



# DTU8 ユーザーズマニュアル

---

プログラム開発支援ソフトウェア

## ご注意

本資料の一部または全部をラピスセミコンダクタの許可なく、転載・複製することを堅くお断りします。

本資料の記載内容は改良などのため予告なく変更することがあります。

本資料に記載されている内容は製品のご紹介資料です。ご使用にあたりましては、別途仕様書を必ずご請求のうえ、ご確認ください。

本資料に記載されております応用回路例やその定数などの情報につきましては、本製品の標準的な動作や使い方を説明するものです。したがって、量産設計をされる場合には、外部諸条件を考慮していただきますようお願いいたします。

本資料に記載されております情報は、正確を期すため慎重に作成したのですが、万が一、当該情報の誤り・誤植に起因する損害がお客様に生じた場合においても、ラピスセミコンダクタはその責任を負うものではありません。

本資料に記載されております技術情報は、製品の代表的動作および応用回路例などを示したものであり、ラピスセミコンダクタまたは他社の知的財産権その他のあらゆる権利について明示的にも黙示的にも、その実施または利用を許諾するものではありません。上記技術情報の使用に起因して紛争が発生した場合、ラピスセミコンダクタはその責任を負うものではありません。

本資料に掲載されております製品は、一般的な電子機器 (AV 機器、OA 機器、通信機器、家電製品、アミューズメント機器など) への使用を意図しています。

本資料に掲載されております製品は、「耐放射線設計」はなされていません。

ラピスセミコンダクタは常に品質・信頼性の向上に取り組んでおりますが、種々の要因で故障することもあり得ます。

ラピスセミコンダクタ製品が故障した際、その影響により人身事故、火災損害等が起こらないようご使用機器でのデイレティング、冗長設計、延焼防止、フェイルセーフ等の安全確保をお願いします。定格を超えたご使用や使用上の注意書が守られていない場合、いかなる責任もラピスセミコンダクタは負うものではありません。

極めて高度な信頼性が要求され、その製品の故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのある機器・装置・システム (医療機器、輸送機器、航空宇宙機、原子力制御、燃料制御、各種安全装置など) へのご使用を意図して設計・製造されたものではありません。上記特定用途に使用された場合、いかなる責任もラピスセミコンダクタは負うものではありません。上記特定用途への使用を検討される際は、事前にローム営業窓口までご相談願います。

本資料に記載されております製品および技術のうち「外国為替及び外国貿易法」に該当する製品または技術を輸出する場合、または国外に提供する場合には、同法に基づく許可が必要です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。また、その他の製品名や社名などは、一般に商標または登録商標です。

Copyright 2008 - 2012 LAPIS Semiconductor Co., Ltd.

---

## ラピスセミコンダクタ株式会社

〒193-8550 東京都八王子市東浅川町 550 番地 1  
<http://www.lapis-semi.com/jp/>

# 目 次

<b>1 章 イントロダクション</b> .....	<b>1</b>
1.1 製品について .....	1
1.1.1 概要 .....	1
1.1.2 特徴 .....	2
1.1.3 Dr.U8 ICE / Dr.U16 ICEモードの特徴 .....	3
1.1.4 シミュレーションモードの特徴 .....	3
1.2 各モードのメニュー .....	3
1.3 マニュアルの構成 .....	6
1.4 表記法 .....	7
<b>2 章 起動の前に</b> .....	<b>8</b>
2.1 動作環境 .....	8
2.2 対応オブジェクトフォーマット .....	8
<b>3 章 起動とロード</b> .....	<b>10</b>
3.1 ハードウェア環境の確認 .....	10
3.2 DTU8 デバッガの起動と起動オプション .....	10
3.2.1 ターゲット設定ダイアログボックス .....	11
3.2.2 uEASEモードにおける起動時の確認ダイアログ .....	12
3.3 メモリマッピングの確認と変更 .....	13
3.4 プログラムファイルの読み込み .....	15
3.4.1 プログラムファイルの読み込みダイアログボックス .....	15
3.4.2 再ロード .....	16
3.4.3 注意事項 .....	17
3.5 プロジェクトファイル .....	19
<b>4 章 基本操作</b> .....	<b>21</b>
4.1 画面とメニュー .....	21
4.2 ウィンドウ .....	22
4.2.1 ウィンドウの表示 .....	22
4.2.2 ウィンドウの複製 .....	23
4.2.3 ウィンドウ領域のダブルクリック .....	24

4.3	その他 .....	24
4.3.1	ツールバー .....	24
4.3.2	ショートカットキー .....	24
4.3.3	その他 .....	25
<b>5</b>	<b>ソースレベルとシンボリックデバッグ .....</b>	<b>27</b>
5.1	ソースウィンドウ .....	27
5.1.1	表示 .....	27
5.1.2	ソースウィンドウ上での変数値表示 .....	27
5.1.3	ソースファイルの検索順序 .....	28
5.1.4	表示ファイルの切り替え .....	29
5.1.5	ソースウィンドウのプロパティ .....	30
5.2	逆アセンブルウィンドウ .....	31
5.2.1	表示 .....	31
5.2.2	逆アセンブルウィンドウの文字列検索 .....	31
5.2.3	逆アセンブルウィンドウのプロパティ .....	33
5.3	シンボリックデバッグ .....	34
5.4	ウィンドウ間のリンク .....	35
5.5	PCの追跡モード .....	36
<b>6</b>	<b>プログラム実行機能 .....</b>	<b>37</b>
6.1	実行モード .....	37
6.1.1	連続実行(エミュレーション実行) .....	37
6.1.2	連続実行(シミュレーション実行) .....	37
6.1.3	カーソルまで実行 .....	37
6.1.4	リセット後に実行 .....	38
6.1.5	ステップイン実行 .....	38
6.1.6	ステップオーバー実行 .....	38
6.1.7	ステップアウト実行 .....	38
6.1.8	Cソースレベルのステップ実行 .....	39
6.2	リセット .....	40
6.3	プログラム実行中に使用可能な機能について .....	41
6.3.1	ウィンドウ .....	41
6.3.2	メニュー .....	42
6.4	実行時間の計測 .....	43
6.5	エミュレーション実行中の終了と再接続 .....	44
6.6	サイクルカウンタ .....	45

6.7	割り込みシミュレーション .....	46
6.7.1	割り込みウィンドウ .....	46
<b>7 章</b>	<b>ブレーク機能 .....</b>	<b>49</b>
7.1	ブレークポイント .....	49
7.1.1	ソース／逆アセンブルウィンドウ上での設定方法 .....	49
7.1.2	一覧表示 .....	50
7.1.3	ブレークポイントブレーク全体の有効化と無効化 .....	51
7.1.4	Cソース行に設定するブレークポイント .....	51
7.1.5	ブレークポイント設定時の注意事項 .....	52
7.2	強制ブレーク .....	53
7.3	ブレーク条件の設定 .....	53
<b>8 章</b>	<b>メモリとレジスタの参照と変更 .....</b>	<b>56</b>
8.1	プログラムカウンタ .....	56
8.2	レジスタ .....	57
8.3	SFR .....	59
8.4	コードメモリ、データメモリ、物理セグメント1以上のメモリ .....	59
8.5	コード領域の保存と比較 .....	63
8.5.1	コード領域の保存 .....	63
8.5.2	エミュレータのコード情報とHEXファイルの比較 .....	65
8.6	データ領域の保存と復帰 .....	66
8.6.1	データ領域の保存 .....	66
8.6.2	データ領域の復帰 .....	67
8.7	リアルタイムRAMモニタ .....	68
<b>9 章</b>	<b>トレース機能 .....</b>	<b>69</b>
9.1	トレースウィンドウ .....	69
9.2	表示モード .....	70
9.3	トレースカウンタの表示 .....	71
9.4	トレースデータの保存 .....	72
9.5	トレースデータの検索 .....	73
9.6	トレースメモリのクリア .....	74
9.7	トレーストリガ .....	75
9.8	その他 .....	76
9.8.1	ソースウィンドウ／逆アセンブルウィンドウへのジャンプ .....	76

9.8.2 指定行へのジャンプ .....	76
<b>10 章 ウォッチ機能 .....</b>	<b>77</b>
10.1 使い方 .....	77
10.2 登録方法 .....	78
10.3 ウォッチウィンドウ上でのデータの変更 .....	80
10.4 ウォッチアイテムリストの保存 .....	81
10.5 ウォッチアイテムリストの読み込み .....	81
10.6 ウォッチアイテムの表示形式の変更, 削除, 移動 .....	81
10.7 リアルタイムウォッチ .....	82
10.7.1 リアルタイムウォッチの表示オプション .....	83
<b>11 章 その他の機能 .....</b>	<b>84</b>
11.1 カバレッジ機能 .....	84
11.1.1 使い方 .....	84
11.1.2 網羅率の表示 .....	85
11.2 マクロ機能 .....	86
11.2.1 使い方 .....	86
11.2.2 単一スクリプトコマンドの実行機能 .....	88
11.3 ログ機能 .....	88
11.3.1 ログウィンドウの使い方 .....	88
11.4 各ウィンドウのカスタマイズ .....	89
11.5 オンラインヘルプ .....	92
11.6 バージョン情報の表示 .....	93
11.7 ファームウェア・アップデート機能 .....	94
11.8 対応機種変更機能 .....	94
11.9 リアルタイムLCDモニタ機能 .....	95
11.9.1 リアルタイムLCDモニタの使い方 .....	95
<b>12 章 付録 .....</b>	<b>97</b>
12.1 シンボリックデバッグと入力形式 .....	97
12.1.1 シンボル .....	97
12.1.2 式 .....	98
12.1.3 変更値リスト .....	99
12.2 ウォッチウィンドウに登録可能な項目 .....	100
12.2.1 ローカル変数の登録について .....	100

12.2.2 グローバル変数の登録について .....	100
12.2.3 関数引数の登録について .....	100
12.2.4 C 変数の型表示について .....	101
12.2.5 C と ASM の混在プログラムについて .....	101
12.2.6 レジスタに割り当てられた変数について .....	101
<b>12.3 最適化された C ソースコードのデバッグ .....</b>	<b>102</b>
<b>12.4 マクロスクリプトリファレンス .....</b>	<b>103</b>
12.4.1 スクリプトコマンド一覧 .....	103
12.4.2 プロジェクトに関するコマンド .....	105
12.4.3 CPU の内容を参照／変更するコマンド .....	105
12.4.4 C変数を参照／変更するコマンド .....	106
12.4.5 コードメモリに関するコマンド .....	107
12.4.6 データメモリに関するコマンド .....	109
12.4.7 物理セグメント 1 以上のメモリに関するコマンド .....	110
12.4.8 エミュレーションに関するコマンド .....	111
12.4.9 リセットに関するコマンド .....	112
12.4.10 ブレークに関するコマンド .....	112
12.4.11 トレースに関するコマンド .....	113
12.4.12 パフォーマンス／カバレッジに関するコマンド .....	113
12.4.13 マクロの実行に関するコマンド .....	114
12.4.14 シンボルの表示／変更／削除に関するコマンド .....	115
12.4.15 その他のコマンド .....	116
<b>12.5 エミュレータ接続時のエラーについて .....</b>	<b>117</b>
12.5.1 Dr.U8 ICE / Dr.U16 ICE / Dr.ICE / uEASE / nanoEASE 接続時の共通のエラーに ついて .....	117
12.5.2 uEASE / nanoEASE 接続時のエラー .....	119
12.5.3 Dr.U8 ICE / Dr.U16 ICE / Dr.ICE 接続時のエラー .....	122





# 1 章 イントロダクション

## 1.1 製品について

### 1.1.1 概要

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガは、弊社の CCU8 コンパイラパッケージおよび MACU8 アセンブラパッケージで作成したプログラムを評価、デバッグするためのソフトウェアツールです。弊社オリジナル 8 ビット RISC プロセッサ nX-U8/100 コアまたは 16 ビット RISC プロセッサ nX-U16/100 コアを搭載したマイクロコントローラ用のアプリケーションプログラム開発を支援します。

DTU8 デバッガは、以下のモードをサポートしています。

- ・『Dr.U8 ICE モード』 Dr.U8 ICE インサーキットエミュレータと接続してデバッグ機能を提供
- ・『Dr.U16 ICE モード』 Dr.U16 ICE インサーキットエミュレータと接続してデバッグ機能を提供
- ・『Dr.ICE モード』 Dr.610XXX 製インサーキットエミュレータと接続してデバッグ機能を提供
- ・『uEASE モード』 uEASE オンチップデバッグエミュレータと接続してデバッグ機能を提供
- ・『nanoEASE モード』 nanoEASE オンチップデバッグエミュレータと接続してデバッグ機能を提供
- ・『シミュレーションモード』 DTU8 デバッガ内部の nX-U8/100 コア、nX-U16/100 コアのシミュレーションエンジンを利用することによってシミュレーションデバッグ機能を提供

DTU8 デバッガは、これらの機能によって、マイクロコントローラ用の組込み型アプリケーションプログラムの評価、デバッグを支援します。

本マニュアルでは、Dr.U8 ICE モード、Dr.U16 ICE モード、Dr. ICE モード、uEASE モード、nanoEASE モード、シミュレーションモード、およびそれすべてに関する説明のために、便宜上、次の記号を使用しています。

**Dr.U8 ICE**

Dr.U8 ICE モードに関する内容です。

**Dr.U16 ICE**

Dr.U16 ICE モードに関する内容です。

**Dr. ICE**

Dr. ICE モードに関する内容です。

**uEASE**

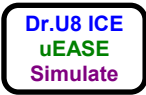
uEASE モードに関する内容です。

**nanoEASE**

nanoEASE モードに関する内容です。

Simulate

シミュレーションモードに関する内容です。

Dr.U8 ICE  
uEASE  
Simulate

モード共通の内容は、上記記号を左記のように組み合わせて表示します。

### 1.1.2 特徴

DTU8 デバッガは、数々の特徴により、高度で使いやすい対話形式のデバッグ環境を提供します。

#### ■ WindowsXP/Vista/7 対応の使いやすい GUI 環境

32 ビットアプリケーションとして作成され、快適な動作環境を提供します。メニュー、ウィンドウ、ダイアログボックス、マウスによる直感的で使いやすい操作感の作業環境を実現しています。

#### ■ ソースレベルデバッグとシンボリックデバッグ機能

C 言語及びアセンブリ言語のソースプログラム上で、ブレークポイント設定、ステップ実行などのソースレベルデバッグが可能です。また、ソースプログラム中で使用した関数名やラベルシンボルなどを、数値の代わりに指定することができます。

さらに、DTU8 デバッガのソースウィンドウは、逆アセンブルウィンドウやトレースウィンドウともリンクしているため、命令コードレベルでの評価作業にも対応しています。

#### ■ 強力なウォッチ機能

C 言語の構造体、共用体、ポインタ型も完全にサポートする強力なウォッチ機能です。

アセンブリ言語レベルにおいても、nX-U8 コアのアドレッシングモードの記述をサポートし、ベース付レジスタ間接参照まで登録可能です。

#### ■ プロジェクト機能

各種の設定状態をプロジェクトとして保存することにより、簡単に設定状態を復帰でき、複数のデバッグ環境設定を切り替えて使用することが可能になります。

#### ■ マクロ実行機能

スクリプト言語で記述したテキスト形式のファイルを読み込んで実行することができます。同じ操作を繰り返す処理や、長時間に渡る機械的な操作などのバッチ処理的なものをこのマクロ機能で実現できます。

#### ■ その他の機能

プログラムファイルとエミュレータメモリの比較

パッチプログラミングに便利な内蔵ラインアセンブラ

内部 RAM データの保存機能

オンラインヘルプ

### 1.1.3 Dr.U8 ICE / Dr.U16 ICE モードの特徴

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

DTU8 デバッガの Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モードには、次の特徴があります。

#### ■ エミュレーション実行中の終了/再接続

エミュレーション実行中に、DTU8 デバッガを終了して、その後、再接続が可能です。

数日にも渡るエーシング評価時に便利です。

### 1.1.4 シミュレーションモードの特徴

Simulate

DTU8 デバッガのシミュレーションモードには、次の特徴があります。

#### ■ nX-U8/100 コアのシミュレーション機能

DTU8 デバッガのシミュレーションエンジンによって、nX-U8/100 コアの命令をシミュレーションできます。DTU8 単体でプログラムの実行が可能で、インサーキットエミュレータを接続する必要もありません。

## 1.2 各モードのメニュー

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガは、いろいろなタイプのエミュレーションシステムの一部として使用されています。このため、DTU8 デバッガは、接続したエミュレーションシステムに応じて、メニューやダイアログボックスを自動的に変化させるようになっており、DTU8 デバッガを使用する際にその違いについて特に意識する必要はありません。

本マニュアルでは、DTU8 デバッガで用意している機能全般を説明しますが、エミュレーションシステムによってはサポートされない機能もあります。サポートされない機能については、メニューに表示されません。各モードのメニュー一覧を以下に示しますので、本マニュアルに記載されている機能のうち、使用対象のモードで使用できない機能については読み飛ばしてください。

メニュー	メニュー内容	Dr.U8 ICE	Dr.U16 ICE	Dr.ICE	uEASE	nano EASE	Simulator
ファイル	プロジェクトを開く	●	●	●	●	●	●
	プロジェクトを保存	●	●	●	●	●	●
	名前を付けてプロジェクトを保存	●	●	●	●	●	●
	プログラムファイルの読み込み	●	●	●	●	●	●
	プログラムファイルの保存	●	●	●	●	●	●
	プログラムファイルの比較	●	●	●	●	●	●
	RAM データファイルの読み込み	●	●	●	●	●	●
	RAM データファイルの保存	●	●	●	●	●	●
	ソースファイルの選択	●	●	●	●	●	●
	DTU8 の終了	●	●	●	●	●	●

## 1章 イントロダクション

メニュー	メニュー内容	Dr.U8 ICE	Dr.U16 ICE	Dr.ICE	uEASE	nano EASE	Simulator
編集	元に戻す	●	●	●	●	●	●
	切り取り	●	●	●	●	●	●
	コピー	●	●	●	●	●	●
	貼り付け	●	●	●	●	●	●
	削除	●	●	●	●	●	●
	すべてを選択	●	●	●	●	●	●
	検索	●	●	●	●	●	●
	置換	●	●	●	●	●	●
表示	ソース	●	●	●	●	●	●
	逆アセンブル	●	●	●	●	●	●
	ステータス	●	●	●	●	●	●
	レジスタ	●	●	●	●	●	●
	SFR	●	●	●	●	●	●
	コードメモリ	●	●	●	●	●	●
	データメモリ	●	●	●	●	●	●
	物理セグメント 1 以上のメモリ	●	●	●	●	●	●
	トレース	●	●	●	—	—	●
	ウォッチ	●	●	●	●	●	●
	リアルタイム RAM モニタ	●	●	—	—	—	—
	ログ	●	●	●	●	●	●
	割り込み	—	—	—	—	—	●
	ツール バー	●	●	●	●	●	●
	ステータス バー	●	●	●	●	●	●
実行	リセット	●	●	●	●	●	●
	プログラムカウンタの変更	●	●	●	●	●	●
	デバイスリセット	—	—	—	●	●	—
	プログラム実行	●	●	●	●	●	●
	カーソルまで実行	●	●	●	●	●	●
	リセット後に実行	●	●	●	●	●	●
	強制ブレーク	●	●	●	●	●	●
	ステップ オーバー	●	●	●	●	●	●
	ステップ イン	●	●	●	●	●	●
	ステップ アウト	●	●	●	●	●	●
	ブレークポイントの設定/解除	●	●	●	●	●	●
	ブレークポイントリスト	●	●	●	●	●	●
	全ブレークポイントの解除	●	●	●	●	●	●
	ブレークポイントブレークを有効にする	●	●	●	●	●	●
	ブレーク条件の設定	●	●	●	●	●	●
	トレーススタートトリガの設定	●	●	—	—	—	—
	シグアウトポイントの追加	—	●	●	—	—	—
	シグアウトポイントリスト	—	●	●	—	—	—
	全シグアウトポイントの解除	—	●	●	—	—	—
	シグアウト信号のリセット	—	●	●	—	—	—

メニュー	メニュー内容	Dr.U8 ICE	Dr.U16 ICE	Dr.ICE	uEASE	nano EASE	Simulator
ツール	マクロの実行	●	●	●	●	●	●
	マクロ実行の停止	●	●	●	●	●	●
	スクリプトコメントの実行	●	●	●	●	●	●
	シンボルリスト	●	●	●	●	●	●
	ウォッチ項目の追加	●	●	●	●	●	●
	リアルタイム LCD モニタ呼び出し	●	●	—	—	—	—
	動作設定	●	●	●	●	●	●
	環境設定	●	●	●	●	●	●
ウィンドウ	アクティブウィンドウを閉じる	●	●	●	●	●	●
	すべて閉じる	●	●	●	●	●	●
	ウィンドウの複製	●	●	●	●	●	●
	重ねて表示	●	●	●	●	●	●
	上下に並べて表示	●	●	●	●	●	●
	左右に並べて表示	●	●	●	●	●	●
	最新の状態に更新	●	●	●	●	●	●
ヘルプ	トピックの検索	●	●	●	●	●	●
	システム情報	●	●	●	●	●	●
	ファームウェアのアップデート	●	●	—	●	●	—
	ICE の機種変更	●	●	—	—	—	—
	バージョン情報	●	●	●	●	●	●

## 1.3 マニュアルの構成

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

このマニュアルの各章の概要は、以下のとおりです。<sup>1</sup>

### ■ 1章 イントロダクション

この章です。

### ■ 2章 起動の前に

DTU8 デバッガを使用するために必要な環境とインストール方法について説明します。

### ■ 3章 起動とロード

DTU8 デバッガの起動方法、プログラム実行のために必要な操作、プロジェクトファイルの構成について説明します。

### ■ 4章 基本操作

DTU8 デバッガの基本的なメニュー操作やウィンドウ操作について説明します。

### ■ 5章 ソースレベルとシンボリックデバッグ

ソースウィンドウを使用したソースレベルデバッグ方法と、逆アセンブルウィンドウを使用したシンボリックデバッグについて説明します。

### ■ 6章 プログラム実行機能

プログラム実行およびプログラム実行に関連深い機能について説明します。

### ■ 7章 ブレーク機能

プログラム実行をブレークさせるための各種ブレーク条件を説明します。

### ■ 8章 メモリとレジスタの参照と変更

プログラムカウンタや各種メモリ、各種レジスタの内容を参照および変更するための操作について説明します。

### ■ 9章 ウォッチ機能

ウォッチウィンドウの使い方やウォッチアイテムの登録方法、その他の操作について説明します。

### ■ 10章 トレース機能

プログラムの実行経過を確認するために使用するトレース機能について説明します。

### ■ 11章 その他の機能

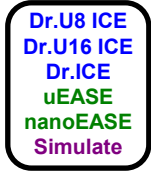
マクロ機能やログ機能、オンラインヘルプなどについて説明します。

### ■ 12章 付録

シンボルや式の入力形式、ウォッチウィンドウの登録アイテム、マクロのスクリプトコマンドおよびエミュレータ接続時のエラーについて説明します。

<sup>1</sup> ユーザーズマニュアルは、チュートリアルとして機能別の章立てで説明しています。メニューとダイアログボックスから機能を探したい場合には、オンラインヘルプを利用してください。また、マニュアルやオンラインヘルプへの記載が間に合わない事柄や特記事項などを説明するオンラインドキュメントも付属しています。

## 1.4 表記法



キー操作は、キーの刻印をカギ括弧で囲み、同時に押すキーがあれば、プラス(+)で連結しています。例えば、文章中に[Shift]+[F1]と記述されているときには、Shift キーを押しながら、F1 キーを押すことを表わしています。

## 2 章 起動の前に

---

### 2.1 動作環境

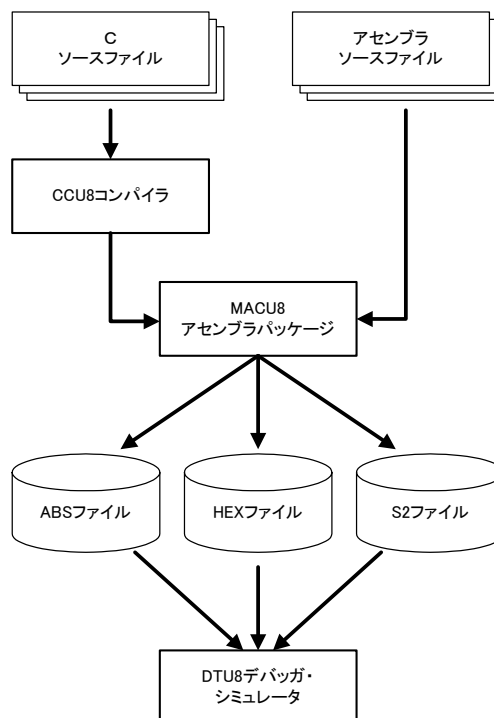
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

#### ■ 動作環境

- WindowsXP/Vista\*/7\*が動作しているパソコン  
\*32bit 版/64bit 版
- 上記オペレーティングシステムが、快適に動作する環境
- SVGA (800×600)以上のグラフィックアダプタとディスプレイ
- 20MB 以上のハードディスクの空き容量
- CD-ROMドライブ
- マウス
- USB ポート

### 2.2 対応オブジェクトフォーマット

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate





DTU8 デバッガは、CCU8 コンパイラと MACU8 アセンブラパッケージで作成したオブジェクトプログラムをサポートしています。

DTU8 デバッガは、U8 アブソリュートオブジェクトフォーマットファイル、インテル HEX フォーマットおよびモトローラ S2 フォーマットのオブジェクトファイルを読み込むことができます。

### ■ U8 アブソリュートオブジェクトファイル(拡張子:. ABS)

U8 アブソリュートオブジェクトフォーマットは、バイナリ形式フォーマットで、ユーザプログラムのプログラムコード(リロケータブルでない、実行可能なコード)と各種のデバッグ情報が格納されています。

DTU8 デバッガのソースウィンドウにソースファイルを表示させ、ソースウィンドウ上でデバッグを行う場合は、CCU8 コンパイラ、RASU8 アセンブラ、RLU8 リンカに、ソースレベルデバッグ情報を生成するオプションを指定して作成した ABS ファイルをロードする必要があります。

### ■ インテル HEX ファイル(拡張子:. HEX)

インテル HEX 形式は、汎用的な ASCII 形式のフォーマットです。市販の P-ROM ライタなどでは、このフォーマットを必要とすることが多いようです。ただし、この形式は、アドレスが 1M バイトを越えるものを表現できません。

HEX ファイルは、インテル HEX 形式であってもユーザシンボルの情報が入れられるように、独自の仕様拡張を行っています。なお、この仕様拡張は、インテル HEX 形式の仕様で認められているもので、シンボル情報付きのインテル HEX 形式であっても、一般のユーティリティや P-ROM ライタで扱うことができます。

なお、インテル HEX 形式では、ファイル中にユーザシンボル情報を格納しますが、パブリックシンボルのみとなるため、不完全なシンボル情報となります。

### ■ モトローラ S2 ファイル(拡張子:. S)

モトローラ S2 形式は汎用的な ASCII 形式のフォーマットです。市販の P-ROM ライタなどでもこのフォーマットはサポートされています。この形式は、アドレスが 16M バイトまで表現できます。

なお、モトローラ S2 形式では、同じファイル名で拡張子が「.SYM」であるファイルにユーザシンボル情報を格納します。このとき、SYM ファイルにはパブリックシンボルのみが格納され、ローカルシンボルの情報が無い不完全なシンボル情報となります。

## 3 章 起動とロード

---

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE

### 3.1 ハードウェア環境の確認

DTU8 デバッガを Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モードまたは uEASE モードで使用する場合は、まずエミュレータやユーザ応用回路の接続および各種設定を確認してください。チェックすべき項目としては、以下の内容が考えられます。

- ・ エミュレータ本体に供給する電源
- ・ 動作周波数(供給クロック)
- ・ ターゲットデバイスの設定
- ・ ユーザ応用回路とユーザケーブルの接続
- ・ プローブケーブルの接続
- ・ その他

上記のチェック項目は、あくまでも参考であり、ハードウェア環境の設定を確認するためには、各エミュレータに付属のマニュアルを参照してください。

**必ず、DTU8 デバッガを起動する前に、エミュレータの電源を入れて、エミュレータをデバッガと接続可能な状態にしてください。**

また、DTU8 デバッガを終了する場合は、エミュレータの電源を落とす前に、DTU8 デバッガを終了してください。<sup>2</sup>

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### 3.2 DTU8 デバッガの起動と起動オプション

DTU8 デバッガを含む開発ツールパッケージのセットアッププログラムは、[スタート]ボタンに DTU8 デバッガのアイコンを登録します。このアイコンを選択することによって DTU8 デバッガを起動できます。また、DTU8 デバッガの起動パラメータにロードするプログラム名を直接指定することも可能です。

DTU8 デバッガの起動オプションは、以下の書式で指定します。

**DTU8.exe 【 *load\_file* 】**

*load\_file* には、起動直後にロードするプロジェクトファイル、または、プログラムファイルを指定します。

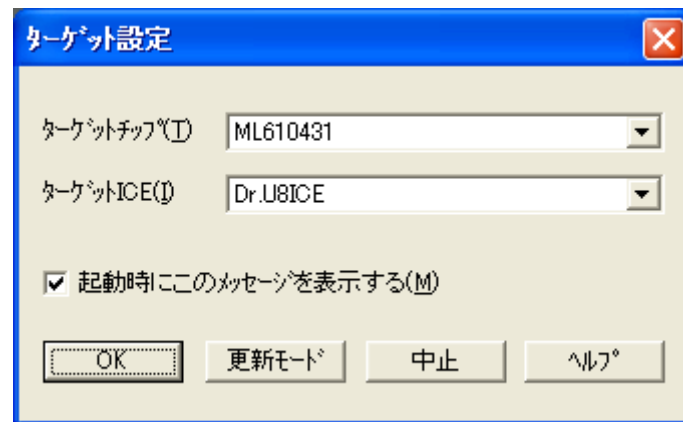
---

<sup>2</sup> DTU8 デバッガは、終了時に、エミュレータと通信を行うため、エミュレータの電源が落とされていると、通信エラーが発生します。

### 3.2.1 ターゲット設定ダイアログボックス

DTU8 を起動すると、以下のようなターゲット設定ダイアログを表示します。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



[ターゲットチップ]フィールドでは、使用するターゲット LSI を選択します。

[ターゲット ICE]フィールドでは、以下の項目が選択できます。

ターゲット ICE	説明
Dr.U8 ICE	Dr.U8 ICE インサーキットエミュレータと接続する場合に選択します
Dr.U16ICE	Dr.U16ICE インサーキットエミュレータと接続する場合に選択します
uEASE	uEASE オンチップデバッグエミュレータと接続する場合に選択します
nanoEASE	nanoEASE オンチップデバッグエミュレータと接続する場合に選択します
SimU8 Simulator	シミュレーションモードで起動する場合に選択します
Dr.ICE	上記以外のインサーキットエミュレータと接続する場合に選択します

[起動時にこのメッセージを表示する]チェックボックスを外すと、次回の起動時に、このダイアログボックスを表示せず、直ちに、DTU8 デバッガが起動します。最後に起動したときの設定から[ターゲット ICE]の設定を変更する場合は、このチェックボックスをチェックしておく必要があります。なお、このチェックボックスの設定は、[ツール]メニューの[環境設定]で開くダイアログボックスの[通信設定]タブでも操作可能です。

[中止]ボタンを押すと、DTU8 デバッガの起動を中止します。

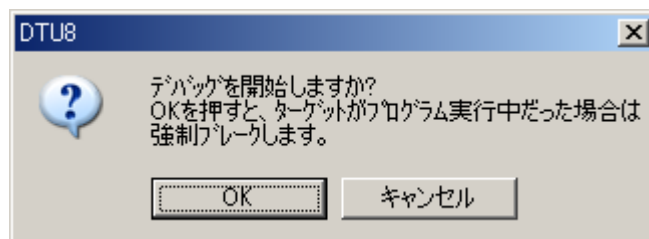
[更新モード]ボタンを押すと、DTU8 デバッガは更新モードで起動します。更新モードでは Dr.U8 ICE および uEASE ファームウェアの更新や対応機種の変更を行うことができます。ファームウェアの更新方法については「11.7 ファームウェア・アップデート機能」を参照してください。対応機種の変更方法については「11.8 対応機種変更機能」を参照してください。

**uEASE  
nanoEASE**

### 3.2.2 uEASE/nanoEASE モードにおける起動時の確認ダイアログ

通信設定ダイアログでターゲット ICE に「uEASE」または「nanoEASE」を選択した場合、以下のダイアログを表示します。

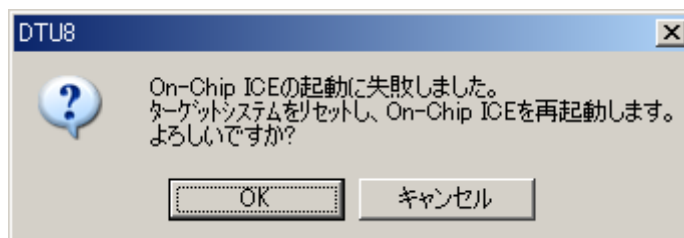
uEASE または nanoEASE は、PC およびターゲット LSI とケーブルで接続された直後はターゲット LSI がプログラム実行中である可能性があるため、その状態からデバッグを開始するかを確認するダイアログを表示します。



ダイアログ上の OK ボタンをクリックすると、uEASE はターゲット LSI 上の On-Chip ICE ブロックを起動し、ターゲットがプログラム実行中だった場合は強制ブレークします。

ダイアログ上のキャンセルボタンをクリックすると、DTU8 を終了します。

On-Chip ICE ブロックの起動または強制ブレークの処理に失敗した場合、以下のようなダイアログを表示します。



ダイアログ上の[OK]ボタンをクリックするとターゲット LSI をリセットし On-Chip ICE を再起動します。ダイアログ上の[キャンセル]ボタンをクリックすると、DTU8 を終了します。

DTU8 を起動したときに、[セキュリティ ID の確認]ダイアログが表示される場合があります。

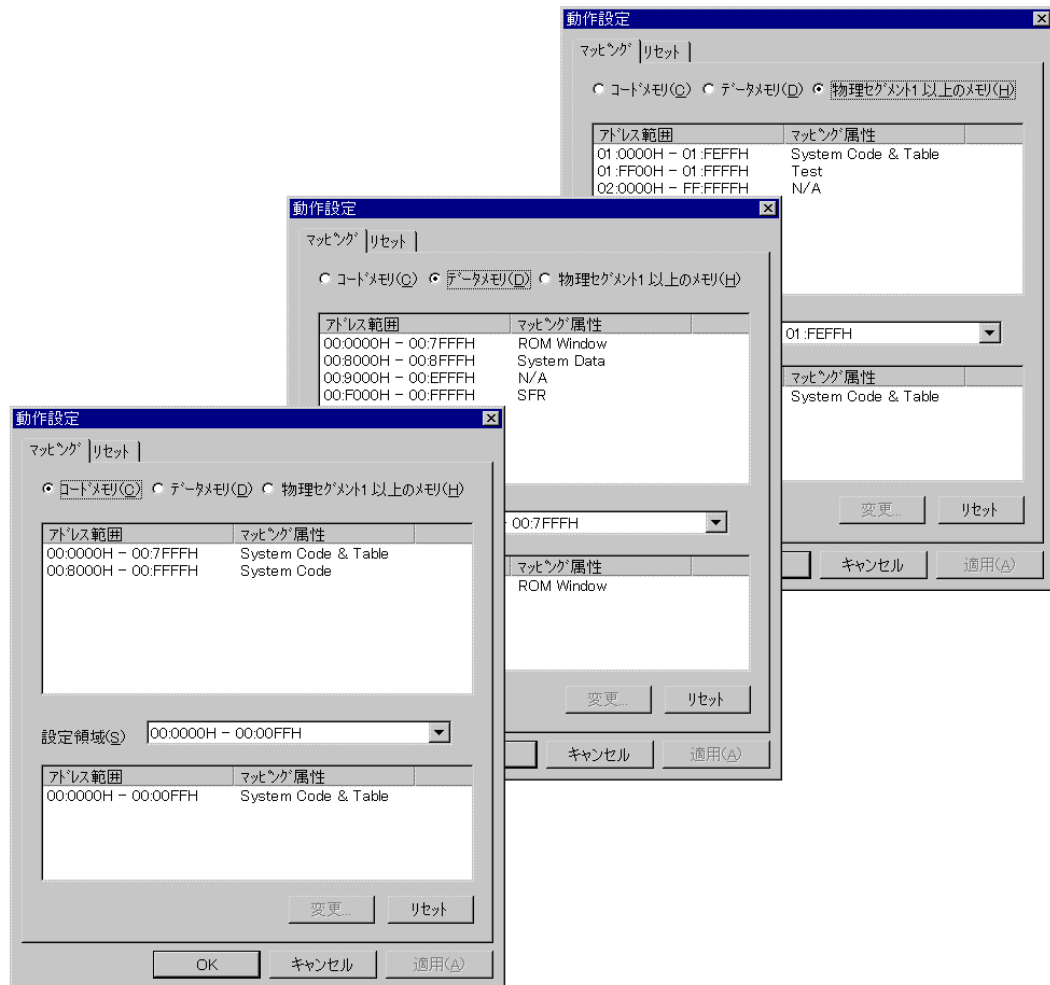
このダイアログが表示された場合には、[フラッシュメモリとセキュリティ設定を初期化する]を選択して[OK]ボタンをクリックしてください。

### 3.3 メモリマッピングの確認と変更

Dr.ICE  
Simulate

メモリマッピングとは、nX-U8/100 コア、nX-U16/100 コアの持つアドレス空間に、メモリの種類を割り付ける作業です。

DTU8 デバッガを起動後、[ツール]メニューの[動作設定]を指定して開くダイアログボックスの[マッピング]タブを選択すると、コードメモリ、データメモリ、および物理セグメント1以上のメモリについて、それぞれのメモリマッピングの確認および変更が可能です。



シミュレーションモードで起動した直後は、設定されているターゲットマイコンのメモリ配置になっています。<sup>3</sup>

各メモリに設定可能なメモリマッピング属性は、エミュレータの設定およびターゲットマイコンに依存します。設定可能なアドレス範囲と属性については、各エミュレータに付属のマニュアルを参照してください。

メモリマッピングを変更する必要がある場合の例として、以下のものがあります。

- 使用していないアドレス領域を N/A にして、ROM N/A area access ブレークや RAM N/A

<sup>3</sup> ただし、プロジェクトファイルを読み込んだ場合は、プロジェクトファイルに保存されたメモリマッピングが復元されます。

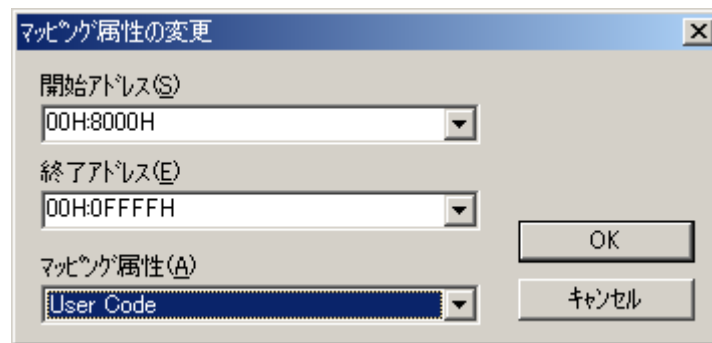
### 3章 起動とロード

area access ブレークを利用して、プログラムの暴走を検知させる

- プログラムコードや使用メモリの最適化の前に、内蔵 ROM や RAM の領域を一時的に増やして、とりあえずプログラムを実行させる。
- ターゲットシステムに置く外部メモリの代わりに、エミュレータ内部のシステムメモリを利用する。特に、外部 ROM を利用するシステムの初期段階では、System Code や System Code & Table を配置して、外部 ROM のコードを評価することになります。

メモリマッピングを変更するには、変更したいメモリの種類(コードメモリ/データメモリ/物理セグメント1以上のメモリ)を選択し、変更したい領域を含む設定領域を選択します。

