ＲＯＭインサーキットエミュレータ

MDX700 の紹介

Rev.1.00　1998/04/01

株式会社ライトウェル ザックス部
営業グループ

ＲＯＭインサーキットエミュレータ

ＭＤＸ７００の紹介

本書は、ＭＤＸ７００がもつ機能、使用できるターゲット システム、
基本操作といった３部構成になっております。

ご購入をご検討されている方、またご使用の初期段階に
ＭＤＸ７００とはどのような製品なのかを
ご理解いただくために、少しでもお役に立てればと思います。

MDX700 ご使用上のお願い

- 本体に接続するケーブル類は、添付される標準のものをご使用下さい。
  改造等による標準以外のものをご使用になられた場合の故障等に関しましては保証いたしかねます。
- 本体に強い衝撃を与えたり、投げつけたりしないで下さい。
  破壊や機器の故障の原因となります。
- 湿気やほこりの多いところ、高温となるところでのご使用、保管は避けて下さい。
- ノイズの多いところでのご使用は避けて下さい。
- 分解、改造はしないで下さい。
  改造等による事故や故障に関しましては保証いたしかねます。

ご注意

- 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書の内容は予告なく変更することがあります。
- 本書に掲載されている製品名や会社名等の固有名詞は、各社の商標または登録商標です。

製作・著作：　株式会社ライトウェル ザックス部
製作年月：　１９９８年３月

本書の読み方

本書は、組込み開発者がＭＤＸ７００をご理解いただけるよう、広範囲にわたり説明しております。
状況に応じ、以下に掲示します事項を中心にご覧いただき、参考にしていただければと思います。
もちろん全てを通してご覧いただければ、尚一層ご理解を深めていただけます。
また、ＭＤＸ７００ではＰＣ版とＷＳ版が用意されておりますが、本書ではＰＣ版についての説明を主としました。ＷＳ版の詳細については別途お問合わせ下さい。

ご購入を検討されている方

１．　ＭＤＸ７００でどんな事ができるの？
２．　ＭＤＸ７００が使用できるターゲットは？

初めてご使用になる方

３．　ＭＤＸ７００を使用してみる！

うまく動かないという方

3-8. エラー メッセージとトラブルシューティング
※ＭＤＸ７００ご使用時には User's Manual もご併用下さい。

シンボルの説明

本書では、次のシンボルを使用します。
本文中にこれらのシンボルがある場合、その環境に限定した説明であることを表します。

Parallel　MDX700 のインターフェース(I/F)がパラレル インターフェース仕様

Ethernet　MDX700 のＩ／Ｆがイーサネット インターフェース仕様

PC/AT　ホスト マシンが PC/AT、または互換機（DOS/V 機含む）

PC-98　ホスト マシンが PC-98

SPARC　ホスト マシンが SPARCstation

MIPS　CPU が MIPS ファミリ

V800　CPU が V800 シリーズ（V830、V850、V850E を含む）

V830　CPU が V830 ファミリ

V850　CPU が V850 ファミリ

V850E　CPU が V850E ファミリ

SH　CPU が SuperH RISC engine ファミリ（SH-1、SH-2、SH-3 の全て）

SH-1　CPU が SH7030 シリーズまたは SH7020 シリーズ

SH-2　CPU が SH7600 シリーズ

SH-3　CPU が SH7700 シリーズ

PowerPC　CPU が PowerPC ファミリ

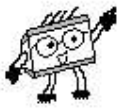
68000　CPU が 68000 ファミリ

もくじ

１．ＭＤＸ７００で
どんな事ができるの？

ＭＤＸ７００の機能や動作原理などその基本的な
仕様について説明します。

1-1．ＭＤＸ７００の機能



まずＭＤＸ７００の概要について説明します。

ＭＤＸ７００は、ＲＯＭソケットまたは同等のコネクタへインサーキットするタイプの組込みソフトウェア開発支援ツールです。よってＣＰＵの形状やその種類に依存しません。

これまでＣＰＵ毎に必要とされたエミュレータ本体が、ＭＤＸ７００では１台で様々な ＣＰＵに対応できます。（ＣＰＵに対応したソフトウェアのみ変更が必要）また､昨今のＣＰＵ周波数の高速化によるエミュレータのリアルタイム性や透過性といった 問題点も気にすることなくデバッグを進められます。

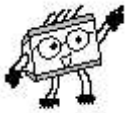
ターゲットシステム上に、ＲＯＭ化されたプログラムを載せているのと同じ環境で、ソフトウェアの開発を支援するツール、それがＭＤＸ７００です。

ＭＤＸ７００で実現できる機能（ホスト マシン上で動作するデバッガで実現）

- ROM 領域の参照、変更
- RAM 領域の参照、変更
- I/O、CPU 内部レジスタの参照、変更
- リアルタイムエミュレーション
- ステップ実行
- ブレークポイントの設定
- C 言語及びアセンブリ言語のソースを使用したデバッグ

（別途 MULTI などの高級言語デバッガが必要）

その機能を実現する為に、
特別な信号をＲＯＭに接続しないとダメなの？



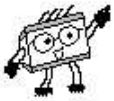
その必要はありません。

ＭＤＸ７００は、ターゲットシステムとの接続はＲＯＭソケットのみで、本体内のエミュレーションメモリでその代行をし、デバッグを行います。

ＭＤＸ７００使用時、ROM 及びＲＡＭ領域の１部を使用してモニタ プログラムを動作させます。これにより ROM だけではなく、RAM や I/O、CPU 内部レジスタ等の参照や変更を可能にします。

1-2.モニタ プログラム

ところでモニタ プログラムって何？



それでは、モニタ プログラムについて説明します。
モニタプログラムは、ＭＤＸ７００と接続したターゲット システムを動作させる際に、ターゲット ＣＰＵ上で最初に動作する管理プログラムです。

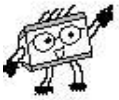
モニタ プログラムは、デバッガからの機能要求を受け取り、それを実行し結果をデバッガに返す、という処理を繰り返しています。

ＲＯＭ領域へのアクセスはデバッガが行ないますが、ＲＡＭ領域へのアクセスやユーザ プログラムの実行は、モニタ プログラムが行ないます。

モニタ プログラムとユーザ プログラムは、同じターゲット システム上で動作するため、ＣＰＵ資源（メモリや例外ベクタ等）との競合をおこさないような工夫が必要になります。

通常はユーザ プログラムが使用しない領域に、モニタ プログラムを配置するように環境設定をします。

そうすると モニタ プログラムはデバッグするプログラムとリンクしないといけないの？



その必要はありません。
モニタ プログラムはＭＤＸ７００を使用する時のみターゲット システム上へ常駐します。
後述するコンフィグレーション ファイルで常駐するアドレスを指定できます。
但し、モニタ プログラムが使用できるスペースがメモリ上に必要となります。

- ＲＯＭ領域　　８Ｋ ～ １６Ｋ Byte
- ＲＡＭ領域　　４Ｋ Byte

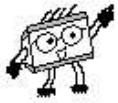
モニタ プログラムは、次の機能要求を実行します。

モニタ プログラムの機能

- ＲＡＭの読み出し
- ＲＡＭの書き込み
- ユーザ プログラムの実行
- ユーザ プログラムのステップ実行 [68000] [PowerPC]
- 次の PC の計算 [MIPS] [SH] [V800]

デバッガは、これらの機能要求を、必要に応じてモニタ プログラムへ送信します。

モニタ プログラムは添付されるの？



はい。ソース ファイル(ｱｾﾝﾌﾞリ言語で記述)も添付されます。

通常モニタ プログラムは添付の実行可能形式ファイルをそのまま使用しますが、モニタ プログラムの変更を必要とする場合があります。

モニタ プログラムの変更が必要なターゲット システム

- ● ＤＲＡＭコントローラなどを初期化しないと、ＲＡＭへのアクセスができない
- ● ＣＰＵの内部レジスタなどを初期化しないと、ＲＡＭへのアクセスができない

※ ここでの RAM はモニタ プログラムが使用する領域を指します。

このような場合は、以下のように RAM へのアクセスができるよう、初期化コードをモニタ プログラムに追加して下さい。

1. モニタ プログラムのソース ファイルをエディタで開きます。
2. ソース ファイルの後方にある、USER_INIT ラベルの位置に、アセンブリ言語で初期化コードを入力します。
3. エディタを終了します。
4. ソース ファイルを前方のコメントを参考にアセンブルし、実行ファイルを作成します。

