

**FR ファミリ**  
**SOFTUNE™ Workbench**  
**コマンドリファレンス**  
**マニュアル**  
V6 対応



**FR ファミリ**  
**SOFTUNE™ Workbench**  
**コマンドリファレンス**  
**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、富士通マイクロプロセッサ / マイクロコントローラ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 のコマンドおよび組込み変数 / 関数についてリファレンス形式で解説したものです。

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。

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。

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。

## ■ 本書の全体構成

本書は、以下に示す 13 の章および付録で構成されています。

### 第 1 章 環境設定コマンド

この章では、環境設定コマンドについて説明します。

### 第 2 章 プログラム実行コマンド

この章では、プログラム実行コマンドについて説明します。

### 第 3 章 ブレーク/イベント制御コマンド

この章では、ブレーク/イベント制御コマンドについて説明します。

### 第 4 章 プログラム実行の解析コマンド

この章では、プログラム実行の解析コマンドについて説明します。

### 第 5 章 メモリ/レジスタ操作コマンド

この章では、メモリ/レジスタ操作コマンドについて説明します。

### 第 6 章 アセンブル/逆アセンブルコマンド

この章では、アセンブル/逆アセンブルコマンドについて説明します。

### 第 7 章 ロード/セーブコマンド

この章では、ロード/セーブコマンドについて説明します。

### 第 8 章 ソースファイル/シンボルコマンド

この章では、ソースファイル/シンボルコマンドについて説明します。

### 第 9 章 コマンドプロシージャコマンド

この章では、コマンドプロシージャコマンドについて説明します。

### 第 10 章 置き換えコマンド

この章では、置き換えコマンドについて説明します。

### 第 11 章 ユーティリティコマンド

この章では、ユーティリティコマンドについて説明します。

### 第 12 章 制御コマンド

この章では、制御コマンドについて説明します。

### 第 13 章 組込み変数/関数

この章では、組込み変数/関数について説明します。

### 付録

付録では、マネージャ関係およびデバッガ関係の、メッセージを掲載しています。

- 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に、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。

# コマンドリファレンス記述形式

コマンドリファレンスの記述形式は、以下のようになっています。

コマンド名
[ 使用可能デバッガ種別 ]
形式
機能
例

コマンド名：

説明が記述されているコマンド名です。

使用可能デバッガ種別：

デバッガの種別により使用可能なコマンドが異なるため、使用可能かどうかを記述しています。

：使用できます。

：使用できます（命令実行中には使用できません）。

×：使用できません。

-：デバッガが存在しません。

形式：

コマンドの書式、パラメータ、コマンド修飾子について説明しています。コマンド入力、この形式に従って入力します。

機能：

コマンドの機能について説明しています。

例：

コマンドの記述例です。実際とは、多少異なる場合があります。



# 目次

<b>第 1 章</b>	<b>環境設定コマンド</b>	<b>1</b>
1.1	INITIALIZE	3
1.2	EXIT	4
1.3	RESET	5
1.4	SET RUNMODE	6
1.5	SHOW RUNMODE	8
1.6	SET MODE	9
1.7	SHOW MODE	11
1.8	SET RADIX	12
1.9	SHOW RADIX	13
1.10	SET SOURCE	14
1.11	SHOW SOURCE	15
1.12	SHOW SYSTEM	16
1.13	SET MAP (type 1)	17
1.14	SET MAP (type 2)	18
1.15	SHOW MAP (type 1)	19
1.16	SHOW MAP (type 2)	20
1.17	ENABLE MAP	21
1.18	DISABLE MAP	22
1.19	CANCEL MAP (type 1)	23
1.20	CANCEL MAP (type 2)	24
1.21	ENABLE VERIFYMODE	25
1.22	DISABLE VERIFYMODE	26
1.23	SHOW VERIFYMODE	27
1.24	SET INPORT	28
1.25	SHOW INPORT	30
1.26	CANCEL INPORT	31
1.27	SET OUTPORT	32
1.28	SHOW OUTPORT	34
1.29	CANCEL OUTPORT	35
1.30	SET INTERRUPT	36
1.31	SHOW INTERRUPT	37
1.32	CANCEL INTERRUPT	38
1.33	COPY VECTOR	39
1.34	SET VECTOR	40
1.35	SHOW VECTOR	41
1.36	ENABLE WATCHDOG	42
1.37	DISABLE WATCHDOG	43
1.38	SHOW WATCHDOG	44
1.39	SET EXTERNALMEMORY	45
1.40	ENABLE EXTERNALMEMORY	46
1.41	DISABLE EXTERNALMEMORY	47
1.42	SHOW EXTERNALMEMORY	48
1.43	SET WATCH	49
1.44	CANCEL WATCH	51

1.45	SET FREQUENCY .....	52
1.46	SHOW FREQUENCY .....	53
1.47	SET REALTIMEMEMORYAREA.....	54
1.48	SHOW REALTIMEMEMORYAREA.....	55
<b>第 2 章</b>	<b>プログラム実行コマンド.....</b>	<b>57</b>
2.1	GO .....	58
2.2	STEP .....	59
2.3	SET STEP .....	61
2.4	SHOW STEP .....	62
2.5	CALL.....	63
2.6	CLEAR CALL.....	65
2.7	SHOW STATUS .....	66
<b>第 3 章</b>	<b>ブレーク/イベント制御コマンド .....</b>	<b>67</b>
3.1	SET BREAK (type 1) .....	69
3.2	SET BREAK (type 2) .....	72
3.3	SET BREAK (type 3) .....	73
3.4	SHOW BREAK .....	75
3.5	CANCEL BREAK.....	77
3.6	ENABLE BREAK .....	78
3.7	DISABLE BREAK .....	79
3.8	SET DATABREAK (type 1) .....	80
3.9	SET DATABREAK (type 2) .....	82
3.10	SHOW DATABREAK.....	84
3.11	CANCEL DATABREAK .....	85
3.12	ENABLE DATABREAK.....	86
3.13	DISABLE DATABREAK.....	87
3.14	SET EVENT.....	88
3.15	SHOW EVENT.....	91
3.16	CANCEL EVENT .....	92
3.17	ENABLE EVENT.....	93
3.18	DISABLE EVENT.....	94
3.19	SET CODEEVENT .....	95
3.20	SHOW CODEEVENT .....	97
3.21	CANCEL CODEEVENT .....	99
3.22	ENABLE CODEEVENT .....	100
3.23	DISABLE CODEEVENT .....	101
3.24	SET DATAEVENT .....	102
3.25	SHOW DATAEVENT .....	104
3.26	CANCEL DATAEVENT.....	105
3.27	ENABLE DATAEVENT .....	106
3.28	DISABLE DATAEVENT .....	107
3.29	SET SEQUENCE (type 1) .....	108
3.30	SET SEQUENCE (type 2) .....	110
3.31	SHOW SEQUENCE .....	111
3.32	CANCEL SEQUENCE .....	112
3.33	ENABLE SEQUENCE .....	113
3.34	DISABLE SEQUENCE .....	114

3.35	SET TRIGGER .....	115
3.36	SHOW TRIGGER .....	116
3.37	ENABLE ALIGNMENTBREAK .....	117
3.38	DISABLE ALIGNMENTBREAK .....	118
3.39	SHOW ALIGNMENTBREAK .....	119
3.40	SET BREAKCONDITION (type 1) .....	120
3.41	SET BREAKCONDITION (type 2) .....	122
3.42	SHOW BREAKCONDITION .....	124
<b>第 4 章</b>	<b>プログラム実行の解析コマンド .....</b>	<b>125</b>
4.1	SET PERFORMANCE .....	127
4.2	SHOW PERFORMANCE .....	128
4.3	CLEAR PERFORMANCE .....	130
4.4	SET COVERAGE .....	131
4.5	SHOW COVERAGE .....	132
4.6	CANCEL COVERAGE .....	135
4.7	CLEAR COVERAGE .....	136
4.8	SHOW CALLS .....	137
4.9	SHOW TIMER .....	138
4.10	CLEAR TIMER .....	140
4.11	SET TRACE (type 1) .....	141
4.12	SET TRACE (type 2) .....	142
4.13	SHOW TRACE (type 1) .....	144
4.14	SHOW TRACE (type 2) .....	146
4.15	CLEAR TRACE .....	147
4.16	ENABLE TRACE .....	148
4.17	DISABLE TRACE .....	149
4.18	SEARCH TRACE .....	150
4.19	SET DATATRACEAREA .....	152
4.20	SHOW DATATRACEAREA .....	154
4.21	CANCEL DATATRACEAREA .....	155
4.22	SHOW DETAILTRACE .....	156
4.23	SET TRACETRIGGER .....	158
4.24	SHOW TRACETRIGGER .....	160
4.25	CANCEL TRACETRIGGER .....	161
4.26	ENABLE TRACETRIGGER .....	162
4.27	DISABLE TRACETRIGGER .....	163
4.28	SET RAMCHECK .....	164
4.29	SHOW RAMCHECK .....	167
4.30	CANCEL RAMCHECK .....	168
4.31	ENABLE RAMCHECK .....	170
4.32	DISABLE RAMCHECK .....	172
<b>第 5 章</b>	<b>メモリ / レジスタ操作コマンド .....</b>	<b>175</b>
5.1	EXAMINE .....	176
5.2	ENTER .....	178
5.3	SET MEMORY .....	180
5.4	SHOW MEMORY .....	182
5.5	SEARCH MEMORY .....	184

5.6	SET REGISTER .....	186
5.7	SHOW REGISTER .....	187
5.8	COMPARE.....	188
5.9	FILL .....	189
5.10	MOVE .....	190
5.11	DUMP .....	191
<b>第 6 章</b>	<b>アセンブル/逆アセンブルコマンド .....</b>	<b>193</b>
6.1	ASSEMBLE .....	194
6.2	DISASSEMBLE .....	195
<b>第 7 章</b>	<b>ロード/セーブコマンド .....</b>	<b>197</b>
7.1	LOAD .....	198
7.2	SAVE .....	201
<b>第 8 章</b>	<b>ソースファイル/シンボルコマンド .....</b>	<b>203</b>
8.1	LIST .....	204
8.2	SET PATH .....	206
8.3	SHOW PATH .....	207
8.4	SHOW SCOPE .....	208
8.5	UP .....	209
8.6	DOWN .....	210
<b>第 9 章</b>	<b>コマンドプロシージャコマンド .....</b>	<b>211</b>
9.1	BATCH.....	212
9.2	QUIT .....	213
<b>第 10 章</b>	<b>置き換えコマンド .....</b>	<b>215</b>
10.1	SET ALIAS .....	216
10.2	SHOW ALIAS .....	217
10.3	CANCEL ALIAS .....	218
10.4	SET VARIABLE .....	219
10.5	SHOW VARIABLE .....	220
10.6	CANCEL VARIABLE.....	221
<b>第 11 章</b>	<b>ユーティリティコマンド.....</b>	<b>223</b>
11.1	SET LOGGING .....	224
11.2	SHOW LOGGING.....	226
11.3	CANCEL LOGGING .....	227
11.4	ENABLE LOGGING.....	228
11.5	DISABLE LOGGING.....	229
11.6	PRINTF.....	230
11.7	SET OUTPUT .....	232
11.8	SHOW OUTPUT .....	233
<b>第 12 章</b>	<b>制御コマンド.....</b>	<b>235</b>
12.1	IF .....	236
12.2	REPEAT .....	237

12.3	WHILE .....	238
12.4	BREAK .....	239
<b>第 13 章</b>	<b>組込み変数 / 関数 .....</b>	<b>241</b>
13.1	%CALL .....	242
13.2	%ERRNUM.....	243
13.3	%ENTRY .....	244
13.4	%STKTOP .....	245
13.5	%RADIX.....	246
13.6	%SCPADR.....	247
13.7	%LOADNUM.....	248
13.8	%BIT, %B, %H, %W, %S, %D.....	249
13.9	%STRGET .....	250
13.10	%STRSTR .....	251
13.11	%STRCMP .....	252
13.12	%STRLEN .....	253
13.13	%STRCAT .....	254
13.14	%SYMLen.....	255
13.15	%TOVAL.....	256
13.16	%TOSTR .....	257
13.17	%EVAL .....	258
13.18	%EVENTNUM .....	259
13.19	%TRIGGERNUM .....	260
13.20	%DEBUGFUNCTION .....	261
<b>付録</b>	<b>.....</b>	<b>263</b>
付録 A	マネージャ関係のエラーメッセージ .....	264
付録 B	デバッガ関係のエラーメッセージ.....	287
付録 C	デバッガ中断メッセージ .....	316
<b>索引</b>	<b>.....</b>	<b>321</b>



# 第1章

---

## 環境設定コマンド

この章では、環境設定コマンドについて説明します。

- 1.1 INITIALIZE
- 1.2 EXIT
- 1.3 RESET
- 1.4 SET RUNMODE
- 1.5 SHOW RUNMODE
- 1.6 SET MODE
- 1.7 SHOW MODE
- 1.8 SET RADIX
- 1.9 SHOW RADIX
- 1.10 SET SOURCE
- 1.11 SHOW SOURCE
- 1.12 SHOW SYSTEM
- 1.13 SET MAP (type 1)
- 1.14 SET MAP (type 2)
- 1.15 SHOW MAP (type 1)
- 1.16 SHOW MAP (type 2)
- 1.17 ENABLE MAP
- 1.18 DISABLE MAP
- 1.19 CANCEL MAP (type 1)
- 1.20 CANCEL MAP (type 2)
- 1.21 ENABLE VERIFYMODE
- 1.22 DISABLE VERIFYMODE
- 1.23 SHOW VERIFYMODE
- 1.24 SET INPORT
- 1.25 SHOW INPORT

## 第 1 章 環境設定コマンド

- 1.26 CANCEL INPORT
- 1.27 SET OUTPORT
- 1.28 SHOW OUTPORT
- 1.29 CANCEL OUTPORT
- 1.30 SET INTERRUPT
- 1.31 SHOW INTERRUPT
- 1.32 CANCEL INTERRUPT
- 1.33 COPY VECTOR
- 1.34 SET VECTOR
- 1.35 SHOW VECTOR
- 1.36 ENABLE WATCHDOG
- 1.37 DISABLE WATCHDOG
- 1.38 SHOW WATCHDOG
- 1.39 SET EXTERNALMEMORY
- 1.40 ENABLE EXTERNALMEMORY
- 1.41 DISABLE EXTERNALMEMORY
- 1.42 SHOW EXTERNALMEMORY
- 1.43 SET WATCH
- 1.44 CANCEL WATCH
- 1.45 SET FREQUENCY
- 1.46 SHOW FREQUENCY
- 1.47 SET REALTIMEMEMORYAREA
- 1.48 SHOW REALTIMEMEMORYAREA



## 1.1 INITIALIZE

---

デバッガを初期化します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

INITIALIZE

### ■ 機能

デバッガを初期化します。

マクロ , エイリアス , デバッグ変数以外のすべての設定が無効となります。

### ■ 例

```
>INITIALIZE
```

## 1.2 EXIT

---

デバッガを終了します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

EXIT

### ■ 機能

デバッガを終了します。

### ■ 例

```
> EXIT
```

## 1.3 RESET

---

MCU にリセット信号を入力します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

RESET

### ■ 機能

MCU にリセット信号を入力します。

### ■ 例

```
> RESET
```

## 1.4 SET RUNMODE

---

MCU の動作モードを設定します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET RUNMODE

#### ● コマンド修飾子

- 動作モードの設定

/TRACE ( 起動時のデフォルト )

動作モードをフルトレースモードにします。

/REALTIME

動作モードをリアルタイムモードにします。

/INTERNALTRACE

動作モードを内部トレースモードにします。

/EXTERNALTRACE

動作モードを外部トレースモードにします。

### ■ 機能

MCU の動作モードを設定します。

動作モードにはフルトレースモード、リアルタイムモード、内部トレースモードおよび外部トレースモードがあり、次のような違いがあります。

#### ● フルトレースモード

トレース機能について制限なく使用できますが、実時間で動作できないという制限があります。

#### ● リアルタイムモード

実時間でのプログラム実行が可能となります。

ただし、次の条件ではトレース情報の出力を充分に出力できずに、トレースデータが欠落することがあります。

- 11 サイクル以内に 3 回以上分岐が行われたとき

- 内部トレースモード

チップに内蔵されたトレース専用メモリに、トレース情報を格納します。実時間でプログラムを実行しますが、その機能が搭載された評価用チップでのみ指定が可能です。

- 外部トレースモード

アダプタボード上に装着されたトレース専用メモリに、トレース情報を格納します。実時間でプログラムを実行しますが、その機能が搭載された評価用チップでのみ指定が可能です。

## ■ 例

```
>SET RUNMODE/TRACE
```

---

### < 注意事項 >

ご使用のエミュレータまたはその接続状態、DSU により指定できる修飾子が異なります。詳しくは SOFTUNE Workbench 機能説明書「MCU 動作モード」を確認してください。

---

## 1.5 SHOW RUNMODE

---

SET RUNMODE コマンドで設定した MCU の動作モードを表示します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW RUNMODE

### ■ 機能

SET RUNMODE コマンドで設定した MCU の動作モードを表示します。

### ■ 例

```
>SHOW RUNMODE  
run mode : TRACE
```

## 1.6 SET MODE

イベントモード, デバッグ機能の設定を行います。

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	
Monitor		×

### ■ 形式

[ 形式 1] SET MODE

[ 形式 2] SET MODE デバッグ機能番号

#### ● パラメータ

- 形式 2

デバッグ機能番号

有効にするデバッグ機能を指定します。選択可能なデバッグ機能とその機能番号は, SHOW MODE コマンドで確認できます。

#### ● コマンド修飾子

- 形式 1

- イベントモードの設定

/NORMAL ( 起動時のデフォルト )

イベントモードをトレースモードにします。

/PERFORMANCE

イベントモードをパフォーマンスモードにします。

- 形式 2

- デバッグ機能の設定

/CONFIG

デバッグ機能の設定を行います。

### ■ 機能

- 形式 1

イベントモードの設定を行います。

- トレースモード

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, シーケンサ関連のコマンド設定が有効となります。

- パフォーマンスモード

イベント機能をパフォーマンス測定に使用します。パフォーマンス関係のコマンド設定が有効となります。

EVENT関係のコマンドは全モードで使用可能ですが、それぞれのモードで異なった値を保持しています。したがって、モードを変更すると、そのモードで以前設定してあった値に戻ります。

また、モード変更をすると、トレース、パフォーマンスのすべてのバッファがクリアされます。起動時は、/NORMAL に設定されています。

- 形式 2

デバッグ機能の設定を行います。

- Real Time Memory モード

リアルタイムモニタ機能を有効にします。プログラム実行中に MCU を一度も停止させることなく、256 バイト × 2 領域分の内容をリアルタイムメモリウィンドウに表示できるようになります。

- RAM Checker モード

RAM チェッカ機能を有効にします。監視アドレスのアクセス履歴をログファイルに記録できるようになります。

選択可能なデバッグ機能はエミュレータまたはその接続形態によって変わります。選択された機能のみ使用可能となります。

また、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、Real Time Memory モードに設定されています。

### ■ 例

```
>SET MODE /PERFORMANCE
```

```
>SET MODE /CONFIG1
```

---

#### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「パフォーマンス測定」、「デバッグモード」を参照してください。

---



## 1.7 SHOW MODE

---

イベントモード, デバッグ機能の設定状態を表示します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW MODE

### ■ 機能

イベントモード, デバッグ機能の設定状態を表示します。

### ■ 例

```
>SHOW MODE
event mode : normal
debug function :
* 0: Real Time Memory
  1: RAM Checker
```

---

### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「パフォーマンス測定」, 「デバッグモード」を参照してください。

---

## 1.8 SET RADIX

---

デフォルト基数を表示します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET RADIX

#### ● コマンド修飾子

/BINARY

省略時の基数を 2 進数に設定します。

/OCTAL

省略時の基数を 8 進数に設定します。

/DECIMAL

省略時の基数を 10 進数に設定します。

/HEXADECIMAL ( デフォルト )

省略時の基数を 16 進数に設定します。

### ■ 機能

デフォルト基数を設定します。

### ■ 例

```
>SET RADIX/HEXADECIMAL
```

## 1.9 SHOW RADIX

---

現在の基数を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW RADIX

### ■ 機能

現在の基数を表示します。

### ■ 例

```
>SHOW RADIX  
default radix : hexadecimal
```

## 1.10 SET SOURCE

---

ソース行表示のモードを設定します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET SOURCE

#### ● コマンド修飾子

- ソース行表示モード

/DISPLAY ( 起動時のデフォルト )

ソース行表示を行うモードに設定します。

/NODISPLAY

ソース行表示を行わないモードに設定します。

### ■ 機能

逆アセンブルリストを表示する場合に、ソース行を付加して表示を行うか、行わないかを設定します。

起動時は、ソース行を表示するモードに設定されています。

### ■ 例

```
>SET SOURCE/DISPLAY
```

## 1.11 SHOW SOURCE

---

SET SOURCE コマンドで設定したソース行表示モードを表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW SOURCE

### ■ 機能

SET SOURCE コマンドで設定したソース行表示モードを表示します。

### ■ 例

```
>SHOW SOURCE  
source mode : display
```

## 1.12 SHOW SYSTEM

---

システム情報を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW SYSTEM

### ■ 機能

システム情報を表示します。

### ■ 例

```
> SHOW SYSTEM
FR Family Softune Workbench V60L07
Debugger type           = Emulator Debugger
MCU type                 = MB91V240
VCpu dll name           = C:\Softune\Bin\Wv911e2.dll
DSU type                 = DSU4
Common version           = V02L09
Monitor version          = V03L02R09
MCU frequency            = 2.000 MHz
Communication device     = USB
```

## 1.13 SET MAP (type 1)

メモリ空間のエリア種別とアクセス属性を設定します。

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET MAP { アドレス | アドレス範囲 }

#### ● パラメータ

アドレス (アドレス式)

属性を設定したいメモリアドレスを指定します。

アドレス範囲 (アドレス式)

属性を設定したいメモリ領域を指定します。

#### ● コマンド修飾子

- アクセス属性の指定

/READ

データリードアクセスを可能とします。

/WRITE

データライトアクセスを可能とします。

/CODE

コードリードアクセスを可能とします。

省略時は /READ/WRITE に設定されます。

### ■ 機能

メモリ空間のエリア種別とアクセス属性を設定します。

最大 31 領域まで設定することができます (1 バイト単位に指定可能です)。

LOAD コマンドによってロードモジュールファイルをロードした場合は、ファイルの情報を元にして自動的に属性を設定します。

各領域合計 128MB まで設定できます。

### ■ 例

```
>SET MAP/READ/WRITE 1000..1FFF
```

## 1.14 SET MAP (type 2)

---

メモリアクセス禁止領域を設定します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET MAP /INACCESSIBLE { アドレス | アドレス範囲 }

#### ● パラメータ

アドレス ( アドレス式 )

メモリアドレスを指定します。

アドレス範囲 ( アドレス式 )

メモリ領域を指定します。

#### ● コマンド修飾子

- アクセス属性の指定

/INACCESSIBLE

メモリアクセス禁止領域を設定します。

### ■ 機能

指定されたメモリアクセス禁止領域を設定します。

最大 16 領域まで設定可能です (1 バイト単位で指定可能)。

### ■ 例

```
>SET MAP/INACCESSIBLE 1000..1FFF
```



## 1.15 SHOW MAP (type 1)

---

メモリ空間のアクセス属性を表示します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SHOW MAP

### ■ 機能

メモリ空間のアクセス属性を表示します。

### ■ 例

```
>SHOW MAP
      address                attribute
00000000 .. 000011FF        read write
00001200 .. FFFFFFFF        undefined
FFFF0000 .. FFFFFFFF        read code
```

## 1.16 SHOW MAP (type 2)

---

指定されたアクセス禁止領域を表示します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW MAP /INACCESSIBLE

#### ● コマンド修飾子

- アクセス属性の指定

/INACCESSIBLE

メモリアクセス禁止領域を表示します。

### ■ 機能

指定されたメモリアクセス禁止領域を表示します。

### ■ 例

```
>SHOW MAP /INACCESSIBLE
no.  en/dis      address
1    enable      00000000..0000FFFF
2    enable      00044000..0007FFFF
```

# 1.17    ENABLE MAP

指定されたアクセス禁止領域を有効にします。

■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

ENABLE MAP/INACCESSIBLE [ マップ番号 ]

- パラメータ  
マップ番号  
アクセス禁止領域の番号を指定します。
- コマンド修飾子  
/INACCESSIBLE  
メモリアクセス禁止領域を有効にします。

■ 機能

指定されたアクセス禁止領域を有効にします。

■ 例

>ENABLE MAP /INACCESSIBLE 2

## 1.18 DISABLE MAP

---

指定されたアクセス禁止領域を無効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

DISABLE MAP/INACCESSIBLE [ マップ番号 ]

#### ● パラメータ

マップ番号

アクセス禁止領域の番号を指定します。

#### ● コマンド修飾子

/INACCESSIBLE

メモリアクセス禁止領域を無効にします。

### ■ 機能

指定されたアクセス禁止領域を無効にします。

### ■ 例

```
>DISABLE MAP /INACCESSIBLE 2
```

# 1.19 CANCEL MAP (type 1)

指定されたアドレスのエリアを未定義属性にします。

■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

■ 形式

CANCEL MAP [ アドレス | アドレス範囲 ]

● パラメータ

アドレス ( アドレス式 )

未定義属性にするアドレスを指定します。

アドレス範囲 ( アドレス式 )

未定義属性にするアドレス範囲を指定します。

● コマンド修飾子

/ALL

マップの設定をすべて未定義属性にします。

■ 機能

指定されたアドレスのエリアを未定義属性にします。

■ 例

>CANCEL MAP/ALL

## 1.20 CANCEL MAP (type 2)

---

指定されたアクセスの禁止領域を削除します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CANCEL MAP [ アドレス | アドレス範囲 ]

#### ● パラメータ

アドレス ( アドレス式 )

メモリアドレスを指定します。

アドレス範囲 ( アドレス式 )

メモリ領域を指定します。

#### ● コマンド修飾子

/INACCESSIBLE

メモリアクセス禁止領域を削除します。

/ALL

すべてのアクセス禁止領域を削除します。

### ■ 機能

指定されたアクセス禁止領域を削除します。

### ■ 例

```
>CANCEL MAP /INACCESSIBLE 00044000..0007FFFF
```

## 1.21 ENABLE VERIFYMODE

---

メモリ書込みを行う際のベリファイを有効に設定します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE VERIFYMODE

### ■ 機能

メモリ書込みを行う際のベリファイを有効に設定します。  
デバッガ起動時はベリファイが有効になっています。

### ■ 例

```
>ENABLE VERIFYMODE
```

## 1.22 DISABLE VERIFYMODE

---

メモリ書込みを行う際のベリファイを無効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

DISABLE VERIFYMODE

### ■ 機能

メモリ書込みを行う際のベリファイを無効に設定します。  
デバッガ起動時はベリファイが有効になっています。

### ■ 例

```
>DISABLE VERIFYMODE
```



## 1.23 SHOW VERIFYMODE

---

ベリファイモード状態を表示します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW VERIFYMODE

### ■ 機能

ベリファイモード ( コマンドでのメモリ書込み時のベリファイ動作が有効/無効を示すモード ) 状態を表示します。

### ■ 例

```
>SHOW VERIFYMODE  
verify mode : enable
```

## 1.24 SET INPORT

指定されたポートに対してデータを入力するように指定します。

### ■ 対応デバugga

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET INPORT ポートアドレス, マスクデータ, データ入力元 [, サイクル数]

#### ● パラメータ

ポートアドレス

ポートのアドレスを指定します。

マスクデータ

マスクデータを指定します。

マスクデータのビットが 1 であるところのみがポートデータとして有効となります。

データ入力元

入力ポートに対してプログラムから読み出されるかまたはプログラム命令実行サイクル数が指定サイクル数を越えた場合に与えるデータの入力元 ( ファイル名, 入力ターミナル ) を指定します。

入力ターミナルは \$TERMINAL と指定します。

ファイル名に \$TERMINAL を使用することはできません。

サイクル数 ( デフォルト 10 進数 )

命令実行サイクル数 (H'1 ~ H'FFFFFFFF) を指定します。

#### ● コマンド修飾子

- アクセス属性

/BYTE ( 省略時のデフォルト )

指定アドレスに 1 バイトアクセスを行ったときをポートアクセスとします。

/HALFWORD

指定アドレスに 2 バイトアクセスを行ったときをポートアクセスとします。

/WORD

指定アドレスに 4 バイトアクセスを行ったときをポートアクセスとします。

/ASCII

入力されたデータの文字コードをそのまま入力値にします。

/ASCII を指定するとアクセスサイズは, 常に /BYTE となります。

## ■ 機能

指定されたポートに対してプログラムから読み出されるたびまたはプログラム命令実行サイクル数が指定サイクル数を超えるたびに、ポートにデータを入力するように指定します。

パラメータにサイクル数を指定しなかった場合は、ポートがプログラムから読み出されるたびに指定されたデータ入力元からデータを読み出します。

パラメータにサイクル数を指定した場合は、ポートのアクセスとは無関係に一定サイクルごとにポートの内容が更新されます。

データ入力元がファイルの場合は、最後のデータを入力するとファイルの先頭に戻ります。

データ入力元が入力ターミナル(\$TERMINAL)の場合は、設定されたポートにリードアクセスされると、データ入力を要求するダイアログが表示されますので、入力データを入力します。

シミュレーションできるポートアドレスは 4096 個までです。

## ■ 例

```
>SET INPORT 0, 1F, INBUF0. DAT
>SHOW INPORT
```

address	bitpattern	size	cycle	input
00000000	0000001F	byte	-----	INBUF0. DAT
0000004F	0000000F	ascii	-----	\$terminal

## 1.25 SHOW INPORT

---

SET INPORT コマンドで設定されている内容を表示します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SHOW INPORT

### ■ 機能

SET INPORT コマンドで設定されている内容を表示します。

### ■ 例

```
>SHOW INPORT
address    bitpattern    size    cycle    input
000000FF  000000FF      byte    -----  $terminal
0000004F   0000000F     ascii    -----  $terminal
```

## 1.26 CANCEL INPORT

---

指定されたポートアドレスのポートのシミュレーションを解除します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

CANCEL INPORT [ ポートアドレス [ , ... ] ]

#### ● パラメータ

ポートアドレス

ポートのアドレスを指定します。

#### ● コマンド修飾子

/ALL

SET INPORT コマンドで設定されているすべての設定を解除します。

### ■ 機能

指定されたポートアドレスのポートのシミュレーションを解除します。

### ■ 例

```
>CANCEL INPORT/ALL
```

## 1.27 SET OUTPORT

指定されたポートに対してデータを出力するように指定します。

### ■ 対応デバugga

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET OUTPORT ポートアドレス, マスクデータ, データ出力先

#### ● パラメータ

ポートアドレス

ポートのアドレスを指定します。

マスクデータ

マスクデータを指定します。

マスクデータのビットが 1 であるところのみがポートデータとして有効となります。

データ出力先

出力ポートに対してプログラムから書き込まれたデータの出力先 ( ファイル名, 出力ターミナル ) を指定します。

出力ターミナルは \$TERMINAL と指定します。

ファイル名に \$TERMINAL を使用することはできません。

また, ファイル名同士でも同じ名前を使用することはできません。

#### ● コマンド修飾子

- アクセス属性

/BYTE ( 省略時のデフォルト )

指定アドレスに 1 バイトアクセスを行ったときをポートアクセスとします。

/HALFWORD

指定アドレスに 2 バイトアクセスを行ったときをポートアクセスとします。

/WORD

指定アドレスに 4 バイトアクセスを行ったときをポートアクセスとします。

/ASCII

データ出力先が \$TERMINAL 指定の場合, ポートに出力されたデータを, アスキーコードとみなして文字に変換して画面に表示します。

データ出力先がファイルの場合は, バイナリコードをそのまま出力します。

/ASCII を指定するとアクセスサイズは, 常に /BYTE となります。

## ■ 機能

指定されたポートに対してプログラムからデータを書き込まれるたびに、指定されたデータ出力先にデータを格納するように指定します。

データ出力先がファイルの場合は、ディスクの容量がいっぱいになるとエラーメッセージを表示し、以降ファイルへのデータの格納を行いません。

データ出力先が出力ターミナル(\$TERMINAL)の場合は、ポートに出力されたデータを 16 進数で画面に表示します。

ただし、コマンド修飾子 "/ASCII" が指定された場合は、データを文字コードとみなしその文字を表示します。

シミュレーションできるポートアドレスは 4096 個までです。

## ■ 例

```
>SET OUTPORT 0, 3F, OU1. DAT
>SHOW OUTPORT
```

address	bitpattern	size	output
00000000	0000003F	byte	OU1. DAT
00000040	000000FF	ascii	\$terminal

## 1.28 SHOW OUTPUT

---

SET OUTPUT コマンドで設定されている内容を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) ×
	(MB2198) ×
Monitor	×

### ■ 形式

SHOW OUTPUT

### ■ 機能

SET OUTPUT コマンドで設定されている内容を表示します。

### ■ 例

```
>SHOW OUTPUT
address      bitpattern    size    output
00000000    0000003F      byte    OU1. DAT
00000040    000000FF      ascii    $TERMINAL
```



## 1.29 CANCEL OUTPORT

---

指定されたポートアドレスのポートのシミュレーションを解除します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

CANCEL OUTPORT [ ポートアドレス [ , ... ] ]

#### ● パラメータ

ポートアドレス

ポートのアドレスを指定します。

#### ● コマンド修飾子

/ALL

SET OUTPORT コマンドで設定されているすべての設定を解除します。

### ■ 機能

指定されたポートアドレスのポートのシミュレーションを解除します。

### ■ 例

```
>CANCEL OUTPORT/ALL
```

## 1.30 SET INTERRUPT

---

割り込み発生条件を設定します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET INTERRUPT 割り込み番号, サイクル数

#### ● パラメータ

割り込み番号 (デフォルト 10 進数)

割り込みのベクタ番号を指定します。

使用可能な割り込みは, 使用しているチップのマニュアルを参照してください。

サイクル数 (デフォルト 10 進数)

命令実行サイクル数 (H'1 ~ H'FFFFFFFF) を指定します。

#### ● コマンド修飾子

/INTERVAL

割り込みが周期的に発生するように指定します。

### ■ 機能

プログラムが指定した命令実行サイクル数以上実行されたときに指定された割り込みを発生させ, 割り込み発生条件を解除します。

"/INTERVAL" を指定した場合は, プログラム実行中に, 指定した命令実行サイクル数ごとに指定された割り込みを発生させるようにします。

割り込み発生条件は, CANCEL INTERRUPT コマンドで解除されるまで有効です。

### ■ 例

```
>SET INTERRUPT 4, 12367
```

## 1.31 SHOW INTERRUPT

---

SET INTERRUPT コマンドで設定されている割込み発生条件を表示します。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SHOW INTERRUPT

### ■ 機能

SET INTERRUPT コマンドで設定されている割込みのベクタ番号, サイクル数 (10 進数), INTERVAL の有無の順で表示します。

INTERVAL の有無は次の表記で表示します。

shot : INTERVAL なし (一度だけ割込みが入る指定)

interval : INTERVAL あり (繰り返し割込みが入る指定)

### ■ 例

```
>SHOW INTERRUPT
no      cycle      kind
18      1258       shot
22      9823       interval
```

## 1.32 CANCEL INTERRUPT

---

SET INTERRUPT コマンドで設定されている割り込み発生条件を解除します。

---

### ■ 対応デバugga

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

CANCEL INTERRUPT [ 割り込み番号 [, ... ] ]

#### ● パラメータ

割り込み番号 ( デフォルト 10 進数 )

割り込みのベクタ番号を指定します。

#### ● コマンド修飾子

/ALL

SET INTERRUPT コマンドで設定されているすべての割り込み発生条件を解除します。

### ■ 機能

SET INTERRUPT コマンドで設定されている割り込み発生条件を解除します。

### ■ 例

```
>CANCEL INTERRUPT/ALL
```

### 1.33 COPY VECTOR

ベクタテーブル初期領域の値を指定された領域へコピーします。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

COPY VECTOR [ ベクタ番号 ] [ , TBR 値 ]

● パラメータ

ベクタ番号 ( デフォルト 10 進数 )

コピーするベクタ番号を指定します。

TBR 値

コピーするベクタテーブルの領域アドレスを指定します。

0xFFFFFFFF 以上のアドレスを指定するとエラーとなります。

省略すると現在のレジスタ TBR の値となります。

● コマンド修飾子

/ALL ( 省略時のデフォルト )

すべての EIT ベクタ値をコピーします。

/MINIMUM

デバッガが使用する最低限の EIT ベクタ値をコピーします。

■ 機能

ベクタテーブル初期領域の値を指定された領域へコピーします。

ベクタテーブル初期領域 (TBR 値 0x000ffc00 で示される領域) の値を TBR 値で指定された領域へコピーします。

レジスタ TBR を書換え、ベクタテーブルの位置を変更する場合は、ベクタテーブル初期領域の値を、コピーする必要があります。

■ 例

```
>COPY VECTOR 11, 100C00
>COPY VECTOR /MINIMUM
```

## 1.34 SET VECTOR

---

ベクタ番号の内容を設定します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET VECTOR ベクタ番号, 設定アドレス値

