

EPU-10/20

取扱説明書



Version 1.2

August 18, 2009

株式会社ソリトンウェーブ

■履歴

平成 21 年 2 月 20 日 第 1 版

平成 21 年 8 月 11 日 第 1.1 版

ISP 端子の接続図の追加

平成 21 年 8 月 18 日 第 1.2 版

ISP 端子の適合コネクタの説明追加

目次

はじめに	4
EPU-10/20 の仕様	4
各部の名称	5
デバイスの取り付け	6
ISP コネクタの利用	6
ISP 端子への接続方法	7
ISP 端子部分の内部回路	8
ISP 端子接続用コネクタについて	9
プログラムのインストールとアンインストール	9
USB ドライバのインストール	9
EPWriter20 の概要	10
EPWriter20 の基本的な使い方	11
タイプの選択	11
インタフェースの選択	11
ニックネームの選択	12
メーカーの選択	12
デバイスの選択	12
データモードの選択	13
電源電圧の選択	13
外部電源の使用	13
データの編集	14
ファイルの読み込み	14
ファイルの保存	14
デバイスデータの読み込み	15
デバイスデータの書き込み	15
デバイスのチェック	15
プログラム	15
詳細設定	16
デバイス消去	16
バッファクリア	16
マクロボタン・バー	16
マクロ設定バー	17
ツールバーのカスタマイズ	17
EPWriter20 の高度な使い方	18
プログラム設定	18
シリアル番号設定	19

EPU-10/20 取扱い説明書

詳細設定	20
1. I2C	20
2. MCW	21
3. SPI	21
ニックネームの編集	22
ギャングプログラマ	23
オプション表面実装デバイスアダプタの使い方	25
アダプタの種類と取り付け方法	25
アダプタの取り付け位置	25
付録・ファイルフォーマット	26

はじめに

このたびは、シリアル EEPROM プログラマをご購入いただきまして、ありがとうございます。EPU-10/20 は、Windows 上の専用アプリケーションを使用して、シリアル EEPROM の I2C デバイス、マイクロワイヤデバイス、SPI デバイス等の読み書きを行うことができます。また、弊社従来品からの大きな改良点としては、USB 対応によって、高速な通信、AC アダプタが不要になったことが挙げられます。さらに、ソフトウェアも従来のものより、さらに使いやすく生まれ変わりました。本マニュアルでは、新しくなったソフトウェア EPWriter20 による EPU-10/20 の使い方をご説明いたします。

EPU-10/20 の仕様

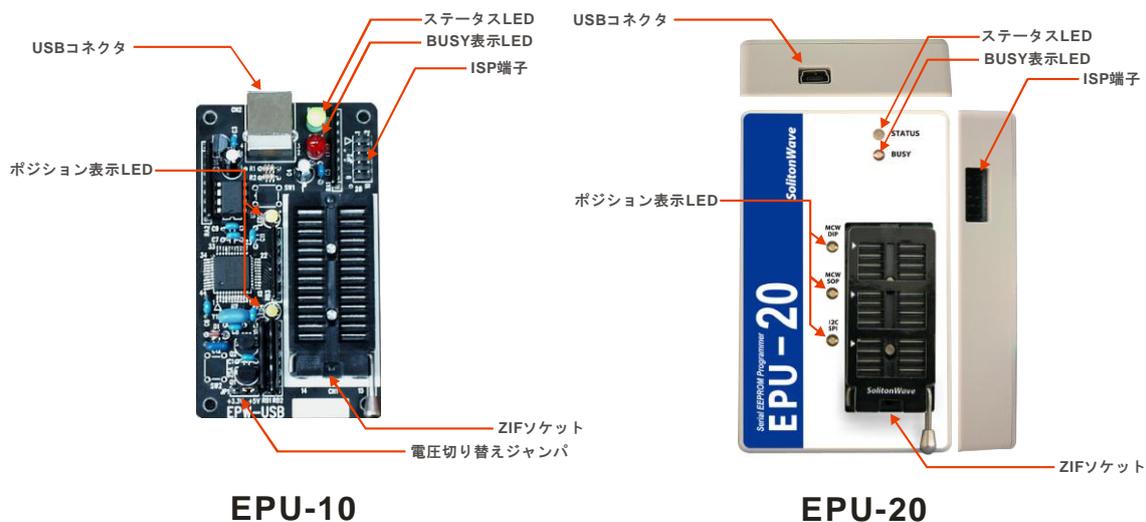
EPU-10/20 の仕様は次の通りです。

項目	内容		
対応 OS	Windows/2000/XP/VISTA (1 つ以上 USB ポートが必要)		
対応デバイス 1 マイクロワイヤデバイス	93C06(256 32x16) 93C46(1K 64x16) 93C56(2K 128x16) 93C57(2K 128x16) 93C66(4K 256x16)	93C76(8K 512x16) 93C86(16K 1024x16) 93CS06(256 32x16) 93CS46(1K 64x16) 93CS56(2K 128x16)	93CS57(2K 128x16)* 93CS66(4K 256x16) 93CS76(8K 512x16) 93CS86(16K 1024x16)
対応デバイス 2 I2C デバイス	24C00(128 bit) 24C01(1K bit) 24C02(2K bit) 24C04(4K bit) 24C08(8K bit) 24C16(16K bit)	24C164(16K bit) 24C32(32K bit) 24C64(64K bit) 24C128(128K bit) 24C256(256K bit) 24C512(512K bit)	24C1024(1024K bit) 24C21(1K bit) 24W21(1K bit) 24C515(512K bit)
対応デバイス 3 SPI/その他のデバイス	25C010(1K) 25C020(2K) 25C040(4K) 25C080(8K) 25C160(16K) 25C320(32K) 25C640(64K) 25C128(128K)	25C256(256K) BR9010(1K) BR9020(2K) BR9040(4K) BR9080(8K) BR9016(16K) M95010(1K) M95020(2K)	M95040(4K) M95080(8K) M95160(16K) M95320(32K) M95640(64K) M95128(128K) M95256(256K)
外部インタフェース	ISP書き込み用端子 PC接続用USBインタフェース		
書き込み電圧	3.3V/5V (EPU-10)、1.8V/2.5V/3.3V/5V(EPU-20)		
付属品	USBケーブル インストールディスク 操作マニュアル(EPU-10)/導入マニュアル(EPU-20)		

※ デバイス名は、デバイスのタイプを表す代表例で書かれていますので、メーカー毎の型番とは異なります。メーカーによっては、93LCXXA や、24WXX など、表と若干異なる表記をしている場合がありますが、メーカーが互換性を保証していない場合を除き、問題なく使用することができます。