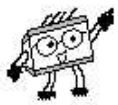
※モニタ プログラムを変更する前には、必ずバックアップを作成して下さい。

モニタ プログラムがロードされるアドレスはコンフィグレーションファイルの指定により自動的に変わります。
従って初期化コードも再配置可能なコードで記述して下さい。

※詳しくは MDX700 User's Manual をご参照下さい。

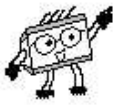
ソース ファイルの変更以外に初期化コードを追加するには、コンフィグレーションファイルへ機械語で追加する方法があります。(後述３-4.項を参照下さい)

ふ～ん。他に気をつけないといけない事はないの？



あります。

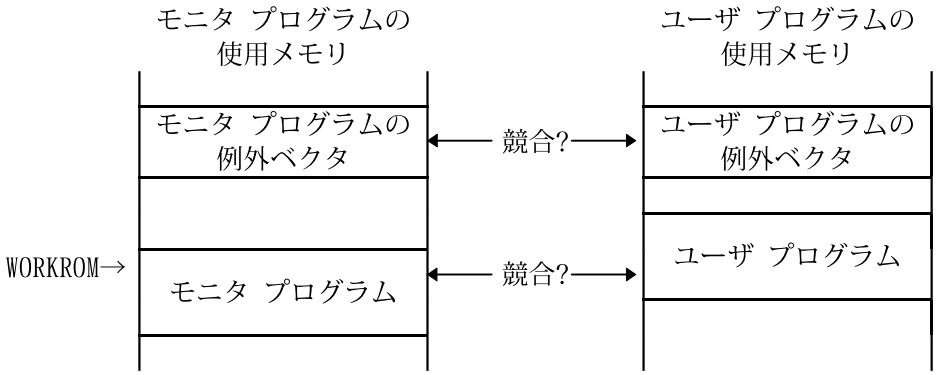
モニタ プログラムとユーザ プログラムが使用するメモリ領域が競合しないようにして下さい。



具体的には次の点になります。

- コンフィグレーション ファイルで指定するモニタ プログラムが常駐する領域をユーザ プログラムが使用しないこと。
- モニタ プログラムの例外ベクタ領域を、ユーザ プログラムで使用しないこと。

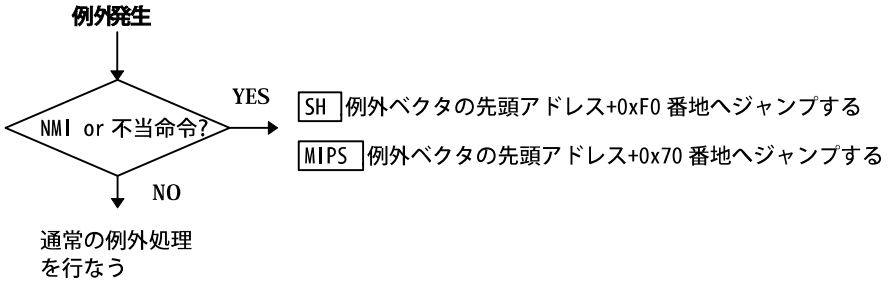
（詳しくは、MDX700 User's Guide 参照下さい）



ユーザプログラムとモニタプログラムが競合している場合は、ユーザプログラムのロード アドレスをモニタプログラムと競合しない位置へ移動して下さい。

例外ベクタのみが競合している場合は、デバッガのコマンドで競合を回避できる場合があります。ユーザ プログラムをダウンロードした後、Ｉコマンドを実行すると、モニタプログラムの例外ベクタが再ロードされます。

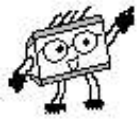
MIPS SH-3の場合、複数の例外がひとつの例外ベクタを使用しているため、どうしても例外ベクタが競合します。これらのＣＰＵを使用する場合は、ＮＭＩと不当命令で使用する例外ベクタの先頭に、次の処理を追加して下さい。



上記の処理はブレークポイントのための不当命令と、強制ブレークのためのＮＭＩが発生した場合のみ、モニタ プログラムへジャンプさせるための処理です。
これらの処理を追加できるように、モニタ プログラムの例外ベクタは二つの分岐命令で構成されています。

リアルタイム OS 等を使用する時は注意が必要だね。

1－3．MDX700 の動作原理



ではここで MDX700 の動作原理について説明します。

デバッガ起動時の内部動作

1. コンフィグレーション ファイルを読み込みます。
2. コンフィグレーション ファイルの WORKROM で指定されたアドレスへ、モニタ プログラムとコンフィグレーション ファイルの内容をダウンロードします。
3. コンフィグレーション ファイルの RESETVECTOR で指定されたアドレスへ、モニタ プログラムをリセット スタートさせるための、リセットベクタをダウンロードします。
4. ＲＥＳＥＴ信号を出力します。ＲＥＳＥＴ信号をターゲット システムに接続している場合は、モニタプログラムがリセットスタートし、モニタ プログラムとの通信を開始します。
5. モニタプログラムから例外発生時のアドレスを取得します。
6. 例外発生時のアドレスから例外ベクタを作成します。
7. 例外ベクタをターゲット システムへダウンロードします。

上記の内部動作により、コンフィグレーション ファイルの WORKROM の変更だけでモニタプログラムは移動できます。また、モニタ プログラムのソース ファイルの中に例外ベクタを記述しないですみます。

ブレークポイント機能

ブレークポイント機能は、ＣＰＵの不当命令例外を利用して実現しています。
ブレークポイントが設定されたアドレスの命令は、ユーザ プログラム実行直前に、不当命令に置き換えられ、ユーザプログラムがブレークした後、自動的に元の命令に戻されます。

ブレークポイントに使用している不当命令は、次のとおりです。

- 0x4AFC 68000
- 0x0000000D MIPS
- 0x00000000 PowerPC
- 0xF00F SH
- 0x7C7C V830
- 0xFFFFFFFF V850
- 0xF840 V850E
- 0x5858 V830 V850 V850E 以外の V800

ステップ実行機能

68000 PowerPC の場合、ステップ実行機能はＣＰＵのトレース例外を利用して実現しています。

MIPS SH V800のＣＰＵの場合、ブレークポイントと同様に不当命令を利用して実現しています。

詳しくは命令実行後のＰＣを計算、その位置にブレークポイント(不当命令)を設定し、ユーザ プログラムを実行します。

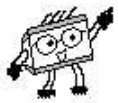
強制ブレーク機能

強制ブレーク機能は、次の例外を使用して実現しています。

- レベル７割り込み 68000
- コンフィグレーションファイルの ABORTVECTOR で指定された例外 PowerPC
- ＮＭＩ例外 MIPS SH V800

デバッガから実行中のユーザプログラムを停止させる操作を行った場合、デバッガは外部トリガ ケーブルのＮＭＩ信号を出力します。これにより、ユーザ プログラムが停止します。

なるほどね。 MDX700 が ROM インサーキットだけで様々な機能を実現できるのに必要なのがモニタ プログラムなんだ。でもモニタ プログラムはユーザ領域で動作するから、ユーザ プログラムとの共存について考慮しないといけないんだね。



そうなんです。特にモニタ プログラムが使用できる領域や例外ベクタの使用に関してはよく確認して下さい。

２．ＭＤＸ７００が
使用できるターゲットは？

対応ＣＰＵやターゲットの制限事項などについて
説明します。

2-1.対応 CPU、ROM

対応 CPU

V800 シリーズ、SuperH RISC engine ファミリ、ＭＩＰＳファミリ、

PowerPC ファミリ、68K ファミリ

※MIPS、SH においてはリトルエンディアンにも対応
※上記以外のＣＰＵをお考えの方は、別途ご相談下さい。

対応 ROM

27256、27512、27010、27020、27040、271000、27C4000、27C8000

271024、274096、29F040、28F400

※上記以外のＲＯＭをお考えの方は、別途ご相談下さい。
※また、MDX700 専用コネクタをターゲット システムへ実装される場合は、
技術フォロー致しますので、ご相談下さい。

