示更新間隔 [ms]	[100 ~65500 の <u>整数</u>]	



(1) 読み込みの場合

表 2.8 リアルタイム表示更新機能(擬似 RRM 機能による読み込み)の対象領域

領域名	E1(Serial)/ E1(JTAG)/E20(Serial)/ E20(JTAG)/EZ Emulator		
	設定	対象	
内蔵 ROM	C+D	全領域	
内蔵 RAM			
データフラッシュ			
ターゲット・メモリ			
エミュレーション・メモリ	_		
CPU レジスタ	不可		
1/0 レジスタ	C+D	全領域	

表 2.9 リアルタイム表示更新機能(RRM 機能による読み込み)の対象領域

領域名	E20(JTAG)【RX6 ス	00, RX700 シリー 、】	シミュレータ	
	設定	対象	設定	対象
内蔵 ROM	A+B+D	自動設定領域 ^注	D	全領域
内蔵 RAM				
データフラッシュ			-	_
ターゲット・メモリ				
エミュレーション・メモリ	_		D	全領域
CPU レジスタ	不可		-	_
I/O レジスタ	A+B+D	自動設定領域 ^注	D	全領域

注 【E20(JTAG)【RX600, RX700 シリーズ】】

RRM 機能の対象領域は,最大4096 バイト(最大1024 バイトまでの領域を4つ)に限定されます。そのため, CS+では,次に示す決定規則(優先順位)に従い,上記制限内によるリアルタイム表示更新の対象を自動的に決定します(プログラム実行直前において,前面に表示しているパネルのみが対象となります)。

(1) ウォッチ パネルに表示しているウォッチ式を上から順に設定(複数のウォッチ パネルを オープンしている場合,パネルの番号の小さい順)

(2) メモリ パネルに表示しているメモリをアドレスの小さい順に設定(複数のメモリ パネルを オープンしている場合,パネルの番号の小さい順)

なお、対象領域にリード禁止領域が含まれている場合は、その領域の表示更新は行いません。

(2) 書き込みの場合

表 2.10 リアルタイム表示更新機能(DMM 機能による書き込み)の対象領域

領域名	E1(Serial)/ E1(JTAG)/ E20(Serial)/ E20(JTAG)/EZ Emulator		E20(JTAG)【RX600, RX700 シリーズ】		シミュレータ	
	設定	対象	設定	対象	設定	対象
内蔵 ROM	不可		不可		D	全領域
内蔵 RAM	C+D	全領域	A+B+D	自動設定 領域 ^注		



領域名	E1(Serial)/ E1(JTAG)/ E20(Serial)/ E20(JTAG)/EZ Emulator		E20(JTAG)【RX600, RX700 シリーズ】		シミュレータ	
	設定	対象	設定	対象	設定	対象
データフラッシュ	不可		不可		-	_
ターゲット・メモリ	C+D	全領域	A+B+D	自動設定 領域 ^注		
エミュレーション・メモリ	_		-	-	D	全領域
CPU レジスタ	不可		不	可	-	-
I/O レジスタ	C+D	全領域	A+B+D	自動設定 領域 ^注	D	全領域

注 【E20(JTAG)【RX600, RX700 シリーズ】】 DMM 機能の対象領域は, RRM 機能の対象領域と同一です。

注意 ローカル変数は、リアルタイム表示更新機能の対象外です。

備考 メモリ パネル/ウォッチ パネルにおける値の書き換え方法についての詳細は、「2.11.1.3 メモリ の内容を変更する」/「2.11.6.6 ウォッチ式の内容を変更する」を参照してください。

擬似 RRM 機能で更新しているメモリ値/ウォッチ式は、メモリ パネルにおいてピンク色で強調表示され、RRM 機能で更新しているメモリ値/ウォッチ式は、アクセス状態に従って次のように強調表示されます(表示の際の文字色/ 背景色は、オプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。

アクセス状態	表示例
リード/フェッチ	00 00 00 00
ライト	00 00 00 00
リードとライト	00 00 00 00
ロスト ^注	00 00 00 00

注

ウォッチ パネルにおけるロスト時のウォッチ式の値は, "?"と表示されます。

図 2.119 擬似 RRM 機能によるメモリ表示の例(メモリ パネル)【E1】【E20】【EZ Emulator】

メモリ1													×
2 🛞	表記()	N)+ (N	. . .	イズ表	記(Z)	•	エン	/コ-	- 14	(<u>C</u>)-	表示	₹(<u>I</u>)-	
📄 停止時	に移動	0x1	000									和動	
00001000 00001010 00001020 00001030 00001040 00001040 00001050	+0 +1 C4 7F 2B FA E7 0E CB 44 20 52 11 23	+2 +2 FF F B4 FI E3 3 16 F 34 F ² E5 E	3 +4 3 89 B AE E 71 A 3F 5 60 7 B9	+5 +1 BD B 3B 51 F5 01 1A D E4 21 40 D	+7 + 0 EE F 0 E9 7 0 EF 0 8 D3 F 7 87 8 0 AA 6	8 +9 F 93 F FA 3 FF 2 7F 3 33 B 1D	+a 82 06 0A 90 82 4D	+b 6A D8 12 66 21 58	+c + F2 F F7 E C5 F FB E 56 E 97 3	d +e 3 E3 4 43 C FD E 55 8 03 2 17	+f CB 5B 0A 1A 07 3D	ASC11 ?.???????????????????? +?????P?.??????? ?D.??.???.??? B47`?`????.??? .#????k.WX?2.= .#?????k.WX?2.=	Â



Г

メモリ1							×
2 🛞	表記(N)	サイン	ズ表記(<u>Z</u>)▼	エンコート	[×] (<u>C</u>)+ 表	示(<u>I</u>)-	
📄 停止時	に移動 0	×1000				有多數力	,
00001000 00001010 00001020 00001030 00001040 00001050 00001060	+0 +1 +2 C4 7F FF 28 FA B4 E7 0E E3 CB 44 16 20 52 34 11 23 E5 32 13 55	+3 +4 +5 F3 89 B FB AE 3 3E 71 F FA 3F 1 F5 60 E E7 B9 4 8C 32 A	5 +6 +7 +8 D B0 EE FF B 50 E9 7F 5 0D EF D3 A D8 D8 F2 4 27 87 93 0 D0 AA 6B 9 E1 0C DC	+9 +a +b +c 93 82 6A F2 FA D6 D8 F7 FF 0A 12 C5 FF 0A 12 C5 FF 0A 22 156 10 4D 58 97 28 02 60 79	+d +e +f F3 E3 CB E4 43 5B FC FD 0A BE 55 1A E8 03 07 32 17 3D 70 48 C3	ASCI1 ?.??????????????????? +??????????????	*

図 2.120 RRM 機能によるメモリ表示の例(メモリ パネル)【シミュレータ】



<u>አ</u> ቲካ1	×
② 參 表記(N) + サイズ表記(Z) + エンコード(C) + 表示(I) +	
停止時に移動 0×1000	移動
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f ASCII 00001000 9C 7B 3E 06 9D 7B 3E 06 9F 7B 3E 06 ?	>.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1 >.1
· [+

2.11.1.5 メモリの内容を検索する

メモリの値の検索は、コンテキスト・メニューの [検索 ...] を選択することによりオープンするメモリ検索 ダイアロ グで行います。検索の際は、メモリ値エリアと文字列エリアのうち、キャレットのあるエリアが対象となります。 このダイアログにおいて、次の手順で操作を行ってください。

図 2.122	メモリ内容の検索	(メモリ検索 ダイアログ)
---------	----------	---------------

メモリ検索	
検索するデータ(<u>D</u>)	
検索する範囲(日)	アドレス範囲を指定する
アドレス(<u>A</u>):	0x0 💌 - 0xfffffff 💌
	前を検索(B) 次を検索(F) キャンセル ヘルプ(H)

注意 1. プログラム実行中に、メモリの内容を検索することはできません。

注意 2. 浮動少数点数値表示している文字列を検索することはできません。

(1) [検索するデータ]の指定 検索するデータを指定します。 テキスト・ボックスに直接入力するか(最大指定バイト数:256 バイト),またはドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10 個)。 検索の対象がメモリ値エリアの場合、そのエリアと同じ表示形式(表示進数/サイズ)でデータを入力する必要 があります。 また、検索の対象が文字列エリアの場合では、検索するデータとして文字列を指定する必要があります。指定し た文字列は、そのエリアで表示しているエンコード形式でデータに変換され検索されます。

なお,このダイアログをオープンする直前にメモリ値を選択していた場合は,デフォルトでその値が表示されます。

(2) [検索する範囲]の指定

検索する範囲を次のドロップダウン・リストより選択します。

アドレス範囲を指定する	[アドレス] で指定するアドレス範囲内で検索を行います。
メモリ・マッピング	選択したメモリ・マッピング範囲内で検索を行います。 このリスト項目は、メモリ・マッピング ダイアログで表示しているメモリ・ マッピングを個々に表示します(ノン・マップ領域を除く)。 表示形式:

(3) [アドレス]の指定

この項目は、「(2) [検索する範囲]の指定」で[アドレス範囲を指定する]を選択した場合のみ有効となります。

メモリ値検索の対象となるアドレス範囲を"開始アドレス-終了アドレス"で指定します。それぞれのテキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字),またはドロップダウン・リストにより入力履歴項目(最大履歴個数:10個)を選択することにより行います。 入力したアドレス式の計算結果を,それぞれ開始アドレス/終了アドレスとして扱います。

ただし、検索可能なアドレスの上限値は、プログラム空間の上限アドレス(0xFFFFFFF)です。

- また, 32 ビットで表現できる値より大きいアドレス値を指定することはできません。
- 備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- 備考 2. "開始アドレス"が空欄の場合は、"0x0"の指定として扱われます。
- 備考 3. "終了アドレス"が空欄の場合は、マイクロコントローラのアドレス空間の上限値の指定として扱われます。
- (4) [前を検索] / [次を検索] ボタンのクリック
 [前を検索] ボタンをクリックすると、指定した範囲内でアドレスの小さい方向に検索を行い、検索結果箇所を メモリ パネル上で選択状態にします。
 [次を検索] ボタンをクリックすると、指定した範囲内でアドレスの大きい方向に検索を行い、検索結果箇所を メモリ パネル上で選択状態にします。

2.11.1.6 メモリの内容を一括して変更(初期化)する

メモリの値を一括して変更(初期化)することができます。 コンテキスト・メニューの[初期化...]を選択することにより,指定したアドレス範囲のメモリ値を一括して変更す るためのメモリ初期化 ダイアログがオープンします。 このダイアログにおいて,次の手順で操作を行ってください。

図 2.123 メモリ内容の一括変更(メモリ初期化 ダイアログ)

メモリ初期化	
開始アドレス/シンボル(S):	終了アドレス/シンボル(E):
• -	(終了アドレス/シンボルを入力 🍺 💌
約期化データ(D:	
₩₩ (初期化データを16進数で入力してくだ	さい。スペース区切りで複数指 💀 💌
OK	**>セル へルプ(H)

- (1) [開始アドレス/シンボル]と[終了アドレス/シンボル]の指定 メモリの内容を初期化するアドレス範囲を[開始アドレス/シンボル]と[終了アドレス/シンボル]に指定し ます。それぞれのテキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字),またはド ロップダウン・リストにより入力履歴項目(最大履歴個数:10個)を選択します。 入力したアドレス式の計算結果を、それぞれ開始アドレス/終了アドレスとして扱います。 なお、マイクロコントローラのアドレス空間よりも大きいアドレス値を指定することはできません。
 - 注意 エンディアンの異なる領域をまたいだアドレス範囲を指定することはできません。

備考	このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシ
	ンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(2) [初期化データ]の指定

メモリに書き込む初期化データを指定します。 16 進数の数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストにより入力履歴項目 (最大履歴個数:10 個)を選択します。

初期化データを複数指定する場合は、1個4バイト(8文字)までのデータを最大16個まで、半角スペースで 区切り指定します。

個々の初期化データは、文字列終端より2文字単位で1バイトと解釈され、奇数文字数の場合は先頭1文字で1 バイトと解釈されます。

なお,バイト数が2バイト以上の場合は,次のように初期化対象となるアドレス範囲のエンディアンに合わせ たバイト列に変換してターゲット・メモリへの書き込み処理を行います。

入力文字列(初期化データ)	書き込みイメージ(バイト単位)			
	リトルエンディアン	ビッグエンディアン		
1	01	01		
0 12	00 12	00 12		
00 012 345	00 12 00 45 03	00 00 12 03 45		
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45		

(3) [OK] ボタンのクリック

[OK] ボタンをクリックします。

指定したアドレス範囲のメモリ領域に,指定した初期化データのパターンを繰り返し書き込みます(パターンの 途中で終了アドレスに達した場合は書き込みを終了します)。

ただし、不正な値やアドレス式を指定している場合、メッセージを表示し、メモリ値の初期化は行いません。

2.11.1.7 メモリの表示内容を保存する

メモリの内容を範囲指定して, テキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得し、このパネル上での表示形式に従ったデータで 保存します。

[ファイル]メニュー→[名前を付けてメモリ・データを保存…]を選択すると、次のデータ保存 ダイアログがオー プンします(この際,パネル上で範囲選択した状態でこの操作を行うと選択範囲のみのメモリ・データを保存すること ができます)。

このダイアログにおいて、次の手順で操作を行ってください。

図 2.124 メモリ・データの保存(データ保存 ダイアログ)

データ保存 - メモ	リ・データ	- ×-
ファイル名(N):	以我们	•
ファイルの種類①	テキスト・ファイル (*カxt)	•
保存範囲 アドレス	/シンボル(<u>A</u>):	
0x00000000	- 0×0000016f	
	【保存(S) キャンセル 〔	ヘルプ(田)

(1) [ファイル名]の指定 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより、ファイル を選択することもできます。

(2) [ファイルの種類]の指定 保存するファイルの形式を次のドロップダウン・リストにより選択します。 選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式(デフォルト)
CSV(カンマ区切り)(*.csv)	CSV 形式 ^注

注 各データを "," で区切り保存します。 なお, データ内に "," が含まれている際の不正形式を避けるため, 各データを """"(ダブルクォー テーション)で括り出力します。

 (3) [保存範囲 アドレス/シンボル]の指定 ファイルに保存する範囲を"開始アドレス"と"終了アドレス"で指定します。 それぞれのテキスト・ボックスに16進数の数値/アドレス式を直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10個)。 なお、パネル上で範囲選択している場合は、デフォルトでその選択範囲がテキスト・ボックスに指定されます。 範囲選択していない場合は、現在のパネルの表示範囲が指定されます。
 備考 このテキスト・ボックスで[Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(4) [保存] ボタンのクリック 指定したファイルに、指定した形式でメモリ・データを保存します。

図 2.125 メモリ・データ保存の際の出力イメージ

【テキスト・ファイル(*.txt)で保存】 (16 進表記 /8 ビット幅 /ASCII コードの場合の例)

【CSV ファイル(*csv)で保存】 (16 進表記 /8 ビット幅 /ASCII コードの場合の例)

備考 [ファイル]メニュー→ [メモリ・データを保存]の選択によりパネルの内容を上書き保存する場合, メモリパネル(メモリ1~4)はそれぞれ個別に扱われます。 また,保存範囲についても,前回指定したアドレス範囲で保存されます。



2.11.2 CPU レジスタを表示/変更する

CPU レジスタ(汎用レジスタ/制御レジスタ)の内容の表示,および値の変更は,次の CPU レジスタ パネルで行います。

[表示] メニュー→[CPU レジスタ]を選択してください。 なお、各エリアの見方、および機能についての詳細は、CPU レジスタ パネルの項を参照してください。

図 2.126	CPU レジスタの内容の表示(CPU レジスタ パネル)

_	CPUレジスタ		×
ツールバーー	🕘 表記(N)+ 💌		
	レジスタ名	個	*
	□□□ 汎用レジスタ		
	ISP	0×00001838	E
	USP I INTR	0×00001734 0×ffff8154	
	PC	0xffff86c5	
	PSW	0×00130000	
	BPSV	0×00000000	
	FINTV	0×0000000	-

ここでは、次の操作方法について説明します。

- 2.11.2.1 値の表示形式を変更する
- 2.11.2.2 CPU レジスタの内容を変更する

2.11.2.3 CPU レジスタの表示内容を保存する

2.11.2.1 値の表示形式を変更する

[値] エリアの表示形式は、ツールバーの次のボタンにより、自由に変更することができます。

表記	値の表示形式を変更する次のボタンを表示します。
🔤 自動	選択している項目(下位項目を含む)の値を規定値で表示します(デフォルト)。
🔤 16 進数	選択している項目(下位項目を含む)の値を 16 進数で表示します。
🔛 符号付き 10 進数	選択している項目(下位項目を含む)の値を符号付き 10 進数で表示します。
🔛 符号なし 10 進数	選択している項目(下位項目を含む)の値を符号なし 10 進数で表示します。
8 進数	選択している項目(下位項目を含む)の値を8進数で表示します。
2 進数	選択している項目(下位項目を含む)の値を2進数で表示します。
ASCII	選択している項目(下位項目を含む)の文字列を ASCII コードで表示します。 対象が 2 バイト以上ある場合は、1 バイトずつの文字を並べて表示します。
Fla Float	選択している項目を Float で表示します。 ただし、4 バイト・データ以外の場合は、規定値で表示します。
Double	選択している項目を Double で表示します。 ただし、8 バイト・データ以外の場合は、規定値で表示します。
Pres .	値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記します。

2.11.2.2 CPU レジスタの内容を変更する

CPU レジスタの値は、編集することができます。

[値] エリアにおいて、対象 CPU レジスタ値を選択したのち再度クリックすると、値が編集モードになります ([Esc] キーの押下で編集モードをキャンセルします)。

値をキーボードより直接編集したのち、[Enter]キーを押下することにより、変更した値がデバッグ・ツールのター ゲット・メモリに書き込まれます。

注意 この操作は、プログラム実行中に行うことはできません。

2.11.2.3 CPU レジスタの表示内容を保存する

[ファイル] メニュー→ [名前を付けて CPU レジスタ・データを保存 …] を選択することにより,名前を付けて保存 ダイアログをオープンし, CPU レジスタのすべての内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存 することができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得します。

図 2.127 CPU レジスタ保存の際の出力イメージ

レジスタ名	值
カテゴリ名 - レジスタ名 :	<i>値</i> :

2.11.3 I/O レジスタを表示/変更する

I/O レジスタの内容の表示,および値の変更は,次の IOR パネルで行います。 [表示] メニュー→ [IOR] を選択してください。 なお,各エリアの見方,および機能についての詳細は,IOR パネルの項を参照してください。

図 2.128 I/O レジスタの内容の表示(IOR パネル)

ェッアー	1			
4	IOR	値	型情報(バイト数)	7102/
	ADO . ADDRA	0×0000	10R(2)	0×00088040
	ADO . ADDRB	0×0000	10R(2)	0×00088042
	ADO . ADDRC	0×0000	10R(2)	0×00088044
	ADO . ADDRD	0×0000	10R(2)	0×00088046
	ADO . ADCSR	0x00	IOR(1)	0×00088050
	ADO . ADCSR . CH	0 × 0	10R(4ピット)	0×00088050.0
	ADO . ADCSR . ADST	0 x0	10R(1ピット)	0×00088050.5
	AD0.ADCSR.ADIE	0 x0	10R(1ピット)	0×00088050.6
	ADO . ADCR	0x00	IOR(1)	0×00088051
	ADO . ADCR. MODE	0 x0	10R(2ピット)	0×00088051.0

ここでは、次の操作方法について説明します。

2.11.3.1 I/O レジスタを検索する
2.11.3.2 I/O レジスタを整理する
2.11.3.3 値の表示形式を変更する
2.11.3.4 I/O レジスタの内容を変更する
2.11.3.5 プログラム実行中に I/O レジスタの内容を表示/変更する

2.11.3.6 I/O レジスタの表示内容を保存する

2.11.3.1 I/O レジスタを検索する

I/O レジスタ名を検索することができます。

検索エリアにおいて、テキスト・ボックスに検索する I/O レジスタ名を指定します(大文字/小文字不問)。キーボードより文字列を直接入力するか(最大指定文字数:512文字)、ドロップダウン・リストより入力履歴項目を選択します(最大履歴:10個)。

次のいずれかのボタンをクリックします。

テキスト・ボックスで指定している文字列を含む I/O レジスタ名を上方向に検索し, 態にします。	検索結果を選択状
テキスト・ボックスで指定している文字列を含む I/O レジスタ名を下方向に検索し, 態にします。	検索結果を選択状

備考 1. カテゴリ(フォルダ)により分類されて非表示の状態の I/O レジスタ名も検索します(展開して選択状 態となります)。

備考 2. 検索対象の文字列入力後, [Enter] キーを押下することにより, 💫 ボタンのクリックと同等の動作を行い, [Shift]+[Enter] キーを押下することにより, 🌈 ボタンのクリックと同等の動作を行います。

2.11.3.2 I/O レジスタを整理する

各 I/O レジスタを任意のカテゴリ(フォルダ)で分類し、ツリー形式を編集することができます。

- 注意 1. カテゴリ内にカテゴリを作成することはできません。
- 注意 2. I/O レジスタの追加/削除はできません。
- (1) カテゴリを新規作成する場合 作成したい I/O レジスタ名にキャレットを移動したのち、ツールバーの り新規カテゴリ名を直接入力します。
- (2) カテゴリ名を編集する場合 編集したいカテゴリ名を選択したのち、再度クリックし、キーボードよりカテゴリ名を直接編集します。
- (3) カテゴリを削除する場合 削除したいカテゴリを選択したのち、ツールバーのメボタンをクリックします。 ただし、削除できるカテゴリは、空のカテゴリのみです。
- (4) 表示順を変更する場合
 I/O レジスタ名をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより、I/O レジスタはカテゴリで分類されます。
 また、カテゴリと I/O レジスタ名の表示の順番(上下位置)も、ドラッグ・アンド・ドロップ操作により自由に変更することができます。
- 2.11.3.3 値の表示形式を変更する

[値]	エリアの表示形式は、	ツールバーの次のボタンにより,	自由に変更することができます。
-----	------------	-----------------	-----------------

表記		値の表示形式を変更する次のボタンを表示します。
	🔤 16 進数	選択している項目の値を 16 進数で表示します(デフォルト)。
	🔜 符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
	🔛 符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
	8 進数	選択している項目の値を8進数で表示します。
	🔤 2 進数	選択している項目の値を2進数で表示します。
	ASCII	選択している項目の値を ASCII コードで表示します。

P	選択している項目の値表示の末尾に、	その値の 16 進数表記を "	()"で囲んで併記し
	ま 9。		

2.11.3.4 I/O レジスタの内容を変更する

I/O レジスタの値は、編集することができます。

[値] エリアにおいて,対象 I/O レジスタ値を選択したのち再度クリックすると,値が編集モードになります([Esc] キーの押下で編集モードをキャンセルします)。

値をキーボードより直接編集したのち、[Enter]キーを押下することにより、変更した値がデバッグ・ツールのター ゲット・メモリに書き込まれます。

- 注意 1. この操作は、プログラム実行中に行うことはできません。
- 注意 2. 読み込み専用の I/O レジスタの値を変更することはできません。
- 備考 1. I/O レジスタのサイズより小さい桁の数値が入力された場合、上位の桁を 0 でパディングします。
- 備考 2. I/O レジスタのサイズより大きい桁の数値が入力された場合,上位の桁をマスクします。
- 備考 3. I/O レジスタの値には ASCII 文字による入力も可能です。
 - I/O レジスタ名 "CRC.CRCDOR" の値に "0x41" を書き込んだ場合 → CRC.CRCDOR に, "0x41" が書き込まれます。
 - I/O レジスタ名 "CRC.CRCDOR" の値に ASCII 文字 "A" を書き込んだ場合 → CRC.CRCDOR に, "0x41" が書き込まれます。

2.11.3.5 プログラム実行中に I/O レジスタの内容を表示/変更する

対象となる I/O レジスタをウォッチ式としてウォッチ パネルに登録することにより、プログラムが停止状態だけでな く、実行状態であっても I/O レジスタの値をリアルタイムに表示/変更することができます。 ウォッチ式についての詳細は、「2.11.6 ウォッチ式を表示/変更する」を参照してください。

2.11.3.6 I/O レジスタの表示内容を保存する

[ファイル] メニュー→ [名前を付けて IOR データを保存 …] を選択することにより,名前を付けて保存 ダイアログ をオープンし,I/O レジスタのすべての内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存することがで きます(このパネル上での表示/非表示の設定に関わらず,すべてのI/O レジスタの値が対象となります)。 ファイルに保存する際は,I/O レジスタの値を再読み込みし,取得した最新の値を保存します。

ただし、読み込み保護対象の I/O レジスタの再読み込みは行いません。最新の内容を保存したい場合は、コンテキスト・メニューの [値を強制読み込み]を選択したのち、ファイルの保存を行ってください。

図 2.129 I/O レジスタ保存の際の出力イメージ

IOR 名	值	型情報(バイト数)	アドレス	
<i>カテゴリ名</i> -IOR <i>名</i> :	<i>値</i> :	<i>型情報(バイト数)</i> :	アドレス :	

2.11.4 グローバル変数/スタティック変数を表示/変更する

グローバル変数,またはスタティック変数の値の表示/変更は,ウォッチパネルで行います。 値の表示/変更を行いたい変数をウォッチ式としてウォッチパネルに登録してください。 ウォッチ式についての詳細は,「2.11.6 ウォッチ式を表示/変更する」を参照してください。

2.11.5 ローカル変数を表示/変更する

ローカル変数の内容の表示,および値の変更は,次のローカル変数パネルで行います。 [表示] メニュー→ [ローカル変数]を選択してください。 目的のローカル変数の内容を表示するためには,スコープ・エリアでスコープの選択をします。 ローカル変数 パネルでは、ローカル変数名や関数名を表示します。また、関数の引数もローカル変数として表示します。 す。

- なお、各エリアの見方、および機能についての詳細は、ローカル変数 パネルの項を参照してください。
- **注意** プログラム実行中は、このパネルには何も表示されません。 プログラムの実行が停止したタイミングで、各エリアの表示を行います。



[名前] エリア [値] エリア [型情報 (バイト数)] エリア [アドレス] エリア

ここでは、次の操作方法について説明します。

2.11.5.1 値の表示形式を変更する 2.11.5.2 ローカル変数の内容を変更する 2.11.5.3 ローカル変数の表示内容を保存する

2.11.5.1 値の表示形式を変更する

[値] エリアの表示形式は、ツールバーの次のボタンにより、自由に変更することができます。

表記		値の表示形式を変更する次のボタンを表示します。	
	自動	このパネル上の値の表記を変数ごとの規定値で表示します(デフォ ルト)。	
	16 進数	このパネル上の値を 16 進数で表示します。	
	🔤 10 進数	このパネル上の値を 10 進数で表示します。	
	┏ᡂ 8 進数	このパネル上の値を8進数で表示します。	
	■ 2 進数	このパネル上の値を2進数で表示します。	
	🔤 配列のインデックスを 10 進数表記	このパネル上の配列のインデックスを 10 進数で表示します(デ フォルト)。	
	遍 配列のインデックスを 16 進数表記	このパネル上の配列のインデックスを 16 進数で表示します。	
	Fla Float	このパネル上の値を Float で表示します。 ただし,4 バイト・データ以外,または型情報を持つ場合は,規定 値で表示します。	
	Double	このパネル上の値を Double で表示します。 ただし,8 バイト・データ以外,または型情報を持つ場合は,規定 値で表示します。	
d'ante		値表示の末尾に,その値の 16 進数表記を"()" で囲んで併記しま す。	
L	ニンコード	 文字列変数のエンコードを変更する次のボタンを表示します。	



ASCII	文字列変数を ASCII コードで表示します(デフォルト)。
Shift_JIS	文字列変数を Shift_JIS コードで表示します。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。

2.11.5.2 ローカル変数の内容を変更する

ローカル変数の値、および引数の値は、編集することができます。

- [値] エリアにおいて、対象ローカル変数値/引数値を選択したのち再度クリックすると、値が編集モードになります ([Esc] キーの押下で編集モードをキャンセルします)。
- 値をキーボードより直接編集したのち、[Enter]キーを押下することにより、変更した値がデバッグ・ツールのター ゲット・メモリに書き込まれます。この際に、値のチェックを行い、型に不適合な場合は編集を無効とします。
 - **注意** この操作は、プログラム実行中に行うことはできません。
 - 備考 1. 変数のサイズより小さい桁の数値が入力された場合,上位の桁を0でパディングします。
 - 備考 2. 変数のサイズより大きい桁の数値が入力された場合、上位の桁をマスクします。
 - 備考 3. 文字配列(char 型, unsigned char 型)に対しては、表示形式に ASCII が選択されている場合、文字列 (ASCII/Shift_JIS/EUC-JP/Unicode(UTF-8/UTF-16))による値の入力も可能です。
 - 備考 4. ローカル変数の値には、次のように ASCII 文字による入力も可能です。
 - ASCII 文字による入力の場合 変数 "ch"の [値] エリアに "'A" を入力 → "ch" が割り当てられているメモリ領域に "0x41" を書き込む
 - 数値による入力の場合
 変数 "ch" の [値] エリアに "0x41" を入力
 → "ch" が割り当てられているメモリ領域に "0x41" を書き込む
 - 文字列(ASCII)による入力の場合
 文字配列 "str"の表示形式をASCIIに設定し、[値]エリアに ""ABC""を入力
 → "str"が割り当てられているメモリ領域に "0x41, 0x42, 0x43, 0x00" を書き込む

2.11.5.3 ローカル変数の表示内容を保存する

[ファイル] メニュー→ [名前を付けてローカル変数データを保存 ...]を選択することにより、名前を付けて保存 ダ イアログをオープンし、ローカル変数のすべての内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存する ことができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得します。

なお、配列、ポインタ型変数、構造体/共用体、CPU レジスタ(部分を表す名前が付与されているもののみ)を展開 表示している場合では、各展開要素の値も保存されます。展開表示していない場合は、先頭に "+" マークが付与され、 値は空欄となります。

図 2.131 ローカル変数保存の際の出力イメージ

スコープ: <i>現在のスコ−</i> [V] 変数 [P] 引数 名前	ープ [F] 関数 値	型情報(バイト数)	アドレス
[V] <i>変数名</i> [1]		<i>型情報(バイト数</i>)	アドレス
-[V] <i>変数名</i> [0]	値	<i>型情報(バイト数</i>)	アドレス
:	:	:	:



2.11.6 ウォッチ式を表示/変更する

C 言語変数, CPU レジスタ, I/O レジスタ, およびアセンブラ・シンボルなどをウォッチ式として, 次のウォッチパネルに登録することにより, それらの値を常にデバッグ・ツールから取得し, 一括して値を監視することができます。 また, ウォッチ式は, プログラムが実行中の状態にあっても値の表示を逐次更新することができます(「2.11.6.7 プロ

グラム実行中にウォッチ式の内容を表示/変更する」参照)。

ウォッチ パネルは, [表示]メニュー→[ウォッチ]→[ウォッチ 1 ~ 4]の選択でオープンします。

ウォッチ パネルは、最大4個までオープンすることができます。各パネルは、タイトルバーの"ウォッチ1"、"ウォッ チ2"、"ウォッチ3"、"ウォッチ4"の名称で識別され、それぞれのウォッチパネルが個別にウォッチ式を登録/管理し、 プロジェクトのユーザ情報として保存されます。

なお,各エリアの見方,および機能についての詳細は,ウォッチパネルの項を参照してください。

注意 プログラム実行中に CPU レジスタの内容を表示/変更することはできません。

図 2.132 ウォッチ式の内容の表示(ウォッチ パネル)



ここでは、次の操作方法について説明します。

2.11.6.1 ウォッチ式を登録する
2.11.6.2 登録したウォッチ式を整理する
2.11.6.3 登録したウォッチ式を編集する
2.11.6.4 ウォッチ式を削除する
2.11.6.5 値の表示形式を変更する
2.11.6.6 ウォッチ式の内容を変更する
2.11.6.7 プログラム実行中にウォッチ式の内容を表示/変更する
2.11.6.8 ウォッチ式をエクスポート/インポートする
2.11.6.9 ウォッチ式の表示内容を保存する

2.11.6.1 ウォッチ式を登録する

ウォッチ式の登録方法には、次の3通りがあります(デフォルトでは、ウォッチ式は登録されていません)。

- **注意 1.** 1 つのウォッチ パネルにおいて、ウォッチ式は 128 個まで登録することができます(上限値を越えて登録しようとした場合、メッセージを表示します)。
- 注意 2. コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データがスタック /レジスタに存在しない場合があります。この場合、対象となる変数をウォッチ式として登録しても値の表示は "?"のままとなります。
- 備考 1. 各ウォッチ パネル(ウォッチ 1 ~ウォッチ 4)上で登録したウォッチ式は、それぞれ個別に管理され、 プロジェクトのユーザ情報として保存されます。
- 備考 2. ウォッチ式は、同名を複数登録することができます。

(1) 他のパネルから登録する場合 CS+の他のパネルから、ウォッチ式を登録することができます。 他のパネルにおいて、ウォッチ式として登録したい対象を任意のウォッチパネル(ウォッチ1~ウォッチ4) 上に直接ドラッグ・アンド・ドロップします。 なお、この操作が可能なパネルと、ウォッチ式として登録可能な対象との関係についての詳細は、「表 A.5 各パ ネルとウォッチ式として登録可能な対象の関係」を参照してください。

達アセンブル1				8	
【2】 30 33 36 50 表示(1)・ ffff8668 2b05 ffff8668 386cff 50: 2 CharBu ffff8670 fbf200 ffff8670 fbf200 fbf28878 fbf2878	f = (chaQg_Int8uf	6 D×006	ここでは, 逆アセンブ/ "g_IntBuf"をウォッチ式 逆アセンブル パネル上 択したのち, ウォッチ グ・アンド・ドロップ	ル パネル上の変数 として登録します。 の変数 "g_IntBuf" を選 パネル上へ直接ドラッ します。	
ffff867c cOfe ffff867e 6708	0#9ቻ1 ■ 🗣 🧞 🖫 >	<│表記(N)- 199		
	ウォッチ式 物にIntBuf	8 (0-00)	値 型情報(パイト数) 00008) int(4)	アドレス メモ 1x00101414	

図 2.133 他のパネルからウォッチ式登録する場合の例

- 備考 ウォッチ式として登録したい対象を選択したのち、または対象文字列のいずれかにキャレットを 移動したのち(対象は自動的に決定されます)、コンテキスト・メニューの[ウォッチ1に登録] を選択することによっても同様にウォッチ式を登録することができます(ただし、ウォッチパネ ル(ウォッチ1)に限定)。
- (2) ウォッチ パネル上で直接登録する場合

任意のウォッチ パネル (ウォッチ1~ウォッチ4) において, ツールバーの 左 ボタンをクリックすると, [ウォッチ式] エリアに次のエントリ・ボックスが表示されます。

図 2.134 ウォッチ式のエントリ・ボックス

ウォッチ式 🛛 🗠	値	型情報()	(イト数)	アドレス
🐳 ars_a	1003761 (0×000f50f1)	int(4)		0×00006
😜 arg_c	1003761 (0×000f50f1)	int(4)		0×00008
	į	5		?
כのエי	リアに直接ウォッチ式を入力	します。		

エントリ・ボックス内に、キーボードより直接ウォッチ式を入力したのち、[Enter]キーを押下します。 なお、この際のウォッチ式の入力形式については、次の表を参照してください。

- 「表 A.6 ウォッチ式の入力形式」
- -「表 A.8 C 言語変数をスコープ指定してウォッチ登録した場合の扱い」
- 「表 A.9 CPU レジスタをスコープ指定してウォッチ登録した場合の扱い」
- 「表 A.10 I/O レジスタをスコープ指定してウォッチ登録した場合の扱い」

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシ ンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(3) 他のアプリケーションから登録する場合 外部エディタなどから、C言語変数 /CPU レジスタ /I/O レジスタ /アセンブラ・シンボルの文字列を選択し、 ウォッチ パネル(ウォッチ1~ウォッチ4)に直接ドラッグ・アンド・ドロップします。 この場合、ドロップした文字列がそのままウォッチ式として登録されます。

2.11.6.2 登録したウォッチ式を整理する

登録したウォッチ式をカテゴリ(フォルダ)で分類し、ツリー形式で表示することができます(デフォルトでは、カ テゴリは存在しません)。

注意 1. カテゴリ内にカテゴリを作成することはできません。

- **注意 2.** 1 つのウォッチ パネルにおいて,カテゴリは 64 個まで作成することができます(上限値を越えて作成しようとした場合,メッセージを表示します)。
- (1) カテゴリを新規作成する場合 作成したい位置にキャレットを移動したのち、ツールバーの ゴリ名を直接入力します。
- (2) カテゴリ名を編集する場合 編集したいカテゴリ名を選択したのち、再度クリックし、キーボードよりカテゴリ名を直接編集します。
- (3) カテゴリを削除する場合
 削除したいカテゴリを選択したのち、ツールバーの ズボタンをクリックします。
- (4) 表示順を変更する場合 登録済みのウォッチ式を作成したカテゴリ内に直接ドラッグ・アンド・ドロップすることにより、ウォッチ式は カテゴリで分類されます。 また、カテゴリとウォッチ式の表示の順番(上下位置)も、ドラッグ・アンド・ドロップ操作により自由に変更 することができます。
- 備考 ウォッチ式/カテゴリを他のウォッチ パネル(ウォッチ 1 ~ウォッチ 4)にドラッグ・アンド・ドロップすると、ドロップ先のウォッチ パネルにウォッチ式/カテゴリがコピーされます。

2.11.6.3 登録したウォッチ式を編集する

登録したウォッチ式は、編集することができます。

対象ウォッチ式をダブルクリックすると、対象ウォッチ式が編集モードになります([Esc] キーの押下で編集モードをキャンセルします)。

キーボードより直接内容を編集し, [Enter] キーを押下してください。

2.11.6.4 ウォッチ式を削除する

登録したウォッチ式を削除する場合は,ウォッチパネルにおいて,削除したいウォッチ式を選択したのち,ツール バーの×ボタンをクリックします。

2.11.6.5 値の表示形式を変更する

[値] エリアの表示形式は、ツールバーの次のボタンにより、自由に変更することができます。

表記		値の表示形式を変更する次のボタンを表示します。
	🔜 自動	選択しているウォッチ式の値の表記を変数ごとの規定値(「表 A.11 ウォッチ式の表 示形式(デフォルト)」参照)で表示します(デフォルト)。
	🔤 16 進数	選択している項目の値を 16 進数で表示します。
	🔜 符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
	🔛 符号無し 10 進数	選択している項目の値を符号なし 10 進数で表示します。
	8 進数	選択している項目の値を8進数で表示します。
	🔤 2 進数	選択している項目の値を2進数で表示します。
	ASCII	選択している項目の値を ASCII コードで表示します。
	Fla Float	選択している項目の値を Float で表示します。 ただし,選択しているウォッチ式が 4 バイト・データの場合のみ有効となります。
	Double	選択している項目の値を Double で表示します。 ただし,選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
E	8	選択している項目の値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記し ます
		6.)。 ただし,16 進数表記をしている場合は併記しません。

2.11.6.6 ウォッチ式の内容を変更する

ウォッチ式の値は、編集することができます。

[値] エリアにおいて、対象ウォッチ式の値をダブルクリックすると、値が編集モードになります([Esc] キーの押下で編集モードをキャンセルします)。

値をキーボードより直接編集したのち、[Enter]キーを押下することにより、変更した値がデバッグ・ツールのター ゲット・メモリに書き込まれます。

ただし, 値を変更できるのは, C 言語変数 /CPU レジスタ /I/O レジスタ /アセンブラ・シンボルと1対1に対応する ウォッチ式のみです。また, 読み込み専用の I/O レジスタの値を変更することもできません。

なお、ウォッチ式の値の編集は、プログラム実行中の状態でも行うことができます。設定方法についての詳細は、 「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

- 備考 1. 変数のサイズより小さい桁の数値が入力された場合,上位の桁を0 でパディングします。
- 備考 2. 変数のサイズより大きい桁の数値が入力された場合、上位の桁をマスクします。
- 備考 3. 文字配列(char 型, unsigned char 型)に対しては、表示形式に ASCII が選択されている場合、文字列 (ASCII/Shift_JIS/EUC-JP/Unicode(UTF-8)/Unicode(UTF-16))による値の入力も可能です。
- 備考 4. ウォッチ式の値には、次のように ASCII 文字による入力も可能です。
 - ASCII 文字による入力の場合
 変数 "ch"の [値] エリアに "A" を入力
 → "ch" が割り当てられているメモリ領域に "0x41" を書き込む
 - 数値による入力の場合 変数 "ch"の[値] エリアに "0x41"を入力 → "ch"が割り当てられているメモリ領域に "0x41"を書き込む
 - 文字列(ASCII)による入力の場合
 文字配列 "str"の表示形式をASCIIに設定し、[値]エリアに ""ABC""を入力
 → "str"が割り当てられているメモリ領域に "0x41, 0x42, 0x43, 0x00" を書き込む

2.11.6.7 プログラム実行中にウォッチ式の内容を表示/変更する

メモリ パネル/ウォッチ パネルでは、プログラムの実行中に、リアルタイムにメモリ/ウォッチ式の内容を表示更 新、および書き換えることができるリアルタイム表示更新機能を備えています。

このリアルタイム表示更新機能を有効にすることにより、プログラムが停止している状態の時だけでなく、実行中の 状態であっても、メモリ/ウォッチ式の値の表示/変更を行うことができます。

設定方法についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」を参照してください。

2.11.6.8 ウォッチ式をエクスポート/インポートする

現在登録しているウォッチ式をファイルにエクスポートし、そのファイルをインポートすることにより、ウォッチ式 を再登録することができます。

この場合、次の操作を行ってください。

(1) ウォッチ式をエクスポートする

現在登録しているウォッチ式(カテゴリを含む)を,インポート可能なファイル形式で保存します。ウォッチ パネルにフォーカスがある状態で,[ファイル]メニュー→[名前を付けてウォッチ・データを保存...]を選択し ます。

オープンする名前を付けて保存 ダイアログにおいて、次の指定を行ったのち、[保存]ボタンをクリックします。

- [ファイル名]:保存するファイル名を指定します(拡張子は "*.csv"に限ります)。

- [ファイルの種類]:"インポート可能 CSV(カンマ区切り)(*.csv)"を選択します。

注意 値,および型情報は保存されません。 また,配列や構造体などのウォッチ式を解析後に展開される項目は保存されません。



	anapie •	• •	Sampletorigen		
整理 ▼ 新しいフォルダー				油 •	0
乗 コンピューター ●● ローカル ディスク (C:)	▲ 名前 ● D ● S	efaultBuild c			
ファイル系(N): Watch Export	+ +	m			
st the day.	505い(カンフヌビルの	1(*.csv)			

図 2.135 ウォッチ式のエクスポート

(2) ウォッチ式をインポートする

(1) でエクスポートしたファイルを、ウォッチ パネルにインポートします。
 ウォッチ式をインポートしたいウォッチ パネルにおいて、コンテキスト・メニューの [ウォッチ式をインポート
 …]を選択します。

オープンするウォッチ式データ・ファイルを開く ダイアログにおいて,先にエクスポートしたファイルを指定したのち,[開く]ボタンをクリックします。

備考 すでにウォッチ式が登録されている場合,最下部のウォッチ式の直後にインポートしたウォッチ 式が登録されます。

図 2.136 ウォッチ式のインポート

🦷 ウォッチ式テータ・ファイルを聞く		*
🚱 🔍 🛛 🖉 « WorkSpace ، Sample ،	✓ 4y Sampleの検索	ρ
整理 ▼ 新しいフォルダー	H • 📶	0
 ● □ンビューター ● ● □ーカル ディスク (C:) 	▲ 名雨 DefaultBuild JefaultBuild Watch_Export.csv E	
ファイル名(N): Watch_Export	 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ 可能CSV(カンマ区) ・ ・ ・ ・ ・ ・ ・ ・ 可能CSV(カンマ区) ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	•



2.11.6.9 ウォッチ式の表示内容を保存する

[ファイル] メニュー→ [名前を付けてウォッチ・データを保存 …] を選択する、またはコンテキスト・メニューの [ウォッチ・データを展開保存 …] を選択することにより、名前を付けて保存 ダイアログをオープンし、ウォッチ式と 値のすべての内容をテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。

ファイルに保存する際は、すべてのウォッチ式の値を再読み込みし、取得した最新の値を保存します。

ただし,読み込み保護対象の I/O レジスタの再読み込みは行いません。最新の内容を保存したい場合は、コンテキスト・メニューの[値を強制読み込み]を選択したのち、ファイルの保存を行ってください。

なお, 配列, ポインタ型変数, 構造体/共用体, レジスタ(部分名がついているもののみ)などの展開表示が可能な ウォッチ式の場合, [名前を付けてウォッチ・データを保存...]から保存した場合と [ウォッチ・データを展開保存...] から保存した場合で動作が異なります。

- [名前を付けてウォッチ・データを保存 ...]から保存した場合 展開表示している場合は、各展開要素の値も保存します。展開表示していない場合は、先頭に "+"マークを付与し て値は空欄になります。
- [ウォッチ・データを展開保存...]から保存した場合 展開状態にかかわらず最大 255 階層まで展開して,各展開要素の値も保存します。

図 2.137 ウォッチ・データ保存の際の出力イメージ

ウォッチ式	値	型情報(バイト数)	アドレス	メモ	
変数式 - カテゴリ名	値	型情報(バイト数)	アドレス	メモ	
変数式	値	型情報(バイト数)	アドレス	メモ	
:	:	:	:	:	

備考 [ファイル]メニュー→ [ウォッチ・データを保存] の選択によりパネルの内容を上書き保存した場合, ウォッチ パネル(ウォッチ 1 ~ 4)はそれぞれ個別に扱われます。



2.12 スタックからの関数呼び出し情報の表示

この節では、スタックからの関数呼び出し情報の表示方法について説明します。 CS+が提供するコンパイラ(CC-RX)は、ANSI規格に沿って関数呼び出し情報をスタックに積んでいます。この関 数呼び出し情報(以降、コール・スタック情報と呼びます)を解析することで、関数の呼び出しの深さ、呼び出し元位 置、および引数などを知ることができます。

2.12.1 コール・スタック情報を表示する

コール・スタック情報の表示は、次のコール・スタックパネルで行います。 [表示] メニュー→ [コール・スタック] を選択してください。 なお、各エリアの見方、および機能についての詳細は、コール・スタックパネルの項を参照してください。

- **注意** プログラム実行中は、このパネルには何も表示されません。 プログラムの実行が停止したタイミングで、各エリアの表示を行います。
- 図 2.138 コール・スタック情報の表示(コール・スタック パネル)



ここでは、次の操作方法について説明します。

2.12.1.1 値の表示形式を変更する 2.12.1.2 ソース行へジャンプする 2.12.1.3 ローカル変数を表示する

2.12.1.4 コール・スタック情報の表示内容を保存する

2.12.1.1 値の表示形式を変更する

ツールバーの次のボタンにより、このパネルの表示形式を変更することができます。

表記		値の表示形式を変更する次のボタンを表示します。
	🔤 自動	このパネル上の値の表記を変数ごとの規定値で表示します(デフォルト)。
	🔜 16 進数	このパネル上の値を 16 進数で表示します。
	🚾 10 進数	このパネル上の値を 10 進数で表示します。
	8 進数	このパネル上の値を8進数で表示します。
	🔤 2 進数	このパネル上の値を2進数で表示します。

Е	ショード	文字列変数のエンコードを変更する次のボタンを表示します。
	ASCII	このパネル上の文字列変数を ASCII コードで表示します(デフォルト)。
	Shift_JIS	このパネル上の文字列変数を Shift_JIS コードで表示します。
	EUC-JP	このパネル上の文字列変数を EUC-JP コードで表示します。
	UTF-8	このパネル上の文字列変数を UTF-8 コードで表示します。
	🔢 UTF-16	このパネル上の文字列変数を UTF-16 コードで表示します。

2.12.1.2 ソース行へジャンプする

行をダブルクリックすることにより、選択行が示す関数呼び出し元のソース行にキャレットを移動した状態でエディ タパネルがオープンします(すでにオープンしている場合は、エディタパネルにジャンプ)。

備考 コンテキスト・メニューの [逆アセンブルヘジャンプ] を選択することにより,現在選択している行が 示す関数呼び出し元のアドレスにキャレットを移動した状態で逆アセンブルパネル(逆アセンブル1) がオープンします(すでにオープンしている場合は,逆アセンブルパネル(逆アセンブル1)にジャ ンプ)。

2.12.1.3 ローカル変数を表示する

コンテキスト・メニューの[このときのローカル変数を表示]を選択することにより、現在選択している行が示す関数のローカル変数を表示するローカル変数パネルをオープンします。

2.12.1.4 コール・スタック情報の表示内容を保存する

[ファイル] メニュー→ [名前を付けてコール・スタック・データを保存 ...] を選択することにより,名前を付けて 保存 ダイアログをオープンし,コール・スタック情報のすべての内容をテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。

ファイルに保存する際は、デバッグ・ツールから最新の情報を取得します。

図 2.139 コール・スタック情報保存の際の出力イメージ

深さ	コール・スタック
0	コール・スタック情報
1	コール・スタック情報
:	



2.13 実行履歴の収集

この節では、プログラムの実行履歴の収集方法について説明します。

ー般的に、プログラムの実行履歴をトレースと呼び、以降の記述で使用します。プログラムが暴走した場合、暴走後のメモリ内容やスタック情報などから原因を探ることは非常に困難ですが、収集したトレース・データの内容を解析することにより、暴走するまでの過程を直接探ることができ、プログラムの潜在的バグを発見するために有効です。

- **注意 1.** 【E20(JTAG)【RX600, RX700 シリーズ】】 トレース機能/リアルタイム RAM モニタ機能(RRM 機能)は一部排他使用の機能です。
- **注意 2.** 【E1】【E20】【EZ Emulator】 アクション・イベントの Printf イベントを使用している場合, Printf イベントが成立するとそれまでの 実行履歴は消去され, イベント成立後の実行履歴が記録されます。
- **注意 3.** 【EZ Emulator【RX100, RX200 シリーズ】】 端子リセットを入れた場合、リセットが入る前のトレース情報は消去されます。EZ Emulator からの再 実行後のトレース情報が再取得されます。
- **注意 4.** 【E1】【E20】【EZ Emulator】 ユーザ・プログラム実行中に端子リセットやウォッチドッグタイマなどによるリセットが発生した場合、リセット前後のトレース記録が正しく行えなくなります。
- **注意 5.** トレース開始イベント,またはトレース終了イベントを設定している場合,プログラム実行中のトレース停止・再開はできません。
- **注意 6.** 【OCD(JTAG)】【OCD(シリアル)】 トレース情報に付加されるタイムスタンプは、フレーム間の経過時間がトレース・クロックの 20 ビッ ト分を超える場合、およびトレース出力でロストが発生した場合、正しい時間となりません。
- 2.13.1 トレース動作の設定をする

トレース機能が開始すると、現在実行中のプログラムの実行過程を記録したトレース・データがトレース・メモリに 収集されます(プログラムの実行が停止すると、自動的にトレース機能も停止します)。

トレース機能を使用するためには、あらかじめトレースの動作に関する設定を行う必要があります。 なお、設定方法は、使用するデバッグ・ツールにより異なります。

2.13.1.1 【E1】の場合
2.13.1.2 【E20】の場合
2.13.1.3 【EZ Emulator】の場合
2.13.1.4 【シミュレータ】の場合

2.13.1.1 【E1】の場合

設定は, プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内で行います。

図 2.140 [トレース] カテゴリ【E1】

ø	FP-2	
	トレース機能の用途	トレース
	トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
	トレース・データ種別	分岐+データアクセス
	データアクセスのバス・マスタ	CPU
	タイム・スタンプ出力	an
	トレース・クロック・カウント・ソース[MHz]	
	トレース・クロック・カウント・ソースの分周比	分周ぬし

(1) [トレース機能の用途] トレース機能のみ使用できます。

(2) [トレース・メモリを使い切った後の動作]
 収集したトレース・データでトレース・メモリがいっぱいになった際の動作を、次のドロップダウン・リストに

ないないになったいの気がないになったいの気が行き、火のキュラングラン・ラストに

トレース・メモリを上	トレース・メモリがいっぱいになると、古いトレース・データに上書きを続けます
書きし実行を続ける	(デフォルト)。

トレースを停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止します (プログラムの実行は停止しません)。
停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止すると 同時にプログラムの実行を停止します。

(3) [トレース・データ種別]

このプロパティは、収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元,分岐先アドレス情報をト レース・データとして収集します。
分岐+データアクセス ^{注 1}	プログラム実行中に発生した分岐元,分岐先アドレス情報および成立したア クセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス ^{注 2}	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・ データとして収集します。

注 1. 【E1(Serial)【RX100, RX200 シリーズ】】 [分岐+データアクセス]のトレース・データを収集することはできません。そのためドロップダ ウン・リストには表示されません。

- 注 2. 【E1(Serial)【RX100, RX200 シリーズ】】 [データアクセス]のトレース・データを収集するには、ポイント・トレースにてアドレス条件を 設定する必要があります。設定しない場合、データアクセスのトレース情報が取得されません。 データアクセスのトレース情報には、指定されたアドレスへのアクセスのみ記録します。なお、 トレース・パネルのアドレスには、必ずポイント・トレースで設定した開始アドレス値が表示さ れます。 ポイント・トレースの詳細については、「2.13.4 条件を満たしたときのみの実行履歴を収集する」 を参照してください。
- (4) [データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グルー プ】

このプロパティは、[トレース・データ種別] プロパティにおいて [分岐+データアクセス] または [データア クセス] を指定した場合にのみ表示されます。

データアクセスのバス・マスタをドロップダウン・リストより指定します。

ドロップダウン・リストには、次のバス・マスタが表示されます。

CPU	CPU からのデータアクセス結果を,トレースパネルに出力します。
DMAC/DTC	DMAC/DTC からのデータアクセス結果を,トレースパネルに出力します。

データアクセスのトレース結果については、指定したバス・マスタからのトレース結果のみをトレースパネルに 出力します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は、[データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 プロパティは表示されません。その場合、バス・マスタは[CPU] 固定となります。

- (5) [タイム・スタンプ出力] このプロパティは、収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合、[はい]を指定してください。
 - **注意 1.** 【E1(Serial)【RX200 シリーズ】】 [トレース・データ種別]プロパティにおいて[分岐]を指定した場合は、タイム・スタンプを出 力することができません。出力する場合は、[データアクセス]を指定してください。

注意 2. 【E1【RX100 シリーズ】】 このプロパティの項目を[いいえ]から変更することはできず、タイム・スタンプを出力するこ とができません。

(6) [トレース・クロック・カウント・ソース [MHz]] タイム・スタンプのカウント・ソースを 0.0001 - 999.999 の間で入力してください。 なお、プロパティが空欄の場合、[接続用設定] タブの [クロック] カテゴリ内の [動作周波数 [MHz]] プロパ ティの設定値がカウント・ソースとして代用されます。

(7) [トレース・クロック・カウント・ソースの分周比] 【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】
 このプロパティは、[タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されます。
 タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。
 ドロップダウン・リストには、次の分周比が表示されます。
 分周なし
 タイム・スタンプ用カウント・ソースの周波数をそのままの値で使用する。

分周なし	ダイム・スタンノ用カワント・ソースの周波数をそのままの値で使用する。
16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。
256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。
4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/4096 倍に変換して使用する。

[トレース・クロック・カウント・ソース [MHz]] プロパティで指定した周波数をさらに指定した値で分周(周波数を1/n倍に変換)した値が実際のタイム・スタンプの単位(1カウントあたりの周波数)になります。

- 注意 1. プログラム実行中にこのプロパティを変更することはできません。
- **注意 2.** トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は、[トレース・クロッ ク・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】プロパティは表示されません。その場合、[分周なし] 固定となります。

2.13.1.2 【E20】の場合

設定は、プロパティパネルの[デバッグ・ツール設定]タブ上の[トレース]カテゴリ内で行います。

図 2.141 [トレース] カテゴリ【E20】

~	FF-2		
	トレース機能の用途	トレース	
	トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける	
	トレース・データ種別	分岐+データアクセス(アクセス・データなし)	
	アクセス・データを取得しない領域の開始アドレス	HEX ()	
	アクセス・データを取得しない領域の終了アドレス	HEM SFFFF	
	データアクセスのバス・マスタ	CPU	
	外部トレース出力	CPU実行優先	
	トレース・メモリ・サイズ[Mバイト]	1	
	タイム・スタンプ出力	はい	
	トレース・クロック・カウント・ソース[MHz]		
	トレース・クロック・カウント・ソースの分周比	分周なし	

(1) [トレース機能の用途]

トレース機能/リアルタイム RAM モニタ機能(RRM 機能)は一部排他使用の機能です。このため、このプロ パティで、どちらの機能を優先的に使用するかを指定します。ここでは、次のドロップダウン・リストより、 [トレース]を選択してください。

トレース	トレース機能を優先的に使用します(デフォルト)。	
	- リアルタイム RAM モニタ機能 使用不可となります。	
リアルタイム RAM モニタ ^{注1}	リアルタイム RAM モニタ機能 (RRM) を優先的に使用します。	
	- トレース機能 制限付きの使用となります。 注2 また、トレース関連のイベントが無効状態となります。	

注 1. 【E20(Serial)】 リアルタイム RAM モニタ機能はサポートしていません。そのため、このプロパティ値には[リア ルタイム RAM モニタ]を指定しないでください。

注 2. 【E20(JTAG)【RX600, RX700 シリーズ】】 トレース機能の一部が使用できなくなります。 制限事項は次のとおりです。

トレース・メモリを使い切った後の動 作	[トレースを停止する], [停止する]は使用不可 [トレース・メモリを上書きし実行を続ける]のみ使 用可
トレース・データ種別	[分岐], [分岐+データアクセス]は使用不可 [データアクセス]のみ使用可
外部トレース出力	[外部出力しない] は使用不可 [CPU 実行優先], [トレース出力優先] は使用可
トレース・メモリ・サイズ [M バイト]	1M バイトのみ使用可

(2) [トレース・メモリを使い切った後の動作]

収集したトレース・データでトレース・メモリがいっぱいになった際の動作を、次のドロップダウン・リストに より指定します。

トレース・メモリを上 書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データに上書きを続けます (デフォルト)。
トレースを停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止します (プログラムの実行は停止しません)。
停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止すると 同時にプログラムの実行を停止します。

(3) [トレース・データ種別]

このプロパティは、[トレース機能の用途]プロパティで[トレース]を選択している場合のみ表示されます。 収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元,分岐先アドレス情報をト レース・データとして収集します。
分岐+データアクセス ^{注1}	プログラム実行中に発生した分岐元,分岐先アドレス情報および成立したア クセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス ^{注 2}	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・ データとして収集します。
分岐+データアクセス (アク セス・データなし) ^{注 3}	プログラム実行中に発生した分岐元,分岐先アドレス情報および成立したア クセス・イベントのデータ情報をトレース・データとして収集します。 ただし,アクセス・データは収集しません。
データアクセス (アクセス・ データなし) ^{注 3}	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・ データとして収集します。 ただし,アクセス・データは収集しません。

- 注 1. 【E20(Serial)【RX100, RX200 シリーズ】】 [分岐+データアクセス]のトレース・データを収集することはできません。そのためドロップダ ウン・リストには表示されません。
- 注 2. 【E20(Serial)【RX100, RX200 シリーズ】】 [データアクセス]のトレース・データを収集するには、ポイント・トレースにてアドレス条件を 設定する必要があります。ポイント・トレースの詳細については、「2.13.4 条件を満たしたとき のみの実行履歴を収集する」を参照してください。
- 注 3. RX66x, RX65N-2M, RX651-2M, RX671, RX72T, RX72M, RX72N グループのみ使用可能です。
- (4) [アクセス・データを取得しない領域の開始アドレス] 【RX65N-2M, RX651-2M, RX66T, RX671, RX72T, RX72M, RX72N, RX66N グループ】
 アクセス・データを取得しない領域の開始アドレスを指定します。
- (5) [アクセス・データを取得しない領域の終了アドレス] 【RX65N-2M, RX651-2M, RX66T, RX671, RX72T, RX72M, RX72N, RX66N グループ】
 アクセス・データを取得しない領域の終了アドレスを指定します。

(6) [データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グルー プ】

このプロパティは、[トレース・データ種別] プロパティにおいて [分岐+データアクセス] または [データア クセス] を指定した場合にのみ表示されます。

データアクセスのバス・マスタをドロップダウン・リストより指定します。 ドロップダウン・リストには、次のバス・マスタが表示されます。

CPU	CPU からのデータアクセス結果を、トレースパネルに出力します。
DMAC/DTC	DMAC/DTC からのデータアクセス結果を,トレースパネルに出力します。

データアクセスのトレース結果については、指定したバス・マスタからのトレース結果のみをトレースパネルに 出力します。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合は、[データアクセスのバス・マスタ]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 プロパティは表示されません。その場合、バス・マスタは[CPU] 固定となります。

(7) [外部トレース出力] 【E20(JTAG)】

収集するトレース・データの外部出力方法を次のドロップダウン・リストにより指定します。

CPU 実行優先	トレース出力より CPU 実行を優先します。 出力されるトレース情報にロストが発生する場合があります。	
トレース出力優先	CPU 実行よりトレース出力を優先します。 トレース出力のために CPU 実行が停止するため,リアルタイム性が損なわれます。	
外部出力しない	トレース情報は出力されません。 マイクロコントローラの内蔵バッファのみを使用します。	

- **注意 1.** [トレース機能の用途] プロパティにおいて,[リアルタイム RAM モニタ]が指定された場合,ド ロップダウンリストより[外部出力しない]は指定できません。
- **注意 2.** [CPU 実行優先]または[トレース出力優先]を指定した場合, トレース パネルにトレース・デー タが表示された状態でステップ・イン実行すると正しいトレース・データが表示されない場合が あります。
- **注意 3.** 【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 [外部出力しない]から [CPU 実行優先],または [トレース出力優先] に変更した場合,タイマ 計測結果が初期化されます。
- (8) [トレース・メモリ・サイズ [M バイト]] 【E20(JTAG)】
 トレース・データを保持するメモリサイズを次のドロップダウン・リストにより指定します。

- 1(デフォルト), 2, 4, 8, 16, 32

- (9) [タイム・スタンプ出力]
 このプロパティは、収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合、[はい]を指定してください。
 - **注意 1.** 【E20(Serial)【RX200 シリーズ】】 [トレース・データ種別]プロパティにおいて[分岐]を指定した場合は、タイム・スタンプを出 力することができません。出力する場合は、[データアクセス]を指定してください。
 - **注意 2.** 【E20【RX100 シリーズ】】 このプロパティの項目を[いいえ]から変更することはできず、タイム・スタンプを出力するこ とができません。
- (10) [トレース・クロック・カウント・ソース [MHz]] タイム・スタンプのカウント・ソースを 0.0001 - 999.999 の間で入力してください。 なお、プロパティが空欄の場合、[接続用設定] タブの [クロック] カテゴリ内の [動作周波数 [MHz]] プロパ ティの設定値がカウント・ソースとして代用されます。
- (11) [トレース・クロック・カウント・ソースの分周比] 【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 このプロパティは、[タイム・スタンプ出力] プロパティにおいて [はい] を指定した場合にのみ表示されます。 タイム・スタンプのカウント・ソースの分周比をドロップダウン・リストより指定します。 ドロップダウン・リストには、次の分周比が表示されます。

分周なし	タイム・スタンプ用カウント・ソースの周波数をそのままの値で使用する。
16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。
256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。
4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/4096 倍に変換して使用する。

[トレース・クロック・カウント・ソース [MHz]] プロパティで指定した周波数をさらに指定した値で分周(周波数を1/n倍に変換)した値が実際のタイム・スタンプの単位(1カウントあたりの周波数)になります。

注意 1. プログラム実行中にこのプロパティを変更することはできません。

注意 2. トレースのタイム・スタンプ分周機能が無いマイクロコントローラの場合は、[トレース・クロッ ク・カウント・ソースの分周比]【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】プロパティは表示されません。その場合、[分周なし] 固定となります。

2

2.13.1.3 【EZ Emulator】の場合

設定は、プロパティパネルの[デバッグ・ツール設定]タブ上の[トレース]カテゴリ内で行います。

図 2.142 [トレース] カテゴリ【EZ Emulator】

đ	トレース	
	トレース機能の用途	トレース
	トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける
	トレース・データ種別	分岐
	タイム・スタンプ出力	an
	トレース・クロック・カウント・ソース[MHz]	

- (1) [トレース機能の用途] トレース機能のみ使用できます。
- (2) [トレース・メモリを使い切った後の動作] 収集したトレース・データでトレース・メモリがいっぱいになった際の動作を、次のドロップダウン・リストにより指定します。

トレース・メモリを上 書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データに上書きを続けます (デフォルト)。
トレースを停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止します (プログラムの実行は停止しません)。
停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止すると 同時にプログラムの実行を停止します。

(3) [トレース・データ種別]

このプロパティは、収集するトレース・データの種別を次のドロップダウン・リストにより指定します。

分岐	プログラム実行中に発生した分岐処理の分岐元,分岐先アドレス情報をト レース・データとして収集します。
分岐+データアクセス ^{注1}	プログラム実行中に発生した分岐元,分岐先アドレス情報および成立したア クセス・イベントのデータ情報をトレース・データとして収集します。
データアクセス ^{注 2}	プログラム実行中に成立したアクセス・イベントのデータ情報をトレース・ データとして収集します。

注 1. 【RX100, RX200 シリーズ】

[分岐+データアクセス]のトレース・データを収集することはできません。そのためドロップダ ウン・リストには表示されません。

注 2. 【RX100, RX200 シリーズ】 [データアクセス]のトレース・データを収集するには、ポイント・トレースにてアドレス条件を 設定する必要があります。ポイント・トレースの詳細については、「2.13.4 条件を満たしたとき のみの実行履歴を収集する」を参照してください。

- (4) [タイム・スタンプ出力]
 このプロパティは、収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定します。タイム・スタンプを出力する場合、[はい]を指定してください。
 - **注意 1.** 【RX200 シリーズ】 [トレース・データ種別]プロパティにおいて[分岐]を指定した場合は、タイム・スタンプを出 力することができません。出力する場合は、[データアクセス]を指定してください。
 - **注意 2.** 【RX100 シリーズ】 このプロパティの項目を[いいえ]から変更することはできず、タイム・スタンプを出力するこ とができません。
- (5) [トレース・クロック・カウント・ソース [MHz]] タイム・スタンプのカウント・ソースを0.0001 - 999.999の間で入力してください。 なお、プロパティが空欄の場合、[接続用設定]タブの[クロック]カテゴリ内の [動作周波数 [MHz]] プロパ ティの設定値がカウント・ソースとして代用されます。

2.13.1.4 【シミュレータ】の場合

設定は、プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内で行います。

図 2.143 [トレース] カテゴリ【シミュレータ】

ø	トレース		
	トレース機能を使用する	していえ	
	実行前にトレース・メモリをクリアする	はい	
	トレース・メモリを使い切った後の動作	トレース・メモリを上書きし実行を続ける	
	トレース・タイム・タグを積算する	していえ	
	トレース・メモリ・サイズ【フレーム】	64K	

- (1) [トレース機能を使用する]
 トレース機能を使用するか否かをドロップダウン・リストにより指定します。
 トレース機能を使用する場合は[はい]を指定してください(デフォルトでは[いいえ]が指定されます)。
- (2) [実行前にトレース・メモリをクリアする] トレース機能を開始する前に、トレース・メモリを一度クリア(初期化)するか否かをドロップダウン・リスト により指定します。 クリアする場合は[はい]を指定してください(デフォルト)。

(3) [トレース・メモリを使い切った後の動作]

トレース・メモリが収集したトレース・データでいっぱいになった際の動作を、次のドロップダウン・リストに より指定します。

トレース・メモリを上 書きし実行を続ける	トレース・メモリがいっぱいになると、古いトレース・データに上書きを続けます (デフォルト)。 [実行前にトレース・メモリをクリアする] プロパティで [はい] を指定している 場合は、再実行時、トレース・データをクリアしたのちトレース・データの書き込 みを行います。
停止する	トレース・メモリがいっぱいになると、トレース・データの書き込みを停止すると 同時にプログラムの実行を停止します。 [実行前にトレース・メモリをクリアする]プロパティで[いいえ]を指定してい る場合は、再び実行してもプログラムは実行せずに停止します。

(4) [トレース・タイム・タグを積算する]

トレースの時間表示を積算表示にするか否かをドロップダウン・リストにより指定します。 トレースの時間表示を積算表示にする場合は [はい]を、差分表示にする場合は [いいえ] を指定してください (デフォルト)。

(5) [トレース・メモリ・サイズ[フレーム]]
 トレース・メモリのサイズ(トレース・フレーム数)をドロップダウン・リストにより指定します。
 なお、トレース・フレームはトレース・データの一単位を表し、フェッチ/ライト/リードなどで、それぞれ1
 つのトレース・フレームを使用します。
 ドロップダウン・リストには、次のトレース・フレーム数が表示されます。

備考 トレース パネルのツールバーの **い**ボタンをクリックすることにより、トレース・メモリを強制 的にクリアすることができます。

64K (デフォルト), 128K, 256K, 512K, 1M, 2M, 3M

2.13.2 実行停止までの実行履歴を収集する

デバッグ・ツールには、プログラムの実行開始から実行停止までの実行履歴を収集する機能があらかじめ用意されて います。

これにより、プログラムの実行を開始することにより自動的にトレース・データの収集が開始し、実行停止とともに トレース・データの収集も終了します。

- なお、収集したトレース・データの確認方法についての詳細は、「2.13.6 実行履歴を表示する」を参照してください。 注意 【シミュレータ】
- ストリング操作命令,および積和演算命令のデータアクセスは,最初と最後のアクセス履歴のみ収集し ます。

備考 この機能は、デバッグ・ツールにデフォルトで設定されているビルトイン・イベントの1つである無条件トレース・イベントにより動作します。
 この無条件トレース・イベントと後述のトレース・イベント(「2.13.3 任意区間の実行履歴を収集する」および「2.13.4 条件を満たしたときのみの実行履歴を収集する」参照)は排他使用のイベントとなります。そのため、トレース・イベントが有効状態で設定されると、無条件トレース・イベントは自動的に無効状態に変更されます。
 なお、トレース・イベントが設定されていない場合、無条件トレース・イベントのチェックを外して無効状態としてもトレース・データは収集されます。

2.13.3 任意区間の実行履歴を収集する

トレース・イベントを設定することにより、プログラムの実行過程において、任意の区間の実行履歴のみをトレース・データとして収集することができます。

- トレース・イベントは、トレース開始イベントとトレース終了イベントで構成されます。 この機能を使用するためには、次の手順で操作を行います。
 - 2.13.3.1 トレース開始イベント/トレース終了イベントを設定する
 - 2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】
 - 2.13.3.3 プログラムを実行する
 - 2.13.3.4 トレース開始イベント/トレース終了イベントを編集する
 - 2.13.3.5 トレース開始イベント/トレース終了イベントを削除する
- **注意 1.** トレース・イベントの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する 留意事項」も参照してください。
- 注意 2. 【シミュレータ】 ストリング操作命令、および積和演算命令のデータアクセスは、最初と最後のアクセス履歴のみ収集します。

注意 3. 【シミュレータ】 トレース開始イベントからトレース終了イベントまでをトレースする場合、シミュレータではトレース 終了イベントがトレース結果として表示されません。このため、シミュレータを使用する場合はトレー ス終了イベントをトレース・データとして表示させる範囲の1行下に設定してください。

2.13.3.1 トレース開始イベント/トレース終了イベントを設定する

エディタ パネル/逆アセンブル パネルにおいて、トレース・データの収集を開始/終了するイベントを設定します。

 (1) 実行系イベントの場合 実行系イベントをトレース開始/終了イベントに設定することにより、任意の箇所でトレース・データの収集を 開始/終了させることができます。 実行系イベントをトレース開始イベントとして設定する場合、トレース・データの収集を開始したい行/アドレ ス注にキャレットを移動したのち、コンテキスト・メニューの[トレース設定]→[トレース開始の設定]を選 択します。 トレース開始イベントが、キャレット位置の行/アドレスに対応する先頭アドレスの命令に対して設定されま す。 また、実行系イベントをトレース終了イベントとして設定する場合、トレース・データの収集を終了したい行/ アドレス^注にキャレットを移動したのち、コンテキスト・メニューの[トレース設定]→[トレース終了の設 定]を選択します。 トレース終了イベントが、キャレット位置の行/アドレスに対応する先頭アドレスの命令に対して設定されま す。

注意 トレース開始イベント/トレース終了イベントは、アドレス表示がない行に設定することはできません。

トレース開始イベント/トレース終了イベントが設定されると、設定した行/アドレスのイベント・エリアに次のイベント・マークが表示されます。

表 2.11 トレース開始イベント/トレース終了イベント・マーク(実行系イベント)

種別	イベント・マーク
トレース開始	1 9
トレース終了	*

図 2.144 実行系イベントを用いたトレース開始イベント/トレース終了イベントの設定例 (逆アセンブル パネルの場合)



(2) アクセス系イベントの場合

アクセス系イベントをトレース開始/終了イベントに設定することにより、任意の変数、または I/O レジスタに 対し、指定したアクセスがあった場合にトレース・データの収集を開始/終了させることができます。 また、この際に、アクセスした値を限定することもできます。 アクセス系イベントで指定できるアクセス種別は次のとおりです。

表 2.12 変数へのアクセス種別

アクセス種別	説明
リード	指定した変数 /l/O レジスタに, リード・アクセスした(読み込みを行った)際にト レース・データの収集を開始/終了します。
ライト	指定した変数 /l/O レジスタに,ライト・アクセスした(書き込みを行った)際にト レース・データの収集を開始/終了します。
リード/ライト	指定した変数 /l/O レジスタに,リード・アクセス,またはライト・アクセスした(読 み込み,または書き込みを行った)際にトレース・データの収集を開始/終了します。

備考 DMAC(Direct Memory Access Controller) / DTC(Data Transfer Controller)によるアクセスは 対象となりません。

- ソース・テキスト/逆アセンブル・テキスト上の変数 ///O レジスタに設定する場合

操作は, ソース・テキスト/逆アセンブル・テキストを表示しているエディタ パネル/逆アセンブル パネル 上で行います。

ソース・テキスト/逆アセンブル・テキスト上の任意の変数,または I/O レジスタを選択したのち,コンテキ スト・メニューより次の操作を行います。ただし,対象となる変数は,グローバル変数/関数内スタティック 変数/ファイル内スタティック変数のみとなります。

なお、この操作を行うことにより、対象変数、または I/O レジスタにトレース・イベント(アクセス系)が設定されたとみなされ、イベントパネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法
リード/ライト	- トレース開始イベント [トレースの設定]→[読み書き時にトレース開始の設定]を選択したのち, [Enter]キーを押下します。
	 トレース終了イベント トレースの設定]→ [読み書き時にトレース終了の設定]を選択したのち、 [Enter] キーを押下します。 この際に、メニュー内のテキスト・ボックスに値を指定した場合、指定した値で読み込み、または書き込みを行った場合のみトレース・データの収集を開始/終了します。 値を指定しない場合は、値にかかわらず、選択している変数に読み込み、または書き込みを行った場合にトレース・データの収集を開始/終了します。

注意 1. カレント・スコープ内の変数が対象となります。

- 注意 2. イベント・エリアにおいて背景色がグレー表示の行(その行が対応するアドレスに変換不可であることを示す)上の変数 ///O レジスタを選択してもトレース・イベントを設定することはできません。このトレース・イベントを設定する際は、イベント・エリアの背景色が白色の行にある変数 ///O レジスタを選択してください。
- 注意 3. ソース・テキスト/逆アセンブル・テキスト上から設定可能なアクセス種別は、リード/ライトのみです。アクセス種別をリードおよびライトへ変更する場合は、トレース開始/終了イベントを設定したのち、イベントパネル上からオープンするダイアログで編集します。 (「2.13.3.4 トレース開始イベント/トレース終了イベントを編集する」参照)

