

**EPWriter20**

**デバイス定義ファイルの作成ガイド**

**EPU-20R3 対応版**

2010/6/10

SolitonWave Co.,Ltd

## 内容

デバイス定義ファイルの構成.....	3
索引ファイル.....	4
デバイス定義ファイルの実際.....	6
I2C デバイス定義ファイル.....	9
マイクロワイヤデバイスの定義ファイル.....	11
SPI デバイスの定義ファイル.....	14

## デバイス定義ファイルの構成

EPWriter20 では、シリアル EEPROM のカテゴリを、マイクロワイヤ、I2C、SPI の 3 つのカテゴリに分けて管理しています。デバイス定義ファイルは、通常、それぞれのカテゴリ別のフォルダに分けて管理されています。それぞれのフォルダには、メーカー別のデバイス定義ファイルと、そのフォルダにあるデバイス定義ファイルの索引となるファイルがあります。

デバイス定義ファイルの索引ファイルは、EPWriter20 のインストールフォルダにある、EPWriter20.ini に記述されています。次の例は、EPWriter20.ini のデバイス定義ファイルの索引ファイルに関する記述例です。

### [EPU-10\_INI]

**I2C=EPUI2C¥EPUI2C.ini**

**MCW=EPUMCW¥EPUMCW.ini**

**SPI=EPUSPI¥EPUSPI.ini**

### [EPU-20\_INI]

**I2C=EPUI2C¥EPUI2C.ini**

**MCW=EPUMCW¥EPUMCW.ini**

**SPI=EPUSPI¥EPUSPI.ini**

索引ファイルは、EPU-10 用と EPU-20 用に分けて指定できるようになっています。

セクション名は、“EPU-10\_INI”が EPU-10 用の定義、“EPU-20\_INI”が、EPU-20 用の定義ファイルとなります。実際には、どちらのハードウェアでもデバイス定義ファイルは共通で使用できるようになっていますので、同じものが指定されています。

索引ファイルは、I2C デバイス、マイクロワイヤデバイス、SPI デバイスのキーの名前が、それぞれ“I2C”,”MCW”,”SPI”となり、それぞれのキーの値が、対応する索引ファイル名になります。索引ファイルは、EPWriter20 のインストールフォルダからの相対パスを含みますので、“I2C=EPUI2C¥EPUI2C.ini”と記述されていれば、EPUI2C.ini は、EPWriter20 のインストールフォルダにある、“EPUI2C”という名前のフォルダの中にあることとなります。

## 索引ファイル

EPWriter20 のデバイスの定義ファイルは、カテゴリ別、メーカー別にファイルを作成し、任意のフォルダに置くことができます。それぞれのカテゴリに属するファイルにどのようなものがあるのかを示すのが、索引ファイルとなります。

索引ファイルは、次のような簡単な構成となっています。

**[IniFiles]**

**Standerd=standerd.ini**

**AKM=AKM.ini**

**ATMEL=ATMEL.ini**

**Catalyst=Catalyst.ini**

**HOLTEK=HOLTEK.ini**

**ISSI=ISSI.ini**

**Microchip=Microchip.ini**

**Philips=Philips.ini**

**Renesas=Renesas.ini**

**ROHM=ROHM.ini**

**Saifun=saifun.ini**

**Samsung=Samsung.ini**

**SEIKO=SEIKO.ini**

**STMicro=STMicro.ini**

**TurboIC=TurboIC.ini**

セクション名は、”IniFiles”となります。また、キーの名前は、EPWriter20 に表示される、メーカー名となります。キーの値は、キーに対応するメーカーのデバイス定義を記述したファイル名となります。

索引ファイルで指定されるデバイス定義ファイルは、パス名を含みませんので、索引ファイルと同じフォルダに存在しなければなりません。

通常、デバイス定義ファイルは、メーカー別に作成しますが、こればあくまで便宜上の物で、必ずメーカー別にしなければならないというわけではありません。実際に、”Standard”は、メーカー名ではなく、標準デバイスとしての定義となります。標準デバイスでは、デバイス毎の最適化がされていないので、アクセスが遅い場合もありますが、標準以外の機能は使用しないため、ほとんどのデバイスで、共通に使用することができるパラメータとなっています。

通常、新しいデバイスを追加する場合は、既存のメーカーファイルに記述を追加するだけでよいので、索引ファイルや、EPWriter20.ini の記述を変更する必要はありません。

