

# FR ファミリ SOFTUNE™ Workbench 操作マニュアル

V6 対応



**FR ファミリ**  
**SOFTUNE™ Workbench**  
**操作マニュアル**  
V6 対応

**富士通マイクロエレクトロニクス株式会社**



# はじめに

## ■ SOFTUNE Workbench とは

この SOFTUNE Workbench は、富士通マイクロコントローラ、FR ファミリのプログラムの開発を行うためのサポートソフトウェアです。

開発マネージャ、シミュレータデバッガ、エミュレータデバッガ、モニタデバッガを一つに統合しており、効率よく開発を進めることができる統合型開発環境です。

## ■ 本書の目的と対象読者

本書は、SOFTUNE Workbench を使用される上での操作方法、開発手順などについて解説したものです。

本書は、SOFTUNE Workbench を使用して各種製品を開発される技術者の方を対象にしています。本書をご一読ください。

## ■ 商標

SOFTUNE は、富士通マイクロエレクトロニクス株式会社の商標です。

Microsoft, 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。

その他の記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

## ■ 本書の全体構成

本書は、以下に示す 4 つの章および付録で構成されています。

### 第 1 章 SOFTUNE Workbench の概要

この章では、SOFTUNE Workbench の概要について説明します。

### 第 2 章 操作

この章では、SOFTUNE Workbench を使用される上での基本的な操作方法ならびに、開発手順について説明します。

### 第 3 章 ウィンドウ

この章では、SOFTUNE Workbench のウィンドウについて詳しく説明します。

### 第 4 章 メニュー

この章では、SOFTUNE Workbench のメニュー構成およびダイアログについて詳しく説明します。

### 付録

付録では、レジスタ名一覧、モニタプログラムのダウンロード、LAN インタフェースの設定、USB インタフェースの設定、モニタデバッガターゲット ROM 部の作成、エミュレータ本体の表示、シミュレータ外部 I/F を記載しています。

- 本資料の記載内容は、予告なしに変更することがありますので、ご用命の際は営業部門にご確認ください。
- 本資料に記載された動作概要や応用回路例は、半導体デバイスの標準的な動作や使い方を示したもので、実際に使用する機器での動作を保証するものではありません。したがって、これらを使用するにあたってはお客様の責任において機器の設計を行ってください。これらの使用に起因する損害などについては、当社はその責任を負いません。
- 本資料に記載された動作概要・回路図を含む技術情報は、当社もしくは第三者の特許権、著作権等の知的財産権やその他の権利の使用権または実施権の許諾を意味するものではありません。また、これらの使用について、第三者の知的財産権やその他の権利の実施ができることの保証を行うものではありません。したがって、これらの使用に起因する第三者の知的財産権やその他の権利の侵害について、当社はその責任を負いません。
- 本資料に記載された製品は、通常の産業用、一般事務用、パーソナル用、家庭用などの一般的用途に使用されることを意図して設計・製造されています。極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、社会的に重大な影響を与えかつ直接生命・身体に対する重大な危険性を伴う用途（原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療機器、兵器システムにおけるミサイル発射制御をいう）、ならびに極めて高い信頼性が要求される用途（海底中継器、宇宙衛星をいう）に使用されるよう設計・製造されたものではありません。したがって、これらの用途にご使用をお考えのお客様は、必ず事前に営業部門までご相談ください。ご相談なく使用されたことにより発生した損害などについては、責任を負いかねますのでご了承ください。
- 半導体デバイスはある確率で故障が発生します。当社半導体デバイスが故障しても、結果的に人身事故、火災事故、社会的な損害を生じさせないよう、お客様は、装置の冗長設計、延焼対策設計、過電流防止対策設計、誤動作防止設計などの安全設計をお願いします。
- 本資料に記載された製品を輸出または提供する場合は、外国為替及び外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。





# 目次

<b>第 1 章</b>	<b>概要</b>	<b>1</b>
1.1	概要	2
1.2	SOFTUNE Workbench とは	3
1.3	SOFTUNE Workbench でのプログラム開発手順	4
<b>第 2 章</b>	<b>操作</b>	<b>5</b>
2.1	ダイアログに入力するパラメータ	6
2.1.1	データ式, アドレス式 (数値定数)	7
2.1.2	データ式, アドレス式 (シンボル, 行番号, 文字定数)	9
2.1.3	データ式, アドレス式 (レジスタ, フラグ名)	11
2.1.4	データ式, アドレス式の演算子	12
2.1.5	アドレス式の指定方法	13
2.1.6	ファイル名の指定方法	14
2.2	起動 / 終了	15
2.3	ワークスペースの作成	16
2.4	プロジェクトの登録	18
2.5	ファイル作成とプロジェクトへの登録	21
2.6	サブプロジェクトの定義	22
2.7	プロジェクト構成の作成	23
2.8	ツール設定	25
2.9	リンカオプションの設定	26
2.10	メイク / ビルド	28
2.10.1	プロジェクトのメイク / ビルド	29
2.11	デバッグ	30
2.12	デバッグのみを行う場合	31
2.13	以前のバージョンの SOFTUNE プロジェクトファイルを読み込む場合	33
2.14	プロジェクトファイルの移動	35
2.15	便利な機能	36
<b>第 3 章</b>	<b>ウィンドウ</b>	<b>37</b>
3.1	ウィンドウ構成	38
3.2	ツールバー	39
3.3	ステータスバー	41
3.4	プロジェクトウィンドウ	42
3.4.1	SRC タブ	43
3.4.2	ABS タブ	55
3.5	アウトプットウィンドウ	60
3.6	エディットウィンドウ (標準エディタ)	63
3.6.1	標準エディタ設定	67
3.7	ソースウィンドウ	72
3.8	シンボルウィンドウ	77
3.9	逆アセンブルウィンドウ	79
3.10	レジスタウィンドウ	82
3.10.1	レジスタの表示設定	84
3.11	メモリウィンドウ	86

3.12	ローカル変数ウィンドウ .....	88
3.13	ウォッチウィンドウ .....	90
3.14	トレースウィンドウ .....	93
3.15	コマンドウィンドウ .....	95
3.15.1	文字列置換設定 .....	98
3.15.2	ロギング .....	102
3.16	リアルタイムメモリウィンドウ .....	103
3.17	カバレッジウィンドウ .....	105
3.18	パフォーマンスウィンドウ .....	107
3.19	RAM チェッカウィンドウ .....	109
3.19.1	RAM チェッカの設定 .....	111
3.19.2	RAM Checker Viewer の起動 .....	116
<b>第4章</b>	<b>メニュー .....</b>	<b>119</b>
4.1	メニューの構成 ( 階層 ) .....	120
4.2	ファイルメニュー .....	123
4.2.1	新規作成 .....	124
4.2.2	開く .....	126
4.2.3	閉じる .....	128
4.2.4	ワークスペースを開く .....	129
4.2.5	ワークスペースを閉じる .....	130
4.2.6	上書き保存 .....	131
4.2.7	名前を付けて保存 .....	132
4.2.8	すべてを保存 .....	133
4.2.9	印刷 .....	134
4.2.10	最新のテキストファイル / 最新のワークスペース .....	135
4.2.11	アプリケーションの終了 .....	136
4.3	編集メニュー .....	137
4.3.1	元に戻す / やり直し .....	138
4.3.2	切り取り / コピー / 貼り付け / 削除 .....	139
4.3.3	すべて選択 .....	140
4.3.4	検索 / 置換 .....	141
4.3.5	ファイルから検索 .....	143
4.3.6	ジャンプ .....	144
4.3.7	ブックマーク .....	145
4.3.8	前のエラー / 次のエラー / 先頭のエラー / 最後のエラー .....	153
4.3.9	プロパティ .....	154
4.4	表示メニュー .....	165
4.4.1	プロジェクト / アウトプット .....	166
4.4.2	シンボル一覧 .....	167
4.4.3	逆アセンブル .....	168
4.4.4	レジスタ .....	170
4.4.5	メモリ .....	172
4.4.6	ローカル変数 .....	179
4.4.7	ウォッチ .....	180
4.4.8	トレース .....	183
4.4.9	コマンド .....	194
4.4.10	ツールバー / ステータスバー / タブ .....	195
4.4.11	フォント .....	196

4.4.12	カバレッジ .....	198
4.4.13	リアルタイムメモリ .....	200
4.4.14	パフォーマンス .....	202
4.4.15	RAM チェッカ .....	207
4.5	プロジェクト .....	208
4.5.1	アクティブプロジェクトの設定 .....	209
4.5.2	プロジェクトの追加 .....	210
4.5.3	メンバの追加 .....	214
4.5.4	ワークスペースの設定 .....	217
4.5.5	プロジェクトの設定 .....	218
4.5.6	カスタマイズビルドの設定 .....	265
4.5.7	プロジェクトの依存関係 .....	270
4.5.8	プロジェクト構成 .....	271
4.5.9	依存関係の更新 .....	275
4.5.10	コンパイル / メイク / ビルド / 中止 .....	276
4.6	デバッグ .....	277
4.6.1	実行 .....	278
4.6.2	停止 .....	279
4.6.3	MCU のリセット .....	280
4.6.4	ブレークポイント .....	281
4.6.5	イベント .....	294
4.6.6	シーケンス .....	298
4.6.7	コールスタック .....	301
4.6.8	時間測定 .....	302
4.6.9	関数コール .....	303
4.6.10	コールクリア .....	305
4.6.11	ベクタ .....	306
4.6.12	ターゲットファイルのロード .....	309
4.6.13	デバッグの開始 / デバッグの終了 .....	310
4.7	環境 .....	311
4.7.1	開発環境 .....	312
4.7.2	デバッグ環境の設定 .....	316
4.7.3	デバッガのメモリマップ .....	342
4.7.4	ツールの設定 .....	343
4.7.5	キーボードの設定 .....	346
4.7.6	エディタの設定 .....	348
4.7.7	エラージャンプの設定 .....	351
4.7.8	ツールの起動 .....	353
4.8	ウィンドウ .....	354
4.8.1	重ねて表示 / 上下に並べて表示 / 左右に並べて表示 .....	355
4.8.2	上下に分割 .....	356
4.8.3	アイコンの整列 .....	357
4.8.4	ウィンドウの更新 .....	358
4.8.5	すべてのウィンドウの更新 .....	359
4.8.6	すべて閉じる .....	360
4.9	ヘルプ .....	361
4.9.1	トピックの検索 .....	362
4.9.2	サポート情報 .....	363
4.9.3	バージョン情報 .....	364

<b>第 5 章</b>	<b>アドインモジュール</b>	<b>367</b>
5.1	カスタマイズバー	368
5.1.1	カスタマイズバーとは？	369
5.1.2	カスタマイズバーのメニュー	371
5.1.3	カスタマイズバーへの登録	372
5.1.4	警告 / エラーメッセージ	381
5.1.5	注意事項	383
5.2	FLASH ロード	384
5.2.1	FLASH ロードとは	385
5.2.2	メニュー	386
5.2.3	制限事項	396
5.2.4	エラーメッセージ	397
<b>付録</b>		<b>399</b>
付録 A	レジスタ名一覧	400
付録 B	モニタプログラムのダウンロード	401
付録 C	LAN インタフェースの設定	403
付録 D	USB インタフェースの設定	406
付録 E	モニタデバッガターゲット ROM 部の作成	407
付録 F	エミュレータ本体の表示	412
付録 G	シミュレータ外部 I/F	414
<b>索引</b>		<b>437</b>

# 第1章

---

## 概要

**SOFTUNE Workbench は、C/C++ コンパイラ、アセンブラ、リンケージキットなどの言語系ツールと、シミュレータデバッガ、エミュレータデバッガ、モニタデバッガを一つに統合し、プログラムの作成からデバッグ、さらに ROM に書き込むデータの作成までをトータルにサポートする統合開発環境です。**

### 1.1 概要

### 1.2 SOFTUNE Workbench とは

### 1.3 SOFTUNE Workbench でのプログラム開発手順

## 1.1 概要

---

**SOFTUNE Workbench で統合される開発ツールの概要を説明します。**

---

### ■ 言語系ツール

SOFTUNE Workbench は、コマンドラインから起動する C/C++ コンパイラ、アセンブラ、リンケージキットなどの言語系のツールをそのまま使用しますが、各ツールのオプション設定用のダイアログを用意しており、簡単に使用できるようにしています。

### ■ デバッガ

シミュレータデバッガ、エミュレータデバッガ、モニタデバッガを一つに統合しています。必要に応じて最適なデバッガを選択して使用することができます。

### ■ その他

別売りの REALOS コンフィグレータをインストールすることによって、複雑な設定無しに連携動作をさせることができます。

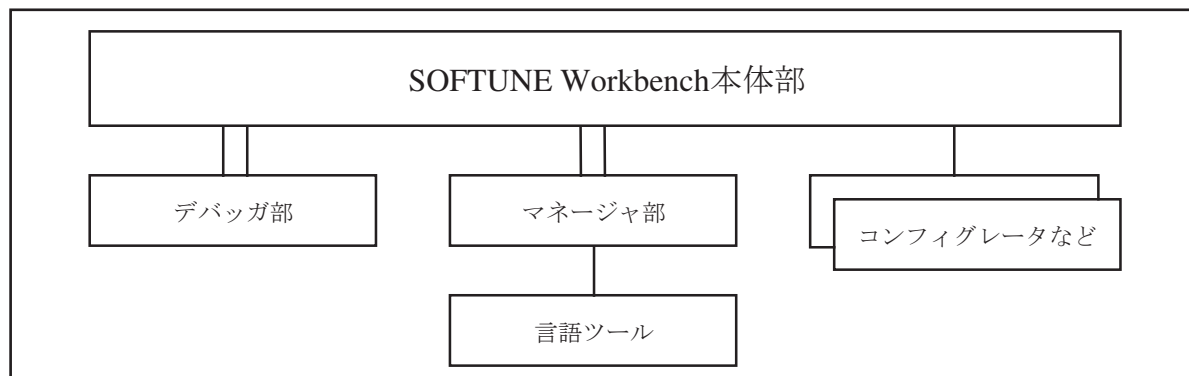
## 1.2 SOFTUNE Workbench とは

SOFTUNE Workbench の基本的な構成について説明します。

### ■ SOFTUNE Workbench の構成

SOFTUNE Workbench の基本的な構造を図 1.2-1 に示します。

図 1.2-1 SOFTUNE Workbench の基本構成図



SOFTUNE Workbench は、図示したように本体部、デバッガ部、マネージャ部の3つの部分から構成されています。

デバッガ部には、シミュレータデバッガ、エミュレータデバッガ、モニタデバッガが含まれていて、必要に応じて切り替えて使用できるようになっています。

マネージャ部は、C/C++ コンパイラ、アセンブラなどの言語系のツールの起動方法やオプション指定などを詳しく知らなくてもプログラムのコーディングやメイクができるようにしています。

コンフィグレータなどは、SOFTUNE Workbench には含まれませんが、別に購入してインストールすることにより、SOFTUNE Workbench 上で連携して動作させることができます。

SOFTUNE Workbench は、プログラムの作成～デバッグまでのすべての作業を、プロジェクトと呼ばれる単位で管理します。プロジェクトにはプログラムを構成するすべてのファイル、C/C++ コンパイラなどのツールのオプション、デバッガの環境設定などが登録されています。

## 1.3 SOFTUNE Workbench でのプログラム開発手順

---

SOFTUNE Workbench を使ったプログラムの開発手順は、次のようになります。

- 1) SOFTUNE Workbench の動作条件を設定する。
  - 2) プロジェクトを設計する。
  - 3) プログラムソースを作成し、メイク / ビルドを実行する。
  - 4) デバッグを行う。
- 

### ■ SOFTUNE Workbench の動作条件を設定する

SOFTUNE Workbench によるプログラム開発を行う場合には、まず最初に「環境」 - 「開発環境の設定」メニューにより開発環境設定ダイアログを開いて、環境変数とプロジェクトの動作設定を行います。詳しい設定方法は、「4.7.1 開発環境」の説明をご覧ください。

ここで設定した環境変数が C/C++ コンパイラなどのツールで参照されます。

### ■ プロジェクトを設計する

開発しようとしているプログラムの情報をプロジェクトに設定します。

新規に開発を始める場合は、「ファイル」 - 「新規作成」メニューで新規作成ダイアログを開き、「ワークスペース / プロジェクトファイル」を選択してください。プロジェクト新規作成のダイアログが開きます。

既にプロジェクトがある場合は、「ファイル」 - 「ワークスペースを開く」メニューで既存のプロジェクトファイルを開くことができます。なお、SOFTUNE V01/V02 のプロジェクトファイルを利用したい場合は、「2.13 以前のバージョンの SOFTUNE プロジェクトファイルを読み込む場合」の説明をご覧ください。

### ■ プログラムソースを作成し、メイク / ビルドを実行する

「ファイル」 - 「新規作成」メニューで新規作成ダイアログを開き、「テキストファイル」を選択してください。エディタが起動しますのでソースプログラムを記述してから、「ファイル」 - 「名前を付けて保存」メニューでファイルをセーブしてください。

必要なソースファイルが作成できたら、「プロジェクト」 - 「メンバの追加」メニューにより、作成したソースファイルをプロジェクトに登録してください。

ソースファイルの登録が完了したら、「プロジェクト」 - 「メイク」メニューまたは「プロジェクト」 - 「ビルド」メニューで、メイクまたはビルドしてください。

コンパイル・アセンブル時に文法エラーが発生した場合は、アウトプットウィンドウのエラー表示位置でマウスの左ボタンをダブルクリックすれば、エラーの発生した行へエラージャンプしますので、ソースを修正して、再度「プロジェクト」 - 「メイク」を実行してください。

### ■ デバッグを行う

エラーがなくなってロードモジュールファイルが作成されたら、デバッグをすることができます。



# 第2章

---

## 操作

この章では、SOFTUNE Workbench の基本的な操作方法を、以下の項目ごとに説明します。

- 2.1 ダイアログに入力するパラメータ
- 2.2 起動 / 終了
- 2.3 ワークスペースの作成
- 2.4 プロジェクトの登録
- 2.5 ファイル作成とプロジェクトへの登録
- 2.6 サブプロジェクトの定義
- 2.7 プロジェクト構成の作成
- 2.8 ツール設定
- 2.9 リンカオプションの設定
- 2.10 メイク / ビルド
- 2.11 デバッグ
- 2.12 デバッグのみを行う場合
- 2.13 以前のバージョンの SOFTUNE プロジェクトファイルを読み込む場合
- 2.14 プロジェクトファイルの移動
- 2.15 便利な機能

## 2.1 ダイアログに入力するパラメータ

---

ダイアログなどでキー入力を求められた場合、パラメータとして記述できる要素には、次の4種類があります。

- データ式
  - アドレス式
  - 識別子
  - ファイル名の指定
- 

### ■ データ式

データ式は、項と演算子から構成されます。C/C++ 言語の式に準拠しており、ほとんどの C/C++ 言語式を認識します。また、行番号やレジスタ指定などいくつかの点が拡張されています。

なお、浮動小数点を含む演算や、文字列を含む演算はサポートしていません。また、演算の途中で発生したオーバフローは無視されます。ゼロ除算は、エラーが発生します。

### ■ アドレス式

アドレス式は、データ式を拡張したものでメモリ位置を表します。データ式と同じく項と演算子から構成されます。使用できる項、演算子はデータ式と同じです。

### ■ 識別子

識別子に使用できる文字は、英字、数字、"\_" で、先頭文字は数字以外でなければなりません。英字の大文字と小文字は区別されます。

### ■ ファイル名の指定

ファイル名の指定は、Windows の規定に準拠します。

## 2.1.1 データ式, アドレス式 (数値定数)

---

データ式, アドレス式の項として数値定数があります。数値定数には, 整数と浮動小数点数があります。

---

### ■ 整数

整数を記述する際はその数値の基数を表現する規定子 (B', Q', D', H', 0x) を数値の前に記述します。規定子を省略した場合は, デフォルト基数が用いられます。デフォルト基数は, 値の入力を必要とする箇所ごとに決まっています。詳細は, 各ダイアログの説明を参照してください。

数値の表現範囲は, 0 ~ H'FFFFFFFF です。ただし, 入力する対象によっては, さらに制限されます。

マイナスの表現は, 例えば -D'1 のように行います。

#### 【記述例】

2 進定数	B'1010
8 進定数	Q'1267
10 進定数	D'1800
16 進定数	H'12AF または 0x12AF

---

#### < 注意事項 >

規定子と数値の間にブランクを入れてはいけません。

---

## ■ 浮動小数点数

浮動小数点数には次の 2 種類をサポートしています。

- 単精度浮動小数点数 (S) : float 型
- 倍精度浮動小数点数 (D) : double 型, long double 型

内部形式とサイズは, C/C++ コンパイラで扱う浮動小数点数型に準じています。

入力形式は次のとおりです。

$[F'][-]\{.dld[.d]\}\{\{SID\}[[+/-]d]\}$

d は符号無しの 10 進数です。

入力した値は最近値丸めを行います。また, 表現された値が正規化数でないときは警告メッセージが表示され, 次の値が入力されます。

- アンダフローするもの : 非正規化数として表現できるものは非正規化数に, それより小さいものは  $\pm 0$  にします。
- オーバフローするもの : 無限大にします。

浮動小数点数は次のように 16 進数で入力することもできます。

H'16進数[.{SID}]

---

### < 注意事項 >

S, D を省略した場合は, D とみなします。

---

## 2.1.2 データ式, アドレス式 (シンボル, 行番号, 文字定数)

データ式, アドレス式の項として, シンボル, 行番号, 文字定数があります。

### ■ シンボル

ソースプログラム中で用いられたシンボルをアドレスとして参照することができます。

シンボルは, C/C++ コンパイラによって生成された型の情報, そのほかの付属情報を持ちます。アセンブラによって生成された場合には, ラベル型となります。

シンボルはモジュール名と関数名とシンボル名から構成され, 次のように指定します。

`[ [モジュール名] [¥関数名] ¥] シンボル名`

C++ では, スコープ解決演算子を用いた次の記述ができます。

`[クラス名 ::] [関数名 ¥] シンボル名`

モジュール名は, ソースプログラムがアセンブリ言語で記述されている場合は ".PROGRAM" 文のオペランドに記述された名前です。また, C/C++ 言語で記述されている場合は, コンパイルするソースファイル名です。

関数名は, C/C++ 言語の関数名ですので, ソースプログラムが C/C++ 言語で記述されている場合のみ有効です。

グローバルシンボルは, "¥シンボル名" または "::シンボル名" と記述することにより区別して使用することができます。

表 2.1-1 にシンボルの記述方式を示します。

なお, シンボルに日本語文字列を使用することはできません。

表 2.1-1 シンボルの記述方式

記述	内容
シンボル	関数内ローカルシンボル    this ポインタでアクセス可能なクラスメンバ (C++ の場合) モジュール内スタティックシンボル    グローバルシンボル
¥シンボルまたは :: シンボル	グローバルシンボル
¥関数 ¥シンボル	カレントモジュールの指定関数内ローカルシンボル
モジュール ¥シンボル	指定モジュール内スタティックシンボルまたは指定モジュール内で定義されているグローバルシンボル
モジュール ¥関数 ¥シンボル	指定モジュールの指定関数内ローカルシンボル
クラス :: シンボル	クラス内シンボル (静的なもののみ有効)
クラス :: 関数 ¥シンボル	クラス関数内シンボル

### ■ 行番号

C/C++コンパイラまたはアセンブラが生成する行番号を用いて、アドレスを参照することができます。C/C++ 言語の場合は 1 行をコンパイルしたときの先頭アドレスを示します。

[ソースファイル名] \$ 行番号

アドレスを行番号で参照する場合は、行番号の先頭に \$ をつけます。行番号は、10 進数でのみ指定可能です。指定形式は以下のとおりです。ソースファイル名の拡張子が、".c" の場合は省略することができます。また、ソースファイル名が英数字以外の場合はダブルクォートで囲んで指定します。

### ■ 文字定数

文字定数は、シングルクォートで囲んだ文字の値です。文字定数には、シングルクォーテーション、¥ を含めることはできません。これらの文字を文字定数として使用する場合は、エスケープ文字を用います。エスケープ文字は、文字列で使うものと同じです。

### ■ マングル名

C++ では、関数の多重定義の実装などのためにマングル名を生成します。

C++ 対応 Workbench では、シンボル名の指定にマングル名が使用できます。

多重定義関数のようにソースプログラム上で同じ名前の関数を区別するためには、マングル名を使用してください。

たとえば、関数 `func (int i)` と `func (short s)` は、それぞれ `func__Fi` と `func__Fs` のマングル名になります。

`CALL func__Fs (10)` のコマンド実行では、`func (short s)` と定義された関数が呼び出されます。

### 2.1.3 データ式, アドレス式 (レジスタ, フラグ名)

---

データ式, アドレス式の項としてレジスタとフラグ名があります。

---

#### ■ レジスタ, フラグ名

データ式において, レジスタおよびフラグ名を指定することができます。この指定により, その時点でのレジスタの内容を表します。

レジスタ, フラグ名の前に % を入れて指定します。

使用できるレジスタ名は, MCU ごとに異なりますので, 「付録 A レジスタ名一覧」を参照してください。

## 2.1.4 データ式，アドレス式の演算子

データ式，アドレス式に使用できる演算子と，その優先順位を表 2.1-2 に示します。

### ■ 演算子

表 2.1-2 演算子の一覧

優先順位	演算子	内 容	演算子の種類
1	() [] . ->	優先順位の変更 添字表現 ストラクチャ ストラクチャポインタ	一次式
2	- & ! ~ * sizeof (型)	負符号 アドレス 論理 NOT ビット NOT メモリ間接 サイズ (バイト) 型キャスト	単項演算子
3	. * -> *	メンバへのポインタ (C++) メンバへのポインタ (C++)	二項演算子
4	* / %	乗算 除算 剰余	
5	+ -	加算 減算	
6	<< >>	左シフト 右シフト	
7	< <= > >=	より小さい より小さいか等しい より大きい より大きい等しい	二項演算子 (関係演算子)
8	== !=	等しい 等しくない	
9	&	ビット AND	二項演算子
10	^	ビット EOR	
11		ビット OR	
12	&&	論理 AND	
13		論理 OR	

#### < 注意事項 >

関係演算子は，比較結果が真の場合は H'1 に，偽の場合には H'0 になります。

なお，C/C++ 言語の条件演算子 (?:)，カンマ演算子 (,)，インクリメントおよびデクリメント演算子 (++，--) はサポートしていません。



## 2.1.5 アドレス式の指定方法

アドレス式の指定には、フルアドレス指定とアドレス範囲指定の2種類があります。

### ■ フルアドレス指定

フルアドレス指定は次の書式になっています。

32ビットアドレス指定

32 ビットアドレス指定：アドレスを指定する式です。

### ■ アドレス範囲指定

メモリの範囲を表すもので二つのアドレス指定からなり、以下の形式を持ちます。

アドレス指定 … アドレス指定（開始アドレスから終了アドレスまで）  
アドレス指定 … +オフセット（開始アドレスから開始アドレス+オフセットまで）

オフセット：アドレス指定（開始アドレス）からの相対値です。

### ■ ビットアドレス指定

ビットアドレスを表現するには、次の表記を用います。また、ビットアドレス属性のシンボルを用いることもできます。ビットアドレス指定が有効になるのは、コマンド修飾子に /BIT を指定したときです。

[アドレス指定]：ビットオフセット

ビットオフセット：ビット位置を指定する値です。

アドレス指定を省略した場合には、0 番地になります。

## 2.1.6 ファイル名の指定方法

---

ファイル名の指定方法は , Windows の規定に準拠します。

---

### ■ ファイル名の指定

[ドライブ名:] [ディレクトリのパス名] ファイル名 [. 拡張子]

ドライブ名を省略した場合は , カレントドライブが選択されます。

## 2.2 起動 / 終了

---

SOFTUNE Workbench の起動と終了のしかたについて説明します。

---

### ■ SOFTUNE Workbench の起動

SOFTUNE V6 で SOFTUNE Workbench を起動するには、[SOFTUNE V6] グループ内の [FR Family SOFTUNE Workbench] アイコンをダブルクリックします。

SOFTUNE Manager V01/02 をインストールされている方は、本プログラムを初めて起動すると、以前のバージョンで設定されていた [エディタの設定]、[ツールの設定]、[エラージャンプの設定] 情報を引き継ぐかどうかのダイアログが表示されます。引き継ぐ場合は、[はい] を、引き継がない場合は、[いいえ] を押してください。

### ■ SOFTUNE Workbench の終了

SOFTUNE Workbench を終了するには、[ファイル] メニューから [終了] を選択するか、ウィンドウの右上の × 印のボタンをクリックしてください。

コンパイル / アセンブル、メイク、ビルドまたは、ツールの実行中には SOFTUNE Workbench は終了することはできません。必ずこれらが終了するかまたは、中断してから SOFTUNE Workbench を終了してください。

## 2.3 ワークスペースの作成

---

**SOFTUNE Workbench でプロジェクトを登録するには、プロジェクトを格納するワークスペースが必要です。以下の手順で作成してください。**

---

### ■ ワークスペースの作成

SOFTUNE Workbench では、次の方法でワークスペースを作成します。

- プロジェクトの新規作成時にワークスペースを作成

プロジェクトを新規作成する際にそのプロジェクトを格納するワークスペースも一緒に作成します。

この場合、ワークスペースの名前と位置がプロジェクトと同じになります。

- プロジェクトオープン時にワークスペースを作成

[ファイル] - [ワークスペースを開く] からプロジェクトをオープンした際にそのプロジェクトを格納するワークスペースも一緒に作成します。

この場合、ワークスペースの名前と位置がプロジェクトと同じになります。ただし、既にワークスペースファイルが存在する場合は、プロジェクトファイルの代わりにそのワークスペースファイルをオープンします。

- 空のワークスペースを作成

プロジェクトが登録されていない空のワークスペースを作成します。プロジェクトは別途登録する必要があります。

この場合、ワークスペースの名前と位置が収容するどのプロジェクトとも重複しないようにプロジェクトを登録することが可能です。

### ■ プロジェクトの新規作成時にワークスペースを作成するには...

- 1) [ファイル] - [新規作成] を選択

- [新規作成] ダイアログが表示されたら、[ファイルの種類] で "ワークスペース / プロジェクトファイル" を選択して、[OK] ボタンをクリックしてください。

- 2) プロジェクトタブを選択

- プロジェクトの [新規作成] ダイアログが表示されたら、[プロジェクト] タブを選択してください。

- 3) [新規にワークスペースを作成] を選択

- [新規にワークスペースを作成] チェックボタンをクリックしてプロジェクトを作成してください。後は、通常のプロジェクト新規作成時と同様に作成してください。

**■ プロジェクトオープン時にワークスペースを作成するには...**

- 1) [ファイル] - [ワークスペースを開く] を選択
- 2) ファイルの種類でプロジェクトファイルを選択
  - [ワークスペースを開く] ダイアログが表示されたら, [ファイルの種類] でプロジェクトファイルを選択してください。
- 3) オープンするプロジェクトを選択
  - オープンしたいプロジェクトファイルを選択してください。
- 4) [開く] ボタンをクリック

**■ 空のワークスペースを作成するには...**

- 1) [ファイル] - [新規作成] を選択
  - [新規作成] ダイアログが表示されたら, [ファイルの種類] で "ワークスペース / プロジェクトファイル" を選択して, [OK] ボタンをクリックしてください。
- 2) ワークスペースタブを選択
  - プロジェクトの [新規作成] ダイアログが表示されたら, [ワークスペース] タブを選択してください。
- 3) ワークスペースタイプを選択
  - ワークスペースタイプとして [ブランクワークスペース] を選択してください。
- 4) ワークスペース名を入力
  - ワークスペース名を入力してください。この名前はワークスペースファイル名として使用されます。また, デフォルトでワークスペースディレクトリにも使用されます (ワークスペースディレクトリは変更可能です)。
- 5) [OK] ボタンをクリック

## 2.4 プロジェクトの登録

---

**SOFTUNE Workbench でソフトウェアの開発やデバッグを行う場合には、プロジェクトが必要です。以下の手順でワークスペースに登録してください。**

---

### ■ プロジェクトの登録

SOFTUNE Workbench では、次の方法でワークスペースにプロジェクトを登録します。新規に登録したプロジェクトがワークスペースのアクティブプロジェクトになります。

- 現在開いているワークスペースに新規プロジェクトを登録

現在開いているワークスペースにプロジェクトを新規作成して登録します。

- 既存のプロジェクトをワークスペースに登録

現在開いているワークスペースに既に作成しているプロジェクトを登録します。

### ■ 現在開いているワークスペースに新規プロジェクトを登録するには...

- 1) プロジェクトを登録するワークスペースを開く
  - プロジェクトを登録したいワークスペースを開いてください。
- 2) [プロジェクト]-[プロジェクトの追加]-[新規作成]メニューを選択
- 3) 現在のワークスペースへ追加を選択
  - [現在のワークスペースへ追加]チェックボタンをクリックしてください。
- 4) プロジェクトタイプを選択
  - プロジェクトで最終的に作成するファイルの種類[プロジェクトタイプ]を選択してください。選択可能なプロジェクトタイプと説明を表 2.4-1 に示します。
- 5) チップ種別とターゲット MCU を選択
  - 選択可能な値がドロップダウンリストに表示されますので、その中から使用するチップとターゲット MCU を選択してください。
- 6) プロジェクト名を入力
  - プロジェクト名を入力してください。この名前はプロジェクトファイル名として使用されます。また、デフォルトでターゲットファイル名および、プロジェクトディレクトリにも使用されます(ターゲットファイル名、プロジェクトディレクトリは変更可能です)。
- 7) プロジェクト間の依存関係を設定
  - プロジェクトを別のプロジェクトのサブプロジェクトとして定義したい場合は、[依存関係]チェックボックスをオンにし、[依存関係]ボックスの一覧からプロジェクト名を選択してください。

## 8) [OK] ボタンをクリック

- プロジェクトディレクトリのサブディレクトリとして "Debug", "Debug ¥ABS", "Debug ¥OBJ", "Debug ¥LST", "Debug ¥OPT" ディレクトリが作成されます。
  - Debug : プロジェクト構成ごとの情報を格納するディレクトリ。プロジェクトを新規作成した場合のデフォルト構成名は "Debug" になります。
  - ABS : ターゲットファイルを格納するディレクトリです。
  - OBJ : オブジェクトファイルを格納するディレクトリです。
  - LST : リストファイルを格納するディレクトリです。
  - OPT : 言語ツールを起動するオプションファイルを一時的に格納するディレクトリです。

プロジェクトタイプに REALOS を選択した場合は、コンフィグレータのセットアップウィザードが開きます。この詳細は、SOFTUNE REALOS に添付のマニュアルを参照してください。

表 2.4-1 プロジェクトタイプ一覧

プロジェクトタイプ	説 明
絶対形式 (ABS)	通常のプログラムファイルを作成します。
相対形式 (REL)	相対形式ファイルを作成します。
ライブラリファイル	ライブラリファイルを作成します。
REALOS (ABS)	リアルタイム OS を使用したプログラムを作成します。

## ■ 既存のプロジェクトをワークスペースに登録するには...

- 1) プロジェクトに登録するワークスペースを開く
  - プロジェクトに登録したいワークスペースを開いてください。
- 2) [プロジェクト]-[プロジェクトの追加]-[既存のプロジェクト]メニューを選択
- 3) 登録するプロジェクトを選択
  - [プロジェクトの追加]ダイアログで登録したいプロジェクトを選択してください。
- 4) プロジェクト間の依存関係を設定
  - プロジェクトを別のプロジェクトのサブプロジェクトとして定義したい場合は、[依存関係]チェックボックスをオンにし、[依存関係]ボックスの一覧からプロジェクト名を選択してください。
- 5) [開く]ボタンをクリック

## ■ アクティブプロジェクトを設定するには...

アクティブプロジェクトとは、メニューの [メイク], [ビルド], [コンパイル/アセンブル], [デバッグの開始], [依存関係の更新] の実行対象となるプロジェクトです。[メイク], [ビルド], [コンパイル/アセンブル], [依存関係の更新] は、アクティブプロジェクトのサブプロジェクトにも影響します。

設定するには、[プロジェクト] - [アクティブプロジェクトの設定] を選択します。選択するとサブメニューが表示されるので、その中からアクティブプロジェクトにしたいプロジェクト名を選択してください。

## ■ ワークスペースに登録したプロジェクトを削除するには...

プロジェクトウィンドウの SRC タブで削除したいプロジェクトを選択します。ショートカットメニューの [プロジェクトの削除] を選択してください。

指定されたプロジェクトはワークスペースから削除されますが、プロジェクトファイルそのものの削除は行いません。

削除したプロジェクトが、ワークスペース内のプロジェクトのサブプロジェクトになっている場合は、プロジェクト間の依存関係も削除されます。

---

### < 注意事項 >

SOFTUNE REALOS がインストールされていないと、プロジェクトタイプに REALOS は表示されません。

---



## 2.5 ファイル作成とプロジェクトへの登録

---

SOFTUNE Workbench で新規にソースファイルを作成し、プロジェクトに登録するには以下のようにします。

---

### ■ 新規ソースファイルの作成方法

- 1) [ファイル] - [新規作成] を選択
  - [新規作成] ダイアログが表示されたら、[ファイルの種類] で "テキストファイル" を選択して、[OK] ボタンをクリックしてください。
- 2) [ファイル] - [名前を付けて保存] を選択
  - [名前を付けて保存] ダイアログが表示されたら、[ファイルの種類] で "テキストファイル" を選択して、[OK] ボタンをクリックしてください。

保存するディレクトリと名前を指定するためのファイルダイアログが開きます。保存するディレクトリを選択して、ファイル名を指定して、[保存] ボタンをクリックしてください。

### ■ 作成したファイルをプロジェクトに登録するには...

[プロジェクト] - [メンバの追加] - [ファイル] メニューを選択してください。メンバ登録するファイルを選択するためにファイルダイアログが表示されます。作成したソースファイルを選択し、プロジェクトウィンドウの SRC タブに挿入するフォルダを選択して、[開く] ボタンをクリックしてください。プロジェクトに登録され、プロジェクトウィンドウの SRC タブの指定したフォルダにファイル名が表示されます。

### ■ 作成したファイルをディレクトリごとプロジェクトに登録するには...

[プロジェクト] - [メンバの追加] - [ディレクトリ] メニューを選択してください。メンバ登録するフォルダを選択するために [メンバの追加 - ディレクトリ] ダイアログが表示されます。作成したソースファイルがあるディレクトリを選択し、プロジェクトウィンドウの SRC タブに挿入するフォルダを選択して、[OK] ボタンをクリックしてください。プロジェクトに登録され、指定したディレクトリ以下のファイル、フォルダがプロジェクトウィンドウの SRC タブの指定したフォルダに表示されます。

登録するファイルの種類は、ダイアログの [ファイルの種類] を設定することで制限することができます。

### ■ プロジェクトに登録したファイルを削除するには...

プロジェクトウィンドウの SRC タブで削除したいファイルを選択します ( 複数のファイルを選択することも可能です )。ショートカットメニューの「削除」を選択してください。指定されたファイルをプロジェクトメンバから削除しますが、ファイルそのものの削除は行いません。「Dependencies」カテゴリと「Debug」カテゴリのファイルは、ユーザが削除することはできません。

## 2.6 サブプロジェクトの定義

---

ここでは、サブプロジェクトの定義の方法について説明します。

---

### ■ サブプロジェクトの定義

サブプロジェクトとは、ほかのプロジェクトに依存しているプロジェクトです。

SOFTUNE Workbench では、次の方法でサブプロジェクトを定義します。

#### ● プロジェクトを登録する際にサブプロジェクトとして定義

プロジェクトを新規作成する際にプロジェクトを別のプロジェクトのサブプロジェクトとして定義します。設定方法については、「2.4 プロジェクトの登録」を参照してください。

#### ● 既存のプロジェクト間でサブプロジェクトを定義

ワークスペース内のプロジェクト間でサブプロジェクトを定義します。

親プロジェクトのサブプロジェクトに別のプロジェクトをサブプロジェクトとして定義できますが、親プロジェクト自身がサブプロジェクトになるような再帰的な定義はできません。

### ■ 既存のプロジェクト間でサブプロジェクトを定義するには...

- 1) [プロジェクト] - [プロジェクトの依存関係] を選択
- 2) サブプロジェクトを定義したい親プロジェクトを選択
  - [プロジェクトの依存関係] ダイアログが表示されたら、サブプロジェクトを定義したい親プロジェクトのプロジェクト名を [プロジェクト名] ボックスから選択してください。
- 3) サブプロジェクトにしたいプロジェクトを選択
  - [依存するプロジェクト] リストに表示されたプロジェクトから、サブプロジェクトにしたいプロジェクトをチェックします。
- 4) [OK] ボタンをクリック

### ■ プロジェクトからサブプロジェクトを削除するには...

- 1) [プロジェクト] - [プロジェクトの依存関係] を選択
- 2) サブプロジェクトを削除したい親プロジェクトを選択
  - [プロジェクトの依存関係] ダイアログが表示されたら、サブプロジェクトを削除したい親プロジェクトのプロジェクト名を [プロジェクト名] ボックスから選択してください。
- 3) 削除したいサブプロジェクトを選択
  - [依存するプロジェクト] リストで削除したいサブプロジェクトのチェックをはずします。
- 4) [OK] ボタンをクリック

## 2.7 プロジェクト構成の作成

---

ここでは、プロジェクト構成の作成方法について説明します。

---

### ■ プロジェクト構成の作成

プロジェクト構成とは、ターゲットファイルの特性を指定する一連の設定のことです。プロジェクト構成を新規に作成することで、プロジェクト内に複数のツール設定を保持することが可能となります。

プロジェクトの新規作成時には、「Debug」というプロジェクト構成名がデフォルトで作成されます。

SOFTUNE Workbench では、次の方法でプロジェクトにプロジェクト構成を作成します。

#### ● 既存のプロジェクト構成の設定に基づき作成

選択した既存のプロジェクト構成の設定に基づき、初期構成が作成されます。新規プロジェクト構成では、必ず元の構成と同じメンバが使われます。

### ■ 既存のプロジェクト構成の設定に基づき作成するには...

- 1) [プロジェクト] - [プロジェクト構成] - [追加と削除] を選択
- 2) 構成を追加するプロジェクトを選択
  - [プロジェクトの追加と削除] ダイアログが表示されたら、プロジェクト構成を追加するプロジェクトを選択してください。
- 3) [追加] ボタンをクリック
  - [追加] ボタンをクリックしてください。[追加] ボタンをクリックすると、[プロジェクト構成の追加] ダイアログが表示されます。
- 4) プロジェクト構成名を入力
  - 新規構成の一意な名前を入力してください。使用可能な文字は "a ~ z", "A ~ Z", "0 ~ 9", "\_" です。
- 5) 設定をコピーするプロジェクト構成の選択
  - 追加するプロジェクト構成の初期設定となる構成を選択します。選択したプロジェクト構成の設定 ( ツールのオプション、ファイル構成やビルドするサブプロジェクト構成など ) がそのままコピーされます。
- 6) [OK] ボタンをクリック
  - [プロジェクト構成の追加] ダイアログで [OK] ボタンをクリックし、[プロジェクト構成の追加と削除] ダイアログで [OK] ボタンをクリックしてください。

## ■ アクティブなプロジェクト構成を設定するには...

アクティブなプロジェクト構成とは、デフォルトで[メイク],[ビルド],[コンパイル/アセンブル],[デバッグの開始],[依存関係の更新]されるプロジェクト構成のことです。

- 1) [プロジェクト] - [プロジェクト構成] - [追加と削除]を選択
- 2) アクティブなプロジェクト構成にしたい構成を選択
  - [プロジェクトの追加と削除]ダイアログが表示されたら、アクティブな構成にしたい構成名を選択してください。
- 3) [アクティブ]ボタンをクリック
  - [アクティブ]ボタンをクリックしてください。クリックすると、指定した構成とそのプロジェクトがアクティブになります。
- 4) [OK] ボタンをクリック

## ■ プロジェクトから特定の構成を削除するには...

- 1) [プロジェクト] - [プロジェクト構成] - [追加と削除]を選択
- 2) プロジェクトから削除したい構成を選択
  - [プロジェクトの追加と削除]ダイアログが表示されたら、削除したい構成名を選択してください。
- 3) [削除] ボタンをクリック
  - [削除] ボタンをクリックしてください。クリックすると、指定した構成が削除されます。プロジェクト内の構成をすべて削除すると、そのプロジェクト自体も削除されます。
- 4) [OK] ボタンをクリック

## 2.8 ツール設定

---

SOFTUNE Workbench でメイク / ビルドするときには、コンパイラ / アセンブラ / リンカなどのツールに適切なオプションを指定する必要があります。これらのオプションは以下のように設定してください。

---

### ■ [プロジェクト] - [プロジェクトの設定] を選択してください

[プロジェクトの設定] ダイアログが表示されます。[プロジェクトの設定] ダイアログボックスで選択したオプションは、複数のプロジェクトに適用できます。適用対象のプロジェクト構成を限定することもできます。たとえば、A 構成と B 構成の設定内容を変えることができます。現在のすべてのプロジェクト構成に同じ設定内容を指定することもできます。

[設定の対象] ボックスでプロジェクト構成を、ツリービューで設定するプロジェクトを指定して、設定するツールのタブを選択してください。

コンパイラ、アセンブラ、リンカライブラリアンを選択した場合、各ツール設定のタブダイアログ最上部でカテゴリを選択するようになっています。ドロップダウンリストを表示してカテゴリを選択してください。カテゴリを選択すると、タブダイアログ内の表示が変わり、それぞれのカテゴリに属するオプションの設定ができるようになります。

リストファイルを出力したり、特定の条件下でのメイク / ビルドを行う必要がある場合以外は、ほとんどの場合、コンパイラおよびアセンブラのオプション設定は不要です。リンカオプションのみ設定してください。リンカオプションの設定は、「2.9 リンカオプションの設定」を参照してください。

### ■ [OK] ボタンをクリックすると設定完了です

必要なすべてのツールのオプション設定が完了したら、[OK] ボタンをクリックしてください。設定したすべてのオプションがプロジェクトに登録されて、メイク / ビルド時に有効になります。

[キャンセル] をクリックすると、今までに行ったすべての設定が解除されます。

---

#### < 注意事項 >

ツールオプションの設定の途中で [更新] ボタンをクリックした場合は、それ以前に設定したオプションに戻すことはできません。

---

## 2.9 リンカオプションの設定

---

**SOFTUNE Workbench でプログラムを作成する場合には、リンカオプションでメモリの配置を確認してください。**

---

### ■ リンカオプションの自動設定

SOFTUNE Workbench では、プロジェクトの新規作成時に選択された MCU の情報を基に、以下のリンカオプションを自動設定します。

- メモリ領域オプションに、MCU の内蔵 ROM/RAM アドレスを設定  
内蔵 ROM の領域名は "\_INROMxx", 内蔵 RAM の領域名は "\_INRAMxx" で出力されます。(xx は 01 から始まる連続した数字)
- 自動配置モードを、モード 2 (リンカによる最適な自動配置) に設定

シングルチップモード以外でプログラムを作成される場合や、セクションの配置をカスタマイズしたい場合などには、リンカオプションの [ 配置 / 結合 ] オプションを設定してください。

### ■ リンカオプションの設定

SOFTUNE Workbench では、メモリマップの指定はメモリ領域へのセクションの配置を基本に考えています。したがって、まずメモリの領域を設定して、それから、そこに配置するセクションの設定を行ってください。

### ■ メモリ領域の設定

ROM/RAM 領域名、スタートアドレス、エンドアドレスを入力し、領域属性 (ROM または RAM) を選択して [ 追加 ] ボタンをクリックしてください。ROM/RAM 領域リストにこの設定が表示されます。ROM/RAM 領域名には、ほかの領域名と衝突しないユニークな名前を付けてください。設定可能な領域の数に制限はありません。

自動配置 (モード 2) が指定された場合、リンカは配置指定の行われていないセクションを ROM/RAM 領域へ自動的に配置します。この時リンカは配置可能な ROM/RAM 領域の検索を [ROM/RAM 領域リスト] の上から順に行いますので [ 上へ ], [ 下へ ] ボタンをクリックして適切な順番に変更してください。

### ■ セクションの設定

ROM/RAM 領域リストに表示されている領域を選択して、[ セクションの設定 ] ボタンをクリックすると、[ セクションの設定 ] ダイアログが開いて、選択した領域に配置するセクションの設定ができるようになります。領域を選択するときは、その領域のスタートアドレスをクリックしてください。

[ セクションの設定 ] ダイアログが開いたら、その領域に配置するセクション名を、配置する順に記述してください。セクション名は一つずつ記述します。記述したら、[ 設定 ] ボタンをクリックして、セクション名リストに登録してください。

メイク/ビルドを実行すると、セクション名リストの上の方に登録されているセクションから順に配置されます。

一つの領域に配置するセクションの設定がすべておわったら、[OK] ボタンをクリックして、リンカオプションの設定ダイアログに戻ります。別の領域の設定も同様に行ってください。

---

参考：

いくつかの領域に対するセクション配置の設定を行う場合には、[セクションの設定] ダイアログの最上部の、ROM/RAM 領域名の表示を変更することによって、その都度リンクオプションの設定ダイアログに戻らなくても、それらのすべての領域のセクション配置の設定を連続的に行うことが可能です。

---

## 2.10 メイク / ビルド

---

**SOFTUNE Workbench でプログラムを作成する方法は、メイクとビルドの 2 とおりあります。**

---

### ■ メイク

修正のあったソースファイルのみをコンパイル / アセンブルして、その後すべてのオブジェクトとライブラリを結合して目的のプログラムを生成します。SOFTUNE Workbench は、プロジェクトウィンドウの SRC タブの「Dependencies」カテゴリに登録されているインクルードファイルについて、その依存関係を認識してコンパイル / アセンブルをします。

メイクは、[プロジェクト] - [メイク] で実行します。

### ■ ビルド

修正のあったソースファイルだけでなく、プロジェクトに登録されているすべてのソースファイルをコンパイル / アセンブルして、その後すべてのオブジェクトとライブラリを結合してターゲットファイルを生成します。

ビルドは、[プロジェクト] - [ビルド] で実行します。

### ■ 中止

メイク、ビルド、コンパイル / アセンブル中に処理を強制的に中断する機能です。

メイク、ビルド、コンパイル / アセンブル中に、[プロジェクト] - [中止] で実行します。



## 2.10.1 プロジェクトのメイク/ビルド

---

SOFTUNE Workbench では、プロジェクトの構成ごとにメイク/ビルドを行うことができます。

---

### ■ プロジェクトのメイク/ビルド

メニューの[メイク],[ビルド]は、アクティブプロジェクトのアクティブな構成に対して行われます。サブプロジェクトを定義している場合は、サブプロジェクトを先にメイク/ビルドします。

アクティブプロジェクトとアクティブな構成を変更する方法については、「2.4 プロジェクトの登録」、「2.7 プロジェクト構成の作成」を参照してください。

### ■ 指定プロジェクトをメイク/ビルドするには...

プロジェクトウィンドウのSRC タブでメイク/ビルドを行いたいプロジェクトを選択します。ショートカットメニューの[メイク]または[ビルド]を選択してください。選択すると、指定したプロジェクトのアクティブな構成でメイク/ビルドをおこないます。サブプロジェクトを定義している場合は、サブプロジェクトを先にメイク/ビルドします。

### ■ メイク/ビルド時のサブプロジェクトの構成を変更するには...

- 1) [プロジェクト]-[プロジェクト構成]-[ビルド時の構成]メニューを選択
- 2) 親プロジェクトと構成を選択
  - [ビルド時の構成の設定]ダイアログが表示されたら、[プロジェクト]ボックスから設定を行いたいプロジェクトとその構成を選択してください。選択すると、サブプロジェクトのメイク/ビルドする構成が表示されます。
- 3) サブプロジェクトの構成を選択
  - [メイク/ビルド時のサブプロジェクトの構成]からメイク/ビルドを行いたい構成を選択してください。
- 4) [OK] ボタンをクリック
  - メイク、ビルド、コンパイル/アセンブル中に処理を強制的に中断する機能です。
  - メイク、ビルド、コンパイル/アセンブル中[プロジェクト]-[中止]メニューで実行します。

## 2.11 デバッグ

---

メイク/ビルドが正常に終了し、アブソリュートファイルが作成された場合は、デバッグセッションに移行して、直ちにそのファイルをデバッグすることができます。

---

### ■ デバッグセッションへの移行

SOFTUNE Workbench でデバッグを行うには、デバッグセッションに移行してください。デバッグセッションへの移行は、[ デバッグ ] メニューから [ デバッグの開始 ] を選択してください。

- プロジェクト作成後、はじめてデバッグを行う場合

デバッグのセットアップウィザードがスタートします。ここで、デバッグの種類やその他の設定を行ってください(「4.7.2.5 セットアップウィザード」参照)。

- プロジェクト作成後 2 回目以降のデバッグの場合

既に設定されているデバッグのモードで起動します。デバッグの種類などを変更したい場合は、一度デバッグを終了([ デバッグ ]-[ デバッグの終了 ] メニューを選択)してから、[ プロジェクトの設定 ] で変更してください。

### ■ モニタプログラムの自動ダウンロード

MB2198 エミュレータをご使用の場合、デバッグセッションへの移行時にエミュレータの種類およびダウンロードされているモニタプログラムのバージョンなどからデバッグ環境を判定し、自動的に最適なモニタプログラムをダウンロードします。(付録 B モニタプログラムのダウンロードを参照してください。)

### ■ ターゲットプログラムのロード

デバッグセッションに入ったら、はじめに、[ デバッグ ] メニューから [ ターゲットプログラムのロード ] を選択して、ターゲットプログラムをロードしてください。これで、作成したプログラムがデバッグにロードされ、デバッグを行う準備がすべて整います。

### ■ デバッグの操作

デバッグの操作方法については、本書の「第 3 章 ウィンドウ」、「第 4 章 メニュー」を参照してください。

デバッグのコマンドについての説明およびデバッグの出力するエラーメッセージについては、『SOFTUNE Workbench コマンドリファレンスマニュアル』を参照してください。

デバッグの機能(MCU 共通機能、MCU チップ依存機能)については、『SOFTUNE Workbench 機能説明書』を参照してください。

## 2.12 デバッグのみを行う場合

**SOFTUNE Workbench を、従来のデバッガと同じように使用することもできます。**

### ■ プロジェクトの作成

SOFTUNE Workbench ではプロジェクトがすべて作業の基本になります。デバッグ時も例外ではありません。したがって、デバッグのみを行う場合にも、デバッグ用のプロジェクトを作成する必要があります。

まず、以下の手順でデバッグ用のプロジェクトとプロジェクトを格納するワークスペースを作成してください。

- 1) [ファイル]メニューから[新規作成]を選択してください。
- 2) ファイルを開くダイアログで[ワークスペース/プロジェクトファイル]を選択してください。
  - 新規作成ダイアログでは、絶対形式 (ABS) を選択してください。
  - プロジェクト名を指定してください。
  - プロジェクトディレクトリを指定してください。
  - ターゲット MCU 名、チップ種別を選択してください。

### ■ ワークスペースの設定

ワークスペースに登録するプロジェクトに共通の設定を行います。

- 1) [プロジェクト]メニューの[ワークスペースの設定]を選択してください。
- 2) ワークスペースの設定ダイアログが開いたら、次の設定を行ってください。
  - ワークスペースオープン時のデバッグ：デバッグを開始する
  - セットアップ情報の保存：保存する

### ■ デバッガに関する設定

- 1) [プロジェクト]メニューの[プロジェクトの設定]を選択してください。
- 2) [プロジェクトの設定]ダイアログが開いたら[デバッグ]タブを開いて、[セットアップ]カテゴリを選択してください。
- 3) セットアップ名を設定してください。

デフォルトのセットアップ名として、[セットアップ名リスト]にプロジェクト名が設定されていて、同じ名前が[有効なセットアップ名]にも設定されています。通常は使用するデバッガの種類によってセットアップ名を区別しますが、デバッガを一種類のみしか使用しないのであればデフォルト名のままでも構いません。

デフォルト名のまま使用の場合は、[セットアップ名リスト]に既に設定されているデフォルト名を選択して、[設定変更]ボタンをクリックしてください。

別の名前を使用したい場合は、[セットアップ名]を指定して、[設定追加]ボタンをクリックしてください。

### ■ セットアップウィザード

[設定追加]または[設定変更]ボタンをクリックすると、デバッガのセットアップウィザードが起動します。後はその指示にしたがって設定してください(セットアップウィ

ザードでのセットアップの方法については、「4.7.2.5 セットアップウィザード」を参照してください)。

セットアップウィザードでのすべての設定が完了したら、最後に「完了」ボタンをクリックしてください。

「プロジェクトの設定」ダイアログに戻ったら「OK」ボタンをクリックしてください。

ここまでの手順が終了したら、プロジェクトを保存して一度閉じてください。

### ■ デバッグ開始

上記のプロジェクトの作成～セットアップウィザードまでの手順が終了している場合は、そのプロジェクトを開くことにより自動的にデバッグセッションに移行し、すぐにデバッグが開始できる状態になります。

「ファイル」メニューの「開く」で、ロードモジュールファイルを指定して、ターゲットプログラムをロードします。

## 2.13 以前のバージョンの SOFTUNE プロジェクトファイルを読み込む場合

以前のバージョンの SOFTUNE で作成されたプロジェクトファイルを読み込むことができます。

### ■ 手順

#### ● SOFTUNE Workbench V3/V5 のプロジェクトの場合

- 1) [ファイル]メニューから[ワークスペースファイルを開く]を選択してください。
- 2) ファイルの種類で「プロジェクトファイル」を選択して、以前のバージョンの SOFTUNE Workbench で作成されたプロジェクトファイルを指定してください。指定されたプロジェクトファイルが以前のバージョンの SOFTUNE Workbench で作成されたものであった場合は、ワークスペースプロジェクト形式ファイルにコンバートするかどうかを問い合わせるダイアログが表示されます。

[はい] : プロジェクトファイルをコンバートしてワークスペースプロジェクト形式で開きます。

[いいえ] : プロジェクトファイルのコンバートを行わず、旧プロジェクト形式で開きます。この場合、一部の機能が使用できません。旧プロジェクトで使用可能な機能については、『SOFTUNE Workbench 機能説明書』の「1.2 プロジェクト管理機能」を参照してください。

[キャンセル] : プロジェクトファイルのオープンを中止します。

#### ● SOFTUNE Manager V01/V02 のバージョン紹介

- 1) [ファイル]メニューから[ワークスペースファイルを開く]を選択してください。
- 2) ファイルの種類で「プロジェクトファイル」を選択して、SOFTUNE Manager で作成されたプロジェクトファイルを指定してください。指定されたプロジェクトファイルが SOFTUNE Manager で作成されたものであった場合は、ワークスペースプロジェクト形式ファイルにコンバートするかどうかを問い合わせるダイアログが表示されます。
- 3) [OK] ボタンをクリックしてコンバートを開始してください。[キャンセル]をクリックした場合は、プロジェクトファイルのオープンを中止します。
- 4) プロジェクト新規作成のウィンドウが開いたら、チップ種別とターゲット MCU を設定して、[OK] ボタンをクリックしてください。
- 5) 完了するとコンバート終了を告げるダイアログが表示されます。[OK] ボタンをクリックしてダイアログを閉じてください。

## ■ バックアップファイル

SOFTUNE Workbench では、プロジェクトファイルのワークスペースプロジェクト形式にコンバートした場合、バックアップを自動的に作成します。バックアップファイルの拡張子は、プロジェクトファイル形式により異なります。また、バックアップファイルの拡張子により、バックアップされたプロジェクトファイルをオープンする方法が異なります。

### ● SOFTUNE Workbench V3/V5

旧プロジェクトファイル (.prj)	.p03
旧オプションデータファイル (.dat)	.d03

### ● SOFTUNE Workbench V01/V02

旧プロジェクトファイル (.prj)	.V01
--------------------	------

SOFTUNE Workbench V03/V05 (-p03) の場合

- バックアップのプロジェクトファイル (.p03) とオプションデータファイル (.d03) の拡張子をそれぞれ ".prj", ".dat" に変更して開く

SOFTUNE Manager V01/V02 (.v01) の場合

- バックアップのプロジェクトファイル (.v01) の拡張子を ".prj" に変更して開く

---

#### < 注意事項 >

- SOFTUNE Manager V01/V02 のプロジェクトの場合は、ツールオプションは引き継がれません。読み込み終了後再設定してください。
  - SOFTUNE Manager V01/V02 の [ 環境変数の設定 ] で設定されていた、[ ユーザインクルードディレクトリ ] や [ ライブラリ検索パス ] は、C/C++ コンパイラ、アセンブラの [ インクルードパス, ] およびリンカの [ ライブラリ検索パス ] オプションとして必ず再設定してください。
  - 指定したプロジェクトファイルと同じディレクトリに同名のワークスペースファイルが存在する場合は、プロジェクトファイルの代わりにワークスペースファイルを開くため、プロジェクトファイルの変換を行いません。
-

## 2.14 プロジェクトファイルの移動

---

ここでは、プロジェクトファイルをほかのディレクトリまたはパソコンへ移行する方法について説明します。

---

### ■ プロジェクトファイルの移行方法

- 1) プロジェクトファイルに登録されているメンバのパスをプロジェクトからの相対パスにしてください。
  - SOFTUNE Workbench では、通常プロジェクトファイルと同じドライブに存在するファイルは相対パスで登録しています。相対パスで登録されているか確認するには、プロジェクトウィンドウのSRCタブのファイルのプロパティで確認してください。プロパティについては、「4.3.9 プロパティ」を参照してください。
- 2) ターゲットディレクトリ、オブジェクトファイルディレクトリ、リストファイルディレクトリをプロジェクトファイルからの相対パスにしてください。
  - SOFTUNE Workbench では、プロジェクトの新規作成時に出力ディレクトリをプロジェクトファイルの相対パスに設定しています。変更、確認するには、「プロジェクトの設定」ダイアログを表示してください。詳しくは、「4.5.5 プロジェクトの設定」を参照してください。
- 3) ツールオプションで指定されているインクルードパス、ライブラリパスなどのディレクトリを相対パスもしくはマクロ記述にしてください。
  - マクロ記述については『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で利用できるマクロ記述」を、ツールオプションについては「4.5.5 プロジェクトの設定」を参照してください。
- 4) 1) ~ 3) の設定が終了したらプロジェクトを保存してください。
- 5) 1) ~ 3) のディレクトリ構造を変更せずにプロジェクトを移動してください。

## 2.15 便利な機能

---

**SOFTUNE Workbench を使用するにあたり、便利な機能を説明します。**

---

### ■ ウィンドウのタブ表示

SOFTUNE Workbench では表示しているウィンドウをタブ表示します。

複数のウィンドウを表示した場合、目的のウィンドウを素早く見つけるのに便利です。

### ■ ウィンドウ間のドラッグアンドドロップ

SOFTUNE Workbench ではデバッガの各ウィンドウ間でのドラッグアンドドロップ機能をサポートしています。現在ドラッグアンドドロップをサポートしているウィンドウは下記のとおりです。ドラッグアンドドロップによる動作に関しては、各ウィンドウの説明を参照してください。

- プロジェクトウィンドウ
- ソースウィンドウ
- メモリウィンドウ
- 逆アセンブルウィンドウ
- ウォッチウィンドウ
- カバレッジウィンドウ
- リアルタイムメモリウィンドウ
- RAM チェッカウィンドウ

### ■ 設定漏れ確認

SOFTUNE Workbench には様々な機能を設定するダイアログがあります。各ダイアログでは設定項目を入力後、[ 設定 ] ボタンを押さないと設定されません。[ 設定 ] ボタンを押さずにダイアログを閉じようとした場合は、設定確認のダイアログが表示され、[ 設定 ] ボタンの押し忘れによる設定漏れを防ぎます。



# 第3章

---

## ウィンドウ

この章では、SOFTUNE Workbench のウィンドウについて説明します。

- 3.1 ウィンドウ構成
- 3.2 ツールバー
- 3.3 ステータスバー
- 3.4 プロジェクトウィンドウ
- 3.5 アウトプットウィンドウ
- 3.6 エディットウィンドウ (標準エディタ)
- 3.7 ソースウィンドウ
- 3.8 シンボルウィンドウ
- 3.9 逆アセンブルウィンドウ
- 3.10 レジスタウィンドウ
- 3.11 メモリウィンドウ
- 3.12 ローカル変数ウィンドウ
- 3.13 ウォッチウィンドウ
- 3.14 トレースウィンドウ
- 3.15 コマンドウィンドウ
- 3.16 リアルタイムメモリウィンドウ
- 3.17 カバレッジウィンドウ
- 3.18 パフォーマンスウィンドウ
- 3.19 RAM チェッカウィンドウ

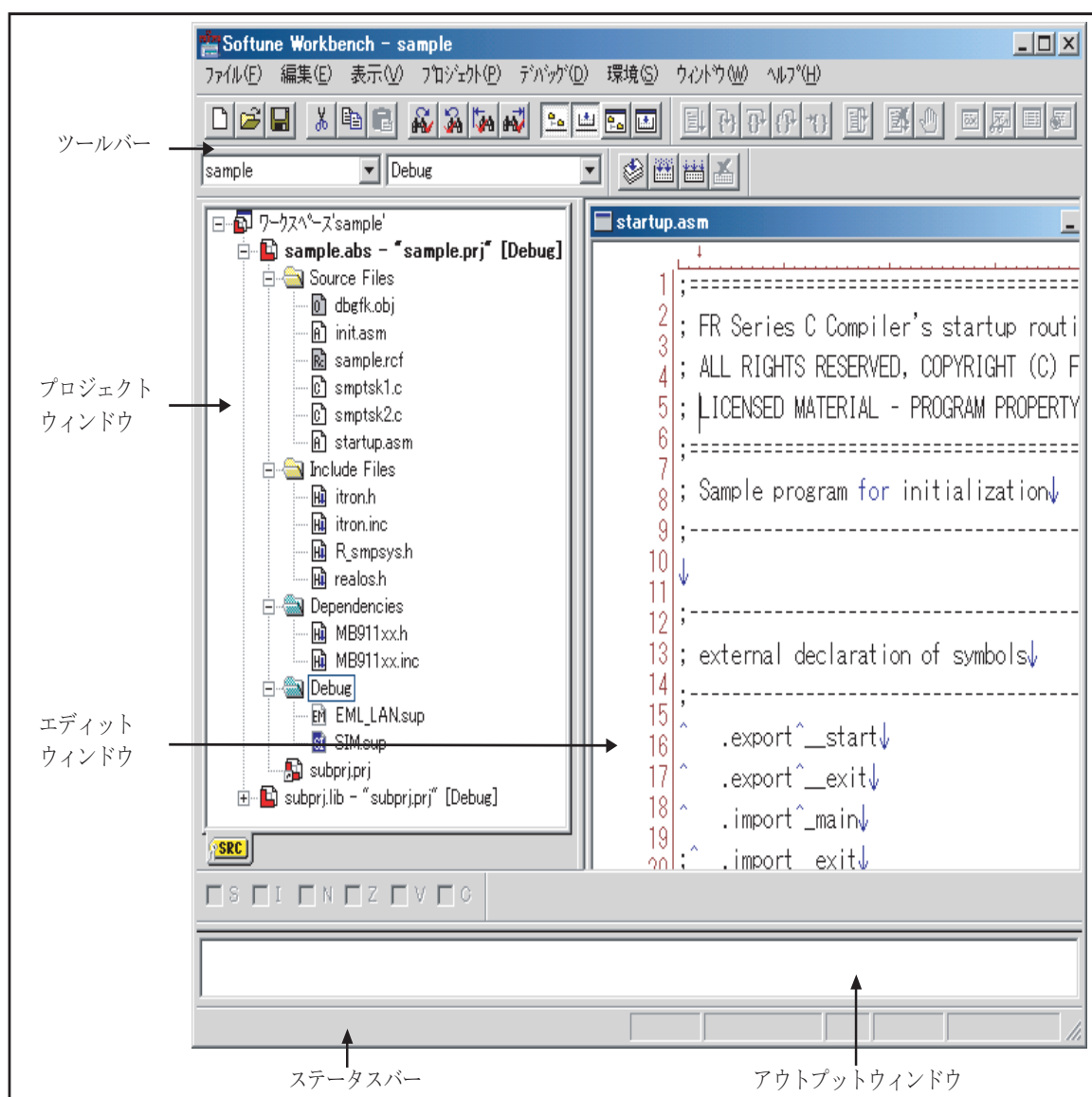
## 3.1 ウィンドウ構成

SOFTUNE Workbench のウィンドウは、メニューバー・ツールバー・ウィンドウ画面、ステータスバーから構成されます。メニューについては、「第4章 メニュー」で説明します。

### ■ メインウィンドウ

SOFTUNE Workbench のメインウィンドウを、図 3.1-1 に示します。この図に示したように、プロジェクトウィンドウ、アウトプットウィンドウなどの子ウィンドウと、ツールバーはメインウィンドウにドッキングして表示させることができます。

















図 3.1-1 メインウィンドウ



## 3.2 ツールバー

ツールバーには、よく使用するコマンドボタンをグループ単位に選択して配置することができます。選択できるグループとそれに含まれるコマンドボタンを以下に示します。

### ■ 共通





	新規作成
	開く
	上書き保存
	切り取り
	コピー
	貼り付け
	元に戻す
	やり直し
	次のエラーへジャンプ
	前のエラーへジャンプ
	最初のエラーへジャンプ
	最後のエラーへジャンプ
	プロジェクトウィンドウのドッキング
	出力ウィンドウのドッキング
	プロジェクトウィンドウの表示
	出力ウィンドウの表示

### ■ プロジェクト













アクティブプロジェクトとそのアクティブな構成名の表示および設定



### ■ ビルド

	コンパイル / アセンブル
	メイク
	ビルド
	メイク / ビルドの中止

## ■ デバッグ

	連続実行
	ステップイン
	ステップオーバー
	ステップアウト
	カーソル位置まで実行
	MCU リセット
	実行の停止
	ブレークポイント設定 / 解除
	レジスタウィンドウ
	ウォッチウィンドウ
	メモリウィンドウ
	逆アセンブルウィンドウ

## ■ フラグ

MCU のコンディションフラグの状態の表示および設定



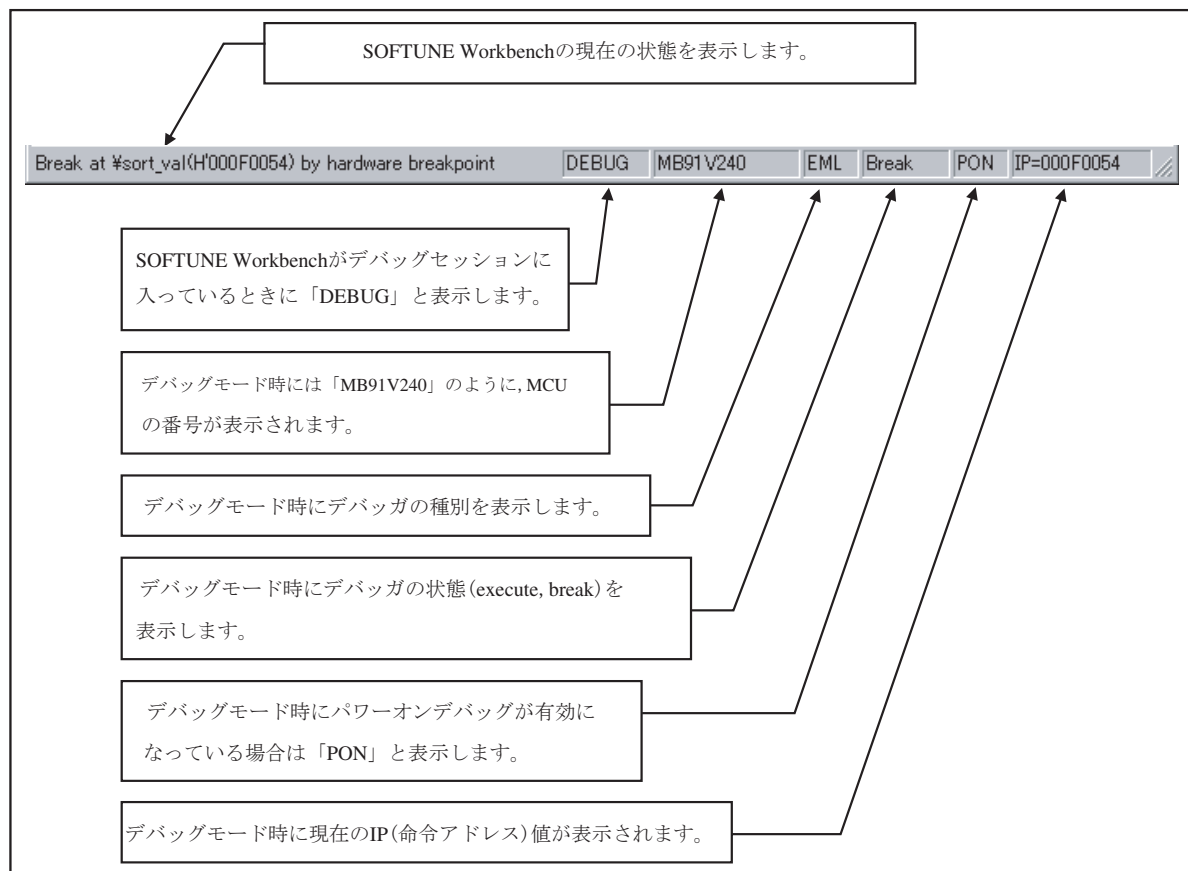
### 3.3 ステータスバー

SOFTUNE Workbench の現在の状態が表示されます。ステータスバーはメインウィンドウの最下部に表示されます。

#### ■ ステータスバー

ステータスバーには、いくつか決められた情報を表示する領域があります。表示される領域と情報について、以下の図 3.3-1 に簡単に示します。

図 3.3-1 ステータスバー



## 3.4 プロジェクトウィンドウ

---

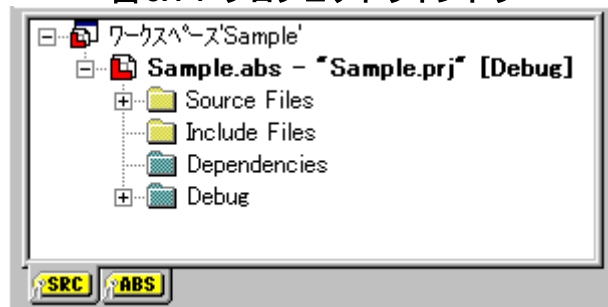
プロジェクトまたはロードモジュールに関する情報を表示します。

---

### ■ プロジェクトウィンドウ

プロジェクトウィンドウの例を図 3.4-1 に示します。

図 3.4-1 プロジェクトウィンドウ



プロジェクトウィンドウの下部にあるタブで、ウィンドウの表示内容を選択できます。  
プロジェクトウィンドウには、以下のタブがあります。

- SRC タブ

プロジェクトに関する情報を表示します。詳細は「3.4.1 SRC タブ」を参照してください。

- ABS タブ

デバッグ情報から取得したソースファイルに関する情報を表示します。詳細は「3.4.2 ABS タブ」を参照してください。

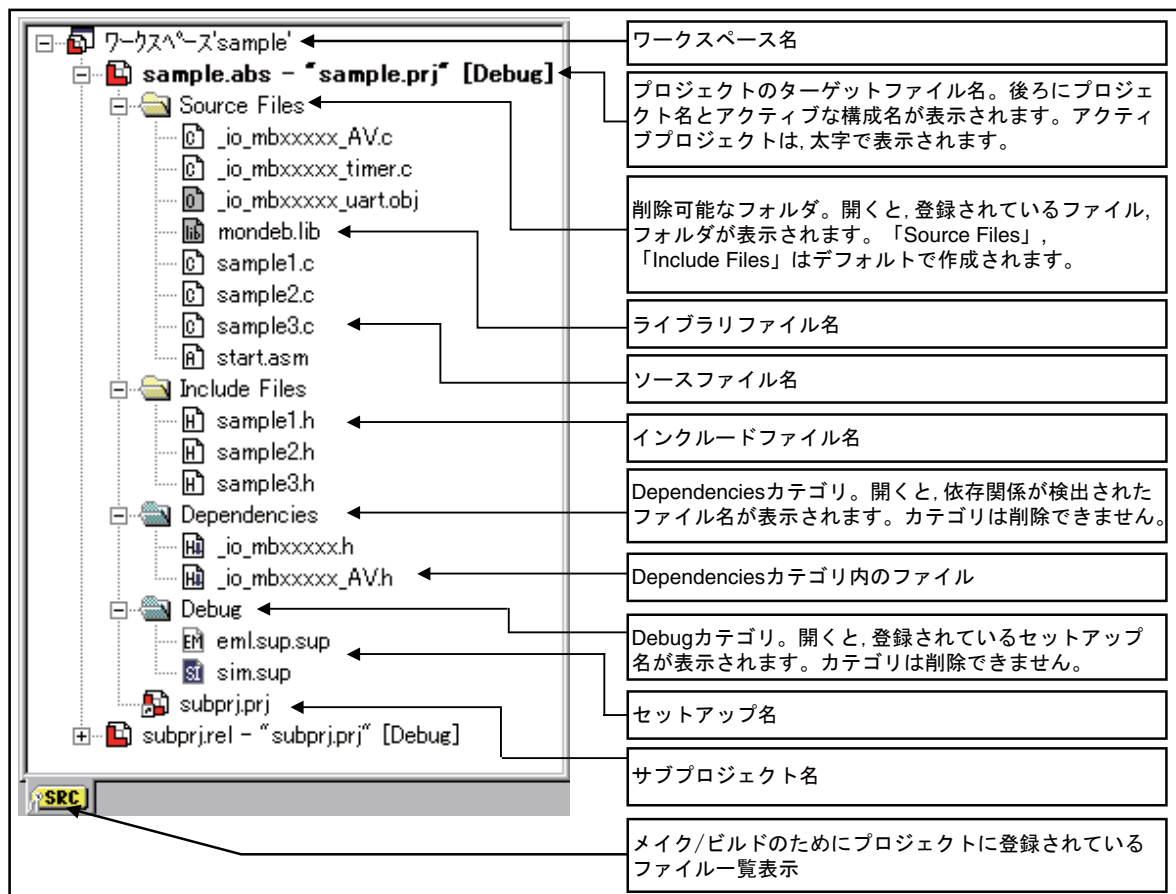
### 3.4.1 SRC タブ

プロジェクト名と、プロジェクトに登録されているファイル名がツリービュー形式で表示されます。

#### ■ SRC タブ

SRC タブの表示内容の例を図 3.4-2 に示します。

図 3.4-2 SRC タブ



#### ■ 機能

- ワークスペース名の表示
- ワークスペースに登録されているすべてのプロジェクトの表示
- サブプロジェクトの表示
 

親プロジェクトの下にそのプロジェクトのサブプロジェクトが表示されます。
- ターゲットファイル名の表示
- プロジェクトに登録されているすべてのソースファイルの表示
 

ソースファイル名をダブルクリックすると、エディタが開いてそのソースファイルを編集できるようになります。

### 第3章 ウィンドウ

- 依存関係が検出されたファイルの表示

ファイル名をダブルクリックすると、エディタが開いてそのファイルを編集できるようになります。

- デバッガのセットアップ名の表示






























デバッガのセットアップ名をダブルクリックすると、そのセットアップ情報にもとづいてデバッグが開始されます。





- ドラッグ アンド ドロップ

SRC タブ内のメンバの移動とエクスプローラからのメンバの登録を行うことができます。



## ■ SRC タブのビットマップイメージ一覧

	ワークスペース
	ターゲットファイル名
	サブプロジェクト
	フォルダ
	カテゴリ
	C++ ソースファイル
	C++ ソースファイル (メイク / ビルド対象外)
	個別オプションが指定されている C++ ソースファイル
	個別オプションが指定されている C++ ソースファイル (メイク / ビルド対象外)
	C ソースファイル
	C ソースファイル (メイク / ビルド対象外)
	個別オプションが指定されている C ソースファイル
	個別オプションが指定されている C ソースファイル (メイク / ビルド対象外)
	アセンブラソースファイル
	アセンブラソースファイル (メイク / ビルド対象外)
	個別オプションが指定されているアセンブラソースファイル
	個別オプションが指定されているアセンブラソースファイル (メイク / ビルド対象外)
	インクルードファイル
	インクルードファイル (依存関係あり)
	ライブラリファイル
	ライブラリファイル (メイク / ビルド対象外)
	オブジェクトファイル
	オブジェクトファイル (メイク / ビルド対象外)
	相対形式ファイル
	相対形式ファイル (メイク / ビルド対象外)
	REALOS コンフィグレーションファイル
	その他のユーザ登録ファイル
	シミュレータデバグ用セットアップ名
	有効なセットアップ名 (シミュレータデバグ)

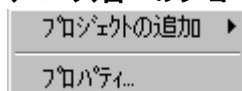
-  エミュレータデバッガ用セットアップ名
-  有効なセットアップ名 (エミュレータデバッガ)
-  モニタデバッガ用セットアップ名
-  有効なセットアップ名 (モニタデバッガ)

## ■ ショートカットメニュー

“ワークスペース名”上で右ボタンクリック

図 3.4-3 にショートカットメニューを表示します。

図 3.4-3 “ワークスペース名”のショートカットメニュー



### ● プロジェクトの追加

メニューには、次の2種類があります。

- 新規作成

ワークスペースにプロジェクトを新規作成して追加します(「4.5.2 プロジェクトの追加」を参照してください)。

- 既存のプロジェクト

ワークスペースに既存のプロジェクトを追加します(「4.5.2 プロジェクトの追加」を参照してください)。

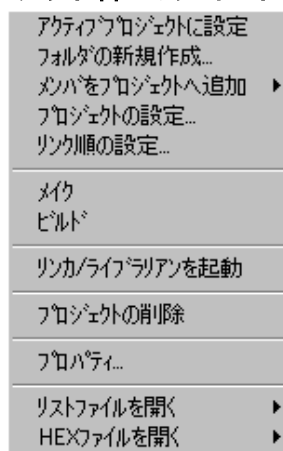
### ● プロパティ

ワークスペースファイルの情報を表示します(「4.3.9 プロパティ」を参照してください)。

“ターゲット名”上で右ボタンクリック

図 3.4-4 にショートカットメニューを表示します。

図 3.4-4 “ターゲット名”のショートカットメニュー



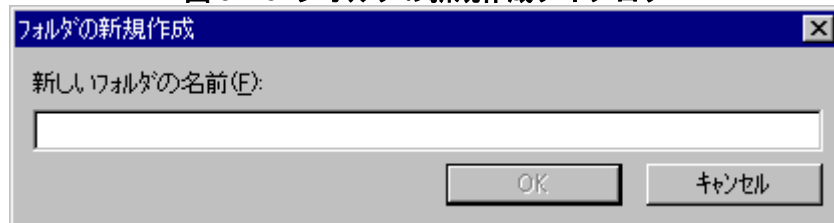
### ● アクティブプロジェクトに設定

現在選択しているプロジェクトをワークスペースのアクティブプロジェクトに設定します。

### ● フォルダの新規作成

選択すると、[フォルダの新規作成] ダイアログ（図 3.4-5）が表示されます。フォルダ名を入力して [OK] ボタンをクリックしてください。フォルダ名を入力すると、プロジェクトの 1 つ下の階層にフォルダが挿入されます。フォルダは作成した順番で並びます。

図 3.4-5 フォルダの新規作成ダイアログ



### ● メンバをプロジェクトへ追加

選択すると、次の 2 種類のサブメニューが表示されます。

#### ・ ファイル

現在選択しているプロジェクトにメンバをファイル単位で登録します（「4.5.3 メンバの追加」を参照してください）。

メンバの挿入先のデフォルトは、プロジェクトの 1 つ下の階層になります。

#### ・ ディレクトリ

現在選択しているプロジェクトにメンバをディレクトリ単位で登録します（「4.5.3 メンバの追加」を参照してください）。

メンバの挿入先のデフォルトは、プロジェクトの 1 つ下の階層になります。

### ● プロジェクトの設定

プロジェクトの設定を行います（「4.5.5 プロジェクトの設定」を参照してください）。

### ● リンク順の設定

選択すると、「リンク順の設定」ダイアログ（図 3.4-6）が表示されます。[リンク順] に表示されているファイルの上から順番にリンクされます。リンク順を変更する場合は、[設定の対象] ボックスで設定を行いたい構成を選択して、ファイル名を選択し [上へ] または [下へ] ボタンで目的の位置までファイルを移動させます。

[エクスポート] ダイアログ（図 3.4-7）を用いると、現在のリンク順を他の構成に反映させることができます。リンク対象外のファイルは、灰色で表示されます。

図 3.4-6 リンク順の設定ダイアログ

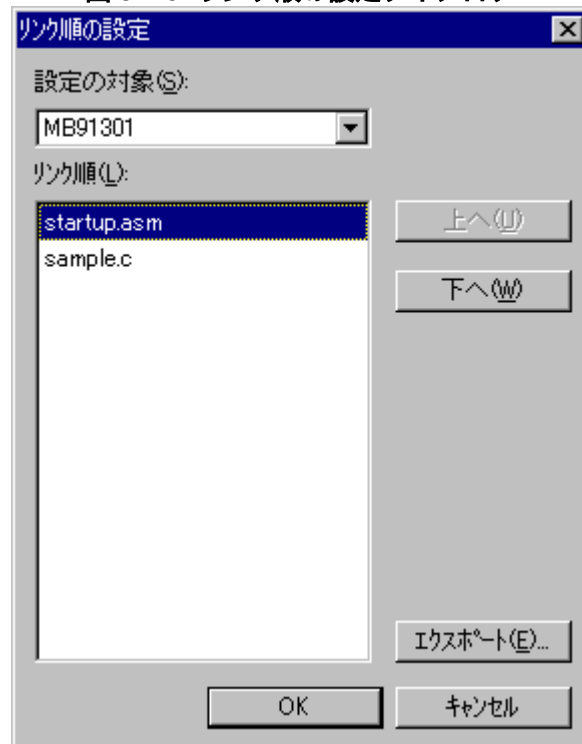
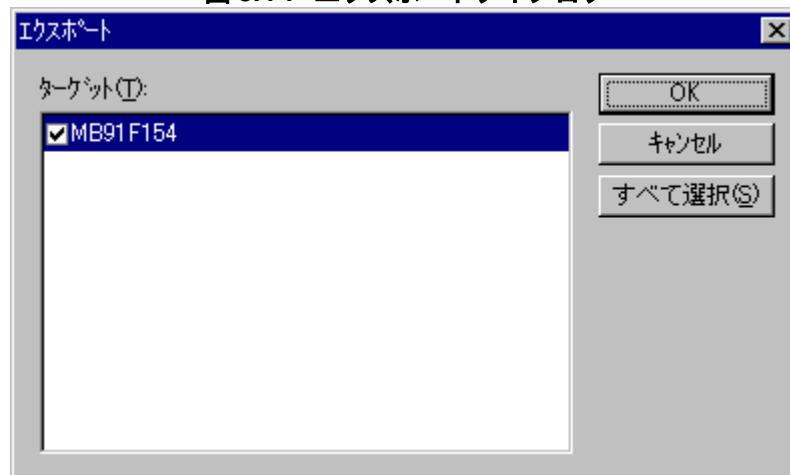


