





BeeHive204AP

非常に高速なユニバーサル 4x48 ピンドライブ・コンカレント・マルチプログラマ-ー 自動プログラマーのコア



BeeProg2AP

非常に高速なユニバーサル4x48ピンドライブ・プログラマ
--- 自動プログラマーのコア

この文書はELNEC s.r.o., Presov, Slovakiaに著作権があります。全著作権所有。本書、又は、その一部は ELNEC s.r.oの書面による事前の許可なくいかなる形またはいかなる形でも複製、複製、翻訳することはできま せん。

制御プログラムはELNEC s.r.o., Presov, Slovakiaの著作権です。制御プログラム、又は、その一部はあらゆ る目的のためにも、いかなる形式でも、いかなる媒体上でも、分析、解体、又は、変更することはできません。

このマニュアルに記載されている情報はリリース時点の正確さを期していますが、全ての製品を継続的に改善しています。www.elnec.comのマニュアルを参照してください。

ELNEC s.r.o. とその代理店はこのマニュアルの誤読、誤用には一切責任を負いません。

E ELNEC s.r.o. は本書に記載されている製品を予告なく変更または改善する権利を留保します。 このマニュアルには企業、ソフトウェア製品などの名称が含まれており、それぞれの所有者の商標である可能性が あります。 ELNEC s.r.o. これらの商標を尊重します。.

COPYRIGHT © 1991 - 2015

Elnec s.r.o. Presov, Slovakia

22nd May 2015

このマニュアルの使い方

このマニュアルでは制御プログラムのインストール方法とプログラマの使用方法について説明しています。 ユーザーにはPCやソフトウェアのインストールに関する経験があることが前提です。コントロール・プログラ ムをインストールしたら印刷されたユーザー・マニュアルではなく、コントロール・プログラム内の状況依存へ ルプを参照することをお勧めします。リビジョンは印刷されたユーザー・マニュアルの前に状況依存へルプ に実装されています。

お買い上げのユーザー様

Elnecプログラマをお買い上げ頂きましてありがとうございます。

現在のバージョンが最新でない場合は、Elnec WEBサイト(www.elnec.com)のサポート/ダウ ンロード・セクションからマニュアルの最新のバージ ョンをダウンロードして下さい。

目次

このマニュアルの使い方	
イントロダクション	7
製品構成	9
必要な PC スペック	10
フリー・アディショナル・サービス:	
BeeHive204AP	12
イントロダクション	13
BeeHive204AP 構成	15
BeeHive204APをPCに接続	16
セルフテストとカリブレーション・チェック	16
テクニカル・スペシフィケーション	
BeeProg2AP	19
イントロダクション	20
BeeProg2AP 構成部分	22
BeeProg2APをPCに接続	23
セルフテストとカリブレーション・チェック	23
BeeProg2AP によるマルチ・プログラミング	25
テクニカル・スペシフィケーション	
セットアップ	
ソフトウエア・セットアップ	
ハードウエア・セットアップ	32
PG4UW	45
PG4UW プログラマ・ソフトウエア	
File[ファイル]	49
Buffer[バッファ]	56
Device[デバイス]	63
Programmer[プログラマ]	93
Options[オプション]	
Help[ヘルプ(英文)]	109
PG4UWMC マルチ制御ソフトウェア	112
Common notes 共通ノート	130
Maintenance[メンテナンス]	131
Software[ソフトウェア]	133
Hardware[ハードウエア]	139
ISP (In-System Programming)[ISP (イン-システム・プログラミング)]	139
その他	142
トラブル・シューティングと保証	143

Troubleshooting[トラブルシューティング]	144
サポートされていないターゲット・デバイスがある場合	144
Warranty terms[保証期間]	145



このマニュアルで使用されている表現

コントロール・プログラムのファンクションは Load, File, Deviceなどの太字で表示されています。<F1>,の様に< >内はコントロール・キーです。

このマニュアルで使用されている用語:

Device - デバイス プログラマブル IC 又は、プログラマブル・デバイス ZIF socket - ZIF ソケット ターグット・デバイスを装着するために使われる ZIF [Zero Insertion Force] ソケット Buffer - バッファ コントロール・ソフトウエアにロードされたデータが PC のメモリー。リードされたデータや読み込みデータがパッファ・メモ リーに置かれます。ファイルの保存はバッファ内のデータが保存されます。 USB ポート - USB プリンター・ポート PC の USB ポートに接続して使用されます。 HEX data format - ヘキサ・データ形式 標準のテキスト・ビューワーで読むことができるデータ・ファイルの形式の1つです。 バイト 5AHは '5' と 'A'のキャラクターとしてストアされます。それは、バイト 35Hと 41H を意味します。 この HEX ファイルの1つの行(1レコード)が開始アドレス、データ・バイトとチェックサムで安全に保たれるすべてのレ

コードを含んでいます。全ての記録はチェックサムで保護されています。







このユーザーズ・マニュアルは次のElnecプログラマをカバーしています: BeeHive204AP と BeeProg2AP.

BeeHive204APは自動プログラマ、及び、自動テスト機器(ATE)のコアです。大量生産プログラミング向け に設計された非常に高速なユニバーサル4x48ピン・ドライブ・コンカレント・マルチ・プログラミング・システムです。 BeeHive204APはBeeHive204の工業用バージョンです。チップは理論上の最大プログラミング速度に近 い速度でプログラムされます。プログラマはビルトインISPコネクタを使用してISP対応チップをインサーキットで プログラミングできます。

BeeProg2APは、自動プログラマおよび自動テスト機器(ATE)のコアでもあります。少量生産プログラミン グ用に設計された48個の強力なピン・ドライバーを備えた非常に高速なユニバーサル・プログラマです。 BeeProg2APはBeeProg2の工業用バージョンです。プログラマはビルトインISPコネクタを使用してISP対応チップをインサーキットでプログラムできます。

保護回路、オリジナル・ブランドのコンポーネント、慎重な製造等の高度な設計によりプログラマの部品と作業の BeeHive204APとBeeProg2APに3年間の保証を提供できます(プログラミング・モジュール・インターフェイス・コ ネクターへのプログラミング・モジュールの挿入は500に制限されます)。この保証条件はElnec社からプログラマを直 接購入するお客様に有効です。Elnecの販売者の保証条件は対象国の法律システム、又は、Elnecの販売者 の保証ポリシーによって異なる場合があります。



製品構成

	BeeHive204AP	BeeProg2AP
programmer	•	•
USB cable	٠	•
ISP cable	4x	•
power cordset	•	•
external power supply	•	•
AP1 PMI selftest pod	1x	•
AP1 ISP connector selftest pod	1x	•
AP1 calibration test pod	1x	
programming module fixating screw	8x	2x
screw with washers for ground connection	•	•
tie and tie mount for fixating cables	2x	٠
detachable part for fastening to bottom plate	2x	
pressure plate		
adjustable bar 1 (57,2mm)		
adjustable bar 2 (74,2mm)		
hex key Nr 2,5		
CD with software and user manual	•	•
Quick Guide	•	•
leaflet Notes about ESD	•	•
sticker register your programmer	•	•
sticker Elnec programmer inside	2x	2x
transport case	•	•

使用する前に、パッケージに次の全ての部分が含まれていることを慎重に確認して下さい。それぞれのパ ーツリストとの間に矛盾がある場合、又は、これらのアイテムが破損している場合は直ちに販売店にご連 絡下さい。



必要な PC スペック

最低必要なPCスペック

	OS - Windows	CPU	RAM [MB]	free disk space [MB]	USB 2.0 high speed	USB 1.1
2x BeeHive204AP	XP	C2D 2,6GHz	1000	1000	•	-
BeeHive204AP	2000	P4	512	400	•	-
BeeProg2AP	2000	P4	512	400	-	•

1024 x 768は最低必要なモニタ解像度

Recommended PC requirements

	OS - Windows	CPU	RAM [MB]	free disk space [MB]	USB 2.0 high speed	2x USB 2.0 high speed controllers
2x BeeHive204AP	Win 7	C2Quad	2000	2000		•
BeeHive204AP	Win 7	C2D	1000	1000	•	
BeeProg2AP	Win 7	C2D	1000	1000	•	

1024 x 768より高いモニタ解像度をお薦めします。.

上記は S/W バージョン 3.38(2/2018) 以降のご使用を前提。最新のバージョンは https://www.elnec.com/sw/pg4uwarc-ondemand.exeをご覧下さい。

もし、2つのプログラマを1台のPCで使用頂くためには、各プログラマを別々のUSB2.0 High speed controller(USB EHCI)に接続してご使用されることを強くお薦めします。更に詳しくは"ハードウエア・セット アップ"の章をご覧下さい。

ディスクの空き容量の要件は使用されているICデバイスのサイズと接続されているプログラミング・サイトの数 によっても異なります。大きなデバイスの場合、ディスクに必要な空き容量は約1000MB+2xデバイス・サイ ズxそのPCに接続されたプログラミング・サイトの数になります。

PCハードウェア/ソフトウェア構成がPG4UW/PG4UWMCの現在のソフトウェア・バージョンと現在の状況に十分適しているかどうかを簡単にチェックします:

Windowsタスクマネージャを実行する(Ctrl+Alt+Del)で・[パフォーマンス]タブを参照して下さい。 最大値を表示する必要があります。プログラミング・システムのフル生産時のCPU使用率の80%



フリー・アディショナル・サービス:

何故、制御プログラムの最新バージョンを使用することが重要ですか?

- 半導体メーカーは製品設計、及び、製造における柔軟性、品質、スピードの必要性をサポートするために新しいテクノロジで製造された新しいパッケージ・タイプの新しいデバイスを継続的に導入しています。ペースを維持し最新の状態に保つため通常は1年以内に7000個以上の新しいデバイスを制御プログラムに実装しています。
- ・更に、典型的なプログラマブル・デバイスはその技術的特性、及び、プロセス歩留まりを維持、又は、 向上させるために寿命期間中にいくつかの変更を受けます。これらの変更はアップグレードする必要 があるプログラミング・アルゴリズムに影響することが良くよくあります(プログラミング・アルゴリズムはプロ グラマに特定のターゲット・デバイスにデータをプログラムする方法を指示する命令セットです)。プログラ ミング・プロセスで最新のアルゴリズムを使用することは高品質の結果を得るための鍵です。多くの場 合、古いアルゴリズムではまだデバイスがプログラムされますが、最適なアルゴリズムで可能なレベルの データ保持は提供されません。最新のアルゴリズムを使用しないとプログラミングの歩留まりが低下し プログラミング時間が長くなることがあります。また、プログラムされたデバイスの長期信頼性に影響を 及ぼすこともあります。
- 我々も間違いは起こしますが、フィードバック等により敏速にバグフィックスを行っております。

当社のコミットメントはこの新しいデバイス用に作成された最新、及び/又は、最適なプログラミング・アルゴリ ズムを使用しているかどうかを確認できるように、これらの新規、又は、変更されたデバイスのサポートをリリ ース前、又は、出来るだけ早く実施することです。

·フリー・テクニカル・サポート(e-mail) ·ウェブ・サイト経由フリー・ライフタイム・ソフトウエア・アップデート

> フリー・ソフトウエア・アップデートはElnec社の https://www.elnec.com/en/download から利用 出来ます。

サポート・プログラムでは次の新しいサービスを提供しています :: Keep-Current と AlgOR.

- Keep-CurrentはElnecがプログラマ向けの最新のバージョンの制御プログラムと更新されたデバイ ス情報を提供するサービスです。Keep-Currentサービスは最新のソフトウェアとドキュメントに最小 限のコストでいつでもアクセスできる、手間のかからない保証です。詳細については www.elnec.comを参照して下さい。
- AlgOR (Algorithm On Request)サービスではElnecソフトウェアから現在のデバイス・リストで未だ 使用できないプログラミング・デバイスのサポートを受けることができます。詳細については www.elnec.comを参照して下さい。



BeeHive204AP





イントロダクション

BeeHive204APは自動プログラマ、及び、自動テスト機器(ATE)のコアです。大量生産プログラ ミング向けに設計された非常に高速なユニバーサル4x48ピン・ドライブ・コンカレント・マルチ・プログ ラミング・システムです。

BeeHive204APは4つのBeeProg2ベースの独立したプログラミング・モジュールで構成されています。チップは理論上の最大プログラミング速度に近い速度でプログラムされます。

BeeHive204APは自動化されたプログラマで使用するためのBeeHive204プログラマの工業用バージョンです。違いは次のとおりです:

- BeeHive204APプログラマの寸法はBeeHive204と比較して、ハンドラーの腕の動きのオーバーヘッドを最小限に抑えるために縮小されています。
- 操作中の振動に強い、より機械的に安定したケース。BeeHive204APのケースはプログラマ本体の上部、又は、下部からプログラマの自動作業場所に固定する準備ができています。
- プログラミング・モジュールの異なった構造、機械的アームによるチップの挿入/交換に十分安定であり、モジュールの交換後もZIFソケットの同じ位置を維持できるようになっています。

2台の**BeeHive204AP**ユニットを1台のコントロールPCに接続して、プログラミング作業場をより有効に 活用できます。 Elnecは、Windows XP Embedded駆動の**BeeHive204AP**コントロール・ユニットを 提供しています。 これは、2台のBeeHive204APプログラマに対応できます。

BeeHive204APは、次の2つの方法で、自動化されたプログラマ(廃止されたプログラマの代わりとして)、又は、ハンドラーに実装できます:

- 標準のPCを使用して、例えば、BeeHive204APを自動プログラマの制御PCに接続できます(USB ハブ、又は、PCのUSBポートを使用して1台のコンピュータに最大2台のBeeHive204APを接続でき ます)。
- 2. BeeHive204APコントロール・ユニットを使用(オプション)
 - 最大2台のBeeHive204APプログラマがBeeHive204APコントロール・ユニットによって制御されます。
 BeeHive204APコントロール・ユニットは産業環境向けに最適化されたWindows XP Embedded
 駆動のコンピュータです。
 - システム内の1台のBeeHive204APコントロール・ユニットがマスターユニットとして機能します。ここでは、 マルチプログラミング制御ソフトウェア、シリアル化エンジン、ホストシステムへのインターフェースも実行しています。
 - BeeHive204APコントロール・ユニットのインターフェースは標準LANおよび外部LANスイッチを介して 行われます。BeeHive204APマスターコントロールユニットとホストシステムのインターフェースはLAN、 又は、RS232(要求に応じて他のインターフェース)経由です。

BeeHive204APを利用可能なサードパーティの自動化されたプログラマ、及び、ハンドラーに実装するに はPG4UWMC制御ソフトウェアのシンプルなリモート・コントロールを使用します。標準プログラミング言語の 実装例がありますが、もちろんこのタスクでお客様をサポートする準備ができています。

ノート:BeeHive204APプログラマの他の標準パラメータについてはBeeHive204の説明を参照してください。

BeeHive204APプログラミング・モジュールにはBeeHive204プログラマ用のモジュールと同じ回路図 がありますが、これらのモジュールはBeeHive204APプログラマの最上部で完全に安定するように機械 的に設計されており、プログラミング・モジュールが交換されても同一の位置を維持することを意図して います。



PLCC、SOIC、PSOP、SSOP、TSOP、TSSOP、TQFP、QFN(MLF)、BGA、及び、他のパッケージには デバイス用の利用可能なプログラミング・モジュールがあります。



ノート:プログラミング・モジュールにはプログラミング・モジュールの左上隅を指す参照ピン(角)があります。必要に応じてプログラミング・モジュールでZIFソケットの他の向きの注文も受け付けます。状況については営業部門にご相談ください。

BeeHive204APプログラマはWindows XPからWindows 10 64ビット迄の全てのバージョンのMS Windowsで動作する快適で使いやすい制御プログラムによって駆動されます。

殆どの場合、BeeHive204APは真にユニバーサルなプログラマであるため、新しいデバイスではプログラミング・モジ ュールとソフトウェアの更新のみが必要です。お客様のニーズに対する当社独自の迅速な対応はOnDemandソ フトウェアによりリクエストから1日以内にソフトウェアの更新を準備できます。

保護回路、オリジナル・ブランドのコンポーネント、慎重な製造を含む高度な設計によりBeeHive204APの 部品と作業に3年間の保証を提供できます(プログラミング・モジュール・インターフェイス・コネクタへのプログラミ ング・モジュールの挿入は500に制限されます)。



BeeHive204AP 構成

- 1. プログラマ・モジュール・インターフェース(PMI)コネクタ
- 2. 動作結果 LEDs
- 3. サイトの電源/スリープ LED
- 4. ISPコネクタ (20ピン・コネクタ 2-1634689-0 TE connectivity社製)
- 5. BeeHive204APを底板に固定するための3つのØ4,5mm穴を備えた取り外し可能なパーツ
- 6. BeeHive204APを上部プレートに固定するためのM4ナット
- 7. プログラミング・モジュール固定ネジ
- 8. 右側電源コネクタ
- 9. 右側PC用 type B USBコネクタ ↔ BeeHive204AP通信ケーブル
- 10. USBケーブルを固定するための右側タイマウント



BeeHive204APの右トップ・ビュー



- 11.後面電源コネクタ
- 12. 後面PC用 type B USBコネクタ ↔ BeeHive204AP 通信ケーブル
- 13. USBケーブル固定のための後面タイマウント
- 14. 温度制御ファン
- 15. "GND" ネジはプログラマの接地に使用することが出来ます



Rear view to BeeHive204AP

BeeHive204AP をPC に接続

プログラマをPCに接続するための推奨事項: 1.プログラマとPC、又は、その他のアースをアース接続します 2.USBケーブルを介してプログラマをPCに接続します 3.電源をプログラマに接続します

セルフテストとカリブレーション・チェック

プログラマが期待通りに動作しないと感じる場合は、標準パッケージに同梱されているAP1 PMIセルフテス トPOD(AP1 ISPコネクタ

・セルフテストPOD)を使用してプログラマ(ISPコネクタ)の"Selftest"を実行してください。

プログラマのセルフテスト

- プログラマ・サイトのプログラミング・モジュール・インタフェース(PMI)コネクタにAP1 PMIセルフテストPODを 挿入します
- プログラマのセルフテストをPG4UW(メニュー Programmer/Selftest plus)で実行します





AP1 PMI セルフテスト pod

ISPコネクタのセルフテス

- AP1 ISPコネクタ・セルフテストPODをプログラマのプログラマ・モジュール・インターフェイス(PMI)コネクタに挿入します
- AP1 ISP コネクタのセルフテスト POD の 20 ピン・コネクタとプログラマの ISP コネクタを出荷時のプログラマ・パッケージに含まれている ISP ケーブルで相互接続します。 ピンが正しく相互接続されていることを確認してください(つまり、1-1、2-2 ... 20-20).
- PG4UWでISPコネクタのセルフテストを実行します(プログラマ/セルフテストISPコネクタ).



AP1 ISP コネクタ・セルフテスト pod



カリブレーション・テスト

- AP1カリブレーション・テストPODをプログラマのプログラマ・モジュール・インターフェイス(PMI)コネクタに挿入します。
- PG4UW(メニュー Programmer/Calibration test)でプログラマのカリブレーション・テストを実行します。



AP1 カリブレーション・テストpod

テクニカル・スペシフィケーション

一般

- 外部電源ユニット:動作電圧AC 100-240V定格、最大90-264 VAC、47-63 Hz、DC出力15V、
 4.3A、出力ケーブル長1200mm(47.2 inch)
- 電力消費最大 60W 動作時
- ・ 寸法: 310x205x61 mm (12.2 x 8.1 x 2.4 inch). プログラミング・モジュールが挿入されていない場合の寸法を測定しました。プログラミング・モジュールが挿 入されたBeeHive204APプログラマの合計の高さはZIFソケットの高さに依存し、80~90mmの間で変 化する可能性があります。
- 重さ (プログラマ) 3.5kg (7.7 lb)
- •操作湿度 5°C ÷ 40°C (41°F ÷ 104°F)
- 操作湿度 20%..80%, 非露結



BeeProg2AP





イントロダクション

BeeProg2APは自動プログラマ、及び、自動テスト機器(ATE)のコアです。少量生産プログラミン グ用に設計された48のパワフルなピンドライバーを備えた非常に高速なユニバーサル・プログラマです。 プログラマはビルトインISPコネクタを使用してISP対応チップをインサーキットでプログラムできます。

BeeProg2APは自動化されたプログラマで使用するためのBeeProg2プログラマの工業用バージョ ンです。違いは次のとおりです:

- BeeProg2APプログラマの寸法はBeeProg2と比較して、ATEマシーンに実装するために縮小されています。
- 操作中の振動に強い、より機械的に安定したケース。BeeProg2APのケースはプログラマ本体の上部、又は、下部からプログラマの自動作業場所に固定する準備ができています。ISPプログラミングの場合、BeeProg2APの場合のみを左側から固定することもできます。
- プログラミング・モジュールの異った構造、機械的アームによるチップの挿入/交換に十分安定であり、モジュールの交換後も ZIF ソケットの同じ位置を維持できるようになっています。

BeeProg2APは標準のPCを使用してBeeProg2プログラマと同じようにISPプログラマとして自動プログラマ、又は、ATEマシンに実装できます。BeeProg2APは自動プログラマの制御PCにも接続できます。USBハブ、又は、PCのUSBポートを使用して最大8台のBeeProg2APを1台のコンピュータに接続できます。

BeeProg2APを利用可能なサードパーティの自動化されたプログラマ、及び、ハンドラーに実装するには PG4UWMC制御ソフトウェアのシンプルなリモート・コントロールを使用します。標準プログラミング言語の実 装例がありますが、もちろんこのタスクでお客様をサポートする準備ができています。

ノート:BeeProg2APプログラマの他の標準パラメーターについてはBeeProg2の説明を参照してください。

BeeProg2APプログラミング・モジュールにはBeeProg2Pプログラマ用のモジュールと同じ回路図があ りますが、これらのモジュールはBeeProg2APプログラマの最上部で完全に安定するように機械的に 設計されており、プログラミング・モジュールが交換されても同一の位置を維持することを意図していま す。

PLCC、SOIC、PSOP、SSOP、TSOP、TSSOP、TQFP、QFN(MLF)、BGA、及び、他のパッケー ジにはデバイス用の利用可能なプログラミング・モジュールがあります。





ノート:プログラミング・モジュールにはプログラミング・モジュールの左上隅を指す参照ピン(角)があります。必要に応じてプログラミング・モジュールでZIFソケットの他の向きの注文も受け付けます。状況については営業部門にご相談ください。

BeeProg2APプログラマはWindows XPからWindows 10 64ビット迄の全てのバージョンのMS Windows で動作する快適で使いやすい制御プログラムによって駆動されます。

殆どの場合、BeeProg2APは真にユニバーサルなプログラマであるため、新しいデバイスではプログラミング・モジュ ールとソフトウェアの更新のみが必要です。お客様のニーズに対する当社独自の迅速な対応はOnDemandソフ トウェアによりリクエストから1日以内にソフトウェアの更新を準備できます。

保護回路、オリジナル・ブランドのコンポーネント、慎重な製造を含む高度な設計によりBeeProg2APの部 品と作業に3年間の保証を提供できます(プログラミング・モジュール・インターフェイス・コネクタへのプログラミン グ・モジュールの挿入は500に制限されます)。



BeeProg2AP 構成部分

- 1. プログラマ・モジュール・インターフェース(PMI) コネクタ
- 2. 動作結果 LEDs
- 3. サイトの源/スリープLED
- 4. ISPコネクタ (20ピン・コネクタ 2-1634689-0 TE connectivity社製)
- 5. BeeHive204APを上部プレートに固定するためのM4ナット
- 6. プログラミング・モジュール固定ネジ



BeeProg2APの右トップ・ビュー



- 7. 電源供給コネクタ
- 8. PCのための type B USBコネクタ ↔ BeeHive204AP通信ケーブル
- 9. USBケーブルを固定するためのタイマウント
- 10. 温度制御ファン
- 11. "GND" ネジはプログラマの接地に使用できます



BeeHive204APの背面ビュー

BeeProg2AP をPC に接続

BeeHive204APをPCに接続 プログラマをPCに接続するための推奨事項: 1.プログラマとPC、又は、その他のアースをアース接続します 2.USBケーブルを介してプログラマーをPCに接続します 3.電源をプログラマに接続します.