各部の名称

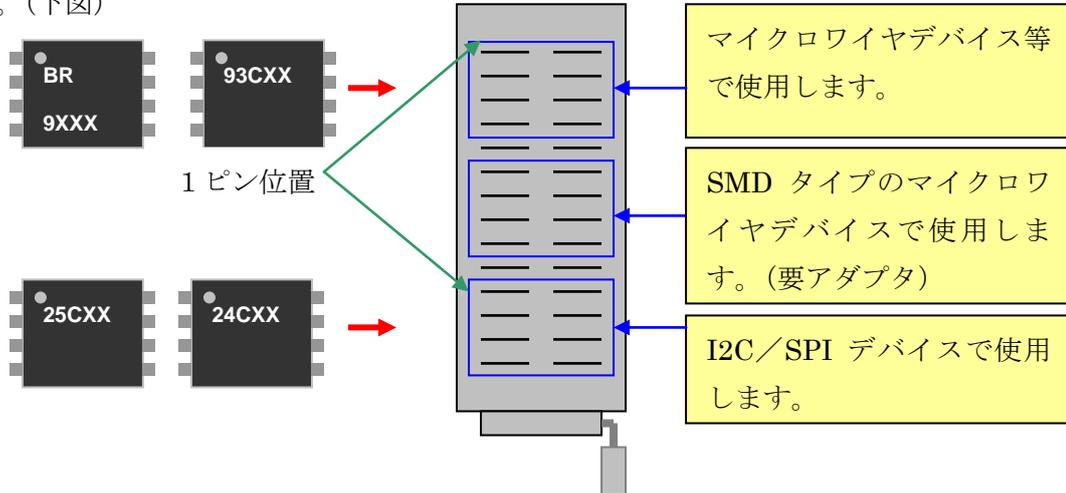
図は、EPU-10 と EPU-20 の各部の名称と機能を表しています。



- **USB コネクタ**
付属の USB ケーブルで PC と接続します
- **ステータス LED**
電源状態とエラー状態(EPU-20 のみ)を表示します。通常は緑色で、エラー発生時は、赤色となります
- **BUSY 表示 LED**
デバイスアクセス中に点灯します。EPU-10 では、エラー表示にも使用される場合があります (アプリケーションによる)
- **ポジション表示 LED**
DIP デバイスの装着位置を示します。点灯した LED の所が 1 番ピンになるように装着します
- **ZIF ソケット**
読み書きを行う DIP デバイスを装着します
- **ISP 端子**
オンボード書き込みを行う場合に使用します
- **電源切り替えジャンパ (EPU-10 のみ)**
ターゲットデバイスに使用する電圧を選択します。EPWriter20 を使用する場合は、ソフトウェアで自動的に切り替わります。

デバイスの取り付け

読み書きを行うデバイスは、ZIF ソケットにセットします。ZIF ソケットは 28 ピンです。マイクロワイヤデバイス (93CXX) や BR9000 シリーズの場合はソケットの上側、I2C デバイス (24CXX) や SPI デバイス (25CXX) の場合は、ソケットの下側の 8 ピンを使用します。(下図)



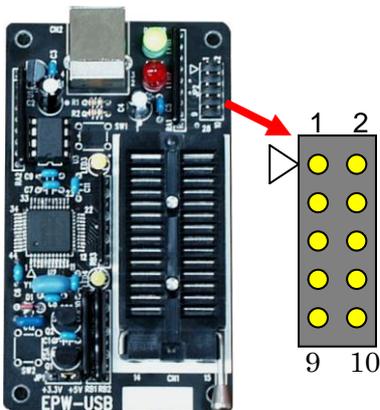
SMD タイプのマイクロワイヤデバイスでは、ピン配置が異なるものがあります。このようなデバイスを、オプションのアダプタを使って使用する場合は、上図のように、真ん中の位置にアダプタを装着します。

デバイスのセットをする場合は、必ず BUSY LED が点灯していないことを確認してください。LED 点灯中にセットするとデバイスを壊す場合がありますので、注意してください。デバイス書き込み用のソフトウェアで、使用するデバイスを選択すると、デバイスをセットする側の LED が点灯しますので、LED ガイドに合わせてデバイスをセットしてください。※面実装タイプのデバイスの場合は、オプションのアダプタが必要になります。

ISP コネクタの利用

ISP コネクタは、基板上のデバイスに書き込みを行う時に使用します。コネクタは、通常の 2.54mm ピッチのヘッダピンとなっています。ISP コネクタのピンアサインは、EPU-10 と EPU-20 とでは異なりますので注意してください。

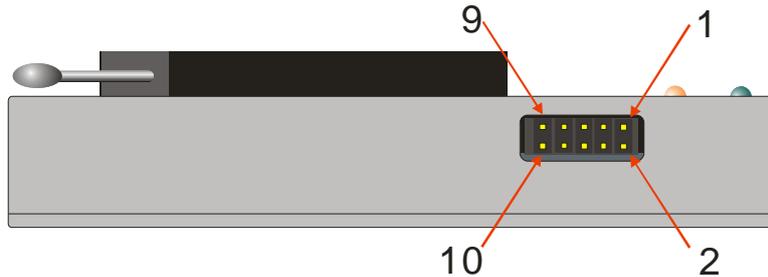
1) EPU-10 の ISP コネクタ (JP2) とピンアサイン



ピン番号	マイクロワイヤ	I2C	SPI
1	VPP	VPP	VPP
2	+5V	+5V	+5V
3	PE	WP	HOLD#
4	DI	SDA	SI
5	PRE	A2	WP#
6	-	-	-
7	CS	A0	CS#
8	SK	SCL	SCK
9	DO	A1	SO
10	GND	GND	GND

※デバイスの同名の信号に接続します。太字以外の信号の接続は任意です。

2) EPU-20 の ISP コネクタとピンアサイン



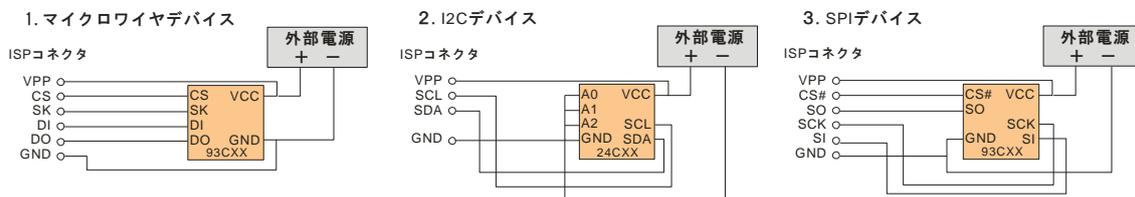
ピン番号	マイクロワイヤ	I2C	SPI
1	+5V	+5V	+5V
2	SK	SCL	SCK
3	VPP	VPP	VPP
4	DO	SDA	SO
5	GND	GND	GND
6	DI	A1	SI
7	GND	GND	GND
8	CS	A0	CS#
9	ORG	WP	HOLD#
10	PRE	A2	WP

※デバイスの同名の信号に接続します。太字以外の信号の接続は任意で

+5V には、USB からの 5V の信号がそのまま出力されます。書き込み用の VPP は、デバイスアクセス時のみ、設定された電圧が印加されます。ターゲットの基板の電源を使用する場合は、ターゲット基板の VPP を ISP コネクタの VPP ピンに接続し、EPWriter20 メイン画面で、「外部電源を使用する」チェックボックスにチェックを入れてください(EPU-10 上の電源切り替えジャンパーピンを変える必要はありません)。ISP 端子で実装済みのデバイスに対して読み書きを行う場合には、デバイスが実装された基板が ISP に対応している必要があります。特に、マイクロワイヤデバイスや SPI デバイスは、1:1 の通信となっていますので、そのまま接続すると信号が衝突して正しく動作しないだけでなく、場合によってはデバイスや基板を壊してしまう可能性もありますので、十分注意してください。