図 3.4-7 エクスポートダイアログ



- メイク/ビルド

現在選択しているプロジェクトのアクティブな構成でメイク/ビルドして、ターゲットファイルを作成します。

- リンカ/ライブラリアンを起動

現在選択しているプロジェクトのアクティブな構成でリンカ/ライブラリアンを起動して、ターゲットファイルを作成します。

- プロジェクトの削除

指定したプロジェクトをワークスペースから削除します。プロジェクトファイルそのものの削除は行いません。

- プロパティ

ファイルの情報を表示します（「4.3.9 プロパティ」を参照してください）。

- リストファイルを開く

表示したいリストファイルをサブメニューから選択してください。

リストファイルが作られていないときは、サブメニューの内容は選択できません。

- HEX ファイルを開く

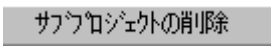
表示したい HEX ファイルをサブメニューから選択してください。

HEX ファイルが作られていないときは、サブメニューの内容は選択できません。

“サブプロジェクト名”上で右ボタンクリック

図 3.4-8 にショートカットメニューを表示します。

図 3.4-8 “サブプロジェクト名”のショートカットメニュー



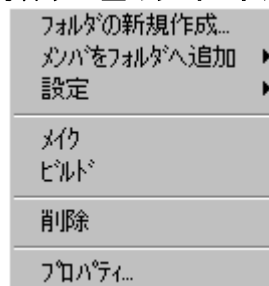
- サブプロジェクトの削除

指定したプロジェクトのサブプロジェクトを削除します。再度サブプロジェクトを定義するには、「4.5.7 プロジェクトの依存関係」を参照してください。

“フォルダ”上で右ボタンクリック

図 3.4-9 にショートカットメニューを表示します。

図 3.4-9 “フォルダ”上のショートカットメニュー



- フォルダの新規作成

選択すると、[フォルダの新規作成] ダイアログ（図 3.4-5）が表示されます。フォルダ名を入力して [OK] ボタンをクリックしてください。

フォルダ名を入力すると、選択しているフォルダ内にフォルダが挿入されます。

フォルダは作成した順番で並びます。

### ● メンバをフォルダへ追加

選択すると、次の2種類のサブメニューが表示されます。

#### • ファイル

現在選択しているプロジェクトにメンバをファイル単位で登録します(「4.5.3 メンバの追加」を参照してください)。

メンバの挿入先のデフォルトは、選択しているフォルダ内になります。

#### • ディレクトリ

現在選択しているプロジェクトにメンバをディレクトリ単位で登録します(「4.5.3 メンバの追加」を参照してください)。

メンバの挿入先のデフォルトは、選択しているフォルダ内になります。

### ● 設定

選択すると、次の2種類のサブメニューが表示されます。

#### • 個別の設定

プロジェクトの設定を行います(「4.5.5 プロジェクトの設定」を参照してください)。

#### • 共通設定に戻す

指定したフォルダ内のすべての個別オプションを共通オプションに戻します。

### ● メイク/ビルド

現在選択しているプロジェクトのアクティブな構成でメイク/ビルドして、ターゲットファイルを作成します。

### ● 削除

選択しているフォルダとフォルダ内のファイルをプロジェクトからすべて削除します。ファイルそのものの削除は行いません。

削除できないファイル(RCFファイル)を含む場合は、フォルダは削除されません。

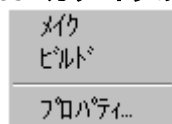
### ● プロパティ

フォルダの情報を表示します(「4.3.9 プロパティ」を参照してください)。

“Dependencies” カテゴリ上で右ボタンクリック

図 3.4-10 にショートカットメニューを表示します。

図 3.4-10 “Dependencies” カテゴリのショートカットメニュー



### ● メイク/ビルド

現在選択しているプロジェクトのアクティブな構成でメイク/ビルドして、ターゲットファイルを作成します。

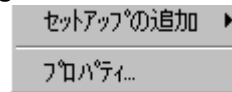
### ● プロパティ

フォルダの情報を表示します(「4.3.9 プロパティ」を参照してください)。

“ Debug ” カテゴリ上で右ボタンクリック

図 3.4-11 にショートカットメニューを表示します。

図 3.4-11 "Debug" カテゴリのショートカットメニュー



#### ● セットアップの追加

選択すると、次の2種類のサブメニューが表示されます。

##### • 新規

選択すると、[セットアップの新規作成]ダイアログ(図 3.4-12)が表示されます。セットアップ名を入力して[OK]ボタンをクリックしてください。

セットアップ名を入力すると、セットアップウィザードが起動します。セットアップウィザードについては、「4.7.2.5 セットアップウィザード」を参照してください。

##### • 参照

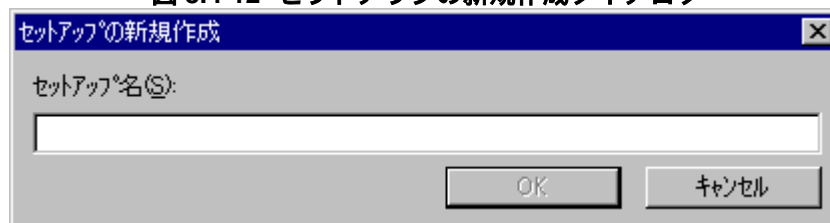
選択すると、[セットアップの新規作成]ダイアログ(図 3.4-12)が表示されます。セットアップ名を入力して[OK]ボタンをクリックしてください。

セットアップ名を入力すると、ファイル選択のダイアログが表示されますのでファイルを選択して、[開く]ボタンをクリックしてください。指定したファイルから情報を読み込んで設定します。

#### ● プロパティ

フォルダの情報を表示します(「4.3.9 プロパティ」を参照してください)。

図 3.4-12 セットアップの新規作成ダイアログ



"ソースファイル名"上で右ボタンクリック

図 3.4-13 にショートカットメニューを表示します。

図 3.4-13 "ソースファイル名" カテゴリのショートカットメニュー



### ● 開く

デバッグセッション時は、指定されたソースファイルをソースウィンドウに開きます。  
デバッグセッション時以外は、指定されたソースファイルをエディットウィンドウに開きます。

### ● コンパイル（アセンブル）

指定されたソースファイルが、C/C++ ソースであるかアセンブラソースであるかを自動的に判定して、コンパイルまたはアセンブルをします。

### ● 設定

選択すると、次の2種類のサブメニューが表示されます。

#### • 個別の設定

プロジェクトの設定を行います（「4.5.5 プロジェクトの設定」を参照してください）。

#### • 共通設定に戻す

指定したフォルダ内のすべての個別オプションを共通オプションに戻します。

### ● 削除

指定したファイルをプロジェクトメンバから解除します。ファイルそのものの削除は行いません。

### ● プロパティ

フォルダの情報を表示します（「4.3.9 プロパティ」を参照してください）。

### ● リストファイルを開く

表示したいリストファイルをサブメニューから選択してください。

リストファイルが作られていないときは、サブメニューの内容は選択できません。

### ● アセンブラソースファイルを開く

C/C++ ソースファイルが選択されている場合、このメニューが表示されます。

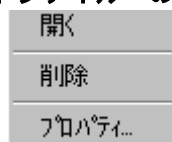
指定された C/C++ ソースファイルから、コンパイラが出力したアセンブラソースファイルが開きます。

コンパイルされていない場合は選択できません。

“インクルードファイル”、“その他のユーザ登録ファイル”上で右ボタンクリック

図 3.4-14 にショートカットメニューを表示します。

図 3.4-14 “インクルードファイル”のショートカットメニュー



### ● 開く

デバッグセッション時は、指定されたインクルードファイルをソースウィンドウに開きます。

デバッグセッション時以外は、指定されたインクルードファイルをエディットウィンドウに開きます。



### ● 削除

指定したファイルをプロジェクトメンバから解除します。ファイルそのものの削除は行いません。

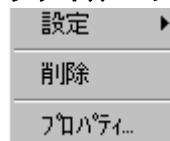
### ● プロパティ

ファイルの情報を表示します。(「4.3.9 プロパティ」を参照してください。)

“ライブラリファイル”, “オブジェクトファイル”, “相対形式ファイル” 上で右ボタンクリック

図 3.4-15 にショートカットメニューを表示します。

図 3.4-15 “ライブラリファイル” のショートカットメニュー



### ● 設定

選択すると、次の2種類のサブメニューが表示されます。

- 個別の設定  
プロジェクトの設定を行います(「4.5.5 プロジェクトの設定」を参照してください)。
- 共通設定に戻す  
指定したフォルダ内のすべての個別オプションを共通オプションに戻します。

### ● 削除

指定したファイルをプロジェクトメンバから解除します。ファイルそのものの削除は行いません。

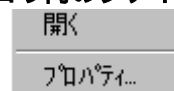
### ● プロパティ

ファイルの情報を表示します(「4.3.9 プロパティ」を参照してください)。

“Dependencies カテゴリ内のファイル” 上で右ボタンクリック

図 3.4-16 にショートカットメニューを表示します。

図 3.4-16 “Dependencies カテゴリ内のファイル” 上のショートカットメニュー



### ● 開く

選択すると、次の2種類のサブメニューが表示されます。

- デバッグセッション時は、指定されたソースファイルをソースウィンドウに開きます。
- デバッグセッション時以外は、指定されたソースファイルをエディットウィンドウに開きます。

### ● プロパティ

ファイルの情報を表示します（「4.3.9 プロパティ」を参照してください）。

“ セットアップ名 ” 上で右ボタンクリック

図 3.4-17 にショートカットメニューを表示します。

図 3.4-17 “ セットアップ名 ” 上のショートカットメニュー



### ● デバッグの開始 / デバッグの終了

デバッグセッション時は、「現在のセッションを終了してデバッグを開始しますか？」という確認のダイアログが表示されます。「はい」を選択してください。

デバッグを終了し、指定したセットアップ情報にもとづいてデバッグを開始します。

デバッグセッション時以外は、確認ダイアログを表示せずデバッグを開始します。

デバッグの開始はアクティブプロジェクトでのみ行えます。

デバッグセッション時は、有効なセットアップのみ「デバッグの開始」が「デバッグの終了」に変わります。「デバッグの終了」を選択すると、デバッグを終了します。

### ● 設定変更

セットアップの設定を変更するためのセットアップウィザードが起動します（「4.7.2.5 セットアップウィザード」を参照してください）。

### ● 削除

デバッグのセットアップ情報をプロジェクトから削除します。

有効なセットアップ情報は削除できません。削除するには一度有効なセットアップを変更してください。

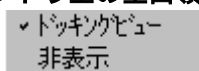
### ● プロパティ

セットアップ情報を表示します（「4.3.9 プロパティ」を参照してください）。

“ プロジェクトウィンドウ上の空白領域 ” で右クリック

図 3.4-18 にショートカットメニューを表示します。

図 3.4-18 “ プロジェクトウィンドウ上の空白領域 ” のショートカットメニュー



### ● ドッキングビュー

チェック状態でフレームとドッキングします。

### ● 非表示

プロジェクトウィンドウが非表示の状態になります。

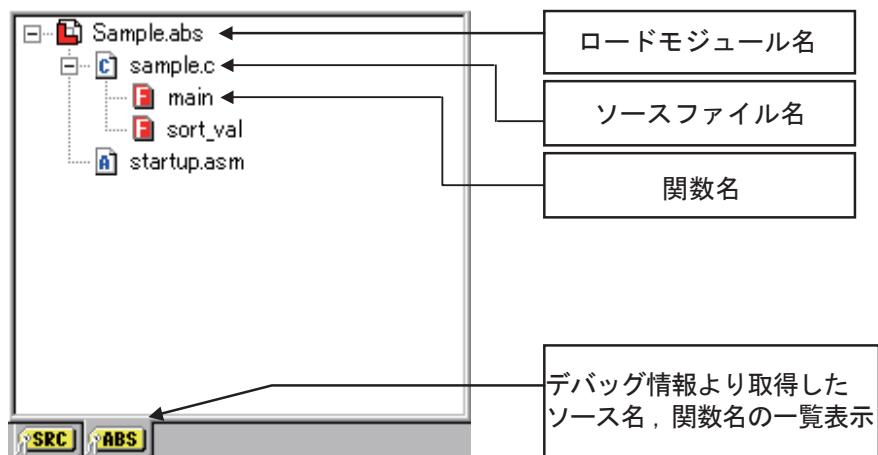
### 3.4.2 ABS タブ

デバッグ情報から取得した情報として、ロードモジュール名、ソースファイル名、関数名をツリービュー形式で表示します。

#### ■ ABS タブ

ABS タブの表示内容の例を図 3.4-19 に示します。

図 3.4-19 ABS タブ



#### ■ 機能

デバッグ情報から次の情報を表示します。

- ロードモジュール名

- ソースファイル名

ソースファイル名のダブルクリックにより、ソースウィンドウを表示します。

- 関数名

関数名のダブルクリックにより、関数定義位置のソースウィンドウへジャンプします。

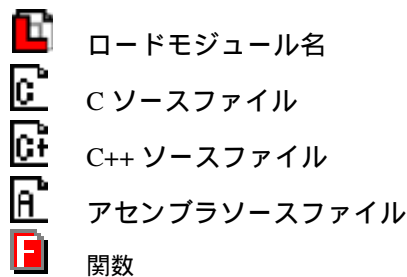
- カバレッジ率

ロードモジュール、ソースファイル、関数のカバレッジ率を表示します。カバレッジ率表示を最新の状態にするには、ABS タブの更新を行う必要があります。

全領域がカバレッジ測定範囲外のカバレッジ率は "(--%)" と表示します。一部の領域がカバレッジ測定範囲外のカバレッジ率は後ろにアスタリスク "\*" を付けて表示します。

この機能は高速版シミュレータデバッグ時のみ有効です。

## ■ ABS タブのビットマップイメージ一覧

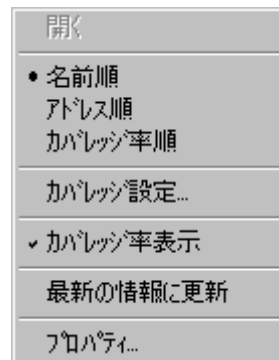


## ■ ショートカットメニュー

“ ロードモジュール名 ” 上で右クリック

図 3.4-20 にショートカットメニューを表示します。

図 3.4-20 “ ロードモジュール名 ” 上のショートカットメニュー



### ● 開く

このメニューは選択できません。

### ● 名前順

ツリー内のアイテムをアルファベット順でソートします。

### ● アドレス順

ツリー内の関数をアドレス順でソートします。その他のアイテムはアルファベット順でソートします。

### ● カバレッジ率順

ツリー内のアイテムをカバレッジ率が低い順でソートします。  
高速版シミュレータデバッグ時のみ有効です。

### ● カバレッジ設定

カバレッジ設定ダイアログを開きます。  
高速版シミュレータデバッグ時のみ有効です。

### ● カバレッジ率表示

カバレッジ率表示の有無を切り替えます。カバレッジ率表示が有効な場合には、各アイテムのカバレッジ率が表示されます。高速版シミュレータデバッグ時のみ有効です。

- 最新の情報に更新

ウィンドウ内容を更新します。カバレッジ率表示が有効な場合には、カバレッジ率表示を最新の状態に更新します。高速版シミュレータデバッグ時のみ有効です。

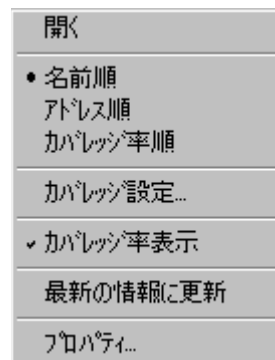
- プロパティ

ロードモジュールの情報を表示します（「4.3.9 プロパティ」を参照してください）。

“ソースファイル名”上で右クリック

図 3.4-21 にショートカットメニューを表示します。

図 3.4-21 “ソースファイル名”上のショートカットメニュー



- 開く

選択したソースファイルをソースウィンドウで開きます。

- 名前順

ツリー内のアイテムをアルファベット順でソートします。

- アドレス順

ツリー内の関数をアドレス順でソートします。その他のアイテムはアルファベット順でソートします。

- カバレッジ率順

ツリー内のアイテムをカバレッジ率が低い順でソートします。  
高速版シミュレータデバッグ時のみ有効です。

- カバレッジ設定

カバレッジ設定ダイアログを開きます。  
高速版シミュレータデバッグ時のみ有効です。

- カバレッジ率表示

カバレッジ率表示の有無を切り替えます。カバレッジ率表示が有効な場合には、各アイテムのカバレッジ率が表示されます。高速版シミュレータデバッグ時のみ有効です。

- 最新の情報に更新

ウィンドウ内容を更新します。カバレッジ率表示が有効な場合には、カバレッジ率表示を最新の状態に更新します。高速版シミュレータデバッグ時のみ有効です。

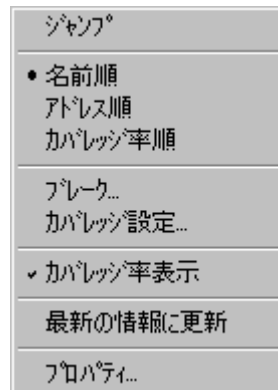
- プロパティ

ソースファイルのプロパティを表示します。

“関数名”上で右クリック

図 3.4-22 にショートカットメニューを表示します。

図 3.4-22 “関数名”上のショートカットメニュー



● ジャンプ

選択した関数をソースウィンドウで開きます。

● 名前順

ツリー内のアイテムをアルファベット順でソートします。

● アドレス順

ツリー内の関数をアドレス順でソートします。

● カバレッジ率順

ツリー内のアイテムをカバレッジ率が低い順でソートします。

高速版シミュレータデバッグ時のみ有効です。

● ブレーク

ブレークダイアログを開き、選択した関数にブレークポイントを設定します。

● カバレッジ設定

カバレッジ設定ダイアログを開きます。

高速版シミュレータデバッグ時のみ有効です。

● カバレッジ率表示

カバレッジ率表示の有無を切り替えます。カバレッジ率表示が有効な場合には、各アイテムのカバレッジ率が表示されます。高速版シミュレータデバッグ時のみ有効です。

● 最新の情報に更新

ウィンドウ内容を更新します。カバレッジ率表示が有効な場合には、カバレッジ率表示を最新の状態に更新します。高速版シミュレータデバッグ時のみ有効です。

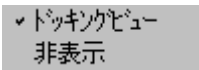
● プロパティ

選択した関数のシンボル情報を表示します。

“プロジェクトウィンドウ上の空白領域”で右クリック

図 3.4-23 にショートカットメニューを表示します。

図 3.4-23 " プロジェクトウィンドウ上の空白領域 " のショートカットメニュー



- ドッキングビュー

チェック状態でフレームとドッキングします。

- 非表示

プロジェクトウィンドウが非表示の状態になります。

## 3.5 アウトプットウィンドウ

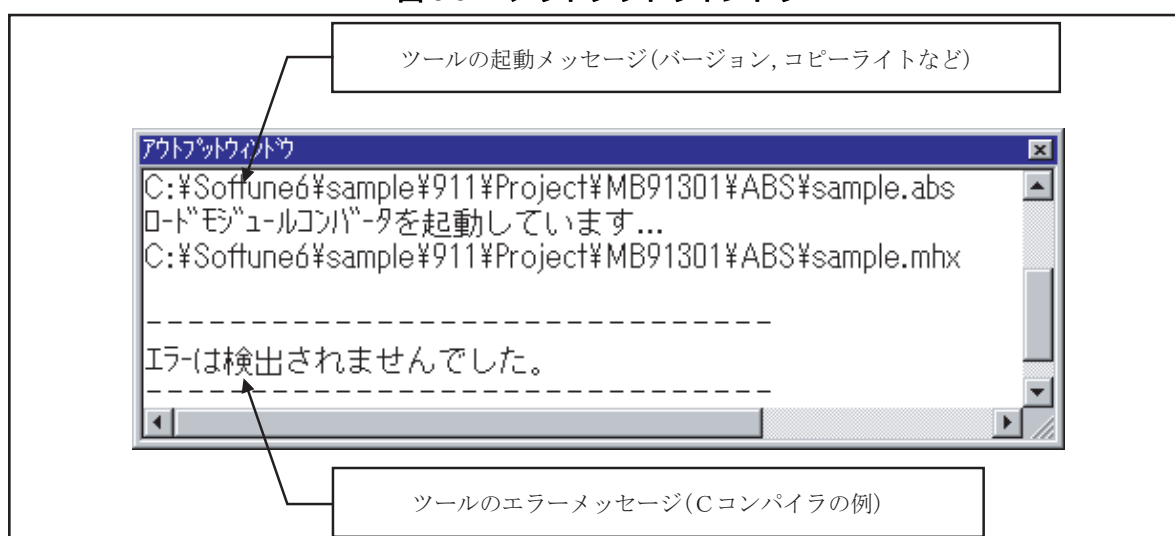
メイク/ビルド時の、コンパイラなどのバージョン情報や、エラーメッセージなどを表示するウィンドウです。

### ■ アウトプットウィンドウ

アウトプットウィンドウの例を図 3.5-1 に示します。

この図の例のようにエラーメッセージが出力されているときは、その行でマウスの左ボタンをダブルクリックすることにより、エラーの発生したソース行へ自動的にジャンプすることができます。

図 3.5-1 アウトプットウィンドウ



### ■ ショートカットメニュー

メニューには次の7種類があります。

- コピー : 選択した文字列をクリップボードにコピーします。
- クリア : アウトプットウィンドウのすべての表示をクリアします。
- ドッキングビュー : ドッキングビューの設定 / 解除をします。
- 非表示 : アウトプットウィンドウを非表示にします。
- フォントの設定 : アウトプットウィンドウ中の表示のフォントを選択します。
- キーワードの強調 : エラーメッセージのエラー番号を強調するかどうかの設定を行います。
- ロギング : アウトプットウィンドウへの出力結果をファイルに保存します。



## ■ 操作

### ● コピー

- 1) コピーしたい文字列を選択してください。
- 2) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 3) [コピー]を選択してください。  
「4.3.2 切り取り / コピー / 貼り付け / 削除」を参照してください。

### ● クリア

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [クリア]を選択してください。

### ● ドッキングビュー

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ドッキングビュー]を選択してください。
  - チェック状態でフレームとドッキングします。

### ● 表示

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [非表示]を選択してください。
  - アウトプットウィンドウが非表示の状態になります。

### ● フォントの設定

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [フォントの設定]を選択してください。  
フォント選択ダイアログ(図 3.5-2)が表示されます。
- 3) [フォント]と[サイズ]を選択して,[OK]ボタンをクリックしてください。

図 3.5-2 フォントの設定ダイアログ



### 第3章 ウィンドウ

#### ● キーワードの強調

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
  - 2) [ キーワードの強調 ] を選択してください。
- チェック状態だとエラーメッセージのキーワード ( エラー番号 ) が赤色で表示されます。

#### ● ログイング

- 1) アウトプットウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ ログイング ] - [ 開始 ] を選択し記録するファイル名を指定してください。アウトプットウィンドウの出力結果がファイルに記録されます。
- 3) ログイングを一時中断したい場合は, [ ログイング ] - [ 記録 ] を選択してチェックをはずしてください。もう一度選択するとログイングが再開されます。チェックがついている状態の時にファイルに記録されます。
- 4) ログイングを終了したい時は, [ ログイング ] - [ 終了 ] を選択してください。

## 3.6 エディットウィンドウ (標準エディタ)

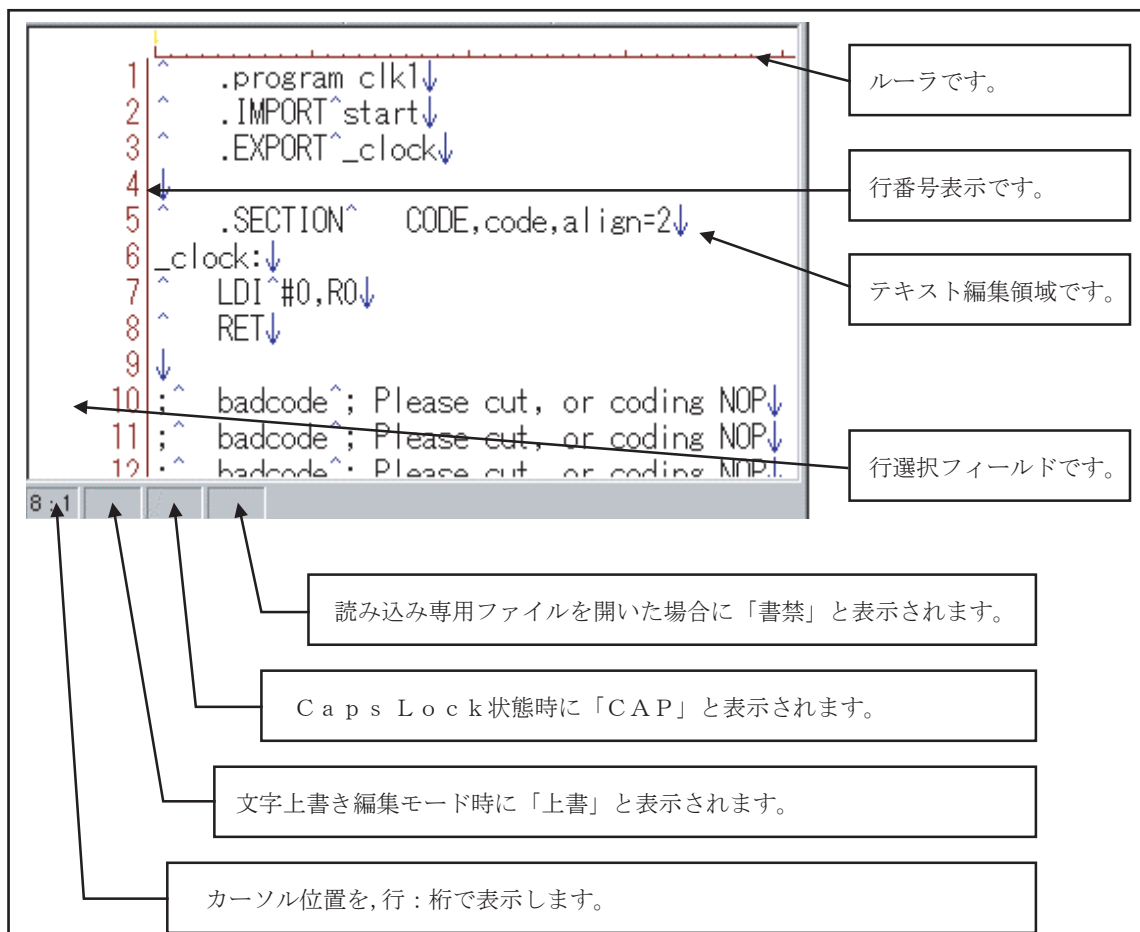
ソースファイルやドキュメントファイルなどの表示と編集に使用するウィンドウです。

### ■ エディットウィンドウ

メインウィンドウにドッキングしている形式のエディットウィンドウの例を、図 3.6-1 に示します。

この図の例のようにメインウィンドウにドッキングした形式で表示できるのは、SOFTUNE Workbench に内蔵している標準エディタを使用している場合に限られます。SOFTUNE Workbench では、任意の外部エディタを登録して使用することができますが、その場合は、SOFTUNE Workbench とは別画面になります。

図 3.6-1 エディットウィンドウ



## ■ ショートカットメニュー

メニューには、次の18種類があります。

- ソースウィンドウ : ソースウィンドウを表示します。
- 元に戻す : 最後の編集を取り消します。
- やり直し : 取り消した編集を再実行します。
- 切り取り : 選択されている文字列をクリップボードに移動します。
- コピー : 選択されている文字列をクリップボードにコピーします。
- 貼り付け : クリップボードの文字列をカーソル位置に設定します。
- 削除 : 選択されている文字列を削除します。
- すべて選択 : そのバッファ中のすべての文字を選択します。
- 印刷 : そのファイルを印刷します。
- プロジェクトへ追加 : そのファイルをアクティブプロジェクトに登録します。
- 検索 : 文字列検索のダイアログを表示します。
- 置換 : 文字列置換のダイアログを表示します。
- ファイルから検索 : 「4.3.5 ファイルから検索」を参照してください。
- 指定行へジャンプ : ジャンプの行番号指定のダイアログを開きます。
- ブックマーク : 「4.3.7 ブックマーク」を参照してください。
- 標準エディタ設定 : 「3.6.1 標準エディタ設定」を参照してください。
- ステータスバー : ウィンドウごとのステータスバーの表示を切り替えます。
- プロパティ : ファイルの情報を表示します。

### ● ソースウィンドウ

- 1) 右ボタンでショートカットメニューを表示し、[ソースウィンドウ]を選択してください。
- SOFTUNE Workbenchがデバッグセッションのときは、ソースウィンドウが表示されます(「3.7 ソースウィンドウ」を参照してください)。デバッグセッションでないときは、このメニューは選択できません。

### ● 元に戻す

- 1) 右ボタンでショートカットメニューを表示し、[元に戻す]を選択してください。
- 最後に行った編集をキャンセルして元に戻します(「4.3.1 元に戻す / やり直し」を参照してください)。

### ● やり直し

- 1) 右ボタンでショートカットメニューを表示し、[やり直し]を選択してください。
- 最後に行った編集のキャンセルをキャンセルします。

### ● 切り取り

- 1) 切り取りたい文字列を選択してください。
  - 2) 右ボタンでショートカットメニューを表示し、[移動]を選択してください。
- 選択した文字列を、エディットウィンドウから削除し、クリップボードに移動します(「4.3.2 切り取り / コピー / 貼り付け / 削除」を参照してください)。

## ● コピー

- 1) コピーしたい文字列を選択してください。
  - 2) 右ボタンでショートカットメニューを表示し、[コピー]を選択してください。
- 選択した文字列を、クリップボードにコピーします(「4.3.2 切り取り / コピー / 貼り付け / 削除」を参照してください)。

## ● 貼り付け

- 1) 文字列を挿入したい位置にカーソルを移動してください。
  - 2) 右ボタンでショートカットメニューを表示し、[貼り付け]を選択してください。
- クリップボード中の文字列を、カーソル位置に挿入します(「4.3.2 切り取り / コピー / 貼り付け / 削除」を参照してください)。

## ● 削除

- 1) 削除したい文字列を選択してください。
  - 2) 右ボタンでショートカットメニューを表示し、[削除]を選択してください。
- 指定された文字列を削除します(「4.3.2 切り取り / コピー / 貼り付け / 削除」を参照してください)。

## ● すべて選択

- 1) 右ボタンでショートカットメニューを表示し、[すべて選択]を選択してください。
- エディットウィンドウ中のすべてのテキストを選択して、反転表示します(「4.3.3 すべて選択」を参照してください)。

## ● 印刷

- 1) 右ボタンでショートカットメニューを表示し、[印刷]を選択してください。
- 印刷ダイアログが表示されます(「4.2.9 印刷」を参照してください)。

## ● プロジェクトへ追加

- 1) 右ボタンでショートカットメニューを表示し、[プロジェクトへ追加]を選択してください。
- 現在編集集中のファイルをプロジェクトに追加します。

## ● 検索 / 置換

- 1) 右ボタンでショートカットメニューを表示し、[検索]または[置換]を選択してください。
- クリップボードの文字列が、[検索する文字列]に設定されて、検索または置換ダイアログが開きます(「4.3.4 検索 / 置換」を参照してください)。

## ● ファイルから検索

- 1) 右ボタンでショートカットメニューを表示し、[ファイルから検索]を選択してください。
- クリップボードの文字列が[検索する文字列]に設定され、ファイルから検索ダイアログが開きます(「4.3.5 ファイルから検索」を参照してください)。

#### ● 指定行へジャンプ

- 1) 右ボタンでショートカットメニューを表示し,[指定行へジャンプ]を選択してください。
- ジャンプ先の行を指定するダイアログが表示されます(「4.3.6 ジャンプ」を参照してください)。
- 2) ジャンプ先の行番号を設定してください。
- 3) [OK] ボタンをクリックしてください。

#### ● ステータスバー

エディットウィンドウごとに付けられているステータスバーの、表示/非表示を切り替えます。

- 1) 右ボタンでショートカットメニューを表示し,[ステータスバー]を選択してください。
- 既に表示されている場合は、ステータスバーの表示を非表示にし、メニューのチェックを消去します。
- 表示されていない場合は、ステータスバーを表示して、メニューの左端にチェックマークを付けます。

#### ● プロパティ

- 1) 右ボタンでショートカットメニューを表示し,[プロパティ]を選択してください。
- ファイルの情報を表示します(「4.3.9 プロパティ」を参照してください)。

## 3.6.1 標準エディタ設定

---

ここでは、標準エディタのカスタマイズについて説明します。

---

### ■ 標準エディタ設定

標準エディタはいくつかの機能をカスタマイズすることができます。

#### ● カスタマイズ可能な項目

- 表示機能
  - 改行マーク表示
  - タブコードマーク表示
  - EOF コードマーク表示
  - 行番号表示
  - 全角空白表示
  - ルーラ表示
  - 自動インデント機能
  - C/C++ キーワード着色表示
  - ASM キーワード着色表示
  - ユーザ定義キーワード着色表示
  - 注釈文着色表示
- フォント
- タブ数
- 表示色
  - ブックマーク
  - エラー行
  - C/C++ 言語キーワード
  - ASM 言語キーワード
  - ユーザ定義キーワード
  - 注釈文
  - 改行
  - タブ
  - 全角空白
- 強調するキーワード

### ■ カスタマイズ手順 (表示機能選択)

- 1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して、[ 標準エディタ設定 ] を選択してください。
  - エディタ設定のダイアログ ( 図 3.6-2 ) が表示されます。
- 2) [ 項目 ] リストから設定または解除したい機能項目を選んで、その項目の左横にあるをクリックしてください。

3) 他の項目の設定がなければ, [OK] ボタンをクリックしてください。

- カラー選択などを行うときは, それらのすべての設定が終わってから [OK] ボタンをクリックしてください。

#### ■ カスタマイズ手順 (表示色選択)

1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して, [標準エディタ設定] を選択してください。

- エディタ設定のダイアログ (図 3.6-2) が表示されます。

2) [ブックマーク], [エラー], [C/C++ 言語], [ASM 言語] の内の設定したい項目のカラーボタンをクリックしてください。[表示 2] タブから「注釈文」「改行」「タブ」「ユーザ定義キーワード」「全角空白」の設定が可能です。

- 色の設定ダイアログが表示されます。

3) 好みの色を選択して, [OK] ボタンをクリックしてください。

- ボタンの色が選択した色に変化します。

4) 他の項目の設定がなければ, [OK] ボタンをクリックしてください。

- フォント選択などを行うときは, それらのすべての設定が終わってから [OK] ボタンをクリックしてください。

#### ■ カスタマイズ手順 (キーワードの追加)

1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して, [標準エディタ設定] を選択してください。

- エディタ設定のダイアログ (図 3.6-2) が表示されます。

2) [表示 2] タブをクリックし, 変更するキーワード種別をコンボボックスから選択して [詳細] ボタンをクリックしてください。

- キーワードダイアログ (図 3.6-4) が表示されます。

3) 強調表示したいキーワードを [新規項目] に入力して [追加] ボタンをクリックしてください。

4) [OK] ボタンをクリックしてください。

- キーワードファイルが変更されます。

5) 他の項目の設定がなければ, [OK] ボタンをクリックしてください。

- 表示機能選択などを行うときは, それらのすべての設定が終わってから [OK] ボタンをクリックしてください。

#### ■ カスタマイズ手順 (タブ数)

1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して, [標準エディタ設定] を選択してください。

- エディタ設定のダイアログ (図 3.6-2) が表示されます。

2) [タブ数] フィールドに表示されている値を修正してください。

3) 他の項目の設定がなければ, [OK] ボタンをクリックしてください。

- 表示機能選択などを行うときは, それらのすべての設定が終わってから [OK] ボタンをクリックしてください。



## ■ カスタマイズ手順 (キーワードの削除)

- 1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して、[標準エディタ設定]を選択してください。
  - エディタ設定のダイアログ (図 3.6-2) が表示されます。
- 2) [表示 2] タブをクリックし、変更するキーワード種別をコンボボックスから選択して[詳細]ボタンをクリックしてください。
  - キーワードダイアログ (図 3.6-4) が表示されます。
- 3) 削除したいキーワードを[キーワード一覧]で選択して[削除]ボタンをクリックしてください。
- 4) [OK] ボタンをクリックしてください。
  - キーワードファイルが変更されます。
- 5) 他の項目の設定がなければ、[OK] ボタンをクリックしてください。
  - 表示機能選択などを行うときは、それらのすべての設定が終わってから [OK] ボタンをクリックしてください。

## ■ カスタマイズ手順 (キーワードの初期化)

- 1) エディットウィンドウで右ボタンをクリックしてショートカットメニューを表示して、[標準エディタ設定]を選択してください。
  - エディタ設定のダイアログ (図 3.6-2) が表示されます。
- 2) [表示 2] タブをクリックし、変更するキーワード種別をコンボボックスから選択して[詳細]ボタンをクリックしてください。
  - キーワードダイアログ (図 3.6-4) が表示されます。
- 3) [リセット] ボタンをクリックしてください。
  - キーワード一覧がデフォルトキーワードにリセットされます。
- 4) [OK] ボタンをクリックしてください。
  - キーワードファイルが変更されます。
- 5) 他の項目の設定がなければ、[OK] ボタンをクリックしてください。
  - 表示機能選択などを行うときは、それらのすべての設定が終わってから [OK] ボタンをクリックしてください。

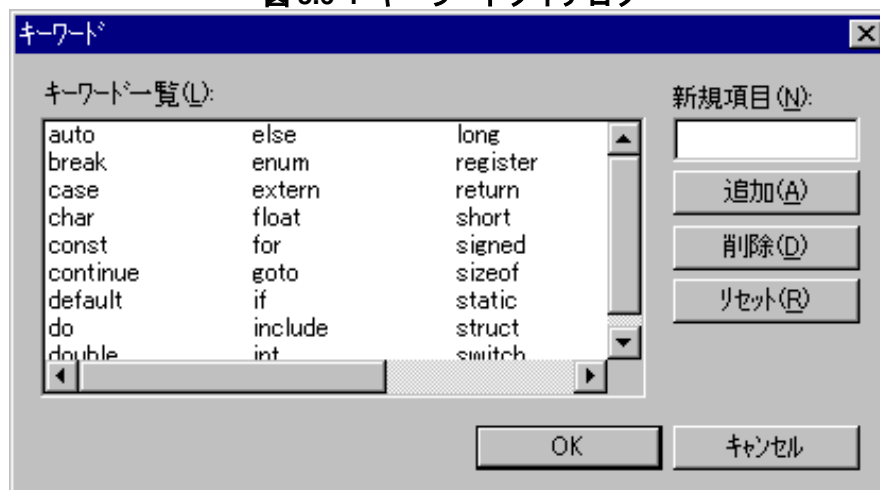
図 3.6-2 エディタ設定ダイアログ



図 3.6-3 エディタ設定ダイアログ (表示 2)



図 3.6-4 キーワードダイアログ



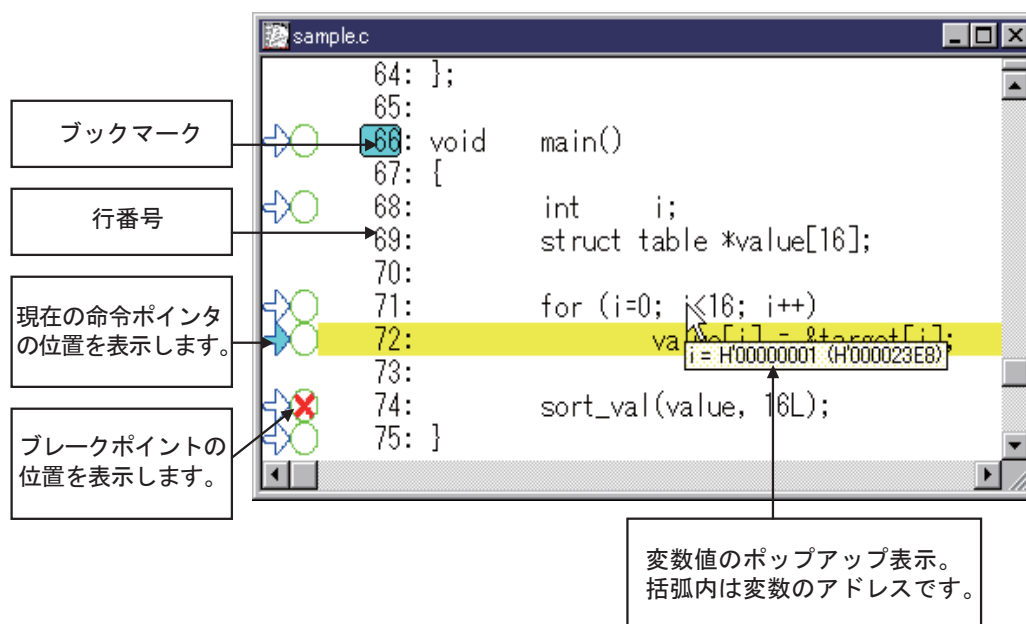
## 3.7 ソースウィンドウ

デバッグ時にソースプログラムを表示するウィンドウです。

### ■ ソースウィンドウ

ソースウィンドウの例を図 3.7-1 に示します。

図 3.7-1 ソースウィンドウの例



### ■ 機能

#### ● ブレークポイントの設定 / 解除

ウィンドウの左側にある " " の " " を左クリックすることにより、ブレークポイントの設定 / 解除を行うことができます。

#### ● 指定行 (アドレス) までの実行

ウィンドウの左側にある " " の " " を左クリックすることにより、その行まで命令実行を行うことができます。PC 位置で行った場合は、ステップ実行 (step/into) を行います。

#### ● 変数値のポップアップ表示

変数名の上にマウスカーソルを静止させることにより、その変数の値をポップアップ表示することができます。

#### ● ブックマーク設定機能

ブックマークを設定すると、検索ツールバーやメニュー、キー操作で瞬時に指定行へ移動することができます。

詳細は「4.3.7 ブックマーク」を参照してください。

### ● ドラッグアンドドロップ

ソースウィンドウにて選択している文字列をドラッグして、メモリ、逆アセンブル、ウォッチウィンドウなどへドロップすることができます。

ドロップ後の動作については各ウィンドウのドラッグアンドドロップの説明を参照してください。

### ● カバレッジ表示

ショートカットメニュー [ カバレッジ表示 ] からソース行単位のカバレッジ表示を行うことができます。混在表示の場合は、機会命令単位のカバレッジ表示を行います。カバレッジ表示を最新の状態にするには、ソースウィンドウの更新を行う必要があります。高速版シミュレータデバッグ時のみ有効となります。

## ■ ショートカットメニュー

- エディット : エディットウィンドウを開きます。
- 更新 : 指定されたウィンドウ内容を更新します。カバレッジ表示が有効な場合には、カバレッジ表示を最新の状態に更新します。高速版シミュレータデバッグ時のみ有効です。
- ラインアセンブル : 「4.4.3 逆アセンブル」を参照してください。
- 検索 : 文字列検索ダイアログを表示します。(図 3.7-2 参照)
- ジャンプ : ジャンプダイアログを表示します。
- カレント PC 移動 : 現在の PC 位置の表示を行います。
- 定義行へ移動 : 関数の定義行へ移動します。
- ブレーク設定・解除 : カーソル位置の行に対して、ブレークポイントの設定または解除を行います。
- ブレーク有効・無効 : カーソル位置の行のブレークポイントを有効または無効にします。
- ブレーク : 「4.6.4 ブレークポイント」を参照してください。
- ウォッチ : ウォッチポイントを追加設定します。
- コールスタック : 「4.6.7 コールスタック」を参照してください。
- カバレッジ設定 : 「4.4.12 カバレッジ」を参照してください。高速版シミュレータデバッグ時のみ有効です。
- プロパティ : ソースファイルのプロパティを表示します。
- カバレッジ表示 : カバレッジ表示の有無を切り替えます。カバレッジ表示が有効な場合には、実行行は緑色で、未実行行は茶色で表示されます。高速版シミュレータデバッグ時のみ有効です。
- 混在表示 : 逆アセンブルリスト表示の有無を切り替えます。
- 停止時にアクティブ : 実行停止時にウィンドウをアクティブにするかどうかを指定します。
- 閉じる : ウィンドウを閉じます。

#### ● エディット

現在のソースファイルを修正するために、エディットウィンドウを開きます。(「3.6 エディットウィンドウ(標準エディタ)」を参照してください。)

#### ● 検索

現在のソースファイル中でテキストの検索を行います。以下の手順で操作してください。

1. ソースウィンドウから右ボタンでショートカットメニューを表示し、[検索]を選択してください。  
検索ダイアログ(図 3.7-2)が開きます。
2. 検索する文字列を設定してください。
3. 検索する方向を設定してください。
4. 必要なら、[大文字と小文字を区別する]にチェックマークを付けてください。
5. [次を検索] ボタンをクリックしてください。  
見つかった箇所が反転表示されます。
6. 検索を終了するときは、[キャンセル] ボタンをクリックしてください。

図 3.7-2 ソースウィンドウ中での検索ダイアログ



#### ● ジャンプ

ソースウィンドウに表示する位置を指定します。以下の手順で操作してください。

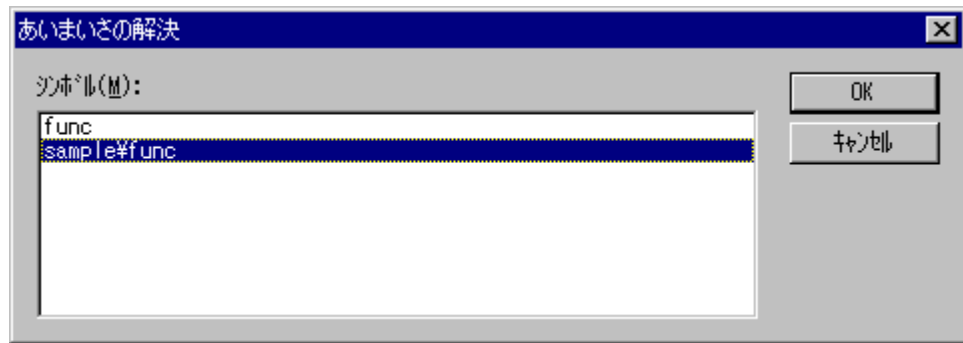
1. ソースウィンドウから右ボタンでショートカットメニューを表示し、[ジャンプ]を選択してください。  
「4.3.6 ジャンプ」を参照してください。
2. 指定形式を選択してください。
3. 表示位置を指定してください。
4. [OK] ボタンをクリックしてください。

#### ● 定義行へ移動

関数の定義行へ移動します。以下の手順で操作してください。

1. ソースウィンドウで関数名を選択または関数名上にカーソルを移動してください。
2. 右ボタンでショートカットメニューを表示し、[定義行へ移動]を選択してください。
3. 同名関数が存在する場合は、[あいまいさの解決] ダイアログが表示されますので、関数名を選択して [OK] ボタンをクリックしてください。

図 3.7-3 あいまいさの解決ダイアログ



## &lt; 注意事項 &gt;

オンデマンドロード使用中（未ロードシンボルが存在する場合）は、定義行にジャンプできないグローバルやスタティック変数もダイアログに表示される場合があります。

## ● ブレーク設定・解除

ブレークポイントを設定または解除したい行にカーソルを移動した後、右ボタンでショートカットメニューを表示し、[ブレーク設定・解除]を選択してください。

カーソル位置の行に対して、ブレークポイントが設定または解除されます。

## ● ブレーク有効・無効

有効または無効にするブレークポイントが設定されている行にカーソルを移動した後、右ボタンでショートカットメニューを表示し、[ブレーク有効・無効]を選択してください。

カーソル位置のブレークポイントの有効 / 無効が切り替わります。

## ● ウォッチ

ウォッチする変数を指定して、ウォッチウィンドウを開きます。以下の手順で操作してください。

1. ソースウィンドウから右ボタンでショートカットメニューを表示し、[ウォッチ]を選択してください。  
「4.4.7 ウォッチ」を参照してください。
2. 変数名を指定してください。
3. 必要に応じて、[モード]を選択してください。
4. [OK] ボタンをクリックしてください。

## ● プロパティ

右ボタンでショートカットメニューを表示し、[プロパティ]を選択してください。

ファイルの情報を表示します。（「4.3.9 プロパティ」を参照してください。）

### ● 混在表示

1. ソースウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
2. [ 混在表示 ] を確認してください。  
[ 混在表示 ] の左にチェックマークがついている場合は、既に混在表示になっています。  
[ 混在表示 ] の左にチェックマークがついていない場合は、混在表示になっていません。
3. [ 混在表示 ] を選択してください。  
既に混在表示になっていた場合は、解除され、チェックマークが消えます。  
混在表示になっていなかった場合は、混在表示になり、チェックマークが付きます。

### ● カバレッジ表示

1. ソースウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
2. [ カバレッジ表示 ] を確認してください。  
[ カバレッジ表示 ] の左にチェックマークがついている場合は、既にカバレッジ表示になっています。  
[ カバレッジ表示 ] の左にチェックマークがついていない場合は、カバレッジ表示になっていません。
3. [ カバレッジ表示 ] を選択してください。  
既にカバレッジ表示になっていた場合は、解除され、チェックマークが消えます。  
カバレッジ表示になっていなかった場合は、カバレッジ表示になり、チェックマークが付きます。

### ● 停止時にアクティブ

実行停止時にソースウィンドウをアクティブにするかどうかを指定します。但しソースウィンドウを新規に表示する場合は常にアクティブにします。以下の手順で操作してください。

1. ソースウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
2. [ 停止時にアクティブ ] を確認してください。  
[ 停止時にアクティブ ] の左にチェックマークがついている場合は、実行停止時にブレークアドレスに対応したソースファイルをアクティブにします。  
ただし実行停止時にブレークアドレスに対応したソースファイルが表示されていない場合は [ 停止時にアクティブ ] の指定にかかわらず、ブレークアドレスに対応するソースウィンドウを開きます。



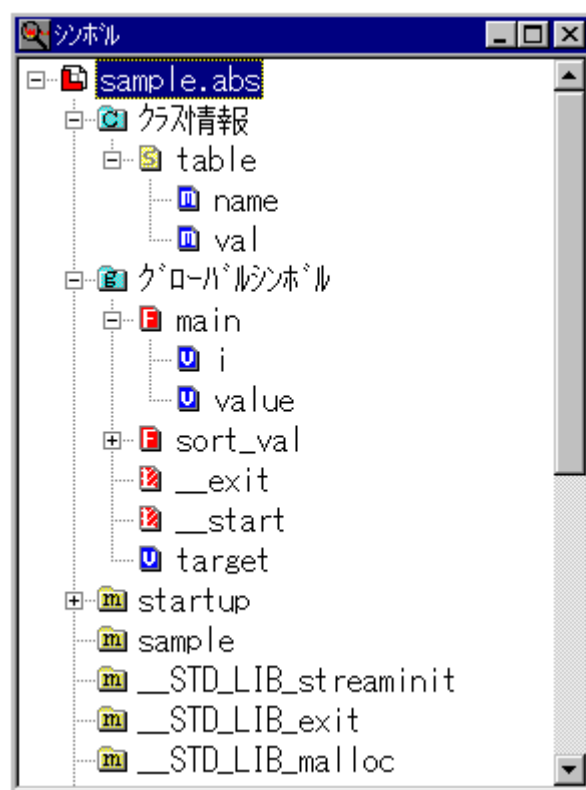
## 3.8 シンボルウィンドウ

現在使用中のプロジェクトのターゲットファイルの名前と、そのファイルで使用されている変数名や関数名などのシンボル名がツリービュー形式で表示されます。

### ■ シンボルウィンドウ

シンボルウィンドウの例を図 3.8-1 に示します。

図 3.8-1 シンボルウィンドウの例



## ■ ショートカットメニュー

メニューには、次の7種類があります。

- ジャンプ : 指定した関数のソースウィンドウを開きます。
- ウォッチ : 指定したシンボルをウォッチポイントとして設定します。
- ブレーク : 「4.6.4 ブレークポイント」を参照してください。
- プロパティ : シンボルのプロパティを表示します。
- マングル名 : シンボル名をマングル名で表示するかどうかを設定します。
- 最新の情報に更新 : 最新のシンボル情報を表示します。
- 閉じる : ウィンドウを閉じます。

### ● ジャンプ

関数名の上で右ボタンをクリックしてショートカットメニューを表示し、[ ジャンプ ] を選択してください。

- その関数の定義されているソースウィンドウが開きます。

### ● ウォッチ

変数名または関数名の上で右ボタンをクリックしてショートカットメニューを表示し、[ ウォッチ ] を選択してください。

- その変数または関数がウォッチに追加されて、ウォッチウィンドウが表示されます。

### ● プロパティ

右ボタンでショートカットメニューを表示し、[ プロパティ ] を選択してください。

- ファイルの情報を表示します ( 「4.3.9 プロパティ」を参照してください ) 。

### ● マングル名

右ボタンでショートカットメニューを表示し、[ マングル名 ] を選択してください。

シンボル名をマングル名で表示する / しないを切り替えます。

### ● 最新の情報に更新

右ボタンでショートカットメニューを表示し、[ 最新の情報に更新 ] を選択してください。

最新のシンボル情報を表示します。

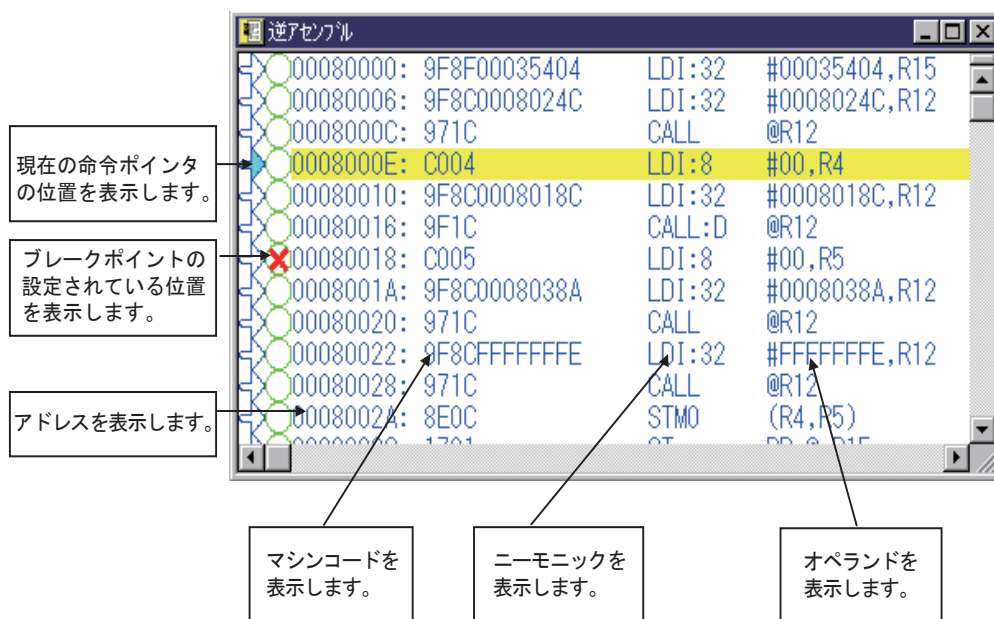
## 3.9 逆アセンブルウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。指定したアドレスから逆アセンブルした結果を表示します。

### ■ 逆アセンブルウィンドウ

逆アセンブルウィンドウの例を図 3.9-1 に示します。

図 3.9-1 逆アセンブルウィンドウの例



### ■ 機能

#### ● ブレークポイントの設定 / 解除

ウィンドウの左側にある " " の " " を左クリックすることにより、ブレークポイントの設定 / 解除を行うことができます。

#### ● 指定行 (アドレス) までの実行

ウィンドウの左側にある " " の " " を左クリックすることにより、その行まで命令実行を行うことができます。PC 位置で行った場合は、ステップ実行 (step/into) を行います。

#### ● ドラッグアンドドロップ

ソースウィンドウから関数名、ラベル、アドレスなどをウィンドウヘドロップするとドロップした文字列のアドレス位置へジャンプします。

#### ● カバレッジ表示

ショートカットメニュー [ カバレッジ表示 ] から機会命令単位のカバレッジ表示を行うことができます。高速版シミュレータデバッガ時のみ有効となります。

## ■ ショートカットメニュー

- ・ ラインアセンブル : 「4.4.3 逆アセンブル」を参照してください。
- ・ ジャンプ : ジャンプダイアログを表示します。
- ・ カレント PC 移動 : 現在の PC 位置の表示を行います。
- ・ ブレーク設定・解除 : カーソル位置の命令に対して、ブレークポイントの設定または解除を行います。
- ・ ブレーク有効・無効 : カーソル位置の命令のブレークポイントを有効または無効にします。
- ・ ブレーク : 「4.6.4 ブレークポイント」を参照してください。
- ・ ウォッチ : 「4.4.7 ウォッチ」を参照してください。
- ・ コールスタック : 「4.6.7 コールスタック」を参照してください。
- ・ カバレッジ設定 : 「4.4.12 カバレッジ」を参照してください。高速版シミュレータデバッグ時のみ有効です。
- ・ カバレッジ表示 : カバレッジ表示の有無を切り替えます。カバレッジ表示が有効な場合には、実行行は緑色で、未実行行は茶色で表示されます。高速版シミュレータデバッグ時のみ有効です。
- ・ 停止時にアクティブ : 実行停止時にウィンドウをアクティブにするかどうかを指定します。
- ・ 閉じる : ウィンドウを閉じます。

### ● ジャンプ

逆アセンブルウィンドウに表示する位置を指定します。以下の手順で操作してください。

- 1) 逆アセンブルウィンドウから右ボタンでショートカットメニューを表示し、[ジャンプ]を選択してください。
  - ジャンプダイアログが開きます（「4.3.6 ジャンプ」を参照してください）。
- 2) 指定形式を選択してください。
- 3) 表示位置を指定してください。
- 4) [OK] ボタンをクリックしてください。

### ● カバレッジ表示

- 1) 逆アセンブルウィンドウで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [カバレッジ表示]を確認してください。

[カバレッジ表示]の左にチェックマークがついている場合は、既にカバレッジ表示になっています。

[カバレッジ表示]の左にチェックマークがついていない場合は、カバレッジ表示になっていません。
- 3) [カバレッジ表示]を選択してください。

既にカバレッジ表示になっていた場合は、解除され、チェックマークが消えます。

カバレッジ表示になっていなかった場合は、カバレッジ表示になり、チェックマークが付きます。

● ブレーク設定・解除

ブレークポイントを設定または解除したい命令にカーソルを移動した後、右ボタンでショートカットメニューを表示し、[ブレーク設定・解除]を選択してください。

カーソル位置の命令に対して、ブレークポイントが設定または解除されます。

● ブレーク有効・無効

有効または無効にするブレークポイントが設定されている命令にカーソルを移動した後、右ボタンでショートカットメニューを表示し、[ブレーク有効・無効]を選択してください。

カーソル位置のブレークポイントの有効/無効が切り替わります。

● 停止時にアクティブ

実行停止時に逆アセンブルウィンドウをアクティブにするかどうかを設定します。ただし、逆アセンブルウィンドウを新規に表示する場合は常にアクティブにします。

以下の手順で操作してください。

- 1) 逆アセンブルウィンドで右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [停止時にアクティブ]を確認してください。

## 3.10 レジスタウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。デバッグ中の MCU レジスタの値の表示と変更を行うためのウィンドウです。

### ■ レジスタウィンドウ

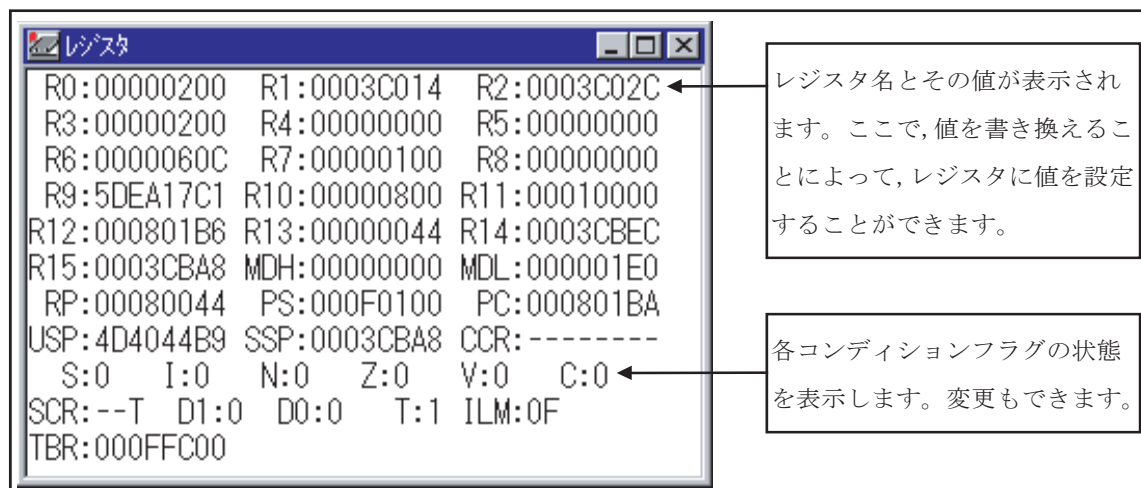
レジスタウィンドウの例を図 3.10-1 に示します。デバッグ時にレジスタ内容を表示するウィンドウです。

レジスタ内容を直接書き換えることができます。レジスタ値に式やシンボルなどで指定して書き換える場合は、ショートカットメニューの「編集」を用いて行います。

ウィンドウサイズに応じてレジスタ配置を行います。

表示するレジスタを選択する場合は、ショートカットメニューの「表示設定」で行います。

図 3.10-1 レジスタウィンドウの例



#### < 注意事項 >

PSR の値によっては、アクセスできなくなるレジスタがあります。このとき "???????" と表示されます。

## ■ ショートカットメニュー

メニューには、次の3種類があります。

- 表示設定 : 「3.10.1 レジスタの表示設定」を参照してください。
- 編集 : レジスタ編集ダイアログを表示します。
- 閉じる : ウィンドウを閉じます。

### ● 編集

レジスタ値の編集を行います。レジスタ値の編集には、この方法以外に、レジスタウィンドウに表示されている数値を直接編集する方法（「4.4.4 レジスタ」を参照してください）もあります。

- 1) レジスタウィンドウで右ボタンをクリックしてショートカットメニューを表示し、  
[編集]を選択してください。
  - レジスタ編集ダイアログが表示されます。
- 2) レジスタ名を選択してください。
- 3) 設定する値を入力してください。
- 4) [OK] ボタンをクリックしてください。

## 3.10.1 レジスタの表示設定

ここでは、レジスタの表示設定について説明します。

### ■ 表示設定

レジスタウィンドウに表示するレジスタの種類を設定します。以下の手順で操作してください。

図 3.10-2 表示レジスタ設定ダイアログ



### ● レジスタの追加

表示するレジスタを追加するには、以下の手順で操作してください。

- 1) レジスタウィンドウで右ボタンをクリックして、ショートカットメニューから「表示設定」を選択してください。
  - 表示設定ダイアログ（図 3.10-2）が開きます。
- 2) 上段の「レジスタ一覧」から表示したいレジスタを選択して、「追加」ボタンをクリックしてください。
  - 選択したレジスタが、下段の「現在の表示レジスタ」に設定されます。
- 3) 表示したいすべてのレジスタを上記の操作で設定してください。
- 4) [OK] ボタンをクリックしてください。



## ● レジスタの削除

表示するレジスタを削除するには、以下の手順で操作してください。

- 1) レジスタウィンドウで右ボタンをクリックして、ショートカットメニューから「表示設定」を選択してください。
  - 表示設定ダイアログ (図 3.10-2) が開きます。
- 2) 下段の「現在の表示レジスタ」から削除したいレジスタを選択して、「削除」ボタンをクリックしてください。
  - 選択したレジスタが、下段の「現在の表示レジスタ」から削除されます。
- 3) 削除したいすべてのレジスタを上記の操作で削除してください。
- 4) [OK] ボタンをクリックしてください。

## ● 初期状態に戻す

表示するレジスタの設定を、SOFTUNE Workbench をインストールしたときの状態に戻す場合は、以下の手順で行ってください。

- 1) レジスタウィンドウで右ボタンをクリックして、ショートカットメニューから「表示設定」を選択してください。
  - 表示設定ダイアログ (図 3.10-2) が開きます。
- 2) 「リセット」ボタンをクリックしてください。
  - 下段の「現在の表示レジスタ」が初期状態に戻ります。
- 3) [OK] ボタンをクリックしてください。

## 3.11 メモリウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。指定したアドレスからのメモリの内容の表示と変更を行うためのウィンドウです。

### ■ メモリウィンドウ

メモリウィンドウの例を図 3.11-1 に示します。デバッグ時にメモリ内容を表示するウィンドウです。

メモリ内容を直接書き換えることができます。メモリ値に式やシンボルなどで指定して書き換える場合は、ショートカットメニューの[編集]で行います。

### ■ 機能

#### ● 変更箇所の表示

実行停止時やモニタリング中に以前の内容から変更になった箇所を赤色で表示します。この機能はメモリウィンドウで現在表示されている範囲内でのみ有効となります。

#### ● 16 バイト固定表示

1 行に表示するバイト数をウィンドウサイズに応じた[自動]と固定表示の[4 バイト]、[8 バイト]、[16 バイト]、[32 バイト]、[64 バイト]から選択することができます。この設定はショートカットメニューの[表示設定]で行います。

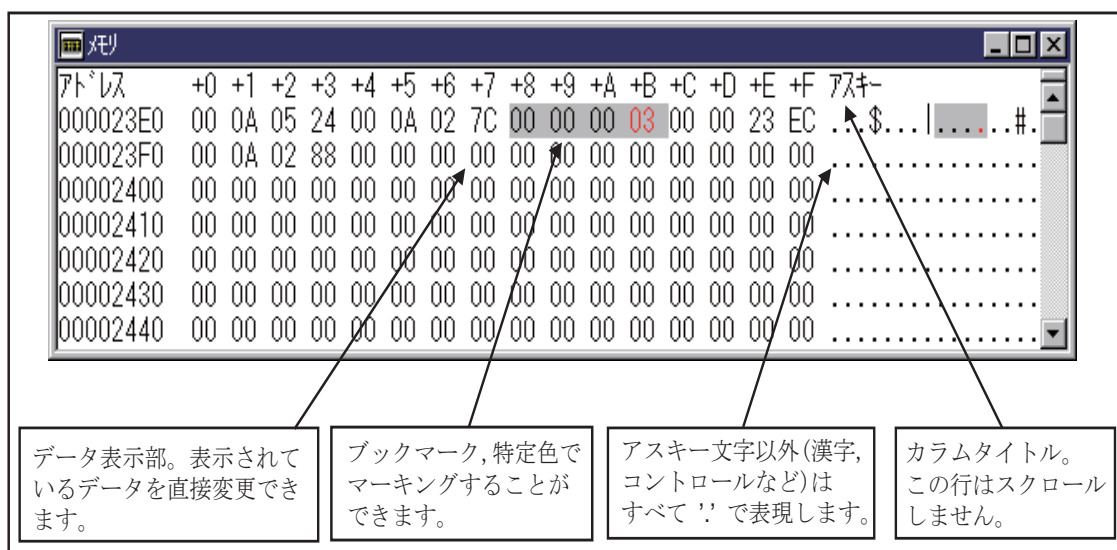
#### ● ブックマーク設定機能

指定したアドレス範囲に色を付けてブックマークを設定できます。ブックマークを設定すると、検察ツールバーやメニュー、キー操作で瞬時に指定行へ移動することができます。詳細は「4.3.7 ブックマーク」を参照してください。

#### ● ドラッグアンドドロップ

ソースウィンドウから変数名、関数名、ラベル、アドレスなどをウィンドウヘドロップすると、ドロップした文字列のアドレスへ移動し、シンボルのアドレス範囲が選択されます。

図 3.11-1 メモリウィンドウの例



## ■ ショートカットメニュー

メニューには、次の 14 種類があります。

- 比較 : 「4.4.5 メモリ」を参照してください。
- 検索 : 「4.4.5 メモリ」を参照してください。
- 特殊操作 : 「4.4.5 メモリ」を参照してください。
- ラインアセンブル : 「4.4.3 逆アセンブル」を参照してください。
- ジャンプ : 「4.3.6 ジャンプ」を参照してください。
- ブックマーク : 「4.4.5 メモリ」を参照してください。
- 編集 : 「4.4.5 メモリ」を参照してください。
- 表示設定 : 「4.4.5 メモリ」を参照してください。
- アクセス禁止領域 : デバッグ環境設定ダイアログの「アクセス禁止領域」タブを呼び出します。
- ブレーク : 「4.6.4 ブレークポイント」を参照してください。
- イベント : 「4.6.5 イベント」を参照してください。
- ウォッチ : 「4.4.7 ウォッチ」を参照してください。
- モニタリング : モニタリングの有効 / 無効を切り替えます。
- 閉じる : ウィンドウを閉じます。

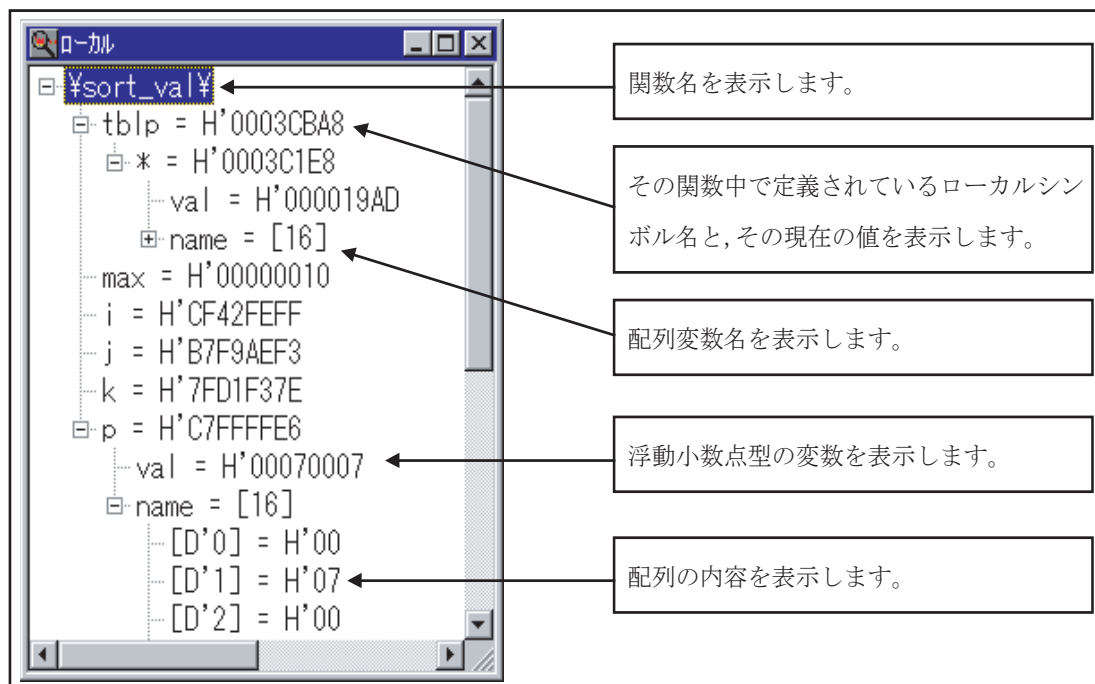
## 3.12 ローカル変数ウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。ローカル変数の内容の表示と変更を行うためのウィンドウです。ローカル変数は関数名をルートにしたツリー形式で表示されます。

### ■ ローカル変数ウィンドウ

ローカル変数ウィンドウの例を図 3.12-1 に示します。

図 3.12-1 ローカルウィンドウの例



### ■ ショートカットメニュー

メニューには、次の 5 種類があります。

- 基数 : 「4.4.6 ローカル変数」を参照してください。
- 編集 : 「4.4.6 ローカル変数」を参照してください。
- メモリウィンドウ : メモリウィンドウの表示とブックマークの設定を行います。
- プロパティ : シンボルのプロパティを表示します。
- 閉じる : ウィンドウを閉じます。

#### ● メモリウィンドウ

選択すると、次の 2 種類のサブメニューが表示されます。

- ジャンプ  
選択している変数のアドレスでメモリウィンドウを表示します。
- ブックマーク追加  
選択している変数のアドレスでメモリウィンドウに対してブックマークの追加を行います。詳しくは「4.3.7 ブックマーク」を参照してください。

- プロパティ

その変数についての情報が表示されます。詳細は、「4.3.9 プロパティ」を参照してください。

## 3.13 ウォッチウィンドウ

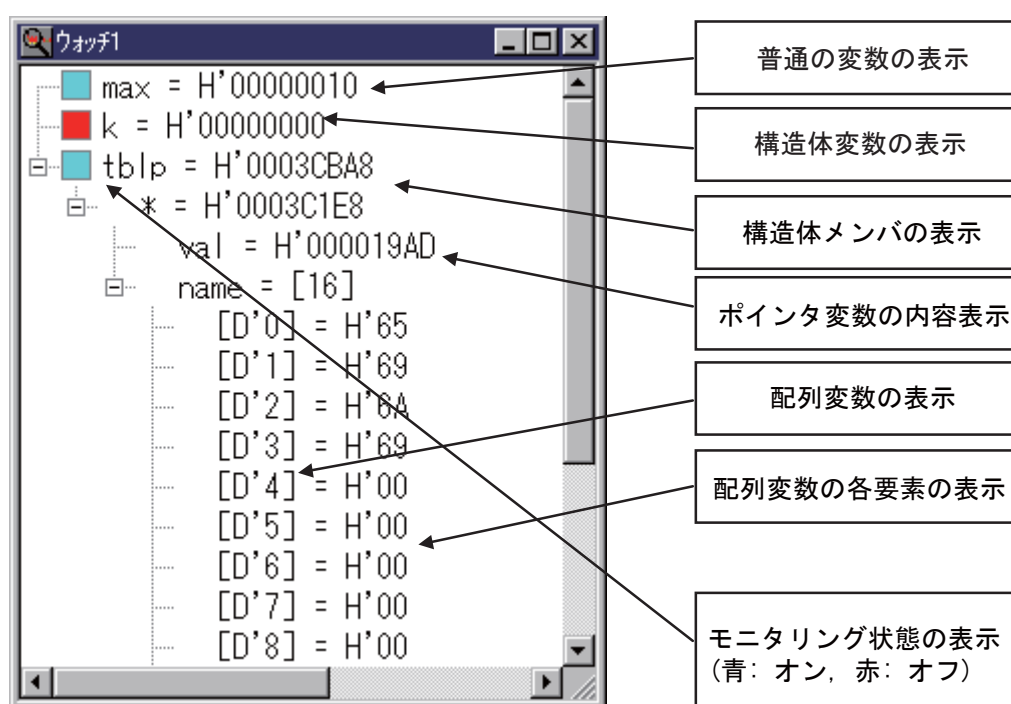
このウィンドウはデバッグセッションでのみ表示されます。指定した変数の値の表示と変更を行うためのウィンドウです。変数はツリー形式で表示されます。

### ■ ウォッチウィンドウ

ウォッチウィンドウは独立した4つのウィンドウを開くことができます。ウィンドウごとに変数を登録できるため、登録時にウィンドウを指定します。

ウォッチウィンドウの例を図3.13-1に示します。

図 3.13-1 ウォッチウィンドウの例



### ■ 機能

#### ● ドラッグアンドドロップ

##### 1. ウィンドウ内の変数の並べ替え

ウィンドウに複数個の変数を登録してある場合、順番を変更したい変数をドラッグし、指定の位置にドロップすると、順番を入れ替えることが可能です。

##### 2. ウォッチウィンドウ間の連携

登録中の変数をドラッグし、別のウォッチウィンドウにドロップすると、変数のコピー/移動が可能です。構造体のメンバをドラッグした場合は、メンバのみ別ウィンドウにコピーされ、ドラッグ元の構造体は削除されずに残ります。

### ● 変数の複数選択

ウィンドウに登録済の変数を複数個同時に選択し、変数の一括操作が可能です。可能な操作は下記のとおりです。


- ドラッグ アンド ドロップによるコピー / 移動
- 基数の設定
- 削除
- モニタリングのオン / オフ設定


### ● リアルタイム領域との連携


ウォッチ変数のアドレスを簡単にリアルタイム領域に設定できます。ウォッチ変数をリアルタイムでモニタリングする場合に便利です。


変数の状態により、下記のようにウィンドウに表示されます。


ウォッチ変数がグローバル変数、静的変数の場合


 : 変数のアドレスがリアルタイム領域外（個別設定：オン）

 : 変数のアドレスがリアルタイム領域外（個別設定：オフ）

 : 変数のアドレスがリアルタイム領域 1 内（個別設定：オン）


 : 変数のアドレスがリアルタイム領域 1 内（個別設定：オフ）


 : 変数のアドレスがリアルタイム領域 2 内（個別設定：オン）

 : 変数のアドレスがリアルタイム領域 2 内（個別設定：オフ）

ウォッチ変数がグローバル変数、静的変数以外の場合

またはアクティブでない場合

 : (個別設定：オン)

 : (個別設定：オフ)

## ■ ショートカットメニュー

メニューには、次の12種類があります。

- 基数 : 「4.4.7 ウォッチ」を参照してください。  
変数を複数個選択している場合は一括で設定します。
- 設定 : 「4.4.7 ウォッチ」を参照してください。
- 要素数 : 「4.4.7 ウォッチ」を参照してください。
- 編集 : 「4.4.7 ウォッチ」を参照してください。
- 削除 : 「4.4.7 ウォッチ」を参照してください。  
変数を複数個選択している場合は一括で削除します。
- 全削除 : 現在登録中の変数をウィンドウから全て削除します。
- アクセス禁止領域 : 「環境」メニューの「デバッグ環境」の「アクセス禁止領域」タブを呼び出します。
- モニタリング : モニタリングの有効/無効を切り替えます。
- 個別設定 : 登録されている変数ごとに、モニタリングのオン/オフを設定します。設定状態は四角いICONの色で区別します。オンは青色、オフは赤色となっています。
- リアルタイム : 登録済のウォッチ変数をリアルタイム領域に設定します。リアルタイムメモリ機能がない品種の場合は無効となります。詳細は「4.4.7 ウォッチ」を参照してください。
- メモリウィンドウ : メモリウィンドウの表示とブックマークの設定を行います。
- プロパティ : ウォッチのプロパティを表示します。
- 閉じる : ウィンドウを閉じます。

### ● メモリウィンドウ

選択すると、次の2種類のサブメニューが表示されます。

- ジャンプ  
選択している変数のアドレスでメモリウィンドウを表示します。
- ブックマーク追加  
選択している変数のアドレスでメモリウィンドウに対してブックマークの追加を行います。詳しくは「4.3.7 ブックマーク」を参照してください。

### ● プロパティ

- 1) 表示されている変数名上で右ボタンをクリックしてショートカットメニューを表示して、[ プロパティ ] を選択してください。  
その変数についての情報が表示されます。詳細は、「4.3.9 プロパティ」を参照してください。
- 2) [ 閉じる ] ボタンをクリックしてください。



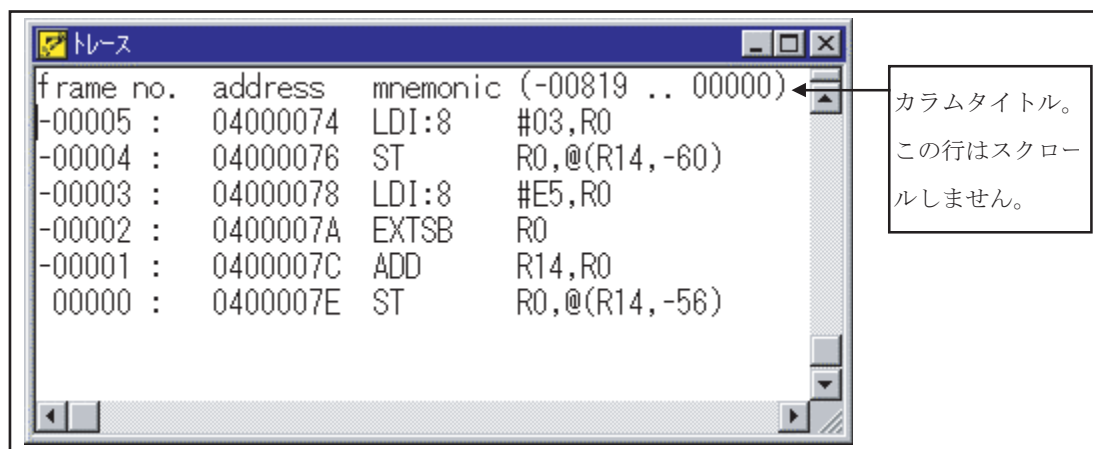
## 3.14 トレースウィンドウ

ここでは、トレースウィンドウについて説明します。このウィンドウはデバッグセッションでのみ表示されます。

### ■ トレースウィンドウ

トレースの結果を表示するウィンドウがトレースウィンドウです。トレース機能については、「4.4.8 トレース」を参照してください。トレースウィンドウの例を図 3.14-1 に示します。

図 3.14-1 トレースウィンドウの例



## ■ ショートカットメニュー

メニューには、次の13種類があります。

- 更新 : トレースウィンドウの表示を更新します。
- ジャンプ : 「4.3.6 ジャンプ」を参照してください。
- バックトレース : 「4.4.8 トレース」を参照してください。
- インストラクション : インストラクションモードで表示します (エミュレータデバッグ時に有効)。
- サイクル : サイクルモードで表示します (エミュレータ, シミュレータデバッグ時に有効)。
- ソース : ソースモードで表示します (エミュレータ, シミュレータデバッグ時に有効)。
- タスク : タスクモードで表示します。
- 設定 : 「4.4.8 トレース」を参照してください。
- 詳細 : DSU3 チップでのみ使用できます。「4.4.8 トレース」を参照してください。
- 検索 : 「4.4.8 トレース」を参照してください。
- セーブ : 「4.4.8 トレース」を参照してください。
- クリア : 表示モードのトレースバッファをクリアします。
- 閉じる : ウィンドウを閉じます。

### ● インストラクション/サイクル/ソース

DSU3 チップでは、本選択はできません。

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
  - 2) [インストラクション], [サイクル], [ソース]のいずれかを選択してください。
- 選択された表示モードで表示し、選択されたサブメニューにチェックマークが付きます。

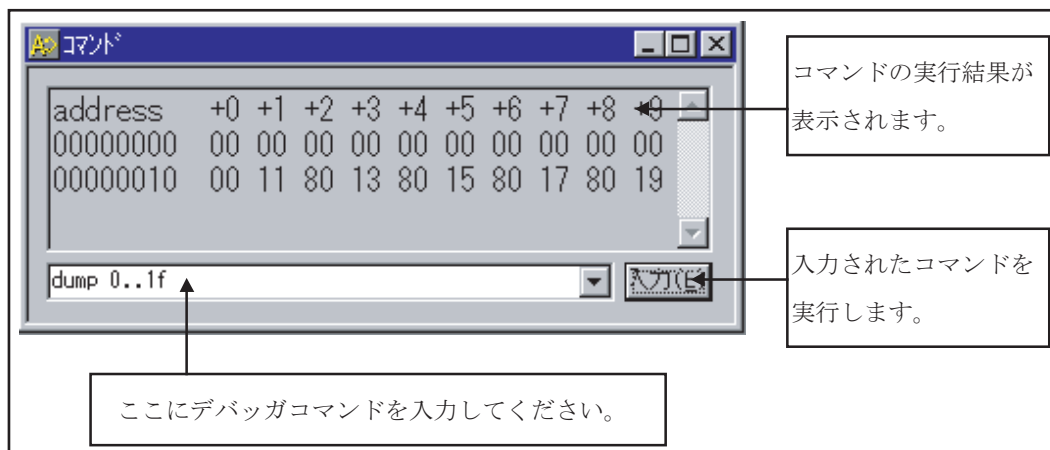
## 3.15 コマンドウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。デバッガのコマンドを直接入力して結果を表示させるためのウィンドウです。

### ■ コマンドウィンドウ

コマンドウィンドウの例を図 3.15-1 に示します。

図 3.15-1 コマンドウィンドウの例



## ■ ショートカットメニュー ( コマンド入力フィールド )

メニューには、次の6種類があります。

- 元に戻す : 直前の編集を取り消します。
- 切り取り : 選択した文字列を、クリップボードに移動します。
- コピー : 選択した文字列を、クリップボードに複写します。
- 貼り付け : クリップボードの文字列を貼り付けます。
- 削除 : 選択した文字列を削除します。
- すべて選択 : 入力されている文字列をすべて選択します。

### ● 元に戻す

- 1) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 2) [ 元に戻す ] を選択してください。

### ● 切り取り

- 1) 切り取る文字列を選択してください。
- 2) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 3) [ 切り取り ] を選択してください。

### ● コピー

- 1) コピーする文字列を選択してください。
- 2) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 3) [ コピー ] を選択してください。

### ● 貼り付け

- 1) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 2) [ 貼り付け ] を選択してください。

### ● 削除

- 1) 削除する文字列を選択してください。
- 2) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 3) [ 削除 ] を選択してください。

### ● すべて選択

- 1) コマンド入力フィールドで右ボタンをクリックして、ショートカットメニューを表示してください。
- 2) [ すべて選択 ] を選択してください。

## ■ ショートカットメニュー ( コマンド入力フィールド以外 )

メニューには、次の5種類があります。

- バッチファイル指定 : バッチファイルを指定するファイルダイアログを表示します。
- エイリアスファイル指定 : エイリアスファイルを指定するファイルダイアログを表示します。
- 文字列置換設定 : コマンド置換ダイアログを表示します。
- ロギング : ログの開始 / 状態表示 / 終了を設定します。
- 閉じる : ウィンドウを閉じます。

### ● バッチファイル指定

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ バッチファイル指定 ] を選択してください。
  - [ ファイルを開く ] ファイルダイアログが表示されます。
- 3) バッチファイルを選択して、[ 開く ] ボタンをクリックしてください。

### ● エイリアスファイル指定

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ エイリアスファイル指定 ] を選択してください。
  - [ ファイルを開く ] ファイルダイアログが表示されます。
- 3) エイリアスファイルを選択して、[ 開く ] ボタンをクリックしてください。

### ● 文字列置換設定

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - [ コマンド置換 ] ダイアログが表示されます。詳細は、「3.15.1 文字列置換設定」を参照してください。

### ● ロギング

ロギングについては、「3.15.2 ロギング」を参照してください。

### ● 閉じる

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [ 閉じる ] を選択してください。

### 3.15.1 文字列置換設定

---

ここでは、コマンドウィンドウの文字列置換設定について説明します。

---

#### ■ 文字列置換設定

コマンドウィンドウで使用するコマンドの別名、変数の定義、マクロ名の定義が行えます。

#### ■ エイリアス

よく使うコマンドに名前を付けて、その名前でコマンドを起動することができます。設定および削除は以下の手順で行ってください。

##### ● 設定

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) 別名を設定してください。
  - コマンド名や既に登録されている別名と衝突しない名前を指定してください。
- 4) その名前に割り当てるコマンドを記述してください。
  - パラメータも含めて記述できます。
- 5) [ 追加 ] ボタンをクリックしてください。

##### ● 削除

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) エイリアス一覧で削除する別名を選択してください。
- 4) [ 削除 ] ボタンをクリックしてください。

図 3.15-2 文字列置換設定のダイアログ



## ■ 変数

よく参照するアドレス範囲などに名前を付けて、その名前をコマンド入力に使用することができます。設定および削除は以下の手順で行ってください。

### ● 設定

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) [ 変数 ] タグをクリックしてください。
  - 変数設定のダイアログ ( 図 3.15-3 ) が表示されます。
- 4) 変数を設定してください。
  - 既に登録されている変数名と衝突しない名前を指定してください。
- 5) その名前に割り当てる文字列を記述してください。
- 6) [ 追加 ] ボタンをクリックしてください。

### ● 削除

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) [ 変数 ] タグをクリックしてください。
  - 変数設定のダイアログ ( 図 3.15-3 ) が表示されます。
- 4) デバッグ変数一覧で削除する変数名を選択してください。
- 5) [ 削除 ] ボタンをクリックしてください。