設定領域を選択すると、その領域の詳細がリストされ、変更したいアドレス範囲に近い項目を選んで、[変更...]ボタンを押すと、以下の[マッピング属性の変更]ダイアログボックスが現れます。



この[マッピング属性の変更]ダイアログボックスで、変更したいアドレス範囲とマッピング属性を指定します。[マッピング属性の変更]ダイアログボックスの[OK]ボタンで、[動作設定]ダイアログボックスの[マッピング]タブで表示しているリストに登録されます。実際に、エミュレータに変更したマッピングを設定するには、[OK]ボタンでダイアログボックスを閉じるか、もしくは、[適用]ボタンを押します。

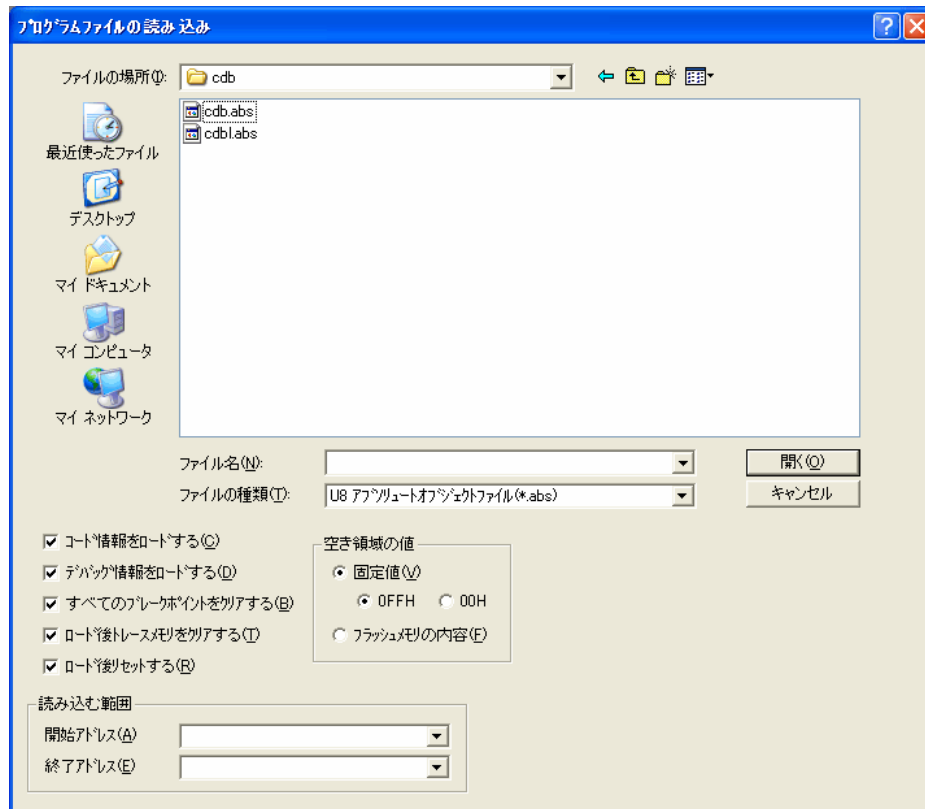
## 3.4 プログラムファイルの読み込み

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### 3.4.1 プログラムファイルの読み込みダイアログボックス

[ファイル]メニューの[プログラムファイルの読み込み]を指定して、デバッグを行うプログラムコードを読み込みます。DTU8 デバッガでは、複数のファイル形式をサポートしますが、アブソリュートオブジェクトファイル(.ABS)を使用することをお勧めします。ソースレベルデバッグを行なうためには、ロードされるファイルの形式が、このアブソリュートオブジェクトファイル(.ABS)形式である必要があります。

[ツール]メニューの[環境設定]で表示されるダイアログボックスでディレクトリタブを選択し、[プログラムファイル]フィールドにアブソリュートオブジェクトファイルの格納されているディレクトリを指定しておけば、プログラム読み込み時にディレクトリを選択する必要がなくなります。



#### ■ プログラムコードを読み込む場合

プログラムコードを読み込む場合は、[コード情報をロードする]チェックボックスをチェックしてください。

#### ■ デバッグ情報を読み込む場合

デバッグ情報とは、ソースレベルデバッグとシンボリックデバッグに必要な情報です。このデバッグ情報をロードすると、ソースウィンドウへのソースの表示や、ウォッチウィンドウにアセンブリ言語ソースで使用したラベルシンボルなどを指定することが可能となります。

[デバッグ情報をロードする]チェックボックスをチェックすると、指定されたオブジェクトファイルの

### 3章 起動とロード

デバッグ情報を読み込みます。なお、デバッグ情報を読み込むと、それまでのデバッグ情報がクリアされます。

#### ■ 読み込み終わった後、ブレークポイントをクリアしたい場合

新たなプログラムコードを読み込むとき、リロケートブルなコードであると、設定されているブレークポイントが無意味になります。[すべてのブレークポイントをクリアする]チェックボックスをチェックしておく、プログラムコードを読み込んだ後にブレークポイントがクリアされます。

#### ■ 読み込み終わった後、トレースメモリをクリアしたい場合

[ロード後とトレースメモリをクリアする]チェックボックスをチェックしておく、プログラムコードを読み込んだ後にトレースメモリがクリアされます。

#### ■ 読み込み終わった後、ターゲットチップをリセットしたい場合

[ロード後リセットする]チェックボックスをチェックしておく、プログラムコードを読み込んだ後にターゲットチップがリセットされ、レジスタや PC 等が初期状態に戻ります。

#### ■ 空き領域の値

プログラムコードを読み込む際に空き領域があった場合には、[空き領域の値]で指定した値で埋めます。

[固定値]を選択した場合、指定できる値は、0FFH または 00H のいずれかです。プログラムファイルに含まれるプログラムコード以外のプログラムメモリの全領域(セグメント 1 以上のメモリを含む)が固定値で埋められます。

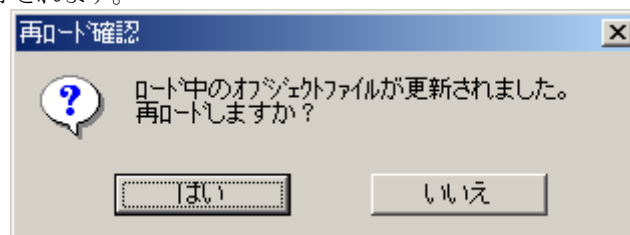
[フラッシュメモリの内容]を選択した場合、空き領域は現在のメモリ内容のまま変わりません。<sup>4</sup>

#### ■ プログラムコードの一部を読み込む場合

ROM テーブルデータを更新する場合や、プログラムコード一部分だけを読み直す必要がある場合には、[開始アドレス]と[終了アドレス]入力フィールドを指定すると、指定したオブジェクトファイルの一部のコードだけを読み込ませることができます。

### 3.4.2 再ロード

最後にデバッグ情報ありで読み込んだプログラムファイルのタイムスタンプが変化すると、以下の確認メッセージが表示されます。<sup>5</sup>

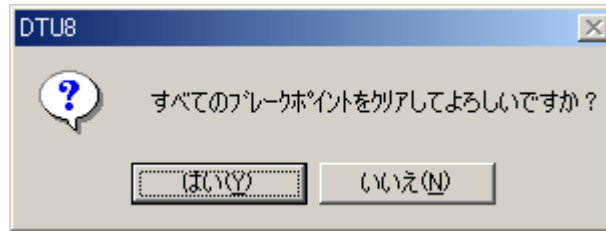


この確認メッセージで、[はい]を指定して、再ロードするときに、ブレークポイントの設定があると、さらに、以下の確認を行います。

<sup>4</sup> uEASE モード、nanoEASE モード、Dr. U8 ICE モード、Dr. U16 ICE モード Dr. ICE モードのときに選択可能です。

<sup>5</sup> このタイムスタンプのチェックは、DTU8 デバッガが非アクティブ (他のアプリケーションを選択) から、アクティブ状態に変化するときに行われます。





これらの機能により、オブジェクトコードのビルドツールとの連携を図っています。

### 3.4.3 注意事項

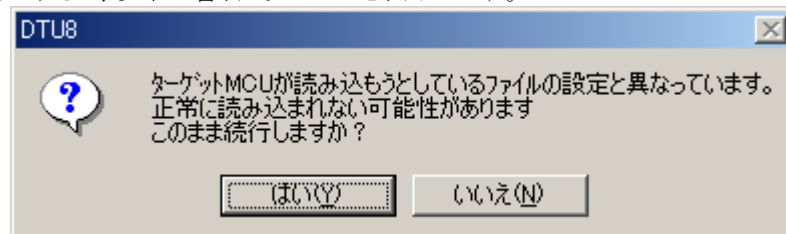
#### ■ ロード可能な領域

プログラムコードは、メモリマッピングが System Code 属性を持つ領域(System Code、System Code & Table、System Code & Data)に設定されている領域に読み込ませることができます。また、NVDATA 属性を持つ領域のコードは、メモリマッピングが NVDATA 属性を持つ領域に読み込ませることができます。これらのマッピング領域以外に読み込ませようとすると、DTU8 デバッガはエラーメッセージを出力し、読み込み処理を中断します。

#### ■ 機種情報

アブソリュートオブジェクトファイル(.ABS)は、そのプログラムファイルを作成時に指定した機種情報(CCU8 コンパイラの /T オプションやアセンブリソースの TYPE 擬似命令)を持っています。

DTU8 デバッガは、アブソリュートオブジェクトファイル(.ABS)が指定されたとき、接続されているエミュレータとアブソリュートオブジェクトファイル(.ABS)の機種情報を比較して、異なる機種のプログラムを読み込もうとすると、以下の警告メッセージを表示します。



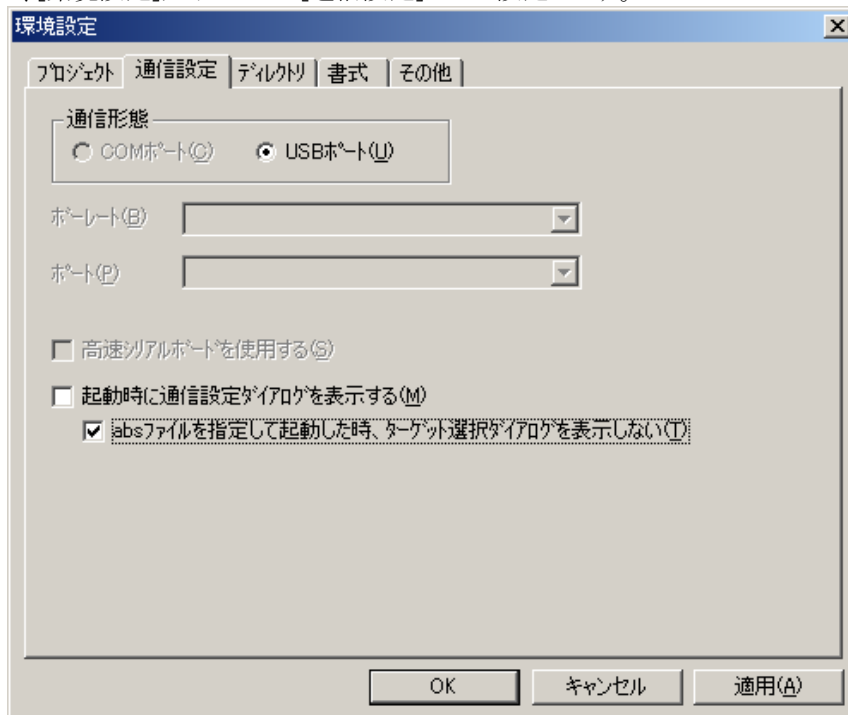
#### ■ ターゲット MCU 選択ダイアログの表示設定

Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モード、uEASE モード、または nanoEASE モードの場合、起動時にターゲット設定ダイアログを表示しない設定にした状態で DTU8 デバッガを起動すると、以下のようなダイアログが表示されます。



この[ターゲット MCU 選択]ダイアログは、アブソリュートオブジェクトファイル(.ABS)指定つきで DTU8 を起動したときのみ表示させないようにすることができます。このダイアログを表示させないよ

うにするには、[環境設定]ダイアログの[通信設定]タブで設定します。



[起動時に通信設定ダイアログを表示する]のチェックを外し、[abs ファイルを指定して起動した時、ターゲット選択ダイアログを表示しない]をチェックすることにより、前述のダイアログを表示しないようにすることができます。

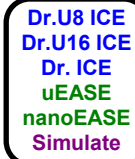
#### ■ 再ロード時のブレークポイント

プログラムを再ロードするときには、基本的にブレークポイントをクリアしてください。

DTU8 デバッガで設定するブレークポイントは、アドレスに関連付けられています。このため、プログラム修正によってソース行のアドレスが変化したプログラムコードを再ロードすると、再ロード前に設定したブレークポイントのアドレスとソース行の対応が崩れます。

ブレークポイントをクリアしない場合として、以下の例があります。

- コード生成に関わらないオプションを変更して、リビルドした場合
- アブソリュートなセグメントの先頭だけにブレークポイントを設定していて、そのアドレスを変更しないような修正を行った場合
- テーブルデータ等の実行アドレスが変化しないプログラム修正の場合
- ブレークポイントをクリアせずに再ロードして、ブレークポイントリストの正当性をオペレータが判断したい場合



## 3.5 プロジェクトファイル

DTU8 デバッガの機能を最大限に活用すると、非常に多くの項目を設定することになります。これらの設定を毎回行うことは、ユーザにとって大きな手間となってしまいます。このため、設定をプロジェクトファイルに保存しておくことにより、次回の起動時に各種設定を容易に復元することができます。<sup>6</sup>

複数の設定をそれぞれ異なったプロジェクトファイルに保存することにより、目的に応じたプロジェクトの使い分けが可能となります。

DTU8 デバッガはそれぞれのターゲット ICE に応じたプロジェクトファイルを使用します。このため、ターゲット ICE の異なるプロジェクトファイルを読み込むことはできませんので、注意してください。したがって、uEASE モード時に保存したプロジェクトファイルを Dr.U8 ICE モード時に読み込むことや、Dr.U8 ICE モード時に保存したプロジェクトファイルを uEASE モード時に読み込むことはできません。

### ■ プロジェクトファイル名

DTU8 デバッガのプロジェクトファイルの拡張子は「.PDT」です。

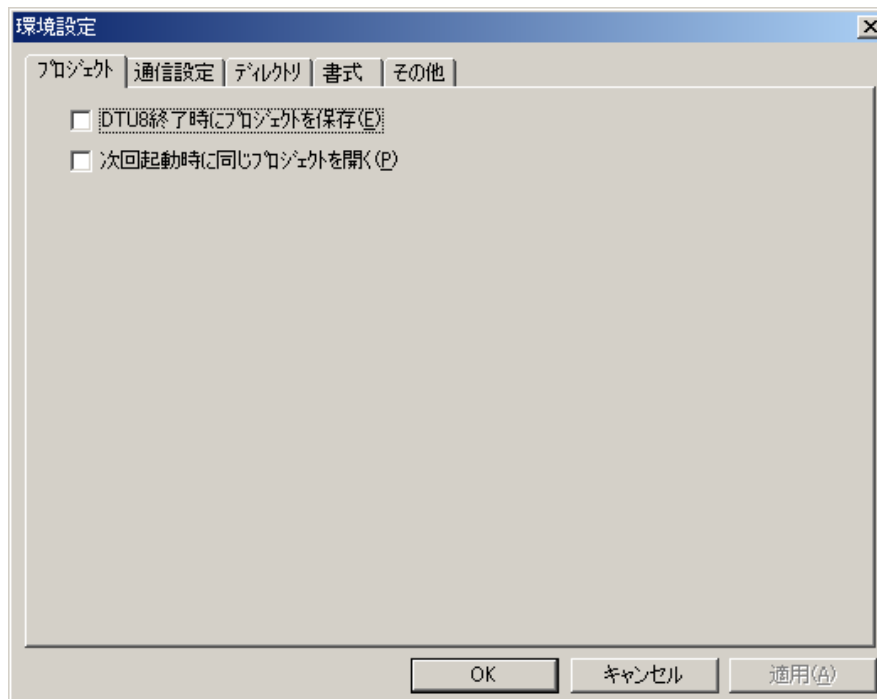
### ■ 保存される項目

- ターゲットマイコン情報
- 最後に読み込んだプログラムファイルとロードオプション
- [ツール]メニューの[動作設定]の内容
  - [メモリマッピング]タブ、[リセット]タブ
- [ツール]メニューの[環境設定]の内容
  - [プロジェクト]タブ、[通信設定]タブ、[ディレクトリ]タブ、[その他]タブ
- デスクトップの状態
- 開いているウィンドウとその配置
- ブレークポイント
- ブレーク設定
- ウォッチ項目
- トレーストリガ設定

<sup>6</sup> プロジェクト機能は、各種の設定状態を保存するものです。ROM、RAM、レジスタなどの内容を保存するものではありません。

#### ■ 環境設定の[プロジェクト]タブ

[ツール]メニューの[環境設定]の[プロジェクト]タブを指定すると、以下のダイアログボックスが開きます。



[DTU8 終了時にプロジェクトを保存]をチェックすると、状態をプロジェクトに保存して、DTU8 デバッガを終了します。また、[次回起動時に同じプロジェクトを開く]をチェックすると、終了時に保存したプロジェクトを、DTU8 デバッガの起動時に、自動的に開きます。

#### ■ プロジェクトファイルの読み込み

[ファイル]メニューの[プロジェクトを開く]で開くダイアログにプロジェクトファイルを指定すると、指定されたプロジェクトファイルがロードされ、各種設定内容が復元されます。

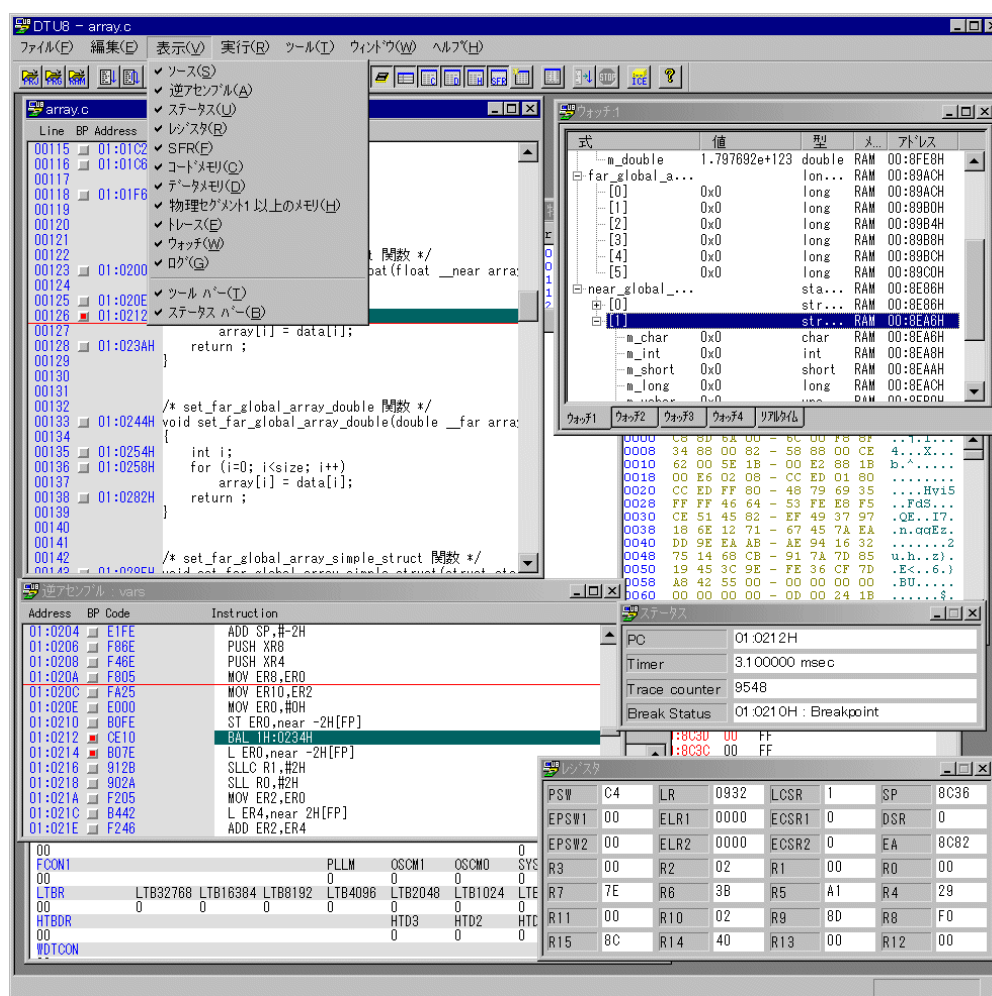
また、[ツール]メニューの[環境設定]で表示されるダイアログボックスの[ディレクトリ]タブの[プロジェクトファイル]フィールドに、あらかじめプロジェクトファイルが格納されているディレクトリを指定しておけば、プロジェクトファイルの読み込み時に、ディレクトリを選択する必要がなくなります。

## 4 章 基本操作

### 4.1 画面とメニュー

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガは、全ての機能をメニュー、ウィンドウ、ダイアログボックスにより操作可能なグラフィカルユーザインターフェース(GUI)を採用しています。<sup>7</sup>



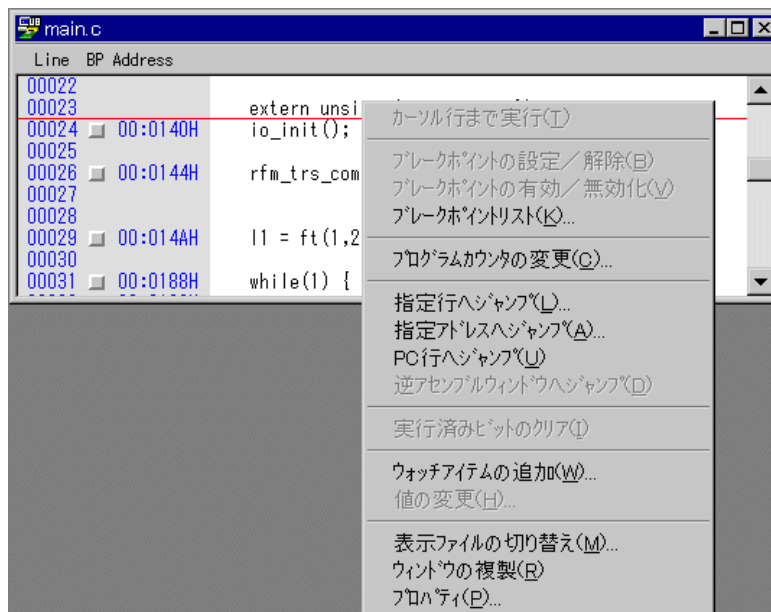
上図のように、多数の種類のウィンドウを使用して、デバッグ作業を行います。全ての機能は、メニューとウィンドウ操作で指定します。メインメニュー(メニューバー)には、頻繁に使用する機能や、どのウィンドウにも属さない機能を配置しています。

また、ウィンドウ領域を右クリックすると、ウィンドウ毎に異なるポップアップメニューが開きます。

<sup>7</sup> 基本的なデバッグ機能に関しては、スクリプトコマンドを用意しています。このスクリプトコマンドは、[ツール]メニューの[スクリプトコマンドの実行]で開くダイアログボックスで1コマンドずつ実行することができます。

## 4章 基本操作

以下は、ソースウィンドウを右クリックして表示されたポップアップメニューの例です。

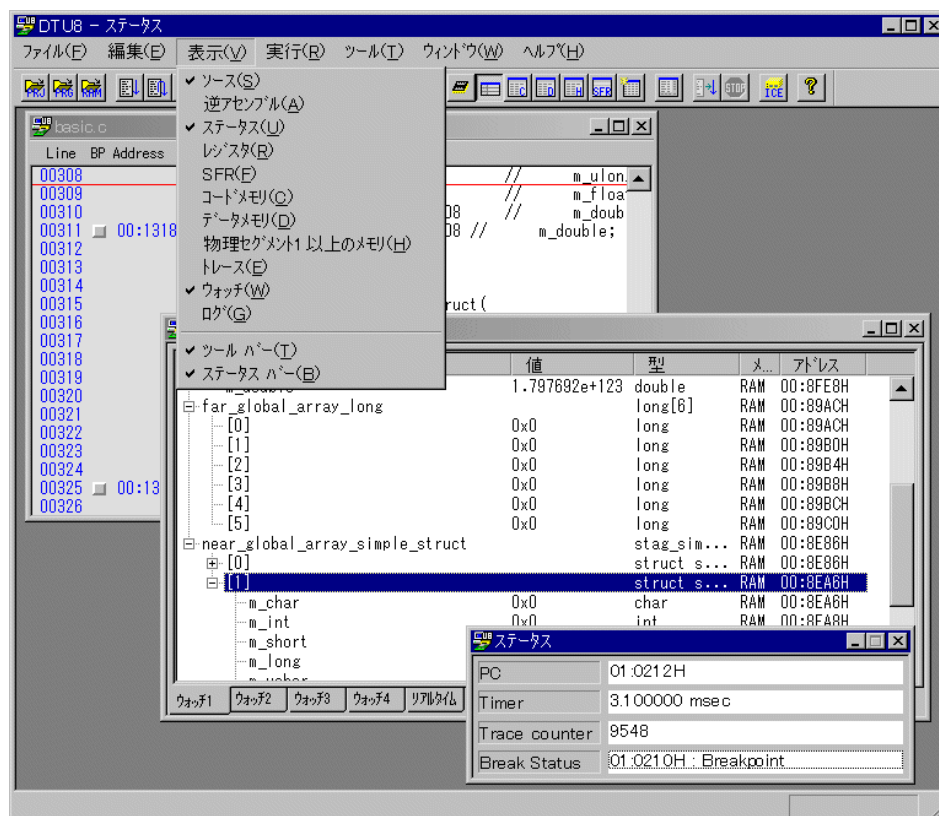


## 4.2 ウィンドウ

### 4.2.1 ウィンドウの表示

ウィンドウを開くには、[表示]メニューを指定します。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



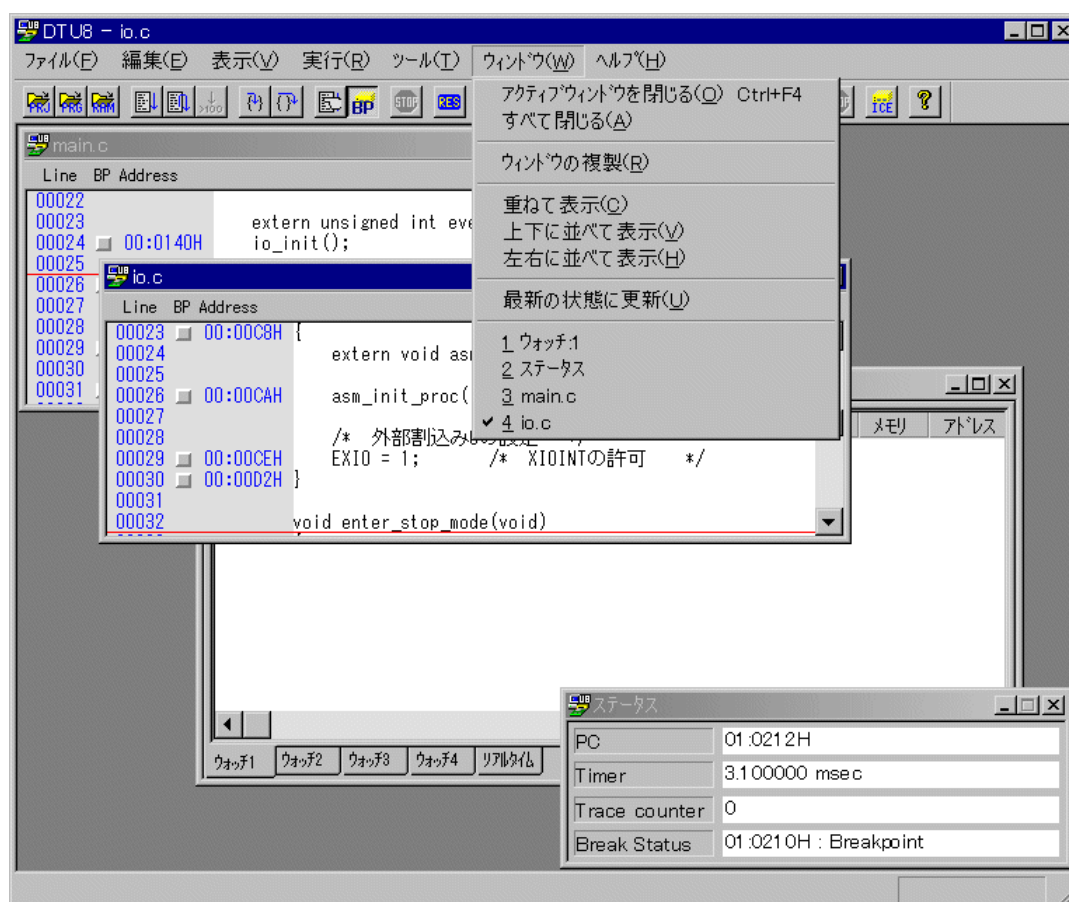
表示しているウィンドウには、チェックマーク(✓)が付いています。チェックマークが付いている[表示]メニューを選択すると、チェックマークが外れて、ウィンドウが閉じます。

また、ソース、逆アセンブル、SFR、トレース、ウォッチ、ログウィンドウは、それぞれのウィンドウ独自の形状のカーソルを持っています。メニューコマンドの中には、カーソル位置の項目に対して、操作を行うものがあります。例えば、[実行]メニューの[ブレークポイントの設定/解除]は、ソースと逆アセンブルウィンドウのカーソル位置のアドレスのブレークポイントをトグルします。

### 4.2.2 ウィンドウの複製

ウィンドウの複製をして、同じ種類のウィンドウを、同時に、参照することができます。<sup>8</sup>

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



表示中の全てのウィンドウは、上図のように、[ウィンドウ]メニューに一覧表示されます。

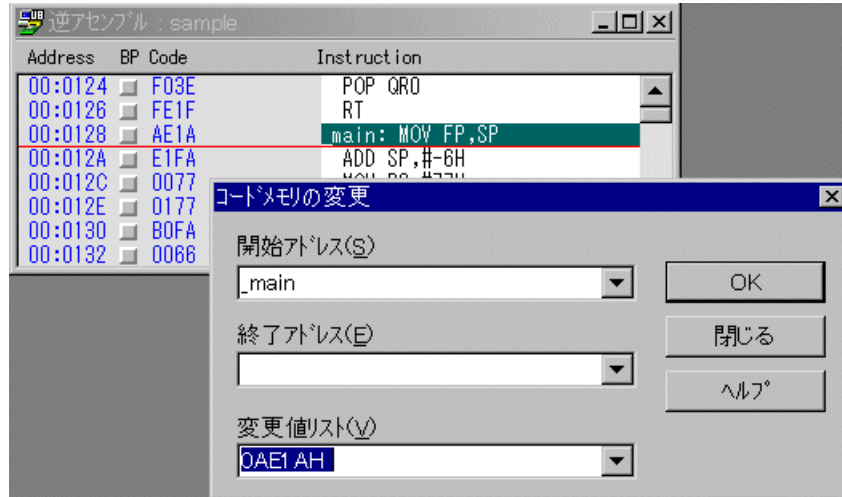
<sup>8</sup> 複製可能なウィンドウは、ソース、逆アセンブル、SFR、コードメモリ、データメモリ、物理セグメント 1 以上のメモリ、トレースです。



### 4.2.3 ウィンドウ領域のダブルクリック

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ウィンドウをダブルクリックすると、指定した対象に応じたダイアログボックスが開きます。以下は、逆アセンブルウィンドウのCodeフィールドをダブルクリックして、コードメモリの変更ダイアログを開いた例です。<sup>9</sup>



## 4.3 その他

### 4.3.1 ツールバー

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

標準的な配置では、上部のメニューバーのすぐ下に、ツールバーがあります。<sup>10</sup>



ツールバーには、頻繁に使用する重要な機能が割り当てられています。ツールバーに割り当てられている機能については、マウスカーソルを合わせると表示されるツールチップとオンラインヘルプを参照してください。

### 4.3.2 ショートカットキー

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

頻繁に使用する重要な機能には、ショートカットキーが割り当てられています。

メニューを表示したときに、ショートカットキーが割り付けられている機能には、ショートカットキーを表示しています。

DTU8 デバッガがサポートするショートカットキーの一覧については、オンラインヘルプを参照してください。

<sup>9</sup> 詳細は、オンラインヘルプのウィンドウリファレンスを参照してください。

<sup>10</sup> ツールバーを、マウスでドラッグ&ドロップすると、任意の位置に配置できます。

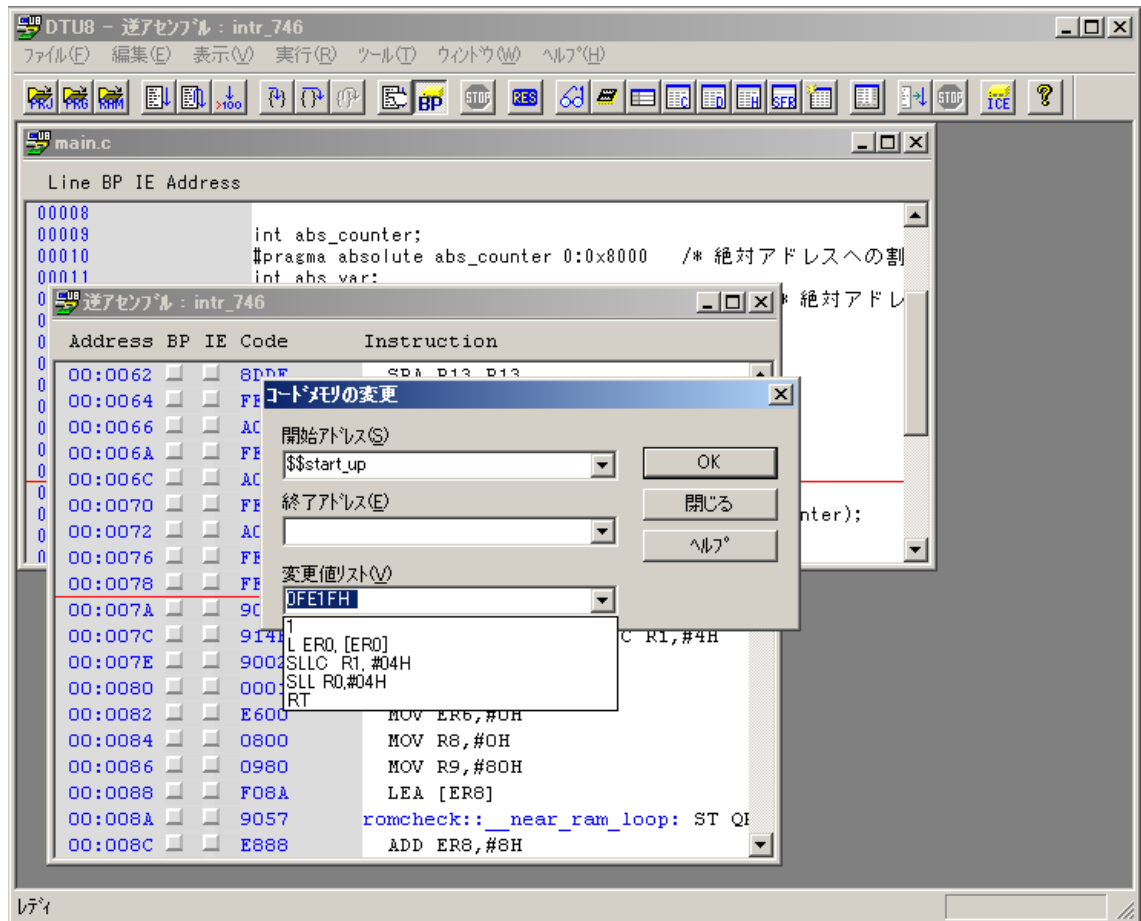


### 4.3.3 その他

## ■ 入力履歴

ダイアログボックスのほとんどの入力フィールドは、過去の入力の履歴を保持しています。<sup>11</sup>

同じ入力であれば、入力履歴のリストから選択することができます。

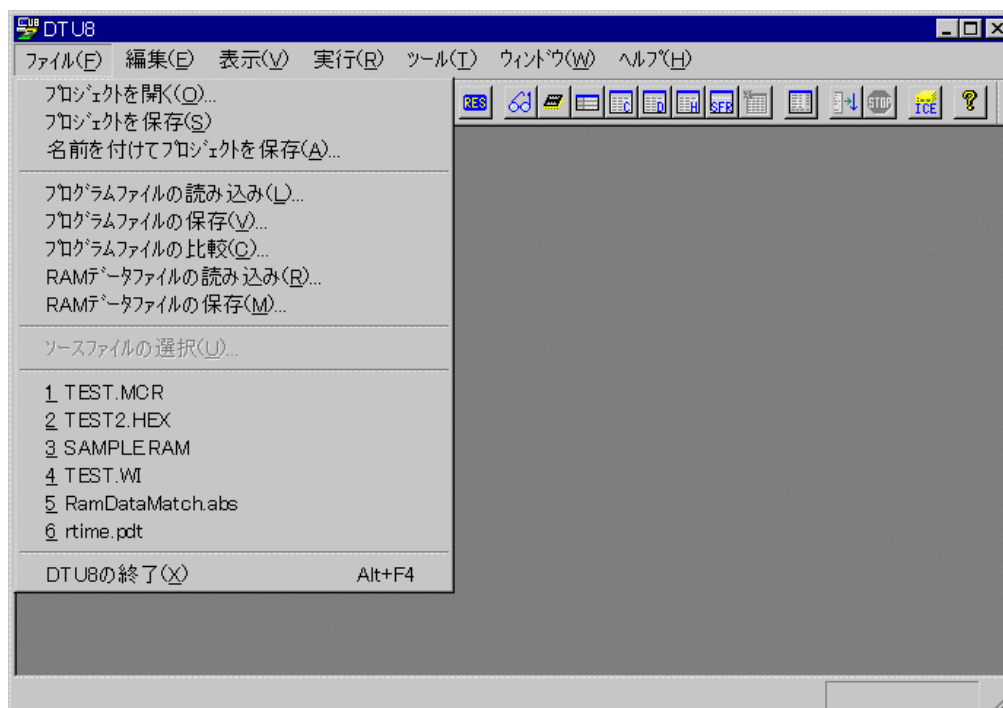


<sup>11</sup> 入力の履歴は、DTU8 デバッガを終了するとクリアされます。

### ■ 最近使ったファイルの履歴

DTU8 デバッガの機能のうち、ファイルを指定する機能を使用すると、そのファイル名が[ファイル]メニューの中に表示されます。<sup>12</sup>

この最近使ったファイルの履歴を指定すると、以前使用したファイルを、簡単に指定することができます。



最近使ったファイルの履歴は、[ツール]メニューの[環境設定]で開くダイアログボックスの[その他]タブにある[ファイルメニューに編集履歴を表示する]チェックボックスで履歴の表示を制御および履歴の個数の設定が可能です。

<sup>12</sup> ロードしたオブジェクトファイルをカレントディレクトリとして、カレントディレクトリのファイルは、最近使ったファイルの名前にディレクトリパスが付きません。

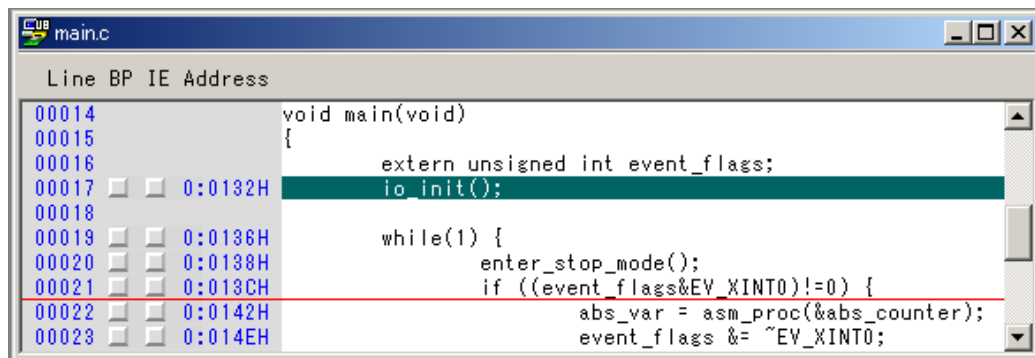
## 5 章 ソースレベルとシンボリックデバッグ

### 5.1 ソースウィンドウ

#### 5.1.1 表示

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガを使用することによって、作成したソースプログラムをソースウィンドウに表示し、基本的なデバッグ作業が可能となります。ソースウィンドウを表示するには、[表示]メニューの[ソース]を選択します。



