# Universal Probe

ソフトウェア・ユーザーズ・マニュアル

## **ARM Writer**

Copyright © 2014 Sohwa & Sophia Technologies Inc.

No. J090960-03



### 目 次

注	意事項7
	使用上の注意8
略	語・用語・記載ルール9
1.	本ソフトウェアの概要11
	1.1. システム要件
	1.2. 特徴
	1.2.1. 各種実行機能12
	1.2.2. メモリ・I/O 操作機能12
	1.3. 電源の入れ方/切り方13
	1.3.1. ターゲットとの接続方法と電源の入れ方13
	1.3.2. 電源の切り方とターゲットからの外し方13
2.	操作方法14
	2.1. 作業フロー
	2.2. 本ソフトウェアの起動
	2.2.1. 起動方法
	2.2.2. 起動オプション
	2.3. プロジェクトファイルの選択
	2.3.1. プロジェクトファイルを新規作成する18
	2.3.2. プロジェクトファイルを開く21
	2.4. プローブの選択
	2.5. License Code の登録
	2.6. CPU の選択
	2.7. JTAG の設定
	2.8. プローブ起動時の設定
	2.9. CPU 動作モードの設定
	2.10. フラッシュメモリ設定
	2.11. フラッシュメモリのクリア
	2.12. オブジェクトデータの書き込み
	2.13. ワークメモリ設定
	2.14. ブロック情報を調べる
	2.15. フラッシュメモリのフィルアップ
	2.16. フラッシュメモリ設定の保存
	2.17. プロジェクトファイルの保存42
	2.18. 本ソフトウェアの終了42

#### 2/185



3.	本ソフトウェアでのデータ表現4	3
	3.1. 数值表現	43
	3.2. アドレス表現	43
	3.3. データ表現	43
	3.4. メモリ・I/O ポートの読み書き	44
4.	メニュー	5
	4.1. ファイル	45
	4.1.1. 閉じる	45
	4.1.2. プロジェクトの新規作成	45
	4.1.3. プロジェクトを開く	47
	4.1.4. プロジェクトの保存	47
	4.1.5. プロジェクトを名前をつけて保存	47
	4.1.6. プロジェクトを閉じる	48
	4.1.7. アプリケーションの終了	48
	4.2. リソース	49
	4.2.1. ダウンロード	49
	4.2.2. アップロード	51
	4.2.3. プローブ環境の設定	53
	4.2.4. バッチファイルの自動実行設定	58
	4.2.5. プローブ起動時の設定	60
	4.2.6. メモリアクセスエリア設定	63
	4.2.7. メモリ/ポート	65
	4.2.8. フラッシュメモリ	65
	4.3. 実行	66
	4.3.1. 実行開始	66
	4.3.2. ブレーク	66
	4.3.3. リセット	66
	4.3.4. キャンセル	67
	4.3.5. ベリファイ(ライト時に比較)	67
	4.3.6. ベリファイオンリー(ライトせずに比較)	67
	4.3.7. チェックサム値の計算	68
	4.4. メモリ/ポート	69
	4.4.1. メモリのフィルアップ	69
	4.4.2. メモリチェック	71
	4.4.3. メモリサーチ	73
	4.4.4. メモリコピー	75
	4.5. フラッシュメモリ	77
	4.5.1. フラッシュメモリ設定ダイアログ	77
	4.5.2. デバイス設定タブ	79
3/	(8)	



	4.5.3. ワークメモリタブ	81
	4.5.4. ブロック情報タブ	82
	4.5.5. 設定ファイルの保存/読み込みタブ	83
	4.5.6. フラッシュメモリのクリアタブ	84
	4.5.7. フラッシュメモリのフィルアップタブ	86
5.	ウィンドウ	87
	5.1. ウィンドウのドッキングについて	87
	5.1.1. ウィンドウの状態	88
	5.1.2. 子ウィンドウ状態の変更方法	90
	5.1.3. ドラッグ&ドロップによるドッキング状態の変更方法	92
	5.1.4. 同じ辺に複数の子ウィンドウをドッキングさせた場合	94
	5.1.5. ドッキングウィンドウのサイズ変更	95
	5.2. ダンプウィンドウ	96
	5.2.1. ダンプウィンドウのコンテキストメニュー	98
	5.2.2. ダンプウィンドウの複数表示	
	5.2.3. "ASCII 文字列"入力の方法	
	5.3. コマンドウィンドウ	
6.	キーボード	105
	6.1. 全てのウィンドウ	
	6.2. ダンプウィンドウ	
	6.3. コマンドウィンドウ	
7.	コマンドラインインターフェース	106
	7.1. コマンドラインでの操作方法	
	7.1.1. コマンドの入力方法	
	7.2.2. 連想選択方式によるコマンド入力	
	7.2.3. 履歴から入力する	
	7.2. コマンド解説	
	7.2.1. ASSIGN とドット(.)(式の評価)	
	7.2.2. BATCH(バッチプログラムの実行)	
	7.2.3. BREAK(ブレーク)	110
	7.2.4. CD(フォルダの変更、ドライブの変更、カレントパス表示)	
	7.2.5. CHECK(メモリチェック)	
	7.2.6. CLOSE(プロジェクトファイルを閉じる)	
	7.2.7. CLS(コマンドウィンドウのクリア)	
	7.2.8. COPY(メモリコピー)	
	7.2.9. DIR(フォルタ内容の参照)	
	7 2 10 DUMP(メモリダンプ)	110
	7.2.10. DOPI (XC)7777	
	7.2.11. ENV(プローブ環境の設定)	



	7.2.12. ERROR_ECHO(エラーメッセージ表示の設定)	
	7.2.13. EXIT(本ソフトウェアの終了)	
	7.2.14. FILL(メモリフィルアップ)	
	7.2.15. FM(フラッシュメモリの設定)	
	7.2.16. FMCLEAR(フラッシュメモリのクリア)	
	7.2.17. FMDEL(フラッシュメモリの設定の削除)	
	7.2.18. FMLOAD(フラッシュメモリへのダウンロードの切り替え)	
	7.2.19. FMFILL(フラッシュメモリのフィルアップ)	134
	7.2.20. FMWORKAREA(フラッシュメモリのワークメモリの設定)	
	7.2.21. GO(CPU の実行)	
	7.2.22. LOAD(オブジェクトデータのロード)	
	7.2.23. LOADPARAM(LOAD コマンドのパラメータ補足)	
	7.2.24. LOG(コマンドウィンドウのロギング制御)	
	7.2.25. MEMATTR(メモリアクセスエリアの設定)	141
	7.2.26. MESSAGEBOX(ユーザーシステムメッセージボックスの表示/非表示)	
	7.2.27. MKDIR(フォルダの作成)	
	7.2.28. NEWBATCH(バッチファイルの作成)	
	7.2.29. OPTION(コマンドウィンドウのオプション設定)	
	7.2.30. QUERY(各種設定状態の参照)	
	7.2.31. RADIX(入力基数の設定)	
	7.2.32. RESET(CPU のリセット)	
	7.2.33. SAVEWIN(コマンドウィンドウのファイル出力)	
	7.2.34. SEARCH(メモリサーチ)	
	7.2.35. SHELLEXE(ファイルの実行)	154
	7.2.36. UPLOAD(オブジェクトデータのアップロード)	
	7.2.37. VERIFY(ベリファイの設定)	
	7.2.38. WAITBREAK(CPU がブレークするまで待つ)	
8.	バッチ機能	161
•		
	8.1. リーク変数	
	8.2. システム変数	
	8.5. 数値演算で使用可能な演算子	
	8.5.1. 演算子	
	8.5.2. 優先度と評価順序	
	8.6. メモリ・I/O のデータ読み書き	
	8.7. 実行制御	
	8.7.1. FOR,FBREAK,NEXT(カウンタ付き繰り返し実行)	
	8.7.2. WHILE,WBREAK,WEND(繰り返し実行)	
_ /	8.7.3. GOTO(無条件分岐)	
3/	183	



制件者信報	185
改訂履歴	
9.3. 記録した手順のバックアップとリストア	
9.2. 記録の終了	
9.1. 手順の記録	
9. スタンドアロン機能	
8.12. WAIT(バッチプログラムの一時停止)	
8.11. BEEP(BEEP 音を鳴らす)	
8.10. PRINT(文字列の表示)	
8.9. KEYIN(キーボードからの入力)	
8.8. ECHO(バッチコマンドの表示/非表示の切り替え)	
8.7.6. QUIT(現在のバッチプログラムの終了)	
8.7.5. END(全てのバッチプログラムの終了)	
8.7.4. IF,ELSEIF,ELSE,ENDIF(条件判断)	



## 注意事項

#### このたびは株式会社 Sohwa & Sophia Technologies 製「Universal Probe」をお買い上げいただき、誠にありがとう ございます。本書に記載されている注意事項などを正しくご理解のうえ、お使いいただきますようお願い申し上げます。

- 1. 本書に記載の製品及び技術で、『外国為替及び外国貿易法』に該当するものを輸出する時、又は、国外に持ち出す時は、日本政府の許可が必要です。
- 2. 本書に記載されている製品は、一般電子機器(事務機器、通信機器、計測機器、家電製品など)に使用されることを意図しております。特別な品質、信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼしたりする恐れのある特定用途機器(自動車・鉄道・船舶・航空・宇宙用機器、交通機器、燃焼機器、安全装置、医療機器、インフラ機器、原子力など)には使用しないでください。もしこれらの機器でご使用になる場合は、お客様の責任のもとでご使用ください。
- 3. 本書の内容の一部または全部を当社の文書による承諾なしに、無断で転載することは固くお断りいたします。
- 4. 本書に記載の内容は、将来予告なしに変更される場合があります。
- 5. 本書に記載の仕様は、お客様の環境、測定条件によって異なる結果が得られる場合があります。
- 6. 運用した結果の影響について、一切の責任を負いかねますのでご了承ください。
- 7. 本書に記載の「使用上のご注意」は、使用者や他者への危害と財産の損害を未然に防ぎ、安全に正しくお使いいただくための重要な注意事項です。ご使用になる前に必ずお読みください。
- 8. 本書に記載されている製品名および商品名は、各社の商標または登録商標です。



連絡先は 株式会社 Sohwa & Sophia Technologies のホームページでご確認ください。 URL > <u>http://www.ss-technologies.co.jp</u>



## 使用上の注意

	下記の注意を守らないと人が死亡する、または重傷を負う可能性があります。
登制	本製品に仕様で規定した範囲外の電源電圧を加えないでください。 範囲外ので電源電圧を加えると、破損・火災の恐れがあります。
<b>1</b> 強制	アース端子が付いているターゲットに使用する場合は、ターゲットや周辺機器のアースを確実に接続してください。機器の故障や感電の恐れがあります。 また、ガス管にアース端子をつながないでください。火災や爆発の原因になります。
<b>公</b> 禁止	本製品に接続した機器を取り付けたまま持ち運ばないでください。 特にケーブルはプラグを持って抜き差ししてください。ケーブルが破損し、火災・感電の恐れがあります。
<b>公</b> 禁止	ケーブルを取り扱う場合は次の点を守ってください。「傷つけない」「加工しない」「無理に曲げない」「ねじらない」「引っ張らない」「物を載せない」「加熱しない」「熱器具に近づけない」「濡れた手で触らない」。 これらを守らないと火災・感電の恐れがあります。 もしケーブルが破損した場合、そのケーブルの使用を中止してください。
<b>公</b> 禁止	雷が鳴りだしたら、電源プラグに触れないでください。感電の原因となります。 落雷により製品が破損したと思われる場合は、本製品の使用を中止してください。
<b>○</b> 禁止	ステープラの針、クリップなどの金属を内部に入れないでください。火災・故障の恐れがあります。
<b>公</b> 禁止	直射日光の当たる場所、熱器具の近く、極端な高温環境、極端な低温環境、振動の激しいところ、金属 や油を含むほこりの多い場所、スパイク系のノイズが発生する場所で使用したり、放置しないでください。 また、強い衝撃を与えないでください。
<b>父</b> 分解禁止	分解・改造・修理しないでください。火災・感電の恐れがあります。
水濡れ禁止	風呂場やコップの近くなど、液体のある場所、湿気の多い場所では使用しないでください。 感電する恐れがあります。 液体が本製品内部に入った場合はすぐに電源を切り、使用を中止してください。
注意	通電中の本製品に長時間触れていると低温やけどになる恐れがあります。 また、本製品を布団などで覆った状態で使用しないでください。
プラグを抜く	もし、異常なにおい・異常な音・発煙・発火した場合、または落としたり、強い衝撃を与えたりして破損、破 損した恐れのある場合は、すぐに電源を切ってください。そのまま使うと重大な事故を起こす可能性があります ので、使用を中止してください。



## 略語・用語・記載ルール

本書で使用する略語・用語や記載ルールについて説明します。

- … 特に記載がない限り、数値はすべてプラスの値とします。 数値について ٠
- ··· 2<sup>10</sup>=1024 を表します。(例:16K=16384) K(大文字) •
- k(小文字)
   … 1000を表します。(例:1kHz=1000Hz)

   [xxxxx]
   … xxxxx というウィンドウタイトルを示します。

   <xxxxx>
   … xxxxx というウィンドウ内の項目名を示します。

   •
- •

本書で使用する注釈・注意点などについては Figure 1 の通りです。



Figure 1



略語・用語の解説は Table 1 の通りです。

略語・用語	説明		
本製品	Universal Probe 本体・付属品を含むもの。		
プローブ	Universal Probe 本体のこと。		
本ソフトウェア	ARM Writer のこと。		
フラッシュメモリ	Flash メモリ、EEPROM などの総称です。		
Serial No.	Serial Numberの略。プローブの ID のこと。		
Software Code	各ソフトウェアの License Code を発行するために必要なライセンス。無償版は不要。		
License Code	本機で使用できる機能を追加するためのコード。		
モジュール	フラッシュメモリに書き込むデータのこと。オブジェクトデータとシンボルデータを含むもの。		
オブジェクトデータ	実際にフラッシュメモリに書き込む、バイナリデータのこと。		
シンボルデータ	変数名や関数名(=シンボル)とアドレスの対応関係を表すデータのこと。		
ホスト PC	本ソフトウェアを動作させる PC のこと。		
ターゲット	Universal Probeによって制御、計測する対象のこと。		
スタンドアロン機能	ホスト PC と接続しなくても動作すること。(電源は必要)		
(N/A)	Not Assigned の略。割り当てられた情報がないことを表す。		
PC	単に PC と記載した場合はプログラムカウンタの略。		

Table 1



## 1. 本ソフトウェアの概要

本ソフトウェアの概要を説明します。

本ソフトウェアは、ARM CPU の「**内蔵フラッシュメモリ**」、および ARM CPU に接続された「**外部フラッシュメモリ**」と「**SPI フラッシュ メモリ**」へデータを書き込むソフトウェアです。

別ソフトウェアとして「SPI Writer」がありますが、機能・対応する構成の違いを Figure 2 に示します。



Figure 2



## 1.1. システム要件

本ソフトウェアを動作させるためには、以下のシステムが必要です。

- Microsoft Windows 7 以降が動作する PC
- CPU: 1GHz 以上 (使用する OS の要件に準拠します)
- Memory: 1GB 以上 (使用する OS の要件に準拠します)
- HDD:空き容量 500MB 以上
- OS: Windows 7 以降 (32bit または 64bit)
- USB 2.0 の空きポート1 つ以上

## 1.2. 特徴

本ソフトウェアには、以下の特徴があります。

- 各種実行機能
- メモリ・I/O 操作機能
- プロジェクトファイルによる動作環境の保存と復元
- 書き込みデータのフォーマットを自動認識するダウンロード機能
- 複数のデータ書き込みをサポート
- バッチ機能により柔軟な書き込み処理を実現

#### 1.2.1. 各種実行機能

- CPU の実行機能
- CPU のブレーク(停止)機能
- CPU のリセット機能

#### 1.2.2. メモリ・I/O 操作機能

- 指定されたアドレス範囲へ特定データを書き込むフィルアップ機能
- 指定されたアドレス範囲のリード/ライトチェック機能
- 指定されたアドレス範囲の文字列やデータの検索するサーチ機能
- メモリマップト I/O アクセス機能
- 各種内部 I/O レジスタの表示および変更機能
- メモリデータのファイル保存機能



## 1.3. 電源の入れ方/切り方

1.3.1. ターゲットとの接続方法と電源の入れ方

以下の手順でプローブとターゲットを接続します。

- 1) ターゲットの電源が切れている事を確認します。
- 2) プローブの USB ケーブルを接続します。
- 3) ターゲットとプローブを接続します。
- 4) ターゲットの電源を入れます。
- 5) 本ソフトウェアを起動し、データの書き込み作業などを行います。

#### 1.3.2. 電源の切り方とターゲットからの外し方

以下の手順でプローブとターゲットを切り離します。

- 1) 本ソフトウェアを終了します。
- 2) ターゲットの電源を切ります。
- 3) ターゲットからプローブを外します。

→ プローブのハードウェア仕様、ターゲットの制限、接続などの詳細については、『ハードウェア・ユーザーズ・マニュアル』を 参照してください。



## 2. 操作方法

本ソフトウェアの操作方法について説明します。

## 2.1. 作業フロー

本ソフトウェアを起動してから、ユーザーのモジュールを書き込み、プロジェクトファイルを保存するまでの、主な工程を Figure 3 に示します。ここに記載されていない操作については「2.操作方法」もしくは「4.メニュー」の項を参照ください。





## 2.2. 本ソフトウェアの起動

本ソフトウェアの起動には、いくつかの方法があります。

#### 2.2.1. 起動方法

#### ●起動方法 1

「デスクトップ」にある本ソフトウェアのアイコンをダブルクリックします。

#### ●起動方法 2

 $\lceil Z_{P}-h 
floor$  \[Theorem \] \[Theorem







#### ●起動方法 3

エクスプローラなどで、ARMWriter.exeやプロジェクトファイル(.armwpj)をダブルクリックします。

						×
○ マ → コンピューター → 05 (     ○     □     ○     □     ○     □	C:) ► AR	RM_Writer 🕨	✓ 4 ARM_V	Vriterの検索		٩
ファイル(E) 編集( <u>E</u> ) 表示(⊻) ツール(	<u>I) へい</u>	プ( <u>H</u> )				_
整理 ▼ 🚺 開く ▼ 印刷 書き	き込む	新しいフォルダー			•	0
🌉 コンピューター	•	名前	更新日時	2種類	サイズ	
👝 OS (C:)		🍌 TEST	2014/09/26 15:20	ファイル フォル…		
🛃 \$Recycle.Bin		test.bak	2014/09/26 17:48	BAK ファイル	1 KB	
) Xilinx		test.exs	2014/09/26 19:37	EXS ファイル	2 KB	
🎳 _rpcs		test.fsh	2014/08/04 11:53	FSH ファイル	4 KB	
퉬 Android_SDK		STM32F10X-128K-EVAL_default_ro	2010/03/25 13:43	MOT ファイル	160 KB	
🐌 android-cts		o test.armwpj	2014/09/26 19:37	UniversalProbe	20 KB	
🎳 archive_db						
\mu ARM_Writer	-					
1 個選択				🎥 コンピュータ	9-	

Figure 5

#### ●起動方法 4

「スタート」→「プログラムとファイルの検索」に、ARMWriter.exe やプロジェクトファイル(.armwpj)を指定します。



Figure 6



以上の操作で下図のように、本ソフトウェアが起動します。

O Universal Probe	- • •
ファイル(E) 表示(⊻) 実行(⊻) ヘルプ(圧)	
ν̄ <sup>5</sup> (	

Figure 7

#### 2.2.2. 起動オプション

本ソフトウェアには、以下の起動オプションがあります。英字の大文字・小文字は区別しません。すべて半角(ASCII コード)で入力してください。

Table 2

起動オプション	内容
-STARTDEFAULT	本ソフトウェア起動時にターゲットをリセットします。(デフォルト)
-STARTRESET	本ソフトウェア起動時にターゲットをリセットします。-STARTDEFAULT と同じです。
-STARTBREAK	本ソフトウェア起動時にターゲットを強制ブレークし、リセットは行いません。
-STARTINITIALIZATIONBREAK	本ソフトウェア起動時に初期化を行い、ターゲットを強制ブレークします。
-STARTINITIALIZATION	本ソフトウェア起動時に初期化のみを行い、リセット・強制ブレークは行いません。

起動オプションは、本ソフトウェアのショートカットのプロパティを開き、「リンク先」の項目に追記して指定します。

●例				
C:¥Program	Files¥Sohwa&Sophia	Technologies¥Universal	Probe¥ARM	Writer¥ARMWriter.exe
-STARTINITIALIZA	TIONBREAK			



## 2.3. プロジェクトファイルの選択

本ソフトウェアは以下の設定を保存するために、必ずプロジェクトファイル(.armwpj)を使用します。

- 書き込むモジュールの情報
- 書き込み環境の設定状態

新規で書き込みを行うには、プロジェクトファイルを作成する必要があります。 既存のプロジェクトファイルを開くと、保存した状態を復元することができます。



フラッシュメモリの設定は保存されません。 フラッシュメモリの設定は[フラッシュメモリ設定]ダイアログで保存してください。

#### 2.3.1. プロジェクトファイルを新規作成する

例として、C:¥ARM\_Writer フォルダに TEST フォルダを作成し、test.armwpj というプロジェクトファイルを作成します。 以下のツールバーボタンまたは、メニューバーをクリックします。 ([プロジェクトの新規作成]ダイアログ詳細)



[プロジェクトの新規作成]ダイアログが開きます。

プロジェクトの新規作成	<b>X</b>	
プロジェクト名 (ℕ):	ОК	
	キャンセル	
_7℃-7゙	<u> 7<sup>°</sup>ロ~7<sup>°</sup>(P)</u>	
ターゲット: 選択されていません		
ライセンス: 選択されていません		
位置(L):		
C:¥ARM_WRITER¥	●照(13)	
Figure 8		•



プロジェクトファイルを保存するフォルダを作成するか、既存のフォルダを選択します。

#### ①フォルダを作成する

参照 ボタンをクリックします。

[フォルダの選択]ダイアログが開きます。

フォルダの選択		<b>x</b>
フォルダ( <u>D</u> ): c:¥arm_writer		ОК
🗁 c:¥	*	キャンセル
		ネットワーク
	-	7ォルダ(E)
ト <sup>*</sup> ライフ <sup>*</sup> (⊻) : ■ c: OS	•	

Figure 9

C:¥ARM\_Writerを選択して、フォルダ ボタンをクリックします。

[フォルダの作成]ダイアログが開きます。

フォルターの作成	<b>×</b>
現在のフォルダ( <u>F</u> ):	C¥ARM_WRITER
作成するフォルダ(⊆):	TEST
0	く キャンセル

Figure 10

<作成するフォルダ>に"TEST"を入力して OK ボタンをクリックします。



フォルダ名に"スペース"、"/"文字は使用できません。



[フォルダの選択]ダイアログに戻ります。

フォルダの選択	<b>x</b>
フォルダ( <u>D</u> ): c:¥arm writer¥test	ОК
Cita nigini kartisak	キャンセル
ARM_WRITER	ネットワーク
<b>.</b>	フォルダ(E)
, ドライブ(⊻) : ■ c: OS	

Figure 11

#### ②フォルダを選択する

作成した TEST フォルダを選択して、 OK ボタンをクリックします。

[プロジェクトの新規作成]ダイアログに戻ります。

#### ③プロジェクト名を指定する

(!)