セルフテストとカリブレーション・チェック

プログラマが期待通りに動作しないと感じる場合は、標準パッケージに同梱されているAP1 PMIセルフテス トPOD(AP1 ISPコネクタ ・セルフテストPOD)を使用してプログラマ(ISPコネクタ)の"Selftest"を実行してください。

プログラマのセルフテスト

- プログラマ・サイトのプログラミング・モジュール・インタフェース(PMI)コネクタにAP1 PMIセルフテストPODを 挿入します.
- プログラマのセルフテストをPG4UW(メニュー Programmer/Selftest plus)で実行します





AP1 PMI セルフテスト pod

ISPコネクタのセルフテス

- AP1 ISPコネクタ・セルフテストPODをプログラマのプログラマ・モジュール・インターフェイス(PMI)コネクタに挿入します
- AP1 ISPコネクタのセルフテストPODの20ピン・コネクタとプログラマのISPコネクタを出荷時のプログラマ・パッケ ージに含まれているISPケーブルで相互接続します。ピンが正しく相互接続されていることを確認してください (つまり、1-1、2-2 ... 20-20).
- PG4UWでISPコネクタのセルフテストを実行します(プログラマ/セルフテストISPコネクタ)



AP1 ISP コネクタ・セルフテスト pod



- AP1カリブレーション・テストPODをプログラマのプログラマ・モジュール・インターフェイス(PMI)コネクタに挿入します。
- PG4UW(メニュー Programmer/Calibration test)でプログラマのカリブレーション・テストを実行します。



AP1 カリブレーション・テスト pod

BeeProg2AP によるマルチ・プログラミング

Select Additional TasksウィンドウでPG4UWのインストール中に、BeeProg2APマルチプログラミン グ・コントロール・サポートのインストールが許可されているかどうかを確認します。 BeeProg2APマルチ・プログラミングの開始には特別な制御プログラムpg4uwmc.exeを実行する必 要があります。このプログラムでユーザーはBeeProg2APを制御プログラムに割り当て、全ての BeeProg2APのためのプロジェクトをロードし、接続、及び、割り当てられた全てのBeeProg2APに対 してPG4UWを実行できます。

テクニカル・スペシフィケーション

一般

- 外部電源ユニット:動作電圧AC 100-240V定格、最大90-264 VAC、47-63Hz、DC出力15V、
 4.3A、出力ケーブル長1000mm(39.4 inch)
- 電力消費最大 20W 動作時, 2W スリープ時
- 寸法: 84x205x61 mm (3.3x8.1x2.4 inch) プログラミング・モジュールが挿入されていない場合の寸法を測定しました。プログラミング・モジュールが挿 入されたBeeProg2APプログラマの合計の高さはZIFソケットの高さに依存し、80~90mmの間で変化 する可能性があります。
- 重さ; 0.7kg (1.5 lb)
- •操作温度: 5°C ÷ 40°C (41°F ÷ 104°F)
- 操作湿度: 20%..80%, 非露結







プログラマ・パッケージには制御プログラム、有用なユーティリティと追加情報を含むCDが入っています。 Elnecのプログラマの働きを示すためにデモンストレーションするためにCDの内容を自由にコピーする許可 が与えられています。

USBポート経由で接続されたプログラマの場合、制御プログラムには正しくインストールされたUSBドライバが必要です。

インストール時に不要な複雑さを避けるためにプログラマをPCに接続する前にソフトウェアをインストールすることを推奨します。

ソフトウエア・セットアップ

配布されたCDをCDドライブに挿入しますとプログラムは自動的(インストールされていない場合は setup.exeを実行します)にインストールします。インストールプログラムはインストールプロセスをガイドし、最 初にコントロール・プログラムを実行する前に必要なすべての手順を実行します。

Step 2



"Install software for programmutton"をクリック





Step 1.



Step 3.



"次へ"ボタンをクリック

Step 4.

B Elnec PG4UW/PG4UWMC セットアップ - □ ×
インストールモード選択 ディフォルト または カスタマイズ インストール時の設定を ディフォルト または カスタマイズ から選択します。
インストールを開始しますのでディフォルト設定またはカスタマイズ設定のどちらかを選択して 下さい。そして次ボタンを押して下さい
● ディフォルト (インストール時の初期)設定を使用します)
○ カスタマイズ (インストール設定を変更したい場合、ここを選択します)
< 戻る(B) 次へ(M) > キャンセル

デフォルト設定では"次へ"ボタンをクリックします。セットアップはStep 6に進みます。 デフォルト設定を変更するには"カスタム"をクリックしてから"次へ"ボタンをクリックします。



Step 5.

🐻 Elnec PG4UW/PG4UWMC セットアップ	1000		×
インストール先の指定 Elnec PG4UW/PG4UWMC のインストール先を指定してください。			R
Elnec PG4UW/PG4UWMC をインストールするフォルダを指定 してください。	して、「次	へ」をクリッ	Ċ
続けるには「次へ」をクリックしてください。別のフォルダーを選択するに(ください。	「参照」を	とクリックして	5
C:¥Program Files (x86)¥Elnec_sw¥Programmer		参照(R)	
このプログラムは最低 0.9 MB のディスク空き領域を必要とします。			
〈戻る旧〉 次への) >	キャン	セル

デフォルトのフォルダを変更するには"参照""ボタンをクリックして下さい。 そして、"次へ"ボタンをクリックします。

Step 6.

Elnec PG4UW/PG4UWMC セットフ	アップ		
ブログラムグループの指定 プログラムアイコンを作成する場所	を指定してください。		
どう セットアップはスタートメコ	ニューにプログラムのショートカ:	ットを作成します	•
続けるには「〉次へ」をクリックしてくた ください。	ださい。違うディレクトリを選択	するには「参照」	をクリックして
Elnec			参照(R)

デフォルトのフォルダを変更するには"参照""ボタンをクリックして下さい。 そして、"次へ"ボタンをクリックします。

PELNEC

Step 7.

Blnec PG4UW/PG4UWMC セットアップ - □	×
追加タスクの選択 実行する追加タスクを選択してください。	R
Elnec PG4UW/PG4UWMC インストール時に実行する追加タスクを選択して、「次へ」を クしてください。	עעל
☑ スタートメニュー(ニアイコンを登録する	
☑ デスクトップにアイコンを登録する	
✓ マルチプログラミング機能をインストールする。:	
- BeeHive208S, BeeHive8S - BeeHive304, BeeHive204, BeeHive204AP, BeeHive204AP-AU - BeeHive4+ - BeeProg3, BeeProg2, BeeProg2C, BeeProg2AP, BeeProg+	
□LPTドライバーのインストール(プログラマのパラレル LPTポートを接続して下さい)	
< 戻る(B) 次へ(N) > キャ	ッンセル
"フリチ・プログラミング操作なインフトーリオス"が選択されている	ーレた球

"マルチ・プログラミング機能をインストールする"が選択されていることを確認します。必要に応じてデフォルト設定を変更します。そして、"次へ"ボタンをクリックします。

Step 8.

0.71	340W/PG4	7					-
1ノスト コンピ 3.38g	ール年間元 ユータにインス g/02.2018、3	」 、トール用ソフト El をインストールする	inec PG4UW/PG4UV 準備が出来ました。	VMC、バージョン		Ę	3
インス クして	、トールを続行 ください。	するには「インス」	~ール」を、設定の確	認や変更を行う	には「戻	る」をクリッ	
イン.	ストール先: C:¥Program	Files (x86)¥Work	prog¥Programmer			^	
プロ・	グラムグルー: Elnec	9:					
追加	ロタスクー覧: スタートメニュ デスクトップに	ーにアイコンを登録 アイコンを登録す ミング機能をイン	录する る ストールする。:				
	マルチプロクラ - BeeHive	208S, BeeHive8S					
	マルチフロクラ - BeeHive - BeeHive - BeeProg	208S, BeeHive8S 304, BeeHive204 3, BeeProg2, Bee	, BeeHive204AP, BeeProg2C, BeeProg2	eeHive204AP-Al AP, BeeProg+	U <mark>, Bee</mark> H	ive4+ 🗸	

設定を確認の上、"インストール"ボタンをクリックして下さい。



Step 9.

Elnec PG4UW/PG4UWMC セットアップ	_		×
インストール状況			-
ご使用のコンピューターに Elnec PG4UW/PG4UWMC をインスト くお待ちください。	ールしています。	JUGS	S.
ファイルを展開しています			
C:¥Program Files (x86)¥Workprog¥Programmer¥Pg4uw_er	n.chm		
			-12
		المراجع ا	ار جل
		177	101

インストレーション・プロセスの開始

Step 10.



セットアップを終了するには"完了"ボタンをクリックして下さい。



プログラマソフトウエアの新しいバージョン

プログラマの全ての機能を活用するためにPG4UWの最新バージョンを使用することをお勧めします。 インターネットサイトhttps://www.elnec.com/download/からプログラマ・ソフトウェアの最新バージョン (pg4uwarc.exeファイル)をダウンロードできます。

pg4uwarc.exeを一時ディレクトリにコピーし、プログラマをPCから切断して起動します。 前の章のステ ップ2からセットアップが開始されます。

ハードウエア・セットアップ

書告:プログラマの通信量が多いため各プログラマをUSB 2.0高速コントローラ(USB EHCI)に接続すること
を推奨します。新しいPCマザーボードの殆どには2つ以上のEHCIコントローラがチップセットに内蔵されてい
ます。そうでない場合は、PCI(PCI-E) USBアドオンカード(ルネサスUSBチップセットの使用を推奨)を使用
できます。マザーボード・チップセットに内蔵されているEHCIを使用する場合は、マザーボードのマニュアル、
又は、マザーボードメーカーのUSBポート・マッピングの技術サポートを参照して各プログラマを分離した
EHCIに接続できるようにします。一般的にはプログラマをPCのUSBポート(USB HUBなし)に直接接続し
マザーボードに搭載された直接(主にPCの背面にある)USBポートに接続することをお勧めします。

プログラマは制御プログラムがインストールされる前にUSBポートに接続されると、Windowsが新しいハード ウェアを検出し、ドライバのインストール方法を選択するように自動的、又は、手動でユーザーに求めます。 プログラマを正しく検出するには、制御プログラムのインストールCDをコンピュータのCD-ROMドライブに挿入 し、次の手順を実行する必要があります:

Step 1.

USB ケーブルをプログラマの type BUSB ポートに直接接続します。

Step 2.

USB ケーブルをプログラマの type A USB2.0 ポートに直接接続します。

Step 3.

電源ケーブルのコネクタをプログラマとウォール・プラグの適切なコネクタに接続します。

Step 4.

プログラマをオンにします。この時点で、全ての作業結果LEDが連続して点灯し、その後LEDがオフになります。



Windows 7とWindows 8.

Step 5.

Iタスク・バーの通知領域(主に右下隅)に下記の通知バブルが表示されます:



プログラマのためのドライバーが正常にインストールされたら以下のように表示されます。



ノート:別のプログラマがPCに接続されている場合(同じUSBポートに接続されている可能性があります)、 "Installing device driver software"[デバイス・ドライバ・ソフトウェアのインストール]が再度起動します。同じ プログラマが他のUSBポートに接続されている場合、追加のドライバーのインストールの必要はありません。

Windows XP, Service Pack 2' Windows Vista:

Step 5.

Windowsは"新しいハードウェアの検出ウィザード"から始まります。



"No, not this time[いいえ、今ではありません]"を選択し、"Next[次へ]"ボタンをクリックして下さい。





"Install the software automatically[ソフトウェアを自動的にインストールする]" を選択し、"Next[次へ]"ボタンをクリックします。

Step 6.



ウィザードがプログラマを検索しドライバーのインストールを自動的に開始します。



Step 7.

プログラマが正常にインストールされると、次のウィンドウが表示されます:



セットアップを終了するために"Finish"ボタンをクリック

Step 8.

"Found new hardware wizard"が各プログラマ(プログラマ・サイト)に対して1回起動されます (BeeHive204APの場合は4回)。Step 5でハードウェアの設定を続けます。 ノート: プログラマの次の接続にPCの別のUSBポートが使用される場合、 "Found new hardware wizard"が

再度起動し新しいUSBドライバーがインストールされます。



ターゲット・システムへのインストール

この章ではBeeHive204APとBeeProg2APのメカニカルな図とターゲット・システムへのBeeHive204APと BeeProg2APのインストールに必要なその他の情報を含みます。

BeeHive204AP

以下の図はプログラミング・モジュールを使用したBeeHive204APの全体的な寸法です。プログラミング・ モジュールがインストールされているプログラマの全高はプログラミング・モジュールのZIFソケットの高さによっ て異なります。合計の高さは**68.5mm+ZIFソケットの高さ**によって決めることができます。



BeeHive204AP寸法 右上図

X軸はZIFの中心はプログラミング・モジュールの中心と同じですが、Y軸では同じではありません。多くのZIFの 場合、それらの中心は画像と同じ位置にありますが、いくつかの特殊な大きなZIFの場合、ZIFの中心はY軸に 移動します。

ノート:

プログラマ間の最小距離は次のとおりです:

- プログラマの対面構成で2+ cm
- プログラマのテイル部と直面構成で5+ cm


USBコネクタと電源ケーブルが背面コネクタに接続されている場合、プログラマの総奥行きは205mm + USBケーブルのBタイプUSBコネクタの長さになります。プログラマの合計深度は205mm + 50mmになる場合があります。 USBコネクタと電源ケーブルが右側のコネクタに接続されている場合、プログラマの合計幅は310mm + USBケーブルのBタイプUSBコネクタの長さになります。プログラマの合計の深さは310mm + 50mmです。



USBケーブルをケースに取り付け、ケーブルを曲げてUSBコネクタの長さを調整します。

DCアダプタをプログラマに正しく接続するには、ケーブル・コネクタの矢印をプログラマ・コネクタの矢印に合わせる 必要があります。



電源コネクタ上の矢印

このコネクタには不要な切断を防ぐロック機構があり 切断するにはケーブル・コネクタのプル・ロッキング・カラーが必要です。



BeeHive204AP はナット付きM4の6個のネジでベースプレート上に取り付けることができます。ネジの長さ はペースプレートの厚さによって異なります。ベースプレートには ϕ 4,5mmの穴をM4の内部ネジで置き換え ることができます。

プログラマをベースプレート上に取り付けるための図面::





BeeHive204APはベースプレートの下にM4の6個のネジで取り付けることができます。ネジの長さはベース プレートの厚さによって異なります。

注意:プログラマのベースプレートに適切に取り付けるには、プログラマの取り付けネジの深さを最小3mmにする必要があります。プログラマの取り付けネジのPCBの損傷を避けるため、プログラマのネジは最大8mmの深さでなければなりません。

プログラマをベースプレートの下に取り付けるための図面:





BeeProg2AP

以下の図はプログラミング・モジュールを使用したBeeProg2APの全体的な寸法です。プログラミング・モジュールがインストールされているプログラマの全高はプログラミング・モジュールのZIFソケットの高さによって 異なります。合計の高さは68.5mm+ZIFソケットの高さによって決めることができます。



BeeProg2AP寸法 右上図

X軸では、ZIFの中心はプログラミング・モジュールの中心と同じですが、Y軸では同じではありません。多くのZIF の場合、それらの中心は画像と同じ位置にありますが、いくつかの特別な大きなZIFの場合、ZIFの中心はY軸 に移動します。



USBコネクタと電源ケーブルが背面コネクタに接続されている場合、プログラマの総奥行きは205mm + USB ケーブルのBタイプUSBコネクタの長さになります。プログラマの合計深度は205mm + 50mmになる場合が あります。



USBケーブルをケースに取り付け、ケーブルを曲げてUSBコネクタの長さを調整します。

DCアダプタをプログラマに正しく接続するには、ケーブル・コネクタの矢印をプログラマ・コネクタの矢印に合わせる 必要があります。



電源供給コネクタの矢印

このコネクタには不要な切断を防ぐロック機構があり 切断するにはケーブル・コネクタのプル・ロッキング・カラー が必要です。



BeeProg2APはベースプレート上にM3の2個のネジで取り付けることができます。ネジの長さはベースプレートの厚さによって異なります。

注意:プログラマのベースプレートに適切に取り付けるには、プログラマの取り付けネジの深さを最小3mmにする必要があります。プログラマの取り付けネジのPCBの損傷を避けるため、プログラマのネジは最大8mmの深 さでなければなりません。

プログラマをベースプレート上取り付けるための図面:

Assembly holes in base plane, when programmer is on base plate





BeeProg2APはベースプレートの下にM4の個のネジで取り付けることができます。ネジの長さはベースプレートの厚さによって異なります。

注意: プログラマのベースプレートに適切に取り付けるには、プログラマの取り付けネジの深さを最小3mmにす る必要があります。プログラマの取り付けネジのPCBの損傷を避けるため、プログラマのネジは最大8mmの深さ でなければなりません。

プログラマをベースプレートの下に取り付けるための図面:

Assembly holes in base plane, when programmer is bellow base plate





BeeProg2APはM3の4個のネジで左側をベースプレートに取り付けることができます。ネジの長さはベースプレートの厚さに依存します。

注意: この方法でBeeProg2APを取り付ける前に、左側にある2本のねじM3を外します。次に、適切な長さの 新しいM3ネジと交換します。プログラマをベースプレートに適切に取り付けるには、プログラマの取り付けネジの深さ を最小3mmにする必要があります。プログラマの取り付けネジのPCBの損傷を避けるためプログラマーネジは最大 8mmの深さでなければなりません。

ベースプレートの左側にプログラマを取り付けるための図面:





PG4UW



PG4UW プログラマ・ソフトウエア

プログラムPG4UWはエルネックの全てプログラマにとって共通の制御プログラムです。前述の全てのオペレー ティング・システムでこれらのプログラムの実行を保証します。Windowsでのバック・グラウンド操作にエラー はありません。

プログラマ・ソフトウエアの使用



コントロール・プログラムの実行



Windows 環境で: PG4UWアイコンをダブル・クリックします。

起動後、制御プログラムPG4UWは自動的に既存の全てのポートをスキャンし接続されたElnecプログラマ を検索します。プログラムPG4UWは全てのElnecプログラマに共通しているためサポートされている全ての プログラマを検索します。

Jート: PG4UWが起動すると、プログラムは完全性のために自己チェックを実行します。成功すると プログラムは標準のユーザー・メニューを表示し、指示を待ちます。

制御プログラムがプログラマと通信できない場合は、エラーコードと考えられる理由(プログラマの切断、 接続不良、電源障害、互換性のないポート等)の説明を含むエラー・メッセージが画面に表示されま す。エラーの原因を取り除き、いずれかのキーを押します。エラー状態が依然として存在する場合、プロ グラムはデモ・モードで動作を再開し、プログラマへのアクセスは不可能です。エラーの原因が見つからな い場合は、トラブル・シューティングのセクションの手順に従ってください。さらに、制御プログラムはプログ ラムされるデバイスでの操作の前にプログラマとの通信をチェックします。



ユーザー画面の説明

ウィンドウズ・プログラム PG4UW

G4UW v3.09/10.2014 - universal control program for Elnec p	rogrammers.	
Buffer Device Programmer Options Help		
👌 🗸 📊 🦚 🗸 🐝 🚺 🚺 🐝 aad Save Load prj Save prj View/Edit Select/def	Select Blank Read Verify Program Erase	PELN
0 0 0 0 0 0 0 0 0 0	🖗 Ų 🖻 🖻 📔	
ogrammer activity log		
100: All programmers are going to be scanned o	an selected communication port	
121. All programmers are going to be scanned t	a selected communication port.	
122-		
123: >> 05.11.2014, 14:39:04		
124: Scanning port(s) for BeeHive4. Site #1	not found. (EC:0003/DE:FFFFFFFF)	
125: Scanning port(s) for BeeHive4+, Site #1 .	not found. (EC:0003/DE:FFFFFFFF)	
126: Scanning port(s) for BeeHive204, Site #1	not found. (EC:0003/DE:FFFFFFFF)	
127: Scanning port(s) for BeeHive204AP, Site #	1 not found. (EC:0003/DE:FFFFFFF)	
128: Scanning port(s) for BeeHive204AP-AU, Sit	e #1 not found. (EC:0003/DE:FFFFFFF)	
129: Scanning port(s) for BeeHive304, Site #1	not found. (EC:00001003/DE:FFFFFFFF)	
130: Scanning port(s) for BeeHive8S, Site #1 .	not found. (EC:0003/DE:FFFFFFFF)	
131: Scanning port(s) for BeeHive208S, Site #1	not found. (EC:0003/DE:FFFFFFFF)	
132: Scanning port(s) for BeeProg	not found. (EC:0003/DE:FFFFFFFF)	
133: Scanning port(s) for BeeProg+	not found. (EC:0003/DE:FFFFFFF)	
134: Scanning port(s) for BeeProg2	not found. (EC:0003/DE:FFFFFFFF)	
135: Scanning port(s) for BeeProg2C	not found. (EC:0003/DE:FFFFFFFF)	Statistics
136: Scanning port(s) for BeeProg2AP	not found. (EC:0003/DE:FFFFFFFF)	Success: 0
137: Scanning port(s) for BeeProg3	found at port USB.	Operational failure: 0
138: Starting-up programmer firmware done.	- land	Other failure: 0
139: Internal programmer firmware is being che	ckea.	Total: 0
140: Checking internal builer memory, wait pie	ase done.	Beset statistics
141: Establishing connection done.		-
143. Flogrammer Seriar Humber, 1200-00014.		Count down: Disable
144-		Bemains 0 of 1
		T Roland Count down
III		F Holda Count down
Idresses (hex)	Programmer	
Org. Size Start End	Type: BeeProg3	Status: Ready
evice x8 8400000 0 83FFFFF	S/N: 1200-00014	Port: USB
		YESI: Manual
iller x8 8400000 0 83FFFFF	Device	
nter x8 8400000 0 83FFFFF e x8	171.711.1.	
amer x8 8499999 9 83FFFFF e x8	Type: Samsung K9E1608U0E [TSOP48]	
after x8 8499999 9 83FFFFF e x8 - - - eckSum: Calculate x8-S [0h.83FFFFFh]	Type: Samsung K9F1G08U0E [TS0P48] Adapter: AP3 TS0P48-18,4mm NAND	
after x8 8489999 9 83FFFFF e x8 - - - eckSum: Calculate x8-S [0h83FFFFFh] rialization:	Type: Samsung K9F1608U0E [TSOP48] Adapter: AP3 TSOP48-18.4mm NAND Note: See also Device info <ctrl+f1>!</ctrl+f1>	
atter x8 8499999 9 83FFFFF e x8 - - - veckSum: Calculate x8-S [0h. 83FFFFFh] initization: None - -	Type: Samsung K9F1608U0E [TS0P48] Adapter: AP3 TS0P48-18.4mm NAND Note: See also Device into <cut+f1>1 To cutstomize device use menu <u>Access Method <al+s></al+s></u></cut+f1>	

