



Prog-Express ユーザーマニュアル

目次

目次.....	2
システム要件.....	5
32-BIT オペレーティング・システム.....	5
64-BIT オペレーティング・システム.....	5
PROG-EXPRESS のライセンス情報	6
フリーウェア・ライセンス条件.....	6
インストール.....	7
ソフトウェア・インストール.....	7
デバイスドライバ・インストール.....	8
自動インストール	8
手動インストール	9
ソフトウェア・モード.....	12
概要.....	12
チップをプログラミングする.....	13
チップをコピーする.....	14
チップを読み込む.....	15
プロダクション・モード.....	16
データエントリー・フィールド	19
プログラマ	19
コピー元チップとコピー先チップ.....	19
コピー元ファイル	19
チップオプション	21
シリアルナンバー	21
プロセスコントロール.....	22
プロセスステップ	23
HEX エディタ	25
HEX エディタメニューとツールバー	26

メニューとツールバー「ファイル」	26
メニューとツールバー「編集」	27
メニューとツールバー「ビュー」	28
メニューとツールバー「プログラマ」	29
HEX エディタ・コンテキストメニュー	31
選択したデータで作業する	32
範囲を選択する	32
範囲を編集する	32
マウスを使ってビューを変更する	33
HEX エディタショートカットキー	34
ソフトウェアオプション	36
一般オプション	36
プロジェクト	38
ログファイルテキスト	45
チップブラウザ	46
チップ自動アイデンティファイ	47
チップオプション	48
チップの個別オプション	48
スプリットオプション	49
シリアルナンバー	50
シリアルナンバーファイル	51
シリアルナンバージェネレーター	52
プロジェクト	53
保存する	53
読み込む	53
その他	54
オペレーターモード	54
ファームウェアアップデート	54
校正する	54
PROG-EXPRESS について	54

PROG-EXPRESS ヘルプ	54
PROG-EXPRESS の遠隔操作	55
コマンドラインパラメータを使った遠隔操作	55
スクリプトファイルを使った遠隔操作	55
リモートファイルコマンド	56
ポーリング機能をオン・オフにするコマンド	56
開くコマンド	56
モードコマンド	56
ファイル選択コマンド	57
プロセスステップコマンド	57
変異チップ選択コマンド	57
自動識別コマンド	57
起動コマンド	58
ログ保存コマンド	58
デバイス保存コマンド	58
ログ消去コマンド	58
チップリスト保存コマンド	59
追加のコマンド	59
サンプルアプリケーション: 特定のデータのプログラミング	59

システム要件

本章は Prog-Express と Microsoft® .NET Framework 2.0 のシステム要件を含みます。

Prog-Express には Microsoft® .NET Framework 2.0 が必要です。これはマニュアル CD に付属しており、必要な場合は自動的にインストールされます。

32-Bit オペレーティング・システム

- 対応オペレーティング・システム: Windows 7, Windows Vista, Windows XP Service Pack 2 以降, Windows Server 2003, Windows 2000 Service Pack 3, Windows ME, Windows 98 Second Edition
- 必須条件: Windows Installer 3.0 (Windows Installer 2.0 以降を必要とする Windows 98/ME 以外)。Windows Installer 3.1 以降推奨。IE 5.01 以降: .NET Framework には Microsoft Internet Explorer 5.01 以降が必要です。
- 最低限のハード・ディスク・スペース: 300 MB (x86)

64-Bit オペレーティング・システム

- 対応オペレーティング・システム: Windows 7 - 64 Bit, Windows Vista 64 Bit, Windows XP 64-bit, Windows Server 2003 x64 Edition
- 必須条件: Windows Installer 3.0 (Windows Installer 2.0 以降を必要とする Windows 98/ME 以外)。Windows Installer 3.1 以降推奨。IE 5.01 以降: .NET Framework には Microsoft Internet Explorer 5.01 以降が必要です。
- 64-Bit 対応: 64-Bit CPU に対応するには最新の 64-Bit バージョンの Windows XP Professional あるいは Windows Server 2003 が必要です。
- 最低限のハード・ディスク・スペース: 630 MB (64-Bit)

Prog-Express のライセンス情報

フリーウェア・ライセンス条件

保証の制限

ソフトウェアや付随資料は現状のままご利用ください。多数のコンピューター構成により、徹底的な検証をしたソフトウェアでさえ誤作動の恐れがありますので、作成者はソフトウェアや付随資料の直接的あるいは間接的な使用によるあらゆる損害に対して賠償責任を負いません。いかなる状況においても、作成者が潜在的な問題を関知していた場合でも、作成者は利益損失、営業停止、情報やデータの紛失、他のソフトウェアの損傷について賠償責任を負いません。ユーザーはソフトウェアの使用に関するあらゆる結果に全責任を負います。

複製

ソフトウェアコレクションや印刷媒体内の **CD-ROM** 搭載によりプログラムを流布及びブックウェアとしての発行は、作成者の書面での明確な許可によってのみ承認されます。

§ 263a German StGB により、ソフトウェアと付属資料の操作、逆コンパイル、分解は最大 5 年の懲役刑あるいは罰金が化せられます。その場合、作成者は提訴し、損害賠償を要求します。

その他

本条件が守られない場合は、作成者はユーザーのソフトウェア・ライセンスを取り消す権利があります。

全ての商品名と商標は、その事実確認の有無に関わらず、その所有者に属するところで認識されます。

ドイツ連邦共和国の法律が適用されます。

インストール

まずは Prog-Express ソフトウェアをインストールしてください。ソフトウェアは、提供された CD または www.batronix.com のダウンロードする箇所から入手できます。

ソフトウェアのインストール後初めて USB チップ・プログラマを接続する事ができます。

ソフトウェア・インストール

本章は Prog-Express のインストールに必要な全ての情報を含んでいます。

セットアップを開始するにはセットアップファイルをダブルクリックするか、CD-ROM のオートプレイ機能をご利用ください。

画面の指示に従ってください。

本ソフトウェアには必須条件のため、Microsoft® .NET Framework 2.0 のインストールが必要な場合もあります。インターネット接続がある場合はご使用のコンピュータが必要なファイルをダウンロードしますし、それ以外ではファイルは CD でも入手できます。

.NET Version X64, x86 Or IA64

32-bit オペレーション・システムをご利用の場合、x86 バージョンをダウンロードしてください。x64 バージョンは 64-bit オペレーション・システムにのみ対応しています。IA64 は 64-Bit オペレーション・システムを運用する 64-bit Intel プロセッサにのみ必要です。

デバイスドライバ・インストール

自動インストール

Prog-Express のセットアップ中、Batronix USB プログラムのドライバは図のように自動的にインストールされます。Windows は USB を自動的にインストールし、画面上に図のような表示が出ます。

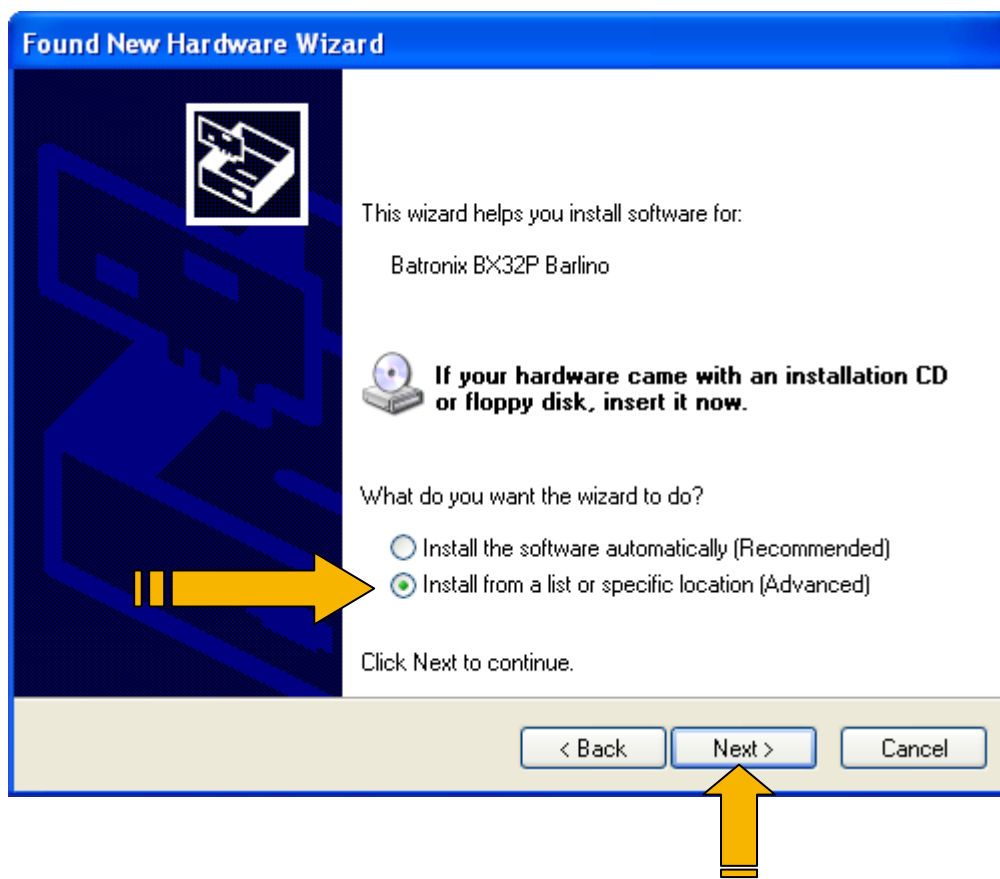
Prog-Express ソフトウェア内にプログラマが表示されない場合は、コンピュータを再起動する必要があります。



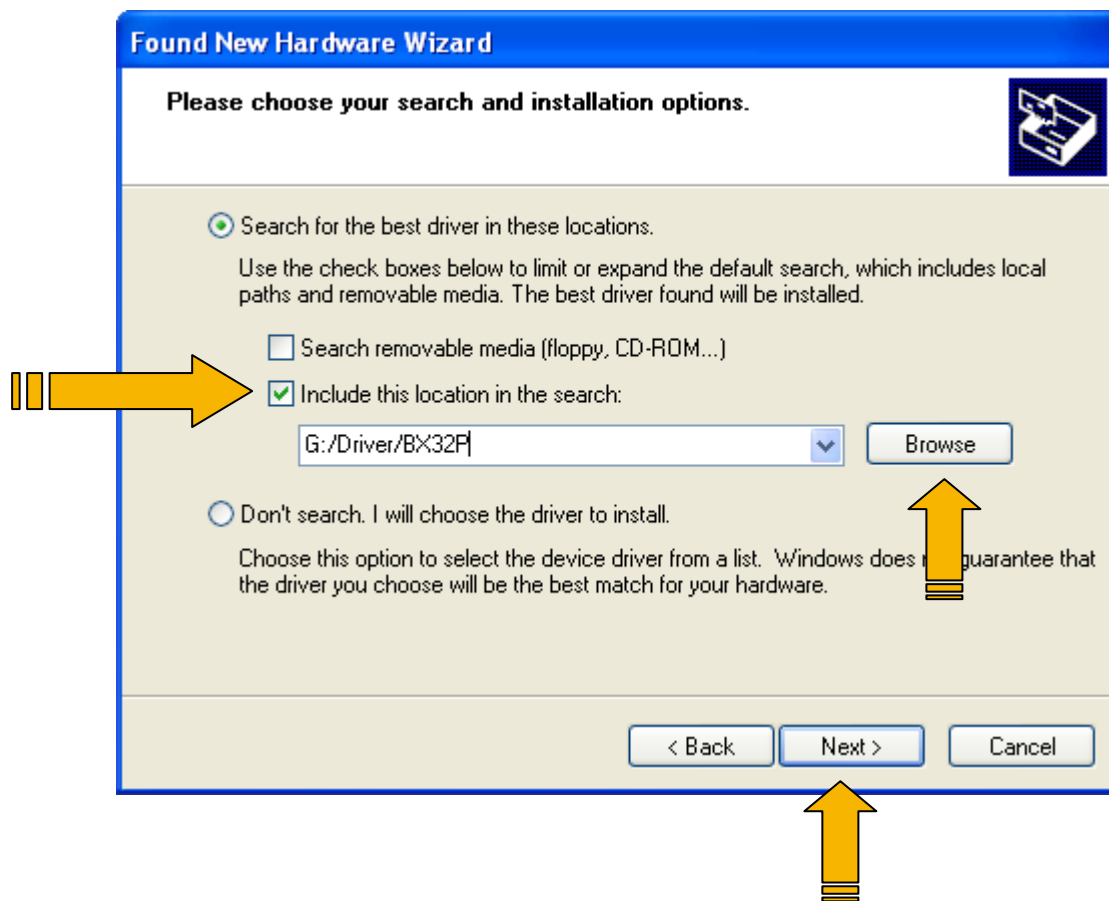
手動インストール

Prog-Express のセットアップでは全ての Batronix USB デバイス・プログラムのドライバが自動的にインストールされます。手動でインストールする必要がある場合は、下記の手順に従って下さい。

ソフトウェアのインストール後初めて、Batronix USB プログラムを接続する事ができます。Windows はすぐにデバイスを新しいデバイスとして検出し、画面に下記のようなメッセージが表示されます。:



2 番目の「リストまたは特定の場所からインストールする」を選択してください。「次へ」を選んでください。

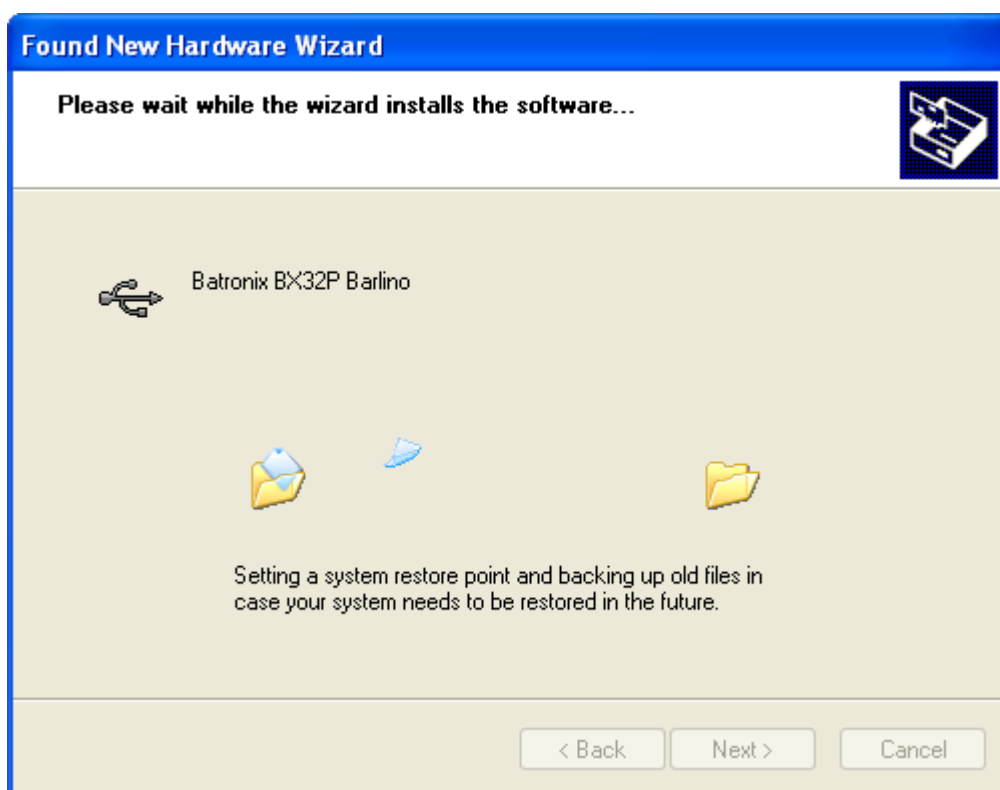


「この場所を検索に含む」を選択し、Prog-Studio インストールの「ドライバ」ディレクトリ内で、ご利用のプログラムのディレクトリ、または CD の「ドライバ」ディレクトリを表示させます。例：C:\Program Files\Batronix\Prog-Express\driver\BX32P

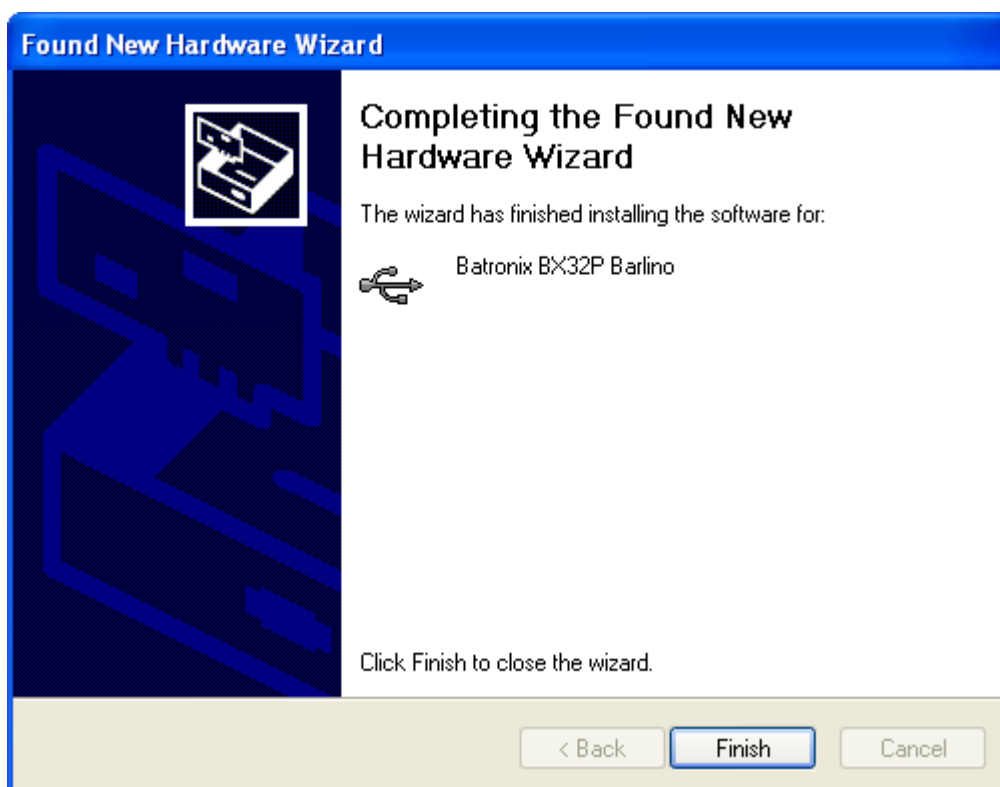
「次へ」を選んでください。



「デバイスは Windows ログテストに合格していません」のメッセージが出たら、「続行」を選択します。



ご使用の PC は 指示したソースを検索中で、プログラマを検出します。ドライバがインストールされます。



ウィザードが次のメッセージとともに終了します：「ウィザードは...へのソフトウェアのインストールを終了します」。「終了」ボタンをクリックしてください。

デバイスドライバはのインストールが完了し、デバイスを使用することができます。

ソフトウェア・モード

本章は Prog-Express の様々なソフトウェア・モードについての情報を含みます。

概要

Prog-Express には5つの基本モードと選択メニューにあるオプション・ダイアログがあります。これらのモードで、ご使用のアプリケーションに最適化されたインターフェースを選択することができます

下記のモードがあります:

- チップをプログラミングする: 1つ以上のチップをプログラミングする
- チップをコピーする: 1つ以上のチップをコピーする
- チップを読み込む: チップからデータを読み込む
- プロダクション・モード: 複数のデバイスを使ってチップをプログラミングする
- Hex-エディタをバッファする: Hex-エディタ内のデータを編集する
- ソフトウェア・オプション: ソフトウェア・オプションを編集する









チップをプログラミングする

「チップをプログラミングする」モードはハードディスクや CD から 1 つ以上のチップにプログラミングするデータに最適化されたインターフェースを提供します。



チップにデータを書込む

	BX32 Batupo II シリアルナンバー: BL2507791 ファームウェアバージョン: 02.30	リフレッシュ
	Winbond W29EE011P-90 アダプター: PLCC32-DIP32 128 Kバイト (1 MBit)	チップ自動アイデンティファイ
	ProgramData1MBit.bin パス: Y:\Files\ 最後の変更: 2011年5月5日 10:21:51	ブラウズ
	チップオプション(任意) 特別なオプションは使用されていません	オプションを変える
	シリアルナンバー(オプション) シリアルナンバーを入れないで下さい	オプションを変える

 **プログラミングプロセスをはじめる** コピー数

このインターフェースは上部のデータエントリー・フィールドと下部のプロセス・コントロール機能で構成されます。

複数のプログラミング・デバイスを接続した場合は最初のフィールドで使用するデバイスを選択してください。次に 2 番目のフィールドで使用するチップを選択してください。次に 3 番目のフィールドでプログラミングしたいファイルを選択してください。チップオプションとシリアルナンバー設定は特別な場合にのみ必要で、通常は設定する必要がありません。プログラミングプロセスを開始するには青いボタンをクリックしてください。

上部のデータエントリーフィールドについては「データエントリーフィールド」の章で解説しています。プロセス・コントロールの一般的な表示については「プロセス・コントロール」の章で解説しています。

チップをコピーする

「チップをコピーする」モードは1つのチップから1つ以上のチップにプログラミングするデータに最適化されたインターフェースを提供します。



チップをコピーする



BX32 Batupo II
シリアルナンバー: BL2507791
ファームウェアバージョン: 02.30

元のチップ: Winbond W29EE011P-90
アダプター: PLCC32-DIP32
128 Kバイト (1 MBit)

コピー先のチップ: Winbond W29EE011P-90
アダプター: PLCC32-DIP32
128 Kバイト (1 MBit)

リフレッシュ

チップ自動アイデンティファイ

チップ自動アイデンティファイ

 **コピープロセスをはじめる**

コピー数
1

このインターフェースは上部のデータエントリー・フィールドと下部のプロセス・コントロール機能で構成されます。

複数のプログラミング・デバイスを接続した場合は最初のフィールドで使用するデバイスを選択してください。次に2番目のフィールドでコピー元のチップを選択し、3番目のフィールドでコピー先のチップを選択してください。異なるソースとコピー先のチップを選択することも可能です。デバイス内でコピーがオリジナルと同様に機能するためには、両方のチップが同じメモリ容量、同じピン・アサインメント、同じコントロール・アルゴリズムを使用している必要があります。コピープロセスを開始するには青色のプレイボタンをクリックし

上部のデータエントリー・フィールドについては「データエントリー・フィールド」の章で解説しています。プロセス・コントロールの一般的な表示については「プロセス・コントロール」の章で解説しています。

てください。

「コピー先のチップ署名を確認」の手順に進む前に、コピー先のチップを挿入するように指示されます。(「コピー先のチップ署名を確認」機能の有無に関わらず表示されます。)

チップを読み込む

「チップを読み込む」モードは1つのチップからデータを読み込み、ファイルに保存する、またはHexエディタで閲覧するために最適化されたインターフェースを提供します。



チップを読み込み

	BX32 Batupo II シリアルナンバー: BL2507791 ファームウェアバージョン: 02.30	リフレッシュ
	元のチップ: Winbond W29EE011P-90 アダプター: PLCC32-DIP32 128 Kバイト (1 MBit)	チップ自動アイデンティファイ
	ReadData.bin パス: Y:\Files\	ブラウズ
	チップオプション(任意) 特別なオプションは使用されていません	オプションを変える

 **読み込みプロセスをはじめる**

このインターフェースは上部のデータエントリー・フィールドと下部のプロセス・コントロール機能で構成されます。

複数のプログラミング・デバイスを接続した場合は最初のフィールドで使用するデバイスを選択してください。次に2番目のフィールドで読み込むチップを選択してください。

プロセス・オプションで「バッファデータを保存する」が有効になっている場合、データは読み込まれるとすぐにファイルに保存されます。このオプションが有効になっている場合、3番目のフィールドを選択することができ、ファイルパスやファイル名を指定することができます。

プロセス・オプションで「Hexエディタを表示する」が有効になっている場合、データは読み込まれるとすぐにHexエディタに表示されます。

コピープロセスを開始するには青色のプレイボタンをクリックしてください。

上部のデータエントリー・フィールドについては「データエントリー・フィールド」の章で解説しています。プロセス・コントロールの一般的な表示については「プロセス・コントロール」の章で解説しています。

プロダクション・モード

「プロダクション」モードは複数のプログラミングデバイス上で複数のチップをプログラミングするために最適化されたインターフェースを提供します。



このモードでは最大 8 個の USB チッププログラマと 8 個のプロフェッショナル・プログラマシリーズ (BX-プログラミングデバイス) を同時に制御することができます。全ての認識されたデバイスがデバイスセレクションリストに表示されます。(図の右側)。

高い生産性を実現するため、個別のプログラマデバイスは個別に制御されます。複数のデバイスの個別のプログラミング・スピードはデバイスを 1 個だけ使用している際とほぼ変わりません。

ハブを使用している場合、不必要に転送速度が制限されないように、PC の USB コネクションとハブは USB 2.0 High Speed モードをサポートしている必要があります。

上部のデータエントリー・フィールドについては「データエントリー・フィールド」の章で解説しています。プロセス・コントロールの一般的な表示については「プロセス・コントロール」の章で解説しています。

プロダクションモードではスクリーン右側にプログラミングデバイス概要が表示されます。スクリーンの上部にはツールバーが表示されます。これによって個別のプログラミングデバイスのチップ、ファイル、オプションの設定を個別に有効にすることができます。特定のプログラミングでデバイスの個別設定が有効な時は対応するボタンがオレンジ色になっています。



現在接続されているプログラミングデバイスのリストをリフレッシュする場合、リフレッシュボタンをクリックしてください。

接続された個々のプログラミングデバイスに対しデバイスの情報と 5、6 個のボタンが個別に表示されます。



プロダクションプロセスからプログラミングデバイスを取り除く場合、このボタンをクリックしてください。このボタンがオレンジ色になっている時はそのプログラミングデバイスがプロダクションプロセスに含まれています。



作業中の複数のプログラミングデバイスの中から特定のプログラミングデバイスを検出するには、このボタンをクリックしてください。個別のデバイスの緑色のオペレーションライトが短く光ります。