既存のファイルに該当しないメーカーのデバイスを追加した場合は、同じフォルダに新しいデバイスファイルを置き、索引ファイルに、新しいメーカーのファイルの記述を追加します。たとえば、“NewDevice”というメーカーの定義ファイルを作成した場合は、次の行を”IniFiles”のセクションに追加します。

**NewDevice=NewDevice.ini**

ファイル名は、任意ですが、一般には、メーカー名をそのままファイル名に使用します。

## デバイス定義ファイルの実際

デバイス定義ファイルには、デバイス毎のパラメータを記述します。

デバイス定義ファイルのセクション名は、デバイス名となります。デバイス名は、1つのデバイス定義ファイル内ではユニークでなければならず、同じセクション名は存在できません。デバイス定義ファイルが異なれば、同じセクション名があっても問題ありませんので、A社とB社で、“24C256”という同じ型番が使用されていても、問題なく記述することができます。

デバイス定義ファイルのパラメータの記述は、必要最小限の記述をすればよいようになっており、すべてのパラメータを記述しなければならないわけではありません。

デバイス定義ファイルの記述方法は、I2C、マイクロワイヤ、SPIそれぞれのデバイスタイプにより、若干異なります。それぞれのデバイス毎の記述方法については後述しますが、ここでは、共通で使用されるパラメータについて、簡単に説明します。パラメータの記述方法の詳細については、それぞれのデバイスのパラメータ記述方法の説明を参照してください。

次の例は、I2C デバイスの記述例です。

**[24C00]**

**MakerName=Standerd**

**RomSize=16**

**PageSize=1**

**ReadSize=0**

**DevMode=1**

**WriteTime=20**

**AdrBitLen=8**

**DeviceType=1**

**Vcc1=500**

**Vcc2=500**

**[24C01]**

**Default=24C00**

**RomSize=128**

この例では、24C00 と 24C01 の 2 つのデバイスを定義しています。ここで使用されているセクション名は、次のようになります。

- **MakerName**  
デバイスのメーカー名を指定します。
- **RomSize**  
対象デバイスのサイズをバイト単位で記述します。バージョン 3.00 以降では、サイズはキロ、メガの単位でも記述できるようになりました。たとえば、次のような記述が可能です。  
RomSize=1024 ;1024 バイト  
RomSize=1K; 1024 バイト  
  
RomSize=1024K ;1024K バイト=1M バイト  
ROmSize=1M ;1M バイト
- **PageSize**  
同時に書き込めるサイズを指定します。サイズの単位は、バイトまたはワードです。ワードデバイスの場合、1ワード毎に書き込む必要がある場合、1を指定することになります。
- **ReadSize**  
同時に読み出せるサイズを指定します。0を指定した場合は、無制限に全てのデータを連続して読み出せることを示します。
- **DevMode**  
対象デバイスのアクセス単位を指定します。1はバイト単位、2はワード単位で、0を指定した場合は、両用デバイスとなります。
- **WriteTime**  
書き込み時間を ms 単位で指定します。書き込み時間が 10ms のデバイスの場合は、10を指定します。
- **AdrBitLen**  
アドレスのビット長をビット単位で指定します。7ビットのアドレスの場合は、7を指定します。このパラメータはバイト単位ではありませんので、注意が必要です。
- **DeviceType**  
デバイスタイプを指定します。現在、次の4つのデバイスタイプが用意されています。  
0:マイクロワイヤデバイス  
1:I2C デバイス  
2:SPI デバイス  
3:SCI(AKM)デバイス  
4:ROHM(9xxx)タイプのデバイス
- **Vcc1 と Vcc2**  
デバイスの電圧を記述します。現在は Vcc1 のみ使用しています。電圧は、10mV 単位

で記述します。(Vを100倍した値) 代表的な値は、次のようになります。

180 :1.8V

250 :2.5V

330 :3.3V

500 :5.0V

実際に使用される電圧は、使用するハードウェアに依存しますので、書き込みハードウェアの電圧設定に無い電圧が指定された場合は、指定電圧と異なる電圧が使用されますので、注意してください。たとえば、455 と記述して、4.55V を指定しても、ハードウェアが 5V しか対応していない場合は、5V が使用されます。

- **Default**

**Default** で指定したデバイスのパラメータをコピーします。24C00 と 24C01 は、ROM のサイズが異なるだけで、そのほかのパラメータはすべて同じですので、次のように、**Default** パラメータを使って、パラメータの記述を省略することができます。