#### ● パラメータ

ベクタ番号

設定するベクタ番号を設定します。

設定アドレス

該当ベクタ番号のルーチン先頭アドレス値です。

### ■ 機能

指定された領域のベクタ番号のアドレス値を設定します。

### ■ 例

```
>SET VECTOR 11, 0FF100
>SHOW VECTOR 11..11
VectorNo.      Address      Symbol
11             00FF100
```

## 1.35 SHOW VECTOR

ベクタ番号の内容を表示します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW VECTOR [ ベクタ番号の範囲 ]

#### ● パラメータ

ベクタテーブルの範囲

表示するベクタ番号の範囲を指定します。

指定方法は、[ ( 始めの番号 ) .. ( 終わりの番号 ) ]

### ■ 機能

ベクタ番号の内容を表示します。

続けて省略して行くと、続きのベクタ番号から表示します。

### ■ 例

```
>SHOW VECTOR 6..8
VectorNo.   Address      Symbol      Factor
6           00000000
7           FF201000    co_1000     Co - processor Absence
8           FF110000    CO_ERROR    Co - processor error
```

## 1.36 ENABLE WATCHDOG

---

ウォッチドッグタイマ機能を有効にします。

---

### ■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE WATCHDOG

### ■ 機能

ウォッチドッグタイマ機能を有効にします。

DSU3 および DSU4 チップのみこの指定ができます。

### ■ 例

```
>ENABLE WATCHDOG
```



# 1.37     DISABLE WATCHDOG

---

ウォッチドッグタイマ機能を無効にします。

---

■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

■ 形式

DISABLE WATCHDOG

■ 機能

ウォッチドッグタイマ機能を無効にします。  
DSU3 および DSU4 チップのみこの指定ができます。

■ 例

>DISABLE WATCHDOG

## 1.38 SHOW WATCHDOG

---

ウォッチドッグタイマ機能の有効 / 無効を表示します。

---

### ■ 対応デバッグ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW WATCHDOG

### ■ 機能

ウォッチドッグタイマ機能の有効 / 無効を表示します。

DSU3 および DSU4 チップのみこの指定ができます。

### ■ 例

```
>SHOW WATCHDOG  
watchdog : enable
```

### 1.39 SET EXTERNALMEMORY

チップセレクト番号を指定して，外部 ROM または外部 RAM の代行メモリを指定します。

■ 対応デバッグ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

■ 形式

SET EXTERNALMEMORY {/ROM{/RAM} チップセレクト番号

● パラメータ

チップセレクト番号 ( デフォルト 16 進数 )

代行メモリを割当てるメモリ領域の，チップセレクト番号を指定して設定します。

● コマンド修飾子

/ROM ( 省略時のデフォルト )

指定したチップセレクト領域を ROM 領域として使用します。

/RAM

指定したチップセレクト領域を RAM 領域として使用します。

■ 機能

外部のメモリに対して，1 領域だけ代行メモリを割当てることができます。

領域の指定は，チップセレクト番号を指定して行います。チップセレクトの機能については，ご使用の LSI 仕様書を参照してください。

コマンド修飾子として /ROM を指定した場合には，ユーザプログラムからの書込みを抑止しますが，ブレークの要因にはなりません。コマンド修飾子を省略した場合には，/ROM として設定します。

本機能は外部メモリ用に専用端子を持つ評価用チップでのみ使用でき，使用可能か不可かはエミュレータが自動的に判定します。

また，代行メモリは，アダプタユニット上に実装されますので，ハードウェアに対する設定やメモリサイズについては，ご使用のアダプタユニットのハードウェアマニュアルを参照してください。

■ 例

```
> SET EXTERNALMEMORY/ROM 1
> SHOW EXTERNALMEMORY
enable CS1 : ROM
```

## 1.40 ENABLE EXTERNALMEMORY

---

外部メモリエミュレーション機能を有効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

ENABLE EXTERNALMEMORY

### ■ 機能

外部メモリエミュレーション機能を有効にします。

### ■ 例

```
> ENABLE EXTERNALMEMORY
> SHOW EXTERNALMEMORY
enable CS1 : ROM
```

## 1.41 DISABLE EXTERNALMEMORY

---

外部メモリエミュレーション機能を無効にします。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

DISABLE EXTERNALMEMORY

### ■ 機能

外部メモリエミュレーション機能を無効にします。

### ■ 例

```
> DISABLE EXTERNALMEMORY
> SHOW EXTERNALMEMORY
disable CS1 : ROM
```

## 1.42 SHOW EXTERNALMEMORY

---

外部メモリエミュレーション機能の設定状態を表示します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW EXTERNALMEMORY

### ■ 機能

外部メモリエミュレーション機能の設定状態を表示します。

### ■ 例

```
>SHOW EXTERNALMEMORY  
CS1 : ROM  
disable CS1 : ROM
```

## 1.43 SET WATCH

指定された変数をウォッチウィンドウへ登録します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET WATCH 変数名 [, ウォッチウィンドウ番号 ]

#### ● パラメータ

変数名

ウォッチウィンドウに登録する変数または式を指定します。

ウォッチウィンドウ番号

変数を追加するウォッチウィンドウの番号を指定します。

省略した場合は、ウォッチウィンドウ 1 が指定されます。

#### ● コマンド修飾子

##### • 基数

/BINARY

変数値を 2 進数で表示することを指定します。

/OCTAL

変数値を 8 進数で表示することを指定します。

/DECIMAL

変数値を 10 進数で表示することを指定します。

/HEXADECIMAL

変数値を 16 進数で表示することを指定します。

##### • ウォッチモード

/AUTOMATIC

変数をデバッグ環境で設定されているウォッチモードで解釈します。

デバッグ環境の設定を変更すると、変数のウォッチモードでも変更されます。

/C

変数を C/C++ 言語式として解釈します。

/ASSEMBLER

変数をアセンブラ式として解釈します。

- データサイズ

/BYTE

アセンブラモード時の表示をバイト長にすることを指定します。

/HALFWORD

アセンブラモード時の表示を 2 バイト長にすることを指定します。

/WORD

アセンブラモード時の表示を 4 バイト長にすることを指定します。

/DWORD

アセンブラモード時の表示を 8 バイト長にすることを指定します。

/SINGLE

アセンブラモード時の表示を単精度浮動小数点数にすることを指定します。

/DOUBLE

アセンブラモード時の表示を倍精度浮動小数点数にすることを指定します。

- 個別モニタリング設定

/MONITORING ( 省略時のデフォルト )

変数の個別モニタリング設定を ON にします。

/NOMONITORING

変数の個別モニタリング設定を OFF にします。

### ■ 機能

変数を指定されたウォッチウィンドウへ登録します。既に登録されている変数を指定した場合は、同じ変数が複数登録されます。

基数、ウォッチモード、データサイズのコマンド修飾子を省略した場合は、デバッグ環境で指定された設定となります。

データサイズの指定は、/ASSEMBLER の設定が指定された場合にのみに有効です。

ウォッチポイントの設定はデバッグ起動時に以前の情報が復元されますので、デバッグ起動時などにバッチファイルでウォッチポイントを設定している場合は、始めに CANCEL WATCH/ALL 等で一度全部のウォッチポイントを削除するようにしてください。

### ■ 例

```
>SET WATCH strsym.a, 1
```

```
>SET WATCH/HEXADECIMAL/ASSEMBLER/WORD/NOMONITORING LABEL1, 1
```



## 1.44 CANCEL WATCH

指定された変数をウォッチウィンドウから削除します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CANCEL WATCH 変数名 [, ウォッチウィンドウ番号]

CANCEL WATCH/ALL [, ウォッチウィンドウ番号]

#### ● パラメータ

変数名

ウォッチウィンドウから削除する変数または式を指定します。

ウォッチウィンドウ番号

変数を削除するウォッチウィンドウの番号を指定します。

省略した場合は、ウォッチウィンドウ 1 が指定されます。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

指定されたウィンドウからすべてのウォッチポイントを削除します。

### ■ 機能

指定されたウォッチウィンドウの変数をへ削除します。同名の変数が複数存在する場合は、最初に登録する変数のみ削除します。

### ■ 例

```
>CANCEL WATCH flag, 1
```

```
>CANCEL WATCH/ALL 2
```

## 1.45 SET FREQUENCY

---

CPU の最大動作周波数を設定します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET FREQUENCY /MAX 最大動作周波数

#### ● パラメータ

最大動作周波数（デフォルト 10 進数）

CPU の最大動作周波数を設定します。

この設定は最大周波数を設定するものであり、実際の動作周波数を変更するものではありません。

#### ● コマンド修飾子

/MAX（省略不可）

CPU の最大動作周波数を設定します。

/RESET

設定した最大動作周波数を CPU 情報ファイルのデフォルト値に戻します。

### ■ 機能

CPU の最大動作周波数を設定します。

設定した最大動作周波数を基に、最適なオートウェイト値を自動的に算出し設定します。

これにより、デバッグ時に最適な応答速度が設定されます。

### ■ 例

```
>SET FREQUENCY /MAX 64
```

---

#### < 注意事項 >

実際に使用する動作周波数より低い値を設定すると、エミュレータが誤動作する可能性があります。

---

## 1.46 SHOW FREQUENCY

---

SET FREQUENCY コマンドで設定されている最大動作周波数を表示します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW FREQUENCY

#### ● コマンド修飾子

/MAX (省略不可)

CPU の最大動作周波数を表示します。

### ■ 機能

SET FREQUENCY コマンドで設定されている , CPU の最大動作周波数を表示します。

max frequency : 現在の CPU の最大動作周波数

### ■ 例

```
>SHOW FREQUENCY/MAX  
max frequency:33MHz
```

## 1.47 SET REALTIMEMEMORYAREA

---

リアルタイムメモリ領域を設定します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET REALTIMEMEMORYAREA 領域番号, 領域アドレス

#### ● パラメータ

領域番号 (デフォルト 10 進数)

設定するリアルタイムメモリ領域番号 (1, 2) を指定します。

領域アドレス (デフォルト 10 進数)

設定するリアルタイムメモリのアドレスを指定します。

### ■ 機能

リアルタイムでモニタリングが可能な、リアルタイムメモリ領域を設定します。

入力した領域アドレスを基に 256byte 分、2 領域を設定できます。

### ■ 例

```
>SET REALTIMEMEMORYAREA 1,1000
```

---

#### < 注意事項 >

- 指定した領域番号に既にリアルタイムメモリ領域が設定されていた場合、上書き設定を行います。
  - 上書きによりリアルタイムメモリ領域が無効になった領域にウォッチ変数が登録されていた場合、下記のメッセージが表示されます。  
「リアルタイムモニタリングが無効となったウォッチ変数があります」
-

## 1.48 SHOW REALTIMEMEMORYAREA

---

SET REALTIMEMEMORYAREA コマンドで設定されているリアルタイムメモリ領域を表示します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW REALTIMEMEMORYAREA

### ■ 機能

SET REALTIMEMEMORYAREA コマンドで設定されている、リアルタイムメモリ領域を表示します。

未設定の場合は「not found」と表示されます。

### ■ 例

```
>SHOW REALTIMEMEMORYAREA
no.      address
1    0003A200 .. 0003A2FF
2    0003B500 .. 0003B5FF
```



## 第2章

---

# プログラム実行コマンド

この章では、プログラム実行コマンドについて説明します。

- 2.1 GO
- 2.2 STEP
- 2.3 SET STEP
- 2.4 SHOW STEP
- 2.5 CALL
- 2.6 CLEAR CALL
- 2.7 SHOW STATUS

## 2.1 GO

---

指定した開始アドレスからプログラムを実行します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

GO [ 開始アドレス ] [, ブレークアドレス ]

#### ● パラメータ

開始アドレス (アドレス式)

プログラムの実行開始アドレスを指定します。

ブレークアドレス (アドレス式)

プログラムの実行停止アドレスを指定します。

#### ● コマンド修飾子

- リターン指定

/RETURN

現在実行中の関数から親関数に戻る位置までプログラムを実行します。

この機能は、C/C++ 言語で作成されたプログラムのみ使用できます。

また、最適化されたプログラムは正常に止まらない場合があります。

### ■ 機能

指定した開始アドレスからプログラムを実行します。

開始アドレスが省略された場合は、現在のプログラムカウンタの示すアドレスから実行します。

GO コマンドで設定されたブレークアドレスは、実行停止時に自動的に削除されます。

コマンド修飾子を省略すると、SET GO コマンドの設定により実行を開始します。

### ■ 例

```
>GO power$20
Break at main$10
>GO power$20, main$5
```



## 2.2 STEP

ソース行単位または機械命令単位でプログラムを実行します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

STEP [ ステップ数 ]

#### ● パラメータ

ステップ数 ( デフォルト 10 進数 )

STEP コマンドの実行回数 (H'1 ~ H'FFFFFFF) を指定します。

ステップ数が省略された場合は , 実行回数は 1 回となります。

#### ● コマンド修飾子

- ステップ単位の指定

/INSTRUCTION

機械命令を 1 命令ずつ実行します。

/LINE

ソース行単位で実行します。

/AUTOMATIC ( 起動時のデフォルト )

ソースウィンドウの表示モードに従って実行単位を次のように自動的に変更します。

- ソースウィンドウがソース行のみ表示の場合は , ソース行単位 (/LINE) に実行します。

- それ以外の場合は , 機械命令単位 (/INSTRUCTION) に実行します。

- ステップ動作の指定

/INTO

呼び出された関数 , サブルーチンおよび割込みハンドラの中まで 1 ステップずつ実行します。

/OVER

/LINE 時の関数呼び出し , /INSTRUCTION 時のサブルーチンコール命令 (CALL 等 ) およびソフト割込み命令 (INT 等 ) を 1 ステップとして実行します。

### ■ 機能

SET STEP コマンドで設定された条件により、ソース行単位または機械命令単位でプログラムを実行します。

コマンド修飾子を指定することにより、SET STEP コマンドで設定された条件を無視できます。

### ■ 例

```
>STEP  
>STEP/INSTRUCTION
```

## 2.3 SET STEP

STEP コマンドでコマンド修飾子を省略した場合のステップ実行条件を指定します。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET STEP

#### ● コマンド修飾子

- ステップ単位の指定

/INSTRUCTION

機械命令を 1 命令ずつ実行します。

/LINE

ソース行単位で実行します。

/AUTOMATIC ( 起動時のデフォルト )

ソースウィンドウの表示モードに従って実行単位を次のように自動的に変更します。

- ソースウィンドウがソース行のみ表示の場合は、ソース行単位 (/LINE) に実行します。
- それ以外の場合は、機械命令単位 (/INSTRUCTION) に実行します。

- ステップ動作の指定

/INTO ( 起動時のデフォルト )

呼び出された関数、サブルーチンおよび割込みハンドラの中まで 1 ステップずつ実行します。

/OVER

/LINE 時の関数呼出し、/INSTRUCTION 時のサブルーチンコール命令 (CALL 等) およびソフト割込み命令 (INT 等) を 1 ステップとして実行します。

### ■ 機能

STEP コマンドでコマンド修飾子を省略した場合のステップ実行条件を指定します。

本プログラム起動時は、AUTOMATIC, INTO の状態になっています。

### ■ 例

>SET STEP/INSTRUCTION

## 2.4 SHOW STEP

---

現在の STEP コマンドのステップ実行条件を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW STEP

### ■ 機能

現在の STEP コマンドのステップ実行条件を表示します。

### ■ 例

```
>SHOW STEP  
step mode : instruction, into
```

## 2.5 CALL

---

指定した関数を実行し、戻り値を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CALL 関数名 ([ 引数 [, ... ] ])

#### ● パラメータ

関数名

呼び出したい関数名を指定します。

引数

C/C++ 言語の引数に準拠します。

ただし、引数として構造体 ( ストラクチャ )、共用体 ( ユニオン ) およびクラスの変数名は指定できません。

#### ● コマンド修飾子

/DISPLAY ( 起動時のデフォルト )

戻り値の表示を指定します。

/NODISPLAY

戻り値の非表示を指定します。

### ■ 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。

CALL コマンドによる関数実行中に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 GO コマンドで再実行させれば、CALL コマンドの実行を継続します。

CALL コマンドの実行を中止する場合は、CLEAR CALL コマンドを使用します。なお、このコマンドはネストすることができません。

CALL コマンドは、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 CALL コマンド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、次のメッセージが表示されます。

**Break at アドレス by Invalid call termination**

GO コマンドで実行を再開させれば CALL コマンドの実行を継続します。

### ■ 制限事項

CALL コマンドにより、レジスタ、メモリ、IO などのリソースは関数呼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出し前の内容を保持し、関数実行後または CLEAR CALL コマンドにより復元します。

その他のリソースに関しては復元しません。

### ■ 例

```
>CALL debug (cmd, p)
return value is H'0001
```

## 2.6 CLEAR CALL

---

CALL コマンドを中断し、レジスタを呼び出す前の状態に戻します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CLEAR CALL

### ■ 機能

CALL コマンドを中断し、レジスタを呼び出す前の状態に戻します。

### ■ 例

```
>CALL debug (cmd, p)
Break at 00FF0F20 by breakpoint
>CLEAR CALL
```

## 2.7 SHOW STATUS

---

MCU の実行状態を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW STATUS

### ■ 機能

MCU の実行状態を表示します。

MCU がブレイクしている場合には、直前のプログラム実行のブレイク要因を表示します。

### ■ 例

```
>SHOW STATUS
MCU status : executing
>SHOW STATUS
break at 0000FF00 by breakpoint
```



## 第3章

---

# ブレーク/イベント 制御コマンド

この章では、ブレーク/イベント制御コマンドについて説明します。

- 3.1 SET BREAK (type 1)
- 3.2 SET BREAK (type 2)
- 3.3 SET BREAK (type 3)
- 3.4 SHOW BREAK
- 3.5 CANCEL BREAK
- 3.6 ENABLE BREAK
- 3.7 DISABLE BREAK
- 3.8 SET DATABREAK (type 1)
- 3.9 SET DATABREAK (type 2)
- 3.10 SHOW DATABREAK
- 3.11 CANCEL DATABREAK
- 3.12 ENABLE DATABREAK
- 3.13 DISABLE DATABREAK
- 3.14 SET EVENT
- 3.15 SHOW EVENT
- 3.16 CANCEL EVENT
- 3.17 ENABLE EVENT
- 3.18 DISABLE EVENT
- 3.19 SET CODEEVENT
- 3.20 SHOW CODEEVENT
- 3.21 CANCEL CODEEVENT
- 3.22 ENABLE CODEEVENT

### 第3章 ブレーク/イベント制御コマンド

- 3.23 DISABLE CODEEVENT
- 3.24 SET DATAEVENT
- 3.25 SHOW DATAEVENT
- 3.26 CANCEL DATAEVENT
- 3.27 ENABLE DATAEVENT
- 3.28 DISABLE DATAEVENT
- 3.29 SET SEQUENCE (type 1)
- 3.30 SET SEQUENCE (type 2)
- 3.31 SHOW SEQUENCE
- 3.32 CANCEL SEQUENCE
- 3.33 ENABLE SEQUENCE
- 3.34 DISABLE SEQUENCE
- 3.35 SET TRIGGER
- 3.36 SHOW TRIGGER
- 3.37 ENABLE ALIGNMENTBREAK
- 3.38 DISABLE ALIGNMENTBREAK
- 3.39 SHOW ALIGNMENTBREAK
- 3.40 SET BREAKCONDITION (type 1)
- 3.41 SET BREAKCONDITION (type 2)
- 3.42 SHOW BREAKCONDITION

## 3.1 SET BREAK (type 1)

指定したブレークアドレスにブレークポイントを設定します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

SET BREAK ブレークアドレス [ パスカウント数 ] [ { コマンド: コマンド... } ]

#### ● パラメータ

ブレークアドレス (アドレス式)

ブレークポイントを設定するアドレスを指定します。

パスカウント数 (デフォルト 10 進数)

ブレークポイントに何回ヒットすると実行を停止するかを設定します (シミュレータデバッガでは 1 ~ 65535, エミュレータデバッガ (FR60Lite, FR80S) では 1 ~ 255)。省略時は 1 になります。

デバッガ種別がシミュレータデバッガまたはエミュレータデバッガ (FR60Lite, FR80S) で有効となります。その他のエミュレータデバッガ, モニタデバッガでは無視されます。

コマンド

ブレークアドレスにヒットした場合に実行したいコマンドリストを指定します。セミコロンを使用することにより、複数のコマンドが指定できます。

デバッガ種別がシミュレータデバッガでのみ有効となります。エミュレータデバッガ, モニタデバッガでは無視されます。

#### ● コマンド修飾子

【Simulator】

/BREAK (省略時のデフォルト)

ブレークポイントにヒットした場合、コマンドリストを処理したあと命令実行を停止します。

/NOBREAK

ブレークポイントにヒットした場合、コマンドリストを処理したあと命令実行を再開します。

#### 【Emulator / Monitor】

- ブレークポイント種別指定

/SOFT

ソフトウェアによるブレークポイントを指定します。

#### 【Emulator】

/COUNT (FR60Lite, FR80S)

パスカウント数の設定を許可します。エミュレータデバッガ (FR60Lite, FR80S) でのみ有効です。

/HARD ( 起動時のデフォルト )

ハードウェアによるブレークポイントを指定します。ROMパッチ機能を有するチップでは、ハードウェアブレーク 5 ポイントすべてを使用している場合に、ROM パッチ機能をハードウェアブレークとして使用できます。詳しくは SOFTUNE Workbench 機能説明書「2.3.4.1 コードブレーク」を参照してください。MB2198 の場合「実行中のブレークポイント設定」が有効の場合のみ指定できます。

## ■ 機能

指定したブレークアドレスにブレークポイントを設定します。

ブレークポイントの設定数は次のとおりです。

Simulator	65535 個
Emulator ( MB2197 )	4096 個 ( ソフトウェア ) , 5 個 ( ハードウェア )
Emulator ( MB2198 )	4096 個 ( ソフトウェア ) , 5 個 ( ハードウェア )
	2 個 ( ハードウェア / カウント )
Monitor	16 個 ( ソフトウェア )

## ■ 例

[Simulator]

```
>SET BREAK/NOBREAK main, 1, {SHOW TRACE : SHOW TIMER}
```

[Emulator]

```
>SET BREAK 00ff0200
```

```
>SET BREAK 00ff0300,3
```

---

< 注意事項 >

- ソフトウェアブレークでは、ROM のような書込みが正しく行えない領域には、ブレークポイントを設定できません。この場合、プログラム実行時にペリファイエラーが発生します。
  - ソフトウェアブレークポイントは命令の先頭アドレスに設定してください。ほかのアドレスにブレークポイントを設定すると、プログラムが暴走する場合があります。
  - ROM パッチブレークを使用するには、ブレークとして使用する ROM パッチのリソースを設定しておく必要があります。詳しくは、SOFTUNE Workbench 操作マニュアル「4.6.4 ブレークポイント」を参照してください。
  - ROM パッチブレークが設定可能なメモリ領域は内蔵 ROM 領域のみです。それ以外の領域を指定した場合は、ブレークを登録できません。
  - ROM パッチブレークにデータ監視条件を設定することはできません。
  - ROM パッチブレークは、必ず命令の先頭アドレスに設定してください。命令の途中でブレークポイントを設定すると、CPU が正しく命令を解釈できず誤動作する場合があります。
  - ROM パッチブレークを設定したアドレスを実行中に読み出すと、ブレーク命令として読み出されます。そのため、実行中にブレークアドレスを読み出すような処理を行う場合は、ROM パッチブレークを削除または無効にしてから行ってください。
  - 実行中に ROM パッチブレークを設定する場合、ハードウェアブレークの設定時に比べ、実行停止時間が長くなります。
  - コマンドリストには下記の実行系のコマンドは指定できません。
    - GO
    - STEP
    - CALL
    - SYSTEMCALL
  - STEP/INTO および STEP/OVER では STUB 機能 ( コマンド実行および命令実行再開 ) は行いません。ブレークポイントにヒットしたあと実行を停止します。
-

## 3.2 SET BREAK (type 2)

指定したブレークアドレスにデータ監視ブレークポイント(ソフトウェア)を設定します。

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET BREAK /BREAKCONDITION ブレークアドレス [, パスカウント数]

#### ● パラメータ

ブレークアドレス (アドレス式)

ブレークポイントを設定するアドレスを指定します。

パスカウント数

ブレークポイントに何回ヒットすると実行を停止するかを設定します (1 ~ 255)。  
省略時は 1 になります。

#### ● 機能分類コマンド修飾子

/BREAKCONDITION

SET BREAKCONDITION で設定した条件を結びつけて、ブレーク条件にします。

#### ● コマンド修飾子

- ・ ブレークポイント種別指定

/HARD (起動時のデフォルト)

ハードウェアによるブレークポイントを指定します。

/COUNT (FR60Lite, FR80S)

パスカウント数の設定を許可します。エミュレータデバッガ (FR60Lite, FR80S) でのみ有効です。

### ■ 機能

指定したブレークアドレスにデータ監視ブレークポイント(ソフトウェア)を設定します。

SET BREAKCONDITION で設定した条件を結びつけて、ブレーク条件にします。

指定したブレークアドレスを通過したとき、SET BREAKCONDITION で指定したブレーク条件をチェックしてブレーク判定をおこないます。

### ■ 例

```
> SET BREAK /BREAKCONDITION 00ff0200
```

### 3.3 SET BREAK (type 3)

指定したブレークアドレスにデータ監視ブレークポイント (ハードウェア) を設定します。

#### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

#### ■ 形式

SET BREAK /DATAWATCH コードアドレス, データアドレス [&= アドレスマスク]  
[, [!] d= データ [&= データマスク ]]

##### ● パラメータ

コードアドレス (アドレス式)

データ監視条件となるコードのアドレスを指定します。

データアドレス (アドレス式)

データ監視条件となるデータのアドレスを指定します。

アドレスマスク (データ式)

データ監視条件となるデータのアドレスに対するマスクビットパターンを指定します。1 が立っているビット位置のアドレスのみを比較対象にします。

データ (データ式)

データ監視条件となるデータのアドレスの内容を指定します。

/ANYTHING を指定した場合, 32 ビットデータを比較対象にします。

(8/16 ビットデータを比較する場合は, マスク指定する必要があります)

データマスク (データ式)

データ監視条件となるデータに対するマスクビットパターンを指定します。1 が立っているビット位置のデータ値のみを比較対象にします。

##### ● コマンド修飾子

- アクセスサイズの指定

/BYTE

バイトアクセスをデータ監視条件に指定します。

/HALFWORD

ハーフワード (2 バイト) アクセスをデータ監視条件に指定します。

/WORD

ワード (4 バイト) アクセスをデータ監視条件に指定します。

/ANYTHING (省略時のデフォルト)

データ長にかかわらず、指定アドレスがアクセスされたときにブレークします。

- アクセス属性の指定

/READ

データリードアクセスをデータ監視条件に指定します。

/WRITE

データライトアクセスをデータ監視条件に指定します。

省略時は /READ/WRITE に設定されます。

#### ■ 機能

指定したブレークアドレスにデータ監視ブレークポイント(ハードウェア)を設定します。

データ監視ブレークとは、指定したコードアクセスに実行がきたとき、データアドレス内のデータを参照して、ブレーク判定をおこなう機能です。

最大4ポイント設定ができますが、SET EVENT と SET TRACETRIGGER の設定によって使用できるポイント数は変動します。

#### ■ 例

```
>SET BREAK /DATAWATCH/WRITE/WORD 00ff0200, 120034&=fffffc,  
!d=1234&=feff
```

---

#### < 注意事項 >

1. FR60Lite使用時以外はこのコマンドは使用できません。詳しくはSOFTUNE Workbench 機能説明書「ブレーク」を参照してください。
  2. /ANYTHING 修飾子を指定した場合、32ビットデータを比較対象にします。  
8/16ビットデータを比較する場合は、マスクデータを指定する必要があります。  
例) 8ビットデータ (0x12) の場合  
SET BREAK /DATAWATCH/ANYTHING 0x10000, D=0x12000000&=0xff000000  
16ビットデータ (0x4567) の場合  
SET BREAK /DATAWATCH/ANYTHING 0x10000, D=0x45670000&=0xffff0000
  3. このコマンドは、イベントモードがパフォーマンスモードの場合、使用できません
-



## 3.4 SHOW BREAK

---

SET BREAK コマンドで設定されているブレークポイントを表示します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW BREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)  
ブレークポイント番号を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

すべてのブレークポイントを表示します。

/NORMAL (FR60Lite, FR80S)

ハードウェアまたはソフトウェアブレークの情報のみを表示します。

/COUNT (FR60Lite, FR80S)

ハードウェア / カウントブレークの情報のみを表示します。

/DATAWATCH (FR60Lite)

データ監視ブレークの情報のみを表示します。イベントモードがパフォーマンスモードの場合、指定できません。

### ■ 機能

SET BREAK コマンドで設定されているブレークポイントを表示します。

## ■ 例

[Simulator]

>SHOW BREAK

no.	en/dis	address	pass-count	symbol
1	enable	00FF0F00	1 ( 1)	
4	disable	00FF20DE	65535 ( 1234)	

Control: BREAK

Command: show timer

[MB2198]

>show break/all

no.	en/dis	ad	dress	kind	cond.	symbol
1	enable	0000F000	hard			

count

no.	en/dis	ad	dress	kind	pass	-count	cond.	symbol
1	enable	00FF0F00	hard		1 ( 1)			* main
4	disable	00FF20DE	hard		65535(1234)			func

data watch

no.	en/dis	address	data-addr	d-add-mask	data	mask	size	access	symbol
1	enable	00FF3032	00008000	-----	00000034	!000000FF	byte	read	

### < 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。

## 3.5 CANCEL BREAK

指定された番号のブレークポイントを解除します。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

CANCEL BREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW BREAK コマンドで参照します。

#### ● コマンド修飾子

/ALL

すべてのブレークポイントを解除します。

/NORMAL (FR60Lite, FR80S)

ハードウェアまたはソフトウェアブレークを解除します。

/COUNT (FR60Lite, FR80S)

ハードウェア / カウントブレークを解除します。

/DATAWATCH (FR60Lite)

データ監視ブレークを解除します。

### ■ 機能

指定された番号のブレークポイントを解除します。

### ■ 例

```
>CANCEL BREAK 1
```

```
>CANCEL BREAK 3,4
```

## 3.6 ENABLE BREAK

指定された番号のブレークポイントを有効にします。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

ENABLE BREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW BREAK コマンドで参照します。

#### ● コマンド修飾子

/ALL

すべてのブレークポイントを有効にします。

/NORMAL (FR60Lite, FR80S)

ハードウェアまたはソフトウェアブレークを有効にします。

/COUNT (FR60Lite, FR80S)

ハードウェア / カウントブレークを有効にします。

/DATAWATCH (FR60Lite)

データ監視ブレークを有効にします。

### ■ 機能

指定された番号のブレークポイントを有効にします。

### ■ 例

```
>ENABLE BREAK 2
```

```
>ENABLE BREAK 3, 4
```

## 3.7 DISABLE BREAK

指定された番号のブレークポイントを無効にします。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

DISABLE BREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW BREAK コマンドで参照します。

#### ● コマンド修飾子

/ALL

すべてのブレークポイントを無効にします。

/NORMAL (FR60Lite, FR80S)

ハードウェアまたはソフトウェアブレークを無効にします。

/COUNT (FR60Lite, FR80S)

ハードウェア / カウントブレークを無効にします。

/DATAWATCH (FR60Lite)

データ監視ブレークを無効にします。

### ■ 機能

指定された番号のブレークポイントを無効にします。

### ■ 例

```
>DISABLE BREAK 2
```

```
>DISABLE BREAK 3, 4
```

## 3.8 SET DATABREAK (type 1)

指定したアドレスをデータアクセスする場合にブレークします。

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET DATABREAK データアクセスアドレス [, パスカウント] [, { コマンド: コマンド ... } ]

#### ● パラメータ

データアクセスアドレス (アドレス式)

データアクセスブレークポイントを設定するアドレスを指定します。

パスカウント数 (デフォルト 10 進数)

ブレークポイントに何回ヒットすると実行を停止するかを設定します。(1 ~ 65535)  
省略時は 1 になります。

コマンド

ブレークアドレスにヒットした場合に実行したいコマンドリストを指定します。セミコロンを使用することにより、複数のコマンドが指定できます。

デバッガ種別がシミュレータデバッガでのみ有効となります。エミュレータデバッガ、モニタデバッガでは無視されます。

#### ● コマンド修飾子

- STUB 機能指定

/BREAK (省略時のデフォルト)

ブレークポイントにヒットした場合、コマンドリストを処理したあと命令実行を停止します。

/NOBREAK

ブレークポイントにヒットした場合、コマンドリストを処理したあと命令実行を再開します。

- 属性指定

/READ

データリードアクセス時にブレークさせます。

/WRITE

データライトアクセス時にブレークさせます。

## ■ 機能

指定したアドレスをデータアクセスする場合にブレークします。

コマンド修飾子で、ブレークするアクセス種別を設定します。

コマンド修飾子を省略すると /READ/WRITE ( リード / ライトのどちらのアクセスでもブレークする設定 ) になります。

コマンド修飾子のサイズ指定をした場合は以下の条件でブレークします。

- 指定アドレスに指定サイズでアクセスされたとき

パスカウント値は、実行のたびに再設定されます。

最大 65535 ポイント設定可能です。

関数内の自動変数を指定した場合には、その変数が確保されている現在のアドレスがデータアクセスアドレスとして設定されます ( 使用上注意が必要です )。

C/C++ 言語の変数のアクセスでブレークさせる場合は、変数の前に & を指定し、変数のアドレスとして指定する必要があります。

## ■ 例

```
>SET DATABREAK &checkflg, 3, {SHOW TRACE:SHOW TIMER}
```

---

### < 注意事項 >

- コマンドリストには下記の実行系のコマンドは指定できません。
    - GO
    - STEP
    - CALL
    - SYSTEMCALL
  - STEP/INTO および STEP/OVER では STUB 機能 ( コマンド実行および命令実行再開 ) は行いません。ブレークポイントにヒットしたあと実行を停止します。
-

## 3.9 SET DATABREAK (type 2)

指定したアドレスをデータアクセスする場合にブレークします。

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	*
Monitor		×

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

SET DATABREAK データアクセスアドレス  
[ , d= ブレークデータ [&= ブレークデータマスク ] ]

#### ● パラメータ

データアクセスアドレス (アドレス式)

データアクセスブレークポイントを設定するアドレスを指定します。

ブレークデータ (デフォルト 16 進数)

ブレーク条件となるデータ値を指定します。

ブレークデータマスク (デフォルト 16 進数)

ブレーク条件となるデータ値の有効ビットを指定します。

#### ● コマンド修飾子

- アクセス属性の指定

/READ

データリードアクセス時にブレークさせます。

/WRITE

データライトアクセス時にブレークさせます。

- アクセスサイズの指定

/BYTE

ブレーク条件となるデータ幅を 8 ビットに指定します。

/HALFWORD

ブレーク条件となるデータ幅を 16 ビットに指定します。

/WORD

ブレーク条件となるデータ幅を 32 ビットに指定します。



- 機能分類の指定

/BREAKCONDITION (MB2198 DSU4 のみ)

SET BREAKCONDHITION で設定した条件を結び付けて、イベント発生条件にします。

## ■ 機能

指定したアドレスをデータアクセスする場合にブレークします。

コマンド修飾子で、ブレークするアクセス種別を設定します。

コマンド修飾子を省略すると /READ/WRITE になります。

最大2ポイント設定可能です。

関数内の自動変数を指定した場合には、その変数が確保されている現在のアドレスがデータアクセスアドレスとして設定されます(使用上注意が必要です)。

C/C++ 言語の変数のアクセスでブレークさせる場合は、変数の前に & を指定し、変数のアドレスとして指定する必要があります。

## ■ 例

```
>SET DATABREAK &checkflg, 3
```

---

### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.10 SHOW DATABREAK

---

SET DATABREAK コマンドで設定したデータアクセスポイントを表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) × (MB2198)
Monitor	×

### ■ 形式

SHOW DATABREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

すべてのデータブレークポイントを表示します。

### ■ 機能

SET DATABREAK コマンドで設定したデータアクセスポイントを表示します。

### ■ 例

[Simulator]

```
>SHOW DATABREAK
```

no.	en/dis	address	read/write	pass-count	symbol
1	enable	00001000	write only	5 ( 2)	

Control: NO BREAK

Command: show status

[MB2198]

```
>SHOW DATABREAK
```

no.	en/dis	address	read/write	pass-count	symbol
1	enable	00002000	read only	1 ( 0)	¥trac
4	disable	00002052	write only	65535 (2345)	

### < 注意事項 >

エミュレータデバッガではFR60LiteまたはFR80S使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.11 CANCEL DATABREAK

指定された番号のデータアクセスブレークポイントを解除します。

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	*
Monitor		×

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

CANCEL DATABREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW DATABREAK コマンドで参照します。

#### ● コマンド修飾子

/ALL

すべてのデータアクセスブレークポイントを解除します。

### ■ 機能

指定された番号のデータアクセスブレークポイントを解除します。

### ■ 例

```
>CANCEL DATABREAK 1
>CANCEL DATABREAK 3,4
```

#### < 注意事項 >

エミュレータデバッガではFR60LiteまたはFR80S使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

## 3.12 ENABLE DATABREAK

---

指定された番号のデータアクセスブレークポイントを有効にします。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) × (MB2198) *
Monitor	×

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

### ■ 形式

ENABLE DATABREAK [ ブレークポイント番号 [, ... ] ]

#### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW DATABREAK コマンドで参照します。

#### ● コマンド修飾子

/ALL

すべてのデータブレークポイントを有効にします。

### ■ 機能

指定された番号のデータアクセスブレークポイントを有効にします。

### ■ 例

```
>ENABLE DATABREAK 2
>ENABLE DATABREAK 3, 4
```

---

#### < 注意事項 >

エミュレータデバッガではFR60LiteまたはFR80S使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

