

Debug Solutions Power Debugger

Debug System for Boundary Scan Board

コマンドリファレンス 付録-E

Deviceファイル解説

(EEPROMメモリ編)

Debug Solutions

Power Debugger

ご注意

1. このソフトウェアの著作権は、Debug Solutions社にあります。
 2. このソフトウェアおよびマニュアルの一部または全てを無断で使用、複製することはできません。
 3. ソフトウェアは、コンピュータ1台につき1セット購入が原則となっております。
 4. このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用可能です。
 5. このソフトウェアおよびマニュアルを運用した結果の影響については、いっさい責任をおいかねますのでご了承ください。
-

目次

1. DEVICEファイル概要	1
2. EEPROM DEVICEファイルで使用される単語.....	2
2. 1 キャラクタセット	2
2. 2 単語	2
2. 3 コメント	2
3. 文法定義.....	3
3. 1 EEPROMメモリDEVICEファイルの構造について.....	3
3. 2 コマンド構造について.....	4
3. 3 DEVICEファイルコマンドリスト.....	4
4. コマンド詳細.....	5
4. 1 DEVVER コマンド	5
4. 2 DEVMAKER コマンド.....	6
4. 3 DEVNAME コマンド.....	7
4. 4 DEVDEF コマンド.....	8
4. 4. 1 <i>Type</i> 定義.....	9
4. 4. 2 <i>Data</i> 定義.....	10
4. 4. 3 <i>Word</i> 定義.....	11
4. 4. 4 <i>SlaveAdr</i> 定義.....	12
4. 4. 5 <i>AdrByte</i> 定義.....	13
4. 4. 6 <i>PageSize</i> 定義.....	14
4. 4. 7 <i>WrTime</i> 定義.....	15
4. 4. 8 <i>ReadCont</i> 定義.....	16
4. 4. 9 <i>CS</i> 定義.....	17
4. 4. 10 <i>Protect</i> 定義.....	18
4. 4. 11 <i>Hold</i> 定義.....	19
4. 5 INSTSET コマンド	20
4. 5. 1 <i>Write</i> 定義.....	21
4. 5. 2 <i>Read</i> 定義.....	22
4. 5. 3 <i>Erase</i> 定義.....	23
4. 5. 4 <i>WriteSR</i> 定義.....	24
4. 5. 5 <i>ReadSR</i> 定義.....	25
4. 5. 6 <i>WriteALL</i> 定義.....	26
4. 5. 7 <i>EraseALL</i> 定義.....	27

4. 5. 8 <i>PageWR</i> 定義	28
4. 5. 9 <i>WriteEn</i> 定義.....	29
4. 5. 10 <i>WriteDs</i> 定義.....	30

1. Deviceファイル概要

EEPROMメモリDeviceファイルはオンボード書き込みを行うメモリデバイスの特性を表現したテキストファイルで、Power Debugger固有のフォーマットです。EEPROMメモリの種別や、インストラクションセットを定義したデバイス毎のファイルを用意することで、デバイスの設定を簡単に行うことができます。

2. EEPROM Deviceファイルで使用される単語

2. 1 キャラクタセット

Deviceファイルでの単語表記は大文字と小文字の認識は起こりません。又使用できる文字は以下のとおりです。

- 1)大文字、小文字 :A, B, C...Z , a, b, c...z
- 2)数字 :0-9
- 3)特殊文字 : “ & ‘ () * , - . ; < = > _ その他の特殊文字はコメントとしては使用可能です。
- 4)区切り文字 :空白文字(スペース) Tab文字

2. 2 単語

BSDLで使用される単語は、1文字または複数のアルファベット文字、数字或いはアンダースコア(_)から構成されます。又、単語の文字数の上限はありません。以下の文字は有効な単語とみなされます。

```
DCF // 有効な単語
DCF_FILE_No1 // 有効な単語
```