<プロジェクト名>に、プロジェクト名(test)を指定します。 このプロジェクト名がプロジェクトファイル名になり、ダイアログ下部にある<位置>ボックスに表示されます。

	プロジェクトの新規作成	×		
	test	<del>ОК</del> ++уури	"test"と入力する	
	- プローブ- プローブ種類: 選択されていません ターゲット: 選択されていません ライセンス: 選択されていません	<u>7'n−7℃₽</u>		
	位置(L): C:¥ARM_WRITER¥TEST¥test.armwpj	参照( <u>R</u> )		
l	Eigure 12			



#### 2.3.2. プロジェクトファイルを開く

保存してあるプロジェクトファイルを開く場合は、以下の操作を行ってプロジェクトファイル(.armwpj)を選択してください。





エクスプローラでプロジェクトファイルをダブルクリックし、プロジェクトファイルを開くこともできます。 また、デスクトップにある本ソフトウェアのアイコンヘドラッグ&ドロップしてプロジェクトファイルを開くこともできます。



## 2.4. プローブの選択

本ソフトウェアで使用するプローブの選択を行います。

[プロジェクトの新規作成]ダイアログの フローフ ボタンをクリックします。

[プローブ選択]ダイアログに、現在接続されているプローブの一覧が表示されます。

プローブ選択		<b>—</b>
<u>シリアル番号</u> HM540000123	ステータス 接続可能	
HM540000124 HM540000125	使用中 う化以未登録	更新( <u>R</u> )
		<u>ライセンス登録(L)</u>
		閉じる(0)

Figure 13

使用するプローブをクリックし、 接続 ボタンをクリックします。

シリアル番号	プローブに登録されているシリアル番号を表示します。		
ステータス	プローブの状態を表示します。		
	接続可能	:	本ソフトウェアに接続し、使用することができます。
	使用中	:	他のアプリケーションで既に使用されています。
			本ソフトウェアで使用することはできません。
	ライセンス未登録	:	ライセンスが登録されていません。
			ライセンス登録 ボタンを押し、ライセンスを登録することで、使
			用可能になります。
接続	選択されているプローブ	に接続	売します。
	ステータスが「接続可能」のプローブを選択時のみ有効です。		
	接続後はダイアログが閉じ、 [プロジェクトの新規作成]ダイアログに戻ります。		
更新	現在接続されているプローブを再検索し、プローブの一覧を更新します。		
ライヤン7登録	ライセンスを登録する為のダイアログが表示されます。		7アログが表示されます。
	ステータスが【接続可能」もしくは「ライセンス未登録」のプローブを選択時のみ有効です。		
	→ 詳細は『2.5. License Code の登録』の章を参照ください。		
閉じる	このダイアログを閉じます。		

Table 3



## 2.5. License Code の登録

#### 本ソフトウェアはライセンスシステムを採用しています。

License Codeの取得には、別売りの Software Code が必要です。

Software Code とプローブの Serial No.を弊社にご連絡いただくと、License Code を発行します。

#### ①License Code の入力について

本ソフトウェア起動時、「プローブ選択]ダイアログに「ライセンス未登録」または、「接続可能」と表示されているものを選択し、 <sup>ライセンス登録</sup> ボタンをクリックすると、License Code の入力ウィンドウが表示されます。 弊社からお送りした License Code を入力していただくと、対象のプローブで本ソフトウェアを使用することができるようになります。

- ライセンスの登録
ライセンスの登録が必要です。 License Code を入力してください。
Serial No: HM540000035
License Code:
※ 半角英数字
※ 区切りのハイフン(ー)の有無は問いません
OK キャンセル

Figure 14

#### ②License Code 入力時の注意事項

License Code の入力は、お送りした情報通りに入力してください。



License Code を送付したメールの内容と一致しているにもかかわらず受け付けられない場合は、当社までお問い合わせください。



#### ③エラー表示一覧

Table 4

表示	意味	対処方法
License Code が不正です。	License Code の書式が不正など、 License Code の解析ができない場合 に表示されます。	License Code が送られてきたメールを ご確認の上、再度 License Code を入 力してください。
License Code と Serial No が一致 しません。 License Code と 本体の Serial No の組み合わせをご確認ください。	ご使用のプローブの Serial Noと、 License Code が対象としているプロー ブの Serial No が一致していない場合に 表示されます。	License Code が送られてきたメールを ご確認の上、再度 License Code を入 力してください。



## 2.6. CPU の選択

CPU 選択のドロップダウンリストから使用する CPU の種類を選択します。

CPUの選択		<b>-</b> ×-
CPU選択( <u>S</u> )	ARM7	•
	ОК	



次に、CPUの種類に応じた、より詳細なデバイスの選択を行います。 メーカー、シリーズを選択するとデバイスが絞り込まれるため、デバイスの選択が簡単になります。

デバイスの初算	明設定	<b>—</b> × <b>—</b>		
メーカー( <u>M</u> )	TOSHIBA	-		
ジリース*( <u>S</u> )	TMPM440	-		
デバイス( <u>D</u> )	TOSHIBA_TMPM440_TMPM440F10XBG (single chip mode)	-		
ファイル	TMPM440F10XBG_SingleChip_Setupfile.ini			
	参照	(B)		
┌接続方法(	の選択			
⊙ JTAG	(J) O SWD( <u>W</u> )			
 ターク <sup>s</sup> ット電	原電圧取得方法の選択			
<ul> <li>VTret</li> </ul>	端子からターゲットの電源電圧を取得する(⊻)			
○ VTref端子を無視する(N)				
JT4 3.3\ 古女郎	G通信/SWD通信を3.3Vの信号レヘルで行います。 なり電圧が低いターケットでは設定しないで下さい。ターケットシステム むる可能性があります。	が		
	0	ĸ		

Figure 16



Table 5		
接続方法の選択	ターゲットと接続するインターフェースを選択します。	
	JTAG :IEEE 1149.1 に準拠した信号インターフェースを使用します。	
	SWD :ARM にて定義されている 2 線式インターフェースを使用します。	
ターゲット電源電圧取得方法の選択	ターゲットとのインターフェース電圧をターゲットから取得するか選択します。	
	VTref 端子からターゲットの電源電圧を取得する :	
	ターゲットからインターフェース用の電源を取得します。1.2V~5.0V に追従します。	
	VTref 端子を無視する:	
	ターゲットからの電源を無視し、インターフェース電圧を 3.3V 固定にします。	



## 2.7. JTAG の設定

[JTAG 設定]ダイアログでは、ご使用のターゲットに合わせて JTAG クロックやデイジーチェーンなどの設定をします。

JTAGの設定	
┌JTAGの加ック周波数の指定───	
自動設定	•
低速加ックの周波数(1~ 1000KHz	) 500 💉 KHz
自動設定 : ターグットとの正常な通信/ 低い周波数を設定します	が確認できた最高周波数の一段階
ーディシーチェーン設定 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	เหล่า
○ JTAG がデイジーチェーン接続されて	13
「Shift IR の設定――	
総ビット要如 🔒	ビット位置 0
「バイパスレジスタの設定―――	
総七字ト要如 1	ビット位置 🕛 📑
ОК	キャンセル

Figure 17

JTAG に接続されているデバイスが1つの場合は、デフォルトのまま OK ボタンをクリックしてください。

Table 6

JTAG クロック周波数の指定	JTAG クロック周波数を選択します。
	自動設定にすると、周波数を自動判別し最適な周波数に設定されます。
	低速クロック指定を選択すると、KHz 単位で指定できます。(最大 1000KHz)
デイジーチェーン設定	JTAG のデイジーチェーン接続状態を指定します。
	具体的には TAP や DAP と呼ばれるものが複数あれば「デイジーチェーン接続されている」を選
	択します。
Shift IR の設定	総ビット数・ビット位置を指定します。
バイパスレジスタの設定	総ビット数・ビット位置を指定します。

「デイジーチェーン」、「Shift IR の設定」、「バイパスレジスタの設定」の詳細については次頁を参照ください。



#### ●デイジーチェーンの設定例 1

2つの CPU がデイジーチェーンした例で解説します。



Figure 18

#### ①上記の例で CPU1 のデバッグをする場合

Shift IR の設定の総ビット数は IR レジスタの合計で 10 になり、CPU1 のビット位置は 6 になります。 バイパスレジスタの設定の総ビット数はバイパスレジスタの合計で 2 になります。 CPU1 のバイパスレジスタのビット位置は 1 になります。

#### ②上記の例で CPU2 のデバッグをする場合

Shift IR の設定の総ビット数は IR レジスタの合計で 10 になり、CPU2 のビット位置は 0 になります。 バイパスレジスタの設定の総ビット数はバイパスレジスタの合計で 2 になります。 CPU2 のバイパスレジスタのビット位置は 0 になります。

#### ●デイジーチェーンの設定例 2

一つのデバイスでも、CPU に複数の TAP や DAP が含まれていて JTAG チェーンとして接続されている場合は「複数」とみなします。 以下の例は、デイジーチェーンされているものが 6 個あるということになります。



Figure 19



## 2.8. プローブ起動時の設定

JTAG コネクタの RTCK 信号とリセット信号について設定します。

また、本ソフトウェア起動時にターゲットをリセットするか強制ブレークするかの選択を行います。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → プローブ起動時の設定	

プローブ起動時の設定
□ RTCKを使用する(R)
リセット信号をアクティフキこする時間( <u>P</u> ) (リセットのハルス幅)(0 ~ 65535)
リセット信号をネケ <sup>、</sup> ート後待つ時間( <u>N</u> ) (0 ~ 65535)
ARM Writer起動時の動作設定(W)
ターケットをリセットして起動します
✓ コプロセッサに対してアクセスをする(C) (MMU、キャッシュ制御を行います。)
メモリアクセスエリア設定(A) メモリアクセス経路の設定は、メモリアクセスエリア設定 から行うことができます。
オプション設定(Q) オプション設定は、弊社からの指示など特別な 場合を除き、操作しないようにしてください。
ОК

Figure 20

通常は表示されたままで問題ありません。 設定が終了したら、 OK ボタンをクリックしてください。



## 2.9. CPU 動作モードの設定

ここではエンディアン、メモリアクセス時にアボート検出を行うかを設定します。

CPU動作モードの設定	
エンディアンの選択 C ビックエンディアン( <u>B</u> )	<ul> <li>• <u>য়৸য়৾য়৾য়৾য়য়য়৾য়য়৾য়য়৾য়য়৾য়য়৾য়য়য়৾য়য়৾য়য়৾য়য়৾য়য়৾</u></li></ul>
┌ メモリアクセス時にアボート検出を行う	
⊙ O <u>F</u> F	0 0 <u>N</u>
	OK

Figure 21

通常はデフォルトのまま OK ボタンをクリックします。

 $\rightarrow$ 

この設定は後で変更することができます。詳細は「<u>4.2.3. プローブ環境の設定</u>」の、③スイッチを参照ください。



## 2.10. フラッシュメモリ設定

書き込み対象のフラッシュメモリの設定を行います。

書き込みやクリアを行うために、まずは対象となるフラッシュメモリの設定が必要ですが、起動時に CPU を一覧から選択した場合は、 CPU の内蔵フラッシュメモリに対する設定は自動的に行われます。

以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → フラッシュメモリ → デバイス設定	

入力内容の詳細は「デバイス設定」タブを参照してください。

必要な設定項目への入力を確認したら、ダイアログ右上の 追加 ボタンを押下し、フラッシュメモリの設定を追加します。

Figure 22

フラッシュメモリの設定状況は、ダイアログ下部のリストで確認することができます。

	R;0×00000000	R;0×0001ffff	WM	MB9AFxx2(128KB)	RODO 4Byte R;0×1fffe080		
<u> </u>							
				n, buru			
E 8.	り加い時設定を	c無視する( <u>1</u> )		9-9849	K;UxlttteUUU - K;Uxltttt	***	

Figure 23



---

## 2.11. フラッシュメモリのクリア

フラッシュメモリにデータが書き込まれている場合、書き替え前にフラッシュメモリのクリアが必要です。 以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。 既にダイアログを開いている場合は、「フラッシュメモリのクリア」タブを選択してください。

ツールボタン メニューバーの操作 ショートカットキー

--- リソース → フラッシュメモリ → フラッシュメモリのクリア

フラッシュメモリ設定	×
デハバス設定       ワークメモリ       フロック情報       設定ファイルの保存/読み込み         フラッシュメモリのクリア       フラッシュメモリのフィルアッフ°       その他の設定         ウリア範囲の指定             ・ 澄録デハバスを全てクリア(G))             クリアアドレス(B)             アトドレス領域の設定             ・アトドレスを含むデドカドイス全体をクリア(N)             ・アトドレスを含むデドカビス全体をクリア(B)             デバカドイス全体をクリア	追加( <u>A</u> ) 肖ᆙ除( <u>D</u> ) 全削除( <u>L</u> ) 閉じる( <u>C</u> )
	¢°ל)ם-ኑ°(ײַ) ז₀プם-ኑ°(ײַ)

Figure 24

クリア範囲をいかのいずれかから選択し、開始 ボタンをクリックします。

#### ①登録デバイスをすべてクリア

登録しているデバイス全てをクリアします。

②アドレスを含むデバイス全体をクリア

<クリアアドレス>に入力したアドレスを含むデバイス全体をクリアします。

#### ③アドレスを含むブロックのみクリア

<クリアアドレス>に入力したアドレスを含むデバイスの1ブロックをクリアします。

→ 詳細は「<u>4.5.6. フラッシュメモリのクリアタブ</u>」を参照してください。



クリア処理中は進行状況が表示されます。

チップ消去中です		
アトシス 経過時間	0×00000000 3 秒	
	キャンセル	



#### クリア処理が終了すると、操作結果が表示されます。

// フラッシュメモリクリアを実行します // Device Clear : 0x00000000 FMCLEAR DEVICE CLEAR : 0x00000000 BLANK CHECK NOT SUPPORT // コマンド実行をします // 終了しました

Figure 26

#### 指定範囲が正常クリアされたかは、ダンプウィンドウで確認します。

📃 ቃ*ንፖንን	ント* ウ1 (	0×08	000	000	- 0x	080	000ff	(0x	100)	)												x
アドレス( <u>S</u> ):	0×080	00000	0			•	O f	877	ドレス	( <u>E</u> )	0x08	80000	)ff			-	アクセスサイズ	<u></u> χ"( <u>Α</u> ):	ロングワ	-h'	-	
テ°ータ( <u>D</u> ):	0×ff				X	✓	• 1	範囲	€( <u>L</u> )		0×10	00	_			•	Q 💡 🗆	更新	しない()	U)		
Address		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ļ	ASCI I			* *
0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000 0×080000	00 10 20 30 40 50 60 70 80 90 a0 b0	<b>ff</b> ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff ff	ff ff ff ff ff ff ff ff ff ff ff					*
0×080000 0×080000 0×080000 0×080000	c0 d0 e0 f0	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	ff ff ff ff	•	<b>₹</b>			<b>.</b>

Figure 27



## 2.12. オブジェクトデータの書き込み

設定したフラッシュメモリにオブジェクトデータをダウンロードします。 以下のメニューを選択し、[ダウンロード設定]ダイアログを表示します。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → ダウンロード	

[ダウンロード]ダイアログに書き込むモジュールの情報を設定し、ダウンロード」ボタンでをクリックします。 ダウンロード処理中は進行状況のダイアログを表示します。

22277 67	
ፖኑՆス	0×08002000
経過時間	4 秒
予想残り時間	47 秒
	キャンセル

```
Figure 28
```



処理対象のフラッシュメモリがブロック情報を使用しない設定の場合、書き込み状況を表示するダイアログは表示されません。

<u>ダンプウィンドウ</u>を表示して、フラッシュメモリにオブジェクトデータがダウンロードできているかを確認します。 ダウンロードできていない場合は、フラッシュメモリの設定状況を再度確認してください。

	うまく書き込めない場合は、
INFO	<ol> <li>フラッシュメモリのクリアを実施しているかご確認ください。</li> <li>[フラッシュメモリ設定]ダイアログにある、&lt;ダウンロード時に設定を無視する&gt;がチェックされていないかご確認ください。チェックされている場合は、フラッシュメモリに対しての書き込み処理を行いません。</li> </ol>



ショートカットキー

---

## 2.13. ワークメモリ設定

書き込み処理を高速化するために使用する、ワークメモリの設定を行います。 以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。 既に[フラッシュメモリ設定]ダイアログを表示している場合は、「ワークメモリ」タブを選択してください。

ツールボタン メニューバーの操作

--- リソース → フラッシュメモリ → ワークメモリ

フラッシュメモリ設定	×
フラッシュメモリのクリア フラッシュメモリのフィルアッフ <sup>®</sup> その他の設定 テジバイス設定 ワークメモリ フロック情報 設定ファイルの保存/読み込み	追加( <u>A</u> )
開始アドレス( <u>S</u> ) R;0×20000000 ▼ 範囲長( <u>R</u> ) 0×2000	閉じる( <u>c</u> )
ワークメモリの退避 フラッシュメモリへのアクセス時 「ワークメモリ内のデ <sup>×</sup> ータを退避 する(E)	ጵ°ታን□∽ኑ°(₩)
	アップロード( <u>U</u> )

Figure 29

指定したエリアが使用可能かどうかを調べるため、設定の前にメモリチェックを実行して読み書き可能であるか調べます。 アドレスを指定し、 メモリチェック ボタンから、指定エリアのメモリチェックを行います。 メモリチェックの実行結果はダイアログの右下に表示されます。



フラッシュメモリ設定	×
フラッシュメモリのクリア フラッシュメモリのフィルアッフ° その他の設定 デバイス設定 ワークメモリ フロック情報 設定ファイルの保存/読み込み	追加( <u>A</u> ) 問題(b)
エリア設定(共通で使用します)	
開始アドレス( <u>S</u> ) R;0×20000000 ▼ 新囲長(R) 0×2000	閉じる( <u>C</u> )
P-ケメモリの退避     [R:0×20000000 - R:0×20001fff ]     フラッシュメモリへのアクセス時     ワークがモリ内のデ <sup>*</sup> -タを退避     する(E)     メモリチェックを行います。     メモリチェックを行います。     メモリチェックを行います。     メモリチェックを行います。     マークがモリ内のデ <sup>*</sup> ータを退避     する(E)	<mark>結果</mark> タ <sup>°</sup> ワ <del>フローՒ (≝)</del> アッフ°ロード( <u>⊎</u> )

Figure 30

ワークメモリを更新 ボタンをクリックして設定範囲にワークエリアを更新します。更新されたワークエリアは、[フラッシュメモリ設定]ダイア ログ左下で確認することができます。

	R;0×08000000	R;0×0801ffff	WM		STM32F103×E	RODO 48	3yte R;0x20000080
E							設定したワークエリア
, □ ダウンロード時設定を無視する( <u>I</u> )						፦	0x20000000 - R;0x20001fff

Figure 31


# 2.14. ブロック情報を調べる

フラッシュメモリへの書き込み前に必要なクリア範囲を削減するため、設定中のフラッシュメモリのブロック情報を調査します。 ブロック情報を調査しておくと、必要なブロックだけをクリアすることができるため、フラッシュメモリの寿命を不必要に縮めたり、不必要 なメモリブロックのクリア処理で処理時間が長くなることを防ぎます。

以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。 既に[フラッシュメモリ設定]ダイアログを表示している場合は、「**ブロック情報**」タブを選択してください。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → フラッシュメモリ → ブロック情報	

- フラッシュメモリ設定	×
フラッシュメモリのクリア フラッシュメモリのフィルアッフ その他の設定 テバイス設定 ワークメモリ フロック情報 設定ファイルの保存/読み込み	追加( <u>A</u> )
75ッジュメモリのフ・ロック情報を調査する クリア操作を行い、設定済みのフラッシュメモリのフ・ロック情報 を調べます。 この情報はフラッシュメモリへの書込み操作時に使用され ます。	前除(D) 全削除(L) 閉じる( <u>C</u> )
取得したブロック情報 ブロック情報を必要とするデバイスが設定されていません	
	¢°ל)ם-ኑ°(עַ) ז₀フ°ם-ኑ°(עַ)

Figure 32

### ①ブロック情報の調査ボタン

<u>デバイス設定</u>をした後に「フロック情報の調査」ボタンをクリックし、設定したフラッシュメモリのブロック情報を調査します。 このとき、設定したフラッシュメモリにブロック情報を使用するものがない場合は、何も処理が行われません。 ブロック情報の調査は、フラッシュメモリのクリアを伴います。以下のメッセージボックスが表示されますので、開始する場合は「OK」 ボタンをクリックしてください。





Figure 33

ブロック情報の調査中は、Figure 34のような進行状況を示すダイアログを表示します。

フラッシュメモリのブロック配置を解析しています						
ፖኑՆス	0×08002800					
経過時間	2秒					
予想残り時間	26 秒					
	キャンセル					

Figure 34

### ②取得したブロック情報リスト

ブロック情報の調査が終了すると、「取得したブロック情報」の欄に結果が表示されます。

取得したブロック情報	
R;0x08000000 - R;0x080003ff	
R;0x08000400 - R;0x080007ff	
R;0x08000800 - R;0x08000bff	
R;0x08000c00 - R;0x08000fff	
R;0x08001000 - R;0x080013ff	
R;0x08001400 - R;0x080017ff	
R:0x08001800 - R:0x08001bff	
R:0x08001c00 - R:0x08001fff	
R;0×08002000 - R;0×080023ff	-

Figure 35



---

# 2.15. フラッシュメモリのフィルアップ

設定したフラッシュメモリに対してフィルアップ処理を行います。 以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。 既に[フラッシュメモリ設定]ダイアログを表示している場合は、「<mark>フラッシュメモリのフィルアップ</mark>」タブを選択してください。

ツールボタン メニューバーの操作 ショートカットキー

 リソース → フラッシュメモリ →	フィルアップ

フラッシュメモリ設定		×
デバイス設定 フラッシュメモリのクリア	ワークメモリ フロック情報 設定ファイルの保存/読み込み フラッシュメモリのフィルアッフ。 その他の設定	追加( <u>A</u> ) 問題( <u>A</u> )
- アドレス範囲の指定 - 開始アドレス(T)	□×08000000 ▼	全削除(L)
○ 終了アドレス( <u>E</u> )	0×000000ff	閉じる( <u>C</u> )
● 範囲長( <u>R</u> )	0×1000	
フィルアッフ°データ(1バイト  0×99	)( <u>B</u> )	
,		אָיָסטַם-⊦יֹ(שַ)
		アップロード( <u>U</u> )

