

MJX440 for TR4102/CW4020 User's Manual

Rev.2.22 2004/03/02

変更履歴

バージョン	変更内容	変更日
2.10	● Flash 機能の内容追加	01/12/06
2.11	● 用語を修正(パラレル I/F ケーブル →パラレル インターフェース ケーブル)	02/01/08
2.12	● Solaris 版の表示対応	02/03/06
2.13	● LAN ボックス製品構成と接続の説明を追加 ● Solaris 版 MJXSERV の機能説明を追加	02/04/03
2.14	● フラッシュ メモリ書込・消去機能でサポートデバイスにインテル、 シャープのそれを追加	03/01/09
2.15	● PDFファイル名変更 (MjxEjtGJ.pdf→MjxEJtgJ.pdf) ● コマンド「変更～I/O ポート」を追加	03/08/20
2.16	● 誤字、脱字修正	03/09/04
2.20	● コンフィグレーションファイルの設定画面の変更	04/02/09
2.21	● 誤字修正	04/02/27
2.22	● 誤字修正	04/03/02

ご注意

- 本マニュアルの一部または全部を無断で複製することはできません。
- 本製品を運用した結果の影響については、いかなる責任も負いません。
- 本製品の仕様および本マニュアルの内容は予告なく変更することがあります。
- Windows は、米国 Microsoft 社の登録商標です。
- MULTI は、米国 Green Hills Software 社の登録商標です。
- Solaris は、米国 Sun Microsystems 社の登録商標です。

©1999-2004 Lightwell Co. , Ltd All rights reserved.

Printed in Japan

住所: 〒167-0051 東京都杉並区荻窪 5-20-12

TEL: 03-3392-3331

FAX: 03-3393-3878

E-mail: ZAXSupport@lightwell.co.jp

URL: <http://www.lightwell.co.jp/ZAX/>

2004 年 3 月

この度は、MJX440 for TR4102/CW4020(以下、本文中は MJX440)をご購入いただきまして、誠にありがとうございます。
とございます。

本マニュアルの内容は、次のとおりです。

第一章 概要

製品構成、MJX440 の概略、各部の名称について記述しています。

第二章 ホスト インターフェースの設定

ホスト インターフェース(PCI カード、PCMCIA カード、USB^{*1}、LAN)の取り付け方法と、そのデバイス ドライバのインストール方法について記述しています。

第三章 ハードウェアの接続

MJX440 とホストの接続方法、MJX440 とターゲット システムの接続方法について記述しています。

第四章 ソフトウェアのインストール

MJX440 を操作するソフトウェアのインストール方法について記述しています。

第五章 MJX440 の環境設定

MJX440 を使用する前に必要な、環境設定の方法について記述しています。

第六章 ソフトウェアの起動と終了

MJX440 を操作するソフトウェアの起動方法と終了方法について記述しています。

第七章 MJXDEBW コマンド

MJXDEBW コマンドの使い方について記述しています。

第八章 MJXDEBW メニューコマンド

MJXDEBW メニューコマンドの使い方について記述しています。

第九章 高速ダウンロード

高速ダウンロードを行なうための手順について記述しています。

付録

仕様、ターゲット システムの制限事項などの技術情報について記述しています。

^{*1} USB I/F は開発中です。

もくじ

第一章 概要.....	8
1.1 製品構成.....	8
1.2 MJX440 の概略.....	13
1.3 各部の名称.....	17
第二章 ホストインターフェースの設定.....	20
第三章 ハードウェアの接続.....	21
3.1 MJX440 とホストの接続.....	21
3.2 EJTAG プローブの接続.....	23
3.3 ROM プローブの接続.....	23
3.3 ROM プローブの接続.....	24
3.4 外部トリガ ケーブルの接続.....	33
3.5 電源の接続と電源投入手順.....	35
第四章 ソフトウェアのインストール.....	37
4.1 Windows の場合.....	37
4.2 Solaris の場合.....	44
第五章 MJX440 の環境設定.....	47
5.1 Windows の場合.....	47
5.2 Solaris の場合.....	50
第六章 ソフトウェアの起動と終了.....	58
6.1 Windows の場合.....	58
6.2 Solaris の場合.....	63
第七章 MJXDEBW コマンド.....	65
ABORT.....	70
BATCH.....	71
BP.....	72
BP/A.....	75
BP/H.....	77
BP/S.....	78
CLEAR.....	79
CONFIG.....	80
DUMP.....	81
EXAMINE.....	82
FILL.....	83
FLASH.....	84
GO.....	86

HISTORY.....	87
INIT.....	89
JOURNAL.....	90
LOAD.....	91
MOVE.....	92
PIN.....	93
QUIT.....	94
REGISTER.....	95
SETLOAD.....	96
SETSAVE.....	97
STEP.....	98
TRACE/M.....	99
TRACE/I.....	100
UNASM.....	102
VERSION.....	103
WAIT.....	104
XPIN.....	105
第八章 MJXDEBW メニューコマンド.....	106
ファイル(F) ~ バッチ(B).....	108
ファイル(F) ~ ジャーナル開始(S).....	109
ファイル(F) ~ ジャーナル停止(E).....	109
ファイル(F) ~ トレース結果(T).....	110
エミュレーション(E) ~ 実行(G).....	111
エミュレーション(E) ~ 再スタート(R).....	111
エミュレーション(E) ~ 中断(B).....	112
エミュレーション(E) ~ ステップ(S).....	112
エミュレーション(E) ~ N-ステップ(N).....	112
エミュレーション(E) ~ ダウンロード(L).....	114
エミュレーション(E) ~ 初期化(I).....	115
エミュレーション(E) ~ フラッシュメモリ(F) ~ 書込(W).....	116
エミュレーション(E) ~ フラッシュメモリ(F) ~ 消去(E).....	117
表示(V) ~ メモリ(M).....	118
表示(V) ~ レジスタ(R).....	120
表示(V) ~ 逆アセンブル(D).....	121
表示(V) ~ トレース結果(T).....	122
表示(V) ~ コマンド応答クリア(C).....	127
表示(V) ~ オプション(O) ~ フォント(F).....	128

表示(V) ~ オプション(O) ~ 表示(V).....	129
変更(M) ~ メモリ(M).....	130
変更(M) ~ I/O ポート(P).....	131
変更(M) ~ レジスタ(R).....	132
変更(M) ~ アセンブラ(A).....	133
変更(M) ~ フィル(F).....	134
変更(M) ~ 移動(M).....	135
設定(S) ~ ブレークポイント(B).....	136
設定(S) ~ トレース(T).....	138
設定(S) ~ 設定の読出(L).....	140
設定(S) ~ 設定の保存(S).....	141
設定(S) ~ コンフィグレーション(C) ~ 設定(S)...	142
設定(S) ~ コンフィグレーション(C) ~ 表示(V)...	144
設定(S) ~ コンフィグレーション ~ 上書き保存(E).....	145
設定(S) ~ コンフィグレーション ~ 名前を付けて保存(A).....	146
ヘルプ(H) ~ トピックの検索(H).....	147
ヘルプ(H) ~ MJXDEBW のバージョン情報(A).....	148
第九章 高速ダウンロード.....	149
付録 A 仕様.....	150
付録 B ターゲット システムの制限事項.....	151
付録 C Extended EJTAG コネクタ.....	152
付録 D ROM プローブ.....	154
付録 E 対応 ROMピンアサイン.....	157
付録 F LED.....	161
付録 G レジスタ名一覧.....	162
付録 H MJX バイナリ ファイル.....	163
付録 I ターゲット システムへのプロービング.....	164
付録 J フラッシュ メモリ デバイス情報定義 ファイル レイアウト.....	171
付録 K 対応フラッシュ メモリー覧.....	175
付録 L Solaris 版 MJXSERV 使用上の注意.....	190
検索項目.....	191

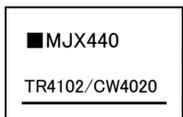
第一章 概要

製品構成、MJX440 の概略、各部の名称について記述しています。

1.1 製品構成

出荷時の MJX440 for TR4102/CW4020 パッケージの中には、次のものが含まれています。万一、欠品がございましたら、弊社までご連絡ください。

- MJX440 for TR4102/CW4020 本体



- AC アダプタ

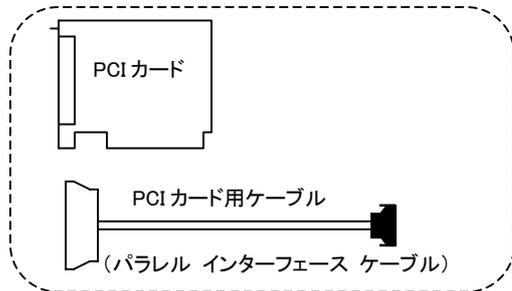


- AC コード

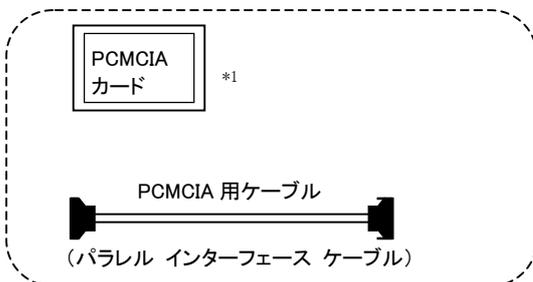


次ページへ続きます。

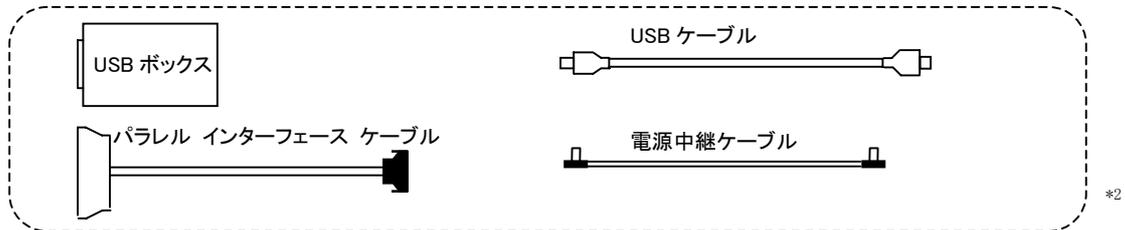
□ ホスト インターフェース



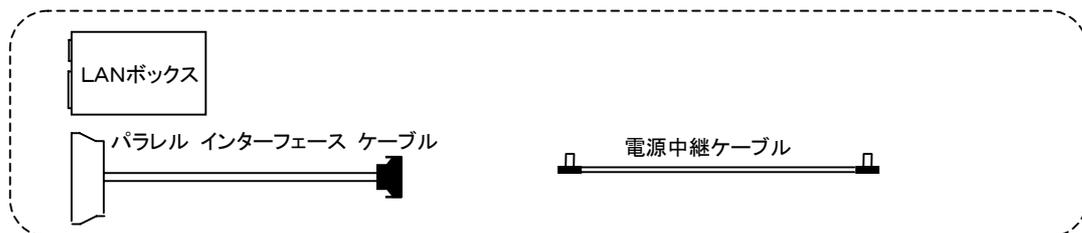
または



または



または



*1 カードの正しい名称は「PC カード」です。

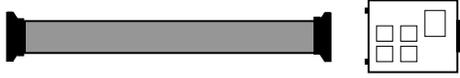
PCMCIA(Personal Computer Memory Card International Association)は PC カードの標準化団体名ですが、このマニュアルでは PC カードを全て「PCMCIA カード」と表記しています。

*2 USB I/F は開発中です。

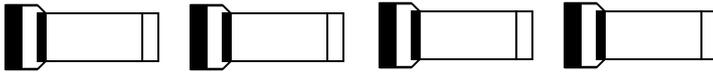
第一章 概要

□ プロブケーブルと基板

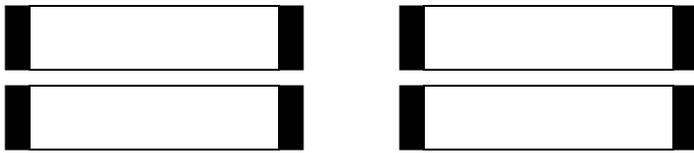
EJTAG プロブ ケーブルと EJTAG プロブ 基板



□ ROM プロブ*¹ 4本 (J-101A、J-102A、J-103AまたはJ-104A 各4本)



□ ROM プロブ ケーブル*¹ 4本



□ 外部トリガ ケーブル 2種類

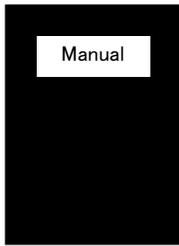


*¹ ROMの種類によって、出荷時に含まれているROMプロブが異なります。詳しくは、「表 1-1 出荷時のROMプロブ」を参照してください。

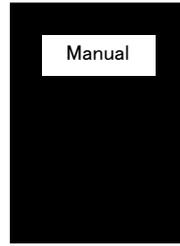
ただし、エミュレーションメモリなしモデルにはROMプロブおよびROMプロブケーブルは添付されません。

第一章 概要

- マニュアル2冊

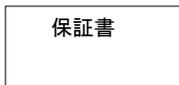


MJX440 for TR4102/CW4020
User's Manual

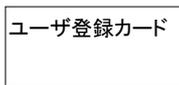


MJX440
Host Interface
Installation Manual

- 保証書

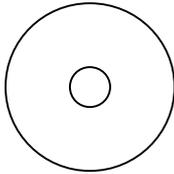


- ユーザー登録カード



【重要】 ユーザー登録カードは、必要事項をご記入の上、弊社までご返送ください。

- CD-ROM



第一章 概要

出荷時の ROM プローブは、次のとおりです。ROM の種類によって異なります。

ROM の種類	ROM プローブ*1	
	ROM プローブ基板	ROM プラグ*2
27010(8bit)	J-101A×4 本	32pin×8*3
27020(8bit)		
27040(8bit)		
27080(8bit)		
271000(8bit)		
27C4000(16bit)	J-102A×4 本	40pin×4
27C8000(16bit)	J-102A×4 本	42pin×4
27C16000(16bit)		
271024(16bit)	J-103A×4 本	40pin×4
272048(16bit)		
274096(16bit)		
27C4000(8bit)	J-104A×4 本	40pin×8*3
27C8000(8bit)	J-104A×4 本	42pin×8*3
27C16000(8bit)		

表 1-1 出荷時の ROM プローブ

*1 ROM プローブの詳細な図は「1.3 各部の名称」、「ROM プローブ」および「付録 D ROM プローブ」を参照してください。

*2 ROM のピン数と接続可能な最大 ROM 個数です。実際の接続数はターゲットシステムにより異なります。

*3 ROM プローブ基板 J-101A、J-104A は ROM プラグが 2 本あるため、同時に 2 つに接続出来ます。

1.2 MJX440 の概略

MJX440 for TR4102/CW4020 は、52ピンの Extended EJTAG コネクタを備えた TR4102 または CW4020 のターゲット システムをデバッグするための、開発支援装置です。

MJX440 のおもな特長

- ターゲット システムの [EJTAG コネクタ](#) を使用するため、接続が簡単です。
- 高速な CPU でも安定して動作します。
- 任意で、[ROM インサーキット接続](#) ができます。ROM 上のプログラムもデバッグできます。
- 高級言語デバッガ [MULTI](#) が使用できます。
- MJXDEBW デバッガで [MJXDEBW コマンド](#) が使用できます。MJXDEBW が持つ機能をフルに使用できます。
- [リアルタイム トレース](#) 機能があります。
- [ハードウェア ブレークポイント](#) 機能があります。
- プログラムのダウンロードが高速です。

JTAG 接続時 480K バイト/秒 *1 (EJTAG クロック20MHzの場合)

ROM インサーキット接続時 4M バイト/秒

最小構成

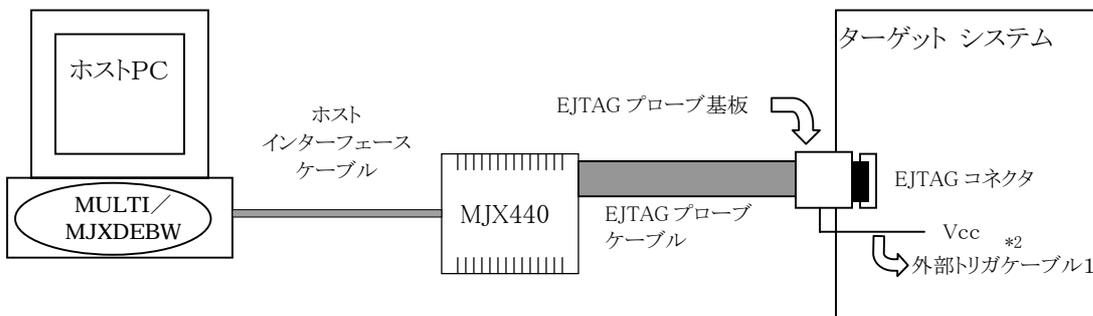


図 1-1 最小構成*1

MJX440 は図 1-1 の最小構成で動作します。最小構成で使用した場合は、ターゲット システムの RAM 上のプログラムをデバッグすることができます。プログラムは EJTAG プロブ ケーブルをとおして、ターゲット システム上の RAM へダウンロードされます。

*1 ダウンロードの速さは JTAG クロックを高く設定する程、高速になります。

*2 外部トリガケーブル1のVCC(クリップ、ケーブル共に赤)を接続する必要があります。

最小構成の場合、ROM 上のプログラムを実行することはできますが、ROM 領域にプログラムをダウンロードしたり、[ソフトウェア ブレークポイント](#)を設定することができません。ROM 上のプログラムをデバッグする場合は、さらに ROM インサーキット接続する必要があります。

ROM インサーキット接続

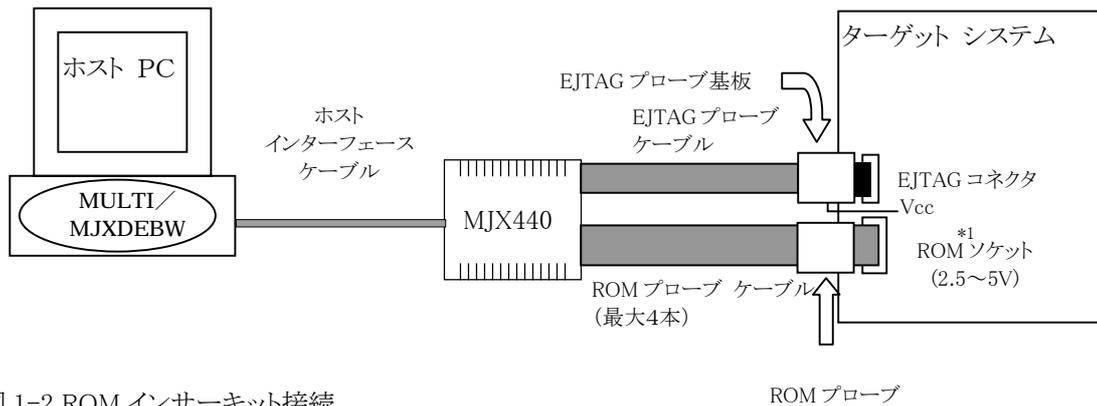


図 1-2 ROM インサーキット接続

図 1-2 のように、ROM インサーキット接続した場合は、ROM 上のプログラムもデバッグできるようになります。ROM 上のプログラムは、ターゲット システムの RAM ではなく、MJX440 内のエミュレーション メモリへダウンロードされます。

また、ROM の電圧は、2.5~5V の範囲内で自動認識されます。

外部トリガ ケーブル接続

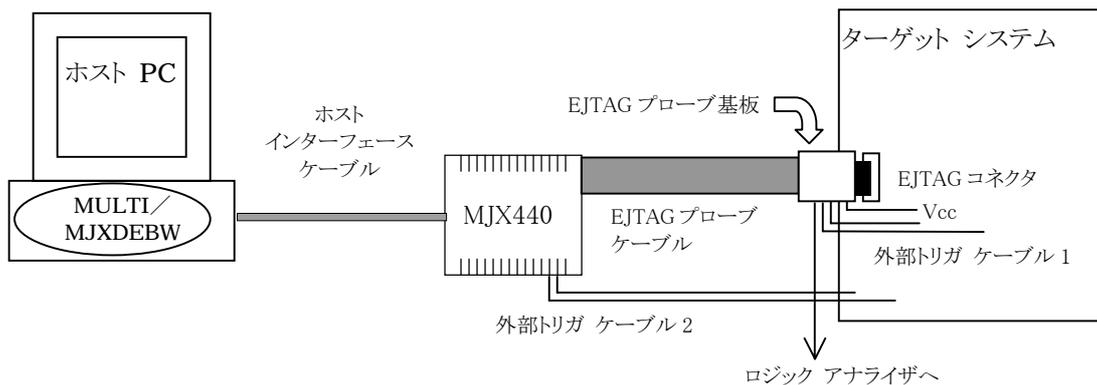


図 1-3 外部トリガ ケーブル接続

*1 外部トリガケーブル1のVCC(クリップ、ケーブル共に赤)を接続する必要があります。

第一章 概要

図 1-3 のように、二種類の[外部トリガ ケーブル](#)を接続し、次の機能を実現させることができます。

- トレース トリガを、ロジック アナライザのトリガ信号として使用する。(外部トリガ ケーブル 1 出力)
- ターゲット システムの信号の状態を、リアルタイム トレース情報として記録する。(外部トリガ ケーブル 1 入力)
- MJXDEBW コマンドで操作できる、汎用信号をターゲット システムで使用する。(外部トリガ ケーブル 2 出力)
- ターゲット システムの信号の状態を、MJX440 上の[LED](#)に表示させる。(外部トリガ ケーブル 2 入力)

外部トリガ ケーブル 1 は、[EJTAG プローブ基板](#)に接続し、外部トリガ ケーブル 2 は、MJX440 本体に接続します。

MJX440 を使用する前の準備作業

MJX440 を使用する前には、次の準備作業が必要です。第二章から第五章までを参照して行なってください。これらは、購入時に一度だけ行なえばよい作業です。

- [ホストインターフェースの設定](#)
- [ハードウェアの接続](#)
- [ソフトウェアのインストール](#)
- [MJX440 の環境設定](#)

準備作業が終了したら、[第六章](#)を参照して、MJX440 を操作するソフトウェア([MULTI](#)または[MJXDEBW](#))を起動してください。ソフトウェアが正常に起動できれば、準備作業は終了です。ソフトウェアが正常に起動できなかった場合は、準備作業に誤りがないか、確認してください。

コンパイラや MULTI の使い方については、それぞれのマニュアルとリリース ノートを参照してください。また、MJX440 独自の機能を使用するための MJXDEBW コマンドは、[第七章](#)および[第八章](#)を参照してください。

高速ダウンロードについては、[第九章](#)を参照してください。

MULTI について

MULTI はいろいろな環境で実行することができる高級言語デバッガです。サーバプログラム MJXSERV を呼び出すことによって、MJX440 の環境で実行させることができますようになります。

MJXDEBW について

MJXDEBW は MJXDEBW コマンドのみをサポートする簡易デバッガです。高級言語デバッグをしない場合や、バッチ処理機能を使って、ターゲット システムの検査をする場合などに使用することができます。

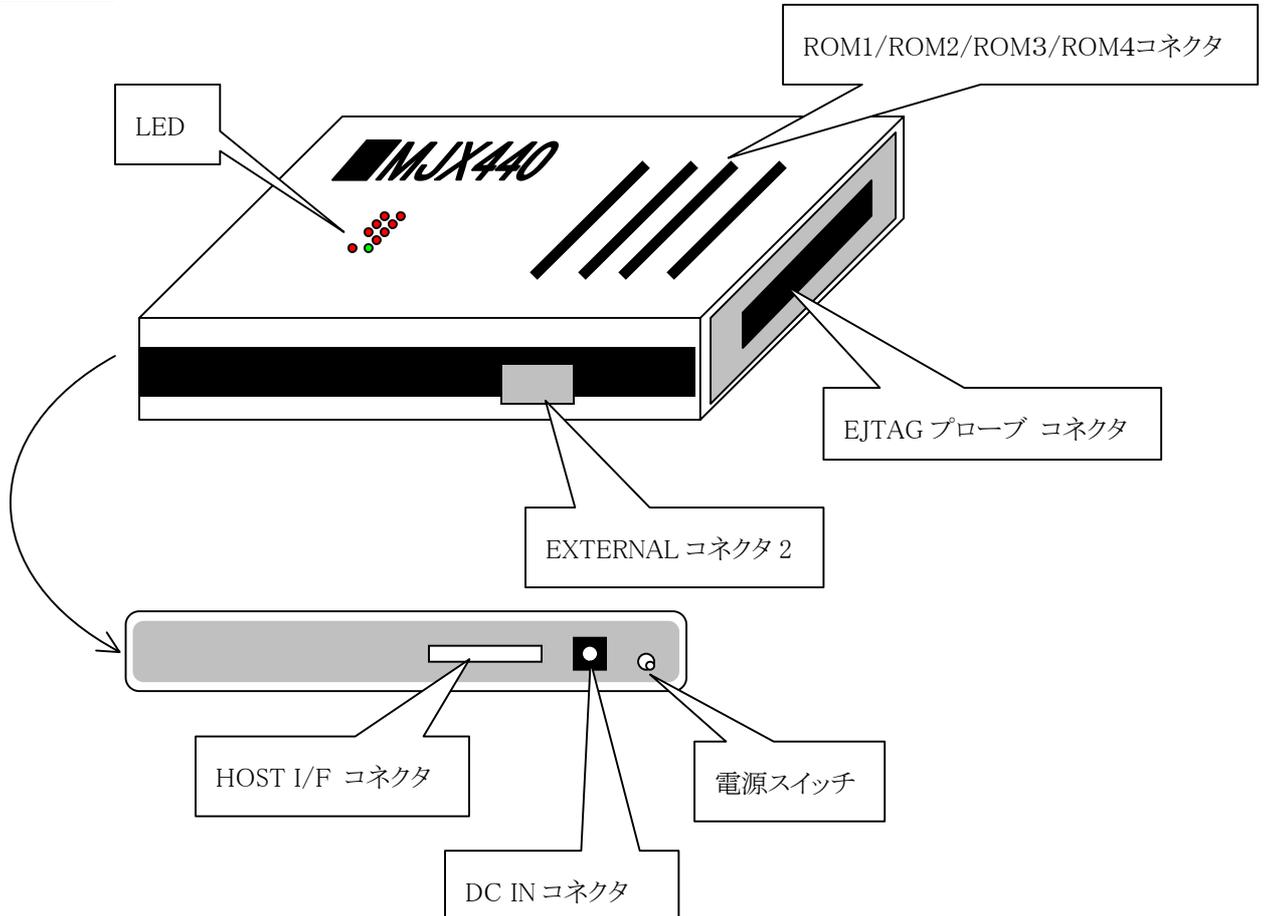
【注意】 Solaris から MJXDEBW を独立に使用することはできません。

*1

*1 Solaris 版 MJXSERV で MJX440 コマンドを実行する方法については「付録L Solaris 版 MJXSERV 使用上の注意」を参照してください。

1.3 各部の名称

MJX440 本体



電源スイッチ

ON で電源投入、OFF で電源切断します。

DC IN コネクタ

AC アダプタを接続します。

HOST I/F コネクタ

ホスト インターフェース ケーブルを接続します。

EJTAG プローブ コネクタ

EJTAG プローブ ケーブルを接続します。

ROM1..ROM4 コネクタ

ROM プローブ ケーブルを接続します。一番内側が ROM1、次が ROM2、一番外側が ROM4 コネクタです。

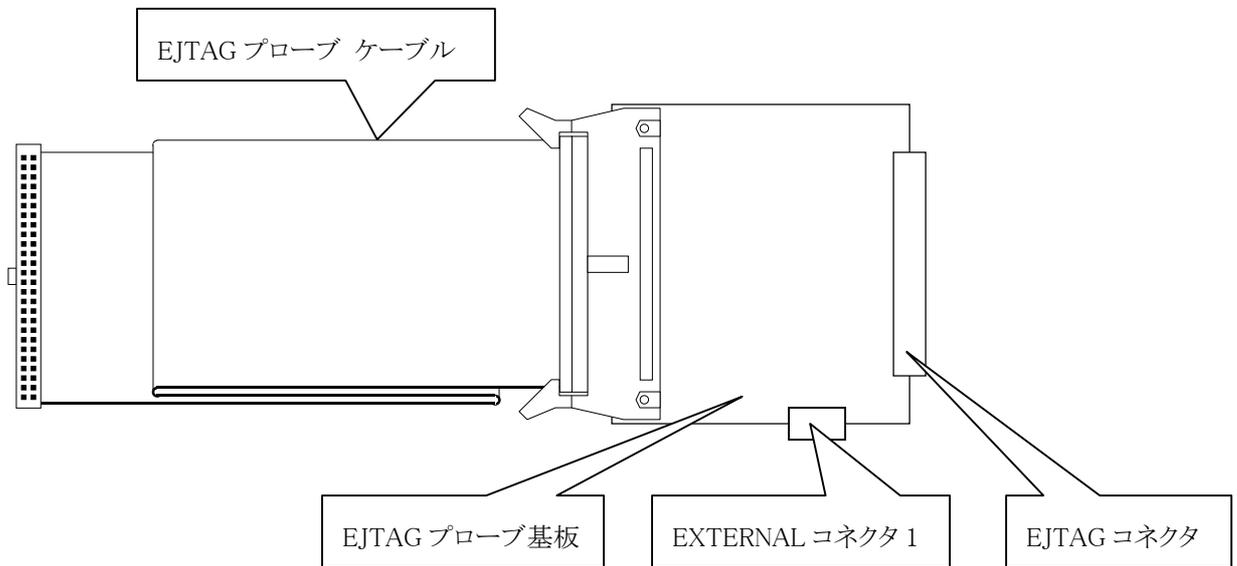
EXTERNAL コネクタ 2

外部トリガ ケーブル 2(2 列タイプ)を接続します。

[LED](#)

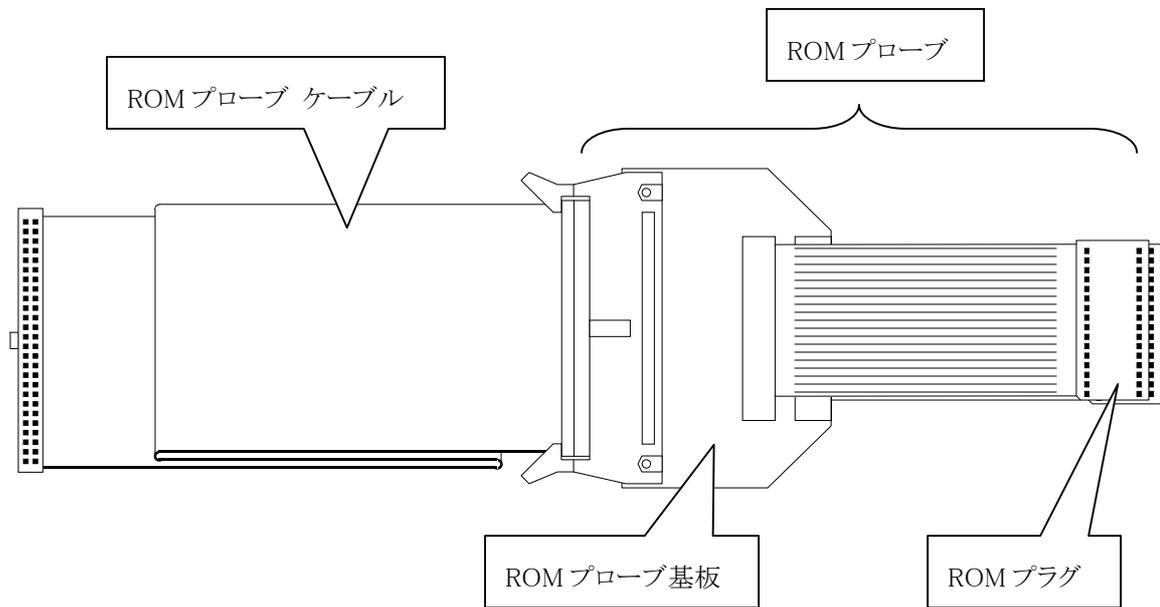
電源やコネクタの接続状態を表示します。

EJTAG プローブ



- | | |
|-----------------|---------------------------------|
| EJTAG プローブ ケーブル | MJX440 と EJTAG プローブ 基板を接続するケーブル |
| EJTAG プローブ 基板 | EJTAG プローブの基板部分 |
| EJTAG コネクタ | ターゲット システムの EJTAG コネクタへ接続する部分 |
| EXTERNAL コネクタ 1 | 外部トリガ ケーブル 1 (1 列タイプ) を接続します。 |

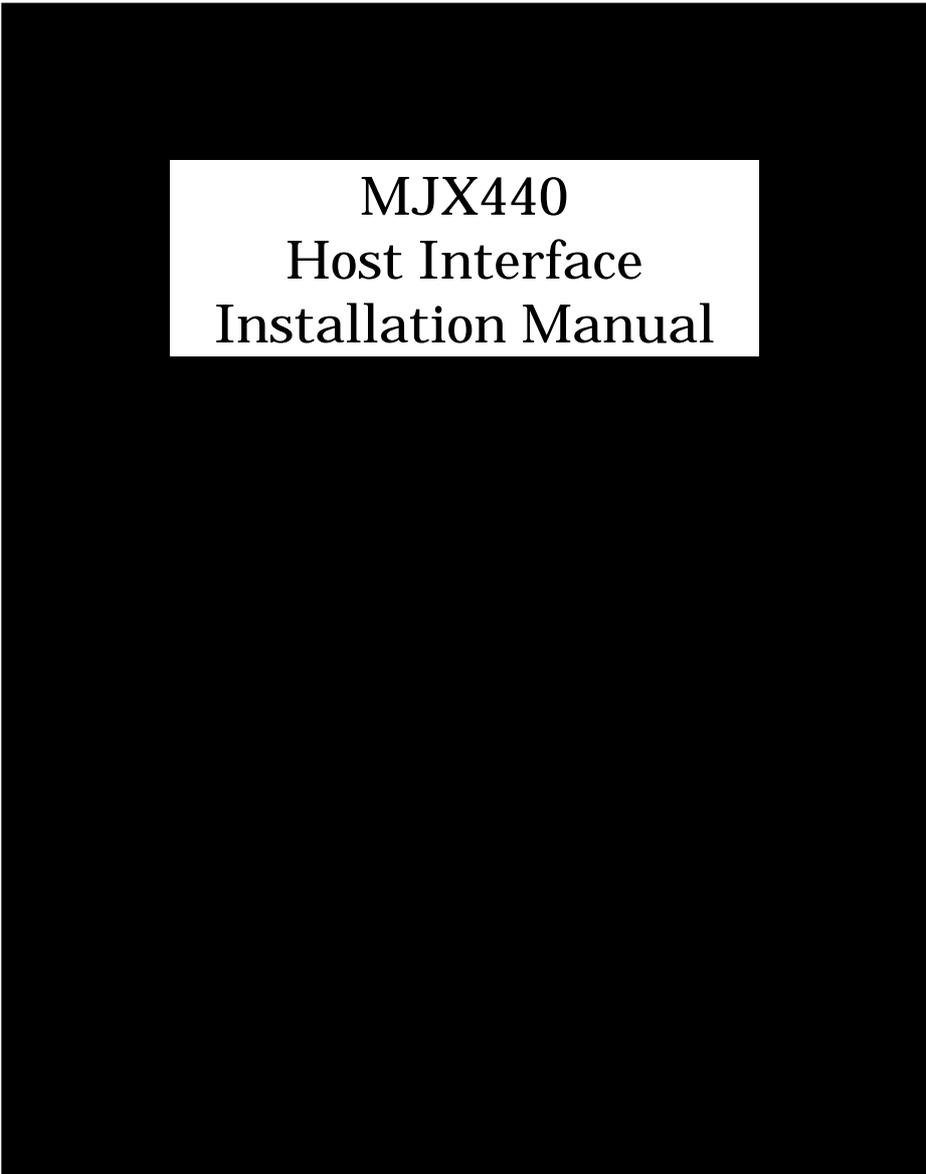
ROM プローブ



- | | |
|---------------|---|
| ROM プローブ ケーブル | MJX440 と ROM プローブ を接続するケーブル |
| ROM プローブ | ターゲット システムの ROM へ接続するプローブ全体の総称 |
| ROM プローブ 基板 | ROM プローブの基板部分 (ジャンパ設定 があります) |
| ROM プラグ | ターゲット システムの ROM ソケットへ接続する部分 |

第二章 ホストインターフェースの設定

ホストインターフェース(PCI カード、PCMCIA カード、USB ボックス*¹、または LAN ボックス)の取り付け方
法と、そのデバイス ドライバのインストール方法については、次の別冊マニュアルを参照してください。

The image shows the cover of a manual. It has a black background with a white rectangular box in the center. Inside the box, the text reads: MJX440, Host Interface, and Installation Manual, stacked vertically.