### 3.13 DISABLE DATABREAK

指定された番号のデータアクセスブレークポイントを無効にします。

#### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	*
Monitor		×

※:「実行中のブレークポイント設定」が無効の場合、ユーザプログラム実行中に使用できません。

#### ■ 形式

DISABLE DATABREAK [ ブレークポイント番号 [, ... ] ]

##### ● パラメータ

ブレークポイント番号 (デフォルト 10 進数)

ブレークポイント番号を指定します。

ブレークポイント番号は、SHOW DATABREAK コマンドで参照します。

##### ● コマンド修飾子

/ALL

すべてのデータブレークポイントを無効にします。

#### ■ 機能

指定された番号のデータアクセスブレークポイントを無効にします。

#### ■ 例

```
>DISABLE DATABREAK 2
>DISABLE DATABREAK 3, 4
```

#### < 注意事項 >

エミュレータデバッガではFR60LiteまたはFR80S使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

## 3.14 SET EVENT

シーケンサまたはパフォーマンスのトリガ要因となるイベントの設定を行います。

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

[ イベントモード : ノーマル ]

SET EVENT アドレス [&= マスク] [, [!] d= データ [&= マスク]] [, p= パスカウント]

[ イベントモード : パフォーマンス ]

SET EVENT アドレス [&= マスク] [, [!] d= データ [&= マスク]] [, b= イベント番号]

### ● パラメータ

アドレス [&= マスク] (アドレス式, データ式)

イベント発生条件となるメモリ位置を指定します。マスクを指定した場合は、マスクのビットが1のところだけが有効になり、ほかは無視されます。

マスクデータを省略した場合は、すべてのビットが有効になります。

C 言語の自動変数は設定できません。

d= データ [&= マスク] (データ式, データ式)

イベント発生条件となるデータを指定します。マスクを指定した場合は、マスクのビットが1のところだけが有効になり、ほかは無視されます。

マスクデータを省略した場合は、すべてのビットが有効になります。

! を指定した場合は、指定されたデータの not 指定になります。

/ANYTHING 修飾子を指定した場合、32 ビットデータを比較対象にします。

(8/16 ビットデータを比較する場合は、マスク指定する必要があります)

p= パスカウント (デフォルト 10 進数)

イベントの発生回数 (1 ~ 16777215) を指定します。

省略した場合は 1 に設定されます。

イベントモードがノーマルの時に指定可能です。

b= イベント番号 (パフォーマンス)

イベント番号 (1 ~ 4) を指定します。

イベントモードがパフォーマンスの場合、このパラメータは省略できません。

## ● コマンド修飾子

### • アクセス属性

/CODE

指定されたアドレスを実行したときをイベント発生条件にします。

FR80S では指定できません。

/READ

指定されたアドレスをリードしたときをイベント発生条件にします。

FR80S では指定できません。

/WRITE

指定されたアドレスをライトしたときをイベント発生条件にします。

FR60Lite では /CODE/WRITE は省略できません。省略時は /CODE を指定したことになります。

FR80S では省略時は /WRITE を指定したことになります。

### • データ長指定

/WORD

イベント条件データをワードとして扱うことを指定します。

/HALFWORD

イベント条件データをハーフワードとして扱うことを指定します。

/BYTE

イベント条件データをバイトとして扱うことを指定します。

/ANYTHING (省略時のデフォルト)

データ長にかかわらず、指定アドレスがアクセスされたときをイベント発生条件にします。

## ■ 機能

[ イベントモード : ノーマル ]

シーケンサのトリガ要因となるイベントの設定を行います。データの指定を省略した場合は、データはすべて無視されます。

イベント設定は最大4個まで設定可能です。ただし、ハードウェアが共用のため、SET BREAK/DATAWATCH と SET TRACETRIGGER との合計ポイントとなります。

シーケンサについて、SET SEQUENCE コマンドを使ってイベント番号を指定し設定を行ってください。

イベント番号は SHOW EVENT で参照するか、組込み関数 %EVENTNUM を用いてアドレスからも参照可能です。

[ イベントモード : パフォーマンス ]

パフォーマンス測定の測定条件となる、開始/終了アドレスをイベント番号で指定します。それぞれのイベント番号に次の意味付けがされています。

イベント 1	区間 1 の開始イベント
イベント 2	区間 1 の終了イベント
イベント 3	区間 2 の開始イベント
イベント 4	区間 2 の終了イベント

#### ■ 例

```
>SET EVENT /READ func1, b=1  
>SET EVENT /WRITE &data[2],!d=h'10, b=2
```

---

#### < 注意事項 >

1. FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」または「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 領域に指定した場合のみ有効となります。
3. イベントモードにより設定内容が異なります。
4. /ANYTHING 修飾子を指定した場合,32 ビットデータを比較対象にします。  
8/16 ビットデータを比較する場合は,マスクデータを指定する必要があります。

例) 8 ビットデータ (0x12) の場合

```
SET EVENT/ANYTHING 0x10000, D=0x12000000&=0xff000000
```

16 ビットデータ (0x4567) の場合

```
SET EVENT/ANYTHING 0x10000, D=0x45670000&=0xffff0000
```

---



## 3.15 SHOW EVENT

---

SET EVENT の設定内容を表示します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW EVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (1 ~ 4) を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

すべてのイベントを表示します。

### ■ 機能

SET EVENT の設定内容を表示します。

### ■ 例

```
>SHOW EVENT
```

---

#### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」または「パフォーマンス測定」を参照してください。

---

## 3.16 CANCEL EVENT

---

指定された番号のイベントを取り消します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

CANCEL EVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (1 ~ 4) を指定します。

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

すべてのイベントを取り消します。

### ■ 機能

指定された番号のイベントを取り消します。

### ■ 例

```
>CANCEL EVENT
```

---

#### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」または「パフォーマンス測定」を参照してください。

---

## 3.17 ENABLE EVENT

---

一时无効にしたイベントを有効にします。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

ENABLE EVENT [ イベント番号 [, ...] ]

#### ● パラメータ

イベント番号

イベント番号 (1 ~ 4) を指定します

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

すべてのイベントを有効にします。

### ■ 機能

一时无効にしたイベントを有効にします。

### ■ 例

```
>ENABLE EVENT
```

---

#### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」または「パフォーマンス測定」を参照してください。

---

## 3.18 DISABLE EVENT

---

イベントを一時的に無効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

DISABLE EVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (1 ~ 4) を指定します

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

すべてのイベントを一時的に無効にします。

### ■ 機能

イベントを一時的に無効にします。

### ■ 例

```
>DISABLE EVENT
```

---

#### < 注意事項 >

FR60Lite または FR80S 使用時以外ではこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」または「パフォーマンス測定」を参照してください。

---

## 3.19 SET CODEEVENT

コードイベントの設定を行います。

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET CODEEVENT イベント番号, アドレス [ &= アドレスマスク ] [, パスカウント]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

アドレス [ &= アドレスマスク ] (アドレス式, データ式)

イベント発生条件となるアドレスを指定します。

アドレスマスクを指定した場合は, アドレスマスクのビットが 1 であるところのみが有効となり, ほかのビットは無視されます。

アドレスマスクデータを省略した場合は, すべてのビットが有効となります。

C/C++ 言語の自動変数は設定できません。

パスカウント (デフォルト 10 進数)

イベントの発生回数 (1 ~ 255) を指定します。

省略した場合は, 1 に設定されます。

#### ● コマンド修飾子

/BREAKCONDITION (MB2198 DSU4 のみ)

SET BREAKCONDITION で設定した条件を結びつけて, イベント発生条件にします。

### ■ 機能

コードイベントの設定を行います。2 ポイントのみ設定可能です。コードブレーク用イベントの設定を行います。

アドレスとアドレスマスク, パスカウント値が設定できます。

DSU3 チップでは, コードイベントをプログラムの停止要因のほかに, トレースの測定開始要因としても利用できます。この切替えは, SET TRACE コマンドで行うことができます。

### ■ 例

```
>SET CODEEVENT 1, function
>SET CODEEVENT 2, loop, 3
```

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.20 SHOW CODEEVENT

指定された番号のイベント設定内容を表示します。

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW CODEEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

すべてのイベントを表示します。

### ■ 機能

指定された番号のイベント設定内容を次の形式で表示します。

no.	en/dis	addr	mask	pass	symbol
no.	:	イベント番号です。			
en/dis	:	有効 (enable) , 無効 (disable) を表示します。			
addr	:	アドレスを 16 進数で表示します。			
mask	:	アドレスのマスクデータです。			
pass	:	パスカウントです。10 進数で表示します。			
symbol	:	アドレスに対応するシンボルまたは行番号を表示します。			

### ■ 例

```
>SHOW CODEEVENT/ALL
```

no.	en/dis	addr	mask	pass	symbol
1	enable	0000FF00	FFFFFFFF	1 ( 1 )	¥function
2	enable	0000EFF0	FFFFFFFF	5 ( 0 )	

#### < 注意事項 >

- 1) 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。
  - 2) コードイベントのヒットカウントは、プリフェッチしただけでカウントされます。
  - 3) FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。
-



## 3.21 CANCEL CODEEVENT

---

指定された番号のイベントを解除します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

CANCEL CODEEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを解除します。

### ■ 機能

指定された番号のイベントを解除します。

### ■ 例

```
>CANCEL CODEEVENT 1  
>CANCEL CODEEVENT/ALL
```

---

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.22 ENABLE CODEEVENT

---

指定された番号のイベントを有効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE CODEEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを有効にします。

### ■ 機能

指定された番号のイベントを有効にします。

### ■ 例

```
>ENABLE CODEEVENT 2  
>ENABLE CODEEVENT/ALL
```

---

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.23 DISABLE CODEEVENT

指定された番号のイベントを無効にします。

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

DISABLE CODEEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを無効にします。

### ■ 機能

指定された番号のイベントを無効にします。

### ■ 例

```
>DISABLE CODEEVENT 2
>DISABLE CODEEVENT/ALL
```

### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

## 3.24 SET DATAEVENT

---

データイベントの設定を行います。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET DATAEVENT イベント番号, アドレス [&= アドレスマスク]  
[, [!] d= データ [&= データマスク ]]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

アドレス [&= アドレスマスク] (アドレス式, データ式)

イベント発生条件となるアドレスを指定します。

アドレスマスクを指定した場合は, アドレスマスクのビットが 1 であるところのみが有効となり, ほかのビットは無視されます。

アドレスマスクデータを省略した場合は, すべてのビットが有効となります。

C 言語の自動変数は設定できません。

[!] D= データ [&= データマスク] (データ式, データ式)

イベント発生条件となるデータ (32 ビット長) を指定します。

データマスクを指定した場合は, データマスクのビットが 1 であるところのみが有効となり, ほかのビットは無視されます。

データマスクを省略した場合は, すべてのビットが有効となります。

! を指定した場合は, 指定されたデータとデータマスクの NOT 指定 (データマスクで指定したビットが, データで指定したビットパターンと一致しないとき, イベント発生条件となる) となります。

なお, ご使用のチップによって, データとデータマスクの指定ができない場合があります。

## ● コマンド修飾子

/READ

指定アドレスにリードアクセスを行ったときをイベント発生条件とします。

/WRITE

指定アドレスにライトアクセスを行ったときをイベント発生条件とします。

省略時は、/READ/WRITE を指定したことになります。

/BYTE (省略時のデフォルト)

指定アドレスに1バイトアクセスを行ったときをイベント発生条件とします。

/HALFWORD

指定アドレスに2バイトアクセスを行ったときをイベント発生条件とします。

/WORD

指定アドレスに4バイトアクセスを行ったときをイベント発生条件とします。

/BREAKCONDITION (MB2198 DSU4 のみ)

SET BREAKCONDITION で設定した条件を結びつけて、イベント発生条件にします。

## ■ 機能

データイベントの設定を行います。2ポイントのみ設定可能です。

コマンド修飾子を省略すると /READ/WRITE (リード/ライトのどちらのアクセスでもイベント発生条件とする設定)、/BYTE (バイトアクセスを行ったときイベント発生条件とする設定) になります。

イベント発生条件となるデータ値を指定した場合は、アドレスマスクが無効になります。

DSU3 チップでは、データイベントをプログラムの停止要因のほかに、トレースの測定開始要因としても利用できます。この切替えは、SET TRACE コマンドで行うことができます。

## ■ 例

```
>SET DATAEVENT 1, flag
>SET DATAEVENT/READ 2, data
```

## &lt; 注意事項 &gt;

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

## 3.25 SHOW DATAEVENT

指定された番号のイベント設定内容を表示します。

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW DATAEVENT [ イベント番号 [, ...] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

すべてのイベントを表示します。

### ■ 機能

指定された番号のイベント設定内容を次の形式で表示します。

no.	en/dis	addr	mask	status	size	symbol
no.	:	イベント番号です。				
en/dis	:	有効 (enable) , 無効 (disable) を表示します。				
status	:	アクセス属性 (R : read, W : write) です。				
addr	:	アドレスを 16 進数で表示します。				
addr_msk	:	アドレスのマスクデータです。				
data_msk	:	データのマスクデータです。				
size	:	データサイズです。				
symbol	:	アドレスに対応するシンボルを表示します。				

### ■ 例

```
>SHOW DATAEVENT/ALL
no. en/dis status addr      addr_msk  data      data_msk size symbol
1   enable R/W   00500000 -----  0000000A FFFFFFFF word ¥R_Tpri_Lev
2   enable R/W   00508038 -----  0051017C FFFFFFFF word ¥task0
```

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

## 3.26 CANCEL DATAEVENT

---

指定された番号のイベントを解除します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

CANCEL DATAEVENT [ イベント番号 [ , ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを解除します。

### ■ 機能

指定された番号のイベントを解除します。

### ■ 例

```
>CANCEL DATAEVENT 1
>CANCEL DATAEVENT/ALL
```

---

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.27 ENABLE DATAEVENT

---

指定された番号のイベントを有効にします

---

### ■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE DATAEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを有効にします。

### ■ 機能

指定された番号のイベントを有効にします。

### ■ 例

```
>ENABLE DATAEVENT 2
>ENABLE DATAEVENT/ALL
```

---

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---



## 3.28 DISABLE DATAEVENT

---

指定された番号のイベントを無効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

DISABLE DATAEVENT [ イベント番号 [, ... ] ]

#### ● パラメータ

イベント番号

イベント番号 (D'1 ~ D'2) を指定します。

#### ● コマンド修飾子

/ALL

すべてのイベントを無効にします。

### ■ 機能

指定された番号のイベントを無効にします。

### ■ 例

```
>DISABLE DATAEVENT 2  
>DISABLE DATAEVENT/ALL
```

---

#### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「ブレーク」を参照してください。

---

## 3.29 SET SEQUENCE (type 1)

---

コードイベントおよびデータイベントのシーケンシャル動作モードを設定します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET SEQUENCE

#### ● コマンド修飾子

/ON

イベント1-> イベント2の順に条件が成立した場合がディレイカウント開始のトリガとなります。

/OFF

イベント1またはイベント2の条件が成立した場合がディレイカウントの開始のトリガとなります。

/CODE

コードイベントに対して処理を行います。

/DATA

データイベントに対して処理を行います。

/ALL

コードイベントおよびデータイベントに対して処理を行います。

## ■ 機能

コードイベントおよびデータイベントのシーケンシャル動作モードを設定します。

/ONを指定した場合にはシーケンシャルモードとなり、イベント1 -> イベント2の順に条件が成立したときに、プログラム停止となります。

/OFFを指定した場合にはORモードとなり、イベント1またはイベント2の条件が成立したときに、プログラム停止となります。

シーケンシャル動作モードは、コードイベント・データイベントについて個別に設定できます。

DSU3 チップでは、次のように動作します。

SET TRACE/TRIGGER コマンドで、コードイベントおよびデータイベントをトレースの測定開始要因にも使用できます。このとき、コードイベントとデータイベントはプログラムの停止要因とはならないため、本コマンドは使用できません。

## ■ 例

```
>SET SEQUENCE/ON
```

---

### < 注意事項 >

FR60Lite 使用時はこのコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」を参照してください。

---

## 3.30 SET SEQUENCE (type 2)

---

シーケンサの設定を行います。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET SEQUENCE イベント番号 [, イベント番号 [, イベント番号]] [, R= イベント番号]

#### ● パラメータ

イベント番号 (デフォルト 10 進数)

トリガとして設定したいイベントの番号 (D'1 ~ D'4) を指定します。

R= イベント番号 (デフォルト 10 進数)

シーケンサの再開条件となるイベントの番号 (D'1 ~ D'4) を指定します。

### ■ 機能

シーケンサの設定を行います。3 レベルの RESTART 付きシーケンス設定を行います。  
SET EVENT で予めイベントポイントの設定を行ってください。

イベント番号指定は組込み関数 %EVENTNUM を用いてアドレスから参照可能です。

### ■ 例

```
>SET SEQUENCE 2, 3, 1, R=4
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

### 3.31 SHOW SEQUENCE

SET SEQUENCE コマンドで設定するイベントのシーケンシャル動作モードを表示します。

#### ■ 対応デバugga

Simulator	x
Emulator	(MB2197) (MB2198)
Monitor	x

#### ■ 形式

SHOW SEQUENCE

#### ■ 機能

SET SEQUENCEコマンドで設定するイベントのシーケンシャル動作モードを表示します。

DSU3 チップでは、次のように動作します。

SET TRACE/TRIGGER コマンドで、コードイベントおよびデータイベントをトレースの測定開始要因にも使用できます。このとき、コードイベントとデータイベントはプログラムの停止要因とはならないため、本コマンドは使用できません。

#### ■ 例

```
>SHOW SEQUENCE
code sequence mode = ON
data sequence mode = OFF
【FR60Lite, FR80S】
>SHOW SEQUENCE
level1      -> level2    -> level3    -> end restart
00FF0010    00FF0020    -----    -----
```

## 3.32 CANCEL SEQUENCE

---

シーケンサの設定を解除します。

---

### ■ 対応デバッグ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

CANCEL SEQUENCE

### ■ 機能

SET SEQUENCE コマンドで指定したシーケンサの設定を一括して解除します。ただし、個々のイベント設定までは削除されません。

### ■ 例

```
>CANCEL SEQUENCE
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用時以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

### 3.33 ENABLE SEQUENCE

---

シーケンサの動作を有効にします。

---

#### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

#### ■ 形式

ENABLE SEQUENCE

#### ■ 機能

シーケンサの動作を有効にします。

#### ■ 例

>ENABLE SEQUENCE

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

## 3.34 DISABLE SEQUENCE

---

シーケンサの動作を無効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

DISABLE SEQUENCE

### ■ 機能

シーケンサの動作を無効にします。

### ■ 例

>DISABLE SEQUENCE

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「シーケンサによる制御」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-



## 3.35 SET TRIGGER

---

TRIG 端子の信号入力をブレーク要因にするか否かを指定します。

---

### ■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET TRIGGER

#### ● コマンド修飾子

/BREAK

TRIG 端子の信号入力をブレーク要因に使用します。

/NOBREAK ( 省略時のデフォルト )

TRIG 端子の信号入力をブレーク要因に使用しません。

### ■ 機能

TRIG 端子の信号入力をブレーク要因にするか否かを指定します。

ブレーク要因に指定すると、TRIG 端子に信号入力があった場合にプログラム実行を中断します。

### ■ 例

```
>SET TRIGGER/BREAK
```

## 3.36 SHOW TRIGGER

---

TRIG 端子から信号入力をブレーク要因に使用するか否かの設定状態を表示します。

---

### ■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW TRIGGER

### ■ 機能

TRIG 端子から信号入力をブレーク要因に使用するか否かの設定状態を表示します。

### ■ 例

```
>SHOW TRIGGER
trigger = No Break
>
>SET TRIGGER/BREAK
>SHOW TRIGGER
trigger = Break
```

## 3.37 ENABLE ALIGNMENTBREAK

---

アライメントエラーブレークを有効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE ALIGNMENTBREAK

#### ● コマンド修飾子

/CODE

コードフェッチによるアライメントエラーブレークを有効にします。

/DATA

データアクセスによるアライメントエラーブレークを有効にします。

コマンド修飾子を省略すると /CODE/DATA となります。

### ■ 機能

アライメントエラーブレークを有効にします。

コードアクセス, データアクセスそれぞれに設定することができます。

### ■ 例

```
>ENABLE ALIGNMENTBREAK/DATA
```

## 3.38 DISABLE ALIGNMENTBREAK

---

アライメントエラーブレークを無効にします。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

DISABLE ALIGNMENTBREAK

#### ● コマンド修飾子

/CODE

コードフェッチによるアライメントエラーブレークを無効にします。

/DATA

データアクセスによるアライメントエラーブレークを無効にします。

コマンド修飾子を省略すると /CODE/DATA となります。

### ■ 機能

アライメントエラーブレークを無効にします。

コードアクセス、データアクセスそれぞれに設定することができます。

### ■ 例

```
>DISABLE ALIGNMENTBREAK/DATA
```

## 3.39 SHOW ALIGNMENTBREAK

---

アライメントエラーブレイク状態を表示します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW ALIGNMENTBREAK

### ■ 機能

アライメントエラーブレイク状態を表示します。

### ■ 例

```
>SHOW ALIGNMENTBREAK  
code = enable  
data = disable
```

## 3.40 SET BREAKCONDITION (type 1)

データ監視ブレーク (ソフトウェア) のブレーク条件を設定します。

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	
Monitor		×

### ■ 形式

SET BREAKCONDITION [/ADDRESS] アドレス, データ

#### ● パラメータ

アドレス (アドレス式)

監視するアドレスを指定します。

データ (データ式)

上述のアドレスの内容に対して, 比較基準となるデータ値を指定します。

#### ● 機能分類コマンド修飾子

/ ADDRESS

監視するデータの対象として, アドレスを指定します。

#### ● コマンド修飾子

/ EQ (起動時のデフォルト)

指定したアドレスの内容が, 指定したデータ値と一致した場合にブレークします。

/ NE

指定したアドレスの内容が, 指定したデータ値と一致しない場合にブレークします。

/ BYTE (起動時のデフォルト)

監視するデータが 8 ビット長であることを指定します。

/ HALFWORD

監視するデータが 16 ビット長であることを指定します。

/ WORD

監視するデータが 32 ビット長であることを指定します。

## ■ 機能

データ監視ブレーク（ソフトウェア）を設定します。

本コマンドは、SET BREAK/BREAKCONDITION、

SET CODEEVENT/BREAKCONDITION または SET DATAEVENT/BREAKCONDITION  
と組み合わせて使用します。

SET BREAK/BREAKCONDITION、SET CODEEVENT/BREAKCONDITION または SET  
DATAEVENT/BREAKCONDITION で指定したアドレスの命令を実行する直前に、エ  
ミュレータ内部でプログラム実行を一度停止し、本コマンドで指定したアドレスの内  
容が指定したデータ値と一致 / 不一致のときにブレークします。条件を満たさない場  
合、プログラム実行を再開します。

最大設定数は1ポイントです。

## ■ 例

```
>SET BREAKCONDITION/ADDRESS/WORD/NE &tskl_stat, 0
```

---

### < 注意事項 >

モニタリング機能を使用する場合、このコマンドは使用しないでください。データ監視ブ  
レーク（ソフトウェア）を設定後、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。

---

## 3.41 SET BREAKCONDITION (type 2)

データ監視ブレーク (ソフトウェア) のブレーク条件を設定します。

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	
Monitor		×

### ■ 形式

SET BREAKCONDITION /REGISTERINDIRECT レジスタ名, オフセット, データ

#### ● パラメータ

レジスタ名

指定したレジスタの内容をアドレスとして評価し, 監視するアドレスを指定します。レジスタ名は R0 ~ R15 が指定できます。

オフセット (デフォルト 10 進数)

指定したオフセット値を上述のレジスタの内容と足し合わせて, アドレスとして評価し, 監視するアドレスを指定します。

データ (データ式)

上述のアドレスの内容に対して, 比較基準となるデータ値を指定します。

#### ● 機能分類コマンド修飾子

/REGISTERINDIRECT

監視するデータの対象として, " レジスタの内容 + オフセット " で示されるアドレスを指定します。

#### ● コマンド修飾子

/EQ (起動時のデフォルト)

指定したアドレスの内容が, 指定したデータ値と一致した場合にブレークします。

/NE

指定したアドレスの内容が, 指定したデータ値と一致しない場合にブレークします。

/BYTE (起動時のデフォルト)

監視するデータが 8 ビット長であることを指定します。

/HALFWORD

監視するデータが 16 ビット長であることを指定します。

/WORD

監視するデータが 32 ビット長であることを指定します。



## ■ 機能

データ監視ブレーク (ソフトウェア) を設定します。

本コマンドは, SET BREAK/BREAKCONDITION,

SET CODEEVENT/BREAKCONDITION または SET DATAEVENT/BREAKCONDITION  
と組み合わせて使用します。

SET BREAK/BREAKCONDITION, SET CODEEVENT/BREAKCONDITION または SET  
DATAEVENT/ BREAKCONDITIONで指定したアドレスの命令を実行する直前に, プロ  
グラム実行を一度停止し, 本コマンドで指定した " レジスタ + オフセット " の内容が指  
定したデータ値と一致 / 不一致のときにブレークします。条件を満たさないときは, プ  
ログラム実行を再開します。

最大設定数は 1 ポイントです。

## ■ 例

```
>SET BREAKCONDITION/REGISTERINDIRECT/HALFWORD/NE R11, 58, 1354
```

---

### < 注意事項 >

モニタリング機能を使用する場合, このコマンドは使用しないでください。データ監視ブ  
レーク (ソフトウェア) を設定後, 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。

---

## 3.42 SHOW BREAKCONDITION

---

SET BREAKCONDITION コマンドで指定した設定情報を表示します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW BREAKCONDITION

### ■ 機能

SET BREAKCONDITION コマンドで指定した設定情報を表示します。

### ■ 例

```
>show breakcondition
addressing  data      size  compare-condition
R7+8244    000000FF byte  ne(not-equal)
```

## 第4章

---

# プログラム実行の 解析コマンド

この章では、プログラム実行の解析コマンドについて説明します。

- 4.1 SET PERFORMANCE
- 4.2 SHOW PERFORMANCE
- 4.3 CLEAR PERFORMANCE
- 4.4 SET COVERAGE
- 4.5 SHOW COVERAGE
- 4.6 CANCEL COVERAGE
- 4.7 CLEAR COVERAGE
- 4.8 SHOW CALLS
- 4.9 SHOW TIMER
- 4.10 CLEAR TIMER
- 4.11 SET TRACE (type 1)
- 4.12 SET TRACE (type 2)
- 4.13 SHOW TRACE (type 1)
- 4.14 SHOW TRACE (type 2)
- 4.15 CLEAR TRACE
- 4.16 ENABLE TRACE
- 4.17 DISABLE TRACE
- 4.18 SEARCH TRACE
- 4.19 SET DATATRACEAREA
- 4.20 SHOW DATATRACEAREA
- 4.21 CANCEL DATATRACEAREA
- 4.22 SHOW DETAILTRACE

## 第 4 章 プログラム実行の 解析コマンド

4.23 SET TRACETRIGGER

4.24 SHOW TRACETRIGGER

4.25 CANCEL TRACETRIGGER

4.26 ENABLE TRACETRIGGER

4.27 DISABLE TRACETRIGGER

4.28 SET RAMCHECK

4.29 SHOW RAMCHECK

4.30 CANCEL RAMCHECK

4.31 ENABLE RAMCHECK

4.32 DISABLE RAMCHECK

## 4.1 SET PERFORMANCE

パフォーマンス測定用バッファフル時の動作を設定します。

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET PERFORMANCE

#### ● コマンド修飾子

/BREAK (省略時のデフォルト)

パフォーマンス測定用のバッファがいっぱいになったときブレイクします。

/NOBREAK

パフォーマンス測定用のバッファがいっぱいになったときブレイクしません。

### ■ 機能

パフォーマンス測定用バッファフル時の動作を設定します。/BREAK を指定することにより、バッファがいっぱいになったときブレイクさせることができます。パフォーマンス測定用バッファは、イベントが 65535 回発生した場合にいっぱいになります。

このコマンドの設定は、イベントモードがパフォーマンスの場合のみ有効です。パフォーマンスの測定ポイントの設定は、SET EVENT コマンドで行います。パフォーマンス測定では次のものを測定します。

#### • 時間測定

2つのイベント間の時間を測定します。2区間の測定ができます。開始、終了イベントは、次の番号の組み合わせになります。

区間1：開始イベント1 - 終了イベント2

区間2：開始イベント3 - 終了イベント4

#### • 発生回数カウント

イベントの発生回数をカウントします。

### ■ 例

```
>SET PERFORMANCE /BREAK
```

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用時以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「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。

## 4.2 SHOW PERFORMANCE

パフォーマンスの設定状態を表示します。

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

< 形式 1>

SHOW PERFORMANCE /STATUS

< 形式 2>

SHOW PERFORMANCE /COUNT [ イベント番号 ]

< 形式 3>

SHOW PERFORMANCE /TIME エリア番号 [, 下限 , 上限 , 表示間隔 ]

#### ● コマンド修飾子

/STATUS

バッファフル時の動作設定を表示します。

/COUNT

イベント発生回数の測定結果を表示します。

/TIME ( 省略時のデフォルト )

指定エリア内の時間測定結果を表示します。

#### ● パラメータ

イベント番号

設定内容を表示させたいイベント番号を表示します ( 1 ~ 4 )。

時間測定表示の場合は , 測定区間の開始 , 終了イベントのどちらの番号を指定しても動作は同じです。

エリア番号

時間測定結果を表示したいエリア番号を指定します ( 1 ~ 2 )。

下限 ( デフォルト 10 進数 )

測定時間をグラフ表示する際の下限值を設定します。単位は 1ns になります。

上限 ( デフォルト 10 進数 )

測定時間をグラフ表示する際の上限值を指定します。単位は 1ns になります。

表示間隔 ( デフォルト 10 進数 )

測定時間をグラフ表示する際の間隔を測定します。単位は 1ns になります。

## ■ 機能

パフォーマンスの設定状態を表示します。

- /STATUS 指定時

パフォーマンス測定用バッファがいっぱいになった場合の動作の設定状態を表示します。

- /COUNT 指定時

イベントの発生回数を表示します。

- /TIME 指定時

測定結果を集計して表示します。表示の上限時間, 下限時間, 間隔を指定できます。

指定する時間の単位は SET TIMERSCALE コマンドの設定に従います。

## ■ 例

```
>SHOW PERFORMANCE /COUNT
```

---

### < 注意事項 >

- 1) FR60Lite または FR80S 使用時以外では, このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「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。
-

## 4.3 CLEAR PERFORMANCE

---

パフォーマンス測定値をクリアします。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

CLEAR PERFOMANCE

### ■ 機能

パフォーマンス測定値をクリアします。

### ■ 例

```
>CLEAR PERFORMANCE
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「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。
-



## 4.4 SET COVERAGE

カバレッジ測定を行う領域を指定します。最大 32 領域まで設定可能です。

### ■ 対応デバッガ

Simulator		*
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

※: 高速版シミュレータデバッガの場合のみ使用可能です。

### ■ 形式

SET COVERAGE [ 測定範囲 ]

#### ● パラメータ

測定範囲（アドレス式）

カバレッジ領域を指定します。コマンド修飾子に /AUTOMATIC を指定した場合は指定できません。

#### ● コマンド修飾子

/AUTOMATIC（省略時のデフォルト）

現在ロードされているモジュールのコード領域が自動的に設定されます。ただし、C コンパイラのライブラリ領域は設定されません。

### ■ 機能

カバレッジ測定を行う領域を指定します。最大 32 領域まで設定可能です。

### ■ 例

```
>SET COVERAGE FE000000..FFFFFFFF
```

## 4.5 SHOW COVERAGE

指定された測定範囲のカバレッジ測定結果を表示します。

### ■ 対応デバッガ

Simulator		*
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

\* 高速版シミュレータデバッガの場合のみ使用可能です。

### ■ 形式

<形式 1>

SHOW COVERAGE [ /STATUS ]

<形式 2>

SHOW COVERAGE [ /GENERAL ] [ 測定範囲 ]

<形式 3>

SHOW COVERAGE { /TOTAL | /DETAIL } [ 測定範囲 ]

<形式 4>

SHOW COVERAGE /SOURCE

[ [ ファイル名 ] 行番号 [ .. 行番号 ] | { アドレス | アドレス範囲 } ]

<形式 5>

SHOW COVERAGE /INSTRUCTION [ { アドレス | アドレス範囲 } ]

<形式 6>

SHOW COVERAGE /MODULE [ { ソースファイル名 | カバレッジ範囲 } [ , カラム数 ] ]

### ● パラメータ

測定範囲（アドレス式）

カバレッジ領域を指定します。

省略した場合は最初の領域から順に表示します。

ファイル名

カバレッジ測定結果を表示したいソースファイル名を指定します。

省略した場合は前回指定されたファイル名となります。

行番号

カバレッジ測定結果を表示したいソースの行番号を指定します。

行番号の先頭には、必ず \$ を付けてください。

行番号を " .. " で区切った場合は指定範囲のソースを表示します。

終了行番号を指定しなかった場合は、19 行分表示します。

**アドレス（アドレス式）**

コード属性のメモリ位置を指定します。

メモリ位置に対応するカバレッジ測定結果を表示させたい場合に指定します。

関数を指定した場合は、関数のアドレス範囲のカバレッジ測定結果を表示します。

関数以外を指定した場合は、19 行分表示します。

**アドレス範囲（アドレス式）**

コード属性のメモリ領域の範囲を指定します。

メモリ位置に対応するカバレッジ測定結果を表示させたい場合に指定します。

**ソースファイル名**

カバレッジ率を表示したいソースファイル名を指定します。

省略した場合はロードモジュール全体のカバレッジ率を表示します。

**カバレッジ率範囲**

表示対象とするカバレッジ率の範囲を指定します。

省略された場合はすべて表示します。

**カラム数**

カバレッジ率のカラム位置（行頭からの文字数）を指定します。

省略された場合のカラム数は 40 です。

**● コマンド修飾子****/STATUS**

カバレッジ機能の有効 / 無効、カバレッジ測定領域を表示します。

**/TOTAL**

指定された測定範囲全体のカバレッジ率を表示します。

**/GENERAL（省略時のデフォルト）**

16 アドレス単位の測定結果を表示します。

**/DETAIL**

1 アドレス単位の測定結果を表示します。

**/SOURCE**

ソース行単位の測定結果を表示します。

**/INSTRUCTION**

機械命令単位の測定結果を表示します。

SET SOURCE コマンドの設定がソース行を付加するモードになっていれば、メモリ位置がソース行に対応している場合は、そのソース行も表示します。

**/MODULE**

ロードモジュールのカバレッジ率を表示します。

## ■ 機能

指定された測定範囲のカバレッジ測定結果を表示します。

/GENERAL を指定した場合、アクセス回数は次のように表示します。

```

.      :      アクセスなし
1 ~ F :      16 アドレス中、アクセスされたアドレスの数
*      :      16 アドレスともアクセスされた場合

```

/DETAIL を指定した場合、アクセス回数は次のように表示します。

```

.      :      アクセスなし
-      :      アクセスあり

```

/SOURCE,/INSTRUCTION を指定した場合、アクセス回数は次のように表示します。

```

.      :      アクセスなし
*      :      アクセスあり
空白   :      コードが生成されないまたは測定範囲外の行

```

/MODULE を指定した場合、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。領域の一部がカバレッジ測定範囲外の場合、カバレッジ率の後ろにアスタリスク "\*" を付けて表示します。

## ■ 例

>SHOW COVERAGE

```

(HEX) 0X0          +1X0          +2X0
      +-----+-----+-----+-----+
address 0123456789ABCDEF0123456789ABCDEF0123456 ... ABCDEF C0(%)
FF000000 **3*F*..... 32.0

```

>SHOW COVERAGE/SOURCE

```

*      70: {
      71:     int i;
      72:     struct table *value[16];
      73:
*      74:     for (i=0; i<16; i++)
*      75:         value[i] = &target[i];
      76:
*      77:     sort_val(value, 16L);
.      78: }

```

>SHOW COVERAGE/MODULE

```

sample.abs ..... (84.03%)
+- startup.asm ..... (90.43%)
+- sample.c ..... (95.17%)
+- samp.c ..... (100.00%)

```

## 4.6 CANCEL COVERAGE

---

カバレッジ測定領域を解除し、カバレッジ測定機能を無効にします。

---

### ■ 対応デバッガ

Simulator		*
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

※: 高速版シミュレータデバッガの場合のみ使用可能です。

### ■ 形式

CANCEL COVERAGE [ 測定範囲 ]

#### ● パラメータ

測定範囲（アドレス式）

削除する測定範囲を指定します。

#### ● コマンド修飾子

/ALL

すべての測定範囲を削除します。

### ■ 機能

カバレッジ測定領域を解除し、カバレッジ測定機能を無効にします。

### ■ 例

```
>CANCEL COVERAGE /ALL
```

## 4.7 CLEAR COVERAGE

---

カバレッジバッファをクリアします。

---

### ■ 対応デバッガ

Simulator		*
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

※: 高速版シミュレータデバッガの場合のみ使用可能です。

### ■ 形式

CLEAR COVERAGE

### ■ 機能

カバレッジバッファをクリアします。

### ■ 例