個別のプログラミングデバイスから別々のチップを選択する場合、このボタンをクリックしてください。これにより例えば 1 つのプロダクションプロセスで 4 つのチップをプログラミングするために 4 つのプログラミングデバイスを使用することができます。



個別のプログラミングデバイスに別々のファイルを使用する場合このボタンをクリックしてください。これにより例えば 1 つのプロダクションプロセスで 4 つのファイルをプログラミングするために 4 つのプログラミングデバイスを使用することができます。



このボタンで個別のプログラミングデバイスの異なる設定（オフセット、スプリットなど）を選択することができます。これにより例えば 1 つのプロダクションプロセスでスプリット設定（奇数、偶数）の異なる 2 つのプログラミングデバイスを使用することができます。



使用するチップがシグネチャーを含まない場合はこのチップの挿入や取り出しは自動的に検出されません。このチップには「チップを待つ」のプロセス手順が使用できません。そのため、各デバイスのプロセス・シーケンスはチップ挿入後、該当デバイスのスタートボタンを押して開始しなければなりません。

BX32 Batupo II (BL2507791)
 チップ: Winbond W29EE011P-90
 ファイル: ProgramData1MBit.bin

成功: 0/1

BX32P Barlino II (BL2507789)
 チップ: Winbond W29EE011P-90
 ファイル: ProgramData1MBit.bin

成功: 0/1












BX40 Bagero II (GF07000003)
 チップ: Winbond W29EE011P-90
 ファイル: ProgramData1MBit.bin

成功: 0/1

BX48 Batego II (DC1909827)
 チップ: Winbond W29EE011P-90
 ファイル: ProgramData1MBit.bin

成功: 0/1

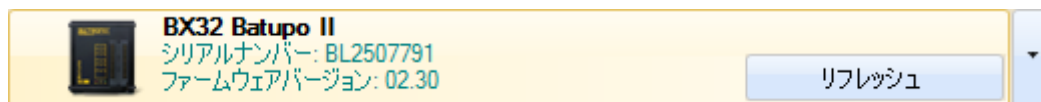
右上隅に各プログラミングデバイスのステータスを示すシンボルが表示されます。

-  ファイルがバッファに読み込み中です。
-  デバイスはチップ挿入を待機中です。
-  チップ署名を検査中です。
-  プロテクションビットが取り出し中です。
-  挿入されたチップはデータ消去中です。
-  チップがブランクか確認中です。
-  チップをプログラミング中です。
-  プログラムされたデータを確認中です。
-  チップは書き込み保護中です。
-  チップオプションを比較中です。
-  デバイスはチップ取り出しを待機中です。

データエントリー・フィールド

「チップをプログラミングする」、「チップをコピーする」、「チップを読み込む」、「プロダクションモード」では画面上部にいくつかのデータエントリー・フィールドがあります。

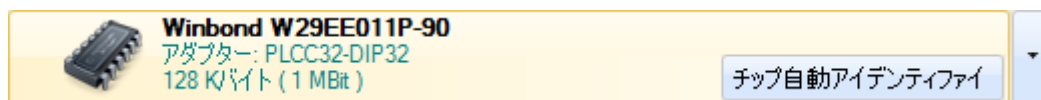
プログラマ



選択したプログラミングデバイスのシリアルナンバーとファームウェア・バージョンがこのフィールドに表示されます。このボタンの左側のエリアを押すと選択したプログラミングデバイスの上の緑色の LED ライトが数度点滅します。この機能は複数のプログラミングデバイスを接続中で、特定のデバイスを識別するのに役立ちます。

右側の矢印をクリックすると接続中の全てのプログラミングデバイスを表示するリストを開きます。この選択リストを開くと、現在接続中のデバイスが検出され、リストがリフレッシュされます。この選択リストで必要なデバイスをクリックし、選択します。

コピー元チップとコピー先チップ

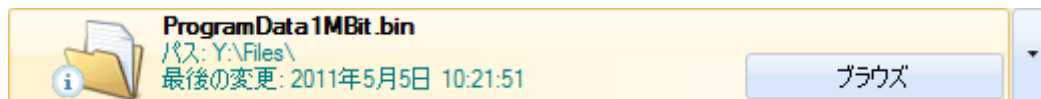


画面上ではチップ名、メモリ容量、必要な場合が適切なアダプタ名が表示されます。ボタンの左エリアをクリックするとチップを選択するためのチップブラウザが開きます。

「チップ自動識別」ボタンは、署名を利用して自動的にチップを選択することができます。（「チップ自動識別」の章を参照）

右側の矢印をクリックすると最近使った 10 個のチップのリストを表示します。この選択リストで必要なチップをクリックし、選択します。

コピー元ファイル



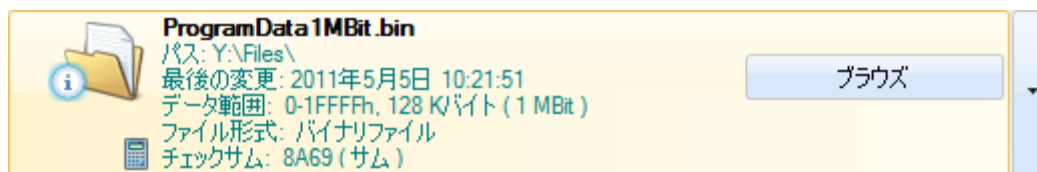
「ファイルを開く」のデータエントリー・フィールドで読み込むファイルを選択します。ファイルのパスと最後にファイルが変更された日時がファイルフレームの下に表示されます。ボタンの左側をクリックするとファイルを選択するためのファイルブラウザが開きます。

右側の矢印をクリックすると最後に使った 10 個のファイルが表示されます。この選択リストで必要なファイルをクリックし、選択します。

最後に使ったファイルの選択リスト

右側の矢印をクリックすると最後に使った 10 個のファイルを開くことができます。必要なファイルをクリックするとファイルを開くことができます。

ファイルの詳細情報

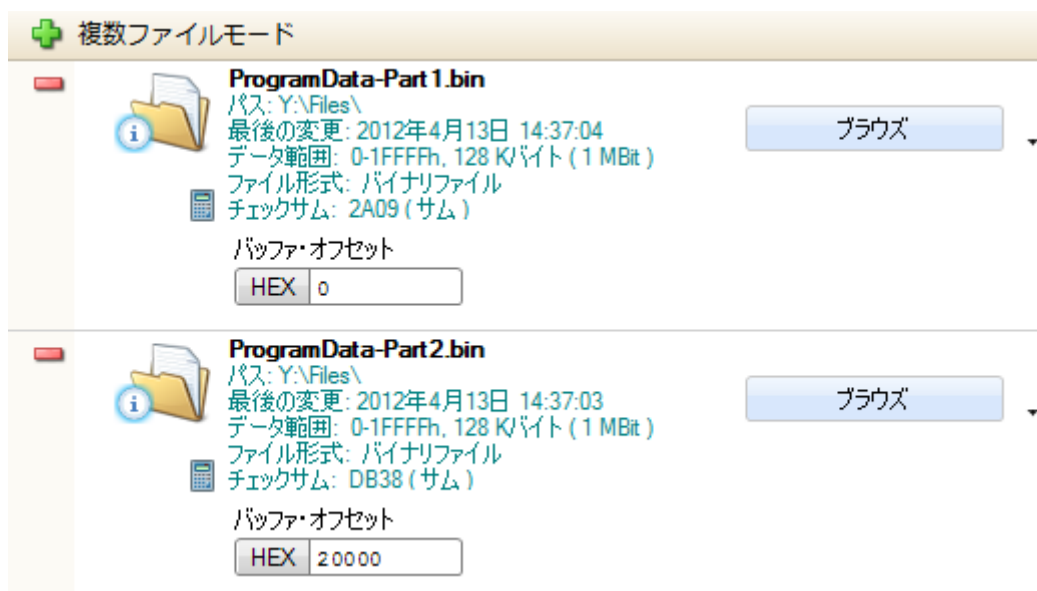


フォルダーアイコンの左側にある小さいiマークをクリックすると、詳細情報が表示されます。計算機マークをクリックすると、チェックサム・アルゴリズムが選択できます。

コンテキストメニュー

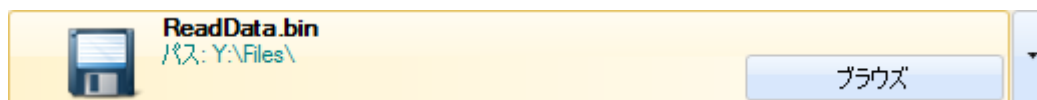
「ファイルを開く」ボタンをマウスで右クリックするとコンテキストメニューが開きます。このコンテキスト・メニューには詳細な機能があります。例えば Explorer でフォルダを開く場合、複数のファイルモードの有効化/無効化や選択したファイルの様々な情報の表示ができます。

複数ファイルモード



複数ファイルモードはコンテキストメニューで有効・無効にできます。複数ファイルモードでは複数のファイルを選択し、次々にプログラミングをすることができます。各ファイルの場所は入力ボックス「バッファオフセット」で調整することができます。

ファイルを保存する



「ファイルを保存する」のデータエントリーフィールドでチップデータを読み込み後、保存場所と保存ファイル名を特定することができます。最近使ったバッファのコンテンツは「バッファデータを保存する」の手順で保存することができます。

右側の矢印を開くと最近使った10個のファイルを表示します。選択リストの中から使用するファイルをクリックし、選択することができます。

「ファイルを保存する」フィールドでは下記の特別な符号を動的置き換えのために使用することができます。：

！ 感嘆符は選択したチップ名と置き換えられます。

シャープは1以上の数字と置き換えられます。既存のファイルでそのファイル名がある場合、数字が1ずつ増えたファイル名がつけられます。これはそのファイル名をもつファイルが無くなるまで続きます。

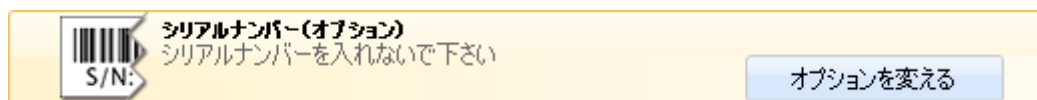
例: ファイル名 AT27C010 からデータを読み込み、「ファイルを保存する」ダイアログでファイル名「File-!-#.bin」を入力します。そのデータはファイル名「File-AT27C010-1.bin」で保存されます。チップを再度読み込み、保存するとそのデータはファイル名「File-AT27C010-2.bin」で保存されます。

チップオプション



チップのデータの焼き付けやチップの特別な機能の利用等でデータを変更するために、特別なチップオプションを使うことができます。この設定が何のために使われるかよくわからない場合またはチップに単にデータを書き込みたい場合は、チップオプションをオンにしないでください。このボタンをクリックするとチップオプション設定の画面が表示されます。詳しくは「チップオプション」の章で解説しています。

シリアルナンバー

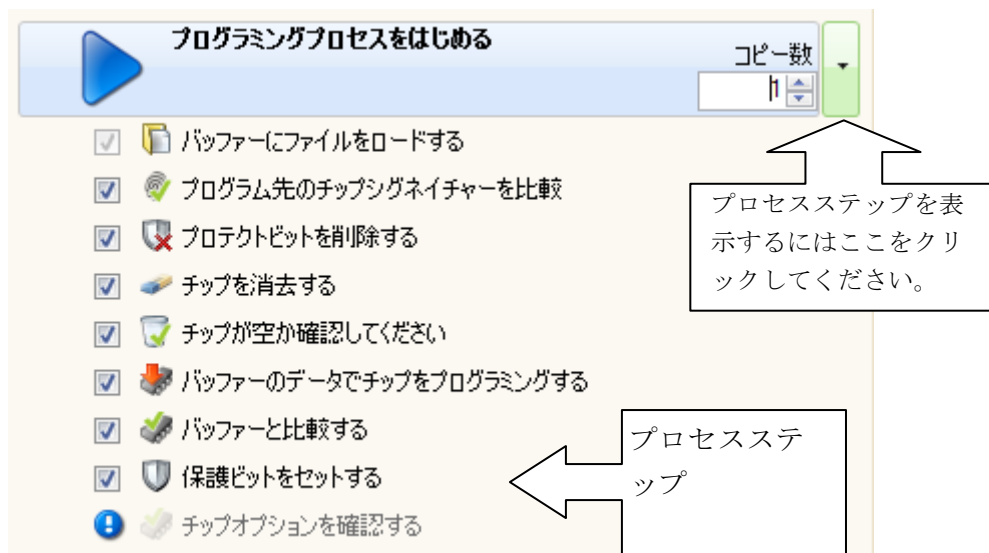


プロダクションプロセスでは書き込み前にデータにシリアルナンバーを入力できます。

このボタンをクリックするとシリアルナンバー設定の画面を表示します。詳しくは「シリアルナンバー」の章で解説しています。

プロセスコントロール

プロセスコントロールについて本章で解説しています。ソフトウェアモード「チップをプログラミングする」、「チップをコピーする」、「チップを読み込む」、「プロダクション」と同じです。各モードには「プレイ」（青い三角形）ボタンのある青いバーがあります。



右側の矢印をクリックすると全手順リストが表示されます。プロセスが開始されるとこのリストが自動的に表示されます。

各プロセスは個別のステップで構成されています。各ステップは個別のチェックボックスでオン・オフの切り替えができます。各ステップのボタンをクリックすると全体のステップの中の一部のステップを個別に実行することができます。