MJX440
Host Interface
Installation Manual

*¹ USB I/F は開発中です。

第三章 ハードウェアの接続

MJX440 とホストの接続方法、MJX440 とターゲット システムの接続方法について記述しています。

【重要】 MJX440 と他の機器を接続は、必ず機器の電源を切ってから行なってください。

3.1 MJX440 とホストの接続

ホストに取り付けた PCI カード、または PCMCIA カードと、MJX440 の [HOST I/F コネクタ](#) を、ホスト インターフェース ケーブルで接続してください。

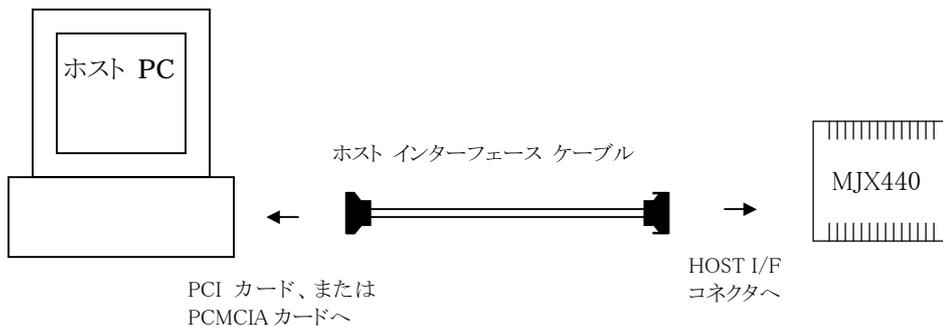


図 3-1 MJX440 とホストの接続

【注意】 PCMCIA 用ケーブルのカードとの接続コネクタの厚みにより、以下の制限事項があります。

- ◆ PCMCIA カード スロットが 2 つある場合でも、カードを 1 枚しか使用できない場合があります。また、下のスロットにしかカードを差せない場合があります。
- ◆ カード 2 枚を無理に差し込むと、PCMCIA カード スロット、および PCMCIA カードのコネクタ部が壊れる場合があります。
- ◆ PCMCIA カード スロットが 1 つだけの機種では、使用できない場合があります。

USB 接続の場合*1

ホスト PC と USB ボックスを USB ケーブルで接してください。USB ボックスの MJX I/F コネクタと、MJX440 の [HOST I/F コネクタ](#)を、パラレル インターフェース ケーブルで接続してください。

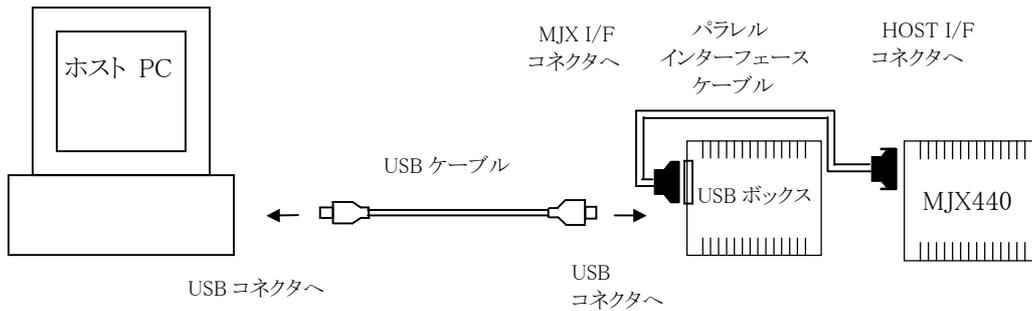


図 3-2 MJX440 とホストの接続 /USB 接続

LAN 接続の場合*2

ホストと LAN ボックスを、HUB を中継して、LAN ケーブルで接続してください。

LAN ボックスの MJX I/F コネクタと MJX440 の [HOST I/F コネクタ](#)をパラレル インターフェース ケーブルで接続してください。

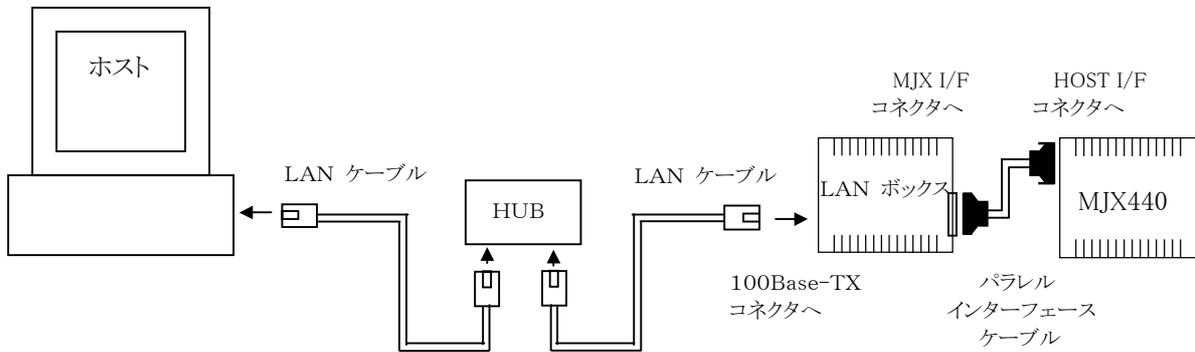


図 3-3 MJX440 とホストの接続 /LAN 接続

*1 USB I/F は開発中です。

*2 LAN ケーブルと HUB には、100Base-TX 用または 10Base-T 用が使用できます。

3.2 EJTAG プローブの接続

図 3-2 のように、MJX440 本体、EJTAG プローブ ケーブル、EJTAG プローブ基板、ターゲット システムを接続してください。

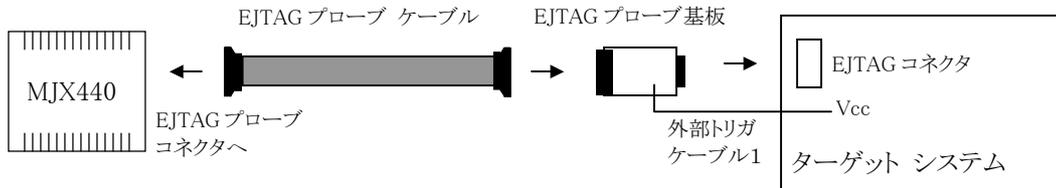


図 3-2 EJTAG プローブの接続

【注意】 EJTAG プローブ ケーブルのコネクタには向きがあります。コネクタの△マーク同士を合わせるように、接続してください。

【注意】 EJTAG プローブ基板を逆差ししないでください。コネクタ同士のピン番号が合うように接続してください。コネクタは逆差しが可能な形状のため十分に注意してください。コネクタは逆差しが可能な形状のため、十分に注意してください。

誤って接続した場合、機器を破損するおそれがあります。

【注意】 EJTAG プローブの接続に加えて、外部トリガ ケーブル 1 の Vcc(クリップ、ケーブル共に赤)を、必ずターゲット システムへ接続してください。詳細は、「3.4 外部トリガ ケーブルの接続」を参照し

3.3 ROM プローブの接続

ROM プローブの接続は、ROM インサーキット接続をする場合のみ必要な作業です。

はじめに、ROM の種類に合わせて、ROM プローブ基板のジャンパを設定してください。詳しい設定方法は、「[付録 D ROM プローブ](#)」を参照してください。

つぎに、添付されている ROM プローブを、ROM プローブ ケーブルに接続してください。

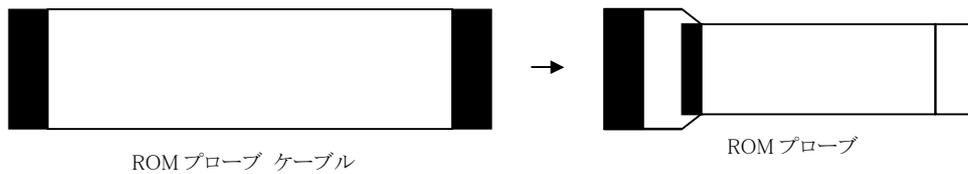


図 3-3 ROM プローブと ROM プローブ ケーブルの接続 1

つぎに、ROM プローブをターゲット システムの ROM ソケットへ接続し、もう一方の ROM プローブ ケーブルを MJX440 の ROM1/ROM2/ROM3/ROM4 コネクタへ接続します。^{*1}

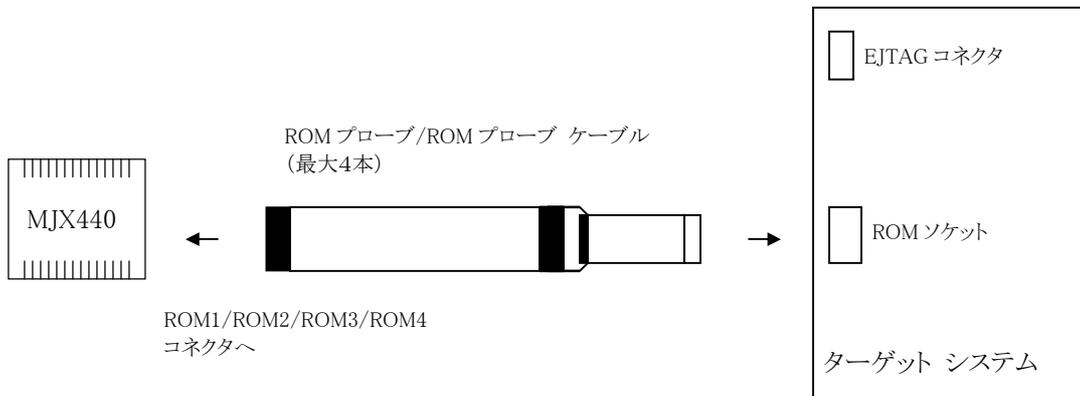


図 3-4 ROM プローブと ROM プローブ ケーブルの接続 2

^{*1} 必要な ROM プローブ本数は使用する ROM のタイプと個数により異なります。

第三章 ハードウェアの接続

ROMプローブの接続方法は、ターゲット システムの次の項目によって異なります。

- ROM1個あたりのデータ バス幅
- ROM の個数
- ROM 全体でのアクセス バス幅

図 3-5-1～3-5-11 の中から、ターゲット システムと一致する接続図をひとつ選び、図にしたがってROMプローブとROMプローブ ケーブルを接続してください。

【注意】 MJX440 およびターゲット システムの電源を切ってから行なってください。

【注意】 ROMプローブを逆差ししないでください。

【注意】 ROMプローブ J-101AとJ-104Aに接続できる二つのROMプラグは、ROMプローブ基板上のシルク印刷文字(JROM1 または JROM2)で区別してください。

バス幅 8 ビット ROM 1 個、ROM アクセス バス幅 8 ビットの場合
(点灯 LED: ROM1)

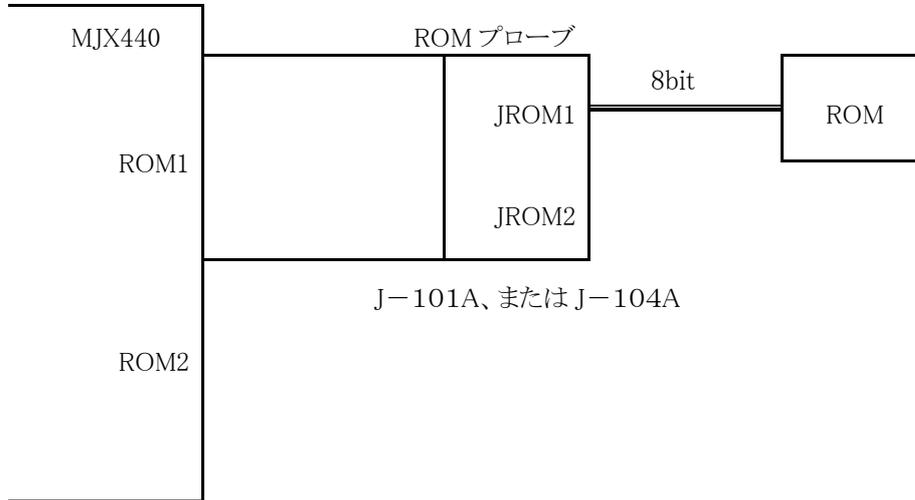


図 3-5-1 ROM プローブの接続 1

バス幅 8 ビット ROM 2 個、ROM アクセス バス幅 8 ビットの場合
(点灯 LED: ROM1)

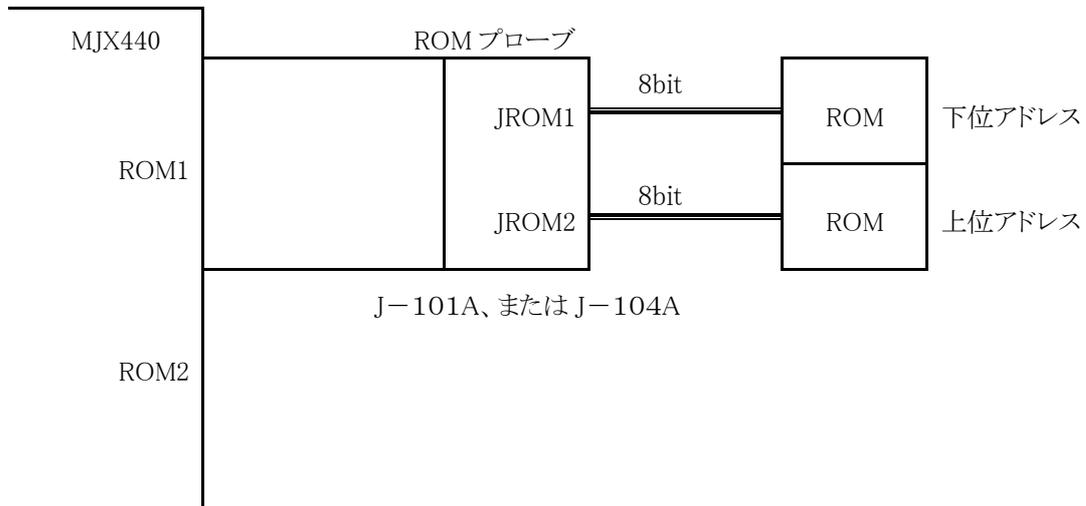


図 3-5-2 ROM プローブの接続 2

バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 8 ビットの場合

(点灯 LED: ROM1、ROM2)

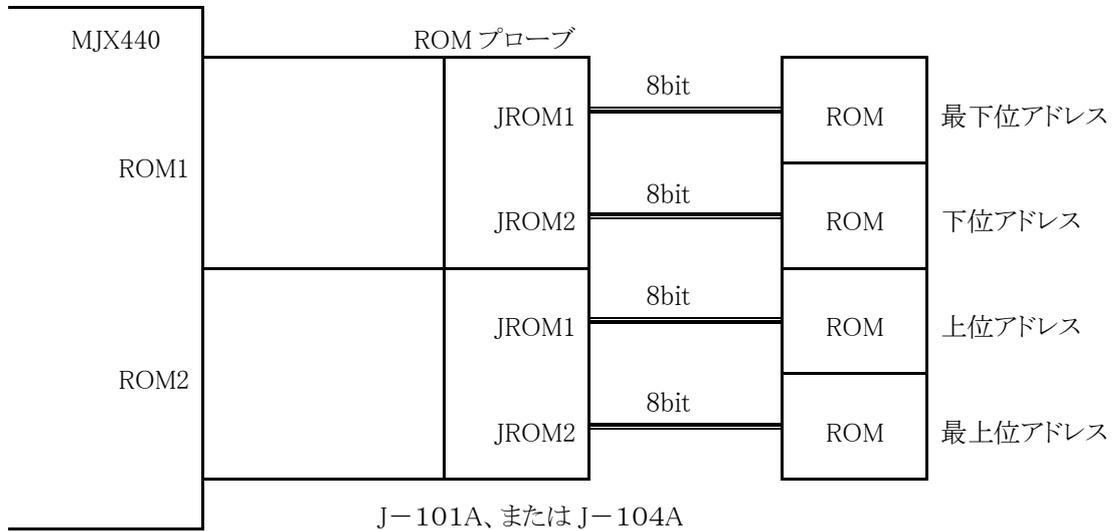


図 3-5-3 ROM プロブの接続 3

バス幅 8 ビット ROM 2 個、ROM アクセス バス幅 16 ビットの場合

(点灯 LED: ROM1)

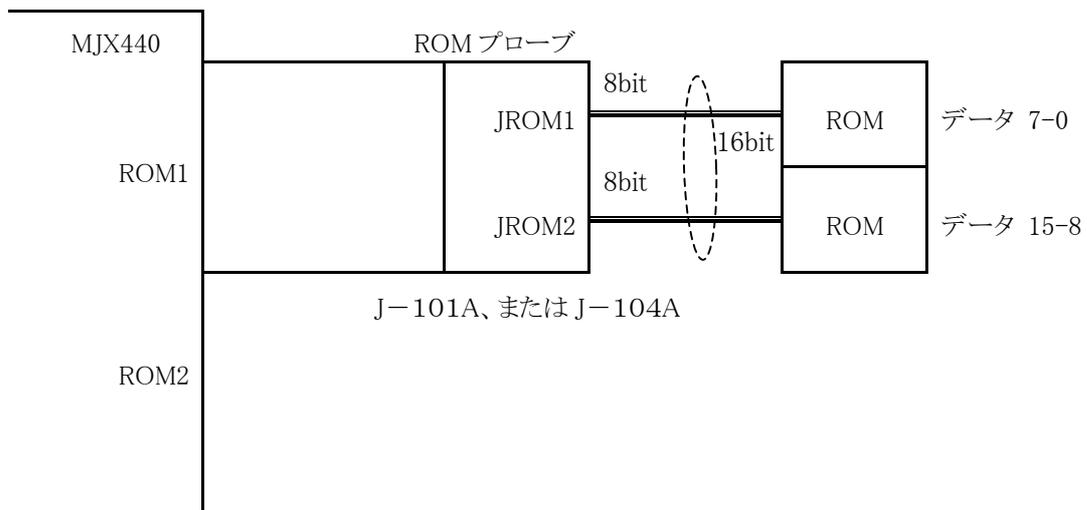


図 3-5-4 ROM プロブの接続 4

バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 16 ビットの場合
 (点灯 LED: ROM1、ROM2)

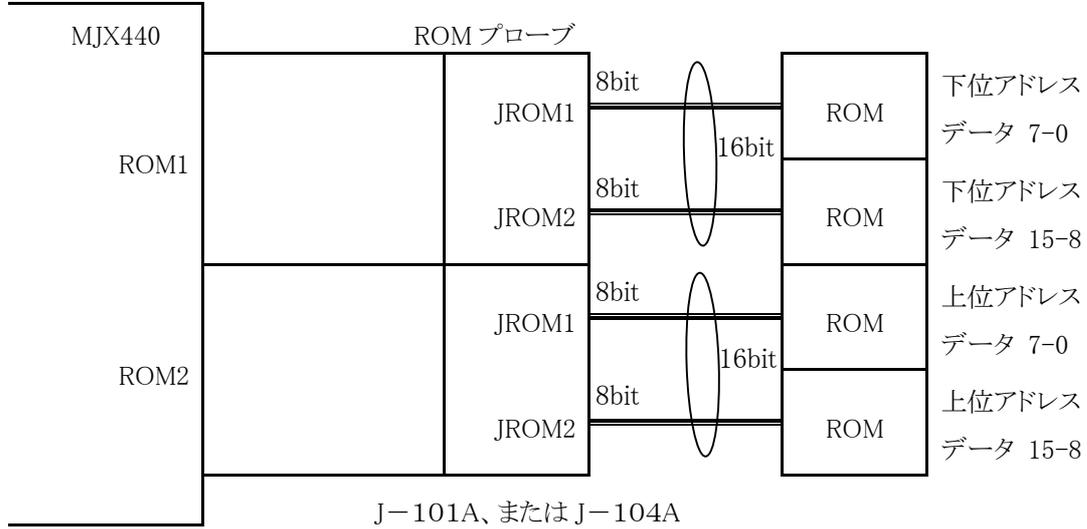


図 3-5-5 ROM プローブの接続 5

バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 32 ビットの場合
 (点灯 LED: ROM1、ROM2)

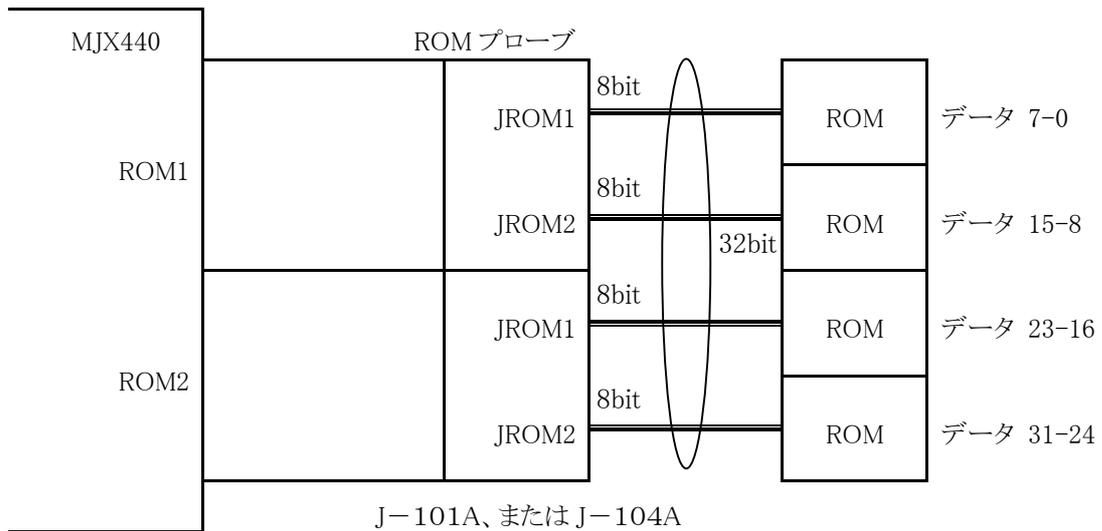


図 3-5-6 ROM プローブの接続 6

バス幅 16ビット ROM 1 個、ROM アクセス バス幅 16ビットの場合
 (点灯 LED: ROM1)

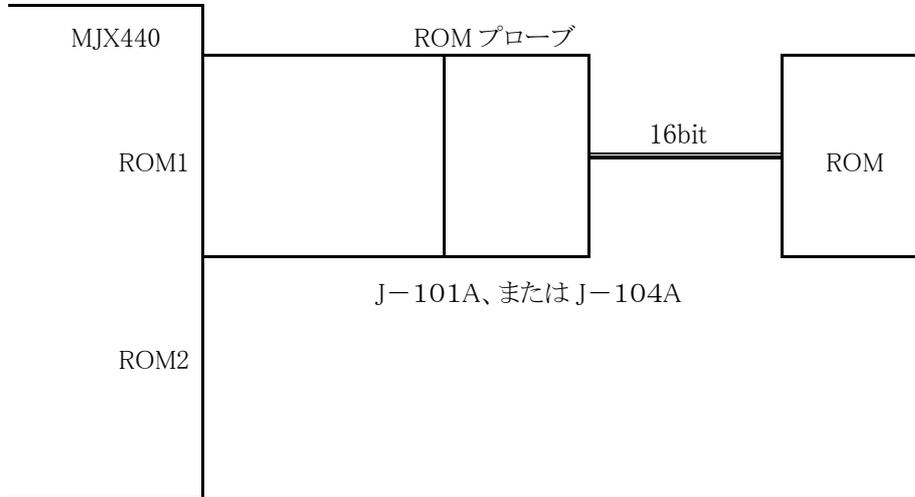


図 3-5-7 ROM プローブの接続 7

バス幅 16ビット ROM 2 個、ROM アクセス バス幅 16ビットの場合
 (点灯 LED: ROM1、ROM2)

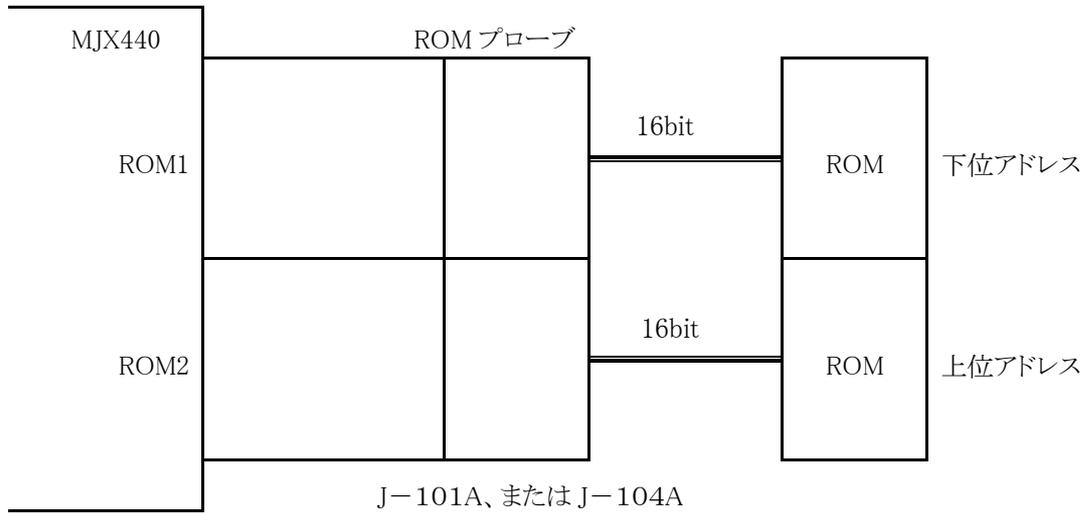


図 3-5-8 ROM プローブの接続 8

バス幅 16ビット ROM 2 個、ROM アクセス バス幅 32ビットの場合
(点灯 LED: ROM1、ROM2)

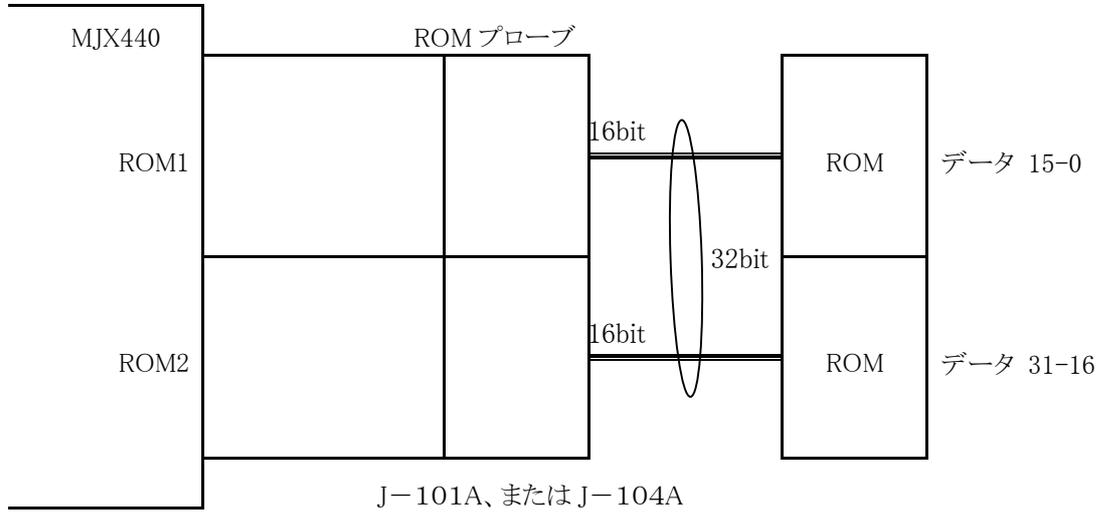


図 3-5-9 ROM プローブの接続 9

バス幅 16ビット ROM 4 個、ROM アクセス バス幅 32ビットの場合
 (点灯 LED: ROM1、ROM2、ROM3、ROM4)

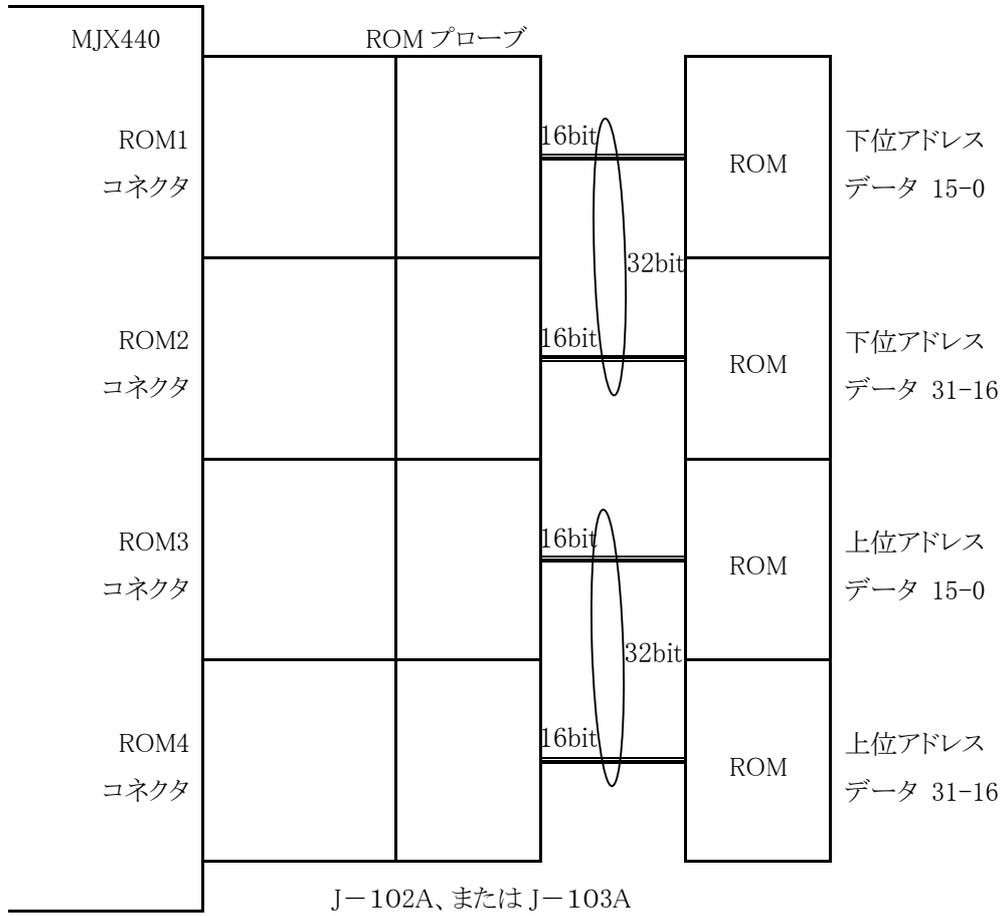


図 3-5-10 ROM プローブの接続 10

第三章 ハードウェアの接続

バス幅 16ビット ROM 4 個、ROM アクセス バス幅 64ビットの場合
 (点灯 LED: ROM1、ROM2、ROM3、ROM4)

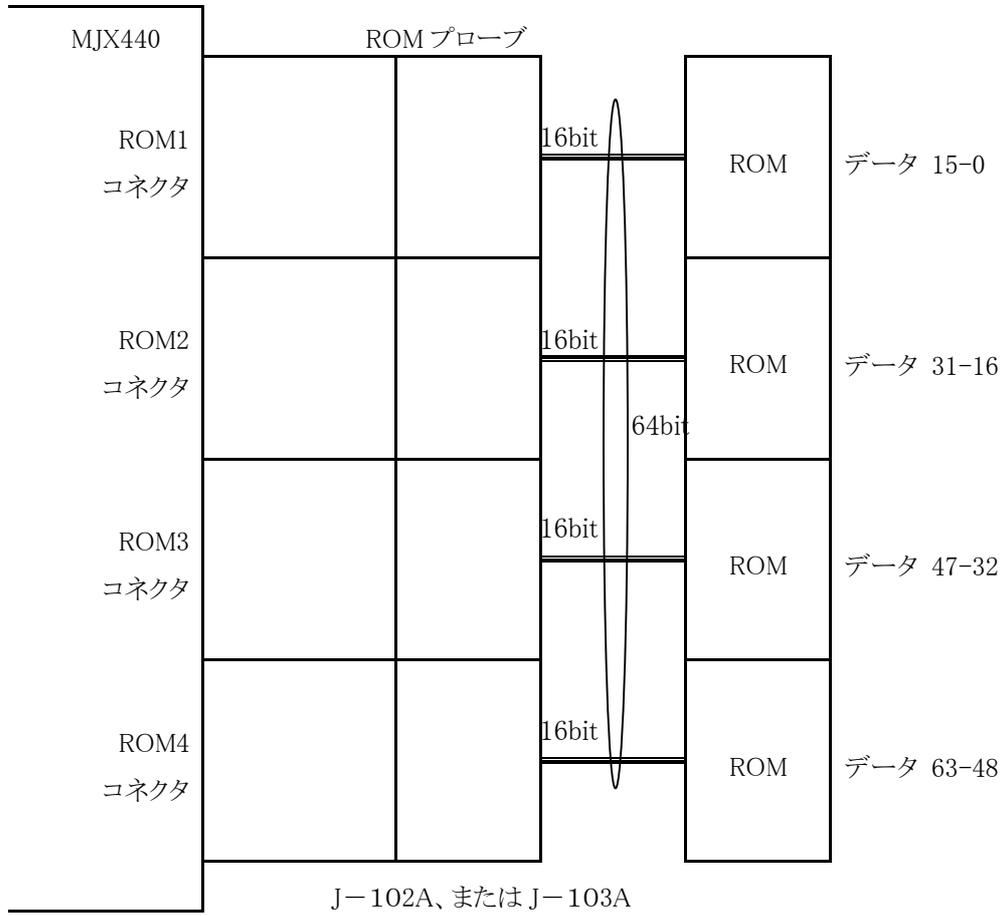


図 3-5-11 ROM プロンプトの接続 11

3.4 外部トリガ ケーブルの接続

【重要】 外部トリガ ケーブル 1 の Vcc(クリップ、ケーブル共に赤)は、必ずターゲット システムの電源(3.3V)へ接続してください。Vcc 以外の信号の接続は任意です。

ターゲット システムの信号の状態をLED に表示させたり、MJX440 が出力するトレース トリガ信号をロジック アナライザの入力として使用する場合は、外部トリガ ケーブルを使用します。これらの機能を使用しない場合は、外部トリガ ケーブルを接続する必要はありません。

外部トリガ ケーブルは、1列タイプの外部トリガ ケーブル1と2列タイプの外部トリガ ケーブル2の二種類があります。外部トリガ ケーブル1は、EJTAG プロブ基板の EXTERNAL コネクタ1へ接続し、外部プロブ コネクタ2は、MJX440 本体の EXTERNAL コネクタ2へ接続します。