2. 3 コメント

2つのスラッシュ(//) から行の最後までがコメントとしてみなされます。改行のみ、もしくは空白行は、//で定義しなくても無視されます。

3. 文法定義

3.1 EEPROMメモリDeviceファイルの構造について

EEPROMメモリDeviceファイルの構造は以下のような構造をとらなければなりません。

```
<DevVer コマンド>  
[<DevMaker コマンド>]  
[<DevName コマンド>]  
{<DevDef コマンド>}  
{<InstSet コマンド>}
```

※Deviceファイル表記においては、その要素順序は上記の構造であらわされる順序をまもらなければなりません。

第3章 文法定義

3. 2 コマンド構造について

Deviceファイルのコマンド構造は以下のような構成をとります。

EEPROMメモリDeviceファイルのコマンドは、以下の書式で構成されます。

コマンド パラメータ1 パラメータ2 パラメータn

コマンド : DevVer , DevMaker , DevName , DevDef , InstSet

パラメータ : コマンドの種類により、必要なパラメータの数は異なります。

コマンド及びそのパラメータは1行で完結し、空白もしくはTABで区切られます。パラメータの数はコマンドによって0もしくは複数のパラメータを持ちます。又パラメータで定義される数値は、10進数もしくは16進数で表現され、16進数で表現する場合は先頭に0xを付与する必要があります。

例:

```
DevDef Sector 40 // 10進数
Sector 0x0 0x00000000 0x0000ffff // 16進数
```

3. 3 Deviceファイルコマンドリスト

EEPROMデバイスファイルで使用されるコマンドを以下に示します。

DevVer	: デバイス定義ファイルバージョンを指定
DevMaker	: デバイスメーカー定義
DevName	: デバイス型各
DevDef	: デバイス信号情報定義
InstSet	: インストラクションセット定義

4. コマンド詳細

4.1 DevVer コマンド

DevVer コマンドは、EEPROMメモリ定義ファイルのバージョンを定義します。

構文 :DevVer *Maj*バージョン *Min*バージョン

Majバージョン : major バージョン[文字列型]

Minバージョン : minor バージョン[文字列型]

例 :DevVer 1 1

※現在のEEPROMメモリ定義ファイルのバージョンは 1.1 です。

第4章 コマンド詳細

4. 2 DevMaker コマンド

DevMaker コマンドは、EEPROMメモリデバイスのメーカーに関する属性を定義します。

構文 :DevMaker *Maker名*

Maker名 : EEPROMメモリデバイスメーカー名[文字列型]

例 :DevMaker ATMEL

※DevMaker コマンドは定義されていなくても、エラーにはなりません。

4. 3 DevName コマンド

DevName コマンドは、EEPROMメモリデバイス名に関する属性を定義します。

構文 :DevName デバイス名

デバイス名 : EEPROMメモリデバイス名[文字列型]

例 :DevName AT24C64A

※DevName コマンドは定義されていなくても、エラーにはなりません。

第4章 コマンド詳細

4. 4 DevDef コマンド

DevDef コマンドは、EEPROMメモリのデバイス設定に関する属性を定義します。

```
DevDef Parameter1 [Parameter2] [Parameter3]
```

Parameter1 : 属性定義定数

Type	: EEPROM種別 (I2C/SPI/MicroWire) の定義
Data	: データビット数の定義
Word	: ワード数の定義
SlaveAdr	: I2CデバイスのSlaveアドレスの定義
AdrByte	: アドレスのバイト数定義
PageSize	: ページサイズ定義
WrTime	: EEPROMデバイスの書き込み時間定義
ReadCont	: MicroWireデバイスのアドレスオートインクリメント機能設定
CS	: CS信号の属性定義
Protect	: Protect信号の属性定義
Hold	: Hold信号の属性定義