「コピー」のデータエントリーフィールドは該当するプロセスが何度実行されたかを特定することができます。これは「プログラミング」モードでは何枚のチップがプログラミングされたかを特定し、「コピーする」モードではコピー元のチップからいくつのコピーが作成されたかを特定することができます。

コントロール



このボタンをクリックしプロセスが開始された後、該当プロセスは上から下の順で実行されます。



このマークをクリックすると（プロセス実行中のみ表示されます）プロセスを中止し、チップはこれ以上プログラミングされません。

スペシャルコントロール（「プロダクションモード」のみ）



このマークをクリックすると（プロセスのみ表示されます）プロセスを中止しチップはこれ以上プログラミングされません。しかし現在有効なプロセスステップは完了します。



このマークはプロダクションモードで中止ボタンをクリックした後に表示されます。通常のプロセス・シーケンスが中止されたことを示し、現在アクティブなプロセスステッ

プのみが完了します。このボタンをクリックすると現在実行中のプロセスがすぐに終了します。

プロセスステップ

ファイルをバッファに読み込む

このプロセスステップでは「ファイルを開く」フィールドで特定したファイルをバッファに読み込みます。

バッファをファイルに保存する

バッファからのデータをファイルに保存します。存在しないファイルは自動的に作成され、既存のファイルは上書きされます。

コピー元/コピー先のチップ署名を確認する

挿入したチップの署名と「コピー元チップ」/「コピー先チップ」の設定を比較します。相違が見つかったら、「プロセスを中止する」、「署名を再確認する」「相違を無視しプロセスを続ける」の選択肢があるダイアログが表示されます。

チップを消去する

このプロセスはプログラミングデバイス内のチップを消去します。一般的にチップを消去するとは全ビットが 1 に設定されるということです。そのため消去されたチップ内の全バイトは FFh になります。(Byte FFh = 11111111)

プログラミングデバイスを使って全てのチップが消去できる訳ではありません。例えば、ガラス窓付きの 27c EPROM's は EPROM イレーザの強い紫外線 C 光線でのみ消去できます。ガラス窓のない 27c EPROM's は消去できないので、新しいデータをプログラミングすることができません。こういったチップには通常 OTP = One Time Programmable の表示があります。

チップが空か確認する

これはチップが消去されたか、つまり全ビットが 1 にセットされたかを確認します。このプロセスではチップの全コンテンツを読み込み、全ビットを確認します。

チップデータをバッファに読み込む

コピー元のチップを読み込みそのデータをバッファに書き込みます。バッファ内のデータは Hex エディタを使って閲覧、表示することができます。(「Hex エディタ」の章を参照)

バッファでチップに書き込む

「プログラミングオプション」、「シリアルナンバー設定」を使って、バッファ内のデータをチップに書き込みます。

バッファでチップを確認する

完全なチップコンテンツを読み込み、全バイトがバッファにある全バイトを比較します。

保護ビットを設定する

後日チップが誤って上書きされることを防ぐいわゆる「保護ビット」を設定します。この機能は全てのチップをサポートしている訳ではありません。

Hex エディタバッファを表示する

これはソフトウェアを Hex エディタモードに切り替えます。

チップ挿入を自動で待機


対応するプログラミングが新しいチップが挿入されるのを待機します。チップが挿入されるとチップ署名を確認します。使用するチップが署名を含まない場合、このチップの取り出しと挿入は自動で検出されません。プロセス機能「チップ挿入を自動で待機」はこのようなチップには無効です。この場合対応するデバイスのスタートボタンを押してプロセスを開始してください。

チップ取り外しを自動で待機

対応するプログラミングデバイスはチップが取り外されるのを待機します。

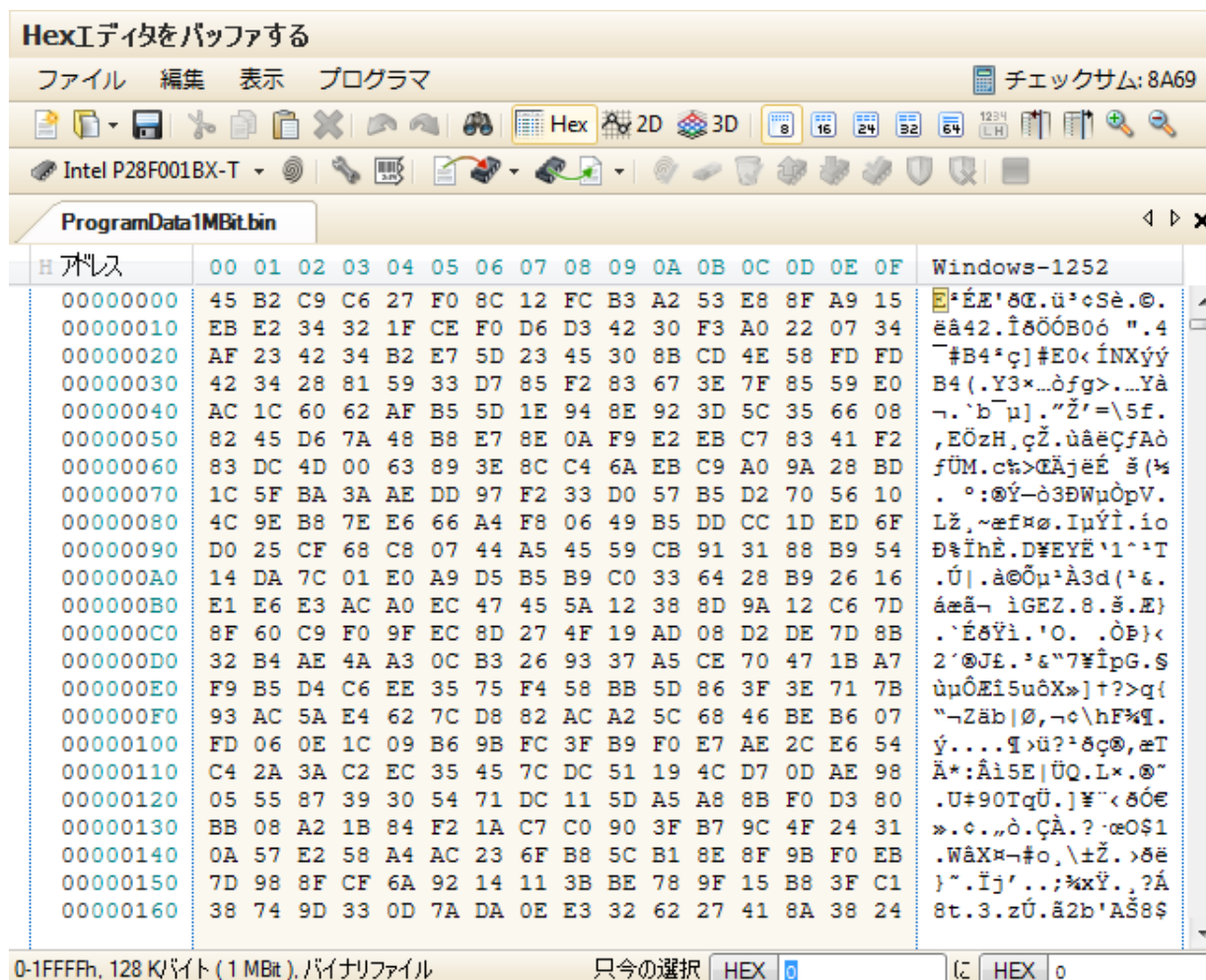
プロセスが完了すると、チップを取り外せることを示すためにデバイスの上の動作ボタンが点滅します。対応するプログラミングはチップが取り外されるのを待機します。チップを取り外した後緑色 LED 動作ボタンが点滅した場合、このデバイスはプログラミングする次のチップを待機していることを示しています。緑色 LED 動作ボタンが点滅しない場合、このデバイスにはこれ以上プログラミングするチップがセットされていないことを示しています。

使用するチップが署名を含まない場合、このチップの取り出しと挿入は自動で検出されません。プロセス機能「チップ挿入を自動で待機」はこのようなチップには無効です。

 プロセスステップが選択したチップやプログラミングデバイスにサポートされていない場合、このマークがそのステップの横に表示されます。

Hex エディタ

Hex エディタはバイナリデータを閲覧、編集するために使用します。これには包括的表示機能と使いやすい編集機能があります。



この編集機能はツールバーボタンかコンテキストメニューから選択することができます。コンテキストメニューはマウスの右クリックでアクセスできます。







ステータスバーはデータサイズ、ファイルタイプ、現在の選択を表示します。現在の選択の最初と最後のアドレスをここで編集することができます。

プログラミングデバイスを使用中の場合、ステータスバーは現在のプロセスステップ情報とプログレスバーを表示します。
















Hex エディタメニューとツールバー

このツールバーは Hex エディタ内のデータをナビゲートし、操作する機能を含みます。











メニューとツールバー「ファイル」

	新しいウィンドウ	新しい空白の Hex エディタ・ウィンドウを開きます。
	ファイルを開く	既存のファイルを開きます。このボタンの右側の矢印をクリックすると最近使った 10 個のファイルのリストが開きます。
	閉じる	現在のドキュメントを閉じます。
	全てのドキュメントを閉じる	開いている全てのドキュメントを閉じます。
	その他全てを閉じる	現在使用しているもの以外の全てのドキュメントを閉じます。
	保存する	現在のバッファコンテンツを保存します。
	名前をつけて保存する	新しいファイル名でデータを保存します。 ファイルフォーマット: ファイルフォーマットは自動検出される (プリセット) か手動で特定できます。この設定は全ての Hex エディタの開始、保存機能に使われます。
	印刷	現在のドキュメントを印刷します。
	ファイルをインポートする	オフセットおよび/または Hex エディタデータとともに既存のファイルをインポートします。
	Hex エディタとファイルのデータと比較する	Hex エディタデータと既存のファイルのデータを比較します。
	ファイルフォーマット	ファイルを開くのにファイルフォーマットを自動検出または固定のファイルフォーマットを特定するかを選択します。
	最近使ったファイル	このダイアログでは最近使ったファイルを開くことができます。

メニューとツールバー「編集」

	取消	最後の操作を取り消します。(取消)
	やり直し	取り消された操作をやり直します。(やり直し)
	切り取り	選択したデータを切り取ります。
	コピー	選択したデータをコピーします。
	フォーマットされたデータをコピー	選択したデータとそのフォーマットをコピーします。
	ペースト	カーソルの位置にコピーしたデータを貼付けます。
	削除	選択したデータを削除します。
	全選択	エディタにある全データを選択します。
	手動選択	データの特定の部分を選択します。
	検索と置換	検索と置換ダイアログを開きます。
	最初のバイトを表示...	FFh ではない最初のバイトを表示します。
	次のバイトを表示...	FFh ではない次のバイトを表示します。
	最後のバイトを表示...	FFh ではない最後のバイトを表示します。
	チェックサムを計算する	選択した範囲のチェックサムを計算します。
	に挿入する	特定の個数の選択可能なデータ・ワードを挿入します。
	選択した部分を満たす	ランダムな数字や数字の調整可能なシーケンスの選択可能な値で選択範囲を埋めます。
	値を増やす/減らす	選択範囲の値を選択した値やパーセント値で増やしたり / 減らしたりできます。
	選択範囲をコピー	選択範囲を選択可能なアドレスにコピーできます。
	データをスワップする	選択範囲の隣り合ったワードをスワップできます。
	データブロックをスワップする	選択範囲を別の範囲とスワップできます。

メニューとツールバー「ビュー」

エディタビュー	Hex エディタビューでの表示。
	メニューのこの部分はアクティベートしたチップ-チューナーモジュール内でのみ閲覧できます！
2D ビュー	2D ビューでのデータ表示。 メニューのこの部分はアクティベートしたチップ-チューナーモジュール内でのみ閲覧できます！
3D ビュー	3D ビューでのデータ表示。 メニューのこの部分はアクティベートしたチップ-チューナーモジュール内でのみ閲覧できます！
ワードビュー	ワードの長さの調整
	データをバイトビュー (8bit) で表示
	データをワードビュー (16 bit) で表示
	データをワードビュー (24 bit) で表示
	データをダブルワードビュー (32 bit) で表示
	データをクアドワードビュー (64 bit) で表示
エンディアン	バイト順を調整。ターゲットシステムにより最初のバイトを最低 (LSB) または最高 (MSB) のバイトにすることができます。
ディスプレイ幅	エディタの幅を増やしたり、減らしたりすることができます。
	エディタの一行に表示されるバイト数を減らします。
	エディタの一行に表示されるバイト数を増やします。
オフセット	
増やす	画面の最初のバイトの位置を右側に移動します。
減らす	画面の最初のバイトの位置を左側に移動します。
	左側
	エディタの左側の表示を選択します。
	右側
	エディタの右側の表示を選択します。
	スクロールを同期させる...
	全ての閲覧可能なエディタのスクロールが同期します。この動作によってデータを手動で比較できます。

メニューとツールバー「プログラマ」

チップをプログラミングする全ての基本機能は Hex エディタにもあります。左側のドロップダウン・ボックスを使ってチップを選択してください。ここからチップを簡単に選択するためにチップブラウザにアクセスできます。