ISP 端子への接続方法

次の図は、EPU の ISP 端子と、デバイスとの接続図です。

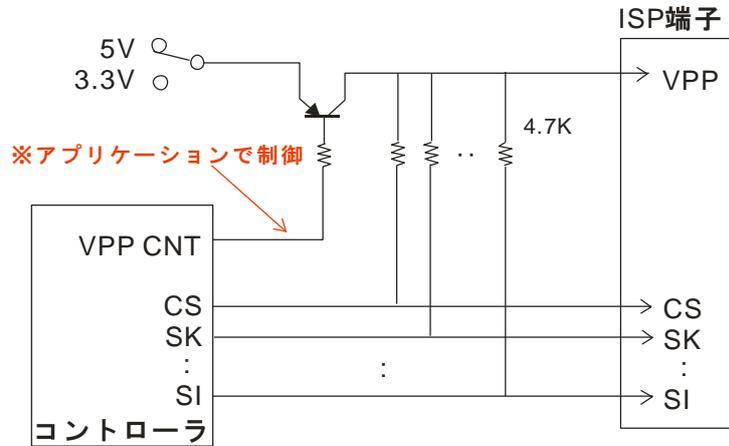


外部基板の消費電流が少ない場合は、EPU 本体から電源を供給して、デバイスの読み書きを行うことができます。EPU 本体から電源を取る場合は、消費電流が 80mA 以下になるように注意してください。外部電源を使用する場合、しない場合、どちらもデバイスの VCC は、ISP の VPP 端子に接続します。外部電源を使用するかどうかの切り替えは、書き込み用のアプリケーションで行います。

ISP 端子部分の内部回路

1) EPU-10 の ISP 端子部分の内部回路

次の図は、EPU-10 の ISP 端子の回路図です。

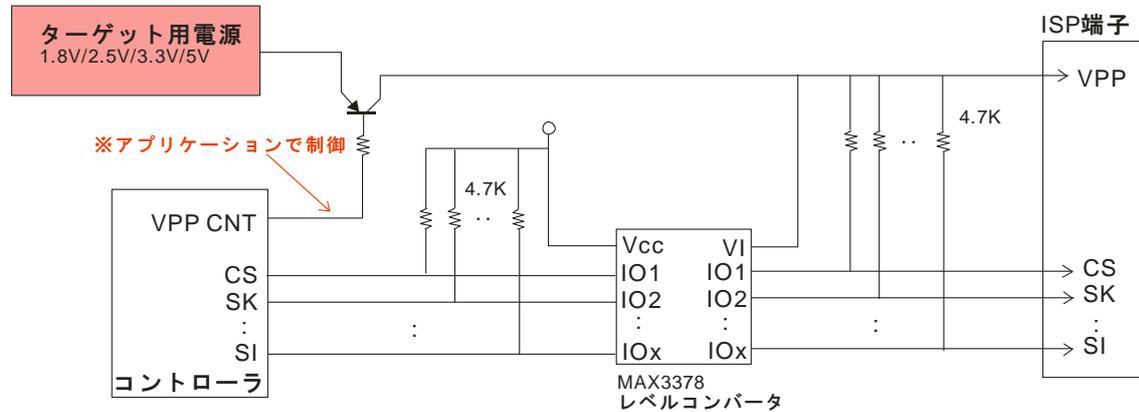


※ISP端子の信号線は、4.7Kでプルアップされています。

デバイスに接続される信号線は、すべて4.7Kでプルアップされています。外部デバイスを、EPU-10の内部電源で使用する場合は、VPP供給用のトランジスタがONとなり、選択されたVPPの電圧で、信号線がプルアップされます。また、外部電源で使用する場合は、トランジスタはOFFとなり、ISP端子から供給されるVPPにより、信号線がプルアップされます。

2) EPU-20 の ISP 端子部分の内部回路

次の図は、EPU-20 の ISP 端子の回路図です。



※ISP端子の信号線は、4.7Kでプルアップされています。

信号線の処理は、EPU-10と同様ですが、3.3V以下の電圧にも対応するため、レベルコンバータICの、MAX3378Eを使用しています。

ISP 端子接続用コネクタについて

ISP 端子に接続するコネクタは、「2550 コネクタ」「信号伝達コネクタ」あるいは「QI コネクタ」と言われる製品が使用可能です。これは、PC のマザーボードで LED などのケーブルをヘッダピンに接続するコネクタです。PC のパーツショップで販売されているものは、1 ピンや 2 ピンなど、ピン数の少ないものが多いので、必要な信号線のみ接続する方法でも利用可能です。また、国内の製品では、日本圧着端子の RF-10 というコネクタが使用可能です。こちらは、10 ピンのコネクタになります。

プログラムのインストールとアンインストール

EPWriter20 をインストールする場合は、付属ディスクの EPWriter20 フォルダにある、SETUP.EXE を起動してください。セットアッププログラムを実行すると、自動でスタートメニューに「EPWriter20」のメニューが追加されます。プログラムをアンインストールする場合は、コントロールパネルの「アプリケーションの追加と削除」から、「EPWriter20」を選択してください。

USB ドライバのインストール

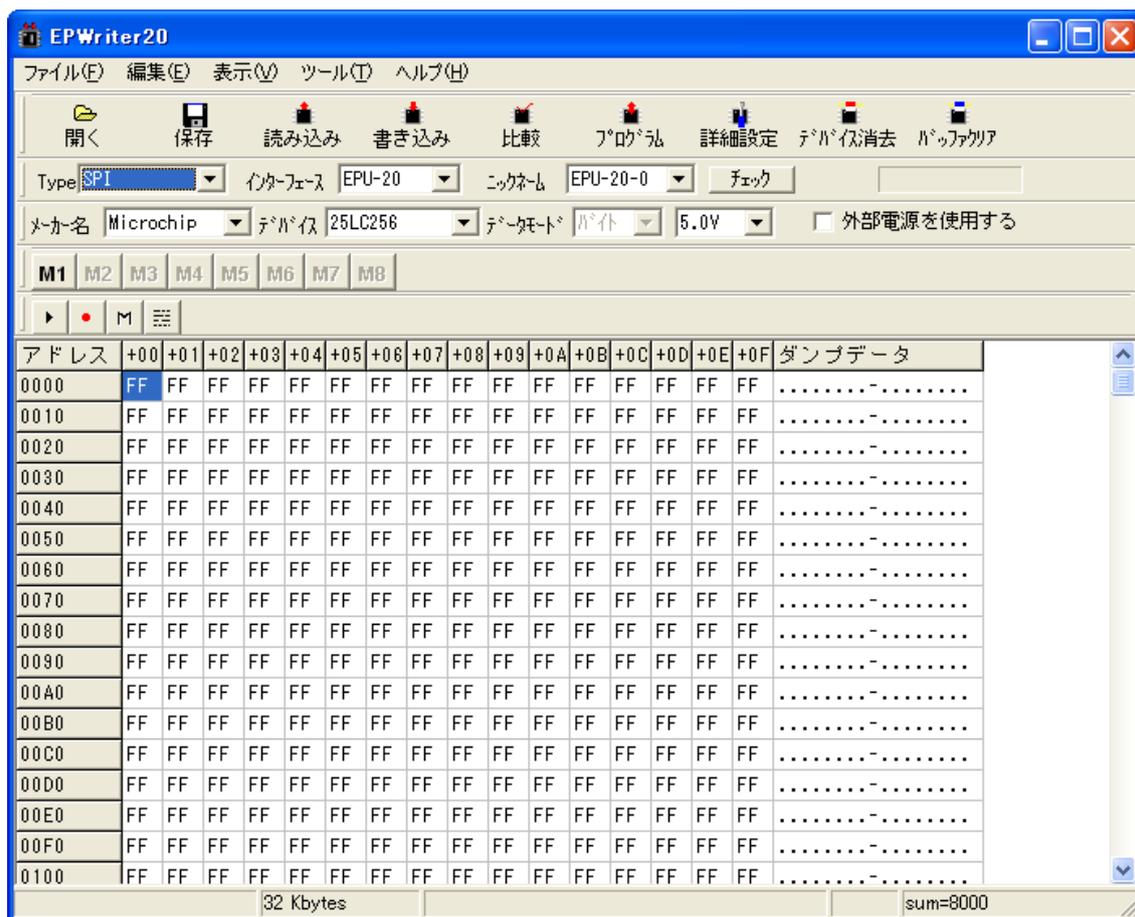
初めて EPU-10/20 を PC に接続すると、USB ドライバの要求画面が表示されますので、付属のインストールディスクを挿入し、ドライバの場所に、CD ドライブの場所を選択してください。またご使用の OS によっては、さらに承認の確認画面が出ますので、こちらも承認ボタンを押して、インストールを継続してください。