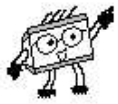
ROM バス幅：8bit、16bit、32bit に標準対応。

64bit は MDX700 を 2 台使用で対応可能。Parallel

ROM アクセスタイム：CS から７５ nsec、ＯＥから５０ nsec

2-2.Flash メモリを ROM として使用する

Ｆｌａｓｈメモリを ROM として使用したいんだけど．．．

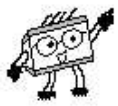


**変換アダプタを使用していただくかＭＤＸ７００専用コネクタを
ターゲット システムに実装して下さい。**

Ｆｌａｓｈメモリは表面実装タイプなのでそのままではＲＯＭプローブを
インサーキットできません。従って同容量のＤＩＰタイプの EPROM と同様の信号線の
配置になるようなＲＯＭソケットをご用意下さい。

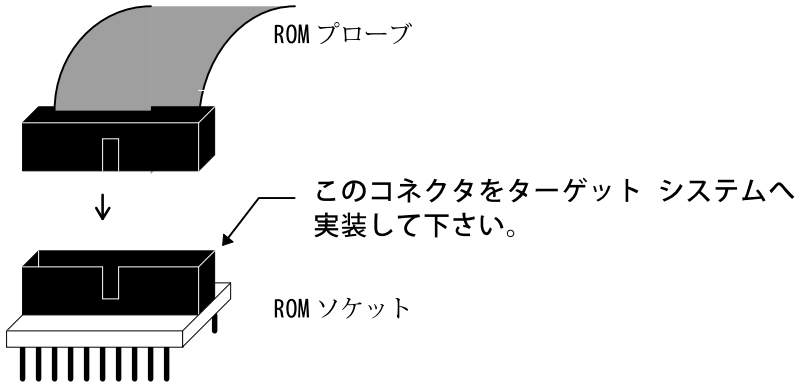
もしくはＭＤＸ７００専用のコネクタをターゲット システム上に実装して下さい。

ＭＤＸ７００専用コネクタはどのように実装するの？



ＭＤＸ７００専用コネクタの実装方法について説明します。

下図の ROM プローブが直接ターゲット システムへインサーキットできるように
コネクタを実装して下さい。



ピン アサインと使用コネクタは下記の通りです。

また、Flash メモリと両方実装される場合は、スイッチなどでＣＳ信号やＯＥ信号の
接続を切り替えられるようにして下さい。

※(重要)使用しないアドレス ピンは OPEN にしておいて下さい。
※ROMCODE については次ページをご覧下さい。

RC28AD/RC28D
（ヒロセ HIF6A-40PA-1.27DSA）

DATA00	A01	B01	ROMCODE0
GND	A02	B02	ROMCODE1
DATA01	A03	B03	ROMCODE2
GND	A04	B04	ROMCODE3
DATA02	A05	B05	ADDR00
GND	A06	B06	ADDR01
DATA03	A07	B07	ADDR02
GND	A08	B08	ADDR03
DATA04	A09	B09	ADDR04
GND	A10	B10	ADDR05
DATA05	A11	B11	ADDR06
GND	A12	B12	ADDR07
DATA06	A13	B13	ADDR08
GND	A14	B14	ADDR09
DATA07	A15	B15	ADDR10
GND	A16	B16	ADDR11
CE*	A17	B17	ADDR12
GND	A18	B18	ADDR13
OE*	A19	B19	ADDR14
GND	A20	B20	ADDR15

RC32AD/RC32D
（ヒロセ HIF6A-52PA-1.27DSA）

DATA00	A01	B01	ROMCODE0
GND	A02	B02	ROMCODE1
DATA01	A03	B03	ROMCODE2
GND	A04	B04	ROMCODE3
DATA02	A05	B05	ADDR00
GND	A06	B06	ADDR01
DATA03	A07	B07	ADDR02
GND	A08	B08	ADDR03
DATA04	A09	B09	ADDR04
GND	A10	B10	ADDR05
DATA05	A11	B11	ADDR06
GND	A12	B12	ADDR07
DATA06	A13	B13	ADDR08
GND	A14	B14	ADDR09
DATA07	A15	B15	ADDR10
GND	A16	B16	ADDR11
CE*	A17	B17	ADDR12
GND	A18	B18	ADDR13
OE*	A19	B19	ADDR14
GND	A20	B20	ADDR15
OPEN	A21	B21	ADDR16
OPEN	A22	B22	ADDR17
OPEN	A23	B23	ADDR18
OPEN	A24	B24	ADDR19
OPEN	A25	B25	ADDR20
OPEN	A26	B26	ADDR21

RC40AD/RC40D
（ヒロセ HIF6A-68PA-1.27DSA）

ROMCODE0	A01	B01	DATA00
ROMCODE1	A02	B02	GND
ROMCODE2	A03	B03	DATA01
ROMCODE3	A04	B04	GND
ADDR00	A05	B05	DATA02
ADDR01	A06	B06	GND
ADDR02	A07	B07	DATA03
ADDR03	A08	B08	GND
ADDR04	A09	B09	DATA04
ADDR05	A10	B10	GND
ADDR06	A11	B11	DATA05
ADDR07	A12	B12	GND
ADDR08	A13	B13	DATA06
ADDR09	A14	B14	GND
ADDR10	A15	B15	DATA07
ADDR11	A16	B16	GND
ADDR12	A17	B17	CEL*
ADDR13	A18	B18	GND
ADDR14	A19	B19	OEL*
ADDR15	A20	B20	GND
ADDR16	A21	B21	DATA08
ADDR17	A22	B22	GND
ADDR18	A23	B23	DATA09
ADDR19	A24	B24	GND
ADDR20	A25	B25	DATA10
ADDR21	A26	B26	GND
OPEN	A27	B27	DATA11
OPEN	A28	B28	GND
DATA15	A29	B29	DATA12
GND	A30	B30	GND
CEU*	A31	B31	DATA13
GND	A32	B32	GND
OEU*	A33	B33	DATA14
GND	A34	B34	GND

0:GND 短絡
1:開放

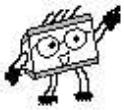
ROMCODE 3	2	1	0	ROM プローブ
0	0	0	0	32K x 8bit
0	0	0	1	64K x 8bit
0	0	1	0	128K x 8bit
0	0	1	1	256K x 8bit
0	1	0	0	512K x 8bit
0	1	0	1	(1024K x 8bit)
0	1	1	0	(2048K x 8bit)
0	1	1	1	(4096K x 8bit)
1	0	0	0	
1	0	0	1	64K x 16bit
1	0	1	0	128K x 16bit
1	0	1	1	256K x 16bit
1	1	0	0	512K x 16bit
1	1	0	1	(1024K x 16bit)
1	1	1	0	(2048K x 16bit)
1	1	1	1	

（ ）内は、MDX700 専用ｺﾈｸﾀ実装時ｻﾎﾟｰﾄ

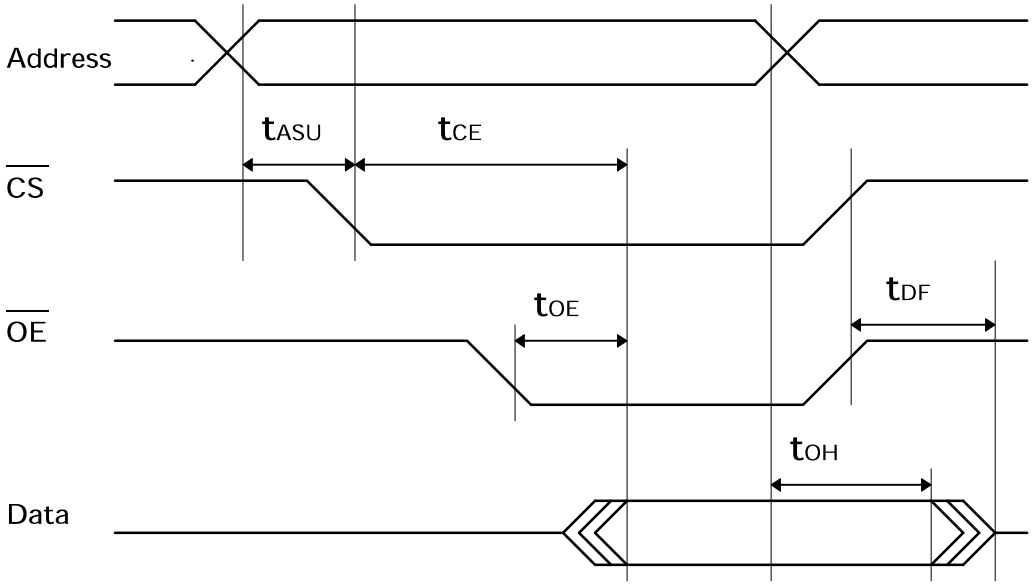
このコネクタを実装した時の回路図をサンプルとしてご参照ください。
（MC 設計サンプル.PDF）