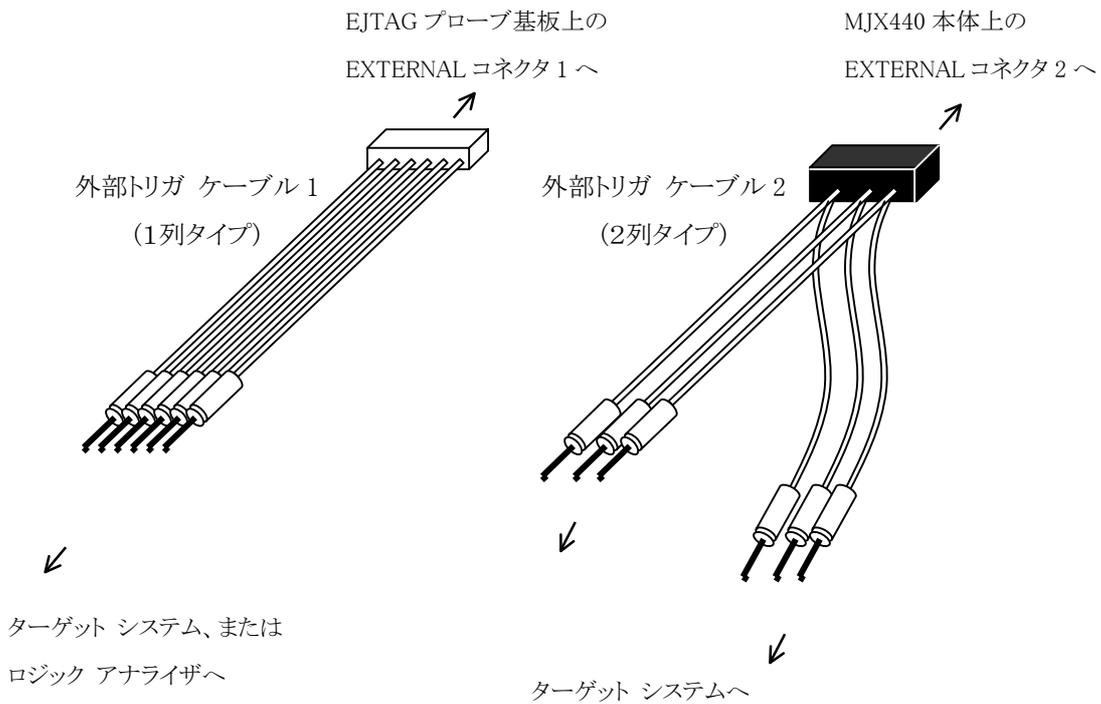


図 3-5 外部トリガ ケーブルの接続

第三章 ハードウェアの接続

外部トリガ ケーブルの信号の詳細は、次のとおりです。信号は、クリップとケーブルの色で区別します。

信号名	色	入出力	機能
EXTIN1	クリップ 黄 ケーブル 茶	TTL 入力	EXTIN1、EXTIN2 および EXTIN3 に接続した信号の状態が、リアルタイム トレース情報として記録されます。
EXTIN2	クリップ 黄 ケーブル 赤	TTL 入力	
EXTIN3	クリップ 黄 ケーブル 橙	TTL 入力	
TRGOUT-	クリップ 緑 ケーブル 緑	3.3V 出力	トレース トリガ通過時に1クロックのLOWパルスを出力します。ロジック アナライザのトリガ入力として使用できます。
Vcc	クリップ赤 ケーブル赤		ターゲット システムの Vcc(3.3V)に接続します。
GND	クリップ 黒 ケーブル 黒		ターゲット システムの GND に接続します。

表 3-1 外部トリガ ケーブル 1 (1 列タイプ、EJTAG プローブ基板に接続)

【重要】 外部トリガ ケーブル 1 の Vcc(クリップ、ケーブル共に赤)は、必ずターゲット システムの電源(3.3V)へ接続してください。Vcc 以外の信号の接続は任意です。

信号名	色	入出力	機能
EXTIN1	クリップ 黄 ケーブル 茶	TTL 入力	接続した信号が HIGH レベルの場合、対応する LED が点灯します。 ^{*1} また、MJXDEBW コマンドの XPIN コマンドで状態を確認することができます。リアルタイム トレースには記録されません。
EXTIN2	クリップ 黄 ケーブル 赤	TTL 入力	
EXTIN3	クリップ 黄 ケーブル 橙	TTL 入力	
EXTOUT1	クリップ 赤 ケーブル 茶	3.3V O.D.出力 ^{*2}	MJXDEBW コマンドの XPIN コマンドで、LOW または HIGH レベルを出力します。
EXTOUT2	クリップ 赤 ケーブル 赤	3.3V O.D.出力 ^{*2}	
GND	クリップ 黒 ケーブル 黒		ターゲット システムの GND に接続します。

表 3-2 外部トリガ ケーブル 2 (2 列タイプ、MJX440 本体に接続)

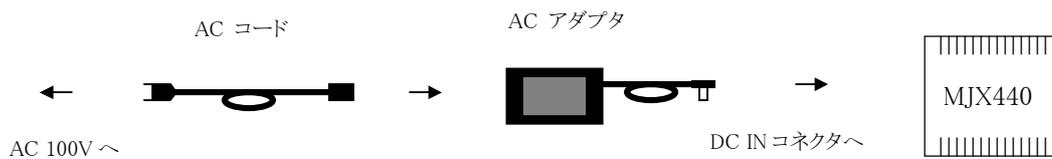
^{*1} MJX440 本体上の信号名は、EXTIN3→EXI 3 のように省略して表記されています。

^{*2} O.D.は、オープン ドレインの略です。

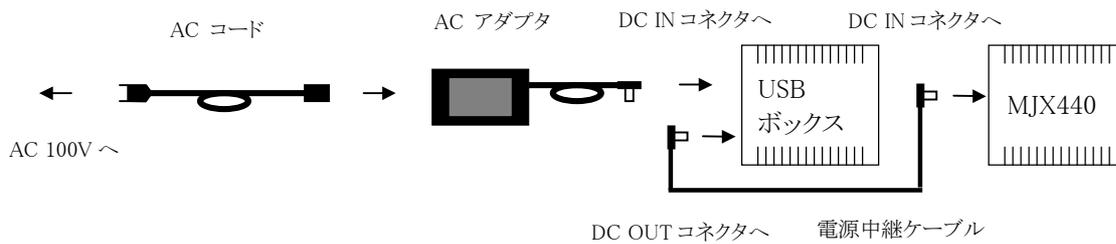
3.5 電源の接続と電源投入手順

すべてのケーブルの接続が終了したら、AC アダプタと AC コードを接続してください。このとき、MJX440 の電源スイッチが OFF の状態になっていることを確認してください。

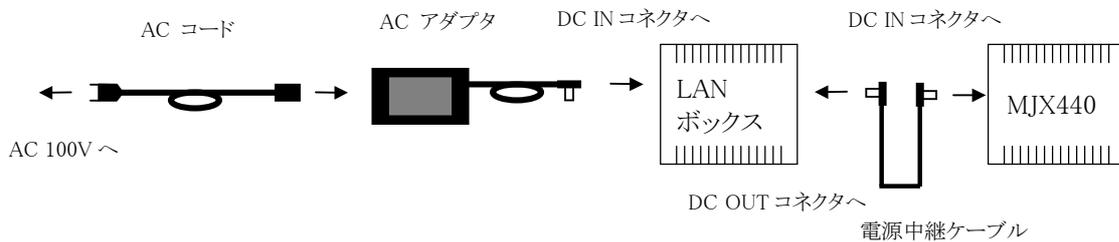
パラレル接続(PCI,PCMCIA カード)の場合



USB 接続の場合*1



LAN 接続の場合



*1 USB I/F は開発中です。

機器の電源投入は、次の手順で行なってください。

1. ホスト PC
2. USB ボックス*1/LAN ボックス(USB 接続/LAN 接続の場合)
3. MJX440
4. ターゲット システム

機器の電源切断は、次の手順で行なってください。

1. ターゲット システム
2. MJX440
3. USB ボックス*1 /LAN ボックス
4. ホスト PC

【重要】 電源投入および切断の手順を間違えると、機器が破壊される場合があります。

【重要】 電源投入時には、機器の接続および着脱をしないでください。

*1 USB I/F は開発中です。

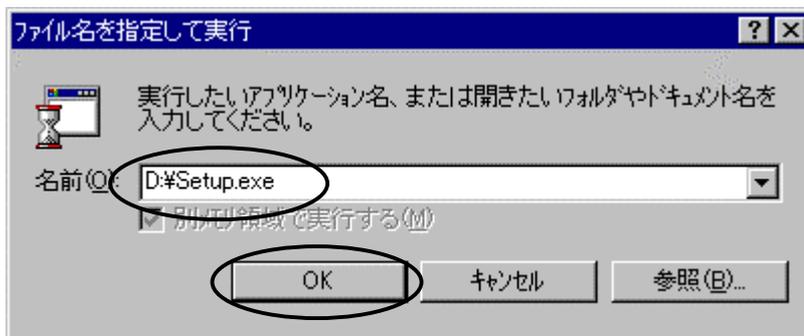
第四章 ソフトウェアのインストール

MJX440 を操作するソフトウェアのインストール方法について記述しています。

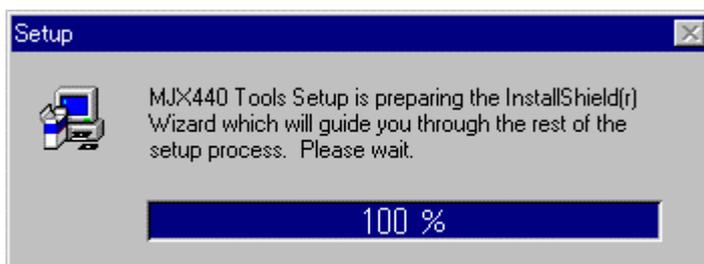
インストールは、以下の手順にしたがって行ってください。

4.1 Windows の場合

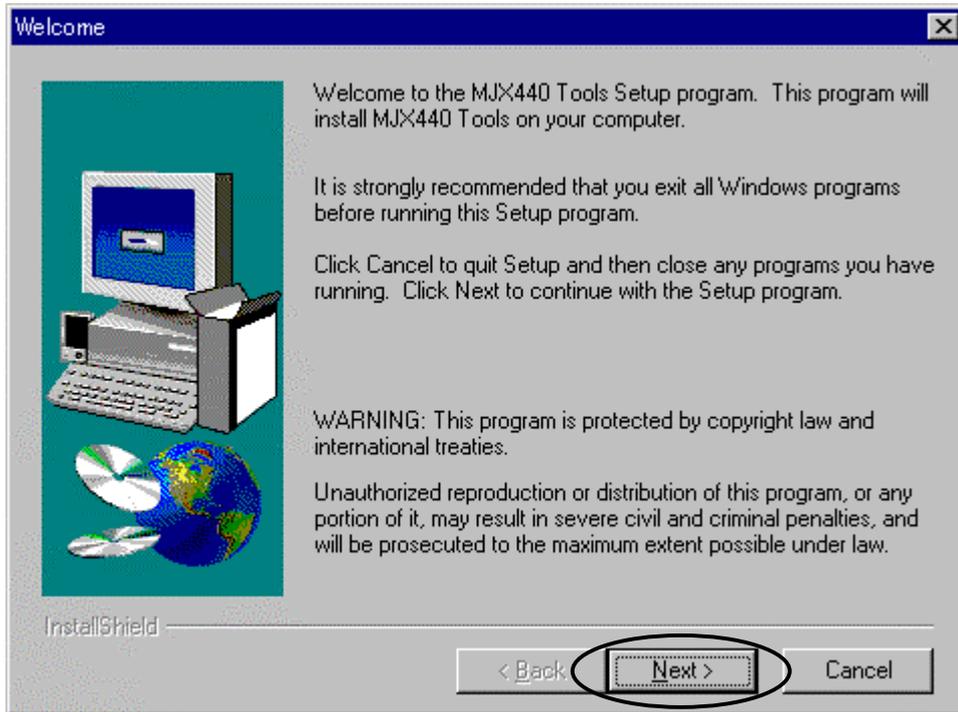
1. Green Hills Software 社の統合開発環境 MULTI をインストールします。
2. CD-ROMドライブに「MJX for TR4102/CW4020 Tools Disk」の CD-ROM をセットします。
3. 「スタート」、「ファイル名を指定して実行(R)…」の順に選択し、「ファイル名を指定して実行」ダイアログを表示します。
4. 「名前(O)」にセットアップ プログラム Setup.exe を指定し、「OK」をクリックします。(CD-ROMドライブが Dドライブの場合、「D:¥Setup.exe」)



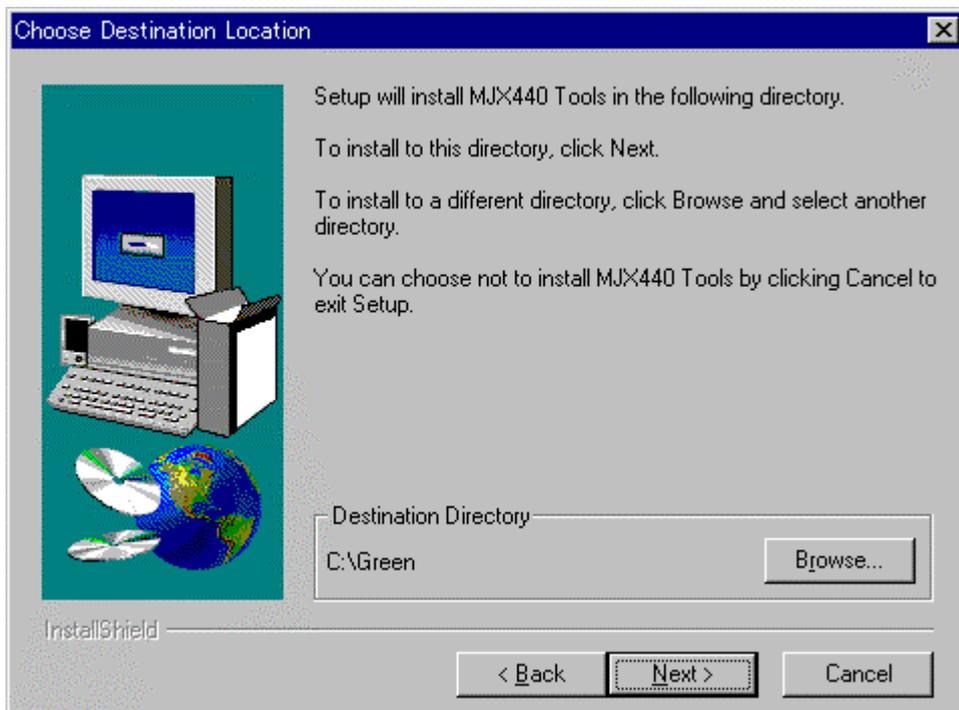
5. セットアップ プログラムが起動します。



6. 「Welcome」ダイアログが表示されるので、「Next >」をクリックします。

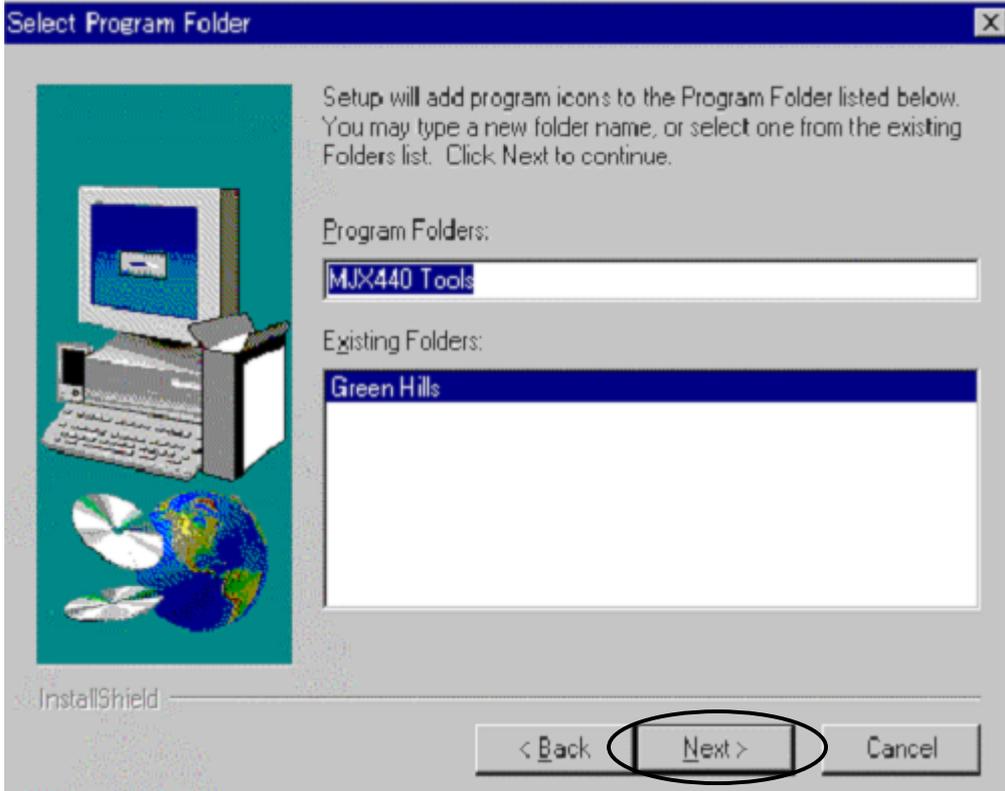


7. 「Choose Destination Location」ダイアログが表示されるので、「Browse...」ボタンをクリックしてインストール先を指定します。先にインストールしたMULTIと同じディレクトリを、インストール先に指定してください。(デフォルトは「C:\Green」です)



- 「Next >」をクリックします。

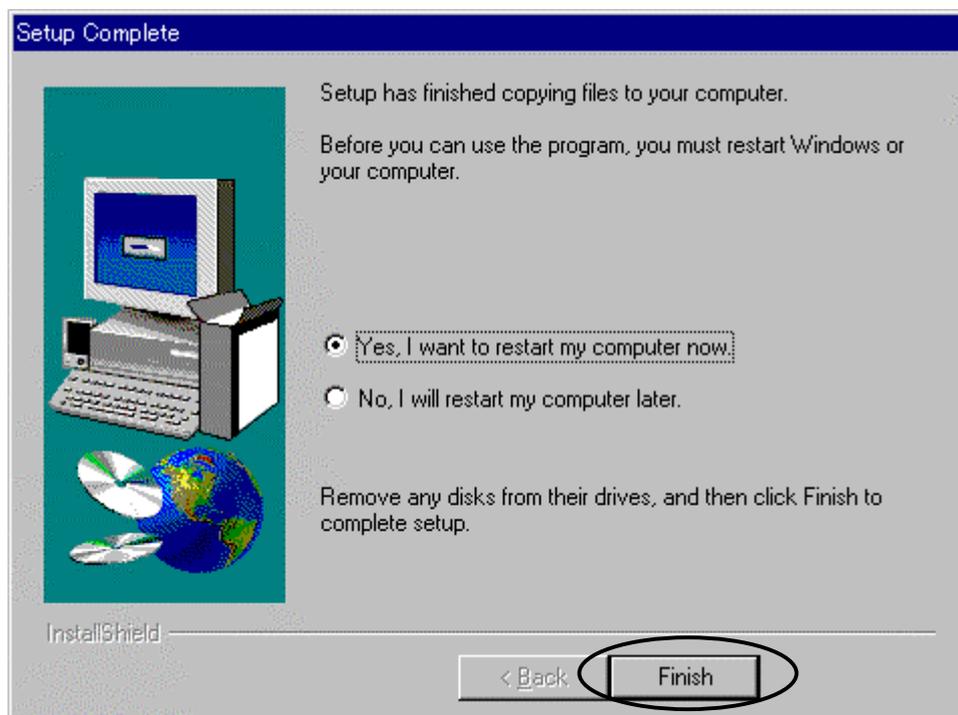
「Select Program Folder」ダイアログが表示されるので、「Next >」ボタンをクリックします。



8. インストールが開始します。
9. インストールが終了すると、作成されたプログラムフォルダが表示されます。



10. 「Setup Complete」ダイアログが表示されるので、「Finish」ボタンをクリックし、システムを再起動します。



【重要】 インストーラはシステムディレクトリ*1にライブラリMSVCRT.DLLをインストールします。古いライブラリはMSVCRTxx.DLL (xxはバージョン番号)のようにリネームされます。

*1 システムディレクトリのデフォルトは、Windows9xの場合はC:\Windows\System、WindowsNT4.0およびWindows2000では、C:\Winnt\System32です。

インストールされるファイルは次のとおりです。

MJXDEBW.EXE	簡易デバッガ プログラム MJXDEBW
MJXCFG.EXE	コンフィグ支援ツール MJXCFG (コンフィグレーション ファイル作成用)
MJXSERV.EXE	MJX 用 MULTI サーバー プログラム MJXSERV
MJXCVT.EXE	MJX バイナリ ファイル変換プログラム MJXCVT
WNTIX.EXE	インストール用プログラム (使用しない)
MJX4020.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (CW4020)
MJX4102.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (TR4102)
MJXARM.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (ARM9)
MJXNB85E.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (NB85E)
MJXV831.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (V831)
MJXV832.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (V832)
Mjx4kc.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (MIPS4kc)
MjxVR55.INI	MJXDEBW 用コンフィグレーション ファイルのサンプル (VR5500)
MJXDBAPI.DLL	MJXSERV-MJXDEBW 通信ライブラリ
MJXASM.DLL	アセンブラ/逆アセンブラ ライブラリ
MJXARM.DLL	ARM ライブラリ
MJXEJTAG.DLL	TR4102/CW4020 ライブラリ
MJXNB85E.DLL	NB85E ライブラリ
MJXV831.DLL	V831 ライブラリ
MJXV832.DLL	V832 ライブラリ
MJ2Ej2R6. DLL	EJTAG2.5/2.6 ライブラリ
Mj2VR55.DLL	V5500 ライブラリ
MSVCP60.DLL	Microsoft Visual C++ ランタイム ライブラリ
MSVCRT.DLL	Microsoft Visual C++ ランタイム ライブラリ
CfgResource.DLL	コンフィグリソース ライブラリ
MjxARMJ.PDF	ARM 和文ヘルプのファイル
MjxEJtgJ.PDF	CW4020/TR4102 和文ヘルプのファイル
MjxEJtgE.PDF	CW4020/TR4102 英文ヘルプのファイル
MjxNB85J.PDF	NB85E 和文ヘルプのファイル
MjxCfgJ.PDF	コンフィグレーションヘルプ画面のファイル
MjxifJ.PDF	ホストインターフェース 和文ヘルプのファイル
MjxifE.PDF	ホストインターフェース 英文ヘルプのファイル
MjxV83XJ.PDF	V831/V832 和文ヘルプのファイル
MjxV83XE.PDF	V831/V832 英文ヘルプのファイル

第四章 ソフトウェアのインストール

MjX440.ICO	MjX440 アイコン ファイル
MjxCARMJ.PDF	MjX330 for ARM 和文ヘルプのファイル
Mj2CARMJ.PDF	MjX330 for ARM (汎用品) 和文ヘルプのファイル
Mj2CEj26J.PDF	MjX330 for EJTAG2.5/2.6 和文ヘルプのファイル
MjxCNB85J.PDF	MjX330 for NB85E 和文ヘルプのファイル
Mj2CVR5J.PDF	MjX330 for VR5500 和文ヘルプのファイル

ソフトウェアのファイル構成は次のとおりです。

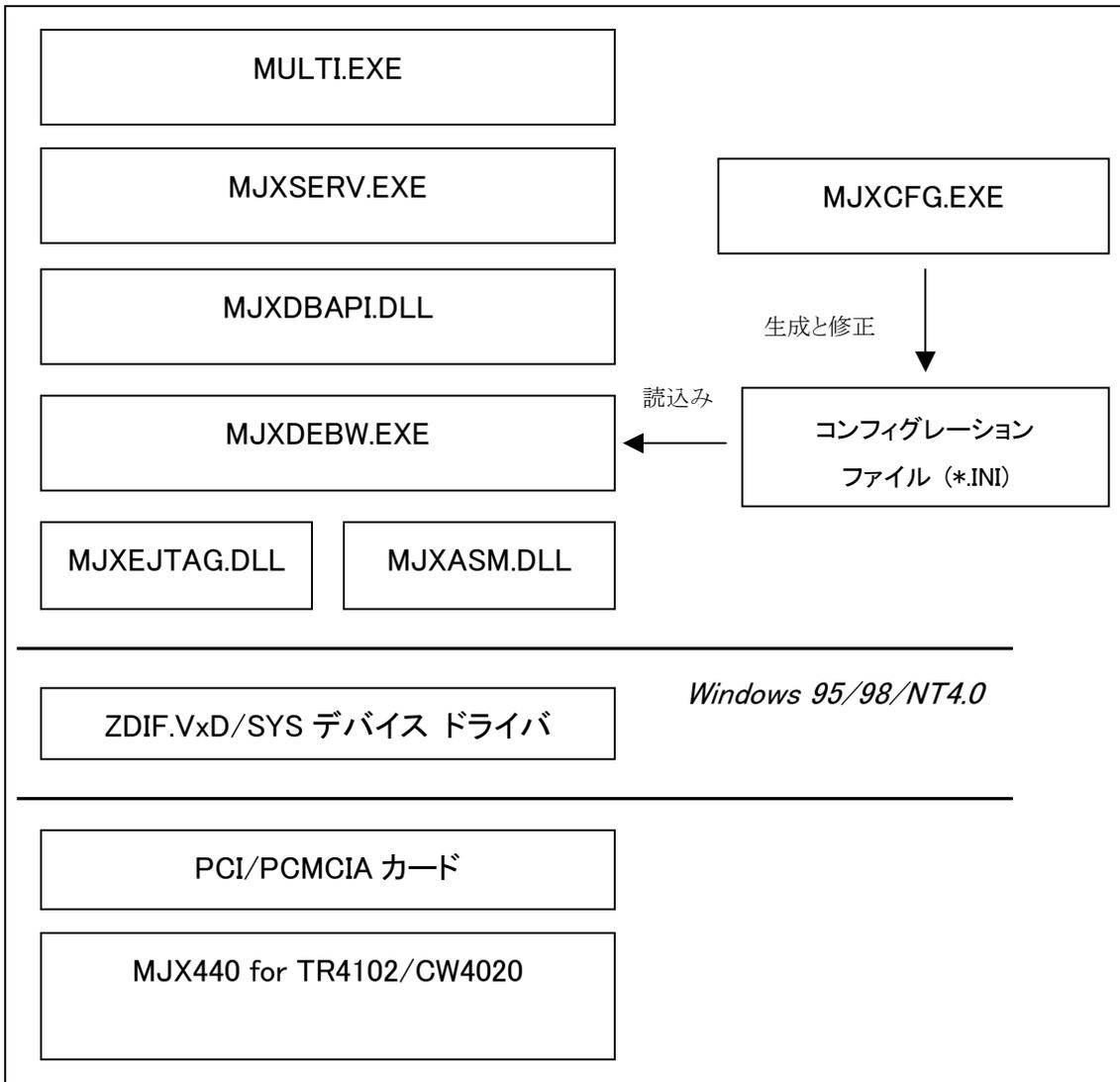


図 4-1 ソフトウェアのファイル構成

- MULTI.EXE は MJXSERV.EXE を子プロセスとして呼び出します。
- MJXSERV.EXE は MJXDBAPI.DLL を経由して MJXDEBW.EXE を子プロセスとして呼び出します。
- MJXDEBW.EXE は MDXSERV.EXE の子プロセスとしてだけでなく、単体でも動作できます。

4.2 Solaris の場合

1. Green Hills Software 社の統合開発環境 MULTI をインストールします。
2. CD-ROMドライブに「MJX for MIPS/EJTEG Tools (for Solaris)」の CD-ROM をセットします。
CD-ROM をマウントします。(通常はオート・マウントされるはずです。)
3. 端末エミュレータ画面を開き、以下のコマンドを入力します。

```
cd {CD-ROM がマウントされたディレクトリ} # {...} は適切なディレクトリ名に置換  
sh setup {MULTI がインストールされたディレクトリ} # {...} は適切なディレクトリ名に置換
```

4. 「MJXSERV setup complete」のメッセージが出力されて、MJXSERV のインストールが終了します。

例: CD-ROM がマウントされたディレクトリを `/cdrom/mjx`、MULTI がインストールされたディレクトリを `/export/home/green/multius_all` としたとき、以下のコマンドを入力します。

```
# cd /cdrom/mjx  
# setup /export/home/green/multius_all  
MJXSERV setup complete
```

第四章 ソフトウェアのインストール

インストールされるファイルは次のとおりです。

mjxserv	MJX440 用 MULTI サーバー プログラム MJXSERV
mjx4020.ini	MJXDEBW 用コンフィグレーション ファイルの サンプル (CW4020)
mjx4102.ini	MJXDEBW 用コンフィグレーション ファイルの サンプル (TR4102)
MjxEJtgJ.pdf	CW4020/TR4102 和文ヘルプのファイル
./zax_mjx/flash/pgm/mips0.b_e	JEDEC スタンダード コンパティブル フラッシュ メモリ 書き込みプログラム (ビッグ エンディアン)
./zax_mjx/flash/pgm/mips0.l_e	JEDEC スタンダード コンパティブル フラッシュ メモリ 書き込みプログラム (リトル エンディアン)
./zax_mjx/flash/device/{デバイス名}.dat	フラッシュ メモリ デバイス情報定義ファイル ファイル名については「 付録 J フラッシュ メモリ デバ イス情報定義ファイル レイアウト 」を参照してください

プロセスとハードウェアの構成は次のとおりです。

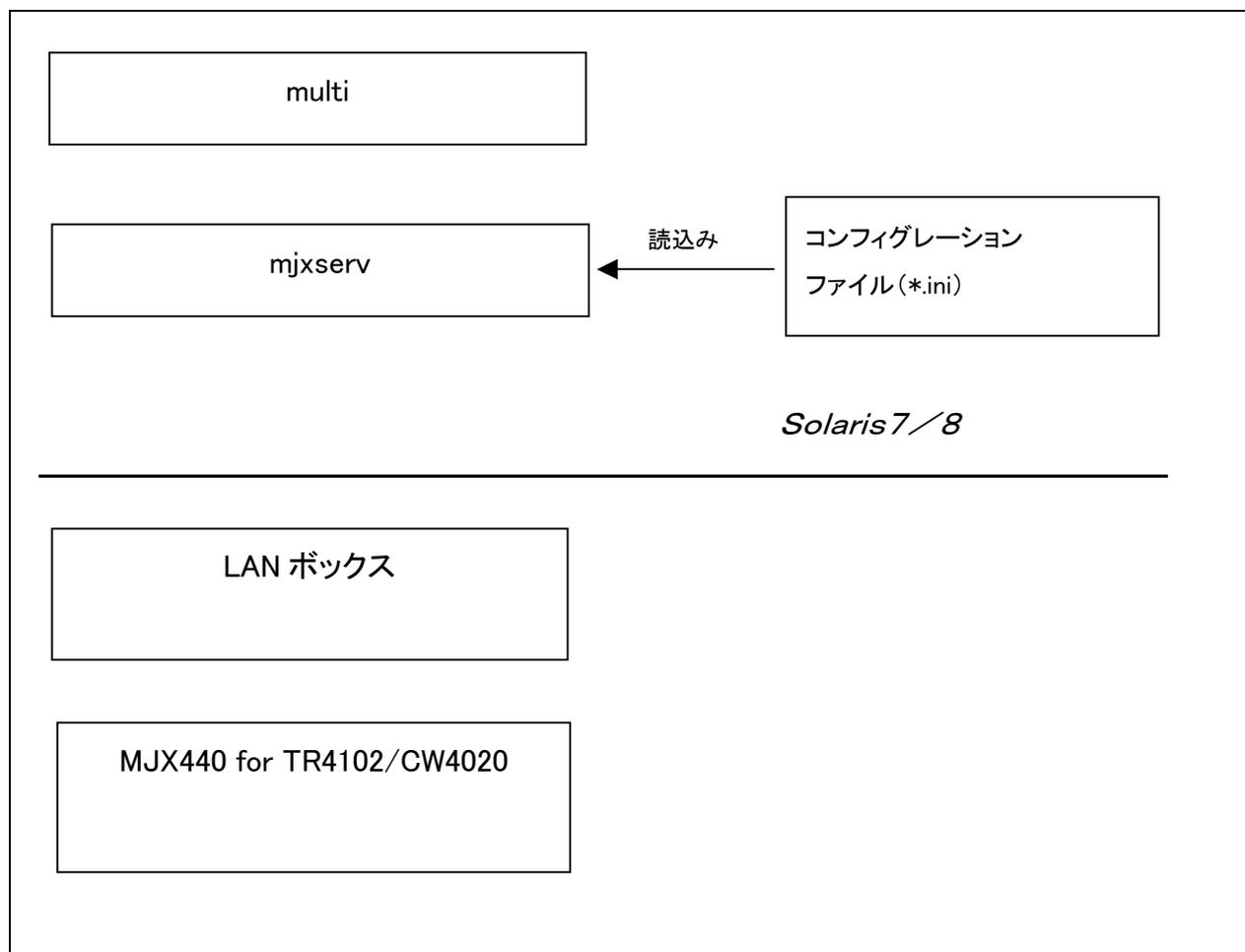


図 4-2 プロセスとハードウェアの構成/Solaris

- multi は mjxserv を子プロセスとして呼び出します。

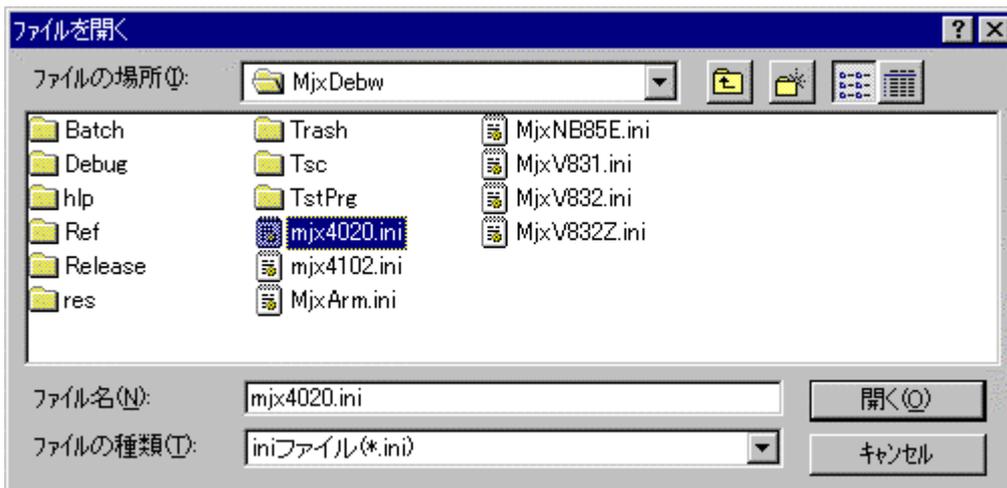
第五章 MJX440 の環境設定

MJX440 を使用する前に必要な、環境設定の方法について記述しています。

5.1 Windows の場合

MJX440 の環境設定は、コンフィグ支援ツール MJXCFG で行ないます。すべての機器の電源を投入した後、スタートメニューから MJXCFG を起動してください。手順は、次のとおりです。

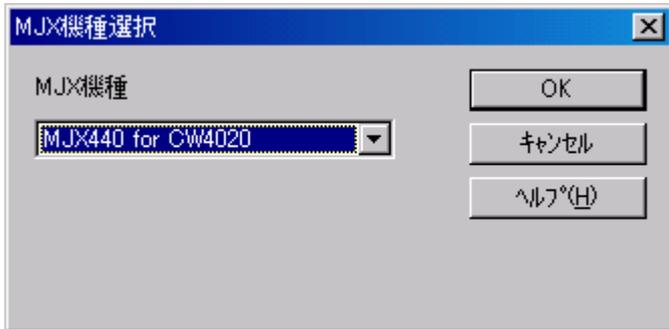
1. スタートメニュー
2. プログラム(P)
3. MJX Tools
4. MjxCfg Config Tool



MJX4020.INI、MJX4102.INI をテンプレートとして利用できます。

新たにコンフィグレーションファイルを作成する場合は、新しいファイル名を指定します。

新規のコンフィグレーション ファイルを指定した場合は、次のダイアログ ボックスが表示されます。



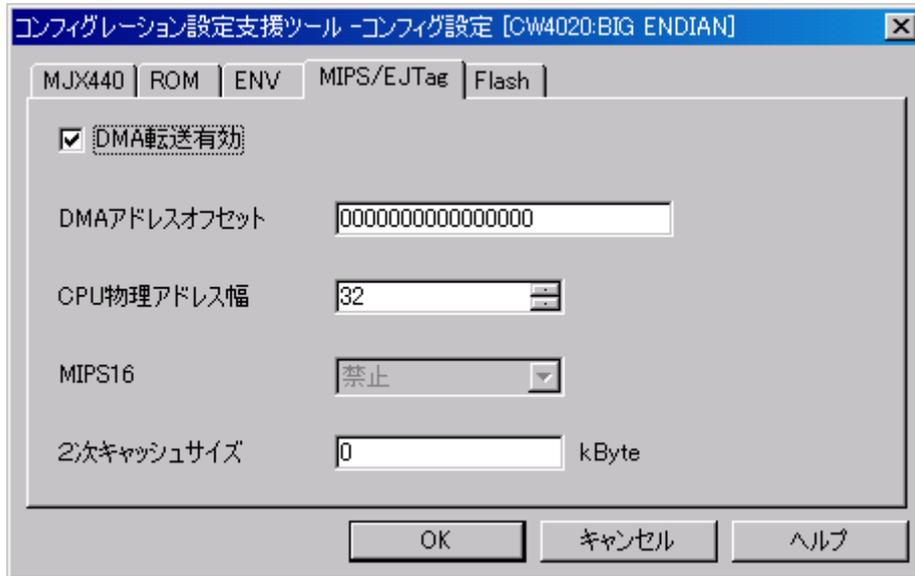
MJX の機種から「MJX440for CW4020」を選択し、**OK** ボタンを押してください。