```
>CLEAR COVERAGE
```

## 4.8 SHOW CALLS

現在の関数までの関数呼出し履歴を表示します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW CALLS [ コールフレーム数 ]

#### ● パラメータ

コールフレーム数 ( デフォルト 10 進数 )

情報が必要なコールフレーム数 (D'1 ~ D'256) を指定します。

### ■ 機能

現在の関数までの関数呼出し履歴を表示します。

パラメータが省略された場合は、最大 256 フレーム表示します。

表示する関数に引数がある場合は、16 進数でその値を表示します。

C/C++言語のデバッグ情報がない場合は、関数名の代わりに関数のアドレスを表示します。

SHOW CALLS コマンドは、C/C++ 言語が関数を呼び出す際のスタック形式に従って、スタックに積まれたデータを解析することにより表示しています。

以下のようなプログラムでは正確に表示できませんのでご注意ください。

- アセンブラで書かれたプログラムには使用できません。
- 最適化したプログラムでは、正常に表示できない場合があります。
- デバッグ情報付きでコンパイルされていない場合、関数名の代わりにアドレスを表示しますが、関数の先頭でブレークしている場合、正常に表示できません。

### ■ 例

```
>SHOW CALLS
cheker (12,8)
main(3,4)
```

## 4.9 SHOW TIMER

実行したプログラムの命令実行サイクル数, ステップ数, 時間を 10 進数で表示します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW TIMER

### ■ 機能

実行したプログラムの命令実行サイクル数, ステップ数, 時間を 10 進数で表示します。  
表示される数値は RESET コマンド実行後からのものと, 直前に GO, STEP, CALL コマンドにより実行を開始した時点からのものです。

シミュレータデバッガでは, サイクル数とステップ数を表示します。

エミュレータデバッガでは, サイクル数を表示します。

モニタデバッガでは, 時間を表示します。

#### 【MB2198】

実行したプログラムの命令実行サイクル数, 実行時間を表示します。

実行時間の分解能は 25ns で最大 (2 の 56 乗 -1) × 25ns まで, サイクル数は最大 (2 の 56 乗 -1) すなわち 72057594037927935 サイクルまで測定可能です。

### ■ 例

```
>SHOW TIMER
```

```
<timer> From reset: 12345678
```

```
From Last Executed: 987654
```

```
<step> From reset: 6010712
```

```
From Last Executed: 501142
```

#### 【MB2198】

```
>SHOW TIMER
```

```
<cycle> From initialize: 4210826410 [Cycle]
```

```
From Last Executed: 362387415 [Cycle]
```

```
<timer> From initialize: 0h00m42s108ms264us250ns [Time]
```

```
From Last Executed: 0h00m03s623ms874us375ns [Time]
```



< 注意事項 >

エミュレータデバッガでは一度の実行で誤差が数十サイクル発生します。誤差が無視できる程小さくなるよう、多くのインストラクションを一度に実行することをお勧めします。  
シミュレータデバッガでは、パイプライン等の実行時間をシミュレートしていません。

---

## 4.10 CLEAR TIMER

---

タイマによる計測実行結果を初期化します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CLEAR TIMER

### ■ 機能

タイマによる計測実行結果を初期化します。

### ■ 例

```
>CLEAR TIMER
```

## 4.11 SET TRACE (type 1)

---

トレースバッファフルブレークの設定を行います。

---

### ■ 対応デバッガ

Simulator		
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

### ■ 形式

SET TRACE

#### ● コマンド修飾子

/BREAK

トレースバッファフルブレークを有効にします。

/NOBREAK ( 起動時デフォルト )

トレースバッファフルブレークを無効にします。

### ■ 機能

トレースバッファフルブレークを有効にすると、バッファフル時に、プログラム実行を中断します。

### ■ 例

```
>SET TRACE/BREAK
```

## 4.12 SET TRACE (type 2)

---

トレースバッファフルブレークの設定を行います。

---

### ■ 対応デバugga

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET TRACE

#### ● コマンド修飾子

- トレースバッファフルブレークの指定

/BREAK

トレースバッファフル時に、プログラム実行を中断します。

/NOBREAK ( 起動時デフォルト )

トレースバッファフル時に、プログラム実行を中断しません。

この時、トレース測定は継続されます。

- 【MB2198 ( FR60Lite, FR80S )】

/STOP

トレースバッファフル時に、プログラム実行を中断しません。

この時、トレース測定は停止されます。

/NOSTOP

トレースバッファフル時に、プログラム実行を中断しません。

この時、トレース測定は継続されます。

- トレースサンプリングモードの指定

/FULL ( 起動時デフォルト )

プログラム実行開始から停止までのトレース測定します。

FR60Lite または FR80S では指定できません。

/TRIGGER

コードイベント、データイベントおよびトレーストリガをトレース制御の要因に使用します。FR60Lite または FR80S では指定できません。

### ■ 機能

トレースバッファフルブレークを有効にすると、バッファフル時に、プログラム実行を中断します。

DSU3 および DSU4 チップでは、コードイベントおよびデータイベントをトレース測定に関連付けることができます。このトレース測定との関連付け方は、次の中から指定できます。これらは、トレースサンプリングモードと呼ばれます。

- フルモード

プログラム実行開始から停止までのトレース測定します。コードイベントおよびデータイベントは、プログラムの実行停止要因に使われます。

- トリガモード

プログラム実行開始時にはトレース測定を開始せず、コードイベントまたはデータイベントから停止までをトレース測定します。

コードイベントとデータイベントのトレースサンプリングモードは、個別に指定できません。

■ 例

```
>SET TRACE/TRIGGER /BREAK
```

## 4.13 SHOW TRACE (type 1)

---

トレースバッファに格納されているトレースデータを表示します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW TRACE [/DATA] [ トレース番号 [.. トレース番号 ] ]

SHOW TRACE /FILE[/APPEND] ファイル名

#### ● 機能分類コマンド修飾子

/DATA ( 省略時のデフォルト )

トレースしたデータを表示します。

#### ● パラメータ

トレース番号 ( デフォルト 10 進数 )

表示させたいトレースデータの番号を 10 進数で指定します。

ファイル名

表示したトレースデータを保存するファイル名を指定します。

#### ● コマンド修飾子

/CYCLE

有効バスサイクルでデータを表示します。DSU3 では、指定できません。

/INSTRUCTION

逆アセンブル表示を行います。DSU3 では、指定できません。

/SOURCE

ソース行単位のトレース結果を表示します。DSU3 では、指定できません。

/ONEFRAME

トレースデータを 1 行のみ表示します。

/FILE

トレースデータをファイルに保存します。

/APPEND

トレースデータをファイルに追加保存します。

/FILE 指定時に有効になります。

## ■ 機能

トレースバッファに格納されているトレースデータを表示します。

サンプリングされたトレースデータには番号が付けられ、実行停止位置(トリガポイント)のトレースデータには0番が、実行停止位置に到達するまでにサンプリングされたトレースデータには負の番号が付けられて表示されます。この番号をフレーム番号と呼びます。

/INSTRUCTION 指定時には、サンプリングされたトレースデータをもとに逆アセンブル表示を行います。

/SOURCE 指定時には、サンプリングされたトレースデータをもとにソース行表示を行います。

DSU3 チップでは、分岐時の PC 情報とデータアクセスの情報を表示します。このコマンドでは、分岐命令の間に実行された命令の情報を表示しないため、/CYCLE, /INSTRUCTION, /SOURCE 指定はできません。分岐命令の間に実行された命令の情報を表示するときは、SHOW DETAILTRACE コマンドをご使用ください。

トレース番号省略時は、最旧トレースデータまたは前回最後に表示したトレース番号の次のトレース番号のトレースデータから表示します。

表示開始トレース番号のみを指定した場合は、表示開始トレース番号から 12 個分表示します。

表示開始トレース番号が最旧トレースデータのトレース番号より小さい場合は、最旧トレースデータから表示します。

/STATUS 指定時には、現在のトレース状態を表示します。

MB2198 では、リアルタイムトレース I/F がある場合に、タイムスタンプを表示します。

/FILE 指定時にはトレースデータをファイルに上書き保存します。

/APPEND が指定されていると追加保存になります。

## ■ 例

```
> SHOW TRACE/SOURCE -65
frame no      source
-00065 :      demo 3.c$489    if (sy->str[0] = ab1)
-00059 :      demo 3.c$491    }
-00055 :      demo 3.c$487    for ( i = 0 ; i < 12 ; i++ ) {
-00052 :      demo 3.c$492                ackdat += 5;
-00047 :      demo 3.c$493                nckdat = ackdat;
-00043 :      demo 3.c$494                return (ab1);
-00042 :      demo 3.c$495    }
-00038 :      demo 3.c$464    if (rc != 0)
-00035 :      demo 3.c$465                sy->dat1 = 0x21;
-00031 :      demo 3.c$467    }
-00027 :      demo 3.c$460    for ( i = 0 ; i < NUM ; i++ ) {
-00024 :      demo 3.c$468    return (0);
> SHOW TRACE/FILE/APPEND    c:¥sample.log
```

## 4.14 SHOW TRACE (type 2)

---

トレース状態を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW TRACE /STATUS

#### ● コマンド修飾子

/STATUS

トレース測定条件およびトレース機能の有効 / 無効，トレースバッファの格納状況を表示します。

### ■ 機能

トレース状態を表示します。トレース測定条件およびトレース機能の有効 / 無効，トレースバッファの格納状況を表示します。

### ■ 例

```
>SHOW TRACE/STATUS  
en/dis = enable  
buffer full = nobreak  
frame no. = -00120 to 00000
```



## 4.15 CLEAR TRACE

---

トレースバッファをクリアします。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

CLEAR TRACE

### ■ 機能

トレースバッファをクリアします。

### ■ 例

>CLEAR TRACE

## 4.16 ENABLE TRACE

---

トレース機能を有効にします。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

ENABLE TRACE

### ■ 機能

トレース機能を有効にします。

### ■ 例

```
>ENABLE TRACE
```

## 4.17 DISABLE TRACE

---

トレース機能を無効にします。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

DISABLE TRACE

### ■ 機能

トレース機能を無効にします。

ただし、MCU動作モードが内部トレースモード、または外部トレースモード(DSU3チップのみ)の場合、トレース機能の無効指定はできません。

### ■ 例

```
>DISABLE TRACE
```

## 4.18 SEARCH TRACE

指定された条件でトレースデータを検索します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

< 形式 1>

SEARCH TRACE [ アドレス [&= マスクデータ ] ] [ , f= 検索開始番号 ]

< 形式 2>

SEARCH TRACE [d= データ [&= マスクデータ ] ] [ , f= 検索開始番号 ]

#### ● パラメータ

アドレス ( アドレス式 )

検索したいアドレスを指定します。

データ ( データ式 )

検索したいデータを指定します。

デバッガ種別がエミュレータデバッガでのみ有効。

マスクデータ ( データ式 )

アドレスおよびデータをマスクして検索します。

1 が立っているビットのみを比較して検索します。

検索開始番号 ( デフォルト 10 進数 )

検索開始フレーム番号を指定します。

省略した場合は、トレースバッファの先頭から検索します。

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

該当するフレームをすべて検索します。

/ONEFRAME

1 フレーム見つかった時点で終了します。

/CYCLE

有効バスサイクル単位でデータを検索します。DSU3 では指定できません。

**/INSTRUCTION**

機械語実行単位に並び変えたデータを検索します。

機械語実行単位に並び変えることができない場合には、マシンサイクルでの検索になります。

**/BYTE (DSU3 のみ)**

検索条件として、データを1バイト長として扱うことを指定します。

データ長指定は、形式(2)でのみ、指定できます。

省略するとデータ長になります。

**/HALFWORD (DSU3 のみ)**

検索条件として、データを2バイト長として扱うことを指定します。

**/WORD (DSU3 のみ)**

検索条件として、データを4バイト長として扱うことを指定します。

**/CODE (DSU3 のみ)**

指定アドレスを命令コードアクセスしたトレースフレームまたはステップを検索します。

**/READ (DSU3 のみ)**

指定アドレスをリードアクセスしたトレースフレームまたはステップを検索します。

**/WRITE (DSU3 のみ)**

指定アドレスをライトアクセスしたトレースフレームまたはステップを検索します。

## ■ 機能

指定された条件でトレースデータを検索します。

トレースデータが見つかった場合は、SHOW TRACE コマンドと同じ形式で表示します。

コマンド修飾子に /ONEFRAME が指定された場合は、1 フレーム見つかった時点でコマンドを終了します。

トレースデータが見つかった場合は、組込み変数 "%SCHFRM" にフレーム番号をセットします。

複数見つかった場合は、最後に見つかった番号をセットします。

## ■ 例

```
>SEARCH TRACE/INSTRUCTION 0xF0AE6
frame no.      address  mnemonic
-00010        :  000F0AE6  ENTER#004
-00009        :  000F0AE8  LEAVE
-00008        :  000F0AEA  LD      @R15+ , RP
-00007        :  000F0AEC  RET
-00006        :  000F0ADE  LEAVE
-00005        :  000F0AE0  LD      @R15+ , RP
-00004        :  000F0AE2  RET
-00003        :  000F0ACE  LD      @ (R14 , -4) , R4
-00002        :  000F0AD0  LEAVE
-00001        :  000F0AD2  LD      @R15+ , RP
00000         :  000F0AD4  ADDSP#4
```

## 4.19 SET DATATRACEAREA

トレース測定を行うデータアクセス領域を指定します。

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SET DATATRACEAREA アドレス [&= アドレスマスク]

#### ● パラメータ

アドレス (アドレス式)

データアクセスされる、トレース測定をしたいアドレスを指定します。

アドレスマスク (データ式)

アドレスとビットマスクを取り、トレース測定をしたいデータアクセスのアドレス領域を指定します。

1 が立っているビットのみを比較して、アドレス領域を設定します。

#### ● コマンド修飾子

/READ

指定アドレス領域をリードアクセスしたとき、トレース測定を行います。

/WRITE

指定アドレス領域をライトアクセスしたとき、トレース測定を行います。

/CODE

指定アドレス領域を命令アクセスしたとき、トレース測定を行います。

### ■ 機能

トレース測定を行うデータアクセス領域を指定します。データアクセス領域は、指定されたアドレスについて、アドレスマスクの1が立っているビットだけをチェックすることで指定されます。

コマンド修飾子を指定しないときは、/READ/WRITE/CODE 指定として処理します。

アドレスマスクを指定しないときは、アドレスマスク H'0FFFFFFFFF として処理します。

### ■ 例

```
>SET DATATRACEAREA/WRITE 180 & 7f
```

## ■ 制限事項

本コマンドは MCU 動作モードと DSU 種別により、以下のように使用条件が異なります。

MCU 動作モード	DSU 種別	
	DSU3	DSU4
フルトレースモード		
リアルタイムモード		
外部トレースモード		アダプタボード接続 :
		ケーブル接続 :
内部トレースモード	×	×

：コマンド有効  
 コマンド修飾子 /CODE が指定可能

：コマンド有効  
 コマンド修飾子 /CODE が指定不可  
 コード属性のデータは常に取得します。

×：コマンド無効  
 全領域に対しリード / ライト / コード属性のデータを常に取得します。

## 4.20 SHOW DATATRACEAREA

---

設定されたデータトレース測定領域を表示します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW DATATRACEAREA

### ■ 機能

設定されたデータトレース測定領域を表示します。  
内部トレースモードの場合は使用できません。

### ■ 例

```
>SHOW DATATRACEAREA  
address      mask      access  
00000000  FFFFFFFF  read/write/code
```



## 4.21 CANCEL DATATRACEAREA

---

設定したデータトレース測定領域を削除します。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

CANCEL DATATRACEAREA

### ■ 機能

設定したデータトレース測定領域を削除します。  
内部トレースモードの場合は使用できません。

### ■ 例

```
>SHOW DATATRACEAREA
address      mask      access
00001000 00001FFF   read

>CANCEL DATATRACEAREA
address      mask      access
00000000 FFFFFFFF read/write/code
```

## 4.22 SHOW DETAILTRACE

---

トレースの詳細表示をおこないます。

---

### ■ 対応デバッガ

Simulator	×
Emulator	(MB2197) (MB2198)
Monitor	×

### ■ 形式

SHOW DETAILTRACE [ フレーム内の表示開始行 ] [ , f= フレーム番号 ]

#### ● パラメータ

フレーム内の表示開始行 ( デフォルト 10 進数 )

指定したフレームについて、詳細表示を開始する行数を指定します。

フレーム番号 ( デフォルト 10 進数 )

表示するフレームの番号を指定します。省略した場合は、前回と同じフレーム番号を使用します。

#### ● コマンド修飾子

/INSTRUCTION ( 省略時のデフォルト )

逆アセンブル表示を行います。

/SOURCE

ソース行単位のトレース結果を表示します。

/ONEFRAME

トレースデータを 1 行のみ表示します。

## ■ 機能

DSU3 チップでのみ指定可能なコマンドです。

SHOW TRACE コマンドで表示した分岐アドレス情報とデータアクセス情報について、SHOW DETAILTRACE は次の処理を行い、結果を表示します。

### ● 分岐アドレス情報の補完

分岐アドレス情報から、分岐命令間の命令を逆アセンブルにより求めます。

### ● データアクセス命令の検索

分岐命令間について、逆アセンブル結果から該当するデータアクセスを行った命令を求めます。

/ONEFRAME を省略した場合には、12 行分の情報を表示します。

## ■ 例

```
>SHOW DETAILTRACE 1, f=-198
frame no.      address      mnemonic (-00198 .. -00194)
-00198      :      000F0052      BNE      000F0048
              :      000F0054      LDI:32 #00005632 , R14
              :      000F005A      LDUH      @R14 , R0
-00197      :      read          1234 at 00005632
              :      000F005C      LDUB      @ (R14 , 2) , R1
-00196      :      read          98 at 00005634
              :      000F005E      LDI:8   #10 , R2
              :      000F0060      STB      R2 , @ (R14 , 4)
-00195      :      write          10 at 00005636
-00194      :      000F0062      BRA      000F0048
```

## 4.23 SET TRACETRIGGER

指定した条件を満たしたとき，トレース測定の状態を制御します。

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SET TRACETRIGGER アドレス [&= アドレスマスク ] [, [!] d= データ [&= データマスク ]]

#### ● パラメータ

##### アドレス (アドレス式)

トレーストリガ条件となるアドレスを指定します。

##### アドレスマスク (データ式)

トレーストリガ条件となるアドレスマスクビットパターンを指定します。

1 が立っているビット位置のアドレス値のみを比較対象にします。

##### データ (データ式)

トレーストリガ条件となるデータを指定します。コード実行条件では指定できません。

/ANYTHING を指定した場合 ,32 ビットデータを比較対象にします。

(8/16 ビットデータを比較する場合は , マスク指定する必要があります)

##### マスクデータ (データ式)

トレーストリガ条件となるデータマスクビットパターンを指定します。

1 が立っているビット位置のデータ値のみを比較対象にします。コード実行条件では指定できません。

#### ● コマンド修飾子

##### - アクセス属性

/READ

データリードアクセスをトレーストリガ条件に指定します。

FR80S では指定できません。

/WRITE(FR80S での省略時のデフォルト)

データライトアクセスをトレーストリガ条件に指定します。

/CODE (FR60Lite での省略時のデフォルト)

コード実行をトレーストリガ条件に指定します。

FR80S では指定できません。

## - データ長指定

/BYTE

バイトアクセスをトレーストリガ条件に指定します。/CODE を指定した場合には、指定できません。

/HALFWORD

ハーフワード (2 バイト) アクセスをトレーストリガ条件に指定します。/CODE を指定した場合には、指定できません。

/WORD

ワード (4 バイト) アクセスをトレーストリガ条件に指定します。/CODE を指定した場合には、指定できません。

/ANYTHING (省略時のデフォルト)

データ長にかかわらず、指定アドレスへのアクセスをトレーストリガ条件に指定します。

## - トレース制御

/ENABLETRACE (省略時のデフォルト)

トレーストリガ条件を満たしたとき、トレース測定を開始します。

/DISABLETRACE

トレーストリガ条件を満たしたとき、トレース測定を停止します。

## ■ 機能

指定した条件を満たしたとき、トレース測定の状態を制御します。

最大 4 ポイント設定できますが、SET BREAK/DATAWATCH と SET EVENT の設定によって使用できるポイント数は変動します。

## ■ 例

```
>SET TRACETRIGGER/CODE/ENABLETRACE FF0000
>SET TRACETRIGGER/READ/WORD/ENABLETRACE 120046 !D=1234&=FFFC
```

## &lt; 注意事項 &gt;

- 1) FR60Lite または FR80S 使用時以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「トレース」を参照してください。
- 2) FR80S では内蔵 RAM 領域に指定した場合のみ有効となります。
- 3) イベントモードがパフォーマンスモードの場合は使用できません。
- 4) /ANYTHING 修飾子を指定した場合、32 ビットデータを比較対象にします。  
8/16 ビットデータを比較する場合は、マスクデータを指定する必要があります。

例) 8 ビットデータ (0x12) の場合

```
SET TRACETRIGGER/ANYTHING 0x10000, D=0x12000000&=0xff000000
```

16 ビットデータ (0x4567) の場合

```
SET TRACETRIGGER/ANYTHING 0x10000, D=0x45670000&=0xffff0000
```

## 4.24 SHOW TRACETRIGGER

---

SET TRACETRIGGER コマンドで設定したトレーストリガの状態を表示します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW TRACETRIGGER [ トリガ番号 [, ...] ]

#### ● パラメータ

トリガ番号 ( デフォルト 10 進数 )

表示するトレーストリガの番号を指定します。

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

SET TRACETRIGGER コマンドで設定したトレーストリガをすべて表示します。

### ■ 機能

SET TRACETRIGGER コマンドで設定したトレーストリガの状態を表示します。

### ■ 例

```
>SHOW TRACETRIGGER

no.   trace-ctrl  address  address-mask  data  d-mask  size  access  symbol
1     enable    00FF0010  00FFFFFF0    ----  ----   ----   code
2     disable   01200430  -----    !A5A5  F5FF   word  read/write  symbol
```

### < 注意事項 >

- 1) FR60Lite または FR80S 使用時以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「トレース」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

## 4.25 CANCEL TRACETRIGGER

---

SET TRACETRIGGER コマンドで設定したトレーストリガを削除します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

CANCEL TRACETRIGGER [ トリガ番号 [, ...] ]

#### ● パラメータ

トリガ番号 (デフォルト 10 進数)

削除するトレーストリガの番号を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

SET TRACETRIGGER コマンドで設定したトレーストリガをすべて削除します。

### ■ 機能

SET TRACETRIGGER コマンドで設定したトレーストリガを削除します。

### ■ 例

```
>CANCEL TRACETRIGGER/ALL
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「トレース」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

## 4.26 ENABLE TRACETRIGGER

---

SET TRACETRIGGER コマンドで設定したトレーストリガを有効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

ENABLE TRACETRIGGER [ トリガ番号 [, ...] ]

#### ● パラメータ

トリガ番号 ( デフォルト 10 進数 )

有効にするトレーストリガの番号を指定します。

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

SET TRACETRIGGER コマンドで設定したトレーストリガをすべて有効にします。

### ■ 機能

SET TRACETRIGGER コマンドで設定したトレーストリガを有効にします。

### ■ 例

```
>ENABLE TRACETRIGGER/ALL
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「トレース」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-



## 4.27 DISABLE TRACETRIGGER

---

SET TRACETRIGGER コマンドで設定したトレーストリガを無効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

DISABLE TRACETRIGGER [ トリガ番号 [, ...] ]

#### ● パラメータ

トリガ番号 (デフォルト 10 進数)

無効にするトレーストリガの番号を指定します。

#### ● コマンド修飾子

/ALL (省略時のデフォルト)

SET TRACETRIGGER コマンドで設定したトレーストリガをすべて無効にします。

### ■ 機能

SET TRACETRIGGER コマンドで設定したトレーストリガを無効にします。

### ■ 例

```
>DISABLE TRACETRIGGER/ALL
```

---

#### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「トレース」を参照してください。
  - 2) イベントモードがパフォーマンスモードの場合は使用できません。
-

## 4.28 SET RAMCHECK

RAM チェッカの設定を行います。

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

[ 形式 1 ] SET RAMCHECK アドレス

[ 形式 2 ] SET RAMCHECK ファイル名

#### ● パラメータ

- 形式 1  
アドレス  
設定する監視アドレスを指定します。
- 形式 2  
ファイル名  
サンプリングデータを保存するログファイル名を指定します。

#### ● コマンド修飾子

- 形式 1
  - データサイズ  
/BYTE  
データサイズをバイトで指定します。
  - /HALFWORD  
データサイズをハーフワードで指定します。監視アドレスの下位 1 ビットは無視されます。
  - /WORD (省略時のデフォルト)  
データサイズをワードで指定します。監視アドレスの下位 2 ビットは無視されます。
  - アクセス属性  
/READ  
データリードアクセスをデータ監視条件に指定します。  
FR80S では指定できません。
  - /WRITE  
データライトアクセスをデータ監視条件に指定します。  
FR60Lite では省略時は, /READ/WRITE に設定されます。  
FR80S では省略時は, /WRITE に設定されます。

- 形式2

/SOFTUNE

ファイルを SOFTUNE 形式で保存する場合に指定します。

拡張子を省略すると ".SRL" が付加されます。

/CSV

ファイルを CSV 形式で保存する場合に指定します。

拡張子を省略すると ".CSV" が付加されます。

## ■ 機能

- 形式1

RAM チェッカの監視アドレス, データサイズ, アクセス属性を設定します。

監視アドレスは8ポイント設定可能です。

監視アドレスは設定した順番でログへ格納されます。

- 形式2

サンプリングデータを取得するログファイルと保存形式を指定します。

コマンド修飾子は省略できません。/SOFTUNE または /CSV の指定が必要です。

## ■ 例

```
> SET RAMCHECK 0x0003C100
> SHOW RAMCHECK
Logging mode      : disable
Logging name      :
Logging format    : SOFTUNE
ch 0 : 0003C000 / word / RW : main¥dat_0
> SET RAMCHECK /BYTE/WRITE 0x0003C110
> SHOW RAMCHECK
Logging mode      : disable
Logging name      :
Logging format    : SOFTUNE
ch 0 : 0003C100 / word / RW : main¥dat_0
ch 1 : 0003C110 / byte / W  : main¥dat_8
> SET RAMCHECK /SOFTUNE TEMP2
> SHOW RAMCHECK
Logging mode      : disable
Logging name      : TEMP2.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 / word / RW : main¥dat_0
ch 1 : 0003C110 / byte / W  : main¥dat_8
```

### < 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「RAM チェッカ」を参照してください。
  - 2) デバッグモードが RAM Checker モードの場合は使用できません。
  - 3) FR80S では内蔵 RAM 領域に指定した場合のみ有効となります。
-

## 4.29 SHOW RAMCHECK

---

RAM チェッカの設定内容を表示します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

SHOW RAMCHECK

#### ● パラメータ

なし

#### ● コマンド修飾子

なし

### ■ 機能

RAM チェッカの設定内容を表示します。

### ■ 例

```
>SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
```

### < 注意事項 >

- 1) FR60Lite または FR80S 使用時以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「RAM チェッカ」を参照してください。
  - 2) デバッグモードが RAM Checker モードの場合のみ使用可能です。
-

## 4.30 CANCEL RAMCHECK

---

RAM チェッカの設定を削除します。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

[ 形式 1] CANCEL RAMCHECK アドレス

[ 形式 2] CANCEL RAMCHECK

#### ● パラメータ

アドレス

削除する監視アドレスを入力します。

アドレスを指定した場合、コマンド修飾子は指定できません。

#### ● コマンド修飾子

/ALL

設定中の監視アドレスをすべて削除します。

コマンド修飾子を指定した場合、パラメータは入力できません。

### ■ 機能

RAM チェッカの監視アドレスを削除します。

### ■ 例

```
>SHOW RAMCHECK
Logging mode    : enable
Logging name    : TEMP.SRL
Logging format   : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
>CANCEL RAMCHECK 0x0003C102
>SHOW RAMCHECK
Logging mode    : enable
Logging name    : TEMP.SRL
Logging format   : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C104 /word / RW : main¥dat_2
```

```
>CANCEL RAMCHECK /ALL  
>SHOW RAMCHECK  
Logging mode      : enable  
Logging name      : TEMP.SRL  
Logging format    : SOFTUNE
```

---

< 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「RAM チェッカ」を参照してください。
  - 2) デバッグモードが RAM Checker モードの場合のみ使用可能です。
-

## 4.31 ENABLE RAMCHECK

---

RAM チェッカのロギング状態を有効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

ENABLE RAMCHECK

- パラメータ

なし

- コマンド修飾子

なし

### ■ 機能

RAM チェッカのロギング状態を有効にします。

イベントモードがパフォーマンス時、このコマンドは使用できません。

### ■ 例

```
>SHOW RAMCHECK
Logging mode    : disable
Logging name    : TEMP.SRL
Logging format  : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
>ENABLE RAMCHECK
>SHOW RAMCHECK
Logging mode    : enable
Logging name    : TEMP.SRL
Logging format  : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
```



< 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「RAM チェッカ」を参照してください。
  - 2) デバッグモードが RAM Checker モードの場合のみ使用可能です。
-

## 4.32 DISABLE RAMCHECK

---

RAM チェッカのロギング状態を無効にします。

---

### ■ 対応デバugga

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

DISABLE RAMCHECK

#### ● パラメータ

なし

#### ● コマンド修飾子

なし

### ■ 機能

RAM チェッカのロギング状態を無効にします。

### ■ 例

```
>SHOW RAMCHECK
Logging mode    : enable
Logging name    : TEMP.SRL
Logging format   : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
>DISABLE RAMCHECK
>SHOW RAMCHECK
Logging mode    : disable
Logging name    : TEMP.SRL
Logging format   : SOFTUNE
ch 0 : 0003C100 /word / RW : main¥dat_0
ch 1 : 0003C102 /byte / RW : main¥dat_1
ch 2 : 0003C104 /word / RW : main¥dat_2
```

< 注意事項 >

- 1) FR60Lite または FR80S 使用时以外では、このコマンドは使用できません。詳しくは SOFTUNE Workbench 機能説明書「RAM チェッカ」を参照してください。
  - 2) デバッグモードが RAM Checker モードの場合のみ使用可能です。
-



# 第5章

---

## メモリ/レジスタ 操作コマンド

この章では、メモリ/レジスタ操作コマンドについて説明します。

- 5.1 EXAMINE
- 5.2 ENTER
- 5.3 SET MEMORY
- 5.4 SHOW MEMORY
- 5.5 SEARCH MEMORY
- 5.6 SET REGISTER
- 5.7 SHOW REGISTER
- 5.8 COMPARE
- 5.9 FILL
- 5.10 MOVE
- 5.11 DUMP

## 5.1 EXAMINE

---

指定された C/C++ 言語の式を解析し, その値を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

\*: DSU3 評価用チップでは, プログラム実行中に使用できません。

### ■ 形式

EXAMINE 式 [ , ... ]

#### ● パラメータ

式 ( アドレス式 )

解析する式を指定します。

#### ● コマンド修飾子

/BINARY

確認する値を 2 進数で表示することを指定します。

/OCTAL

確認する値を 8 進数で表示することを指定します。

/DECIMAL

確認する値を 10 進数で表示することを指定します。

/HEXADECIMAL

確認する値を 16 進数で表示することを指定します。

### ■ 機能

指定された C/C++ 言語の式を解析し, その値を表示します。

変数が指定された場合はその内容が表示されます。

構造体型, 共用体型およびクラスの変数を指定するとすべてのメンバの値を表示し, 配列名のみを指定するとその配列の内容をすべて表示します。

コマンド修飾子の表示基数を省略した場合は, SET RADIX コマンドで指定された基数となります。

## ■ 例

```
>EXAMINE strsym
strsym =      {
                a = H'20
                b = H'4A30
                c = H'3012
            }
>EXAMINE strsym.a
strsym.a = H'20
>EXAMINE flags  [0]
flags  [0] = H'03
>EXAMINE flags
flags  [0] = H'05
flags  [1] = H'50
flags  [2] = H'10
flags  [3] = H'2A
>EXAMINE/DECIMAL count
count = D'12
>EXAMINE/HEXADECIMAL count
count = H'0C
>EXAMINE/DECIMAL fwork
fwork = 2.36S+1
```

## 5.2 ENTER

---

指定された変数に指定データを代入します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※:DSU3 評価用チップでは、プログラム実行中に使用できません。

### ■ 形式

ENTER 変数 = データ

#### ● パラメータ

変数 ( アドレス式 )

データを格納する変数を指定します。

データ ( データ式 )

格納する値を指定します。

#### ● コマンド修飾子

##### • データ長

/BYTE

格納する値をバイト長で指定されたメモリ位置へ格納します。

/HALFWORD

格納する値を 2 バイト長として指定されたメモリ位置へ格納します。

/WORD

格納する値を 4 バイト長で指定されたメモリ位置へ格納します。

/DWORD

格納する値を 8 バイト長で指定されたメモリ位置へ格納します。

/SINGLE

格納する値を単精度浮動小数点数として指定されたメモリ位置へ格納します。

/DOUBLE

格納する値を倍精度浮動小数点数として指定されたメモリ位置へ格納します。



## ■ 機能

指定された変数に指定データを代入します。

コマンド修飾子の型を指定すると、指定の大きさでデータを代入することができます。

## ■ 例

```
>ENTER tmcnt = 10
>ENTER work = 6A5
>ENTER tmp = 1DF2BF
>ENTER fsymbol = F'10.55S+2
>ENTER/WORD work = 12345678
```

## 5.3 SET MEMORY

データを指定されたコマンド修飾子の型 (タイプ) で, 格納アドレスによって指定したメモリ位置に格納します。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※: DSU3 評価用チップでは, プログラム実行中に使用できません。

### ■ 形式

SET MEMORY [ 格納アドレス ] = データ [ , ... ]

#### ● パラメータ

格納アドレス (アドレス式)

データを格納するメモリ位置を指定します。

データ (データ式)

格納する値を指定します。

#### ● コマンド修飾子

##### • データ長

/BIT

格納する値をビット長で指定されたメモリ位置へ格納します。

/BYTE (省略時のデフォルト)

格納する値をバイト長で指定されたメモリ位置へ格納します。

/HALFWORD

格納する値を 2 バイト長として指定されたメモリ位置へ格納します。

/WORD

格納する値を 4 バイト長で指定されたメモリ位置へ格納します。

/DWORD

格納する値を 8 バイト長で指定されたメモリ位置へ格納します。

/SINGLE

格納する値を単精度浮動小数点数として, 指定されたメモリ位置へ格納します。

/DOUBLE

格納する値を倍精度浮動小数点数として, 指定されたメモリ位置へ格納します。

/STRING

格納する値を文字列で指定し, アスキーコードで, 指定されたメモリ位置へ格納します。

## ■ 機能

データを指定されたコマンド修飾子の型 (タイプ) で、格納アドレスによって指定したメモリ位置に格納します。

格納アドレスを省略した場合は、SHOW MEMORY コマンドまたは SET MEMORY コマンドで最後にアクセスしたメモリ位置の次のメモリ位置に対して格納します。

データの型も最後のアクセスと同じものとなります。

格納アドレスにピリオド "." のみを指定した場合は、SHOW MEMORY コマンドまたは SET MEMORY で最後にアクセスしたメモリ位置に対して格納します。

データの型も最後のアクセスと同じものとなります。

コマンド修飾子の型 (タイプ) を省略すると、/BYTE になります。

## ■ 例

```
>SET MEMORY/BYTE 1000 = 10
>SET MEMORY/HALFWORD 1030 = 6A5
>SET MEMORY/WORD 1050 = 1DF2BF
>SET MEMORY/STRING 2000 = "ST"
>SET MEMORY . = 45
>SET MEMORY/BIT 8000:3 = 1
>SET MEMORY/SINGLE 2050 = F'10.55S+2
```

## 5.4 SHOW MEMORY

アドレスまたはアドレス範囲によって指定されたメモリ位置の内容を、指定されたデータの型で表示します。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

※: DSU3 評価用チップでは、プログラム実行中に使用できません。

### ■ 形式

SHOW MEMORY [ { アドレス | アドレス範囲 } [, ... ] ]

#### ● パラメータ

アドレス (アドレス式)

確認するメモリ位置のアドレスを指定します。

アドレス範囲 (アドレス式)

確認するメモリ領域の範囲を指定します。

#### ● コマンド修飾子

/BIT

確認する値を 1 ビット長で表示することを指定します。

/BYTE (省略時のデフォルト)

確認する値を 1 バイト長で表示することを指定します。

/HALFWORD

確認する値を 2 バイト長で表示することを指定します。

/WORD (32 ビット)

確認する値を 4 バイト長で表示することを指定します。

/DWORD

確認する値を 8 バイト長で表示することを指定します。

/SINGLE

確認する値を単精度浮動小数点数で表示することを指定します。

/DOUBLE

確認する値を倍精度浮動小数点数で表示することを指定します。

/ASCII

確認する値をアスキー文字で表示することを指定します。

/STRING

確認する値を文字列で表示することを指定します。

/BINARY

確認する値を2進数で表示することを指定します。

/OCTAL

確認する値を8進数で表示することを指定します。

/DECIMAL

確認する値を10進数で表示することを指定します。

/HEXADECIMAL

確認する値を16進数で表示することを指定します。

## ■ 機能

アドレスまたはアドレス範囲によって指定されたメモリ位置の内容を、指定されたデータの型で表示します。ただし、/BITを指定した場合は、アドレス範囲を指定することはできません。

アドレスおよびアドレス範囲を省略した場合は、SHOW MEMORY または SET MEMORY コマンドで最後にアクセスしたメモリ位置の次のメモリ位置を表示します。

データの型も最後のアクセスと同じものとなります。

アドレスにピリオド(".")のみを指定した場合は、SHOW MEMORY または SET MEMORY コマンドで最後にアクセスしたメモリ位置を表示します。

データの型も最後のアクセスと同じものとなります。

コマンド修飾子の型(タイプ)を省略すると、/BYTEを指定したことになります。

また、コマンド修飾子の表示基数を省略した場合は、SET RADIX コマンドで指定された基数となります。

## ■ 例