Figure 36

### ①開始ボタン

フィルアップ範囲とフィルアップするデータを指定して 開始 ボタンをクリックすると、フィルアップ処理を開始します。 フィルアップ中は進行状況を示すダイアログを表示します。

フラッシュメモリにアクセスしています	
書き込み中です アドレス 0x08000000	
(二キャンセル」)	
Figure 37	



処理対象のフラッシュメモリがブロック情報を使用しない設定の場合は、進行状況を示すダイアログは表示されません。



フィルアップ処理が終了すると、ダイアログ右下に処理結果が表示されます。



Figure 38

### 指定範囲が正常にフィルアップされたか、 ダンプウィンドウで確認します。

💷 ቃ ኦንፖ ዕብ	■ ダンプウィンドウ1 0x08000000 - 0x080000ff (0x100)																			
アドレス( <u>S</u> ):	0×08	00000	00			•	0 #	冬了ア	ドレス	( <u>E</u> )	0×0	8000	Dff			-	アクセスサイスヾ( <u>/</u>	<ul> <li>D: ロングワード</li> </ul>	-	
テ°-५( <u>D</u> ):	0×99				X	✓	• 1	範囲長	€( <u>L</u> )		0×1	00	_	_	_	•	0 000 🗆 🗉	更新しない(山)		
Address		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII		
0x080000	000	<u>99</u>	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	<mark>-</mark> )劔劔劔劔劔	劒劒	-
0x080000	)10 120	99	99	99 00	99	99 00	99	99 aa	99 00	99	99 00	99 aa	99	99	99	99	99 99	动动动动动动	/劒劒 1/副/副	
0x080000	)30	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	动动动动动动	1999-999 1	
0×080000	)40	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	动动动动动	翩翩	
0×080000	)50	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	剑剑剑剑剑剑剑	劒劒  ふうふう	
0×080000	)60 )70	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	动动动动动动	1999999 161161	
0×080000	080	99	<u>99</u>	99	<u>99</u>	99	<u>99</u>	<u>99</u>	<u>99</u>	<u>99</u>	99	<u>99</u>	<u>99</u>	<u>99</u>	<u>99</u>	<u>99</u>	99	刻刻刻刻刻	·····································	_
		<u>00</u> .∢	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	An An An An An An An I	ianan D	-

Figure 39



---

# 2.16. フラッシュメモリ設定の保存

#### フラッシュメモリの設定をファイルへ保存します。

以下のメニューを選択し、[フラッシュメモリ設定]ダイアログを表示します。

既に[フラッシュメモリ設定]ダイアログを表示している場合は、「設定ファイルの保存/読み込み」タブを選択してください。

ツールボタン	メニューバーの操作	ショートカットキー

#### --- リソース → フラッシュメモリ → 設定ファイル

フラッシュメモリ設定	×
フラッシュメモリのフィルアップ       その他の設定         デバイス設定       ワークメモリ       フロック情報       設定ファイルの保存/読み込み         フラッシュメモリ設定ファイル       フラッシュメモリ設定ファイルの保存/読み込みを行います。       ファイルを開く(0)         設定ファイルを読み込んだ場合、設定済みのディータは 全て上書きされます。       上書き保存(Y)         名前を付けて保存(S)	追加( <u>A</u> ) 肖ᆙ除( <u>D</u> ) (全削除(L) 閉じる( <u>C</u> )
設定ファイル名: C:¥ARM_WRITER¥test.fsh	¢°ליטם-⊦°(ע)

Figure 40

### ファイルから開く

保存してあるフラッシュメモリの設定ファイルを読み込み、現在のプロジェクトで使用する<設定ファイル名>に登録します。

#### ②上書き保存

<設定ファイル名>で表示されているファイルに、現在の設定を上書き保存します。

#### ③名前を指定して保存

名前を指定して保存 ボタンをクリックし、表示されるファイルダイアログでフラッシュメモリの設定ファイルを保存します。 保存をすると、現在のプロジェクトで使用する<設定ファイル名>に登録します。



プロジェクトファイルを保存しても、フラッシュメモリの設定は保存されません。 必ずこの操作でフラッシュメモリの設定を保存してください。

#### ④<設定ファイル名>

現在のプロジェクトで使用する設定ファイルを表示します。

41 / 185 Universal Probe ソフトウェア・ユーザーズ・マニュアル - ARM Writer



# 2.17. プロジェクトファイルの保存

プロジェクトの設定状態をプロジェクトファイルへ保存することができます。 「プロジェクトの保存」を参照してください。

# 2.18. 本ソフトウェアの終了

本ソフトウェアを終了するには以下のメニューを選択するか、ウィンドウ右上隅の X ボタンをクリックします。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → アプリケーションの終了	

[アプリケーションの終了]ダイアログが表示され、終了の前にプロジェクトを保存するかどうかを選択します。



Figure 41

### ①プロジェクトに保存

作成したプロジェクトファイルを上書き保存して終了します。

### ②プロジェクトに名前をつけて保存

別のプロジェクトファイルとして保存して終了します。

### ③プロジェクトに保存しない

現在のプロジェクト設定状況を保存せずに終了します。



# 3. 本ソフトウェアでのデータ表現

本ソフトウェアでのデータ表現方法について説明します。

# 3.1. 数值表現

本ソフトウェアでは数値表現として2進数,10進数,16進数を扱う事ができます。

		Table 7
書式	基数	例
0x<数值>	16進数	0x12345678
H'<数值>	16進数	H'12345678
@<数值>	2 進数	@01011101
<数値>	10 進数	12346578

# 3.2. アドレス表現

本ソフトウェアでアドレスを表現する場合は、Table 8 で示す書式と、演算子を組み合わせたものになります。

Table 8			
書式	意味	備考	
<アドレス値>	論理アドレス		
mmu:<アドレス値>	論理アドレス		
r;<アドレス値>	物理アドレス		
INR:<アドレス値>	内蔵 I/O レジスタ空間		
APB:<アドレス値>	アクセス経路	CPU コアにメモリアクセス命令を実行させる方式。 APB 経路の論理アドレスでアクセスする。Cortex A/R 時のみ有 効。	
AHB:<アドレス値>	アクセス経路	メモリに直接アクセスする方式。 AHB経路の物理アドレスでアクセスする。MMUによるアドレス変換 は行なわれない。Cortex A/R 時のみ有効。	

# 3.3. データ表現

本ソフトウェアでデータを表現する場合は、Table 7 で示す書式と、演算子を組み合わせたものになります。 また、本ソフトウェアではバイト=1 バイト、ワード=2 バイト、ロングワード=4 バイトとして扱っています。



# 3.4. メモリ・I/O ポートの読み書き

コマンドウィンドウの ASSIGN またはドット(.)コマンドや、バッチプログラム内の条件式でメモリや I/O の内容を読み書きするには、以下のように指定します。

	Table 9
表記	意味
[アドレス式] または [アドレス式].B	指定されたアドレスのバイトデータ
[アドレス式].W	指定されたアドレスのワード(2 バイト)データ
[アドレス式].L	指定されたアドレスのロングワード(4 バイト)データ

#### ●例

通常のメモリ空間はアドレ	通常のメモリ空間はアドレスを入力するだけです。		
.[0x4000].B=0x10	//0x4000 番地へ 0x10 をバイトデータでライトする		
.[0x4000].W	//0x4000 番地のワードデータを参照する。		
.\$A=[0x4000].L	//ワーク変数\$A へ 0x4000 番地の 4 バイトデータを代入する		
if([0x4000].W==0x1234)	//0x4000 番地のワードデータが 0x1234 の時、真		
内蔵  /0 レジスタ空間は  NF	?:を付加して入力します。		
.[INR:0x4000].B=0x10	//0x4000 番地へ 0x10 をバイトデータでライトする		
.[INR:0x4000].W	//0x4000 番地のワードデータを参照する		



# 4. メニュー

本ソフトウェアのメニューについて説明します。

# 4.1. ファイル

「ファイル」メニューではプロジェクトに関するファイル操作を行います。

### 4.1.1. 閉じる

現在フォーカスしている子ウィンドウを閉じます。 子ウィンドウが無い場合は選択できません。

## 4.1.2. プロジェクトの新規作成

プロジェクトの新規作成を行います。 初めて使用する場合は、下記ツールボタンもしくはメニューを選択し、プロジェクトを作成してください。 プロジェクトはプロジェクトファイル(.armwpj)に保存されます。

ツールボタン	メニューバーの操作		ショートカットキー
₽ <u></u>	ファイル → プロジェクトの新規作成		Ctrl + P
	プロジェクトの新規作成		
	プロジェクト名(N):	OK キャンセル	

<u> </u>	
<u></u>	
	<u>+</u> +vtu 7 <sup>°</sup> I-7 <sup>°</sup> (P)

Figure 42



<b>T</b> 1		10	
lah		()	
iab	IC.	ΤU	

プロジェクト名	プロジェクト名を指定します。	
プローブ	プローブ種類 :プローブの種類	
	ターゲット : ターゲット名	
	ライセンス	
	Size-limited $\rightarrow$ 1KByte の書き込み容量制限版のライセンスです。	
	Available $\rightarrow$ 1KByte の書き込み容量制限の無いライセンスです。	
	[プローブの選択]ダイアログが表示され、現在接続されているプローブ一覧が表示されます。	
∫∐−∫(F)	使用するプローブを選択するにはプローブ名の部分をクリックし、OK ボタンをクリックします。	
位置	フルパスでセーブされるプロジェクトファイル名が表示されます。	
参照	[フォルダの選択]ダイアログが表示され、プロジェクトファイルをセーブするフォルダを選択します。	
ОК	プロジェクトファイルを作成します。	
キャンセル	プロジェクトファイルを作成を中止します。	



## 4.1.3. プロジェクトを開く

保存してあるプロジェクトファイルを開きます。前回の作業終了時の設定が復元されます。

プロジェクトを開いても、[フラッシュメモリ設定]ダイアログで設定した内容は復元されません。 「設定ファイルの保存/読み込み」タブで別途開く必要があります。

## 4.1.4. プロジェクトの保存

現在設定されている各種情報をプロジェクトファイルへ上書き保存します。





プロジェクトの保存を実行しても[フラッシュメモリ設定]ダイアログで設定した内容は保存されません。 [フラッシュメモリ設定]ダイアログの中で保存しておく必要があります。

# 4.1.5. プロジェクトを名前をつけて保存

現在設定されている各種情報を別のプロジェクトファイルへ保存します。





# 4.1.6. プロジェクトを閉じる

現在開いているプロジェクトを閉じます。 閉じる前に[アプリケーションの終了]ダイアログが表示されます。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → プロジェクトを閉じる	

# 4.1.7. アプリケーションの終了

本ソフトウェアを終了します。 詳細は「**本ソフトウェアの終了**」を参照してください。



# 4.2. リソース

「リソース」メニューではフラッシュメモリへの読み書きを行います。

### 4.2.1. ダウンロード

[ダウンロード設定]ダイアログでは書き込むモジュールを設定し、実際の書き込み指示を行います。 書き込むモジュールは、複数指定する事ができます。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース $ ightarrow$ ダウンロード	



Figure 43



モジュールリスト	ダウンロードするモジュールのリストです。
	Ctrl や Shift キーを押しながら選択すると、複数のファイルを選択できます。
フォーマット	ダウンロードするモジュールのオブジェクトフォーマットを選択します。
	デフォルトでは自動選択が選択されます。
	複数のモジュールを選択している場合、この項目はそれぞれのモジュールごとに指定する必
	要があります。
オブジェクトデータをダウンロード	オブジェクトデータをダウンロードする場合にチェックします。この項目はそれぞれのモジュールご
	とに指定する必要があります。
シンボル情報をダウンロード	本ソフトウェアでは使用できません。
次回起動時に再ダウンロード	プロジェクトファイルを開いた時に、自動的にモジュールをダウンロードします。
モジュール情報	モジュールのパス、ファイルサイズ、タイムスタンプが表示されます。
);e tu	ダウンロードするモジュールを追加します。
	[ファイルを開く]ダイアログが表示され、その中で追加するモジュールを選択します。
	追加されたモジュールはモジュールリストに表示されます。
削除	モジュールリストで選択されているモジュールをリストから削除します。
	選択されたモジュールをダウンロードします。
9'JJU-F	<オブジェクトデータをダウンロード>の項目にチェックを入れておく必要があります。
	モジュールリストで複数のモジュールが選択されている場合は、選択されたモジュール全てをダ
	ウンロードします。
ダウンロード前にチップイレースを自	このチェックボックスが ON の場合は、ダウンロード前にフラッシュメモリの内容を全てクリアしま
動実行する	す。クリアはダウンロードの操作1回につき1度だけ実行されます。
	複数のモジュールを選択してダウンロードする場合でもクリアは1度だけしか実行されません。

Table 11

(INFO)

オブジェクトデータのダウンロードでメモリアクセスに関するエラーが頻発する場合は、メニューの[プローブ環境の設定]を 選択して<ベリファイを有効にする>を設定してください。 <ベリファイを有効にする>を設定することにより、本ソフトウェアからのメモリライトが正しく書き込まれたかチェックを行うこ とが出来ます。標準設定では、ベリファイは無効になっています。

容量制限版でダウンロードできるオブジェクトデータのサイズは 1KByte までです。



## 4.2.2. アップロード

アップロードは、フラッシュメモリ上のデータを読み出し、各種フォーマットに変換してファイルに保存する機能です。 以下のメニューを選択すると[アップロード]のダイアログが開きます。 アップロードをするには[アップロード]ダイアログを表示して、ファイルに保存したいメモリ範囲とフォーマットを指定します。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース $ ightarrow$ アップロード	

アッフ°ロート*		×
フォーマット (M): ファイル名 (E): アップロード範囲の設定 開始アドレス ( <u>A</u> ): C 終了アドレス ( <u>E</u> ): で レングス ( <u>L</u> ):	∧°4†リテ°-୬	7ッ7泊ート(U) 閉じる
- PCアドルスの設定 ■ PCアドルスを指定 (P):		

Figure 44



		Table 12			
フォーマット	ファイルフォーマットは、以下の中から選択します。				
	フォーマット	メモリ範囲	PC アドレスの 指定	自動判別	備考
	インテル標準 HEX	0~64K バイト	可能	可能	
	インテル拡張 HEX	0~1M バイト	可能	可能	セグメント値が付 加されます。
	インテル 32 ビット HEX	0~4G バイト	可能	可能	
	モトローラ S 型 HEX(S1-S9)	0~64 <i>K</i> バイト	可能	可能	
	モトローラS型 HEX(S2-S8)	0~16M バイト	可能	可能	
	モトローラ S 型 HEX(S3 – S 7)	0~4G バイト	可能	可能	
	高速ダウンロード (SHF)	0~4G バイト	不可能	可能	当社独自のフォ ーマットです。
	バイナリ	0~4G バイト	不可能	不可能	バイナリデータとし てアップロードしま す。
		D欄の「可能」とは、本 るかどうかを指します。	ソフトウェアでダウン	ンロードを行う隊	祭、フォーマットを自
ファイル名	保存するファイル名を指定	定します。			
アップロード範囲の指定	開始アドレス	:開始アドレスを指定	言します。		
	終了アドレス	:終了アドレスを指定	ミします。		
	範囲長	:終了アドレス可範	囲長のどちらかを選	観して指定し	ます。
PC アドレスを指定	インテル HEX またはモト す。	ローラ HEX フォーマッ	トを選択した場合	は、 PC の開始	治アドレスを指定できま
アップロード	メモリのデータを読み出し	、指定したファイルにイ	呆存します。		



# 4.2.3. プローブ環境の設定

プローブ環境の設定を行います。

以下のメニューを選択すると[プローブ環境の設定]のダイアログが開きます。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → プローブ環境の設定	
		<b>—</b>
	The hole of the second second	

クロック メモリ   スイッチ   ホーリンケ   - SWDのクロック周波数の指定
10.00MHz
低速如妙の周波数(1~ 1000KHz) 500 ÷ KHz
自動設定 : ターゲットとの正常な通信が確認できた最高周波数の一段階低 い周波数を設定します
OK キャンセル

Figure 45



**①クロック** 

JTAG/SWD クロックおよびユーザークロックの周波数を設定します。 JTAG/SWD のエラーが発生する場合は周波数を下げてください。

プローブ環境の設定	×
クロック メモリ スイッチ ホーリング	
SWDの加ック周波数の指定	
10.00MHz	
低速加ックの周波数(1~ 1000KHz) 500 - KHz	
自動設定 : ターグットとの正常な通信が確認できた最高周波数の一段階低 い周波数を設定します	
OK キャンセ	UL I

Figure 46

Tal	bl	е	1	3
iui		<u> </u>	-	-

JTAG のクロック周波数の指定	JTAG のクロック周波数を選択します。
	※低速クロックの周波数は 1~1000KHz までの間で指定できます。



②メモリ

本ソフトウェアでのメモリ操作に関する基本設定を行います。

2 <sup>°</sup> ローフ <sup>™</sup> 環境の設定 <sup>クロック</sup> メモリ スイッチ メモリライト時のヘリファイ □ <u>ヘリファイを有効</u>	:  ポーリンゲ   の設定 こする(V)		×
テラォルトのメモリアクセス	サイズの設定 ―――		
○ バイト( <u>B</u> )	© 7−ŀ( <u>W</u> )	<ul> <li>ロングワード(L)</li> </ul>	
MMU アトシス変換プロトコル	の指定(12)		
I			<u> </u>
		OK 🛸	キャンセル

Figure 47

### ●メモリライト時のベリファイの設定

Table 14

ベリファイを有効にする 本ソフトウェアがメモリをライトする時に、正しく書き込まれたかを再度リードしてチェックします。 正しく書き込まれなかった場合には、メッセージが表示されます。

### ●デフォルトのメモリアクセスサイズの設定

Table 15

バイト	本ソフトウェアがメモリをアクセスする時のデフォルトのメモリアクセスサイズを選択します。
ワード	
ロングワード	



ここで設定された値は、ダンプウィンドウなどでメモリアクセスする時に使用されます。



# ③スイッチ

ターゲット CPU の設定をします。

プローブ環境の設定		×
クロック メモリ スイッチ ホペーリング		1
- エンディアンの)選択	<ul> <li>IKITY-279711</li> </ul>	
┌ メモリアクセス時にアボート検出を行う──		
	с о <u>н</u>	
	OK ++>>t	211

Figure 48

т.	-		1	$\boldsymbol{c}$
Ia	וט	ie.	T	0
-	-	-		-

エンディアンの選択	ビッグエンディアン:命令解析などをビッグエンディアンで行います。	
	リトルエンディアン:命令解析などをリトルエンディアンで行います。	
メモリアクセス時のアボートチェック	OFF : メモリアクセス時にアボートチェックを行いません。	
	ON :メモリアクセス時にアボートチェックを行います。	



# ④ポーリング

ターゲットの状態を監視するためのポーリング間隔を設定します。

7゚ローブ環境の設定	×
ካਧック XEU Zイッチ ホペーリング	1
- ホペーリング間隔の設定	
間隔( <u>E</u> ): 00ms	
□ 既定値とする(S)	
	テウォルト値に戻す( <u>R</u> )
	既定値に戻す(T)
	OK キャンセル

Figure 49

Τ-	L - 1	-	1	7
Ia	DI	e	1	/
		-	_	

間隔	ポーリング間隔を 100ms ごとに設定にします。
	100ms から 4,294,967,200ms までが有効範囲です。
既定値とする	指定した値を既定値とし、以後、新規作成したプロジェクトはこの値になります。
デフォルト値に戻す	出荷時の値(500ms)に戻します。
既定値に戻す	既定値として設定した値に戻します。



# 4.2.4. バッチファイルの自動実行設定

### バッチプログラムを自動実行するタイミングを指定します。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → バッチファイルの自動実行設定	

バッチファイルの自動実行設定	×
□ 起動時(S):プロジェクトの読み込み前 (ProjectNamesta)	
🔲 起動時( <u>F</u> ):プロジェクトの読み込み後 (ProjectName.fst)	
□ リセット時( <u>R</u> )(ProjectNamerst)	
□ CPU実行前( <u>B</u> )(ProjectNamebfr)	
□ CPUブレーウf後(A)(ProjectName.afr)	
🗌 ターゲットのPowerOn検出時( <u>0</u> )(ProjectName.pon)	
🗌 ターゲットのPowerOff検出時( <u>P</u> )(ProjectName.pof)	
□ ブレーウポイント変更前(K)(ProjectNamebps)	
□ フラッシュメモリ書き込み・消去前(E)(ProjectName bflash)	
□ フラッシュメモリ書き込み・消去後(①(ProjectName.aflash)	
(	

Figure 50



	Table 18
起動時	本ソフトウェア起動時(プロジェクトの内容を読み込む前)にバッチファイルを実行します。
(プロジェクトの読み込み前)	これにより、各種設定及びウィンドウの情報などを読み込む前に、必要な処理をバッチで実
	行でき、不正なメモリなどのアクセスを回避できます。
	バッチファイル名 : {プロジェクト名}.sta
起動時	本ソフトウェア起動時(プロジェクトの内容を読み込んだ後)にバッチファイルを実行します。
(プロジェクトの読み込み後)	バッチファイル名 : {プロジェクト名}.fst
リセット時	リセットボタンを押した時にバッチファイルを実行します。
	バッチファイル名 : {プロジェクト名}.rst
CPU 実行前	CPU 実行前にバッチファイルを実行します。
	バッチファイル名 : {プロジェクト名}.bfr
CPU ブレーク後	CPU ブレーク後にバッチファイルを実行します。
	バッチファイル名 : {プロジェクト名}.afr
ターゲット電源 ON 時	ターゲット電源 ON 時にバッチファイルを実行します。
	バッチファイル名 : {プロジェクト名}.pon
ターゲット電源 OFF 時	ターゲット電源 OFF 時にバッチファイルを実行します。
	バッチファイル名 : {プロジェクト名}.pof

ツールバーのボタンを押すことにより以下の名前のバッチファイルを実行します。 バッチファイル名 : {プロジェクト名}.wrt

ツールボタン	メニューバーの操作	ショートカットキー
TTE		



バッチファイルはプロジェクトファイルと同じフォルダに保存してください。



## 4.2.5. プローブ起動時の設定

JTAG インターフェースの RTCK 信号とリセット信号について設定します。 また、本ソフトウェア起動時にターゲットをリセットするか強制ブレークするかの選択を行います。 以下のメニューを選択し、[プローブ起動時の設定]ダイアログを表示します。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → プローブ起動時の設定	

プロ-ブ起動時の設定	x
■ RTCKを使用する( <u>R</u> )	
リセット信号をアクティフリこする時間(P) (リセットのハッルス幅)(0 ~ 65535)	
リセット信号をネケート後待つ時間( <u>N</u> ) (0 ~ 65535) 0 ・ ms	
ARM Writer起動時の動作設定(W)	
ターケットをリセットして起動します	
✓ コプロセッサに対してアクセスをする(C) (MMU、キャッシュ制御を行います。)	
メモリアクセスエリア設定(A) メモリアクセス経路の設定は、メモリアクセスエリア設定 から行うことができます。	£
オフジョン設定(Q) オフジョン設定は、弊社からの指示など特別な 場合を除き、操作しないようにしてください。	î
ОК	