Toolbars[ツールバー]

メイン・メニューの下によく使用されるボタン・ショートカットを持ったツールバーがあります。 ツールバーはメニューのオプション->ビューで変更することが出来ます。

Log window[ログ・ウィンドウ]

ログ・ウィンドウはPG4UWでの殆どの操作の流れについての情報を含みます。 操作:

- ・PG4UWの開始
- ・プログラマ・サーチ
- ・ファイル/プロジェクト・ロード/セーブ
- ・デバイスの選択

・デバイス操作(デバイス・リード、ブランク・チェック、プログラミング等)

・リモートコントロール・アプリケーションの接続と切断

その他の各種情報

これらの情報は問題が有った時のためにカット&ペースでテキスト・ファイルにコピーすることも出来ますが、通常 はヘルプ->プロブレム・レポート作成をクリックしますと

"PG4UW_LOG_windows_content_xxxx.zip"としてデスクトップ上に作成しサポートのためにメールに添付して送ることが出来ます。



Panel Addresses[パネル・アドレス]

パネル・アドレスは現在選択されているデバイスの実際のアドレス範囲,ロードされたファイルとバッファの開始と終了 アドレス設定についての情報を含んでいます。ある種のデバイスではメニューのデバイス->デバイス・オプション->操 作オプションによってデフォルト・デバイスとバッファ・アドレス範囲を変更できます。

パネル・アドレスには Split、Serialization、及び、Buffer Checksumの現在の状態に関する高度な情報も含まれています。それぞれのオプションの詳細については、次をご覧ください:

- · Split[スプリット] メニュー デバイス/デバイス・オプション/操作オプション
- · Serialization[シリアライゼーション] メニュー デバイス/デバイス・オプション/シリアライゼーション
- ・Checksum[チェックサム] メニュー バッファ/メインウィンドウに表示されるセクション・チェックサムのチェックサム

Panel Programmer[パネル・プログラマ]

パネル・プログラマは現在選択されたプログラマについての情報を含みます。

- ・ プログラマ・タイプ
- ・コンピュータに接続されているプログラマのポート
- ・ 次の何れかのプログラマの状態
- · Ready プログラマが接続され、正常に見つかり作業準備が出来ている
- ・ Not found プログラマが見つかりません
- ・ Demo ユーザーがダイアログ Find programmer[プログラマの検索]でオプション(ボタン)でデモを選択時
- ・YES! mode 一部のタイプのプログラマは次のいずれかの方法で次のデバイス動作を開始する特別なモードを 使用することができます。
- ・制御プログラムのダイアログ Repeat により手動
- ・ボタン START!により手動でプログラマに直接配置されます。
- ・自動 プログラマは自動的にデバイスの取り外しと新しいデバイスの挿入を検出します。

更に詳しくは Programmer / Automatic YES!をご覧ください。

Panel Device[パネル・デバイス]

現在選択されているデバイスに付いての情報を含みます。

- ・デバイス名(タイプ)とマニュファクチャラー
- ・現在選択されているプログラマで使用する必要のあるデバイス・アダプタ(又は、モジュール)
- ・詳細なデバイス情報ダイアログの参照。メニュー デバイス/デバイス情報でも使用できます。
- ・高度なデバイス・オプションの参照 これは一部のデバイスでのみ利用可能です。

Panel Statistics[パネル・スタティスティクス]

現在選択されているデバイスに付いての統計情報を含みます。

- ・成功、失敗とデバイス操作合計の数
- ・カウント-ダウン・ステータスが残りのデバイスの表示

統計とカウント-ダウン・オプションはメニューコマンド Device / Device options / Statistics[デバイス/デバイス・オ プション/統計]、又は、パネル Statistics[統計]をマウスの右ボタンでクリックし、ポップアップ・メニューから項目 Statistics[統計]を選択することで使用できます。

Panel File[パネル・ファイル]

パネルは PG4UW メイン・ウィンドウの下部に配置されます。パネルには現在ロードされているファイル、又は、プロ ジェクトの名前、サイズ、及び、日付が表示されます。



<ホット・キーの	リスト>	
<f1></f1>	ヘルプ	ヘルプを呼ぶ
<f2></f2>	セーブ	ファイルの保存
<f3></f3>	ロード	ファイルをバッファにロード
<f4></f4>	エディット	バッファのビュー/編集
<f5></f5>	選択/デフォルト	最後に選ばれた 10 のデバイス・リストからターゲット・デバイスを
選択		
<alt+f5></alt+f5>	選択/手動	デバイス/ベンダー名をタイプすることでターゲット・デバイスを選択
<f6></f6>	ブランク	ブランク・チェック
<f7></f7>	リード	デバイスの内容をバッファに読み込み
<f8></f8>	ベリファイ	ターゲット・デバイスとバッファの内容を比較
<f9></f9>	プログラム	ターゲット・デバイスをプログラム
<alt+q></alt+q>	保存せずに終了	プログラムを終了
<alt+x></alt+x>	保存して終了	設定を保存してプログラムを終了
<ctrl+f1></ctrl+f1>	デバイス情報	現在のデバイスの追加情報を表示
<ctrl+f2></ctrl+f2>	イレース	与えられた値でバッファをフィル
<ctrl+shift+f2></ctrl+shift+f2>	・ フィル・ランダム・データ	ランダム値でバッファをフィル

File[ファイル]

メニューメニューはソース・ファイル(binary, MOTOROLA, MOS Technology, Intel (extended) HEX, Tektronix, ASCII space, JEDEC及びPOF のフォーマット)の各種操作として、設定とビュー・ディレクトリー、ド ライブ変更、ファイルのロードとセーブの為のバッファ・メモリの開始アドレスと終了アドレスの変更に使用します。

File / Load[ファイル/ロード]

ファイル形式を解析した後、指定されたファイルからバッファにデータをロードします。ご使用に合った形式(binary, MOTOROLA, MOS Technology, Tektronix, Intel (extended) HEX, ASCII space, JEDEC 及び POF)を選んでください。コントロール・プログラムは、最後の有効なマスク情報をファイル・リストに順次記録して行 きます。options / Save options コマンドで、コンフィグ・ファイルにマスク情報をセーブ出来ます。

<F3>によっていつでもどのメニューからでもこのメニューを呼び出す事が出来ます。

File formats[ファイルフォーマット]の説明:

ASCII HEX フォーマット

各バイトデータは 2 つの 16 進数で表され、他の全てのデータ・バイトから空白スペースによって区切られています。 データ:バイトのためのアドレスは\$Annnn, キャラクターのシーケンスを使ってセットされます。nnnn は 4 つの 16 進数 アドレスです。後ろにカンマが必要です。各データバイトがアドレスを持っているが明白でない場合、明示的なアド レスがデータストリームに含まれていない限りデータ・バイトは連続してアドレス指定されます。最初のデータバイトの 前にアドレス部分が設定されていない場合は、ファイルは 0 から開始します。ファイル STX(Control-B)文字 (0x02)で始まり ETX(Control-C)文字(0x03)で終わります。

ノート: チェックサム部分は\$Sとカンマ文字間の4つの16進数文字列で構成されます。チェックサムはファイルの最後の部分になります。

ASCII HEX ファイルの例:データ"Hello, World"をアドレスの 0x1000 にロードしています: ^B \$A1000: 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 0A ^C \$S0452,



ASCII SPACE フォーマット

ASCII HEX とよく似た非常に単純な HEX フォーマットで開始(STX)と終了(ETX)文字列の無い HEX フォーマットです。各データバイトは 2 つの 16 進数文字として表現され、他のすべてのデータバイトの空白で区切られています。アドレス・フィールドはデータ・バイトから空白で区切られています。アドレスは 4-8 の 16 進数文字のシーケンスを使用して設定されます。

ASCII SPACE ファイル例:データ"Hello, World"をアドレスの 0x1000 にロードしています: 0001000 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 0A

Straight HEX フォーマット

ASCII HEX と同様の非常に単純な HEX ファイル・フォーマットでアドレスとチェックサムとスタート(STX)とエンド (ETX)を持たないフォーマットです。各バイトデータは2つの16進数で表され全ての他のデータ・バイトからはスペー スによって区切られています。

Straight HEX ファイルの例:データ"Hello, World"を含む: 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 0A

Samsung HEX フォーマット

Samsung HEX フォーマットは Intel HEX フォーマットを少し修正したもので、ファイルの形式、及び 構成等は Intel HEX フォーマットと略同等ですのでソフトウェアでは Intel HEX ファイル形式として認識され示されます。

特殊な x16 フォーマットのノート:

- Intel HEXx16はTMS320Fデバイスのための16bitデータ・ワードを持ったIntel Hexフォーマットです。
- Motorola HEXx16はTMS320Fデバイスのための16bitデータ・ワードを持ったMotorolaファイル・フォーマットです。

Automatic file format recognition[自動ファイル形式認識] チェック・ボックスにチェックをいれますと自動的に プログラムがファイル形式を検出します。プログラムがサポートしている形式の中からファイル形式を検出出来ない 場合は、バイナリー・フォーマットであることが考えられます。

[自動ファイル形式認識]チェックボックスをオフにすると[選択されたファイル形式]パネルの利用可能なファイル形式の 一覧から手動で希望のファイル形式を選択できます。デフォルトは[オプション/一般オプション]の[ファイル・オプション]タ ブの[ロード・ファイル形式]になります。

ノート:プログラムはASCII Hexフォーマットのファイルを自動的に認識出来ません、バイナリーとして認識されます。 自動ファイル形式認識のためのオプションを無効にしてASCII Hexフォーマットのダウンロードを行って下さい。

Panel Additional operation[パネル追加操作]

C チェック・ボックス Erase buffer before loading[ロードする前にパッファを消去]</mark>にチェックをいれますと入力さ れたイレース値を使って全てのバッファ・データをイレースするようにプログラムに指示します。バッファ消去はファイル を読み込む前に直ちに実行されます。これはバイナリーと全ての HEX ファイル形式のための機能です。このワンショ ット設定を使用すると現在の Hex file options のメニューの Options /General options で Erase buffer before loading オプションを無効にします。

チェック・ボックスにSwap bytes[スワップ・バイト] がチェックにされていますと、ユーザーはファイル読み込み中 に16ビット・ワード(又は、2-バイト・ワード)内でスワッピング・バイト機能をアクティベートすることが出来ます。 この機能はモトローラのバイト形式のファイル(ビッグ・エンディアン)でロードする時に便利です。標準のロード・ ファイルではリトル・エンディアンのバイト形式を使用します。



Jート: ビッグ-エンディアンとリトル-エンディアンとはコンピュータのメモリーに書き込まれるバイトのシーケンス順を現す 方法です。ビッグ-エンディアンはビッグ・エンド(Most significant 最上位の値)が最初にストアされます。例えば、 ス)にストアされます。リトル-エンディアンはリトル・エンド(シーケンスの最下位の値)が最初にストアされます。例えば、 ビッグ-エンディアンのコンピュータではヘキサ・デシマル値 4F52 は2バイトが要求されストレージ・アドレス 1000H に は 4F52H としてストアされます。4FH はストレージ・アドレス 1000H に、そして、52H はバイト・アドレス 1001H と なります。リトル-エンディアン・システムでは、それは 524FH (1000H が 52H、そして、アドレス 1001H が 4FH)とし てストアされます。

4F52H がメモリーにストアされる内容を次に示します:

アドレス ビッグ・エンディアン・システム リトル・エンディアン・システム 1000H 4FH 52H 1001H 52H 4FH

Add blank spare area[ブランク・スペア領域の追加] - (for NAND Flash devices) チェックにしますとファイ ルのロード中にバッファ(選択したデバイスによる)内の関連する位置に空白のスペア領域データを追加します。

Paneral Buffer offset for loading[ローディングのためのバッファ・オフセット]

パネル Buffer offset for loading[ロード用のバッファ・オフセット]はファイルからバッファヘデータをロードするた めのワンショット・オフセット設定が含まれています。この設定はバッファに格納するためにロードされたデータのオ プションのオフセットを指定するために使用されます。Load file[ロード・ファイル]ダイアログ・ウィンドウを開くと、オ フセットは常にデフォルト設定Noneになります。これは、バッファに読み込みデータを格納するためにオフセットが 使用されないこと意味します。

利用出来るオフセット・オプション:

None[なし] ファイルからバッファへのローディングにオフセットは適応されません。

Positive offset[ポジティブ・オフセット] バッファにデータを格納するために現在のアドレスに追加されるオフセット 値のセット。このオフセットは全てのフォーマットで使用でき、現在のバッファ構成が x8 の場 合は x8 形式で、現在のバッファ構成が x16 の場合は x16 形式で使用されます。 Negative offset[ネガティブ・オフセット] モードは2つのオプションを持っています:

Negative offset[ネガティブ・オフセット]とAutomatic negative offset[自動ネガティブ・オフセット] -手動、又は、自動の2つの方法でセット: 手動設定の場合は、オプションの負のオフセット[Negative offset]を 使用し、希望のオフセット値を編集ボックスに入力します。自動オフセット検出はオプション[自動負のオフセット [Automatic negative offset]]を使用してください。この値は、現在のアドレスからバッファにデータを保存する ために減算されます。

ネガティブ・オフセット値(手動定義、又は、自動検出)はデータをバッファに保存するため現在のアドレスから 減算されます。

ネガティブ・オフセットは全てのHEXファイルにのみ適応され、そして、常にx8フォーマットを使用します。ネガ ティブ・オフセット設定はバイナリ・ファイルと他のHEXで無いファイルでは無視されます。

ノート:

- ・ 負のオフセットの値は実際のアドレスから減算されているので減算結果が負の数の場合は読み込めません。従って、正しい値を設定するように注意して下さい
 ・
- 特殊な場合にのみネガティブ・オフセットの自動設定をお勧めします。このオプションはファイル内のある種の間 違ったデータを扱うことが出来るヒューリスティック解析が含まれています。断片化されたアドレス範囲が含まれ ていたり、そして、選択されたデバイスのサイズを超えた様な問題のあるファイルに対して使用されます - ある 種のブロックは無視することが出来ます。



 Automatic negative offsetオプションは確実に指定されたブロックを持ったHEXファイルを必要とするような 特殊なある種のデバイスでは利用出来ません - 例えばMicrochip PICmicroデバイス。それらの特殊なデ バイスは手動オフセット設定(None, Positive offset, Negative offset)のみを利用して下さい。

ネガティブ・オフセットの使用例:

ファイルはMotorola S - フォーマットによるデータを含みます。 アドレス FFFF0Hで開始されたデータ・ブロック。 それは3バイトのアドレス・アレイの長さを持ったS2フォーマットです。 全てのデータ読み取りに対して負のオフセットオプションと負のオフセット値をFFFF0Hに設定できます。

これは現在の実際のアドレスからオフセットが差し引かれデータがバッファ・アドレス0から書き込まれることを意味します。

ファイル・フォーマットとエラー・コードのリスト

サポートされたフォーマットの幾つかでファイルのダウンロード中にエラーが起こり得ます。エラーはLOGウィンド ウに次の様に書かれます "Warning: error #xxy in line rrr", xxはファイル・フォーマット・コード, yはエラー・ コードと rrr は10進数での行番号

ファイル・フォーマット・コード:

- #00y binary
- #10y ASCII Space
- #20y Tektronix
- #30y Extended Tektronix
- #40y Motorola
- #50y MOS Technology
- #60y Intel HEX

ロード・ファイル・エラー・コード:

- #xx1 bad first character header
- #xx2 bad character in current line
- #xx3 bad CRC
- #xx4 bad read address
- #xx5 bad length of current line
- #xx6 too big negative offset
- #xx7 address is out of buffer range
- #xx8 bad type of selected file format
- #xx9 the file wasn't loaded all

File / Save[ファイル/保存]

作成された、変更された、又は、指定されたディスクにデバイスから読み取られたデータをバッファーに保存 します。保存されたファイルのファイル形式は、サポートされている形式のリスト・ボックスから選択できます。 ファイルに保存するバッファの一部を正確に指定するバッファ開始アドレスとバッファ終了アドレスを入力す ることもできます。現在サポートされているファイル形式はbinary, MOTOROLA, MOS Technology, Tektronix, Intel(extended) HEX, ASCII space, JEDECとPOF.

Swap bytes[スワップ・バイト] チェック・ボックスが表示されている場合、ユーザーはファイル書き込み中に 16 ビット・ワード(又は、2-バイト・ワード)内でスワッピング・バイト機能をアクティベートすることが出来ます。この機能 はファイルをモトローラ表現のバイト形式(ビッグ・エンディアン)でロードする時に便利です。標準のセーブ・ファイル 操作ではリトル・エンディアンのバイト形式を使用しています。



リザーブ・キー<F2>によってこのメニューをいつでも呼び出すことが出来ます。

File / Load project[ファイル/ロード・プロジェクト]

このオプションはデバイスの保存されたコンフィギュレーション・バッファ・データとユーザー・インターフェース・コンフィギュレーションを含んだプロジェクト・ファイルをローディングするために使用されます。

標準ダイアログ Load project[ロード・プロジェクト]は追加のウィンドウを含みます - Project description[プロジェクト説明] - ダイアログの下に置かれています。このウィンドウはダイアログ Load project で現在選択されているプロジェクト・ファイルの情報を表示ためのものです。 プロジェクト情報は下記を含みます:

- プロジェクト内で最初に選択されたデバイスのメーカーと名前
- プロジェクト作成日時
- ユーザーが書いたプロジェクトの説明(それは任意のテキスト、通常はプロジェクトの著者とある種の注釈 にすることができます)
- **ノート:** シリアライゼーションがオンになっているプロジェクト

シリアライゼーションは以下の手順でプロジェクト・ファイルから読み込まれます:

- プロジェクトに記述したシリアライゼーション設定が受け付けられます。
- 追加のシリアライゼーション・ファイルを検索実行さます。ファイルが検出された場合、それが読み込まれ、そして、追 加ファイルからのシリアライゼーション設定が受け付けられます。追加のシリアライゼーション・ファイルは常に特定のプ ロジェクト・ファイルに関連付けらされます。追加のシリアライゼーション・ファイルの設定が受け入れられる時、プロジ ェクトのシリアライゼーション設定は無視されます。

追加のシリアライゼーションファイル名はプロジェクト・ファイル名に拡張子 ".sn"を付けることによってプロ ジェクト・ファイル名から導出されます。 追加のシリアライゼーション・ファイルは常に制御プログラムのディレクトリ"serialization\"に置かれます。

サンプル:

プロジェクト・ファイル名: my_work.prj コントロール・プログラムのディレクトリ: c:\Program Files\Programmer\

追加のシリアライゼーション・ファイルは:

c:\Program Files\Programmer\serialization\my_work.prj.sn

追加のシリアライゼーション・ファイルはデバイスのプログラム成功後に作成され更新されます。 追加のシリアライゼー ション・ファイルを作成するための唯一の要件はシリアライゼーションをオンにしたロード・プロジェクトです。 ファイルが存在し、現在保存されたプロジェクトに関連する場合、 コマンドFile/Save project[ファイル/プロジェクトをセープ]は追加のシリアライゼーション・ファイルを削除します

Enter job identification dialog [ジョブ・アイデンティフィケーション入力ダイアログ]

このダイアログはプロテクトされたプロジェクト・ファイルをローディングされた時に表示されます。

2つの編集可能フィールドを持っています:

- オペレーターID このパラメータはプログラマのオペレーターを認識するため使用されます。オペレーター IDは3文字以上でなければいけません。プロテクトされたプロジェクトのJob Reportを作 成する時にパラメータが必要ですのでユーザーはオペレーターIDを入力しなければいけま せん。
- Job ID入力 現在行っている作業のJob ID認証を入力します。



ノート: ダイアログEnter job identificationはパスワードのダイアログではありません。オペレーター認証の値と Job IDは情報の目的に単にJob Reportを含んでいます。プロテクトされた/又は、暗号化されたプロジェクト・ パスワードとは関係ありません。

File / Save project[ファイル/プロジェクトをセーブ]

このオプションはプロジェクト・ファイルの保存に使用されます。プロジェクト・ファイルには保存されたデバイス構成とバッファ・データの設定が含まれます。プロジェクト・ファイルに保存されたデータはメニュー・コマンドの File/Load project[ファイル/ロード・プロジェクト]を使用していつでも復元できます。

ファイル・リストから実際に選択されたプロジェクトの説明

Save project[セーブ・プロジェクト]ダイアログ内に現在選択されているプロジェクト・ファイルに付いての情報が 表示されます。このボックスは情報のためですので書き込みは出来ません。

セーブされるプロジェクトの説明

上半分は現在選択されているデバイス、プログラム・モード、日時等の実際のプログラム構成についての情報を表示しますが内容等の変更は出来ません。これらの実際のプログラム設定は、プロジェクト記述ヘッダの作成に使用 されます。