接続中のプログラマ

プログラマ選択

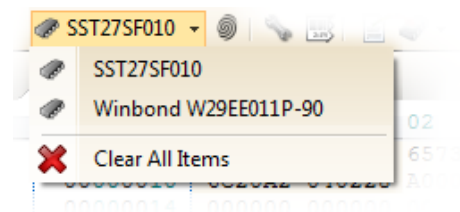


プログラマ表示

デバイスが特定できるように選択したデバイスの上の動作ライトが点滅します。



チップブラウザの(ツールバー) 選択。右側の三角ボタンを一度クリックすると最近使ったチップのリストを開くことができます。



チップブラウザ

チップブラウザを開きます。



自動識別

チップ自動識別(「チップ自動アイデンティファイ」の章参照)

最近使った 10 個のチップ

直接選択するために最近使った 10 個のチップを表示します。



チップオプション

ここでチップの個別オプションを設定することができます。



シリアルナンバー










ここでチップのシリアルナンバーをセットすることができます。



完全なプログラマプロセスを開始します。プロセスのデフォルト設定はプロセスステップ「チップ署名を確認する」、「保護ビットを取り除く」、「チップを消去する」、「チップが空か確認する」、「バッファでチップをプログラミングする」、「保護ビットを設定する」を含みます。チップにサポートされていないプロセスステップは自動的に無効にされます。黒い矢印をクリックし、プロセスステップリストを開き、その有効・無効を変更することができます。



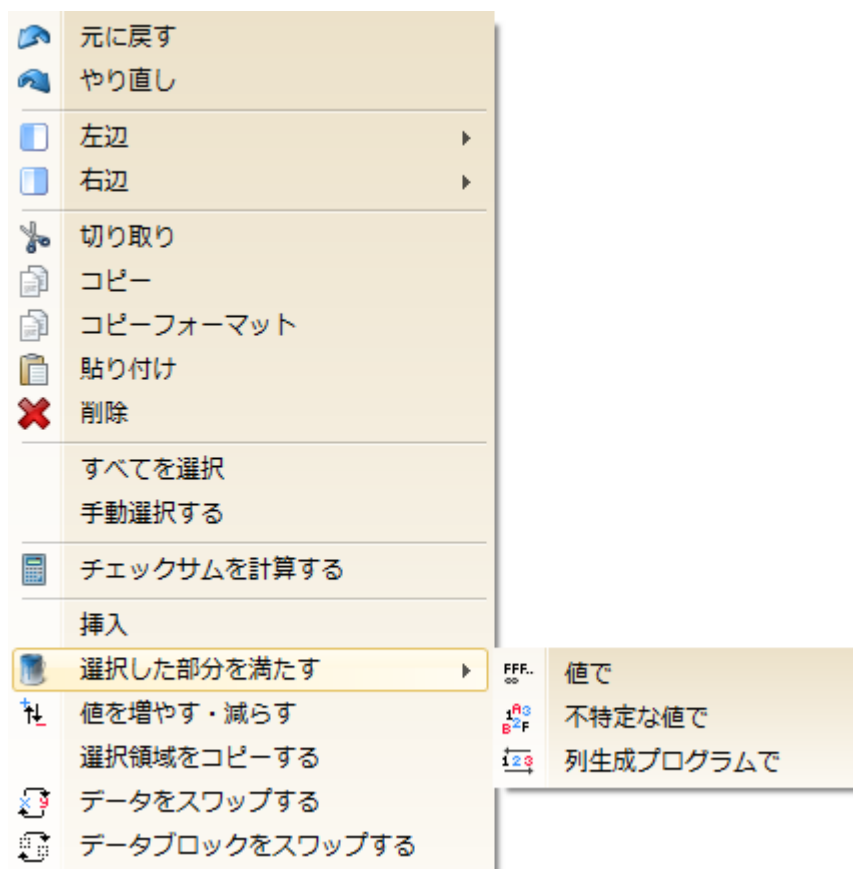
完全な読み込みプロセスを開始します。プロセスのデフォルト設定はプロセスステップ「チップ署名を確認する」、「チップをバッファに読み込む」、「チップをバッファで確認する」を含みます。黒い矢印をクリックし、その有効・無効を変更することができます。

	チップ署名を確認する	チップ署名を確認します。
	チップを消去する	チップを消去します。
	チップが空か確認する	チップが消去されたか確認します。
	チップデータ読み込み	チップデータをバッファに読み込みます。
	...	
	チップをプログラミングする...	バッファデータをチップにプログラミングします。
	チップデータを確認する...	バッファデータとチップデータを比較します。
	保護ビットを設定する	保護ビットを設定します。
	保護ビットを取り除く	保護ビットを除去します。
	現在のプロセスを中止する	現在のプログラミングプロセスを中止します。 (他のモジュールで実行中のプロセスには適用されません！)

Hex エディタ・コンテキストメニュー

Hex エディタでマウスの右クリックをするとコンテキストメニューを開くことができます。

使用できる機能は「Hex エディタメニューとツールバー」の章で解説しています。

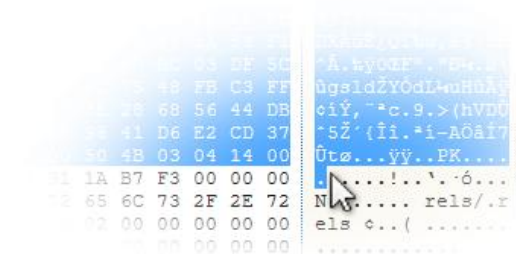


選択したデータで作業する

このオプションは Hex-Editor のツールバーの選択ボックスとコンテキストメニューから 選択することができます。

範囲を選択する

マウスで選択します。



Hex-Editor 内で 16 進数コードの特定の範囲を選択し、編集することができます。選択範囲は青色でマークアップされます。

入力による詳細な選択



画面の下側にあるエントリーフィールドに使用するブロックの最初と最後の位置を入力して範囲を選択することができます。この方法でより広い範囲を選択することができます。

範囲を編集する

これはメニュー「編集」、ツールバー、コンテキストメニューから選択することができます。

手動選択

どのデータを選択するか特定することができます。

選択範囲のチェックサムを計算する

選択可能なアルゴリズムを使って選択範囲のチェックサムの値を計算します。(sum, MD5, SHA-1, EPT1 CRC16 または CRC32)

値で選択範囲を満たす

選択範囲を特定の値で満たします。

ランダムな値で選択範囲を満たす

選択範囲をランダムな値で満たします。

シーケンスジェネレータで選択範囲を満たす

選択範囲を値の特定の値域で満たします。

値を増やす/減らす

特定の数字やパーセンテージ値で選択範囲の全ての値を増やしたり減らしたりします。

選択範囲をコピーする

選択したデータを特定のアドレスにコピーします。コピー先の範囲を広げるか上書きするか選択することができます。

データをスワップする

設定により最初と 2 番目のバイト/ワード/ダブルワード/クアドワードの値を互いに交換することができます。1 つ以上のペアが選択された場合も同じ方法で交換することができます。(例えばバイト 1 をバイト 2 と、バイト 3 をバイト 4 と切り替える等)

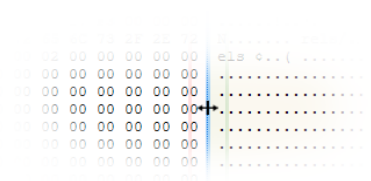
データブロックをスワップする

選択範囲のデータを特定のオフセットアドレスで開始したデータと交換します。

マウスを使ってビューを変更する

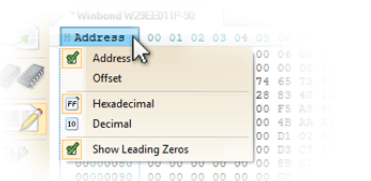
「ビュー」メニューの中の調整オプション以外にも Hex エディタ内でマウスを使って直接できるいくつかの設定があります。

ディスプレイ幅を調整する



マウスを使ってディスプレイ幅を変更することができます。Hex エディタ内の左右の範囲の分け目をクリックしてください。同じ機能を使って右寄せにすることができます。

アドレス表示の調整



「アドレス」の列をクリックすると、「アドレス」（ワードアドレス）またはオフセット（バイトアドレス）のどちらかを選ぶ選択ダイアログを表示します。1 バイトのワード長の場合どちらも同じです。アドレスは先行ゼロ列有りまたは無しで 10 進法または 16 進数で表示できます。.

Hex エディタショートカットキー

Hex エディタには下記のショートカットキーがあります。:

- <CTRL+Z>: 取消
- <CTRL+Y>: やり直し
- <Page Up>: 現在表示中のページの先頭に移動します。カーソルが既に一行目にある場合は 1 ページ上に移動します。
- <Page Down>: 現在表示中のページの最後に移動します。カーソルが既に一行目にある場合は 1 ページ下に移動します。
- <CTRL+Home>: Hex エディタ内の最初のバイトにジャンプします。
- <End>: 現在の列の最後のバイトにジャンプします。
- <CTRL+End>: Hex エディタ内の最後のバイトにジャンプします。
- <Insert>: 上書きモードと挿入モードを切り替えます。上書きモード（標準設定）では新しいデータが入力されると、現在カーソルが置かれているデータが上書きされます。挿入モードでは新しいデータは既存のデータを上書きせずに挿入されます。
- <CTRL+A>: 全データを選択します。
- <Shift+Cursor key>: 選択範囲を広げます。
- <CTRL+X>: 選択したデータを切り取り、クリップボードに貼付けます。
- <CTRL+C>: 選択したデータをクリップボードにコピーします。
- <CTRL+V>: クリップボードからデータを貼付けます。
- <Delete>: 選択したデータを削除します。
- <Tab>: HEX モードと ANSI モードを切り替えます。
- <CTRL+H>: 16 進数ビュー
- <CTRL+D>: 10 進数ビュー
- <CTRL+T>: テキストビュー
- <CTRL+Cursor key right/left>: ディスプレイ幅を広げる / 狭める
- <CTRL+Alt+ Cursor key right/left>: ディスプレイオフセットを広げる / 狭める
- <CTRL+1>: 1 バイトのワード長の表示 (8 Bit)

- <CTRL+2>: 2 バイトのワード長の表示 (16 Bit)
- <CTRL+3>: 4 バイトのワード長の表示(32 Bit)
- <CTRL+4>: 8 バイトのワード長の表示(64 Bit)

ソフトウェアオプション

ソフトウェアオプションは Prog-Express を開いたり閉じる際に、メッセージの表示、オーディオファイルのプレイバック、特別なプログラミングオプション、オペレーターモード、言語設定を設定するために使用できます。



ソフトウェアオプションは別々のページ「一般」、「プロジェクト」、「セキュリティ」、「ファイルの関連付け」、「言語」、「アドオン」で表示できます。

一般オプション

ソフトのオプション

全般	プロジェクト	セキュリティ	ファイルの関連付け	言語	アドオン
----	--------	--------	-----------	----	------

Prog-Express スタートオプション

- ☒ 最後に使ったセッティングを自動で読み込む
- ☐ 最後に使ったプロジェクトを自動で読み込む
- ☐ 次のプロジェクトを自動で読み込む

☐ Batronix ウィンドウデコレーション

PROG-EXPRESS の終了オプション

- ☒ プロジェクトファイルを自動で保存する

作業ディレクトリ

- ☐ 作業中のディレクトリ内のファイルダイアログを開く

自動アップデート

- ☐ 起動時にオンラインでアップデートファイルの検索

メッセージ

- ☒ プログラミング中に書き込みデータがチップの容量を超える場合は警告する
- ☐ チップが特殊な配置になる場合には、警告を表示する

音

- ☒ プロセスが無事に終了した時に音声ファイルを鳴らす:
- ☒ プロセスの失敗の時に音声ファイルを鳴らす:

Prog-Express スタートオプション

ソフトウェアは起動する際に、最近使ったプロジェクトまたは特定のプロジェクトを直接読み込むことができます。1つのプロジェクトは選択したモード、選択したチップ、プログラミングオプション、シリアルナンバーオプション、有効なプロセスステップを含みます。

Batronic ウィンドウデコレーション

ユーザーは Windows の標準フレームのデコレーションをプログラムの特定のフレームデザインに変更することができます。

Prog-Express 終了オプション

Prog-Express を終了する際ソフトウェアは実際の設定を最後に読み込んだファイルに自動的に保存します。(デフォルト)

作業ディレクトリ

ソフトウェアは常に特定のディレクトリでプロジェクトブラウザを開始できます。このオプションを使用しない場合、ブラウザは最近使ったディレクトリで開始します。

自動アップデート

ソフトウェアの実行中にオンラインでアップデートを確認することができます。アップデートが可能な場合自動でダウンロードし、インストールします。

メッセージ

ここで個別のメッセージの表示のオン・オフを切り替えることができます。

音

プロセス終了後またはプログラミングエラーの後ソフトウェアはサウンドファイルを流すことができます。いくつかのファイルは Prog-Express ソフトウェアのサブフォルダ「音」に入っています。ご使用の PC からお持ちの.wav サウンドファイルを使用することもできます。

プロジェクト

ソフトのオプション

全般	プロジェクト	セキュリティ	ファイルの関連付け	言語	アドオン
----	--------	--------	-----------	----	------