Figure 51

設定が終了したら、 OK ボタンをクリックします。

#### RTCK を使用する

JTAG インターフェースの RTCK 信号の機能を使用しているかどうかを設定します。 RTCK (Return TCK)信号は、ターゲットからの出力信号です。 チェックをするとプローブは RTCK 信号を使用した JTAG 制御を行います。 RTCK を使用するかはプローブが自動的に判別しますが、チェックされた設定で動作しない場合は、チェックを外して下さい。



ターゲットの仕様と逆の設定をすると、本ソフトウェアが正常に起動しませんのでご注意ください。



#### Table 19

チェック状態	RTCK 信号の機能を使用します。
非チェック状態	RTCK 信号の機能を使用しません。

#### ②リセット信号をアクティブにする時間

本ソフトウェアでリセットを実行した時に、JTAG インターフェースのリセット信号をアサートする時間を設定します。デフォルトは100ms です。リセット信号をアサートにしてはいけない場合は 0ms に設定してください。

#### ③リセット信号をネゲート後待つ時間

リセット信号のアクティブ時間終了から JTAG 制御を開始するまでの時間を設定します。

デフォルトは Oms です。

JTAG のリセット信号を受けてターゲットが CPU に対するリセット信号を解除するまでの時間より長く設定する必要があります。この 設定が解除までの時間より短いと、CPU がリセット状態の時に本ソフトウェアが JTAG 制御を行ってしまい、正しく制御ができないた め、正常に起動しません。



Figure 52



#### ④本ソフトウェア起動時にターゲットをリセットする

本ソフトウェアの起動時にターゲットをリセットするか、強制ブレークするかを選択します。 この設定はプロジェクトファイルに保存されるので、次回同じプロジェクトファイルで本ソフトウェアを起動した際には設定が復元されま す。

また、この設定は本ソフトウェアの起動オプションでも設定することができます。本ソフトウェアの起動オプションは、プロジェクトファイル に保存されている設定よりも優先度が高いため、「プロジェクトファイルの設定値は利用したいが、起動時の動作だけ一時的に変え たい」というような場合に便利です。

	Table 20
ターゲットをリセットして起動します	起動時にリセットします。(デフォルト)
ターゲットをブレークして起動します	起動時に強制ブレークします。
初期化のみを行い起動します	起動時に初期化のみを行い、リセット・強制ブレークは行いません。

本ソフトウェアはデフォルトの状態で起動すると起動時にターゲットをリセットします。

しかし、たとえばプローブとケーブルを接続していない、動作中のターゲットCPUを調べたいとき(ターゲットの動作状態を維持したまま プログラムを途中からデバッグしたいとき)は、通常の起動を行うと起動時にリセットしてしまうため、これらのオプションでリセットしない ようにします。



ただし、ターゲットがブレーク状態のときに本ソフトウェアを「起動時にブレークする」モードで起動した場合、本ソフトウェア はターゲットからレジスタなどの情報を正しく取得できない場合がありますのでご注意ください。(この場合でも、ターゲット をリセットすれば、それ以降正常にデバッグできるようになります。) 「起動時にブレークする」は、実行状態のターゲットに対して有効です。

### ⑤コプロセッサに対してアクセスをする

コプロセッサに対してアクセスをするかを選択します。

本ソフトウェアがコプロセッサに対してアクセスを行い、自動で MMU やキャッシュの制御をします。しかし、ターゲットによってはこれらに アクセスすると正しく本ソフトウェアが起動しないものがあり、そのような場合はチェックを外してください。

Table 21

チェック状態	コプロセッサにアクセスします。(デフォルト)
非チェック状態	コプロセッサにアクセスしません。

#### ⑥メモリアクセスエリア設定

このボタンをクリックすると[メモリアクセスエリア設定]ダイアログが開きます。 本ソフトウェアからターゲット上のメモリアクセス方法やアクセス制限するエリアを設定することができます。



## 4.2.6. メモリアクセスエリア設定

ターゲットのメモリ空間上で以下のような理由によって、本ソフトウェアからターゲットのメモリへのアクセス方法やアクセスを禁止するエリアを設定するためのダイアログです。

- メモリが存在しないアドレス空間へのアクセスを制限する。
- 初期化されていないメモリ空間でアクセスできない。
- メモリへのアクセス経路(APB/AHB)を変更してアクセスの速度を上げる。



エリアは物理アドレスのみ設定できます。

モリアクセスエリアの設定				<b>-X</b>	
エリア設定 ———					
開始アト・レス( <u>S</u> )	): R;0×00000000	•	設定(1)		
● 終了アドレス( <u>E</u> )	R:0xffffffff	•	初期化(I)		
○ 範囲長(L):	0×100000000			_	
☑ リート許可( <u>F</u>	り 🔽 ライト許可()	w)			
C APB( <u>P</u> )					
		 %=+11 z +			
АРВ:СРОЦЛ АНВ:火刑に直	_メモリアリセス中学を美 接アクセスする方式で	ध⊺∂ए२/ इ.	11/1090		
ѵ҄҄҄҄҄҄҄Ҥӈҏҏҏӡӷ					
間始かり	終了アドレフ	U N - 5	小怒路		
R;0x00000000	R;Oxffffffff	許可意	F可 AHB		
	[ 	~~~	و بر ب		

Figure 53



		Table 22					
	開始アドレス	設定するエリアの開始アドレスを設定します。					
	終了アドレス	設定するエリアの終了アドレスを設定します。					
	新田트	設定するエリアの範囲長を設定します。					
	戰四天	ラジオボタンにより、終了アドレスと範囲長のどちらかが選択できます。					
		メモリリードとメモリライトのアクセス禁止/許可を設定します。					
エリア設定	アクセス属性	リード許可 :メモリリードを許可します。					
		ライト許可 : メモリライトを許可します。					
工力成定		メモリアクセスの処理方法を選択します。					
	経路	APB : CPU にメモリアクセス命令を実行させる処理です。					
		AHB : CPUを介さずに直接アクセスする方法です。					
	設定	設定ボタンを押すことによって、設定した内容をメモリアクセスエリアリストに追加します。					
	初期化	メモリアクセスエリアリストの設定状態を初期化します。					
		メモリアクセスエリアの設定が表示されます。					
		表示は左から					
		• 開始アドレス					
		• 終了アドレス(または範囲長)					
メモリアクセスエリ	アリスト	• リード処理の状態(許可/禁止)					
		<ul> <li>ライト処理の状態(許可/禁止)</li> </ul>					
		<ul> <li>アクセス経路の状態(APB/AHB)</li> </ul>					
		終了アドレスと範囲長の表示は、[エリア設定]のラジオボタンの設定、又はヘッダーの終					
		了アドレスとレングスの部分をクリックすることで切り替わります。					
ОК		設定を有効にして、ダイアログを閉じます。					
キャンセル		本ダイアログを開いてから行った操作を全てキャンセルし、ダイアログを閉じます。					



アドレス( <u>S</u> ):	R:0×200	00000		•	0	終了)	ንት レス	( <u>E</u> )	R;0x	2000	)00ff			-	アクセスサイス"( <u>A</u> ):	<ul> <li>ロングワート*</li> </ul>	
デ─タ( <u>D</u> ):	0×00			X	/ •	範囲	€(L)		0×10	)0				•	Q 🛛 🗆 更新	新しない(旦)	
Address	+	0 +1	+2	+3 -	+4 +5	5 +6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII	
R;0×2000		00	00	00 0		00	08	00	10	00	20	00	10	00	00		•••
R;0x2000	10020 0	0 00 0 be	00	20 (	)0 00	00	00	00	00	02	00	00	00	00	00		•••
R;0×2000	0030	? ??	??	?? 1	?? ??	???	??	??	??	??	??	??	??	??	??	22222222222	???
R;0x2000	0040 ?	? ?? 2 22	??	?? { 22 0	?? ?? >> >>	2 ??	??	??	??	??	??	??	??	??	?? 22	??????????????	??? >>>
R;0x2000	0060 2	? ??	??	?? 1	22 22	??	??	??	??	??	??	??	??	??	??	???????????????????????????????????????	???
		<u>î</u> ^^	~~	~~ /	<u></u>		~~	~~	~~	~~	^^	~~	~~	~~	^^		~~~

# 4.2.7. メモリ/ポート

→ 詳細は「4.4. メモリ/ポート」を参照ください。

# 4.2.8. フラッシュメモリ

→ 詳細は「4.5. フラッシュメモリ」を参照ください。



# 4.3. 実行

「実行」メニューでは、ターゲットの CPU の実行状態をコントロールします。

### 4.3.1. 実行開始

現在の PC からプログラムの実行を開始します。 ブレークするまで CPU を実行します。



4.3.2. ブレーク

実行中のプログラムを強制的にブレークし、本ソフトウェアに制御を戻します。 無限ループ状態でも強制ブレークが可能ですが、CPUの暴走状態では強制ブレークできないことがあります。



4.3.3. リセット

ターゲットをリセットします。

CPU 上でプログラムが実行中の場合、プログラムの実行はリセットされ、リセット番地で停止します。

リセットはプローブの SRST 信号をアサートするため、プローブの SRST 信号がターゲットに接続されており、ターゲット側の回路が CPU をリセットできるようになっている必要があります。 SRST 信号が接続されていない場合、リセットはできません。

「4.2.5. プローブ起動時の設定」の②リセット信号をアサートする時間が 0ms の場合は、SRST 信号をアサートせず、PC をリセット番地に書き換えるだけの動作になります。

メニューバーの操作	ショートカットキー
実行 → リセット	
	メニューバーの操作 実行 → リセット



## 4.3.4. キャンセル

ターゲットの動作異常や、アクセス出来ない領域へアクセスしてしまったなど、予期しない動作や不正な処理が行われた時にホスト PC(+プローブ)と、ターゲット間の通信が不能になる事があります。

本コマンドをを実行する事で通信不能状態から復帰する場合がありますが、正しく制御できないことが多いため、復帰できた場合は、一旦プロジェクトファイルを保存し、すべてを再起動してください。

ツールボタン	メニューバーの操作	ショートカットキー
	実行 → キャンセル	

## 4.3.5. ベリファイ(ライト時に比較)

このチェックをして書き込みを実行すると、書き込み後にベリファイを行います。

ツールボタン	メニューバーの操作	ショートカットキー
Vfy	実行 → ベリファイ(ライト時に比較)	

# 4.3.6. ベリファイオンリー(ライトせずに比較)

このチェックをして書き込みを実行すると、書き込みは行わずにベリファイのみを行います。

ツールボタン	メニューバーの操作	ショートカットキー
Vfy only	実行 → ベリファイオンリー(ライトせずに比較)	



## 4.3.7. チェックサム値の計算

ターゲットのフラッシュメモリに正しくモジュールが書き込めたかを、書き込みモジュールのチェックサムとフラッシュメモリのデータのチェック サムを計算し、比較することができます。

	Table 23
書き込みデータのチェックサム	フラッシュメモリに書き込むデータのチェックサムを表示します。
	書き込み手順が記録されていない場合は、何も表示されません。
ターゲット上のデータのチェックサム	フラッシュメモリに書き込まれているデータのチェックサムを表示します。
	書き込み手順が記録されていない場合は、何も表示されません。
範囲	フラッシュメモリに書き込むデータのアドレス範囲を表示します。
	書き込み手順が記録されていない場合、"None"を表示します。
計質	フラッシュメモリに書き込むデータと、フラッシュメモリに書き込まれているデータのチェックサムを
前昇	計算します。書き込み手順が記録されていない場合、"None"を表示します。



本機能を使用するには予めプローブに書き込み手順を記録しておく必要があります。 書き込み手順の記録については「9.スタンドアロン機能」を参照してください。



# 4.4. メモリ/ポート

「メモリ/ポート」メニューではメモリに対する処理を行います。

4.4.1. メモリのフィルアップ

指定範囲のメモリを指定されたデータでフィルアップします。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → メモリのフィルアップ	

メモリフィルアッフ。	<b>—</b>
アト <sup>*</sup> レス範囲の設定 開始アト <sup>*</sup> レス( <u>S</u> ) 0×20000000 ▼ ○ 終了アト <sup>*</sup> レス(E) ▼ ○ 範囲長( <u>B</u> ) 0×1000	フィルアッフ <sup>*</sup> テ <sup>*</sup> ータの設定 フィルアッフ <sup>*</sup> テ <sup>*</sup> ータ( <u>D</u> ) <sup>0×5a</sup> テ <sup>*</sup> ータサイス <sup>*</sup> の設定 ● ハ <sup>*</sup> イト ○ ロンウ <sup>*</sup> ワート <sup>*</sup> ○ ワート <sup>*</sup> ○ ウワット <sup>*</sup> ワート <sup>*</sup> ( <u>Q</u> )
ሃቺሃጆጛセスサイズ(₩)	終了しました 開始 閉じる

Figure 54

### ①アドレス範囲の設定

フィルアップするアドレス範囲を設定します。

Table 24

開始アドレス	開始アドレスを指定します。
終了アドレス	終了アドレスを設定します。
範囲長	範囲長を指定します。

#### ②フィルアップデータの設定

	Table 25
フィルアップデータ	フィルアップするデータを指定します。
データサイズの設定	フィルアップするデータのサイズを指定します。 例 1 )
	ロングワードを選択してフィルアップデータを「0x5A」と指定すると、「0x0000005A」というデータでフ ィルアップ処理を行います。
	例 2) バイトを選択してフィルアップデータを「0x1234」と指定すると、「0x34」でフィルアップ処理を行いま
	<i>y</i> •



#### ③メモリアクセスサイズ

フィルアップするときのアクセスサイズを指定します。 デフォルトは[プローブ環境の設定]ダイアログの、「メモリ」タブにある、「デフォルトのメモリアクセスサイズの設定」で設定します。

#### ④メモリフィルアップ結果

メモリフィルアップの結果を表示します。 指定されたメモリ範囲が大きいとき、以下のダイアログが表示されることがあります。 あまり長い時間終了しない場合は、キャンセルできます。

プローブ応答待ち	
経過時間	[#43:S]
	00:00:01
	キャンセル

Figure 55

実行結果は、コマンドウィンドウにも表示されます。

יאלאכי אלא איז ארא איז ארא איז איז ארא איז איז איז איז איז איז איז איז איז אי	×			
// フィルアップを実行します // (0×20000000 - 0×20000fff ロングワードアクセス 0x5a) FILL 0×20000000 TO 0×20000fff ASIZE LWORD DATA 0×5a BYTE // 終了しました				
  BREAK             LOG-OFF  MPCore DISABLE  No Sync パッヂファイルの実行	Ŧ			
<pre>&lt;&lt; &gt; &gt;&gt; BATCH BREAK CD CHECK CLOSE CLS</pre>				

Figure 56



### 4.4.2. メモリチェック

 $\langle ! \rangle$ 

指定されたメモリアクセスサイズで指定範囲のメモリ領域のリード/ライトを行い、メモリをチェックします。 リード/ライトするデータはランダムです。 エラーが発見された場合、その時のエラー発生アドレス、ライトデータ、リードデータを表示します。 なお、エラーの数が 100 個に達するとメモリチェックを中止します。

ツールボタン	メニューバーの操作	ショートカットキー
リソース $ ightarrow$ メモリ/ポート $ ightarrow$ メモリチェック		

チェックを行ったメモリ領域内のデータは、全てチェック用データで上書きされますので注意してください。

メモリチェック	<b>X</b>
アトットン範囲の設定	Xモリチェック結果
開始アドレス( <u>S</u> ) 0×20000000 👻	<u> ንት፣ አ. 5/ት አ</u>
○ 終了アドレス(E)	
● 範囲長(L) 0×1000	
›››››››››››››››››››››››››››››››››››››	
テ°∽ݡサイズ(₫) パイト	
開始開じる	

Figure 57

#### ①アドレス範囲の設定

メモリチェックするアドレス範囲を設定します。

Tabl	е	26
10101	~	20

開始アドレス	開始アドレスを指定します。
終了アドレス	終了アドレスを設定します。
範囲長	範囲長を指定します。

#### ②メモリアクセスサイズ

メモリチェックするときのアクセスサイズを指定します。 デフォルトは[<mark>フローブ環境の設定</mark>]ダイアログの、「メモリ」タブにある、「デフォルトのメモリアクセスサイズの設定」で設定します。



③データサイズ

メモリチェックを行うデータサイズを選択します。

#### ④メモリチェック結果

メモリチェックの結果を表示します。 指定されたメモリ範囲が大きいとき、以下のダイアログが表示されることがあります。 あまり長い時間終了しない場合は、キャンセルできます。

プローブ応答待ち		
経過時間	[#43:S]	
00:00:01		
	キャンセル	

Figure 58

実行結果は、コマンドウィンドウにも表示されます。

273454 7458	×		
// メモリ検査を実行します // (0×20000000 - 0×20000fff ロングワードアクセス) CHECK 0×20000000 TO 0×20000fff ASIZE LWORD SIZE BYTE			
// 終了しました	-		
BREAK I LOG-OFF MPCore DISABLE No Sync バッチファイルの実行			
$\geq$			
K K K K K K K K K K K K K K K K K K K			

Figure 59


#### 4.4.3. メモリサーチ

指定範囲のメモリ内を検索し、指定データと一致する、または一致しないデータのアドレスを表示します。 検索するデータは、数値もしくはアスキー文字列が指定できます。

ツールボタン	メニューバーの操作		ショートカットキー
	リソース $ ightarrow$ メモリ/ポート $ ightarrow$ メモリサーチ		
	xモリサーチ	×	
	「7トジス範囲の設定		

	2 2 mmers
開始アドレス( <u>S</u> ) 0×20000000 👻	開始しました 0×200000ff
○ 終了フドレス(E)	0x200001fe 0x200002fd
● 範囲長( <u>R</u> ) 0×1000	0×2000031C 0×200004fb 0×200005fa
メモリアクセスサイス <sup>*</sup> (M) □ンク <sup>*</sup> ワート <sup>*</sup>	0×200006f9 0×200007f8
	0x200008f7 0x200009f6
サーチテ <sup>ッ</sup> ータ( <u>D</u> ) <sup>0</sup> ×ff	0x20000af5 0x20000bf4
┌データサイズの設定―――	0x20000df2
💿 ስ* ተト 🛛 🔿 ロンク* ワート*	0x20000ef1 0x20000ff0
C ワード C ウワッドワード( <u>Q</u> )	終了しました
開始 閉じる	

Figure 60

#### ①アドレス範囲の設定

メモリチェックするアドレス範囲を設定します。

Table 27

開始アドレス	開始アドレスを指定します。
終了アドレス	終了アドレスを設定します。
範囲長	範囲長を指定します。

#### ②メモリアクセスサイズ

メモリサーチするときのアクセスサイズを指定します。 デフォルトは[<mark>プローブ環境の設定</mark>]ダイアログの、「メモリ」タブにある、「デフォルトのメモリアクセスサイズの設定」で設定します。



#### ③サーチデータの設定

Table 28

サーチデータ	サーチするデータを指定します。		
	不一致データを指定する場合は、データの先頭へエクスクラメーションマーク(!)を付けます。		
	文字列の不一致条件は指定できません。		
	数値での指定 : 0x34, 128, !0x56 など		
	アスキー文字列 : "abcdef", "ghijk" など		
データサイズの設定	サーチするデータのサイズを指定します。		

#### ④サーチ結果

メモリサーチで検索されたアドレスを表示します。 実行結果は、コマンドウィンドウにも表示されます。

ባረት እስለ እስለ እስለ እስለ እስለ እስለ እስለ እስለ እስለ እስለ	×
// (0x20000000 - 0x20000fff ロンク`ワート`アクセス 0xff)	-
SEARCH 0x20000000 TO 0x20000fff ASIZE LWORD MODE MATCH BYTE 0xff	
// 0×200000ff	
// UX200001fe	
// 0x2000031C	
// 0x200005fa	
// 0x200006f9	
// 0x200007f8	
// 0x200008t7	
// UX20000916 // 0x20000-f5	
$// 0 \times 20000 a 13$	
// 0x20000cf3	
// 0x20000df2	
// 0x20000ef1	=
// 0x20000110	
// 終了しました	
	-
DREAK	
STATUT BATUT BREAK UD CHEUK	

Figure 61



#### 4.4.4. メモリコピー

#### 指定範囲のメモリコピーを行います。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → メモリコピー	

Xモリ転送			
- アドレス範囲の設定	- 転送メモリの設定		
開始アドレス( <u>A</u> )	C Iミュレーションメモリ∗ からユーサ*メモリ(I)		
○ 終了アドレス( <u>E</u> )	C 2-ザンモリからエミュレーションメモリ* (U)		
○ 範囲長(L)	<ul> <li>● メモリコヒ°~(0) 転送アトドレス(<u>R</u>)</li> </ul>		
*注意:エミュレーションメモリは予めマッピングしておく必要があります。			
›››››››››››››››››››››››››››››››››››››	メモリ車転送		
	開始 閉じる		

Figure 62

#### ①アドレス範囲の設定

メモリチェックするアドレス範囲を設定します。

Table 29			
開始アドレス	開始アドレスを指定します。		
終了アドレス	終了アドレスを設定します。		
範囲長	範囲長を指定します。		

#### ②転送メモリの設定

Table 30			
エミュレーションメモリからユーザーメモリ	本ソフトウェアでは使用できません。		
ユーザーメモリからエミュレーションメモリ	本ソフトウェアでは使用できません。		
メモリコピー	<転送アドレス>に転送先の開始アドレスを指定します。		

#### ③メモリアクセスサイズ

メモリサーチするときのアクセスサイズを指定します。 デフォルトは[<mark>フローブ環境の設定</mark>]ダイアログの、「メモリ」タブにある、「デフォルトのメモリアクセスサイズの設定」で設定します。



### 実行結果は、コマンドウィンドウにも表示されます。

יאלאכי אלאכי איז איז איז איז איז איז איז איז איז אי	x
// 終了しました	-
// メモリ転送を実行します // (0x20001000 - 0x20001fff ロングワードアクセス -> 0x20001000) COPY 0x20001000 TO 0x20001fff ASIZE LWORD MEM ADDR 0x20001000 // 終了しました	
BREAK I LOG-OFF MPCore DISABLE No Sync 画面消去	_
<pre> CLS COPY CPRREAD CPRWRITE</pre>	