EPWriter20 の概要

EPWriter20 は、ギャングプログラマ機能や、スクリプト、電源電圧の切り替え機能を持つ、最新アプリケーションです。EPWriter20 では、PC に複数の EPU-10 または EPU-20 を接続して、同時に書き込みを行うことができます。工場などで、同一データを大量に書き込みしたい場合などに非常に便利です。また、スクリプト機能にも対応していますので、同じ処理を何度も繰り返し行いたい場合や、製品のテストなどに活用できます。また、EPWriter20 では、簡易型のアプリケーションと異なり、1つのアプリケーションで、マイクロワイヤデバイス、SPI デバイス、I2C デバイスのすべてに対応しています。そのため、デバイス選択の前にデバイスタイプやメーカー名を選択する必要がありますが、基本的な操作はファイルの読み書き、データの編集も含めて簡易型アプリケーションと同様です。

EPWriter20 の基本的な使い方

プログラムの起動には、スタートメニューの「EPWriter20」を選択してください。
 それぞれのデバイスタイプ (I2C, MCW, SPI) に対しての基本操作は、全て同じです。以下の説明では、SPI デバイス対象とした説明を中心に、EPWriter20 の基本的な使い方をご説明いたします。アプリケーションによる違いがある部分については、高度な使い方にて説明いたします。次の画面は、EPWriter20 を起動した時のメイン画面です。



タイプの選択



EEPROM にアクセスするには、まずこのコンボボックスでアクセスしたい EEPROM のタイプを選択します。I2C, MCW, SPI の3つのタイプから選択します。ご使用の EEPROM がどのタイプなのかは、それぞれのデータシートをご覧になるか、タイプを切り替えると、自動的にメーカー名、デバイス名のコンボボックスが更新されますので、そちらを参照下さい。

インターフェースの選択



このコンボボックスで、ご使用のプログラマ(EPU-10/EPU-20)に合ったインタフェースを選択します。また、EPU-10 を使用する場合は EPWriter20 の電圧切り替え機能、ニックネーム機能には対応していませんので、自動的に操作ができなくなります。

ニックネームの選択

ニックネーム EPU-20-0 ▼

PC にプログラマを接続すると、このコンボボックスにプログラマがリストアップされます。複数のプログラマを接続している場合は、ここで利用したいプログラマを選択します。

EPU-20 のプログラマに、NickNamer.exe で名前を付けると、設定した名前がここに表示されます。NickNamer.exe は、メニューからツール>ニックネームの編集を選択する事で呼び出すことができます。複数の EPU-20 を使い分ける場合は、プログラマを識別するために、名前を登録しておくとう便利です。名前を登録していない場合は、EPU-10 と同様、EPU-20-0,EPU-20-1 のように、PC に接続した順番に連番が付けられます。

”チェック” ボタンを押すと、現在選択しているプログラマの LED が点滅し、接続を確認することができます。

メーカーの選択

メーカー名 Microchip ▼

利用したいメーカーの設定をします。メーカー名がわからない場合は Standard に設定して使用してください。切り替えることにより、デバイスのコンボボックスも自動で更新されます。メーカー名はデバイスのタイプによって異なります。下記は SPI デバイスの場合のメーカー名リストです。

- AKM
- ATMEL
- Catalyst
- ISSI
- Microchip
- Renesas
- ROHM
- Saifun
- Standard
- STMicro
- TurboIC
- Others

デバイスの選択

デバイス 25LC256 ▼

EPU-10/20 取扱い説明書

編集画面の上にある、デバイス選択のコンボボックスで、使用するデバイスを選択します。デバイス名は、メーカー別のデバイスで表示されています。例えば、メーカー名が Saifun であれば、SA25F005, SA25F010, SA25F080 という風に表示されます。また、メーカー名を Standard にすると、デバイスコンボボックスにはタイプ別の総称が表示されます。例えば 93C46 というデバイス名を選択した場合は、BR93LC46A や AT93C46 のような、名前に 93C46 を含むデバイスを使用することができます。このようにデバイスにはメーカー毎に個別の番号が付けられ、アクセススピード等に細かな違いがあるものの、メモリ容量やアクセス方法には互換性があるので、93C46 といった、タイプを表す番号を使用しています。

EPU-10/20 を接続した状態でデバイスの選択を行うと、デバイスの 1 番ピンの位置を表す LED が点灯しますので、点灯位置がデバイスの 1 番ピンになるように、デバイスを装着してください。もし、EPU-10/20 を接続していない状態でデバイスの選択を行っても、その後で EPU-10/20 を接続し、チェックボタンを押せば、同様に一番ピンの位置を表す LED を点灯させることができます。

SMD タイプのマイクロワイヤデバイスや BR9XXX デバイスで、オプションアダプタを使用する場合は、デバイスによって、装着位置が異なる場合がありますので、ご注意ください。詳しくは、オプションアダプタの説明書をご参照ください。

データモードの選択

データモード

データモードコンボボックスでは、バイト(リトルエンディアン表記)モードか、ワード(ビッグエンディアン表記)モードを切り替えることができます。選択の可否はデバイスに依存します。例えば、ATMEL の MCW デバイス AT93C46D の場合は、ユーザが 1K の容量構成を 128K×8 か 64×16 で選択して使用できるようになっているので、データモードの選択が可能となります。

電源電圧の選択

このコンボボックスでは、電源電圧を切り替える事ができます。一つのデバイスに対するアクセス時だけでなく、ギャングプログラマなどで複数のデバイスにアクセスする時にもこのコンボボックスを利用することで、任意の電圧を印加する事ができます。この機能は、DLL バージョンが 2.0 以降で、外部電源を使用しない場合、かつ EPU-20 使用時のみの機能です。

外部電源の使用

外部電源を使用する

このチェックボックスにチェックを入れると、プログラマ(EPU-10/20)からデバイスに対して電源が供給されなくなります。これは、ターゲットの基板の電源を使用する場合に使用

しますので、通常のプログラマ上でのデバイス読み書き時には使用しません。ISP を利用したデバイスの読み書きに関しては、ISP コネクタの利用を参照下さい。

データの編集

アドレス	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09	+0A	+0B	+0C	+0D	+0E	+0F	ダンプデータ
0000	FF~.....															
0010	FF~.....															
0020	FF~.....															
0030	FF~.....															
0040	FF~.....															