2-3.データ アクセスのタイミング

データ アクセスのタイミングは？

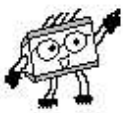


下図を参照下さい。



	min	max	
t_{ASU}	0		CS assert to address valid
t_{CE}		75n sec	CS access time
t_{OE}		50n sec	OE access time
t_{OH}	10n sec		output hold time
t_{DF}		40n sec	output floating time

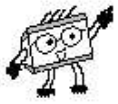
これはターゲット システムを設計する時にハードウェアの
エンジニアと良く話し合わないといけないね。
その時ってフォローはしてくれるの？



はい。いつでもサポートグループか担当営業までお知らせ下さい。

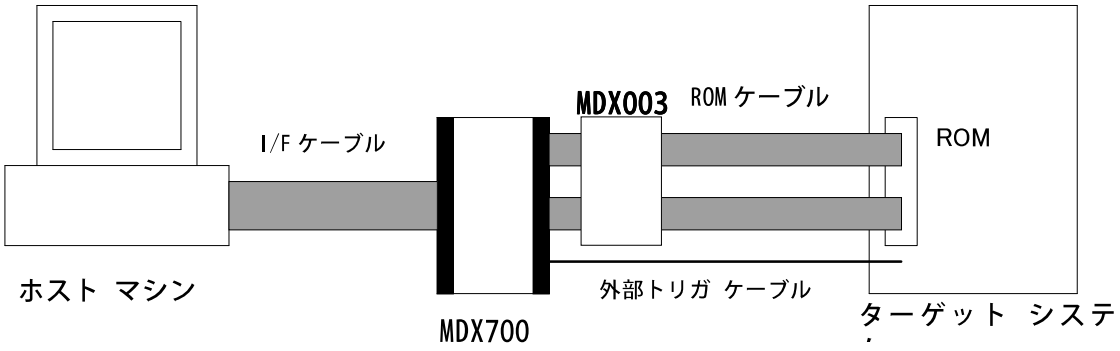
2-4.MDX003（５－３Ｖ Level Converter）

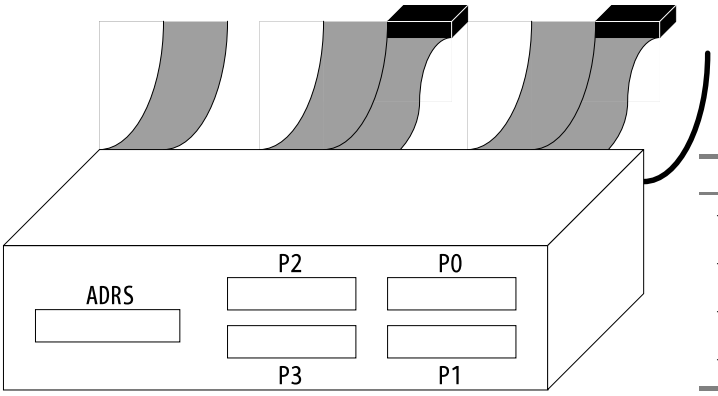
ターゲット システムの ROM 周辺回路が３.３Ｖ仕様
なんだけど そのまま使えるの？



ＭＤＸ７００内部は５Ｖ仕様となっております。

ＲＯＭ周辺回路が３Ｖ仕様の場合は、ＭＤＸ７００とターゲット システムの間に電圧変換アダプタＭＤＸ００３を接続して下さい。

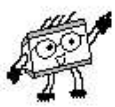




ＭＤＸ００３（5-3VLevel Converter）

	max	min
V_{IH}		2.0
V_{IL}	0.8	
V_{OH}		2.4
V_{OL}	0.4	

このＭＤＸ００３はＭＤＸ７００に添付されているの？

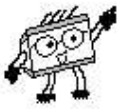


いえ。別途オプションにてご購入いただきます。

定価￥９８，０００ でＭＤＸ７００ １台にＭＤＸ００３が１台必要となります。

2-5.制限事項と注意事項

ターゲット システムで
注意しないといけないことは？

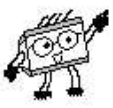


これまで説明してきました内容と以下の内容を合わせて確認して下さい。

ターゲット システムの制限事項

1. ＣＰＵ及びＲＯＭ、ＲＡＭ周辺が、ハード的に完全に動作している。
2. ＲＯＭに８～１６Ｋバイト、ＲＡＭに４Ｋバイトの空き容量がある。
3. ＲＥＳＥＴ信号とＮＭＩ信号を接続できる方が望ましい。
4. ＲＯＭソケットもしくはＭＤＸ７００専用コネクタが実装されている。
5. ＲＯＭがバンク化されていない。
   （ＲＯＭとＲＯＭ以外の資源が、同一アドレス空間を切替えて使用しない）
6. ＲＯＭがキャッシュされていない、又はROMがページアクセスされない。
7. ＲＯＭをバイトでアクセスできる。
   (CPUがROMをバイトアクセスした場合､ROMの指定番地のみアクセスをする)
8. ＲＯＭのアドレス信号が変化するとき、ＣＳまたはＯＥが非アクティブになる。
9. ＲＯＭのＣＳまたはＯＥが上位アドレスをデコードした信号である。
10. ＲＯＭが複数個実装されている場合、すべてのＲＯＭのアドレス信号が同一信号である。
11. ＲＡＭ上でプログラムが実行できる。
12. 68000 ＲＯＭまたはＲＡＭがファンクション コード信号(FC0～FC2)で分割されていない。
13. PowerPC ＣＰＵはビッグ エンディアンのみで動作する。

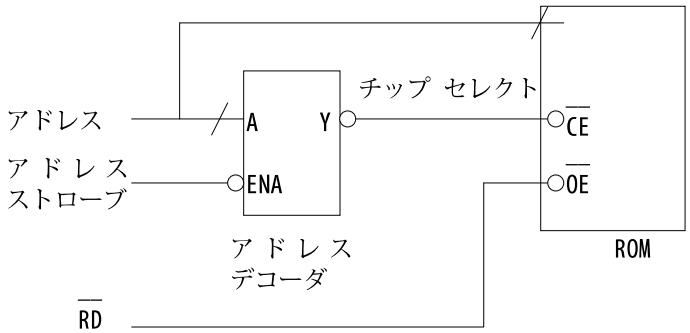
項目 8. 9.についてもう少し 詳しく教えて下さい。



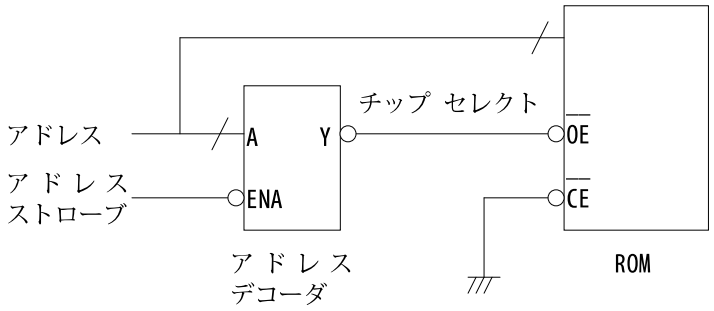
はい。次ページの図を参照下さい。

また、内容についてご質問がありましたらサポートグループか担当営業へ
お問合わせ下さい。

ＭＤＸ７００が動作するターゲット システム

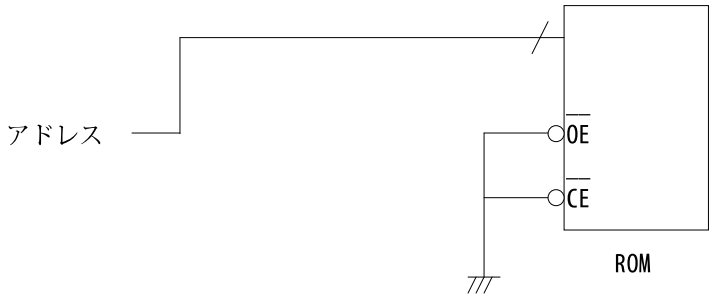


ＭＤＸ７００が条件付きで動作するターゲット システム

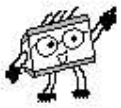


- ＲＯＭ読み込みデータ バス幅が１６ｂｉｔの場合、８ｂｉｔＲＯＭなら２個、１６ｂｉｔＲＯＭなら１個までしか使用できません。
- ＲＯＭ読み込みデータ バス幅が８ｂｉｔの場合、ＲＯＭを１個しか使用できません。