Figure 63



## 4.5. フラッシュメモリ

## 4.5.1. フラッシュメモリ設定ダイアログ

フラッシュメモリ関連の操作を行うダイアログです。 ダイアログ左上部のタブを切り替えることで、以下の7つの操作を行うことができます。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → フラッシュメモリ → デバイス設定	
1) フラッ 2) ワーク 3) ブロッ 4) 設定 5) フラッ 6) フラッ 7) そのイ	シュメモリの追加 ウメモリの操作 ック情報の取得および表示 こファイルの保存/読み込み シュメモリのクリア シュメモリのフィルアップ 地の設定	の追加操作お上び削除・
タノを1	切り替えることで「こうの操作を行うことか」 す。 り り り の 深作を行うことか 削除を行います。	
2	ラッシュメモリ設定	×
1	フラッシュメモリのクリア フラッシュメモリのフィルアッフ。 その他の設定 通知であ	
	デパペスの種類(I) <sup>メールー(</sup> M) CUSIOM ▼ 対ム・(2) その他 ▼ 全削除(L)	
	Access_tor_my_kead/write_program	
	開始がい(S) R;0x08000000 ▼ P 10 00000000000000000000000000	
	~_у-ド/ライト プログラム(使用タイプ	
	リードライト用プログラム処理開始アドレス(Ⴒ) R;0×20000080 ▼ アップロード/ダウン/	 コードダイアログを聞きます。
	アトドレスや範囲長のアライメント(G) 4 Byte Alignment ▼	
	□ プログラムを実行する直前でブレーク(」)	)]
	□ メモリリードもプログラムを使用(R)       アップロード(U)	)
-		
	① 設定済みのフラッシュデバイスを表示するリストです。	
		$ \ge $
C	「 ダウンロード時設定を無視する( <u>I</u> ) 「 ダウンロード時設定を無視する( <u>I</u> )	
② ダウンロー	-ド時にフラッシュメモリの設定 ワークメモリの設	定エリアを表示しま
を無効にしま	ਰ. ਰ.	
	Figure 64	,



## ①デバイスリスト

追加したフラッシュメモリの設定状況が表示されます。 リスト内の各項目の内容は、左から順に以下のように表示されます。

Table 31				
有効/無効	設定されたフラッシュメモリの有効/無効を示します。			
	チェックした状態で有効、チェックが無い状態でそのフラッシュメモリの設定は無効となります。			
開始アドレス	フラッシュメモリの開始アドレスです。			
終了アドレス	フラッシュメモリの終了アドレスです。			
ワークメモリ	ワークメモリを使用する場合は"WM"と表示されます。			
	使用しない場合、ここは空欄になります。			
ブロック情報	ブロック情報を使用する場合は"BI"と表示されます。			
	使用しない場合はこのエントリは空欄になります。			
デバイス名	設定するフラッシュメモリのデバイス名です。			
その他付加情報	通常デバイス	S-	:直列接続数	
		P-	:並列接続数	
		-BitBus	: バス幅	
	リード/ライト用プログラ	R-	: リードモード (0=OFF, 1=ON)	
	ムを使用するデバイス	D-	: デバッグモード (0=OFF, 1=ON)	
		-Byte	:アライメント	
		0x	: リード/ライト用プログラムの処理開始アドレス	

※表中の"-"は任意の整数値を示します。

#### ②ダウンロード時設定を無視する

デバイスリスト上での有効/無効の設定にかかわらず、オブジェクトデータファイルのダウンロード操作時に、全てのフラッシュメモリの設定を無視します。





## 4.5.2. デバイス設定タブ

#### フラッシュメモリの設定(追加)を行います。

ここで設定した範囲をフラッシュメモリとして扱い、ダウンロードなどの処理は自動的にフラッシュメモリ対応の動作をします。 また、一部のフラッシュメモリではワークメモリを使用した高速書き込みに対応しており、設定したエリアに対して高速書き込みを行う かも設定します。

通常デリ         フラッシュメモリ設定         フラッシュメモリのツア         ア・ウメモリ         ア・カメモリ         ア・カメモリ         ア・カメモリ         ア・ウメモリ         ア・カメモリ         ア・ウメモリ         ア・カ・(M)         アリカン         ア・カン         ア・カン	Y1/X       Y1/2         Y2/4/7777*       その他の設定         Y2/4/7777*       その他の設定         Y2/4/7777*       その他の設定         Y1/2       Y1/2         Y1/2       Y1/2
	■ R;0×08000000 R;0×0801ffff WM STM32F103×B R0D0 4Byte R;0×20000080          ■ R;0×08000000 R;0×0801ffff WM       STM32F103×B R0D0 4Byte R;0×20000080         ■ R;0×08000000 R;0×0801fff       ■ R;0×200000080         ■ R;0×08000000 R;0×0801fff       ■ R;0×20000000 - R;0×20001fff

Figure 65

各設定項目を入力後、[フラッシュメモリ設定]ダイアログ右上の 追加 ボタンを押すことにより、フラッシュメモリの設定(追加)を行います。



「追加」ボタンを押さずにその他のタブに切り替えたり、フラッシュメモリ設定ダイアログを閉じてしまうと、設定していた内容は破棄されます。



以下に各設定項目の内容を示します。

Table 32			
メーカー	設定するフラッシュメモリのメーカーを選択してください。		
	<種類>ボックスに表示されるフラッシュメモリが、ここで選択したメーカーでフィルタリングされます。		
	CUSTOM は、リード/ライト用プログラムを使用する。		
サイズ	ターゲット上に実装されて	いるフラッシュメモリのサイズを選択してください。	
	<種類>ボックスに表示さ	れるフラッシュメモリが、ここで指定サイズでフィルタリングされます。	
種類	ターゲット上に実装されているフラッシュメモリを選択してください。		
開始アドレス	ターゲット上で割り付いて	いるフラッシュメモリの開始アドレスを指定してください。	
範囲長	ターゲット上で割り付いているフラッシュメモリの範囲を指定してください。		
	変更不可能なタイプのフ	ラッシュメモリではグレー表示されます。	
ターゲットのワークメモリを	フラッシュメモリの高速書き込みを行う場合に使用します。		
使用する	<作成したリードライト用プログラム>を使用するタイプのフラッシュメモリもこのエリアを使用します。		
ブロック情報を使用する	ブロック情報を使用した書き込み処理を行います。		
通常タイプ	並列接続数	ターゲット上のフラッシュメモリの並列接続数を指定します。	
		同一アドレスに割り当てられ、バス幅を増やす場合の接続方法です。	
	直列接続数	ターゲット上フラッシュメモリの直列接続数を指定します。	
		同一バスに割り当てられ、アドレス範囲を増やす場合の接続方法です。	
リード/ライト用プログラム	リード/ライト用プログラ	リード/ライト用プログラムの処理開始アドレスを設定してください。	
使用タイプ	ム処理開始アドレス	最後に追加したフラッシュメモリの設定のアドレスが全ての設定に反映され	
		ます。※複数の開始アドレスを設定することはできません!	
	アドレスや範囲長のア	リード/ライト用プログラムに渡すアドレスや範囲長のアライメントを選択し	
	ライメント	ます。リストに無い場合はリードライト用プログラムで独自に設定するように	
		してください。	
	プログラムを実行する	本ソフトウェアでは使用できません。	
	直前でブレークする		
	メモリリードもプログラム	メモリリードする場合もリード/ライト用プログラムを使用します。	
	を使用		
		メモリコマンドビルダーでリード処理を登録した場合はこのチェックボックスを ON にしてください。	



連続したアドレスでフラッシュメモリが割りついていない場合は、複数に分けて登録するようにしてください。 フラッシュメモリ用ワークメモリは登録した複数のフラッシュメモリで共通に使用されます。



## 4.5.3. ワークメモリタブ

高速書き込みに対応するフラッシュメモリに対して、書き込みの際に使用するワークメモリのエリア設定を行います。

ワークメモリとして使用するエリアを 指定します。	
フラッシュメモリ設定	
フラッシュメモリのクリア フラッシュメモリのフィルアッフ <sup>®</sup> その他の設定 デバイス設定 ワークメモリ フロック情報 設定ファイルの保存/読み込み	追加(A)
	えるかメモリチェックで確認します。
開始アドレス(S) R;0×20000000 ▼ 範囲長(R) 0×2000	指定エリアをワークエリアとして更 新処理を行います。
	に表示されます。
ワークメモリの退避     フークメモリの退避     フラッシュメモリへのアクセス時     「ワークメモリへのアクセス時     マークメモリ内のデ <sup>×</sup> ータを退避     する(E)     フークメモリののテ <sup>×</sup> ータを退避	
ワークエリア使用直前にワークエリアの内	אַיָּלָאַם-וּיּ(שֵ) זײָס°ם-וּי <u>ּ(ש</u> )

Figure 66

ワークエリアは、複数の異なるフラッシュメモリがリストに追加されていても、すべてで共通に使用されます。

ワークエリアにオブジェクトデータを配置する場合は、「データを退避する」にチェックを入れてワークエリア内のデータを一時的に退避するようにしてください。



## 4.5.4. ブロック情報タブ

設定中のフラッシュメモリに対するブロック情報の調査、および取得されているブロック情報の表示を行います。



#### Figure 67

#### ①ブロック情報について

ブロック情報とは、設定されているフラッシュメモリに対して、書き込み操作時に必要とされる最小クリア範囲です。 本ソフトウェアは、この情報を使用することでクリア範囲を最小限にして書き込み処理の高速化や書き込み回数の削減を行うことが できます。

ブロック情報を使用する場合、フラッシュメモリに対しては、書き込み(ダウンロード)前にユーザーがクリア操作を行う必要がなくなりま す。ブロック情報を使用しない、またはブロック情報を調査していない場合のフラッシュメモリに対しては書き込み(ダウンロード)前に書 き込み領域を手動でクリアする必要があります。

#### ②ブロック情報の調査についての注意点

ブロック情報の調査時に、フラッシュメモリリストに登録されている全フラッシュメモリに対して複数回のクリア操作が本ソフトウェアによっ て行われます。調査後のフラッシュメモリはクリアした状態になるため、オブジェクトデータを書き込んでいた場合は再度書き込みを行 ってください。

ー度調査したブロック情報はフラッシュメモリの設定ファイルに保存されるので、設定ファイルを読み込んでおけば本ソフトウェア起動ご とにブロック情報を調査する必要はありません。



## 4.5.5. 設定ファイルの保存/読み込みタブ

フラッシュメモリ設定ファイルへの保存、および読み込みを行います。 新規プロジェクトの場合、デフォルトでは {プロジェクトファイル名}.fsh というファイルが設定されます。



Figure 68



### 4.5.6. フラッシュメモリのクリアタブ

フラッシュメモリの内容をクリアします。

フラッシュメモリ設定	×
デバイス設定       ワークメモリ       プロック特報       設定ファイルの保存/読み込み         フラッシュメモリのワリア       フラッシュメモリのフィルアッフ。       その他の設定         ウリア範囲の指定       ●       ・       ・         ・ 登録デ・ハ・イスを全てクリア(G)        開始(S)          クリアト・レス(B)       ■       ■       ・         アト・レス領域の設定       ●       ●       ●         ●       アト・レスを含むデ・ハ・イス全体をクリア(N)       ●       ●         ●       アト・レスを含むデ・n・イス全体をクリア(N)       ●       ●         ●       アト・レスを含むデ・n・イス全体をクリア(B)       ●       ●         ●       デ・n・イス全体をクリア       ▼       ●         ●       ア・ト・レスを実合します       ●       ●         // フラッシュメモリクリアを実行します       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●	<u>追加(A)</u> 削除(D) 全削除(L) 閉じる(C)
// コンバ実行をします // 終了しました	アップロード( <u>U</u> )

Figure 69

以下の3通りの方法でフラッシュメモリをクリアすることができます。

Table 33

登録デバイスを全てクリア	設定されている全フラッシュメモリに対してクリア操作を行います。
アドレスを含むデバイス全体をクリア	<クリアアドレス>で指定したアドレスを含むフラッシュメモリの1デバイス分をクリアします。
アドレスを含むブロックのみクリア	<クリアアドレス>で指定したアドレスを含むフラッシュメモリの1ブロックをクリアします。



対象となる領域のイメージを Figure 70 に示します。



Figure 70

### デバイス全体のクリア時には3通りのクリア方法を選択することができます。

Table 34		
デバイス全体をクリア	フラッシュメモリの書き込み状態に関わらず、フラッシュメモリに対してクリア操作を行います。	
書き込み済みのデバイス全体をクリア	ブランクチェックを行います。	
	書き込み済みのデータがある場合はフラッシュメモリに対してクリア操作を行います。	
書き込み済みのブロックのみクリア	ブランクチェックを行います。	
	書き込み済みのデータがあるブロックに対してクリア操作が行われます。	



ブランクチェックはフラッシュメモリの開始アドレスから終了アドレスまでの領域で、書き込み済みのデータがあるかを検査します。0xFF 以外のデータは書き込み済みデータと判断されます。



## 4.5.7. フラッシュメモリのフィルアップタブ

指定したアドレス範囲のフラッシュメモリに任意のデータを書き込みます。

アドレス範囲とフィルアップするデータを指定して 開始 ボタンをクリックすると処理を開始します。 フィルアップを行う前に、フラッシュメモリの設定が済んでいる必要があります。

フラッシュメモリ設定	×
デッバイス設定 ワークメモリ フロック情報 設定ファイルの保存/読み込み フラッシュメモリのクリア フラッシュメモリのフィルアップ その他の設定	追加( <u>A</u> )
- アドレス範囲の指定	<u>削除(D)</u>
開始アドレス( <u>T</u> ) R;0×08000000 マ	
○ 終了アドレス(E) R;0xfffffff	閉じる( <u>C</u> )
● 範囲長( <u>R</u> ) 0×100	
7/ルアップデータ(1バイト)(B) 0x99 // 7ラッシュメモリフィルを実行します // (R;0x08000000 - 0x99) ロマント*実行をします // 終了しました	ጵ°ታን⊡∽ኑ°(₩) ア"フ°⊡∽ኑ°(U)

Figure 71

Tab	ام	35
iab	IC.	JJ

アドレス範囲の指定	開始アドレス	:フラッシュメモリへのフィルアップを開始するアドレスを指定します。
	終了アドレス	:フラッシュメモリへのフィルアップを終了するアドレスを指定します。
	範囲長	:開始アドレスからの範囲を指定します。
フィルアップデータ	フラッシュメモリに書き込む任意のデータを指定します。	
	指定できるデ-	ータサイズはバイトサイズのみです。



# 5. ウィンドウ

本ソフトウェアのウィンドウについて説明します。

# 5.1. ウィンドウのドッキングについて

本ソフトウェアの子ウィンドウは、メインウィンドウの上下左右のいずれかの辺に張り付けて位置を固定したり、メインウィンドウの外側 に出して配置したりすることができます。

Oniversal Probe - A     Jァイル(E) 表示(V) リソ	ARM Writer (ST_STM32F103_STM32F103VB) (SWD) - test.armwpj
<u>•</u>	
	27459757
	// フラッシュメモリクリアを実行します BLANK CHECK NOT SUPPORT // 終了しました
	// フラッシュメモリフィルを実行します // (R;0x08000000 - 0x99) // 終了しました
73 0×20000000 - 0×2	BRFAK 「「「「I OG-OFF MPCore DISABLE No Sync 画面消去
<00	
+0 +1 +2 +3	
	0 00 00 08 00 10 00 20 00 0 1 00 00 00 00 00 02 00 04 0 アドレス(S): 0x20001000 マ の終了アドレス(E) 0x200010ff マ アウセフ
00 be 00 00 0	0 00 00 00 00 00 00 00 00 00 00 00 00 0
40 41 42 43 4	4 45 46 47 48 49 4a 4b 4c 4 Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
50 51 52 53 5 60 61 62 63 6	4       55       56       57       58       59       5a       5b       5c       5       0×20001000       99
	0×20001040 0×20001050 0×20001050 0×20001060 99 99 99 99 99 99 99 99 99 99 99 99 99
	BREAK         TARGET   MPCore DISABLE  N //

Figure 72



## 5.1.1. ウィンドウの状態

### ①MDI 子ウィンドウ (通常のウィンドウ)

本ソフトウェアのメインウィンドウの内で自由に移動・伸縮が可能なウィンドウです。 各ウィンドウはこの状態で開きます。

Oniversal Probe - ARM Writer (ST_STM32F103_STM32F103VB) (SWD) - test.armwpj     フィルドー 表示(ハールトー(R) 車行(G) コンド・ワ(ハー) ムル"(ト)	
■ 9" <sup>3</sup> 7 <sup>2</sup> 9/3 <sup>k*</sup> 95	
7ドレス(S): 0×20000000 · C 終7アドレス(E) 0×200000ff · アクセスサイス(A): ロンクワード ·	
デー(20): 「 範囲長(1) 0×100	
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII	i i i i i i i i i i i i i i i i i i i
	<u> </u>
য় সায় সায	
BREAK BREAK	e DISABLE N 🏑

Figure 73

MDI 子ウィンドウは、メインウィンドウの外に出ることができません。

🚺 Universal Probe - ARM Writer (ST_STM32F103_STM32F103VB) (SWD) - test.armwpj	
771M(E) 表示(Y) リリース(B) 実行(G) 9/2ド*の(W) N/2*(H)	
マンウンドウからはシャートが分け層のフロ	5+2/+2
メインワインドウからはの山した部プは隠れて見	[A/42°
BREAK         TARGET MPCore DISABLE N //	
Figure 74	



#### ②ドッキングウィンドウ

メインウィンドウの上下左右のいずれかの辺に張り付いた状態のウィンドウです。(下図) ドッキング状態でも、ウィンドウの伸縮は可能です。また、ドッキングした状態から他の辺に付け替えることもできます。

Universal Probe - ARM Writer (ST_STM32F103_STM32F103VB) (SWD) - test.armwpj	
7ァイル(E) 表示(Y) リソース(R) 実行(G) ワィンドワ(W) ヘルプ(H)	
7ドルス(S): 0×20000000 - C 終7アドルス(E) 0×200000ff マアクセスサイス(A): ロンクワード -	
デー火①: 🛛 🕅 🕅 🗰 囲長(山) 🛛 🕼 🗤 100 🔽 🔍 🔍 🔽 更新しない 🖤	
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII	*
	<u>^</u>

Figure 75

#### ③フローティングウィンドウ

フローティング状態の時、子ウィンドウはメインウィンドウの外側に出ることができます。 ただし、本ソフトウェアを終了、またはプロジェクトを閉じたときは、フローティング状態のウィンドウも自動的に閉じます。

フローティングウィンドウをメイ 出す。	ンウィンドウの外に	
ダ"ンプ9インド95 アドレス(S): 0x20000000 データ(D): Address +0 +1 +2	<ul> <li>○ 終了アドレス(E)</li> <li>○ 範囲長(L)</li> <li>-3 +4 +5 +6 +7 ···</li> </ul>	

Figure 76



### 5.1.2. 子ウィンドウ状態の変更方法

子ウィンドウの状態を変更するには、変更したい子ウィンドウのタイトルバーを右クリックします。 すると下図のようなコンテキストメニューが表示されますので、変更したい状態やドッキングしたい位置を指定します。



Figure 77

以下に、メニューの図と各項目の説明をします。

Table 36





メニュー項目			
Docked			
		ドッキングする辺は、下辺→左辺→右辺→下辺→…の順に切り替わります。	
Floating		対象の子ウィンドウをフローティング状態にします。	
		対象の子ウィンドウを MDI 子ウィンドウ状態にします。	
MDI Child		子ウィンドウがフローティング状態で、メインウィンドウの外側にあったときは、自動的にメイン	
		ウィンドウの内側に移動します。	
	Тор	対象の子ウィンドウをドッキング状態にし、メインウィンドウの上辺に張り付けます。	
Decked to	Left	対象の子ウィンドウをドッキング状態にし、メインウィンドウの左辺に張り付けます。	
Docked to	Bottom	対象の子ウィンドウをドッキング状態にし、メインウィンドウの下辺に張り付けます。	
	Right	対象の子ウィンドウをドッキング状態にし、メインウィンドウの右辺に張り付けます。	
	Minimized	対象の子ウィンドウを MDI 子ウィンドウ状態にし、最小化(アイコンの状態に)します。	
MDI Child as	Maximized	対象の子ウィンドウを MDI 子ウィンドウ状態にし、最大化します。	
	Restored	対象の子ウィンドウを MDI 子ウィンドウ状態にし、普通のサイズで表示します。	

Table 37



## 5.1.3. ドラッグ&ドロップによるドッキング状態の変更方法

#### ①ドッキングしている辺を変更する

子ウィンドウがドッキング状態、またはフローティング状態のときは、「5.1.2. 子ウィンドウ状態の変更方法」の他に、その子ウィンドウのタイトルバーをドラッグ&ドロップすることによって、ドッキングさせる辺を変更することができます。 子ウィンドウのタイトルバーをドラッグし、ドッキングしたいメインウィンドウの辺へ移動させると枠の形状が変化しますので、ドロップする と位置が変わります。

<u> </u>

Figure 78





#### ②ドッキング状態とフローティング状態の変更

ドッキング状態の子ウィンドウをメインウィンドウの外側または中央付近にドラッグすると、フローティング状態に変化します。 フローティング状態の子ウィンドウをメインウィンドウの辺の近くヘドラッグ&ドロップすると、ドッキング状態に変化します。



Figure 80

#### ③MDI 子ウィンドウへの変更

ドラッグ&ドロップ操作によって、ドッキング状態またはフローティング状態の子ウィンドウを MDI 子ウィンドウに変更することはできません。MDI 子ウィンドウ状態に変更したい場合は、子ウィンドウのタイトルバーを右クリックして表示されるコンテキストメニューから 「MDI Child」を選択してください。



## 5.1.4. 同じ辺に複数の子ウィンドウをドッキングさせた場合

メインウィンドウの同じ辺に複数の子ウィンドウをドッキングさせた場合、表示方法として以下の2種類の状態があります。

- 1) 既にドッキングしている子ウィンドウよりさらに内側にドッキングした状態
- 2) メインウィンドウの一辺(ドッキングしている辺)に複数の子ウィンドウがドッキングした状態

以下の図は、(1)の例です。

Universal Probe - ARM Writer (ST_STM32F103_STM32F103VB) (SWD) - test.armwpj	- • •
77(N(E) 表示(Y) リソース(R) 実行(G) ワインドワ(W) ヘルプ(H)	
ר *ליליל *לידע ליליליל *ליליל *ליליל	
// フラッシュメモリクリアを実行します BLANK CHECK NOT SUPPORT // 終了しました	•
// フラッシュメモリフィルを実行します // (R;0x08000000 - 0x99) // 終了しました	E
BREAK / / / / / / / / / / / / / / / / / / /	
>     <	
<u>אי א</u> רי איז 15 מינער איז 15 מינער איז 10 מינער א	X
7ドレス(S): 0x20000000 ・ C 終了7ドレス(E) 0x200000ff ・ アクセスサイス(A): ロンクワード・	
〒→(①): 「 範囲長(1) 「0×100 「 更新しない(1)	
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII	*
	<u>_</u>
BREAK	ore DISABLE N

