Universal Probe

ソフトウェア・ユーザーズ・マニュアル

Memory Command Builder

Copyright © 2014 Sohwa & Sophia Technologies Inc.

No. J090962-03



目 次

使用上の注意 略語・用語・記載ルール 6 1. 本ソフトウェアの概要 7 1.1.本ソフトウェアの概要 7 1.1.本ソフトウェアの概要 7 1.1.本ソフトウェアの概要 7 1.1.本ソフトウェアの概要 7 1.1.本ソフトウェアのの概要 7 1.1.本ソフトウェアのの概要 7 1.1.本ソフトウェアの概要 7 1.1.本ソフトウェアのの概要 7 1.1.本ソフトウェアのの概要 7 1.1.本ソフトウェアのの概要 7 1.1.本ソフトウェアのして 12 1.2.システム要件 13 1.3.J.ARN CPU 内蔵ファシュメモリの書も込み概要 14 1.3.J. Nem OFU 内蔵フルメモリの書も込み概要 16 1.3.S. PIト/ライト用フログラムの内部構成 21 1.3.S. U-ト/ライト用フログラムの内部構成 21 1.1.正動方法 21 2.1.正動方法 22 2.2.起動方法 22 2.3. 起動方法 22 3.1. Memory Command Builder ウィンドウ 23 3.1. Memory Command Builder ウィンドウ 23 3.1. Memory Command Builder ウィンドウ 21 3.3. バランキロ酸定 アイル作成情報ウィンドウ 32 4. ファイルの保存と読み込み 34 5. 印刷 36 5.1.	注	意事項	4
略語・用語・記載ルール 6 1. 本ソフトウェアの概要 7 1.1. キソフトウェアのでついて 7 1.2. システム要件 7 1.3. クリンドウェアについて 7 1.3. キリンドウェアについて 7 1.3. キリンドウェアについて 7 1.3. キリンドウェアについて 7 1.3. キリンドウェアについて 7 1.3. ARM CPU 内蔵フラッシュメモリの書き込み概要 10 1.3.3. SPI シリアルメモリの書き込み概要 11 1.3.5. リードノライト用ブログラムの内部構成 21 1.3.5. リードノライト用ブログラムの内部構成 21 2.1. 起動方法 22 2.2. 起動方法 21 2.1. 起動方法 22 2.2. 起動方法 22 2.3. 起動後の操作方法 22 3.1. Memory Command Builder ウィンドウ 23 3.1. Memory Command Builder ウィンドウ 23 3.1. / アメーク以び保存と読み込み 34 4. ファイルの保存と読み込み 34 4. ファイルの保存と読み込み 34 5. 印刷 36 5.1. 印刷ダイアログ 35 5.2. ベージ設定ダイアログ 35 5.3. 印刷ブレビュー 35 6. コマンドー覧表 40 6. コマンドー覧表 40 6. コマンドー覧表		使用上の注意	5
1. 本ソフトウェアの概要 7 1.1 本ソフトウェアについて 7 1.2 システム要件 7 1.3.1 ARM CPU 内蔵フラッシュXモUの書き込み概要 1 1.3.1 ARM CPU 内蔵フラッシュXモUの書き込み概要 1 1.3.2 ARM CPU 外部フラッシュXモUの書き込み概要 1 1.3.3. SPI シUTNXEVの書き込み概要 1 1.3.4 内蔵 RAM の用途について 11 1.3.5. リード/ライト用プログラムの内部構成 21 2.1 起動方法 21 2.2 起動方法 21 2.3 起動後の操作方法 22 3.1 Memory Command Builder ウィンドウ 23 3.1. Memory Command Builder ウィンドウ 24 3.2. デバイス情報ウィンドウ 25 3.4. ファイルの保存 34 4.1. ファイルの保存 34 4.2. ファイルの保存 34 4.2. ファイルの協动込み 35 5.1 印刷ダイアログ 35 5.2 ページ酸定ダイアログ 35 5.3. 印刷フレビュ 35 6. コマンドー覧表 40 6. コマンドー覧表 40 6. コマンドー 35	略	語・用語・記載ルール	6
1.1. キンフトウェアについて. 1.2. システム要件 1.3. フラッシュメモリ書き込みの仕組み 1.3. フラッシュメモリの書き込み概要 1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要 1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要 1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	1.	本ソフトウェアの概要	7
1.3. フラッシュメモリ書き込みの仕組み 1 1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要 1 1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要 1 1.3.3. SPI シリアルメモリの書き込み概要 1 1.3.4. 内蔵 RAM の用途について 1 1.3.5. リード/ライト用プログラムの内部構成 21 2.1. 起動方法 21 2.1. 起動方法 21 2.1. 起動方法 21 2.1. 起動方法 22 2.2. 起動時の画面 22 2.3. 起動後の操作方法 23 3.1. Memory Command Builder ウィンドウ 21 3.1. Memory Command Builder ウィンドウ 22 3.1. / Tyr/Lorg存と読み込み 34 4.1. ファイルの保存と読み込み 34 4.1. ファイルの保存 34 4.2. ファイルの読み込み 34 5. 印刷 36 5.1. 印刷ダイアログ 35 5.1. 印刷ダイアログ 35 5.2. ベージ設定ダイアログ 35 5.3. 印刷ブレビュ 35 6. コマンドー覧表 40 6.1. JTAG 接続 40		 1.1. 本ソフトウェアについて 1.2. システム要件 	7 7
1.3.1. ARM CPU 9MB フラッシュメモリの書き込み概要 1 1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要 1 1.3.3. SPI シリアルメモリの書き込み概要 1 1.3.4. 内蔵 RAM の用途について 1 1.3.5. リード/ライト用プログラムの内部構成 21 2. 起動方法 21 2.1. 起動方法 21 2.1. 起動方法 22 2.2. 起動時の画面 22 2.3. 起動後の操作方法 22 3.1. Memory Command Builder ウィンドウ 21 3.1. Memory Command Builder ウィンドウ 21 3.1. Memory Command Builder ウィンドウ 21 3.3. バラメータ設定ウィンドウ 30 3.4. フアイルの保存と読み込み 34 4. フアイルの保存 34 4.1. ファイルの保存 34 4.2. ファイルの保存 36 5.1. 印刷 36 5.1. 印刷ダイアログ 36 5.1. 印刷ダイアログ 36 5.2. ページ設定ダイアログ 37 5.3. 印刷 アレビュー 38 6. コマンドー覧表 40 6. コマンドー覧表 40 6. コマンドー 40		1.3. フラッシュメモリ書き込みの仕組み	8
1.3.3. SPI シリアルメモリの書き込み概要 14 1.3.4. 内蔵 RAM の用途について 11 1.3.5. リード/ライト用プログラムの内部構成 21 2. 起動方法 21 2.1. 起動方法 22 2.2. 起動時の画面 22 2.3. 起動後の操作方法 23 3. 面面の説明 23 3.1. Memory Command Builder ウインドウ 22 3.2. デバイズ情報ウィンドウ 21 3.3. /「ラメータ設定ウィンドウ 22 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ 32 3.4. ファイルの保存と読み込み 34 4.1. ファイルの保存 34 4.2. ファイルの保存 36 5. 印刷 36 5.1. 印刷ダイアログ 37 5.2. ページ設定ダイアログ 37 5.3. 印刷ブレビュ 36 5.1. 印刷ダイアログ 37 5.3. 印刷ブレビュ 36 5.1. 印刷ダイアログ 37 5.3. 印刷ブレビュ 37 5.		1.3.1. ARM CPU 内蔵フラッシュメモリの書さ込み概要 1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要	8 12
1.3.4. Pjie RAM 0/H2EL 0/1 (1) 1.3.5. リード/ライト用プログラムの内部構成		1.3.3. SPI シリアルメモリの書き込み概要	16
2. 起動方法 21 2.1. 起動方法 2 2.2. 起動時の画面 2 2.3. 起動後の操作方法 23 3. 画面の説明 23 3.1. Memory Command Builder ウィンドウ 21 3.2. デバイZ情報ウィンドウ 21 3.3. パラメータ設定ウィンドウ 21 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ 31 3.4. ファイルの保存と読み込み 34 4.1. ファイルの保存 34 4.2. ファイルの読み込み 36 5. 印刷 36 5.1. 印刷ダイアログ 32 5.2. ページ設定ダイアログ 32 5.3. 印刷ブレビュー 36 5.1. JTAG 接続 40 6. コマンドー覧表 40 6.1. JTAG 接続 40		1.3.4. 内蔵 RAM の用速について	17
2.1. 起動方法	2.	起動方法	21
3. 画面の説明 23 3.1. Memory Command Builder ウィンドウ 21 3.2. デバイス情報ウィンドウ 21 3.3. パラメータ設定ウィンドウ 21 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ 31 4. ファイルの保存と読み込み 34 4.1. ファイルの保存 34 4.2. ファイルの保存 36 5. 印刷 36 5.1. 印刷ダイアログ 36 5.2. ページ設定ダイアログ 37 5.3. 印刷プレビュー 36 6. コマンドー覧表 40 6.1. JTAG 接続 40		 2.1. 起動方法 2.2. 起動時の画面 2.3 記動後の操作方法 	21 21 21
3.1. Memory Command Builder ウィンドウ 2: 3.2. デバイス情報ウィンドウ 2: 3.3. パラメータ設定ウィンドウ 3: 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ 3: 4. ファイルの保存と読み込み 34 4.1. ファイルの保存 3: 4.2. ファイルの読み込み 3: 5. 印刷 3: 5. 印刷 3: 5. 印刷ブレビュー 3: 6. コマンドー覧表 4: 6.1. JTAG 接続 4: 6.2. ON 地域 4:			
4. ファイルの保存と読み込み 34 4.1. ファイルの保存 34 4.2. ファイルの読み込み 31 5. 印刷 36 5.1 印刷ダイアログ 36 5.2. ページ設定ダイアログ 37 5.3. 印刷プレビュー 36 6. コマンドー覧表 40 6.1. JTAG 接続 40	3.	画面の説明	23
4.1. ファイルの保存 34 4.2. ファイルの読み込み 35 5. 印刷 36 5.1. 印刷ダイアログ 36 5.2. ページ設定ダイアログ 36 5.3. 印刷プレビュー 36 6. コマンドー覧表 40 6.1. JTAG 接続 40	3.	 画面の説明	23 23 28 30 32
5. 印刷 36 5.1. 印刷ダイアログ	3.	 画面の説明	23 23 28 30 32 34
5.1. 印刷ダイアログ	3. 4.	 画面の説明 3.1. Memory Command Builder ウィンドウ 3.2. デバイス情報ウィンドウ 3.3. パラメータ設定ウィンドウ 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ ファイルの保存と読み込み 4.1. ファイルの保存 4.2. ファイルの読み込み 	23 23 30 32 34 34 35
6. コマンドー覧表	3. 4. 5.	 画面の説明. 3.1. Memory Command Builder ウィンドウ. 3.2. デバイス情報ウィンドウ. 3.3. パラメータ設定ウィンドウ. 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ. ファイルの保存と読み込み. 4.1. ファイルの保存. 4.2. ファイルの読み込み. 	23 23 30 32 34 34 35 36
6.1. JTAG 接続	3. 4. 5.	画面の説明. 3.1. Memory Command Builder ウィンドウ. 3.2. デバイス情報ウィンドウ. 3.3. パラメータ設定ウィンドウ. 3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ. ファイルの保存と読み込み. 4.1. ファイルの保存. 4.2. ファイルの読み込み. 5.1. 印刷ダイアログ. 5.2. ページ設定ダイアログ. 5.3. 印刷プレビュー.	 23 23 28 30 32 34 35 36 36 37 38
5.2. SPI 接続	 3. 4. 5. 6. 	画面の説明	 23 23 28 30 32 34 34 35 36 36 37 38 40