[Line]	ソースファイルの行番号
[BP]	ブレークポイント設定／解除チェックボックス
[IE]	実行済みビット(シミュレーションモードのみ)
[Address]	プログラムコードアドレス

ソースウィンドウ上で緑色の反転表示された行は、現在のPCを表しています。また、赤色のアンダーラインは、カーソル行を表しています。<sup>13</sup>

ソースウィンドウ上に表示されているテキストを選択して、クリップボードへコピーすることができます。クリップボードにコピーした内容は、いろいろなダイアログの入力フィールドで使うことができます。

#### 5.1.2 ソースウィンドウ上での変数値表示

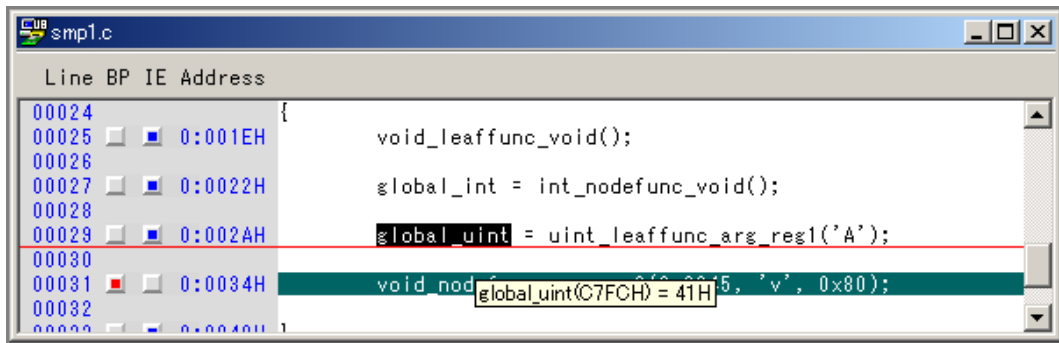
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ソースウィンドウ上で選択した変数のアドレスと値を表示する機能です。

ソースウィンドウ上で、変数名を選択状態にし、マウスカーソルをその変数名上へ移動すると、変数のアドレスと値をバルーン表示します。

以下の例は、変数”global\_uint”のアドレス(0C7FCH)と値(41H)をバルーン表示しています。

<sup>13</sup> ソースウィンドウは、水平スクロールバーを持っていません。ただし、[←][→][Home][End]キーで、水平スクロールが可能です。



選択した変数が、配列や構造体および共用体の名前であった場合は、

「変数名(アドレス) = {...}」

のように表示します。

本機能では、以下の変数および式をバルーン表示します。

- ・ 符号付きおよび符号なしの char、int、long、enum 型などの変数単体
- ・ 任意の添え字付きの配列(array[10]など)
- ・ メンバ演算子を使用した構造体、共用体のメンバ(struct.member など)
- ・ 特定のアドレスが指し示されるポインタ式(struct->member、\*pointer、\*(val+2)など)
- ・ SFR<sup>14</sup>

ローカル変数や static 宣言されたグローバル変数を表示する場合、現在の PC が選択された変数が定義されているスコープ内でない場合はバルーン表示されません。

### 5.1.3 ソースファイルの検索順序

ソースウィンドウにソースファイルを表示するためには、ソースレベルデバッグ情報を含むアブソリュートオブジェクト形式のオブジェクトプログラムを読み込む必要があります。<sup>15</sup>

CCU8 コンパイラまたはRASU8 アセンブラが、オブジェクトファイルを生成したときに、ソースファイル名をデバッグ情報の一部として出力しています。

DTU8 デバッガは、読み込んだアブソリュートオブジェクトプログラムのデバッグ情報を参照して、ソースファイルを開きます。このとき、ソースファイルを、以下の順序で探します。

#### ■ 絶対パス表現のソースファイル名の場合

- その絶対パスで指定されたファイルを開くことを試みます。
- 絶対パスで指定されたパスにファイルが見つからない場合、[ツール]メニューの[環境設定]の[ディレクトリ]タブで指定する[ソースファイル]に設定されたディレクトリを探します。

#### ■ 相対パス表現のソースファイル名の場合

- 読み込んだアブソリュートオブジェクトファイル(.ABS)が存在するディレクトリをカレントとして、そこからの相対パスでファイルを開くことを試みます。
- 読み込んだアブソリュートオブジェクトファイル(.ABS)が存在するディレクトリをカレントして、そ

<sup>14</sup> 参照することにより、状態が変化する SFR が存在する場合があります。事前に対象機種のハードウェアマニュアルをご確認の上、バルーン表示機能をご利用ください。

<sup>15</sup> アブソリュートオブジェクトファイルについては、「2 章 起動の前に」を参照してください。

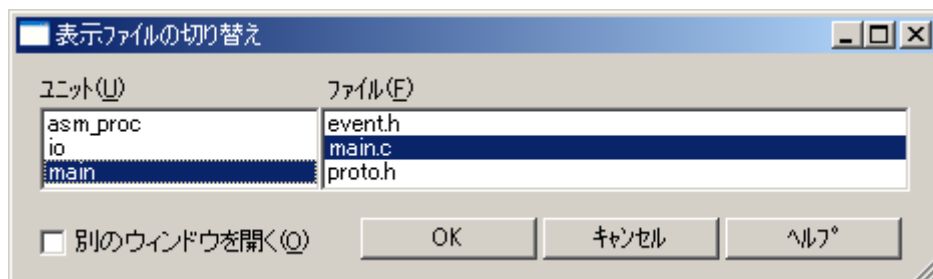
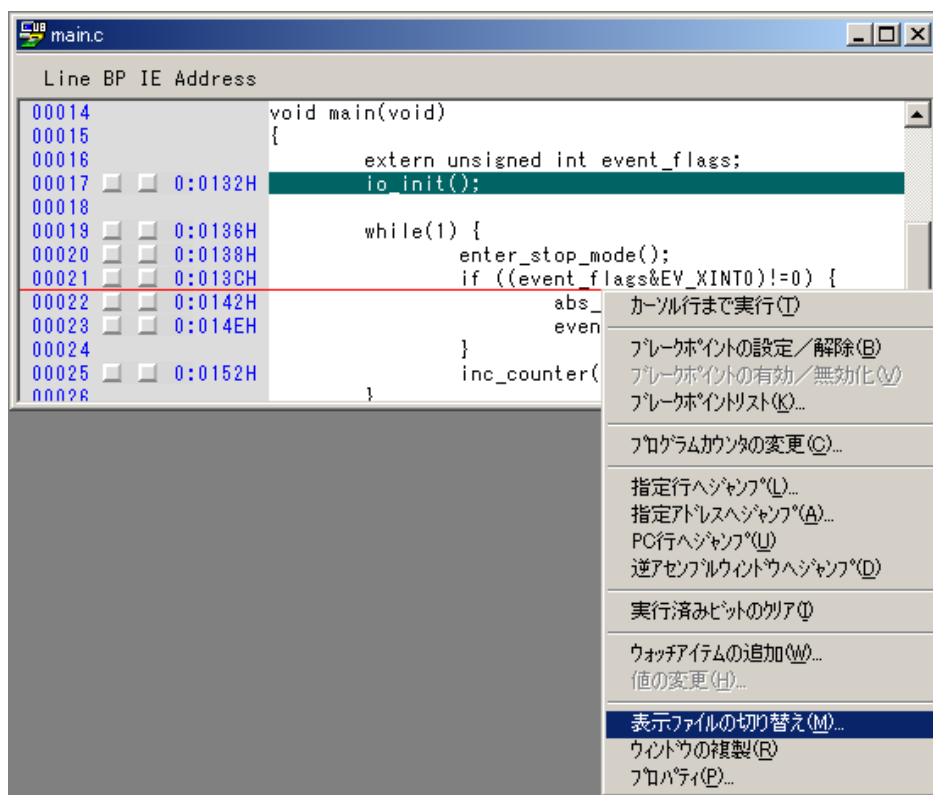
ここからの相対パスで見つからない場合には、[ツール]メニューの[環境設定]の[ディレクトリ]タブで指定する[ソースファイル]に設定されたディレクトリをカレントとして、そこからの相対パスでファイルを開くことを試みます。

- III. 相対パスでファイルが見つからない場合には、[ツール]メニューの[環境設定]の[ディレクトリ]タブで指定する[ソースファイル]に設定されたディレクトリを探します。

#### 5.1.4 表示ファイルの切り替え

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

アブソリュートオブジェクトファイルを読み込んでから初めてソースウィンドウを開く場合、main()関数を含むソースファイルか、現在の PC を含むソースファイルを開きます。プログラムソースが複数で構成されている場合に、他のソースファイルを開きたいときは、ソースウィンドウを右クリックして表示されるポップアップメニューの[表示ファイルの切り替え]を選択します。



[表示ファイルの切り替え]ダイアログボックスでは、左側には、読み込んだプログラムを構成するユニット(コンパイル、アセンブルの単位)がリストで表示されています。ユニットを選択して、右側にリ

ストされるユニットを構成するファイル名の中から、表示したいファイルを選択し[OK]ボタンをクリックすることにより、表示ファイルを切り替えることができます。<sup>16</sup>

[表示ファイルの切り替え]ダイアログボックスで、[別のウィンドウ開く]をチェックすると、新しいウィンドウに、選択したファイルを表示します。

なお、この[表示ファイルの切り替え]ダイアログボックスは、[ファイル]メニューの[ソースファイルの選択]を指定して表示することもできます。この場合には、ソースウィンドウを表示していない状態でも、表示するファイルを指定してソースウィンドウを開くことができます。

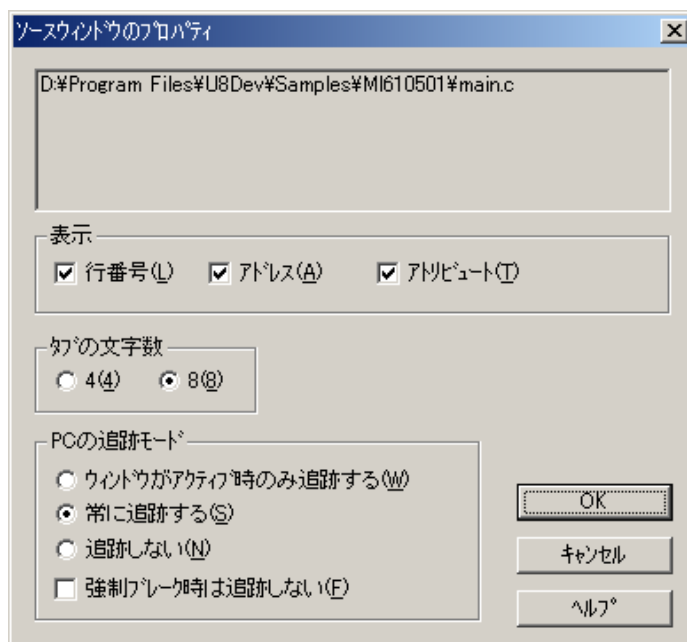
また、ポップアップメニューの [指定アドレスへジャンプ]で開くダイアログボックスを使うと、指定したアドレスを含むソースファイルに切り替わります。

[表示ファイルの切り替え]ダイアログボックスを閉じる場合は、[キャンセル]ボタン、またはタイトルバーに置かれた[閉じる]ボタンをクリックしてください。

### 5.1.5 ソースウィンドウのプロパティ

ソースウィンドウを右クリックして開くポップアップメニューの[プロパティ]を開くと、以下のダイアログボックスが開きます。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



このプロパティダイアログで、表示されているファイルのパスの確認や、ソースウィンドウの表示形式を変更できます。

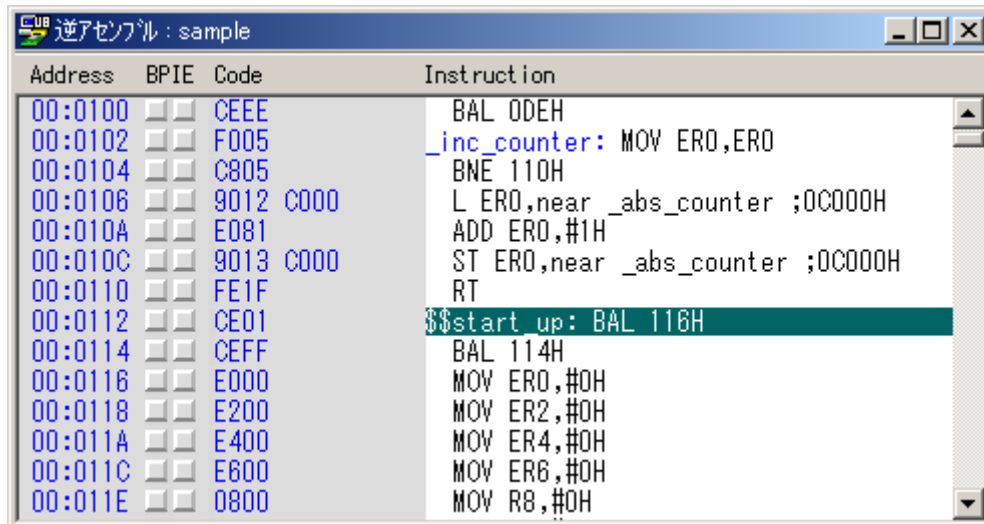
## 5.2 逆アセンブルウィンドウ

### 5.2.1 表示

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガは、エミュレータのメモリ(Code 属性を持つ領域)の内容を逆アセンブルして表示する逆アセンブルウィンドウを持っています。

逆アセンブルウィンドウを使用すると、命令コードレベルのデバッグが可能になります。



[Address]	コードメモリのアドレス
[BP]	ブレークポイント設定／解除チェックボックス
[IE]	実行済みビット(シミュレーションモードのみ)
[Code]	コード(ワード値のリスト)

逆アセンブルウィンドウ上で緑色の反転表示された行は、現在のPCを表しています。また赤色のアンダーラインは、カーソル行を表しています。<sup>17</sup>

逆アセンブルウィンドウ上に表示されているテキストを選択して、クリップボードへコピーすることができます。クリップボードにコピーした内容は、いろいろなダイアログの入力フィールドで使うことができます。

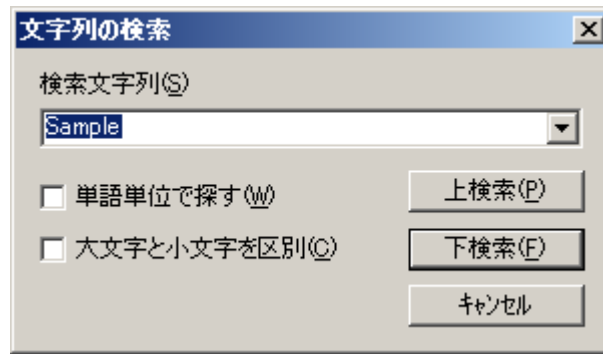
### 5.2.2 逆アセンブルウィンドウの文字列検索

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

逆アセンブルウィンドウ上の”Instruction”フィールドに表示された文字列を検索することができます。

逆アセンブルウィンドウがアクティブな状態で、[編集]メニューの[検索]を選択すると、次のようなダイアログボックスを表示し、逆アセンブルウィンドウ上の文字列を検索できます。検索対象は、逆アセンブルウィンドウ上の”Instruction”フィールドに表示された文字列のみであり、”Address”、”BP”、”IE”および”Code”の各フィールドは検索しません。

<sup>17</sup> 逆アセンブルウィンドウは、水平スクロールバーを持っていません。ただし、[←][→][Home][End]キーで、水平スクロールが可能です。



■[検索文字列]入力フィールド

検索したい文字列を入力します。

■[単語単位で探す]チェックボックス

このチェックボックスをチェックすると、検索文字列を単語毎に検索します。例えば、検索文字列に”rom”を指定した場合、”romwindow”や”from”などの単語は検索しません。

■[大文字と小文字を区別する]チェックボックス

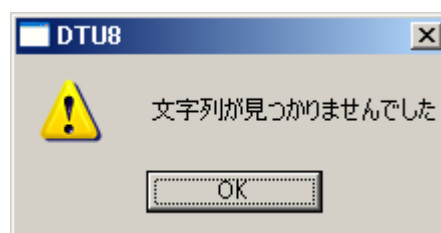
このチェックボックスをチェックすると、検索文字列の大文字と小文字まで比較して検索します。例えば、検索文字列に”abc”を指定した場合、”ABC”や”Abc”などは検索しません。

■[上検索]ボタン、[下検索]ボタン

このボタンをクリックすると、検索を開始します。

[下検索]ボタンをクリックすると、現在のカーソル位置から逆アセンブルウィンドウの末尾に向かって文字列を検索します。末尾まで文字列が見つからない場合、ウィンドウの先頭から検索を継続し、ウィンドウ全体を検索します。文字列が見つからない場合、「文字列が見つかりませんでした」というメッセージを表示します。

[上検索]ボタンをクリックすると、逆アセンブルウィンドウの先頭に向かって文字列を検索します。



[上検索]ボタンまたは[下検索]ボタンをクリックして、文字列の検索を実行すると、文字列の検索ダイアログは閉じます。続けて検索する場合は、F3キー(下検索)またはShift+F3キー(上検索)を押してください。



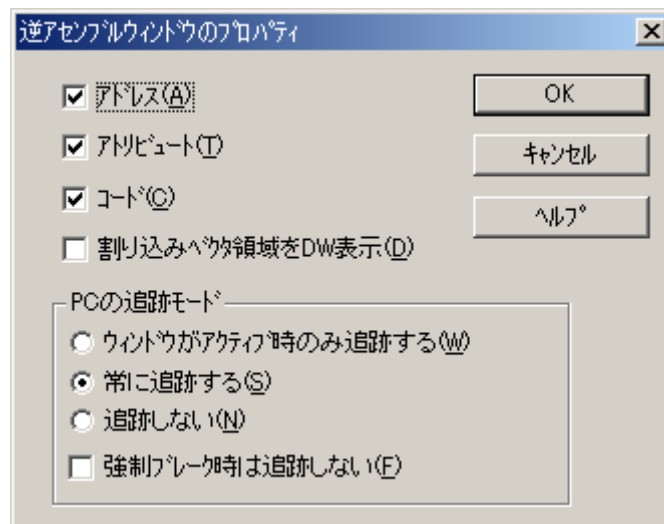
検索中には、次のようなダイアログを表示します。



[中断]ボタンをクリックすると、検索を中断することができます。検索を中断した場合、カーソルは検索前の位置から変化しません。

### 5.2.3 逆アセンブルウィンドウのプロパティ

逆アセンブルウィンドウを右クリックして開くポップアップメニューの[プロパティ]を開くと、以下のダイアログボックスが開きます。



プロパティで、逆アセンブルウィンドウの表示形式を変更できます。

[割り込みベクタ領域を DW 表示]をチェックすると、0:0000h から 0:00FEh 番地を「DW XXXXH」の形式で表示して、割り込みベクタのアドレスを参照することができます。

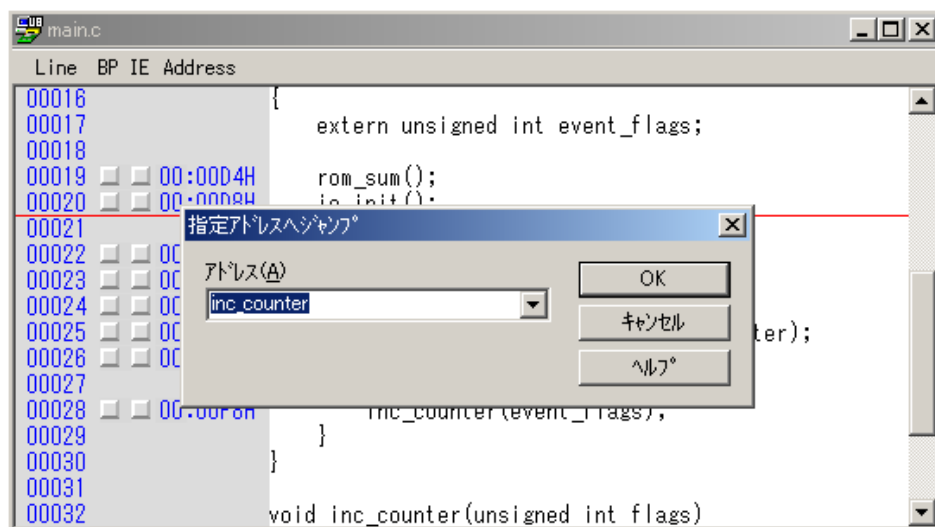
## 5.3 シンボリックデバッグ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

## ■ 数値指定入力時のシンボル指定

数値を指定する入力フィールドには、「付録 13.1 シンボル」で説明するシンボルを指定できます。

例えば、ソースウィンドウのポップアップメニューの[指定アドレスへジャンプ]を指定すると、カーソルの移動先のアドレス値を指定するためのダイアログボックスが開きます。



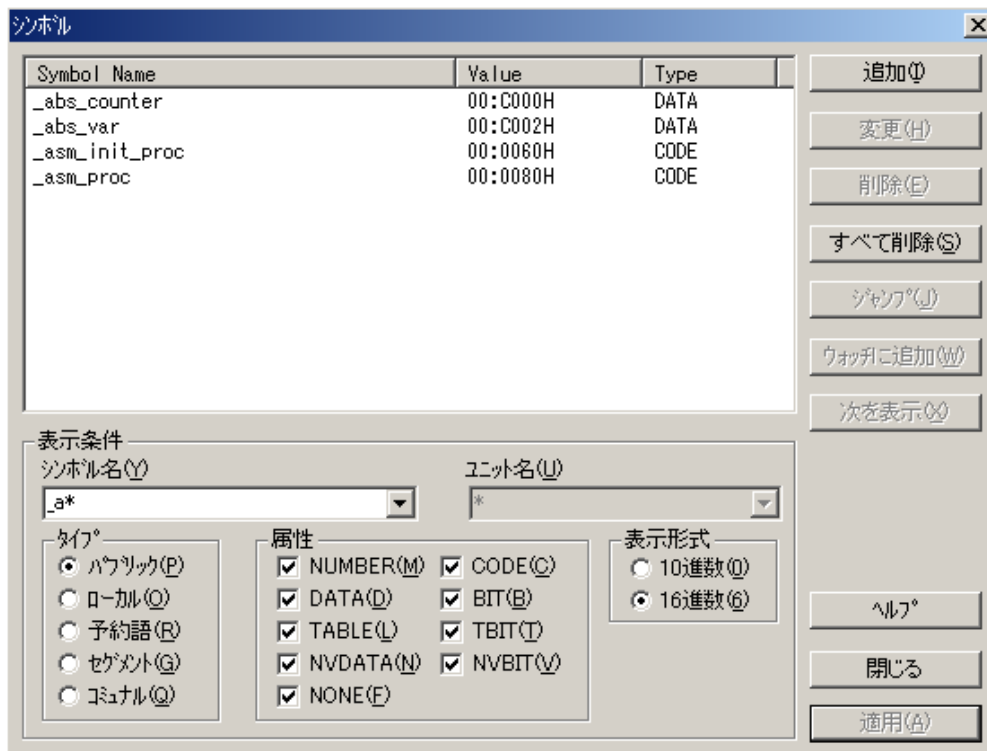
ジャンプ先にC関数名を指定すると、そのC関数を含むソースファイルに切り替わり、指定されたC関数の先頭にカーソルが移動します。<sup>18</sup>

逆アセンブルウィンドウ、コード、データ、物理セグメント 1 以上のメモリでも、同様な方法で、任意のアドレスに、カーソルをジャンプさせることができます。

## ■ 登録されたシンボルの確認

アセンブリ言語レベルのシンボル情報は、[ツール]メニューの[シンボルリスト]を指定して開くダイアログボックスで、登録の確認、定義、削除、追加などのメンテナンスが可能です。

<sup>18</sup> ソースウィンドウのソースをダブルクリックすると単語を選択できます。その状態で、[CTRL]+[C]でクリップボードにコピーし、ダイアログボックスの入力ボックスで、[CTRL]+[V]でコピーしたテキストを貼り付けることができます。



表示条件を設定して、[適用]ボタンを押すと、条件にマッチするシンボルがリストされます。リストのタイトル(Symbol Name, Value, Type)をクリックすると、指定したタイトルの内容で並び替えて表示します。

## 5.4 ウィンドウ間のリンク

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガのウィンドウには、多くの種類がありますが、それを相互に関連付けて、デバッグ作業を支援します。

### ■ ソースウィンドウから逆アセンブルウィンドウへの移動

ソースウィンドウのポップアップメニューから[逆アセンブルウィンドウへジャンプ]を選択すると、ソースウィンドウ上のカーソル位置のアドレスを、逆アセンブルウィンドウで表示します。これは、ソースウィンドウ上で表示しているアドレスをマウスでダブルクリックしても同様に機能します。

実行コードを持たないソース行からは、逆アセンブルウィンドウにジャンプできません。

### ■ 逆アセンブルウィンドウからソースウィンドウへの移動

逆アセンブルウィンドウのポップアップメニューから[ソースウィンドウへジャンプ]を選択すると、逆アセンブルウィンドウのカーソル位置のアドレスを含むソース行を、ソースウィンドウに表示します。

逆アセンブルウィンドウのカーソル位置のアドレスを含むソース行がない場合には、ソースウィンドウがアクティブ(最前面)になるだけです。

### ■ ウォッチウィンドウへの項目の追加

ソースウィンドウや逆アセンブルウィンドウで表示しているテキストを選択し、ポップアップメニューの[ウォッチアイテムの追加]を選択すると、選択した文字列をウォッチウィンドウへ登録することができます。詳しくは、「10 章 ウォッチ機能」を参照してください。

■ **トレースウィンドウからの移動**  
(トレース機能を持つエミュレーションシステムおよびシミュレーションモードの場合)

トレースウィンドウのポップアップメニューの[ソースウィンドウへジャンプ]や[逆アセンブルウィンドウへジャンプ]で、対応するトレース PC のソースウィンドウへ移動できます。

## 5.5 PC の追跡モード

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ソースウィンドウと逆アセンブルウィンドウは、PC が変化するイベントで、自動的に PC を追跡するようにウィンドウを切り替える設定にできます。この PC の追跡モードは、それぞれのウィンドウのプロパティで設定します。

■ **[ウィンドウがアクティブ時のみ追跡する]モード**

これは、ウィンドウがアクティブ(最前面)のときに、PC を追跡するモードです。

例えば、複数のソースウィンドウを表示しているとき、PC を追跡させるウィンドウを、アクティブに(選択)するだけで、PC を追跡させることができます。

■ **[常に追跡する]モード**

これは、ウィンドウのアクティブ状態に関わらず、常に、PC を追跡するモードです。

最初に開いたソースウィンドウと逆アセンブルウィンドウは、このモードに設定されています。

このモードを利用すると、ソースウィンドウと逆アセンブルウィンドウで、同時に、PC を追跡させることができます。

■ **[追跡しない]モード**

これは、PC が変化するイベントで、ウィンドウの表示が変化しないモードです。

複製されたソースウィンドウと逆アセンブルウィンドウは、このモードに設定されています。

このモードは、複数のソースウィンドウを開いているときに、ひとつだけ[常に追跡する]モードで追跡させて、他のウィンドウを[追跡しない]モードに設定することにより、表示が変化しないウィンドウを参照と設定に利用するような使い方ができます。

■ **[強制ブレーク時は追跡しない]モード**

これは、[PC の追跡モード]の設定に関わらず、強制ブレーク時には PC を追跡しないようにするモードです。

[強制ブレーク時は追跡しない]チェックボックスをチェックすると、強制ブレークした場合に限り、各ウィンドウの表示において PC を追跡しないようになります。

## 6 章 プログラム実行機能

### 6.1 実行モード

#### 6.1.1 連続実行(エミュレーション実行)

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE

DTU8 デバッガを Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モード、uEASE モードまたは nanoEASE モードで起動した状態で、エミュレータを連続実行させることをエミュレーション実行と呼びます。

この実行モードは、エミュレーションシステムの最も基本となるもので、ターゲットマイコンと同等な動作を行う状態です。

プログラムを連続実行するには、[実行]メニューから[プログラム実行]を選択します。

この実行モードは、ブレイク条件が成立するか、強制的にブレイクされるまで継続します。

プログラム実行中は、ステータスバーに「エミュレーション中」と表示されています。[ツール]メニューの[環境設定]で表示される環境設定ダイアログのその他タブで、[プログラム実行中にエミュレーション中ダイアログを表示する]をチェックしておけば、エミュレーション実行中を表すダイアログボックスが表示されるようになります。

#### 6.1.2 連続実行(シミュレーション実行)

Simulate

DTU8 デバッガをシミュレーションモードで起動した状態で、ロードしたプログラムを 1 サイクルずつシミュレーションしながら実行することをシミュレーション実行と呼びます。

この実行モードでは、DTU8 デバッガが PC のアドレスの命令コードを解析し、命令に応じた処理を行います。

プログラムを連続実行するには、[実行]メニューから[プログラム実行]を選択します。

この実行モードは、ブレイク条件が成立するか、強制的にブレイクされるまで継続します。

プログラム実行中は、ステータスバーに「エミュレーション中」と表示されています。[ツール]メニューの[環境設定]で表示される環境設定ダイアログのその他タブで、[プログラム実行中にエミュレーション中ダイアログを表示する]をチェックしておけば、プログラム実行中を表すダイアログボックスが表示されるようになります。

#### 6.1.3 カーソルまで実行

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

連続実行と同等な実行モードですが、アクティブな状態のソースウィンドウまたは逆アセンブルウィンドウのカーソルのある位置まで実行するためのモードです。

ソースウィンドウまたは逆アセンブルウィンドウ上で、カーソルを目的の行へ移動して、ポップアップメニューの[カーソル位置まで実行]を選択します。<sup>19</sup>

この実行モードでは、カーソル位置に PC が到達する以前に有効なブレイクポイントが設定され

<sup>19</sup> ソースウィンドウの場合には、実行コードがある行(ブレイクポイントが設定可能な行)でのみ有効になるメニューです。

ていれば、そのブレークポイントでブレークします。

#### 6.1.4 リセット後に実行

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

[実行]メニューから[リセット後に実行]を選択すると、まずリセットしてからプログラム実行を開始します。プログラムの先頭から再実行したいときは、この機能が便利です。エミュレーションモードの場合は、エミュレータをリセットした後プログラム実行が開始されます。

#### 6.1.5 ステップイン実行

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ステップイン実行を行うためには、[実行]メニューの[ステップイン]を選択します。

ステップイン実行は、Cソースを表示しているソースウィンドウがアクティブであれば、1ソース行単位、それ以外であれば1命令単位で実行します。実行したソース行がCの関数呼び出しの場合や、実行した命令がサブルーチンコール命令(BL命令)の場合、C関数やサブルーチンの中(サブルーチンの先頭アドレス)にPCが移行します。

ステップイン実行の直後に割り込みが発生した場合には、受け付けた割り込みルーチンの先頭でステップイン実行処理が終了します。

このため、ステップイン実行を使用すると、プログラムの流れを詳細に追うことができます。

ステップイン実行中は、一時的に、ブレークポイントが無効化され、ブレークポイントが設定されているアドレスを通過しても、引き続きステップイン実行処理が行われます。

ソースウィンドウ上のステップイン実行で、(関数コール、サブルーチンコール、割り込みによって)分岐が発生したときに、PCがデバッグ情報のないアドレスへ移行した場合には、逆アセンブルウィンドウがアクティブになります。デバッグ情報がない領域にリターンした場合も同様に、逆アセンブルウィンドウがアクティブになります。

#### 6.1.6 ステップオーバー実行

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ステップオーバー実行を行うためには、[実行]メニューの[ステップオーバー]を選択します。

ステップオーバー実行は、Cソースを表示しているソースウィンドウがアクティブであれば、1ソース行単位、それ以外であれば1命令単位で実行します。実行した命令がCの関数呼び出しの場合や、実行した命令がサブルーチンコール命令(BL命令)の場合、C関数およびサブルーチンから復帰するまで実行します。<sup>20</sup>

ステップオーバー実行中は、一時的に、ブレークポイントが無効化され、ブレークポイントが設定されているアドレスを通過しても、引き続きステップオーバー実行処理が行われます。

また、ステップオーバー実行中に割り込みが発生した場合には、割り込みルーチンから復帰するまで実行します。このため、ステップオーバー実行を使用すると、サブルーチンや割り込みルーチンがスキップされるため、メインの処理を追う場合に有効です。

ソースウィンドウ上のステップオーバー実行で、デバッグ情報がない領域にリターンした場合、ステップイン実行と同様に、ソースウィンドウ上でPCが追跡できなくなります。

#### 6.1.7 ステップアウト実行

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ステップアウト実行を行うためには、[実行]メニューの[ステップアウト]を選択します。

<sup>20</sup> アセンブリ言語レベルのステップオーバー実行で、関数、サブルーチン、及び、割り込みルーチンをスキップする場合には、復帰アドレスまで連続実行処理になります。

ステップアウトを実行すると、現在のプログラムカウンタの位置にある関数の最後まで命令を実行し、関数の呼び出し元に戻ります。<sup>21</sup> ただし、[実行]メニューの[ブレーク条件の設定]で開くダイアログで指定する[アドレスブレーク]の条件が、一時的に無効化されます。<sup>22</sup>

ステップアウト機能は、Cデバッグ情報を元に実現しているため、ステップアウトに関するCデバッグ情報を持たないアセンブリソースレベルでは、ステップアウト機能は無効となります。

したがって、強制ブレークをした場合や逆アセンブリウィンドウ上にてステップ実行を行った場合は、ステップアウト機能が無効となっている場合があります。この場合には、C ソースウィンドウをアクティブにした上で、現在表示中の PC 行から次の行までステップ実行を行うことにより、ステップアウト機能が有効となります。

ステップアウト機能は、ステップアウトの実行が可能な場合にのみ選択可能となります。それ以外の場合には、[実行]メニューの[ステップアウト]およびツールバーの[ステップアウト]ボタンはグレイアウト表示となり、ステップアウト機能は選択不可となります。

### 6.1.8 C ソースレベルのステップ実行

#### ■ エミュレーションライブラリおよび C 標準ライブラリの呼び出し

CCU8 コンパイラは、nX-U8 コア、nX-U16 コアがサポートしていないデータ型の算術演算を含む記述があるとき、エミュレーションライブラリを使用したコードを生成します。<sup>23</sup>

このため、PC を追跡するモードになっているソースウィンドウ上でCソースを表示しているときでも、エミュレーションライブラリを含む C コードをステップイン実行したときに、予期せず逆アセンブルウィンドウがアクティブになります。これは、PC がソースデバッグ情報を含む領域から外れてしまったことが原因です。同様に、C 標準ライブラリを含む C コードをステップイン実行したときにも逆アセンブルウィンドウがアクティブになります。

CCU8 コンパイラが生成するコードに対して、エミュレーションライブラリおよび C 標準ライブラリを使用しているかどうかを意識することは大変な負担になります。

そこで、[ツール]メニューの[環境設定]で開くダイアログボックスの[その他]タブに、[C ソースレベルでは常に C ライブラリをステップオーバー実行する]チェックボックスが用意されています。このチェックボックスをチェックしておくと、DTU8 デバッガは、ソースレベルのステップイン実行で、CCU8 コンパイラのエミュレーションライブラリおよび C 標準ライブラリが実行されるとき、自動的にステップオーバーに切り替えます。

ただし、エミュレーションライブラリ実行中または C 標準ライブラリ実行中に割り込みが発生する場合には、次のことに注意する必要があります。通常のステップイン実行は、割り込みルーチンの先頭でブレークするようにステップ実行を行います。[C ソースレベルでは常に C ライブラリをステップオーバー実行する]をチェックすると、エミュレーションライブラリ実行中または C 標準ライブラリ実行中に割り込みが発生しても、割り込みルーチンの先頭で止まらずに、エミュレーションライブラリまたは C 標準ライブラリを呼び出した行の次まで実行します。

<sup>21</sup> コンパイラの最適化によっては、ステップアウトが予期しない動作になることがあります。例えば、f1()の末尾で f2()を呼び出し、f2()の末尾でf3()を呼び出すというような記述をした場合、末尾呼び出しの最適化(BL 命令の代わりに B 命令を使う最適化)が行われます。このような場合、最下層の関数である f3()からステップアウトを行うと、f2()に戻るのではなく、f1()の呼び出し元にまで戻ることになります。

<sup>22</sup> [ステップアウト実行]の実行モードは、[ブレーク条件の設定]ダイアログボックスの[アドレスブレーク]機能を利用して、実現しているためです。

<sup>23</sup> エミュレーションライブラリの種類については、CCU8 ユーザーズマニュアルを参照してください。





### ■ Cコード中のアセンブリ命令の記述

CCU8 コンパイラは、`#asm`、`#endasm` 前処理指令、`#pragma asm`、`#pragma endasm` プラグマ、`_asm`、`_endasm` キーワードを使用して、C ソースコード中に、アセンブリ命令を記述できます。  
これらのアセンブリ命令の記述は、そのブロックが C コードの 1 行とみなされます。

## 6.2 リセット

リセットを行う場合には、[実行]メニューの[リセット]を選択します。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

### ■ Dr. U8 ICE, Dr. U16 ICE, Dr. ICE モード時のリセット

Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モード時は、エミュレータに搭載されているターゲットマイコンをリセットします。

uEASE  
nanoEASE

### ■ uEASE モード, nanoEASE モード時のリセット

uEASE モード、nanoEASE モード時は、uEASE または nanoEASE と接続しているターゲットマイコンをリセットします。リセットが成功すると、「ターゲットをリセットしました。」という内容のメッセージダイアログを表示します。

Simulate

### ■ シミュレーションモード時のリセット

シミュレーションモード時は、DTU8 デバッガ内部のシミュレーションエンジンをリセットします。

uEASE  
nanoEASE

### ■ RESET\_N 端子によるリセット

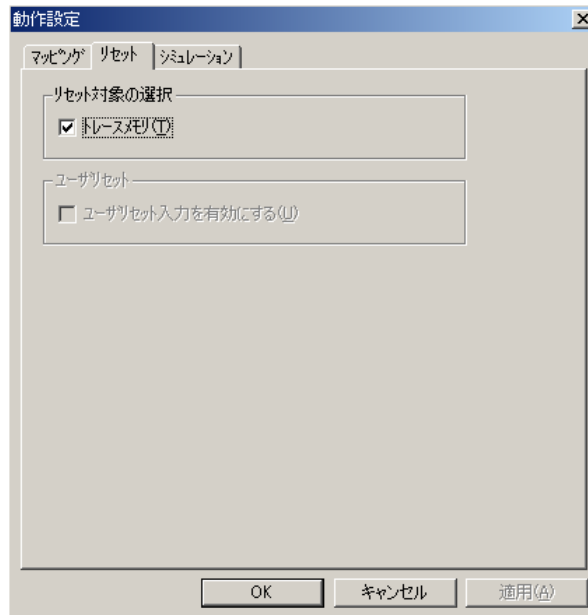
uEASE モード、nanoeEASE モード時は、[実行]メニューの[デバイスリセット]を選択することにより uEASE または nanoEASE からターゲットマイコンに対し RESET\_N 端子によるリセットを行うことができます。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

### ■ リセット時の設定

接続するエミュレータの種類によっては、[ツール]メニューの[動作設定]を指定して開くダイアログボックスの[リセット]タブがあることがあります。





[動作設定]ダイアログボックスの[リセット]タブで、リセットを実行したときの初期化対象を設定できます。

## 6.3 プログラム実行中に使用可能な機能について

### 6.3.1 ウィンドウ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

プログラム実行中は、以下のウィンドウが使用可能です。その他のウィンドウは、アクティブにすることはできませんが、スクロールやサイズの変更はできません。閉じることはできますが、新たに開くことはできません。