図 3.15-3 変数設定のダイアログ



## ■ マクロ

よく使用する特定のコマンドの組み合わせに名前を付けて、それをマクロとして使用することができます。設定および削除は以下の手順で行ってください。

### ● 設定

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) [ マクロ ] タグをクリックしてください。
  - マクロ設定のダイアログ ( 図 3.15-4 ) が表示されます。
- 4) マクロ名を設定してください。
  - 既に登録されているマクロ名と衝突しない名前を指定してください。
- 5) 名前に割り当てるコマンドを記述してください。
  - 別名, 変数, ほかのマクロも記述することができます。
- 6) [ 追加 ] ボタンをクリックしてください。



## ● 削除

- 1) コマンドウィンドウ上で右ボタンをクリックし、ショートカットメニューを表示してください。
- 2) [ 文字列置換設定 ] を選択してください。
  - 文字列置換設定のダイアログ ( 図 3.15-2 ) が表示されます。
- 3) [ マクロ ] タグをクリックしてください。
  - マクロ設定のダイアログ ( 図 3.15-4 ) が表示されます。
- 4) マクロ一覧で削除するマクロ名を選択してください。
- 5) [ 削除 ] ボタンをクリックしてください。

図 3.15-4 マクロ設定のダイアログ



## 3.15.2 ロギング

ここでは、SOFTUNE Workbench でのロギングについて説明します。

### ■ ロギング

コマンドウィンドウでのコマンド実行のログをファイルに記録することをロギングといいます。ロギングの制御は、コマンドウィンドウで右ボタンをクリックして、ショートカットメニューを表示し、[ロギング]を選択することで行います。

ショートカットメニューで[ロギング]を選択すると、次のサブメニューが表示されます。

- 開始
- 状態
- 終了

#### ● 開始

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
  - 2) [開始]を選択してください。
- ロギングファイルの選択のためのファイルダイアログが開きます。
- 3) ロギングファイル名を指定して、[保存]ボタンをクリックしてください。

#### ● 状態

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
  - 2) [状態]を選択してください。
- ロギング状態を表示するダイアログ (図 3.15-5) が開きます。
- 3) 必要なら、状態および出力メッセージ制御の変更を行ってください。
  - 4) [OK] ボタンをクリックしてください。

#### ● 終了

- 1) 右ボタンをクリックしてショートカットメニューを表示してください。
- 2) [終了]を選択してください。

図 3.15-5 ロギングの状態を表示するダイアログ



### 3.16 リアルタイムメモリウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。リアルタイム領域内の指定したアドレスからミラーメモリの内容を表示するためのウィンドウです。

■ リアルタイムメモリウィンドウ

MB2198 エミュレータのリアルタイムメモリウィンドウの例を図 3.16-1 に示します。デバッグ時に デバッグ環境設定ダイアログ の[リアルタイム領域]タブで指定されたミラーメモリ内容を表示するウィンドウです。

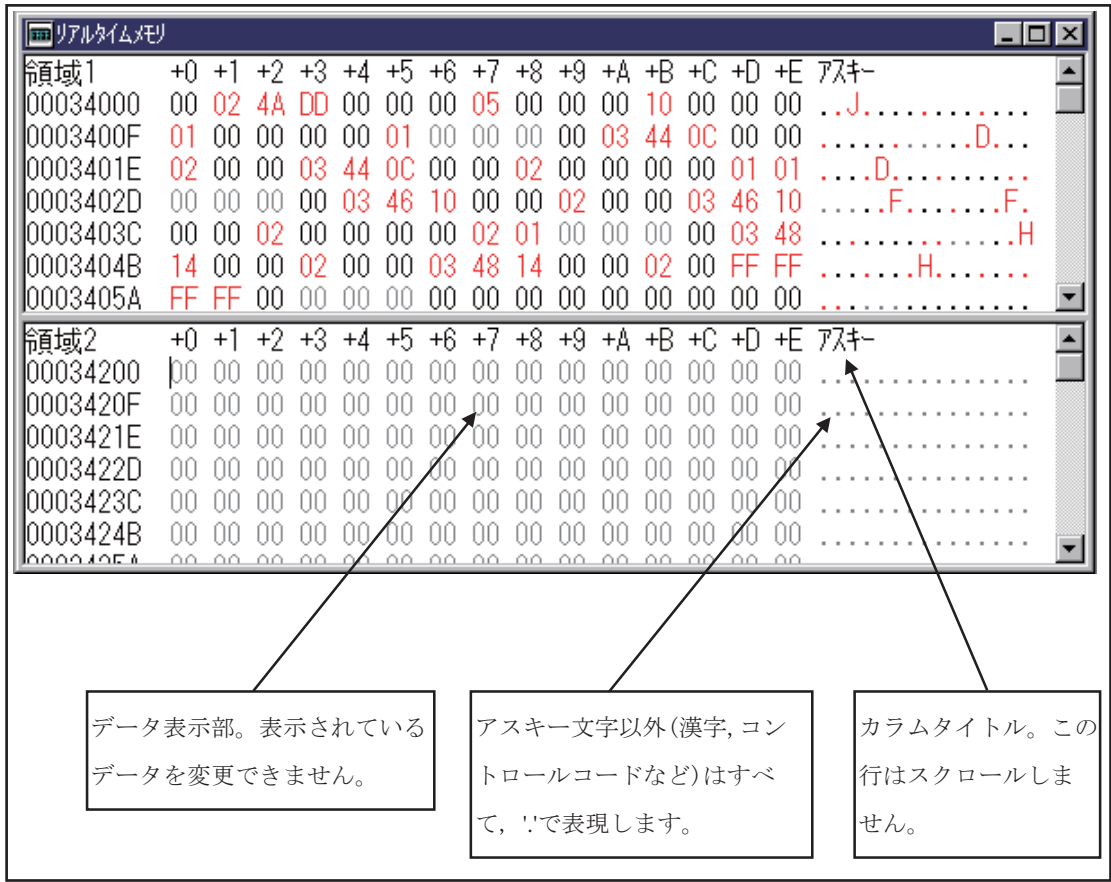
メモリ内容を直接書き換えることはできません。表示領域を変更したい場合は、ショートカットメニューの[領域]で行います。

■ 機能

● ドラッグアンドドロップ

ソースウィンドウから変数名、関数名、ラベル、アドレスなどをウィンドウヘドロップするとドロップした文字列のアドレスへジャンプします。

図 3.16-1 リアルタイムメモリウィンドウの例



## ■ ショートカットメニュー

- 領域 : デバッグ環境設定ダイアログの [リアルタイム領域] タブで指定された領域の先頭から表示します。
- 表示設定 : 「4.4.13 リアルタイムメモリ」を参照してください。
- 領域指定 : デバッグ環境設定ダイアログの [リアルタイム領域] タブを呼び出します。
- モニタリング : モニタリングの有効 / 無効を切り替えます。
- 閉じる : ウィンドウを閉じます。

---

### < 注意事項 >

- この機能はエミュレータまたはその接続形態により使用できない場合があります。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.9 リアルタイムモニタ」を参照してください。
  - FR80S 使用時に内蔵 RAM 領域以外にリアルタイム領域を設定した場合、データが更新されません。
-

### 3.17 カバレッジウィンドウ

このウィンドウはデバッグセッションでのみ表示されます。カバレッジ測定結果を表示するためのウィンドウです。

#### ■ カバレッジウィンドウ

カバレッジウィンドウの例を図 3.17-1 に示します。デバッグ時にカバレッジ測定結果を表示するウィンドウです。このウィンドウは高速版シミュレータデバッガで表示されます。

カバレッジ機能の詳細については「4.4.12 カバレッジ」を参照してください。

図 3.17-1 カバレッジウィンドウ (16 アドレス単位)

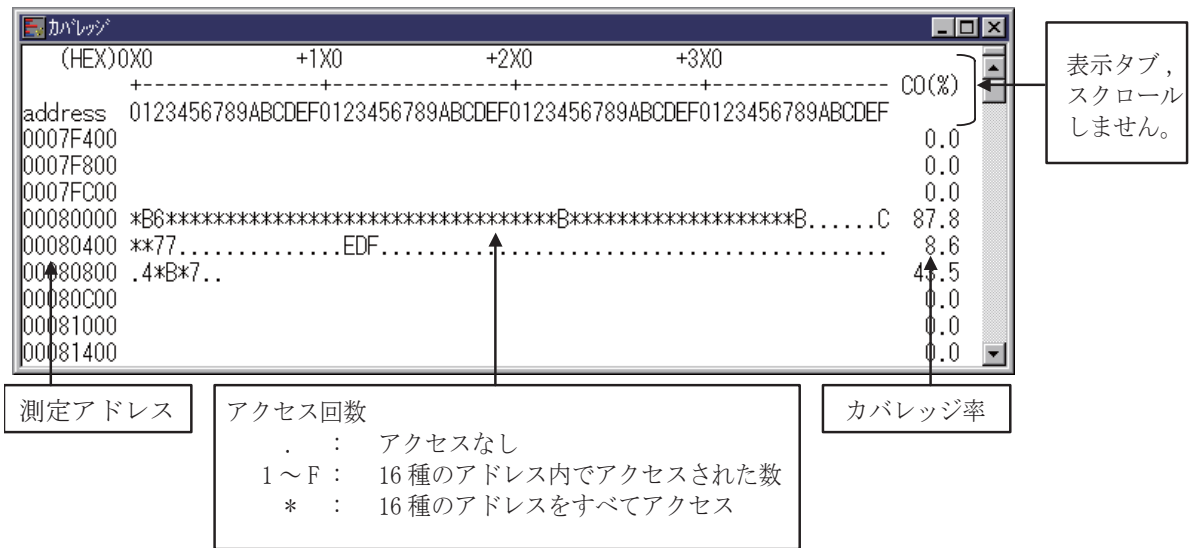


図 3.17-2 カバレッジウィンドウ (1 アドレス単位)



### ■ 機能

#### ● ドラッグアンドドロップ

ソースウィンドウから変数名, 関数名, ラベル, アドレスなどをウィンドウヘドロップするとドロップした文字列のアドレスへジャンプします。

### ■ ショートカットメニュー

- 更新 : カバレッジウィンドウの表示を更新します。
- ジャンプ : ジャンプダイアログを開きます。
- ソースウィンドウ : カーソル位置のアドレスを元にソースウィンドウを表示します。アドレスに対応する行番号が存在しない場合は, 逆アセンブルウィンドウを表示します。
- 16 アドレス単位 : 16 アドレス単位で表示します。
- 1 アドレス単位 : 1 アドレス単位で表示します。
- 設定 : 「4.4.12 カバレッジ」を参照してください。
- 全カバレッジ率 : 「4.4.12 カバレッジ」を参照してください。
- クリア : カバレッジ測定バッファをクリアします。
- 閉じる : ウィンドウを閉じます。

## 3.18 パフォーマンスウィンドウ

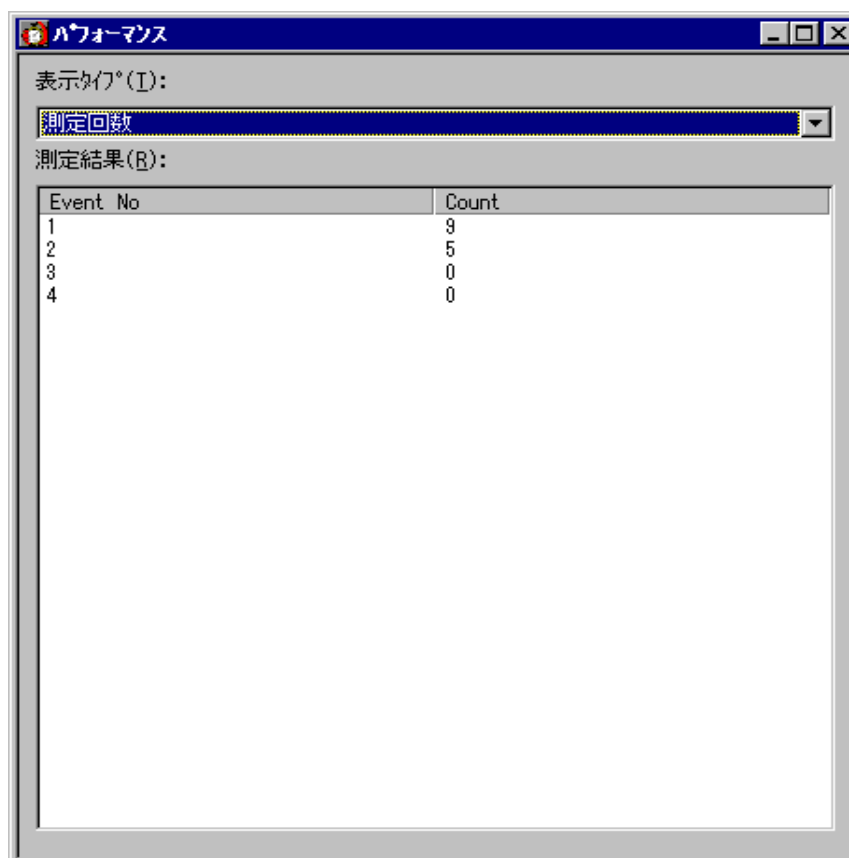
パフォーマンス測定結果を表示するためのウィンドウです。

### ■ パフォーマンスウィンドウ

パフォーマンス機能については、「4.4.14 パフォーマンス」を参照してください。

パフォーマンスウィンドウの例を図 3.18-1 および図 3.18-2 に示します。デバッグ時にパフォーマンス測定結果を表示するウィンドウです。このウィンドウは、エミュレータデバッガでイベントモードがパフォーマンスモードのとき表示されます。

図 3.18-1 パフォーマンスウィンドウ (測定回数)



The screenshot shows a window titled 'パフォーマンス' (Performance). Inside, there is a section '表示項目(I):' (Display Item(I):) with a dropdown menu set to '測定回数' (Measurement Count). Below this is a section '測定結果(R):' (Measurement Result(R):) containing a table with two columns: 'Event No' and 'Count'.

Event No	Count
1	9
2	5
3	0
4	0

図 3.18-2 パフォーマンスウィンドウ



- 表示タイプ : 測定時間で固定です。
- 表示イベント : 測定区間を選択します。
- 測定結果 : 測定結果を集計して表示します。  
最小, 最大, 平均は測定時間時が表示されます。

## ■ ショートカットメニュー

- 更新 : パフォーマンスウィンドウの表示を更新します。
- 設定 : 「4.4.14 パフォーマンス」を参照してください。
- 表示範囲 : 「4.4.14 パフォーマンス」を参照してください。
- クリア : パフォーマンスバッファをクリアします。
- コピー : 「4.4.14 パフォーマンス」を参照してください。
- 閉じる : ウィンドウを閉じます。

### < 注意事項 >

1. FR60Lite または FR80S 使用時以外では, この機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.8 パフォーマンス測定」を参照してください。
2. イベントモードがトレースモードの場合は使用できません。



## 3.19 RAM チェッカウィンドウ

このウィンドウはデバッグセッションで表示できます。ロギング状態の表示とモニタリング表示を行います。

### ■ RAM チェッカウィンドウ

メインメニュー[表示]-[RAM チェッカ]を選択すると、RAM チェッカウィンドウが開きます。

図 3.19-1 RAM チェッカメニュー選択

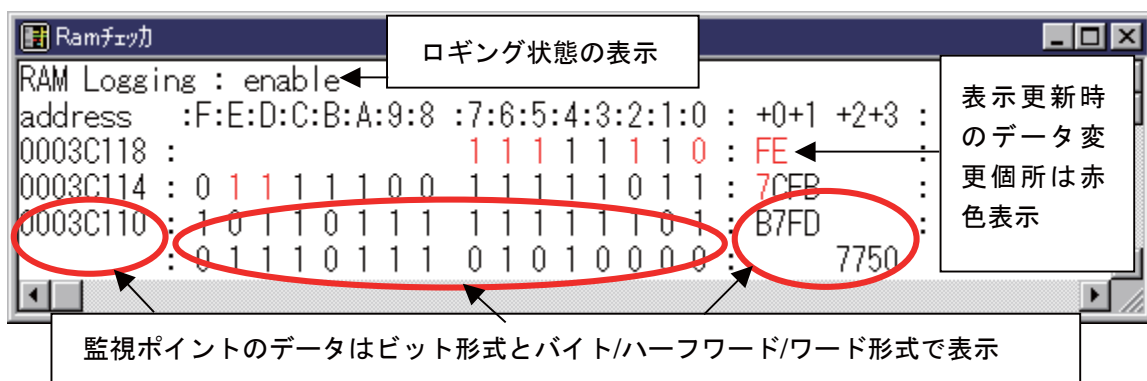


### ■ 機能

#### ● ドラッグ アンド ドロップ

ウィンドウヘドロップされた文字列を監視アドレスとして RAM チェッカウィンドウに登録します。

図 3.19-2 RAM チェッカウィンドウ



RAM チェッカウィンドウは、ロギング状態の表示とモニタリング表示を行います。

- ロギング状態が Disable だとモニタリングも無効になります。
- モニタリング表示のデータ更新間隔は 100ms 固定です。
- データ形式は、ビット形式とバイトまたはワード形式の 2 通りで表示します。
- データ内容の表示は、前回の更新からメモリ内容に変化のあったデータを赤色表示します。
- メモリ内容の編集はできません。

---

#### < 注意事項 >

- FR60Lite または FR80S 使用時以外では、この機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.12 RAM チェッカ」を参照してください。
  - RAM チェッカウィンドウのロギング状態を Enable で実行すると、他ウィンドウ(メモリ、ウォッチなど)のモニタリング表示は、設定状態にかかわらず無効になります。
  - RAM チェッカウィンドウを使用する場合、デバッグモードを RAM Checker モードに設定してください。デバッグモードが RAM Checker モード以外に設定されている場合、RAM チェッカウィンドウを開くことができません。  
RAM チェッカウィンドウを開いた状態でデバッグモードを RAM Checker モード以外にした場合、RAM チェッカウィンドウは自動的に閉じます。  
デバッグモードの設定は、メニュー [ 環境 ]-[ デバッグ環境の設定 ]-[ デバッグ機能の選択 ]から行ってください。RAM Checker モードに設定できない場合は、ご使用のエミュレータの環境では RAM チェッカ機能を使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.12 RAM チェッカ」を参照してください。
  - イベント機能の使用方法に関しては、『SOFTUNE Workbench 機能説明書』「2.3.7 トレース」、『SOFTUNE Workbench 操作マニュアル』「4.4.8 トレース」を参照してください。
-

### 3.19.1 RAM チェッカの設定

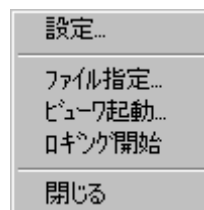
RAM チェッカの設定について説明します。

#### ■ 設定

各設定は RAM チェッカウィンドウのショートカットメニュー，またはコマンド入力によりおこないます。ここではショートカットメニューによる設定について説明します。コマンドによる設定は、『SOFTUNE Workbench コマンドリファレンスマニュアル』を参照してください。

#### ■ ショートカットメニュー

図 3.19-3 RAM チェッカウィンドウのショートカットメニュー



#### ■ 監視アドレス

ショートカットメニュー [設定 ...] により設定ダイアログを開きます。

図 3.19-4 設定ダイアログ



設定した監視アドレスは、チェックアドレスリストの表示順で RAM チェッカウィンドウに表示されます。新たに監視アドレスを設定するとリストの最後に追加されます。リストの順番は変更可能です。

サンプリング間隔 (1ms)、モニタリング間隔 (100ms) は変更できません。

### 1. 監視アドレスの設定手順

- 監視アドレスをシンボルまたはアドレスで入力してください。
- 監視アドレスのデータサイズを " バイト " または " ハーフワード ", " ワード " から選択してください。
- 監視アドレスのアクセス属性を設定してください。ただし FR80S では指定できず、必ずライト属性となります。
- " 追加 " ボタンを押すと、リストに登録されます。
- "OK" ボタンを押すと、設定ダイアログの内容が RAM チェッカウィンドウに反映されます。

### 2. 監視アドレスの削除

- 削除する監視アドレスをリストから選択してください (複数選択可)。
- " 削除 " ボタンを押すと、リストから削除されます。
- "OK" ボタンを押すと、設定ダイアログの内容が RAM チェッカウィンドウに反映されます。

### 3. 表示位置の変更

- 変更したい監視アドレスをリストから選択してください (複数選択不可)。
- " 上へ " または " 下へ " ボタンで表示位置を選択してください。
- "OK" ボタンを押すと、設定ダイアログの内容が RAM チェッカウィンドウに反映されます。

---

#### < 注意事項 >

- FR60Lite または FR80S 使用時以外では、この機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.12 RAM チェッカ」を参照してください。
  - FR80S では監視アドレスを内蔵 RAM 領域に設定した場合のみ有効となります。
  - データサイズがハーフワードの場合、監視アドレスの下位 1 ビットは無視されます。データサイズがワードの場合、監視アドレスの下位 2 ビットは無視されます。
-

## ■ ログファイル

ショートカットメニュー [ ファイル指定 ... ] によりファイル設定ダイアログを開きます。

図 3.19-5 ファイル設定ダイアログ



サンプリングデータを保存するログファイル，データの保存形式などを指定してください。

### 1. ファイル名

保存するログファイル名を指定してください。拡張子を省略した場合，選択したファイル形式に合わせて，自動的に拡張子が付加されます。

SOFTUNE 形式の場合，".SRL" が付加されます。

CSV 形式の場合，".CSV" が付加されます。

### 2. ファイル形式

保存ファイル形式を，SOFTUNE 形式と CSV 形式から選択してください。

SOFTUNE 形式：RAM Checker Viewer で表示する場合（推奨）

CSV 形式：RAM Checker Viewer 以外で表示する場合（\*）

---

### < 注意事項 >

CSV 形式は必要とするデータのサイズが SOFTUNE 形式と比較して約 4 倍になります。

---

### 3. ファイルの上書き制御

一度作成されたログファイルを上書きせずにデータ取得させる機能です。

この機能を有効にするとログファイルを自動で別名保存します。

チェックボックスをチェックすると有効になります。

動作例

指定されたログファイル ( filename.srl ) が存在した場合 ,

**filename.srl    filename\_001.srl**

としてログファイルが作成されます。

同様に , filename\_001.srl も存在した場合 ,

**filename.srl    filename\_002.srl**

としてログファイルが作成されます。

同様に , filename\_002.srl, filename\_003.srl, ... と存在した場合 ,

**filename.srl    filename\_003.srl**

•

•

**filename.srl    filename\_xxx.srl**

となっていきます。

---

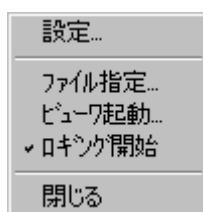
#### < 注意事項 >

1. ログファイルの保存先は , 内蔵 HDD のみサポートしています。保存先がネットワーク上や外部 HDD, 外部ディスク (CD,DVD,MO など ) 等の場合は , サポート対象外となります。
  2. RAM チェッカのログファイルを保存するには , ディスクに 500MB 以上の空きが必要となります。ディスク容量が 500MB 未満になるとロギングを中断します。
- 

## ■ ロギング開始

RAM チェッカのロギング状態を制御します。ショートカットメニューがチェックされていれば有効です。この状態で実行を開始すると , ログデータの取得と RAM チェッカウィンドウでのモニタリング表示が行われます。

図 3.19-6 ロギング有効状態



## ■ ロギング中の CPU 停止

RAM チェッカ中にパワーオンデバッグが可能です。

---

### < 注意事項 >

1. 下限電圧は使用するマイコンの動作下限電圧を設定して下さい。
  2. パワーオンデバッグ復帰時からデータアクセスがあるまで ICE は不定な値を出力します。  
この時のデータは Viewer では表示されません。
-

## 3.19.2 RAM Checker Viewer の起動

---

RAM Checker Viewer の起動について説明します。

---

### ■ ビューワ起動

ショートカットメニュー [ ビューワ起動 ... ] により、ビューワの起動ダイアログ ( 図 3.19-7 ) が起動します。RAM Checker Viewer 起動時に開くログファイルを選択し、[ 起動 ... ] ボタンにより RAM Checker Viewer ( 図 3.19-8 ) を起動します。

図 3.19-7 ビューワの起動ダイアログ

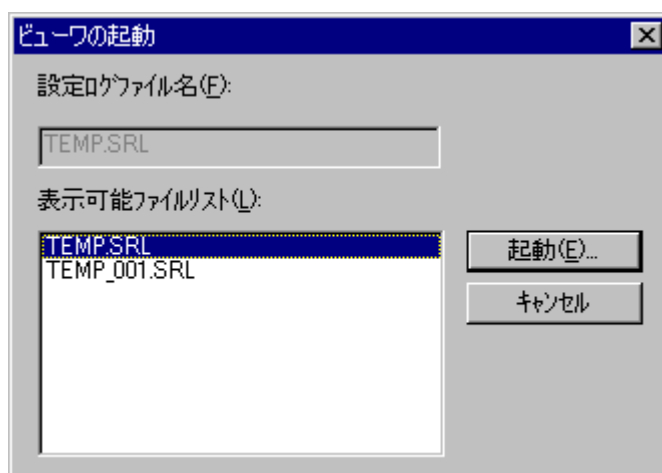
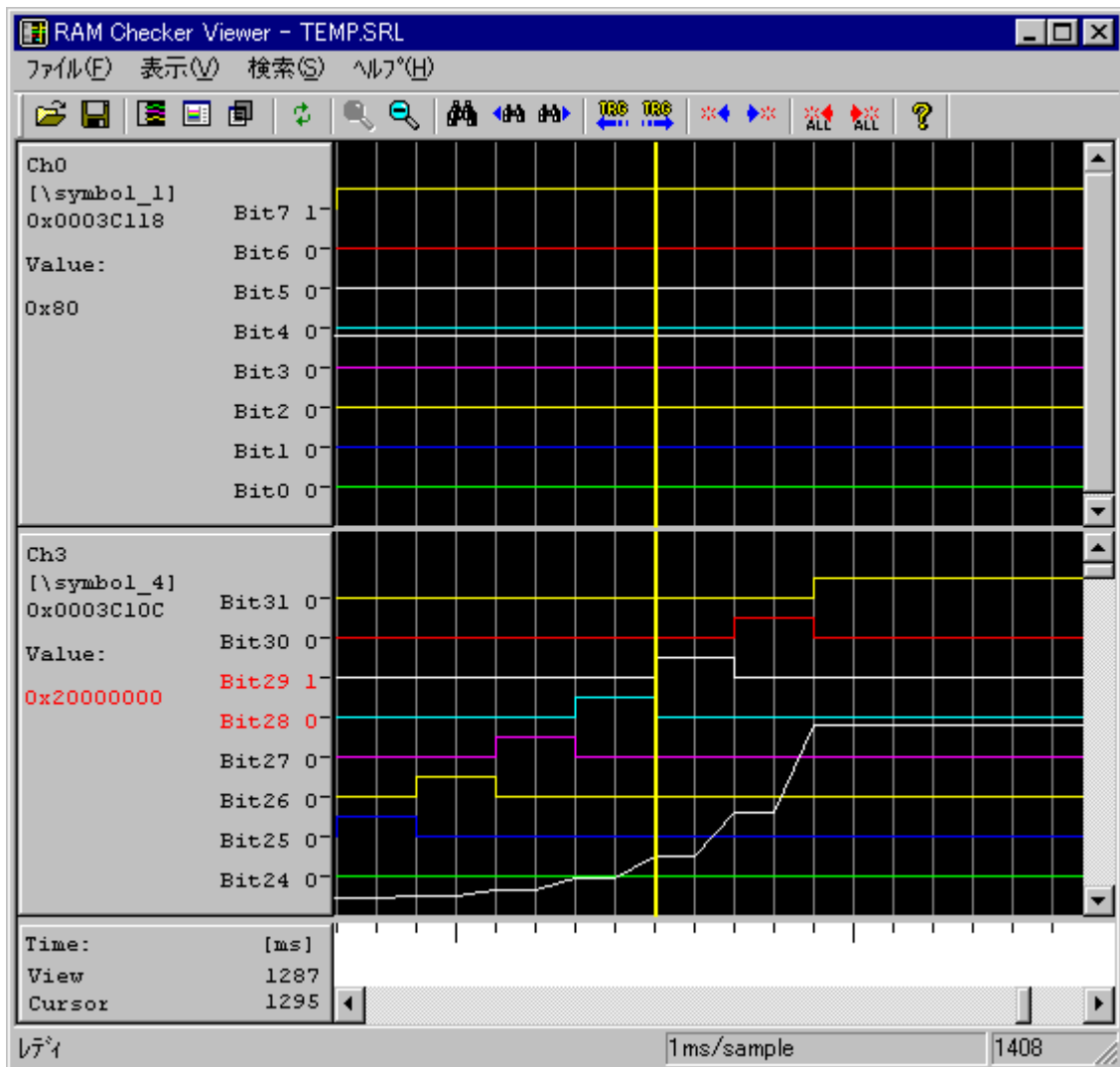




図 3.19-8 RAM Checker Viewer



または、Windows スタートメニューの[ SOFTUNE V6 ][ FR Family SOFTUNE Workbench Tool ]にある、[ RAM Checker Viewer ]メニューを実行してください。

RAM Checker Viewer に関する説明は、別途 FswbRView.pdf およびヘルプを参照してください。



# 第4章

---

## メニュー

**SOFTUNE Workbench のメニューの構成と、メニューから起動される各ダイアログの説明をします。**

- 4.1 メニューの構成 (階層)
- 4.2 ファイルメニュー
- 4.3 編集メニュー
- 4.4 表示メニュー
- 4.5 プロジェクト
- 4.6 デバッグ
- 4.7 環境
- 4.8 ウィンドウ
- 4.9 ヘルプ

## 4.1 メニューの構成 ( 階層 )

---

**SOFTUNE Workbench のメニュー構成について述べます。**

---

### ■ メニュー構成

SOFTUNE Workbench のメインウィンドウには、以下に示したメニューが表示されます。

[ファイル] [編集] [表示] [プロジェクト] [デバッグ] [環境] [ウィンドウ] [ヘルプ]
---

それぞれのメニュー項目には、以下の機能が割り付けられています。

#### ● ファイル

- 新規作成
- 開く
- 閉じる
- ワークスペースを開く
- ワークスペースを閉じる
- 上書き保存
- 名前を付けて保存
- すべてを保存
- 印刷
- 最新のテキストファイル
- 最新のワークスペース
- アプリケーションの終了

#### ● 編集

- 元に戻す
- やり直し
- 切り取り
- コピー
- 貼り付け
- 削除
- すべて選択
- 検索
- 置換
- ファイルから検索
- ジャンプ
- ブックマーク
- 前のエラー
- 次のエラー
- 先頭のエラー

- 最後のエラー
- プロパティ

● 表示

- プロジェクト
- アウトプット
- シンボル一覧
- 逆アセンブル
- レジスタ
- メモリ
- ローカル変数
- ウォッチ
- トレース
- カバレッジ
- パフォーマンス
- コマンド
- リアルタイムメモリ
- RAM チェッカ
- ツールバー
- ステータスバー
- タブ
- フォント

● プロジェクト

- アクティブプロジェクトの設定
- プロジェクトの追加
- メンバの追加
- ワークスペースの設定
- プロジェクトの設定
- カスタマイズビルドの設定
- プロジェクトの依存関係
- プロジェクト構成
- 依存関係の更新
- コンパイル
- メイク
- ビルド
- 中止

● デバッグ

- 実行
- 停止
- MCU のリセット
- ブレークポイント

## 第4章 メニュー

- ブレーク設定・解除
- イベント
- シーケンス
- コールスタック
- 時間測定
- 関数コール
- コールクリア
- ベクタ
- ターゲットファイルのロード
- デバッグの開始 / 終了

### ● 環境

- 開発環境
- デバッグ環境の設定
- デバッガのメモリマップ
- ツールの設定
- キーボードの設定
- エディタの設定
- エラージャンプの設定
- ツールの起動

### ● ウィンドウ

- 重ねて表示
- 上下に並べて表示
- 左右に並べて表示
- 上下に分割
- アイコンの整列
- ウィンドウの更新
- すべてのウィンドウの更新
- すべて閉じる

### ● ヘルプ

- トピックの検索
- サポート情報
- バージョン情報

## 4.2 ファイルメニュー

---

SOFTUNE Workbench の [ ファイル ] メニューには , ワークスペースおよびファイルのアクセスに関する機能 , 印刷機能 , SOFTUNE Workbench の終了の機能があります。

---

### ■ プロジェクトおよびファイルのアクセスに関する機能

プロジェクトおよびファイルアクセス関連の機能には以下のものがあります。

- 新規作成
- 開く
- 閉じる
- ワークスペースを開く
- ワークスペースを閉じる
- 上書き保存
- 名前を付けて保存
- すべてを保存
- 最新のテキストファイル
- 最新のワークスペース

### ■ その他の機能

その他の機能には以下のものがあります。

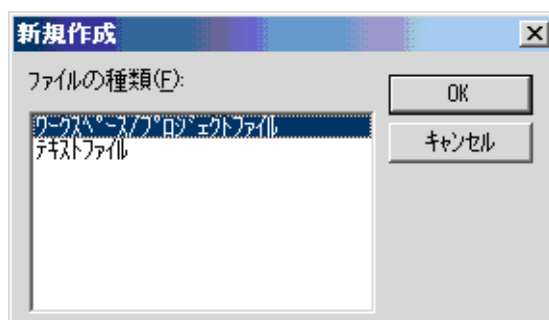
- 印刷
- アプリケーションの終了

## 4.2.1 新規作成

ワークスペース/プロジェクトおよびファイルを新規に作成する機能です。ワークスペース/プロジェクトファイルを作成するのか、テキストファイルを作成するのかを問い合わせるダイアログが表示されます(図4.2-1)。

### ■ 新規作成ダイアログ

図 4.2-1 新規作成ダイアログ

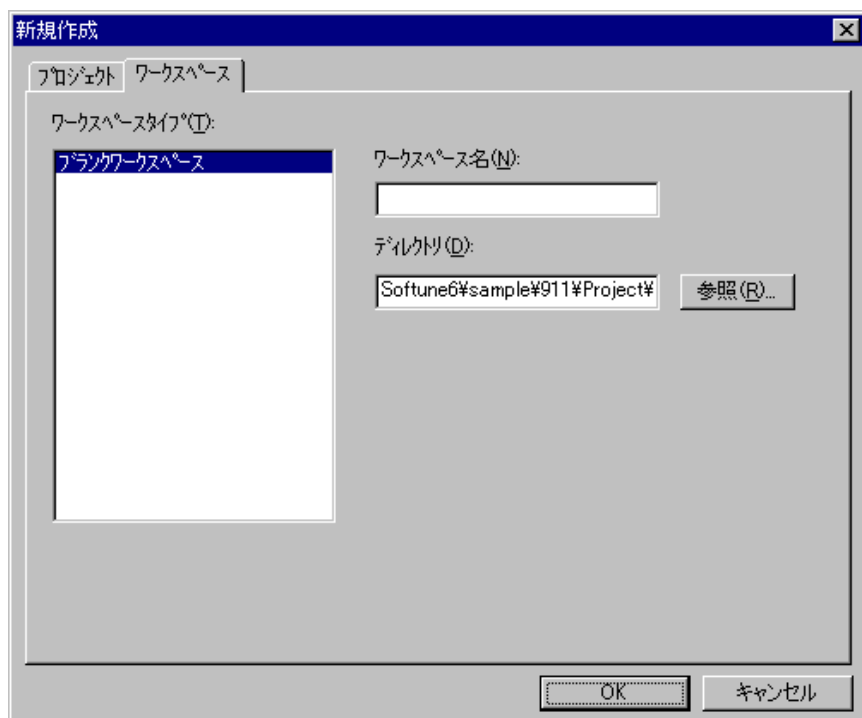


### ■ 新規ワークスペースを作成する場合

新規ワークスペースを作成するには、「ワークスペース/プロジェクトファイル」を選択してください。プロジェクトの「新規作成」ダイアログが開きますので、「ワークスペース」タブを選択してください(図4.2-2)。ここで、作成するワークスペースに関する基本的な項目を設定します。

詳しい設定手順は、「2.3 ワークスペースの作成」を参照してください。

図 4.2-2 プロジェクトの新規作成ダイアログ(ワークスペース)



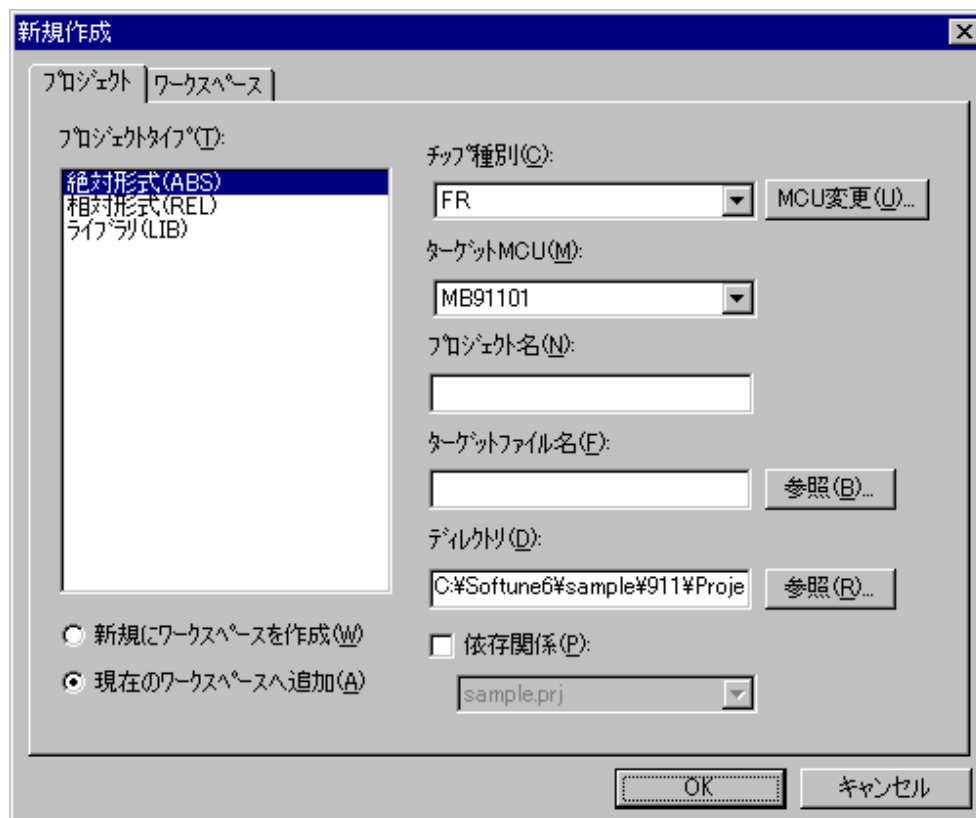


## ■ 新規プロジェクトを作成する場合

新規プロジェクトを作成するには、[ワークスペース / プロジェクトファイル]を選択してください。プロジェクトの[新規作成]ダイアログが開きますので、[プロジェクト]タブを選択してください(図4.2-3)。ここで、作成するプロジェクトに関する基本的な項目を設定します。

新規プロジェクトを作成する際に、新規にワークスペースを作成する場合は「2.3 ワークスペースの作成」を、現在のワークスペースに追加する場合は「2.4 プロジェクトの登録」を参照してください。

図 4.2-3 プロジェクトの新規作成ダイアログ (プロジェクト)



## ■ ソースファイルやドキュメントファイルなどを作成する場合

ソースファイルやドキュメントファイルなどを作成する場合は、テキストファイルを選択してください。新規にファイルを作成する状態でエディタが起動します。

## 4.2.2 開く

既存のファイルを開く機能です。開くファイルの種類を問い合わせるダイアログが表示されます (図 4.2-4)。

### ■ 開くダイアログ

ファイルを開きます。

ファイルの種類では、下記が選択できます。

ワークスペースファイル

プロジェクトファイル

ロードモジュールファイル

ソースファイル

バイナリファイル (デバッグセッション時のみ)

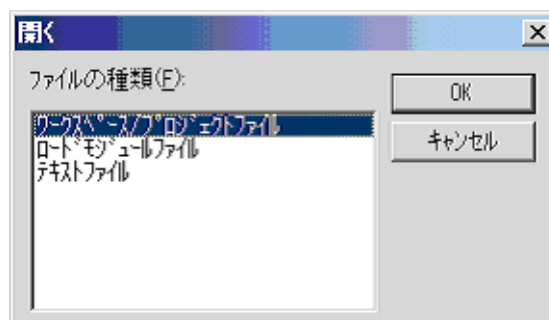
エイリアスファイル (デバッグセッション時のみ)

カバレッジファイル (デバッグセッション時のみ)

バッチファイル (デバッグセッション時のみ)

テキストファイル

図 4.2-4 開くダイアログ



## ■ ロードモジュールファイルを開く場合

- 既にプロジェクトファイルを開いていて、デバッグセッションに入っている場合

デバッグするロードモジュール名を選択するためのファイルダイアログが開きます。デバッグ情報のみをロードしたい場合には、[ デバッグ情報のみ ] のチェックボックスを設定してください。

シミュレータデバッグ時には、メモリマップの設定を自動でおこないたい場合、[ 自動マップ設定 ] のチェックボックスを設定してください。これが設定されていない場合は、ロードモジュールを開く前にメモリマップの設定をおこなっておいてください。

オンデマンドロードする場合には、[ オンデマンドロード ] のチェックボックスを設定してください。

- 既にプロジェクトファイルを開いていて、デバッグセッションに入っていない場合

デバッグセッションに入るように指示するダイアログが表示されます。[OK] ボタンで自動的にデバッグセッションに入ります。

[キャンセル] でファイルを開く動作をキャンセルします。

- ワークスペースファイルが開かれていない場合

まず、プロジェクトを新規に作成するか既存のワークスペース/プロジェクトを開くかを確認するためのダイアログが表示されます。

- プロジェクトを新規に作成する場合は、[ はい ] をクリックしてください。
- 既存のワークスペース/プロジェクトを読み込む場合は、[ いいえ ] をクリックしてください。
- [キャンセル] でファイルを開く動作をキャンセルします。

## ■ バイナリファイルを開く場合

[ 開始アドレス ] には、バイナリデータの読み込み開始アドレスを指定してください。

## ■ バッチファイルを開く場合

[ アイコン状態で実行 ] を設定すると、アイコン状態でバッチを開始します。

## 4.2.3 閉じる

---

既に関いているファイルやウィンドウを閉じる機能です。

---

### ■ 閉じる

エディットウィンドウで開いているファイルまたはレジスタウィンドウのようなデバッグ用のウィンドウを閉じます。

#### ● エディットウィンドウの場合

修正してセーブしていないファイルがある場合は、セーブするかどうか確認するダイアログが表示されます。

- [はい] : ファイルに上書き保存します。
- [いいえ] : ファイルに保存しないでエディットウィンドウを閉じます。
- [キャンセル] : 閉じる動作をキャンセルします。

#### ● レジスタウィンドウのようなデバッグ用のウィンドウの場合

直ちにウィンドウを閉じます。

## 4.2.4 ワークスペースを開く

---

既存のワークスペースファイルを開く機能です。

---

### ■ 既存のワークスペースを開く

開くファイルを選択するためのファイルダイアログが表示されます。

- 既にワークスペースを開いていて、デバッグセッションの場合

デバッグを終了してもよいかどうか確認するダイアログが表示されます。

- [OK] ボタン : デバッグを終了してワークスペースを開く処理を続けます。
- [キャンセル] : ワークスペースを開く処理をキャンセルします。

- 既にワークスペースを開いていて、デバッグセッションでない場合

ワークスペースファイルを選択するためのファイルダイアログが表示されます。正しいワークスペースファイル名を指定すれば、現在開いているワークスペースファイルを閉じる処理を実行後、指定したワークスペースファイルを開きます。

もし、ファイルがエディット中で保存されていない場合は、そのファイルを保存するかどうか確認するダイアログが表示されます。

- [はい] : そのファイルを上書きして処理を続けます。
- [いいえ] : そのファイルを保存しないで処理を続けます。
- [キャンセル] : ワークスペースを開く処理をキャンセルします。

- ワークスペースを開いていない場合

ワークスペースファイルを選択するためのファイルダイアログが表示されます。正しいワークスペースファイル名を指定すれば、指定したワークスペースファイルを開きます。

ワークスペースを開いたときは、前回そのワークスペースを保存したときのウィンドウ表示を復元します。

- [キャンセル] : エディット中のファイルを開いたままにして処理を続けます。
- ワークスペースファイルの替わりにプロジェクトファイルを開くこともできます。詳細については、「2.3 ワークスペースの作成」を参照してください。

## 4.2.5 ワークスペースを閉じる

---

現在開いているワークスペースを閉じる機能です。

---

### ■ 編集集中のファイルがない場合

ワークスペースを閉じるときに、現在のワークスペースの情報を保存するかどうか確認するダイアログが表示されます。

- [はい] : ワークスペースの情報を保存してからプロジェクトを閉じます。
- [いいえ] : ワークスペースの情報を保存しないでプロジェクトを閉じます。
- [キャンセル] : ワークスペースを閉じる動作をキャンセルします。

ただし以下の場合には、確認するダイアログは表示されません。

- ワークスペース、プロジェクトに変更がない場合
- [ワークスペースを閉じる時にセーブの問い合わせを行う] の設定を行っていないとき。

### ■ 編集集中のファイルがある場合

エディットウィンドウで編集集中のファイルがない場合と同様に、まず、現在のワークスペースの情報を保存するかどうか確認するダイアログが表示されます。[はい]または[いいえ]で処理を継続すると、その後、エディット中のファイルを保存するかどうか確認するダイアログが表示されます。

- [はい] : エディット中のファイルを保存してからワークスペースを閉じます。
- [いいえ] : エディット中のファイルを保存しないでワークスペースを閉じます。
- [キャンセル] : エディット中のファイルを開いたままにしてワークスペースを閉じます。

## 4.2.6 上書き保存

---

現在開いているファイルに上書きします。

---

### ■ 上書き保存ダイアログ

一度操作を行ったファイル名を対象に上書き保存を行います。

ファイルの種類では、下記が選択できます。

テキストファイル

ワークスペースファイル

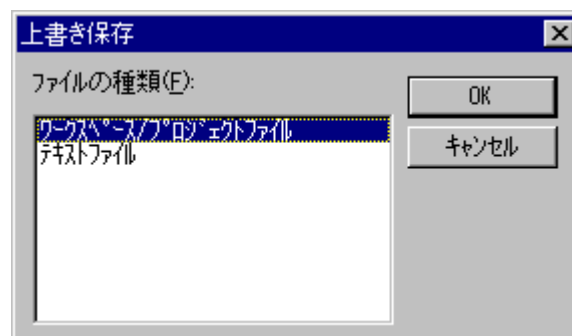
バイナリファイル（デバッグセッション時のみ）

エイリアスファイル（デバッグセッション時のみ）

セットアップファイル（デバッグセッション時のみ）

カバレッジファイル（デバッグセッション時のみ）

図 4.2-5 上書き保存ダイアログ



## 4.2.7 名前を付けて保存

---

ファイルに新しい名前を付けて保存します。

---

### ■ 名前を付けて保存

名前を付けて保存します。

ファイルの種類では、下記が選択できます。

テキストファイル

アウトプットウィンドウ

プロジェクトファイル

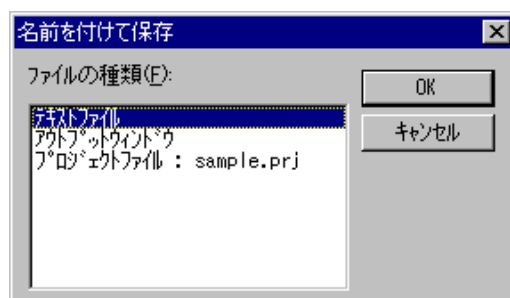
バイナリファイル (デバッグセッション時のみ)

エイリアスファイル (デバッグセッション時のみ)

セットアップファイル (デバッグセッション時のみ)

カバレッジファイル (デバッグセッション時のみ)

図 4.2-6 名前を付けて保存ダイアログ



### ■ プロジェクトファイルを保存する場合

アクティブプロジェクトの保存を行います。[ファイルの種類] コンボボックスでプロジェクトファイルの保存形式を選択することができます。プロジェクト形式については、『SOFTUNE Workbench 機能説明書』の「1.2 プロジェクト管理機能」を参照してください。

「ワークスペースプロジェクト形式 (\*.prj)」を選択した場合は、ワークスペースプロジェクト形式(ワークスペース対応以降のプロジェクト)でプロジェクトのすべての情報を保存します。旧プロジェクト形式でオープンしたプロジェクトをワークスペースプロジェクト形式で上書きすると、変換を行って旧プロジェクトの制限が解除されます。以降の保存はワークスペースプロジェクト形式になります。

「旧プロジェクト形式 (\*.prj)」を選択した場合は、旧プロジェクト形式(ワークスペース対応以前のプロジェクト)でプロジェクトのアクティブな構成の情報を保存します。ワークスペースプロジェクト形式でオープンしたプロジェクトを旧プロジェクト形式で上書きすることはできません。その場合は、別名で保存してください。

### ■ バイナリファイルを保存する場合

[開始アドレス]には、保存開始アドレスを指定します。以前にロードされていればそのときの開始アドレスが設定されます。

[終了アドレス]には、保存終了アドレスを指定します。以前にロードされていればそのときの終了アドレスが設定されます。

---

#### < 注意事項 >

プロジェクトファイルを元のディレクトリと違う場所に保存した場合は、相対パス等で記述されたメンバが参照されなくなる可能性があります。その場合は、メンバを元のディレクトリ構成を保ったまま移動させてください。

---



## 4.2.8      すべてを保存

---

現在エディット中のファイルをすべて上書き保存します。

---

### ■ 保存対象

保存の対象になるファイルは、エディットウィンドウで開いているソースファイルなどのテキストファイルです。ワークスペースファイル、プロジェクトファイルやアウトプットウィンドウの内容の保存は行いません。

また、エディットウィンドウで開いているテキストファイルであっても、それが編集されていなければ保存の対象にはなりません。

デバッグセッション時は、一度取り扱ったファイルで、バイナリファイル、エイリアスファイル、セットアップファイル、カバレッジファイルがあります。

### ■ 新規に作成したテキストの処理

新規に作成して、まだ一度も保存を行っていないテキストがある場合は、保存するファイル名を入力するためのファイルダイアログを開きますので、ファイル名を指定してください。

## 4.2.9 印刷

---

ファイルの内容を印刷します。

---

### ■ 印刷対象

現在エディットウィンドウで開いているファイルの内容を印刷します。それ以外のウィンドウの内容は印刷できません。

アウトプットウィンドウの内容を印刷したい場合は、以下のいずれかの方法で行ってください。

- アウトプットウィンドウの内容をエディットウィンドウにコピーして印刷します。

以下の手順で、アウトプットウィンドウの内容をエディットウィンドウにコピーしてから、[ファイル]メニューで[印刷]を選択してください。

- 1) アウトプットウィンドウの内容をすべて選択してください。
- 2) マウスの右ボタンでメニューを表示し、[コピー]を選択してください。
- 3) 新規にエディットウィンドウを開いて、[編集]メニューから[貼り付け]を選択してください。

- アウトプットウィンドウの内容をいったんファイルに保存してから印刷します。

アウトプットウィンドウの内容をファイルに保存する方法は、「4.2.7 名前を付けて保存」を参照してください。

保存したファイルをエディットウィンドウで開いてから、[ファイル]メニューから[印刷]を選択してください。

## 4.2.10 最新のテキストファイル / 最新のワークスペース

---

デバッガのソースウィンドウで開いたテキストファイルと、使用したワークスペースを、最近のものからそれぞれ5個まで記憶します。

---

### ■ 最新のテキストファイル

デバッガのソースウィンドウで開いたテキストファイルを、最近開いたものからさかのぼって最大5個まで記憶していて、選択することで直ちに開くことができます。

エディットウィンドウで開いたファイルはここには記憶されません。

### ■ 最新のワークスペース

最近使用したワークスペースファイルを最大5個まで記憶していて、選択することで直ちに開くことができます。

---

#### < 注意事項 >

既にファイルが削除されていたり、別のディレクトリへ移動されていたりする場合には、ここにリストされていても開くことはできません。

---

## 4.2.11 アプリケーションの終了

---

**SOFTUNE Workbench を終了します。**

---

### ■ エディット中の場合

エディット中のファイルを保存するかどうかを確認するダイアログが表示されます。

- [ はい ] : エディット中のファイルを保存してから終了します。
- [ いいえ ] : エディット中のファイルを保存しないで終了します。
- [ キャンセル ] : SOFTUNE Workbench を終了しません。

### ■ ワークスペースが開かれている場合

まず、現在のワークスペースの情報を保存するかどうかを確認するダイアログが表示されます。

- [ はい ] : ワークスペースの情報をファイルに保存します。
- [ いいえ ] : ワークスペースの情報をファイルに保存しません。
- [ キャンセル ] : SOFTUNE Workbench を終了しません。

次に、エディット中のファイルがなければ直ちに終了します。エディット中のファイルがある場合は、エディット中の場合に述べた処理が行われます。

### ■ デバッグ中の場合

まず、ワークスペースが開かれている場合の処理と同様に、現在のワークスペースの情報を保存するかどうかを確認するダイアログが表示されます。ここで、[ はい ] または [ いいえ ] で処理を継続すると自動的にデバッグセッションを終了します。

次に、エディット中のファイルがなければ直ちに終了します。エディット中のファイルがある場合は、エディット中の場合に述べた処理が行われます。

## 4.3 編集メニュー

---

ファイルの編集，文字列の検索，エラージャンプなどの機能があります。

---

### ■ ファイルの編集機能

ファイル編集の機能には以下のものがあります。

- 元に戻る
- やり直し
- 切り取り
- コピー
- 貼り付け
- 削除
- すべて選択

### ■ 文字列の検索機能

文字列の検索および置換ができます。

- 文字列の検索
- 文字列の置換
- ファイルから検索

### ■ ジャンプ機能

ウィンドウの任意の箇所にすばやく移動できます。

- ジャンプ

### ■ ブックマーク機能

ブックマークの設定やブックマークへのジャンプが行えます。

- ブックマーク

### ■ エラージャンプ

エラージャンプの機能には以下のものがあります。

- 前のエラー
- 次のエラー
- 先頭のエラー
- 最後のエラー

### ■ プロパティ機能

アクティブなウィンドウの情報を表示します。

- プロパティ

### 4.3.1 元に戻す / やり直し

---

既に行った編集を取り消して、その編集前の状態に戻すことができます。また、取り消した編集を再び元の状態に戻すことができます。

---

#### ■ 対象

この機能は、エディットウィンドウに対する編集についてのみ有効です。また、この機能では、文字および行を連続して追加または削除した「一連の操作」を複数回取り消したり元に戻したりできます。

---

#### < 注意事項 >

この機能は、エディットウィンドウに対してのみ有効です。デバッグセッション時のメモリウィンドウ、レジスタウィンドウなどで値を変更した場合には元に戻せませんので注意してください。大きな編集作業を行うと、[元に戻す / やり直し] がすべて実行できない場合があります。

---

## 4.3.2 切り取り / コピー / 貼り付け / 削除

選択した範囲の文字列を、切り取り / コピー / 削除します。また、切り取り / コピーした文字列を、カーソル位置に挿入したり選択した文字列と置き換えたりできます。

### ■ 切り取り

エディットウィンドウで選択した文字列を切り取り、クリップボードに移動します。エディットウィンドウ以外では実行できません。

ここで切り取った文字列は、後で貼り付けることができます。

### ■ コピー

エディットウィンドウとデバッグセッション時のシンボルウィンドウ、逆アセンブルウィンドウ、レジスタウィンドウ、メモリウィンドウ、ローカルシンボルウィンドウ、ウォッチウィンドウ、トレースウィンドウ、コマンドウィンドウで選択した文字列をクリップボードにコピーします。

ここでコピーした文字列は、後で貼り付けることができます。

### ■ 貼り付け

あらかじめ、切り取りまたはコピーによってクリップボードにコピーされている文字列を、エディットウィンドウの現在のカーソル位置に挿入します。このとき、エディットウィンドウに選択されている文字列があれば、その文字列を、クリップボードの文字列で置き換えます。

### ■ 削除

エディットウィンドウで選択した文字列を削除します。エディットウィンドウ以外では実行できません。

削除した文字列は、後で貼り付けることができません。

表 4.3-1 各編集機能とウィンドウの関係一覧

ウィンドウ	切り取り	コピー	貼り付け	削除
エディットウィンドウ				
プロジェクトウィンドウ	×	×	×	×
アウトプットウィンドウ	×		×	×
シンボルウィンドウ	×		×	×
逆アセンブルウィンドウ	×		×	×
レジスタウィンドウ	×		×	×
メモリウィンドウ	×	(*1)	×	×
ローカルシンボルウィンドウ	×	(*2)	×	×
ウォッチウィンドウ	×	(*2)	×	×
トレースウィンドウ	×	(*1)	×	×
コマンドウィンドウ	×		×	×
カバレッジウィンドウ	×	(*1)	×	×
パフォーマンスウィンドウ	×		×	×

\*1：カラムタイトル行以外の行のコピーが可能です。

\*2：シンボル名のみコピーが可能です。

### 4.3.3      すべて選択

---

ウィンドウ中に表示されているすべての文字列を選択状態にします。

---

#### ■ 対象

この機能は、エディットウィンドウとアウトプットウィンドウでのみ動作します。それ以外のウィンドウでは動作しません。



### 4.3.4 検索 / 置換

エディットウィンドウに表示されているテキストファイル中から任意の文字列を検索し、ほかの文字列に置換することができます。

#### ■ 対象

検索 / 置換の対象となるのはエディットウィンドウに表示されているテキストファイルのみです。アウトプットウィンドウその他のウィンドウは対象になりません。

#### ■ 検索

検索ダイアログ (図 4.3-1) が表示されたら、検索文字列を指定してください。また、ここで文字列検索時の条件 (表 4.3-2) を指定することができます。

文字列が見つかった場合は、その文字列が反転表示されます。

文字列が見つかった場合も、見つからなかった場合も、検索ダイアログは自動的に閉じませんので、検索終了時には [キャンセル] をクリックしてダイアログを閉じてください。

デバッグ中は、ソースウィンドウに対しても検索可能です。

図 4.3-1 検索ダイアログ



## ■ 置換

置換ダイアログ (図 4.3-2) が表示されたら、検索文字列と置換文字列を指定してください。このとき、文字列検索時の条件 (表 4.3-2) を指定することができます。

文字列が見つかった場合は、その文字列が反転表示されます。ここで、[ 置換 ] ボタンをクリックすると文字列を置き換えます。また、[ すべて置換 ] をクリックすると、それ以降検索終了までに見つかったすべての文字列を置き換えます。

文字列が見つかり置換が実施された場合も、そうでなかった場合も、置換ダイアログは自動的に閉じませんので、検索終了時には [ キャンセル ] をクリックしてダイアログを閉じてください。

図 4.3-2 置換ダイアログ



表 4.3-2 文字列検索時の条件と内容の一覧

条 件	内 容
単語単位で探す	ある文字列の部分文字列を検索対象外にします。 "able" などのようにほかの単語の一部になる可能性のある独立した単語を検索するときに有効です。
大文字と小文字を区別する	アルファベットの大文字と小文字を区別して検索します。
正規表現	検索文字列を正規表現で指定します。ワイルドカードなどが使用可能になります。
方向	検索する方向を指定します (上へ, 下へ)。"上へ" はファイルの先頭方向へ, "下へ" はファイルの末尾方向へ検索します。
位置	検索を開始する位置を指定します (カーソル, テキスト先頭, テキスト末尾)。

### 4.3.5 ファイルから検索

指定したファイルに指定した文字列があるかどうか検索します。

#### ■ ファイルから検索

指定したファイルに指定した文字列があるかどうか検索します。検索結果はアウトプットウィンドウに表示します。

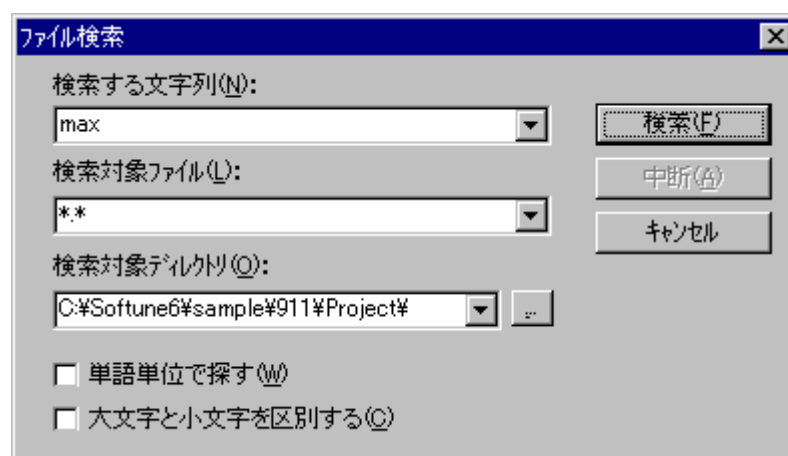
複数のファイルを一度に検索できます。

アウトプットウィンドウの出力でダブルクリックすることによってすぐにその行を編集できます。

#### ■ 検索手順

- 1) [編集]-[ファイルから検索]メニューを選択してください。
  - ファイル検索ダイアログ(図 4.3-3)が表示されます。
- 2) [検索する文字列]を指定してください。
  - ダイアログを開いたときは、クリップボードに入っている文字列が表示されます。
- 3) [検索対象ファイル]を設定してください。
  - ワイルドカードによる指定もできます。
- 4) 検索対象ディレクトリを指定してください。
  - 指定フィールドの右横のボタンでディレクトリ検索のダイアログが表示されます。
- 5) 必要に応じて[単語単位で探す],[大文字と小文字を区別する]を指定してください。
- 6) [検索]ボタンをクリックしてください。
  - 検索結果がアウトプットウィンドウにリアルタイムで表示されます。
- 7) 検索を中断する場合は,[中断]ボタンをクリックしてください。
  - ファイル検索が中断されます。
- 8) 検索が終了したら,[キャンセル]をクリックしてください。

図 4.3-3 ファイル検索ダイアログ



### 4.3.6 ジャンプ

エディット中のテキストファイルの任意の行へ、カーソル位置を移動させます。また、デバッグセッション時は、ソース行、逆アセンブル、メモリ、トレースの表示開始位置を指定します。

#### ■ エディットウィンドウ

ジャンプダイアログ(図 4.3-4)が表示されたら、ジャンプ先の行番号を指定してください。エディットウィンドウがアクティブになっている場合には、SOFTUNE Workbench がデバッグセッションであるかどうかに関係なく、エディットウィンドウ中でのカーソルが移動します。

図 4.3-4 ジャンプダイアログ (エディット)



#### ■ デバッグセッション

デバッグセッション時にエディットウィンドウ以外のウィンドウがアクティブなときには、図 4.3-5 のダイアログが開きます。[指定形式]を選択してから、それにしたがった表現方法で[表示位置]を指定してください。

[指定形式]で[アドレス]を指定した場合は、[ウィンドウ]でジャンプ先のウィンドウをソース、メモリ、逆アセンブルウィンドウなどから指定できます。

[指定形式]には次のものが選択可能です。

- 行番号
- アドレス
- フレーム

[ウィンドウ]には次のウィンドウが選択可能です。

- ソースウィンドウ
- メモリウィンドウ
- 逆アセンブルウィンドウ
- リアルタイムメモリウィンドウ

図 4.3-5 ジャンプダイアログ (デバッグ)



### 4.3.7 ブックマーク

---

ソース、メモリ、エディットウィンドウの特定の位置を頻繁に参照する場合は、ブックマークを設定しておく便利です。ブックマークを設定すると、指定した箇所にマークが付き、メニューまたはキー操作で特定箇所へすぐにジャンプできるようになります。ブックマークは、解除するまで有効です。

---

#### ■ ブックマーク

ブックマークとは、ウィンドウ内の特定の位置を指定し、メニューまたはキー操作ですぐにジャンプできる機能です。

#### ■ 対応するウィンドウ

ブックマークは下記のウィンドウで有効です。

- ソースウィンドウ
- メモリウィンドウ
- エディットウィンドウ

#### ■ ブックマークメニュー

ブックマーク関連のメニューは [ 編集 ] - [ ブックマーク ] 内にあります。

サブメニューには下記の7種類があります。メニューはデバッグセッション時、各ウィンドウがアクティブなときにのみ有効となります。

- ブックマーク
- ブックマークの設定・解除
- 次のブックマーク
- 前のブックマーク
- ウィンドウ内の次のブックマーク
- ウィンドウ内の前のブックマーク
- ウィンドウ内のブックマークをすべて削除

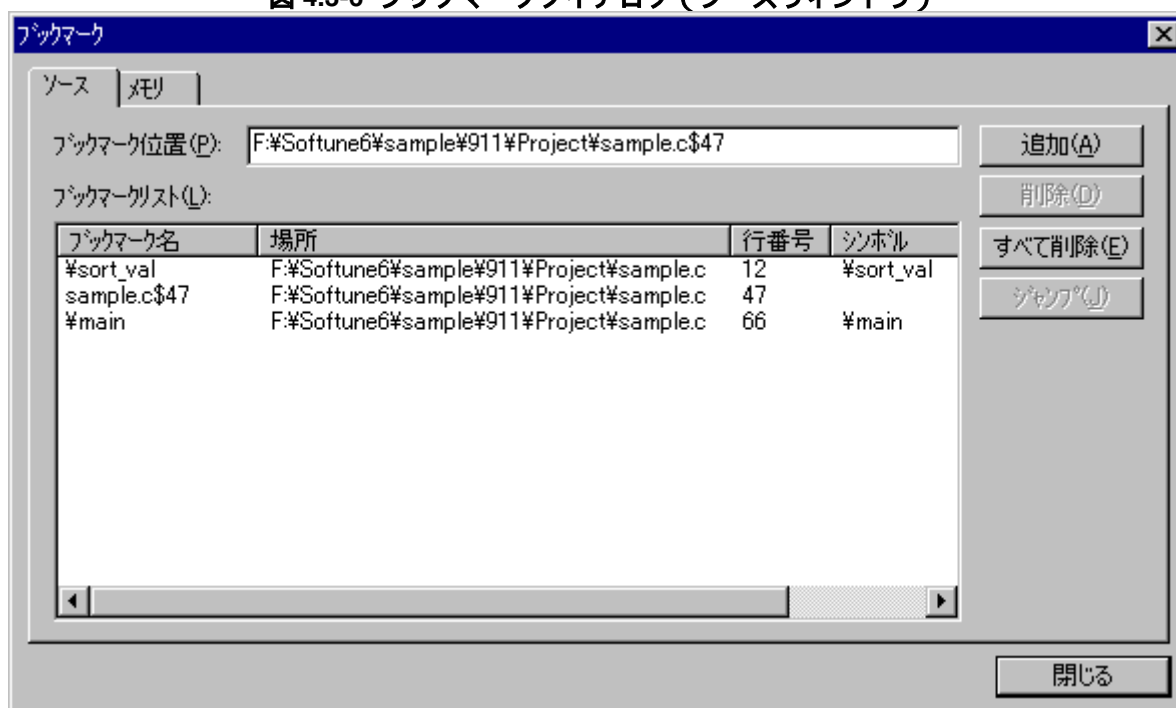
### 4.3.7.1 ブックマーク - ソースウィンドウ

ここではソースウィンドウがアクティブな場合の動作について説明します。

#### ■ ブックマークダイアログを表示

[編集] - [ブックマーク] - [ブックマーク] を選択すると、ソースウィンドウに設定したブックマーク一覧を表示する、図 4.3-6 のダイアログが開きます。

図 4.3-6 ブックマークダイアログ (ソースウィンドウ)



- ブックマーク位置
  - ブックマークを設定する位置を指定するフィールドです。相対パスや関数名、ラベル名での指定ができます。
- ブックマークリスト
  - ソースウィンドウで現在設定されているブックマークの一覧が表示されます。
  - ブックマーク名 : ブックマーク名が表示されます。行番号またはシンボル名がブックマーク名となります。
  - 場所 : ブックマークが設定されているファイル名が表示されます。
  - 行番号 : ブックマークが設定されている行番号が表示されます。
  - シンボル : ブックマーク名に割り付けられているシンボルが表示されます。シンボルが割り付けられている場合は、ファイルの変更によりシンボルの行番号が変更されてもデバッグ情報を基にブックマーク位置を更新します。割付可能なシンボルに関しては注意事項を参照してください。
- [追加] ボタン
  - [ブックマーク位置] で指定された位置にブックマークを設定します。

- [ 削除 ] ボタン
  - ブックマークリスト中で選択されているブックマークを削除します。
- [ すべて削除 ] ボタン
  - ブックマークリスト中のすべてのブックマークを削除します。
- [ ジャンプ ] ボタン
  - ブックマークリスト中で選択されているブックマークの位置まで、ソースウィンドウの表示開始位置を移動させます。

## ■ ブックマークを設定または解除

- 1) ブックマークを設定または解除したい行にカーソルを移動させてください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ブックマークの設定・解除 ] を選択
  - [ 検索 ] ツールバーの [ ブックマークの設定・解除 ] をクリック
  - [ ブックマーク ] ダイアログの [ ソース ] タブからブックマークを追加、または削除

## ■ 次のブックマークへ移動

- 1) ソースウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ 次のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ 次のブックマーク ] をクリック
- 3) ウィンドウ内に次のブックマークが存在しない場合は、次のファイルのブックマークへ移動します。

## ■ 前のブックマークへ移動

- 1) ソースウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ 前のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ 前のブックマーク ] をクリック
- 3) ウィンドウ内に1つ前のブックマークが存在しない場合は、1つ前のファイルのブックマークへ移動します。

## ■ ウィンドウ内の次のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ウィンドウ内の次のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ ウィンドウ内の次のブックマーク ] をクリック

## ■ ウィンドウ内の前のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ウィンドウ内の前のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ ウィンドウ内の前のブックマーク ] をクリック

## ■ ウィンドウ内のブックマークを全て削除

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [編集]-[ブックマーク]-[ウィンドウ内のブックマークをすべて削除]を選択
  - [検索]ツールバーの[ウィンドウ内のブックマークをすべて削除]をクリック
  - [ブックマーク]ダイアログの[ソース]タブから[すべて削除]をクリック

---

### < 注意事項 >

ブックマーク名に割り付けることができないシンボルは、自動変数と構造体、共用体、クラスのメンバとなります。

同名シンボルが複数定義されている場合、想定したシンボル以外が使用される場合があります。同名シンボルが存在するシンボルを指定する場合は、モジュール名、関数名を指定してシンボルを指定してください。

---



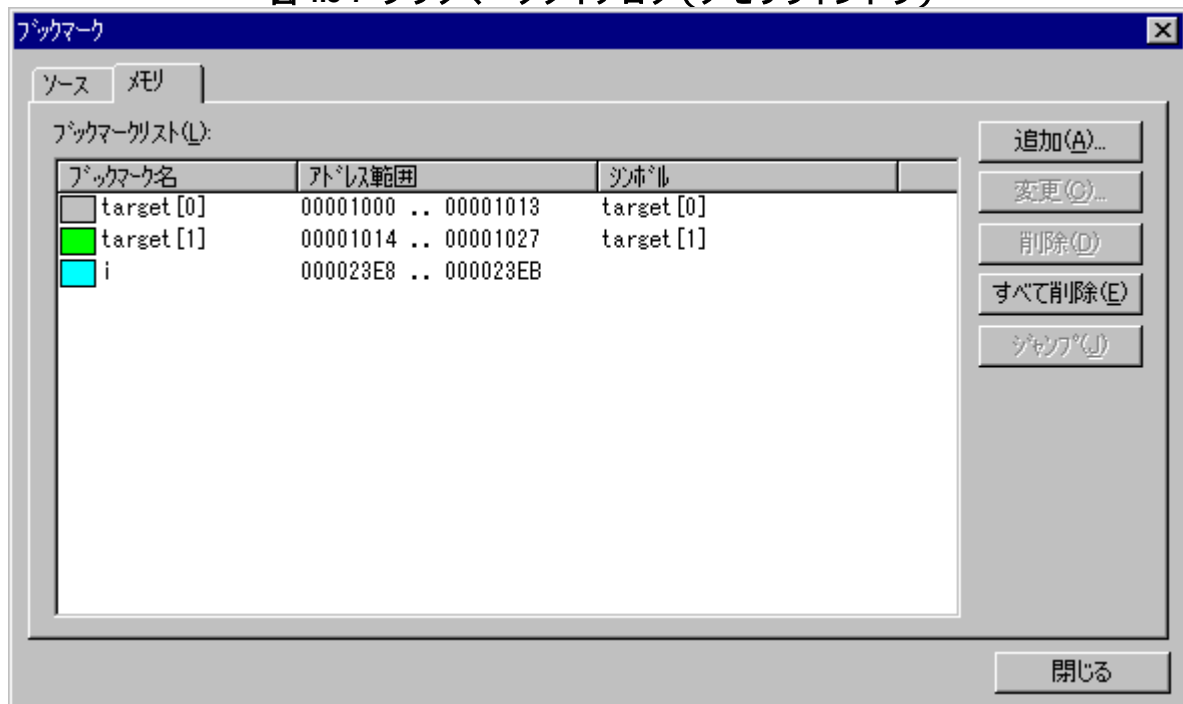
### 4.3.7.2 ブックマーク - メモリウィンドウ

ここではメモリウィンドウがアクティブな場合の動作について説明します。

#### ■ ブックマークダイアログを表示

[編集] - [ブックマーク] - [ブックマーク] を選択すると、メモリウィンドウに設定したブックマーク一覧を表示する、図 4.3-7 のダイアログが開きます。

図 4.3-7 ブックマークダイアログ (メモリウィンドウ)



- ブックマークリスト
  - メモリウィンドウで現在設定されているブックマークの一覧が表示されます。
    - ブックマーク名 : ブックマーク名とマーキング色が表示されます。アドレスまたはシンボル名がブックマーク名となります。
    - アドレス範囲 : ブックマークが設定されているアドレス範囲が表示されます。
    - シンボル : ブックマーク名に割り付けられているシンボルが表示されます。シンボルが割り付けられている場合は、ファイルやツールオプションの変更によりシンボルの配置アドレスが変更されてもデバッグ情報を基にブックマーク位置を更新します。割付可能なシンボルに関しては注意事項を参照してください。
- [追加] ボタン
  - [ブックマーク追加] ダイアログを表示し、ブックマークを追加します。詳細は「4.4.5 メモリ」を参照してください。
- [変更] ボタン
  - [ブックマーク追加] ダイアログを表示し、ブックマークリスト中の選択されたブックマークの設定を変更します。

- [ 削除 ] ボタン
  - ブックマークリスト中で選択されているブックマークを削除します。
- [ すべて削除 ] ボタン
  - ブックマークリスト中のすべてのブックマークを削除します。
- [ ジャンプ ] ボタン
  - ブックマークリスト中で選択されているブックマークの位置まで、メモリウィンドウの表示開始位置を移動させます。

### ■ ブックマークを設定または解除

- 1) ブックマークを設定または解除したいアドレス範囲を選択してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ブックマークの設定・解除 ] を選択
  - [ 検索 ] ツールバーの [ ブックマークの設定・解除 ] をクリック
  - ショートカットメニューの [ ブックマーク追加 ] からブックマークを設定（詳細は「4.4.5 メモリ」を参照してください。）
  - [ ブックマーク ] ダイアログの [ メモリ ] タブからブックマークを追加、または削除

### ■ ウィンドウ内の次のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ウィンドウ内の次のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ ウィンドウ内の次のブックマーク ] をクリック

### ■ ウィンドウ内の前のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ウィンドウ内の前のブックマーク ] を選択
  - [ 検索 ] ツールバーの [ ウィンドウ内の前のブックマーク ] をクリック

### ■ ウィンドウ内のブックマークを全て削除

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ウィンドウ内のブックマークをすべて削除 ] を選択
  - [ 検索 ] ツールバーの [ ウィンドウ内のブックマークをすべて削除 ] をクリック
  - [ ブックマーク ] ダイアログの [ メモリ ] タブから [ すべて削除 ] をクリック

---

#### < 注意事項 >

ブックマーク名に割り付けることができないシンボルは、自動変数と構造体、共用体、クラスのメンバとなります。

同名シンボルが複数定義されている場合、想定したシンボル以外が使用される場合があります。同名シンボルが存在するシンボルを指定する場合は、モジュール名、関数名を指定してシンボルを指定してください。

---

### 4.3.7.3 ブックマーク - エディットウィンドウ

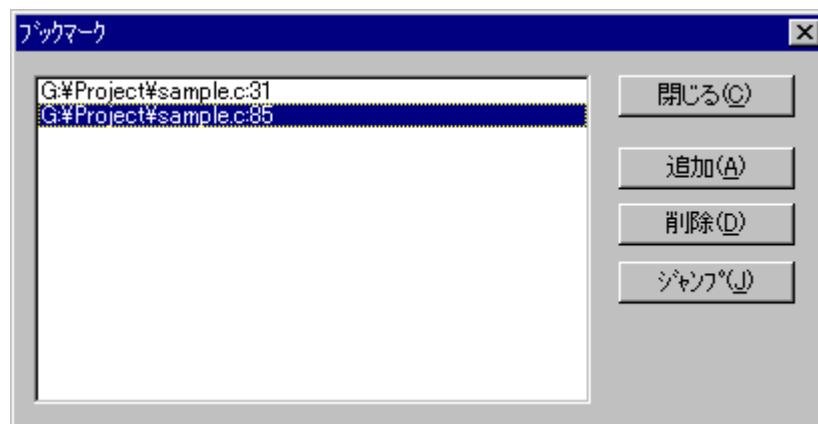
ここではエディットウィンドウがアクティブな場合の動作について説明します。

#### ■ ブックマークダイアログを表示

以下のどちらかのメニューを選択すると、エディットウィンドウに設定したブックマーク一覧を表示する、図 4.3-8 のダイアログが開きます。

1. [ 編集 ] - [ ブックマーク ] - [ ブックマーク ]
2. エディットウィンドウ - ショートカットメニュー [ ブックマーク ]

図 4.3-8 ブックマークダイアログ (エディットウィンドウ)



- ブックマークリスト
  - エディットウィンドウで現在設定されているブックマーク(ファイル名, 行番号)の一覧が表示されます。
- [ 追加 ] ボタン
  - エディットウィンドウにてカーソルを当てている位置にブックマークを設定します。
- [ 削除 ] ボタン
  - ブックマークリスト中で選択されているブックマークを削除します。複数選択時は選択項目を全て削除します。
- [ ジャンプ ] ボタン
  - ブックマークリスト中で選択されているブックマークの位置まで、エディットウィンドウの表示開始位置を移動させます。

#### ■ ブックマークを設定または解除

- 1) ブックマークを設定または解除したい行にカーソルを移動させてください。
- 2) 次のいずれかの操作を行ってください。
  - [ 編集 ] - [ ブックマーク ] - [ ブックマークの設定・解除 ] を選択
  - [ 検索 ] ツールバーの [ ブックマークの設定・解除 ] をクリック
  - [ ブックマーク ] ダイアログの [ 追加 ] または [ 削除 ] ボタンをクリック

### ■ ウィンドウ内の次のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [編集]-[ブックマーク]-[ウィンドウ内の次のブックマーク]を選択
  - [検索]ツールバーの[ウィンドウ内の次のブックマーク]をクリック

### ■ ウィンドウ内の前のブックマークへ移動

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [編集]-[ブックマーク]-[ウィンドウ内の前のブックマーク]を選択
  - [検索]ツールバーの[ウィンドウ内の前のブックマーク]をクリック

### ■ ウィンドウ内のブックマークを全て削除

- 1) ウィンドウ内にカーソルがあることを確認してください。
- 2) 次のいずれかの操作を行ってください。
  - [編集]-[ブックマーク]-[ウィンドウ内のブックマークをすべて削除]を選択
  - [検索]ツールバーの[ウィンドウ内のブックマークをすべて削除]をクリック

### 4.3.8 前のエラー / 次のエラー / 先頭のエラー / 最後のエラー

---

メイク/ビルド, コンパイル, アセンブルでエラーの検出されたソース行へ, エディットウィンドウのカーソルを移動させます。

---

#### ■ 前のエラー

アウトプットウィンドウのカーソル位置の直前のエラー行にジャンプします。カーソル位置より前にエラーの発生した行がない場合は、折り返して最後のエラー行にジャンプします。

#### ■ 次のエラー

アウトプットウィンドウのカーソル位置の直後のエラー行にジャンプします。カーソル位置より後にエラーの発生した行がない場合は、折り返して最初のエラー行にジャンプします。

#### ■ 先頭のエラー

アウトプットウィンドウに出力されている最初のエラー行にジャンプします。

#### ■ 最後のエラー

アウトプットウィンドウに出力されている最後のエラー行にジャンプします。

### 4.3.9 プロパティ

---

指定されたファイルに関する情報を表示します。

---

#### ■ プロパティ

ファイルの情報、モジュール情報、シンボルの情報の表示を行います。現在アクティブになっているウィンドウによって、表示される情報が異なります。

ウィンドウには以下の 6 種類があります。

- プロジェクトウィンドウ
- エディットウィンドウ
- ソースウィンドウ
- シンボルウィンドウ
- ローカルウィンドウ
- ウォッチウィンドウ

### 4.3.9.1 プロパティ - プロジェクトウィンドウ

ここでは、プロジェクトウィンドウのプロパティについて説明します。

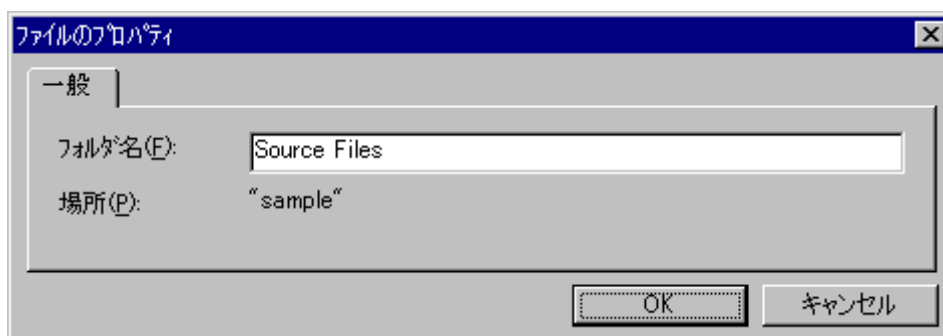
#### ■ プロパティ：一般

図 4.3-9 プロパティ - プロジェクトウィンドウ (一般 1)



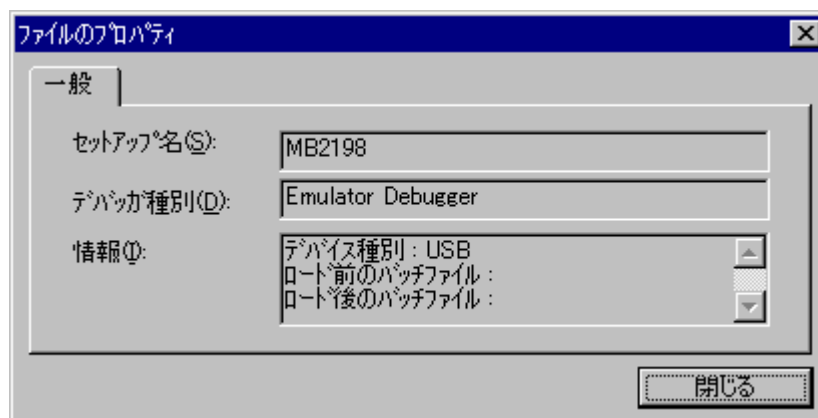
- ファイル名
  - 選択されているファイルのフルパスを表示します。
- ディレクトリ
  - 選択されているファイルのディレクトリを表示します。エディットが有効な場合は、ファイルの参照先を変更することができます。相対パスで記述すると、プロジェクトファイルに相対パスで記述されます。
- ファイルの種類
  - プロジェクトで管理しているファイルの種類が表示されます。
- サイズ
  - ファイルサイズを表示します。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。

図 4.3-10 プロパティ - プロジェクトウィンドウ (一般 2)



- フォルダ名
  - 選択されているフォルダのフォルダ名を表示します。エディットが有効な場合は変更が可能です。
- 場所
  - 選択されているフォルダの場所を表示します。

図 4.3-11 プロパティ - プロジェクトウィンドウ (一般 3)

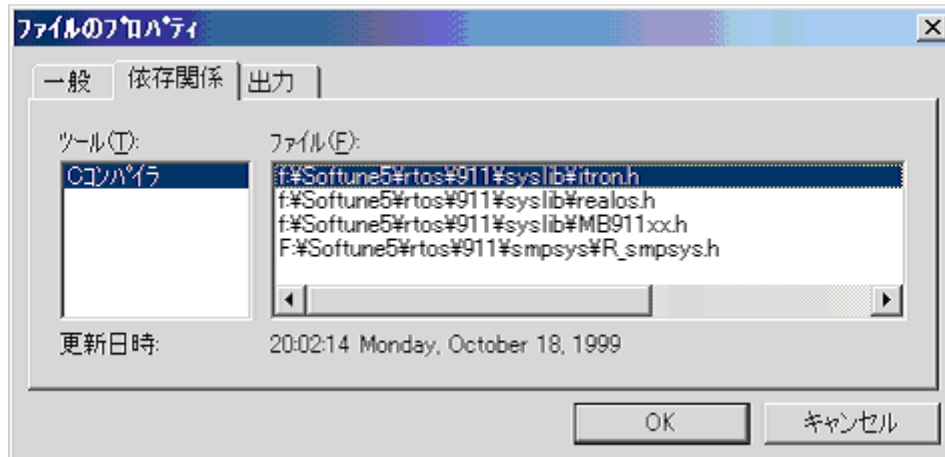


- セットアップ名
  - 選択されているセットアップ名を表示します。
- デバッガ種別
  - デバッガのセットアップに設定されているデバッガ種別を表示します。
- 情報
  - デバイス種別, ホスト名, ポート名, ボーレート, ロード前のパッチファイル名, ロード後のパッチファイル名などを表示します。表示される情報はデバイス種別等によって異なります。



## ■ プロパティ：依存関係

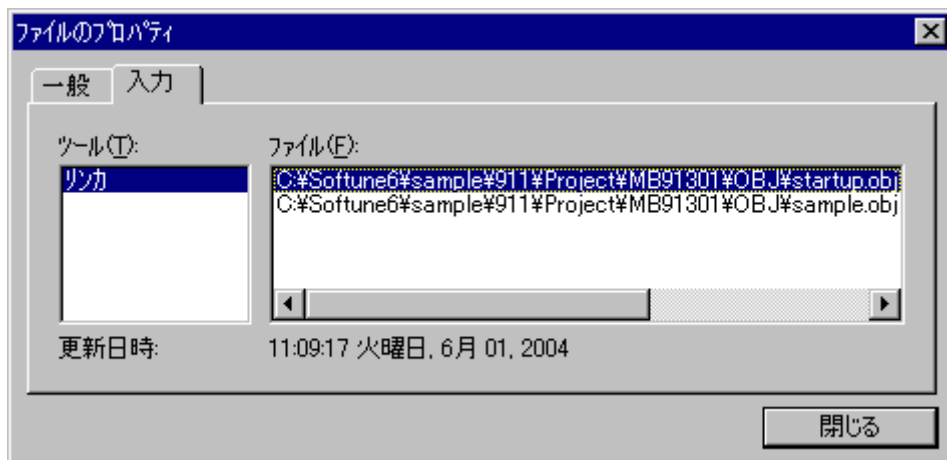
図 4.3-12 プロパティ - プロジェクトウィンドウ (依存関係)