改訂履歴	
製诰者情報	



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。
1 強制	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。 範囲外の電源電圧を加えると、破損・火災の恐れがあります。
1 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。
公 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。
公 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。
公 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。
人	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。
人	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。
父 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 .
- K(大文字) ··· 2¹⁰=1024 を表します。(例:16K=16384) ٠
- 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。
 •
- •
- … xxxxx というウィンドウ内の項目名を示します。 <xxxxx>

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。

	Table 1
略語・用語	説明
本ソフトウェア	Memory Command Builder のこと。
フラッシュメモリ	Flash メモリ、EEPROM などの総称。
プローブ	Universal Probe 本体のこと。
ターゲット	Universal Probeによって制御、計測する対象のこと。



1. 本ソフトウェアの概要

1.1. 本ソフトウェアについて

当社の製品であるデバッガ(WATCHPOINT)やフラッシュメモリライターは標準でいくつかのフラッシュメモリ、SPI シリアルメモリへのアクセスをサポートしています。

本ソフトウェアを使用することで、標準でサポートしていないフラッシュメモリや SPI シリアルメモリに対して、お客様ご自身で設定ファイルを用意することができ、標準でサポートしていないメモリにもアクセスすることが可能になります。

本ソフトウェアはフラッシュメモリとSPIシリアルメモリ操作時に必要な、フラッシュメモリ設定ファイル(*.fsh)を作成するツールです。

メモリヘアクセスする処理は、「リード」、「ライト」、「ブロックイレース」、「チップイレース」の4種類があります。(SPI で接続する場合は「リード」、「ライト」、「ブロックイレース」、「チップイレース」、「ステータスリード」の5種類です。)

これらの処理で、どのようなコマンドをどのような手順で実行するかを組み立てる必要がありますが、操作は非常に簡単で、コマンド 一覧から実行したいコマンドをドラッグ&ドロップし、処理する順番に配置するだけです。

1.2. 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)
- .NET Framework 4.5 以上

本ソフトウェアで作成したフラッシュメモリ設定ファイルを使用するためには ARM Writer または SPI Writer、WATCHPOINT が必要です。



1.3. フラッシュメモリ書き込みの仕組み

Memory Command Builder で作成する**フラッシュメモリ設定ファイル(*.fsh)**は、当社製ソフトウェアである ARM Writer や SPI Writer で標準サポートされていないフラッシュメモリを読み書きするための設定を追加するファイルです。 JTAG/SWD インターフェースを経由して ARM CPU 内蔵のフラッシュメモリや、CPU に接続されたフラッシュメモリ(外部フラッシュメ モリ)へ書き込みを行うためには、書き込むユーザーデータ(=オブジェクトデータ)とメモリの制御プログラム(=リード/ライト用プログラ ム)を CPU の内蔵 RAM にコピーし、CPU 自身にフラッシュメモリへのメモリライト処理を実行させます。