ＭＤＸ７００が動作しないターゲット システム



注意事項



全対応ＣＰＵ共通

1. モニタ プログラムが使用するよう設定されたメモリ領域に、ユーザ プログラムをダウンロードすると、デバッガが使用できなくなります。
2. NMI 信号をターゲット システムへ接続していない場合は、デバッガの操作で、ユーザ プログラムを強制ブレークさせないで下さい。

MIPS

1. ユーザプログラムは、カーネルモードで TLB を使用しないメモリ空間を使用して下さい。(0x80000000～0xBFFFFFFF)
2. ユーザプログラムは、Ｒ２７レジスタを使用できません。
   Ｒ２７レジスタはモニタ プログラムが使用しています。
3. ユーザプログラム実行中に、ステータスレジスタのＢＥＶビットを変更しないで下さい。(通常１)
4. 例外ハンドラ内では、ステータスレジスタＥＸＬをクリアした以降の命令上にしか、ブレークポイントを設定できません。
   (多重割り込みを許可する)
5. また EPC/ErrorEPC のレジスタをスタックに退避した以降の命令上にしか、ブレークポイントを設定できません。

V800

1. 次の命令はステップ実行できません。
   HALT、LDSR、RETI、TRAP
2. 例外ハンドラ内では、EIPC/EIPSW/FEPC/FEPSW のレジスタをスタックに退避した以降の命令上にしか、ブレークポイントを設定できません。

V830

1. 次の命令はステップ実行できません。
   BRKRET、 STBY
2. 例外ハンドラ内では、EIPC/EIPSW/FEPC/FEPSW のレジスタをスタックに退避した以降の命令上にしか、ブレークポイントを設定できません。

SH-3

1. TRAPA 命令上にブレークポイントを設定できません。ブレークポイントで止まることはできますがそこから継続して実行することができません。継続実行する場合には、TRAPA 命令上のブレークポイントを解除する必要があります。
2. 例外ハンドラ内では、SPC/SSR のレジスタをスタックに退避した以降の命令上にしか、ブレークポイントを設定できません。

SH | 68000

1. VBR レジスタを、コンフィグレーション ファイル中で初期化して下さい。ユーザプログラム実行中に VBR レジスタの値が変更されると、デバッガが使用できなくなります。

PowerPC

1. 例外ハンドラ内では、SRR0/SRR1 のレジスタをスタックに退避した以降の命令上にしか、ブレークポイントを設定できません。

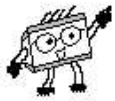
３．ＭＤＸ７００を
使用してみる！

ではＭＤＸ７００を実際の手順に添って動作させて
みましょう。

3-1.製品構成

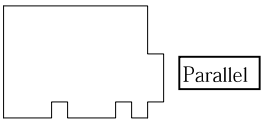
ＭＤＸ７００購入時には何が含まれているの？

ではＭＤＸ７００の標準構成を紹介します。

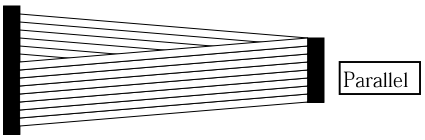


ＭＤＸ７００本体

電源ケーブル

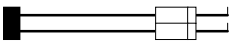


I/F ボード

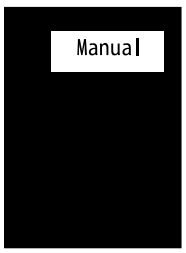


I/F ケーブル

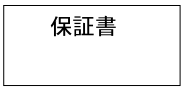
ROM プローブ、ソケット



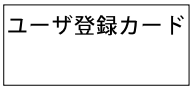
外部 トリガ ケーブル



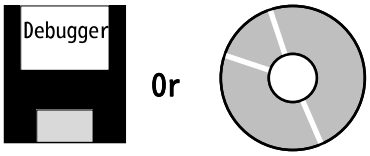
User's Manual



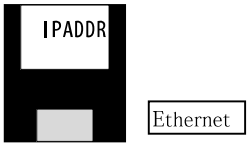
保証書



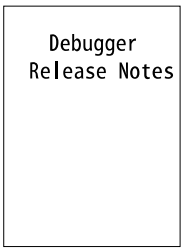
ユーザ登録カード



デバッガ（3.5" or CD-ROM）



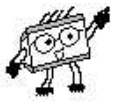
IP アドレス設定用 FD（3.5"）



リリースノート

3-3.I/F ボードの設定 Parallel

ＭＤＸ７００のI/F ボードってホスト マシンのどこへ接続するの？

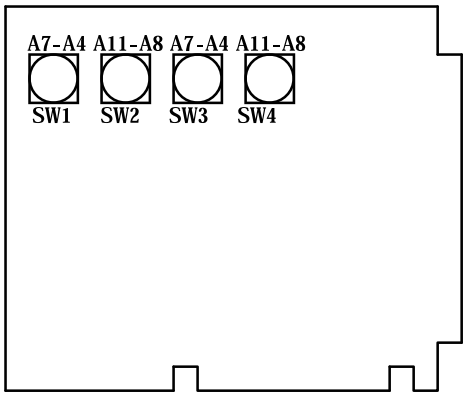


PC/AT はＩＳＡスロット、PC-98 シリーズは C-BUS スロットへ挿入します。

またI/F の通信はパラレルとなっており、以下のI/O アドレス空間を使用しています。
Windows95 の Plug & Play には対応しておりませんので、他のツール等で既に使用されているI/O アドレスと重なる場合はI/F ボード上のスイッチでご変更下さい。