ターゲットCPU*1とエンディアンを選択し、**OK** ボタンを押してください。

*1 ターゲットCPUはTR4102またはCW4020を指定してください。

次のダイアログ ボックスが表示されます。



ダイアログ ボックスが表示されたら、MJX440 やターゲット システムの環境に合わせてフィールドを設定し、**OK** ボタンを押してください。

コンフィグレーション設定項目の詳細は「MJX440/MJX330 MjxCfg User's Manual (Mjxcfg.PDF)」をご覧ください。

5.2 Solaris の場合

MJX440 の環境設定をするには、コンフィグレーション ファイルを適当なエディタ(例えば vi)を使用して作成します。その際、テンプレートとして、mjx4020.ini(または mjx4102.ini)を利用できます。

【注意】 コンフィグレーション設定は慎重に行ってください。コンフィグレーション設定が間違っていると MJXSERV が正常に動作しない場合があります。

コンフィグレーション ファイルは Windows の INI ファイルと同様次のような構造を持ちます。

```
[セクション]
キー=値
.....
```

以下に、コンフィグレーション設定項目の詳細を示します。キーの説明中、「=」の右辺は、設定値の一例です。

1. [TARGET] セクション

MIPS/EJTAG に関する固有の設定を行います。

ターゲット CPU

```
キー   : CPUTYPE=4
値     : 3:4102
        4:4020/4030
```

エンディアン

```
キー   : ENDIAN=0
値     : 0:BIG ENDIAN
        1:LITTLE ENDIAN
```

DMA 転送有効 の指定

キー : EJTAGDMATRNSENA=1

値 : 0: 無効

1: 有効

- 補足 :
- ターゲットボード上のメモリアクセスに、CPU の DMA 機能を使うかどうかを指定します。
 - DMA 転送を有効にすると高速にメモリアクセスできるため、オブジェクトファイルのダウンロードも高速になります。
 - DMA 転送はメモリの物理アドレスに直接書込みを行います。従って仮想記憶オペレーティングシステムを使用していて論理アドレスと物理アドレスのマッピングが不明な場合や論理アドレスが不連続な物理アドレスにマッピングされている場合、DMA 転送は使用できません。

DMA アドレスオフセット

キー : EJTAGDMAOFFSET=0000000000000000

値 : 0000000000000000

- 補足 :
- DMA 転送は物理アドレスに直接書込みを行います。
 - 論理アドレスと物理アドレスにオフセットがある場合指定します。

CPU 物理アドレス幅

キー : MIPS PHYADDR=32

値 : 32

- 補足 :
- CPU の物理アドレス幅を 32~64bit の範囲で指定します。
 - MJX440 for TR4102/CW4020 では 32 または 64 以外の設定は禁止です。

MIPS16

キー : MIPS16ENA=0

値 : 0

- 補足 :
- MIPS16 命令を使用するかどうかを指定します。
 - 現在のバージョンでは MIPS16 命令は使用できません。

2次キャッシュサイズ

- キー : MIPS2NDCACHE=0
値 : 0
補足 : ● 2次キャッシュのサイズを kByte 単位で指定します。
● 2次キャッシュが無い場合は 0 を指定してください。

以下は未使用キーです

- キー : EPCAVAIL=0
キー : NB85ETRCCLK=0
キー : NB85EMEMMODE=0
キー : ARMETM=0
キー : ARMTRACEPORTSIZE=0
キー : ARMTRACECLOCKEDGE=0
キー : ARMCACHESIZE=0
キー : ARMTDOSAMPLING=0
キー : ARMMMDCTRL=00
キー : ARMDBGACK=0
キー : ARMPORTSEL=0
キー : ARMPORTMODE=0

2. [MJX440] セクション

MJX440 本体に関わる設定を行います。

コンフィグ・バージョン

キー : VERSION=120
値 : 120
補足 : ● 変更不可

JTAG クロック

キー : JCLOCK=1
値 : 0: 40MHz
1: 20MHz
補足 : ● MJX440 を起動するときの JTAG クロック周波数を指定します。
● 周波数が高いほど、高速にターゲットボードにアクセスできます。
● 指定した周波数でターゲットボードに正しくアクセスできない場合は低い周波数を指定してください。

MJX440 に実装されているエミュレーションメモリの容量

キー : EMMVOL=3
値 : 0: 0 Mbyte
1: 4 Mbyte
2: 8 Mbyte
3: 16 Mbyte

ホストインターフェースの種別

キー : IFTYPE=3
値 : 0: PCI
1: PCMCIA
2: USB*1
3: LAN
補足 : ●

以下は未使用キーです

キー : DBINT_EDGE=0

*1 USB I/F は開発中です。

3. [ROMEMU] セクション

エミュレーションメモリに関する設定を行います。

ROM 先頭アドレス (16 進数)

キー : TOPADDR=000000001FC00000

値 : アドレス

- 補足 :
- ターゲット システムの ROM 領域の先頭アドレスを論理アドレス(16 進数)で指定します。
 - 複数の ROM を実装しているターゲット システムの場合は、一番下位アドレスにマッピングされている ROM の先頭アドレスを指定してください。
(例) 0x00100000 番地から始まる ROM と、0x00180000 番地から始まる ROM が実装されているターゲット システムの場合は、0x00100000 を指定してください。
 - 頭の「0x」は記入しません。

ROM イメージ領域の先頭アドレス (16 進数)

キー : ROMIMAGE=00000000FFFFFFFF

値 : アドレス

- 補足 :
- アドレスをフルデコードしていないターゲット システムや異なる論理アドレスが同一の物理アドレスを CPU では、ROM 領域と同じようにアクセスできる ROM イメージ領域が存在する場合があります。この領域をエミュレーション メモリ領域に割り当てるために ROM イメージを有効に設定したいとき、ROM イメージ領域の先頭アドレスを論理アドレスを 16 進数で指定します。
 - 頭の「0x」は記入しません。

ROM タイプ

キー : ROMTYPE=0

- 値 :
- 0: 64Kx16Bit(1MBit)
 - 1: 128Kx8Bit(1MBit)
 - 2: 128Kx16Bit(2MBit)
 - 3: 256Kx8Bit(2MBit)
 - 4: 256Kx16Bit(4MBit)
 - 5: 512Kx8Bit(4MBit)
 - 6: 512Kx16Bit(8MBit)
 - 7: 1Mx8Bit(8MBit)
 - 8: 1Mx16Bit(16MBit)

- 9: 2Mx8Bit(16MBit)
- 10: 2Mx16Bit(32MBit)
- 11: 4Mx8Bit(32MBit)
- 12: 4Mx16Bit(64MBit)

補足 : ● ターゲット システムで使用している ROM の種別を指定します。

使用 ROM 個数

キー : ROMCNT=0

- 値 :
- 0: 0
 - 1: 1
 - 2: 2
 - 3: 4
 - 4: 8

補足 : ● エミュレーションする ROM 個数を指定します。

ROM アクセスバス幅

キー : BUSWIDTH=2

- 値 :
- 0: 32 bit
 - 1: 16 bit
 - 2: 8 bit
 - 3: 64 bit

補足 : ● ターゲット システムの ROM が読み込まれるときのデータ バス幅を指定します。
● CPU のデータ バス幅とは異なる場合がありますので注意してください。

4. [ENVIRONMENT] セクション

ホストコンピュータに関する設定を行います。

転送バッファサイズ

キー : TRNSSIZE=4

値 : N: N x 64 Kbyte (N = 1 ~ 7)

- 補足 :
- MJXSERV が使用する転送バッファの大きさを 64 Kbyte 単位で指定します。
 - デバッガ プログラムには階層があるため、実際には 4 倍程度のメモリが消費されます。
 - あまり大きな数字を指定するとページングが発生してかえって低速になります。この数字を大きくしても劇的な変化は期待できません。

5. [PATCH] セクション

汎用レジスタ名を表示する形式の指定

キー : GNUREGNAME=0

値 : 0: 通常のレジスタ名 (大文字)

1: 通常のレジスタ名 (小文字)

2: GNU 形式のレジスタ名 (大文字)

3: GNU 形式のレジスタ名 (小文字)

- 補足 : ● 通常のレジスタ名は[付録 G レジスタ名一覧](#)を参照してください。

第六章 ソフトウェアの起動と終了

MJX440 を操作するソフトウェアの起動方法と終了方法について記述しています。

【注意】 ソフトウェアを起動する前には必ず「第五章 MJX440 の環境設定」を行ってください。

【注意】 MJX440 を操作するソフトウェアは、同時にひとつしか実行できません。

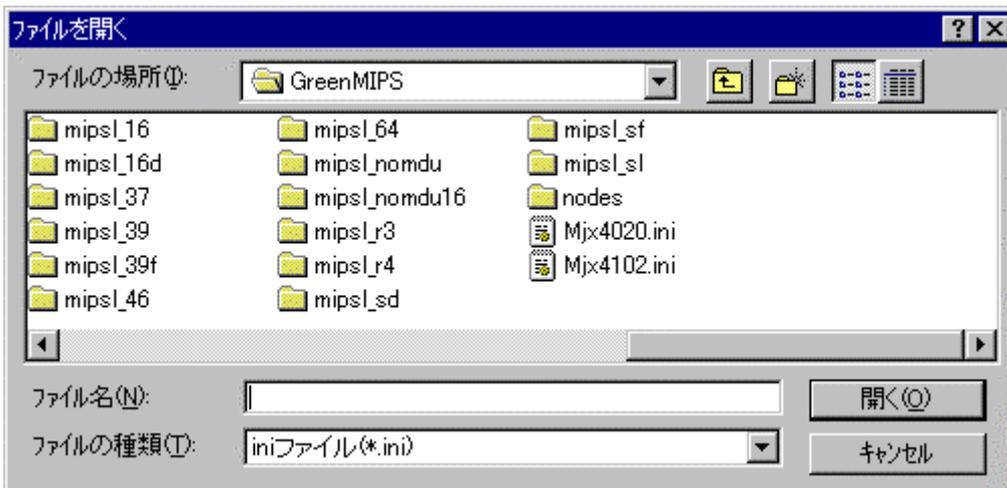
6.1 Windows の場合

MULTI を使用する場合

[MULTI](#) を使って MJX440 を操作するためには、MULTI 本体を起動した後、次のコマンドで MJXSERV をリモート接続してください。^{*1}

```
remote mjxserv
```

上記コマンドを入力すると、コンフィグレーション ファイルの入力をするための、ダイアログ ボックスが開きます。ここで、「[第五章 MJX440の環境設定](#)」で作成した ターゲット システム用のコンフィグレーション ファイルを指定してください。^{*2}

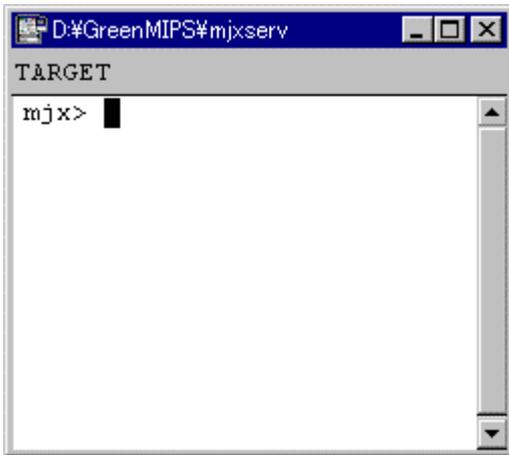


正常にリモート接続できた場合は、MULTI のターゲット ウィンドウと MJXDEBW のウィンドウが開きます。

^{*1} ビルダから操作する場合は、サーバ名を `mjxserv` と指定し、**リモート** ボタンを押します。

^{*2} コンフィグレーション ファイルを指定するダイアログ ボックスで、ファイル名の指定に十数秒以上の時間がかかると、MULTI は「Server Message Timed Out, Terminate Connection?」というメッセージを表示します。このメッセージが表示された場合は、必ず「いいえ」のボタンを押してください。

MULTI のターゲット ウィンドウ



MJXDEBW のウィンドウ



コンフィグレーション ファイルは、MJXSERV の引数として指定することもできます。次のコマンドは、コンフィグレーション ファイルとして *Mjx4020.ini* を指定します。

```
remote mjxserv Mjx4020.ini
```

第六章 ソフトウェアの起動と終了

LAN 接続で LAN ボックスの IP アドレスを明示的に指定したい場合、*-h* オプションを付加します。
次のコマンドは、IP アドレスとして *192.168.0.55* を指定します。

```
remote mjxserv -h192.168.0.55
```

終了するには、*quit* コマンドを入力してください。

```
quit
```

MJXDEBW を使用する場合

MJXDEBWを使ってMJX440 を操作するためには、スタート メニューから MJXDEBW を起動してください。
手順は、次のとおりです。

1. スタート メニュー
2. プログラム(P)
3. MJX440 Tools
4. MJXDEBW Debugger

MJXDEBW が起動されると、コンフィグレーション ファイルの入力をするための、ダイアログ ボックスが開きます。ここで、ターゲット システム用のコンフィグレーション ファイルを指定してください。



MJXDEBW が正常に起動できた場合は、MJXDEBW のウィンドウが開きます。



第六章 ソフトウェアの起動と終了

コンフィグレーション ファイルは、MJXDEBW の引数として指定することもできます。次のコマンドは、コンフィグレーション ファイルとして *Mjx4020.ini* を指定します。

```
MJXDEBW Mjx4020.ini
```

LAN 接続で LAN ボックスの IP アドレスを明示的に指定したい場合、*-h* オプションを付加します。次のコマンドは、IP アドレスとして *192.168.0.55* を指定します。

```
MJXDEBW -h192.168.0.55
```

終了するためには、*quit* コマンドを入力してください。

```
quit
```

6.2 Solaris の場合

MULTIを使ってMJX440を操作するためには、MULTI本体を起動した後、次のコマンドでMJXSERVをリモート接続してください。^{*1}

```
remote mjxserv
```

コンフィグレーションファイルを、MJXSERVの引数として指定することもできます。次のコマンドは、コンフィグレーションファイルとして *Mjx4020.ini* を指定します。

```
remote mjxserv Mjx4020.ini
```

LANボックスのIPアドレスを明示的に指定したい場合、*-h*オプションを付加します。次のコマンドは、IPアドレスとして *192.168.0.44* を指定します。

```
remote mjxserv -h192.168.0.44
```

コンフィグレーションファイルを MJXSERV の引数として指定しないときは {MULTI のインストール・ディレクトリ}/mjxserv.ini を既定値とします。

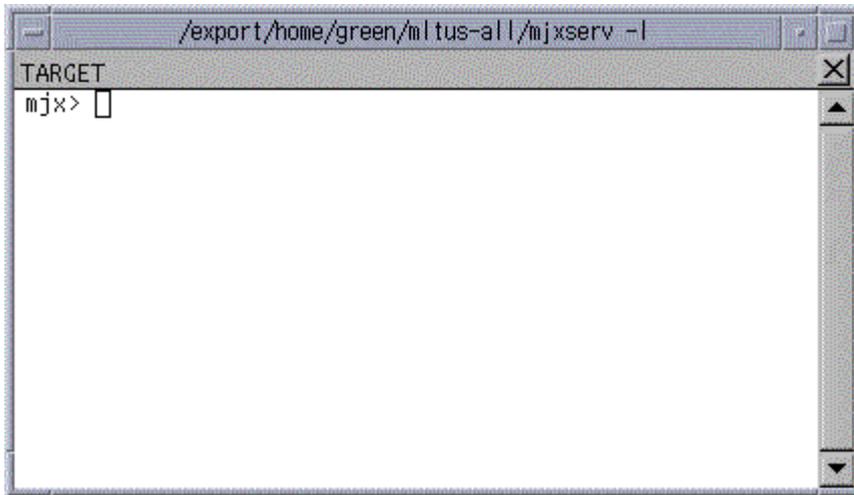
IP アドレスの指定を省略したときは、ホスト名 *mjx* で LAN ボックスに接続します。

必要に応じて、MULTI の *SERVERTIMEOUT* 変数を変更してください。

正常にリモート接続できた場合は、MULTI のターゲット ウィンドウが開きます。

^{*1} ビルダから操作する場合は、サーバー名を *mjxserv* と指定し、リモート ボタンを押します。

MULTI のターゲット ウィンドウ



終了するためには、MULTI のコマンド ウィンドウで `quit` コマンドを入力してください。

quit

第七章 MJXDEBW コマンド

Windows の場合

MJXDEBW コマンドは、MJXDEBW のコマンド入力フィールドから入力できるコマンドです。
MJXDEBW ウィンドウ内の、コマンド サブ ウィンドウには、次の機能があります。

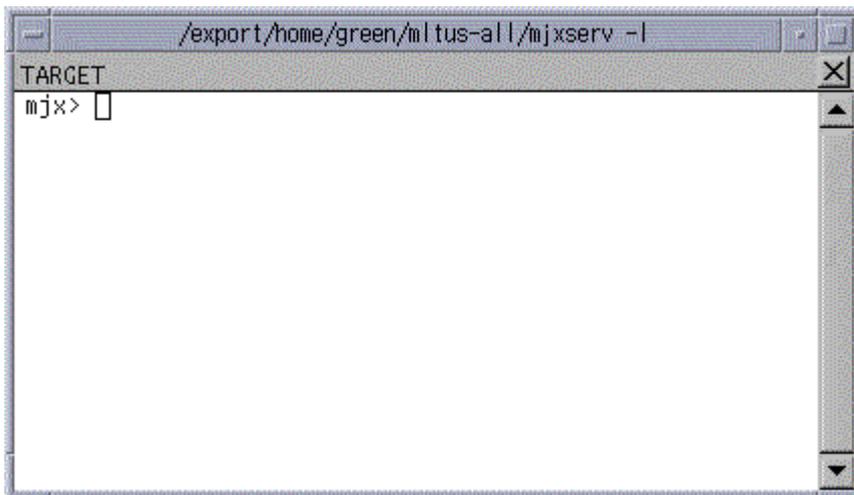


- 下位側は、MJXDEBW コマンドを入力するためのコマンド入力フィールドです。
- 上位側は、MJXDEBW コマンドの実行結果を表示するコマンド応答フィールドです。
- コマンド応答フィールドに表示できる行数は、「表示」-「オプション」-「表示」メニューで変更することができます。

Solaris の場合

MJXDEBW コマンドは、MULTI のターゲット ウィンドウ内で実行するコマンドです。コマンド プロンプトが表示されている場面で入力できます。結果の表示後、再びコマンド プロンプトが表示されます。

MULTI のターゲット ウィンドウ



MJXDEBW コマンドの一覧は、次のとおりです。

<u>ABORT</u>	ユーザ プログラムを強制停止します
<u>BATCH</u>	バッチ ファイルに記述された MJXDEBW コマンドを実行します
<u>BP</u>	ブレークポイントを表示/解除/有効/無効にします
<u>BP/A</u>	データ アクセス ブレークポイントを設定します
<u>BP/H</u>	命令ハードウェア ブレークポイントを設定します
<u>BP/S</u>	命令ソフトウェア ブレークポイントを設定します
<u>CLEAR</u>	コマンド応答フィールドをクリアします
<u>CONFIG</u>	<u>MJXDEBW の環境</u> を表示/変更します
<u>DUMP</u>	メモリの内容を表示します
<u>EXAMINE</u>	メモリの内容を変更します
<u>FILL</u>	メモリの内容をフィルします
<u>GO</u>	ユーザ プログラムを実行します
<u>HISTORY</u>	リアルタイム トレース結果を表示します
<u>INIT</u>	MJXDEBW を再初期化します
<u>JOURNAL</u>	コマンド実行結果をファイルへ出力します
<u>LOAD</u>	ファイルをメモリへダウンロードします
<u>MOVE</u>	メモリの内容をブロック転送します
<u>PIN</u>	ピンを有効/無効にします
<u>QUIT</u>	MJXDEBW を終了します
<u>REGISTER</u>	レジスタの内容を表示/変更します
<u>SETLOAD</u>	ブレークポイントの設定とトレースの設定を、ファイルから読み込みます
<u>SETSAVE</u>	ブレークポイントの設定とトレースの設定を、ファイルへ保存します
<u>STEP</u>	ユーザ プログラムをステップ実行します
<u>TRACE/M</u>	リアルタイム トレースのトレース モードを表示/設定します
<u>TRACE/I</u>	リアルタイム トレースのトレース条件を表示/設定します
<u>UNASM</u>	メモリの内容を逆アセンブル表示します
<u>VERSION</u>	バージョンを表示します
<u>WAIT</u>	ユーザ プログラムが停止するまで待ちます
<u>XPIN</u>	外部トリガ ケーブルの信号の状態を表示/設定します
<u>FLASH</u>	フラッシュ メモリの内容を書き換え、消去します

コマンド名の省略

コマンド名は、他のコマンド名と区別できる長さまで、省略することができます。

- AB [ABORT](#)と同じです。
- D [DUMP](#)と同じです。
- CL [CLEAR](#)と同じです。
- CO [CONFIG](#)と同じです。
- C エラーです。CLEAR か CONFIG の区別ができません。

コマンドライン表記のルール

コマンドラインは以下のルールに従って表記します。

- [] :省略可能な引数
- 斜体文字 :変数となる引数
- 標準文字 :固定の文字引数
- {aa|bb|cc} :| で区切られた文字列を選択
- [xxx...] :xxx が繰り返し指定が可能であることを示す
- △ :スペース
 表示簡略化のため行末に連続する ,(カンマ)は省略可能です。

MULTIを使用する場合の注意

[MULTI](#)を使用する場合、次の MJXDEBW コマンドは、無視されます。

[QUIT](#) [MJXDEBW](#)の終了

また、次の MJXDEBW コマンドは、実行しないでください。

BP/S	命令ソフトウェア ブレークポイントを設定
GO	ユーザ プログラムの実行
STEP	ユーザ プログラムのステップ実行
REG reg=val	レジスタの変更

これらの MJXDEBW コマンドを実行した場合、それ以降、[MULTI](#)と MJXDEBW 間の整合性がとれなくなる危険性があります。ソフトウェア ブレークポイントの設定、ユーザ プログラムの実行、レジスタの変更は、MULTI のコマンドで行なうようにしてください。

MULTI 上でレジスタの変更をしても、ただちに[MJXDEBW](#)のレジスタは更新されません。MJXDEBW のレジスタは、ユーザ プログラムを実行する直前に更新されます。

BATCH

バッチファイルに記述された MJXDEBW コマンドを実行します

形式:

BATCH *file*

バッチ ファイル *file* を一行ずつ読み込み、MJXDEBW コマンドとして実行します。

引数:

file MJXDEBW コマンドが記述されたバッチ ファイル名 (テキスト ファイル)

例:

BAT INIT.TXT

(INIT.TXT ファイルをバッチ ファイルとして実行します)

補足:

- BATCH コマンドは、4 レベルまでネスティングすることができます。
(バッチファイル中から別のバッチファイルを呼び出せます)
- バッチ ファイルの中の「;」で始まる行は、コメントになります。
- バッチ ファイルの中の [QUIT](#) コマンドは、[BATCH](#) コマンドのみを終了し、[MJXDEBW](#) は終了しません。

BP

ブレークポイントを表示／解除／有効／無効にします

形式:

BP {A H S}	設定されているブレークポイントを表示します。
BP N AN HN SN	設定できるブレークポイントの数を表示します。
BP/C <i>num</i> * A* H* S*	ブレークポイントを解除します。
BP/D <i>num</i> * A* H* S*	ブレークポイントを無効にします。
BP/E <i>num</i> * A* H* S*	ブレークポイントを有効にします。

引数:

A	データ アクセス ブレークポイント指定
H	命令ハードウェア ブレークポイント指定
S	命令ソフトウェア ブレークポイント指定
	A、H、S 省略時は、すべてのブレークポイント指定
N	すべてのブレークポイント数指定
AN	データ アクセス ブレークポイント数指定
HN	命令ハードウェア ブレークポイント数指定
SN	命令ソフトウェア ブレークポイント数指定
<i>num</i>	ブレークポイント番号指定 (10 進数)
*	すべてブレークポイント指定
A*	すべてのデータ アクセス ブレークポイント指定
H*	すべての命令ハードウェア ブレークポイント指定
S*	すべての命令ソフトウェア ブレークポイント指定

例:

BP	(すべてのブレークポイントを表示します)
BP N	(設定できるすべてのブレークポイントの数を表示します)
BP/C S*	(すべての命令ソフトウェア ブレークポイントを解除します)
BP/D H*	(すべての命令ハードウェア ブレークポイントを無効にします)
BP/E A*	(すべてのデータ アクセス ブレークポイントを有効にします)

補足:

ブレークポイントは、次の形式で表示されます。

No.	Type	ASID	Address	AddrMask	C/T	R/W	Data	DataMask	Enb
1	S/W	--	11111111	--	--	--	--	--	Enb
2	INST	0012	00222222	FF000000	--	--	--	--	Dis
3	ACCS	--	00003333	--	T	W	11111111	FFFFFFFF00000000	Enb
4	PROC	--	00444444	--	--	WC	--	FFFFFFFFFFFF0000	Enb

表示される各フィールドの意味は、次のとおりです。「--」は、指定されていないフィールドを表します。

No. ブレークポイント番号。(1 から始まる 10 進数)

Type ブレークポイント タイプ。

- S/W 命令ソフトウェア ブレークポイント
- INST 命令ハードウェア ブレークポイント
- ACCS データ アクセス ブレークポイント*¹
- PROC プロセッサ バス ブレークポイント*¹
- CPLX 複合ブレークポイント*¹

ASID ブレークポイント アドレスの Address Space ID (16 進数)

Address ブレークポイント アドレス。プロセッサ バス ブレークポイントでは物理アドレス、それ以外は仮想アドレス (16 進数)

AddrMask ブレークポイント アドレスに対するマスク。無効ビットが 1、有効ビットが 0 (16 進数)

C/T 複合ブレークポイント、またはトレース トリガ。

- C 複合令ブレークポイントの構成要素として使用*¹
- T トレース トリガとして使用
- 通常のブレークポイントとして使用

*¹ 未サポート

第七章 MJXDEBW コマンド

R/W	データ アクセス方法。 (データ アクセス ブレークポイントの場合) R データ読み込みブレークポイント W データ書き込みブレークポイント R/W データ アクセス ブレークポイント*1 (プロセッサ バス ブレークポイントの場合) R 非キャッシュ領域の命令/データ読み込みブレークポイント W 非キャッシュ領域の命令/データ書き込みブレークポイント WC キャッシュ領域の命令/データ書き込みブレークポイント
Data	ブレークポイント データ。データ アクセス ブレークポイント、またはプロセッサ バス ブレークポイントの場合のみ有効 (16 進数)
DataMask	ブレークポイント データに対するマスク。無効ビットが 1、有効ビットが 0 (16 進数)
Enb	ブレークポイントの有効/無効。 Enb 有効 Dis 無効

*1 データ読み込み、またはデータ書き込みでブレークするブレークポイント。

BP/A

データアクセス ブレークポイントを設定します*1

形式:

BP/A[/{R}/W}] [*asid*:]*addr* [*mask*|!*mask*] [, {/B|/W|/L|/D} *value* [*vmask*|!*vmask*]

データ アクセス ブレークポイントを設定します。

引数:

/R データ読み込みブレークポイント指定
/W データ書き込みブレークポイント指定
/R、*/W* 省略時は、読み込みと書き込みの両方のアクセス指定
asid メモリ空間 ID (16 進数、省略時は未使用)
addr ブレークポイント アドレス指定 (16 進数)
mask *addr* の無効ビット指定 (16 進数)
!*mask* *addr* の有効ビット指定 (16 進数)
/B 8 ビット *value* 指定
/W 16 ビット *value* 指定
/L 32 ビット *value* 指定
/D 64 ビット *value* 指定
/B、*/W*、*/L*、*/D* 省略時は、*value* の桁数によって自動認識 (2 桁で 8 ビット)
value ブレークポイント データ指定 (16 進数)
vmask *value* の有効ビット指定 (16 進数)
!*vmask* *value* の無効ビット指定 (16 進数)

例:

BP/A 1000 (アドレス 0x1000 番地のデータ アクセスでブレーク)
BP/A F000 00FF (アドレス 0xF000~0xF0FF 番地のデータ アクセスでブレーク)
BP/A F000 !FF00 (アドレス 0xF000~0xF0FF 番地のデータ アクセスでブレーク)
BP/A 10000, /L 22 (アドレス 0x10000 番地のデータ アクセスで、データが 32 ビットで 0x00000022 のときブレーク)
BP/A /R 1000, 22 (アドレス 0x1000 番地の読み込みで、データが 8 ビットで 0x22 のときブレーク)
BP/A /R 1000, 4400 00FF (アドレス 0x1000 番地の読み込みで、データが 16 ビットで

*1 未サポート

0x4400~0x44FF のときブレーク)

BP/A /R 1000, 4400 !FF00
(アドレス 0x1000 番地の読み込みで、データが 16 ビットで
0x4400~0x44FF のときブレーク)

BP/A /W 1000, 22
(アドレス 0x1000 番地の書き込みで、データが 8 ビットで 0x22
のときブレーク)

補足:

- データ アクセス ブレークポイントを設定すると、指定したアドレスからのデータ読み込み後、または指定したアドレスへのデータ書き込み後にユーザ プログラムを停止させることができます。
- *mask*と *vmask*は、有効にするビットを 0 に、無効にするビットを 1 に指定してください。
- *!mask*と *!vmask*は、有効にするビットを 1 に、無効にするビットを 0 に指定してください。
- トレース トリガとして、データ アクセス トリガを指定している場合は、設定できるブレークポイントの数がその分減ります。(TRACE/I コマンドを参照)
- 設定可能なデータ アクセス ブレークポイントの数は、「BP AN」コマンドで確認できます。
- データ アクセス ブレークポイントは、CPU の Debug Support Unit (DSU) の機能を使用しています。

BP/H

命令ハードウェア ブレークポイントを設定します

形式:

BP/H [*asid*:]*addr* [*mask* | *!mask*]

命令ハードウェア ブレークポイントを設定します。

引数:

asid メモリ空間 ID (16 進数、省略時は未使用)*addr* ブレークポイント アドレス指定 (16 進数)*mask* *addr* の無効ビット指定 (16 進数)*!mask* *addr* の有効ビット指定 (16 進数)

例:

BP/H F000 (アドレス 0xF000 番地の命令を実行する直前にブレーク)

BP/H F000 00FF (アドレス 0xF000~0xF0FF 番地の命令を実行する直前にブレーク)

BP/H F000 !FF00 (アドレス 0xF000~0xF0FF 番地の命令を実行する直前にブレーク)

補足:

- 命令ハードウェア ブレークポイントを設定すると、指定したアドレスの命令を実行する直前にユーザ プログラムを停止させることができます。
- *mask* は、有効にするビットを 0 に、無効にするビットを 1 に指定してください。
- *!mask* は、有効にするビットを 1 に、無効にするビットを 0 に指定してください。
- トレース トリガとして、命令ハードウェア トリガを指定している場合は、設定できるブレークポイントの数がその分減ります。(TRACE/I コマンドを参照)
- 設定可能な命令ハードウェア ブレークポイントの数は、「BP HN」コマンドで確認できません。
- 命令ハードウェア ブレークポイントは、CPU の Debug Support Unit (DSU) の機能を使用しています。

BP/S

命令ソフトウェア ブレークポイントを設定します

形式:

BP/S *addr* 命令ソフトウェア ブレークポイントを設定します。

引数:

addr ブレークポイント アドレス指定 (16 進数)

例:

BP/S F0000000 (アドレス 0xF0000000 番地の命令実行直前にブレーク)

補足:

- 命令ソフトウェア ブレークポイントを設定すると、指定したアドレスの命令を実行する直前にユーザ プログラムを停止させることができます。
- 命令ソフトウェア ブレークポイントは、RAM 領域、または ROM インサーキットしている ROM 領域にのみ設定することができます。
- 設定可能な命令ソフトウェア ブレークポイントの数は、128 です。

CLEAR

コマンド応答フィールドをクリアします

形式:

CLEAR

コマンド応答フィールドの表示をすべて消去します。

引数:

例:

CLEAR

補足:

- Solaris 版 MJXSERV ではサポートされません。

CONFIG

[MJXDEBW の環境](#)を表示/変更します

形式:

CONFIG 現在設定されている MJXDEBW の環境を表示します。
 CONFIG/S *item=value* MJXDEBW の環境設定項目 *item*を *value* に設定します。

引数:

item 環境設定項目、次のいずれかを指定

JCLOCK	JTAG クロック
TOPADDR	ROM の先頭アドレス

value 環境設定項目に設定する値

20 または 40 JCLOCK の場合 (単位MHz)
 16 進数アドレス TOPADDR の場合

例:

CONFIG (MJXDEBW の環境を表示します)
 CONFIG/S JCLOCK=20 (JTAG クロックを 20MHz に設定します)
 CONFIG/S TOPADDR=20000 (ROM の先頭アドレスを 0x20000 に設定します)

補足:

- Solaris 版 MJXSERV ではサポートされません。

DUMP

メモリの内容を表示します

形式:

DUMP/{B|W|L|D} [[*asid*:]*addr1*,[*addr2*]]

アドレス *addr1* から *addr2* までのメモリの内容を表示します。

引数:

/B 8 ビット指定
 /W 16 ビット指定
 /L 32 ビット指定
 /D 64 ビット指定
asid メモリ空間 ID (16 進数、省略時は現在値)
addr1 メモリ表示開始アドレス (16 進数)
addr2 メモリ表示終了アドレス (16 進数、省略時は *addr1*+3F)

例:

DUMP/B 1000 (0x1000 番地から 64 バイトのメモリを 8 ビットで表示します)
 DUMP/L 2000,20FF (0x2000 から 0x20FF 番地のメモリを 32 ビットで表示します)
 DUMP (前回の DUMP コマンドの続きを表示します)

補足:

- *addr2* 省略時は、64 バイト分のメモリを表示します。
- *addr1* 省略時は、前回の DUMP コマンドの続きを表示します。
- サイズ省略時は、前回のサイズが適用されます。

EXAMINE

メモリの内容を変更します

形式:

EXAMINE/{B|W|L|D} [*asid*:]*addr*=*data1*[,*data2*...]