- 登録したウォッチ式に設定する場合

Г

操作は、ウォッチパネル上で行います。

対象となるウォッチ式を選択したのち(複数選択不可),コンテキスト・メニューより次の操作を行います。 ただし、対象となるウォッチ式は、グローバル変数/関数内スタティック変数/ファイル内スタティック変数 ///Oレジスタのみとなります。

なお、この操作を行うことにより、対象ウォッチ式にトレース・イベント(アクセス系)が設定されたとみな され、イベントパネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法
リード/ライト	- トレース開始イベント [トレース出力]→[読み書き時にトレース開始の設定]を選択したのち, [Enter] キーを押下します。
	- トレース終了イベント [トレース出力] → [読み書き時にトレース終了の設定]を選択したのち, [Enter] キーを押下します。 この際に,メニュー内のテキスト・ボックスに値を指定した場合,指定された値で読 み込み,または書き込みを行った場合のみトレース・データの収集を開始/終了しま
	す。 値を指定しない場合は,値に関わらず,選択しているウォッチ式に読み込み,または 書き込みを行った場合にトレース・データの収集を開始/終了します。

注意 1. カレント・スコープ内のウォッチ式が対象となります。

カレント・スコープ外のウォッチ式を対象とする場合は、スコープ指定したウォッチ式を選択 してください。

注意 2. ウォッチ式から設定可能なアクセス種別は、リード/ライトのみです。アクセス種別をリード およびライトへ変更する場合は、トレース開始/終了イベントを設定したのち、イベントパネ ル上からオープンするダイアログから編集します。(「2.13.3.4 トレース開始イベント/トレー ス終了イベントを編集する」参照)

トレース開始イベント/トレース終了イベントが設定されると、イベントパネル上において、トレース・イベントとして1つにまとめて管理されます(トレース・イベント項目の "+" マークをクリックすることにより、設定したトレース開始イベント/トレース終了イベントの詳細情報が表示されます)。

注意 1. トレース開始イベント/トレース終了イベントの両方を設定した場合,開始および終了条件が成立するたびにトレース・データを収集します。この場合,トレース・パネルで正しく逆アセンブリ表示できません。

注意 3. 【E20(JTAG)【RX600, RX700 シリーズ】】
 [トレース]カテゴリの[トレース機能の用途]プロパティにおいて、リアルタイム RAM モニタを指定した場合、トレース開始/終了イベントを使用しないでください。また、イベント パネルにおいて、設定されたイベントは削除してください。

- 備考 1. トレース開始イベント/トレース終了イベントのいずれかが有効状態で設定されると、イベントパネ ル上の無条件トレース・イベントのチェックが自動的に外れ、設定されたトレース開始イベント成立か らトレース終了イベント成立までのトレース・データが収集されます。
- 備考 2. トレース開始イベントが不要な場合は、未設定でもかまいません。未設定の場合、プログラム実行開始 がトレース・データの収集開始となります。同様にトレース終了イベントが未設定の場合、プログラム 実行停止がトレース・データの収集停止となります。
- 備考 3. イベントの設定状態によりイベント・マークは異なります(「2.17.1 設定状態(有効/無効)を変更する」参照)。 また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが 設定されていることを示すイベント・マーク(FP)が表示されます。
- 備考 4. 【シミュレータ】 トレース開始イベント/トレース終了イベントのいずれかが有効状態で設定されると、プロパティパ ネルの[デバッグ・ツール設定]タブ上の[トレース]カテゴリ内[トレース機能を使用する]プロパ ティの指定を自動的に[はい]に変更し、トレース機能が有効化されます。

2.13.3.2 複数のイベントを組み合わせる【E1】 【E20】 【EZ Emulator】

トレース開始イベントに対して2つ以上のイベントを設定した場合、トレース開始イベントが次のような組み合わせ 条件を満たした時点でトレース・データの収集を開始させることができます。

また、トレース終了イベントに対して2つ以上のイベントを設定した場合、トレース終了イベントのいずれかひとつでも条件が成立した時点でトレース・データの収集を終了します。

表 2.13 トレース開始イベントに対する組み合わせ条件

組み合わせ条件	説明
OR	設定されたトレース開始イベントの内, いずれかひとつでも条件が成立した時点でトレー ス・データの収集を開始します。
AND	時間軸に関係なく,設定された全てのトレース開始イベントの条件が成立した時点でトレース・データの収集を開始します。
シーケンシャル	指定した順序で設定されたトレース開始イベントの条件が成立するとトレース・データの収 集を開始します。 また、トレース開始イベント内のイベントを1点のみリセットイベント(Rイベント)とし て登録することができます。 登録されたイベントが成立した場合、その時点まで成立していた他のトレース開始イベント の条件成立は全てクリアされます。

なお,設定したトレース・イベントを編集する場合は, [表示] メニュー→ [イベント] の選択でオープンするイベン ト パネル上において,トレース・イベントを選択したのち,コンテキスト・メニューの「条件の編集…」をクリックし てください。

この操作によりオープンするダイアログで編集します。

ダイアログによる編集の詳細については、「2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】」を参照してください。

注意 イベントの組み合わせに関しては、「(1) 組み合わせ条件の編集」の注意も参照してください。

2.13.3.3 プログラムを実行する

プログラムを実行します(「2.9 プログラムの実行」参照)。

トレース開始イベント/トレース終了イベントに設定されている条件が成立した時点で、トレース・データの収集を 開始/終了します。

なお, 収集したトレース・データの確認方法についての詳細は, 「2.13.6 実行履歴を表示する」を参照してください。
2.13.3.4 トレース開始イベント/トレース終了イベントを編集する

設定したトレース開始イベント/トレース終了イベントを編集する場合は、[表示]メニュー→ [イベント] の選択で オープンするイベント パネル上において、トレースの詳細情報に表示されたトレース開始/終了イベント(実行系およ びアクセス系)を選択したのち、コンテキスト・メニューの「条件の編集…」をクリックしてください。

この操作によりオープンするダイアログで編集します。

実行系イベントダイアログによる編集の詳細については、「2.17.4.1 実行系イベントを編集する」を参照してください。

また,アクセス系イベントダイアログによる編集の詳細については,「2.17.4.2 アクセス系イベントを編集する」を 参照してください。

2.13.3.5 トレース開始イベント/トレース終了イベントを削除する

設定したトレース開始イベント/トレース終了イベントを削除する場合は、イベント・エリア上のイベント・マーク を右クリックすることで表示されるコンテキスト・メニューの[イベント削除]を選択します。

また, [表示] メニュー→ [イベント] の選択でオープンする イベント パネル上において, トレースの詳細情報に表示されたトレース開始/終了イベント(実行系およびアクセス系)を選択したのち, ツールバーのズボタンをクリックする操作でも削除することができます。(「2.17.5 イベントを削除する」参照)。

注意 イベント・エリア上のイベント・マークからトレース開始/終了イベントのいずれかを削除した場合, 対応した全てのイベント・マークが削除されます。

2.13.4 条件を満たしたときのみの実行履歴を収集する

ある条件を満たしたときのみプログラムの実行履歴を収集することができます。

- (1) 変数 /I/O レジスタへのアクセスが発生したとき ポイント・トレース・イベントを設定することにより、任意の変数、または I/O レジスタに対し、指定したアク セスがあった場合にのみ、その情報をトレース・データとして収集します。 次のいずれかの操作により、ポイント・トレース・イベントを設定してください。
 - **注意 1.** ポイント・トレース・イベントの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する留意事項」も参照してください。
 - **注意 2.** 【シミュレータ】 ストリング操作命令,および積和演算命令は,最初と最後のデータアクセスのみがイベント チェック対象となります。
 - 備考 1. DMAC (Direct Memory Access Controller) / DTC (Data Transfer Controller) によるアクセスは 対象となりません。
 - 備考 2. 【シミュレータ】 ポイント・トレース・イベントのいずれかが有効状態で設定されると、プロパティ パネルの[デ バッグ・ツール設定]タブ上の[トレース]カテゴリ内[トレース機能を使用する]プロパティ の指定を自動的に[はい]に変更し、トレース機能が有効化されます。
 - 備考 3. ポイント・トレース・イベントとトレース開始/終了イベントを同時に設定した場合は、デバッグ・ツールにより動作が異なります。
 - 【E1】【E20】【EZ Emulator】 トレース開始/終了イベントで設定した区間内でポイント・トレース・イベントの条件に一致 したデータのみをトレース・データとして収集します。

-【シミュレータ】 トレース開始/終了イベントで設定した区間内の実行履歴、およびトレース開始/終了イベン トで設定した区間とは関係なくポイント・トレース・イベントの条件に一致したデータをト レース・データとして収集します。

 (a) ソース・テキスト/逆アセンブル・テキスト上の変数 /I/O レジスタへのアクセスの場合 操作は、ソース・テキスト/逆アセンブル・テキストを表示しているエディタ パネル/逆アセンブル パネル 上で行います。
 対象となる変数、または I/O レジスタを選択したのち、指定するアクセス種別にしたがって、コンテキスト・ メニューより次の操作を行います。ただし、対象となる変数は、グローバル変数/関数内スタティック変数/ ファイル内スタティック変数のみとなります。
 なお、この操作を行うことにより、対象変数 /I/O レジスタにポイント・トレース・イベントが設定されたと みなされ、イベント パネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法	
リード	[トレース設定] → [値をトレースに記録 (読み込み時)] を選択します。	
ライト	[トレース設定]→[値をトレースに記録(書き込み時)]を選択します。	
リード/ライト	[トレース設定]→[値をトレースに記録(読み書き時)]を選択します。	

備考 カレント・スコープ内の変数が対象となります。

(b) 登録したウォッチ式へのアクセスの場合

操作は、ウォッチ パネル上で行います。 対象となるウォッチ式を選択したのち、コンテキスト・メニューより次の操作を行います(「2.11.6 ウォッチ 式を表示/変更する」参照)。

ただし、対象となるウォッチ式は、グローバル変数/関数内スタティック変数/ファイル内スタティック変数 /I/O レジスタのみとなります。

なお、この操作を行うことにより、対象ウォッチ式にポイント・トレース・イベントが設定されたとみなされ、イベントパネルで管理されます(「2.17 イベントの管理」参照)。

アクセス種別	操作方法
リード	[トレース出力] → [値をトレースに記録 (読み込み時)] を選択します。
ライト	[トレース出力] → [値をトレースに記録(書き込み時)] を選択します。
リード/ライト	[トレース出力] → [値をトレースに記録 (読み書き時)] を選択します。

備考 カレント・スコープ内のウォッチ式が対象となります。

カレント・スコープ以外のウォッチ式を対象とする場合は、スコープ指定したウォッチ式を選 択してください。

ポイント・トレース・イベントの設定が完了したのち、プログラムを実行します(「2.9 プログラムの実行」参照)。

プログラム実行中,設定したポイント・トレース・イベントの条件が満たされた場合,その情報がトレース・ データとして収集されます。トレース・データの確認方法についての詳細は,「2.13.6 実行履歴を表示する」を 参照してください。

なお、ポイント・トレース・イベントとして設定したトレース・イベント(アクセス系)を編集する場合は、 [表示] メニュー→ [イベント] の選択でオープンするイベント パネルにおいて、編集したいポイント・トレー スの詳細情報に表示されたトレース・イベント(アクセス系)を選択したのち、コンテキスト・メニューの [条 件の編集 ...] をクリックします(「2.17.4.2 アクセス系イベントを編集する」参照)。

また,ポイント・トレース・イベントとして設定したトレース・イベント(アクセス系)を削除する場合は, [表示] メニュー→ [イベント] の選択でオープンするイベントパネルにおいて,削除したいポイント・トレー ス・イベントの詳細情報に表示されたトレース・イベント(アクセス系)を選択したのち,同パネルのツール バーの ★ボタンをクリックします(「2.17 イベントの管理」参照)。

2.13.5 実行履歴の収集を停止/再開する【E20】 【シミュレータ】

プログラム実行中に実行履歴の収集を一時的に停止、および再開することができます。

- 注意 1. プログラム実行中にトレース機能を停止/再開する場合、リアルタイム性を損なう可能性があります。
- 注意 2. トレース開始イベント,またはトレース終了イベントを設定した場合,この機能は使用できません。
- **注意 3.** 【E1】【E20【RX200 シリーズ】】【EZ Emulator】 プログラム実行中にトレース機能を停止/再開する機能はサポートしていません。

注意 4. 【E20(JTAG)【RX600, RX700 シリーズ】】 プロパティ パネルの [デバッグ・ツール設定] タブにおける [トレース] カテゴリ内の [外部トレース 出力] プロパティで [外部出力しない]を選択した場合, この機能は使用できません。使用する場合は, [CPU 実行優先] または [トレース出力優先]を選択してください。

注意 5. 【E20(JTAG)【RX600, RX700 シリーズ】】 プロパティ パネルの[デバッグ・ツール設定]タブにおける[トレース]カテゴリ内の[トレース機能 の用途]プロパティで[リアルタイム RAM モニタ]を選択した場合,この機能は使用できません。使 用する場合は[トレース]を選択してください。

注意 6. アクション・イベントを設定した場合、この機能は使用できません。

2.13.5.1 実行履歴の収集を一時的に停止する場合

プログラム実行中にトレースパネル上のコンテキスト・メニュー→[トレース停止]を選択することにより、トレース・データの収集を一時的に停止することができます。

プログラムを停止せずにトレース機能のみを停止させ、その時点までのトレース・データを確認する場合などに使用 します。

2.13.5.2 実行履歴の収集を再開する場合

プログラム実行中にトレース機能を停止した場合、トレースパネル上のコンテキスト・メニュー→ [トレース開始] を選択することにより、再度トレース・データの収集を開始することができます。 なお、再開前に収集したトレースパネルのトレース・データは一度クリアされます。

2.13.6 実行履歴を表示する

収集したトレース・データの表示は、次のトレースパネルで行います。 [表示]メニュー→[トレース]を選択してください。 トレース・データは、デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが、表示モー ドを選択することにより、そのどちらか一方のみを表示させることもできます。

なお,各エリアの見方,および機能についての詳細は、トレースパネルの項を参照してください。

図 2.145 トレース・データの表示(トレース パネル)

香号	時間 (htmints,ms,µs,ns)	行番号/アドレス	ソース/逆アセンブル	アドレス データ
548-4	00h00win00s000ws297us281ns	0xffff864d	+127 CMP \$1AH, R1	
548-5		sort.c#35	for(i=ktgapt i<10; i=itgap)[
548-5	00h00win00s000ws287us281ns	0xffff864f	+128 BLT.B _sort+1AH	
548	00h00win00s000ws287µs280ns			0x00001710 R 16927
550	00h00min00s000ms300us800ns			0x000018e4 R 5
551	00h00win00s000ws001us120ns			0x00001660 <mark>R 5</mark>
552	00h00win00s000ws302us081ns			0x100016e4 <u>9 10</u>
553	00h00win00s000ws302µs401ns			0x000018e4 R 10
554		sort.c#35	for(i=ktgap; i<10; i=itgap){	
554	00h00win00s000ws002us720ns	0xffffl651	+101 MOV.L OCHER01.R1	
554-1		sort.cl35	<pre>for(i*k+papt i<10; i*i+pap){</pre>	
554-1	00h00win00s000ws302µs720ns	Dxffff8653	+138 400 \$1H,RI	
554-2		sort.c#35	for(i=ktgapt i<10; i=itgap){	
554-2	00h00win00s000ws302us720ns	0xffff8655	+135 MOY.L R1.0CH[R0]	
554-3		sort.c#84	for(k=0; k <psp; k++){<="" td=""><td></td></psp;>	
			1	h

ツールバー

ここでは、次の操作方法について説明します。

2.13.6.1 表示モードを変更する 2.13.6.2 値の表示形式を変更する 2.13.6.3 他のパネルと連動させる

2.13.6.1 表示モードを変更する

次のツールバーのボタンをクリックすることで、用途に応じて表示モードを変更することができます。 ただし、トレース機能が動作中の場合は無効となります。

表 2.14 トレース パネルの表示モード

ボタン	表示モード	表示内容
医 管 第11	混合表示モード	命令(逆アセンブル)/ラベル名/ソース・テキスト(対応するソース行)/ポイ ント・トレース結果を表示します(デフォルト)。

RENESAS

ボタン	表示モード	表示内容
<u>ت</u>	逆アセンブル表 示モード	命令(逆アセンブル)/ラベル名/ポイント・トレース結果を表示します。
1	ソース表示モー ド	ソース・テキスト(対応するソース行)を表示します。 ただし、デバッグ情報が存在しない箇所を実行した場合は、"デバッグ情報のない 区間の実行"と表示します。

図 2.146 ソース表示モードの例(トレース パネル)

<u>ት</u> /-አ	×
🛛 💱 🛞 🛞 品 表記(N)+ 🞵 🏂 観 🌯 🗹	
「番号 時間(h:min:s,ms,µs,ns) 行番号/アドレス ソース/逆アセンブル	· •
1252-5 sort.c#62 a[i] = tmp[8 - i];	
1252-6 sort.c#62 a[i] = tmp[8 - i];	
1252-7 sort.c#82 a[i] = tmp[9 - i];	
1252-8 sort.c062 a[i] = tmp[9 - i];	
1252-9 sort.c#62 a[i] = tmp[8 - i];	
1252-10 sort.c#61 for(i=0; i<10; i++){	
1252-11 sort.c#81 for(i=0; i<10; i++){	
1252-12 sort.c#61 for(i=0; i<10; i++){	
1260 sort.c#64 }	
1263 Tutorial.c#71 p_sam->s0=a[0];	
1263-1 Tutorial.c#71 p_sam->s0=a[0];	
1263-2 Tutorial.cll71 p_sam->s0=a[0];	
1263-3 Tutorial.c#72 p_sam->sl=a[1];	
1263-4 Tutorial.c#72 p_sam->s1=a[1];	
1263-5 Tutorial.c#72 p_sam->s1=a[1];	
1263-6 Tutorial.c#73 p_sam->s2=a[2];	-
	F .

2.13.6.2 値の表示形式を変更する

[行番号 / アドレス] エリア/ [アドレス] エリア/ [データ] エリアの表示形式は、ツールバーの次のボタンにより、自由に変更することができます。 ただし、プログラム実行中は、ボタンは無効となります。

	にたし、プログラム夫打中は、小グブは無効となりより。		
	表記 値の表示形式を変更する次のボタンを表示します。		
-		このパネル上の値を 16 進数で表示します(デフォルト)。	
		このパネル上の値を 10 進数で表示します。	
	Oet	このパネル上の値を8進数で表示します。	
	Bin	このパネル上の値を2進数で表示します。	

2.13.6.3 他のパネルと連動させる

現在選択している行のアドレスをポインタとして、他のパネルで対応箇所を連動して表示させることができます (フォーカスの移動は行いません)。

ツールバーの ゴボタンをクリックすると、エディタ パネルと連動開始します。 ツールバーの ボタンをクリックすると、逆アセンブル パネルと連動開始します。 なお、再度クリックすることにより、連動を中止します。

備考 コンテキスト・メニューの [ソースヘジャンプ] / [逆アセンブルヘジャンプ] を選択することにより,現在選択している行のアドレスに対応するソース行/アドレスにキャレットを移動した状態で,エ ディタ パネル/逆アセンブル パネルがオープンします (フォーカスの移動を行います)。

図 2.147

2.13.7 トレース・メモリをクリアする

収集したトレース・データの内容をクリアするには、ツールバーの 🛂 ボタンをクリックします。 ただし、トレース機能が動作中は無効となります。

備考 【シミュレータ】 プロパティ パネルの [デバッグ・ツール設定] タブ上の [トレース] カテゴリ内の [実行前にトレー ス・メモリをクリアする] プロパティにおいて, [はい] を指定している場合は, プログラムの実行ご とにトレース・メモリがクリアされます。

2.13.8 トレース・データを検索する

収集したトレース・データの検索は、ツールバーの 離 ボタンをクリックすることによりオープンするトレース検索 ダイアログにより行います(プログラム実行中は無効)。

このダイアログにおいて、次の操作を行ってください。

トレース・データの検索(トレース検索ダイアログ)

なお、タブ選択エリア上のタブを選択することにより、命令レベル、またはソース・レベルでトレース・データを検 索することができます。

ただし、命令レベルでトレース・データの検索を行う場合は、トレースパネルを混合表示モード、または逆アセンブ ル表示モードで表示している必要があります。

また、ソース・レベルで検索を行う場合は、混合表示モード、またはソース表示モードで表示している必要があります。

検索条件の指定 フェッチ・アドレス(工):		-	(範囲を指定する場合 ▶▶▼
命令(<u>M</u>): アクセス・アドレス(<u>A</u>):		-	(範囲を指定する場合 📦 👻
アクセスの種類(S) データ(D):	(指定なし) MEX	v v	▶★ (範囲を指定する ▶ ▼
検索範囲の指定 番号(U):		-	

ここでは、次の操作方法について説明します。

2.13.8.1 命令レベルで検索する 2.13.8.2 ソース・レベルで検索する

2.13.8.1 命令レベルで検索する

命令レベルでトレース・データを検索します。 [命令レベル] タブを選択したのち,次の手順で操作を行ってください。

図 2.148 命令レベルでのトレース・データの検索

トレース検索					
命令レベル ソース・レベ	١L				
検索条件の指定					
フェッチ・アドレス(工):		- (筆記	回を指定する場合	÷ 10 -	
命令(M)		-			
アクセス・アドレス(<u>A</u>):		- ())55	囲を指定する場合	ê 10 v	
アクセスの種類(S)	(指定なし)				
データ(D):	HEX	- HEX	(範囲を指定する	5 10 -	
検索範囲の指定					
番号(<u>U)</u> :		-		-	
	前を検索(日)	次を検索(E)	キャンセル	~117(H)	
ます (最大履歴数:1 フェッチ・アドレス0 ドレス値を指定するこ 右側のテキスト・ボッ クスに指定された固定 なお,マイクロコント マスクして扱います。 また,32 ビットで表	0個)。)指定は範囲で指定す ことにより範囲を指定 ックスが空欄,またに ミアドレスで検索を行 、ローラのアドレスS 現できる値より大き	することができます 定します。 は"(範囲を指定す 行います。 空間よりも大きいア いアドレス値を指5	⁻ 。この場合は る場合に入力) 'ドレス値が指 定することはて	, 左右両方の "の場合は, 定された場合 ⁵ きません。	ラテキスト・ボックスにフ 左側のテキスト・ボッ ht, 上位のアドレス値を
[命令]の指定 検索条件として必要な ここで指定した文字列 命令をテキスト・ボッ (最大履歴数:10個) なお、検索の際は、ブ	⊊場合,命令の文字 りをトレースパネル ックスに直接入力する 。 大文字/小文字は区別	列を指定します。 の[ソース / 逆アセ るか,またはドロッ 別せず,部分一致も	マンブル] エリ マダウン・リ •検索の対象と	ア内より検索 ストより入力 します。	きします。 履歴項目を選択します
[アクセス・アドレス 検索条件として必要な 16 進数でアドレス値 目を選択します(最大 アクセス・アドレスの ドレス値を指定するこ 右側のテキスト・ボッ クスによてたれた固定 なお、マイクロコンし	()の指定 、場合、アクセス・フ をテキスト・ボック 、履歴数:10個)。 つ指定は範囲で指定す ことにより範囲を指定 ックスが空欄、またに シクスが空欄、またに、 マーラのアドレス? ローラのアドレス? ローラのアドレス?	アドレスを指定しま スに直接入力するが することができます 定します。 よ"(範囲を指定す 行います。 空間よりも大きいア	ます。 か、またはドロ つ。この場合は る場合に入力) アドレス値が指	 ・ップダウン・ ・ 左右両方の "の場合は、 定された場合 	・リストより入力履歴項 ラテキスト・ボックスにつ 左側のテキスト・ボッ な、上位のアドレス値を
[アクセスの種類]の この項目は[アクセラ アクセスの種類(リー 選択します。	ション こうにようへさ)指定 、・アドレス]の指定 -ド/ライト、リート	定が指定された場合 ド、ライト、ベクタ	こ,っここはし このみ有効とな マ・リード、DN	します。 ります。 MA)をドロッ	ップダウン・リストより

RENESAS

- 注意 【E1】【E20】【EZ Emulator】
 トレース パネル上で収集可能なアクセスの種類は、リード、ライトのみです。そのため、ドロップダウン・リストよりリード/ライト、ベクタ・リードおよび DMA を選択しないでください。
 【シミュレータ】
 トレース パネル上で収集可能なアクセスの種類は、リード、ライト、ベクタ・リードです。そのため、ドロップダウン・リストよりリード/ライトおよび DMA を選択しないでください。
- (5) 「データ」の指定 この項目は「アクセス・アドレス」の指定が指定された場合のみ有効となります。 アクセスした数値を指定します。
 16 進数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10 個)。
 数値の指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスにデータを指定することにより範囲を指定します。
 右側のテキスト・ボックスが空欄、または"(範囲を指定する場合に入力)"の場合は、左側のテキスト・ボックスに指定された固定数値で検索を行います。
- (6) [番号]の指定
 検索するトレース・データの範囲を、トレースパネルの[番号]エリアに表示されている番号で指定します。
 左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します(デフォルトでは、"0"~"*最終番号*"が指定されます)。
 10進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10個)。
 左側のテキスト・ボックスが空欄の場合は、"0"の指定として扱われます。
 右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。
- (7) [前を検索] / [次を検索] ボタンのクリック
 [前を検索] ボタンをクリックすると、番号の小さい方向に検索を行い、検索結果箇所をトレース パネル上で選択状態にします。
 [次を検索] ボタンをクリックすると、番号の大きい方向に検索を行い、検索結果箇所をトレース パネル上で選択状態にします。
- 2.13.8.2 ソース・レベルで検索する

ソース・レベルでトレース・データを検索します。 [ソース・レベル] タブを選択してください。



図 2.149 ソース・レベルでのトレース・データの検索

レース検索	
命令レベル シース	- LASIL
検索対象の指定	
 リース行を指定 ・ 間約名を指定 	」て実行箇所を検索(Q) 、7先頭アドル2の実行箇所を検索(T)
◎ グローバル変数	名を指定してアクセス箇所を検索(③)
検索条件の指定	
ソース行(<u>S</u>):	
閲数:名(N):	v
変数名(⊻):	¥
種類(匠):	参照/代入
変数値(<u>A</u>):	
検索範囲の指定	
番号(<u>U)</u> :	
	前を検索(B) 次を検索(E) キャンセル ヘルプ(H)

(1) ソース行を指定して検索する場合(デフォルト)

[検索対象の指定] エリアにおいて, "ソース行を指定して実行箇所を検索"を選択したのち, 次の操作を行いま す。

- [ソース行] の指定

ここで指定した文字列をトレースパネルの[行番号/アドレス]エリア内より検索します。 検索するソース行に含まれる文字列を,テキスト・ボックスに直接入力するか,またはドロップダウン・リス トより入力履歴項目を選択します(最大履歴数:10個)。 なお,検索の際は,大文字/小文字は区別せず,部分一致も検索の対象とします。

- なの, 快系の际は, 人文子/小文子は区別です, 部方一致も快系の対象
- 例 1. main.c#40
- 例 2. main.c

例 3. main

- [番号] の指定

検索するトレース・データの範囲を、トレースパネルの [番号] エリアに表示されている番号で指定します。 左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します(デフォルトでは、"0" ~ " *最終番 号*"が指定されます)。 10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を 選択します(最大履歴数:10 個)。 左側のテキスト・ボックスが空欄の場合は、"0"の指定として扱われます。

右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。

- [前を検索] // [次を検索] ボタンのクリック [前を検索] ボタンをクリックすると、番号の小さい方向に検索を行い、検索結果箇所をトレース パネル上で 選択状態にします。 [次を検索] ボタンをクリックすると、番号の大きい方向に検索を行い、検索結果箇所をトレース パネル上で 選択状態にします。
- (2) 関数名を指定して検索する場合 [検索対象の指定]エリアにおいて、"関数名を指定して先頭アドレスの実行箇所を検索"を選択したのち、次の 操作を行います。
 - [関数名]の指定 検索する関数名を、テキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目 を選択します(最大履歴数:10個)。

なお、検索の際は、大文字/小文字を区別し、完全一致のみを検索の対象とします。 「番号」の指定 検索するトレース・データの範囲を、トレースパネルの「番号」エリアに表示されている番号で指定します。 左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します(デフォルトでは、"0"~"最終番 号"が指定されます)。 10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を 選択します(最大履歴数:10 個)。 左側のテキスト・ボックスが空欄の場合は、"0"の指定として扱われます。 右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。 - [前を検索] / [次を検索] ボタンのクリック [前を検索] ボタンをクリックすると, 番号の小さい方向に検索を行い, 検索結果箇所をトレース パネル上で 選択状態にします。 [次を検索]ボタンをクリックすると,番号の大きい方向に検索を行い,検索結果箇所をトレース パネル上で 選択状態にします。 (3) グローバル変数名を指定して検索する場合 [検索対象の指定] エリアにおいて、" グローバル変数名を指定してアクセス箇所を検索"を選択したのち、次の 操作を行います。 [変数名]の指定 検索する変数名を、テキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目 を選択します(最大履歴数:10個)。 なお、検索の際は、大文字/小文字を区別し、完全一致のみを検索の対象とします。
 [種類]の指定
 アクセスの種類(参照/代入(デフォルト)、参照、代入)をドロップダウン・リストより選択します。 - [変数値] の指定 アクセスした変数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項 目を選択します(最大履歴数:10個)。 変数値の指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスに変数値を指定 することにより範囲を指定します。 右側のテキスト・ボックスが空欄の場合は、左側のテキスト・ボックスに指定された固定変数値でアクセス箇 所を検索を行います。 - [番号] の指定 検索するトレース・データの範囲を、トレース パネルの [番号] エリアに表示されている番号で指定します。 左右のテキスト・ボックスに,それぞれ開始番号と終了番号を指定します(デフォルトでは,"0" 〜 " *最終番* 号"が指定されます)。 10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を 選択します(最大履歴数:10個)。 左側のテキスト・ボックスが空欄の場合は、"0"の指定として扱われます。 右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。 - [前を検索] / [次を検索] ボタンのクリック [前を検索] ボタンをクリックすると、番号の小さい方向に検索を行い、検索結果箇所をトレース パネル上で 選択状態にします。 [次を検索] ボタンをクリックすると、番号の大きい方向に検索を行い、検索結果箇所をトレース パネル上で 選択状態にします。

2.13.9 実行履歴の表示内容を保存する

収集したトレース・データの内容を範囲指定して、テキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存することができます。ファイルに保存する際は、デバッグ・ツールから最新の情報を取得し、このパネル上での表示形式に 従ったデータで保存します。

[ファイル] メニュー→ [名前を付けてトレース・データを保存 ...]を選択すると、次のデータ保存 ダイアログが オープンします。

このダイアログにおいて、次の手順で操作を行ってください。

図 2.150 実行履歴の保存 (データ保存 ダイアログ)

データ保存・トレ	ース・データ
ファイル名(N):	C:¥Test¥sample¥トレース・データ
ファイルの種類(工): - 保存範囲 番号(E	「テキスト・ファイル (*.txt) ▼):
4088	• - 4094 •
	保存(S) キャンセル ヘルプ(H)

 (1) [ファイル名]の指定 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより、ファイル を選択することもできます。
 (2) [ファイルの種類]の指定

保存するファイルの形式を次のドロップダウン・リストにより選択します。 選択できるファイルの形式は次のとおりです。

リスト表示	形式
テキスト・ファイル (*.txt)	テキスト形式(デフォルト)
CSV(カンマ区切り)(*.csv)	CSV 形式 ^注

注 各データを "," で区切り保存します。 なお、データ内に "," が含まれている際の不正形式を避けるため、各データを """"(ダブルクォー テーション)で括り出力します。

(3) [保存範囲 番号]の指定

ファイルに保存する範囲を"開始トレース番号"と"終了トレース番号"で指定します。 それぞれのテキスト・ボックスに10進数の数値を直接入力するか、またはドロップダウン・リストより入力履 歴項目を選択します(最大履歴数:10個)。 なお、すべてのトレース・データを保存する場合は、左側のドロップダウン・リストにおいて、[すべてのト レース・データ]を選択してください(右側のテキスト・ボックスは無効)。 パネル上で範囲選択している場合は、デフォルトでその選択範囲がテキスト・ボックスに指定されます。範囲選 択していない場合は、現在のパネルの表示範囲が指定されます。

(4) [保存] ボタンのクリック 指定したファイルに,指定した形式でトレース・データを保存します。

図 2.151 トレース・データ保存の際の出力イメージ

番号	時間	行番号 / アドレス	ソース / 逆アセンブル	アドレス	データ
<i>番号</i> :	<i>時間</i>	<i>行番号 / アドレス</i>	<i>ソース / 逆アセンブル</i>	<i>アドレス</i>	データ
	:	:	:	:	:

2.14 実行時間の計測

この節では、プログラムの実行時間の計測方法について説明します。

注意 1. 【EZ Emulator【RX200 シリーズ】】 端子リセットを入れた場合, EZ Emulator からの再実行時に実行時間はクリアされます。

注意 2. 【E1/E20/EZ Emulator【RX600 シリーズ】】 割り込み / 例外発生回数と RTE/RTFI 命令の発行回数が同じで、それぞれ対応付けが取れる場合である ことを前提としています。もし、割り込み / 例外発生回数と RTE/RTFI 命令の発生回数が異なる場合、 それぞれの対応付けが取れないため、正しく処理サイクルを計測することができません。 なお、ネストは 16 回分までしか保持できません。したがって、17 回以上の割り込み・例外でネストが 発生した場合、正しく計測ができません。

2.14.1 タイマ計測動作の設定をする【E1】【E20】【EZ Emulator】

タイマ計測機能を行うためには、あらかじめ次のタイマ計測に関する設定を行う必要があります。 設定は、プロパティパネルの[接続用設定]タブ上の[クロック]カテゴリ内の[動作周波数[MHz]【E1】【E20】 【EZ Emulator】] プロパティ、およびプロパティパネルの[デバッグ・ツール設定]タブ上の[タイマ]【E1】【E20】 【EZ Emulator】カテゴリ内で行います。

注意 【RX140 グループ以外の RX100 シリーズ】 タイマ機能は未サポートとなります。

図 2.152 [クロック]カテゴリ

カウンタを64ビットで使用する

2092	
メイン・クロック・ソース	EXTAL
メイン・クロック周波数[MHz]	12.5000
動作問波数[MHz]	25,0000
内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	いいえ

図 2.153 [タイマ] カテゴリ【RX600, RX700 シリーズ】

*▲ タ*イマ

いいえ

(1) [動作周波数 [MHz]]

カウント値を時間に変換する際に使用する動作周波数を指定します。 0.0001 ~ 999.999(単位: MHz)の範囲の数値を直接入力で指定します。

- **注意 1.** タイマ計測イベントの計測結果は、このプロパティで入力した動作周波数とカウント値より算出 されます。そのため、実行中に動作周波数を切り替えるプログラムを用いた場合、正しい計測結 果は表示されません。
- **注意 2.** タイマ測定中にリセットが入った場合は、リセット期間中も計測が行われます。但し、計測用の クロック設定も初期化されるため正しい計測が行えない場合があります。
- (2) [カウンタを 64 ビットで使用する]【RX600, RX700 シリーズ】
 計測カウンタを 32 ビット ×2 本で使用するか、64 ビット ×1 本で使用するかを指定します。
 [はい]を指定した場合、64 ビットの計測カウンタが使用可能ですが、計測区間は1区間となります。
 - **注意** 【RX200 シリーズ, RX140 グループ】 計測カウンタは 24 ビット ×1 本のため、このプロパティは表示されません。

2.14.2 実行開始から停止までの実行時間を計測する

デバッグ・ツールには、プログラムの実行開始から実行停止までの実行時間(Run-Beak 時間)を測定する機能があらかじめ用意されています。

したがって、プログラムの実行を開始することにより、自動的に実行時間の計測を行います。 計測結果は、次のいずれかの方法で確認することができます。 ここでは、次の確認方法について説明します。

2.14.2.1 ステータス・バーでの確認 2.14.2.2 イベントパネルでの確認

- **注意 1.** Run-Break タイマ・イベントの計測結果は、計測開始からプログラム実行までの時間、および、プログラム実行停止から計測停止までの時間を含みます。また、計測クロックソースのクロック精度誤差を含む参考値です。
- 注意 2. ステータス・バーおよびイベント パネルに表示される実行時間は、プログラム実行から停止の間の時間です。なお、ステップ・イン、ステップ・オーバー、およびリターン・アウトでは計測できません。
 【E1】【E20】
 100us 未満の値は切り捨てられます。
- **注意 3.** EZ Emulator では, Run-Break タイマ機能は未サポートとなります。
- 備考 この機能は、デバッグ・ツールにデフォルトで設定されているビルトイン・イベントの1つである Run-Break タイマ・イベントにより動作します。 Run-Break タイマ・イベントは、常に有効状態に設定されているイベントです(変更不可)。
- 2.14.2.1 ステータス・バーでの確認

プログラムの実行停止後,メイン・ウインドウ上のステータス・バーにおいて計測結果を表示します(計測をしていない場合は"未計測"と表示)。

備考 EZ Emulator では Run-Break タイマ機能をサポートしていないため,常に " 未計測 " と表示されます。

図 2.154 Run-Break タイマ・イベントの測定結果例(ステータス・バー)

BREAK	i Oxffff86cd	RX E20(JTAG)	200.000 µs	N
		このエリフ	アに計測結果を表示	します。

2.14.2.2 イベント パネルでの確認

プログラムの実行停止後, [表示] メニュー→ [イベント] の選択によりオープンするイベント パネル上において, Run-Break タイマ・イベントとして計測結果を表示します。

図 2.155 Run-Break タイマ・イベントの測定結果例(イベント パネル)【E1】【E20】

ተለጋት		×
× 🖲 🖲 🖬 🖏 🐼		
名前	□ 「「」「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」	コメント
🔽 警 無条件トレース	-	
🔽 警 Run-Breakタイマ	総実行時間:200000 r	18

図 2.156 Run-Break タイマ・イベントの測定結果例(イベント パネル)【シミュレータ】

140F	J
名前 ▽ 詳細情報 (1/) 「1/) 「1/) 「1/) 「1/) (2/) (2/) (2/) (2/) (2/) (2/) (2/) (2]
🔽 警 無条件トレース	
(📝 警 Run-Breakタイマ - 総実行時間: 138060710 ns 実行サイクル数:13806071 実行命令数:8014096)	

2.14.3 任意区間の実行時間を計測する

タイマ計測イベント(タイマ開始イベント/タイマ終了イベント)を設定することにより、プログラムの実行過程に おいて、任意の区間の実行時間を計測することができます。 この機能を使用するためには、次の手順で操作を行います。

2.14.3.1 タイマ開始イベント/タイマ終了イベントを設定する
2.14.3.2 プログラムを実行する
2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】
2.14.3.4 タイマ開始イベント/タイマ終了イベントを編集する
2.14.3.5 タイマ開始イベント/タイマ終了イベントを削除する

- **注意 1.** タイマ計測イベントの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する 留意事項」も参照してください。
- 注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX200 シリーズ】】 トレース パネルのタイム・スタンプ情報は、タイマ計測用カウンタを用いて実現しています。そのため、イベント パネルにてタイマ計測イベントを設定した場合、期待したタイム・スタンプ情報が表示されません。

 注意 3. 【E1】【E20】【EZ Emulator】 タイマ開始/終了イベントの追加/削除,またはタイマ計測イベントの編集を行った場合,次回計測時の計測結果は累積されません。
 【シミュレータ】 タイマ開始/終了イベントの追加/削除/編集を行った場合,次回計測時の計測結果は累積されません。

2.14.3.1 タイマ開始イベント/タイマ終了イベントを設定する

エディタ パネル/逆アセンブル パネル/ウォッチ パネルにおいて,タイマ計測を開始/終了するイベントを設定します。

(1) タイマ開始イベントの設定方法

 エディタパネル/逆アセンブルパネルから設定する場合 タイマ計測を開始したい行/アドレス^{注1}にキャレットを移動したのち、コンテキスト・メニューの[タイマ 設定]→[実行時にタイマ開始]→[タイマ*^{注2}に設定]を選択します。 タイマ開始イベントが、キャレット位置の行/アドレスに対応する先頭アドレスの命令に対して設定されま す。 タイマ計測を開始したい変数にキャレットを移動したのち、コンテキスト・メニューの[タイマ設定]→[読 み書き時にタイマ開始]→[タイマ*^{注2}に設定]を選択します。 タイマ開始イベントが、キャレット位置の変数に対して設定されます。

- ウォッチ パネルから設定する場合 タイマ計測を開始したいウォッチ式^{注3}にキャレットを移動したのち、コンテキスト・メニューの[タイマ設 定]→[読み書き時にタイマ開始]→[タイマ*^{注2}に設定]を選択します。 タイマ開始イベントが、ウォッチ パネルのウォッチ式に対して設定されます。

(2) タイマ終了イベントの設定方法

 エディタパネル/逆アセンブルパネルから設定する場合 タイマ計測を終了したい行/アドレス^{注1}にキャレットを移動したのち、コンテキスト・メニューの[タイマ 設定]→[実行時にタイマ終了]→[タイマ*^{注2}に設定]を選択します。 タイマ終了イベントが、キャレット位置の行/アドレスに対応する先頭アドレスの命令に対して設定されま す。 タイマ計測を終了したい変数にキャレットを移動したのち、コンテキスト・メニューの[タイマ設定]→[読 み書き時にタイマ終了]→[タイマ*^{注2}に設定]を選択します。 タイマ終了イベントが、キャレット位置の変数に対して設定されます。

 ウォッチ パネルから設定する場合 タイマ計測を終了したいウォッチ式^{注3}にキャレットを移動したのち、コンテキスト・メニューの[タイマ設 定]→[読み書き時にタイマ終了]→[タイマ*^{注2}に設定]を選択します。 タイマ終了イベントが、ウォッチ パネルのウォッチ式に対して設定されます。

- 注 1. タイマ開始イベント/タイマ終了イベントは、アドレス表示がない行に設定することはできません。
- 注 2. 【E1】【E20】【EZ Emulator】

[タイマ*に設定]メニューの*には、タイマ計測区間の番号としてチャネル番号が表示されます。タ イマ開始イベント/タイマ終了イベントを1区間で設定する場合、同じチャネル番号を選択してくだ さい。

なお、選択可能なチャネル番号は、マイクロコントローラおよびプロパティ パネルの [デバッグ・ ツール設定] タブ上の [タイマ] 【E1】 【E20】 【EZ Emulator】 カテゴリの次のプロパティ設定により異 なります。

マイクロコントローラ	[カウンタを 64bit で 使用する] プロパティ	機能
RX600, RX700 シリー ズ	いいえ	[タイマ 1 に設定] および [タイマ 2 に設定] の 2 区間 (32bit) から指定可能
	はい	[タイマ1に設定]の1区間 (64bit) のみを指定可能
RX200 シリーズ	_	[タイマ1に設定]の1区間 (24bit) のみを指定可能

【シミュレータ】

1区間固定で区間指定がないため、[タイマ*に設定]メニューは表示されません。

- 注 3. 対象となるウォッチ式は、グローバル変数/関数内スタティック変数/ファイル内スタティック変数/ IO レジスタのみとなります。
- 注意 1. 【E1】【E20】【EZ Emulator】 タイマ開始/終了イベントのどちらか片方の設定でもタイマ計測できます。タイマ開始イベントのみを 設定した場合、プログラムの実行停止でタイマ計測は終了します。また、タイマ終了イベントのみを設 定した場合、プログラムの実行開始でタイマ計測を開始します。 【シミュレータ】 タイマ開始/終了イベントの両方の設定でタイマ計測が有効になります。そのため、タイマ開始/終了 イベントのどちらか片方の設定ではタイマ計測できません。

注意 2. 【シミュレータ】 タイマ開始/終了イベントはそれぞれ 1 点のみ設定可能です。複数のイベントを組み合わせることは できません。

図 2.157 ソース・テキスト上の変数に対するタイマ開始イベント(アクセス系)の設定例【E1】【E20】【EZ Emulator】



RENESAS



図 2.158 ソース・テキスト上の変数に対するタイマ開始イベント(アクセス系)の設定例【シミュレータ】









タイマ開始イベント/タイマ終了イベントが設定されると、該当行/アドレス/ウォッチ式のイベント・エリアに次のイベント・マークが表示されます。

また、イベントパネル上において、タイマ計測イベントとして1つにまとめて管理されます(タイマ計測イベント項目の "+" マークをクリックすることにより、設定したタイマ開始イベント/タイマ終了イベントの情報を確認することができます)。

表 2.15 タイマ開始イベント/タイマ終了のイベント・マーク

種別	イベント・マーク
タイマ開始	
タイマ終了	<u>19</u>

図 2.161 タイマ開始イベント/タイマ終了イベントの設定例(逆アセンブル パネルの場合)

	イベント・エリ	ア		
タイマ開始イベントが設定	rta			
されていることを示します。	77:		for(i=0;	i<1000;
	00420	+3 +a	16	HOVE
タイマ終了イベントが設定	00422	+b	17	HOVE
されていることを示します。	00423	+c	44e803	CNPV
	78:	τI	0609	NOP();

注意 【シミュレータ】

タイマ開始/終了イベントが同じ命令実行で成立した場合、計測状態によって次の動作となります。

- タイマ計測が開始されている場合
- タイマ計測を停止し、計測結果に反映します。
- タイマ計測が開始されていない場合
 タイマ計測を行いません。
- 備考 イベントの設定状態によりイベント・マークは異なります(「2.17.1 設定状態(有効/無効)を変更する」参照)。 また、すでにイベントが設定されている箇所で、新たにイベントを設定した場合は、複数のイベントが設定されていることを示すイベント・マーク([↓])が表示されます。
- 2.14.3.2 プログラムを実行する

プログラムを実行します(「2.9 プログラムの実行」参照)。 タイマ開始イベント/タイマ終了イベントが設定されている命令が実行された際に、タイマ計測を開始/終了します。 計測結果は、プログラムの実行停止後、[表示]メニュー→[イベント]の選択によりオープンするイベントパネルにおいて、タイマ計測イベントとして次のように確認することができます。

なお, このタイマ計測イベントは, タイマ開始イベント, またはタイマ終了イベントのいずれかが設定された場合に, イベント パネルでのみ表示されるイベント種別です。

図 2.162	タイマ計測イベント	(タイマ開始イベント	/タイマ終了イベント)	の測定結果例【E1】【E20】
---------	-----------	------------	-------------	-----------------

名前	詳細情報 ス	-		コメント
警 Run-Break夕	イマ 総実行時	朝間: 2197000000	ns	
139/マ計測1 総実行時間	総実行# パスカウント	湘間:0 ns 開始// 平均実行時間	終了の総数:2 最大実行時間	最小実行時間
0 ns	-	-	-	-
名前 (Ph/malibia	######################################	19 19	10	コメント
約94 Varim2 総実行時間	総美(194) パスカウント	4個:Uns 開始// 平均実行時間	除了00轮数:2 最大実行時間	最小実行時間
0 ns	-	-	-	-
詳細情報 開始 実行後 sc	rtc#16 0xffff85e3	00 - 0~1000 1	125	

図 2.163 タイマ計測イベント(タイマ開始イベント/タイマ終了イベント)の測定結果例【EZ Emulator】

名前	詳細情	R		-1.CXC
※無条件トレース (2)タイマ計測1 総実行時間	総実行部 パスカウント	- 湘間:0 ns 開始// 平均実行時間	終了の総数:2 最大実行時間	最小実行時間
0 ns	-	-	-	-
attailate 開始リード/ライト	g_IntBuf 0x1404	- 0×1407 == 0xt	>	
終了 実行後 sort.c	#14 0x11118693			



図 2.164 タイマ計測イベント(タイマ開始イベント/タイマ終了イベント)の測定結果例【シミュレータ】

名前	3李都個翁祥≣ ▽			ועאב
警無条件トレース			-	
(29タイマ計測)	総実行時間	:0ns 開始	/終了の総数:2	
総実行時間	パスカウント	平均実行	朝間 最大実行時間	最小実行時間
0 ns	0	0 ns	0 ns	0 ns
目羊糸用作表意用	2.2	-		· · · · · · · · · · · · · · · · · · ·
間始リードノライト	g IntBuf 0x142a	c - 0x142f ≕	= 0xb	
終了 夷行前 sort	c#17 0xffff85e5			
名前	T III BEAR			עצר
			- H. Zhan Wall and she	

2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】

開始イベント、および終了イベントを設定したタイマ計測イベントの詳細情報は、タイマ計測詳細設定 ダイアログ 【E1】【E20】【EZ Emulator】にて編集します。このダイアログは、イベント パネル上にて編集したいタイマ計測イベントを選択してコンテキスト・メニューの [条件の編集 ...]を選択することで表示します。

(1) 計測項目を編集する場合

タイマ計測詳細設定ダイアログでは、計測項目として次から指定できます。

計測項目	機能
経過サイクル ^注	指定区間の経過サイクル数 (ICLK) を計測します。
経過サイクル(スーパバイザモードサイクル)	スーパバイザモードにいる間の経過サイクル数 (ICLK) を計 測します。
例外・割り込み処理サイクル	例外処理を含む割り込み処理に要したサイクル数 (ICLK) を 計測します。
例外処理サイクル	例外処理に要したサイクル数 (ICLK) を計測します。
割り込み処理サイクル	割り込み処理に要したサイクル数 (ICLK) を計測します。
有効命令発行回数	実行完了した命令数を計測します。
例外・割り込み受け付け回数	例外処理を含む割り込みを受け付けた回数を計測します。
例外受付回数	例外処理が発生した回数を計測します。
割り込み受付回数	割り込みを受け付けた回数を計測します。

注 【RX200 シリーズ , RX140 グループ】

計測項目には、[経過サイクル]のみが表示されます。

- 備考 1. 経過サイクル,例外・割り込み処理サイクル,例外処理サイクル,割り込み処理サイクルを指定 した場合は,総実行時間を計測します。
- 備考 2. 有効命令発行回数,例外・割り込み受付回数,例外受付回数,割り込み受付回数を指定した場合 は、通過回数(パスカウント)を計測します。

備考 3. 平均実行時間,最大実行時間,最小実行時間は測定できません。

(2) 一度だけ計測を編集する場合

[一度だけ計測]プロパティを[はい]と指定することで指定区間を一度だけ計測してタイマ計測を終了します。 なお,指定区間を通過した合計値を計測する場合,[いいえ]を指定してください。

備考	【RX600, RX700 シリーズ】 以下の計測実施中にタイマ開始イベントおよびタイマ終了イベントが成立しても、一度も計測条 件が成立しなかった場合、計測結果は表示されません。 - 有効命令発行回数 - 例外・割り込み受け付け回数 - 例外受付回数
	- 割り込み受付回数

注意 【RX600, RX700 シリーズ】 [はい]を指定した場合,タイマ開始イベントは,イベントが2回成立するとタイマ終了イベント が成立していなくても計測を停止します。

2.14.3.4 タイマ開始イベント/タイマ終了イベントを編集する

設定したタイマ開始イベント/タイマ終了イベントを編集する場合は, [表示] メニュー→ [イベント] の選択でオー プンするイベント パネル上において, タイマ計測の詳細情報に表示されたタイマ開始/終了イベント(実行系およびア クセス系)を選択したのち, コンテキスト・メニューの「条件の編集…」をクリックしてください。

この操作によりオープンするダイアログで編集します。

実行系イベントダイアログによる編集の詳細については、「2.17.4.1 実行系イベントを編集する」を参照してください。

また,アクセス系イベントダイアログによる編集の詳細については,「2.17.4.2 アクセス系イベントを編集する」を 参照してください。

2.14.3.5 タイマ開始イベント/タイマ終了イベントを削除する

設定したタイマ開始イベント/タイマ終了イベントを削除する場合は、イベント・エリア上のイベント・マークを右 クリックすることで表示されるコンテキスト・メニューの[イベント削除]を選択します。

また、[表示] メニュー→ [イベント] の選択でオープンする イベント パネル上において、タイマ計測の詳細情報に 表示されたタイマ開始/終了イベント(実行系およびアクセス系)を選択したのち、ツールバーの ズボタンをクリッ クする操作でも削除することができます。(「2.17.5 イベントを削除する」参照)。

注意 イベント・エリア上のイベント・マークからタイマ開始/終了イベントのいずれかを削除した場合,対応した全てのイベント・マークが削除されます。

2.14.4 測定可能時間の範囲

Run-Break タイマ・イベント(「2.14.2 実行開始から停止までの実行時間を計測する」参照),またはタイマ計測イベント(「2.14.3 任意区間の実行時間を計測する」参照)によるタイマ計測の測定可能時間の範囲は次のとおりです。

デバッグ・ツール	Run-Break タイマ・イベント		タイマ計測イベント	
E1(Serial)/E1(JTAG)/	最小	100 マイクロ秒	最小	—
E20(39(11)) E20(31AG) 【RX600, RX700 シリー ズ】	最大	約 72 時間	最大	CPU クロック周波数に依存 32 ビットカウンタ ×2 チャネル or 64 ビットカウンタ ×1 チャネル オーバフロー検出あり
E1(Serial)/E1(JTAG)/	最小	100 マイクロ秒	最小	-
E20(Serial)/E20(JTAG) 【RX200 シリーズ, RX140】	最大	約 72 時間	最大	CPU クロック周波数に依存 24 ビットカウンタ ×1 チャネル オーバフロー検出あり
シミュレータ	CPUクロック周波数に依存 64 ビットカウンタ オーバフロー検出あり		CPU グ 64 ビッ オーバ	フロック周波数に依存 ットカウンタ 、フロー検出あり

表 2.16 測定可能時間の範囲

2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】

この節では、カバレッジ機能を使用した、カバレッジ測定について説明します。

カバレッジ測定の方法にはいくつかの種類がありますが、CS+では次の領域を対象に、ソース行/関数に対する フェッチ系のコード・カバレッジ測定(COカバレッジ)、および変数に対するアクセス系のデータ・カバレッジ測定を 行います。

カバレッジ測定の対象となる領域は次のとおりです。

表 2.17 カバレッジ測定の対象領域

デバッグ・ツール	対象領域
シミュレータ	内蔵 ROM/RAM, エミュレーション ROM/RAM
E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】	任意の領域(4M バイト単位で最大 4 箇所)

- 備考 C0 カバレッジ:命令網羅率(ステートメント・カバレッジ) たとえば、コード内のすべての命令(ステートメント)を 少なくとも 1 回は実行した場合、C0 = 100 % となります。
- **注意 1.** E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】では、トレース機能やリアルタイム RAM モニタ機能と同時に使用することはできません。
- **注意 2.** E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】では、デー タ・カバレッジは測定できません。
- 注意 3. E20(JTAG)【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】では、ホットプラグイン接続でカバレッジ計測できません。 ホットプラグイン接続する場合は、プロパティ パネルの[デバッグ・ツール設定]タブ上の[カバレッジ]【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】 カテゴリ内の[コード・カバレッジ機能を使用する]プロパティを[いいえ]に設定してください。
- **注意 4.** E20(JTAG)【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】で は、プログラム実行中に「システムリセットを発行しました。」というエラー・メッセージが表示され た場合、プログラム実行開始からシステムリセット発生時点までの実行はカバレッジ測定されません。

2.15.1 カバレッジ測定の設定をする

カバレッジ機能を使用するためには、あらかじめカバレッジ測定に関する設定を行う必要があります。 設定は、プロパティパネルの[デバッグ・ツール設定] タブ上の [カバレッジ]【シミュレータ】、または [カバレッ ジ]【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】カテゴリ内で行いま す。

図 2.165 [カバレッジ] カテゴリ【シミュレータ】

▲ カバレッジ

カバレッジ機能を使用する	00
カバレッジ結果を再利用する	いいえ



図 2.166 [カバレッジ] カテゴリ【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】

4	カバレッジ	
	コード・カバレッジ機能を使用する	はい
	カバレッジ計測方法	CPU実行優先
a	カバレッジ計測範囲	[4]
	a [0]	FFC00000
	開始アドレス	HEX FFC00000
	カバレッジ計測領域サイズEMバイト] 4
	▶ [1]	
	▷ [2]	
	▶ [3]	
	カバレッジ結果を再利用する	いいえ

- (1) [カバレッジ機能を使用する]
 カバレッジ機能を使用するか否かをドロップダウン・リストにより指定します。
 カバレッジ機能を使用する場合は[はい]を選択してください(デフォルトでは[いいえ]が指定されます)。
- (2) [カバレッジ結果を再利用する] このプロパティは、[カバレッジ機能を使用する]プロパティにおいて[はい]を指定した場合のみ表示されます。
 デバッグ・ツールと切断時に、現在取得しているコード・カバレッジ測定結果を自動保存し、次回デバッグ・ ツールと接続した際に、保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。
 前回取得したコード・カバレッジ測定結果の内容を再現する場合は、[はい]を選択してください(デフォルトでは[いいえ]が指定されます)。
- (3) [コード・カバレッジ機能を使用する] コード・カバレッジ機能を使用するか否かをドロップダウン・リストにより指定します。 コード・カバレッジ機能を使用する場合は[はい]を選択してください(デフォルトでは[いいえ]が指定されます)。

デバッグ・ツールと切断中のみ指定することができます。

(4) [カバレッジ計測方法]
 このプロパティは、[コード・カバレッジ機能を使用する] プロパティにおいて [はい] を指定した場合のみ表示されます。
 カバレッジデータの計測方法をドロップダウン・リストにより指定します。
 ドロップダウン・リストには、次のカバレッジ計測方法が表示されます。

- CPU 実行優先, カバレッジ計測優先

[CPU 実行優先]を指定した場合、データ出力時のロストによりカバレッジデータが失われる可能性があります。

[カバレッジ計測優先]を指定した場合、カバレッジデータ出力のために CPU 実行が停止し、プログラム実行のリアルタイム性が損なわれる可能性があります。

(5) [カバレッジ計測範囲]

このプロパティは、[コード・カバレッジ機能を使用する]プロパティにおいて[はい]を指定した場合のみ表示されます。

カバレッジを計測する範囲を指定します。 カバレッジ計測範囲は 4M バイト単位で 4 つまで指定できます。 開始アドレスは、4M バイト単位の値を指定してください(入力値が 4M バイト単位の値でなかった場合は、自動的に補正されます)。

- (6) [カバレッジ結果を再利用する] このプロパティは、[コード・カバレッジ機能を使用する]プロパティにおいて[はい]を指定した場合のみ表示されます。 デバッグ・ツールと切断時に、現在取得しているコード・カバレッジ測定結果を自動保存し、次回デバッグ・ ツールと接続した際に、保存した測定結果の内容を再現するか否かをドロップダウン・リストにより指定します。
- 2.15.2 カバレッジ測定結果を表示する

プログラムの実行が開始すると自動的にカバレッジ測定が開始し、実行停止とともにカバレッジ測定も終了します。

(1) コード・カバレッジ率

(a) ソース行/逆アセンブル行に対するコード・カバレッジ率の表示 対象となるプログラムを表示しているエディタパネル/逆アセンブルパネルで行われます。 各パネルでは、表 2.18 に示す計算方法で算出されたコード・カバレッジ率を基に、対象ソース・テキスト行 /逆アセンブル結果行の背景色が表 2.19 のように色分け表示されます。 ただし、デバッグ・ツールと切断時、またはプログラム実行中は、結果の表示を行いません。 なお、取得したコード・カバレッジ測定結果は、エディタパネル/逆アセンブルパネル上のコンテキスト・ メニューの[カバレッジ情報のクリア]を選択することにより、すべてリセットすることができます(各パネ ル上の色分け表示もクリアされます)。

表 2.18 ソース行/逆アセンブル行に対するコード・カバレッジ率の計算方法

パネル	計算方法
エディタ パネル	"ソース行と対応するアドレス範囲内で実行されたバイト数 "÷"ソース行と対応 するアドレス範囲内の総バイト数 "
逆アセンブル パネル	" 逆アセンブル結果行と対応するアドレス範囲内で実行されたバイト数 " ÷" 逆ア センブル結果行と対応するアドレス範囲内の総バイト数 "

表 2.19 コード・カバレッジ測定結果の表示色(デフォルト)

コード・カバレッジ率	背景色
100 %	ソース・テキスト/逆アセンブル結果
1 ~ 99 %	ソース・テキスト/逆アセンブル結果
0%(未実行)	ソース・テキスト/逆アセンブル結果

- 備考 1. 各パネルにおけるコード・カバレッジ測定結果の表示更新は、プログラム停止ごとに自動的に 行われます。
- 備考 2. 上記の背景色は,オプション ダイアログにおける [全般 フォントと色] カテゴリの設定に依存します。
- 備考 3. 上記の背景表示は、対象領域外(「表 2.17 カバレッジ測定の対象領域」参照)の行に対しては 行われません。
- 備考 4. ダウンロードしているロード・モジュールの更新日時より,現在オープンしているソース・ ファイルの更新日時が新しい場合,エディタパネルではコード・カバレッジ測定結果の表示は 行われません。

図 2.167 コード・カバレッジ測定結果の表示例(エディタ パネル)



RENESAS



図 2.168 コード・カバレッジ測定結果の表示例(逆アセンブル パネル)

- (b) 各関数に対するコード・カバレッジ率の表示
 各関数に対するコード・カバレッジ率(関数の網羅率)は、解析ツールの関数一覧パネル内[コード・カバレッジ[%]]項目で確認することができます。
 " 関数のコード・カバレッジ率"についての詳細は、「CS+ 統合開発環境 ユーザーズマニュアル 解析ツール 編」を参照してください。
- (2) データ・カバレッジ率【シミュレータ】
 各変数に対するデータ・カバレッジ率は、解析ツールの変数一覧 パネル内 [データ・カバレッジ [%]] 項目で確認することができます。
 "変数のデータ・カバレッジ率"についての詳細は、「CS+ 統合開発環境 ユーザーズマニュアル 解析ツール編」
 を参照してください。



2.16 プログラム内へのアクションの設定

この節では、プログラム内に、指定したアクションを設定する操作方法について説明します。

2.16.1 printf を挿入する

アクション・イベントの1つである Printf イベントを設定することにより、プログラムの実行を任意の箇所で一瞬停 止させたのち、ソフトウエア処理により printf コマンドを実行させ、指定した変数式の値を出力 パネルに出力すること ができます。

この機能を使用するためには、次の手順で操作を行ってください。

2.16.1.1 Printf イベントを設定する
 2.16.1.2 プログラムを実行する
 2.16.1.3 出力結果を確認する
 2.16.1.4 Printf イベントを編集する

- **注意 1.** Printf イベントの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する留意事項」も参照してください。
- **注意 2.** ステップ実行中(📴 / 📴 / 🏝),またはブレーク関連のイベントを無視した実行中(应)に Printf イベントは発生しません。
- 2.16.1.1 Printf イベントを設定する

エディタ パネル/逆アセンブル パネル上で, printf コマンドを実行させたい箇所に Printf イベントを設定します。 エディタ パネル/逆アセンブル パネルにおいて, Printf イベントを設定したい行/アドレス^注にキャレットを移動し たのち, コンテキスト・メニューの [アクション・イベントの登録 ...]を選択すると, 次のアクション・イベント ダイ アログがオープンします。

このダイアログにおいて、次の操作を行ってください。

注 Printf イベントは、アドレス表示がない行に設定することはできません。

図 2.169 Printf イベントを設定する(アクション・イベント ダイアログ: [Printf イベント] タブ)

アクション・イベント	
Printf イベント 状態保存イベント 割り込みイベント設定	
出力文字列(Q): 入力例) サンブル:	
変数式(业): 入力例) aaa, bbb, ccc	
(ここに変数式を入力してください)	
アドレス(<u>A</u>):	
main 💌	
出力 パネル での書示例)	
727/k aaa = 10, DDD = 20 CCC = 30	
	-
OK THE AND AND]

(1) [出力文字列]の指定

出力パネルに出力する際に付与する文字列をキーボードより直接入力で指定します。 なお、出力する文字列は、1行分のみ入力可能です(空白可)。

RENESAS

- (2) [変数式]の指定
 Printf イベントの対象となる変数式を指定します。
 変数式は、テキスト・ボックスに直接入力で指定します(最大指定文字数:1024文字)。
 ","で区切ることにより、1つの Printf イベントとして 10 個までの変数式を指定することができます。
 エディタ パネル/逆アセンブルパネルにおいて、変数式を選択した状態でこのダイアログをオープンした場合
 では、選択している変数式がデフォルトで表示されます。
 なお、変数式として指定できる基本入力形式と、その際に Printf イベントとして出力される値についての詳細
 は、「表 A.15 変数式と出力される値の関係(Printf イベント)」を参照してください。
 - 備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- (3) [アドレス]の指定
 Printf イベントを指定するアドレスを指定します。
 デフォルトで,現在の指定位置のアドレスを表示します。
 編集する場合は、テキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字)、またはドロップダウン・リストにより入力履歴項目(最大履歴個数:10個)を選択します。
 - 備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- (4) [OK] ボタンのクリック
 ここで指定した Printf イベントをエディタ パネル/逆アセンブル パネル上のキャレット位置の行に設定します。
 Printf イベントが設定されると、エディタ パネル/逆アセンブル パネルのイベント・エリアに
 マークが表
 示され、イベント パネルで管理されます(「2.17 イベントの管理」参照)。

2.16.1.2 プログラムを実行する

プログラムを実行します(「2.9 プログラムの実行」参照)。 プログラムを実行することにより、Printf イベントを設定した箇所の命令実行直前でプログラムを一瞬停止し、指定 した変数式の値を出力パネルに出力します。

2.16.1.3 出力結果を確認する

出力 パネルの [デバッグ・ツール] タブでは,指定した変数式の値が次のように出力されます(図 A.28 Printf イベントの出力結果フォーマット」参照)。

図 2.170 Printf イベントの出力結果例

出力	区
サンブル、global_a=0(0x00000000)よ	[出力文字列]: サンプル
ハードウエア・ブレークにより停止しました。。	[変数式]: global_a
[EOF]	上記を指定した場合の出力結果
すべてのメッセージ (*デバッグ・ツール (*ラピ・	・ ッド・ビルド <u>/</u> "ビルド・ツール / ・

2.16.1.4 Printf イベントを編集する

一度設定した Printf イベントを編集することができます。

編集を行う場合は、イベント パネルにおいて、編集対象の Printf イベントを選択したのち、コンテキスト・メニューの [条件の編集 ...]を選択します。オープンするアクション・イベント ダイアログにおいて、編集が必要な項目を編集 したのち、[OK] ボタンをクリックします。

2.16.2 割り込みイベントを挿入する【シミュレータ】

アクション・イベントの1つである割り込みイベントを設定することにより、プログラムの実行中に任意の箇所で割 り込みを発生させることができます。 この機能を使用するためには、次の手順で操作を行ってください。

2.16.2.1 割り込みイベントを設定する 2.16.2.2 プログラムを実行する 2.16.2.3 割り込みイベントを編集する

注意割り込みイベントの設定に関しては(有効イベント数の制限など),「2.17.7 イベント設定に関する留意事項」も参照してください。

2.16.2.1 割り込みイベントを設定する

エディタ パネル/逆アセンブル パネル上で、割り込み要求を発生させたい箇所に割り込みイベントを設定します。 エディタ パネル/逆アセンブル パネルにおいて、割り込みイベントを設定したい行/アドレス^注にキャレットを移動 したのち、コンテキスト・メニューの [アクション・イベントの登録 ...]を選択すると表示されるアクション・イベン トダイアログで [割り込みイベント設定] タブ【シミュレータ】を選択します。

このダイアログにおいて,次の操作を行ってください。

注割り込みイベントは、アドレス表示がない行に設定することはできません。

図 2.171 割り込みイベントを設定する(アクション・イベント ダイアログ:[割り込みイベント設定]タブ)

ア	クション・イベント
F	Printf イベント 状態保存イベント 割り込みイベント設定
	割り込みべりを
	優先順位:
-	
	OK APPEN ANA
(1)	[割り込みベクタ]の指定
	割り込みベクタ番号を指定します。 0~ 255 の範囲で直接入力により指定してください。
(2)	[優先順位]の指定
	割り込み優先順位を指定します。
	-【RX610 グループ】 1~8の範囲で直接入力に上り指定してください
	8を指定した場合は、高速割り込みとして動作します。
	- 【RX610 グループ以外】
	1 ~ 16 の範囲で直接人力により指定してください。

RENESAS

16を指定した場合は、高速割り込みとして動作します。

 (3) [OK] ボタンのクリック ここで指定した割り込みイベントをエディタ パネル/逆アセンブル パネル上のキャレット位置の行に設定します。
 割り込みイベントが設定されると、エディタ パネル/逆アセンブル パネルのイベント・エリアに 2 マークが表示され、イベント パネルで管理されます(「2.17 イベントの管理」参照)。

2.16.2.2 プログラムを実行する

プログラムを実行します(「2.9 プログラムの実行」参照)。 プログラムを実行することにより、割り込みイベントを設定した箇所の命令実行直前で割り込み要求が発生し、CPU が割り込みを受け付けると割り込み例外処理を実行します。

- **注意 1.** 発生した割り込み要求が受け付けられ、割り込み例外処理が実行されると、割り込み要求はクリアされます。
- **注意 2.** 発生した割り込み要求が保留中に、別の割り込みイベントによる割り込み要求が発生すると、割り込み 優先順位が高い割り込みのみ有効となります。
- 2.16.2.3 割り込みイベントを編集する

一度設定した割り込みイベントを編集することができます。

編集を行う場合は、イベント パネルにおいて、編集対象の割り込みイベントを選択したのち、コンテキスト・メニューの [条件の編集 ...]を選択します。オープンする割り込みイベント詳細設定 ダイアログ【シミュレータ】において、編集が必要な項目を編集したのち、[OK] ボタンをクリックします。

- (1) [パス・カウント]の編集
 1~16383の範囲で直接入力により指定してください。
 入力したパス・カウント分のイベント条件を満たした時点で、イベント成立となります。
- (2) [割り込み]の編集 割り込みベクタと優先順位を編集することができます。編集方法はアクション・イベントダイアログの[割り込みイベント設定]タブ【シミュレータ】と同じです(「(1) [割り込みベクタ]の指定」、「(2) [優先順位]の 指定」参照)。



2.17 イベントの管理

イベントとは、"アドレス 0x1000 番地をフェッチした"、"アドレス 0x2000 番地にデータを書き込んだ" などのデ バッグにおけるマイクロコントローラの特定の状態を指します。

CS+では、このイベントを任意の箇所でのブレーク、トレース動作の開始/終了、タイマ計測の開始/終了などのデバッグ機能のアクション・トリガとして利用します。

この節では、これらのイベントを管理する方法について説明します。

イベントは、一括して次のイベントパネルで管理されます。

[表示] メニュー→ [イベント] を選択してください。

イベントパネルでは、現在設定されているイベントの詳細情報を一覧で確認することができ、各イベントの削除、設 定状態(有効/無効)の切り替え、詳細情報の表示、および設定の変更を行うことができます。

なお,各エリアの見方,および機能についての詳細は,イベントパネルの項を参照してください。

備考 複数のロード・モジュール・ファイルをダウンロードした場合, イベント パネルの詳細情報にはロー ド・モジュール名が表示されます。

図 2.172 設定したイベントの表示(イベント パネル)【E1】【E20】











2.17.1 設定状態(有効/無効)を変更する

対象となるイベント名のチェック・ボックスのチェックを変更することで、イベントの設定状態を変更することがで きます(イベントの設定状態を変更すると、対応してイベント・マークも変化します)。 イベントの設定状態には、次の種類があります。

図 2.175 イベント名のチェック・ボックス

~	\delta ハードウェア・ブレーク0001	
↑	ĺ ĺ ĺ	
チェック・ボックス	- イベント・マーク イベント名	Z

表 2.20 イベントの設定状態

>	有効状態	指定されている条件の成立で、対象となるイベントが発生します。 チェックを外すことにより、イベントを無効状態にすることができます。
	無効状態	指定されている条件が成立しても、対象となるイベントは発生しません。 チェックすることにより、イベントを有効状態にすることができます。
	保留状態	指定されている条件が, デバッグ対象のプログラムでは設定することができません。チェッ ク・ボックスを操作することはできません。

- 備考 1. Run-Break タイマ・イベント(EZ Emulator では未サポート)を無効状態/保留状態にすることはできません。
- 備考 2. イベントの状態は、エディタ パネル/逆アセンブル パネル上のイベント・マークを右クリックすることで表示される、メニューからの選択でも変更することができます。
- 備考 3. 無条件トレース・イベントとトレース・イベントにおける有効/無効状態の設定は、排他制御となります。このため、ビルトイン・イベントである無条件トレース・イベントは、デフォルトで有効状態で設定されていますが、トレース開始イベント/トレース終了イベントのいずれかが設定されると同時に自動的に無効状態に変更され、トレース開始イベント/トレース終了イベントを1つにまとめたトレース・イベントが有効状態になります。

また逆に、設定されているトレース・イベントを無効状態にすると、自動的に無条件トレース・イベントが有効状態となります。

2.17.2 特定のイベント種別のみ表示する

ツールバーの次のボタンをクリックすることで、特定のイベント種別のみを表示することができます。

	ハードウエア・ブレーク関連のイベントを表示します。
III [E20] [EZ Emulator]	ソフトウエア・ブレーク関連のイベントを表示します。
\$	トレース関連のイベントを表示します。
3	タイマ関連のイベントを表示します。
	アクション・イベント(Printf イベント,割り込みイベント【シミュレータ】) を表示します。
	ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・イベ ント)を表示します。 ^注

注

EZ Emulator の場合, Run-Break タイマ・イベントは未サポートとなります。

2.17.3 イベントのアドレスにジャンプする

次のボタンをクリックすることにより、現在選択しているイベントのアドレスに対応して、各パネルにジャンプします。

ただし、トレース・イベント/タイマ計測イベント/ビルトイン・イベント(無条件トレース・イベント/Run-Break タイマ・イベント)を選択している場合は、このボタンは無効となります。

	選択しているイベントが設定されているアドレスに対応するソース行にキャレットを移動した 状態で,エディタ パネルがオープンします。
*	選択しているイベントが設定されているアドレスに対応する逆アセンブル結果にキャレットを 移動した状態で, 逆アセンブル パネルがオープンします。
	選択しているイベントが設定されているアドレスに対応するメモリ値にキャレットを移動した 状態で,メモリパネルがオープンします。

2.17.4 イベントの詳細設定を編集する

ここでは、各種イベント詳細設定の編集方法を説明します。なお、タイマ計測イベントの編集に関しては、「2.14.3.3 タイマ計測イベントを編集する【E1】【E20】【EZ Emulator】」を、アクション・イベント(Printf イベント、割り込み イベント)の編集に関しては、「2.16 プログラム内へのアクションの設定」を参照してください。

2.17.4.1 実行系イベントを編集する

2.17.4.2 アクセス系イベントを編集する

2.17.4.3 イベントの組み合わせ条件を編集する【E1】【E20】【EZ Emulator】

2.17.4.1 実行系イベントを編集する

実行系イベントのアドレス条件およびパス・カウント条件を編集することができます。

実行系イベントは、イベントパネルにおいて次の操作により表示される実行イベント詳細設定 ダイアログにて編集します。

ハードウエア・ブレーク	編集したいハードウエア・ブレークにキャレットを移動したのち、コンテ キスト・メニューの「条件の編集 …」を選択
トレース	編集したいトレース内の実行系イベントにキャレットを移動したのち、コ ンテキスト・メニューの「条件の編集 …」を選択