PC/AT ISA ボード

- 0x0100～0x010F 及び 0x0110～0x011F



SW1 第一 I/O アドレスの A7-A4
（出荷時、0）

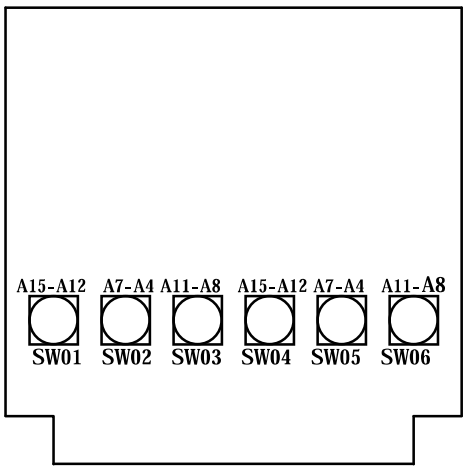
SW2 第一 I/O アドレスの A11-A8
（出荷時、1）

SW3 第二 I/O アドレスの A7-A4
（出荷時、1）

SW4 第二 I/O アドレスの A11-A8
（出荷時、1）

PC-98 C-BUS ボード

- 0x01D0～0x01DF 及び 0x02D0～0x02DF



SW01 第一 I/O アドレスの A15-A12
（出荷時、0）

SW02 第一 I/O アドレスの A11-A8
（出荷時、1）

SW03 第一 I/O アドレスの A7-A4
（出荷時、D）

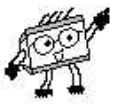
SW04 第二 I/O アドレスの A15-A12
（出荷時、0）

SW05 第二 I/O アドレスの A11-A8
（出荷時、2）

SW06 第二 I/O アドレスの A7-A4
（出荷時、D）

3-3.ＲＯＭプローブ・外部 トリガ ケーブルの接続

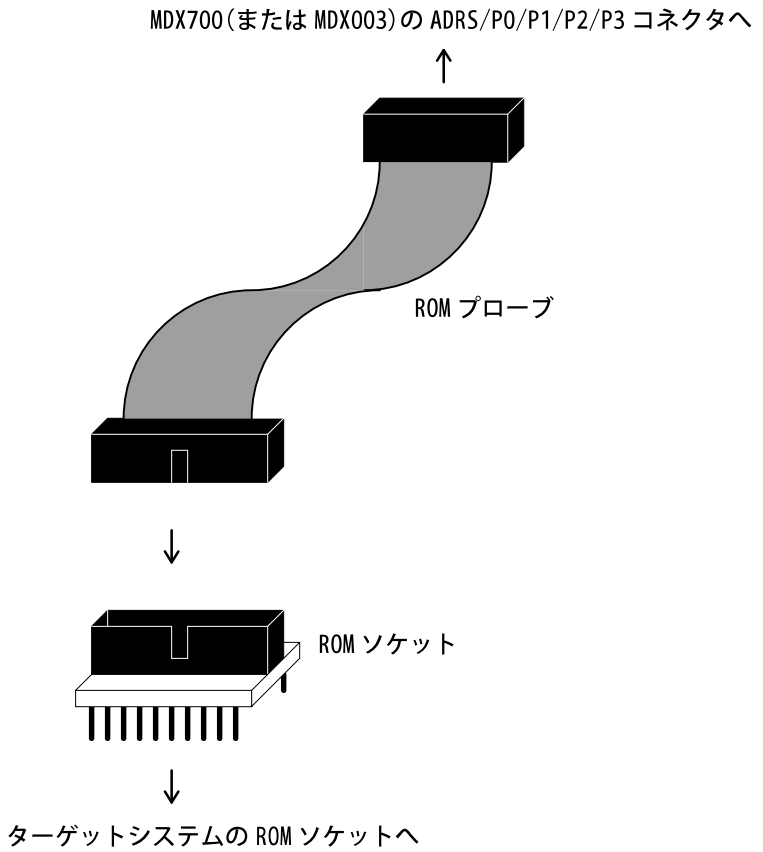
ＭＤＸ７００ではターゲット システムとの接続は
ＲＯＭプローブのみでしたよね。



そうです。 あと外部 トリガ ケーブルも接続できると便利ですよ。

ＭＤＸ７００(またはＭＤＸ００３)とターゲット システムは、ＲＯＭプローブとＲＯＭソケット接続します。接続手順は次のとおりです。

はじめに、すべてのＲＯＭソケットにＲＯＭプローブを装着して下さい。
ＭＤＸ７００専用コネクタを実装された方は、直接ＲＯＭプローブを装着して下さい。



つぎにＲＯＭソケットをターゲットシステムへ接続し、もう一方のＲＯＭプローブをＭＤＸ７００(またはＭＤＸ００３)の ADRS/P0/P1/P2/P3 コネクタへ接続します。

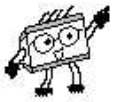
※ＲＯＭプローブとＭＤＸ７００との接続の詳細は User's Manual を参照下さい。

※各種接続は、ＭＤＸ７００及びターゲット システムの電源が必ず off になっていることを確認した上で行って下さい。

※ＲＯＭプローブ、ＲＯＭソケットは逆差し等しないようご注意下さい。

外部 トリガ ケーブルはどこへ接続するの？
接続するとなぜ便利なの？

外部 トリガ ケーブルはＲＥＳＥＴ信号とＮＭＩ信号に接続します。

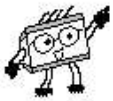


これらを接続することにより、次のことが可能になります。

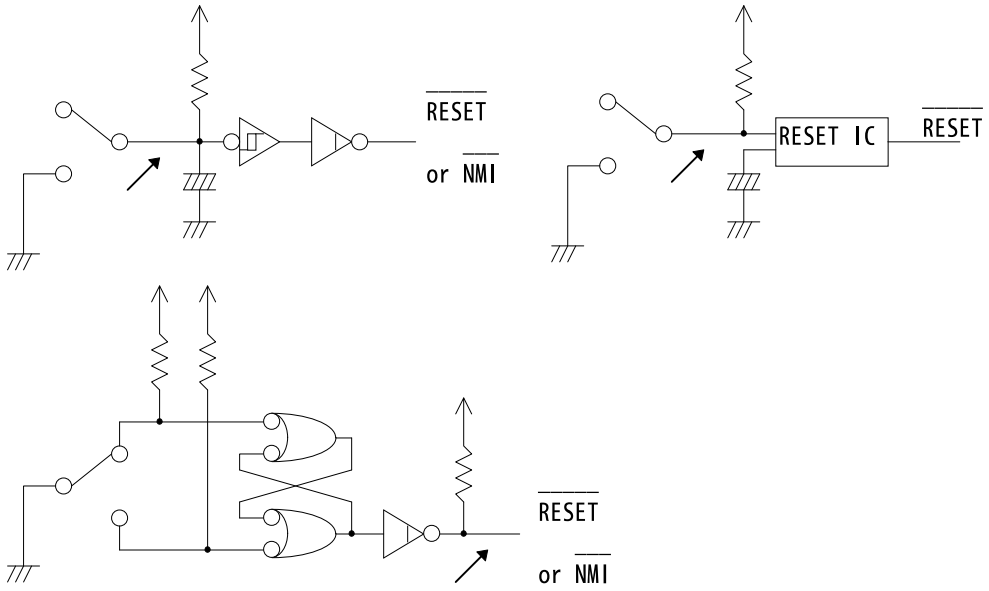
- ＲＥＳＥＴ信号の接続で、余分な手間をかけず起動できます。
  ※未接続時には、手動でターゲットシステムにＲＥＳＥＴをかける必要があります。
- ＮＭＩ信号の接続により、ユーザ プログラム実行中、もしくは暴走時にデバッガのコマンドにより強制ブレークを実行できます。

外部 トリガ ケーブルが接続できる推奨回路を教えて！

下図のような回路であれば問題ありません。



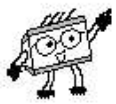
ＭＤＸ７００から出力されるＲＥＳＥＴ信号とＮＭＩ信号は、負論理オープンコレクタ出力（７４０６相当）です。ターゲット システムのリセット生成回路やＮＭＩ生成回路上でプルアップ抵抗が付加されており、グランドにショートしても損傷しない個所へ接続して下さい。下図の ↗ 位置がこれに相当する回路の例になります。



※PowerPC でＮＭＩ信号を接続する場合、後述のコンフィグレーションファイル中の ABORTVECTOR を必ず設定して下さい。

3-4.コンフィグレーション ファイル

I/F ボードの設定をしてホスト マシンへ組込んだし
ターゲット システムへも ROM プローブを接続したし
これでＭＤＸ７００を使えるね。

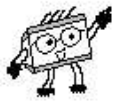


もう１つ設定をする必要があります。

１-２.項でＭＤＸ７００が機能するためにモニタプログラムがあり、ターゲット システムへ常駐すると説明しました。

このモニタ プログラムを常駐させる位置や CPU の種類などターゲット システムの環境を設定するファイルがコンフィグレーション ファイルです。

ふ～ん。それはどんな風に作成するの？



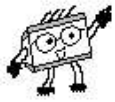
デバッガと共にＭＤＸ．ＣＦＧというファイルが添付されています。

このファイルを変更するか参考にして、ターゲットシステムに合わせたファイルを作成して下さい。
この時にターゲットシステム毎にファイル名を変更されると便利です。

※使用するコンフィグレーション ファイルは環境変数 MDX_CFG= *filename* で指定できます。（デフォルトはＭＤＸ．ＣＦＧです。）

コンフィグレーション ファイルの例

```
*
*  MDX700 configuration for DVE-V830/20 (DENSAN V830 VME board)
*
MONITOR         mdxv800.abs     ; monitor program
CPU             V830            ; CPU type
PORT            0x0100          ; Host interface I/O address
BUS             32              ; bus width
ROM             0x7FFC0000      ; ROM start address
ROM_IMAGE       0xFFFC0000      ; ROM image start address
ROMSIZE         0x00040000      ; ROM size
WORKROM         0x7FFFD000      ; work ROM start address
WORKROMSIZE     0x00002000      ; work ROM size
WORKRAM         0x00010000      ; work RAM start address
WORKRAMSIZE     0x00001000      ; work RAM size
RESETVECTOR     0x7FFFFFF0      ; RESET vector addres(ffffffff = not used)
TIMER           200000          ; RESET & communication port timeout
*
*  Register Initialize
*
REG_R3          0x00008000      ; stack pointer
REG_PC          0x00002000      ; program counter
```



