

# MJX330 for ARM User's Manual

Rev. 1.03 2004/03/02

変更履歴

バージョン	変更内容	変更日
1.00	● 新規作成	04/02/04
1.01	● 誤字修正	04/02/20
1.02	● 誤字修正	04/02/27
1.03	● 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。

©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 03 月

この度は、MJX330 for ARM(以下、本文中は MJX330)をご購入いただきまして、誠にありがとうございます。

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、MJX330 の概略、各部の名称について記述しています。

## **第二章 デバイス ドライバのインストール**

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**

MJX330 とホストの接続方法、MJX330 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**

MJX330 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 MJX330 の環境設定**

MJX330 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**

MJX330 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**

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。

## もくじ

第一章 概要.....	9
1.1 製品構成.....	9
1.2 MJX330 の概略 .....	10
1.3 各部の名称.....	13
第二章 デバイス ドライバのインストール.....	15
2.1 デバイス ドライバのインストール手順.....	15
2.1.1 Windows98/98SE の場合.....	15
2.1.2 Windows2000 の場合.....	21
2.1.3 WindowsNT4.0 の場合 .....	27
2.1.4 WindowsXP の場合.....	32
2.2 デバイス ドライバのアンインストール手順.....	37
2.2.1 Windows98/98SE の場合.....	37
2.2.2 Windows2000 の場合.....	39
2.2.3 WindowsNT4.0 の場合 .....	39
2.2.4 WindowsXP の場合.....	40
第三章 ハードウェアの接続.....	41
3.1 MJX330 とホストの接続.....	41
3.2 MJX330 とターゲットの接続.....	42
第四章 ソフトウェアのインストール.....	43
第五章 MJX330 の環境設定 .....	50
第六章 ソフトウェアの起動と終了 .....	53
第七章 MJXDEBW コマンド.....	60
ABORT .....	66
BATCH .....	67
BP .....	68
BP/S .....	72
BP/Wn .....	73
BP/WC .....	75
CLEAR .....	76
CONFIG.....	77
DUMP .....	80
EXAMINE.....	81
FILL .....	82
FLASH .....	83

GO.....	85
HISTORY.....	86
INIT.....	88
JOURNAL.....	89
LOAD.....	90
MOVE.....	91
QUIT.....	92
REGISTER.....	93
SETLOAD.....	94
SETSAVE.....	95
STEP.....	96
TRACE.....	97
TRACE/C.....	98
TRACE/D.....	100
TRACE/E.....	102
TRACE/I.....	103
TRACE/P.....	105
TRACE/Q.....	109
TRACE/R.....	110
TRACE/S.....	111
TRACE/T.....	113
UNASM.....	114
VERSION.....	115
WAIT.....	116
<b>第八章 MJXDEBW メニューコマンド.....</b>	<b>117</b>
ファイル(F) ~ バッチ(B).....	118
ファイル(F) ~ ジャーナル開始(S).....	119
ファイル(F) ~ ジャーナル停止(E).....	119
ファイル(F) ~ トレース結果(T).....	120
エミュレーション(E) ~ 実行(G).....	121
エミュレーション(E) ~ 再スタート(R).....	121
エミュレーション(E) ~ 中断(B).....	122
エミュレーション(E) ~ ステップ(S).....	122
エミュレーション(E) ~ N-ステップ(N).....	123
エミュレーション(E) ~ ダウンロード(L).....	124
エミュレーション(E) ~ 初期化(I).....	125
エミュレーション(E) ~ フラッシュメモリ(F) ~ 書込(W).....	126

エミュレーション(E) ~ フラッシュメモリ(F) ~ 消去(E).....	127
表示(V) ~ メモリ(M).....	128
表示(V) ~ レジスタ(R).....	130
表示(V) ~ 逆アセンブル(D).....	131
表示(V) ~ トレース結果(T).....	132
表示(V) ~ コマンド応答クリア(C).....	133
表示(V) ~ オプション(O) ~ フォント(F).....	134
表示(V) ~ オプション(O) ~ 表示(V).....	135
変更(M) ~ メモリ(M).....	136
変更(M) ~ レジスタ(R).....	137
変更(M) ~ アセンブラ(A).....	138
変更(M) ~ フィル(F).....	139
変更(M) ~ 移動(M).....	140
設定(S) ~ ブレークポイント(B).....	141
設定(S) ~ ブレークポイント(B)... ウォッチポイント.....	143
設定(S) ~ ブレークポイント(B)... ウォッチポイント連携.....	144
設定(S) ~ トレース(T).....	145
設定(S) ~ 設定の読出(L).....	159
設定(S) ~ 設定の保存(S).....	160
設定(S) ~ コンフィグレーション(C) ~ 設定(S).....	161
設定(S) ~ コンフィグレーション ~ 表示(V).....	163
設定(S) ~ コンフィグレーション ~ 上書き保存(E).....	164
設定(S) ~ コンフィグレーション ~ 名前を付けて保存(A).....	165
ヘルプ(H) ~ トピックの検索(H).....	166
ヘルプ(H) ~ MJXDEBW のバージョン情報(A).....	167
第九章 高速ダウンロード.....	168
付録 A 仕様.....	169
付録 B ターゲット システムの制限事項.....	170
付録 C JTAG コネクタ.....	171
付録 D レジスタ名一覧.....	173
付録 E MJX バイナリ ファイル.....	174
付録 F ターゲット システムへのプロービング.....	175
付録 G 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.....	177
付録 H 対応フラッシュ メモリー一覧.....	181
付録 I MjxRDI サーバーの詳細.....	195
I-1 Semihosting 機能.....	195
I-2 ハードウェア ブレークポイント リソース 割り当て規則.....	197

付録 J デバイス ドライバのトラブルシューティング .....	198
付録 K ARM ステートと Thumb ステートの指定方法 .....	204
検索項目 .....	205



# 第一章 概要

製品構成、MJX330 の概略、各部の名称について記述しています。

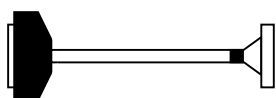
## 1.1 製品構成

出荷時の MJX330 for ARM 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。

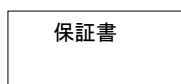
- MJX330 for ARM 本体



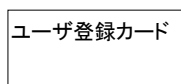
- プローブケーブル



- 保証書

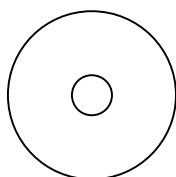


- ユーザー登録カード



**【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



## 1.2 MJX330 の概略

MJX330 for ARM は、JTAG コネクタを備えた ARM プロセッサのターゲット システムをデバッグするための、開発支援装置です。

### MJX330 のおもな特長

- ターゲット システムの [JTAG コネクタ](#) を使用するため、接続が簡単です。
- PC カード Type II で実現されているため、フィールドでの使用に適しています。
- 高速な CPU でも安定して動作します。
- 高級言語デバッガ [MULTI](#) が使用できます。
- MJXDEBW デバッガ上で [MJXDEBW コマンド](#) が使用できます。MJX330 が持つ機能をフルに使用できます。
- ハードウェアブレークポイント機能があります。
- ダウンロード速度 400K バイト/秒 \*1 (JTAG クロック20MHzの場合)。

### 構成

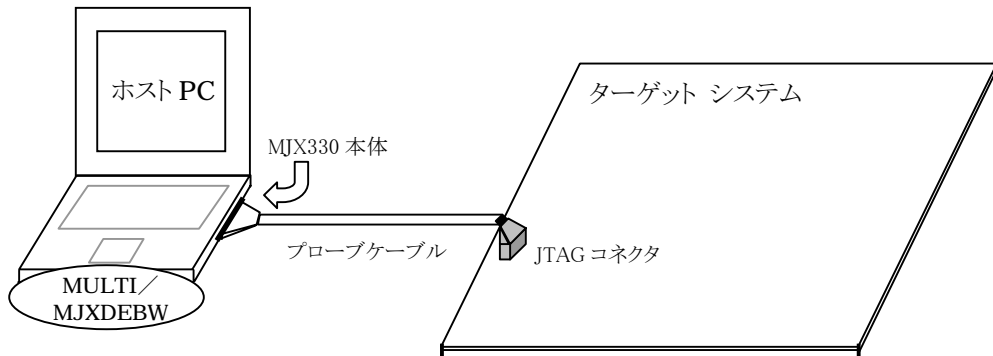


図 1-1 構成

\*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。

## 第一章 概要

MJX330 は図 1-1 の構成で動作します。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。プログラムはプローブケーブルを通して、ターゲット システム上の RAM へダウンロードされます。

MJX330 は、ROM 上のプログラムを実行することはできませんが、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。

### **MJX330 を使用する前の準備作業**

MJX330 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ってください。「ハードウェアの接続」以外は、購入時に一度だけ行えばよい作業です。

- [デバイスドライバのインストール](#)
- [ハードウェアの接続](#)
- [ソフトウェアのインストール](#)
- [MJX330 の環境設定](#)

準備作業が終了したら、[第六章](#)を参照して、MJX330 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、MJX330 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

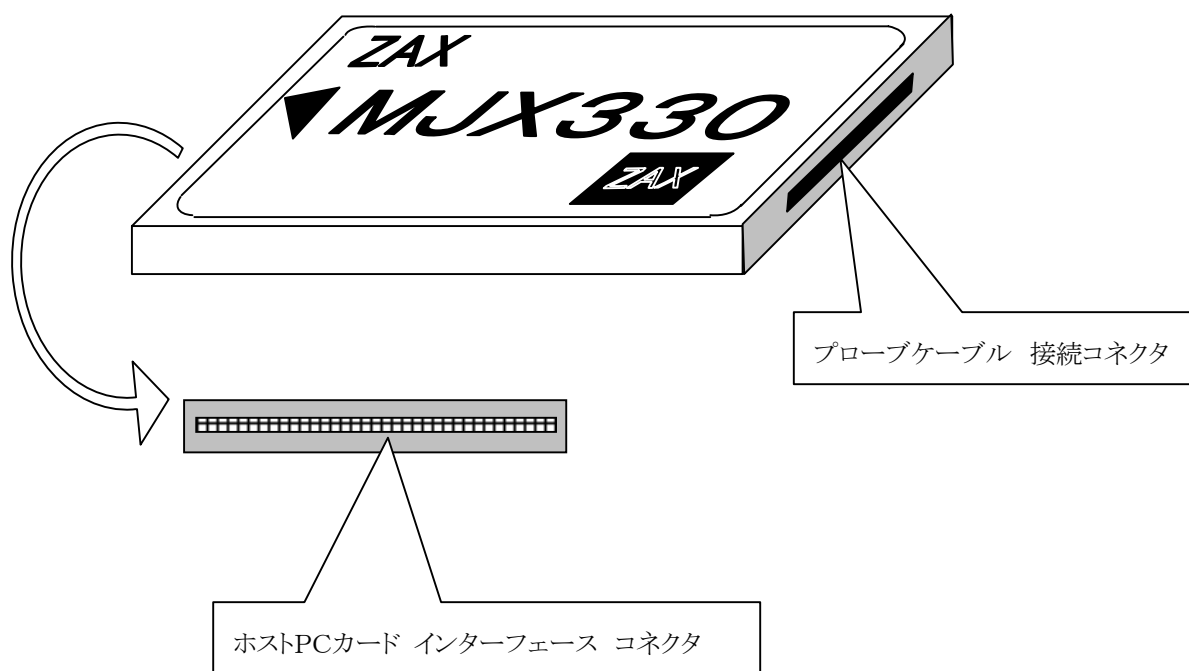
MULTI はいろいろな環境で実行することができる高級言語デバッガです。サーバープログラム MJXSERV を呼び出すことによって、MJX の環境で実行させることができますようになります。

### 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。

## 1.3 各部の名称

### MJX330 本体



ホストPCカード インターフェース コネクタ	ホストPCの PC カードスロットに挿入します。
プローブケーブル 接続コネクタ	プローブケーブルを接続します。

プローブケーブル



プローブケーブル

MJX330 と JTAG を接続するケーブル

JTAG コネクタ

ターゲット システムの JTAG コネクタへ接続するコネクタ

ホストPC側コネクタ

MJX330 へ接続するコネクタ

## 第二章 デバイス ドライバのインストール<sup>\*1</sup>

【重要】 MJX330 はスタンバイには対応していません。ノートPCの消費電力設定をオフにしてご使用ください。

### 2.1 デバイス ドライバのインストール手順

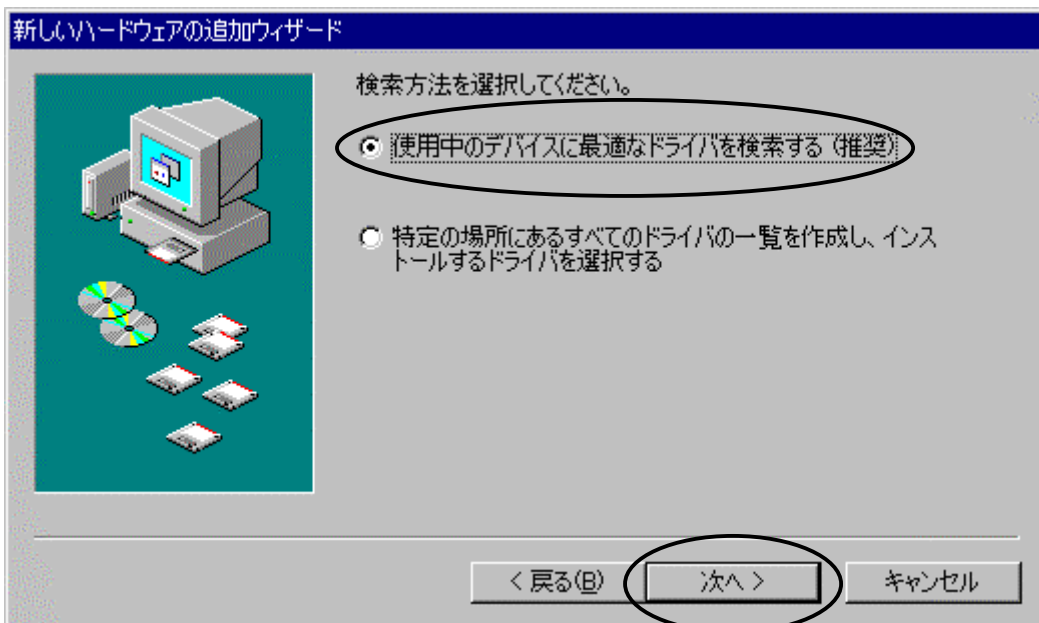
#### 2.1.1 Windows98/98SE の場合

ノートPCの電源オンの状態でPCカードを差すと、新しいハードウェアの追加ウィザードによりカードが検出され以下のダイアログが表示されますので、「次へ>」をクリックします。



\*1 すでに PCMCIA I/F カードにより MJX440 をご使用のお客様はこの操作は不要です。

「使用中のデバイスに最適なドライバを検出する(推奨)」を選択し、「次へ>」をクリックします。



CD-ROMドライブに「MJX for ARM7/9 Tools Disk」のCD-ROMをセットします。

「検索場所の指定(L)」を選択し、ドライバのあるディレクトリ D:¥DRIVER¥WIN9X (CD-ROMドライブがDドライブの場合)を指定します。

「次へ>」をクリックします。





## 第二章 デバイス ドライバのインストール

ドライバ ファイルのあるディレクトリが正しい場合は、以下のダイアログが表示されますので、「次へ」をクリックします。

「このデバイス用のドライバが見つかりませんでした。」と表示された場合は、「戻る(B)」をクリックし、ドライバ ファイルのある正しいディレクトリを指定します。



自動的にドライバがインストールされ、以下のダイアログが表示されます。

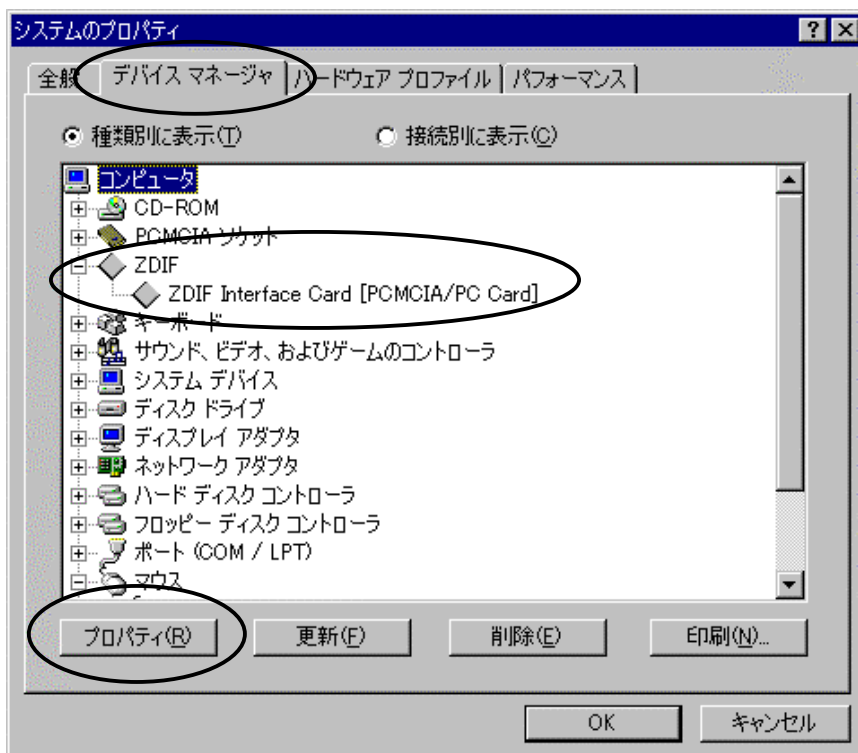
正しくインストールされた場合は「ピポッ」という音がしますので「完了」をクリックします。

「ブー」という音がした場合は正しくインストールされていないので、インストールを終了させてから、「[付録J デバイス ドライバのトラブルシューティング](#)」を参考にしてトラブルシュートを行なってください。



インストールが正しく行われたことを確認するために、コントロール パネルから「システム」をダブル クリックして開き、「デバイス マネージャ」タブをクリックします。

次に「ZDIF」クラスをクリックして、その下にある「ZDIF Interface Card [PCMCIA/PC Card]」のプロパティを表示します(ダブル クリックするか、「プロパティ(R)」をクリック)。



「リソース」タブをクリックし、「競合するデバイス」が「競合はありません。」になっていることを確認します。



## 第二章 デバイス ドライバのインストール

また、「I/O の範囲」が以下のいずれかになっていることを確認します。設定値は、動作環境によって異なります。

0220 – 023F

0260 – 027F

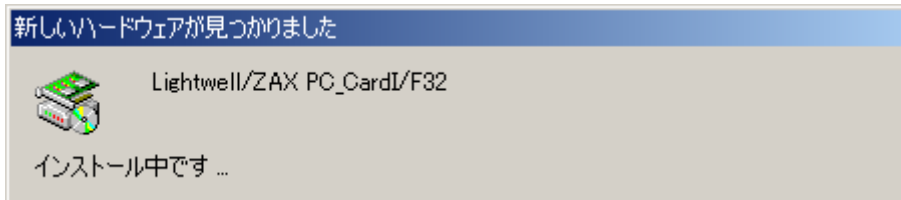
02E0 – 02FF

0320 – 033F

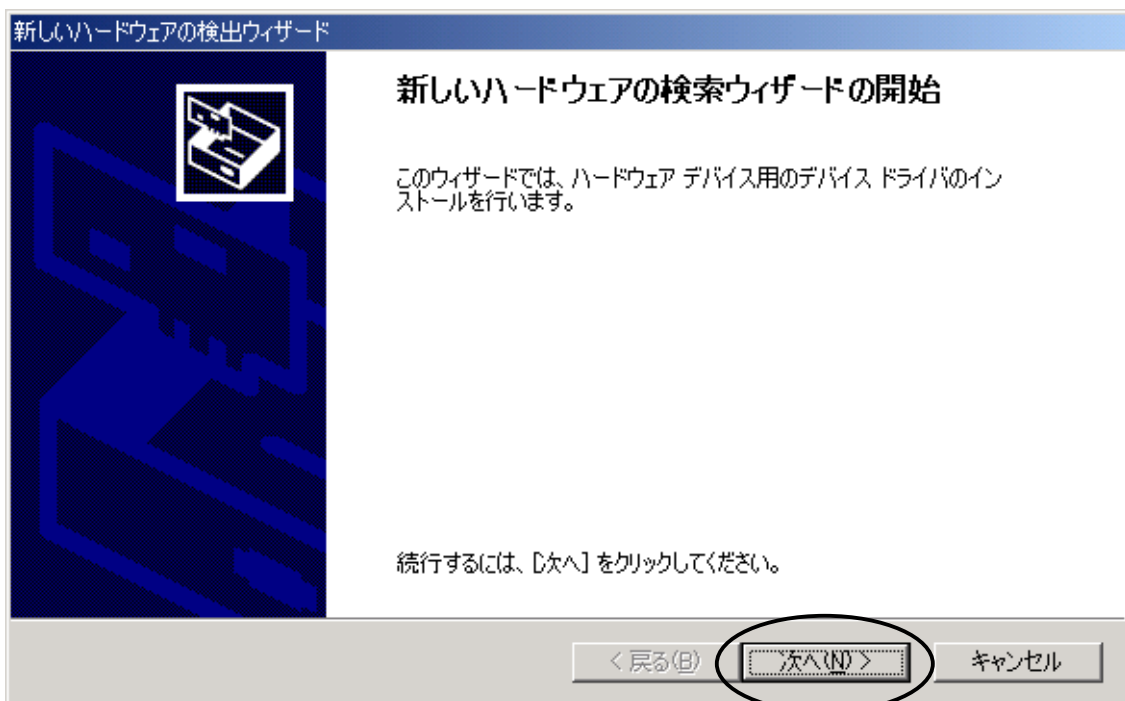
03E0 – 03FF

## 2.1.2 Windows2000 の場合

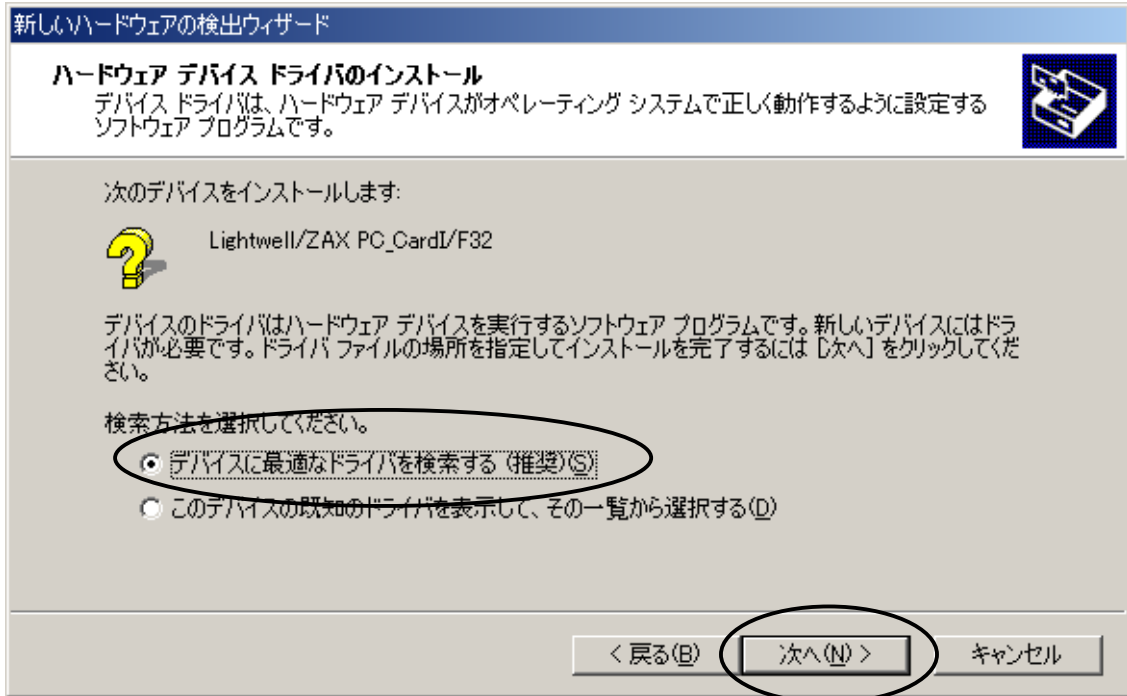
ノート PC の電源オンの状態で PC カードを差すと、ハードウェア ウィザードが起動します。



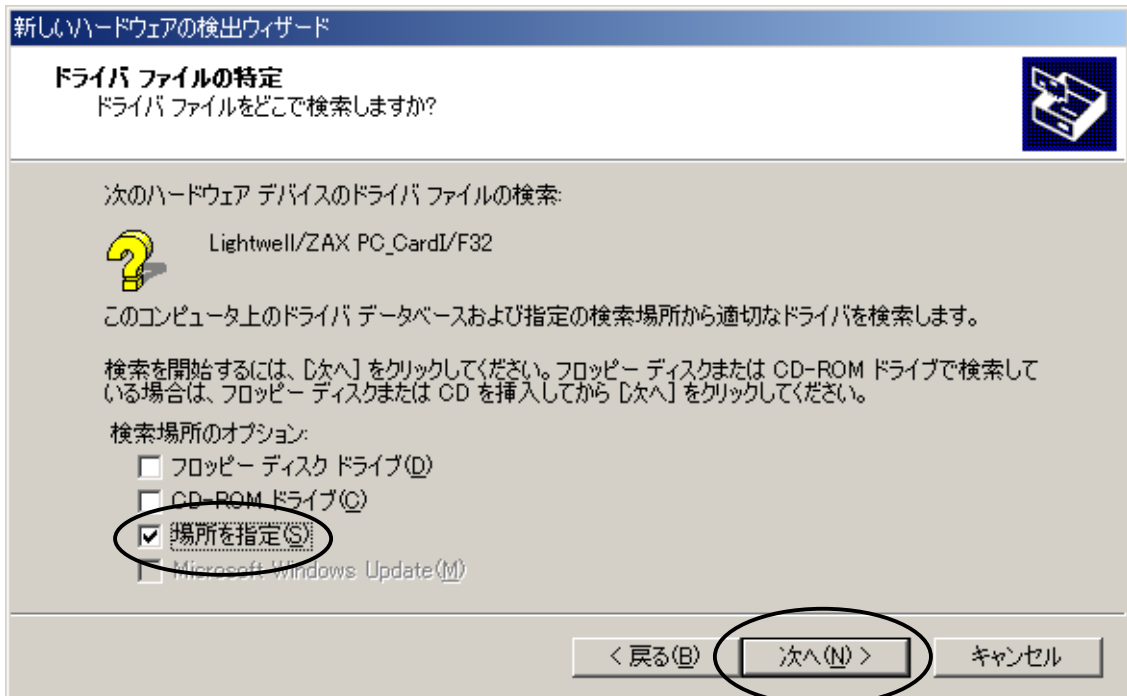
次のウィンドウが表示されたら、「次へ」をクリックしてください。



次のウィンドウが表示されたら、「デバイスに最適なドライバを検索する」を選択し、「次へ」をクリックしてください。

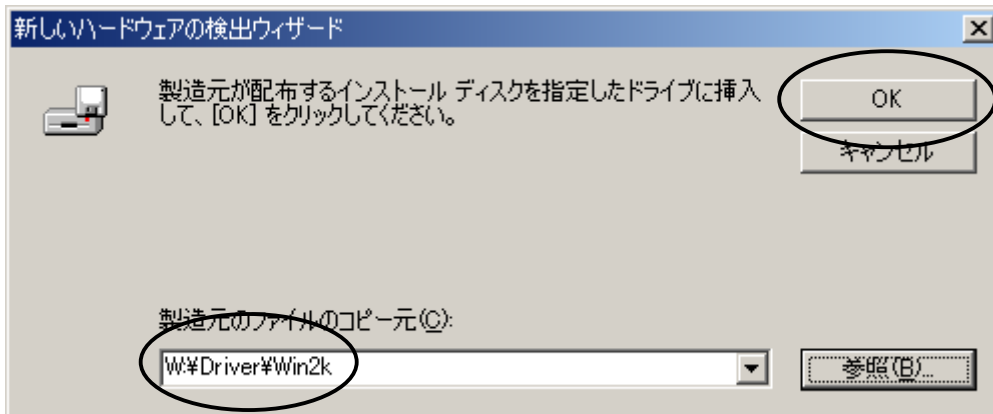


次のウィンドウが表示されたら、「場所を指定」を選択し、「次へ」をクリックしてください。

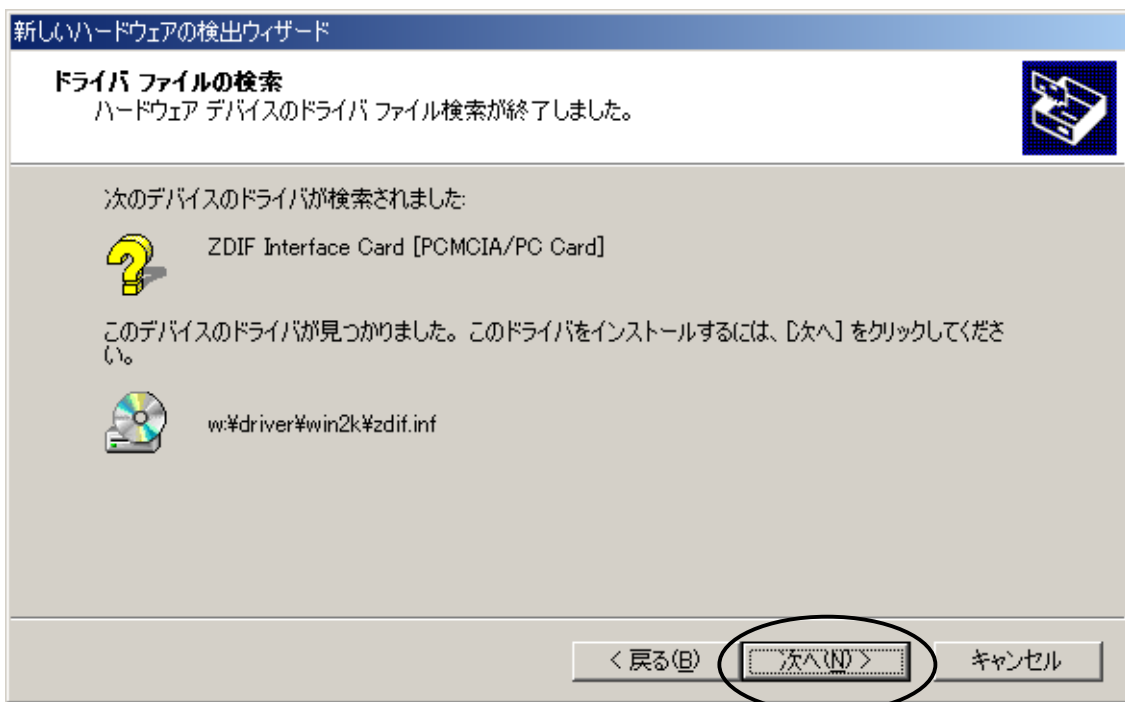


## 第二章 デバイス ドライバのインストール

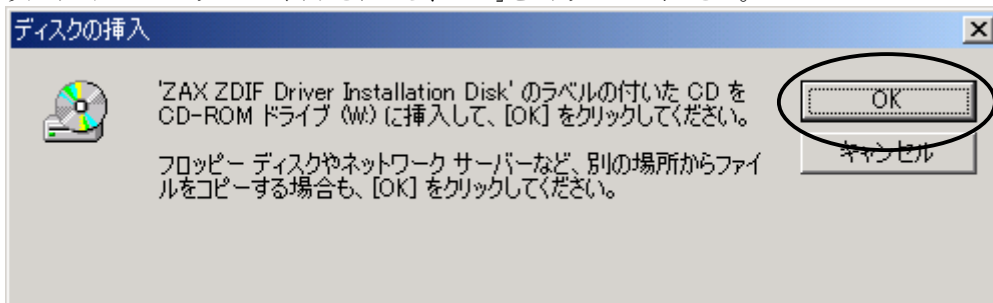
次のウィンドウが表示されたら、「M/JX for ARM7/9 Tools Disk」CD-ROM をドライブにセットしてください。次に、「検索場所の指定」を選択し、W:¥Driver¥Win2k (CD-ROMドライブが W の場合)と入力し、「OK」をクリックしてください。