アドレス *addr* のメモリをデータ *data1*[,*data2*...]に変更します。

引数:

/B 8ビット指定
 /W 16ビット指定
 /L 32ビット指定
 /D 64ビット指定
asid メモリ空間 ID (16進数、省略時は現在値)
addr メモリ変更開始アドレス (16進数)
data1 data2
 メモリ変更データ (16進数)

例:

EXAMINE/B 1000=55 (0x1000 番地のメモリを 8ビットデータ 0x55 に変更します)
 EXAMINE/W 3000=1,2,3 (0x3000 番地からのメモリを 16ビットデータ 0x0001、0x0002、
 0x0003 に変更します。)

補足:

- サイズ省略時は、前回のサイズが適用されます。

FILL

メモリの内容を変更します

形式:

`FILL/{B|W|L|D} [asid:]addr1,{addr2},data`

アドレス *addr1* から *addr2* までのメモリを、データ *data* でフィルします。

引数:

<code>/B</code>	8 ビット指定
<code>/W</code>	16 ビット指定
<code>/L</code>	32 ビット指定
<code>/D</code>	64 ビット指定
<i>asid</i>	メモリ空間 ID (16 進数、省略時は現在値)
<i>addr1</i>	メモリ フィル開始アドレス (16 進数)
<i>addr2</i>	メモリ フィル終了アドレス (16 進数)
<i>data</i>	フィル データ (16 進数)

例:

`FILL/B 0,3FF,FF` (0x0 から 0x3FF 番地のメモリを 8 ビットデータ 0xFF でフィルします)

`FILL/W 1000,1FFF,0` (0x1000 から 0x1FFF 番地のメモリを 16 ビットデータ 0x0000 でフィルします)

補足:

- 8M バイト以上のフィルはできません。(制限事項)

FLASH

フラッシュ メモリの内容を書き換え、消去します

形式:

FLASH/{WRITE|UPDATE} *file_name* [, [*device_type*, *access_bus_width*]]

フラッシュ メモリの内容を書き換えます。

FLASH/{ERASEONE|ERASEALL} *addr* [, [*device_type*, *access_bus_width*]]

フラッシュ メモリの内容を消去します。

/WRITE フラッシュ メモリにデータを書き込みます(既定値)

/UPDATE フラッシュ メモリのデータを更新します (セクタ単位で消去後、
書き込み)

/ERASEONE フラッシュ メモリの内容をセクタ単位に消去します

/ERASEALL フラッシュ メモリの内容を全て(チップ単位に)消去します

引数:

file_name ダウンロードするファイル名 (書き込みデータ)

addr 消去アドレス (16 進数で指定。絶対アドレスです、先頭からのオフセッ
トではありません)

device_type フラッシュ デバイス名 (省略時は、コンフィグレーション設定値を使用)

access_bus_width アクセス バス幅 (省略時は、コンフィグレーション設定値を使用)

1 1 バイト アクセス

2 2 バイト アクセス

4 4 バイト アクセス

例:

FLASH/WRITE *load.hex*,,2 (load.hex ファイルの内容をコンフィグレーション設定値で指
定されたフラッシュ デバイスに 2 バイト単位で書き込む)

FLASH/ERASEONE *BFC00040,AM29F080B*

(AM29F080B の BFC00040 (16 進) のアドレスに該当するセクタ
を消去)

補足:

- オプション省略時は、/WRITE を既定値とします。
- ファイル形式 (MJX バイナリ、S レコード、インテル HEX、COFF 形式)は自動認識されます。
- {MjxDebw.exe の格納ディレクトリ}\zax_mjx\flash\device の下に{device_type}.dat ファイルを作成して、デバイス情報定義ファイルが存在する必要があります。
ファイル レイアウトは、「[付録 J フラッシュ メモリ デバイス情報定義ファイル レイアウト](#)」を参照してください。
- ソフトウェア ブレークポイントは全てクリアされます。
- ロックビットのセット・リセットの機能があるフラッシュ デバイスでは、ロックビットがクリアされている必要があります。
- 本機能に対応するフラッシュ メモリの一覧は、「[付録 K 対応フラッシュ メモリ一覧](#)」にあります。
- 本機能はターゲット ボード上の RAM に書き込みプログラム本体、フラッシュ メモリ デバイス情報、更新データ (セクタ単位) をロードして実行されます。
実行に必要なメモリ容量は、18 K バイト + (フラッシュ デバイスの全セクタ数 x 8 バイト) + 更新されるセクタ (複数あるときはそのうちの最大) サイズです。
本機能を使用する場合、事前にRAMを使用可能にする必要があります。(手順はCPU及びターゲットボードに依存します。)
- 本機能を使用する場合、フラッシュ領域に対するキャッシュ設定を無効にする必要があります。
- 「[第八章 MJXDEBW メニューコマンド](#)」参照。

GO

ユーザ プログラムを実行します

形式:

GO `[[asid:]addr]` アドレス *addr* からユーザ プログラムを実行します。

引数:

asid メモリ空間 ID (16 進数、省略時は未使用)

addr ユーザ プログラム開始アドレス (16 進数)

例:

GO 1000 (0x1000 番地からユーザ プログラムを実行します)

GO (現在の PC からユーザ プログラムを実行します)

補足:

- Windows 版では
ユーザ プログラムが自発的に終了しない場合、WAIT あるいは ABORT コマンドで強制終了させます。(タイムアウトはありません。)
- Solaris 版では
デフォルトで 3 分のタイムアウト値でユーザ プログラムは強制終了されます。
この値を変更したいときは、事前に WAIT コマンドを発行して、値を設定してください。
GO コマンド実行後の WAIT コマンドは無効です。また、ABORT コマンドもサポートされません。
- [MULTI](#)を使用している場合は、実行しないでください。
- *addr* 省略時は、現在の PC から実行します。

- リアルタイム トレースが通常モードの場合、不完全なアドレスが表示されることがあります。

INIT

MJX を再初期化します

形式:

INIT MJX を再初期化します。

補足:

- ターゲット システムの CPU もリセットされます。

JOURNAL

コマンド実行結果をファイルへ出力します

形式:

JOURNAL/{A|W} *file*[,*mode*[,*echo*]]

コマンド実行結果をファイル *file* へ出力するようにします。

JOURNAL/E

ファイルへの出力を終了し、ファイルをクローズします。

引数:

<i>/A</i>	追記出力指定
<i>/W</i>	新規出力指定 (省略時)
<i>file</i>	出力ファイル名指定
<i>mode</i>	出力モード指定
IN	コマンドのみ出力
OUT	コマンド実行結果のみ出力
ALL (省略時)	コマンドとコマンド実行結果を出力
<i>echo</i>	エコー モード指定
OFF	ファイルへ出力する内容は画面に表示しない
ON (省略時)	ファイルへ出力する内容を画面にも表示する

例:

JOURNAL TEST.TXT	(コマンド実行結果を TEST.TXT ファイルへ出力するようにします)
JOURNAL/E	(ファイルへの出力を終了し、ファイルをクローズします)
JOURNAL/A TEST.TXT	(コマンド実行結果をファイル TEST.TXT へ追記出力するようにします)

補足:

LOAD

ファイルをメモリへダウンロードします

形式:

LOAD *file* [,*offset*] MJX バイナリ*¹、Sレコード、インテル HEX、または COFF 形式のファイル *file* をメモリへダウンロードします。

LOAD/R *file* [,*offset*] MJX バイナリ ファイル *file* を エミュレーション メモリ経由 でメモリへダウンロードします。

引数:

file ダウンロードするファイル名

offset オフセット アドレス (省略時 0)

例:

LOAD PROG1.ABS (ファイル PROG1.ABS をメモリへダウンロードします)

LOAD PROG1.ABS,2000 (ファイル PROG1.ABS を実アドレス+0x2000 番地のメモリへダウンロードします)

LOAD/R PROG2.MJX (ファイル PROG2.MJX をエミュレーション メモリ経由でメモリへダウンロードします)

補足:

- ファイル形式は、自動認識されます。
- ファイル名の拡張子を省略した場合は、.mjx が補完されます。
- LOAD/R コマンド時、[MJX バイナリ ファイル](#)の中に、エミュレーション メモリを指すレコードがある場合、正しくダウンロードできません。
- LOAD/R コマンドを実行するとエミュレーションメモリ内容は破壊されます。

*¹ MJX バイナリ ファイルについては「第九章 高速ダウンロード」を参照してください。

MOVE

メモリの内容をブロック転送します

形式:

MOVE/{B|W|L|D} [*asid*:]*addr1*,{*addr2*},{*asid*:}*addr3*

アドレス *addr1* から *addr2* までのメモリを *addr3* へブロック転送します。

引数:

/B	8 ビット指定
/W	16 ビット指定
/L	32 ビット指定
/D	64 ビット指定
<i>asid</i>	メモリ空間 ID (16 進数、省略時は現在値)
<i>addr1</i>	転送元メモリ開始アドレス (16 進数)
<i>addr2</i>	転送元メモリ終了アドレス (16 進数)
<i>addr3</i>	転送先メモリ アドレス (16 進数)

例:

MOVE 1000,10FF,2000 (0x1000 から 0x10FF 番地までのメモリを 0x2000 番地へブロック転送します)

補足:

- 8M バイト以上のブロック転送はできません。(制限事項)
- 転送元と転送先のメモリブロックがオーバーラップしている場合の結果は不定です。

QUIT

[MJXDEBW](#)を終了します

形式:

QUIT MJXDEBW を終了します。

補足:

- バッチ ファイルの中の QUIT コマンドは、[BATCH](#)コマンドのみを終了し、MJXDEBW は終了しません。
- Solaris 版 MJXSERV ではサポートされません。

REGISTER

レジスタの内容を表示/変更します

形式:

REGISTER	すべてのレジスタの内容を表示します。
REGISTER/GP	すべての汎用レジスタの内容を表示します。
REGISTER/CPO	すべてのコプロセッサ 0 レジスタの内容を表示します。
REGISTER/F	すべての浮動小数点レジスタの内容を表示します。
REGISTER <i>reg</i>	レジスタ <i>reg</i> の内容を表示します。
REGISTER/GP <i>num</i>	汎用レジスタ番号 <i>num</i> の内容を表示します。
REGISTER/CPO <i>num</i>	コプロセッサ 0 レジスタ番号 <i>num</i> の内容を表示します。
REGISTER/F <i>num</i>	浮動小数点レジスタ番号 <i>num</i> の内容を表示します。
REGISTER <i>reg=data</i>	レジスタ <i>reg</i> をデータ <i>data</i> に変更します。
REGISTER/GP <i>num=data</i>	汎用レジスタ番号 <i>num</i> をデータ <i>data</i> に変更します。
REGISTER/CPO <i>num=data</i>	コプロセッサ 0 レジスタ番号 <i>num</i> をデータ <i>data</i> に変更します。
REGISTER/T [<i>tlbnum</i>]	TLB エントリを表示します。

引数:

<i>reg</i>	レジスタ名指定 (「 付録 G レジスタ名一覧 」を参照してください)
<i>num</i>	レジスタ番号指定 (10 進数、0~31)
<i>data</i>	レジスタ変更データ (16 進数)
<i>tlbnum</i>	TLB エントリ番号 (10 進数、0~最大 TLB エントリ番号、省略時はすべてのエントリ)

例:

REGISTER	(すべてのレジスタの内容を表示します。)
REGISTER R1	(レジスタ R1 の内容を表示します)
REGISTER PC=2000	(PC を 0x2000 に変更します)
REGISTER/T	(すべての TLB エントリを表示します)

補足:

- MULTI を使用している場合は、レジスタを変更しないでください。

SETLOAD

ブレイクポイントの設定とトレースの設定を、ファイルから読み出します。

形式:

SETLOAD *filename* ブレイクポイントの設定とトレースの設定を読み出します。

引数:

filename 読み出すファイル名

例:

SETLOAD abc.mjs (abc.mjs ファイルを読み出します)

補足:

- *filename* は SETSAVE で作成されたファイルを指定してください

SETSAVE

ブレイクポイントの設定とトレースの設定を、ファイルへ保存します

形式:

SETSAVE *filename* ブレイクポイントの設定とトレースの設定をファイルへ保存します。

引数:

filename 保存するファイル名

例:

SETSAVE abc.mjs (abc.mjs ファイルを保存します)

補足:

- 保存したファイルは SETLOAD で指定することができます

STEP

ユーザ プログラムをステップ実行します

形式:

STEP [*num*] ユーザ プログラムを *num*回ステップ実行します

引数:

num ステップ実行回数指定 (10 進数、省略時は 1)

例:

STEP (ユーザ プログラムを 1 回ステップ実行します。)

STEP 10 (ユーザ プログラムを 10 回ステップ実行します。)

補足:

- MULTI を使用している場合は、実行しないでください。

TRACE/M

リアルタイム トレースのトレース モードを表示/設定します

形式:

TRACE/M FL トレース モードを完全モードに設定します。
 TRACE/M RT トレース モードを通常モードに設定します。
 TRACE トレース モードとすべてのトレース条件を表示します。

引数:

例:

TRACE (トレース モードを表示)
 TRACE/M FL (トレース モードを完全モードに設定します)

補足:

- 完全モードと通常モードの違いは、次のとおりです。

モード名	意味	備考
完全モード	すべての分岐命令実行の情報をトレース情報として記録します。	正確にトレース情報を記録することを優先させます。 そのため、分岐が連続した場合は分岐アドレスがトレース情報として出力されるまで一時的にCPUを停止させます。 (プログラムがリアルタイムには実行されないことがあります。)
通常モード	できるだけ分岐命令実行の情報をトレース情報として記録します。	CPUがプログラムを実時間で(リアルタイムに)実行することを、トレース記録よりも優先します。 そのため、分岐が連続した場合は、分岐アドレスが不完全なままトレース情報として記録されることがあります。その場合は history コマンドによるトレース結果表示が出来ないことがあります。

TRACE / I

リアルタイム トレースのトレース条件を表示/設定します

形式:

TRACE / I	BM[, <i>length</i>]	トレース条件をビギン モニタに設定します。
TRACE / I	EM	トレース条件をエンド モニタに設定します。
TRACE / I	BT, <i>num</i>	トレース条件をビギン トリガに設定します。
TRACE / I	ET, <i>num</i>	トレース条件をエンド トリガに設定します。
TRACE / I	MT, <i>length</i> , <i>num</i>	トレース条件をミッド トリガに設定します。
TRACE / R		すべてのトレース条件を無効にします。
TRACE		すべてのトレース条件とトレース モードを表示します。

引数:

<i>length</i>	トレース トリガ発生からトレースを停止させるまでの長さ指定 (10 進数、省略時 524288)
<i>num</i>	トレース トリガとして使用するブレークポイント番号 (10 進数)

例:

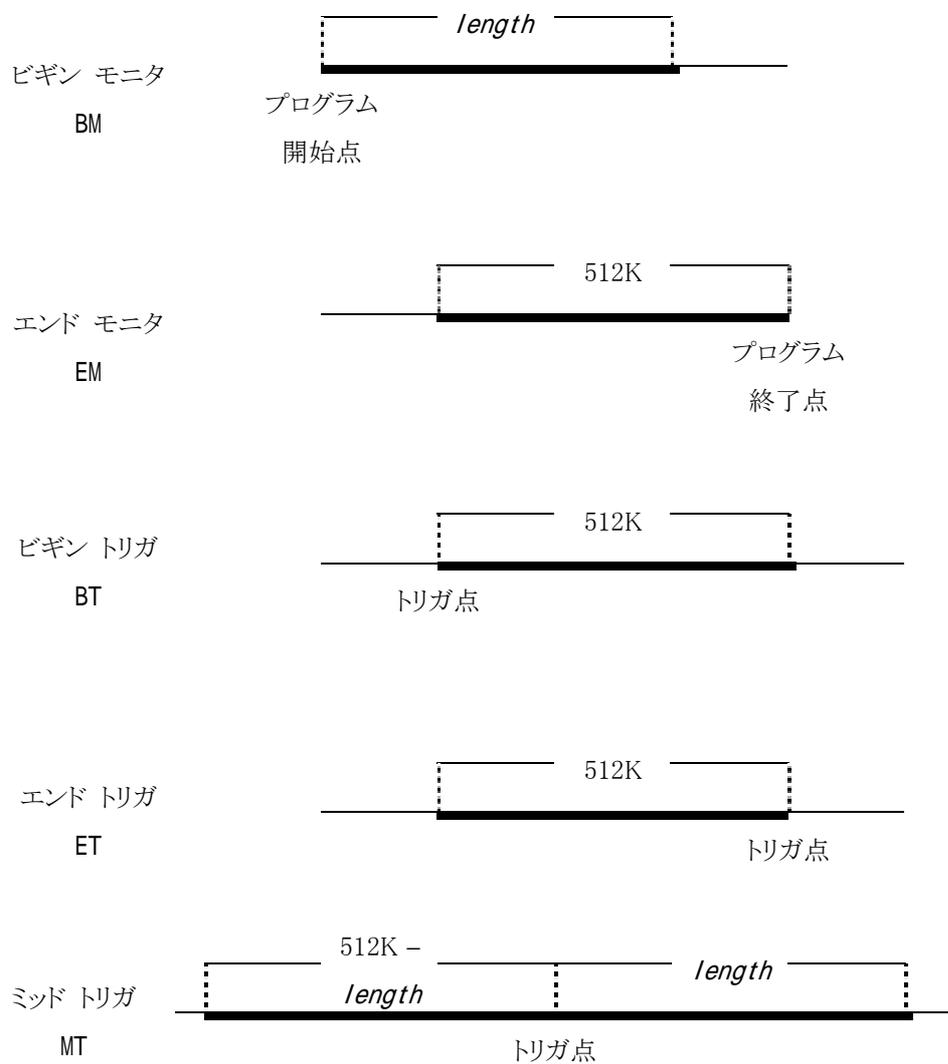
TRACE / I	BM	(トレース条件をビギン モニタに設定します)
TRACE / I	BT, 1	(トレース条件をビギン トリガに設定し、トレース トリガをブレークポイント番号 1 に設定します)
TRACE / I	MT, 2048, 3	(トレース条件をミッド トリガに設定し、トレース トリガをブレークポイント番号 3 に設定し、トレース トリガ発生からトレースを停止させるまでの長さを 2048 に設定します)
TRACE		(すべてのトレース条件を表示します)
TRACE / R		(すべてのトレース条件を無効にします)

補足:

- トレース トリガとして使用できるブレークポイントは、次の 2 種類です。トレース トリガとして設定した場合は、ブレークポイントとして機能しません。

データ アクセス ブレークポイント ([BP/A](#))命令ハードウェア ブレークポイント ([BP/H](#))

トレース範囲:



UNASM

メモリの内容を逆アセンブル表示します

形式:

UNASM `[[asid:]addr[,count]]`

アドレス *addr* から *count* 命令分のメモリの内容を逆アセンブル表示します。

引数:

asid メモリ空間 ID (16 進数、省略時は現在値)
addr 逆アセンブル表示メモリ開始アドレス (16 進数)
count 逆アセンブル表示命令数 (10 進数、省略時 16)

例:

UNASM 1000 (0x1000 番地から 16 命令分のメモリを逆アセンブル表示します)

UNASM (前回の UNASM コマンドの続きを表示します)

補足:

- *count* 省略時は、16 命令分のメモリを逆アセンブル表示します。
- *addr* 省略時は、前回の UNASM コマンドの続きを表示します。

VERSION

ソフトウェアのバージョンを表示します

形式:

VERSION

次のバージョン情報を表示します。

MJXDEBW	MJXDEBW.EXE ソフトウェアのバージョン
DLL	CPU 依存 DLL ライブラリのバージョン
DRV	デバイス ドライバのバージョン
MJX	エミュレータのバージョン
CPU	CPU 情報
	DCU のバージョン

例:

VERSION

補足:

WAIT

ユーザ プログラムが停止するまで待ちます

形式:

`WAIT [time]` ユーザ プログラムが停止するまで待ちます。指定時間内に停止しない場合は、ユーザ プログラムを強制停止します。

引数:

time 待ち時間 (10 進数、単位ミリ秒、省略時は∞)

例:

[GO](#) (ユーザ プログラムを実行します)
`WAIT 1000` (ユーザ プログラムが停止するまで待ちます。1 秒以内に停止しない場合は、ユーザ プログラム強制停止します)

補足:

- 待ち時間はミリ秒単位で指定できますが、実際の精度は百ミリ秒程度です。
- Windows 版では、バッチファイル内で使用します。
- Solaris 版では、GO コマンドのタイムアウト値を変更するのに、GO コマンドに先行して発行します。GO コマンド実行後の発行は無効です。

XPIN

外部トリガ ケーブルの信号の状態を表示/設定します

形式:

XPIN 外部トリガ ケーブルの信号の状態を表示します。
 XPIN *ch, level* 外部トリガ ケーブルの出力信号 *ch* をレベル *level* に設定します。

引数:

ch 外部トリガ ケーブルの出力信号ピンの指定
 1 EXTOUT1
 2 EXTOUT2
level 出力ピンのレベル指定
 0 LOW レベル
 1 HIGH レベル

例:

XPIN (外部トリガ ケーブルの信号の状態を表示します)
 XPIN 1,0 (外部トリガ ケーブルの出力信号EXTOUT1をLOWレベルに設定します)

補足:

- 出力信号は負論理です。

第八章 MJXDEBW メニューコマンド

この章ではデバッガ[MJXDEBW](#)のメニュー コマンドの使い方について記述しています。

【注意】 メニューコマンドは Solaris 版 MJXSERV ではサポートされません。

メニュー コマンドを使用すると、「第七章 [MJXDEBW コマンド](#)」に相当することを、メニューから実行させることができます。各メニューには、次のようにコマンドが割り当てられています。



図8-1 MJXDEBW のメニューバー

ファイル(F)	ファイル メニューには、ファイルからコマンドを読み込みバッチ処理を行なうコマンド、コマンドの実行結果をファイルに残すジャーナル処理のコマンドなどが割り当てられています。
エミュレーション(E)	エミュレーション メニューには、ユーザ プログラムの実行制御、プログラムのダウンロード、初期化コマンドが割り当てられています。
表示(V)	表示メニューには、次のウィンドを表示するためのコマンドが含まれています。 メモリ ウィンドウ、レジスタ ウィンドウ、逆アセンブル ウィンドウ、トレース結果ウィンドウ。
変更(M)	メモメモリ/レジスタなどを変更するためのコマンドが割り当てられています。
設定(S)	ブレークポイントやトレース モードを設定するためのコマンドが割り当てられています。

第八章 MJXDEBW メニューコマンド

ウィンドウ(W)	ウィンドウの表示形式を変更するためのコマンドが割り当てられています。 (Windows 準拠)
ヘルプ(H)	MJXDEBW のバージョン情報を表示するためのコマンドが割り当てられています。

ファイル(F) ~ バッチ(B)...

BATCH

機能:

バッチ ファイルに記述された MJXDEBW コマンドを連続して実行します。

操作:



ダイアログ ボックスで、ファイル名を指定し、「開く」ボタンを押してください。

補足:

- 「第七章 [BATCH](#)」参照。

ファイル(F) ~ ジャーナル開始(S)...

JOURNAL

機能:

コマンド実行結果をファイルへ出力する、ジャーナル機能を開始します。

操作:



ダイアログ ボックスで、出力ファイル名、ジャーナル モード、ジャーナル エコー、書き込みモードを指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [JOURNAL](#)」参照。

ファイル(F) ~ ジャーナル停止(E)

JOURNAL/E

機能:

ジャーナル機能を終了します。

補足:

- 「第七章 [JOURNAL](#)」参照。

ファイル(F) ~ トレース結果(T)...

JOURNAL
HISTORY

機能:

トレース結果をファイルへ出力します。

操作:



ダイアログ ボックスで、表示開始サイクル、サイクル数、表示モード、出力ファイルを指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [JOURNAL](#)」、「第七章 [HISTORY](#)」参照。

エミュレーション(E) ～ 実行(G)...

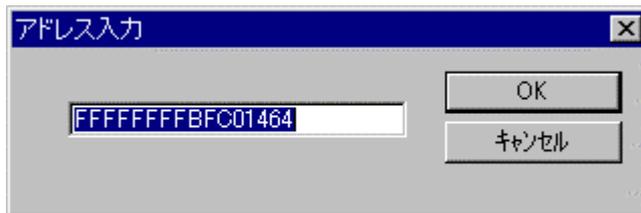
GO *addr*

Ctrl+G

機能:

プログラムの実行を開始するアドレスを指定してユーザ プログラムを実行します。

操作:



ダイアログ ボックスで、プログラムの実行を開始するアドレス(16 進数)を指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [GO](#)」参照。

エミュレーション(E) ～ 再スタート(R)

GO

F5

機能:

現在の PC のアドレスからユーザ プログラムを実行します。

補足:

- 「第七章 [GO](#)」参照。

エミュレーション(E) ~ 中断(B)

ABORT
Ctrl+B

機能:

ユーザ プログラムを強制停止します。

補足:

- 「第七章 [ABORT](#)」参照。

エミュレーション(E) ~ ステップ(S)

STEP
F10

機能:

ユーザ プログラムをステップ実行します。

補足:

- 「第七章 [STEP](#)」参照。

エミュレーション(E) ~ N-ステップ(N)...

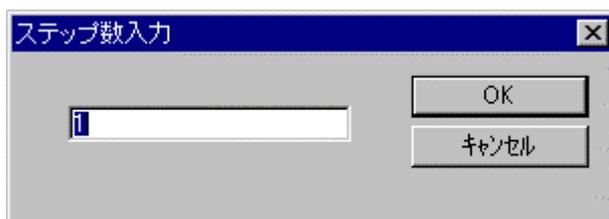
STEP *num*

Ctrl+S

機能:

ユーザ プログラムを指定回数分ステップ実行します。

操作:



ダイアログ ボックスで、ステップ実行回数(10 進数)を指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [STEP](#)」参照。

エミュレーション(E) ~ ダウンロード(L)...

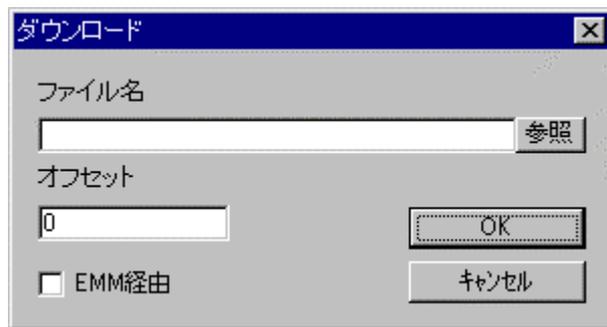
LOAD

Ctrl+L

機能:

MJX バイナリ^{*1}、S レコード、インテル HEX、または COFF 形式のファイル *file* をメモリへダウンロードします。

操作:



ダイアログ ボックスで、ファイル名、オフセット、EMM 経由を指定し、「OK」ボタンを押してください。

補足:

- ファイル形式は、自動認識されます。
- 「第七章 [LOAD](#)」参照。

^{*1} MJX バイナリ ファイルの作成方法については「第九章 高速ダウンロード」を参照してください。

エミュレーション(E) ～ 初期化(I)

INIT

機能:

[MJXDEBW](#)を再初期化します。

補足:

- 「第七章 [INIT](#)」参照。

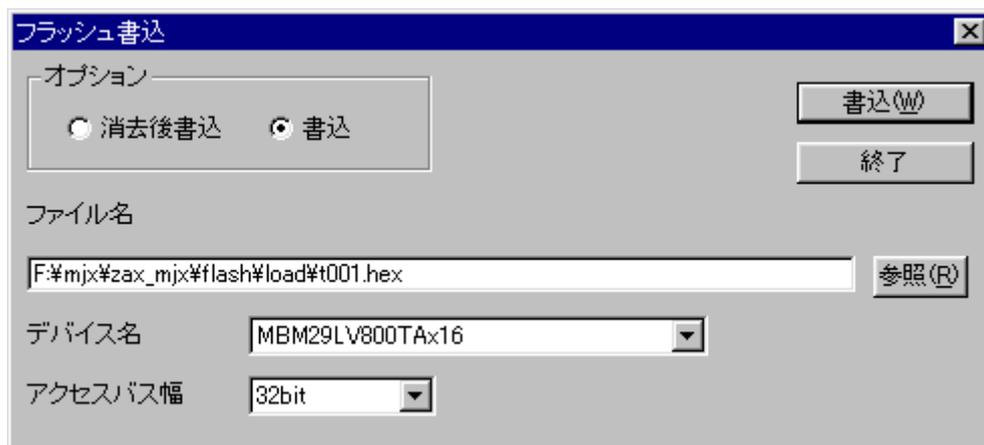
エミュレーション(E) ~ フラッシュメモリ(F) ~ 書込(W)...

FLASH/WRITE

機能:

MJX バイナリ、Sレコード、インテル HEX、または COFF 形式ファイルをフラッシュ メモリへ書き込みます。

操作:



ダイアログ ボックスで、ファイル名、デバイス名、アクセス バス幅を指定して、「書込」ボタンを押してください。

補足:

- ファイル形式は自動認識されます。
- 「第七章 [FLASH](#)」参照。

エミュレーション(E) ~ フラッシュメモリ(F) ~ 消去(E)...

FLASH/ERASEONE

FLASE/ERASEALL

機能:

指定された消去アドレスに該当するフラッシュメモリをセクタ消去、またはチップ消去します。

操作:



ダイアログボックスで消去アドレス、デバイス名、アクセスバス幅を指定して、「消去」ボタンを押してください。

補足:

- 「第七章 [FLASH](#)」参照。

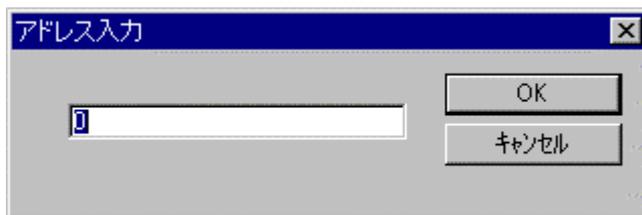
表示(V) ~ メモリ(M)...

DUMP

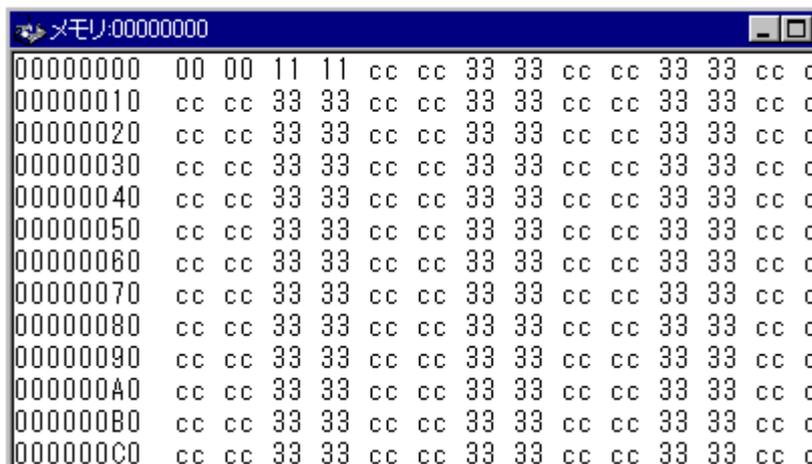
機能:

メモリの内容を表示する、メモリ ウィンドウを開きます。

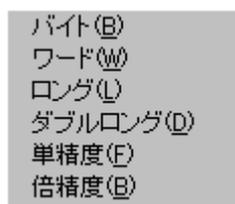
操作:



ダイアログ ボックスで、メモリ表示を開始するアドレス(16 進数)を指定し、「OK」ボタンを押してください。



新しくメモリ ウィンドウが表示されます。



メモリ ウィンドウ内で、マウスの右ボタンを押すと、ポップアップ メニューが表示され、表示形式を変更することができます。

補足:

- メモリ ウィンドウは複数開くことができます。開くことができるウィンドウ個数は、「[表示\(V\)](#)
[～ オプション\(O\) ～ 表示\(V\)](#)」メニューで変更することができます。
- 「第七章 [DUMP](#)」参照。

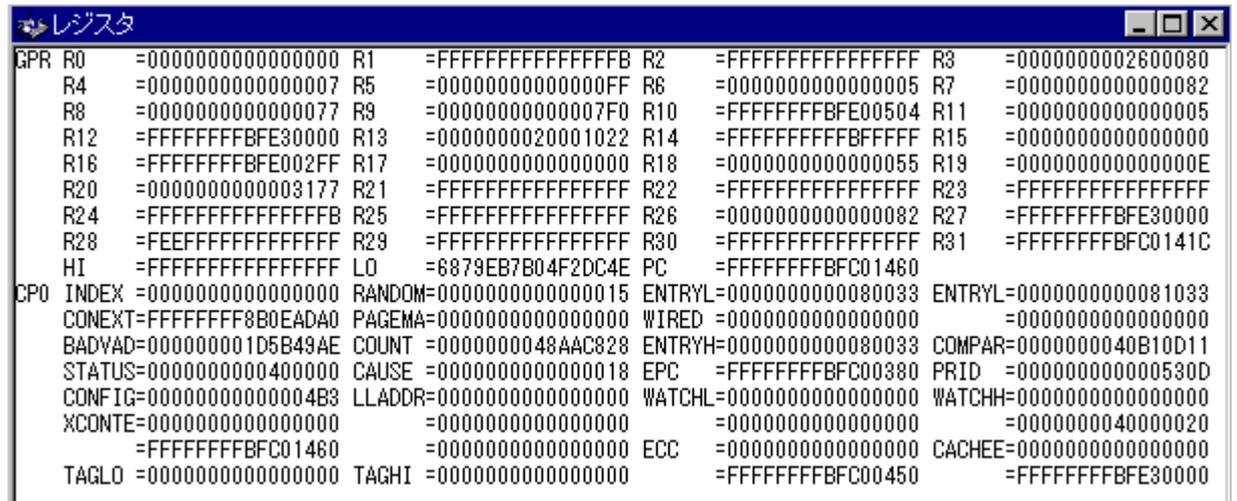
表示(V) ~ レジスタ(R)

REGISTER

機能:

レジスタの内容を表示する、レジスタ ウィンドウを開きます。

操作:



```

レジスタ
GPR R0 =0000000000000000 R1 =FFFFFFFFFFFFFFFB R2 =FFFFFFFFFFFFFFF R3 =0000000002600080
R4 =0000000000000007 R5 =00000000000000FF R6 =0000000000000005 R7 =0000000000000082
R8 =0000000000000077 R9 =00000000000007F0 R10 =FFFFFFFFBFE00504 R11 =0000000000000005
R12 =FFFFFFFFBFE30000 R13 =0000000020001022 R14 =FFFFFFFFBFFFFFFF R15 =0000000000000000
R16 =FFFFFFFFBFE002FF R17 =0000000000000000 R18 =0000000000000055 R19 =000000000000000E
R20 =0000000000003177 R21 =FFFFFFFFFFFFFFF R22 =FFFFFFFFFFFFFFF R23 =FFFFFFFFFFFFFFF
R24 =FFFFFFFFFFFFFFFB R25 =FFFFFFFFFFFFFFF R26 =0000000000000082 R27 =FFFFFFFFBFE30000
R28 =FFFFFFFFFFFFFFF R29 =FFFFFFFFFFFFFFF R30 =FFFFFFFFFFFFFFF R31 =FFFFFFFFBFC0141C
HI =FFFFFFFFFFFFFFF LO =6879EB7B04F2DC4E PC =FFFFFFFFBFC01460
CPO INDEX =0000000000000000 RANDOM=0000000000000015 ENTRYL=000000000080033 ENTRYR=0000000000000000
CONEXT=FFFFFFFF8B0EADA0 PAGEMA=0000000000000000 WIRED =0000000000000000
BADYAD=000000001D5B49AE COUNT =0000000048AAC828 ENTRYH=000000000080033 COMPAR=0000000040B10D11
STATUS=0000000000400000 CAUSE =0000000000000018 EPC =FFFFFFFFBFC00380 PRID =000000000000530D
CONFIG=00000000000004B3 LLADDR=0000000000000000 WATCHL=0000000000000000 WATCHH=0000000000000000
XCONTE=0000000000000000 =0000000000000000 =0000000000000000 =0000000040000020
=FFFFFFFFBFC01460 =0000000000000000 ECC =0000000000000000 CACHEE=0000000000000000
TAGLO =0000000000000000 TAGHI =0000000000000000 =FFFFFFFFBFC00450 =FFFFFFFFBFE30000

```

新しくレジスタ ウィンドウが表示されます。

補足:

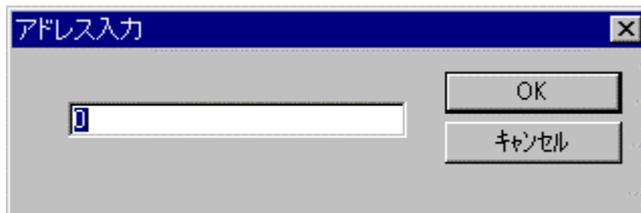
- 「第七章 [REGISTER](#)」参照。

表示(V) ~ 逆アセンブル(D)...

UNASM

機能:

メモリの内容を逆アセンブル表示する、逆アセンブラ ウィンドウを開きます。



操作:

ダイアログ ボックスで、逆アセンブル表示を開始するアドレス(16 進数)を指定し、「OK」ボタンを押してください。

新しく逆アセンブラ ウィンドウが表示されます。

Address	Hex	Instruction	Comment
00000000	00001111	.WORD	0x00001111
00000004	8894B7B3	LWL	R20,0xFFFFB7B3 (R4)
00000008	CCCC7333	.WORD	0xCCCC7333
0000000C	DEDE1333	LD	R30,0x1333 (R22)
00000010	CC4C6323	.WORD	0xCC4C6323
00000014	57DC39A3	BNEL	R30,R28,0x0000E6A4
00000018	CCCC1733	.WORD	0xCCCC1733
0000001C	68CE3033	LDL	R14,0x3033 (R6)
00000020	CCCC3B33	.WORD	0xCCCC3B33
00000024	E40C3633	SWC1	R12,0x3633 (R0)
00000028	ECCC3337	.WORD	0xECCC3337
0000002C	18CC9237	.WORD	0x18CC9237
00000030	CCCC3333	.WORD	0xCCCC3333
00000034	8C8C4333	LW	R12,0x4333 (R4)
00000038	C4CC33BB	LWC1	R12,0x33BB (R6)
0000003C	CC8C6633	.WORD	0xCC8C6633
00000040	CCCC3B32	.WORD	0xCCCC3B32

補足:

- 逆アセンブラ ウィンドウは複数開くことができます。開くことができるウィンドウ個数は、「表示(V) ~ オプション(O) ~ 表示(V)」メニューで変更することができます。
- 「第七章 [UNASM](#)」参照。

表示(V) ~ トレース結果(T)...