EPWriter20 の起動画面は、バッファの編集画面となっています。EEPROM に書き込むデータは、一旦このバッファに読み込むか、この画面上でデータを編集してから、デバイスに書き込まれます。バッファの編集画面は、バイナリエディタとなっています。変更したいアドレスにカーソルを持って行き、0~9、A~F の文字で 16 進入力を行うことにより、データを変更できます。また、画面の右側は、同じ行のデータの ASCII ダンプ表示となっていますので、文字列等の確認が出来るようになっています。ただし、ASCII ダンプは表示のみで、編集はできません。また、制御コード等は、全てピリオド '.' で表示されます。16 進入力は、大量のデータ編集には向きません。この場合、他のアプリケーションデータを作成して、ファイルの読み込みでデータを読み込むことができます。ファイルからの読み込みでは、インテル HEX 形式とモトローラ S 形式、バイナリ形式に対応しています。バイナリ形式は、データをそのままファイル化した、単純な形式です。

ファイルの読み込み



ファイルの読み込みは、「開く」ボタンを押すか、ファイルメニューから「ファイルを開く」を選択し、読み込むファイルを選択してください。

読み込んだファイルのデータは、プログラムのバッファ上に格納され、画面上で編集可能になります。ファイル読み込むデータの形式は、バイナリファイル(*.BIN *.DAT)とインテル HEX ファイル(*.HEX)、さらにモトローラ S ファイル(*.MOT *.S)の 3 種類をサポートしています。

ファイルの保存



EPWriter20 は、編集したデータや、デバイスから読み込んだデータを、ファイルに保存することができます。データをファイルに保存する場合は、「保存」ボタンを押すか、「ファイル」メニューから、「ファイルの保存」を選択してください。

ファイルに保存できる形式は、バイナリファイル(*.BIN *.DAT)とインテル HEX ファイル(*.HEX)、モトローラ S ファイル(*.MOT *.S)、テキストファイル(*.TXT)の 4 種類です。

ファイルメニューから、「ファイルを保存」を選択し、続いてファイルの保存形式を選択し、ファイルに保存します。

デバイスデータの読み込み



装着したデバイスのデータを読み出すには、「読み出し」ボタンを押すか、「ツール」メニューから、「デバイスデータの読み込み」を選択してください。デバイスの全てのデータが、プログラムのバッファ上に読み込まれます。バッファに読み込んだデータは、画面上で編集したり、ファイルに保存したりすることができます。また、デバイスをコピーする場合は、この状態で、別のデバイスを装着して、書き込みを行います。

デバイスデータの書き込み



装着したデバイスにバッファのデータを書き込むには、「書き込み」ボタンを押すか、「ツール」メニューから、「デバイスの書き込み」を選択してください。バッファ上のデータを、装着したデバイスに書き込みます。

デバイスのチェック



デバイスに正しくデータが書き込まれたかどうかを調べる場合は、デバイスのチェックを行います。「ツール」メニューから、「デバイスのベリファイ」を選択するか、「比較」ボタンを押してください。バッファの内容とデバイスの内容を比較し、データに違いがあれば、エラーメッセージが表示されます。

プログラム



「プログラム」は、プログラム設定(メニューのツール>プログラム設定)で、設定したプログラムシーケンスに従ってデバイスへの書き込みを行います。「ツール」メニューから、「プログラム」を選択するか、「プログラム」ボタンを押してください。

※ 通常の書き込みでは、書き込みが正常に行われたかどうかを確認するため、必ずプログラム設定でベリファイを選択してください。もしくは、書き込み後に比較を行うようにしてください。プログラム設定に関しては後述のプログラム設定を参照下さい。

詳細設定



各デバイスのパラメータ設定や、デバイスのステータスビットを設定する事が出来ます。詳細設定をするには、「詳細設定」ボタンを押すか、「ツール」メニューの「詳細設定」を選択します。8ビットデバイスの場合は、アクセス単位は1バイトで、16ビットデバイスの場合は、アクセス単位は2バイトとなります。また、16ビットデバイスの場合、アドレスがバイトアドレスを示しているため、奇数のアドレスは設定できませんので、ご注意ください。また、簡易アプリケーションと異なる点として、詳細設定からの任意アドレスの読み込み、書き込みができなくなっています。その代わりに、スクリプトで行えるようになっていきますので、スクリプトをご活用下さい(スクリプトに関しては別紙「スクリプトとマクロの利用の手引」を参照下さい)。

デバイス消去



デバイス全体の消去を行います。デバイス消去コマンドの無いデバイスの場合は、全てのデータをFFにします。高速書き込みと併用すると、デバイスの書き込み時間を短縮することができます。デバイスの特定の領域にだけ、アクセスしたい場合には、スクリプトで行うことができます。スクリプトに関しては別紙「スクリプトとマクロの利用の手引」をご覧ください。

バッファクリア



編集バッファを全てFFにクリアします。編集を最初からやり直す場合などに使用します。デバイスのデータをクリアしたい場合は、バッファをクリアしてから、デバイスの書き込みを行ってください。

マクロボタン・バー



最大8個までのマクロをボタンにアサインして、ワンクリックでマクロを実行することができます。マクロボタン・バーを表示するには、「表示」メニューから「マクロボタン・バー」を選択して下さい。マクロボタンを設定するには、「ツール」メニューの「マクロボタンの設定」を選択するか、下記のマクロ設定バーからマクロ設定ボタンを押します。マクロボタンの設定や、詳しいマクロの使用方法については、別紙「スクリプトとマクロの利用の手引」を参照下さい。

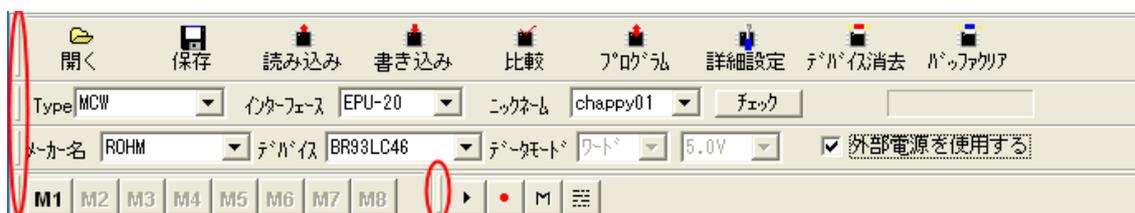
マクロ設定バー



マクロの再生/記録、マクロボタン設定ウィンドウの表示、スクリプトエディタの表示を行えます。マクロ設定バーを表示するには「表示」メニューから「マクロ設定バー」を選択して下さい。マクロやスクリプトエディタに関しては、別紙「スクリプトとマクロの利用の手引」を参照下さい。