#### ■ ソースウィンドウ

ソースファイルを表示します。表示するファイルを切り替えることができます。

#### ■ ステータスウィンドウ

プログラム実行中にステータスウィンドウを表示すると、[Timer]フィールドが更新され続け、プログラム実行の経過時間を確認できます。

#### ■ ログウィンドウ

エミュレーション中にログを出力する機能はありません。編集、保存、クリア等のログウィンドウを操作する上での制限はありません。

#### ■ ウォッチウィンドウ（リアルタイムウォッチ機能に対応したエミュレータの場合）

プログラム実行中は、[リアルタイム]タブに登録された項目が更新され続けます。プログラム実行開始前に、[リアルタイム]タブにウォッチ項目を追加しておきます。<sup>24</sup>

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

<sup>24</sup> リアルタイムウォッチに登録可能なアイテムの個数は、エミュレーションモードの場合はバイト指定で最大 8 個、シミュレーションモードの場合は、バイト指定で最大 16 個です。

### ■ リアルタイム RAM モニタウィンドウ

プログラム実行により変更された RAM の内容をリアルタイムに確認できます。

Dr.U8 ICE  
Dr.U16 ICE

## 6.3.2 メニュー

プログラム実行中にも使用可能なメニューは、以下のものになります。

### ■ [ファイル]メニューの[ソースファイルの選択]

ソースウィンドウの表示ファイルを切り替えられます。

### ■ [ファイル]メニューの[終了]

「6.5 エミュレーション実行中の終了と再接続」を参照してください。

### ■ [編集]メニュー

アクティブなウィンドウで意味のある項目だけ、有効になります。

### ■ [表示]メニュー

使用可能なウィンドウの項目だけ、有効になります。

### ■ [実行]メニューの[強制ブレーク]

エミュレーション実行を強制的にブレークする項目です。

### ■ [実行]メニューの[ブレークポイントリスト]

設定されているブレークポイントのリストを参照できます。変更はできません。

### ■ [実行]メニューの[ブレーク条件の設定]

設定されているブレーク条件の設定を参照できます。変更はできません。

### ■ [ツール]メニューの[シンボルリスト]

登録されているシンボルの参照ができます。変更はできません。

### ■ [ツール]メニューの[動作設定]

設定状態の確認だけです。変更はできません。

### ■ [ツール]メニューの[環境設定]

通信設定は、プログラム実行中には変更できません。それ以外については、プログラム実行中の状態に関係なく、設定が可能です。

### ■ [ヘルプ]メニュー

エミュレーション実行中の状態に関係なく、全ての機能が使用できます。

### ■ [実行]メニューの[リアルタイム LCD モニタの呼び出し]

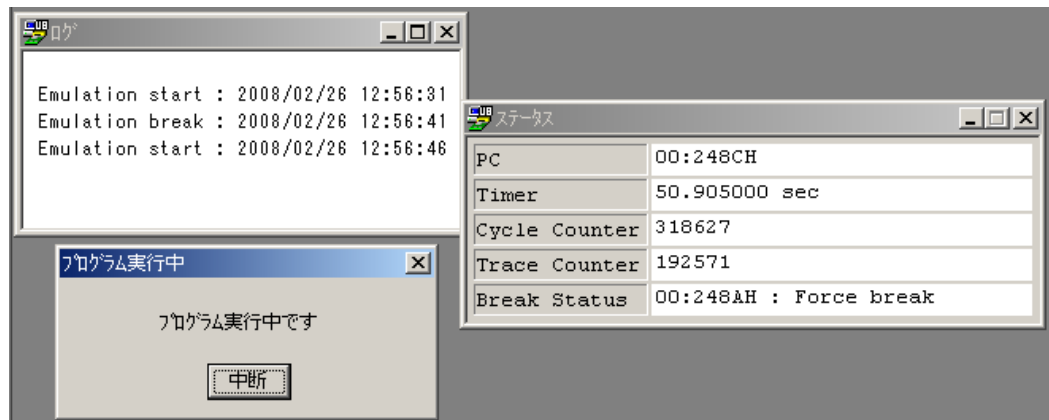
LCD イメージチェックツールを起動し、リアルタイムに LCD 表示を更新することができます。

Dr.U8 ICE  
Dr.U16 ICE

## 6.4 実行時間の計測

DTU8 デバッガは、実行時間計測機能をサポートします。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



上の図は、Dr.U8 ICE モード時の表示を示しています(ステータスウィンドウの表示内容は、動作モードにより異なります)。

Dr.U8 ICEモード、Dr.U16 ICEモード、Dr.ICEモード、uEASEモードおよびnanoEASEモードでは、エミュレーション実行中にステータスウィンドウを表示すると、[Timer]フィールドにエミュレーション開始からの実行時間を表示します。<sup>25</sup>

シミュレーションモードでは、シミュレーション実行中にステータスウィンドウを表示すると、[Time]フィールドにシミュレーション開始からの実行時間を表示します。<sup>26</sup>

また、[ツール]メニューの[環境設定]で開くダイアログボックスの[その他]タブにある[プログラム実行の開始/終了時刻をログに出力する]をチェックすると、ログウィンドウに、毎エミュレーションの開始時刻と終了時刻が記録されます。<sup>27</sup>

### ■ 計測時間のクリア

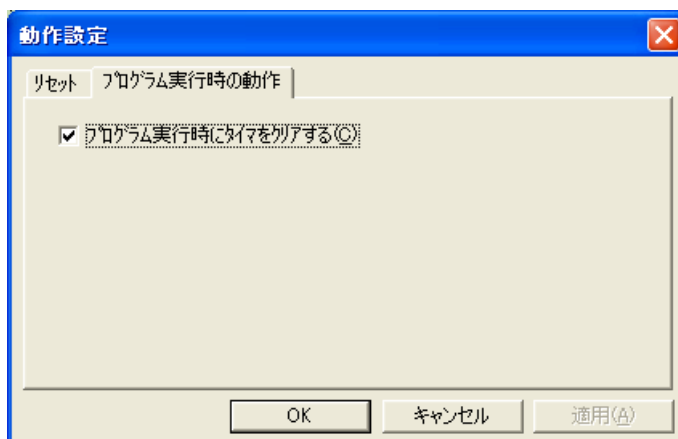
エミュレーション開始時に計測時間をクリアするかどうかを[ツール]メニューの[動作設定]で開くダイアログの[プログラム実行時の動作]タブで設定することができます。[プログラム実行時にタイマをクリアする]チェックボックスをチェックすると、エミュレーションを開始時に計測時間がクリアされます。ただし、[実行]メニューの[リセット後に実行]でエミュレーションを開始した場合は必ず計測時間はクリアされます。また、[実行]メニューの[リセット]でも計測時間はクリアされます。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

<sup>25</sup> この実行時間計測は、ハードウェア機能で実現されています。このため、精度や計測可能時間は、接続したエミュレータのハードウェア機能に依存するので、エミュレータ付属のマニュアルを参照してください。

<sup>26</sup> シミュレーションモードでの実行時間は、シミュレーションエンジンに内蔵されたサイクルカウンタを時間に換算したものになります。

<sup>27</sup> ログに表示される実行開始時刻と終了時刻は、DTU8 が動作しているパソコンのカレンダータイマを使用しているため、精度を保証できません。



## 6.5 エミュレーション実行中の終了と再接続

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

エミュレータがエミュレーション実行している最中でも、DTU8 デバッガを終了することができます。そして、同じ PC で DTU8 デバッガを起動すると、エミュレータに再接続することができます。

この機能により、長時間、あるいは、数日にも及ぶことが想定されるエージング評価等で、ユーザ応用回路に接続したエミュレータを実行させておき、DTU8 デバッガを動作させているパソコンを他の用途で使うことや、電源を OFF することが可能になります。

DTU8 デバッガのエミュレーション実行中の終了と再接続機能を使用する際には、以下の事柄に注意してください。

### ■ 同じ DTU8 デバッガを使って再接続を行ってください。

エミュレーション実行中に DTU8 デバッガを終了するとき、DTU8 デバッガは、ウィンドウの設定状態をプロジェクトファイルに自動で上書き保存するとともに、再接続に必要な情報を ini ファイルに保持します。このため、他のパソコンの DTU8 デバッガでは、再接続できません。

### ■ プログラムファイルの更新はしないでください。

再接続を行うと、DTU8 デバッガはブレーク後にエミュレーション実行しているプログラムファイルの再ロードを行います。エミュレーション実行しているプログラムコードが更新されていると、エミュレータと DTU8 デバッガとの間でプログラムコードの不整合が発生します。

### ■ 再接続時にはブレークするまで使用できる機能が制限されます。

再接続時にエミュレーション実行が継続していたときは、ブレークするまでは通常のエミュレーション中の制限に加えて以下の機能が制限されます。

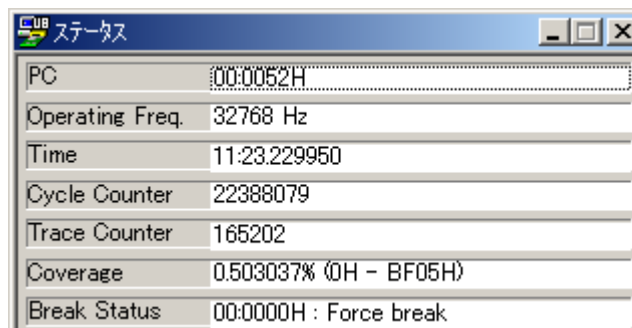
- ・ ソースウィンドウは開くことができません。
- ・ リアルタイムウォッチ機能は使用できません。
- ・ 「ブレーク条件の設定」は確認することができません。
- ・ リアルタイム LCD モニタ機能は使用できません。

## 6.6 サイクルカウンタ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

サイクルカウンタは、プログラム実行時に命令サイクルをカウントするためのカウンタです。サイクルカウンタは、DTU8 デバッガのステータスウィンドウの[Cycle Counter]フィールドに表示されます。

28

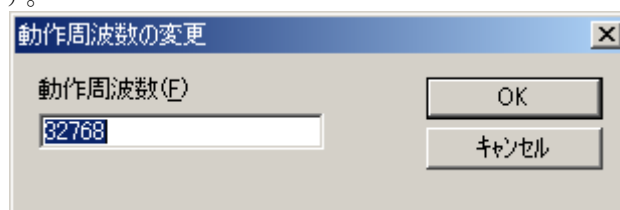


Simulate

シミュレーションモード時にステータスウィンドウに表示される[Time]は、このサイクルカウンタの値と[Operating Freq]フィールドで表示されている動作周波数をもとに算出されます。

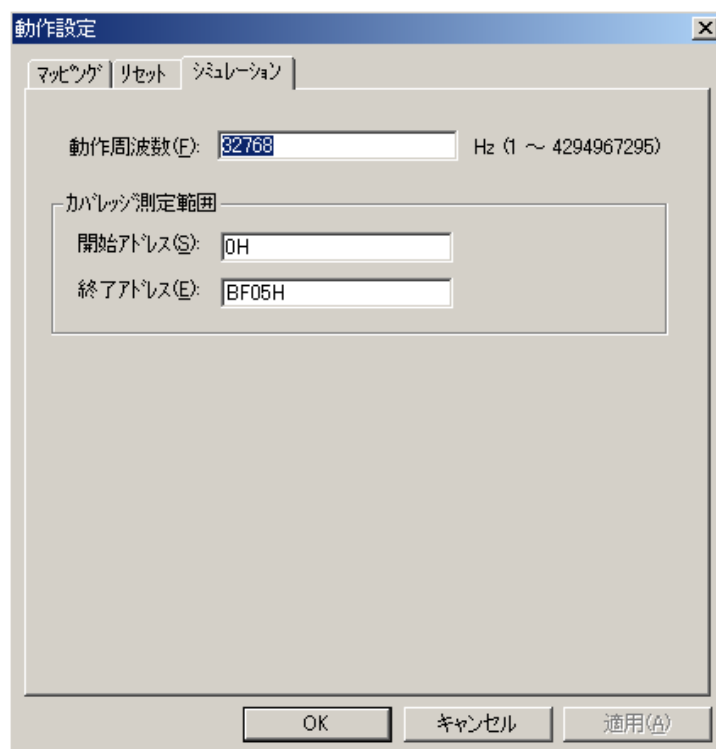
[Operating Freq]フィールドの動作周波数は、ステータスウィンドウのポップアップメニューの[動作周波数の設定]または、[ツール]メニューの[動作設定]で開くダイアログの[シミュレーション]タブで変更することができます。

ステータスウィンドウのポップアップメニューの[動作周波数の設定]を選択すると、次のダイアログボックスが表示されます。



[ツール]メニューの[動作設定]で開くダイアログの[シミュレーション]タブを選択すると、以下のダイアログボックスが表示されます。

<sup>28</sup> サイクルカウンタは、シミュレーションモードと Dr. U8 ICE モード、Dr. U16 ICE モード、Dr. ICE モード時(ただし、旧式の Dr. 610XXX インサーキットエミュレータ接続時を除く)のみ使用可能です。サイクルカウンタのビット長は、シミュレーションモードの場合 64 ビット、Dr. U8 ICE モード、Dr. U16 ICE モード、Dr. ICE モードの場合 32 ビットです。サイクルカウンタをサポートしていない場合は、[Cycle Counter]フィールドは表示されません。



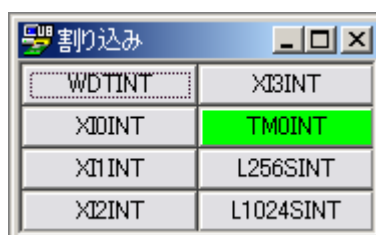
## 6.7 割り込みシミュレーション

Simulate

シミュレーションモードでは、割り込みシミュレーション機能をサポートしています。割り込みウィンドウにより、任意の箇所で割り込みを発生させることができます。

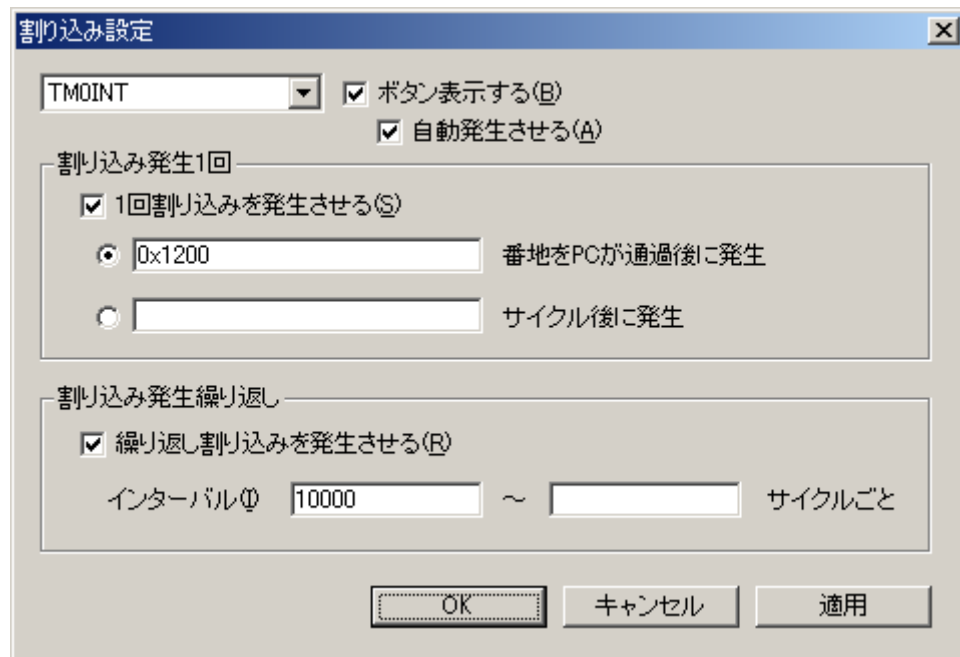
### 6.7.1 割り込みウィンドウ

割り込みウィンドウは、[表示]メニューの[割り込み]を選択することにより表示されます。



割り込みウィンドウには、ターゲット MCU が持つ割り込み要因が、ボタンとして表示されます（このボタンを割り込み要因ボタンと表現します）。これらのボタンを左クリックすることにより、対応する割り込み要求を発生させることができます。

これらの割り込み要因ボタンに対して、さまざまな設定を行うことが可能です。各割り込み要因ボタンに対する設定は、設定する割り込み要因ボタン部分をマウスで右クリックし、ポップアップメニューの[割り込み設定...]を選択して表示される[割り込み設定]ダイアログから行います。



[割り込み設定]ダイアログは、マウスで右クリックを行った割り込み要因ボタンの割り込みが選択された状態となり、割り込み設定を変更することができます。

[OK]ボタンをクリックすると、現在表示中の割り込み設定が割り込みウィンドウへ反映され、ダイアログを閉じます。

[キャンセル]ボタンをクリックすると、現在表示中の割り込み設定は割り込みウィンドウへ反映されずに、ダイアログを閉じます。

[適用]ボタンをクリックすると、現在表示中の割り込み設定が割り込みウィンドウへ反映されます。この場合、ダイアログは閉じません。[割り込み設定]ダイアログの左上にある割り込み要因の種類をリストボックスから選択し、別の割り込み要因を続けて設定することができます。

各設定項目について、以下に説明します。

### ■ ボタン表示する

このチェックボックスをチェックすると、割り込みウィンドウにボタンを表示します。[自動発生させる]がチェックされている場合は、ボタンを非表示にすることはできません。

### ■ 自動発生させる

このチェックボックスをチェックすると、[割り込み開始]および[繰り返し]の項目が有効になり、設定された条件で割り込みを自動的に発生させることができます。自動発生させる場合、[割り込み開始]か[繰り返し]のどちらか、または両方を設定する必要があります。

また、自動発生を設定した割り込み要因であることが識別できるよう、ボタンの色が変化します。ボタンの色は、[環境設定]ダイアログの[書式]タブで変更可能です。

### ■ 割り込み発生 1 回

[1回割り込みを発生させる]をチェックすると、チェックボックス以下の項目が有効となります。

指定したアドレスを通過した時に割り込み要求を発生させるか、または、現在のプログラムの位置を0として指定サイクルだけ命令を実行した後に割り込み要求を発生させるかを指定します。

アドレスを入力する場合、アドレスの入力範囲はプログラム領域の範囲でなければなりません。入力がない場合や範囲外のアドレスが指定された場合にはエラーダイアログが表示され、再入力を促します。

サイクル数を入力する場合、サイクル数の入力範囲は、1～0xFFFFFFFFFFFFFFFF(10 進入力可)となります。入力がない場合や範囲外のサイクル数が指定された場合にはエラーダイアログが表示され、再入力を促します。

割り込み開始の 2 項目は、指定割り込みが1回発生すると、割り込みウィンドウのボタンは、自動発生を終了したことが分かるように、ボタンの色が自動発生未設定の色に変わります。

### ■ 割り込み発生繰り返し

[繰り返し割り込みを発生させる]をチェックすると、チェックボックス以下の項目が有効となり、割り込みのインターバル値を入力できるようになります。インターバルを入力する場合、インターバルの入力範囲は、1～0xFFFFFFFFFFFFFFFF(10 進入力可)となります。入力がない場合や範囲外のサイクル数が指定された場合にはエラーダイアログを表示し、再入力を促します。左側の欄にのみ入力した場合には、必ずそのサイクル数で割り込み要求を発生させます。範囲指定(右側の欄も使用)すると、DTU8 はその範囲内でインターバルの値を割り込みのたびにランダムに決定します。

割り込みシミュレーションはインターバルカウンタ値を、現在のプログラムの位置を0としてカウントアップを開始します。カウントアップ値が指定サイクルになると割り込みが発生します。割り込みが発生するとカウンタ値を 0 として、カウントアップを継続し、繰り返し割り込みを発生させます。



## 7 章 ブレーク機能

### 7.1 ブレークポイント

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

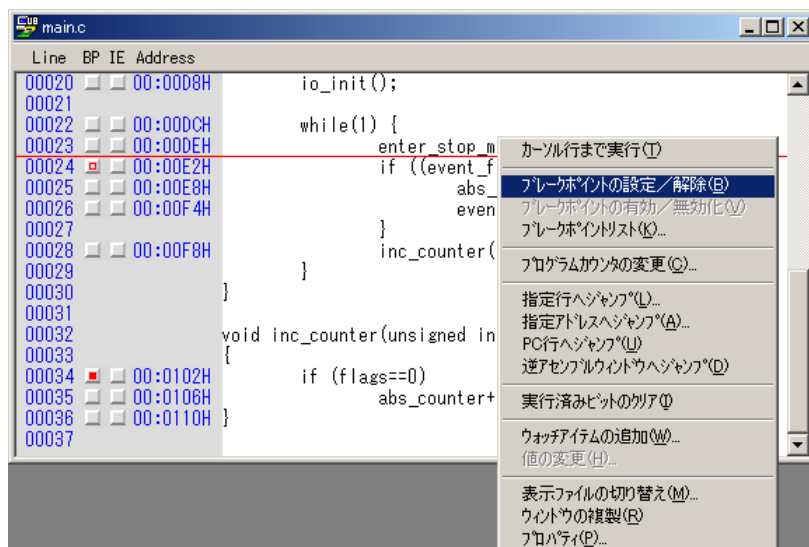
DTU8 デバッガは、任意のコードアドレスにブレークポイントを設定することができます。プログラム実行中に、有効なブレークポイントを設定したアドレスを PC が通過する直前にブレークが発生します。

#### 7.1.1 ソース／逆アセンブルウィンドウ上での設定方法

ソースウィンドウや逆アセンブルウィンドウを使用して、ブレークポイントの設定/解除や無効化/有効化するには、いくつかの方法があります。

##### ■ ウィンドウの BP ボタン

ソースウィンドウ、または、逆アセンブルウィンドウのポップアップメニューの[プロパティ]で開くダイアログボックスの中の[アトリビュート]をチェックすると、ウィンドウ上に BP ボタンの列が並びます。



この BP ボタンは、ブレークポイントの設定可能な行と設定状態を表しています。

特に、ソースウィンドウでは、BP ボタンを持つ行は、実行コードを持つことを示し、かつ、ブレークポイントを設定できる行にもなります。

赤い■マークが付いているBPボタンは、有効なブレークポイントが設定されているアドレスであることを示します。赤い□マークが付いているBPボタンは、無効化されたブレークポイントが設定されていることを示します。何も付いていないBPボタンは、ブレークポイントが設定されていません。<sup>29</sup>

<sup>29</sup> C ソース行についている BP ボタンの赤い■と□マークは、その行のアドレス範囲の中にブレークポイントがあることを示します。

ブレークポイントが設定されていない BP ボタンをマウスでクリックすると、ブレークポイントを設定します。既にブレークポイントが設定されている BP ボタンをマウスでクリックすると、ブレークポイントを解除します。

### ■ ウィンドウのポップアップメニュー

ソースウィンドウ、または、逆アセンブルウィンドウで、ブレークポイントを設定したい行にカーソルを移動して、それぞれのウィンドウのポップアップメニューの[ブレークポイントの設定/解除]を指定すると、カーソル位置のアドレスにブレークポイントが未設定であればブレークポイントが設定されます。既に、ブレークポイントが設定されている場合には、ブレークポイントを解除します。

また、ブレークポイントが設定されている行であれば、ポップアップメニューの[ブレークポイントの有効/無効化]を使用すると、一時的にブレークポイントを無効化できます。無効化されているブレークポイントは、再度、[ブレークポイントの有効/無効化]で有効化できます。

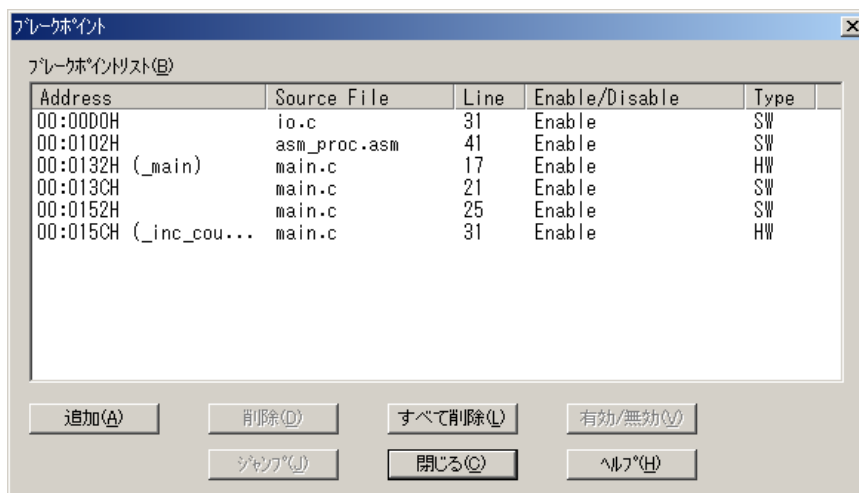
### ■ [実行]メニューの[ブレークポイントの設定/解除]

ソースウィンドウ、または、逆アセンブルウィンドウで、ブレークポイントを設定したい行にカーソルを移動して、[実行]メニューの[ブレークポイントの設定/解除]を指定すると、カーソル位置のアドレスにブレークポイントが未設定であれば設定されます。既に、ブレークポイントが設定されている場合には、ブレークポイントを解除します。<sup>30</sup>

#### 7.1.2 一覧表示

[実行]メニューやソース/逆アセンブルウィンドウのポップアップメニューから[ブレークポイントリスト]を選択すると、設定されているブレークポイントの一覧を表示します。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



- |            |   |
|------------|---|
| [追加]ボタン    | ダイアログボックス上から、直接ブレークポイントを追加します。                |
| [削除]ボタン    | 選択したブレークポイントを削除します。                           |
| [すべて削除]ボタン | すべてのブレークポイントを削除します。                           |
| [ジャンプ]ボタン  | 選択したアドレスに対応するソースウィンドウもしくは逆アセンブルウィンドウにジャンプします。 |

<sup>30</sup> [実行]メニューの[ブレークポイントの設定/解除]は、ショートカットキーとして、[F9]キーに割り付けられています。



### 7.1.3 ブレークポイントブレーク全体の有効化と無効化

ブレークポイントブレーク機能自体を有効/無効化することができます。

[実行]メニューの[ブレークポイントブレークを有効にする]を指定すると、チェックマーク(✓)がトグルします。

このメニューにチェックマーク(✓)が付いているときには、ブレークポイントブレークが有効な状態であることを表します。また、チェックマーク(✓)が外れているときには、ブレークポイントが設定されていても、ブレークポイントでブレークしなくなります。

実行(R)	ツール(T)	ウィンドウ(W)	ヘルプ(H)
リセット(R)			Ctrl+F5
プログラムカウンタの変更(P)...			
プログラム実行(E)			F5
カーソルまで実行(I)			F7
リセット後に実行(U)			
強制ブレーク(B)			Shift+F5
ステップオーバー(S)			F10
ステップイン(I)			F11
ブレークポイントの設定/解除(B)			F9
ブレークポイントリスト(K)...			
全ブレークポイントの解除(A)			
✓ ブレークポイントブレークを有効にする(V)			
ブレーク条件の設定(O)...			
シンクアウトポイントの追加(Y)...			
シンクアウトポイントリスト(N)...			
全シンクアウトポイントの解除(L)			
シンクアウト信号のリセット(O)			


[実行]メニューの[ブレークポイントブレークを有効にする]メニューには、ツールバーの中の  トグルボタンに割り当てられていて、ブレークポイントブレークが有効な場合は、 トグルボタンが押下された(へこんだ)状態となります。


### 7.1.4 C ソース行に設定するブレークポイント

#### ■ C ソース行では先頭に設定

C ソースの 1 行は、そのほとんどが複数の命令コードで構成されています。ソースウィンドウ上で表示している C ソース行に対してブレークポイントを設定すると、その行の先頭アドレスにブレークポイントを設定します。

#### ■ C ソース行の BP フィールド表示

BP フィールドに表示する赤い  は、対応する C ソース行のアドレス範囲に有効なブレークポイントが設定されていることを示します。

BP フィールドに表示する赤い  は、対応する C ソース行のアドレス範囲に設定されているブレークポイントの全てが無効化されていることを示します。

#### ■ C ソース行のブレークポイントの解除や無効/有効化

ソースウィンドウで、C ソース行に設定されているブレークポイントを解除するとき、対応する C ソー

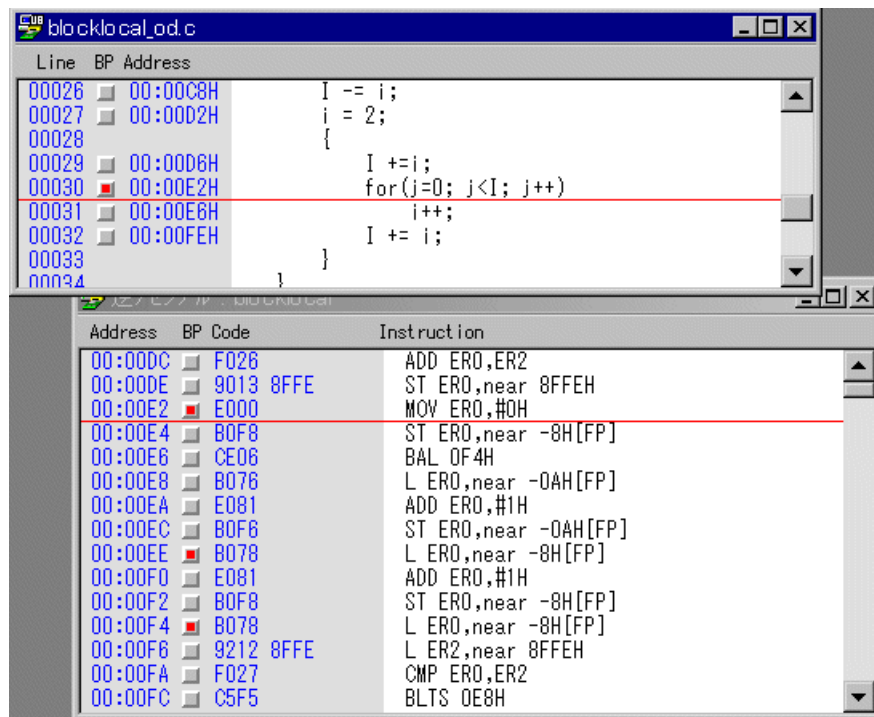
ス行のアドレス範囲に設定されている全ブレークポイントが解除されます。

また、Cソース行のブレークポイントを無効/有効化も、対応するCソース行のアドレス範囲に設定されている全ブレークポイントが対象になります。

逆アセンブルウィンドウやブレークポイントリストを使用して、Cソース行の先頭アドレス以外に設定した場合に、ソースウィンドウで対応ソース行を確認できますが、解除や無効/有効化する際には注意してください。

### ■ 複数の領域に展開されるCソース

Cソースの記述によっては、1行が複数の領域に展開されることがあります。例えば、以下の図の中のソースウィンドウの00030行目のfor文は、Cソースを最適化せずにコンパイルすると、3箇所に展開されます。このようなときには、ソースウィンドウにブレークポイントを設定すると、逆アセンブルウィンドウで表示されているように3箇所のアドレスに設定されます<sup>31</sup>。



### 7.1.5 ブレークポイント設定時の注意事項<sup>32</sup>

#### ■ ベクタ領域のブレークポイント

逆アセンブルウィンドウ上で、ベクタテーブル領域(0:0h~0:0FFh)に対するブレークポイントの設定する際は、十分に、注意してください。

ブレークポイントとして設定したアドレスをベクタアドレスとするリセット及び割り込みが発生したとき、その実行結果は、予想できないものとなります。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

<sup>31</sup> uEASE および nanoEASE では、Cソースの1行が複数の領域に展開される場合、先頭の領域のアドレス1箇所のみブレークポイントが設定されます。

<sup>32</sup> DTU8 デバッガのブレークポイントは、実行コードを置き換えるタイプのソフトウェアブレークを使用しています。このため、実行中に置き換えたコードをデータアクセスで参照すると、予想できない結果になります。

### ■ テーブル領域のブレークポイント

テーブルデータ領域(命令実行で参照するデータアドレス)に、ブレークポイントを設定しないでください。

ブレークポイントを設定したアドレスに対してデータ転送命令等でアクセスした場合、そのアクセスデータは正常な値になりません。特にストア命令でデータを書き換えると、そのエミュレーション中は書き換えたアドレスでのブレークポイントブレークは発生しなくなります。また、他の要因でエミュレーションがブレークすると、その書き換えたデータはエミュレーション実行前のデータに戻ります。

## 7.2 強制ブレーク

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ブレークポイントやその他のブレーク条件が設定されていない場合、プログラムはブレークすることなく実行し続けます。

DTU8 デバッガでプログラムの実行を強制的に中断する場合は、[実行]メニューの[強制ブレーク]を選択します。また、[ツール]メニューの[環境設定]の[その他] プロパティシートで[プログラム実行中にエミュレーション中ダイアログを表示する]がチェックされている場合は、エミュレーション中ダイアログの[中断]ボタンを押すことでエミュレーションを中断できます。

## 7.3 ブレーク条件の設定

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガで[実行]メニューの[ブレーク条件の設定]を選択すると、次のようなダイアログボックスが表示されます。

[ブレーク条件の設定]ダイアログボックスで設定するブレーク条件は、接続されているエミュレータの機能に依存したブレーク条件となります。使用できないブレーク条件はグレースアウト表示され、入力が無効になります。

### ■ トレースカウントブレーク

トレースメモリのトレースカウンタ値が設定値に達したときブレークが発生します。トレースカウントブレークを設定すると、プログラム実行中にトレースされた分がカウンタからデクリメントされ、0 になるとブレークが発生します。

### ■ パワーダウnbrレーク

プログラム実行中にパワーダウンモード (HALT や STOP モード) になると、ブレークが発生します。

### ■ アドレスブレーク

ソースウィンドウや逆アセンブルウィンドウで設定するブレークポイントとは異なり、このブレークには実行カウント数を指定することができます。

[ブレークアドレス]に設定したアドレスが[設定値]に指定した回数だけ実行されるとブレークが発生します。カウント値を最初に設定する場合は、[設定値]に回数を入力後[>>]ボタンを押してください

い。

#### ■ エクスターナルブレーク

エミュレータに接続した外部ブレーク信号入力プロンプトに、[エクスターナルブレーク条件]で設定した信号入力でブレークします。

#### ■ ROM の N/A 領域アクセスブレーク

未使用領域やプログラム実行不可領域を実行したときにブレークが発生します。通常は常にチェックしてください。

#### ■ RAM の N/A 領域アクセスブレーク

未使用領域やデータ転送命令でアクセスできない領域をアクセスしたときにブレークが発生します。通常は常にチェックしてください。

#### ■ RAM マッチブレーク

プログラム実行中にアクセスする RAM アドレスを条件としてブレークが発生します。

[RAM アドレス]フィールドには、チェックする RAM アドレスを指定します。[RAM アドレスマスク]を指定した場合、プログラム実行中にアクセスした RAM アドレスと[RAM アドレスマスク]に指定した値とのビット論理積(AND)の結果が対象 RAM アドレスとなります。

[アクセス方法]フィールドには、プログラム実行中のアクセス方法を指定します。[Read], [Write], [Read/Write]から選択してください。

[マッチデータ指定]をチェックすると、指定された RAM アドレスのデータをチェックしてブレークが発生します。チェックするデータ値を[RAM データ]に指定し、データのサイズを[アクセス単位]に指定します。また、チェックするデータ値には[RAM データマスク]で指定するマスク値を適用できます。[RAM データマスク]を指定した場合、条件に合致した RAM アドレスの内容と[RAM データマスク]に指定した値とのビット論理積(AND)した結果が[RAM データ]の値と比較されます。

[成立条件]には、[Equal]または[Not Equal]を指定します。

RAM マッチブレークは 2 つまで指定することができ、その 2 つの RAM マッチを AND 条件もしくは OR 条件で組み合わせて使用することができます。OR 条件が指定された場合、2 つの RAM マッチのうち、どちらか一方が成立するとブレークが発生します。AND 条件が指定された場合、2 つの RAM マッチの両方が成立するとブレークが発生します。

#### ■ パスカウント指定

RAM マッチ条件の組み合わせによって、パスカウント指定が可能になります。[RAM マッチパスカウント]をチェックすると、条件の成立回数を指定することができます。また、[カウント値]の表示は、エミュレータに設定されている回数値を示します。他の条件でブレークした場合には、更新されていることがあります。[>>]ボタンで、再設定できます。

Dr.U8 ICE モード、Dr.U16 ICE モードの場合は、2 つの RAM マッチそれぞれについてパスカウントを指定することが可能です。

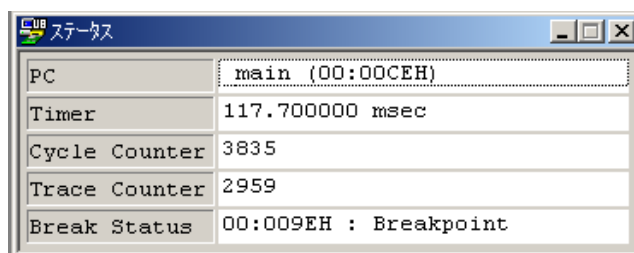
## 8 章 メモリとレジスタの参照と変更

### 8.1 プログラムカウンタ

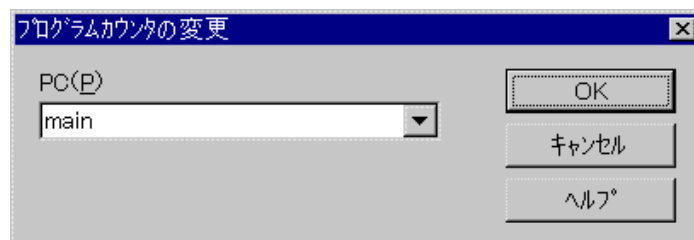
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

#### ■ ステータスウィンドウ

ステータスウィンドウの[PC]フィールドは、プログラムカウンタの内容を表示しています。



プログラムカウンタの内容を変更するには、ステータスウィンドウのポップアップメニューの[プログラムカウンタの変更]を選択するか、値が表示されている部分をダブルクリックしてください。



#### ■ ソースウィンドウと逆アセンブルウィンドウ

ソースウィンドウと逆アセンブルウィンドウは、プログラムカウンタに対応する行を緑の反転表示で示しています。

さらに、ソースウィンドウと逆アセンブルウィンドウのポップアップメニューの[プログラムカウンタの変更]を指定すると、カーソル位置のアドレスがデフォルト値と設定された[プログラムカウンタの変更]ダイアログボックスが開きます。

#### ■ その他

ウォッチウィンドウのウォッチ項目としてPCを登録して、参照できます。[実行]メニューの[プログラムカウンタの変更]を指定すると、プログラムカウンタを変更できます。<sup>33</sup>

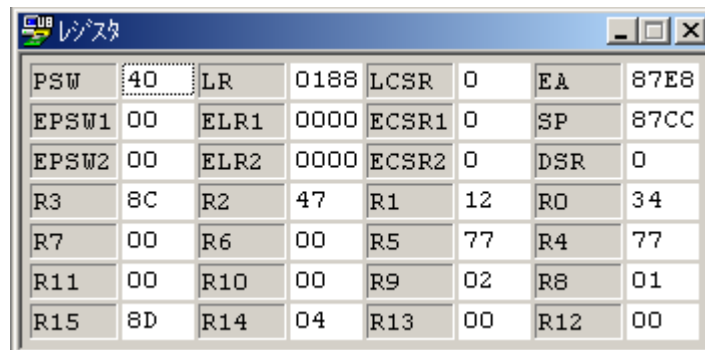
<sup>33</sup> N/A 領域や TEST 領域にマッピングされたアドレスを指定することはできません。



## 8.2 レジスタ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

各種レジスタは、レジスタウィンドウに表示されます。



PSW	40	LR	0188	LCSR	0	EA	87E8
EPSW1	00	ELR1	0000	ECSR1	0	SP	87CC
EPSW2	00	ELR2	0000	ECSR2	0	DSR	0
R3	8C	R2	47	R1	12	R0	34
R7	00	R6	00	R5	77	R4	77
R11	00	R10	00	R9	02	R8	01
R15	8D	R14	04	R13	00	R12	00