HISTORY

機能:

リアルタイム トレース結果の内容を表示する、トレース結果ウィンドウを開きます。



操作:

ダイアログボックスで、表示開始サイクル、表示モードを指定し、「OK」ボタンを押してください。
トレース結果ウィンドウが表示されます。

トレース結果				
745	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
993	BFC01468	00000000	NOP	
1240	BFC01460	2694FFFF	ADDIU	R20,R20,0xFFFFFFFF
1241	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
1489	BFC01468	00000000	NOP	
1736	BFC01460	2694FFFF	ADDIU	R20,R20,0xFFFFFFFF
1737	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
1985	BFC01468	00000000	NOP	
2232	BFC01460	2694FFFF	ADDIU	R20,R20,0xFFFFFFFF
2233	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
2481	BFC01468	00000000	NOP	
2728	BFC01460	2694FFFF	ADDIU	R20,R20,0xFFFFFFFF
2729	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
2977	BFC01468	00000000	NOP	
3224	BFC01460	2694FFFF	ADDIU	R20,R20,0xFFFFFFFF
3225	BFC01464	1680FFFE	BNE	R20,R0,0xBFC01460
3473	BFC01468	00000000	NOP	

(逆アセンブラ表示の例)

(1)サイクル番号

CPU サイクルの番号を示します。

零点はトレースモードにより異なります。

- ビギンモニター - プログラム開始点
- エンドモニター - プログラム終了点
- ビギントリガー - トリガー点
- エンドトリガー - トリガー点
- ミッドトリガー - トリガー点

(2)外部入力信号

外部入力 EXTI1,EXTI2 および EXTI3 信号のレベルを示します。

CPU サイクルとの関連により外部信号の遷移状態が判ります。

外部入力信号はトレースクロックに同期して捕捉されます。

トレースクロック = CPU クロック/2

トレースクロック = CPU クロック/4

の場合は、外部入力信号は 2 サイクルまたは 4 サイクル毎に表示されます。

(3)PCST

CPU の実行状態を示す簡単な文字列を表示します。

PCST コード、シンボル、表示文字列と CPU 実行状態を以下の表に示します。

PCST コード	シンボル	表示	CPU 実行状態
111	STL	Pipeline Stall	パイプラインストール
110	JMP	Jump instruction	分岐命令実行および PC アドレス出力の開始
101	BRT	Branch taken	直接分岐命令の実行(PC アドレス出力なし)
100	EXP	Exception	例外発生および例外コード出力開始
011	SEQ	Sequential	非分岐命令実行
010	TST	Trigger at stall	トリガー発生(パイプラインストール中)
001	TSQ	Trigger at exec	トリガー発生(非分岐命令実行中)
000	DBM	Debug mode	デバッグモード

(4)アドレス情報表示領域

(i)PCST が JMP の場合

分岐命令による分岐先のアドレスと ASID(ASID トレースが有効な場合)が表示されます。

JMP ステータスに続くアドレス出力中に再度 JMP ステータスが発生した場合、

先行する JMP ステータスに伴うアドレス出力は中止されます。

この場合、下位の有効なビット数が表示されます。分岐先アドレスの決定は出来ません。

(ii)PCST が EXP の場合

例外コードと例外ベクタアドレスが表示されます。

EXP ステータスに続く例外コード出力中に再度 EXP ステータスが発生した場合、

先行する EXP ステータスに伴う例外コード出力は中止されます。

この場合、例外ベクタアドレスの表示は行いません。

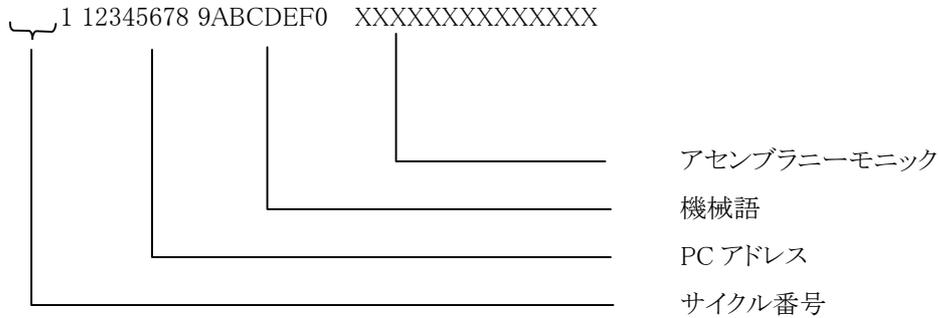
例外コードは 32bitCPU の場合 3bit、64bitCPU の場合 4bit です。

(iii)その他の場合

アドレス情報はありません。

2 アセンブラ表示

アセンブラ表示フォーマット



(1)サイクル番号

CPU サイクルの番号を示します。

零点はトレースモードにより異なります。

- ビギンモニター - プログラム開始点
- エンドモニター - プログラム終了点
- ビギントリガー - トリガー点
- エンドトリガー - トリガー点
- ミッドトリガー - トリガー点

(2)PC アドレス

命令のアドレスを示します。

(3)機械語

命令の機械語コードを示します。

(4)アセンブラニーモニック

アセンブラニーモニックを示します。

注意事項

(1)アドレス喪失

JMP ステータスに続くアドレス出力中に再度発生した JMP ステータスにより先行する JMP ステータスに伴うアドレス出力が中止された場合、アドレスの決定は出来ません。再度完全なアドレスが得られるまで、アドレス喪失状態となりこの間のアセンブラ表示は出来ません。中止された例外コード出力に付いても同様です。

(2)命令-PCST ミスマッチ

アセンブラ表示解析中に PCST と実際の命令(解析したアドレスに従う)が合わなかった場合、すなわち一方が分岐命令を示しているのもう一方が非分岐命令を示しているときは、トレース捕捉回路の異常と考え、トレース結果アセンブラ表示は中止されます。

表示(V) ~ コマンド応答クリア(C)

CLEAR

機能:

コマンド応答フィールドをクリアします。

補足:

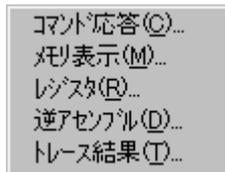
- 「第七章 [CLEAR](#)」参照。

表示(V) ~ オプション(O) ~ フォント(F)

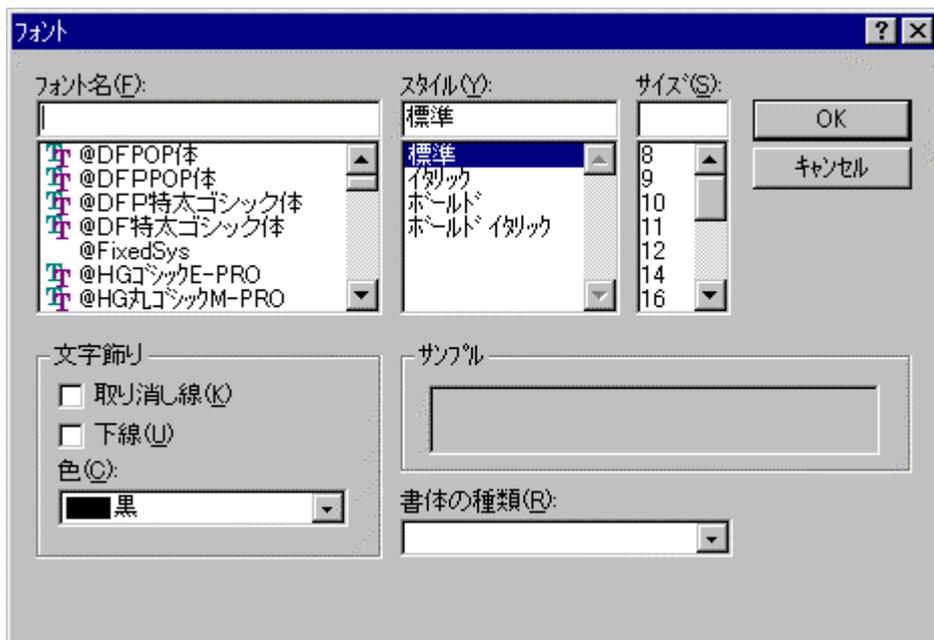
機能:

ウィンドウに表示する文字のフォントを変更します。

操作:



メニューで、フォントを変更したいウィンドウを選びます。



ダイアログ ボックスで、フォント、スタイル、サイズを指定し、「OK」ボタンを押してください。

補足:

表示(V) ~ オプション(O) ~ 表示(V)

機能:

コマンド ウィンドウ、メモリ ウィンドウ、逆アセンブル ウィンドウの属性を指定します。

操作:



ダイアログ ボックスで、各ウィンドウの属性を指定し、「OK」ボタンを押してください。

補足:

-

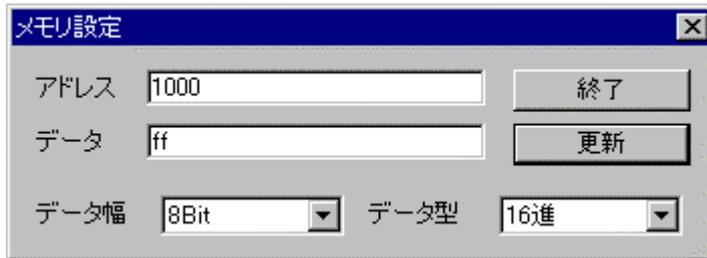
変更(M) ~ メモリ(M)

EXAMINE

機能:

メモリの内容を変更します。

操作:



ダイアログ ボックスで、変更するメモリのアドレス(16 進数)、データ、データ幅、データ型を指定し、「更新」ボタンを押してください。

「更新」ボタンを押すとメモリが変更され、ダイアログのアドレスが自動的に進みます。続けてメモリを変更する場合は、同様の手順を繰り返してください。

メモリの変更が終了したら、「終了」ボタンを押してください。

補足:

- 「第七章 [EXAMINE](#)」参照。

変更(M) ~ I/O ポート(P)...

機能:

I/O ポートの内容を変更します。

操作:



ダイアログ ボックスで、変更する I/O ポートの アドレス(16 進数)、データ、データ幅、データ型 アドレス増分を指定し、「書込」または「読出」ボタンを押してください。

「次へ」ボタンを押すと I/O ポートアクセスを行わずにアドレスの増減ができます。

補足:

-

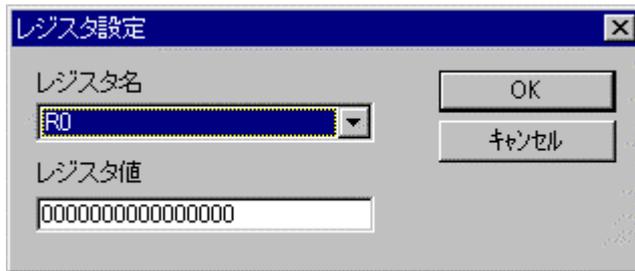
変更(M) ~ レジスタ(R)...

REGISTER

機能:

レジスタの内容を変更します。

操作:



ダイアログ ボックスで、レジスタ名、レジスタ値を指定し、「OK」ボタンを押してください。

補足:

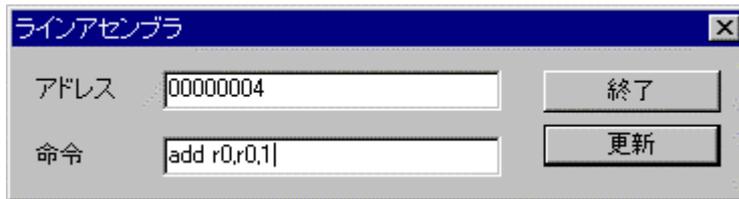
- 「第七章 [REGISTER](#)」参照。

変更(M) ~ アセンブラ(A)...

機能:

メモリの内容をアセンブラで変更します。

操作:



ダイアログ ボックスで、変更するメモリのアドレス(16 進数)、アセンブラの命令を指定し、「更新」ボタンを押してください。

「更新」ボタンを押すとメモリが変更され、ダイアログのアドレスが自動的に進みます。続けてメモリを変更する場合は、同様の手順を繰り返してください。

メモリの変更が終了したら、「終了」ボタンを押してください。

補足:

- シンボルは入力できません。
- 16 進数は、0x1234 のように先頭に「0x」を付加してください。

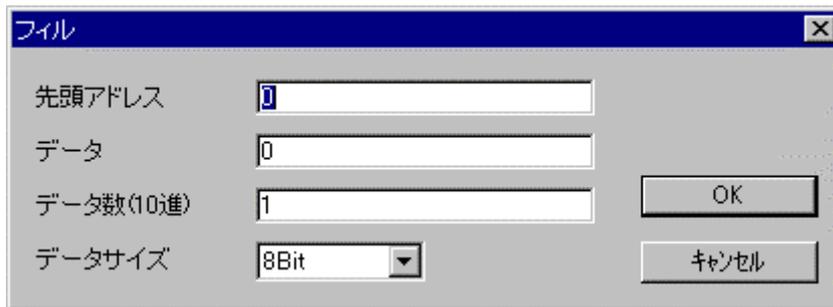
変更(M) ~ フィル(F)...

FILL

機能:

メモリの内容をフィルします。

操作:



The screenshot shows a dialog box titled "フィル" (Fill). It has a standard Windows-style title bar with a close button. The dialog contains the following fields and controls:

- 先頭アドレス** (Start Address): A text input field containing the value "0".
- データ** (Data): A text input field containing the value "0".
- データ数(10進)** (Data Count): A text input field containing the value "1".
- データサイズ** (Data Size): A dropdown menu currently set to "8Bit".
- OK** and **キャンセル** (Cancel) buttons are located on the right side of the dialog.

ダイアログ ボックスで、変更するメモリの先頭アドレス(16 進数)、データ(16 進数)、データ数(10 進数)、データ サイズを指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [FILL](#)」参照。

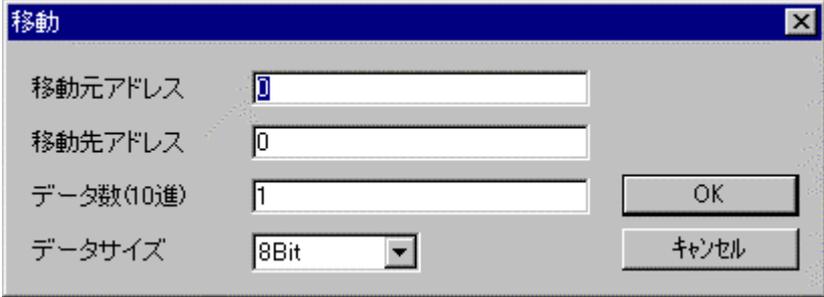
変更(M) ~ 移動(M)...

MOVE

機能:

メモリの内容をブロック転送します。

操作:



移動

移動元アドレス 0

移動先アドレス 0

データ数(10進) 1

データサイズ 8Bit

OK

キャンセル

ダイアログ ボックスで、変更するメモリの移動元アドレス(16 進数)、移動先アドレス(16 進数)、データ数(10 進数)、データ サイズを指定し、「OK」ボタンを押してください。

補足:

- 「第七章 [MOVE](#)」参照。

設定(S) ~ ブレークポイント(B)...

BP/S

機能:

ブレークポイントを設定します

操作:

ブレークポイント設定

設定

ブレークタイプ: H/W命令

データサイズ: ANY

ブレークポイント: 0

アドレスマスク有効: 0

データ比較有効: 0

データマスク有効: 0

ASID有効: 0

登録(B)

終了

有効化

削除(D)

すべて削除(A)

現在の設定

Type	ASID	Address	AddrMask	C/T	R/W	Data	DataMask	Enb

表示選択: すべて

最大H/W命令ブレーク数: 4 最大H/Wアクセスブレーク数: 2

ダイアログボックスで、設定フレームにあるブレークタイプを選択し、データサイズ(16進数)、ブレークポイント(16進数)、アドレスマスク有効(16進数)、データ比較有効(16進数)、データマスク有効(16進数)、ASID有効(16進数)を設定します。「登録」ボタンを押すと、ブレークポイントを設定できます。

設定されたブレークポイントは、現在の設定に表示されます。

表示選択を選択すると、現在の設定の表示が変更されます。

現在の設定の中からひとつのブレークポイントを選択し「無効化」ボタンを押すと、ブレークポイントが無効化できます。同様の手順で「有効化」と「削除」ができます。

「すべて削除」ボタンを押すと、ブレークポイントをすべて削除します。

ブレークポイントの設定が終了したら、「OK」ボタンを押してください。

補足：

- 設定フレーム内の登録は、ブレークタイプを選択すると、それ以下の項目は設定必要な場合のみ入力できるようになっています。
- 登録、無効化、有効化、削除は、直ちにブレークポイントに設定されます。「キャンセル」ボタンを押しても設定のキャンセルはできません。
- 「第七章 [BP/S](#)」参照。
- トレース トリガとして再利用したブレークポイントは削除できません。トレース トリガを無効にしてから削除してください。

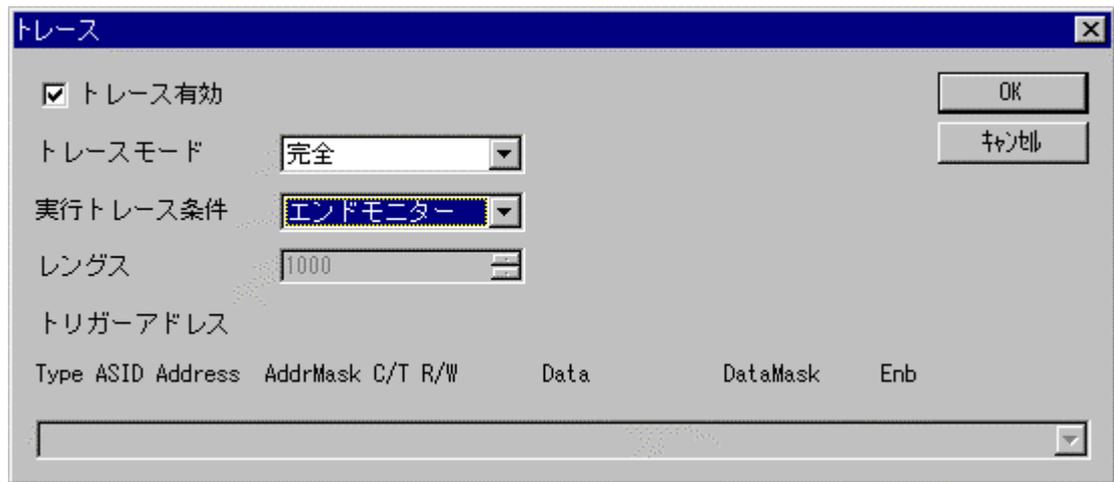
設定(S) ~ トレース(T)...

TRACE

機能:

リアルタイム トレースのトレース モードを表示/設定します。

操作: ダイアログボックスでトレースモードを設定し、「OK」ボタンを押してください。
トレースモードの設定は次のとおりです。



トレース有効 チェックするとトレース全体が有効になります。

トレースモード トレースモードは命令トレースのモードを指定します。トレースモードは以下から選択します。

- ビギン モニター
- エンド モニター
- ビギン トリガー
- エンド トリガー
- ミッド トリガー

第八章 MJXDEBW メニューコマンド

レンジス トレースバッファの長さを指定します。長さの単位は CPU クロックです。「トリガーモード」がビギンモニター／ミッドトリガーのとき設定できます。「レンジス」に設定できる値は 2～524287 です。(両エッジクロックトレースの場合)

トリガーアドレス トリガー アドレス情報を設定します。「トレース モード」がビギン トリガー／ミッド トリガー／エンド トリガーの場合指定できます。

補足:

- 「第七章 [TRACE](#)」参照。

設定(S) ~ 設定の読出(L)...

SETLOAD

機能:

ブレイクポイントの設定とトレースの設定を、ファイルから読み込みます。

操作:



ダイアログ ボックスで、ファイル名を指定し、「開く」ボタンを押してください。

補足:

- 「[設定\(S\) ~ 設定の保存\(S\)...](#)」メニューで作成されたファイルを指定してください。
- 「第七章 [JOURNAL](#)」参照。

設定(S) ~ 設定の保存(S)...

SETSAVE

機能:

ブレイクポイントの設定とトレースの設定を、ファイルへ保存します。

操作:



ダイアログ ボックスで、ファイル名を指定し、「開く」ボタンを押してください。

補足:

- 保存したファイルは「[設定\(S\) ~ 設定の読出\(L\)...](#)」メニューで指定することができます。

設定(S) ~ コンフィグレーション(C) ~ 設定(S)

...

CONFIG

機能:

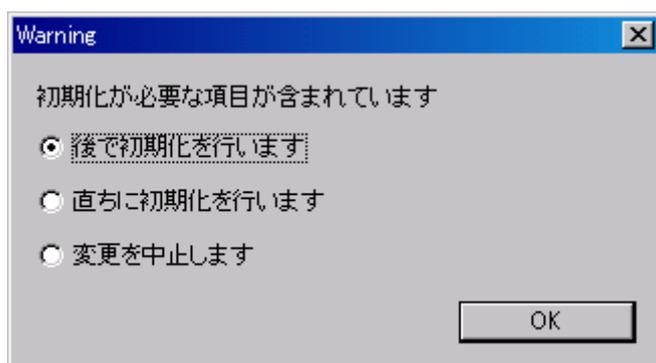
[MJXDEBW](#)の環境を設定します。

操作:



コンフィグレーション設定支援ツールが起動します。

変更する項目を設定してください。



初期化が必要な項目が変更された場合、確認ダイアログが表示されます。
適切な処置を選択し「OK」ボタンを押してください。

補足：

- MJXDEBW 起動中に変更できない項目は反転表示されます。
- 「第七章 [CONFIG](#)」参照。

設定(S) ~ コンフィグレーション(C) ~ 表示(V)

...

CONFIG

機能:

[MJXDEBW](#)の環境を表示します。

操作:



補足:

- 「第七章 [CONFIG](#)」参照。

設定(S) ~ コンフィグレーション ~ 上書き保存(E)

CONFIG

機能:

現在のコンフィグレーションの設定を MJXDEBW 起動時に指定したコンフィグレーションファイルへ上書き保存します。

操作:

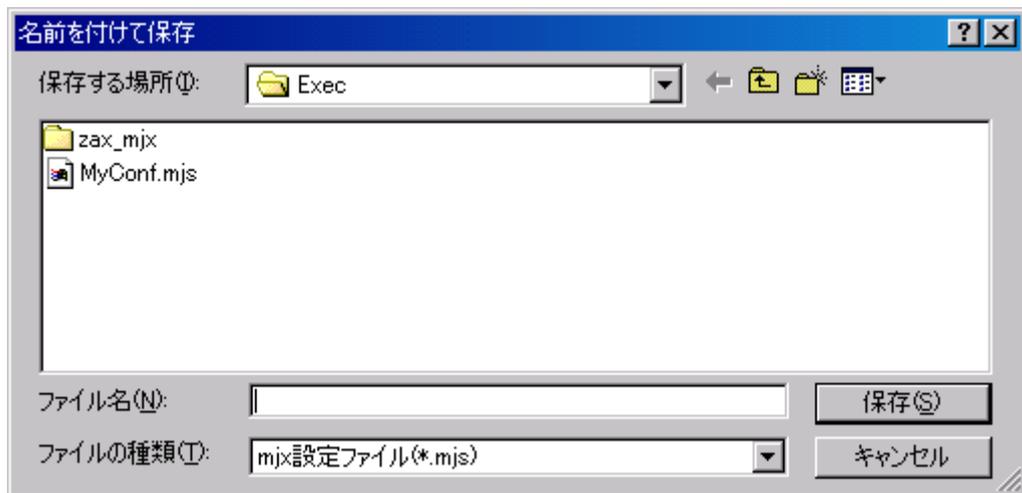
設定(S) ~ コンフィグレーション ~ 名前を付けて保存(A)

CONFIG

機能:

現在のコンフィグレーションの設定を新たなコンフィグレーションファイルへ保存します。

操作:



ダイアログボックスでファイル名を指定し、「保存」ボタンを押してください。

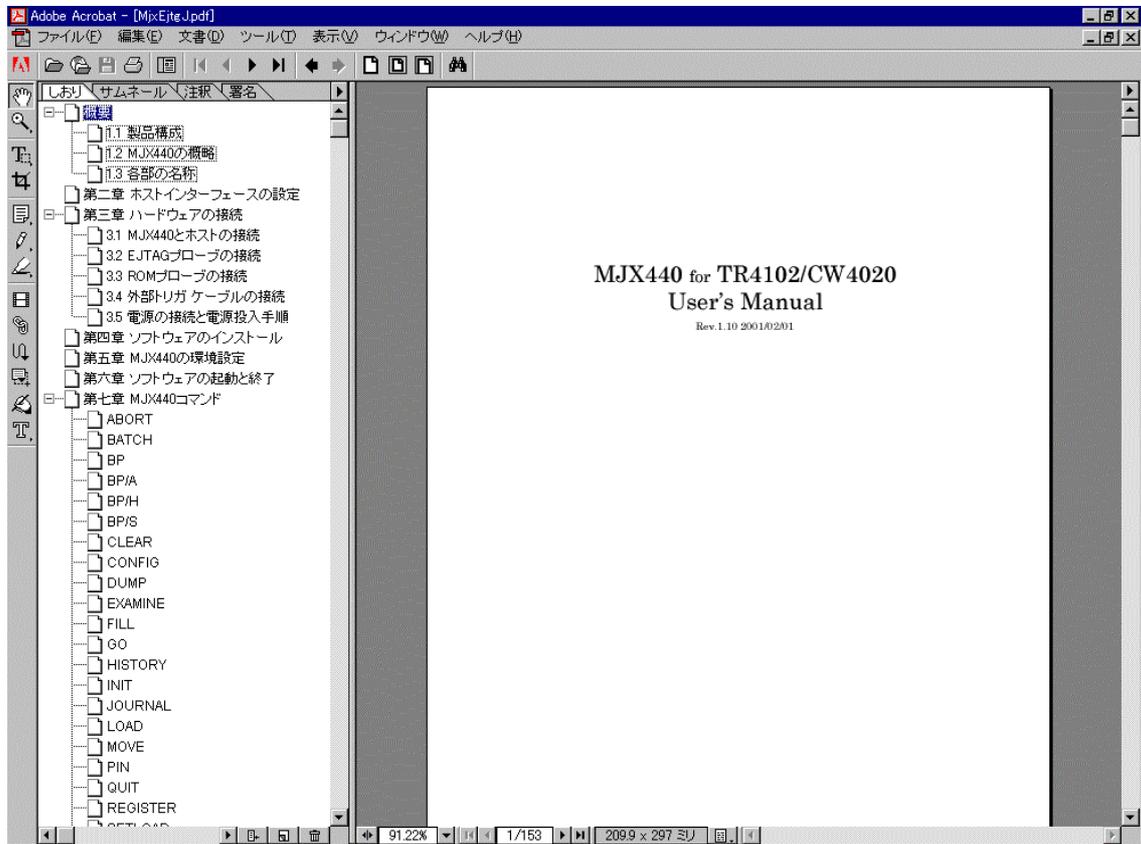
ヘルプ(H) ~ トピックの検索(H)

機能:

MJX440 for TR4102/CW4020 に関するマニュアル画面が表示されます。

操作:

マニュアル画面は「しおり」またはしおりの最後に記述されている「検索項目」のシートから参照してください。



ヘルプ(H) ~ MJXDEBW のバージョン情報(A)

VERSION

機能:

バージョンを表示します

操作:



補足:

- 「第七章 [VERSION](#)」参照。

第九章 高速ダウンロード

高速ダウンロードを行なうための手順について記述しています。

[MJX バイナリ ファイル](#)を作成することによって、プログラムを高速でダウンロードすることができます。およそのダウンロード速度は、次のとおりです。

- プログラムのダウンロードが高速です。
JTAG 接続時 480K バイト/秒 *1 (EJTAG クロック20MHzの場合)
ROM インサーキット接続時 4M バイト/秒

MJX バイナリ ファイルは、ファイル変換プログラム MJXCVT を使用して作成します。MJXCVT は、S レコード ファイルを MJX バイナリ ファイルへ変換するプログラムです。使用方法は、次のとおりです。MS-DOS プロンプト内から実行してください。

MJXCVT の使用方法

```
mjxcvt [-o offset] infile [outfile]
```

-o offset 出力ファイルにオフセット アドレスを加算する

infile 入力ファイル名

outfile 出力ファイル名 (省略時は、infile の拡張子を .mjx にしたファイル名)

MJX バイナリ ファイルのダウンロード

MJXCVT で作成された[MJX バイナリ ファイル](#)は、MJXDEBW コマンドの LOAD コマンドでダウンロードすることができます。

```
load myfile.mjx
```

*1 ダウンロードの速さは JTAG クロックを高く設定する程、高速になります。

付録 A 仕様

本体寸法	43mm(高さ) x 149mm(幅) x 186mm(奥行き)
本体重量	1100g
電源 (AC アダプタ)	入力 AC100~240V 50Hz/60Hz 出力 DC 5V±5% 3.0A
ROM プローブ ケーブル	300mm
外部トリガ ケーブル	300mm
使用温度範囲	0°C~35°C
保存温度範囲	-10°C~55°C
周囲湿度範囲	30%~85%
対応 ROM	「付録 D ROM プローブ」 参照
対応 ROM 個数	8 ビット 1 個、2 個、4 個、8 個 16 ビット 1 個、2 個、4 個
エミュレーション メモリ容量	最大 16M バイト
アクセス タイム	CS から 50n 秒
インターフェース	パラレル (PCI カード、PCMCIA カード) または USB* ¹ 、LAN
ターゲット インターフェース	EJTAG コネクタ ROM ソケット
対応 CPU	TR4102/CW4020(LSI Logic)
ダウンロード速度	480K バイト/秒 (EJTAG)、 4M バイト/秒 (ROM インサーキット接続時)
対応デバッグ	Green Hills MULTI (対応 OS は Windows 98/NT4.0/2000、 Solaris7/8)
ブレークポイント機能	命令フェッチ × 2 ソフトウェア × 128 メモリ アクセス × 4
トレース機能	トレース メモリ 512K ワード トレース クロック 125MHz max (CPU に依存) トレース条件 5 種
制限事項	「付録 B ターゲット システムの制限事項」 参照

*¹ USB I/F は開発中です。

付録 B ターゲット システムの制限事項

MJXDEBW を使用するためには、ターゲット システムが次の条件を満足している必要があります。

- 52 ピンの Extended EJTAG 仕様のコネクタを実装している。(「[付録 C Extended EJTAG コネクタ](#)」参照)
- [外部トリガ ケーブル 1](#) (1 列タイプ)の Vcc 信号を、ターゲット システムの電源(3.3V)に接続できる。

さらに、ROM インサーキット接続するためには、次の条件を満足している必要があります。

- ROM ソケットが実装されている。
- ROM がバンク化されていない。
- ROM が複数個実行されている場合、すべての ROM のアドレス信号が同じか、ROM のアドレスが連続している。

付録 C Extended EJTAG コネクタ

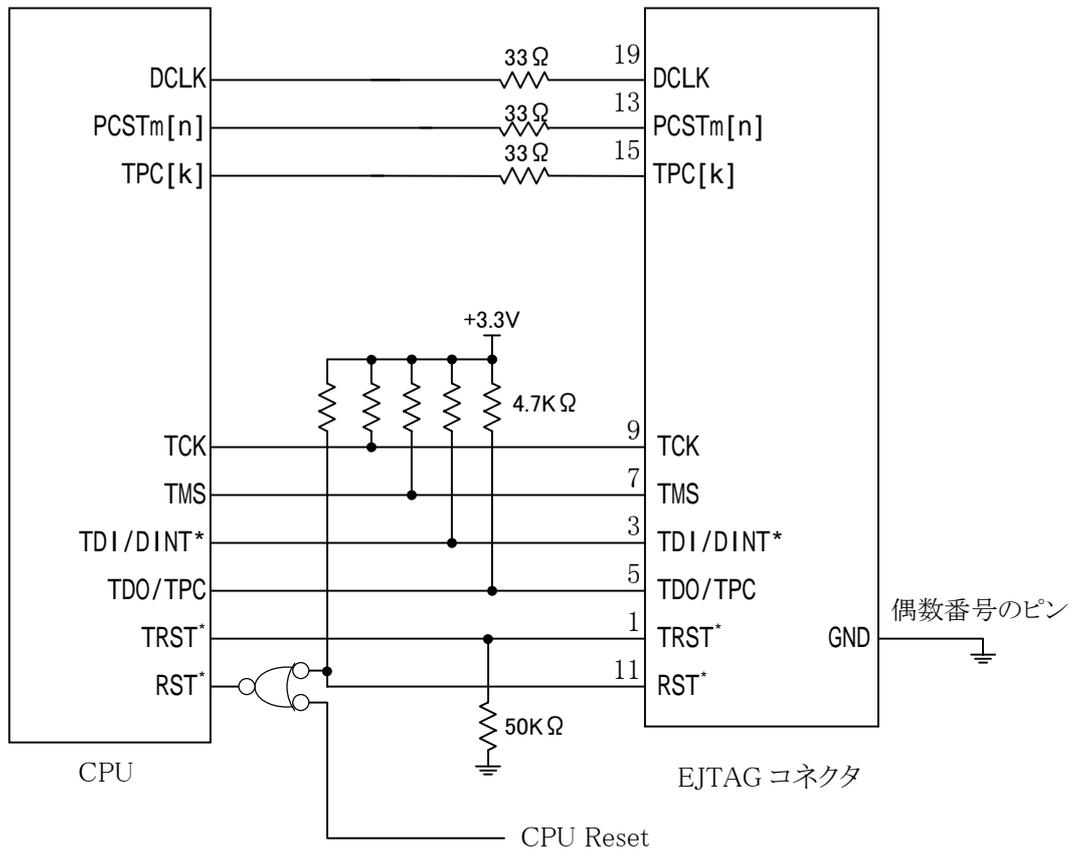
ピンアサイン

TRST*	01	02	GND
TDI/DINT*	03	04	GND
TDO/TPC	05	06	GND
TMS	07	08	GND
TCK	09	10	GND
RST*	11	12	GND
PCST[0]	13	14	GND
PCST[1]	15	16	GND
PCST[2]	17	18	GND
DCLK	19	20	GND
TPC[2]	21	22	GND
PCST2[0]	23	24	GND
PCST2[1]	25	26	GND
PCST2[2]	27	28	GND
TPC[3]	29	30	GND
PCST3[0]	31	32	GND
PCST3[1]	33	34	GND
PCST3[2]	35	36	GND
TPC[4]	37	38	GND
PCST4[0]	39	40	GND
PCST4[1]	41	42	GND
PCST4[2]	43	44	GND
TPC[5]	45	46	GND
TPC[6]	47	48	GND
TPC[7]	49	50	GND
TPC[8]	51	52	GND

推奨コネクタ

- PRECIDIP 社製 852-10-052-10-001 (ストレート)
- PRECIDIP 社製 852-10-052-20-001 (ライトアングル)
- PRECIDIP 社製 852-10-052-30-001 (ストレート、面実装)

ターゲット システムの EJTAG コネクタ推奨回路



注意事項

- CPU～EJTAG コネクタ間のパターンは、できるだけ短くしてください。(50mm 以下)
- TCK～TCK 間と DCLK～DCLK 間のパターンは、GND でシールドしてください。
- 上記推奨回路は、全端子を記入しておりません。未記入の PCST および TPC は推奨回路に準じてください。
- EJTAG コネクタには、電源が接続されておりません。外部トリガ ケーブル 1 の Vcc をターゲット システムの電源 (3.3V) に接続してください。電源検出に使用します。

付録 D ROM プローブ

ROM プローブ基板 J-101A ジャンパ設定:

ROM プラグ	ジャンパ設定	ROM 容量	対応 ROM	ROM メーカー
32pin (27010)	JMP1 1-2 ショート JMP2 2-3 ショート	128K x 8bit (0x20000byte)	HN27C101AG μ PD27C1001AD TC571000D TC571000AD TC57H1000AD M5M27C101K MBM27C1001-nnZ 27010 27C010 Am27C010	Hitachi NEC Toshiba Toshiba Toshiba Mitsubishi Fujitsu intel intel AMD
32pin (27020)	JMP1 1-2 ショート JMP2 1-2 ショート	256K x 8bit (0x40000byte)	μ PD27C2001D M5M27C201K Am27C020	NEC Mitsubishi AMD
32pin (27040)	JMP1 1-2 ショート JMP2 1-2 ショート	512K x 8bit (0x80000byte)	HN27C4001G μ PD27C4001DZ TC574000D TC574000DI M5M27C401K MBM27C4001-nnZ 27040 Am27C040	Hitachi NEC Toshiba Toshiba Mitsubishi Fujitsu intel AMD
32pin (27080)	JMP1 1-2 ショート JMP2 1-2 ショート	1024K x 8bit (0x100000byte)	Am27C080	AMD
32pin (271000)	JMP1 2-3 ショート JMP2 2-3 ショート	128K x 8bit	HN27C301AG μ PD27C1000AD TC571001D TC571001AD TC57H1001AD M5M27C100K MBM27C1000-nnZ	Hitachi NEC Toshiba Toshiba Toshiba Mitsubishi Fujitsu