ツールバーのカスタマイズ

「表示」メニューから、任意のツールバーを選択することで、EPWriter20 メイン画面の表示を自由にカスタマイズする事ができます。

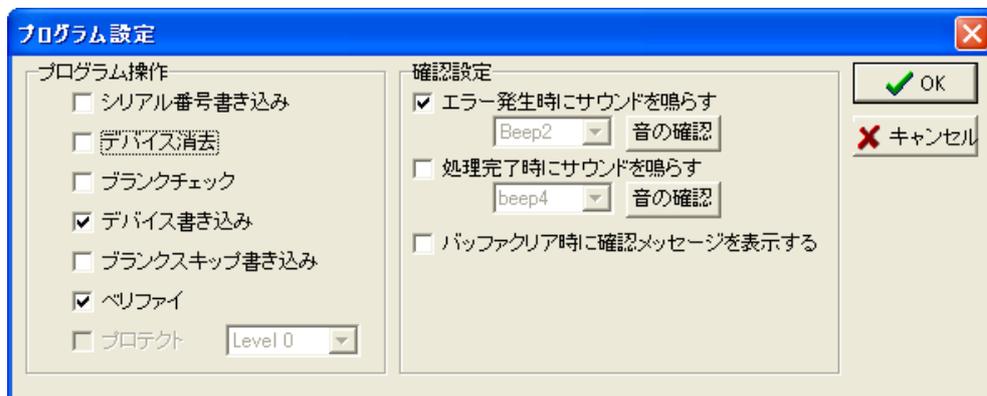


上記は全ての表示バーを表示させた画面です。各バーの左端にある小さなタブをドラッグ & ドロップすることで任意の位置にバーを持っていくことができます。これにより、自由な配置が可能となります。マクロボタン・バーはマクロボタンを設定していない場合は表示されませんので、ご注意ください。

EPWriter20 の高度な使い方

ここでは、EPWriter20 の高度な使い方を説明いたします。内容としては、詳細なプログラムのシーケンス設定や、詳細設定、ギャングプログラマ、スクリプトエディタ、ISP コネクタの利用法などを説明いたします。

プログラム設定



プログラム設定画面を表示するには、「ツールメニュー」から「プログラム設定」を選択します。この画面では、プログラムシーケンスの設定(プログラム操作)と、確認設定を行うことができます。プログラムシーケンスとは、プログラムボタンを押した時、もしくは、「ツールメニュー」から、「プログラム」を押した時のプログラムの流れを指します。この機能を使えば、よく利用するプログラムの流れをワンクリックで行うことができます。プログラムシーケンスは上から下へ進められ、チェックを外すとスキップ、チェックをつけると実行されます。上の画面の場合は、デバイス書き込みをした後、ベリファイが行われます。各操作の説明は以下の通りです。

プログラム操作	説明
シリアル番号書き込み	シリアル番号設定で設定したシリアル番号を書き込みます。
デバイス消去	デバイスを消去します。
ブランクチェック	デバイスが空の状態(全て FF)かどうかをチェックします。
デバイス書き込み	通常のデバイス書き込みです。
ブランクスキップ書き込み	ブランク(FF)をスキップして書き込みます。
ベリファイ	バッファデータとデバイスデータを比較します。

EPU-10/20 取扱い説明書

プロテクト	本バージョンでは未実装となります。
-------	-------------------

※デバイスの書き込みと、ブランクスキップの書き込みは同時に選択できません。

確認設定では、エラー発生時や、処理完了時にサウンドを鳴らすかどうかを選択することができます。サウンドは Windows のデフォルトの音源を利用しており、「音の確認」ボタンを押すことで確認することができます。また、「バッファクリア時に確認メッセージを表示する」にチェックを入れると、バッファクリアボタンを押した時に、バッファクリアをするかどうかのプロンプトが出ますので不注意で現在のバッファをクリアしてしまうのを防ぐことができます。すべての設定が済んだら、OK ボタンを押して設定を反映させます。

シリアル番号設定

シリアル番号設定ウィンドウを開くには「ツールメニュー」から、シリアル番号設定を選択してください。ここでは、デバイスに書き込むシリアル番号を設定することができます。シリアル番号は、書き込む毎に連番で増加していきます。例えば、開始番号が0であれば、次は1、次は2という風に増加します。シリアル番号は、プログラム設定の「シリアル番号書き込み」にチェックが入っていれば、プログラム時には自動的にシリアル番号がデバイスに印付されます。また、このウィンドウの「書き込み」ボタンを押せば、バッファにシリアル番号が書き込まれます。このボタンを使って実際にどのようにシリアル番号が印付されるかを試すことができます。各設定項目の説明は以下の通りです。

開始番号		シリアル番号の開始番号を指定します。
開始アドレス(HEX)		シリアル番号の書き込みを開始するアドレスを指定します。
データタイプ	バイナリ	バイナリで番号を書き込みます。
	テキスト	テキスト(ASCII)で番号を書き込みます。
データ形式	バイナリ/16進	16進数表現で番号を表示します。
	BCD/10進	10進数表現で番号を表示します。

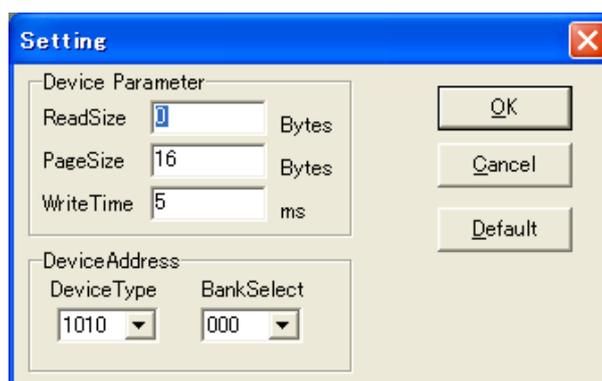
バイト数/桁数	シリアル番号を書き込むバイト数/桁数を指定します。	
書き込み方法	バイト配列	バイト単位で書き込みます。
	ワード配列	ワード単位で書き込みます。
シリアル番号書き込み前に確認する	チェックすると、書き込み前に確認画面がでます。	

詳細設定



各タイプデバイスのページサイズや書き込み時間など、デバイスごとの詳細な設定を行う事ができます。「ツール」メニューから、「詳細設定」を選択するか、「詳細設定」ボタンを押すと、タイプ毎に以下のような画面が開きます。デバイス毎にデフォルトの値が設定されていますので、デフォルトボタンを押せば、変更を加えた後でも初期状態に戻す事ができます。また、設定を反映させるには、OK ボタンを、反映させたくないときはキャンセルを押してください。各パラメータの説明は以下の通りです。