4. 4. 1 Type 定義

EEPROMメモリデバイスの種別を定義します。

構文 :DevDef Type *EEPROM種別*

EEPROM種別 : EEPROMメモリデバイスの種別[文字列定義型]。I2C/SPI/MicroWire のいずれかを定義します。

例 :DevDef Type SPI

EEPROM種別により、定義内容が異なります。

DevDef コマンド	I2C	SPI	MicroWire
Type	○	○	○
Data	○	○	○
Word	○	○	○
SlaveAdr	○	—	—
AdrByte	○	○	—
PageSize	○	○	○
WrTime	○	○	○
ReadCont	—	—	○
CS	—	○	○
Protect	○	○	○
Hold	—	○	—

第4章 コマンド詳細

4. 4. 2 Data 定義

EEPROMメモリデバイスのデータビット数を示します。

構文 :DevDef Data データビット数

データ信号数 : EEPROMメモリデバイスのデータ信号本数[DWORD型]
8、16 又は 32 を指定します。

例 :DevDef Data 16

4. 4. 3 Word 定義

EEPROMメモリデバイスのワード数を指定します。

構文 :DevDef Word ワード数

ワード数 : EEPROMメモリデバイスのワード数[DWORD型]
{ 16、32、64、128、256、512、1024、2048、4096、8192、16384、
32768、131072、262144、524288 } から指定します。

例 :DevDef Data 256

第4章 コマンド詳細

4. 4. 4 SlaveAdr 定義

EEPROM(I2C)メモリデバイスのヘッダデータ(デバイスコードとスレーブアドレス)を指定します。

その文字には、“01aAxX*”が使用されます。又文字数は、7Bitアドレスフォーマットの時は7文字、10ビットアドレスフォーマットのときは10文字で構成されます。

構文 :DevDef SlaveAdr *SlaveAddress*定義

*SlaveAddress*定義 : I2Cデバイスのワード数[文字列定義型]

文字には、“01aAxX*”が使用されます。又文字数は、7Bitアドレスフォーマットの時は7文字、10ビットアドレスフォーマットのときは10文字で構成されます。

- 0 : デバイスコードの“L”レベルを設定します。
- 1 : デバイスコードの“H”レベルを設定します。
- a : スレーブアドレス(アドレスピンと比較)
- A : “a”と同じ
- x : 無視
- X : xと同じ
- * : 拡張アドレス設定(アドレスの定義をヘッダ部のビットを使用して指定する場合)

例 :DevDef SlaveAdr 1010xxx

4. 4. 5 A d r B y t e 定義

EEPROMメモリデバイスのワードアドレスのバイト数を指定します。

構文 :DevDef AdrByte *アドレスバイト数*

アドレスバイト数: EEPROMメモリデバイスのワードアドレスのバイト数[BYTE型]

1、2、3 又は 4 を指定します。

例 :DevDef AdrByte 2

第4章 コマンド詳細

4. 4. 6 PageSize 定義

EEPROMメモリデバイスのページサイズを指定します。

構文 :DevDef PageSize ページサイズ

ページサイズ : EEPROMメモリデバイスのページサイズのバイト数[DWORD型]
{ 1、2、4、8、16、32、64、128、256、512、1024 } から指定します。

例 :DevDef PageSize 16

4. 4. 7 W r T i m e 定義

EEPROMメモリデバイスの書き込み時間をmsec単位で設定します。

構文 :DevDef WrTime *書き込み時間*

書き込み時間 : EEPROMメモリデバイスの書き込み時間(msec)[DWORD型]

例 :DevDef WrTime 10

第4章 コマンド詳細

4. 4. 8 ReadCont 定義

EEPROMメモリ(MicroWire)デバイスの連続読み出し可否定義。

構文 :DevDef ReadCont 有効/無効

有効/無効 : EEPROMメモリ(MicroWire)デバイスの連続読み出し有効/無効を設定します。

[文字列定義型]

ON : 連続読み出し有効

OFF : 連続読み出し無効

例 :DevDef ReadCont ON

4. 4. 9 CS 定義

Chip Select 信号の属性を定義します。

構文 :DevDef CS 有効/無効 極性

有効/無効 : CS信号の有効/無効を設定します。[文字列定義型]