付録D ROMプローブ

ROMプローブ基板 J-102A ジャンパ設定

ROM プラグ	ジャンパ設定	ROM 容量	対応 ROM	ROM メーカー
40pin (27C4000 16bit)	JMP1 1-2 ショート	256K x 16bit (0x80000byte)	HN27C4000G Am27C400	Hitachi AMD
42pin (27C8000 16bit)	JMP1 2-3 ショート	512K x 16bit (0x100000byte)	μ PD27C8000 Am27C800	NEC AMD
42pin (27C16000 16bit)	JMP1 1-2 ショート	1024K x 16bit (0x200000byte)		

ROMプローブ基板 J-103A ジャンパ設定

ROM プラグ	ジャンパ設定	ROM 容量	対応 ROM	ROM メーカー
40pin (271024)	JMP1 2-3 ショート	64K x 16bit (0x20000byte)	HN27C1024HG μ PD27C1024D μ PD27C1024AD TC57H1024D TC57H1024AD MBM27C1024-nnZ 27210 27C210 Am27C1024	Hitachi NEC NEC Toshiba Toshiba Fujitsu intel intel AMD
40pin (272048)	JMP1 1-2 ショート	128K x 16bit (0x40000byte)	Am27C2048	AMD
40pin (274096)	JMP1 1-2 ショート	256K x 16bit (0x80000byte)	HN27C4096G HN27C4096HG HN27C4096AG HN27C4096AHG TC574096D MBM27C4096-nnZ 27240 Am27C4096	Hitachi Hitachi Hitachi Hitachi Toshiba Fujitsu intel AMD

付録D ROMプローブ

ROMプローブ基板 J-104A ジャンパ設定

ROM プラグ	ジャンパ設定	ROM 容量	対応 ROM	ROM メーカー
40pin (27C4000 8bit)	JMP1 1-2 ショート	512K x 8bit (0x80000byte)	HN27C4000G Am27C400	Hitachi AMD
42pin (27C8000 8bit)	JMP1 2-3 ショート	1024K x 8bit (0x100000byte)	μ PD27C8000 Am27C800	NEC AMD
42pin (27C16000 8bit)	JMP1 1-2 ショート	2048K x 8bit (0x200000byte)		

付録E 対応 ROMピンアサイン

Vpp	1	32	Vcc
A16	2	31	PGM*
A15	3	30	NC
A12	4	29	A14
A7	5	28	A13
A6	6	27	A8
A5	7	26	A9
A4	8	25	A11
A3	9	24	OE*
A2	10	23	A10
A1	11	22	CE*
A0	12	21	I/O7
I/O0	13	20	I/O6
I/O1	14	19	I/O5
I/O2	15	18	I/O4
Vss	16	17	I/O3

J-101A 27010

Vpp	1	32	Vcc
A16	2	31	PGM*
A15	3	30	A17
A12	4	29	A14
A7	5	28	A13
A6	6	27	A8
A5	7	26	A9
A4	8	25	A11
A3	9	24	OE*
A2	10	23	A10
A1	11	22	CE*
A0	12	21	I/O7
I/O0	13	20	I/O6
I/O1	14	19	I/O5
I/O2	15	18	I/O4
Vss	16	17	I/O3

J-101A 27020

Vpp	1	32	Vcc
A16	2	31	A18
A15	3	30	A17
A12	4	29	A14
A7	5	28	A13
A6	6	27	A8
A5	7	26	A9
A4	8	25	A11
A3	9	24	OE*
A2	10	23	A10
A1	11	22	CE*
A0	12	21	I/O7
I/O0	13	20	I/O6
I/O1	14	19	I/O5
I/O2	15	18	I/O4
Vss	16	17	I/O3

J-101A 27040

A19	1	32	Vcc
A16	2	31	A18
A15	3	30	A17
A12	4	29	A14
A7	5	28	A13
A6	6	27	A8
A5	7	26	A9
A4	8	25	A11
A3	9	24	OE*
A2	10	23	A10
A1	11	22	CE*
A0	12	21	I/O7
I/O0	13	20	I/O6
I/O1	14	19	I/O5
I/O2	15	18	I/O4
Vss	16	17	I/O3

J-101A 27080

付録E 対応 ROMピンアサイン

Vpp	1	32	Vcc
OE*	2	31	PGM*
A15	3	30	NC
A12	4	29	A14
A7	5	28	A13
A6	6	27	A8
A5	7	26	A9
A4	8	25	A11
A3	9	24	A16
A2	10	23	A10
A1	11	22	CE*
A0	12	21	I/O7
I/O0	13	20	I/O6
I/O1	14	19	I/O5
I/O2	15	18	I/O4
Vss	16	17	I/O3

J-101A 271000

A17	1	40	A8
A7	2	39	A9
A6	3	38	A10
A5	4	37	A11
A4	5	36	A12
A3	6	35	A13
A2	7	34	A14
A1	8	33	A15
A0	9	32	A16
CE*	10	31	BYTE*/Vpp
Vss	11	30	Vss
OE*	12	29	I/O15/A-1
I/O0	13	28	I/O7
I/O8	14	27	I/O14
I/O1	15	26	I/O6
I/O9	16	25	I/O13
I/O2	17	24	I/O5
I/O10	18	23	I/O12
I/O3	19	22	I/O4
I/O11	20	21	Vcc

J-102A/J-104A
27C4000

A18	1	42	NC
A17	2	41	A8
A7	3	40	A9
A6	4	39	A10
A5	5	38	A11
A4	6	37	A12
A3	7	36	A13
A2	8	35	A14
A1	9	34	A15
A0	10	33	A16
CE*	11	32	BYTE*/Vpp
Vss	12	31	Vss
OE*	13	30	I/O15/A-1
I/O0	14	29	I/O7
I/O8	15	28	I/O14
I/O1	16	27	I/O6
I/O9	17	26	I/O13
I/O2	18	25	I/O5
I/O10	19	24	I/O12
I/O3	20	23	I/O4
I/O11	21	22	Vcc

J-102A/J-104A
27C8000

付録E 対応 ROMピンアサイン

A18	1	42	A19
A17	2	41	A8
A7	3	40	A9
A6	4	39	A10
A5	5	38	A11
A4	6	37	A12
A3	7	36	A13
A2	8	35	A14
A1	9	34	A15
A0	10	33	A16
CE*	11	32	BYTE*/Vpp
Vss	12	31	Vss
OE*	13	30	I/O15/A-1
I/O0	14	29	I/O7
I/O8	15	28	I/O14
I/O1	16	27	I/O6
I/O9	17	26	I/O13
I/O2	18	25	I/O5
I/O10	19	24	I/O12
I/O3	20	23	I/O4
I/O11	21	22	Vcc

J-102A/J-104A
27C16000

Vpp	1	40	Vcc
CE*	2	39	PGM*
I/O15	3	38	NC
I/O14	4	37	A15
I/O13	5	36	A14
I/O12	6	35	A13
I/O11	7	34	A12
I/O10	8	33	A11
I/O9	9	32	A10
I/O8	10	31	A9
Vss	11	30	Vss
I/O7	12	29	A8
I/O6	13	28	A7
I/O5	14	27	A6
I/O4	15	26	A5
I/O3	16	25	A4
I/O2	17	24	A3
I/O1	18	23	A2
I/O0	19	22	A1
OE*	20	21	A0

J-103A 271024

Vpp	1	40	Vcc
CE*	2	39	PGM*
I/O15	3	38	A16
I/O14	4	37	A15
I/O13	5	36	A14
I/O12	6	35	A13
I/O11	7	34	A12
I/O10	8	33	A11
I/O9	9	32	A10
I/O8	10	31	A9
Vss	11	30	Vss
I/O7	12	29	A8
I/O6	13	28	A7
I/O5	14	27	A6
I/O4	15	26	A5
I/O3	16	25	A4
I/O2	17	24	A3
I/O1	18	23	A2
I/O0	19	22	A1
OE*	20	21	A0

J-103A 272048

付録E 対応 ROMピンアサイン

Vpp	1	40	Vcc
CE*	2	39	A17
I/O15	3	38	A16
I/O14	4	37	A15
I/O13	5	36	A14
I/O12	6	35	A13
I/O11	7	34	A12
I/O10	8	33	A11
I/O9	9	32	A10
I/O8	10	31	A9
Vss	11	30	Vss
I/O7	12	29	A8
I/O6	13	28	A7
I/O5	14	27	A6
I/O4	15	26	A5
I/O3	16	25	A4
I/O2	17	24	A3
I/O1	18	23	A2
I/O0	19	22	A1
OE*	20	21	A0

J-103A 274096

付録 F LED

PWR (緑)	電源が供給されているとき点灯
EJTAG (赤)	外部トリガ ケーブル 1 の Vcc 信号が HIGH レベルのとき点灯
EXI1 (赤)	外部トリガ ケーブル 2 の EXTIN1 信号が HIGH レベルのとき点灯
EXI2 (赤)	外部トリガ ケーブル 2 の EXTIN2 信号が HIGH レベルのとき点灯
EXI3 (赤)	外部トリガ ケーブル 2 の EXTIN3 信号が HIGH レベルのとき点灯
ROM1 (赤)	図 3-5-1～3-5-11 を参照
ROM2 (赤)	同上
ROM3 (赤)	同上
ROM4 (赤)	同上

付録 G レジスタ名一覧

(汎用レジスタ)			(浮動小数点レジスタ)		
R0	CP0_0	PC	FGR0	FPR0	FCR0
R1	CP0_1	HI	FGR1	FPR1	FCR31
R2	CP0_2	LO	FGR2	FPR2	
R3	CP0_3		FGR3	FPR3	
R4	CP0_4		FGR4	FPR4	
R5	CP0_5		FGR5	FPR5	
R6	CP0_6		FGR6	FPR6	
R7	CP0_7		FGR7	FPR7	
R8	CP0_8		FGR8	FPR8	
R9	CP0_9		FGR9	FPR9	
R10	CP0_10		FGR10	FPR10	
R11	CP0_11		FGR11	FPR11	
R12	CP0_12		FGR12	FPR12	
R13	CP0_13		FGR13	FPR13	
R14	CP0_14		FGR14	FPR14	
R15	CP0_15		FGR15	FPR15	
R16	CP0_16		FGR16	FPR16	
R17	CP0_17		FGR17	FPR17	
R18	CP0_18		FGR18	FPR18	
R19	CP0_19		FGR19	FPR19	
R20	CP0_20		FGR20	FPR20	
R21	CP0_21		FGR21	FPR21	
R22	CP0_22		FGR22	FPR22	
R23	CP0_23		FGR23	FPR23	
R24	CP0_24		FGR24	FPR24	
R25	CP0_25		FGR25	FPR25	
R26	CP0_26		FGR26	FPR26	
R27	CP0_27		FGR27	FPR27	
R28	CP0_28		FGR28	FPR28	
R29	CP0_29		FGR29	FPR29	
R30	CP0_30		FGR30	FPR30	
R31	CP0_31		FGR31	FPR31	

付録 H MJX バイナリ ファイル

バイナリ ファイル構成

ヘッダ
データ
ヘッダ
データ
⋮
ヘッダ
データ
終了ヘッダ

ヘッダ (16 バイト長)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
'M'	'J'	'1'	00	len1	len2	len3	00	00	00	00	adr1	adr2	adr3	adr4	00

len1: データ バイト長 (MSB)

len2: データ バイト長

len3: データ バイト長 (LSB)

adr1: 論理アドレス (MSB)

adr2: 論理アドレス

adr3: 論理アドレス

adr4: 論理アドレス (LSB)

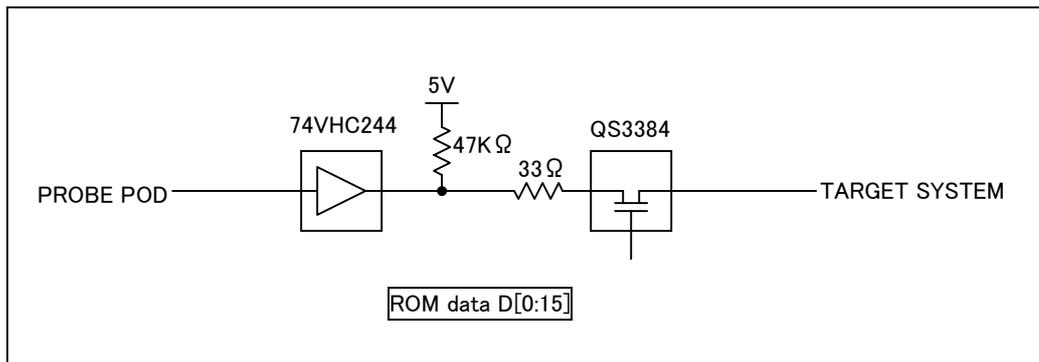
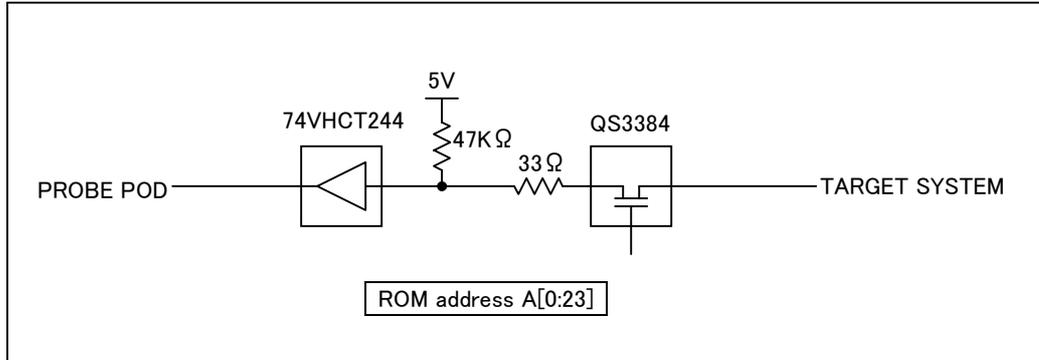
データ (可変、データ バイト長)

XX	⋮	XX													
----	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----

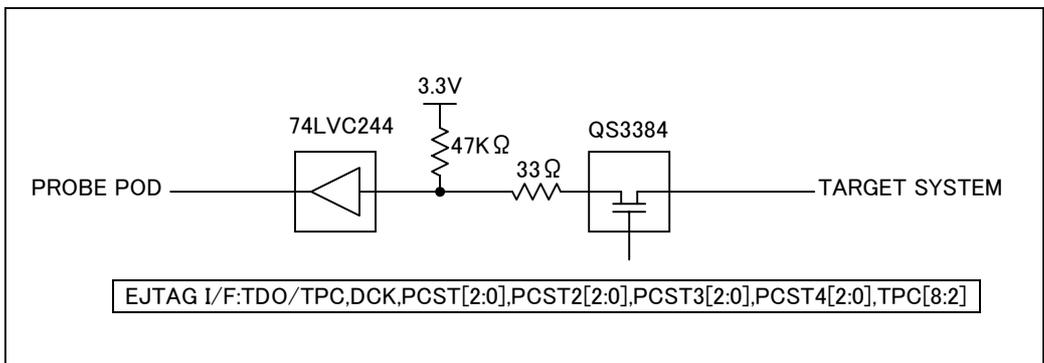
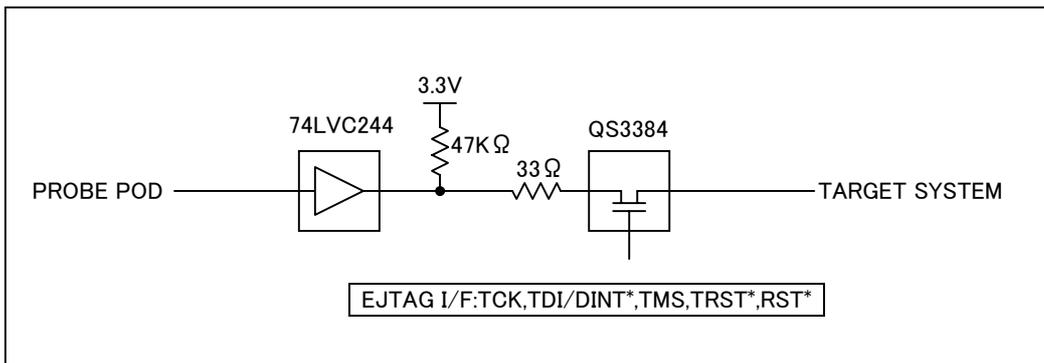
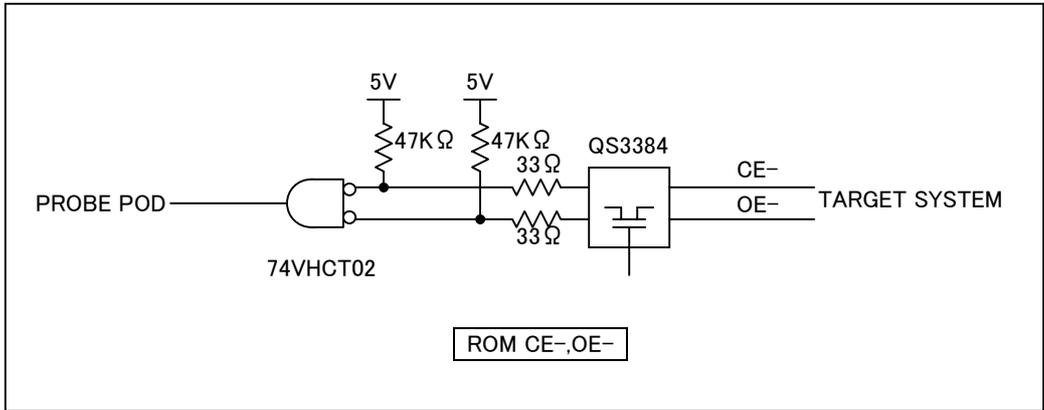
終了ヘッダ (16 バイト長)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
'M'	'J'	'1'	00	00	00	00	00	00	00	00	00	00	00	00	00

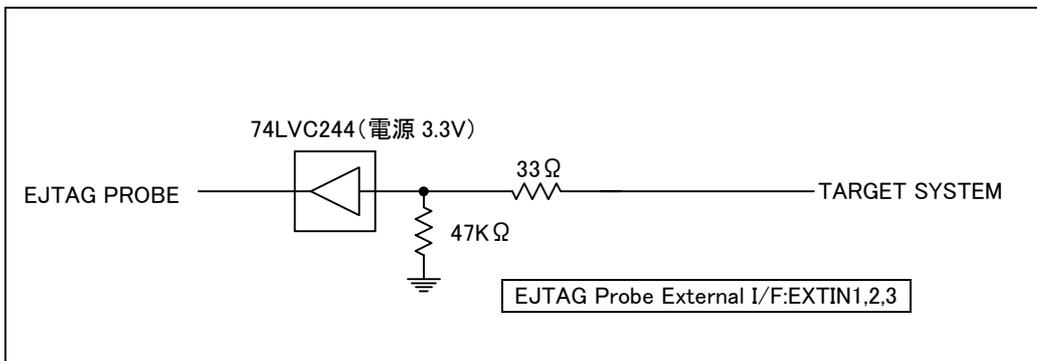
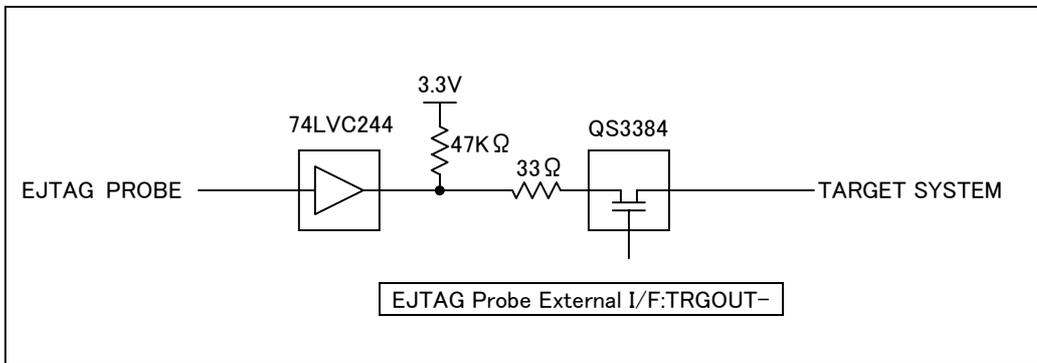
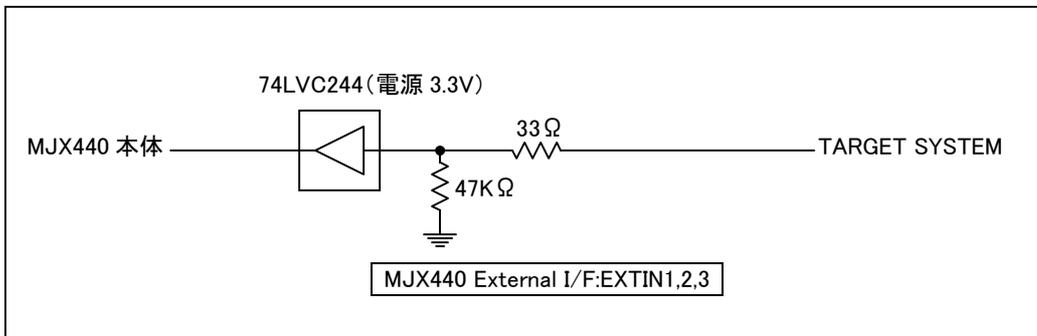
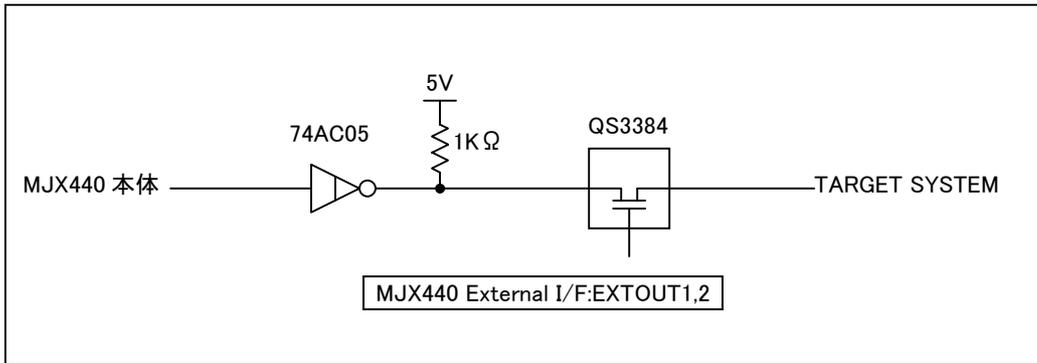
付録I ターゲット システムへのプロービング



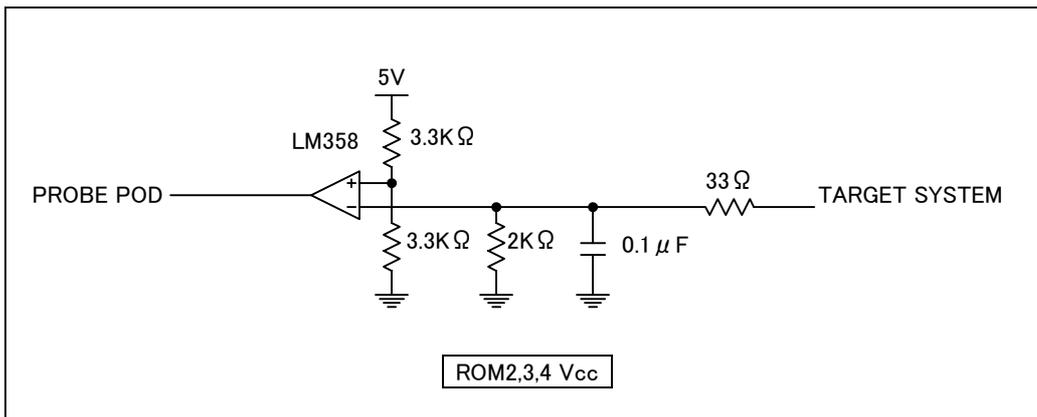
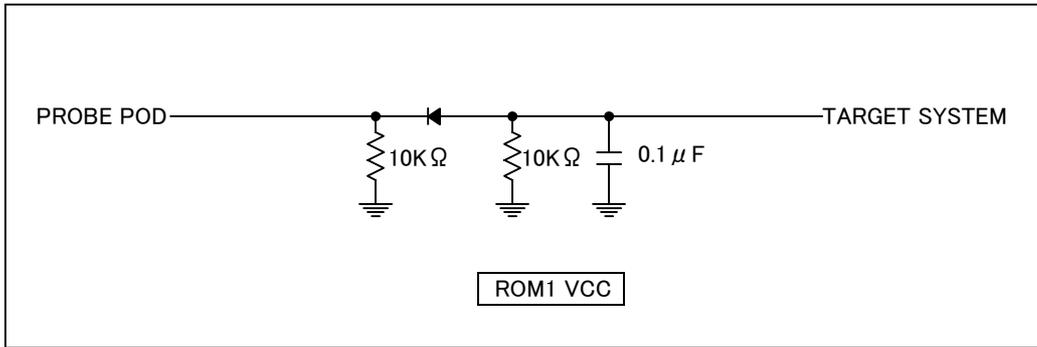
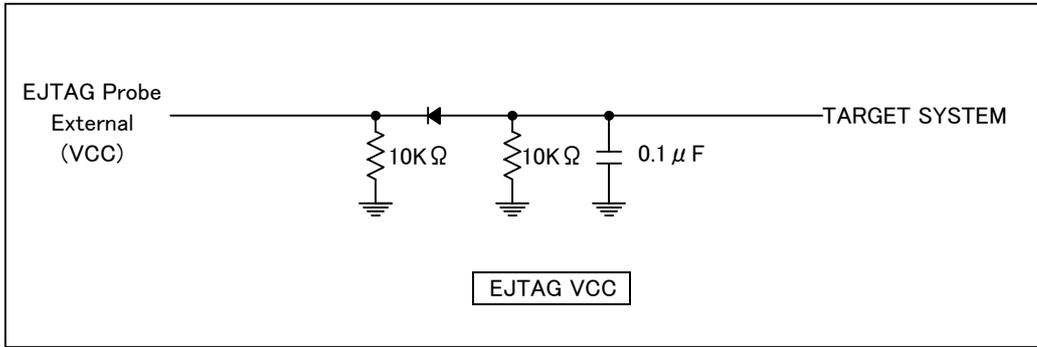
付録I ターゲット システムへのプロービング



付録I ターゲット システムへのプロービング

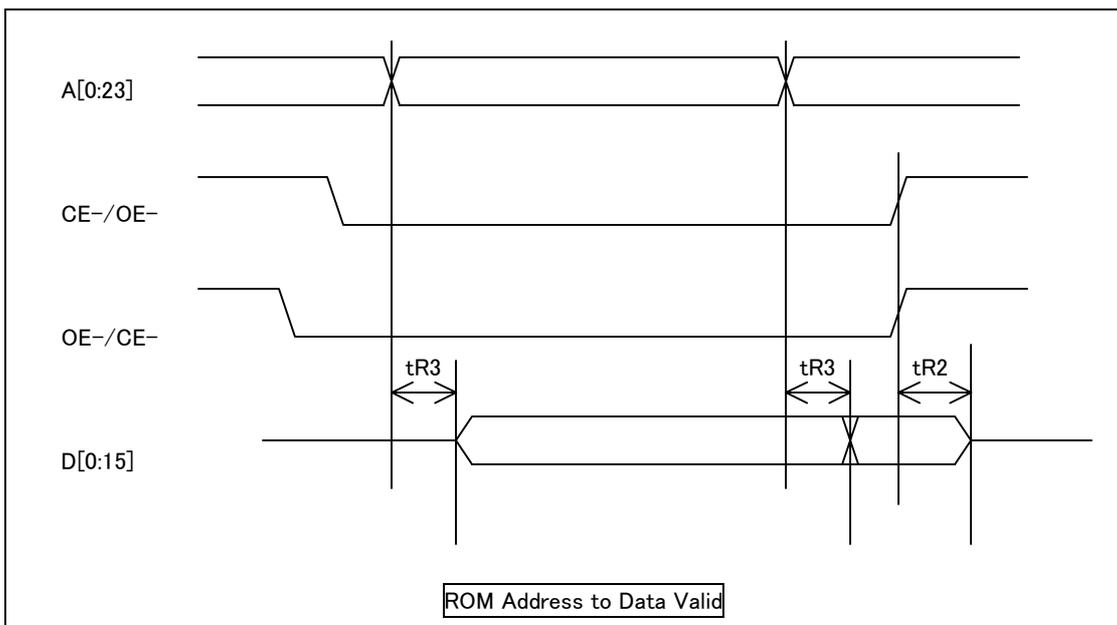
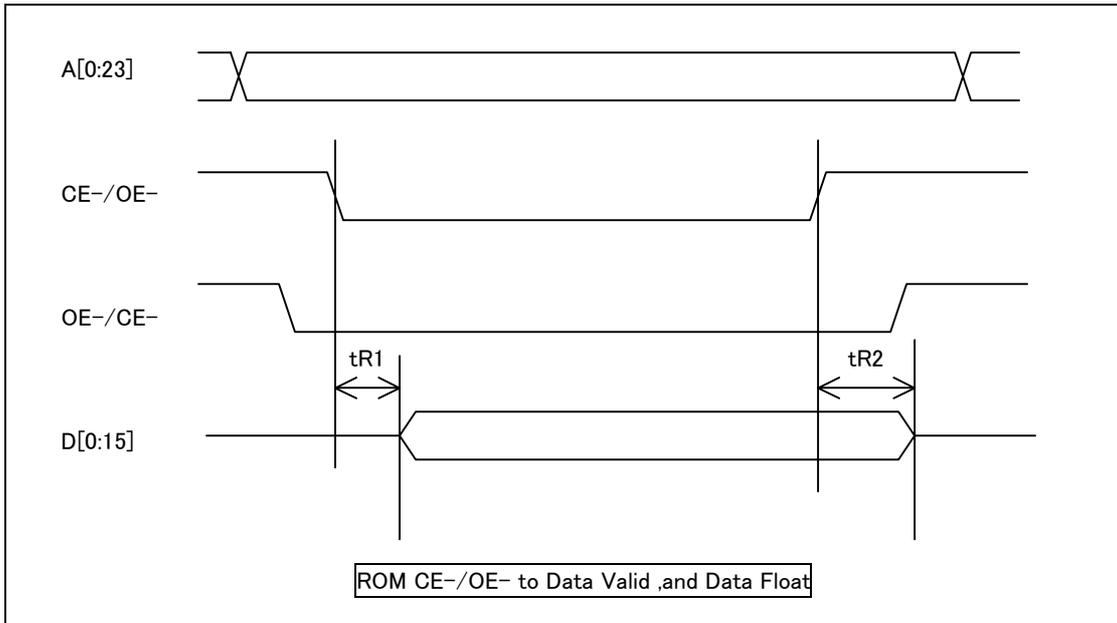


付録I ターゲット システムへのプロービング



付録I ターゲット システムへのプロービング

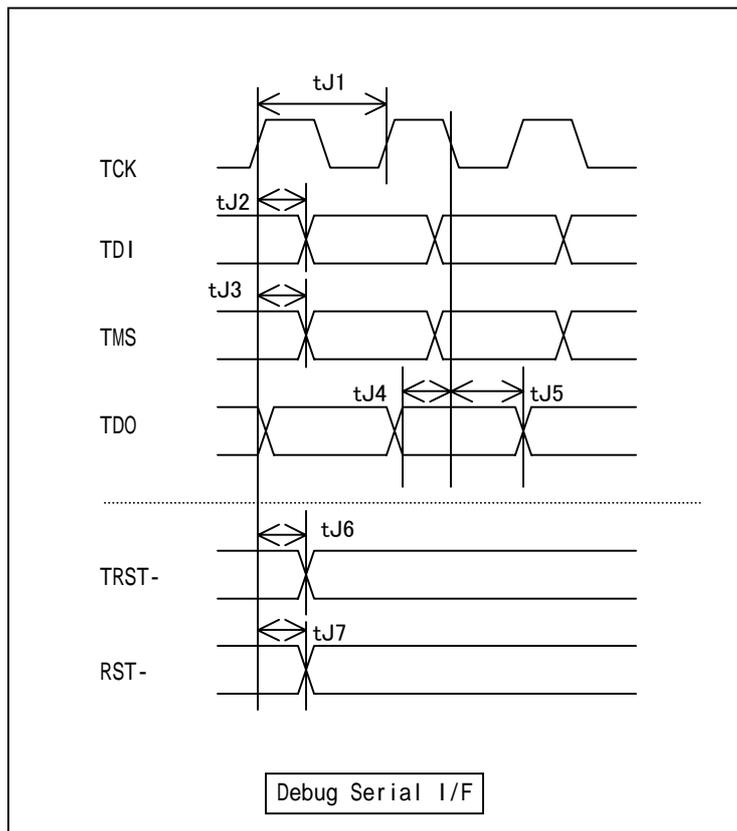
符号	項 目	PROBE POD(TYP)
tR1	CE-/OE- to Data Valid Delay	50nS
tR2	CE-/OE- to Data Float Delay	30nS
tR3	Address to Data Valid Delay	50nS



付録I ターゲット システムへのプロービング

符号	項 目	PROBE		
		MIN	TYP	MAX
tJ1	TCK Clock Period		25	
tJ2	TDI Valid Delay	5	16	
tJ3	TMS Valid Delay	5	12	
tJ4	TDO in Setup Time	20		
tJ5	TDO in Hold Time	0		
tJ6	TRST- Active Delay		5	
tJ7	RST- Active Delay		6	

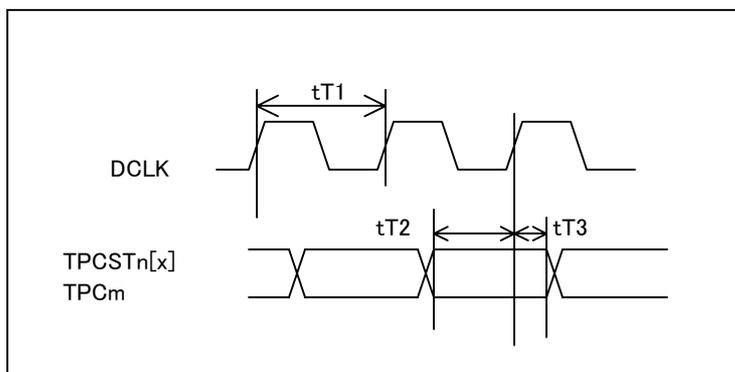
数値は全てターゲット システム上での値



E JTAG (TRACE)

符号	項 目	PROBE		
		MIN	TYP	MAX
tT1	DCLK Clock Period			125MHz
tT2	Trace data in Setup Time	8*		
tT3	Trace data in Hold Time	0		

数値は全てターゲット システム上での値



tT2 は、8~23nsec で、1Clock の遅延を持ってサンプリングします。

付録 J フラッシュ メモリ デバイス情報定義 ファイル レイアウト

{MjxDebw.exe の格納ディレクトリ}¥zax_mjx¥flash¥device の下に、{型式}×{構成}.dat のファイル名で作成します。

ファイル名の例は

{型式}×{構成}.dat ファイル名の例

【型式、構成の例】

型式	構成
AM29DS323DT	2M×16
AM29DS163DB	2M×8 / 1M×16

- 型式 AM29DS323DT、構成 2M×16 の場合
AM29DS323DT には 16bit 構成しかないので
AM29DS323DT.dat
となります。
- 型式 AM29DS163DB、構成 2M×8 / 1M×16 の場合
AM29DS163DB には 8bit 構成と 16bit 構成の 2 タイプがあるので
AM29DS163DBx8.dat
AM29DS163DBx16.dat
の2ファイルになります。

ファイル レイアウトは下記の通りです。一行に複数の項目を記述するときは、項目を空白文字で区切ります。

行	項目	
1	マニファクチャ コード ^(注1)	チップ消去機能無フラグ ^(注2)
2	デバイス コード ^(注3)	
3	アクセス モード ^(注4)	コマンド・インターフェース ^(注5)
4	コマンド入力 第1バス サイクル 書き込み アドレス ^(注6)	
5	コマンド入力 第2バス サイクル 書き込み アドレス ^(注6)	
6	セクタ数 ^(注7)	
7	第1 ~ n セクタ サイズ ^(注8)	セクタ数 ^(注8)
	...	
	...	
-	第i ~ 最後のセクタ サイズ	セクタ数