組み合わせブレーク	編集したい組み合わせブレーク内の実行系イベントにキャレットを移動し
【E1】【E20】【EZ Emulator】	たのち,コンテキスト・メニューの「条件の編集 …」を選択
タイマ計測	編集したいタイマ計測内の実行系イベントにキャレットを移動したのち, コンテキスト・メニューの「条件の編集」を選択

(1) アドレスの編集

[アドレス]カテゴリ内の[アドレス]プロパティに 16 進数の数値を入力して OK ボタンをクリックしてください。

(2) パス・カウントの編集

[パス・カウント]カテゴリ内の[パス・カウント]プロパティに 10 進数の数値を入力して OK ボタンをクリッ クしてください。入力したパス・カウント分のイベント条件を満たした時点で、イベント成立となります。 パス・カウントの指定可能値は、デバッグ・ツールにより次のように異なります。

- [E1] [E20] [EZ Emulator]
- 1 ~ 256

-【シミュレータ】

1 ~ 16383

注意 1. 【E1】【E20】【EZ Emulator】

パス・カウントは、全イベント(アクセス系イベント含む)中で1つのイベントに対してのみ1 以外の値を指定可能です。 2つのイベントに1以外のパス・カウントを設定した場合、エラーとなります。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】 パス・カウントは、1 以外の値を指定できません。

2.17.4.2 アクセス系イベントを編集する

アクセス系イベントのアドレス条件およびデータ条件を編集することができます。

なお,アドレスおよびパス・カウントの編集方法については,「(1)アドレスの編集および(2)パス・カウントの編 集」を参照してください。

アクセス系イベントは、イベント パネル上で次の操作により表示されるアクセスイベント詳細設定 ダイアログにて編 集します。

トレース	編集したいトレース内のアクセス系イベントにキャレットを移動したのち, コンテキスト・メニューの「条件の編集 …」を選択
ポイント・トレース	編集したいポイント・トレース内のアクセス系イベントにキャレットを移 動したのち,コンテキスト・メニューの「条件の編集 …」を選択
組み合わせブレーク 【E1】【E20】【EZ Emulator】	編集したい組み合わせブレーク内のアクセス系イベントにキャレットを移 動したのち,コンテキスト・メニューの「条件の編集 …」を選択
タイマ計測	編集したいタイマ計測内のアクセス系イベントにキャレットを移動したの ち,コンテキスト・メニューの「条件の編集 …」を選択

(1) アドレス条件の編集

[アドレス条件]カテゴリの[比較条件]プロパティにおいて比較条件をドロップダウン・リストより指定してください。

指定した比較条件により、[アドレス]カテゴリ内の[アドレス]プロパティに表示されたアドレス値と組み合わ せて次のようにアドレス条件が設定できます。

- [指定無し]を指定した場合 アドレス値を条件とします。アドレス値に対するアクセスが発生した場合に、条件成立となります。

- [アドレス範囲]を指定した場合【E1】【E20】【EZ Emulator】 アドレス値を開始アドレスとします。[アドレス条件]カテゴリ内に追加される[終了アドレス]プロパティに おいて,終了アドレス値を入力してください。設定した開始/終了アドレスの範囲内に対してアクセスが発生

した場合に,条件成立となります。 また,同じく追加される[範囲条件]プロパティにおいて,アドレス範囲外にも設定できます。

注意 1. 【RX100, RX200 シリーズ】 アドレス範囲による比較条件は未サポートです。

- **注意 2.** 【RX600, RX700 シリーズ】 設定可能なアドレス範囲による比較条件は、1 点のみです。
- **注意 3.** 【シミュレータ】 アドレス条件は指定できません。
- [マスク付き比較]を指定した場合【E1】【E20】【EZ Emulator】 アドレス値に対するマスク値を設定できます。[アドレス条件]カテゴリ内に追加される[マスク値]プロパ ティにおいて、16進数のアドレスマスク値を入力してください。マスク付きアドレスと一致したアドレスに 対してアクセスが発生した場合、条件成立となります。 また、同じく追加される[比較]プロパティにおいて、アドレス比較条件を不一致にも設定できます。
- **注意** マスクは,条件となるアドレス値に対してビット単位でマスク値 "0"を Don't Care とします。 例) 0x1000 ~ 0x1FFF のアドレス条件を設定する場合 アドレス値: 0x1000
 - マスク値:0xF000

(2) データ条件の編集

- [アクセス種別]

アクセス種別として次の項目から指定します。

リード	リードアクセスが発生した場合に、条件成立となります。
ライト	ライトアクセスが発生した場合に、条件成立となります。
リード/ライト	リードまたはライトアクセスが発生した場合に、条件成立となります。

- [アクセスサイズ]

アクセスサイズとして次の項目から指定します。

指定無し	全てのアクセスサイズに対して アクセスが発生した場合に,条件成立となります。
バイト	バイトサイズのアクセスが発生した場合に、条件成立となります。
ワード	ワードサイズのアクセスが発生した場合に,条件成立となります。
ロングワード	ロングワードサイズのアクセスが発生した場合に、条件成立となります。

- [比較条件]

データ比較条件を指定します。

デバッグ・ツールにより、比較条件は次のように異なります。

【E1】【E20】【EZ Emulator】	指定値一致 (==)	データ比較条件が一致した場合に,条件成立となります。
	指定値不一致 (!=)	データ比較条件が不一致した場合に,条件成立となります。



【シミュレータ】	一致 (==)	データが指定値と一致で条件成立となります。
	不一致 (!=)	データが指定値と不一致で条件成立となります。
	符号反転 ^注	前回アクセス時のデータと今回アクセス時のデータで符号が 反転した場合に条件成立となります。
	差分 ^注	前回アクセス時のデータと今回アクセス時のデータの差分が 指定値を超えた場合に条件成立となります。
	指定値より大きい (>)	データが指定値より大で条件成立となります。
	指定値より小さい (<)	データが指定値より小で条件成立となります。
	指定値以上 (>=)	データが指定値以上で条件成立となります。
	指定值以下 (<=)	データが指定値以下で条件成立となります。
	範囲内 (<=Value<=)	データが指定値の範囲内で条件成立となります。 ([比較データ 1] <= データ <= [比較データ 2])
	範囲外 !(<=Value<=)	データが指定値の範囲外で条件成立となります。 (データ < [比較データ 1] [比較データ 2] < データ)
	指定値無し	比較データを指定しません。

- 注 [符号反転],および[差分]は前回データを比較するため、リセット後,および条件成立1回 目の判定は常に不成立となります。
- 備考 【シミュレータ】 このプロパティは, [アクセスサイズ]プロパティにおいて, [指定無し]を指定した場合は表 示されません。
- [比較データ]【E1】【E20】【EZ Emulator】 データ値を 16 進数で指定します。
- [比較データ 1]【シミュレータ】 データ値を 16 進数で指定します。 このプロパティは、[比較条件]プロパティにおいて、次の項目を指定した場合に表示されます。 [一致 (==)] / [不一致 (!=)] / [指定値より大きい (>)] / [指定値より小さい (<)] / [指定値以上 (>=)] / [指定値以下 (<=)] / [範囲内 (<=Value<=)] / [範囲外 !(<=Value<=)]
- [比較データ 2]【シミュレータ】 データ値を 16 進数で指定します。 このプロパティは, [比較条件]プロパティにおいて, [範囲内 (<=Value<=)] / [範囲外 !(<=Value<=)]を指定 した場合のみ表示されます。
- [データマスクを指定する] 比較データ値を入力した場合にプロパティが表示され、データ値に対するマスク値を指定するか否かを指定で きます。[はい]を指定した場合、データマスクを指定できます。
- 備考 【シミュレータ】 このプロパティは、[比較条件]プロパティにおいて、次の項目を指定した場合に表示されま す。 [一致 (==)] / [不一致 (!=)] / [指定値より大きい (>)] / [指定値より小さい (<)] / [指定 値以上 (>=)] / [指定値以下 (<=)] / [範囲内 (<=Value<=)] / [範囲外 !(<=Value<=)]

- [マスク値] [データマスクを指定する]プロパティにおいて、[はい]を指定した場合にプロパティが表示されます。16進数のデータマスク値を入力して、マスク付きデータと比較条件が一致したデータアクセスが発生した場合、条件成立となります。

- [差分]【シミュレータ】 比較データを 16 進数で指定します。 このプロパティは, [比較条件]プロパティにおいて, [差分]を指定した場合のみ表示されます。
- [符号]【シミュレータ】 比較するデータに符号があるか否かを指定します。 このプロパティは, [比較条件]プロパティにおいて, 次の項目を指定した場合に表示されます。

[差分]/[指定値より大きい (>)]/[指定値より小さい (<)]/[指定値以上 (>=)]/[指定値以下 (<=)] /[範囲内 (<=Value<=)]/[範囲外 !(<=Value<=)]

- (3) バス・マスタの編集【E1/E20【 RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グ ループ】】
 - [バス・マスタ]

バス・マスタとして次の項目から指定します。

CPU	指定したデータアクセスが、CPU からのアクセスにより発生した場合に、条件 成立となります。
DMAC/DTC	指定したデータアクセスが,DMAC/DTC からのアクセスにより発生した場合 に,条件成立となります。

2.17.4.3 イベントの組み合わせ条件を編集する【E1】 【E20】 【EZ Emulator】

複数のイベントを設定した組み合わせブレークおよびトレースに対する組み合わせ条件を編集します。トレースの組 み合わせ条件は、タブを切り替えることで開始と終了のそれぞれの条件を設定することができます。

イベントの組み合わせ条件は、イベントパネル上で次の操作により表示される組み合わせ条件設定 ダイアログ【E1】 【E20】【EZ Emulator】にて編集します。

端子リセットを入れた場合, EZ Emulator からの再実行時に組み合わせイベント情報はクリアされます。

組み合わせブレーク	組み合わせブレークにキャレットを移動したのち,コンテキスト・メ ニューの「条件の編集 …」を選択
トレース	トレースにキャレットを移動したのち,コンテキスト・メニューの「条件 の編集 …」を選択

(1) 組み合わせ条件の編集

組み合わせ条件をプルダウンメニューにより, [OR], [AND] および [シーケンシャル] から選択します。組み合わせ条件ごとの動作説明については、「22.10.5 複数のブレーク・イベントを組み合わせる(組み合わせブレーク)【E1】【E20】【EZ Emulator】」および「2.13.3.2 複数のイベントを組み合わせる【E1】【E20】【EZ Emulator】」を参照してください。

図 2.176 [組み合わせ]項目

組み合わせ:	OR	*
	OR AND シーケンシャル	

- 注意 1. トレース・イベントにおいてトレース終了イベントの組み合わせ条件は, [OR] のみです。
- **注意 2.** 組み合わせ条件における [AND] および [シーケンシャル] は、組み合わせブレークとトレース開始イベントとで兼用です。そのため、どちらか一方で [AND] および [シーケンシャル] を組み合わせ条件に設定した場合、もう片方は、[OR] でのみ設定できます。
- **注意 3.** 【RX600, RX700 シリーズ】 組み合わせ条件がシーケンシャルの場合,アクセス系イベントは,リセットイベント,および, 3 番目までのイベントに指定できます。
- **注意 4.** 【RX600, RX700 シリーズ】 組み合わせ条件がシーケンシャルの場合,アドレス範囲条件を設定したアクセス系イベントは1 番目のイベントにのみ指定できます。
- **注意 5.** 【RX100, RX200 シリーズ】 組み合わせ条件がシーケンシャルの場合,アクセス系イベントは2番目までのイベントに指定で きます。

 (2) 対象イベントの編集
 [↑上]および[↓下]ボタンで対象イベントリスト内のイベントの順序を変更します。このボタンは、組み合わ せ条件をシーケンシャルに選択し、対象イベントリスト内のイベントにキャレットを移動した場合のみ有効とな ります。

注意 【EZ Emulator】

図 2.177	対象イベントリス	ト内のイベント順序変更	(シーケンシャル)
---------	----------	-------------	-----------

組み合わ)せ条件 💽 💽
ブレーク 組み合	わせ: シーケンシャル 🔹
対象イ	
No.	1/0/h
1	実行後 sort.c#50 0xffff8670 下
2	実行後 sortc#35 0xffff85de
3	リード e_CharBuf 0x1000 - 0x1000 == 0xb Rイベント
4	リード/ライト e_IntBuf 0x1404 - 0x1407 =
リセット	イベンド
110	ト 単語除
	OK キャンセル ヘルプ(H)

(3) リセットイベントの編集

リセットイベントを登録する場合、[Rイベント]ボタンをクリックします。このボタンは、組み合わせ条件を シーケンシャルに選択して対象イベントリスト内のイベントにキャレットを移動した場合のみ有効となります。 また、リセットイベントの登録を解除する場合、[解除]ボタンをクリックします。このボタンは、リセットイ ベントリスト内のイベントにキャレットを移動して有効となります。

- 注意 1. リセットイベントは1点のみ指定可能です。 リセットイベントが登録された状態でRイベントボタンを押した場合、対象イベントリスト内で 選択したイベントと登録済みのリセットイベントが置き換わります。
- 注意 2. リセットイベントが登録された状態で組み合わせ条件を [シーケンシャル]から [OR] または [AND] に変更した場合、リセットイベントリスト内のイベントは対象イベントに含まれません。 対象イベントとする場合、リセットイベントを登録解除してから組み合わせ条件を変更してくだ さい。
- **注意 3.** 組み合わせ条件がシーケンシャルの場合、リセットイベントに登録したイベントは、パス・カウント条件を指定できません。
- **注意 4.** 【RX600, RX700 シリーズ】 アドレス範囲条件を指定したイベントは、リセットイベントに登録できません。また、リセット イベントに登録したイベントは、アドレス範囲条件を指定できません。


組み合わせ条件
ブレーク
組み合わせ: シーケンシャル 🗸
対象イベント
No. 1701
1 実行後 sort.c#50 0xffff8670 下
2 実行後 sort.c#35 0xffff85de
3 リード e_CharBuf 0x1000 - 0x1000 == 0xb Rイベント
4 リード/ライト e_IntBuf 0x1404 - 0x1407 =
リセットイベント
イベント 解除
OK キャンセル ヘルプ(日)

図 2.178 リセットイベント登録 (シーケンシャル)

図 2.179 リセットイベント登録解除(シーケンシャル)

組み合わ)せ条件 💽 💽
ブレーク	
組み合	わせ: シーケンシャル 🔹
対象在	
No.	1/心ト 上
1	実行後 sort.c#50 0xffff8670 下
2	リード c_CharBuf 0x1000 - 0x1000 == 0xb
3	リード/ライト e_IntBuf 0x1404 - 0x1407 = Rイベント
リセット イベン 実行行	イベント: ト 資 sort.c#35 0xffff85de
	OK キャンセル ヘルプ(H)

2.17.5 イベントを削除する

設定したイベントを削除するには、対象イベントを選択したのち、ツールバーの X ボタンをクリックします。 なお、複数のイベントで設定されたイベントの場合、詳細情報内のイベントを個別に削除することもできます。 ただし、ビルトイン・イベントである無条件トレース・イベント /Run-Break タイマ・イベントを削除することはで きません。

- 備考 1. タイマ開始/終了イベント、トレース開始/終了イベントおよびアクション・イベントについては、エ ディタ パネル/逆アセンブル パネル上で表示されているイベント・マークからコンテキスト・メ ニューの [イベント削除]を選択することで、イベントを削除することができます。
- 備考 2. ソフトウエア・ブレーク/ハードウエア・ブレークについては、エディタ パネル/逆アセンブル パネル上で表示されているイベント・マークをクリックすることで、イベントを削除することができます。
- 備考 3. 設定したイベントを一度にすべて削除する場合は、コンテキスト・メニューの[すべて選択]を選択したのち, X ボタンをクリックします(ビルトイン・イベントを除く)。

2.17.6 イベントにコメントを入力する

設定した各イベントに対して、ユーザが自由にコメントを入力することができます。

コメントの入力は、コメントを入力したいイベントを選択したのち、[コメント] エリアをクリックし、任意のテキストをキーボードから直接入力します([Esc] キーの押下で編集モードをキャンセルします)。

コメントを編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、編集を完了します。

なお、コメントは最大256文字まで入力することができ、使用中のユーザの設定として保存されます。

2.17.7 イベント設定に関する留意事項

ここでは、各種イベントの設定を行う際の留意事項を示します。

2.17.7.1 有効イベント数の制限 2.17.7.2 実行中に設定/削除可能なイベント種別 2.17.7.3 その他の注意事項

2.17.7.1 有効イベント数の制限

有効状態で同時に設定可能なイベントの個数には、次の制限があります。 したがって、新たに有効状態のイベントを設定する際にこの制限数を越えてしまう場合は、いったん設定しているイ ベントのいずれかを無効状態にする必要があります。

表 2.21 有	肓効イベント	数の制限	[RX600.	RX700	シリーズ】
----------	--------	------	---------	-------	-------

イベント種別	デバッグ・ツール	
	E1(Serial) / E1(JTAG) / E20(Serial) / E20(JTAG) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 ^{注 9}	0
ハードウエア・ブレーク (実行系:実行前)	8 ^{注1}	1024 ^{注 8}
ブレーク・イベント(実行系:実行後)	8 ^{注1}	0
ブレーク・イベント (アクセス系:リード , ライト , リード / ライト)	4注1注3	1024
組み合わせブレーク	8+4 ^{注1注2注5}	0
トレース(トレース開始/トレース終了)	8+4 ^{注1注2注6}	256
ポイント・トレース	4 ^{注1注3}	256
タイマ計測(タイマ開始/タイマ終了)	8+4 ^{注1注2注6}	1 ^{注7}
アクション(Printf イベント)	256 ^{注 9}	1024 ^{注 8}
アクション(割り込みイベント)	0	256



イベント種別	デバッグ・ツール	
	E1(Serial) / E20(Serial) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 ^{注 9}	0
ハードウエア・ブレーク(実行系:実行前)	4 ^{注4}	1024 ^{注 8}
ブレーク・イベント(実行系:実行後)	4 ^{注4}	0
ブレーク・イベント (アクセス系:リード , ライト , リード / ライト)	2 ^{注4}	1024
組み合わせブレーク	4+2 ^{注4}	0
トレース(トレース開始/トレース終了)	4+2 ^{注4注5注6}	256
ポイント・トレース	4 ^{注4}	256
タイマ計測(タイマ開始/タイマ終了)	4+2 ^{注4注6}	1 ^{注7}
アクション(Printf イベント)	256 ^{注 9}	1024 ^{注 8}
アクション(割り込みイベント)	0	256

表 2.22 有効イベント数の制限【RX200 シリーズ, RX140 グループ】

表 2.23 有効イベント数の制限【RX140 グループ以外の RX100 シリーズ】

イベント種別	デバッグ・ツール	
	E1(Serial) / E20(Serial) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	256 ^{注 9}	0
ハードウエア・ブレーク(実行系:実行前)	4 ^{注 4}	1024 ^{注 8}
ブレーク・イベント(実行系:実行後)	4 ^{注4}	0
ブレーク・イベント (アクセス系:リード , ライト , リード / ライト)	2 ^{注4}	1024
組み合わせブレーク	4+2 ^{注4}	0
トレース(トレース開始/トレース終了)	4+2 ^{注4注5注6}	256
ポイント・トレース	4 ^{注4}	256
タイマ計測(タイマ開始/タイマ終了)	0	1 ^{注7}
アクション(Printf イベント)	256 ^{注 9}	1024 ^{注 8}
アクション(割り込みイベント)	0	256

- 注 1. 実行アドレスに対するイベント 8 点とデータアクセスに対するイベント 4 点があり,各機能で兼用します。
- 注 2. イベントの組み合わせは最大 8 点まで設定可能です。

注 3. 【E20(JTAG】 トレース機能/リアルタイム RAM モニタ (RRM) 機能は一部排他使用の機能です。そのため, RRM 機 能を使用時はポイント・トレースを使用することができません。また, アクセス・イベントは RRM ブ ロックに空きがある場合のみ使用可能です。

- 注 4. 実行アドレスに対するイベント 4 点とデータアクセスに対するイベント 2 点があり,各機能で兼用します。
- 注 5. 組み合わせブレークの組み合わせ条件がシーケンシャルの場合、マイクロコントローラにより次のよう に最大イベント設定数が異なります。

- -【RX600, RX700 シリーズ】 7 点 + リセットイベント (R イベント) -【RX100, RX200 シリーズ】 3 点 + リセットイベント (R イベント)
- 注 6. トレース開始/終了イベント、タイマ計測開始/終了イベントにおいて実行系イベントの最大数を設定 した場合,エディタパネル/逆アセンブルパネルにおいてコンテキストメニューの[ここまで実行] は使用できません。また, CPU リセット後に指定シンボル位置まで実行する機能も使用できません。
- 注7. タイマ開始/終了イベントは、実行アドレスまたはデータアクセスに対するイベントをそれぞれ1点のみ設定可能です。
- 注 8. ハードウエア・ブレーク (実行系 : 実行前) とアクション (Printf イベント) は,各機能で兼用します。
- 注 9. ソフトウエア・ブレークとアクション (Printf イベント) は, 各機能で兼用します。

2.17.7.2 実行中に設定/削除可能なイベント種別

プログラム実行中、またはトレーサ/タイマ実行中に設定/削除可能なイベント種別は次のとおりです。

表 2.24 実行中に設定/削除可能なイベント種別

イベント種別	デバッグ・ツール	
	E1(Serial) / E1(JTAG) / E20(Serial) / E20(JTAG) EZ Emulator	シミュレータ
ソフトウエア・ブレーク	△ ^{注1注2}	-
ハードウエア・ブレーク(実行系:実行前)	△ ^{注2}	0
ブレーク・イベント(実行系:実行後)	△ ^{注2}	-
ブレーク・イベント (アクセス系:リード , ライト , リード / ライト)	∆ ^{注2}	0
トレース(トレース開始/トレース終了)	×	
ポイント・トレース	×	0
タイマ計測(タイマ開始/タイマ終了)	×	×
アクション(Printf イベント)	△ ^{注1注2}	0
アクション(割り込みイベント)	-	0

〇:可能

- △ : プログラムの実行を一瞬停止することで可能
- □ : トレーサ動作中は不可
- :非サポート
- ×:不可
- 注 1. 内蔵 RAM のみ設定可能
- 注 2. 低消費電力状態中は設定不可

2.17.7.3 その他の注意事項

- ローカル変数にイベントを設定することはできません。
- ステップ実行中(リターン・アウト実行を含む)、およびコンテキスト・メニューの[ここまで実行]によるプログラム実行中、イベントは発生しません。
- デバッグ対象のプログラムを再ダウンロードすることにより、既存のイベント設定位置が命令の途中になる場合に おける該当イベントの再設定方法は次のとおりです。

- デバッグ情報がある場合 イベント設定位置は常にソース・テキスト行の先頭に移動します。
- デバッグ情報がない場合 プロパティ パネルの [ダウンロード・ファイル設定] タブ上の [ダウンロード] カテゴリ内 [イベント設定位 置の自動変更方法] プロパティの設定に依存します。



2.18 フック処理を設定する

この節では、フック処理機能を使用し、デバッグ・ツールにフックを設定するための操作方法について説明します。 フック処理を設定することで、ロード・モジュールのダウンロード前後や CPU リセット後に、I/O レジスタ /CPU レ ジスタの値を自動的に変更することができます。

- フック処理の設定は,プロパティ パネルの[フック処理設定]タブ上の[フック処理設定]カテゴリ内で行います。

備考 たとえば、[ダウンロード前] プロパティで I/O レジスタを設定することにより、ダウンロードを高速 に行うことができます。

また、外部 RAM へのダウンロードも、同様の設定で容易に行うことができます。

図 2.180 [フック処理] カテゴリ

▲ フック処理設定

	Construction of the second sec	
Þ	ダウンロード前	ダウンロード前[0]
\triangleright	ダウンロード後	ダウンロード後[0]
Þ	ブレーク中のCPUリセット後	ブレーク中のCPUリセット後[0]
Þ	実行開始前	実行開始前[0]
⊳	ブレーク後	ブレーク後[0]

表 2.25 [フック処理] カテゴリのプロパティ

プロパティ	タイミング
ダウンロード前	ロード・モジュール・ファイルをダウンロードする直前に,指定した処理を行い ます。
ダウンロード後	ロード・モジュール・ファイルをダウンロードした直後に,指定した処理を行い ます。
ブレーク中の CPU リセット後	CPU リセット直後に,指定した処理を行います。
実行開始前	プログラムの実行開始直前に,指定した処理を行います。
ブレーク後	プログラムの実行がブレークした直後に,指定した処理を行います。

[フック処理設定]カテゴリ内の各プロパティは、フック処理を行うタイミングを示し、プロパティ値の"[]"内は、現 在指定されている処理の数を示します(デフォルトで設定されているフック処理はありません)。

フック処理を行いたいプロパティに,目的の処理を次の手順で指定します。

処理の指定は,該当するプロパティを選択すると欄内右端に表示される[…]ボタンをクリックすることでオープン する,次のテキスト編集 ダイアログ上で行います。

図 2.181 テキスト編集 ダイアログのオープン

	ファク処理設定		
Þ	ダウンロード前	ダウンロード前[0]	
\triangleright	ダウンロード後	ダウンロード後[0]	



テキスト編集	
テキスト(ゴ):	
	-
4	P
	OK キャンセル ヘルプ(H)

図 2.182 フック処理を設定する(テキスト編集 ダイアログ)

このダイアログにおいて、目的の処理を直接入力により指定します。 処理の指定形式は次のいずれかです。

【指定形式:1】

I/O レジスタの内容を、数値に自動的に書き換えます。

I/O レジスタ名 数値

【指定形式:2】

CPU レジスタの内容を, 数値に自動的に書き換えます。

CPU レジスタ名 数値

【指定形式:3】

Python スクリプト・パス(絶対パス/プロジェクト・フォルダを基点とした相対パス)で指定したスクリプト・ファイルを実行します。

Source Python スクリプト・パス

備考 1. 処理の指定の際, 行頭に "#" を付与することにより, その行はコメント扱いとなります。

備考 2. 半角スペースは、タブ文字でも代用可能です。

注意 デバッガのフック処理から Python スクリプトを実行する場合,以下のコマンドが記載可能です。

debugger.Register.GetValue debugger.Register.SetValue debugger.Memory.GetValue debugger.Memory.SetValue

それ以外の Python コマンドを使用したい場合, Python コンソールの Hook コマンドを使用してください。

1 処理につき 64 文字まで入力可能で、各プロパティごとに 128 個までの処理を指定することができます(テキスト 編集 ダイアログ上の [テキスト] エリア内の 1 行が 1 処理に相当)。

処理の指定が完了したのち、[OK] ボタンをクリックすると、指定した処理がプロパティ パネル上に反映されます。

図 2.183 フック処理設定の例

テキスト編集	X
テキスト(<u>1</u>):	
CRCCRCDOR 0×00	*
4	
OK キャンセル	くそう



2.19 消費電流測定【E2】

消費電流測定は、デバッグ対象のシステム全体の消費電流値を測定する機能です。

注意 消費電流測定は RX100, RX200 シリーズでのみ対応しています。

[使い方]

E2 エミュレータの消費電流測定は、ターゲット・システム全体の電流値を測定します。 E2 エミュレータのターゲットへの電源供給機能と E2 拡張インタフェースを利用して実現しています。 本機能を使用する場合は、必ずデバッグ・ツール接続前に、プロパティ パネルの[接続用設定]タブ上の[エミュ レータからの電源供給をする(最大 200mA)]プロパティを[はい]に、[E2 拡張インタフェースを使用する]プ ロパティを[エミュレータからの電源供給で使用する]に設定してください。

(1) 消費電流測定 パネルを開く

デバッグ・ツールに E2 エミュレータを選択し、ソリューション一覧 パネルを開き、消費電流測定の [GO] ボ タンをクリックすると、消費電流測定 パネル [E2] が開きます。



図 2.184 消費電流測定 パネル【E2】

(2) 条件(測定条件,トリガ条件,通過ポイント)を設定する 消費電流測定では、事前に条件を設定することで、検出したい電流値の遷移のパターンをとらえることが可能となります。事前に設定可能な条件は、大きく分けて測定条件、トリガ条件、通過ポイントです。

備考 条件はデフォルトで以下のように設定されています。 記録メモリを使い切った後の動作:記録メモリを上書きして実行し続ける サンプリング時間:10us 取得の条件:すべて トリガ機能を使用しない 通過ポイントの設定なし

試しに測定のみを行いたい場合は、「(3)測定を開始する」から行ってください。

(a) 測定条件を設定する
 電流測定値の記録に関する条件を設定します。
 設定は消費電流測定パネル【E2】のツールバーの
 をクリックして表示される消費電流測定条件設定 ダイアログ【E2】で行います。

図 2.185	消費電流測定条件設定 ダイ	(アログ
2.100	们只电加加之不自以足了。	

測定条件設定		×	
記録メモリを使い切った後の動作(M):	プログラムを停止する	~	
サンプリング時間(S):	10us	~	
取得条件			1
条件(<u>C</u>):		すべて ~	
チャネル(<u>A</u>):		ch0 🗸	
外部トリガ入力の種類(工):		High 🗸	
通過ポイント範囲(<u>R</u>):	\sim	~	
	OK	キャンセル ヘルプ(日)]

- <1> 記録メモリを使い切った後の動作を選択する 電流値を記録する E2 エミュレータ内のメモリを使い切った後の動作を以下から選択します。
 - 記録メモリを上書きして実行し続ける
 - 記録を停止する(プログラムの実行は継続)
 - プログラムを停止する
- <2> サンプリング時間を選択する 以下の中から測定のサンプリング時間を選択します。 1us, 2us, 5us, 10us, 20us, 50us, 100us, 200us, 500us, 1ms
- <3> 取得条件を選択する 特定の条件を満たしたときにのみ、電流値を E2 エミュレータ内のメモリに記録するように設定することが できます。

条件は以下の中から選択可能です。

- すべて

特定の条件を指定せず、測定開始後にサンプリングした電流値をすべて記録します。

- 外部トリガ入力検出中

E2 拡張インタフェースの外部トリガ入力機能を用いて,指定した信号を検出している間のみ記録します。 本項目を指定した場合,検出対象の外部トリガ入力のチャネルと検出する信号のレベルを指定する必要が あります。

また、デバッグ・ツールの接続前に外部トリガ入力の設定を正しく行っている必要があります。

- 備考 E2 拡張インタフェースの外部トリガ入力機能の設定についての詳細は、「消費電流チュー ニングソリューション(E2 エミュレータ、CS+ 編) アプリケーションノート」を参照して ください。
- 通過ポイント間
 [通過ポイント範囲]の左側の入力欄に指定した通過ポイントを開始条件、右側の入力欄に指定した通過 ポイントを終了条件とします。
 開始条件に指定した通過ポイントが検出されてから、終了条件に指定した通過ポイントが検出されるまでの間の電流値を記録します。
 通過ポイントについての詳細は、「(c)通過ポイントを設定する」を参照してください。

 (b) トリガ条件を設定する 測定中に電流値が特定の条件を満たしたことをトリガとして、プログラム停止や E2 エミュレータによる外部 トリガ出力が可能です。 設定は消費電流測定パネル【E2】のツールバーの ダイアログ【E2】で行います。



- <1> トリガ機能を使用するかどうかを設定する トリガ機能を使用する場合は、[トリガ機能を使用する]をチェックしてください。 デフォルトではチェックされていません。
- <2> トリガの動作を設定する トリガ検出時の動作を以下の中から選択します。
 - プログラムを停止する プログラムの実行を停止します。
 - 外部トリガを出力する E2 拡張インタフェースの外部トリガを出力します。

外部トリガを出力する場合は、以下の設定を行います。

- チャネルを選択する E2 拡張インタフェースの外部トリガを出力するチャネルを指定します。
- 出力する信号を選択する High パルスを出力するか,条件が成立している間 High レベルを出力し続けるかを選択します。 High パルスを出力する場合は,パルス幅も指定してください。
- 備考 外部トリガ出力機能についての詳細は、「消費電流チューニングソリューション(E2 エミュレータ、CS+編)アプリケーションノート」を参照してください。
- <3> トリガの条件を設定する トリガの条件として、電流値がしきい値以上、しきい値以下、範囲内、範囲外が選択可能です。

それぞれトリガとして検出したい電流値、または電流値の範囲を指定してください。

注意 電流値の範囲は、左側の入力欄の値が右側の入力欄の値より小さくなるようにしてください。

また、電流値の条件だけでなく、その条件を満たした時間に関するサブ条件(検出幅の条件)も設定できま す。検出幅を指定すると、一瞬だけ条件を満たすようなノイズを無視することができます。 検出幅の条件は、しきい値以上(条件が指定した時間以上続いた場合に検出する)、または範囲内(条件が 指定した時間の範囲内の場合に検出)から選択できます。

検出幅には、サンプリング時間の倍数を指定します。 実際に検出対象となる時間はサンプリング時間によって変化しますので注意してください。

(c) 通過ポイントを設定する

通過ポイントは、指定したソース行が実行された際にタイムスタンプを記録する機能です。 通過ポイントを設定することにより、消費電流測定のグラフ上で通過ポイントの実行時間と電流値の遷移をあ わせて確認することができます。

注意 通過ポイント使用時は、デバッグ・コンソールを使用しないでください。

通過ポイントはグラフ上で色付きの縦線として確認できます。



図 2.187 通過ポイント

通過ポイントの設定は,エディタ パネル上で開いた対象ソースの行にキャレットを合わせ,コンテキスト・ メニューの [通過ポイント設定] で行います。





現在設定されている通過ポイントの一覧は、消費電流測定パネル【E2】の下部に表示されます。 設定済みの通過ポイントの有効/無効の切り替えは、この一覧の左端のチェック・ボックス、またはエディタ パネル上のコンテキスト・メニューから行います。

図 2.189	通過ポイ	ン	トー覧
---------	------	---	-----

	カーソル	通過回激	位置	線色
2		8	r_cg_timer_user.c#153(0x001B5)	Maroon
		0	r_cg_timer_user.c#69	Gray
		0	r_cg_timer_user.c#209	Gray
		9	r_cc_timer_user.c#181(0x001A4)	Purple
2		1	r_main.c#78(0x00287)	Fuchsia
Π.		0	r_main.c#76	Gray

注意 1. 以下の場合,測定開始時に通過ポイントは設定できず,強制的に無効になります。

- 設定位置が NOP 命令上でない場合, NOP 命令上であってもデバッグ・ツールが NOP 命令 を検出できない場合

- プロパティ パネルの [デバッグ・ツール設定] タブ上の [システム] カテゴリ内において, 以下のいずれかのプロパティを [はい] に設定した場合
 - [実行開始前に指定ルーチンを実行する]
 - [ブレーク後に指定ルーチンを実行する]
 - [内蔵プログラム ROM を書き換えるプログラムをデバッグする]
- 注意 2. 通過ポイントは NOP 命令の位置に設定する必要があります。 プロジェクトで使用するコンパイラが CC-RX V2.08.00 以上、および設定位置がアクティブ・ プロジェクトに登録しているソース・ファイル上の場合は、測定前に自動で設定位置に NOP 命令を挿入し、測定を開始することができます。なお、プロパティ パネルの[ダウンロード・ ファイル設定] タブ上の [ダウンロード] カテゴリ内の [ダウンロード後に CPU をリセット する] プロパティが [はい] の場合、リビルド&ダウンロードでリセットされるため、注意し てください。
- (3) 測定を開始する E2 エミュレータの機能を使用してターゲット・システム消費電流測定を開始します。 消費電流測定パネル【E2】のツールバーの 🦓 , または 🙀 をクリックすることで測定が開始されます。

図 2.190 消費電流測定 パネルのツールバー【E2】

强 🕄 📵 🗐 🐓 🔜 🔲 🏧 🚺 🚺 🕨 通道ボイント 💿 🔍 💭 😼 🔒 🗙 📓 🗙 🎼

注意 メイン・ウインドウ上の 🍙 のクリックで開始する実行では, 消費電流は測定しません。

- (a)
 をクリックした場合 測定前のリビルド&ダウンロード行った後に、測定を開始します。
 ただし、デバイスが通過ポイントに対応していない場合や、アクティブ・プロジェクトのコンパイラが CC-RX V2.08.00 未満のバージョンの場合は、リビルド&ダウンロードを行いません。
 消費電流測定 パネル【E2】の[測定時に指定した通過ポイント用ビルド・オプションを今後も付加する]を チェックしている場合、測定前のリビルド&ダウンロードで指定したオプションがビルド・ツールの[消費電 流測定用の NOP 命令挿入のパラメータ]プロパティに反映されます。
- (b) をクリックした場合 測定前のリビルド&ダウンロードは行いません。
 リビルド&ダウンロードに伴うリセットを発生させずに測定したい場合はこちらを選択してください。
 なお、本ボタンは、デバイスが通過ポイントに対応していない場合やアクティブ・プロジェクトのコンパイラが CC-RX V2.08.00 未満のバージョンの場合は表示されません。
- (4) 測定を終了する 測定はプログラムの実行停止と同時に終了します。 プログラムの実行停止は、ブレークポイントやトリガ条件の設定、または をクリックすることで行うことができます。
- (5) グラフを表示する 測定終了と同時に、消費電流測定パネル【E2】のグラフ表示エリアに測定結果の電流値グラフが表示されます。 意図したとおりの電流値となっているか確認してください。

パネル上では、カーソル(赤い縦線)をドラッグすることで、カーソル位置の時間や電流値が情報表示エリアに 表示されます。 検索を使用すると、検索条件にヒットした位置にカーソルが移動しますので、その位置の時間や電流値が確認で きます。 マーカーA、マーカーB(緑の縦破線)をドラッグすることで、マーカー間の電流の平均値/最大値などを確認 することもできます。 また、通過ポイントを対象に検索することで、確認したい処理が実行された位置をすぐに表示することができま す。

2.20 デバッグ・コンソールの使用

この節では、標準入出力を実行するためのデバッグ・コンソールの使用方法について説明します。 プログラム内の標準ライブラリ関数、例えば、scanf 関数によるキーボード入力データの読み込み、printf 関数による データの出力は、次のデバッグ・コンソール パネルで行います。

図 2.191 デバッグ・コンソール パネル

デバッグ・コンソール	×
1	*
	Ŧ

なお、デバッグ・コンソールの各機能についての詳細は、デバッグ・コンソール パネルを参照してください。

デバッグ・コンソール パネルによる標準入出力を C/C++ プログラムで使用する場合は、プログラム内に低水準イン タフェース・ルーチンが必要です。

CS+ では、次のデバッグ・コンソール機能用のファイルを持つ、E1 / E20 用のサンプル・プロジェクト 「RX610_Tutorial_DebugConsole」を提供しています。

- 低水準インタフェース・ルーチン(C言語部分)用のファイル(lowsrc.c, lowsrc.h)

- 低水準インタフェース・ルーチン: open, close, read, write, および lseek 標準入出力により呼び出されます。
- 入出カライブラリの初期設定:_INIT_IOLIB
- オープンした全関数のクローズ:_CLOSEALL それぞれ、初期設定ルーチン・ファイル resetprg.c 内の初期設定ルーチン "PowerON_Reset_PC" により呼び 出されます。
- 低水準インタフェース・ルーチン(アセンブリ言語部分)用のファイル(lowlvl.src)
 - 1 文字の出力を行う関数: _charput
 - 1 文字の入力を行う関数: _charget それぞれ,低水準インタフェース・ルーチン "write", "read" により呼び出されます。
- main 関数を含むソース・ファイル(DebugConsole_Sample.c)
 - main 関数内で標準ライブラリ関数 scanf, および printf を呼び出します。
- 注意 【E1】【E20】【EZ Emulator】
- デバッグ・コンソールを使用する場合, プログラム中の charput, charget 関数内でシステムクロック を切り替えないでください。システム・クロックを切り替えるとエミュレータとマイクロコントローラ の送受信に影響を与え, データが正常に送受信できない恐れがあります。 デバッグ・コンソール機能を使用する場合は,データの送受信のために, ユーザ・プログラムのリアル タイム性が損なわれます。
- 備考 1. 低水準インタフェース・ルーチンについては,「CC-RX コンパイラ ユーザーズマニュアル」を参照し てください。
- 備考 2. 【シミュレータ】 標準入出力を実行する方式として、エミュレータと同様の方式とシミュレータ固有の方式をそれぞれサ ポートしています。シミュレータ固有方式では、標準入出力の他にファイル入出力を使用できます。 各サポート方式は、プロパティパネル上の[デバッグ・ツール設定]タブの[ストリーム入出力]【シ ミュレータ】カテゴリ内の[ストリーム入出力モードを選択]プロパティを指定することで切り替わり

ます。

シミュレータ方式の入出力機能については、「B.入出力機能」を参照してください。 デバッグ・コンソール機能のサンプル・プロジェクトを使用するには、次の手順で操作を行ってください。

- (1) サンプル・プロジェクトを読み込む
 - CS+ のスタート パネル 上の[サンプル・プロジェクトを読み込む]から,E1 / E20 用のサンプル・プロジェ クト「RX610_Tutorial_DebugConsole」を読み込みます。
 - 備考 " スタート パネル " についての詳細は, 「CS+ 統合開発環境 ユーザーズマニュアル プロジェクト 操作編」を参照してください。
- (2) 低水準インタフェース・ルーチン(アセンブリ言語部分)用のファイルを編集する【シミュレータ】 ストリーム入出力の方式をシミュレータモードとして使用する場合、低水準インタフェース・ルーチン(アセン ブリ言語部分)用のファイル(lowlvl.src)の内容をシミュレータ用に編集する必要があります。lowlvl.srcの内 容を以下のシミュレータ用のサンプルへ置き換えてください。

	.GLB	charput charget
SIM_IO	.EQU Oh .SECTION	P,CODE
;; ;cha	 rput:	
;		
_charpu	t:	
	MOV.L MOV B	#10_60f,K2 R1 [R2]
	MOV . L	#1220000h.R1
	MOV.L	# PARM, R3
	MOV.L	R2, [R3]
	MOV.L	R3, R2
	MOV.L	#SIM_IO,R3
	JSR	R3
	RTS	
;		
,		
, charge	t:	
	MOV.L	#1210000h,R1
	MOV.L	#IO_BUF,R2
	MOV.L	#PARM,R3
	MOV.L	R2,[R3]
	MOV.L	R3, R2
	MOV.L	#SIM_IO,R3
	JSR	R3
	MOV.L	#10_BUF, R2
	MOVU.B	[R2], R1
·		
; I/O 1	Buffer	
;		
	.SECTION	B, DATA, ALIGN=4
PARM:	.BLKL	1
	.SECTION	B_1, DATA
IO_BUF:	.BLKB .END	1

- (3) 使用するデバッグ・ツールを選択する 使用するデバッグ・ツールの選択/切り替えは、プロジェクト・ツリー パネル上の [品種名 デバッグ・ツール 名(デバッグ・ツール)] ノードを右クリックすることで表示されるコンテキスト・メニューにより行います。
 - 備考 デバッグ・ツールの選択についての詳細は、「2.3.1 使用するデバッグ・ツールを選択する」を参 照してください。
- (4) プロパティ パネルの設定を変更する【シミュレータ】
 シミュレータを使用する場合、プロパティ パネルの上の[デバッグ・ツール設定]タブの[ストリーム入出力]
 【シミュレータ】カテゴリ内において、次の設定を行います。

図 2.192 [ストリーム入出力] カテゴリ

⊿ ストリーム入出力

ストリーム入出力モードを選択	シミュレータモード
ストリーム入出力機能を使用する	いいえ
ストリーム入出力用アドレス	HEX 0

- (a) [ストリーム入出力モードを選択]
 - エミュレータと同じ低水準インタフェース・ルーチン(アセンブリ言語部分)用のファイルを使用する場合 は、[エミュレータモード]を選択してください(デフォルトでは[シミュレータモード]が指定されていま す)。
- (b) [ストリーム入出力機能を使用する] このプロパティは、[ストリーム入出力モードを選択] プロパティにおいて [シミュレータモード] を選択した場合にのみ表示されます。
 標準入出力またはファイル入出力を行う場合、[はい]を選択してください(デフォルトでは[いいえ]が指定されます)。
- (c) [ストリーム入出カ用アドレス] このプロパティは、[ストリーム入出カモードを選択] プロパティにおいて [シミュレータモード] を選択した場合にのみ表示されます。 低水準インタフェース・ルーチン(アセンブリ言語部分)に、シミュレータ用のサンプルを使用している場合は、"0"(デフォルト)を指定してください。
- (5) ダウンロードを実行する メイン・ウインドウ上の[デバッグ]メニュー→[ビルド&デバッグ・ツールへダウンロード]を選択します。 これにより、デバッグ・ツールと接続したのち、ビルド/ダウンロードを行います(「2.5.1 ダウンロードを実 行する」参照)。
- (6) デバッグ・コンソール パネルをオープンする
 デバッグ・コンソール パネルをオープンします。
- (7) プログラムを実行する プログラムを実行します(「2.9 プログラムの実行」参照)。
- (8) デバッグ・コンソールパネル上でデータの入出力を行う プログラムを実行することにより、デバッグ・コンソールパネル上でデータの入出力を行います。 本プログラムは、scanf 関数によりキーボードからパネル上へ入力した 10 個のデータを読み込み、入力データ を昇順にソートしたのち、printf 関数によりパネル上に出力します。





- main 関数内の printf 関数により、データを格納する配列 "a" の要素名をパネル上に出力します。
- キーボードによりパネル上へデータを入力すると、main 関数内の scanf 関数によりデータを読み込みます。

デバッグ・エンソール	R
デバッグ・エンソール (************************************	
### Data Input ### a[0]=0	÷

- main 関数内の printf 関数により、ソートしたデータをパネル上に出力します。

備考 デバッグ・コンソール パネルのコンテキスト・メニューから [ローカル・エコー・バック有効/ 無効]を選択し、ローカル・エコー・バック機能を無効状態にした場合、入力した文字は表示し ません (ローカル・エコー・バックを実行しません)。



2.21 入力値について

この節では、各パネル/ダイアログに値を入力する際の留意事項について説明します。

2.21.1 入力規約

各パネル/ダイアログへの入力規約を次に示します。

(1) 文字セット

入力を許可している文字セットは次のとおりです。

表 2.26 文字セットの一覧

文字セット	概要
ASCII	半角のアルファベット(英字)、半角の数字、および半角の記号
Shift-JIS	全角のアルファベット(英字), 全角の数字, 全角の記号, ひらがな, 全 角のカタカナ, 漢字, および半角のカタカナ
EUC-JP	全角のアルファベット(英字), 全角の数字, 全角の記号, ひらがな, 全 角のカタカナ, 漢字, および半角のカタカナ
UTF-8	全角のアルファベット(英字), 全角の数字, 全角の記号, ひらがな, 全 角のカタカナ, 漢字(中国語を含む), および半角のカタカナ
UTF-16 (Unicode)	全角のアルファベット(英字), 全角の数字, 全角の記号, ひらがな, 全 角のカタカナ, 漢字(中国語を含む), および半角のカタカナ

(2) エスケープ・シーケンス

入力を許可しているエスケープ・シーケンスは次のとおりです。

表 2.27 エスケープ・シーケンスの一覧

エスケープ・シーケンス	値	意味
¥0	0x00	null文字
¥a	0x07	アラート
¥b	0x08	バックスペース
¥t	0x09	水平タブ
¥n	0x0A	改行
¥ν	0x0B	垂直タブ
¥f	0x0C	フォーム・フィード
¥r	0x0D	キャリッジ・リターン
¥ "	0x22	ダブルクォーテーション
¥'	0x27	シングルクォーテーション
¥?	0x3F	疑問符 ?と入力された場合も疑問符として扱います。
11	0x5C	バックスラッシュ

(3) 数 值

数値を入力する際に許可している進数は次のとおりです。

進数表記	概要
2 進数	0b で始まり、0 ~ 1 の数値が続く数値 (英字の大文字/小文字については、不問)
8 進数	0 で始まり, 0 ~ 7 の数字が続く数値
10 進数	0以外で始まり、0~9の数字が続く数値
16 進数	0x で始まり,0~9の数字,および a ~ f の英字が続く数値 (英字の大文字/小文字については,不問) ただし,HEX マークが表示されている入力エリアでは,0x の接頭辞は必 要ありません。

(4) 式と演算子

式とは、定数、CPU レジスタ名、I/O レジスタ名、シンボル、およびこれらを演算子で結合したものを示します。

式には、アドレス式とウォッチ式があります。シンボルのアドレスを必要とする式をアドレス式、シンボルの値 を必要とする式をウォッチ式と呼びます。

(a) アドレス式と演算子

アドレス式では、シンボルのアドレスを使用して演算します。CPU レジスタ名が記述された場合のみ、値を 使用して演算します。

アドレス式の基本入力形式は次のとおりです。

式	説明
C/C++ 言語変数名 ^{注 1,注 2}	C/C++ 言語の変数のアドレス
ゴ[式 ^{注 3}]	配列のアドレス
<i>式</i> .メンバ名 ^{注 4}	構造体/共用体/クラスのメンバのアドレス
<i>式-</i> > メンバ名 ^{注 4}	ポインタの指し示す構造体/共用体/クラスのメンバのアドレス
ウォッチ式 .* キャスト式	メンバ変数へのポインタのアドレス
ウォッチ式 ->* キャスト式	メンバ変数へのポインタのアドレス
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタのアドレス
ラベル名 ^{注 5} , EQU シンボル名 ^注 ⁵ , [即値]	ラベルのアドレス,EQU シンボルの値,即値アドレス
整定数	アドレス

表 2.29 アドレス式の基本入力形式

注 1. C89, C99, C++ 言語の変数を表します。

- 注 2. C 言語変数の値がレジスタに割り付いている場合は、エラーになります。
- 注 3. インデックスとして入力された式は、ウォッチ式として解析します。
- 注 4. 基底クラスのメンバ変数を指定する場合、メンバ名の前にスコープを指定してください (例: variable.BaseClass::member)。
- 注 5. ラベル名または EQU シンボル名に "\$" が含まれている場合,名前を "{}" で囲んでください (例: {\$Label})。 "I" は虚数のキーワードとなるため,CPU レジスタの "I" を指定する場合は、":REG" を付加して ください(例: I:REG)。

また、「表 2.29 アドレス式の基本入力形式」から、次の演算子を用いたアドレス式を構成することができます。

式	説明
(式)	演算順序の指定
- <i>式</i>	符号反転
! <i>式</i>	論理否定
~ 式	ビット反転
<i>式 * 式</i> ^注	乗算
<i>式 / 式</i> ^注	除算
<i>式</i> % 式 ^注	剰余算
<i>式</i> + 式 ^注	加算
<i>式</i> - 式 ^注	減算
<i>式</i> & 式 ^注	ビットごとの論理積
<i>式</i> ^	ビットごとの排他的論理和
<i>式</i> 式 ^注	ビットごとの論理和

表 2.30 演算子を用いたアドレス式の構成

注 変数および関数は、変数、関数、整定数以外と演算子で結合することはできません(例:C言 語変数名 +I/O レジスタ名)。

(b) ウォッチ式と演算子

ウォッチ式ではシンボルの値を使用して演算します。値が存在しない場合のみ、シンボルのアドレスを使用して演算します(例:main()+1)。 ウォッチ式の基本入力形式は次のとおりです。

表 2.31 ウォッチ式の基本入力形式

式	説明
C/C++ 言語変数名 ^{注 1}	C/C++ 言語の変数の値
式[式]	配列の要素値
<u>式</u> .メンバ名 ^{注 2}	構造体/共用体/クラスのメンバ値
<i>式</i> -> メンバ名 ^{注 2}	ポインタの指し示す構造体/共用体/クラスのメンバ値
ウォッチ式 .* キャスト式	メンバ変数へのポインタの値
ウォッチ式 ->* キャスト式	メンバ変数へのポインタの値
* <i>±</i> ť	ポインタの変数の値
& <i>式</i>	配置アドレス
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名 ^{注 3} , EQU シンボル名 ^注 ³ , [即値]	ラベルの値,EQU シンボルの値,即値アドレスの値
整定数	整数の定数値
浮動定数	 浮動小数点の定数値
文字定数	文字定数值

注 1.

C89, C99, C++ 言語の変数を表します。

- 注 2. 基底クラスのメンバ変数を指定する場合、メンバ名の前にスコープを指定してください。 (例:variable.BaseClass::member)
- 注 3. ラベル名または EQU シンボル名に "\$" が含まれている場合,名前を "{}" で囲んでください (例: {\$Label})。 虚数の値には,大文字の "I" を掛けてください(例:1.0+2.0*I)。"I" は虚数のキーワードとな るため, CPU レジスタの "I" を指定する場合は,":REG" を付加してください(例: I:REG)。

また、「表 2.31 ウォッチ式の基本入力形式」から、次の演算子を用いたウォッチ式を構成することができます。次表の演算子は、C 言語仕様に従って式を解析します。

式	説明
(演算順序の指定
! <i> </i>	論理否定
~ <i>ヹ</i>	ビット反転
<i>式 * 式</i> ^注	乗算
<i>式 式</i> 注	除算
<i>式</i> %式 ^注	剰余算
<i>式</i> + 式 ^注	加算
<i>式 - 式</i> 注	減算
<i>式</i> & 式 ^注	ビットごとの論理積
<i>式 ^ 式</i> ^注	ビットごとの排他的論理和
式 式注	ビットごとの論理和

表 2.32 演算子を用いたウォッチ式の構成

- 注 変数および関数は、変数、関数、整定数以外と演算子で結合することはできません(例:C言 語変数名 +I/O レジスタ名)。
- **注意 1.** 異なるソース・ファイルに無名名前空間を記述し、その中に同名の変数を定義した場合、ウォッチパネルでは、最初に見つかる変数の情報を表示します。
- **注意 2.** 以下のプログラムに定義されたメンバ変数ポインタ "mp1" をウォッチ パネル,およびローカル変数 パネルに登録した場合,型名に "int Foo::*" ではなく "int *" と表示されます。

class Foo {
 int m1;
};
int Foo::*mp1 = &Foo::m1;

- 注意 3. 下記のような char 型の一次元配列がレジスタやメモリの複数個所に割り付いていた場合は、ウォッチ パネル、およびローカル変数 パネルに配列 "array" を登録しても値のカラムに文字列を表示できません。 char array[5] = "ABCD";
- **注意 4.** ローカル変数 パネルの [スコープ] にて "カレント"以外を選択中は、レジスタに割り付いた変数の値 は正しく表示できません。また、その変数の値を編集することもできません。
- **注意 5.** 以下の条件をすべて満たす変数を定義した場合、ウォッチパネル、ローカル変数パネルでは、対象の メンバ変数の割り付き位置文字列が変数全体の割り付き位置文字列で表示されます。

【条件】

- <1> 定義した変数が複数のアドレスやレジスタに割り付いている。 (アドレスカラムに2つ以上のアドレスやレジスタ名が表示される場合)
- <2> 変数に以下の型のメンバが定義されている。 構造体、クラス、配列、共用体のいずれか

【例】

```
struct Mem {
    long m_base;
};
struct Sample {
    long m_a;
    struct Mem m_b; <- 条件 <2> に該当
};
main () {
    struct Sample obj;
}
```

表示結果:

"obj"	_	{ R1:REG, R2:REG }	(struct Sample)
L m_a	0x00000000	{ R1:REG }	(long)
L m_b	-	{ R1:REG, R2:REG }	(struct Base)
L m_base	0x00000000	{ R2:REG }	(long)

注意 6. ウォッチ パネルで変数を他の型ヘキャストした場合、Cスタイルのキャストを行います。 このため、例えば仮想継承クラスの基底クラスへのキャスト結果は、プログラム中で記述したキャスト 結果と異なります。

```
class AAA [
   int m_aaa;
} objA;
class BBB : public AAA { //BBBは AAA を継承している
   int m bbb;
} objB;
class CCC { //CCC は AAA を継承していない
   int m ccc;
} objC
class AAA* pa = objA;
class BBB* pb = objB;
class CCC* pc = objC;
"(AAA*)pa" ... 使用可能
"(BBB*)pb" ... 使用可能
"(AAA*)pb" ... 使用可能
"(CCC*)pc" ... 使用可能
"(AAA*)pc" ...pc の指すアドレスを型 AAA の先頭アドレスとみなしてキャスト
プログラミング上のキャストイメージ: (AAA*)((void*)pc)
```

注意 7. ウォッチ パネルで 0x10000 を超えるサイズの変数は値,型,アドレスが「?」表示になります。

2.21.2 シンボル名の入力補完機能

シンボル名の入力補完とは、アドレス式などを入力する際に、プログラム中に存在するシンボル名のリストから1つ を選択することにより、ユーザのシンボル名の入力作業を補佐する機能です。

シンボル名のリストは、この機能に対応するテキスト・ボックスにおいて、目的のシンボル名の一部が入力されてい る状態で [Ctrl] + [Space] キーを押下することにより表示されます。リスト内において、目的のシンボル名をダブル クリックすることで(または、[↑] / [↓] キーによりシンボル名を選択したのち [Space] / [Enter] キーを押 下)、入力中のシンボル名が補完されます。

なお、この際に、[Space] / [Enter] キー以外のキーが押下された場合、または現在操作対象としているパネル/ ダイアログからフォーカスが移動した場合は、シンボル名のリストは消失します(シンボル名の入力補完は行われません)。

- **注意 1.** テキスト・ボックスにおいて、1 文字も入力されていない場合、または候補が1 つも存在しない場合は、シンボル名のリストは表示されません。
- **注意 2.** コンパイラの最適化により、関数の引数の型がプログラム上で定義した型と異なる型に変換されること があります(例: func (class Foo obj); → func (class Foo *obj);)。 この場合、シンボル名の入力補完機能では最適化後の関数名を出力します。
- 備考 シンボル名の入力時にこの機能を使用できるか否かは,該当するパネル/ダイアログの入力エリアの説 明を参照してください。

XTU1		
● 表記(N)- ● 体止時(お勧 sub ● (休止時(お勧 sub) ● (小し) ● (休止時(お勧 sub) ● (休止時(お勧 sub) ● (休止時(お勧 sub) ● (休止時(お助 s	サイズ表記(Z)・ [Ctrl] + [Space] キーを押下しま ¥sample¥DefaultBuild#sample.abs [*] \$ <u>a.h</u> 01(int.int) ¥sample¥DefaultBuild#sample.abs [*] \$ <u>a.h</u> 02(int.int) ま示されるリストの中から目的の シンボル名を選択します。	ます。 ****** **sample*DefaultBuild*sample.abs"\$sub03(int*)

図 2.193 シンボル名の入力補完機能

2.21.3 入力不備箇所に対するアイコン表示

CS+が提供する一部のダイアログでは、不正な文字列が入力された際、設定すべき値として誤っていることを示す アイコンを該当箇所に表示することにより、入力の不備を警告します。



2.22 E1/E20/EZ エミュレータとマイクロコントローラの相違点

E1/E20/EZ エミュレータとマイクロコントローラの相違点については、「E1/E20/E2 エミュレータ, E2 エミュレータ Lite ユーザーズマニュアル別冊(RX 接続時の注意事項)」を参照してください。



2.23 その他の使用上の注意事項【E1】【E20】【EZ Emulator】

使用上の注意事項については、「E1/E20/E2 エミュレータ, E2 エミュレータ Lite ユーザーズマニュアル別冊(RX 接続時の注意事項)」を参照してください。



A. ウインドウ・リファレンス

この付録では、CS+でデバッグを行う際に使用するウインドウ/パネル/ダイアログについての詳細を説明します。

A.1 説 明

次に、デバッグに関するウインドウ/パネル/ダイアログの一覧を示します。

表 A.1 ウインドウ/パネル/ダイアログ一覧

ウインドウ/パネル/ダイアログ名	機能概要
メイン・ウインドウ	プログラムの実行制御,および各パネルのオープン
プロジェクト・ツリー パネル	使用するデバッグ・ツールの選択
プロパティ パネル	プロジェクト・ツリー パネルで選択しているデバッグ・ツールにつ いて,詳細情報の表示,および設定の変更
メモリ パネル	メモリの値の表示、および値の変更
逆アセンブル パネル	メモリ値を逆アセンブルした結果の表示、ライン・アセンブル、お よび命令レベル・デバッグ
CPU レジスタ パネル	CPU レジスタ(プログラム・レジスタ/システム・レジスタ)の 内容の表示,および値の変更
IOR パネル	I/O レジスタの内容の表示,および値の変更
ローカル変数 パネル	ローカル変数の内容の表示、および値の変更
ウォッチ パネル	登録したウォッチ式の内容の表示、および値の変更
コール・スタック パネル	関数呼び出しのコール・スタック情報の表示
トレース パネル	デバッグ・ツールから取得したトレース・データの表示
イベント パネル	設定イベントの詳細情報の表示, 有効/無効の切り替え, および削 除
出力 パネル	ビルド・ツール/デバッグ・ツール/各プラグインから出力される メッセージ, または検索・置換 ダイアログ による一括検索を行っ た際の結果の表示
デバッグ・コンソール パネル	標準入出力
メモリ・マッピング ダイアログ	メモリ・マッピングの設定
ダウンロード・ファイル ダイアログ	ダウンロードする際のファイルの選択,およびダウンロード条件の 設定
アクション・イベント ダイアログ	アクション・イベントの設定
表示桁数設定 ダイアログ	メモリ パネルにおけるメモリ値の表示桁数の設定
アドレス・オフセット設定 ダイアログ	メモリ パネルにおけるアドレス表示のオフセット値の設定
メモリ初期化 ダイアログ	メモリの初期化
メモリ検索 ダイアログ	メモリの検索
印刷アドレス範囲設定 ダイアログ	逆アセンブル パネルにおける印刷範囲の設定
トレース検索 ダイアログ	トレース・データの検索
組み合わせ条件設定 ダイアログ【E1】 【E20】【EZ Emulator】	イベント組み合わせ条件の表示、および設定の変更



ウインドウ/パネル/ダイアログ名	機能概要
タイマ計測詳細設定 ダイアログ【E1】 【E20】【EZ Emulator】	タイマ・イベントに関する詳細情報の表示、および設定の変更
実行イベント詳細設定 ダイアログ	実行系イベントに関する詳細情報の表示、および設定の変更
アクセスイベント詳細設定 ダイアログ	アクセス系イベントに関する詳細情報の表示、および設定の変更
割り込みイベント詳細設定 ダイアログ【シ ミュレータ】	割り込みイベントに関する詳細情報の表示、および設定の変更
COM ポート設定 ダイアログ	デバッグ・コンソール パネルにおける COM ポートの設定
スクロール範囲設定 ダイアログ	メモリ パネル/逆アセンブル パネルにおけるスクロール範囲の設 定
指定位置へ移動 ダイアログ	指定した位置にキャレットを移動
データ保存 ダイアログ	各パネルの表示内容、およびアップロード・データの保存
関数変数アクセス表 パネル	変数にアクセスしている関数の直交表での表示
消費電流測定 パネル【E2】	消費電流測定ソリューションの機能の中心となるパネル
消費電流測定条件設定 ダイアログ【E2】	消費電流測定条件の設定
消費電流測定トリガ条件設定 ダイアログ 【E2】	消費電流測定時のトリガ条件の設定
消費電流測定検索 ダイアログ【E2】	消費電流測定 パネル【E2】の検索
電流値保存 ダイアログ【E2】	消費電流測定時の電流値データの保存



メイン・ウインドウ

CS+ を起動した際,最初にオープンするウインドウです。 デバッグを行う際は,このウインドウからプログラムの実行制御,および各パネルのオープン操作を行います。

図 A.1 メイン・ウインドウ



ここでは、次の項目について説明します。

す。

- [オープン方法]

- [各エリアの説明]

[オープン方法]

- Windows の [スタート] → [すべてのプログラム] → [Renesas Electronics CS+] → [CS+ for CC(RL78,RX,RH850)] を選択
- 備考 Windows 8.1 の場合は,スタート画面の [CS+ for CC (RL78,RX,RH850)] を選択してください。 Windows 10 の場合は,Windows の [スタート] メニューから [すべてのアプリ] → [Renesas Electronics CS+] → [CS+ for CC (RL78,RX,RH850)] を選択してください。

[各エリアの説明]

- (1) メニューバー
 デバッグ関連のメニュー項目は次のとおりです。
 備考 各メニューから引き出される項目は、ユーザ設定 ダイアログでカスタマイズすることができま
 - (a) [表示] [表示]メニューの各項目,および機能は次のとおりです(デフォルト)。

ウォッチ	ウォッチ パネルをオープンするために、次のカスケード・メニューを表示し ます。 ただし、デバッグ・ツールと切断時は無効となります。
 ウォッチ 1	ウォッチ パネル(ウォッチ 1)をオープンします。
ウォッチ 2	ウォッチ パネル(ウォッチ 2)をオープンします。
ウォッチ 3	ウォッチ パネル(ウォッチ 3)をオープンします。
ウォッチ 4	ウォッチ パネル(ウォッチ 4)をオープンします。
ローカル変数	ローカル変数 パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
コール・スタック	コール・スタック パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
メモリ	メモリ パネルをオープンするために、次のカスケード・メニューを表示しま す。
L ₁	ただし、デバッグ・ツールと切断時は無効となります。
メモリ 1	メモリ パネル(メモリ 1)をオープンします。
メモリ2	メモリ パネル(メモリ 2)をオープンします。
メモリ 3	メモリ パネル(メモリ 3)をオープンします。
メモリ4	メモリ パネル(メモリ 4)をオープンします。
IOR	IOR パネルをオープンします。 ただし,デバッグ・ツールと切断時は無効となります。
CPU レジスタ	CPU レジスタ パネルをオープンします。 ただし,デバッグ・ツールと切断時は無効となります。
トレース	トレース パネルをオープンします。 ただし, デバッグ・ツールと切断時は無効となります。
逆アセンブル	逆アセンブル パネルをオープンするために,次のカスケード・メニューを表示します。 ただし,デバッグ・ツールと切断時は無効となります。
逆アセンブル 1	逆アセンブル パネル(逆アセンブル 1)をオープンします。
逆アセンブル 2	逆アセンブル パネル(逆アセンブル 2)をオープンします。
逆アセンブル 3	逆アセンブル パネル(逆アセンブル 3)をオープンします。
逆アセンブル 4	逆アセンブル パネル(逆アセンブル 4)をオープンします。
イベント	イベント パネルをオープンします。 ただし、デバッグ・ツールと切断時は無効となります。
デバッグ・コンソール	デバッグ・コンソール パネルをオープンします。 ただし,デバッグ・ツールと切断時は無効となります。
現在の PC 位置を開く	カレント PC 位置(PC レジスタ値)をエディタ パネルで表示します。 ただし,デバッグ・ツールと切断時は無効となります。
ジャンプ前の位置へ戻る	定義箇所へジャンプ(「2.6.2.4 シンボル定義箇所へ移動する」参照)する前の位置へ戻ります。
ジャンプ先の位置へ進む	[ジャンプ前の位置へ戻る]を実行する前の位置へ進みます。
タグ・ジャンプ	エディタ パネル/出力 パネルにおいて、キャレットのある行にファイル名/ 行/桁の情報がある場合、該当するファイルの該当行/該当桁へジャンプし ます。

CS+ V8.06.00

(b) [デバッグ]
 [デバッグ] メニューの各項目,および機能は次のとおりです(デフォルト)。

-		
ラン	゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙゙	アクティブ・プロジェクトで現在選択しているデバッグ・ツールに,指定され たファイルをダウンロードします。 デバッグ・ツールと切断時の場合は,自動的にデバッグ・ツールに接続し,ダ ウンロードを実行します。 ただし,プログラム実行中,またはビルド(ラピット・ビルドを除く)実行中 は無効となります。
н С	゙゙ヹ゙゙ルド&デバッグ・ツー ゚゚゚ヽヘダウンロード	プロジェクトのビルドを行い、ビルド後にアクティブ・プロジェクトで現在選 択しているデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は、自動的にデバッグ・ツールに接続し、ダ ウンロードを実行します。 ただし、プログラム実行中、またはビルド(ラピット・ビルドを除く)実行中 は無効となります。 なお、ビルドに失敗した場合、ダウンロードは実行しません。
1, 33	リビルド&デバッグ・ ノールヘダウンロード	プロジェクトのリビルドを行い、リビルド後にアクティブ・プロジェクトで現 在選択しているデバッグ・ツールにダウンロードを実行します。 デバッグ・ツールと切断時の場合は、自動的にデバッグ・ツールに接続し、ダ ウンロードを実行します。 ただし、プログラム実行中、またはビルド(ラピット・ビルドを除く)実行中 は無効となります。 なお、リビルドに失敗した場合、ダウンロードは実行しません。
7	「バッグ・ツールへ接続	アクティブ・プロジェクトで現在選択しているデバッグ・ツールに接続しま す。 ただし、デバッグ・ツールと接続時、ビルド(ラピット・ビルドを除く)実行 中、またはサポート範囲外のバージョンのコンパイラを使用している場合は無 効となります。
거 【 【	ヾット・プラグイン E1(JTAG)】 E20(JTAG)】	現在実行中のターゲット・システムをデバッグするために、アクティブ・プロジェクトで現在選択しているデバッグ・ツールとホット・プラグイン接続します(「2.4.3 CS+にデバッグ・ツールをホットプラグイン接続する【E1(JTAG)】 【E20(JTAG)】」参照)。 ただし、デバッグ・ツールと接続時は無効となります。
ラフ	⁼゙バッグ・ツールから ╯ップロード …	メモリ内容をファイルに保存するためのデータ保存 ダイアログをオープンしま す。 ただし、プログラム実行中、ビルド(ラピット・ビルドを除く)実行中、また はデバッグ・ツールと切断時は無効となります。
ラ胀	^デ バッグ・ツールから切 f	現在接続中のデバッグ・ツールとの通信を切断します。 ただし、プログラム実行中、ビルド(ラピット・ビルドを除く)実行中、また はデバッグ・ツールと切断時は無効となります。
仮ル	5用するデバッグ・ツー /	使用するデバッグ・ツールを選択するためのカスケード・メニューを表示します。 なお、プロジェクトで選択しているマイクロコントローラの種類により、表示 されるデバッグ・ツールは異なります。
	RX E2	E2 を使用します。
	RX E2 Lite	E2 Lite を使用します。
	RX E1(Serial)	E1 を Serial 通信方式で使用します。
	RX E1(JTAG)	E1 を JTAG 通信方式で使用します。
	RX E20(Serial)	E20 を Serial 通信方式で使用します。
	RX E20(JTAG)	E20 を JTAG 通信方式で使用します。
Ì	RX EZ Emulator	EZ Emulator を使用します。
	RX シミュレータ	シミュレータを使用します。



_		
傦	亭止	現在実行中のプログラムを強制的に停止します。 ただし,プログラム停止時,またはデバッグ・ツールと切断時は無効となりま す。
Ę	尾行	プログラムをカレント PC 位置から実行し,設定されているブレーク・イベン トの条件が成立した場合,実行中のプログラムを停止します。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,また はデバッグ・ツールと切断時は無効となります。
5	ブレークせずに実行	プログラムをカレント PC 位置から実行し,設定されているブレーク・イベン ト/アクション・イベントを無視してプログラムの実行を続けます。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,また はデバッグ・ツールと切断時は無効となります。
7	ミテップ・イン	カレント PC 位置からステップ実行し ^注 ,各パネルの内容を更新します。 関数呼び出しの場合は,呼び出された関数の先頭で停止します。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,また はデバッグ・ツールと切断時は無効となります。
7	ミテップ・オ ーバー	カレント PC 位置からステップ実行し ^注 , 各パネルの内容を更新します。 サブルーチン分岐命令による関数呼び出しの場合は, その関数内のソース行/ 命令すべてを1ステップとみなして実行し, 関数から戻る箇所まで実行します (サブルーチン分岐命令を実行したときと同じネストになるまで, ステップ実 行します)。 なお,サブルーチン分岐命令以外の場合, [ステップ・イン] の選択と同じ動 作となります。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,また はデバッグ・ツールと切断時は無効となります。
	リターン・アウト	現在の関数からリターンするまで(呼び出し元関数に戻るまで)実行します ^注 。 ただし、プログラム実行中、ビルド(ラピット・ビルドを除く)実行中、また はデバッグ・ツールと切断時は無効となります。
С	PUリセット	CPU をリセットします(プログラムは実行しません)。 ただし、ビルド(ラピット・ビルドを除く)実行中、またはデバッグ・ツール と切断時は無効となります。
	リスタート	CPU をリセットしたのち,リセット番地からプログラムを実行します。 ただし,ビルド(ラピット・ビルドを除く)実行中,またはデバッグ・ツール と切断時は無効となります。
うぞ	デバッグ・ツールの状態 E巻き戻す	最後に自動保存されたデバッグ・ツールの状態に巻き戻します。 なお、巻き戻されるデータは読み書き可能なメモリとレジスタの値に限られま す。 このデバッグ機能を使用するにはオプション・ダイアログで設定する必要があ ります。
ラ 伊	デバッグ・ツールの状態 R存	デバッグ・ツールの状態の保存と復帰に関するメニューです。 なお、保存されるデータは読み書き可能なメモリとレジスタの値に限られま す。
	デバッグ・ツールの状 態復帰 n	デバッグ・ツールの状態を n 番目のデータ・ファイルから復帰します。
	デバッグ・ツールの状 態保存 n	現在のデバッグ・ツールの状態をn番目のデータとしてファイルに保存します。

ステップ実行には、ソース・レベル単位と命令レベル単位の実行方法があります。 注 詳細は、「2.9.3 プログラムをステップ実行する」を参照してください。

 (2) デバッグ・ツールバー
 デバッグ・ツールバーは、プログラムの実行を制御するためのコマンドをまとめたボタン群です。 各ボタン,および機能は次のとおりです(デフォルト)。

各ツールバーのボタンは、ユーザ設定 ダイアログでカスタマイズすることができます。また、同 備考 1. ダイアログにより、新規にツールバーを作成することもできます。

備考 2.	ツールバー上を右クリックすることで表示されるコンテキスト・メニューにより,ツールバー上 に表示/非表示するグループを選択することができます。
	プロジェクトのビルドを行い、ビルド後にアクティブ・プロジェクトのデバッグ・ツールにダ ウンロードを実行します。 デバッグ・ツールと切断時の場合は、自動的にデバッグ・ツールに接続し、ダウンロードを実 行します。 ただし、ビルドに失敗した場合、ダウンロードは実行されません。
D.	[テハック] メニュー→ [ヒルト& テハック・ツールヘタウンロート] の選択と同等です。 アクティブ・プロジェクトのデバッグ・ツールに,指定されたファイルをダウンロードします。 デバッグ・ツールと切断時の場合は,自動的にデバッグ・ツールに接続し,ダウンロードを実 行します。 ただし,プログラム実行中,またはビルド(ラピット・ビルドを除く)実行中は無効となりま す。 [デバッグ] メニュー→ [デバッグ・ツールヘダウンロード] の選択と同等です。
H)	CPU をリセットします(プログラムは実行しません)。 ただし、ビルド(ラピット・ビルドを除く)実行中、またはデバッグ・ツールと切断時は無効 となります。 [デバッグ] メニュー→ [CPU リセット] の選択と同等です。
	現在実行中のプログラムを強制的に停止します。 ただし, プログラム停止時, またはデバッグ・ツールと切断時は無効となります。 [デバッグ] メニュー→ [停止] の選択と同等です。
	プログラムをカレント PC 位置から実行し,設定されているブレーク・イベントの条件が成立 した場合,実行中のプログラムを停止します。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,またはデバッグ・ ツールと切断時は無効となります。 [デバッグ]メニュー→[実行]の選択と同等です。
	プログラムをカレント PC 位置から実行し,設定されているブレーク・イベント/アクショ ン・イベントを無視してプログラムの実行を続けます。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,またはデバッグ・ ツールと切断時は無効となります。 [デバッグ]メニュー→ [ブレークせずに実行]の選択と同等です。
н.)	CPU をリセットしたのち, リセット番地からプログラムを実行します。 ただし, ビルド(ラピット・ビルドを除く)実行中, またはデバッグ・ツールと切断時は無効 となります。 [デバッグ] メニュー→ [リスタート] の選択と同等です。
91 	カレント PC 位置からでステップ実行し ^注 ,各パネルの内容を更新します(ステップ・イン実 行)。 関数呼び出しの場合は,呼び出された関数の先頭で停止します。 ただし,プログラム実行中,ビルド(ラピット・ビルドを除く)実行中,またはデバッグ・ ツールと切断時は無効となります。 [デバッグ]メニュー→ [ステップ・イン]の選択と同等です。
	カレント PC 位置からステップ実行し ^注 , 各パネルの内容を更新します(ステップ・オーバー 実行)。 サブルーチン分岐命令による関数呼び出しの場合は、その関数内のソース行/命令すべてを1 ステップとみなして実行し、関数から戻る箇所まで実行します(サブルーチン分岐命令を実行 したときと同じネストになるまで、ステップ実行します)。 なお、サブルーチン分岐命令以外の場合, だなンのクリックと同じ動作となります。 ただし、プログラム実行中、ビルド(ラビット・ビルドを除く)実行中、またはデバッグ・ ツールと切断時は無効となります。 [デバッグ] メニュー→ [ステップ・オーバー]の選択と同等です。
2	現在の関数からリターンするまで(呼び出し元関数に戻るまで)実行します ^注 (リターン・ア ウト実行)。 ただし、プログラム実行中、ビルド(ラピット・ビルドを除く)実行中、またはデバッグ・ ツールと切断時は無効となります。 [デバッグ]メニュー→ [リターン・アウト]の選択と同等です。