下半分はユーザー編集可能で通常はプロジェクト作成者やメモで構成されたプロジェクトの説明(任意のテキス ト)が含まれます。

チェックボックEncrypt project file (with password)[暗号化プロジェクト・ファイル(パスワード付き)]は暗 号アルゴリズムを使った特殊なフォーマットでプロジェクトをセーブするために使用されます。これはパスワード無 しでソフトウエアにプロジェクト・ファイルをロードするのを防ぎます。キーでボタンをクリックした後、保存されるプ ロジェクトのための暗号化パスワードを指定するのに使用されるpasswordダイアログが表示されます。

チェックボックス Set Protected mode of software after loading of this project file[このプロジェク ト・ファイルを読み込んだ後、保護モードを設定する]はProtectedモードと呼ばれる特別なモードでプロジェク トを保存するために使用されます。キーでボタンをクリックした後、保存するプロジェクトの保護モードパスワード を指定するためのpasswordダイアログとオペレータのミスを防ぐための他のセキュリティ・オプション(他のプロジェ クトの読み込みを無効にする、デバイス操作制限)が表示されます。アクティブなProtected modeでセーブさ れたプロジェクトはProtected mode projects[プロテクティド・モード・プロジェクト]と呼ばれる特殊なプロジ ェクトです。Protected modeプロジェクトに付いての更に詳しい情報は Options /Protected modeをご 覧下さい。Protected modeがアクティブな時、ソフトウエアはプログラマ・アクティブ・ログの右上角のlabel Protected modeによってこれを表示します。

推奨: Encrypt project file (with password)[暗号化プロジェクト・ファイル(パスワード付き)]と Set Protected mode of software after loading of this project file[このプロジェクト・ファイルを読み込ん だ後、保護モードを設定する]のためのパスワードは同じものではあってはいけません。

チェックボックス Require project file checksum before first programming[最初のプログラミング の前の必要プロジェクト・ファイルのチェックサム]がアクティブな時,ロード・プロジェクトの後に最初のデバイ ス・プログラミングの開始前にソフトウエアはユーザーに正しいプロジェクト・ファイルのためのユニークなIDの入 力を聞いてきます。この機能は正しいプロジェクト・ファイルが最新にロードされたかを追加チェックするために 推奨されます。また、このチェックボックスはアクティブ Protected mode[プロテクティド・モード]で使用する ことをお薦めします。プロジェクト・ファイルの固有のIDの必要がアクティブな場合、ソフトウェアは制御プログ ラムのメイン・ウィンドウの一番下のステータス行にプロジェクト・ファイル名の隣にラベル(ID)によって表示さ れます。



Jート: オプション最初のプログラミングの前のRequire project file unique IDは以前の最初のプログラミングの前のRequire project file checksum[必要プロジェクト・ファイルのチェックサム]の替わりです。ジェネリック・チェックサムよりユニークIDの利点は、固有のIDがメイン・デバイス・バッファのデータから計算されるだけでなくデバイスと使用可能なデバイス設定で使用されるセコンダリー・バッファ・データから計算されることです。プロジェクト・ファイルのチェックサムの必要がアクティブな場合、ソフトウェアが制御プログラムのメイン・ウィンドウの一番下のステータス行にプロジェクト・ファイル名の隣にこのラベル(CSum)を表示します。このオプションはSave project[プロジェクト・ファイルを読み込んだ後にアクティブ にすることができます。

File / Reload file [ファイル/ファイルを再ロード]

最近使用したファイルを再ロードするためにこのオプションを選んで下さい。

ファイルを使用した時、Reload file[再ロード・ファイル]リストに追加されます。ファイルは使用した時間の順番にリストされます。最後に使用したファイルは以前に使用したファイルの前にリストされます。

ファイルの再ロード:

1.ファイル・メニューからReload file[再ロード・ファイル]を選択 2.最後に使用したファイルのリストが表示されます。再ロードしたいファイルをクリック

ノート: ファイルを再ローディングする時、そのファイルは最後にロード/セーブされたファイルで使用されたファ イル形式が使用されます。

File / Reload project [ファイル/プロジェクトの再ロード]

最近使用したプロジェクトを再ロードするためにこのオプションを選んで下さい。

プロジェクトを使用した時、Reload project[再ロード・プロジェクト]リストに追加されます。プロジェクト は使用した時間の順番にリストされます。最後に使用したファイルは以前に使用したファイルの前にリスト されます。

プロジェクトの再ロード:

1.ファイル・メニューからReload project{再ロード・プロジェクト}を選択.

2.最後に使用したプロジェクトのリストが表示されます。再ロードしたいプロジェクトをクリック

File / Project options[ファイル/プロジェクト・オプション]

このオプションは実際にロードされたプロジェクトの表示/編集プロジェクト・オプションのために使用されます。プロジェクト・オプションは次のプロジェクト・データに含まれているプロジェクトのベーシックな説明を意味します:

- デバイス名とマニュファクチャラ
- プロジェクト作成日付
- ユーザー定義プロジェクト説明(任意テキスト), 例えば、更に詳細なプロジェクト説明のための作者と他のテキ スト・データ

ユーザーはユーザー定義プロジェクト説明のみ直接編集することが出来ます。 デバイス名, マニュファクチャラー,プロ ジェクト・デートとプログラム・バージョンはプログラムにより自動的に生成されます。



File / Load encryption table[ファイル/暗号テーブルのロード]

このコマンドはディスクからバイナリー・ファイルでのデータをロードします。そして、それらをメモリーの一部にセーブ、暗 号(セキュリティ)テーブルのためにリザーブされます。

File / Save encryption table[ファイル/暗号テーブルの保存]

このコマンドは暗号テーブルが含まれたメモリーの部分の内容を、バイナリー・データとしてディスクのファイルに 書込みます。

File / Exit without save[ファイル/保存せずに終了]

このコマンドはヒープの割り当てを解除しディスク上のバッファを破棄して(存在する場合)、操作システムに戻しま す。

File / Exit and save [ファイル/終了と保存]

このコマンドはヒープの割り当てを解除しディスク上のバッファを破棄して(存在する場合)、最後に選択されたデバイ スの現在の設定を保存し操作システムに戻ります。

Buffer[/\\vec{y}\mathcal{Z}p]

メニューBuffer[バッファ]はバッファ操作、ブロック操作、ストリングでのバッファの部分のフィル、イレース、チェッ クサムと編集とその他(検索とストリングス再配置、印刷...)の項目でもビューに使用します。

Buffer / View/Edit[/\superiodication / Linear /

このコマンドはバッファのデータのview(ビュー・モード) 又は、edit(編集モード)するのに使用します(ビューはDUMP モードのみ)。オブジェクトの編集をするための選択は矢印キーを使用してください。編集したデータはカラーで強調 されます。

<F4> ホット・キーでも使用出来ます。

View/Edit Buffer[ビュー/バッファ編集]

このダイアログはバッファのデータをView(ビュー・モード)又は、edit(編集モード)に使用されます。選択され たチップのために配置されたデータの領域外のバッファのデータはグレイ・バックグラウンドで示されます。

次のコマンドがバッファ・データの編集のために利用できます。全てのコマンドが全てのシチュエーションで利用 できるわけではありません。選択されたデバイスとデバイスのために使用されるバッファに依存します。 ヘルプの表示

F1

F2 フィルのための開始と終了ブロックと要求されるhex(又は、ASCII)文字列をセットして下さ い。フィル・ブロックは選択されたバッファのブロックを要求されたhex(又は、ASCII)文字列 で埋め尽くします。

Ctrl+F2 指定したブランク値でバッファを消去

Ctrl+Shift+F2 ランダム・データでバッファをフィル

- バッファ・データをバイナリ・ファイルに保存します。このコマンドは2次バッファでのみ使用できます。 Shift+F2 セカンダリ・バッファはMicrochip PICmicroデバイス用のData EEPROM等の一部のデバイス で使用される特別な領域です。Load/Save data to/from Main buffer[メイン・バッファヘ/から データのロード/セーブ]コマンドはメイン・メニュー "File"で利用できます。メイン・アプリケーション・ ウィンドウにロード、セーブ・ボタンもあります。
- コピー・ブロックは新しいアドレス上の現在のバッファのデータの指定されたブロックを新しいアドレス F3 ヘコピーするのに使用されます。ターゲット・アドレスはソース・ブロック・アドレスの外である必要 はありません。



Shift+F3	バイナリ・ファイルからバッファにデータをロードします。このコマンドはセコンダリ・バッファーでのみ使
	用できます。詳細については上記のSave Buffer Dataコマンド(Shift + F2)の注意事項を参照してイギュン
F4	mu o c v c c v o ブロックの移動は新しいアドレス上へ現在のバッファのデータのブロックを移動するのに使用し
	ます。ターゲット・アドレスはソース・ブロック・アドレスの外である必要はありません。ソース・
F5	アドレス・フロック(乂は、一部分)はフランク・キャラクターによりフイルされます。 ユロップ・バイト・コマンドは現在のバッファ・ブロックのバイト・ペアのハイとローの順番をユロッ
10	プします。このブロックは偶数アドレスで開始しなければいけません。そして、バイトの偶数で
	なければいけません。もし、この条件が満たされないと、プログラムはアドレス自身を修正し
	ます(開始アドレスは低い偶数アドレスに移動されるか、又は、終了アドレスか高い奇数アドレフに移動されます)
F6	プリント・バッファ
F7	文字列検索(最大長 16 ASCII キャラクタ)
F8	文字列を検索し置き換え(最大16 ASCII キャラクタ)
F9	現在のアドレスを変更
F10	ビュー/編集モードを変更
F11	バッファ・データ・ビューのモードを8ビットと16ビットの間で切り替えます。 View/Edit mode
	Durier Indicator[Lユー/編集モート・ハッファ・イフ ティケータ]の石のホタフをマワスでク リックすることでも行えます。このボタンは実際のデータ・ビュー・モード(8ビット又は、16ビ
	ット)も表示します。
F12	チェックサム・ダイアログはバッファの選択されたブロックのチェックサムをカウントします。
Arrow keys	カーソル上下左右移動
Home/End	現在の行の開始/終了へジャンプ
PgUp/PgDn	前/次ページへジャンプ
Ctrl+PgUp/PgD)n 現在のページの開始/終了へジャンプ
Ctrl+Home/End	Ⅰ 現在のデバイスの開始/終了へジャンプ
Shift+Home/En	d 現在のバッファの開始/終了へジャンプ
Backspace	カーソルを1つ左へバック

Jート: キャラクタ 20H - FFH(ASCIIモード)と番号 0..9, A..F(HEXモード)は即座に編集エリアの内容を変 更します。 警告: ワード・デバイスに対するASCIIキャラクタの編集は出来ません。

Print buffer [プリント・バッファ]

このコマンドは選択されたバッファの部分をプリンター、又は、ファイルに書くことが出来ます。プログラムは表示されている選択されたバッファのブロックを外部テキスト・エディターでプリント、又は、ファイルにセーブすることが出来ます。デフォルトでシンプルなテキスト・エディターnotepad.exeが設定されています。

Print bufferダイアログには次のオプションがあります:

Block start[フロック開始] バッファ内の選択されたブロックの開始アドレスを定義 Block end[フロック終了] バッファ内の選択されたブロックの終了アドレスを定義

外部エディタ

この項目は選択されたブロックのためのテキスト・ビューワーとして使用される外部プログラムのパスと名前を 定義します。デフォルトではnotepad.exeに設定されています。ユーザーは大きなテキスト・ファイルで使用 できるwordpad.exe等のテキスト・エディターも定義することが出来ます。ユーザー定義テキスト・エディター でユーザーは印刷又は、バッファの選択されたブロックをファイルに保存することが出来ます。 外部エディターのパスと名前は自動的にディスクにセーブされます。



Find[テキスト検索]ダイアログ・ボックス

テキスト入力ボックスに検索のための文字列を入力し検索を始めるために<Find>を選ぶか、又は、中止する場合は<Cancel>を選んで下さい。

Direction[検索方向]ボックスは検索する方向を指定します。現在のカーソル位置から開始(編集モード)。 Forward[フォワード](現在の位置、又は、バッファの最初からバッファの終了)がデフォルトです。Backward[バック ワード]は始めに向かって検索します。ビュー・モードでは全バッファを検索します。

Origin[オリジン]は検索を開始する場所を指定します。

Find & Replace dialog box [テキストの検索と置換]ダイアログ・ボックス]

Text to find[検索のためのテキスト]文字列入カボックスに検索のための文字列を入力し、そして、 Replace with[に置換]入カボックスに置換のための文字列ストリングスを入力します。

Options [オプション]ボックスで置換のプロンプトを選択することができます:プログラムが例を検出した場合、 プログラムを変更する前に尋ねられます。

Originn[オリジン]は検索を開始する場所を指定します。

Direction[検索方向]ボックスは検索する方向を指定します。現在のカーソル位置から開始(編集モード)。 Forward[フォワード](現在の位置、又は、バッファの最初からバッファの終了)がデフォルトです。Backward[バック ワード]は始めに向かって検索します。ビュー・モードでは全バッファを検索します。

ダイアログ・ウィンドウを閉じるには <Esc> 又は、Cancel[キャンセル] ボタンをクリックします。

Replace[置換]ボタンを押しますとダイアログ・ボックスが閉じられ、そして、クエスチョン・ウィンドウが表示されます。このウィンドウは下記の選択を含んでいます:

Yes 見つかった項目の置換と次を検索

No 現在のものを置換ずに次を検索

Replace All 見つかった全ての項目を置換

Abort search このコマンドについて

View/Edit buffer for PLD. [ビュー/PLDのためのバッファ編集]

Ctrl+F2	指定したブランク値でバッファを消去
Ctrl+Shift+F2	ランダム・データでバッファをフィル
F9	アドレスヘ
F10	ビュー/編集のモードを変更
F11	バッファのデータ・ビューのモードで1ビットと8ビット・ビューの間を切替えます。
	View/Edit mode buffer indicator[ビュー/編集モード・バッファ・インディケータ]の右の
	ボタンをマウスでクリックすることでも行えます。このボタンは実際のデータ・ビュー・モード(1
	ビット又は、8ビット)も表示します。
Arrow keys	カーソル上下左右移動
Home/End	現在の行の開始/終了へジャンプ
PgUp/PgDn	前/次ページへジャンプ
Ctrl+PgUp/PgDn	現在のページの開始/終了へジャンプ
Ctrl+Home/End	編集エリアの開始/終了へジャンプ
Backspace	カーソルを1つ左(戻る)へ移動
ノート: 文字列()と1は	編集領域の内容を即座に変更



Buffer / Fill block [バッファ/ブロックのフィル]

このコマンドを選択することで要求したhex(又は、ASCII)文字列によりバッファの選択されたブロックをフィルします。

オプション"Allow address history logging"選択は最後に確認された値のセーブをアクティベートします。 これらは各デバイスで別個にセーブされ、カウントは最後の15個に制限されています。

ノート:アドレス履歴値は全てのバッファ・データ操作ダイアログで共通です。

デフォルト・アドレス範囲は選択されたデバイスのバッファ範囲に従ってセットされます。 選択オプション"Maintain last inserted values"は次回このダイアログを開いた時に前回確認された値が デフォルトとして再ロードされます。

Buffer / Copy block [バッファ/ブロックのコピー]

このコマンドは新しいアドレス上の現在のバッファのデータの指定されたブロックをコピーするのに使用されます。タ ーゲット・アドレスはソース・ブロック・アドレスの外である必要はありません。

Buffer / Move block [バッファ/ブロックの移動]

このコマンドは新しいアドレス上の現在のバッファのデータのブロックを移動するのに使用します。ターゲット・アド レスはソース・ブロック・アドレスの外側である必要はありません。ソース・アドレスのブロック(又は、一部分)は一 般的にブランク・キャラクターによりフィルされます。

オプション"Allow address history logging"選択は最後に確認された値のセーブをアクティベートします。 これらは各デバイスで別個にセーブされ、カウントは最後の15個に制限されています。 ノート: アドレス履歴値は全てのバッファ・データ操作ダイアログで共通です。

デフォルト・アドレス範囲は選択されたデバイスのバッファ範囲に従ってセットされます。 選択オプション"Maintain last inserted values"は次回このダイアログを開いた時に前回確認された値がデフ ォルトとして再ロードされます。

Buffer / Swap block[バッファ/スワップ・ブロック]

このコマンドはユーザーが選択したスワップ・モードに応じてバイトペアのハイ-とロー-の順番、バイト内の4ビット、 又は、ニブルをスワップします。スワップ操作は開始アドレスと終了アドレスで指定されたバッファ・ブロック行わ れます。このブロックは偶数アドレスで開始され、そして、偶数のバイトを持たなければいけません。もし、この 条件が満たされないと、プログラムはアドレス自身を修正します。(開始アドレスは下位の偶数アドレスに移 動されるか、又は、終了アドレスが上位の奇数アドレスに移動されます。

次のスワップ・モードが利用出来、ユーザーは選択することが出来ます:

- 1. Swap 2-bytes inside 16-bit words 16-bitワード内をバイトペアでスワップ
- 2. Swap 4-bytes inside 32-bit words 32-bitワード内の4バイトをスワップ
- 3. Swap nibbles inside bytes 各バイト内でハイ-とロー-のニブルをスワップ
- 4. Mirror bits inside bytes バイト内のビットを逆にします。

バッファ内のスワップ操作の例:

開始アドレス0から終了アドレスNまでのスワップ・バイト操作は次のテーブルによりバッファ内のデータを修正:



Address	Original Data	Swap 2-bytes inside 16-bit words	Swap 4-bytes inside 32-bit words	Swap nibbles inside bytes	Mirror bits inside bytes
0000h	b0	b1	b3	b0n	b0m
0001h	b1	b0	b2	b1n	b1m
0002h	b2	b3	b1	b2n	b2m
0003h	b3	b2	b0	b3n	b3m
0004h	b4	b5	b7	b4n	b4m
0005h	b5	b4	b6	b5n	b5m
0006h	b6	b7	b5	b6n	b6m
0007h	b7	b6	b4	b7n	b7m

b0, b1, b2 ... はアドレス0, 1, 2..からのオリジナル・バッファ・バイト値 b0n, b1n, b2n... は次のルールによるニブル・スワップされたオリジナル・バイト b0, b1, b2:

Original Byte bits	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Nibble-swapped Byte Bits	bit 3	bit 2	bit 1	bit 0	bit 7	bit 6	bit 5	bit 4
Original Byte bits	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Mirrored Byte Bits	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7

オプション"Allow address history logging"を選択しますと最後に確認された値のセーブをアクティベートします。 これらは個々のデバイスのために別々にセーブされ、カウントは最後の15項目に制限されています。 ノート: アドレス履歴値は全てのバッファ・データ操作ダイアログで共通です。

デフォルトのアドレス範囲は選択されたデバイスのバッファ範囲に従ってセットされます。

オプション"Maintain last inserted values"を選択しますと次回にこのダイアログを開いた時に以前に確認 された値がデフォルトとして再ロードされます。

Buffer / Erase [パッファ/イレース]

このコマンドを選択しますとバッファの内容をトピカル・ブランクでフィルします。

オプション"Allow address history logging"選択は最後に確認された値のセーブをアクティベートします。 これらは各デバイスで別個にセーブされ、カウントは最後の15個に制限されています。 ノート: アドレス履歴値は全てのバッファ・データ操作ダイアログで共通です。

デフォルト・アドレス範囲は選択されたデバイスのバッファ範囲に従ってセットされます。 選択オプション"Maintain last inserted values"は次回このダイアログを開いた時に前回確認された値がデフ ォルトとして再ロードされます。

<Ctrl+F2> はどのメニューにいてもこのメニューを呼び出すことが出来ます。

Buffer / Fill random data[パッファ/ランダム・データをフィル]

このコマンドを選択しますとバッファの内容をランダム・データでフィルします。

オプション"Allow address history logging"選択は最後に確認された値のセーブをアクティベートします。 これらは各デバイスで別個にセーブされ、カウントは最後の15個に制限されています。 ノート: アドレス履歴値は全てのバッファ・データ操作ダイアログで共通です。



デフォルト・アドレス範囲は選択されたデバイスのバッファ範囲に従ってセットされます。 選択オプション"Maintain last inserted values"は次回このダイアログを開いた時に前回確認された値がデフ ォルトとして再ロードされます。

<Shift+Ctrl+F2> はどのメニューにいてもこのメニューを呼び出すことが出来ます。

Buffer / Duplicate buffer content[パッファ/パッファ内容のコピー]

このコマンドはソースEPROMの範囲内のバッファ内容を宛先EPROMの範囲にコピーします。この手順は例 えば27C512 EPROMから27C256 EPROM位置に使用される場合に適しています。 ノート: 手順は常にバッファ開始アドレス00000hを使用

Buffer / Checksum [パッファ/チェックサム]

PG4UWのバッファにストアされたデータのチェックサムはバッファのデータが正しいかを照合するのに便利です。

PG4UW はチェックサムに関する次の機能を含んでいます:

• タブ Checksum calculator[チェックサム・カイキュレータ],これはバッファ内の各種のデータ・ブロックの各種チェックサムを計算し、そして、表示出来るオン-デマンドの checksum calculator[チェックサム・カイキュレータ] です。(*1)

 タブMain checksum options[メイン・チェックサム・オプション]はテーブル AddressとPG4UWのLog windowsでPG4UWのメイン・ウィンドウに表示されるメイン・チェックサム値を持ったAutomatic checksum calculator[自動チェックサムの計算]のためのオプションが含まれています。(*2)

Checksum calculator[チェックサム・カリキュレータ] はオン-デマンド・チェックサム・カリキュレータを含んでいます。(*1)

フィールド From address[アドレスから]とTo address[アドレスへ]がチェックサム計算のためのアドレス範囲の入力に使用されます。アドレスはチェック・ボックスEnabled[有効]にチェックが入っている時のみ使用することが出来ます。アドレスは常にバイト・アドレスとして定義されます。

グループ Exclude buffer block(s) from checksum calculation[チェックサム計算からパッファ・フロックを除外する]は例えばserialization[シリアライゼーション]には便利です。シリアライゼーションは通常はバッファの指定されたアドレスのデータを修正します。従ってシリアライゼーション・エンジンによってデバイスのプログラミングの前に或るアドレスのデータが変更された時はバッファのチェックサムのチェックに問題があります。シリアライゼーションに使用されたバッファ(データ・ブロック)の一部がチェックサム計算が除外された場合はバッファ・データのチェックサムはシリアライゼーションによって変更されません。1つまたは複数の除外ブロックを指定できます。