**[24C01]**

**Default=24C00**

**RomSize=128**

**Default** のパラメータは、ネスティングさせることが可能ですが、ネスティングが深すぎたり、相互参照をすとアプリケーションにエラーが発生する可能性がありますので、注意してください。

※ネスティングは、現在の仕様では 10 レベルまで可能です。



## I2C デバイス定義ファイル

次の表は、I2C デバイス定義ファイルで使用できるパラメータです。

### 1. デバイス共通パラメータ

キー	機能	備考
<b>Default</b>	設定の元になるデバイス名	
<b>MakerName</b>	メーカー名	
<b>RomSize</b>	ROM サイズ (バイト)	K,M が使用可能(Ver.3.00)
<b>DeviceType</b>	デバイスのタイプ番号 0:マイクロワイヤ 1:I2C 2:SPI 3:SCI(AKM) 4.ROHM(9xxx)	
<b>PageSize</b>	連続書き込み可能なサイズ	DEF=1
<b>ReadSize</b>	連続読み込み可能なサイズ 0は無制限	DEF=1
<b>DevMode</b>	デバイスのアクセスタイプ 0:両用デバイス 1:バイトモード 2:ワードモード	DEF=1
<b>WriteTime</b>	書き込み時間(ms)	DEF=20ms
<b>AdrBitLen</b>	アドレスのビット長 (ビット)	DEF=8
<b>Vcc1</b>	デバイスの電圧×100	DEF=500(5V)
<b>Vcc2</b>	現在未使用	Vcc1 と同じ値を設定
<b>MaxClock</b>	最高周波数(KHz)	DEF=400
<b>FirmVersion</b>	使用可能なファームウェアバージョン	

### 2. I2C デバイス専用パラメータ

キー	機能	備考
<b>AdrMask</b>	アドレスのマスク 無効ビットを 0 にする	DEF=-1 現在未使用
<b>ExtDevType</b>	拡張デバイスタイプ 0:通常デバイス 1:VCLK タイプのデバイス	DEF=0
<b>A0Bit</b>	デバイスアドレスの A0 ビットの位置に出力 するアドレスのビット番号	DEF=-1 (未使用)
<b>A1Bit</b>	デバイスアドレスの A1 ビットの位置に出力 するアドレスのビット番号	DEF=-1 (未使用)
<b>A2Bit</b>	デバイスアドレスの A2 ビットの位置に出力 するアドレスのビット番号	DEF=-1 (未使用)

#### ■設定例 (24C04)

I2C デバイスは、バス上に最大 8 個のデバイスを接続し、デバイスアドレスの 3 つのビットで、特定のデバイスを選択できるようになっていますが、一部のデバイスでは、内部に

疑似的に複数のデバイスを持ち、デバイスアドレスで、内部のデバイスの何れかを選択するような仕様になっています。たとえば、24C04 は 512 バイトのメモリですが、アドレスビットは、8 ビットしかありません。24C04 は、内部的には、256 バイトのメモリセルを内部に 2 つ持つような構造で、デバイスアドレスの A0 ビットで、2 つのセルを切り替えるようになっています。従って、24C04 は、次のような記述になります。

#### [24C04]

**MakerName=Standard**

**RomSize=512** ;ROM のサイズ  
**PageSize=1** ;1 バイトづつ書き込む  
**ReadSize=0** ;連続読み出しは、無制限に可能  
**DeviceType=1** ; I2C デバイス  
**DevMode=1** ;バイトモード (通常 I2C デバイスは常にバイトモード)  
**WriteTime=20** ;書き込み時間  
**AdrBitLen=8** ;アドレスビット長  
**Vcc1=500** ;電源電圧=5V  
**Vcc2=500** ;電源電圧=5V  
**A0Bit=8** ;デバイスアドレスの A0 ビットに、アドレスのビット 8 を設定

## マイクロワイヤデバイスの定義ファイル

次の表は、マイクロワイヤデバイス定義ファイルで使用できるパラメータです。

### 1. デバイス共通パラメータ

キー	機能	備考
<b>Default</b>	設定の元になるデバイス名	
<b>MakerName</b>	メーカー名	
<b>RomSize</b>	ROM サイズ (バイト)	K,M が使用可能(Ver.3.00)
<b>DeviceType</b>	デバイスのタイプ番号 0:マイクロワイヤ 1:I2C 2:SPI 3:SCI(AKM) 4.ROHM(9xxx)	
<b>PageSize</b>	連続書き込み可能なサイズ	DEF=1
<b>ReadSize</b>	連続読み込み可能なサイズ 0は無制限	DEF=1
<b>DevMode</b>	デバイスのアクセスタイプ 0:両用デバイス 1:バイトモード 2:ワードモード	DEF=1
<b>WriteTime</b>	書き込み時間(ms)	DEF=20ms
<b>AdrBitLen</b>	アドレスのビット長 (ビット)	DEF=8
<b>Vcc1</b>	デバイスの電圧×100	DEF=500(5V)
<b>Vcc2</b>	現在未使用	Vcc1 と同じ値を設定
<b>MaxClock</b>	最高周波数(KHz)	DEF=400
<b>FirmVersion</b>	使用可能なファームウェアバージョン	