ON : CS信号が有効

OFF : CS信号が無効

極性 : 有効/無効設定がONのとき、CS信号の極性を定義します。[文字列定義型]

NEG : 負論理-アクティブL

POS : 正論理-アクティブH

例: DevDef CS ON NEG H

第4章 コマンド詳細

4. 4. 10 Protect 定義

Protect 信号の属性を定義します。

構文 :DevDef Protect 有効/無効 極性

有効/無効 : Protect信号の有効/無効を設定します。[文字列定義型]

ON : Protect 信号が有効

OFF : Protect 信号が無効

極性 : 有効/無効設定がONのとき、Protect 信号の極性を定義します。

[文字列定義型]

NEG : 負論理—アクティブL

POS : 正論理—アクティブH

例: DevDef Protect ON NEG

4. 4. 11 Hold 定義

Hold 信号の属性を定義します。

構文 :DevDef Hol 有効/無効 極性

有効/無効 : Hold 信号の有効/無効を設定します。[文字列定義型]

ON : Hold 信号が有効

OFF : Hold 信号が無効

極性 : 有効/無効設定がONのとき、Hold 信号の極性を定義します。

[文字列定義型]

NEG : 負論理-アクティブL

POS : 正論理-アクティブH

例: DevDef Hold ON NEG

第4章 コマンド詳細

4.5 InstSet コマンド

InstSet コマンドは、EEPROMメモリ(SPI, MicroWire)デバイスのインストラクションセットを定義します。

InstSet *Parameter1 Parameter2*

Parameter1 : インストラクションセット種別

- Write : 書き込みインストラクションセットを指定します。**必須指定です。**
- Read : 読み出しインストラクションセットを指定します。**必須指定です。**
- Erase : データ消去インストラクションセットを指定します。**オプションです*1。**
- WriteSR : ステータスレジスタ書き込みインストラクションセットを指定します。**オプションです*1。**
- ReadSR : ステータスレジスタ読み出しインストラクションセットを指定します。**オプションです*1。**
- WriteALL : チップ全体同一データ書き込みインストラクションセットを指定します。**オプションです*1。**
- EraseALL : チップ全体データ消去インストラクションセットを指定します。**オプションです*1。**
- PageWrite : ページ書き込みインストラクションセットを指定します。**オプションです*1。**
- WriteEn : 書き込み許可インストラクションセットを指定します。**オプションです*1。**
- WriteDs : 書き込み禁止インストラクションセットを指定します。**オプションです*1。**

※インストラクションセットが未定義で指定しない場合、*Parameter2* に 全ビット“x” を指定してください。

EEPROM種別により、定義内容が異なります。

InstSet コマンド	I2C	SPI	MicroWire
Write	—	○	○
Read	—	○	○
Erase	—	—	△
WriteSR	—	△	—
ReadSR	—	△	—
WriteALL	—	—	△
EraseALL	—	—	△
PageWrite	—	—	△
WriteEn	—	△	△
WriteDs	—	△	△

○ : 必須命令、△ : オプション、— : 無効

4. 5. 1 Write 定義

Write インストラクションセットを定義します。

構文 :InstSet Write インストラクションセット定義

インストラクションセット定義 : Write シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a”と同じ

x : 無視

X : xと同じ

例: InstSet Write 00000010

※Write インストラクションセットを定義は必須指定です。

第4章 コマンド詳細

4. 5. 2 Read 定義

Read インストラクションセットを定義します。

構文 :InstSet Read インストラクションセット定義

インストラクションセット定義 : Read シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a”と同じ

x : 無視

X : xと同じ

例: InstSet Read 00000010

※Read インストラクションセットを定義は必須指定です。

4. 5. 3 Erase 定義

MicroWireデバイスの Erase インストラクションセットを定義します。

構文 :InstSet Erase インストラクションセット定義

インストラクションセット定義 : Erase シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a”と同じ