- 計算されたチェックサム・タイプの値を表示するフィールド:後述のタイプの説明をご覧下さい。
- STRAIGHTは追加の調整無しのチェックサム計算の結果
- NEGATEDはチェックサムの反転、従って SUM + NEG. = FFFFH.
- SUPPLEMENTはチェックサムの補数、従って SUM + SUPPL. = 0 (+ carry).
- Insert checksum options [チェックサムの挿入オプション]ボックス このボックスはCalculate & insert[計算と挿入]操作のための次のオプションを含みます:
 - Insert checksum[チェックサムの挿入] Calculate & insert[計算と挿入]操作が実行された時に バッファに書込まれるチェックサムの種類
 - Insert at address[アドレスの挿入] Calculate & insert[計算と挿入]が実行された時に選択されたチェックサムの結果を書込むバッファのアドレス。アドレスは <From address> から <To address>の範囲内で指定することが出来ません。



アドレスは常にバイト・アドレスとして定義されます。

• Size [サイズ] 選択されたチェックサム結果が書込まれるバッファのサイズ。チェックサムのサイズはByte(8-bit)、 Word(16-bit)、又は、DWORD(32-bit)です。選択されたチェックサム・サイズより小さい場合 は、チェックサム値の下位バイトのみがバッファに書込まれます。

ノート: もし、ワード・サイズが選択されますと、チェックサム値のロー・バイトがInsert address[アドレス挿入]ボックスで指定されたアドレスが書込まれ、そして、ハイ・バイトが1つず つインクレメントされたアドレスに書込まれます。DWORDに対しても同様です。

- Calculate button[計算ボタン] Calculateボタンをクリックしますと、バッファ内の選択されたブロックのチェックサムの計算します。バッファへの書込みは行われません。
- Calculate & insert button[計算と挿入ボタン] Calculate & insert ボタンをクリックしますと、バッファ 内の選択されたブロックのチェックサムを計算され、そして、選択されたチェックサムがInsert address[アドレス挿入]で指定されたアドレスでバッファに書込まれます。この機能はByte, Word, CRC-CCITTと CRC-XMODEMチェックサムで利用出来ます。
- ・ Close button[クロース・ボタン] ダイアログChecksum[チェックサム]を閉じます。

(*1) これらの値はプロジェクトに保存されません。それぞれの新しいデバイス選択でデフォルトに初期化されます。

タブ Main checksum options[メイン・チェックサム・オプション]は自動チェックサム計算のモードをセットす ることが出来ます。(*2)

- グループ Custom address range for main checksum[メイン・チェクサムのためのカスタム・アドレス]
- Enabled[有効にする] ユーザー定義アドレスがバッファのデータのチェックサムの計算に使用されます、他方、 Disabled[無効]の場合、バッファのデータのチェックサムの計算にグローバル・バッファ開始とバッファ 終了アドレ スが使用されます。
- フィールドFrom address[アドレスから] とTo address[アドレスまで] はメイン・チェックサム計算のアドレス 範囲の入力に使用されます。アドレスはチェックボックスEnabled[有効にする]にチェックが入っている時のみ使 用されます。
- 選択グループ Checksum type[チェックサム・タイプ]はメイン・チェックサムに使用する希望するチェックサムの種類を選択出来ます。詳しは下記のチェックサム・タイプをご覧下さい。
- ・フィールド Checksum[チェックサム] は最後に計算されたチェックサムの実際の値を含みます。
- グループExclude buffer block(s) from checksum calculation[チェックサムからパッファ・ブロック を除外] – はチェックサム計算のタブと同じです。
- ボタンApply[適用]は Main checksum options[メイン・チェックサム・オプション]からのチェックサム設定を確認するために使用します。ノート:一度ボタンが押されますと、前回のチェックサム設定は失われます。
- ボタンClose[閉じる] はチェックサム・ダイアログを閉じるために使用されます。もし、設定で変更を加えた場合、Apply[適用]を押すまで変更は反映されません。

(*2) それらの値はコフィギュレーション・ファイルとプロジェクト・ファイルにストアされます。 プロジェクト・ファイルからの 設定が優先されます。

Checksum types[チェックサム・タイプ]

Byte sum (x8)

バッファ・データは現在のバッファのビュー・モード(x8/x16/x1)構成に関係なくバイトごとに加算されます。32ビット を超えるキャリービットは無視されます。このチェックサム・モードでは文字列(x8)をメイン・プログラム・ウィンドウの チェックサム値の後に表示されます。

Word sum Little Endian (x16)

バッファ・データは現在のバッファのビューモードの構成に関係なくワード単位で加算されます。32ビットを超える 任意のキャリー・ビットは無視されます。このチェックサム・モードはメイン・プログラム・ウィンドウのチェックサム値の 後に表示され文字列(x16LE)よって示されます。リトル・エンディアンはバッファのチェックサムがリトル・エンディア ン・モードでバッファから読み出されワードから計算されます。



Word sum Big Endian (x16)

バッファ・データは現在のバッファのビューモードの構成に関係なくワード単位で加算されます。32ビットを超える 任意のキャリービットは無視されます。このチェックサム・モードはメイン・プログラム・ウィンドウのチェックサム値の後 に表示され文字列(x16BE)よって示されます。ビッグエンディアンはバッファのチェックサムがビッグ・エンディアンモ ードでバッファから読み出されワードから計算されます。

CRC-CCITT

バッファ・データは多項式 x¹⁶+x¹²+x⁵+1(0x1021)を使ってbyteをwordで加算,初期値 0,そして、 XOR out 0, reflexions in/out はoff.

CRC-XMODEM

バッファ・データは多項式 x^16+x^15+x^2+1(0x8005)を使ってbyteをwordで加算, 初期値 0.

CRC-16

バッファ・データは多項式x^16+x^15+x^2+1(0x8005)を持った標準CRC-16 アルゴリズムを使ってbyteを wordで加算, 初期値 0, そして、XOR out 0.

CRC-32

バッファ・データは多項式0x04C11DB7を持った標準CRC-32 アルゴリズムを使ってbyteをDWORDで加算, 初期値0xFFFFFFF, そして、XOR out 0Xfffffff.

MD5

MD5 hash は32桁の16進数のシーケンス(128 bits)で表示されます。

SHA-1

"Secure Hash Standard" は40桁の16進数のシーケンス(160 bits)で表示されます。)

Checksum forms[チェックサム形式]

Straight - 追加の調整無しのチェックサム Negated - チェックサムを反転 SUM + NEG. = FFFFH. Supplement はチェックサムの補数 SUM + SUPPL. = 0 (+ carry). Device dependent checksum[デバイス依存チェックサム] - いくつかのデバイスが適応されます。 例えば、STMicroelectronics's STM8ファミリー。メイン・チェックサムのチェックサム・モードはメイン・プログラ ムのラベル・チェックサム上でクリックすることでポップ・アップ・メニュー(又は、メニュー・ショートカット)でセットすること が出来ます。 Shift+Ctrl+1 - Byte sum (x8), Shift+Ctrl+2 - Word sum Little Endian (x16) Shift+Ctrl+3 - Word sum Big Endian (x16) etc... Wordは16-bit word. DWORD(は32-bit word.

Device[デバイス]

メニュー Device[デバイス]はデフォルトのリストから希望するデバイス・タイプを選択出来ます - デバイス選択, デバイスからのデータの読み出し, デバイスのブランク・チェック, プログラム, ベリファイとイレース

Device / Select from default devices[デバイス/デフォルト・デバイスから選択]

このウィンドウはデフォルト・デバイスのリストからデバイスのタイプを選択することが出来ます。これはデバイス・オプションで最後に選択されたデバイスにストアされる周期バッファです。このリストはFile / Exit and save[ファイル/終了 とセーブ]コマンドによりディスクに保存されます。



現在のデバイスの追加情報を表示したい場合は<**Ctrl+F1>**キーを使います。このコマンドはデバイスのサイズ、構成、プログラミング・アルゴリズムとこのデバイスがサポートされているプログラマー(追加モジュールを含む)のリストを提供します。パッケージ情報と他の一般的な情報もご覧頂けます。

デフォルト・デバイスのリストから現在のデバイスを削除するためには キーを使用します。このリストを空白に することはできません。最後のデバイスはバッファに残っておりますので、キーは受け付けられません。

Device / Select device... [テバイス/テバイス選択....]

このウィンドウは現在のプログラマによりサポートされている全てのデバイスのタイプを選択することが出来ます。デバイスを名前、タイプ又は、マニファクチャラーにより選択することが可能です。

ノート:

ソフトウェアでプログラマブル・デバイスの名前はチップの上部に表示されていたり、データシートのパーツ番 号の セクションに記載されている全てのキャラクターが含まれているわけではありません。名前はデバイスの識別に必要 な全てキャラクターが含まれていますが、プログラミングに影響がない例えば、温度コード、スピードコード、梱包タイ プコードは含まれていません。そのようなコード文字が名前の最後にある場合は省略され、 途中にある場合ばx'に置き換えられています。

例えば::

- デバイス Am27C512-150, Am27C512-200と27C512-250はソフトウエアではAm27C512と表示されます。
- S29GL064N11TF1010デバイスはソフトウェアではS29GL064NxxTxx01と表示されます。

もし、あるデバイスで2つ表示されていて2番目にサフィックスx16とある場合、それはプログラミング・アルゴリズ ムがより早いワード・モードを提供していることを意味します。

選択されたデバイスは自動的にデフォルト・デバイスのバッファにセーブされます。このバッファは Device/Select from default devices[デパイス/デフォルト・デバイスからの選択] コマンドからアクセスす ることが出来ます。

Search mask [検索マスク]フィールドではデバイス名、マニュファクチャラー及び/又は、プログラミング・アダプ タ名によるデバイス・リスト全体のフィルタリングのためのマスクを入力できます。フィルタ項目(フラグメント)の区 切り文字としてのスペースには"OR"機能があります。スペースを含む正確なフィルタ文字列を入力する場合 は引用符文字 "を使用してください。

サンプル:

アダプタを必要としないデバイスを確認する必要があり、そのようなデバイスにはデバイス・リストのア ダプタ列に次のメモ文字列があることがわかります。Note:in ZIF socket of programmer。 希望するデバイスのみを表示するのに適したフィルタは「ZIF内」です(引用符を含む)。フィルタ文 字列は大文字と小文字を区別しません。例えば、"ZIF"は"zif"と同じです。

もし、現在のデバイスについての追加情報を表示した場合は、ボタンDevice info、又は、 <Ctrl+F1> キーを使ってください。このコマンドはデバイスのサイズ、構成、プログラミング・アルゴリ ズムとこのデバイスがサポートされているプログラマ(追加モジュールを含む)のリストを提供します。パ ッケージ情報と他の一般的な情報もご覧頂けます。

現在表示されているデバイス・リストはボタン Save currently displayed list to file[現在表示され ているリストをファイルに保存]を押すことでテキスト・ファイルに保存することができます。



Select device ... / All[テバイス選択.../全部]

このウィンドウは現在のプログラマでサポートされている全てのデバイスからターゲット・デバイスのタイプを選択する ことが出来ます。サポートされているデバイスはリスト・ボックスに表示されます。

デバイスは希望のマニファクチャラー名とデバイス番号をリストの行でダブル・クリックするか、又は、希望のマニファ クチャラー名とデバイス番号をサーチ・ボックス(セパレート・キャラクターとしてキー<Space>を使用)で入力する、 そして、<Enter>を押すか、又は、OK ボタンをクリックすることで選択することが出来ます。

現在選択されているデバイスを反映せずにデバイス選択をキャンセルするには、いつでも、<Esc> キーを押すか、 又は、Cancel ボタンをクリックして下さい。

選択されたデバイスは自動的にデフォルト・デバイスのバッファにセーブされます。このバッファは Device/Select from default devices[デバイス/デフォルト・デバイスからの選択] コマンドからアクセスす ることが出来ます。

もし、現在のデバイスについての追加情報を表示した場合は、ボタンDevice info、又は、<Ctrl+F1> キー を使ってください。このコマンドはデバイスのサイズ、構成、プログラミング・アルゴリズムとこのデバイスがサポートさ れてるプログラマー(追加モジュールを含む)のリストを提供します。パッケージ情報と他の一般的な情報もご覧 頂けます。

Select device ... / Only selected type[デバイス選択.../選択されたタイプのみ]

このウィンドウはデバイスの希望するデバイス・タイプを選択することが出来ます。最初に、マウス、又は、カーソル・キーを使ってデバイス・タイプ(例、EPROM)を選択、そして、デバイスのサブ・タイプ(例、64Kx8 (27512))を選択して下さい。そうしますとマニファクチャラーのリストとデバイスが表示されます。

デバイスは希望のマニファクチャラー名とデバイス番号をリストの行でダブル・クリックするか、又は、希望のマニファ クチャラー名とデバイス番号をサーチ・ボックス(セパレート・キャラクターとしてキー<Space>を使用)で入力する、 そして、<Enter>を押すか、又は、OK ボタンをクリックすることで選択することが出来ます。

現在選択されているデバイスを反映せずにデバイス選択をキャンセルするには、いつでも、<Esc> キーを押すか、 又は、Cancel ボタンをクリックして下さい。

選択されたデバイスは自動的にデフォルト・デバイスのバッファにセーブされます。このバッファは Device/Select from default devices[デバイス/デフォルト・デバイスからの選択] コマンドからアクセスす ることが出来ます。

もし、現在のデバイスについての追加情報を表示した場合は、ボタンDevice info、又は、<Ctrl+F1> キー を使ってください。このコマンドはデバイスのサイズ、構成、プログラミング・アルゴリズムとこのデバイスがサポートさ れてるプログラマー(追加モジュールを含む)のリストを提供します。パッケージ情報と他の一般的な情報もご覧 頂けます。



Device / Select EPROM /Flash by ID[デバイス/IDによるEPROM選択]

このコマンドはデバイスIDを読むことでアクティブ・デバイスとしてEPROMを自動選択するのに使用します。プログラマはチップに焼き付けられているマニファクチャラーとデバイスIDを読むことでEPROMを自動的に認識します。これはこの機能をサポートしているEPROM、又は、Flashのみに適応されます。もし、デバイスがチップIDとマニファクチャーIDをサポートしていないときは UNKOWN又は、NOT SUPPORTED DEVICEであることを告げるメッセージを表示します。

他に一致したチップIDとマニュファクチャーIDが検知されますと、これらのデバイスのリストが表示されます。リストから その番号(又は、マニファクチャー名)を選ぶことで、このリストから対応デバイスを選択することが出来ます。そして、 <Enter>を押すか、又は、OK ボタンをクリックして下さい。現在選択されているデバイスを反映せずにデバイス選 択をキャンセルするときは、いつでも、<Esc>キーを押すか、又は、Cancel ボタンをクリックして下さい。

警告:

制御プログラムは28ピンと32ピンのEPROMとFlashのみをサポートします。どのプログラマもピン番号を自動的 に決定します。他のプログラマはこの番号を手動で入力する必要があります。

プログラマはソケットの相応のピンに高電圧を印加します。これはシステムがデバイスIDを読み取るために必要です。 EPROM、又は、Flash以外のデバイスをソケットに挿入しないでください。プログラマが高電圧を印加すると破損 することがあります。このコマンドを次のように適用することはお勧めしません。

- 2764と27128のEPROMタイプ。そのほとんどはIDがサポートしていないためです。
- 非標準のピン配置を有するFlashメモリ (e.g. Firmware Hub Flash)
- A9ピンでVid 電圧を受け付けないFlashメモリ
- 低電圧 EPROとFlashメモリ

Device / Device options[デバイス/デバイス・オプション]

このメニューの全ての設定はプログラミング・プロセス、シリアライゼーションと関連ファイルのコントロールに使用されます。

Device / Device options / Operation options [デバイス/デバイス・オプション/操作オ プション]

このコマンドのすべての設定はプログラミング・プロセスのコントロールに使用されます。これは現在のデバイスとプログラマのタイプに関連した項目を含むフレキシブルな環境です。項目は現在のデバイスに対しては有効ですが、現在のプログラマでサポートされていないものはディスエーブルになっています。これらの設定はFile / Exit and save[ファイル/終了と保存] コマンドにより関連デバイスと共にディスクにセーブされます。

ー般的に使用されている用語はプログラマのユーザー・マニュアルでも説明されています。ここで使用される特別な用語はそれぞれのチップの製造元が使用する用語です。使用する全ての用語の説明についてはプログラムしたいチップのドキュメントをお読みください。

一般的に使用される項目のリスト:

アドレス・グループ

 Device start address
 デバイス開始アドレス (デフォルト・0)

 Device end address
 デバイス終了アドレス (デフォルト・デバイス・サイズ-1)

 Buffer start address
 パッファ開始アドレス (デフォルト 0)

 Split[スプリット]
 (デフォルト なし)

このオプションはプログラミングやデバイスを読み取る際にバッファの特殊なモードを設定することが出来ます。16-bit、 又は、32-bitのアプリケーションを8bitのデータ・メモリ・デバイスに書き込みに使用する際に分割オプションを使用す ると特に有用です。



次の表はバッファからデバイスとデバイスからバッファのデータ転送を説明しています。

Split type	Device Buffer	Address assignment
None	Device [ADDR]	Buffer [ADDR]
Even	Device [ADDR]	Buffer [2*ADDR]
Odd	Device [ADDR]	Buffer [1+ (2*ADDR)]
1./4	Device [ADDR]	Buffer [4*ADDR]
2./4	Device [ADDR]	Buffer [1+(4*ADDR)]
3./4	Device [ADDR]	Buffer [2+(4*ADDR)]
4./4	Device [ADDR]	Buffer [3+(4*ADDR)]

実際のアドレスは次のようになります:(全てのアドレスは16進数です)

Split type	Device addresses	Buffer addresses
None	00 01 02 03 04 05	00 01 02 03 04 05
Even	00 01 02 03 04 05	00 02 04 06 08 0A
Odd	00 01 02 03 04 05	01 03 05 07 09 0B
1./4	00 01 02 03 04 05	00 04 08 0C 10 14
2./4	00 01 02 03 04 05	01 05 09 0D 11 15
3./4	00 01 02 03 04 05	02 06 0A 0E 12 16
4./4	00 01 02 03 04 05	03 07 0B 0F 13 17

用語説明:

デバイス・アドレス ADDR へのアクセスは Device[ADDR]として書き込まれます。 バッファ・アドレス ADDR へのアクセスは Buffer[ADDR]として書き込まれます。 ADDR 値は 0 からデバイス・サイズ(バイト)です。 全てのアドレスはバイト指向アドレスです。 グループ Insertion test[インサーション・テスト]:

Insertion test[装着テスト] (デフォルト ENABLE)

有効の場合、プログラマは ZIF ソケット(導通試験)との接続が正しいかチップの全てのピンをチェックします。プログラ マはチップの誤装着と逆挿しの接触不良を認識します。

Device ID チェック・エラーは操作を終了 (デフォルト ENABLE)

プログラマは各選択アクションの前に ID チェックを提供します。これはデバイスの製造業者によって定義された ID コードをデバイスから読み出した ID コードとを比較します。 ID エラーの場合、制御プログラムが次のように動作します:

•項目をENABLE[有効]にセットしている場合、選択された動作は終了します。

•項目を DISABLE[無効]にセットしている場合は、選択した動作が続行されます。制御プログラムは単に ID エラーに関する警告メッセージを LOG ウィンドウに書き込みます。

有効にした場合、プログラマはプログラムされるチップの電子 ID をチェックします。

ノート:

ある種の古いチップは電子ID機能を持っていません。一部の特殊なケースでは、制御プログラムのデバイスIDチェッ ク設定が「有効」に設定されていても、チップ内のコピー保護機能が設定されていればマイクロコントローラはIDを提 供しません。



グループ Command execution[コマンド実行]:

 プログラミング前にブランク・チェック(デフォルト DISABLE)

 プログラミング前にイレース
 (デフォルト DISABLE)

 読み込み後のベリファイ
 (デフォルト ENABLE)

 ベリファイ
 (ONCE, TWICE)

 ベリファイ・オプション
 (nominal VCC ±5%, nominal VCC ±10%, VCCmin VCCmax)

グループ Target system power supply parameters[ターゲット・システム電源供給パラメータ]

このグループは一部のタイプのデバイスの ISP モードで使用できます。次の設定が含まれています:

Enable target system power supply[ターゲット・システムの電源供給を有効にする] - プログラマからターゲット・システムへの電源供給を可能にします。プログラムされたデバイスとのアクションの前にターゲット・システムのための供給電源がスイッチオンされ、そして、アクションが完了した後にスイッチオフされます。もし、操作が有効にされた後に定義されたレベルでの ISP の信号を保持する場合、プル-アップ/プル-ダウン抵抗が無効にされた後、プログラマは電源供給のスイッチをオフにします。

Voltage[電圧] - ターゲット・システムへの供給電源。供給電圧は 2V~6V です。

ノート:ターゲット・システムに供給する電圧値はターゲット・システムへの電流に依存します。ターゲット・システムへの適切な電源供給に達すためには、適切な電圧と最大電流値が定義される必要があります。最大電流値はタ ーゲット・システムの実際の消費電流と同じで可能な限り正確でなければいけません。

Max. current[最大電流] - 電源供給されたターゲット・システムの最大電流消費。電流消費範囲は 0~ 300mA

Voltage rise time[電圧立ち上がり時間] - ターゲット・システム電源供給電圧の立ち上がり時間のスキュー・レートを決定します。(供給電圧をスイッチオン)

Target supply settle time[ターゲット供給セトル時間] - ターゲット・システムで供給した電圧が設定値で安定し、ターゲット・システムでプログラムされるデバイスでの動作が準備される時間を決定。

Voltage fall time[電圧立ち下がり時間] - ターゲット・システム電源供給電圧の立ち下がり時間のスキュー・レートを決定します。(供給電圧をスイッチオフ)

Power down time[/パワー・ダウン時間] - ターゲット・システム内でターゲット・システムの電源供給がスイッチオフ された後、残留供給電圧を維持する時間(例えば充電されたコンデンサから)決定。この時間の経過後ターゲッ ト・システムは供給電源なしでプログラマから安全に切り離すことが出来ます。

グループTarget system parameters[ターゲット・システムのパラメータ]

このグループはある種のデバイスのタイプの ISP モードで利用出来ます。次のセッティングを含みます: Oscillator frequency (in Hz)[オシレータ周波数(in Hz)] - デバイス(ターゲット・システム)のオシレータの周波数。 コントロール・プログラムはそれによってプログラミング速度をセットしますので、正しい値をセットする必要があります。

Supply voltage (in mV) 供給電圧(in mV) - ターゲット・システム側の供給電源。制御プログラムは全てのアク ションの前にターゲット・システムで入力された供給電圧をチェック。又は、セット(プログラマの種類によって異なりま す)します。



Disable test supply voltage[供給電圧のテストを無効にする] - デバイスでの動作の前に Supply voltage edit[供給電圧エディット]ボックスでセットしたプログラムされたデバイスの供給電圧の測定とチェックを無効にします。

Delay after reset active[リセット・アクティブ後の遅延] - このパラメータはデバイスで動作を開始するためにリセット信号アクティブ後の遅延を決定します。この遅延はデバイスのリセット回路で使用されるデバイスの値に依存し、そして、次の値から選択することが出来ます:10ms, 50ms, 100ms, 500ms 又は、1s.