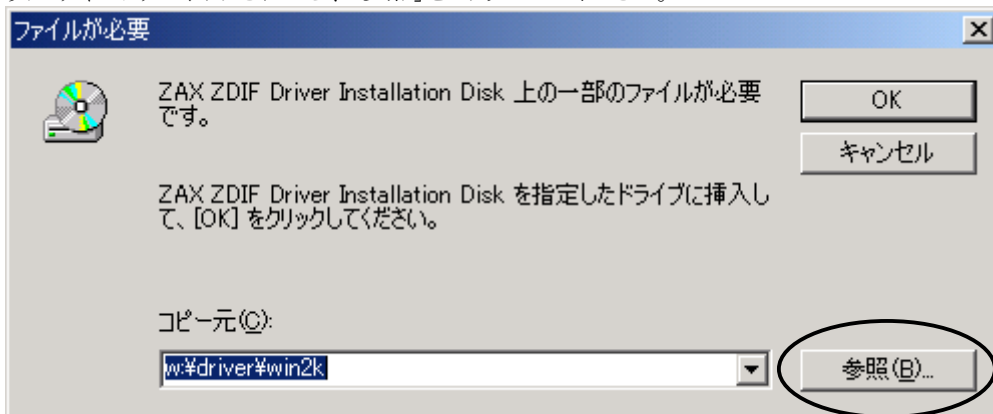
次のウィンドウが表示されたら、「OK」をクリックしてください。



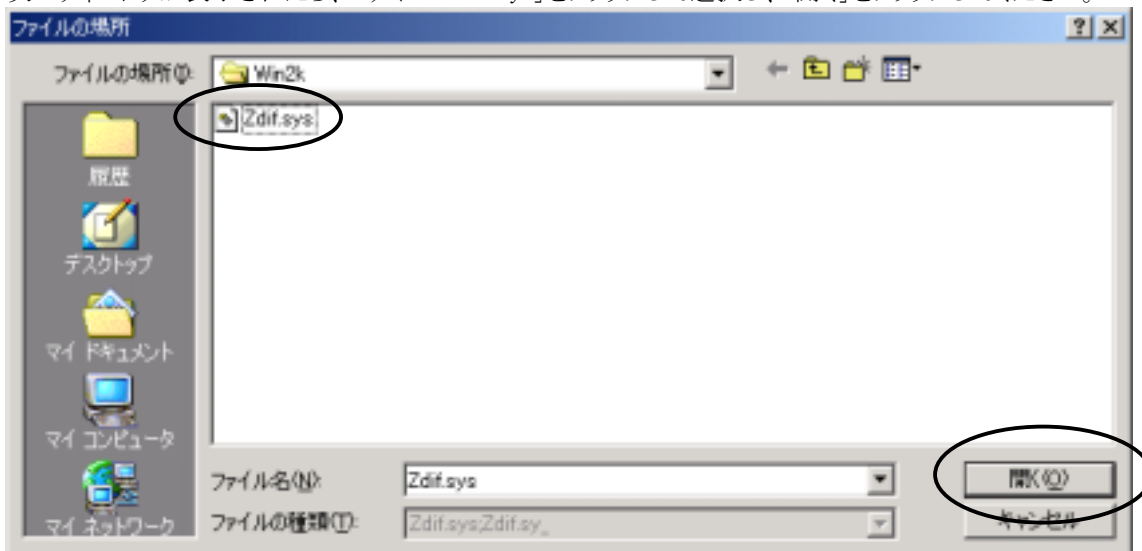
次のダイアログボックスが表示されたら、「OK」をクリックしてください。



次のウィンドウが表示されたら、「参照」をクリックしてください。

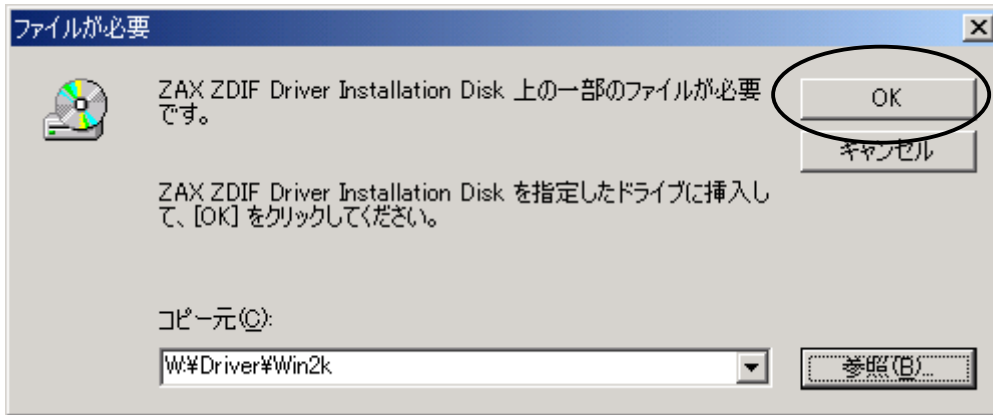


次のウィンドウが表示されたら、ファイル「Zdif.sys」をクリックして選択し、「開く」をクリックしてください。

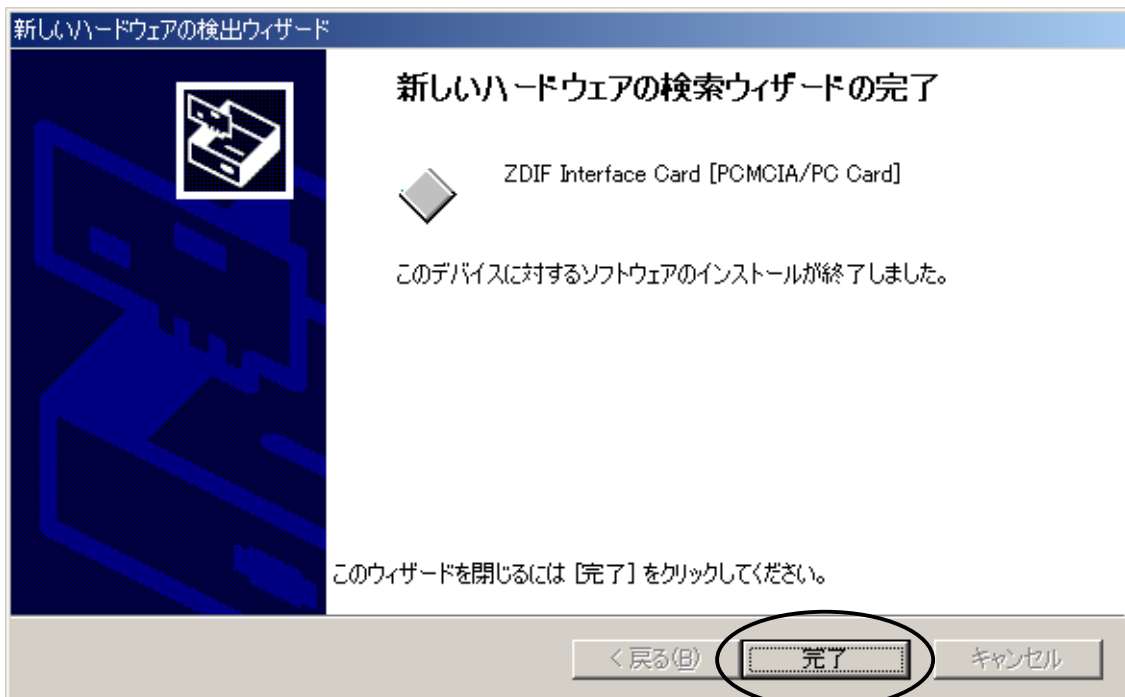




次のウィンドウが表示されたら、「OK」をクリックしてください。

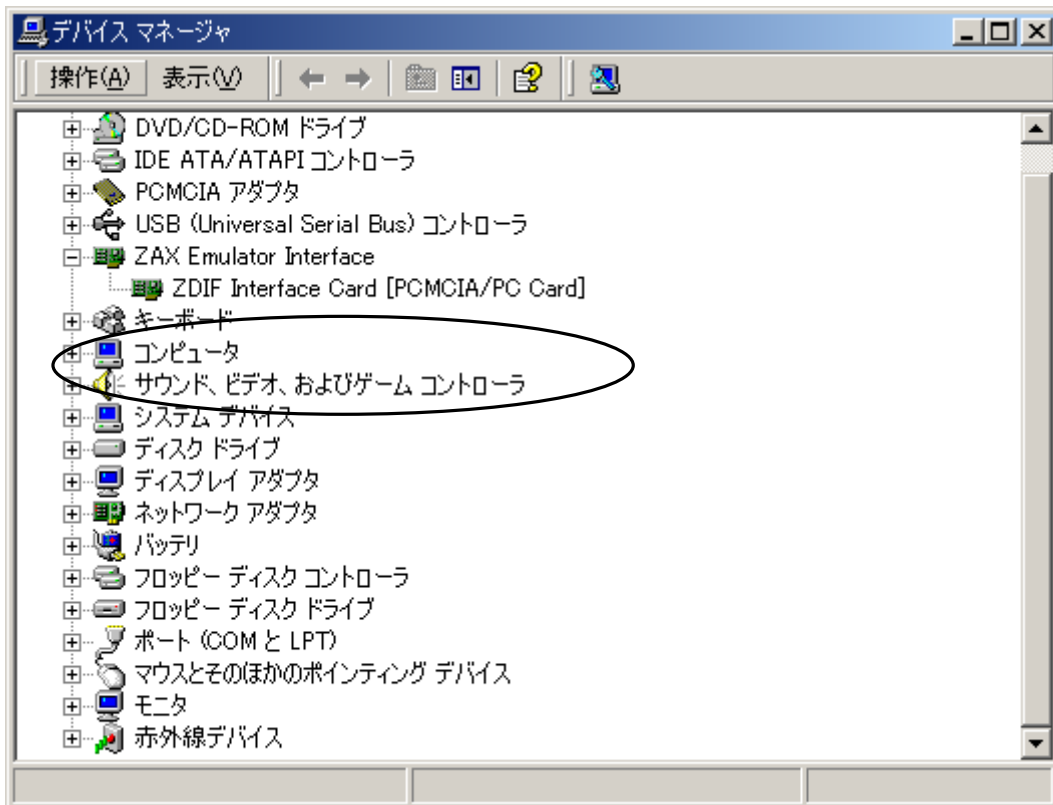


インストール完了を示す次のウィンドウが表示されたら、「完了」をクリックしてください。



ドライバのインストールが終了したら、次の手順でインストールが正常に終了したか確認することができます。

スタートメニュー → 設定 → コントロール パネル → システム → ハードウェア  
デバイス マネージャを選択  
「ZAX Emulator Interface」左の「+」をクリック



「ZDIF Interface Card [PCMCIA/PC Card]」が表示されれば正常です。

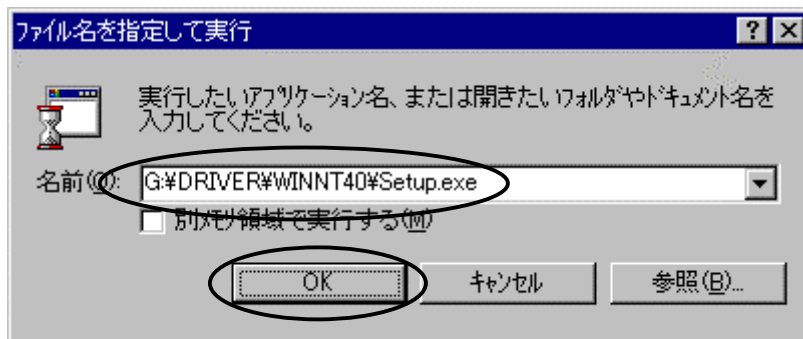
### 2.1.3 WindowsNT4.0 の場合

ノート PC の電源がオフしている状態で PC カードを差した後、電源をオンします。

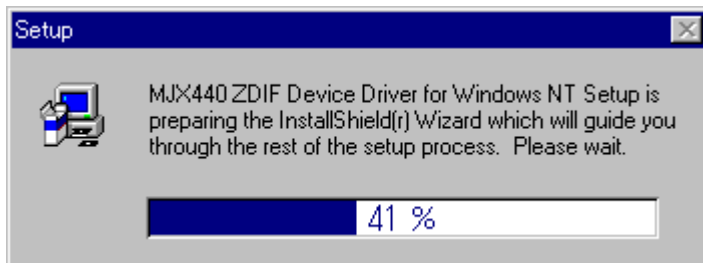
Windows NT4.0 が起動したら、CD-ROMドライブに「*MJX for ARM7/9 Tools Disk*」の CD-ROM をセットします。

「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。

「名前(O)」に `G:¥DRIVER¥WINNT40¥Setup.exe`(CD-ROM ドライブが G ドライブの場合)、を指定し、「OK」をクリックします。



セットアップ プログラムが起動します。



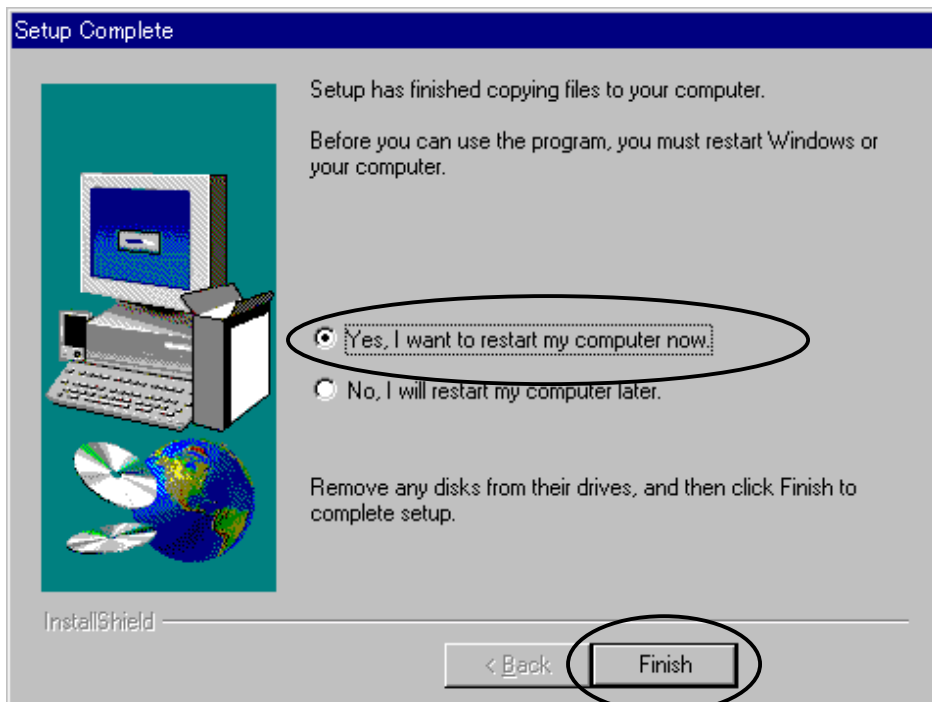
「Next >」をクリックします。

自動的にドライバがインストールされます。



ドライバのインストール後に以下のダイアログが表示されます。

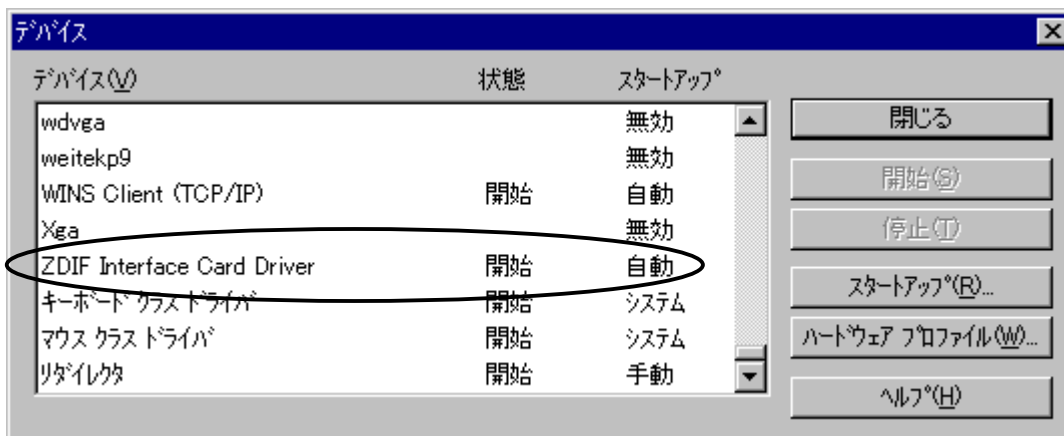
「Yes, I want to restart my computer now」を選択し、「Finish」をクリックします。



Windows NT4.0 が再起動します。

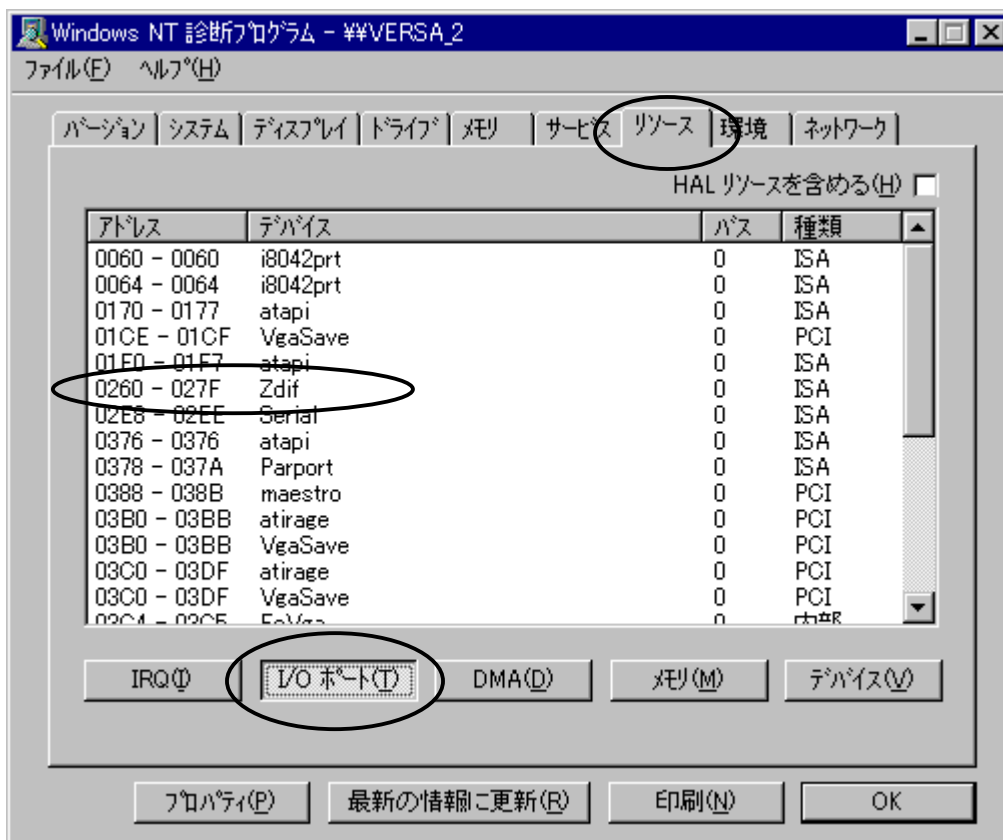
インストールが正しく行われたことを確認するために、コントロール パネルから「デバイス」をダブル クリックして開き、「ZDIF Interface Card Driver」を探します。

状態が「開始」に、スタートアップが「自動」になっていることを確認します。



続いて、「スタート」、「プログラム(P)」、「管理ツール (共通)」から「Windows NT 診断プログラム」を起動します。

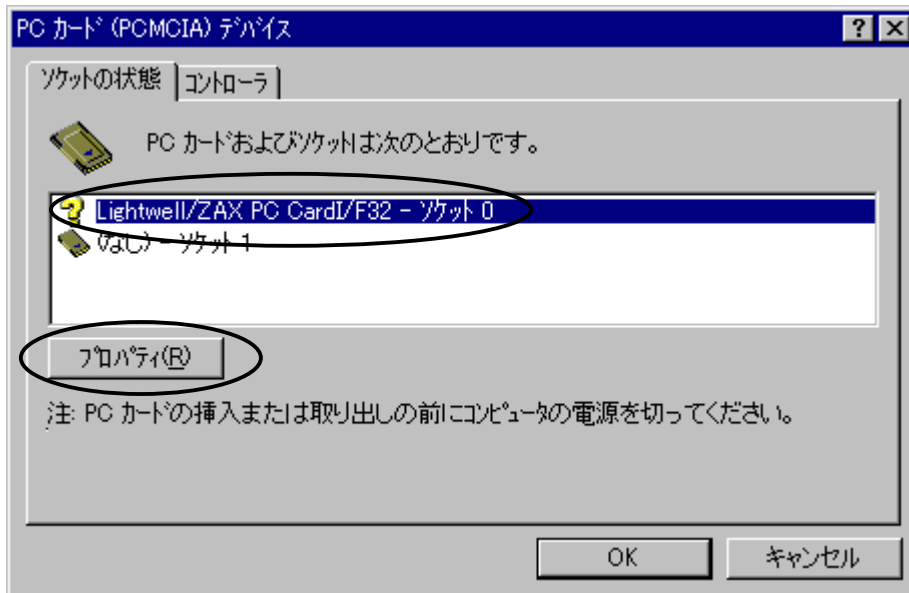
「リソース」タブをクリックし、「I/O ポート(T)」をクリックします。



「デバイス」で Zdif を探し、PC カードに割り当てられた I/O ポートが「アドレス」に表示されていることを確認します。アドレスの設定値は、動作環境によって異なります。

コントロール パネルで「PC カード(PCMCIA)」をダブル クリックして開きます。

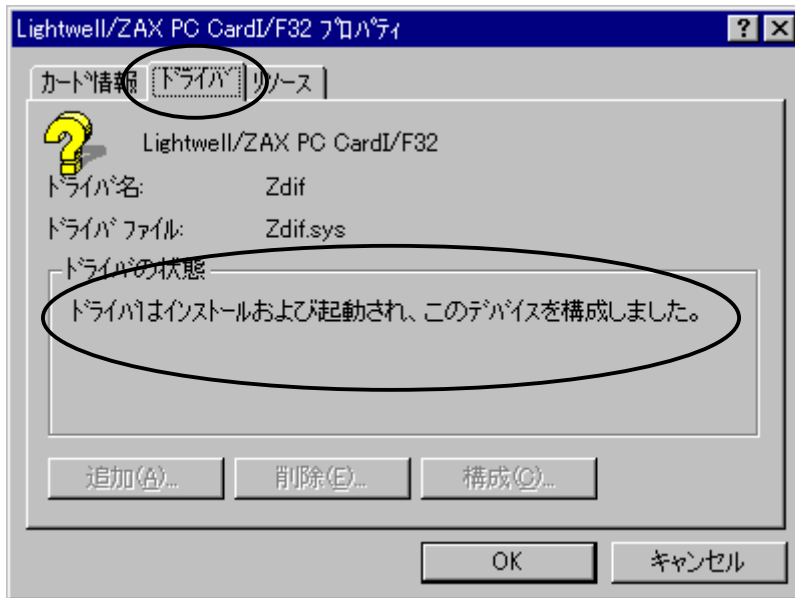
「Lightwell/ZAX PC Card I/F32」をクリックして選択し、「プロパティ(R)」ボタンをクリックします。



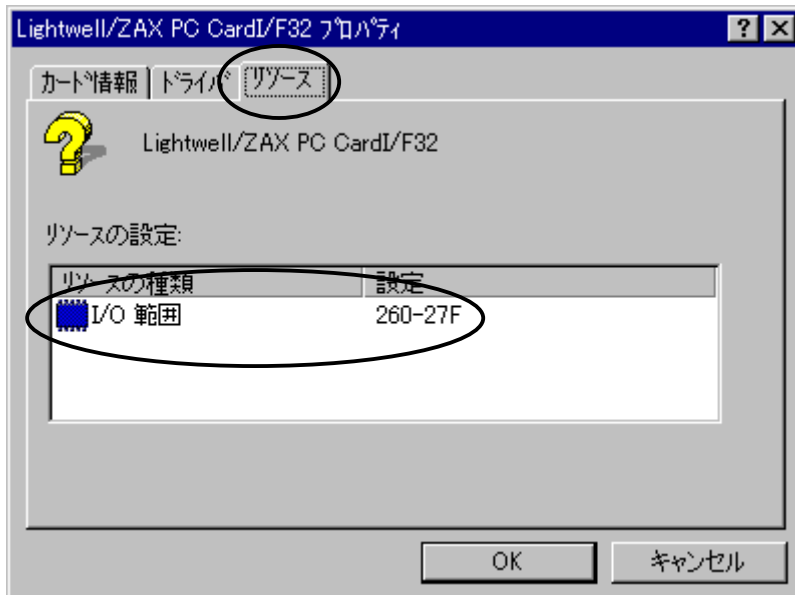
「カード情報」タブをクリックし、「デバイスの状態」が「デバイスは正常に動作しています。」になっていることを確認します。



「ドライバ」タブをクリックし、「ドライバの状態」が「ドライバはインストールおよび起動され、このデバイスを構成しました。」になっていることを確認します。



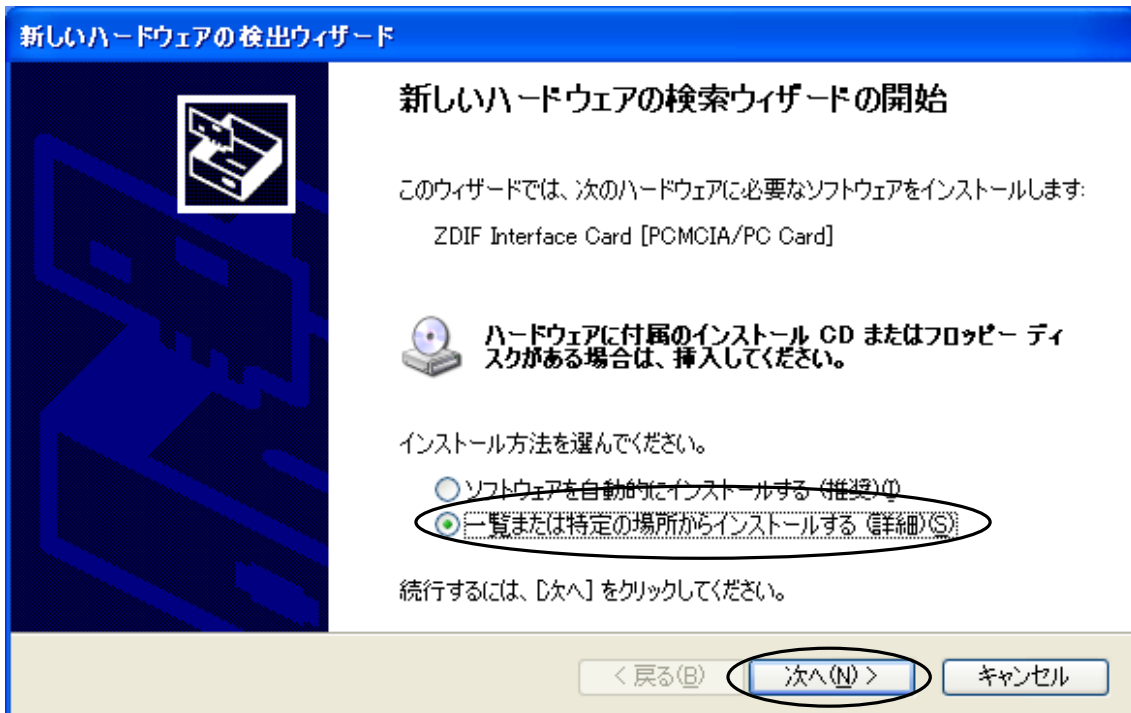
「リソース」タブをクリックし、「リソースの設定」、「I/O 範囲」が以下のいずれかになっていることを確認します。



- 0220 – 023F
- 0260 – 027F
- 02E0 – 02FF
- 0320 – 033F
- 03E0 – 03FF

### 2.1.4 WindowsXP の場合

ノートPC の電源オンの状態で PC カードを差すと、ハードウェア ウィザードが起動します。  
次のウィンドウが表示されたら、「一覧または特定の場所からインストールする」を選んで、「次へ」をクリックしてください。