データ処理

未使用バイトを満たす

 16進法

高度なフリーチップアドレスエリアにバッファデータを反映する

☐ あるチップから、より容量の大きなチップにデータをプログラムするにはこのオプションを有効にしてください。

16ビットチップのバイト順に並べ替える:

- ☒ 最上位ビットの前に 最下位ビットをプログラムする (LSB – MSB、標準)
- ☐ 最下位ビットの前に 最上位ビットをプログラムする (MSB – LSB)

Multi-Pass確認 (BX32-II, BX32P-II, BX40-I/II, BX48-I/II, BX448 & BX848のみ)

- ☐ 最小の電圧を確認
- ☒ 公称電圧を確認
- ☐ 最大の電圧を確認

セキュリティとテスト機能

- ☒ ピン接触をテストする(BX48, BX448 & BX848のみ)
- ☒ 過電流のモニタリングを有効にする(BX48, BX448 & BX848のみ)
- ☒ チップのチェックサムを計算し、ログ領域に表示

データ処理

ここでは数種類の特別設定ができます。

未使用バイトとはデータが定義されていないバイトのことです。例えば小さいファイルをそれより大きいチップにプログラミングした場合にこれが起こります。

バイトシーケンスはチップをプログラミングする場合にのみ適用されます。ファイルからの順番でバイトが使用されるか特定することができます

Multi-Pass 比較

BX40 は multi-pass 比較をサポートしています。このプロセス中チップデータは読み込まれ、異なる供給電圧と何度も比較されます。データに不一致があった場合比較が中断され、エラー表示がでます。



使用される電圧はチップの許容作動電圧によります。例えば多くのチップは $\pm 10\%$ の電圧範囲を許容しています。この場合例えば公称電圧が 5 ボルトであれば比較は 4.5、5.0、5.5 ボルトまで許容されます、プログラミングしたチップの比較を繰り返すとプログラムデータを高度に保証することができます。これにより 100%プログラミングされていないチップを除外し、最終的にデバイスを非常に強い変動電圧の状態に置くことを防ぐことができます。

セキュリティとテスト機能

ある種のプログラマはプロセス開始前にピン接触をテストし、プロセス実行中にチップ電流を測定することができます。これらの機能はセキュリティを強化するので定期的に有効にしましょう。しかしながらいくつか特別なケースではこの機能を無効にした方がいい場合もあります。例えばダメージのある恐れがあるチップの電流入力が製造者が指定している電流よりも大幅に高い場合、過電流測定はオフにできます。製造者が指定しているよりも高い電流が必要になりますが、こうすることでチップを使用することができます。

セキュリティ

ソフトのオプション

全般	プロジェクト	セキュリティ	ファイルの関連付け	言語	アドオン
オペレーターモード					
設定		ソフトウェア	プロジェクト		
アプリケーション開始後オペレーターモードをアクティベートしてください		<input type="checkbox"/>	<input type="checkbox"/>		
モードセクションもロックする		<input type="checkbox"/>	<input type="checkbox"/>		
オプションダイアログとオペレーターモードをパスワードでロックする		<input type="checkbox"/>	<input type="checkbox"/>		
パスワード		<input type="text"/>	<input type="text"/>		
 PROG-EXPRESSのタイトルバーにある鍵アイコンをクリックしてオペレーターモードを有効にする					
 ソフトウェアやプロジェクトコンテキストでオペレーターモードが有効な場合、Prog-Expressは常に制限のある方を利用します					

オペレーターモード

オペレーターモードを有効にするとソフトウェアは故意にせよ任意にせよ変更ができなくなります。このオペレーターモードはソフトウェアエンジニアが全設定を調整し、連続してプロダクションを実行するプロダクション環境に推奨されます。

オペレーターモードが有効な間、ファイル、選択したチップ、プログラミングオプション、シリアルナンバー設定、有効なプロセスステップが変更できません。ソフトウェアオプション内ではモード変更もできません。

オペレーターモードは **Prog-Express** のタイトルバーの鍵マークをクリックし、有効・無効の切り替えができます。パスワード保護が有効な場合、ソフトウェアはオペレーターモードを有効・無効にする前にパスワードの入力を求めます。

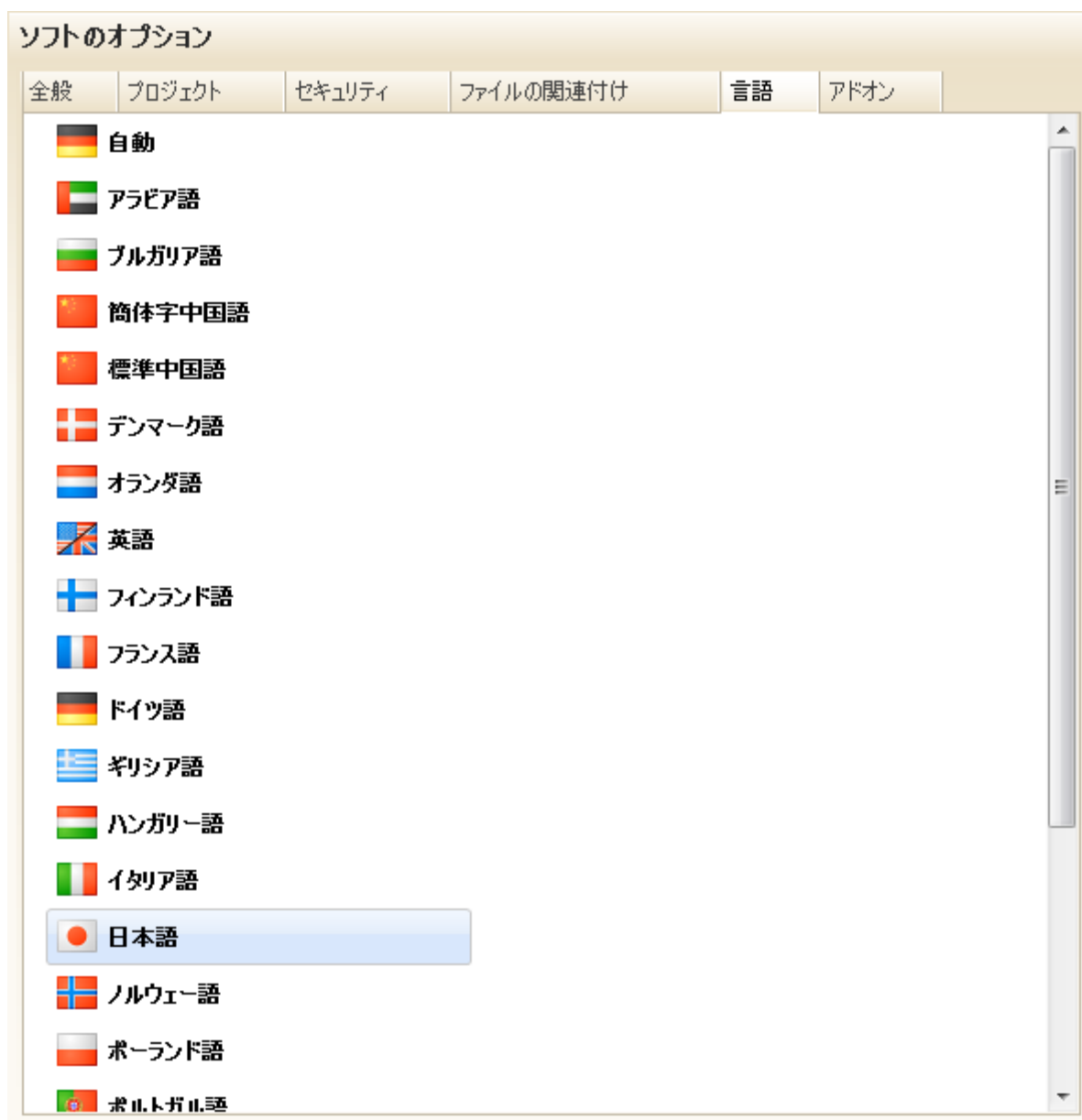
この設定は1つのプロジェクトにもソフトウェア自体にも使用できます。ソフトウェア設定とプロジェクト設定でオペレーターモードが有効な場合、**Prog-Express** は常に制限のある方を利用します。

ファイルの関連付け



この部分ではファイル拡張子を Prog-Express に関連付けすることができます。対象の関連づけするファイルをダブルクリックすると Prog-Express が自動的に起動します。この部分で設定をするには管理者権限が必須です。

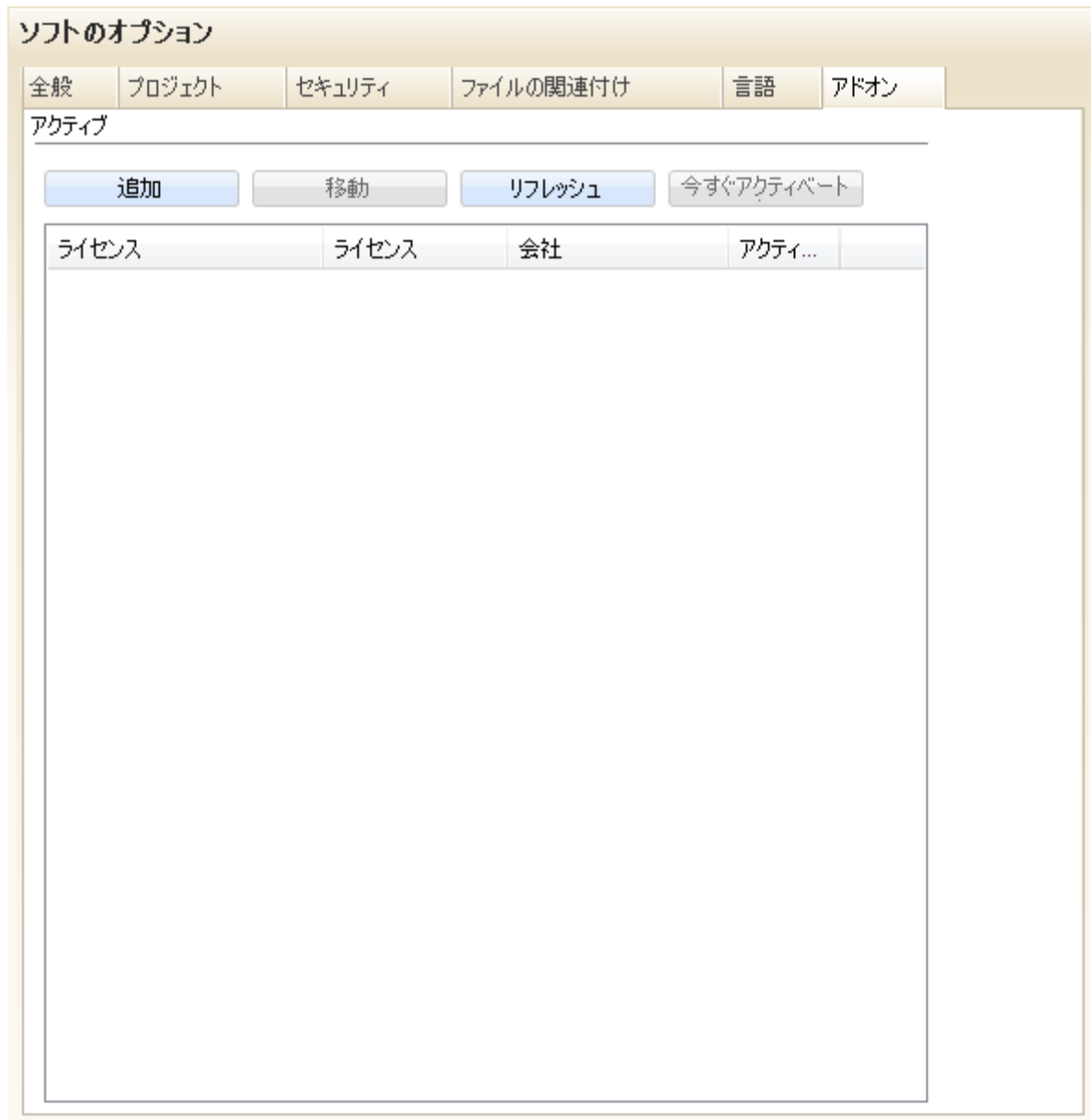
言語設定



必要な言語をクリックするだけで選択できます。適用できる場合は現在サポートしている全言語を選択できるようにウェブサイト www.batronix.com で最新の Prog-Express をダウンロードしてください。