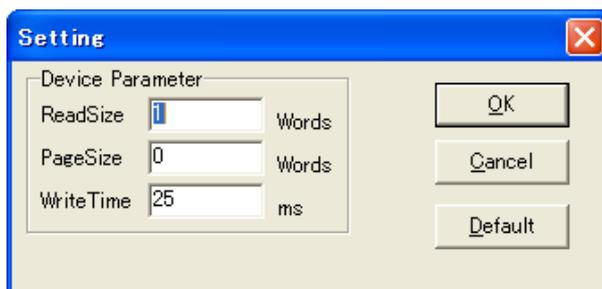
1. I2C



Device Parameter	ReadSize	1 回に読み出し可能なサイズ。(0 は無制限)
	PageSize	1 回に書き込めるサイズ
	WriteTime	1 ページの書き込みに必要な時間 (最大)
Device Address	DeviceType	デバイスの種別を表す 4 ビットのデータです。EEPROM の場合、通常 1010 です。
	BankSelect	デバイスを選択する 3 ビットの値です。

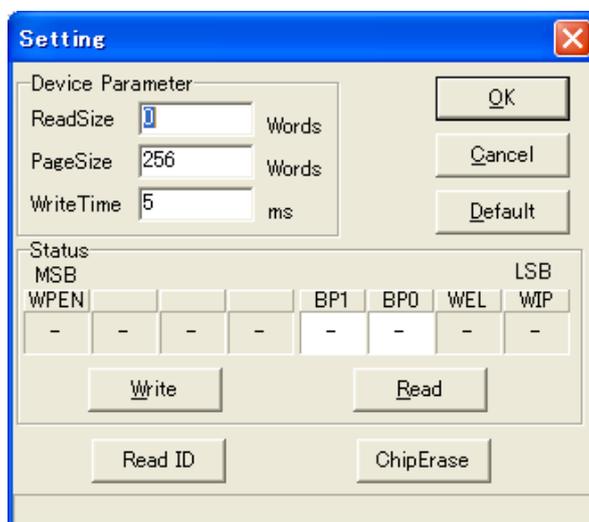
※パラメータの詳細は、データシートを参照してください。

2. MCW



Device Parameter	ReadSize	1 回に読み出し可能なサイズ。(0は無制限)
	PageSize	1 回に書き込めるサイズ
	WriteTime	1 ページの書き込みに必要な時間 (最大)

3. SPI



Device Parameter	ReadSize	1 回に読み出し可能なサイズ。(0は無制限)
	PageSize	1 回に書き込めるサイズ
	WriteTime	1 ページの書き込みに必要な時間 (最大)

- Status

ステータスビットの設定と確認を行います。ステータスを読み出す場合は、**Read** ボタンを押すと、画面にステータスビットの内容が表示されます。8 ビットのデータの、変更可能な部分 (背景が白のビット) をクリックすると、データが反転します。さらに、**Write** ボタンを押すと、変更した内容を書き込むことができます。

- Read ID

Read ID コマンドに対応しているデバイスの場合、**ID** を読み出して、表示することができます。

- Chip Erase

チップイレーズや、セクタイレーズに対応しているデバイスは、デバイスの消去が可能です。

※デバイスによって任意の設定項目があるため、設定できないパラメータは、自動的に操作できないようになっています。詳しくは各デバイスのデータシートを参照下さい。

ニックネームの編集

NickNamer.exe は EPU-20 のみの特殊アプリケーションです。接続した EPU-20 にお好きな名前をつけることができます。一台の PC で複数台の EPU-20 と接続している時（例えばギャングプログラマを使用している時）などに、個別のハードウェアを区別するのに便利です。また、ニックネームが未設定の EPU-20 に対しては、メイン画面のニックネームのリストボックスには EPU-20-0, EPU-20-1, EPU-20-2, ... という風にデフォルト名が表示されます。EPU-10 の場合はニックネームをつけられませんので、常時 EPU-10-0, EPU-10-1, ... と表示されます。



番号	詳細説明
1	該当デバイスに対してアクセスするかどうかを選択するチェックボックス。
2	デバイス番号。PC に接続した順で 1 から順に振り分けられる便宜的なものです。
3	このテキストスペースではニックネームの書き込み/表示を行います。
4	チェックボタンを押すと、デバイス側の LED が点灯し、どのデバイスに対応しているかわかります。
5	再読み込みボタンを押すと、すでに書き込まれているニックネームを再度読み込みます。
6	書き込みボタンを押すと、テキストスペースに書き込まれたニックネームをデバイスに書き込みます。

EPU-10/20 取扱い説明書

ギャングプログラマ

ギャングプログラマとは、一度に最大 10 台の EPU-10/20 を接続し、書き込みや消去など同作業を一度に行う為のもの。この機能は、工場などで、大量に EEPROM の書き込みを行う際に、大変便利です。

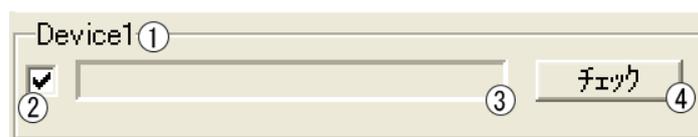


ギャングプログラム例（一度に 5 台の EPU-20 を接続）



メイン画面

ギャングプログラマを開くには、「ツールメニュー」から、「ギャングプログラマ」を選択してください。操作方法は一台で使用する場合とほとんど同じです。しかし、チェックボックスが付いていることからお察しの通り、このチェックボックスで、通信するハードウェアを選択することができます。よって、特定の複数のハードウェアに書き込みたい場合はまずチェックボックスで通信したいハードウェアを選択し、その後、デバイス操作を行ってください。各パラメータの説明は以下の通りです。



デバイス画面

番号	説明
1	デバイス名。ニックネームが付いている場合はニックネームが入ります。
2	実行するかどうかのチェックボックス。チェックを外すと、そのデバイスだけプログラムを実行しません。
3	実行ステータス画面。実行状態の表示や、実行結果が表示されます。
4	チェックボタンを押すと、デバイス側のLEDが点灯し、どのデバイスに対応しているかわかります。

デバイス操作

ボタン	説明
	デバイスを消去します。
	デバイスに書き込まれたデータとバッファデータを比較します。
	書き込みを行います。
	プログラム設定にしたがってプログラムを実行します。

設定保存



これは、EPU-20でのみ使用可能です。"保存"を押すと、現在のギャングプログラムの設定（デバイス番号とニックネームの組み合わせ）を保存することができます。"読み込み"を押すと、前回保存した設定を読み込みます。また、"起動時に設定を読み込む"にチェックを入れておくと、次にギャングプログラムを開いたときに設定を自動的に読み込んでくれます。

オプション表面実装デバイスアダプタの使い方

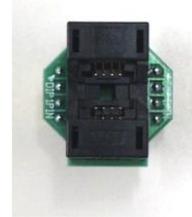
シリアル EEP プログラムの表面実装用アダプタは、パッケージのサイズにより、2 種類のアダプタがあります。必ず、ご使用になるデバイスにあったアダプタを使用してください。

(表面実装デバイスアダプタは、別売です。)

アダプタの種類と取り付け方法

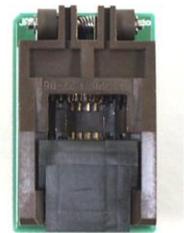
* ROM-08SP-08DP-111 アダプタ

パッケージの横幅が 3.8mm のデバイス (SN タイプ:150mil) のデバイス用です。ほとんどのデバイスがこのタイプです。アダプタの外枠を押し下げて、「SOP 1PIN」が書き込むデバイスの 1PIN に合うようにデバイスをセットします。