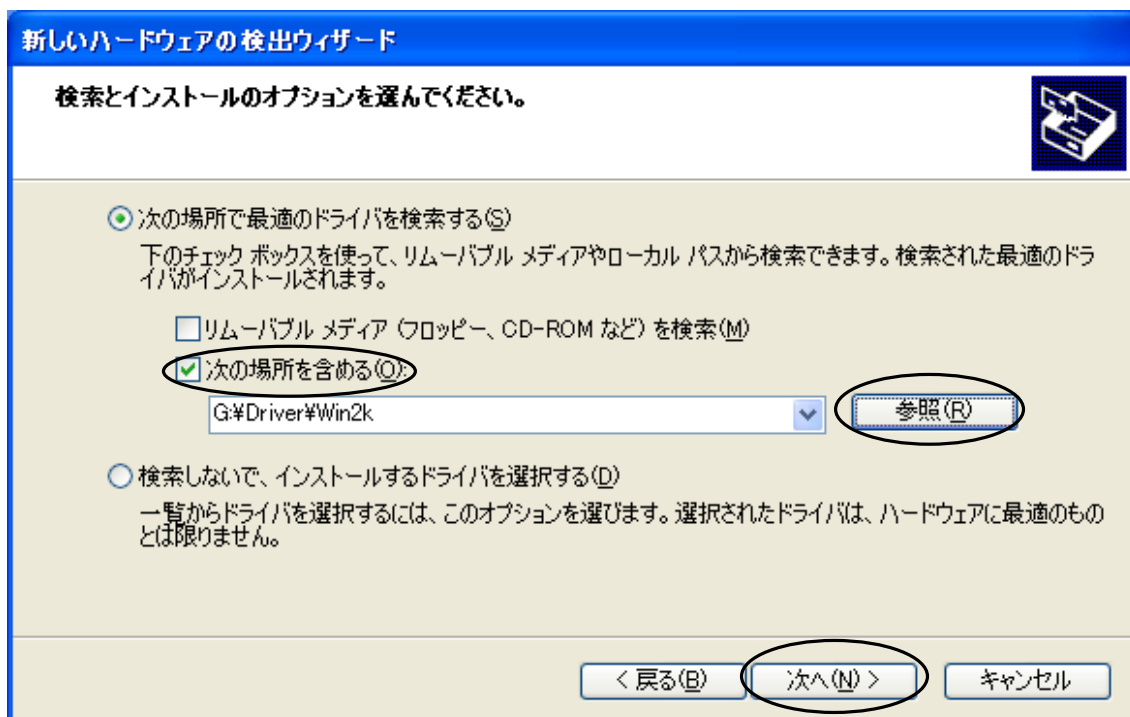




次のウィンドウが表示されたら、「Mjx for ARM7/9 Tools Disk」CD-ROM をドライブにセットしてください。

次に、「次の場所を含める」を選択し、参照ボタンをクリックして

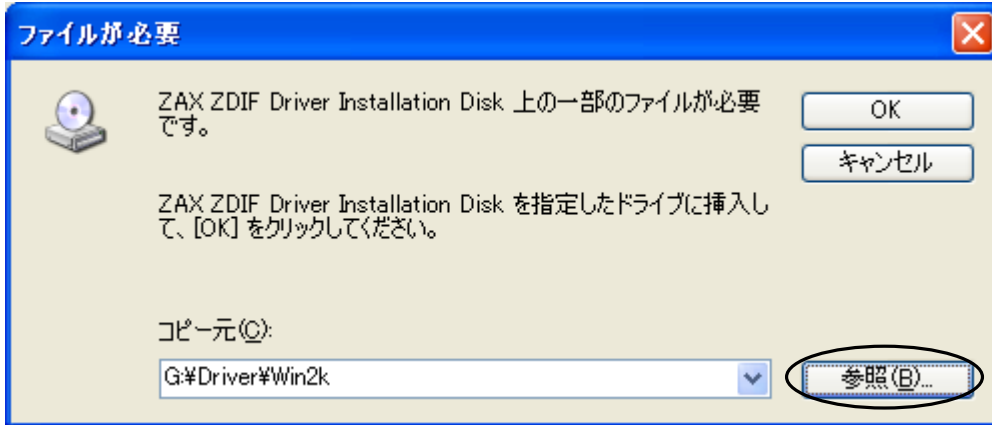
G:\Driver\Win2k(CD-ROMドライブが G の場合)を選択してください。選択できたら「次へ」をクリックしてください。



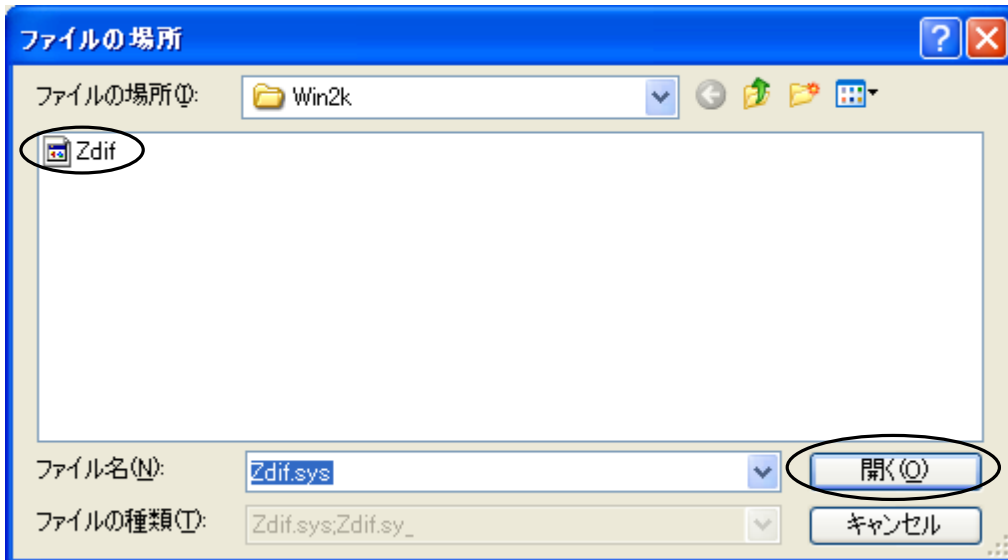
次のダイアログボックスが表示されたら、「OK」をクリックしてください。



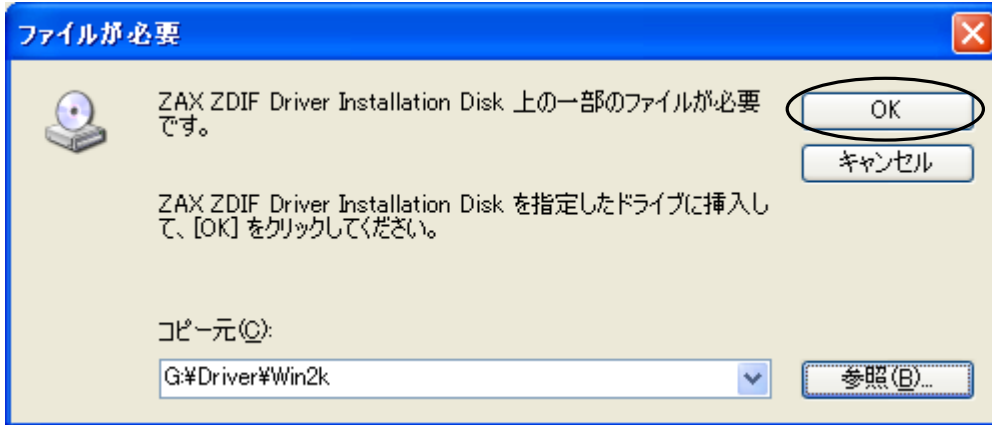
次のウィンドウが表示されたら、「参照」をクリックしてください。



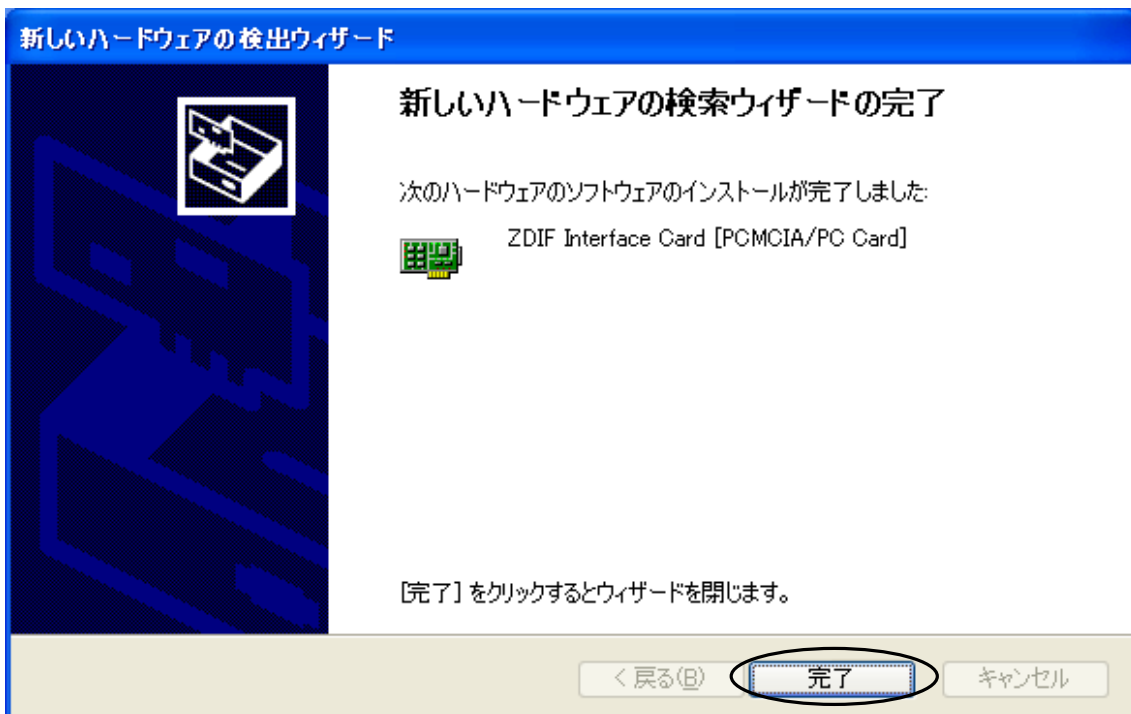
次のウィンドウが表示されたら、ファイル「Zdif.sys」をクリックして選択し、「開く」をクリックしてください。



次のウィンドウが表示されたら、「OK」をクリックしてください。



インストール完了を示す次のウィンドウが表示されたら、「完了」をクリックしてください。



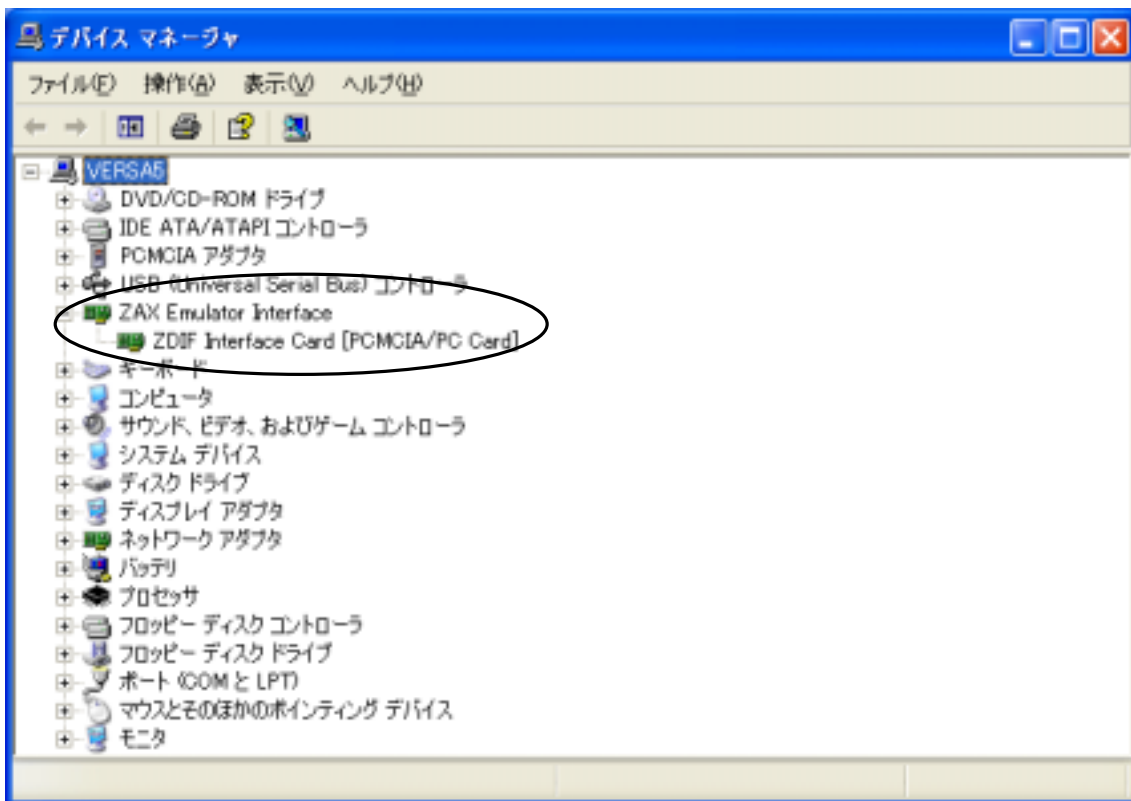
## 第二章 デバイス ドライバのインストール

ドライバのインストールが終了したら、次の手順でインストールが正常に終了したか確認することができます。

スタートメニュー → マイ コンピュータ → システムタスク(システム情報を表示する)を選択して、システムプロパティの「ハードウェア」タブをクリックします。

デバイスマネージャを選択して、

「ZAX Emulator Interface」左の「+」をクリックします。



「ZDIF Interface Card [PCMCIA/PC Card]」が表示されれば正常です。

## 2.2 デバイス ドライバのアンインストール手順

### 2.2.1 Windows98/98SE の場合

- ①PC カードを差した状態で、コントロール パネルから「システム」をダブル クリックして開き、「デバイス マネージャ」タブをクリックします。
- ②次に「ZDIF」クラスをクリックして、その下にある「ZDIF Interface Card [PCMCIA/PC Card]」をクリックして選択し、「削除(E)」ボタンをクリックします。
- ③「デバイス削除の確認」ダイアログが表示されるので、「OK」ボタンをクリックします。

◆PC カードのデバイス ドライバの情報を完全に削除したい場合は、INF ファイルも削除します。

- ①エクスプローラを起動し「**表示(V)**」メニューから「**フォルダ オプション(O)...**」を選択し、「フォルダ オプション」ダイアログで「**表示**」タブをクリックします。
- ②「**登録されているファイルの拡張子は表示しない**」のチェックを外し(オフ)、「**全てのファイルを表示する**」をチェック有り(オン)にして、「**OK**」をクリックします。

これで INF ファイルが表示されるようになります。

- ③エクスプローラでシステム ディレクトリ(デフォルトは「C:\Windows」)の下の「**Inf\Other**」ディレクトリを開き、「**Zaxzdif.inf**」ファイルを削除します。

## 2.2.2 Windows2000 の場合

- ①PC カードを差した状態で、**コントロール パネル**から「**システム**」をダブル クリックして開き、「**ハードウェア**」タブから「**デバイス マネージャ**」ボタンをクリックします。
- ②「**ZAX Emulator Interface**」クラスをクリックして、その下にある「**ZDIF Interface Card [PCMCIA]**」をダブルクリックして表示したプロパティウィンドウの「**ドライバ**」タブを選択します。そして「**削除(E)**」ボタンをクリックします。
- ③「デバイス削除の確認」ダイアログが表示されるので、「**OK**」ボタンをクリックします。

## 2.2.3 WindowsNT4.0 の場合

- ①PC カードを差した状態で、**コントロール パネル**から「**デバイス**」をダブル クリックして開き、「**ZDIF Interface Card Driver**」をクリックして選択します。
- ②続いて「**停止(T)**」ボタンをクリックします。
- ③「ZDIF Interface Card Driver デバイスを停止してもよろしいですか」というダイアログが表示されるので、「**はい(Y)**」をクリックします。

デバイスが停止し、ドライバがアンロードされます。

- ④**コントロール パネル**から「**アプリケーションの追加と削除**」をダブル クリックして開き、「**インストールと削除**」タブをクリックします。
- ⑤「**MJX440 ZDIF Device Driver for Windows NT**」をクリックして選択し、「**追加と削除(R)**」ボタンをクリックします。

- ⑥「**Confirm File Deletion**」ダイアログが表示されるので、「**はい(Y)**」をクリックします。

自動的にドライバがアンインストールされます。

- ⑦「**OK**」をクリックし、終了します。

- ⑧Windows NT4.0 をシャットダウンさせノート PC の電源をオフした後、PC カードを抜きます。

## 2.2.4 WindowsXP の場合

①PC カードを差した状態で、スタート → マイ コンピュータを選択して、システムタスクの「システム情報を表示する」をクリックして、システムプロパティウインドウを表示します。

②システムプロパティウインドウで「ハードウェア」タブを選択し、「デバイスマネージャ」をクリックしてデバイスマネージャウインドウを表示します。

③「**ZAX Emulator Interface**」 クラスをクリックして、その下にある「**ZDIF Interface Card [PCMCIA]**」を選択して右クリックします。そして、「削除」を選択します。

④「デバイス削除の確認」ダイアログボックスが表示されるので、「OK」ボタンをクリックします。



## 第三章 ハードウェアの接続

MJX330 とホストの接続方法、MJX330 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。

### 3.1 MJX330 とホストの接続

#### MJX330 とホストの接続

プローブケーブルを MJX330 に接続し、MJX330 をホストの PC カードスロットに取り付けてください。

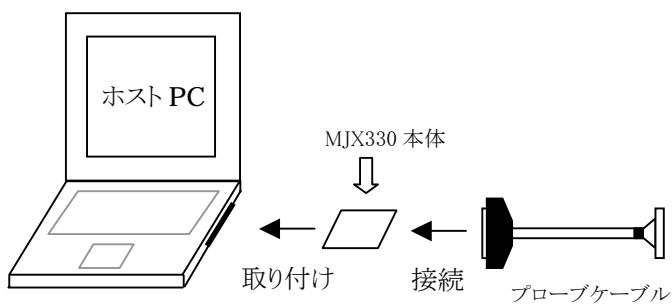


図 3-1 MJX330 とホストの接続

- 【注意】** プローブ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。
- ◆ PC 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。
  - ◆ カード 2 枚を無理に差し込むと、PC カード スロット、および MJX330 カードのコネクタ部が壊れる場合があります。
  - ◆ PC 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。

## 3.2 MJX330 とターゲットの接続

### MJX330 とターゲットの接続

図 3-1 のとおり、MJX330 とプローブケーブルを接続した後、図 3-2 のように、ターゲット システムを接続してください。

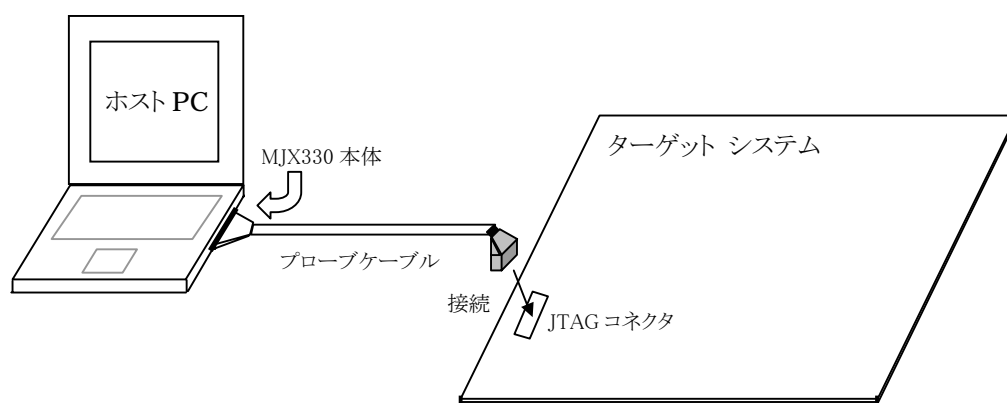


図 3-2 MJX330 とターゲットの接続

**【注意】** ターゲットシステムへプローブケーブルを接続するときは、必ずターゲットシステムの電源を切ってから行ってください。

**【注意】** 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。

## 第四章 ソフトウェアのインストール

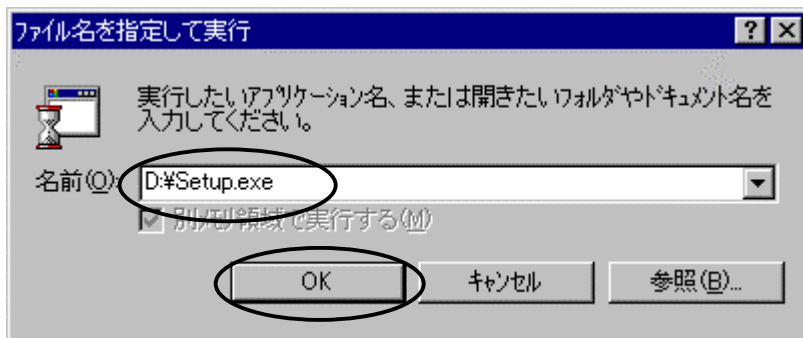
MJX330 を操作するソフトウェアのインストール方法について記述しています。

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。

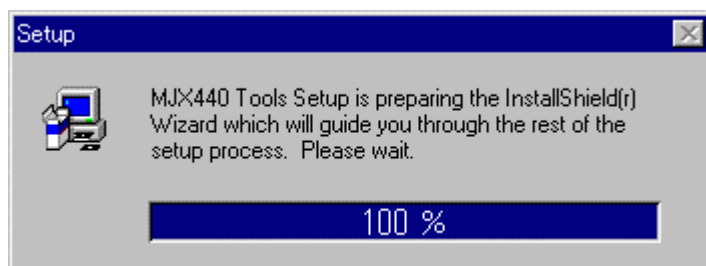
1. Green Hills Software 社の統合開発環境 MULTI をインストールします。
2. CD-ROMドライブに「MJX for ARM7/9 Tools Disk」の CD-ROM をセットします。

「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。

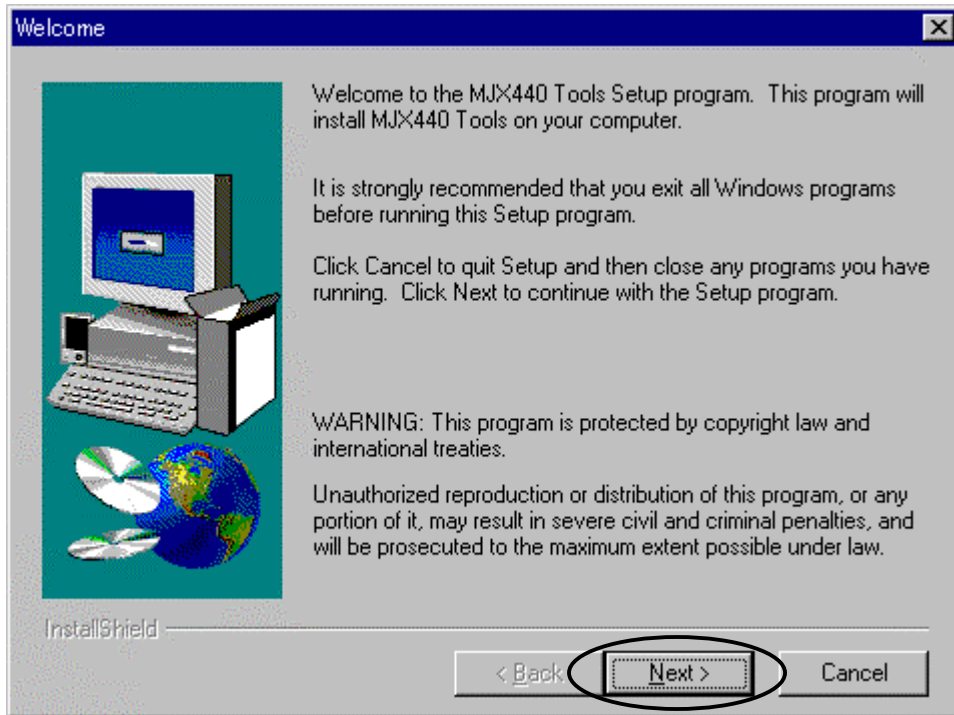
3. 「名前(O)」にセットアップ プログラム Setup.exe を指定し、「OK」をクリックします。(CD-ROMドライブが Dドライブの場合、「D:¥Setup.exe」)



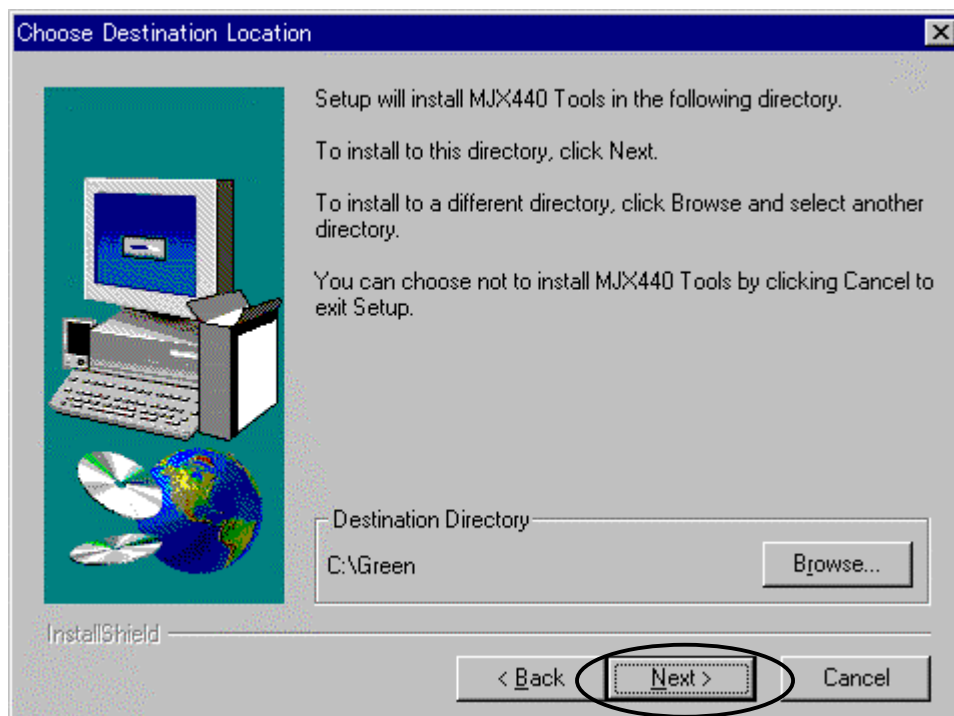
4. セットアップ プログラムが起動します。



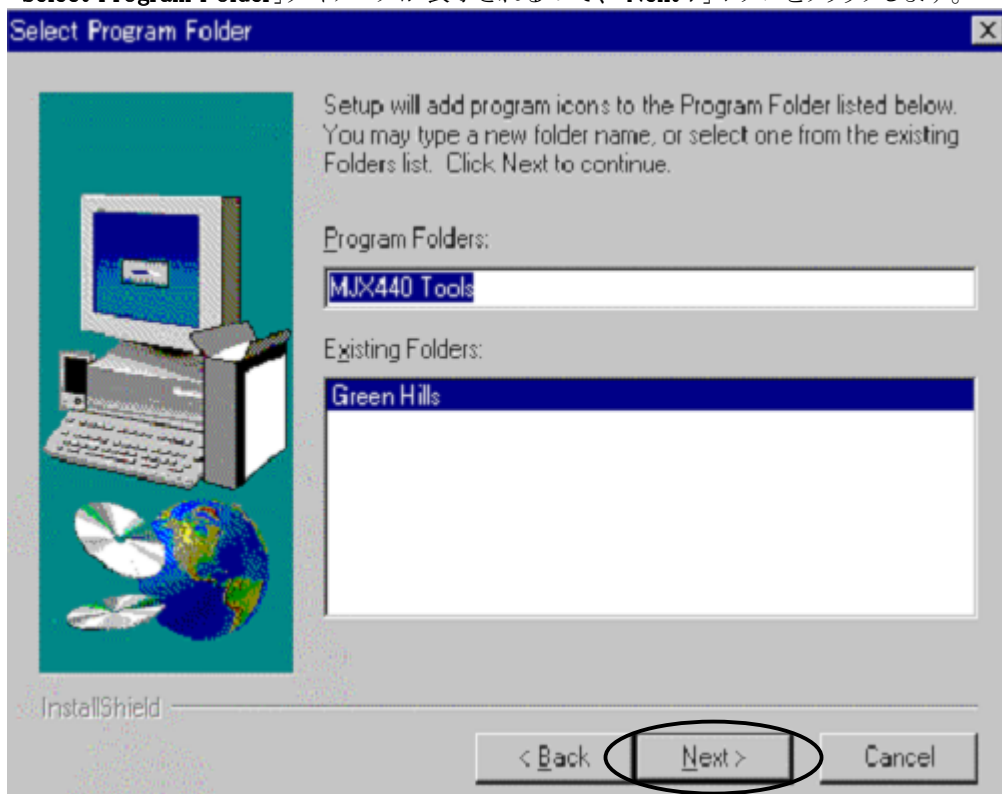
5. 「Welcome」ダイアログが表示されるので、「Next >」をクリックします。