```
>SHOW MEMORY/DECIMAL 1000
00001000 = D'12
>SHOW MEMORY/BINARY 1000
00001000 = B'00001100
>SHOW MEMORY/HEXADECIMAL 1000..1001
00001000 = H'0C
00001001 = H'41
>SHOW MEMORY/HEXADECIMAL/HALFWORD 1000
0000 1000 = D'410C
>SHOW MEMORY/HEXADECIMAL/WORD 1000
00001000 = H'0030410C
>SHOW MEMORY/HEXADECIMAL 1000 1020
00001000 = H'0C
00001020 = H'E3
>SHOW MEMORY/ASCII 1001
00001000 = 'A'
>SHOW MEMORY/SINGLE/DECIMAL 1030
00001030 = 2.36S+1
>SHOW MEMORY/BYTE 1000
000 01000 = H'0C
>SHOW MEMORY .
00001000 = H'0C
>SHOW MEMORY
00001001 = H'41
```

## 5.5 SEARCH MEMORY

指定されたデータを指定されたメモリの中から検索し、一致したアドレスを表示します。

### ■ 対応デバッグ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SEARCH MEMORY アドレス範囲 = データ [ , ... ] [ , S= スキップバイト数 ]

#### ● パラメータ

アドレス範囲 ( アドレス式 )

検索を行うメモリ領域を指定します。

データ ( データ式 )

検索するデータを指定します。

スキップバイト数 ( データ式 )

検索する際のスキップバイト数を指定します。

(H'1 ~ H'FFFF) が指定できます。

省略するとデータ長になります。

#### ● コマンド修飾子

##### • データ長

/BYTE ( 省略時のデフォルト )

データをバイト長として検索します。

/HALFWORD

データを2バイト長として検索します。

2バイトごとスキップして検索します。

/WORD

データを4バイト長として検索します。

4バイトごとスキップして検索します。

/DWORD

データを8バイト長として検索します。

8バイトごとスキップして検索します。

/ASCII

データをアスキー文字列として検索します。

**■ 機能**

指定されたデータを指定されたメモリの中から検索し、一致したアドレスを表示します。

**■ 例**

```
>SEARCH MEMORY 2000..3000 = 88  
found at = 00002050  
found at = 00002577  
found at = 00002BDF
```

## 5.6 SET REGISTER

---

指定されたレジスタ , フラグに値を設定します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET REGISTER レジスタ名 = データ

#### ● パラメータ

レジスタ名

変更したいレジスタおよびフラグ名を指定します。

データ ( データ式 )

レジスタ , フラグに設定したい値を指定します。

### ■ 機能

指定されたレジスタ , フラグに値を設定します。

### ■ 例

```
>SET REGISTER PC = 1000  
>SET REGISTER C = 1
```

---

#### < 注意事項 >

- レジスタ TBR について , I/O 領域とベクタテーブルが重なるように指定するとプログラムが暴走するので , ご注意ください。
  - レジスタ TBR について , H'0FFFFFFC00 を超える値を指定するとパラメータエラーとなります。
-



# 5.7 SHOW REGISTER

指定されたレジスタおよびフラグの内容を表示します。表示は 16 進数で行われます。

## ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

## ■ 形式

SHOW REGISTER レジスタ名

- パラメータ  
レジスタ名  
確認したいレジスタおよびフラグ名を指定します。
- コマンド修飾子  
/ALL ( 省略時のデフォルト )  
すべてのレジスタ , フラグの内容を表示します。

## ■ 機能

指定されたレジスタおよびフラグの内容を表示します。表示は 16 進数で行われます。  
フラグレジスタの内容は , フラグがセットされていなければ , " - " を表示し , セットされていればそのフラグ名を表示します。

## ■ 例

```
>SHOW REGISTER PC
PC = 00FF0000
>SHOW REGISTER
R0 = 00000000    R1 = 00000000    R2 = 00000000    R3 = 00000000
R4 = 00000000    R5 = 00000000    R6 = 00000000    R7 = 00000000
R8 = 00000000    R9 = 00000000    R10 = 00000000   R11 = 00000000
R12 = 00000000   R13 = 00000000    R14 = 00000000   R15 = 0000FFC0
MDH = 00000000   MDL = 00000000    RP = 00000000    PS = FFFFFFFF
PC = 000FF000    USP = 0000E000    SSP = 0000FFC0   CCR = --SINZVC
SCR = --T        ILM = 1F          TBR = 000FFC00
```

## 5.8 COMPARE

---

メモリ内容の比較を行います。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

COMPARE 比較元アドレス範囲, 比較先アドレス

#### ● パラメータ

比較元アドレス範囲 (アドレス式)

比較元のメモリ領域を指定します。

比較先アドレス (アドレス式)

比較先アドレスを指定します。

### ■ 機能

メモリ内容の比較を行います。

比較の結果, 相違箇所がない場合は「Not found」と表示します。

相違箇所があった場合は, 左側に比較元のメモリ位置とその内容を, 右側に比較先のメモリ位置とその内容を 16 進数で表示します。

### ■ 例

```
>COMPARE 2000..3000, 4000
address      source      destination      address
00002050     35         10             00004050
00002051     40         00             00004051
```

## 5.9 FILL

指定されたメモリ領域へ任意のデータを充てんします。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

FILL アドレス範囲 = データ [ , ... ]

#### ● パラメータ

アドレス範囲 ( アドレス式 )

充てんを行うメモリ範囲を指定します。

データ ( データ式 )

充てんするデータを指定します。

#### ● コマンド修飾子

##### ・ データ長

/BYTE ( 省略時のデフォルト )

データをバイト長で充てんすることを指定します。

/HALFWORD

データを2バイト長で充てんすることを指定します。

/WORD

データを4バイト長で充てんすることを指定します。

/DWORD

データを8バイト長で充てんすることを指定します。

/ASCII

データをアスキー文字列で充てんすることを指定します。

### ■ 機能

指定されたメモリ領域へ任意のデータを充てんします。

### ■ 例

```
>FILL 2000..2FFF = 23
```

## 5.10 MOVE

---

指定されたメモリのデータを指定された転送先へ転送します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

MOVE 転送元アドレス範囲, 転送先アドレス

#### ● パラメータ

転送元アドレス範囲 ( アドレス式 )

転送を行うメモリ領域を指定します。

転送先アドレス ( アドレス式 )

転送先のメモリ位置を指定します。

### ■ 機能

指定されたメモリのデータを指定された転送先へ転送します。

### ■ 例

```
>MOVE 2000..3000, 4000
```

## 5.11 DUMP

指定されたメモリの内容をダンプします。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

\*: DSU3 評価用チップでは、プログラム実行中に使用できません。

### ■ 形式

DUMP [ { 開始アドレス | アドレス範囲 } ]

#### ● パラメータ

開始アドレス ( アドレス式 )

ダンプを開始するメモリ位置のアドレスを指定します。

アドレス範囲 ( アドレス式 )

ダンプするメモリ領域の範囲を指定します。

#### ● コマンド修飾子

- 表示単位

/BIT

ビット単位でダンプします。

/BYTE ( 起動時のデフォルト )

バイト単位でダンプします。

/HALFWORD

2 バイト単位で表示します。

/WORD

4 バイト単位で表示します。

/DWORD

8 バイト単位で表示します。

### ■ 機能

指定されたメモリの内容をダンプします。

開始アドレスのみ指定した場合は、出力ウィンドウの 16 行分表示します。

パラメータを省略した場合は、前回のコマンド実行で最後に表示されたメモリ位置の次から表示します。

## 第5章 メモリ/レジスタ 操作コマンド

### ■ 例

```
>DUMP 100..118
address  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ---ascii--
00000100 00 00 41 42 43 00 00 00 00 00 00 00 00 00 00 00 ..ABC.....
00000110 53 49 4D 55 4C 41 54 4F 52                               SIMULATOR
>
>DUMP/HALFWORD 100..118
address  +0  +2  +4  +6  +8  +A  +C  +E  ---ascii--
00000100 0000 4241 0043 0000 0000 0000 0000 0000 ..ABC.....
00000110 4953 554D 414C 4F54 0052                               SIMULATOR
>
>DUMP/BIT 5
address   :7 :6 :5 :4 :3 :2 :1 :0  HEX
00000005  0  1  1  0  1  0  1  0   6A
00000006  1  1  1  1  0  1  0  0   F4
```

## 第6章

---

# アセンブル/ 逆アセンブルコマンド

この章では、アセンブル/逆アセンブルコマンドについて説明します。

6.1 ASSEMBLE

6.2 DISASSEMBLE

## 6.1 ASSEMBLE

入力されたニーモニックとオペランドをラインアセンブルして、命令コードを指定されたメモリ位置へ格納します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

ASSEMBLE [ 開始アドレス ] = アセンブル文字列

#### ● パラメータ

開始アドレス (アドレス式)

ラインアセンブルしたコードを格納したいメモリの先頭位置を指定します。

アセンブル文字列 (文字列)

ラインアセンブルする文字列を指定します。文字列をダブルクォーテーション (") でくくってください。

### ■ 機能

入力されたニーモニックとオペランドをラインアセンブルして、命令コードを指定されたメモリ位置へ格納します。

開始アドレス省略時は、前回最後に命令コードを格納したメモリ位置の次のメモリ位置からニーモニックの入力待ちとなります。

### ■ 例

```
>ASSEMBLE 1000 = "RET"
>ASSEMBLE 1006 = "ADD #1, R1"
>DISASSEMBLE 1000
00001000      RET
00001002      LDI #0, R0
00001004      LDUB @R0, R1
00001006      ADD #1, R1
00001008      STB R1, @R0
0000100A
```



## 6.2 DISASSEMBLE

指定されたメモリ位置の内容を逆アセンブル表示します。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

\*: DSU3 評価用チップでは、プログラム実行中に使用できません。

### ■ 形式

DISASSEMBLE [ { 開始アドレス | アドレス範囲 } ]

#### ● パラメータ

開始アドレス ( アドレス式 )

逆アセンブルするメモリの先頭位置を指定します。

アドレス範囲 ( アドレス式 )

逆アセンブルするメモリの範囲を指定します。

### ■ 機能

指定されたメモリ位置の内容を逆アセンブル表示します。

表示は出力ウィンドウに行われます。

開始アドレスのみを指定した場合は、16 行分表示します。

パラメータに "." のみを指定した場合は、現在のプログラムカウンタのアドレスから表示します。

パラメータ省略時は、前回最後に表示した行の次の行から 16 行分表示します。

SET SOURCE コマンドの設定がソース行を付加するモードになっていれば、メモリ位置がソース行に対応している場合は、そのソース行も表示します。

### ■ 例

```
>DISASSEMBLE 1000..1002
00001000      9720      RET
00001002      C000      LDI:8      #0, R0
>DISASSEMBLE .
000FF000      1781      ST          RP, @-R15
000FF002      0F07      ENTER      #01C
000FF004      C010      LDI:8      #1, R0
000FF006      7FF0      STB        R0, @(R14, -1)
```



## 第7章

---

# ロード/セーブコマンド

この章では、ロード/セーブコマンドについて説明します。

7.1 LOAD

7.2 SAVE

## 7.1 LOAD

---

指定されたファイルのロードを行います。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

LOAD ファイル名 [, アドレス] [, ファイルオフセット [, バイト数]]

#### ● パラメータ

##### ファイル名

ロードしたいファイルのファイル名を指定します。

デフォルト拡張子は、コマンド修飾子によって異なります。

##### アドレス (アドレス式)

メモリイメージファイルをロードするメモリ位置を指定します。

コマンド修飾子が、"/BINARY" のときのみ有効です。

ほかの修飾子の場合は、指定するとエラーとなります。

##### ファイルオフセット (データ式)

指定されたファイルの読み込み開始データのオフセットを指定します。

省略した場合は、ファイルの先頭からとなります。

コマンド修飾子が、"/BINARY" のときのみ有効です。

ほかの修飾子の場合は、指定するとエラーとなります。

##### バイト数 (データ式)

読み込みデータ数を指定します。

省略した場合は、すべてのデータを読み込みます。

コマンド修飾子が、"/BINARY" のときのみ有効です。

ほかの修飾子の場合は、指定するとエラーとなります。

#### ● コマンド修飾子

##### /OBJECT (省略時のデフォルト)

ロードモジュールファイルをロードします。

デフォルトの拡張子は、".abs" です。

##### /DEBUG

ロードモジュールファイル内のデバッグ情報のみをロードします。

デフォルト拡張子は、".abs" です。

**/BINARY**

バイナリ形式のメモリイメージファイルをロードします。

デフォルト拡張子は、".bin" です。

アドレスの指定は省略できません。

**/COVERAGE**

カバレッジデータファイルをロードします。デフォルト拡張子は ".cov" です。高速版シミュレータデバッガの場合のみ使用可能です。

**/ALIAS**

エイリアスファイル ( コマンド別名定義 , マクロコマンド定義 ) をロードします。

デフォルト拡張子は、".lst" です。

**/AUTOMATIC ( 省略時のデフォルト )**

シミュレータデバッガの場合、ロード時に自動的にマップ設定を行います。

**/MANUAL**

シミュレータデバッガの場合、ロード時に自動的にマップ設定を行わないため、あらかじめ SET MAP コマンドでマップ領域の設定が必要です。

**/READ**

シミュレータデバッガの場合、AUTOMATIC 修飾子が有効な場合、データセグメントの ROM 領域を /READ 属性として設定します。この修飾子が省略されると /READ /CODE 属性が設定されます。

**/ONDEMAND**

ロードモジュールファイルをオンデマンドロードします。この修飾子は "/OBJECT" または "/DEBUG" を指定した時のみ、指定することができます。

## ● データ長

**/BYTE ( 省略時のデフォルト )**

データをバイト長でロードします。

**/HALFWORD**

データを 2 バイト長でロードします。

**/WORD**

データを 4 バイト長でロードします。

**/DWORD**

データを 8 バイト長でロードします。

## ■ 機能

指定されたファイルのロードを行います。次のファイルのロードができます。

## ● ロードモジュールファイル

リンカで作成した絶対形式オブジェクトファイルです。

## ● メモリイメージファイル

SAVE コマンドでセーブしたメモリイメージファイルです。ロードする場合は、必ずアドレスを指定してください。

## 第7章 ロード/セーブコマンド

- カバレッジデータファイル

SAVE コマンドでセーブしたカバレッジデータファイルです。

- エイリアスファイル

コマンド別名, マクロ定義が格納されたファイルです。ファイル名の拡張子を省略すると, それぞれデフォルトの拡張子を付加してファイルをオープンします。

### ■ 例

```
>LOAD debug  
>LOAD/BINARY data.bin, FE0000
```

## 7.2 SAVE

コマンド修飾子を省略するか、または "/BINARY" の指定を行ったときは、指定されたメモリの内容をメモリイメージ (データのみのバイナリ形式) でファイルにセーブします。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SAVE ファイル名 [, { アドレス範囲 | モジュール名 }]

#### ● パラメータ

##### ファイル名

メモリの内容をセーブするファイル名を指定します。

ファイル名の拡張子を省略した場合は、次の拡張子が付加されます。

- ・メモリイメージでセーブする場合 ".bin"
- ・カバレッジデータをセーブする場合 ".cov"
- ・カバレッジ測定結果を CSV 形式でセーブする場合 ".csv"
- ・コマンド別名、マクロ定義をセーブする場合 ".lst"

##### アドレス範囲 (アドレス式)

セーブしたいメモリ領域を指定します。

コマンド修飾子が /BINARY の場合のみ有効です。

ほかの修飾子の場合は指定するとエラーとなります。

##### モジュール名

セーブしたいカバレッジ測定結果のモジュール名を指定します。

コマンド修飾子が /COVERAGE の場合のみ有効です。

省略した場合はモジュール全体のカバレッジ率をセーブします。

#### ● コマンド修飾子

- ・ファイルの種類

##### /BINARY (省略時のデフォルト)

メモリ内容をバイナリ形式でメモリイメージファイルへセーブします。

デフォルト拡張子は、".bin" です。

アドレス範囲の指定は省略できません。

### /COVERAGE

SET COVERAGE コマンドで指定された全領域のカバレッジデータをセーブします。デフォルト拡張子は、".cov" です。アドレス範囲の指定は無効です。

### /ALIAS

コマンド別名定義、マクロコマンド定義をエイリアスファイルにセーブします。

デフォルト拡張子は、".lst" です。

アドレス範囲の指定は無効です。

## ● 保存形式

### /CSV

CSV 形式でファイルをセーブします。

コマンド修飾子が /COVERAGE の場合のみ有効です。

## ■ 機能

コマンド修飾子を省略するか、または "/BINARY" の指定を行ったときは、指定されたメモリの内容をメモリイメージ（データのためのバイナリ形式）でファイルにセーブします。

この場合、アドレス範囲の指定は省略できません。

"/COVERAGE" を指定した場合は、SET COVERAGE で指定された全領域のカバレッジ測定データをセーブします。

同時に "/CSV" を指定すると、モジュールのカバレッジ測定結果を CSV 形式でセーブします。

"/ALIAS" を指定した場合は、コマンド別名定義、マクロコマンド定義をエイリアスファイルにセーブします。

## ■ 例

```
>SAVE memo.bin, 0..00000fff  
>SAVE /COVERAGE/CSV cov.csv, sample.c
```



## 第8章

---

# ソースファイル/ シンボルコマンド

この章では、ソースファイル/シンボルコマンドについて説明します。

- 8.1 LIST
- 8.2 SET PATH
- 8.3 SHOW PATH
- 8.4 SHOW SCOPE
- 8.5 UP
- 8.6 DOWN

## 8.1 LIST

---

指定した行番号に対応するソース行を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

LIST [ { [ ファイル名 ] 行番号 [ .. 行番号 ] | アドレス } ]

#### ● パラメータ

##### ファイル名

表示したいソースファイル名を指定します。

省略した場合は前回指定されたファイル名となります。

##### 行番号

表示したいソースの行番号を指定します。

行番号の先頭には、必ず \$ を付けてください。

行番号を " .. " で区切った場合は指定範囲のソースを表示します。

##### アドレス ( アドレス式 )

コード属性のメモリ位置を指定します。

メモリ位置に対応するソースを表示させたい場合に指定します。

### ■ 機能

指定した行番号に対応するソース行を表示します。

パラメータに "." のみを指定した場合は、現在のプログラムカウンタに対応するソース行から表示します。

パラメータ省略時は、プログラム実行等によりプログラムカウンタが書き換わった場合は、現在のプログラムカウンタに対応するソース行から表示します。

それ以外は、前回最後に表示した行の次の行から 19 行分表示します。

## ■ 例

```
>LIST PROGRAM.C$2..$3
2:      x = x+1 ;
3:      printf ("%d ¥ n", x) ;
>LIST subdisp
30:      subdisp ( )
31:      {
32:          int i;
33:
34:          for (i = p; i > = 1; i-- )
35:              printf ( "data [%d] = %d ¥ n", i, data [i] );
36:
          .
          .
          .

>LIST.
53:      switch (*s) {
54:          case '0' : z = " "; return (z) ;
55:          case '1' : z = "a"; return (z) ;
56:          case '2' : z = "b"; return (z) ;
```

## 8.2 SET PATH

---

ソースファイルを検索するときに使用するディレクトリを指定します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET PATH [ ソース検索ディレクトリ名 [, ... ] ]

#### ● パラメータ

ソース検索ディレクトリ名

ソースファイルを検索するディレクトリを指定します。

#### ● コマンド修飾子

/APPEND

指定した検索ディレクトリ指定を現在の設定に追加します。

### ■ 機能

ソースファイルを検索するときに使用するディレクトリを指定します。

ソースファイルは指定された検索ディレクトリ順 ( 左から順番に ) で検索されます。

パラメータ省略時は, ソース検索ディレクトリ名が削除されます ( 検索ディレクトリをカレントディレクトリとします ) 。

### ■ 例

```
>SET PATH A:¥
```

# 8.3 SHOW PATH

現在有効なソースファイル検索ディレクトリを表示します。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

SHOW PATH

■ 機能

現在有効なソースファイル検索ディレクトリを表示します。

■ 例

```
>SHOW PATH
source file search path = a:¥
```

## 8.4 SHOW SCOPE

---

現在のプログラムカウンタが指すメモリ位置が含まれるモジュール名および関数名を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW SCOPE

### ■ 機能

現在のプログラムカウンタが指すメモリ位置が含まれるモジュール名および関数名を表示します。

### ■ 例

```
>SHOW SCOPE  
current scope = SIEVE ¥sub_main¥
```

# 8.5 UP

親関数ヘスコープを移します。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

UP

■ 機能

親関数ヘスコープを移します。  
UP/DOWN の情報は ,MCU 実行を行ったとき ,RESET を行ったとき ,プログラムカウンタを変更したときにはクリアされます。

■ 例

```
>UP
Current Scope = demo¥sort¥
```

## 8.6 DOWN

---

子関数ヘスコープを移します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

DOWN

### ■ 機能

子関数ヘスコープを移します。

UP/DOWN の情報は ,MCU 実行を行ったとき ,RESET を行ったとき ,プログラムカウンタを変更したときにはクリアされます。

### ■ 例

```
>DOWN
Current Scope = demo¥check¥
```



## 第9章

---

# コマンドプロシージャ コマンド

この章では、コマンドプロシージャコマンドについて説明します。

9.1 BATCH

9.2 QUIT

## 9.1 BATCH

指定したコマンドプロシージャファイルに含まれているコマンドを実行します。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

BATCH ファイル名 [, 実パラメータ [, ... ]]

#### ● パラメータ

ファイル名

実行させたいコマンドプロシージャを記述してあるファイル名を指定します。

デフォルト拡張子は, ".prc" です。

実パラメータ

コマンドプロシージャに必要な実パラメータを指定します。

#### ● コマンド修飾子

/ICON

コマンドプロシージャ実行時にデバッガをアイコン化して実行します。

コマンドプロシージャが終了すると元のサイズに戻ります。

### ■ 機能

指定したコマンドプロシージャファイルに含まれているコマンドを実行します。

バッチ処理のネスト (プロシージャファイルコールのネスト) は8レベルまで可能です。

実パラメータは, 指定された順にコマンドプロシージャファイル中の仮パラメータ "%P0 ~ %P9" と置き換えられます。

仮パラメータの数が指定された実パラメータの数よりも多い場合は, 残りの仮パラメータは空文字列 (ヌル) に置き換えられます。

仮パラメータの数が指定された実パラメータの数よりも少ない場合は, 残りのパラメータは無視されます。

また, 指定された実パラメータの数は, "%NP" で参照できます。

### ■ 例

```
>BATCH TST.PRC, 0, 0FFF, BRK
```

## 9.2 QUIT

コマンドプロシージャ実行中に実行された場合は、コマンドプロシージャ処理を終了します。制御コマンドの中に書かれた場合は、制御コマンドのすべてのループを抜け出します。コマンド待ちで実行された場合は、何も実行されません。

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

QUIT

### ■ 機能

コマンドプロシージャ実行中に実行された場合は、コマンドプロシージャ処理を終了します。

制御コマンドの中に書かれた場合は、制御コマンドのすべてのループを抜け出します。

コマンド待ちで実行された場合は、何も実行されません。

### ■ 例

コマンドプロシージャファイルの内容

```
IF %NP < 2
    QUIT
ENDIF
SET VARIABLE I = 0
SET VARIABLE ADDR = %P0
WHILE %I<%P1
    SET MEMORY %ADDR = %I
    SET VARIABLE I = %I+1
    IF %ADDR == H'FFFFFF
        QUIT
    ELSE
        SET VARIABLE ADDR = %ADDR+1
    ENDIF
ENDW
```



# 第10章

---

## 置き換えコマンド

この章では、置き換えコマンドについて説明します。

10.1 SET ALIAS

10.2 SHOW ALIAS

10.3 CANCEL ALIAS

10.4 SET VARIABLE

10.5 SHOW VARIABLE

10.6 CANCEL VARIABLE

## 10.1 SET ALIAS

---

コマンドの別名 ( コマンドエイリアス ) を定義します。

---

### ■ 対応デバッグ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET ALIAS 別名 = コマンド文字列

#### ● パラメータ

別名 ( 識別子 )

コマンドの別名を指定します。

コマンド文字列

別名で置き換えたいコマンド文字列 ( コマンド名 , コマンド修飾子およびパラメータ ) をダブルクォーテーション ( " ) で囲んで指定します。

### ■ 機能

コマンドの別名 ( コマンドエイリアス ) を定義します。

よく使うコマンドはコマンド別名を定義しておくとう便利です。

コマンド別名のネストはできません。

コマンド別名定義の中に別のコマンド別名を含むことはできません。

### ■ 例

```
>SET ALIAS BP = "SET BREAK 00FF0300 , 3"  
>SET ALIAS E = "ENTER"  
>SET ALIAS R = "SHOW REGISTER"
```

# 10.2 SHOW ALIAS

定義されているコマンド別名リストを表示します。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

SHOW ALIAS

■ 機能

定義されているコマンド別名リストを表示します。

■ 例

```
>SHOW ALIAS
T   : STEP
D   : EXAMINE
PC  : SHOW REGISTER PC
>
```

## 10.3 CANCEL ALIAS

---

指定したコマンド文字列の別名 ( コマンドエイリアス ) を解除します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CANCEL ALIAS [ 別名 [ , ... ] ]

#### ● パラメータ

別名 ( 識別子 )

解除したいコマンド別名を指定します。

#### ● コマンド修飾子

/ALL

すべてのコマンド文字列の別名を解除します。

### ■ 機能

指定したコマンド文字列の別名 ( コマンドエイリアス ) を解除します。

### ■ 例

```
>CANCEL ALIAS BP  
>
```



# 10.4 SET VARIABLE

デバッグ変数を定義します。

## ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

## ■ 形式

SET VARIABLE デバッグ変数名 = 置き換え文字列

### ● パラメータ

デバッグ変数名 ( 識別子 )

定義したいデバッグ変数を指定します。

置き換え文字列

デバッグ変数と置き換える文字列を指定します。

## ■ 機能

デバッグ変数を定義します。

定義されたデバッグ変数は、コマンド指定時のパラメータ部の一部として使用できません。

使用されたデバッグ変数は、このコマンドで定義された置き換え文字列にそのまま置き換えられます。

デバッグ変数には、パラメータ部に指定できるものはすべて定義できます。

例えば、文字列を定義する場合には文字列を、式を定義する場合には式をそのまま定義できます。

デバッグ変数は定義されたそのままの形で置き換えを行いますので、同一のデバッグ変数を用いた定義 ( 例えばインクリメント等 ) では注意が必要です。

## ■ 例

```
>SET VARIABLE ADDR = 0309+12
>SET VARIABLE STR = "ABCDEF"
>SET MEMORY/STRING %ADDR = %STR
  (SET MEMORY/STRING 0309+12 ="ABCDE" に置き換えられます。)
>SET VARIABLE CNT = 1
>WHILE %CNT<5
* PRINTF "val [ %d ] = %d \n", %CNT, %CNT
* SET VARIABLE CNT = %EVAL( %CNT+1)
  ( %EVAL 関数で CNT の設定文字列が限度を越えないようにしています。)
* ENDW
```

## 10.5 SHOW VARIABLE

---

指定されたデバッグ変数の定義を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW VARIABLE [ デバッグ変数名 [, ... ] ]

#### ● パラメータ

デバッグ変数名 ( 識別子 )

表示したいデバッグ変数名を指定します。

#### ● コマンド修飾子

/ALL ( 省略時のデフォルト )

すべてのデバッグ変数を表示します。

### ■ 機能

指定されたデバッグ変数の定義を表示します。

### ■ 例

```
>SET VARIABLE CNT = 1
>WHILE %CNT<5
*SHOW VARIABLE CNT
*PRINTF "CNT = %d\n", %CNT
*SET VARIABLE CNT = %CNT+1
*ENDW
CNT : 1
CNT = 1
CNT : 1+1
CNT = 2
CNT : 1+1+1
CNT = 3
CNT : 1+1+1+1
CNT = 4
>
```

# 10.6 CANCEL VARIABLE

指定されたデバッグ変数を解除します。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

CANCEL VARIABLE [ デバッグ変数名 [ , ... ] ]

● パラメータ

デバッグ変数名 ( 識別子 )

解除したいデバッグ変数名を指定します。

● コマンド修飾子

/ALL

すべてのデバッグ変数を解除します。

■ 機能

指定されたデバッグ変数を解除します。

■ 例

```
>CANCEL VARIABLE CHKADR, X, Y
>
```



# 第11章

---

## ユーティリティコマンド

この章では、ユーティリティコマンドについて説明します。

11.1 SET LOGGING

11.2 SHOW LOGGING

11.3 CANCEL LOGGING

11.4 ENABLE LOGGING

11.5 DISABLE LOGGING

11.6 PRINTF

11.7 SET OUTPUT

11.8 SHOW OUTPUT

## 11.1 SET LOGGING

---

ロギングファイルをオープンし、ロギングを開始します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET LOGGING [ ファイル名 ]

#### ● パラメータ

ファイル名

ログファイルのファイル名を指定します。

デフォルト拡張子は、".LOG" です。

省略した場合は、DEBUG.LOG というファイルでロギングします。

#### ● コマンド修飾子

/OPEN ( 省略時のデフォルト )

指定されたファイルを新規にオープンします。

/APPEND

指定されたファイルの最後からログの内容を追加するようにします。

/EXPANSION ( 省略時のデフォルト )

コマンドリストとその結果をロギングします。

/UNEXPANSION

結果のみをロギングします。

/COMMAND

ユーザ入力のみをロギングします。

## ■ 機能

ロギングファイルをオープンし、ロギングを開始します。

コマンド修飾子 "/APPEND" を指定した場合は、前のファイルの内容は失われません。

ロギングするデータを選択することができます。

コマンド修飾子に "/COMMAND" を指定するとキー入力したもののみがロギングされるため、そのファイルをそのまま、コマンドプロシージャファイルとして用いることができます。

## ■ 例

```
>SET LOGGING filename.log  
>  
>SET LOGGING/COMMAND filename.log
```

## 11.2 SHOW LOGGING

---

ロギングの状態を表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW LOGGING

### ■ 機能

ロギングの状態を表示します。

### ■ 例

```
>SHOW LOGGING  
en/dis           : ENABLE  
logging file     : logfile.log  
logging data     : EXPANSION
```



## 11.3 CANCEL LOGGING

---

ロギング設定を解除し, ロギングファイルをクローズします。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

CANCEL LOGGING

### ■ 機能

ロギング設定を解除し, ロギングファイルをクローズします。

### ■ 例

```
>CANCEL LOGGING
```

## 11.4 ENABLE LOGGING

---

ロギングを有効にします。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

ENABLE LOGGING

### ■ 機能

ロギングを有効にします。

### ■ 例

```
>ENABLE LOGGING
```

# 11.5     DISABLE LOGGING

ロギングを無効にします。

■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

■ 形式

DISABLE LOGGING

■ 機能

ロギングを無効にします。  
ENABLE LOGGING コマンドで再度有効にすることができます。

■ 例

>DISABLE LOGGING

## 11.6 PRINTF

---

指定された文字列や、指定された書式での式の値を画面上に表示します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

PRINTF " 書式制御文字列 " [, 式 [, ... ]]

#### ● パラメータ

書式制御文字列

画面上に表示する文字列と、式の値を表示するための書式を指定します。

書式指定は、ダブルクォーテーション ( " ) で囲みます。

"% [flag] [width] [.precision] [l] type"

%

書式指定を行って表示したい場合に指定します。

% 記号の後ろに書式指定でない文字が続く場合には、その文字を表示します。

flag

表示を右詰めにするか、左詰めにするか、または 8 進数、16 進数を表す 0、0x、0X の出力制御を指定します。

省略した場合は右詰めで表示します。

変換表示形式 (type) が、b または f の場合は無効です。

- : 左詰め

# : 変換表示形式が、o、x、X の場合に数値の前にそれぞれ 0、0x、0X を付けて表示します。

width

出力する整数の最小桁数を指定します。

変換結果が指定した桁数より小さい場合には、その桁数になるように空白が補われます。

右詰め表示で 0 を補いたい場合には、先頭に 0 をつけて桁数を指定します。

変換表示形式 (type) が、b または f の場合は無効です。

## precision

出力する整数の最小桁数を指定します。

変換結果が指定した桁数より小さい場合には、その桁数になるように 0 が補われます。

変換表示形式 (type) が , b または f の場合は無効です。

|

変換表示形式 (type) が , d, u, o, x の場合に言語式の値を long, unsigned long 型とみなして表示するかを指定します。

l 指定しなかった場合は、式の値を short, unsigned short 型とみなします。

## type

式の値の変換表示形式の指示をします。

d : 符号付きの 10 進数

u : 符号なしの 10 進数

o : 符号なしの 8 進数

x : 符号なしの 16 進数 (10 ~ 15 を abcdef と小文字で表示します。)

X : 符号なしの 16 進数 (10 ~ 15 を ABCDEF と大文字で表示します。)

c : 1 文字

b : 符号なしの 2 進数

s : 文字列 (アドレス指定のみ有効です。最大文字数は 128 バイトまでです。)

## 式

表示したい式を記述します。

## ■ 機能

指定された文字列や、指定された書式での式の値を画面上に表示します。

## ■ 例

```
>PRINTF "ABC = %d¥n", datflg
ABC = 3
```

## 11.7 SET OUTPUT

---

ユーザプログラムが停止した場合，PC が示す位置のデバッグ情報に従ってソースウィンドウを開きにいきます。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SET OUTPUT

#### ● コマンド修飾子

/SOURCE (省略時のデフォルト)

ソースウィンドウが開かれるとき，ファイルが存在しない場合でもソースウィンドウを混在モードで開きます。

/INSTRUCTION

ソースウィンドウが開かれるとき，ファイルが存在しない場合，逆アセンブルウィンドウを開きます。

### ■ 機能

ユーザプログラムが停止した場合，PC が示す位置のデバッグ情報に従ってソースウィンドウを開きにいきます。

このとき，目的のソースファイルが見つからない場合の動作を指定します。

### ■ 例

```
>SET OUTPUT /SOURCE
```

## 11.8 SHOW OUTPUT

---

SET OUTPUT コマンドで設定した表示モードを表示します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

SHOW OUTPUT

### ■ 機能

SET OUTPUT コマンドで設定した表示モードを表示します。

### ■ 例

```
>SHOW OUTPUT  
source mode : source
```





# 第12章

---

## 制御コマンド

この章では、制御コマンドについて説明します。

12.1 IF

12.2 REPEAT

12.3 WHILE

12.4 BREAK

## 12.1 IF

---

式が真と評価される場合は、IF の直後のコマンドリストを実行し、偽と評価される場合は、ELSE の次に指定したコマンドリストを実行します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

IF 式  
コマンドリスト  
[ELSEIF 式  
コマンドリスト]  
[ELSE  
コマンドリスト]  
ENDIF

#### ● パラメータ

式  
指定されたコマンドリストの実行条件式を指定します。  
コマンドリスト  
実行させたいコマンドを指定します。

### ■ 機能

式が真と評価される場合は、IF の直後のコマンドリストを実行し、偽と評価される場合は、ELSE の次に指定したコマンドリストを実行します。

ELSE を省略した場合は、式が偽なら何も実行しません。

制御コマンドは、マクロまたはバッチからのみ使用可能です。

### ■ 例

```
IF %R0 == 0
    print "OK!!"
else
    print "NG!!"
endif
```

## 12.2 REPEAT

---

コマンドリストで指定したコマンドリストを実行後, UNTIL の式を評価します。偽と評価される間, コマンドリストの実行を繰り返します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

```
REPEAT  
  コマンドリスト  
UNTIL 式
```

#### ● パラメータ

コマンドリスト  
実行させたいコマンドを指定します。

式  
指定されたコマンドリストの実行条件式を指定します。

### ■ 機能

コマンドリストで指定したコマンドリストを実行後, UNTIL の式を評価します。偽と評価される間, コマンドリストの実行を繰り返します。

制御コマンドは, マクロまたはバッチからのみ使用可能です。

### ■ 例

```
REPEAT  
  STEP  
UNTIL %PC == main
```

## 12.3 WHILE

---

指定した式が真と評価される間, コマンドリストの実行を繰り返します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

WHILE 式  
コマンドリスト  
ENDW

#### ● パラメータ

式  
指定されたコマンドリストの実行条件式を指定します。  
コマンドリスト  
実行させたいコマンドを指定します。

### ■ 機能

指定した式が真と評価される間, コマンドリストの実行を繰り返します。  
制御コマンドは, マクロまたはバッチからのみ使用可能です。

### ■ 例

```
WHILE %PC != function  
    STEP  
ENDW
```

## 12.4 BREAK

---

制御構造を脱出します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

BREAK

### ■ 機能

制御構造を脱出します。

REPEAT コマンドおよび WHILE コマンドのコマンドリスト内にのみ有効です。

制御コマンドは、マクロまたはバッチからのみ使用可能です。

### ■ 例