- ツール
  - 実行される言語ツールを表示します。
- ファイル
  - 依存関係のあるファイルの一覧を表示します。ファイル名を選択することにより、そのファイルの更新日時が表示されます。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。

## ■ プロパティ：入力

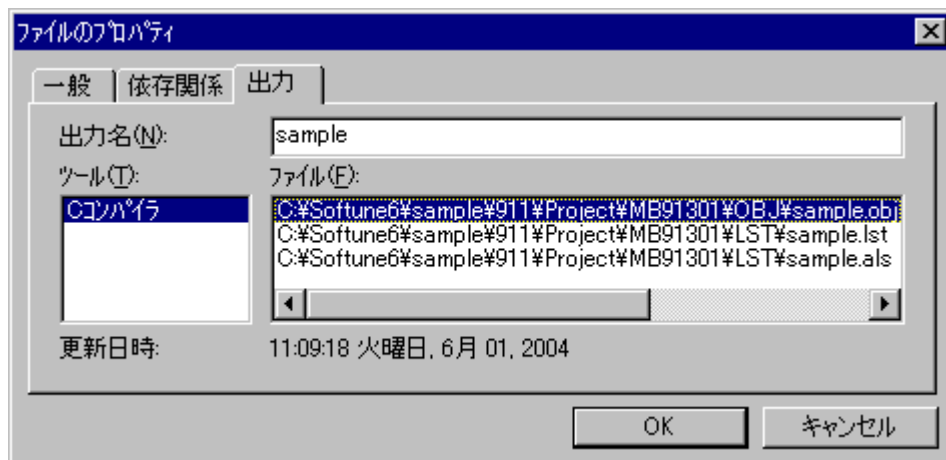
図 4.3-13 プロパティ - プロジェクトウィンドウ (入力)



- ツール
  - 実行される言語ツールを表示します。
- ファイル
  - ターゲットファイルを作成する時に使用されるファイルが順番に表示されます。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。

## ■ プロパティ：出力

図 4.3-14 プロパティ - プロジェクトウィンドウ (出力)



- 出力名
  - 選択されているファイルの言語ツールによって出力されるすべてのファイルの主ファイル名です。
- ツール
  - 実行される言語ツールを表示します。
- ファイル
  - 選択されているファイルの言語ツールによって出力されるファイルが表示されます。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。

### 4.3.9.2 プロパティ - エディットウィンドウ

ここでは、エディットウィンドウのプロパティについて説明します。

#### ■ プロパティ：一般

図 4.3-15 プロパティ - エディットウィンドウ (一般)



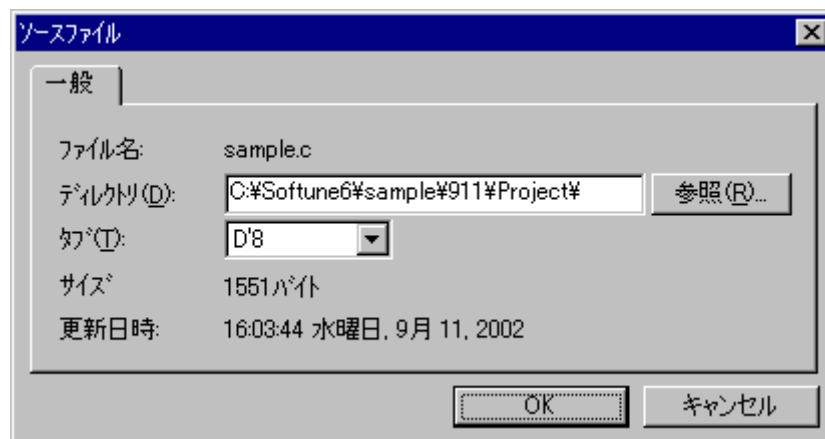
- ファイル名
  - 選択されているファイルのフルパスを表示します。
- サイズ
  - ファイルサイズを表示します。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。
- タブサイズ
  - タブコードのサイズを表示します。

### 4.3.9.3 プロパティ - ソースウィンドウ

ここでは、ソースウィンドウのプロパティについて説明します。

#### ■ プロパティ：一般

図 4.3-16 プロパティ - ソースウィンドウ (一般)



- ファイル名
  - 選択されているファイルのファイル名を表示します。
- ディレクトリ
  - ソースファイルが存在するディレクトリを表示します。ディレクトリを変更すると、そのディレクトリ中でファイルを探して表示し直します。
- タブ
  - タブコードのサイズ (空白文字に換算して何文字分になるか) を表示します。この値を変更すると、新しい設定値で再度表示し直します。
- サイズ
  - ファイルサイズを表示します。
- 更新日時
  - ファイルが最後に更新された日付と時刻を表示します。

### 4.3.9.4 プロパティ - シンボルウィンドウ

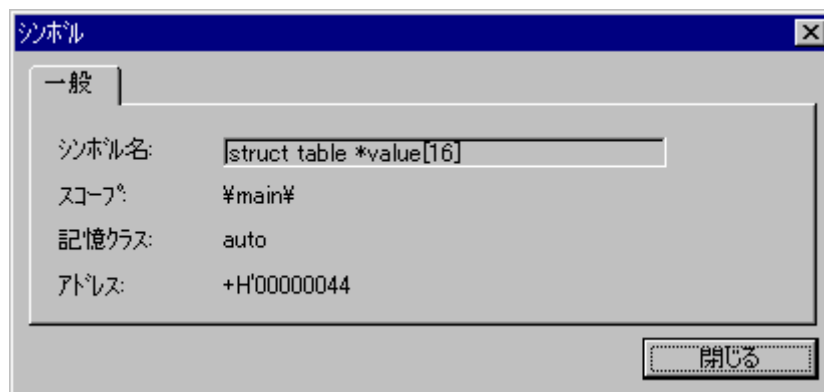
---

ここでは、シンボルウィンドウのプロパティについて説明します。

---

#### ■ プロパティ：一般

図 4.3-17 プロパティ - シンボルウィンドウ (一般)



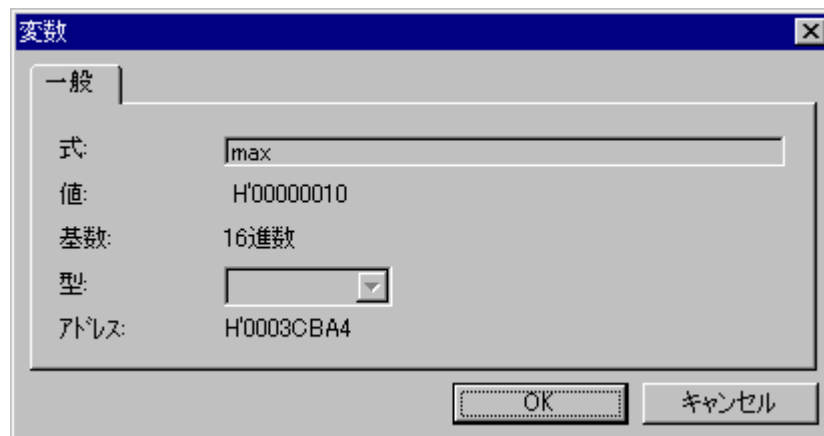
- シンボル名
  - 選択されているシンボル名を表示します。
- スコープ
  - 選択されているシンボルのスコープ位置を表示します。
- 記憶クラス
  - 選択されているシンボルの記憶クラスを表示します。
- アドレス
  - 選択されているシンボルのアドレスを表示します。

### 4.3.9.5 プロパティ - ローカルウィンドウ

ここでは、ローカルウィンドウのプロパティについて説明します。

#### ■ プロパティ：一般

図 4.3-18 プロパティ - ローカルウィンドウ (一般)



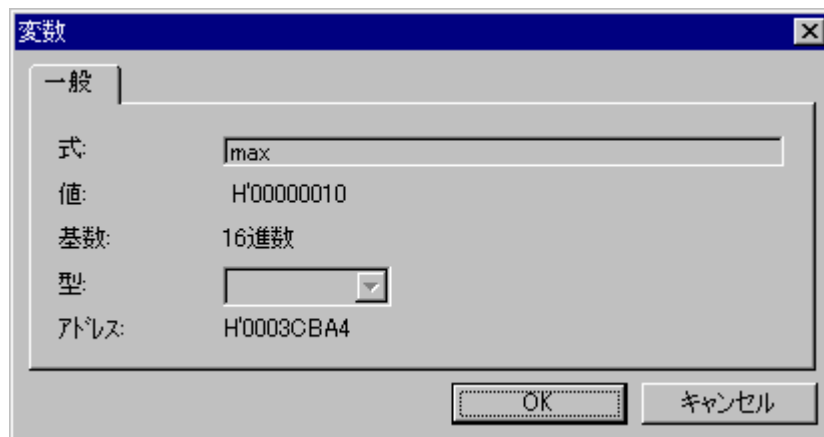
- 式
  - 選択されている項目を表示します。
- 値
  - 選択されている式の値を表示します。
- 基数
  - 値の表示基数を表示します。
- 型
  - 選択されている式の型を表示します。
- アドレス
  - シンボルアドレスを表示します。

### 4.3.9.6 プロパティ - ウォッチウィンドウ

ここでは、ウォッチウィンドウのプロパティについて説明します。

#### ■ プロパティ：一般

図 4.3-19 プロパティ - ウォッチウィンドウ (一般)



- 式
  - 選択されている項目を表示します。
- 値
  - 選択されている式の値を表示します。
- 基数
  - 値の表示基数を表示します。
- 型
  - 選択されている式の型を表示します。
- アドレス
  - シンボルアドレスを表示します。



## 4.4 表示メニュー

---

各ウィンドウを表示します。また、ツールバー、ステータスバーを表示または非表示にします。

---

### ■ ウィンドウ表示 / 非表示関係

以下のウィンドウの表示 / 非表示の設定ができます。

- プロジェクトウィンドウ
- アウトプットウィンドウ

### ■ ウィンドウ表示関係 (デバッグセッション)

デバッガの各ウィンドウを表示します。これはデバッグセッションでのみ可能です。

- シンボル一覧
- 逆アセンブル
- レジスタ
- メモリ
- ローカル変数
- ウォッチ
- トレース
- カバレッジ
- コマンド
- リアルタイムメモリ
- パフォーマンス
- RAM チェッカ

### ■ ツールバー / ステータスバー関係

ツールバー / ステータスバーの表示と非表示の設定ができます。また、ツールバーに表示するツールボタンのセットの選択ができます。

- ツールバー
- ステータスバー

### ■ タブ切り替えの表示 / 非表示関係

タブ切り替えの表示と非表示の設定ができます。タブ切り替えを表示することでエディット、ソース、メモリ、ウォッチウィンドウなどを容易に切り替えることができます。

- タブ

### ■ フォント関係

各ウィンドウのフォントを変更できます。

- フォント

## 4.4.1 プロジェクト/アウトプット

---

プロジェクトウィンドウおよびアウトプットウィンドウの表示/非表示の切り替えを行います。

---

### ■ プロジェクトウィンドウ

プロジェクトウィンドウが表示されているときは、メニューの[プロジェクト]の左側にチェックマークが付きます。

プロジェクトウィンドウを非表示にしてもプロジェクトは閉じられません。

再度表示すると、以前表示されていた位置に表示されます。

### ■ アウトプットウィンドウ

アウトプットウィンドウが表示されているときは、メニューの[アウトプットウィンドウ]の左側にチェックマークが付きます。

アウトプットウィンドウを非表示にしても、アウトプットウィンドウの内容はクリアされません。

再度表示すると、以前表示されていた位置に表示されます。

アウトプットウィンドウを非表示にした状態で、メイク/ビルドまたはコンパイル、アセンブル、依存関係の更新、中断を実行すると、自動的にアウトプットウィンドウが表示状態になります。

メイク/ビルドまたはコンパイル、アセンブル中に、アウトプットウィンドウを非表示にしても、エラーメッセージが失われることはありません。アウトプットウィンドウを再度表示すればすべて表示されます。

## 4.4.2 シンボル一覧

---

シンボルウィンドウを表示します。

---

### ■ シンボル一覧

シンボルウィンドウを開いて、そこに、ターゲットファイルで使用されているすべてのシンボルをツリー形式で一覧表示します。

この機能はデバッグセッション時のみ使用可能です。

既にシンボルウィンドウが開かれているときは、そのウィンドウをアクティブにします。

### 4.4.3 逆アセンブル

逆アセンブルウィンドウを表示します。

#### ■ 逆アセンブル

逆アセンブルウィンドウを開いて、そこに、指定されたアドレスからの逆アセンブル表示を行います。また、ショートカットメニューよりラインアセンブルも可能です。

この機能はデバッグセッション時のみ使用可能です。

- 逆アセンブルウィンドウが開かれている場合

そのウィンドウをアクティブにします。

- 逆アセンブルウィンドウが開かれていない場合

表示開始アドレスを指定するダイアログ ( 図 4.4-1 ) が表示されます。表示を始めるアドレスを指定して [OK] ボタンをクリックしてください。

図 4.4-1 表示開始アドレス指定ダイアログ



- 指定形式

表示開始位置を表す形式を指定します。( 行番号 / アドレス / フレーム )

- 表示位置

逆アセンブルを指定形式で示した位置から表示します。

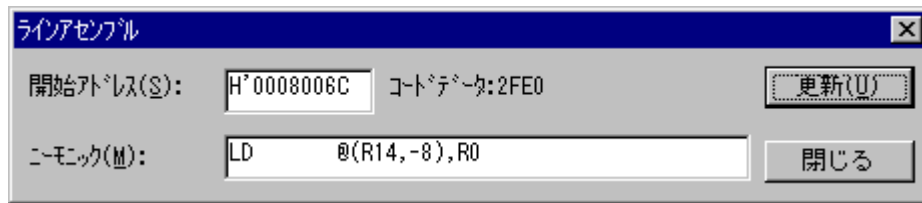
- ウィンドウ

ジャンプ先のウィンドウを指定します。詳細は「4.3.6 ジャンプ」を参照してください。逆アセンブルウィンドウの場合、デフォルトは「逆アセンブル」となります。

## ■ ラインアセンブル

ショートカットメニューより[ラインアセンブル]を選択すると、ラインアセンブルのダイアログ(図4.4-2)が開きます。

図 4.4-2 ラインアセンブルダイアログ



[ニーモニック]のエディットボックスにニーモニックを書き込んで、[更新]ボタンをクリックすると、今書き込んだニーモニックがアセンブルされて開始アドレスから設定されます。その後、開始アドレスは次のアドレスに自動的に進められます。

ニーモニックを書き込むアドレスを変更したいときは、[開始アドレス]を変更してください。

ニーモニックの変更が終了したら、[閉じる]ボタンをクリックしてください。

## 4.4.4 レジスタ

---

レジスタウィンドウを表示します。

---

### ■ レジスタ

レジスタウィンドウを開いて、そこに、ターゲット MCU の表示選択されているレジスタ名と、それぞれのレジスタの保持している値が表示されます。

この機能はデバッグセッション時のみ使用可能です。

既にレジスタウィンドウが開かれているときは、そのウィンドウをアクティブにします。

### ■ レジスタ値の変更

レジスタウィンドウに表示されているレジスタの保持している値を、ここで直接変更することができます。変更は手順で行います。

#### ● 完全に変更するとき

- 1) レジスタ名または表示値をダブルクリックしてください。
  - レジスタ名または表示値が反転表示されます。
- 2) 設定したい値を上位桁（左側の桁）から順に 16 進数で指定してください。
  - 最下位ビットの桁（最も右側の桁）まで修正したところで、自動的に設定されます。

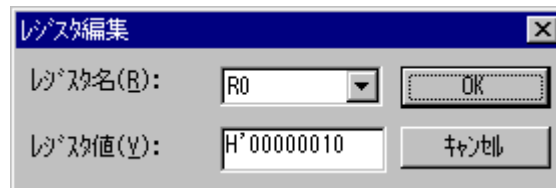
#### ● 一部の桁（ビットフィールド）を変更するとき

- 1) 変更したいレジスタの値表示の変更したい桁をクリックしてください。
  - クリックした桁の左側にカーソルが現れます。
- 2) 変更したい値（16 進数 1 桁）を設定してください。
  - カーソルが自動的に右隣の桁に移動します。もし、変更した桁が、最下位の桁であった場合はその時点で自動的にレジスタ値が更新されます。
- 3) 必要な桁数だけの変更が終わったら、今変更した以外のレジスタ名またはレジスタ値表示をクリックしてください。
  - この操作を行わずにレジスタウィンドウを閉じてしまった場合は、変更した値はレジスタに設定されません。
  - 最下位の桁まで変更したときはこの操作は不要です。

## ● レジスタ名を選択して変更するとき

- 1) レジスタウィンドウで右ボタンをクリックしてショートカットメニューを表示し、  
「編集」を選択してください。
  - レジスタ編集ダイアログ (図 4.4-3) が表示されます。
- 2) レジスタ名を選択してください。
- 3) 設定する値を入力してください。
- 4) [OK] ボタンをクリックしてください。

図 4.4-3 レジスタ編集ダイアログ



- レジスタ名  
レジスタ名を指定します。
- レジスタ値  
レジスタに設定する値を指定します。

## 4.4.5 メモリ

メモリウィンドウを表示します。

### ■ メモリ内容の表示

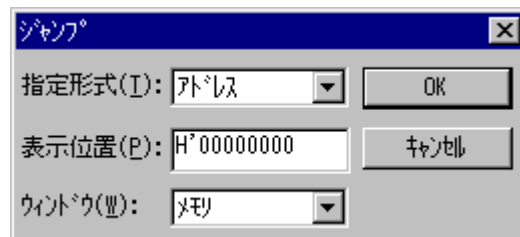
メモリウィンドウを開いて、そこに、指定されたアドレスからのメモリ内容のダンプ表示を行います。また、ショートカットメニューよりメモリ内容の変更も可能です。

この機能はデバッグセッション時のみ使用可能です。

- メモリウィンドウが開かれている場合  
そのウィンドウをアクティブにします。
- メモリウィンドウが開かれていない場合

表示開始アドレスを指定するダイアログ (図 4.4-4) が表示されます。

図 4.4-4 表示開始アドレス指定ダイアログ



- ・ 指定形式  
表示開始位置を表す形式を指定します。(行番号 / アドレス / フレーム)
- ・ 表示位置  
逆アセンブルを指定形式で示した位置から表示します。
- ・ ウィンドウ  
ジャンプ先のウィンドウを指定します。詳細は「4.3.6 ジャンプ」を参照してください。メモリウィンドウの場合、デフォルトは「メモリ」となります。

### ■ メモリ内容の変更

ダンプ表示されている値を直接書き換えることでメモリの内容を変更することができます。アスキー文字列表示を行っている場合はそのフィールドに文字列を入力すると、自動的に対応するアドレスにそれぞれの文字のアスキーコードが設定されます。

アドレスフィールドの表示を変更することで、メモリダンプを開始するアドレスを任意に変更することができます。



## ■ ブックマーク追加

メモリウィンドウ内で右ボタンをクリックしてショートカットメニューを表示し、[ブックマーク追加]を選択してください。ブックマーク追加のダイアログが表示されます。(図 4.4-5、図 4.4-6 参照)

図 4.4-5 ブックマーク追加ダイアログ (アドレス)

The dialog box titled 'ブックマーク追加' (Bookmark Add) contains the following fields and controls:

- ブックマーク名(B):** Text input field containing 'H^00001000'. To its right is an 'OK' button.
- 色(C):** A color selection dropdown menu. To its right is a 'キャンセル' (Cancel) button.
- 範囲指定 (Range Specification):** A group box containing two radio buttons:
  - ☒ アドレス(A) (Address)
  - ☐ シンボル(S) (Symbol)
- 開始アドレス(I):** Text input field containing 'H^00001000'.
- 終了アドレス(E):** Text input field containing 'H^0000113F'.

図 4.4-6 ブックマーク追加ダイアログ (シンボル)

The dialog box titled 'ブックマーク追加' (Bookmark Add) contains the following fields and controls:

- ブックマーク名(B):** Text input field containing 'target [0]'. To its right is an 'OK' button.
- 色(C):** A color selection dropdown menu. To its right is a 'キャンセル' (Cancel) button.
- 範囲指定 (Range Specification):** A group box containing two radio buttons:
  - ☐ アドレス(A) (Address)
  - ☒ シンボル(S) (Symbol)
- 変数名(N):** Text input field containing 'target [0]'.

### ● ブックマーク名

ブックマークの名前を指定します。デフォルトは「範囲指定」の設定により以下のようになります。

範囲指定が「アドレス」の場合：開始アドレス

範囲指定が「シンボル」の場合：変数名

### ● 色

ブックマークの背景色を指定します。

### ● 範囲指定

ブックマークのアドレス範囲の指定方法を指定します。

アドレス：開始アドレス，終了アドレスで指定します。

シンボル：シンボル名でアドレス範囲を指定します。

- 開始アドレス

ブックマークの開始アドレスを指定します。

- 終了アドレス

ブックマークの終了アドレスを指定します。

- 変数名

ブックマークを設定するシンボル名を指定します。

## ■ メモリ内容の検索

メモリウィンドウ内で右ボタンをクリックしてショートカットメニューを表示し、「検索」を選択してください。

検索ダイアログ (図 4.4-7) が表示されます。

図 4.4-7 検索ダイアログ



- 開始アドレス

検索する範囲の開始アドレスを指定してください。

- 終了アドレス

検索する範囲の終了アドレスを指定してください。

- データ種別

データ種別から選択してください (バイト / ハーフワード / ワード / ダブルワード / アスキー)。

- 検索データ

データ種別にあったデータを指定してください。アスキーを選択した場合は文字列を指定してください。アスキー以外のデータ種別で、連続した複数のデータを検索したい場合は、個々のデータをコンマで区切って記述してください。

- スキップデータ数

検索範囲を連続して検索する場合は、1 を設定してください。2 以上の値を設定した場合は、そのバイト数ごとにスキップして検索します。

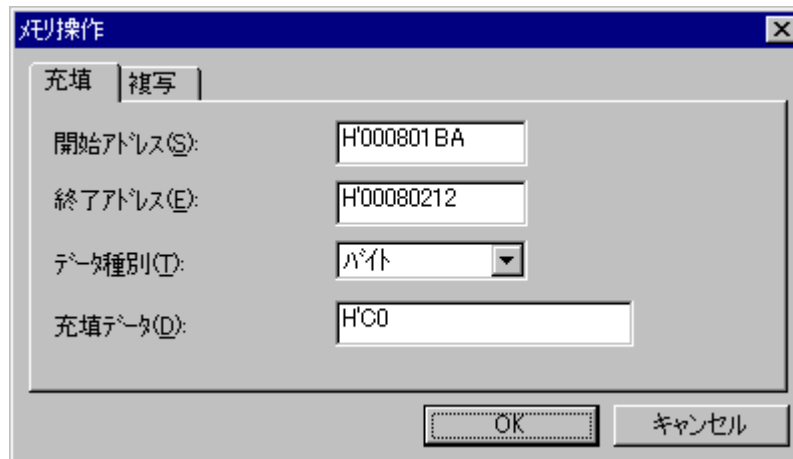
## ■ 特殊操作

メモリ範囲へのデータ充填・複写する場合は、デバッグを開始してメモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「特殊操作」を選択してください。メモリ操作のダイアログが表示されます。

[OK] ボタンを押すと、開かれているタブの機能を開始します。

### ● 充填

図 4.4-8 メモリ操作ダイアログ ( 充填 )

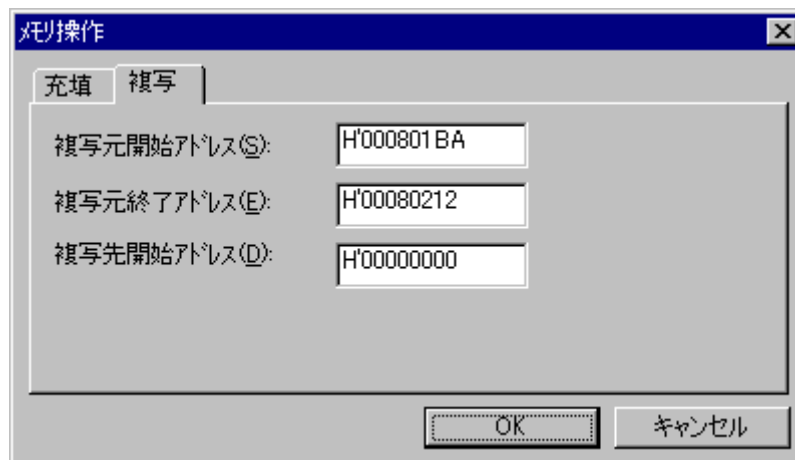


充填したい場合は、「充填」タブを開いて、開始アドレス、終了アドレス、データ種別、充填データを設定してください。

- 開始アドレス
  - 充填する領域の先頭アドレスを指定してください。このアドレスから充填データが入ります。
- 終了アドレス
  - 充填する領域の最終アドレスを指定してください。このアドレスまで充填データが入ります。
- データ種別
  - 充填データの種別を指定します ( バイト / ハーフワード / ワード / ダブルワード / アスキー )。
- 充填データ
  - 指定したメモリ領域に充填するデータです。"," でデータを区切ることにより、複数データを指定できます。

● 複写

図 4.4-9 メモリ操作ダイアログ (複写)



複写したい場合は、[複写]タブを開いて、複写元開始アドレス、複写元終了アドレス、複写先開始アドレスを設定してください。

- 複写元開始アドレス
  - 複写元領域の先頭アドレスを指定してください。このアドレスのデータからコピーを開始します。
- 複写元終了アドレス
  - 複写元領域の最終アドレスを指定してください。このアドレスのデータまでがコピーされます。
- 複写先開始アドレス
  - 複写先領域の先頭アドレスを指定してください。

## ■ メモリブロックの比較

メモリブロックの比較をする場合は、メモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「比較」を選択してください。比較のダイアログが表示されます（図 4.4-10 参照）。

### ● 比較元開始アドレス

比較元領域の先頭アドレスを指定してください。このアドレスのデータから比較を開始します。

### ● 比較元終了アドレス

比較元領域の最終アドレスを指定してください。このアドレスのデータまでが比較されます。

### ● 比較先開始アドレス

比較先領域の先頭アドレスを指定してください。

図 4.4-10 比較ダイアログ



例えば、スキップに 4 を指定した場合は、0 番地の次は 4 番地、8 番地、...というように 4 バイトごとにスキップして検索します。

### 【例】

00000000 01 02 03 01 02 01 02 03 04 01 01 02 03 のようなデータがある場合、

データ種別：バイト 検索データ：01 スキップ：1 のときは、すべての 01 が発見されますが、スキップ：2 の時は、00000000 番地と 0000000A 番地にある 01 のみが発見され、それ以外の 01 は発見されません。

データ種別：バイト 検索データ：01, 02 スキップ：3 のときは、00000000 番地と 00000003 番地にある 01, 02 のみが発見され、00000005 番地と 0000000A 番地の 01, 02 は発見されません。

## ■ 編集

メモリ内容の編集をする場合は、メモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「編集」を選択してください。編集のダイアログが表示されます（図 4.4-11）。

- アドレス

編集アドレスを指定します。

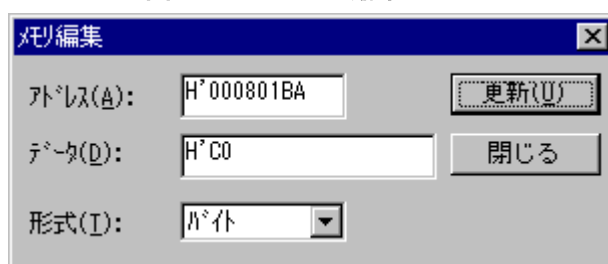
- データ

書き換えるメモリ内容を指定します。

- 形式

書き換えるデータサイズを指定します（バイト/ハーフワード/ワード/ダブルワード）。

図 4.4-11 メモリ編集ダイアログ



## ■ 表示設定

メモリウィンドウの表示形式を設定する場合は、メモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「表示設定」を選択してください。表示設定のダイアログが表示されます（図 4.4-12）。

- 表示形式

ウィンドウの表示形式を指定します。

ビット、バイト、ハーフワード、ワード、ダブルワードの中から選択できます。

- 表示形式

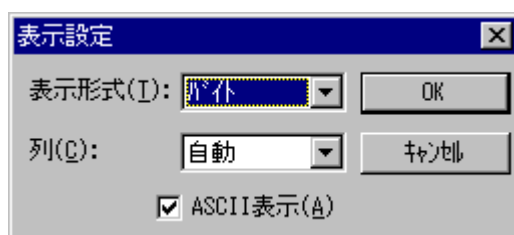
ウィンドウ右側の ASCII 文字が表示されるかどうかを選択します。

- 列

一行に表示するバイト数を指定します。

自動、4 バイト、8 バイト、16 バイト、32 バイト、64 バイトの中から選択できます。

図 4.4-12 表示設定ダイアログ



## 4.4.6 ローカル変数

ローカル変数ウィンドウを表示します。

### ■ ローカル変数表示

現在の命令ポイントの存在する関数のローカル変数を、関数名をルートにしてツリー表示します。ここでは、表示する変数を追加したり、表示を取り消したりすることはできません。

変数値がプログラムの実行によって変更された場合は、自動的に更新されますので、プログラムの実行によって変数値がどのように変化していくを見ることができます。また、変数値を変更してデバッグを継続することができます。

### ■ 基数設定

変数値を表示するときの基数を設定します。以下の手順で、変数ごとに個々に設定できます。

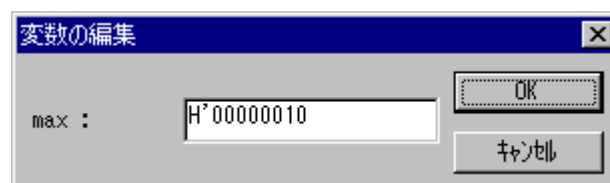
- 1) 基数を変更したい変数の行で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [ 基数 ] にマウスカーソルをあわせてください。
  - 選択可能な基数のリストがサブメニューに表示されます。
- 3) 基数を選択してください。

### ■ 変数値の変更

以下の手順で、変数値を変更することができます。

- 1) 変更したい変数のある行で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [ 編集 ] を選択してください。
  - 変数の編集ダイアログ ( 図 4.4-13 ) が表示されます。
- 3) 変数値を設定して、[OK] ボタンをクリックしてください。

図 4.4-13 変数の編集



## 4.4.7 ウォッチ

ウォッチウィンドウを表示します。

### ■ ウォッチウィンドウ

ウォッチウィンドウには指定した変数の値がツリービュー形式で表示されます。設定した変数値がプログラムの実行によって変更された場合は、自動的に更新されますので、プログラムの実行によって変数値がどのように変化していくかを見ることができます。

ショートカットメニューにより表示された変数値の変更などができます。

表示する変数の設定は以下の手順で行います。

- 1) ウィンドウ内で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [設定] をクリックしてください。
  - ウォッチ設定ダイアログ (図 4.4-14) が表示されます。
- 3) 変数名を入力してください。また、必要に応じて [モード] も選択してください。
- 4) [ウォッチ] で表示するウォッチウィンドウの番号を選択してください。
- 5) [OK] ボタンをクリックしてください。

図 4.4-14 ウォッチ設定ダイアログ



- 変数名
  - 表示する変数を指定します。
- モード
  - 表示する変数を C 言語式か、アセンブラ言語式かを指定します。自動の場合は、変数が定義された言語式になります (自動 / C 言語 / アセンブラ)。
- ウォッチ
  - 変数を設定するウォッチウィンドウを番号で指定します。
- リアルタイム自動設定
  - 変数アドレスを自動的にリアルタイム領域に設定するかどうかを指定します。

### ■ 基数設定

変数値を表示するときの基数を設定します。以下の手順で、変数ごとに個々に設定できます。

- 1) 基数を変更したい変数の行で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [基数] にマウスカーソルをあわせてください。
  - 選択可能な基数のリストがサブメニューに表示されます。
- 3) 基数を選択してください。

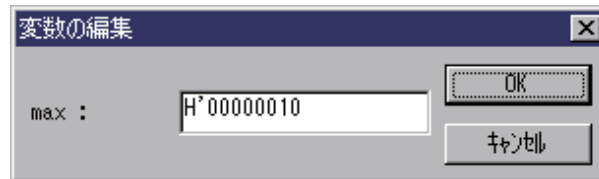


## ■ 変数値の変更

以下の手順で、変数値を変更することができます。

- 1) 変更したい変数のある行で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [編集]を選択してください。
  - 変数の編集ダイアログ (図 4.4-15) が表示されます。
- 3) 変数値を設定して、[OK] ボタンをクリックしてください。

図 4.4-15 変数の編集ダイアログ



## ■ 不要になった変数表示の取り消し

一度設定した変数を見る必要がなくなったときは、以下の手順でウィンドウから表示を消すことができます。

- 1) 表示を取り消したい変数の行で右ボタンをクリックしてください。
  - ショートカットメニューが表示されます。
- 2) [削除]を選択してください。

### < 注意事項 >

配列の要素や、ストラクチャなどのメンバを、個々に表示取り消しはできません。もし、各要素や各メンバの行で [削除] を実行した場合は、その配列やストラクチャそのものの表示が取り消されます。

## ■ リアルタイム領域設定

ウォッチ変数を簡単にリアルタイム領域に設定できます。変数をリアルタイムでモニタリングする場合に便利です。

設定方法は2とおりあります。

- 1) ウォッチ変数の新規登録時。
    - ウォッチ設定ダイアログ (図 4.4-14) で [リアルタイム自動設定] にチェック。
  - 2) 既に登録済のウォッチ変数
    - ウォッチウィンドウの変数上で右クリックし [リアルタイム] メニューを選択。
- 上記のいずれもリアルタイム領域 (256byte × 2 領域) の設定状況により下記のように設定されます。
- リアルタイム領域に未設定の領域がある場合
    - 未設定の領域に設定されます。デフォルトは領域 1 となります。
  - リアルタイム領域が 2 領域ともに設定済で、ウォッチ変数が 2 領域の範囲外の場合
    - リアルタイム領域を変更するかどうかの確認ダイアログ (図 ) を表示します。

図 4.4-16 リアルタイム領域選択ダイアログ

リアルタイム領域変更

リアルタイム領域は既に2領域とも設定されています。  
下記内容を設定する場合、領域番号を指定してください。

リアルタイム領域への設定内容

アドレス範囲 : H'00034000 ... H'000340FF

変数情報

変数名 : val アドレス : H'0003400C

上書きするリアルタイム領域番号を指定

領域番号

☒ 領域1(A) ☐ 領域2(B)

領域	アドレス範囲	ウォッチ登録数
1	0x00034300 .. 0x000343FF	0
2	0x00034200 .. 0x000342FF	0

OK キャンセル

#### ● アドレス範囲

設定する（または設定済の）ウォッチ変数のアドレスを表示します。このアドレスを基に、リアルタイム領域に設定したいアドレスを入力します。

終了アドレスは開始アドレスから 256byte になるように表示され、変更はできません。

#### ● 変数情報

設定する（または設定済の）ウォッチ変数の変数名とアドレスを表示します。変更はできません。

なお指定した変数のサイズが 256byte 以上の場合、次のメッセージが表示されます。

「サイズが 256byte 超過のため、アドレス範囲にご注意ください」

#### ● 領域番号

上記のウォッチ変数を基に、上書きするリアルタイム領域番号を設定します。

#### < 注意事項 >

- ウォッチ変数がシンボル以外の場合、リアルタイム領域の自動設定は行いません。
- ウォッチ変数が構造体や配列、クラスなどの場合、親のみカウントされるためウォッチ登録数は 1 となります。

## 4.4.8      トレース

---

トレースウィンドウを表示します。

---

### ■ トレース

今まで実行したアドレスと命令をさかのぼって表示する機能です。デバッグの途中で、有効 / 無効の切り替えができます。トレース結果の表示は、機械命令ごとの表示、サイクル表示、ソースごとの表示のいずれかを選択できます。

### ■ 更新

トレースウィンドウの表示は、デバッグの進行にあわせてリアルタイムでの表示変更はしません。したがって、最新のトレース表示を行うには、トレースウィンドウで右ボタンをクリックしてショートカットメニューを表示し、[ 更新 ] を選択してください。

### ■ バッファサイズ

トレースデータは、トレースバッファにバッファリングされています。トレースバッファは無限ではありませんので、デバッグが進行していくとそのうちに一杯になってしまいます。トレースバッファが一杯になったときに、実行中のプログラムを停止させることができます。

---

#### < 注意事項 >

FR80S 使用時は内蔵 RAM 領域へのライトアクセスのデータのみトレースバッファに格納されます。ただし MCU 動作モードが「外部トレースモード」以外の場合はデータが格納されません。

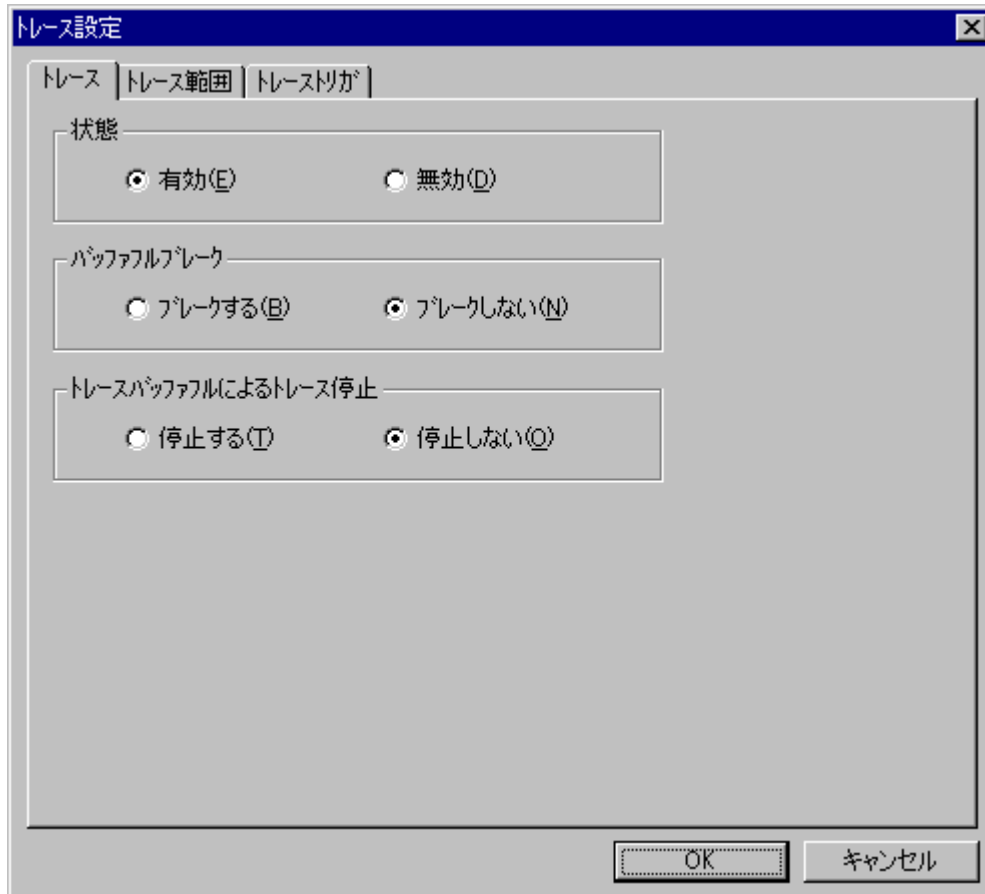
---

## ■ トレース設定

ショートカットメニューから「設定」を選択してください。

### ● トレース

図 4.4-17 トレース設定ダイアログ



- 状態
  - 制御状態を指定します。
- バッファフルブレーク
  - トレースバッファフルブレークをするか否かを指定します。
- トレースバッファフルによるトレース停止
  - トレースバッファフルによるトレースの停止をするか否か指定します。MB2198で表示されます。

## ● データトレース範囲【DSU3, DSU4】

図 4.4-18 データトレース範囲

トレース設定

トレース | **トレース範囲** | トレーストリガ

取得データ

☒ リード(R)    ☒ ライト(W)    ☒ コード(C)

表示範囲

状態

☒ 有効(E)    ☐ 無効(D)

アドレス(A): H'00000000

アドレスマスク(M): H'FFFFFFFF

OK    キャンセル

- 取得データ  
トレース取得をおこなうデータ属性を指定します。  
コード属性は DSU4 の場合のみ有効です。
- 状態  
指定範囲のトレースを取得するか否かを指定します。  
DSU4 の場合のみ有効です。
- アドレス  
トレース取得をおこなうアドレスを指定します。
- アドレスマスク  
トレース取得をおこなうアドレスのマスク値を指定します。

## &lt; 注意事項 &gt;

MCU 動作モードと DSU 種別により，以下のようにトレースデータを取得します。

MCU 動作モード	DSU 種別	
	DSU3	DSU4
フルトレースモード	コード，[ リード ]，[ ライト ]	コード，[ リード ]，[ ライト ]
リアルタイムモード	コード，[ リード ]，[ ライト ]	コード，[ リード ]，[ ライト ]
外部トレースモード	コード，[ リード ]，[ ライト ]	アダプタボード接続 [ コード ]，[ リード ]，[ ライト ]
		ケーブル接続 [ コード ]，[ リード ]，[ ライト ]
内部トレースモード	コード，リード，ライト	コード，リード，ライト

[    ] の属性のデータは取得するか否かの指定が可能です。

## ● トレーストリガ【FR60Lite, FR80S】

図 4.4-19 トレーストリガ

トレース設定

トレース | トレース範囲 | トレーストリガ

アドレス(R): H'00010000 設定(S)

種別(Y): データ 詳細(D)...

トレース動作

☒ トレース開始(C) ☐ トレース停止(P) イベントリスト(V)...

残り個数: 1

設定内容(L)

状態	種別	アドレス	動作	シンボル
enable	Data	00010000	Start	

有効(E) 無効(D) 削除(D) 全削除(A) 変更(H)...

OK キャンセル

- アドレス  
トリガポイントを設定するアドレスまたはシンボルを指定します。
- 種別  
トレーストリガの種別を指定します。(コード/データ)  
ただし FR80S ではデータのみとなります。
- トレース動作  
設定したトレーストリガがヒットした場合のトレース動作を指定します。  
(トレース開始/トレース停止)
- 残り個数  
選択中のトレーストリガ種別の設定可能個数を表示します。
- 設定内容  
現在設定されているトレーストリガの内容を表示します。
- [設定] ボタン  
設定されたアドレスにトレーストリガを設定します。
- [詳細] ボタン  
アドレス以外の詳細条件を設定する、トレーストリガ詳細設定ダイアログを表示します。

- [ イベントリスト ] ボタン  
全イベントの設定状況を確認するためのイベント一覧ダイアログを表示します。
- [ 有効 ] ボタン  
トレーストリガリスト中で選択されているトレーストリガを有効にします。
- [ 無効 ] ボタン  
トレーストリガリスト中で選択されているトレーストリガを無効にします。  
トレーストリガが無効になるだけで、設定そのものが解除されたわけではありません。
- [ 削除 ] ボタン  
トレーストリガリスト中で選択されているトレーストリガの設定を削除します。
- [ 全削除 ] ボタン  
トレーストリガリスト中ですべてのトレーストリガの設定を削除します。
- [ 変更 ] ボタン  
トレーストリガリスト中の選択されたトレーストリガの設定を変更します。

< 注意事項 >

- FR60Lite または FR80S 使用時以外ではこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.7 トレース」を参照してください。
- FR80S では内蔵 RAM 領域に設定した場合のみ有効となります。

● トレーストリガの詳細設定ダイアログ【FR60Lite, FR80S】

図 4.4-20 トレーストリガの詳細設定ダイアログ





- トレース動作  
設定したトレーストリガがヒットした場合のトレース動作を指定します。  
(トレース開始/トレース停止)
- アドレス  
トレーストリガを設定するアドレスを指定します。
- アドレスマスク  
指定したアドレスに対するマスク値を指定します。
- サイズ (バイト / ハーフワード / ワード)  
データアクセス時のデータサイズを指定します。
- 属性  
データアクセス時の属性を指定します。  
ただし FR80S では指定できず、必ずライト属性となります。
- サイズ無視  
アクセスデータ長にかかわらず、指定アドレスがアクセスされた場合をトリガ条件にします。
- 比較条件  
データ比較条件を指定します。  
無効 : トリガ条件にデータを指定しません。  
データ一致 : 指定データと一致した場合をトリガ条件にします。  
データ NOT : 指定データと一致しない場合をトリガ条件にします。
- データ  
トリガ条件となるデータを指定します。
- データマスク  
指定したデータに対するマスク値を指定します。

---

< 注意事項 >

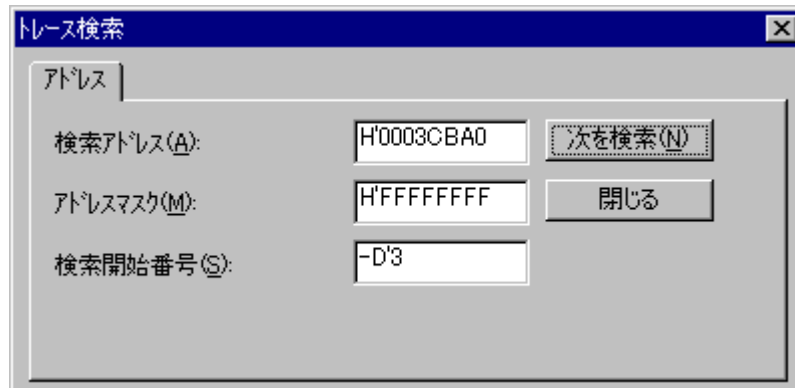
- FR60Lite または FR80S 使用時以外ではこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.7 トレース」を参照してください。
  - FR80S では内蔵 RAM 領域に設定した場合のみ有効となります。
-

## ■ トレース検索

ショートカットメニューから [ 検索 ] を選択してください。

### ● アドレス

図 4.4-21 トレース検索ダイアログ (アドレス)



- 開始アドレス
  - 検索する範囲の開始アドレスを指定してください。
- 終了アドレス
  - 検索する範囲の終了アドレスを指定してください。
- 検索開始番号
  - 検索を開始するフレーム番号を指定します。

## ■ アドレスマスクについて (例)

検索アドレスとアドレスマスクと、実際に見つかるアドレスとの関係は、例えば表 4.4-1 のようになります。

表 4.4-1 検索アドレス・アドレスマスク・見つかるアドレスの関係

検索アドレス	アドレスマスク	マッチするアドレス
H'F000F0CA	H'FFFFFFFF	H'F000F0CA
	H'FFFF0000	H'F0000000 から H'F000FFFF までの任意のアドレス

## ■ バックトレース

ショートカットメニューから [ バックトレース ] を選択してください。

図 4.4-22 バックトレースウィンドウ



### ● フレーム番号

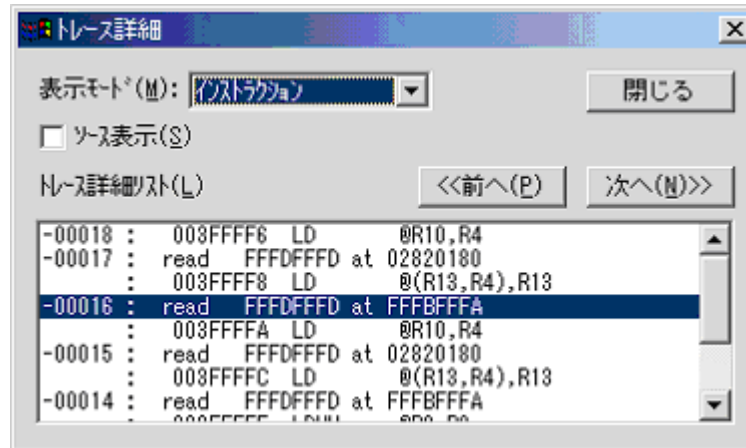
トレースウィンドウに対応するフレーム番号を指定します。

## ■ トレース詳細【DSU3】

ショートカットメニューから「詳細」を選択してください。

1 分岐区間ごとの詳細なトレースデータを表示します。

図 4.4-23 トレース詳細ダイアログ



### ● 表示モード

トレースデータの表示について、インストラクション表示で行うかソース表示で行うかを指定します。

### ● ソース表示

インストラクション表示のとき、ソース行情報がある場合にはソース行も混在させて表示できます。

### ● 前へ

現在表示している分岐区間の直前に実行された 1 分岐区間分の情報を表示します。

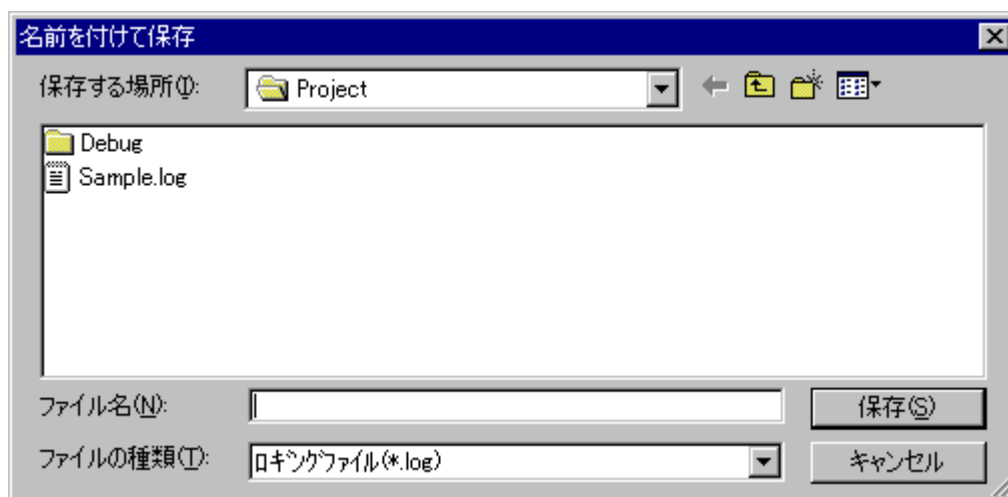
### ● 次へ

現在表示している分岐区間の直後に実行された 1 分岐区間分の情報を表示します。

## ■ セーブ

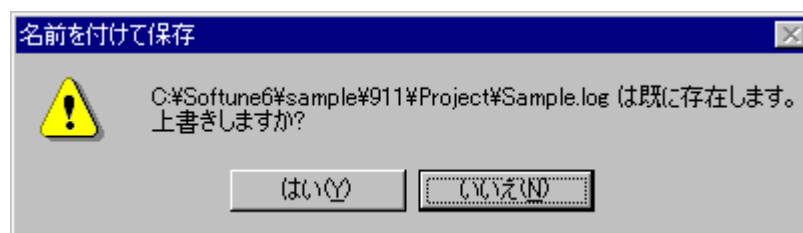
ショートカットメニューから [ セーブ ] を選択して下さい。  
指定したファイル名でトレースデータをセーブすることができます。

図 4.4-24 トレースデータセーブダイアログ



ファイル名指定時に、既存ファイルを指定すると上書き保存するか否か選択します。

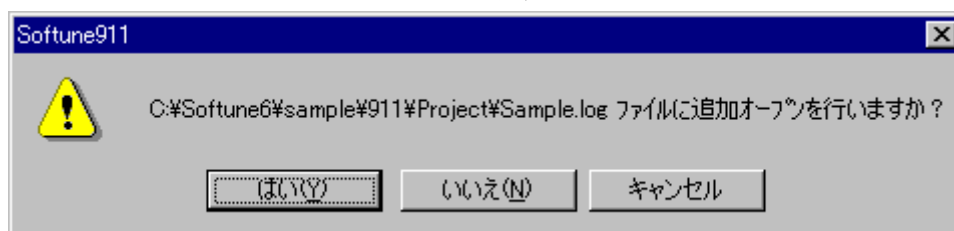
図 4.4-25 上書き保存



[ はい ]: 追加保存するか否か選択します。

[ いいえ ]: トレースデータをファイルに保存しません。

図 4.4-26 追加保存



[ はい ]: トレースデータを追加保存します。

[ いいえ ]: トレースデータを上書き保存します。

[ キャンセル ]: トレースデータの保存をキャンセルします。

## 4.4.9 コマンド

---

コマンドウィンドウを表示します。

---

### ■ コマンド

コマンドウィンドウを表示して、そこでデバッガのコマンドを直接入力して実行することができます。コマンドウィンドウで実行した結果は、コマンドウィンドウに表示されます。使用可能なコマンドについては、『SOFTUNE Workbench コマンドリファレンス』を参照してください。

## 4.4.10 ツールバー / ステータスバー / タブ

---

ツールバーの表示内容の設定，ステータスバー，タブ切り替えの表示 / 非表示の切り替えを行います。

---

### ■ ツールバー

ツールバーに表示するツールボタンの組を，次に示す中から任意に選択できます。

- 共通
- 検索
- ビルド
- デバッグ
- フラグ
- プロジェクト

それぞれの組に含まれるボタンについては，「3.2 ツールバー」を参照してください。

### ■ ステータスバー

ステータスバーは，表示 / 非表示の切り替えのみができます。ステータスバーの表示内容については，「3.3 ステータスバー」を参照してください。

### ■ タブ

タブ切り替えの表示 / 非表示の切り替えができます。タブ切り替え表示にすると，各ウィンドウがタブ付きで表示され，ウィンドウ間の切り替えが容易になります。

## 4.4.11 フォント

---

各ウィンドウのフォントを変更します。

---

### ■ フォントの設定

各ウィンドウで現在設定されているフォント情報（フォント名やサイズ）を表示したり，その設定を変更することができます。またフォントの設定をすべて初期状態にリセットすることも可能です。

デバッグ中には，デバッグ関連のウィンドウ（ソースウィンドウ等）のフォントを変更することが可能となります。

### ■ フォントの変更

以下の手順でフォントの変更を行います。

- 1) [表示]-[フォント]メニューを選択してください。
  - フォントの設定ダイアログ（図 4.4-27）が表示されます。
- 2) 変更したいウィンドウを選択し [フォント変更] ボタンをクリックしてください。
  - フォントの設定ダイアログ（図 4.4-28）が表示されます。
- 3) フォント名とサイズを指定して [OK] ボタンをクリックしてください。ウィンドウにより選択できるフォント種別は異なります。
- 4) フォントの設定ダイアログ（図 4.4-27）の [OK] ボタンをクリックするとウィンドウのフォントが変更されます。

### ■ フォントのリセット

以下の手順でフォントのリセットを行います。

- 1) [表示]-[フォント]メニューを選択してください。
  - フォントの設定ダイアログ（図 4.4-27）が表示されます。
- 2) [すべてリセット] ボタンをクリックしてください。
- 3) [ウィンドウ] に表示されているウィンドウのフォントがすべて初期状態にリセットされます。
- 4) [OK] ボタンをクリックするとウィンドウのフォントが初期状態に戻ります。



図 4.4-27 フォントの設定

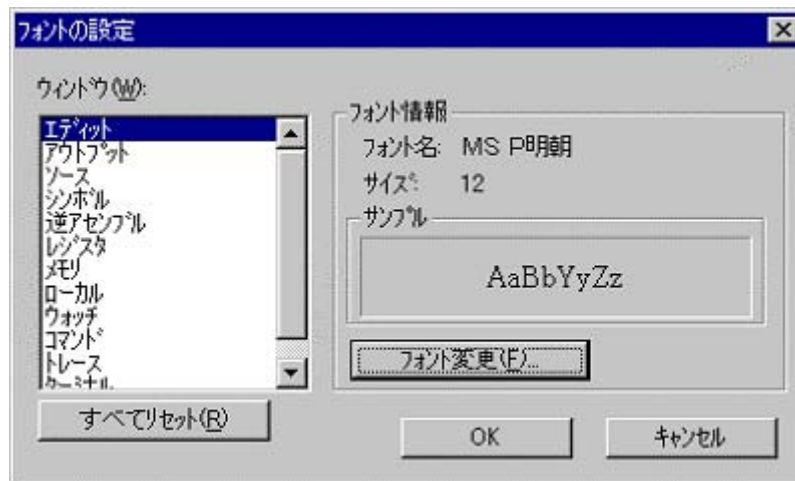


図 4.4-28 フォントの設定 (2)



## 4.4.12 カバレッジ

カバレッジウィンドウを表示します。

### ■ カバレッジウィンドウ

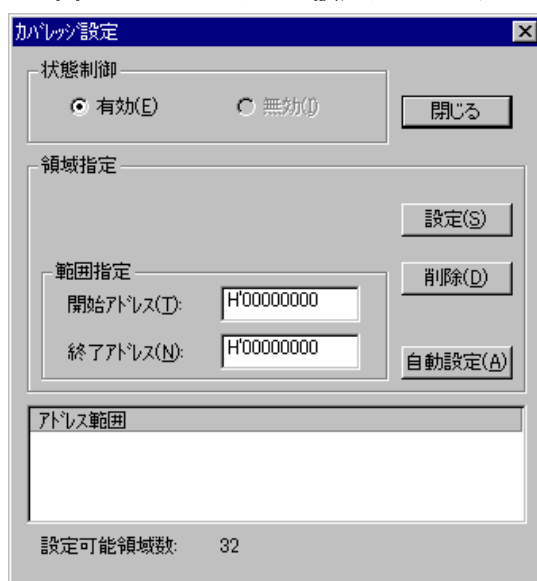
カバレッジ測定結果を表示するウィンドウを開きます。カバレッジの設定や全カバレッジ率の表示がおこなえます。

この機能は高速版シミュレータデバッガで使用可能です。

### ■ カバレッジ設定

カバレッジウィンドウのショートカットメニューの[設定]を選択してください。

図 4.4-29 カバレッジ設定ダイアログ



#### ● 状態制御

有効または無効を選択します。

高速版シミュレータデバッガの場合は常に有効となります。

#### ● 領域指定

[範囲指定]を設定します。

領域を削除したい場合は[削除]ボタンをクリックしてください。

ロードモジュールファイルを読み込み後に自動的に領域を設定したい場合は

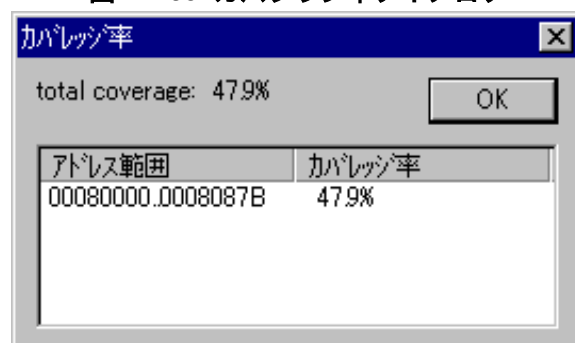
[自動設定]をクリックしてください。

### ■ カバレッジ率

カバレッジウィンドウのショートカットメニューの[全カバレッジ率]を選択してください。

トータルカバレッジ率とアドレス範囲ごとのカバレッジ率が表示されます。

図 4.4-30 カバレッジ率ダイアログ



## 4.4.13 リアルタイムメモリ

---

リアルタイムメモリウィンドウを表示します。

---

### ■ リアルタイムメモリウィンドウ

ミラーメモリを表示するウィンドウを開きます。指定されたアドレスからのメモリ内容のダンプ表示を行います。この機能は MB2198 エミュレータでのみ使用可能です。

ユーザプログラムによりアクセスされた位置が色分けされます。アクセスがされていない値は実メモリと異なっている場合があります ( ユーザプログラムによるアクセスのみ有効であり、デバッグの操作によるメモリアクセスでは値が有効になりません )。

- メモリ内容が無効

表示されているミラーメモリの内容は不定値です ( 灰色表示箇所 )。

- メモリ内容が有効

表示されているミラーメモリの内容は実メモリの内容と同じです ( 黒色または赤色表示箇所 )。

実行時にアクセスされた箇所のみメモリ内容は有効で、文字を黒色で表示します。また、メモリ内容が変更になった箇所は赤色表示となります。

- メモリウィンドウが開かれている場合

そのウィンドウをアクティブにします。

- メモリウィンドウが開かれていない場合

デバッグ環境設定ダイアログの [リアルタイム領域] タブで指定した領域が表示されます。

### ■ 領域

リアルタイムメモリウィンドウの表示領域を移動する場合は、リアルタイムメモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから [領域] を指定し、[領域 1] または [領域 2] を選択してください。デバッグ環境設定ダイアログの [リアルタイム領域] タブで指定した領域を表示します。

## ■ 表示設定

リアルタイムメモリウィンドウの表示形式を設定する場合は、リアルタイムメモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「表示設定」を選択してください。表示設定のダイアログが表示されます（図 4.4-31）。

### ● 表示形式

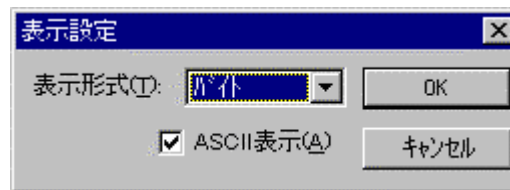
ウィンドウの表示形式を指定します。

ビット、バイト、ハーフワード、ワード、ダブルワードの中から選択できます。

### ● ASCII 表示

ウィンドウ右側の ASCII 文字が表示されるかどうかを選択します。

図 4.4-31 表示設定ダイアログ



## ■ 領域設定

リアルタイムメモリウィンドウの表示領域を設定する場合は、リアルタイムメモリウィンドウでマウスの右ボタンをクリックしてメニューを表示し、そこから「領域設定」を選択してください。デバッグ環境設定ダイアログの「リアルタイム領域」タブが表示されます。

---

### < 注意事項 >

この機能はエミュレータまたはその接続形態により使用できない場合があります。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.9 リアルタイムモニタ」を参照してください。

---

## 4.4.14 パフォーマンス

---

パフォーマンスウィンドウを表示します。

---

### ■ パフォーマンスウィンドウ

パフォーマンス測定結果を表示するウィンドウを開きます。パフォーマンスの制御と設定が行えます。

### ■ パフォーマンスモード設定

パフォーマンスウィンドウのショートカットメニューの「設定」を選択してください。

図 4.4-32 パフォーマンスモード設定ダイアログ



- バッファフルブレイク

パフォーマンスバッファフルブレイクをするか否かを指定します。

### ■ パフォーマンス区間指定

パフォーマンスウィンドウのショートカットメニューの「設定」を選択してください。

図 4.4-33 パフォーマンス区間指定ダイアログ



START はパフォーマンス測定開始条件, END は測定終了条件です。

START と END の両方とも入力してください。

- 属性

開始 / 終了条件の属性を指定します。コードまたはデータを選択できます。ただし FR80S ではデータのみとなります。

- アドレス

開始 / 終了条件を設定するアドレスまたはシンボルを指定します。

- 詳細

開始 / 終了条件の詳細設定を設定することができます。ボタンをクリックするとイベント詳細設定ダイアログが表示されます。

図 4.4-34 パフォーマンスイベント詳細設定ダイアログ



- アドレス

開始 / 終了条件を設定するアドレスまたはシンボルを指定します。

以下は、イベント設定 (パフォーマンス区間設定) ダイアログの属性で DATA を選択した場合にのみ指定できます。

- アドレスマスク

指定したアドレスに対するマスク値を指定します。

- サイズ (バイト / ハーフワード / ワード)

データアクセス時のデータサイズを指定します。

- 属性

データアクセス時の属性を指定します。ただし FR80S では指定できず、必ずライト属性となります。

- サイズ無視

アクセスデータ長にかかわらず、指定アドレスがアクセスされた場合を開始 / 終了条件とします。

- 比較条件

データ比較条件を指定します。

- 無効 : 遷移条件にデータを指定しません。
- データ一致 : 指定データと一致した場合を開始 / 終了条件とします。
- データ NOT : 指定データと一致しない場合を開始 / 終了条件とします。

- データ

データアクセス時のデータを指定します。

- データマスク

指定したデータに対するマスク値を指定します。



## ■ パフォーマンス表示指定

パフォーマンスウィンドウのショートカットメニューの[表示範囲]を選択してください。

図 4.4-35 パフォーマンス表示設定ダイアログ



- 表示モード  
自動またはマニュアルを選択します。マニュアルを選択した場合, [下限値] と [間隔] を設定してください。
- 下限値  
測定結果詳細表示の表示開始時間を指定してください。
- 間隔  
測定結果詳細表示の表示間隔を指定してください。
- 単位  
測定単位は 1ns 固定です。

---

### < 注意事項 >

1. FR60Lite または FR80S 使用時以外では, この機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』2.3.8 パフォーマンス測定」を参照してください。
  2. FR80S では内蔵 RAM 領域に設定した場合のみ有効となります。
  3. イベントモードがトレースモードの場合は使用できません。
-

## ■ パフォーマンス測定結果のコピー

パフォーマンスウィンドウに表示されている測定結果をクリップボードに出力することが可能です。パフォーマンスウィンドウに測定結果が表示されている状態で、ショートカットメニューの[コピー]を選択してください。

### 【例】

MB2198 にて表示タイプが「測定時間」の場合

event = 1 -> 2

min time = 7.050

max time = 26.225

avr time = 13.763

time (μs)			count
-----+-----			
0.000 -	7.999		1
8.000 -	8.999		0
9.000 -	9.999		0
10.000 -	10.999		2
11.000 -	11.999		0
12.000 -	12.999		1
13.000 -	13.999		1
14.000 -	14.999		2
15.000 -	15.999		0
16.000 -	16.999		0
17.000 -	17.999		0
18.000 -			1
-----+-----			
total			8

MB2198 にて表示タイプが「測定時間」の場合

no. count

1 8

2 8

3 15

4 26

### < 注意事項 >

ショートカットメニューの[コピー]によりクリップボードに出力される内容は、SHOW PERFORMANCE コマンドにより出力される内容と同じです。詳しくは『SOFTUNE Workbench コマンドリファレンスマニュアル』の「4.2 SHOW PERFORMANCE」を参照してください。

## 4.4.15 RAM チェッカ

---

RAM チェッカウィンドウを表示します。

---

### ■ RAM チェッカ

RAM チェッカウィンドウを開いて、そこに、ロギング状態の表示と監視アドレスのモニタリング表示を行います。また、ショートカットメニューより監視アドレスの設定やロギングの ON/OFF が可能です。

## 4.5 プロジェクト

---

プロジェクト関連の設定 , メイク / ビルドを行います。

---

### ■ プロジェクトの設定関連

- アクティブプロジェクトの設定
- プロジェクトの追加
- メンバの追加
- ワークスペースの設定
- プロジェクトの設定
- カスタマイズビルドの設定
- プロジェクトの依存関係
- プロジェクト構成
- 依存関係の更新

### ■ メイク / ビルド関連

- コンパイル
- メイク
- ビルド
- 中止

---

#### < 注意事項 >

デバッガセッション時は , 一部メニューが選択できません。設定を変更したい場合はデバッガを終了してください。

---

## 4.5.1 アクティブプロジェクトの設定

---

アクティブの切り替えを行います。

---

### ■ アクティブプロジェクトの設定

サブメニューにワークスペース内のプロジェクト名が表示されますので、アクティブプロジェクトにしたいプロジェクト名をクリックしてください。

アクティブプロジェクトは、メニューのプロジェクト名の左側にチェックマークが付きます。

## 4.5.2 プロジェクトの追加

---

ワークスペースにプロジェクトを追加します。

---

### ■ プロジェクトの追加

プロジェクトの追加には、次の2種類があります。

- 新規作成

プロジェクトを新規作成してワークスペースに登録します。

- 既存のプロジェクト

既に存在するプロジェクトをワークスペースに登録します。

---

#### < 注意事項 >

登録するプロジェクトと同じ名前のプロジェクトが、ワークスペース内に存在する場合はワークスペースに登録することができません。

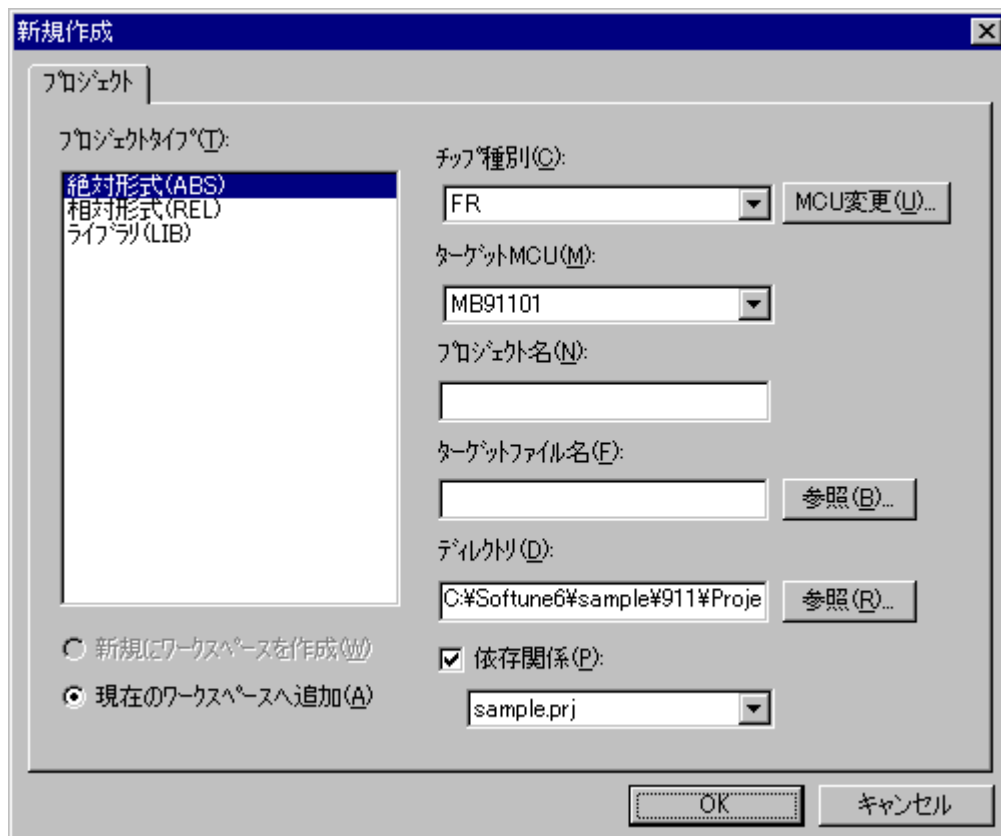
---

### 4.5.2.1 プロジェクトの追加 - 新規作成

ワークスペースにプロジェクトを新規作成して登録します。

#### ■ プロジェクトの追加 - 新規作成

図 4.5-1 プロジェクトの新規作成ダイアログ



- プロジェクトタイプ
  - プロジェクトタイプを選択します。プロジェクトタイプの種類については、「2.4 プロジェクトの登録」を参照してください。
- 新規にワークスペースを作成
  - ワークスペースを新規に作成してプロジェクトを登録します。ここでは、選択できません。
- 現在のワークスペースへ追加
  - 現在開いているワークスペースにプロジェクトを登録します。
- チップ種別
  - チップ種別を選択します。
- ターゲット MCU
  - ターゲット MCU を選択します。
- プロジェクト名
  - プロジェクト名を指定します。

## 第4章 メニュー

- ターゲットファイル名
  - ターゲットファイル名を指定します。
- ディレクトリ
  - プロジェクトのディレクトリを指定します。
- 依存関係
  - コンボボックスでプロジェクト名を指定すると、登録するプロジェクトを指定したプロジェクトのサブプロジェクトとして定義します。
- MCU 変更
  - MCU 変更ダイアログを表示します。

### ■ 追加手順

プロジェクトの追加の手順については、「2.4 プロジェクトの登録」を参照してください。

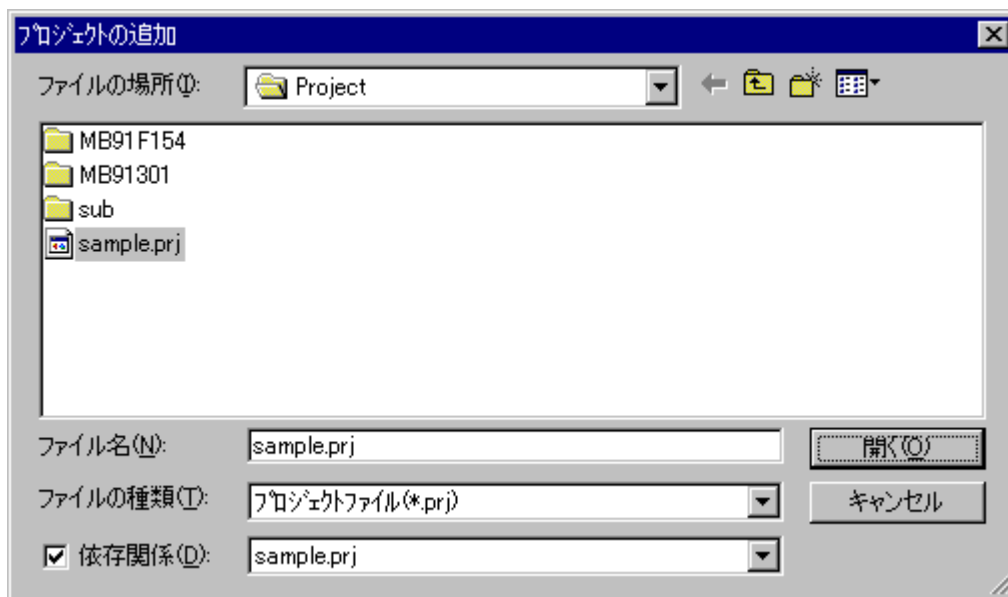


## 4.5.2.2 プロジェクトの追加 - 既存のプロジェクト

ワークスペースに既存のプロジェクトを登録します。

### ■ プロジェクトの追加 - 既存のプロジェクト

図 4.5-2 プロジェクトの追加ダイアログ



- 依存関係
  - コンボボックスでプロジェクト名を指定すると、登録するプロジェクトを指定したプロジェクトのサブプロジェクトとして定義します。

### ■ 追加手順

プロジェクトの追加の手順については、「2.4 プロジェクトの登録」を参照してください。

### 4.5.3 メンバの追加

---

プロジェクトにファイルを追加します。

---

#### ■ メンバの追加

メンバの追加には、次の 2 種類があります

- ファイル

ファイルを指定して、プロジェクトに登録します。

- ディレクトリ

ディレクトリを指定して、ディレクトリ内のファイル、フォルダをプロジェクトに登録します。

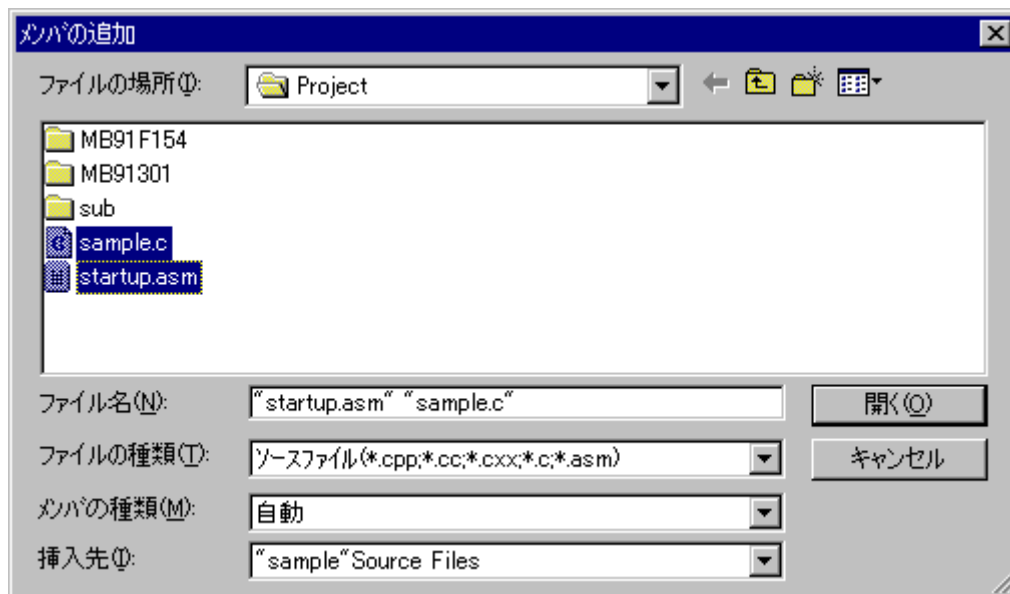
### 4.5.3.1 メンバの追加 - ファイル

ファイルを指定して、プロジェクトに登録します。

#### ■ メンバの追加 - ファイル

ファイル選択用のファイルダイアログボックスが表示されます。ここでは複数のファイルを一度に選択することができます。

図 4.5-3 メンバの追加ダイアログ



- メンバの種類
  - 登録するファイルを指定したメンバーの種類として、プロジェクトに登録します。  
"自動"を選択した場合は、拡張子からメンバーの種類を判断します。
- 挿入先
  - ファイルを登録するプロジェクトと挿入先のフォルダを指定します。" "で囲まれている文字列がプロジェクト名です。

#### ■ 追加手順

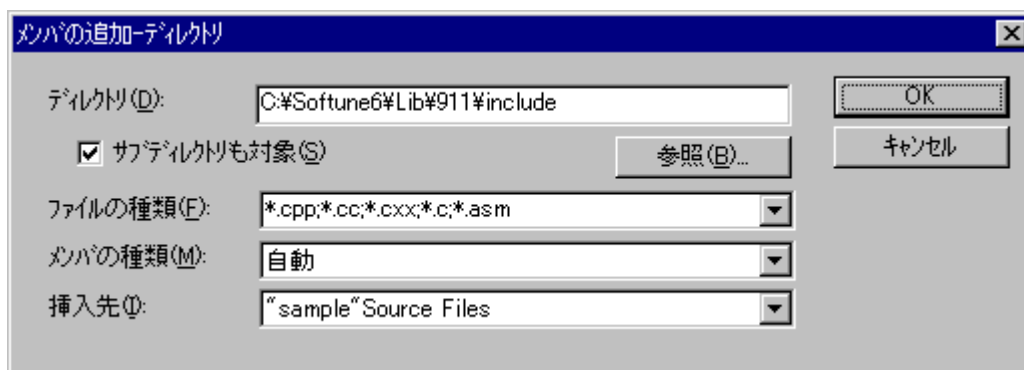
メンバーの追加の手順については、「2.5 ファイル作成とプロジェクトへの登録」を参照してください。

### 4.5.3.2 メンバの追加 - ディレクトリ

ディレクトリを指定して、ディレクトリ内のファイル、フォルダをプロジェクトに登録します。

#### ■ メンバの追加 - ディレクトリ

図 4.5-4 メンバの追加 - ディレクトリダイアログ



- ディレクトリ
  - 登録したいファイルが格納されているディレクトリを指定します。
- サブディレクトリも対象
  - オンにすると、指定したディレクトリのサブディレクトリ内のファイルを追加の対象にします。サブディレクトリはプロジェクトにフォルダとして階層的に作成されます。
- ファイルの種類
  - 指定した拡張子のファイルのみをプロジェクトに登録するようにします。
- メンバの種類
  - 登録するファイルを指定したメンバーの種類として、プロジェクトに登録します。  
" 自動 " を選択した場合は、拡張子からメンバーの種類を判断します。
- 挿入先
  - ファイルを登録するプロジェクトと挿入先のフォルダを指定します。"" で囲まれている文字列がプロジェクト名です。

#### ■ 追加手順

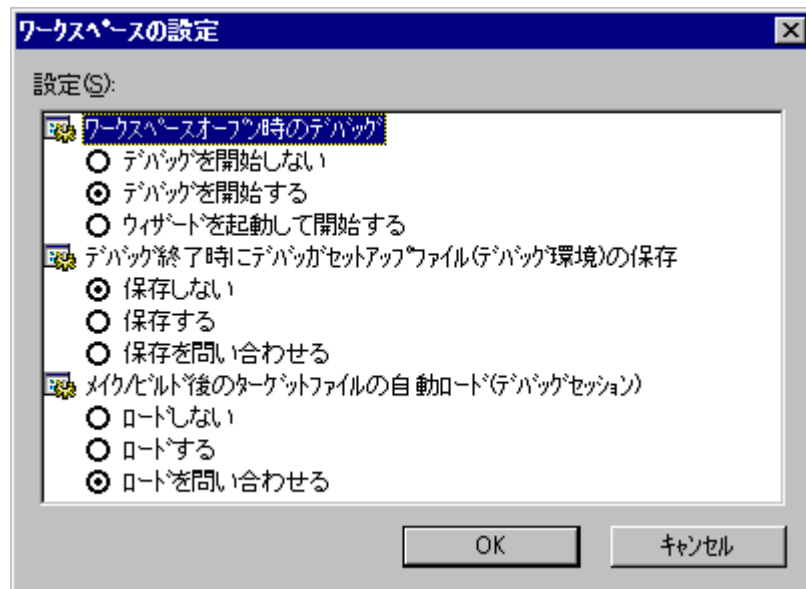
メンバーの追加の手順については、「2.5 ファイル作成とプロジェクトへの登録」を参照してください。

## 4.5.4 ワークスペースの設定

ワークスペースに関する基本的な設定を行います。

### ■ ワークスペースの設定

図 4.5-5 ワークスペースの設定ダイアログ



- ワークスペースオープン時のデバッグ
  - ワークスペースを開いたときのデバッグ動作を指定します（デバッグを開始しない / デバッグを開始する / ウィザードを起動して開始する）。
- デバッグ終了時にデバッグセットアップファイル（デバッグ環境）の保存
  - デバッグセッション終了時のセットアップ情報の保存を指定します（保存しない / 保存する / 保存を問い合わせる）。
- メイク / ビルド後にターゲットファイルを自動ロードする（デバッグセッション）
  - デバッグセッションでメイク / ビルドを行った後にターゲットファイルを再ロードするかどうかを指定します。（ロードしない / ロードする / ロードを問い合わせる）

## 4.5.5 プロジェクトの設定

プロジェクトに関する設定を行います。

### ■ プロジェクトの設定

プロジェクトの設定ダイアログは、設定を行う対象を指定する部分と設定を行う部分に分かれています。

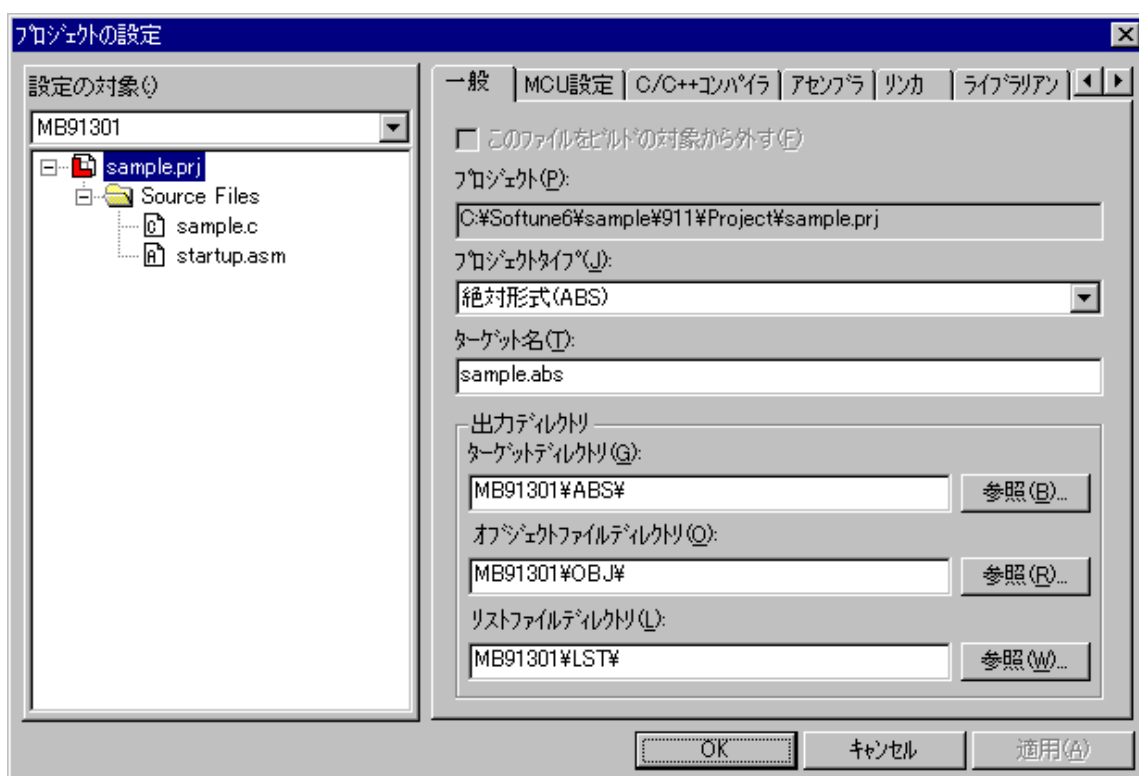
ここで設定した値は、[適用] ボタンまたは [OK] ボタンをクリックすると有効になります。

ディレクトリは、プロジェクトからの相対パスで記述することが可能です。

設定を行う対象が複数の場合、値の表示は次のようになります。

- エディット
  - すべての対象で等しい場合は、そのまま表示されます。
  - 一つでも等しくない対象がある場合は、空白になります。
- チェックボタン
  - すべての対象で等しい場合は、そのまま表示されます。
  - 一つでも等しくない対象がある場合は、淡色表示になります。

図 4.5-6 プロジェクトの設定

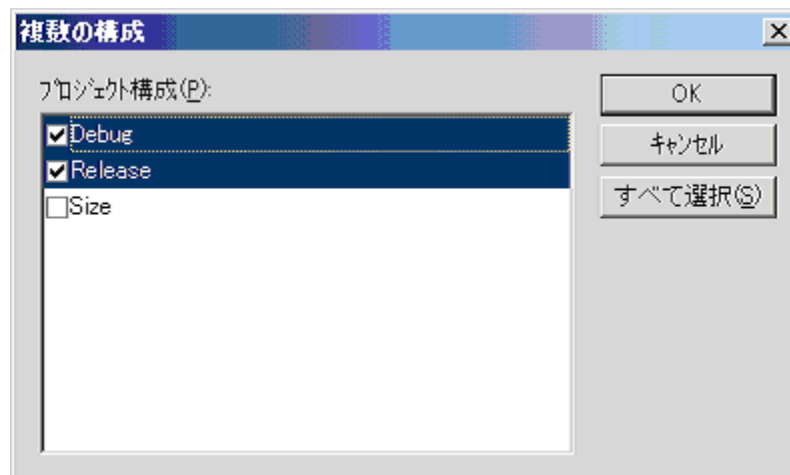


## ■ 設定を行う対象

ダイアログの左側のコンボボックスとツリービューで設定を行う対象を指定します。

- 設定の対象 - コンボボックス：ワークスペース内のプロジェクトの構成名がすべて表示されます。
  - 構成名：選択した構成に対して設定を行います。
  - 「すべての構成」：すべての構成に対して設定を行います。
  - 「複数の構成」：「複数の構成」ダイアログ (図 4.5-7) が表示され、ダイアログで指定した複数の構成に対して設定を行います。
- 設定の対象 - ツリービュー：コンボボックスで指定した構成名を持つすべてのプロジェクトが表示されます。選択するアイテムによって設定できる項目が異なります。複数選択することにより、一度に変更することが可能です。異なる種類のアイテムを選択した場合は、重複する項目のみ設定できます。
  - プロジェクト : [一般], [MCU 設定], 共通オプション ([C/C++ コンパイラ], [アセンブラ], [リンカ], [ライブラリアン]), [デバッグ] の設定を行うことができます。
  - C/C++ ソースファイル : [一般], 個別オプション ([C/C++ コンパイラ]) の設定を行うことができます。
  - アセンブラソースファイル : [一般], 個別オプション ([アセンブラ]) の設定を行うことができます。
  - ライブラリファイル : [一般] の設定を行うことができます。
  - オブジェクトファイル : [一般] の設定を行うことができます。
  - 相対形式ファイル : [一般] の設定を行うことができます。
  - フォルダ : フォルダ内にあるすべてのファイルの設定を行います。