SPI インターフェースで直接フラッシュメモリに書き込む SPI Writer には、CPU 経由で書き込む仕組みはありません。
CPU 経由でフラッシュメモリに書き込む ARM Writer には、SPI インターフェースで書き込む仕組みはありません。

1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要

INFO

ARM CPU 内蔵フラッシュメモリヘオブジェクトデータを書き込む手順の概要を説明します。



① リード/ライト用プログラムを JTAG/SWD インターフェース経由で内蔵 RAM ヘコピーします。

Figure 2

② ARM Writer で<ダウンロード前にフラッシュメモリを全て消去する>オプションにチェックが付いている場合は、最初にチップイレースを行います。チップイレース処理の先頭で、CPU のフラッシュメモリにアクセスするための初期化を実行します。





③ リード/ライト用プログラムは、CPUの初期化が終わると内蔵フラッシュメモリの消去を行います。



Figure 4

④ 消去が終わると、オブジェクトデータを JTAG/SWD インターフェース経由で"4K バイトだけ"内蔵 RAM にコピーします。



Figure 5

⑤ コピーしたリード/ライト用プログラムの書き込み処理を、ARM Writer から実行します。



Figure 6



CPU の初期化はチップイレースでも実行していますが、書き込み処理でも必要です。



Figure 7

⑦ CPU の初期化が終わると書き込みを開始します。



Figure 8



⑧ 4K バイト分のライト終わると、次の 4K バイトを ARM Writer に要求します。





⑨ 次の 4K バイトデータを JTAG/SWD インターフェース経由で内蔵 RAM にコピーします。



Figure 10

⑩ 前回書き込んだアドレスの続きから、新たな 4K バイトのデータを書き込みます。



Figure 11

⑪ オブジェクトデータを全て処理し終わるまで、⑧から⑩を繰り返します。



1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要

ARM CPU 外部フラッシュメモリヘオブジェクトデータを書き込む手順の概要を説明します。

① リード/ライト用プログラムを JTAG/SWD インターフェース経由で内蔵 RAM ヘコピーします。



Figure 12

② ARM Writer で<ダウンロード前にフラッシュメモリを全て消去する>オプションにチェックが付いている場合は、最初にチップイレースを行います。チップイレース処理の先頭で、外部フラッシュメモリにアクセするための初期化を実行します。



Figure 13



③ リード/ライト用プログラムは、CPUの初期化が終わると内部リソースを操作して、内蔵フラッシュメモリの消去を行います。



Figure 14

④ 消去が終わると、ユーザーデータを JTAG/SWD インターフェース経由で"4K バイトだけ"内蔵 RAM にコピーします。 オブジェクトデータはこの後、4K バイト単位で ARM CPU の内蔵 RAM にコピーされます。



Figure 15

⑤ コピーしたリード/ライト用プログラムの書き込み処理を、ARM Writer から実行します。







⑥ リード/ライト用プログラムは、ライト処理の前にフラッシュメモリに書き込むための CPU の初期化を行います。



Figure 17

⑦ CPU の初期化が終わるとライトを開始します。



Figure 18



⑧ 4K バイト分のライト終わると、次の 4K バイトを ARM Writer に要求します。



Figure 19

⑨ 次の 4K バイトデータを JTAG/SWD インターフェース経由で内蔵 RAM にコピーします。



Figure 20

⑩ 前回書き込んだアドレスの続きから、新たな 4K バイトのデータを書き込みます。



Figure 21

⑪ オブジェクトデータを全て処理し終わるまで、⑧から⑩を繰り返します。



SPI シリアルメモリヘオブジェクトデータを書き込む手順の概要を説明します。

① リード/ライト用プログラムをプローブの内蔵メモリヘコピーします。



Figure 22

② SPI Writer で<ダウンロード前にチップイレースを自動実行する>オプションにチェックが付いている場合は、最初にチップイレースを行います。



Figure 23

③ オブジェクトデータをプローブの内蔵メモリヘコピーします。コピーは 4K バイト単位で行われます。



Figure 24

④ コピーしたリード/ライト用プログラムの書き込み処理を、SPI Writer から実行します。



Figure 25



⑤ 4K バイト分の書き込みが終わると、次の 4K バイトを SPI Writer に要求します。





⑥オブジェクトデータを全て処理し終わるまで、③から⑤を繰り返します。

1.3.4. 内蔵 RAM の用途について

「1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要」や「1.3.2. ARM CPU 外部フラッシュメモリの書き込み概要」にある内蔵 RAM の用途について説明します。

内蔵 RAM には、ARM Writer で「ワークメモリ」と呼ばれるメモリ領域を割り当てます。 ワークメモリに割り当てる容量は、通常 8K バイトです。 8K バイトに満たない RAM 容量の CPU では、存在する RAM 容量の範囲内で設定してください。

ここでは8Kバイトのワークメモリを使用するという例で説明します。

8K バイトのワークメモリには、「パラメータ領域」、「プログラム領域」、「データ領域」が存在し、それらの領域の標準アドレスは 「Figure 27」で示すようなオフセットで配置されます。



内蔵 RAM

Figure 27



① ワークメモリの先頭アドレス

ワークメモリの先頭アドレスは内蔵 RAM の範囲内であればどこでも構いませんが、4 バイト単位のアドレスで設定する必要があります。また、ワークメモリがすべて内蔵 RAM 領域に収まるよう設定しなければなりません。 通常は内蔵 RAM の先頭アドレスを指定します。





② ワークメモリの範囲長

ワークメモリとして使用するサイズを指定します。 通常は 8K バイトを指定してください。 内蔵 RAM のサイズが 8K バイト未満の場合は、内蔵 RAM に収まる値を設定してください。

③ ワークメモリの「パラメータ領域」

ワークメモリの先頭 128 バイト(オフセット=0x0000~0x007F)は、リード/ライト用プログラムを使用する際のパラメータ領域です。 この領域はリード/ライト用プログラムとプローブ間の情報伝達などのために使用します。 ユーザーが操作する必要はありません。

Table 2				
オフセット	説明			
+0x0000	【処理開始時】リード/ライト用プログラムの動作モード設定 (プローブ→リード/ライト用プログラム) 1:リード 2:ライト 3:ブロックイレース 4:チップイレース 【処理終了時】リード/ライト用プログラムの実行結果表示 (リード/ライト用プログラム→プローブ) 0:正常終了 0以外:エラー終了			
+0x0004	処理対象アドレス 動作モードがリードの場合、その時リード処理する先頭アドレスを表します。 動作モードがライトの場合、その時ライト処理する先頭アドレスを表します。 動作モードがブロックイレース、チップイレースの場合、その時イレースするブロックやデバイスのアドレスを表します。			
+0x0008	データ格納領域の先頭アドレス 動作モードがリードの場合、リードデータはここで指定するアドレス以降に格納します。 動作モードがライトの場合、ライトデータはここで指定するアドレス移行に格納しています。			
+0x000C	データ長 動作モードがリードの場合、リードするデータの長さを指定します。 動作モードがライトの場合、ライトするデータの長さを指定します。			