```
WHILE 1
  if %PC == main
    BREAK
  ENDIF
STEP
ENDW
```



# 第13章

---

## 組込み変数 / 関数

この章では、組込み変数 / 関数について説明します。

- 13.1 %CALL
- 13.2 %ERRNUM
- 13.3 %ENTRY
- 13.4 %STKTOP
- 13.5 %RADIX
- 13.6 %SCPADR
- 13.7 %LOADNUM
- 13.8 %BIT, %B, %H, %W, %S, %D
- 13.9 %STRGET
- 13.10 %STRSTR
- 13.11 %STRCMP
- 13.12 %STRLEN
- 13.13 %STRCAT
- 13.14 %SYMLen
- 13.15 %TOVAL
- 13.16 %TOSTR
- 13.17 %EVAL
- 13.18 %EVENTNUM
- 13.19 %TRIGGERNUM
- 13.20 %DEBUGFUNCTION

## 13.1 %CALL

---

最後に実行した CALL コマンドの復帰値に置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%CALL

### ■ 機能

最後に実行したCALLコマンドの復帰値に置き換わります。関数の復帰値がvoid, double  
の場合は 0 を返します。

### ■ 例

```
>CALL func(100,200)
    return value is H'40
>ENTER val=%CALL + 0x80
```



## 13.2 %ERRNUM

---

コマンドウィンドウから実行した最後のエラー番号に置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%ERRNUM

### ■ 機能

コマンドウィンドウから実行した最後のエラー番号に置き換わります。  
0 の場合はエラーがないことを表します。

### ■ 例

```
>PRINTF "ERROR NO. = %d¥n", %ERRNUM  
ERROR NO. = 5
```

## 13.3 %ENTRY

---

ロードされているロードモジュールの実行開始アドレスに置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%ENTRY

### ■ 機能

ロードされているロードモジュールの実行開始アドレスに置き換わります。  
実行開始のエントリがない場合は , 0 になります。

### ■ 例

```
>PRINTF "ENTRY = 0x%X¥n", %ENTRY  
ENTRY = 0x10000
```

## 13.4 %STKTOP

---

ロードされているロードモジュールのスタック領域の先頭アドレスに置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STKTOP

### ■ 機能

ロードされているロードモジュールのスタック領域の先頭アドレスに置き換わります。

スタック領域がない場合は, 0 になります。

### ■ 例

```
>PRINTF "STACK = 0x%X\n", %STKTOP  
STACK = 0x80000
```

## 13.5 %RADIX

---

現在設定されている基数 ("BINARY", "OCTAL", "DECIMAL", "HEXADECIMAL") に置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%RADIX

### ■ 機能

現在設定されている基数 ("BINARY", "OCTAL", "DECIMAL", "HEXADECIMAL") に置き換わります。

### ■ 例

```
>PRINTF " 基数 = "  
>PRINTF %TOSTR(%RADIX)  
基数 = HEXADECIMAL
```

## 13.6 %SCPADR

---

現在のスコープアドレスに置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%SCPADR

### ■ 機能

現在のスコープアドレスに置き換わります。

### ■ 例

```
>PRINTF " スコープ = 0x%X¥n", %SCPADR
スコープ = 0x18300
```

## 13.7 %LOADNUM

---

最後にロードしたバイナリファイルのサイズに置き換わります。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%LOADNUM

### ■ 機能

最後にロードしたバイナリファイルのサイズに置き換わります。

### ■ 例

```
>PRINTF " バイト数 = %d¥n",    %LOADNUM  
バイト数 = 584
```

## 13.8 %BIT, %B, %H, %W, %S, %D

指定アドレスのメモリを読み出し、その内容に置き換わります。

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198) *
Monitor	

\*: DSU3 評価用チップでは、プログラム実行中に使用できません。

### ■ 形式

%BIT ( アドレス )

%B ( アドレス )

%H ( アドレス )

%W ( アドレス )

%S ( アドレス )

%D ( アドレス )

### ● パラメータ

アドレス

メモリ内容を読み出すアドレスを指定します。

### ■ 機能

指定アドレスのメモリを次のサイズで読み出し、その内容に置き換わります。

%BIT    ビットデータ

%B      バイトデータ

%H      ハーフワードデータ

%W      ワードデータ

%S      単精度浮動小数点数データ

%D      倍精度浮動小数点数データ

### ■ 例

```
>PRINTF "10000 = 0x%X¥n",    %W (10000)
10000 = 0xAABBAACC
```

## 13.9 %STRGET

---

文字列中の指定位置から指定文字数分の文字列に置き換わります。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STRGET ( 文字列 , 文字位置 , 文字数 )

#### ● パラメータ

文字列

対象となる文字列を指定します。

文字位置

切り出す文字位置を指定します ( 先頭からの何文字目か ) 。

文字数

切り出す文字数を指定します。

### ■ 機能

文字列中の指定位置から指定文字数分の文字列に置き換わります。

### ■ 例

```
>PRINTF %TOSTR (%STRGET ("abcdefghijklmn", 3, 4) )
cdef
```



## 13.10 %STRSTR

---

文字列 1 の中に文字列 2 が含まれているかを検索します。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STRSTR ( 文字列 1, 文字列 2)

#### ● パラメータ

文字列 1

対象となる文字列を指定します。

文字列 2

検索する文字列を指定します。

### ■ 機能

文字列 1 の中に文字列 2 が含まれているかを検索します。

含まれている場合は、文字列 1 の文字位置に置き換わります。

含まれていない場合は、0 となります。

### ■ 例

```
>PRINTF "%d¥n", %STRSTR ("abcdefghijklmn", "fg")
```

6

## 13.11 %STRCMP

---

文字列 1 と文字列 2 を比較します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STRCMP ( 文字列 1, 文字列 2)

#### ● パラメータ

文字列 1, 文字列 2

比較する文字列を指定します。

### ■ 機能

文字列 1 と文字列 2 を比較します。

一致している場合は , 0 となり , 一致していない場合は , 1 となります。

### ■ 例

```
>PRINTF "%d¥n", %STRCMP ("abcde", "fg")
1
>PRINTF "%d¥n", %STRCMP ("abcde", "abcde")
0
```

## 13.12 %STRLEN

---

文字列の文字数に置き換えます。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STRLEN ( 文字列 )

#### ● パラメータ

文字列

対象となる文字列を指定します。

### ■ 機能

文字列の文字数に置き換えます。

### ■ 例

```
>PRINTF "%d¥n",  %STRLEN ("abcde")  
5
```

## 13.13 %STRCAT

---

文字列 1 と文字列 2 を結合し, その文字列に置き換えます。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%STRCAT ( 文字列 1, 文字列 2)

#### ● パラメータ

文字列 1, 文字列 2

結合する文字列を指定します。

### ■ 機能

文字列 1 と文字列 2 を結合し, その文字列に置き換えます。

### ■ 例

```
>PRINTF %TOSTR (%STRCAT ("abcde", "fg") )  
abcdefg
```

## 13.14 %SYMLEN

---

指定されたシンボルの大きさを返します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%SYMLEN ( シンボル名 )

#### ● パラメータ

シンボル名

シンボルを指定します。

### ■ 機能

指定されたシンボルの大きさを返します。

### ■ 例

```
>PRINTF "%d¥n", %SYMLEN ("abcde")
```

```
2
```

## 13.15 %TOVAL

---

指定された文字列の両端にあるダブルクォーテーション ( " ) を削除します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%TOVAL ( 文字列 )

#### ● パラメータ

文字列

文字列を指定します。

### ■ 機能

指定された文字列の両端にあるダブルクォーテーション ( " ) を削除します。

文字列以外のパラメータしか記述できない部分に , ダブルクォーテーション内の文字列を指定する場合に使用します。

### ■ 例

```
>SET BREAK %TOVAL ("main")
```

## 13.16 %TOSTR

---

指定された文字列をダブルクォーテーション ( " ) で括ります。

---

### ■ 対応デバugga

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%TOSTR ( 文字列 )

#### ● パラメータ

文字列

パラメータに指定できるすべてのものが指定できます。

### ■ 機能

指定された文字列をダブルクォーテーション ( " ) で括ります。

文字列のみのパラメータしか記述できない部分に , 指定した文字列を指定する場合に使用します。

### ■ 例

```
>PRINTF %TOSTR (main)
main
```

## 13.17 %EVAL

---

指定された式を評価します。

---

### ■ 対応デバッガ

Simulator	
Emulator	(MB2197) (MB2198)
Monitor	

### ■ 形式

%EVAL ( 式 )

#### ● パラメータ

式

評価する式を指定します。

### ■ 機能

指定された式を評価します。

### ■ 例

```
>PRINTF "%d¥n", %EVAL (10+20+30)
60
```



## 13.18 %EVENTNUM

---

指定したアドレスに設定したイベントの番号を取得します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

%EVENTNUM ( アドレス )

#### ● パラメータ

アドレス

自動的に設定されたイベント番号を読み出すアドレスを指定します。

### ■ 機能

指定したアドレスに設定したイベントの番号を取得します。

イベントが設定されていないアドレスを指定した場合, 0 を返します。

### ■ 例

```
>PRINTF "event no : %d¥n",  %EVENTNUM (F0000)
event no : 1
```

## 13.19 %TRIGGERNUM

---

指定したアドレスに設定したトレーストリガの番号を取得します。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

% TRIGGERNUM ( アドレス )

#### ● パラメータ

アドレス

自動的に設定されたトレーストリガ番号を読み出すアドレスを指定します。

### ■ 機能

指定したアドレスに設定したトレーストリガの番号を取得します。

トレーストリガが設定されていないアドレスを指定した場合、0 を返します。

### ■ 例

```
>PRINTF "trace trigger no : %d\n", %TRIGGERNUM (F0000)
trace trigger no : 2
```

## 13.20 %DEBUGFUNCTION

---

現在のデバッグ機能番号に置き換わります。

---

### ■ 対応デバッガ

Simulator		×
Emulator	(MB2197) (MB2198)	×
Monitor		×

### ■ 形式

%DEBUGFUNCTION

### ■ 機能

現在のデバッグ機能番号に置き換わります。

### ■ 例

```
>PRINTF "debug function : %d¥n", %DEBUGFUNCTION
debug function : 0
```



**マネージャ関係およびデバッガ関係の、メッセージを掲載します。**

付録 A マネージャ関係のエラーメッセージ

付録 B デバッガ関係のエラーメッセージ

付録 C デバッガ中断メッセージ

## 付録 A マネージャ関係のエラーメッセージ

マネージャ関係のエラーメッセージを以下に示します。

### ■ マネージャ関係のエラーメッセージ

E4002W	メモリ不足です。
	Insufficient memory.

【内容】システムのメモリが足りなくなりました。

【対処】他のプログラムを終了してから実行してください。

E4011W	レジストリに登録できませんでした。
	Registration not possible.

【内容】システムのレジストリに書き込めませんでした。

【対処】他のプログラムを終了してから実行してください。

E4012W	関数の呼び出しに失敗しました。EXE ファイルのバージョンが古い可能性があります。
	Function call failed. Exefile is old.

【内容】プログラムファイルと DLL ファイルのバージョンが合っていません。

【対処】SOFTUNE Workbench の最新版を再インストールしてください。

E4013W	関数の呼び出しに失敗しました。DLL のバージョンが古い可能性があります。
	Failed function call. DLLfile is old.

【内容】プログラムファイルと DLL ファイルのバージョンが合っていません。

【対処】SOFTUNE Workbench の最新版を再インストールしてください。

E4020W	CPU 情報ファイルのバージョンが古いです。
	CPU information file version is different. Contains uninterpretable information.

【内容】CPU 情報ファイルが古いため、必要な情報がありません。

【対処】最新の CPU 情報ファイルを入手してください。

E4021W	CPU 情報ファイルに記述されているチップ種別は対象ではありません。
	Chip type in CPU information file is not applicable.

【内容】違う種類の CPU 情報ファイルが指定されました。

【対処】正しい CPU 情報ファイルを指定してください。

E4022W	CPU 情報ファイルを指定してください。
	Please enter CPU infomation file.

【内容】CPU 情報ファイルが見つかりません。

【対処】CPU 情報ファイルのディレクトリを入力してください。

E4023W	ツールオプションデータが不正です。ツールのオプションはデフォルト値が再設定されました。
	Illegal tool option data. Default data is set.

【内容】プロジェクトファイル内のツールオプションに不正な値が書き込まれています。

【対処】ツールオプションを再設定してください。

E4024W	プロジェクトファイルに設定されている CPU 情報に誤りがあります。デフォルト値を設定します。
	Invalid CPU infomation. Set default value.

【内容】CPU 情報ファイル内に不正なデータが書き込まれています。

【対処】最新の CPU 情報ファイルを入手してください。

E4100W	アクセスが拒否されました。
	Access was denied.

【内容】ファイルのアクセスができません。

【対処】ファイルがライト禁止、リード禁止になっている可能性があります。ファイル属性を見直してください。

E4110W	オープンするファイルが多すぎます。
	Too many open files.

【内容】オープン可能な最大ファイル数を超過しています。

【対処】他のファイルをクローズしてください。

E4120W	不正なディレクトリです。
	Directory does not exist.

【内容】ディレクトリが見つかりません。

【対処】正しいディレクトリ名を入力してください。

E4121W	ドライブの準備ができていません。
	Drive is not ready.

【内容】ドライブをアクセスできません。

【対処】ドライブの状態をチェックしてください。

E4122W	ディレクトリパスは無効です。
	Path is invalid.

【内容】ディレクトリが見つかりません。

【対処】正しいディレクトリ名を入力してください。

E4123W	ディレクトリが作成できません。
	Unable to create directory.

【内容】ディレクトリの作成に失敗しました。

【対処】ディレクトリが書き込み禁止か、ディレクトリ名が正しくない場合が考えられます。

E4124W	ディレクトリを削除できません。
	Unable to delete directory.

【内容】ディレクトリの削除に失敗しました。

【対処】ディレクトリが書き込み禁止か、ディレクトリ内のファイルを別のプロセスが使用中であることが考えられます。

E4125W	出力先のディスクがいっぱいです。
	Destination disk is full.

【内容】ディスクの残り容量が不足しています。

【対処】不要なファイルを削除してください。

E4126W	カレントディレクトリは削除できません。
	Could not be removed because it is the current directory.

【内容】カレントディレクトリを削除しようとしています。

【対処】カレントディレクトリを移動してから削除してください。

E4127W	このディレクトリにアクセスできません。
	This directory cannot be access.

【内容】ディレクトリへのアクセスが拒否されました。

【対処】ディレクトリのアクセス権が無いことが考えられます。



E4130W	ファイルをオープンできません。
	File cannot be open.

【内容】ファイルのオープンに失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4131W	ファイルをクローズできません。
	File cannot be close.

【内容】ファイルのクローズに失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4132W	ファイルを読み込めません。
	File cannot be read.

【内容】ファイルのリードに失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4133W	ファイルが書き込めません。
	File cannot be written.

【内容】ファイルのライトに失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4134W	ファイルを作成できません。
	File cannot create.

【内容】ファイルの作成に失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4135W	ファイルを削除できません。
	File cannot be delete.

【内容】ファイルの削除に失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4136W	ファイル名を変更できません。
	File cannot be change name.

【内容】ファイル名の変更に失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4137W	ファイルをコピーできません。
	File cannot be copied.

【内容】ファイルのコピーに失敗しました。

【対処】ファイルまたはディレクトリのアクセス権が無いことが考えられます。

E4138W	ファイルが見つかりません。
	File not found.

【内容】ファイルが見つかりませんでした。

【対処】ファイル名を見直してください。

E4140W	ファイルが存在しません。ファイルを作成しますか？
	File not found. Do you create this file?

【内容】ファイルが見つかりませんでした。

【対処】新しいファイルを作る場合は OK ボタンを押してください。

E4142W	アクセス中に共有違反が発生しました。
	A sharing violation occurred while accessing.

【内容】同じファイルを別のプロセスが使っています。

【対処】ファイルを使っているプログラムを終了してください。まれに、プログラムを終了した後もファイルが使用状態のままになってしまう場合があります。この場合は Windows をリブートしてください。

E4143W	アクセス中にロッキング違反が発生しました。
	A locking violation occurred while accessing.

【内容】同じファイルを別のプロセスが使っています。

【対処】ファイルを使っているプログラムを終了してください。まれに、プログラムを終了した後もファイルが使用状態のままになってしまう場合があります。この場合は Windows をリブートしてください。

E4200W	プロジェクトファイルのフォーマットが正しくありません。
	The project file format is illegal.

【内容】プロジェクトファイルが正しく読めません。

【対処】SOFTUNE Workbench のプロジェクトファイルではないか、プロジェクトファイルが壊れている可能性があります。新しいプロジェクトファイルを作ってください。

E4201W	プロジェクトファイルは MCU タイプが異なるのでオープンすることはできません。
	Project file cannot be opened - CPU type is different.

【内容】異なるファミリの MCU 用のプロジェクトファイルです。

【対処】必要な MCU 用の新しいプロジェクトファイルを作ってください。

E4202W	プロジェクトファイルにセーブできませんでした。
	Unable to save project file.

【内容】プロジェクトファイルの書き込みでエラーが発生しました。

【対処】ディスクの残り領域が足りないか、プロジェクトファイルが書き込み禁止になっているなどの原因が考えられます。

E4204W	プロジェクトファイルの CPU 情報が不正です。デフォルト値を設定します。
	Illegal CPU infomation of project file. Setting default value.

【内容】プロジェクトファイルに記録されている CPU の情報が正しくないため、デフォルトの値に置き換えました。

【対処】そのプロジェクトの CPU に関する設定値を見直してください。

E4205W	プロジェクトファイルに記述されているターゲットファイルディレクトリが存在しません。ディレクトリを作成しますか？
	Target file directory not found. Create a directory?

【内容】プロジェクトファイルの書き込み先ディレクトリがありません。

【対処】OK ボタンを押すとディレクトリを作成します。

E4206W	プロジェクトファイルに記述されているリストファイルディレクトリが存在しません。ディレクトリを作成しますか？
	List file directory not found. Create a directory?

【内容】リストファイルの書き込み先ディレクトリがありません。

【対処】OK ボタンを押すとディレクトリを作成します。

E4207W	プロジェクトファイルに記述されているオブジェクトファイルディレクトリが存在しません。ディレクトリを作成しますか？
	Object file directory not found. Create a directory?

【内容】オブジェクトファイルの書き込み先ディレクトリがありません。

【対処】OK ボタンを押すとディレクトリを作成します。

E4210W	プロジェクト名を指定してください。
	Please specify the project name.

【内容】プロジェクト名が指定されていません。

【対処】プロジェクト名を入力してください。

E4211W	プロジェクトディレクトリを指定してください。
	Please specify the project directory

【内容】プロジェクトのディレクトリが指定されていません。

【対処】ディレクトリを入力してください。

E4212W	ターゲットファイル名を指定してください。
	Please specify the target file name.

【内容】ターゲットファイル名が指定されていません。

【対処】ターゲットファイル名を入力してください。

E4213W	指定できない文字が含まれています。¥/:,;*?"' "<>
	Includes characters that cannot be designated. ¥/:,;*?"' "<>

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4214W	指定できない文字が含まれています。;,*?"' "<>
	Includes characters that cannot be designated. ;,*?"' "<>

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4215W	指定できない文字が含まれています。;,*?"' "<>
	Includes characters that cannot be designated. ;,*?"' "<>

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4220W	ターゲットファイル名を指定してください。
	Please specify the target file name.

【内容】ターゲットファイル名が指定されていません。

【対処】ターゲットファイル名を入力してください。

E4221W	指定のディレクトリが存在しません。ディレクトリを作成しますか？
	Directory not found. Do you create this directory?

【内容】ディレクトリが存在しません。

【対処】ディレクトリ名を入力してください。

E4222W	ディレクトリが作成できませんでした。
	Unable to create directory.

【内容】ディレクトリの作成に失敗しました。

【対処】ファイル書き込みが禁止されている可能性があります。

E4223W	ターゲット MCU が変更されました。CPU 情報をデフォルト値に変更します。
	Changed target MCU. CPU information changed to default value.

【内容】ターゲット MCU を変更すると、それまでに設定した CPU の情報はデフォルト値に戻ります。

【対処】CPU の情報を再設定してください。

E4224W	ターゲット MCU を指定してください。
	Specify target MCU.

【内容】ターゲット MCU が指定されていません。

【対処】ターゲット MCU 名を入力してください。

E4225W	プロジェクトタイプを指定してください。
	Specify project type.

【内容】プロジェクトタイプが指定されていません。

【対処】プロジェクトタイプを指定してください。

E4226W	指定できない文字が含まれています。.;*?"' <>
	Includes characters that cannot be designated. .; * ? " ' < >

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4227W	オブジェクトファイルディレクトリを指定してください。
	Please specify Object File Directory.

【内容】オブジェクトファイルの書き込み先ディレクトリが指定されていません。

【対処】ディレクトリ名を指定してください。

E4228W	リストファイルディレクトリを指定してください。
	Please specify List File Directory.

【内容】リストファイルの書き込み先ディレクトリが指定されていません。

【対処】ディレクトリ名を指定してください。

E4230W	二重指定です。
	Double specification.

【内容】同じ指定が既に使われています。

【対処】別の指定に変更してください。

E4232W	セットアップファイルが登録されていません。自動登録します。
	Setup file is not registered. Registered automatically.

【内容】デバッグを起動するには、セットアップファイルが必要です。セットアップファイルがまだ指定されていない場合は、プロジェクト名と同じ名前でセットアップファイルを作成します。

【対処】自動作成されたセットアップファイルは、[プロジェクト] - [プロジェクトの設定] - [デバッグ] で必要な項目を設定してください。

E4233W	有効なセットアップファイルが登録されていません。自動登録します。
	Available setup file is not registered. Registered automatically.

【内容】デバッグを起動するには、セットアップファイルが必要です。セットアップファイルがまだ指定されていない場合は、プロジェクト名と同じ名前でセットアップファイルを作成します。

【対処】自動作成されたセットアップファイルは、[プロジェクト] - [プロジェクトの設定] - [デバッグ] で必要な項目を設定してください。

E4234W	タイトル名を指定してください。
	Please specify the title.

【内容】タイトル名が指定されていません。

【対処】タイトルを設定してください。

E4240W	既に登録済みのメンバです。
	Already a registered member.

【内容】指定されたファイルは、既にプロジェクトに登録されています。

【対処】ファイル名を見直してください。

E4241W	このファイル名は既に登録されています。
	This file name has already been registered.

【内容】指定されたファイルは、既にプロジェクトに登録されています。

【対処】ファイル名を見直してください。

E4242W	ファイルが見つかりません。登録しますか？
	File not found. Do you registered?

【内容】存在しないファイルをプロジェクトに登録しようとしています。

【対処】ファイル名が正しければ、そのまま登録してください。エディタを起動する際、ファイルを新規作成するかどうかの問い合わせが表示されます。

E4243W	選択したファイルが多すぎます。
	Too many select files.

【内容】選択したファイルの総数が最大値を超えています。

【対処】選択数を減らしてください。

E4301W	ツールを起動するためのコマンドラインの生成ができません。
	Unable to create command line.

【内容】言語ツールを起動するためのオプションファイルの生成に失敗しました。

【対処】プロジェクトディレクトリ下の OPT サブディレクトリのアクセス権やディスク容量を確認してください。

E4302W	起動に失敗しました。
	Failed during start.

【内容】ツールの起動ができませんでした。

【対処】ツール名などが間違っている可能性があります。ツールの設定を見直してください。

E4303W	起動コマンドが長すぎます。
	Command Line too long.

【内容】起動するコマンドラインが長すぎます。(最大 2048 文字)

【対処】オプションパラメータを見直してください。

E4304W	エディタの起動に失敗しました。
	Failed during start editor.

【内容】登録された外部エディタが起動できませんでした。

【対処】エディタの実行ファイル名を見直してください。

E4305W	コンパイラ / アセンブラが起動されています。
	Compiler/Assembler is started.

【内容】ツールの起動中にプロジェクトを閉じようとしています。

【対処】中断ボタンでツールを終了してから，プロジェクトを閉じてください。

E4306W	メイクが起動されています。
	Make function is started.

【内容】ツールの起動中にプロジェクトを閉じようとしています。

【対処】中断ボタンでツールを終了してから，プロジェクトを閉じてください。

E4307W	ビルドが起動されています。
	Build function is started.

【内容】ツールの起動中にプロジェクトを閉じようとしています。

【対処】中断ボタンでツールを終了してから，プロジェクトを閉じてください。

E4308W	依存関係の更新が起動されています。
	Include Dependencies is started.

【内容】ツールの起動中にプロジェクトを閉じようとしています。

【対処】中断ボタンでツールを終了してから，プロジェクトを閉じてください。

E4309W	ツールが起動されています。
	Tool is started.

【内容】ツールの起動中にプロジェクトを閉じようとしています。

【対処】中断ボタンでツールを終了してから，プロジェクトを閉じてください。

E4400W	セットアップファイルは読み取り専用です。セットアップ情報の変更は保存できません。
	Setup file is read only. Setup information is not saved.

【内容】セットアップファイルへの書き込みができませんでした。

【対処】セットアップファイルを書き込み可能な状態にしてください。

E4401W	デバッグ環境ファイルが見つかりません。
	Setup file not found.

【内容】デバッグ環境ファイルが見つかりません。

【対処】インストールが正常に行なわれていない可能性がありますので，インストールをやり直してください。



E4402W	デバッグ環境ファイルのフォーマットが正しくありません。
	The setup file format is illegal.

【内容】デバッグ環境ファイル内に不正なデータが書き込まれています。

【対処】インストールが正常に行なわれていない可能性がありますので、インストールをやり直してください。

E4420W	アドレスの最大値は xxxx です。
	Maximum of address is xxxx.

【内容】アドレスの最大値を超えています。

【対処】アドレス指定を見直してください。

E4421W	開始アドレスが終了アドレスを超えています。
	The start address exceeds the end address.

【内容】指定されたアドレス範囲が間違っています。

【対処】アドレス範囲の指定を見直してください。

E4422W	指定されたアドレス範囲は既に設定されています。
	The designated address is already designated.

【内容】既に登録されているアドレス範囲を指定しました。

【対処】アドレス範囲を見直してください。

E4601W	二重指定です。
	Double specification.

【内容】指定された項目は既に存在します。

【対処】指定内容を見直してください。

E4603W	ツールオプションデータが不正です。
	Illegal tool option data.

【内容】ツールオプションデータに必要なデータがありません。

【対処】ツールオプションの確認ダイアログを開いて [ OK ] ボタンを押すと、入力が必要としているコントロールが表示されますので入力してください。

E4604W	指定されていないコントロールがあります。
	There is no control data.

【内容】指定されていないコントロールがあります。

【対処】データを指定してください。

E4605W	指定できない文字が含まれています。
	Includes characters that cannot be designated.

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4606W	指定できない文字が含まれています。 , ; * ? " ' < >
	Includes characters that cannot be designated. , ; * ? " ' < >

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4607W	指定できない文字が含まれています。
	Includes characters that cannot be designated.

【内容】これらの文字は使用できません。

【対処】名前を変更してください。

E4610W	行数の範囲は 20 ~ 255 です。
	The range of the number of lines is 20-255.

【内容】行数の制限を超えています。

【対処】行数を変更してください。

E4611W	桁数の範囲は 80 ~ 1023 です。
	The range of the number of columns is 80-1023.

【内容】桁数の制限を超えています。

【対処】桁数を変更してください。

E4612W	桁数の範囲は 70 ~ 1023 です。
	The range of the number of columns is 70-1023.

【内容】桁数の制限を超えています。

【対処】桁数を変更してください。

E4613W	タブ数の範囲は 0 ~ 32 です。
	The range of the number of tabs is 0 -32.

【内容】タブ数の制限を超えています。

【対処】タブ数を変更してください。

E4614W	マクロ名を指定してください。
	Please specify the macro name.

【内容】マクロ名が指定されていません。

【対処】マクロ名を指定してください。

E4615W	インクルードパスを指定してください。
	Please specify the include path.

【内容】インクルードパスが指定されていません。

【対処】インクルードパスを指定してください。

E4616W	同名のマクロ名がすでに登録されています。内容を変更しますか？
	Already a registered macro name. Do you change contents?

【内容】既に存在するマクロ名を指定しました。

【対処】変更する場合は OK ボタンを押してください。

E4620W	スタートアドレスを指定してください。
	Please specify the start address.

【内容】スタートアドレスがありません。

【対処】スタートアドレスを指定してください。

E4621W	エンドアドレスを指定してください。
	Please specify the end address.

【内容】エンドアドレスがありません。

【対処】エンドアドレスを指定してください。

E4622W	スタートアドレスがエンドアドレスより大きすぎます。
	The start address is larger than the end address.

【内容】アドレス範囲が間違っています。

【対処】アドレス範囲を見直してください。

E4623W	正しいスタートアドレスを指定してください。
	Please specify a correct start address.

【内容】スタートアドレスが間違っています。

【対処】正しいスタートアドレスを指定してください。

E4624W	正しいエンドアドレスを指定してください。
	Please specify a correct end address.