6. 「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。

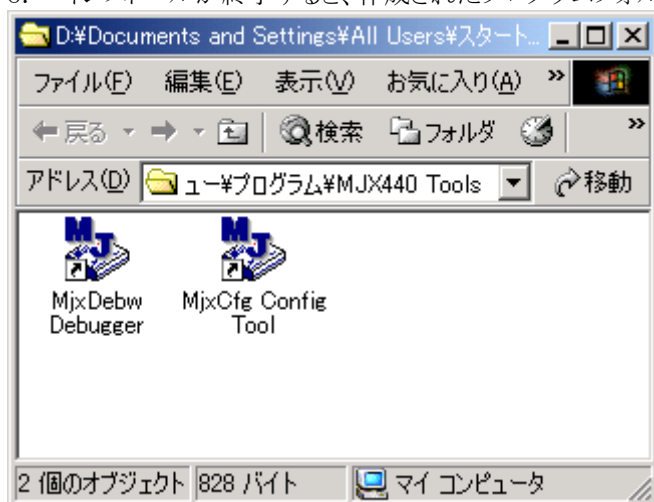


「Select Program Folder」ダイアログが表示されるので、「Next >」ボタンをクリックします。

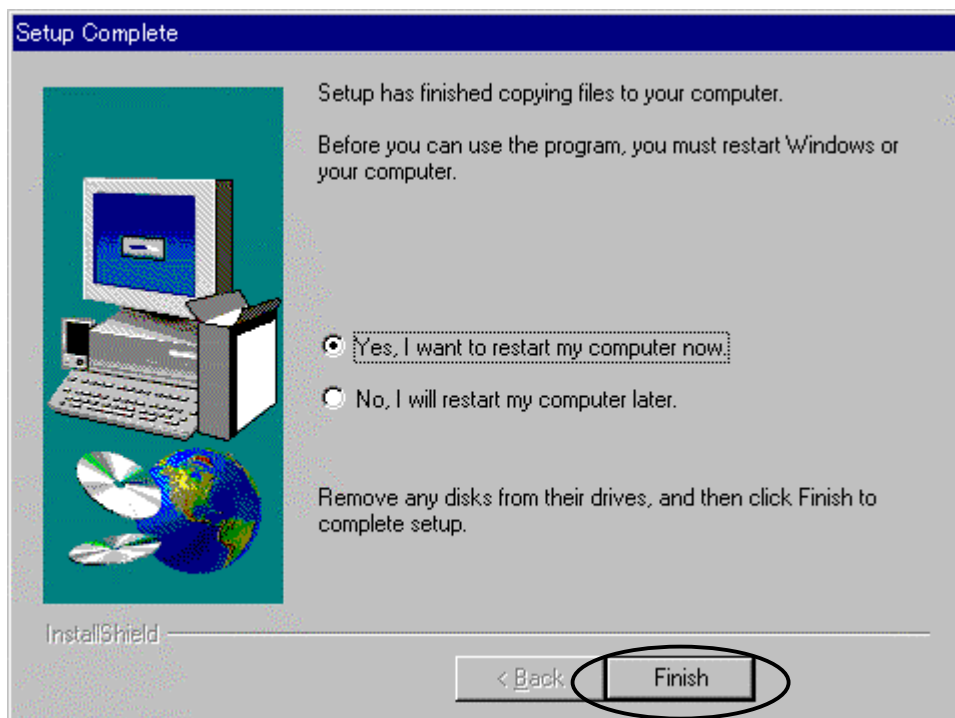


7. インストールが開始します。

8. 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。



9. 「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)
MJXDDBAPI.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

この他にサブディレクトリにフラッシュメモリ書込に使用するデバイス定義ファイルがインストールされます。



プロセスとハードウェアの構成は次のとおりです。

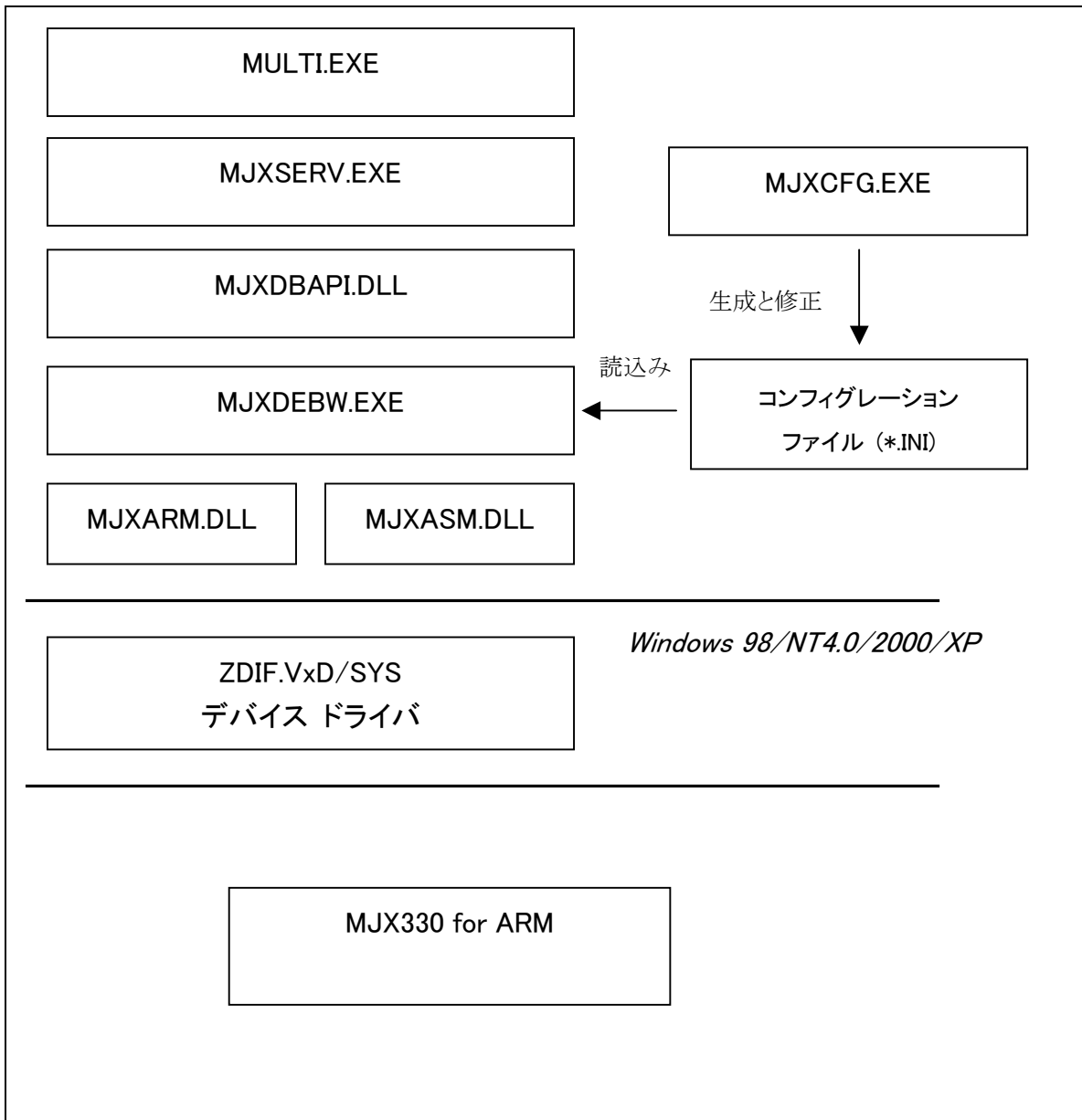


図 4-1 プロセスとハードウェアの構成

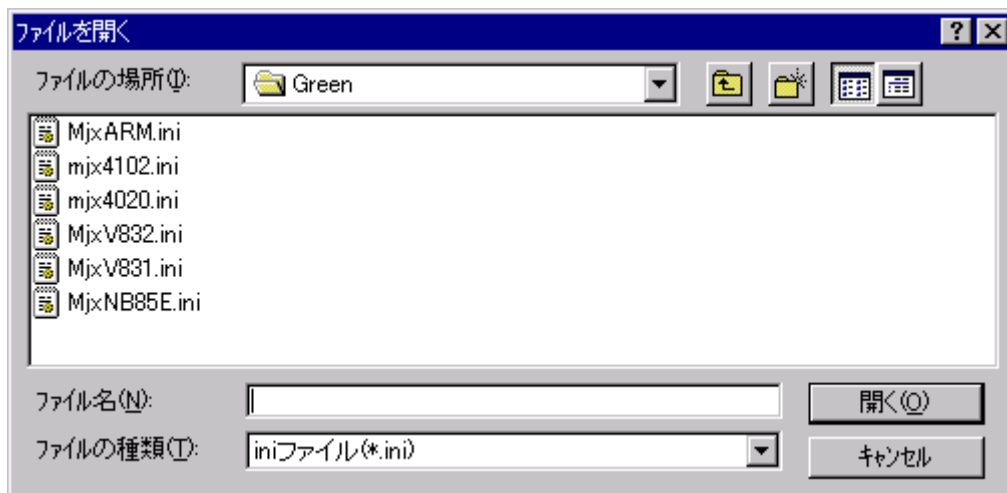
- MULTI.EXE は MJXSERV.EXE を子プロセスとして呼び出します。
- MJXSERV.EXE は MJXDBAPI.DLL を経由して MJXDEBW.EXE を子プロセスとして呼び出します。
- MJXDEBW.EXE は MDXSERV.EXE の子プロセスとしてだけでなく、単体でも動作できます。

## 第五章 MJX330 の環境設定

MJX330 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。

MJX330 の環境設定は、コンフィグ支援ツール MJXCFCG 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 MJXCFCG を起動してください。手順は、次のとおりです。

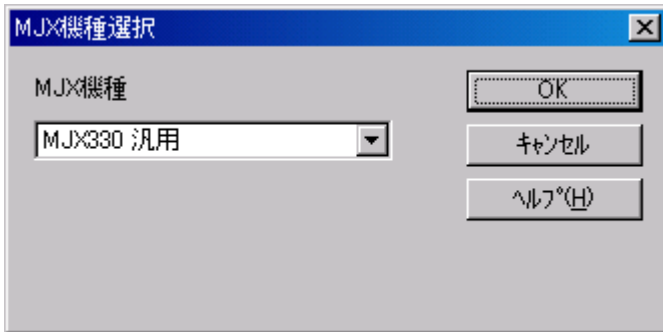
1. スタートメニュー
2. プログラム(P)
3. MJX Tools
4. MjxCfg Config Tool



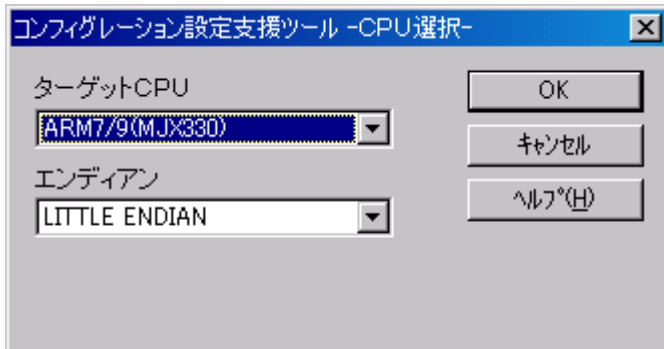
テンプレートとして MJXARM.INI を利用できます。

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「MJX330 汎用」を選択し、**OK** ボタンを押してください。

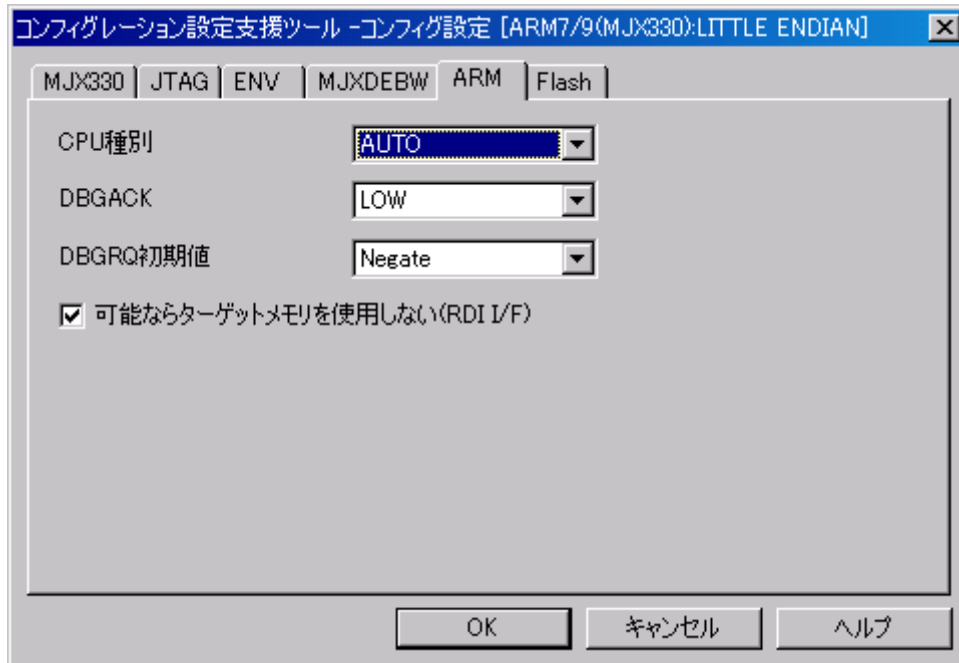


ターゲットCPU\*1とエンディアンを選択し、**OK** ボタンを押してください。

---

\*1 ターゲットCPUはARM7/9 を指定してください。

コンフィグレーションファイルを指定し、ARM タブをクリックすると、次のダイアログ ボックスが表示されます。



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

## 第六章 ソフトウェアの起動と終了

MJX330 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 MJX330 の環境設定」を行ってください。

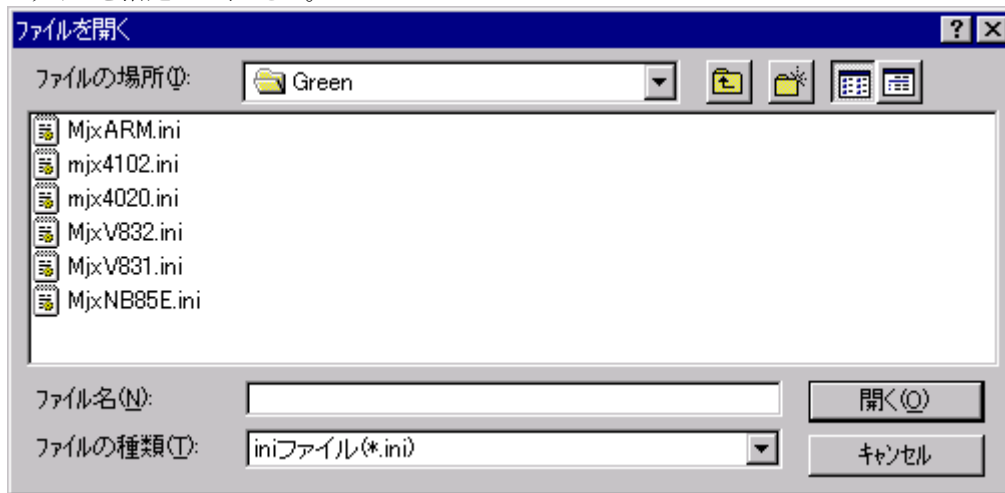
**【注意】** MJX330 を操作するソフトウェアは、同時にひとつしか実行できません。

### MULTI を使用する場合

[MULTI](#) を使って MJX330 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。<sup>\*1</sup>

```
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 MJX330 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。<sup>\*2</sup>



\*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。

\*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。

正常にリモート接続できた場合は、MJXDEBW のウィンドウが開きます。

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 *MjxArm.ini* を指定します。

```
remote mjxserv MjxArm.ini
```

終了するには、*quit* コマンドを入力してください。

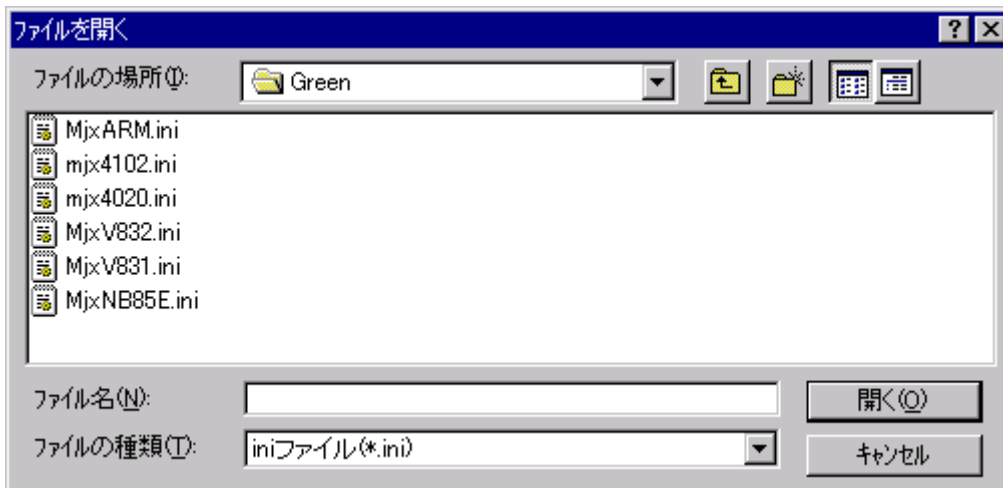
```
quit
```

### MJXDEBW を使用する場合

[MJXDEBW](#)を使ってMJX330 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. MJX 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。



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 *MjxArm.ini* を指定します。

```
MJXDEBW MjxArm.ini
```

終了するためには、*quit* コマンドを入力してください。

```
quit
```



### AXD を使用する場合

MjxRDI サーバーは ADS(ARM Developer Suite ARM 社製)から MJX330 for ARM を使用するためのインターフェース プログラムです。

ADS との接続にはインターフェース仕様[RDI Ver1.5.1]を使用しています。

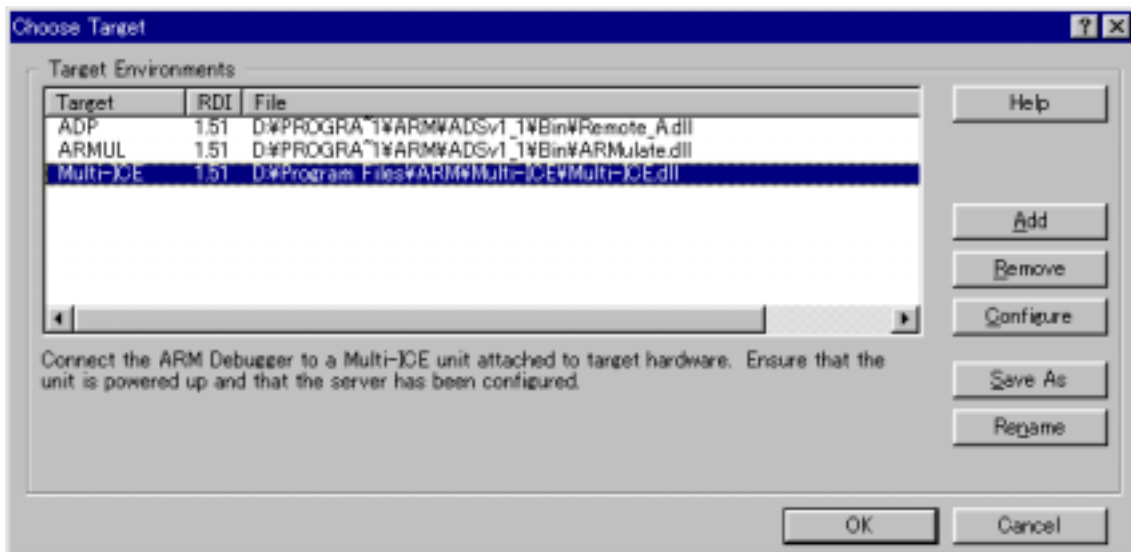
**【注意】** SDT ではご使用になれません。

#### (1) 起動方法

##### ① 初回

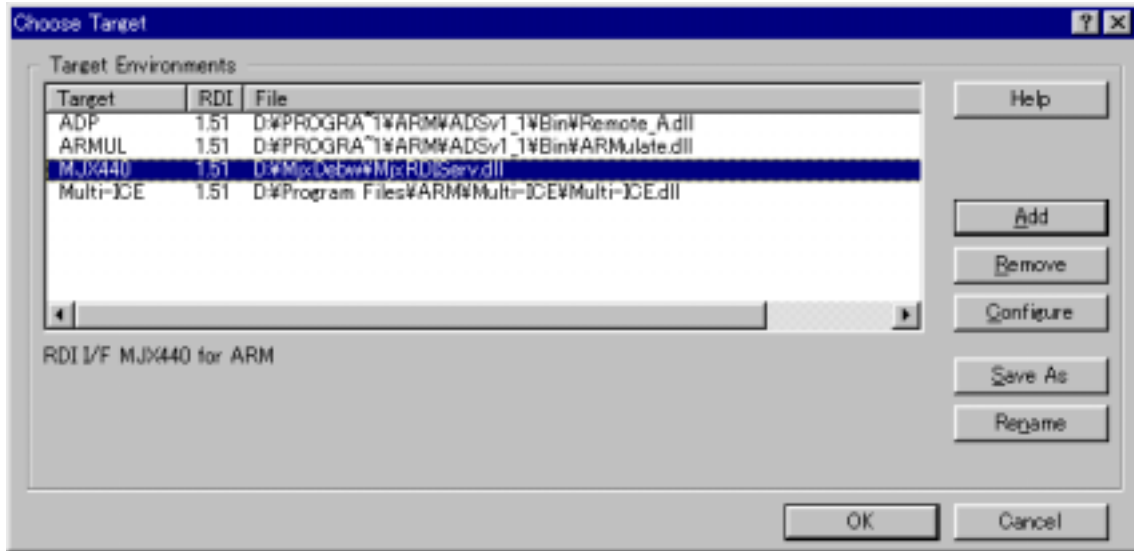
AXD デバッガを起動します。

[Option]—[Cofigure Target]から Choose Target ダイアログを開きます。

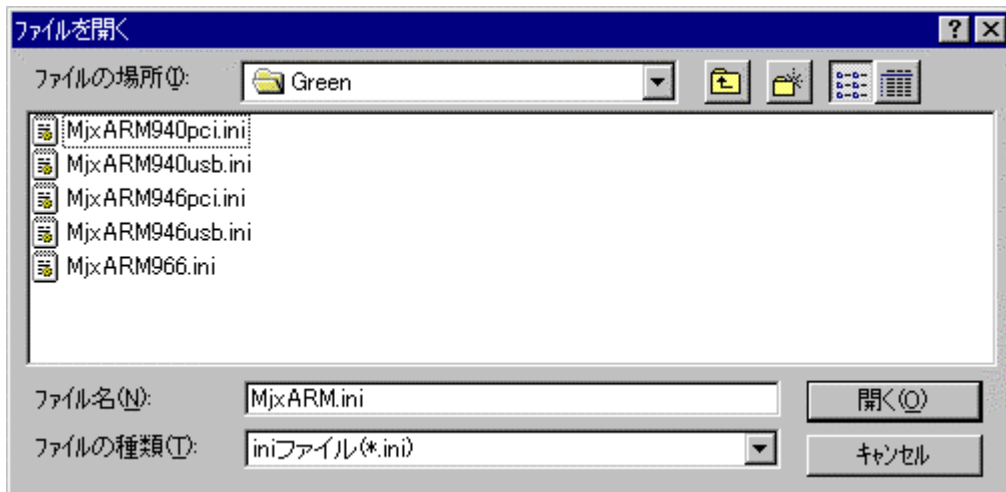


Add ボタンを押して、ファイルを開くダイアログで *MJX for ARM7/9 Tools Disk* をインストールしたディレクトリからファイル *MjxRDIServ.dll* を選択します。

Choose Target ダイアログで OK ボタンを押します。



続いて、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 MJX330 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。



初回起動時にはこのダイアログが3回表示されます。(AXD のみ)  
すべて同じコンフィグレーション ファイルを指定してください。

② 2回目以降

AXD デバッガを起動します。

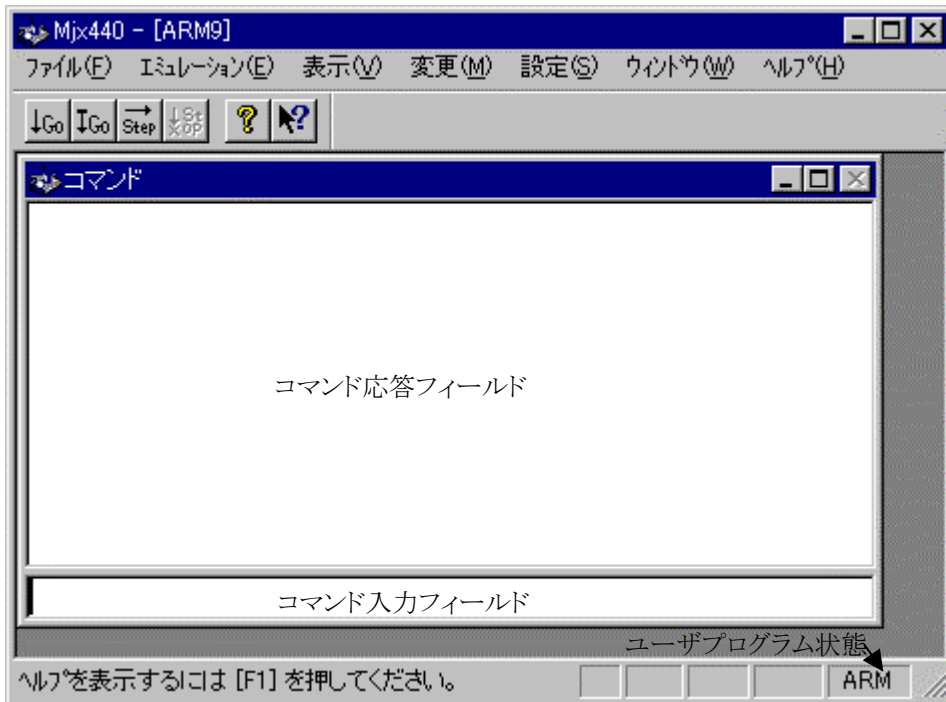
ファイルを開くダイアログがでますのでコンフィグレーション ファイルを指定します。



## 第七章 MJXDEBW コマンド

この章ではデバッガ MJXDEBW のコマンドの使い方について記述しています。

MJXDEBW コマンドは、コマンド ウィンドウ内で実行するコマンドです。コマンド入力フィールドから入力し、コマンド応答フィールドへ結果を表示します。



- コマンド応答フィールドに表示できる行数は、「[表示\(V\) ~ オプション\(O\) ~ 表示\(V\)...](#)」メニューによって変更することができます。
- コマンド入力フィールドは履歴 バッファを持っています。以前に入力したコマンドは、上下のカーソル キーによって呼び出すことができます。保持するコマンドの行数は、「[表示\(V\) ~ オプション\(O\) ~ 表示\(V\)...](#)」メニューによって変更することができます。
- ステータス行の右端に、ユーザ プログラムの状態が常に表示されます。

ARM	ARM ステートでユーザ プログラム停止中
Thumb	Thumb ステートでユーザ プログラム停止中
RUN	ユーザ プログラム実行中

- ステートは次のコマンドで変更することができます。

*reg thumb=0* (ARM ステートにする)

*reg thumb=1* (Thumb ステートにする)

MJXDEBW コマンドの一覧は、次のとおりです。

<a href="#"><u>ABORT</u></a>	ユーザ プログラムを強制停止します
<a href="#"><u>BATCH</u></a>	バッチ ファイルに記述された MJXDEBW コマンドを実行します
<a href="#"><u>BP</u></a>	ブレークポイントを表示/解除/有効/無効にします
<a href="#"><u>BP/S</u></a>	命令ソフトウェア ブレークポイントを設定します
<a href="#"><u>BP/Wn</u></a>	ウォッチポイントを設定します
<a href="#"><u>BP/WC</u></a>	ウォッチポイントの連携モードを設定します
<a href="#"><u>CLEAR</u></a>	コマンド応答フィールドをクリアします
<a href="#"><u>CONFIG</u></a>	MJXDEBW の環境を表示/変更します
<a href="#"><u>DUMP</u></a>	メモリの内容を表示します
<a href="#"><u>EXAMINE</u></a>	メモリの内容を変更します
<a href="#"><u>FILL</u></a>	メモリの内容をフィルします
<a href="#"><u>FLASH</u></a>	フラッシュ メモリの内容を書き換え、消去します
<a href="#"><u>GO</u></a>	ユーザ プログラムを実行します
<a href="#"><u>HISTORY</u></a>	リアルタイム トレース結果を表示します*1
<a href="#"><u>INIT</u></a>	MJXDEBW を再初期化します
<a href="#"><u>JOURNAL</u></a>	コマンド実行結果をファイルへ出力します
<a href="#"><u>LOAD</u></a>	ファイルをメモリへダウンロードします
<a href="#"><u>MOVE</u></a>	メモリの内容をブロック転送します
<a href="#"><u>QUIT</u></a>	MJXDEBW を終了します
<a href="#"><u>REGISTER</u></a>	レジスタの内容を表示/変更します
<a href="#"><u>SETLOAD</u></a>	ブレークポイントおよびトレースの設定をファイルから読み出します
<a href="#"><u>SETSAVE</u></a>	ブレークポイントおよびトレースの設定をファイルへ保存します
<a href="#"><u>STEP</u></a>	ユーザ プログラムをステップ実行します

---

\*1 MJX330 for ARM はトレースメモリを持っておりませんので、このコマンドは機能致しません。

<a href="#"><u>TRACE</u></a>	リアルタイム トレースの全トレース条件を表示します*2
<a href="#"><u>TRACE/C</u></a>	リアルタイム トレースのカウンタ条件を設定/解除します*2
<a href="#"><u>TRACE/D</u></a>	リアルタイム トレースのデータ トレース条件を設定/解除します*2
<a href="#"><u>TRACE/E</u></a>	リアルタイム トレースのトレース モードを設定します*2
<a href="#"><u>TRACE/I</u></a>	リアルタイム トレースの命令トレース条件を設定します*2
<a href="#"><u>TRACE/P</u></a>	リアルタイム トレースのトレース条件をカスタム設定します*2
<a href="#"><u>TRACE/Q</u></a>	リアルタイム トレースのETMの構成と現在の状態を表示します*2
<a href="#"><u>TRACE/R</u></a>	リアルタイム トレースの全トレース条件を無効にします*2
<a href="#"><u>TRACE/S</u></a>	リアルタイム トレースのシーケンサ条件を設定/解除します*2
<a href="#"><u>TRACE/T</u></a>	リアルタイム トレースを強制停止します*2
<a href="#"><u>UNASM</u></a>	メモリの内容を逆アセンブル表示します
<a href="#"><u>VERSION</u></a>	バージョンを表示します
<a href="#"><u>WAIT</u></a>	ユーザ プログラムが停止するまで待ちます

---

\*2 MJXはトレースメモリを持っておりません。ロジック アナライザー等の外部装置によりETM信号をモニターする場合にご使用ください。

### 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	<a href="#">ABORT</a> と同じです
D	<a href="#">DUMP</a> と同じです
CL	<a href="#">CLEAR</a> と同じです
CO	<a href="#">CONFIG</a> 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。

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



## 第七章 MJXDEBW コマンド

これらの MJXDEBW コマンドを実行した場合、それ以降、MULTI と MJXSERV 間の整合性がとれなくなる危険性があります。命令ソフトウェア 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。

# ABORT

ユーザ プログラムを強制停止します

形式:

ABORT

実行中のユーザ プログラムを強制的に停止します。

引数:

例:

GO

(ユーザ プログラムを実行します)

ABORT

(ユーザ プログラムを強制停止します)

補足:

- 「第八章 [エミュレーション\(E\) ~ 中断\(B\)](#)」参照。

## BATCH

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

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。
- 「第八章 [ファイル\(F\) ~ バッチ\(B\)...](#)」参照。

## 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△[{S W}]	設定されているブレークポイントを表示します。
BP/C△{n Wn S* W* *}	ブレークポイントを解除します。
BP/D△{n *}	命令ソフトウェア ブレークポイントを無効にします。
BP/E△{n *}	命令ソフトウェア ブレークポイントを有効にします。

引数:

S	命令ソフトウェア ブレークポイント指定。
W	ウォッチポイント指定。 S、W省略時は、すべてのブレークポイント指定。
n	命令ソフトウェア ブレークポイントの ID 番号指定。(10 進数)
Wn	ウォッチポイントの ID 番号指定。(n=0 または n=1)
S*	すべての命令ソフトウェア ブレークポイント指定。
W*	すべてのウォッチポイント指定。
*	すべて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△S	(命令ソフトウェア ブレークポイントを表示します)
BP/C△*	(すべてのブレークポイントを解除します)
BP/C△W*	(すべてのウォッチポイントを解除します)
BP/D△5	(番号が 5 の命令ソフトウェア ブレークポイントを無効にします)
BP/E△*	(すべての命令ソフトウェア ブレークポイントを有効にします)

補足:

- 「第八章 [設定\(S\) ~ ブレークポイント\(B\)...](#)」参照。

補足 1:

命令ソフトウェア ブレークポイントは、次の形式で表示されます。