x : 無視

X : xと同じ

例: InstSet Erase 11aaaaaaa

※Eraseインストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

第4章 コマンド詳細

4. 5. 4 WriteSR 定義

SPIデバイスの Write Status Register(ステータスレジスタ書き込み)インストラクションセットを定義します。

構文 :InstSet WriteSR インストラクションセット定義

インストラクションセット定義 : WriteSR(ステータスレジスタ書き込み)シーケンスのインストラクションセットを設定します。[文字列定義型]
文字には、“01xX*” が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

x : 無視

X : x と同じ

例: InstSet WriteSR 00000001

※WriteSR(ステータスレジスタ書き込み)インストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

4. 5. 5 ReadSR 定義

SPIデバイスの Read Status Register(ステータスレジスタ読み出し)インストラクションセットを定義します。

構文 :InstSet ReadSR インストラクションセット定義

インストラクションセット定義 : ReadSR(ステータスレジスタ書き込み)シーケンスのインストラクションセットを設定します。[文字列定義型]
文字には、“01xX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

x : 無視

X : x と同じ

例: InstSet ReadSR 00000101

※ReadSR(ステータスレジスタ読み出し)インストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

第4章 コマンド詳細

4. 5. 6 WriteALL 定義

MicroWireデバイスの WriteALL インストラクションセットを定義します。

構文 :InstSet WriteALL インストラクションセット定義

インストラクションセット定義 : WriteALL シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a”と同じ

x : 無視

X : xと同じ

例: InstSet WriteALL 0001xxxxx

※WriteALLインストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

4. 5. 7 EraseALL 定義

MicroWireデバイスの EraseALL インストラクションセットを定義します。

構文 :InstSet EraseALL インストラクションセット定義

インストラクションセット定義 : EraseALL シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a”と同じ

x : 無視

X : xと同じ

例: InstSet EraseALL 0010xxxx

※WriteALLインストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

第4章 コマンド詳細

4. 5. 8 PageWR 定義

MicroWireデバイスの PageWrite インストラクションセットを定義します。

構文 :InstSet PageWR インストラクションセット定義

インストラクションセット定義 : Page Write シーケンスのインストラクションセットを設定します。

[文字列定義型]

文字には、“01aAxX*” が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

a : MicroWireデバイス定義時、アドレスビットを示します。

A : “a” と同じ

x : 無視

X : x と同じ

例: InstSet PageWR 11aaaaaa

※PageWRiインストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

4. 5. 9 WriteEn 定義

Write Enable(書き込み許可設定)インストラクションセットを定義します。

構文 :InstSet WriteEn インストラクションセット定義

インストラクションセット定義 : WriteEnable(書き込み許可設定)シーケンスのインストラクションセットを設定します。[文字列定義型]
文字には、“01xX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。
1 : デバイスコードの“H”レベルを設定します。
x : 無視
X : x と同じ

例: InstSet WriteEn 00000110

※WriteEn(書き込み許可設定)インストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

第4章 コマンド詳細

4. 5. 10 WriteDs 定義

Write Disable(書き込み禁止設定)インストラクションセットを定義します。

構文 :InstSet WriteDs インストラクションセット定義

インストラクションセット定義 : WriteDisable(書き込み禁止設定)シーケンスのインストラクションセットを設定します。[文字列定義型]
文字には、“01xX*”が使用されます。

0 : デバイスコードの“L”レベルを設定します。

1 : デバイスコードの“H”レベルを設定します。

x : 無視

X : x と同じ

例: InstSet WriteDs 00000110

※WriteDs(書き込み禁止設定)インストラクション未使用、又は未定義のときは、全ビット“x”を指定します。

デバッグソリューションズがここで提供する情報は、正確かつ信頼できるものと考えておりますが、その使用に関する責務は一切負いません。ここに記載される情報は、2005年3月におけるものです。訂正、変更、改版に追従していない場合があります。最終的な確認はヘルプデスクにお問い合わせ下さい。

デバッグソリューションズ
Debug Solutions

Web <http://www.debsol.com>
E-Mail mail@debsol.com