Inactive level of ISP signals ISP[信号の待機レベル] - このパラメータはターゲット・デバイスへのアクセス終了 後 ISP 信号のレベルを決定します。ISP コネクターの信号はプル-アップ(信号が 22k 抵抗を経由して供給電圧に 接続されている)、又は、プル-ダウン(信号が 22k 抵抗を経由してグランドに接続されている)にセットすることが出 来ます。

Keep ISP signals at defined level after operation[操作後定義されたレベルで ISP 信号を保持] - ター ゲット・デバイスにアクセス終了後に ISP 信号のセットレベルを保持します。制御プログラムは警告ウィンドウを表示 することでアクティベートされたプル-アップ/プル-ダウン抵抗を示します。ユーザーがこのウィンドウ制御プログラムを閉 じますと抵抗を無効化します。

グループ プログラミング・パラメータ

このグループはある種のデバイス・タイプに利用出来ます。プログラムされるデバイスのブロック、又は、領域、チップの プロテクトのセッティングを含みます。

グループ イレース・パラメータ

このグループはある種のデバイス・タイプに利用出来ます。選択されたデバイスのイレース・モードのセッティングを含みます。

Device / Device options / Serialization[デバイス・オプション/シリアライゼー ション]

シリアライゼーションはプログラムの特殊なモードです。シリアライゼーション・モードがアクティブな時、各デ バイスにプログラミングする前に指定した値が自動的にバッファの前もって定義されたアドレスに挿入され ます。そして、次から次へとデバイスをプログラムする時、自動的にシリアル番号の値が.変更してデバイス のプログラミングの前にバッファに挿入されます。従って、各々のデバイスが特有のシリアル番号を持つこと が出来ます。

シリアライゼーションには3つのタイプがあります:

- ・インクレメンタル[増加]・モード
- ファイルからのモード
- カスタム・ジェネレーター・モード

ダイアログ Serialization[シリアライゼーション] はシリアライゼーションがオンにされた場合にプロジェクト・ファイ ルで使用される関連したシリアライゼーション位置ファイルのための設定も含まれています。さらに詳しくは"シリア ライゼーションとプロジェクト"をご覧下さい。



シリアライゼーションのペーシック・ルール:

シリアライゼーションは最近選択されたデバイスにのみ関連付けられます。新しいデバイスを選択すると、シリアライゼーション設定がリセットされます(シリアライゼーションは無効に設定されます)。

最近のデバイスのシリアライゼーション設定はデバイスのプロジェクト・ファイル、又は、アプリケーションが閉じられたときのコンフィギュレーション・ファイルと他の設定と共に保存されます。

シリアライゼーション・エンジンは各デバイス・プログラミングが開始される前に新しい(次の)シリアル番号を要求します(ノート1を参照)。

 ・
 使用されたシリアル番号はシリアル番号の値の後に(*)で示されます。シリアル番号を使用すると、次のデバイス・
 プログラミングは次のシリアル番号を使用します

ノート 1): オプションのSerial number usage if programming action fails[プログラミング動作が失敗 した場合のシリアル番号の使用]により以前のデバイス・プログラミングの結果が失敗した場合、プログラミング前 に新しいシリアル番号要求を呼び出すことを抑制できます:

- Reuse generated serial number for next programmed device[次のプログラムされるデ バイスのために生成されたシリアル番号を再使用]オプションが選択されている場合、以前のデバイス操作結果 が失敗した場合に新しい(次の)シリアル番号の要求が抑制されます。つまり、使用されたシリアル番号が再び 使用され、正常なデバイス・プログラミングが完了するまで同じシリアル番号が使用されます。
- Throw away (use the serial number only once, regardless result of the programming) Throw away[捨てる](プログラミングの結果に関係なくシリアル番号を1回だけ使用する)が選択された場合、以前のプログラミング操作の結果に関係なく、各デバイス操作の前に新しいシリアル番号の要求が実行されます。

シリアライゼーションはある種のデバイスのタイプに対してはPG4UWコントロール・プログラムのメイン・バッファ、又は、 使用可能な拡張バッファを操作することが出来ます。例えば、データEEPROMメモリーを搭載したマイクロチップ PIC16FXXX等、一部のタイプのデバイスで使用可能な拡張バッファで動作します。どのバッファをシリアライゼーシ ョン・ルーチンにより使用するかはダイアログ Serialization[シリアライゼーション]で選択可能です。Buffer[バッファ] 設定ボックスが表示されていない場合、現在選択のデバイスのシリアライゼーション・モードは拡張バッファをサポー トしていません。

Device / Device options / Serialization / Incremental mode & SQTP 「デバイス」デバイス・オプション/シリアライゼーション/インクレメント・モード & SQTP1

Incremental mode & SQTP[インクレメンタル・モードとSQTP]は各プログラム・デバイスに個別のシリアル番号を割り当てることが出来ます。各デバイスのプログラム操作に対してユーザーにより入力された開始番号が指定されたステップで増加され、そして、各デバイスのプログラミングに先立ち、選択されたフォーマットで指定されたバッファ・アドレスにロードされます。

インクレメンタル・モードのためにユーザーが修正することが出来るオプションには以下の項目があります:

S / N size[S /N サイズ]

S/Nサイズ・オプションはバッファに書込まれるシリアル値のバイトの数を定義します。S/NサイズではBin(バイナリー) シリアライゼーション・モードの値は1-8が有効で、そして、ASCII シリアライゼーション・モードでは1-16の値が有効 値です。

Addres[アドレス]

アドレス・オプションはシリアル値が書込まれるバッファ・アドレスを指定します。アドレス範囲はデバイスの開始と終 了のアドレスの範囲内でなければいけません。アドレスはシリアル値の最後(最上位、又は、最下位)バイトがデバ イスの開始と終了のアドレス範囲の中に指定されなければいけませんので、正しく指定されなければいけません。



Start value[スタート値]

スタート値オプションはシリアライゼーションが開始されるイニシャル値を指定します。一般的にシリアライゼーションの 最大値は32bit long wordで\$1FFFFFです。実際のシリアル値が最大値を超えた場合は、シリアル番号の3 つの最上位ビットがゼロにセットされます。このアクションの後、数値は常に0...\$1FFFFFFの間隔内にあります (これはオーバーフロー処理の基本スタイルです)。

Step[ステップ]

ステップ・オプションはシリアル値のインプリメンテーションの増加ステップを指定します。

S/N mode[S/N モード]

S/Nモード・オプションはバッファに書込まなければいけないシリアル値の形式を定義します。2つのオプションが利用 できます:

- ASCII
- Bin

ASCII - シリアル番号がASCII文字列としてバッファに書込まれることを意味します。例えば、番号\$0528CDは ASCIIモードで 30h 35h 32h 38h 43h 44h ('0' '5' '2' '8' 'C' 'D') としてバッファに書込まれます, 即ち、6バ イトです。

Bin - シリアル番号が直接バッファに書込まれることを意味します。もし、シリアル番号が1バイト長以上の場合, 2 つの可能なバイト・オーダーの1つに書くことができます。バイト・オーダーはSave to buffer[バッファにセーブ]項目で 変更することが出来ます。

Style[スタイル]

スタイル・オプションはシリアル番号ベースを定義します。2つのオプションがあります:

- Decimal [デシマル 10進数]
- Hexadecimal[ヘキサデシマル 16進数]

Decimal[デシマル]番号は'0' から '9'のキャラクターを使って入力と表示がされます。Hexadecimal[ヘキサ・デ シマル]番号は'A' から'F'のキャラクターを使います。

特別なケースはBinary Dec[バイナリー・デシマル]で、これはBCD番号スタイルを意味します。BCDはデシマル 番号がヘキサ・デシマル番号にストアされることを意味します、即ち、各ニブルが0から9までの値を持たなければ いけません。AからFの値はBCD番号のニブルとしては使用出来ません。 シリアル開始値とステップの数字を入れるまえに"Style"[スタイル]オプションでベースを選択して下さい。

Save to buffer[バッファヘセーブ]

Save to buffer[パッファにセーブ]オプションはバッファに書込むためのシリアル値のバイト・オーダーを指定します。 このオプションはBin S / N モード(ASCII モードには役立ちません)に対して使用されます。 2つのオプションが利用出来ます:

• LSByte first(インテルのプロセッサーで使用されています)はシリアル番号の最下位バイトを最初にバッファの最下位アドレスに置きます。

• MSByte first(モトローラのプロセッサーで使用されています)は最上位バイトを最初にバッファの最下位アドレスに置きます。

Split serial number[スプリット・シリアル番号]

オプションはシリアル番号を個々のバイトに分割し、そして、バッファの各N番目のアドレスにバイトを配置することが出来ます。この機能はデバイスのシリアル番号をRETLW、又は、NOP命令のグループとしてプログラム・メモリの一部とすることが出来る時、マイクロチップ社のPICデバイスのためのSQTPシリアライゼーション・モードのために特に有用です。詳細については以下のサンプルのサンプル2を参照して下さい。

次のスプリット・オプションが利用可能:

- ・ チェック・ボックス "Split serial number" –スプリット機能のターンオン/オフ
- Split gap スプリット・シリアル番号のフラグメント間に置くバイト数を指定
- S/N fragment size-シリアル番号はこのオプションにより指定されたサイズでフラグメントに分割されます。



サンプル:

サンプル 1:

アドレス7FFFAHでAT29C040デバイスにシリアル番号を書く、シリアル番号のサイズは4バイト、開始 値は16000000H、インクリメンタル・ステップは1、シリアル番号の形式はバイナリ、そして、最下位バイ トはデバイスのシリアル番号の下位アドレスに配置されます。

上記に記載のシリアライゼーションを作成するにはシリアライゼーション・ダイアログで次の設定をする必要があり ます:

サンプル 2:

次のサンプルはシリアル番号がマイクロチップPIC16F628デバイスに対してRETLW命令にスプリットされる時の SQTPシリアライゼーション・モードの使用方法を示します。

ノート:シリアル・クイック・ターン・プログラミング(SQTP)はマイクロチップ社のPICマイクロコントローラのシリアル・プロ グラミングのために、マイクロチップ社に指定された標準方式です。マイクロチップPICデバイスを使用すると各マイク ロコントローラに固有のシリアル番号をプログラムすることができます。この数はエントリコード、パスワード、又は、ID 番号として使用することが出来ます。

シリアライゼーションはリテラル・データとして、シリアル番号のバイトで、RETLW(リターンリテラルW)命令を連続使 用して行われます。シリアライズするには、インクリメンタル・モードのシリアライゼーション、又は、From file modeシ リアライゼーションを使用することが出来ます。

インクリメンタル・シリアライゼーションはシリアル番号を分割するSplit機能を提供します。シリアル番号分割機 能は偶数又は奇数バイトに分割された増加数を使用することが出来、シリアル番号の各バイトの間に RETLW命令コードが挿入されます。

"From file"シリアライゼーションは独自のシリアル番号ファイルを使用しています。このファイルは色々なシリアル番 号で構成することが出来ます。この番号はSQTPに適した形式を持つことが出来ます。たとえばRETLW b1 RETLW b2等です。注意:PG4UWのシリアル・ファイル形式はマイクロチップ社のMPLABによって生成される SQTPシリアル・ファイルとは互換性がありません。


Example 2a

Microchip PIC16F628デバイスに対してシリアライゼーションの分割を使用するとRETLW命令で分割します。

PIC16F628は14ビット幅命令ワードを持っています。RETLW命令は14ビット・オペコードを持っています:

説明 MSB 14-Bit word LSB RETLW リテラルをWで返す 11 01xx kkkk kkkk

xxは00に置き換えることが出来、そして、kはデータ・ビット、即ち、シリアル番号バイト

RETLW命令のオペコードはKKはデータ・バイト(シリアル番号バイト)であるヘキサデシマル 34KKH です。

4つのRETLW命令の一部としてシリアル番号1234ABCDHをデバイスPICに書き込むと仮定しましょう。シ リアル番号の最も高いバイトが最上位バイトです。アドレス40Hのデバイス・プログラム・メモリにシリアル番号 を書きたいとします。シリアル番号はこの状況で非常に便利に分割しました。シリアル番号分割のないシリア ライゼーションは次の番号をバッファとデバイスに書き込みます:

ノート: アドレス80Hはバッファがバイト構成を持っており、PICはワード構成を有していますので、プログラム・メモリのアドレス40Hと同等であるためです。バッファがワード構成 x16を持っている場合、アドレスは40Hと番号 1234ABCDHは次のようにバッファに配置されます:

アドレス データ 0000040 ABCD 1234 xxxx xxxx xxxx xxxx xxxx xxxx xxxx

RETLW命令を使いたいと仮定しますとバッファは: アドレス データ 0000040 34CD 34AB 3434 3412 xxxx xxxx xxxx xxxx

これは次のステップで行うことが出来ます:

A) アドレス40Hに4つのRETLW命令をメイン・バッファに書き込みます(これは手作業によるバッファの編集、又は、適切な内容のファイルをロードすることによって行うことができます)。各RETLW命令の下位8ビットは重要ではありません、それは各RETLW命令の下位8ビットには、シリアライゼーションの正しいシリアル番号のバイトが書き込まれる為です。

デバイスのプログラムを開始する前のバッファの内容は例えば以下の様に見えます:

各RETLW命令の8ビットはゼロです。それらは如何なる値も持つことが出来ます。

B) 次のようなシリアライゼーション・オプションをセット:

S/N size:	4 Bytes
Address:	40H
Star value:	1234ABCDH
Step:	1
S/N mode:	BIN
Style:	HEX

Save to buffer:LS Byte firstSplit serial number:checkedSplit gap:1 byte(s)S/N fragment size:1 byte(s)上述のスプリット設定はシリアル番号をバイト単位で分割して2バイト毎にバッファすることを意味します。正しいシリアル番号はデバイス・プログラミング動作が開始する前に厳密に設定されます。

最初のデバイスがプログラミングされる時のシリアル番号のバッファ内容は: アドレス データ 0000040 34CD 34AB 3434 3412 xxxx xxxx xxxx xxxx

2番目のデバイスは: アドレス データ 0000040 34CE 34AB 3434 3412 xxxx xxxx xxxx xxxx

次のデバイスは同じフォーマットのシリアル番号を持ち、各デバイスに対して1でインクリメントされます。

Example 2.b

Microchip PIC24FJ256デバイスのためのNOP命令を持ったシリアライゼーション・スプリットの使用

デバイス PIC24FJ256は24ビット幅の命令ワードを持っています。NOP命令はコード00xxxxhを持っています。 Microchip MPLAB®で指定されているSQTPシリアライゼーションと同じ方法のシリアライゼーションを使用すると します:

次のステップでこれを行うことが出来ます:

A) PG4UWのメイン・バッファのアドレス800hにNOP命令(00xxxxh)を書き込みます。これは編集バッファを手動、 又は、正しい内容を持ったファイルをロードすることによって行うことが出来ます。PG4UWバッファ内のアドレス800h はPIC24Fxxxプログラム・メモリのアドレス200hと同等です。詳細についてはPG4UWのPIC24FJ256デバイス用 のデバイス情報を見て下さい。

デバイスのプログラムを開始する前のアドレス800hのNOPでのバッファの内容は例えば以下の様に見えます:

xx - はバイト値を意味します。

B) 次のようなシリアライゼーション・オプションをセット: S/N

size:	3 bytes
Address:	800h
Start value:	123456h
Step:	1
S/N mode:	BIN
Style:	HEX
Save to buffer:	LS byte first
Split serial numbe	r: checked
Split gap:	2 byte(s)
S/N fragment size	: 2 byte(s)

上述のスプリットの設定はフラグメント間の2バイトのギャップで16ビット(2バイト)サイズのフラグメントにシリアル 番号のスプリットをバッファします。正しいシリアル番号はデバイスのプログラミング操作が開始される前にしっかり と設定されます。 最初のデバイスがプログラミングされる時のシリアル番号のバッファ内容は:



アドレス データ 0000800 56 34 00 00 12 00 00 00 xx xx xx xx xx xx xx xx xx xx

2番目のデバイスは:

アドレス データ

0000800 57 34 00 00 12 00 00 00 xx xx xx xx xx xx xx xx xx

次のデバイスは各デバイスに対して1でインクリメントされる同じフォーマットのシリアル番号を持ちます。

Example 3:

次のサンプルはシリアル番号スプリット・ギャップが2と3に設定されている代わりサンプル2aと同じシリアライゼーション・オプションを使用しています。

スプリット・ギャップが2バイトにセットされている時、バッファ内容は次のように見えます: バイト・バッファ構成: アドレス データ 0000080 CD xx xx AB xx xx 34 xx xx 12 xx xx xx xx xx xx

ワード16 バッファ構成: アドレス データ

0000040 xxCD ABxx xxxx xx34 12xx xxxx xxxx xxxx

スプリット・ギャップが3バイトにセットさている時、バッファ内容は次の様に見えます:

Bバイト・バッファ構成: アドレス データ 0000080 CD xx xx xX AB xx xx xx 34 xx xx xx 12

ワード16 バッファ構成:

アドレス データ

0000040 xxCD xxxx xxAB xxxx xx34 xxxx xx12 xxx

ノート: シリアライゼーション・オプションの効果がわからない時は、バッファに書き込まれる実際のシリアル番号をテストすることが可能です。テストは次のステップで行うことが出来ます:

1.ダイアログ"シリアライゼーション"で希望するシリアライゼーションを選択し、OKボタンで確認します。

2.ダイアログ"デバイス操作オプション"でインサーション・テストとDevice IDチェックを無効にします。

3.ZIFソケットにデバイスが装着されていないことを確認して下さい。

4.デバイス・プログラム操作を実行(ある種のデバイスではプログラミングの開始前にプログラミング・オプションを選択す る必要があります)

5. プログラミング操作が終了後(殆どの場合、デバイスが装着されていませんのでエラーとなります)、どこにシリアル番号が置くかはアドレスのメイン・バッファ(View/Edit/バッファ)で見て下さい。

J-ト: シリアライゼーションのためのアドレスは常に制御プログラムが現在のデバイスに使用している実際のデバ イスとバッファの構成に対して割り当てられます。もし、バッファ構成がバイトorg (x8)であれば、シリアライゼーショ ン・アドレスはバイト・アドレス。もし、バッファ構成がバイトより広い、即ち、16ビット・ワード(x16)ならシリアライゼ ーション・アドレスはワード・アドレスになります。



Device / Device options / Serialization / Classic From file mode 「デバイス/デバイス・オプション/シリアライセーション/クラシック・フロム・ファイル・モード」

Classic From-file mode[クラシック・フロム・ファイル・モード]を使用する場合、シリアライゼーション・フ ァイルにはシリアル値が直接含まれています。シリアライゼーション・データはシリアライゼーション・ファイルか らファイルに指定されたアドレスのバッファに直接読み込まれます。 クラシック・フロム・ファイル・モードは PG4UWコントロール・プログラムのメイン・ウインドウとi情報ウインドウに"From File"シリアライゼーション としてパネル"Serialization"で表示されます。

2つのユーザー・オプションがあります:

Start label[スタート・ラペル]

開始ラベルは入力ファイルの開始ラベルを定義します。ファイルからのシリアル値は定義された開始ラベル から読み取りを開始します。

ファイル名

Classis from file[クラシック・フロム・ファイル]のためのシリアライゼーションの入力ファイルは正しい形式でなければ いけません。

ファイル・フォーマット

クラシック From-fileシリアライゼーション入力ファイルはテキスト形式です。このファイルはバッファ・アドレスとバッファに 書き込むデータを定義するバイトのアドレスと配列が含まれます。入力ファイルにはテキスト形式の形式があり、そ の構造は次のとおりです:

[label1] addr byte0 byte1 .. byten ... [labeln] addr byte0 byte1 .. bytem , addr byte0 byte1 ... bytek ベーシック・パート オプショナル・パート

; Comment[コメント]

意味は:

ペーシック・パート

基本部分はバッファ・アドレスとバッファに書き込むバイトの配列を定義します。基本部分は常にラベルの行の後 に定義する必要があります。

オプショナル・パート

オプショナル部分は2番目のバイトの配列とバッファに書き込むバッファ・アドレスを定義します。オプショナル部分の 一部はデータの基本部分の後に定義することができます。

label1. labeln – ラベル

ラベルは入力ファイルの各行の識別子です。これらはファイルの各行のアドレス指定に使用されます。ラベルはユニ ークでなければいけません。ファイルの行をアドレス指定するとは、ユーザーが入力する必要な開始ラベルはシリアル 値の読み込みを開始する入力ファイルでの行を定義します。

addr -

Addrはアドレスに続くデータを書き込むバッファ・アドレスを定義します。

byte0..byten, byte0..bytem, byte0..bytek -

バイト配列byte0..byten、byte0..bytemとbyte0..bytekはバッファに書き込むために割り当てられるデータを定 義しています。アドレスに続く1つのデータ・フィールドの最大バイト数は64バイトです。データ・バイトはアドレスaddr からaddr+nまでのバッファに書き込まれます。



特定のバイトをバッファに書き込むプロセスは次のとおりです: byte0 to addr byte1 to addr + 1 byte2 to addr + 2

byten to addr + n

Optional part[オプショナル部分]は最初のデータ部分から文字","(カンマ)で区切られ、その構造は最初の データ部分と同じです。即ち、アドレスとそれに続くデータバイトの配列です。

特別使用の文字:

[]- ラベルは角括弧の中に定義する必要があります。

',' – データのベーシック・パート[基本部分]とオプショナル部分を区切る文字

';' - セミコロン文字はコメントの先頭を意味します。" ; " から行末までの全ての文字は無視されます。コメントは 個々の行、又は、定義行の最後に置くことができます。

ノート:

- ラベル名は「['と']'を除く全ての文字を含めることができます。ラベル名は大文字と小文字を区別しないように分析されます。即ち、文字 'a'は 'A'と同じで 'b'は 'B'と同じです。
- 入力ファイルの全てのアドレスとバイト番号の値は16進数です。
- 許容されるアドレス値のサイズは1~4バイトです。
- 1行のデータ配列の許容サイズは1から64バイトの範囲です。1行に2つのデータ配列がある場合、それらのサイズの合計は最大80バイト迄です。
- ・ 正しいアドレスをセットするように注意してください。アドレスはデバイスの開始アドレスとデバイス終了アドレスの範囲内で定義する必要があります。アドレスが範囲外の場合、警告ウィンドウが表示されシリアライ ゼーションは無効にセットされます。
- シリアライゼーションのためのアドレスは制御プログラムが現在のデバイスに使用している実際のデバイス構成とバ ッファ構成に常に割り当てられます。バッファ構成がバイト構成の場合(x8)、シリアライゼーション・アドレスはバイ ト・アドレスになります。バッファ構成がバイトよりも広い場合、例えば、16ビットワード(x16)の場合、シリアライゼ ーション・アドレスはワード・アドレスになります。