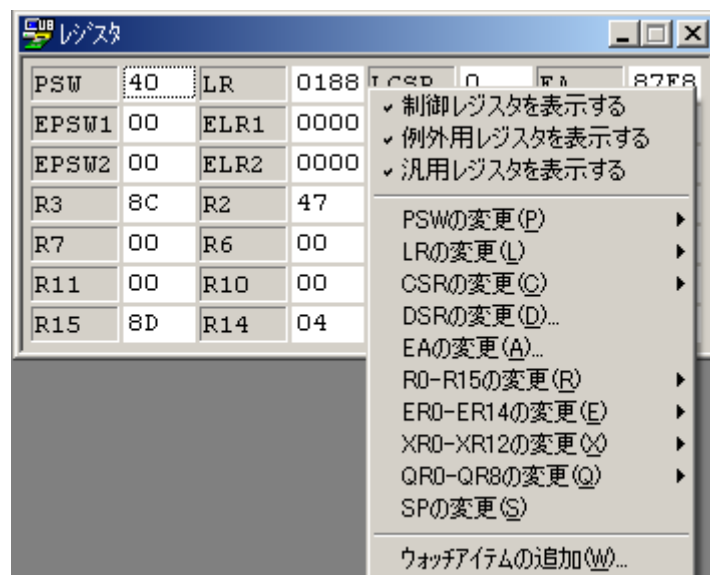
レジスタウィンドウに表示されているレジスタの内容を変更するには、レジスタウィンドウのポップアップメニューの[xxx の変更]を選択するか、各レジスタの値が表示されている部分をダブルクリックしてください。

また、参照したいレジスタをウォッチウィンドウのウォッチ項目に登録することもできます。

### ■ レジスタウィンドウのカスタマイズ

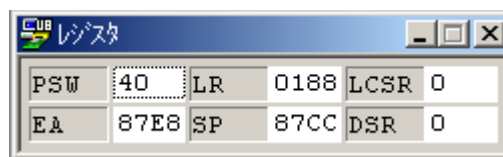
必要なレジスタだけをレジスタウィンドウに表示することができます。

レジスタウィンドウのポップアップメニューで、制御レジスタの表示、例外用レジスタの表示および、汎用レジスタの表示を選択できます。



### ■ 制御レジスタの表示

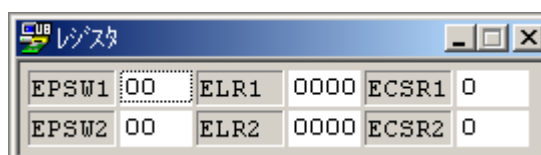
[制御レジスタを表示する]をチェックすると、以下のレジスタ群をレジスタウィンドウへ表示します。



PSW	40	LR	0188	LCSR	0
EA	87E8	SP	87CC	DSR	0

### ■ 例外用レジスタの表示

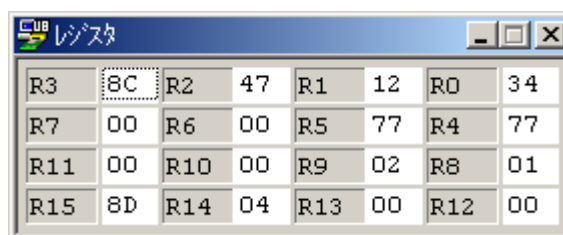
[例外用レジスタを表示する]をチェックすると、以下のレジスタ群をレジスタウィンドウへ表示します。



EPSW1	00	ELR1	0000	ECSR1	0
EPSW2	00	ELR2	0000	ECSR2	0

### ■ 汎用レジスタの表示

[汎用レジスタを表示する]をチェックすると、以下のレジスタ群をレジスタウィンドウへ表示します。



R3	8C	R2	47	R1	12	R0	34
R7	00	R6	00	R5	77	R4	77
R11	00	R10	00	R9	02	R8	01
R15	8D	R14	04	R13	00	R12	00

これらの表示レジスタ群を組み合わせることにより、必要なレジスタだけをウィンドウに表示させることができます。

## 8.3 SFR

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

SFRは、SFRウィンドウに表示されます。各種SFRの内容を変更する場合は、SFRウィンドウのポップアップメニューの[値の変更]を選択するか、各SFRが表示されている部分をダブルクリックしてください。ビットSFRが表示されている部分をダブルクリックすると、ビットSFRの値を変更できます。<sup>34</sup>

SFR Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DSR								DSR0
00								0
STPACP								
Write Only								
SBYCON							STP	HLT
00							0	0
FCON0							ENOSC	SYSCLK
00							0	0
FCON1				PLL	OSCM1	OSCM0	SYSCL	SYSCLK
00				0	0	0	0	0
LTBR	LTB32768	LTB16384	LTB8192	LTB4096	LTB2048	LTB1024	LTB512	LTB256
00	0	0	0	0	0	0	0	0

### ■ 検索

SFR を検索する場合は、SFR ウィンドウのポップアップメニューの[検索]を選択します。ダイアログボックスに目的の SFR 名を指定してください。

### ■ ウォッチウィンドウへの追加

現在のカーソル位置にあるSFRをウォッチウィンドウに登録する場合は、SFRウィンドウのポップアップメニューの[ウォッチアイテムの追加]を選択します。

## 8.4 コードメモリ、データメモリ、物理セグメント1以上のメモリ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### ■ 各種メモリウィンドウの表示

コードメモリのダンプウィンドウを表示するには、[表示]メニューの[コードメモリ]を選択します。コードメモリウィンドウでは、物理セグメント0のコードアドレス空間の内容を表示／変更することができます。命令モニタとインストラクションコードを参照する場合は逆アセンブルウィンドウの方が便利ですが、テーブルデータなどを参照する場合は、コードメモリウィンドウが有効です。<sup>35</sup>

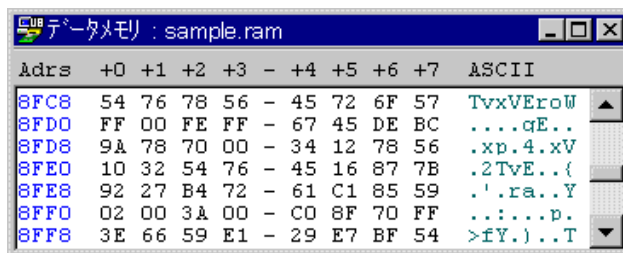
Adrs	+0	+1	+2	+3	+4	+5	+6	+7
0000	D2	87	3A	00	3C	00	1F	FE
0008	30	00	00	E0	34	00	00	01
0010	02	03	04	05	06	07	08	09
0018	0A	0B	0C	0D	0E	0F	FE	FE
0020	FE	FD	FC	FB	FA	F9	F8	F7
0028	F6	F5	F4	F3	F2	F1	F0	70

<sup>34</sup> SFR に外部回路が接続されている場合は、エミュレーションのブレーク発生時に、外部回路へのアクセスが起こります。また、SFR ウィンドウが表示されている状態では、DTU8 の画面更新時に、SFR へのアクセスが起こります。

<sup>35</sup> コードメモリウィンドウは、[ツール]メニューの[環境設定]で表示される環境設定ダイアログボックスの[その他]タブで、[ウィンドウメニューの” 最新の状態に更新” を選択した時にコードメモリも更新する]をチェックしておけば、更新することができます。

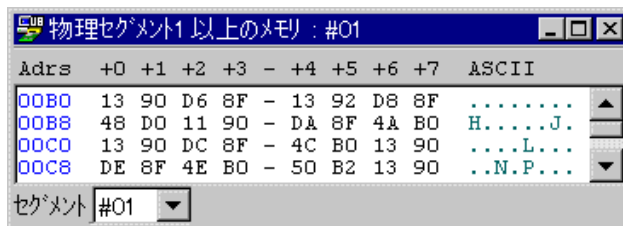
データメモリのダンプウィンドウを表示するには、[表示]メニューの[データメモリ]を選択します。データメモリウィンドウでは、物理セグメント 0 のデータアドレス空間の内容を表示／変更することができます。

データメモリウィンドウでは、SFR 領域がすべて”\*”で表示されます。また、SFR の内容を変更することはできません。



Adrs	+0	+1	+2	+3	-	+4	+5	+6	+7	ASCII
8FC8	54	76	78	56	-	45	72	6F	57	TvxVEroW
8FD0	FF	00	FE	FF	-	67	45	DE	BC	....qE..
8FD8	9A	78	70	00	-	34	12	78	56	.xp.4.xV
8FE0	10	32	54	76	-	45	16	87	7B	.2TvE..{
8FE8	92	27	B4	72	-	61	C1	85	59	..'.ra..Y
8FF0	02	00	3A	00	-	C0	8F	70	FF	.....p.
8FF8	3E	66	59	E1	-	29	E7	BF	54	>fY.)..T

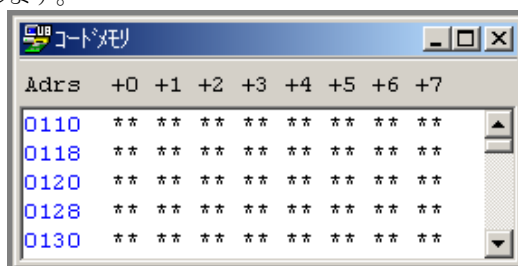
物理セグメント1以上のメモリのダンプウィンドウを表示するには、[表示]メニューの[物理セグメント1以上のメモリ]を選択します。物理セグメント1以上のメモリウィンドウでは、物理セグメントが1以上のメモリ空間の内容を表示／変更することができます。



Adrs	+0	+1	+2	+3	-	+4	+5	+6	+7	ASCII
00B0	13	90	D6	8F	-	13	92	D8	8F	.....
00B8	48	D0	11	90	-	DA	8F	4A	B0	H.....J.
00C0	13	90	DC	8F	-	4C	B0	13	90	....L...
00C8	DE	8F	4E	B0	-	50	B2	13	90	..N.P...

セグメント #01

コードメモリウィンドウ、データメモリウィンドウ、および物理セグメント 1 以上のメモリウィンドウを開く際にエラーが発生した場合、DTU8 はエラーメッセージを表示し、エラーが発生した行以降のデータをすべて”\*”で表示します。

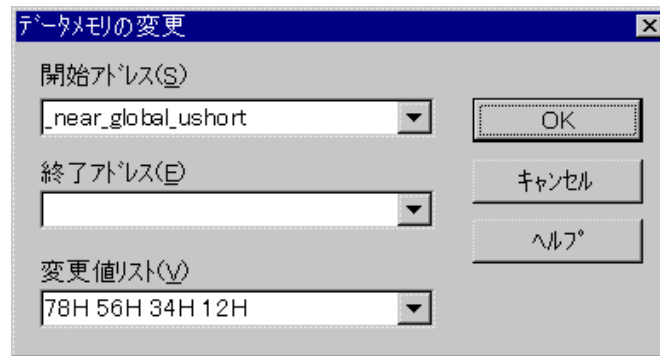


Adrs	+0	+1	+2	+3	+4	+5	+6	+7
0110	**	**	**	**	**	**	**	**
0118	**	**	**	**	**	**	**	**
0120	**	**	**	**	**	**	**	**
0128	**	**	**	**	**	**	**	**
0130	**	**	**	**	**	**	**	**

### ■ データの変更<sup>36</sup>

ダンプウィンドウのポップアップメニューの[xxx メモリの変更]を選択すると、次のダイアログボックスが開き、データを変更することができます。

<sup>36</sup> N/A 領域や TEST 領域にマッピングされたアドレスの内容は変更することはできません。



変更値リストには、空白、カンマ(,), セミicolon(;)のいずれかで区切った数値のリストを指定します。コードメモリ、および物理セグメント1以上のメモリの場合は、変更値リストにワードデータを指定することができます。<sup>37</sup>

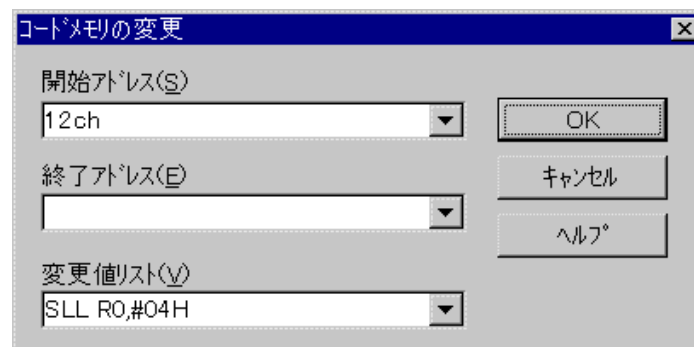
常に、[開始アドレス]の値を指定する必要があります。[終了アドレス]を指定すると、[開始アドレス]から[終了アドレス]の間のメモリの内容を、変更値リストの繰り返しの値に変更します。<sup>38</sup>

[物理セグメント 1 以上のメモリの変更]ダイアログで、[開始アドレス]と[終了アドレス]へアドレスを入力する場合、対象となる物理セグメントを明示的に指定してください。例えば、物理セグメント1の2000H のアドレスを指定する場合、01H:2000H と指定してください。

## ■ 命令ニーモニックでメモリの変更

コードメモリおよび物理セグメント1以上のメモリを変更する場合には、変更値リストにアセンブリ言語を記述することができます。

アセンブリ言語を指定した場合は、[OK]ボタンではダイアログボックスが閉じずに、開始アドレスが次のアドレスに更新されるため、プログラミングをするように、メモリが変更できます。

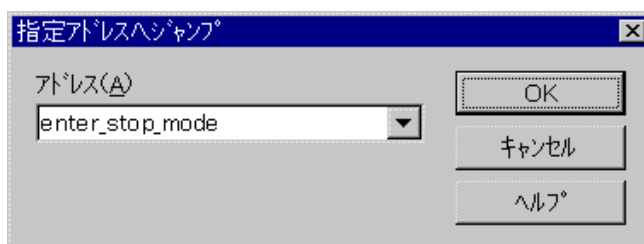


## ■ 指定アドレスヘジャンプ

コードメモリ、データメモリ、及び物理セグメント 1 以上のメモリウインドウのポップアップメニューの[ジャンプ]を選択すると、次のダイアログボックスが開き、指定アドレスヘジャンプできます。

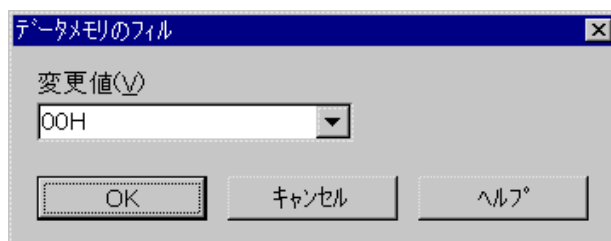
<sup>37</sup> コードメモリを変更すると、現在表示中のソースウインドウとコードメモリの内容が一致しくなくなります。[ツール]メニューの[環境設定]で表示される環境設定ダイアログのその他タブで、[デバッグ情報ロード後のコード変更時に確認メッセージを出力する]をチェックしておけば、コードメモリの変更時に確認することができます。

<sup>38</sup> 開始アドレスが奇数アドレスで、変更値リストに指定した値がワード単位の値であった場合、開始アドレスは偶数アドレスに補正されます。例えば、開始アドレスが 2001H で、変更値リストが 1234H であった場合、開始アドレスは 2000H に補正され、2000H に 34H が、2001H に 12H が書き込まれます。



### ■ データのフィル

コードメモリ、データメモリ、及び物理セグメント1以上のメモリインドウのポップアップメニューの[フィル]を選択すると、次のダイアログボックスが開き、全データ(物理セグメント1以上のメモリインドウでは同一セグメント内の全データ)を同じ値に変更することができます<sup>39</sup>。この機能は、データメモリや物理セグメント1以上のメモリのデータ領域を初期化するときに便利です。



### ■ ソースウィンドウと逆アセンブルウィンドウからのメモリ変更

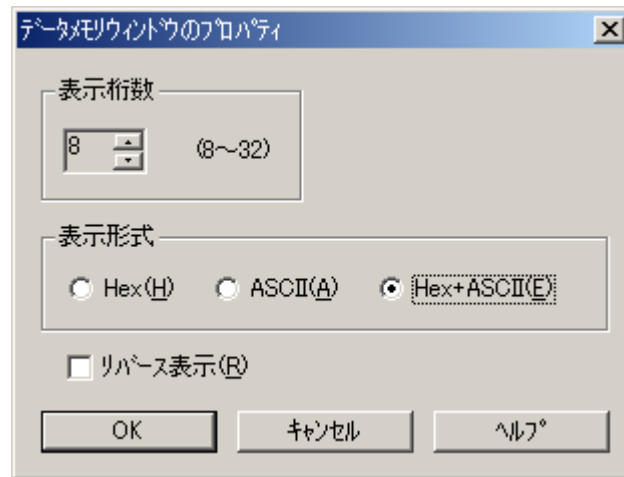
ソースウィンドウ、または、逆アセンブルウィンドウで、変更したい領域を示す変数やラベルを選択して、ポップアップメニューの[値の変更]を指定すると、選択した変数やラベルのアドレス空間をチェックして、[コードメモリの変更]、[データメモリの変更]、[物理セグメント1以上のメモリの変更]ダイアログボックスが開きます。また、開始アドレスのデフォルト値として、選択した変数やラベルが登録されていますので、そのまま、変更値のリストを指定することができます。

### ■ メモリウィンドウに表示する1行あたりのデータ数の変更

コードメモリウィンドウ、データメモリウィンドウおよび物理セグメント#1以上のメモリウィンドウに表示する1行あたりのデータ数を変更することができます。

コードメモリウィンドウ、データメモリウィンドウ、物理セグメント#1以上のメモリウィンドウのポップアップメニューから[プロパティ]を選択すると、次のようなダイアログを表示します。

<sup>39</sup> uEASE および nanoEASE では、データメモリウィンドウのみ、本メニューが選択可能です。コードメモリ、物理セグメント1以上のメモリに対して同等の操作を行うには、ポップアップメニューの[コードメモリの変更]または、[物理セグメント1以上のメモリの変更]を選択し、対象のメモリの開始アドレスと終了アドレスを指定してデータの変更を行ってください。



#### ■[表示桁数]入力フィールド

1 行に表示するデータ数を指定します。8～32 の範囲の偶数を指定できます。

例えば、データメモリウィンドウの表示桁数を 32 に設定した場合、次のように 1 行に 32 アドレス分を表示します。

メモ帳																																
Adrs	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+1A	+1B	+1C	+1D	+1E	+1F
0000	C8	8D	6A	00	6C	00	F8	8F	34	88	00	FF	58	88	00	FF	62	00	66	1D	00	FF	90	1D	00	FF	02	FF	CC	ED	01	80
0020	CC	ED	FF	80	48	FF	69	35	FF	FF	46	64	53	FE	E8	F5	CE	51	45	82	EF	49	37	97	18	6E	12	FF	67	45	7A	EA
0040	DD	9E	EA	AB	AE	FF	16	32	75	14	68	CB	91	7A	7D	85	19	45	3C	9E	FE	36	CF	7D	A8	42	55	00	00	00	00	
0060	00	00	00	00	0D	00	2C	1D	00	FF	01	CE	FF	CE	00	E0	00	E2	00	E4	00	E6	00	08	80	09	8A	F0	57	90	88	E8
0080	90	79	FC	A1	00	78	FA	C1	00	0A	0C	F0	34	1D	AF	90	52	90	FF	70	02	C8	FF	71	20	C9	AF	90	52	92	AF	90
00A0	52	94	AF	90	50	96	AF	90	50	97	80	E4	F0	C9	01	A2	0B	C8	01	A4	09	C8	6F	90	02	98	82	E0	7F	90	23	98
00C0	82	E2	FE	E4	F8	C8	E3	CE	6F	90	00	98	81	E0	7F	90	21	98	81	E2	FF	E4	F8	C8	DA	CE	01	F0	34	0D	00	E3
00E0	30	90	00	F0	3C	1C	5E	FE	1A	AE	FE	E1	6E	F8	5E	F4	05	F8	25	FA	00	E0	FE	B0	0C	CE	42	B0	7E	B2	26	F0
0100	05	F2	85	F0	7E	B4	46	F0	20	92	01	92	7E	B0	81	E0	FE	B0	7E	B0	A7	F0	F1	C5	1E	F4	2E	F8	EA	A1	1E	FE
0120	1F	FE	5E	FE	1A	AE	FE	E1	6E	F8	5E	F4	05	F8	20	8A	00	E0	FE	B0	0C	CE	7E	B0	06	F0	05	F2	44	B4	46	F2
0140	86	F0	22	92	AF	90	03	92	7E	B0	81	E0	FE	B0	7E	B0	42	B2	27	F0	F0	C5	1E	F4	2E	F8	EA	A1	1E	FE	1F	FE
0160	5E	FE	1A	AE	FE	E1	6E	F8	5E	F4	05	F8	25	FA	00	E0	FE	B0	0B	CE	7E	B0	06	F0	05	F2	42	B4	46	F2	86	F0
0180	22	92	03	92	7E	B0	81	E0	FE	B0	7E	B0	A7	F0	F2	C5	1E	F4	2E	F8	EA	A1	1E	FE	1F	FE	5E	FE	1A	AE	FE	E1
01A0	6E	F8	6E	F4	05	F8	20	8A	00	E0	FE	B0	13	CE	7E	B0	2B	91	2A	90	05	F2	44	B4	46	F2	86	F0	05	F6	22	90
01C0	28	A2	02	00	AF	90	63	90	9F	FE	69	A2	02	00	7E	B0	81	E0	FE	B0	7E	B0	42	B2	27	F0	E9	C5	2E	F4	2E	F8
01E0	EA	A1	1E	FE	1F	FE	5E	FE	1A	AE	FE	E1	6E	F8	5E	F4	05	F8	25	FA	00	E0	FE	B0	0C	CE	42	B0	7E	B2	26	F0

## 8.5 コード領域の保存と比較

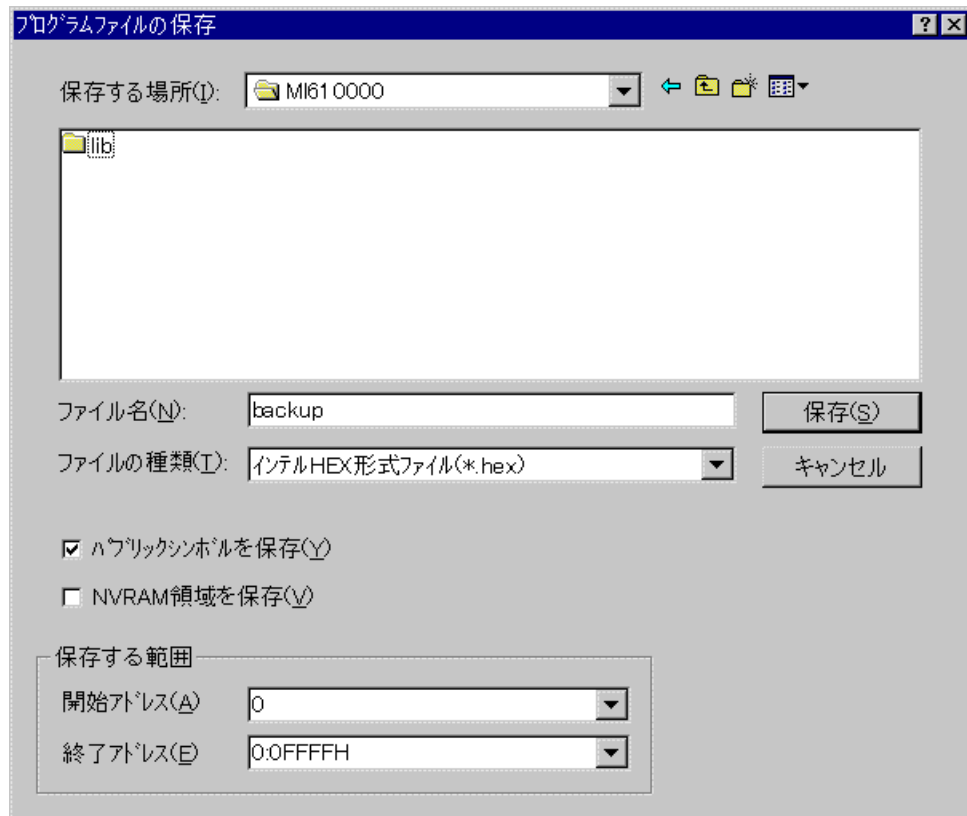
### 8.5.1 コード領域の保存

コード領域の ROM テーブルのパラメータを調整したときや、パッチプログラムを追加したときなど、現在のプログラムコードを保存しておきたい場合があります。DTU8 デバッガは、変更したプログラムコードをファイルに保存することができます。

プログラムコードをファイルに保存するには、[ファイル]メニューの[プログラムファイルの保存]を選択します。<sup>40</sup>

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

<sup>40</sup> [ツール]メニューの[環境設定]で表示されるダイアログボックスの[ディレクトリ]タブの[プログラムファイル]フィールドに、ファイルの格納先となるディレクトリを指定しておけば、毎回ディレクトリを選択する必要がなくなります。



#### ■ プログラムコードとシンボル情報を保存する場合

プログラムコードとシンボル情報を保存する場合は、[パブリックシンボルを保存]チェックボックスをチェックします。

#### ■ プログラムコードだけを保存する場合

プログラムコードだけを保存する場合は、[パブリックシンボルを保存]チェックボックスのチェックを外します。

#### ■ プログラムコードの一部を保存する場合

あるアドレス範囲のみを保存する場合は、[開始アドレス]と[終了アドレス]入力フィールドに保存する範囲を指定します。

#### ■ NVRAM 領域を保存する場合

[NVRAM 領域を保存]チェックボックスをチェックすると、データメモリ (物理セグメント 0) に NVRAM 領域がある場合はその領域だけを別ファイルに保存します。

インテル HEX ファイルで保存する場合は XNV、モトローラ S2 ファイルで保存する場合は SNV にそれぞれ拡張子を変えて保存されます。

#### ■ アクセスできない領域の扱い

保存の対象の中に N/A 領域等のアクセスができない領域を含む場合、その領域をスキップして、保存処理が行われます。

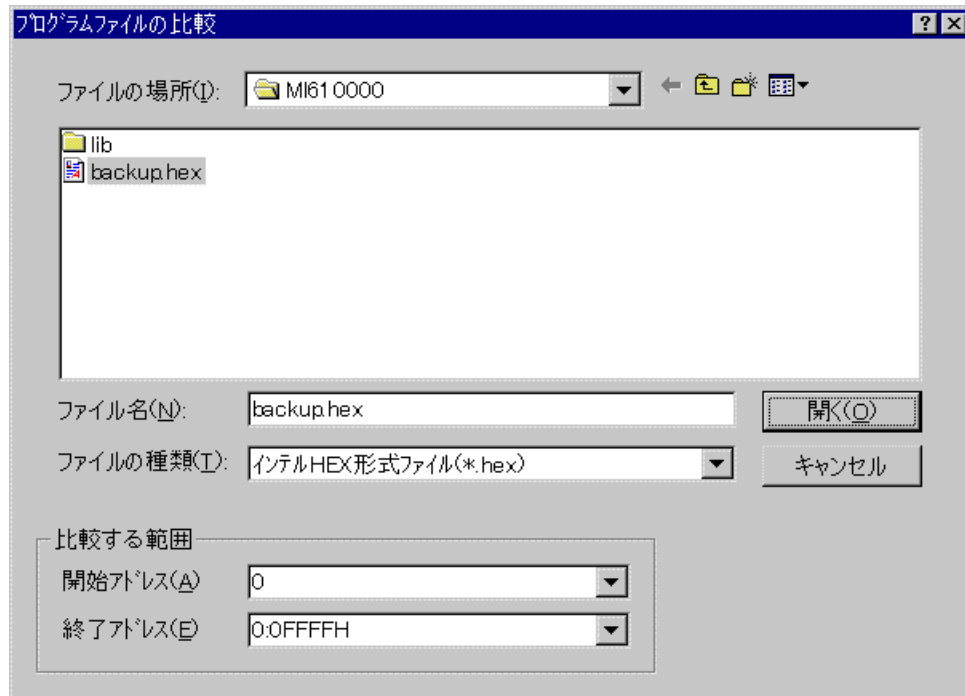


### 8.5.2 エミュレータのコード情報と HEX ファイルの比較

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

エミュレータ内のコード領域の内容を HEX ファイルと比較することにより、ロードしたプログラムの確認と、変更した領域の確認ができます。

[ファイル]メニューの[プログラムファイルの比較]を指定すると、以下のダイアログボックスが開きます。



#### ■ プログラムファイルの全領域を比較する場合

ロードしたプログラムコードとプログラムファイルの全領域を比較する場合は、[開始アドレス]と[終了アドレス]入力フィールドに何も設定せずに、ファイルを選択します。

#### ■ プログラムファイルの一部を比較する場合

ロードしたプログラムコードとプログラムファイルの一部を比較する場合は、[開始アドレス]と[終了アドレス]入力フィールドに比較したいアドレスを指定し、ファイルを選択します。

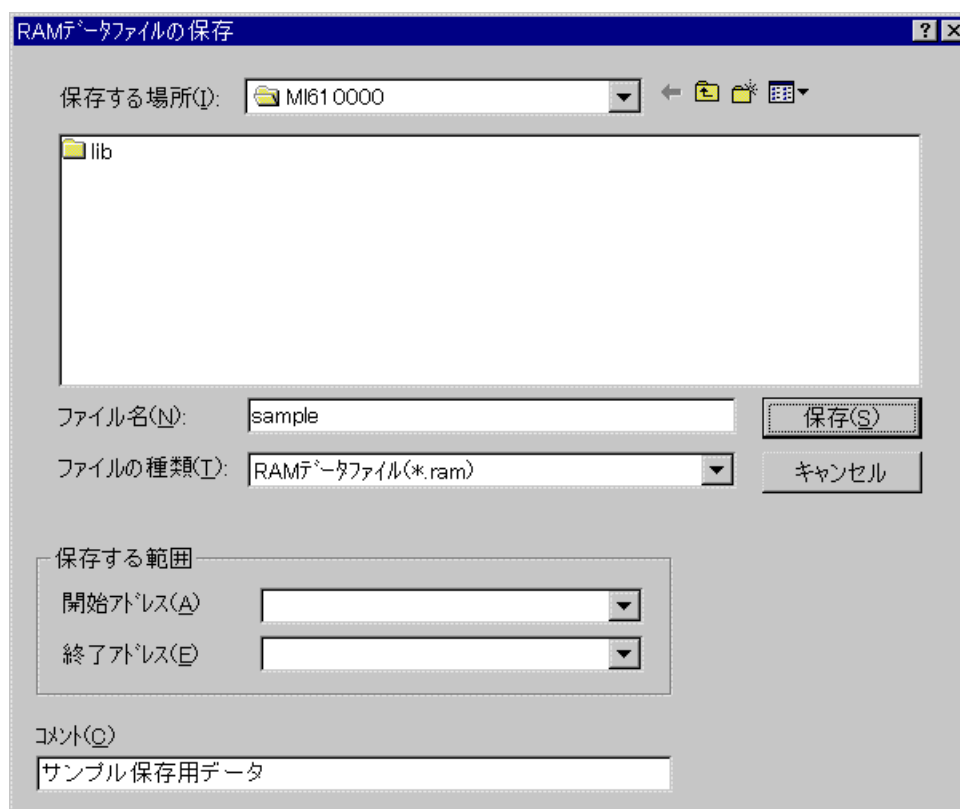
## 8.6 データ領域の保存と復帰

### 8.6.1 データ領域の保存

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ある特定の状況下において、データメモリの状態を保存しておきたい場合があります。DTU8 デバッガは、現在のデータメモリの内容をファイルに保存することができます。

データメモリをファイルに保存するには、[ファイル]メニューの[RAMデータファイルの保存]を選択します。<sup>41</sup>



#### ■ データメモリの一部を保存する場合

あるアドレス範囲のみを保存する場合は、[開始アドレス]と[終了アドレス]入力フィールドに保存する範囲を指定します。

#### ■ アクセスできない領域の扱い

保存の対象の中に N/A 領域等のアクセスができない領域を含む場合、その領域をスキップして、保存処理が行われます。

#### ■ SFR 領域の扱い

SFR 領域は、アクセスできない領域同様に、保存時にスキップされます。

<sup>41</sup> [ツール]メニューの[環境設定]を選択し、表示される環境設定ダイアログのディレクトリタブの[RAM データファイル]フィールドに、ファイルの格納先となるディレクトリを指定しておけば、毎回ディレクトリを選択する必要がなくなります。

### ■ コメントの挿入

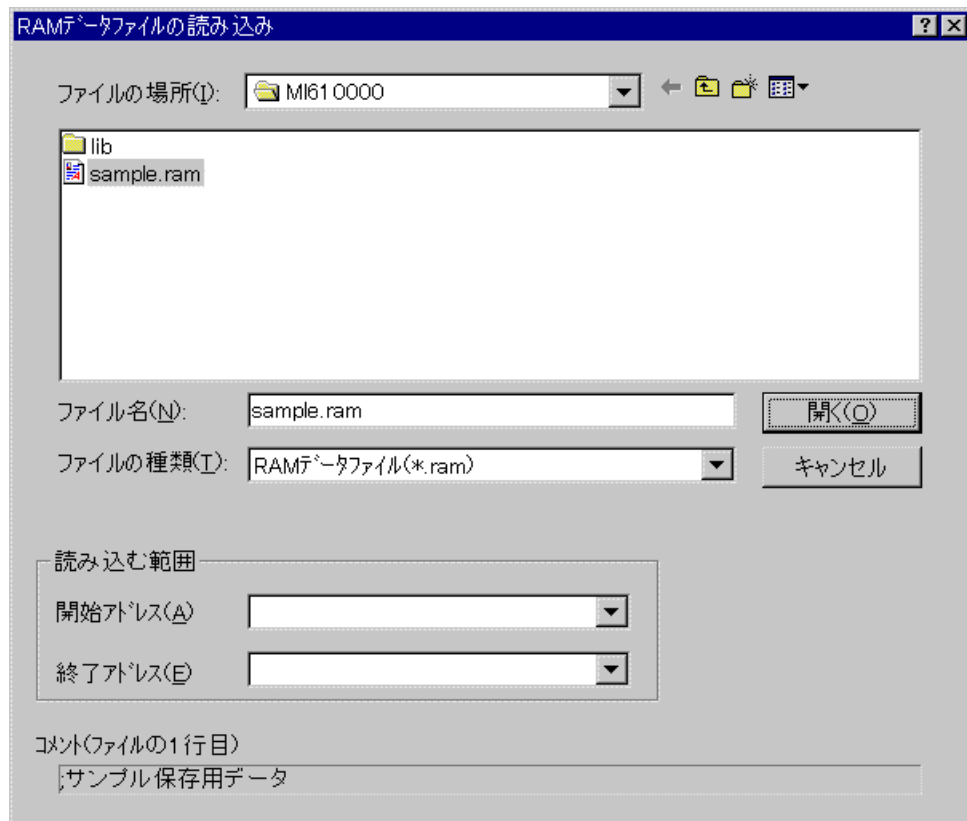
RAM データファイルの保存ダイアログのコメントフィールドに入力した文字列は、保存対象ファイルに、コメントとして挿入されます。このコメントは、[ファイル]メニューの[RAM データファイルの読み込み]を選択して表示される RAM データファイルの読み込みダイアログで参照できます。

### 8.6.2 データ領域の復帰

DTU8 デバッガは、[ファイル]メニューの[RAM データファイルの保存]によって保存されたデータを、再度データメモリへ復帰することができます。

データメモリファイルを復帰するには、[ファイル]メニューの[RAM データファイルの読み込み]を選択します。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate



### ■ データメモリの一部を復帰する場合

あるアドレス範囲のみを復帰する場合は、[開始アドレス]と[終了アドレス]入力フィールドに復帰する範囲を指定します。

### ■ コメントの表示

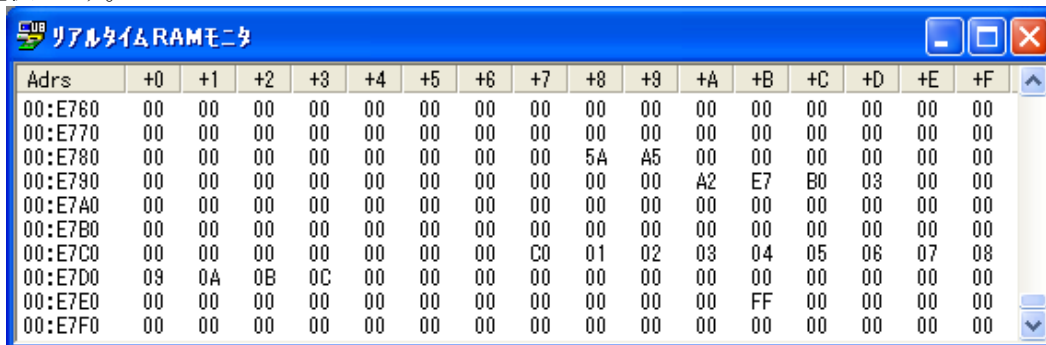
RAM データファイルに格納されているコメントが、コメントフィールドに表示されます。

## 8.7 リアルタイム RAM モニタ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

リアルタイム RAM モニタウィンドウでは1 セグメント内の RAM のうち最大 16K バイトの領域の内容をエミュレーション中にリアルタイムに確認することができます。

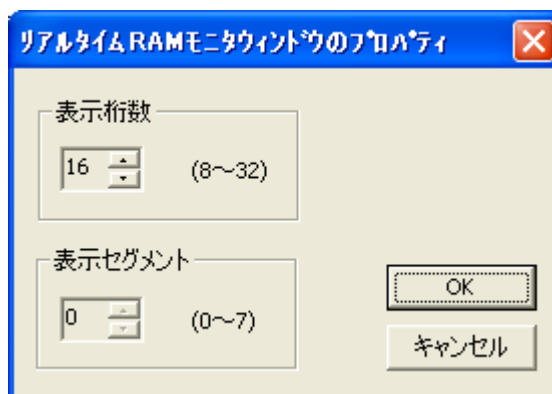
リアルタイム RAM モニタウィンドウを表示するには、[表示]メニューの[リアルタイム RAM モニタ]を選択します。



Adrs	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00:E760	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00:E770	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00:E780	00	00	00	00	00	00	00	00	5A	A5	00	00	00	00	00	00
00:E790	00	00	00	00	00	00	00	00	00	00	A2	E7	B0	03	00	00
00:E7A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00:E7B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00:E7C0	00	00	00	00	00	00	00	C0	01	02	03	04	05	06	07	08
00:E7D0	09	0A	0B	0C	00	00	00	00	00	00	00	00	00	00	00	00
00:E7E0	00	00	00	00	00	00	00	00	00	00	FF	00	00	00	00	00
00:E7F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

#### ■ リアルタイム RAM ウィンドウに表示するセグメントと1行あたりのデータ数の変更

リアルタイム RAM ウィンドウに表示するセグメントと1行あたりのデータ数を変更することができます。ポップアップメニューから[プロパティ]を選択すると、次のようなダイアログを表示します。



- [表示桁数]フィールド

1 行に表示するデータ数を指定します。8～32 の範囲の偶数を指定できます。

- [表示セグメント]フィールド

リアルタイム RAM モニタウィンドウに表示する RAM のあるセグメントを指定します。RAM の存在するセグメントのみが選択でき、RAM の存在しないセグメントを選択した場合はエラーメッセージを表示します。

## 9 章 トレース機能

トレース機能は、エミュレーション機能でプログラムを実行中に、その実行経過をトレースメモリと呼ばれるメモリに保存する機能です。

トレース機能は、エミュレーションシステムに依存する機能です。エミュレーションシステムによってはサポートされない場合があります。この場合、メニューには表示されません。

トレースメモリは、256K ステップの容量を持ち、格納されているデータが 256K ステップを越えると、トレースメモリの古いデータから順にオーバーライトされていくため、ブレーク時には、最大、256K ステップまでさかのぼって、実行履歴を確認できます。