### 2. マイクロワイヤデバイス専用パラメータ

キー	機能	備考
<b>ExtDevType</b>	拡張デバイスタイプ Bit0:SecurityType Bit1:ChipErase 可 Bit2:WriteAll 可 ビットの OR を設定。チップイレーズと WriteAll 可能の場合 6 となる	DEF=0
<b>SecurityType</b>	セキュリティタイプ 0:通常デバイスタイプ 1:標準セキュリティタイプ 2: ProgramEnable ピンを持つタイプ(セキュリティなし)	DEF=0
<b>Endian</b>	エンディアンの設定 Little :リトルエンディアン Big :ビッグエンディアン Special:特殊デバイス	DEF=Big 一部のリトルエンディアンのデバイスで設定。 リトルエンディアンの場合、DLL でワードモードの書き込み時と読み出し時に、バイトスワップを行う。

#### ■エンディアンについて

マイクロワイヤデバイスには、デバイスによって、メモリの配置が異なります。多くのマイクロワイヤデバイスは、ビッグエンディアンとなっています。この場合、バイトモードで書き込んだデータは、ワードモードでは、次のように読み出せます。(バイトモードでアドレス 0 と 1 に、それぞれ 55h と aah を書き込んだ場合です。

#### ビッグエンディアンのデータ配置

バイトアドレス	0	1
バイトデータ	55H	AAH
ワードアドレス	0	
ワードデータ	55AAH	

ビッグエンディアンは、ワードデータの上位バイトが、バイトデータの低位アドレス、データがバイトデータの上位アドレスになるように配置されます。

ビッグエンディアンの場合は、バイトアクセスでもワードアクセスでも、必ず上位バイト、低位バイトの順でアクセスされます。

これに対して、リトルエンディアンのデバイスの場合は、データ配置は次のようになります。

#### リトルエンディアンのデータ配置

バイトアドレス	0	1
バイトデータ	55H	AAH
ワードアドレス	0	
ワードデータ	AA55H	

リトルエンディアンの場合は、メモリの低位アドレスに、ワードデータの低位バイトが配置され、上位アドレスに、上位バイトが配置されます。ワードモードでデータを読み出す場合、上位バイト、低位バイトの順で読みだされますが、バイトモードでは、低位バイト、上位バイトの順となります。INI ファイルで `Endian=Little` とすると、DLL 内部でこの変換を行い、DLL から読みだされるデータは、ビッグエンディアンと同じ、55AAH となります。INI ファイルで、この指定を行わないと、バイトモードとワードモードで、データが異なってしまいますので、注意が必要です。また、書き込みを行う際も、DLL 内部で返還を行いますので、アプリケーションから書き込むデータは、ビッグエンディアンと同様、55AAH となります。

マイクロワイヤデバイスには、上記の 2 種類のほか、メモリ空間を上位側と下位側に分け、ワードデータの低位バイトが、メモリ空間の下位側、上位バイトがメモリ空間の上位側に格納されるデバイスがあります。

**Endian=Special**

とした場合、この特殊なタイプのデバイスの指定となります。93C46 タイプの特殊デバイスの場合、データ配置は次のようになります。

#### 9 3 C46 特殊デバイスのデータ配置

バイトアドレス	0	40H
バイトデータ	55H	AAH
ワードアドレス	0	
ワードデータ	AA55H	

ワードモードのアドレス：00H～3FH

バイトモードのアドレス：00H～7FH

特殊デバイスの場合は、バイトモードでのアドレスの最上位ビットが、ワードモードでの、上位バイトと下位バイトの選択になります。このデバイスの場合、00H～3FH のアドレスでは、バイトモードの値と、ワードモードの下位バイトの値が一致します。また、40H～7FH のアドレス範囲では、ワードモードの上位バイトにアクセスされます。

特殊デバイスの場合、ビッグエンディアンの場合と同様に、DLL 内部でのデータ変換は行われません。このデバイスの場合は、バイトアクセスの場合と、ワードアドレスの場合で、データが異なりますので、必ず使用するモードでデータを読み書きする必要があります。

#### ■設定例 (Microchip 93AA46A)

[93AA46A]