【内容】エンドアドレスが間違っています。

【対処】正しいエンドアドレスを指定してください。

E4625W	ROM/RAM 領域名を指定してください。
	Please specify the ROM/RAM area name.

【内容】ROM / RAM 領域名が指定されていません。

【対処】ROM / RAM 領域名を指定してください。

E4626W	セクション名を指定してください。
	Please specify the section name.

【内容】セクション名が指定されていません。

【対処】セクション名を指定してください。

E4627W	アドレスの最大値は 0xFFFFFFFF です。
	Maximum of address is 0xFFFFFFFF.

【内容】アドレスの最大値を超えています。

【対処】アドレス指定を見直してください。

E4628W	アドレスの最大値は 0xFFFFF です。
	Maximum of address is 0xFFFFF.

【内容】アドレスの最大値を超えています。

【対処】アドレス指定を見直してください。

E4629W	アドレスの最大値は 0xFFFF です。
	Maximum of address is 0xFFFF.

【内容】アドレスの最大値を超えています。

【対処】アドレス指定を見直してください。

E4630W	バンクを超えたアドレス指定はできません。
	Cannot specify address over bank.

【内容】複数のバンクにまたがったアドレスを指定しています。

【対処】ひとつのバンク内のアドレスを指定してください。

E4631W	シンボル名を指定してください。
	Specify symbol name.

【内容】シンボル名がありません。

【対処】シンボル名を指定してください。

E4632W	設定値を指定してください。
	Specify set value.

【内容】設定値がありません。

【対処】設定値を入力してください。

E4635W	同名のシンボル名がすでに登録されています。内容を変更しますか？
	This symbol name has already been registered. Change the setting?

【内容】既に登録されたシンボル名で登録しようとしています。

【対処】変更する場合は、OK ボタンを押してください。

E4636W	同名の ROM/RAM 領域名がすでに登録されています。内容を変更しますか？
	This ROM/RAM area name has already been registered. Change the setting?

【内容】既に登録されている ROM/RAM 領域名を指定しています。

【対処】変更する場合は、OK ボタンを押してください。

E4637W	同名のセクション名がすでに登録されています。内容を変更しますか？
	This section name has already been registered. Change the setting?

【内容】既に登録されているセクション名を指定しています。

【対処】変更する場合は、OK ボタンを押してください。

E4638W	先頭のセクション名にはアドレス指定が必要です。
	Address must be specified to leader section name.

【内容】先頭のセクションにアドレスが指定されていません。

【対処】アドレスを指定してください。

E4639W	このセクション名は他の ROM/RAM 領域ですでに設定されています。
	This section name has already been specified in another ROM/RAM area.

【内容】既に登録されている ROM/RAM 領域名を指定しています。

【対処】ROM/RAM 領域名を見直してください。

E4640W	正しいアドレスを指定してください。
	Specify exact address.

【内容】アドレスの指定が間違っています。

【対処】正しいアドレスを入力してください。

E4701W	指定のディレクトリが存在しません。設定しますか？
	Specified directory does not exist. Specify?

【内容】存在しないディレクトリが指定されています。

【対処】指定に間違いが無ければ OK ボタンを押してください。

E4702W	複数のディレクトリ記述はできません。
	Cannot specify multiple directories.

【内容】ひとつのディレクトリしか設定できません。

【対処】ひとつのディレクトリだけを設定してください。

E4703W	環境変数の設定値が不正です。
	Illegal Environment Variable.

【内容】設定された値が不正です。

【対処】設定値を見直してください。

E4740W	指定の実行ファイルは存在しませんが、リストに登録しますか？
	This executable file does not exist. Register in the list?

【内容】実行ファイル名に記述されているファイルが見つかりません。

【対処】ファイル名を確認してください。

E4741W	タイトルが指定されていません。
	Title is not specified.

【内容】タイトル名が指定されていません。

【対処】タイトルを設定してください。

E4742W	実行ファイル名が指定されていません。
	Executable file is not specified.

【内容】実行ファイル名が指定されていません。

【対処】実行ファイル名を指定してください。

E4743W	登録数が最大です。これ以上登録できません。
	The registration count is maximum. You cannot register any more.

【内容】これ以上設定できません。

【対処】不要な設定を削除してください。

E4744W	シンタックスが不正です。定義されていないマクロ記述が存在します。
	Syntax error. Illegal macro is specified.

【内容】オプションまたは実行時のディレクトリに定義されていないマクロ記述があります。

【対処】シンタックスを確認してください。

E4745W	タイトルが長すぎます。
	Title is too long.

【内容】タイトルが長すぎます。

【対処】タイトルを短くしてください。

E4746W	実行ファイル名が長すぎます。
	Execute file name is too long.

【内容】実行ファイルが長すぎます。

【対処】ファイル名を短くしてください。

E4747W	オプションが長すぎます。
	Option too long .

【内容】オプション指定が長すぎます。

【対処】オプションを減らしてください。

E4748W	実行時のディレクトリが長すぎます。
	The executing directory too long .

【内容】ディレクトリが長すぎます。

【対処】ディレクトリを短くしてください。

E4749W	指定のディレクトリが存在しません。ディレクトリを作成しますか？
	Directory not found. Create this directory?

【内容】指定されたディレクトリが見つかりませんでした。

【対処】ディレクトリが間違っていない場合は、OK ボタンを押してください。

E4750W	同名のタイトルがすでに登録されています。内容を変更しますか？
	Already a registered title. Do you change contents?

【内容】既に登録されているタイトルが指定されています。

【対処】変更する場合は，OK ボタンを押してください。

E4752W	起動するツールが存在しません。
	Start tool does not exist.

【内容】起動するツールが見つかりません。

【対処】登録されたツール名，ディレクトリ名を見直してください。

E4760W	登録されているエラーシンタックスのフォーマットが変換できませんでした。
	The registered error syntax format cannot be converted.

【内容】アウトプットウィンドウに表示されたエラーメッセージが解析できません。

【対処】[ 環境 ] - [ エラージャンプの設定 ] のシンタックスリストの設定を見直してください。

E4761W	シンタックスが不正です。定義されていないマクロ記述が存在します。
	Syntax error. Undefined Macro.

【内容】定義されていないマクロが指定されています。

【対処】シンタックスを確認してください。

E4762W	シンタックスが不正です。%f，%* の区切りを示す文字が定義されていないため正しく解析できません。
	Syntax error. Undefined separate of ' %f, '%* '.

【内容】%f，%\* の終了を示す区切り文字が入力されていません。

【対処】%f，%\* マクロ記述には，これらの終了を判断するための区切り文字が必要です。マクロ記述の次の文字を区切り記号として認識しています。

E4763W	シンタックスが不正です。マクロ記述が重複しています。
	Syntax error. Duplicate Macro syntax.

【内容】%f，%l，%h のマクロが重複して記述されています。

【対処】シンタックスを見直してください。

E4764W	シンタックスが不正です。¥記述が不正です。
	Syntax error. Invalid ¥syntax .

【内容】¥記述が不正です。¥t，¥l，¥¥ など以外で¥が使われています。

【対処】シンタックスを見直してください。



E4765W	シンタックスが不正です。%[] の記述が不正です。
	Syntax error. Invalid '%[]' syntax.

【内容】%[] マクロ記述が不正です。

【対処】[] の対応がとれていない可能性があります。シンタックスを見直してください。

E4766W	シンタックスが不正です。%f が記述されていません。
	Syntax error. Don't describe '%f'.

【内容】%f または %h マクロが記述されていません。

【対処】エラージャンプの設定のシンタックスには必ず %f または %h を指定してください。

E4767W	シンタックスが不正です。%[...] の間に不正なマクロ記述が存在します。
	Syntax error. Invalid Macro into '%[...]'.

【内容】%[] マクロの間に不正なマクロ記述が存在しています。

【対処】%[] マクロの中に記述できるマクロは %% または %[] のみです。

E4768W	同じシンタックスがすでに存在しています。内容を変更しますか？
	Already a registered syntax. Do you change contents?

【内容】既に同じシンタックスが登録されています。

【対処】内容を変更する場合は OK ボタンを押してください。

E4769W	シンタックスが指定されていません。
	Syntax not specified.

【内容】シンタックスがありません。

【対処】シンタックスを指定してください。

E4771W	シンタックス文字列が長すぎます。
	Syntax too long.

【内容】シンタックス文字列が長すぎます。

【対処】シンタックスを短くしてください。

E4772W	コメントが長すぎます。
	Comment too long.

【内容】コメントが長すぎます。

【対処】コメントを短くしてください。

E4773W	登録数が最大です。これ以上登録できません。
	The registration count is maximum. You cannot register any more.

【内容】登録数が最大値を超えました。

【対処】不要な設定を見直してください。

E4774W	同じシンタックスが SYSTEM に設定されています。変更できません。
	The same syntax has already been set in the SYSTEM. It cannot be changed.

【内容】同じ内容が既に SYSTEM として設定されています。

【対処】SYSTEM として登録されている内容は変更できません。

E4780W	タイトルが指定されていません。
	Title not specified.

【内容】タイトル名がありません。

【対処】タイトルを指定してください。

E4781W	実行ファイル名が指定されていません。
	Execute filename not specified.

【内容】実行ファイル名が指定されていません。

【対処】実行ファイル名を指定してください。

E4782W	オプションが指定されていません。
	Option not specified.

【内容】オプションが指定されていません。

【対処】オプションを指定してください。

E4783W	同名のタイトルがすでに存在しています。内容を変更しますか？
	Already a registered title. Do you change contents?

【内容】既に登録されているタイトルが指定されています。

【対処】変更する場合は、OK ボタンを押してください。

E4784W	シンタックスが不正です。定義されていないマクロ記述が存在します。
	Syntax error. Undefined Macro.

【内容】定義されていないマクロが指定されています。

【対処】シンタックスを確認してください。

E4785W	シンタックスが不正です。マクロ記述が重複しています。
	Syntax error. Duplicate Macro syntax.

【内容】 %f , %l , %h のマクロが重複して記述されています。

【対処】 シンタックスを見直してください。

E4786W	シンタックスが不正です。 %f が記述されていません。
	Syntax error. Don't describe '%f'.

【内容】 %f または %h マクロが記述されていません。

【対処】 エラージャンプの設定のシンタックスには必ず %f または %h を指定してください。

E4789W	登録数が最大です。これ以上登録できません。
	The registration count is maximum. You cannot register any more.

【内容】 登録数が最大値を超えました。

【対処】 不要な設定を削除してください。

E4790W	リスト中のエディタが選択されていません。
	Editor in list not selected.

【内容】 操作すべきエディタが選択されていません。

【対処】 エディタリストの中から必要なエディタを選択してから操作してください。

E4791W	標準エディタは削除・変更できません。
	The standard editor cannot delete and change.

【内容】 標準エディタを削除または変更しようとしています。

【対処】 標準エディタは、SOFTUNE Workbench の内蔵エディタです。削除・変更はできません。

E4792W	指定の実行ファイルが存在しませんが、リストに登録しますか？
	This executable file does not exist. Register in the list?

【内容】 指定された実行ファイルが見つかりません。

【対処】 実行ファイル名、ディレクトリ名に間違いが無ければそのまま登録してください。

E4793W	有効なエディタは削除できません。
	The valid editor cannot delete.

【内容】 [使用するエディタ] として選択されているエディタを削除しようとしてしました。

【対処】 [使用するエディタ] を別のエディタに変更してから削除してください。

E4794W	指定のディレクトリが存在しません。ディレクトリを作成しますか？
	Directory not found. Create this directory?

【内容】指定されたディレクトリが見つかりません。

【対処】ディレクトリを作成する場合は、OK ボタンを押してください。

E4795W	タイトルが長すぎます。
	Title too long.

【内容】タイトル名が最大文字数を超過しています。

【対処】タイトル名を短くしてください。

E4796W	実行ファイル名が長すぎます。
	Execute file name too long.

【内容】実行ファイル名が長すぎます。

【対処】実行ファイル名を短くしてください。

E4797W	オプション文字列が長すぎます。
	Option string too long.

【内容】オプション文字列が最大文字数を超過しています。

【対処】オプション文字列を短くしてください。

E4798W	実行時のディレクトリが長すぎます。
	The executing directory too long.

【内容】ディレクトリが長すぎます。

【対処】ディレクトリを短くしてください。

## 付録 B デバッガ関係のエラーメッセージ

デバッガ関係のエラーメッセージを以下に示します。

### ■ デバッガ関係のエラーメッセージ

F9201S	セットアップファイルが見つかりません。
	Invalid setup file (not found ).

【内容】指定されたセットアップファイルが見つかりませんでした。

【対処】起動オプションのセットアップファイル指定で指定しているファイルが存在していることを確認してください。

F9202S	セットアップファイルの記述に誤りがあります。
	Invalid command or parameter (in setup file ).

【内容】セットアップファイル中に不正なコマンドあるいはパラメータがあります。  
または、MCU 設定情報が変更されたため、パラメータあるいはコマンドを変更する必要があります。

【対処】セットアップウィザードを使用して起動し直してください。

F9203S	マシン依存ファイルの実行エラーが発生しました。
	Invalid machine program (execution error).

【内容】すでに実行されているか、システムリソース不足により実行処理が行えません。

【対処】実行状態を確認し、実行されていない場合は、表示ウィンドウを閉じるもしくは、他の起動プログラムを終了してください。

F9204S	ディスク容量が足りません。
	The disk space is insufficient.

【内容】ロギング中に保存先のディスクが 500MB 未満になりました。

【対処】ディスクの空き容量を増やして下さい。

F9401S	エミュレーションポッドが違います。
	Invalid emulation pod or MCU cable (unmatch or no-connect).

【内容】エミュレーションポッド、もしくは、MCU ケーブルが本製品用のものではありません。または、正しく接続されていません。

【対処】エミュレータの電源を切り、エミュレーションポッド、MCU ケーブルを確認し、正常に接続後、再起動してください。

F9402S	エミュレータのモニタプログラムが違います。
	Invalid emulator hardware monitor program (unmatch ).

【内容】エミュレータ内部にロードされているモニタプログラムが本製品のものではありません。

【対処】本製品に付属しているローダプログラムを起動して、モニタプログラムをロードした後、再起動してください。詳細は、[インストール説明書]を参照してください。

F9403S	エミュレータのハードウェアエラーです。[" 詳細情報 "]
	Emulator hardware error. ["detailed information"]

【内容】エミュレータハードが正常動作できなくなりました。または、詳細情報に示される例外を検出したため、処理を中断しました。

【対処】・ "RAM checker over-flow"

他のアプリケーションなどの影響により、指定された間隔でログデータを取得できなかったため、RAM チェッカのロギングを中断しました。ロギング中は、マシンに負荷のかかる動作を行わないでください。

・ 詳細情報なし

MCU が正常動作をしているかを確認してください。エミュレータ本体をリセットし、再起動してください。頻繁に発生する場合は、エミュレータハード、または MCU、ターゲットシステムの故障が考えられます。

F9404S	エミュレータのモニタプログラムのバージョンが違います。
	Invalid emulator hardware monitor program version (old).

【内容】エミュレータハード内のモニタプログラムのバージョンが古いために動作できません。

【対処】モニタローダプログラムを用いて、製品に添付しているモニタをロードしてください。詳細は、[インストール説明書]を参照してください。

F9405S	バスエラーが発生しました。リセットが必要です。
	A bus error occurs. To issue reset is necessary for resource.

【内容】メモリへアクセスしたところ、バスエラーが発生しました。

【対処】バスエラーが発生したため、エミュレータから MPU を制御できません。リセットを自動的に発生しました。

F9406S	プロジェクトとターゲットのチップ種別が違います
	Invalid either chip classification in project or chip on board.

【内容】ターゲットのチップが、プロジェクトで指定されているチップ種別と違います。

【対処】ターゲットのチップ、もしくはプロジェクトで指定されているチップ種別を確認してください。

F9407S	通信デバイスが認識できません。デバッガを終了します。
	Cannot recognize a communication device and terminate debugger.

【内容】エミュレータに接続されている通信デバイスが認識できなくなりました。デバッガを強制終了します。

【対処】ホスト PC とエミュレータ間の接続を確認してください。

F9408S	ターゲットと ICE 接続環境の整合性がありません。
	Don't match target for ICE environment.

【内容】セットアップファイルで指定してある ICE 設定と接続している ICE 環境が異なります。

【対処】セットアップウィザードを起動してセットアップファイルの ICE 種別を変更するか、ICE の環境を正しいものに变更してください。

F9410S	コンフィギュレーションボードが接続されていません。
	A configuration board is not connected.

【内容】コンフィギュレーションボードが正しく接続されていません。

【対処】エミュレータの電源を切り、コンフィギュレーションボードを確認し、正常に接続後、再起動してください。

F9601S	通信エラーが発生しました。
	Invalid communication status (or cable connection ).

【内容】通信回線の状態が異常であるか、ケーブルが正しく接続されていません。

【対処】通信回線の状態を確認してください。

この場合には、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。デバッグ中に USB ケーブルを抜差しした場合にも、通信エラーとなります。

F9602S	通信デバイス名が違います。
	Invalid communication device name (or not specified ).

【内容】通信デバイス名の指定に誤りがあります。

【対処】インストールファイル中の通信デバイス名を確認してください。

F9603S	インストールファイル内に INTERFACE が設定されていません。
	Invalid INTERFACE ( not specified in install file ).

【内容】インストールファイル内に INTERFACE の指定がありません。

【対処】インストールファイルを確認してください。

F9604S	"WINSOCK.DLL" が初期化できません。
	Cannot initialize "WINSOCK.DLL".

【内容】 WINSOCK.DLL が初期化できません。

【対処】 ご使用の LAN ソフトに対応した "WINSOCK.DLL" が必要です。LAN ソフトのマニュアル等をご参照の上、Windows のディレクトリ、または、PATH の通ったディレクトリに入れてください。

F9901S	メモリが足りません。
	Memory allocation error.

【内容】 起動のための十分なメモリが確保できません。

【対処】 ホスト PC の空きメモリを増やして、再起動してください。

F9902S	OS から異常なエラーが返されました。
	System error.

【内容】 OS に異常があり、正常に起動できませんでした。

【対処】 OS を起動し直して、本プログラムを再度起動してください。

F9903S	必要な DLL ファイルが見つかりませんでした。
	A necessary DLL file was not found.

【内容】 必要な DLL ファイルをロードできませんでした。

【対処】 SOFTUNE Workbench を再インストールしてください。

F9904S	CPU 情報のバージョンが旧バージョンです。
	The version of CPU information file is an old version.

【内容】 CPU 情報ファイルが古いため、正しく情報を設定できません。

【対処】 お使いの CPU 情報ファイルを最新のファイルに更新してください。

F9905S	USB 通信に必要なファイルが見つかりませんでした。
	A necessary file for the USB communication is not found.

【内容】 USB 通信に必要な DLL ファイルが見つかりませんでした。

【対処】 SOFTUNE Workbench を再インストールしてください。

W1001S	データがアンダフローしました。
	Invalid data value (underflow).

【内容】 指定した精度においてデータがアンダフローしました。

【対処】 精度、またデータの値を見直してください。



W1002S	データがオーバーフローしました。
	Invalid data value (overflow).

【内容】指定した精度においてデータがオーバーフローしました。

【対処】精度，またデータの値を見直してください。

W1101S	二重定義シンボルがありました。
	Invalid symbol (multiple).

【内容】二重定義シンボルがありました。

【対処】LOAD コマンド実行時に出力された場合は，ロードモジュールに対応するソースファイルを見直してください。

W1102S	ロードモジュールに実行開始アドレス情報がありません。
	Invalid code section or entry data (not found in load module).

【内容】ロードしたロードモジュールには，コードセクション及びエントリー情報が入っていませんでした。この場合プログラムカウンタ (PC) は設定されません。

【対処】プログラムカウンタ (PC) を設定してからプログラムを実行してください。

W1103S	メモリ不足のためコマンドヒストリが使いません。
	Command history buffer allocation error (in host memory).

【内容】コマンドヒストリ用のバッファをホスト PC の内部メモリに取れません。

【対処】ホスト PC の内部メモリを増設してください。このまま使用すると，コマンドヒストリ機能は使用できません。

W1104S	指定アドレスが 16 ビット境界または 32 ビット境界に合っていません。
	Invalid address (mis-alignment).

【内容】FR ファミリ MCU では，16 ビットデータ，32 ビットデータは，それぞれ 16 ビット境界，32 ビット境界のアドレスしかアクセスできません。

【対処】指定アドレスを見直してください。

W1201S	HELP コマンドファイルが見つかりません。
	Invalid HELP command file (not found).

【内容】HELP コマンドファイルが正しい位置に存在しません。

【対処】HELP コマンドファイルを正しい位置に置いてください。

W1202S	異なった MCU シリーズ用のロードモジュールです。
	Loaded different series's file.

【内容】プロジェクトで設定したターゲット MCU と異なった MCU シリーズのロードモジュールをロードしました。異なった MCU シリーズのロードモジュールでは、命令動作が正しく行えない場合があります。

【対処】目的のファイルでない場合は、ファイルを確認後、再ロードしてください。

W1401S	実行時間タイマがオーバーフローしました。
	Invalid timer (overflow).

【内容】プログラム実行中に実行時間タイマがオーバーフローしました。

【対処】測定時間を短くしてください。

W1402S	パフォーマンスバッファがオーバーフローしました。
	Invalid performance measuring data (buffer full).

【内容】実行中にパフォーマンス測定データを格納するバッファがいっぱいになりました。バッファがいっぱいになった以降は、測定されていません。

【対処】測定回数を少なくしてください。

W1403S	パスカウントがオーバーフローしました。
	Invalid pass count (overflow).

【内容】パスカウントがオーバーフローしました。

【対処】式中の項を確認の上、再入力してください。

W1404S	コマンド実行中にユーザリセットが入りました。
	User reset.

【内容】コマンド実行中に MCU にユーザリセットが入りました。

【対処】[ 実行 ] - [ 連続実行 ] メニューを再入力してください。

W1901S	セットアップファイルは読み取り専用です。セットアップ情報の変更は保存できません。
	The setup file is read-only. The change in setup information cannot be preserved.

【内容】セットアップファイルが読み取り専用のため、保存できませんでした。

【対処】セットアップ名に対応するセットアップファイルのファイル属性から読み取り属性をはずしてください。

W1902S	CPU 情報が正しくありません。
	Invalid CPU information data.

【内容】CPU 情報ファイルの情報が正しくありません。

【対処】最新の CPU 情報ファイルを入手してください。

W1903S	DLL のバージョンが古い可能性があります。
	There is a possibility with an old version of DLL.

【内容】プログラム本体と DLL ファイルのバージョンが合っていません。

【対処】最新の DLL ファイルをインストールしてください。

W1904S	セットアップファイルを更新する必要があります。セットアップウィザードを起動します。
	Start "Setup Wizard" to update the setup file.

【内容】MCU 情報等が変更されたため、セットアップファイルの情報を更新する必要があります。

【対処】セットアップウィザードにより、ファイルの更新を行ってください。

E4001S	コマンド名に誤りがあります。
	Command error.

【内容】コマンドの文法またはラインアセンブラの文法に誤りがあります。

【対処】文法、パラメータを確認の上、再入力してください。

E4002S	コマンド修飾子名に誤りがあります。
	Command qualifier error.

【内容】コマンド修飾子名に誤りがあるか、指定されたコマンド修飾子はそのコマンドに存在しません。

【対処】コマンド修飾子名を確認の上、再入力してください。

E4003S	文法に誤りがあります。
	Syntax error.

【内容】コマンドの文法またはラインアセンブラの文法に誤りがあります。

【対処】文法、パラメータを確認の上、再入力してください。

E4004S	パラメータが多すぎます。
	Invalid parameter count (over limit).

【内容】パラメータの数が多すぎます。

【対処】コマンドの文法を確認の上、再入力してください。

E4005S	パラメータが足りません。
	Invalid parameter omission.

【内容】省略できないパラメータを省略しました。

【対処】コマンドの文法を確認の上、再入力してください。

E4006S	パラメータに誤りがあります。
	Parameter error.

【内容】パラメータの指定方法に誤りがあります。パラメータ名を間違えているか、数値として認識できません。

【対処】コマンドの文法または入力基数を確認の上、正しく再入力してください。

E4007S	計算式のオペランドに誤りがあります。
	Invalid operand.

【内容】式中に不正な項があります。または、浮動小数点数を用いて演算しようとしていました。浮動小数点数による演算はできません。

【対処】式中の項を確認の上、再入力してください。

E4008S	演算子に誤りがあります。
	Invalid operator.

【内容】式中に不正な演算子があります。

【対処】式中の演算子を確認の上、再入力してください。

E4009S	計算式のオペランドがありません。
	Syntax error (operand not found).

【内容】式中の多項演算子に項がありません。

【対処】式を確認の上、正しく入力してください。

E4010S	ダブルクォートまたはシングルクォート記号が足りません。
	Syntax error ( ' ' or ' ' not found).

【内容】式中の " あるいは ' に対する右側の " あるいは ' がありません。または、1文字以上からなる文字定数です。

【対処】式を確認の上、正しく入力してください。

E4011S	計算式のネストレベルが深すぎます。
	Invalid nest level (over limit).

【内容】式中の ( ) , \* , [ ] のネストレベルが 16 を超えています。または、構造体、共用体のネストレベルが 16 を超えています。

【対処】式を簡単にしてください。

E4012S	計算式で 0 による除算が起こりました。
	Syntax error (dividing by zero).

【内容】式中において 0 による除算があります。

【対処】式中の項を確認の上，再入力してください。

E4013S	アドレスの指定に誤りがあります。
	Invalid address specifying.

【内容】終了アドレスが開始アドレスよりも小さい場合，または指定された範囲が複数の領域にまたがっている場合が考えられます。

【対処】アドレスを確認の上，再入力してください。

E4014S	0x01 ~ 0xff 以外のビットパターンが指定されました。
	Invalid bit pattern (over 1 to 0x ff).

【内容】指定されたビットパターンが 0x1 ~ 0xff 以外の値です。

【対処】ビットパターンを確認の上，再入力してください。

E4015S	0 ~ 31 以外のビットオフセットが指定されました。
	Invalid bit offset (over 0 to 31).

【内容】指定されたビットオフセットが 0 ~ 31 以外の値です。

【対処】ビットオフセットを確認の上，再入力してください。

E4016S	レジスタ名に誤りがあります。
	Invalid register or flag name (not found).

【内容】指定されたようなレジスタ / フラグ名は存在しません。

【対処】レジスタ / フラグ名を確認して，再入力してください。

E4017S	シンボルが見つかりません。
	Invalid symbol (not found).

【内容】指定されたシンボルはシンボルテーブル中に存在しません。または，指定されたシンボルがローカル変数で，そのシンボルパスネームがカレントスコープに登録されていません。

【対処】シンボル名の記述に誤りがないか，そのシンボルが属するモジュールのシンボル情報がシンボルテーブルに登録されているかを確認して，再入力してください。シンボルが属するモジュールのシンボル情報がシンボルテーブルに登録されている場合はシンボルパスネームをつけて指定するか，カレントスコープにシンボルパスネームを登録してください。

E4018S	コマンド別名が見つかりません。
	Invalid command alias (not found).

【内容】指定されたようなコマンドの別名は存在しません。

【対処】コマンドの別名を確認して再入力してください。

E4019S	行番号が見つかりません。
	Invalid line number (not found).

【内容】ソースファイル中に指定された行番号はありません。または、ソースファイルに対応したロードモジュールファイル(行番号情報)がロードされていません。

【対処】ソースファイルを確認して、再入力してください。または、ソースファイルに対応したロードモジュールファイルをロードしてください。

E4020S	行番号が大きすぎます。
	Invalid starting display line number (over ending line number).

【内容】ソース行表示開始行番号が表示終了行番号より大きいです。

【対処】行番号を確認の上、再入力してください。

E4021S	サイクル数の指定が 0 です。
	Invalid cycle count (0).

【内容】サイクル数に 0 が指定されました。

【対処】サイクル数を確認の上、再入力してください。

E4022S	ブレークポイント番号が見つかりません。
	Invalid break point number (not found).

【内容】指定されたブレークポイント番号は存在しません。

【対処】ブレークポイント番号を確認してください。

E4023S	データブレークポイント番号が見つかりません。
	Invalid data break point number (not found).

【内容】指定されたデータブレークポイント番号は存在しません。

【対処】データブレークポイント番号を確認してください。

E4024S	ウォッチポイント番号が見つかりません。
	Invalid watch point number (not found).

【内容】指定されたウォッチポイント番号は存在しません。

【対処】ウォッチポイント番号を確認してください。

E4025S	トレース開始番号に終了番号より大きな値が指定されました。
	Invalid starting display trace number (over ending number).

【内容】表示開始トレース番号が表示終了トレース番号より大きいです。

【対処】トレース番号を確認の上，再入力してください。

E4026S	書式制御文字列に誤りがあります。
	Invalid format statement characters.

【内容】指定された書式制御文字列に誤りがあります。

【対処】書式制御文字列を確認の上，再入力してください。

E4027S	シンボルのスコープ指定に誤りがあります。
	Invalid symbol (not found) path name.

【内容】指定されたシンボルパス名は存在しません。

【対処】シンボルパス名を確認の上，再入力してください。

E4028S	関数が見つかりません。
	Invalid function (not found, or argument error).

【内容】指定された関数は存在しません。または関数の引数指定に誤りがあります。

【対処】関数または引数を確認の上，再入力してください。

E4029S	構造体・共用体のメンバ名がありません。
	Invalid expression (used variable of structure or union type).

【内容】構造体または共用体型の変数を言語式中の項には使用できません。

【対処】式を見直してください。ただし，&演算子を前置きすれば使用できます。

E4030S	行番号に対するアドレスが見つかりません。
	Invalid address (not found).

【内容】行番号に対応するアドレスが存在しません。

【対処】行番号を見直してください。

E4031S	変数がアクティブではありません。
	Invalid automatic variable reference.

【内容】指定された自動変数を変数が定義されている関数外で参照しようとしてしました。

【対処】自動変数は，変数が定義されている関数内でなければ参照できません。

E4032S	構造体・共用体のメンバが見つかりません。
	Invalid variable specifying.

【内容】指定された変数は、構造体または共用体変数のメンバではありません。

【対処】構造体または共用体のメンバを確認してください。

E4033S	浮動小数点数の形式に誤りがあります。
	Floating point data format error.

【内容】浮動小数点データの形式が間違っています。

【対処】浮動小数点データの形式を見直してください。

E4034S	マクロコマンドが見つかりません。
	Invalid macro command definition (not found).

【内容】指定されたマクロコマンド名は存在しません。

【対処】マクロコマンド名を確認の上、再入力してください。

E4035S	シンボルのアドレスが見つかりません。
	Invalid symbol address (not found)

【内容】レジスタ変数またはレジスタに割り付けた変数のアドレスを参照しました。  
例)