現在接続中のデバッグ・ツールとの通信を切断します。 ただし、ビルド(ラピット・ビルドを除く)実行中、またはデバッグ・ツールと切断時は無効 となります。 [デバッグ]メニュー→ [デバッグ・ツールから切断]の選択と同等です。

注 ステップ実行には、ソース・レベル単位と命令レベル単位の実行方法があります。 詳細は、「2.9.3 プログラムをステップ実行する」を参照してください。

- パネル表示エリア
 各種パネルを表示するエリアです。
 表示内容についての詳細は、各パネルの項を参照してください。
- (4) ステータス・バー ステータス・バーは、次の情報を表示します。

図 A.2 ステータス・バー



(a) ステータス・メッセージ次のメッセージを表示します。

- 選択しているメニュー項目の簡易説明

- パネル/ダイアログにおいて入力値が不正な場合のメッセージ

- 検索・置換 ダイアログにより検索した際に、指定文字列が見つからなかった場合のメッセージ

- ブレークした際のブレーク要因(「2.10 プログラムの停止(ブレーク)」参照)

- (b) フォーカス・パネルのステータス情報 現在フォーカスのあるパネルのステータス情報(キャレット位置や上書き/挿入モードなどの情報)を表示し ます。 ただし、ステータス情報を持たないパネルの場合は非表示となります。
- (c) 実行状態

プログラムの現在の実行状態を次のアイコンと文字列で示します。 ただし、デバッグ・ツールと切断時の場合は非表示となります。

プログラムの状態	表示内容
実行中	▶ RUN
停止中	BREAK
ステップ実行中	STEP

(d) CPU 状態【E1(JTAG)】【E20(JTAG)】

デバッグ・ツールの現在の CPU の状態を表示します。 なお、同時に複数の状態になっている場合は "&" で区切って状態を列挙して表示します。 ただし、デバッグ・ツールと切断時の場合は非表示となります。

デバッグ・ツール	表示内容	CPU 状態	
E1(JTAG) E20(JTAG)	Reset	リセット状態	
	PowOff	ターゲットに電源が供給されていない状態	
	Sleep	スリープ・モード中	
	Standby	スタンバイ・モード中	

備考 CPU が上記以外の状態の場合は何も表示しません。

(e) カレント PC 位置 現在のカレント PC 位置の値を 16 進数で表示します。

なお、プログラム実行中の表示内容は、プロパティ パネルの [デバッグ・ツール設定] タブ上における [レジスタ] カテゴリ内の設定によって次のように切り替わります。

プログラムの状態	[実行中に PC 表示を行う] プロパティの状態	表示内容	
停止中	_	現在のカレント PC 位置	
実行中はい		- [PC 表示更新間隔 [ms]] プロパティにて設定し た更新間隔ごとのカレント PC 位置 ^{注 1}	
	いいえ	実行中	

また、このエリアにマウスを重ねることにより、次の情報をポップアップ表示します。

- カレント PC : 0x *カレント PC 値*(*ソース名* # 行数^{注 2})

ただし、デバッグ・ツールと切断時の場合は非表示となります。

注 1. 【RX100 シリーズ】 プログラム実行中の PC 表示機能をサポートしていないため, PC 値は表示されません。

注 2. 情報の取得が不可能な場合は、"シンボル名 + オフセット値"となります。

(f) デバッグ・ツールとの接続状態

現在のデバッグ・ツールとの接続状態を次のアイコンと文字列で示します。

接続状態	表示内容	
接続中	💷 デバッグ・ツール名	
切断中		

(g) Run-Break タイマ結果

Run-Break タイマの計測結果(「2.14.2 実行開始から停止までの実行時間を計測する」参照)を表示します。 表示単位は計測結果に依存します。

ただし、デバッグ・ツールと切断時の場合は非表示となります。

状態	表示内容
計測していない状態	未計測
計測中	計測中
オーバフローした場合	OVERFLOW

備考 EZ Emulator では Run-Break タイマ機能をサポートしていませんので,常に"未計測"と表示 されます。

(h) デバッグ・ツールの状態

現在のデバッグ・ツールの各機能の状態を次のアイコンで示します。 ただし、デバッグ・ツールと切断時の場合は非表示となります。

機能	動作中	停止中(使用する)	使用しない
トレース	N	<u>\$</u>	19 g
タイマ	ø	<u>0</u>	_
カバレッジ【シミュレータ】 【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】			

注意 1. 【E1】【E20】【EZ Emulator】

トレース機能は、対象アイコンをクリックしても"使用しない"状態にはなりません。

注意 2. 【E20(JTAG)】

トレース機能のアイコンをクリックすることにより, プロパティ パネルの [デバッグ・ツール 設定] タブ上の [トレース] カテゴリ内 [トレース機能の用途] プロパティの設定内容が, ト レース/リアルタイム RAM モニタと切り替わります。

注意 3. 【シミュレータ】

トレース機能およびカバレッジ機能は、プログラムが停止中の場合、対象アイコンをクリック することにより、使用する/使用しないの状態を変更することができます。 また、プロパティパネルの[デバッグ・ツール設定] タブにおける次のプロパティの設定内容 が、はい/いいえと切り替わります。

- [トレース] カテゴリ→ [トレース機能を使用する] プロパティ
- [カバレッジ]【シミュレータ】カテゴリ→ [カバレッジ機能を利用する] プロパティ


プロジェクト・ツリー パネル

プロジェクトの構成要素(マイクロコントローラ,ビルド・ツール,デバッグ・ツールなど)をツリー形式で表示します。 なお、使用するデバッグ・ツールの選択/切り替えは、このパネル上で行います。

図 A.3 プロジェクト・ツリー パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [プロジェクト・ツリー] を選択

[各エリアの説明]

(1) プロジェクト・ツリー エリア プロジェクトの構成要素を次のノードでツリー表示します。



ノード	説明
<i>品種名 デバッグ・ツール名</i> (デバッグ・ツー ル)	 - 品種名 選択しているマイクロコントローラの品種名(RX)を表示します。 - デバッグ・ツール名 プロジェクトで使用するデバッグ・ツール名(E2, E2 Lite, E1(Serial), E1(JTAG), E20(Serial), E20(JTAG), EZ Emulator, シミュレータ)を表示します。 新規プロジェクト作成時は、シミュレータが設定されます。

ノードを選択すると、その詳細情報(プロパティ)がプロパティ パネルに表示され、設定の変更を行うことが できます(プロパティ パネルがオープンしていない場合は、ノードをダブルクリックすることでオープンしま す)。

[コンテキスト・メニュー]

使用するデバッグ・ツール		使用するデバッグ・ツールを選択するためのカスケード・メニューを表示します。 なお、プロジェクトで選択しているマイクロコントローラの種類により、表示されるデバッグ・ツールは異なります(「表 2.1 マイクロコントローラの 種類と接続可能なデバッグ・ツールの関係」参照)。		
	<i>品種名</i> E2	E2 を使用します。		
	<i>品種名</i> E2 Lite	E2 Lite を使用します。		
	<i>品種名</i> E1(Serial)	E1 をシリアル通信方式で使用します。		
	品種名 E1(JTAG)	E1 を JTAG 通信方式で使用します。		
	<i>品種名</i> E20(Serial)	E20 をシリアル通信方式で使用します。		
	<i>品種名</i> E20(JTAG)	E20 を JTAG 通信方式で使用します。		
	品種名 EZ Emulator	EZ Emulator を使用します。		
品種名 シミュレータ		シミュレータを使用します。		
- 1	[゚] ゚゚ロパティ	選択しているデバッグ・ツールのプロパティをプロパティ パネルに表示しま す。		



プロパティ パネル

プロジェクト・ツリー パネルで選択しているデバッグ・ツールについて、カテゴリ別に詳細情報の表示、および設定 の変更を行います。

図 A.4 プロパティ パネル(E20(JTAG)を選択した場合の例)

	プロパティ	×
(1)—	A RX E20(JTAG) のプロパティ	P -+
(1)	プロパティ	2048 128 32 EXTAL 12.5000 25.0000 CペVえ 2048 12.5000 25.0000 CペVえ 000 100-ドを16進32桁で指定 100 1024 シングルチップモード シングルチップモード シングルチップモード レittle=endianデータ
	エンディアン メ 外部フラッシュ	Little-endianデータ
	▶ 外部フラッシュ定義ファイル	[4]
	内蔵ROM/RAM	
(3)	接続用設定 (デバッグ・ツール設定 / ダウンロード・ファ・	(川設定 / フック処理設定 / ・

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[編集] メニュー (プロパティ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- プロジェクト・ツリー パネルにおいて、使用する [*品種名 デバッグ・ツール名*(デバッグ・ツール)] ノードを選 択したのち、[表示] メニュー、またはコンテキスト・メニューより [プロパティ] を選択
- プロジェクト・ツリー パネルにおいて,使用する [*品種名 デバッグ・ツール名* (デバッグ・ツール)] ノードをダ ブルクリック

RENESAS

備考 このパネルがすでにオープンしている場合, プロジェクト・ツリー パネル上において, 使用する [*品 種名 デバッグ・ツール名*(デバッグ・ツール)] ノードを選択することにより, 選択したデバッグ・ ツールの詳細情報を表示します。

[各エリアの説明]

- (1) 対象名エリア
 プロジェクト・ツリー パネルで選択しているデバッグ・ツールの名称を表示します。
- (2) 詳細情報表示/変更エリア プロジェクト・ツリー パネルで選択しているデバッグ・ツールの詳細情報を、カテゴリ別のリスト形式で表示 し、設定の変更を直接行うことができるエリアです。
 マークは、そのカテゴリ内に含まれているすべてのプロパティ項目が展開表示されていることを示し、また、 田マークは、カテゴリ内のプロパティ項目が折りたたみ表示されていることを示します。展開/折りたたみ表 示の切り替えは、このマークのクリック、またはカテゴリ名のダブルクリックにより行うことができます。 なお、各プロパティ項目設定欄内に表示される HEX マークは、16 進数入力専用のテキスト・ボックスである ことを示します。 カテゴリ、およびそれに含まれるプロパティ項目の表示内容/設定方法についての詳細は、該当するタブの項を 参照してください。
- (3) タブ選択エリア

タブを選択することにより、詳細情報を表示するカテゴリが切り替わります。 このパネルには、次のタブが存在します(各タブ上における表示内容/設定方法についての詳細は、該当するタ ブの項を参照してください)。

- [接続用設定] タブ
- [デバッグ・ツール設定] タブ
- [ダウンロード・ファイル設定] タブ
- [フック処理設定] タブ

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。	
切り取り	プロパティの値を編集中の場合、選択している文字列を切り取ってクリップ・ ボードに移動します。	
コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。	
貼り付け	プロパティの値を編集中の場合,クリップ・ボードの内容を挿入します。	
削除	プロパティの値を編集中の場合,選択している文字列を削除します。	
すべて選択	プロパティの値を編集中の場合,選択しているプロパティの値の文字列をすべ て選択します。	

[[編集] メニュー (プロパティ パネル専用部分)]

[コンテキスト・メニュー]

【文字列編集中以外の場合】

デフォルトに戻す	選択しているプロパティ項目の設定値をデフォルトに戻します。
すべてデフォルトに戻す	現在選択しているタブ上の設定値をすべてデフォルトに戻します。

【文字列編集中の場合】

元に戻す	直前に行ったプロパティの値の編集作業を取り消します。	
切り取り	プロパティの値を編集中の場合、選択している文字列を切り取ってクリップ・ ボードに移動します。	



コピー	選択しているプロパティの値の文字列をクリップ・ボードにコピーします。
貼り付け	プロパティの値を編集中の場合,クリップ・ボードの内容を挿入します。
削除	プロパティの値を編集中の場合、選択している文字列を削除します。
すべて選択	プロパティの値を編集中の場合,選択しているプロパティの値の文字列をすべ て選択します。



[接続用設定] タブ

[接続用設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [内蔵 ROM/RAM]
- (2) [エンディアン]【シミュレータ】
- (3) [クロック]
- (4) [エミュレータとの接続]【E1】【E20】
- (5) [ターゲット・ボードとの接続]【E1】【E20】【EZ Emulator】
- (6) [E2 拡張インタフェース]【E2】
- (7) [フラッシュ] 【E1】 【E20】 【EZ Emulator】
- (8) [CPUの動作モード]【E1】【E20】【EZ Emulator】
- (9) [外部フラッシュ]【E1】【E20】
- (10) [周辺機能シミュレーション]【シミュレータ】

[各カテゴリの説明]

(1) [内蔵 ROM/RAM]

内蔵 ROM/RAM に関する詳細情報の表示を行います。

内蔵 ROM サイズ [K バ	選択しているマイクロコントローラの内蔵 ROM サイズを表示します。			
1 F]	デフォルト	- ROM 内蔵品の場合 <i>選択しているマイクロコントローラの内蔵 ROM サイズ</i>		
		- ROM レス品の場合 <i>0</i>		
	変更方法			
内蔵 RAM サイズ [K バ	選択しているマイクロコントローラの内蔵 RAM サイズを表示します。			
1 F]	デフォルト	選択しているマイクロコントローラの内蔵 RAM サイズ		
	変更方法	変更不可		
データフラッシュ・メ モリ・サイズ [K バイト	選択している 示します。	るマイクロコントローラのデータフラッシュ・メモリ領域のサイズを表		
] 【E1】【E20】【EZ Emulator】	デフォルト	<i>選択しているマイクロコントローラのデータフラッシュ・メモリ・ サイズ</i>		
	変更方法	変更不可		

(2) [エンディアン] 【シミュレータ】

CPU エンディアンに関する詳細情報の表示、および設定の変更を行います。

CPU エンディアン	マイクロコントローラのエンディアンを指定します。 ビルド・ツールのプロパティで設定されたエンディアン情報を取得して表示しま す。		
	デフォルト	Little-endian データ	
	変更方法	- ビルド&デバッグ・ツールの場合 変更不可	
		- デバッグ・ツールのみの場合 ドロップダウン・リストによる選択 ただし, デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	- ドロップダウン・リストによる次のいずれか Little-endian データ, Big-endian データ	

(3) [クロック]

クロックに関する詳細情報の表示、および設定の変更を行います。

メイン・クロック・ ソース	- 内蔵 HOCO 搭載品の場合 メイン・クロックを EXTAL 周波数と内蔵 HOCO から指定します。				
[E1] [E20] [EZ Emulator]	- 内蔵 HOCO 非搭載品の場合 メイン・クロックとして EXTAL 周波数を表示します。				
	デフォルト	EXTAL			
	変更方法	ドロップダウン・リストによる選択 ただし, デバッグ・ツールと切断中の場合のみ変更可			
	指定可能值	EXTAL	メイン・クロックが EXTAL 周波数として動作 します		
		НОСО	メイン・クロックが内蔵 HOCO として動作し ます。		
メイン・クロック周波 数 [MHz] 【E1】【E20】【EZ	EXTAL 周波 なお, このつ Emulator】]	数を MHz 単位で指定 l プロパティは [メイン・ プロパティにおいて [E	ンます。 クロック・ソース 【E1】【E20】【EZ EXTAL] を指定した場合のみ有効となります。		
Emulator	デフォルト	空欄			
	変更方法	キーボードからの直接入力 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能值	0.0001~99.9999(単位:MHz)			
動作周波数 [MHz]	動作周波数(ICLK)を MHz 単位で指定します。				
Enulator	デフォルト	空欄			
	変更方法	キーボードからの直接入力			
	指定可能值	0.0001 ~ 999.999 (.0001 ~ 999.999(単位:MHz)		
内蔵フラッシュ・メモ リ書き換え時のクロッ	内蔵フラッシュ・メモリを書き換える際, デバッガによるメイン・クロック・ ソースの操作を許可するか否かを指定します。				
ク操作を許可する 【E1】【E20】【EZ	デフォルト	いいえ			
Emulator】	変更方法	ドロップダウン・リストによる選択 ただし, デバッグ・ツールと切断中の場合のみ変更可			
	指定可能值	はい	メイン・クロック・ソースの操作を許可しま す。		
		いいえ	メイン・クロック・ソースの操作を許可しま せん。		
システムクロック	CPU の動作周波数を指定します。				
(IULK) 周波釵 [MHZ] 【シミュレータ】	デフォルト	選択しているマイクロコントローラに依存			
	変更方法	キーボードからの直接入力 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能值	能値 1 ~ 1000 の整数(単位:MHz)			

(4) [エミュレータとの接続] 【E1】 【E20】

	エミュレータシリアル No.	接続するエミュレータのシリアル番号を選択します。 ^注		
		デフォルト	空欄	
		変更方法	ドロップダウン・リストによる選択 ただし, デバッグ・ツールと切断中の場合のみ変更可	
		指定可能值	使用するエミュレータに依存	



- 注 使用するデバッグ・ツールにおいて E1(JTAG), または E1(Serial) を選択した場合, E1 エミュ レータのシリアル No. が列挙され, E20(Serial), または E20(JTAG) を選択した場合, E20 エミュ レータのシリアル No. が列挙されます。 空欄時に接続した場合, 検索して最初に見つかったエミュレータのシリアル番号を自動で選択し, 接続します。このとき, 自動で選択されたエミュレータのシリアル番号はプロジェクト情報には 保存されません。
- (5) [ターゲット・ボードとの接続]【E1】【E20】【EZ Emulator】

ターゲット・ボードとの接続状態に関する詳細情報の表示、および設定の変更を行います。

MCU 動作電圧 [V] 【EZ Emulator】	ユーザ基板に実際に供給しているマイクロコントローラの動作電圧値を指定しま す。			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力 ただし、デバッグ・ツールと切断中の場合のみ変更可		
	指定可能 値	0.0001 ~ 9.9999	(単作	立:V)
エミュレータから電源 供給をする(最大 200mA)【E1】【E20】	E1 からター E20 は電源(] を表示しま	ーゲット・システムに電源を供給するか否かを指定します。 原供給機能をサポートしていません。そのため,プロパティ値は[いいえ ます。		
	デフォル ト	いいえ		
	変更方法	-【E1】の場合 ドロップダウン・リストによる選択 ただし,デバッグ・ツールと切断中の場合のみ変更可		
		-【E20】の場合 変更不可		
	指定可能 値	はい	電源	を供給します。
		いいえ	電源	を供給しません。
電源供給インタフェー ス	エミュレータからターゲット・ボードに電源を供給するインタフェースを選択し ます。			
(E2)	デフォル ト	USER I/F		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	USER I/F		ユーザ・インタフェースを使用します。
		E2 expansion I/F		E2 拡張インタフェースを使用します。

供給電圧 【E1】	E1 からター なお, このつ 【E20】] プロ	ゲット・ボードに供給する電圧を指定します。 プロパティは [エミュレータから電源供給をする(最大 200mA)【E1】 コパティにおいて [はい] を指定した場合のみ表示されます。
	デフォル ト	3.3V
	変更方法	-【E1】【E2 Lite】 ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可 -【E2】 ドロップダウン・リストによる選択、またはキーボードからの直 接入力
	指定可能 値	-【E1】 ドロップダウン・リストによる次のいずれか ^注 3.3V, 5.0V
		- 【E2 Lite】 3.3V
		-【E2】 テキスト入力による次の範囲 1.8 ~ 5.0(単位:V)
通信方式	エミュレータ 式を表示しる	タとターゲット・システム上の CPU と通信を行う際に接続する通信方 ます。
	デフォル ト	- 【E1(JTAG)】 【E20(JTAG)】 JTAG
		- 【E1(Serial)】【E20(Serial)】【EZ Emulator】 <i>FINE</i>
		- 【E2 Lite】 【E2】 JTAG
	変更方法	-【E1】【E20】【EZ Emulator】 変更不可
		-【E2 Lite】【E2】 変更不可,またはドロップダウン・リストによる選択 変更可能かどうかは選択しているマイクロコントローラの種類と 通信方式によって異なります。マイクロコントローラの種類と通 信方式の関係については、「2.2.1 ホスト・マシンとの接続を確認 する」を参照してください。
	指定可能 値	-【E2 Lite】【E2】 <i>JTAG, FINE</i>



JTAG クロック [MHz] 【E1(JTAG)】 【E20(JTAG)】	エミュレータ す。 なお,このこ のみ表示され	タとターゲット・システム上の CPU 間の JTAG 通信速度を指定しま プロパティは [通信方式] プロパティにおいて [JTAG] を指定した場合 ぃます。
	デフォル ト	- 【E1(JTAG)】 【E20(JTAG)】 16.5
		- 【E2 Lite】 6.00
		- [E2] 16.5
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可
	指定可能	- ドロップダウン・リストによる次のいずれか
	値	- 【E1(JTAG)】 【E20(JTAG)】 16.5, 12.38, 6.188, 3.094, 1.547
		- 【E2 Lite】 6.00, 3.00, 1.50
		- 【E2】 16.5, 12.5, 6.25, 3.125, 1.562
FINE ボーレート [bps] 【E1(Serial)】 【E20(Serial)】【EZ	エミュレータ なお,このこ み表示される	タとターゲット・システム上の CPU 間の FINE 通信速度を指定します。 プロパティは [通信方式] プロパティにおいて [FINE] を指定した場合の ます。
Emulator	デフォル ト	- 【E1(Serial)】 【E20(Serial)】 【E2】 2000000
		- 【EZ Emulator】 1000000
		- 【E2 Lite】 1500000
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可
	指定可能	- ドロップダウン・リストによる次のいずれか
		- 【E1(Serial)】 【E20(Serial)】 【E2】 2000000, 750000, 500000, 250000
		- 【EZ Emulator】 1000000, 500000, 250000
		- 【E2 Lite】 1500000, 750000, 500000, 250000

注 指定可能な電圧値は、選択するマイクロコントローラによって異なります。

(6) [E2 拡張インタフェース]【E2】

E2 拡張インタフェースに関する詳細情報の表示,および設定の変更を行います。

E2 拡張インタフェース E を使用する う ず	E2 拡張インタフェースを使用するか否かを選択します。			
	デフォルト	使用しない		
	変更方法	ドロップダウン・リストによる選択		
	指定可能値	使用しない	E2 拡張インタフェースを使用しません。	
		ターゲット電源で使 用する	ターゲットから供給される電源で E2 拡張イ ンタフェースを使用します。	
		エミュレータからの 電源供給で使用する	エミュレータからの電源供給で E2 拡張イン タフェースを使用します。	

(7)

[フラッシュ]【E1】【E20】【EZ Emulator】 フラッシュ書き換えに関する詳細情報の表示,および設定の変更を行います。

ID コードの入力モード	ID ⊐−ドの.	の入力モードを指定します。				
	デフォルト	ID コードを16 進 32 桁で指定				
	変更方法	ドロップダウン・リスト ただし, デバッグ・ツー.	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能値	ID コードを 16 進 32 桁 で指定	ID コードを 16 進数の 32 桁で入力します。			
		ID コードを 16 文字以 内で指定 (ASCII)	ID コードを ASCII 16 文字以内で入力しま す。			
ID コード	ID コードプ	コテクトを解除するための	ID コードを指定します。			
	デフォルト	- [ID コードを 16 進 32 ホ <i>FFFFFFFFFFFFFFFFFF</i>	行で指定]の場合 FFFFFFFFFFFFFFFF			
		- [ID コードを 16 文字以内で指定 (ASCII)] の場合 <i>空欄</i>				
	変更方法	キーボードからの直接入力 ただし、デバッグ・ツールと切断中の場合のみ変更可				
	指定可能値	 - [ID コードを 16 進 32 株 32 桁の 16 進 ID コードの入力は、32 例:下記のように確保 0011223344556677885 const unsigned lo 0x00112233, 0x44556677, 0x8899aabb, 0xccddeeff }; フラッシュプログラマ 合があります。ご使用 仕様を確認してください 	行で指定]の場合 2 ビットのデータ単位で並べて入力します。 した場合, 29aabbccddeeffと入力します。 ongOSISreg[4] = { では ID コードの並び順が CS+ と異なる場 のフラッシュプログラマの ID コード入力の い。			
		- [ID コードを 16 文字以 ASCII コードの 16 文字	内で指定 (ASCII)] の場合 聲(16 文字に満たない場合は 0 が付加)			

ワーク RAM 開始アドレ ス	デバッガが使用するワーク RAM の配置アドレスを指定します。 アドレス値は4バイト単位の値を指定してください。(入力値が4バイト単位の値 でなかった場合は,自動的に補正されます。) 指定したワーク RAM の配置アドレスから指定バイト分は,デバッガのファーム ウェアが使用します。 ^注		
	デフォルト	選択しているマイクロコントローラに依存	
	変更方法	キーボードからの直接入力 ただし,デバッグ・ツールと切断中の場合のみ変更可	
	指定可能値	選択しているマイクロコントローラの内蔵 RAM 領域に合わせたアド レス値	
ワーク RAM サイズ [バ	デバッガが使用するワーク RAM のサイズを表示します。		
イト]	デフォルト	選択しているマイクロコントローラに依存	
	変更方法	変更不可	

注 メモリ内容の退避・復帰が行われるためユーザ・プログラムでもこの領域を使用できますが、 ワーク RAM に配置する領域は、以下の用途では使用できません。

- DMA または DTC 機能の転送元・転送先
- DTC ベクタテーブル,および転送情報配置アドレス

- DMAC, DTC 起動要因の割り込みベクタ

(8) [CPU の動作モード] 【E1】【E20】【EZ Emulator】 マイクロコントローラの動作モードに関する詳細情報の表示,および設定の変更を行います。

モード端子設定	モード端子で設定している動作モードを指定します。			
	デフォルト	ォルト <i>シングルチップモード</i>		
	変更方法	変更方法 ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可 指定可能値 - ドロップダウン・リストによる次のいずれか ^{注1}		
	指定可能值			
		-【E1】【E20】 シングルチップモード,ユーザブートモード		
		-【EZ Emulator】 シングルチップモード		
USB ブートプログラム の消去を許可する 【E1】【E20】	USB ブートプログラムが格納されたマイクロコントローラを使用してユーザブー トモードで起動する際に、ユーザブート領域上の USB ブートプログラムを消去す るか否かを指定します。 このプロパティは、[モード端子設定]プロパティにおいて[ユーザブートモード] を指定した場合のみ表示します。ただし、ユーザブートモードで起動する際に、 ユーザブート領域上の USB ブートプログラムを消去する必要のないマイクロコン トローラの場合は表示されません。			
	デフォルト	いいえ		
	変更方法 ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可		よる選択 と切断中の場合のみ変更可	
	指定可能值	<i>はい</i>	ユーザブート領域上の USB ブートプロ グラムを消去します。	
		いいえ	ユーザブート領域上の USB ブートプロ グラムを消去しません。	

r					
レジスタ設定	レジスタで設定する動作モードを指定します。				
	デフォルト	シングルチッ	プモード		
	変更方法	ドロップダウ: ただし, デバ・	ン・リストによる選択 ッグ・ツールと切断中の場合のみ変更可		
	指定可能值	- ドロップダ	ウン・リストによる次のいずれか ^{注 1}		
		-【E1】【E20】 <i>シングルチップモード,内蔵</i> ROM 有効拡張モード,内蔵 R(効拡張モード			
		-【EZ Emulator】 シングルチップモード			
エンディアン	プロジェク丨	~のエンディア ン	νを表示します。 ^{注 2}		
	デフォルト	Little-endian 5	Little-endian データ		
	変更方法	- ビルド&デバッグ・ツールの場合 変更不可			
		- デバッグ・ツールのみの場合 ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可			
	指定可能値	- ドロップダ Little-endian	ウン・リストによる次のいずれか ∩ データ, Big-endian データ		
起動バンクを変更する	デバッグ・ツ	リールと接続する	る際に起動バンクを変更するか否かを指定します。		
[RX651-2M, RX65N- 2M, RX66N, RX671,	デフォルト	いいえ			
RX72M, RX72N グルー プ】	変更方法	ドロップダウン・リストによる選択			
	指定可能值	はい	起動バンクを変更します。		
		いいえ 起動バンクを変更しません。			
起動バンク	デバッグ・ツールと接続する際に変更する起動バンクを指定します。				
[RX651-2M, RX65N- 2M, RX66N, RX671,	デフォルト	バンクロ			
RX72M, RX72N グルー プ】	変更方法	ドロップダウン・リストによる選択			
-	指定可能値	- ドロップダウン・リストによる次のいずれか <i>バンク0, バンク1</i>			

注 1. 指定可能な動作モードは、選択するマイクロコントローラによって異なります。

注 2. MDE 端子があるマイクロコントローラを選択した場合,プロジェクトのエンディアンとマイクロ コントローラの MDE 端子の状態を合わせてください。エンディアンが異なる場合,正常に接続 できません。

(9) [外部フラッシュ]【E1】【E20】外部フラッシュに関する詳細情報の表示、および設定の変更を行います。



外部フラッシュ定義 ファイル	外部フラッジ メインプロバ ます。 サブプロパラ に展開表示さ イル以外のサ	ンュ定義ファイルを指定し ペティには、登録可能なタ ティには、ファイル名、フ されます。なお、外部フラ ナブプロパティは表示され	します。 ^注 ▶部フラッシュ定義ファイルの数が表示され ▶ドレス範囲およびダウンロード条件が下段 ラッシュ定義ファイルが未登録の場合,ファ ℩ません。	
	デフォル ト	- メイン項目 <i>4</i> - サブ項目 <i>空欄</i>		
	変更方法	 メイン項目 変更不可 サブ項目 <ファイル> 外部フラッシュメモリダイアログ【E1】【E20】による選択 このプロパティを選択すると欄内右端に表示される[]ボタンを クリックすることで外部フラッシュメモリダイアログがオープン します。 <開始アドレス> 変更不可 <終了アドレス> 変更不可 <ダウンロード前に外部フラッシュ ROM を消去する> ドロップダウン・リストによる選択 		
	指定可能 値	はい	ダウンロード前に,外部フラッシュ ROM を消去します。	
		いいえ	ダウンロード前に,外部フラッシュ ROM を消去しません。	
	表示内容	 ・メイン項目 外部フラッシュ定義 ・サブ項目 外部フラッシュ定義 ・ンデックス([1]~[4 およびダウンロード 選択状況を表示します 	ファイル数 ファイルの " + " マークをクリックすると,イ])毎に定義ファイル名,開始/終了アドレス 前に外部フラッシュ ROM を消去するか否かの す。	

注 マイクロコントローラの動作モードがシングルチップモードの場合は,設定した内容は反映され ません。

(10) [周辺機能シミュレーション] 【シミュレータ】 周辺機能シミュレーションに関する詳細情報の表示、および設定の変更を行います。



周辺機能シミュレー ションモジュール	使用可能な属 定します。	周辺機能シミュレーショ ン	νモジュールを表示し,使用するか否かを指	
	デフォル ト	- メイン項目 <i><使用可能な周辺機能シミュレーションモジュール数></i>		
		- サブ項目 <i><周辺機能シミュレーションモジュール名></i> 使用しない		
	変更方法	- メイン項目 変更不可		
		- サブ項目 周辺機能シミュレーションモジュール名は変更不可 周辺機能シミュレーションモジュールを使用するか否かは、ド ロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中の場合のみ変更可		
	指定可能 値	使用する	周辺機能シミュレーションモジュールを使 用します。	
		使用しない	周辺機能シミュレーションモジュールを使 用しません。	
	表示内容	- メイン項目 使用可能な周辺機能日 - サブ項目	Eジュール数	
		周辺機能シミュレーン	ションモジュール名 ションモジュールの使用状況	
周辺クロックレート	周辺クロックと内部クロックの比(周辺1クロックが内部クロックいくつに相当 するか)を指定します。			
	デフォル ト	デフォル 1 ト 1		
	変更方法	ドロップダウン・リス	トによる選択	
	指定可能 値	- ドロップダウン・リス 1, 2, 3, 4, 6, 8,	ストによる次のいずれか 12, 16, 24, 32, 64	



[デバッグ・ツール設定] タブ

[デバッグ・ツール設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。

- (1) [メモリ]
- (2) [実行中のメモリ・アクセス]
- (3) [レジスタ]
- (4) [E2 拡張インタフェース]【E2】
- (5) [ブレーク] [E1] [E20] [EZ Emulator]
- (6) [システム] 【E1】 【E20】 【EZ Emulator】
- (7) [トレース]
- (8) [タイマ] 【E1】 【E20】 【EZ Emulator】
- (9) [カバレッジ]【シミュレータ】
- (10) [ストリーム入出力]【シミュレータ】
- (11) [実行モード]【シミュレータ】
- (12) [命令デコードキャッシュ]【シミュレータ】
- (13) [カバレッジ]【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】

[各カテゴリの説明]

- (1) [メモリ]
 - メモリに関する詳細情報の表示、および設定の変更を行います。

なお、表示されるメモリ種別についての詳細は、メモリ・マッピング ダイアログの項を参照してください。

メモリ・マッピング	現在のメモリ	モリ・マッピングの状況をメモリ領域の種別 ^{注1} ごとに表示します。 		
	デフォルト	[マイクロ	コントローラ固有のメモリ・マッピング領域種別の合計]	
	変更方法	メモリ・マ メモリ・マ 端に表示さ (このパネ)	マッピング ダイアログによる指定 マッピング ダイアログは、このプロパティを選択すると右 たれる […] ボタンをクリックすることでオープンします ル上でマッピング値を変更することはできません)。	
	表示内容 メモリ・ なお,各 表示しま		?ッピングの状況をメモリ領域の種別ごとに表示します。 < モリ種別の "+" マークをクリックすると,次の詳細情報を - 。	
		- メモリ種別		
		- 開始アト	ドレス	
		- 終了アドレス		
		- アクセス幅 [ビット]【E1】【E20】【EZ Emulator】 ^{注 2}		
		- エンディアン ^{注2注3}		
メモリ書き込み時にべ	メモリ値の衫	刃期化を行う	際に、ベリファイを行うか否かを指定します。	
リファイを行う 【E1】【E20】	デフォルト	はい		
【EZ Emulator】	変更方法	ドロップダウン・リストによる選択		
	指定可能值	はい	ベリファイを行います。	
		いいえ	ベリファイを行いません。	

注 1. デバイス・ファイルに登録されているメモリ・マッピング領域の種別です。

- 注 2. アクセス幅およびエンディアンは、メモリ種別が外部領域においてデバッグ・ツールが切断中の み変更可能です。また、指定可能値は選択したマイクロコントローラによって異なります。
- 注 3. エンディアン情報は、メモリ種別によって表示内容、および指定可能値が異なります。

- 外部領域【E1】【E20】【EZ Emulator】 ドロップダウン・リストより次のいずれかを指定します。 MCU のエンディアンと同じ, MCU のエンディアンと異なる

- I/O レジスタ領域 次のいずれかを表示します。 *Little-endian データ, Big-endian データ*
- エミュレーション ROM 領域,エミュレーション RAM 領域【シミュレータ】 次のいずれかを指定します。 *Little-endian データ, Big-endian データ*
- それ以外の領域 エンディアン情報は表示されません。
- (2) [実行中のメモリ・アクセス] プログラム実行中のメモリ・アクセス(リアルタイム表示更新機能(「2.11.1.4 プログラム実行中にメモリの内 容を表示/変更する」参照))に関する詳細情報の表示,および設定の変更を行います。

実行を一瞬停止してア クセスする 【E1】【E20】【EZ Emulator】 ト	プログラム実行中にはアクセスできないメモリ領域に対して,アクセスを許可する か否かを指定します。			
	デフォル ト	いいえ		
	変更方法	ドロップタ	ドロップダウン・リストによる選択	
	指定可能	はい	実行を一旦停止し,読み込み/書き込みを行います。	
	值	いいえ	実行中にアクセスは行いません。	
実行中に表示更新を行 う	プログラム事 かを指定しま	≷行中に,ウ ます。	フォッチ パネル/メモリ パネルの表示内容を更新するか否	
	デフォル ト	はい		
	変更方法	ドロップタ	マウン・リストによる選択	
	指定可能	はい	実行中に表示を更新します。	
	但	いいえ	実行中に表示を更新しません。	
表示更新間隔 [ms]	プログラム実行中に,ウォッチパネル/メモリパネルの表示内容を更新する間隔 を 100 ms 単位で指定します。 なお,このプロパティは,[実行中に表示更新を行う]プロパティにおいて[はい] を指定した場合のみ表示されます。			
	デフォル ト	500		
	変更方法	キーボードからの直接入力		
	指定可能 値	100 ~ 65500 の整数(単位:100 ms 未満の端数切り上げ)		
リアルタイム表示更新 を自動設定する 【E20(JTAG)【RX600, RX700 シリーズ】】	RRM 領域を自動的に設定するか否かを指定します。 なお,このプロパティは、[トレース機能の用途]プロパティにおいて[リア イム RAM モニタ]を指定し、且つ[実行中に表示更新を行う]プロパティ て[はい]を指定した場合のみ表示されます。		定するか否かを指定します。 k, [トレース機能の用途]プロパティにおいて[リアルタ 定し, 且つ [実行中に表示更新を行う] プロパティにおい 湯合のみ表示されます。	
	デフォル ト	はい		
	変更方法	ドロップタ	マウン・リストによる選択	
	指定可能 値	はい	自動的にリアルタイム表示更新の設定を行います。	
		いいえ	リアルタイム表示更新設定はしません。	

(3) [レジスタ]

プログラム実行中におけるステータス・バーへの PC 値表示、および表示更新間隔の変更を行います。

実行中に PC 表示を行 う デフォ ト	プログラム事	尾行中に PC	値をステータス・バーへ表示 ^注 するか否かを指定します。	
	注意	消費電 す。	流測定中は, このプロパティは [いいえ] に固定となりま	
	デフォル ト	いいえ		
	変更方法	ドロップタ	「ウン・リストによる選択	
	指定可能	はい	実行中にステータス・バーへ PC 値を表示します。	
		いいえ	実行中にステータス・バーへ PC 値を表示しません。 ステータス・バーには、「実行中」と表示されます。	
PC 表示更新間隔 [ms]	プログラム実行中にステータス・バーへ表示される PC 値の更新間隔を 1 位で指定します。 なお、このプロパティは[実行中に PC 表示を行う] プロパティにおいて を指定した場合のみ表示されます。		ータス・バーへ表示される PC 値の更新間隔を 100ms 単 :[実行中に PC 表示を行う]プロパティにおいて,[はい] されます。	
	デフォル ト	500		
	変更方法	キーボードからの直接入力		
	指定可能 値	100 ~ 65500 の整数(単位:100 ms 未満の端数切り上げ)		

注 【E1/E20/EZ Emulator【RX100 シリーズ】】

プログラム実行中の PC 表示機能をサポートしていないため、PC 値は表示されません。

(4) [E2 拡張インタフェース]【E2】

· E2 拡張インタフェースに関する詳細情報の表示,および設定の変更を行います。

外部トリガ入力	外部トリガ入力に関する設定を行います。 チャネル毎に動作を選択してください。				
チャネル番号	チャネル番号	- - ヤネル番号を表示します。			
(サフプロパティ)	デフォルト	0または1			
	変更方法	変更不可			
使用する	このチャネル番号の外部トリガ入力を使用するか否かを指定します。				
(サフフロバティ)	デフォルト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能值	定可能値 はい	当該チャネルの外部トリガ入力を使用します。		
	L'	いいえ	当該チャネルの外部トリガ入力を使用しません。		

入力信号	入力信号を指	入力信号を指定します。					
(サフフロハティ)	デフォルト	立ち上がり	エッジ				
	変更方法	ドロップダ	ウン・リ ン	ストによる選択			
	指定可能値	立ち上がり	エッジ	当該チャネルの外部トリガ入力で立ち上がり エッジを検出します。			
		立ち下がり	エッジ	当該チャネルの外部トリガ入力で立ち下がり エッジを検出します。			
		両エッジ		当該チャネルの外部トリガ入力で両エッジを 検出します。			
		High		当該チャネルの外部トリガ入力で High を検出 します。			
		Low		当該チャネルの外部トリガ入力で Low を検出 します。			
外部トリガ入力時の動	外部トリガス	、カ時の動作	.カ時の動作を表示します。				
1⊧ (サブプロパティ)	デフォルト	プログラムを停止する					
	変更方法	変更不可					
外部トリガ出力	外部トリガ出 チャネル毎に	∃カに関する設定を行います。 Ξ動作を選択してください。					
チャネル番号	チャネル番号を表示します。						
(サフブロパティ)	デフォルト	0または1					
	変更方法	変更不可					
使用する	このチャネル	ν番号の外部	トリガ出力	」を使用するか否かを指定します。			
(サノノロハナイ)	デフォルト	いいえ					
	変更方法	ドロップダウン・リストによる選択					
	指定可能值	はい	当該チャ	ネルの外部トリガ出力を使用します。			
		いいえ	当該チャ	ネルの外部トリガ出力を使用しません。			
出力タイミング	出力タイミン	•グを表示し	ます。				
	デフォルト	 プログラム停止					
	変更方法	変更不可					
出力信号	出力信号を表	衰示します。					
	デフォルト	High パルス	ス				
	変更方法	方法 変更不可					
パルス幅 [us] (サブプロパティ)	パルス幅を指	定します。					
	デフォルト	1					
	変更方法	キーボード	からの直接	妾入力			
	指定可能值	1 ~ 65535	の整数				

(5) [ブレーク] 【E1】 【E20】 【EZ Emulator】
 ブレーク機能に関する詳細情報の表示,および設定の変更を行います。

優先的に使用するブ レークポイントの種類	エディタ パネル/逆アセンブル パネルにおいて、ソース行、または実行アドレス に対してマウスのワンクリック操作でブレークポイントを設定する際に、優先的に 使用するブレークポイントの種別を指定します。					
	デフォル ト	ハードウエア・ブレーク				
	変更方法	ドロップダウン・リストによる選択				
	指定可能 値	ソフトウエア・ブレー ク	ソフトウエア・ブレークポイントを優先的 に設定します。			
		ハードウエア・ブレー ク	ハードウエア・ブレークポイントを優先的 に設定します。			

(6) [システム]【E1】【E20】【EZ Emulator】 エミュレーションシステムに関する詳細情報の表示,および設定の変更を行います。

内蔵プログラム ROM を書き換えるプログラ	内蔵プログラム ROM の書き換えを伴うプログラム(ROM P/E モードを使用した プログラムなど)をデバッグするか否かを指定します。 ^{注 1, 2}					
ムをナハックする	デフォル ト	いいえ	いいえ			
	変更方法	ドロップダウン・リストによる選択 ただし、デバッグ・ツールと切断中のみ変更可				
	指定可能 値	はい	内蔵プログラム ROM の書き換えを伴うプログラムをデ バッグします。			
		いいえ	内蔵プログラム ROM の書き換えを伴うプログラムをデ バッグしません。			
内蔵データフラッシュ を書き換えるプログラ	内蔵データ: ドを使用した					
ムをナハックする	デフォル ト	いいえ				
	変更方法	ドロップダウン・リストによる選択 ただし,デバッグ・ツールと切断中のみ変更可				
	指定可能 値	はい	内蔵データフラッシュの書き換えを伴うプログラムをデ バッグします。			
		いいえ	内蔵データフラッシュの書き換えを伴うプログラムをデ バッグしません。			
実行開始前に指定ルー	プログラム実行直前に指定ルーチンを実行するか否かを指定します。					
チンを実行する	デフォル ト	いいえ				
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可				
	指定可能	はい	プログラム実行直前に指定ルーチンを実行します。			
	100	いいえ	プログラム実行直前に指定ルーチンを実行しません。			

実行開始前に実行する ルーチン	プログラム実行直前に実行するアドレスを指定します。 このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティにおし はい]を指定した場合のみ表示します。						
	デフォル ト	空欄					
	変更方法	キーボート ただし, フ	^ゞ からの直接入力 [゜] ログラム実行停止中のみ変更可				
	指定可能 値	選択するマイクロコントローラに依存					
ブレーク後に指定ルー	プログラム	ブレーク直後	に指定ルーチンを実行するか否かを指定します				
チンを実行する	デフォル ト	いいえ	いいえ				
	変更方法	ドロップタ ただし, フ	『ウン・リストによる選択 ログラム実行停止中のみ変更可				
	指定可能	はい	プログラムブレーク直後に指定ルーチンを実行します。				
	1旦	いいえ	プログラムブレーク直後に指定ルーチンを実行しません。				
ブレーク後に実行する ルーチン	プログラムブレーク直後に実行するアドレスを指定します。 このプロパティは, [ブレーク後に指定ルーチンを実行する]プロパティにお はい]を指定した場合のみ表示します。						
	デフォル ト	空欄					
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可					
	指定可能 値	選択するマ	マイクロコントローラに依存				
指定ルーチン実行機能 用ワーク RAM 開始アド レス	指定ルーチン実行機能用のワーク RAM 配置アドレスを指定します。 アドレス値は4バイト単位の値を指定してください。(入力値が4バイト単位の値 でなかった場合は、自動的に補正されます。) なお、このプロパティは、[実行開始前に指定ルーチンを実行する]プロパティま たは、[ブレーク後に指定ルーチンを実行する]において[はい]を指定した場合の み表示します。						
	デフォル ト	0x1000 ^{注3}					
	変更方法	キーボードからの直接入力 ただし,プログラム実行停止中のみ変更可					
	指定可能 値	選択してい レス値	いるマイクロコントローラの内蔵 RAM 領域に合わせたアド				
指定ルーチン実行機能 用ワーク RAM サイズ [バイト]	指定ルーチン実行機能用のワーク RAM のサイズを表示します。 このプロパティは,[実行開始前に指定ルーチンを実行する]プロパティまたは ブレーク後に指定ルーチンを実行する]において[はい]を指定した場合のみ表 します。						
	デフォル ト	オル 選択しているマイクロコントローラに依存					
	変更方法	変更不可					

注 1. 【E1】【E20】【EZ Emulator】

内蔵フラッシュ ROM(プログラム ROM 領域)をユーザ・プログラムで書き換えた後に、メモリ パネル等で書き換えた内容を参照する場合は、[内蔵プログラム ROM を書き換えるプログラムを デバッグする]プロパティを[はい]に設定してください。 [内蔵プログラム ROM を書き換えるプログラムをデバッグする]プロパティを[いいえ]に設定 している場合は,書き換えた内容を参照することができません。 同様にデータフラッシュ領域では,[内蔵データフラッシュを書き換えるプログラムをデバッグする]プロパティを[はい]に設定してください。

- 注 2. 【E1】【E20】【EZ Emulator】 ユーザ・プログラム実行中にユーザ・プログラムで書き換えた内蔵フラッシュ ROM (プログラ ム領域,データフラッシュ領域)の内容をメモリパネル等で参照しないでください。 ユーザ・プログラム実行中に,書き換えた領域を参照した場合,正しく値が読み出せません。
- 注 3. 選択しているマイクロコントローラによって, 値が異なる場合があります。
- 注 4. 【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループのオプション設定メモ リを書き換えるプログラムをデバッグする場合, [内蔵データフラッシュを書き換えるプログラム をデバッグする]プロパティを[はい]に設定してください。
- (7) [トレース] トレース機能に関する詳細情報の表示,および設定の変更を行います。

注意 【E20(JTAG)【RX600, RX700 シリーズ】】 トレース機能/リアルタイム RAM モニタ機能(RRM 機能)は一部排他使用の機能です。

トレース機能の用途 【E1】【E20】 【EZ Eulator】	トレース機能をリアルタイム RAM モニタ機能(RRM 機能)として使用するか否 かを指定します。					
	デフォル ト	トレース	トレース			
	変更方法	-【E1(Seria 変更不可	I)】【E1(JTA	G)】【E20(Serial)】【EZ Emulator】		
		-【E20(JTAG)】 ドロップダウン・リストによる選択 ただし,プログラム実行停止中のみ変更可				
	指定可能	トレース		トレース機能を優先的に使用します。		
	値	リアルタイム RAM モ ニタ		リアルタイム RAM モニタ機能(RRM 機能) を優先的に使用します。		
トレース機能を使用す	トレース機能					
ぁ 【シミュレータ】	デフォル ト	いいえ				
	変更方法	ーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー				
	指定可能 値	はい	トレース機能を使用します。			
		いいえトレース機能を使用しません。				
実行前にトレース・メ	実行前にトレース・メモリをクリアするか否かを指定します。					
モリをクリアする 【シミュレータ】	デフォル ト	はい				
	変更方法	ドロップダロ	シン・リスト	トによる選択		
	指定可能	はい	実行前にトレース・メモリをクリアします。			
	1旦	いいえ	実行前にト	レース・メモリをクリアしません。		



トレース・メモリを使	トレースの取得モードを指定します。					
い切った後の動作 	デフォル ト	トレース・メモリを上着	書きし実行を続ける			
	変更方法	ドロップダウン・リストによる選択 ただし,プログラム実行停止中のみ変更可				
	指定可能 値	トレース・メモリを 上書きし実行を続け る	トレース・メモリを使い切った後もトレー ス・データを上書きします。			
		トレースを停止する 【E1】【E20】【EZ Emulator】	トレース・メモリを使い切った時点でト レース・データの書き込みを停止します。			
		停止する	トレース・メモリを使い切った時点でプロ グラムの実行とトレース・データの書き込 みを停止します。			
トレース・データ種別	トレース取	得するデータ種別を指定し	します。			
[E1] [E20] [EZ Emulator]	デフォル ト	分岐				
	変更方法	ドロップダウン・リストによる選択 ただし,プログラム実行停止中のみ変更可				
	指定可能 値	 ドロップダウン・リストによる次のいずれか - RX66x, RX65N-2M, RX651-2M, RX671, RX72T, RX72M, RX72N グループ【E20】 分岐,分岐+データアクセス,データアクセス,分岐+データアクセス(アクセス・データなし),データアクセス(アクセス・データなし) - RX600, RX700 シリーズ 分岐,分岐+データアクセス,データアクセス - RX100, RX200 シリーズ 分岐,データアクセス 				
アクセス・データを取得したい領域の開始マ	アクセス・	データを取得しない領域の	の開始アドレスを指定します。			
待じない _限 域の開始ア ドレス 【E20】【RX65N-2M,	デフォル ト	内蔵 RAM 領域の開始アドレス				
RX651-2M, RX66T, RX66N, RX671, RX72T,	変更方法	キーボードによる直接ス	λ			
RX72M, RX72N グルー プ】	指定可能 値	0x0 ~ 0xFFFFFFFFの	16 進数			
アクセス・データを取	アクセス・	データを取得しない領域(の終了アドレスを指定します。			
得しない領域の終了ア ドレス 【E20】【RX65N-2M,	デフォル ト	内蔵 RAM 領域の終了ア	アドレス			
RX651-2M, RX66T, RX66N, RX671, RX72T,	変更方法	キーボードによる直接	λ <u>,</u>			
RX72M, RX72N グルー プ】	指定可能 値	0x0 ~ 0xFFFFFFF の 16 進数				



データアクセスのバス・ マスタ 【E1/E20【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M,	データアクセスを発生させたバス・マスタを指定します。 このプロパティは、[トレース・データ種別]プロパティにおいて、[分岐+データ アクセス]、または[データアクセス]を指定した場合のみ選択可能です。 このプロパティは、[トレース機能の用途]プロパティにおいて、[リアルタイム RAM モニタ]を指定した場合、[CPU]固定となります。					
RX72N グループ】】	デフォル ト	CPU	CPU			
	変更方法	ドロップダウ ただし, プロ	ウン・リス コグラム実行	トによる選択 テ停止中のみ変更可		
	指定可能 値	CPU		CPU からのデータアクセス結果を,トレー スパネルに出力します。		
		DMAC/DTC		DMAC/DTC からのデータアクセス結果を, トレースパネルに出力します。		
外部トレース出力	トレースの	外部出力方法。	を指定します	f 。		
[E20(JIAG) [RX600, RX700 シリーズ]]	デフォル ト	CPU 実行優	CPU 実行優先			
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可				
	指定可能	CPU 実行優	先	トレース出力より CPU 実行を優先します。		
	值	トレース出力	力優先	CPU 実行よりトレース出力を優先します。		
		外部出力しな	ない	トレース情報は出力されません。		
トレース・メモリ・サ	トレース・メモリのサイズを指定します。					
イス [M ハイト] 【E20(JTAG)【RX600, RX700 シリーズ】】	デフォル ト	1				
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可				
	指定可能 値	- ドロップタ 1, 2, 4,	ダウン・リス <i>8, 16,</i> 32	ストよる次のいずれか		
タイム・スタンプ出力 【E1】【E20】 【EZ Emulator】	 収集するトレース・データにタイム・スタンプ情報を付加するか否かを指定す。注3 このプロパティは、[トレース機能の用途]プロパティにおいて、[トレース 定した場合のみ選択可能です。 					
	デフォル ト	いいえ				
	変更方法	ドロップダウ ただし. プロ	ウン・リス コグラム実行	トによる選択 テ停止中のみ変更可		
	指定可能	はい	タイム・ス	スタンプ情報を付加します 。		
	但	いいえ	タイム・ス	スタンプ情報を付加しません。		

トレース・クロック・ カウント・ソース [MHz] 【E1】【E20】 【EZ Emurator】	カウント値 す。 ただし, [タ み表示され	ウント値からタイム・スタンプを算出するためのカウント・ソースを入力しま 。 だし, [タイム・スタンプ出力]プロパティにおいて, [はい]を指定した場合の 表示されます。			
	デフォル ト	空欄			
	変更方法	キーボード	こよる直接入力		
	指定可能 値	0.0001 ~ 999.999			
トレース・クロック・ カウント・ソースの分 周比 【E1/E20【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M,	トレースのタイム・スタンプ用カウント・ソースの分周比を指定します。 このプロパティは、[タイム・スタンプ出力]プロパティにおいて、[はい]を指定 した場合のみ選択可能です。 [トレース・クロック・カウント・ソース [MHz]]プロパティで指定した周波数を さらに指定した値で分周(周波数を1/n倍に変換)した値が実際のタイム・スタン プの単位(1カウントあたりの周波数)になります。				
RX72N クルーフ】】	デフォル ト	分周なし			
	変更方法	ウン・リストによる選択 コグラム実行停止中のみ変更可			
	指定可能 値	分周なし	タイム・スタンプ用カウント・ソースの周波数をその ままの値で使用する。		
		16 分周	タイム・スタンプ用カウント・ソースの周波数を 1/16 倍に変換して使用する。		
		256 分周	タイム・スタンプ用カウント・ソースの周波数を 1/256 倍に変換して使用する。		
		4096 分周	タイム・スタンプ用カウント・ソースの周波数を 1/ 4096 倍に変換して使用する。		
トレース・タイム・タ	トレースパ	ネルに表示す	るトレース時間の表示方法を指定します。		
クを積昇する 【シミュレータ】	デフォル ト	いいえ			
	変更方法	ドロップダウン・リストによる選択 ただし, プログラム実行停止中のみ変更可			
	指定可能	はい	トレースの時間表示を積算値で表示します。		
		いいえ	トレースの時間表示を差分値で表示します。		
トレース・メモリ・サ イズ[フレーム]	トレース・・ ます。	トレース・データを保持するメモリ・サイズをトレース・フレーム ^{注2} 数で指定し ます。			
[9510-9]	デフォル ト	64K			
	変更方法	ドロップダ	ウン・リストによる選択		
	指定可能 値	- ドロップ? 64K, 128	ダウン・リストにより次のいずれか K, 256K, 512K, 1M, 2M, 3M		

注 1. エディタ パネル/逆アセンブル パネルにおいて, コンテキスト・メニュー→ [トレース開始の設定] / [トレース終了の設定]を選択した場合, このプロパティは自動的に [はい] に変更されます。

注 2. トレース・フレームはトレース・データの一単位を表します。 フェッチ/ライト/リードなどで、それぞれ1つのトレース・フレームを使用します。 注 3.RX100 シリーズでは、このプロパティは[いいえ]固定となり、トレース・データにタイム・スタンプ情報 を付加することはできません。

(8) [タイマ] 【E1】 【E20】 【EZ Emulator】
 タイマ機能に関する詳細情報の表示,および設定の変更を行います。

カウンタを 64 ビットで 使用する 【RX600, RX700 シリー ズ】	64 ビットの測定カウンタを使用するか否かを指定します。 ただし、[はい]を指定した場合、計測区間は1区間となります。				
	デフォル ト	いいえ			
	変更方法	ドロップダウン・リストによる選択			
	指定可能 値	はい	64 ビット ×1 本の計測カウンタを使用します。		
		いいえ	32 ビット ×2 本の計測カウンタを使用します。		

注意 RX140 グループ以外の RX100 シリーズでは、タイマ機能は未サポートとなります。

(9) [カバレッジ] 【シミュレータ】

カバレッジ機能を利用	カバレッジ機能を使用するか否かを指定します。			
する	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択 ただし,プログラム実行停止中のみ変更可		
	指定可能	はい	カバレッジ機能を使用します。	
	恒	いいえ	カバレッジ機能を使用しません。	
カバレッジ結果を再利 用する	デバッグ・ツールと接続時/切断時に、カバレッジ測定結果のロード/セーブを 行うか否かを指定します。 このプロパティは、[カバレッジ機能を利用する] プロパティにおいて[はい]を 指定した場合のみ表示されます。			
	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能	はい	カバレッジ計測結果のロード/セーブを行います。	
	值	いいえ	カバレッジ計測結果のロード/セーブを行いません。	

(10) [ストリーム入出力] 【シミュレータ】

ストリーム入出力モー ドを選択	ストリーム入出力モードを指定します。				
	デフォル ト	シミュレータモード			
	変更方法	ドロップダウン・リストによる選択 ただし, デバッグ・ツールと切断中のみ変更可			
	指定可能 値	シミュレータモード	シミュレータ固有の入出力方式を使用しま す。 標準入出力およびファイル入出力が使用で きます。		
		エミュレータモード	エミュレータと同様の入出力方式を使用し ます。 標準入出力のみ使用できます。		



-

ストリーム入出力機能 を使用する	ストリームフ このプロパラ ミュレータモ	入出力機能を使用するか否かを指定します。 ティは. [ストリーム入出力モードを選択] プロパティにおいて [シ モード] を指定した場合にのみ表示されます。		
	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択 ただし, プログラム実行停止中のみ変更可		
	指定可能 値	はい	ストリーム入出力機能を使用します。	
		いいえ	ストリーム入出力機能を使用しません	
ストリーム入出力用ア ドレス	ユーザプログラムから 入出力の開始位置を指 このプロパティは. [フ ミュレータモード]を		準入出力またはファイル入出力を行うためのストリーム します。 トリーム入出力モードを選択]プロパティにおいて[シ 定した場合にのみ表示されます。	
	デフォル ト	0		
	変更方法	キーボードからの直接入力 ただし、プログラム実行停止中のみ変更可		
	指定可能 値	0x0 ~ 0xF	FFFFFF の 16 進数	

(11) [実行モード]【シミュレータ】

実行モードを選択	シミュレージ 行を停止する	ションエラー るか否かを指	、または例外が発生した場合に、ユーザプログラムの実 定します。	
	デフォル ト	停止		
	変更方法	ドロップダウン・リストによる選択 ただし, プログラム実行停止中のみ変更可		
	指定可能	停止	シミュレーションを停止します。	
	値	続行	シミュレーションを続行します。	
未定義命令例外処理発 生時に停止する	未定義命令(の このプロパラ 場合のみ表示	列外発生時に ティは、[実行 示されます。	ユーザプログラムの実行を停止するか否かを指定します。 テモードを選択]プロパティにおいて[停止]を指定した	
	デフォル ト	はい		
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可		
	指定可能	はい	シミュレーションを停止します。	
	値	いいえ	シミュレーションを停止しません。	



特権命令例外発生時に 停止する	特権命令例外 このプロパラ 場合のみ表示	▶発生時にユ ティは, [実≀ 示されます。	ーザプログラムの実行を停止するか否かを指定します。 テモードを選択]プロパティにおいて[停止]を指定した	
	デフォル ト	はい		
	変更方法	ドロップダ ただし, フ	、 ウン・リストによる選択 プログラム実行停止中のみ変更可	
	指定可能	はい	シミュレーションを停止します。	
	但	いいえ	シミュレーションを停止しません。	
アクセス例外発生時に 停止する	アクセス例タ このプロパラ モジュールが た場合のみす	▶発生時にユ ティは, [周〕 が存在し, [雪 長示されます	ーザプログラムの実行を停止するか否かを指定します。 辺機能シミュレーションモジュール]プロパティに MPU 実行モードを選択]プロパティにおいて[停止]を指定し 。	
	デフォル ト 変更方法	はい		
		ドロップダウン・リストによる選択 ただし, プログラム実行停止中のみ変更可		
	指定可能	はい	シミュレーションを停止します。	
	值	いいえ	シミュレーションを停止しません。	
浮動小数点例外発生時 に停止する 【FPU 搭載品】	単精度浮動/ します。 このプロパラ 場合のみ表え	い数点例外発 ティは, [実ネ 示されます。	生時にユーザプログラムの実行を停止するか否かを指定 _{テモ} ードを選択] プロパティにおいて[停止]を指定した	
	デフォル ト	Itu		
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可		
	指定可能	はい	シミュレーションを停止します。	
	但	いいえ	シミュレーションを停止しません。	
割り込み発生時に停止 する	割り込み発生 このプロパラ 場合のみ表示	±時にユーザプログラムの実行を停止するか否かを指定します。 ティは, [実行モードを選択] プロパティにおいて [停止] を指定した 示されます。		
	デフォル ト	はい		
	亦百七注	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可		
	发史기丛	ただし、フ	『ログラム実行停止中のみ変更可	
	指定可能	ただし, フ はい	[°] ログラム実行停止中のみ変更可 シミュレーションを停止します。	

RENESAS

INT 命令実行時に停止 する	INT 命令実行 このプロパラ 場合のみ表示	5時にユーザ ティは, [実れ 示されます。	プログラムの実行を停止するか否かを指定します。 _{テモー} ドを選択]プロパティにおいて[停止]を指定した	
	デフォル ト	はい		
	変更方法	ドロップダウン・リストによる選択 ただし,プログラム実行停止中のみ変更可		
	指定可能	はい	シミュレーションを停止します。	
	值	いいえ	シミュレーションを停止しません。	
BRK 命令実行時に停止 する	BRK 命令実 このプロパラ 場合のみ表示	行時にユーサ ティは, [実行 示されます。	^デ プログラムの実行を停止するか否かを指定します。 _{テモ} ードを選択]プロパティにおいて[停止]を指定した	
	デフォル ト 変更方法	Itu		
		ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可		
	指定可能	はい	シミュレーションを停止します。	
	値	いいえ		

(12) [命令デコードキャッシュ] 【シミュレータ】

命令デコード結果を	命令デコート	デコードキャッシュ機能を使用するか否かを指定します。		
キャッシュして シミュ レーション速度を向上 する	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択 ただし、プログラム実行停止中のみ変更可		
	指定可能 値	はい	デコードキャッシュ機能を使用します。	
		いいえ	デコードキャッシュ機能を使用しません。	

(13) [カバレッジ] 【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】

コード・カバレッジ機	コード・カノ	、レッジ機能を使用するか否かを指定します,	
能を利用する	デフォル ト	いいえ	
	変更方法	ドロップダ ただし, デ	`ウン・リストによる選択 `バッグ・ツールと切断中のみ変更可
	指定可能 値	はい	コード・カバレッジ機能を使用します。
		いいえ	コード・カバレッジ機能を使用しません。

カバレッジ計測方法	カバレッジラ このプロパラ [はい] を指	データの計測 ティは, [コ- 定した場合の	方法を指定し - ド・カバレ のみ表示され	、ます。 ッジ機能を利用する]プロパティにおいて ます。
	デフォル ト	CPU 実行	憂先	
	変更方法	ドロップダ ただし, フ	[゛] ウン・リス ゜ログラム実行	トによる選択 庁停止中のみ変更可
	指定可能 値	CPU 実行	憂先	CPU 実行を優先します。データ出力時のロ ストによりカバレッジデータが失われる可 能性があります。
		カバレッジ	計測優先	カバレッジ計測を優先します。カバレッジ データ出力のために CPU 実行が停止し、プ ログラム実行のリアルタイム性が損なわれ る可能性があります。
カバレッジ計測範囲	カバレッジ メインプロパ サブプロパラ 下段に展用 開始アドレス 位の値プロパラ [はい] を指	を計測する範 パティには, ティには,開 長示されます くは,4Mバ・ かった場合は ティは,[コ- 定した場合の	囲を指定しま 登録可能なス 。 イト単位の値 、 自動的に補 ード・カバレ のみ表示され	ミす。 コバレッジ計測範囲の数が表示されます。 カバレッジ計測領域サイズ [M バイト] が を指定してください(入力値が 4M バイト単 捕正されます)。 ッジ機能を利用する] プロパティにおいて ます。
	デフォル ト	- メイン項 4 - サブ項目 <開始ア <i>空欄</i> くカバレ 4	 ドレス> ッッジ計測領は	或サイズ [M バイト] >
	変更方法	- メ変 イフ - メ変 サブ開ーカ 東 - サ く 開 ーカ 更 た し て の 日 ア ー レ の た 、 変 た 、 で 、 の 、 で 、 で 、 の 、 で 、 で 、 の 、 の 、 の	目 「 」 - ドからの直打 ッジ計測領却 「 デバッグ・Y	度 入力 或サイズ[M バイト]> ソールと切断中のみ変更可
	指定可能 値	0x0 ~ 0xF	FFFFFF の	16 進数
カバレッジ結果を再利 用する	デバッグ・2 行うか否かを このプロパラ [はい] を指	ソールと接続 を指定します ティは、[コ- 定した場合の	ード・カバレ のみ表示され	ニ, カバレッジ測定結果のロード/セーブを ッジ機能を利用する] プロパティにおいて ます。
	デフォル ト	いいえ		
	変更方法	ドロップダ	· ウン・リス	
	指定可能	はい	カバレッジ	計測結果のロード/セーブを行います。
	1旦	いいえ	カバレッジ	

[ダウンロード・ファイル設定] タブ

[ダウンロード・ファイル設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。 なお、ダウンロード方法については、「2.5 ダウンロード/アップロード」を参照してください。

(1) [ダウンロード]

(2) [デバッグ情報]

[各カテゴリの説明]

(1) [ダウンロード]

ダウンロードに関する詳細情報の表示、および設定の変更を行います。

ダウンロードするファ イル	ダウンロードするファイルを指定します ^{注1} 。 サブプロパティとして、ダウンロードするファイル名、およびダウンロード条件 を下段に展開表示します。					
	デフォルト	[ダウンロ-	ードするファイルの数]			
	変更方法	ダウンロー ダウンロー と欄内右端 します(プ ことはでき	ダウンロード・ファイル ダイアログによる選択 ダウンロード・ファイル ダイアログは、このプロパティを選択する と欄内右端に表示される […] ボタンをクリックすることでオープン します(プロパティ パネル上でダウンロード・ファイルを指定する ことはできません)。			
ダウンロード後に CPU	ダウンロート	ド後に CPU をリセットするか否かを指定します。				
をリセットする	デフォルト	ltι				
	変更方法	ドロップダ	ウン・リストによる選択			
	指定可能值	はい	ダウンロード後に CPU をリセットします。			
		いいえ	ダウンロード後に CPU をリセットしません。			
ダウンロード前にフ ラッシュ ROM を消去	ダウンロード前にフラッシュ ROM(プログラム ROM)を消去するか否かを指定します。					
する 【E1】【E20】	デフォルト	はい				
[EZ Emulator]	変更方法	ドロップダ	ウン・リストによる選択			
	指定可能值	はい	ダウンロード前にフラッシュ ROM を消去します。			
		いいえ	ダウンロード前にフラッシュ ROM を消去しません。			
ダウンロード前にデー	ダウンロート	ド前にデータ	フラッシュ ROM を消去するか否かを指定します。			
タフラッシュ ROM を 消去する	デフォルト	いいえ				
【E1】【E20】 【EZ Emulator】	変更方法	ドロップダ	ウン・リストによる選択			
	指定可能值	はい	ダウンロード前にデータフラッシュ ROM を消去しま す。			
		いいえ	ダウンロード前にデータフラッシュ ROM を消去しません。			

イベント設定位置の自 動変更方法	再ダウンロ- レス)が命令	-ドすることにより, 現在 うの途中になる場合の再調	E設定されているイベントの設定位置(アド G定方法を指定します ^{注 2} 。
	デフォルト	イベントを保留にする	
	変更方法	ドロップダウン・リス	トによる選択
	指定可能値	命令の先頭に移動す る	命令の先頭アドレスに対象イベントを再設 定します。
		イベントを保留にす る	対象イベントを保留状態にします。

注 1. メイン・プロジェクト/サブプロジェクトでビルド対象に指定しているファイルは、ダウンロードの対象ファイルから削除することはできません(デフォルトで自動的にダウンロード・ファイルとして登録されます)。 なお、ダウンロード可能なファイル形式については、「表 2.2 ダウンロード可能なファイル形式」を参照してください。

^{(2) [}デバッグ情報] デバッグ情報に関する詳細情報の表示,および設定の変更を行います。

CPU リセット後に指定 シンボル位置まで実行	CPU リセット後に、プログラムを指定シンボル位置まで実行するか否かを指定します。					
ग ৩	デフォルト	はい				
	変更方法	ドロップダウン・リストによる選択				
	指定可能值	はい	プログラムを指定シンボル位置まで実行します。			
		いいえ	CPU リセット後にプログラムを実行しません。			
指定シンボル	CPU リセット後に、プログラムを実行して停止する位置を指定します。 なお、このプロパティは、[CPU リセット後に指定シンボル位置まで実行する]プ ロパティにおいて[はい]を選択した場合のみ表示されます。					
	デフォルト	_main				
	変更方法	キーボードからの直接入力				
	指定可能值	0~" <i>アドレス空間の終了アドレス</i> "のアドレス式				
デバッグするオーバー レイ・セクションを設	ダウンロードしたロード・モジュール内にオーバーレイ・セクションが存在する 場合に, デバッグ対象とするセクションを選択します。					
定する	デフォルト	 ロードモジュール内にオーバーレイ・セクションが存在しない合い いいえ ロードモジュール内にオーバーレイ・セクションが存在する場(たい) 				
	変更方法	変更不可				

注 2. デバッグ情報がないイベント設定位置のみが対象となります。デバッグ情報がある場合のイベント設定位置は、常にソース・テキスト行の先頭に移動します。

オーバーレイ・セク ション	オーバーレィ スグループに ション(優労 なお,このこ プロパティ項	(・セクションが存在するアドレスグループを表示します。各アドレ に定義されたオーバーレイ・セクションの中から、デバッグするセク モセクション)を選択します。 プロパティは、[デバッグするオーバーレイ・セクションを設定する] 頁目において[はい]と表示された場合のみ表示されます。
	デフォルト	なし
	変更方法	- [ファイル],[開始アドレス],[終了アドレス]項目 変更不可
		- [優先セクション]項目 ドロップダウン・リストによる選択
	表示内容	各アドレスグループ番号の "+" マークをクリックすると、次の詳細情 報が表示されます。 [優先セクション]項目にて選択されているセクション名がデバッグ 対象のセクションになります。
		- ファイル
		- 開始アドレス
		- 終了アドレス
		- 優先セクション
メモリ使用量の上限サ イズ [M バイト]	デバッグ情幸 す。 メモリ不足か あります。た する場合があ	&の読み込みで使用するメモリ量の上限サイズ [M バイト] を指定しま ^K 発生する場合,上限サイズを小さくすることで改善される可能性が こだし,上限サイズを小さくするとデバッグ・ツールの応答性が低下 5ります。
	デフォルト	500
	変更方法	キーボードからの直接入力
	指定可能值	100~1000の整数

- **注意 1.** オーバーレイ・セクションに関する設定内容は、プロジェクトファイルに保存しません。ロード・モジュールのダウンロード後は、デバッグ対象のセクションを再設定してください。
- **注意 2.** [優先セクション]項目の変更によって切り替わる情報は、デバッグ情報のみです。デバッガは 対象セクションのデータのコピーは行いません。



[フック処理設定] タブ

[フック処理設定] タブでは、次に示すカテゴリごとに詳細情報の表示、および設定の変更を行います。 なお、フック処理の設定方法については、「2.18 フック処理を設定する」を参照してください。

(1) [フック処理設定]

[各カテゴリの説明]

(1) [フック処理設定]

フック処理に関する詳細情報の表示,および設定の変更を行います。 なお,このタブ上のプロパティの設定は,各プロパティを選択すると右端に表示される[…]ボタンをクリック することでオープンするテキスト編集 ダイアログで行います(このパネル上で処理を指定することはできません)。

注意 1 処理につき 64 文字まで入力可能で,各プロパティごとに 128 個までの処理を指定することが できます (テキスト編集 ダイアログ上の [テキスト] エリア内の 1 行が 1 処理に相当)。

ダウンロード前	ロード・モジ	ジュール・ファイルをダウンロードする直前に行う処理を指定します。
	デフォルト	ダウンロード前[0]("[]"内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定可能值	次のいずれか
		- I/O <i>レジスタ名</i> + 半角スペース + <i>数値</i> 【処理】I/O <i>レジスタの内容を数値に自動的に書き換えます。</i>
		- CPU レジスタ名+半角スペース + 数値 【処理】CPU レジスタの内容を数値に自動的に書き換えます。
		- Source+半角スペース +Python スクリプト・パス 【処理】Python スクリプト・パスで指定したスクリプト・ファイ ルを実行します。
ダウンロード後	ロード・モシ	ジュール・ファイルをダウンロードした直後に行う処理を指定します。
ダウンロード後	ロード・モシ デフォルト	ジュール・ファイルをダウンロードした直後に行う処理を指定します。 ダウンロード後[0]("[]"内は現在の指定処理数を示す)
ダウンロード後	ロード・モジ デフォルト 変更方法	ジュール・ファイルをダウンロードした直後に行う処理を指定します。 ダウンロード後[0]("[]"内は現在の指定処理数を示す) テキスト編集 ダイアログによる指定
ダウンロード後	 ロード・モジ デフォルト 変更方法 指定可能値 	ジュール・ファイルをダウンロードした直後に行う処理を指定します。 ダウンロード後[0]("[]"内は現在の指定処理数を示す) テキスト編集 ダイアログによる指定 次のいずれか
ダウンロード後	 ロード・モジ デフォルト 変更方法 指定可能値 	ジュール・ファイルをダウンロードした直後に行う処理を指定します。 ダウンロード後[0]("[]"内は現在の指定処理数を示す) テキスト編集 ダイアログによる指定 次のいずれか - I/O レジスタ名+半角スペース+数値 【処理】I/O レジスタの内容を数値に自動的に書き換えます。
ダウンロード後	ロード・モジデフォルト変更方法指定可能値	ジュール・ファイルをダウンロードした直後に行う処理を指定します。 ダウンロード後[0]("[]"内は現在の指定処理数を示す) テキスト編集 ダイアログによる指定 次のいずれか - I/O レジスタ名+半角スペース+数値 【処理】I/O レジスタの内容を数値に自動的に書き換えます。 - CPU レジスタ名+半角スペース+数値 【処理】CPU レジスタの内容を数値に自動的に書き換えます。