Dr.U8 ICE モード、Dr.U16 ICE モード、Dr.ICE モードの場合、トレースはインストラクション単位です。1 ステップに 1 命令分のトレース内容を表示します。ただし、RAM をアクセスする命令の場合はアクセスするデータのバイト数分を表示します。シミュレーションモードの場合、トレースはサイクル単位です。1 ステップに 1 サイクル分のトレース内容を表示します。

### 9.1 トレースウィンドウ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースウィンドウは、エミュレータのトレースメモリの内容を表示します。

トレースウィンドウを開く場合は、[表示]メニューの[トレース]を選択します。

プログラムを実行していない場合などトレースメモリに何もデータがない状態では、空のウィンドウが開きます。

トレースウィンドウでは、最上行に表示されているデータが最も古くトレースされたものであり、最下行が最も新しくトレースされたデータを表しています。トレースウィンドウの上部(タイトルバーの下)に表示されているラベルは、各データの名称を表しています。

トレースウィンドウは、直前の状態から変化したデータを赤色で表示します。

Loc	PC	PSW(CZSOMHE)	RAMAdr	Data	Probe	Source
000042	0:00A6	00 000000	00:E7FF	00	00	add er2, #2
000043	0:00A8	20 001000	00:E7FF	00	00	add er4, #-2
000044	0:00AA	C4 1100010	00:E7FF	00	00	bnz init loop2
000045	0:00AC	C4 1100010	00:E7FF	00	00	bal init loop
000046	0:0074	C4 1100010	00:E7FF	00	00	l er0, r10:[ea+]
000047	0:0074	C4 1100010	00:0016	FF	00	
000048	0:0078	A4 1010010	00:0017	FF	00	
000049	0:0078	A4 1010010	00:0017	FF	00	cmp r0, #0ffh
000050	0:007A	40 0100000	00:0017	FF	00	bne skip
000051	0:007C	40 0100000	00:0017	FF	00	cmp r1, #0ffh
000052	0:007E	40 0100000	00:0017	FF	00	beq init end
000053	0:00C0	40 0100000	00:0017	FF	00	bl \$\$content of init
000054	0:0006	40 0100000	00:0017	FF	00	rt
000055	0:00C4	40 0100000	00:0017	FF	00	mov r0, #0
000056	0:00C6	40 0100000	00:0017	FF	00	st r0, DSR
000057	0:00C6	40 0100000	00:F000	00	00	

### ■ LOC

トレースメモリの位置を表す情報です。値が小さいほど古いデータです。

### ■ PC

トレースされたときのプログラムカウンタ値です。

### ■ PSW (CZSOMHE)

PSW のバイト値と、各フラグおよびフィールドを表示します。

### ■ RAM Adr

データメモリを書き換える命令が実行されたときのデータメモリアドレスです。

### ■ Data

データメモリを書き換える命令が実行されたときの書き込みデータです。

nX-U8 コアの機種の場合、書き込みデータは常にバイト値で表示します。nX-U16 コアの機種の場合、書き込みデータが 2 バイトのときはワード値で、書き込みデータが 1 バイトのときはバイト値 (左詰め) で表示します。

### ■ Probe

トレースプローブケーブル PROBE0～PROBE7 のデータをバイト値で表示します。なお、シミュレーションモードでは、\*\*が表示されます。

### ■ Source / Disassembly

トレースされた PC に対応するソース行または実行命令を表示します。

この表示は、トレースウインドウのポップアップメニューの[プロパティ]で、ソース表示または逆アセンブル表示に切り替えることができます。

## 9.2 表示モード

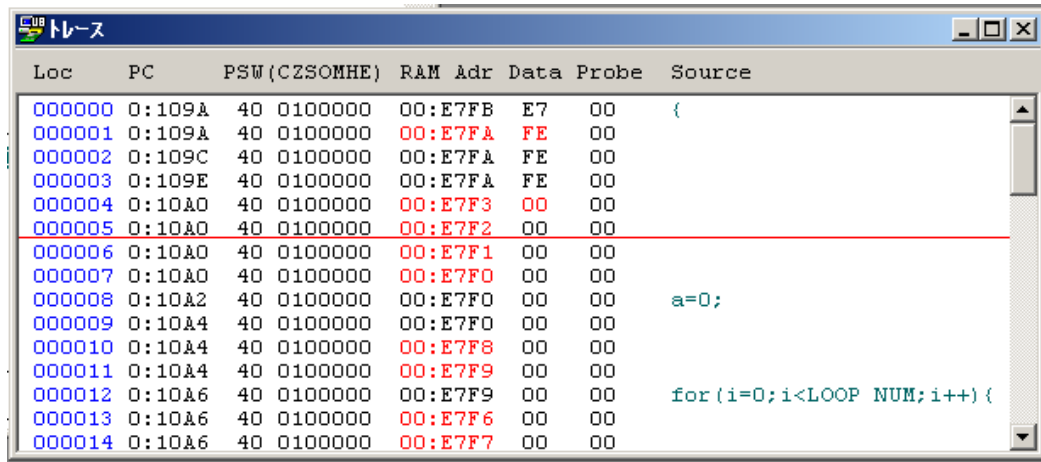
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースウインドウの[Source / Disassembly]フィールドの表示は、ポップアップメニューでトレースウインドウのプロパティで、切り替えることができます。



### ■ ソース表示モード

トレースされた PC が、ソースデバッグ情報が存在する領域であれば、対応するソース行を表示します。ソースデバッグ情報のない領域は、実行コードの内容を逆アセンブルして表示します。



Loc	PC	PSW (CZSOMHE)	RAM	Adr	Data	Probe	Source
000000	0:109A	40 0100000	00:E7FB	E7	00	00	{
000001	0:109A	40 0100000	00:E7FA	FE	00	00	
000002	0:109C	40 0100000	00:E7FA	FE	00	00	
000003	0:109E	40 0100000	00:E7FA	FE	00	00	
000004	0:10A0	40 0100000	00:E7F3	00	00	00	
000005	0:10A0	40 0100000	00:E7F2	00	00	00	
000006	0:10A0	40 0100000	00:E7F1	00	00	00	
000007	0:10A0	40 0100000	00:E7F0	00	00	00	
000008	0:10A2	40 0100000	00:E7F0	00	00	00	a=0;
000009	0:10A4	40 0100000	00:E7F0	00	00	00	
000010	0:10A4	40 0100000	00:E7F8	00	00	00	
000011	0:10A4	40 0100000	00:E7F9	00	00	00	
000012	0:10A6	40 0100000	00:E7F9	00	00	00	for (i=0; i<LOOP_NUM; i++) {
000013	0:10A6	40 0100000	00:E7F6	00	00	00	
000014	0:10A6	40 0100000	00:E7F7	00	00	00	

C ソースコードは、複数の命令コードからなりますが、トレースされた PC が C ソースの先頭アドレスのときに、トレースウィンドウに表示します。

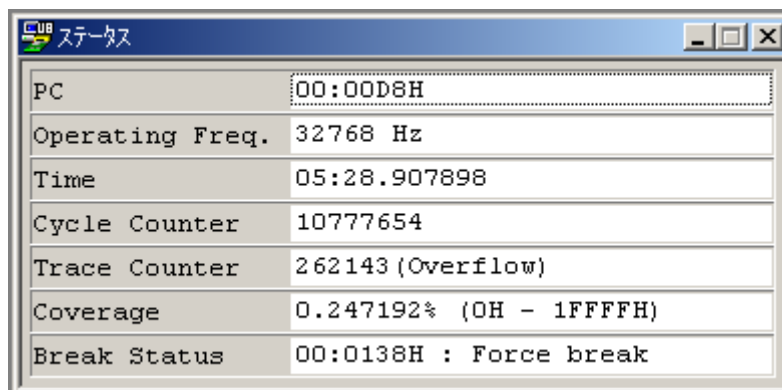
### ■ 逆アセンブル表示モード

トレースされた PC の実行コードを逆アセンブルして表示します。

## 9.3 トレースカウンタの表示

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースメモリに格納されているステップ数を示すトレースカウンタは、ステータスウィンドウの [Trace Counter] フィールドに表示されています<sup>42</sup>。



PC	00:00D8H
Operating Freq.	32768 Hz
Time	05:28.907898
Cycle Counter	10777654
Trace Counter	262143 (Overflow)
Coverage	0.247192% (0H - 1FFFFH)
Break Status	00:0138H : Force break

トレースメモリの内容が 256K ステップを越えた場合には、ステータスウィンドウの [Trace Counter] フィールドに、「(Overflow)」と表示されます。

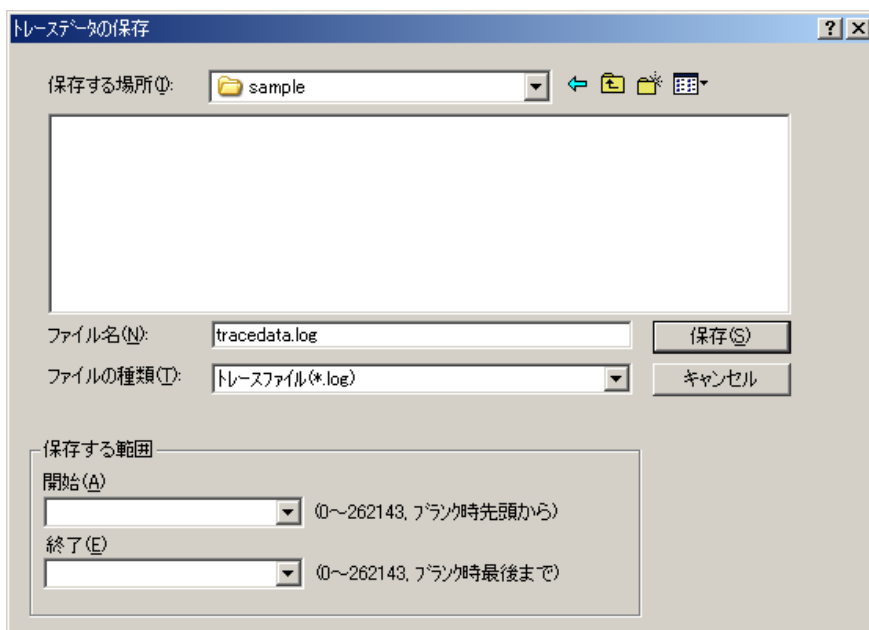
<sup>42</sup> ステータスウィンドウに表示される内容は、接続しているエミュレーションシステムの種類によって異なります。例で示しているステータスウィンドウは、シミュレーションモード時のものです。

## 9.4 トレースデータの保存

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースメモリの内容を、ディスクファイルに保存できます。

トレースメモリの内容をファイルへ保存するには、ポップアップメニューの[トレースデータの保存]を選択します。



[保存する範囲]に何も指定しない場合は、トレースメモリの全領域がファイルに保存されます。  
保存したデータは、以下のようなテキストデータです。

.....

```
002909 0:00B2 40 0100000 00:0739 FF FF BL 0H:6H
002910 0:0006 40 0100000 00:0739 FF FF RT
002911 0:00B6 40 0100000 00:073A 7E FF L R0,0H:[EA]
```

Loc	PC	PSW (CZSOMHE)	RAM	Adr	Data	Probe	Source
002912	0:00B6	40 0100000	00:073A	7E	FF		
002913	0:00BA	00 0000000	00:073A	7E	FF		B_main ;600H
002914	0:0600	00 0000000	00:073A	7E	FF		io_init(); /* I/O の初期化 */
002915	0:037A	00 0000000	00:87CF	00	FF		{
002916	0:037A	00 0000000	00:87CE	00	FF		
002917	0:037A	00 0000000	00:87CD	06	FF		
002918	0:037A	00 0000000	00:87CC	04	FF		
002919	0:037C	00 0000000	00:87CC	04	FF		asm_init_proc();
002920	0:0436	00 0000000	00:87CB	00	FF		PUSH LR ; サブルーチンコールがあるときに必要。
002921	0:0436	00 0000000	00:87CA	00	FF		
002922	0:0436	00 0000000	00:87C9	03	FF		
002923	0:0436	00 0000000	00:87C8	80	FF		

.....



32 ステップ毎に、タイトルを挿入します。

この保存したトレースデータの[Source / Disassembly]フィールドは、ポップアップメニューでトレースウィンドウのプロパティで設定する表示モードの形式で保存します。

なお、トレースデータを保存すると、トレースウィンドウのタイトルに保存ファイルの名前を表示しています。

## 9.5 トレースデータの検索

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

256K ステップ分のトレースデータは、非常に大きなものとなり、その全体を保存するのは、かなりの時間がかかります。このため、エミュレータは、それ自身が持つトレースメモリの内容を検索する機能をサポートしています。

トレースウィンドウのポップアップメニューの[検索]を選択すると、以下のダイアログボックスが開きます。

検索

検索対象(T) PC

検索値(V) \_main

マスク(K) 0Fh:0FFFFh

回数(U) 1 (1~262143)

検索方向

☒ 新しい→古い(N)

☐ 古い→新しい(O)

OK

キャンセル

ヘルプ°

### ■ 検索対象フィールド

検索対象を選択します。

- PC : プログラムカウンタ
- PSW : プログラムステータスワード
- C : キャリフラグ
- Z : ゼロフラグ
- S : サインフラグ
- OV : オーバーフローフラグ
- MIE : マスタインタラプトイネーブルフラグ
- HC : ハーフキャリフラグ
- ELEVEL : 割り込みレベル
- RAM Address : リード／ライトされた RAM アドレス
- RAM Data : リード／ライトされた RAM データ
- Probe Data : プローブデータ

### ■ 検索方向ラジオボタン

- 新しい → 古い : 最新のトレースデータから古い方へ検索します
- 古い → 新しい : 最古のトレースデータから新しい方へ検索します

### ■ 検索値フィールド

検索条件として、検索対象の値を指定します。検索対象が PC の場合は、最下位 1 ビットは 0 でマスクされます。

### ■ マスクフィールド

検索対象の値とのマスク値です。マスク値を指定すると、検索対象の値とのビット単位の論理積 (AND) の結果が、検索値と同じになると、マッチします。

指定しない場合には、マスクを行いません。

### ■ 回数フィールド

検索値との一致回数です。1 から現在のトレースカウンタ値までの値が指定可能です。

Dr.ICE モードでは、検索値との比較は検索対象の値が変化したときのみ行われます。従って、同一データが連続している場合は、その先頭のみが比較の対象となります。

## 9.6 トレースメモリのクリア

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースウィンドウの[トレースメモリのクリア]を指定します。

また、[ツール]メニューの[動作設定]で開くダイアログボックスの[リセット]タブで、リセット対象の選択で[トレースメモリ]をチェックすると、リセット時に、トレースメモリをクリアします。

## 9.7 トレーストリガ

Dr.U8 ICE  
Dr.U16 ICE

トレーストリガを指定することにより、トレースの開始・終了を制御できます。

[実行]メニューの[トレーススタートトリガの設定]を選択すると以下のダイアログボックスが開きます。

### ■ トレーススタートトリガ

- フリーラン : エミュレーション開始と同時にトレースを開始します。このチェックボックスをチェックすると、“PC マッチ”および“RAM マッチ”はチェックできません。
- PC マッチ : [アドレス]に設定したアドレスが[パスカウント設定値]に指定した回数だけ実行されるとトレースを開始します。カウント値を最初に設定する場合は、[設定値]に回数を入力後[セット>>>]ボタンを押してください。また、[カウント値]の表示は、エミュレータに設定されている回数値を示します。トレースを開始する前にブレイクした場合には、更新されていることがあります。[セット>>>]ボタンで、再設定できます。
- RAM マッチ: プログラム実行中にアクセスする RAM アドレスを条件としてトレースを開始します。[RAM アドレス]フィールドには、チェックする RAM アドレスを指定します。[RAM アドレスマスク]を指定した場合、プログラム実行中にアクセスした RAM アドレスと[RAM アドレスマスク]に指定した値とのビット論理積 (AND) の結果が対象 RAM アドレスとなります。[アクセス方法]フィールドには、プログラム実行中のアクセス方法を指定します。[Read], [Write], [Read/Write]から選択してください。[マッチデータ指定]をチェックすると、指定された RAM アドレスのデータをチェックしてトレースを開始します。チェックするデータ値を[RAM データ]に指

定し、データのサイズを[アクセス単位]に指定します。また、チェックするデータ値には[RAM データマスク]で指定するマスク値を適用できます。[RAM データマスク]を指定した場合、条件に合致した RAM アドレスの内容と[RAM データマスク]に指定した値とのビット論理積(AND)した結果が[RAM データ]の値と比較されます。[成立条件]には、[Equal]または[Not Equal]を指定します。[データマッチパスカウント]をチェックすると、条件の成立回数を指定することができます。カウント値を最初に設定する場合は、[設定値]に回数を入力後[セット>>]ボタンを押してください。また、[カウント値]の表示は、エミュレータに設定されている回数値を示します。トレースを開始する前にブレイクした場合には、更新されていることがあります。[セット>>]ボタンで、再設定できます。

#### ■ トレースカウンタオーバーフロー時にトレースを停止する

このチェックボックスをチェックすると、トレースメモリの内容が 256K ステップを越えた場合にトレースを停止します。トレースメモリがオーバーライトされないため、プログラムの開始時からの実行経過を確認することができます。

## 9.8 その他

### 9.8.1 ソースウィンドウ/逆アセンブルウィンドウへのジャンプ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースウィンドウのポップアップメニューの[ソースウィンドウへジャンプ]を指定すると、カーソル位置のトレース PC を含む行のソースウィンドウへジャンプします。

トレースウィンドウのポップアップメニューの[逆アセンブルウィンドウへジャンプ]を指定すると、カーソル位置のトレース PC を含む行の逆アセンブルウィンドウへジャンプします。

さらに、トレースウィンドウの[PC]フィールドをダブルクリックすると、ソースウィンドウや逆アセンブルウィンドウにジャンプします。

ソース表示モードの場合には、トレース PC が、ソースコードの領域であればソースウィンドウにジャンプし、ソースコードの領域外であれば逆アセンブルウィンドウにジャンプします。

逆アセンブル表示モードの場合には、逆アセンブルウィンドウにジャンプします。

### 9.8.2 指定行へのジャンプ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

トレースウィンドウのポップアップメニューの[ジャンプ]を指定すると、トレースメモリの行番号(LOC フィールド)を指定して、カーソルをジャンプできます。

トレースウィンドウの[LOC]フィールドをダブルクリックしても、ジャンプ先の行番号を指定するためのダイアログボックスを表示させることができます。

## 10 章 ウォッチ機能

### 10.1 使い方

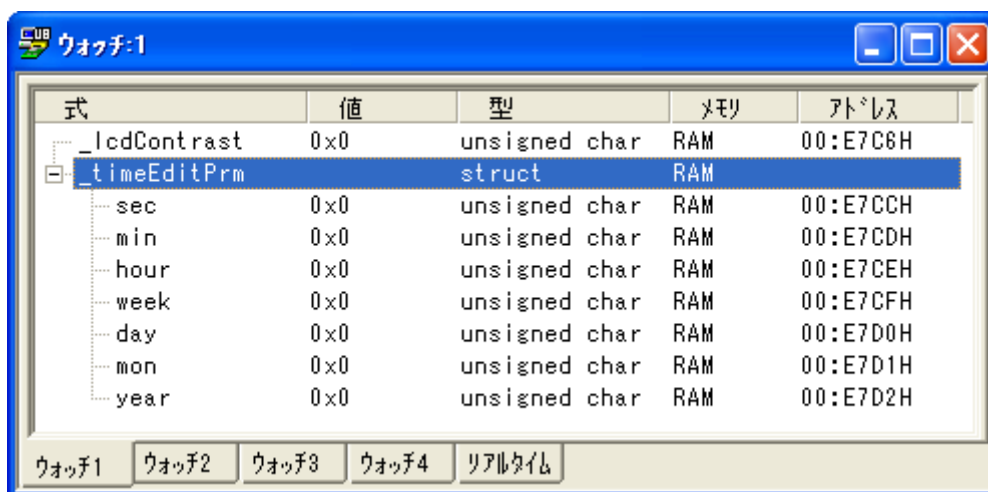
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ウォッチウィンドウには、注目する C 変数をはじめ、データアドレスやレジスタ, SFR 等を登録することができます。

レジスタの間接参照など、複雑なアドレッシングモードもウォッチウィンドウに登録できます。

ラベルシンボルを使ったメモリの内容を参照する場合も、ウォッチウィンドウは有効です。

ウォッチウィンドウを表示するには、[表示]メニューの[ウォッチ]を選択します。



[表示]メニューの[ウォッチ]を選択して新規にウォッチウィンドウを開いたときは、表示項目が何も登録されていないため、空のウォッチウィンドウが開きます。ウォッチウィンドウは、ポップアップメニューの[アイテムの追加]などで項目を登録してからご使用ください。

上の例ではウォッチ 1～ウォッチ 4 とリアルタイムウォッチの計 5 つのタブが表示されていますが、使用するエミュレータの種類によってウォッチタブの表示は変化します。

登録可能な項目に関しては、「付録 12.2 ウォッチウィンドウに登録可能な項目」を参照してください。

## 10.2 登録方法

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### ■ ウォッチウィンドウのポップアップメニューから登録

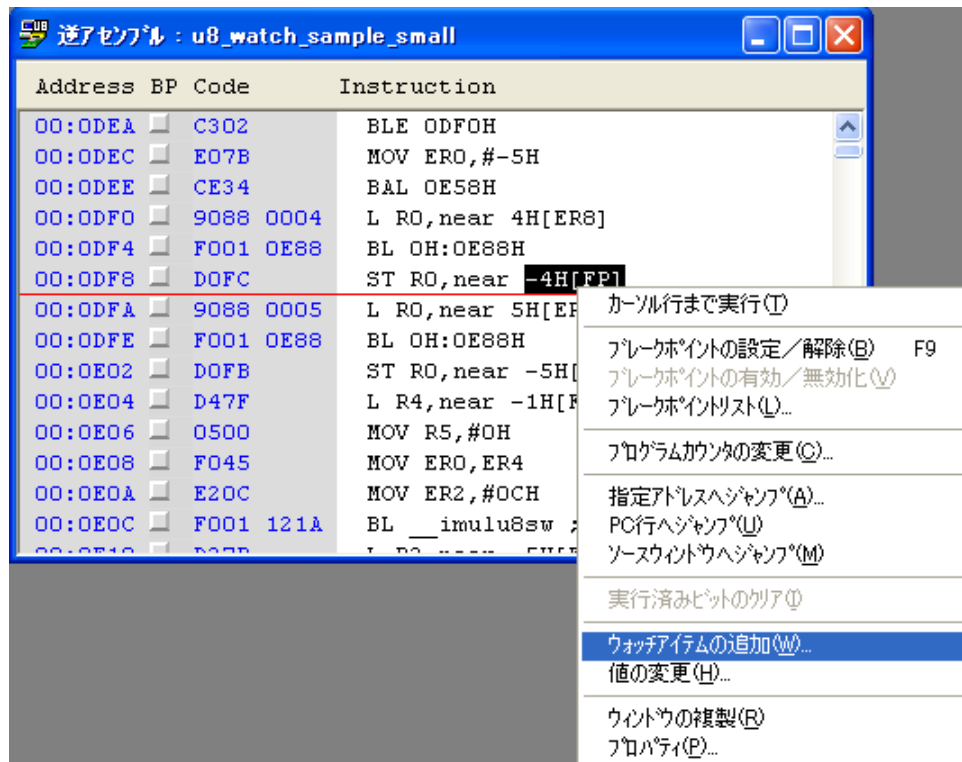
ウォッチウィンドウのポップアップメニューの[アイテムの追加]を選択すると、次のダイアログボックスが表示され、ウォッチ項目を追加することができます。

[種類]	登録するウォッチ項目の種類を指定します。
[アドレス]	登録するウォッチ項目を指定します。
[アドレス属性]	アドレス式の場合、式の指すメモリを指定します。[ビット]はデータメモリを指します。C 変数では設定の必要がなく、無効になります(グレー表示になります)。
[データサイズ]	アドレス式の場合、表示する値のサイズを指定します。C 変数では設定の必要がなく、無効になります(グレー表示になります)。
[表示バイト数]	データアドレス指定時の表示するデータの個数を指定します。C 変数では指定できません(グレー表示になります)。
[基数]	表示するデータの基数を選択します。
[リバース表示]	チェックするとデータの値を左から降順に表示します。C 変数では指定できません(グレー表示になります)。

### ■ ソースウィンドウと逆アセンブルウィンドウ

ソースウィンドウや逆アセンブルウィンドウ上で、登録する文字列をマウスで右クリックしてポップアップメニューの[ウォッチアイテムの追加]を選択するとウォッチアイテムのダイアログボックスが表示され、ウォッチウィンドウに登録することができます。

また、追加したいアイテムをマウスで選択してから、ポップアップメニューの[ウォッチアイテムの追加]を指定することもできます。-4H[FP]のように、1単語以上のものを登録する場合には範囲選択してください。



### ■ SFR ウィンドウとレジスタウィンドウ

SFR ウィンドウやレジスタウィンドウのポップアップメニューの[ウォッチアイテムの追加]を選択すると、カーソル位置の SFR またはレジスタをウォッチウィンドウに登録できます。

### ■ シンボルリスト

[ツール]メニューの[シンボルリスト]を選択すると、シンボルリストダイアログが表示されます。[ウォッチに追加]ボタンを押すと現在選択されているシンボルがウォッチウィンドウへ登録されます。

### ■ アイテム登録の制限事項

ウォッチウィンドウの1つのタブに登録可能なアイテム数は、最大 254 個までです。通常に使用できるタブは4つありますので、一度に登録可能なアイテム数は、 $254 \times 4$  となります。 $254 \times 4$  を超えるアイテムが必要な場合は、ウォッチアイテムの保存、ウォッチアイテムの削除、ウォッチアイテムの読み込みみを利用してください。

## 10.3 ウォッチウィンドウ上でのデータの変更

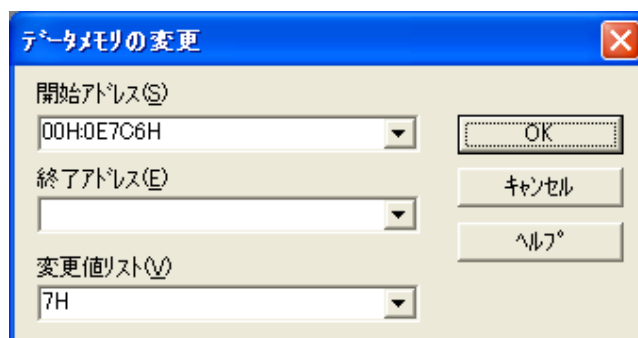
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ウォッチウィンドウに登録されたアイテムの値を変更する機能です。

ウォッチウィンドウに登録されているアイテムをマウスでダブルクリックするか、またはアイテムを選択し、ポップアップメニューから[値の変更]を選択すると、アイテムの値を変更することができます。



ポップアップメニューの[値の変更]を選択すると、次のダイアログボックスを表示します。



この例では、変数”\_lcdContrast”の内容(0E7C6H の RAM の内容)を 7H に変更します。変更値リストには、シンボルも指定可能です。終了アドレスを指定した場合は、開始アドレスから終了アドレスまでの範囲に、変更値リストで指定したデータを書込みます。

上記は RAM の値を変更する場合の例です。値を変更するアイテムの種類により表示するダイアログは、次の表のように異なります。

対象となるアイテム	表示するダイアログ
RAM	[データメモリの変更]ダイアログ
ROM	[コードメモリの変更]ダイアログ
物理セグメント#1 以上のアドレス	[物理セグメント#1 以上のメモリの変更]ダイアログ
SFR	[SFR の変更]ダイアログ
レジスタ	[レジスタの変更]ダイアログ



## 10.4 ウォッチアイテムリストの保存

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

登録したウォッチアイテムのリストは、ファイルに保存できます。

デバッグを開始するときに、毎回同じ項目をウォッチウィンドウに登録するような場合は、この機能を利用すると便利です。

ウォッチウィンドウに表示されているアイテムリストをファイルに保存するには、ポップアップメニューの[アイテムリストの保存]を選択します。1回の操作によって保存されるのは、現在表示しているウォッチタブのアイテムのみです。

現在ウォッチへ登録されているすべてのウォッチアイテムは、DTU8 デバッガのプロジェクトファイルにも保存されます。

### ■ 保存時のデータについて

この機能は、登録されている項目の保存は行いますが、表示されている値は保存されません。

## 10.5 ウォッチアイテムリストの読み込み

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ファイルに保存したウォッチアイテムリストを、ウォッチウィンドウに展開する場合は、ポップアップメニューの[アイテムリストの読み込み]を選択します。<sup>43</sup>

ウォッチアイテムリストの保存機能と本機能を利用することにより、ソースファイル毎のウォッチアイテムや関数毎のウォッチアイテムを個別に用意することができます。

### ■ ウォッチアイテムリストの読み込み時の注意事項

ウォッチウィンドウの1つのタブに登録可能なアイテム数は、最大 254 個です。ウォッチアイテムリストの読み込み時には、1つのタブの登録アイテム数が 254 個を超えないようにしてください。

## 10.6 ウォッチアイテムの表示形式の変更，削除，移動

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### ■ ウォッチアイテムの並べ替え

登録されているウォッチアイテムの位置を移動する場合は、移動するアイテムを選択し、以下の操作を行います。

- 上に移動する場合

ウォッチウィンドウのポップアップメニューの[1 つ上へ]を選択します。または[Shift]+[↑]キーを押します。

- 下に移動する場合

ウォッチウィンドウのポップアップメニューの[1 つ下へ]を選択します。または[Shift]+[↓]キーを押します。

### ■ ウォッチアイテムの基数変更

登録されているウォッチアイテムの基数を変更する場合は、変更するアイテムを選択し、ポップアップメニューの[アイテムのプロパティ]を選択します。

<sup>43</sup> [ツール]メニューの[環境設定]を選択し、表示される環境設定ダイアログのディレクトリタブの[ウォッチアイテムファイル]フィールドに、ファイルの格納先されているディレクトリを指定しておけば、毎回ディレクトリを選択する必要がなくなります。

### ■ ウォッチアイテムの削除

登録されているウォッチアイテムを削除する場合は、ポップアップメニューの[アイテムの削除]を選択します。

また、登録されているすべてのアイテムを削除する場合は、ポップアップメニューの[アイテムをすべて削除]を選択します。

### ■ ウォッチアイテムのコピー

ポップアップメニューの[アイテムを別のタブへコピー]を使用することにより、登録されているアイテムを個別に選択して、別のタブへコピーすることができます。

### ■ ウォッチアイテムの移動

ポップアップメニューの[アイテムを別のタブへ移動]を使用することにより、登録されているアイテムを個別に選択して、別のタブへ移動することができます。

この場合、移動元のウォッチアイテムは削除されます。

## 10.7 リアルタイムウォッチ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

ウォッチウィンドウには、プログラム実行中に値の変化を監視可能な[リアルタイム]タブが装備されています。[リアルタイム]タブに RAM アドレスを登録することによって、プログラムを実行させた状態で登録アドレスに対するデータの変化を監視することができます。

リアルタイムウォッチは、エミュレータによってはサポートされないものがあります。詳しくは使用するエミュレータのマニュアルを参照してください。

[リアルタイム]タブに登録可能なものは以下となります。

- RAM として使用可能なメモリに割り当てられたアドレス
- SFR
- アセンブリ言語ソースレベルの DATA 属性を持つシンボル
- RAMとして使用可能なメモリに割り当てられたC変数<sup>44</sup>

また、[リアルタイム]タブに登録可能な数は、次のとおりです。

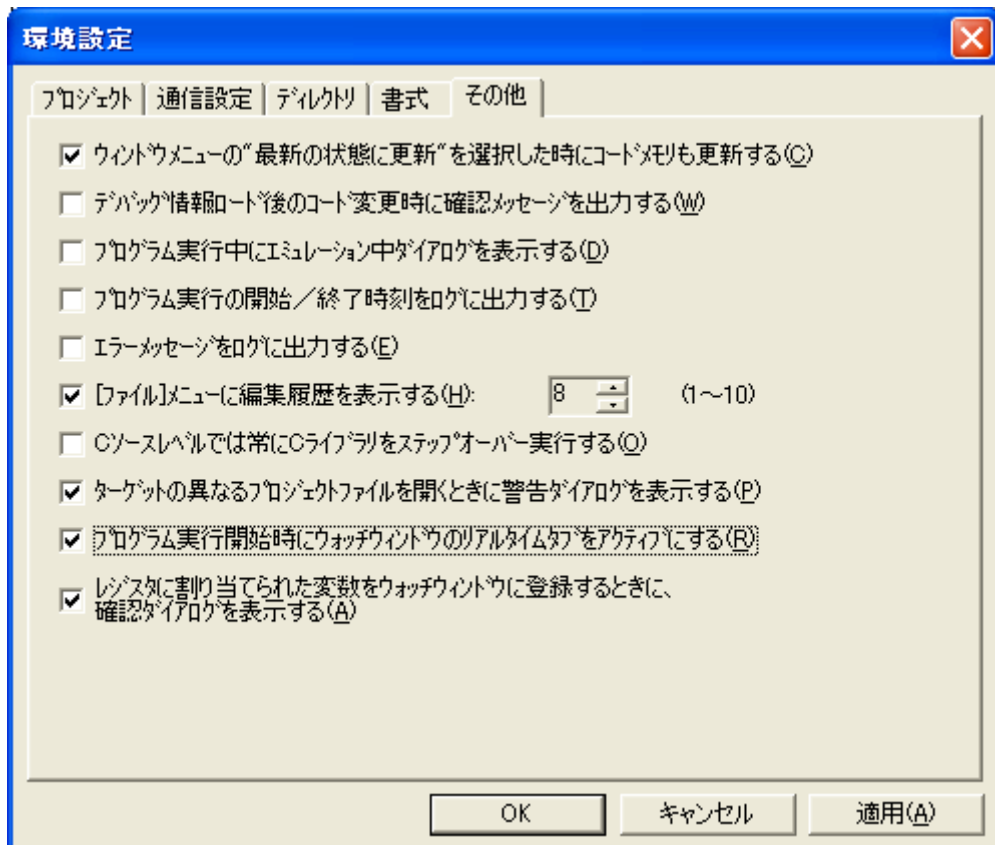
- エミュレーションモードの場合、最大 8 アドレス。各アドレスについて 8/16/32 ビットのサイズを指定可能。
- シミュレーションモードの場合、最大 16 アドレス。各アドレスについて 8/16/32 ビットのサイズを指定可能。

<sup>44</sup> 配列や構造体など子アイテムを持つ変数の場合は、子アイテムは表示されません。

### 10.7.1 リアルタイムウォッチの表示オプション

プログラム実行が開始されたときに、リアルタイムウォッチタブを前面に表示するかどうかを設定することができます。

[ツール]メニューの[環境設定]を選択して表示される[環境設定]ダイアログボックスから[その他]タブを選択すると、次のようなダイアログを表示します。



#### ■ [プログラム実行開始時にウォッチウィンドウのリアルタイムタブをアクティブにする]

このチェックボックスをチェックすると、プログラム実行を開始したときに、ウォッチウィンドウのリアルタイムタブを前面に表示します。ただし、ウォッチウィンドウが閉じている場合や、リアルタイムタブにウォッチアイテムが登録されていない場合は、前面表示は行いません。

初めてDTU8 デバッガを起動した場合、このチェックボックスはチェックされていません。

なお、このチェックボックスの設定情報は、プロジェクトへ保存されます。

# 11 章 その他の機能

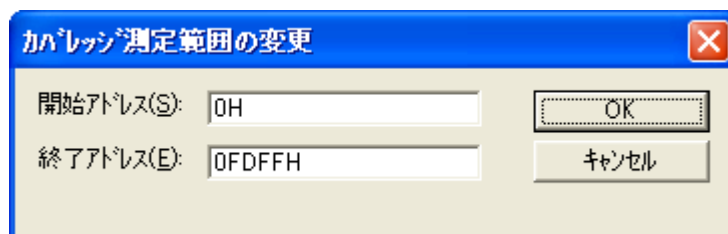
Dr.U16 ICE  
Simulate

## 11.1 カバレッジ機能

DTU8 デバッガのDr.U16ICEモードおよびシミュレーションモードでは、実行済みコードアドレスの情報からプログラムの実行網羅率を算出するカバレッジ機能をサポートしています。<sup>45</sup>

### 11.1.1 使い方

ステータスウィンドウのポップアップメニューから[カバレッジ測定範囲設定]を選択するか、ステータスウィンドウの[Coverage]フィールドをダブルクリックすると、以下のダイアログが開きます。



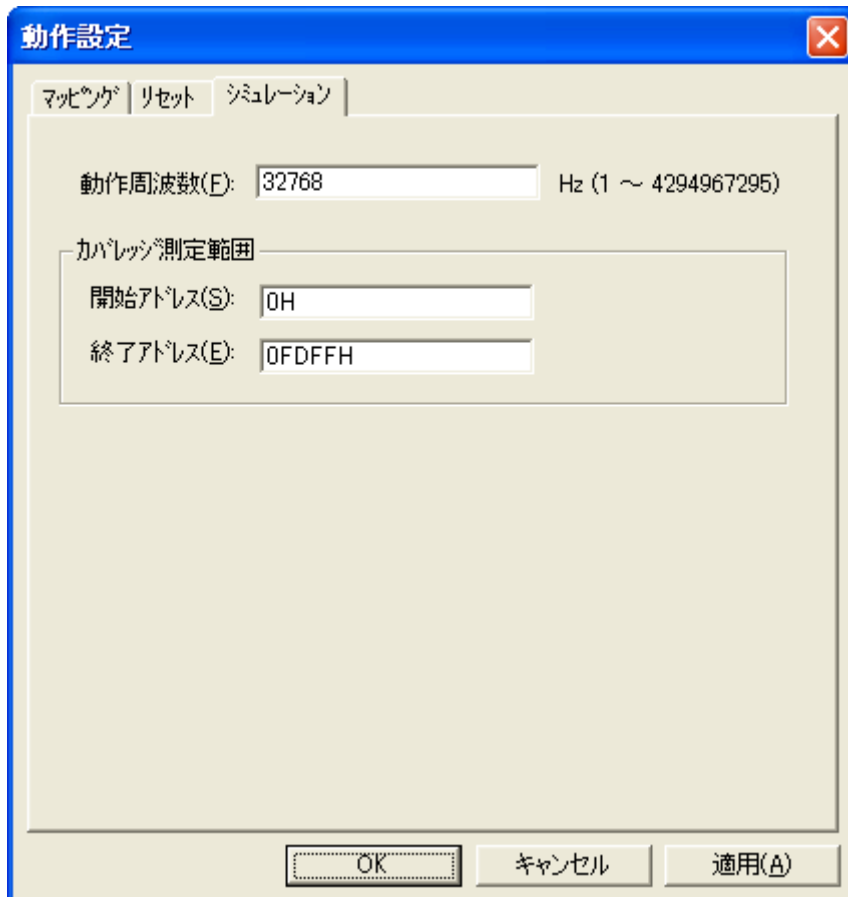
[開始アドレス] カバレッジ測定を開始するアドレスを指定します。

[終了アドレス] カバレッジ測定を終了するアドレスを指定します。

カバレッジ測定範囲に網羅率を計測したいアドレス範囲(通常は、読み込んだプログラムの先頭アドレスから終了アドレスまでとなります)を入力しプログラムを実行すると、シミュレーション実行がブレイクした時点での実行網羅率がステータスウィンドウの[Coverage]フィールドに表示されます。

また、シミュレーションモードの場合は[ツール]メニューの[動作設定]で開くダイアログの[シミュレーション]タブに、以下のようなカバレッジ測定範囲の入力フィールドがありカバレッジ測定範囲の設定が可能です。