18/47



+0x0010	処理終了アドレス
	処理が終了した後、ここで指定するアドレスに分岐させます。
+0x0014	(予約)
+0x0018	(予約)
+0x001C	(予約)
+0x0020	プローブ使用領域
:	ここはアクセスしないでください。
+0x003C	
+0x0040	(予約)
:	
+0x007F	

④ ワークメモリの「プログラム領域」

リード/ライト用プログラムを配置する領域です。 この領域の先頭アドレスは、必ず「①ワークメモリの先頭アドレス」+0x0080の位置になります。 ユーザーが操作する必要はありません。

⑤ ワークメモリの「データ領域」

データ領域の先頭アドレスは、以下のように決まります。

- ・ ワークメモリの範囲長が 8K バイト未満の場合 → 自動的にワークメモリのサイズの 1/2 の位置になります。
- ワークメモリの範囲長が 8K バイト以上の場合 → ワークメモリの先頭から 4K バイトの位置になります。

8K バイトのワークメモリを設定している場合は、4K バイト位置からが「データ領域」となります。

⑥ その他

ARM Writer がリード/ライト用プログラムの各処理を呼び出した直後は、ターゲットの CPU レジスタは以下の値になります。 ユーザーが操作する必要はありません。

ARM の場合

	Table 3	
レジスタ	説明	
PC	リード/ライト用プログラムの先頭アドレス (標準ではワークメモリの先頭アドレス+0x80の位置)	
R14	処理終了アドレス	
R13	データ領域の先頭アドレス	
	(データ領域の直前はリード/ライト用プログラムのスタックとして使用することができます)	
R0	パラメータ領域の先頭アドレス	
CPSR	0xD3 (IRQ, FIQ 禁止、ARM State、Supervisor Mode)	



Thumb2 の場合

Table 4

レジスタ	説明
PC	リード/ライト用プログラムの先頭アドレス(標準ではワークメモリの先頭アドレス+0x80の位置)
LR	処理終了アドレス
SP_main,	データ領域の先頭アドレス
SP_process	(データ領域の直前はリード/ライト用プログラムのスタックとして使用することができます)
R0	パラメータ領域の先頭アドレス
xPSR	0x01000000

 $\langle \mathbf{b} \rangle$

Table 3、Table 4 に記載していないレジスタは不定です。

1.3.5. リード/ライト用プログラムの内部構成

Memory Command Builderはフラッシュメモリ設定ファイル(*.fsh)を作成するツールですが、そのファイルの中に、リード/ライト用プログラムが含まれます。

リード/ライト用プログラムには、Memory Command Builder で作成した「リード」、「ライト」、「ブロックイレース」、「チップイレース」のタブの手順がひとつのバイナリコードに変換されて含まれており、このバイナリコードがワークメモリの「プログラム領域」にコピーされます。

ARM/SPI Writer は書き込みやチップイレースを実行する時、プローブを経由して、それらの手順を実行するように指示をしています。



Figure 29

Memory Command Builder は ARM CPU の初期化プログラムを自動で生成しません。したがって、各手順の先頭で CPU を 初期化する手順を組む必要があります。(※「1.3.1. ARM CPU 内蔵フラッシュメモリの書き込み概要」や「1.3.2. ARM CPU 外 部フラッシュメモリの書き込み概要」で、CPU の初期化を何度も実行しているのはこのためです。) さらに、書き込み対象のデバイスがデバイス自身の初期化を要求するものである場合は、その手順も初期化処理の中に含める必 要があります。

SPI Writer の場合は、デバイス側で要求される初期化処理以外は必要ありません。



2. 起動方法

2.1. 起動方法

「スタート」→「すべてのプログラム」→「Memory Command Builder」→「Memory Command Builder」より、Memory Command Builder を起動します。

2.2. 起動時の画面

起動時の画面を Figure 30 に示します。

ファイル(E) 編集(E) ヘルブ(H) デノバイス情報: デノバイス情報を設定してください。 リード ライト ブロックイレース チップイレース コマンドー覧 Figure 30	😔 Memory Command Builder		×
予約 (1) (1) デバイス備報を設定してください。 (1) レード ライト ブレース コマンドー覧 (1) (1) コマンドー覧 (1) (1) リード ウイレース (1) コマンドー覧 (1) (1) リード (1)	ファイル(E) 編集(E) ヘルプ(出)		
デバイス情報: デバイス情報を設定してください。 設定 リード ライト ブロックイレース チップイレース コマンド一覧			
Jマンドー覧 Jマンドー覧 Figure 30	デバイス情報: デバイス情報を設定してください。	設定	
Figure 30	リード ライト ブロックイレース チップイレース		
Figure 30	買ードー覧		
Figure 30			
Figure 30		<u>م</u>	51
Figure 30			
Figure 30		•	
Figure 30	F: 20		
	Figure 30		



2.3. 起動後の操作方法

起動後は、最初にデバイスの情報を設定する必要があります。

設定 ボタンを押すか、「デバイス情報を入力してください」と表記されているテキストボックスをダブルクリックし、[デバイス情報]ウィンドウを表示します。

[デバイス情報]ウィンドウにてプローブとターゲット間の接続タイプを指定すると、コマンド一覧に使用できるコマンドが表示されます。

Memory Command Builder	- 0 X
ファイル(E) 編集(E) ヘルプ(H)	
: 🞦 😂 🛃 🛞 🚔 🔯 🛤 🗙 🔹 🔹 🖺	
デバイス情報: LPC-Test	設定
リード ライト ブロックイレース チップイレース	
JT JT JT JT ADD AND CALL DEC DEFINE END GOTO IF INC LABEL LSHIFT NOT OR RBIT READ RET RSHIFT SBIT	*
SETSUB	
WAIT	T

Figure 31

→ 接続タイプの設定方法は「3.2. デバイス情報ウィンドウ」を参照してください。



3. 画面の説明

3.1. Memory Command Builder ウィンドウ

Memory Command Builder のウィンドウです。

$\begin{pmatrix} 6 \end{pmatrix}$
😁 Memory Command B/ roter 7
ファイル(E) 編集(E) ヘルプ(出)
デバイス情報: LPC-Test 設定
$ \mathbf{y} - \mathbf{k} \mathbf{k} \mathbf{y} - \mathbf{k} \mathbf{k} $
コマンド一覧 演算元データと演算データの論理積を演算元データにセットする : AND,演算元データ,演算データ
ADD (3) 演算データ・アドレス指定(A:0x*******)レジスタ指定(REG* 0~9) 演算データ・アドレス指定(A:0x********)レジスタ指定(REG* 0~9) 数値
DEC. (5)
END 呼び出し先: LABEL /
GOTO 2.AND
INC 換算元ナータ: 0×100000 演算テータ: 0×01010
LSHIFT
DRT
READ
RET
RSHIFT
SBIT
SET
SUB
WAIT · · · · · · · · · · · · · · · · · · ·