MakerName=Microchip ;メーカー名  
RomSize=128 ;ROM サイズ  
PageSize=1 ;同時書き込みサイズ  
ReadSize=0 ;連続読み出し無制限  
DevMode=1 ;MCW デバイス  
WriteTime=6 ;書き込み時間=6ms  
AdrBitLen=6 ;アドレスビット長  
ExtType=6 ;チップイレース、Write ALL 可能  
Vcc1=500 ;5V デバイス  
Vcc2=500  
Endian=Big

## SPI デバイスの定義ファイル

次の表は、SPI デバイス定義ファイルで使用できるパラメータです。

### 1. デバイス共通パラメータ

キー	機能	備考
<b>Default</b>	設定の元になるデバイス名	
<b>MakerName</b>	メーカー名	
<b>RomSize</b>	ROM サイズ (バイト)	K,M が使用可能(Ver.3.00)
<b>DeviceType</b>	デバイスのタイプ番号 0:マイクロワイヤ 1:I2C 2:SPI 3:SCI(AKM) 4.ROHM(9xxx)	
<b>PageSize</b>	連続書き込み可能なサイズ	DEF=1
<b>ReadSize</b>	連続読み込み可能なサイズ 0は無制限	DEF=1
<b>DevMode</b>	デバイスのアクセスタイプ 0:両用デバイス 1:バイトモード 2:ワードモード	DEF=1
<b>WriteTime</b>	書き込み時間(ms)	DEF=20ms
<b>AdrBitLen</b>	アドレスのビット長 (ビット)	DEF=8
<b>Vcc1</b>	デバイスの電圧×100	DEF=500(5V)
<b>Vcc2</b>	現在未使用	Vcc1 と同じ値を設定
<b>MaxClock</b>	最高周波数(KHz)	DEF=400
<b>FirmVersion</b>	使用可能なファームウェアバージョン	

### 2. マイクロワイヤデバイス専用パラメータ

キー	機能	備考
<b>A0Bit</b>	コマンドビットに埋め込むA0のビット番号	DEF=-1 (未使用)
<b>A8Bit</b>	コマンドビットに埋め込むA8のビット番号	DEF=-1 (未使用)
<b>StatusBitName</b>	ステータスビット名を LSB からカンマ区切りで並べる	DEF= WIP,WEL,BP0,BP1,,,WPEN
<b>StatusBitMask</b>	ステータスビットのマスク。書き込み可能ビットを1にする	
<b>ChipErase</b>	チップイレーズコマンド:0はコマンドなし	DEF=0
<b>SectorErase</b>	セクタイレーズコマンド:0はコマンドなし	DEF=0
<b>ReadId</b>	ID 読み出しコマンド:0はコマンドなし	DEF=0
<b>IdLen</b>	ID の長さ (バイト)	DEF=0
<b>ExtDevType</b>	拡張デバイスタイプ 0:SPI,1:BR,2:AKM,3:AAI	DEF=0
<b>AAIPageSize</b>	AAI モード時の、ページサイズ	
<b>AAICommand</b>	AAI モードの書き込みコマンド	DEF=0
<b>AAIWriteTime</b>	AAI 書き込み時の書き込み時間(us)	DEF=0

#### ■ 設定例 (SST SST25VF512)

[SST25VF512]

FirmVersion=300

;ファームウェア Ver.3.00 以降でのみ使用可能

RomSize=65536	;ROM サイズ
MakerName=SST	;メーカー名
DeviceType=2	;SPI デバイス
ExtDevType=3	;AAI タイプ
PageSize=4096	;書き込み可能ページサイズ
ReadSize=0	;無制限の連続読み出し可能
DevMode=1	;バイトモード
WriteTime=0	;書き込み時間 (AAI 書き込みのため、このパラメータは無効)
AdrBitLen=24	;アドレスビット長
Vcc1=330	;電源電圧 3.3V
Vcc2=330	
MaxClock=20000	;最高クロック 20MHz
StatusBitName=BUSY,WEL,BP0,BP1,,,AAI,BPL	;ステータスビット名
StatusBitMask=0x8C	;ビット 2,3,7 が書き込み可能
ChipErase=0x60	;チップイレースコマンド
SectorErase=0x20	;セクタイレースコマンド
ReadId=0x90	;リード ID コマンド
IdLen=4	;ID の長さ
AAIPageSize=2	;AAI 書き込み時のページサイズ=2
AAICommand=0xad	;AAI コマンド
AAIWriteTime=10	;AAI 書き込み時間 (10 $\mu$ s)