ブレーク中の CPU リ セット後	ブレーク中の CPU リセット直後に行う処理を指定します。	
	デフォルト	<i>ブレーク中の CPU リセット後 [0]</i> ("[]" 内は現在の指定処理数を示 す)
	変更方法	テキスト編集 ダイアログによる指定
	指定可能值	次のいずれか
		- I/O <i>レジスタ名</i> + 半角スペース + <i>数値</i> 【処理】I/O <i>レジスタの内容を数値に自動的に書き換えます。</i>
		- CPU レジスタ名+半角スペース + <i>数値</i> 【処理】CPU レジスタの内容を数値に自動的に書き換えます。
		- Source+半角スペース +Python スクリプト・パス 【処理】Python スクリプト・パスで指定したスクリプト・ファイ ルを実行します。
実行開始前	プログラムの実行開始直前に行う処理を指定します。	
	デフォルト	<u>実行開始前[0]</u> ("[]"内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定可能值	次のいずれか
		- I/O <i>レジスタ名</i> + 半角スペース + <i>数値</i> 【処理】I/O <i>レジスタの内容を数値に自動的に書き換えます。</i>
		- CPU レジスタ名+半角スペース + <i>数値</i> 【処理】CPU レジスタの内容を数値に自動的に書き換えます。
		- Source+ 半角スペース +Python スクリプト・パス 【処理】Python スクリプト・パスで指定したスクリプト・ファイ ルを実行します。
ブレーク後	プログラムの実行がブレークした直後に行う処理を指定します。	
	デフォルト	ブレーク後[0]("[]"内は現在の指定処理数を示す)
	変更方法	テキスト編集 ダイアログによる指定
	指定可能值	次のいずれか
		- I/O <i>レジスタ名</i> + 半角スペース + <i>数値</i> 【処理】I/O <i>レジスタの内容を数値に自動的に書き換えます。</i>
		- CPU レジスタ名 + 半角スペース + 数値 【処理】CPU レジスタの内容を数値に自動的に書き換えます。
		- Source+半角スペース +Python スクリプト・パス 【処理】Python スクリプト・パスで指定したスクリプト・ファイ ルを実行します。



メモリ パネル

メモリの内容の表示,変更を行います(「2.11.1 メモリを表示/変更する」参照)。

- このパネルは、最大4個までオープンすることができます。各パネルは、タイトルバーの"メモリ1"、"メモリ2"、"メモリ3"、"メモリ4"の名称で識別されます。
- プログラムの実行後、メモリの値が変化すると表示を自動的に更新します(ステップ実行時には、ステップ実行ごとに表示を逐次更新)。
- また、リアルタイム表示更新機能を有効にすることにより、プログラム実行中であっても、値の表示をリアルタイムに更新することも可能です。
- なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。
- 図 A.5 メモリ パネル × メモリ1 [ツールバー] 😰 🛞 | 表記(N)+ | サイズ表記(Z)+ | エンコード(C)+ | 表示(I)+ (1) -移動 ÷ +2 ± 0 ± 2 +4+6÷ +9 $\pm a$ $\pm c$ te ASC . 08 62 D3 CE E0 2C 62 B9 FF 78 A4 D3 7F 32 EF .b???.b??x2??.? 00000000 03 00000010 11 80 FA 7F 23 44 F0 9F 6C DE 02 91 29 DC 07 84 .??. D??!?.?)?.? .1;...U?1?1?1?;? .17.7.717..17\$.y?.??u?n.????a 00000020 20 18 DD 3B 02 00 75 F9 C4 FA C4 8A FF AF 0E 90 CF D4 04 1A 00000030 08 08 EF C8 00 88 10 DB DF BF 24 20 F6 6E 02 F1 DC FF 20 04 79 DA 10 90 F0 75 BF 1000000040 61 00000050 82 80 F2 61 01 00 C8 94 71 9D D4 59 F7 7E 0E 85 ?a..??q??Y? .? 000000000 02 41 95 FD 94 0C AA FF 3C F7 01 05 5F BD .A???,??<?.._??. 3F 10 .17.&1710..07K. .17..1717.7c{. 00000070 18 08 BF E8 01 26 E7 AE F2 6F 00 00 6F 9F 4B 04 50 OF FF C0 05 00 B1 9C 63 7B 04 8B CF 08 80 20 .??9q?????.&K?.. .@w??4??~~.f{?.? \$~??..?o.?..???? 10 94 FA 39 71 A0 D0 D9 FA D3 00 26 4B F3 0E 15 000000a0 0A 40 77 9B CD 34 D1 8F 7E 7E 00 66 7B FE 0E B0 00000060 24 22 BE EE 01 18 93 6F 0F B0 08 08 FD E3 9C 80 (2) (3) (4)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (メモリ パネル専用部分)]
- [[編集] メニュー (メモリ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [メモリ] → [メモリ1~4] を選択

[各エリアの説明]

- (1) 表示位置指定エリア アドレス式を指定することにより、メモリ値の表示開始位置を指定することができます。 次の指定を順次行います。
 - (a) アドレス式の指定 表示したいメモリ値のアドレスとなるアドレス式をテキスト・ボックスに直接入力します。最大 1024 文字ま での入力式を指定することができ、その計算結果を表示開始位置アドレスとして扱います。 ただし、マイクロコントローラのアドレス空間よりも大きいアドレス式を指定することはできません。

RENESAS
- 備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置の シンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- 備考 2. 指定したアドレス式がシンボルを表現し、サイズが判明する場合では、そのシンボルの先頭ア ドレスから終了アドレスまでを選択状態で表示します。
- (b) アドレス式の自動/手動評価の指定

表示開始位置を変更するタイミングは、[停止時に移動]チェック・ボックスの指定、および [移動] ボタン により決定します。

[停止時に移動]	V	プログラム停止後, 自動的にアドレス式の評価を行い, その計算結果のアドレスに キャレットが移動します。
		プログラム停止後, アドレス式の評価を自動的に行いません。 この場合, [移動] ボタンをクリックすることにより, アドレス式の評価を行いま す。
[移動]		[停止時に移動] チェック・ボックスのチェックをしなかった場合, このボタンを クリックすることによりアドレス式の評価を行い, その計算結果のアドレスにキャ レットが移動します。

(2) アドレス・エリア

メモリのアドレスを表示します(16 進数表記固定)。 デフォルトで,0x0 番地より表示を開始します。

ただし、コンテキスト・メニューの[表示アドレス・オフセット値の設定 …]を選択することでオープンするア ドレス・オフセット設定 ダイアログにより、開始アドレスにオフセット値を設定することができます。 アドレス幅は、プロジェクトで指定しているマイクロコントローラのメモリ空間のアドレス幅となります。 このエリアを編集することはできません。

注意 設定されたオフセット値は、メモリ値エリアの表示桁数の指定に従って自動的に変更されます。

(3) メモリ値エリア

マイクロコントローラにマッピングされているプログラム用のメモリ値を表示/変更します。 メモリ値の表示進数/表示幅/表示桁の指定は、ツールバーのボタン、またはコンテキスト・メニューの[表記]/[サイズ表記]/[表示]の選択により行います(「2.11.1.2 値の表示形式を変更する」参照)。 メモリ値として表示されるマークや色の意味は次のとおりです(表示の際の文字色/背景色はオプションダイ アログにおける[全般-フォントと色]カテゴリの設定に依存)。

表示例(デフォルト)			説明	
00	文字色	青	ユーザにより, 値が変更されているメモリ値([Enter] キーによりターゲッ	
	背景色	標準色	ト・メモリに書さ込まれます)	
00	文字色	標準色	シンボルが定義されているアドレスのメモリ値(ウォッチ式の登録を行うこ とができます)	
(下線)	背景色	標準色		
00	文字色	茶色	プログラムの実行により, 値が変化したメモリ値 ^{注 1} - ツールバーの 💮 ボタンをクリックすると, 強調表示をリセットします。	
	背景色	クリーム		
00	文字色	ピンク	リアルタイム表示更新機能を行っているメモリ値	
	背景色	標準色		

表示例(デフォルト)				説明
00	文字色	標準色	リード/	リアルタイム表示更新機能を行っている場合、現在のメモ
	背景色	薄緑	フェッチ	リ値のアクセス状態
00	文字色	標準色	ライト	
	背景色	オレンジ		
00	文字色	標準色	リードとライ	
	背景色	薄青	F	
00	文字色	白	ロスト	
	背景色	ライトグ レー		
00	文字色	グレー	リード不可の領	
	背景色	標準色		
??	文字色	グレー	メモリがマッピ	ングされていない領域,書き換え不可能領域 (IO レジスタ
	背景色	標準色	│領域 /I/O 保護領	域/= など), またはメモリ値の取得に失敗した場合
**	文字色	標準色	2 プログラム実行中に、リアルタイム表示更新領域以外の領域を表示指定した 場合、またはメモリ値の取得に失敗した場合	
	背景色	標準色		

- 注 1. プログラム実行直前において、メモリ パネルで表示されていたアドレス範囲のメモリ値のみが対象となります。また、プログラムの実行前後での値の比較であるため、実行結果が同一値となった場合は強調表示を行いません。
- 注 2. 【E1】【E20】【EZ Emulator】 I/O 保護領域は,"00"と表示されます。
- **注意** このエリアの表示桁数は、コンテキスト・メニューの[サイズ表記]の指定に従って自動的に変更します。

このエリアは、次の機能を備えています。

(a) ポップアップ表示

メモリ値にマウス・カーソルを重ねることにより、マウス・カーソルが指しているアドレスに対して前方向に 存在する一番近いシンボルを基準にして、次の内容をポップアップ表示します。 ただし、シンボル情報が存在しない場合(下線が非表示の場合)はポップアップ表示は行いません。

variable	+ 0x14
シンボル名	オフセット値

シンボル名	シンボル名を表示します。
オフセット値	アドレスにシンボルが定義されていない場合は,前方向に存在する一番近いシンボルか らのオフセット値を表示します(16 進数表示固定)。

- (b) リアルタイム表示更新機能 リアルタイム表示更新機能を使用することにより、プログラムが停止している状態の時だけでなく、実行中の 状態であっても、メモリ値の表示/変更を行うことができます。 リアルタイム表示更新機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」 を参照してください。
- (c) メモリ値の変更 メモリ値の変更は、対象メモリ値にキャレットを移動したのち、直接キーボードより編集することで行います。

RENESAS

メモリ値を編集すると変更箇所の表示色が変化し、この状態で [Enter] キーを押下することにより変更した 値がターゲット・メモリに書き込まれます([Enter] キーの押下前に [Esc] キーを押下すると編集をキャン セルします)。

メモリ値の変更方法についての詳細は、「2.11.1.3 メモリの内容を変更する」を参照してください。

(d) メモリ値の検索/初期化

コンテキスト・メニューの [検索 …] を選択することにより,指定したアドレス範囲のメモリ内容を検索す るためのメモリ検索 ダイアログをオープンします(「2.11.1.5 メモリの内容を検索する」参照)。 また、コンテキスト・メニューの [初期化 …] を選択することにより,指定したアドレス範囲のメモリ内容 を一括して変更するためのメモリ初期化 ダイアログをオープンします(「2.11.1.6 メモリの内容を一括して変 更(初期化)する」参照)。

(e) コピー/貼り付け

メモリ値をマウスにより範囲選択することで、その箇所の内容を文字列としてクリップ・ボードにコピーする ことができ、その内容をキャレット位置に貼り付けることができます。

これらの操作は、コンテキスト・メニューの選択、または[編集]メニューの選択により行います。

ただし,貼り付け操作は,貼り付け対象の文字列とそのエリアの表示形式(表示進数/ビット幅)が一致する 場合のみ可能です(表示形式が一致しない場合は、メッセージを表示します)。

なお,このエリアで扱うことができる文字コードと文字列は次のとおりです(これ以外の文字列を貼り付けた 場合は、メッセージを表示します)。

文字コード	ASCII
文字列	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, A, B, C, D, E, F

(f) ウォッチ式の登録

シンボルが定義されているアドレスでは、メモリ値に下線が表示され、ウォッチ式として登録可能であること を示します。

このメモリ値を選択,またはメモリ値のいずれかにキャレットを置いた状態で,コンテキスト・メニューの [ウォッチ1に登録]を選択することにより,指定したアドレスのシンボル名がウォッチ式としてウォッチパ ネル(ウォッチ1)に登録されます。

注意 下線表示のないメモリ値をウォッチ式に登録することはできません。

(g) メモリ値の保存 [ファイル] メニュー→ [名前を付けてメモリ・データを保存...]を選択することにより、データ保存 ダイア ログをオープンし、このパネルの内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存すること ができます。

メモリ値の保存方法についての詳細は、「2.11.1.7 メモリの表示内容を保存する」を参照してください。

(h) 表示の拡大/縮小

メモリ パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより、メモリ パネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、メモリパネルの表示を拡大し、表示が大きく見やすくなります(最大 300%)。

- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、メモリパネルの表示を縮小し、表示が小 さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。



(4) 文字列エリア

メモリの値を文字コードに変換して表示します。 文字コードの指定は、ツールバーのボタンのクリック、またはコンテキスト・メニューの[エンコード]の選択 により行います(デフォルトではASCIIコードで表示します)。 また、このエリアでは、メモリ値を浮動小数点数値に変換して文字列として表示することができます。この場合 は、コンテキスト・メニューの[エンコード]より次の項目の選択を行ってください。

項目	表記		サイズ
Half-Precision Float	半精度浮動小数点数		16 ビット
	数值	<i><符号><仮数部</i> > e <i><符号</i> > <i><指数部</i> >	
	無限大	Inf, および-Inf	
	非数	NaN	
	例	+ 1.234e+1	
Float	単精度浮動小数点数		32 ビット
	数值	<i><符号><仮数部</i> > e <i><符号</i> > <i><指数部</i> >	
	無限大	Inf, および-Inf	
	非数	NaN	
	例	+1.234567e+123	
Double	倍精度浮動小数点数		64 ビット
	数值	<i><符号><仮数部</i> > e <i><符号</i> > <i><指数部</i> >	
	無限大	Inf, および-Inf	
	非数	NaN	
	例	+1.2345678901234e+123	
Float Complex	単精度浮動小数点数の複素数		64 ビット
	<単精度浮動小数点数><単精		
Double Complex	倍精度浮動小数点数の複素数		128 ビット
	<倍精度浮動小数点数><倍精度浮動少数点数>*1		
Float Imaginary	単精度浮動少数点の虚数	少数点の虚数	
	< <i>単精度浮動小数点数</i> > *1		
Double Imaginary	倍精度浮動小数点数の虚数		64 ビット
	<倍精度浮動少数点数> *1		

注意 指定されている文字コード,または浮動小数点数値の最小サイズが"表示バイト数×表示桁数バイト数"より大きい場合,このエリアには何も表示されません。

このエリアは、次の機能を備えています。

(a) 文字列の変更

現在,文字コードとして [ASCII] が指定されている場合のみ,文字列を変更することができます。 文字列の変更は,対象文字列にキャレットを移動したのち,直接キーボードより編集することで行います。 文字列を編集すると変更箇所の表示色が変化し,この状態で [Enter] キーの押下することにより変更した値 がターゲット・メモリに書き込まれます([Enter] キーの押下前に [Esc] キーを押下すると編集をキャンセ ルします)。

RENESAS



(b) 文字列の検索 コンテキスト・メニューの [検索...]を選択することにより、文字列を検索する為の メモリ検索 ダイアログ をオープンします(「2.11.1.5 メモリの内容を検索する」参照)。

(c) コピー/貼り付け 文字列をマウスにより範囲選択することで、その箇所の内容を文字列としてクリップ・ボードにコピーするこ とができ、その内容をキャレット位置に貼り付けることができます。 これらの操作は、コンテキスト・メニューの選択、または[編集]メニューの選択により行います。 ただし、貼り付け操作は、文字コードとして ASCII が指定されている場合のみ可能です(ASCII 以外が指定さ れている場合は、メッセージを表示します)。

[ツールバー]

8	デバッグ・ツールから最新の情報を取得し、表示を更新します。
1	プログラム実行により値が変化した箇所を示す強調表示をリセットします。 ただし, プログラム実行中は無効となります。
表記	メモリ値の表示形式を変更する次のボタンを表示します。 ただし, プログラム実行中は無効となります。
🔜 16 進数	メモリ値を 16 進数で表示します(デフォルト)。
🔛 符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
🔛 符号無し 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を8進数で表示します。
回 2進数	メモリ値を2進数で表示します。
サイズ表記	メモリ値のサイズの表示形式を変更する次のボタンを表示します。 ただし, プログラム実行中は無効となります。
[]4ビット	メモリ値を4ビット幅で表示します。
1バイト	メモリ値を8ビット幅で表示します(デフォルト)。
1162 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
114 バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード	文字列のエンコードを変更する次のボタンを表示します。
	たたし、フロクラム美行中は無効となります。
ASCII	文字列を ASCII コードで表示します(デフォルト)。
Shift_JIS	文字列を Shift_JIS コードで表示します。
EUC-JP	文字列を EUC-JP コードで表示します。
UTF-8	文字列を UTF-8 コードで表示します。
UTF-16	文字列を UTF-16 コードで表示します。
Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
Float	文字列を単精度浮動小数点数値で表示します。
Double	文字列を倍精度浮動小数点数値で表示します。
Float Complex	文字列を単精度浮動少数点数の複素数で表示します。
Double Complex	文字列を倍精度浮動少数点数の複素数で表示します。
Float Imaginary	文字列を単精度浮動小数点数の虚数で表示します。
Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表示	表示形式を変更する次のボタンを表示します。
■ スクロール範囲を 設定 …	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンします。
表示桁数を設定	メモリ値エリアの表示桁数を変更するため,表示桁数設定 ダイアログがオープンしま す。
表示アドレス・オフ セット値を設定	アドレス・エリアに表示するアドレスおよびオフセット値を設定するため,アドレ ス・オフセット設定 ダイアログがオープンします。

[[ファイル] メニュー (メモリ パネル専用部分)]

メモリ パネル専用の [ファイル] メニューは次のとおりです(その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

メモリ・データを保存	メモリの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存します(「(g) メモリ値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてメモリ・デー タを保存…]の選択と同等の動作となります。
名前を付けてメモリ・データ を保存	メモリの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に 保存するために, データ保存 ダイアログをオープンします(「(g) メモリ値の保 存」参照)。

[[編集] メニュー (メモリ パネル専用部分)]

メモリパネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。 ただし、プログラム実行中はすべて無効となります。

コピー	選択している範囲を文字列としてクリン	ップ・ボードにコピーします。
貼り付け	クリップ・ボードにコピーされている	文字列をキャレット位置に貼り付けます。
	- メモリ値エリアに貼り付ける場合:	「(e) コピー/貼り付け」参照
	- 文字列エリアに貼り付ける場合:	「(c) コピー/貼り付け」参照

RENESAS

検索	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は、メモリ値エリアと文字列エリアのうち、キャレットのあ るエリア内となります。

[コンテキスト・メニュー]

ウォッチ1に登録	キャレット位置のシンボルをウォッチパネル(ウォッチ1)に登録します。 ウォッチ式として登録される際は変数名として登録されるため、スコープにより 表示されるシンボル名は変化します。 ただし、キャレット位置のメモリ値に対応するアドレスにシンボルが定義されて いない場合は無効となります(「(f) ウォッチ式の登録」参照)。
検索	メモリ検索 ダイアログをオープンします。 検索対象となる箇所は、メモリ値エリアと文字列エリア(浮動小数点数値表示を 選択している場合を除く)のうち、キャレットのあるエリア内となります。 ただし、プログラム実行中は無効となります。
初期化	メモリ初期化 ダイアログをオープンします。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し、表示を更新します。
コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。 ただし、プログラム実行中は無効となります。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 ただし、プログラム実行中は無効となります。
	- メモリ値エリアに貼り付ける場合: 「(e) コピー/貼り付け」参照
	- 文字列エリアに貼り付ける場合: 「(c) コピー/貼り付け」参照
表記	メモリ値エリアの表示進数を指定するため、次のカスケード・メニューを表示し ます。 ただし、プログラム実行中は無効となります。
16 進数	メモリ値を 16 進数で表示します(デフォルト)。
符号付き 10 進数	メモリ値を符号付き 10 進数で表示します。
符号なし 10 進数	メモリ値を符号なし 10 進数で表示します。
8 進数	メモリ値を8進数で表示します。
2 進数	メモリ値を2進数で表示します。
サイズ表記	メモリ値エリアの表示幅を指定するため、次のカスケード・メニューを表示しま す。 ただし、プログラム実行中は無効となります。
4ビット	メモリ値を4ビット幅で表示します。
1バイト	メモリ値を8ビット幅で表示します(デフォルト)。
2 バイト	メモリ値を 16 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
4バイト	メモリ値を 32 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。
8バイト	メモリ値を 64 ビット幅で表示します。 対象メモリ領域のエンディアンに従って値を変換します。

エンコード		文字列エリアの表示形式を指定するため、次のカスケード・メニューを表示しま オ
		。 ただし、プログラム実行中は無効となります。
	ASCII	文字列を ASCII コードで表示します(デフォルト)。
	Shift_JIS	文字列を Shift_JIS コードで表示します。
	EUC-JP	文字列を EUC-JP コードで表示します。
	UTF-8	文字列を UTF-8 コードで表示します。
	UTF-16	文字列を UTF-16 コードで表示します。
	Half-Precision Float	文字列を半精度浮動小数点数値で表示します。
	Float	文字列を単精度浮動小数点数値で表示します。
	Double	文字列を倍精度浮動少数点数値で表示します。
	Float Complex	文字列を単精度浮動小数点数の複素数で表示します。
	Double Complex	文字列を倍精度浮動小数点数の複素数で表示します。
	Float Imaginary	文字列を単精度浮動少数点数の虚数で表示します。
	Double Imaginary	文字列を倍精度浮動小数点数の虚数で表示します。
表示		表示形式を変更するため、次のカスケード・メニューを表示します。
	スクロール範囲を設定 …	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンしま す。
	表示桁数を設定	メモリ値エリアの表示桁数を設定するため、表示桁数設定 ダイアログがオープン します。
	表示アドレス・オフセット 値を設定	アドレス・エリアに表示するアドレスのオフセット値を設定するため、アドレ ス・オフセット設定 ダイアログがオープンします。
強調表示		チェックすることにより、プログラムの実行により値が変更されたメモリ値を強 調表示します(デフォルト)。 ただし、プログラム実行中は無効となります。
リアルタイム表示更新設定		リアルタイム表示更新設定のため,次のカスケード・メニューを表示します (「(b) リアルタイム表示更新機能」参照)。
	 リアルタイム表示更新全体 設定	リアルタイム表示更新機能の全般設定を行うため,プロパティ パネルをオープン します。



X

.

逆アセンブル パネル

メモリ内容を逆アセンブルした結果(逆アセンブル・テキスト)の表示,ライン・アセンブル(「2.6.4 ライン・アセ ンブルを行う」参照),命令レベル・デバッグ(「2.9.3 プログラムをステップ実行する」参照),およびコード・カバ レッジ測定結果の表示【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX72T, RX72M, RX72N グループ】】(「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】」参照)を行います。

このパネルは,最大4個までオープンすることができます。各パネルは,タイトルバーの"逆アセンブル1","逆ア センブル2","逆アセンブル3","逆アセンブル4"の名称で識別されます。

混合表示モードにすることにより、コード・データに対応するソース・ファイル中のソース・テキストも表示することができます(デフォルト)。

なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- **注意** このパネルにフォーカスがある状態でプログラムをステップ実行した場合,実行単位は命令レベル単位 となります(「2.9.3 プログラムをステップ実行する」参照)。
- 備考 2. [ファイル]メニュー→[印刷 ...]を選択することにより,現在このパネルで表示している画像イメージ を印刷することができます。
- 逆アセンブルト 🕃 🛐 🌛 🎉 🎢 表示(1)+ [ツールバー]-RTSD 6703 HOCH ffff85a2 36: tutorial(); main: 🔶 ffff85a4 tutorial ffff85a7 BRA-B main 40: void tutorial (void) tutorial: 7100cc ffff85a9 ADD #-34H,R0,R0 p_sam = &g_Sample; 47: ffff85ac NOV .L #000000408H.R1 fb1208040000 ffff85b2 a301 NOV .L R1,30H[R0] 48: init(p_sam); ffff85b4 NOV -L 30H[R0],R1 ab01 057f0000 ffff85b8 BSR - A _init for(i=0; i<10; i++ 3e0b00) (NOV .L 50: 3 ffff85ba #00H,2CH[R0] ffff85bd BRA-B 2e28 RR-1 tutorial+3CH 4 (1) (2) (3)
- 図 A.6 逆アセンブル パネル(混合表示モードの場合)





図 A.8 逆アセンブル パネル(コード・カバレッジ測定結果を表示した場合)【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N】】



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (逆アセンブル パネル専用部分)]
- [[編集] メニュー (逆アセンブル パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [逆アセンブル] → [逆アセンブル 1 ~ 4] を選択



[各エリアの説明]

- (1) イベント・エリア イベントの設定が可能な行は、背景色を白色で表示します(背景色がグレー表示の行は、イベントの設定が不可 能であることを示すします)。 また、現在設定しているイベントがある場合、そのイベント設定行に、イベント種別を示すイベント・マークを 表示します。 このエリアは、次の機能を備えています。
 - (a) ブレークポイントの設定/削除 ブレークポイントを設定したい箇所をマウスでクリックすることにより、容易にブレークポイントを設定する ことができます。
 ブレークポイントは、クリックした行位置に対応する先頭アドレスの命令に対して設定されます。
 ブレークポイントを設定すると、設定した行にイベント・マークが表示されます。また、設定したブレークポ イントの詳細情報がイベントパネルに反映されます。
 なお、すでにいずれかのイベント・マークが表示されている箇所において、この操作を行った場合は、そのイ ベントを削除し、ブレークポイントの設定は行いません。
 ただし、イベントの設定は、背景色が白色で表示されている行に対してのみ行うことができます。
 ブレークポイントの設定方法についての詳細は、「2.10.2 任意の場所で停止する(ブレークポイント)」を参 照してください。
 - (b) 各種イベントの状態変更

各種イベント・マークを右クリックすることにより、次のメニューが表示され、選択したイベントの状態の変更を行うことができます。

有効化	選択しているイベントを有効状態にします。 指定されている条件の成立で、対象となるイベントが発生します。 なお、複数のイベントが設定されていることを示すイベント・マーク(いう) を選択している場合は、設定されているすべてのイベントを有効状態にします。
無効化	選択しているイベントを無効状態にします。 指定されている条件が成立しても、対象となるイベントは発生しません。 なお、複数のイベントが設定されていることを示すイベント・マーク(を選択している場合は、設定されているすべてのイベントを無効状態にします。
イベント削除	選択しているイベントを削除します。 なお,複数のイベントが設定されていることを示すイベント・マーク(を選択している場合は,設定されているすべてのイベントを削除します。
イベント パネルで詳細 を表示	選択しているイベントの詳細情報を表示するイベント パネルをオープンします。

(c) ポップアップ表示

イベント・マークにマウス・カーソルを重ねることにより、そのイベントのイベント名/詳細情報/イベント に付加されたコメントをポップアップ表示します。 なお、該当箇所に複数のイベントが設定されている場合、最大3つまで、各イベントの情報を列挙して表示 します。

(2) アドレス・エリア

行ごとの逆アセンブル開始アドレスを表示します(16進数表記固定)。 また、カレント PC 位置(PC レジスタ値)を示すカレント PC マーク())を表示します。 アドレス幅は、プロジェクトで指定しているマイクロコントローラのメモリ空間のアドレス幅となります。 なお、混合表示モード時におけるソース・テキスト行に対しては、開始アドレスに対応するソース・ファイル中 の行番号(xxx:)を表示します。 このエリアは、次の機能を備えています。

(a) ポップアップ表示

アドレス/ソース行番号にマウス・カーソルを重ねることにより、次の情報をポップアップ表示します。

アドレス	形式: 例 1: 例 2:	< <i>ロード・モジュール名</i> > ^注 \$< <i>ラベル 名</i> >+< <i>オフセット値</i> > test1.out\$main + 0x10 subfunction + 0x20
ソース行番号	形式: 例 1: 例 2:	< <i>ロード・モジュール名</i> > ^注 \$ < <i>ファイル名</i> > # < <i>行番号</i> > test1.out\$main.c#40 main.c#100



- 注 ロード・モジュール名は、ロード・モジュール・ファイルが複数ダウンロードされている場合 のみ表示します。
- (3) 逆アセンブル・エリア対象となるソース・テキスト行に続き、逆アセンブル結果行を次のように表示します。
 - 図 A.9 逆アセンブル・エリアの表示内容(混合表示モードの場合)



ラベル行		アドレスにラベルが定義されている場合は, ラベル名を表示し, 行全体を薄緑 色で強調表示します。	
カレント PC 行		カレント PC 位置(PC レジスタ値)のアドレスと対応する行を強調表示 ^{注1} し ます。	
ブレークポイント設定行		ブレークポイントが設定されている行を強調表示 ^{注 1} します。	
ソース・テキスト行		コード・データに対応するソース・テキストを表示します ^{注 2} 。	
逆アセンブ ル結果行	オフセット 値	アドレスにラベルが定義されていない場合は、一番近いラベルからのオフセット値を表示します ^{注 3} 。	
	命令コード	逆アセンブルの対象となったコードを 16 進数で表示します。	
	命令	逆アセンブル結果として命令を表示します。ニモニックは青色で強調表示しま す。	

- 注 1. 強調色は、オプション ダイアログにおける [全般 フォントと色] カテゴリの設定に依存します。
- 注 2. ツールバーの ボタン (トグル)のクリック,またはコンテキスト・メニューの [混合表示 モード] のチェックを外すことにより,ソース・テキストを非表示にすることができます (デ フォルトでチェックされています)。
- 注 3. オフセット値はデフォルトでは表示されません。表示する場合は、ツール・バーの
 ガリック、またはコンテキスト・メニューの [ラベルのオフセットを表示] を選択してください。

このエリアは、次の機能を備えています。

- (a) ライン・アセンブル
 表示されている命令/コードは、編集(ライン・アセンブル)することができます。
 操作方法についての詳細は、「2.6.4 ライン・アセンブルを行う」を参照してください。
- (b) 命令レベルでのプログラム実行 このパネルにフォーカスがある状態でプログラムをステップ実行することにより、命令レベル単位で実行を制 御することができます。 操作方法についての詳細は、「2.9.3 プログラムをステップ実行する」を参照してください。
- (c) 各種イベントの設定 コンテキスト・メニューの[ブレークの設定]/[トレース設定]/[タイマ設定]を選択することにより, 現在キャレットのあるアドレス/行に各種イベントを設定することができます。 イベントを設定することにより,対応するイベント・マークがイベント・エリアに表示されます。また,設定 したイベントの詳細情報がイベントパネルに反映されます。 ただし,イベントの設定は、イベント・エリアにおいて、背景色が白色で表示されている行に対してのみ行う ことができます。 イベントの設定方法についての詳細は次を参照してください。

- 「2.10.3 任意の場所で停止する(ブレーク・イベント)【E1】 【E20】 【EZ Emulator】」
- 「2.10.4 変数 /I/O レジスタへのアクセスで停止する」
- 「2.13.3 任意区間の実行履歴を収集する」
- 「2.13.4 条件を満たしたときのみの実行履歴を収集する」
- 「2.14.3 任意区間の実行時間を計測する」
- 備考 ブレークポイントの設定/削除は、イベント・エリアにおいても簡単に行うことができます (「(a) ブレークポイントの設定/削除」参照)。
- (d) ウォッチ式の登録
 表示されている C 言語変数 /CPU レジスタ /IO レジスタ /アセンブラ・シンボルをウォッチ式としてウォッチ パネルに登録することができます。
 操作方法についての詳細は、「2.11.6.1 ウォッチ式を登録する」を参照してください。
- (e) シンボル定義箇所への移動 シンボルを参照している命令にキャレットを移動した状態で、ツールバーの ポタンをクリック、またはコンテキスト・メニューの [シンボルへ移動] を選択することにより、キャレット位置のシンボルが定義されているアドレスにキャレット位置を移動します。 また、この操作に続き、ツールバーの ポタンをクリック、またはコンテキスト・メニューの [アドレスに 戻る] を選択すると、キャレット移動前のシンボルを参照している命令にキャレット位置を戻します (アドレ スはシンボルを参照している命令のアドレス値を表示)。
- (f) ソース行/メモリ値へのジャンプ コンテキスト・メニューの[ソースへジャンプ]を選択することにより,現在のキャレット位置のアドレスに 対応するソース行にキャレットを移動した状態でエディタパネルがオープンします(すでにオープンしてい る場合は,エディタパネルにジャンプ)。 また,同様に[メモリへジャンプ]を選択することにより,現在のキャレット位置のアドレスに対応するメモ リ値にキャレットを移動した状態でメモリパネル(メモリ1)がオープンします(すでにオープンしている 場合は,メモリパネル(メモリ1)にジャンプ)。
- (g) コード・カバレッジ測定結果の表示【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】
 カバレッジ機能を有効としている場合、プログラムの実行により取得したコード・カバレッジ測定結果を基 に、カバレッジ測定対象領域に相当する行を強調表示します。
 カバレッジ測定についての詳細は、「2.15 カバレッジの測定【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】」を参照してください。
- (h) 逆アセンブル・データの保存 [ファイル]メニュー→ [名前を付けて逆アセンブル・データを保存 ...]を選択することにより、データ保存 ダイアログをオープンし、このパネルの内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存す ることができます。 逆アセンブル・データの保存方法についての詳細は、「2.6.2.5 逆アセンブル結果の表示内容を保存する」を 参照してください。
- (i) 表示の拡大/縮小

逆アセンブルパネルにフォーカスがある状態で、メイン・ウインドウのツールバーある拡大率のドロップダウン・リストを変更することにより、逆アセンブルパネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、逆アセンブルパネルの表示を拡大し、表示が大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、逆アセンブルパネルの表示を縮小し、表示が小さくなります(最小 25%)。
- 備考 オプション ダイアログの設定により,次の項目をカスタマイズすることができます。
 - 表示フォント
 - 予約語/コメントの色分け



[ツールバー]

	デバッグ・ツールから最新の情報を取得し、表示を更新します。
	逆アセンブル結果とソース・テキストとの対応を表示する, 混合表示モー ドに設定します (デフォルト)。
1	キャレット位置をカレント PC 値に追従するように指定します。
	選択しているシンボルの定義位置へキャレットを移動します。
(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	🧱 ボタンで移動する直前の位置(<i>アドレス</i>)へ移動します。
表示	逆アセンブル・エリアの表示形式を変更する次のボタンを表示します。
🔁 ラベルのオフセットを表示	ラベルのオフセット値を表示します。アドレスにラベルが定義されていな い場合,一番近いラベルからのオフセット値を表示します。
🙀 アドレス値をシンボルで表示	アドレス値を " シンボル + オフセット値 " で表示します(デフォルト)。 ただし,アドレス値にシンボルが定義されている場合は,シンボルのみを 表示します。
	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープ ンします。

[[ファイル] メニュー(逆アセンブルパネル専用部分)]

逆アセンブル パネル専用の[ファイル]メニューは次のとおりです(その他の項目は共通)。 ただし,プログラム実行中はすべて無効となります。

逆アセンブル・データを保存	逆アセンブルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(h) 逆アセンブル・データの保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて逆アセンブル・ データを保存…]の選択と同等の動作となります。
名前を付けて逆アセンブル・ データを保存	逆アセンブルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存するために, データ保存 ダイアログをオープンします(「(h) 逆ア センブル・データの保存」参照)。
印刷	このパネルの内容を印刷するために、印刷アドレス範囲設定 ダイアログをオープ ンします。

[[編集] メニュー(逆アセンブル パネル専用部分)]

逆アセンブル パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

_ コピー	行を選択している場合,選択している行の内容を文字列としてクリップ・ボード にコピーします。 編集モードの場合,選択している文字列をクリップ・ボードにコピーします。
名前の変更	キャレット位置の命令/命令コードを編集するために、編集モードに移行します (「2.6.4 ライン・アセンブルを行う」参照)。 ただし、プログラム実行中は無効となります。
検索	検索・置換 ダイアログを[一括検索]タブが選択状態でオープンします。
置換	検索・置換 ダイアログを[一括置換]タブが選択状態でオープンします。
移動	指定したアドレスヘキャレットを移動するため,指定位置へ移動 ダイアログを オープンします。

[コンテキスト・メニュー]

【逆アセンブル・エリア/アドレス・エリア】

ウォッチ1に登録	選択している文字列,またはキャレット位置の単語をウォッチ式としてウォッチ パネル(ウォッチ1)に登録します(単語の判断は現在のビルド・ツールに依 存)。 ウォッチ式として登録する際は変数名として登録されるため,スコープにより 表示されるシンボル名は変化します。
アクション・イベントの登録	キャレット位置のアドレスにアクション・イベントを設定するため, アクショ ン・イベント ダイアログをオープンします。
ここまで実行	カレント PC 値で示されるアドレスから,キャレット位置の行に対応するアドレスまでプログラムを実行します。 ただし,プログラム実行中,または [ビルド&デバッグ・ツールへダウンロード]実行中は無効となります。
PC をここに設定	カレント PC 値を現在キャレットのある行のアドレスに変更します。 ただし、プログラム実行中、または [ビルド & デバッグ・ツールへダウンロー ド] 実行中は無効となります。
移動	指定したアドレスへキャレットを移動するため, 指定位置へ移動 ダイアログを オープンします。
シンボルへ移動	選択しているシンボルの定義位置へキャレットを移動します。
 アドレスへ戻る	[シンボルへ移動] で移動する直前の位置 (アドレス) へ移動します。 ただし, アドレスにシンボル名が表示されていない場合は無効となります。



ブレークの設定	ブレーク関連のイベントを設定するために、次のカスケード・メニューを表示します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます (「(1) イベント・エリア」参照)。
ハード・ブレークの設定	キャレット位置のアドレスにブレークポイント(ハードウエア・ブレーク・イ ベント)を設定します(「2.10.2 任意の場所で停止する(ブレークポイント)」 参照)。
ソフト・ブレークの設定 【E1】【E20】【EZ Emulator】	キャレット位置のアドレスにブレークポイント(ソフトウエア・ブレーク・イ ベント)を設定します(「2.10.2 任意の場所で停止する(ブレークポイント)」 参照)。
読み込みブレークを設定 【シミュレータ】	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/I/O レジスタに,リード・アク セスのブレーク・イベントを設定します(「2.10.4.1 変数 /I/O レジスタへのブ レーク・イベント(アクセス系)を設定する」参照)。
書き込みブレークを設定 【シミュレータ】	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/l/O レジスタに,ライト・アク セスのブレーク・イベントを設定します(「2.10.4.1 変数 /l/O レジスタへのブ レーク・イベント(アクセス系)を設定する」参照)。
読み書きブレークを設定 【シミュレータ】	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/l/O レジスタに,リード/ライ ト・アクセスのブレーク・イベントを設定します(「2.10.4.1 変数 /l/O レジス タへのブレーク・イベント(アクセス系)を設定する」参照)。
組み合わせブレークを設定 【E1】【E20】【EZ Emulator】	キャレット位置のアドレス/キャレット位置,または選択している変数(グローバル変数/関数内スタティック変数/ファイル内スタティック変数)/I/Oレジスタでのブレークを,組み合わせブレークイベントの条件として設定します(「2.10.3.1 ブレーク・イベント(実行系)を設定する」参照)。
読み込み組み合わせブレーク を設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として、キャレット位置、または選択している変数 (グローバル変数/関数内スタティック変数/ファイル内スタティック変数)/// Oレジスタのリード・アクセスによるブレーク・イベントを設定します (「2.10.4.1 変数///Oレジスタへのブレーク・イベント(アクセス系)を設定す る」参照)。
書き込み組み合わせブレーク を設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として、キャレット位置、または選択している変数 (グローバル変数/関数内スタティック変数/ファイル内スタティック変数)/// Oレジスタのライト・アクセスによるブレーク・イベントを設定します (「2.10.4.1 変数/I/Oレジスタへのブレーク・イベント(アクセス系)を設定す る」参照)。
読み書き組み合わせブレーク を設定 【E1】【E20】【EZ Emulator】	組み合わせブレークの条件として、キャレット位置、または選択している変数 (グローバル変数/関数内スタティック変数/ファイル内スタティック変数)/// Oレジスタのリード/ライト・アクセスによるブレーク・イベントを設定しま す(「2.10.4.1 変数///Oレジスタへのブレーク・イベント(アクセス系)を設 定する」参照)。
ブレーク動作の設定	プロパティ パネルをオープンし,ブレーク機能の設定を行います。



トレース設定	トレース関連のイベントを設定するために、次のカスケード・メニューを表示
	します。 なお、イベントは、イベントの設定が可能な行のみ設定することができます (「(1) イベント・エリア」参照)。
トレース開始の設定	キャレット位置のアドレスの命令が実行された際に、プログラムの実行履歴を 示すトレース・データの収集を開始するトレース開始イベントを設定します (「2.13.3.1 トレース開始イベント/トレース終了イベントを設定する」参照)。 【シミュレータ】 プロパティ パネル上の [トレース] カテゴリ内 [トレース機能を使用する] プロパティの設定を自動的に [はい] にします。
トレース終了の設定	キャレット位置のアドレスの命令が実行された際に、プログラムの実行履歴を 示すトレース・データの収集を終了するトレース終了イベントを設定します (「2.13.3.1 トレース開始イベント/トレース終了イベントを設定する」参照)。 【シミュレータ】 プロパティ パネル上の [トレース] カテゴリ内 [トレース機能を使用する] プロパティの設定を自動的に [はい] にします。
値をトレースに記録(読み込 み時)	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/I/O レジスタにリード・アクセ スした際に,その値をトレース・メモリに記録するポイント・トレース・イベ ントを設定します(「(1)変数/I/O レジスタへのアクセスが発生したとき」参 照)。
値をトレースに記録(書き込 み時)	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/I/O レジスタにライト・アクセ スした際に,その値をトレース・メモリに記録するポイント・トレース・イベ ントを設定します(「(1)変数/I/O レジスタへのアクセスが発生したとき」参 照)。
値をトレースに記録(読み書 き時)	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/I/O レジスタにリード/ライ ト・アクセスした際に,その値をトレース・メモリに記録するポイント・ト レース・イベントを設定します(「(1) 変数/I/O レジスタへのアクセスが発生し たとき」参照)。
読み書き時にトレース開始の 設定	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)/I/O レジスタにリード/ライ ト・アクセスした際に,トレースの記録を開始するトレース・イベントを設定 します(「2.13.3.1 トレース開始イベント/トレース終了イベントを設定する」 参照)。
読み書き時にトレース終了の 設定	キャレット位置,または選択している変数(グローバル変数/関数内スタ ティック変数/ファイル内スタティック変数)///O レジスタにリード/ライ ト・アクセスした際に,トレースの記録を終了するトレース・イベントを設定 します(「2.13.3.1 トレース開始イベント/トレース終了イベントを設定する」 参照)。
トレース結果の表示	トレース パネルをオープンし、取得したトレース・データを表示します。
トレース動作の設定	プロパティ パネルをオープンし、トレース機能の設定を行います。 ただし、トレーサ動作中は無効となります。



タイマ設定 ^{注 1}	タイマ関連のイベントを設定するために、次のカスケード・メニューを表示します(「2.14.3 任意区間の実行時間を計測する」参照)。 なお、イベントは、イベントの設定が可能な行のみ設定することができます (「(1) イベント・エリア」参照)。
実行時にタイマ開始	キャレット位置のアドレスの命令が実行された際に、プログラムの実行時間の 計測を開始するタイマ開始イベントを設定します(「(1) タイマ開始イベントの 設定方法」参照)。
実行時にタイマ終了	キャレット位置のアドレスの命令が実行された際に、プログラムの実行時間の 計測を終了するタイマ終了イベントを設定します(「(2) タイマ終了イベントの 設定方法」参照)。
読み書き時にタイマ開始	キャレット位置,または選択している変数(グローバル変数/関数内スタティック変数/ファイル内スタティック変数)/I/O レジスタにリード/ライト・アクセスした際に,プログラムの実行時間の計測を開始するタイマ開始イベントを設定します(「(1)タイマ開始イベントの設定方法」参照)。
 タイマ <n> に設定 【E1】【E20】【EZ Emulator】</n>	タイマ開始イベントを設定するチャネル ^{注 2} を指定します。
読み書き時にタイマ終了	キャレット位置,または選択している変数(グローバル変数/関数内スタティック変数/ファイル内スタティック変数)///Oレジスタにリード/ライト・アクセスした際に,プログラムの実行時間の計測を終了するタイマ終了イベントを設定します(「(2)タイマ終了イベントの設定方法」参照)。
タイマ <n> に設定 【E1】【E20】【EZ Emulator】</n>	タイマ終了イベントを設定するチャネル ^{注 2} を指定します。
タイマ結果の表示	イベント パネルをオープンし,タイマ関連のイベントのみ表示します。
カバレッジ情報をクリア 【シミュレータ】【E20【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】】	現在デバッグ・ツールが保持しているコード・カバレッジ測定結果をすべてク リアします。 ただし、使用するデバッグ・ツールがカバレッジ機能をサポートしていない場 合、この項目は表示されません。
命令の編集	キャレット位置の行の命令を編集するために、編集モードに移行します (「2.6.4 ライン・アセンブルを行う」参照)。 ただし、プログラム実行中は無効となります。
コードの編集	キャレット位置の行の命令コードを編集するために、編集モードに移行します (「2.6.4 ライン・アセンブルを行う」参照)。 ただし、プログラム実行中は無効となります。
表示	逆アセンブル・エリアの表示内容を設定するために、次のカスケード・メ ニューを表示します。
ラベルのオフセットを表示	ラベルのオフセット値を表示します。アドレスにラベルが定義されていない場 合, 一番近いラベルからのオフセット値を表示します。
アドレス値をシンボルで表示	アドレス値を"シンボル + オフセット値 "で表示します(デフォルト)。 ただし,アドレス値にシンボルが定義されている場合は,シンボルのみを表示 します。
スクロール範囲を設定 …	スクロール範囲を設定するためのスクロール範囲設定 ダイアログがオープンし ます。
混合表示	逆アセンブル結果とソース・テキストとの対応を表示する, 混合表示モードに 設定します (デフォルト)。
ソースヘジャンプ	キャレット位置のアドレスに対応するソース行にキャレットを移動した状態 で,エディタ パネルがオープンします。
メモリヘジャンプ	キャレット位置のアドレスに対応するメモリ値にキャレットを移動した状態 で、メモリ パネル(メモリ 1)がオープンします。



- 注 1. 【E1】【E20】【EZ Emulator】
 - RX140 グループ以外の RX100 シリーズでは,タイマ機能は未サポートとなります。
- 注 2. 指定可能なチャネル数については、以下のとおり RX600, RX700 シリーズと RX200 シリーズ, RX140 グループで異なります。 RX600, RX700 シリーズ: 2 (32bit×2) または 1 (64bit×1) RX200 シリーズ, RX140 グループ: 1 (24bit×1)

【イベント・エリア】(【E1】 【E20】 【EZ Emulator】)

ハードウエア・ブレークを優 先する	マウスのワンクリック操作で設定できるブレークの種類をハードウエア・ブレー クポイントとします(プロパティ パネル上の[ブレーク]【E1】【E20】【EZ Emulator】カテゴリ内[優先的に使用するブレークポイントの種類]プロパティ の設定に反映されます)。
ソフトウエア・ブレークを優 先する	マウスのワンクリック操作で設定できるブレークの種類をソフトウエア・ブレー クポイントとします(プロパティ パネル上の[ブレーク]【E1】【E20】【EZ Emulator】カテゴリ内[優先的に使用するブレークポイントの種類]プロパティ の設定に反映されます)。



CPU レジスタ パネル

CPU レジスタ(汎用レジスタ/制御レジスタ)の内容の表示,および値の変更を行います(「2.11.2 CPU レジスタを 表示/変更する」参照)。

- なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。
- 備考パネル上の各エリアの区切り線をダブルクリックすることにより,該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。
- 図 A.10 CPU レジスタ パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (CPU レジスタ パネル専用部分)]
- [[編集] メニュー (CPU レジスタ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [CPU レジスタ] を選択

[各エリアの説明]

(1) [レジスタ名] エリア

レジスタの種別をカテゴリ(フォルダ)として分類し,各レジスタ名を一覧表示します。 表示される各アイコンの意味は次のとおりです。 なお,カテゴリ名/レジスタ名を編集/削除することはできません。

Ы	このカテゴリに属するレジスタ名を表示している状態を示します。アイコンをダブルクリッ ク,または "-"マークをクリックすると,カテゴリを閉じてレジスタ名を非表示にします。
	このカテゴリに属するレジスタ名が非表示の状態を示します。アイコンをダブルクリック,または "+" マークをクリックすると,カテゴリを開いてレジスタ名を表示します。
11	レジスタ名を示します。アイコンをダブルクリック,または "+" / "-" マークをクリックする と,下階層のレジスタ名(レジスタの部分を表す名称)を表示/非表示します。



表示されるカテゴリ名/レジスタ名は次のとおりです(各レジスタ名の先頭の "+" マークの数は、表示される際の階層の深さを示します)。

表 A.2 [汎用レジスタ] カテゴリとレジスタ名【RX】

レジスタ名(別名)	ビット幅	レジスタ名(別名)	ビット幅
+ R0	32	+ R8	32
+ R1	32	+ R9	32
+ R2	32	+ R10	32
+ R3	32	+ R11	32
+ R4	32	+ R12	32
+ R5	32	+ R13	32
+ R6	32	+ R14	32
+ R7	32	+ R15	32

表 A.3 [倍精度浮動小数点データレジスタ]カテゴリとレジスタ名【RX】注5

レジスタ名(別名)	ビット幅	レジスタ名(別名)	ビット幅
+ DR0	64	+ DR8	64
+ DR1	64	+ DR9	64
+ DR2	64	+ DR10	64
+ DR3	64	+ DR11	64
+ DR4	64	+ DR12	64
+ DR5	64	+ DR13	64
+ DR6	64	+ DR14	64
+ DR7	64	+ DR15	64



レジスタ名	ビット幅	レジスタ名	ビット幅	レジスタ名	ビット幅
+ ISP	32	+ FPSW ^{注 2}	32	+ ACC ^{注 3}	64
+ USP	32	++ FS	1	+ ACC0 ^{注4}	72
+ INTB	32	++ FX	1	+ ACC1 ^{注4}	72
+ PC	32	++ FU	1	+ EXTB ^{注4}	32
+ PSW	32	++ FZ	1	+ DPSW ^{注 5}	32
++ IPL	4 ^{注1}	++ FO	1	+ DCMR ^{注5}	32
++ PM	1	++ FV	1	+ DECNT ^{注5}	32
++ U	1	++ EX	1	+ DEPC ^{注 5}	32
++	1	++ EU	1		
++ O	1	++ EZ	1	-	
++ S	1	++ EO	1	-	
++ Z	1	++ EV	1	-	
++ C	1	++ DN	1	-	
+ BPC	32	++ CE	1	-	
+ BPSW	32	++ CX	1	-	
+ FINTV	32	++ CU	1	-	
	<u> </u>	++ CZ	1	-	
		++ CO	1	-	
		++ CY	1	-	
		++ RM	2		

表 A.4	[制御レジスタ]	カテゴリとレジスタ名	[RX]

- 注 1. RX610 グループでは、3 ビットとなります。
- 注 2. FPU を搭載していないマイクロコントローラでは、FPSW レジスタをサポートしていません。
- 注 3. RX71M, RX66x, RX65x, RX64M, RX671, RX24x, RX23x, RX72T, RX72M, RX72N グループでは, ACC レジスタをサポートしていません。
- 注 4. ACC0, ACC1, EXTB レジスタは, RX71M, RX66x, RX65x, RX64M, RX671, RX24x, RX23x, RX72T, RX72M, RX72N グループのみでサポートしています。
- 注 5. 倍精度浮動小数点コプロセッサを搭載していないマイクロコントローラでは、サポートしていません。

このエリアは、次の機能を備えています。

- (a) ウォッチ式の登録
 CPU レジスタ/カテゴリをウォッチ式としてウォッチ パネルに登録することができます。
 操作方法についての詳細は、「2.11.6.1 ウォッチ式を登録する」を参照してください。
 - 備考 1. カテゴリを対象としてウォッチ式の登録を行った場合,そのカテゴリに属するすべての CPU レジスタがウォッチ式として登録されます。
 - 備考 2. 登録したウォッチ式には、自動的にスコープ指定が付与されます。
 - **注意** 【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 ACC0, ACC1 レジスタをウォッチ式としてウォッチ パネル**に登録して, 値の表示形式として** [Float] または [Double] が選択された場合は、[16 進数] が選択された場合と同じ値表示になりま す。

(2) [値] エリア

各 CPU レジスタの値を表示/変更します。

表示進数は、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。また、常に 16 進数表示を併記する表示形式を選択することもできます。

CPU レジスタの値として表示されるマークや色の意味は次のとおりです(表示の際の文字色/背景色はオプ ション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。

表示例(デフォルト)		オルト)	説明
0×0	文字色	青色	ユーザにより、値が変更されている CPU レジスタの値([Enter] キーにより
	背景色	標準色	ターゲット・メモリに書き込まれます)
0x0	文字色	茶色	プログラムの実行により、値が変化した CPU レジスタの値
	背景色	クリーム	フロクラムを再実行させることにより,強調色をリセットします。

このエリアは、次の機能を備えています。

(a) CPU レジスタ値の変更

CPU レジスタ値の変更は、対象 CPU レジスタ値を選択したのち、再度クリックし、キーボードからの直接入 カにより行います([Esc] キーの押下で編集モードをキャンセルします)。 CPU レジスタ値を編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、 デバッグ・ツールのレジスタに書き込まれます。

(b) CPU レジスタ値の保存

[ファイル] メニュー→ [名前を付けて CPU レジスタ・データを保存…]を選択することにより,名前を付けて保存 ダイアログをオープンし,このパネルのすべての内容をテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。 CPU レジスタ値の保存方法についての詳細は,「2.11.2.3 CPU レジスタの表示内容を保存する」を参照してください。

(c) 表示の拡大/縮小
 CPU レジスタ パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより、CPU レジスタ パネルの表示を拡大/縮小することができます。
 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、CPU レジスタ パネルの表示を拡大し、表示が大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、CPU レジスタ パネルの表示を縮小し、表示が小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。



[ツールバー]

	デバッグ・ツールから最新の情報を取得し,表示を更新します。 ただし,プログラム実行中は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
自動	選択している項目(下位項目を含む)の値を規定値で表示します(デフォルト)。
🔤 16 進数	選択している項目(下位項目を含む)の値を 16 進数で表示します。
🔛 符号付き 10 進数	選択している項目(下位項目を含む)の値を符号付き 10 進数で表示します。
🄛 符号なし 10 進数	選択している項目(下位項目を含む)の値を符号なし 10 進数で表示します。
8 進数	選択している項目(下位項目を含む)の値を8進数で表示します。
2 進数	選択している項目(下位項目を含む)の値を2進数で表示します。
ASCII	選択している項目(下位項目を含む)の文字列を ASCII コードで表示します。対象が 2 バイト以上ある場合は、1 バイトずつの文字を並べて表示します。
Filoat	選択している項目を Float で表示します。 ただし,4 バイト・データ以外の場合は,規定値で表示します。
Double	選択している項目を Double で表示します。 ただし,8 バイト・データ以外の場合は,規定値で表示します。
28	値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記します。

[[ファイル] メニュー (CPU レジスタ パネル専用部分)]

CPU レジスタ パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

CPU レジスタ・データを保存	このパネルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(b) CPU レジスタ値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて CPU レジス タ・データを保存 …]の選択と同等の動作となります。
名前を付けて CPU レジスタ・ データを保存	このパネルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存するために,名前を付けて保存 ダイアログをオープンします(「(b) CPU レジスタ値の保存」参照)。

[[編集] メニュー (CPU レジスタ パネル専用部分)]

CPU レジスタ パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り、クリップ・ボードにコピーします。 ただし、文字列を編集中の場合のみ有効となります。
コピー	編集中の場合,選択している文字列をクリップ・ボードにコピーします。 行を選択している場合,レジスタ/カテゴリをクリップ・ボードにコピーしま す。 なお,コピーした項目は,ウォッチパネルに貼り付け可能です。
貼り付け	クリップ・ボードにコピーされている文字列をキャレット位置に貼り付けます。 ただし、文字列を編集中の場合のみ有効となります。
すべて選択	すべての項目を選択状態にします。
検索	検索・置換 ダイアログを[一括検索]タブが選択状態でオープンします。
置换	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。



[コンテキスト・メニュー]

ウォッチ1に登録	選択しているレジスタ名/カテゴリをウォッチ パネル(ウォッチ 1)に登録しま す。
コピー	編集中の場合,選択している文字列をクリップ・ボードにコピーします。 行選択している場合,レジスタ項目/カテゴリをクリップ・ボードにコピーしま す。 なお,コピーした項目は,ウォッチパネルに貼り付け可能です。
表記	表示形式を指定するため、次のカスケード・メニューを表示します。
自動	選択している項目(下位項目を含む)の値を規定値で表示します(デフォルト)。
16 進数	選択している項目(下位項目を含む)の値を 16 進数で表示します。
符号付き 10 進数	選択している項目(下位項目を含む)の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目(下位項目を含む)の値を符号なし 10 進数で表示します。
8 進数	選択している項目(下位項目を含む)の値を8進数で表示します。
2 進数	選択している項目(下位項目を含む)の値を2進数で表示します。
ASCII	選択している項目(下位項目を含む)の文字列を ASCII コードで表示します。 対象が 2 バイト以上ある場合は、1 バイトずつの文字を並べて表示します。
Float	選択している項目を Float で表示します。 ただし,4 バイト・データ以外の場合は,規定値で表示します。
Double	選択している項目を Double で表示します。 ただし,8 バイト・データ以外の場合は,規定値で表示します。
16 進数値を併記	値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記します。

注意

【RX71M, RX66x, RX65x, RX64M, RX671, RX72T, RX72M, RX72N グループ】 ACC0, ACC1 レジスタに対して [Float] または [Double] が選択された場合は, [16 進数] が選択された 場合と同じ値表示になります。



IOR パネル

I/O レジスタの内容の表示,および値の変更を行います(「2.11.3 I/O レジスタを表示/変更する」参照)。 なお,このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- 注意
 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは、読み込み保護対象となるため、値の読み込みは行いません([値] に "?" を表示)。
 読み込み保護対象の I/O レジスタの内容を取得したい場合は、コンテキスト・メニューの[値を強制読み込み]を選択してください。
- 備考 パネル上の各エリアの区切り線をダブルクリックすることにより,該当エリアの内容を省略することな く表示可能な最小幅に変更することができます。
- 図 A.11 IOR パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (IOR パネル専用部分)]
- [[編集] メニュー (IOR パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [IOR] を選択

[各エリアの説明]

(1) 検索エリア I/O レジスタ名の検索を行います。

~	検索対象の文字列を指定します(大文字/小文字不問)。
	キーボードより文字列を直接入力するか(最大指定文字数:512 文字), ドロップダ
	ウン・リストより入力履歴項目を選択します(最大履歴数:10 個)。



テキスト・ボックスで指定している文字列を含む I/O レジスタ名を上方向に検索し, 検索結果を選択状態にします。
テキスト・ボックスで指定している文字列を含む I/O レジスタ名を下方向に検索し, 検索結果を選択状態にします。

備考 1. カテゴリ(フォルダ)により分類されて非表示の状態の I/O レジスタ名も検索します(展開して 選択状態となります)。

備考 2. 検索対象の文字列入力後, [Enter] キーを押下することにより, いボタンのクリックと同等の動作 を行い, [Shift]+[Enter] キーを押下することにより, かボタンのクリックと同等の動作を行いま す。

(2) [IOR] エリア

I/O レジスタの種別をカテゴリ(フォルダ)として分類し、各 I/O レジスタ名を一覧表示します。 表示される各アイコンの意味は次のとおりです。

H	このカテゴリに属する I/O レジスタ名を表示している状態を示します。アイコンをダブルク リック,または "-" マークをクリックすると、カテゴリを閉じ I/O レジスタ名を非表示にしま す。 なお、カテゴリはデフォルトでは存在しません。必要な場合は、カテゴリを新規作成したのち、 ツリーの編集を行ってください。
	I/O レジスタ名が非表示の状態を示します。アイコンをダブルクリック,または "+" マークをクリックすると,カテゴリを開き I/O レジスタ名を表示します。 なお,カテゴリはデフォルトでは存在しません。必要な場合は,カテゴリを新規作成したのち, ツリーの編集を行ってください。
3	I/O レジスタ名を示します。

備考 このエリアのヘッダ部をクリックすることにより、カテゴリ名を文字コード順でソートします (カテゴリ内 I/O レジスタ名も同様にソートします)。

このエリアは、次の機能を備えています。

(a) ツリーの編集

各 I/O レジスタを任意のカテゴリ(フォルダ)で分類し、ツリー形式を編集することができます。 カテゴリを新規に作成する場合は、作成したい I/O レジスタ名にキャレットを移動したのち、ツールバーの ボタンのクリック、またはコンテキスト・メニューの [カテゴリを作成] を選択し、任意にカテゴリ名称 を入力することにより行います(最大指定文字数:1024文字)。 なお、カテゴリを削除する場合は、削除したいカテゴリを選択したのち、ツールバーの ノ、またはコンテキスト・メニューの [削除] を選択します。ただし、削除できるカテゴリは、空のカテゴリ のみです。

また、カテゴリ名を編集する場合は、編集したいカテゴリ名を選択したのち、次のいずれかの操作により行い ます。

- 再度クリック後、キーボードよりカテゴリ名を直接編集
- [編集] メニュー→ [名前の変更] を選択後, キーボードよりカテゴリ名を直接編集
- [F2] キーを押下後、キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち、I/O レジスタ名をカテゴリ内に直接ドラッグ・アンド・ドロップすることにより、 各 I/O レジスタをカテゴリで分類したツリー形式で表示します。 同様に、カテゴリと I/O レジスタ名の表示の順番(上下位置)も、ドラッグ・アンド・ドロップ操作により自 由に変更することができます。

- **注意 1.** カテゴリ内にカテゴリを作成することはできません。
- 注意 2. I/O レジスタの追加/削除はできません。

(b) ウォッチ式の登録 I/O レジスタ/カテゴリをウォッチ式としてウォッチ パネルに登録することができます。 操作方法についての詳細は、「2.11.6.1 ウォッチ式を登録する」を参照してください。

- 備考 1. カテゴリを対象としてウォッチ式の登録を行った場合,そのカテゴリに属するすべての I/O レジスタがウォッチ式として登録されます。
- 備考 2. 登録したウォッチ式には、自動的にスコープ指定が付与されます。

(c) 表示の拡大/縮小

IOR パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・ リストを変更することにより、IOR パネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、IOR パネルの表示を拡大し、表示が大き く見やすくなります(最大 300%)。

- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、IOR パネルの表示を縮小し、表示が小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

(3) [値] エリア

I/O レジスタの値を表示/変更します。

表示進数は、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。また、常に 16 進数表示を併記する表示形式を選択することもできます。

I/O レジスタの値として表示されるマークや色の意味は次のとおりです(表示の際の文字色/背景色はオプションダイアログにおける[全般 - フォントと色]カテゴリの設定に依存)。

表示例(デフォルト)		オルト)	説明
0x0	文字色	青色	ユーザにより, 値が変更されている I/O レジスタの値([Enter] キーにより
	背景色	標準色	「ターゲット・メモリに書き込まれます)
0x0	文字色	茶色	プログラムの実行により、値が変化した I/O レジスタの値
	背景色	クリーム	「ツールハーの「茶」ホタン、またはコンテキスト・メニューの [表示色をリ セット] を選択することにより、強調表示をリセットします。
?	文字色	グレー	読み込み保護対象の I/O レジスタ ^注 の値
	背景色	標準色	

注 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタを示します。 読み込み保護対象の I/O レジスタの内容を取得する場合は、コンテキスト・メニューの[値を強 制読み込み]を選択することにより行ってください。

注意 1 バイト /2 バイト I/O レジスタと、1 バイト /2 バイト I/O レジスタに割り付けられている1 ビット I/O レジスタでは、値を取得するタイミングが異なります。このため、同一の I/O レジスタの 値を表示していても値が異なる場合があります。

備考 このエリアのヘッダ部をクリックすることにより、値を数値の昇順でソートします。

このエリアは、次の機能を備えています。

(a) I/O レジスタ値の変更

 I/O レジスタの値の変更は、対象 I/O レジスタ値を選択したのち、再度クリックし、キーボードからの直接入 カにより行います([Esc] キーの押下で編集モードをキャンセルします)。
 I/O レジスタ値を編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、デ バッグ・ツールのターゲット・メモリに書き込まれます。
 I/O レジスタ値の変更方法についての詳細は、「2.11.3.4 I/O レジスタの内容を変更する」を参照してください。

- (b) I/O レジスタ値の保存 [ファイル]メニュー→ [名前を付けて /OR データを保存 ...]を選択することにより、名前を付けて保存 ダ イアログをオープンし、I/O レジスタのすべての内容をテキスト・ファイル (*.txt) /CSV ファイル (*.csv) に保存することができます。 I/O レジスタ値の保存方法についての詳細は、「2.11.3.6 I/O レジスタの表示内容を保存する」を参照してくだ さい。
- (4) [型情報(バイト数)]エリア 各 I/O レジスタの型情報を次の形式で表示します。

- < I/O レジスタの種類> [< アクセス属性> < すべてのアクセス可能サイズ>] (< サイズ>)

RENESAS

アクセス属性	アクセス属性として、次のいずれかをを表示します。	
	R	リードのみ可能
	W	ライトのみ可能
	R/W	リード/ライト可能
すべてのアクセス可能サイズ	アクセス可能なサイズをビット単位で小さい順に","で区切り列挙します。	
サイズ	I/O レジスタのサイズを表示します。 バイト単位で表示可能な場合はバイト単位で、ビット単位でのみ表示可能な 場合はビット単位で単位を付与して表示します。	

例 1. 「IOR [R/W 1.8] (1 バイト)」の場合 リード/ライト可能, 1 ビット・アクセス /8 ビット・アクセス可能, サイズが 1 バイトの I/O レ ジスタ

例 2. 「IOR [R/W 1] (1 ビット)」の場合 リード/ライト可能, 1ビット・アクセス可能, サイズが1ビットの I/O レジスタ

備考 このエリアのヘッダ部をクリックすることにより、型情報を文字コード順でソートします。

(5) [アドレス] エリア

各 I/O レジスタがマッピングされているアドレスを表示します(16 進数表記固定)。

ただし、ビット・レジスタの場合は、次の例のようにビット・オフセット値を付与して表示します。

- 例 1. 「0xFF40」の場合 アドレス "0xFF40" に割り当てられている
- 例 2. 「0xFF40.4」の場合 アドレス "0xFF40" のビット 4 に割り当てられている(ビット・レジスタ)
- 備考 このエリアのヘッダ部をクリックすることにより、アドレスを数値の昇順でソートします。

8	デバッグ・ツールから最新の情報を取得し,表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし,プログラム実行中は無効となります。
(選択している I/O レジスタに対して、プログラム実行により値が変化したことを示す 強調表示をリセットします。 ただし、プログラム実行中は無効となります。
	新規カテゴリ(フォルダ)を追加します。テキスト・ボックスに直接カテゴリ名を入 カします。 なお、新規に作成できるカテゴリの数に制限はありませが、カテゴリ内にカテゴリを 作成することはできません。 ただし、プログラム実行中は無効となります。
×	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は、そのカテゴリを削除します(I/O レジスタの削除 不可)。
表記	値の表示形式を変更する次のボタンを表示します。

[ツールバー]

	🔤 16 進数	選択している項目の値を 16 進数で表示します(デフォルト)。
	🔜 符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
	🔛 符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
	6 進数	選択している項目の値を8進数で表示します。
	1 進数	選択している項目の値を2進数で表示します。
	ASCII	選択している項目の値を ASCII コードで表示します。
0000	20 20	選択している項目の値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記し ます。

[[ファイル] メニュー (IOR パネル専用部分)]

IOR パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

IOR データを保存	このパネルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(b) I/O レジスタ値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けて IOR データを 保存 …]の選択と同等の動作となります。
名前を付けて IOR データを保 存	このパネルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存するために,名前を付けて保存 ダイアログをオープンします(「(b) I/O レ ジスタ値の保存」参照)。

[[編集] メニュー (IOR パネル専用部分)]

IOR パネル専用の [編集] メニューは次のとおりです(その他の項目はすべて無効)。

切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します(I/O レ ジスタ/カテゴリの切り取り不可)。
コ ピー	選択している範囲の文字列をクリップ・ボードにコピーします。 I/O レジスタ/カテゴリが選択状態の場合は、その項目をコピーします。 なお、コピーした項目は、ウォッチ パネルに貼り付け可能です。
貼り付け	テキストが編集状態の場合,クリップ・ボードの内容をキャレット位置に挿入し ます(I/O レジスタ/カテゴリの貼り付け不可)。
削除	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は,その項目を削除します(I/O レジスタの削除 不可)。
すべて選択	テキストが編集状態の場合,すべての文字列を選択します。 テキストが編集状態以外の場合,すべての I/O レジスタ/カテゴリを選択状態に します。
名前の変更	選択しているカテゴリの名称を編集します。
検索	検索エリアのテキスト・ボックスにフォーカスを移動します。
移動	指定した I/O レジスタヘキャレットを移動するため, 指定位置へ移動 ダイアログをオープンします。

RENESAS

[コンテキスト・メニュー]

ウォッチ1に登録	選択している I/O レジスタ/カテゴリをウォッチ パネル(ウォッチ 1)に登録し ます。
最新の情報に更新	デバッグ・ツールから最新の情報を取得し,表示を更新します。 読み込み保護対象の I/O レジスタの再読み込みは行いません。 ただし,プログラム実行中は無効となります。
値を強制読み込み	読み込み保護対象の I/O レジスタの値を 1 回強制的に読み込みます。
移動	指定位置へ移動 ダイアログをオープンします。
カテゴリを作成	新規カテゴリ(フォルダ)を追加します。テキスト・ボックスに直接カテゴリ名 を入力します。 なお、新規に作成できるカテゴリの数に制限はありませが、カテゴリ内にカテゴ リを作成することはできません。 ただし、プログラム実行中は無効となります。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 I/O レジスタ/カテゴリが選択状態の場合は、その項目をコピーします。 なお、コピーした項目は、ウォッチ パネルに貼り付け可能です。
削除	選択している範囲の文字列を削除します。 空のカテゴリが選択状態の場合は,その項目を削除します(I/O レジスタの削除 不可)。
表記	表示形式を指定するため、次のカスケード・メニューを表示します。
16 進数	選択している項目の値を 16 進数で表示します(デフォルト)。
符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
符号なし 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を8進数で表示します。
2 進数	選択している項目の値を2進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
16 進数値を併記	選択している項目の値表示の末尾に,その値の 16 進数表記を"()"で囲んで併 記します。
 表示色をリセット	選択している I/O レジスタに対して、プログラム実行により値が変化したことを 示す強調表示をリセットします。



ローカル変数 パネル

ローカル変数の内容の表示,および値の変更を行います(「2.11.5 ローカル変数を表示/変更する」参照)。 なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- **注意 1.** プログラム実行中は、このパネルには何も表示されません。プログラムの実行が停止したタイミングで、各エリアの表示を行います。
- **注意 2.** コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データがスタック / レジスタに存在しない場合があります。この場合は対象となる変数は表示されません。
- 備考 パネル上の各エリアの区切り線をダブルクリックすることにより、該当エリアの内容を省略することな く表示可能な最小幅に変更することができます。
- 図 A.12 ローカル変数 パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (ローカル変数 パネル専用部分)]
- [[編集] メニュー (ローカル変数 パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [ローカル変数] を選択

[各エリアの説明]

(1) スコープ・エリア

表示するローカル変数のスコープをドロップダウン・リストにより選択します。 選択できる項目は次のとおりです。

項目	動作
カレント	カレント PC 値のスコープでのローカル変数を表示しま す。
< <i>深さ</i> >< <i>関数名</i> ()[ファイル名# 行番号] > ^注	呼び出し元の関数のスコープでのローカル変数を表示し ます。 プログラム実行後,選択したスコープが存在するかぎり, ここで選択したスコープを保ちます。



注 コール・スタック パネルで表示している関数呼び出し元を表示します。

(2) [名前]エリア ローカル変数名や関数名を表示します。関数の引数もローカル変数として表示します。 また、配列、ポインタ型変数、構造体/共用体は、階層構造をツリー形式で表示します。 このエリアを編集することはできません。 表示される各アイコンの意味は次のとおりです。

	変数を示します。 Auto 変数,内部スタティック変数,Register 変数の表示も行います ^注 。 配列,ポインタ型変数,構造体/共用体は,階層構造をツリー形式で表示します。 先頭に "+"マークがある場合は,これをクリックすることにより次を展開表示します(展開後 "- マークに変化)。			
	配列	配列中の全要素		
	ポインタ型変数	ポインタが指し示す先の変数 なお、ポインタが指し示す先がポインタの場合は、さらに "+" マークを付与し、 これをクリックすることにより参照先を表示します。 ただし、ポインタの指す値が不明な場合は、"?" を表示します。		
	構造体/共用体 構造体/共用体の全メンバ			
%	引数を示します。			
	関数を示します。			

注 Auto 変数を表示する場合,関数のプロローグ(関数の"{")やエピローグ(関数の"}")ではロー カル変数の正確な値を表示することができません(Auto 変数のアドレスは,スタック・ポインタ (SP)からの相対アドレスとなり,関数内で SP の値が確定するまで確定しません。プロローグ やエピローグでは SP の操作が行われており,SP の値が確定していません。このため,プロロー グやエピローグでは正確な値の表示ができません)。

このエリアは、次の機能を備えています。

(a) ウォッチ式の登録

C 言語変数をウォッチ式としてウォッチ パネルに登録することができます。 操作方法についての詳細は、「2.11.6.1 ウォッチ式を登録する」を参照してください。

備考 登録したウォッチ式には、自動的にスコープ指定が付与されます。

(b) メモリへのジャンプ

コンテキスト・メニューの [メモリヘジャンプ] を選択することにより, 選択しているローカル変数が配置さ れているアドレスにキャレットを移動した状態でメモリ パネル (メモリ 1) がオープンします (すでにオー プンしている場合はメモリ パネル (メモリ 1) にジャンプ)。

(c) 表示の拡大/縮小

ローカル変数 パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロッ プダウン・リストを変更することにより、ローカル変数 パネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、ローカル変数 パネルの表示を拡大し、表示が大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、ローカル変数 パネルの表示を縮小し、表示が小さくなります (最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

- (3) [値]エリア
 - ローカル変数の値を表示/変更します。

表示進数や文字列のエンコードは、ツールバーのボタン、またはコンテキスト・メニューより選択することがで きます。また、常に 16 進数表示を併記する表示形式を選択することもできます。 ローカル変数の値として表示されるマークや色の意味は次のとおりです(表示の際の文字色/背景色はオプショ ンダイアログにおける[全般 - フォントと色]カテゴリの設定に依存)。

表示例(デフォルト)			説明
0×0	文字色	青色	ユーザにより、値が変更されているローカル変数値([Enter] キーにより
	背景色	標準色	ターケット・メモリに書き込まれます)
0x0	文字色	茶色	プログラムの実行により、値が変化したローカル変数値 ^注
	背景色	クリーム	フロクラムを冉ひ実行することにより,強調色がリセットされます。
?	文字色	グレー	ローカル変数の値を取得できない場合
	背景色	標準色	

注 プログラムの実行開始位置からブレークした位置で同じ変数名を表示していて、かつ、その変数 値が変化している場合が対象となります。

このエリアは、次の機能を備えています。

(a) ローカル変数値/引数値の変更

ローカル変数値,および引数値の変更は,対象ローカル変数値を選択したのち,再度クリックし,キーボード からの直接入力により行います([Esc] キーの押下で編集モードをキャンセルします)。 ローカル変数値,および引数値を編集したのち,[Enter] キーの押下,または編集領域以外へのフォーカスの 移動により,デバッグ・ツールのターゲット・メモリに書き込まれます。 ローカル変数値/引数値の変更方法についての詳細は,「2.11.5.2 ローカル変数の内容を変更する」を参照し てください。

- (b) ローカル変数値の保存 [ファイル]メニュー→ [名前を付けてローカル変数データを保存...]を選択することにより、名前を付けて 保存ダイアログをオープンし、このパネルのすべての内容をテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存することができます。 ローカル変数値の保存方法についての詳細は、「2.11.5.3 ローカル変数の表示内容を保存する」を参照してく ださい。
- (4) [型情報(バイト数)]エリア ローカル変数の型名を表示します。表記はC言語の記述に従います。
 配列の場合は"[]"内に要素数を,関数の場合は"()"内にサイズ(バイト数)を付与して表示します。
 なお,このエリアを編集することはできません。
- (5) [アドレス] エリア ローカル変数のアドレスを表示します。変数がレジスタに割り当てれられている場合は、レジスタ名を表示しま す。 このエリアを編集することはできません。

[ツールバー]

ツールバーの各ボタンは、プログラム実行中は無効となります。

2	デバッグ・ツールから最新の情報を取得し、表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。



自動	このパネル上の値の表記を変数ごとの規定値で表示します。
 16 進数	このパネル上の値を 16 進数で表示します。
<mark>回</mark> 10 進数	このパネル上の値を 10 進数で表示します。
6 進数	このパネル上の値を8進数で表示します。
回 2進数	このパネル上の値を2進数で表示します。
🔤 配列のインデックスを 10 進数表記	このパネル上の配列のインデックスを 10 進数で表示します(デフォ ルト)。
遍 配列のインデックスを 16 進数表記	このパネル上の配列のインデックスを 16 進数で表示します。
Fla Float	このパネル上の値を Float で表示します。 ただし,4 バイト・データ以外,または型情報を持つ場合は,規定値 で表示します。
Double	このパネル上の値を Double で表示します。 ただし,8 バイト・データ以外,または型情報を持つ場合は,規定値 で表示します。
20	値表示の末尾に,その値の 16 進数表記を "()" で囲んで併記しま す。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
ASCII	文字列変数を ASCII コードで表示します。
Shift_JIS	文字列変数を Shift_JIS コードで表示します(デフォルト)。
EUC-JP	文字列変数を EUC-JP コードで表示します。
UTF-8	文字列変数を UTF-8 コードで表示します。
UTF-16	文字列変数を UTF-16 コードで表示します。

[[ファイル] メニュー (ローカル変数 パネル専用部分)]

ローカル変数 パネル専用の [ファイル] メニューは次のとおりです(その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

ローカル変数データを保存	このパネルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(b) ローカル変数値の保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてローカル変数 データを保存 …]の選択と同等の動作となります。
名前を付けてローカル変数 データを保存	このパネルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存するために,名前を付けて保存 ダイアログをオープンします(「(b) ロー カル変数値の保存」参照)。

[[編集] メニュー (ローカル変数 パネル専用部分)]

ローカル変数パネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。

コピー	選択している行の内容,または文字列をクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
名前の変更	選択しているローカル変数の値を変更するために、編集モードに移行します (「2.11.5.2 ローカル変数の内容を変更する」参照)。 ただし、プログラム実行中は無効となります。
検索	検索・置換 ダイアログを[一括検索]タブが選択状態でオープンします。



置換 ...