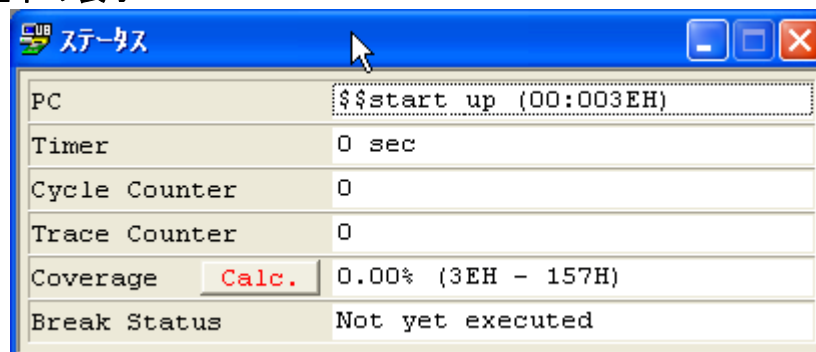
<sup>45</sup> Dr. ICE(旧式を除く)を接続した場合、IE(実行済ビット)を参照することができますが、IEビットを集計して網羅率を表示することはできません。



[開始アドレス]           カバレッジ測定を開始するアドレスを指定します。

[終了アドレス]           カバレッジ測定を終了するアドレスを指定します。

### 11.1.2 網羅率の表示



(Dr.U16ICE モード時のステータスウィンドウ)

実行網羅率は、ステータスウィンドウの[Coverage]フィールドに、次のようなフォーマットで表示されます。

Coverage           *rate*% (*startaddress* - *endaddress*)

*startaddress* と *endaddress* には、カバレッジ測定を行うためのプログラムアドレス範囲を指定します。*rate* は、*startaddress* と *endaddress* 内の全命令数に対する実行した命令の割合をパーセントで

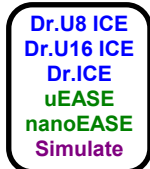
表した値です。

Dr.U16ICE モードでは、Coverage 欄に「Calc.」ボタンが表示されます。「Calc.」ボタンの文字が赤色の場合、表示フィールドが更新されていないことを表しています。実行網羅率を確認したいときに、「Calc.」ボタンを押して表示フィールドを更新してください。

起動直後や実行済みビットがクリアされたとき、エミュレーションを開始したときに、「Calc.」ボタンの文字が赤色になります。

実行網羅率は、ソースウィンドウおよび逆アセンブルウィンドウのIE(実行済み)ビットをもとに算出されています。したがって、実行網羅率を初期化する場合は、ソースウィンドウまたは逆アセンブルウィンドウのポップアップメニューから、[実行済みビットのクリア]を選択してください。

### 11.2 マクロ機能



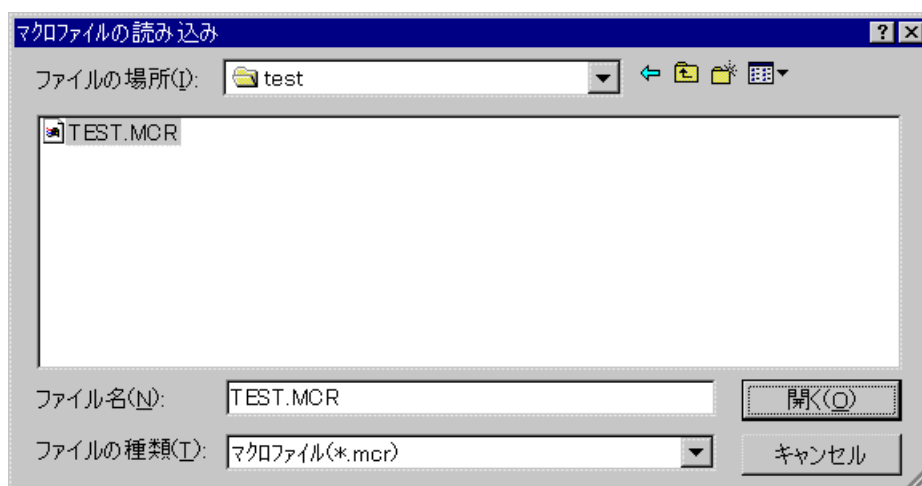
DTU8 デバッガには、スクリプト言語を使用するマクロ機能が搭載されています。外部テキストエディタ等で、テキストファイルにスクリプト言語で作成したマクロファイルを作成し、DTU8 デバッガに指定してマクロ実行させます。

実行した結果は、ログウィンドウへ出力されます。

使用可能なスクリプト言語の一覧は、「付録 12.4 マクロスクリプトリファレンス」を参照してください。

#### 11.2.1 使い方

市販のエディタ等でスクリプト言語を記述したテキストファイルを作成したら、[ツール]メニューの[マクロの実行]を選択します。



ダイアログボックスから、実行するスクリプトファイルを選択すると、マクロの実行が開始されます。実行結果は、ログウィンドウに出力されます。必要に応じて、ログの内容をファイルに保存することも

可能です。<sup>46</sup>

## ■ 例

オブジェクトプログラムをロードし、ブレークポイントを設定し、実行して、ブレークが発生したときのデータアドレスの状態に応じて表示をするというマクロの例です。

```

CLRLOG                ; ログのクリア
ECHO "TEST1 MACRO("$DATE ")  ; タイトルヘッダを出力
LOD TEST.ABS          ; オブジェクトプログラムの読み込み
SBP _sub1              ; sub1 にブレークポイントを設定
ECHO "START AT " $TIME  ; 開始時刻の出力
RST                    ; リセット
G                      ; エミュレーション実行開始
ECHO "STOP AT " $TIME   ; 終了時刻を出力
DBS                    ; ブレークステータスの表示
DDM _AREA1             ; データメモリの表示
DSYM _sub1              ; sub1 のシンボル情報表示
IF PC == 7CH GOTO OK   ; 7CH 番地でブレークしたなら、OK へ
ECHO "NG"              ;
GOTO FIN               ;
:OK
ECHO "GOOD!"           ;
:FIN
SAVELOG TEST1.LOG      ; ログをディスクファイルに保存

```

このスクリプトファイルを実行すると、TEST1.LOG が生成されます。

以下は、TEST1.LOG のサンプル出力です。

```

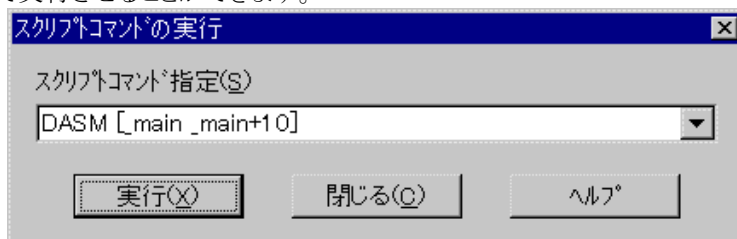
> TEST1 MACRO(2001/07/02)
> LOD TEST.ABS                ; オブジェクトプログラムの読み込み
> SBP _sub1                    ; sub1にブレークポイントを設定
> START AT 15:38:05
> RST                          ; リセット
> G                            ; エミュレーション実行開始
> STOP AT 15:38:06
> DBS                          ; ブレークステータスの表示
Break status
Breakpoint
> DDM _AREA1                   ; データメモリの表示
8BFE : 77H
> DSYM _sub1                   ; sub1のシンボル情報表示
Symbol Name                    Value                                Type
-----
_sub1                          7CH (124)                  CODE
> IF PC == 7CH GOTO OK        ; 7CH番地でブレークしたなら、OKへ
>
> GOOD!
> SAVELOG TEST1.LOG           ; ログをディスクファイルに保存

```

<sup>46</sup> [ツール]メニューの[環境設定]を選択し、表示される環境設定ダイアログのディレクトリタブの[マクロファイル]フィールドに、スクリプトファイルの格納されているディレクトリを指定しておけば、毎回ディレクトリを選択する必要がなくなります。

### 11.2.2 単一スクリプトコマンドの実行機能

[ツール]メニューの[スクリプトコマンド実行]を使用すると、DTU8 デバッガ上で単一のスクリプト言語を指定して実行させることができます。



スクリプトコマンド指定フィールドに、スクリプト言語を入力し、[実行ボタン]を押すと、実行結果がログウィンドウへ出力されます。

逆アセンブルウィンドウや各種メモリのダンプウィンドウに表示されている内容をファイルに保存する場合は、本機能を応用することで可能となります。

例えば、逆アセンブルウィンドウ上の逆アセンブル結果をファイルに保存する場合は、[ツール]メニューの[スクリプトコマンドの実行]を選択し、スクリプトコマンドに”DASM [開始アドレス 終了アドレス]”を指定して[実行]ボタンを押すと、ログウィンドウに結果が表示されます。ログウィンドウのポップアップメニューの[ログの保存]を選択して、逆アセンブル結果をファイルへ保存できます。

## 11.3 ログ機能

ログ機能とは、評価経過を記録するための機能で、DTU8 デバッガのログウィンドウによって実現しています。

ログウィンドウは、簡単なテキストエディタと同程度の機能を持ち、文字列の検索や置換、テキスト編集が可能です。マクロの出力にコメント等を記入することも可能です。

### 11.3.1 ログウィンドウの使い方

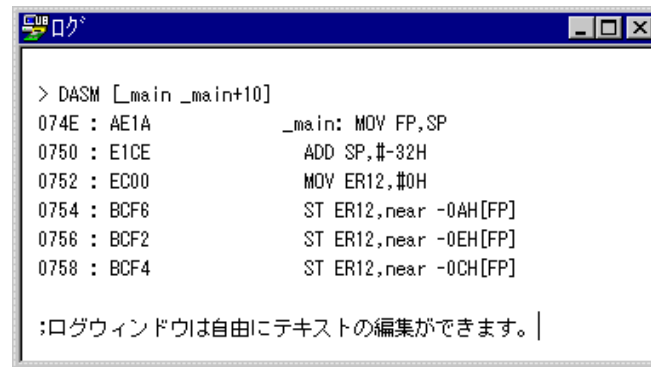
[表示]メニューの[ログ]を選択すると、ログウィンドウが開きます。

ログウィンドウには、次の内容を表示することができます。

- マクロの実行結果
- プログラムファイルの比較結果
- エミュレーションの開始／終了時刻
- エラー／ワーニングメッセージ<sup>47</sup>

<sup>47</sup> エラー／ワーニングメッセージをログウィンドウへ出力するためには、[ツール]メニューの[環境設定]で表示されるダイアログボックスのその他タブで、[メッセージをログに出力する]チェックボックスをチェックする必要があります。



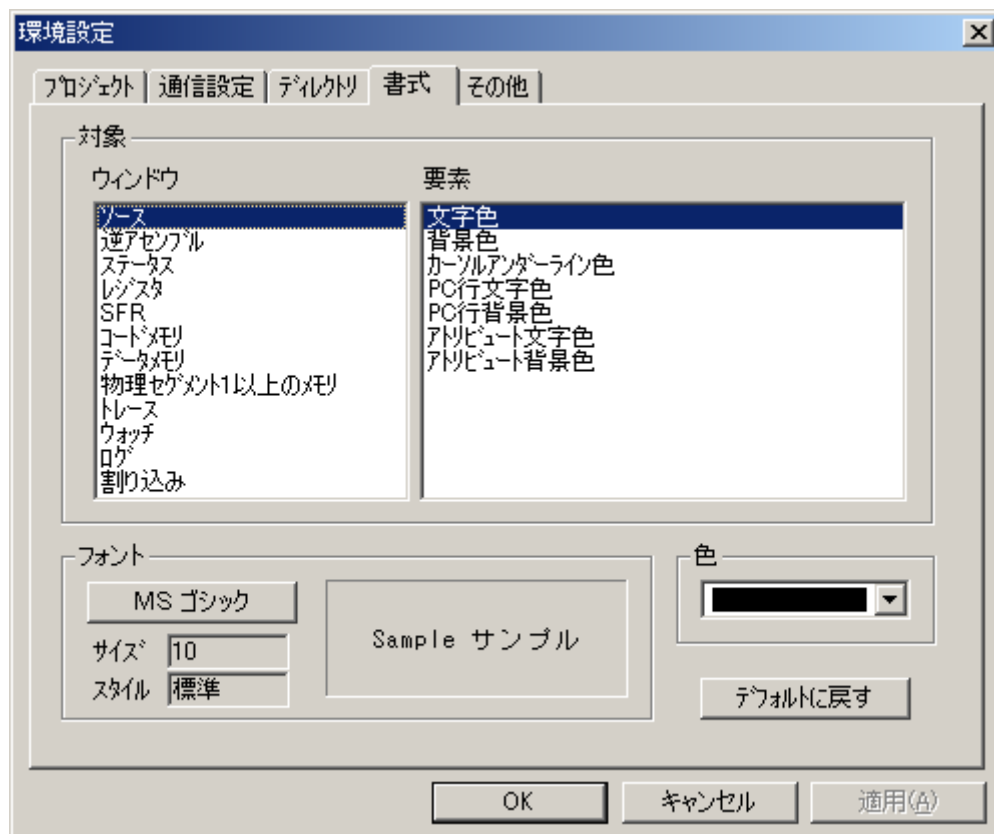


## 11.4 各ウィンドウのカスタマイズ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

各ウィンドウの表示フォントや色を変更することができます。

[ツール]メニューの[環境設定]を選択して表示される[環境設定]ダイアログボックスから[書式]タブを選択すると、次のようなダイアログを表示します。

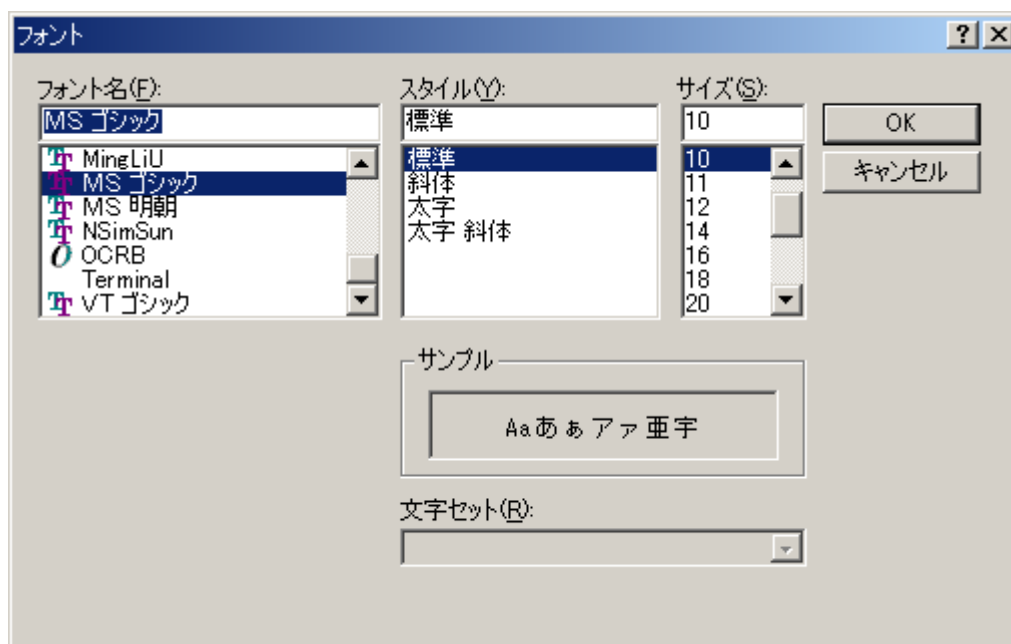


[書式]タブの[フォント]フィールドには、現在[対象]フィールドで選択されているウィンドウのフォント名とサイズを表示します。フォント名が表示されたボタン(以後「フォント名ボタン」と呼びます)をクリックすると、フォントを変更することができます。

### ■ フォントの変更

フォント名ボタンには、現在選択されているウィンドウのフォント名が表示されています。このボタ

ンをクリックすると、次のようなフォント設定ダイアログボックスを表示します。



[フォント名]で変更するフォントを選択し、[スタイル]と[サイズ]にフォントのスタイルとフォントのサイズを指定してください。

DTU8 デバッガで設定可能なフォントは、等幅フォントのみです。MS P Gothicなどのプロポーショナルフォントや、縦書きフォントはフォント名リストに表示しません。

### ■ フォントが変更可能なウィンドウ

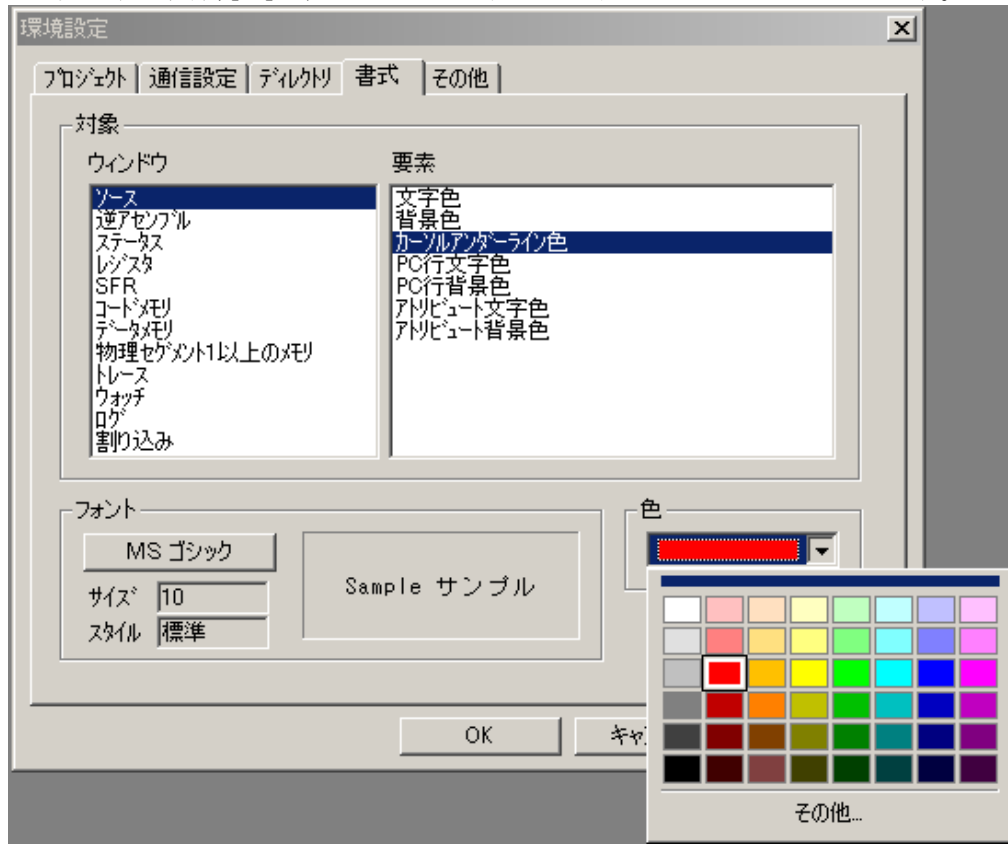
以下のウィンドウは、フォントを変更することができます。

- ・ ソース
- ・ 逆アセンブル
- ・ SFR
- ・ コードメモリ
- ・ データメモリ
- ・ 物理セグメント 1 以上のメモリ
- ・ トレース
- ・ ウォッチ
- ・ ログ

ステータスウィンドウ、レジスタウィンドウ、および割り込みウィンドウは、色の変更のみ可能でフォントは変更できません。

## ■ 色の変更

色を変更する場合、[色]フィールドのプルダウンにより表示されるリストから選択します。



## ■ デフォルトに戻す

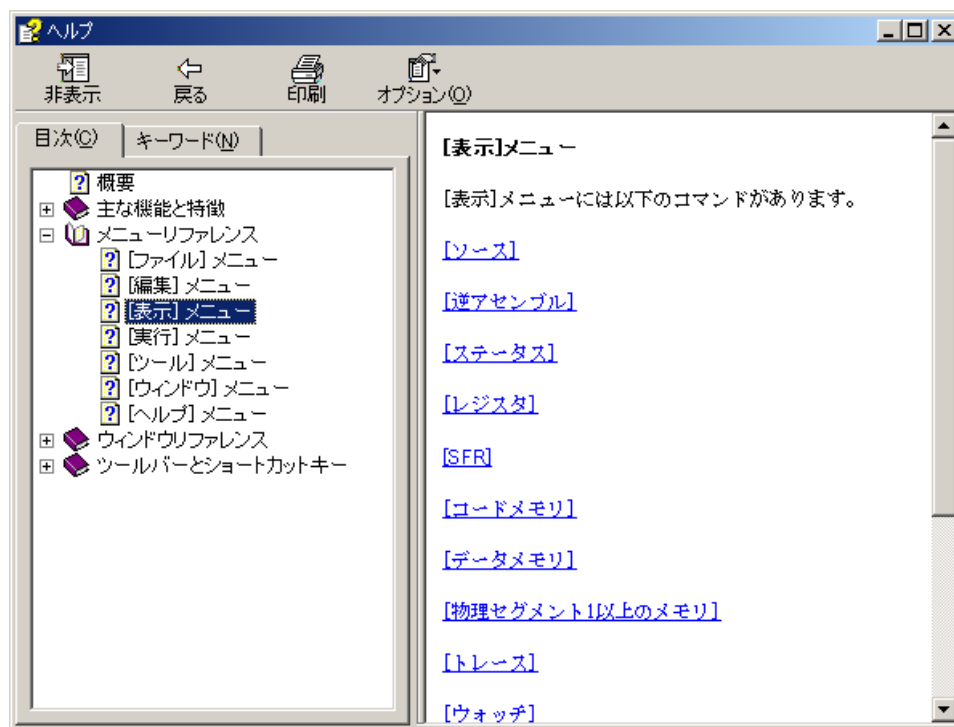
[デフォルトに戻す]ボタンをクリックすると、現在選択されているウィンドウに対するフォントと色の設定をデフォルトに戻すことができます。

各ウィンドウのフォント情報は、DTU8 デバッガのプロジェクトファイルに保存されます。

## 11.5 オンラインヘルプ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8 デバッガには、メニューや各ウィンドウのポップアップメニュー、ダイアログボックスのすべてに、HTML ヘルプフォーマットのオンラインヘルプが用意されています。

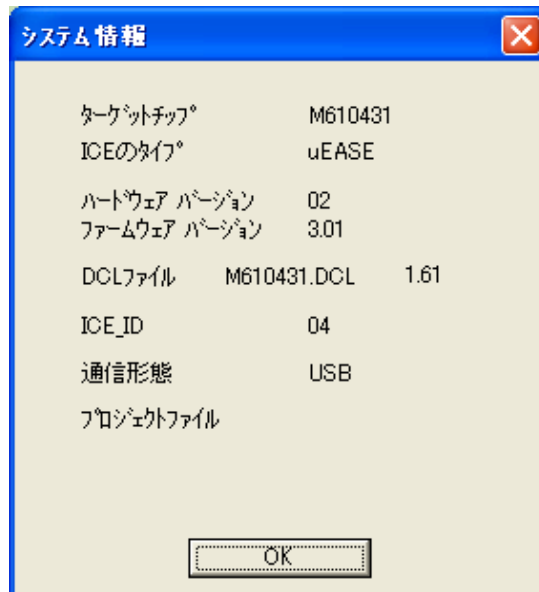


ユーザーズマニュアルは、各デバッガの機能のチュートリアルとなっていますので、メニューとウィンドウの完全なリファレンスは、オンラインヘルプを参照してください。

## 11.6 バージョン情報の表示

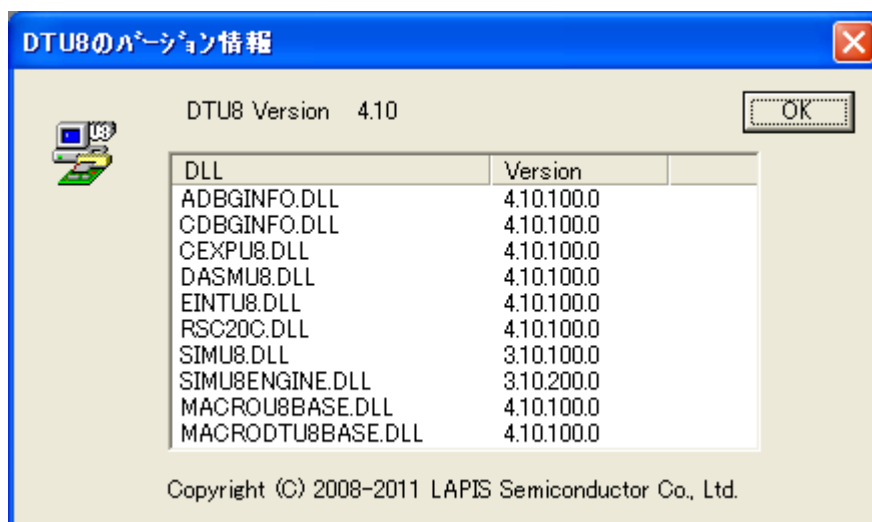
Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

[ヘルプ]メニューの[システム情報]を選択すると、ターゲットチップ、デバッグモード(ICE のタイプ)、接続しているエミュレータのハードウェアとファームウェアのバージョンやTCDファイル、DCLファイルのバージョン、通信形態および通信ボーレートとシリアルポート、プロジェクトファイル名等が表示されます。表示される内容は、デバッグのモードにより異なります。以下は、uEASE モードの場合の表示例です。



Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

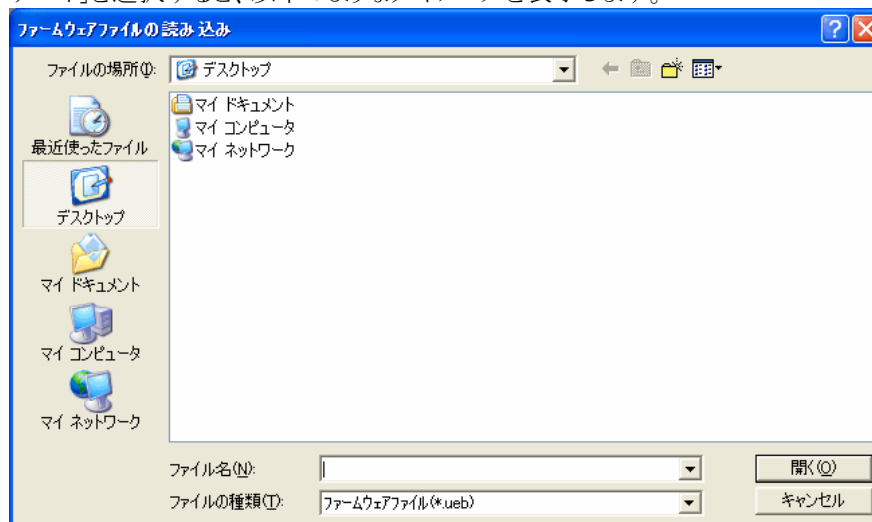
DTU8 デバッガの[ヘルプ]メニューの[バージョン情報]を選択すると、現在使用している DTU8 および関係する DLL のバージョンを表示します。



## 11.7 ファームウェア・アップデート機能

Dr.U8 ICE  
Dr.U16 ICE  
uEASE  
nanoEASE

DTU8 デバッガを使用して Dr.U8 ICE, Dr.U16 ICE, uEASE および nanoEASE オンチップデバッグエミュレータのファームウェアをアップデートすることができます。[ヘルプ]メニューの[ファームウェアのアップデート]を選択すると、以下のようなダイアログを表示します。



ファームウェアアップデート用ファイルを選択し、[開く]ボタンを押すとファームウェアのアップデートが実行されます。

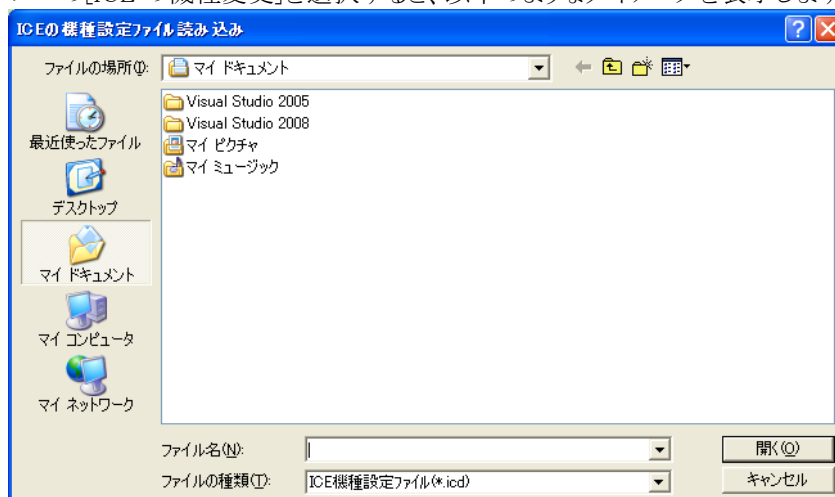
### ■ ファームウェアアップデート時の注意事項

ファームウェアのアップデートが完了したら、DTU8 デバッガを終了します。Dr.U8 ICE, Dr.U16ICE の場合は ICE の電源を一旦切り、再度電源を入れなおして再起動してください。uEASE, nanoEASE オンチップデバッグエミュレータの場合は、uEASE, nanoEASE オンチップデバッグエミュレータの USB ケーブルを一旦抜いてから挿しなおし uEASE, nanoEASE オンチップデバッグエミュレータを再起動してください。

## 11.8 対応機種変更機能

Dr.U8 ICE  
Dr.U16 ICE

DTU8 デバッガを使用して Dr.U8 ICE および Dr.U16 ICE の対応機種を変更することができます。[ヘルプ]メニューの[ICE の機種変更]を選択すると、以下のようなダイアログを表示します。



機種変更用ファイルを選択し、[開く]ボタンを押すと機種を変更することができます。

機種変更機能については、『Dr.U8 ICE ユーザーズマニュアル』または『Dr.U16 ICE ユーザーズマニュアル』をご参照ください。

## 11.9 リアルタイム LCD モニタ機能

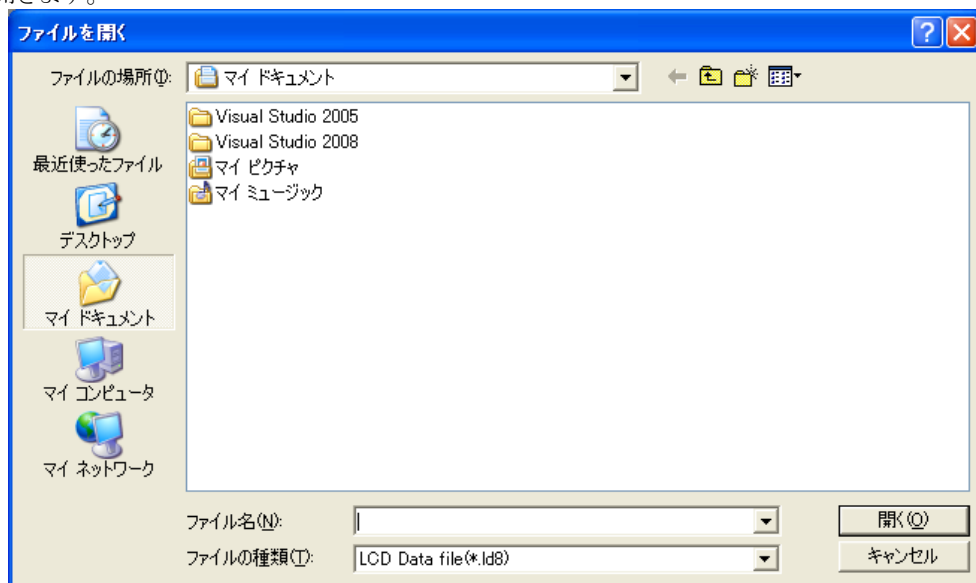
Dr.U8 ICE  
Dr.U16 ICE

リアルタイム LCD モニタ機能とは、Dr.U8 ICE から読み出した LCD 端子情報 (COM/SEG) を DTU8 経由で LCD イメージチェックツールに送り、LCD 表示イメージをリアルタイムに更新する機能です。

### 11.9.1 リアルタイム LCD モニタの使い方

本機能を使用する前に、LCD イメージアサインメントツールを使用して LCD パネルを定義し、定義内容を LCD データファイルに保存してください。LCD パネルの定義方法については、『LCD イメージツール ユーザーズマニュアル』の「5. LCD イメージアサインメントツール」を参照してください。

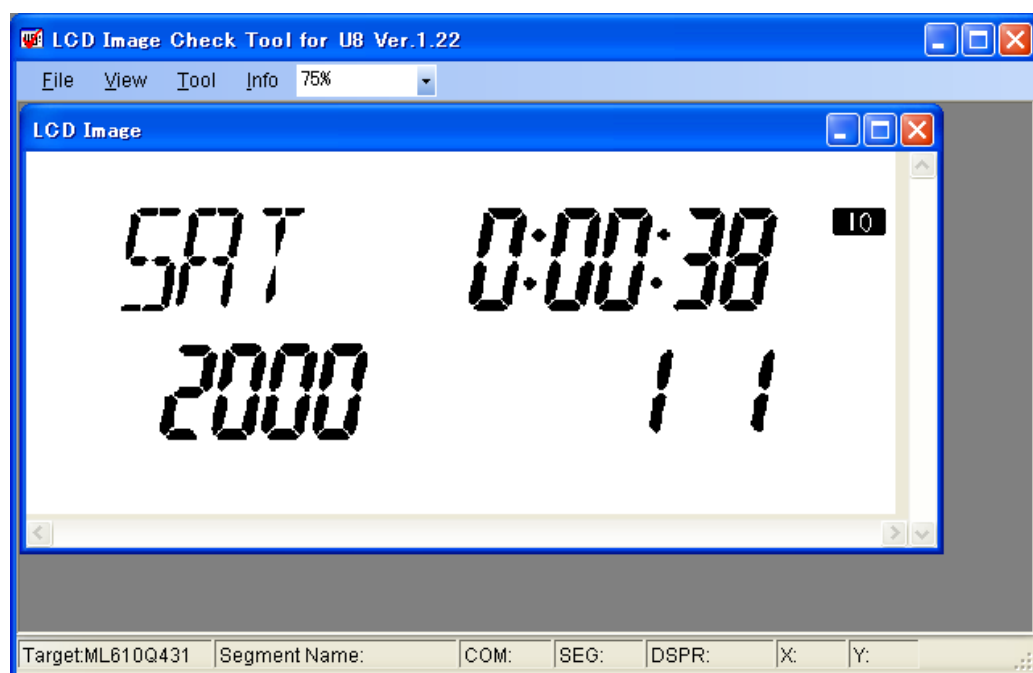
DTU8 で[ツール]メニューの[LCD モニタ]を選択すると、LCD データファイルを選択するダイアログが開きます。



作成済みの LCD データファイルを指定して OK ボタンを押すと、LCD イメージチェックツールが起動します。

LCD イメージチェックツールで、「Tool」->「Start Realtime LCD Monitor Mode」を選択し、リアルタイム LCD モニタモードを開始します。

DTU8 で LCD パネルを駆動するプログラムをロードし、エミュレーションを開始すると、プログラム動作にしたがって LCD イメージチェックツールの LCD パネル表示がリアルタイムに更新されます。





## 12 章 付録

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### 12.1 シンボリックデバッグと入力形式

#### 12.1.1 シンボル

シンボルは、ある特定の値を表す名前です。シンボルを使用することにより、定数で指定するよりは覚えやすい名前で、アドレスや数値を指定することができます。

DTU8 デバッガは、式中にある次のような文字列をシンボルとして認識します。

- \* 英字アルファベット(a～z A～Z)または特殊文字( \$ ?)で始まる文字列
- \* 2 文字目以降の文字が、英字アルファベット(a～z A～Z)、特殊文字( \$ ?)、または、数字(0～9)のいずれか
- \* タブ(09H)、スペース(20H)、キャリッジリターン(0DH)、演算子 (+ - \* / % & | ^ ! . < > = ( ) ), またはその他の特殊記号(¥ [ ] { } : ; @ # “ ‘ , )が現れる前まで

#### ■ シンボル情報の準備

DTU8 デバッガでシンボリックデバッグを行うためには、RASU8 アセンブラでデバッグ情報を生成する必要があります。

RASU8 アセンブラでデバッグ情報を生成するには、/D オプションを指定してください。

#### ■ シンボル値

シンボルの持つ値です。

#### ■ シンボルのアトリビュート

シンボルの持つ値の種類を表します。アトリビュートには、次のものがあります。

- |          |                 |
|----------|-----------------|
| * CODE   | コードアドレスシンボル     |
| * DATA   | データアドレスシンボル     |
| * NVDATA | NV データアドレスシンボル  |
| * TABLE  | テーブルアドレスシンボル    |
| * BIT    | ビットアドレスシンボル     |
| * NVBIT  | NV ビットアドレスシンボル  |
| * TBIT   | テーブルビットアドレスシンボル |
| * NUMBER | 数値シンボル          |

#### ■ シンボルのタイプ

シンボルが定義された時の種類を表します。シンボルタイプは、シンボルの検索順序に影響します。シンボルタイプには、リザーブドワードシンボルとユーザ定義シンボルがあります。

リザーブドワードシンボルは、ターゲットマイコンで使用されているシンボルです。主に、ターゲットマイコンの SFR 空間に割り付けられています。DTU8 デバッガの起動時に、機種情報ファイル(DCL

ファイル)から読み込まれ、値の変更はできません。

ユーザ定義シンボルは、RASU8 アセンブラのソースプログラム中でプログラマが定義するシンボルです。各種アセンブラの EQU, CODE, DATA 等の疑似命令やラベルによって登録されたシンボルは、すべてユーザ定義シンボルとなります。

## ■ シンボルの一覧表示

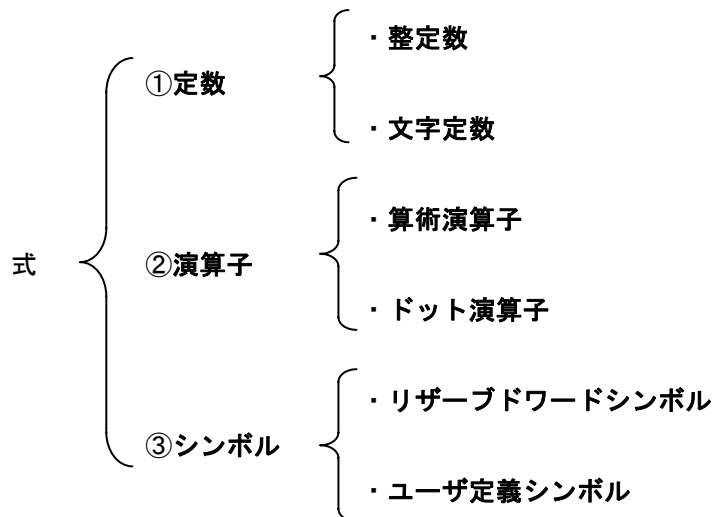
DTU8 デバッガの[ツール]メニューの[シンボルリスト]で、リザーブワードシンボルやユーザ定義シンボルの一覧を参照することができます。

またこのダイアログボックスでは、ユーザ定義シンボルの追加、削除、値の変更等が可能です。

### 12.1.2 式

DTU8 デバッガで扱う式は、定数、演算子、シンボルの3種類の要素から構成されています。

式の構成要素は、次のように表すことができます。



## ■ 定数

定数は、アドレス値やデータ値を直接指定するものです。定数には、整数定数と文字定数があります。

### － 整数定数

先頭文字が 0～9 までの数字で始まり、下表内の使用可能な文字で構成された文字列を整数定数として扱います。整数定数は、2, 8, 10, 16 進数での表現が可能です。これらの表現形式(基数)を区別するために、下表で示す基数指定子を数値の後に続けて(0x の場合は数値の前に)記述します。

先頭文字がレター属性を持つ文字(A～F, a～f, \_)となる場合は、シンボルと区別するために 0 を先頭につけます。また、整数定数中のアンダースコア( \_ )はスキップし、英字の大文字／小文字の区別は行いません。

基数	使用可能な文字	基数指定子	記述例
2 進数	0, 1, _	B	1010B
8 進数	0～7, _	O, Q	271O, 514Q
10 進数	0～9, _	何もつけない	1234
16 進数	0～9, A～F, a～f, _	H, 0x	753H, 0C6E7H
	※A～F, a～f で始まる値は、先頭に 0 を付加する。		0x332