Figure 32

① デバイス情報

設定したデバイス名が表示されます。 テキストボックスをダブルクリック、または ________ ボタンを押すことで、デバイス情報を設定するウィンドウを表示します。

→ 詳細は「3.2. デバイス情報ウィンドウ」を参照してください。



② 処理切り替え

組み立てる処理を切り替える場合は、それぞれの処理のタブをクリックします。 編集を加えた処理のタブには、* のマークが付きます。





_			_
Th	h		5
Ia	v	IE.	_

リード	フラッシュからの読み込み処理
ライト	フラッシュへの書き込み処理
ブロックイレース	ブロック消去処理
	※フラッシュメモリ内一部メモリの消去
チップイレース	チップ消去処理
	※フラッシュメモリ内全メモリの消去
ステータスリード	ステータスデータ読み込み処理
	※「デバイス情報」の「接続タイプ」が「SPI」の時のみ表示。ステータス待ちコマンドで呼び出される。

③ コマンド一覧

組み立てリストに追加可能なコマンドが表示されます。接続タイプを設定することにより、表示されるコマンドが変わります。 コマンドを追加する方法には、メニューバーから「編集」→「追加」、ツールバーの → ボタン、コンテキストメニューの「追加」、コマン ドのドラッグ&ドロップ、コマンドのダブルクリックがあります。 コンテキストメニューの「追加」は、コマンド一覧内で右クリックすると表示されます。 コマンドを追加すると、[パラメータ設定]ウィンドウが表示されるので、パラメータを設定します。 パラメータ設定時にキャンセルを選択した場合は、コマンドの追加を取り消します。

コマンド一覧のコマンドを組み立てリストまでドラッグすると、挿入される位置が点線で表示されます。 ドロップすると、点線で表示されたコマンドの1つ上に選択したコマンドを挿入します。



Figure 34 はドラッグ&ドロップ時のウィンドウです。

C Memory Command Builder	- C X
ファイル(F) 編集(E) ヘルプ(H)	
: 🞦 📂 🛃 🚱 🖪 🕰 🛤 🗙 🔹 🔹 🖺	5
デバイス情報: LPC-Test	設定
*リード ライト ブロックイレース チップイレース	
コマンド一覧 演算元データを一1した結果を演算元データにセットする : DEC,演算元データ	
ADD (福田) (REG* 0~9)	
AND	
CALL	
DEC	
DEFINE セット先: 0x12000000 セットデータ: 0x0555	
GOTO 2.SET	
IF	
INC セット先: 0x12000000 セットデータ: 0x0555	
LABEL _ 3.SET	
LSHIFT	
NOT セット先: 0x12000000 セットデータ: 0x02AA	
OR 4.SET	
セット先: 0x12000000 セットテータ: 0x0555	
RSHIFT 5.END	
SBIT	
SET	
SUB	-
WAIT T	•

Figure 34

④ コマンド情報出力テキスト

コマンド一覧内で選択されているコマンド情報を表示します。

⑤ 組み立てリスト

パラメータを設定した各コマンドの情報を表示します。 組み立てリスト内のコマンドをダブルクリックすると、コマンドのパラメータ編集が可能です。 組み立てリスト内を右クリックすると、コンテキストメニューを表示します。 Shift キーを押しながらリスト内のコマンドを選択することで、コマンドの複数選択が可能です。 Figure 35 は複数選択時の状態です。



😑 Memory Command Builder			
ファイル(E) 編集(E) ヘルプ(H)			
デバイス情報: LPC-Test	設定		
*リード ライト ブロックイレー	-ス チップイレース		
コマンド一覧	フログラム終了		
ADD			
AND			
CALL	1.SET		
DEFINE			
END	セット先: 0x12000000 セットデータ: 0x0555		
GOTO	2.SET		
IF			
INC	ビット元: 0X12000000 ビットテータ: 0X00AA		
	3.5ET		
NOT	セット先: 0x12000000 セットデータ: 0x02AA		
OR	4.SET		
RBIT			
READ	セット先: 0x12000000 セットデータ: 0x0555		
REI	5.END		
SBIT			
SET			
SUB	· · · · · · · · · · · · · · · · · · ·		
WAIT -	۰ III >		

Figure 35

→ パラメータの設定方法については「3.3. パラメータ設定ウィンドウ」を参照してください。

⑥ メニューバー

●ファイル(F)

	Table 6	
新規作成	新規の組み立てリストを作成します。	
開く	組み立てリストのファイルを読み込みます。	
上書き保存	組み立てリストのファイルを上書き保存します。	
名前を付けて保存	現在の組み立てリストに別の名前をつけて保存します。	
フラッシュメモリ設定ファイル出力	[フラッシュメモリ設定ファイル作成情報]ウィンドウを表示します。	
印刷	印刷ダイアログを表示します	
印刷プレビュー	印刷プレビューを表示します。	
ページ設定	印刷のページ設定画面を表示します。	
終了	Memory Command Builder を終了します。	



●編集(E)

Table 7		
追加	組み立てリスト内のコマンドが選択されているとき:	
	選択されたコマンドの上にコマンドを挿入します。	
	組み立てリスト内のコマンドが選択されていないとき:	
	組み立てリストの一番下にコマンドを追加します。	
削除	組み立てリスト内で選択したコマンドを削除します。	
上に移動	組み立てリスト内の選択されたコマンドを1つ上に移動します。	
下に移動	組み立てリスト内の選択されたコマンドを1つ下に移動します。	
<u>コピー</u>	組み立てリスト内の選択されたコマンドをコピーします。	
ペースト	コピーしたコマンドを、組み立てリスト内の選択されたコマンドの上に貼りつけます。	

●ヘルプ(H)

Table 8		
ヘルプ	ヘルプ情報を表示します。	
バージョン情報	バージョン情報を表示します。	

⑦ ツールバー

Table 9			
新規作成	*]	祖み立てリスト内のコマンドをすべて削除し、新規作成します。	
開く	Ì	組み立てリストのファイルを読み込みます。	
上書き保存	H	組み立てリストのファイルを上書き保存します。	
フラッシュメモリ設定ファイル出力	\bigcirc	[フラッシュメモリ設定ファイル作成情報]ウィンドウを表示します。	
印刷	4	印刷ダイアログを表示します。	
印刷プレビュー	4	印刷プレビューを表示します。	
追加	2	組み立てリスト内のコマンドが選択されているとき: 選択されたコマンドの上にコマンドを挿入します。 組み立てリスト内のコマンドが選択していないとき: 組み立てリスト内の一番下にコマンドを追加します。	
削除	\times	組み立てリスト内で選択したコマンドを削除します。	
上に移動	4	組み立てリスト内の選択されたコマンドを1つ上に移動します。	
下に移動	-	組み立てリスト内の選択されたコマンドを1つ下に移動します。	
<u>コピー</u>		組み立てリスト内の選択されたコマンドをコピーします。	
ペースト		コピーしたコマンドを、組み立てリスト内の選択されたコマンドの上に貼りつけます。	



3.2. デバイス情報ウィンドウ

[デバイス情報]ウィンドウは、デバイス情報を設定するためのウィンドウです。

[Memory Command Builder]ウィンドウで、デバイス情報が表示されているテキスト部分をダブルクリックするか、 認定 ボタンを押すことで、 [デバイス情報]ウィンドウを表示します。