最初のエントリー「自動」はご使用のオペレーションシステムの言語設定を使用します。上記の言語が英語のスペルのアルファベット順に並んでいます。

アドオン



メニューの部分「アドオン」で多くの機能を得るために拡張モジュールを追加または有効にすることができます。

ログファイルテキスト

ログテキスト内に既に終了したあるいは実行中のプロセスが記録されます。

プラス・マイナスマークをマウスでクリックするとノードが拡大したり、縮小したりします。新しいプロセスが開始するとソフトウェアは最近のプロセスノードを縮小します。

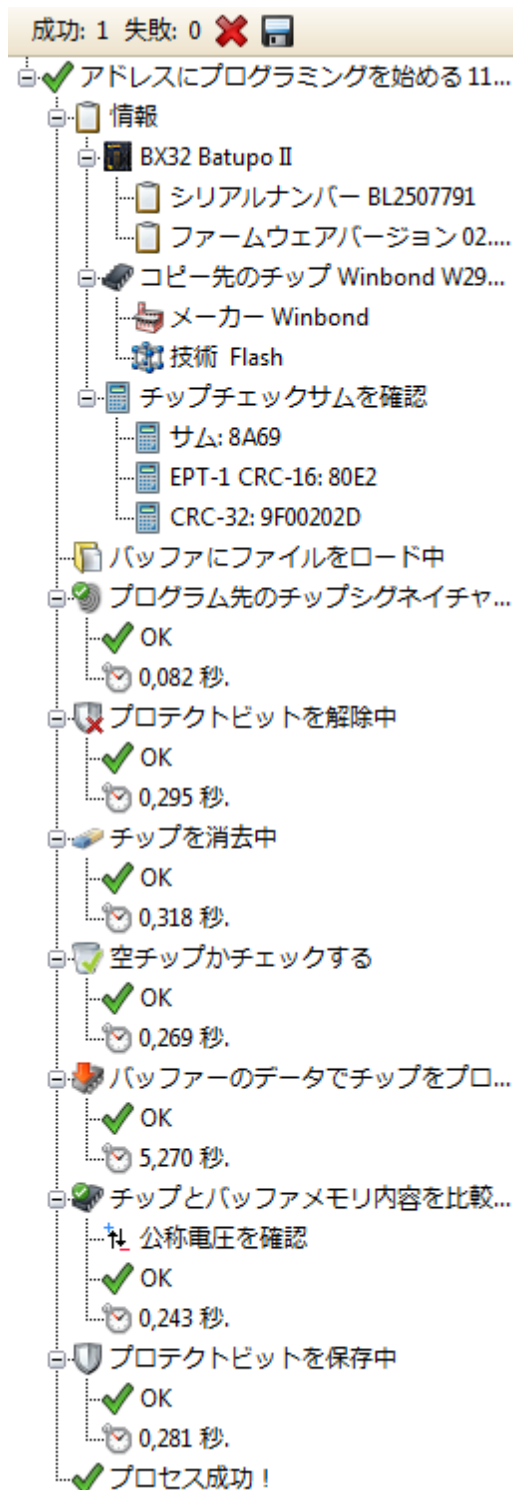
各プロセスには情報ノードとプロセス内で使用される各プロセスステップのノードがあります。情報ノードは使用中のプログラマ、チップ、ファイル、チップチェックサムなどプロセスについて一般情報を含みます。

様々な理由で、ファイルチェックサムはチップチェックサムとは異なる場合があります。チップの一具からプログラミングする場合、ファイルはチップよりも容量が大きく、より多くのデータを含むことができる場合があります。また特別なプログラミング設定やシリアルナンバーを使用することで異なる場合もあります。「チップをバッファと比較する」機能は正確なプログラミングを確実にします。一般的に Prog-Express の「チップをバッファと比較する」機能はファイルチェックサムをチップチェックサムと比較するよりも十分に安全です。なぜなら「チップをバッファと比較する」機能はデータのチェックサムだけでなく、全バイトを完全に比較するからです。

情報ノード内にはメッセージを確認するユーザーの時間などを含んだプロセス所要時間が表示されます。各プロセスステップの所要時間がプロセスステップノードにリストアップされます。

中断したプロセスは赤色の中止サイン、失敗したプロセスは赤い X サイン、完了したプロセスは緑色のチェックマークで示されます。

赤い X マークを一度クリックすると全ログテキストが消去されます。




チップブラウザ


チップブラウザはプログラムの複数の場所から有効にできます。例えば、ソフトウェアモード「チップをプログラミングする」内でコピー先チップを選ぶチップ選択です。

チップブラウザには異なるそれである可能性のあるチップから正しいチップを選ぶのに役立つ様々な機能があります。

チップはツリー構造で整理され、下記のオプションを使ってこの構造の深さを定義付けることができます。：

 技術ごとにチップをグループ分けする。

このオプションが有効になると全てのチップが技術によってグループ分けされ、例えばフラッシュチップと EPROM を簡単に識別できるようになっています。

 製造者ごとにチップをグループ分けする。

このオプションが有効になると全てのチップが製造者ごとにグループ分けされます。

両方のオプションは同時に有効にできます。チップはまず技術でグループ分けされてから製造者ごとにグループ分けされます。

フィルターオプション

検索基準をテキストフィールドに入力し、フィルターオプションを有効にするとチップ名にその基準を含むチップのみが表示されます。

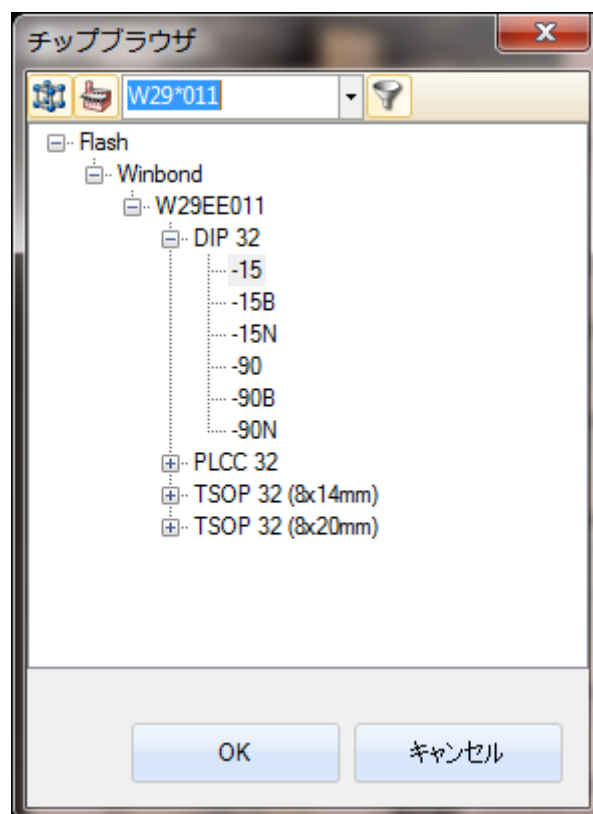
フィルタリングには下記の4つのワイルド・カードを使用することができます。：

% この記号はあらゆるタイプの文字や文字数を示すことができます。例えば「27%512」の返り値は「27512」 / 「27c512」 / 「27SF512」です。

* アスタリスクは%と同じ機能を持ちます。

_ アンダースコアは1文字を置き換えます。例えば「27_512」の返り値は「27C512」や「27E512」で、「27512」または「27SF512」という結果は出ません。

? 疑問符はアンダースコアと同じ機能を持ちます。



チップ自動アイデンティファイ

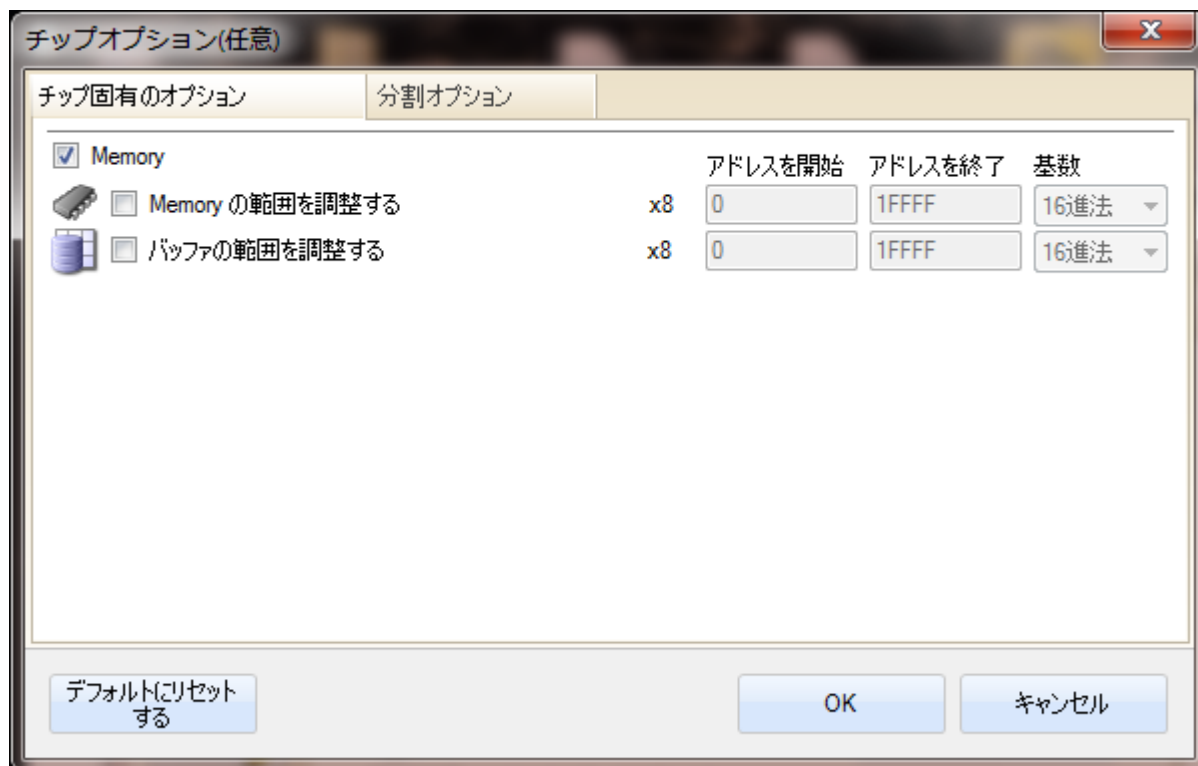
多くのチップは特別な機能で読解できる署名を含んでいます。この署名は通常は製造者と識別子を組み合わせたもので、これによりチップタイプが明確に判明します。署名を含まず、ソフトウェアに自動的に確認されないチップもあります。

この署名を読み込むためには多くのチップはアドレスライン A9 に比較的高い電圧 (12.5V) を必要とすることにご注意ください。この電圧は個別のピンでこの電圧を処理できない他のチップにダメージを与える恐れがあります。

Prog-Express にはプロセス設定により自動 確認機能があります。

チップオプション

チップの個別オプション



メモリ

ある種の IC には 1 つ以上のメモリエリアがあります。例えば様々なマイクロコントローラには個別のプログラムメモリ、データメモリ、コンフィギュレーションメモリがあります。各メモリにはこのタブで個別に調整コントロールができます。

メモリ範囲を調整する

デフォルト設定では全てのメモリ範囲が使用されます。部分的な範囲を特定する場合、「メモリ範囲を調整する」を選択してください。使用する範囲は「開始アドレス」と「終了アドレス」で設定できます。この範囲のメモリアドレスのみがプログラムされ、読み込まれ、確認されます。

バッファ範囲を調整する

通常はデータは最初のファイル（バッファ）アドレスから使用されます。必要な場合、オフセットとして「開始アドレス」を指定できます。オフセット情報はチップのデータ幅に基づきます。16-bit チップ（アドレスにつき 2 バイトのデータ）では、オフセットアドレス 2 を使用すると最初の 4 バイトをスキップします。

特別チップオプション

EEPROM のシリアルアドレスまたはコンフィギュレーション設定のようにチップに特別なオプションがある場合、それらもここで表示されます。

スプリットオプション



3つの異なるスプリット機能があります。:

1. スプリット機能なし: 全データはチップに通常の方法でプログラミングされています。
2. 偶数/奇数のアドレスによりスプリットする: 「偶数」の設定を使うと偶数アドレスを持つ全バッファデータがチップにプログラミングされ、「奇数」を使うと奇数アドレスを持つ全バッファデータがチップにプログラミングされます。
3. 4つのアドレス部分にスプリットする: 4つの各ブロックの内どのアドレスロケーションをプログラミングするかを選択できます。4つの連続するアドレスのうち1、2、3、4番目のアドレスを選択することができます。

シリアルナンバー

チッププロダクションでは大抵チップが独自のシリアルナンバーまたはアドレスを持っている必要があります。このために次の設定が可能です。

シリアルナンバー

☐ シリアルナンバーを入れないで下さい

☐ ファイルからシリアルナンバーをロードする

現在の位置

DEC 1

☒ シリアルナンバーを自動作成する

最初のシリアルナンバー・バイトのチップアド

HEX 0

シリアルナンバーのバイト数

DEC 8

次のシリアルナンバー

DEC 1

ステップの増分量:

HEX 1

エンディアン

ビッグエンディアン (H-L)

基数

10進法

数値形式

テキスト

16進法コード

30 30 30 30 30 30 30 31

アスキー

00000001

OK 中断

シリアルナンバーファイル

「ファイルからシリアルナンバーをロードする」で、シリアルナンバーファイルを指定し、そこからシリアルナンバーを読み込むことができます。各焼き付けプロセス完了後ファイルから次のラインが読み込まれます。

シリアルナンバーファイルは下記のフォーマットを持っている必要があります。:

「#」から始まるラインはコメントラインでスキップされます。シリアルナンバーがあるラインにはコメントを入れることができません。

各値の間にスペースやタブを置くことはできます。

シリアルナンバーラインは次のようにフォーマットしていなければなりません。:ラインは16進数アドレスで開始し、後ろにコロン、コンマで分けられた16進数の値のリストが続きます。

例:

16進数の値は様々なフォーマットで入力できます。

```
#Testdata....
#This comment is allowed
1A0h:1, 2, 3, 4, 5, 6, 7, 8, 9
1A0h:11,12,13,14,15,16,17,18,19          #This comment is not allowed
1A0h:21h, 0x22, 23 , &H24 ,25, 26, 27 , 28 , 29
1A0h:*Line with errors*....
```

4E,10,F2 : 付加なしの単純16進数

4Eh,10h,F2h : 接尾辞としてhが付いた単純16進数

0x4E,0x10,0xF2 : 接頭辞として0xが付いた単純16進数

&h4E, &h10, &hF2 : 接頭辞として&hがついた単純16進数

ASCII データの入力も可能ですが16進数の等価に換算する必要があります。例えば MAC アドレスのリスト (コンピュータネットワークのハードウェアアドレス)は下記になります。:

The Address is: 7F-3C-01-47-FF-04 and should be stored in the chip starting at 05A2h :

```
7 F - 3 C - 0 1 - 4 7 - F F - 0 4
05A2h: 37,46,2D,33,43,2D,30,31,2D,34,37,2D,46,46,2D,30,34
```

各シリアルナンバーはチップに書かれており、その値の長さを規定したアドレスから始まります。

シリアルナンバージェネレーター

このジェネレーターはその設定によってシリアルナンバーを生成します。下記の設定が可能です。:

最初のシリアルナンバーバイトのチップアドレス: シリアルナンバーの最初のバイトはそのアドレスに保存されます。以下のシリアルナンバーのバイトがこのアドレスから順次位置に保存されます。

シリアルナンバーバイトの数: バイトの中のシリアル長の長さです。設定を8にするとシリアルナンバーが8桁になります。

次のシリアルナンバー: 次のプログラムサイクルで使うシリアルナンバーです。各プログラミングの完了後「ステップの増分量」の値から増やすことができます。

ステップの増分量: 各プログラミングの完了後シリアルナンバーはこの値で増えていきます。

エンディアン: エンディアンはシリアルナンバーの **LSB** (最下位バイト) あるいは **MSB** (最上位バイト) が最初の位置「最初のシリアルナンバー・バイトのチップアドレス」に保存されているかどうかを特定できます・

基数: 生成されたシリアルナンバーは10進数システムあるいは16進数システムに基づきます。

数値形式: 生成されたシリアルナンバーは **ASCII** テキストあるいはバイナリ値として作ることができます。

Hex コード / ASCII: これらの2つのフィールドは指定の設定によって次に生成されるシリアルナンバー (次のシリアルナンバー) のプレビューを表示します。

プロジェクト

プロジェクトでは「プログラミング」、「コピーする」、「読み込む」、「シリーズプロダクション」ソフトウェアモードに関連する設定を保存し、呼び出すことができます。

各ソフトウェアモードのチップ選択、ファイル選択、チップオプション、シリアルナンバー設定、コピー数、有効/無効のプロセスステップが保存されます。さらにプロダクションモードでは、チップ、ファイル、全ての接続しているプログラミングデバイスに規定されたチップオプションが保存されます。

プロジェクトファイルは「**pep**」拡張子を使用します。これは「**Prog-Express Project**」の略称です。ファイルは標準テキストファイルとして保存され、標準テキストエディタでも開くことができます。自動化と遠隔操作のためにプロジェクトファイルは別のアプリケーションで作成し、**Prog-Express** に読み込むこともできます。

保存する



保存ボタンはファイルブラウザを開きます。ここでプロジェクトファイルの場所とファイル名を入力し、保存することができます。そのファイルが既に存在する場合、上書きされるか、プロセスが中断されます。

全てのダイアログと全てのソフトウェアモードの設定が保存されます。

読み込む



読み込み中は、ファイルブラウザで使用するプロジェクトファイルを選択すると、全ての保存された設定が復元されます。


注意!プロジェクトを読み込む際に以前の設定は全て消去され、バッファのみが維持されます。

その他


ソフトウェアの右上に表示されているのは追加的な機能です。:「オペレーターモード」、「ファームウェアアップデート」、「校正する」、「Prog-Express について」、「Prog-Express ヘルプ」があります。




オペレーターモード

 誤った使用や見直しに対する設定を保存することができます。


ファームウェアアップデート

 この機能で最新のファームウェアがプログラマに転送されます。


校正する

 ヘルプと校正ツールでプログラマはテストされその電圧制御を校正することができます。この機能は全てのプログラマデバイスにサポートされている訳ではありません。

Prog-Express について

 このソフトウェアについての情報ウィンドウを表示します。

Prog-Express ヘルプ

 ソフトウェアマニュアルを開きます。

Prog-Express の遠隔操作

特別なアプリケーションで外部プログラムを使って Prog-Express ソフトウェアを遠隔操作できます。例えばデバイス規定の調整値や測定値の自動プログラミングをする場合などに使用できます。

コマンドはコントロールファイルまたはコマンドラインパラメータを使用して実行しているときに Prog-Express ソフトウェアに送ることができます。

コマンドラインパラメータを使った遠隔操作

1 つ以上のコマンドはコマンドを使って送ることができます。全てのコマンドはスラッシュで始まります。ある種のコマンドは読み込むファイル名などのパラメータ値が必要です。パラメータはスペースで分けることができ、ファイルパスは引用符で囲む必要があります。

例:

```
Prog-Express.exe /open „C:\Directory\FileName.bin”
```

```
Prog-Express.exe /mode program /run 2 /exit
```

Prog-Express アプリケーションの一例です。つまり毎回新しいソフトウェアのインスタンスを開始することなく、ソフトウェアを実行中にコマンドラインパラメータを使って新しいコマンドをソフトウェアに送ることができます。

スクリプトファイルを使った遠隔操作

Prog-Express がコマンドラインパラメータ“/remotefile filename”で開始するとそのファイルの全コマンドが実行されます。“ポーリング”コマンドを使うとファイルの変更を継続的にチェックします。ファイルが変更され次第ファイルの全コマンドが実行されます。

スクリプトファイルはセミコロンでマークしたコメントを含むことができます。

例:

```
; Sample script  
open "C:\Directory\FileName.bin"  
mode program  
run 2  
exit
```

リモートファイルコマンド

「リモートファイル」コマンドはファイルからコマンドを読み込み、実行する際に使用できます。

例:

- 例: `remotefile „C:\Directory\Remote.txt“` は指定したファイルからコマンドを読み込み、実行します。

ポーリング機能をオン・オフにするコマンド

「ポーリング・オン」コマンドを実行すると、「リモートファイル」で指定したコマンドの変更を継続的にチェックします。外部プログラムによってファイルが変更され次第ファイルに含まれるコマンドは Prog-Express によって実行されます。

「ポーリング・オフ」コマンドを使用すると測定がオフになります。「ポーリング・オン」コマンドにはミリ秒で特定されるファイルの変更を区間ごとにチェックする追加的なパラメータがあります。

例:

- `Poll on` 遠隔操作ファイルの測定をオンにします。
- `Poll off` 遠隔操作ファイルの測定をオフにします。
- `Poll on 2000` 遠隔操作ファイルの測定をオンにします。ファイルは2秒（2000 ミリ秒）ごとに変更をチェックされます。

開くコマンド

「開く」コマンドでプロジェクトファイル(.pep)またはデータファイル(.bin, .hex, .mhx,...)を開くことができます。「.pep」拡張子をもたない全ファイルは Hex エディタに読み込まれます。全 .pep ファイルはプロジェクトファイルとして認識され、開かれます。例:

- `open “C:\Directory\FileName.bin”` Hex エディタに指定のファイルを読み込みます。
- `open “C:\Directory\ProjectSettings.pep”` 保存された Prog-Express 設定とファイルを読み込みます。

モードコマンド

ソフトウェアモードは「モード」コマンドで設定できます。有効なパラメータは「プログラミングする」、「コピーする」、「読み込む」、「プロダクション」、「Hex エディタ」、「オプション」です。例: “Prog-Express.exe /mode program” はソフトウェアをプログラムモードにします。例:

- `mode program` ソフトウェアを「プログラミングモード」にします。
- `mode production` ソフトウェアを「シリーズプロダクションモード」にします。

ファイル選択コマンド

「ファイル選択」コマンドは「チップをプログラミングする」、「プロダクション」モード、または「チップを読み込む」モードのデータターゲットファイルのデータコピー元のファイルを設定します。例:

- `selectfile "C:\Directory\FileName.bin"` 現在選択している「プログラミングする」、「読み込む」「プロダクション」モードで特定のファイルをデータコピー元/コピー先として設定します。

プロセスステップコマンド

「プロセスステップ」コマンドで、各プロセスステップのオン・オフの切り替えとパラメータのオン・オフの切り替えをすることができます。

例:

- `processstep 3 on` 現在のソフトウェアモードのプロセスステップ3をオンにします。
- `processstep 5 off` 現在のソフトウェアモードのプロセスステップ5をオフにします。

変異チップ選択コマンド

「変異チップ選択」コマンドで現在のモードでチップを設定することができます。

固有の「変異チップ ID」をパラメータとして指定する必要があります。「チップリスト保存」コマンドで作成できるチップリストの中で正しい変異チップを見つけることができます。(下記参照)

例:

- `selectchipvariant 697` 現在のモードでチップを「変異チップ ID」697 で設定します。

自動識別コマンド

この「自動識別」コマンドでチップは自動的に識別されます。

ファイル名は追加のパラメータとして指定できます。これにより、Prog-Express は変異チップ ID とチップ名をこのファイルに保存します。

例:

- `autoidentify` Prog-Express の有効なモードでチップ自動識別機能を実行します。
- `autoidentify "C:\Directory\chip.txt"` 変異チップ ID とチップ名を指定のファイルに保存します。

起動コマンド

有効なソフトウェアモードのプロセスは「起動」コマンドで開始できます。追加のパラメータとしてプロセスサイクルの数を指定できます。

「起動」コマンドの後に他のコマンドが送信されると、このコマンドはプロセスが完了してから実行されます。このような方法で複数のプロセスを次から次へと実行することができます。

例:

- `run` プロセスを開始します。
- `run 10` 10個のコピーのプロセスを開始します。

ログ保存コマンド

「ログ保存」コマンドを使ってログスクリーンの内容をファイルに保存することができます。

ファイル名は追加のパラメータと指定できます。これが指定されない場合、Prog-Express はログデータをデスクトップの「log.txt」というファイルに保存します。

例:

- `savelog` ファイル log.txt.にログデータを保存します。
- `savelog "C:\Directory\mylog.txt"` ログデータを指定のファイルに保存します。

デバイス保存コマンド

全ての接続された Batronix USB デバイスの情報が「デバイス保存」コマンドでファイルに保存できます。

このコマンドはデバイスナンバー、名前、シリアルナンバー、ファームウェアバージョン、実際のジョブ、進行パーセンテージ、接続されたデバイスの情報を保存します。

追加のパラメータとしてファイル名を指定できます。これが指定されない場合 Prog-Express はデバイスデータをデスクトップのファイル「deviceinfo.txt」に保存します。

例:

- `savedeviceinfo` デバイスデータを deviceinfo.txt ファイルに保存します。
- `savedeviceinfo "C:\Directory\devices.txt"` デバイスデータを指定のファイルに保存します。

ログ消去コマンド

ログスクリーンの内容は「ログ消去」コマンドで消去することができます。

チップリスト保存コマンド

「チップリスト保存」コマンドはプログラマがサポートする全てのチップが載ったテキストファイルを作成します。最初のパラメータとしてプログラマの省略名(BX32、BX32P、BX40、BX48)を指定する必要があります。2番目のパラメータではプログラマのハードウェアバージョンを指定する必要があります。(1、2)

3番目のパラメータとしてファイル名を指定できます。これが指定されない場合 Prog-Express はチップリストをデスクトップの「chiplist.csv」ファイルに保存します。

Examples:

- `savechiplist BX48 2` 「BX48 Batego II」のチップリストをデスクトップの「chiplist.csv」に保存します。
- `savechiplist BX48 2 „C:\Directory\chip.csv“` チップリストを特定のファイル名とディレクトリで保存します。

追加のコマンド

- `hide` Prog-Express ソフトウェアを隠します。
- `show` 「hide」コマンドの後、Prog-Express ソフトウェアを表示します。
- `exit` Prog-Express を終了します。

サンプルアプリケーション: 特定のデータのプログラミング

ケーススタディ: プロダクション環境では特別なソフトウェアは特定のデバイスから測定データを記録するので、対応する調整値をメモリチップにプログラミングする必要があります。

まずチップオプションなどの使用する特別な設定、シリアルナンバー、プログラミングするチップとファイルが通常の Prog-Express 作業で設定され、.pep ファイルとして保存されます。

それから特別なソフトウェアがコマンドライン呼び出しで Prog-Express を起動します。:

`Prog-Express.exe /remotefile "C:\Directory\Remote.txt" /poll on`

これは Prog-Express を開始し、remote.txt ファイルの変更を測定します。特別なソフトウェアはまずデバイスをテストし、調整データを adjustments.bin ファイルに保存し、下記内容とともに remote.txt file を保存します。:

```
;Sample application
open "C:\Directory\Project.pep"      ;Loads the program settings
mode program                        ;Switches to "Programming" mode
run                                  ;Starts the process
savelog                             ;Saves the log data to the file log.txt
```

そして特別なソフトウェアは log.txt ファイルを測定し、評価し、次のデバイスに続きます。