検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

コンテキスト・メニューの各項目は、プログラム実行中は無効となります。

5	フォッチ1に登録	選択しているローカル変数をウォッチ パネル(ウォッチ 1)に登録します。
-	1ピー	選択している行の内容、または文字列をクリップ・ボードにコピーします。
7	長記	表示形式を指定するために、次のカスケード・メニューを表示します。
	自動	このパネル上の値の表記を変数ごとの規定値で表示します(デフォルト)。
	16 進数	このパネル上の値を 16 進数で表示します。
	10 進数	このパネル上の値を 10 進数で表示します。
	8 進数	このパネル上の値を8進数で表示します。
	2 進数	このパネル上の値を2進数で表示します。
	配列のインデックスを 10 進 表記	このパネル上の配列のインデックスを 10 進数で表示します(デフォルト)。
	配列のインデックスを 16 進 表記	このパネル上の配列のインデックスを 16 進数で表示します。
	Float	このパネル上の値を Float で表示します。 ただし,4 バイト・データ以外,または型情報を持つ場合は,規定値で表示しま す。
	Double	このパネル上の値を Double で表示します。 ただし,8 バイト・データ以外,または型情報を持つ場合は,規定値で表示しま す。
1	6 進数値を併記	値表示の末尾に、その値の 16 進数表記を"()"で囲んで併記します。
-	ニンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
	ASCII	文字列変数を ASCII コードで表示します。
	Shift_JIS	文字列変数を Shift_JIS コードで表示します(デフォルト)。
	EUC-JP	文字列変数を EUC-JP コードで表示します。
	UTF-8	文字列変数を UTF-8 コードで表示します。
	UTF-16	文字列変数を UTF-16 コードで表示します。
;	くモリヘジャンプ	選択している行が示すアドレスにキャレットを移動した状態で, メモリ パネル (メモリ 1) がオープンします。
ウォッチ パネル

登録したウォッチ式の内容の表示,および値の変更を行います(「2.11.6 ウォッチ式を表示/変更する」参照)。 このパネルは,最大4個までオープンすることができます。各パネルは,タイトルバーの"ウォッチ1","ウォッチ 2","ウォッチ3","ウォッチ4"の名称で識別され,それぞれ個別にウォッチ式の登録/削除/移動を行うことができ ます。

ウォッチ式の登録はこのパネル上から行えますが、エディタ パネル/逆アセンブル パネル/メモリ パネル/ CPU レジスタ パネル/ローカル変数 パネル/ IOR パネルより行うことも可能です。

ウォッチ式が登録されている状態のパネルをクローズした場合,そのパネルは非表示となりますが,登録されていた ウォッチ式の情報は保持されます(再度そのパネルをオープンした際に,ウォッチ式が登録されている状態でオープン します)。

プログラムの実行後,ウォッチ式の値が変化すると表示を自動的に更新します(ステップ実行時には,ステップ実行 ごとに表示を逐次更新)。

また、リアルタイム表示更新機能を有効にすることにより、プログラム実行中であっても、値の表示をリアルタイム に更新することも可能です。

なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

注意 プログラム実行中に CPU レジスタの内容を表示/変更することはできません。

- 備考 パネル上の各エリアの区切り線をダブルクリックすることにより,該当エリアの内容を省略することな く表示可能な最小幅に変更することができます。
- 図 A.13 ウォッチ パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (ウォッチ パネル専用部分)]
- [[編集] メニュー (ウォッチ パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [ウォッチ] → [ウォッチ 1 ~ 4] を選択

[各エリアの説明]

(1) [ウォッチ式] エリア
 登録しているウォッチ式を一覧で表示します。
 このエリアの表タイトル部をクリックすることにより、一覧内のウォッチ式をアルファベット順でソートすることができます。
 また、カテゴリ(フォルダ)を自由に作成してウォッチ式を分類し、ツリー形式で表示することができます
 (「(a) ツリーの編集」参照)。

表示される各アイコンの意味は次のと	:お	りです	•
-------------------	----	-----	---

T	このカテゴリに属するウォッチ式を表示している状態を示します。アイコンをダブルクリッ ク,または "-" マークをクリックすると,カテゴリを閉じウォッチ式を非表示にします。
Ô	このカテゴリに属するウォッチ式が非表示の状態を示します。アイコンをダブルクリック, または "+" マークをクリックすると、カテゴリを開きウォッチ式を表示します。
Ð	ウォッチ式が変数であることを示します。 配列,ポインタ型変数,構造体/共用体を示すウォッチ式の先頭には,"+"/"-"マークを表 示し,これをクリックすることにより展開/折りたたみ表示します。
*	ウォッチ式が関数であることを示します。
125	ウォッチ式が即値であることを示します。
∱(€)	ウォッチ式が式であることを示します。
	ウォッチ式が I/O レジスタであることを示します。
	ウォッチ式が CPU レジスタであることを示します。 下階層のレジスタ(レジスタの部分)を持つウォッチ式の先頭には, "+" / "-" マークを表示 し, これをクリックすることにより展開/折りたたみ表示します。

このエリアは、次の機能を備えています。

(a) ツリーの編集

ウォッチ式をカテゴリ(フォルダ)で分類し、ツリー形式で表示することができます。 カテゴリを新規に作成する場合は、作成したい位置にキャレットを移動したのち、ツールバーの ボメンの クリック、またはコンテキスト・メニューの [カテゴリを作成]を選択し、任意にカテゴリ名称を入力するこ とにより行います。

なお, カテゴリを削除する場合は, 削除したいカテゴリを選択したのち, ツールバーの 🗙 ボタンのクリッ ク, またはコンテキスト・メニューの [削除] を選択します。

また、作成したカテゴリ名を編集する場合は、編集したいカテゴリ名を選択したのち、次のいずれかの操作により行います。

- 再度クリック後、キーボードよりカテゴリ名を直接編集
- [編集] メニュー→ [名前の変更] を選択後、キーボードよりカテゴリ名を直接編集
- [F2] キーを押下後、キーボードよりカテゴリ名を直接編集

カテゴリを作成したのち、登録済みのウォッチ式をカテゴリ内に直接ドラッグ・アンド・ドロップすることに より、各ウォッチ式をカテゴリで分類したツリー形式で表示します。 同様に、カテゴリとウォッチ式の表示の順番(上下位置)も、ドラッグ・アンド・ドロップ操作により自由に 変更することができます。

- **注意 1.** カテゴリ内にカテゴリを作成することはできません。
- **注意 2.** 1 つのウォッチ パネルにおいて,カテゴリは 64 個まで作成することができます(上限値を越えて作成しようとした場合,メッセージを表示します)。

備考 ウォッチ式/カテゴリを他のウォッチ パネル(ウォッチ 1 ~ウォッチ 4)にドラッグ・アンド・ドロップすると、ドロップ先のウォッチ パネルにウォッチ式/カテゴリがコピーされます。

(b) 展開/折りたたみ表示

配列,ポインタ型変数,構造体/共用体/クラス,レジスタ(部分を表す名前がついているもののみ)を示す ウォッチ式の先頭には、 "+"マークを表示し、これをクリックすることにより次を展開表示します(展開後 "-" マークに変化)。

ウォッチ式	展開表示の際の内容			
配列	配列中の全要素 コンテキスト・メニューの [表記] → [ASCII] を選択することにより,文字列とし て表示可能です(最大表示文字数:256文字)。 ただし,エンコードの種類により表示不可能な場合は,"."または"?"を表示します。			
ポインタ型変数	ポインタが指し示す先の変数			

ウォッチ式	展開表示の際の内容
構造体/共用体/ クラス	構造体/共用体/クラスの全メンバ
レジスタ	レジスタを構成するビット/ビット列の名称 例)PSW レジスタの場合 IPL レジスタ, PM レジスタ, U レジスタ, I レジスタ O レジスタ, S レジスタ, Z レジスタ, C レジスタ

(c) 新規ウォッチ式の登録

新規にウォッチ式を登録する方法には、次の3通りがあります。

- <1> 他のパネルからのウォッチ式の登録
 - 他のパネル上において、ウォッチ式として登録したい対象に対して、次のいずれかの操作を行います。
 - 対象文字列を選択したのち、任意のウォッチパネル(ウォッチ1~ウォッチ4)上のこのエリアに直接
 ドラッグ・アンド・ドロップ(エディタパネルを除く)
 - 対象文字列を選択したのち、または対象文字列のいずれかにキャレットを移動したのち(対象は自動的に 決定されます)、コンテキスト・メニューの[ウォッチ1に登録]を選択
 - 対象文字列を[編集] メニュー→ [コピー] したのち, 任意のウォッチ パネル (ウォッチ1~ウォッチ 4) 上のこのエリアで[編集] メニュー→ [貼り付け] を選択

なお、この操作が可能なパネルとウォッチ式として登録可能な対象との関係は次のとおりです。

パネル名	ウォッチ式として登録可能な対象
エディタ パネル	C/C++ 言語変数 ^{注 1} /CPU レジスタ /I/O レジスタ/アセンブラ・シンボル
逆アセンブル パネル	C/C++ 言語変数 ^{注 1} /CPU レジスタ /I/O レジスタ/アセンブラ・シンボル
CPU レジスタ パネル	CPU レジスタ ^{注 2}
ローカル変数 パネル	C/C++ 言語変数 ^{注1} (ローカル変数)
IOR パネル	I/O レジスタ ^{注 2}

表 A.5 各パネルとウォッチ式として登録可能な対象の関係

- 注 1. C89, C99, C++ 言語の変数を表します。
- 注 2. 自動的にスコープ指定がウォッチ式に付与されます。
- <2> ウォッチ パネル上での直接登録

任意のウォッチパネル(ウォッチ1~ウォッチ4)において、ツールバーの はコンテキスト・メニューの [新規ウォッチ式を追加]を選択することにより、このエリアの最下段に新規 ウォッチ式用のエントリ・ボックスが表示されます。 エントリ・ボックスの [ウォッチ式] エリアにおいて、キーボードより直接ウォッチ式を入力したのち、 [Enter] キーを押下します。 この際のウォッチ式の入力形式は次のとおりです。

表 A.6 ウォッチ式の入力形式

ウォッチ式	表示する値
C/C++ 言語変数名 ^{注 1}	C/C++ 言語の変数の値
<i>ウォッチ</i> 式[<i>ウォッチ式</i>]	配列の要素値
<i>ウォッチ式</i> .メンバ名 ^{注 2}	構造体/共用体/クラスのメンバ値
<i>ウォッチ式</i> -> メンバ名 ^{注 2}	ポインタの指し示す構造体/共用体/クラスのメンバ値
ウォッチ式 .* <i>キャスト式</i>	メンバ変数へのポインタの値
ウォッチ式 ->* <i>キャスト式</i>	メンバ変数へのポインタの値



ウォッチ式	表示する値
* <i>ウォッチ式</i>	ポインタの変数の値
& <i>ウォッチ式</i>	配置アドレス
(型名) <i>ウォッチ式</i>	指定された型へキャストした値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名 ^{注 3} , EQU シンボル名 ^{注 3} , [即値]	ラベルの値,EQU シンボルの値,即値アドレスの値
整定数	整数の定数値
浮動定数	浮動小数点の定数値
文字定数	文字定数值

注 1. C89, C99, C++ 言語の変数を表します。

- 注 2. 基底クラスのメンバ変数を指定する場合、メンバ名の前にスコープを指定してください。 (例:variable.BaseClass::member)
- 注 3. ラベル名または, EQU シンボル名に "\$" が含まれている場合,名前を "{}" で囲んでください。 (例:{\$Label}) 虚数の値には,大文字の "I" を掛けてください(例:1.0+2.0*I)。"I" はキーワードとなるた め, CPU レジスタの "I" を指定する場合は ":REG" を付加してください(例: I:REG)。

また、ウォッチ式は、スコープを指定して登録することができます。スコープ指定してウォッチ式を登録し た場合の扱いは次のとおりです。

スコープ指定	ロード・モジュー ル名 ^{注1}	ソース・ファ イル名 ^{注 1}	関数名 ^{注 2}	検索対象
prog\$file#func	prog	file	func	スタティック関数
prog\$func	prog	グローバル	func	グローバル関数
file#func	カレント	file	func	スタティック関数
func	カレント	カレント	func	すべて ^{注 2}

表 A.7 C/C++ 言語関数をスコープ指定してウォッチ登録した場合の扱い

注 1. ロード・モジュール名またはファイル名に空白や以下の記号が含まれている場合,名前をダ ブルクォーテーション""で囲んでください(例:"c:¥folder¥prog.abs"\$file.c#func)。 \$, #, (,), [,], &, ^, ~, %, +, -, *, /, :, ?, ', |, ¥, <, >, !

注 2. カレント PC 値のスコープからスタティック関数, グローバル関数の順で検索します。ス コープ範囲外のスタティック関数は検索対象外になります。 名前空間内に定義されている関数を指定する場合は,スコープを記入しないでください (例:Scope::func)。 また,同名の関数が存在する場合はパラメータの型名を明記してください(例:func(int, int))。

表 A.8 C 言語変数をスコープ指定してウォッチ登録した場合の扱い

スコープ指定	ロード・モ ジュール名 ^{注 1}	ソース・ ファイル名 ^{注1}	関数名 注2	変数名 注2	検索対象
prog\$file#func#var	prog	file	func	var	スタティック関数内スタ ティック変数 ^{注 2,3}
prog\$file#var	prog	file	グローバル	var	ファイル内スタティック変数



スコープ指定	ロード・モ ジュール名 _{注1}	ソース・ ファイル名 ^{注1}	関数名 注2	変数名 注2	検索対象
prog\$var	prog	グローバル	グローバル	var	グローバル変数
file#func#var	カレント	file	func	var	スタティック関数内スタ ティック変数 ^{注 2,3}
file#var	カレント	file	グローバル	var	ファイル内スタティック変数
var	カレント	カレント	カレント	var	すべて ^{注 4}

注 1. ロード・モジュール名またはファイル名に空白や以下の記号が含まれている場合,名前をダ ブルクォーテーション""で囲んでください(例: "c:¥folder¥prog.abs"\$file.c#func#var)。 \$, #, (,), [,], &, ^, ~, %, +, -, *, /, :, ?, ', |, ¥, <, >, !

注 2. 名前空間内に定義されている関数および変数を指定する場合は,スコープを記入してください(例:Scope::func)。 また,同名の関数が存在する場合はパラメータの型名を明記してください(例:func(int, int))。

- 注 3. カレント PC 値が指定関数内にある場合は、スタティック宣言されていないローカル変数も 検索対象となります。
- 注 4. カレント PC 値のスコープからローカル変数,ファイル内スタティック変数,グローバル変数の順で検索します。スコープ範囲外のローカル変数およびファイル内スタティック変数は,検索対象外となります。
- 表 A.9 CPU レジスタをスコープ指定してウォッチ登録した場合の扱い

スコープ指定	レジスタ・バンク	CPU レジスタ名
R10:REG	(なし)	R10

表 A.10 I/O レジスタをスコープ指定してウォッチ登録した場合の扱い

スコープ指定	I/O レジスタ名
AD0.ADCR:IOR	AD0.ADCR
AD0.ADCR	AD0.ADCR

備考 1. このエリアで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名 を補完することができます。(「2.21.2 シンボル名の入力補完機能」参照)。

- 備考 2. 即値は数値として扱われます。また、即値に演算子を使用することができます。
- 備考 3. ウォッチ式として、シンボルを使用した演算式を指定することができます。
- 備考 4. 同名の C/C++ 言語変数 /CPU レジスタ /I/O レジスタが存在する際に,スコープ指定せずにそれらを登録した場合,次の順にシンボルを解決し,値を表示します。 C/C++ 言語変数> CPU レジスタ> I/O レジスタ
- 備考 5. 同名のローカル変数とグローバル変数が存在する際に、スコープを指定せずにシンボル名の み登録した場合、カレント PC 値のスコープを元にシンボルを解決し、値を表示します。
- 備考 6. IOR パネル /CPU レジスタ パネルよりウォッチ式を登録した場合,ウォッチ式には自動的に スコープ指定が付与されます。
- 備考 7. ウォッチ式として単に "I" と指定した場合,虚数のキーワードとして解釈します。レジスタ "I" の値を取得する場合は,レジスタの後ろに ":REG" を付加してください。

<3> 他のアプリケーションからの登録 外部エディタなどから、C/C++ 言語変数 /CPU レジスタ /I/O レジスタ / アセンブラ・シンボルの文字列を 選択し、次のいずれかの操作を行います。

- 対象文字列を, 任意のウォッチパネル(ウォッチ1~ウォッチ4)上のこのエリアに直接ドラッグ・アンド・ドロップ

- 対象文字列をクリップ・ボードにコピーしたのち,任意のウォッチパネル(ウォッチ1~ウォッチ4) 上のこのエリアで[編集]メニュー→ [貼り付け]を選択

- **注意 1.** 1 つのウォッチ パネルにおいて,ウォッチ式は 128 個まで登録することができます(上限値を 越えて登録しようとした場合,メッセージを表示します)。
- **注意 2.** コンパイラによる最適化のため、対象となる変数を使用していないブロックでは変数データが スタック/レジスタに存在しない場合があります。この場合、対象となる変数をウオッチ式と して登録しても値の表示は "?" のままとなります。
- 備考 1. 各ウォッチ パネル(ウォッチ 1 ~ ウォッチ 4)上で登録したウォッチ式は、それぞれ個別に管理され、プロジェクトのユーザ情報として保存されます。
- 備考 2. ウォッチ式は、同名を複数登録することができます。
- (d) ウォッチ式の編集

登録済みのウォッチ式の編集は、対象ウォッチ式をダブルクリックし、対象ウォッチ式を編集モードにしたのち、キーボードから編集内容を直接入力して行います([Esc] キーの押下で編集モードをキャンセルします)。

ウォッチ式を編集したのち、[Enter] キーを押下すると編集を完了します。

- (e) ウォッチ式の削除 ツールバーのメボタンのクリック、またはコンテキスト・メニューの[削除]を選択することにより、選択 しているウォッチ式を削除します。
- (f) 各種イベントの設定

コンテキスト・メニューの[アクセス・ブレークの設定] / [トレース出力]を選択することにより,選択しているウォッチ式に各種イベントを設定することができます。 アクセス系のブレーク・イベントが設定された場合,ウォッチ式のアイコンが変化します(ウォッチ式のアイコンの下にブレーク・イベントのイベント・マークを重ねて表示)。トレース・イベントの場合は,ウォッチ式のマークに変化はありません。 イベントを設定することにより,設定したイベントの詳細情報がイベントパネルに反映されます。

ただし、イベントの設定は、対象となるウォッチ式がグローバル変数/関数内スタティック変数/ファイル内 スタティック変数 /I/O レジスタの場合にのみ行うことができます。 イベントの設定方法についての詳細は次を参照してください。

- 「2.10.4 変数 /I/O レジスタへのアクセスで停止する」

- 「2.13.4 条件を満たしたときのみの実行履歴を収集する」

- 「2.14.3 任意区間の実行時間を計測する」

(g) メモリ定義アドレスへのジャンプ

コンテキスト・メニューの [メモリヘジャンプ] を選択することにより,選択しているウォッチ式が定義され ているアドレスにキャレットを移動した状態でメモリ パネル (メモリ 1) がオープンします (すでにオープ ンしている場合は、メモリ パネル (メモリ 1) にジャンプ)。 ただし、同時に複数のウォッチ式を選択している場合、または I/O レジスタ /CPU レジスタを選択している場 合は、無効となります。

(h) 表示の拡大/縮小 ウォッチパネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダ ウン・リストを変更することにより、ウォッチパネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、ウォッチパネルの表示を拡大し、表示が 大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、ウォッチ パネルの表示を縮小し、表示が 小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

(2) [値] エリア

登録しているウォッチ式の値を表示/変更します。 なお、ウォッチ式が関数ポインタの場合は、関数名を表示します。 表示進数やエンコードは、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。 また、常に 16 進数値を併記する表示形式を選択することもできます。 なお、デフォルトの表示形式は、ウォッチ式の型に依存して、次のように自動的に決定されます。

表 A.11 ウォッチ式の表示形式(デフォルト)

ウォッチ式の型	表示形式
char, signed char, unsigned char	ASCII 文字に続き"()"内に 16 進数値を併記
short, signed short, short int, signed short int, int, signed, signed int, long, signed long, long int, signed long int	符号付き 10 進数値に続き"()"内に 16 進数値を併記
unsigned short, unsigned short int, unsigned, unsigned int, unsigned long, unsigned long int	符号なし 10 進数値に続き"()"内に 16 進数値を併記
float, float _Complex, float _Imaginary	Float(サイズが4バイトの場合)値 ^注
double, long double, double _Complex, long double _Complex, double _Imaginary, long dou- ble _Imaginary	Double(サイズが8バイトの場合)値 ^注
char, signed char, unsigned char へのポインタ	文字列 エンコード:Shift_JIS
char, signed char, unsigned char 以外へのポイン タ	16 進数
char, signed char, unsigned char 型の配列	文字列 エンコード:Shift_JIS
bit, boolean, _boolean	符号なし 10 進数値に続き"()"内に 16 進数値を併記
列挙型	列挙定数値に続き"()"内に 16 進数値を併記
ラベル,即値アドレス,EQU シンボル	符号付き 10 進数値に続き"()"内に 16 進数値を併記
ビット・シンボル	符号なし 10 進数値に続き"()"内に 16 進数値を併記
その他	16 進数

注 浮動小数点型および複素数型の値は、近似値(round to nearest)で丸めたものを表示します。

また、ウォッチ式の値として表示されるマークや色の意味は次のとおりです(表示の際の文字色/背景色はオプ ション ダイアログにおける [全般 - フォントと色] カテゴリの設定に依存)。

表示例(デフォルト)		ルト)	説明
0x0	文字色	青色	ユーザにより, 値が変更されているウォッチ式の値([Enter] キーにより
	背景色	標準色	│ ターケット・メモリに書き込まれます) │
0x0	文字色	ピンク	リアルタイム表示更新機能を行っているウォッチ式の値
	背景色	標準色	
0x0	文字色	茶色	プログラムの実行により、値が変化したウォッチ式の値
	背景色	クリーム	│ ツールバーの 🕎 ホタン, またはコンテキスト・メニューの [表示色をリ │ セット] を選択することにより, 強調表示をリセットします。
?	文字色	グレー	存在しない変数をウォッチ式として登録した場合、またはウォッチ式の値を 取得できなかった場合(変数がスコープを外れた場合など)
	背景色	標準色	

備考 1. 読み込み動作によってマイクロコントローラが動作してしまう I/O レジスタは、読み込み保護対象となり、値の読み込みは行いません。読み込み保護対象の I/O レジスタ の内容を読み込みたい 場合には、コンテキスト・メニューの [値を強制読み込み]を選択してください。

備考 2. 各ウォッチ式は、登録された順序で値の取得を行います。 このため、同一の I/O レジスタを複数登録した場合、値を取得するタイミングに差が生じるため、 表示される値が異なる場合があります。 備考 3. 16 進数値を併記している場合では,指定表記の値と 16 進数の値を個別に読み出します。 このため,値を取得するタイミングに差を生じるため,指定表記値と 16 進数値が異なる場合が あります。

このエリアは、次の機能を備えています。

- (a) リアルタイム表示更新機能
- リアルタイム表示更新機能を使用することにより、プログラムが停止している状態の時だけでなく、実行中の 状態であっても、登録したウォッチ式の値の表示/変更を行うことができます。 リアルタイム表示更新機能についての詳細は、「2.11.1.4 プログラム実行中にメモリの内容を表示/変更する」 を参照してください。
- (b) ウォッチ式の値の変更 ウォッチ式の値の変更は、対象ウォッチ式の値を選択したのち、再度クリックし、キーボードからの直接入力 により行います([Esc] キーの押下で編集モードをキャンセルします)。 ウォッチ式の値を編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、 ターゲット・メモリに書き込まれます。 ウォッチ式の値の変更方法についての詳細は、「2.11.6.6 ウォッチ式の内容を変更する」を参照してください。
- (c) ウォッチ式の値の保存

[ファイル] メニュー→ [名前を付けてウォッチ・データを保存...]を選択することにより,名前を付けて保存ダイアログをオープンし,このパネルのすべての内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存することができます。 また、コンテキスト・メニューの[ウォッチ・データを展開保存...]を選択することにより,名前を付けて保存ダイアログをオープンし,選択したウォッチ式の内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存することができます。 ウォッチ式の値の保存方法については、「2.11.6.9 ウォッチ式の表示内容を保存する」を参照してください。

(3) [型情報(バイト数)]エリア ウォッチ式に対して、次の形式で型情報を表示します。

ウォッチ式	表示形式		
単独の CPU レジスタ	<i>CPU レジスタの種類</i> >(<i><サイズ^{注1}></i>)		
単独の I/O レジスタ	< I/O レジスタの種類>(<アクセス属性> <アクセス・タイプ> <サイズ ^{注1} >)		
	アクセス属性	R : W : R/W :	読み出しのみ可能 書き込みのみ可能 読み出し/書き込み可能
	アクセス・タイプ	1 : 8 : 16 : 32 :	1 ビット・アクセス可能 バイト・アクセス可能 ハーフ・ワード・アクセス可能 ワード・アクセス可能
判別不能	?		
上記以外	く C コンパイラの#	制定に従っ	ったウォッチ式の型 ^{注 2} >(< <i>サイズ</i> ^{注 1} >)

- 注 1. ウォッチ式のサイズをバイト単位で示します。
 - ただし,ビット IOR/C 言語ビット・フィールドについては,ビット単位で表示し,数値の末尾に " ビット " 表記を付与します。
- 注 2. ウォッチ式をコンパイルする際に、どの型として扱われるかを示します。 double, long double 型は、CC-RX のプロパティの [共通オプション] タブ - [CPU] カテゴリ内に ある「double 型,および long double 型の精度」に従った型名を出力します。"単精度として扱う "が設定された場合は float 型、"倍精度として扱う"が設定された場合は double 型の型名とサイ ズを出力します。
- (4) [アドレス] エリア 各ウォッチ式がマッピングされているアドレスを表示します(16 進数表記固定)。 ただし、ウォッチ式が、単独の CPU レジスタの場合は "-" を、また判別不能の場合では、"?" を表示します。
 - 備考 ウォッチ式が I/O レジスタで、ビット・レジスタの場合は、次のようにビット・オフセット値を 付与して表示します。
 - 例 アドレス "0xFF40" のビット 4 に割り当てられている(ビット・レジスタ)の場合 表示内容: 0xFF40.4

(5) [メモ] エリア

ウォッチ式/カテゴリに対して、ユーザが自由にコメントを入力することができます。 このエリアに入力したコメントの内容は、各ウォッチ式/カテゴリに対して個別に保持され、プロジェクトの ユーザ情報として保存されます。したがって、ウォッチ式/カテゴリを削除すると、対応するメモの内容も破棄 されます。 ただし、配列、レジスタなどを展開表示している場合、各展開要素に対してコメントを入力することはできません。 コメントを編集する場合は、編集したい項目をダブルクリックすることにより、選択した項目が編集モードとな ります([Esc] キーの押下で編集モードをキャンセルします)。最大 256 文字までの文字列をキーボードより直 接入力することができます(改行コードは無効)。

文字列編集後, [Enter] キーの押下, または編集領域以外へのフォーカスの移動により, 文字列編集を完了します。

[ツールバー]

2	登録しているウォッチ式のすべての値を再取得し,表示を更新します。 ただし,読み込み保護対象の I/O レジスタの再読み込みは行いません。
1	選択しているウォッチ式に対して、プログラムの実行により値が変化したことを示す 強調表示をリセットします。 ただし、プログラム実行中は無効となります。
*	新規ウォッチ式を登録します。テキスト・ボックスに直接ウォッチ式を入力します (「(c) 新規ウォッチ式の登録」参照)。 なお,1つのウォッチ パネルに登録可能なウォッチ式数は,最大128個までです。
	新規カテゴリ(フォルダ)を追加します。テキスト・ボックスに直接カテゴリ名を入 カします。 なお、1 つのウォッチ パネルに作成可能なカテゴリ数は、最大 64 個までです(カテ ゴリ内のカテゴリ作成は不可)。
×	選択している範囲の文字列を削除します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
表記	値の表示形式を変更する次のボタンを表示します。
唐朝	選択しているウォッチ式の値の表記を変数ごとの規定値(「表 A.11 ウォッチ式の表 示形式(デフォルト)」参照)で表示します(デフォルト)。
🔤 16 進数	選択している項目の値を 16 進数で表示します。
🔙 符号付き 10 進数	選択している項目の値を符号付き 10 進数で表示します。
🔛 符号無し 10 進数	選択している項目の値を符号なし 10 進数で表示します。
8 進数	選択している項目の値を8進数で表示します。
1 進数	選択している項目の値を2進数で表示します。
ASCII	選択している項目の値を ASCII コードで表示します。
FIB Float	選択している項目の値を Float で表示します。 ただし,選択しているウォッチ式が 4 バイト・データの場合のみ有効となります。
Double	選択している項目の値を Double で表示します。 ただし,選択しているウォッチ式が 8 バイト・データの場合のみ有効となります。
100 C	選択している項目の値表示の末尾に,その値の 16 進数表記を"()"で囲んで併記し ます。 ただし,16 進数表記をしている場合は併記しません。

[[ファイル] メニュー (ウォッチ パネル専用部分)]

ウォッチ パネル専用の [ファイル] メニューは次のとおりです (その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

ウォッチ・データを保存	このパネルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(c) ウォッチ式の値の保存」参照)。 なお,起動後に初めてこの項目を選択した場合は,[名前を付けてウォッチ・ データを保存 …]の選択と同等の動作となります。
名前を付けてウォッチ・デー タを保存	このパネルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存するために、名前を付けて保存 ダイアログをオープンします(「(c) ウォッ チ式の値の保存」参照)。

[[編集] メニュー (ウォッチ パネル専用部分)]

ウォッチパネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。

切り取り	選択範囲の文字列を切り取り、クリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は、その項目を切り取ります。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲を文字列としてクリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は、その項目をコピーします。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
貼り付け	テキストが編集状態の場合、クリップ・ボードの内容をキャレット位置に挿入し ます。 テキストが編集状態以外の場合で、ウォッチ式がクリップ・ボードにコピーされ ている場合は、コピーされているウォッチ式をキャレット位置に登録します。
削除	選択している範囲の文字列を削除します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
すべて選択	テキストが編集状態の場合, すべての文字列を選択します。 テキストが編集状態以外の場合, すべてのウォッチ式/カテゴリを選択状態にし ます。
名前の変更	選択しているウォッチ式、またはカテゴリの名称を編集します。
検索	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

7	? クセス・ブレークの設定	この項目は,選択しているウォッチ式がグローバル変数/関数内スタティック 変数/ファイル内スタティック変数,および I/O レジスタの場合のみ有効です (複数選択不可)。 アクセス系のブレーク・イベントを設定するために,次のカスケード・メ ニューを表示します(「2.10.4.1 変数/I/O レジスタへのブレーク・イベント (アクセス系)を設定する」参照)。
	読み込みブレークを設定 【シミュレータ】	選択しているウォッチ式に、リード・アクセスのブレーク・イベントを設定しま す。
	書き込みブレークを設定 【シミュレータ】	選択しているウォッチ式に、ライト・アクセスのブレーク・イベントを設定しま す。
	読み書きブレークを設定 【シミュレータ】	選択しているウォッチ式に, リード/ライト・アクセスのブレーク・イベント を設定します。
	読み込み組み合わせブレーク を設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に、リード・アクセスのブレーク・イベントを、組み 合わせブレーク・イベントの条件として設定します。
	書き込み組み合わせブレーク を設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に、ライト・アクセスのブレーク・イベントを、組み 合わせブレーク・イベントの条件として設定します。
	読み書き組み合わせブレーク を設定【E1】【E20】【EZ Emulator】	選択しているウォッチ式に, リード/ライト・アクセスのブレーク・イベント を, 組み合わせブレーク・イベントの条件として設定します。
H	~ レース出力	この項目は、選択しているウォッチ式がグローバル変数/関数内スタティック 変数/ファイル内スタティック変数,および I/O レジスタの場合のみ有効です (複数選択不可)。 トレース関連のイベントを設定するために、次のカスケード・メニューを表示 します(「2.13.3 任意区間の実行履歴を収集する」および「2.13.4 条件を満た したときのみの実行履歴を収集する」参照)。 イベント設定後、対象となるウォッチ式の先頭にイベント・マークが表示され ます。
	値をトレースに記録(読み込 み時)	選択しているウォッチ式にリード・アクセスした際に,その値をトレース・メ モリに記録するポイント・トレース・イベントを設定します。(「(1) 変数 /l/O レジスタへのアクセスが発生したとき」参照)。
	値をトレースに記録(書き込 み時)	選択しているウォッチ式にライト・アクセスした際に,その値をトレース・メ モリに記録するポイント・トレース・イベントを設定します。(「(1) 変数 /l/O レジスタへのアクセスが発生したとき」参照)。
	値をトレースに記録(読み書 き時)	選択しているウォッチ式にリード/ライト・アクセスした際に、その値をト レース・メモリに記録するポイント・トレース・イベントを設定します。(「(1) 変数 /I/O レジスタへのアクセスが発生したとき」参照)。
	読み書き時にトレース開始の 設定	選択しているウォッチ式に対して、リード/ライト・アクセスした際にトレー スの記録を開始するトレース・イベントを設定します(「2.13.3.1 トレース開 始イベント/トレース終了イベントを設定する」参照)。
	読み書き時にトレース終了の 設定	選択しているウォッチ式に対して、リード/ライト・アクセスした際にトレー スの記録を終了するトレース・イベントを設定します(「2.13.3.1 トレース開 始イベント/トレース終了イベントを設定する」参照)。
	トレース	トレース パネルをオープンし,取得したトレース・データを表示します。

タイマ設定 ^{注1}	この項目は、選択しているウォッチ式がグローバル変数/関数内スタティック 変数/ファイル内スタティック変数、および I/O レジスタの場合のみ有効です (複数選択不可)。 タイマ関連のイベントを設定するために、次のカスケード・メニューを表示し ます(「2.14.3 任意区間の実行時間を計測する」参照)。 イベント設定後、対象となるウォッチ式の先頭にイベント・マークが表示され ます。
読み書き時にタイマ開始	選択しているウォッチ式に対して、リード/ライト・アクセスした際にタイマ を開始するイベントを設定します。(「(1) タイマ開始イベントの設定方法」参 照)。
タイマ <n> に設定 【E1】【E20】【EZ Emulator】</n>	タイマ開始イベントを設定するチャネル ^{注 2} を指定します。
読み書き時にタイマ終了	選択しているウォッチ式に対して、リード/ライト・アクセスした際にタイマ を終了するイベントを設定します。(「(2) タイマ終了イベントの設定方法」参 照)。
タイマ <n> に設定 【E1】【E20】【EZ Emulator】</n>	タイマ終了イベントを設定するチャネル ^{注 2} を指定します。
リアルタイム表示更新設定	リアルタイム表示更新設定のため,次のカスケード・メニューを表示します (「(a) リアルタイム表示更新機能」参照)。
リアルタイム表示更新全体設 定	リアルタイム表示更新機能の全般設定を行うため,プロパティ パネルをオープ ンします。
最新の情報に更新	登録しているウォッチ式のすべての値を再取得し,表示を更新します。 ただし,読み込み保護対象の I/O レジスタの再読み込みは行いません。
値を強制読み込み	読み込み保護対象の I/O レジスタの値を強制的に一度読み込みます。 ただし,プログラム実行中は無効となります。
新規ウォッチ式を追加	新規ウォッチ式を登録します。テキスト・ボックスに直接ウォッチ式を入力し ます(「(c) 新規ウォッチ式の登録」参照)。 なお、1 つのウォッチ パネルに登録可能なウォッチ式数は、最大 128 個までで す。
カテゴリを作成	新規カテゴリ(フォルダ)を追加します。テキスト・ボックスに直接カテゴリ 名を入力します。 なお,1 つのウォッチ パネルに作成可能なカテゴリ数は,最大 64 個までです (カテゴリ内のカテゴリ作成は不可)。
削除	選択している範囲の文字列を削除します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を削除します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
切り取り	選択している範囲の文字列を切り取ってクリップ・ボードに移動します。 ウォッチ式/カテゴリが選択状態の場合は、その項目を切り取ります。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
コピー	選択している範囲の文字列をクリップ・ボードにコピーします。 ウォッチ式/カテゴリが選択状態の場合は,その項目をコピーします。
貼り付け	テキストが編集状態の場合、クリップ・ボードの内容をキャレット位置に挿入 します。 テキストが編集状態以外の場合で、ウォッチ式がクリップ・ボードにコピーさ れている場合は、コピーされているウォッチ式をキャレット位置に登録します。 ただし、ウォッチ式の展開項目を選択している場合は無効となります。
名前の変更	選択しているウォッチ式、またはカテゴリの名称を編集します。
ウォッチ式をインポート	ウォッチ式をインポートするために、ウォッチ式データ・ファイルを開く ダイ アログをオープンします(「2.11.6.8 ウォッチ式をエクスポート/インポートす る」参照)。

表記		表示形式を指定するため、次のカスケード・メニューを表示します。
	自動	選択している項目の表記を変数ごとの規定値(「表 A.11 ウォッチ式の表示形式 (デフォルト)」参照)で表示します(デフォルト)。
	16 進数	選択している項目を 16 進数で表示します。
	符号付き 10 進数	選択している項目を符号付き 10 進数で表示します。
	符号なし 10 進数	選択している項目を符号なし 10 進数で表示します。
	8 進数	選択している項目を8進数で表示します。
	2 進数	選択している項目を2進数で表示します。
	ASCII	選択している項目を ASCII コードで表示します。
	16 進数値を併記	選択している項目の値表示の末尾に,その値の 16 進数表記を"()"で囲んで 併記します。 ただし,16 進数表記をしている場合は併記しません。
	Float	選択している項目を Float で表示します。 ただし,選択しているウォッチ式が 4 バイト・データ以外,または型情報を持 つ場合は,規定値(「表 A.11 ウォッチ式の表示形式(デフォルト)」参照)で 表示します。
	Double	選択している項目を Double で表示します。 ただし,選択しているウォッチ式が 8 バイト・データ以外,または型情報を持 つ場合は,規定値(「表 A.11 ウォッチ式の表示形式(デフォルト)」参照)で 表示します。
	配列のインデックスを 10 進表 記	すべての配列のインデックスを 10 進数で表示します。
	配列のインデックスを 16 進表 記	すべての配列のインデックスを 16 進数で表示します。
L	ニンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
	ASCII	選択している項目を ASCII コードで表示します。
	Shift_JIS	選択している項目を Shift_JIS コードで表示します(デフォルト)。
	EUC-JP	選択している項目を EUC-JP コードで表示します。
	UTF-8	選択している項目を UTF-8 コードで表示します。
	UTF-16	選択している項目を UTF-16 コードで表示します。
÷	トイズ表記	サイズを指定するため、次のカスケード・メニューを表示します。
	1バイト	選択している項目を8ビット・データとして表示します。
	2バイト	選択している項目を 16 ビット・データとして表示します。
	4バイト	選択している項目を 32 ビット・データとして表示します。
	8バイト	選択している項目を 64 ビット・データとして表示します。
,	「モリヘジャンプ	選択しているウォッチ式が定義されているアドレスヘキャレットを移動した状態でメモリ パネル (メモリ 1)をオープンします (「(g) メモリ定義アドレスへのジャンプ」参照)。
쿺		選択しているウォッチ式に対して、プログラムの実行により値が変化したこと を示す強調表示をリセットします。 ただし、プログラム実行中は無効となります。

ウォッチ・データを展開保存	選択しているウォッチ式の内容を指定したテキスト・ファイル(*.txt)/CSV
	ファイル(*.csv)に保存するために、名前を付けて保存 ダイアログをオープン
	します(「(c) ウォッチ式の値の保存」参照)。

- 注 1. 【E1】【E20】【EZ Emulator】
 - RX100 シリーズでは,タイマ機能は未サポートとなります。
- 注 2. 指定可能なチャネル数については、以下のとおり RX600, RX700 シリーズと RX200 シリーズで異なり ます。 RX600, RX700 シリーズ: 2 (32bit×2) または 1 (64bit×1)

RX600, RX700 シリース:2(32bit×2) または1(64b RX200 シリーズ:1(24bit×1)



コール・スタック パネル

関数呼び出しのコール・スタック情報の表示を行います(「2.12.1 コール・スタック情報を表示する」参照)。 なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- **注意** プログラム実行中は、このパネルには何も表示されません。 プログラムの実行が停止したタイミングで、各エリアの表示を行います。
- 図 A.14 コール・スタック パネル

	コール・スタック 🗵	Ē
[ツールバー] -	2 表記(N)+ エンコード(C)+ 100 ×	
	深さ コール・スタック]
	0 main()[C#WorkSpace#TutorialWE1E20#RX600#Tutorial_LittleEndian#Tutoria	j
	1 PowerON_Reset_PCO[C4WorkSpace¥Tutorial¥E1E20¥RX600¥Tutorial_Little	
		1
	(1) (2)	

- ここでは、次の項目について説明します。
- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (コール・スタック パネル専用部分)]
- [[編集] メニュー (コール・スタック パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [コール・スタック] を選択

[各エリアの説明]

(1) [深さ] エリア
 呼び出しの深さを表示します。
 カレント PC 位置を表示している行を0とし、呼び出し元に1から順に番号を付与します。

 (2) [コール・スタック] エリア 現在のソース位置とスタックに積まれているコール・スタック情報(関数呼び出し元位置/各関数の引数など) を表示します。 ツールバーの ボノビデボタン、またはコンテキスト・メニューの[引数表示]/[モジュール・ファイル名表 示]の選択による状態により、このエリアに表示する表示形式は次のように異なります。

状態	表示形式
- 引数を表示する	<関数> (< 引数> = < 引数値 ^注 > ,)[< モジュール・ファイル名
- モジュール・ファイル名を表示する	>3ヘンドイル石>#ヘ打笛号>](ナンオルト)

状態	表示形式
- 引数を表示する - モジュール・ファイル名を表示しな い	< <u>関数</u> > (<i><引数</i> > = <i><引数値^注></i> ,)[<i><ファイル名</i> > # <i><行番号</i> >]
- 引数を表示しない - モジュール・ファイル名を表示する	< <i>関数</i> > ()[< <i>モジュール・ファイル名</i> > \$ < <i>ファイル名</i> > # < <i>行</i> <i>番号</i> >]
- 引数を表示しない - モジュール・ファイル名を表示しな い	< <i>関数</i> > ()[< <i>ファイル名</i> > # < <i>行番号</i> >]

注 引数値が文字列の場合,最大20文字まで表示します。

備考 配列の引数は,配列としてではなくポインタとして渡されます(C言語仕様)。そのため,引数が 配列の場合,ポインタ扱いとして表示します。

このエリアは、次の機能を備えています。

(a) ソース行/逆アセンブル行へのジャンプ

コンテキスト・メニューの [ソースヘジャンプ] を選択することにより,現在のキャレット位置の関数呼び出 し元のソース行にキャレットを移動した状態でエディタパネルがオープンします(すでにオープンしている 場合は,エディタパネルにジャンプ)。 また,同様に [逆アセンブルヘジャンプ] を選択することにより,現在のキャレット位置の関数呼び出し元の アドレスを示す逆アセンブル行にキャレットを移動した状態で逆アセンブルパネル(逆アセンブル1)が オープンします(すでにオープンしている場合は、逆アセンブルパネル(逆アセンブル1)にジャンプ)。

備考行をダブルクリックすることでも、対象ソース行へジャンプすることができます。

(b) コール・スタック情報の保存 [ファイル] メニュー→ [名前を付けてコール・スタック・データを保存 ...]を選択することにより、名前を 付けて保存 ダイアログをオープンし、このパネルのすべての内容をテキスト・ファイル (*.txt) /CSV ファイ ル (*.csv) に保存することができます。 コール・スタック情報の保存方法についての詳細は、「2.12.1.4 コール・スタック情報の表示内容を保存す る」を参照してください。

(c) 表示の拡大/縮小

コール・スタック パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより、コール・スタック パネルの表示を拡大/縮小することができます。

また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、コール・スタック パネルの表示を拡大し、 表示が大きく見やすくなります(最大 300%)。

- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、コール・スタック パネルの表示を縮小し、 表示が小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

[ツールバー]

ツールバーの各ボタンは、プログラム実行中は無効となります。

2	デバッグ・ツールから最新の情報を取得し、表示を更新します。
表記	値の表示形式を変更する次のボタンを表示します。

🞰 自動	このパネル上の値の表記を変数ごとの規定値で表示します(デフォルト)。
🔤 16 進数	このパネル上の値を 16 進数で表示します。
10 進数	このパネル上の値を 10 進数で表示します。
8 進数	このパネル上の値を8進数で表示します。
2 進数	このパネル上の値を2進数で表示します。
エンコード	文字列変数のエンコードを変更する次のボタンを表示します。
ASCII	このパネル上の文字列変数を ASCII コードで表示します(デフォルト)。
Shift_JIS	このパネル上の文字列変数を Shift_JIS コードで表示します。
EUC-JP	このパネル上の文字列変数を EUC-JP コードで表示します。
UTF-8	このパネル上の文字列変数を UTF-8 コードで表示します。
UTF-16	このパネル上の文字列変数を UTF-16 コードで表示します。
~	モジュール・ファイル名を付加して表示します(デフォルト)。
~~	関数呼び出しのパラメータ(引数)を付加して表示します(デフォルト)。

[[ファイル] メニュー (コール・スタック パネル専用部分)]

コール・スタック パネル専用の [ファイル] メニューは次のとおりです(その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

コール・スタック・データを 保存	このパネルの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存します(「(b) コール・スタック情報の保存」参照)。 なお,起動後に初めてこの項目を選択した場合は,[名前を付けてコール・ス タック・データを保存 …]の選択と同等の動作となります。
名前を付けてコール・スタッ ク・データを保存	このパネルの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル(*.csv) に保存するために,名前を付けて保存 ダイアログをオープンします(「(b) コー ル・スタック情報の保存」参照)。

[[編集] メニュー (コール・スタック パネル専用部分)]

コール・スタックパネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。
すべて選択	項目をすべて選択状態にします。
検索	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換	検索・置換 ダイアログを [一括置換] タブが選択状態でオープンします。

[コンテキスト・メニュー]

コンテキスト・メニューの各項目は、プログラム実行中は無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします。	
モジュール・ファイル名表示	モジュール・ファイル名を付加して表示します(デフォルト)。	
引数表示	関数呼び出しのパラメータ(引数)を付加して表示します(デフォルト)。	

_		
4	表記	表示形式を指定するために、次のカスケード・メニューを表示します。
	自動	このパネル上の値の表記を変数ごとの規定値で表示します(デフォルト)。
	16 進数	このパネル上の値を 16 進数で表示します。
	10 進数	このパネル上の値を 10 進数で表示します。
	8 進数	このパネル上の値を8進数で表示します。
	2 進数	このパネル上の値を2進数で表示します。
-	エンコード	文字コードを指定するため、次のカスケード・メニューを表示します。
	ASCII	文字列変数を ASCII コードで表示します(デフォルト)。
	Shift_JIS	文字列変数を Shift_JIS コードで表示します。
	EUC-JP	文字列変数を EUC-JP コードで表示します。
	UTF-8	文字列変数を UTF-8 コードで表示します。
	UTF-16	文字列変数を UTF-16 コードで表示します。
ì	· 逆アセンブルヘジャンプ	選択している行が示す関数呼び出し元のアドレスにキャレットを移動した状態 で,逆アセンブル パネル(逆アセンブル 1)がオープンします。
	ノースヘジャンプ	選択している行が示す関数呼び出し元のソース行にキャレットを移動した状態 で,エディタ パネルがオープンします。
	このときのローカル変数を表 示	選択している行が示す関数のローカル変数を表示するローカル変数 パネルをオー プンします。



トレース パネル

プログラムの実行履歴を記録したトレース・データの表示を行います(「2.13 実行履歴の収集」参照)。 トレース・データは、デフォルトで逆アセンブル・テキストとソース・テキストを混合して表示しますが、表示モー ドを選択することにより、そのどちらか一方のみを表示させることもできます。 プログラムの実行停止後、最新のトレース・データが表示されるよう表示位置を自動更新します。 なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- **注意** 【E20(JTAG)【RX600, RX700 シリーズ】】 トレース機能/リアルタイム RAM モニタ機能(RRM 機能)は一部排他使用の機能です。
- 備考パネル上の各エリアの区切り線をダブルクリックすることにより,該当エリアの内容を省略することなく表示可能な最小幅に変更することができます。







図 A.17 トI	ノース	パネル【シミュレータ】						
	<u>ት</u> ሥታአ							8
[ツールバー]-	3) 🛞 🛞 🏔 表E(N)+ 🗂 🔮	S 🖾 🖏 🗹 👘					
L_	香泉	時間 (h:win:s.ws.us.ns)	行香号/アドレス	ソース/運	7センブル		アドレス	データ ・
	1833	00h00ain00s000as000us003ns	0xffff000c	+5	MOV.L	\$05H, 10H[R0]		
	1814	00h00win00s000ws000us000ns					0x000003ec	P 5
	1835		Tutorial.c#72	pap = 5;				
	1835	00h00min00s000ms000µs025ns	0511110601	+8	BRA.W	_sort+98H		
	1836		Tutorial.c270	shile(gap	> 0 >1			
	1836	00h00win00cem000ce00nim0000	0xfffff06a0	+158	MOV.L	10H[R0].RI		
	1817	00h00a1n00s000as000us000ns					0x000003ec	R 5
	11.1		Tutorial.c278	while(pup	>0.01			
	1838	00h00ain00s000as000us008ns	Dxffff06a2	+155	CMP	10H.RI		
	1111	ANA ANA 1-ANA ANA - ANA - ANA -	lutoris1.cE/8	while) pup	20.01			
	1835	00h00m1n00s000ms000us008ns	Extrinee4	+157	BLE-B	_50/11+A2H		
	114	000.00-1-00-000000000	Tutoris1.c2/3	while(pap	20.21	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
	1848	000000000000000000000000000000000000000	COTTITUERS	+128	BRA. W	_5011+681		
	11/1	00500+1+001+000++000++0002++	0.44440212	411	MW I	toou ocuipol		
	1845	00500=i=00=000==000==000==	001110012	*11	MOTION IN	#0001/00010003	0-000008-8	
	1846						0.0000000000	•
		l (,	l,		I
	(1)	(2)	(3)		(•	4)	(5)	(6)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [[ファイル] メニュー (トレース パネル専用部分)]
- [[編集] メニュー (トレース パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [トレース] を選択
- エディタ パネル/逆アセンブル パネルにおいて、コンテキスト・メニューの [トレース設定] → [トレース結果の 表示]を選択

[各エリアの説明]

- (1) [番号]エリアトレース・フレームに対応するトレース番号を表示します。
- (2) [時間 (h:min:s,ms,µs,ns)] エリア プログラムの実行開始から、各フレームの命令実行、またはメモリ・アクセスの要因が発生するまでに要した時 間を"時間、分、秒、ミリ秒、マイクロ秒、ナノ秒"の単位で表示します。
 - 備考 【シミュレータ】 時間表示を積算値とするか差分値とするかは、プロパティパネルの[デバッグ・ツール設定]タ ブ上の[トレース]カテゴリ内[トレース・タイム・タグを積算する]プロパティの設定に依存 します。
 - 注意 1. 【E1】【E20】【EZ Emulator】 タイム・スタンプ情報をトレース・データに付加するか否かは、プロパティパネルの[デバッ グ・ツール設定]タブ上の[トレース]カテゴリ内[タイム・スタンプ出力]プロパティで指定 します(「2.13.1 トレース動作の設定をする」参照)。なお、RX100 シリーズでは[タイム・スタ ンプ出力]プロパティの項目を変更することはできず、タイム・スタンプ情報をトレース・デー タに付加することはできません。
 - **注意 2.** 【E1】【E20】【EZ Emulator】 タイム・スタンプ情報は、プログラム実行開始からの時間ではなく、先頭サイクルを基点とした 相対時間を表示します。
 - **注意 3.** 【E1】【E20】【EZ Emulator】 タイム・スタンプ情報は, プロパティ パネルの[ダウンロード・ファイル設定] タブ上の[ト

(3)

	レース] カテゴリ イクロコントロ- マイクロコントロ RX600 シリー RX200 シリー	I内 [トレース・クロック・カウント・ソース [MHz]] プロパティの設定値とマ -ラごとのカウント・ソースでカウントした値によって算出されます。 コーラのカウンタの幅は,次のとおりです。 ズ:20 ビット ズ:24 ビット ス:24 ビット
	したタイム・スク ただし,連続する ス・データにはJ	シンプ情報が表示されます。 シトレース・フレーム間で2回以上オーバフローが発生した場合,以降のトレー Eしいタイム・スタンプ情報が表示されません。
注意 4.	【E1(Serial) / E20 トレース・パネル イベント パネル されません。	i(Serial) / EZ Emulator【RX200 シリーズ】】 レのタイム・スタンプ情報は , タイマ計測用カウンタを用いています。そのため , にてタイマ計測イベントを設定した場合,期待したタイム・スタンプ情報が表示
注意 5.	トレースのタイ』 [MHz]) は,マイ	ム・スタンプ情報のカウント周波数(トレース・クロック・カウント・ソース クロコントローラにより下記のように異なります。
	【RX610, RX621, (1)FXTAL 周波数	RX62N, RX62T, RX62G グループ】 / × 8<= 100 MHz の場合
		コック・カウント・ソース [MHz] = EXTAL 周波数 × 8
	(2)EXTAL 周波剱 トレース・クロ	. × 8> 100 MHz の場合 コック・カウント・ソース [MHz] = EXTAL 周波数 × 4
	【RX71M, RX66x (1)EXTAL または (a)SCKCR.ICk トレース・ (b)SCKCR.ICk	, RX65x, RX64M, RX630, RX631, RX63N, RX63T グループ】 : PLL をシステムクロック (ICLK) のクロックソースに使用する場合 (注) < を 1 分周に設定した場合 フロック・カウント・ソース [MHz] = 選択クロックソースの 1 分周 < を 1 分周以外に設定した場合
	トレース・4 (2)"EXTAL またに トレース・クロ	フロック・カウント・ソース [MHz] = 選択クロックソースの2分周 ま PLL をシステムクロック (ICLK) のクロックソースに使用しない場合 コック・カウント・ソース [MHz] = 選択クロックソースの1分周
	【RX210, RX21A トレース・クロ	, RX220 グループ】 コック・カウント・ソース [MHz] = ICLK(パフォーマンスカウンタを使用)
	注:ユーザプロ レースのタイム プ値の変化分に!	ブラム動作中にシステムクロック (ICLK) のクロックソースを切り替えた場合、ト ・スタンプ情報のカウントソースも切り替わります。この場合、タイム・スタン 単純に周波数を乗じても絶対時間にはなりません。
[行番号/ア	ドレス] エリア 	
テセンフル印 表示進数や文 きます。	っのアトレス, ま 字列のエンコード	には、ツールバーのボタン、またはコンテキスト・メニューより選択することがで
表示形式は次	のとおりです。	
表示	行の種類	表示形式
命令(逆ア	センブル)	<アドレス>
ソース・テキ	キスト	<ファイル名> # < 行番号>
ラベル		
ポイント・	トレース結果	_

備考次の実行履歴を表示しないため、行番号は連番にはなりません。

- CPU レジスタ・アクセス

- 無効フェッチ

 (4) [ソース / 逆アセンブル] エリア 収集したトレース・データを次のように表示します。 なお、表示モードの選択により、このエリアに表示される項目は異なります(「(a) 表示モード」参照)。

図 A 18	「ソース / 逆アヤンブル]	エリアの表示内容	(デフォルト)
四 A. 10			() / / // ! /

ラベル名・	ソース/逆アセンブル →nain)	
	tutorial(); +0 BSR.W	_tutorial
	_tutorial: void tutorial(void)	
	+0 ADD	#-34H, R0, R0
	+3 WOV.L	ジース・テキスト 単00000408H,R1
オフセット値	P_sam = &g_Sample; → +9 WOV.L	R1, 3 HIR0] 命令(逆アセンブル結果)
	init(p_san); +11 MOV.L	30H[R0],R1
	init(p_san); +13	_init

ラベル名	アドレスにラベルが定義されている場合は、ラベル名を表示します。
オフセット値	アドレスにラベルが定義されていない場合は,一番近いラベルからのオフセット 値を表示します。
ソース・テキスト	混合表示モード/ソース表示モードを選択している場合,対応するソース・テキ ストを表示します。 ただし,デバッグ情報が存在しない箇所を実行した場合は,"デバッグ情報なし" と表示します。
命令(逆アセンブル結 果)	混合表示モード/逆アセンブル表示モードを選択している場合,対応する命令 (逆アセンブル結果)を表示します ^注 。ニモニックは強調表示されます。

注 トレース・データの取りこぼしがあった場合は、"(LOST)"を表示し、該当行全体をエラー色で表 示します(エラー色はオプション ダイアログにおける [全般 - フォントと色] カテゴリの設定に 依存)。

このエリアは、次の機能を備えています。

(a) 表示モード

ツールバーのボタン,またはコンテキスト・メニューの選択により,次の3つの表示モードを選択することができます。

表示モード	表示内容
混合表示モード	命令(逆アセンブル)/ラベル名/ソース・テキスト(対応するソース行)/ データアクセス情報を表示します(デフォルト)。
逆アセンブル表示モード	命令(逆アセンブル)/ラベル名/データアクセス情報を表示します。
ソース表示モード	ソース・テキスト(対応するソース行)を表示します。 ただし,デバッグ情報が存在しない箇所を実行した場合は," デバッグ情報な し " と表示します。

(b) ソース行/逆アセンブル行へのジャンプ

コンテキスト・メニューの [ソースヘジャンプ] を選択することにより,現在のキャレット位置の行に対応するソース行にキャレットを移動した状態でエディタパネルがオープンします(すでにオープンしている場合は,エディタパネルにジャンプ)。 また,同様に [逆アセンブルヘジャンプ] を選択することにより,現在のキャレット位置の行のフェッチ・ア

ドレスにキャレットを移動した状態で逆アセンブル パネル(逆アセンブル 1)がオープンします(すでに オープンしている場合は, 逆アセンブル パネル(逆アセンブル 1)にジャンプ)。

(c) 他のパネルとの連動 ツールバーの パレイ ボタン,またはコンテキスト・メニューの [ウインドウ連動] → [ソースと連動] / [逆アセンブルと連動] を選択することにより、このパネル上のキャレット位置のアドレスをポインタとして、 エディタ パネル/逆アセンブル パネルで対応箇所を連動して表示させることができます(フォーカスの移動 は行いません)。

- (d) ポップアップ表示
 マウス・カーソルを行に重ねることにより、その行に対応するすべてのエリア(項目)のデータを縦並びに ポップアップ表示します。
- (e) トレース・データの保存 [ファイル]メニュー→ [名前を付けてトレース・データを保存 ...]を選択することにより、データ保存 ダイ アログをオープンし、このパネルの内容をテキスト・ファイル(*.txt)/CSV ファイル(*.csv)に保存するこ とができます。 トレース・データの保存方法についての詳細は、「2.13.9 実行履歴の表示内容を保存する」を参照してください。
- (f) 表示の拡大/縮小 トレースパネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより、トレースパネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。
 - [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、トレースパネルの表示を拡大し、表示が 大きく見やすくなります(最大 300%)。
 - [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、トレースパネルの表示を縮小し、表示が 小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

- (5) [アドレス] エリア メモリ・アクセスの対象アドレスを表示します。
 ただし、I/O レジスタへのアクセスの場合は、アドレスの代わりに I/O レジスタ名を表示します(アクセスが複数ある場合は次の行に表示)。
 表示進数は、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。
- (6) [データ] エリア

アクセスしたデータ値,およびその際のアクセス種別を表示します。 ただし,CPU レジスタ・アクセスは表示しません。

表示進数や文字列のエンコードは、ツールバーのボタン、またはコンテキスト・メニューより選択することができます。

データ値,およびアクセス種別の表示形式は次のとおりです(表示の際の文字色/背景色はオプション ダイア ログにおける [全般 - フォントと色] カテゴリの設定に依存)。

表示例(デフォルト)			メモリ・アクセス種別		
R <i>データ値</i>	文字色	標準色	リード・アクセス		
	背景色	薄緑			
W <i>データ値</i>	文字色	標準色	ライト・アクセス		
	背景色	オレンジ			
VECT データ値	文字色	標準色	ベクタ・リード・アクセス		
【シミュレータ】	背景色	薄緑			

[ツールバー]

2	デバッグ・ツールから最新の情報を取得し,表示を更新します。 ただし,プログラム実行中は無効となります。
5	トレース・メモリをクリア(初期化)し,このパネルの表示もクリアします。 ただし,プログラム実行中は無効となります。
۲	プログラム実行中に停止したトレース機能を再開します。 ただし、プログラム停止中、またはトレーサ動作中は無効となります。
۲	プログラム実行中にトレース機能を一時的に停止します。 ただし、プログラム停止中、またはトレーサ停止中は無効となります。



通		トレース検索 ダイアログをオープンします。
表記	3	値の表示形式を変更する次のボタンを表示します。 ただし, プログラム実行中は無効となります。
5	Head	このパネル上の値を 16 進数で表示します(デフォルト)。
8	Dee	このパネル上の値を 10 進数で表示します。
E	Det	このパネル上の値を8進数で表示します。
E	Bin	このパネル上の値を2進数で表示します。
ŋ		選択している行に連動してエディタ パネルをスクロールします。
劉		選択している行に連動して逆アセンブル パネルをスクロールします。
		表示モードを混合表示モードにします(デフォルト)。 ただし,プログラム実行中は無効となります。
<u>و</u>		表示モードを逆アセンブル表示モードにします。 ただし、プログラム実行中は無効となります。
		表示モードをソース表示モードにします。 ただし、プログラム実行中は無効となります。

[[ファイル] メニュー (トレース パネル専用部分)]

トレース パネル専用の[ファイル]メニューは次のとおりです(その他の項目は共通)。 ただし,プログラム実行中はすべて無効となります。

トレース・データを保存	トレース・データの内容を前回保存したテキスト・ファイル(*.txt)/CSV ファイ ル(*.csv)に保存します(「(e)トレース・データの保存」参照)。 なお、起動後に初めてこの項目を選択した場合は、[名前を付けてトレース・ データを保存…]の選択と同等の動作となります。
名前を付けてトレース・デー タを保存	トレース・データの内容を指定したテキスト・ファイル(*.txt)/CSV ファイル (*.csv)に保存するために、データ保存 ダイアログをオープンします(「(e) ト レース・データの保存」参照)。

[[編集] メニュー (トレース パネル専用部分)]

トレース パネル専用の [編集] メニューは次のとおりです (その他の項目はすべて無効)。 ただし、プログラム実行中はすべて無効となります。

コピー	選択している行の内容を文字列としてクリップ・ボードにコピーします(複数行 選択不可)。
検索	トレース検索 ダイアログをオープンします。

[コンテキスト・メニュー]

トレース・クリア	トレース・メモリをクリア(初期化)し,このパネルの表示もクリアします。 ただし,プログラム実行中は無効となります。
トレース開始	プログラム実行中に停止したトレース機能を再開します。 ただし,プログラム停止中,またはトレーサ動作中は無効となります。
トレース停止	プログラム実行中にトレース機能を一時的に停止します。 ただし,プログラム停止中,またはトレーサ停止中は無効となります。
検索	トレース検索 ダイアログをオープンします。 ただし,プログラム実行中は無効となります。

-	1 ピ ー	選択している行の内容を文字列としてクリップ・ボードにコピーします(複数行 選択不可)。 ただし、プログラム実行中は無効となります。			
泪	記合表示	表示モードを混合表示モードにします。 ただし、プログラム実行中は無効となります。			
ž	ゼアセンブル表示	表示モードを逆アセンブル表示モードにします。 ただし、プログラム実行中は無効となります。			
2	ノース表示 	表示モードをソース表示モードにします。 ただし、プログラム実行中は無効となります。			
쿺	表記	表示進数を指定するために、次のカスケード・メニューを表示します。 ただし、プログラム実行中は無効となります。			
	16 進数	このパネル上の値を 16 進数で表示します(デフォルト)。			
	10 進数	このパネル上の値を 10 進数で表示します。			
	8 進数	このパネル上の値を8進数で表示します。			
	2 進数	このパネル上の値を2進数で表示します。			
2	7インドウ連動	他のパネルとの連動を行うために、次のカスケード・メニューを表示します。			
	ソースと連動	選択している行に連動してエディタ パネルをスクロールします。			
	逆アセンブルと連動	選択している行に連動して逆アセンブル パネルをスクロールします。			
ji ji	を アセンブルヘジャンプ	選択している行のフェッチ・アドレスにキャレットを移動した状態で, 逆アセン ブル パネル(逆アセンブル 1)がオープンします。			
2	ノースヘジャンプ	選択している行に対応するソース行にキャレットを移動した状態で, エディタ パネルがオープンします。			
,	モリヘジャンプ	選択している行に対応するメモリ値にキャレットを移動した状態で, メモリ パネ ルがオープンします。			



イベント パネル

エディタ パネル/逆アセンブル パネル/ウォッチ パネル上で設定したイベントの詳細情報の表示,設定状態の有効 /無効の切り替え,および削除などを行います(「2.17 イベントの管理」参照)。 なお,このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

- 備考 1. イベントの設定に関しては、「2.17.7 イベント設定に関する留意事項」を参照してください。
- 備考 2. 解析ツールの関数 パネル/変数 パネルで設定したイベントもこのパネルで管理されます。
- 備考 3. パネル上の各エリアの区切り線をダブルクリックすることにより,該当エリアの内容を省略することな く表示可能な最小幅に変更することができます。
- 図 A.19 イベント パネル【E1】【E20】【シミュレータ】







ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]



- [[編集] メニュー (イベント パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [イベント] を選択
- エディタ パネル/逆アセンブル パネルにおいて、コンテキスト・メニューの [タイマ設定] → [タイマ結果の表示] を選択
- エディタ パネル/逆アセンブル パネルにおいて、イベント・マークにキャレットを移動したのち、コンテキスト・ メニューの [イベント パネルで詳細を表示]を選択

[各エリアの説明]

備考

(1) [名前]エリア 現在設定されているイベント名を次の形式で一覧表示します。



```
ツールバーのボタンの選択により、表示するイベント種別を限定することができます(「[ツール
バー]」参照)。
```

(a) チェック・ボックス
 イベントの設定状態を表示/変更します。
 なお、イベントの設定状態を変更すると、対応してイベント・マークも変化します。

1	有効状態	指定されている条件の成立で、対象となるイベントが発生します。 チェックを外すことにより、イベントを無効状態にすることができます。
	無効状態	指定されている条件が成立しても、対象となるイベントは発生しません。 チェックすることにより、イベントを有効状態にすることができます。
	保留状態	指定されている条件が, デバッグ対象のプログラムでは設定することができません。 チェック・ボックスを操作することはできません。

- 備考 1. Run-Break タイマ・イベント(EZ Emulator では未サポート)を無効状態/保留状態にすることはできません。
- 備考 2. 無条件トレース・イベントとトレース・イベントにおける有効/無効状態の設定は、排他制御となります。このため、ビルトイン・イベントである無条件トレース・イベントは、デフォルトで有効状態で設定されていますが、トレース開始イベント/トレース終了イベントのいずれかが設定されると同時に自動的に無効状態に変更され、トレース・イベント(トレース開始イベント/トレース終了イベントを1つにまとめたイベント)が有効状態になります。また逆に、設定されているトレース・イベントを無効状態にすると、自動的に無条件トレース・イベントが有効状態となります。
- 注意 【シミュレータ】 タイマ計測イベントを有効状態にするためには、タイマ開始イベントとタイマ終了イベントを 両方設定する必要があります。
- (b) イベント・マーク イベント・マークは、イベントの種別を示すとともに、現在の設定状態を示します。 表示されるイベント・マークとその意味は次のとおりです。

表 A.12 イベント・マーク

イベント種別	有効状態	無効状態	保留状態	備考
ハードウエア・ブレーク		Ş	-	_

イベント種別		有効状態	無効状態	保留状態	備考
ソフトウエア・ブレーク 【E1】【E20】【EZ Emulator】		÷	Ŵ	٩	_
組 【E	み合わせブレーク 1】【E20】【EZ Emulator】			•	_
無	条件トレース	_0	×	_	-
Rı	in-Break タイマ	_	—	—	_
۲	レース	P	\$	1	イベント パネルでのみ表示
	トレース開始	s e	si in the second se	J.Y.	エディタ パネル/逆アセンブル
	トレース終了	1 2	1	.∰¥	ハネルでのみ表示
 タイマ計測		<u>19</u>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Ø	イベント パネルでのみ表示
	タイマ開始	<u>13</u>	13	ø	エディタ パネル/逆アセンブル
	タイマ終了	<mark>1</mark> 9	1	Ø	ハネルでのみ表示
ポイント・トレース		* 0	*×	*	_
Printf イベント		s 😪	- V	1	_
割り込みイベント 【シミュレータ】		S	-	Ś	_
上記イベントの複数設定		2 注 1	注2	注3	エディタ パネル/逆アセンブル パネルでのみ表示

注 1. 複数のイベントの中で、1 つでも有効状態のイベントがある場合。

注 2. 複数のイベントの中で、有効状態のイベントがなく、1 つでも無効状態のイベントがある場合。

注 3. 複数のイベントのすべてが保留状態の場合。

(c) イベント名

イベント名として、イベント種別と ID 番号を表示します。 ID 番号は、イベント種別ごとに 0001 からの番号が自動的に付与されます(一度設定したイベントを削除した 場合でも ID 番号の振り直しは行いません)。 表示されるイベント種別は次のとおりです。

表 A.13 イベント種別

イベント種別	説明
ハードウエア・ブレー ク (ブレーク ^{注 1})	デバッグ・ツールが、プログラム実行中にブレーク条件を逐次確認し、条件を満 たした際にプログラムをブレークさせるイベントです。 →「2.10.2 任意の場所で停止する(ブレークポイント)」参照 →「2.10.4 変数 ///O レジスタへのアクセスで停止する」参照
ソフトウエア・ブレー ク (ブレーク ^{注 1}) 【E1】【E20】【EZ Emulator】	ブレークさせるアドレスの命令コードをブレーク用の命令に書き換え,その命令 を実行する際にプログラムをブレークさせるイベントです。 →「2.10.2 任意の場所で停止する(ブレークポイント)」参照
組み合わせブレーク 【E1】【E20】【EZ Emulator】	デバッグ・ツールが、プログラム実行中に複数のブレーク条件を逐次確認し、組 み合わせ条件を満たした際にプログラムをブレークさせるイベントです。 →「2.10.5 複数のブレーク・イベントを組み合わせる(組み合わせブレーク) 【E1】【E20】【EZ Emulator】」参照