8	デバイス情報 1		
	項目名	設定値	
	デバイス名		
	接続タイプ		
	アドレス		
	範囲長		
	データアライメント		
	ワークメモリアドレス		
	ワークメモリ範囲長		
	メモリデバイスの名前		
		3 設定	4

Figure 36

Figure 36の接続タイプ以外の各項目をダブルクリックすると、設定値の入力ダイアログを表示します。

□ デバイス名	× 設定 キャンセル	
	Figure 37	
ボタンを押し、入力した項目を設定します。		



Figure 36の接続タイプ項目をダブルクリックすると、接続タイプの選択ダイアログを表示します。 プローブとターゲットの間のインターフェースを選択します。





① 項目名

Table 10			
デバイス名			
接続タイプ	プローブとターゲット間の接続インターフェースを設定します。		
	- JTAG もしくは SWD 接続の場合は JTAG を選択。		
	- SPI 接続の場合は SPI を選択。		
アドレス	対象デバイスの開始アドレスを設定します。		
範囲長	対象デバイスのアドレス範囲長を設定します。		
データアライメント	リード/ライト用プログラムの処理するデータのアライメントを設定します。		
	- 1 でアライメント調整しません。		
	- それ以外は、2 °でなければなりません。		
ワークメモリアドレス	ワークメモリの先頭アドレスを設定します。		
ワークメモリ範囲長	ワークメモリの範囲長を設定します。		

ワークメモリについては「1.3.4. 内蔵 RAM の用途について」を参照してください。

$\langle \mathbf{b} \rangle$

どちらの接続タイプでもフラッシュメモリ設定ファイルは生成できますが、それぞれに対応したツールをご購入頂く必要があります。

② 項目情報出力テキスト

ボタン上にあるテキストで、表内の選択された項目情報を表示します。

③ 設定ボタン

デバイス情報を更新して、[デバイス情報]ウィンドウを閉じます。

④ キャンセルボタン

ウィンドウを閉じます。



3.3. パラメータ設定ウィンドウ

パラメータ設定ウィンドウは、コマンドのパラメータを設定するウィンドウです。

😁 1.ADD	
2	
項目名	設定値
演算元データ	
演算データ	
	3 4
	設定 キャンセル

Figure 39

Figure 39の各項目をダブルクリックすると、設定値を入力するダイアログが表示されます。

30/47





① コメント

コマンドについてのコメントを入力することができます。

② 項目名

選択されたコマンドのパラメータを表示します。

③ 設定ボタン

パラメータ情報を更新して、パラメータ設定ウィンドウを閉じます。

④ キャンセルボタン

ウィンドウを閉じます。

Figure 41 はパラメータを設定したコマンドを組み立てリストに複数追加した例です。

😁 Memory Command Builder
ファイル(E) 編集(E) ヘルプ(H)
デバイス情報: LPC-Test 設定
*リード ライト ブロックイレース チップイレース
コマンド一覧 ADD AND CALL DEC
DEFINE 1番目のアドレス END セット先: 0x1200000 セットデータ: 0x0555 GOTO 2.SET IF 1番目のデータ INC セット先: 0x1200000 セットデータ: 0x00AA
LABEL 3.SET LSHIFT 2番目のアドレス NOT セット先: 0x12000000 セットデータ: 0x02AA OR 4.SET
RBIT 2番目のデータ READ セット先: 0x12000000 セットデータ: 0x0555
RSHIFT SBIT SET SET
SUB WAIT * * *

Figure 41

組み立てリスト内のコマンド表示フォーマット

(行番号).コマンド名 コメント(緑色で表示) パラメータ	

Figure 42



3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ

メニューバーから「ファイル」→「フラッシュメモリ設定ファイル出力」を選択することで、[フラッシュメモリ設定ファイル作成情報]ウィンドウ を表示します。①~④を設定した後、「作成開始」ボタンを押すことでフラッシュメモリ設定ファイルを作成します。

 フラッシュメモリ設定ファイル作成情報 命令セット 	2 エンディアン
ARM	Little
出力ファイル	3
	(4) 5 作成開始 キャンセル

Figure 43

命令セット

命令セットを選択します。接続タイプで SPI を選択している場合、この設定は無視されます。

	Table 11
ARM	Cortex-M ではないコアの場合に選択します。
Thumb-2	Cortex-M シリーズのコアの場合に選択します。

②エンディアン

エンディアンを選択します。(リトルエンディアンのみの対応です。)

	Table 12	
Little	リトルエンディアンでフラッシュメモリ設定ファイルを作成します。	

③ 出力ファイル

出力するフラッシュメモリ設定ファイルを選択します。拡張子は *.fsh です。

④作成開始ボタン

フラッシュメモリ設定ファイルを作成して、ウィンドウを閉じます。



⑤ キャンセルボタン

ウィンドウを閉じます。

	作成時に以下のダイアログが表示された場合、コマンドの記述が間違っている場合があります。
	警告メッセージ フラッシュメモリファイルが作成できませんでした。
	OK 各コマンドを確認して再度作成してみてください。



4. ファイルの保存と読み込み

4.1. ファイルの保存

メニューバーから「ファイル」→「上書き保存」、「名前を付けて保存」を選択、または、ツールバーの 🧾 ボタンを押すことでファイルを 保存します。

新規で保存を行う場合、「名前を付けて保存」を選択します。

名前を付けて保存		×
O v ↓ Txt	← Txtの検索	٩
整理 ▼ 新しいフォルダー		!≡ ▾ 🔞
☆ お気に入り ^ 名前	更新日時	種類
🚺 ダウンロード 🛛 📋 組立コマンド	2014/08/23 13:35	テキスト ドキュ
📃 デスクトップ 📄 組立コマンド2	2014/08/22 9:05	テキスト ドキュ
◎ 最近表示した場門		
 ⇒イブラリ ドキュメント ピクチャ ビデオ 		
● ミュージック 👻 < 👘		•
ファイル名(N): 組立コマンド3		•
ファイルの種類(<u>T</u>): 組立コマンドリスト(*.txt)		•
▲ フォルダーの非表示	保存(S)	キャンセル

Figure 44



4.2. ファイルの読み込み

メニューバーから「ファイル」→「開く」を選択、または、ツールバー上の 📂 ボタンを押すことで、ファイルを読み込みます。 Figure 45 のダイアログで、読み込むファイルを選択します。

開<			×
🕞 🖓 🕹 🕨 Txt		▼ fr Txtの検索	٩
整理 ▼ 新しいフォルタ	·-	8==	• 🔳 🔞
🚖 お気に入り	名前	更新日時	種類
🚺 ダウンロード	🗎 組立コマンド	2014/08/23 13:35	テキスト ドキュ
📃 デスクトップ	📄 組立コマンド 2	2014/08/22 9:05	テキスト ドキュ
1 最近表示した場所			
 ⇒イブラリ ドキュメント ビクチャ ビデオ ミュージック 			
	····································	タカマンドロフ	h (* hd)
J74.	で石(四): 祖立コイントイ	▼ 祖立コマントリス	F(*.txt) ▼
		開<(0) ▼	キャンセル