このコンフィグレーション ファイルの設定は正確に行って下さい。

ここで誤りがありますと、ＭＤＸ７００を起動することができなくなります。
また新規に作成される場合はコメント行以外の行を省略したり、項目名を変更したりしないで下さい。

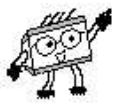
※確実にコンフィグレーションファイルを作成する為に次の情報を確認して下さい。

- **Ｉ／Ｆボードの Ｉ／Ｏアドレス** Parallel
- **ＣＰＵ名**
- **ＲＯＭ読み込みデータ バス幅**
- **ＲＯＭの先頭アドレスと容量（ＲＯＭプローブが接続されたＲＯＭ領域）**
- **モニタ プログラムに開放できるＲＯＭ領域**
- **モニタ プログラムに開放できるＲＡＭ領域**

この設定が誤っているために動作しないといったケースが多々あります。
特にＲＯＭの容量やモニタプログラムに開放する領域などご注意下さい。

※ コンフィグレーションファイルの詳細は、User's Manual 参照下さい。

そういえば、ターゲット システムの初期化コードを
コンフィグレーション ファイルに追加できるんでしたよね？



はい。ただし 16 進数の機械語でしか記述できませんのでご注意下さい。

ただモニタプログラムのバージョンが更新された時に、再度モニタプログラムを変更するといったことが、この方法だと必要ありません。

コンフィグレーションファイルに機械語を記述する場合は、次のように INIT_CODE という項目を使用します。ひとつの機械語のサイズは、ＣＰＵの命令の最小サイズです。

ＣＰＵのエンディアンに関係なく上位ビットから記述して下さい。

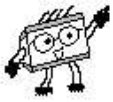
※ コンフィグレーションファイルに記述した初期化コードは、モニタプログラムのソースに記述した初期化コードより優先されます。

追加の例（以下をコンフィグレーションファイルの最後尾に追加します。）

```
*
*  USER_INIT code
*
INIT_CODE       0xfc00          ; out.w   r0, 0x007C[r0]
INIT_CODE       0x007c
INIT_CODE       0x181f          ; jmp     [lp]                  # return
```

3-5.電源投入

さぁコンフィグレーション ファイルの設定もできたし
もう大丈夫だよね？



設定が間違っていなければこれでＭＤＸ７００を使用する準備は完了しました。

では電源を投入しましょう。機器の電源投入は、次の手順で行なって下さい。

1. ホスト マシン
2. ＭＤＸ７００
3. ターゲット システム

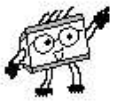
機器の電源切断は、次の手順で行なって下さい。

1. ターゲット システム
2. ＭＤＸ７００
3. ホスト マシン

※電源投入および切断の手順を間違えると、機器が破壊される場合があります。
※電源投入時には、機器の接続および着脱をしないで下さい。

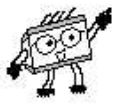
電源は投入できたけど、どうすればＭＤＸ７００で
ターゲット システムの情報を見られるの？

それでは、ＭＤＸ７００で対応しているデバッガについて説明しましょう。



3-6.対応デバッガとその使用方法

ＭＤＸ７００は、２種類のデバッガに対応しています。



１つはＣ/Ｃ++、アセンブラ ソースレベルデバッグを行う為の高級言語デバッガです。
現在ＭＤＸ７００は、ＭＵＬＴＩ・ＳｉｎｇｌｅＳｔｅｐ・ＸＲＡＹといった高級言語デバッガに対応しています。

これらのツールと併用することで、C++/C のソースレベルデバッグを可能にします。
具体的には、ソースライン上へのブレークポイントの設定、ソース内変数の参照と変更、ソースライン毎のステップ実行といった機能を実現できます。

※高級言語デバッガの詳細は、それぞれのカタログやマニュアルを参照下さい。

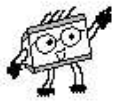
もう１つは簡易デバッグに用いるＭＤＸＤＥＢです。

これはＭＤＸ７００の立ち上げやターゲットシステム上のメモリのチェック、ダウンローダとした使い方など初期段階や最終工程での使用に適しています。

また、各 高級言語デバッガが持つ特徴をそのままにＭＤＸウィンドウにより
ＭＤＸＤＥＢのコマンドも使用することもできます。

へぇ～。 ＭＤＸＤＥＢの使い方を教えて！

ＭＤＸＤＥＢの使用方法について説明します。



ＭＤＸＤＥＢは MS-DOS 版と Windows 95／3.1 版があります。

MS-DOS 版

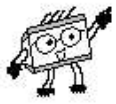
ＭＤＸＤＥＢ．ＥＸＥを実行します。

Windows 95／3.1 版

ＭＤＸＤＥＢＷ．ＥＸＥを実行します。

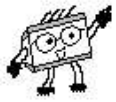
あとはコマンドを入力するだけのシンプルなデバッガです。

ＭＤＸＤＥＢでできることは



次ページＭＤＸＤＥＢのコマンド一覧をご覧下さい。

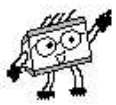
ＭＤＸＤＥＢでは以下のコマンドをサポートしています。



各コマンドのシンタックスや詳細は User's Manual を参照下さい。

ラインアセンブル（Ａ）：メモリの内容をアセンブルコードで変更(MIPS 系のみ)
ブレークポイント（Ｂ）：ブレークポイントを表示、設定（最大６４個）
ＣＦＧの内容確認（Ｃ）：コンフィグレーション ファイルの内容を表示
メモリダンプ　（Ｄ）：メモリの内容をバイト/ワード/ロングで表示
メモリの変更　（Ｅ）：メモリの内容をアドレス単位で変更
（Ｆ）：指定範囲内のメモリの内容を指定データで変更
プログラムの実行（Ｇ）：現在のＰＣもしくは指定したアドレスからプログラムを実行
ヘルプの表示　（Ｈ）：ヘルプメッセージを表示
ＭＤＸの初期化　（Ｉ）：モニタプログラムを再ロードし、ＭＤＸ７００再初期化
ＭＤＸメモリテスト（Ｋ）：ＭＤＸ７００のエミュレーションメモリをテスト
ダウンロード　（Ｌ）：ＭＤＸバイナリファイル、Ｓレコード、インテルＨＥＸ、ＣＯＦＦファイルをメモリへダウンロード
高速ダウンロードにはＭＤＸバイナリファイルを使用
メモリ内容の転送　（Ｍ）：指定範囲内のメモリ内容を指定アドレス以降に転送
Ｉ／Ｏのアクセス　（Ｐ）：Ｉ／Ｏポートの内容を表示/変更
ＭＤＸＤＥＢの終了（Ｑ）：ＭＤＸＤＥＢを終了
レジスタのアクセス（Ｒ）：レジスタの内容を表示/変更
ＦＰＵﾚｼﾞｽﾀの内容も表示可能。
ステップ実行　（Ｓ）：プログラムを１ラインづつ実行
（ＭＵＬＴＩ制御）（Ｔ）：MULTI デバッガを使用時、restart 時の再ロードを制御
オフにすることで再実行が速くなります。
バージョン表示　（Ｖ）：モニタプログラムなどのバージョンを表示

高速ダウンロードに使用する
MDX バイナリファイルって何？



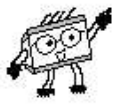
これはＭＤＸＣＶＴ.ＥＸＥを利用して S レコード、IEEE695 ファイルから作成するＭＤＸ７００高速ダウンロード専用のバイナリファイルです。