図 4.5-7 複数の構成ダイアログ



## ■ 設定する項目

タブ形式で、以下の項目の設定 / 変更ができます。

- 一般
  - プロジェクトタイプ, ターゲット名, 出力ディレクトリ, ファイルのビルド設定などの設定を行います。
- MCU 設定
  - チップ種別, ターゲット MCU など MCU に関する設定を行います。
- C/C++ コンパイラ
  - C/C++ コンパイラオプションの設定を行います。
- アセンブラ
  - アセンブラオプションの設定を行います。
- リンカ
  - リンカオプションの設定を行います。
- ライブラリアン
  - ライブラリアンオプションの設定を行います。
- コンバータ
  - コンバータの起動やコンバータオプションの設定を行います。
- デバッグ
  - デバッグオプションやデバッガのセットアップ情報の設定を行います。



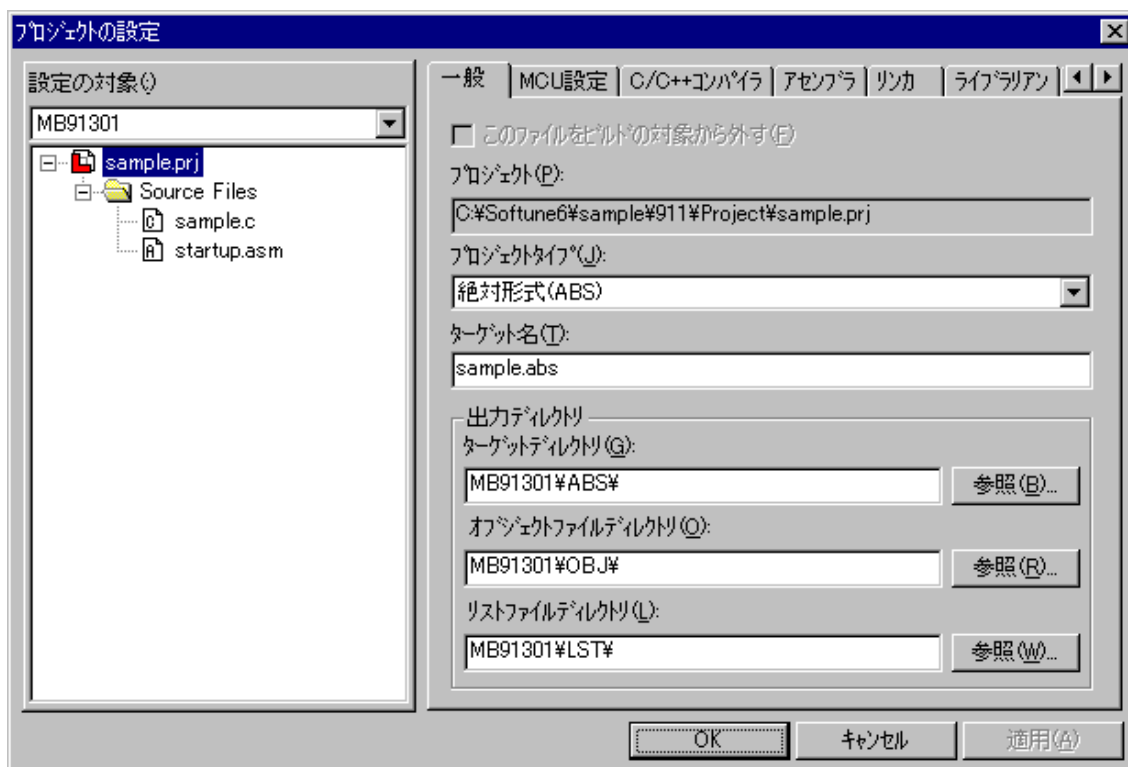
### 4.5.5.1 一般

ここでは、プロジェクトの設定の一般で設定できる項目について説明します。

#### ■ 一般の設定方法

- 1) プロジェクトの設定ダイアログで、[ 一般 ] タブをクリックしてください。
- 2) 必要に応じて以下の設定を行ってください。
  - [ このファイルをビルドの対象から外す ], [ プロジェクトタイプ ], [ ターゲット名 ], [ 出力ディレクトリ ]

図 4.5-8 プロジェクトの設定 (一般)



- このファイルをビルドの対象から外す
  - ツリービューで選択しているファイルをメイク / ビルドの対象から外します。この項目は、C/C++ ソースファイル、アセンブラソースファイル、ライブラリファイル、オブジェクトファイル、相対形式ファイルを選択している場合に設定できます。
- プロジェクト
  - プロジェクトファイルのフルパスを表示します。
- プロジェクトタイプ
  - ツリービューで選択しているプロジェクトファイルの種類を設定します (絶対形式 (ABS) / 相対形式 (REL) / ライブラリ (LIB))。この項目は、プロジェクトファイルを単体で選択している場合に設定できます。
  - プロジェクトタイプを構成ごとに設定することはできません。

- ターゲット名
  - ターゲットファイルの主ファイル名を設定します。この項目は、プロジェクトのみ（複数可）を選択している場合に設定できます。
- ターゲットディレクトリ
  - ターゲットファイルのディレクトリを設定します。この項目は、プロジェクトのみ（複数可）を選択している場合に設定できます。
- オブジェクトファイルディレクトリ
  - コンパイル、アセンブルで出力するオブジェクトファイルの出力先ディレクトリを設定します。この項目は、プロジェクトのみ（複数可）を選択している場合に設定できます。
- リストファイルディレクトリ
  - コンパイル、アセンブル、ビルドで出力するリストファイルの出力先ディレクトリを設定します。この項目は、プロジェクトのみ（複数可）を選択している場合に設定できます。

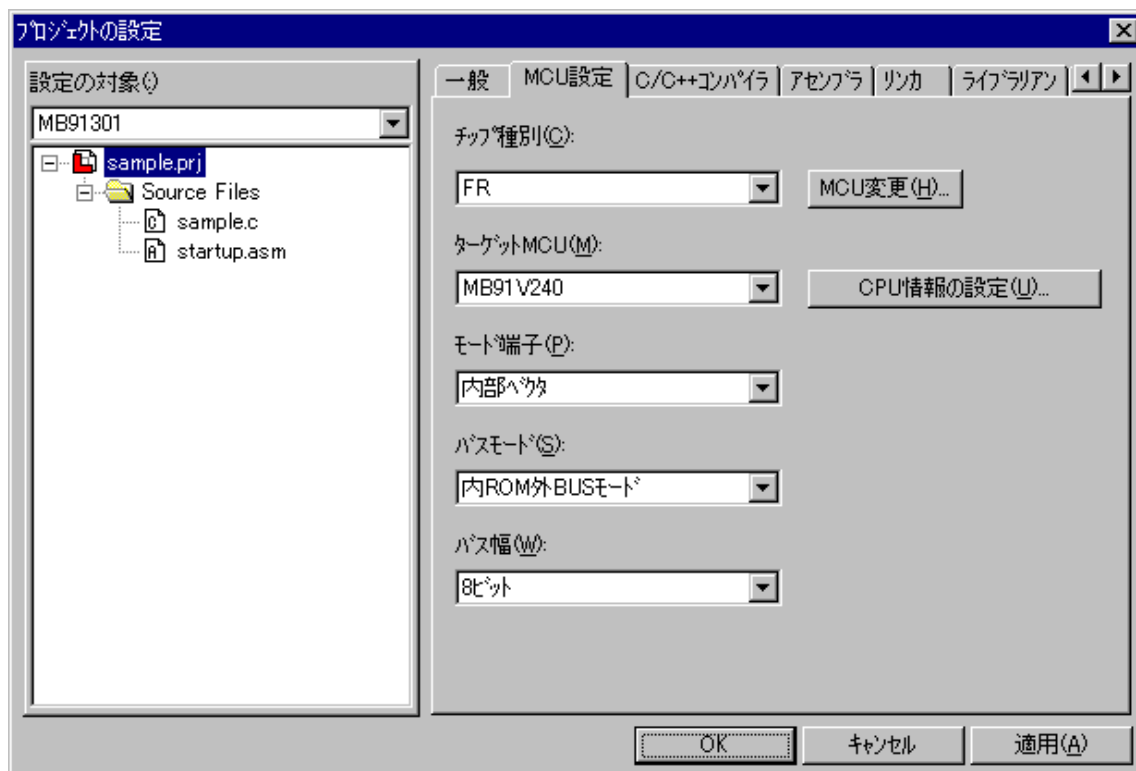
## 4.5.5.2 MCU 設定

ここでは、プロジェクトの設定の MCU 設定で設定できる項目について説明します。

### ■ MCU 設定の設定方法

- 1) プロジェクトの設定ダイアログで、[MCU 設定] タブをクリックしてください。
- 2) 必要に応じて以下の設定を行ってください。
  - [チップ種別]、[ターゲット MCU]、[モード端子]、[バスモード]、[バス幅]、[MCU 変更]、[CPU 情報の設定]

図 4.5-9 MCU 設定ダイアログ



- チップ種別
  - チップの種別を選択します。
- ターゲット MCU
  - [チップ種別] で選択したチップ種別のターゲット MCU を選択します。
- CPU 情報の設定
  - CPU 情報の設定ダイアログを表示します。
- モード端子
  - ある品種について、リセットコマンドを実行するときにエミュレータがモード端子を参照できない場合があります。この場合、本機能を用いて、参照するペクタ領域を選択する必要があります。

### 自動

リセットコマンド実行時に、自動的にリセットベクタを参照します。

### 内部ベクタ

リセットコマンド実行時に、内部ベクタを参照します。

### 外部ベクタ

リセットコマンド実行時に、外部ベクタを参照します。

- バスモード

- 品種によっては、アクセスが禁止されているメモリ領域がある場合があります。本エミュレータデバッガでは、メモリを操作するコマンドを実行するときに、指定したアドレスがアクセス禁止領域内に存在しないか判定する機能があります。このアクセス禁止領域はバスモードによって異なりますので、このリストから選択してください。

未対応の品種では、このリストは表示されません。

### 無効

アクセス禁止領域との比較を行いません。

### シングルチップモード

シングルチップモード時のアクセス禁止領域を参照します。

### 内部 ROM 外 BUS モード

内部 ROM 外バス時のアクセス禁止領域を参照します。

### 外部 ROM 外 BUS モード

外部 ROM 外バス時のアクセス禁止領域を参照します。

- バス幅

ターゲット MCU でバスモード設定可能なチップを選択しバスモードで外バスを選択すると、バス幅の設定が可能です。

- MCU 変更

MCU 変更ダイアログを表示します。

## ■ CPU 情報の設定の設定方法

### ● CPU 情報の設定 - DSU

図 4.5-10 CPU 情報の設定 (DSU)



- DSU 選択
  - DSU タイプを選択します (DSU1/DSU2/DSU3/DSU4)。

### ● CPU 情報の設定 - IRQ と ICR

図 4.5-11 CPU 情報の設定 (IRQ と ICR)



- Vector と ICR の関係について
  - 外部割り込み (Vector) と割り込み制御レジスタ (ICR) の対応はターゲット MCU により異なります。
  - シミュレータデバッガにおいて外部割り込みと割り込み制御レジスタの対応を指定します。
- 外部割り込みと ICR の対応設定
  - ベクタ番号に対応する割り込み制御レジスタ (ICR) を設定します。ベクタ番号を選択して [ 変更 ] ボタンを押下すると、以下のダイアログ ( 図 4.5-12 ) が表示されます。

図 4.5-12 外部割り込みと ICR の対応



- CPU 情報の設定 - 遅延割り込み

図 4.5-13 CPU 情報の設定 (遅延割り込み)



- ベクタ番号
  - 遅延割り込み発生時のベクタ番号を設定します。

---

< 注意事項 >

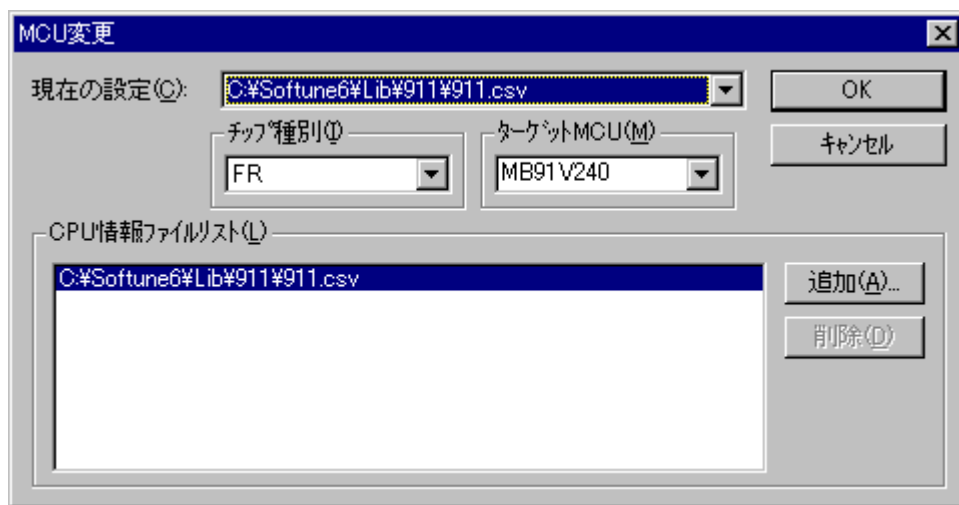
品種によっては、使用できない場合があります。

---

## ■ MCU 変更ダイアログの設定方法

- 1) [MCU 変更] ボタンをクリックしてください。
  - MCU 変更ダイアログ (図 4.5-14) が開きます。
- 2) [現在の設定] で使用する CPU 情報ファイルを選択してください。使用する CPU 情報ファイルが存在しない場合は, [CPU 情報ファイルリスト] の [追加] ボタンからファイルを追加します。
  - 選択した CPU 情報ファイルの MCU の一覧が [チップ種別], [ターゲット MCU] に表示されます。
- 3) 変更したい MCU を [チップ種別], [ターゲット MCU] から選択して, [OK] ボタンをクリックしてください。
  - 変更した内容が反映されます。

図 4.5-14 MCU 変更ダイアログ



- 現在の設定  
使用する CPU 情報ファイルを選択します。選択した CPU 情報ファイルの MCU の一覧が [チップ種別], [ターゲット MCU] に表示されます。
- チップ種別  
チップの種別を選択します。
- ターゲット MCU  
[チップ種別] で選択したチップ種別のターゲット MCU を選択します。
- CPU 情報ファイルリスト  
登録されている CPU 情報ファイル一覧を表示します。
- 追加  
CPU 情報ファイルを追加します。
- 削除  
[CPU 情報ファイルリスト] で選択されている CPU 情報ファイルを削除します。ただし, デフォルトで検索されるファイルは削除できません。

### 4.5.5.3 C/C++ コンパイラオプションの設定

---

ここでは、C/C++ コンパイラのオプションの設定方法について説明します。

---

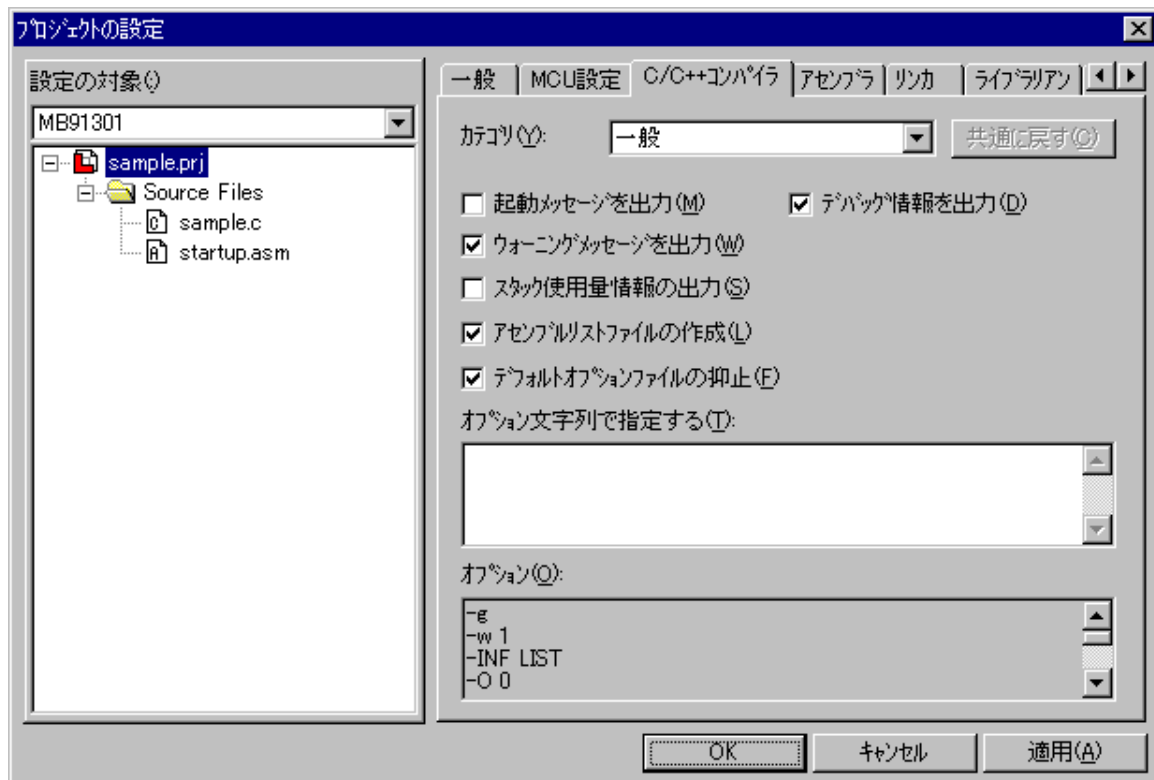
#### ■ C/C++ コンパイラオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[C/C++ コンパイラ] をクリックしてください。
- 2) [カテゴリ] を選択してください。
  - [一般]、[マクロ名の設定]、[インクルードパス]、[最適化]、[C++]、[言語仕様] が選択できます。
  - どのカテゴリを選択しても、ダイアログの下部には、現在設定されているすべての C/C++ コンパイラオプションを表示します。
  - 個別オプションを設定している場合は、[共通に戻す] ボタンをクリックすることによって、共通オプションに戻すことができます。
  - オプションの記述では、便利なマクロ記述を使用することができます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。



## ■ [一般] に属するオプションの設定方法

図 4.5-15 一般オプションの設定



ここでは、以下のオプションの設定ができます。これらは、チェックマークがついている場合に、それぞれの機能を実現します。

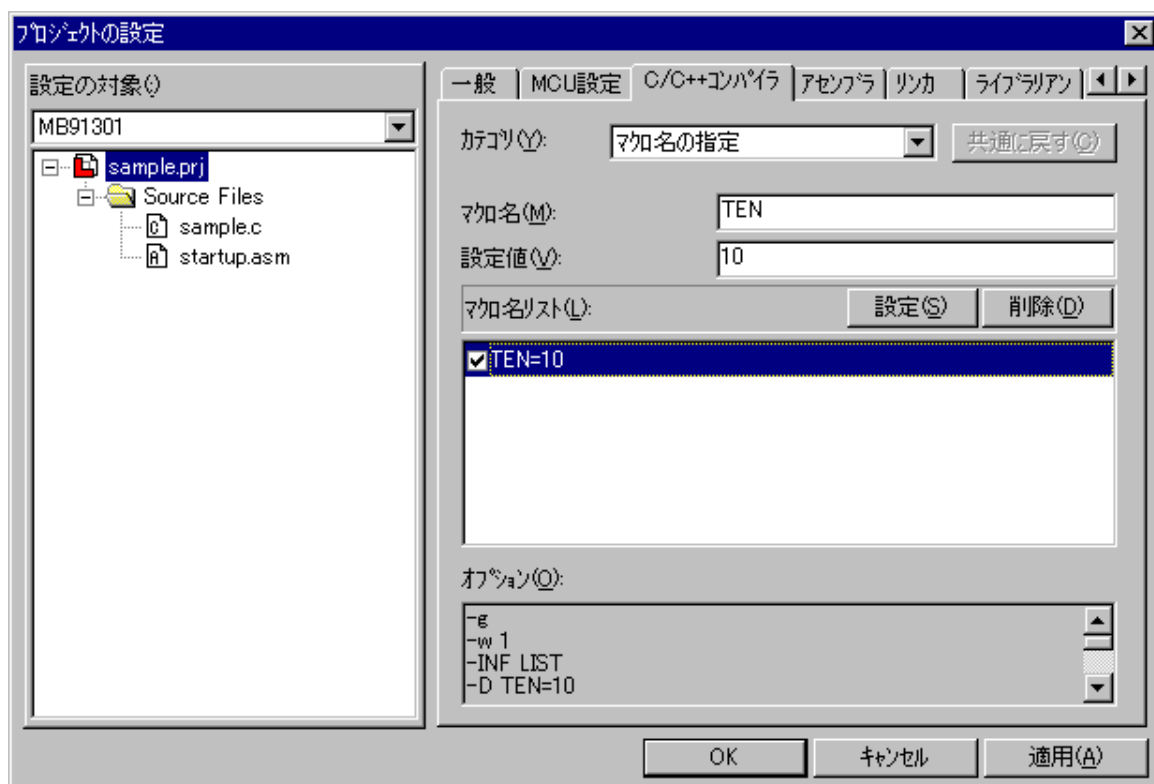
- 起動メッセージを出力 (-V)
- デバッグ情報を出力 (-g)
- ウォーニングメッセージを出力 (またはウォーニングレベル) (-w)
- スタック使用量情報の出力 (-INF STACK)
- アセンブルリストファイルの作成 (-INF LIST)
- デフォルトオプションの抑止 (-Xdof)

[ウォーニングメッセージを出力]は、CPU 種別によっては [ウォーニングレベル] (レベル 0 ~ 3) を選択する場合があります。

[オプション文字列で指定する]では、C/C++ コンパイラのすべてのオプションを、コマンドラインからの起動オプションと同様に記述できます。C/C++コンパイラオプション設定のどのカテゴリにも属さないオプションを指定する場合に、ここに直接記述してください。

## ■ マクロ名の設定方法

図 4.5-16 マクロ名の設定ダイアログ



設定を行う対象が複数の場合、一部のみに存在するマクロ名は灰色で表示されます。

- 1) [ マクロ名の指定 ] カテゴリを選択してください。
  - マクロ名の設定ダイアログ ( 図 4.5-16 ) が表示されます。
- 2) [ マクロ名 ] を指定してください。
- 3) 必要に応じて、[ 設定値 ] を指定してください。
- 4) [ 設定 ] ボタンをクリックしてください。
  - 指定したマクロ名は、デファイン (-D) オプションとして設定されます。
- 5) アンデファイン (-U) として設定したい場合は、[ マクロ名リスト ] 中のチェックマークを解除してください。

### < 注意事項 >

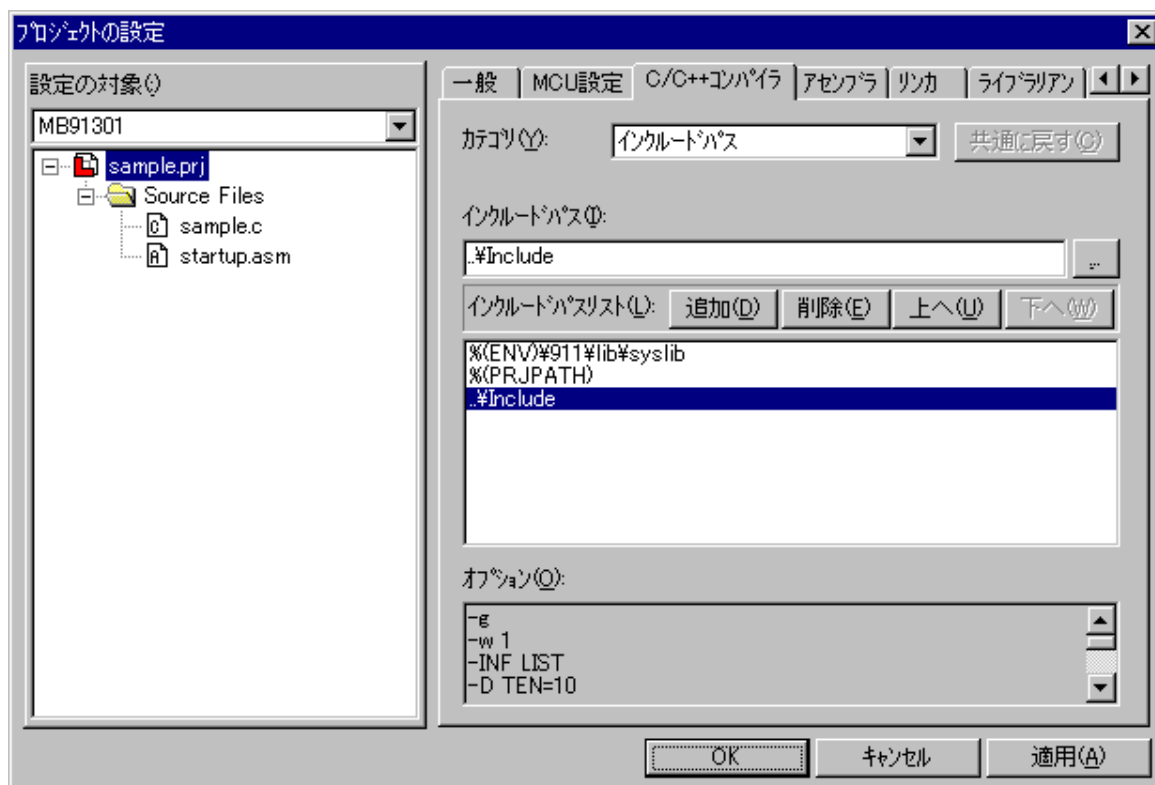
アンデファイン設定した場合は、同一のマクロ名に対して、デファインとアンデファインの両方のオプションが出力されますが、アンデファインが優先されるため動作上は問題ありません。

## ■ マクロ名の設定解除方法

- 1) [ マクロ名の指定 ] カテゴリを選択してください。
  - マクロ名の設定ダイアログ ( 図 4.5-16 ) が表示されます。
- 2) [ マクロ名リスト ] から解除したいマクロ名を選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。

## ■ インクルードパスの設定方法

図 4.5-17 インクルードパスの設定ダイアログ



- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-17 ) が表示されます。
- 2) [ インクルードパス ] を指定してください。
  - 入力フィールドの右横の参照ボタンをクリックすると、ディレクトリを選択することができます。
- 3) [ 追加 ] ボタンをクリックしてください。
  - [ インクルードパスリスト ] の最後に追加されます。

## ■ インクルードパスの設定解除方法

- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-17 ) が表示されます。
- 2) [ インクルードパスリスト ] から解除したいパスを選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。

## ■ インクルードパスの検索順位の変更方法

インクルードパスの検索は, [ インクルードパスリスト ] の上から順に行います。以下の手順で, リストへの登録順を変更できます。

- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-17 ) が表示されます。
- 2) [ インクルードパスリスト ] から順序を変更したいパスを選択してください。
- 3) [ 上へ ] ボタンまたは [ 下へ ] ボタンをクリックして, 適切な位置まで移動させてください。

## ■ [最適化] に属するオプションの設定方法

図 4.5-18 最適化オプション設定ダイアログ



ここでは、以下のオプションの設定ができます。これらは、ドロップダウンリストに表示される選択肢から選択してください。

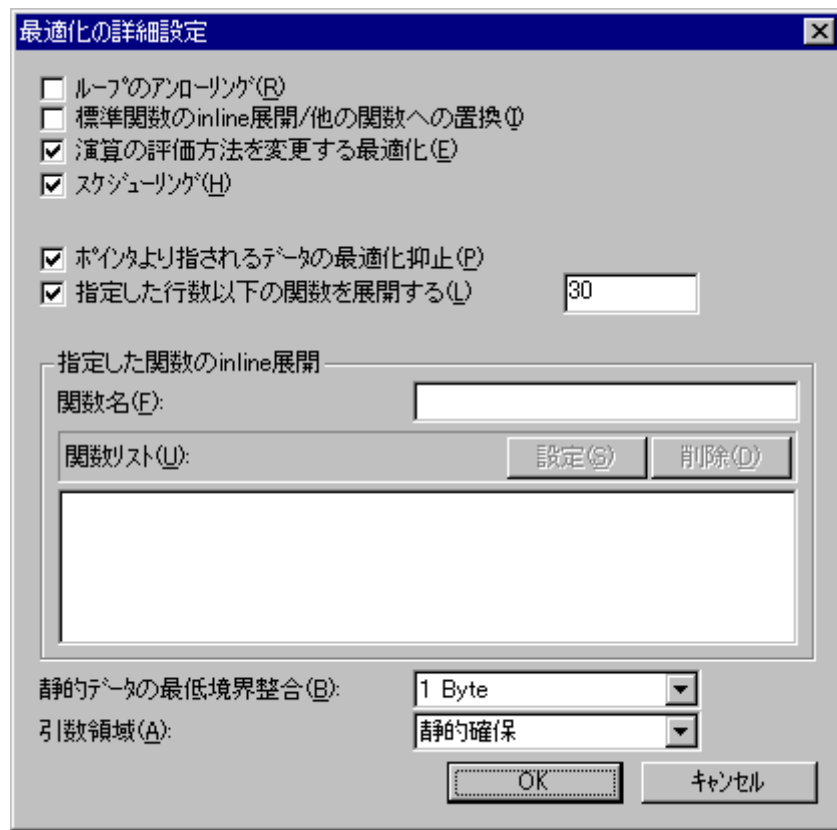
- 最適化レベル (-O) (なし / レベル 1 ~ 4/ 速度優先 / サイズ優先)
- コード領域の外部シンボルサイズ (20 ビット / 32 ビット)
- データ領域の外部シンボルサイズ (20 ビット / 32 ビット)

## &lt; 注意事項 &gt;

最適化レベルを " 速度優先 " または " サイズ優先 " を選択した場合、[ コード領域の外部シンボルサイズ ] および [ データ領域の外部シンボルサイズ ] は " 20 ビット " に自動的に変更されます。それぞれのシンボルサイズは、最適化レベルの変更をおこなってから変更してください。

## ■ [最適化] の「詳細設定」に属するオプションの設定方法

図 4.5-19 最適化の詳細設定ダイアログ



ここでは、以下のオプションの設定ができます。

- ループのアンローリング (-K UNROLL)
- 標準関数の inline 展開 / ほかの関数への置換 (-K LIB)
- 演算の評価方法を変更する最適化 (-K EOPT)
- スケジューリング (-K SCHEDULE)
- ポインタより指されるデータの最適化を抑制 (-K NOALIAS)
- 指定した行数以下の関数を展開する (-xauto)
- 指定した関数を inline 展開 (-x)
- 静的データの最低境界整合 (-K A4A1) (1Byte/4Byte)
- 引数領域 (-K SARGIDARG) (静的確保 / 動的確保)

## ■ 指定した関数の inline 展開の設定方法

- 1) [最適化] カテゴリを選択してください。
  - 最適化の設定ダイアログ (図 4.5-18) が表示されます。
- 2) [最適化レベル] を 0 以外にして [詳細設定] ボタンをクリックしてください。
- 3) [関数名] を指定してください。
- 4) [設定] ボタンをクリックしてください。

## ■ 指定した関数の inline 展開の設定解除方法

- 1) [最適化] カテゴリを選択してください。
  - 最適化の設定ダイアログ (図 4.5-18) が表示されます。
- 2) [最適化レベル] を 0 以外にして [詳細設定] ボタンをクリックしてください。
- 3) [関数リスト] から解除したい関数名を選択してください。
- 4) [削除] ボタンをクリックしてください。

---

### < 注意事項 >

最適化レベルを変更した場合、詳細設定ダイアログで設定されるオプションは、最適化レベルに応じて初期化されます。

---

## ■ [ 言語仕様 ] に属するオプションの設定方法

図 4.5-20 言語仕様オプション設定ダイアログ



ここでは、以下のオプションの設定ができます。

- 接尾語が指定されていない浮動小数点数を float 型として扱う  
(-K FCONSTIDCONST)
- 割込み関連関数のインライン展開 (-K NOINTLIB)
- \_\_io 型修飾子を volatile 型として扱う (-K NOVOLATILE)
- char 型の最上位ビットを符号ビットとして扱う (-K SCHAR)
- ITRON システムコールのインライン展開 (-K REALOS)
- int 型をビットフィールドに指定した場合最上位ビットを符号ビットとして扱う  
(-K SBIT)
- 静的変数のメモリ格納順をソース記述順にする (-verorder)
- C 言語ソースで "/\*" 以降をコメントとする (-B)
- 言語仕様レベル (-J alcle) (EC++/ANSI/ANSI + 拡張)



## ■ [C++] に属するオプション

図 4.5-21 C++ 設定ダイアログ



ここでは、以下のオプションの設定ができます。

- テンプレートを自動的に実体化する (`--no_auto_instantiation`)
- テンプレートの実体化の制御 (`-t nonelusedlllocalall`) (生成しない / 生成する / ローカルで出力 / すべて出力)
- 旧 for ループ初期化スコーピングを使用する (`--old_for_init`)
- 代替キーワードを使用する (`--alternative_tokens`)

#### 4.5.5.4 アセンブラオプションの設定

---

ここでは、アセンブラのオプションの設定方法について説明します。

---

##### ■ アセンブラオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[ アセンブラ ] タブをクリックしてください。
- 2) [ カテゴリ ] を選択してください。
  - [ 一般 ], [ マクロ名の設定 ], [ インクルードパス ], [ ターゲット依存 ], [ リスト出力 ] が選択できます。
  - 設定されたオプションは、ダイアログの下部にある [ オプション ] で確認することが可能です。
  - 個別オプションを設定している場合は、[ 共通に戻す ] ボタンをクリックすることによって、共通オプションに戻すことができます。
  - オプションの記述では、便利なマクロ記述を使用することができます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。

## ■ [一般] に属するオプションの設定方法

図 4.5-22 一般オプションの設定



ここでは、以下のオプションの設定ができます。

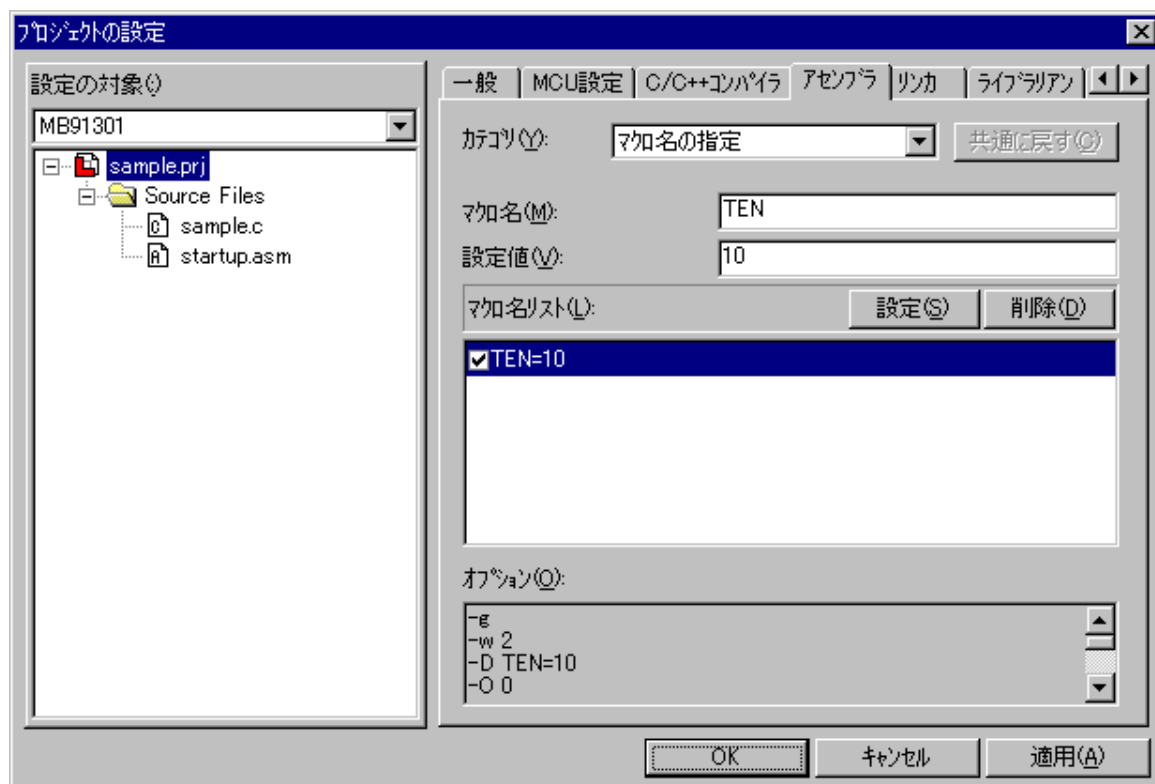
- 起動メッセージを出力 (-V)
- デバッグ情報を出力 (-g)
- デフォルトオプションの抑止 (-Xdof)
- ウォーニングレベル (-w) (レベル 0 ~ 3)

[ウォーニングレベル] は、ドロップダウンリストの中から選択してください。

[オプション文字列で指定する] では、アセンブラのすべてのオプションを、コマンドラインからの起動オプションと同様に記述できます。アセンブラオプション設定のどのカテゴリにも属さないオプションを指定する場合に、ここに直接記述してください。

## ■ マクロ名の設定方法

図 4.5-23 マクロ名の設定ダイアログ



設定を行う対象が複数の場合、一部のみに存在するマクロ名は灰色で表示されます。

- 1) [マクロ名の指定] カテゴリを選択してください。
  - マクロ名の設定ダイアログ (図 4.5-23) が表示されます。
- 2) [マクロ名] を指定してください。
- 3) 必要に応じて、[設定値] を指定してください。
- 4) [設定] ボタンをクリックしてください。
  - 指定したマクロ名は、デファイン (-D) オプションとして設定されます。
- 5) アンデファイン (-U) として設定したい場合は、[マクロ名リスト] 中のチェックマークを解除してください。

### < 注意事項 >

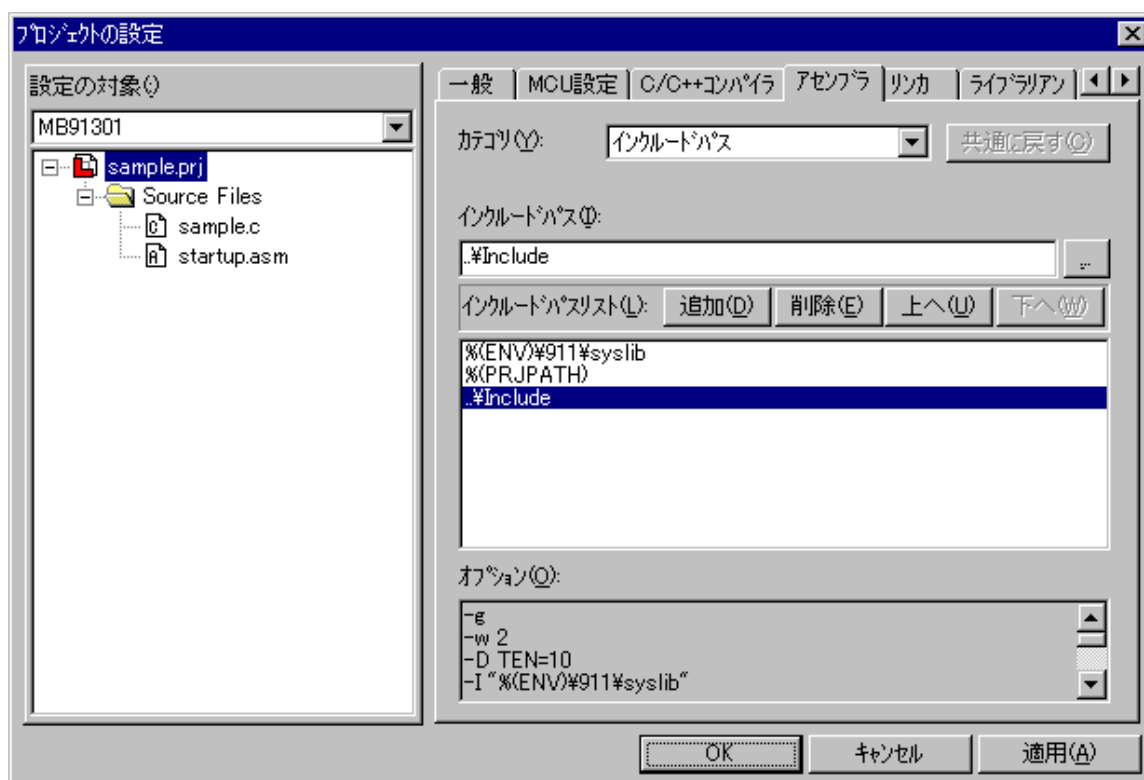
アンデファイン設定した場合は、同一のマクロ名に対して、デファインとアンデファインの両方のオプションが出力されますが、アンデファインが優先されるため動作上は問題ありません。

## ■ マクロ名の設定解除方法

- 1) [ マクロ名の指定 ] カテゴリを選択してください。
  - マクロ名の設定ダイアログ ( 図 4.5-23 ) が表示されます。
- 2) [ マクロ名リスト ] から解除したいマクロ名を選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。

## ■ インクルードパスの設定方法

図 4.5-24 インクルードパス設定ダイアログ



- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-24 ) が表示されます。
- 2) [ インクルードパス ] を指定してください。
  - 入力フィールドの右横の参照ボタンをクリックすると、ディレクトリを選択することができます。
- 3) [ 追加 ] ボタンをクリックしてください。
  - [ インクルードパスリスト ] の最後に追加されます。

## ■ インクルードパスの設定解除方法

- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-24 ) が表示されます。
- 2) [ インクルードパスリスト ] から解除したいパスを選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。

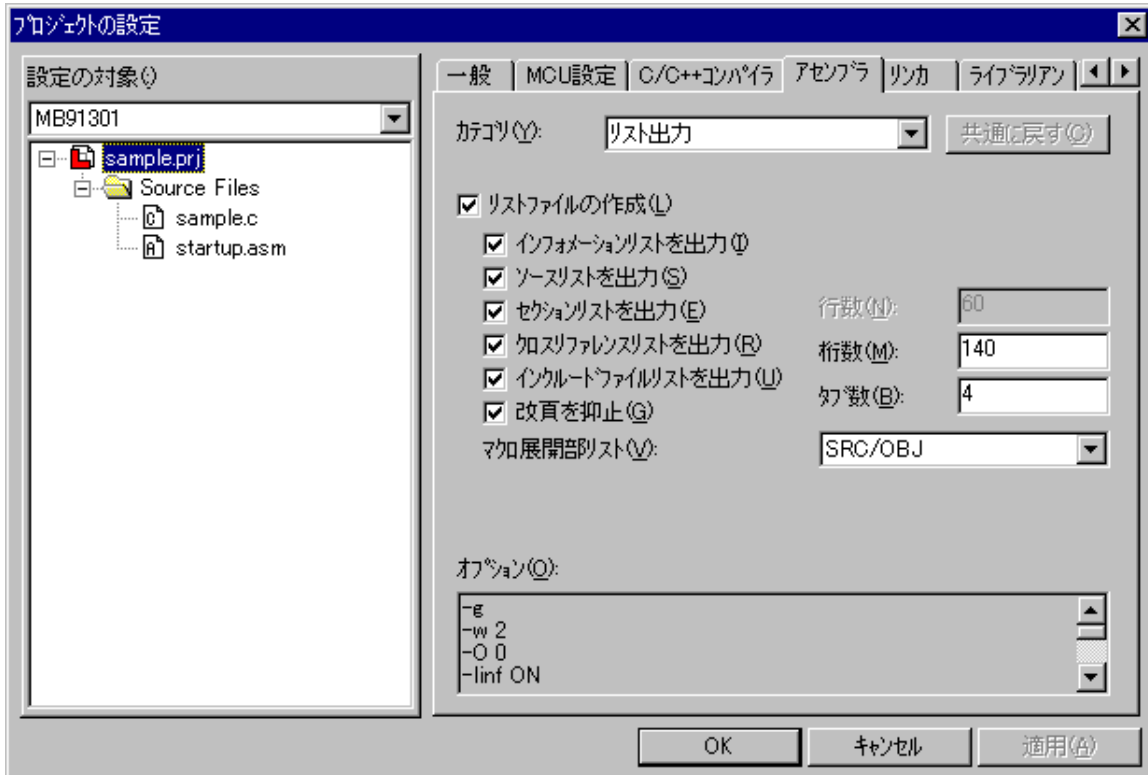
## ■ インクルードパスの検索順位の変更方法

インクルードパスの検索は , [ インクルードパスリスト ] の上から順に行います。以下の手順で , リストへの登録順を変更できます。

- 1) [ インクルードパス ] カテゴリを選択してください。
  - インクルードパスの設定ダイアログ ( 図 4.5-24 ) が表示されます。
- 2) [ インクルードパスリスト ] から順序を変更したいパスを選択してください。
- 3) [ 上へ ] ボタンまたは [ 下へ ] ボタンをクリックして , 適切な位置まで移動させてください。

## ■ リスト出力の設定方法

図 4.5-25 リスト出力の設定ダイアログ



- 1) [リスト出力]カテゴリを選択してください。
  - リスト出力の設定ダイアログ (図 4.5-25) が表示されます。
- 2) リストファイルを出力する場合は, [リストファイルの作成] にチェックマークを設定してください。
  - リストファイルを出力しないときは, このダイアログではほかに設定する項目はありません。リストファイルを出力する場合のみ, 3) 以降を実行してください。
- 3) どのようなリストファイルを出力するかを選択してください。チェックマークがついた項目が選択されます。
  - インフォメーションリストを出力
  - ソースリストを出力
  - セクションリストを出力
  - クロスリファレンスリストを出力
  - インクルードファイルリストを出力
- 4) 改ページの指定をしてください。[改頁を抑止] にチェックマークが付いている場合は, 改ページしません。
  - 改ページしない場合は, 行数の設定はできません。
- 5) [マクロ展開部リスト] を, ドロップダウンリストの項目から選択してください (出力しない, SRC/OBJ, OBJ)。
- 6) 必要に応じて, [行数], [桁数], [タブ数] を設定してください。

## ■ [ ターゲット依存 ] に関するオプション

図 4.5-26 ターゲット依存設定ダイアログ



ここでは、以下のオプションの設定ができます。

- 最適化レベル (-O) (なし / レベル 1 / レベル 2)
- FPU チャンネル番号 (-FPU) (FPU なし / チャンネル 0 ~ 15)



### 4.5.5.5 リンカオプションの設定

---

ここでは、リンカのオプションの設定方法について説明します。

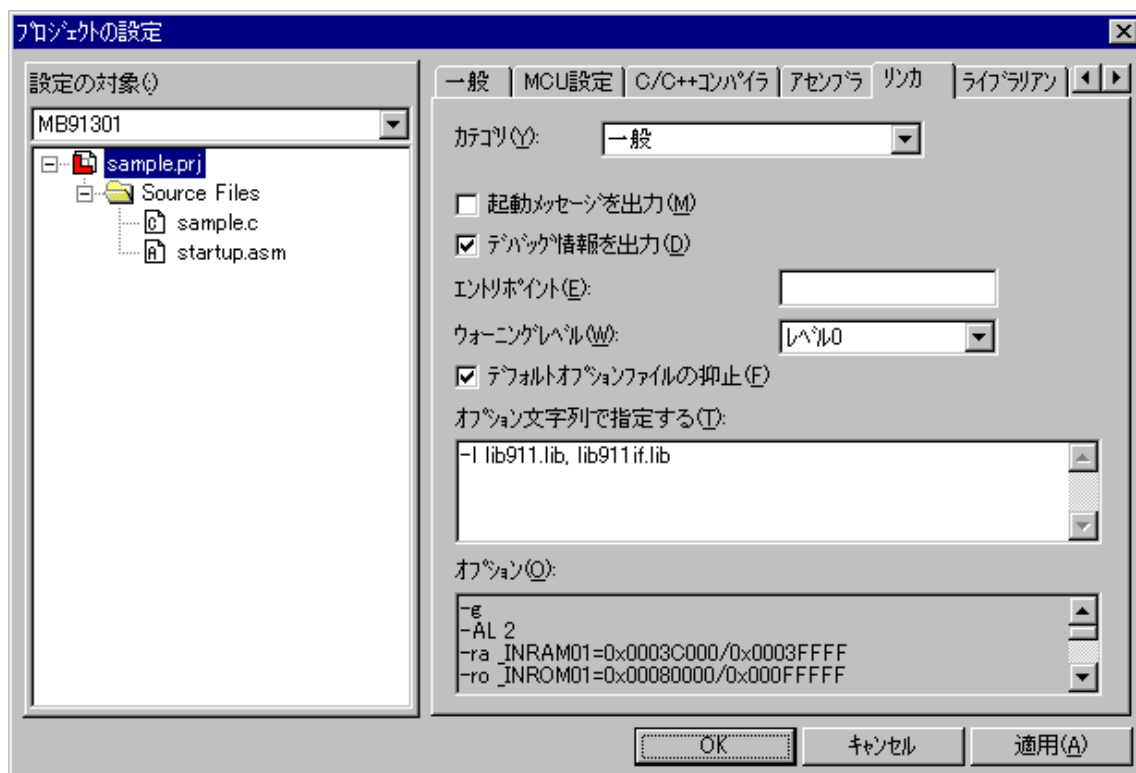
---

#### ■ リンカオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[ リンカ ] タブをクリックしてください。
- 2) [ カテゴリ ] を選択してください。
  - [ 一般 ], [ 配置 / 結合 ], [ シンボル定義 ], [ リスト出力 ], [ 絶対形式アセンブルリスト ], [ ライブラリ制御 ], [ レジスタバンク ] が選択できます。
  - 設定されたオプションは、ダイアログの下部にある [ オプション ] で確認することが可能です。
  - オプションの記述では、便利なマクロ記述を使用することができます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。

## ■ [一般] に属するオプションの設定方法

図 4.5-27 一般の設定ダイアログ



ここでは、以下のオプションの設定ができます。

- 起動メッセージを出力 (-V)
- デバッグ情報を出力 (-g)
- デフォルトオプションファイルの抑止 (-Xdof)
- エントリポイント (-e)
- ウォーニングレベル (-w) (レベル 0 ~ 2)
- オプション文字列で指定する

「エントリポイント」は、デバッガでロードしたときに PC に設定されるアドレスです。必ず、グローバルシンボルで指定してください。

「ウォーニングレベル」は、ドロップダウンリストの中から選択してください。

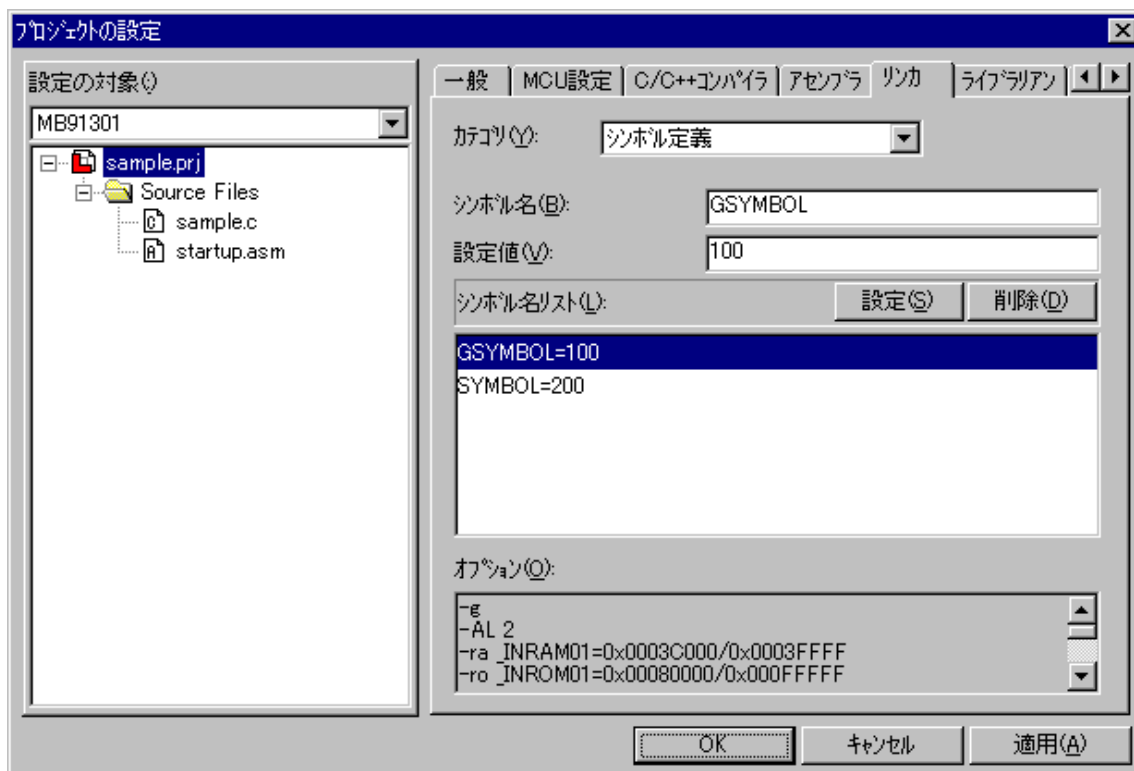
「オプション文字列で指定する」では、リンカのすべてのオプションを、コマンドラインからの起動オプションと同様に記述できます。リンカオプション設定のどのカテゴリにも属さないオプションを指定する場合に、ここに直接記述してください。

## ■ 配置 / 結合の設定について

「4.5.5.6 セクションの配置 / 結合指定」を参照してください。

## ■ シンボル定義の設定方法

図 4.5-28 シンボル定義の設定ダイアログ



設定を行う対象が複数の場合、一部のみに存在するシンボル名は灰色で表示されます。

- 1) [シンボル定義] カテゴリを選択してください。
  - シンボル定義の設定ダイアログ (図 4.5-28) が表示されます。
- 2) [シンボル名] を指定してください。
- 3) [設定値] を指定してください。
- 4) [設定] ボタンをクリックしてください。

## ■ シンボル定義の設定解除方法

- 1) [シンボル定義] カテゴリを選択してください。
  - シンボル定義の設定ダイアログ (図 4.5-28) が表示されます。
- 2) [シンボル名リスト] から解除したいシンボル名を選択してください。
- 3) [削除] ボタンをクリックしてください。

## ■ リスト出力の設定方法

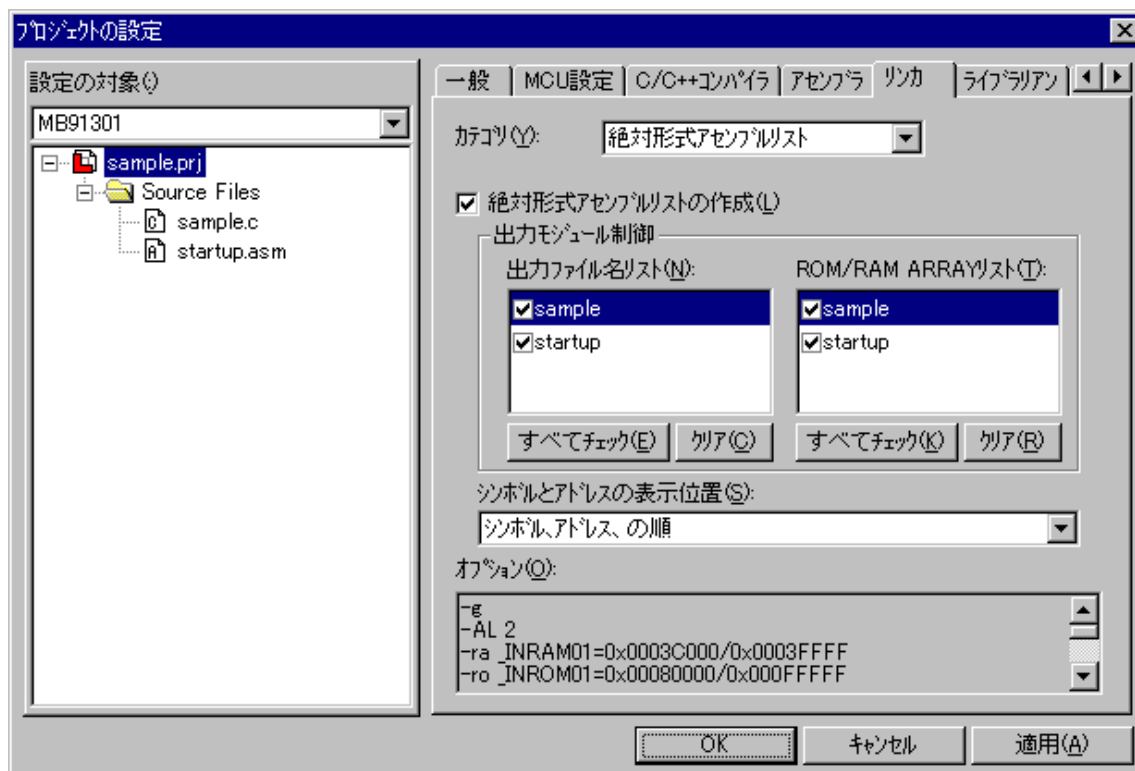
図 4.5-29 リスト出力の設定ダイアログ



- 1) [リスト出力] カテゴリを選択してください。
  - リスト出力の設定ダイアログ (図 4.5-29) が表示されます。
- 2) 作成するリストにチェックマークを設定してください。
  - マップリストファイルの作成
  - 外部シンボル相互参照情報リストを作成
  - ローカルシンボルリストを作成
  - セクション詳細マップリスト
    - 上記のどのリストファイルも出力しないときは、このダイアログではほかに設定する項目はありません。いずれかのリストファイルを出力する場合のみ、3)以降を実行してください。
- 3) 必要に応じて、[メモリ使用情報リストの出力] をチェックしてください。
- 4) 必要に応じて、[名前省略解除] をチェックしてください。
  - リストファイルの1行に出力しきれない長い名前のシンボルを折り返してすべて出力されます。
- 5) 改ページの指定をしてください。[改頁を抑止] にチェックマークが付いている場合は、改ページしません。
  - 改ページしない場合は、行数の設定はできません。
- 6) 必要に応じて、[行数]、[桁数] を設定してください。

## ■ 絶対形式アセンブルリストに関するオプションの設定方法

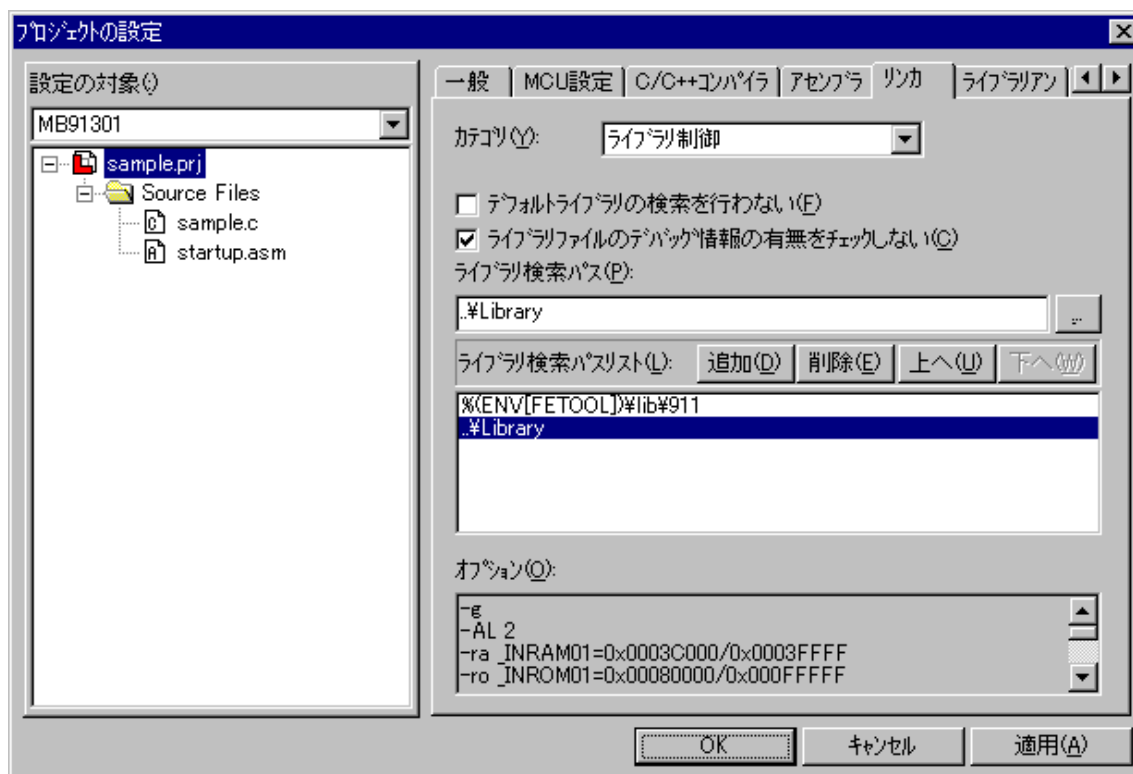
図 4.5-30 絶対アセンブルリストの設定ダイアログ



- 1) [絶対形式アセンブルリスト] カテゴリを選択してください。
  - 絶対形式アセンブルリストの設定ダイアログ (図 4.5-30) が表示されます。
- 2) [絶対形式アセンブルリストの作成] にチェックマークを設定してください。
  - 絶対形式アセンブルリストファイルを出力しないときは、このダイアログではほかに設定する項目はありません。絶対形式アセンブルリストファイルを出力する場合のみ、3) 以降を実行してください。
- 3) 絶対形式アセンブルリストを出力したいモジュールを、[出力ファイル名リスト] から選択して、ファイル名にチェックを付けてください。
- 4) ROM/RAM ARRAY リストを出力したいモジュールを、[ROM/RAM ARRAY リスト] から選択して、ファイル名にチェックを付けてください。
- 5) [シンボルとアドレスの表示位置] をドロップダウンリスト (シンボル、アドレスの順 / アドレス、シンボルの順) から選択してください。

## ■ ライブラリ制御に関するオプションの設定方法

図 4.5-31 ライブラリ制御の設定ダイアログ



### ■ デフォルトライブラリを使用しない場合の設定方法

- 1) [ ライブラリ制御 ] カテゴリを選択してください。
  - ライブラリ制御の設定ダイアログ ( 図 4.5-31 ) が表示されます。
- 2) [ デフォルトライブラリの検索を行わない ] に、チェックマークを付けてください。

### ■ ライブラリファイルのモジュールに対するデバッグ情報存在チェックを抑止する場合の設定方法

- 1) [ ライブラリ制御 ] カテゴリを選択してください。
  - ライブラリ制御の設定ダイアログ ( 図 4.5-31 ) が表示されます。
- 2) [ ライブラリファイルのデバッグ情報の有無をチェックしない ] に、チェックマークを付けてください。

## ■ ライブラリ検索パスの設定方法

- 1) [ ライブラリ制御 ] カテゴリを選択してください。
  - ライブラリ制御の設定ダイアログ ( 図 4.5-31 ) が表示されます。
- 2) [ ライブラリ検索パス ] を設定してください。
  - 入力フィールドの右隣にある [ 参照 ] ボタンをクリックすると、パスの参照ができます。
- 3) [ 追加 ] ボタンをクリックしてください。
  - [ ライブラリ検索パスリスト ] の最後に追加されます。

## ■ ライブラリ検索パスの設定解除方法

- 1) [ ライブラリ制御 ] カテゴリを選択してください。
  - ライブラリ制御の設定ダイアログ ( 図 4.5-31 ) が表示されます。
- 2) [ ライブラリ検索パスリスト ] から解除したいパスを選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。

## ■ ライブラリ検索パスの検索順位の変更方法

ライブラリ検索パスの検索は、[ ライブラリ検索パスリスト ] の上から順に行います。以下の手順で、リストへの登録順を変更できます。

- 1) [ ライブラリ制御 ] カテゴリを選択してください。
  - ライブラリ制御の設定ダイアログ ( 図 4.5-31 ) が表示されます。
- 2) [ ライブラリ検索パスリスト ] から順序を変更したいパスを選択してください。
- 3) [ 上へ ] ボタンまたは [ 下へ ] ボタンをクリックして、適切な位置まで移動させてください。

### 4.5.5.6 セクションの配置 / 結合指定

ここでは、リンカのオプションのうち、セクションの配置 / 結合についてのオプション設定方法について説明します。

#### ■ セクションの配置 / 結合指定方法

- 1) プロジェクトの設定ダイアログで、[リンカ]タブをクリックしてください。
- 2) [配置 / 結合]カテゴリを選択してください。
  - 配置 / 結合の設定ダイアログ (図 4.5-32) が表示されます。
- 3) [自動配置] をドロップダウンリスト (なし/モード/モード2) から選択してください。
  - なし
    - セクションの自動配置を行いません。
  - モード1
    - [ROM/RAM領域名リスト]で指定した領域内へセクション配置を行う時、アブソリュートセクションが存在していれば配置アドレスが重ならないようにリロケータブルセクションの配置を行います。このとき、アライメント値およびサイズの大きいセクションから順に配置し、空き領域が最小となるような最適配置を行います。
  - モード2
    - 配置指定が行われなかったセクションに対してそのセクションの種別から、ROM領域に配置すべきか、RAM領域に配置すべきかをリンカが判断し、それぞれの領域内の空き領域に自動配置します。

これ以降は、具体的にセクションをどのように配置するかによって、設定方法が異なります。

図 4.5-32 配置 / 結合の設定ダイアログ

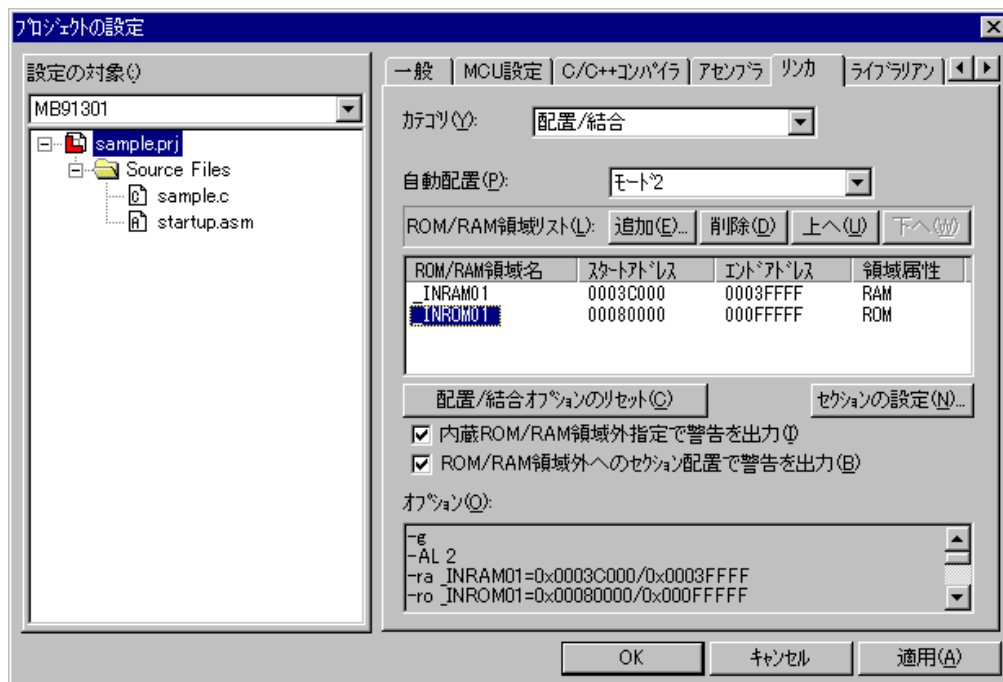




図 4.5-33 セクションの設定ダイアログ



### ■ ROM/RAM の領域を指定してそこにセクションを配置する場合

- 1) ROM/RAM 領域を定義してください。
  - ROM/RAM 領域の定義方法を参照してください。
- 2) [ROM/RAM 領域リスト] から設定したい領域を選択してください。
- 3) [セクションの設定] ボタンをクリックしてください。
  - セクションの設定ダイアログ (図 4.5-33) が開きます。
- 4) その領域に配置するセクション名を指定してください。
  - ワイルドカード文字も使用できます。
- 5) 必要に応じて、内容種別をドロップダウンリストから選択して設定してください。  
(なし /code/data/stack/const/IO)
- 6) [設定] ボタンをクリックしてください。
  - [セクション名リスト] の最後に追加されます。
- 7) すべての設定が完了したら、[OK] ボタンをクリックしてください。

領域内でのセクションの配置順は、[セクション名リスト] の順になります。この順序を入れ替えたいときは、セクション配置順の変更を参照してください。

## ■ 先頭アドレス指定してそこにセクションを配置する場合

- 1) [ セクションの設定 ] ボタンをクリックしてください。
  - セクションの設定ダイアログ ( 図 4.5-33 ) が開きます。
- 2) [ROM/RAM 領域名] にドロップダウンリストから, [ アドレスで指定 ] を選択してください。
- 3) その領域に配置するセクション名を指定してください。
  - ワイルドカード文字も使用できます。
- 4) 配置する先頭アドレスを指定してください。
- 5) 必要に応じて, 内容種別をドロップダウンリストから選択して設定してください。
- 6) [ 設定 ] ボタンをクリックしてください。
  - [ セクション名リスト ] の最後に追加されます。
- 7) すべての設定が完了したら, [OK] ボタンをクリックしてください。

複数のセクションを, 指定したアドレスから一続きに配置したい場合は, 上記の 6) の操作後, [ セクション名 ] と [ 内容種別 ] のみを設定して, [ 設定 ] ボタンをクリックする操作を繰り返してください。

セクションは, このときに指定した順序で配置されます。配置順を入れ替えたいときは, セクション配置順の変更を参照してください。

## ■ セクション配置順の変更

- 1) [ セクションの設定 ] ボタンをクリックしてください。
  - セクションの設定ダイアログ ( 図 4.5-33 ) が開きます。
- 2) [ROM/RAM 領域名] で変更したい領域を選択してください。直接アドレス指定した場合は, [ アドレスで指定 ] を選択してください。
  - その領域 ( またはアドレス ) に配置するセクションの一覧が, [ セクション名リスト ] に表示されます。
- 3) 順序を変更したいセクション名を選択して, [ 上へ ] ボタンまたは [ 下へ ] ボタンをクリックして, 希望する位置へ移動してください。
  - アドレスで指定の場合は, アドレスで指定の場合の注意もご覧ください。

## ■ アドレスで指定の場合の注意

[ セクション名リスト ] 中の記述と, 生成されるリンカオプションの関係の例を示します。セクションの配置順を変更する場合には, 十分注意して行ってください。

【例 1】[ セクション名リスト ] が以下の内容の場合

```
sec1=H'00001000
sec2
sec3=H'0000F000
sec4
```

生成されるセクションは, 以下になります。

```
-sc sec1+sec2=H'00001000 -sc sec3+sec4=H'0000F000
```

【例 2】sec4 の配置順を以下のように変更した場合

```
sec1=H'00001000
sec2
```

```
sec4
```

```
sec3=H'0000F000
```

生成されるセクションは、以下になります。

```
-sc sec1+sec2+sec4=H'00001000 -sc sec3=H'0000F000
```

## ■ ROM/RAM 領域の定義方法

- 1) [ 追加 ] ボタンをクリックしてください。
  - ROM/RAM 領域名の指定ダイアログ ( 図 4.5-34 ) が開きます。

図 4.5-34 ROM/RAM 領域名の指定ダイアログ



- 2) [ROM/RAM 領域名] を指定してください。
  - 既に設定している名前と重複しない名前を付けてください。
- 3) [ 領域属性 ] を選択してください。
  - [ROM] または [RAM] が選択可能です。
- 4) [ スタートアドレス ] と [ エンドアドレス ] を指定してください。
  - これは、ROM/RAM 領域の先頭および最終アドレスです。
- 5) [OK] ボタンをクリックしてください。
  - [ROM/RAM 領域リスト] に登録されます。

自動配置 ( モード 2 ) の時、リンクは配置可能な ROM/RAM 領域の検索を [ROM/RAM 領域リスト] の上から順に行いますので [ 上へ ], [ 下へ ] ボタンをクリックして適切な位置へ移動してください。

## ■ ROM/RAM 領域定義の解除方法

- 1) [ROM/RAM 領域リスト] から削除したい領域を選択してください。
- 2) [ 削除 ] ボタンをクリックしてください。

## ■ ROM/RAM 領域の指定が内蔵 ROM/RAM の範囲外の場合に警告を出力する方法

- 1) [ 内蔵 ROM/RAM 領域外指定で警告を出力 ] に、チェックマークを付けてください。
  - ダイアログで内蔵ROM/RAM領域外が指定されると警告ダイアログが表示されるようになります。
  - 内蔵 ROM/RAM 領域外が設定されていると、リンク時に警告を出力します。

## ■ セクション配置が ROM/RAM 領域の指定範囲外の場合に警告を出力する方法

- 1) [ROM/RAM 領域外へのセクション配置で警告を出力] に、チェックマークを付けてください。
  - 指定の ROM/RAM 領域 (-ro, -ra オプション) または、MCU 内蔵 ROM/RAM 領域の外にセクションが配置されていると、リンク時に警告を出力します。

## ■ MCU 情報を元に [ 配置 / 結合 ] のオプションを初期化する場合

1) [ 配置 / 結合オプションのリセット ] ボタンをクリックしてください。

- 配置 / 結合オプションがリセットされます。

リセットを行うと、配置 / 結合オプションは以下のように設定されます。

- 自動配置 (-AL 2)
  - 自動配置はモード 2 ( リンカによる最適な自動配置 ) が設定されます。
- ROM/RAM 領域 (-ro, -ra)
  - ROM/RAM 領域をすべてクリアし、現在選択されている MCU の内蔵 ROM/RAM アドレスを設定します。
- セクション情報 (-sc)
  - すべてクリアします。
- ROM/RAM 領域の指定が内蔵 ROM/RAM の範囲外るとき警告するを出力する (-check\_rora)
  - 警告を出力するように設定します。ただし、シングルチップモードがない MCU が選択された場合は、警告を出力しないようにします。
- セクション配置が ROM/RAM 領域の指定範囲外るときに警告を出力する (-check\_locate)
  - 警告を出力するようにします。

### 4.5.5.7 ライブラリアンオプションの設定

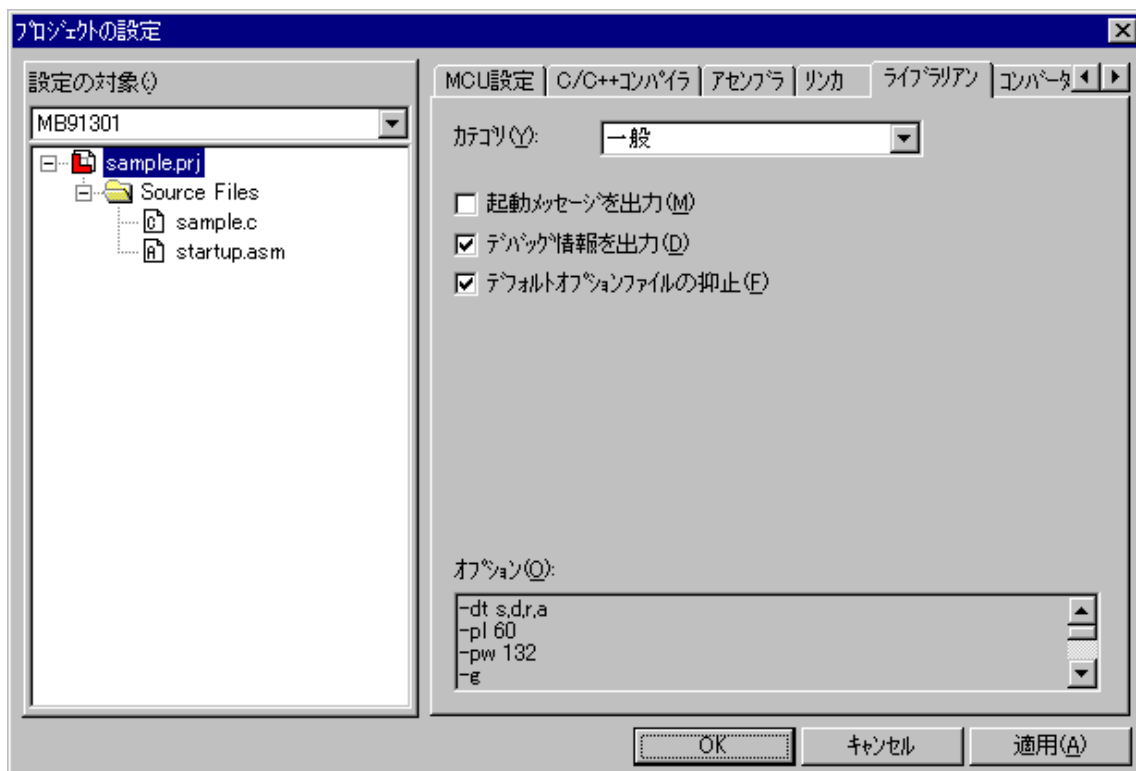
ここでは、ライブラリアンのオプションの設定方法について説明します。

#### ■ ライブラリアンオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[ライブラリアン] タブをクリックしてください。
- 2) [カテゴリ] を選択してください。
  - [一般]、[リスト出力] が選択できます。
  - 設定されたオプションは、ダイアログの下部にある [オプション] で確認することが可能です。

#### ■ [一般] に属するオプションの設定方法

図 4.5-35 ライブラリアンオプションの設定ダイアログ (一般)



ここでは、以下のオプションの設定ができます。

- 起動メッセージを出力 (-V)
- デバッグ情報を出力 (-g)
- デフォルトオプションの抑止 (-Xdof)

## ■ リスト出力の設定方法

図 4.5-36 ライブラリアンオプションの設定ダイアログ (リスト出力)



- 1) 出力内容を選択してください。
  - [ モジュール毎のセクション名とサイズを出力 ]
  - [ モジュール毎の外部定義シンボルを出力 ]
  - [ モジュール毎の外部参照シンボルを出力 ]
  - [ 全体の外部定義シンボルと未解決外部参照シンボルを出力 ]
- 2) 改ページしたくない場合は, [ 改頁を抑止 ] にチェックマークを付けてください。
  - 改ページしない場合, [ 行数 ] の設定はできません。
- 3) 必要に応じて, [ 行数 ], [ 桁数 ] を設定してください。

## ■ ライブラリアンの起動

プロジェクトタイプが「ライブラリ」の場合にライブラリアンは起動されます。  
プロジェクトタイプの変更は, 「4.5.5.1 一般」を参照してください。

### 4.5.5.8 コンバータオプションの設定

ここでは、コンバータのオプションの設定方法について説明します。

#### ■ コンバータオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[ コンバータ ] タブをクリックしてください。
- 2) ロードモジュールコンバータを起動する場合は、[ ロードモジュールコンバータを起動 ] をチェックしてください。
- 3) 変換形式を選択してください。
  - モトローラ S フォーマット (f2ms)
    - リンカ出力の絶対形式ロードモジュールを S フォーマットに形式変換します。  
0 ~ 0xFFFFFFFF 番地のデータが変換対象になります。
  - インテル HEX 形式 (f2is)
    - リンカ出力の絶対形式ロードモジュールを HEX8 フォーマットに形式変換します。0 ~ 0xFFFF 番地のデータが変換対象になります。
    - この形式は、旧版との互換性維持のため残されています。HEX8 フォーマットへの変換にはインテル HEX32 形式 (f2hs) を使用されることを推奨いたします。
  - インテル拡張 HEX 形式 (f2es)
    - リンカ出力の絶対形式ロードモジュールを HEX16 フォーマットに形式変換します。0 ~ 0xFFFFF 番地のデータが変換対象になります。
    - この形式は、旧版との互換性維持のため残されています。HEX16 フォーマットへの変換にはインテル HEX32 形式 (f2hs) を使用されることを推奨いたします。
  - インテル HEX32 形式 (f2hs)
    - リンカ出力の絶対形式ロードモジュールを HEX フォーマットに形式変換します。0 ~ 0xFFFFFFFF 番地のデータが変換対象になります。
- 4) 必要に応じて以下の設定を行ってください。
  - 共通オプション
    - 起動メッセージを出力 (-V)
    - デフォルトオプションの抑止 (-Xdof)
  - モトローラ S フォーマット (f2ms) 用オプション
    - 出力フォーマット - 指定しない  
データのアドレスに合わせて、S1/S2/S3 レコードの混在で出力します。
    - 出力フォーマット - S1 レコード (16-bit address) (-S1)  
S1 レコード (出力可能範囲 0x0000 ~ 0xFFFF) で出力します。
    - 出力フォーマット - S2 レコード (24-bit address) (-S2)  
S2 レコード (出力可能範囲 0x000000 ~ 0xFFFFFF) で出力します。
    - 出力フォーマット - S3 レコード (32-bit address) (-S3)  
S3 レコード (出力可能範囲 0x00000000 ~ 0xFFFFFFFF) で出力します。
    - 出力ファイルの整形 (-adjust)

フォーマットアジャスタを自動的に呼び出し、データ出力形式を整形します。

- 出力範囲 (-ran)  
出力ファイルの整形 (-adjust) オプション指定時に、整形する範囲をアドレスで指定します。[ 自動 ] の場合、整形対象となる開始 / 終了アドレスを絶対形式ロードモジュールから取得し、自動的に設定します。
- パディングデータ (-p)  
出力ファイルの整形 (-adjust) オプション指定時、ファイルにデータの存在しない箇所が指定した値のデータで埋まります。
- インテル HEX32 フォーマット (f2hs) 用オプション
  - 出力フォーマット - 指定なし  
データのアドレスに合わせて、HEX8/HEX16/HEX32 の混在で出力します。
  - 出力フォーマット - HEX8 (16-bit address) (-I16)  
HEX8 フォーマット (出力可能範囲 0x0000 ~ 0xFFFF) で出力します。
  - 出力フォーマット -HEX16 (20-bit address) (-I20)  
HEX16 フォーマット (出力可能範囲 0x00000 ~ 0xFFFFF) で出力します。
  - 出力フォーマット - HEX32 (32-bit address) (-I32)  
HEX32 フォーマット (出力可能範囲 0x00000000 ~ 0xFFFFFFFF) で出力します。
  - 出力ファイルの整形 (-adjust)  
フォーマットアジャスタを自動的に呼び出し、データ出力形式を整形します。
  - スタートアドレスレコードの出力 (-entry)  
スタートアドレスレコードを出力します。スタートアドレスレコードは、ロード時のスタートアドレスとして使用されます (デバッグ時)。
  - 出力範囲 (-ran)  
出力ファイルの整形 (-adjust) オプション指定時に、整形する範囲をアドレスで指定します。[ 自動 ] の場合、整形対象となる開始 / 終了アドレスを絶対形式ロードモジュールから取得し、自動的に設定します。
  - パディングデータ (-p)  
出力ファイルの整形 (-adjust) オプション指定時、ファイルにデータの存在しない箇所が指定した値のデータで埋まります。



オプションの記述では、便利なマクロ記述を使用することができます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で利用できるマクロ記述」を参照してください。

図 4.5-37 コンバータオプションの設定ダイアログ



### 4.5.5.9 デバッグオプションの設定

ここでは、デバッグのオプションの設定方法について説明します。

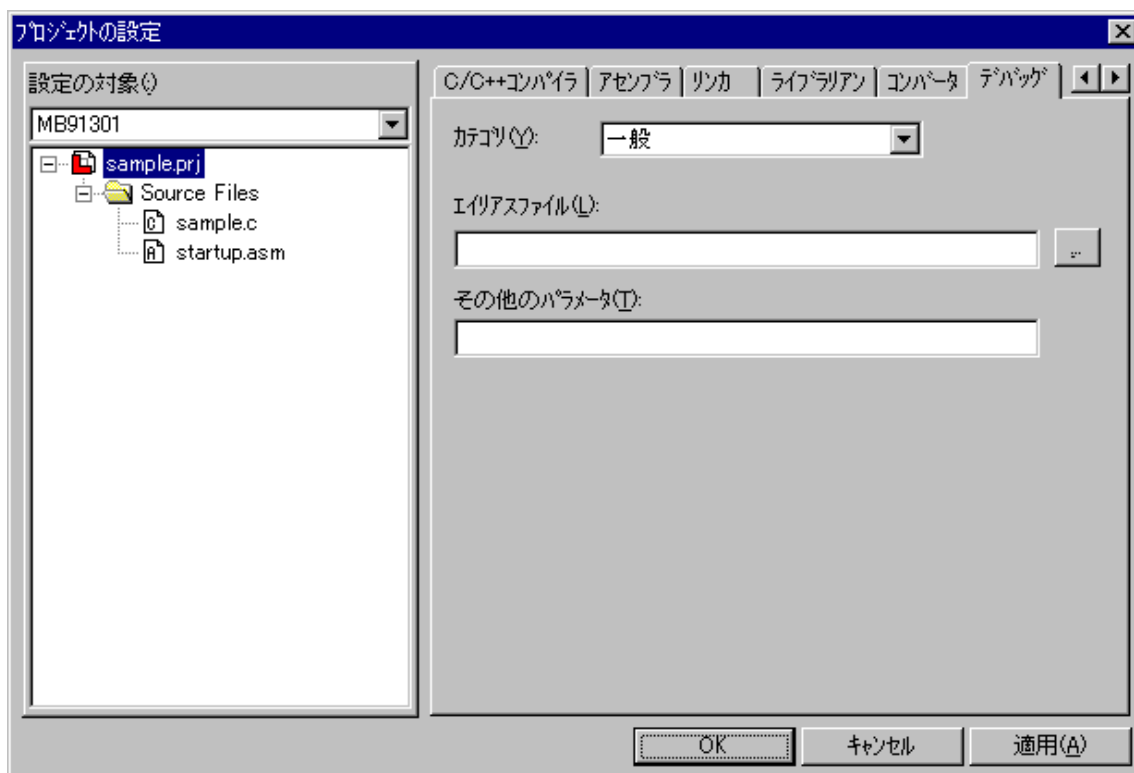
#### ■ デバッグオプションの設定方法

- 1) プロジェクトの設定ダイアログで、[デバッグ]タブをクリックしてください。
  - デバッグオプション設定ダイアログ(図4.5-38)が開きます。
- 2) [カテゴリ]を選択してください。
  - [一般]、[セットアップ]が選択できます。

#### ■ [一般]に属するオプションの設定方法

- 1) エイリアスファイルを設定してください。
  - 入力フィールドの右隣の参照ボタンをクリックすると、ファイルの参照ができます。
- 2) 上記以外のオプションを設定する場合は、[その他のパラメータ]に記述してください。