Figure 45



5. 印刷

5.1. 印刷ダイアログ

メニューバーから「ファイル」→「印刷」を選択、または、ツールバーの 📑 ボタンのクリックで、[印刷]ダイアログを表示します。 □OK ボタンを押すことで印刷を開始します。

印刷 1	
- プリンター	✓ プロパティ(P)
状態: 準備完了 種類:	
場所: コメント: 2	371ルへ出力(L)
印刷範囲	印刷語發
 すべて(<u>A</u>) 	部数(<u>C</u>): 1 <u></u>
C ページ指定(G) ページから(E)	☑ 部単位で印刷(<u>○</u>)
 ページまで(1) ○ 選択した部分(S) 	123 4 5
	OK ++>1211

Figure 46

① プリンター

印刷するプリンターを設定します。

② 印刷範囲

印刷する範囲を設定します。

③ 印刷部数

印刷する部数を設定します。

④ OK ボタン

設定した内容で、印刷を開始します。

⑤ キャンセルボタン

ウィンドウを閉じます。



5.2. ページ設定ダイアログ

メニューバーから「ファイル」→「ページ設定」を選択することで、[ページ設定]ウィンドウを表示します。

ページ設定	
	Numerican Specific Specific Termination () 1 1 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
サイズ(<u>Z</u>):	A4 (210x297mm)
給紙方法(S):	ブリンターの設定に(2)・
-印刷の向き	余白 (ミリ)
€ 縦(_)	左(L): 10 右(R): 10
C 横(A)	L(I): 10 F(3) 10 4
	OK ++>>セル

Figure 47

①用紙

印刷する用紙の大きさと、給紙方法を設定します。

2 余白

印刷するページの余白を設定します。

③ OK ボタン

画面の内容で設定します。

④ キャンセルボタン

ウィンドウを閉じます。



5.3. 印刷プレビュー

メニューバーから「ファイル」→「印刷プレビュー」を選択、またはツールバー上の ____ボタンのクリックで、[印刷プレビュー]ウィンドウを表示します。

印刷プレビュー画面を生成する際、生成しているページ数を表すダイアログ(Figure 48)が表示されます。



Figure 48



Figure 49

① 印刷ボタン

印刷を開始します。

② ズームボタン

プレビューを拡大、縮小します。



③ ページ表示ボタン

ウィンドウに表示するページ数を変更します。



Figure 50

④ 閉じるボタン

ウィンドウを閉じます。

⑤ ページ選択

指定したページに移動します。



6. コマンド一覧表

6.1. JTAG/SWD 接続

JTAG/SWD 接続向けのコマンドは Table 13 の通りです。

	Table 13	
ADD	<演算元データ>と<演算データ>の和を<演算元データ>にセットします。	
(加算)	アドレス値指定の場合"A:"を先頭につけます。	
	ADD,演算元データ,演算データ	
	演算元データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9	
	演算データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9	
	:数值指定	
AND (論理積演算)	<演算元データ>と<演算データ>の論理積を<演算元データ>にセットします。 アドレス値指定の場合"A:"を先頭につけます。	
	AND,演算元データ,演算データ	
	演算元データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9	
	演算データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9	
CALL		
CALL (呼び出し)	アドレス値指定の場合"A:"を先頭につけます。	
	CALL,呼び出し先	
	呼び出し先 : アドレス指定(A:0x0000000) : ラベル名	
DEC (デクリメント)	<演算元データ>のデクリメント(-1 した)結果を<演算元データ>にセットします。 アドレス値指定の場合"A:"を先頭につけます。	
	DEC,演算元データ	
	演算元データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9	

	Sohwa & Sophia Technologies
DEFINE	SET 命令で使用する数値を文字列に定義します。
(定義)	
	DEFINE,デファイン名,データ 1,データ 2,
	デファイン名 : データを置き換える文字列
	データ : 数値
	: 最大 6 個指定可能
	 ・を指定すると、同じアドレスにデータ1、データ2)をセットし、SET コマンドで<デファイン名 <p>>を指定すると、同じアドレスにデータ1→データ2の順番でセットします。 </p> 例えば、フラッシュメモリへのアクセスを行う前や PLL の設定を反映させるために、特定のアドレスへ 0xAA、0x55 を書き込む場合などに便利です。
END	プログラムを終了します。
(プログラムの終了)	パラメータはありません。
GOTO	指定した<ラベル>の位置へジャンプします。
(777)	GOTO,ラベル
	ラベル : ジャンプ先のラベル
IF (名作)	<比較値 1>と<比較値 2>が<条件>で示される条件で比較した時、結果が真の時に<ジャンプ
(条件)	- 先>ヘジヤンノし、偽の時は次のコイントを実行します。 アドレフ値指定の提合"Δ・"を先頭につけます
	IF,条件,比較値 1,比較値 2,ジャンプ先
	条件 : 比較条件 "EQ": <比較値 1> = <比較値 2> "LT": <比較値 1> < <比較値 2> "LT": <比較値 1> ≤ <比較値 2> "LE": <比較値 1> ≤ <比較値 2> "GT": <比較値 1> > <比較値 2> "GE": <比較値 1> ≥ <比較値 2> "NF": <比較値 1> ≥ <比較値 2>
	NL < Lt (Lt (Lt (Lt (Lt (Lt (Lt (Lt (Lt (Lt
	こ 一致恒恒正 ジャンプ先 ・ ラベルタ
INC (インクリメント)	<演算元データ>のインクリメント(+1した)結果を<演算元データ>にセットします。 アドレス値指定の場合"A:"を先頭につけます。
	INC,演算元データ
	演算元データ : アドレス指定(A:0x0000000) : レジスタ指定(REG*)
	*=0~9
LABEL	*=0~9 <ラベル名>を定義します。
LABEL (ラベル名)	*=0~9 <ラベル名>を定義します。
LABEL (ラベル名)	*=0~9 <ラベル名>を定義します。 LABEL,ラベル名 ラベル名 : ラベル名
LABEL (ラベル名)	*=0~9 <ラベル名>を定義します。 LABEL,ラベル名 ラベル名 : ラベル名

	Sohwa & Sophia Technologies
SHIFT	<演算元データ>を<シフト数>左にビットシフトして<演算元データ>にヤットします。
ビット左シフト)	アドレス値指定の場合"A:"を先頭につけます。
	LSHIFT,演算元データ,シフト数
	演算元データ : アドレス指定(A:0x00000000) : レジスタ指定(REG*)
	*=0~9
OT Amzevak	<演算元テータ>の論理合定結果を<演算元テータ>にセットします。
論理合正演算)	アトレス値指定の場合"A:"を光顕につけます。
	NOT,演算元データ
	一
	・ レンスジョル(KLG) *-0~9
R	-0-5
(論理和演算)	アドレス値指定の場合"A:"を先頭につけます。
	OR,演算元データ,演算データ
	海笛デデータ ・ アドレス指定(Δ・0×00000000)
	· しジスク指定(RFG*)
	*=0~9
	ーし、フ 演算データ ・ アドレス指定(A・0x0000000)
	: レジスタ指定(REG*)
	*=0~9
	: 数值指定
BIT	<演算元データ>の<ビット位置>のビットをクリア(=0)して、<演算元データ>にセットします。
ビットクリア)	アドレス値指定の場合"A:"を先頭につけます。
	RBIT,演算元データ,ビット位置
	演算元データ : アドレス指定(A:0x0000000)
	: レジスタ指定(REG*)
	*=0~9
	ビット位置 : 数値
EAD	指定されたデータ数だけコピーします。
データコピー)	
(, , ,	READ,転送先データ,転送元データ,データ長
	転送先/転送元データ: アドレス指定(0x0000000)
	: レジスタ指定(REG*)
	*=0~9
	データ長 : レジスタ指定(REG*)
	*=0~9
	: 数值指定
ET	呼び出し元の CALL コマンドの次のコマンドに戻ります。
戻り)	パラメータはありません。