* ROM-08SP-08DP-117 アダプタ

パッケージの横幅が 5.2mm のデバイス (SM タイプ:208mil) のデバイス用です。マイクロチップ社の一部製品が、このタイプになっています。アダプタの蓋を外し、「SOP 1PIN」が書き込むデバイスの 1PIN に合わせてセットします。



アダプタの取り付け位置

面実装アダプタは、シリアル EEPROM プログラム USB の書き込み用ソケットにセットします。書き込み対象のデバイスにより、セットする場所が異なりますので、注意してください。また、アダプタの「DIP 1 PIN」と書かれたピンが、次の図の△マークの位置になるように、アダプタをセットしてください。

<p>The diagram shows a black plastic adapter socket with three distinct sections highlighted in color: yellow at the top, red in the middle, and green at the bottom. To the right of the socket are three vertical labels: a yellow label with 'EEP', a red label with 'OPT', and a green label with 'I2C'. Three colored arrows (yellow, red, green) point to the corresponding sections of the socket.</p>	<p>黄色い部分</p>	<p>マイクロワイヤデバイス (93CXX) や、BR9XXX デバイスで、VCC が 8 ピンのタイプはこの位置にアダプタをセットします。</p>
<p>赤い部分</p>	<p>マイクロワイヤデバイス (93CXX) や BR9XXX デバイスで、VCC が 2 ピンのタイプは、この位置にアダプタをセットします。</p>	
<p>緑の部分</p>	<p>I2C デバイス (24CXX) と SPI デバイスで、VCC が 8 ピンのタイプは、この位置にアダプタをセットします。</p>	

付録・ファイルフォーマット

シリアル EEPROM プログラムでは、インテル HEX ファイル形式、モトローラ S ファイル形式、バイナリ形式の、3つのファイルフォーマットをサポートしています。

(1) インテル HEX ファイルフォーマット

このファイル形式は、インテル社のインテル HEX フォーマットの形式です。インテル HEX ファイルは、次のような構成になっています。

コードセグメントレコード
データレコード
データレコード
:
コードセグメントレコード
データレコード
データレコード
:
ファイル終了レコード

図のように、インテル HEX ファイルフォーマットでは、1つのコードセグメントレコードに対して、複数のデータレコードを持つことが出来ます。また、コードセグメントレコードとデータレコードの組も複数もつことが出来ます。

それぞれのレコードは、次のような構造になっています。

1レコード(一行)の構造					
REC MARK	REC LEN	LOAD ADR	REC TYPE	DATA	CHECK SUM

それぞれのレコードの詳細は、以下のようになります。

1) コードセグメントレコード

フィールド	内容
REC MARK	セミコロン (:) に固定
REC LEN	“02”に固定
LOAD ADR	“0000”に固定
REC TYPE	“02”に固定 (セグメントを表す)
DATA	“X000”が入る。Xの部分は、’0’~’F’までのセグメントの値。
CHECK SUM	REC LEN から DATA フィールドまでの文字列の、全バイトの総和の2の補数

2) データレコード

EPU-10/20 取扱い説明書

フィールド	内容
REC MARK	セミコロン (:) に固定
REC LEN	“nn” nn はデータフィールドに格納されるデータのバイト数
LOAD ADR	“nnnn” nnnn は、データのロードアドレス
REC TYPE	“00”に固定 (データレコードを表す)
DATA	データが入る。データは、1 バイトを 2 文字の 16 進数で格納する。
CHECK SUM	REC LEN から DATA フィールドまでの文字列の、全バイトの総和の 2 の補数

3) ファイル終了レコード

フィールド	内容
REC MARK	セミコロン (:) に固定
REC LEN	“00”に固定
LOAD ADR	“0000”に固定
REC TYPE	“01”に固定 (ファイル終了レコードを表す)
DATA	なし
CHECK SUM	“FF”に固定

(2) モトローラ S フォーマット

このファイル形式は、モトローラ社のモトローラ S フォーマットの形式です。モトローラ S フォーマットは、次のような構成になっています。

ヘッダレコード(option)
データレコード
データレコード
:
エンドレコード

モトローラ S フォーマットには、セグメントレコードはありません。データの先頭は全て S で始まり、レコードタイプ(REC TYPE)によってヘッダ/データ/エンドレコードを判別するようになっています。ただし、ヘッダレコードはオプション (ファイル名を格納したりするのに使用する) のため、EPWriter20 中で.MOT/.S 形式で保存する時は、ヘッダレコードは省略しています。それぞれのレコードは、次のような構造になっています。

1レコード(一行)の構造				
REC TYPE	BYTE COUNT	LOAD ADR	DATA	CHECK SUM

それぞれのレコードの詳細は、以下のようになります。

1) データレコード (オプション)

フィールド	内容
REC TYPE	“S0”に固定
BYTE COUNT	“nn” nn はアドレス、データ、チェックサムバイトの合計バイト数
LOAD ADR	“0000”に固定
DATA	書き込みデータ (ファイル名など)
CHECK SUM	BYTE COUNT から DATA フィールドまでの文字列を、バイト単位で総和した結果の 1 の補数

2) データレコード

フィールド	内容
REC TYPE	“S1”, “S2”, “S3”のいずれか*
BYTE COUNT	“nn” nn はアドレス、データ、チェックサムバイトの合計バイト数
LOAD ADR	“nnnn”[16bit], “nnnnnn”[24bit], “nnnnnnnn”[32bit]のいずれかで、それぞれ n には、データのロードアドレスが入る。データの大きさによりアドレスサイズが変わる
DATA	書き込みデータ
CHECK SUM	BYTE COUNT から DATA フィールドまでの文字列を、バイト単位で総和した結果の 1 の補数

*アドレスが 16bit の場合は”S1”、24bit の場合は”S2”、32bit の場合は”S3”

3) エンドレコード

フィールド	内容
REC TYPE	“S7”, “S8”, “S9” のいずれか*
BYTE COUNT	“nn” nn はアドレス、データ、チェックサムバイトの合計バイト数
LOAD ADR	“nnnn”[16bit], “nnnnnn”[24bit], “nnnnnnnn”[32bit]のいずれかで、それぞれ n には、データのロードアドレスが入る。通常はデータアドレスに続く番地が入る
DATA	なし
CHECK SUM	BYTE ADR から DATA フィールドまでの文字列を、バイト単位で総和した結果の 1 の補数

*アドレスが 16bit の場合は”S9”、24bit の場合は”S8”、32bit の場合は”S7”

(3) バイナリフォーマット

バイナリフォーマットでは、データをそのままファイル化します。0 番地目からのデータを順番に格納したファイルとなります。ファイルサイズは、使用するメモリのサイズ以下で無ければなりません。

memo

EPU-10/20 取扱い説明書

平成 21 年 2 月 20 日	第 1 版
平成 21 年 8 月 11 日	第 1.1 版
平成 21 年 8 月 11 日	第 1.2 版

発行所 株式会社ソリトンウェーブ
〒101-0023 東京都千代田区神田松永町 17-15
大野ビル 4F
TEL:03-5256-0956 / FAX:03-5256-0956
電子メール：info@solitonwave.co.jp
ホームページ：<http://www.solitonwave.co.jp>