```
SET DATABREAK &r
```

変数が割り付けられたレジスタがCPUのアドレス空間上に存在する場合は、そのアドレスを参照することができます。

【対処】レジスタ変数およびレジスタに割り付けた変数のアドレスを参照することはできません。

E4101S	コマンドのネストレベルが深すぎます。
	Invalid command list nest level (over 8).

【内容】コマンドプロシージャ、コマンドマクロ、および、ブレークポイントのコマンドリスト実行のネストが8レベルを超えています。

【対処】コマンド実行を見直してください。

E4102S	シンボルが定義できません。
	Symbol definition error.

【内容】ホスト PC のメモリに、コマンド実行に必要な十分な空きメモリがありませんでした。パソコン環境 (MS-DOS) において、多くのデバイスドライバを組み込みすぎた場合などに発生します。

【対処】ホスト PC の空きメモリを増やしてから、本プログラムを再起動してください。



E4103S	OS コマンドが実行できません。
	OS command error.

【内容】OS コマンドの実行ができません。コマンドシェルのフォーマットが異常です。

【対処】正常なフォーマットのコマンドシェルを起動してください。

E4104S	コマンドシェルが見つかりません。
	Invalid command shell (not found).

【内容】コマンドシェルが見つかりません。

【対処】コマンドシェルが起動できるように環境変数等の見直しをしてください。

E4105S	コマンド別名に登録できない文字が含まれています。
	Invalid alias string.

【内容】コマンド別名に登録できない文字が含まれています。

【対処】コマンド別名の登録を見直して、再入力してください。

E4106S	同じマクロコマンドが既に登録されています。
	Invalid macro command name (registered already).

【内容】同じマクロコマンドが既に登録されています。

【対処】マクロコマンド名を見直して、再入力してください。

E4107S	マップの領域指定に誤りがあります。
	Invalid memory map definition.

【内容】メモリマップの定義が複雑すぎるため、これ以上領域を定義できません。メモリの領域属性を設定していった時、異なる属性の領域が多すぎると内部テーブルがあふれます。

【対処】メモリマップを簡略化してください。

E4108S	メモリが足りません。
	Memory allocation error.

【内容】ホスト PC のメモリに、コマンド実行に必要な十分な空きメモリがありませんでした。多くのデバイスドライバを組み込みすぎた場合などに発生します。

【対処】ホスト PC の空きメモリを増やしてから、本プログラムを再起動してください。

E4109S	アドレスが最大値 を超えました。
	Object loading error.

【内容】オブジェクトのロード先が、アドレスの最大値を超えました。

【対処】オブジェクトの大きさとロード先または、指定したアドレスを確認してください。

E4110S	ログファイルは既にオープンしています。
	Log file open error (already).

【内容】ログファイルは既にオープンしています。

【対処】新しいログファイルをオープンするためには、今のログファイルを閉じてから行ってください。

E4111S	マップ未定義アドレスをアクセスしました。
	Memory access error.

【内容】未定義属性のメモリをアクセスしようとしてしました。アドレスの部分には、エラーの原因となるアクセスを行ったアドレスが表示されます。

【対処】メモリマップを確認してください。

E4112S	構造体・共用体のネストレベルが深すぎます。
	Invalid nest level of structure or union (over 16).

【内容】デバッグ情報テーブルをホスト PC のメモリ上に作成することができませんでした。

【対処】ホスト PC の空きメモリを増やして、本プログラムを再起動してください。

E4113S	デバッグ情報テーブルを作成できませんでした。
	Debug data table creation error.

【内容】デバッグ情報テーブルをホスト PC のメモリ上または、インストールファイル中の TMP で指定されているディレクトリ上に作成することができませんでした。

【対処】ホスト PC の空きメモリを増やして、本プログラムを再起動してください。または、インストールファイル中の TMP で指定されているディレクトリの状態を調べてください。

E4114S	ログファイルがオープンされていません。
	Logging control command error.

【内容】ログファイルがオープンされていない状態で、ログファイルへの操作が行われました。

【対処】ログファイルがオープンされているかを確認してください。

E4115S	同じコマンド別名が既に登録されています。
	Invalid alias name (registered already).

【内容】同じコマンド別名が既に登録されています。

【対処】コマンド別名を見直して、再入力してください。

E4116S	コマンド別名が見つかりません。
	Invalid alias name (not found).

【内容】指定されたコマンド別名は存在しません。

【対処】コマンド別名を確認の上，再入力してください。

E4117S	データの型が合いません。
	Data type error.

【内容】データの型が合っていないです。

【対処】データを確認後，再入力してください。

E4118S	構造体・共用体は指定できません。
	Invalid member name (not specified).

【内容】構造体または，共用体名は指定できません。

【対処】メンバ名を付けて指定してください。

E4119S	ブレークポイントが設定できません。
	Break point and data break point setting error.

【内容】ブレークポイントおよび，データブレークポイントの設定ができません。  
MB2198 エミュレータの場合には，データ監視ブレークの設定ができない場合  
にも表示されます。

【対処】ブレークポイントの設定内容および，設定数を確認してください。  
MB2198 エミュレータの場合，ブレークポイントがトレーストリガやシーケン  
サと重複している可能性があります。イベントリストで設定状態を確認して  
ください。

E4120S	CALL コマンドが既に実行中です。
	CALL command error.

【内容】CALL コマンドが既に実行中です。CALL コマンドはネストできません。

【対処】CLEAR CALL コマンドにより CALL を中断するか，もしくは，GO，STEP コ  
マンドでコール動作が終了するまで実行させて，その後，CALL コマンドを  
行ってください。

E4121S	親関数が見つかりません。
	Invalid function (at the top).

【内容】これ以上上位の関数がないか，C 言語以外のプログラムから呼び出されてい  
ます。

【対処】現在の関数を確認してください。

E4122S	これ以上 DOWN できません。
	Invalid function (at the bottom).

【内容】これ以上下位の関数が無いか，C 言語以外のプログラムから呼び出されています。

【対処】現在の関数を確認してください。

E4123S	カバレッジ領域数が多すぎます。
	Invalid coverage map (over-full).

【内容】カバレッジ領域をこれ以上設定することはできません。

【対処】カバレッジ領域に指定を簡略化してください。

E4124S	カバレッジの設定に誤りがあります。
	Coverage area setting error.

【内容】カバレッジ領域の設定がありません。

【対処】カバレッジ領域の設定を行って指定してください。

E4125S	カバレッジ領域指定に誤りがあります。
	Invalid coverage area.

【内容】カバレッジ領域外を指定しました。

【対処】カバレッジ領域を確認して指定してください。

E4126S	カバレッジファイルではありません。
	Invalid coverage file.

【内容】カバレッジファイル以外を指定しました。

【対処】ファイル内容を確認してください。

E4127S	デバッグ情報がロードされていません。
	Invalid debug data (not loaded).

【内容】デバッグ情報ファイルがロードされていません。

【対処】デバッグ情報ファイルをロードしてからカバレッジ指定を行ってください。

E4128S	マップ領域がオーバーラップしています。
	Mapping overlap.

【内容】指定されたマップ領域が他の領域と重なっています。

【対処】マップ指定を確認の上，再入力してください。

E4129S	指定アドレスが、サイズ指定によって決まる境界に合っていません。
	Invalid address (mis-alignment).

【内容】16 ビットデータ、32 ビットデータは、それぞれ 16 ビット境界、32 ビット境界のアドレスしかアクセスできません。

【対処】指定アドレスを見直してください。

E4130S	現在の位置を表示するソースウィンドウがオープンできません。
	Cannot open current source window.

【内容】現在の位置を表示するソースウィンドウが設定されているソース検索ディレクトリ内に見つかりませんでした。

【対処】ソースファイルのあるディレクトリを設定してください。

E4131S	現在のデバグ種別では使用できません。
	Cannot be used in current mode of debugger.

【内容】デバグの種類に応じて使用できる機能が異なります。

【対処】デバグの種類を確認してください。

E4132S	現在のデバグ機能では使えないコマンドです。
	Command error(debug mode violation).

【内容】デバグ機能に応じて使用できる機能が異なります。

【対処】[ 環境 ] - [ デバグ環境の設定 ] - [ デバグ機能の選択 ] メニューでデバグ機能の設定を確認してください。

E4133S	タスクデバグ機能が有効ではありません。
	Task debugging cannot be used.

【内容】タスクデバグ機能が有効なオブジェクトがロードされていない場合、タスクデバグ機能は使用できません。

【対処】タスクデバグ機能付きオブジェクトをロードしてください。

E4134S	MMU アクセスエラーが発生しました。
	MMU data access exception.

【内容】MMU アクセスエラーが発生しました。

【対処】不正なメモリをアクセスしていないか確認してください。

E4135S	レジスタ例外が発生しました。
	Register exception.

【内容】未定義のレジスタをアクセスしています。

【対処】復旧できない場合はターゲットの電源を入れ直してください。正しいレジスタをアクセスしているか確認してください。

E4136S	並列命令の先頭アドレスではありません。
	Not the first address of a parallel instruction.

【内容】指定されたアドレスは並列に実行される命令の先頭のアドレスではありません。

【対処】並列に実行される命令の先頭のアドレスを指定して下さい。

E4137S	レジスタにアクセスできません。
	Register access error.

【内容】レジスタに対してアクセスできません。

【対処】チップのマニュアルを参照してください。

E4138S	64 ビット境界を越えるアドレスとサイズの組み合わせは指定できません。
	Invalid combination of size and address beyond 64 bits boundary.

【内容】指定アドレスから 64 ビット境界を跨ぐようなサイズは指定できません。

【対処】指定アドレスから 64 ビット境界に収まるようなサイズを指定してください。

E4139S	設定数を超えています。
	Over point

【内容】指定した機能は、設定数を超えています。

【対処】MB2198 エミュレータで、トレーストリガやデータ監視ブレーク RAM チェックを設定したときに表示されます。

現在の設定数と設定状態を確認し、不要な設定を削除して改めて設定してください。

E4140S	指定した番号が見つかりません。
	Invalid point number

【内容】指定したポイント番号が見つかりません。

【対処】イベントの削除などで、指定したポイント番号が存在しない場合に表示します。ポイント番号を確認してください。

E4141S	同じアドレスが設定されています。
	The same address is set

【内容】同じアドレスに対して、同じ機能を設定しています。

【対処】例えば、シーケンサの設定のとき、複数のレベルに対して同じアドレスを設定した場合などに表示します。現在の設定数と設定状態を確認し、不要な設定を削除して改めて設定してください。

E4142S	監視条件が設定されていません。
	Invalid break condition

【内容】データ監視ブレーク条件が設定されていません。

【対処】MB2198 エミュレータで、イベント用リソースが不足して監視条件ブレークが設定できないときに表示されます。現在の設定数と設定状態を確認し、不要な設定を削除して改めて設定してください。

E4143S	このデバイスではサポートされていません。
	No supported function with this device. (MBXXXX)

【内容】このデバイスではサポートされていない機能を指定しました。

【対処】評価用チップまたはエミュレータに実装されていない機能を指定した可能性があります。エラーメッセージ内の MB 番号を基に、LSI 仕様書または Workbench のコマンドリファレンスを参照してください。

E4144S	RAM チェッカは USB デバイスでご使用ください。
	Please use the RAM checker function with an USB device.

【内容】通信デバイスに USB 以外（RS または LAN）が使用されているため、RAM チェッカが使用できません。

【対処】RAM チェッカを使用する場合は、通信デバイスに USB を使用してください。

E4201S	ファイルがアクセスできません。
	File access error.

【内容】ファイルに対してアクセスができませんでした。

【対処】ホスト PC のディスクの状態を調べてください。

E4202S	ファイルがクローズできません。
	File close error.

【内容】ファイルのクローズ処理ができません。

【対処】ホスト PC のディスクの状態を調べてください。

E4203S	ファイルがオープンできません。
	File open error.

【内容】ファイルがオープンできません。

【対処】ファイル名またはホスト PC のディスクの状態を調べてください。または、ファイルのあるディレクトリを確認してください。

E4204S	ファイルが書き込めません。
	Data write error.

【内容】ファイルヘータを書き込めません。

【対処】ホスト PC のディスクの状態を調べてください。

E4205S	アドレスに対する行番号が見つかりません。
	Invalid line number (not found).

【内容】指定されたアドレスに対応するソース行が存在しません。ソースウィンドウ表示で、対応するソース行が存在しない場合も表示されます。

【対処】アドレスを見直してください。または、デバッグ情報付きのロードモジュールをロードしてください。

E4206S	エイリアスファイルがロードできません。
	Alias file load error.

【内容】指定されたエイリアスファイルがロードできません。

【対処】エイリアスファイル名またはホスト PC のディスクの状態を調べてください。または、ファイルのあるディレクトリを確認してください。

E4207S	エイリアスファイルがセーブできません。
	Alias file save error.

【内容】指定されたエイリアスファイルがセーブできません。

【対処】ホスト PC のディスクの状態を調べてください。

E4208S	ファイルの形式が違います。
	Invalid file format.

【内容】ロードしようとしたファイルの形式が違います。

【対処】ファイルの内容を確認してください。



E4209S	ファイルがリードできません。
	Open file read error.

【内容】オープンしたファイルの読み出し途中でエラーが発生しました。

【対処】読み出し中のファイル（ドライブ）を確認してください。

E4301S	割込み番号に誤りがあります。
	Invalid interrupt factor number.

【内容】指定された割込み番号は存在しません。

【対処】割当て可能な割込み番号を指定してください。(IRQ0 ~ IRQ47)

E4302S	I/O バッファ番号に誤りがあります。
	Invalid I/O buffer number.

【内容】指定された I/O バッファ番号は存在しません。

【対処】I/O バッファは 0 ~ 3 個までシミュレータデバッガでは用意されています。

E4303S	ポートアドレスが大きすぎます。
	Invalid port address .

【内容】ポートアドレスの範囲を越えたアドレスが指定されました。

【対処】ポートアドレスは、MCU の I/O 領域内しか指定できません。MCU の I/O 領域アドレス内のアドレスを指定してください。

E4304S	データ出力先は既に使われています。
	Invalid output destination.

【内容】データ出力先に既に使用されている出力先が指定されました。

【対処】まだ使用されていないデータの出力先を指定してください。

E4305S	ポート数が多すぎます。
	Invalid port count .

【内容】指定されたポートの数が 4096 個を超えてしまいました。

【対処】4096 個を超えるポートの指定はできません。

E4306S	シミュレーションメモリが確保できません。
	Simulation memory allocation error.

【内容】シミュレーションメモリをホスト PC の内部メモリ上に作成することができませんでした。

【対処】ホスト PC の内部メモリを増設してください。

E4307S	入力ポートに割り当てたファイルの内容に誤りがあります。
	Invalid inport data file.

【内容】入力ポートに割り当てたファイル名が間違っているか、またはそのファイルが存在しません。

【対処】ファイルの記述形式を確認してください。

E4401S	メモリベリファイエラーが発生しました。
	Verify error.

【内容】コマンドによるメモリへの書込み時にベリファイエラーが発生しました。

【対処】値の変化する I/O 領域に書込みをしていないか、メモリが実装されているか、また、メモリに異常がないかを調べてください。

E4402S	エミュレーションメモリでパリティエラーが発生しました。
	Parity error (at emulation memory).

【内容】エミュレーションメモリへのアクセスでパリティエラーが発生しました。

【対処】エミュレータ本体をリセットし、再起動してください。頻繁に発生する場合は、エミュレーションメモリの故障が考えられます。

E4403S	エミュレータの作業用メモリでパリティエラーが発生しました。
	Parity error (at debug memory).

【内容】エミュレータ作業用メモリのアクセスでパリティエラーが発生しました。

【対処】エミュレータ本体をリセットし、再起動してください。頻繁に発生する場合は、エミュレータ作業用メモリの故障が考えられます。

E4404S	MCU 実行中は使えないコマンドです。
	Command error (MCU is busy).

【内容】MCU 実行中に実行できないコマンドを実行しようとしてしました。

【対処】コマンドを確認してください。

E4405S	Reserved
	Reserved

【内容】

【対処】

E4406S	Reserved
	Reserved

【内容】

【対処】

E4407S	Reserved
	Reserved

【内容】

【対処】

E4408S	イベント番号が見つかりません。
	Invalid event number (not found).

【内容】 指定されたイベント番号は存在しません。

【対処】 イベント番号を確認してください。

E4409S	レベル番号が見つかりません。
	Invalid level number (not found).

【内容】 指定されたレベル番号が存在しません。

【対処】 レベル番号を確認してください。

E4410S	このイベントモードでは使えないコマンドです。
	Command error (event mode violation).

【内容】 イベントモードに違反するコマンドを指定しました。

【対処】 [ 環境 ] - [ デバッグ環境の設定 ] - [ デバッグ環境設定 ] メニューでイベントモードの設定を確認してください。

E4411S	ラッチ番号が見つかりません。
	Invalid latch number (not found).

【内容】 指定されたラッチ番号が存在しません。

【対処】 ラッチ番号を確認してください。

E4412S	ユーザシステムからの電源電圧に異常があります。
	Invalid supply voltage.

【内容】ユーザシステムから供給されている電源電圧に異常があります。

【対処】ユーザシステムの電源電圧を見直してください。

E4413S	MCU が STOP モード中か、またはシステムクロックに異常があります。
	MCU is in STOP mode, or Invalid system clock.

【内容】ユーザシステムから供給されるシステムクロックに異常があります。

【対処】ユーザシステムのシステムクロックを見直してください。

E4414S	MCU がリセットできません。
	MCU reset error.

【内容】MCU のリセットが正常に行えません。

【対処】リセット時に読み出したモードデータ、リセットベクタが正しい値ではないことが考えられます。正しい値を設定してやり直してください。なお、ユーザメモリからモードデータを読み出す場合に、このエラーが発生するとユーザメモリが読めなくなるため、エミュレーションメモリにマップしてから、[ デバッグ ] - [ MCU のリセット ] メニューを実行してください。

E4415S	MCU が応答しません。
	Invalid MCU.

【内容】MCU が動作できない状態にあるため、コマンドが実行できません。

- 【対処】
1. リセットベクタ、モードデータを設定して [ デバッグ ] - [ MCU のリセット ] メニューを実行してください。
  2. ユーザシステム側でスリープ、ストップ、ホールド状態を解除するか、リセットベクタモードデータを設定して [ デバッグ ] - [ MCU のリセット ] メニューを実行してください。ただし、ホールド状態は [ デバッグ ] - [ MCU のリセット ] メニューでは、解除できません。
  3. コマンドの実行結果を確認してください。

E4416S	分岐先レベル番号に誤りがあります。
	Invalid jump level number.

【内容】シーケンサの分岐先レベル番号に誤りがあります。

【対処】分岐先レベル番号を見直してください。なお、指定するレベルと同じレベルには、分岐できません。

E4417S	内部 ROM リアルタイムモードでは使えないコマンドです。
	Command error (on internal ROM real-time mode).

【内容】MCU の実行モードがネイティブモードのため、コマンドが実行できません。

【対処】MCU の実行モードをデバッグモードに変更してください。

E4418S	ユーザリセットが入っているためコマンドが実行できません。
	Command error (user reset).

【内容】ユーザリセットが入っているため、コマンドが実行できません。また、既にユーザリセットが解除されていてもこのエラーが出ます。

【対処】ユーザリセットを解除後、[ デバッグ ] - [ MCU のリセット ] メニューを実行してから、再度コマンドを実行してください。

E4419S	ABORT コマンドが実行できません。
	Abort command error.

【内容】SLEEP、STOP 状態のため、ABORT コマンドが実行できません。

【対処】SLEEP、STOP 状態を解除してください。

E4420S	ハードウェアスタンバイ状態のためコマンドが実行できません。
	Command error (hardware standby).

【内容】ハードウェアスタンバイ状態のため、コマンドが実行できません。また、既にハードウェアスタンバイが解除されていてもこのエラーが出ます。

【対処】ハードウェアスタンバイを解除後、[ デバッグ ] - [ MCU のリセット ] メニューを実行してから、再度コマンドを実行してください。

E4421S	現在のタイマモードでは、このコマンドは実行できません。
	Command error (timer-mode violation).

【内容】タイマモードがタイマの場合には SHOW CYCLE、CLEAR CYCLE コマンドが、タイマモードがサイクルの場合には SHOW TIMER、CLEAR TIMER コマンドが実行できません。

【対処】タイマモードを確認の上、再入力してください。

E4422S	ブレイクポイントが見つかりません。
	Invalid break point (not found).

【内容】ソフトウェアブレイクポイントが設定されているアドレスの内容が、プログラム実行によって書換えられたため、ソフトウェアブレイクポイントが無効になりました。または、ソフトウェアブレイクポイントの埋め戻し時にエラーが発生し、ソフトウェアブレイクポイントがメモリに残ってしまいました。この場合、ロードされているプログラムの内容、および、ソフトウェアブレイクポイントの設定内容は保証されません。

【対処】ソフトウェアブレイクをすべて削除し、プログラムの内容を見直してください。ソフトウェアブレイクがメモリ内に残っている場合は、プログラムをロードし直してください。

E4423S	モニタ内部でスタックチェック例外が発生しました。
	Monitor hit stack-check function.

【内容】ユーザプログラムへの復帰時にモニタ内部でスタックチェック例外が発生しました。

【対処】スタックチェック機能を無効にするか、使用できるスタック領域を増やしてください。

E4424S	ユーザリソースアクセス中に例外が発生しました。[“ 詳細情報 ”]
	Exception occurred while accessing user resource. ["detailed information"]

【内容】指定のユーザリソースをアクセス中に、詳細情報に示される例外を検出したため、処理を中断しました。

【対処】詳細情報を元に、LSI 仕様書に記述されている例外発生要因から該当する事項を確認してください。

E4425S	区間番号に誤りがあります。
	Invalid area number.

【内容】指定された区間番号は存在しません。

【対処】設定可能な区間番号を指定してください。

E4426S	このイベントモードでは使えないコマンドです。(Performance)
	Command error(event mode violation<Performance>).

【内容】イベントモードがパフォーマンスモードのため、コマンドが実行できません。

【対処】イベントモードをコマンドが使用可能なモードに変更してください。

E4427S	このイベントモードでは使えないコマンドです。(Single trace)
	Command error(event mode violation<Single trace>).

【内容】イベントモードがシングルトレースモードのため、コマンドが実行できません。

【対処】イベントモードをコマンドが使用可能なモードに変更してください。

E4428S	このイベントモードでは使えないコマンドです。(Multi trace)
	Command error(event mode violation<Multi trace>).

【内容】イベントモードがマルチトレースモードのため、コマンドが実行できません。

【対処】イベントモードをコマンドが使用可能なモードに変更してください。

E4501S	メモリベリファイエラーが発生しました。
	Verify error.

【内容】コマンドによるメモリへの書き込み時にベリファイエラーが発生しました。

【対処】値の変化する I/O 領域に書き込みをしていないか、メモリが実装されているか、また、メモリに異常がないかを調べてください。

E4502S	スタック領域が確保されていません。
	Illegal stack area.

【内容】モニタデバッガで使用するスタック領域がアクセス可能ではありません。

【対処】スタック領域を正しく確保してください。

E4503S	システムコールを実行できる状態ではありません。
	System call error (cannot execute).

【内容】システムコールを正常に実行できる状態ではありません。

【対処】システムコールの発行できる状態で実行してください。割り込み禁止状態である場合などが考えられます。

E4504S	このコマンドは、組み込まれていません。
	This command is not built-in.

【内容】ターゲット側に対応する機能が組み込まれていません。

【対処】ターゲット側プログラムに機能を組込んでください。

E4601S	通信エラーが発生しました。
	Invalid communication status (or cable connection).

【内容】通信回線の状態が異常であるか，ケーブルが正しく接続されていません。

【対処】回線の接続状態を確認してください。

E4602S	パラレルアダプタが接続されていません。
	Communication : Parallel adapter not connected.

【内容】パラレル通信アダプタが接続されていません。

【対処】パラレル通信アダプタを正しく接続し，再度実行してください。

E4603S	パラレルアダプタのバージョンが適合しません。
	Communication : Mismatch parallel adapter version.

【内容】パラレル通信アダプタの版数が古い場合，通信できません。

【対処】最新のパラレル通信アダプタをご用意ください。

E4604S	通信エラー：指定されたホスト名が見つかりません。
	Communication : Cannot find host name.

【内容】指定されたホスト名が hosts ファイルに登録されていません。

【対処】ホスト名を hosts ファイルに登録してください。詳細は，SOFTUNE Workbench 操作マニュアル 付録 C[ LAN インタフェースの設定 ]を参照してください。

E4605S	通信エラー：ICE のポート番号が取得できません。
	Communication : Cannot find port number.

【内容】services ファイル内に ICE のポート番号が定義されていません。

【対処】ポート番号を services ファイルに登録してください。詳細は，SOFTUNE Workbench 操作マニュアル 付録 C[ LAN インタフェースの設定 ]を参照してください。

E4606S	通信エラー：デバイスに接続できません。
	Communication : Cannot open device.

【内容】指定されたデバイスに異常があるか，正しく接続されていません。

【対処】指定されたデバイスが正しく接続されているか確認してください。

E4607S	通信エラー：タイムアウト
	Communication : Time out.

【内容】送信情報に対する受信情報が一定時間内に受信されませんでした。

【対処】指定されたデバイスが正しく接続されているか確認してください。



E4901S	タイマリソースが足りません。
	Not enough timer resource.

【内容】Windows のタイマリソースが使用できません。

【対処】他のアプリケーションを終了し、再度実行してください。

E4902S	そのキーコードは定義できません。
	The key code cannot be defined.

【内容】そのキーコードは定義できません。

【対処】他のキーコードに定義してください。

## 付録 C デバッガ中断メッセージ

デバッガ中断メッセージを以下に示します。

### ■ デバッガ中断メッセージ

Break at アドレス by breakpoint

【内容】ソフトウェアブレークポイントによってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by hardware breakpoint

【内容】ハードウェアブレークポイント（GO コマンドで指定したブレークポイントも含む）によってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by code event break（No. コードイベント番号）

【内容】コードイベントによってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。コードイベント番号は、ブレーク要因となったコードイベントの番号です。

Break at アドレス by code event break (sequential)

【内容】コードイベント 1, 2 によるシーケンシャルブレークによってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by data event break (No. データイベント番号)

【内容】データイベントによってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。データイベント番号は、ブレーク要因となったデータイベントの番号です。

Break at アドレス by data event break (sequential)

【内容】データイベント 1, 2 によるシーケンシャルブレークによってブレークした場合に表示されます。アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by trace buffer full
------------------------------------

【内容】トレースバッファフルによってブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by alignment error break (code)
---

【内容】コードフェッチによるアライメントエラーによってブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by alignment error break (data)
---

【内容】データアクセスによるアライメントエラーによってブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by external trigger break
---

【内容】エミュレータが持っている TRIG 端子への外部信号の入力によってブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by trace lost break
-----------------------------------

【内容】トレースデータの欠落によりブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by data break at アクセスアドレス
---

【内容】データブレイクポイントによってブレイクした場合には表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。  
 アクセスアドレスは、ブレイク要因となったアクセスを行ったアドレスです。

Break at アドレス by guarded access アクセス種別 at アクセスアドレス
--

【内容】コードフェッチができない領域のコードフェッチアクセス、またはデータリードができない領域のリードアクセス、またはデータライトができない領域のライトアクセスのいずれかによってブレイクした場合には表示されます。  
 メモリの属性、またはプログラムに誤りがあると考えられます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。  
 アクセス種別は、ブレイク要因となったアクセスの種類です。  
 アクセスアドレスは、ブレイク要因となったアクセスを行ったアドレスです。

Break at アドレス by dispatch task from task ID=< ディスパッチ元タスク ID > to task ID=< ディスパッチ先タスク ID >

【内容】タスクディスパッチブレイクによってブレイクした場合に表示されます。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by system call < システムコール名 > on task ID=< タスク ID >

【内容】システムコールブレイクによってブレイクした場合に表示されます。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。  
ブレイク要因となったシステムコール名と、そのシステムコールが発行されたタスク ID を表示します。

Break at アドレス by command abort request

【内容】デバッグ ][ 停止 ]メニューの入力によってブレイクした場合に表示されます。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by output file overflow

【内容】出力ポートのデータ出力ファイルへの書き込みができなかったためにブレイクした場合に表示されます。  
出力ポートのデータ出力ファイルを確認してください。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by stop abnormal action

【内容】プリフィックス命令の後に実行不可能な命令が付加されていたためにブレイクした場合に表示されます。  
プログラムに誤りがあると考えられますので、プログラムの確認をしてください。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

Break at アドレス by invalid call termination

【内容】CALL コマンドは、現在の PC が指すアドレスにブレイクポイントを設定し、そのアドレスにリターンするよう、RP レジスタを設定して実行します。このため、CALL コマンド実行中に元の PC のアドレスを実行するとブレイクしてしまいます。  
このように、CALL コマンドの完了前にブレイクした場合に表示されます。  
そのまま GO コマンドで実行再開するか、CLEAR CALL コマンドで CALL コマンドの実行を中断してください。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

## Break at アドレス by EIT ( 付加情報 )

- 【内容】EIT の発生によりブレイクした場合に表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。  
 付加情報は LSI マニュアルを参照してください。

## Break at アドレス by step command

- 【内容】ステップ（INTO）実行でブレイクした場合に、SHOW STATUS コマンドで表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

## Break at アドレス by call command

- 【内容】ステップ（OVER）実行でブレイクした場合に、SHOW STATUS コマンドで表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

## Break at アドレス by violation to combine instructions

- 【内容】シミュレータデバッガにおいて、1 並列命令内の許可されていない組合せの命令を実行してブレイクした場合に表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

## Break at アドレス by slot issue violation

- 【内容】シミュレータデバッガにおいて、1 並列命令内でスロット内に発行できない命令があるにもかかわらず命令を実行してブレイクした場合に表示されます。  
 アドレスは、実行を停止した（次に実行する命令の）アドレスです。

## Break at アドレス by resource write-write confliction

- 【内容】シミュレータデバッガにおいて、1 並列命令内に同じメモリまたはレジスタをライトアクセスする複数個の命令を実行する場合に表示されます。  
 アドレスは、ブレイク要因となった命令に対して、次に実行する並列命令のアドレスです。

## Break at アドレス by datawatchbreak

- 【内容】エミュレータデバッガにおいて、データ監視ブレイクによってブレイクした場合に表示されます。  
 アドレスは、ブレイク要因となる命令のあったアドレスを示します。

Break at アドレス by unknown break factor
---------------------------------------

【内容】ブレーク要因が不定の場合に表示されます。  
アドレスは、実行を停止した（次に実行する命令の）アドレスです。

# 索引

## Symbols

%B	
機能	249
形式	249
対応デバッグ	249
例	249
%BIT	
機能	249
形式	249
対応デバッグ	249
例	249
%CALL	
機能	242
形式	242
対応デバッグ	242
例	242
%D	
機能	249
形式	249
対応デバッグ	249
例	249
%DEBUGFUNCTION	
機能	261
形式	261
対応デバッグ	261
例	261
%ENTRY	
機能	244
形式	244
対応デバッグ	244
例	244
%ERRNUM	
機能	243
形式	243
対応デバッグ	243
例	243
%EVAL	
機能	258
形式	258
対応デバッグ	258
例	258
%EVENTNUM	
機能	259
形式	259
対応デバッグ	259
例	259
%H	
機能	249
形式	249
対応デバッグ	249
例	249
%LOADNUM	
機能	248
形式	248
対応デバッグ	248
例	248

%RADIX	
機能	246
形式	246
対応デバッグ	246
例	246
%S	
機能	249
形式	249
対応デバッグ	249
例	249
%SCPADR	
機能	247
形式	247
対応デバッグ	247
例	247
%STKTOP	
機能	245
形式	245
対応デバッグ	245
例	245
%STRCAT	
機能	254
形式	254
対応デバッグ	254
例	254
%STRCMP	
機能	252
形式	252
対応デバッグ	252
例	252
%STRGET	
機能	250
形式	250
対応デバッグ	250
例	250
%STRLLEN	
機能	253
形式	253
対応デバッグ	253
例	253
%STRSTR	
機能	251
形式	251
対応デバッグ	251
例	251
%SYMLEN	
機能	255
形式	255
対応デバッグ	255
例	255
%TOSTR	
機能	257
形式	257
対応デバッグ	257
例	257
%TOVAL	
機能	256

形式 .....	256
対応デバッグ .....	256
例 .....	256
%TRIGGERNUM	
機能 .....	260
形式 .....	260
対応デバッグ .....	260
例 .....	260
%W	
機能 .....	249
形式 .....	249
対応デバッグ .....	249
例 .....	249

## A

### ASSEMBLE

機能 .....	194
形式 .....	194
対応デバッグ .....	194
例 .....	194

## B

### BATCH

機能 .....	212
形式 .....	212
対応デバッグ .....	212
例 .....	212

### BREAK

機能 .....	239
形式 .....	239
対応デバッグ .....	239
例 .....	239

## C

### CALL

機能 .....	64
形式 .....	63
対応デバッグ .....	63
例 .....	64

### CANCEL ALIAS

機能 .....	218
形式 .....	218
対応デバッグ .....	218
例 .....	218

### CANCEL BREAK

機能 .....	77
形式 .....	77
対応デバッグ .....	77
例 .....	77

### CANCEL CODEEVENT

機能 .....	99
形式 .....	99
対応デバッグ .....	99
例 .....	99

### CANCEL COVERAGE

機能 .....	135
形式 .....	135
対応デバッグ .....	135

例 .....	135
---------	-----

### CANCEL DATABREAK

機能 .....	85
形式 .....	85
対応デバッグ .....	85
例 .....	85

### CANCEL DATAEVENT

機能 .....	105
形式 .....	105
対応デバッグ .....	105
例 .....	105

### CANCEL DATATRACEAREA

機能 .....	155
形式 .....	155
対応デバッグ .....	155
例 .....	155

### CANCEL EVENT

機能 .....	92
形式 .....	92
対応デバッグ .....	92
例 .....	92

### CANCEL INPORT

機能 .....	31
形式 .....	31
対応デバッグ .....	31
例 .....	31

### CANCEL INTERRUPT

機能 .....	38
形式 .....	38
対応デバッグ .....	38
例 .....	38

### CANCEL LOGGING

機能 .....	227
形式 .....	227
対応デバッグ .....	227
例 .....	227

### CANCEL MAP (type 1)

機能 .....	23
形式 .....	23
対応デバッグ .....	23
例 .....	23

### CANCEL MAP (type 2)

機能 .....	24
形式 .....	24
対応デバッグ .....	24
例 .....	24

### CANCEL OUTPORT

機能 .....	35
形式 .....	35
対応デバッグ .....	35
例 .....	35

### CANCEL RAMCHECK

機能 .....	168
形式 .....	168
対応デバッグ .....	168
例 .....	168

### CANCEL SEQUENCE

機能 .....	112
形式 .....	112
対応デバッグ .....	112
例 .....	112

### CANCEL TRACETRIGGER

機能 .....	161
----------	-----



形式 .....	161
対応デバッグ .....	161
例 .....	161
CANCEL VARIABLE	
機能 .....	221
形式 .....	221
対応デバッグ .....	221
例 .....	221
CANCEL WATCH	
機能 .....	51
形式 .....	51
対応デバッグ .....	51
例 .....	51
CLEAR CALL	
機能 .....	65
形式 .....	65
対応デバッグ .....	65
例 .....	65
CLEAR COVERAGE	
機能 .....	136
形式 .....	136
対応デバッグ .....	136
例 .....	136
CLEAR PERFORMANCE	
機能 .....	130
形式 .....	130
対応デバッグ .....	130
例 .....	130
CLEAR TIMER	
機能 .....	140
形式 .....	140
対応デバッグ .....	140
例 .....	140
CLEAR TRACE	
機能 .....	147
形式 .....	147
対応デバッグ .....	147
例 .....	147
COMPARE	
機能 .....	188
形式 .....	188
対応デバッグ .....	188
例 .....	188
COPY VECTOR	
機能 .....	39
形式 .....	39
対応デバッグ .....	39
例 .....	39

## D

DISABLE ALIGNMENTBREAK	
機能 .....	118
形式 .....	118
対応デバッグ .....	118
例 .....	118
DISABLE BREAK	
機能 .....	79
形式 .....	79
対応デバッグ .....	79
例 .....	79
DISABLE CODEEVENT	

機能 .....	101
形式 .....	101
対応デバッグ .....	101
例 .....	101
DISABLE DATABREAK	
機能 .....	87
形式 .....	87
対応デバッグ .....	87
例 .....	87
DISABLE DATAEVENT	
機能 .....	107
形式 .....	107
対応デバッグ .....	107
例 .....	107
DISABLE EVENT	
機能 .....	94
形式 .....	94
対応デバッグ .....	94
例 .....	94
DISABLE EXTERNALMEMORY	
機能 .....	47
形式 .....	47
対応デバッグ .....	47
例 .....	47
DISABLE LOGGING	
機能 .....	229
形式 .....	229
対応デバッグ .....	229
例 .....	229
DISABLE MAP	
機能 .....	22
形式 .....	22
対応デバッグ .....	22
例 .....	22
DISABLE RAMCHECK	
機能 .....	172
形式 .....	172
対応デバッグ .....	172
例 .....	172
DISABLE SEQUENCE	
機能 .....	114
形式 .....	114
対応デバッグ .....	114
例 .....	114
DISABLE TRACE	
機能 .....	149
形式 .....	149
対応デバッグ .....	149
例 .....	149
DISABLE TRACETRIGGER	
機能 .....	163
形式 .....	163
対応デバッグ .....	163
例 .....	163
DISABLE VERIFYMODE	
機能 .....	26
形式 .....	26
対応デバッグ .....	26
例 .....	26
DISABLE WATCHDOG	
機能 .....	43
形式 .....	43
対応デバッグ .....	43

例 .....	43
DISASSEMBLE .....	
機能 .....	195
形式 .....	195
対応デバッグ .....	195
例 .....	195
DOWN .....	
機能 .....	210
形式 .....	210
対応デバッグ .....	210
例 .....	210
DUMP .....	
機能 .....	191
形式 .....	191
対応デバッグ .....	191
例 .....	192

## E

ENABLE ALIGNMENTBREAK .....	
機能 .....	117
形式 .....	117
対応デバッグ .....	117
例 .....	117
ENABLE BREAK .....	
機能 .....	78
形式 .....	78
対応デバッグ .....	78
例 .....	78
ENABLE CODEEVENT .....	
機能 .....	100
形式 .....	100
対応デバッグ .....	100
例 .....	100
ENABLE DATABREAK .....	
機能 .....	86
形式 .....	86
対応デバッグ .....	86
例 .....	86
ENABLE DATAEVENT .....	
機能 .....	106
形式 .....	106
対応デバッグ .....	106
例 .....	106
ENABLE EVENT .....	
機能 .....	93
形式 .....	93
対応デバッグ .....	93
例 .....	93
ENABLE EXTERNALMEMORY .....	
機能 .....	46
形式 .....	46
対応デバッグ .....	46
例 .....	46
ENABLE LOGGING .....	
機能 .....	228
形式 .....	228
対応デバッグ .....	228
例 .....	228
ENABLE MAP .....	
機能 .....	21
形式 .....	21

対応デバッグ .....	21
例 .....	21
ENABLE RAMCHECK .....	
機能 .....	170
形式 .....	170
対応デバッグ .....	170
例 .....	170
ENABLE SEQUENCE .....	
機能 .....	113
形式 .....	113
対応デバッグ .....	113
例 .....	113
ENABLE TRACE .....	
機能 .....	148
形式 .....	148
対応デバッグ .....	148
例 .....	148
ENABLE TRACETRIGGER .....	
機能 .....	162
形式 .....	162
対応デバッグ .....	162
例 .....	162
ENABLE VERIFYMODE .....	
機能 .....	25
形式 .....	25
対応デバッグ .....	25
例 .....	25
ENABLE WATCHDOG .....	
機能 .....	42
形式 .....	42
対応デバッグ .....	42
例 .....	42
ENTER .....	
機能 .....	179
形式 .....	178
対応デバッグ .....	178
例 .....	179
EXAMINE .....	
機能 .....	176
形式 .....	176
対応デバッグ .....	176
例 .....	177
EXIT .....	
機能 .....	4
形式 .....	4
対応デバッグ .....	4
例 .....	4

## F

FILL .....	
機能 .....	189
形式 .....	189
対応デバッグ .....	189
例 .....	189

## G

### GO

機能 .....	58
形式 .....	58
対応デバッグ .....	58
例 .....	58

## I

### IF

機能 .....	236
形式 .....	236
対応デバッグ .....	236
例 .....	236

### INITIALIZE

機能 .....	3
形式 .....	3
対応デバッグ .....	3
例 .....	3

## L

### LIST

機能 .....	204
形式 .....	204
対応デバッグ .....	204
例 .....	205

### LOAD

機能 .....	199
形式 .....	198
対応デバッグ .....	198
例 .....	200

## M

### MOVE

機能 .....	190
形式 .....	190
対応デバッグ .....	190
例 .....	190

## P

### PRINTF

機能 .....	231
形式 .....	230
対応デバッグ .....	230
例 .....	231

## Q

### QUIT

機能 .....	213
形式 .....	213
対応デバッグ .....	213
例 .....	213

## R

### REPEAT

機能 .....	237
形式 .....	237
対応デバッグ .....	237
例 .....	237

### RESET

機能 .....	5
形式 .....	5
対応デバッグ .....	5
例 .....	5

## S

### SAVE

機能 .....	202
形式 .....	201
対応デバッグ .....	201
例 .....	202

### SEARCH MEMORY

機能 .....	185
形式 .....	184
対応デバッグ .....	184
例 .....	185

### SEARCH TRACE

機能 .....	151
形式 .....	150
対応デバッグ .....	150
例 .....	151

### SET ALIAS

機能 .....	216
形式 .....	216
対応デバッグ .....	216
例 .....	216

### SET BREAKCONDITION ( type 1 )

機能 .....	121
形式 .....	120
対応デバッグ .....	120
例 .....	121

### SET BREAKCONDITION ( type 2 )

機能 .....	123
形式 .....	122
対応デバッグ .....	122
例 .....	123

### SET BREAK ( type 1 )

機能 .....	70
形式 .....	69
対応デバッグ .....	69
例 .....	70

### SET BREAK ( type 2 )

機能 .....	72
形式 .....	72
対応デバッグ .....	72
例 .....	72

### SET BREAK ( type 3 )

機能 .....	74
形式 .....	73
対応デバッグ .....	73
例 .....	74

### SET CODEEVENT

機能 .....	95
----------	----

形式 .....	95
対応デバッグ .....	95
例 .....	95
SET COVERAGE	
機能 .....	131
形式 .....	131
対応デバッグ .....	131
例 .....	131
SET DATABREAK ( type 1 )	
機能 .....	81
形式 .....	80
対応デバッグ .....	80
例 .....	81
SET DATABREAK ( type 2 )	
機能 .....	83
形式 .....	82
対応デバッグ .....	82
例 .....	83
SET DATAEVENT	
機能 .....	103
形式 .....	102
対応デバッグ .....	102
例 .....	103
SET DATATRACEAREA	
機能 .....	152
形式 .....	152
対応デバッグ .....	152
例 .....	152
SET EVENT	
機能 .....	89
形式 .....	88
対応デバッグ .....	88
例 .....	90
SET EXTERNALMEMORY	
機能 .....	45
形式 .....	45
対応デバッグ .....	45
例 .....	45
SET FREQUENCY	
機能 .....	52
形式 .....	52
対応デバッグ .....	52
例 .....	52
SET INPORT	
機能 .....	29
形式 .....	28
対応デバッグ .....	28
例 .....	29
SET INTERRUPT	
機能 .....	36
形式 .....	36
対応デバッグ .....	36
例 .....	36
SET LOGGING	
機能 .....	225
形式 .....	224
対応デバッグ .....	224
例 .....	225
SET MAP ( type 1 )	
機能 .....	17
形式 .....	17
対応デバッグ .....	17
例 .....	17

SET MAP ( type 2 )	
機能 .....	18
形式 .....	18
対応デバッグ .....	18
例 .....	18
SET MEMORY	
機能 .....	181
形式 .....	180
対応デバッグ .....	180
例 .....	181
SET MODE	
機能 .....	9
形式 .....	9
対応デバッグ .....	9
例 .....	10
SET OUTPUT	
機能 .....	33
形式 .....	32
対応デバッグ .....	32
例 .....	33
SET OUTPUT	
機能 .....	232
形式 .....	232
対応デバッグ .....	232
例 .....	232
SET PATH	
機能 .....	206
形式 .....	206
対応デバッグ .....	206
例 .....	206
SET PERFORMANCE	
機能 .....	127
形式 .....	127
対応デバッグ .....	127
例 .....	127
SET RADIX	
機能 .....	12
形式 .....	12
対応デバッグ .....	12
例 .....	12
SET RAMCHECK	
機能 .....	165
形式 .....	164
対応デバッグ .....	164
例 .....	165
SET REALTIMEMEMORYAREA	
機能 .....	54
形式 .....	54
対応デバッグ .....	54
例 .....	54
SET REGISTER	
機能 .....	186
形式 .....	186
対応デバッグ .....	186
例 .....	186
SET RUNMODE	
機能 .....	6
形式 .....	6
対応デバッグ .....	6
例 .....	7
SET SEQUENCE ( type 1 )	
機能 .....	109
形式 .....	108

対応デバッグ .....	108
例 .....	109
SET SEQUENCE ( type 2 ) .....	
機能 .....	110
形式 .....	110
対応デバッグ .....	110
例 .....	110
SET SOURCE .....	
機能 .....	14
形式 .....	14
対応デバッグ .....	14
例 .....	14
SET STEP .....	
機能 .....	61
形式 .....	61
対応デバッグ .....	61
例 .....	61
SET TRACETRIGGER .....	
機能 .....	159
形式 .....	158
対応デバッグ .....	158
例 .....	159
SET TRACE ( type 1 ) .....	
機能 .....	141
形式 .....	141
対応デバッグ .....	141
例 .....	141
SET TRACE ( type 2 ) .....	
機能 .....	142
形式 .....	142
対応デバッグ .....	142
例 .....	143
SET TRIGGER .....	
機能 .....	115
形式 .....	115
対応デバッグ .....	115
例 .....	115
SET VARIABLE .....	
機能 .....	219
形式 .....	219
対応デバッグ .....	219
例 .....	219
SET VECTOR .....	
機能 .....	40
形式 .....	40
対応デバッグ .....	40
例 .....	40
SET WATCH .....	
機能 .....	50
形式 .....	49
対応デバッグ .....	49
例 .....	50
SHOW ALIAS .....	
機能 .....	217
形式 .....	217
対応デバッグ .....	217
例 .....	217
SHOW ALIGNMENTBREAK .....	
機能 .....	119
形式 .....	119
対応デバッグ .....	119
例 .....	119
SHOW BREAK .....	

機能 .....	75
形式 .....	75
対応デバッグ .....	75
例 .....	76
SHOW BREAKCONDITION .....	
機能 .....	124
形式 .....	124
対応デバッグ .....	124
例 .....	124
SHOW CALLS .....	
機能 .....	137
形式 .....	137
対応デバッグ .....	137
例 .....	137
SHOW CODEEVENT .....	
機能 .....	97
形式 .....	97
対応デバッグ .....	97
例 .....	97
SHOW COVERAGE .....	
機能 .....	134
形式 .....	132
対応デバッグ .....	132
例 .....	134
SHOW DATABREAK .....	
機能 .....	84
形式 .....	84
対応デバッグ .....	84
例 .....	84
SHOW DATAEVENT .....	
機能 .....	104
形式 .....	104
対応デバッグ .....	104
例 .....	104
SHOW DATATRACEAREA .....	
機能 .....	154
形式 .....	154
対応デバッグ .....	154
例 .....	154
SHOW DETAILTRACE .....	
機能 .....	157
形式 .....	156
対応デバッグ .....	156
例 .....	157
SHOW EVENT .....	
機能 .....	91
形式 .....	91
対応デバッグ .....	91
例 .....	91
SHOW EXTERNALMEMORY .....	
機能 .....	48
形式 .....	48
対応デバッグ .....	48
例 .....	48
SHOW FREQUENCY .....	
機能 .....	53
形式 .....	53
対応デバッグ .....	53
例 .....	53
SHOW INPORT .....	
機能 .....	30
形式 .....	30
対応デバッグ .....	30

例 .....	30
SHOW INTERRUPT	
機能 .....	37
形式 .....	37
対応デバッグ .....	37
例 .....	37
SHOW LOGGING	
機能 .....	226
形式 .....	226
対応デバッグ .....	226
例 .....	226
SHOW MAP (type 1)	
機能 .....	19
形式 .....	19
対応デバッグ .....	19
例 .....	19
SHOW MAP (type 2)	
機能 .....	20
形式 .....	20
対応デバッグ .....	20
例 .....	20
SHOW MEMORY	
機能 .....	183
形式 .....	182
対応デバッグ .....	182
例 .....	183
SHOW MODE	
機能 .....	11
形式 .....	11
対応デバッグ .....	11
例 .....	11
SHOW OUTPORT	
機能 .....	34
形式 .....	34
対応デバッグ .....	34
例 .....	34
SHOW OUTPUT	
機能 .....	233
形式 .....	233
対応デバッグ .....	233
例 .....	233
SHOW PATH	
機能 .....	207
形式 .....	207
対応デバッグ .....	207
例 .....	207
SHOW PERFORMANCE	
機能 .....	129
形式 .....	128
対応デバッグ .....	128
例 .....	129
SHOW RADIX	
機能 .....	13
形式 .....	13
対応デバッグ .....	13
例 .....	13
SHOW RAMCHECK	
機能 .....	167
形式 .....	167
対応デバッグ .....	167
例 .....	167
SHOW REALTIMEMEMORYAREA	
機能 .....	55

形式 .....	55
対応デバッグ .....	55
例 .....	55
SHOW REGISTER	
機能 .....	187
形式 .....	187
対応デバッグ .....	187
例 .....	187
SHOW RUNMODE	
機能 .....	8
形式 .....	8
対応デバッグ .....	8
例 .....	8
SHOW SCOPE	
機能 .....	208
形式 .....	208
対応デバッグ .....	208
例 .....	208
SHOW SEQUENCE	
機能 .....	111
形式 .....	111
対応デバッグ .....	111
例 .....	111
SHOW SOURCE	
機能 .....	15
形式 .....	15
対応デバッグ .....	15
例 .....	15
SHOW STATUS	
機能 .....	66
形式 .....	66
対応デバッグ .....	66
例 .....	66
SHOW STEP	
機能 .....	62
形式 .....	62
対応デバッグ .....	62
例 .....	62
SHOW SYSTEM	
機能 .....	16
形式 .....	16
対応デバッグ .....	16
例 .....	16
SHOW TIMER	
機能 .....	138
形式 .....	138
対応デバッグ .....	138
例 .....	138
SHOW TRACETRIGGER	
機能 .....	160
形式 .....	160
対応デバッグ .....	160
例 .....	160
SHOW TRACE ( type 1 )	
機能 .....	145
形式 .....	144
対応デバッグ .....	144
例 .....	145
SHOW TRACE ( type 2 )	
機能 .....	146
形式 .....	146
対応デバッグ .....	146
例 .....	146

SHOW TRIGGER	
機能 .....	116
形式 .....	116
対応デバッグ .....	116
例 .....	116
SHOW VARIABLE	
機能 .....	220
形式 .....	220
対応デバッグ .....	220
例 .....	220
SHOW VECTOR	
機能 .....	41
形式 .....	41
対応デバッグ .....	41
例 .....	41
SHOW VERIFYMODE	
機能 .....	27
形式 .....	27
対応デバッグ .....	27
例 .....	27
SHOW WATCHDOG	
機能 .....	44
形式 .....	44
対応デバッグ .....	44
例 .....	44
STEP	
機能 .....	60
形式 .....	59
対応デバッグ .....	59
例 .....	60

## U

### UP

機能 .....	209
形式 .....	209
対応デバッグ .....	209
例 .....	209

## W

### WHILE

機能 .....	238
形式 .....	238
対応デバッグ .....	238
例 .....	238

## え

### エラーメッセージ

デバッグ関係のエラーメッセージ .....	287
マネージャ関係のエラーメッセージ .....	264

## て

### デバッグ

デバッグ関係のエラーメッセージ .....	287
-----------------------	-----

## ま

### マネージャ

マネージャ関係のエラーメッセージ .....	264
------------------------	-----





CM71-00330-4

---

**富士通マイクロエレクトロニクス・CONTROLLER MANUAL**

FR ファミリ

SOFTUNE<sup>TM</sup> Workbench

コマンドリファレンスマニュアル

V6 対応

---

2008 年 6 月 第 4 版発行

発行 **富士通マイクロエレクトロニクス株式会社**

編集 マーケティング統括部 ビジネス推進部

---