		Sohwa & Sophia Technologies	
RSHIFT	<演算元データ>を<>	シフト数>右にビットシフトして、<演算元データ>にセットします。	
(ビット右シフト)	アドレス値指定の場合	"A:"を先頭につけます。	
	RSHIFT,演算元デー	タ,シフト数	
	演算元データ	: アドレス指定(A:0x0000000)	
		: レジスタ指定(REG*)	
	シフト教	*=0~9 · 数值	
SBIT			
(ビット設定)	アドレス値指定の場合	"A:"を先頭につけます。	
	SBIT,演算元データ,		
	演算元データ	: アドレス指定(A:0x0000000)	
		: レジスタ指定(REG*)	
	ビット/古聖	*=0~9 · 粉庙	
SFT	しット11/回 マヤットキットマンヤット=	: 致旭 ギータッをセット/ きす	
(データセット)	アドレス値指定の場合 セットデータの型を指定	"A:"を先頭につけます。また、アドレス指定の場合、接尾辞を付けることにより できます。	
	SET,セット先,セットデ	ータ	
	セット先	: 直接アドレス指定(A:0x0000000)	
		: レジスタアドレス指定(A:REG*)	
		*=0~9	
		: レジスタ指定(REG*) *-0.0	
	データ型指定接尾辞		
		: 16 ビット S(A:REGOS レジスタアドレス指定 16 ビット)	
		: 32ビット W(A:0x0000000W 直接アドレス指定 32ビット)	
	セットデータ	: アドレス指定(A:0x0000000)	
		: レンスダ指定(REG*) *-0~9	
		: 数值指定	
		: 定義名指定 (DEFINE 命令で定義した名前)	
	例:		
		この のどち 「古坊つど」 フセテ 0 ビットニ タックタン	
		JU,UXJJ 単按アドレス相圧 & Lツドナーダアグビス AAAA レジスタアドレス指定 32 ビットデータアクセス	
	SET REG0,REG1	レジスタへのレジスタ値設定	
	SET REG2, FLASH	_ADDRESS 定義名を使用したレジスタ設定	
SUB	<演算元データ>と<演	頃算データ>の減算結果を<演算元データ>にセットします。	
(減算)	アトレス旭指正の場合"A:"を光頭につけます。		
	SUB,演算元データ,源	眞算データ	
	演算元データ	: アドレス指定(A:0×0000000)	
		: レジスタ指定(REG*)	
		$*=0\sim9$	
	浜昇テータ	: アトレス指定(A:0x0000000) ・ しぶつ々指定(PEC*)	
		*=0~9	

43 / 47 Universal Probe ソフトウェア・ユーザーズ・マニュアル – Memory Command Builder

<比較値 1>と<比較値 2>が<条件>で示される条件を比較し、真になるまで待機します。 アドレス値指定の場合"A:"を先頭につけます。		
WAIT,条件,比較值 1,比較值 2		
条件 :	 : 比較条件 "EQ": <比較值 1> = <比較值 2> "LT": <比較值 1> < <比較值 2> "LE": <比較值 1> ≤ <比較值 2> "GT": <比較值 1> > <比較值 2> "GT": <比較值 1> > <比較值 2> "GE": <比較值 1> ≥ <比較值 2> "NE": <比較值 1> ≠ <比較值 2> 	
比較値 1/比較値 2 : : :	: アドレス指定(A:0x0000000) : レジスタ指定(REG*) *=0~9 : 数値指定	
指定されたデータ数だけコ WRITE,転送先データ,	ピーします。 転送元データ,データ長	
転送先/元データ : :	: アドレス指定(0x0000000) : レジスタ指定(REG*) *=0~9	
データ長 : :	: レジスタ指定(REG*) *=0~9 : 数値指定	
	<比較値 1>と<比較値 アドレス値指定の場合"A WAIT,条件,比較値 1, 条件 比較値 1/比較値 2 指定されたデータ数だけこ WRITE,転送先データ, 転送先/元データ データ長	

Sohwa & Sophia

REGO から REG9 は CPU のレジスタとは関係ありません。 また REGO から REG2 は 32 ビットの変数で、以下の初期値が設定されています。

(INFO)

REG0: 対象アドレス (リード・ライト・ブロックイレース・チップイレースで使用)

REG1: データ格納領域の先頭アドレス (リード・ライトで使用)

REG2: データバイト数 (リード・ライトで使用)

REG3~REG9は32ビットの変数で、初期化はされません。



6.2. SPI 接続

SPI 接続向けのコマンドは Table 14 の通りです。

Table 14				
CS 信号セット	CS 信号を<データ>の状態にセットします。			
	<u>データ</u> :数値={0(=Low), 1(=High)}			
コマンドセット	<ビットサイス>で指定したサイスで<コマンド>をセットします。			
	ビットサイブ・・・・、物店			
	コマンド・数値			
アドレスセット				
	ビットサイズ : 数値			
データ書き込み	指定したサイズ分のデータを書き込みます。			
	「ライト」タフでのみ使用可能です。			
	出力データサイブ ・ 数値 0けと限かし			
 データ読み込み				
	「リード」タブでのみ使用可能です。			
	入力データサイズ : 数値。0は上限なし。			
ステータス読み込み	指定したサイズ分のステータスデータを読み込みます。			
	「ステータ人リード」タフでのみ使用可能です。			
	ビットサイズ			
ステータス待ち	ステータスリードのスクリプトを繰り返し実行します。			
	指定ビット : 数値			
	セットデータ : 数値			
	(行5時間)(11)(11)(11)(11)(11)(11)(11)(11)(11)(1			
	待ち時間 : 数値。最大値 10000000 (10 秒)。			
時間待ち(ミリ秒)	指定した時間待ちます。			
	待ち時間 : 数値。最大値 10000 (10 秒)。			



改訂履歴

版数	改訂日	改訂内容
01	2014/09/29	初版
02	2014/10/23	1.2. システム要件 の以下を変更。
		- Memory を 1GB 以上に修正。
		- フラッシュメモリ設定ファイルを使用するソフトウェアに ARM Writer と SPI Writer を追加。
		3.4. フラッシュメモリ設定ファイル作成情報ウィンドウ の作成時に表示される警告メッセージの説明
		を追加。
		6.1. JTAG 接続 の以下を変更。
		- DEFINE コマンドの説明を追加。
		- SET コマンドにレジスタアドレス指定とデータ型指定接尾辞、定義名の説明および使用例を追
		ከበ。
		- アドレス指定方法補足を削除。
03	2014/11/17	誤記修正。



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/