ＲＯＭ領域内にありＷＯＲＫＲＯＭで指定した領域と重複しないオブジェクトを変換します。

ＭＤＸＣＶＴでもコンフィグレーション ファイル内の項目を参照しています。

3-7.ＭＤＸ７００の起動

さあ、それではＭＤＸ７００を使用してみましょう。

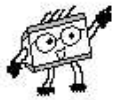


まず３-５.項の手順にそって電源を投入して下さい。
そしてまずはＭＤＸＤＥＢを使用して設定等に誤りがなく正常に起動できるか
どうか確認してみましょう。

```
> MDXDEB
MDXDEB - MDX700 Debugger version 3.45 (V830)
MDXERR: communication port timeout:
Initialize error - run monitor program, then type I command.
>
```

あっ。エラー メッセージが出た！

デバッガとＭＤＸ７００が正常に通信できていないようですね。



外部トリガ ケーブルの RESET 信号はターゲット システムと接続されていますか。
接続されていなければ、ここでターゲット システムを手動でリセットした後に
Ｉ コマンドを入力して下さい。

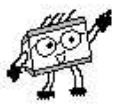
もし 推奨のＲＥＳＥＴ信号を接続しているにもかかわらずこのエラーメッセージが
出るようでしたら３-8.項のエラー メッセージとトラブルシューティングを参照
下さい。

じゃ、ターゲット システムの RESET スイッチを押してっと

```
> I
MDXDEB - MDX700 Debugger version 3.45 (V830)
>
```

うん。今度はエラーなく起動できたよ。

そうしましたら、ＭＤＸＤＥＢのコマンドを使用してみましょう。



ＲＯＭやＲＡＭの内容を参照してみたり、変更してみたりしてターゲットシステム
との通信が正常に行えることを確認して下さい。

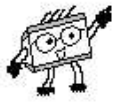
問題がなければ、高級言語デバッガを起動してアプリケーションのデバッグを
進めます。

※高級言語デバッガの取扱いについては、それぞれのマニュアルをご参照下さい。

3-8.エラー メッセージとトラブルシューティング

使用中に発生する可能性の高いエラー メッセージについて教えて？

わかりました。過去のサポートで問合わせの多かったものを紹介します。



その他のエラー メッセージについては、MDX700 User's Manual を参照下さい。

MDXERR: bad communication port

Ｉ/Ｆボードにアクセスできません。
ＭＤＸ７００の電源が投入されているか、確認して下さい。また、Ｉ/Ｆボードのスイッチの設定とコンフィグレーション ファイルの PORT の値が一致しているか、確認して下さい。

MDXERR: bad configuration file: xxxx

コンフィグレーション ファイルの内容が間違っています。
設定内容が正しいか確認して下さい。また(boundary)が表示された場合、通信ポートの先頭アドレスが要求バウンダリ上になるように、WORKROM と WORKROMSIZE を設定して下さい。　詳しくは、User's Manual 付録Ｉを参照下さい。

MDXERR: bad monitor file:

モニタ プログラムの内容が正しく読み込めません。
モニタ プログラムが、Ｓレコード ファイルになっているか、確認して下さい。

MDXERR: communication port timeout:

デバッガとモニタ プログラムとの通信が行なえません。
ＭＤＸ７００とターゲット システムの接続が正しいか、確認して下さい。

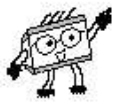
MDXERR: communication port timeout: (but monitor was started)

デバッガとモニタ プログラムとの通信が行なえません。
しかし、モニタ プログラムの起動には成功していました。WORKRAM のアドレスにＲＡＭが存在するか、確認して下さい。また、モニタ プログラムを変更された場合、変更したコードが正しいか、確認して下さい。

サポートへの問合わせの内、コンフィグレーションファイルの設定ミスが、ほとんどで、次いで モニタプログラムの修正ミスとなっています。

もしエラー メッセージが出たらどうやって確認すればいい

問合わせの多いエラーメッセージについて対処方法を説明します。



MDXERR: communication port timeout: デバッガとモニタ プログラム間の通信が行なえないことをあらわすエラー メッセージであると説明しました。
使用されている過程で最も出現する可能性が高いエラーです。
このエラーが表示された場合は、まず、次の項目を確認して下さい。

- ＭＤＸ７００とターゲット システムの電源が投入されているか
- ＭＤＸ７００とターゲット システムの接続が正しいか、ＲＯＭプローブを逆差ししていないか
- コンフィグレーション ファイルの設定が正しいか

もし、エラーの原因がわからない場合は、次の手順にしたがって下さい。
エラーの原因をある程度特定できます。

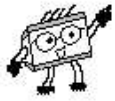
ステップ１

1. ＲＥＳＥＴ信号を接続している場合は、それを外す
2. モニタプログラムを変更している場合は、出荷時のものに戻す
3. ＭＤＸＤＥＢを起動する(エラーは無視する)
4. ＥコマンドでＲＯＭ領域のメモリを書き換える
5. Ｄコマンドで書き換えたメモリの内容を確認する

ＲＯＭ領域の書き換えができる場合、次の項目が正しく設定されています。
ＲＯＭ領域の書き換えができない場合、これらの項目をもう一度確認して下さい。

- ＭＤＸ７００とホスト マシンの接続
- Ｉ/Ｆボードのスイッチの設定 Parallel
- コンフィグレーション ファイルの PORT Parallel

ステップ 2



1. 手動でターゲット システムをリセットする
2. Ｖコマンドでバージョンを表示する

エラーなく実行できる場合は、さらに次の項目が正しく設定されています。
エラーが表示される場合、これらの項目を、もう一度確認して下さい。

- ＭＤＸ７００とターゲット システムの接続
- コンフィグレーション ファイルの BUS
- コンフィグレーション ファイルの ROM
- コンフィグレーション ファイルの ROMSIZE
- コンフィグレーション ファイルの WORKROM
- コンフィグレーション ファイルの WORKROMSIZE
- コンフィグレーション ファイルの RESETVECTOR

ステップ 3

1. Ｉコマンドで再初期化をする

エラーなく実行できる場合は、さらに次の項目が正しく設定されてます。
エラーが表示される場合、これらの項目を、もう一度確認して下さい。

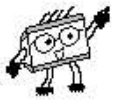
- コンフィグレーション ファイルの WORKRAM
- コンフィグレーション ファイルの WORKRAMSIZE
- コンフィグレーション ファイルの TIMER

ステップ 4

1. ＲＥＳＥＴ信号を使用する場合、ターゲットシステムへ接続してから、
   もう一度ステップ１の３.から繰り返して下さい。
   エラーが起こる場合、ターゲット システムの接続先を確認して下さい。
   また､コンフィグレーション ファイルの TIMER を増やしてみて下さい。

2. モニタプログラムを変更している場合、変更後のモニタプログラムを
   指定して、もう一度ステップ１の３.から繰り返して下さい。
   エラーが起こる場合､変更した部分が間違っていないか確認して下さい。

ＭＤＸ７００が故障しているかどうか切り分ける方法は？



本体のエミュレーションメモリが故障していないか確認します。

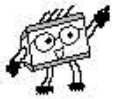
ＭＤＸＤＥＢが正常に起動できた後、Ｋコマンドを実行する。

ＭＤＸ７００のエミュレーションメモリに対してＲｅａｄ/Ｗｒｉｔｅチェックを行い、問題なければ ＯＫと表示される。

もしここでエラーが出た場合は、エラーが起こったアドレス、データ、期待値を表示しますので、その情報をもとにサポートグループまでお問合わせ下さい。

もし、どう確認しても次のステップに進めなかったら

その場合は、下記のサポートグループまでお問合わせ下さい。



この時、次の点をご連絡下さい。

- ターゲット システムの情報
  ＣＰＵ、ＲＯＭの種類と個数、ＲＯＭとＲＡＭの先頭アドレスと容量
  ＭＩＰＳ、ＳＨはエンディアン など
- 現在使用されているコンフィグレーション ファイルの内容
- 表示されるエラー メッセージ
- 操作内容(電源投入からエラーが出力されるまで)

サポートグループ連絡先

ＴＥＬ 03−3392−3331 ＦＡＸ 03−3393−3878

E-mail ＺＡＸＳｕｐｐｏｒｔ@ｌｉｇｈｔｗｅｌｌ．ｃｏ．ｊｐ

ＭＤＸ７００の基本について説明してまいりました。

詳細については、MDX700 User's Manual をご参照下さい。

索　引