### – 文字定数

文字定数は、一重引用符(')で囲まれた 1 文字またはエスケープシーケンスです。一重引用符の後が円記号(¥)以外であれば、文字定数は、その文字の ASCII コードを値として持ちます。一重引用符の後が円記号の場合は、その後の記述に従って 00H～0FFH の値が与えられます。円記号とそれに続く記述をエスケープシーケンスと呼びます。

### ■ 演算子

DTU8 デバッガの式中での演算子による演算は、32 ビット符号なしで行われます。演算結果が負であった場合は、2 の補数表現となります。オーバーフローは無視されます。

演算子には、算術演算子、ドット演算子があります。

### – 算術演算子

- + 2 項の加算
- 2 項の減算
- & 2 項の論理積

### – ドット演算子

. ((左項\*[ビット数])+右項)

この演算子は、ビットアドレスをデータアドレスとビット位置で表現するために用意された演算子です。

## 12.1.3 変更値リスト

ソースウィンドウ、逆アセンブルウィンドウのポップアップメニューの[値の変更]や、各種ダンプウィンドウのポップアップメニューの[xxx メモリの変更]を選択した場合、変更値の入力をリストで行うことができます。

変更値のリストには、即値またはシンボルが指定可能で、値と値の区切り文字には、カンマ(,)、空白、セミicolon(;)が使用できます。

変更値リストのデフォルト基数は、10 進数です。16 進数値を指定する場合は、10H のように”H”を付加する必要があります。また、A～F で始まる値には、0ABH のように先頭に 0 を付加する必要があります。

また、変更値リストには、-(マイナス)値を指定することはできません。

## 12.2 ウォッチウィンドウに登録可能な項目

ウォッチウィンドウには、以下の項目を登録することができます。

- C 変数
- アセンブリシンボル
- プログラムカウンタ (PC)
- プログラムステータスワード (PSW, EPSW, EPSW1~3, C, Z, S, OV, MIE, HC, ELEVEL)
- レジスタ直接  
(R0~15, ER0~14, XR0~12, QR0~8, LR, ELR, ELR1~2, EA, CSR, LCSR, ECR1~2, DSR, BP, FP, SP)
- レジスタビット直接 (Rn.bit)
- データメモリ直接 (アドレス)
- SFR (SFR 名)
- 汎用レジスタ間接アドレッシング [ERm]
- 6 ビットベース付 FP 間接アドレッシング Disp6[FP]
- 6 ビットベース付 BP 間接アドレッシング Disp6[BP]
- 16 ビットベース付汎用レジスタ間接アドレッシング Disp16[ERm]
- EA レジスタ間接アドレッシング [EA]

### 12.2.1 ローカル変数の登録について

ローカル変数をウォッチウィンドウへ登録する場合は、ローカル変数の定義されている関数スコープ内に現在の PC がなければなりません。

また、複数のブロックに同一名称のローカル変数が定義されている場合、ウォッチウィンドウに表示される値は、現在の PC を含むブロックのローカル変数の値となります。

ローカル変数をウォッチへ登録後、現在の PC が目的の関数に入っても、関数プロローグや関数エピローグでは、正しく値を表示できない場合があります。

レジスタに割り当てられたローカル変数をウォッチウィンドウへ登録する場合、変数がレジスタに割り当てられていることを通知する旨の確認ダイアログが表示されます。このダイアログは、環境設定のオプションにより表示しないようにすることができます。設定方法については、「付録 12.2.6 レジスタに割り当てられた変数について」を参照してください。

### 12.2.2 グローバル変数の登録について

グローバル変数は、現在の PC に関係なく、常にウォッチウィンドウへ登録できます。

型の違う同名のグローバル変数が定義されている場合、その変数をウォッチウィンドウへ登録すると、サイズの大きい方が登録されます。

### 12.2.3 関数引数の登録について

enum 型の関数引数をウォッチウィンドウへ登録した場合は、値に列挙シンボルが表示されません。

レジスタに割り当てられた関数引数をウォッチウィンドウへ登録する場合、変数がレジスタに割り当てられていることを通知する旨の確認ダイアログが表示されます。このダイアログは、環境設定の

オプションにより表示しないようにすることができます。設定方法については、「付録 12.2.6 レジスタに割り当てられた変数について」を参照してください。

#### 12.2.4 C 変数の型表示について

型宣言の複雑な変数をウォッチウィンドウへ登録した場合、「型」フィールドの内容が、宣言した通りに表示されない場合があります。

例えば、次のような宣言がこれに当たります。

```
extern void f1(void);
extern void f2(void);
extern void f3(void);
const void (*const func_ptr[3])(void) = { f1, f2, f3 };
```

ウォッチウィンドウへ関数へのポインタである func\_ptr を登録すると、「型」フィールドには、

```
void([3]*) ()
```

と表示されます。

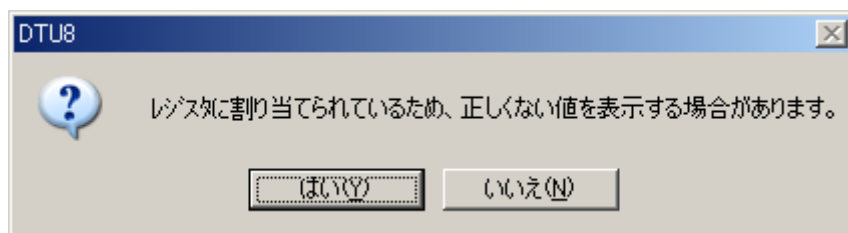
#### 12.2.5 C と ASM の混在プログラムについて

アセンブラ記述によって確保した領域を、C 記述上で配列の要素数を省略して extern 宣言している場合、その配列をウォッチウィンドウへ登録することはできません。このような場合は、extern 宣言にて明示的に配列のサイズを定義してください。

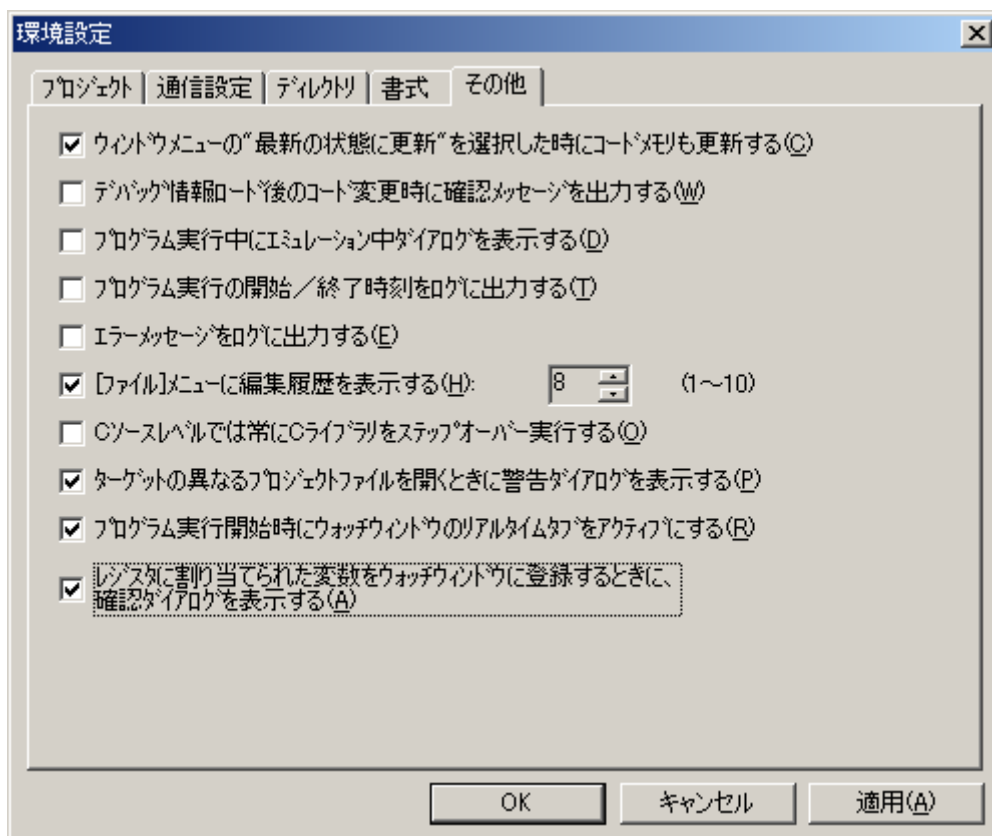
#### 12.2.6 レジスタに割り当てられた変数について

ローカル変数や関数引数がレジスタに割り当てられている場合、割り当てられたレジスタが一時的に別の用途に使用され、ウォッチウィンドウ上では期待する値と異なる値を表示する場合があります。

このため、レジスタに割り当てられたローカル変数や関数引数をウォッチウィンドウに登録する際、DTU8 デバッガは以下のような確認ダイアログを表示します。



この確認ダイアログは、[環境設定]ダイアログのオプションにより表示しないようにすることができます。



[環境設定]ダイアログの[その他]タブで、「レジスタに割り当てられた変数をウォッチウィンドウに登録するときに、確認ダイアログを表示する」チェックボックスのチェックをはずすと、前述の確認ダイアログが表示されないようになります。

## 12.3 最適化された C ソースコードのデバッグ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

CCU8 コンパイラによって最適化された C ソースコードをロードすると、本来あるべき実行文にアドレスが表示されない、あるいはブレークポイントが設定できないなどの状況になることがあります。これは、コンパイラの最適化であるコードの畳み込みや冗長コードの削除、コードの移動等が発生しているためで、DTU8 デバッガのソースウィンドウ上での動作が、C 記述上ではあり得ない順序で行われることもあります。

### ■ 例

以下は、最適化によって、コードが削除されている例です。

```

Line BP Address
00033 void dead_code3(void)
00034 {
00035 00:00C4H a = 10;
00036 00:00CAH r = p + q;
00037 if (a == 10) {
00038 00:00D8H fn1();
00039 } else if (a > 10) {
00040 fn2(); // 最適化されるコード
00041 } else {
00042 r = p * q; // 最適化されるコード
00043 }
00044 }

```

Line39～43 は、C 記述上ではコードが存在しなければなりませんが、CCU8 コンパイラの最適化によって、削除されているため、[BP]ボタンやアドレスが表示されていないことがわかります。

## 12.4 マクロスクリプトリファレンス

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

### 12.4.1 スクリプトコマンド一覧

#### ■ プロジェクトに関するスクリプトコマンド

PJLOD	プロジェクトファイルの読み込み
PJSAV	プロジェクトファイルの保存

#### ■ CPU の内容を参照変更するスクリプトコマンド

D	SFR, レジスタの内容を表示
C	SFR, レジスタの変更
DPC	PC を表示
CPC	PC を変更

#### ■ C 変数を参照変更するスクリプトコマンド

WATCH	C 変数の内容を表示
-------	------------

#### ■ コードメモリに関するスクリプトコマンド

DCM	コードメモリの内容を表示
CCM	コードメモリの変更
MCM	コードメモリのコピー
LOD	プログラムファイルの読み込み
SAV	プログラムファイルの保存
VER	プログラムファイルとコードメモリの比較
ASM	指定文字列のアセンブル
STARTASM	連続アセンブルの開始
ENDASM	連続アセンブルの終了
DASM	逆アセンブル
MMAP	コードメモリのマッピングの内容確認・変更

### ■ データメモリに関するスクリプトコマンド

DDM	データメモリの内容を表示
CDM	データメモリの変更
MDM	データメモリのコピー
DLOD	データメモリファイルの読み込み
DSAV	データメモリファイルの保存
PMAP	データメモリのマッピングの内容確認・変更
SFRSAV	SFR の内容をファイルへ保存

### ■ 物理セグメント1以上のメモリに関するスクリプトコマンド

DGM	物理セグメント1以上のメモリの内容を表示
CGM	物理セグメント1以上のメモリの変更
MGM	物理セグメント1以上のメモリのコピー
GMAP	物理セグメント1以上のメモリのマッピングの内容確認・変更

### ■ エミュレーションに関するスクリプトコマンド

G	プログラムの実行
STP	ステップイン実行
STPOV	ステップオーバー実行

### ■ リセットに関するスクリプトコマンド

RST	リセット
-----	------

### ■ ブレークに関するスクリプトコマンド

SBC	ブレーク条件の表示・設定・解除
DBP	ブレークポイントリストの表示
SBP	ブレークポイントの設定
RBP	ブレークポイントの解除
DBS	ブレークステータスの表示

### ■ パフォーマンス／カバレッジに関するスクリプトコマンド

TIME	タイマ値の表示
CCC	サイクルカウンタの設定
DCC	サイクルカウンタの表示
DCV	カバレッジの表示
SCV	カバレッジ測定範囲の設定

### ■ マクロ実行に関するスクリプトコマンド

PAUSE	マクロ実行のポーズ
ECHO	指定文字列をログへ出力
GOTO	マクロ実行行のラベルジャンプ
IF	マクロ実行行の条件分岐

### ■ シンボルの表示／変更／削除に関するスクリプトコマンド

DSYM	シンボルの内容を表示
------	------------



CSYM	シンボルの内容を変更
RSYM	シンボルの削除
SSYM	シンボルの登録

### ■ その他のスクリプトコマンド

VERSION	DTU8 のバージョン情報を表示
EXIT	DTU8 の終了
SAVELOG	マクロ実行結果(ファイル)を保存
CLRLOG	ログのクリア

## 12.4.2 プロジェクトに関するコマンド

ディスクファイル内のプロジェクトをDTU8へロードする。

<b>【書式】</b>	ディスクファイル内のプロジェクトファイルをロード
PJLOD <i>filename</i>	
<i>filename:</i>	ロードプロジェクトファイル名

プロジェクトの内容をディスクファイルにセーブする。

<b>【書式】</b>	プロジェクトの内容をディスクファイルにセーブ
PJSAV <i>filename</i>	
<i>filename:</i>	セーブファイル名

## 12.4.3 CPU の内容を参照／変更するコマンド

SFR, レジスタの内容を表示する。

<b>【書式】</b>	SFR, レジスタの内容を表示
D <i>parm</i>	
<i>parm:</i>	<i>Reg_mnemonic</i> または SFR
<i>Reg_mnemonic:</i>	PSW                    プログラムステータスワード
	EPSW                    PSW 退避レジスタ
	EPSW1～2PSW 退避    レジスタ 1～2
	LR                      リンクレジスタ
	ELR                    リンクレジスタアドレッシング
	ELR1～2                リンクレジスタ 1～2
	CSR                    コードセグメントレジスタ
	LCSR                   LCSR レジスタ
	ECSR1～2CSR 退避    レジスタ 1～2
	DSR                    データセグメントレジスタ
	EA                      EA レジスタ
	SP                      スタックポインタ
	R0～R15                バイト型レジスタ 0～15
	ER0～ER14             ワード型レジスタ 0～14
	XR0～XR12             ロングワード型レジスタ 0～12
	QR0～QR8              クワッドワード型レジスタ 0～8

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

SFR, レジスタの内容を変更する。

【書式1】 レジスタの内容を変更

C *Reg\_mnemonic* = *data*

<i>data</i> :	変更するデータ
<i>Reg_mnemonic</i> :	PSW                    プログラムステータスワード
	EPSW                    PSW 退避レジスタ
	EPSW1～2PSW 退避    レジスタ 1～2
	LR                      リンクレジスタ
	ELR                    リンクレジスタアドレッシング
	ELR1～2                リンクレジスタ 1～2
	CSR                    コードセグメントレジスタ
	LCSR                   LCSR レジスタ
	ECSR1～2CSR 退避    レジスタ 1～2
	DSR                    データセグメントレジスタ
	EA                     EA レジスタ
	SP                     スタックポインタ
	R0～R15                バイト型レジスタ 0～15
	ER0～ER14             ワード型レジスタ 0～14
	XR0～XR12             ロングワード型レジスタ 0～12
	QR0～QR8              クワッドワード型レジスタ 0～8

【書式2】 SFR の内容を変更

C *Sfr\_mnemonic* = *data*

<i>Sfr_mnemonic</i> :	sfr 空間に割り付けてあるレジスタ
<i>data</i> :	変更するデータ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

プログラムカウンタの内容を表示する。

【書式】 プログラムカウンタの内容を表示

DPC

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

プログラムカウンタの内容を変更する。

【書式】 プログラムカウンタの内容を変更

CPC *address*

<i>address</i> :	変更するプログラムカウンタ値 (コードアドレス)
------------------	--------------------------

## 12.4.4 C変数を参照／変更するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

C変数の内容を参照する。

【書式1】 C 変数の内容を表示

WATCH *valueable\_name*

<i>valueable_name</i> :	C 変数名
-------------------------	-------

## 12.4.5 コードメモリに関するコマンド

コードメモリの内容を表示する。

【書式1】 単一アドレスのコードメモリ内容表示

DCM *address**address*:

コードメモリアドレス

【書式2】 アドレス範囲のコードメモリ内容表示

DCM [*start-address end-address*]*start-address*:

表示開始コードメモリアドレス

*end-address*:

表示終了コードメモリアドレス

コードメモリの内容を変更する。

【書式1】 単一アドレスのコードメモリ内容変更

CCM *address* = *data**address*:

コードメモリアドレス

*data*:

変更するデータ

【書式2】 アドレス範囲のコードメモリ内容変更

CCM [*start-address end-address*] = *data**start-address*:

変更開始コードメモリアドレス

*end-address*:

変更終了コードメモリアドレス

*data*:

変更するデータ

コードメモリの内容を指定アドレス以降に複写する。

【書式】 コードメモリの複写

MCM [*start-address end-address*] *trans-address**start-address*:

転送元コードメモリ開始アドレス

*end-address*:

転送元コードメモリ終了アドレス

*trans-address*:

転送先コードメモリ開始アドレス

ディスクファイル内のプログラムをコードメモリへロードする。

【書式】 ディスクファイル内のプログラムをコードメモリへロード

LOD *filename* 【*option* ・ ・ *option*】*filename*:

ロードファイル名

*option*:

/NS

デバッグ情報をロードしない

/N

コード情報をロードしない

/B

読み込み後、全ブレークポイントビットをクリアする

/R

読み込み後、リセットする

コードメモリの内容をディスクファイルにセーブする。

【書式】 コードメモリ内容をディスクファイルにセーブ

SAV *filename format* [[*start-address end-address*]] 【*option* ・ ・ *option*】*filename*:

セーブファイル名

*format*:

HEX

インテル HEX 形式で保存

S

モトローラ S フォーマットで保存

*start-address*:

セーブ開始アドレス

*end-address*:

セーブ終了アドレス

*option*:

/NS

シンボル情報をセーブしない

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
SimulateDr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
SimulateDr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
SimulateDr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
SimulateDr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

VER *filename*  
*filename*: 比較ファイル名

【書式 2】 ディスクファイル内の指定アドレス範囲の内容と コードメモリの内容を比較

VER *filename* [*start-address end-address*]  
*filename*: 比較ファイル名

アセンブルを行う。

【書式1】 指定文字列のアセンブル(単一アドレス指定型)

ASM address “assemble-string”	コードメモリアドレス
address:	

【書式2】 指定文字列のアセンブル(連続型)

## ASM “*assemble-string*”

※ この書式を使用する場合は、STARTASM コマンドで開始アドレスを指定し、ENDASM コマンドまで連続して記述することができる。

アセンブルを開始する。

【書式】 連続アセンブルの開始

**STARTASM** *address*

*address:*                      コードメモリアドレス

アセンブルを終了する。

【書式】 連続アセンブルの終了

ENDASM

指定されたコードメモリのアドレス範囲の逆アセンブルを行う。

【書式 1】 単一アドレスの逆アセンブル

DASM address	【option · · option】	
address:		コードメモリアドレス
option:	/NC	インストラクションコードを表示しない
	/NL	アドレスを表示しない

【書式2】 アドレス範囲の逆アセンブル

DASM [ <i>start-address end-address</i> ] <b>【option · · option】</b>	
<i>start-address</i> :	逆アセンブル開始アドレス
<i>end-address</i> :	逆アセンブル終了アドレス
<i>option</i> :	インストラクションコードを表示しない
	/NL アドレスを表示しない

## コードメモリのマッピング

Dr.ICE  
Simulate

【書式 1】 現在のコードメモリマッピングの設定内容出力  
MMAP

【書式 2】 コードメモリマッピングの設定

MMAP [*start-address end-address*] = *parm*

<i>start-address</i> :	先頭アドレス
<i>end-address</i> :	終了アドレス
<i>parm</i> :	SC システムメモリ空間
	UC ユーザメモリ空間
	N 未使用領域空間

## 12.4.6 データメモリに関するコマンド

## データメモリの内容を表示する。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

【書式 1】 単一アドレスのデータメモリの内容表示

DDM *address*

*address*: データメモリアドレス

【書式 2】 アドレス範囲のデータメモリの内容表示

DDM [*start-address end-address*]

<i>start-address</i> :	表示開始データメモリアドレス
<i>end-address</i> :	表示終了データメモリアドレス

## データメモリの内容を変更する。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

【書式 1】 単一アドレスのデータメモリの内容変更

CDM *address* = *data*

<i>address</i> :	データメモリアドレス
<i>data</i> :	変更するデータ

【書式 2】 アドレス範囲のデータメモリの内容変更

CDM [*start-address end-address*] = *data*

<i>start-address</i> :	変更開始データメモリアドレス
<i>end-address</i> :	変更終了データメモリアドレス
<i>data</i> :	変更するデータ

※ アドレスには、SFR 領域は指定不可。

## データメモリの内容を指定アドレス以降に複写する。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

【書式】 データメモリの複写

MDM [*start-address end-address*] *trans-address*

<i>start-address</i> :	転送元データメモリ開始アドレス
<i>end-address</i> :	転送元データメモリ終了アドレス
<i>trans-address</i> :	転送先データメモリ開始アドレス

※ アドレスには、SFR 領域は指定不可。

## ディスクファイル内のデータファイルをデータメモリへロードする。

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

【書式】 ディスクファイル内のデータファイルをデータメモリへロード

DLOD *filename*

*filename*: ロードファイル名

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

データメモリの内容をディスクファイルにセーブする。

【書式】 データメモリ内容をディスクファイルにセーブ  
 DSAV *filename* **[[start-address end-address]]**  
*filename*: セーブファイル名  
*start-address*: セーブ開始アドレス  
*end-address*: セーブ終了アドレス  
 ※ アドレスには、SFR 領域は指定不可。

Dr.ICE  
Simulate

データメモリのマッピング

【書式 1】 現在のデータマッピングの設定内容出力  
 PMAP

【書式 2】 データメモリマッピングの設定  
 PMAP **[start-address end-address] = parm**  
*start-address*: 先頭アドレス  
*end-address*: 終了アドレス  
*parm*: SD システムメモリ空間  
 UD ユーザメモリ空間  
 N 未使用領域空間

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

SFRの内容をファイルへ保存

【書式】 SFR 内容をディスクファイルにセーブ  
 SFRSAV *filename* **[[start-address end-address]]**  
*filename*: セーブファイル名  
*start-address*: セーブ開始アドレス  
*end-address*: セーブ終了アドレス  
 ※ アドレスには TRG ファイルで指定した領域のみ指定可能。

## 12.4.7 物理セグメント 1 以上のメモリに関するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

物理セグメント1以上のメモリの内容を表示する。

【書式 1】 単一アドレスの物理セグメント 1 以上のメモリの内容表示  
 DGM *address*  
*address*: 物理セグメント 1 以上のメモリアドレス  
 【書式 2】 アドレス範囲の物理セグメント 1 以上のメモリの内容表示  
 DGM **[start-address end-address]**  
*start-address*: 表示開始物理セグメント 1 以上のメモリアドレス  
*end-address*: 表示終了物理セグメント 1 以上のメモリアドレス

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

物理セグメント1以上のメモリの内容を変更する。

【書式 1】 単一アドレスの物理セグメント 1 以上のメモリの内容変更  
 CGM *address* = *data*  
*address*: 物理セグメント 1 以上のメモリアドレス  
*data*: 変更するデータ  
 【書式 2】 アドレス範囲の物理セグメント 1 以上のメモリの内容変更  
 CGM **[start-address end-address] = data**  
*start-address*: 変更開始物理セグメント 1 以上のメモリアドレス  
*end-address*: 変更終了物理セグメント 1 以上のメモリアドレス  
*data*: 変更するデータ

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

物理セグメント1以上のメモリの内容を指定アドレス以降に複写する。

【書式】 物理セグメント1以上のメモリの複写

MGM [*start-address end-address*] *trans-address*

*start-address*: 転送元物理セグメント1以上のメモリ開始アドレス  
*end-address*: 転送元物理セグメント1以上のメモリ終了アドレス  
*trans-address*: 転送先メモリ開始アドレス

Dr.ICE  
Simulate

物理セグメント1以上のメモリのマッピング

【書式1】 現在の物理セグメント1以上のメモリのマッピングの設定内容出力

GMAP

【書式2】 物理セグメント1以上のメモリのマッピングの設定

GMAP [*start-address end-address*] = *parm*

<i>start-address</i> :		先頭アドレス
<i>end-address</i> :		終了アドレス
<i>parm</i> :	SC	システムコード空間
	SD	システムデータ空間
	SCD	システムコード・データ空間
	UC	ユーザコード空間
	UD	ユーザデータ空間
	UCD	ユーザコード・データ空間
	N	未使用領域空間

## 12.4.8 エミュレーションに関するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

リアルタイムエミュレーションを行う。

【書式1】 エミュレーション実行

G

G *address*

【書式2】 単一アドレスブレークのエミュレーション実行

G 【*address*】 , *break-address*

*address*: ユーザプログラムの実行開始アドレス  
*break-address*: ブレークアドレス

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ステップオーバー実行を行う。

【書式】 ステップ実行

STPOV 【*address*】 【 , *count* 】

*address*: ステップ実行開始アドレス  
*count*: 実行ステップ数

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ステップイン実行を行う。

【書式】 ステップイン実行

STP 【*address*】 【 , *count* 】

*address*: ステップ実行開始アドレス  
*count*: 実行ステップ数

12.4.9 リセットに関するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

エミュレータのシステム全体を初期化する。

【書式】 エミュレータのシステム全体を初期化  
RST

12.4.10 ブレークに関するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

ブレークコンディションの設定を行う。

【書式 1】 現在のブレークコンディションの設定内容出力  
SBC

【書式 2】 トレースフルブレークを有効  
SBC TF *overflow*  
*overflow:* トレースオーバーフローポイント

【書式 3】 エクスターナルブレークコンディションを有効  
SBC XP *condition*  
*condition:* DOWN High から Low への変化でブレーク  
UP Low から High への変化でブレーク

【書式 4】 エクスターナルブレークコンディション 2 を有効  
SBC XP2 *condition*  
*condition:* DOWN High から Low への変化でブレーク  
UP Low から High への変化でブレーク

【書式 5】 パワーダウンブレークコンディションを有効  
SBC PD

【書式 6】 ブレークコンディションの解除  
SBC ~*mnemonic*  
*mnemonic:* PD パワーダウンブレーク  
\* すべてのブレークを解除する

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ブレークポイントの内容を表示する。

【書式】 ブレークポイントリストの内容を表示  
DBP

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ブレークポイントを設定する。

【書式】 ブレークポイントを設定  
SBP *address*  
*address:* ブレークポイントアドレス

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ブレークポイントの内容をクリアする。

【書式】 ブレークポイントビットの内容をクリア  
RBP *bp-address*  
*bp-address:* *address* ブレークポイントアドレス  
\* すべてのブレークポイントをクリア



【書式】 ブレークステータス(ブレーク要因)を表示  
DBS

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

【書式1】 全てのトレースメモリの内容を表示  
DTM

【書式3】 新しいデータから表示ステップ数分のトレースメモリの内容を表示  
DTM BACK *step\_number*  
*step\_number*: 表示ステップ数

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

**【書式】** トレースポイントの内容を表示  
DTP

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

【書式】                      トレースポイントの内容をクリア  
RTP

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

<b>【書式1】</b>	トレースメモリの全内容をディスクファイルへセーブ
TRSAV <i>filename</i>	
<i>filename</i> :	セーブファイル名

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE

**【書式】** タイマの値を表示  
TIME

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

【書式】 サイクルカウンタの値を表示  
DCC

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
Simulate

サイクルカウンタの値を設定する。

【書式】 サイクルカウンタの値を設定  
CCC *count*  
*count*: サイクルカウンタの値

Dr.U16 ICE  
Simulate

カバレッジを表示する。

【書式】 カバレッジと設定されたカバレッジ測定範囲を表示  
DCV

Dr.U16 ICE  
Simulate

カバレッジの測定範囲を設定する。

【書式 1】 設定されているカバレッジ測定範囲を表示する  
SCV

【書式 2】 カバレッジ測定範囲の開始アドレス／終了アドレスを設定する  
SCV [*start\_address end\_address*]  
*start\_address*: カバレッジ測定範囲の開始アドレス  
*end\_address*: カバレッジ測定範囲の終了アドレス

### 12.4.13 マクロの実行に関するコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

マクロの実行を任意のキー(スペース以外)が入力されるまで一時中断する。

【書式】 マクロの実行を一時中断  
PAUSE

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

指定文字列を出力する。

【書式】 指定文字列をログへ出力  
ECHO *statement*  
*statement*: *statement perm*  
*perm*: 二重引用符で囲んだ文字列  
\$TIME 時刻を表示  
\$DATE 年月日を表示

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

マクロ実行行を指定ラベルへジャンプする。

【書式】 マクロ実行行の指定ラベルへのジャンプ  
GOTO *label*  
:*label*

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

マクロ実行行の条件分岐。

【書式】 マクロ実行行の条件分岐  
IF *compare statement*  
*compare*: *value-comp*  
*lastline-comp*  
*value-comp*: *obj cond value*  
*obj*: PSW, EPSW, EPSW1～EPSW2  
LR, ELR, ELR1～ELR2  
CSR, LCSR, ECSR1～ECSR2  
DSR, EA

R0～R15, ER0～ER14, XR0～XR12, QR0～QR8  
 SP, PC  
 SFR sfr 名  
*data-address*  
 CVAR C 変数名  
*cond*: ==, !=  
*value*: 式  
*data-address*: データアドレス式  
*lastline-comp*: LASTLINE *cond* “*string*” 直前行全体の文字列と比較  
 LASTLINE (n) *cond* “*string*” 直前行の n 番目の単語と比較  
*statement*: THEN *command*  
 GOTO *label*  
*command*: 任意の 1 つのスクリプトコマンド

#### 12.4.14 シンボルの表示／変更／削除に関するコマンド

登録されているユーザ定義シンボルを表示する。

【書式】 登録されているユーザ定義シンボルを表示

DSYM *symbol*  
*symbol*: シンボル名 (ワイルドカード記述が可能)

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

登録されているユーザ定義シンボルの値を変更する。

【書式】 登録されているユーザ定義シンボルの値を変更

CSYM *symbol* = *data*  
*symbol*: シンボル名  
*data*: シンボル値

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

登録されているユーザ定義シンボルを削除する。

【書式】 登録されているユーザ定義シンボルを削除

RSYM *symbol*  
*symbol*: シンボル名 (ワイルドカード記述が可能)

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

ユーザ定義シンボルを追加する。

【書式】 ユーザ定義シンボルを追加

SSYM *symbol* = *data* 【*attribute*】  
*symbol*: シンボル名  
*data*: シンボル値  
*attribute*: シンボル属性  
 /N NUMBER  
 /C CODE  
 /D DATE  
 /T TABLE  
 /ND NVDATA  
 /B BIT DATA  
 /TB TABLE BIT  
 /NB NVBIT

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

12.4.15 その他のコマンド

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8デバッガのバージョンを表示する。  
【書式】 DTU8 デバッガのバージョンを表示  
VERSION

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

DTU8デバッガを終了する。  
【書式】 DTU8 デバッガを終了  
EXIT

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

マクロの実行結果の保存を行う。  
【書式】 マクロの実行結果の保存  
SAVELOG *logname*  
*logname*: LOG 情報を格納するファイル名

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE  
Simulate

LOG情報をクリアする。  
【書式】 LOG 情報をクリア  
CLRLOG

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE  
uEASE  
nanoEASE

## 12.5 エミュレータ接続時のエラーについて

エミュレータ (Dr.U8 ICE, Dr.U16 ICE, Dr.ICE インサーキットエミュレータ、uEASE または nanoEASE オンチップデバッグエミュレータ) を接続して DTU8 を起動しているときにエラーコードを受け取った場合、DTU8 はそのエラーコードに対するエラーメッセージを表示します。

ここでは、エミュレータ接続時に表示されるエラーメッセージとそのエラーに対する対処方法について示します。

### 12.5.1 Dr.U8 ICE / Dr.U16 ICE / Dr.ICE / uEASE / nanoEASE 接続時の共通のエラーについて

ここでは、Dr.U8 ICE、Dr.U16 ICE、Dr.ICE、uEASE または nanoEASE のいずれかのエミュレータと接続している時に発生するエラーのうち、エミュレータの種類によらず、エラーの内容や対処方法が同じものを示します。

エラーメッセージ	説明および対処方法
USB：デバイスのオープンに失敗しました。	USB による通信が失敗した場合に左記エラーが表示されます。  USB ケーブルおよび接続が正しいかを確認してください。
USB：デバイスのクローズに失敗しました。	
USB：EP $n$ パイプが開けません	
USB：パイプのクローズに失敗しました	
USB：バルクアウト転送に失敗しました	
USB：バルクイン転送に失敗しました	
USB：不正なデータを受信しました	DTU8 または接続しているエミュレータが誤動作をしている可能性があります。  エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー： 6000H 指定されたデータ型が LWORD/WORD/BYTE/BIT 以外です。	DTU8 またはエミュレータが誤動作をしている可能性があります。  エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー： 6001H 開始・終了アドレスが指定できる範囲を超えています。	

エラーメッセージ	説明および対処方法
ICE エラー: 6002H 指定されたリソース No が存在しません。	DTU8 またはエミュレータが誤動作をしている可能性があります。 エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6003H データ個数が設定出来る範囲を超えています。	DTU8 またはエミュレータが誤動作をしている可能性があります。DTU8 を起動中に USB ケーブルを抜き差しした場合にもこのエラーが表示される場合があります。 エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6005H 比較条件データが範囲外です。	DTU8 またはエミュレータが誤動作をしている可能性があります。 エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6006H ブレーク No が存在しません。	
ICE エラー: 6011H ハードウェアブレークポイントが設定可能数を超えています。	設定可能数を超えてハードウェアブレークポイントを設定しようとした場合に表示されます。 DTU8 またはエミュレータが誤動作をしている可能性があります。 エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6200H エミュレーション中のため実行できません。	指定されたコマンドはエミュレーション中では実行できません。ブレークした状態で指定してください。

エラーメッセージ	説明および対処方法
ICE エラー: 6201H エミュレーション又はステップ中ではありません。	DTU8 またはエミュレータが誤動作をしている可能性があります。 エミュレータの接続を確認し、DTU8 およびエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6300H ダイレクトインストラクションの動作が異常です。	
ICE エラー: 630AH ターゲット情報が設定されていません。	
ICE エラー: 630CH ターゲット LSI の情報が取得できません。	
ICE エラー: 630DH ファームウェアのアップデートモード設定に失敗しました。接続を確認し、DTU8 と uEASE を再起動してください。	
ICE エラー: 6FFAH ICE から予期しないエラーを受信しました。	

### 12.5.2 uEASE / nanoEASE 接続時のエラー

**uEASE  
nanoEASE**

ここでは、uEASE または nanoEASE と接続している時に表示されるエラーメッセージを示します。Dr.U8 ICE 接続時と同じエラーメッセージが表示される場合でも、対処方法が異なる場合は、ここに示しています。

エラーメッセージ	説明および対処方法
ICE エラー: 0001H ターゲットシステムとの通信が切断されました。DTU8 およびターゲットシステムを再起動してください。	ユーザターゲットボードとの通信が切断されました。uEASE / nanoEASE とユーザターゲットボードの接続を確認し、DTU8、uEASE/nanoEASE、およびターゲットボードを再起動してください。

エラーメッセージ	説明および対処方法
ICE エラー: 6100H 書き込み時のベリファイ処理で値が一致しません。	uEASE / nanoEASE と接続しているターゲットマイコンのフラッシュメモリが破損している可能性があります。DTU8 を起動中に USB ケーブルを抜き差しした場合にもこのエラーが表示される場合があります。 uEASE / nanoEASE の接続を確認し、DTU8 および、uEASE / nanoEASE を再起動した後、再度書き込みを行ってみてください。それでもこのエラーが検出される場合には、ターゲットマイコンを交換してください。
ICE エラー: 6302H リセットに失敗しました。	uEASE / nanoEASE とユーザターゲットボードの接続を確認し、DTU8、uEASE/nanoEASE、およびユーザターゲットボードを再起動してください。再起動してもエラーが発生する場合は、ターゲットマイコンを交換してください。
ICE エラー: 6303H FLASH メモリへのアクセスに失敗しました。	uEASE / nanoEASE とユーザターゲットボードの接続を確認し、DTU8、uEASE/nanoEASE、およびユーザターゲットボードを再起動してください。再起動してもエラーが発生する場合は、ターゲットマイコンを交換してください。
ICE エラー: 6304H タイムアウトが発生しました。	uEASE / nanoEASE とユーザターゲットボードの接続に問題がある可能性があります。接続を確認し、再起動してください。
ICE エラー: 6305H ターゲット LSI の電源電圧(VTref)がフラッシュメモリ制御に必要な電圧を満たしていません。	ターゲットマイコンの電源電圧 (VTref) がフラッシュメモリ制御に必要な電圧を満たしていません。電源を確認してください。
ICE エラー: 6306H ブロックアドレスが設定できる範囲を超えています。	DTU8 または uEASE / nanoEASE が誤動作をしている可能性があります。 uEASE / nanoEASE の接続を確認し、DTU8 および uEASE / nanoEASE を再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。



エラーメッセージ	説明および対処方法
ICE エラー: 6307H On-Chip ICE ブロックの起動に失敗しました。	uEASE / nanoEASE とユーザターゲットボードの接続を確認し、DTU8、uEASE / nanoEASE、およびユーザターゲットボードを再起動してください。再起動してもエラーが発生する場合は、ターゲットマイコンを交換してください。
ICE エラー: 6308H uEASE が供給する VDDL 電圧値が異常です。	uEASE / nanoEASE が誤動作している可能性があります。 uEASE / nanoEASE の接続を確認し、DTU8 および、uEASE / nanoEASE を再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6309H ターゲットの VTref 電圧が異常です。ターゲットの電源を確認し、DTU8 およびターゲットシステムを再起動してください。	ターゲットの VTref 電圧が正常範囲外です。ターゲットの電源を確認し、DTU8 およびターゲットボードを再起動してください。
ICE エラー: 630BH エミュレーション中のみに受信可能なファンクションを受信しました。	DTU8 または uEASE / nanoEASE が誤動作をしている可能性があります。 uEASE / nanoEASE の接続を確認し、DTU8 および、uEASE / nanoEASE を再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。

## 12.5.3 Dr.U8 ICE / Dr.U16 ICE / Dr.ICE 接続時のエラー

Dr.U8 ICE  
Dr.U16 ICE  
Dr.ICE

ここでは、Dr.U8 ICE, Dr.U16 ICE, Dr.ICE と接続している時に表示されるエラーメッセージを示します。uEASE 接続時と同じエラーメッセージが表示される場合でも、対処方法が異なる場合は、ここに示しています。

エラーメッセージ	説明および対処方法
ICE エラー: 6100H 書き込み時のバリファイ処理で値が一致しません。	DTU8 または接続している Dr.U8 ICE / Dr.U16 ICE / Dr.ICE インサーキットエミュレータが誤動作をしている可能性があります。接続を確認し、DTU8 とエミュレータを再起動してください。再起動してもエラーが発生する場合は、エラーの発生状況を弊社営業までご連絡ください。
ICE エラー: 6302H リセットに失敗しました。	
ICE エラー: 6304H タイムアウトが発生しました。	
ICE エラー: 6FFFH 未定義のファンクションが発行されました。	

DTU8  
ユーザーズマニュアル  
SQ003096E101

---

2012 年 12 月      第 21 版発行

©2008-2012 LAPIS Semiconductor Co., Ltd.

---