イベント種別	説明
無条件トレース	プログラムの実行開始と同時に自動的にトレース・データを収集し、実行停止と ともにトレース・データの収集を停止します。 このイベントは、ビルトイン・イベント ^{注2} であるため、削除することはできま せん(デフォルトで有効状態で設定されています)。 →「2.13.2 実行停止までの実行履歴を収集する」参照
Run-Break タイマ	プログラムの実行開始と同時に自動的にプログラムの実行時間の計測を開始し、 実行停止とともに実行時間の計測を終了します。このイベントは、ビルトイン・ イベント ^{注2} であるため、削除することはできません(デフォルトで有効状態で 設定されています)。 →「2.14.2 実行開始から停止までの実行時間を計測する」参照
トレース	トレース開始イベント、およびトレース終了イベントにより設定された条件を満 たした際に、トレース・データの収集を開始/終了するイベントです(トレース 開始イベント/トレース終了イベントのいずれかが設定されると表示されます)。 →「2.13.3 任意区間の実行履歴を収集する」参照
タイマ計測 ^{注 3}	タイマ開始イベント,およびタイマ終了イベントにより設定された条件を満たした際に,プログラムの実行時間の計測を開始/終了するイベントです(タイマ開始イベント/タイマ終了イベントのいずれかが設定されると表示されます)。 →「2.14.3 任意区間の実行時間を計測する」参照
ポイント・トレース	プログラムの実行により,指定した変数 /I/O レジスタにアクセスした際に,その 情報をトレース・メモリに記録するイベントです。 →「2.13.4 条件を満たしたときのみの実行履歴を収集する」参照
Printf イベント	プログラムの実行を任意の箇所で一瞬停止させたのち、ソフトウエア処理により printf コマンドを実行させるイベントです(アクション・イベント)。 →「2.16.1 printf を挿入する」参照
割り込みイベント 【シミュレータ】	

- 注 1. マウスのワンクリック操作により設定されたブレークポイント(「2.10.2.2 ブレークポイント を設定する」参照)は、"ブレーク"と表示します。
- 注 2. デバッグ・ツールにデフォルトで設定されているイベントです。
- 注 3. [E1] [E20] [EZ Emulator] タイマ計測区間の番号としてイベント種別にチャネル番号が付与されます(例:タイマ計測 1)。タイマ計測可能な区間数は、マイクロコントローラによって異なります。なお、RX100 シリーズでは、タイマ機能は未サポートとなります。 - RX600, RX700 シリーズ:2区間

- RX200 シリーズ:1区間 【シミュレータ】

タイマ計測可能な区間数は、1区間です。

- 備考 上記のイベント種別のほか、解析ツールの関数 パネル/変数 パネルでイベント(ブレークポイ ント/ブレーク・イベント)を設定した場合、次のイベント種別を表示します。
 - 関数へのブレークポイントの場合: "関数への先頭へのブレーク"
 - 変数へのブレーク・イベントの場合: "変数のアクセス・ブレーク"
- (d) 表示の拡大/縮小

イベント パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダ ウン・リストを変更することにより、イベント パネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、イベント パネルの表示を拡大し、表示が 大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、イベント パネルの表示を縮小し、表示が 小さくなります(最小 50%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

(2) [詳細情報]エリア 各イベントに関する詳細情報を表示します。 表示される情報の内容は、イベント種別によって異なります。 イベント種別ごとの詳細情報の見方は次のとおりです。

表 A.14 イベント種別ごとの詳細情報

イベント種別		表示内容 ^{注 1}
ハードウエア・ブレーク (発生条件:実行系)	表示形式 1	<発生条件> <ファイル名# 行番号> <アドレス>
	表示例	実行前 main.c#39 0x100
		実行前 — 0x300
	表示形式 2	< <i>発生条件</i> > < <i>シンボル</i> + <i>オフセット</i> > <アドレス>
	表示例	実行前 funcA + 0x10 0x100
		実行前 — 0x300
ハードウエア・ブレーク (発生条件 : アクセス系)	表示形式 1	< <i>発生条件</i> > <ファイル名 # 変数名> <アドレス(範囲)> <比較条件> <比較値>
	表示例	リード main.c#variable1 0x100 - 0x101 == 0x5
		ライト sub.c#variable2 0x200 - 0x200 == 0x7
		リード/ライト sub2.c#variable3 0x300 - 0x303 == 0x8
	表示形式 2	< <i>発生条件</i> > <ファイル名# 関数名# 変数名> <アドレス(範 <i>囲)</i> > <比較条件> <比較値>
	表示例	リード main.c#func1#variable1 0x100 - 0x101 == 0x10
	表示形式 3	< <i>発生条件</i> > <変数名> <アドレス(範囲)> <比較条件> <比較値>
	表示例	ライト variable1 0x100 - 0x101 == 0x10
ソフトウエア・ブレーク	表示形式 1	<発生条件> <ファイル名# 行番号> <アドレス>
	表示例	実行前 main.c#40 0x102
		実行前 sub.c#101 0x204
	表示形式 2	< <i>発生条件</i> > < <i>シンボル</i> + オフセット> <アドレス>
	表示例	実行前 funcA + 0x12 0x102
組み合わせブレーク (発生条件:実行系, アクセ	表示形式	<組み合わせ条件> < <i>組み合わせブレークの詳細情報></i>
ス系) 【E1】【E20】【EZ Emulator】	表示例	OR
		- 実行後 main.c#100 0x300
		- 実行後 funcA + 0x10 0x100
		- ライト sub.c#variable2 0x200 - 0x200 == 0x7
		- リード/ライト sub2.c#variable3 0x300 - 0x303 == 0x8
無条件トレース	表示形式	_
	表示例	—



イベント種別		表示内容 ^{注1}
Run-Break タイマ	表示形式	総実行時間 : < <i>総実行時間</i> >
	表示例	総実行時間 : 1000ms
		総実行時間 : OVERFLOW
	表示形式 2	総実行時間 : < <i>総実行時間</i> > 実行サイクル数 : < <i>総実行サイク ル数</i> > 実行命令数 : < <i>総実行命令数</i> >【シミュレータ】
	表示例	総実行時間 : 3300ns 実行サイクル数 : 330 実行命令数 : 200
トレース (発生条件:実行系、アクセ ス系) 【E1】【E20】【EZ Emulator】	表示形式	開始:< <i>トレース開始の組み合わせ条件</i> > 終了:< <i>トレース終 了の組み合わせ条件</i> > 開始/終了の総数:< <i>トレース開始/ト レース終了イベントの総数</i> > ^{注 2} < <i>開始/終了</i> > < <i>トレース開始/トレース終了の詳細情報</i> >
	表示例	開始:OR 終了:OR 開始/終了の総数:6
		- 開始 実行後 main.c#100 0x300
		- 開始 実行後 funcA + 0x100 0x300
		- 開始 ライト variable1 0x100 - 0x101 == 0x10
		- 終了 実行後 main.c#200 0x100
		- 終了 実行後 funcA + 0x10 0x100
		- 終了 リード main.c#variable1 0x100 - 0x101 == 0x5
トレース (発生条件:実行系,アクセ ス系)	表示形式	開始:OR 終了:OR 開始/終了の総数:< <i>トレース開始/ト レース終了イベントの総数</i> > ^{注 2} <i><開始/終了</i> > < <i>トレース開始/トレース終了の詳細情報</i> >
	表示例	開始:OR 終了:OR 開始/終了の総数:6
		- 開始 実行前 main.c#100 0x300
		- 開始 実行前 funcA + 0x100_0x300
		- 開始 ライト variable1 0x100 - 0x101 == 0x10
		- 終了 実行前 main.c#200 0x100
		- 終了 実行前 funcA + 0x10 0x100
		- 終了 リード main.c#variable1 0x100 - 0x101 == 0x5
タイマ計測 (発生条件:実行系,アクセ	表示形式	総実行時間 : く <i>総実行時間</i> > 開始/終了の総数 : <タイマ開始 /タイマ終了イベントの総数> ^{注 2}
ス系) 【E1】【E20】【EZ Emulator】		- < <i>総実行時間</i> > <パス・カウント>
		- <開始/終了> <タイマ開始/タイマ終了の詳細情報>
	表示例	総実行時間 : 10ms 開始/終了の総数 : 6
		- 総実行時間 : 10ms パス・カウント : 5
		- 開始 実行後 main.c#100 0x300
		- 開始 実行後 funcA + 0x30 0x100
		- 開始 ライト variable1 0x100 - 0x101 == 0x10
		- 終了 実行後 main.c#100 0x300
		- 終了 実行後 funcA + 0x50 0x100
		- 終了 リード main.c#variable1 0x100 - 0x101 == 0x5

イベント種別		表示内容 ^{注 1}
タイマ計測 (発生条件:実行系,アクセ ス系) 【シミュレータ】	表示形式	総実行時間:< <i>総実行時間</i> >開始/終了の総数:< <i>タイマ開始</i> / <i>タイマ終了イベントの総数</i> > ^{注 2}
		- <総実行時間> <パス・カウント> <平均実行時間> <最 大行時間> <最小実行時間>
		- <開始/終了> <タイマ開始/タイマ終了の詳細情報>
	表示例	総実行時間 : 10ms 開始/終了の総数 : 2
		- 総実行時間 : 10ms パス・カウント : 5 平均実行時間 : 2ms 最大実行時間 : 4ms 最小実行時間 : 1ms
		- 開始 ライト variable1 0x100 - 0x101 == 0x10
		- 終了 実行前 main.c#100 0x300
ポイント・トレース	表示形式 1	< <i>発生条件</i> > < <i>変数名</i> > <変数の アドレス>
(充生余件:アクセス糸)	表示例	リード variable1 0x100
	表示形式 2	< <i>発生条件</i> > <ファイル名# <i>変数名</i> > <変数の <i>アドレス</i> >
	表示例	ライト sub.c#variable2 0x200
	表示形式 3	< <i>発生条件</i> > < <i>ファイル名</i> # 関数名 # <i>変数名</i> > <変数の <i>アドレ</i> <i>ス</i> >
	表示例	リード/ライト sub.c#func1#variable3 0x300
Printf イベント (アクション・イベント)	表示形式	< <i>発生条件</i> > <ファイル名# 行番号> <アドレス> < Print イ ベントの設定>
	表示例	実行前 main.c#39 0x100 aaa, bbb, ccc
		実行後 sub.c#100 0x200 aaa の結果の表示 : aaa
割り込みイベント (アクション・イベント) 【シミュレータ】	表示形式	< <i>発生条件</i> > <ファイル名# 行番号> <アドレス> 割り込み ベクタ : < <i>割り込みベクタ</i> > 優先順位 : < <i>割り込み優先順位</i> >
	表示例	実行前 main.c#39 0x100 割り込みベクタ : 1c 優先順位 : 7

注 1. 表示形式の詳細は次のとおりです。

< <i>発生条件</i> >	次の条件のいずれか 1 つを表示します。 【E1】【E20】 実行系: 実行前,実行後 アクセス系: リード,ライト,リード/ライト 【シミュレータ】 実行系: 実行前 アクセス系: リード,ライト,リード/ライト
< <i>ファイル名</i> # 行番号 >	ソース・ファイル名とソース・ファイル中の行番号を表示します。表 示形式はウォッチ式のスコープ指定式と同等です。 複数のロード・モジュール・ファイルをダウンロードしている場合で は、< <i>ロード・モジュール名</i> \$ <i>ファイル名#行番号</i> >を表示します。 なお、逆アセンブルパネルで設定されたイベントでは、次の場合、< 行番号>を<シンボル+オフセット>形式で表示します。
	 - 行情報があり、指定されたイベント設定位置が行情報の先頭でない場合 - 行情報がなく、シンボル情報がある場合
	また,次の場合は, <i>行番号を</i> "一"で表示します。 - 行情報がなく,シンボル情報がない場合

< 変数名>	ソース・ファイル中の変数名を表示します。表示形式はウォッチ式の スコープ指定式と同等です。
<比較条件>	比較の条件(==)を表示します。比較値が指定されなかった場合は表示しません。
<比較値>	比較値を表示します。比較値が指定されなかった場合は表示しません。
<アドレス>	指定された変数の、メモリ領域中の開始アドレス - 終了アドレスを表示します(16 進数表記固定)。
<組み合わせ条件>	次の条件のいずれか1つを表示します。 OR, AND, シーケンシャル
< <i>組み合わせブレーク の詳細情報</i> >	組み合わせブレーク・イベントに関する詳細情報を表示します。
< <i>総実行時間</i> >	タイマの総実行時間の測定結果を表示します。 単位は, ns/µs,/ms/s/min のいずれか 1 つが表示されます(ただ し, "min" の場合は "s" も同時に表示)。 なお, タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照), または不正な値の場合は "OVERFLOW" を表示します。
< <i>トレース開始の組み 合わせ条件</i> >	次の条件のいずれか1つを表示します。 OR, AND, シーケンシャル
< <i>トレース終了の組み 合わせ条件</i> >	OR 条件を表示します。
< <i>トレース開始/ト レース終了イベントの 総数</i> >	トレース開始イベントおよびトレース終了イベントの総数を表示しま す。
<i><開始/終了</i> >	詳細情報の内容が,開始イベントか終了イベントかを表示します。
<開始/終了> <トレース開始/ト レース終了の詳細情報 >	詳細情報の内容が,開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。
<開始/終了> <トレース開始/ト レース終了の詳細情報 > <パス・カウント>	 詳細情報の内容が,開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は "OVERFLOW"を表示します。
<開始/終了> <トレース開始/ト レース終了の詳細情報 > <パス・カウント> <平均実行時間>	詳細情報の内容が,開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は"OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/minのいずれか1つが表示されます(ただ し、"min"の場合は"s"も同時に表示)。
<開始/終了> <トレース開始/ト レース終了の詳細情報 > <パス・カウント> <平均実行時間> <最大実行時間>	 詳細情報の内容が、開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」参照)、または不正な値の場合は "OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/minのいずれか 1 つが表示されます(ただし、"min"の場合は "s"も同時に表示)。 タイマの最大実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/minのいずれか 1 つが表示されます(ただし、"min"の場合は "s" も同時に表示)。
< <i>開始/終了></i> <ドレース開始/ドレース終了の詳細情報	詳細情報の内容が、開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は"OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s"も同時に表示)。 タイマの最大実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s"も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s"も同時に表示)。
< <i>開始/終了></i> < <i>トレース開始/ト レース終了の詳細情報</i> > < <i>パス・カウント></i> < <i>平均実行時間></i> < <i>最大実行時間></i> < <i>最小実行時間></i> < <i>総実行サイクル数</i> >	詳細情報の内容が、開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は "OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最大実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。
< <i>開始/終了></i> < <i>ドレース開始/ト</i> <i>レース終了の詳細情報</i> ><パス・カウント><平均実行時間><最大実行時間><総実行サイクル数><総実行命令数>	詳細情報の内容が、開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は"OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最大実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 単位は、ns/µs,/ms/s/min のいずれか 1 つが表示されます (ただ し、"min"の場合は "s" も同時に表示)。 Run-Break 間の総実行サイクル数の測定結果を表示します。
< <i>開始/終了></i> < <i>トレース開始/ト レース終了の詳細情報</i> > < <i>パス・カウント></i> < <i>平均実行時間></i> < <i>最大実行時間></i> < <i>最小実行時間></i> < <i>総実行サイクル数></i> < <i>総実行命令数></i> < <i>Print イベントの設 定</i> >	 詳細情報の内容が、開始イベントか終了イベントかを表示します。 トレース開始イベントおよびトレース終了イベントに関する詳細情報 を表示します。 タイマのパス・カウントを表示します。 なお、タイマ・オーバフロー発生時(「2.14.4 測定可能時間の範囲」 参照)、または不正な値の場合は "OVERFLOW"を表示します。 タイマの平均実行時間の測定結果を表示します。 単位は、ns/µs/ms/s/min のいずれか 1 つが表示されます (ただし、"min" の場合は "s" も同時に表示)。 タイマの最大実行時間の測定結果を表示します。 単位は、ns/µs/ms/s/min のいずれか 1 つが表示されます (ただし、"min" の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 単位は、ns/µs/ms/s/min のいずれか 1 つが表示されます (ただし、"min" の場合は "s" も同時に表示)。 タイマの最小実行時間の測定結果を表示します。 アイマの最小実行時間の測定結果を表示します。 Run-Break 間の総実行サイクル数の測定結果を表示します。 Run-Break 間の総実行命令数の測定結果を表示します。 アクション・イベント ダイアログ上で指定した、

<割り込み優先順位> アクション・イベントダイアログ、または割り込みイベント詳細設定 ダイアログ【シミュレータ】上で指定した優先順位を表示します。

注 2. この行をクリックすることにより、下行の詳細情報を表示します。

(3) [コメント] エリア 設定されている各イベントに対して、ユーザが自由にコメントを入力できるエリアです。

コメントの入力は、コメントを入力したいイベントを選択後、このエリアをクリックするか、またはコンテキス ト・メニューの[コメントの編集]を選択したのち、任意のテキストをキーボードから直接入力します([Esc] キーの押下で編集モードをキャンセルします)。 コメントを編集したのち、[Enter] キーの押下、または編集領域以外へのフォーカスの移動により、編集を完了 します。

なお、コメントは最大256文字まで入力することができ、使用中のユーザの設定として保存されます。

[ツールバー]

注

×	選択しているイベント,およびイベント条件を削除します。 ただし,ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・ イベント)を削除することはできません。
	ハードウエア・ブレーク関連のイベントを表示します(デフォルト)。
[] [E1] [E20] [EZ Emulator]	ソフトウエア・ブレーク関連のイベントを表示します(デフォルト)。
\$	トレース関連のイベントを表示します(デフォルト)。
3	タイマ関連のイベントを表示します(デフォルト)。
V	アクション・イベント関連(Printf イベント,割り込みイベント)を表示します (デフォルト)。
9	ビルトイン・イベント関連(無条件トレース・イベント /Run-Break タイマ・イベ ント)を表示します(デフォルト)。
	選択しているイベント ^注 が設定されているアドレスに対応するソース行にキャレットを移動した状態で,エディタパネルがオープンします。
	選択しているイベント ^注 が設定されているアドレスに対応する逆アセンブル結果に キャレットを移動した状態で,逆アセンブル パネル(逆アセンブル 1)がオープン します。
	選択しているイベント ^注 が設定されているアドレスに対応するメモリ値にキャレットを移動した状態で,メモリパネル(メモリ1)がオープンします。

トレース・イベント/タイマ計測イベント/ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・イベント)以外のイベントが対象となります。

[[編集] メニュー (イベント パネル専用部分)]

イベントパネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。

削除	選択しているイベントを削除します。 ただし,ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・ イベント)を削除することはできません。
すべて選択	このパネルに表示されているすべてのイベントを選択状態にします。
検索	検索・置換 ダイアログを [一括検索] タブが選択状態でオープンします。
置換	検索・置換 ダイアログを[一括置換]タブが選択状態でオープンします。

[コンテキスト・メニュー]

有効化	選択しているイベントを有効状態にします。 ただし、選択しているイベントがすでに有効状態の場合は無効となります。
無効化	選択しているイベントを無効状態にします。 ただし,選択しているイベントがすでに無効状態の場合は無効となります。
削除	選択しているイベント,およびイベント条件を削除します。 ただし,ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・ イベント)を削除することはできません。
すべて選択	現在表示しているすべてのイベントを選択状態にします。
表示選択	表示するイベント種別を限定するために,次のカスケード・メニューを表示しま す。 デフォルトでは,すべての項目が選択されています。
ハードウエア・ブレーク	ハードウエア・ブレーク関連のイベントを表示します。
ソフトウエア・ブレーク	ソフトウエア・ブレーク関連のイベントを表示します。
タイマ	タイマ関連のイベントを表示します。
トレース	トレース関連のイベントを表示します。
アクション・イベント	アクション・イベント(Printf イベント)を表示します。
ビルトイン・イベント	ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ)を表示 します。
タイマ設定	タイマ関連の設定をするために,次のカスケード・メニューを表示します。 ただし,タイマ関連のイベントを選択している場合のみ有効です。
タイマの初期化	選択しているイベント(Run-Break タイマ・イベントを除く)で使用するタイマ を初期化します。
ナノ秒表示	選択しているイベントのタイマ結果をナノ秒(ns)単位で表示します。
マイクロ秒表示	選択しているイベントのタイマ結果をマイクロ秒(μs)単位で表示します。
ミリ秒表示	選択しているイベントのタイマ結果をミリ秒(ms)単位で表示します。
秒表示	選択しているイベントのタイマ結果を秒(s)単位で表示します。
分表示	選択しているイベントのタイマ結果を分(min)単位で表示します。
メモリヘジャンプ	選択しているイベント ^{注 1} が設定されているアドレスに対応するメモリ値にキャ レットを移動した状態で、メモリ パネル(メモリ 1)がオープンします。
逆アセンブルヘジャンプ	選択しているイベント ^{注 1} が設定されているアドレスに対応する逆アセンブル結 果にキャレットを移動した状態で,逆アセンブル パネル(逆アセンブル 1)が オープンします。
ソースヘジャンプ	選択しているイベント ^{注1} が設定されているアドレスに対応するソース行にキャ レットを移動した状態で,エディタ パネルがオープンします。
条件の編集	選択しているイベント ^{注 2} に対応した詳細設定ダイアログがオープンします。 トレース・イベントおよび組み合わせブレーク・イベントを選択した場合,組み 合わせ条件設定 ダイアログ【E1】【E20】【EZ Emulator】がオープンします。 Printf イベントを選択した場合は,アクション・イベント ダイアログをオープン します。
コメントの編集	選択しているイベントのコメントを編集モードにします。 すでにコメントが存在する場合は、その文字列のすべてを選択状態にします。

注 1.

トレース・イベント/タイマ計測イベント/ビルトイン・イベント(無条件トレース・イベント /Run-Break タイマ・イベント)以外のイベントが対象となります。

注 2. 次のイベントが対象となります。 - ハードウエア・ブレーク/実行系イベント/アクセス系イベント/タイマ計測イベント【E1】【E20】


出力 パネル

CS+が提供している各種コンポーネント(デバッグ・ツールを含む,設計ツール/ビルド・ツール/解析ツールなど)から出力されるメッセージの表示,または検索・置換ダイアログによる一括検索を行った際の結果,およびPrintf イベント(「2.16.1 printfを挿入する」参照)による出力結果の表示を行います。 メッセージは,出力元のツールごとに分類されたタブ上でそれぞれ個別に表示されます。

図 A.21 出力 パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[ファイル] メニュー (出力 パネル専用部分)]
- [[編集] メニュー (出力 パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [出力] を選択

[各エリアの説明]

(1) メッセージ・エリア 各ツールから出力されたメッセージ、検索結果、および Printf イベントによる出力結果を表示します。 検索結果(一括検索)の表示では、検索を行うごとに、以前のメッセージをクリアしたのち新しいメッセージを 表示します([すべてのメッセージ] タブを除く)。 なお、メッセージの表示色は、出力メッセージの種別により、次のように異なります(表示の際の文字色/背景)

なお,メッセージの表示色は,出力メッセージの種別により,次のように異なります(表示の際の文字色/背景 色はオプション ダイアログにおける[全般 - フォントと色]カテゴリの設定に依存)。

メッセージ種別	表示例	(デフォノ	レト)	説明
通常メッセージ	AaBbCc	文字色	黒	何らかの情報を通知する際に表示されます。
		背景色	白	
警告メッセージ	AaBbCc	文字色	青	操作に対して、何らかの警告を通知する際に表示
		背景色	標準色	されます。
エラー・メッセージ	AaBbCc	文字色	赤	致命的なエラー,または操作ミスにより実行が不
		背景色	薄グレー	可能な場合に表示されます。

このエリアは、次の機能を備えています。

(a) タグ・ジャンプ

出力されたメッセージをダブルクリック,またはメッセージにキャレットを移動したのち [Enter] キーを押 下することにより,エディタ パネルをオープンして該当ファイルの該当行番号を表示します。 これにより,ビルド時に出力されたエラー・メッセージなどから,ソース・ファイルの該当するエラー行へ ジャンプすることができます。

- (b) ヘルプの表示 警告メッセージ,またはエラー・メッセージを表示している行にキャレットがある状態で、コンテキスト・メニューの[メッセージに関するヘルプ]を選択するか、または[F1]キーを押下することにより、その行のメッセージに関するヘルプを表示します。
- (c) ログの保存

[ファイル] メニュー→ [名前を付けて出力-タブ名を保存...]を選択することにより,名前を付けて保存 ダ イアログをオープンし,現在選択しているタブ上に表示されている全内容をテキスト・ファイル(*.txt)に保 存することができます(非選択状態のタブ上のメッセージは保存の対象となりません)。

(d) 表示の拡大/縮小

出力パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・ リストを変更することにより、出力パネルの表示を拡大/縮小することができます。 また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、出力パネルの表示を拡大し、表示が大き く見やすくなります (最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、出力パネルの表示を縮小し、表示が小さくなります(最小 25%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

(2) タブ選択エリア メッセージの出力元を示すタブを選択します。

デバッグ・ツールでは、次のタブを使用します。

タブ名	説明
すべてのメッセージ	CS+ が提供している全コンポーネント(デバッグ・ツールを含む,設計 ツール/ビルド・ツール/解析ツールなど)から出力されるメッセージを 表示します(ラピッド・ビルドの実行によるメッセージを除く)。
デバッグ・ツール	CS+ が提供している各種コンポーネント(デバッグ・ツールを含む,設計 ツール/ビルド・ツール/解析ツールなど)から出力されるメッセージの うち,デバッグ・ツールが出力するメッセージを表示します。
検索・置換	検索・置換 ダイアログによる一括検索結果を表示します。

注意新たなメッセージが非選択状態のタブ上に出力されても、自動的なタブの表示切り替えは行いません。この場合、タブ名の先頭に "*" マークが付加し、新たなメッセージが出力されていることを示します。

[[ファイル] メニュー (出力 パネル専用部分)]

出力パネル専用の [ファイル] メニューは次のとおりです(その他の項目は共通)。 ただし、プログラム実行中はすべて無効となります。

出力 - <i>タブ名</i> を保存	現在選択しているタブ上に表示されている内容を,前回保存したテキスト・ファ イル(*.txt)に保存します(「(c)ログの保存」参照)。 なお,起動後に初めてこの項目を選択した場合は,[名前を付けてタブ名を保存 …]の選択と同等の動作となります。 ただし,ビルド実行中は無効となります。
名前を付けて出力 - <i>タブ名</i> を保 存	現在選択しているタブ上に表示されている内容を,指定したテキスト・ファイル (*.txt)に保存するために,名前を付けて保存 ダイアログをオープンします (「(c) ログの保存」参照)。

[[編集] メニュー (出力 パネル専用部分)]

出力パネル専用の[編集]メニューは次のとおりです(その他の項目はすべて無効)。

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にします。
検索	検索・置換 ダイアログを[クイック検索]タブが選択状態でオープンします。
置換	検索・置換 ダイアログを[一括置換]タブが選択状態でオープンします。

[コンテキスト・メニュー]

コピー	選択している文字列をクリップ・ボードにコピーします。
すべて選択	現在選択しているタブ上に表示されているすべてのメッセージを選択状態にしま す。
クリア	現在選択しているタブ上に表示されているすべてのメッセージを消去します。
タグ・ジャンプ	エディタ パネルをオープンし,キャレット位置のメッセージに該当するファイル の該当行番号にジャンプします。
検索の中止	現在実行中の検索を中止します。 ただし,検索を実行していない場合は無効となります。
メッセージに関するヘルプ	現在のキャレット位置のメッセージに関するヘルプを表示します。 ただし, 警告メッセージ/エラー・メッセージのみが対象となります。



デバッグ・コンソール パネル

コンソールとプログラム間で、データのやり取りを行います。標準ライブラリ関数を実装したプログラムを実行する ことにより、実現しています。 なお、このパネルは、デバッグ・ツールと接続時のみオープンすることができます。

図 A.22 デバッグ・コンソール パネル



ここでは、以下の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [[編集] メニュー (デバッグ・コンソール パネル専用部分)]
- [コンテキスト・メニュー]

[オープン方法]

- [表示] メニュー→ [デバッグ・コンソール] の選択

[各エリアの説明]

- (1) 入出カエリア プログラム内に実装した標準ライブラリ関数、例えば、scanf 関数によるキーボード入力データの読み込み、 printf 関数によるデータの出力を行うことができます。 また、パネル上での COM ポートの指定により、対象の COM ポートへプログラムの標準入出力をリダイレクト できます。 この機能を使用するには、プログラム内にデバッグ・ツールが提供する低水準インタフェース・ルーチンの実装 が必要です(「2.20 デバッグ・コンソールの使用」参照)。
 備考 【シミュレータ】 シミュレータが提供する入出力機能の詳細については、「B.入出力機能」を参照してください。
 - (a) 表示の拡大/縮小

デバッグ・コンソール パネルにフォーカスがある状態で、メイン・ウインドウのツールバー上にある拡大率のドロップダウン・リストを変更することにより、デバッグ・コンソール パネルの表示を拡大/縮小することができます。

また、[Ctrl] キーとマウス・ホイールを組み合わせることでも変更可能です。

- [Ctrl] キーを押下しながらマウス・ホイールを前方に動かすと、デバッグ・コンソール パネルの表示を拡大し、表示が大きく見やすくなります(最大 300%)。
- [Ctrl] キーを押下しながらマウス・ホイールを後方に動かすと、デバッグ・コンソール パネルの表示を縮小し、表示が小さくなります(最小 25%)。

なお,拡大率を変更した状態のパネルをクローズした場合,拡大率は保持されます(再度パネルをオープンする際に,変更した拡大率でオープンします)。

[[編集] メニュー(デバッグ・コンソール パネル専用部分)]

デバッグ・コンソール パネル専用の [編集] メニューは以下のとおりです(その他の項目は共通です)。

コピー	選択している文字列をクリップ・ボードにコピーします。
貼り付け	クリップ・ボードの内容をキャレット位置に挿入します。
すべて選択	このパネルに表示しているすべての文字列を選択状態にします。

[コンテキスト・メニュー]

コピー	選択している文字列をクリップ・ボードにコピーします。
貼り付け	クリップ・ボードの内容をキャレット位置に挿入します。
クリア	このパネルの表示をクリアします。
デバッグ・コンソール有効 / 無 効	デバッグ・コンソール機能の有効(デフォルト)/無効を選択します(トグル)。 なお、デバッグ・コンソール機能を無効化にすると、パネルの背景色がグレー になります。
COM ポートの設定	マイクロコントローラからの通信をリダイレクトするホスト・マシン上の COM ポートを設定するために, COM ポート設定 ダイアログをオープンします。
ログ・ファイル指定	このパネルに表示している内容を指定したログ・ファイル(*.log)に保存する ために, ログ・ファイルの指定 ダイアログをオープンします。 ロギングを開始します。
 ロギング開始 / 停止	ロギングの開始(デフォルト)/停止を選択します(トグル)。 ただし、ログ・ファイルを指定していない場合、この項目は無効になります。
ローカル・エコー・バック有効 / 無効	ローカル・エコー・バックの有効(デフォルト)/無効を選択します(トグ ル)。 ローカル・エコー・バックが無効の場合,入力したデータはこの パネル上に出 カしません。



メモリ・マッピング ダイアログ

メモリ・マッピングの設定をメモリ種別ごとに行います。



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]



[オープン方法]

- プロパティ パネルの [デバッグ・ツール設定] タブにおいて, [メモリ] カテゴリ内 [メモリ・マッピング] プロ パティを選択することにより表示される [...] ボタンをクリック

- **注意 1.** 【シミュレータ】 プログラム実行中は、このダイアログをオープンすることはできません。
- 注意 2. 【E1】【E20】【EZ Emulator】 デバッグ・ツールに接続後、このダイアログをオープンすることはできません。

[各エリアの説明]

- (1) 追加メモリ・マッピング指定エリア 新たに追加するメモリ・マッピングの情報を指定します。
 - (a) [メモリ種別] 追加するメモリ・マッピングのメモリ種別を次のドロ

追加するメモリ・マッピングのメモリ種別を次のドロップダウン・リストより選択します(デフォルトで選択 される項目は使用するデバッグ・ツールに依存)。

エミュレーション ROM 領域	エミュレーション ROM 領域を追加します。
【シミュレータ】	シミュレータ代替 ROM を使用します。
エミュレーション RAM 領域	エミュレーション RAM 領域を追加します。
【シミュレータ】	シミュレータ代替 RAM を使用します。
I/O 保護領域 【E1】【E20】【EZ Emulator】	I/O 保護領域を追加します。 外部領域のうち, デバッガからリードされないようにするアドレス範 囲を指定してください。

- **注意 1.** 【シミュレータ】 追加できるメモリ・マッピングは 16 バイト単位です。16 バイト境界以外で設定した場合は, [OK] ボタンを押下後に設定した領域を含む 16 バイト境界の領域に補正します。
- **注意 2.** 【シミュレータ】 追加されたメモリ・マッピングへのアクセスは、読み出し、書き込みともに1サイクルで動作 します。
- **注意 3.** 【E1】【E20】【EZ Emulator】 追加される I/O 保護領域は, [OK] ボタンを押下後に I/O 保護領域が含まれる外部領域のアクセ ス幅で補正されます。

(b) [アドレス範囲]

追加するメモリ・マッピングの開始アドレスと終了アドレスを指定します。それぞれのテキスト・ボックス に、16 進数を直接入力します。 エミュレータの場合、I/O 保護領域は外部領域と重なる領域にのみ追加できます。また、各外部領域の境界を

またぐ指定はできません。 シミュレータの場合、次のメモリ種別と重なる領域にメモリ・マッピングを追加することはできません(この エリアの[追加]ボタンをクリックした際に、メッセージを表示します)。

- [内蔵 ROM 領域]
- [内蔵 RAM 領域]
- [IO レジスタ領域]
- (c) [アクセス幅] アクセス幅は設定できません。
- (d) ボタン

ボタン	機能
追加	このエリアで指定した内容をメモリ・マッピングに追加します。 追加されたメモリ・マッピングは、[マッピング済みメモリ一覧] エリアに表示されま す。なお、[OK] ボタンを押下するまでは、変更内容の設定は行われません。

(2) [マッピング済みメモリー覧] エリア

(a) 一覧の表示

追加メモリ・マッピング指定エリアで追加したメモリ・マッピングと、マイクロコントローラ内のメモリ・ マッピングの情報を表示します。このエリアを編集することはできません。

メモリ種別	次のメモリ種別を表示します。
	- 内蔵 ROM 領域
	- 内蔵 RAM 領域
	- I/O レジスタ領域
	- 外部領域(CS7/CS6//CS0)【E1】【E20】【EZ Emulator】
	- その他のメモリ領域【E1】【E20】【EZ Emulator】
	- 予約領域【E1】【E20】【EZ Emulator】
	- I/O 保護領域【E1】【E20】【EZ Emulator】
	- エミュレーション ROM 領域【シミュレータ】
	- エミュレーション RAM 領域【シミュレータ】
	- ノン・マップ領域【シミュレータ】
アドレス範囲	アドレス範囲を< <i>開始アドレス</i> > - < <i>終了アドレス</i> >で表示します。 "0x" を付与した 16 進数表示固定です。
サイズ	サイズを 10 進数で表示します(単位:バイト /K バイト)。
アクセス幅	アクセス幅を表示します(単位:ビット) ^注 。

注 シミュレータではアクセス幅は未サポートのため、固定値(8ビット)を表示します。 なお、アクセス幅の設定値は、シミュレーションの実行時間には影響しません。

(b) ボタン

ボタン	機能
削除	このエリアで選択しているメモリ・マッピングを削除します。 エミュレータの場合、削除できるメモリ領域は I/O 保護領域のみです。 シミュレータの場合、削除できるメモリ領域は、エミュレーション ROM 領域/エ ミュレーション RAM 領域のいずれかです(マイクロコントローラ内のメモリ・マッ ピングを削除することはできません)。

ボタン	機能
ОК	現在設定されているメモリ・マッピングをデバッグ・ツールに設定し,このダイ アログをクローズします。
キャンセル	メモリ・マッピングの変更を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

ダウンロード・ファイル ダイアログ

ダウンロードする際のファイルの選択,およびダウンロード条件の設定を行います(「2.5 ダウンロード/アップロード」参照)。

プロジェクト(メイン・プロジェクト/サブプロジェクト)でビルド対象に指定しているファイルは、自動的にダウンロードの対象ファイルとして登録されます(削除不可)。

図 A.25 ダウンロード・ファイル ダイアログ

	ダウンロード・ファイル				
Γ	ダウンロード・ファイル一覧(E):		ダウンロード・ファイルのプロパティ(P):		
(1) –	Tutorial.ebs	10	 ダウンロード・ファイル情報 ファイルの種類 オブジェクトをダウンロードする シンボルをダウンロードする PIC/PIDオフセットを指定する 入力補完候能用の情報を生成する 	DefaultBuild#TutoriaLabs ロード・モジュール・ファイル はい はい いいえ はい	- (2)
[機能ボタン]-	isto(a) if (R*(E)	,	ファイル ダウンロードするファイルを指定します。 OK	キャンセル へルプ(H)	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- プロパティ パネルの [ダウンロード・ファイル設定] タブにおいて, [ダウンロード] カテゴリ内 [ダウンロード するファイル] プロパティを選択することにより表示される [...] ボタンをクリック
- 注意 プログラム実行中は、このダイアログをオープンすることはできません。

[各エリアの説明]

- (1) [ダウンロード・ファイル一覧] エリア
 - (a) 一覧の表示
 ダウンロードするファイル名の一覧を表示します。デフォルトで、プロジェクト(メイン・プロジェクト/サ ブプロジェクト)においてビルド対象に指定しているファイル名を表示します(削除不可)。
 ここでの表示順序が、ダウンロードの際の実行順序となります。
 新規にダウンロード・ファイルを追加する場合は、このエリア内の[追加]ボタンをクリックし、[ダウン ロード・ファイルのプロパティ]エリアにおいて、追加するファイルのダウンロード条件を指定することにより行います。

(b) ボタン

ボタン	機能
Î	選択しているファイルを1行上に移動します。 ただし,最上部のファイル,またはプロジェクトのビルド対象に指定しているファイル を選択している場合は無効となります。
Ţ	選択しているファイルを1行下に移動します。 ただし、最下部のファイル、またはプロジェクトのビルド対象に指定しているファイル を選択している場合は無効となります。
追加	ー覧に空欄の項目("-")を1つ追加し,選択状態にします。 [ダウンロード・ファイルのプロパティ]エリアにおいて,追加するファイルのダウン ロード条件を指定してください。 ただし,すでに20個以上のファイルが登録されている場合は無効となります。
削除	選択しているファイルを一覧から削除します。 ただし、プロジェクトのビルド対象に指定しているファイルは削除することはできません。

備考 1. ファイル名にマウス・カーソルを合わせることにより、対象ファイルのパス情報をポップアップ 表示します。

備考 2. ファイル名をマウスでドラッグすることにより、一覧内の表示順序を変更することができます。 ただし、プロジェクトでビルド対象に指定しているファイルの表示順序を変更することはできま せん。

- (2) [ダウンロード・ファイルのプロパティ] エリア
- (a) [ダウンロード・ファイル情報]
 [ダウンロード・ファイル一覧]エリアで選択しているファイルに対して、ダウンロード条件の表示/設定変更を行います。
 また、[追加]ボタンにより、新規にダウンロード・ファイルを追加する場合は、ここで追加ファイルのダウンロード条件を指定します。

ファイル	ダウンロード	するファイルを指定します。
	デフォルト	<i>ファイル名</i> (ただし,新規追加の場合は空欄)
	変更方法	キーボードからの直接入力,またはこの項目を選択すると欄内右端に表 示される […] ボタン ^{注 1} のクリックによりオープンするダウンロード するファイルを選択 ダイアログによる指定
	指定可能値	「表 2.2 ダウンロード可能なファイル形式」参照 最大指定文字数:259文字
ファイルの種類	ダウンロード	するファイルのファイル形式を指定します。
	デフォルト	ロード・モジュール・ファイル
	変更方法	ドロップダウン・リストによる選択
	指定可能值	次のいずれか
		- ロード・モジュール・ファイル
		- ヘキサ・ファイル
		- S レコード・ファイル
		- バイナリ・データ・ファイル
オフセット	この項目は, ド・ファイル アドレスから	ダウンロードするファイルがヘキサ・フォーマット,およびSレコー の場合のみ表示されます。指定したファイルのダウンロードを開始する のオフセット値を指定します。
	デフォルト	0
	変更方法	キーボードからの直接入力
	指定可能值	0x0~0xFFFFFFFの16進数



開始アドレス	この項目は, み表示されま	ダウンロードするファイルがバイナリ・データ・フォーマットの場合の ます。指定したファイルをダウンロードする開始アドレスを指定します。		
	デフォルト	0		
	変更方法	キーボードからの直接入力		
	指定可能值	0x0 ~ 0xFFFFFFFF の 16	0x0 ~ 0xFFFFFFF の 16 進数	
オブジェクトをダ ウンロードする	この項目は, のみ表示され かを指定しま	ダウンロードするファイル ,ます。指定したファイルか ;す。	がロード・モジュール・フォーマットの場合 らオブジェクト情報をダウンロードするか否	
	デフォルト	はい		
	変更方法	ドロップダウン・リスト	こよる選択	
	指定可能值	はい	オブジェクト情報をダウンロードします。	
		いいえ	オブジェクト情報をダウンロードしません。	
シンボルをダウン	この項目は、	ダウンロードするファイル	がロード・モジュール・フォーマットの場合	
ロートする	のみ表示され 指定したファ	<i>、</i> ます。 ゚イルからシンボル情報をダ	、 ウンロードするか否かを指定します ^{注 2} 。	
	デフォルト	はい		
	変更方法	ドロップダウン・リスト	こよる選択	
	指定可能值	はい	シンボル情報をダウンロードします。	
		いいえ	シンボル情報をダウンロードしません。	
PIC/PID オフセッ トを指定する	ダウンロード よび PID(Po 更するか否か フセット ", '	ダウンロードするロード・モジュールの PIC(Position Independent Code)領域,お よび PID(Position Independent Data)領域の位置をロード・モジュール作成時から変 更するか否かを指定します。この項目を " はい " に切り替えると,サブ項目に "PIC オ フセット ", "PID オフセット " 項目を表示します。		
	デフォルト	'オルト いいえ		
	変更方法	ドロップダウン・リスト	こよる選択	
	指定可能值	はい	PIC/PID オフセットを指定します ^{注 3} 。	
		いいえ	PIC/PID オフセットを指定しません。	
PIC オフセット	ット ロード・モジュール作成時のアドレスからのオフセット値を入力します。 例えば、プログラム・セクションの開始アドレスが 0x1000 番地からだった場合 項目に "1000" を入力すると、対象のセクションは 0x2000 番地にダウンロードさ す。		らのオフセット値を入力します。 アドレスが 0x1000 番地からだった場合に本 フションは 0x2000 番地にダウンロードされま	
	デフォルト	0		
	変更方法	キーボードからの直接入ス	b	
	指定可能值	0x0 ~ 0xFFFFFFFFの 16	3 進数	
PID オフセット	ロード・モジ	ュール作成時に指定した P	ID レジスタへ設定するオフセット値を入力し	
	ます。 例えば, ロー 項目に "200"	ロード・モジュールの実行時に PID レジスタに 0x200 を設定する場合は,本 200" を入力してください。		
	デフォルト	0		
	変更方法	キーボードからの直接入ス	b	
	指定可能值	0x0 ~ 0xFFFFFFF の 16	5 進数	



入力補完機能用の 情報を生成する	 この項目は、ダウンロードするファイルがロード・モジュール・フォーマットの:のみ表示されます。 選択されたファイルからシンボル情報をダウンロードする際に、シンボル名の入 完機能用の情報を生成するか否かを指定します。 ウォッチパネルやメモリパネルなどでシンボル名の入力補完機能を使う場合は"(を選択してください。なお、この情報はシンボルのダウンロード中に生成するため ダウンロードに時間が掛かる場合があります。 デフォルト はい 		がロード・モジュール・フォーマットの場合 ダウンロードする際に、シンボル名の入力補 します。 ンボル名の入力補完機能を使う場合は"はい" シンボルのダウンロード中に生成するため、 ます。
	変更方法	ドロップダウン・リストロ	こよる選択
	指定可能值	はい	入力補完機能用の情報を生成します。
		いいえ	入力補完機能用の情報を生成しません。

注 1. [ダウンロード・ファイル一覧] エリアにおいて、プロジェクトのビルド対象のファイルを選択 している場合、またはプログラム実行中は、[...] ボタンは表示されません。

注 2. シンボル情報をダウンロードしない場合、ソース・レベル・デバッグを行うことはできません。

注 3. PIC/PID 機能(「2.7 PIC/PID 機能の使用」参照)を使って作成していないロード・モジュール に対して " はい " を選択した時のデバッグ動作は保証できません。

ボタン	機能
ОК	ダウンロード・ファイルの設定を終了し,このダイアログをクローズします。
キャンセル	ダウンロード・ファイルの変更を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



アクション・イベント ダイアログ

アクション・イベントの設定を行います(「2.16 プログラム内へのアクションの設定」参照)。 なお、このダイアログは、デバッグ・ツールと接続時のみオープンすることができます。

注意 アクション・イベントの設定に関しては(有効イベント数の制限など),(「2.17.7 イベント設定に関する留意事項」も参照してください。

図 A.26 アクション・イベント ダイアログ

	アクション・イベント
(1)-	Printf イベント 状態保存イベント 割り込みイベント設定
	出力文字列(Q): 入力例) サンプル:
	がたまたの 3 th ///D and http://www.inter-
	2000 TVUV: ACTINU ass, bbb, ccc
	(ここに変数式を入力してくたさい)
	アドレス(<u>A</u>):
	main 💌
(2)	
	出力パネルでの表示例
	サンプル: aaa = 10, bbb = 20 ccc = 30
[機能ボタン]-	OK +++>tel Alt (H)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにおいて, アクション・イベントを設定したい行にキャレットを移動したのち, コンテキスト・メ ニュー→ [アクション・イベントの登録 ...]を選択
- 逆アセンブル パネルにおいて,アクション・イベントを設定したいアドレスにキャレットを移動したのち,コンテ キスト・メニュー→ [アクション・イベントの登録 ...]を選択
- イベント パネルにおいて,アクション・イベントを選択したのち,コンテキストメニュー→ [条件の編集 ...]を選 択

[各エリアの説明]

- (1) タブ選択エリア タブを選択することにより、登録するアクション・イベントが切り替わります。 このダイアログには、次のタブが存在します。
 - [Printf イベント] タブ
 - [状態保存イベント] タブ

- [割り込みイベント設定] タブ【シミュレータ】

注意 コンテキスト・メニューの [条件の編集 ...]の選択によりこのダイアログをオープンした場合, このエリアは非表示となります。

(2) イベント条件設定エリア
 アクション・イベントの詳細条件を設定します。
 設定方法についての詳細は、該当するタブの項を参照してください。

ボタン	機能
ОК	アクション・イベントの設定を終了し, 指定したアクション・イベントを指定し た位置に設定します。
キャンセル	アクション・イベントの設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



[Printf イベント] タブ

アクション・イベントとして、Printf イベントの設定を行います(「2.16.1 printf を挿入する」参照)。 Printf イベントとは、プログラムの実行を指定した箇所で一瞬停止させ、ソフトウエア処理によりコマンド(printf) を実行させる機能です。Printf イベントを設定すると、このイベントを設定した箇所の命令実行直前にプログラムが一 瞬停止し、このダイアログで指定した変数式の値を出力パネルに出力します。

図 A.27 アクション・イベント ダイアログ : [Printf イベント] タブ

(アクション・イベント	×
	Printf イベント 状態保存イベント 割り込みイベント設定	
	出力文字列(Q): 入力例) サンブル:	
(1)-		
(2)	変数式(V): 入力例D aaa, bbb, ccc	
(²)	(ここに変動式を入力してください)	
(3)	アドレス(<u>A</u>):	_
(3)	main	-
	出力 パネル での表示例0	
	$\frac{1}{2}\sqrt{2}$ // $\frac{1}{2}$ // $\frac{1}{2}$ // $\frac{1}{2}$ // $\frac{1}{2}$ // $\frac{1}{2}$ // $\frac{1}{2}$	
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
		Ξ
[機能ボタン]-	OK AVYTA ANT	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにおいて, Printf イベントを設定したい行にキャレットを移動したのち, コンテキスト・メニュー → [アクション・イベントの登録 ...] を選択
- 逆アセンブル パネルにおいて, Printf イベントを設定したいアドレスにキャレットを移動したのち, コンテキス ト・メニュー→ [アクション・イベントの登録 ...] を選択
- イベント パネルにおいて, Printf イベントを選択したのち, コンテキストメニュー→ [条件の編集 ...]を選択

[各エリアの説明]

- (1) [出力文字列] エリア
 出力パネルに出力する際に付与する文字列をキーボードより直接入力で指定します(最大指定文字数:1024文字)。
 なお、出力する文字列は、1行分のみ入力可能です(空白可)。
- (2) [変数式] エリア
 Printf イベントの対象となる変数式を指定します。
 変数式は、テキスト・ボックスに直接入力で指定します(最大指定文字数:1024 文字)。
 "," で区切ることにより、1 つの Printf イベントとして 10 個までの変数式を指定することができます。

エディタ パネル/逆アセンブル パネルにおいて、変数式を選択した状態でこのダイアログをオープンした場合では、選択している変数式がデフォルトで表示されます。 なお、変数式として指定できる基本入力形式と、その際に Printf イベントとして出力される値は次のとおりで す。

変数式	出力される値
C/C++ 言語変数名 ^{注 1}	C/C++ 言語の変数の値
<u>変数式[変数式]</u>	配列の要素値
<i>変数式</i> .メンバ名 ^{注2}	構造体/共用体/クラスのメンバ値
<i>変数式-</i> > メンバ名 ^{注 2}	ポインタの指し示す構造体/共用体/クラスのメンバ値
変数式 .* キャスト式	メンバ変数のポインタの値
変数式 ->* キャスト式	メンバ変数のポインタの値
* 変数式	ポインタの変数の値
& 変数式	配置アドレス
(型名)変数式	指定された型へキャストした値
CPU レジスタ名	CPU レジスタの値
I/O レジスタ名	I/O レジスタの値
ラベル名 ^{注 3} / EQU シンボル名 ^{注 3} / [即値]	ラベルの値/ EQU シンボルの値/即値アドレスの値

主 Δ 15	亦物ポレ出わされる値の	期区(Printf イベント)
衣 A.15	2 奴式と山力される 胆の	判除(PШШ イ ハノト)

注 1. C89, C99, C++ 言語の変数を表します。

- 注 2. 基底クラスのメンバ変数を指定する場合、メンバ名の前にスコープを指定してください (例: variable.BaseClass::member)。
- 注 3. ラベル名または EQU シンボル名に "\$" が含まれている場合,名前を "{}" で囲んでください (例: {\$Label})。 虚数の値には,大文字の "I"を掛けてください(例:1.0+2.0*I)。"I" はキーワードとなるため, CPU レジスタの "I" を指定する場合は ":REG" を付加してください(例:1:REG)。
- 備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより、現在のキャレット位置のシ ンボル名を補完することができます。(「2.21.2 シンボル名の入力補完機能」参照)

(3) [アドレス] エリア
Printf イベントを設定するアドレスを指定します。
テキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字),またはドロップダウン・リ
ストにより入力履歴項目(最大履歴個数:10個)を選択します。デフォルトで,現在の指定位置のアドレスを
表示します。
 備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシ
ンボル名を補完することができます。(「2.21.2 シンボル名の入力補完機能」参照)

なお、出力パネル上における、Printfイベントによる出力結果のフォーマットは次のとおりです。

図 A.28 Printf イベントの出力結果フォーマット

指定された文字列 変数式1=値1,変数式2=値2,変数式3=値3,...

指定された文字列	[出力文字列] で指定した文字列
変数式1~10	[変数式]で指定した文字列

值1~10	* <i>変数式1 ~ 10</i> "に対する変数値 値は変数の型に応じた表示形式(「素 A 11 ウォッチ式の表示形式(デフォルト)」参
	照)で表示します(指定された変数式を取得できない場合は、"?"を表示)。 また、"()"内に 16 進数値も併記します(表示不能の場合は"-"を表示)。

ボタン	機能
ОК	Printf イベントの設定を終了し,ここで指定した Printf イベントをエディタ パネ ル/逆アセンブル パネル上のキャレット位置の行/アドレスに設定します。
キャンセル	Printf イベントの設定を無効とし,このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



[状態保存イベント] タブ

アクション・イベント発生時のデバッグ・ツールの状態の保存/復帰の設定を行います。 なお、復元されるデータは読み書き可能なメモリとレジスタの値に限られます。

図 A.29 アクション・イベント ダイアログ:[状態保存イベント] タブ

	アクション・イベント		
	Printf イベント 状態保存イベント 割り	込みイベント設定	
	状態を保存または復帰する場所を選択してください。		
(1)	デバッグ・ツールの状態保存1(1)	◎ デバッグ・ツールの状態保存2(2)	
(')	◎ デバッグ・ツールの状態保存3(3)	◎ デバッグ・ツールの状態保存4(4)	
(2)	デバッグ・ツールの状態復帰1(B)	○ デバッグ・ツールの状態(猿帰2(E))	
(2)-	◎ デバッグ・ツールの状態復帰3(S)	◎ デバッグ・ツールの状態復帰4①	
(3)	アドレス(<u>A</u>):		
(3)	(アドレスを入力してください)	•	
[機能ボタン] -		OK キャンセル ヘルプ(出)	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにおいて,状態保存イベントを設定したい行にキャレットを移動したのち, コンテキスト・メ ニュー→ [アクション・イベントの登録 ...]を選択
- 逆アセンブル パネルにおいて,状態保存イベントを設定したいアドレスにキャレットを移動したのち,コンテキス ト・メニュー→ [アクション・イベントの登録...]を選択
- イベント パネルにおいて,状態保存イベントを選択したのち,コンテキストメニュー→ [条件の編集...]を選択

[各エリアの説明]

- (1) [デバッグ・ツールの状態保存 n] エリア アクション・イベント発生時,デバッグ・ツールの状態を n 番目のデータとしてファイルに保存します。
- (2) [デバッグ・ツールの状態復帰 n] エリア アクション・イベント発生時,デバッグ・ツールの状態を n 番目のデータ・ファイルから復帰します。
- (3) [アドレス] エリア 状態保存イベントを設定するアドレスを指定します。 テキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024 文字),またはドロップダウン・リ ストにより入力履歴項目(最大履歴個数:10 個)を選択します。デフォルトで,現在の指定位置のアドレスを 表示します。

ボタン	機能
ОК	状態保存イベントの設定を終了し,ここで指定した状態保存イベントをエディタ パネル/逆アセンブル パネル上のキャレット位置の行/アドレスに設定します。
キャンセル	状態保存イベントの設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



[割り込みイベント設定] タブ【シミュレータ】

割り込みイベントの設定を行います(「2.16.2 割り込みイベントを挿入する【シミュレータ】」参照)。 割り込みイベントとは、プログラムの実行を指定した箇所で割り込み要求を発生させる機能です。割り込みイベント を設定すると、このイベントを設定した箇所の命令実行直前に割り込み要求が発生し、CPU が割り込み要求を受け付け ると割り込み例外処理を実行します。

図 A.30 アクション・イベント ダイアログ:[割り込みイベント設定]タブ

	アクション・イベント
	Printf イベント 状態保存イベント 割り込みイベント設定
(1)-	割り込みべりタ
(2)	優先順位:
L	
[機能ボタン] –	OK キャンセル ヘルプ(出)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- エディタ パネルにおいて、割り込みイベントを設定したい行にキャレットを移動したのち、コンテキスト・メ ニュー→ [アクション・イベントの登録 ...]を選択
- 逆アセンブル パネルにおいて、割り込みイベントを設定したいアドレスにキャレットを移動したのち、コンテキス ト・メニュー→ [アクション・イベントの登録…]を選択
- イベント パネルにおいて,割り込みイベントを選択したのち,コンテキストメニュー→ [条件の編集...]を選択

[各エリアの説明]

- (1) [割り込みベクタ]エリア 割り込みベクタ番号を指定します。
 0~255の範囲で直接入力により指定してください。
- (2) [優先順位]エリア
 優先順位を指定します。
 -【RX610 グループ】
 - 1~8の範囲で直接入力により指定してください。

8を指定した場合は、高速割り込みとして動作します。

-【RX610 グループ以外】 1~16の範囲で直接入力により指定してください。 16を指定した場合は、高速割り込みとして動作します。

ボタン	機能
ОК	割り込みイベントの設定を終了し、ここで指定した割り込みイベントをエディタ パネル/逆アセンブル パネル上のキャレット位置の行/アドレスに設定します。
キャンセル	割り込みイベントの設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



表示桁数設定 ダイアログ

メモリパネルにおいて、メモリ値の表示桁数の設定を行います。

図 A.31 表示桁数設定 ダイアログ

(表示桁数設定		
(1)-	表示桁数(<u>C</u>):	16]
[機能ボタン] -		OK キャンセル ヘルプ(日)]

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて, コンテキスト・メニューの [表示] → [表示桁数を設定...]を選択

[各エリアの説明]

 (1) [表示桁数] エリア 表示する桁数を 10 進数で直接入力により指定します。
 指定可能な値の範囲は、現在のメモリ パネルにおける [サイズ表記]の設定により、次のように異なります。

サイズ表記	指定可能な範囲		
4ビット	2~512注		
1バイト	1 ~ 256		
2バイト	1 ~ 128		
4バイト	1 ~ 64		
8バイト	1 ~ 32		

注 偶数値でのみ指定できます(奇数値が入力された場合,1つ大きな値に変更されます)。

[機能ボタン]

ボタン	機能
ОК	指定した桁数でメモリ値を表示します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて、コンテキスト・メニューの [表示] → [表示アドレス・オフセット値を設定 ...]を選択

[各エリアの説明]

- (1) [表示アドレス・オフセット値]エリア アドレス表示のオフセット値を16進数で直接入力により指定します。 指定可能な値の範囲は、現在のメモリパネルにおいて1行に表示されているメモリのバイト数により、次のように異なります。
 - 指定可能な範囲: 0x0 ~ ([サイズ表記]の設定×表示桁数)-1

例

[サイズ表記]:1 バイト/表示桁数:16 桁の場合

オフセット値	アドレス・エリアの表示内容		
0x0(デフォルト)	0000 0010 0020		
0x1	0001 0011 0021		
0x2	0002 0012 0022		

[機能ボタン]

ボタン	機能
ОК	指定したオフセット値でメモリのアドレス表示を行います。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。

メモリ初期化 ダイアログ

メモリ値の初期化を行います(「2.11.1.6 メモリの内容を一括して変更(初期化)する」参照)。 指定したアドレス範囲のメモリ領域に,指定した初期化データのパターンを繰り返し書き込みます。

図 A.33 ン	33 メモリ初期化 ダイアログ			
		メモリ初期化		
(1)-	$\left\{ \right\}$	開始アドレス/シンボル(S):	終了アドレス/シンボル(E): (終了アドレス/シンボルを入力 回	
(2) -		初期化データ(): ■■■ (初期化データを16進数で入力してくださ	い。スペース区切りで複数指 🝺 📼	
[機能ボタン] ⁻	-[ОК	**>セル ヘルプ(H)	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて、コンテキスト・メニュー→ [初期化 ...]を選択

[各エリアの説明]

(1) 範囲指定エリア

メモリ値を初期化するアドレス範囲を [開始アドレス/シンボル] と [終了アドレス/シンボル] に指定しま す。それぞれのテキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024 文字),またはド ロップダウン・リストにより入力履歴項目(最大履歴個数:10 個)を選択します。 入力したアドレス式の計算結果を,それぞれ開始アドレス/終了アドレスとして扱います。 なお、マイクロコントローラのアドレス空間よりも大きいアドレス値を指定することはできません。

注意 エンディアンの異なる領域をまたいだアドレス範囲を指定することはできません。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

(2) [初期化データ] エリア

メモリに書き込む初期化データを指定します。 初期化データの指定は、16 進数の数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストにより入力履歴項目(最大履歴個数:10 個)を選択することにより行います。 初期化データを複数指定する場合は、1 個 4 バイト(8 文字)までのデータを最大 16 個まで、半角スペースで

区切り指定します。 個々の初期化データは、文字列終端より2文字単位で1バイトと解釈され、奇数文字数の場合は先頭1文字で1 バイトと解釈されます。

なお,バイト数が2バイト以上の場合は,次のようにプロジェクトのエンディアンに合わせたバイト列に変換 してターゲット・メモリへの書き込み処理を行います。

入力文字列(初期化データ)	書き込みイメージ(バイト単位)		
	リトル・エンディアン	ビック・エンディアン	
1	01	01	
0 12	00 12	00 12	
00 012 345	00 12 00 45 03	00 00 12 03 45	



入力文字列(初期化データ)	書き込みイメージ(バイト単位)	
	リトル・エンディアン	ビック・エンディアン
000 12 000345	00 00 12 45 03 00	00 00 12 00 03 45

ボタン	機能
ОК	指定したアドレス範囲のメモリ領域に,指定した初期化データのパターンを繰り 返し書き込みます(パターンの途中で終了アドレスに達した場合は書き込みを終 了します)。
キャンセル	メモリ値の初期化の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



メモリ検索 ダイアログ

メモリ値の検索を行います(「2.11.1.5 メモリの内容を検索する」参照)。 このダイアログをオープンする直前にメモリパネル上でキャレットが存在した、メモリ値エリア/文字列エリアのど ちらかが検索の対象となります。

	メモリ検索		
(1) –	検索するデータ(<u>D</u>):		
(2) —	検索する範囲(<u>R</u>)	アドレス範囲を指定する	
(3) —	アドレス(<u>A</u>):	0x0 💌	- 0xffffffff
[機能ボタン] -	1	前を検索(B) 次を検索(F)	キャンセル ヘルプ(H)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]
- [オープン方法]
 - メモリ パネルにおいて、コンテキスト・メニュー→ [検索 ...]を選択

[各エリアの説明]

- (1) 「検索するデータ]エリア 検索するデータを指定します。 テキスト・ボックスに直接入力するか(最大指定バイト数:256 バイト),またはドロップダウン・リストより 入力履歴項目を選択します(最大履歴数:10 個)。 検索の対象がメモリパネル上のメモリ値エリアの場合、そのエリアと同じ表示形式(表示進数/サイズ)で データを入力する必要があります。 また,検索の対象が文字列エリアの場合では、検索するデータとして、文字列を指定する必要があります。指定 した文字列は、そのエリアで表示しているエンコード形式でデータに変換され検索されます。 なお、このダイアログをオープンする直前にメモリ値を選択していた場合は、デフォルトでその値が表示されま す。
- (2) [検索する範囲] エリア

検索する範囲を次のドロップダウン・リストより選択します。

アドレス範囲を指定する	[アドレス] エリアで指定したアドレス範囲内で検索を行います。
メモリ・マッピング	選択したメモリ・マッピング範囲内で検索を行います。 このリスト項目は、メモリ・マッピングダイアログで表示しているメモリ・ マッピングを個々に表示します(ノン・マップ領域を除く)。 表示形式:<メモリ種別><アドレス範囲><サイズ>

(3) [アドレス] エリア

この項目は、[検索する範囲] エリアで [アドレス範囲を指定する] を選択した場合のみ有効となります。 メモリ値検索の対象となるアドレス範囲を"開始アドレス"と"終了アドレス"で指定します。それぞれのテキ スト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字),またはドロップダウン・リスト により入力履歴項目(最大履歴個数:10個)を選択します。 入力したアドレス式の計算結果を、それぞれ開始アドレス/終了アドレスとして扱います。 ただし、検索可能なアドレスの上限値は、プログラム空間の上限アドレス(0xFFFFFFF)です。 また、32ビットで表現できる値より大きいアドレス値を指定することはできません。

- 備考 1. このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- 備考 2. "開始アドレス"が空欄の場合は、"0x0"の指定として扱われます。
- 備考 3. "終了アドレス"が空欄の場合は、マイクロコントローラのアドレス空間の上限値の指定として扱われます。

ボタン	機能
前を検索	[検索する範囲] エリア/ [アドレス] エリアで指定した範囲内で、アドレスの 小さい方向に検索を行います。検索結果箇所をメモリパネル上で選択状態にし ます。 ただし、不正な値を指定している場合、またはプログラム実行中は、メッセージ を表示し、メモリ値の検索は行いません。 また、メモリパネルが非表示の場合、または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となりま す。
次を検索	[検索する範囲] エリア/ [アドレス] エリアで指定した範囲内で, アドレスの 大きい方向に検索を行います。検索結果箇所をメモリパネル上で選択状態にし ます。 ただし,不正な値を指定している場合,またはプログラム実行中は,メッセージ を表示し,メモリ値の検索は行いません。 また,メモリパネルが非表示の場合,または他のパネルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合,このボタンは無効となりま す。
キャンセル	メモリ値の検索の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



印刷アドレス範囲設定 ダイアログ

逆アセンブル パネルの内容を印刷する際に、対象となるアドレス範囲の指定を行います。

図 A.35 印刷アドレス範囲設定 ダイアログ

(印刷アドレス範囲設定
_	印刷するアドレスの指定方法を選択してください。
(1) —	 現在の表示範囲(D) 現在の選択範囲(C) 第定した範囲(E) 開始アドレス(S): (開始アドレスを入力してくだ 画 - (終了アドレスを入力してくだ 画 -
[機能ボタン]	OK キャンセル ヘルプ(ビ

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブル パネルにおいて, [ファイル] メニュー→ [印刷 ...] を選択

[各エリアの説明]

- (1) 範囲指定エリア印刷する範囲を指定するために、次のオプション・ボタンのいずれか1つを選択します。
 - (a) [現在の表示範囲](デフォルト) 逆アセンブル パネルで現在表示している範囲のみを印刷します。
 - (b) [現在の選択範囲] 逆アセンブル パネルで現在選択している範囲のみを印刷します。 ただし、逆アセンブル パネルにおいて、何も選択していない場合は無効となります。
 - (c) [指定した範囲] 印刷の対象となるアドレス範囲を[開始アドレス]と[終了アドレス]で指定します。 それぞれのテキスト・ボックスにアドレス式を直接入力するか(最大指定文字数:1024文字),またはドロッ プダウン・リストにより入力履歴項目(最大履歴個数:10個)を選択します。

[機能ボタン]

ボタン	機能	
ОК	指定した範囲で逆アセンブル パネルの内容を印刷するために,このダイアログを クローズして Windows の印刷用 ダイアログをオープンします。	
キャンセル	範囲選択の設定を無視し、このダイアログをクローズします。	

備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより,現在のキャレット 位置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

ボタン	機能
ヘルプ	このダイアログのヘルプを表示します。



トレース検索 ダイアログ

トレース・データの検索を行います(「2.13.8 トレース・データを検索する」参照)。 命令レベル/ソース・レベルを選択して検索することができます。

図 A.36	トレ	ース検索 ダイアログ				
		トレース検索)
	(1)	命令レベル ソース・レベ	01			
	П	検索条件の指定				
		フェッチ・アドレス(工):		- (第2日	を指定する場合 🕨 👻	
		命令(M)		•		
		アクセス・アドレス(<u>A</u>):		- (1131#	を指定する場合 📦 👻	
	(2) —	アクセスの種類(S):	(指定なし)	Y		
		データ(<u>D</u>):	HEX	- HEX (範囲を指定する 📦 👻	
		検索範囲の指定				
		番号(U):		-	•	
	Г		at the state of the local data	· http://www.co		
機能ボタ	기민		利を使業(<u>日</u>)	大を使来し		

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレース パネルにおいて, ツールバーの 🚉 ボタンをクリック
- トレース パネルにおいて、コンテキスト・メニュー→ [検索 ...]を選択

[各エリアの説明]

- (1) タブ選択エリア
 タブを選択することにより、検索するレベルが切り替わります。
 このダイアログには、次のタブが存在します。
 - [命令レベル] タブ
 - [ソース・レベル] タブ
- (2) 検索条件設定エリア 検索する際の詳細条件を設定します。 表示内容/設定方法についての詳細は、該当するタブの項を参照してください。



ボタン	機能
前を検索	指定した範囲内で、番号の小さい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし、不正な値を指定している場合、またはプログラム実行中は、メッセージ を表示し、トレース・データの検索は行いません。 また、トレースパネルが非表示の場合、または他のパネルにフォーカスがある 状態からこのダイアログへフォーカスを移動した場合、このボタンは無効となり ます。
次を検索	指定した範囲内で,番号の大きい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし,不正な値を指定している場合,またはプログラム実行中は、メッセージ を表示し,トレース・データの検索は行いません。 また、トレースパネルが非表示の場合,または他のパネルにフォーカスがある 状態からこのダイアログへフォーカスを移動した場合,このボタンは無効となり ます。
キャンセル	トレース・データの検索の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



[命令レベル] タブ

取得したトレース・データを命令レベルで検索します。

- 注意 トレースパネルをソース表示モードで表示している場合、このタブで命令レベルの検索を行っても対象を正しく検索することはできません。
 命令レベルの検索を行う際は、混合表示モード、または逆アセンブル表示モードで表示を行ってください。
- 図 A.37 トレース検索 ダイアログ:[命令レベル]タブ

	トレース検索			
	命令レベル ソース・レイ	CIL .		
Г	検索条件の指定			
	フェッチ・アドレス(工):		 (範囲を指知) 	定する場合 🕨 👻
	命令(M)			
(1)	アクセス・アドレス(<u>A</u>):		 (範囲を指定 	Egā場合 ⊯ ▼
	アクセスの種類(S):	(指定なし)	¥	
	データ(<u>D</u>):	HEX	▼ - MEX (第1日を	指定する 🕨 👻
	検索範囲の指定			
(2)-	番号(U):		-	•
「继船书方、」	L	前を絵奏(中)	(から検索(2)) ないいわ	
		HURSTEEN(D)	Weizmu) TYJC	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレース パネルにおいて, ツールバーの 🚉 ボタンをクリック
- トレース パネルにおいて, コンテキスト・メニュー→ [検索...]を選択

[各エリアの説明]

(1) [検索条件の指定]エリア

(a) [フェッチ・アドレス]
 検索条件として必要な場合、フェッチ・アドレスを指定します。
 アドレス式をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10個)。
 フェッチ・アドレスの指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスにアドレス式を指定することにより範囲を指定します。
 右側のテキスト・ボックスが空欄、または"(範囲を指定する場合に入力)"の場合は、左側のテキスト・ボックスに指定された固定アドレスで検索を行います。

なお、マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は、上位のアドレス値 をマスクして扱います。 また、32 ビットで表現できる値より大きいアドレス値を指定することはできません。

- (b) [命令] 検索条件として必要な場合、命令の文字列を指定します。 ここで指定した文字列をトレースパネルの「ソース/逆アセンブル」エリア内より検索します。 命令をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します (最大履歴数:10個)。 なお、検索の際は、大文字/小文字は区別せず、部分一致も検索の対象とします。
- [アクセス・アドレス] (c) 検索条件として必要な場合、アクセス・アドレスを指定します。 アドレス式をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択 します(最大履歴数:10個)。 アクセス・アドレスの指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスに アドレス式を指定することにより範囲を指定します。 右側のテキスト・ボックスが空欄、または"(範囲を指定する場合に入力)"の場合は、左側のテキスト・ ボックスに指定された固定アドレスで検索を行います。 なお、マイクロコントローラのアドレス空間よりも大きいアドレス値が指定された場合は、上位のアドレス値 をマスクして扱います。 また、32ビットで表現できる値より大きいアドレス値を指定することはできません。

(d) [アクセスの種類]

この項目は[アクセス・アドレス]が指定された場合のみ有効となります。 アクセスの種類を次のドロップダウン・リストより選択します。 アクセスの種類を限定しない場合は、"(指定なし)"を選択してください。

(指定なし)
リード/ライト
リード
ライト
ベクタ・リード【シミュレータ】
DMA

注意 [E1] [E20] [EZ Emulator] トレース パネル上で収集可能なアクセスの種類は、リードおよびライトです。そのため、ド ロップダウン・リストよりリード/ライト、ベクタ・リードおよび DMA を選択しないでくだ さい。 【シミュレータ】 トレース パネル上で収集可能なアクセスの種類は、リード、ライトおよびベクタ・リードで す。そのため、ドロップダウン・リストよりリード/ライトおよび DMA を選択しないでくだ さい。

(e) [データ]

この項目は「アクセス・アドレス」が指定された場合のみ有効となります。 アクセスした数値を指定します。 16 進数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択 します(最大履歴数:10個)。 数値の指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスにデータを指定す ることにより範囲を指定します。 右側のテキスト・ボックスが空欄,または"(範囲を指定する場合に入力)"の場合は、左側のテキスト・ ボックスに指定された固定数値で検索を行います。

- [検索範囲の指定] エリア (2)
 - (a) [番号]

検索するトレース・データの範囲を、トレースパネルの「番号」エリアに表示されている番号で指定します。 左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します(デフォルトでは、"0"~" 最終番 *号*"が指定されます)。 10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を 選択します(最大履歴数:10個)。

右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。

ボタン	機能
前を検索	指定した範囲内で,番号の小さい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし,不正な値を指定している場合はメッセージを表示し,トレース・データ の検索は行いません。また,トレースパネルが非表示の場合,または他のパネ ルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合,こ のボタンは無効となります。
次を検索	指定した範囲内で,番号の大きい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし,不正な値を指定している場合はメッセージを表示し,トレース・データ の検索は行いません。また,トレースパネルが非表示の場合,または他のパネ ルにフォーカスがある状態からこのダイアログへフォーカスを移動した場合,こ のボタンは無効となります。
キャンセル	トレース・データの検索の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



[ソース・レベル] タブ

取得したトレース・データをソース・レベルで検索します。

- 注意 トレース パネルを逆アセンブル表示モードで表示している場合、このタブでソース・レベルの検索を行っても対象を正しく検索することはできません。 ソース・レベルの検索を行う際は、混合表示モード、またはソース表示モードで表示を行ってください。
- 図 A.38 トレース検索 ダイアログ:[ソース・レベル] タブ

	トレース検索	
	命令レベル ソース・	
(1)_	 検索対象の指定 ● ソース行を指定して ● 関数名を指定して ● グローバル変数名 	て実行箇所を検索(①) 「先頭アドレスの実行箇所を検索(丁) を指定してアクセス箇所を検索(①)
	検索条件の指定	
	ソース行(S):	
	閲数:名(N):	v
(2)-	変数名(⊻):	
	/種漢頁(<u>K</u>):	参照/代入
	変数値(<u>A</u>):	WEX THE
	検索範囲の指定	
(3)-	番号(U):	
「 継 能 ギ 友 、 」 「		(前本投帯(D)) (方本投帯(C)) (私いうけの) へいかいの
「		

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- トレース パネルにおいて, ツールバーの 🚉 ボタンをクリック
- トレース パネルにおいて, コンテキスト・メニュー→ [検索 ...]を選択

[各エリアの説明]

(1) [検索対象の指定]エリア 検索する対象を次のオプション・ボタンの中から選択します。

ソース行を指定して実行箇所を検索	指定したソースの実行箇所を検索します(デフォルト)。 検索条件として [ソース行] の指定のみが有効となります。
関数名を指定して先頭アドレスの実行	指定した関数の実行箇所を検索します。
箇所を検索	検索条件として [関数名] の指定のみが有効となります。



グローバル変数名を指定してアクセス	指定したグローバル変数をアクセスした箇所を検索します。
箇所を検索	検索条件として「変数名」/ [種類] / [変数値] の指定のみが有
	効となります。

- (2) [検索条件の指定] エリア
 - (a) [ソース行]

この項目は"ソース行を指定して実行箇所を検索"が選択された場合のみ有効となります。 ここで指定した文字列をトレースパネルの[行番号/アドレス]エリア内より検索します。検索するソース行 に含まれる文字列を,テキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項 目を選択します(最大履歴数:10個)。 なお、検索の際は、大文字/小文字は区別せず、部分一致も検索の対象とします。

- なの、 快系の际は、 人文子/ 小文子は区別ビッ、 即力一致も快。
- 例 1. main.c#40
- 例 2. main.c
- 例 3. main
- (b) [関数名]

この項目は"関数名を指定して先頭アドレスの実行箇所を検索"が選択された場合のみ有効となります。 検索する関数名を、テキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目 を選択します(最大履歴数:10個)。 なお、検索の際は、大文字/小文字を区別し、完全一致のみを検索の対象とします。

(c) [変数名]