注1: 16 進数で入力します。

注2: チップ消去機能がないときは 1、あるときは 0 もしくは、空白を入力します。

注3: 16 進数で入力します。

注4: 00: アクセス モードが 8ビット

10: アクセス モードが 8/16ビット選択可で、8ビット アクセス

11: アクセス モードが 8/16ビット選択可で、16ビット アクセス

注5: コマンド・インターフェースの種別を入力します。

0: JEDEC 標準型コマンドと互換 (AMD タイプ)

1: インテル系の CUI (コマンド・ユーザ・インターフェース) を使用 (インテル タイプ)

注6: コマンド(セクタ イレース、チップ イレース、プログラムなど)を入力するときのコマンド入力アドレスを 16 進数で指定します。

インテル系のデバイスへの書き込み・消去では使用しません。0 を入力しておいてください。次ページに、AMD AM29F160DB (2 M x 8-Bit、または 1M x 16-Bit 構成)での例を示します。

コマンド		第 1 バス サイクル		第 2 バス サイクル		第 3 バス サイクル		第 4 バス サイクル		第 5 バス サイクル		第 6 バス サイクル	
		アドレス	データ										
書込み	16 ビット	555	AA	2AA	55	555	A0	PA	PD	—	—	—	—
	8 ビット	AAA		555		AAA				—	—	—	—
チップ 消去	16 ビット	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
	8 ビット	AAA		555		AAA		555		AAA			
セクタ 消去	16 ビット	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30
	8 ビット	AAA		55		AAA		AAA		555			

表の「書込み」シーケンスにしたがって、指定アドレスにデータを入力することで、フラッシュ メモリへデータを書き込みすることができます。

PA、PD は実際に書き込みを行いたいアドレスとデータです。また SA は消去したいセクタのアドレスです。

デバイス情報定義ファイルの「コマンド入力 第1バス サイクル 書き込み アドレス」には表の「第1バス サイクル アドレス(555 または AAA)」を、「コマンド入力 第2バス サイクル 書き込み アドレス」には「第2バス サイクル アドレス(2AA または 555)」を指定します。

16ビット モードの場合には、16ビット アドレスング モードで入力します。

詳しくは、各デバイスのデータ シートを参照してください。

注 7: セクタの総数を 10 進数で入力します。

注 8: セクタ サイズとセクタ数を入力します。セクタ サイズは 16 進数で、セクタ数は 10 進数で指定します。

セクタ数が 1 の場合は、指定を省略できます。

セクタ サイズはバイト単位で指定します。

例えば、サイズ 0x2000 のセクタが 8 個連続していて、その後、0x10000 のセクタが 15 連続している構成の場合は次のように指定します。

2000 8

10000 15

付録J フラッシュ メモリ デバイス情報定義 ファイル レイアウト

以下に AMD AM29F160DB の デバイス情報定義ファイルのサンプルを示します。

2Mx8-Bit 構成: AM29F160DBx8.dat、1Mx16-Bit 構成: AM29F160DBx16.dat。

AM29F160DB x8. dat	AM29F160DB x16. dat	注
0 1	0 1	マニファクチャ コード
d 8	2 2 d 8	デバイス コード
1 0	1 1	アクセス モード
a a a	5 5 5	第 1 バス サイクル 書き込み アドレス
5 5 5	2 a a	第 2 バス サイクル 書き込み アドレス
3 5	3 5	セクタ総数
4 0 0 0	4 0 0 0	第 1 セクタ サイズ
2 0 0 0	2 0 0 0	第 2 セクタ サイズ
2 0 0 0	2 0 0 0	セクタ数を省略 (= 1)
8 0 0 0	8 0 0 0	
1 0 0 0 0 3 1	1 0 0 0 0 3 1	第 5 - 35 セクタ サイズ

チップ消去機能無フラグ、および、コマンド・インターフェースの項は既定値を使用するというで入力していません (0: 機能有、0: JEDEC 互換)。

次に インテル 28F640K3 の デバイス情報定義ファイルのサンプルを示します。

28F640K3.dat (16-Bit 構成のみなので、ファイル名に"x16" はつきません。)

28F640K3. dat	注
8 9 1	マニファクチャ コード、チップ消去機能無フラグ
8 8 0 1	デバイス コード
1 1 1	アクセス モード、コマンド・インターフェース
0	第 1 バス サイクル 書き込み アドレス
0	第 2 バス サイクル 書き込み アドレス
6 4	セクタ総数
2 0 0 0 0 6 4	第 1-64 セクタ サイズ

チップ消去機能無フラグに 1、コマンド・インターフェースの項にも 1 が設定されています。

第 1、2 バス サイクル 書き込み アドレスの項にはいずれも 0 を設定しています。

付録 K 対応フラッシュ メモリー一覧

1. AMD

型式	構成	電源電圧
AM29BDS323DT	2Mx16	1.8V
AM29DS163DB	2Mx8/1Mx16	1.8V
AM29DS163DT	2Mx8/1Mx16	1.8V
AM29DS323DB	4Mx8/2Mx16	1.8V
AM29DS323DT	4Mx8/2Mx16	1.8V
AM29SL800CB	1Mx8/512Kx16	1.8V
AM29SL800CT	1Mx8/512Kx16	1.8V
AM29SL160CB	2Mx8/1Mx16	1.8V
AM29SL160CT	2Mx8/1Mx16	1.8V
AM29DL400BB	512Kx8/256Kx16	3V
AM29DL400BT	512Kx8/256Kx16	3V
AM29DL800BB	1Mx8/512Kx16	3V
AM29DL800BT	1Mx8/512Kx16	3V
AM29DL161DB	2Mx8/1Mx16	3V
AM29DL161DT	2Mx8/1Mx16	3V
AM29DL162DB	2Mx8/1Mx16	3V
AM29DL162DT	2Mx8/1Mx16	3V
AM29DL163DB	2Mx8/1Mx16	3V
AM29DL163DT	2Mx8/1Mx16	3V
AM29DL164DB	2Mx8/1Mx16	3V
AM29DL164DT	2Mx8/1Mx16	3V
AM29DL322DB	4Mx8/2Mx16	3V
AM29DL322DT	4Mx8/2Mx16	3V
AM29DL323DB	4Mx8/2Mx16	3V
AM29DL323DT	4Mx8/2Mx16	3V
AM29DL324DB	4Mx8/2Mx16	3V
AM29DL324DT	4Mx8/2Mx16	3V
AM29BL802CB	512Kx16	3V

続き

型式	構成	電源電圧
AM29BL162CB	1Mx16	3V
AM29PL160CB	2Mx8/1Mx16	3V
AM29LV200BB	256Kx8/128Kx16	3V
AM29LV200BT	256Kx8/128Kx16	3V
AM29LV400BB	512Kx8/256Kx16	3V
AM29LV400BT	512Kx8/256Kx16	3V
AM29LV800BB	1Mx8/512Kx16	3V
AM29LV800BT	1Mx8/512Kx16	3V
AM29LV160BB	2Mx8/1Mx16	3V
AM29LV160BT	2Mx8/1Mx16	3V
AM29LV160DB	2Mx8/1Mx16	3V
AM29LV160DT	2Mx8/1Mx16	3V
AM29LV320DB	4Mx8/2Mx16	3V
AM29LV320DT	4Mx8/2Mx16	3V
AM29LV640D	4Mx16	3V
AM29LV641D	4Mx16	3V
AM29LV001BB	128Kx8	3V
AM29LV001BT	128Kx8	3V
AM29LV002BB	256Kx8	3V
AM29LV002BT	256Kx8	3V
AM29LV004BB	512Kx8	3V
AM29LV004BT	512Kx8	3V
AM29LV008BB	1Mx8	3V
AM29LV008BT	1Mx8	3V
AM29LV116DB	2Mx8	3V
AM29LV116DT	2Mx8	3V
AM29LV010B	128Kx8	3V
AM29LV040B	512Kx8	3V
AM29LV081B	1Mx8	3V
AM29LV017D	2Mx8	3V
AM29LV033C	4Mx8	3V
AM29LV065D	8Mx8	3V
AM29F002BB	256Kx8	5V

続き

型式	構成	電源電圧
AM29F002BT	256Kx8	5V
AM29F002NBB	256Kx8	5V
AM29F002NBT	256Kx8	5V
AM29F004BB	512Kx8	5V
AM29F004BT	512Kx8	5V
AM29F200BB	256Kx8/128Kx16	5V
AM29F200BT	256Kx8/128Kx16	5V
AM29F400BB	512Kx8/256Kx16	5V
AM29F400BT	512Kx8/256Kx16	5V
AM29F800BB	1Mx8/512Kx16	5V
AM29F800BT	1Mx8/512Kx16	5V
AM29F160DB	2Mx8/1Mx16	5V
AM29F160DT	2Mx8/1Mx16	5V
AM29F010B	128Kx8	5V
AM29F040B	512Kx8	5V
AM29F080B	1Mx8	5V
AM29F016D	2Mx8	5V
AM29F017D	2Mx8	5V
AM29F032B	4Mx8	5V

2. ATMEL

型式	構成	電源電圧
AT49F001	128Kx8	5V
AT49F001T	128Kx8	5V
AT49F001N	128Kx8	5V
AT49F001NT	128Kx8	5V
AT49BV001	128Kx8	3V
AT49BV001T	128Kx8	3V
AT49BV001N	128Kx8	3V
AT49BV001NT	128Kx8	3V
AT49LV001	128Kx8	3V
AT49LV001T	128Kx8	3V
AT49LV001N	128Kx8	3V
AT49LV001NT	128Kx8	3V
AT49F002	256Kx8	5V
AT49F002T	256Kx8	5V
AT49F002N	256Kx8	5V
AT49F002NT	256Kx8	5V
AT49BV002	256Kx8	3V
AT49BV002T	256Kx8	3V
AT49BV002N	256Kx8	3V
AT49BV002NT	256Kx8	3V
AT49LV002	256Kx8	3V
AT49LV002T	256Kx8	3V
AT49LV002N	256Kx8	3V
AT49LV002NT	256Kx8	3V
AT49F2048	128Kx16	5V
AT49F2048A	256Kx8/128x16	5V
AT49BV2048	128Kx16	3V
AT49LV2048	128Kx16	3V
AT49BV2048A	256Kx8/128x16	3V
AT49LV2048A	256Kx8/128x16	3V
AT49F4096A	512Kx8/256Kx16	5V
AT49BV4096A	512Kx8/256Kx16	3V
AT49LV4096A	512Kx8/256Kx16	3V

続き

型式	構成	電源電圧
AT49BV4096	256Kx16	3V
AT49LV4096	256Kx16	3V
AT49F4096	256Kx16	5V
AT49F008A	1Mx8	5V
AT49F008AT	1Mx8	5V
AT49F8192A	1Mx8/512Kx16	5V
AT49F8192AT	1Mx8/512Kx16	5V
AT49BV008A	1Mx8	3V
AT49BV008AT	1Mx8	3V
AT49BV8192A	1Mx8/512Kx16	3V
AT49BV8192AT	1Mx8/512Kx16	3V
AT49LV8192A	1Mx8/512Kx16	3V
AT49F8192	512Kx16	5V
AT49F8192T	512Kx16	5V
AT49F8011	1Mx8/512Kx16	5V
AT49F8011T	1Mx8/512Kx16	5V
AT49BV8192	512Kx16	3V
AT49BV8192T	512Kx16	3V
AT49LV8192	512Kx16	3V
AT49LV8192T	512Kx16	3V
AT49BV8011	1Mx8/512Kx16	3V
AT49BV8011T	1Mx8/512Kx16	3V
AT49LV8011	1Mx8/512Kx16	3V
AT49LV8011T	1Mx8/512Kx16	3V
AT49BV160	2Mx8/1Mx16	3V
AT49BV160T	2Mx8/1Mx16	3V
AT49BV161	2Mx8/1Mx16	3V
AT49BV161T	2Mx8/1Mx16	3V
AT49LV160	2Mx8/1Mx16	3V
AT49LV160T	2Mx8/1Mx16	3V
AT49LV161	2Mx8/1Mx16	3V
AT49LV161T	2Mx8/1Mx16	3V

続き

型式	構成	電源電圧
AT49BV1604	2Mx8/1Mx16	3V
AT49BV1604T	2Mx8/1Mx16	3V
AT49BV1614	2Mx8/1Mx16	3V
AT49BV1614T	2Mx8/1Mx16	3V
AT49BV1604A	2Mx8/1Mx16	3V
AT49BV1604AT	2Mx8/1Mx16	3V
AT49BV1614A	2Mx8/1Mx16	3V
AT49BV1614AT	2Mx8/1Mx16	3V
AT49LV1614A	2Mx8/1Mx16	3V
AT49LV1614AT	2Mx8/1Mx16	3V
AT49BV320	4Mx8/2Mx16	3V
AT49BV320T	4Mx8/2Mx16	3V
AT49BV321	4Mx8/2Mx16	3V
AT49BV321T	4Mx8/2Mx16	3V
AT49LV320	4Mx8/2Mx16	3V
AT49LV320T	4Mx8/2Mx16	3V
AT49LV321	4Mx8/2Mx16	3V
AT49LV321T	4Mx8/2Mx16	3V

3. ST

型式	構成	電源電圧
M29F010B	1Mx8	5V
M29F102BB	1Mx16	5V
M29F002BT	2Mx8	5V
M29F002BB	2Mx8	5V
M29F002BNT	2Mx8	5V
M29F200BT	2Mx8/1Mx16	5V
M29F200BB	2Mx8/1Mx16	5V
M29F040B	4Mx8	5V
M29F400BT	4Mx8/2Mx16	5V
M29F400BB	4Mx8/2Mx16	5V
M29F080A	8Mx8	5V
M29F800AT	8Mx8/4Mx16	5V
M29F800AB	8Mx8/4Mx16	5V
M29F016D	16Mx8	5V
M29W010B	1Mx8	3V
M29W102BT	1Mx16	3V
M29W102BB	1Mx16	3V
M29W022BT	2Mx8	3V
M29W022BB	2Mx8	3V
M29W200BT	2Mx8/1Mx16	3V
M29W200BB	2Mx8/1Mx16	3V
M29W004BT	4Mx8	3V
M29W004BB	4Mx8	3V
M29W040B	4Mx8	3V
M29W400BT	4Mx8/2Mx16	3V
M29W400BB	4Mx8/2Mx16	3V
M29W008AT	8Mx8	3V
M29W008AB	8Mx8	3V
M29W800AT	8Mx8/4Mx16	3V
M29W800AB	8Mx8/4Mx16	3V
M29W160DT	16Mx8/8Mx16	3V
M29W160DB	16Mx8/8Mx16	3V

4. 富士通

型式	構成	電源電圧
MBM29F002BC	256Kx8	5V
MBM29F002TC	256Kx8	5V
MBM29F200BC	256Kx8/128Kx16	5V
MBM29F200TC	256Kx8/128Kx16	5V
MBM29F040C	512Kx8	5V
MBM29F004BC	512Kx8	5V
MBM29F004TC	512Kx8	5V
MBM29F400BC	512Kx8/256Kx16	5V
MBM29F400TC	512Kx8/256Kx16	5V
MBM29F080A	1Mx8	5V
MBM29F800BA	1Mx8/512Kx16	5V
MBM29F800TA	1Mx8/512Kx16	5V
MBM29F017A	2Mx8	5V
MBM29F016A	2Mx8	5V
MBM29F160BE	2Mx8/1Mx16	5V
MBM29F160TE	2Mx8/1Mx16	5V
MBM29F033C	4Mx8	5V
MBM29LV002BC	256Kx8	3V
MBM29LV002TC	256Kx8	3V
MBM29LV200BC	256Kx8/128Kx16	3V
MBM29LV200TC	256Kx8/128Kx16	3V
MBM29LV004BC	512Kx8	3V
MBM29LV004TC	512Kx8	3V
MBM29LV400BC	512Kx8/256Kx16	3V
MBM29LV400TC	512Kx8/256Kx16	3V
MBM29DL400BC	512Kx8/256Kx16	3V
MBM29DL400TC	512Kx8/256Kx16	3V
MBM29LV008BA	1Mx8	3V
MBM29LV008TA	1Mx8	3V
MBM29LV080A	1Mx8	3V
MBM29LV800BA	1Mx8/512Kx16	3V
MBM29LV800TA	1Mx8/512Kx16	3V
MBM29DL800BA	1Mx8/512Kx16	3V

続き

型式	構成	電源電圧
MBM29DL800TA	1Mx8/512Kx16	3V
MBM29LV800BE	1Mx8/512Kx16	3V
MBM29LV800TE	1Mx8/512Kx16	3V
MBM29LV016B	2Mx8	3V
MBM29LV016T	2Mx8	3V
MBM29LV017	2Mx8	3V
MBM29LV160B	2Mx8/1Mx16	3V
MBM29LV160T	2Mx8/1Mx16	3V
MBM29PL160BD	2Mx8/1Mx16	3V
MBM29PL160TD	2Mx8/1Mx16	3V
MBM29DL161BD	2Mx8/1Mx16	3V
MBM29DL161TD	2Mx8/1Mx16	3V
MBM29DL162BD	2Mx8/1Mx16	3V

5. INTEL

型式	構成	電源電圧
28F008SA	1Mx8	5V
28F008B3T	1Mx8	3V
28F008B3B	1Mx8	3V
28F800B3T	512Kx16	3V
28F800B3B	512Kx16	3V
28F016B3T	2Mx8	3V
28F016B3B	2Mx8	3V
28F160B3T	1Mx16	3V
28F160B3B	1Mx16	3V
28F004S5	512Kx8	5V
28F008S5	1Mx8	5V
28F016S5	2Mx8	5V
28F016SA	2Mx8	5V
28F004S3	512Kx8	3V
28F008S3	1Mx8	3V
28F016S3	2Mx8	3V
28F004B5T	512Kx8	5V
28F004B5B	512Kx8	5V
28F200B5T	256Kx8/128Kx16	5V
28F200B5B	256Kx8/128Kx16	5V
28F400B5T	512Kx8/256Kx16	5V
28F400B5B	512Kx8/256Kx16	5V
28F800B5T	1Mx8/512Kx16	5V
28F800B5B	1Mx8/512Kx16	5V
28F320J5	4Mx8/2Mx16	5V
28F640J5	8Mx8/4Mx16	5V
28F160S3	2Mx8/1Mx16	3V
28F320S3	4Mx8/2Mx16	3V
28F160S5	2Mx8/1Mx16	5V
28F320S5	4Mx8/2Mx16	5V
28F800F3T	512Kx16	3V
28F800F3B	512Kx16	3V
28F160F3T	1Mx16	3V

続き

型式	構成	電源電圧
28F160F3B	1Mx16	3V
28F800C3T	512Kx16	3V
28F800C3B	512Kx16	3V
28F160C3T	1Mx16	3V
28F160C3B	1Mx16	3V
28F320C3T	2Mx16	3V
28F320C3B	2Mx16	3V
28F640C3T	4Mx16	3V
28F640C3B	4Mx16	3V
28F320J3A	4Mx8/2Mx16	3V
28F640J3A	8Mx8/4Mx16	3V
28F128J3A	16Mx8/8Mx16	3V
28F640K3	4Mx16	3V
28F128K3	8Mx16	3V
28F256K3	16Mx16	3V
28F640K18	4Mx16	1.8V
28F128K18	8Mx16	1.8V
28F256K18	16Mx16	1.8V

続き

型式	構成	電源電圧
28F160F3B	1Mx16	3V
28F800C3T	512Kx16	3V
28F800C3B	512Kx16	3V
28F160C3T	1Mx16	3V
28F160C3B	1Mx16	3V
28F320C3T	2Mx16	3V
28F320C3B	2Mx16	3V
28F640C3T	4Mx16	3V
28F640C3B	4Mx16	3V
28F320J3A	4Mx8/2Mx16	3V
28F640J3A	8Mx8/4Mx16	3V
28F128J3A	16Mx8/8Mx16	3V
28F640K3	4Mx16	3V
28F128K3	8Mx16	3V
28F256K3	16Mx16	3V
28F640K18	4Mx16	1.8V
28F128K18	8Mx16	1.8V
28F256K18	16Mx16	1.8V

6. シャープ

型式	構成	電源電圧
LH28F004BVT-TL85	512Kx8	Smart Voltage
LH28F004BVT-BL85	512Kx8	Smart Voltage
LH28F008SCN-L12	1Mx8	Smart Voltage
LH28F008SCT-L85	1Mx8	Smart Voltage
LH28F008SCHT-L85	1Mx8	Smart Voltage
LH28F008SCHT-L12	1Mx8	Smart Voltage
LH28F008BJT-BTLZ1	1Mx8	3V
LH28F016SCT-L95	2Mx8	Smart Voltage
LH28F016SCN-L12	2Mx8	Smart Voltage
LH28F016SCT-L12	2Mx8	Smart Voltage
LH28F016SCHT-L95	2Mx8	Smart Voltage
LH28F160BJE-TTL90	2Mx8/1Mx16	3V
LH28F160BJHE-TTL90	2Mx8/1Mx16	3V
LH28F160BJE-BTL90	2Mx8/1Mx16	3V
LH28F160BJHE-BTL90	2Mx8/1Mx16	3V
LH28F160BJB-TTL90	2Mx8/1Mx16	3V
LH28F160BJHB-TTL90	2Mx8/1Mx16	3V
LH28F160BJB-BTL90	2Mx8/1Mx16	3V
LH28F160BJHB-BTL90	2Mx8/1Mx16	3V
LH28F160BJHB-BTL90	1Mx16	3V
LH28F160S3NS-L10	2Mx8/1Mx16	3V
LH28F160S3T-L10A	2Mx8/1Mx16	3V
LH28F160S3HT-L10A	2Mx8/1Mx16	3V
LH28F160S3HB-L10A	2Mx8/1Mx16	3V
LH28F160S3B-L10A	2Mx8/1Mx16	3V
LH28F160S3HNS-L10	2Mx8/1Mx16	3V
LH28F320BFE-PTTL70	2Mx16	3V
LH28F320BFHE-PTTL70	2Mx16	3V
LH28F320BFE-PTTL80	2Mx16	3V
LH28F320BFHE-PTTL80	2Mx16	3V
LH28F320BFE-PBTL80	2Mx16	3V
LH28F320BFHE-PBTL80	2Mx16	3V
LH28F320BFHE-PTTLZ1	2Mx16	3V

付録K 対応フラッシュ メモリー一覧

続き

型式	構成	電源電圧
LH28F320BFHE-PBTLZ2	2Mx16	3V
LH28F320BFN-PTTLZH	2Mx16	3V
LH28F320BJE-PTTL90	4Mx8/2Mx16	3V
LH28F320BJHE-PTTL90	4Mx8/2Mx16	3V
LH28F320BJE-PBTL90	4Mx8/2Mx16	3V
LH28F320BJHE-PBTL90	4Mx8/2Mx16	3V
LH28F320BJB-PTTL90	4Mx8/2Mx16	3V
LH28F320BJHB-PTTL90	4Mx8/2Mx16	3V
LH28F320BJB-PBTL90	4Mx8/2Mx16	3V
LH28F320BJHB-PBTL90	4Mx8/2Mx16	3V
LH28F320BJD-TTL80	2Mx16	3V
LH28F400BVE-TL85	512Kx8/256Kx16	Smart Voltage
LH28F400BVE-TL12	512Kx8/256Kx16	Smart Voltage
LH28F400BVHE-TL85	512Kx8/256Kx16	Smart Voltage
LH28F400BVHE-TL12	512Kx8/256Kx16	Smart Voltage
LH28F400BVN-TL85	512Kx8/256Kx16	Smart Voltage
LH28F400BVN-TL12	512Kx8/256Kx16	Smart Voltage
LH28F400BVE-BL85	512Kx8/256Kx16	Smart Voltage
LH28F400BVE-BL12	512Kx8/256Kx16	Smart Voltage
LH28F400BVHE-BL12	512Kx8/256Kx16	Smart Voltage
LH28F400BVN-BL85	512Kx8/256Kx16	Smart Voltage
LH28F640BFE-PTTL80	4Mx16	3V
LH28F640BFHE-PTTL80	4Mx16	3V
LH28F640BFE-PBTL80	4Mx16	3V
LH28F640BFHE-PBTL80	4Mx16	3V
LH28F640BFE-PTTL90	4Mx16	3V
LH28F640BFHE-PTTL90	4Mx16	3V
LH28F640BFE-PBTL90	4Mx16	3V
LH28F640BFHE-PBTL90	4Mx16	3V
LH28F640BFB-PTTL90	4Mx16	3V
LH28F640BFN-PTTLZ2	4Mx16	3V
LH28F800BJE-PTTL90	1Mx8/512Kx16	3V
LH28F800BJE-PTTL10	1Mx8/512Kx16	3V

付録K 対応フラッシュ メモリー一覧

続き

型式	構成	電源電圧
LH28F800BJHE-PTTL90	1Mx8/512Kx16	3V
LH28F800BJE-PBTL90	1Mx8/512Kx16	3V
LH28F800BJHE-PBTL90	1Mx8/512Kx16	3V
LH28F800BJHE-PTTL10	1Mx8/512Kx16	3V
LH28F800BJE-PBTL10	1Mx8/512Kx16	3V
LH28F800BJHE-PBTL10	1Mx8/512Kx16	3V
LH28F800BJB-PTTL90	1Mx8/512Kx16	3V
LH28F800BJB-PTTL10	1Mx8/512Kx16	3V
LH28F800BJHB-PTTL90	1Mx8/512Kx16	3V
LH28F800BJHB-PTTL10	1Mx8/512Kx16	3V
LH28F800BJB-PBTL90	1Mx8/512Kx16	3V
LH28F800BJB-PBTL10	1Mx8/512Kx16	3V
LH28F800BJHB-PBTL90	1Mx8/512Kx16	3V
LH28F800BJHB-PBTL10	1Mx8/512Kx16	3V
LH28F800BJHG-PBTL1FZ5	512Kx16	3V
LH28F800BVE-TTL90	1Mx8/512Kx16	3V
LH28F800BVE-TTL10	1Mx8/512Kx16	3V
LH28F800BVHE-TTL10	1Mx8/512Kx16	3V
LH28F800BVN-TTL90	1Mx8/512Kx16	3V
LH28F800BVN-TTL10	1Mx8/512Kx16	3V
LH28F800BVB-TTL90	1Mx8/512Kx16	3V
LH28F800BVB-TTL10	1Mx8/512Kx16	3V
LH28F800BVHB-TTL10	1Mx8/512Kx16	3V
LH28F800BVE-BTL90	1Mx8/512Kx16	3V
LH28F800BVE-BTL10	1Mx8/512Kx16	3V
LH28F800BVHE-BTL10	1Mx8/512Kx16	3V
LH28F800BVN-BTL90	1Mx8/512Kx16	3V
LH28F800BVN-BTL10	1Mx8/512Kx16	3V
LH28F800BVB-BTL90	1Mx8/512Kx16	3V
LH28F800BVB-BTL10	1Mx8/512Kx16	3V
LH28F800BVE-TV85	1Mx8/512Kx16	5V
LH28F800BVN-BV10	1Mx8/512Kx16	5V
LH28F800BVHB-BV85	1Mx8/512Kx16	5V

付録 L Solaris 版 MJXSERV 使用上の注意

以下に使用する上での注意事項・制限事項をまとめます。

1. Solaris 版 MJXSERV は MULTI のターゲットウィンドウを介して MJXDEBW コマンドをサポートします。
2. 使用可能な接続形態は LAN ボックスを介した LAN 接続のみです。(「第三章 1. [MJX440 とホストの接続](#)」参照。)
3. Solaris7 以上の Sun Sparc Workstation でご使用いただけます。
4. 使用できるのは、第七章で記述されているコマンド インターフェースのみで、第八章のメニュー インターフェースはサポートされません。
5. 環境設定のためのコンフィグ支援ツール MJXCFG は提供されておりません。適当なエディタを使って、コンフィグレーション ファイルを作成してください。(「[第五章 MJX440 の環境設定](#)」参照。)
6. Windows 版では、コンフィグレーション ファイルを指定せずに MJXSERV に接続したとき、ファイル ダイアログで選択する仕様になっていますが、Solaris 版では規定値の `mjxserv.ini` を開きます。(「[第六章 ソフトウェアの起動と終了](#)」参照。)
7. TRACE コマンドその他を実行するとき MULTI の SERVETIMEOUT 変数の既定値では時間切れになる場合があります。適切な値を設定してください。
8. Solaris 版では ABORT、CLEAR、CONFIG、QUIT コマンドはサポートされません。また GO、WAIT コマンドの使い方に変更があります。「[第七章 MJXDEBW コマンド](#)」の該当箇所を参照してください。

検索項目

[MJX440 について](#)

[MJX440 のおもな特長](#)

[最小構成](#)

[ROM インサーキット接続](#)

[外部トリガ ケーブル接続](#)

[MJX440 を使用する前の準備作業](#)

[MULTI について](#)

[MJXDEBW について](#)

[ホストインターフェースの設定](#)

[ハードウェアの接続](#)

[MJX440 とホストの接続](#)

[EJTAG プローブの接続](#)

[ROM プローブの接続](#)

[ROM プローブの接続方法](#)

[外部トリガ ケーブルの接続](#)

[外部トリガ ケーブル 1](#)

[外部トリガ ケーブル 2](#)

[電源の接続と電源投入手順](#)

[MJX440 を操作するソフトウェアのインストール](#)

[MJX440 を操作するソフトウェアのインストール方法](#)

[インストールされるファイル](#)

[ソフトウェアのファイル構成](#)

[MJX440 の環境設定](#)

[コンフィグ支援ツール MJXCFG](#)

[MJX440を操作するソフトウェアの起動と終了](#)

[MULTI を使用する場合](#)

[MJXDEBW を使用する場合](#)

[MJXDEBW コマンド](#)

[MJXDEBW コマンドの一覧](#)

[コマンド名の省略](#)

[MULTI を使用する場合の注意](#)

[次ページ](#)

[MJXDEBW メニューコマンド](#)

[MJXDEBW メニューコマンドの一覧](#)

[高速ダウンロード](#)

[仕様](#)

[本体寸法](#)

[本体重量](#)

[電源 \(AC アダプタ\)](#)

[ROM プローブ ケーブル](#)

[外部トリガ ケーブル](#)

[使用温度範囲](#)

[保存温度範囲](#)

[周囲湿度範囲](#)

[対応 ROM](#)

[対応 ROM 個数](#)

[エミュレーション メモリ容量](#)

[アクセス タイム](#)

[インターフェース](#)

[ターゲット インターフェース](#)

[対応 CPU](#)

[ダウンロード速度](#)

[対応デバッグ](#)

[ブレークポイント機能](#)

[トレース機能](#)

[制限事項](#)

[ターゲット システムの制限事項](#)

[EJTAG コネクタ](#)

[ROM プローブ基板](#)

[ROM プローブ基板 J-101A ジャンパ設定](#)

[ROM プローブ基板 J-102A ジャンパ設定](#)

[ROM プローブ基板 J-103A ジャンパ設定](#)

[ROM プローブ基板 J-104A ジャンパ設定](#)

[前ページ](#)

[次ページ](#)

[対応 ROM ピンアサイン](#)

[J-101A 27010](#)

[J-101A 27020](#)

[J-101A 27040](#)

[J-101A 27080](#)

[J-101A 271000](#)

[J-102A/J-104A/27C4000](#)

[J-102A/J-104A/27C8000](#)

[J-102A/J-104A/27C16000](#)

[J-103A 271024](#)

[J-103A 272048](#)

[J-103A 274096](#)

[LED](#)

[レジスタ名一覧](#)

[MJX バイナリ ファイル](#)

[バイナリ ファイル構成](#)

[ヘッダ](#)

[データ](#)

[終了ヘッダ](#)

[ターゲット システムへのプロービング](#)

[LAN ボックス製品 について](#)

[Solaris 版について](#)

[前ページ](#)

MJX440 について

< 関連項目 >

- [MJX440 のおもな特長](#)
- [最小構成](#)
- [ROM インサーキット接続](#)
- [外部トリガ ケーブル接続](#)
- [MJX440 を使用する前の準備作業](#)

[検索項目の先頭ページ](#)

ハードウェアの接続

< 関連項目 >

- [MJX440 とホストの接続](#)
- [EJTAG プロブの接続](#)
- [ROM プロブの接続](#)
- [外部トリガ ケーブルの接続](#)
- [電源の接続と電源投入手順](#)

[検索項目の先頭ページ](#)

ROM プローブの接続

<関連項目>

- [ROM プローブの接続方法](#)

[検索項目の先頭ページ](#)

外部トリガ ケーブルの接続

<関連項目>

- [外部トリガ ケーブル 1](#)
- [外部トリガ ケーブル 2](#)

[検索項目の先頭ページ](#)

ROM プローブの接続方法

<関連項目>

- [バス幅 8 ビット ROM 1 個、ROM アクセス バス幅 8 ビットの場合](#)
- [バス幅 8 ビット ROM 2 個、ROM アクセス バス幅 8 ビットの場合](#)
- [バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 8 ビットの場合](#)
- [バス幅 8 ビット ROM 2 個、ROM アクセス バス幅 16 ビットの場合](#)
- [バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 16 ビットの場合](#)
- [バス幅 8 ビット ROM 4 個、ROM アクセス バス幅 32 ビットの場合](#)
- [バス幅 16 ビット ROM 1 個、ROM アクセス バス幅 16 ビットの場合](#)
- [バス幅 16 ビット ROM 2 個、ROM アクセス バス幅 16 ビットの場合](#)
- [バス幅 16 ビット ROM 2 個、ROM アクセス バス幅 32 ビットの場合](#)
- [バス幅 16 ビット ROM 4 個、ROM アクセス バス幅 32 ビットの場合](#)
- [バス幅 16 ビット ROM 4 個、ROM アクセス バス幅 64 ビットの場合](#)

[検索項目の先頭ページ](#)

MJX440 を操作するソフトウェアのインストール

< 関連項目 >

- [MJX440 を操作するソフトウェアのインストール方法](#)
- [インストールされるファイル](#)
- [ソフトウェアのファイル構成](#)

[検索項目の先頭ページ](#)

MJX440を操作するソフトウェアの起動と終了

<関連項目>

- [MULTIを使用する場合](#)
- [MJXDEBWを使用する場合](#)

[検索項目の先頭ページ](#)

MJXDEBW コマンド

<関連項目>

- [MJXDEBW コマンドの一覧](#)
- [コマンド名の省略](#)
- [MULTIを使用する場合の注意](#)

[検索項目の先頭ページ](#)

MJXDEBW メニューコマンド

<関連項目>

- [MJXDEBW メニューコマンドの一覧](#)

[検索項目の先頭ページ](#)

ROMプローブ基板

<関連項目>

- [ROMプローブ基板 J-101A ジャンパ設定](#)
- [ROMプローブ基板 J-102A ジャンパ設定](#)
- [ROMプローブ基板 J-103A ジャンパ設定](#)
- [ROMプローブ基板 J-104A ジャンパ設定](#)

[検索項目の先頭ページ](#)

対応 ROM ピンアサイン

< 関連項目 >

• [J-101A 27010](#)

• [J-101A 27020](#)

• [J-101A 27040](#)

• [J-101A 27080](#)

• [J-101A 271000](#)

• [J-102A/J-104A/27C4000](#)

• [J-102A/J-104A/27C8000](#)

• [J-102A/J-104A/27C16000](#)

• [J-103A 271024](#)

• [J-103A 272048](#)

• [J-103A 274096](#)

[検索項目の先頭ページ](#)

MJX バイナリ ファイル

<関連項目>

• [バイナリ ファイル構成](#)

• [ヘッダ](#)

• [データ](#)

• [終了ヘッダ](#)

[検索項目の先頭ページ](#)

Solaris 版について

< 関連項目 >

- [ソフトウェアのインストール](#)
- [MJX440 の環境設定](#)
- [ソフトウェアの起動と終了](#)
- [MJXDEBW コマンド](#)
- [付録 L Solaris 版 MJXSERV 使用上の注意](#)

[検索項目の先頭ページ](#)

LAN ボックス製品 について

< 関連項目 >

- [ハードウェアの接続](#)
- [LAN 接続で MULTI を使用する場合\(Windows\)](#)
- [LAN 接続で MJXDEBW を使用する場合\(Windows\)](#)
- [LAN 接続で MULTI を使用する場合\(Solaris\)](#)

[検索項目の先頭ページ](#)