```
[Software]
ID Sts State Addr
0 ENA ARM 00001000
1 DIS ARM 00002000
2 ENA ARM 00003000
3 ENA Thumb 00004000
```

表示される各フィールドの意味は、次のとおりです。

<b>ID</b>	ブレークポイントの ID 番号。(0 から始まる 10 進数)
<b>Sts</b>	ブレークポイントの有効/無効。 <b>ENA</b> 有効 <b>DIS</b> 無効
<b>State</b>	ブレークポイントのステート。 <b>ARM</b> ARM ステート <b>Thumb</b> Thumb ステート
<b>Addr</b>	ブレークポイント アドレス。(16 進数)

補足 2:

ウォッチポイントは、次の形式で表示されます。

[Watchpoint - NORMAL]

ID	Type	Addr	AddrMask	Data	DataMask	RW	Siz	State	Mode	DBGEXT
W0	INST	00020000	00000000	00000000	FFFFFFFF	--	--	ARM	USER	----
W1	DATA	00050000	00000000	0000FF00	000000FF	R	16	-----	PRIV	----

表示される各フィールドの意味は、次のとおりです。「---」は未定義フィールド、またはタイプにより指定できないフィールドはをを表します。

ID	ウォッチポイントの番号。	
	W0	ウォッチポイント 0
	W1	ウォッチポイント 1
Type	ウォッチポイント タイプ。	
	INST	命令指定ウォッチポイント
	DATA	データ指定ウォッチポイント
Addr	ウォッチポイント アドレス。(16 進数)	
AddrMask	ウォッチポイント アドレスに対するマスク。無効ビットを1、有効ビットを0で指定。(16 進数)	
Data	ウォッチポイント データ。(16 進数)	
DataMask	ウォッチポイント データに対するマスク。無効ビットを1、有効ビットを0で指定。(16 進数)	
RW	データ アクセス方法指定。	
	R	データ読み込み指定
	W	データ書き込み指定
Siz	データ アクセスサイズ指定。	
	8	8 ビット アクセス指定
	16	16 ビット アクセス指定
	32	32 ビット アクセス指定
State	ステート指定。	
	ARM	ARM ステート指定
	Thumb	Thumb ステート指定
Mode	モード指定	
	USER	ユーザ モード指定
	PRIV	特権モード指定

DBGEXT	DBGEXT0、または DBGEXT1 信号の状態指定。信号名はウォッチポイント番号に依存。(ウォッチポイント 1 の場合は、DBGEXT1)
LOW	LOW 指定
HIGH	HIGH 指定

また、タイトルの意味は、次のとおりです。

[Watchpoint - NORMAL]	通常モード。ウォッチポイント 0 とウォッチポイント 1 を独立したブレイクポイントとして使用します。
[Watchpoint - CHAIN]	チェーン モード。ウォッチポイント 1 からウォッチポイント 0 へ移行したときブレイクする、ひとつのブレイクポイントとして使用します。
[Watchpoint - RANGE]	範囲モード。ウォッチポイント 0 とウォッチポイント 1 で指定した範囲内でブレイクする、ひとつのブレイクポイントとして使用します。

## 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[,{ARM|Thumb}]` 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)  
 ARM        ARM ステート メモリ領域指定。  
 Thumb      Thumb ステート メモリ領域指定。  
             ARM、Thumb 省略時は、現在のステートと同じ指定。  
             頭文字1文字でも指定可能。(A は ARM と同じ)

例:

<code>BP/S△1000</code>	(現在とステートと同じメモリ領域のアドレス 0x1000 番地に命令ソフトウェア ブレークポイントを設定します)
<code>BP/S△2000,ARM</code>	(ARM ステート メモリ領域のアドレス 0x2000 番地に命令ソフトウェア ブレークポイントを設定します)
<code>BP/S△3000,T</code>	(Thumb ステート メモリ領域のアドレス 0x3000 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 領域に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。
- 命令ソフトウェア ブレークは、未定義命令を使用します。
- 「第八章 [設定\(S\) ~ ブレークポイント\(B\)...](#)」参照。



## BP/Wn

ウォッチポイントを設定します

形式:

BP/Wn△INST, [A=addr] △ [AM=amask] △ [D=data] △ [DM=dmask] [, /opt] [, /opt]...

命令ウォッチポイントを設定します。

BP/Wn△DATA, [A=addr] △ [AM=amask] △ [D=data] △ [DM=dmask] [, /opt] [, /opt]...

データ ウォッチポイントを設定します。

引数:

<b>Wn</b>	ウォッチポイント ID 番号指定。
<b>W0</b>	ウォッチポイント 0 指定
<b>W1</b>	ウォッチポイント 1 指定
<b>INST</b>	命令ウォッチポイント指定。
<b>DATA</b>	データウォッチポイント指定。
<b>A=addr</b>	ウォッチポイント アドレス。(16 進数)
<b>AM=amask</b>	ウォッチポイント アドレスに対するマスク。無効ビットを 1、有効ビットを 0 で指定。 (16 進数)
<b>D=data</b>	ウォッチポイント データ。(16 進数)
<b>DM=dmask</b>	ウォッチポイント データに対するマスク。無効ビットを 1、有効ビットを 0 で指定。 (16 進数)
<b>/opt</b>	命令ウォッチポイント オプション指定。繰り返し、順不同で指定できます。
<b>ARM</b>	ARM ステート指定
<b>THUMB</b>	Thumb ステート指定
<b>USER</b>	ユーザ モード指定
<b>PRIV</b>	特権モード指定
<b>LOW</b>	DBGEXT0 または DBGEXT1 信号 LOW 指定
<b>HIGH</b>	DBGEXT0 または DBGEXT1 信号 HIGH 指定
<b>/opt</b>	データ ウォッチポイント オプション指定。繰り返し、順不同で指定できます。
<b>READ</b>	データ読み込み指定
<b>WRITE</b>	データ書き込み指定
<b>8</b>	8 ビット アクセス指定
<b>16</b>	16 ビット アクセス指定
<b>32</b>	32 ビット アクセス指定
<b>USER</b>	ユーザ モード指定
<b>PRIV</b>	特権モード指定
<b>LOW</b>	DBGEXT0 または DBGEXT1 信号 LOW 指定



## BP/WC

ウォッチポイントの連携モードを設定します

形式:

BP/WC△{NORMAL|CHAIN|RANGE}

ウォッチポイントの連携モードを設定します。

引数:

NORMAL 通常モード指定。

CHAIN チェーンモード指定。

RANGE 範囲モード指定。

例:

BP/WC△CHAIN (ウォッチポイント連携モードをチェーンモードに指定します)

BP/WC△NORMAL (ウォッチポイント連携モードを通常モードに指定します)

補足:

- 「第八章 [設定\(S\) ~ ブレークポイント\(B\)...](#)」参照。

## 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:

- 「第八章 [表示\(V\) ～ コマンド応答クリア\(C\)](#)」参照。

# 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変更可能なMJXDEBWの環境設定を表示します。
CONFIG/S <i>item=value</i>	MJXDEBWの環境設定項目 <i>item</i> を <i>value</i> に設定します。 変更は直ちに行われ、初期化が必要な項目は初期化が行われます。
CONFIG/M <i>item=value</i>	MJXDEBWの環境設定項目 <i>item</i> を <i>value</i> に設定します。 初期化が不要な項目は直ちに設定が行われます。 初期化が必要な項目は <i>value</i> の設定は保留されます。 このときは保留された <i>item</i> の設定を実行するために別途 INIT コマンドを実行する必要があります。

引数:

<i>item</i>	環境設定項目、次のいずれかを指定
JTAG_DETECT	検出モード
JTAG_ORDER	対象デバイス順位
IRLEN	IR 長
PRE_DEVICE	前段チェインデバイス数
PRE_IRLEN	前段チェイン総 IR 長
PST_DEVICE	後段チェインデバイス数
PST_IRLEN	後段チェイン総 IR 長
INV_TDO	TDO クロック反転
JCLOCK	JTAG クロック
JCLK_SEL	起動時 JTAG クロック周波数
SRCTIME	探索回数
CHKTIME	確認回数
INIT_RESERCH	init コマンドで再探索する
RST_HOLD	リセット保持時間
RST_RCV	リセット後待ち時間

JTAG\_TOUT JTAG タイムアウト時間

*value* 環境設定項目に設定する値

JTAG\_DETECT の場合

MAN IR 長／前段・後段チェーンデバイス数・総 IR 長によりデバッグ対象 CPU を指定

AUTO JTAG チェインに接続されているデバイス数とそれぞれの IR 長を測定し、対象デバイス順位のための指定でデバッグ対象 CPU を指定

JTAG\_ORDER の場合

1 から 16 デバッグ対象 CPU が JTAG チェイン中の何番目のデバイスかを指定

IRLEN の場合

4 から 16 デバッグ対象 CPU の JTAG インストラクションレジスタ (IR) のビット長を指定

PRE\_DEVICE の場合

0 から 15 JTAG チェインに複数のデバイスが接続されている場合、デバッグ対象 CPU よりも前方 (TDI 側) に接続されているデバイスの総数を指定

PRE\_IRLEN の場合

0 から 255 JTAG チェインに複数のデバイスが接続されている場合、デバッグ対象 CPU よりも前方に接続されている各デバイスの IR 長の総和を指定

PST\_DEVICE の場合

0 から 15 JTAG チェインに複数のデバイスが接続されている場合、デバッグ対象 CPU よりも後方 (TDO 側) に接続されているデバイスの総数を指定

PST\_IRLEN の場合

0 から 255 JTAG チェインに複数のデバイスが接続されている場合、デバッグ対象 CPU よりも後方に接続されている各デバイスの IR 長の総和を指定

INV\_TDO の場合

OFF TCK 立ち上がり(標準)  
ON TCK 立ち下がり

JCLOCK の場合

1 から 4096 1~4096指定

JCLK\_SEL の場合

DIV クロック分周比指定  
AUTO 自動最高探索指定  
RTCK RTCK 同期指定 (ARM9E-S コアのみ)

SRCTIME の場合

1 から 256 自動探索する回数を指定

CHKTIME の場合

1 から 256 自動探索時に、試行周波数で確実に動作するかを確認する回数を指定

INIT\_RESERCH の場合

OFF init コマンドで再探索しない  
ON init コマンドで再探索する

RST\_HOLD の場合

1 から 60000 MJXDEBW プログラムがターゲット リセット時に JTAG システムリセット信号を保持する時間を指定 (ms 単位)

RST\_RCV の場合

1 から 60000 リセット後にデバッガ動作を開始するまでの待ち時間を指定 (ms 単位)

JTAG\_TOUT の場合

1 から 60000 JTAG 動作時にタイムアウトエラーと判定する時間を指定 (ms 単位)

例:

CONFIG (MJXDEBW の環境を表示します)

補足:

- 「第八章 [設定\(S\) ~ コンフィグレーション\(C\) ~ 設定\(S\)...](#)」参照。

## DUMP

メモリの内容を表示します

形式:

DUMP[/{B|W|L}] [*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

*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)

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。
- ビット指定省略時は、前回のサイズが適用されます。
- 「第八章 [表示\(V\) ~ メモリ\(M\)...](#)」参照。



# EXAMINE

メモリの内容を変更します

形式:

EXAMINE/{B|W|L} *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
<i>addr</i>	メモリ変更開始アドレス (16進数)
<i>data1, data2</i>	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:

- 「第八章 [表示\(V\) ~ メモリ\(M\)...](#)」参照。

# FILL

メモリの内容をフィルします

形式:

`FILL/{B|W|L} 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:

`/B`      8 ビット指定  
`/W`      16 ビット指定  
`/L`      32 ビット指定  
*addr1*    メモリ フィル開始アドレス (16 進数)  
*addr2*    メモリ フィル終了アドレス (16 進数)  
*data*     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:

- 「第八章 [変更\(M\) ~ フィル\(F\)...](#)」参照。

## 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]  $\Delta$  *file\_name* [, [*device\_type*, *access\_bus\_width*, *offset*]]

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  
*offset*         ダウンロードファイルで指定された書き込みアドレスに、この値 (16 進数)  
               を加算したアドレスにデータが書き込まれます (省略時は = 0)

例:

FLASH/WRITE $\Delta$ 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 デバイスへ 2 バイト単位で書き込む)

FLASH/ERASEONE $\Delta$ 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 G 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 H 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 [エミュレーション\(E\) ～ フラッシュメモリ\(F\) ～ 書込\(W\)](#)」参照。
- 「第八章 [エミュレーション\(E\) ～ フラッシュメモリ\(F\) ～ 消去\(E\)](#)」参照。

# GO

ユーザ プログラムを実行します

形式:

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

引数:

*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:

- ユーザ プログラムがブレークポイントなどで停止しない場合、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。)
- 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。
- 「付録 K [ARM ステートと Thumb ステートの指定方法](#)」参照。
- 「第八章 [エミュレーション\(E\) ~ 実行\(G\)...](#)」参照。
- 「第八章 [エミュレーション\(E\) ~ 再スタート\(R\)](#)」参照。

## HISTORY\*1

リアルタイム トレース結果を表示します

形式:

HISTORY	リアルタイム トレース結果の有効なサイクル番号の範囲を表示します。
HISTORY/P△[ <i>start</i> [, <i>count</i> ]]	サイクル番号 <i>start</i> から <i>count</i> 数分のリアルタイム トレース結果を、サイクル形式で表示します。
HISTORY/A△[ <i>start</i> [, <i>count</i> ]]	サイクル番号 <i>start</i> から <i>count</i> 数分のリアルタイム トレース結果を、サイクル形式と命令アドレスで表示します。
HISTORY/D△[ <i>start</i> [, <i>count</i> ]]	サイクル番号 <i>start</i> から <i>count</i> 数分のリアルタイム トレース結果を、逆アセンブル形式で表示します。
HISTORY/M△[ <i>start</i> [, <i>count</i> ]]	サイクル番号 <i>start</i> から <i>count</i> 数分のリアルタイム トレース結果を、サイクル形式と逆アセンブル形式で表示します。

引数:

<b><i>start</i></b>	表示開始サイクル番号 (10進数、省略時は前回からの継続、初期値は有効なサイクルの先頭のサイクル番号)
<b><i>count</i></b>	表示サイクル数 (10進数、省略時は前回に設定した値、初期値は 16)

例:

HISTORY	(リアルタイム トレース結果の有効なサイクル番号の範囲を表示します)
HISTORY/D△0,70	(サイクル番号 0 から 70 サイクル数分のリアルタイム トレース結果を、逆アセンブル形式で表示します)
HISTORY/P△-10,20	(サイクル番号-10 から 20 サイクル数分のリアルタイム トレース結果を、サイクル形式で表示します)

---

\*1 MJX330 for ARM はトレースメモリを持っておりませんので、このコマンドは機能致しません。

補足:

- リアルタイム トレースのトレース モードを設定する場合は、TRACE コマンドを使用してください。
- **start** は、サイクル番号 0 を起点とするオフセットで指定します。トレース条件により、サイクル番号 0 の点は、次のように異なります

ビギン モニタ	プログラム開始点
エンド モニタ	プログラム終了点(トレース中止点)
ビギン トリガ	トリガ点
エンド トリガ	トリガ点
ミッド トリガ	トリガ点
レンジ	最初のトレース有効範囲命令実行点
カスタム	トレースバッファメモリ先頭サイクル

- 命令アドレス形式表示はトレース設定において全分岐モードが有効な場合のみ表示可能です。
- 命令アドレス形式で表示されるアドレスのうち、オーバフロー直前およびブレイクポイント直前のアドレスが正しく表示されない場合があります。(トレース ポート幅 16-bit 場合のみ)
- サイクル形式および命令アドレス形式はトレースが終了していれば、ユーザ プログラムが実行中でもトレース結果を表示することができます。
- 「第八章 [表示\(V\) ~ トレース結果\(T\)...\\*1](#)」参照。

# 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。
- 「第八章 [エミュレーション\(E\) ~ 初期化\(I\)](#)」参照。



# 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:

- /W を指定し、かつ同名のファイルが存在する場合、そのファイルの内容は失われます。
- 「第八章 [ファイル\(F\) ~ ジャーナル開始\(S\)...](#)」参照。
- 「第八章 [ファイル\(F\) ~ ジャーナル停止\(E\)](#)」参照。

## 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\*<sup>1</sup>、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。

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)

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。
- 「第八章 [エミュレーション\(E\) ~ ダウンロード\(L\)...](#)」参照。

---

\*<sup>1</sup> 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。

## MOVE

メモリの内容をブロック転送します

形式:

MOVE[/{B|W|L}] *addr1*,*addr2*,*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

*addr1* 転送元メモリ開始アドレス (16進数)

*addr2* 転送元メモリ終了アドレス (16進数)

*addr3* 転送先メモリ アドレス (16進数)

例:

MOVE/B△1000,10FF,2000 (0x1000 から 0x10FF 番地までのメモリをバイト単位で 0x2000 番地へブロック転送します)

補足:

- ビット指定省略時は、8ビット指定が適用されます。
- 転送元領域と転送先領域が重なっている場合の結果は不定です。
- 「第八章 [変更\(M\) ~ 移動\(M\)...](#)」参照。

## 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。

## 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△*reg*=*data*   レジスタ *reg*をデータ *data*に変更します。

引数:

*reg*     レジスタ名指定  
          「付録 D [レジスタ名一覧](#)」を参照してください  
*data*    レジスタ変更データ (16 進数)

例:

REGISTER                   (すべてのレジスタの内容を表示します。)  
 REGISTER△PC=2000       (PC を 0x2000 に変更します)  
 REGISTER△R3=4000       (R3 を 0x4000 に変更します)

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。
- R13レジスタは LR、R14レジスタは SP の別名でも指定できます。
- 「第八章 [表示\(V\) ~ レジスタ\(R\)](#)」参照。

## SETLOAD

ブレイクポイントおよびトレースの設定をファイルから読み出します

形式:

SETLOAD△ *filename*      ブレイクポイントおよびトレースの設定をファイルから読み出します。

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:

- 「第八章 [設定\(S\) ～ 設定の読出\(L\)...](#)」参照。

## SETSAVE

ブレイクポイントおよびトレースの設定をファイルへ保存します

形式:

SETSAVE△ *filename*

ブレイクポイントおよび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:

- 同じ名前のファイルが存在する場合、上書きします。
- 「第八章 [設定\(S\) ～ 設定の保存\(S\)...](#)」参照。

## 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。
- 「第八章 [エミュレーション\(E\) ～ ステップ\(S\)](#)」参照。
- 「第八章 [エミュレーション\(E\) ～ N-ステップ\(N\)...](#)」参照。





## TRACE/C<sup>\*1</sup>

リアルタイム トレースのカウンタ条件を設定/解除します

形式:

TRACE/CN△ <i>count</i>	カウント値を <i>count</i> に設定します。
TRACE/C△{CNT REL}	カウンタ イベント、またはカウンタ再ロード イベントを解除します。
TRACE/C△{CNT REL}, I, {M0 M1 M2}, <i>addr</i>	命令実行時に発生させるカウンタ イベント、またはカウンタ再ロード イベントを設定します。
TRACE/C△{CNT REL}, {A W R}, {M0 M1 M2}, <i>addr</i> [, <i>data</i> [, <i>datamask</i> ]]	データ アクセス時に発生させるカウンタ イベント、またはカウンタ再ロード イベントを設定します。
TRACE/CR△{CNT REL}, I, {M0 M1 M2}, <i>saddr</i> , <i>eaddr</i>	指定アドレス範囲の命令実行時に発生させるカウンタ イベント、またはカウンタ再ロード イベントを設定します。
TRACE/CR△{CNT REL}, {A W R}, {M0 M1 M2}, <i>saddr</i> , <i>eaddr</i> [, <i>data</i> [, <i>datamask</i> ]]	データ アクセス時に発生させるカウンタ イベント、またはカウンタ再ロード イベントを設定します。

引数:

<i>count</i>	カウント値指定 (10 進数)
CNT	カウンタ イベント指定
REL	カウンタ再ロード イベント指定
I	命令実行時条件指定
M0	アドレス マスク指定なし
M1	アドレス マスク A0 指定
M2	アドレス マスク A0&A1 指定
<i>addr</i>	命令実行アドレス、またはデータ アクセス アドレス指定 (16 進数)
A	データ読み込み、書き込み条件指定
W	データ書き込み条件指定
R	データ読み込み条件指定

\*1 MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。



## TRACE/D\*1

リアルタイム トレースのデータ トレース条件を設定/解除します

形式:

TRACE/D△{A|W|R}, {MO|M1|M2}, {IN|EX}, *addr*[, *data*[, *datamask*]]

指定アドレスのアクセス時にトレースさせる、データ トレース条件を設定します。

TRACE/DR△{A|W|R}, {MO|M1|M2}, {IN|EX}, *saddr*, *eaddr*[, *data*[, *datamask*]]

指定アドレス範囲のアクセス時にトレースさせる、データ トレース条件を設定します。

TRACE/DC△{*index*|\*}

データ トレース条件を解除します。

TRACE/DP△{A|D|AD}

データ トレース出力条件を指定します。

引数:

A	データ読み込み、書き込み条件指定
W	データ書き込み条件指定
R	データ読み込み条件指定
MO	アドレス マスク指定なし
M1	アドレス マスク A0 指定
M2	アドレス マスク A0&A1 指定
IN	指定アドレス(範囲)内指定
EX	指定アドレス(範囲)外指定
<i>addr</i>	データ アクセス アドレス指定 (16 進数)
<i>data</i>	アクセス データ指定 (16 進数)
<i>datamask</i>	アクセス データ マスク指定 (16 進数、マスク ビットを 1 で指定)
<i>saddr</i>	範囲先頭アドレス指定 (16 進数)
<i>eaddr</i>	範囲終了アドレス指定 (16 進数)
<i>index</i>	データ トレース条件のインデックス指定 (1 で始まる 10 進数、TRACE コマンドで表示される)
*	すべてのデータ トレース条件指定
A	アドレスのみトレース出力
D	データのみトレース出力
AD	アドレスとデータをトレース出力

\*1 MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。

例:

TRACE/DR△A,M2,IN,1000,2000

(0x1000 番地から 0x2000 番地未満のアドレスに対するアクセスをトレース対象にします。)

TRACE/D△M2,EX,1200

(0x1200 番地に対する書き込みアクセスをトレース対象から除外します。)

TRACE/DP△AD

(トレース対象となったアクセスのアドレスとデータをトレース情報に出力します。)

TRACE/DC△\*

(データ トレースをすべて解除します。)

補足:

- 「第八章 [設定\(S\) ～ トレース\(T\)...](#)」参照。

TRACE/E<sup>\*1</sup>

リアルタイム トレースのトレース モードを設定します

形式:

TRACE/E△{FULL|REAL}, {CAC|NOCAC}, {ALL|IND}, {CPRT|NOCPRT}, D=*depth*

トレース モードを設定します。

引数:

FULL    トレース モードを完全モードに設定します。(完全モード ON)  
 REAL    トレース モードをリアル モードに設定します。(完全モード OFF)  
 CAC    Cycle-accurate Tracing モードを設定します。(サイクル精度 ON)  
 NOCAC    Cycle-accurate Tracing モードを解除します。(サイクル精度 OFF)  
 ALL    全分岐トレースを設定します。(全分岐トレース ON)  
 IND    全分岐トレースを解除します。(全分岐トレース OFF)  
 CPRT    CPRTトレースを設定します。(CPRTトレース ON)  
 NOCPRT    CPRTトレースを解除します。(CPRTトレース OFF)  
*depth*    FIFO レベル指定

補足:

- 完全モードでは、ETMトレースFIFO 残容量がFIFOレベル指定以下になった場合、CPU コアに対して FIFOFULL 信号を出力し、CPU をストールさせます。  
CPU の実装により完全モードが使用できない場合があります。
- Cycle-Accurate Tracing モードでは、不要なトレース データのトレース バッファへの書き込み停止を抑制します。換言すると、すべてのトレース サイクルでトレース データがトレース バッファへ書き込まれます。Cycle-Accurate Tracing モードでは、トレース可能な命令数が減りますが、トレース バッファ インデックスを数えることにより、プログラム実行時間が計測できます。
- 通常の場合、ETMは間接分岐命令に対してのみ分岐情報をトレース出力します。全分岐トレース モードでは直接/間接を問わずすべての分岐命令に対して分岐情報をトレース出力します。全分岐トレース モードを指定すると、ユーザプログラム実行中でもパケット形式および命令アドレス形式のトレース結果表示が行なえます。
- CPRTトレースON設定では、CPUレジスタとコプロセッサ レジスタ間データ転送がデータトレース対象となります。この設定は TRACE/D コマンドによるデータ トレース設定とは無関係に設定可能です。
- 「第八章 [設定\(S\) ~ トレース\(T\)...](#)」参照。

\*1 MJX はトレースメモリを持っておりません。外部装置によりETM信号をモニターする場合にご使用ください。



TRACE/I△RA, *length*, *saddr1-eaddr1*[, *saddr2-eaddr2*]...

命令トレース条件を、命令実行アドレスの範囲で指定します。

TRACE/I△CU

カスタム設定コマンド(TRACE/P)を有効にします。

引数:

BM	ビギン モニタ指定
EM	エンド モニタ指定
BT	ビギン トリガ指定
ET	エンド トリガ指定
MT	ミッド トリガ指定
RA	レンジ指定
<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	命令実行時トリガ指定
M0	アドレス マスク指定なし
M1	アドレス マスク A0 指定
M2	アドレス マスク A0&A1 指定
<i>addr</i>	命令実行アドレス、またはデータ アクセス アドレス指定 (16 進数)
A	データ 読みこみ、書きこみ条件指定なし
W	データ書きこみ条件指定
R	データ読みこみ条件指定
<i>data</i>	アクセス データ指定 (16 進数)
<i>datamask</i>	アクセス データ マスク指定 (16 進数、マスク ビットを 1 で指定)
<i>saddr</i>	範囲先頭アドレス指定 (16 進数)
<i>eaddr</i>	範囲終了アドレス指定 (16 進数)
CTR	カウンタ トリガ指定
SEQ1	シーケンサ 1 トリガ指定
SEQ2	シーケンサ 2 トリガ指定
SEQ3	シーケンサ 3 トリガ指定

例:

補足:

- トレース条件の詳細な説明は、「第八章 [設定\(S\) ~ トレース\(T\)...](#)」を参照してください。



## TRACE/P\*1

リアルタイム トレースのトレース条件をカスタム設定します

注意: 「TRACE/P」コマンドを実行する前には、必ず「TRACE/I CU」コマンドを実行してください。

形式 (一覧):

TRACE/P△MJX, ...	トリガー モードを設定します。(MJX)
TRACE/P△ECR, ...	ECR レジスタを設定します。
TRACE/P△TRE, ...	トリガー イベントを設定します。
TRACE/P△TEE, ...	TraceEnable イベントを設定します。
TRACE/P△TEC, ...	TraceEnable コントロール 1~3 を設定します。
TRACE/P△FIFO, ...	FIFOFULL 領域および FIFOFULL レベルを設定します。
TRACE/P△VDE, ...	ViewData イベントを設定します。
TRACE/P△VDC, ...	ViewData コントロール 1~3 を設定します。
TRACE/P△ACV, ...	アドレス コンパレータ値を設定します。
TRACE/P△AAT, ...	アドレス コンパレータ タイプを設定します。
TRACE/P△DC, ...	データ コンパレータ値を設定します。
TRACE/P△ICV, ...	カウンタ初期値を設定します。
TRACE/P△CEE, ...	カウンタ イネーブル イベントを設定します。
TRACE/P△CRE, ...	カウンタ再ロード イベントを設定します。
TRACE/P△SQE, ...	シーケンサ遷移イベントを設定します。
TRACE/P△EOE, ...	外部出力イベントを設定します。
TRACE/P△ISR, ...	Implementation Register を設定します。

形式と引数:

TRACE/P△MJX, {ON|OFF}, {TRG|AUTO}, *length*

ON	指定
OFF	指定
TRG	指定
AUTO	指定
<i>length</i>	指定

\*1 MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。

(プロトコルバージョン 0 または 1 の場合)

TRACE/P△ECR, {FULL|REAL}, {CAC|NOCAC}, {ALL|IND}, {CPRT|NOCPRT}  
 , {DBGQRQ|NODBGRQ}, {NODATA|DATA|ADDR|D\_AND\_A}

(ETM プロトコルバージョン 2 以降の場合)

TRACE/P△ECR, {FULL|REAL}, {CAC|NOCAC}, {ALL|IND}, {CPRT|NOCPRT}  
 , {DBGQRQ|NODBGRQ}, {NODATA|DATA|ADDR|D\_AND\_A}  
 , {ID0|ID8|ID16|ID32}

TRACE/P△TRE, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

*condition*:

{A|NOTA|A\_AND\_B|NOTA\_AND\_B|NOTA\_AND\_NOTB|A\_OR\_B|NOTA\_OR\_B|NOTA\_OR\_NO  
 TB}

*Aresource*, *Bresource*: {ACMP|ARNG|E IWP|MMD|CTR|SEQ|EXTI}

*Anum*, *Bnum*: リソース番号 (10 進数)

TRACE/P△TEE, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

TRACE/P△TEC, {IN|EX}, MMD=*bits*, ARC=*bits*

TRACE/P△TEC, {ONOFF|NOONOFF}, {IN|EX}, MMD=*bits*, ARC=*bits*, AC=*bits*  
 , ON=*bits*, OFF=*bits*

*bits*:ビット設定値

TRACE/P△FIFO, {IN|EX}, MMD=*bits*, ARC=*bits*, LVL=*num*

*bits*:ビット設定値

*num*:FIFOFULL レベル (10 進数)

TRACE/P△VDE, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

TRACE/P△VDC, {EXONLY|NOEXONLY}, ACIN=*bits*, ACEX=*bits*, ARCIN=*bits*  
, ARCEX=*bits*, MMDIN=*bits*, MMDEX=*bits*

*bits*:ビット設定値

TRACE/P△ACV, *num*, *value* アドレス コンパレータ値を設定します。

*num*:アドレス コンパレータ番号 (10 進数)

*value*:設定値

TRACE/P△AAT, *num*, {NODATA|DATAENA}, {M0|M1|M2}  
, {FETCH|EXE|LOAD\_STORE|LOAD|STORE}

TRACE/P△AAT, *num*, {NODATA|MATCH|NOMATCH}, {M0|M1|M2}  
, {FETCH|EXE|EXE\_PASS|EXE\_FAIL|LOAD\_STORE|LOAD|STORE}

*num*:アドレスコンパレータ番号 (10 進数)

TRACE/P△DC, *num*, *data*, *mask*

*num*:アドレスコンパレータ番号 (10 進数) 注)データコンパレータ番号ではありません

*data*:データ設定値

*mask*:データマスク設定値

TRACE/P△ICV, *num*, *value*

*num*:カウンタ番号 (10 進数)

*value*:初期値 (10 進数)

TRACE/P△CEE, *num*, {CYCLE|EVENT}, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

TRACE/P△CRE, *num*, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

TRACE/P△SQE, {S12|S21|S23|S31|S32|S13}, *condition*, *Aresource*, *Anum*,  
*Bresource*, *Bnum*

TRACE/P△EOE, *num*, *condition*, *Aresource*, *Anum*, *Bresource*, *Bnum*

TRACE/P△ISR, *num*, *value*

*num*:レジスタ番号 (10 進数)

*value*:設定値

例:

補足:

- 「第八章 [設定\(S\) ～ トレース\(T\)...](#)」参照。

## TRACE/Q<sup>\*1</sup>

リアルタイム トレースの ETM の構成と現在の状態を表示します

形式:

TRACE/Q

ETM の構成と現在の状態を表示します。

例:

TRACE/Q

補足:

- 「第八章 [設定\(S\) ~ トレース\(T\)...](#)」参照。

---

\*1 MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。

## TRACE/R<sup>\*1</sup>

リアルタイム トレースの全トレース条件を無効にします

形式:

TRACE/R 全トレース条件を無効にします

例:

TRACE/R

補足:

- 「第八章 [設定\(S\) ~ トレース\(T\)...](#)」参照。

---

\*1 MJX はトレースメモリを持っておりません。外部装置によりETM信号をモニターする場合にご使用ください。

## TRACE/S<sup>\*1</sup>

リアルタイム トレースのシーケンサ条件を設定/解除します

形式:

TRACE/S△{S12|S21|S23|S31|S32|S13}

シーケンサ条件を解除します。

TRACE/S△{S12|S21|S23|S31|S32|S13}, I, {M0|M1|M2}, *addr*

命令実行時に発生させるシーケンサ条件を設定します。

TRACE/S△{S12|S21|S23|S31|S32|S13}, {A|W|R}, {M0|M1|M2}, *addr*[, *data*[, *datamask*]]

データ アクセス時に発生させるシーケンサ条件を設定します。

TRACE/SR△{S12|S21|S23|S31|S32|S13}, I, {M0|M1|M2}, *saddr*, *eaddr*

指定アドレス範囲の命令実行時に発生させるシーケンサ条件を設定します。

TRACE/SR△{S12|S21|S23|S31|S32|S13}, {A|W|R}, {M0|M1|M2}, *saddr*, *eaddr*

[, *data*[, *datamask*]]

指定アドレス範囲のデータ アクセス時に発生させるシーケンサ条件を設定します。

引数:

S12	シーケンサ 1→2 状態遷移指定
S21	シーケンサ 2→1 状態遷移指定
S23	シーケンサ 2→3 状態遷移指定
S31	シーケンサ 3→1 状態遷移指定
S32	シーケンサ 3→2 状態遷移指定
S13	シーケンサ 1→3 状態遷移指定
I	命令実行時条件指定
M0	アドレス マスク指定なし
M1	アドレス マスク A0 指定
M2	アドレス マスク A0&A1 指定
<i>addr</i>	命令実行アドレス、またはデータ アクセス アドレス指定 (16 進数)
A	データ読みこみ、書きこみ条件指定
W	データ書き込み条件指定
R	データ読み込み条件指定

\*1 MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。





## TRACE/T<sup>\*1</sup>

リアルタイム トレースを強制停止します

形式:

TRACE/T

補足:

- リアルタイムトレースを停止させると、ユーザプログラム実行中でもサイクル形式または命令アドレス形式でトレース結果表示ができます。

---

\*1 MJX はトレースメモリを持っておりません。外部装置によりETM信号をモニターする場合にご使用ください。

## UNASM

メモリの内容を逆アセンブル表示します

形式:

UNASM△[*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:

*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。
- 「付録 K [ARM ステートと Thumb ステートの指定方法](#)」参照。
- 「第八章 [表示\(V\) ~ 逆アセンブル\(D\)...](#)」参照。

# 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

例:

VERSION

補足:

- 「第八章 [ヘルプ\(H\) ~ MJXDEBW のバージョン情報\(A\)](#)」参照。

## 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。
- バッチファイル内で使用します (GO コマンドの実行後に発行します)。

## 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。

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。
変更(M)	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。
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:

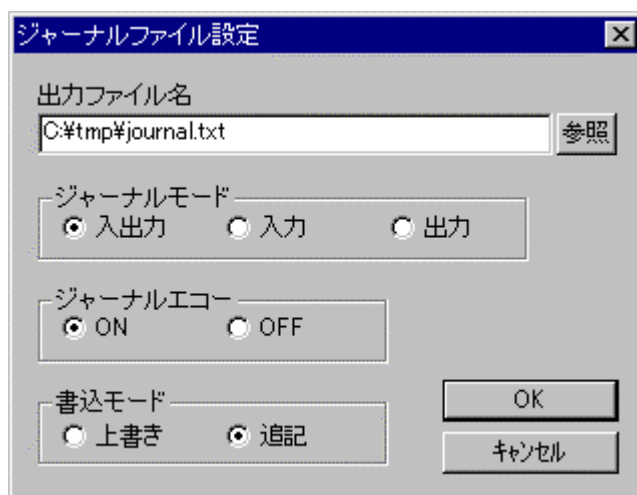
- 「第七章 [BATCH](#)」参照。

## 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)...

---

HISTORY

機能:

トレース結果をファイルへ出力します。

補足:

- MJX330 for ARM はトレースメモリを持っておりませんので、このコマンドは機能致しません。



## エミュレーション(E) ~ 実行(G)...

---

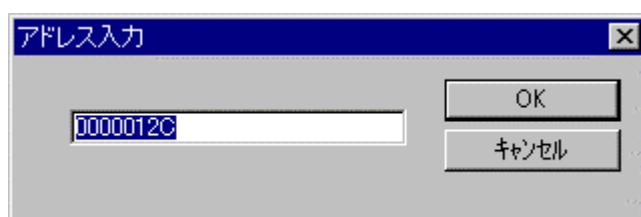
GO *addr*

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:

- 「付録 K [ARM ステートと Thumb ステートの指定方法](#)」参照。
- 「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)...

---

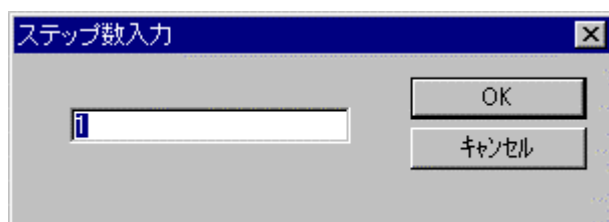
STEP num

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)...

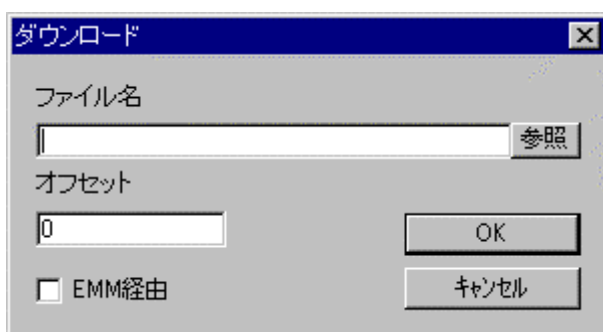
---

LOAD  
Ctrl+L

機能:

MJX バイナリ\*<sup>1</sup>、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。
- MJX330 では EMM 経由の指定はできません。
- 「第七章 [LOAD](#)」参照。

---

\*<sup>1</sup> 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:

MJX を再初期化します。

補足:

- 「第七章 [INIT](#)」参照。

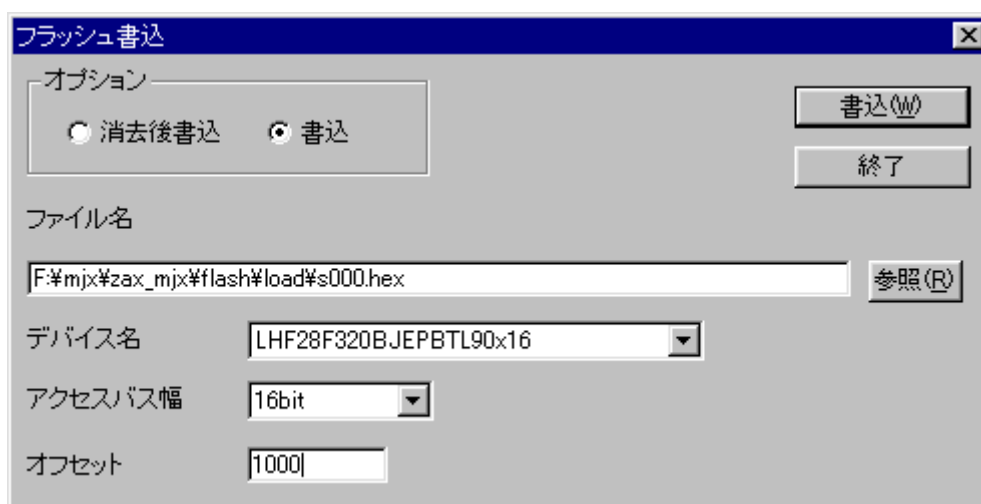
## エミュレーション(E) ~ フラッシュメモリ(F) ~ 書込(W)...

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、オフセットを指定して、「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。

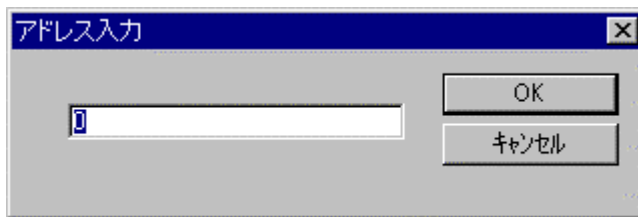
## 表示(V) ~ メモリ(M)...

DUMP

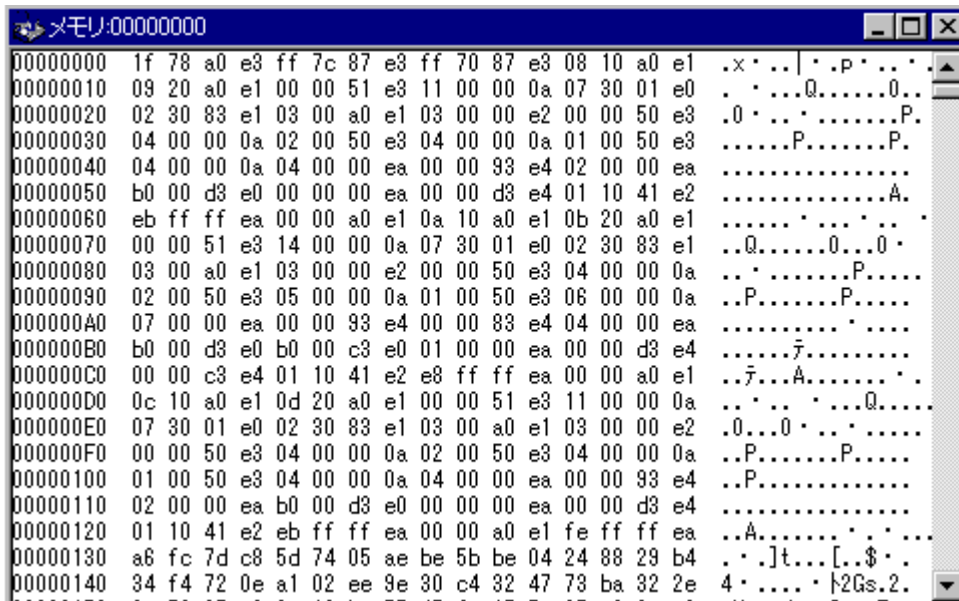
機能:

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。

バイト(B)  
 ワード(W)  
 ロング(L)  
 ダブルロング(D)  
 単精度(F)  
 倍精度(B)



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。

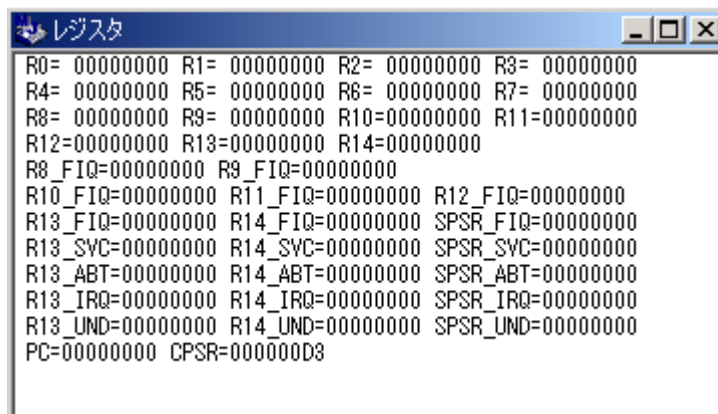
## 表示(V) ~ レジスタ(R)

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
R0= 00000000 R1= 00000000 R2= 00000000 R3= 00000000
R4= 00000000 R5= 00000000 R6= 00000000 R7= 00000000
R8= 00000000 R9= 00000000 R10=00000000 R11=00000000
R12=00000000 R13=00000000 R14=00000000
R8_FIQ=00000000 R9_FIQ=00000000
R10_FIQ=00000000 R11_FIQ=00000000 R12_FIQ=00000000
R13_FIQ=00000000 R14_FIQ=00000000 SPSR_FIQ=00000000
R13_SVC=00000000 R14_SVC=00000000 SPSR_SVC=00000000
R13_ABT=00000000 R14_ABT=00000000 SPSR_ABT=00000000
R13_IRQ=00000000 R14_IRQ=00000000 SPSR_IRQ=00000000
R13_UND=00000000 R14_UND=00000000 SPSR_UND=00000000
PC=00000000 CPSR=000000D3
```