この項目は"グローバル変数名を指定してアクセス箇所を検索"が選択された場合のみ有効となります。 検索する変数名を、テキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目 を選択します(最大履歴数:10個)。 なお、検索の際は、大文字/小文字を区別し、完全一致のみを検索の対象とします。

- (d) [種類] この項目は"グローバル変数名を指定してアクセス箇所を検索"が選択された場合のみ有効となります。 アクセスの種類(参照/代入(デフォルト),参照,代入)をドロップダウン・リストより選択します。
- アクセスの裡類(参照/ 代入(テノオルト)、参照、代入)をトロップダリン・リストより選択しま (e) 「変数値]

この項目は"グローバル変数名を指定してアクセス箇所を検索"が選択された場合のみ有効となります。 アクセスした変数値を16進数で指定します。 変数値をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択しま す(最大履歴数:10個)。 変数値の指定は範囲で指定することができます。この場合は、左右両方のテキスト・ボックスに変数値を指定 することにより範囲を指定します。 右側のテキスト・ボックスが空欄の場合は、左側のテキスト・ボックスに指定された固定変数値でアクセス箇 所を検索を行います。

(3) [検索範囲の指定]エリア

(a) [番号]

検索するトレース・データの範囲を、トレースパネルの[番号] エリアに表示されている番号で指定します。 左右のテキスト・ボックスに、それぞれ開始番号と終了番号を指定します(デフォルトでは、"0" ~ "*最終番 号*"が指定されます)。 10 進数で番号をテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を 選択します(最大履歴数:10 個)。 左側のテキスト・ボックスが空欄の場合は、"0"の指定として扱われます。 右側のテキスト・ボックスが空欄の場合は、最終番号の指定として扱われます。

ボタン	機能		
前を検索	指定した範囲内で、番号の小さい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし、不正な値を指定している場合メッセージを表示し、トレース・データの 検索は行いません。また、トレースパネルが非表示の場合、または他のパネル にフォーカスがある状態からこのダイアログへフォーカスを移動した場合、この ボタンは無効となります。		
ボタン	機能		
-------	--	--	--
次を検索	指定した範囲内で、番号の大きい方向に検索を行います。 検索結果箇所をトレースパネル上で選択状態にします。 ただし、不正な値を指定している場合メッセージを表示し、トレース・データの 検索は行いません。また、トレースパネルが非表示の場合、または他のパネル にフォーカスがある状態からこのダイアログへフォーカスを移動した場合、この ボタンは無効となります。		
キャンセル	トレース・データの検索の設定を無効とし、このダイアログをクローズします。		
ヘルプ	このダイアログのヘルプを表示します。		



備考

組み合わせ条件設定 ダイアログ【E1】【E20】【EZ Emulator】

イベントパネル上で選択した組み合わせブレーク,またはトレース・イベントの組み合わせ条件として詳細情報の変 更を行います。

- イベントの設定に関しては、「2.17.7 イベント設定に関する留意事項」を参照してください。
- 図 A.39 組み合わせ条件設定ダイアログ(組み合わせブレーク)

	組み合わせ条件
(1) —	ブレーク
(2)	組み合わせ: シーケンシャル 👻
(3) —	対象イベント: 上 No. イベント 1 実行後 Tutorial c#67 0xffff0607 2 ライト e_htBuf 0x628 = 0x62b == 0x10
(4)	リセットイベント イベント 実行後 Tutorials#93 0xffff06b9
[機能ボタン]―	OK キャンセル ヘルプ(出)

図 A.40 組み合わせ条件設定ダイアログ(トレース・イベント)

	組み合わせ条件
(1) —	開始条件
(2)	組み合わせ: OR ・
	対象イベンド
	1/01 L
(3)	リード/ライト e_IntBuf 0x628 - 0x62b 下
(3) —	実行後 Tutorialに#93 0xffff0607 Rイベント
(4)	リセット ACA 解除
(4)	
[機能ボタン]	OK キャンセル ヘルプ(出)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において,詳細情報を編集したい組み合わせブレークにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集...]を選択
- イベント パネル上において,詳細情報を編集したいトレースにキャレットを移動したのち, コンテキスト・メニュー→ [条件の編集 ...]を選択

[各エリアの説明]

(1) タブ選択エリア

タブを選択することにより、条件に合った組み合わせ条件の詳細情報を表示します。 イベントパネル上で選択するイベントによって次の異なるタブが表示されます。

組み合わせブレーク	[ブレーク]タブのみが存在します。		
トレース	[開始条件]タブ,および[終了条件]タブが存在します。 タブを切り替えることでそれぞれの組み合わせ条件を設定できます。		

- (2) 組み合わせ条件選択エリア
 - (a) [組み合わせ]

組み合わせ条件を次のドロップダウン・リストより指定します。

条件	機能
OR	設定されたイベントのうちいずれか1つが成立した場合、条件が成立します。
AND	時間軸に関係なく設定された全てのイベントが成立した場合、条件が成立します。
シーケンシャル	設定されたイベントが指定された順序で成立した場合、条件が成立します。

注意 イベントの組み合わせに関しては、「(1) 組み合わせ条件の編集」の注意も参照してください。

(3) 対象イベント条件表示エリア

(a) リストの表示

組み合わせの対象となるイベントを一覧でリストに表示します。

No.	リストの上から順に番号が割り振られます。 組み合わせ条件で[シーケンシャル]を選択した場合のみ表示され、この番号が条件 成立の順序になります。
イベント	イベント条件の詳細情報を表示します。 イベント パネル上で表示される詳細情報と同じ表示内容です。

(b) ボタン

ボタン	機能
↑上	対象イベントリスト内で選択したイベントの順序が1つ上げます。 このボタンは、組み合わせ条件で[シーケンシャル]を設定したときのみ有効です。
ל↓	対象イベントリスト内で選択したイベントの順序が1つ下げます。 このボタンは,組み合わせ条件で[シーケンシャル]を設定したときのみ有効です。
Rイベント	対象イベントリスト内で選択したイベントをリセットイベントリストに移動します。 このボタンは,組み合わせ条件で [シーケンシャル] を設定したときのみ有効です。

- (4) リセットイベント条件表示/解除エリア
 - (a) リストの表示
 リセットイベントとして登録されたイベントの詳細条件を表示します。
 このリストに表示されたリセットイベントが成立した場合、その時点まで成立していた他のイベント条件は全てクリアされます。

注意 リセットイベントに関しては、「(3) リセットイベントの編集」の注意も参照してください。

(b) ボタン

ボタン	機能
解除	リセットイベントリスト内のイベントを対象イベントリストに移動します。 このボタンは、組み合わせ条件で[シーケンシャル]を設定したときのみ有効です。

ボタン	機能
ОК	ダイアログで指定した詳細設定を組み合わせブレーク,またはトレースに設定 し,このダイアログをクローズします。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



タイマ計測詳細設定 ダイアログ【E1】 【E20】 【EZ Emulator】

イベントパネル上で選択したタイマ・イベントの詳細情報の表示,および変更を行います。 このダイアログではタイマ・イベントのアドレス値を変更することはできません。変更する場合は、タイマ・イベントを削除して新たに作成し直してください。タイマ・イベントの設定に関しては、「2.14 実行時間の計測」を参照して ください。

図 A.41 タイマ計測詳細設定ダイアログ

	詳細	
[ツールバー]-	21 21 I	
Г	▲ 計測設定 計測項目	(祭過サイカル)
	一度だけ計測	(X),Ż
(1) —		
H	計測項目	
(2) —	計測項目を表示します。	
[機能ボタン] —		OK キャンセル ヘルプ(出)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において, 詳細情報を編集したいタイマ・イベントにキャレットを移動したのち, コンテキスト・メニュー→ [条件の編集 ...] を選択

[各エリアの説明]

- (1) 詳細情報表示/変更エリア イベントパネル上で選択しているタイマ・イベントの詳細情報を、カテゴリのリスト形式で表示し、設定の変 更を直接行うことができるエリアです。
- (2) プロパティの説明エリア詳細情報表示/変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示/変更エリアにてカテゴリを表示にします。
21	詳細情報表示/変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に 並び替えます。

[カテゴリの説明]

(1) [計測設定]

タイマ計測に関する詳細設定の表示,および変更を行います。

計測項目 ^注	- - RX600, RX700 シリーズの場合 計測項目を指定します。			
	- RX200 シリーズ, RX140 グループの場合 計測項目に経過サイクルが表示されます。			
	デフォル ト	経過サイクル		
	変更方法	ドロップダウン・	リストによる選択	
	指定可能 値	ドロップダウン・ 【RX600, RX700 * <i>経過サイクル</i> , 例外・割り込み 割り込み処理サ 例外・割り込み 【RX200 シリーズ <i>経過サイクル</i>	リストによる次のいずれか シリーズ】 <i>経過サイクル(スーパバイザモードサイクル),</i> 処理サイクル,例外処理サイクル, イクル,有効命令発行回数, 受付回数,例外受付回数,割り込み受付回数 (,RX140 グループ】	
一度だけ計測	指定された計測区間を一度だけ計測して終了するか否かを指定します。		計測して終了するか否かを指定します。	
	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	はい	指定された計測区間を一度だけ計測して終了しま す。	
		いいえ	指定された計測区間を通過した分だけ計測値が加 算されます。	

【RX140 グループ以外の RX100 シリーズ】 タイマ機能は未サポートとなります。

[機能ボタン]

注

ボタン	機能
ОК	ダイアログで指定した詳細設定をタイマ・イベントに設定し, このダイアログを クローズします。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において,詳細情報を編集したいハードウエア・ブレークにキャレットを移動したのち,コンテ キスト・メニュー→ [条件の編集...]を選択
- イベント パネル上において,詳細情報を編集したいトレース内の実行系イベントにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集...]を選択

- イベント パネル上において, 詳細情報を編集したいタイマ計測内の実行系イベントにキャレットを移動したのち, コンテキスト・メニュー→ [条件の編集 ...]を選択

[E1] [E20] [EZ Emulator]

- イベント パネル上において,詳細情報を編集したい組み合わせブレーク内の実行系イベントにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集...]を選択

[各エリアの説明]

- (1) 詳細情報表示/変更エリア イベントパネル上で選択している実行系イベントの詳細情報を、カテゴリのリスト形式で表示し、設定の変更 を直接行うことができるエリアです。
- (2) プロパティの説明エリア詳細情報表示/変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示/変更エリアにてカテゴリを表示にします。
2↓	詳細情報表示/変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に 並び替えます。

[カテゴリの説明]

(1) [アドレス]

アドレス	実行系イベントが設定されているアドレスを表示、および指定します。		
	デフォル ト	選択したイベントに依存	
	変更方法	キーボードからの直接入力	
	指定可能 値	0x0 ~ 0xFFFFFFF	

(2) [パス・カウント]

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で,イベント成立となります。		
	デフォル ト	1	
	変更方法	キーボードからの直接入力	
	指定可能 値	- 【E1】【E20】【EZ Emulator】 1 ~ 256	
		-【シミュレータ】 1~16383	

注意 1. 【E1】【E20】【EZ Emulator】
 パス・カウントは、全イベント(アクセス系イベント含む)中で1つのイベントに対してのみ1
 以外の値を指定可能です。
 2つのイベントに1以外のパス・カウントを設定した場合、エラーとなります。

注意 2. 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】 パス・カウントは、1 以外の値を指定できません。

ボタン	機能
ОК	ダイアログで指定した詳細設定を実行系イベントに設定し, このダイアログをク ローズします。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。





イベント パネル上で選択したアクセス系イベントの詳細情報の表示,または変更を行います。 なお,アクセス系イベントとはイベントパネル上において次のイベントが対象です。

- 組み合わせブレークの詳細情報内におけるリード,ライト,およびリード/ライト【E1】【E20】【EZ Emulator】
- ハードウエア・ブレークの詳細情報内におけるリード、ライト、およびリード/ライト【シミュレータ】
- トレースの詳細情報内におけるリード、ライト、およびリード/ライト
- ポイント・トレースの詳細情報内におけるリード、ライト、およびリード/ライト
- タイマ計測の詳細情報内におけるリード、ライト、およびリード/ライト



図 A.43 アクセスイベント詳細設定ダイアログ【E1】【E20】【EZ Emulator】

図 A.44 アクセスイベント詳細設定ダイアログ【シミュレータ】				
[ツールバー] − [(1) −	EXATNO FERMED 定 9 イ ア ロ Fill PFL2 PFL2	「シミュレータ」 ● ●		
(2) -	符号 符号を表示します。			
[機能ボタン]-		OK キャンセル ヘルプ(H)		

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において,詳細情報を編集したいトレース内のアクセス系イベントにキャレットを移動したの ち,コンテキスト・メニュー→ [条件の編集...]を選択
- イベント パネル上において,詳細情報を編集したいポイント・トレース内のアクセス系イベントにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集 ...]を選択
- イベント パネル上において,詳細情報を編集したいタイマ計測内のアクセス系イベントにキャレットを移動したの ち,コンテキスト・メニュー→ [条件の編集...]を選択

[E1] [E20] [EZ Emulator]

- イベント パネル上において,詳細情報を編集したい組み合わせブレーク内のアクセス系イベントにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集...]を選択

【シミュレータ】

- イベント パネル上において,詳細情報を編集したいアクセス系のハードウエア・ブレークにキャレットを移動したのち,コンテキスト・メニュー→ [条件の編集...]を選択

[各エリアの説明]

- (1) 詳細情報表示/変更エリア イベントパネル上で選択しているアクセス系イベントの詳細情報を、カテゴリのリスト形式で表示し、設定の 変更を直接行うことができるエリアです。
- (2) プロパティの説明エリア 詳細情報表示/変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

詳細情報表示/変更エリアにてカテゴリを表示にします。
詳細情報表示/変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に 並び替えます。

[カテゴリの説明]

表示および変更可能なプロパティは、使用するデバッグ・ツールにより異なります。

(1) [E1] [E20] [EZ Emulator] の場合

- (2) [シミュレータ]の場合
- (1) [E1] [E20] [EZ Emulator] の場合
 - (a) [アドレス]

アドレス	アクセス系イベントが設定されているアドレスを表示、および指定します。		
	デフォル ト	選択したイベントに依存	
	変更方法	キーボードからの直接入力	
	指定可能 値	0x0 ~ 0xFFFFFFF	

(b) [アドレス条件]

比較条件	アドレス比較条件を指定します。		
	デフォル ト	指定無し	
	変更方法	ドロップダウン・リストによる選択	
	指定可能	指定無し	アドレス比較条件を指定しません。
	1追	アドレス範囲 ^{注1注2}	アドレス範囲による比較条件を指定します。
		マスク付き比較	マスク付き比較条件を指定します。
範囲条件	アドレス比較範囲条件を指定します。 このプロパティは, [比較条件]プロパティにおいて[アドレス範囲]を選択した ときのみ表示されます。		
	デフォル ト	範囲内 (<=Value<=)	
	変更方法	ドロップダウン・リストによる選択	
	指定可能 値	範囲内 (<=Value<=)	範囲内のアクセスで条件成立します。
		範囲外 !(<=Value<=)	範囲外のアクセスで条件成立します。



終了アドレス	終了アドレスを指定します。 開始アドレスは, [アドレス]プロパティで表示された値となります。 このプロパティは, [比較条件]プロパティにおいて[アドレス範囲]を選択した ときのみ表示されます。			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFF		
マスク値	アドレスマスク値を 16 進数で指定します。 ^{注 3} このプロパティは, [比較条件]プロパティにおいて[マスク付き比較]を選択し たときのみ表示されます。			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFFF		
比較	比較条件を指定します。 このプロパティは、[比較条件]プロパティにおいて[マスク付き比較]を選 たときのみ表示されます。		コパティにおいて [マスク付き比較] を選択し	
-	デフォル ト	指定値 (==)		
	変更方法			
	指定可能 値	指定値 (==)	アドレス比較条件と一致で成立します。	
		指定值以外 (!=)	アドレス比較条件と不一致で成立します。	

注 1. 【RX100, RX200 シリーズ】

アドレス範囲による比較条件は未サポートです。

注 2. 【RX600, RX700 シリーズ】 設定可能なアドレス範囲による比較条件は、1 点のみです。

注 3. マスクは,条件となるアドレス値に対してビット単位でマスク値"0"を Don't Care とします。 例) 0x1000 ~ 0x1FFF のアドレス条件を設定する場合 アドレス値: 0x1000 マスク値: 0xF000

(c) [データ条件]

アクセス種別 - - - - - - - - - - - - - - - - - - -	アクセス種別を指定します。		
	デフォル ト	リード/ ライト	
	変更方法	ドロップダウン・リストによる選択	
	指定可能 値	リード	アクセス種別をリードアクセスとします。
		ライト	アクセス種別をライトアクセスとします。
		リード/ライト	アクセス種別をリード,およびライトアク セスとします。

アクセスサイズ	アクセスサイズを指定します。			
	デフォル ト	指定無し		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	指定無し	アクセスサイズを指定しません。 全てのアクセスサイズで成立します。	
		バイト	アクセスサイズをバイトとします。	
		ワード	アクセスサイズをワードとします。	
		ロングワード	アクセスサイズをロングワードとします。	
比較データ	比較データる	を 16 進数で指定します	o	
	デフォル ト	空欄		
	変更方法	キーボードからの直接	入力	
	指定可能 値	0x0 ~ 0xFFFFFFF		
データマスクを指定す	データマスクを指定するか否かを指定します。			
\$	デフォル ト	いいえ		
	変更方法	ドロップダウン・リストによる選択		
	指定可能	はい	データマスクを指定します。	
	1旦	いいえ	データマスクを指定しません。	
マスク値	データマスクを 16 進数で指定します。 ^注 このプロパティは, [データマスクを指定する]プロパティにおいて[はい] 定した場合のみに表示されます。		す。 ^注 を指定する] プロパティにおいて [はい] を指	
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFF		
比較条件	データ比較約 このプロパー 定した場合の	データ比較条件を指定します。 このプロパティは, [データマスクを指定する]プロパティにおいて[はい]を指 定した場合のみに表示されます。		
	デフォル ト	指定値 (==)		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	指定値 (==)	データ比較条件が一致で成立します。	
		指定值以外 (!=)	データ比較条件が不一致で成立します。	

注

マスクは、条件となるデータ値に対してビット単位でマスク値 "0"を Don't Care とします。

(d) [バス・マスタ] 【RX71M, RX66x, RX65x, RX64M, RX66T, RX671, RX72T, RX72M, RX72N グループ】

バス・マスタ	バス・マスタを指定します。			
	デフォル ト	CPU		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	CPU	指定したデータアクセスが、CPU からのアクセ スにより発生した場合に条件成立します。	
		DMAC/DTC	指定したデータアクセスが、DMAC/DTC からの アクセスにより発生した場合に条件成立します。	

注意 データアクセスのバス・マスタ選択機能が無いマイクロコントローラの場合, このプロパティ は表示されません。

(e) [パス・カウント]

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で、イベント成立となります。		
	デフォル ト	1	
	変更方法	キーボードからの直接入力	
	指定可能 値	1 ~ 256	

注意 1. 【E1】【E20】【EZ Emulator】
 パス・カウントは、全イベント(実行系イベント含む)中で1つのイベントに対してのみ1以外の値を指定可能です。
 2つのイベントに1以外のパス・カウントを設定した場合、エラーとなります。

- **注意 2.** 【E1(Serial) / E20(Serial) / EZ Emulator【RX100, RX200 シリーズ】】 パス・カウントは、1 以外の値を指定できません。
- (2) [シミュレータ]の場合
 - (a) [アドレス]

アドレス	アクセス系イベントが設定されているアドレスを表示、および指定します。		
	デフォル ト	選択したイベントに依存	
	変更方法	キーボードからの直接入力	
	指定可能 値	0x0 ~ 0xFFFFFFF	

(b) [アドレス条件]

比較条件	アドレス比較	狡条件を表示します。
	デフォル ト	指定無し
	変更方法	変更不可

(c) [データ条件]

アクセス種別 アクセス種別を指定します。				
	デフォル ト	選択したイベントに依存		
	変更方法	ドロップダウン・リストによる選択		
	指定可能	リード	アクセス種別をリードアクセスとします。	
	但	ライト	アクセス種別をライトアクセスとします。	
		リード / ライト	アクセス種別をリード,およびライトアク セスとします。	
アクセスサイズ	アクセスサイズを指定します。			
	デフォル ト	選択したイベントに依存		
	変更方法	ドロップダウン・リストによる選択		
	指定可能 値	指定無し	アクセスサイズを指定しません。 全てのアクセスサイズで成立します。	
		バイト	アクセスサイズをバイトとします。	
		ワード	アクセスサイズをワードとします。	
		ロングワード	アクセスサイズをロングワードとします。	



比較条件	データ比較条件を指定します。 このプロパティは、[アクセスサイズ]プロパティにおいて[指定無し]を指定 した場合は表示されません。				
	デフォル ト	一致 (==)			
	変更方法	ドロップダウン・リストによる選択			
	指定可能 値	一致 (==)	データが指定値と一致で成立します。		
		不一致 (!=)	データが指定値と不一致で成立します。		
		符号反転	前回アクセス時のデータと今回アクセス時 のデータで符号が反転した場合に成立しま す。 ^{注 1}		
		差分	前回アクセス時のデータと今回アクセス時 のデータの差分が指定値を超えた場合に成 立します。 ^{注1}		
		指定値より大きい (>)	データが指定値より大で成立します。		
		指定値より小さい (<)	データが指定値より小で成立します。		
		指定值以上 (>=)	データが指定値以上で成立します。		
		指定值以下 (<=)	データが指定値以下で成立します。		
		範囲内 (<=Value<=)	データが指定値の範囲内で成立します。 ([比較データ 1] <= データ <= [比較データ 2])		
		範囲外 !(<=Value<=)	データが指定値の範囲外で成立します。 (データ < [比較データ 1] [比較データ 2] < データ)		
		指定無し	比較データを指定しません。		
比較データ 1	比較データを 16 進数で指定します。 このプロパティは, [比較条件] プロパティにおいて以下を指定した場合に表示 されます。				
	- [一致 (==)]				
	- [不一致 (!=)]				
	- [指定値より大きい (>)]				
	- [指定値より小さい (<)]				
	- [指定值以上 (>=)]				
	- L				
	- [範囲内 (*	(<=Value<=)]			
	デフォル ト	空欄			
	変更方法	キーボードからの直接入力			
	指定可能 値	0x0 ~ 0xFFFFFFF			

比較データ 2	比較データを 16 進数で指定します。 このプロパティは, [比較条件]において以下を指定した場合に表示されます。			
	- [範囲内 (<=Value<=)]			
	- [範囲外 !(<=Value<=)]			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFF		
差分	比較データを 16 進数で指定します。 このプロパティは, [比較条件] において [差分] を指定した場合のみ表示され ます。			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFF		
データマスクを指定す る	データマスクを指定するか否かを指定します。 このプロパティは,[比較条件]において以下を指定した場合に表示されます。			
	- [一致 (==)]			
	- [不一致 (!=)]			
	- [指定値より大きい (>)]			
	- [指定値より小さい (<)]			
	- [指定值以上 (>=)]			
	- [指定值以	下 (<=)]		
	- [範囲内 (<	<=Value<=)]		
	- [範囲外 !(<=Value<=)]			
	デフォル ト	いいえ		
	変更方法	ドロップダウン・リス	トによる選択	
	指定可能	はい	データマスクを指定します。	
		いいえ	データマスクを指定しません。	
マスク値	データマスクを 16 進数で指定します。 ^{注2} このプロパティは, [データマスクを指定する]プロパティにおいて[はい]を指 定した場合のみに表示されます。			
	デフォル ト	空欄		
	変更方法	キーボードからの直接入力		
	指定可能 値	0x0 ~ 0xFFFFFFFF		



符号	比較するデータに符号があるか否かを指定します。 このプロパティは, [比較条件] において以下を指定した場合に表示されます。		
	- [差分]		
	- [指定値よ	り大きい (>)]	
	- [指定値よ	り小さい (<)]	
	- [指定值以	上 (>=)]	
	- [指定值以	下 (<=)]	
	- [範囲内 (<=Value<=)]		
	- [範囲外 !(<=Value<=)]		
	デフォル <i>符号あり</i> ト		
	変更方法 ドロップダウン・リストによる選択		
	指定可能	符号あり	符号ありでデータを比較します。
	値 	符号なし	符号なしでデータを比較します。

注 1. [符号反転],および [差分] は前回データを比較するため、リセット後、および条件成立 1 回 目の判定は常に不成立となります。

注 2. マスクは,条件となるデータ値に対してビット単位でマスク値 "0"を Don't Care とします。

ボタン	機能
ОК	ダイアログで指定した詳細設定をアクセス系イベントに設定し, このダイアログ をクローズします。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



割り込みイベント詳細設定 ダイアログ【シミュレータ】

イベントパネル上で選択した割り込みイベントの詳細情報の表示、および変更を行います。

図 A.45 割り込みイベント詳細設定ダイアログ【シミュレータ】

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [カテゴリの説明]
- [機能ボタン]

[オープン方法]

- イベント パネル上において,詳細情報を編集したい割り込みイベントにキャレットを移動したのち, コンテキスト・メニュー→ [条件の編集 ...]を選択

[各エリアの説明]

- (1) 詳細情報表示/変更エリア イベントパネル上で選択している割り込みイベントの詳細情報を、カテゴリのリスト形式で表示し、設定の変 更を直接行うことができるエリアです。
- (2) プロパティの説明エリア 詳細情報表示/変更エリアで選択したカテゴリやプロパティの簡単な説明を表示します。

[ツールバー]

	詳細情報表示/変更エリアにてカテゴリを表示にします。
21	詳細情報表示/変更エリアにてカテゴリを非表示にしてプロパティ項目のみを昇順に 並び替えます。

[カテゴリの説明]

(1) [アドレス]

アドレス	割り込みイベントが設定されているアドレスを表示します。 ここでは,アドレス値を指定することはできません。		
	デフォル ト	選択したイベントに依存	
	変更方法	変更不可	

(2) [パス・カウント]

パス・カウント	パス・カウントを 10 進数で指定します。 パス・カウント分のイベント条件を満たした時点で、イベント成立となります。	
	デフォル ト	1
	変更方法	キーボードからの直接入力
	指定可能 値	1 ~ 16383

(3) [割り込み]

割り込みベクタ	割り込みベクタを指定します。		
	デフォル ト	選択したイベントに依存	
	変更方法	キーボードからの直接入力	
	指定可能 値	0x00 ~ 0xFF	
優先順位	割り込み優ら	込み優先順位を指定します。	
	デフォル ト	選択したイベントに依存	
	変更方法	キーボードからの直接入力	
	指定可能 値	-【RX610 グループ】 1~8 8を指定した場合は、高速割り込みとして動作します。	
		-【RX610 グループ以外】 1 ~ 16 16 を指定した場合は、高速割り込みとして動作します。	

ボタン	機能
ОК	ダイアログで指定した詳細設定を割り込みイベントに設定し、このダイアログを クローズします。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



COM ポート設定 ダイアログ

マイクロコントローラからの通信をリダイレクトするホスト・マシン上の COM ポートを設定します。

図 A.46	CO	COM ポート設定 ダイアログ			
		COMポート設定			
	(1)-	COMポード(P): COM1 ・ ボーレード(<u>B</u>): システム ・			
[機能ボタ	ン]-[接続(2) キャンセル ヘルブ(出)			

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- デバッグ・コンソール パネルにおいて、コンテキスト・メニュー→ [COM ポートの設定 ...]を選択

[各エリアの説明]

- [COM ポート] エリア リダイレクト先の COM ポートを選択します。ホスト・マシン上に利用可能な COM ポートがない場合、このド ロップダウン・リストは空になります。
- (2) [ボーレート] エリア リダイレクト先の COM ポートとのボーレートを選択します。 ドロップダウン・リストには、次のボーレートが表示されます。"システム"を選択した場合は、デバイス・マ ネージャの設定に従います。 システム、1200、2400、4800、9600、19200、38400、115200

ボタン	機能
	COM ポートに接続(デフォルト)/切断します。 COM ポートに接続状態の場合は,ボタンの表示名は " 切断 " になります。 COM ポートから切断状態の場合は,ボタンの表示名は " 接続 " になります。
キャンセル	COM ポートの設定を無効とし,このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



スクロール範囲設定 ダイアログ

メモリパネル/逆アセンブルパネルの垂直スクロール・バーのスクロール範囲の設定を行います。 適正な範囲を設定することにより、パネルの垂直スクロール・バー上のスライダの大きさが変化し、マウスによるド ラッグなどの操作性が向上します。

- **注意** このダイアログによりスクロール範囲を設定したのち、ライン・アセンブルなどの実行により指定した アドレス式が表すアドレスに変更が生じても、スクロール範囲の修正は行いません。
- 備考 [Page Up] / [Page Down] / [↑] / [↓] キー,スクロール・バー端のボタン,またはジャンプ 系のメニュー項目の選択による移動は、スクロール範囲外でも可能です。

図 A.47 スクロール範囲設定 ダイアログ

[スクロール範囲設定		
(1)	開始アドレス(<u>S</u>):	(開始アドレスを入力してください)	
(2)	終了アドレス(<u>E</u>):	(終了アドレスを入力して(ださい)	-
[機能ボタン] ―		OK キャンセル	

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- メモリ パネルにおいて,ツールバーの [表示]→ **(葉)** ボタンをクリック
- メモリ パネルにおいて, コンテキスト・メニューの [表示] → [スクロール範囲の設定 ...] を選択
- 逆アセンブル パネルにおいて,ツールバーの [表示] → 🚺 ボタンをクリック
- 逆アセンブル パネルにおいて、コンテキスト・メニューの [表示] → [スクロール範囲の設定 ...]を選択

[各エリアの説明]

(1) [開始アドレス]エリア スクロールする範囲の開始アドレスを指定します。 アドレス式をテキスト・ボックスに直接入力するか(最大指定文字数:1024文字),またはドロップダウン・リ ストより入力履歴項目を選択します(最大履歴数:10個)。 なお、ドロップダウン・リスト内の"全範囲"を指定すると、スクロール範囲の設定は行いません(範囲は制限 されません)。

- 備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより,現在のキャレット位 置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。
- (2) [終了アドレス]エリア スクロールする範囲の終了アドレスを指定します。 アドレス式をテキスト・ボックスに直接入力するか(最大指定文字数:1024 文字),またはドロップダウン・リ ストより入力履歴項目を選択します(最大履歴数:10 個)。 ただし,[開始アドレス]において、"全範囲"を指定している場合、このエリアは無効となります。
 - 備考 このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより,現在のキャレット位 置のシンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

ボタン	機能
ОК	指定したスクロール範囲を対象パネルに設定し、開始アドレスを表示の先頭とし て対象パネルにキャレットを移動します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



指定位置へ移動 ダイアログ

指定した位置にキャレットを移動します。

- 図 A.48 指定位置へ移動 ダイアログ (1)-
- ここでは、次の項目について説明します。
- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- [編集] メニュー→ [移動 …] を選択
- 逆アセンブル パネルにおいて, コンテキスト・メニューの [移動 ...]を選択
- IOR パネルにおいて、コンテキスト・メニューの [移動 ...]を選択

[各エリアの説明]

 (1) [アドレス/シンボル] / [IOR] エリア キャレットを移動したい箇所を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:1024文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 対象となるパネルにより,指定内容は次のように異なります。

対象パネル	指定内容
逆アセンブル パネル	アドレス式
IOR パネル	I/O レジスタ名

備考 逆アセンブルパネルよりこのダイアログをオープンした場合,このテキスト・ボックスで [Ctrl] + [Space] キーを押下することにより,現在のキャレット位置のシンボル名を補完することがで きます(「2.21.2 シンボル名の入力補完機能」参照)。

ボタン	機能
ОК	指定した位置を表示の先頭として対象パネルにキャレットを移動します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



データ保存 ダイアログ

逆アセンブル パネル/メモリ パネル/トレース パネルの表示内容,およびアップロード・データの保存(「2.5.3 アップロードを実行する」参照)を行います。

なお、このダイアログは、デバッグ・ツールと接続時のみオープンすることができる	とす 。
--	-------------

図 A.49	デー	-タ保存 ダイアログ			
		データ保存 - 逆ア	データ保存 - 逆アセンブル・データ		
	(1) –	ファイル名(凶):	C:¥WorkSpace¥Tutorial¥逆アセンブル1 ・ 「」()		
	(2) -	ファイルの種類①	「テキスト・ファイル (*.txt) ▼		
	Γ	-保存範囲 アドレス	/シンボル(<u>A</u>):		
	(3)—	_sbrk+0x28			
[機能ボク	タン]-[保存(S) キャンセル ヘルプ(H)		

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 逆アセンブル パネルにフォーカスがある状態で, [ファイル] メニュー→ [名前を付けて逆アセンブル・データを 保存...]を選択
- メモリ パネルにフォーカスがある状態で, [ファイル] メニュー→ [名前を付けてメモリ・データを保存 ...]を選 択
- トレース パネルにフォーカスがある状態で, [ファイル] メニュー→ [名前を付けてトレース・データを保存 ...] を選択
- [デバッグ] メニュー→ [デバッグ・ツールからアップロード ...]を選択

[各エリアの説明]

- (1) [ファイル名] エリア 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[...] ボタンをクリックすることでオープンするデータ保存ファイルを選択 ダイアログにより、ファイル を選択することもできます。 なお、パス情報を含まずファイル名のみを指定した場合は、プロジェクト・フォルダが対象となります。
- (2) [ファイルの種類]エリア 保存するファイルの形式を次のドロップダウン・リストにより選択します。 保存する対象により、選択できるファイルの形式が次のように異なります。
 - (a) パネルの表示内容を保存する場合

テキスト・ファイル (*.txt)	テキスト形式(デフォルト)
CSV(カンマ区切り)(*.csv)	CSV 形式 ^注

- 注 各データを "," で区切り保存します。 なお, データ内に "," が含まれている際の不正形式を避けるため, 各データを """"(ダブル クォーテーション)で括り出力します。
- (b) アップロード・データを保存する場合 選択できるファイル形式は,「表 2.3 アップロード可能なファイル形式」を参照してください。
- (3) [保存範囲 xxx] エリア

ファイルに保存する際の保存範囲を指定します。

それぞれのテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10個)。

保存する対象により、指定方法が次のように異なります。

保存対象	説明
逆アセンブル パネル	保存するアドレス範囲を,開始アドレスと終了アドレスで指定します。 16 進数の数値,またはアドレス式による入力が可能です。 パネル上で範囲選択している場合は,デフォルトでその選択範囲が指定されます。 範囲選択していない場合は,現在のパネルの表示範囲が指定されます。
メモリ パネル	保存するメモリ範囲を,開始アドレスと終了アドレスで指定します。 16 進数の数値,またはアドレス式による入力が可能です。 範囲選択していない場合は,現在のパネルの表示範囲が指定されます。
トレース パネル	- 保存範囲を指定する場合 保存するトレース範囲を開始トレース番号 ^注 と終了トレース番号で指定します。 10 進数の数値のみ入力が可能です。
	- すべてのトレース・データを保存する場合 左側のドロップダウン・リストにより、[すべてのトレース・データ]を選択しま す。右側のテキスト・ボックスが無効となり、現在取得しているトレース・デー タのすべてが保存の対象となります。 デフォルトでは、現在のパネルの表示範囲が指定されます。
アップロード・デー タ	保存するメモリ範囲を開始アドレスと終了アドレスで指定します。 16 進数の数値、またはアドレス式による入力が可能です。
アップロード・デー タ	保存するメモリ範囲を開始アドレスと終了アドレスで指定します。 16 進数の数値、またはアドレス式による入力が可能です。

注 トレース パネル上の [番号] エリアに表示されている番号を示します。

備考 このテキスト・ボックスで [Ctrl]+[Space] キーを押下することにより,現在のキャレット位置のシ ンボル名を補完することができます(「2.21.2 シンボル名の入力補完機能」参照)。

ボタン	機能
保存	指定したファイルに、指定した形式でデータを保存します。
キャンセル	データ保存の設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



関数変数アクセス表 パネル

変数にアクセスしている関数を直交表で表示するパネルです。

図 A.50 関数	変数アクセス	表 パネル									
	関数実数アクセ	ス表									×
[ツールバー] –											<u>.</u>
(1) _	変数名 マ9 _Stdout _Stderr _Files _slptr Fixed_Yeo… i j	Excep_RIIC1_TEII	7₽	PowerON_Reset	~₽	msin ⊽-⇔ R(1)M(1) R(1)M(1)	sbrk ⊽	⊨ abort	⊽#	sub 1 R(1)# R(1)#	Ø≉ ^ (1)
	<										>
(2) —	、関数変数ア	かしス表(直交表) /									•

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]

- [表示] メニュー→ [関数変数アクセス表] を選択

[各エリアの説明]

- (1) 表エリア
 - (a) [関数変数アクセス表(直交表)] タブ アクティブ・プロジェクトにおいて、変数にアクセスしている関数を直交表で表示します。
 この情報はクロス・リファレンス情報とマップ情報を用いて作成しています。情報が表示されない場合は、 ツールバーの右端の (直交表を生成するためのビルド・オプションを有効にしてビルドするボタン)を クリックして情報を生成してください。

また、C ソースを静的解析して得られた関数の変数へのアクセス状態を表示します。

- "R"と記載されているセルは変数値を読み込んだことを表します。かっこ内の数字は関数内での変数の読み 込み箇所の数を示します。
- "W"と記載されているセルは変数値を書き込んだことを表します。かっこ内の数字は関数内での変数の書き 込み箇所の数を示します。

変数名、または変数へのアクセス状態を表しているセルをダブルクリックすることで、その定義箇所をエディ タで表示します。

(2) タブ選択エリア 変数にアクセスしている関数を直交表で表示する[関数変数アクセス表(直交表)]タブがあります。

[ツールバー]

ツールバーは、コンパイラがクロスリファレンスの生成に対応している場合のみ表示します。

クロス・リファレンスの生成に関するビルド・オプションが無効の場合,有効にしてビルド します。その際,テキスト・エディタで編集中のファイルがある場合は,該当ファイルを保 存します。

[[]オープン方法]

消費電流測定 パネル【E2】

消費電流測定ソリューションの機能の中心となるパネルです。 消費電流測定ソリューションについての詳細は、「2.19 消費電流測定【E2】」を参照してください。

- **注意 1.** 消費電流測定は RX100, RX200 シリーズでのみ対応しています。
- **注意 2.** 測定結果には誤差が発生します。消費電流測定のスペック(測定可能な最大消費電流 / 最大時間や消費 電流の分解能等)については、E2 エミュレータのユーザーズ・マニュアルを参照してください。
- 図 A.51 消費電流測定 パネル



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [ツールバー]
- [コンテキスト・メニュー]

[オープン方法]

- [デバッグ] メニュー→ [デバッグ・ソリューション] → [消費電流測定] を選択
- ソリューション一覧 パネル上の消費電流測定の [GO] ボタンをクリック

[各エリアの説明]

- (1) 警告表示エリア 消費電流測定の現在の測定条件とデバッグ・ツール接続前に設定すべき条件に矛盾があった場合,警告を表示し ます。 なお,表示すべき警告がない場合,このエリアは非表示となります。
- (2) グラフ表示エリア 消費電流測定結果の電流値データを縦軸、時間を横軸にしてグラフ表示します。 ドラッグ可能な補助線として、以下のものを用意しています。

補助線	説明
カーソル	赤色の縦線
マーカー A, マーカー B	緑色の縦線
検索用補助線	橙色の横線

カーソル、マーカーをドラッグすると、情報表示エリアの情報が追随して更新されます。 検索用補助線は、検索の基準となる電流のしきい値を簡易に指定するためのものです。

備考 消費電流測定中はデータがクリアされ、カーソル、マーカーはドラッグできません。

(3) 情報表示エリア

グラフ表示エリアに表示されている測定データのグラフに関する各値を表示します。

- (a) [平均電流値]
 マーカーAとマーカーBの間の電流値の平均値を表示します。
 マーカーが非表示のときは空欄となります。
- (b) [最大電流値] マーカー A とマーカー B の間の最大の電流値を表示します。 マーカーが非表示のときは空欄となります。
- (c) [電流値] カーソル位置の電流値を表示します。 カーソルが非表示の場合は空欄となります。
- (d) [時間]
 カーソル位置の時間を表示します。
 カーソルが非表示の場合は空欄となります。
- (e) [マーカー A]
 マーカー A の位置の時間を表示します。
 マーカー A が非表示の場合は空欄となります。
- (f) [マーカー B]
 マーカー B の位置の時間を表示します。
 マーカー B が非表示の場合は空欄となります。
- (g) [B-A]
 マーカーAとマーカーBの間の時間を表示します。
 マーカーが非表示の場合は空欄となります。
- (4) 通過ポイントー覧表示エリア 設定されている通過ポイントの一覧の表示と、通過ポイント設定時のビルド&ダウンロードにかかわる設定を行います。
 - (a) [測定時に指定した通過ポイント用ビルド・オプションを今後も付加する]
 通過ポイント用 nop 命令の自動挿入機能でビルドした場合、ビルドに使用したオプションをビルド・ツールのプロパティに反映します。
 - **注意** このチェック・ボックスは、現在のアクティブ・プロジェクトのコンパイラが CC-RX V2.08.00 以上の場合のみ有効となります。
 - (b) [通過ポイントー覧] 現在設定されている通過ポイントを一覧で表示します。 グラフ表示エリアに測定結果が表示されている場合、一覧の各行を選択した状態でセルをダブルクリックする と、その行の表す通過ポイントの位置へグラフ表示エリアのカーソルが移動します。

各項目の説明は以下です。

<1> チェック・ボックス

チェック・ボックスがチェック状態の通過ポイントが消費電流測定時に有効となります。

- <2> カーソル グラフ表示エリア上で、その行の表す通過ポイント上にカーソルがあることを表します。 カーソルがある場合、セルの背景色が赤になります。
- <3> 通過回数 その行の表す通過ポイントを測定中に何回通過したかを表示します。

<4> 位置

その行の表す通過ポイントの位置を「ファイル名#行番号(アドレス)」の形式で表示します。

例 main.c#16(0xFFF0E)

注意 アドレスは測定時にアドレスが決定している場合のみ表示します。

<5> 線色

グラフ表示エリア上で、その行の表す通過ポイントの縦線の色を表示します。

[`]	ソ–	ール	バー	-]
L	ノー	ール	/ \-	- J

	プログラムを実行し,消費電流の測定を開始します。 アクティブ・プロジェクトのコンパイラが CC-RX V2.08.00 以上の場合,アクティ ブ・プロジェクトに登録されているソース上の通過ポイント位置に nop 命令を挿 入するためのリビルド&ダウンロードを行います。
	プログラムを実行し, 消費電流の測定を開始します。 ただし, 通過ポイント位置に nop 命令を挿入するためのリビルド&ダウンロード は行いません。 このボタンは, アクティブ・プロジェクトのコンパイラが CC-RX V2.08.00 以上の 場合のみ表示されます。
	メイン・ウインドウ上のデバッグ・ツールバーの同ボタンの機能と同じです。
	測定条件の設定を行うための消費電流測定条件設定 ダイアログ【E2】をオープン します。
5	測定時のトリガ条件の設定を行うための消費電流測定トリガ条件設定 ダイアログ 【E2】をオープンします。
•	グラフ表示エリア上でマーカー A を表示領域内の位置へ移動します。
•	グラフ表示エリア上でマーカー B を表示領域内の位置へ移動します。
	グラフ表示エリア上でカーソルを表示領域内の位置へ移動します。
8	グラフ表示エリア上のカーソル、マーカー A、マーカー B、検索用補助線の表示/ 非表示を切り替えます。
₹	検索用補助線と検索モードの条件に一致する最初の位置を検索します。 見つかった場合はグラフ表示エリア上のカーソルを検出位置へ移動します。
4 1	検索用補助線と検索モードの条件に一致するカーソル位置より前かつ最初の位置 を検索します。 見つかった場合はグラフ表示エリア上のカーソルを検出位置へ移動します。
	検索用補助線と検索モードの条件に一致するカーソル位置より後かつ最初の位置 を検索します。 見つかった場合はグラフ表示エリア上のカーソルを検出位置へ移動します。
N	検索用補助線と検索モードの条件に一致する最後の位置を検索します。 見つかった場合はグラフ表示エリア上のカーソルを検出位置へ移動します。
•	 ↓ ↓ ↓ から検索する際の検索モードを選択します。 検索モードは"しきい値以上", "しきい値以下", "範囲内", "範囲外", "通過ポイント"から選択可能です。
0,	詳細な条件を指定して検索するための消費電流測定検索 ダイアログ【E2】をオー プンします。
<u>ل</u>	グラフ表示エリアのカーソル移動をエディタ パネルと連動させるかどうかを選択 します。 ON の場合、カーソルが通過ポイント上に停止すると、通過ポイントを設定した ソース位置をエディタ パネルで開きます。

	グラフ表示エリアのカーソル移動を逆アセンブル パネルと連動させるかどうかを 選択します。 ON の場合,カーソルが通過ポイント上に停止すると,通過ポイントを設定した ソース位置を逆アセンブル パネルで開きます。エディタ パネルとの連動も ON に なっている場合,エディタ パネルとの連動が優先されます。
	消費電流測定の測定結果の電流値データを CSV 形式で保存するために,電流値保存 ダイアログ【E2】をオープンします。 本ボタンで保存したファイルは,
	- 『こうしん CSV ファイルを読み込みんでグラフ表示エリアに表示します。
T	消費電流測定の測定結果,および測定時の条件設定をまとめて CSV,または Microsoft Office Excel ブック形式でファイルに保存するために,ファイル保存ダイ アログをオープンします。
×	グラフ表示エリアから測定結果のグラフをクリアします。
**	グラフ表示エリアから 羄 で読み込んだデータのグラフをクリアします。

[コンテキスト・メニュー]

測定電流値保存	消費電流測定の測定結果の電流値データを CSV 形式で保存するために、電流値保存 ダイアログ【E2】をオープンします。 本メニューで保存したファイルは、[測定電流値読み込み]メニューによりグラフ 表示エリアに表示することができます。
測定電流値読み込み	[測定電流値保存] メニューで保存した CSV ファイルを読み込みんでグラフ表示 エリアに表示します。
測定結果クリア	グラフ表示エリアから測定結果のグラフをクリアします。
読み込みデータをクリア	グラフ表示エリアから[測定電流値読み込み]メニューで読み込んだデータのグ ラフをクリアします。
時間の単位	グラフ表示エリアの X 軸の時間単位を切り替えるためのカスケード・メニューを 表示します。
秒	グラフ表示エリアの X 軸の時間単位を秒に切り替えます。
ミリ秒	グラフ表示エリアのX軸の時間単位をミリ秒に切り替えます。
マイクロ秒	グラフ表示エリアのX軸の時間単位をマイクロ秒に切り替えます。
拡大	グラフ表示エリアのX軸,およびY軸の表示範囲を拡大するためのカスケード・ メニューを表示します。
拡大	グラフ表示エリアのX軸、およびY軸の表示範囲を拡大します。
×軸を拡大	グラフ表示エリアの X 軸のみ表示範囲を拡大します。
Y軸を拡大	グラフ表示エリアのY軸のみ表示範囲を拡大します。
縮小	グラフ表示エリアの X 軸,および Y 軸の表示範囲を縮小するためのカスケード・ メニューを表示します。
縮小	グラフ表示エリアのX軸、およびY軸の表示範囲を縮小します。
X軸を縮小	グラフ表示エリアの X 軸のみ表示範囲を縮小します。
Y軸を縮小	グラフ表示エリアのY軸のみ表示範囲を拡大します。
拡大/縮小をリセット	グラフ表示エリアの X 軸,および Y 軸の表示範囲を初期値に戻します。



消費電流測定条件設定 ダイアログ【E2】

消費電流測定 パネル【E2】において,測定条件の設定を行います。

注意 消費電流測定は RX100, RX200 シリーズでのみ対応しています。

図 A.52 消費電流測定条件設定 ダイアログ

	測定条件設定		×
(1) –	記録メモリを使い切った後の動作(M):	記録メモリを上書きして実	行を続ける 🗸 🗸 🗸
(2) _	サンプリング時間(<u>S</u>):	10us	~
Γ	取得条件		
	条件(<u>C</u>):		বন্দে 🗸
(3) —	チャネル(<u>A</u>):		ch0 🗸
	外部トリガ入力の種類(工):		High 🗸
	通過ポイント範囲(日):	~ -	\sim
_ [機能ボタン]		OK	キャンセル ヘルプ(日)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 消費電流測定 パネル【E2】において, ツールバーの 拥 を選択

[各エリアの説明]

(1) [記録メモリを使い切った後の動作]

消費電流の測定中にエミュレータの記録メモリを使い切った後のデバッガの動作を次のドロップダウン・リスト により選択します。

記録メモリを上書きして実行 を続ける	記録メモリのデータを先頭から上書きしながらプログラムの実行を続けます。
記録を停止する	記録メモリへの書き込みのみを停止し,プログラムの実行は継続します。
プログラムを停止する	記録メモリへの書き込みもプログラムの実行も停止します。

- (2) [サンプリング時間]
 消費電流測定時のサンプリング時間を選択します。
 選択可能なサンプリング時間は以下のとおりです。
 1us, 2us, 5us, 10us, 20us, 50us, 100us, 200us, 500us, 1ms
- (3) [取得条件]エリア エミュレータの記録メモリに書き込む条件に関する設定を行います。
 - (a) [条件] エミュレータの記録メモリに書き込むための条件の種類を次のドロップダウン・リストにより選択します。 すべて 特に制限を設けず、すべての測定データを記録します。

外部トリガ入力検出中	E2 拡張インタフェースの外部トリガ入力検出機能を用いて、外部ト リガの入力が検出されている間のみ測定データを記録します。
通過ポイント間	測定開始の通過ポイントが検出されてから測定終了の通過ポイント が検出されるまでの間のみ測定データを記録します。

(b) [チャネル] [条件]が"外部トリガ入力検出中"の場合、検出対象の E2 拡張インタフェースの外部トリガ入力チャネルを "ch0", "ch1"から選択します。

- (c) [外部トリガ入力の種類]
 [条件]が"外部トリガ入力検出中"の場合、検出対象の外部トリガ入力信号を"High"、"Low"から選択します。
- (d) [通過ポイント範囲]

[条件] が"通過ポイント間"の場合,対象となる2つの通過ポイントを選択します。 ドロップダウン・リストには,現在プロジェクトに設定されている通過ポイントが表示されます。 左側に設定した通過ポイントが検出されてから,右側に設定した通過ポイントが検出されるまでの間の測定 データのみを記録します。

ボタン	機能
ОК	このダイアログで設定した測定条件を消費電流測定 パネル【E2】に反映します。
キャンセル	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



消費電流測定トリガ条件設定 ダイアログ【E2】

消費電流測定 パネル【E2】において、トリガ条件の設定を行います。

注意 消費電流測定は RX100, RX200 シリーズでのみ対応しています。

図 A.53 消費電流測定トリガ条件設定 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 消費電流測定 パネル【E2】において、ツールバーの 🔸 を選択

[各エリアの説明]

- (1) [トリガ機能を使用する]
 トリガ機能を使用するかどうかを指定します。
- (2) [トリガ動作設定] エリア トリガ検出時の E2 エミュレータの動作を設定します。
(a) [トリガ検出時の動作]

トリガ検出時の動作を次のドロップダウン・リストにより選択します。

プログラムを停止する	プログラムの実行を停止します。
外部トリガを出力	E2 拡張インタフェースの外部トリガ信号出力機能を用いて,外部トリガ信 号を出力します。

(b) [チャネル]

[トリガ検出時の動作]が"外部トリガを出力"の場合、出力対象の E2 拡張インタフェースの外部トリガ出力 チャネルを "ch0"、"ch1" から選択します。

(c) 外部トリガ出力の種類
外部トリガとして出力する信号の種類を [High パルス], [条件が成立している間 High レベルを出力] から選択します。
[High パルス] を選択した場合, パルス信号の幅を1~65535の10進数値(単位:マイクロ秒)で指定します。

- (3) [トリガ条件設定] エリア
 - トリガ条件を設定します。
 - (a) [条件]

トリガ条件となる電流値の変化の種類を次のドロップダウン・リストにより選択します。

しきい値以上	測定中の電流値が指定した電流値以上になった瞬間を検出
しきい値以下	測定中の電流値が指定した電流値以下になった瞬間を検出
範囲内	測定中の電流値が指定した電流値の範囲に入った瞬間を検出
範囲外	測定中の電流値が指定した電流値の範囲から外れた瞬間を検出

(b) [電流値]

トリガ条件の対象となる特定の電流値,または電流値範囲を uA 単位で指定します。 [条件]が"しきい値以上","しきい値以下"の場合は電流値を1つ,[条件]が"範囲内","範囲外"の場合の場合は電流値を2つ指定します。

注意 電流値の範囲は、左側の入力欄の値が右側の入力欄の値より小さくなるようにしてください。

(c) [検出幅を指定する]

[条件]が"しきい値以上"、"しきい値以下"、"範囲内"の場合、検出した瞬間ではなく電流値の変化を検出 してからその状態が特定の時間続いたときのみトリガを検出するかどうかを指定します。 トリガを検出する場合はチェック・ボックスを選択し、時間幅の条件と時間幅を指定します。

<1> 時間幅の条件

時間幅の条件を次のドロップダウン・リストにより選択します。

しきい値以上	[条件] で指定した条件が指定した時間以上続いた場合に検出	
範囲内	[条件] で指定した条件が続いた時間が, 指定した時間の範囲内の場合のみ検出	

<2> 時間幅

消費電流測定条件設定 ダイアログ【E2】の [サンプリング時間] で選択したサンプリング時間の倍数を指定します。

実際の時間幅はサンプリング時間の倍数になります。

[機能ボタン]

ボタン	機能	
ОК	このダイアログで設定したトリガ条件を消費電流測定 パネル【E2】に反映しま す。	
キャンセル	設定を無効とし、このダイアログをクローズします。	
ヘルプ	このダイアログのヘルプを表示します。	



消費電流測定検索 ダイアログ【E2】

消費電流測定 パネル【E2】において,詳細な条件を指定して検索を行います。

- **注意** 消費電流測定は RX100, RX200 シリーズでのみ対応しています。
- 図 A.54 消費電流測定検索 ダイアログ



ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 消費電流測定 パネル【E2】において, ツールバーの 🔍 を選択

[各エリアの説明]

- (1) [検索条件]エリア 検索条件を設定します。
 - (a) [条件] 検索条件の種類を次のドロップダウン・リストにより選択します。

しきい値以上	測定中の電流値が指定した電流値以上になった瞬間を検出	
しきい値以下	測定中の電流値が指定した電流値以下になった瞬間を検出	
範囲内	測定中の電流値が指定した電流値の範囲に入った瞬間を検出	
範囲外	測定中の電流値が指定した電流値の範囲から外れた瞬間を検出	

RENESAS

 通過ポイント	指定した通過ポイントを検出
------------	---------------

(b) [電流值]

検索条件の対象となる特定の電流値,または電流値範囲を uA 単位で指定します。 [条件]が"しきい値以上","しきい値以下"の場合は電流値を1つ,[条件]が"範囲内","範囲外"の場合の場合は電流値を2つ指定します。

注意 電流値の範囲は、左側の入力欄の値が右側の入力欄の値より小さくなるようにしてください。

(c) [検出幅を指定する]

[条件]が"しきい値以上"、"しきい値以下"、"範囲内"の場合、検出した瞬間ではなく電流値の変化を検出 してからその状態が特定の時間続いたときのみ検索するかどうかを指定します。 検索する場合はチェック・ボックスを選択し、時間幅の条件と時間幅を指定します。

<1> 時間幅の条件

時間幅の条件を次のドロップダウン・リストにより選択します。

しきい値以上	[条件] で指定した条件が指定した時間以上続いた場合に検出	
範囲内	[条件] で指定した条件が続いた時間が, 指定した時間の範囲内の場合のみ検出	

<2> 時間幅

消費電流測定条件設定 ダイアログ【E2】の [サンプリング時間] で選択したサンプリング時間の倍数を指定します。

実際の時間幅はサンプリング時間の倍数になります。

(d) [通過ポイント]

[条件] が"通過ポイント"の場合,対象の通過ポイントを"すべて",または現在設定されている通過ポイントから選択します。

「拙台	s Hř	h	N , T
して成用	ミイト	~	_

ボタン	機能
先頭を検索	このダイアログで設定した検索条件で、測定データ全体の先頭を検索します。
前を検索	このダイアログで設定した検索条件で,消費電流測定 パネル【E2】のグラフ表 示エリアのグラフのカーソル位置の時間から時間が進む方向に検索します。
次を検索	このダイアログで設定した検索条件で,消費電流測定 パネル【E2】のグラフ表 示エリアのグラフのカーソル位置の時間から時間が戻る方向に検索します。
末尾を検索	このダイアログで設定した検索条件で,測定データ全体の末尾を検索します。
閉じる	設定を無効とし、このダイアログをクローズします。
ヘルプ	このダイアログのヘルプを表示します。



電流値保存 ダイアログ【E2】

消費電流測定 パネル【E2】において, 電流値データの保存を行います。

注意 消費電流測定は RX100, RX200 シリーズでのみ対応しています。

図 A.55 電流値保存 ダイアログ

	電流值保存		×
(1) –	ファイル名(E):	(ファイル名を入力してください)	✓ 参照(B)
(2) –	保存範囲 9	v us - 300009	v us
[機能ボタン]-		(保存(⑤)	キャンセル ヘルプ(出)

ここでは、次の項目について説明します。

- [オープン方法]
- [各エリアの説明]
- [機能ボタン]

[オープン方法]

- 消費電流測定 パネル【E2】において、ツールバーの 🔜 を選択

[各エリアの説明]

 (1) [ファイル名] エリア 保存するファイル名を指定します。 テキスト・ボックスに直接入力するか(最大指定文字数:259文字),またはドロップダウン・リストより入力 履歴項目を選択します(最大履歴数:10個)。 また、[参照…] ボタンをクリックすることでオープンする電流値データ保存ファイルを選択 ダイアログにより、ファイルを選択することもできます。 なお、パス情報を含まずファイル名のみを指定した場合は、プロジェクト・フォルダが対象となります。

 (2) [保存範囲]エリア 保存する電流値データの範囲を開始時間と終了時間で指定します(単位:us)。10 進数の数値のみ入力が可能です。
それぞれのテキスト・ボックスに直接入力するか、またはドロップダウン・リストより入力履歴項目を選択します(最大履歴数:10個)。
"すべてのデータ"を選択した場合は、開始時間に現在取得している電流値データの最小時間、終了時間に現在取得している電流値データの最大時間を指定したものとみなします。

[機能ボタン]

ボタン	機能	
保存	指定したファイルに, CSV 形式でデータを保存します。	
キャンセル	データ保存の設定を無効とし、このダイアログをクローズします。	
ヘルプ	このダイアログのヘルプを表示します。	

RENESAS

B. 入出力機能

ここでは、低水準インタフェース・ルーチン(アセンブリ言語部分)におけるシミュレータ用の入出力機能別の指定 内容について説明します。

シミュレータでは,標準入出力,およびファイル入出力用の機能を提供しています。

シミュレータ用の入出力機能は、プロパティパネル上の[デバッグ・ツール設定]タブの[ストリーム入出力]【シ ミュレータ】カテゴリ内の[ストリーム入出力モードを選択]プロパティにおいて、[シミュレータモード]を指定した 場合使用できます。

B.1 標準入出力/ファイル入出力

低水準インタフェース・ルーチンから呼ばれる、入出力を行うファイルの内容を入出力機能用のプログラムに置き換えます。

標準入出力用の機能 GETC, PUTC については、1 文字の入出力を行う関数 "charput", "charget" (_charput, _charget) をデバッグ・コンソール機能用のプログラムに置き換え、デバッグ・コンソール パネル上へデータの入出力 を行います。

以下に入出力機能の一覧を示します。

分類	機能名	説明
標準入出力	GETC	標準入力から1バイト入力します。
	PUTC	標準出カへ1バイト出力します。
ファイル入出力	FOPEN	ファイルをオープンします。
	FCLOSE	ファイルをクローズします。
	FGETC	ファイルから1バイト入力します。
	FPUTC	ファイルへ1バイト出力します。
	FEOF	ファイルの終端をチェックします。
	FSEEK	ファイル・ポインタを指定された位置に移動します。
	FTELL	ファイル・ポインタの現在位置を取得します。

備考 低水準インタフェース・ルーチンについては、「CC-RX コンパイラ ユーザーズマニュアル」を参照し てください。

シミュレータの入出力機能を実現するためには、まず、プロパティパネルの[デバッグ・ツール設定]タブ上の[ストリーム入出力]【シミュレータ】カテゴリ内の[ストリーム入出力用アドレス]プロパティにおいて、入出力用の特定のアドレスを指定します。0~"アドレス空間の終了アドレス"の範囲のアドレス式を直接入力で指定してください。

図 B.1 [ストリーム入出力] カテゴリ

▲ ストリーム入出力

-		
	ストリーム入出力モードを選択	シミュレータモード
	ストリーム入出力機能を使用する	いいえ
\subset	ストリーム入出力用アドレス	HEX ()

シミュレータは、プログラムの命令を実行中に、指定したアドレスへのサブルーチン分岐命令(BSR, JSR)を検出 すると、R1, R2 レジスタの内容をパラメータとして入出力処理を行います。 したがって、入出力処理を行う前にプログラム内に次の設定をしておきます。

- 機能コード(R1 レジスタ)

MSB	1バイト	1バイト	1バイト	1バイト	LSB
	0x01	機能コード	0x00	0x00	



- パラメータ・ブロックのアドレス(R2 レジスタ) パラメータ・ブロックの内容は、各入出力機能の説明を参照してください。 MSB LSB
- パラメータ・ブロックおよびストリーム入出力バッファ領域の確保
- パラメータ・ブロックの各パラメータにアクセスする場合は、該当するパラメータのサイズでアクセスしてくださ い。



GETC

標準入力から1バイト入力します。

1バイト	1バイト	1バイト	1バイト
0x01	0x21	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

[パラメータ]

パラメータ	説明
<i>入力バッファ先頭アドレス</i> (入力)	入力データを書き込むバッファの先頭アドレス

[詳細説明]

- 標準入力から1バイト入力します。

[使用例]

_charge	t:		
	MOV.L	#1210000h,R1	; GETC の機能コードを R1 に設定
	MOV.L	#IO_BUF,R2	; 入力バッファ先頭アドレスを R2 に設定
	MOV.L	#PARM,R3	; パラメータ・ブロックのアドレスを R3 に設定
	MOV.L	R2,[R3]	; 入力バッファ先頭アドレスをパラメータ・ブロックに設定
	MOV.L	R3,R2	; R3(パラメータ・ブロックのアドレス)を R2 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	MOV.L	#IO BUF,R2	; 入力バッファ先頭アドレスを R2 に設定
	MOVU.B	[R2],R1	; 入力バッファ先頭 1 バイト(取得した 1 バイト文字)を R1 に設定
	RTS		; 呼び出し元のアドレスに戻る
	SECTION	B DATA ALICN=4	
DADM.	. SECTION	1	・パラメータ・ブロック領域
PARM:	.BLAL SECTION	בידבת ב בידבת ב	; ハリケーチ・ノロソノ限級
TO DUE.	. DIVI		、1山カバッファ
TO BOL:	.BLKL	Ţ	



٦

PUTC

標準出力へ1バイト出力します。

1バイト	1バイト	1バイト	1バイト
0x01	0x22	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

[パラメータ]

パラメータ	説明
<i>出力バッファ先頭アドレス</i> (入力)	出力データを格納しているバッファの先頭アドレス

[詳細説明]

- 標準出力へ1バイト出力します。

_charpu	t:		
	MOV.L	#IO_BUF,R2	; 出力バッファ先頭アドレスを R2 に設定
	MOV.B	R1,[R2]	; R1(出力文字)を出力バッファに設定
	MOV.L	#1220000h,R1	; PUTC の機能コードを R1 に設定
	MOV.L	#PARM,R3	; パラメータ・ブロックのアドレスを R3 に設定
	MOV.L	R2,[R3]	; 出力バッファ先頭アドレスを出力バッファに設定
	MOV.L	R3,R2	; R3(パラメータ・ブロックのアドレス)を R2 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	RTS		; 呼び出し元のアドレスに戻る
	.SECTION	B,DATA,ALIGN=4	
PARM:	.BLKL	1	; パラメータ・ブロック領域
	.SECTION	B_1,DATA	
IO_BUF:	.BLKL	1	; 入出力バッファ



FOPEN

ファイルをオープンします。

[機能コード (R1 レジスタ)]

1バイト	1バイト	1バイト	1バイト
0x01	0x25	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	実行結果	ファイル番号
+2 +4	オープン・モード	未使用
+6	ファイル名先頭ア	ドレス

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:エラー
ファイル番号(出力)	オープン処理以降のファイルへのアクセスで使用する番号
<i>オープン・モード</i> (入力)	0x00 : "r" 0x01 : "w" 0x02 : "a" 0x03 : "r+" 0x04 : "w+" 0x05 : "a+" 0x10 : "rb" 0x11 : "wb" 0x12 : "ab" 0x13 : "r+b" 0x13 : "r+b" 0x15 : "a+b" 各モードの内容は以下の通りです。 "r" : 読み出し用にオープンします。 "w" : 空ファイルを書き込み用にオープンします。 "a" : ファイルの最後から書き込み用にオープンします。 "r+" : 読み出し, 書き込み用にオープンします。 "w+" : 空ファイルを読み出し, 書き込み用にオープンします。 "w+" : 空ファイルを読み出し, 書き込み用にオープンします。 "a+" : 読み出し追加用にオープンします。 "b" : バイナリ・モードでオープンします。
ファイル名先頭アドレス(入力)	ファイル名を格納している領域の先頭アドレス

[詳細説明]

- [FOPEN] によってファイルをオープンすると、ファイル番号を返します。 以後のファイル入出力、ファイルのクローズ等では、このファイル番号を用います。
- 同時にオープンできる最大ファイル数は256 です。

RENESAS

fileope	en:		
	MOV.L MOV.L MOV.L MOV.B MOV.L MOV.L JSR	R2,R5 #PARM,R2 R1,4h:5[R2] R5,2h:5[R2] #01250000h,R1 #SIM_IO,R5 R5	; R2(オープン・モード)を R5 に設定 ; パラメータ・ブロックのアドレスを R2 に設定 ; R1(ファイル名先頭アドレス)を R2 + 4 バイトに設定 ; R5 を R2 + 2 バイト(オープン・モード)に設定 ; FOPEN の関数コードを R1 に設定 ; システム・コールのアドレスを R5 に設定 ; システム・コール
	NOP MOV.L MOV.B MOV.B RTS	#PARM,R2 1h:5[R2],R1 R1,[R3] [R2],R1	; パラメータ・ブロックを R2 に設定 ; R2 + 1 バイト(ファイル番号)を R1 に設定 ; R1 を R3(ファイル番号のポインタ)の指す先に設定 ; R2 の先頭 1 バイト(実行結果)を R1 に設定 ; 呼び出し元のアドレスに戻る
PARM:	.SECTION .BLKL	B,DATA,ALIGN=4 2	; パラメータ・ブロック領域



FCLOSE

ファイルをクローズします。

1バイト	1バイト	1バイト	1バイト
0x01	0x06	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	<i>実行結果</i>	ファイル番号

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:エラー
ファイル番号(入力)	ファイルのオープン時に返す番号

[詳細説明]

- ファイルをクローズします。

_fileclos	e:			
М	IOV.L	#PARM,R2	;	パラメータ・ブロックのアドレスを R2 に設定
М	IOV.B	R1,1h:5[R2]	;	R1 (ファイル番号)を R2 + 1 バイトに設定
М	IOV.L	#01060000h,R1	;	FCLOSE の関数コードを R1 に設定
М	IOV.L	#SIM_IO,R3	;	システム・コールのアドレスを R3 に設定
J	SR	R3	;	システム・コール
М	IOV.L	#PARM,R2	;	パラメータ・ブロックを R2 に設定
М	IOV.B	[R2],R1	;	R2 の先頭 1 バイト(実行結果)を R1 に設定
R	TS		;	呼び出し元のアドレスに戻る
	.SECTION E	8,DATA,ALIGN=4		
PARM:	.BLKL	1	;	パラメータ・ブロック領域



FGETC

ファイルから1バイトのデータを読み出します。

[機能コード(R1 レジスタ)]

1バイト	1バイト	1バイト	1バイト
0x01	0x27	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト	
+0	実行結果	ファイル番号	
+2 +4	未使用		
+6	入力バッファ	先頭アドレス	

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:EOF 検出
ファイル番号(入力)	ファイルのオープン時に返す番号
<i>入力バッファ先頭アドレス</i> (入力)	入力データを書き込むバッファの先頭アドレス

[詳細説明]

- ファイルから1バイトのデータを読み出します。

_fcharg	et:		
	MOV.L	R2,R5	; R2(ファイル番号)を R5 に設定
	MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
	MOV.L	R1,4h:5[R2]	; R1(入力バッファ)を R2 + 4 バイトに設定
	MOV.B	R5,1h:5[R2]	; R5 を R2 + 1 バイト(ファイル番号)に設定
	MOV.L	#01270000h,R1	; FGETC の関数コードを R1 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	NOP		
	MOV.L	#PARM,R1	; パラメータ・ブロックを R1 に設定
	MOV.B	[R1],R1	; R1 の先 頭 1 バイト(実行結果)を R1 に設定
	RTS		; 呼び出し元のアドレスに戻る
	.SECTION	B,DATA,ALIGN=4	
PARM:	.BLKL	2	; パラメータ・ブロック領域



FPUTC

ファイルへ1バイトのデータを書き込みます。

[機能コード(R1 レジスタ)]

1バイト	1バイト	1バイト	1バイト
0x01	0x28	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	<i>実行結果</i>	ファイル番号
+2	未使用	
+4 +6	出力バッファ	<i>「先頭アドレス</i>

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:エラー
ファイル番号(入力)	ファイルのオープン時に返す番号
<i>出力バッファ先頭アドレス</i> (入力)	出力データを格納しているバッファの先頭アドレス

[詳細説明]

- ファイルへ1バイトのデータを書き込みます。

fcharp	ut:		
_	MOV.L	R2 , R5	; R2(ファイル番号)を R5 に設定
	MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
	MOV.L	#IO_BUF,R4	; 出力バッファ先頭アドレスを R4 に設定
	MOV.L	R4,4h:5[R2]	; R4(出力バッファ)を R2+4 バイトに設定
	MOV.B	R1,[R4]	; R1(出力文字)を R4 の指す先(出力バッファ)に設定
	MOV.B	R5,1h:5[R2]	; R5 を R2+1 バイト(ファイル番号)に設定
	MOV.L	#01280000h,R1	; FPUTC の関数コードを R1 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	MOV.L	#PARM,R1	; パラメータ・ブロックのアドレスを R1 に設定
	MOV.B	[R1],R1	; R1 の先頭 1 バイト(実行結果)を R1 に設定
	RTS		; 呼び出し元のアドレスに戻る
	.SECTION	B,DATA,ALIGN=4	
PARM:	.BLKL	2	; パラメータ・ブロック領域
	.SECTION	B 1,DATA	· · · · · · · · · · · · · · · · · · ·
IO_BUF:	.BLKL	1	; 入出カバッファ



FEOF

ファイルの終端をチェックします。

1バイト	1バイト	1バイト	1バイト
0x01	0x0B	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	実行結果	ファイル番号

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0 : EOF でない -1 : EOF 検出
ファイル番号(入力)	ファイルのオープン時に返す番号

[詳細説明]

- ファイルの終端をチェックします。

_fpe	of:		
	MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
	MOV.B	R1,1h:5[R2]	; R1(ファイル番号)を R2 + 1 バイトに設定
	MOV.L	#010B0000h,R1	; FEOF の関数コードを R1 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	MOV T.	#PARM R2	・ パラメータ・ブロックを R2 に設定
	MOV.B	[R2].R1	; R2 の先頭 1 バイト(実行結果)をR1 に設定
	RTS	[1],1.1	; 呼び出し元のアドレスに戻る
	OF CHILDREN		
	.SECTION	B, DATA, ALIGN=4	
PARM	: .BLKL	1	; ハフメータ・フロック領域



FSEEK

ファイル・ポインタを指定された位置に移動します。

[機能コード(R1 レジスタ)]

1バイト	1バイト	1バイト	1バイト
0x01	0x0C	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	実行結果	ファイル番号
+2 +4	ディレクション	未使用
+6	オフセット	

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:エラー
ファイル番号(入力)	ファイルのオープン時に返す番号
ディレクション (入力)	0 : オフセットはファイルの先頭からのバイト数 1 : オフセットは現在のファイル・ポインタからのバイト数 2 : オフセットはファイルの最後尾からのバイト数
オフセット (入力)	ディレクションで指定した位置からのバイト数

[詳細説明]

- ファイル・ポインタを指定された位置に移動します。



1		
_fpseek:		
MOV.L	R2,R5	; R2(オフセット)を R5 に設定
MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
MOV.L	R5,4h:5[R2]	; R5 を R1 + 4 バイト(オフセット)に設定
MOV.B	R1,1h:5[R2]	; R1(ファイル番号)を R1 + 1 バイトに設定
MOV.B	R3,2h:5[R2]	; R3(ディレクション)を R2 + 2 バイト(ディレクション)
		; に設定
MOV.L	#010C0000h,R1	; FSEEK の関数コードを R1 に設定
MOV.L	#SIM_IO,R5	; システム・コールのアドレスを R5 に設定
JSR	R5	; システム・コール
NOP		
MOV.L	#PARM,R1	; パラメータ・ブロックのアドレスを R1 に設定
MOV.B	[R1],R1	; R1 の先頭 1 バイト(実行結果)を R1 に設定
RTS		; 呼び出し元のアドレスに戻る
.SECTIO	N B, DATA, ALIGN=4	
PARM: .BLK	IL 2	; パラメータ・ブロック領域



FTELL

ファイル・ポインタの現在位置を取得します。

```
[機能コード(R1 レジスタ)]
```

1バイト	1バイト	1バイト	1バイト
0x01	0x0D	0x00	0x00

[パラメータ・ブロック(R2 レジスタ)]

	1バイト	1バイト
+0	実行結果	ファイル番号
+2 +4	未使用	
+6	オフセット	

[パラメータ]

パラメータ	説明
<i>実行結果</i> (出力)	0:正常終了 -1:エラー
ファイル番号(入力)	ファイルのオープン時に返す番号
オフセット (出力)	現在のファイル・ポインタの位置(ファイル先頭からのバイト数)

[詳細説明] ファイル・ポインタ

- の現在位置を取得します。

_fptell	:		
	MOV.L	R2 , R5	; R2(オフセット)を R5 に設定
	MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
	MOV.B	R1,1h:5[R2]	; R1(ファイル番号)を R2 + 1 バイトに設定
	MOV.L	#010D0000h,R1	; FTELL の関数コードを R1 に設定
	MOV.L	#SIM_IO,R3	; システム・コールのアドレスを R3 に設定
	JSR	R3	; システム・コール
	NOP		
	MOV.L	#PARM,R2	; パラメータ・ブロックのアドレスを R2 に設定
	MOV.L	4h:5[R2],R1	; R2 + 4 バイト(新しいオフセット値)を R1 に設定
	MOV.L	R1,[R5]	; R1 を R5(オフセットのポインタ)の先に設定
	MOV.B	[R2] , R1	; R2 の先 頭 1 バイト(実行結果)を R1 に設定
	RTS		; 呼び出し元のアドレスに戻る
	.SECTION	B,DATA,ALIGN=4	
PARM:	.BLKL	2	; パラメータ・ブロック領域



改訂記録

Rev.	発行日	改定内容	
		ページ	ポイント
1.00	2021.06.01	-	初版発行

CS+ V8.06.00 ユーザーズマニュアル RX デバッグ・ツール編				
発行年月日	2021年6月1日 Rev.1.00			
発行	ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24(豊洲フォレシア)			

CS+ V8.06.00