Figure 81



Universal Probe - ARM Writer (ST_STM32F103_STM32F103VB) (SWI	D) - test.armwpj
ファイル(E) 表示(V) リソース(R) 実行(G) ワインドワ(W) ヘルプ(H)	
💁 😤 🔛 🛄 🛄 🖸 🖬 🔛 🛄 🖤	
9° 27° 9428° 95	
アドレス(S): 0x20000000  ・ (於了アドレス(E) 0x2000000ff	// フラッシュンモリクリアを実行します ^   BLANK_CHECK_NOT_SUPPORT
データ(D): 0×100	// 終了しました
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B 🚍	
	$// (R; 0 \times 08000000 - 0 \times 99)$
	· · · · · · · · · · · · · · · · · · ·
	BREAK [         LOG-OFF MPCore DISABLE No Sync 画面消去
-	>
	CLS COPY
	BREAK

Figure 82

## 5.1.5. ドッキングウィンドウのサイズ変更

ドッキング状態の子ウィンドウのサイズ変更は、マウスカーソルで子ウィンドウ周辺の枠をドラッグすることで行います。

<ul> <li>・ 終了アドレス(E) 0×200000</li> <li>● 範囲長(L) 0×100</li> <li>+5 +6 +7 +8 +9 +A</li> </ul>
---



## 5.2. ダンプウィンドウ

メモリ内容をダンプして表示するウィンドウです。ダンプウィンドウは複数開くことができ、以下の特徴があります。

- 指定された範囲のメモリデータを表示します。
- 各種フォーマットでの表示が可能です。
- 符号付き/符号なし整数での1バイト、2バイト、4バイト表示が可能です。
- 16 進数での1バイト、2バイト、4バイト表示が可能です。
- 実数での4バイト、8バイト、10バイト、12バイト表示が可能です。
- データの変更が可能です。
- 読みだした際に変更のあったデータは赤色に変わるため、データの変化が容易に判別できます。
- 物理/論理アドレスでの表示が可能です。
- 表示された内容を CSV 形式やテキスト形式で出力することができます。



アドレスビュー、データビュー、文字列ビューの各表示領域は、境界線をマウスでドラッグすることによりサイズを調整することができます。

ダイアログバー





Table 38		
アドレス	表示開始アドレスを指定します。	
終了アドレス, 範囲長	終了アドレス、または、表示させる範囲を指定します。	
	数値を入力後、リターンキーを押すと指定範囲のメモリ内容をを表示します。	
0	<アドレス>-<終了アドレス>or<範囲長>で指定した範囲を、再読み込みします。再読	
(表示ボタン)	み込みした際に、前回の表示値から値が変化した場合は、データビューの数値が赤文字で	
	表示されます。	
0	[アドレスフォーマット]、[データフォーマット]、[アクセスサイズ]を変更するためのコンテキストメ	
🧕 (コンテキストメニュー表示)	ニューが表示されます。	
□ 更新しない(世)	このチェックをしておくと、表示ボタンで再読み込みしても表示内容を更新しません。	
<アクセスサイズ>	メモリデータのアクセスサイズを変更します。	
データ	カーソルがある位置のデータを表示します。	
	データ値を変更したい場合、この値を変更してリターンキーを押します。	
	演算式も使用できます。 例) 0x1+0x10	
	※下記の表示フォーマットの時は"/"演算式でのデータ指定は出来ません。	
	<ul> <li>符号付き8バイト整数</li> </ul>	
	<ul> <li>符号なし8バイト整数</li> </ul>	
	• 8 バイト 16 進数	
<b>又</b> (変更ボタン)	<データ>の値をメモリに書き込みます。	
(キャンセルボタン)	<データ>の表示が、現在カーソルが示している値に戻ります。	



5.2.1. ダンプウィンドウのコンテキストメニュー

データが表示されている領域で、右クリックをするとコンテキストメニューが表示され、表示形式の選択やデータのコピーができます。

アトッレスフォーマット ► デドータフォーマット ⊧ ፖሳቲአサイズ Þ ファイルへ出力する データをコピー(C) 文字列をコピ-(S) アドレスとデータをコピ-(D) アトッレスとデッータと文字列をコピー(A) データと文字列をコピ-(F)

Figure 85

	Table 39
アドレスフォーマット	表示アドレスのフォーマットを変更します。
	物理アドレス表示
データフォーマット	メモリデータの表示フォーマットを変更します。
	符号付き1バイト整数
	符号付き2バイト整数
	符号付き4バイト整数
	符号付き8バイト整数
	符号なし1バイト整数
	符号なし2バイト整数
	符号なし4バイト整数
	符号なし8バイト整数
	1 バイト 16 進数
	2 バイト 16 進数
	4 バイト 16 進数
	8 バイト 16 進数
	4 バイト実数
	8 バイト実数
	10 バイト実数
	12 バイト実数
アクセスサイズ	メモリデータのアクセスサイズを変更します。
	バイト
	ワード
	ロングワード



ファイルへ出力する	[ファイルへ出力]ダイアログが表示されます。出力形式は CSV 形式とテキスト形式があ
	り、CSV 形式の場合はオプションの指定が可能です。
データをコピー	データビューで、範囲選択領域にある文字列をクリップボードにコピーします。
	※キーボードから Ctrl + C を入力してもコピーできます。
文字列をコピー	文字列ビューの範囲選択領域にある文字列をクリップボードにコピーします。
	文字列ビューが表示されている時だけ有効になります。
アドレスとデータをコピー	アドレスビューとデータビューの範囲選択領域にある文字列をクリップボードにコピーします。
アドレスとデータと文字列をコピー	アドレスビューとデータビューと文字列ビューの範囲選択領域にある文字列をクリップボード
	にコピーします。
	文字列ビューが表示されている時だけ有効になります。
データと文字列をコピー	データビューと文字列ビューの範囲選択領域にある文字列をクリップボードにコピーします。
	文字列ビューが表示されている時だけ有効になります。

## ①[ファイルへ出力]ダイアログ

7711人出力		<b>—</b> ×-
ファイル名(E):	参照	OK
- 出力形式		キャンセル
- オプジョン		

Figure 86

Tab	le	40
iub	IC.	70

ファイル名	指定したファイル名で保存します。(CSV 形式を選択された場合、拡張子の".CSV"を付けてください)。
出力形式	CSV 形式、テキスト形式のいずれかの選択が出来ます。
オプション	数値書式を選択します。



### ②領域選択の方法

マウスもしくはキーボードで表示された値を選択することができます。

💷 ቃ* ኦፖ° ዓイኦト* ዓፖ	0x20000	000 - 0x	200	000ff	(0x100	))								- • •
アドレス( <u>S</u> ): 0×20	000000		•	〇終	የንጉኮ	ג( <u>E</u> )	0×20	0000	)ff			-	アウセスサイス"( <u>A</u> ):	ロングワード 💌
テ°-%( <u>D</u> ): 0×5d		X	1	◉ 範	囲長(し	)	0×10	0				•	Q 💡 🗆 更新U	.tal i(U)
Address	+0 +1	+2 +3	+4	+5 -	+6 +7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII 🗧
0×20000000 0×20000010 0×20000030 0×20000040 0×20000050 0×20000060 0×20000060 0×20000070 0×20000080 0×20000080	00 00 20 00 00 be 00 00 40 41 50 51 60 61 70 71 10 b5 04 40	00 00 00 20 00 00 42 43 52 53 62 63 72 73 04 00 40 10	00 01 00 00 44 54 64 74 60	00 ( 00 ( 00 ( 45 ( 55 ( 75 ( 69 ( d) )	00 08 00 00 00 00 00 00 46 47 56 57 66 57 66 67 76 77 00 0c	00 00 00 48 58 58 58 78 58 78 58 78 58	10 00 00 49 59 69 79 04 -0	00 02 00 4a 5a 6a 7a 60	20 00 00 4b 5b 6b 7b 61 00	00 04 00 4c 5c 6c 7c 20 92	01 00 00 4d 5d 6d 7d 68	00 00 00 4e 5e 6e 7e 80 20	00 00 00 4f 5f 6f 7f 1e 00	.t @ABCDEFGHIJKLMNO PQRSTUVWXYZ[¥]^_ `abcdefghijklmno pqrstuvwxyz{l}~ .t`a.h∎. .≋@ ≈@ ≈ #%~

Figure 87

	Table 41
マウスから領域を選択する	データビューでドラッグ&ドロップすると、その範囲を選択します。
キーボードから領域を選択する	データビューで Shift キーを押しながらカーソルを移動させると、カーソルを移動した範囲を選 択します。
領域選択を解除する	データビューでマウスを左クリックするか、キーボード操作でカーソルを移動すると選択状態を 解除します。



#### ③クリップボードへのコピー

範囲選択し、コンテキストメニューから「〜コピー」を選択すると、選択した範囲をクリップボードにコピーします。 キーボードから「Ctrl + C」を入力したときは、「データをコピー」の動作になります。

	Table 42
データをコピー	6b f7 1b d6 71 cc 77 4c b1 35 71 46 c3 e2 31 28 b2 54 63 5c 12 ca d3 f3 03 f0 9f ad 36 9e 43 3f d0 2d 47 ef b1 1d f4 80 43 0e 83 64 47 c4 3c 0a b1 1e 11 4e
文字列をコピー	k · ∃ q フ wL 7 5qF テ · (イ Tc¥. ハモ · ・」 6 曚?ミーG · . ・ C. ヅ G ト<. 7 N
アドレスとデータをコピー	R;0x00a205e06b f7 1b d6 71 cc 77 4c b1 35 71 46R;0x00a205f0c3 e2 31 28 b2 54 63 5c 12 ca d3 f3 03 f0 9f adR;0x00a2060036 9e 43 3f d0 2d 47 ef b1 1d f4 80 43 0e 83 64R;0x00a2061047 c4 3c 0a b1 1e 11 4e
アドレスとデータと文字列をコピー	R;0x00a205e0  6b f7 1b d6 71 cc 77 4c b1 35 71 46   k・ヨ q 7 wL 7 5qF R;0x00a205f0 c3 e2 31 28 b2 54 63 5c 12 ca d3 f3 03 f0 9f ad   テ・(イ Tc¥. ^モ・・ı R;0x00a20600 36 9e 43 3f d0 2d 47 ef b1 1d f4 80 43 0e 83 64   6 曚?ミ-G・. ・ C. ヅ R;0x00a20610 47 c4 3c 0a b1 1e 11 4e   Gト<. 7 N ※右端で折り返しています
データと文字列をコピー	6b       f7       1b       d6       71       cc       77       4c       b1       35       71       46               k ⋅ ∃ q 7 wL 7 5qF         c3       e2       31       28       b2       54       63       5c       12       ca       d3       f3       03       f0       9f       ad         $\bar{\tau} \cdot (4 \text{ Tc}¥. \wedge F \cdot \bullet 1)$ 36       9e       43       3f       d0       2d       47       ef       b1       1d       f4       80       43       0e       83       64               6 \overline{k}?\$       -G · . • C. "Y         47       c4       3c       0a       b1       1e       11       4e               G         K · N



## 5.2.2. ダンプウィンドウの複数表示

下記ツールボタンをクリックする度に、新規のダンプウィンドウを開くことができます。

ツールボタン	メニューバーの操作	ショートカットキー
MEM	表示 → ダンプウィンドウ	

<u>۹</u> ۶۲۵۶ و	ンドウ7 0x20000000	- 0x200000ff	(0x100)			
7ドレス( <u>S</u> ):	0×20000000	<b>、</b> 〇 総	冬了アドレス( <u>E</u> )  0×20	10000ff	· アクセスサイス( <u>A</u> ): ロングワート	•
テ°-タ( <u>D</u> ):	0×5d	Í 🗙 🗸 🤉 🗉	施囲長(L) Ox10	10 -	• 🔍 🖓 🗆 更新しない(山)	
Address	+0 +1 +2	+3 +4 +5	+6 +7 +8 +9	+A +B +C +D +	E +F ASCII	
0×200000	00 00 00 00	00 00 00	00 08 00 10	00 20 00 01 0	0 00	
0×200000 0×200000	# 9*ን7° 94ንት* 98	0x20001000	- 0x200010ff (0x:	100)		
0×200000	アドレス( <u>S</u> ): 0×20	1001000	▼ ○ 終了ア	'ドレス( <u>E</u> )  0×200010ff	💽 アウセスサイズ( <u>A</u> )	: ロングワート <sup>×</sup>
0×200000	¯ <sup>*</sup> −匆( <u>D</u> ): 0×99	1	- 🚺 🗸 🖲 範囲長	冕( <u>L</u> ) 0×100	🗾 🔍 🗆 🗉	新しない(旦)
0×200000	Address	+0 +1 +2	+3 +4 +5 +6	+7 +8 +9 +A +	B +C +D +E +F	ASCII 🚊
0×200000	0×20001000	<mark>99</mark> 99 99	99 99 99 99	99 99 99 99 9	9 99 99 99 99	
0~200000	0×20001010 0×20001020	99 99 99	99 99 99 99 99	99 99 99 99 99 9	9 99 99 99 99 99 9 99 99 99 99	动动动动动动动动
	0×20001030	99 99 99	99 99 99 99	99 99 99 99 99 9	9 99 99 99 99 99	刻刻刻刻刻刻刻刻刻
	0×20001040	99 99 99	99 99 99 99	99 99 99 99 9	9 99 99 99 99	> 創刻刻刻刻刻刻刻刻
	0×20001050	99 99 99	99 99 99 99 99	99 99 99 99 99 9	9 99 99 99 99 99	动动动动动动动
	0×20001070	99 99 99	<u>99</u> 99 99 99 99	99 99 99 99 99 9	9 99 99 99 99	刻刻刻刻刻刻刻刻
	0×20001080	99 99 99	99 99 99 99	99 99 99 99 9	9 99 99 99 99 99	

Figure 88



同じアドレス範囲のウィンドウを 2 つ表示しておき、片方のダンプウィンドウに<更新しない>のチェックマークを入れておく と、メモリの変更内容を見比べることができます。



### 5.2.3. "ASCII 文字列"入力の方法

1 バイト表示モードでは、下記の手順でカーソル位置から任意の文字列データをメモリに書き込むことが出来ます。

1) 文字列データを書き込む最初のアドレス位置にカーソルを移動します。

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII	*
0×20000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		 -
0x20000010 0x20000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0×20000030 0×20000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
										Fi	gur	e 8	9					

2) <データ>ボックスにダブルクォーテーションで囲んだ文字列を入力し、Enter キーを押します。

データ(<u>D</u>): 『c:¥ARM"|

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII	
0×20000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0×20000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	<u>.</u>	
0×20000020	00	00	63	3a	5c	41	52	4d	00	00	00	00	00	00	00	00	c:¥ARM <mark>.</mark>	
0×20000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
IO. 200000 MO	100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		

Figure 90

ダブルクォーテーションで囲まれた文字列はそのまま文字コードに変換し、カーソル位置から1バイトごとに書き込みます。 書き込みの際には以下の注意点・制限があります。

- 入力された文字列に対して、エスケープシーケンスなどのコードデータへの変換処理は行いません。
- キーボードから入力できないコードや、ダブルクォーテーションはメモリデータを直接入力してください。
- コンボボックスに入力できる文字列長は、ダブルクォーテーションも含めて 256 バイトです。
- 表示および書き込まれるデータは、全て Shift-JIS として扱われます。
- 書き込む文字列が未表示範囲に及んだ場合、未表示範囲への書き込みは行いません。



# 5.3. コマンドウィンドウ

コマンドライン入力を行うウィンドウです。以下の特徴があります。

- ・ キーボードやマウスでの連想選択方式によって、素早いコマンド操作が可能です。
- コマンド実行結果の表示とコマンドのロギング機能があります。
- バッチ機能による自動実行が可能です。
- バッチファイル作成機能があります。

下記ツールボタンをクリックすることで、コマンドウィンドウを開くことができます。

 ツールボタン
 メニューバーの操作
 ショートカットキー

 <br>
 <br>

 <br/>

 <br/>

ל"אלאל פיאלא איז איז איז איז איז איז איז איז איז אי	X
C:¥ARM_WRITER	
	ステータスバー
	2
  BREAK         LOG-OFF  MPCore DISABLE  No Sync ディレクトリの変更	
Image: CD     CHECK     CLOSE     CLS     COPY	

Figure 91

→ コマンドの入力方法は『7. コマンドラインインターフェース』を参照ください。

ステータスバーには、現在の CPU の状態や各種測定機能の状態、コマンドラインインターフェースのガイドが表示されます。

クリップボードにコピーした文字列をバッチファイルとして実行することができます。コマンド入力部で右クリックし、コンテキストメニューを開いて「バッチを貼り付け」を選択するか、コマンド入力部にカーソルを合わせた状態で「Ctrl + B」を入力します。



# 6. キーボード

一般的な Windows キーボード操作以外の、キーボード操作について説明します。

# 6.1. 全てのウィンドウ

全てのウィンドウで以下のキーボード操作ができます。

<u>ESC</u> … <u>ブレーク</u><sup>\*1</sup> F5 … <mark>実行開始</mark>

\*1:コマンドウィンドウでコマンド入力中は、コマンドのキャンセル機能を優先します。

## 6.2. ダンプウィンドウ

ダンプウィンドウでは以下の様なキーボード操作ができます。

□ ▲ □ ↓ □ ↓ □ ↓ □ → □ ・・・ データビュー、文字列ビュー内のカーソルを移動します。

## 6.3. コマンドウィンドウ

コマンドウィンドウでは以下の様なキーボード操作ができます。

スペース … コマンドやパラメータを確定します。

- ESC … コマンド入力中の場合は、コマンドやパラメータを1つキャンセルします。 コマンドが入力されていない場合は強制ブレークします。
- リターン … コマンドを確定して実行します。省略されたパラメータは前回入力されたものが設定されます。



# 7. コマンドラインインターフェース

コマンドウィンドウのコマンド入力・操作方法について説明します。 コマンド入力は Figure 92 のコマンドラインで行います。

2775, 2475, 2475	
>cd C:¥ARM WRITER	
	_
BREAK	—

Figure 92

# 7.1. コマンドラインでの操作方法

コマンド名やパラメータ名を覚えていなくても、最初の数文字を入力するだけで該当するコマンド名やパラメータ名が表示され、絞り 込んでいく事でコマンド入力を簡単に行うことができます。

7.1.1. コマンドの入力方法

コマンドラインをクリックし、フォーカスを移動します。 コマンドを入力して、リターンキーを押すことでコマンドを実行します。 パラメータは省略することができるものがあり、省略した場合は前回入力した内容を適用します。

7.2.2. 連想選択方式によるコマンド入力



## 7.2.3. 履歴から入力する

コマンドラインで 1011 キーを押すと、コマンドヒストリを表示することができます。 ヒストリから実行したいコマンドを選択し、リターンキーを押すことで過去のコマンドを再実行することができます。



コマンドラインへの全角の入力はできません。 また、パスを指定のする際、スペースの入力は受け付けないため、スペースを含むフォルダ名は使用できません。



## 7.2. コマンド解説

## 7.2.1. ASSIGN とドット(.)(式の評価)

パラメータで指定された式を評価します。

このコマンドでは、メモリや I/O データの参照や変更、またはレジスタデータの参照や変更ができます。





Figure 93

#### ●<式>

評価する式を指定します。

#### ●例

Table 43

.[0x4000].B=0x10	0x4000 番地へ 0x10 を 1 バイトデータとしてライトする。
.[0x4000].W=0x10	0x4000 番地へ 0x10 を 2 バイトデータとしてライトする。
.[0x4000].L=0x10	0x4000 番地へ 0x10 を 4 バイトデータとしてライトする。
.[INR:0x4000].B=0x10	内蔵 I/O レジスタの 0x4000 番地へ 0x10 を 1 バイトデータとしてライトする。
.[P;0x4000].B=0x10	I/O アドレスの 0x4000 番地へ 0x10 を 1 バイトデータとしてライトする。

詳細は「<mark>3.2. アドレス表現</mark>」、「<mark>3.4. メモリ・I/O ポートの読み書き</mark>」を参照してください。



 $\rightarrow$ 

ドットを使った場合は、コマンドウィンドウのヒストリには残りません。


# 7.2.2. BATCH(バッチプログラムの実行)

指定されたバッチプログラムを実行します。 バッチプログラムの詳細は、[バッチ機能]を参照してください。

### ●書式



Figure 94

## ● <バッチファイル名>

バッチプログラムファイル名を指定します。

## ●<引数>

バッチプログラムへ渡す引数を指定します。 各引数はスペースで区切ります。 また各引数は、バッチプログラム内の\$1~\$9 に設定されます。 \$0 はコマンド全体を示します。 10 個目以降の引数は無視します。



# 7.2.3. BREAK(ブレーク)

実行中の CPU を強制ブレークします。 このコマンドは、以下のメニューと同じ機能です。



### ●書式





# 7.2.4. CD(フォルダの変更、ドライブの変更、カレントパス表示)

指定されたパスへ作業フォルダを移動します。 フォルダ名を省略した場合は、現在の作業フォルダが表示されます。

#### ●書式



Figure 96

## ●<パス名>

変更するパス名を指定します。

#### ●例

[フォルダ名を指定した時の表示] 移動した後の作業フォルダを表示します。

ገኛንት ሲላንት ባ
area vier v
>CD c:¥ARM_Writer
c:¥ARM_Writer

Figure 97

[フォルダ名を省略した時の表示] 現在の作業フォルダを表示します。

(!)