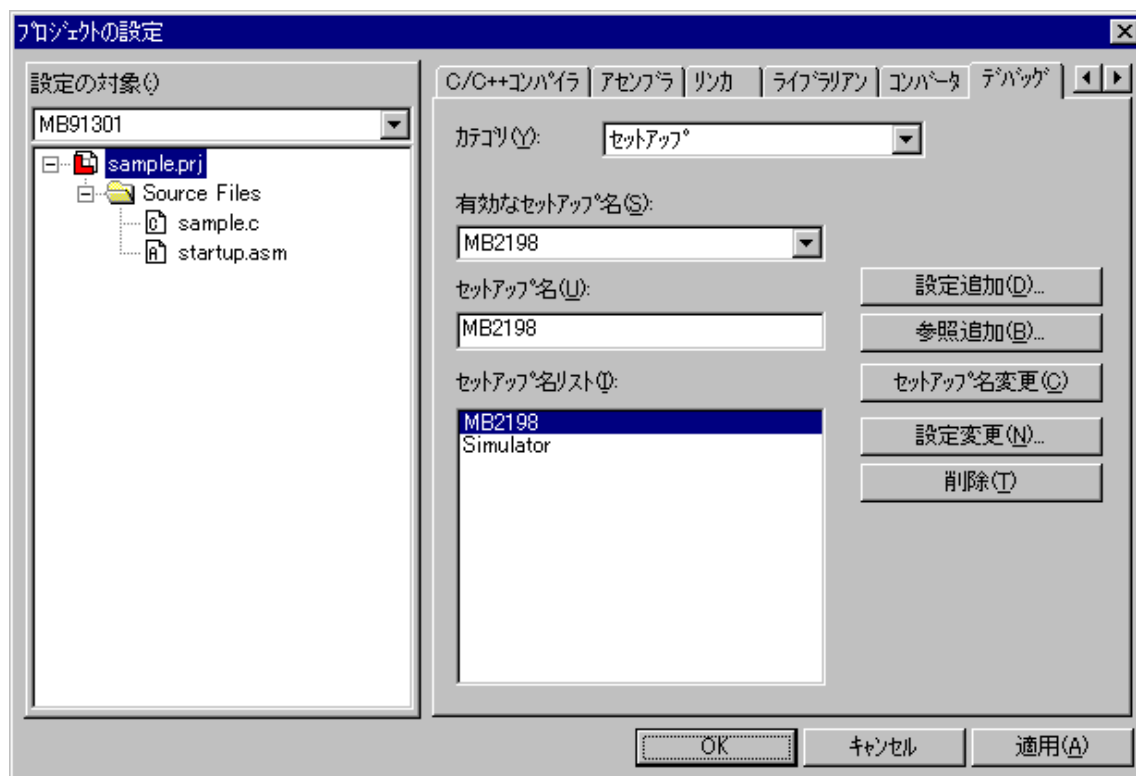
図 4.5-38 デバッグオプション設定ダイアログ (一般)



## ■ セットアップの設定

- 1) プロジェクトの設定ダイアログで,[デバッグ]タブをクリックしてください。
- 2) [セットアップ]カテゴリを選択してください。
- 3) [セットアップ名]の設定を行ってください。
  - 既に登録されているセットアップ名と重複しない名前を付けてください。
- 4) [設定追加]ボタンまたは[参照追加]ボタンをクリックしてください。
  - [設定追加]は,セットアップウィザードを起動して,新規に設定を追加します。  
「4.7.2.5 セットアップウィザード」を参照してください。
  - [参照追加]は,既に設定されているファイルから情報を読み込んで,設定します。  
ファイル選択のダイアログが表示されますのでファイルを選択して,[開く]ボタンをクリックしてください。

図 4.5-39 デバッグオプション設定ダイアログ (セットアップ)



### ■ セットアップの設定削除

- 1) プロジェクトの設定ダイアログで,[デバッグ]タブをクリックしてください。
- 2) [セットアップ]カテゴリを選択してください。
- 3) [セットアップ名リスト]から削除するセットアップ名を選択してください。
- 4) [削除]ボタンをクリックしてください。

### ■ セットアップの設定変更

- 1) プロジェクトの設定ダイアログで,[デバッグ]タブをクリックしてください。
- 2) [セットアップ]カテゴリを選択してください。
- 3) [セットアップ名リスト]から変更するセットアップ名を選択してください。
- 4) [設定変更]ボタンをクリックしてください。
  - セットアップウィザードが起動します。「4.7.2.5 セットアップウィザード」を参照してください。

### ■ セットアップ名の変更

- 1) プロジェクトの設定ダイアログで,[デバッグ]タブをクリックしてください。
- 2) [セットアップ]カテゴリを選択してください。
- 3) [セットアップ名リスト]から変更するセットアップ名を選択してください。
- 4) [セットアップ名]を入力してください。
- 5) [セットアップ名変更]ボタンをクリックしてください。

## 4.5.6 カスタマイズビルドの設定

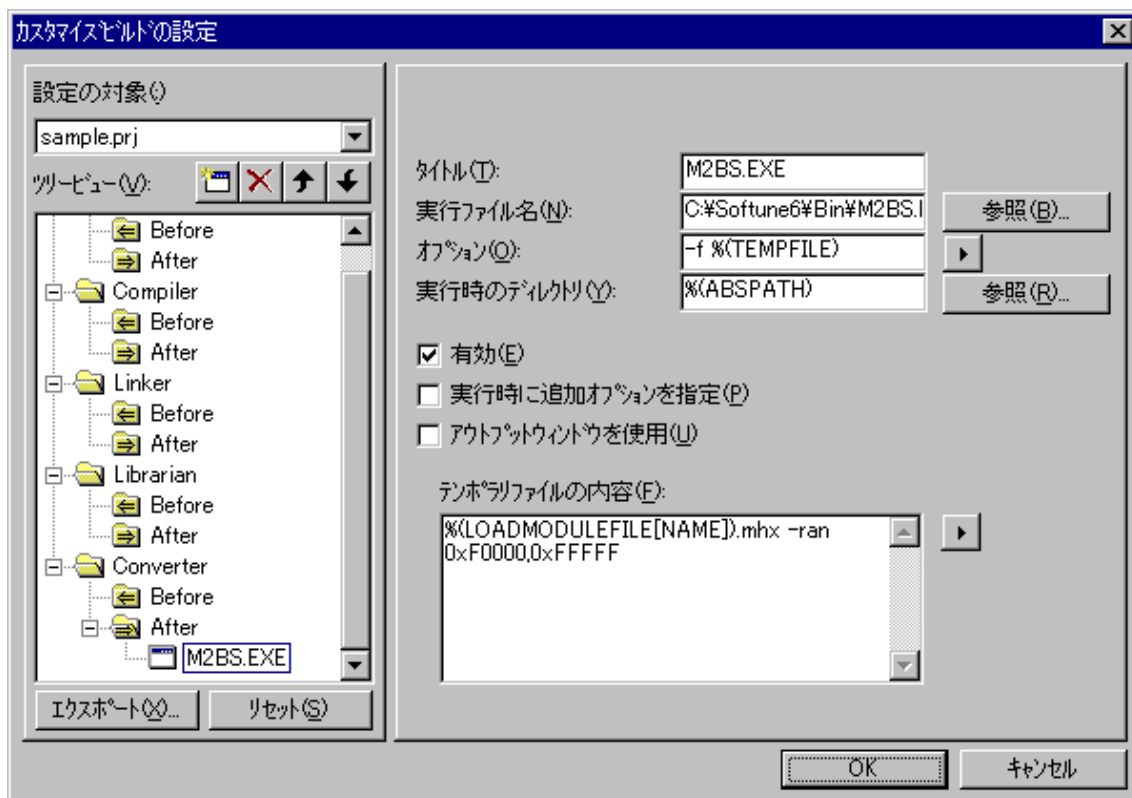
ここでは、メイクやビルド時に言語ツールを実行する前後に、別ツールを起動する方法について説明します。

### ■ ビルドのカスタマイズ機能について

SOFTUNE Workbench では、コンパイル/アセンブル/メイク/ビルドの実行時に、自動的に言語ツールの実行前や実行後に別ツールを動作させることが可能です。この機能を使用することにより、コンパイラの実行前にお客様で独自に作成されたツールを動作させたり、リンカの実行後にオブジェクトモジュールの変換ツールを動作させることが可能となります。

この設定はプロジェクトごとに保存されます。

図 4.5-40 カスタマイズビルドの設定ダイアログ



### ■ ボタン一覧

-  新規作成
-  削除
-  上に移動
-  下に移動

## ■ 設定の対象

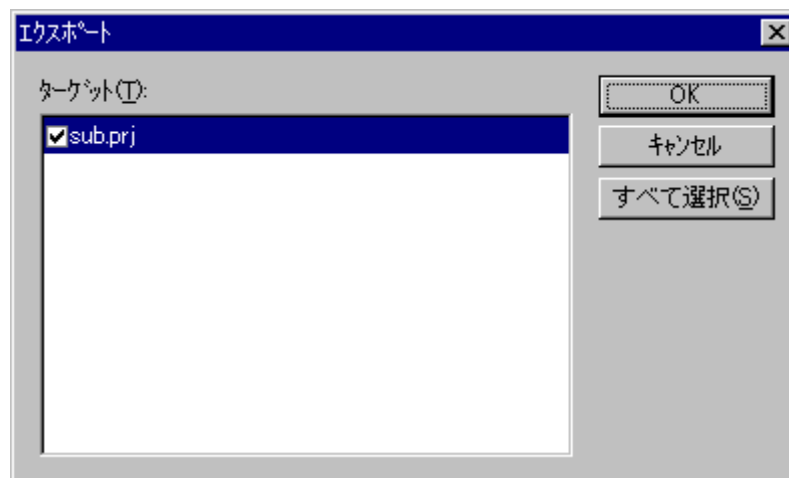
「設定の対象」には「デフォルト」と「プロジェクト名」への設定の2種類あります。  
現在どちらを設定しているかは、「設定の対象」をご覧ください。

- デフォルト
  - プロジェクトを開いていないときに、カスタマイズビルドの設定をおこなうとデフォルトの設定が変更できます。
  - ここで設定されたカスタマイズビルドの設定は、プロジェクトを新規作成する時に参照され、プロジェクトにコピーされます。
  - 「設定の対象」には「デフォルト」と表示されます。
  - \* 古いバージョンで作成されたプロジェクトを開いた場合にもこの設定が参照され、プロジェクトにコピーされます。
- プロジェクト名
  - ここで選択しているプロジェクトに対してカスタマイズビルドの設定を行います。開いているプロジェクトでコンパイル/アセンブル/メイク/ビルドを実行する時に設定されたツールが動作します。

## ■ エクスポート

「エクスポート」ボタンをクリックすると、「エクスポート」ダイアログ(図4.5-41)が表示されます。「エクスポート」ダイアログでプロジェクトを指定することで、ワークスペース内のほかのプロジェクトにツール情報をコピーすることができます。

図 4.5-41 エクスポートダイアログ



## ■ リセット

現在設定されている状態をクリアしてデフォルトの設定に戻したい場合に、[リセット] ボタンをクリックします。[リセット] ボタンは設定対象がプロジェクトの時に使用できます。

## ■ タイトル

ツール名を記述します。ツール名は重複してもかまいません。

## ■ 実行ファイル名

実行するツールのファイル名を記述します。

## ■ オプション

実行するツールのオプションを記述します。ここでは、マクロが記述できます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。

このフィールドの右横にあるボタンをクリックすると、使用できるマクロのリストが表示されます。たとえば、[ビルドファイル] - [ディレクトリ] をクリックすると、オプションのカーソル位置に % (FILE [PATH]) が挿入されます。

## ■ 実行時のディレクトリ

ツールを特定のディレクトリで実行したい場合に指定します。指定されない場合はプロジェクトディレクトリで実行されます。

## ■ 有効

コンパイル/アセンブル/メイク/ビルドの実行時にツールを実行するかどうかを指定します。チェックマークがはずされるとツールは実行されません。一時的にツールの実行を無効にしたい場合にチェックマークをはずします。

## ■ 実行時に追加オプションを指定

ツールを実行する時に追加オプションを指定できるダイアログを表示するかどうかを指定します。実行時にオプションを追加する場合には、チェックマークを付けます。

追加パラメータの設定ダイアログで設定されたオプションは、[オプション] で指定された文字列の最後にそのまま追加されます。

## ■ アウトプットウィンドウを使用

ツールの実行結果をアウトプットウィンドウに表示したい場合はチェックマークを付けます。アウトプットウィンドウを使用する場合にはいくつかの注意事項があります。詳細は『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。

## ■ テンポラリファイルの内容

[ オプション ] の指定に % (TEMPFILE) マクロが指定された場合には SOFTUNE Workbenchはツールを実行するときに一時的なファイルを新規に作成し、実行が終了すると削除します。この一時的に作成されるファイルへ書き込む内容をここに記述します。例えば、ツールへのオプションが非常に長くなるような場合に、[ オプション ] に % (TEMPFILE) を指定し、ここにオプションを記述することも可能となります。(ただし、実行するツールがオプションをファイル内に記述できる場合のみ有効です)

ここでは、マクロが記述できます。マクロについては、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で使用できるマクロ記述」を参照してください。

## ■ 設定手順

- 1) [ プロジェクト ] - [ カスタマイズビルドの設定 ] を選択してください。
  - カスタマイズビルドの設定ダイアログ (図 4.5-40) が表示されます。
- 2) ツリービューからツールを登録する位置を選択し新規作成ボタンを押してください。
  - カテゴリを選択して新規作成ボタンを押した場合はカテゴリの最後に、ツールを選択して新規作成ボタンを押した場合は選択したツールの次に「NEWTOL」というタイトルのツールが作成されます。
  - Before, After カテゴリ内のツールは上から順番に実行されます。
- 3) 登録するツールのタイトルを設定してください。
- 4) 登録するツールの実行ファイル名を設定してください。
  - このフィールドの右横にある [ 参照 ] ボタンをクリックすると、ファイル選択のファイルダイアログが表示され、そこからツールの実行ファイル名を選択できます。
- 5) 必要に応じて、オプションを設定してください。
- 6) 必要に応じて、実行時のディレクトリを設定してください。
  - このフィールドの右横にある [ 参照 ] ボタンをクリックすると、ディレクトリ選択のダイアログが表示され、そこから実行時のディレクトリを選択できます。
- 7) 必要に応じて、[ 有効 ], [ 実行時に追加オプションを指定 ], [ アウトプットウィンドウを使用 ] を設定してください。
  - ツリービューからカテゴリを選択またはツールを複数選択して、[ 有効 ], [ 実行時に追加オプションを指定 ], [ アウトプットウィンドウを使用 ] の一括変更ができます。
- 8) 必要に応じて、テンポラリファイルの内容を設定してください。
- 9) [OK] ボタンをクリックすると設定が完了します。



## ■ 削除手順

- 1) [プロジェクト] - [カスタマイズビルドの設定] を選択してください。
  - カスタマイズビルドの設定ダイアログ (図 4.5-40) が表示されます。
- 2) ツリービューから削除したいツールのタイトルを選択してください。
  - カテゴリを選択した場合はカテゴリ内のツールがすべて削除されます。
  - ツールを複数選択することもできます。
- 3) 削除ボタンをクリックしてください。

## ■ 起動順序の変更手順

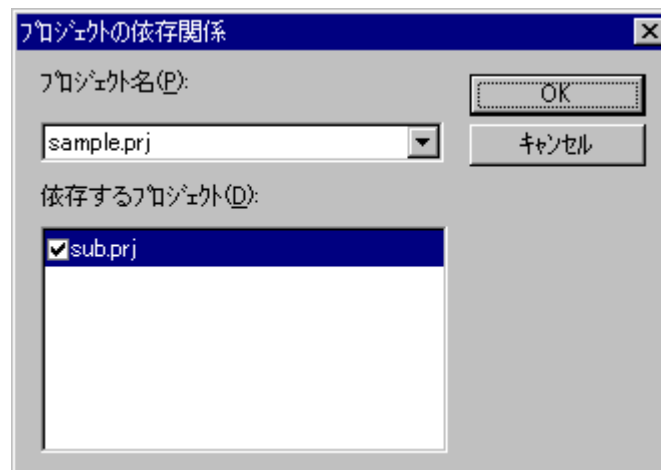
- 1) [プロジェクト] - [カスタマイズビルドの設定] を選択してください。
  - カスタマイズビルドの設定ダイアログ (図 4.5-40) が表示されます。
- 2) ツリービューから起動順序を変更したいツールを選択します。
  - Before, After カテゴリ内のツールは上から順番に実行されます。
- 3) 上へボタン, 下へボタンを押して起動したい順番にツールを移動します。
  - 移動できる範囲はツールが属するカテゴリ内のみです。

## 4.5.7 プロジェクトの依存関係

プロジェクトにサブプロジェクトを定義します。

### ■ プロジェクトの依存関係

図 4.5-42 プロジェクトの依存関係ダイアログ



- プロジェクト名
  - サブプロジェクトを定義または削除するプロジェクトのプロジェクト名です。
- 依存するプロジェクト
  - 「プロジェクト名」で選択しているプロジェクトが依存可能なプロジェクト名が表示されます。
  - チェックがついているプロジェクト名が「プロジェクト名」で選択しているプロジェクトのサブプロジェクトになります。

### ■ プロジェクトの依存関係の定義手順

定義手順については、「2.6 サブプロジェクトの定義」を参照してください。

## 4.5.8 プロジェクト構成

---

プロジェクト構成の設定を行います。

---

### ■ プロジェクト構成

プロジェクト構成の設定には、次の 2 種類があります。

- 追加と削除

プロジェクト構成の追加、削除とアクティブな構成の変更を行います。

- ビルド時の構成

サブプロジェクトのメイク / ビルドする時の構成を設定します。

### 4.5.8.1 プロジェクト構成 - 追加と削除

プロジェクト構成の追加と削除を行います。

#### ■ プロジェクト構成 - 追加と削除

図 4.5-43 プロジェクト構成の追加と削除ダイアログ



- プロジェクトと構成
  - ワークスペース内のすべてのプロジェクトとその構成が表示されます。
- 追加
  - [追加] ボタンをクリックすると, [プロジェクト構成の追加] ダイアログ (図 4.5-44) が表示され, プロジェクト構成を追加することができます。
- 削除
  - 選択しているプロジェクト, 構成を削除します。
- アクティブ
  - 選択しているプロジェクト, 構成をアクティブにします。

図 4.5-44 プロジェクト構成の追加



プロジェクト構成の追加

プロジェクト名(P):  
sample.prj

プロジェクト構成(F):

設定のコピー(C):  
MB91301

OK  
キャンセル

- プロジェクト名
  - 構成を追加するプロジェクト名が表示されます。
- プロジェクト構成
  - 新しく追加するプロジェクト構成名を設定します。
- 設定のコピー
  - 設定をコピーする構成を選択します。

#### ■ 設定手順

プロジェクト構成の追加，削除，アクティブの手順については，「2.7 プロジェクト構成の作成」を参照してください。

## 4.5.8.2 プロジェクト構成 - ビルド時の構成

サブプロジェクトをメイク / ビルドする時の構成を設定します。

### ■ プロジェクト構成 - ビルド時の構成

図 4.5-45 ビルド時の構成ダイアログ



- プロジェクト名
  - サブプロジェクトの親プロジェクトを選択します。
- 構成
  - [プロジェクト]で選択されている親プロジェクトの構成を選択します。
- メイク / ビルド時のサブプロジェクトの構成
  - [構成]で選択されている親プロジェクトの構成をメイク / ビルドした時のサブプロジェクトの構成を選択します。

### ■ 設定手順

設定の手順については、「2.10.1 プロジェクトのメイク / ビルド」を参照してください。

## 4.5.9 依存関係の更新

---

インクルードファイルの依存関係の更新を行います。

---

### ■ 依存関係の更新

プロジェクトファイルに登録されているすべてのソースファイルをチェックして、それらで使用されているすべてのインクルードファイルをプロジェクトに登録します。登録されたインクルードファイルはプロジェクトウィンドウの SRC タブの [Dependencies] カテゴリに表示されます。

## 4.5.10 コンパイル/メイク/ビルド/中止

---

コンパイル/メイク/ビルドを実行します。

---

### ■ コンパイル

指定されたソースファイルのみを、ソースファイルおよびインクルードファイルが修正されているかどうかに関係なくコンパイルします。リンクは行いません。

指定されたファイルが、アセンブラソースファイルであったときは、アセンブルを行います。

### ■ メイク

プロジェクトに登録されているすべてのソースファイルおよびインクルードファイルをチェックして、修正されているファイルのみをコンパイル/アセンブルします。

さらに、ライブラリファイル、その他のオブジェクトファイルなどで変更されているものがあればそれらを新たにリンクして、目的のファイルを作成します。

### ■ ビルド

プロジェクトに登録されているすべてのソースファイルを、修正の有無に関わらずすべてコンパイル/アセンブルします。

さらに、すべてのオブジェクトファイルおよびライブラリファイルをリンクして、目的のファイルを作成します。

### ■ 中止

コンパイル/メイク/ビルドを実行中に、何らかの理由でそれらの実行を中止したい場合に使用してください。

### ■ ソースファイルの保存の確認

コンパイル(アセンブル)、メイク、ビルドを行う前に、標準エディタで編集集中であり、まだ保存していないファイルがある場合は、保存してから実行します。ただし、[環境] - [開発環境の設定] - [ワークスペース]で、[コンパイル/アセンブル時にセーブの問い合わせを行う]にチェックしていた場合は、保存するかどうかの確認ダイアログが表示され、[いいえ]をクリックした場合は保存しない(エディット前のソース)でコンパイルします。



## 4.6 デバッグ

---

デバッグの開始と終了 , デバッグセッション時のデバッガの制御を行います。

---

### ■ デバッグの開始と終了関係

- ターゲットファイルのロード
- デバッグの開始 / デバッグの終了

### ■ デバッグセッション時のデバッガの制御関係

- 実行
- 停止
- MCU のリセット
- ブレークポイント
- ブレーク設定・解除
- イベント
- シーケンス
- コールスタック
- 時間測定
- 関数コール
- コールクリア
- ベクタ

## 4.6.1 実行

---

ここでは、デバッガのプログラム実行機能について説明します。

---

### ■ 実行

実行には、次の6つの機能があります。

- 連続実行

現在の PC の位置から連続して実行します。ブレークポイントに到達するか、[ デバッグ ] メニューから [ 停止 ] が選択されると、実行を停止します。

- ステップイン

ステップ実行を行い、PC を次の命令のアドレスに移動して停止します。関数の呼び出し命令を実行したときは、その関数の最初で停止します。

- ステップオーバー

ステップ実行を行い、PC を次の命令の先頭に移動して停止します。関数の呼び出し命令を実行したときは、その関数をすべて実行して、PC を関数呼び出し命令の次の命令のアドレスに移動して停止します。

- ステップアウト

現在の関数を終わりまで実行し、関数呼び出し元にリターンして、関数呼び出し命令の次の命令のアドレスに PC を移動して停止します。

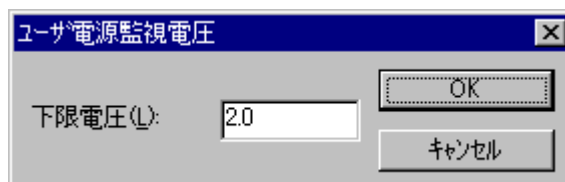
- カーソル位置まで実行

ソースウィンドウ上、逆アセンブルウィンドウ上で現在のカーソルのあるアドレスの直前の命令まで実行して、PC をカーソルのあるアドレスに移動して停止します。

- パワーオンデバッグ

パワーオンデバッグ開始時、ユーザ電源監視ダイアログを表示しますので、下限電圧の設定、確認が行えます。

図 4.6-1 ユーザ電源監視電圧ダイアログ



---

< 注意事項 >

パワーオンデバッグ機能は評価 MCU の品種により使用できない場合があります。詳しくは販売担当部門またはサポート部門へお問い合わせください。

---

## 4.6.2 停止

---

ここでは、デバッガのプログラム実行停止機能について説明します。

---

### ■ 停止

デバッガでプログラムを実行中に、強制的に実行を中断したいときに使用します。プログラムが停止すると、最後に実行した命令の次の命令のアドレスに PC が移動します。ソース行表示および逆アセンブル表示も、停止したときの PC の値に合わせて更新されます。

---

#### < 注意事項 >

モニタデバッガでは、停止操作はできません。

---

## 4.6.3 MCUのリセット

---

ここでは、デバッガのMCUリセット機能について説明します。

---

### ■ MCU リセット

MCU にリセットをかけます。

- エミュレータデバッガの場合

エミュレータにリセット信号を発行します。

- シミュレータデバッガの場合

実チップと同様に、リセットにより初期化されるレジスタについては、レジスタの初期値を設定します。その他のレジスタは、0 クリアします。

MCU へのリセットでは、ブレークポイント、マップ設定、ウォッチポイント、プログラムで使用している変数の内容などは変更されません。

---

#### < 注意事項 >

モニタデバッガでは、リセット操作はできません。

---

## 4.6.4 ブレークポイント

---

ここでは、ブレークポイントの設定、解除、一覧表示、変更などについて説明します。

---

### ■ ブレークポイント

デバッガでプログラムを実行させるときに、あるアドレスを PC が通過するまたは、あるアドレスのデータにプログラムがアクセスするときに、そこでプログラムの実行を停止したいことがあります。その位置のことを、ブレークポイントといいます。

### ■ コードブレークポイント

PC が設定したアドレスを通過するとき（そこを実行しようとしたときに）にプログラムを停止させるようなブレークポイントを、コードブレークポイントといいます。

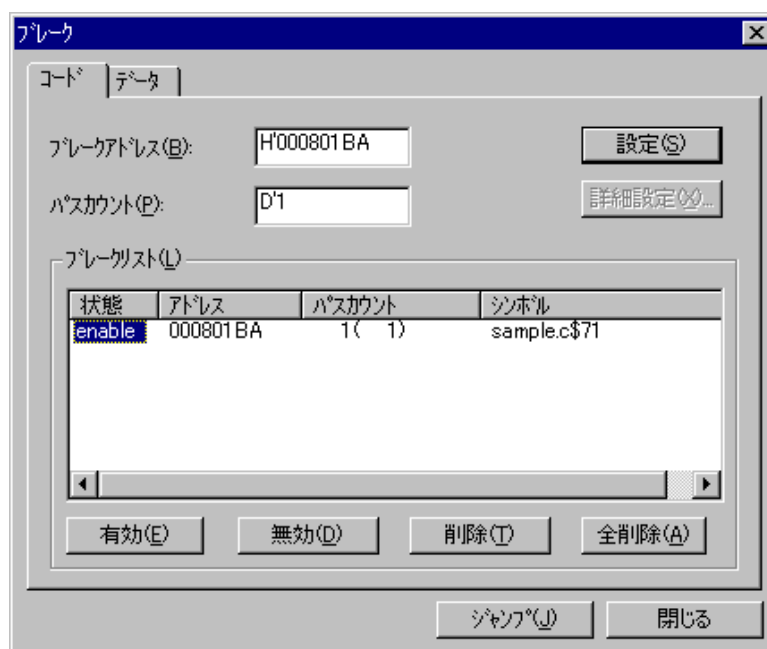
### ■ コードブレークポイントの設定

図 4.6-2 ブレークダイアログ (コード)

【エミュレータデバッガ (FR60Lite, FR80S)】



【シミュレータデバッガ】



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- 種別
  - ブレークポイントの種別を選択するフィールドです。エミュレータデバッガでのみ有効です。FR60Lite では「ハードウェア/データ監視」も選択できます。

- 残り個数
  - 選択中のブレイク種別の設定可能な残り個数を表示します。  
FR60LiteまたはFR80Sでは残り個数が10個以下になった場合に表示します。ROMパッチ機能を有するチップでは、ROMパッチのリソースをブレイクとして使用すると、ハードウェアブレイクの残り個数が増加します。
- パスカウント
  - PCが何回目にそこを通過するときにブレイクするかを設定するフィールドです。
- ブレイクリスト
  - 現在設定されているコードブレイクポイントの一覧が表示されます。  
 状態 : 有効・無効が表示されます。  
 種別 : ブレイクポイントの種別が表示されます。  
 アドレス : 設定されているアドレスが表示されます。  
 属性 : ブレイクポイントの属性が表示されます。  
 パスカウント : 設定したパスカウントが表示されます。( ) 内には、現在までに、そこを通過した回数が表示されます。  
 シンボル : そのアドレスに割り付けられているシンボルまたはソースファイルの行番号が表示されます。
- [ジャンプ] ボタン
  - ブレイクリスト中で選択されているブレイクポイントのコード位置まで、ソースウィンドウの表示開始位置を移動させます。
- [設定] ボタン, [追加] ボタン
  - 指定したアドレスにブレイクポイントを設定します。既にブレイクリストに設定されているアドレスを指定したときは、そのブレイクポイントの設定内容が変更されます。
- [詳細] ボタン
  - 指定したブレイクアドレスのブレイクポイントについて、詳細な設定ができます。FR60Lite または FR80S で有効です。詳細は「4.6.4.1 ブレイクポイントの詳細設定」を参照してください。
- [イベントリスト] ボタン
  - 全イベントの設定状況を確認するためのイベントリストを表示します。FR60Lite でブレイク種別が「ハードウェア/データ監視」を選択時に有効です。
- [ROM パッチ] ボタン
  - ROMパッチのリソースをハードウェアブレイクとして使用するためのROMパッチダイアログを表示します。
- [有効] ボタン
  - ブレイクリスト中で選択されているブレイクポイントを有効にします。
- [無効] ボタン
  - ブレイクリスト中で選択されているブレイクポイントを無効にします。ブレイクポイントが無効になるだけで、設定そのものが削除されるわけではありません。
- [削除] ボタン
  - ブレイクリスト中で選択されているブレイクポイントの設定を削除します。
- [全削除] ボタン
  - ブレイクリスト中のすべてのブレイクポイントの設定を削除します。

- ・ [変更] ボタン
  - ブレークリスト中の選択されたブレークポイントの設定を変更します。シミュレータデバッガでは表示されません。

#### < 注意事項 >

ブレークポイントのヒットカウントはユーザプログラム実行中には更新されません。したがってユーザプログラム実行中に表示されるヒットカウントはプログラム実行開始前の値となります。

## ■ ROM パッチ

チップが持つ ROM パッチのリソースをハードウェアブレークとして使用するための設定を行います。ROM パッチブレークは、ハードウェアブレークをすべて使用している場合に、番号の若いチャンネルからハードウェアブレークとして使用します。ROM パッチ機能を有するチップでのみ使用できます。

## ■ ROM パッチの設定

図 4.6-3 ROM パッチダイアログ



### ● リソース

ROM パッチのリソースの使用状況が表示されます。

ch.: Workbenchでブレークとして使用するROMパッチのチャンネルを指定します。チェックを付けたチャンネルは、ハードウェアブレークの空きがない場合にハードウェアブレークとして使用されます。ブレークが設定されているチャンネルのチェックを外した場合は、対応するブレークも一緒に削除します。

機能: ROM パッチのチャンネルの使用状況を表示します。

「BREAK」 - ブレークとして使用されていることを表します。

「RESERVED」- Workbench 以外で ROM パッチが使用されていることを表します。

「 」 - 未使用であることを表します。

アドレス: 設定されているアドレスが表示されます。

シンボル: そのアドレスに割り付けられているシンボルまたはソースファイルの行番号が表示されます。

## ■ データブレークポイント

設定したアドレスのデータにアクセスしたときにプログラムを停止させるようなブレークポイントを、データブレークポイントといいます。エミュレータデバッガの場合、FR60Lite または FR80S 使用時以外ではこの機能は使用できません。



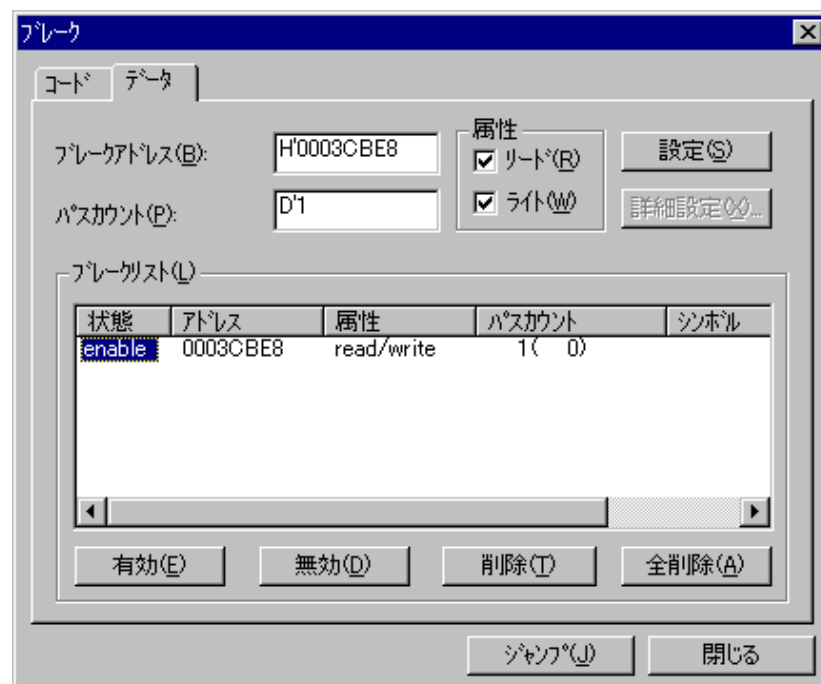
## ■ データブレイクポイントの設定

図 4.6-4 ブレークダイアログ (データ)

【エミュレータデバッガ】



【シミュレータデバッガ】



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- 残り個数
  - 選択中のブレーク種別の設定可能な残り個数を表示します。
- パスカウント
  - 指定した属性で、そこを何回目にアクセスするときにブレークするかを設定するフィールドです。シミュレータデバグのみ有効です。
- 属性
  - リードアクセスか、ライトアクセスかまたは両方なのかを指定します。
- ブレークリスト
  - 現在設定されているデータブレークポイントの一覧が表示されます。
    - 状態 : 有効・無効が表示されます。
    - アドレス : 設定されているアドレスが表示されます。
    - データアドレス : データ監視を行うアドレスが表示されます。
    - 属性 : ブレークポイントの属性が表示されます。
    - パスカウント : 設定したパスカウントが表示されます。( ) 内には、現在までに、そこを通過した回数が表示されます。
    - シンボル : そのアドレスに割り付けられているシンボルが表示されます。
- [ジャンプ] ボタン
  - ブレークリスト中で選択されているブレークポイントのデータをメモリウィンドウで表示します。
- [設定] ボタン, [追加] ボタン
  - 指定したアドレスにブレークポイントを設定します。既にブレークリストに設定されているアドレスを指定したときは、そのブレークポイントの設定内容が変更されます。
- [詳細] ボタン
  - 指定したブレークアドレスのブレークポイントについて、詳細な設定ができます。詳細は「4.6.4.1 ブレークポイントの詳細設定」を参照してください。
- [有効] ボタン
  - ブレークリスト中で選択されているブレークポイントを有効にします。
- [無効] ボタン
  - ブレークリスト中で選択されているブレークポイントを無効にします。ブレークポイントが無効になるだけで、設定そのものが解除されるわけではありません。
- [削除] ボタン
  - ブレークリスト中で選択されているブレークポイントの設定を削除します。
- [全削除] ボタン
  - ブレークリスト中のすべてのブレークポイントの設定を解除します。
- [変更] ボタン
  - ブレークリスト中の選択されたブレークポイントの設定を変更します。  
MB2197 では表示されません。

## ■ ブレーク設定・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、現在カーソルのあるアドレスにブレークポイントを設定したり、そこに設定されているブレークポイントを解除するには、以下のようにすると簡単に行うことができます。

- [デバッグ]メニューから[ブレーク設定・解除]を選択してください。
  - 一度選択することにより、設定と解除を交互に繰り返します。
- それぞれのウィンドウのブレークポイント表示フィールドで左ボタンをクリックしてください。
  - 一度クリックすることにより、設定と解除を交互に繰り返します。

## 4.6.4.1 ブレークポイントの詳細設定

ここでは、ブレークポイントの詳細設定について説明します。

### ■ コードブレーク詳細 (シミュレータデバッガ)

図 4.6-5 コードブレーク詳細ダイアログ (シミュレータデバッガ)



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- パスカウント
  - PC が何回目にそこを通過するときにブレークするかを設定するフィールドです。
- 操作
  - ブレークポイントヒット時の、コマンド処理後の再実行または実行停止を設定します。
- コマンド
  - ブレークポイントヒット時に実行するコマンド列を設定します。

### ■ データブレーク詳細 (シミュレータデバッガ)

図 4.6-6 データブレーク詳細ダイアログ (シミュレータデバッガ)



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。ブレークアドレス
- パスカウント
  - PCが何回目にそこを通過するときにブレークするかを設定するフィールドです。
- 属性
  - データアクセス時の属性を指定します。
- 操作
  - ブレークポイントヒット時の、コマンド処理後の再実行または実行停止を設定します。
- コマンド
  - ブレークポイントヒット時に実行するコマンド列を設定します。

< 注意事項 >

STUB 機能の詳細については『SOFTUNE Workbench 機能説明書』の「2.1.7 STUB 機能」を参照してください。

## ■ コードブレーク詳細 (MB2198)

図 4.6-7 コードブレーク詳細ダイアログ (MB2198)



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- 残り個数
  - 選択中のブレーク種別の設定可能な残り個数を表示します。
- 種別
  - ブレークポイントの種別を選択するフィールドです。
- 監視条件
  - チェックするとデータ監視条件を有効にします。
- データ監視条件
  - データ監視条件を設定します。図 4.6-8 のダイアログが表示されます。

図 4.6-8 データ監視条件設定ダイアログ

データ監視条件を設定します。1ポイントのみ設定可能です。

- ブレイクアドレス
  - 設定されているブレイクアドレスが表示されています。
- 比較条件
  - データ監視をおこなう条件を指定します。
- 間接条件
  - アドレス間接またはレジスタ間接のどちらかを指定します。
- レジスタ番号 (R0+ .. R15+)
  - レジスタ間接を選択した場合に、レジスタ番号を指定します。
- 比較アドレス
  - データ監視時のアドレス条件を設定するフィールドです。レジスタ間接を指定した場合は、オフセット値を指定します。
- サイズ (バイト / ハーフワード / ワード)
  - データアクセス時のデータサイズを指定します。
- 比較データ
  - データ監視時の比較データを設定するフィールドです。

#### < 注意事項 >

モニタリング機能を使用する場合、データ監視条件の設定は行わないでください。データ監視条件設定後モニタリング機能を使用すると「MCU 実行中は使えないコマンドです」のエラーメッセージが表示されます。

## ■ コードブレイク詳細：デバッグ種別（ハードウェア/カウント）

図 4.6-9 コードブレイク詳細ダイアログ（ハードウェア/カウント）



- ブレイクアドレス

ブレイクポイントを設定するアドレスを指定するフィールドです。

- アドレスマスク

指定したアドレスに対するマスク値を指定するフィールドです。

- パスカウント

指定した属性で、そこを何回目にアクセスするときにブレイクするかを設定するフィールドです。

- 監視条件

チェックするとデータ監視条件を有効にします。

- データ監視条件

データ監視条件を設定します。図 4.6-8 のダイアログが表示されます。

---

### < 注意事項 >

FR60Lite または FR80S 使用時以外はこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.4 ブレイク」を参照してください。

---

## ■ コードブレーク詳細：デバッグ種別（ハードウェア/データ監視）

図 4.6-10 コードブレーク詳細ダイアログ（ハードウェア/データ監視）

- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- アドレス
  - データ監視をおこなうアドレスまたはシンボルを指定するフィールドです。
- アドレスマスク
  - 指定したアドレスに対するマスク値を指定するフィールドです。
- サイズ（バイト / ハーフワード / ワード）
  - データアクセス時のデータサイズを指定します。
- 属性
  - データアクセス時の属性を指定します。
- サイズ無視
  - アクセスデータ長にかかわらず、指定アドレスがアクセスされた場合をブレーク条件にします。
- 比較条件
  - データ比較条件を指定します。
    - 無効           : ブレーク条件にデータを指定しません。
    - データ一致   : 指定データと一致した場合をブレーク条件にします。
    - データ NOT   : 指定データと一致しない場合をブレーク条件にします。
- データ
  - データアクセス時のデータを設定するフィールドです。
- データマスク
  - 指定したデータに対するマスク値を指定するフィールドです。



## &lt; 注意事項 &gt;

FR60Lite または FR80S 使用時以外はこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.4 ブレーク」を参照してください。

## ■ データブレーク詳細 (MB2198)

図 4.6-11 データブレーク詳細ダイアログ



- ブレークアドレス
  - ブレークポイントを設定するアドレスを指定するフィールドです。
- サイズ (バイト / ハーフワード / ワード)
  - データアクセス時のデータサイズを指定します。
- 属性
  - データアクセス時の属性を指定します。
- データ一致
  - データおよびデータマスクを指定するときには、チェックします。
- データ
  - データアクセス時のデータを設定するフィールドです。
- データマスク
  - 指定したデータに対するマスク値を指定するフィールドです。

## &lt; 注意事項 &gt;

FR60Lite または FR80S 使用時以外はこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.4 ブレーク」を参照してください。

## 4.6.5 イベント

ここでは、SOFTUNE Workbench でのイベント設定について説明します。

### ■ イベントの設定

イベントの設定は、図 4.6-12 のダイアログで行うことができます。この機能はエミュレータデバッガでのみ使用可能です。

#### ● コードイベント

図 4.6-12 イベントダイアログ (コード)

イベント

コード データ

イベント番号(N): 1 追加(P)

アドレス(A): H'000801BA No symbol

アドレスマスク(M): H'FFFFFFFF パスカウント(S): D'1

イベントモード

☐ ORモード(O) ☒ シーケンシャルモード(S)

イベントリスト(L)

no.	en/dis	addr	mask	pass	cond	symbol
1	enable	000801BA	FFFFFFFF	1(	0)	
2	enable	000801EC	FFFFFFFF	5(	0)	

有効(E) 無効(D) 削除(T) OK キャンセル

- イベント番号
  - イベント番号 (1 または 2) を指定します。
- アドレス
  - イベント発生条件となるアドレスを指定します。
- アドレスマスク
  - アドレスマスクを指定します。ビットが1であるところのみを比較対象とします。
- パスカウント
  - イベントの発生回数を指定します。指定範囲は 1 ~ 255 です。

- イベントモード
  - DSU3 チップでは、トレースサンプリングモードによって表示が変わります。フルモードのときには有効表示ですが、トリガモードのときにはイベントモードは無効になるため、選択できないように無効表示となります。
  - OR モードは、イベント 1 または、イベント 2 の条件が成立した場合にイベントのトリガとなります。
  - シーケンシャルモードは、イベント 1 イベント 2 の順に条件が成立した場合にイベントのトリガとなります。
- イベントリスト
  - 現在のイベントの設定状態を表示します。

● データイベント

図 4.6-13 イベントダイアログ (データ)

イベント

コード データ

イベント番号(N): 1 [追加(P)]

アドレス(A): H'74AD77FE No symbol

アドレスマスク(M): H'FFFFFFFF

データ(D): H'03 サイズ(S): バイト

データマスク(K): H'FF ☐ データNOT(N)

イベントモード

☒ ORモード(O) ☐ シーケンシャルモード(S)

属性

☒ リード(R) ☒ ライト(W)

イベントリスト(L)

no.	en/dis	status	addr	addr_msk	data	data_msk	size	cond
1	enable	R/W	74AD77FE	-----	00000003	000000FF	byte	
2	enable	R/W	74AD77E2	-----	00000010	000000FF	byte	

[有効(E)] [無効(B)] [削除(T)]

OK キャンセル

- イベント番号
  - イベント番号 (1 または 2) を指定します。
- アドレス
  - イベント発生条件となるアドレスを指定します。
- アドレスマスク
  - アドレスマスクを指定します。ビットが 1 であるところのみを比較対象とします。データ値が有効のときには、無視されます。

- データ
  - イベント発生条件となるデータを指定します。DSU タイプが "DSU2" 以降で有効です。
- データマスク
  - データマスクを指定します。ビットが 1 であるところのみを比較対象とします。DSU タイプが "DSU2" 以降で有効です。
- データ NOT
  - データ値が一致しないときを条件とします。DSU タイプが "DSU2" 以降で有効です。
- サイズ
  - データのアクセスサイズを指定します (バイト / ハーフワード / ワード)。
- 属性
  - データのアクセス属性を指定します (リード / ライト)。
- イベントモード
  - DSU3 以降のチップでは、トレースサンプリングモードによって表示が変わります。フルモードのときには有効表示ですが、トリガモードのときにはイベントモードは無効になる (イベント 2 の条件成立でイベントのトリガとなる) ため、選択できないように無効表示となります。
  - OR モードは、イベント 1 または、イベント 2 の条件が成立した場合にイベントのトリガとなります。
  - シーケンシャルモードは、イベント 1 イベント 2 の順に条件が成立した場合にイベントのトリガとなります。
- イベントリスト
  - 現在のイベントの設定状態を表示します。

---

### < 注意事項 >

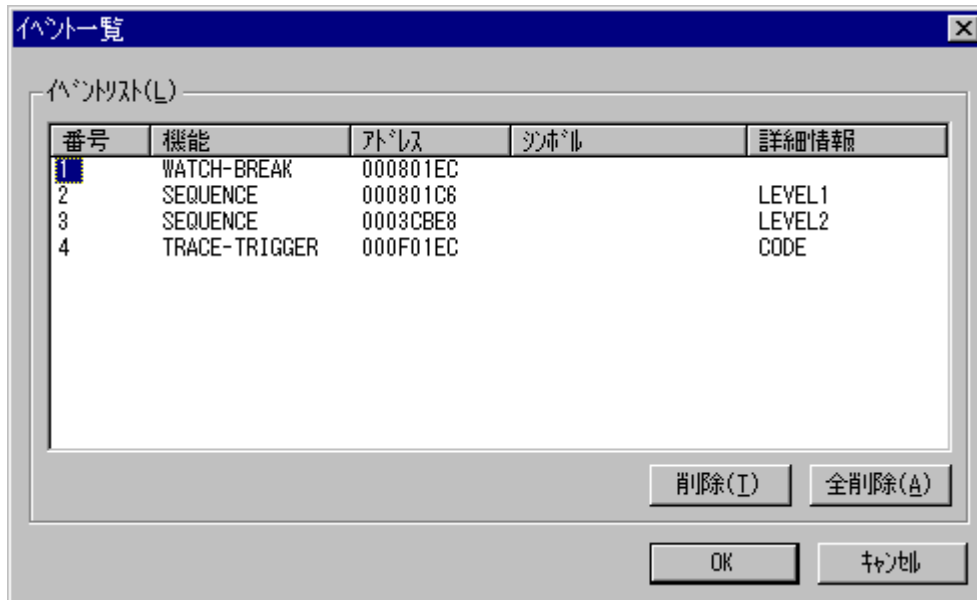
- FR60Lite 使用時はシーケンサまたはトレーストリガが機能強化されているため、この機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.5 シーケンサによる制御」または「2.3.7 トレース」を参照してください。
  - コードイベントのヒットカウントはユーザプログラム実行中には更新されません。したがってユーザプログラム実行中に表示されるヒットカウントはプログラム実行開始前の値となります。
-

## ■ イベント一覧

現在設定されているイベント一覧を表示します。この機能は FR60Lite または FR80S で使用可能です。

### ● イベント一覧

図 4.6-14 イベント一覧ダイアログ



- イベントリスト  
現在設定されているイベントを表示します。
- [ 削除 ] ボタン  
イベントリスト中で選択されているイベントを削除します。
- [ 全削除 ] ボタン  
イベントリスト中の全イベントを削除します。

## 4.6.6 シーケンス

ここでは、シーケンスを設定する各ダイアログについて説明します。

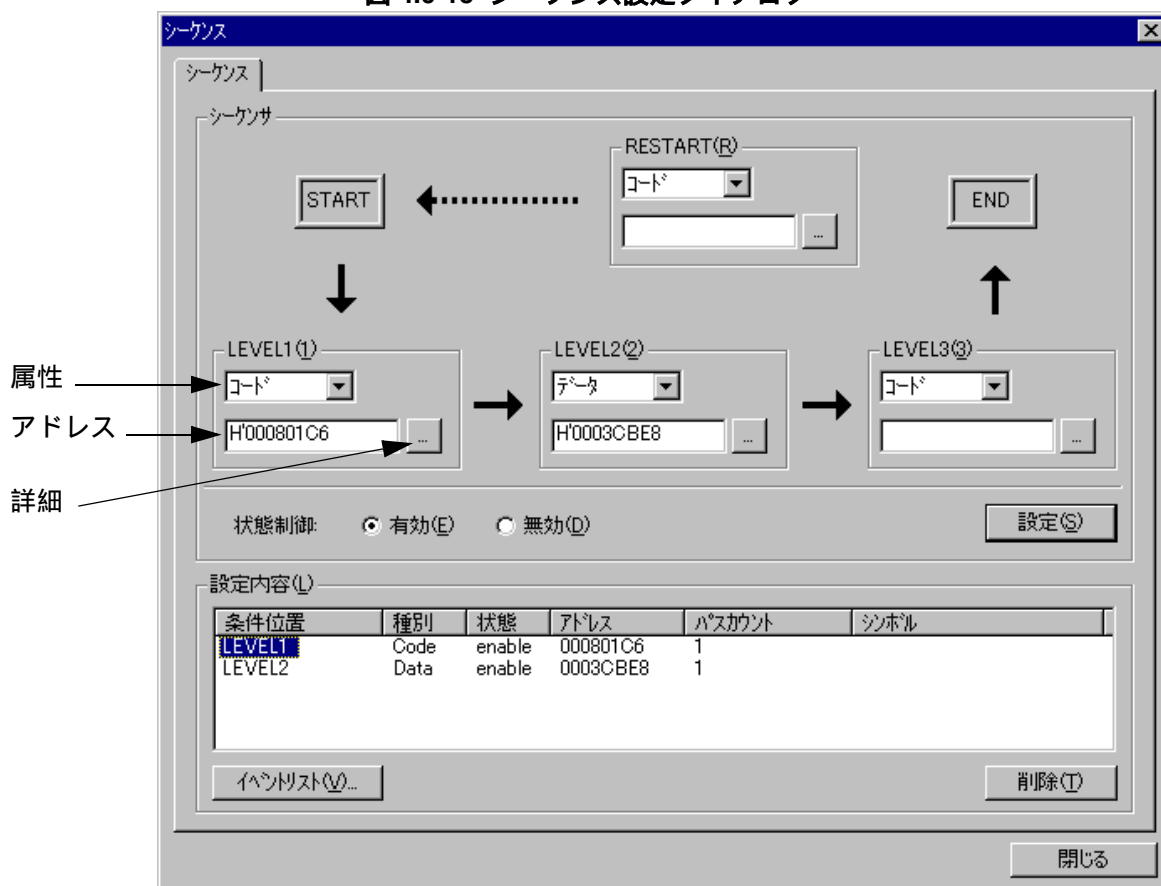
### ■ シーケンス

シーケンスの設定状態を表示するダイアログを開きます。

この機能は FR60Lite または FR80S で使用可能です。

### ■ シーケンス設定

図 4.6-15 シーケンス設定ダイアログ



### ● シーケンサ

シーケンサの遷移条件を設定します。LEVEL1 から LEVEL3 までの最大 3 レベルの設定が可能です。また、シーケンサの開始状態に戻す RESTART の設定が可能です。設定は必ず LEVEL1 から順番におこないます。正しい設定がおこなわれていない、または、設定を削除した場合はそれ以降の設定は全て削除されます。

属性：遷移条件の属性を指定します。コードまたはデータから選択します。ただし FR80S ではデータのみとなります。

アドレス：遷移条件を設定するアドレスまたはシンボルを指定します。

詳細：遷移条件の詳細設定をおこないます。図 4.6-16 のダイアログが表示されます。

- 状態制御

設定したシーケンサの有効、無効を設定します。

- [ 設定 ] ボタン

指定されたシーケンサの遷移条件を設定します。

- 設定内容

現在設定されている、シーケンサの遷移条件を表示します。

- [ イベントリスト ] ボタン

全イベントの設定状況を確認するためのイベント一覧ダイアログを表示します。

- [ 削除 ] ボタン

設定内容を削除します。

## ■ シーケンス詳細設定

図 4.6-16 シーケンス詳細設定ダイアログ

シーケンス詳細

LEVEL(Y): LEVEL1

アドレス(A): H'0003C000

パスカウント(P): D'1

アドレスマスク(K): H'FFFFFFFF

サイズ(S): バイト

属性

- ☒ リード(R)
- ☐ ライト(W)
- ☐ サイズ無視(Z)

比較条件

- ☐ 無効(I)
- ☒ データ一致(Q)
- ☐ データNOT(N)

データ(D): H'00000008

データマスク(M): H'FFFFFFFF

OK キャンセル

- LEVEL

遷移条件を設定する LEVEL を設定します。

- アドレス

遷移条件を設定するアドレスまたはシンボルを指定します。

- パスカウント

トリガヒット時のアクセス回数を指定します。

- アドレスマスク

指定したアドレスに対するマスク値を指定するフィールドです。

## 第 4 章 メニュー

- サイズ (バイト / ハーフワード / ワード)

データアクセス時のデータサイズを指定します。

- 属性

データアクセス時の属性を指定します。ただし FR80S では指定できず、必ずライト属性となります。

- サイズ無視

アクセスデータ長にかかわらず、指定アドレスがアクセスされた場合をシーケンサの遷移条件にします。

- 比較条件

データ比較条件を指定します。

無効 : 遷移条件にデータを指定しません。

データ一致 : 指定データと一致した場合を遷移条件にします。

データ NOT : 指定データと一致しない場合を遷移条件にします。

- データ

データアクセス時のデータを設定するフィールドです。

- データマスク

指定したデータに対するマスク値を指定するフィールドです。

---

### < 注意事項 >

FR60Lite または FR80S 使用时以外ではこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.5 シーケンサの制御」を参照してください。

---



## 4.6.7 コールスタック

ここでは、SOFTUNE Workbench でのコールスタックについて説明します。

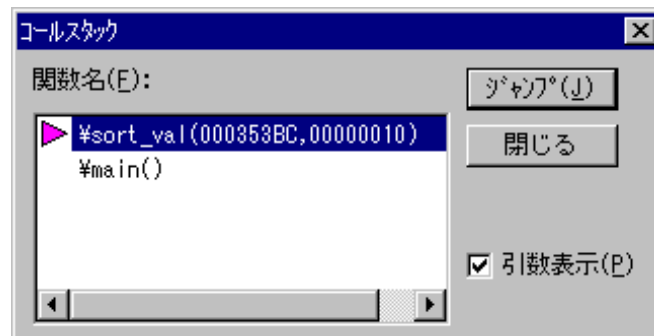
### ■ コールスタック

通常、プログラムはいくつかのサブルーチンの集合になっています。そのため、デバッグを進めていくと、あるルーチンから別のルーチンが呼び出され、さらにそこからまた別のルーチンが呼び出されるというように、何段にもなる関数の呼び出しが発生します。

コールスタックには、この呼び出しの関係が保持されています。ここにリストされている関数名をダブルクリックする。またはクリックしてジャンプボタンを押すことにより、その関数をすぐにソースウィンドウに表示して見ることができます。

その際、関数名の横には三角形のシンボルマークが表示されます。

図 4.6-17 コールスタックダイアログ



関数名リストの最後（最下行）に記述された関数が親になります。そこから、その上にある関数が呼び出され、さらにそこからその上の方にある関数が呼び出されています。このようにして、最上段に記述されている関数が現在の PC のある関数です。

リターンすると上の方の関数から順にリストから消されていきます。

#### ● 引数表示

引数表示にチェックマークがついている場合は、図 4.6-17 に示したように関数名の後に引数の値を表示します。

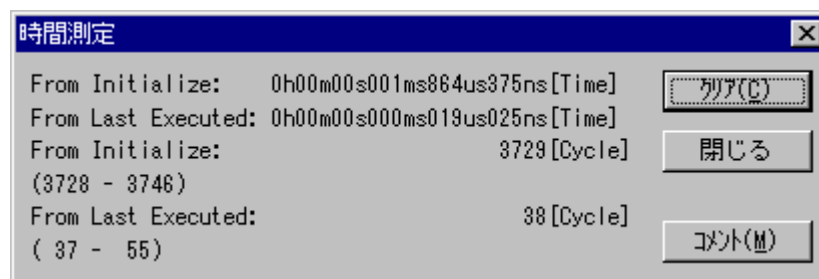
チェックマークがついていない場合は、関数名の後ろに「[()]」のみが表示されます。

## 4.6.8 時間測定

ここでは、SOFTUNE Workbench での時間測定について説明します。

### ■ 時間測定

図 4.6-18 時間測定ダイアログ



#### ● 表示される項目

	時間	サイクル数	ステップ数
SIM	×		
EML (MB2197)	×		×
EML (MB2198)			×
MON		×	×

#### ● From Initialize

[ クリア ] ボタンを押してからの実行累計を示します。

#### ● From Last Executed

直前の実行による実行時間を示します。

#### < 注意事項 >

測定結果には誤差があります。詳細は『SOFTUNE Workbench 機能説明書』「2.2.5 実行サイクル数測定」「2.3.6 実行時間測定」「2.4.3 実行時間測定」を参照してください。

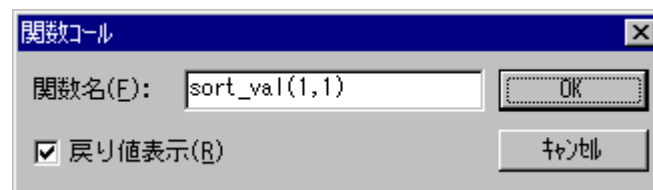
## 4.6.9 関数コール

ここでは、SOFTUNE Workbench の関数コール機能について説明します。

### ■ 関数コール

デバッグ中にプログラムの流れとは無関係に、指定した関数を起動することができます。この機能を関数コールと呼びます。

図 4.6-19 関数コールダイアログ



関数コールのダイアログ (図 4.6-19) が表示されたら、呼び出したい関数に正しい引数を付けて指定してください。

C/C++ 言語の引数に準拠しますが、次の引数は指定できません。

- 構造体 (ストラクチャ)
- 共用体 (ユニオン)
- クラス

図 4.6-20 関数呼び出し結果ダイアログ



呼び出した関数の処理を終了してリターンしたら、関数からのリターン値を表示するダイアログが表示されます (図 4.6-20)。

#### 【例】

関数定義が、`int sub (int param);` であるときは、以下のように指定します。

- `sub (10)` : 直接、定数値を指定するとき
- `sub (ii)` : 変数 `ii` を指定するとき

## ■ 機能

指定した関数を実行し、戻り値を表示します。戻り値は組込み変数 %CALL に設定されます。指定された関数の引数は、仮引数の型で評価して実行します。指定された実引数の個数が仮引数の個数より多い場合、多い分の実引数は int 型で評価します。

関数コールによる実行中、ブレークポイント等にヒットすると、その位置でブレークします。その後、連続実行で再実行させれば、関数コールの実行を継続します。

関数コールの実行を中止する場合は、メインメニュー [ デバッグ ] - [ コールクリア ] コマンドを使用します。

なお、関数コールはネストすることができません。

関数コールは、現在の PC にブレークポイントを設定し、そこへリターンするようにリターンアドレスをセットしたうえで、関数を呼び出します。従って、関数コールで呼び出された関数が現在の PC を通過した場合、関数の途中でブレークしてしまいます。

この場合は、次のメッセージが表示されます。

Break at アドレス by Invalid call termination

連続実行で実行を再開させれば関数コールの実行を継続します。

---

### < 注意事項 >

関数コールにより、レジスタ、メモリ、IO などのリソースは関数呼び出し前の状態から変化する可能性があります。

レジスタに関しては関数呼び出し前の内容を保持し、関数実行後またはコールクリアにより復元します。その他のリソースに関しては復元しません。

---

## 4.6.10 コールクリア

---

ここでは、SOFTUNE Workbench のコールクリア機能について説明します。

---

### ■ コールクリア

関数コール（「4.6.9 関数コール」参照）で、ブレークポイントなどにより実行停止した後、その関数コールを最後まで実行しないで元の状態に戻したいときに使用します。

コールクリアを行うと、直ちに呼び出した関数から復帰します。この場合は呼び出した関数が最後まで実行されていないので、関数呼び出し結果の表示は行われません。

## 4.6.11 ベクタ

ここでは、SOFTUNE Workbench のベクタの表示と変更のしかたについて説明します。

### ■ ベクタ

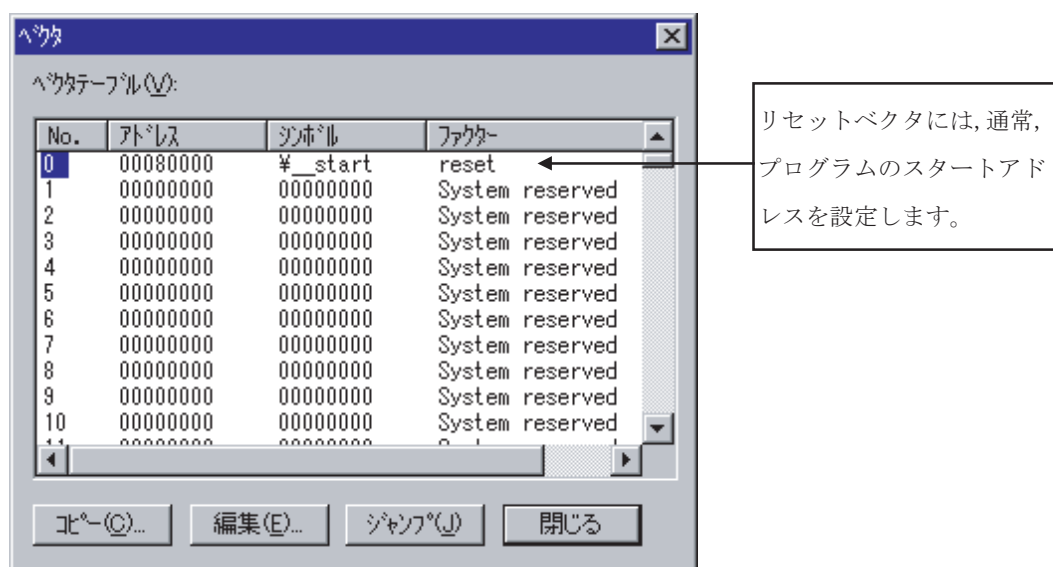
MCU にリセットがかかった場合や、さまざまな要因により割込み処理の要求が発生したとき、MCU はその割込みの種類に応じてあらかじめ決められているアドレスに設定されているデータを割込み処理ルーチンのアドレスとして PC に設定します。この割込み処理ルーチンのアドレスを設定しているアドレスをベクタといいます。ベクタは MCU の品種によってあらかじめ決められています。

### ■ ベクタの表示と設定

#### ● 表示

ベクタ表示ダイアログを 図 4.6-21 に示します。

図 4.6-21 ベクタ表示ダイアログ

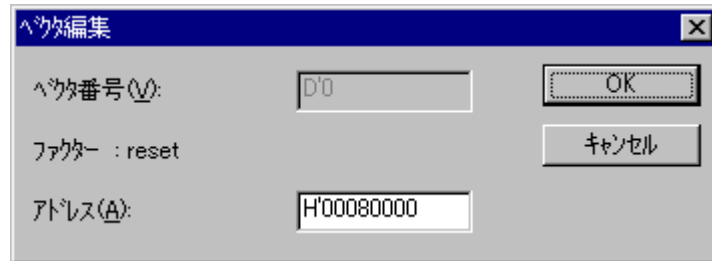


## ● アドレスの設定

ベクタに設定されているアドレスを変更する場合は、以下の手順で行ってください。

- 1) ベクタテーブルの番号を選択して、[編集] ボタンをクリックしてください。
  - ベクタ編集ダイアログ (図 4.6-22) が表示されます。
- 2) アドレスを設定して、[OK] ボタンをクリックしてください。

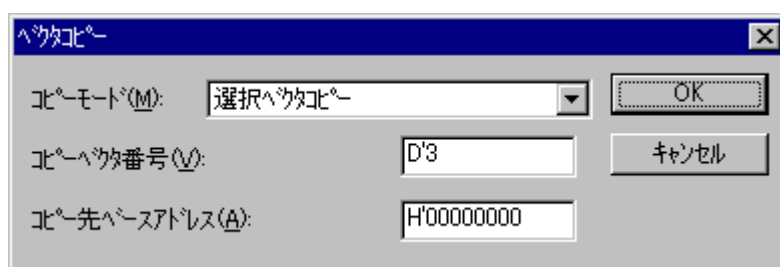
図 4.6-22 ベクタ編集ダイアログ



## ■ コピー

- 1) ベクタテーブルの番号を選択して、[コピー] ボタンをクリックしてください。
  - ベクタコピーダイアログ (図 4.6-23) が表示されます。
- 2) [コピーモード] を選択してください (すべてのベクタコピー / デバッガ使用ベクタコピー / 選択ベクタコピー)。
- 3) [コピーベクタ番号] を設定してください。
- 4) [コピー先ベースアドレス] を設定してください。
- 5) [OK] ボタンをクリックしてください。

図 4.6-23 ベクタコピーダイアログ



## ■ ジャンプ

ベクタテーブルに設定されているアドレスに配置されているプログラムのソースを表示します。以下の手順で実行してください。

- 1) ベクタ番号を選択してください。
- 2) [ジャンプ] ボタンをクリックしてください。

ベクタテーブルに正しい値 (プログラムの開始アドレス) が設定されていないと、ソース表示ができませんので、逆アセンブル表示となります。

---

### < 注意事項 >

ジャンプ機能は、単に、飛び先のプログラムを表示するだけです。プログラムカウンタを変更してプログラムの制御をベクタテーブルに書かれたアドレスに移すことはしません。

---



## 4.6.12 ターゲットファイルのロード

ここでは、SOFTUNE Workbench でのデバッグ対象ファイルのロードについて説明します。

### ■ ターゲットファイル

デバッグの対象となるのは、ABS 形式のターゲットファイルです。プロジェクトのターゲットファイルとして登録されています。

SOFTUNE Workbench で、ソースプログラムの作成、コンパイル/アセンブル、リンクを実行して、コンパイル/アセンブル、リンクのエラーがなくなり、ABS 形式のファイルが作成できたら、デバッグを始めることができます。

### ■ ターゲットファイルのロード

ターゲットファイルをロードする前に、[デバッグ]メニューから[デバッグの開始]を選択して、SOFTUNE Workbench をデバッグセッションにしてください。デバッグセッションになったら、[デバッグ]メニューから[ターゲットファイルのロード]を選択して、ターゲットファイルを読み込んでください。ターゲットファイル読み込み中は、読み込みの進捗状況を表示するダイアログ(図 4.6-24)が表示されます。

図 4.6-24 ターゲットファイル読み込みの進捗表示



読み込みが終了すると、PC にエントリポイントが設定されて、エントリポイントを含むモジュールのソース行表示が行われた状態で停止します。

これ以降、ステップ実行や連続実行などを行い、デバッグを進めてください。

## 4.6.13 デバッグの開始 / デバッグの終了

---

デバッグセッションの開始と終了を行います。

---

### ■ デバッグの開始

SOFTUNE Workbench をデバッグセッションにします。これ以降デバッグに関するコマンドが使用可能になります。デバッグセッションに移行したら、はじめにターゲットファイルのロード（「4.6.12 ターゲットファイルのロード」を参照）を実行してください。

### ■ デバッグの終了

SOFTUNE Workbench のデバッグセッションを終了します。

## 4.7 環境

---

SOFTUNE Workbench の実行環境の設定 , デバッガモードでの環境設定 , その他のツールの設定などを行います。

---

### ■ SOFTUNE Workbench の環境設定

開発環境

### ■ デバッガの環境設定

- デバッグ環境の設定
- デバッガのメモリマップ

### ■ その他の設定

- ツールの設定
- キーボードの設定
- エディタの設定
- エラージャンプの設定
- ツールの起動

## 4.7.1 開発環境

SOFTUNE Workbench の動作の設定と、コンパイラなどの言語系ツールが必要とする環境変数の設定を行います。

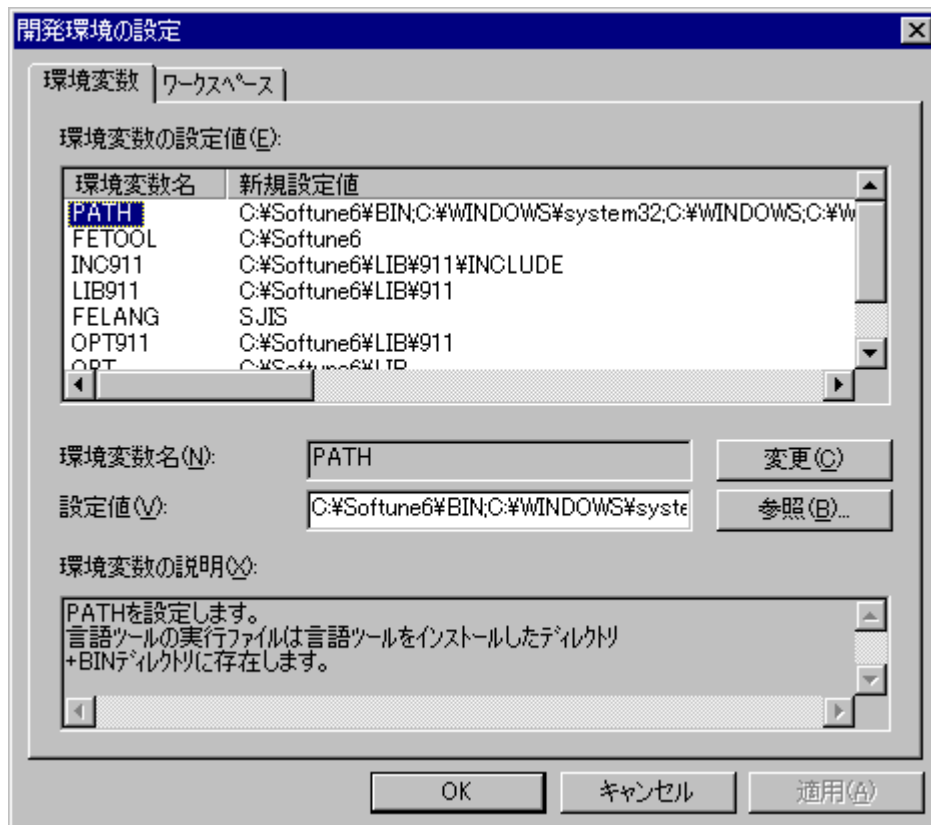
### ■ 環境変数

図 4.7-1 に環境変数設定のダイアログを示します。ここでは、表 4.7-1 に示した環境変数の設定を行います。

表 4.7-1 環境変数名一覧

環境変数名	説 明
FETOOL	言語ツールをインストールした基準のディレクトリ
INC911	インクルードファイルが存在するディレクトリ
LIB911	ライブラリファイルが存在するディレクトリ
OPT911	言語ツールのデフォルトオプションファイルが存在するディレクトリ
FELANG	言語ツールの出力するメッセージの文字コード系切り替え (SJIS：日本語 (shift JIS), ASCII：英語)
PATH	言語ツールの存在するディレクトリ
TMP	作業用一時ファイルを作成するディレクトリ

図 4.7-1 開発環境の設定 (環境変数) ダイアログ



設定は以下の手順で行ってください。

- 1) [ 環境変数の設定値 ] の一覧より、設定変更したい環境変数名を選択してください。
  - 現在の設定値が、[ 設定値 ] フィールドに表示されます。
  - その環境変数についての簡単な説明が、[ 環境変数の説明 ] に表示されます。
- 2) [ 設定値 ] フィールドの内容を変更してください。
- 3) [ 変更 ] ボタンをクリックしてください。

## ■ ワークスペース

SOFTUNE Workbench の以下の動作の設定を行います。

- 起動時に前回のワークスペースを開く

この項目をチェックすると、SOFTUNE Workbench を起動したときに、前回開いていたワークスペースファイルを自動的に開くようになります。

- コンパイル/アセンブル時にオプションを表示する

この項目をチェックすると、C/C++ コンパイラ/アセンブラを起動するときに指定するオプションを、アウトプットウィンドウに表示するようになります。

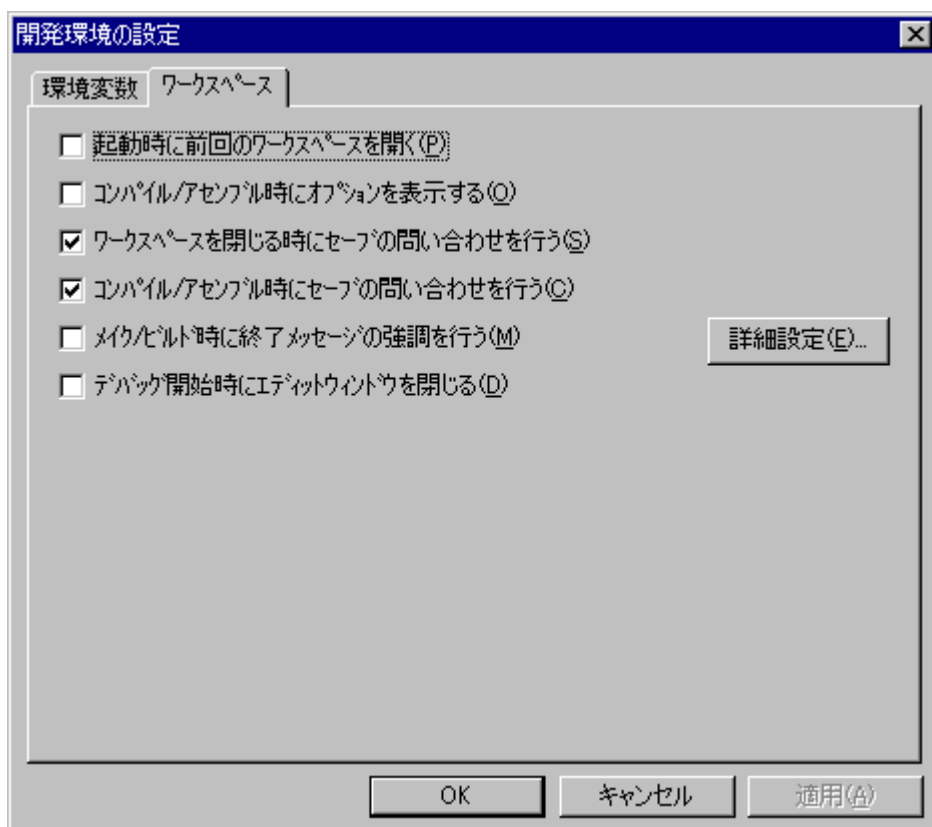
- ワークスペースを閉じるときにセーブの問い合わせを行う

この項目をチェックすると、ワークスペースを閉じるときに、現在開いているワークスペースをワークスペースファイルに上書き保存するかどうか、問い合わせるようになります。

- コンパイル/アセンブル時にセーブの問い合わせを行う

この項目をチェックすると、コンパイル/アセンブルする前に現在編集集中のファイルを上書き保存するかどうか問い合わせるようになります。

図 4.7-2 開発環境の設定 (ワークスペース) のダイアログ



- メイク/ビルド時に終了メッセージの強調を行う

この項目をチェックすると、コンパイル/アセンブル/メイク/ビルド時の終了メッセージ(中断、正常、警告、エラー、致命的エラー、起動失敗)の表示色を変更することができます。

表示色を変更したい場合は、このフィールドの右横にある[詳細設定]ボタンをクリックしてください。終了メッセージの表示色ダイアログ(図4.7-3)が表示されますので、表示色を変更してください。

図 4.7-3 終了メッセージの表示色



- デバッグ開始時にエディットウィンドウを閉じる

この項目をチェックすると、デバッガを起動したときに開いているエディットウィンドウを閉じるようになります。

## 4.7.2 デバッグ環境の設定

---

デバッグ環境の設定を行います。これはデバッグセッションのときのみ有効になります。

---

### ■ 設定する項目

サブメニューから以下の項目を選択して設定します。

- I/O ポート
- 割込み
- デバッグ環境設定
- デバッグ機能の選択
- セットアップウィザード

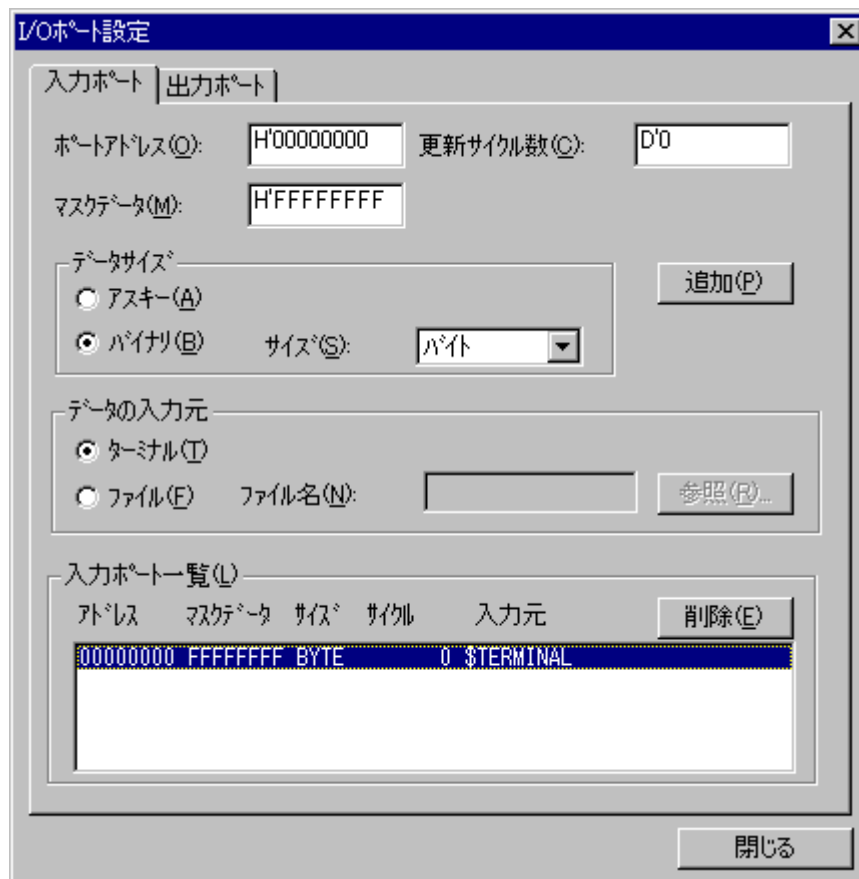


### 4.7.2.1 I/O ポート

ここでは、I/O ポートの設定手順について説明します。

#### ■ 入力ポートの設定

図 4.7-4 入力ポート設定ダイアログ



- ポートアドレス  
ポートアドレスを指定します。
- マスクデータ  
アドレスマスクを指定します。ビットが1であるところのみを比較対象とします。
- データサイズ  
データの入力種別を選択します。バイナリを選択した場合、サイズを選択します。(バイト/ハーフワード/ワード)

● データの入力元

ポートデータの入力元を指定します。

• 入力ターミナル

- 入力ポート設定で、[ データの入力元 ] を [ ターミナル ] に設定した場合、実行中に入力要求があると、入力ターミナルダイアログが表示されます。
- 入力ポート設定で、データタイプが [ アスキー ] 指定の場合は、アスキー入力が可能です。入力ポート設定で、データタイプが [ バイナリ ] 指定の場合は、バイナリ入力が可能です。

図 4.7-5 入力ターミナル



● 入力ポート一覧

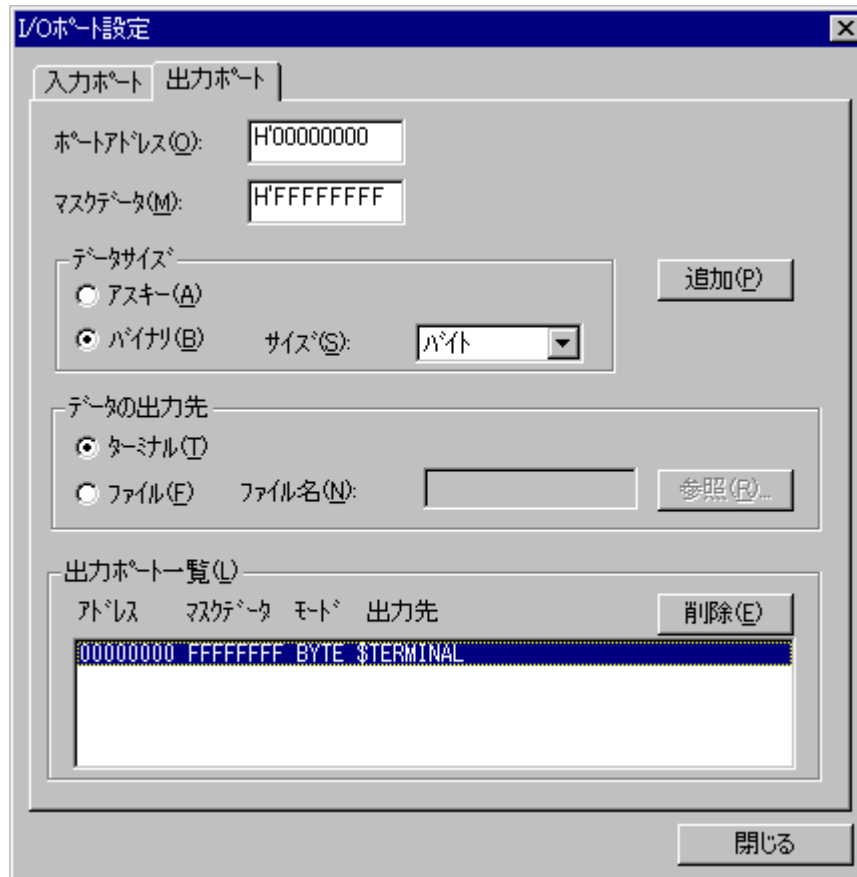
現在指定されているポートを表示します。

■ 入力ポートの設定解除

- 1) [ 入力ポート一覧 ] から解除したい設定を選択してください。
- 2) [ 削除 ] ボタンをクリックしてください。
- 3) すべての設定解除が完了したら、[ 閉じる ] ボタンをクリックしてください。

## ■ 出力ポートの設定

図 4.7-6 出力ポート設定ダイアログ



- ポートアドレス

ポートアドレスを指定します。

- マスクデータ

アドレスマスクを指定します。ビットが1であるところのみを比較対象とします。

- データサイズ

データの出力種別を選択します。バイナリを選択した場合、サイズを選択します。(バイト/ハーフワード/ワード)

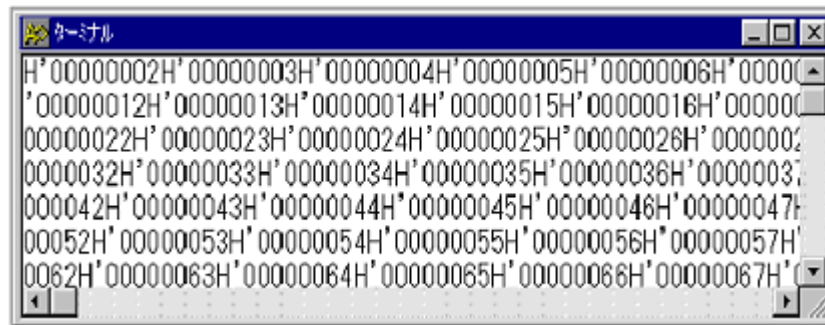
● データの出力先

ポートデータの出力先を指定します。

• 出力ターミナル

- 出力ポート設定で,[データの出力先]を[ターミナル]に設定した場合,実行中に出力要求があると,ターミナルウィンドウが表示されます。また,データタイプにより,出力内容が変わります。

図 4.7-7 出力ターミナル (バイナリ)



● 出力ポート一覧

現在指定されているポートを表示します。

■ 出力ポートの設定解除

- 1) [出力ポート一覧] から解除したい設定を選択してください。
- 2) [削除] ボタンをクリックしてください。
- 3) すべての設定解除が完了したら,[閉じる] ボタンをクリックしてください。

## 4.7.2.2 割り込み

ここでは、割り込みの設定の手順について説明します。

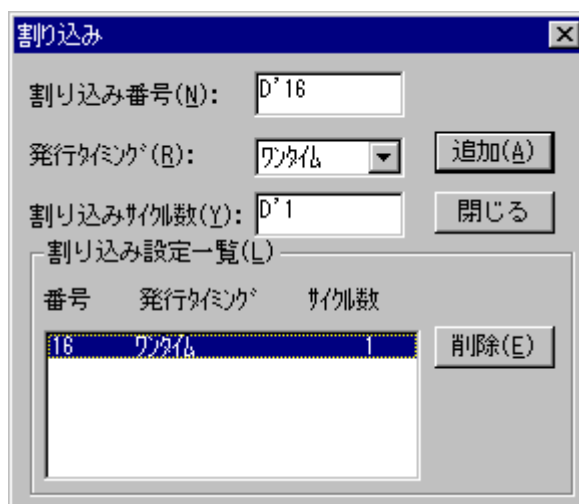
### ■ 割り込み設定手順

- 1) 図 4.7-8 のダイアログが開いたら、[ 割り込み番号 ] を設定してください。
- 2) [ 発行タイミング ] を選択してください。
  - [ ワンタイム ] と [ インターバル ] が選択できます。
- 3) [ 割り込みサイクル数 ] を設定してください。
- 4) [ 追加 ] ボタンをクリックしてください。
  - 割り込み番号、発行タイミング、割り込みサイクル数が [ 割り込み設定一覧 ] に表示されます。
- 5) すべての設定が完了したら、[ 閉じる ] ボタンをクリックしてください。

### ■ 割り込みの設定解除

- 1) 図 4.7-8 のダイアログが開いたら、[ 割り込み番号 ] を設定してください。
- 2) [ 割り込み設定一覧 ] から解除したい設定を選択してください。
- 3) [ 削除 ] ボタンをクリックしてください。
- 4) すべての設定解除が完了したら、[ 閉じる ] ボタンをクリックしてください。

図 4.7-8 割り込み設定ダイアログ



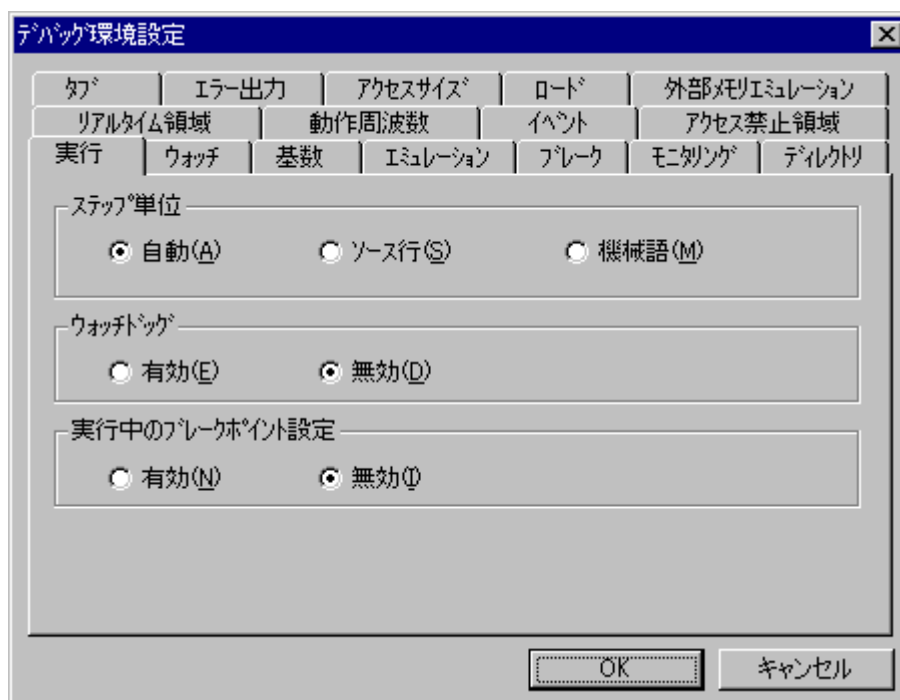
### 4.7.2.3 デバッグ環境設定

ここでは、デバッグ環境の設定の手順について説明します。

#### ■ デバッグ環境設定手順

- 実行

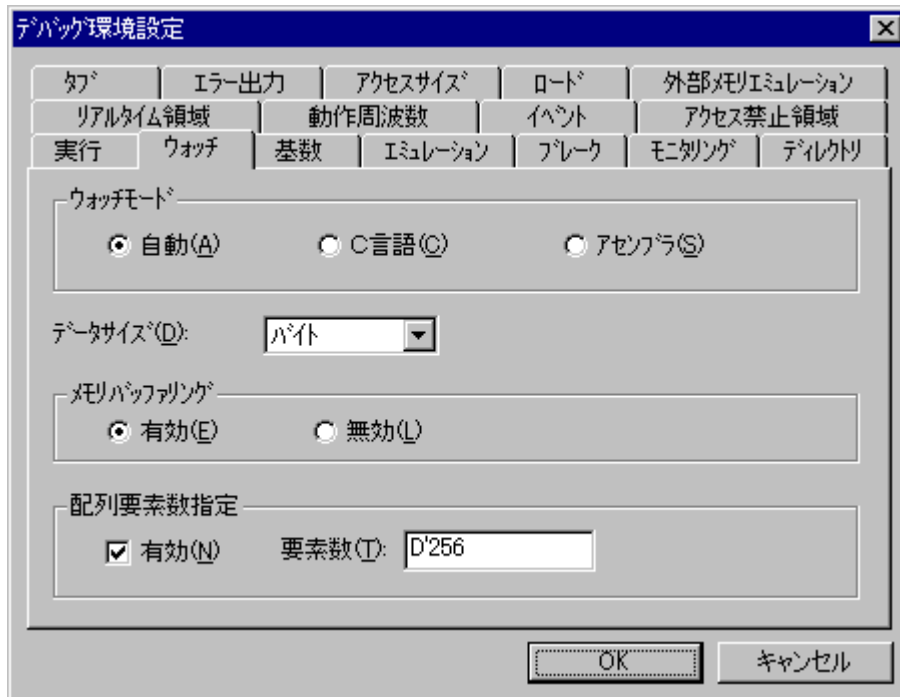
図 4.7-9 デバッグ環境設定 (実行)