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。

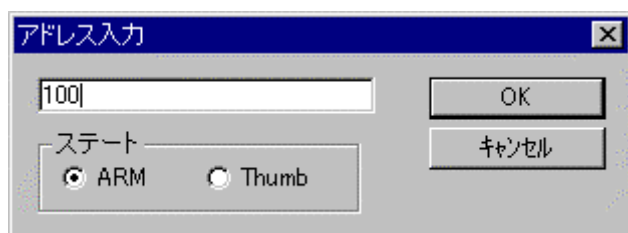
## 表示(V) ~ 逆アセンブル(D)...

UNASM

機能:

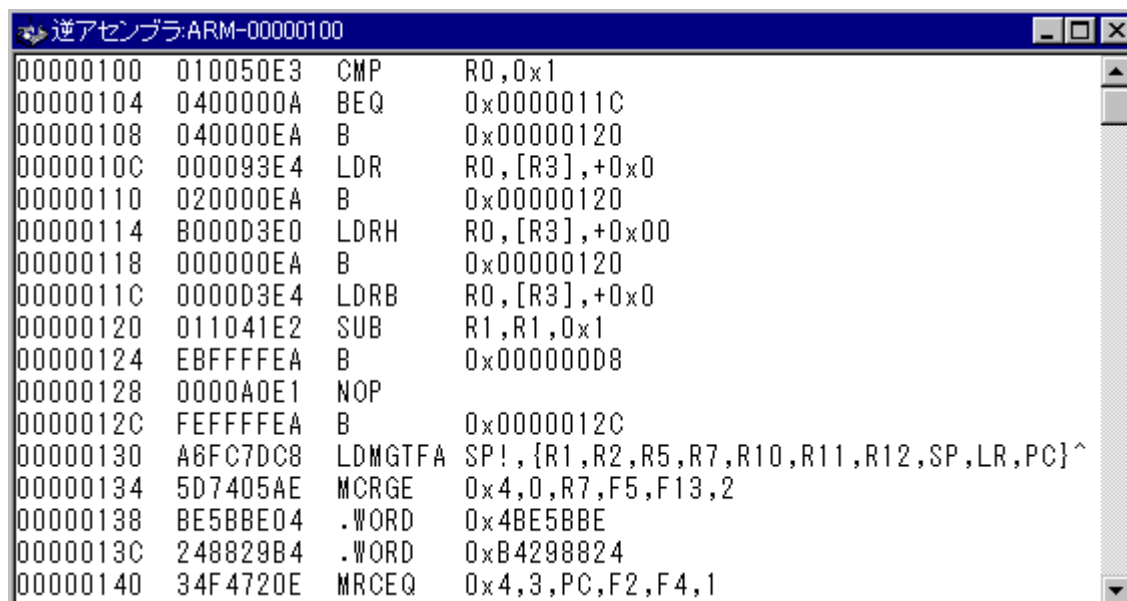
メモリの内容を逆アセンブル表示する、逆アセンブラ ウィンドウを開きます。

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)、状態を指定し、「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。



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。

補足:

- MJX330 for ARM はトレースメモリを持っておりませんので、このコマンドは機能致しません。

## 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:

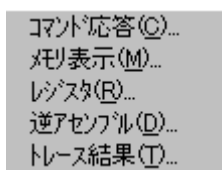
- 「第七章 [CLEAR](#)」参照。

## 表示(V) ~ オプション(O) ~ フォント(F)

機能:

ウイ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インドウを選びます。



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。

## 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。

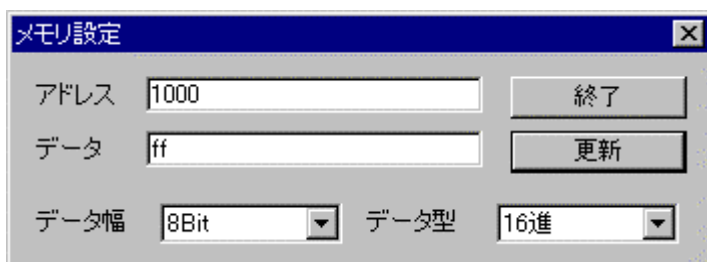
## 変更(M) ~ メモリ(M)...

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) ~ レジスタ(R)...

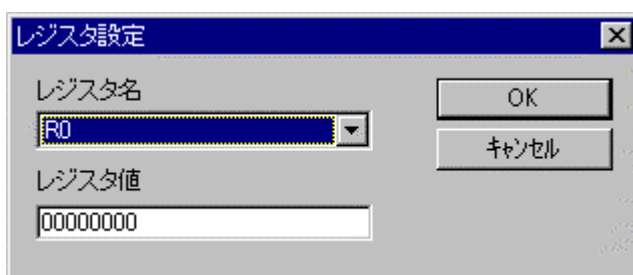
---

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。

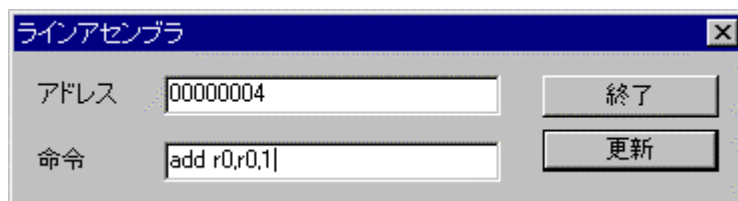
## 変更(M) ~ アセンブラ(A)...

---

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。
- 現在のステートの命令のみ受け付けます。
- 「付録 K [ARM ステートと Thumb ステートの指定方法](#)」参照。

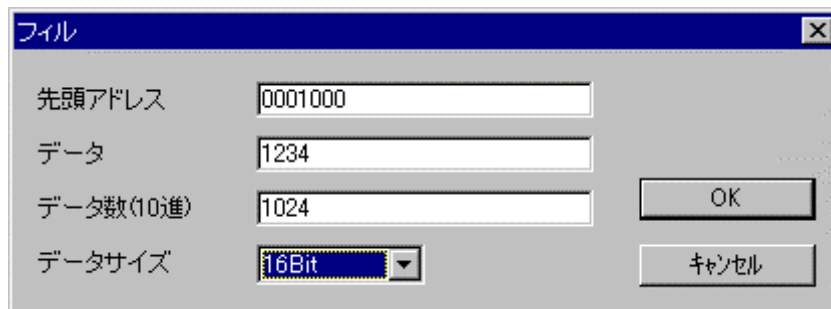
## 変更(M) ~ フィル(F)...

FILL

機能:

メモリの内容をフィルします。

操作:



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。

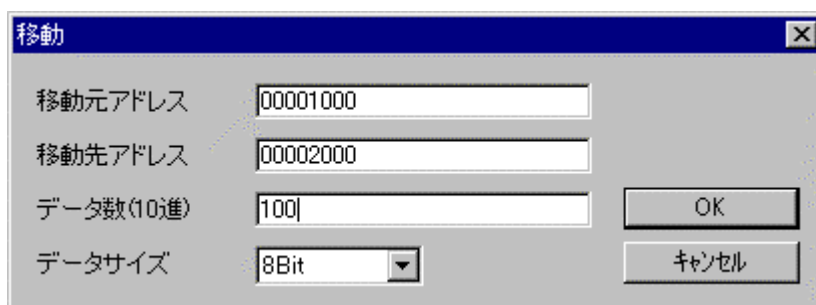
## 変更(M) ~ 移動(M)...

MOVE

機能:

メモリの内容をブロック転送します。

操作:



The screenshot shows a dialog box titled "移動" (Move). It contains the following fields and controls:

- 移動元アドレス: 00001000
- 移動先アドレス: 00002000
- データ数(10進): 100
- データサイズ: 8Bit (dropdown menu)
- Buttons: 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:

命令ソフトウェア ブレークポイントを設定します

操作:



ダイアログ ボックスのソフトウェア タブで、アドレス(16 進数)、状態を指定し、「登録」ボタンを押すと、命令ソフトウェア ブレークポイントを設定できます。

設定された命令ソフトウェア ブレークポイントは、現在の設定に表示されます。

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、ブレークポイントを無効化できます。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、ソフトウェア ブレークポイントをすべて削除します。

命令ソフトウェア 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:

- 登録、無効化、有効化、削除は、直ちにソフトウェア 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。

## 設定(S) ~ ブレークポイント(B)... ウォッチポイント

BP/Wn

機能:

ウォッチポイントを設定します。

操作:

ダイアログ ボックスのウォッチポイント0またはウォッチポイント1タブで、タイプ、アドレス(16進数)、アドレス マスク(16進数)、データ(16進数)、データ マスク(16進数)、リード/ライト、サイズ、ステート、モード、DBGEXn 信号を指定し、「OK」ボタンを押すと、ウォッチポイントを設定できます。

補足:

- マスク有効がチェックされていない場合は、マスクは 0x00000000 と見なされます。
- 「第七章 [BP/Wn](#)」参照。

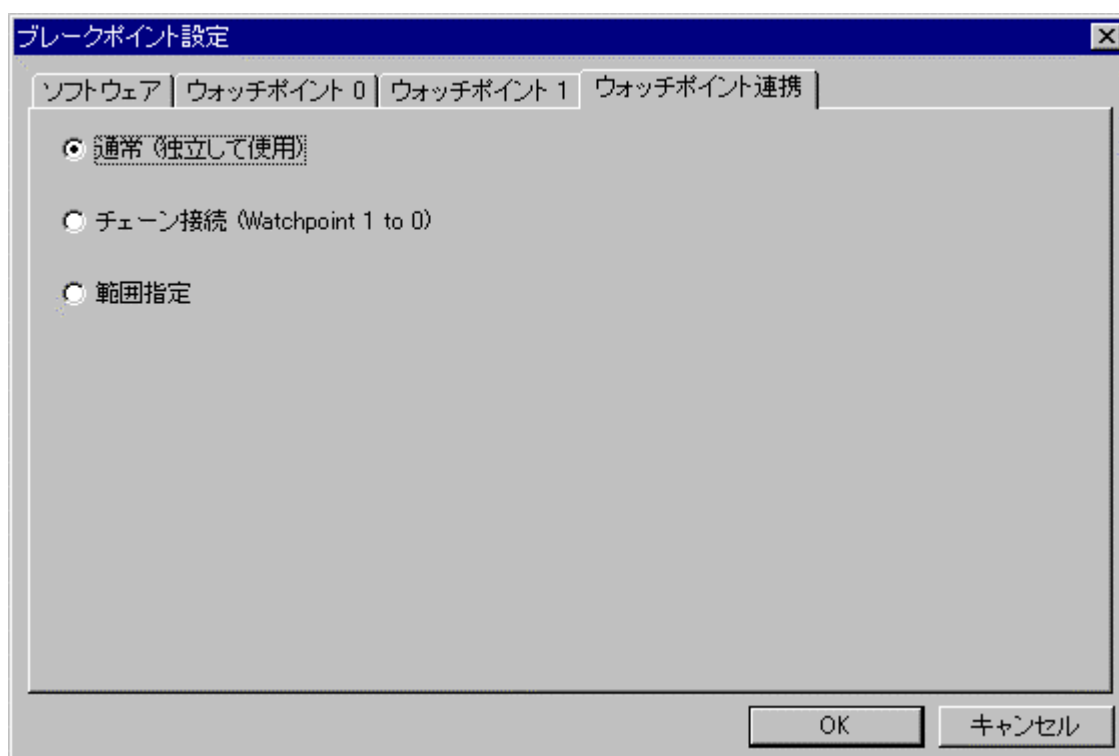
## 設定(S) ~ ブレークポイント(B)... ウォッチポイント連携

BP/WC

機能:

ウォッチポイントの連携モードを設定します。

操作:



ダイアログ ボックスで、ウォッチポイントの連携モードのひとつを選択し、「OK」ボタンを押してください。

補足:

- 「第七章 [BP/WC](#)」参照。



## 設定(S) ~ トレース(T)...\*<sup>1</sup>

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 ボックスの各ページで、トレース モードを設定し、「OK」ボタンを押してください。  
各ページで設定できるトレース モードは次のとおりです。

命令	命令トレースの条件とオプションの設定を行います。またトレース全体を有効にする設定はこのページにあります。
データ	データ トレースの条件とオプションの設定を行います。データトレースは命令トレースが有効なときのみ出力されます。命令トレースがレンジモードで無効な区間では発生しません。
カウンタ	カウンタ条件設定を行います。命令トレース設定の「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 トリガーで「トリガー ソース」がカウンタ イベントの場合のみ指定できます。
シーケンサ	シーケンサ状態遷移条件設定を行います。命令トレース設定の「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 トリガーで「トリガー ソース」がシーケンサ1/シーケンサ2/シーケンサ3の場合のみ指定できます。
ステータス	ETM のリソースの構成とステータスを表示します。

\*<sup>1</sup> MJX はトレースメモリを持っておりません。外部装置により ETM 信号をモニターする場合にご使用ください。

操作 1: 命令ページでは、命令トレースの条件とオプションの設定を行います。:

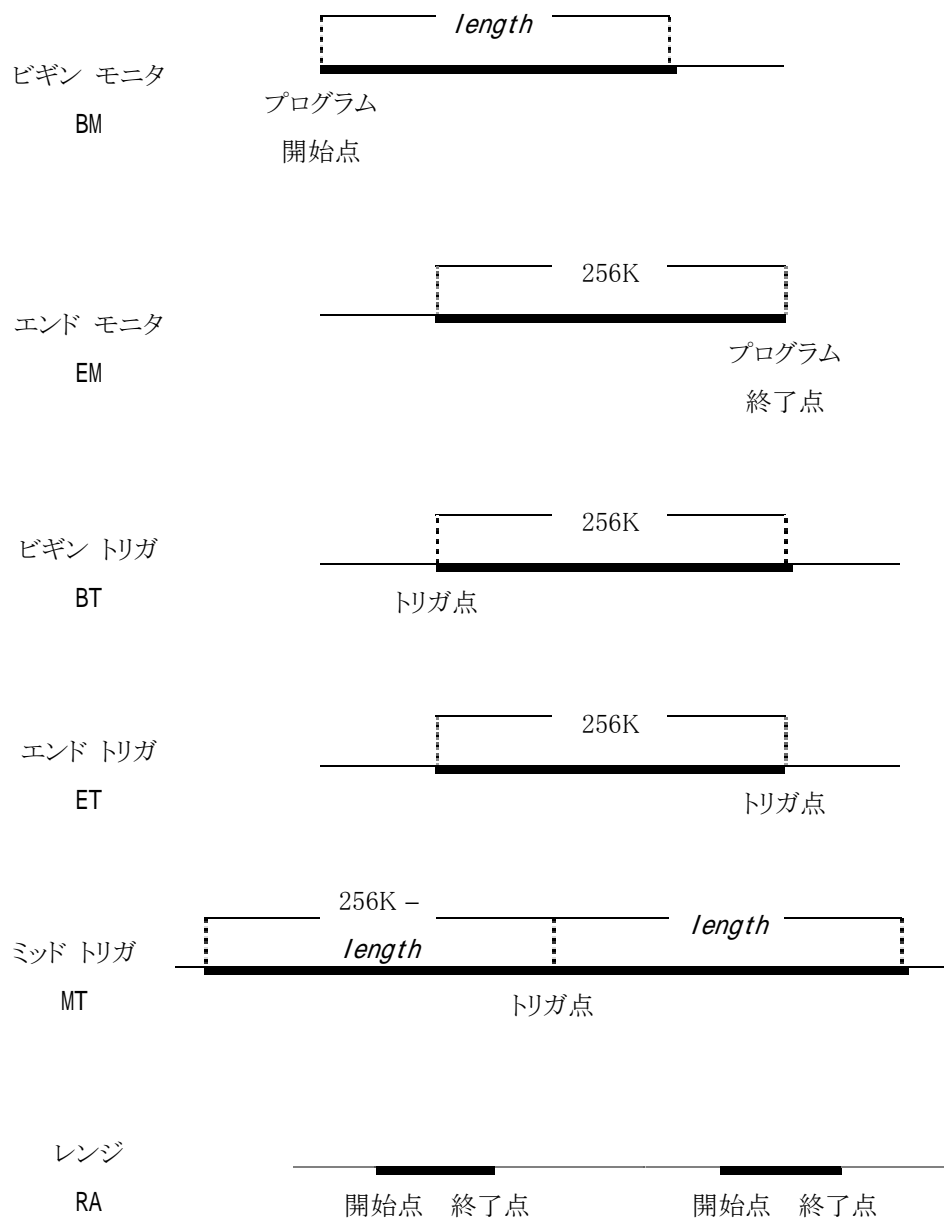


トレース有効      チェックするとトレース全体が有効になります。

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
- レンジ
- カスタム

トレース範囲:



- ビギン トリガーはミッド トリガーのレンジス=最大値と同一の意味です。エンド トリガーはミッド トリガーのレンジス=最小値と同一の意味です。
- length カウンタがカウント アップした段階でトレース バッファへの書き込みは終了します。このとき、ETMトレース FIFO にデータがあった場合、捨てられます。エンド モニタ使用時は特にご注意ください。

- カスタム設定では ETM の全機能を使用することができます。ETM レジスタレベルで設定を行います。

トリガー ソース 「トリガー ソース」は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 トリガーの場合のトリガー ポイント要因を選択します。トレース ソースは以下から選択します。

命令実行	指定のアドレスの命令実行をトリガーとします。
データ アクセス	指定のアドレスのデータアクセスをトリガーとします。アクセスデータの指定が可能です。
READ アクセス	指定のアドレスの READ データアクセスをトリガーとします。アクセスデータの指定が可能です。
WRITE アクセス	指定のアドレスの WRITE データアクセスをトリガーとします。アクセスデータの指定が可能です。
カウンタイベント	カウンタのカウンタ アップをトリガーとします。カウンタの条件設定はカウンタ設定ページで行います。
シーケンサ1	シーケンサの状態遷移をトリガーとします。シーケンサの条件設定はシーケンサ設定ページで行います。
シーケンサ2	同上
シーケンサ3	同上

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／レンジのとき設定できます。「レンジス」に設定できる値は 2～262141 です。(両エッジクロックトレースの場合)

カスタム                    カスタム設定を行うボタンです。

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 トリガーで「トリガー ソース」が命令実行／データ アクセス／READ アクセス／WRITE アクセスの場合指定できます。

トリガー アドレス        トリガー点のアドレスを指定します。

サイズ マスク	トリガー アドレスの下位ビット マスクを指定します。次から選択します。 no mask A0 A0 & A1
データ	アクセスデータを指定します。データ指定を行うにはチェックを ON にしてください。「トリガースource」が命令実行のときは設定できません。
データ マスク	アクセス データのビット マスクを指定します。データ マスク指定を行うにはチェックを ON にしてください。またデータ指定が有効でないと設定できません。「トリガースource」が命令実行のときは設定できません。
設定(S)	トリガー条件を設定します。
解除(R)	トリガー条件を解除します。
アドレスレンジ	命令アドレス範囲情報を設定します。「トレースモード」がレンジの場合指定できません。
「開始アドレス」	
「終了アドレス」	命令アドレス範囲の開始アドレスと終了アドレスを指定します。トレースが実行される範囲は開始アドレス $\leq$ 命令アドレス $<$ 終了アドレスです。命令アドレス範囲は複数範囲設定することができます。
追加(A)	命令アドレス範囲を追加します。
削除(D)	命令アドレス範囲を削除します。
全削除(F)	命令アドレス範囲を全削除します。
完全モード	完全トレースモードでトレースを実行します。完全トレースモードではトレースデータ FIFO バッファがオーバーフローする前に CPU 実行をストールさせます。トレースデータの取りこぼしがない反面、ユーザ プログラムのリアルタイム性が損なわれません。
FIFO レベル	完全トレースモードにおいて CPU 実行をストールさせる FIFO の残量を指定します。FIFO 深さは CPU のインプリメントに依存しますので、CPU 設計者にお問い合わせください。

- 全分岐トレース 全分岐トレースを可能にします。通常では直接分岐命令に対して分岐情報は出力されません。全分岐トレースを ON にするとすべての分岐命令に対して分岐情報が出力されます。全分岐トレースを ON にすることで、ユーザ プログラム実行中にアドレス表示モードトレース結果表示が可能になります。
- CPRTトレース MCR/MRC 命令による CPU-コプロセッサ間データ転送をトレース対象にします。
- サイクル精度 サイクル精度トレースを有効にします。通常では有効なトレース情報がない場合、MJ はトレースバッファメモリへのトレース情報の取り込みを行いません。サイクル精度トレースを有効にすると、すべてのサイクルでトレース情報の取り込みます。特定の命令間の実行時間を計測することができます。

操作2: データ ページでは、データ トレースの条件設定を行います。データ トレースは命令トレースが有効なときのみ出力されます。命令トレースがレンジ モードで無効な区間では発生しません。



アクセス データ アクセス タイプを指定します。次から選択します。

データアクセス

READ アクセス

WRITE アクセス

サイズ マスク データアクセスアドレスの下位ビットマスクを指定します。次から選択します。

No mask

A0

A0 & A1

領域 領域条件を指定します。次から選択します。

Include

exclude

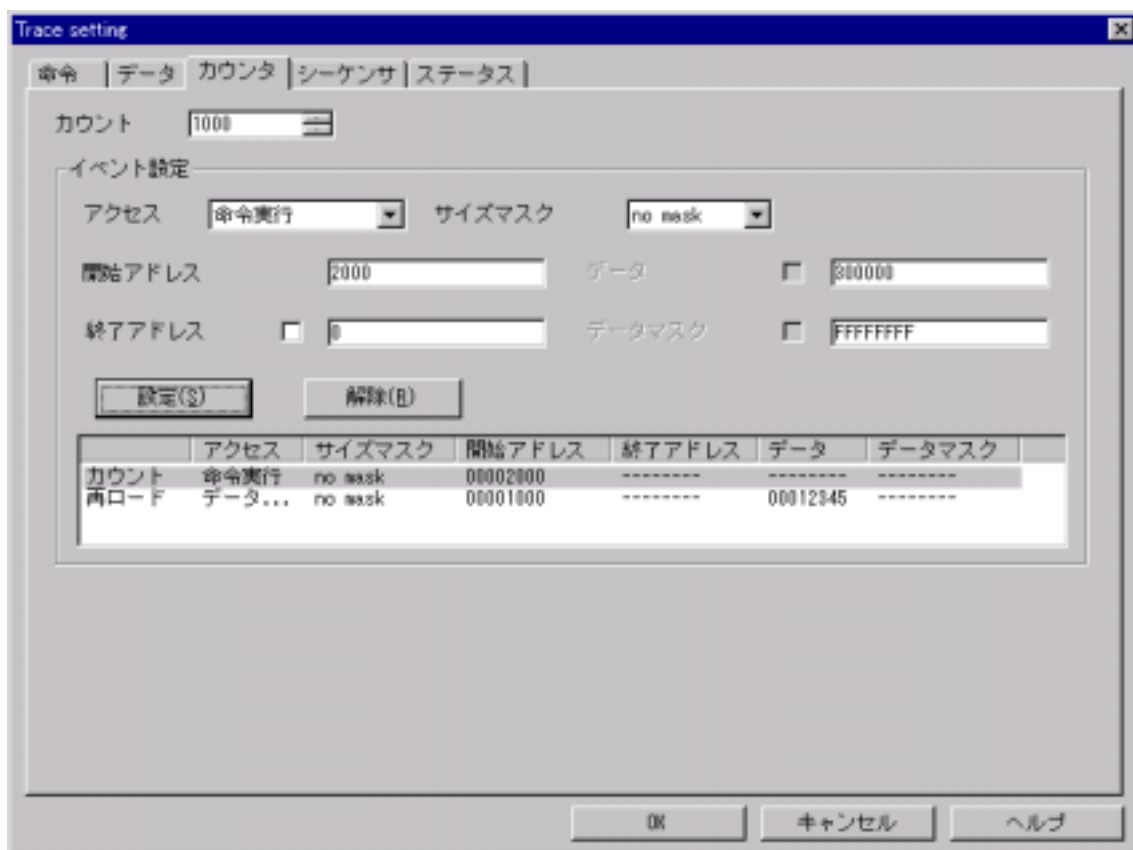
Exclude を指定すると指定アドレス(範囲)がデータトレースから除外されます。「開始アドレス」「終了アドレス」データアクセスアドレス範囲の開始アドレスと終了アドレスを指定します。単一アドレスを指定する場合は、終了アドレスのチェックを OFF にします。トレースが実行される範囲は開始アドレス  $\leq$  データアクセスアドレス  $<$  終了アドレスです。

データ	アクセスデータを指定します。データ指定を行うにはチェックを ON にしてください。
データマスク	アクセスデータのビットマスクを指定します。データマスク指定を行うにはチェックを ON にしてください。またデータ指定が有効でないと設定できません。
追加(A)	データアクセスアドレス(範囲)を追加します。
削除(D)	データアクセスアドレス(範囲)を削除します。
全削除(F)	データアクセスアドレス(範囲)を全削除します。
トレース出力	<p>トレース出力するデータ情報種別を指定します。次から選択します。</p> <p>アドレスのみ</p> <p>データのみ</p> <p>データおよびアドレス</p>

「トレース出力」設定はすべてのデータトレース条件に対して共通です。注) データトレース条件はどのデータアクセスをトレース対象にするか決定します。一方「トレース出力」設定はトレース対象となった結果そのアクセスのどの情報をトレースデータとして出力するかを決定します。



操作 3: カウンタ ページでは、カウンタ条件設定を行います。命令トレース設定の「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 トリガーで「トリガー ソース」がカウンタ イベントの場合のみ指定できます。

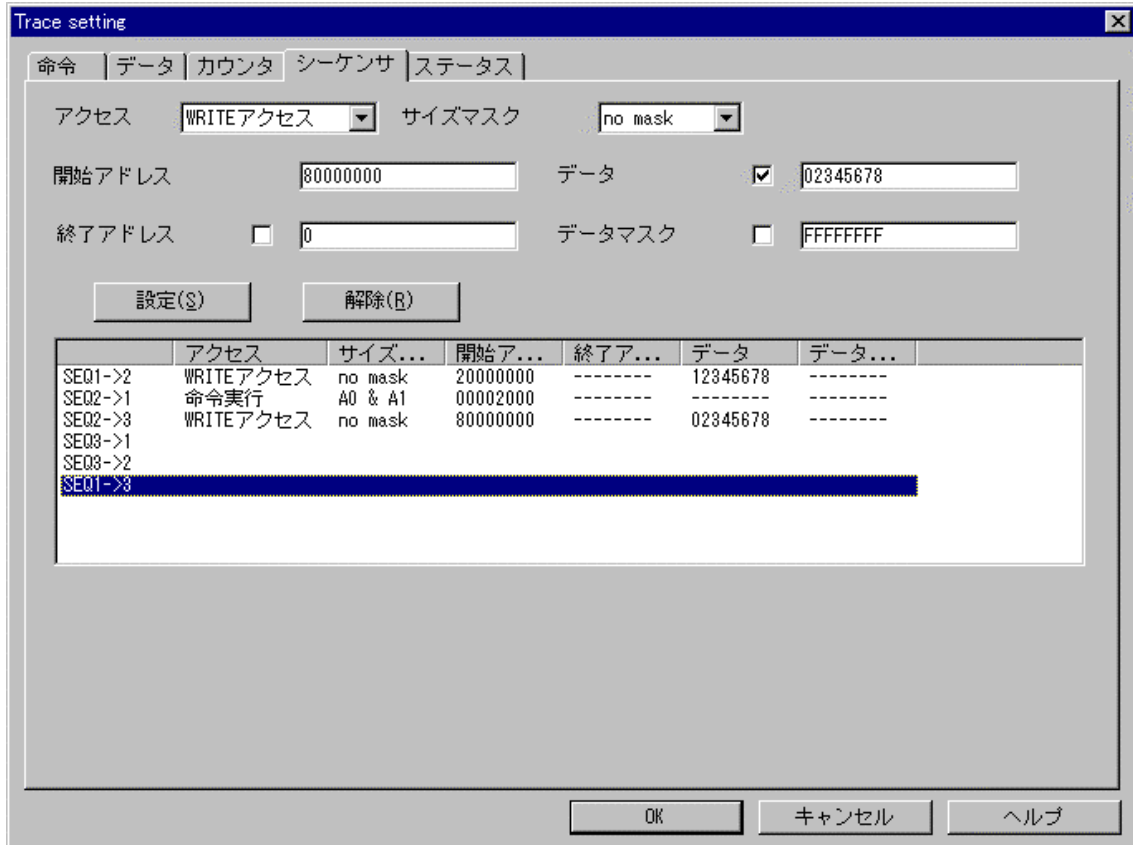


カウンタ カウンタのカウンタ値を指定します。設定できる範囲は 1～65535 です。

イベント設定 カウンタイベントをカウンタイベントと再ロードイベントについてそれぞれ個別に設定します。カウンタイベントが発生するたびにカウンタのカウンタアップが行われ、カウンタ値に達した段階でトリガーが発生します。また再ロードイベントが発生した場合、カウンタのカウンタ値のリセットが行われます。

カウンタイベントの条件指定についてはデータトレース設定ページをご覧ください。

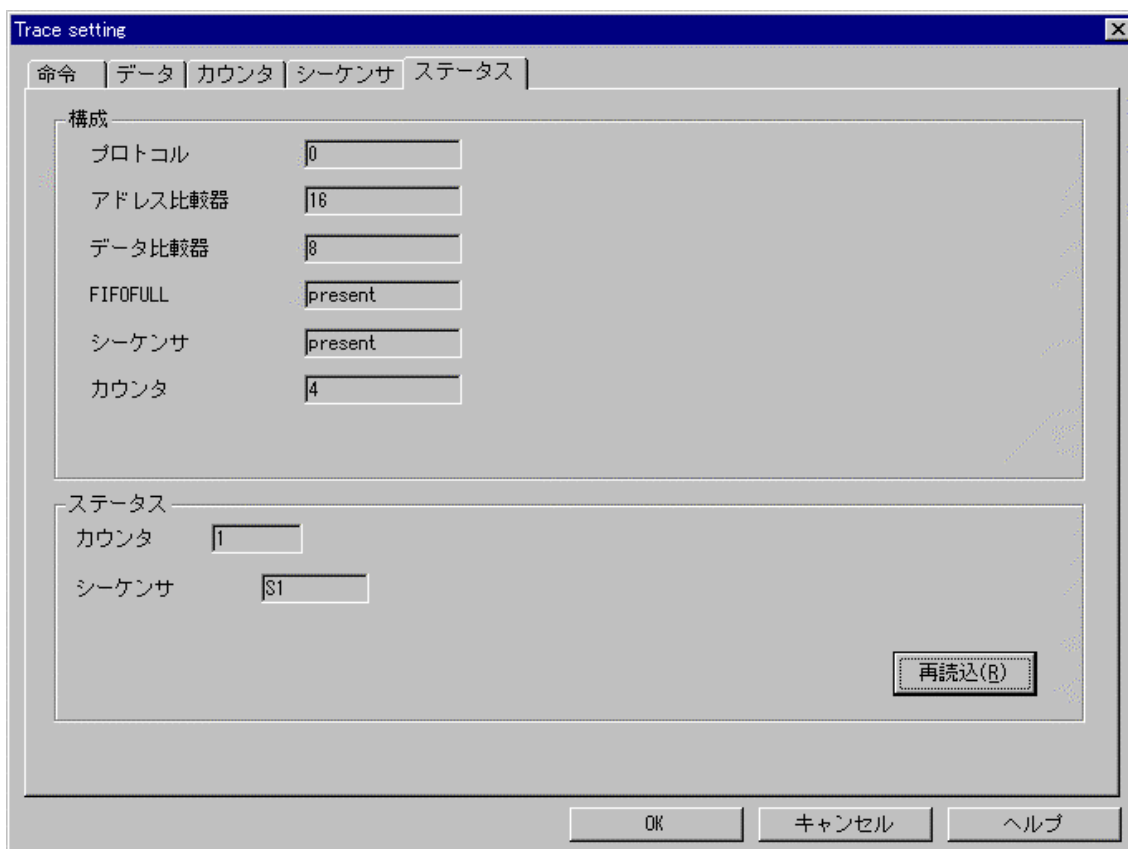
操作 4: シーケンサ ページでは、シーケンサ状態遷移条件設定を行います。命令トレース設定の「トレースモード」がビギントリガー/ミッドトリガー/エンドトリガーで「トリガーソース」がシーケンサ1/シーケンサ2/シーケンサ3の場合のみ指定できます。



ETM には3状態のシーケンサが組み込まれています。各状態間の状態遷移イベントをそれぞれ個別に設定します。シーケンサの状態が命令トレースの「トリガーソース」で指定した状態に遷移した段階でトリガーが発生します。ユーザ プログラム開始時は状態1から開始します。

状態遷移イベントの条件指定についてはデータトレース設定ページをご覧ください。

操作 5: ステータス ページでは、ETM のリソースの構成とステータスを表示します。



#### 構成

プロトコル	MJXDEBW はプロトコルバージョン 0 および 1 に対応しています。
アドレス比較器	アドレス比較器の個数を表示します。
データ比較器	データ比較器の個数を表示します。
FIFOFULL	FIFOFULL 機能が実装されているかどうかを表示します。
シーケンサ	シーケンサが実装されているかどうかを表示します。
カウンタ	カウンタの個数を表示します。標準設定では、カウンタは一つのみ使用できます。

ステータス

カウンタ	カウンタの現在のカウント値を表示します。
シーケンサ	シーケンサの現在のステートを表示します。
オーバーフロー継続中	<b>FIFO</b> オーバーフローが発生中であることを示します。
再読込	ステータスを読み込みます。

補足:

- 「アドレス比較器」「データ比較器」は有限なリソースです。

アドレス比較器はシングルポイント比較器であり二つのアドレス比較器を組み合わせることにより、アドレス範囲比較をすることができます。

アドレス比較器とデータ比較器は固定的に結びついています。すなわちアドレス比較器にはデータ比較器を持つものと持たないものがあります。

- アドレス比較器の動的割り当て

アドレス比較器のユーザーはトリガーアドレス/アドレス範囲/データトレース/カウンタイベント/シーケンサイベントです。

MJXDEBW は要求があるたびに比較器の動的割り当てを行います。比較器リソースが尽きた段階でそれ以上の設定ができなくなります。

動的割り当てアルゴリズムはデータ比較よりもレンジ比較を優先します。この結果、比較器リソース要求の順番によってはデータ比較器を有効に割り当てられない場合があります。

比較器リソースを有効に使用するためには比較器リソース要求を以下の優先順位で行ってください。

- 1 データ比較付きアドレス範囲比較(高)
- 2 データ比較なしアドレス範囲比較
- 3 データ比較付きアドレス比較
- 4 データ比較なしアドレス比較(低)

- 「第七章 [TRACE](#)」参照。

操作 6: トレースのカスタム設定を行います。カスタム設定では、ETM のレジスタレベルでトレース設定を行なうため、ETM に関する完全な知識が必要になります。

(カスタム設定のコントロールのページ)



## 設定(S) ~ 設定の読出(L)...

---

機能:

ブレイクポイントおよびトレースの設定を、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 [SETLOAD](#)」参照。

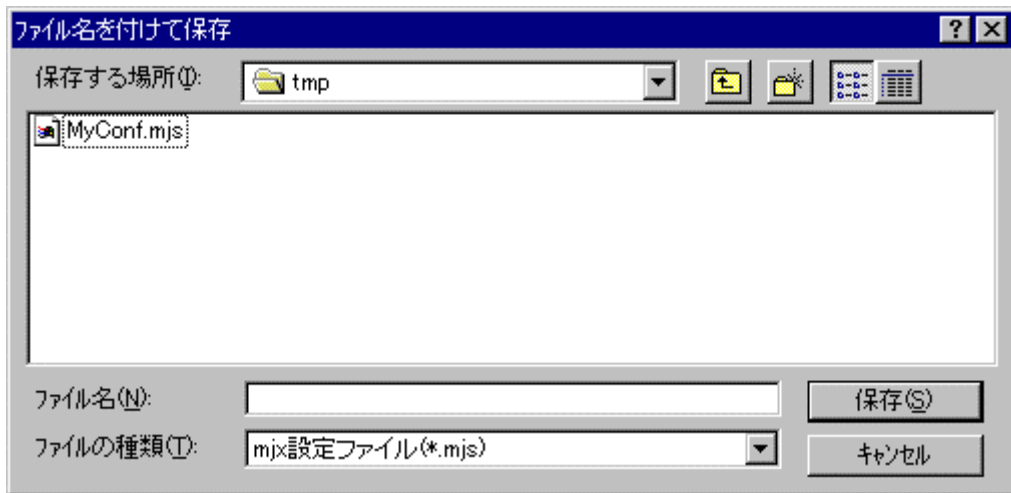
## 設定(S) ~ 設定の保存(S)...

---

機能:

ブレイクポイントおよびトレースの設定を、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、「保存」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。
- 「第七章 [SETSAVE](#)」参照。



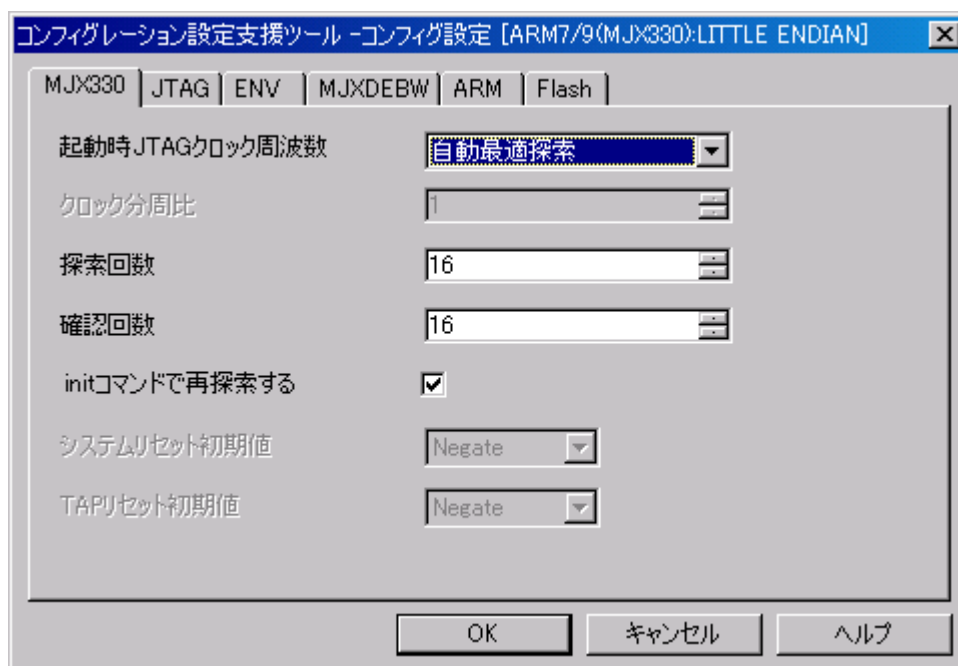
## 設定(S) ~ コンフィグレーション(C) ~ 設定(S)...

CONFIG

機能:

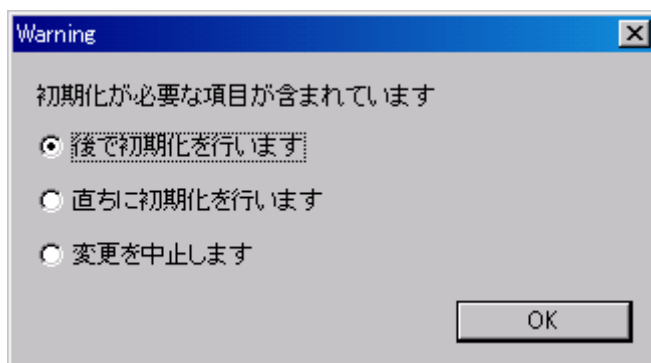
MJXDEBW の環境を変更します。

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。

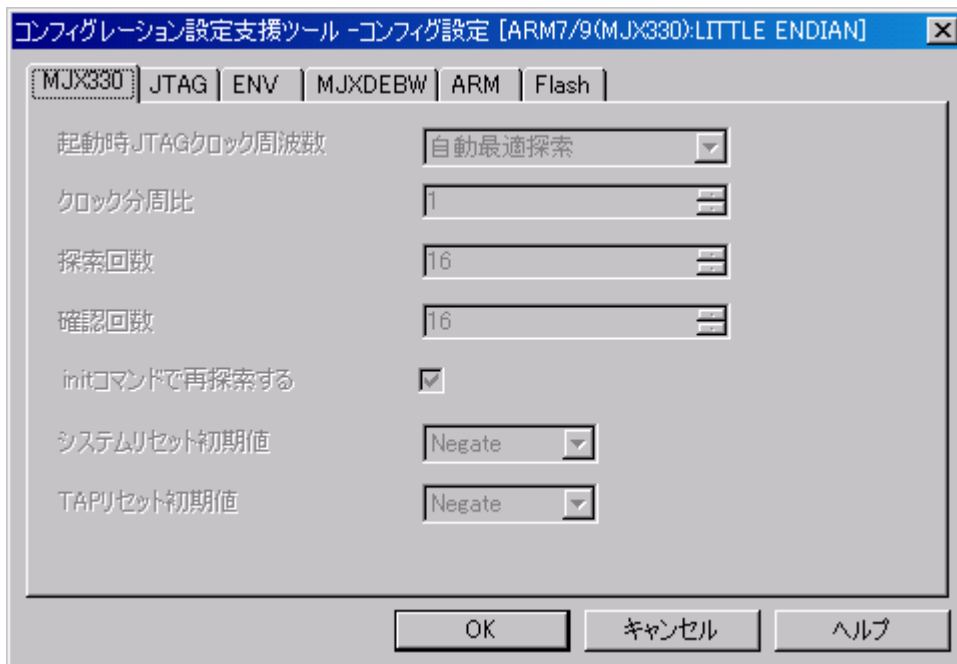
## 設定(S) ~ コンフィグレーション ~ 表示(V)

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:

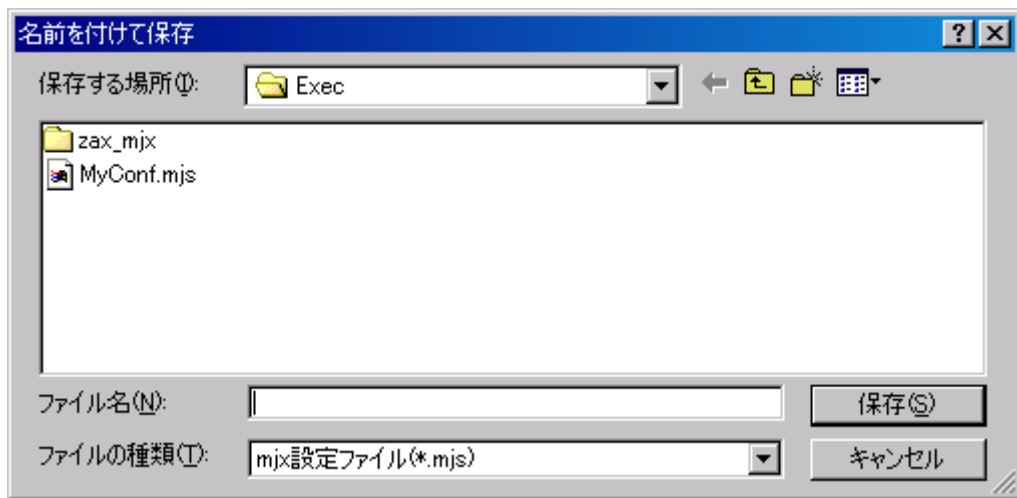
## 設定(S) ~ コンフィグレーション ~ 名前を付けて保存(A)

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。

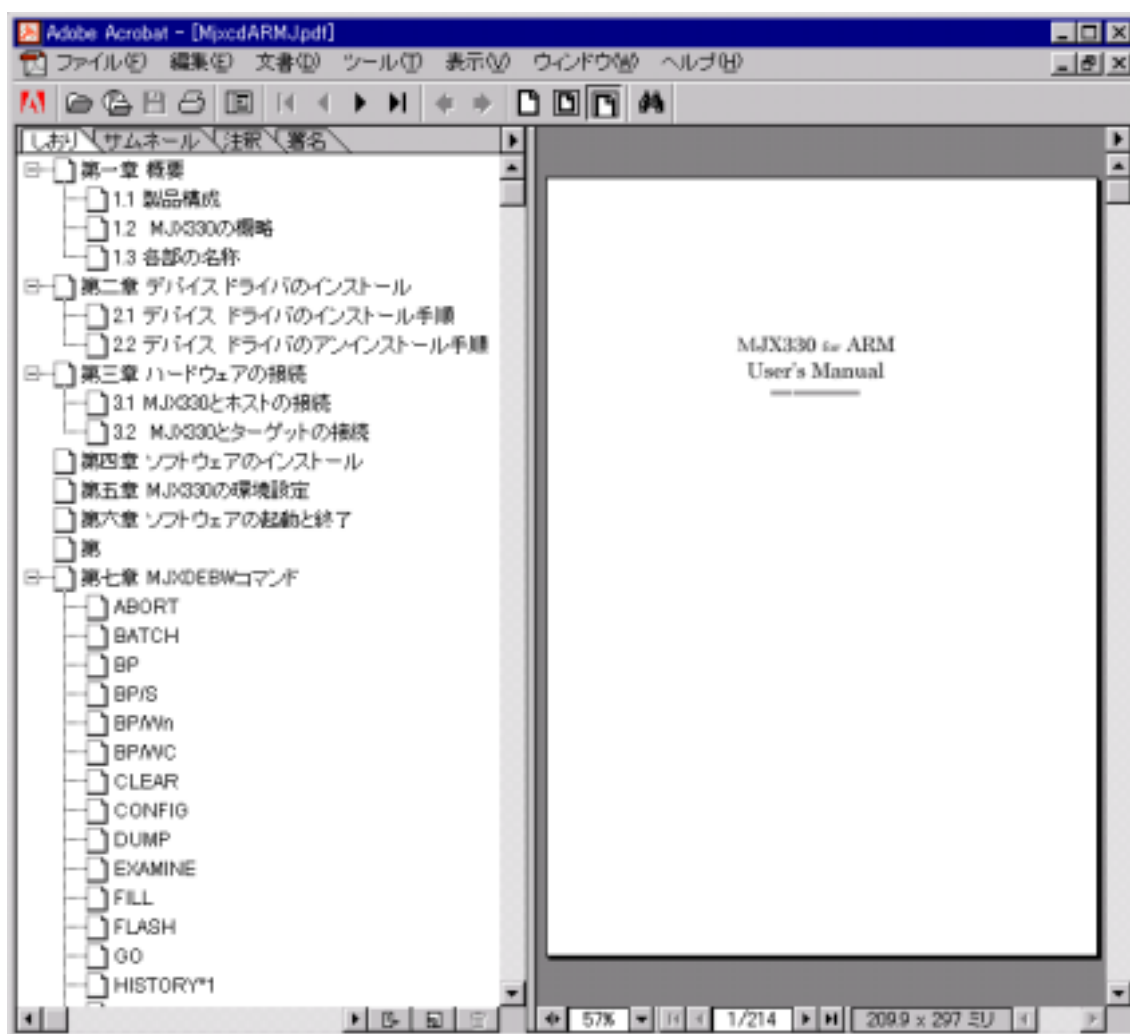
## ヘルプ(H) ~ トピックの検索(H)