	אלאני >cd C:¥ARM_WRIT	ſER			
		Figure 98			
ファイル名・フォルダ名は、 E	] DOS 仕様の" 8 \$	文字.3 文字"の	形式で指定する必	ふ要があります。	$\overline{)}$



# 7.2.5. CHECK(メモリチェック)

●書式

指定されたアドレス範囲のメモリチェックを行います。 コマンド実行後は、メモリチェック結果が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → メモリチェック	

<アドレス範囲> ٦ Γ <コマンド> CHECK 開始アドレス LENGTH 範囲長 ΤO 終了アドレス <モード> <アクセスサイズ> SIZE BYTE ASIZE WORD LWORD <データマスク> MASK DISABLE <マスクデータ> ENABLE

Figure 99

● <開始アドレスアドレス範囲値>

	lable 44	
開始アドレス	メモリチェックを行う開始アドレスを指定します。	
範囲長	開始アドレスからの長さをバイト数で指定します。	
終了アドレス	終了アドレスを指定します。	



### ●<モード>

Table 45

SIZE	データサイズの設定をします。
ASIZE	アクセスサイズの設定をします。
MASK	データマスクの設定をします。

#### ● <アクセスサイズ>

Table 46					
BYTE	バイトアクセスでメモリチェックします。				
WORD	ワードアクセスでメモリチェックします。				
LWORD	ロングワードアクセスでメモリチェックします。				

● <データマスク>

DISABLE	データマスクを指定しません。
ENABLE <マスクデータ>	マスクデータを指定します。
	チェック対象外にしたいビット位置が1となるデータを指定します。
	データ長は SIZE で指定したサイズになります。
	例) 0x01 = 最下位ビットはチエックしない。



# 7.2.6. CLOSE(プロジェクトファイルを閉じる)

現在開いているプロジェクトファイルを閉じます。 コマンドを実行すると、確認のダイアログが表示されます。 確認のダイアログを表示せずに終了する場合は <u>EXIT コマンド</u>を使用します。

#### ●書式





# 7.2.7. CLS(コマンドウィンドウのクリア)

現在開いているコマンドウィンドウをクリアします。

## ●書式





# 7.2.8. COPY(メモリコピー)

●書式

指定されたアドレス範囲のメモリデータを目的のメモリへコピーします。 コマンド実行後は、メモリコピー結果が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → <mark>メモリコピー</mark>	

<アドレス範囲> ٦ Γ <コマンド> COPY 開始アドレス 範囲長 LENGTH 終了アドレス TO <コマンド> MEM <コピー先アドレス> ADDR i <アクセスサイズ> ASIZE BYTE WORD LWORD



#### ●<開始アドレスアドレス範囲値>

Table 48

開始アドレス	メモリコピーを行う開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

#### ● <コマンド>

Table 49

MEM	メモリコピー先の<コピー先アドレス>を指定します。
ASIZE	アクセスサイズを指定します。

#### ● <コピー先アドレス>

コピー先アドレスを指定します。

#### ● <アクセスサイズ>

BYTE	バイトアクセスでコピーします。
WORD	ワードアクセスでコピーします。
LWORD	ロングワードアクセスでコピーします。



# 7.2.9. DIR(フォルダ内容の参照)

指定されたパスの内容を参照します。 パス名を省略した場合は、現在の作業フォルダの全ファイルの内容が表示されます。 このコマンドは、DOSのDIRコマンドと同じです。

#### ●書式



Figure 103

#### ● <パス名>

参照するパス名を指定します。

●例

<u>פאלאליאליאכ</u>								
		<dir></dir>	2014-09-26	17:49				*
		<dir></dir>	2014-09-26	17:49				
STM32F~1	MOT	163752	2010-03-25	13:43	STM32F10X-128	8K-EVAL_default	_rom.mot	
TEST		<dir></dir>	2014-09-26	15:20	TEST			
TEST~1	ARM	64	2014-09-26	17:48	test.armwpj			=
test	bak	64	2014-09-26	15:33	test.bak			
test	exs	0	2014-09-26	17:48	test.exs			
test	fsh	3305	2014-08-04	11:53	test.fsh			-
BREAK I LOG-OFF MPCore DISABLE No Sync ディルかり内容の表示								
Х								
$\langle \langle \rangle$	>>	DIR	DUM	P	ENV	ERROR_ECHO	EXIT	

Figure 104



# 7.2.10. DUMP(メモリダンプ)

指定されたアドレス範囲のメモリデータを指定された表示フォーマットで表示します。 このコマンドは、ダンプウィンドウでの操作と同じです。



### ●書式



#### ● <開始アドレスアドレス範囲値>

Table 51

開始アドレス	メモリダンプを行う開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

#### ●<モード>

	Table 52
FORMAT	表示フォーマットを指定します。
ASIZE	アクセスサイズを指定します。

#### ● <表示フォーマット>

Table 53		
HEX	16 進数で表示します。	
DEC	10 進数で表示します。	
UDEC	符号なし 10 進数で表示します。	
REAL	実数で表示します。	

#### ● <表示サイズ>

Table 54

BYTE	バイトサイズで表示します。	
WORD	ワードサイズで表示します。	
LWORD	ロングワードサイズで表示します。	
4BYTE	4 バイト実数形式で表示します。	例) -1.073726E+008
8BYTE	8 バイト実数形式で表示します。	例) -9.255963134931E+061
10BYTE	10 バイト実数形式で表示します。	例) -4.7798665708109333850E+0986
12BYTE	12 バイト実数形式で表示します。	例) -4.761268151994454601E+0986

### ● <アクセスサイズ>

	Tabl	e 55	
BYTE	バイトアクセスでメモリをリードします。		
WORD	ワードアクセスでメモリをリードします。		
LWORD	ロングワードアクセスでメモリをリードします。		



# 7.2.11. ENV(プローブ環境の設定)

各種のプローブ環境の設定を行います。 コマンド実行後は、現在の設定状態が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → <u>プローブ環境の設定</u>	



Figure 106

### ●書式



# ●<モード>

# MEMSIZE

エミュレータシステムがメモリをアクセスするときのデフォルトのアクセスサイズを指定します。

	Table 56
BYTE	バイトサイズでアクセスします
WORD	ワードサイズでアクセスします
LWORD	ロングワードサイズでアクセスします

#### ENDIAN

エンディアンを指定します。

Tab	le	57
		<u> </u>

BIG	ビッグエンディアンに設定します
LITTLE	リトルエンディアンに設定します

#### ABORTCHK

メモリアクセス時にアボートチェックを行うか指定します。

Table 58

OFF	メモリアクセス時のアボートチェックを OFF にします
ON	メモリアクセス時のアボートチェックを ON にします

JTAGCLOCK

JTAG/SWD クロックの周波数を指定します。

AUTO	自動設定します。
SLOWCLOCK	次のパラメータで具体的な周波数を設定します。入力する数値の単位は KHz で数値のみ入力します。
周波数選択	1~1000の数値を指定します。



# 7.2.12. ERROR\_ECHO(エラーメッセージ表示の設定)

エラーメッセージの表示方法を指定します。

# ●書式



Figure 107

## ●<設定>

Table 60

OFF	エラーメッセージをメッセージボックスで表示します。
ON	エラーメッセージをコマンドウィンドウに表示します。



# 7.2.13. EXIT(本ソフトウェアの終了)

本ソフトウェアを終了します。 このコマンドは以下の操作と同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → アプリケーションの終了	

●書式



Figure 108

# ●<モード>

	Table 61
SAVE	現在のプロジェクトファイルに上書き保存して終了します。
SAVEAS	別名のプロジェクトファイルに保存し、終了します。
NOTSAVE	プロジェクトファイルを保存せずに終了します。

## ●<ファイル名>

保存するファイル名を入力します。ファイル名のみの場合は現在の作業ディレクトリに保存します。



# 7.2.14. FILL(メモリフィルアップ)

指定されたアドレス範囲のメモリデータを指定されたデータでフィルアップします。 コマンド実行後は、実行結果が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → <u>メモリのフィルアップ</u>	

●書式





### ● <アドレス範囲>

	Table 62
開始アドレス	メモリフィルを行う開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

#### ●<モード>

	Table 63
DATA	フィルデータを指定します。
ASIZE	アクセスサイズを指定します。

### ● <フィルデータ>

ここで指定したデータでメモリをフィルアップします。

### ● <データサイズ>

BYTE	バイトデータでメモリフィルします。
WORD	ワードデータでメモリフィルします。
LWORD	ロングワードデータでメモリフィルします。

#### ● <アクセスサイズ>

BYTE	バイトアクセスでメモリフィルします。
WORD	ワードアクセスでメモリフィルします。
LWORD	ロングワードアクセスでメモリフィルします。



# 7.2.15. FM(フラッシュメモリの設定)

フラッシュメモリを設定します。

### ●書式



Figure 110



#### ●種類

Table 66

DEVICE	フラッシュメモリ名を指定します。
CUSTOM	フラッシュメモリへのアクセスにはアクセスプログラムを使用します。

#### ● <デバイス名>

フラッシュメモリのデバイス名は、「メーカー名\_サイズ\_品種その他」の書式になっています。 ※アンダーバーの箇所は、実際はスペースです。

<デバイス名>には、フラッシュメモリの設定の①デバイスリストに表示される内容(下図の赤い〇で囲まれた部分)をダブルクォーテ ーション(")で囲って入力してください。

	R;0x00100000 R;0xffe00000	R;0×00107fff R;0×ffffffff	WM WM	BI	RENESAS 32K R5F56108 Internal Data Flash(32KB) RENESAS 2M R5F56108 Internal FlashMemory(2MB)	P1 P1
<					100	>
= þ*	ウンロード時設定を	:無視する( <u>I</u> )			ワ-ታኦቺሃ <mark>R;</mark> 0x0001e000 - R;0x0001ffff	ĺ

Figure 111

#### ● <開始アドレス>

フラッシュメモリの開始アドレスを指定します。

#### ● <並列接続数>

フラッシュメモリが同一アドレスにいくつ接続されているか指定します。

#### ● <直列接続数>

フラッシュメモリが直列にいくつ接続されているか指定します。

#### ● <ワークメモリ>

書込み時にワークメモリを使用するかどうかを指定します。

USEWM	書込み時にワークメモリを使用します。		
NOUSEWM	書込み時にワークメモリを使用しません。		



### ● <アドレス範囲>

リード/ライト用プログラムを使ってアクセスするフラッシュメモリの範囲を指定します。

	Table 68
開始アドレス	フラッシュメモリの開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

#### ● <プログラム開始アドレス>

リード/ライト用プログラムの開始アドレスを指定します。

#### ● <アライメント>

リード/ライト用プログラムに渡すアドレスや範囲長のアライメントを選択します。 以下に無い場合はリード/ライト用プログラム内で調整してください。

Table 69		
NOALIGN	調整しません。	
2	2 バイトで調整します。	
4	4バイトで調整します。	
8	8 バイトで調整します。	
16	16 バイトで調整します。	
32	32 バイトで調整します。	
64	64 バイトで調整します。	
128	128 バイトで調整します。	
256	256 バイトで調整します。	
512	512 バイトで調整します。	
1024	1024 バイトで調整します。	

#### ●<停止>

リード/ライト用プログラムを開始する直前でプログラムを停止するかを指定します。

BREAK	本ソフトウェアでは使用できません。		
NOBREAK	停止せずにリード/ライト用プログラムを実行します。		



### ● <リードアクセスステータス>

リードアクセス時にもリード/ライト用プログラムを使用するか指定します。

Table 71		
READ USE	リードアクセス時にリード/ライト用プログラムを使用します。	
READ NOUSE	リードアクセス時にはリード/ライト用プログラムを使用しません。	

### ● <ステータス>

現在入力しているフラッシュメモリに対しての設定を有効にするかどうか指定します。

	Table 72
ENABLE	設定を有効にします。
DISABLE	設定を無効にします。



# 7.2.16. FMCLEAR(フラッシュメモリのクリア)

フラッシュメモリの内容をクリアします。

### ●書式



Figure 112

### ●<モード>

クリア範囲を指定します。

Table 73

ALL	<開始アドレス>を含むデバイスの全セクタをクリアします
BLOCK	<指定アドレス>を含むブロックをクリアします。

#### ●<開始アドレス>

クリア開始アドレスを指定します。

#### ● <指定アドレス>

特定のアドレスを指定して処理する場合のアドレス。

#### ● <クリアモード>

デバイス全セクタクリア時のクリア処理方法を選択します。

NOCHECK	フラッシュメモリを全てクリアします。	
DEVICE	書き込み済みのフラッシュメモリの全ブロックをクリアします。	
BLOCK	フラッシュメモリの書き込み済みブロックのみクリアします。	



# 7.2.17. FMDEL(フラッシュメモリの設定の削除)

フラッシュメモリの設定を削除します。

#### ●書式



Figure 113

#### ●<種類>

削除するフラッシュメモリの種類を指定します。

Table 75

ALL	登録してある全てのフラッシュメモリの設定を削除します。
DEVICE	デバイス名を指定して削除します。
CUSTOM	リード/ライト用プログラムを使用したフラッシュメモリの設定を削除します。

#### ● <開始アドレス>

削除するフラッシュメモリの開始アドレスを指定します。

#### ● <デバイス名>

削除するフラッシュメモリのデバイス名を「メーカー名\_サイズ\_品種その他」といった形式で指定します。 ※アンダーバーの箇所は実際はスペースです。



# 7.2.18. FMLOAD(フラッシュメモリへのダウンロードの切り替え)

フラッシュメモリの設定を一時的に有効/無効に切り替えてプログラムのダウンロード先を変更します。 このコマンドは、以下のメニュー操作をし、表示されたウィンドウ内の<ダウンロード時設定を無視する>の設定によっても変更可能 です。





Figure 114

# ● <ステータス>

フラッシュメモリへのダウンロードの有効/無効を指定します。

DISABLE	フラッシュメモリへダウンロードしません。
ENABLE	フラッシュメモリへダウンロードします。



# 7.2.19. FMFILL(フラッシュメモリのフィルアップ)

指定したアドレス範囲のフラッシュメモリに、指定したデータをフィルアップします。 コマンド実行後は、実行結果が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → フラッシュメモリ → フィルアップ	

●書式



Figure 115

## ● <アドレス範囲>

Table 77

開始アドレス	メモリフィルを行う開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

## ● <フィルデータ>

ここで指定したデータでフラッシュメモリをフィルアップします。



# 7.2.20. FMWORKAREA(フラッシュメモリのワークメモリの設定)

フラッシュメモリのワークメモリを指定します。

#### ●書式



Figure 116

#### ● <アドレス範囲>

Table 78

開始アドレス	フラッシュメモリのワークメモリの開始アドレスを指定します。
範囲長	フラッシュメモリのワークメモリの範囲を指定します。
終了アドレス	終了アドレスを指定します。

#### ● <ステータス>

ワークメモリを使用する際にデータを保存するかどうかを指定します。

SAVE	データを保存します。
NOSAVE	データを保存しません。
	ワークメモリを使用してフラッシュメモリにアクセスするとワークメモリの範囲内のデータは上書きされます。



# 7.2.21. GO(CPUの実行)

指定された開始位置から、CPUの実行を開始します。 このコマンドは、以下のメニューと同じ機能です。



Figure 117

●<モード>

実行を開始する位置を指定します

Table 80

PC 現在の PC から CPU を実行します。



# 7.2.22. LOAD(オブジェクトデータのロード)

指定されたファイルのオブジェクトデータをダウンロードします。 リロードの場合は、指定されたダウンロードリストのファイルを再ダウンロードします。 削除の場合は、指定されたリスト番号のファイルをダウンロードリストから削除します。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → <u>ダウンロード</u>	



#### ●書式

Figure 118

## ●<モード>

ダウンロードのモードを指定します。

Table 81

OBJECT	指定されたオブジェクトデータファイルをダウンロードします。	•
RELOAD	指定された番号のファイルから再ダウンロードします。	
DELETE	指定された番号のファイルをダウンロードリストから削除します。	

## ●<ファイル名>

ダウンロードするオブジェクトデータファイル名を指定します。



#### ● <番号>

モジュールリストに登録されている1から始まる番号を指定します。 QUERY コマンドの LOAD パラメータにて、番号を確認することができます。

#### ● <フォーマット>

ダウンロードするファイルのフォーマット名を指定します。 通常は AUTO を指定してください。 指定できるフォーマット名は以下のとおりです。

フォーマット名	概要
AUTO	ファイルフォーマットを自動認識します。
COFF	COFF 形式を指定します。
ELF	ELF 形式を指定します。
IHEX	インテルヘキサ形式を指定します。
MHEX	モトローラへキサ形式を指定します。
SHF	オリジナル高速ダウンロード形式を指定します。
SAUF	SAUF 形式を指定します。
BINARY <開始アドレス>	バイナリデータとして、<開始アドレス>からダウンロードします。

Table 82

#### ● <チップイレース>

	Table 83
ENABLE	ダウンロードする前にチップイレースを実行します。
DISABLE	ダウンロードする前にチップイレースを実行しません。



目的の CPU などによって対応するファイルフォーマットが異なります。 本ソフトウェアでは、OBJECT に対応しているものが使用可能です。



容量制限版でダウンロードできるオブジェクトデータのサイズは <u>1KByte</u>までです。



# 7.2.23. LOADPARAM(LOAD コマンドのパラメータ補足)

LOAD コマンドのパラメータを補足します。

# ●書式



Figure 119

## ●<モード>

Table 84

CLEAR	LOAD コマンドのパラメータ補足データを初期化します。
INQUIRY	本ソフトウェアでは使用できません。
LOADLOCAL	本ソフトウェアでは使用できません。
OFFSET	オブジェクトデータのオフセット値を設定します。ロードするモジュールがバイナリの場合は、LOAD コマンドのオ
	フセットが使用されます。<オフセット値>のデフォルトは0です。



# 7.2.24. LOG(コマンドウィンドウのロギング制御)

コマンドウィンドウの表示内容をロギングする/しない(ファイルに保存する/しない)を設定します。 ロギング機能は、ON または ADD を入力した後からの表示内容がファイルに保存されます。

#### ●書式



Figure 120

### ●<モード>

記録モードを指定します。

Table 85

ON	新しくファイルを作成してロギングを開始します。
ADD	既存のファイルに追加してロギングを開始します。
OFF	ロギングを終了します。

## ●<ファイル名>

ログを保存するファイル名を指定します。



NEWBATCH コマンドと併用する場合、同一のファイルを指定しないでください。



# 7.2.25. MEMATTR(メモリアクセスエリアの設定)

パラメータに指定された条件でメモリアクセスエリアの設定を行います。 コマンド実行後は、現在のメモリアクセスエリアの設定状態が表示されます。 このコマンドは、以下のメニューと同じ機能です。



#### Figure 121

#### ● <アドレス範囲>

Table 86	
開始アドレス	メモリアクセスエリアを割り付ける開始アドレスを指定します。
範囲長	開始アドレスからの長さをバイト数で指定します。
終了アドレス	終了アドレスを指定します。

#### ● <アクセス属性>

ACCESS	<アドレス範囲>で指定した範囲に対して、リード/ライト許可を設定します。
READONLY	<アドレス範囲>で指定した範囲に対して、リード許可/ライト禁止を設定します。
WRITEONLY	<アドレス範囲>で指定した範囲に対して、リード禁止/ライト許可を設定します。
DISABLE	<アドレス範囲>で指定した範囲に対して、リード/ライト禁止を設定します。



# ● <アクセス経路>

APB	<アドレス範囲>で指定した範囲に対して、APB 方式でアクセスを行うように設定します。
AHB	<アドレス範囲>で指定した範囲に対して、AHB 方式でアクセスを行うように設定します。



7.2.26. MESSAGEBOX(ユーザーシステムメッセージボックスの表示/非表示)

ユーザーシステム電源 ON/OFF メッセージボックスの表示/非表示を指定します。

#### ●書式





#### ●<設定>

 Table 89

 ON
 メッセージボックスの表示をします。

 OFF
 メッセージボックスの表示をしません。



# 7.2.27. MKDIR(フォルダの作成)

指定されたパスのフォルダを作成します。 このコマンドは、DOSのMKDIRコマンドと同じです。

#### ●書式



Figure 123

### ● <パス名>

作成するフォルダのパスを指定します。


## 7.2.28. NEWBATCH(バッチファイルの作成)

バッチファイルを新規作成または既存のファイルに追加します。 このコマンドでバッチファイルの作成が開始された後のコマンドは、パラメータのチェックはされますが、実際のコマンドは実行されず、コ マンドラインの文字列がファイルに保存されます。 OFF は、バッチファイルの作成を終了します。

### ●書式



Figure 124

## ●<モード>

記録モードを指定します。

Table 90

ON	新規にバッチファイルを作成し、コマンドの記録を開始します。
ADD	既存のバッチファイルに追加してコマンドの記録を開始します。
OFF	バッチファイルの作成を終了します。

## ●<ファイル名>

作成または追加するバッチファイル名を指定します。



LOG コマンドまたは SAVEWIN コマンドと併用する場合、同一のファイルを指定しないでください。



## 7.2.29. OPTION(コマンドウィンドウのオプション設定)

コマンドウィンドウでのオプションを設定します。 MORE を ON した場合は、各コマンドの実行結果が1 画面を越える時に一時停止します。 MORE を OFF した場合は、停止せずにコマンド結果を全て表示します。

### ●書式



Table 91

## ●<モード>

コマンド実行時の MORE 機能を指定します。

Table 92

ENABLE	表示結果が1画面を越える時、一時停止します。
DISABLE	表示結果が1画面を越える時、停止せずにスクロールします。

### ●例

>OPTION MORE DISABLE [DISABLE] MORE CONTROL



## 7.2.30. QUERY(各種設定状態の参照)

各種の設定状態を参照します。

### ●書式



Table 93

### ●<モード>

参照する種類を指定します。

	Table 94
各種モード	説明
ENV	現在のプローブ環境の設定状態を参照します。以下のメニューと同じ機能です。
	リソース → <u>プローブ環境の設定</u>
FM	フラッシュメモリのデバイス設定情報を参照します。
LOAD	現在登録されているダウンロードリストを参照します。以下のメニューと同じ機能です。
	リソース → <u>ダウンロード</u>
LOADPARAM	LOADPARAM コマンドの設定を参照します。
MEMATR	メモリアクセスエリアの設定情報を参照します。
MESSAGEBOX	ユーザーシステムメッセージボックスの表示/非表示の設定を参照します。
RADIX	入力基数の設定を参照します。
OPTION	現在のコマンドウィンドウのオプションの設定状態を参照します。



参照可能なモードは製品や製品の構成によって異なります。 実行可能なモードは QUERY コマンドを実行すると、一覧で確認することができます。



## 7.2.31. RADIX(入力基数の設定)

### 入力基数の設定を行います。 出力の基数には影響を与えません。



Figure 125

### ●<基数>

入力基数の10進数/16進数を指定します。

Tab	le	95
iuo		55

DECIMAL	入力の基数を 10 進数にします。
HEX	入力の基数を 16 進数にします。
	この基数のとき、10 進数を入力することはできません。 10 進数を扱いたい場合は、RADIX コマンドで 10
	進数に切り替えてください。また、その時、16 進数の数値の先頭には"0x"を付けるようにしてください。



## 7.2.32. RESET(CPU のリセット)

CPUをリセットします。 このコマンドは、以下のメニューと同じ機能です。



## ●書式



Figure 126



## 7.2.33. SAVEWIN(コマンドウィンドウのファイル出力)

現在開いているコマンドウィンドウ履歴をファイルに出力します。 LOGコマンドと異なり、既に実行された表示内容が保存対象です。

### ●書式



### Figure 127

### ●<モード>

Tabl	ρ	96
iuu		50

NEW	新規にファイルを作成し保存します。
ADD	既存のファイルに追記保存します。

### ●<ファイル名>

保存するファイル名を指定します。

### ●例

>SAVEWIN NEW C:¥ARM\_Writer¥test.log Log New <C:¥ARM\_Writer¥test.log>

>SAVEWIN ADD C:¥ARM\_Writer¥test.log Log Add <C:¥ARM\_Writer¥test.log>



NEWBATCH コマンドと併用する場合、同一のファイルを指定しないでください。



## 7.2.34. SEARCH(メモリサーチ)

指定されたメモリ範囲から、指定されたデータを検索します。 コマンド実行後は、実行結果が表示されます。 このコマンドは、以下のメニューと同じ機能です。

ツールボタン	メニューバーの操作	ショートカットキー
	リソース → メモリ/ポート → <u>メモリサーチ</u>	

●書式





### ● <アドレス範囲>

Table 97

開始アドレス	メモリサーチを行う開始アドレスを指定します
範囲長	開始アドレスからの長さをバイト数で指定します
終了アドレス	終了アドレスを指定します

#### ● <サーチデータ>

メモリサーチするデータを指定します。 文字を検索する場合は、シングルクォーテーション(')で囲みます。 文字列を検索する場合は、ダブルクォーテーション('')で囲みます。 なお、文字列の中にスペースやタブを直接入力する事はできませんが、(¥x20)や(¥x09)と言った形式で指定できます。