- ステップ単位：ステップ実行時のステップ単位を指定します。
  - 自動：ウィンドウの表示状態に応じて自動的に設定します。
  - ソース行：ソース行単位にステップします。
  - 機械語：機械語単位にステップします。
- ウォッチドッグ
  - プログラム実行時に、ウォッチドッグタイマを有効にするか無効にするかを指定します。
- 実行中のブレークポイント設定
  - 実行中のブレークポイント設定を有効にすると、ユーザプログラム実行中にブレークポイントの設定が可能になります。
  - MB2198にてDSU4タイプの評価用チップを使用する場合にのみ有効となります。

## ● ウォッチ

図 4.7-10 デバッグ環境設定 (ウォッチ)



- ウォッチモード
  - 自動 : 解析結果に応じて自動的に設定します。
  - C 言語 : C/C++ 言語式として解釈します。
  - アセンブラ : アセンブラ式として解釈します。
- データサイズ : アセンブラモード時の表示サイズを設定します。
  - バイト / ハーフワード / ワード
- メモリバッファリング
  - 有効 : 配列や構造体など変数の場合 , 変数全体のメモリをまとめて読み出します。アクセスサイズは、先頭の変数サイズとなります。
  - 無効 : 配列や構造体などの変数の場合 , 各要素 , 各メンバ単位にメモリを読み出します。

デフォルトは、有効となっています。

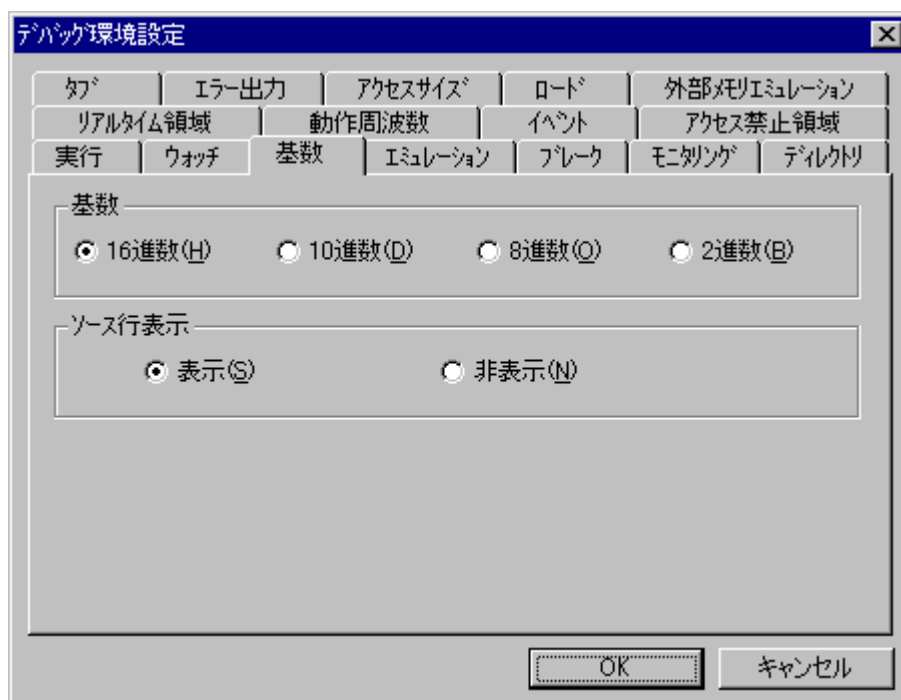
無効にするとウォッチウィンドウ , ローカルウィンドウの表示が遅くなる場合があります。
- 配列要素数指定
  - 有効 : 配列をウォッチ変数として登録 , 展開するときに , 配列が指定した要素数より大きい場合 , 警告ダイアログを表示します。
  - 要素数 : 配列要素数 ( デフォルトは D'256 ) を指定する。

## &lt; 注意事項 &gt;

メモリバッファリングを有効にした場合 , 固定サイズでのリードを要求する I/O などの値は正しく表示できません。

● 基数

図 4.7-11 デバッグ環境設定 ( 基数 )



- 基数
  - 数値を表示 / 解析する基数を設定します。
- ソース行表示
  - ソース行の表示 / 非表示を切り換えます。



● エミュレーション (エミュレータデバッガのみ有効)

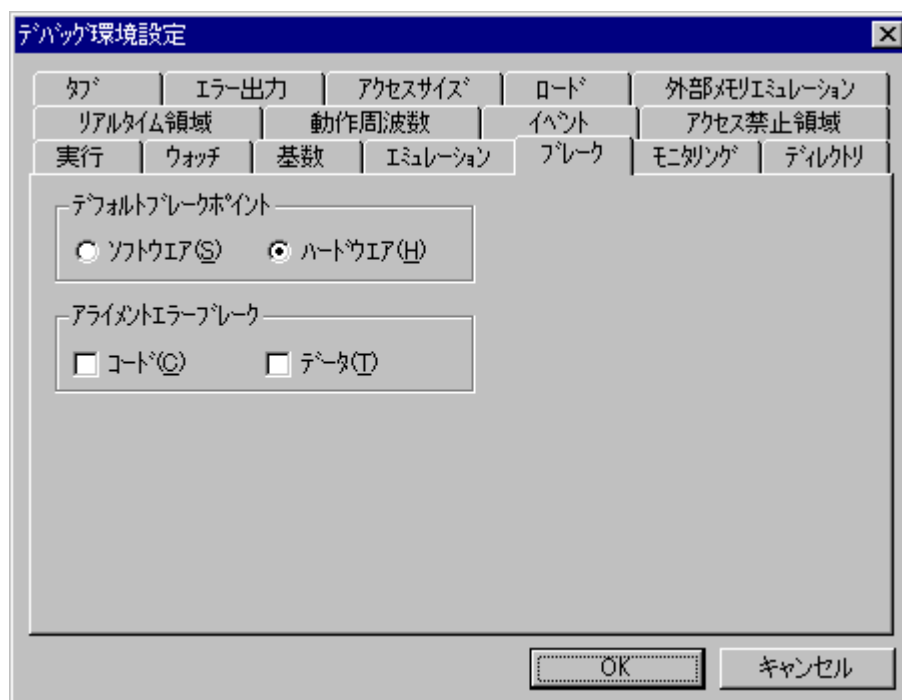
図 4.7-12 デバッグ環境設定 (エミュレーション)



- メモリのペリファイ動作
  - コマンドでメモリ書込みを行ったときにペリファイをするかどうかを選択します。
- TRIG 入力
  - TRIG 端子の入力の有効 / 無効を選択します。
- MCU 動作モード
  - MCU の動作モードを選択します。未実装のモードは表示されません。
- 命令キャッシュの自動フラッシュ
  - 命令キャッシュを自動的にフラッシュするかを設定します。
  - 命令キャッシュが実装されていない場合には、表示されません。
- データキャッシュの自動フラッシュ
  - データキャッシュを自動的にフラッシュするかを設定します。
  - データキャッシュが実装されていない場合には、表示されません。

- ブレーク (エミュレータデバッグのみ有効)

図 4.7-13 デバッグ環境設定 (ブレーク)



- デフォルトブレークポイント
  - コードブレークポイントのデフォルト種別を選択します。
- アライメントエラーブレーク
  - アライメント違反が発生したときに MCU の実行を中断するかどうかを選択します。

## ● RFCR【DSU2】

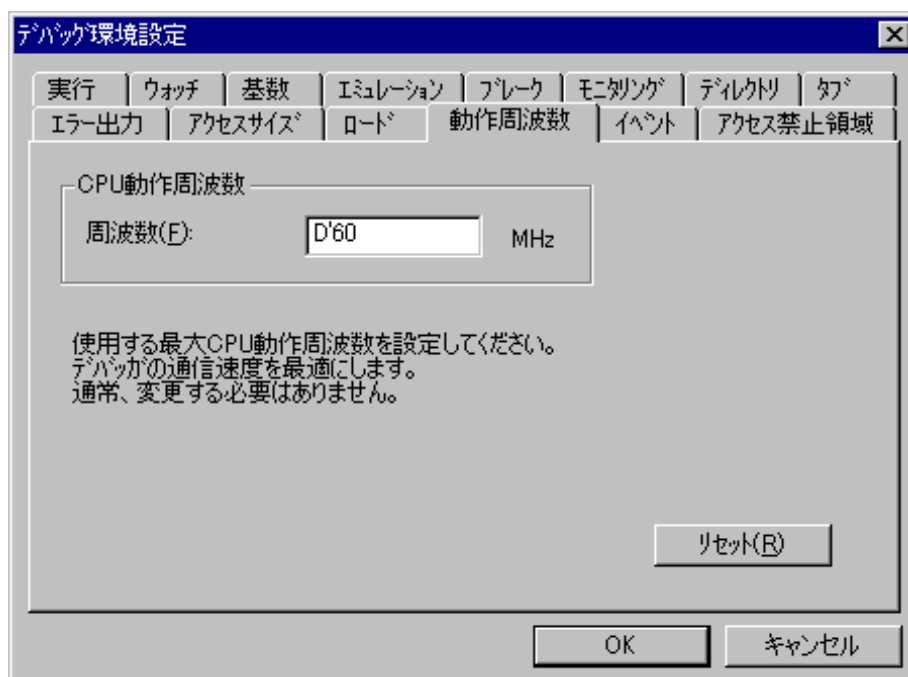
図 4.7-14 デバッグ環境設定 (RFCR)



- RFCR レジスタの自動制御
  - RFCR レジスタの値を自動的に設定するかどうかを指定します。
- 実行時設定値
  - 実行中に RFCR レジスタに設定する値を指定します。
- ブレーク時設定値
  - ブレーク中に RFCR レジスタに設定する値を指定します。

● 動作周波数【DSU3/DSU4】

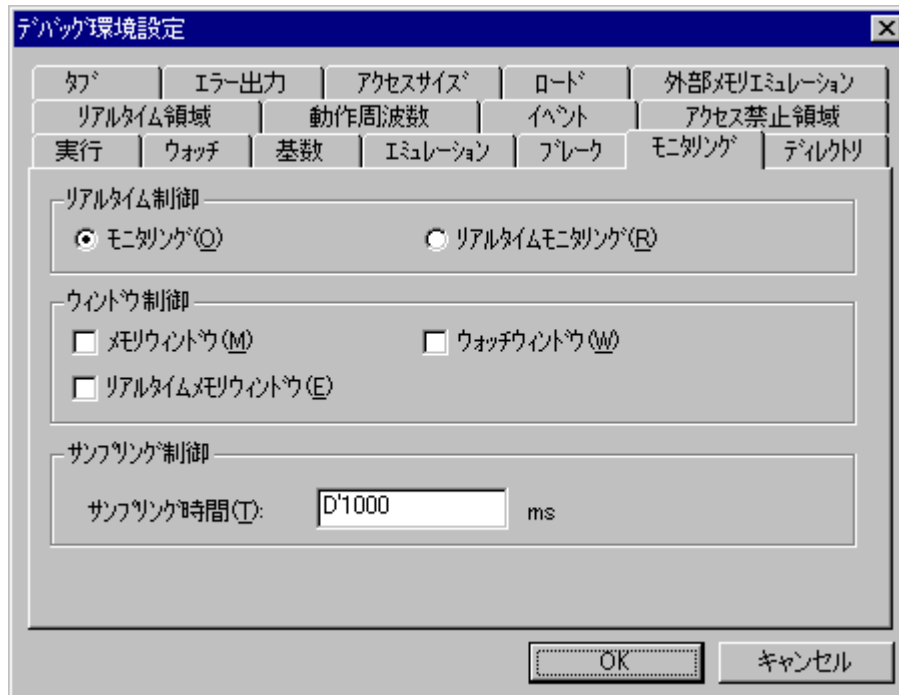
図 4.7-15 デバッグ環境設定 (動作周波数)



- 周波数
  - CPU の最大動作周波数を設定します。  
この設定は最大動作周波数を設定するものであり、実際の動作周波数は変更されません。

## ● モニタリング

図 4.7-16 デバッグ環境設定 (モニタリング)



- リアルタイム制御
  - リアルタイム制御を設定します。
  - モニタリング
    - 擬似オンザフライによるモニタリングを行います。
  - リアルタイムモニタリング
    - 擬似オンザフライによるモニタリングを抑止し、ミラーメモリのみモニタリングします。
  - リアルタイムモニタリングが選択されると、メモリウィンドウのモニタリングは行いません。また、ウォッチウィンドウはリアルタイム領域のみモニタリングを行います。
- ウィンドウ制御
  - メモリウィンドウ
    - メモリウィンドウをモニタリングするかを選択します。
  - ウォッチウィンドウ
    - ウォッチウィンドウをモニタリングするかを選択します。
  - リアルタイムメモリウィンドウ
    - リアルタイムメモリウィンドウをモニタリングするかを選択します。
- サンプリング制御
  - サンプリング時間
    - サンプリング時間を指定します。
    - MB2197 の場合           : 最小 1000ms
    - MB2198 DSU4 の場合 : 最小 100ms

< 注意事項 >

1. リアルタイムモニタリング機能はエミュレータまたはその接続形態により使用できない場合があります。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.9 リアルタイムモニタ」を参照してください。
2. MB2198 DSU3 タイプの評価チップを用いてデバッグするとき、本タブ上の機能は無効になります。
3. MB2197 および MB2198 DSU4 の擬似オンザフライによるモニタリングは、MCU を一度停止させてからメモリの内容を読み出し、再度実行を行っています。

● リアルタイム領域【MB2198】

図 4.7-17 デバッグ環境設定 (リアルタイム領域)



- 領域番号  
領域番号を指定します。(1 または 2)
- 先頭アドレス  
リアルタイム領域の先頭アドレスを指定します。下位 8 ビットはマスクされます。
- メモリコピー  
ウィンドウのオープン時に実メモリの内容を表示します。

< 注意事項 >

FR80S 使用時に内蔵 RAM 領域以外に設定した場合、リアルタイムメモリウィンドウにてデータが更新されません。

## ● アクセス禁止領域

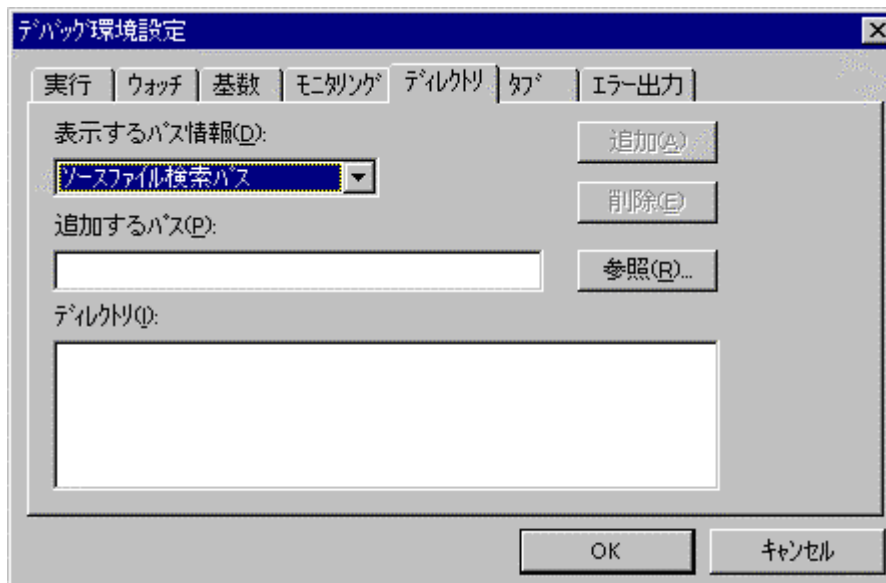
図 4.7-18 デバッグ環境設定 (アクセス禁止領域)

デバッガのメモリへのアクセスを抑止する機能です。最大 16 領域まで設定可能です (1 バイト単位で指定可能)。

- 開始アドレス  
設定する開始アドレスを指定します。
- 終了アドレス  
設定する終了アドレスを指定します。
- 一覧  
現在設定されている領域を表示します。  
チェックを外した領域は、禁止領域の設定が無効になります。

● ディレクトリ

図 4.7-19 デバッグ環境設定 (ディレクトリ)



- 表示するパス情報
  - パスを設定する項目を選択します。
- 追加するパス
  - 追加するパスを設定します。
- ディレクトリ
  - 現在設定されている内容を表示します。

ディレクトリの設定手順は以下のとおりです。

- 1) [ 表示するパス情報 ] を選択してください。
- 2) [ 追加するパス ] を設定してください。
  - [ 追加するパス ] 設定フィールドの右横にある [ 参照 ] ボタンをクリックするとパスの選択ができます。
- 3) [ 追加 ] ボタンをクリックしてください。
- 4) ほかに設定する項目がなければ [OK] ボタンをクリックしてください。

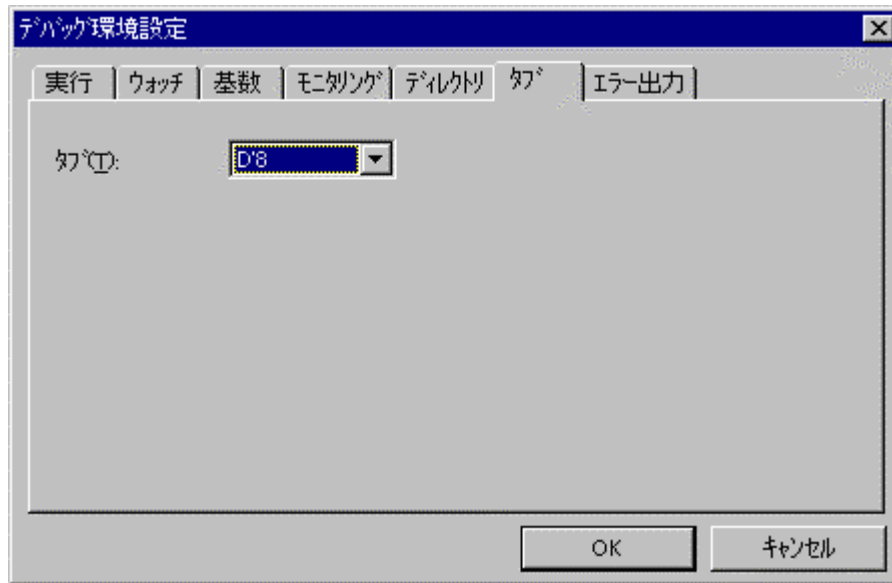
ディレクトリの設定解除手順は以下のとおりです。

- 1) [ ディレクトリ ] から削除したいディレクトリを選択してください。
- 2) [ 削除 ] ボタンをクリックしてください。
- 3) ほかに設定する項目がなければ [OK] ボタンをクリックしてください。



## ● タブ

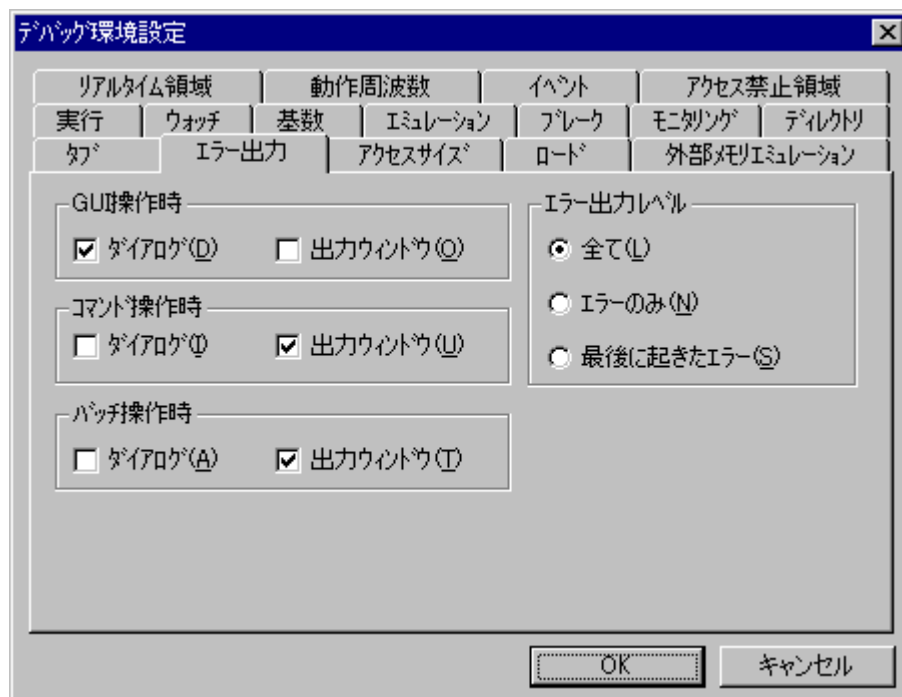
図 4.7-20 デバッグ環境設定 (タブ)



- タブ
  - タブを選択します (D'4/D'8)。

● エラー出力

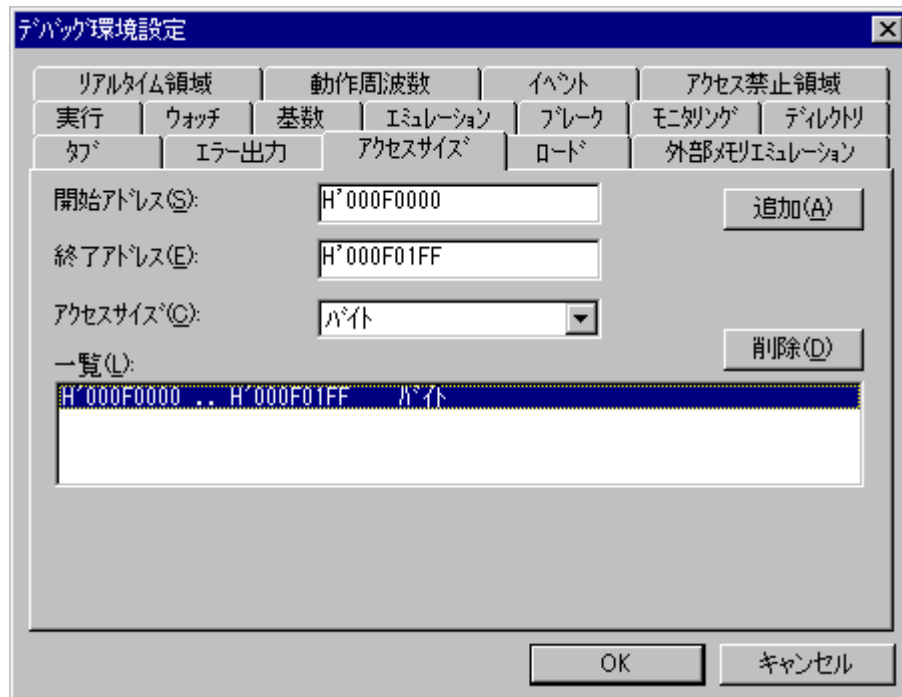
図 4.7-21 デバッグ環境設定 (エラー出力)



- GUI 操作時
  - GUI 操作時のエラー出力方法を選択します。
- コマンド操作時
  - コマンド操作時のエラー出力方法を選択します。
- バッチ操作時
  - バッチ操作時のエラー出力方法を選択します。
- エラー出力レベル
  - 複数エラーが発生した際の出力を設定します。

## ● アクセスサイズ

図 4.7-22 デバッグ環境設定 (アクセスサイズ)



デバッガがメモリをアクセスする時のアクセスサイズを設定する機能です。

この設定を行わない場合は、コマンド修飾子または、最適なサイズでメモリアクセスを行います。

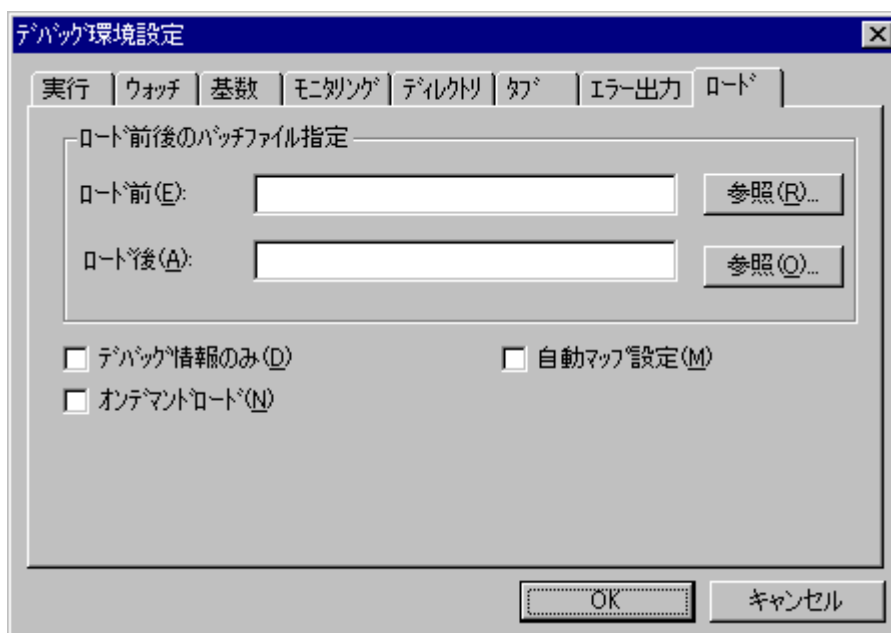
内蔵リソースについては、自動設定しますのでこの機能で設定は不要です。

ただし、FILL, MOVE, COMPARE コマンドは、すべてバイトアクセスになります。

- 開始アドレス
  - 設定する開始アドレスを指定します。
- 終了アドレス
  - 設定する終了アドレスを指定します。
- アクセスサイズ
  - 設定するアクセスサイズを指定します。(バイト / ハーフワード / ワード)
- 一覧
  - 現在設定されている領域を表示します。

● ロード

図 4.7-23 デバッグ環境設定 (ロード)



プロジェクトに登録されているターゲットファイルをロードする時の環境について設定します。

- ロード前後のバッチファイル指定
  - ロード前  
ターゲットファイルをロードする前に実行されるバッチファイルを指定します。これはデバッグのセットアップウィザードで変更することも可能です。
  - ロード後  
ターゲットファイルをロードした後に実行されるバッチファイルを指定します。これはデバッグのセットアップウィザードで変更することも可能です。
- デバッグ情報のみ
  - デバッグ情報のみをロードするかどうかを指定します。チェックされるとデバッグ情報のみがロードされます。
- 自動マップ設定
  - 自動マップ設定を有効にするかどうかを設定します。チェックすると自動マップ設定が有効になります。
- オンデマンドロード
  - オンデマンドロードするかどうかを設定します。チェックするとデバッグ情報をオンデマンドロードします。

## ● 外部メモリエミュレーション【DSU4 (MB2198)】

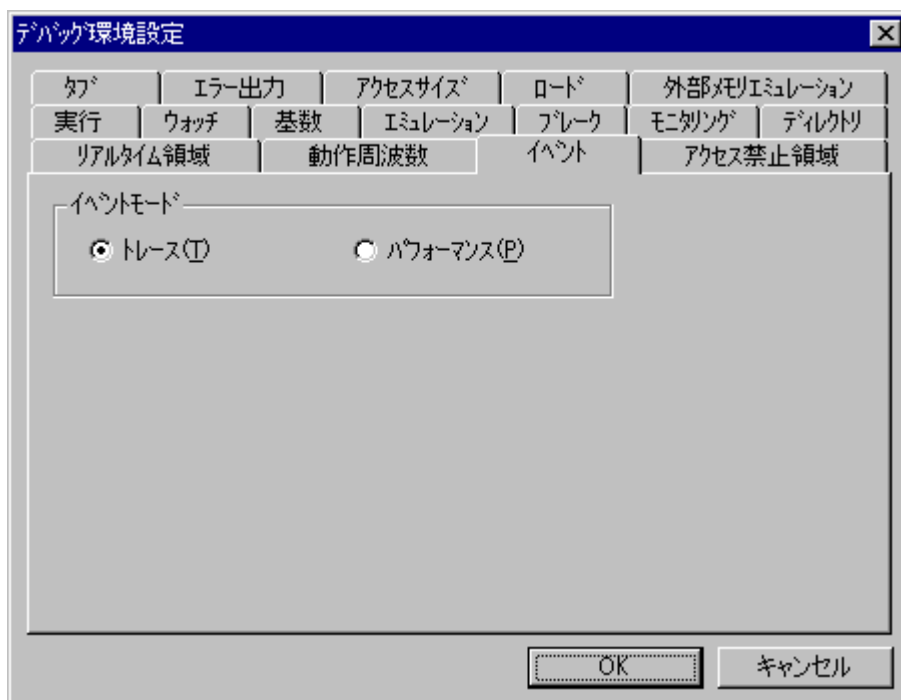
図 4.7-24 デバッグ環境設定 (外部メモリエミュレーション)



- 有効 / 無効
  - 外部メモリエミュレーション機能の有効 / 無効を指定します。
- チップセレクト
  - 外部バスに出力できるチップセレクト番号を指定します。
    - FR 系 : CS1 ~ CS5
    - FRex 系 : CS0 ~ CS7
- メモリ種別
  - 外部メモリへのライトアクセス許可 / 禁止を指定します。

● イベント【MB2198(FR60Lite, FR80S)】

図 4.7-25 デバッグ環境設定 (イベント)



- イベントモード
  - トレース  
イベント機能をトレース関連に使用します。データ監視，トレーストリガ，シーケンサ関連の機能が有効となります。
  - パフォーマンス  
イベント機能をパフォーマンス測定に使用します。パフォーマンス関連の機能が有効となります。

< 注意事項 >

FR60Lite または FR80S 使用时以外ではこの機能は使用できません。詳しくは『SOFTUNE Workbench 機能説明書』「2.3.8 パフォーマンス測定」を参照してください。

#### 4.7.2.4 デバッグ機能の選択

---

ここでは、デバッグ機能（デバッグモード）の選択手順について説明します。

---

##### ■ デバッグ機能（デバッグモード）の選択手順

1. [ 選択可能リスト ] から使用したい機能を選択してください。
2. [ OK ] ボタンをクリックまたは機能名をダブルクリックしてください。

デバッグ機能（デバッグモード）が切り替わります。

図 4.7-26 デバッグ機能の選択



---

##### < 注意事項 >

- 選択可能なデバッグモードはエミュレータまたはその接続形態によって変わってきます。
  - 各デバッグモードの機能については、『SOFTUNE Workbench 機能説明書』「2.3.1.6 デバッグモード」を参照してください。
  - デバッグモードを変更すると、トレース、パフォーマンスのデータがクリアされます。起動時は RealTimeMemory モードに設定されています。
-

## 4.7.2.5 セットアップウィザード

---

ここでは、デバッガのセットアップウィザードの操作の手順について説明します。

---

### ■ セットアップウィザード操作手順

- 1) [デバッガ起動の選択] で [セットアップ名] を選択して、[OK] ボタンをクリックしてください。
  - 確認のダイアログが出ます。
- 2) [OK] ボタンをクリックしてください。
  - デバッガのセットアップウィザードが起動します。
- 3) [次へ] ボタンをクリックしてください。
- 4) [デバッガ種別] を選択して、[次へ] ボタンをクリックしてください。
  - この先は、デバッガ種別により設定が異なります。
  - これ以降、設定をやり直す場合は、[戻る] ボタンをクリックしてください。一つ前の設定画面に戻ります。

### ■ エミュレータデバッガを選択した場合の手順

- 1) エミュレータの種類を選択してください。
- 2) [次へ] ボタンをクリックしてください。
- 3) デバイスの種類 ([RS232C], [LAN], [USB] のいずれか) を選択してください。
  - RS232C を選択した場合は、ポートとボーレートを設定してください。
  - LAN を選択した場合は、ホスト名を設定してください。
- 4) [次へ] ボタンをクリックしてください。
- 5) モニタプログラムをデバッグ開始時に自動ロードするかどうかの設定を行ってください。
- 6) [次へ] ボタンをクリックしてください。
- 7) ターゲットファイルをデバッグ時に自動ロードするかどうかの設定を行ってください。
- 8) 必要に応じて、ロード前後のバッチファイルの指定を行ってください。
  - それぞれの入力フィールド右横の [参照] ボタンでファイルの選択ができます。
- 9) [次へ] ボタンをクリックしてください。
- 10) セットアップファイルへ設定する項目を選択してください。
  - [指定する] を選択した場合は、[設定] ボタンをクリックして、項目選択ダイアログを表示し、設定する項目を選択して [OK] ボタンをクリックしてください。
- 11) [次へ] ボタンをクリックしてください。
- 12) [完了] ボタンをクリックしてください。
  - これで、エミュレータのセットアップは完了です。



## ■ シミュレータデバッガを選択した場合の手順

- 1) ターゲットファイルをデバッグ時に自動ロードするかどうかの設定を行ってください。
- 2) 必要に応じて、ロード前後のバッチファイルの指定を行ってください。
  - それぞれの入力フィールド右横の[参照]ボタンでファイルの選択ができます。
- 3) [次へ]ボタンをクリックしてください。
- 4) セットアップファイルへ設定する項目を選択してください。
  - [指定する]を選択した場合は、[設定]ボタンをクリックして、項目選択ダイアログを表示し、設定する項目を選択して[OK]ボタンをクリックしてください。
- 5) [次へ]ボタンをクリックしてください。
- 6) [完了]ボタンをクリックしてください。
  - これで、シミュレータデバッガのセットアップは完了です。

## ■ モニタデバッガを選択した場合の手順

- 1) デバイスの種類([RS232C])を選択してください。
  - RS232Cを選択した場合は、ポートとボーレートを設定してください。
- 2) [次へ]ボタンをクリックしてください。
- 3) ターゲットファイルをデバッグ時に自動ロードするかどうかの設定を行ってください。
- 4) 必要に応じて、ロード前後のバッチファイルの指定を行ってください。
  - それぞれの入力フィールド右横の[参照]ボタンでファイルの選択ができます。
- 5) [次へ]ボタンをクリックしてください。
- 6) セットアップファイルへ設定する項目を選択してください。
  - [指定する]を選択した場合は、[設定]ボタンをクリックして、項目選択ダイアログを表示し、設定する項目を選択して[OK]ボタンをクリックしてください。
- 7) [次へ]ボタンをクリックしてください。
- 8) [完了]ボタンをクリックしてください。
  - これで、モニタデバッガのセットアップは完了です。

### 4.7.3 デバッガのメモリマップ

デバッガのメモリマップの設定を行います。

#### ■ メモリマップの設定 (シミュレータデバッガ時のみ表示)

デバッグ対象システムと同様のメモリマップをデバッガ内部に確保します。アドレス範囲、その範囲に対するアクセス属性、内容 (プログラムコードまたはデータ) が設定できます。メモリマップはデバッグ対象プログラムの ABS 形式ファイルを読み込めば自動的に設定されますが、ここで任意に設定することもできます。

図 4.7-27 メモリマップ設定ダイアログ



- 開始アドレス
  - 設定する開始アドレスを指定します。
- 終了アドレス
  - 設定する終了アドレスを指定します。
- 属性
  - メモリ空間の属性を指定します (リード/ライト/コード)。
- マップ領域
  - 現在設定されている領域を表示します。

## 4.7.4 ツールの設定

SOFTUNE Workbench から直接起動するツールの設定を行います。

### ■ ツールについて

この機能は、C/C++ コンパイラやアセンブラなどの SOFTUNE Workbench の基本機能を担当するツールではなく、簡単なフィルタなどの補助ツールを SOFTUNE Workbench から直接起動できるように組み込む機能です。

例えば、"dir" などを組み込んでおけば、DOS プロンプトから dir コマンドを実行したのと同じ結果を SOFTUNE Workbench のアウトプットウィンドウに出力することができます。

図 4.7-28 ツールの設定ダイアログ



## ■ 設定手順

- 1) [環境] - [ツールの設定] を選択してください。
  - ツールの設定ダイアログ (図 4.7-28) が表示されます。
- 2) 既に登録されている名前と重複しないタイトルを設定してください。
- 3) 登録するツールの実行ファイル名を指定してください。
  - このフィールドの右横にある [参照] ボタンをクリックすると、ファイル選択のファイルダイアログが表示され、そこからツールの実行ファイル名を選択できます。
- 4) オプションを設定してください。
  - ここで、マクロ記述が利用できます。マクロ記述については、『SOFTUNE Workbench 機能説明書』の「1.11 マネージャ部で利用できるマクロ記述」を参照してください。
- 5) 実行時のディレクトリを記述してください。
  - 実行時に特定のディレクトリに移動する必要がなければ、記述しなくてもかまいません。
  - このフィールドの右横にある [参照] ボタンをクリックすると、ファイル選択のファイルダイアログが表示され、そこから実行時ディレクトリを選択できます。
- 6) 必要に応じて、[実行時に追加オプションを指定] および [アウトプットウィンドウを使用] にチェックマークを付けてください。
  - [実行時に追加オプションを指定] にチェックした場合は、ツールを起動したときに、追加オプションの入力を要求します。
  - [アウトプットウィンドウを使用] にチェックした場合は、ツールの出力 (標準出力装置または標準エラー出力装置への出力) をアウトプットウィンドウに出力するようにします。
- 7) [設定] ボタンをクリックしてください。

## ■ 削除手順

- 1) [環境] - [ツールの設定] を選択してください。
  - ツールの設定ダイアログ (図 4.7-28) が表示されます。
- 2) ツールリストから削除したいツールのタイトルを選択してください。
- 3) [削除] ボタンをクリックしてください。

## ■ 変更手順

- 1) [環境] - [ツールの設定] を選択してください。
  - ツールの設定ダイアログ (図 4.7-28) が表示されます。
- 2) ツールリストから変更したいツールのタイトルを選択してください。
  - [タイトル], [実行ファイル名], [オプション], [実行時のディレクトリ], [実行時に追加オプションを指定], [アウトプットウィンドウを使用] に設定されている値が表示されます。
- 3) [実行ファイル名], [オプション], [実行時のディレクトリ], [実行時に追加オプションを指定], [アウトプットウィンドウを使用] を変更してください。
  - [タイトル] を変更すると, 別のツールとして追加登録されます。
- 4) [設定] ボタンをクリックしてください。
  - 変更するかどうか問い合わせるダイアログが表示されます。
- 5) [はい] をクリックしてください。

## ■ ツールの設定例

### ● notepad を使用する場合

タイトル：メモ帳  
 実行ファイル名：notepad.exe  
 オプション：%f  
 実行時のディレクトリ：%x  
 実行時に追加オプションを指定：チェックなし  
 アウトプットウィンドウを使用：チェックなし

### ● dir コマンドを登録する場合

タイトル：Dir  
 実行ファイル名：commnd.com  
 オプション：/c dir  
 実行時のディレクトリ：  
 実行時に追加オプションを指定：チェックなし  
 アウトプットウィンドウを使用：チェックあり

## 4.7.5 キーボードの設定

---

ショートカットキーの定義を行うことができます。

---

### ■ 設定手順

- 1) 種類を選択してください。
  - 機能の一覧が「機能一覧」に表示されます。
- 2) 「機能一覧」から設定したい機能を選択してください。
  - 選択された機能の説明が、ダイアログ下部の説明欄に表示されます。また、既に割り当てられている機能を選択した場合は、「現在の設定」に、現在割り当てられているキーが表示されます。
- 3) 「新しい割り当て」にフォーカスを設定してから、機能に割り当てたいキーをキーボードで指定してください（キーを押してください）。
- 4) 「設定」ボタンをクリックしてください。

### ■ 既に割り当てられているキーの解除手順

- 1) 解除したいキーに対応づけられている機能を選択してください（設定手順の 1）、2）を参照してください）。
  - 「現在の設定」に、現在割り当てられているキーが表示されます。
- 2) 「現在の設定」に表示されたキーのリストから、削除したいキーを選択してください。
- 3) 「削除」ボタンをクリックしてください。
  - 本当に削除してよいかどうかの確認のダイアログが表示されます。
- 4) 「閉じる」ボタンをクリックしてください。

### ■ 既に割り当てられているキーの変更手順

- 1) 既に割り当てられているキーを解除してください（既に割り当てられているキーの解除手順を参照してください）。
- 2) 「新しい割り当て」にフォーカスを設定してから、その機能に割り当てたいキーをキーボードで指定してください（そのキーを押してください）。
- 3) 「設定」ボタンをクリックしてください。

### ■ 現在の設定状況の一覧を見るには

「定義一覧」ボタンをクリックしてください。キー定義リストが表示されます。

### ■ すべてのキー設定を初期状態に戻したい場合

「リセット」ボタンをクリックしてください。

< 注意事項 >

- [ 設定 ] ボタン , [ リセット ] ボタンをクリックした後は , その設定 ( リセット ) の取り消しはできません。誤ってクリックした場合は , 再度設定し直してください。
- 設定可能なキーは ( 表 4.7-2 ) を参照してください。
- 一つの機能に複数のキーを設定することができます。その場合は設定されているどのキーも同じ機能になります。

図 4.7-29 キー設定ダイアログ

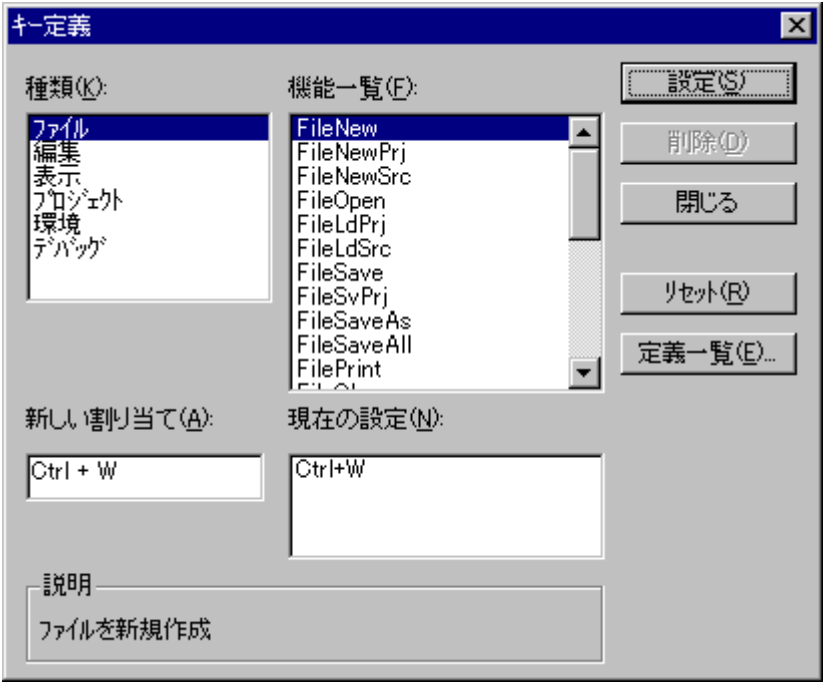


表 4.7-2 設定可能なキーの一覧

キー	説 明
CTRL + A ~ Z	CTRL キーを押しながら A ~ Z のどれかのキーを押します。
SHIFT + CTRL + A ~ Z	SHIFT キーと CTRL キーを押しながら A ~ Z のどれかのキーを押します。
SHIFT + F1	SHIFT キーを押しながら F1 キーを押します。

## 4.7.6 エディタの設定

任意のエディタを登録して、標準エディタの代わりに使用することができます。

### ■ エディタの登録

SOFTUNE Workbench にあらかじめ内蔵されている標準エディタの代わりに使用するエディタを登録します。登録したら実際のファイル編集を行う前に、そのエディタをSOFTUNE Workbench のエディタとして設定してください。ファイルの編集は、ここに登録されているエディタのうち、[使用するエディタ]に設定されているエディタで行います。

図 4.7-30 エディタの設定ダイアログ

エディタの設定

使用するエディタ(U): 標準エディタ

タイトル(T):

実行ファイル名(N):

オプション(O):

実行時のディレクトリ(D):

参照(R)...

参照(R)...

設定(E)

エディタリスト(L):

標準エディタ

削除(D)

マクロ記述

%f = %  
%l = Line Number

%f = Filename  
%x = Project Path

OK キャンセル



## ■ 登録手順

- 1) [環境] - [エディタの設定] を選択してください。
  - エディタの設定ダイアログ (図 4.7-30) が表示されます。
- 2) 既に登録されている名前と衝突しないユニークなタイトルを設定してください。
- 3) 登録するエディタの実行ファイル名を指定してください。
  - このフィールドの右横にある [参照] ボタンをクリックすると、ファイル選択のファイルダイアログが表示され、そこからエディタの実行ファイル名を選択できます。
- 4) オプションを設定してください。
  - ここで、マクロ記述が利用できます。マクロ記述については、『SOFTUNE Workbench 機能説明書』の「1.9 外部エディタ登録」を参照してください。
- 5) 実行時のディレクトリを記述してください。
  - 実行時に特定のディレクトリに移動する必要がなければ、記述しなくてもかまいません。
  - このフィールドの右横にある [参照] ボタンをクリックすると、ファイル選択のファイルダイアログが表示され、そこから実行時ディレクトリを選択できます。
- 6) [設定] ボタンをクリックしてください。

## ■ 削除手順

- 1) [環境] - [エディタの設定] を選択してください。
  - エディタの設定ダイアログ (図 4.7-30) が表示されます。
- 2) エディタリストから削除したいエディタのタイトルを選択してください。
- 3) [削除] ボタンをクリックしてください。

## ■ 変更手順

- 1) [環境] - [エディタの設定] を選択してください。
  - エディタの設定ダイアログ (図 4.7-30) が表示されます。
- 2) エディタリストから変更したいエディタのタイトルを選択してください。
  - タイトル、実行ファイル名、オプション、実行時のディレクトリに設定されている値が表示されます。
- 3) 実行ファイル名、オプション、実行時のディレクトリを変更してください。
  - タイトルを変更すると、別のエディタとして追加登録されます。
- 4) [設定] ボタンをクリックしてください。
  - 変更するかどうか問い合わせるダイアログが表示されます。
- 5) [はい] をクリックしてください。

## ■ 使用するエディタの設定方法

- 1) 登録手順にそって, 外部エディタを登録してください。
- 2) [使用するエディタ]のフィールドにある[ ] ボタンをクリックしてください。
  - 登録されているエディタのタイトルが, ドロップダウンリストに表示されます。
- 3) リストから使用したいエディタのタイトルを選択してください。

## ■ 設定例

### ● 富士通 Power EDITOR の設定例

タイトル : Power EDITOR

実行ファイル名 : c : ¥Powered¥powered.exe

オプション : "%f"-g%l

実行時のディレクトリ : %x

上記入力後, [設定] ボタンをクリックするとエディタリストに登録されます。

登録後, [使用するエディタ] で "Power EDITOR" を選択し, [OK] ボタンをクリックしてください。

## 4.7.7 エラージャンプの設定

さまざまなツールのエラーメッセージのパターンを登録して、エラージャンプさせることができます。

### ■ 設定手順

- 1) [環境] - [エラージャンプの設定] を選択してください。
  - エラージャンプの設定ダイアログ (図 4.7-31) が開きます。
- 2) シンタックスを入力してください。
  - シンタックスについての詳細は、『SOFTUNE Workbench 機能説明書』の「1.7 エラージャンプの機能」を参照してください。
- 3) 必要ならコメントを入力してください。
  - 各シンタックスにコメントを付けることができます。
- 4) [設定] ボタンをクリックしてください。

図 4.7-31 エラージャンプの設定ダイアログ



### ■ 削除手順

- 1) [環境] - [エラージャンプの設定] を選択してください。
  - エラージャンプの設定ダイアログ (図 4.7-31) が開きます。
- 2) シンタックスリストから削除するシンタックスを選択してください。
- 3) [削除] ボタンをクリックしてください。

## ■ 変更手順

既に設定されているシンタックスを変更するには以下の手順で行ってください。

- 1) [環境] - [エラージャンプの設定] を選択してください。
  - エラージャンプの設定ダイアログ (図 4.7-31) が開きます。
- 2) シンタックスリストから変更するシンタックスを選択してください。
  - シンタックスとコメントがそれぞれのフィールドに表示されます。
- 3) シンタックスとコメントを修正して, [設定] ボタンをクリックしてください。
  - 新規に追加設定されます。
- 4) 不要なシンタックス (変更前のシンタックス) を削除してください。

## ■ 解析順序の変更, 適用の ON/OFF

### ● 解析順序の変更

エラーメッセージの解析は, シンタックスリストの上の方に登録されているパターンから行います。正しく解析を行うためには, 解析の順序を変更しなければならない場合があります。以下の手順で, 解析順序の変更ができます。

- 1) [環境] - [エラージャンプの設定] を選択してください。
  - エラージャンプの設定ダイアログ (図 4.7-31) が開きます。
- 2) シンタックスリストから順序を変更するシンタックスを選択してください。
- 3) [上へ] ボタンまたは [下へ] ボタンをクリックして, 設定したい位置まで移動してください。

### ● 適用の ON/OFF

[適用] のチェックマークをはずすと, 登録されているシンタックスでの解析を行わなくなります。

シンタックスリストには, 適用のチェックマークがついている状態のときは ON, 適用のチェックマークがはずれている状態では OFF と表示されます。

## ■ 設定例

### ● エラー形式が, エラーメッセージ: 行番号 ファイル名の場合

シンタックス: %\*: %l %f

コメント: sample

---

### < 注意事項 >

シンタックスリストに SYSTEM と表示されているシンタックスは削除できません。

---

## 4.7.8 ツールの起動

登録されている外部ツールを起動します。

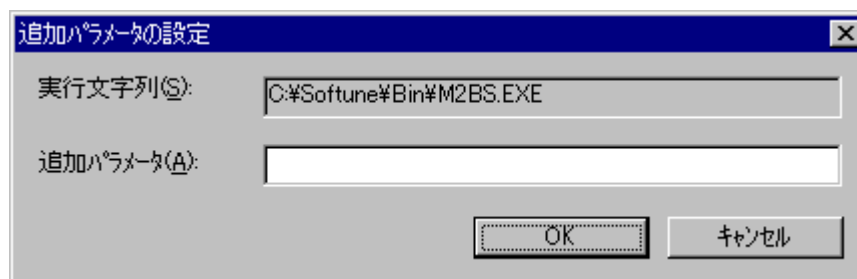
### ■ ツールの起動

「4.5.3.1 メンバの追加 - ファイル」で設定したツールがサブメニューに登録されています。これを選択することでツールを起動できます。

ツールの設定時に、「追加オプションを指定」にチェックを付けておくと、ツールを起動する前に追加オプションの設定のためのダイアログ (図 4.7-32) が開きます。ここで、追加オプションを設定して [OK] ボタンをクリックしてください。

ツールの設定時に指定したパラメータの後ろに、ここで指定したパラメータを追加してツールを起動します。

図 4.7-32 追加パラメータの設定ダイアログ



## 4.8 ウィンドウ

---

ウィンドウ表示を制御します。

---

### ■ ウィンドウの表示方法に関する制御

- 重ねて表示
- 上下に並べて表示
- 左右に並べて表示
- 上下に分割
- アイコンの整列
- ウィンドウの更新
- すべてのウィンドウの更新
- すべての閉じる

### ■ 開いているウィンドウのリスト

現在開かれている (アイコン化されているウィンドウも含めて) ウィンドウの一覧表示 9 個までが出力されます。10 個以上のウィンドウを開いている場合は、[ その他のウィンドウ ] の表示がでます。

## 4.8.1 重ねて表示 / 上下に並べて表示 / 左右に並べて表示

---

各子ウィンドウ (例えば, ソースウィンドウ, レジスタウィンドウ, 逆アセンブルウィンドウなど) の表示の形態を指定します。

---

### ■ 重ねて表示

現在開かれている子ウィンドウを重ねて表示します。

### ■ 上下に並べて表示

現在開かれている子ウィンドウを上下に並べて, 親ウィンドウいっぱいに表示します。

### ■ 左右に並べて表示

現在開かれている子ウィンドウを左右に並べて, 親ウィンドウいっぱいに表示します。

## 4.8.2 上下に分割

---

ウィンドウを上下に分割する位置を指定します。

---

### ■ 上下に分割

上下に分割可能なウィンドウにおいて、上下に分割する位置の指定を行います。上下に分割可能なウィンドウは、以下のとおりです。

- ソースウィンドウ
- 逆アセンブルウィンドウ
- トレースウィンドウ
- メモリウィンドウ
- カバレッジウィンドウ



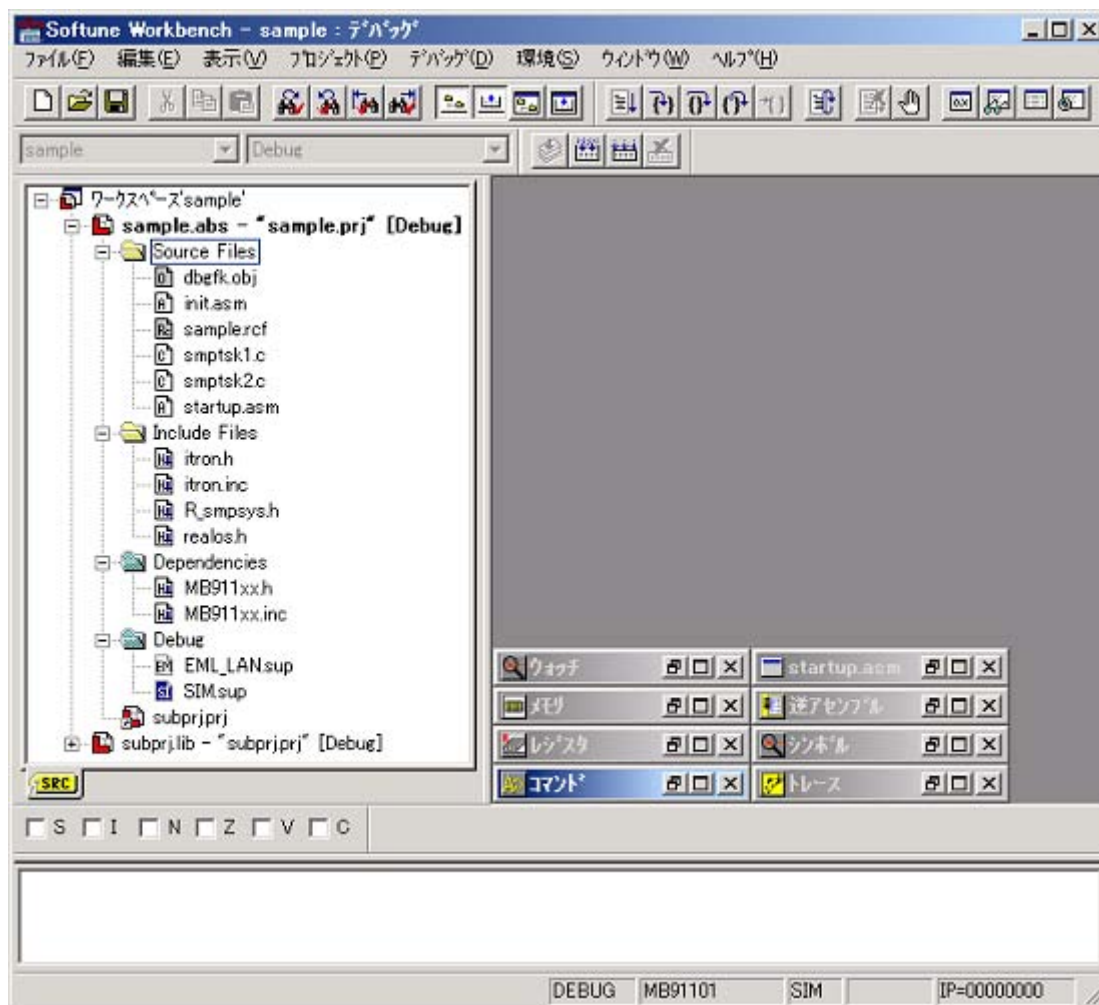
### 4.8.3 アイコンの整列

最小化されたウィンドウのアイコンの配置を整理します。

#### ■ アイコンの整列

最小化されているウィンドウをすべて、SOFTUNE Workbench のメインウィンドウに整列させます (図 4.8-1)。最小化されていないウィンドウには影響しません。

図 4.8-1 アイコンの整理実行後のウィンドウの状態



## 4.8.4 ウィンドウの更新

---

アクティブなウィンドウを最新の情報にします。

---

### ■ ウィンドウの更新

現在アクティブなウィンドウの情報を更新します。

## 4.8.5 すべてのウィンドウの更新

---

開いているすべてのウィンドウを最新の情報にします。

---

### ■ すべてのウィンドウの更新

プロジェクトウィンドウの SRC タブとアウトプットウィンドウ以外の現在開いているウィンドウの情報をすべて更新します。

## 4.8.6 すべて閉じる

---

開いているすべてのウィンドウを閉じます。

---

### ■ すべて閉じる

プロジェクトウィンドウとアウトプットウィンドウ以外の、現在開いているウィンドウをすべて閉じます。編集中でまだ保存されていないファイルエディットウィンドウがある場合は、保存するかどうかを問い合わせるダイアログが表示されます。

外部エディタでファイルを開いている場合は、この機能ではそれを閉じることできません。

ウィンドウをすべて閉じても、SOFTUNE Workbench の状態はそのままですので、例えばデバッグ中の場合でもレジスタ値などに影響はありません。

## 4.9 ヘルプ

---

オンラインヘルプを表示します。

---

### ■ オンラインヘルプ

- トピックの検索
- サポート情報

### ■ バージョン情報

バージョン情報

## 4.9.1 トピックの検索

---

キーワードでヘルプの項目を検索します。

---

### ■ 目次

オンラインヘルプの目次を階層的に表示します。目次から見たい項目を探す場合に使用します。

### ■ キーワード

知りたい項目を直接指定してヘルプファイルを検索します。

## 4.9.2 サポート情報

---

添付のサポート情報ファイルを , エディットウィンドウに開きます。

---

### ■ サポート情報

サポート情報には , 添付のマニュアルに記載されていない情報が記述されています。  
SOFTUNE Workbench をご使用になられる前に , ご一読されることをお勧めします。

## 4.9.3 バージョン情報

SOFTUNE のバージョン情報を表示します。

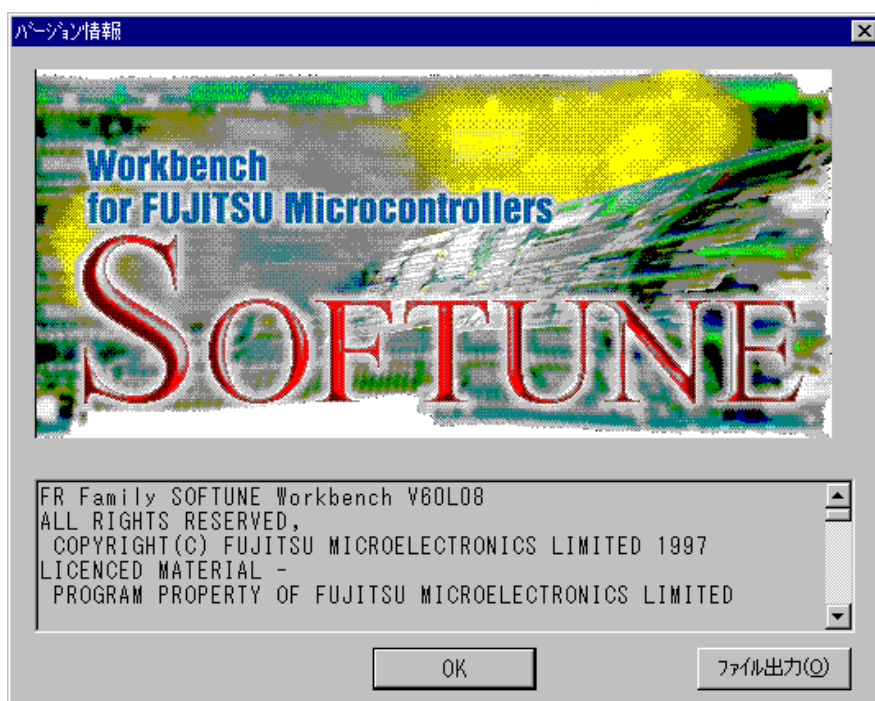
### ■ 非デバッグセッション時のバージョン表示

ダイアログを開いて、SOFTUNE Workbench のロゴマークとバージョン番号を表示します。

### ■ デバッグセッション時のバージョン表示

ダイアログを開いて、SOFTUNE Workbench のロゴマークとバージョン番号、さらに現在選択しているデバッガの詳細な情報を表示します。

図 4.9-1 バージョン情報



#### ● 表示内容

バージョン情報ダイアログに表示される内容は下記のとおりです。

```
FR Family SOFTUNE Workbench VXXLXX
ALL RIGHTS RESERVED, COPYRIGHT(C) FUJITSU LIMITED 1997
LICENCED MATERIAL - PROGRAM PROPERTY OF FUJITSU LIMITED
```

```
=====
Cpu information file path : CPU 情報ファイルのパス
Cpu information file version : CPU 情報ファイルのバージョン
=====
```

```
Add in DLLs
```



---

SiCmn

Product name : SOFTUNE Workbench

File Path : SiC911.dll のパス

Version : SiC911.dll のバージョン

---

SiiEd

File Path : SiiEd3.ocx のパス

Version : SiiEd3.ocx のバージョン

---

SiM911

Product name : SOFTUNE Workbench

File Path : SiM911.dll のパス

Version : SiM911.dll のバージョン

---

Language Tools

- FR Family SOFTUNE C/C++ Compiler のバージョン

File Path : fcc911s.exe のパス

- FR Family SOFTUNE Assembler のバージョン

File Path : fasm911s.exe のパス

- FR Family SOFTUNE Linker のバージョン

File Path : flnk911s.exe のパス

- FR Family SOFTUNE Librarian のバージョン

File Path : flib911s.exe のパス

- SOFTUNE FJ-OMF to S-FORMAT Converter のバージョン

File Path : f2ms.exe のパス

- SOFTUNE FJ-OMF to INTEL-HEX Converter のバージョン

File Path : f2is.exe のパス

- SOFTUNE FJ-OMF to INTEL-EXT-HEX Converter のバージョン

File Path : f2es.exe のパス

- SOFTUNE FJ-OMF to HEX Converter のバージョン

File Path : f2hs.exe のパス

---

SiOsM

Product name : Softune Workbench

File Path : SiOsM911.dll のパス

Version : SiOsM911.dll のバージョン

---

FR Series Debugger DLL

## 第 4 章 メニュー

Product name : SOFTUNE Workbench

File Path : SiD911.dll のパス

Version : SiD911.dll のバージョン

-----  
Debugger type : 現在のデバッガ種別

MCU type : 現在選択しているターゲット MCU

VCpu dll name : 現在使用している VCpu dll のパスと名前

DSU type : 現在使用している DSU 種別【MB2197/MB2198】

Common version : モニタ ( 共通 ) のバージョン【MB2198】

Monitor version : モニタ ( 依存 ) のバージョン【MB2197/MB2198】

Configuration board ID : コンフィギュレーションボードの ID【MB2198( コンフィギュレーションボード接続時 )】

Configuration board version : コンフィギュレーションボードのバージョン【MB2198( コンフィギュレーションボード接続時 )】

MCU frequency : 動作周波数【MB2198】

Communication device : デバイス種別【MB2197/MB2198】

Baud rate : 通信ボーレート【MB2197/MB2198(RS 接続時 )】

Host name : LAN ホスト名【MB2197/MB2198(LAN 接続時 )】

USB name : USB 名【MB2198(USB 接続時 )】

REALOS version : REALOS のバージョン

=====

Current path : 現在使用しているプロジェクトのパス

Language : 現在使用している言語

Help file path : ヘルプファイルのパス

### ● ファイル出力

バージョン情報ダイアログに表示されている内容をファイルに保存します。

デフォルトのファイル名は「SOFTUNE\_SYS.txt」です。

---

### 参考 :

SOFTUNE Workbench について弊社への問い合わせの際には、必ずここに表示された内容を通知してください。

---

# 第5章

---

## アドインモジュール

SOFTUNE Workbench のアドインモジュールについて説明します。

5.1 カスタマイズバー

5.2 FLASH ロード

## 5.1 カスタマイズバー

---

**SOFTUNE Workbench のカスタマイズバーについて説明します。**

---

- 5.1.1 カスタマイズバーとは？
- 5.1.2 カスタマイズバーのメニュー
- 5.1.3 カスタマイズバーへの登録
  - 5.1.3.1 バッチファイルの登録
  - 5.1.3.2 Workbench メニューの登録
  - 5.1.3.3 外部ツールの登録
- 5.1.4 警告 / エラーメッセージ
- 5.1.5 注意事項

## 5.1.1 カスタマイズバーとは？

カスタマイズバーとそのインストール方法について説明します。

### ■ カスタマイズバーとは？

カスタマイズバーとは、デバッガ起動中に使用するバッチファイルや Workbench メニュー、外部ツールをツールバーに登録して、ボタン1つで簡単に呼び出せる機能です。

最大10個まで登録可能で、1度設定した項目はワークスペース表示時に復元されます。

図 5.1-1 カスタマイズバー



カスタマイズバーにバッチファイルまたは Workbench メニューを登録すると、ワークスペース表示時に登録した番号のボタンが有効になります。

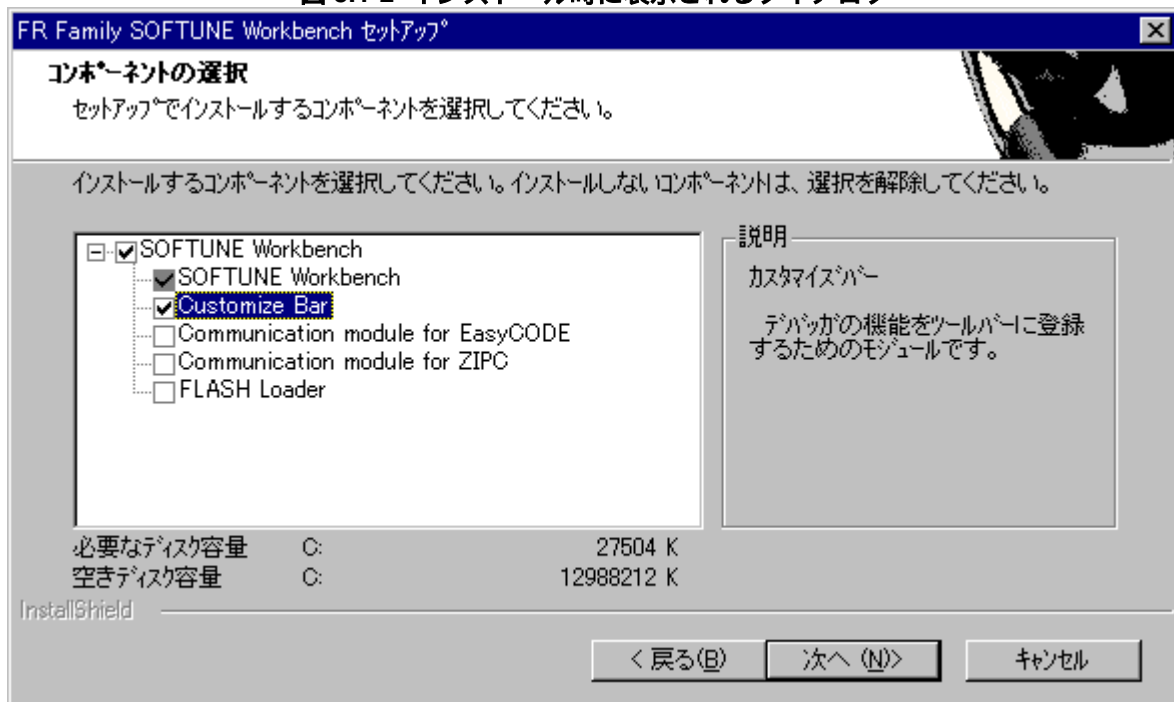
またツールバーのボタンにマウスカーソルを当てると、現在の登録内容が表示されます。

### ■ カスタマイズバーのインストール方法

SOFTUNE Workbench インストール時に表示されるダイアログ（図 5.1-2）にて「Customize Bar」をチェックすると、カスタマイズバーがインストールされます。

なお、SOFTUNE Workbench インストール時にカスタマイズバーをインストールしなかった場合でも、カスタマイズバーのみの追加インストールが可能です。

図 5.1-2 インストール時に表示されるダイアログ



## 第 5 章 アドインモジュール

カスタマイズバーをインストールすると、SOFTUNE Workbench の [ 表示 ] メニュー内に [ カスタマイズバー ] ( 図 5.1-3 ) が追加され、カスタマイズバー用ツールバー ( 図 5.1-1 ) が表示されます。

## 5.1.2 カスタマイズバーのメニュー

カスタマイズバーのメニューについて説明します。

### ■ カスタマイズバーのメニュー

[ カスタマイズバー ] には2つのサブメニューがあります。

設定 ... カスタマイズバーにバッチファイルや Workbench メニューを登録します。

本メニューはワークスペースを開くと有効になります。

表示 ... カスタマイズバー用ツールバーの表示 / 非表示を切り替えます。

本メニューは SOFTUNE Workbench 起動時は常に有効です。

図 5.1-3 [ カスタマイズバー ] メニュー



## 5.1.3 カスタマイズバーへの登録

カスタマイズバーへの登録について説明します。

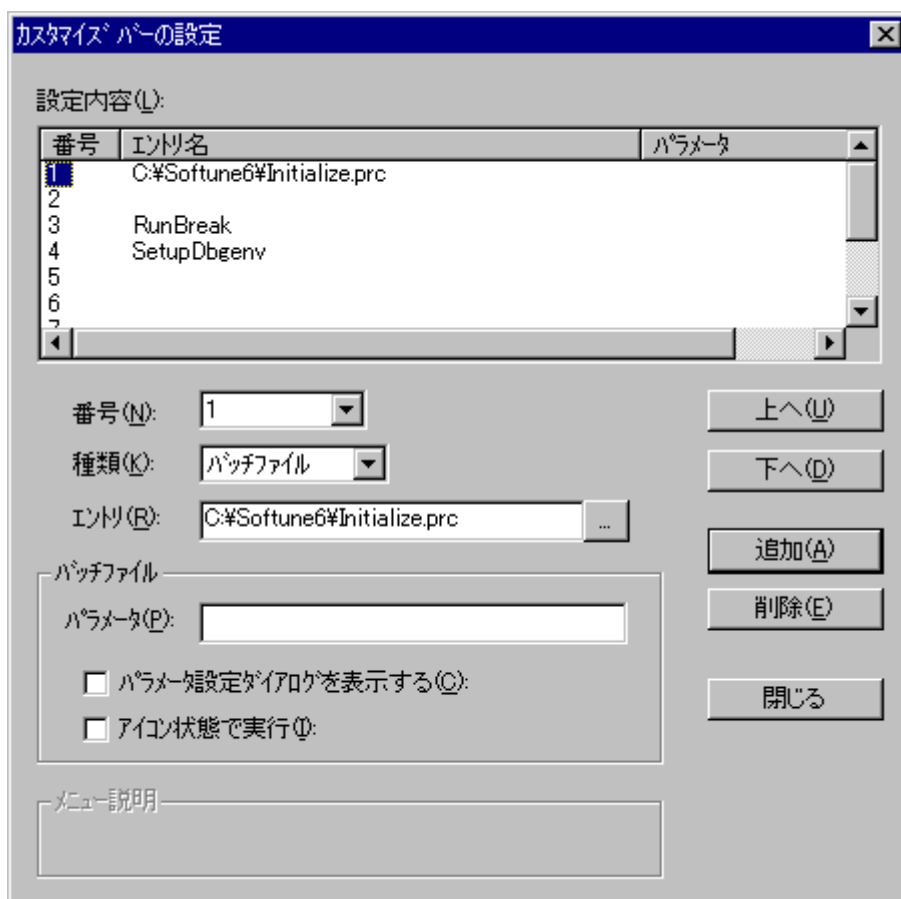
### ■ カスタマイズバーへの登録

カスタマイズバーには「バッチファイル」または「Workbench メニュー」、「外部ツール」を登録できます。

登録は[表示]-[カスタマイズバー]-[設定]を選択すると表示される「カスタマイズバー設定ダイアログ(図 5.1-4)」にて行います。

詳細な登録手順については「5.1.3.1 バッチファイルの登録」、「5.1.3.2 Workbench メニューの登録」、「5.1.3.3 外部ツールの登録」を参照してください。

図 5.1-4 カスタマイズバー設定ダイアログ





## ■ ダイアログの説明

### ● 設定内容

カスタマイズバーに登録した設定内容が表示されます。

### ● 番号

カスタマイズバーに登録する番号を指定します。1 ~ 10 が指定できます。

### ● 種類

カスタマイズバーに登録する項目を『バッチファイル』または『メニュー』のどちらかを選択します。

### ● エントリ

#### 1. 「種類」にて「バッチファイル」を選択した場合

カスタマイズバーに登録するバッチファイル名を入力します。一覧から選択したい場合は、右の参照ボタンを押すと表示される『ファイルを開くダイアログ (図 5.1-5)』にて指定できます。

#### 2. 「種類」にて「メニュー」を選択した場合

カスタマイズバーに登録する Workbench メニューを入力します。右の参照ボタンを押すと表示される『メニュー一覧ダイアログ (図 5.1-7)』から指定します。

#### 3. 「種類」にて「ツール」を選択した場合

カスタマイズバーに登録する外部ツールを入力します。右の参照ボタンを押すと表示される『ツール選択ダイアログ (図 5.1-11)』から指定します。

### ● パラメータ (「種類」が「バッチファイル」の場合のみ有効)

バッチファイル実行時のパラメータを入力します。

### ● パラメータ設定ダイアログを表示する (「種類」が「バッチファイル」の場合のみ有効)

カスタマイズバーによるバッチファイル実行時にパラメータを設定 (変更) 可能な『パラメータ設定ダイアログ (図 5.1-6)』を表示します。

### ● アイコン状態で実行 (「種類」が「バッチファイル」の場合のみ有効)

カスタマイズバーによるバッチファイル実行時に Workbench をアイコン状態にします。

### ● メニュー説明 (「種類」が「メニュー」の場合のみ有効)

カスタマイズバーに登録する Workbench メニューの説明を表示します。

### ● 上へ

「設定内容」に表示されている登録内容の順番を 1 つ上に変更できます。既に 1 つ上の番号に登録されている場合はその順番を入れ替えます。

### ● 下へ

「設定内容」に表示されている登録内容の順番を 1 つ下に変更できます。既に 1 つ下の番号に登録されている場合は順番を入れ替えます。

### ● 追加

バッチファイルまたは Workbench メニュー、外部ツールをカスタマイズバーに登録します。指定した番号に既に登録されている場合は、上書き登録を行います。

● 削除

カスタマイズバーの登録内容を削除します。「設定内容」から削除したい番号を指定します。

図 5.1-5 ファイルを開くダイアログ

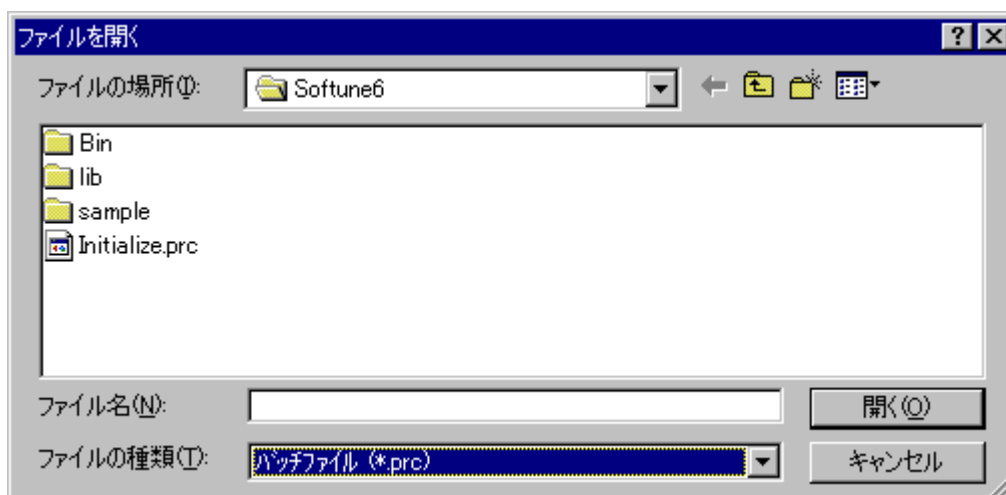


図 5.1-6 パラメータ設定ダイアログ

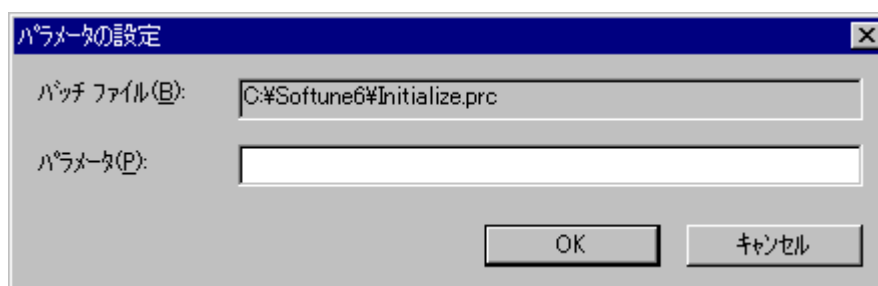


図 5.1-7 メニュー一覧ダイアログ

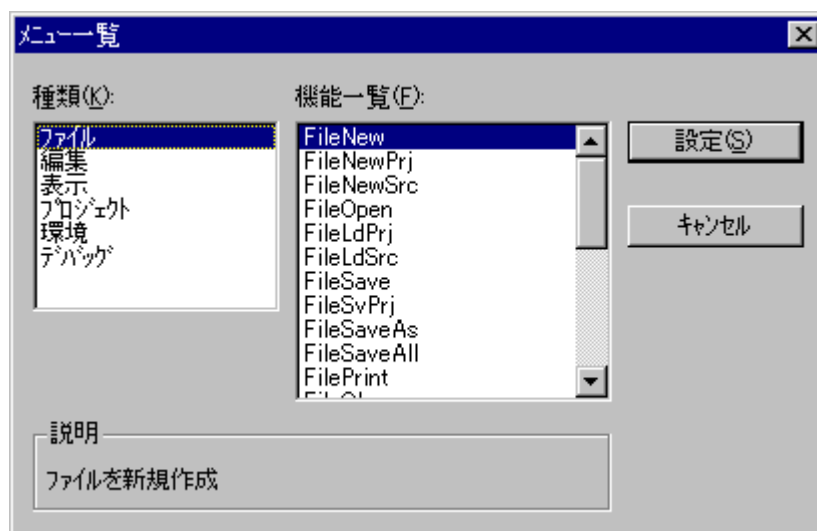


図 5.1-8 ツール選択ダイアログ

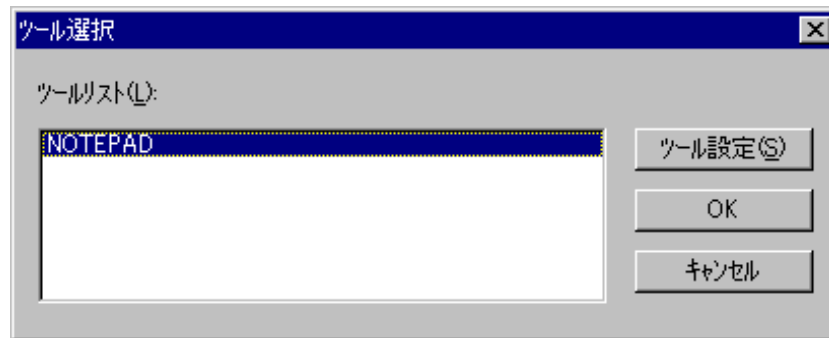


図 5.1-9 ツールの設定ダイアログ



### 5.1.3.1 バッチファイルの登録

---

カスタマイズバーにバッチファイルを登録する方法について説明します。

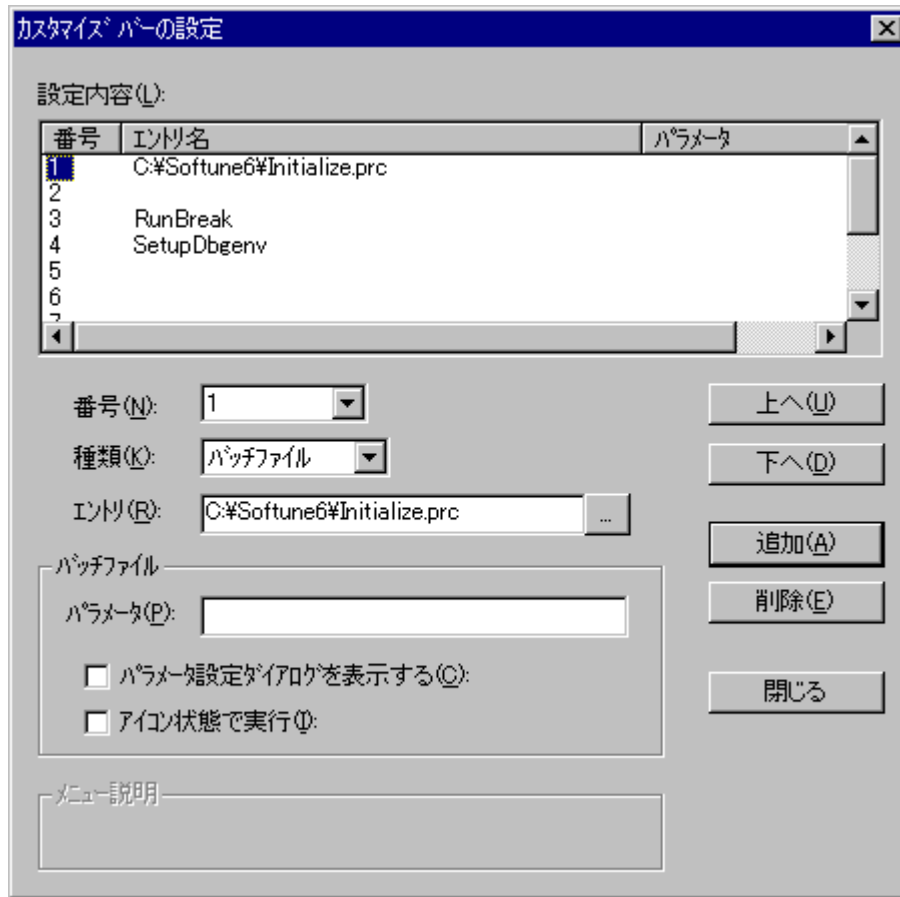
---

#### ■ バッチファイルの登録手順

1. ダイアログの表示  
[ 表示 ] - [ カスタマイズバー ] - [ 設定 ] を選択し『カスタマイズバー設定ダイアログ ( 図 5.1-10 )』を表示します。
2. 番号  
カスタマイズバーに登録する番号を選択します。
3. 種類  
「バッチファイル」を選択します。
4. エントリ  
カスタマイズバーに登録するバッチファイル名を入力します。右の参照ボタンを押すと表示される『ファイルを開くダイアログ ( 図 5.1-5 )』から指定することも可能です。
5. パラメータ  
指定したバッチファイルにパラメータを指定する場合、入力します。複数のパラメータを指定する場合はカンマで区切って入力してください。
6. 『パラメータ設定ダイアログ ( 図 5.1-6 )』を表示する  
カスタマイズバーによるバッチファイル実行時にパラメータを設定したい場合、チェックします。バッチファイルを実行するたびにパラメータを指定 ( 変更 ) したい場合に便利です。
7. アイコン状態で実行  
カスタマイズバーによるバッチファイル実行時に Workbench をアイコン状態にした場合、チェックします。
8. 追加  
2 ~ 7 の入力内容を確認し、正しければ「追加」を押します。

以上でバッチファイルの登録が完了です。登録内容は「設定内容」に表示され、ワークスペース表示時に復元されます。

図 5.1-10 カスタマイズバー設定ダイアログ (バッチファイル登録時)



### 5.1.3.2 Workbench メニューの登録

---

カスタマイズバーに Workbench メニューを登録する方法について説明します。

---

#### ■ Workbench メニューの登録手順

1. ダイアログの表示  
[ 表示 ] - [ カスタマイズバー ] - [ 設定 ] を選択し『カスタマイズバー設定ダイアログ ( 図 5.1-11 )』を表示します。
2. 番号  
カスタマイズバーに登録する番号を選択します。
3. 種類  
「メニュー」を選択します。
4. エントリ  
カスタマイズバーに登録する Workbench メニューを入力します。右の参照ボタンを押すと表示される『メニュー一覧ダイアログ ( 図 5.1-7 )』から指定します。
5. 追加  
2 ～ 4 の入力内容を確認し、正しければ『追加』を押します。

以上で Workbench メニューの登録が完了です。登録内容は『設定内容』に表示され、ワークスペース表示時に復元されます。

図 5.1-11 カスタマイズバー設定ダイアログ (メニュー登録時)



### 5.1.3.3 外部ツールの登録

カスタマイズバーに外部ツールを登録する方法について説明します。

#### ■ 外部ツールの登録手順

1. ダイアログの表示  
[ 表示 ] - [ カスタマイズバー ] - [ 設定 ] を選択し『カスタマイズバー設定ダイアログ ( 図 5.1-12 )』を表示します。
2. 番号  
カスタマイズバーに登録する番号を選択します。
3. 種類  
「ツール」を選択します。
4. エントリ  
カスタマイズバーに登録する外部ツールを入力します。右の参照ボタンを押すと表示される『ツール選択ダイアログ ( 図 5.1-8 )』から指定します。  
ツール選択ダイアログには[ 環境 ] - [ ツールの設定 ] メニューで表示される『ツールの設定ダイアログ ( 図 5.1-9 )』にて設定した内容ツールが表示されます。なお『ツールの設定ダイアログ』は『ツール選択ダイアログ』内の「ツールの設定」ボタンにより表示することも可能です。
5. 追加  
2 ~ 4 の入力内容を確認し、正しければ『追加』を押します。

以上で外部ツールの登録が完了です。登録内容は「設定内容」に表示され、ワークスペース表示時に復元されます。

図 5.1-12 設定ダイアログ - 外部ツール登録時

カスタマイズバーの設定

設定内容(L):

番号	エントリ
1	C:\Softune\Initialize.prc
2	
3	RunBreak
4	SetupDbgen
5	
6	NOTEPAD
7	

番号(N): 6

種類(K): ツール

エントリ(R): NOTEPAD 参照(R)...