機能:

MJX330 for ARM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。

- 400K バイト/秒\*<sup>1</sup> (JTAG クロック20MHz)

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」\*<sup>2</sup> 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
```

---

\*<sup>1</sup> 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。

\*<sup>2</sup> Windows 2000/NT/XP に関しては「コマンドプロンプト」内から実行してください。



## 付録 A 仕様

本体寸法	PC カード TYPE II (85.6mm(縦)×54.0mm(横)×5.0mm(厚さ))
本体重量	33g
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%
インターフェース	PCMCIA
ターゲット インターフェース	<a href="#">JTAG コネクタ</a>
対応 CPU	ARM7、ARM9、ARM9E
ダウンロード速度	400K バイト/秒*1 (JTAG クロック20MHz)
対応デバッガ	Green Hills <a href="#">MULTI</a> ARM      AXD Gnu      GDB
ブレークポイント機能	ハードウェア      × 2 ソフトウェア      × 128
制限事項	「 <a href="#">付録 B ターゲット システムの制限事項</a> 」参照

---

\*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 B ターゲット システムの制限事項

MJX330 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。

- JTAG コネクタを実装している。

## 付録 C JTAG コネクタ

### ターゲットシステムの JTAG コネクタ(部品面)

ピンアサイン

VTRef* <sup>1</sup>	01	02	VSupply
nTRST	03	04	GND
TDI	05	06	GND
TMS	07	08	GND
TCK	09	10	GND
RTCK	11	12	GND
TDO	13	14	GND
nSRST	15	16	GND
DBGRQ	17	18	GND
DBGACK	19	20	GND

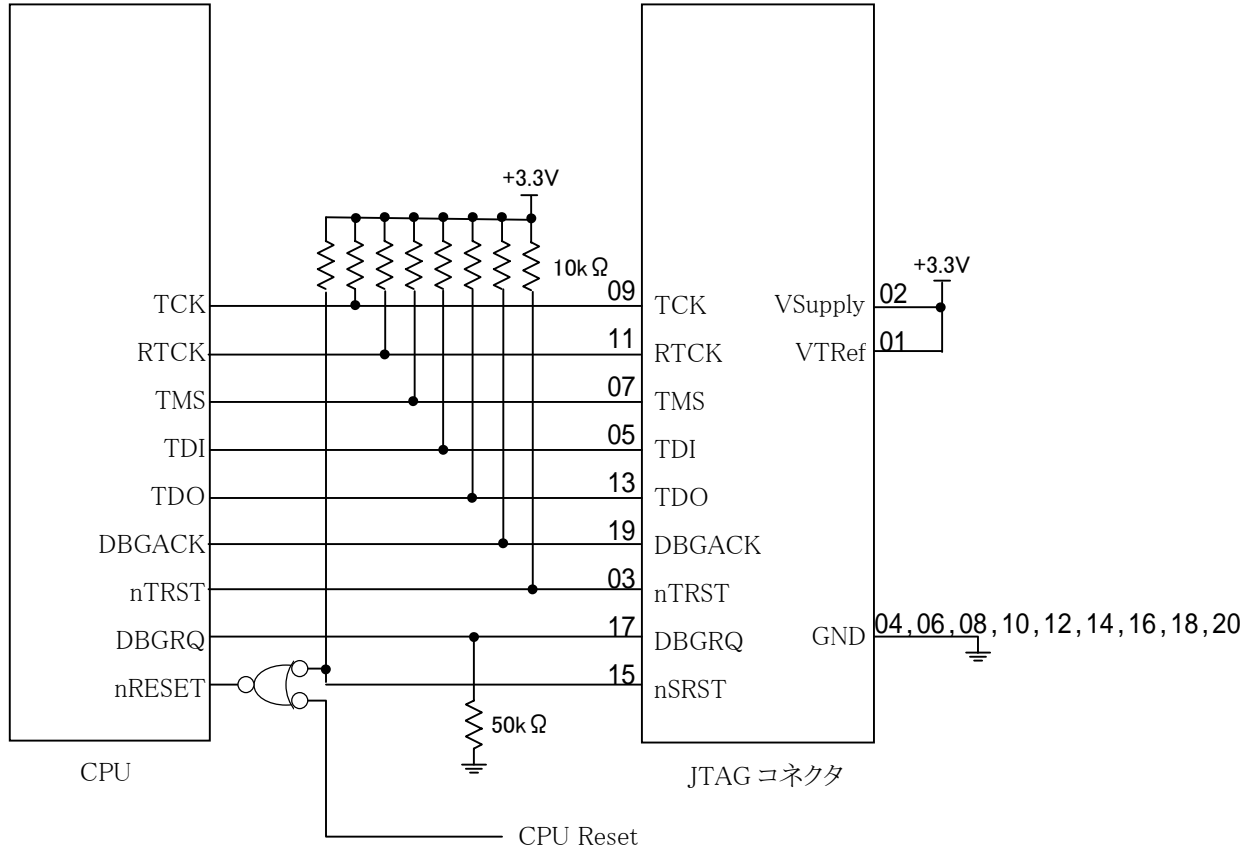
推奨コネクタ

- ヒロセ電機製 HIF3F-20PA-2.54DSA (ストレート)
- ヒロセ電機製 HIF3F-20PA-2.54DS (ライトアングル)

---

\*1 MJX330 側は N.C.

ターゲット システムの JTAG コネクタ推奨回路



注意事項

- CPU～JTAG 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 信号のパターンは、GND でシールドしてください。
- MJX330 は、VSupply をターゲットプロービングバッファの電源に使用しています。

## 付録D レジスタ名一覧

(汎用レジスタ)

R0					
R1					
R2					
R3					
R4					
R5					
R6					
R7					
R8	R8_FIQ				
R9	R9_FIQ				
R10	R10_FIQ				
R11	R11_FIQ				
R12	R12_FIQ				
R13	R13_FIQ	R13_SVC	R13_ABT	R13_IRQ	R13_UND
R14	R14_FIQ	R14_SVC	R14_ABT	R14_IRQ	R14_UND
PC					
CPSR	SPSR_FIQ	SPSR_SVC	SPSR_ABT	SPSR_IRQ	SPSR_UND

## 付録 E 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)

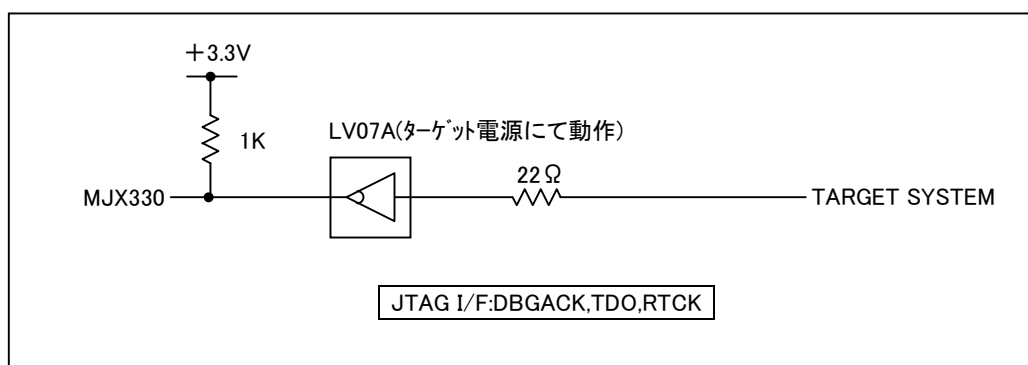
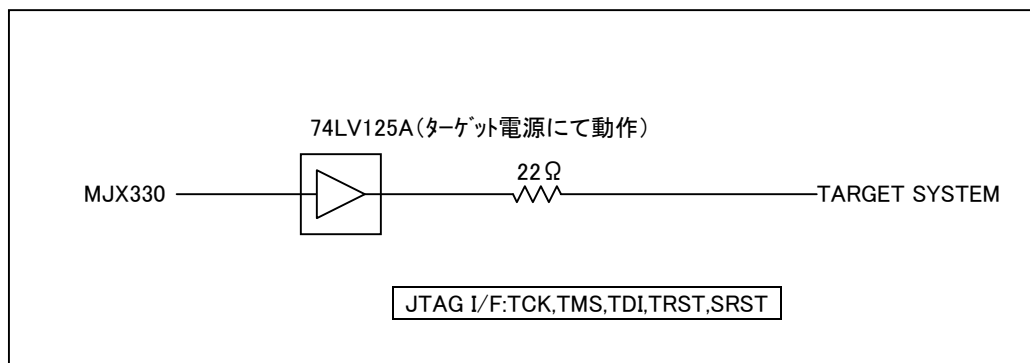
データ (可変、データ バイト長)

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

終了ヘッダ (16 バイト長)

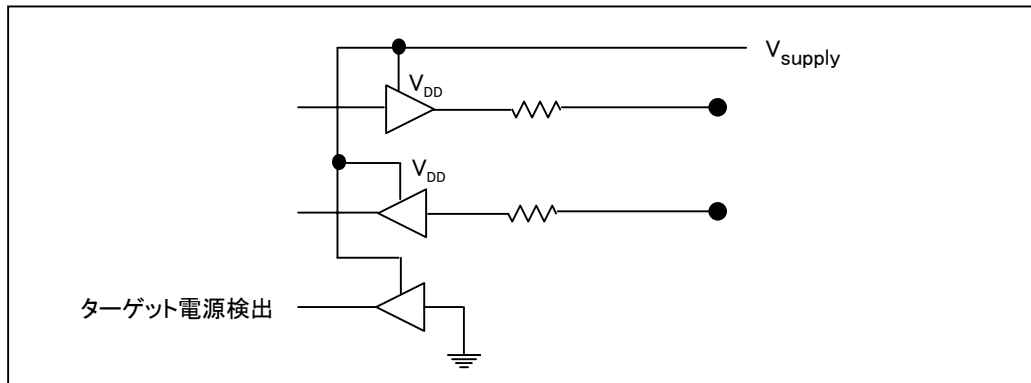
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
'M'	'J'	'1'	00	00	00	00	00	00	00	00	00	00	00	00	00

## 付録 F ターゲット システムへのプロービング



DBGREQ 信号はオープンになっています。

付録F ターゲット システムへのプロービング





## 付録 G 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(第7行以降)、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 <sup>(注1)</sup>	チップ消去機能無フラグ <sup>(注2)</sup>
2	デバイス コード <sup>(注3)</sup>	
3	アクセス モード <sup>(注4)</sup>	コマンド・インターフェース <sup>(注5)</sup>
4	コマンド入力 第1バス サイクル 書き込み アドレス <sup>(注6)</sup>	
5	コマンド入力 第2バス サイクル 書き込み アドレス <sup>(注6)</sup>	
6	セクタ数 <sup>(注7)</sup>	
7	第 1 ~ n セクタ サイズ <sup>(注8)</sup>	セクタ数 <sup>(注8)</sup>
	...	
	...	
-	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	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	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出力することで、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

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 H 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

続き

型式	構成	電源電圧
AM29F002BB	256Kx8	5V
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



続き

型式	構成	電源電圧
AT49BV4096A	512Kx8/256Kx16	3V
AT49LV4096A	512Kx8/256Kx16	3V
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

続き

型式	構成	電源電圧
AT49LV161	2Mx8/1Mx16	3V
AT49LV161T	2Mx8/1Mx16	3V
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

続き

型式	構成	電源電圧
MBM29DL800BA	1Mx8/512Kx16	3V
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

## 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 H 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 H 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 I MjxRDI サーバーの詳細

### I-1 Semihosting 機能

Semihosting 機能は SWI 命令を使用して実現します。

#### Semihosting 実装方法

- VectorCatch レジスタが利用できる CPU  
VectorCatch レジスタにより SWI 命令を捕捉します。  
VectorCatch レジスタ設定値に関わらず SWI 命令は MjxRDI サーバーにより捕捉されます。
- VectorCatch レジスタが利用できない CPU  
ブレイクポイントを使用して SWI 命令を捕捉します。  
SWI ハンドラが ReadOnly メモリ領域にある場合、ハードウェア ブレイクポイント リソースを1点使用します。

SWI 命令を捕捉されたあと MjxRDI サーバーは SWI 番号を確認して、Semihosting 用の番号でなかった場合、そのままアプリケーション プログラムを再開します。そのため、ユーザー用の SWI 命令と共存することができます。ただし、SWI 命令の発行でいったん必ず MjxRDI サーバーに制御が移るため、アプリケーション プログラムはリアルタイムでは実行できません。

Semihosting 機能を有効にするためには、デバッグ内部変数 \$semihosting\_enabled を 1 にします。  
\$semihosting\_enabled=2 は DCC-Semihosting 機能ですが、MjxRDI サーバーでは利用できません。

#### ユーザー定義の SWI ハンドラとの共存

ユーザー用の SWI 命令をリアルタイムで実行したい場合はデバッグ内部変数 \$semihosting\_vector により semihostingSWI アドレスを移動します。ユーザー アプリケーション プログラムの変更が必要です。アプリケーション プログラムは SWI 命令の解釈を行い、アプリケーションで定義していないと判断した場合にのみ通過する命令のアドレスを semihosting\_vector へ設定します。アプリケーション プログラムが semihosting\_vector へ到着した段階では SWI ハンドラ開始時のすべてのレジスタが保存されている必要があります。

SWI 命令が Semihosting 用でないと MjxRDI サーバーが判断した場合は、アプリケーション プログラムが semihosting\_vector の次の命令から再開されます。

そのためアプリケーション プログラムではこの後に未定義 SWI の処理するプログラムを記述しなければなりません。

semihosting\_vector により SWI ベクターを移動した場合、すべての CPU で SWI 捕捉にブレイクポイントが利用されることに注意してください。

## I-2 ハードウェア ブレークポイント リソース 割り当て規則

### ブレークポイント

ブレークポイントはブレークポイント アドレスのメモリがリード/ライト可能メモリであった場合はソフトウェア ブレークポイントを使用します。リードオンリー メモリであった場合はハードウェア ブレークポイントを使用します。

### ウォッチポイント

ブレークポイントはすべてハードウェア ブレークポイントを使用します。

### MjxRDI サーバーが暗黙のうちに使用するハードウェア ブレークポイント

MjxRDI サーバーは、以下の条件のときハードウェア ブレークポイントを使用します。

- VectorCatch レジスタが使用できない CPU 機種において Semihosting 機能を使用する場合で、かつ semihosting\_vector 領域がリードオンリーメモリである場合。
- VectorCatch レジスタが使用できる CPU 機種であってもデバッガ内部変数 \$semihosting\_vector により semihosting\_vector アドレスが設定されている場合で、かつ semihosting\_vector 領域がリードオンリーメモリである場合。
- ユーザーが明示的にあるいは MjxRDI サーバーが暗黙のうちにソフトウェア ブレークポイントを使用する場合

**【注意】** ソフトウェア ブレークポイントは未定義命令を使用して実現されます。

Vector Catch レジスタが使用できる CPU 機種であっても未定義命令の捕捉にはハードウェア ブレークポイント リソースが使用されます。

### ハードウェア ブレークポイント リソース割り当て規則

ハードウェア ブレークポイント リソースはユーザーが明示的に指定する場合あるいは MjxRDI サーバーが暗黙のうちに指定する場合に関わらず先に要求があった方にリソースが割り当てられます。

ハードウェア ブレークポイント リソースが満杯であった場合、要求は却下されます。

### ハードウェア ブレークポイント リソース

ハードウェア ブレークポイントは EmbeddedICE マクロセルに含まれる WatchPoint 機能を使用して実現します。EmbeddedICE マクロセルには WatchPoint が2チャンネル含まれています。

## 付録 J デバイス ドライバのトラブルシューティング

### J-1 PC カードを差すとホスト PC がハングアップする場合 (Windows 95/98/98SE)

一部のノート PC\*<sup>1</sup> では、ドライバのインストールのために PC カードを差すと、ハングアップしてしまうことがあります。

そのような場合は一旦強制的に電源をオフし、電源オフの状態ですべての PC カードを差してから、電源をオンしてください。

その後、「2.1 デバイス ドライバのインストール手順 [2.1.1 Window98/98SE の場合](#)」にしたがってドライバをインストールしてください。

---

\*<sup>1</sup> SONY VAIO PCG-505 でこの現象を確認しています

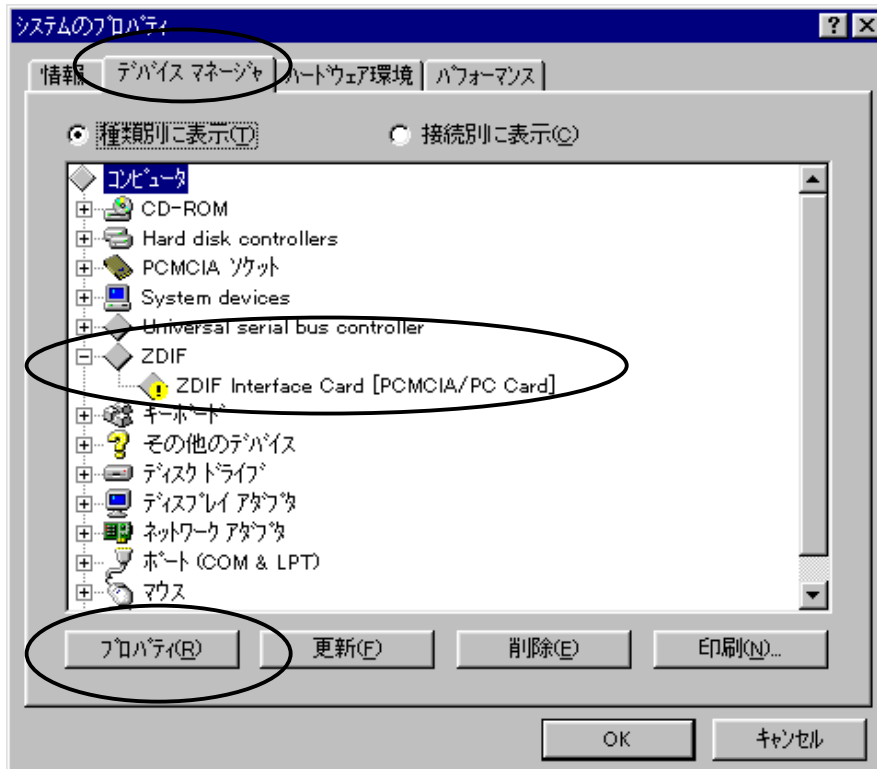
## J-2 リソースの競合、またはリソースの空きがない場合 (Windows 98/98SE)

PC カードのデバイス ドライバのインストール終了時に「ピポッ」ではなく「ブー」という音がした場合は、ドライバのインストールに失敗しています。

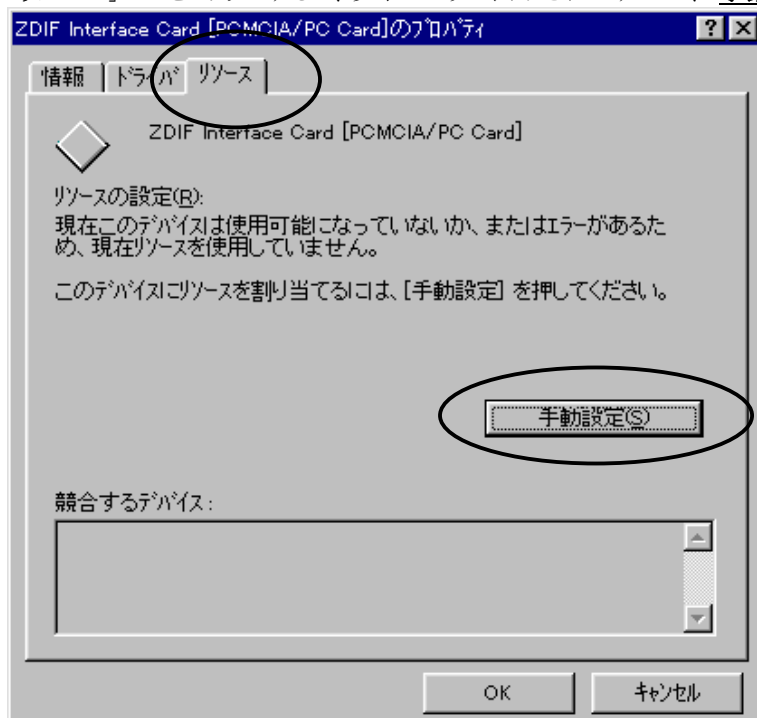
まずコントロール パネルから「システム」をダブル クリックして開き、「デバイス マネージャ」タブをクリックします。

次に「ZDIF」クラスをクリックして、その下にある「ZDIF Interface Card [PCMCIA/PC Card]」のプロパティを表示します(ダブル クリックするか、「プロパティ(R)」をクリック)。

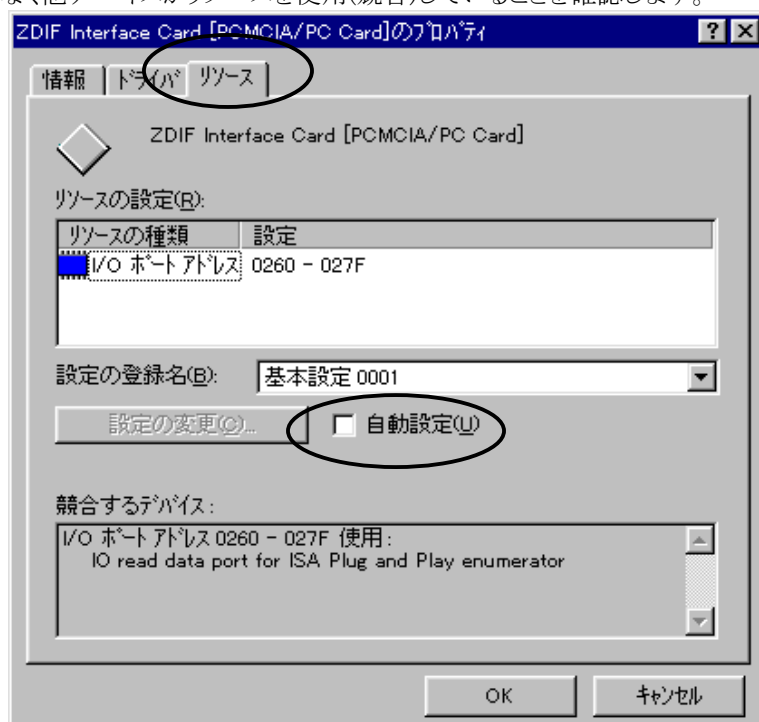
先頭の「!」マークは、デバイス ドライバのインストールに問題があったことを示しています。



「リソース」タブをクリックすると、以下のように表示されますので、「**手動設定(S)**」をクリックします。



「自動設定(U)」がオフ(チェック オフ)になっていること、「競合するデバイス」が「競合はありません。」ではなく他デバイスがリソースを使用(競合)していることを確認します。

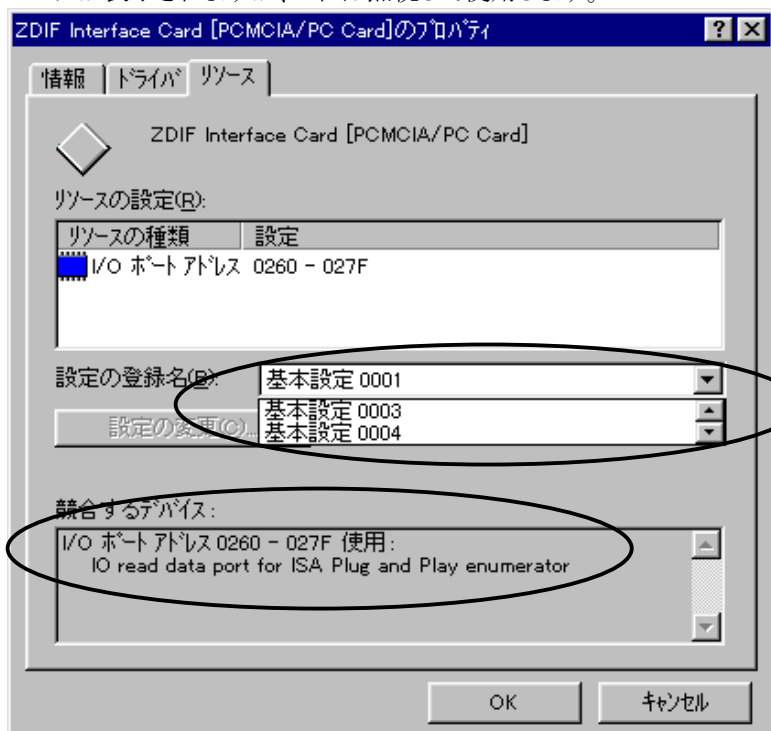




「設定の登録名(B)」で「基本設定 000」から「基本設定 004」までを順に選択してみます。

「競合するデバイス」に「競合はありません。」と表示されるものがあればこれを使用します。

その場合、Plug & Play 機構による自動設定ではなく、手動による設定を行ったこととなりますので、ワーニングが表示されますが、これは無視して使用します。



「基本設定 000」から「基本設定 004」までの全ての設定で競合が発生している場合は、競合している他デバイスのリソースを変更して空にしない限り、PC カードを使用することはできません。

PC カードが使用するリソース(I/O ポート)は以下のうちのどれか一つです。

0220 - 023F = 基本設定 000

0260 - 027F = 基本設定 001

02E0 - 02FF = 基本設定 002

0320 - 033F = 基本設定 003

03E0 - 03FF = 基本設定 004

競合している他デバイスのリソース(I/O ポート)も変更できない場合は、PC カードを使用することはできません。

### J-3 割り込みを使用しない PC カードを受け付けないホスト PC の場合 (Windows 98/98SE)

PC カードのデバイス ドライバのインストール終了時に「ブー」という音がしてインストールに失敗したものの、「リソースの競合、またはリソースの空きがない場合」のパターンには当てはまらない (リソースの競合がない、またはドライバがデバイス マネージャに登録されていない等)場合は、以下の手順を試してみてください。

一部のノート PC<sup>\*1</sup>では、割り込みを使用しない PC カードを正しく認識できないことがあります。この場合は一旦、ドライバのアンインストールを行った後、”擬似的に割り込みを使用するように設定したインストール”を行います。

まず、「[2.2 デバイス ドライバのアンインストール手順](#)」を参考にして、デバイス ドライバを削除します。INF ファイルも削除してください。

一旦ノート PC をシャットダウンして電源オフし、PC カードを抜きます。

次に、「[2.1 デバイス ドライバのインストール手順](#)」にしたがってドライバを再インストールしますがその際、ドライバ ファイルのあるディレクトリには

D:¥Driver¥Win9x¥ThinkPad (CD-ROMドライブが D の場合)

を指定してください。

このディレクトリに含まれるドライバ ファイルは「D:¥Driver¥Win9x」と同じですが、PC カードが割り込みを使用するように擬似的に設定する INF ファイルが入っています。

なお、このインストールが正しく行われるためには、割り込みが最低 1 つは空いている必要があります。

---

\*1 IBM ThinkPad シリーズの一部旧機種でこの現象を確認しています。

## J-4 ハードウェア ウィザードが起動せず、デバイス ドライバをインストールできない場合(Windows 98/98SE)

ノート PC に PC カードを差しても全く認識されず、ハードウェア ウィザードも起動しない場合があります\*1。

この場合は PC カードを使用することはできません。

詳細は弊社サポートまでお問い合わせください。

## J-5 リソースの空きがない場合(WindowsNT4.0)

PC カードは以下のリソース(I/O ポート)のうち、いずれか一つ空いているものを使用します。

0220 – 023F

0260 – 027F

02E0 – 02FF

0320 – 033F

03E0 – 03FF

これらのリソースが全て他デバイスで使用されていて空きがない場合は、競合している他デバイスのリソースを変更して空きにしない限り、PC カードを使用することはできません。

---

\*1 IBM ThinkPad 380ED 2635 7AJ でこの問題が報告されています。

## 付録 K ARM ステートと Thumb ステートの指定方法

ARM ステートと Thumb ステートを変更する場合は、MJXDEBW のコマンド入力フィールドに次のコマンドを入力してください。

*reg△ thumb=0* (ARM ステートにする)

*reg△ thumb=1* (Thumb ステートにする)

## 検索項目

### [MJX330 について](#)

[MJX330 のおもな特長](#)

[構成](#)

[MJX330 を使用する前の準備作業](#)

### [MULTI について](#)

### [MJXDEBW について](#)

[デバイス ドライバのインストール](#)

[ハードウェアの接続](#)

[MJX330 とホストの接続](#)

[MJX330 とターゲットの接続](#)

### [MJX330 を操作するソフトウェアのインストール](#)

[MJX330 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](#)

### [MJX330 の環境設定](#)

[コンフィグ支援ツール MJXCFG](#)

### [MJX330 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](#)

### [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](#)

[次ページ](#)

[仕様](#)

[本体寸法](#)

[本体重量](#)

[使用温度範囲](#)

[保存温度範囲](#)

[周囲湿度範囲](#)

[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](#)

[制限事項](#)

[ターゲット システムの制限事項](#)

[JTAG コネクタ](#)

[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](#)

[前ページ](#)

---

**MJX330 について**

< 関連項目 >

- [MJX330 のおもな特長](#)
- [構成](#)
- [MJX330 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 >

- [MJX330 とホストの接続](#)
- [MJX330 とターゲットの接続](#)

[検索項目の先頭ページ](#)

---



---

**MJX330 を操作するソフトウェアのインストール**

< 関連項目 >

- [MJX330 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](#)

---

---

**MJX330 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](#)

---

---

**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](#)

---