#### ● <表示サイズ>

メモリサーチのサイズを指定します。 検索データに文字列を指定した場合は、指定したサイズが無効となり文字列のサイズで検索を行います。

-	_			~ ~	
	2	h		(10	
		( )		90	
	<b>u</b>	~	· C	20	

BYTE	バイトデータでメモリサーチします。
WORD	ワードデータでメモリサーチします。
LWORD	ロングワードデータでメモリサーチします。

### ● <サーチ条件>

検索条件を指定します。 MATCHの検索は1バイト分アドレスを進めながら検索します。 NOMATCHの検索は<開始アドレス>からサーチデータサイズごとに検索を行います。

	Table 99
МАТСН	サーチデータと一致するデータを検索します。
NOMATCH	サーチデータと一致しないデータを検索します。
	検索データに文字列を指定できません。指定を行った場合は、データ文字列の先頭1文字データを有効と
	して検索を行います。



## ● <アクセスサイズ>

アクセスサイズを指定します。

Table 100

BYTE	バイトアクセスでサーチします。
WORD	ワードアクセスでサーチします。
LWORD	ロングワードアクセスでサーチします。



## 7.2.35. SHELLEXE(ファイルの実行)

指定されたファイルを実行します。

### ●書式



Figure 129

## ● <パス名>

実行するファイルパス名を指定します。

### ●例

>SHELLEXE C:¥ARM\_Writer¥test.exe



## 7.2.36. UPLOAD(オブジェクトデータのアップロード)

指定されたオブジェクトデータをアップロードします。 このコマンドは、以下のメニューと同じ機能です。

●書式



Figure 130

●<ファイル名>

アップロードするファイル名を指定します。

## ● <アドレス範囲>

Table 101		
開始アドレス	アップロードする開始アドレスを指定します。	
長さ	開始アドレスからの長さをバイト数で指定します。	
終了アドレス	終了アドレスを指定します。	

## ● <出力フォーマット>

出力フォーマットを指定します。

Table 102

IHEX64K	インテル HEX 64K で出力します。	
IHEX1M	インテル HEX 1M で出力します。	
IHEX4G	インテル HEX 4G で出力します。	
MHEX64K	モトローラ HEX 64K で出力します。	
MHEX16M	モトローラ HEX 1M で出力します。	
MHEX4G	モトローラ HEX 4G で出力します。	
SHF	ソフィア高速ダウンロード形式で出力します。	
BINARY	バイナリ形式で出力します。	



### ● <PC 開始アドレス>

PCの開始アドレス値を出力するかどうかを指定します。 出力フォーマットがモトローラ HEX、インテル HEX の場合、PC の値を HEX ファイルに盛り込むかどうかを選択します。 バイナリでは無効です。

### ● <PC アドレス値>

出力フォーマットがモトローラ HEX、インテル HEX の場合で、PC の開始アドレスを出力する場合、アドレス値を指定します。 バイナリでは無効です。

#### ●例

UPLOAD c:¥wp¥updata1.hex 0x1200 LENGTH 0x100 MHEX64K ENABLE\_PC 0x5678 出力ファイル名:c:¥wp¥updata1.hex 開始アドレス:0x1200 範囲長:0x100(終了アドレスは0x12FF まで) フォーマット:モトローラ HEX(64K バイト長) PC アドレス値:出力する、値は0x5678 UPLOAD updata2.hex 0x40000200 T0 0x400045FF IHEX4G DISABLE\_PC 出力ファイル名:updata2.hex(カレントフォルダに出力されます。) 開始アドレス:0x40000200 終了アドレス:0x400045FF(範囲長は0x4400 バイト) フォーマット:インテル HEX(4G バイト長) PC アドレス値:出力しない



## 7.2.37. VERIFY(ベリファイの設定)

ベリファイの設定/解除を行います。 このコマンドは、以下のメニューと同じ機能です。



### ●書式



Figure 131

## ●<設定>

ベリファイの設定/解除を指定します。

Table 103

ON	ベリファイの設定を「ベリファイ(ライト後に比較)」か「ベリファイオンリー(ライトせずに比較)」にします。
OFF	ベリファイの設定を「無効」にします。

## ● <メモリライト>

「ベリファイ」か「ベリファイオンリー」かを指定します。

 Table 104

 MEMORY\_WRITE
 ENABLE
 「ベリファイ(ライト後に比較)」にします。

 MEMORY\_WRITE
 DISABLE
 「ベリファイオンリー(ライトせずに比較)」にします。



>VERIFY ON	MEMORY_WRITE DISABLE
[ON	] VERIFY
[DISABLE	] VERIFY WRITE
>VERIFY ON	MEMORY_WRITE ENABLE
[ON	] VERIFY
[ENABLE	] VERIFY WRITE
>VERIFY OF	F
[OFF	] VERIFY



## 7.2.38. WAITBREAK(CPU がブレークするまで待つ)

CPU がブレークするまで待ちます。

このコマンドを実行すると、以下のいずれかの条件に合うまで次のコマンドを実行しません。

- CPU がブレークする。
- 指定時間を過ぎる。
- コマンドをキャンセルする。(キャンセル方法は後述)

時間を指定してコマンドを実行した場合、指定時間が過ぎても CPU がブレークしなければ、このコマンドを終了します。また、指定時間に達する前に CPU がブレークした場合は、指定時間に到達していなくてもブレークした時点でこのコマンドが終了します。

CPU がブレークしているときにこのコマンドを実行すると、このコマンドはすぐに終了します。

このコマンドを実行してから CPU がブレークするまで 3 秒以上かかると下図のようなダイアログが表示され、 CPU がブレークするのを 待っていることを示します。

ここで キャンセル ボタンをクリックする(また ESC はキーを押す)と、このコマンドを中断することができます。

※CPU は実行したままの状態です。



Figure 132

●書式



Figure 133

### ●<モード>

CPU がブレークするまで無限に待ち続けるか、指定時間を過ぎればタイムアウトするかを選択します。

	Table 105
INFINITE	CPU がブレークするまで無限に待ち続けます。
TIME <待ち時間>	指定された待ち時間(単位:ms)の間、CPU がブレークするまで待ちます。
	指定できる時間は 0~65534(ms)の間です。65535 以上の値を設定した場合、65534 に
	丸めます。



### ●例

CPU を実行し、ブレークするまで無限に待ち続けます。 >GO >WAITBREAK INFINITE CPU を実行し、ブレークするまで 500 ms 間待ちます。 >GO >WAITBREAK TIME 500



# 8. バッチ機能

本ソフトウェアは、コマンドラインで実行可能な全てのコマンドをバッチプログラムによってバッチ処理させることができます。 また、ワーク変数やシステム変数、実行制御機能を使用してバッチプログラムの実行制御ができます。

バッチ機能は、バッチファイルに記述されている命令(コマンドや変数の定義など)を一行ずつ実行していきます。 命令を実行するには、行の終端に改行文字が含まれていなければなりません。

本ソフトウェアのバッチプログラムは、実行したコマンドの結果を待たずに次のコマンドを実行していきます。 例えば、CPU がブレークした後に次の命令を実行したい場合は、以下のような記述にします。

### ●例: CPU がブレークした場合に print PC を実行する

CPU がブレークした場合に「print PC」で PC(プログラムカウンタ)の表示を行って FOR ループを抜けます。 CPU の状態を 10 回チェックしてもブレークしなかった場合は、PC の表示を行わずに次の命令を実行します。

TF CPUSTATUS==0 //CPU かフレークしているかチェックする。
print PC //PC を表示する。
FBREAK //FOR ループを抜ける。
ENDIF
wait 1 //1 秒待つ。
NEXT \$A

コマンドウィンドウには、バッチファイルを容易に作成するための NEWBATCH コマンドがあり、コマンドウィンドウに入力したコマンドを バッチファイルへ保存する機能があります。

バッチプログラムからバッチプログラムの呼び出しも可能となっており、ネストの上限は規定していませんが、Windows リソースの範囲 に限ります。



【メモリチェックの場合の省略例】 <モード>の項目("SIZE"or"MASK"or"ASIZE")を入力した場合、その後の項目の<アクセスサイズ>や<データマ スク>など<モード>に関係している項目は省略できません。

コマンドは大文字、小文字を区別しません。

(i INFO)



## 8.1. ワーク変数

バッチプログラム内で使用するワーク変数は、バッチプログラムの起動時にパラメータとして渡される「**バッチ引数**」、1つのバッチプログ ラム内だけで有効な「**ローカル変数**」、全てのバッチプログラムで有効な「**グローバル変数**」が用意されています。 各変数のスコープのイメージを Figure 134 に示します。 バッチ引数は参照のみ可能なグローバル変数です。

[Batch1]
\$aa=5
\$BB=10
[Batch2]
\$aa=30
\$BB=20
print \$aa ←5 を表示。 print \$BB ←20 を表示。
\$0=100 ←エラー(引数は参照のみ)

Figure 134

#### ●使用可能なワーク変数

**•** /5ii

Table 106

変数の種類	変数名と内容		例
バッチ引数	\$0 \$1~\$9	:コマンドライン全体の文字列 :バッチ引数の1番目~9番目	\$0
グローバル変数	\$(先頭が英	長大文字から始まる文字列)	\$GLOBAL
ローカル変数	\$(先頭が英	を大文字以外から始まる文字列)	\$local

変数名に使える文字として有効なものは、以下の通りです。 A~Z、a~z、0~9、\_(アンダーバー)

• 179		
.\$a=0x10 .\$abc=0x50 .\$B0=[0x40000].W if(\$001==0x1234)	//ローカル変数\$a へ 0x10 を代入する。 //ローカル変数\$abc へ 0x50 を代入する。 //グローバル変数\$B0 へ 0x4000 番地のワードデータを代入する。 //ローカル変数\$001 の値が 0x1234 の時、真。	



## 8.2. システム変数

### 本ソフトウェアではバッチプログラム内で使用するシステム変数が Table 107 の様に定義されています。 これらはバッチプログラム中で参照できます。

Table 107		
変数名	値と意味	
CPUSTATUS	0 : ブレーク中 1 : 実行中	

•	列
---	---

|--|--|--|



## 8.3. ラベル

バッチプログラム内で分岐先などに使用するラベルを定義します。 ラベルは、行頭からコロン(:)で始まります。 ラベル行に、コマンドは記述できません。

### ●例

: COME\_HERE



## 8.4. コメント

バッチプログラム内でコメント行を記述する時に指定します。 コメントは、行の先頭または、途中からスラッシュを2つ(//)続けて指定します。 //以降の文字列はコメントとして扱われます。 コメント行はバッチプログラムの実行には影響しません。

### ●例

// This is Comment Line
if (\$a==0x1234) // if \$a equal to 0x1234



以下のコマンドでは、コマンドと同一行(コマンドの後ろ)にコメントを記述することができません。 batch, check, copy, dump, exit, fill, mkdir, newbatch, option, print, search, upload

以下のように記述すると、エラーになります。 batch test.bat // comment



## 8.5. 数値演算で使用可能な演算子

## 8.5.1. 演算子

アドレス式などの数値演算で使用可能な演算子は Table 108 の通りです。

		Table 108
種類	記号	意味
代数演算子	+	加算
	_	減算
	*	乗算
	/	除算
	%	剰余
比較演算子	==	等しい
	!=	等しくない
	<	より小さい
	>	より大きい
	<=	以下
	>=	以上
シフト演算子	<<	左シフト
	>>	右シフト
論理演算子	&&	AND
		OR
	&	ビットごとの AND
	I	ビットごとの OR
	^	ビットごとの XOR
	~	NOT
代入演算子		右辺を左辺に代入(※以下の代入演算も使用可能です)
	=	+=, -=, *=, /=, %=, <<=, >>=, &=,  =
その他	()	カッコ



## 8.5.2. 優先度と評価順序

C 言語の優先度とほぼ同じですが、代入演算子が独特です。 ※演算子の項目に記載しているカンマ(,)は区切りの記号です。

	Table 109
演算子	結合規則
+(符号), -(符号), ~, !(~と同じ処理)	左から右
*, *=, /, /=, %, %=	左から右
+, +=, -, -=	左から右
<<, <<=, >>, >>=	左から右
<, <=, >, >=	左から右
==, !=	左から右
&, &=	左から右
^, ^=	左から右
,  =	左から右
88	左から右
	左から右

カッコを使って式をグループ化できます。

例えば、ビットテストをする場合は、if((x&mask)==0)のように、カッコでくくることができます。



演算子と変数(または、値)の間には、スペースを挿入できません。



# 8.6. メモリ・I/O のデータ読み書き

→ 詳細は「3.4. メモリ・I/Oポートの読み書き」を参照ください。



## 8.7. 実行制御

## 8.7.1. FOR, FBREAK, NEXT(カウンタ付き繰り返し実行)

FOR 行から NEXT 行までの一連のコマンドを<条件式>が成立している間繰り返します。 FOR-NEXT 中に FBREAK が実行された場合は、ただちに FOR-NEXT のループを抜け出します。

### ●書式

FOR <ワーク変数>=<初期値> TO <条件式> [STEP <ステップ値>] コマンド.... [FBREAK] コマンド.... NEXT <ワーク変数>

Table 110								
パラメータ	説明							
<ワーク変数>	繰り返し処理をするためのカウンタ変数を指定します。							
	ワーク変数は、\$A~\$Z,\$a~\$z の 52 個のワーク変数から指定します。							
	NEXT で指定するワーク変数は、FOR で指定されたワーク変数と同じものでなければなりません。							
<初期値>	ワーク変数へ設定する初期値を符号付き整数で指定します。							
<条件式>	繰り返し処理を制御する条件式を指定します。条件式には、以下を使用することができます。							
	• 演算子							
	<ul> <li>ワーク変数</li> </ul>							
	• システム変数							
	・ メモリ・I/O データ							
	• 数值							
<ステップ値>	1回の繰り返し処理が終わった時にワーク変数へ加算する値を符号付き整数で指定します。							
	STEP を省略した場合は、1 になります。							

### ●例

FOR \$A=10 TO \$A<100 STEP 10 IF \$A==50 FBREAK ENDIF DUMP 0 LENGTH \$A NEXT \$A



## 8.7.2. WHILE, WBREAK, WEND(繰り返し実行)

<条件式>が真(0 以外)の間、WHILE-WEND 間の一連のコマンドを繰り返し実行します。 <条件式>が偽(0)であれば、ループを終了します。 WHILE-WEND 中に WBREAK が実行された場合は、ただちに WHILE-WEND のループを抜け出します。

### ●書式

WHILE <条件式> コマンド.... [WBREAK] コマンド.... WEND

	Table 111
パラメータ	。 説明
<条件式>	繰り返し処理を制御する条件式を指定します。

. \$A=0	A=0	
WHILE	ILE \$A<100	
IF	IF \$A==50	
١	WBREAK	
EN	ENDIF	
DU	DUMP O LENGTH \$A	
. \$/	\$A+=10	
WEND	ND	



## 8.7.3. GOTO(無条件分岐)

バッチプログラムを<ラベル>の行へ分岐します。

### ●書式

GOTO <ラベル>

	Table 112
パラメータ	説明
<ラベル>	バッチプログラム中の分岐先のラベル名を指定します。

:L00P			
コマンド			
GOTO LOOP			



## 8.7.4. IF, ELSEIF, ELSE, ENDIF(条件判断)

<条件式>が真(0以外)の時、ELSEIF、ELSE または ENDIF の行までコマンドを実行します。ELSEIF はいくつでも指定できます。

### ●書式

```
IF<条件式>
コマンド....
[ELSEIF <条件式>]
[コマンド....]
[ELSE]
[コマンド....]
ENDIF
```

Tab	le	1	13
iuo		-	тJ

パラメータ	説明
<条件式>	実行を制御する条件式を指定します。

IF \$A>	\$B				
DUMP	0 LENGTH \$A				
ELSEIF	\$A==\$B				
DUMP	0x10 LENGTH	\$A			
ELSEIF	\$A<\$B				
DUMP	0x20 LENGTH	\$B			
ELSE					
DUMP	0x30 LENGTH	\$B			
ENDIF					



## 8.7.5. END(全てのバッチプログラムの終了)

現在実行中のネストしているバッチプログラムを含む、全てのバッチプログラムを終了します。

## ●書式

END

END	
ENDIF	



## 8.7.6. QUIT(現在のバッチプログラムの終了)

### 現在実行中のバッチプログラムを終了します。 ネストしている場合は現在のバッチプログラムだけを終了し、呼び出し元のバッチプログラムに戻ります。

### ●書式

QUIT

●例			
IF \$A>\$B			
QUIT			
ENDIF			



# 8.8. ECHO(バッチコマンドの表示/非表示の切り替え)

バッチプログラム中のコマンドの表示/非表示を切り替えます。

### ●書式

ECHO {ON|OFF}

●例	
IF \$A>\$B	
ECHO ON	
ELSE	
ECH0 OFF	
ENDIF	



## 8.9. KEYIN(キーボードからの入力)

<文字列>が指定されている場合には、コマンドウィンドウのステータスバーにガイド文字列として<文字列>を表示し、キーボードからの入力を待ちます。ワーク変数が指定されている場合には、キーボードから入力された文字列をワーク変数へ代入します。

キーボードからの文字列は、リターンキー(Enter)が入力されると入力終了となります。

<文字列>と<ワーク変数>が共に指定されなかった場合は、入力された文字列を式として評価し、結果を表示します。 この時、代入式などが指定されると、左辺の式へ右辺の式の結果が代入されます。

●書式

KEYIN [<文字列> [<ワーク変数>]]

lable 114		
パラメータ	説明	
<文字列>	コマンドウィンドウのステータスバーに表示するガイド文字列を指定します。	
<ワーク変数>	キー入力された値を設定するためのワーク変数を指定します。	

- 1 1 1 1 1

●例

KEYIN "A=" \$A //コマンドウィンドウに A=と表示してキー入力待ちになる。



## 8.10. PRINT(文字列の表示)

指定された<数値式>を評価し、<フォーマット>で指定された書式でコマンドウィンドウへ表示します。 <文字列>が指定されている場合には、<数値式>が表示される前にコマンドウィンドウへ表示されます。 <文字列>、<数値式>は空白で区切っていくつでも指定する事ができます。

### ●書式

PRINT {[<文字列>] [<数值式>][<フォ-マット>]} +

パラメータ	説明
<文字列>	コマンドウィンドウへ表示するガイド文字列を指定します。
<数值式>	キー入力された値を設定するためのワーク変数を指定します。
<フォーマット>	数値式を表示するときのフォーマットを指定します。

Table 115

#### ● <フォーマット>

Table 116

書式	説明
なし	デフォルトの表示形式。 4 バイト 16 進数値とカッコで囲んで符号つき 10 進数値を表示する。
.#B	2 バイトの2 進数値で表示する。
.#LB	4 バイトの2 進数値で表示する。
.#D	2 バイトの符号付き 10 進数値で表示する。
.#LD	4 バイトの符号付き 10 進数値で表示する。
.#U	2 バイトの符号なし 10 進数値で表示する。
.#LU	4 バイトの符号なし 10 進数値で表示する。
.#H	2 バイトの 16 進数値で表示する。
.#LH	4 バイトの 16 進数値で表示する。

// 実行結果
// 実行結果
// 実行結果
// 実行結果
1 // 実行結果



# 8.11. BEEP(BEEP 音を鳴らす)

ビープ音を鳴らします。

### ●書式

BEEP

### ●例

IF \$A > \$B			
BEEP			
ENDIF			



BEEP 音を鳴らす時は、Windows のサウンド設定で「一般の警告音」を設定する必要があります。



# 8.12. WAIT(バッチプログラムの一時停止)

指定された秒数だけバッチプログラムを停止します。

### ●書式

WAIT <秒数>

	Table 117
パラメータ	説明
<秒数>	バッチプログラムを停止する秒数を指定します。

IF \$A > \$B	
WAIT 10	//10 秒待つ。
ENDIF	



# 9. スタンドアロン機能

スタンドアロン動作とは、書き込みデータや書き込み手順をプローブに記録させておき、ホスト PC と接続しなくても電源を供給した 状態でプローブの RUN ボタンを押すことによって、記録させておいた処理を単体で実行するものです。

### 1) 書き込みデータと書き込み手順の記録

ホスト PC に接続し、通常の手順でターゲットにデータを書き込みます。 このとき、書き込みデータと書き込み手順をプローブに記録します。



Figure 135

### 2) ホスト PC から切り離す

ホスト PC から切り離し、ターゲットと接続します。





### 3) 電源を接続し、RUN ボタンを押す

例えば充電池で電源を供給し、プローブの POWER LED が点灯したことを確認して RUN ボタンを押します。




#### 4) 書き込み処理を行います

書き込み中は、STATUS LED が点滅します。



Figure 138

#### 5) 書き込みが完了すると、結果を LED で通知します

正常終了時は、STATUS LED が緑色に点灯し、エラー終了時は ERROR LED が赤色に点灯します。



Figure 139



### 9.1. 手順の記録

本ソフトウェアを起動すると、以下の画面が表示されます。 赤丸のボタンをクリックすると操作手順を記録するモードになります。 この状態でプロジェクトファイルを開き(または新規作成して)、**フラッシュメモリへの書き込みを一度実行します**。



Figure 140

記録中は、ウィンドウタイトル部に「短押記録中」と表示されます。



Figure 141

### 9.2. 記録の終了

記録を終了するには、本ソフトウェアを閉じるか、プロジェクトを閉じます。



### 9.3. 記録した手順のバックアップとリストア

記録させた手順は、ホストPCにバックアップすることができます。

また、バックアップした手順をリストアすることもできます。

リストアは別のプローブにも行えますが、リストア先のプローブに本ソフトウェアの License Code が登録されていないと RUN ボタンを 押してもエラーになります。



Figure 142

т I		11	$\sim$
lab	le	$\perp \perp$	δ

-ב־א	説明
短押記録開始	このメニューを選択すると、手順記録モードになります。
本体メモリアップロード	プローブに記録した手順をホスト PC にアップロード(=バックアップ)します。
本体メモリダウンロード	ホスト PC にアップロード(=バックアップ)した手順をプローブにダウンロード(=リストア)します。



スタンドアロン機能と、手順のバックアップ、リストアは以下の様な場合に便利です。

- 工場で書き込みを並行で行う場合
- 保守・修理でメモリ内容を頻繁にリストアする場合



# 改訂履歴

版数	改訂日	改訂内容
01	2014/09/30	初版。
02	2014/10/20	7.2.22.LOAD の書式を修正



## 製造者情報



# 株式会社 Sohwa & Sophia Technologies

	〒215-8588
[本社]	神奈川県川崎市麻生区南黒川 6-2
	ホームページ: <u>http://www.ss-technologies.co.jp</u>

#### 子会社



Unit 5-2, Level 5, Tower 6, Avenue 5, The Horizon, Bangsar South No.8, Jalan Kerinchi 59200, Kuala Lumpur, Malaysia

HomePage : http://www.sohwa-m.com.my/