ツール

実行ファイル名(F): C:\WINDOWS\system32\notep

オプション(O):

実行時のディレクトリ(P): C:\WINDOWS\system32\

☐ 実行時に追加オプションを指定(I)

☐ アウトワークスペースを使用(W)

上へ(U) 下へ(D) 追加(A) 削除(E) 閉じる



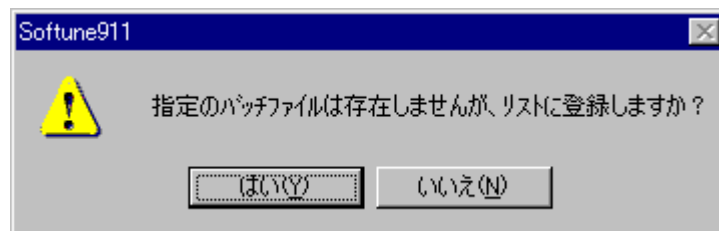
## 5.1.4 警告 / エラーメッセージ

カスタマイズバー使用時に表示される警告 / エラーメッセージについて説明します。

### ■ 警告メッセージ

1. カスタマイズバーにバッチファイルを登録する際（「5.1.3.1 バッチファイルの登録」）、「エントリ」に入力したバッチファイルが存在しないにも関わらず「追加」を押すと、以下の警告メッセージ（図 5.1-13）を表示します。

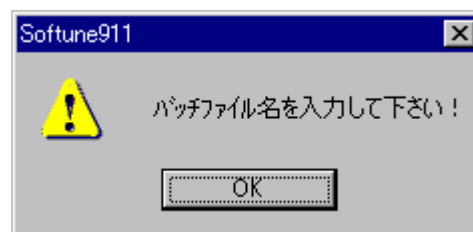
図 5.1-13 警告メッセージ



### ■ エラーメッセージ

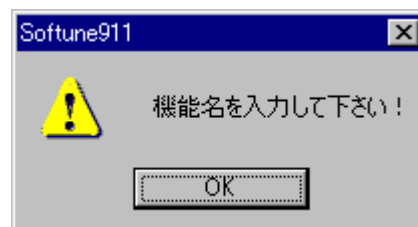
1. カスタマイズバーにバッチファイルを登録する際（「5.1.3.1 バッチファイルの登録」）、「エントリ」にバッチファイルを入力せず「追加」を押すと、以下のエラーメッセージ（図 5.1-14）を表示します。

図 5.1-14 エラーメッセージ 1



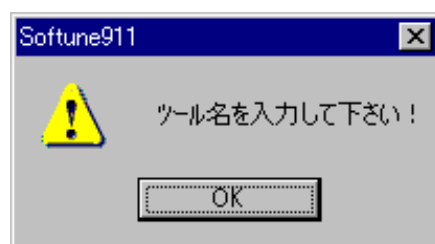
2. カスタマイズバーに Workbench メニューを登録する際（「5.1.3.2 Workbench メニューの登録」）、「エントリ」に機能名を入力せず「追加」を押すと、以下のエラーメッセージ（図 5.1-15）を表示します。

図 5.1-15 エラーメッセージ 2



3. カスタマイズバーに外部ツールを登録する際（「5.1.3.3 外部ツールの登録」）、「エントリ」にツール名を入力せず「追加」を押すと、以下のエラーメッセージ（図 5.1-15）を表示します。

図 5.1-16 エラーメッセージ 3



## 5.1.5 注意事項

---

カスタマイズバー使用時の注意事項について説明します。

---

### ■ 注意事項

1. カスタマイズバーはワークスペース表示時以外は使用できません。ワークスペース表示時に前回の設定内容が復元され、カスタマイズバーのボタンも有効になります。
2. カスタマイズバーの登録内容はツールごとに保存されます。そのためプロジェクトやワークスペース、セットアップファイルを切り替えた場合でも、復元内容は共通となります。
3. カスタマイズバーにバッチファイルを登録する場合、バッチファイル名はカレントディレクトリ（＝プロジェクトディレクトリ）からの相対パス、もしくは絶対パスで入力してください。
4. カスタマイズバーにバッチファイルを登録する場合、複数のパラメータを指定するときは必ずカンマで区切って入力してください。「パラメータ設定ダイアログ（図 5.1-6）」の場合も同様です。
5. カスタマイズバーにバッチファイル、またはデバッガ起動時に有効となる Workbench メニューを登録した場合、デバッガ非起動時はボタンを押しても実行されません。

## 5.2 FLASH ロード

---

**SOFTUNE Workbench の FLASH ロードについて説明します。**

---

5.2.1 FLASH ロードとは

5.2.2 メニュー

5.2.2.1 ターゲットファイルのロード

5.2.2.2 メイク後ロード

5.2.2.3 ビルド後ロード

5.2.2.4 ファイル指定ロード

5.2.2.5 FLASH 消去

5.2.2.6 ロード環境設定

5.2.3 制限事項

5.2.4 エラーメッセージ

## 5.2.1 FLASH ロータとは

---

FLASH ロータの概要について述べます。

---

### ■ FLASH ロータとは

SOFTUNE Workbench 上からターゲット MCU 内蔵の FLASH メモリへダウンロードするプログラムです。

ダウンロード可能なファイルは、プロジェクトのターゲットファイルのほか、モトローラ S フォーマットやインテル HEX、バイナリファイルがあります。

ダウンロードの対象領域は FLASH メモリ全領域のほか、セクタ領域、また複数の連続したセクタ領域も可能です。

## 5.2.2 メニュー

[ FLASH メモリ ] メニューには、ターゲットファイルのロード、メイクまたはビルド後のロード、FLASH メモリの消去などの機能があります。

### ■ [FLASH メモリ] メニュー

FLASH ロードをインストールすると、SOFTUNE Workbench に [ FLASH メモリ ] メニューが追加されます。( 図 5.2-1 参照 )

図 5.2-1 [FLASH メモリ] メニュー



[ FLASH メモリ ] メニューには以下のものがあります。

- ターゲットファイルのロード
- メイク後ロード
- ビルド後ロード
- ファイル指定ロード
- FLASH 消去
- ロード環境設定

### ■ 使用条件

この機能はエミュレータデバグが起動時に、FLASH ロードに対応するターゲット MCU を選択した場合のみ有効になります。ターゲット MCU の設定方法については『SOFTUNE Workbench 操作マニュアル』の「4.5.5.2 MCU 設定」を参照してください。また、FLASH ロードの対応品種については、弊社販売担当部門またはサポート部門までお問い合わせ下さい。

### 5.2.2.1 ターゲットファイルのロード

---

プロジェクトのターゲットファイルを FLASH メモリにダウンロードします。

---

#### ■ 機能

プロジェクトのターゲットファイルを FLASH メモリにダウンロードします。ダウンロードする FLASH メモリ領域はロード環境設定ダイアログ(「5.2.2.6 ロード環境設定」参照)にて設定します。

#### ■ 注意事項

ターゲットファイルは、必ずロードするFLASHメモリ領域に合わせて作成して下さい。FLASH メモリ範囲内のデータのみダウンロードします。

## 5.2.2.2      メイク後ロード

---

プロジェクトのソースプログラムをメイクし、作成されたターゲットファイルを FLASH メモリにダウンロードします。

---

### ■ 機能

プロジェクトのソースプログラムをメイクし、作成されたターゲットファイルを FLASH メモリにダウンロードします。SOFTUNE Workbench のメイク機能については『SOFTUNE Workbench 機能説明書』の「1.4 メイク / ビルド機能」を参照してください。ダウンロードする FLASH メモリ領域はロード環境設定ダイアログ（「5.2.2.6 ロード環境設定」参照）にて設定します。

### ■ 注意事項

ターゲットファイルは、必ずロードするFLASHメモリ領域に合わせて作成して下さい。FLASH メモリ範囲内のデータのみダウンロードします。



### 5.2.2.3 ビルド後ロード

---

プロジェクトのソースプログラムをビルドし、作成されたターゲットファイルを FLASH メモリにダウンロードします。

---

#### ■ 機能

プロジェクトのソースプログラムをビルドし、作成されたターゲットファイルを FLASH メモリにダウンロードします。SOFTUNE Workbench のビルド機能については『SOFTUNE Workbench 機能説明書』の「1.4 メイク / ビルド機能」を参照してください。ダウンロードする FLASH メモリ領域はロード環境設定ダイアログ（「5.2.2.6 ロード環境設定」参照）にて設定します。

#### ■ 注意事項

ターゲットファイルは、必ずロードするFLASHメモリ領域に合わせて作成して下さい。FLASH メモリ範囲内のデータのみダウンロードします。

## 5.2.2.4 ファイル指定ロード

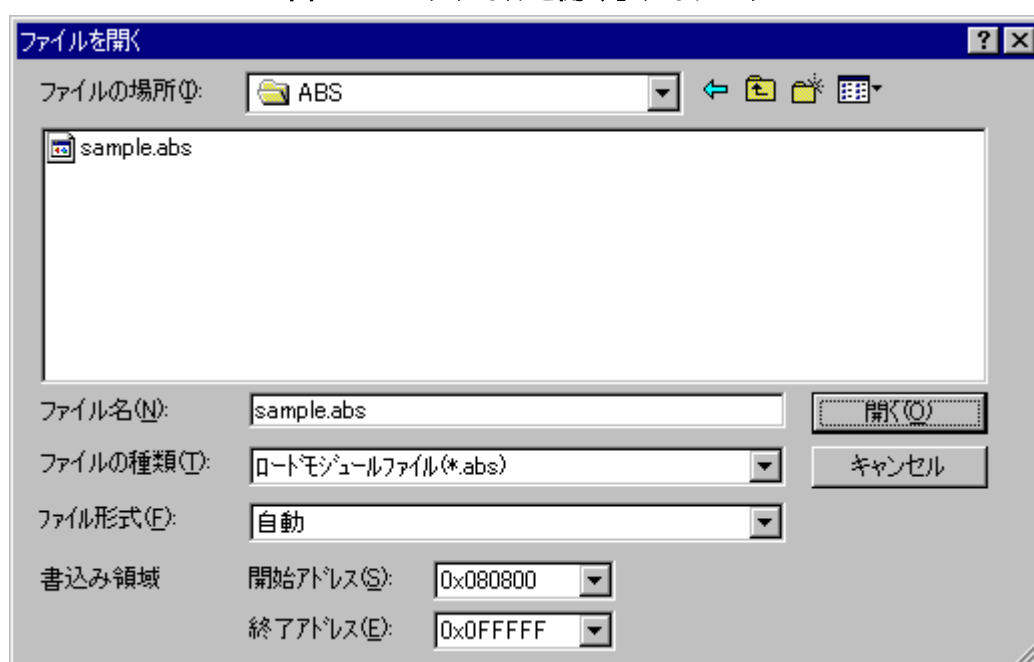
ファイルをダイアログ (図 5.2-2) から選択し, FLASH メモリにダウンロードします。

### ■ 機能

ファイルを指定し, FLASH メモリへダウンロードします。プロジェクトのターゲットファイル以外を FLASH メモリにダウンロードしたい場合に選択します。

[ ファイル指定ロード ] メニューを選択すると, 図 5.2-2 のダイアログが表示されます。

図 5.2-2 「ファイルを開く」ダイアログ



- ファイル名  
選択したファイル名が表示されます。
- ファイルの種類  
拡張子により, 表示するファイルを限定する場合に指定します。
  - ロードモジュールファイル ( \*.abs )  
拡張子が .abs のファイルのみ表示します。
  - モトローラ S フォーマット ( \*.ahx, \*.mhx )  
拡張子が .ahx または .mhx のファイルのみ表示します。
  - インテル HEX ( \*.hex, \*.ihx, \*.ehx )  
拡張子が .hex, .ihx または .ehx のファイルのみ表示します。
  - バイナリファイル ( \*.bin )  
拡張子が .bin のファイルのみ表示します。
  - 全てのファイル  
拡張子に関わらず全てのファイルを表示します。

- ファイル形式

ファイルの拡張子とファイル形式は必ずしも一致しない場合があります。ここでは、選択したファイルの形式を指定します。

- 自動

指定したファイルの拡張子により、下記のようにファイル形式を判別します。

拡張子が .abs の場合：ロードモジュールファイル

拡張子が .ahx または .mhx の場合：モトローラ S フォーマット

拡張子が .hex, .ihx, .ehx のいずれかの場合：インテル HEX

拡張子が .bin の場合：バイナリファイル

拡張子が上記以外の場合：バイナリファイル

- ロードモジュールファイル

指定したファイルの拡張子に関係なく、ファイル形式をロードモジュールファイルとします。

- モトローラ S フォーマットファイル

指定したファイルの拡張子に関係なく、ファイル形式をモトローラ S フォーマットとします。

- インテル HEX ファイル

指定したファイルの拡張子に関係なく、ファイル形式をインテル HEX とします。

- バイナリファイル

指定したファイルの拡張子に関係なく、ファイル形式をバイナリファイルとします。

- 書込み領域

選択したファイルをダウンロードする FLASH メモリ領域を指定します。ロード環境設定ダイアログ（「5.2.2.6 ロード環境設定」参照）にて指定した領域は反映されません。

この情報は、デバッガ再起動時に復元されます。

- 開始アドレス

書込みまたは消去を行う FLASH 領域の開始アドレスを選択します。

- 終了アドレス

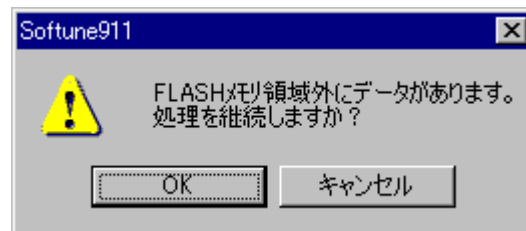
書込みまたは消去を行う FLASH 領域の終了アドレスを選択します。

## ■ 注意事項

ファイル形式にて選択した項目により、以下のようにダウンロードを行います。

- ロードモジュール / モトローラ S フォーマット / インテル HEX ファイルの場合  
ファイル内のデータが指定した FLASH メモリ領域に入っているかどうか自動的にチェックを行います。この際、FLASH メモリ領域から外れていた場合は以下のメッセージダイアログを表示します。(図 5.2-3 参照)

図 5.2-3 メッセージダイアログ



- [ OK ] を選択した場合  
ダウンロードを行います。指定した書き込み領域範囲内のデータのみダウンロードを行います。そのため、ダウンロード後のプログラムが正常に動かない可能性がありますのでご注意ください。
- [ キャンセル ] を選択した場合  
ダウンロードを中止します。
- バイナリファイルの場合  
指定した書き込み領域の開始アドレスからデータをダウンロードします。

## 5.2.2.5 FLASH 消去

---

FLASH メモリを消去します。

---

### ■ 機能

FLASH メモリを消去します。

消去する領域はロード環境設定ダイアログ(「5.2.2.6 ロード環境設定」参照)にて設定した領域です。

## 5.2.2.6 ロード環境設定

FLASH メモリへのダウンロード，または消去の対象となる領域を指定します。

### ■ 機能

FLASH メモリへのダウンロード，または消去を行う場合の範囲を指定します。また，FLASH ロードにて使用する内蔵リソースの領域を表示します。

図 5.2-4 「ロード環境設定」ダイアログ



- MCU
 

現在選択中の MCU を表示します。(変更不可)
- FLASH メモリ領域
 

FLASH メモリに対し，消去または書き込みを行う場合の領域を指定します。

ここで選択した領域は [ ファイル指定ロード ] 機能以外の全ての機能で有効になります。

コンボボックスには FLASH メモリのセクタ領域の開始 / 終了アドレスが全て表示されます。デフォルトは FLASH メモリ全領域が表示されます。

この情報は，デバッガ再起動時に復元されます。

  - 開始アドレス
 

書き込みまたは消去対象となる FLASH メモリの開始アドレスを選択します。
  - 終了アドレス
 

書き込みまたは消去対象となる FLASH メモリの終了アドレスを選択します。

- 内蔵 RAM 領域の指定

FLASH ロードにて使用する内蔵 RAM の領域を表示します。(変更不可)

- DATA 開始アドレス

FLASH メモリヘダウンロードするデータを一時的に格納する領域の先頭アドレスを表示します。

- DATA 終了アドレス

FLASH メモリヘダウンロードするデータを一時的に格納する領域の終了アドレスを表示します。

- CODE 開始アドレス

FLASH メモリヘダウンロードするプログラムを一時的に格納する領域の先頭アドレスを表示します。

- CODE 終了アドレス

FLASH メモリヘダウンロードするプログラムを一時的に格納する領域の終了アドレスを表示します。

## 5.2.3 制限事項

---

FLASH ロードを使用する際には、以下の制限事項があります。

---

### ■ 制限事項

FLASH ロードを使用する際には、以下の制限事項があります。

1. 中間ファイルの作成

プロジェクトのターゲットファイルまたはロードモジュールファイルをダウンロードする場合、中間ファイルとしてモトローラ S フォーマットとバイナリファイルを作成します。一方、モトローラ S フォーマットまたはインテル HEX ファイルをダウンロードする場合は、バイナリファイルのみ作成します。

これらの中間ファイルは指定したファイルと同じディレクトリに作成されますが、削除しません。ファイルのサイズが大きい場合は中間ファイルのサイズにご注意下さい。

2. ブレークポイントの無効化

FLASH ロードによるダウンロード実行時、一時的に設定中のブレークポイント（イベントを含む）を全て無効にします。またダウンロード終了時は設定中のブレークポイントを全て有効にします。この際、あらかじめ無効に設定していたブレークポイントまで有効になりますのでご注意ください。

3. 内蔵 RAM の使用

FLASH ロードによるダウンロード実行時、ロード環境設定ダイアログ（「5.2.2.6 ロード環境設定」参照）にて表示される領域を使用します。このため、重要なデータやプログラムが置かれている場合は上書きされますのでご注意ください。

4. レジスタの使用

FLASH ロードによるダウンロード実行時、各レジスタを使用します。このため、重要なデータが置かれている場合はご注意ください。

5. 動作環境の初期化

FLASH ロードによるダウンロード開始時、最適な動作環境に自動的に変更します。そのため、動作環境を設定している場合はダウンロード終了後、再度設定を行う必要があります。

6. 環境変数の設定

FLASH ロードによりダウンロードするファイルのフルパスに、以下の文字列を使用しないでください。ファイルが正常にダウンロードできない場合があります。

MODR, GCR, PCTR, FWTC, FSTR, FETOOL, CMDADRa1, CMDADRa2, FSADR, FEADR, DRAM, SIZE, IADR, FILEOFFSET, CNTNUEFLG, QUITFLG



## 5.2.4 エラーメッセージ

---

FLASH ロード使用時に表示されるエラーメッセージについて述べます。

---

### ■ エラーメッセージ

指定されたファイルは存在しません。

【内容】 指定されたファイルが見つかりませんでした。

【対処】 指定したディレクトリ内のファイルを確認してください。

指定された領域は FLASH メモリ領域外です。

【内容】 指定したセクタアドレスは FLASH メモリの領域外です。

【対処】 FLASH ロードの情報ファイルのバージョンが古い可能性があります。  
SOFTUNE Workbench の最新版を再インストールしてください。

モトローラ S フォーマットファイルが存在しません。

【内容】 プロジェクトのターゲットファイルがあるディレクトリ内にモトローラ S フォーマットファイルがありません。

【対処】 ロードモジュールファイルからモトローラ S フォーマットへの変換に失敗した可能性があります。SOFTUNE インストールディレクトリの BIN の下に f2ms.exe が存在するか確認してください。

バイナリファイルが存在しません。

【内容】 プロジェクトのターゲットファイルがあるディレクトリ内に、ターゲットファイルと同名のバイナリファイルがありません。

【対処】 モトローラ S フォーマットまたはインテル HEX からバイナリファイルへの変換に失敗した可能性があります。SOFTUNE インストールディレクトリの BIN の下に m2bs.exe または h2bs.exe が存在するか確認してください。

フラッシュメモリの消去に失敗しました。

【内容】 FLASH メモリの消去に失敗しました。

【対処】 FLASH メモリに異常がないか確認してください。

フラッシュメモリの書き込みに失敗しました。

【内容】 FLASH メモリの書き込みに失敗しました。

【対処】 FLASH メモリに異常がないか確認してください。

ファイルがアクセスできません。

【内容】 指定したファイルがアクセスできません。

【対処】 指定したファイルのアクセス権がないことが考えられます。

ファイルの形式が違います。

【内容】 指定したモトローラ S フォーマットまたはインテル HEX ファイルの形式が違います。

【対処】 ファイルを正しい形式で作成してください。

アドレスの指定に誤りがあります。

【内容】 FLASH セクタアドレスの指定に誤りがあります。

【対処】 指定した FLASH セクタアドレスの終了アドレスより開始アドレスの方が大きい値になっています。正しいアドレス範囲を設定してください。

メモリベリファイエラーが発生しました。

【内容】 FLASH 書き込み中にメモリベリファイエラーが発生しました。

【対処】 FLASH メモリが実装されているか、FLASH メモリに異常がないかを確認してください。

ターゲットファイルがオープンできません。

【内容】 プロジェクトのターゲットファイルがオープンできません。

【対処】 プロジェクトのターゲットファイルが作成されていない可能性があります。ターゲットファイルを作成してください。

付録では、レジスタ名一覧、モニタプログラムのダウンロード、LAN インタフェースの設定、USB インタフェースの設定、モニタデバッガターゲット ROM 部の作成、エミュレータ本体の表示シミュレータ外部 I/F を記載します。

付録 A レジスタ名一覧

付録 B モニタプログラムのダウンロード

付録 C LAN インタフェースの設定

付録 D USB インタフェースの設定

付録 E モニタデバッガターゲット ROM 部の作成

付録 F エミュレータ本体の表示

付録 G シミュレータ外部 I/F

## 付録 A レジスタ名一覧

---

レジスタ名の一覧を表示します。

---

### ■ レジスタ

SOFTUNE Workbench で操作できるレジスタは、MCU により異なります。以下に一覧を表示します。

汎用レジスタ	:	R0 ~ R15
プログラムカウンタ	:	PC
プロセッサステータス	:	PS
テーブルベースレジスタ	:	TBR
リターンポインタ	:	RP
システムスタックポインタ	:	SSP
ユーザスタックポインタ	:	USP
乗除算結果レジスタ	:	MDH, MDL
フラグ	:	CCR, S, I, N, Z, V, C
システムコンディション	:	SCR, D1, D0, T
割込みレベル	:	ILM
仮想アキュムレータ	:	AC
フレームポインタ	:	FP
スタックポインタ	:	SP

これらのレジスタ名は、アドレス指定時などに % を前置きする形で用いることができます。

---

#### < 注意事項 >

システムコンディションの T フラグはエミュレータで使用しています。  
レジスタコマンドで値の変更を行っても無効になります。

---

## 付録 B モニタプログラムのダウンロード

エミュレータデバッガを使用するには、ご使用になるチップに応じたモニタプログラムをエミュレータに書き込む必要があります。

この作業をモニタプログラムのダウンロードといいます。

MB2198 エミュレータでは、デバッグ開始時にモニタプログラムの種類とバージョンを確認して、自動的にダウンロードを行うことができます。

### ■ エミュレータ本体が MB2197 の場合

モニタプログラムのダウンロードは、以下の手順で行います。

- 1) エミュレータとパソコンを RS-232C または LAN で接続します。
  - LAN で接続する場合は、付録の「付録 C LAN インタフェースの設定」を参照してください。
- 2) エミュレータ本体のリセットスイッチを押しながら、エミュレータの電源を ON にします。
  - エミュレータ本体の READY の LED が点灯していることを確認してください。
- 3) Windows スタートメニューの [SOFTUNE V6] - [FR Family SOFTUNE Workbench Tool] にある、[Monitor Loader] メニューを実行します。
  - モニタロードプログラムが起動します。
- 4) ロードするモニタプログラムを選択します。
  - 使用するチップに応じたモニタプログラムを選択してください。
- 5) 通信種別を指定します。
  - RS-232C を使用する場合は、通信ポートとボーレートを指定します。
  - LAN を使用する場合は、エミュレータのホスト名を指定します。
- 6) [ロードの開始] をクリックします。
  - これで、モニタプログラムがエミュレータにロードされます。
- 7) [ファイル] メニュー [終了] を選択して、モニタロードプログラムを終了してください。

付表 B-1 チップに応じたモニタプログラム

チップ種別	対応するチップ	モニタプログラム
FR30 (DSU1 タイプ)	MB91171 MB91172 MB91173	20DSU1.HEX
FR30 (DSU2 タイプ)	MB91V101 MB91174 MB91191	20DSU2.HEX
FR30 (DSU3 タイプ)	MB91110	20DSU3.HEX

## ■ エミュレータ本体が MB2198 の場合

モニタプログラムのダウンロードは、以下の手順で行います。

3-a) ~ 5-a) はモニタプログラムの自動ロード設定の手順を、3-b) ~ 5-b) はモニタローダを使用した手順を記しました。

- 1) エミュレータとパソコンを RS-232C, LAN または USB で接続します。
  - LAN で接続する場合は、付録の「付録 C LAN インタフェースの設定」を参照してください。
- 2) エミュレータ本体のリセットスイッチを押しながら、エミュレータの電源を ON にします。
  - エミュレータ本体の READY の LED が点灯していることを確認してください。
- 3-a) Windows スタートメニューの [SOFTUNE V6] - [FR Family SOFTUNE Workbench] メニューを実行します。
  - SOFTUNE Workbench が起動します。
- 4-a) ワークスペースおよびプロジェクトを作成し、セットアップウィザードを起動します。
  - ICE 種別で MB2198 を選別し、[次へ] ボタンをクリックするとモニタプログラムの自動ロードチェックボックスが表示されますので、チェックしてください。
- 5-a) [デバッグ] メニュー [デバッグの開始] を選択してエミュレータデバッグを起動します。
  - "¥ インストールディレクトリ ¥LIB¥911¥" にあるモニタプログラムを自動的にダウンロードします。
  - これで、デバッグ開始時にモニタプログラムがエミュレータに自動ロードされます。
  - ダウンロードが終わるとメッセージダイアログ ("ロードが正常に終了しました。") を表示しますので "OK" ボタンを押すと、デバッグを開始します。
- 3-b) Windows スタートメニューの [SOFTUNE V6] - [FR Family SOFTUNE Workbench Tool] にある、[Monitor Loader] メニューを実行します。
  - モニタローダプログラムが起動します。
- 4-b) ロードするモニタプログラムを選択します。
  - 使用するチップに応じたモニタプログラムを選択してください。
- 5-b) 通信種別を指定します。
  - RS-232C を使用する場合は、通信ポートとボーレートを指定します。
  - LAN を使用する場合は、エミュレータのホスト名を指定します。
- 6-b) [ロードの開始] をクリックします。
  - これで、モニタプログラムがエミュレータにロードされます。
- 7-b) [ファイル] メニュー [終了] を選択して、モニタローダプログラムを終了してください。

---

### < 注意事項 >

MB2198 エミュレータでは、モニタプログラムの数が増えています。モニタプログラムに関する詳細については、CD のルートディレクトリにあるリリースノートをご参照ください。

---

## 付録 C LAN インタフェースの設定

---

LAN で通信を行うには、パソコン側の設定とエミュレータ側の設定を行う必要があります。IP アドレスやポートアドレスなどは、LAN の管理者に相談して設定してください。

---

### ■ パソコン側の設定

- 1) TCP/IP プロトコルをインストールします。
  - 「コントロールパネル」の「ネットワークとダイヤルアップ接続」「ローカルエリア接続」の「プロパティ」で行います。
- 2) エミュレータに割り付けた IP アドレスを HOSTS ファイルに追加します。追加する内容は、以下のとおりです。
  - IP アドレス      ホスト名
    - SYSTEM32¥DRIVERS¥ETC に存在します。アドミニストレータの権限のあるユーザで設定を行ってください。
- 3) エミュレータのポートアドレス、サービス名の登録を SERVICES ファイルに行います。デフォルトでは、サポートアドレス 5001 とサービス名 fjicesv となっています。追加する内容は、以下のとおりです。
  - fjicesv              5001/tcp
    - SYSTEM32¥DRIVERS¥ETC に存在します。アドミニストレータの権限のあるユーザで設定を行ってください。

## ■ エミュレータ側の設定

- エミュレータが MB2197 の場合、以下の手順で行います。

- 1) エミュレータとパソコンを RS-232C で接続します。
- 2) エミュレータの電源を ON にします。
- 3) スタートメニューの [SOFTUNE V6] - [FR Family SOFTUNE Workbench Tool] にある [LAN Address] メニューを実行します。
  - LAN アドレス設定プログラムが起動します。
- 4) [通信の設定] をクリックして、RS-232C の設定を行います。
- 5) [読み出し] をクリックして、現在のエミュレータ設定状態を読み出します。
- 6) IP Address および Port Address を設定します。それぞれ、パソコン側の設定した値を示します。
- 7) Mac Address は、通常 Universal を使用します。Local を使用する場合は、LAN アドレスを管理している管理者に相談の上、ご使用ください。
- 8) [設定] メニュー [終了] を選択して、LAN アドレス設定プログラムを終了してください。
- 9) エミュレータのリセットボタン（背面にあるボタン）を押して、エミュレータをリセットしてください。

- エミュレータが MB2198 の場合、以下の手順で行います。

- 1) エミュレータとパソコンを RS-232C または USB で接続します。
- 2) エミュレータの電源を ON にします。
- 3) スタートメニューの [SOFTUNE V6] - [FR Family SOFTUNE Workbench Tool] にある [LAN Address] メニューを実行します。
  - LAN アドレス設定プログラムが起動します。
- 4) [通信の設定] をクリックして、RS-232C または USB の設定を行います。
- 5) [読み出し] をクリックして、現在のエミュレータ設定状態を読み出します。
- 6) IP Address, サブネットマスクおよび Port Address を設定します。それぞれ、パソコン側で設定した値を示します。
- 7) Mac Address は、通常 Universal を使用します。Local を使用する場合は、LAN アドレスを管理している管理者に相談の上、ご使用ください。
- 8) [設定] メニュー [終了] を選択して、LAN アドレス設定プログラムを終了してください。
- 9) エミュレータのリセットボタン（背面にあるボタン）を押して、エミュレータをリセットしてください。



---

**< 注意事項 >**

MB2198 エミュレータでは、エミュレータ本体の状態表示 LCD と 3 個の設定スイッチを使用することでも、IP アドレス、サブネットマスクと Port アドレスを設定することができます。入力手順は次のとおりです。

- 1) エミュレータ本体の ENTER キーを押下して、パラメータ入力モードに切替えます。
  - 2) 上下キーで "LAN" を選んで、ENTER キーを入力します。
  - 3) 上下キーで "IP Address" を選んで、ENTER キーを入力します。データの入力は 1 桁ずつ行い、上下キーで桁を選択して ENTER キーで決定し、上下キーで数値を切り替え ENTER キーで決定します。入力の終了は、最下位アドレス入力後、上キーで "OK" にカーソルを合わせ、ENTER キー入力で行います。
  - 4) 上下キーで "Subnet Mask" を選んで、ENTER キーを入力します。  
データの入力手順は IP アドレスの入力と同様です。
  - 5) 上下キーで "Port Address" を選んで、ENTER キーを入力します。  
データの入力手順は IP アドレスの入力と同様です。
  - 6) 上下キーで "Exit" を選んで、ENTER キーを入力します。
-

## 付録 D USB インタフェースの設定

---

**USB で通信を行うには、パソコンに USB ドライバをインストール必要があります。**

---

### ■ USB ドライバのインストール

USB ドライバをインストールすることにより、エミュレータ (MB2198) とパソコンを USB で接続できます。

USB ドライバは、次の手順でインストールできます。

- SOFTUNE Workbench V60L06 以降の場合  
SOFTUNE インストール時に USB ドライバも自動的にインストールします。
- SOFTUNE Workbench V60L05 以前の場合
  - 1.エミュレータとパソコンを USB ケーブルで接続します。
  - 2.エミュレータの電源を ON にすると、OS が USB ドライバのインストールを要求しますので、SOFTUNE インストールディレクトリの下の Drivers を指定してください。

---

#### < 注意事項 >

Windows Vista のユーザアカウント (UAC) 機能について

- UAC機能が有効になっている場合、USBドライバは自動的にインストールされません。このため SOFTUNE Workbench インストール時は UAC 機能を無効にいただく必要があります。UAC 機能を無効にするには [ コントロールパネル ]-[ ユーザアカウント ] の [ ユーザアカウント制御の有効化または無効化 ] において [ ユーザアカウント制御 (UAC) を使ってコンピュータの保護に役立たせる ] のチェックを外してください。

UAC 機能はウイルスや不正な操作、操作ミスなどによって、管理者権限が必要なプログラムなどが自動的にシステムにインストールされてしまうのを防ぐための機能です。よって SOFTUNE のインストール後は、再度有効にしておくことをお勧めします。

---

## 付録 E モニタデバッガターゲット ROM 部の作成

ここでは、モニタデバッガターゲット ROM 部の作成方法について説明します。

- ターゲットシステムの構成
- ターゲットシステムの作成手順
- サンプルプログラムの説明

### ■ ターゲットシステムの作成

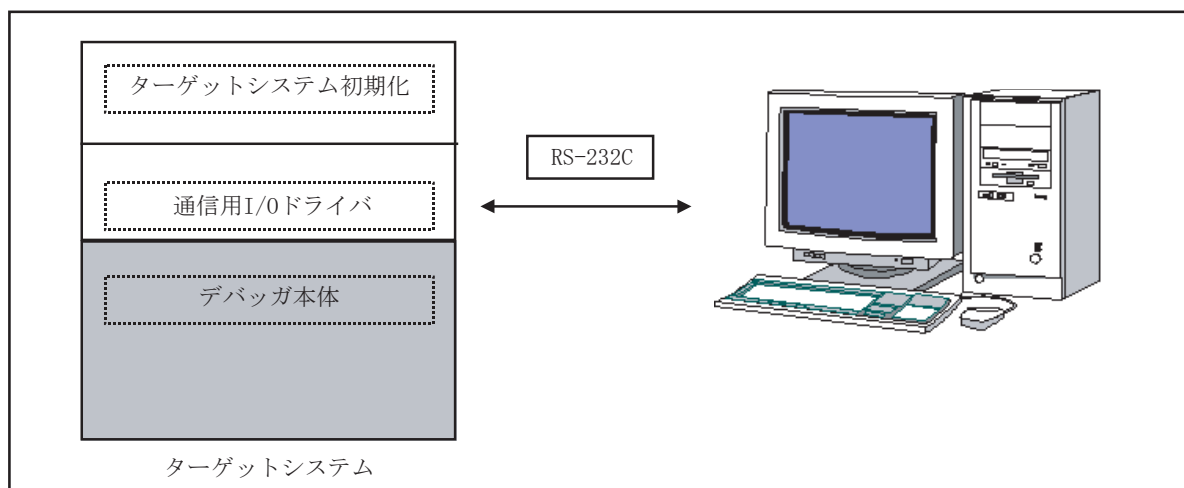
モニタデバッガを使用するためには、デバッガ本体(mod911.rel) のほかに次に示すものを追加してターゲットシステムを作成する必要があります。

- ターゲットシステムの初期化ルーチン
- ホストシステムと通信するための I/O ドライバ
- ベクタテーブル

### ■ ターゲットシステムの構成

付図 E-1 にモニタデバッガの構成を示します。

付図 E-1 モニタデバッガの構成



### ■ ホストシステムとの通信について

ターゲットシステムはホストシステムとの通信を RS-232Cで行います。このため、ターゲットシステムに、通信ハードウェアを用意する必要があります。

### ■ プログラムの中断 (ABORT スイッチ) について

プログラムの強制中断は、ユーザ NMI を使います。強制中断機能を使う場合は、ターゲットハードウェアに NMI スイッチを付けてください。

## ■ ターゲットシステムの作成手順

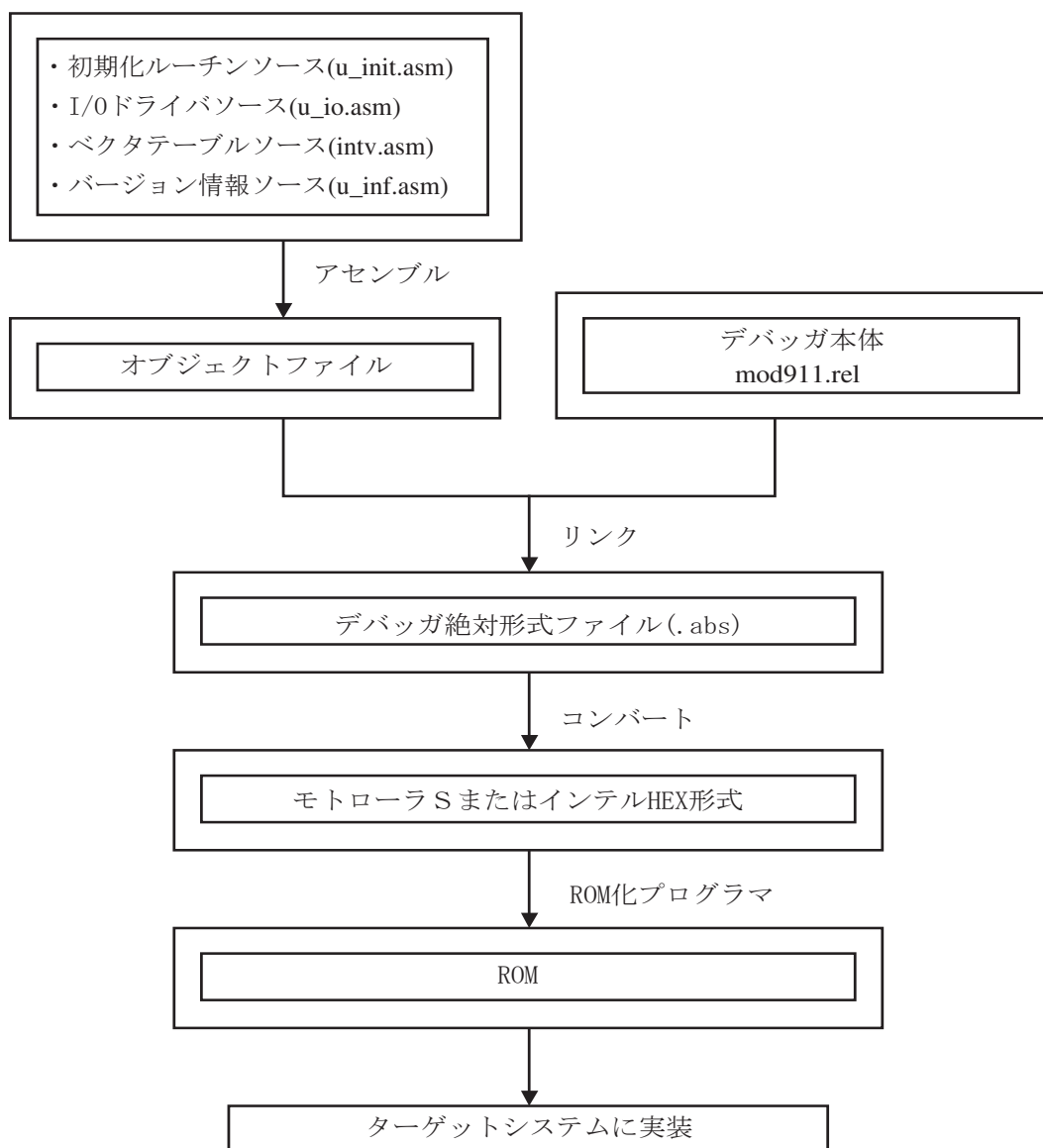
ターゲットシステムの作成手順を以下に示します。また、作成手順フローを付図 E-2 に示します。

- 1) 初期化ルーチン、I/O ドライバ、ベクタテーブルを使用するハードウェアの仕様に合わせて作成し、アセンブルします。
- 2) 1) で生成したオブジェクトファイルとデバッガ本体 (mod911.rel) とをリンクします。
- 3) 1) で生成したデバッガ絶対形式ファイルをモトローラ S またはインテル HEX 形式にコンバートします。
- 4) 3) で生成したモトローラ S またはインテル HEX ファイルを ROM ライタに転送し、ROM 化します。

モニタデバッガ本体 mod911.rel は、以下にあります。

"¥インストールディレクトリ¥LIB¥911¥"

付図 E-2 ターゲットシステムの作成手順フロー



## ■ サンプルプログラムの説明

初期化ルーチン、I/O ドライバ、ベクタテーブルは、サンプルとして以下のファイルを提供しています。

- train.inc           CPU の I/O アドレスなどの定義
- u\_init.asm        初期化 (通信, タイマ), スタック領域設定
- u\_io.asm          ドライバ (通信, タイマ)
- intv.asm         割込みベクタテーブル
- link.opt         リンカへの入力ファイル
- flag911.inc      リソース選択フラグの設定等
- mod911.rel      モニタ本体

## ■ 提供サンプルファイル

ユーザがターゲットシステムに組み込むためのサンプルとして、以下のファイルを提供していますので参考にしてください。

組み込む場合は、これらのファイルを必要に応じて書き換えて作成することをお勧めします。

なお、デバッガ本体から外部参照しているため、シンボル名が固定されている場合があります。そのようなシンボルについては、以下の説明中では名前の後ろに (\*) をつけていますので、シンボル名を変更せずにそのままの名前で使用してください。

### ● train.inc

u\_io.asm などで使用しているラベルの定義ファイルです。I/O アドレスなどを定義しています。

### ● u\_init.asm

- u\_init (\*)
  - デバッガの使用する通信、タイマなどの初期化を行います。Abort およびタイマ処理を使用する場合は、それぞれ \_abtflg (\*), \_timflg (\*) の 1 バイトのフラグに "1" を設定してください。使用しない場合は "0" を設定してください。

### ● u\_io.asm

- inithrd (\*)
  - ターゲットシステムの初期化を行います。CPU の初期設定 (メモリインタフェースの設定など) を行います。

---

### < 注意事項 >

デバッグするユーザプログラム中ではこれらの設定は行わないようにしてください。ユーザプログラムにより、再設定されるとデバッガが正常に動作しない場合があります。

---

- initrs
  - 通信用のポートの初期化を行います。受信、送信はポーリング制御です。
- \_\_clr\_abrt (\*)
  - Abort スイッチがある場合にソフトウェアでの要因クリアが必要なときの処理を

ここに記述します。

- `_inittime`
  - 実行時間の測定を行う機能 ("SHOW TIMER" コマンド) を使う場合に測定用のタイマの初期化を行います。使用しない場合は "RET" のみを記述してください。
- `__clrtim (*)`
  - 測定用のタイマをクリアしてスタートします。タイマ機能を使用しない場合は , "RET" のみを記述してください。
- `__readtim (*)`
  - 測定用のタイマを読み出します。タイマ機能を使用しない場合は , "RET" のみを記述してください。
- `_getchr (*)`
  - 1 バイトの受信データをデバッガに渡します。
- `_putchr (*)`
  - デバッガから渡された 1 バイトデータを送信します。

● `intv.asm`

デバッガの使用する割込みベクタテーブルです。デバッガはレジスタ TBR の値が , "FFC00" で動作しています。

● `u_inf.asm`

- `user_version_information (*)`
  - SOFTUNE Workbench のバージョン情報に表示するユーザ固有の文字列を定義します。V50L02 以上で利用可能です。31 文字まで定義できます。文字列の終端に "¥0 (null) " をいれてください。表示の必要がない場合も "¥0 (null) " のみいれてください。

● `link.opt`

サンプルプログラムのオブジェクトとデバッガ本体 (mod911.rel) をリンクするとき使用する , リンカへの入力ファイルのサンプルです。

---

< 注意事項 >

"STACK" と "DATA" セクション領域は , この順序で必ずリンクしてください。  
デバッガがこの領域をゼロクリアしています。

---

## ■ モニタデバッガが制御できない場合の処置

モニタデバッガが制御できない(ホストシステムとターゲットの通信ができない)場合があります。通信できない原因として、以下のことが考えられます。

- 連続実行中(デバッグ状態が Execute のとき)
- ターゲットプログラムが暴走しているとき

このときは以下の手順で、再度デバッガを起動しなおしてください。

- 1) 「デバッグ (D)」 - 「停止 (A)」を選択します。(もしくは実行の停止ボタンを押します。)
- 2) 中断ダイアログの「中断 (A)」を押してください。
  - このとき MCU をリセットすることはできません。
  - 何度か表示されますが、無視してください。
- 3) 「Abort が実行できません。」の警告がでたら「OK」を押してください。
- 4) デバッガを終了してターゲットシステムをリセットします。
- 5) 再度デバッガを起動してください。

## 付録 F エミュレータ本体の表示

---

MB2198 エミュレータでは、エミュレータ本体の状態を示す LCD ( 液晶 ) が実装されています。

---

### ■ 通常動作時の LCD 表示

通常動作時には、次の文字列が表示されます。

#### ● デバッガ起動前

NO CONNECT \*

[Enter] :Menu

"NO CONNECT" で、接続しているホストがない状態を示します。

"\*" の位置には、コマンド受付中を示す回転アニメーションが表示されます。

" [Enter] :" の行では、Enter ボタンを押した時の動作を示します。

#### ● デバッガ起動後

STATUS Vol:?? \*

[Enter] :Menu

"STATUS" の行では、"??" の位置にターゲットのステータスを表示します。

表示される文字列は、"EXEC" または "BREAK" です。

"\*" の位置には、コマンド受付中を示す回転アニメーションが表示されます。

" [Enter] :" の行では、Enter ボタンを押した時の動作を示します。

### ■ エミュレータデバッガ実行中のエラー表示

いずれのエラーが LCD に表示された場合でも、デバッグを終了してからエミュレータの電源を再投入してください。RS-232C 通信エラーの場合には、ケーブル内の断線も考えられますのでご確認ください。

#### ● RS-232C 通信エラー

RS232C error.

Check Parameter.

#### ● 不法割込みエラー

System error.

Illegal INT

#### ● REALOS で異常を検出

SYSTEM DOWN

Please RESET



## ■ エミュレータ本体へモニタプログラムをロードした場合のエラー表示

いずれのエラーが LCD に表示された場合でも、エミュレータの電源を再投入し、再度モニタプログラムのロードを行ってください。

### ● フラッシュ消去エラー

erase error.

Call to FUJITSU

### ● フラッシュ書込みエラー

WRITE error.

Call to FUJITSU

### ● 起動時の RAM チェックでエラー発生

RAM error.

Call to FUJITSU.

## 付録 G シミュレータ外部 I/F

高速版シミュレータデバッガには、周辺シミュレーションモジュールを作成するための I/F を用意しています。

ここではその I/F について説明します。

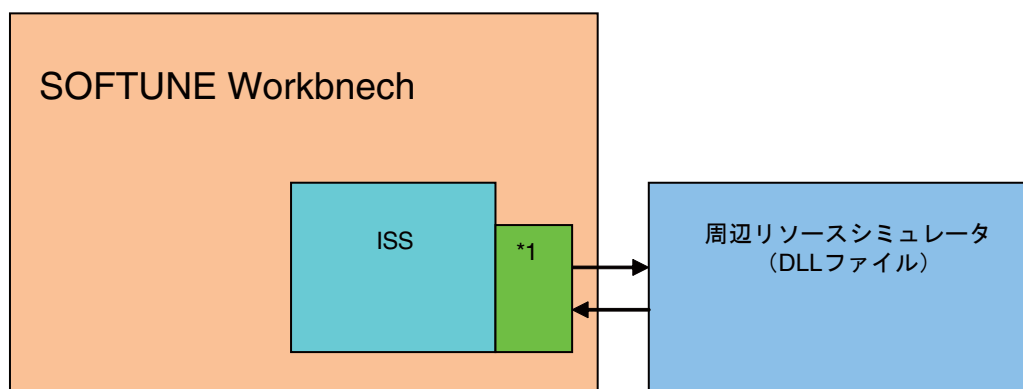
### ■ 外部 I/F DLL の概要

SOFTUNE Workbench のシミュレータデバッガでは、周辺リソースを使用するアプリケーションのデバッグを支援するために I/O シミュレーション機能（ポート / 割込み）をサポートしています。

ただし、簡易的な機能であり、複雑な周辺リソースのシミュレーションには対応できないため、ISS（命令セットシミュレータ）とユーザが C 言語で記述した周辺リソースシミュレータを結合するインターフェースをサポートします。

この周辺リソースシミュレータは、Windows の DLL（ダイナミック・リンク・ライブラリ）ファイルで作成します。

### ■ 構成



\*1：外部I/Fモジュール

周辺リソースシミュレータは、SOFTUNE Workbench のシミュレータデバッガ起動時に読み込まれます。

ISS（命令セットシミュレータ）と周辺リソースシミュレータとの連携は、以下のタイミングで呼び出します。

- デバッガ起動
- デバッガ初期化完了
- デバッガ終了
- ターゲットリセット
- メモリ（I/O）リード直前
- メモリ（I/O）ライト直後
- フェッチ動作直前
- 割込み発生
- タイマイベント発生

周辺リソースシミュレータから ISS ( 命令セットシミュレータ ) を操作する機能は、以下の機能があります。

- メモリリード / ライト
- レジスタリード / ライト
- 割込み要因設定
- 命令実行の中断要求

## ■ シミュレータ外部 I/F 仕様

### 【関数一覧】

#### (1) ISS    DLL

- (1) SSDI\_Entry ( 起動 )
- (2) SSDI\_Init ( 初期化 )
- (3) SSDI\_Ready ( レディ )
- (4) SSDI\_End ( 終了 )
- (5) SSDI\_Reset\_Event ( リセット通知 )
- (6) SSDI\_Read\_Event ( リード通知 )
- (7) SSDI\_Write\_Event ( ライト通知 )
- (8) SSDI\_Execute\_Event ( 実行命令通知 )
- (9) SSDI\_Interrupt\_Event ( 割込み通知 )
- (10) SSDI\_Timer\_Event ( タイマ通知 )

#### (2) DLL    ISS

- (11) SSDI\_Read\_Memory ( メモリリード )
- (12) SSDI\_Write\_Memory ( メモリライト )
- (13) SSDI\_Read\_Register ( レジスタリード )
- (14) SSDI\_Write\_Register ( レジスタライト )
- (15) SSDI\_Set\_Interrupt ( 割込み要因設定 )
- (16) SSDI\_Set\_Timer ( タイマ設定 )
- (17) SSDI\_Request\_Abort ( 中断要求 )
- (18) SSDI\_Set\_Area ( 領域設定 )
- (19) SSDI\_Clear\_Cycle ( サイクルクリア )
- (20) SSDI\_Execute\_MCU ( プログラム実行 )
- (21) SSDI\_GetCurrent Execute Addr ( 実行アドレス取得 )

## G.1 SSDI\_Entry ( 起動 )

---

### 【書式】

```
int SSDI_Entry(int id, HINSTANCE hInstance)
```

### 【引数】

int	id;	// DLL ID
HINSTANCE	hInstance	// 外部 I/F モジュールのハンドル

### 【復帰値】

NORMAL(0) を返してください。

NORMAL(0) 以外の場合、デバugg起動に失敗します。

### 【説明】

デバugg起動時に呼び出しを行います。

必要な初期化を行ってください。

### 【備考】

DLL ID ( id ) は常に 0 が渡されます。

外部 I/F モジュールのハンドル ( hInstance ) は、ISS のエントリ取得などに使用します。

この時点では、外部 I/F モジュールの呼び出しは行えません。

例 )

```
typedef int (WINAPI *LPFNREADMEMORY)();
LPFNREADMEMORY Read_Memory;
extern "C" int WINAPI SSDI_Entry(int id, HINSTANCE if)
{
    int my_dll_id = id;
    FARPROC pF_Read_Memory = ::GetProcAddress(if, "SSDI_Read_Memory");
    Read_Memory = (LPFNREADMEMORY)pF_Read_Memory;
    :
}
int func()
{
    if (ReadMemory(addr, size, len, data) != len)
        error();
    :
}
```

## G.2 SSDI\_Init ( 初期化 )

---

**【書式】**

```
int SSDI_Init(void)
```

**【引数】**

なし

**【復帰値】**

NORMAL(0) を返してください。

NORMAL(0) 以外の場合 , デバッガ起動に失敗します。

**【説明】**

デバッガ起動時に呼び出しを行います。

必要な初期化を行ってください。

**【備考】**

この時点から外部 I/F モジュールの呼び出しが可能となります。

## G.3 SSDI\_Ready (レディ)

---

【書式】

```
int SSDI_Ready(void)
```

【引数】

なし

【復帰値】

NORMAL(0) を返してください。

NORMAL(0) 以外の場合、デバッガ起動に失敗します。

【説明】

デバッガ起動が完了した時に呼び出しを行います。

【備考】

デバッガ開始時バッチファイル実行およびデバッガ開始時の自動ロードを行った後に呼び出されます。

## G.4 SSDI\_End ( 終了 )

---

**【書式】**

void SSDI\_End(void)

**【引数】**

なし

**【復帰値】**

なし

**【説明】**

デバッガ終了時に呼び出しを行います。  
必要な終了処理を行ってください。

**【備考】**

## G.5 SSDI\_Reset\_Event (リセット通知)

---

【書式】

```
void SSDI_Reset_Event(void)
```

【引数】

なし

【復帰値】

なし

【説明】

デバッガのコマンドによるリセットまたは、リセットが発行されたことを通知します。  
リソースの初期化などを行います。

【備考】

すべての割込み状態を OFF にします。



## G.6 SSDI\_Read\_Event (リード通知)

---

### 【書式】

```
int SSDI_Read_Event(READ_EVENT *info)
```

### 【引数】

```
typedef struct {
    unsigned long    addr;           // アクセスアドレス
    int              size;           // アクセスサイズ (1:Byte / 2:Word / 4:LONG)
    unsigned long    total_cycle;    // トータルサイクル数
    unsigned long    inst_cycle;     // 命令先頭からのサイクル数
    unsigned long    *data;          // 読み込みデータ
    unsigned long    *cycle;         // アクセスにかかるサイクル数
} READ_EVENT;
```

### 【復帰値】

```
=0          読み込みデータ有効
!=0         読み込みデータ無効
```

### 【説明】

命令実行により、リードアクセスイベントが発生したことを通知します。

読み込みデータ有効の場合は、読み込みデータ( info.data )を読み込みデータとして ISS は動作します。読み込みデータを ISS のメモリ上に反映する場合は、SSDI\_Write\_Memory() を使用してデータを書込む必要があります。

読み込みデータ無効の場合は、ISS のメモリヘデータリード動作を行います。

アクセスにかかるサイクル数( info.cycle )は、リードアクセスに対してかかるサイクル数を設定してください。ISS では、この値をサイクル数に加えます。

このアクセスにより、異常がある場合は、SSDI\_Request\_Abort() を使用して命令実行を停止してください。

### 【備考】

命令先頭からのサイクル数( inst\_cycle )は、常に 0 を通知します。

## G.7 SSDI\_Write\_Event (ライト通知)

---

### 【書式】

```
int SSDI_Write_Event(WRITE_EVENT *info)
```

### 【引数】

```
typedef struct {
    unsigned long    addr;           // アクセスアドレス
    int              size;           // アクセスサイズ (1:Byte / 2:Word / 4:LONG)
    unsigned long    data;           // 書き込みデータ
    unsigned long    total_cycle;    // トータルサイクル数
    unsigned long    inst_cycle;     // 命令先頭からのサイクル数
    unsigned long    *cycle;         // アクセスにかかるサイクル数
} WRITE_EVENT;
```

### 【復帰値】

```
=0          書き込みデータ有効
!=0         書き込みデータ無効
```

### 【説明】

命令実行により、ライトアクセスイベントが発生したことを通知します。

書き込みデータ有効の場合は、ISS はメモリへの書き込み動作を行いません。そのため、ISS のメモリへ反映する場合は、SSDI\_Write\_Memory() を使用してデータを書込む必要があります。

書き込みデータ無効の場合は、ISS のメモリへデータ書き込み動作を行います。

アクセスにかかるサイクル数 ( info.cycle ) は、ライトアクセスに対してかかるサイクル数を設定してください。ISS では、この値をサイクル数に加えます。

このアクセスにより、異常がある場合は、SSDI\_Request\_Abort() を使用して命令実行を停止してください。

### 【備考】

命令先頭からのサイクル数 ( inst\_cycle ) は、常に 0 を通知します。

## G.8 SSDI\_Execute\_Event ( 実行命令通知 )

---

**【書式】**

```
void SSDI_Execute_Event(EXECUTE_EVENT *info)
```

**【引数】**

```
typedef struct {  
    unsigned long    addr;           // アクセスアドレス  
    unsigned long    total_cycle;    // トータルサイクル数  
} EXECUTE_EVENT;
```

**【復帰値】**

なし

**【説明】**

命令実行により、フェッチアクセスイベントが発生したことを通知します。  
このイベントは、命令実行直前に命令の先頭アドレスのみを通知します。

**【備考】**

## G.9 SSDI\_Interrupt\_Event ( 割込み通知 )

---

**【書式】**

```
void SSDI_Interrupt_Event(INTERRUPT_EVENT *info)
```

**【引数】**

```
typedef struct {  
    int            int_number;    // 割込み番号  
    unsigned long  total_cycle;   // トータルサイクル数  
    unsigned long  int_cycle;    // 割込み処理サイクル数  
    unsigned long  *cycle;       // 遅延するサイクル数  
} INTERRUPT_EVENT;
```

**【復帰値】**

なし

**【説明】**

命令実行により、割り込みイベントが発生したことを通知します。

遅延するサイクル数 ( info.cycle ) は、処理で発生したサイクル数を設定してください。  
ISS では、この値をサイクル数に加えます。

このイベントにより、異常がある場合は、SSDI\_Request\_Abort() を使用して命令実行を  
停止してください。

**【備考】**

割込み処理サイクル数 ( int\_cycle ) は、常に 0 を通知します。

## G.10 SSDI\_Timer\_Event ( タイマ通知 )

---

**【書式】**

```
void SSDI_Timer_Event(TIMER_EVENT *info)
```

**【引数】**

```
typedef struct {  
    unsigned long    total_cycle;    // トータルサイクル数  
    unsigned long    inst_cycle;     // 前回のイベントからの差分サイクル数  
    int              time_id;        // 設定したタイマ ID  
    unsigned long    *cycle;         // 遅延するサイクル数  
} TIMER_EVENT;
```

**【復帰値】**

なし

**【説明】**

ISS ( 命令セットシミュレータ ) に設定しているタイマイイベント発生を通知します。  
遅延するサイクル数 ( info.cycle ) は , 処理で発生したサイクル数を設定してください。  
ISS では , この値をサイクル数に加えます。  
このイベントにより , 異常がある場合は , SSDI\_Request\_Abort() を使用して命令実行を  
停止してください。

**【備考】**

Timer イベントの条件設定は , SSDI\_Set\_Timer() で設定してください。

## G.11 SSDI\_Read\_Memory (メモリリード)

---

### 【書式】

```
int SSDI_Read_Memory(unsigned long addr, int size, int length, void *data)
```

### 【引数】

unsigned long	addr;	// アクセスアドレス
int	size;	// アクセスサイズ ( 1:Byte / 2:Word / 4:LONG )
int	length;	// リードデータ個数
void	*data;	// データ格納領域

### 【復帰値】

読み込みが完了したデータ個数

復帰値とリードデータ個数 ( length ) が異なる場合は、エラーが発生しています。

### 【説明】

メモリの読み込みを行います。

データ格納領域 ( data ) は、アクセスサイズで型が異なります。

Byte ( 1 バイト )                      unsigned char []

Word ( 2 バイト )                      unsigned short []

Long ( 4 バイト )                      unsigned long []

### 【備考】

このアクセスでは、リード通知は行いません。

## G.12 SSDI\_Write\_Memory (メモリライト)

---

### 【書式】

```
int SSDI_Write_Memory(unsigned long addr, int size, int length, void *data)
```

### 【引数】

unsigned long	addr;	// アクセスアドレス
int	size;	// アクセスサイズ ( 1:Byte / 2:Word / 4:LONG )
int	length;	// ライトデータ個数
void	*data;	// データ格納領域

### 【復帰値】

書き込みが完了したデータ個数

復帰値とライトデータ個数 ( length ) が異なる場合は、エラーが発生しています。

### 【説明】

メモリの書き込みを行います。

データ格納領域 ( data ) は、アクセスサイズで型が異なります。

Byte ( 1 バイト )	unsigned char []
----------------	------------------

Word ( 2 バイト )	unsigned short []
----------------	-------------------

Long ( 4 バイト )	unsigned long []
----------------	------------------

### 【備考】

このアクセスでは、ライト通知は行いません。

## G.13 SSDI\_Read\_Register (レジスタリード)

---

### 【書式】

```
int SSDI_Read_Register(int reg_no, unsigned long *data)
```

### 【引数】

int	reg_no;	// レジスタ番号
unsigned long	*data;	// データ格納領域

### 【復帰値】

=0	正常終了
!=0	エラー

### 【説明】

レジスタの読み込みを行います。

### 【備考】

レジスタ番号は、インクルードファイル (SSDI\_REGISTER.H) に定義されています。

SSDI\_REGISTER.H は X:\¥YYY¥LIB¥911¥SSDI に格納されています。

X:\¥YYY SOFTUNE をインストールしたドライブおよびフォルダ



## G.14 SSDI\_Write\_Register (レジスタライト)

---

**【書式】**

```
int SSDI_Write_Register(int reg_no, unsigned long data)
```

**【引数】**

int	reg_no;	// レジスタ番号
unsigned long	data;	// データ格納領域

**【復帰値】**

=0	正常終了
!=0	エラー

**【説明】**

レジスタの書き込みを行います。

**【備考】**

レジスタ番号は、インクルードファイル (SSDI\_REGISTER.H) に定義されています。

SSDI\_REGISTER.H は X:\¥YYY¥LIB¥911¥SSDI に格納されています。

X:\¥YYY SOFTUNE をインストールしたドライブ及びフォルダ

## G.15 SSDI\_Set\_Interrupt ( 割込み要因設定 )

---

### 【書式】

```
int SSDI_Set_Interrupt(int int_no, int sw)
```

### 【引数】

int	int_no;	// 割込み番号
int	sw;	// 割込み状態 ( =0:OFF / =1:ON )

### 【復帰値】

=0	正常終了
!=0	エラー

### 【説明】

割込み状態の設定を行います。

割込みが受理された場合、コールバック ( SSDI\_interrupt\_Event ) がコールされます。

### 【備考】

割込み状態が ON にした場合、常に割り込み要求が発生します。割込み要因フラグをクリアしたタイミングで割込み状態を OFF にしてください。

## G.16 SSDI\_Set\_Timer ( タイマ設定 )

---

**【書式】**

```
int SSDI_Set_Timer(int no, unsigned long cycle, int sw, int id)
```

**【引数】**

int	no;	// タイマ設定番号
unsigned long	cycle;	// サイクル数
int	sw;	// 条件 ( 0: 繰り返し / 1: 1 回のみ )
int	id;	// 0 を指定してください

**【復帰値】**

タイマ設定番号 ( 0 ~ 31 )

=-1                   エラー

**【説明】**

設定した時点から設定したサイクル数を経過した時点でタイマイイベントを発生させます。

タイマ条件としては、繰り返しと 1 回のみの 2 通りあります。

タイマ設定の解除は、サイクル数 ( cycle ) を -1 で設定します。

タイマ設定番号を -1 で設定した場合には、空いている番号に設定されます。

**【備考】**

## G.17 SSDI\_Request\_Abort ( 中断要求 )

---

【書式】

```
void SSDI_Request_Abort(char *message)
```

【引数】

```
char          *message;    // 中断要因メッセージ
```

【復帰値】

なし

【説明】

ISS ( 命令セットシミュレータ ) に対して中断要求を行います。

この要求を行った場合 , ISS は現在実行している命令を終了した時点で中断します。

中断メッセージ ( message ) は , デバッガの中断メッセージとして表示します。

【備考】

## G.18 SSDI\_Set\_Area (領域設定)

### 【書式】

```
int SSDI_Set_Area(int no, unsigned long start, unsigned long size, int attribute, int id)
```

### 【引数】

int	no;	// 領域設定番号
unsigned long	start;	// 領域先頭アドレス
unsigned long	size;	// 領域サイズ
unsigned long	attribute;	// 属性
int	id;	// 0 を指定してください

### 【復帰値】

領域設定番号 (0 ~ 31)  
 =-1 エラー

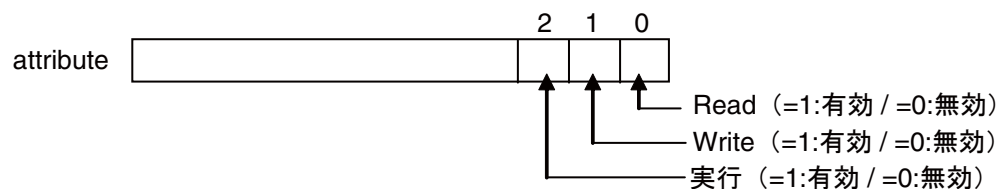
### 【説明】

リードイベント、ライトイベント、実行命令イベントの発生する領域を設定します。

領域設定を行った領域にアクセスがあった場合にのみイベントが発生します。そのため、SSDI\_Init (初期化) で領域設定を行ってください。

領域設定番号を -1 として設定した場合は、領域設定アドレス / 領域サイズが有効となり、その他の場合は、領域設定番号に設定されている内容に対して属性の設定を行います。

属性は、以下のように設定します。



領域設定を削除する場合は、領域設定番号に削除する領域の番号を設定して、属性を 0 で設定してください。

設定できる領域は、32 領域あります。設定数を越えた場合は、エラーとなります。

### 【備考】

## G.19 SSDI\_Clear\_Cycle ( サイクルクリア )

---

【書式】

```
void SSDI_Clear_Cycle(void)
```

【引数】

なし

【復帰値】

なし

【説明】

ISS ( 命令セットシミュレータ ) で管理しているトータルサイクル数の初期化を行います。

【備考】

## G.20 SSDI\_Execute\_MCU (プログラム実行)

---

**【書式】**

```
void SSDI_Execute_MCU(void)
```

**【引数】**

なし

**【復帰値】**

なし

**【説明】**

ISS ( 命令セットシミュレータ ) に対してプログラム開始を行います。  
すでにプログラム実行を開始している場合は、無視されます。

**【備考】**

## G.21 SSDI\_GetCurrentExecuteAddr (実行アドレス取得)

---

【書式】

unsigned long SSDI\_GetCurrentExecuteAddr(void)

【引数】

なし

【復帰値】

現在実行している命令のアドレス

【説明】

現在実行している命令のアドレスを取得します。

【備考】



# 索引

## A

ABORT スイッチ プログラムの中断 (ABORT スイッチ) について .....	407
--	-----

## C

C++ [C++] に属するオプション .....	237
C/C++ コンパイラオプション C/C++ コンパイラオプションの設定方法 .....	228
CPU 情報 CPU 情報の設定の設定方法 .....	225

## I

I/O ポート 出力ポートの設定 .....	319
出力ポートの設定解除 .....	320
入力ポートの設定 .....	317
入力ポートの設定解除 .....	318
inline 展開 指定した関数の inline 展開の設定解除方法 .....	235
指定した関数の inline 展開の設定方法 .....	235

## L

LCD 表示 通常動作時の LCD 表示 .....	412
-------------------------------	-----

## M

MCU 設定 MCU 設定の設定方法 .....	223
MCU リセット MCU リセット .....	280
MCU 変更ダイアログ MCU 変更ダイアログの設定方法 .....	227

## R

ROM/RAM ROM/RAM の領域を指定してそこにセクション を配置する場合 .....	253
ROM/RAM 領域の指定が内蔵 ROM/RAM の 範囲外の場合に警告を出力する方法 .....	255
ROM/RAM 領域の定義方法 .....	255
セクション配置が ROM/RAM 領域の指定範囲外 のときに警告を出力する方法 .....	255
ROM パッチ ROM パッチの設定 .....	284

## S

SOFTUNE Workbench SOFTUNE Workbench の環境設定 .....	311
SOFTUNE Workbench の起動 .....	15
SOFTUNE Workbench の構成 .....	3
SOFTUNE Workbench の終了 .....	15
SOFTUNE Workbench の動作条件を設定する .....	4

## U

USB ドライバ USB ドライバのインストール .....	406
-----------------------------------	-----

## あ

アイコン	
アイコンの整列	357
アウトプットウィンドウ	
アウトプットウィンドウ	60, 166
ショートカットメニュー	60
操作	61
アセンブラオプション	
アセンブラオプションの設定方法	238
アドレス	
アドレスで指定の場合の注意	254
アドレス式	
アドレス式	6
アドレスマスク	
アドレスマスクについて (例)	191
アドレス範囲指定	
アドレス範囲指定	13

## い

依存関係	
依存関係の更新	275
一般	
[一般]に属するオプションの設定方法	229, 239, 246
イベント	
イベント一覧	297
イベントの設定	294
インクルードパス	
インクルードパスの検索順位の変更方法	232, 242
インクルードパスの設定解除方法	232, 242
インクルードパスの設定方法	231, 241
印刷	
印刷対象	134

## う

ウィンドウ	
ウィンドウの更新	358
ウィンドウの表示方法に関する制御	354
すべてのウィンドウの更新	359
開いているウィンドウのリスト	354
ウィンドウ表示	
ウィンドウ表示 / 非表示関係	165
ウィンドウ表示関係 (デバッグセッション)	165
ウォッチウィンドウ	
ウォッチウィンドウ	90, 180
プロパティ: 一般	164
ショートカットメニュー	92
上書き保存	
上書き保存ダイアログ	131

## え

エイリアス	
エイリアス	98
エディタ	
エディタの登録	348

削除手順	349
使用するエディタの設定方法	350
設定例	350
登録手順	349
変更手順	349
エディット	
エディット中の場合	136
エディットウィンドウ	
エディットウィンドウ	63, 144
ショートカットメニュー	64
プロパティ: 一般	160
エミュレータ	
エミュレータ側の設定	404
エミュレータデバッグ	
エミュレータデバッグを選択した場合の手順	340
エミュレータ本体	
エミュレータ本体が MB2197 の場合	401
エミュレータ本体が MB2198 の場合	402
エミュレータ本体へモニタプログラムを	
ロードした場合のエラー表示	413
エラー	
最後のエラー	153
先頭のエラー	153
次のエラー	153
前のエラー	153
エラージャンプ	
エラージャンプ	137
解析順序の変更, 適用の ON/OFF	352
削除手順	351
設定手順	351
設定例	352
変更手順	352
エラー表示	
エミュレータデバッグ実行中のエラー表示	412
エミュレータ本体へモニタプログラムを	
ロードした場合のエラー表示	413
演算子	
演算子	12

## か

重ねて表示	
重ねて表示	355
カスタマイズ	
カスタマイズ手順 (表示色選択)	68
カスタマイズ手順 (キーワードの削除)	69
カスタマイズ手順 (キーワードの初期化)	69
カスタマイズ手順 (キーワードの追加)	68
カスタマイズ手順 (タブ数)	68
カスタマイズ手順 (表示機能選択)	67
カバレッジ	
カバレッジウィンドウ	198
カバレッジ設定	198
環境	
SOFTUNE Workbench の環境設定	311
デバッグの環境設定	311
環境変数	
環境変数	312
関数コール	
関数コール	303

## き

キーボード	
既に割り当てられているキーの解除手順	346
既に割り当てられているキーの変更手順	346
すべてのキー設定を初期状態に戻したい場合	346
設定手順	346
キーワードの削除	
カスタマイズ手順 (キーワードの削除)	69
キーワードの初期化	
カスタマイズ手順 (キーワードの初期化)	69
キーワードの追加	
カスタマイズ手順 (キーワードの追加)	68
基数	
基数設定	179, 180
起動	
SOFTUNE Workbench の起動	15
機能選択	
カスタマイズ手順 (表示機能選択)	67
逆アセンブル	
逆アセンブル	168
逆アセンブルウィンドウ	
逆アセンブルウィンドウ	79
ショートカットメニュー	80
行番号	
行番号	10
切り取り	
切り取り	139

## け

言語系ツール	
言語系ツール	2
言語仕様	
[言語仕様] に属するオプションの設定方法	236
検索	
キーワード	362
検索	141
トピックの検索	362
目次	362
検索機能	
文字列の検索機能	137

## こ

構成	
SOFTUNE Workbench の構成	3
コードブレーク	
コードブレーク詳細: デバッグ種別	
(ハードウェア / データ監視)	292
コードブレーク詳細 (MB2198)	289
コードブレークポイント	
コードブレークポイントの設定	281
コールクリア	
コールクリア	305
コールスタック	
コールスタック	301
コピー	
コピー	139

コマンド	
コマンド	194
コマンドウィンドウ	
ショートカットメニュー (コマンド入力	
フィールド)	96
ショートカットメニュー (コマンド入力	
フィールド以外)	97
コマンドウィンドウ	95
コンバータオプション	
コンバータオプションの設定方法	259
コンパイル	
コンパイル	276

## さ

最適化	
[最適化] に属するオプションの設定方法	233
[最適化] の「詳細設定」に属するオプションの	
設定方法	234
指定した関数の inline 展開の設定解除方法	235
指定した関数の inline 展開の設定方法	235
削除	
削除	139
サポート情報	
サポート情報	363
サンプルプログラム	
サンプルプログラムの説明	409

## し

シーケンス	
シーケンス	298
時間測定	
時間測定	302
識別子	
識別子	6
実行	
実行	278
シミュレータデバッグ	
シミュレータデバッグを選択した場合の手順	341
ジャンプ機能	
ジャンプ機能	137
終了	
SOFTUNE Workbench の終了	15
詳細設定	
[最適化] の「詳細設定」に属するオプションの	
設定方法	234
シンボル	
シンボル	9
シンボル一覧	167
シンボルウィンドウ	
ショートカットメニュー	78
シンボルウィンドウ	77
プロパティ: 一般	162
シンボル定義	
シンボル定義の設定解除方法	247
シンボル定義の設定方法	247

## す

ステータスバー	
ステータスバー .....	41, 195
ツールバー / ステータスバー関係 .....	165

## せ

整数	
整数 .....	7
セクション	
ROM/RAM の領域を指定してそこにセクション	
を配置する場合 .....	253
セクションの設定 .....	26
セクションの配置 / 結合指定方法 .....	252
セクション配置が ROM/RAM 領域の指定範囲外	
のときに警告を出力する方法 .....	255
セクション配置順の変更 .....	254
先頭アドレス指定してそこにセクションを	
配置する場合 .....	254
絶対形式アセンブリリスト	
絶対形式アセンブリリストに関するオプションの	
設定方法 .....	249
設定完了	
OK ボタンをクリックすると設定完了です .....	25
セットアップウィザード	
セットアップウィザード .....	31
セットアップウィザード操作手順 .....	340
先頭アドレス指定	
先頭アドレス指定してそこにセクションを	
配置する場合 .....	254

## そ

ソースウィンドウ	
ソースウィンドウ .....	72
プロパティ：一般 .....	161
ソースファイル	
新規ソースファイルの作成方法 .....	21
ソースファイルの保存の確認 .....	276
ソースファイルやドキュメントファイルなどを	
作成する場合 .....	125
その他	
その他 .....	2
その他の設定 .....	311

## た

ターゲットシステム	
ターゲットシステムの構成 .....	407
ターゲットシステムの作成 .....	407
ターゲットシステムの作成手順 .....	408
ターゲットファイル	
ターゲットファイル .....	309
ターゲットファイルのロード .....	309
ターゲットプログラム	
ターゲットプログラムのロード .....	30
対象	
対象 .....	138, 140, 141
タブ数	
カスタマイズ手順 (タブ数) .....	68

## ち

置換	
置換 .....	142
注意	
アドレスで指定の場合の注意 .....	254
中止	
中止 .....	28, 276

## つ

ツール	
削除手順 .....	344
設定手順 .....	344
ツールについて .....	343
ツールの起動 .....	353
ツールの設定例 .....	345
変更手順 .....	345
ツールバー	
共通 .....	39
ツールバー .....	195
ツールバー / ステータスバー関係 .....	165
デバッグ .....	40
ビルド .....	39
フラグ .....	40
プロジェクト .....	39

## て

提供サンプル	
提供サンプルファイル .....	409
停止	
停止 .....	279
データ式	
データ式 .....	6
データブレイク	
データブレイク詳細 (MB2198) .....	293
データブレイクポイント	
データブレイクポイントの設定 .....	285
テキスト	
新規に作成したテキストの処理 .....	133
テキストファイル	
最新のテキストファイル .....	135
デバッグ	
デバッグ .....	2
デバッグに関する設定 .....	31
デバッグの環境設定 .....	311
デバッグの操作 .....	30
デバッグ	
デバッグ .....	40
デバッグ開始 .....	32
デバッグセッション時のデバッグの制御関係	
.....	277
デバッグ中の場合 .....	136
デバッグの開始 .....	310
デバッグの開始と終了関係 .....	277
デバッグの終了 .....	310
デバッグを行う .....	4
デバッグオプション	
[一般] に属するオプションの設定方法 .....	262
セットアップの設定 .....	263
セットアップの設定削除 .....	264

セットアップの設定変更 .....	264
セットアップ名の変更 .....	264
デバッグオプションの設定方法 .....	262
デバッグ環境	
設定する項目 .....	316
デバッグセッション	
デバッグセッション .....	144
デバッグセッションへの移行 .....	30
デバッグ環境	
デバッグ環境設定手順 .....	322
デフォルトライブラリ	
デフォルトライブラリを使用しない場合の 設定方法 .....	250

## と

ドキュメントファイル	
ソースファイルやドキュメントファイルなどを 作成する場合 .....	125
特殊	
特殊操作 .....	175
閉じる	
すべて閉じる .....	360
閉じる .....	128
取り消し	
不要になった変数表示の取り消し .....	181
トレース	
更新 .....	183
トレース .....	183
トレース検索 .....	190
トレース詳細【DSU3】 .....	192
トレース設定 .....	184
トレースウィンドウ	
ショートカットメニュー .....	94
トレースウィンドウ .....	93

## な

名前を付けて保存	
名前を付けて保存 .....	132
並べて表示	
左右に並べて表示 .....	355
上下に並べて表示 .....	355

## は

バージョン	
デバッグセッション時のバージョン表示 .....	364
非デバッグセッション時のバージョン表示 .....	364
バージョン情報	
バージョン情報 .....	361
配置 / 結合の設定について	
配置 / 結合の設定について .....	246
バイナリファイル	
バイナリファイルを保存する場合 .....	132
バイナリファイルを開く場合 .....	127
パソコン	
パソコン側の設定 .....	403
バックアップファイル	
バックアップファイル .....	34

バクトレース	
バクトレース .....	191
バッチファイル	
バッチファイルを開く場合 .....	127
バッファサイズ	
バッファサイズ .....	183
貼り付け	
貼り付け .....	139

## ひ

ビットアドレス指定	
ビットアドレス指定 .....	13
表示色選択	
カスタマイズ手順 ( 表示色選択 ) .....	68
標準エディタ	
標準エディタ設定 .....	67
開く	
開くダイアログ .....	126
ビルド	
アウトプットウィンドウを使用 .....	267
エクスポート .....	266
オプション .....	267
起動順序の変更手順 .....	269
削除手順 .....	269
実行時に追加オプションを指定 .....	267
実行時のディレクトリ .....	267
実行ファイル名 .....	267
指定プロジェクトをメイク / ビルドするには... .....	29
設定手順 .....	268
設定の対象 .....	266
タイトル .....	267
テンポラリファイルの内容 .....	268
ビルド .....	28, 39, 276
ビルドのカスタマイズ機能について .....	265
プロジェクトのメイク / ビルド .....	29
ボタン一覧 .....	265
メイク / ビルド関連 .....	208
メイク / ビルド時のサブプロジェクトの構成を 変更するには... .....	29
有効 .....	267
リセット .....	267

## ふ

ファイル	
ファイルの編集機能 .....	137
編集集中のファイルがある場合 .....	130
編集集中のファイルがない場合 .....	130
ファイル名	
ファイル名の指定 .....	6, 14
ファイルメニュー	
その他の機能 .....	123
プロジェクトおよびファイルのアクセスに関する 機能 .....	123
フォント	
フォント関係 .....	165
フォントの設定 .....	196
フォントの変更 .....	196
フォントのリセット .....	196

ブックマーク機能	
ブックマーク機能	137
浮動小数点数	
浮動小数点数	8
フラグ	
フラグ	40
フラグ名	
レジスタ、フラグ名	11
フルアドレス指定	
フルアドレス指定	13
ブレイク	
ブレイク設定・解除	287
ブレイクポイント	
ROM パッチ	284
コードブレイク詳細 (MB2198)	289
コードブレイクポイント	281
データブレイクポイント	284
ブレイクポイント	281
プログラムソース	
プログラムソースを作成し、メイク/ビルドを 実行する	4
プロジェクト	
[プロジェクト] - [プロジェクトの設定] を 選択してください	25
CPU 情報の設定の設定方法	225
MCU 設定の設定方法	223
アクティブなプロジェクト構成を設定するには...	24
アクティブプロジェクトの設定	209
アクティブプロジェクトを設定するには...	20
一般の設定方法	221
既存のプロジェクト間でサブプロジェクトを 定義するには...	22
既存のプロジェクト構成の設定に基づき 作成するには...	23
既存のプロジェクトをワークスペースに 登録するには...	19
現在開いているワークスペースに 新規プロジェクトを登録するには...	18
作成したファイルをディレクトリごと プロジェクトに登録するには...	21
作成したファイルをプロジェクトに 登録するには...	21
サブプロジェクトの定義	22
設定する項目	220
設定手順	273, 274
設定を行う対象	219
追加手順	212, 213
プロジェクト	39
プロジェクトおよびファイルのアクセスに 関する機能	123
プロジェクトからサブプロジェクトを 削除するには...	22
プロジェクトから特定の構成を削除するには...	24
プロジェクト構成	271
プロジェクト構成 - 追加と削除	272
プロジェクト構成の作成	23
プロジェクト構成 - ビルド時の構成	274
プロジェクトに登録したファイルを 削除するには...	21
プロジェクトの依存関係	270
プロジェクトの依存関係の定義手順	270

プロジェクトの作成	31
プロジェクトの設定	218
プロジェクトの設定関連	208
プロジェクトの追加	210
プロジェクトの追加 - 既存のプロジェクト 削除するには...	213
プロジェクトの追加 - 新規作成	211
プロジェクトの登録	18
プロジェクトファイルの移行方法	35
プロジェクトファイルを保存する場合	132
プロジェクトを設計する	4
メイク/ビルド関連	208
ワークスペースに登録したプロジェクトを 削除するには...	20
プロジェクトウィンドウ	
プロジェクトウィンドウ	166
プロパティ: 依存関係	157
プロパティ: 一般	155
プロパティ: 出力	159
プロパティ: 入力	158
プロジェクトファイル 手順	33
プロパティ	
プロパティ	154
プロパティ: 依存関係	157
プロパティ: 一般	155, 160, 161, 162, 163, 164
プロパティ機能	
プロパティ機能	137
分割	
上下に分割	356

## へ

ベクタ	
コピー	308
ジャンプ	308
ベクタ	306
ベクタの表示と設定	306

## ヘルプ

オンラインヘルプ	361
バージョン情報	361

## 編集機能

ファイルの編集機能	137
-----------	-----

## 変数

不要になった変数表示の取り消し	181
変数	99

## 変数値

変数値の変更	179, 181
--------	----------

## ほ

## ホストシステム

ホストシステムとの通信について	407
-----------------	-----

## 保存

名前を付けて保存	132
バイナリファイルを保存する場合	132
プロジェクトファイルを保存する場合	132
保存対象	133

## ま

マクロ	
マクロ .....	100
マクロ名	
マクロ名の設定解除方法 .....	231, 241
マクロ名の設定方法 .....	230, 240
マングル名	
マングル名 .....	10

## め

メイク	
指定プロジェクトをメイク/ビルドするには...	29
プロジェクトのメイク/ビルド .....	29
メイク .....	28, 276
メイク/ビルド関連 .....	208
メイク/ビルド時のサブプロジェクトの構成を 変更するには...	29
メインウィンドウ	
メインウィンドウ .....	38
メニュー	
メニュー構成 .....	120
メモリ	
編集 .....	178
メモリ内容の検索 .....	174
メモリ内容の表示 .....	172
メモリ内容の変更 .....	172
メモリウィンドウ	
ショートカットメニュー .....	87
表示設定 .....	178
メモリウィンドウ .....	86
メモリブロック	
メモリブロックの比較 .....	177
メモリマップ	
メモリマップの設定 ( シミュレーションデバッグ 時のみ表示 ) .....	342
メンバ	
追加手順 .....	215, 216
メンバの追加 .....	214
メンバの追加 - ディレクトリ .....	216
メンバの追加 - ファイル .....	215

## も

文字定数	
文字定数 .....	10
文字列置換	
文字列置換設定 .....	98
モニタデバッグ	
モニタデバッグが制御できない場合の処置 .....	411
モニタデバッグを選択した場合の手順 .....	341
モニタプログラム	
エミュレータ本体が MB2197 の場合 .....	401
エミュレータ本体が MB2198 の場合 .....	402
モニタプログラム自動ダウンロード	
モニタプログラムの自動ダウンロード .....	30

## ら

ライブラリアン	
[一般]に属するオプションの設定方法 .....	257
ライブラリアンオプションの設定方法 .....	257
ライブラリアンの起動 .....	258
リスト出力の設定方法 .....	258
ライブラリ制御	
ライブラリ制御に関するオプションの設定方法 .....	250
ライブラリファイル	
ライブラリファイルのモジュールに対する デバッグ情報存在チェックを抑止する場合 の設定方法 .....	250
ライブラリ検索パス	
ライブラリ検索パスの検索順位の変更方法 .....	251
ライブラリ検索パスの設定解除方法 .....	251
ライブラリ検索パスの設定方法 .....	251
ラインアセンブル	
ラインアセンブル .....	169

## り

リアルタイムメモリ	
表示設定 .....	201
リアルタイムメモリウィンドウ .....	200
領域 .....	200
リアルタイムメモリウィンドウ	
ショートカットメニュー .....	104, 108
リアルタイムメモリウィンドウ .....	103
リスト出力	
リスト出力の設定方法 .....	243, 248
リンカオプション	
セクションの設定 .....	26
メモリ領域の設定 .....	26
リンカオプションの自動設定 .....	26
リンカオプションの設定 .....	26
リンカオプションの設定方法 .....	245

## れ

レジスタ	
表示設定 .....	84
レジスタ .....	170, 400
レジスタ値の変更 .....	170
レジスタ, フラグ名 .....	11
レジスタウィンドウ	
ショートカットメニュー .....	83
レジスタウィンドウ .....	82

## ろ

ローカルウィンドウ	
プロパティ: 一般 .....	163
ローカル変数	
ローカル変数表示 .....	179
ローカル変数ウィンドウ	
ショートカットメニュー .....	88
ローカル変数ウィンドウ .....	88

## 索引

ロードモジュール	
ロードモジュールファイルを開く場合 .....	127
ロギング	
ロギング .....	102

## わ

ワークスペース	
空のワークスペースを作成するには... ..	17
既存のワークスペースを開く .....	129
最新のワークスペース .....	135
プロジェクトオープン時にワークスペースを 作成するには... ..	17

プロジェクトの新規作成時にワークスペースを 作成するには... ..	16
ワークスペース .....	314
ワークスペースが開かれている場合 .....	136
ワークスペースの作成 .....	16
ワークスペースの設定 .....	31, 217
ワークスペース / プロジェクト	
新規プロジェクトを作成する場合 .....	125
新規ワークスペースを作成する場合 .....	124
割込み	
割込み設定解除 .....	321
割込み設定手順 .....	321



CM71-00328-4

---

**富士通マイクロエレクトロニクス・CONTROLLER MANUAL**

FR ファミリ

SOFTUNE™ Workbench

操作マニュアル

V6 対応

---

2008 年 6 月 第 4 版発行

---

発行	<b>富士通マイクロエレクトロニクス</b>
編集	マーケティング統括部 ビジネス推進部

---