Classic From file[クラシック・フロム・ファイル]シリアライゼーションの典型的な入力ファイルの例]:

[nav1] A7890 78 89 56 02 AB CD ; comment1

- [nav2] A7890 02 02 04 06 08 0A
- [nav3] A7890 08 09 0A 0B A0 C0 ; comment2
- [nav4] A7890 68 87 50 02 0B 8D
- [nav5] A7890 A8 88 59 02 AB 7D

;次の行には2番目の定義も含まれます [nav6] A7890 18 29 36 42 5B 6D, FFFF6 44 11 22 33 99 88 77 66 55 16

;これは最後の行です-ファイルの最後

この例のファイルでは labels "nav1", "nav2", …"nav6"の6つのシリアル値が定義されています。各値はアドレス\$A7890のバッファに書き込まれます。 全ての値のサイズは6バイトです。 "nav6"ラベルの行はまたアドレス \$FFFF6にバッファリングされサイズが10バイトである第2の定義値を持っています。即ち、この値の最後のバイトは アドレス\$FFFFFに書き込まれます。

ノート: シリアライゼーションのアドレスは制御プログラムが現在のデバイスに使用している実際のデバイス構成とバッファ構成に常に割り当てられます。バッファ構成がバイト構成の場合、(x8)、シリアライゼーション・アドレスはバイト・アドレスになります。バッファ構成がバイトよりも広い場合、例えば、16ビットワード(x16)の場合、シリアライゼーション・アドレスはワード・アドレスになります。



Device / Device options / Serialization / Playlist From file mode 「デバイス/デバイス・オプション/シリアライゼーション/ファイルからのモート」

Playlist From-file modelプレイリスト・ファイルからのモード1を使用するとシリアライゼーション・ファイルは含まれ るシリアル値を直接は持っていません。ファイルはシリアライゼーション・データが含まれている外部ファイルの名前 のリストが含まれています。シリアライゼーション・データはこれらの外部データ・ファイルから読み出され、各ファイル は1つのシリアライゼーション・ステップ(1つのデバイスがプログラムされる)を意味します。Playlist From-file mode[プレイリスト・ファイルからのモード]はPG4UW制御プログラムのメイン・ウィンドウと情報ウィンドウに "From-file-pl[プレイリスト・ファイルからのモード]"シリアライゼーションとして"Serialization[シリアライゼーション]" パネルに表示されます。

ファイル・フォーマット

From-file[ファイルから]シリアライゼーション・プレイリスト・ファイルはシリアライゼーション・データを持ったファイル名の リストを含みます。そのファイル・フォーマットはクラシック・シリアライゼーション・ファイル・フォーマットに似ています。ファ イル・フォーマットの違いはプレイリスト・ファイルにおいては次の通りです:

- 1. playlistファイルはファイルの最初に空白行でない特別なヘッダを持つ必要があります。そのヘッダ行のフォーマッ トはテキスト形式です。 FILETYPE=PG4UW SERIALIZATION PLAYLIST FILE
- 2. 各シリアル・データ・バッチは別の行で次のフォーマットで表わされます。 [label x] datafilename

labelx - ラベルを現わします。

ラベルは入力ファイルの各行が空白でない事を示すための識別子です。これらはファイルの各行をアドレス指定す るために使用されます。ラベルはファイル内でユニークである必要があります。ファイルの行のアドレス指定はユーザー により入力された必要な開始ラベルがシリアル値の読み込みを開始する入力ファイルの行を定義することを意味し ます。

datafilename[データ・ファイル名] - シリアライゼーション・データを含むデータ・ファイルの名前を定義します。シリ アライゼーションが新しいシリアル値を必要とする場合、データ・ファイルは標準のPG4UW "Load file[ロードファイ ルリ"の手順で、PG4UWのバッファヘロードされます。ファイル形式はバイナリー又は、ヘキサ・ファイル(Intel Hex等) に対応しています。自動認識システムは適切なファイル形式を認識し、そして、正しいファイル形式のファイルのロー ドを行います。データ・ファイル名はペアレント(playlist)のシリアライゼーション・ファイルと関連しています。

playlist シリアライゼーション・ファイルのサンプル:

;---- 次のファイル・ヘッダーが必要です。-------FILETYPE=PG4UW SERIALIZATION PLAYLIST FILE

:----- シリアライゼーション・データ・ファイルの参照 [nav1] file1.dat [nav2] file2.dat [nav3] file3.dat

[label n] filex.dat ;----- end of file --

シリアライゼーション・タイプ From-file playlistのより詳細で完全に機能するサンプル例については、次のように PG4UWインストレーション・ディレクトリのExamples\subdirectoryにあるサンプル・ファイルを参照してくださ 1.1.

<PG4UW inst dir>\Examples\Serialization\fromfile playlist example\



一般的なパスは以下の様になります:

C:\Program Files (x86)\Elnec_sw\Programmer\Examples\Serialization\ fromfile_playlist_example\

次のステップでシリアライゼーションをテストすることが出来ます:

1. PG4UWを実行

2. ELNECプログラマが接続されて正しくPG4UWで認識されている必要があります。

3. 希望するデバイスを選択、イレース可能なメモリ・デバイスをお薦めします。(OTPメモリではありません)

- 4. Device | Device Options | Serializationメニューからダイアログを選択
- 5. パネルFrom-file modeオプションでFrom-file modeをセットしサンプルのシリアライゼーション・ファイル fromfile_playlist.serを選択して下さい。
- 6. 新しいシリアライゼーションの設定を受け付けるためにOKボタンをクリックします。
- 7. デバイス操作で "Program[プログラム]"を実行して下さい。

PG4UWのメイン・ウィンドウでシリアライゼーションがラベルを表示し、またデバイスのプログラミング中とプログラミングのリピートを情報プログレス・ウィンドウで見ることが出来ます。

使用されたファイルで追加の操作

このグループ・ボックスには操作の3つのタイプが含まれています。ユーザーは"Playlist From-file mode"で使用されたシリアライゼーション・データ・ファイルの操作の1つを選択することが出来ます。次の操作が利用可能:

- option Do nothing プログラムは使用されたシリアライゼーション・データ・ファイルでいずれの操作も行いません。
- option Move used file to specified directory
 プログラムは使用されたシリアライゼーション・データ・ファイルをユーザー指定の使用されたシリアライゼーション・ファ イルのディレクトリに移動します。
- ・option Delete used file プログラムは使用されたシリアライゼーション・データ・ファイルを削除します。

ディレクトリ

このオプションは"playlist From-file"シリアライゼーション・モードでオプション"Move used file to specified directory[指定されたディレクトリに使用するファイルを移動]"が選択されますと利用出来ます。ユーザーがどのシリアライゼーション・データ・ファイルに移動するかのターゲット・ディレクトリーを指定することが出来ます。

次のエラー表示がPlaylist From-fileシリアライゼーションで使用されます:

- s/n error #3 シリアライゼーション・データ・ファイルは存在しません。
- s/n error #34 使用されたシリアライゼーション・データ・ファイルを削除出来ません(シリアライゼーション・ファイルは書き込みプロテクト・ディスクに置かれているかも知れません)
- s/n error #35 使用されたシリアライゼーション・データ・ファイルを使用されたシリアライゼーション・ファイルのター ゲット・ディレクトリーへ移動出来ません(シリアライゼーション・ファイルは書き込みプロテクト・ディ スクに置かれているか、又は、ターゲット・ディレクトリーが存在しないかも知れません)

Device / Device options / Serialization / Custom generator mode 「デバイス/デバイス・オプション/シリアライゼーション/カスタム・ジェネレーター・モード1

ユーザーが自分でシリアライゼーション・システムを全て持つ場合は、カスタム・ジェネレータ・シリアライゼーション・モードが最もフレキシブルなシリアライゼーション・モードを提供します。

シリアライゼーションのCustom generator mode[カスタム・ジェネレータ]モードが選択された時、PG4UW、又 は、PG4UWMCで各デバイスがプログラムされる前にユーザーが作成したプログラムによって"on-the-fly[オンザ フライ]"でシリアル番号が生成されます。カスタム・ジェネレーター・モードのシリアライゼーションはユーザーが望むユ ニークなシリアル番号のシーケンスを生成することが出来ます。シリアル番号はリニア・シーケンス、又は、完全な 非リニア・シーケンスとしてインクリメントすることが出来ます。ユーザー作成シリアル番号ジェネレーター・プログラ ムの詳細は以下のCustom generator programセクションで説明します。



Examples:

また、.exeとC/C++のソースファイルも利用可能です。ファイルは次のPG4UWインストレーション・ディレクトリ Examples\subdirectoryに有ります:

<PG4UW_inst_dir>\Examples\Serialization\customgenerator_example 一般的なパスはこの ように見えます:

C:\Program Files (x86)\Elnec_sw\Programmer\Examples\Serialization\ customgenerator_example\

PG4UWコントロール・ソフトウエアのCustom generator serialization[カスタム・ジェネレーター・シリアライゼー ション] のための次のオプションがあります:

ダイアログ "Serialization"の**Mode**パネル・オプションでCustom generator modeを選択。次のオプションが表示されます:

Serialization data file[シリアライセーション・データ・ファイル]

現在のシリアル番号が含まれるデータ・ファイルのパスと名前を指定します。デバイスをプログラムする時、 PG4UWソフトウェアはユーザーが作成したデータ・ファイルを更新するシリアル番号ジェネレーターを呼び出しま す。データ・ファイルの推奨される拡張子は.datです。弊社のカスタマーの多くがBP Microsystems社のプログ ラマも使用しているためユーザーは普通同じシリアライゼーション・ソフトウェアを使用することを希望するためです。 従って、シリアライゼーション・データ・ファイルはBPマイクロ社のソフトウェアで利用できる "Complex serialization"の.datファイルと互換性が有ります。

ノート:データ・ファイルは全て定期的にデバイスのプログラミング中にシリアライゼーションで上書きされます。希望する.datファイルの正しい名前を確実に入力して下さい。例えば:"c:\serial_files\serial.dat"です。

Serialization generator[シリアライゼーション・ジェネレーター]

シリアライゼーション・データ・ファイルが生成される実行ファイルのためのパスと名前を指定します。

最初のシリアル番号

このオプションはカスタム・ジェネレーター・シリアライゼーション・プログラムに渡される最初のシリアル番号を指定する 必要があります。番号は入力されますと16進数形式で表示されます。

最後のシリアル番号

このオプションは許可されたシリアル番号の最大値を指定します。値がゼロでない場合に、シリアライゼーション・ジェ ネレーター・プログラムへ渡されます。ジェネレーターは最後のシリアル番号の値を、テストし、そして、現在のシリアル 番号が最後のシリアル番号より大きい場合には、そのシリアライゼーション.datファイルに適切なエラー内容を持った シリアル .datファイルを生成します。最後のシリアル番号の値がゼロの場合、その値はジェネレーター・プログラムに 渡されません。

チェック・ボックス Call generator with -RESULT parameter after device operation completed [デバイスの操作が完了した後に-RESULTパラメータと共に、ジェネレータをコール]

この新しいオプションは特別な目的を持っています。特殊なパラメータ-RESULTでカスタム・ジェネレータを呼び出 す要件がある場合は、チェックボックスにチェックを入れておく必要があります。それ以外の場合は、チェックを外して オフにしておく必要があります(デフォルトの状態はオフです)チェックした場合、各デバイス操作が完了した後、カス タム・ジェネレーターはデバイス操作の結果がOK、又は、エラーに関係なくPG4UW制御プログラムによって呼び出 されます。ジェネレーターのパラメータはPG4UWシリアライゼーション・エンジンによって作成されます。2つのパラメー タが使用されます:

-RESULT[n] =TRUE / FALSE



nはマルチプログラミングが使用されている場合のオプショナルなプログラマ・サイトの番号。TRUEはデバイス操作が OKで終了したことを意味します。FALSEはデバイス操作がエラーで終了したことを意味します。

-N<serial number>

シリアライゼーション・ジェネレーターの通常の呼び出しと同様で、現在のシリアル番号を指定

Custom generator program[カスタム・ジェネレータ・プログラム]

カスタム・ジェネレーター・プログラム、又は、シリアライゼーション・ジェネレータはシリアル番号のユニークなシーケンス を発生しそのシリアル・データをシリアライゼーション.datファイルに書き込むプログラムです。 このプログラムはユーザー側で作成します。シリアライゼーション・プログラムのパスと名前は、カスタム・ジェネレータ ー・モード・オプションのシリアライゼーション・オプションのダイアログで指定する必要があります。

プログラムは新しいシリアル・データが生成される必要ある度にPG4UWから呼び出されます。これは通常各デ バイスのプログラミング操作の前に行われます。PG4UW制御プログラムはシリアライゼーション・プログラムにコマ ンド・ライン・パラメータを渡し、そして、シリアライゼーション・プログラムはPG4UW制御プログラムによって読み込 まれるシリアライゼーション.datファイルを生成します。以下のコマンド・ライン・パラメータが使用されます:

-N<serial number> 現在のシリアル番号を指定

-E<serial number> 最終(又は、最後)のシリアル番号を指定。

パラメータはPG4UWソフトウエアのダイアログ "シリアライゼーション"で最後のシリアル番号の値がゼロで無い時の み渡されます。シリアライゼーション・プログラムは、もし現在のシリアル番号が最後のシリアル番号よりも大きい場合、 シリアライゼーション.datファイルにエラー・レコードT06を返します。詳細については"シリアライゼーション.datファイル 形式"のセクションを見て下さい。

Serialization .dat file format t[シリアライゼーション .dat ファイル・フォーマット]

シリアライゼーション・ジェネレータによって生成されたシリアライゼーション.datファイルは次のテキスト形式でなければ いけません。シリアライゼーション.datファイルはレコードとシリアル・データ・セクションで構成されています。

レコードは以下に説明する様にTxxプリフィックスの1つで始まる行です。"xx"の値はレコード・タイプのコードを表します。レコードはPG4UWソフトウェアにシリアライゼーションの状態(現在と最後のシリアル番号、シリアライゼーション・データとデータフォーマット、エラー等)を知らせるために使用されます。必要なレコードはレコードT01, T02, T03とT04です。その他のレコードはオプションです。

T01:<serial number> コマンド・ライン・パラメーター -N<serial number>によってジェネレーターに渡す現在の シリアル番号が含まれています。

T02:<serial number> PG4UWが次のシリアライゼーションで使用する次のシリアル番号値を含んでいます。 この値はPG4UWに現在のシリアル番号の次のシリアル番号を知らせシリアライゼー ション・ジェネレーターで生成されます。

T03:<data format code> シリアライゼーション・・データ形式を指定。次のフォーマットがサポートされています:

T03:50 又は、T03:55 ASCIIスペース・データ形式

T03:99 - Intel Hexデータ形式

T04:

シリアライゼーション・データが次の行からファイルの最後に続くことを示します。シリア ライゼーション・データは例えばIntel Hex, ASCII Space等々の標準のASCIIデー タ・ファイル形式の1つで保存されます。データに使用するフォーマットはレコードT03 で指定する必要があります。



Example: 典型的なシリアライゼーション・データ・ファイル:

T01:000005 T02:001006 T03:99 T04: :03000000000096B89 :0300030000005F5 :02000C005A0197 :01003F004F71 : 00000001FF

ファイルは以下の情報で構成されています:

line T01 - 現在のシリアル番号 000005h

- line T02 最終(最後)のシリアル番号 001006h
- line T03 行 T04の後のシリアライゼーション・データ形式はIntel Hexです。
- line T04 デバイス・プログラミングの前にPG4UWのバッファにロードされるシリアライゼーション・データ、データ はインテルHEXフォーマットで表現されます。

オプショナル・レコードは:

- T05:<message> ワーニング、又は、エラー・メッセージ。このレコードはシリアライゼーションが中止されたために起 こり、そして、PG4UWソフトウエアでワーニング、又は、エラー・メッセージが表示されます。
- T06: 現在のシリアル番号が制限より大きい。

このレコードはシリアライゼーションを停止し、PG4UWソフトウェアにより警告、又は、エラー・ メッセージが表示される原因となります。シリアライゼーションをオフにする理由は現在のシリアル番 号が許可された最大値の最終シリアル番号より大きいためです。このレコードは-Eコマンド・ライ ン・パラメータが指定されている場合に使用することが出来、それはシリアライゼーション・ダイアログ でシリアル値の指定がゼロでないことを意味します。

T11:<message> 余り重要でないワーニング、又は、メッセージ。 シリアライゼーションは停止されません。 カスタム-ジェネレーター・シリアライゼーションでのデバイス・プログラミングのフローチャート

カスタム・ジェネレータのシリアライゼーションが使用される場合、各デバイスのプログラミングが開始される 前に、シリアライゼーション・エンジンがシリアル・datファイルを生成するために実行可能なシリアライゼーシ ョンを呼び出します。PG4UWのシリアライゼーション・エンジンはシリアライゼーション・ジェネレーターを呼 び出すために適切なコマンド・ライン・パラメータを管理します。..datファイルからのデータは直ちに内部プロ グラマー・バッファに読み出され、そして、プログラミング・デバイス用のデータとして使用されます。また、次 のシリアル番号情報(レコード T02)はPG4UWに記憶されています。

デバイス・プログラミングの典型的なフローチャートは次の通りです:

- 1. プログラミング・バッチを開始
- 2. デバイス装着テスト
- 3. シリアライゼーション・シーケンスは4つのステップからなります:
 - ・シリアライゼーション.datファイルを発生させるために適切なコマンド・ライン・パラメータでシリアライゼーション・ ジェネレーターを呼び出す
 - ・利用可能なシリアライゼーション.datファイルを待つ
 - シリアライゼーション .datファイルのデータをプログラマー・バッファへ読み込む(データはプログラミング・デバイスのために使用)
 - ・データを読み込んだ後シリアライゼーション.datファイルを削除
 - 4. デバイス・プログラミング



- 5. デバイス・ベリフィケーション
- 6. 操作結果のチェック

これは全てPG4UWコントロール・プログラムによって管理されます。シリアライゼーション・ジェネレーターの操作結果はどの操作とも関係がありません。コントロール・プログラムは要求されたコマンド・ライン・パラメータでシリアライゼーション・ジェネレーターを呼び出します。

OK - PG4UWは次のシリアル番号の要求をします。次のシリアル番号はステップ3で .datファイルから読み 込まれています。シリアライゼーション・ジェネレーターの呼び出しにより、コマンド・ラインで指定された次のシリ アル番号を持ちます。

ERROR - PG4UWは新しいシリアル番号の要求をしません。最後のシリアル番号は次のデバイスで使用 されます。次のシリアライゼーション・ジェネレーターの呼び出しはコマンド・ラインで指定された最後のシリアル 番号を持ちます。

- 7. 次のデバイスへのプログラミングを繰り返しますか?
 - Yes
 ステップ2へ行く

 No
 ステップ8を継続
- 8. プログラミング・バッチの終了
- ノート:
- エラー・プログラミングの場合、最後のシリアル番号が使用されますが、ジェネレーターはステップ3で呼び出されます。とにかくもし同じ番号が以前にプログラムされたデバイス用として用いた場合であっても、呼び出されます。
- もし、シリアライゼーション.datファイルのエラーが検出された場合、プログラムPG4UWはシリアライゼーション・エラーを報告し、即プログラミングのバッチ処理を中止します。

Device/Device options/Statistics[デバイス/デバイス・オプション/スタティスティクス(統 計)]

スタティクスは選択されたタイプのデバイスで処理されるデバイス操作の実際のカウントについての情報を提供しま す。もし、1つのデバイスが1つの操作に対応している場合、即ち、プログラミング、デバイス操作の数がプログラムさ れるデバイスと同じ場合です。

スタティクス[統計]の次の機能はCount down[カウント・ダウン]です。カウント・ダウンはデバイス操作の数、そして、デバイス操作がおこなうべきデバイスの数をチェックします。それぞれの成功したデバイス操作の後にカウント・ ダウンのカウンターは反対に減少します。カウント・ダウンはユーザーが定義したデバイスの開始番号を持っていま す。カウント・ダウン値がゼロに達しますと、指定したデバイスの数が完了し、そして、カウント・ダウンの完了についてのユーザー・メッセージが表示されます。

Statistics[スタティスティクス] ダイアログは下記のオプションを含んでいます:

チェック・ボックス Program[プログラム], Verify[ペリファイ], Blank[ブランク], Erase[イレース] と Read[リード] はスタティクス値がインクレメントされた後でオプションを定義します。

如何なる選択され実行されたデバイス操作もTotal カウンターをインクリメントし、そして、デバイス操作の結果 (成功、又は、失敗)に応じてSuccess[成功]又は、Failure[失敗]になります。部分操作の組み合わせも1 つの操作としてカウントされます。例えば、Readの後のVerifyを含むRead操作は1つの操作です。Eraseと/又 は、Verify操作を含むプログラム操作も1つの操作としてカウントされます。

チェック・ボックス Count down[カウント・ダウン] はカウント・ダウンの有効、又は、無効を設定します。カウント・ ダウンに続くエディット・ボックスはカウント・ダウンが開始されるカウンターの最初の番号を定義します。

Statistics[スタティクス] ダイアログはStatistics パネルで右マウス・ボタンを押して、そして、表示されている項 目Staticsはクリックすることで開くことができます。



Statistics[スタティクス] ダイアログは7つの値が含まれます – Success[成功], Operational Failure[操作 失敗], Adapter test failure[アダプター・テスト失敗], Insertion test failure[インサーション・テスト失敗]、 ID check failure[IDチェック失敗] と他のFailure[失敗](prog. SW, HW)とTotal[合計]

値の意味は:

Success 成功して完了した操作の数 Operational failure デバイス・エラーで失敗した操作の数 Adapter test failure アダプターによる失敗した操作の数 Insertion test failure アダプターの誤った位置により失敗した操作の数 ID check failure デバイスからの ID コードの読み出しで失敗した操作の数 Other failure(prog. SW, HW) ハードウエア・エラー、又は、制御ソフトウエアのエラーにより失敗した操作の 数

Total 全操作数

実際のStatistics[統計]値はメイン・ウィンドウのStatistics[統計]パネルに表示されます。

Statistics[スタティクス] パネルは4つの統計値を含みます - Success[成功], Operational Failure[操作失 敗], Other Failure[他の失敗]Total[合計]と2つのカウントダウン情報値Count down[カウント・ダウン]と Remains[残り]

値の意味は:

成功して完了した操作の数 Success デバイス・エラーで失敗した操作の数 **Operational failure** Other failure デバイス・エラー以外の理由で失敗した操作の数 Total 全操作数 Count down カウント・ダウン(有効、又は、無効)の情報 Remains デバイス操作の残り数の情報 Jート:新しいデバイス・タイプが選択されたとき、すべてのスタティクス値はゼロにセットされ、そして、Count down[カウント・ダウン] はDisabled[ディスエーブルド] にセットされています。 StatisticsパネルのReset[リセット] ボタンはスタティクス値をリセットします。 StatisticsパネルのReload Count down「カウント・ダウンの再ロード」 ボタンはカウント・ダウンに初期値を再口 ードします。 PG4UWソフトウェアを使用する場合は、PG4UWを閉じるときに統計情報がログ・ウィンドウに保存されます。 PG4UWMCソフトウェアのマルチプログラミングの場合、統計情報はジョブ・ヤルフテスト・サマリーレポートに保存さ れます。

Device/Device options/Associated file[デバイス/デバイス・オプション/関連ファイル]

このコマンドはターゲット・デバイスの関連ファイルを設定するために使用されます。これはデフォルト・デバイス選択リ スト又は、コントロール・プログラムをスタートした後にバッファに自動的にロードすることが出来るファイルです。

ユーザーはファイル名ボックスで関連ファイル名を編集することが出来ます。パス名をフルに付けて下さい。コントロール・プログラムはディスクのこのファイルの存在をチェックします。また、このファイルの自動ロードを有効、又は、無効も 変更出来ます。

File / Exit and save[ファイル/終了と保存] コマンドで両方、すなわち、関連ファイルと自動ロードの有効をディスク にセーブ出来ます。



Device/Device options/Special options[デバイス/デバイス・オプション/スペシャル・オ プション]

使用する全ての用語の説明についてはプログラムしたいチップのドキュメントをお読みください。このメニュー項 目の名前が"View/Edit …"で始まる場合、デバイスの読み込みコマンドはチップ構成の内容を読み込みこ のメニュー・コマンドで表示および編集できます。

Device / Blank check[デバイス/ブランク・チェック]

このコマンドはデバイスのブランク・チェックを行ないます。コントロール・プログラムはINFO[情報]ウィンドウとLOG に警告のメッセージを書くことにより、このアクションを報告します。

メニュー・コマンド Device / Device options / Operation options [デバイス/デバイス・オプション/操作オプ ション] で利用できる操作オプションをカスタマイズすることが出来ます。

Device / Read[デバイス/リード]

このコマンドはバッファに全デバイス、又は、その一部分を読み込むことが出来ます。また、リードはチップのコンフ ィギュレーション(もし、存在し、読み取り可能な場合)の内容も読み取ります。スペシャル・デバイス・コンフィギュ レーション領域はメニューView/Edit bufferとメニューDevice / Device options / Special options[デバイス/ デバイス・オプション/特別オプション](Alt+S)で利用できるダイアログで見たり編集することが出来ます。

コントロール・プログラムはINFO[情報]ウィンドウとLOGにメッセージを書くことによりこのアクションの終了を報告します。

メニュー・コマンド Device / Device options / Operation options [デバイス/デバイス・オプション/操作オプション] は標準で他のワーキング・エリアを設定することが出来ます。このメニュー・コマンドでオプションVerify data after reading[読み出し後にデータをペリファイ] を設定することは、デバイスの読み出しにより高い信頼性を持たせるこ とを意味します。

Device / Verify[デバイス/ベリファイ]

このコマンドは利用可能な特別な領域を含むデバイス全体のコンテンツをバッファ内のデータと比較します。 制御プログラムはベリファイ操作の結果を情報ウィンドウとログ・ウィンドウに報告します。

PG4UWのメニュー・コマンド Device / Device options / Operation options [デバイス/デバイス・オプション/ 操作オプション] で利用できる操作オプションをカスタマイズすることが出来ます。

PG4UWのダイアログ General options(Options/General options [オプション/一般オプション] のタブ**Errors**は 見つかったエラーをユーザーが指定したレポート・ファイルに書き込む方法を制御できます。また、最初の見つかった 45のエラーはLogウィンドウに書きこまれます。

ノート:

- イリファイ操作はソフトウエアでデータをチップ全体の内容を比較しますので、従って、不完全にプログラムされたチップの場合 プログラミングの後のベリフィケーションではエラーは無くても、ソロ(単体)ベリファイ操作はパスしないかも知れません。
- ベリファイ操作はデータのアクティブなリード・プロテクションを持ったプロテクトされたデバイスの場合もエラーを報告 することが出来ます。



Device / Program[デバイス/プログラム]

このコマンドはデバイスのプログラミングを実行します。コントロール・プログラムは情報ウィンドウとLOGのメッセージによりこのアクションの結果を報告します。

PG4UWのメニュー・コマンド Device / Device options / Operation options[デバイス/デバイス・オプション/ 操作オプション] はプログラムされる領域をカスタマイズすることができ、そして、その他の操作オプションを設定するこ とが出来ます。

Device / Erase[デバイス/イレース]

このコマンドはデバイスのイレースを実行します。コントロール・プログラムは情報ウィンドウとLOGのメッセージよりこの アクションの結果を報告します。

PG4UWのメニュー・コマンド Device / Device options / Operation options[デバイス/デバイス・オプション/操作 オプション] は利用できる操作オプションをカスタマイズすることができます。

イレース後、もし、デバイス(チップ)がイレース・ベリファイ・コマンドをサポートしていない場合、ブランク・チェック操作が イレース操作のベリファイ成功を代行します。

Device / Test[デバイス/テスト]

このコマンドはこのテストをサポートするプログラマでサポートされているデバイス(例えば、statics RAM)のリ ストから選択されたデバイスのテストを実行します。

sRAMは3つのベーシック・ステップでおこなわれます:

・データドライバ[デバイス出力ピン]機能のテスト

ドライバ・テスト... D0..D7のテストはCE\, OE\ とWE\の信号反応を示します: - 最初のサイクルでアドレス0x0(CE/=L WE/=L OE/=H) にデータ0x55を書き込み、同じアドレス(CE/=L WE/=H OE/=L)から読みだされたデータと比較します。データは有効でなければいけません。 - そして他の組み合わせ制御ピン(CE/=L WE/=H OE/=H), (CE/=H WE/=H OE/=L), ..., がセットされ、 データが無効であることもチェックします - データバスドライバは非アクティブです。

• sRAMテスト, ペーシック・パーツ

プログラマはランダムなデータをsRAMデバイスに書き込んだ後、その内容を確認します。

• RAMテスト, アドバンスト(オプショナル)

"Walking one"(1を書き込む)と"Walking zero"(0を書き込む) テストを行うことが出来ます。

http://www.google.com/search?q=memory+test+walking+one

http://www.google.com/search?q=memory+test+walking+zero

ノート:

- ビットのリークを検出する目的で書き込み動作とプログラムされたデータのベリファイ後との間の遅延を(デバイ スが供給されている状態で)選択することが可能です。
- プログラマは信号ピン上の電流が大きすぎるか、又は、アナログ・エラーを検出することはできません。
- 全てのテストは低周波数(テストされるデバイスの最高速度と比較される)で行われるため、このようなテストの 使用は制限されています。



結論:

- デバイス・プログラマはsRAMの健全性に関する基本的な回答のみを提供できます
- sRAMをより深くテストする必要がある場合は、特化されたsRAMテスタを使用してください。

Device / IC test[デバイス/ICテスト]

このコマンドはICに対するテスト・セクションをアクティベートします。主に標準のロジックIC。ICはテクノロジーの種類ごとにグループ/ライブラリに分類されます。

最初に適切なライブラリ、希望のデバイス、テスト・ベクトルの実行モード(LOOP、SINGLE STEP)を選 択します。制御シーケンスとテスト結果はプログラマのアクティビティ・ログに表示されます。必要な場合には、 テスト・ベクトルをユーザーが直接定義することが可能です。テスト・ベクトルの作成の構文と方法の詳細な 説明はプログラム・インストール・フォルダにあるexample_e.libファイルに記述されています。

ノート:

- ICのテストはある程度(かなり低い)速度でテスト・ベクトルを使用して行われます。テスト・ベクトルによるテストで はチップの全ての欠陥を検出することはできません。言い換えれば、ICテストが "FAIL"と報告した場合、デバイ スに欠陥があります。しかし、"PASS"が報告された場合は、チップがテストに合格したことを意味しますが、テス トされたICの他の主にダイナミックなパラメータをチェックするテストに合格しない可能性があります。
- ・プログラマの立ち上がりご立ち下がりエッジはチップのプログラミングに合わせて調整されるためチップに欠陥がない(例えばカウンタなど)にもかかわらず一部のチップのテストが失敗することがあります。

Device / Jam/VME/SVF/STAPL/mDOC ... Player

Jam STAPLはAltera® 社により開発され、そして、プログラマブル・ロジック・デバイス(PLD)製造業者, プログラミング装置メーカーとテスト装置製造業者のコンソーシアムによりサポートされています。

Jam[™] 標準テストとプログラミング言語(STAPL), JEDEC standard JESD-71はISP(イン-システム・プログラミン グ)の目的のための標準ファイル・フォーメットです。 Jam STAPL はフリー・ライセンスのオープン・スタンダードです。 それはIEEE 1149.1 Joint Test Action Group (JTAG)インターフェースを使用したプログラミング・デバイスと電気 回路システムのテストのプログラミング、又は、コンフィギュレーションをサポートしています。 デバイスはプログラム、又 は、ベリファイ出来ますが、 Jam STAPLはデバイスのリードのような他の機能は一般的には許可されていません。

Jam STAPLプログラミング・ソリューションは2つのコンポーネントで構成されています: Jam Composer とJam Player Jam Composerはプログラムであり、一般的にデバイスにデザインをプログラムするのに必要なユーザー・データとプログラ ミング・アルゴリズムを含むJam file(.jam)を生成し、プログラマブル・ロジック・ベンダーにより書かれています。 Jam PlayerはJamファイルを読み取り、プログラミングやJTAGチェーンでデバイスのテストのためのベクターを適用するプ ログラムです。

デバイスはプログラマのZIFソケット、又は、ISPコネクタを介してターゲット・システムにプログラムできます。制御プログラム で選択されたデバイスの名前の後に[PLCC44](Jam)、又は、(ISP-Jam)サフィックスで示されます。JTAGチェーンを 介したプログラム、及び、テストは複数のデバイスが可能です:JTAGチェーン(ISP-Jam)



さらに詳しい情報はウェブサイト: http://www.altera.com でアプリケーション・ノートを見て下さい: "AN 425: Altera デバイスをプログラムするために Jam Player を使用", "AN 100: イン-システム・プログラマビリティ・ガイドライン", "AN 122: 組み込みプロセッサー経由でISP & ICRのためにJam STAPLを使用" と詳 細のための関連アプリケーション・ノート。

ソフトウエア・ツール:

Altera: MAX+plus II, Quartus II, SVF2Jam utility (シリアル・ベクター・ファイルをJam fileに変換), LAT2Jam utility (ispLSI3256A JEDECファイルをJamファイルに変換);

Xilinx: Xilinx ISE Webpack又は、Foundation software(ユティリティ SVF2Jamで使用するためにSTAPLフ ァイル、又は、SVFファイルを生成);;

Actel: Actel Libero® Integrated Design Environment (IDE) (STAPLEファイルと/又は、PDBファイルを生成), Actel FlashPro (PDBファイルをSTAPLEファイルに変換).

JAM player dialog[JAM playerダイアログ]

Jam Player			×
Action default			~
Variables			
Identifier	Optional	Recommended	^
DO_ERASE	×		
DO_BLANKCHECK	×		
DO_PROGRAM	н		
DO_VERIFY	ж		
DO_READ_UES	×		
DO_READ_USERCODE	×		
DO_SECURE	ж		
DO_SECURE_ALL	×		~
OK Cancel Device according to Jam file EPM7064AE	Inform	ation <u>H</u> elp	

Jam Player version 1 (Action とVariables controlsをご覧ください)

Jam Player		
Action PROGRAM		v
Procedures		
Identifier	Optional	Recommended
DO_BLANK_CHECK	ж	
DO_VERIFY		ж
DO_SECURE	н	
DO_LOW_TEMP_PROGRAMMING	н	
DO_DISABLE_ISP_CLAMP	ж	
DO_READ_USERCODE	×	
OK Cancel (Device according to Jam file EPM7064AE	Information	n <u>H</u> elp

Jam Player version 2 (ActionとProcedures controlsをご覧ください)



Action[動作]

実行したい動作を選択して下さい。

version 2の Jam file は動作で構成されています。動作は実行させる手順の呼び出しを含んでいます。 version 1の Jam file はステートメント'action'と'procedure'を認識しないため、従って、Actionの選択はアクセス 可能ではありません。プログラム・フローはプリフィックス DO_something を持ったブーリアン変数に応じた命令を実 行するために開始します。プリフィックス DO_something の新しいブーリアン変数が必要な場合はご連絡下さい。

Procedures[手順]

プログラム・フローは各手順からのステートメントを実行します。手順はオプションと推奨をご使用下さい。推奨 手順は暗黙的にマークされています。必要に応じて手順を有効、又は、無効にすることができます。Jam Playerはマークされた手順のみを実行します。その他の手順は無視されます。手順の数はJamファイルに依存 して異なります。

Variables[変数]

version 1のJam fileはステートメント'action'と'procedure'を認識しません。プログラム・フローはプリフィックス DO_somethingを持ったブーリアン変数に応じた命令を実行するために開始します。Jam Playerはアルゴリズ ム内の全てのマークされたDO_somethingを実行します。変数(手順)は定数であり、それはJamファイルに依 存しません。もし、プリフィックスDO_somethingを持った新しいブーリアン変数が必要な場合はご連絡下さい。

ок

マークされた適切な手順で選択されたアクションを受付けます。

Information

Jamファイルについての情報を表示。ダイアログでノートとソース・ファイルをプレビューすることが出来ます。

Device according to Jam file

ファイルは特定のデバイスのために作られています。デバイス名はJamファイルのNOTE identifier DEVICEに含まれています。デバイス名はダイアログ Select device[デバイス選択]で選択されたデバイスの名前と同じでなければいけません。デバイスが異なる場合、ソフトウェアがJam Playerの開始中の警告メッセージによってこの状況を示します。

JAM file information dialog[JAM file情報ダイアログ]

rmation		
otes Source File		
Identifier	Text	
CREATOR	QUARTUS II JAM COMPOSER 4.0	
DATE	2004/03/22	
DEVICE	EPM7064AE	
FILE	c4_epm7064AELC44_10.pof	
TARGET	1	
IDCODE	170640DD	
USERCODE	FFFFFFF	
CHECKSUM	000EBCEC	
SAVE_DATA	DEVICE_DATA	
SAVE_DATA_VARIABLES	V0, A12, A13, A25, A43, A92, A94, A95, A50	
STAPL_VERSION	JESD71	
JAM_VERSION	2.0	
ALG_VERSION	35	
		Canaal
		Lance



ノート:ステートメントはJamファイルに関する情報をストアするために使用されます。NOTEフィールドにストアさる 情報には特定のJamプログラムに関連したあらゆるタイプのドキュメントと属性が含んでいます。

ソース・ファイルには Jam 言語でのプログラムが含まれています。Jam プログラムはステートメントのシーケンスで構成されています。Jam のステートメントはオプション、命令と引数であるラベルを含みセミコロン(;)で終了します。引数はリテラル定数、変数、又は、目的のデータ・タイプ(即ち、ブーリアン、又は、整数)での式になります。各ステートメントは通常 Jam プログラムの 1 行を占有しますが、これは必須ではありません。改行は終了コメントを除いて Jam 言語の構文に重要ではありません。アポストロフィ文字(')を使用してコメントを表わすことができますが、これはインタプリタによって無視されます。文法は行の長さ、ステートメントの長さ、又は、プログラム・サイズのための制限を指定していません。より詳しい情報はウェブサイト上で見つけることが出来ます:<u>http://www.altera.com</u>

拡張子 .jbcの付いたJamファイルはエディタで表示出来ないJam STAPLE Byteコード形式です。

XILINXデバイスのためのJEDファイルからJam STAPLEファイルへの変換:

・フリー・ダウンロード Xilinx Integrated Software Environment(ISE) 6.3i ソフトウェアのインストール: WebPACK_63_fcfull_i.exe + 6_3_02i_pc.exe (315MB 程)

·Xilinx ISE 6/Accessories/iMPACT を起動

·ダイアログ "Operation Mod Selection: What do you want to do first?" 選択: "Prepare Configuration Files,

・ダイアログ "Prepare Configuration Files: I want create a:" 選択: "Boundary-Scan File",

- ・ダイアログ "Prepare Boundary-Scan File: I want create a:" 選択: "STAPL File",
- ・ダイアログ "Create a New STAPL File" 拡張子.staplの Jam ファイル名を入力,
- ・ダイアログ "Add Device" 拡張子.jed の JED ファイルを選択,
- ・作成された JTAG チェーンでデバイスを選択します(例: XC2C32A(マウスの左ボタン)、及び、シーケンス操作を 選択します(例: Erase, Blank, Program, Verify; マウスの右ボタン)),
- ・メニューの選択項目で"Output/Stapl file/Stop writing to Stapl file"を選択
- ・PG4UWを実行、デバイスを選択、助ち: Xilinx XC2x32A [QFG32](Jam), Jamファイルをロード(ファイル・タイプの選択: STAPL Fileを選択)
- ・"Device operation options Alt+O"を選択、"Jam configuration"ボタンを押します。 警告 メニューからの デバイス選択"Select Devices"とJamファイルは恐らく違います! 続けますか?" はいを選択(Xilinxソフトウェア は行: NOTE "DEVICE" "XC2x32A";をJamファイルに含んでいません)。 ダイアログ"Jam player"でアクショ ンと手順を選択、ダイアログの終了、ツール・バーからボタン"Play Jam"をクリック、そして、Logウィンドウを読 みます。

STAPLEファイルを使用したACTELデバイスのプログラミングについての情報

PG4UWプログラムでのActel社製flash FPGAのプログラムはActel Jam playerを使用して実行されます。このプロ グラミング・ソリューションにより特別なコンテンツ・ツールバー・ボタン- Play STAPLが生成されます。これはJam以外 のプログラミング・デバイスに関連する全ての一般的な操作アイコン(プログラム、消去、ペリファイ...)を置き換えます。

Actel デバイスの操作(program, erase, verify...)は以下のいくつかのステップを含んでいます。

*.stp (STAPLE ファイル)をロード

メイン・ツールバー上の"Load"アイコンをクリックして適切なSTAPLEファイル(例えば、Actel design software LIBRO IDEによって生成されたファイル)をロード。このSTAPLEファイルにはSTAPLEファイルには、デザインをデバイスにプログラム するために必要なユーザーデータとプログラミング・アルゴリズムが含まれています。



動作を選択

STAPLE ファイルのロード完了後、デバイス操作オプション(Alt+O ショートキー)/STAPL configuration....(STAPL configuration)で意図する動作の操作を選択して下さい。デバイスをプログラムするにはアクション・リスト内の PROGRAMを選択して下さい。プログラミング・ファイルに関する全ての動作のリストに関する説明は<http://www.actel.com>の ACTEL FlashPro User's Guideを参照して下さい。

動作を実行

選択したアクションを実行させるためにPlay STAPLボタンをクリックして下さい。操作(即ち、プログラミング)が完了 しますとLogウィンドウに"Exit Code = 0... Success"と表示されます。

ACTELテバイス用のPDBファイルからJAM STAPLEへの変換に関する情報

Actel PDB ファイルは Actel プログラマ、即ち、FlashPro プログラマのみでサポートされている専用ファイル・フォーマットです。PG4UW コントロール・プログラムは Actel デバイスを Jam STAPL ファイルでのみプログラムすることが出来ます。従って、PDBと STAPL 間のファイル変換が必要です。

ACTELデバイスのためにPDBファイルをJam STAPLファイルに変換:

- ソフトウエア・ツールFlashProをインストール(Actel Libero tool suiteのコンポーネント、又は、スタンド・アローン・バージョンとして<http://www.actel.com>からダウンロード)
- FlashProを実行
- New Project[新規プロジェクト]ボタンをクルック、又は、ファイル・メニューからNew Project[新規プロジェクト] を選択、そして、Project nameフィールドにプロジェクト名を入力して下さい。希望するプログラミング・モードを 選択 - single device[シング・デバイス]、又は、chain[チェーン]を選択しOKをクリックして下さい。
- コンフィギュレーション・メニューからLoad Programming File[プログラミング・ファイルのロード]を選択し、そして、 変換するための一致する*.pdbファイルを選択して下さい。
- ファイル・メニューから、Export/Export Single Device STAPL File...を選択し、ファイル名を入力し、そして、 指定したディレクトリにSTAPLファイルをエクスポートするためにSave[セーブ]ボタンをクリック
- PDBファイルからSTAPLへの変換が完了し、作成した*.stpファイルはActelデバイスのプログラミングために使用ことが出来ます。

Actelについてよくある質問

Q: 既にプログラムされた Actel デバイスをどのように ID チェック/ベリファイを行うのですか?

A: これを行うための幾つかのオプションがあります。各オプション(action)は既にプログラムされた Actel デバイスをロ ードされた STAPL ファイルで互いに比較しベリファイする方法です。次に STP ファイルでの適切なアクションを記載 します:

DEVICE_INFO: デバイスをリードし、ログ・ウィンドウに表示されているデバイスにプログラムされるプログラミング環 境のチェックサムをご覧下さい。この値は手動でSTAPLファイル(情報ウィンドウでも見ることが出来ます)のヘッダー の値と比較することが出来ます。注意:プログラムされたデバイスのチェックサムの値は実際の(壊れているかも知れ ません)デバイス・データの内容からカウントされませんが、この値はプログラミング中にスペシャル・メモリ・ローカライゼ ーションに保存され、そして、それだけ読み取っています! VERIFY_DEVICE_INFO: 前のオプションと類似して いますが、違いはプログラムされたデバイスのチェックサムとSTAPLEファイルのチェックサムを自動で比較します。比 較の結果はメッセージ・ウィンドウに success 又は、error の何れかで表示されます。VERIFY: STAPLE ファイル の内容とプログラムされたデバイス内容のデータの比較に関して、最も安全ですが、最も遅い(オプション1と2の約 1 秒に比べデバイスの要領に依存しますが数 10 秒)オプションです。選択されたファミリー機能の比較(FPGA Array, targeted FlashROM pages, security setting...)は bit ごとに実行されます。そして、もし、データのミスマ ッチが起こりエラー・メッセージがログ・ウィンドウに書かれますとベリフィケーション・プロセスは早期に終了することが出 来ます。



Q: PG4UW で2つの異なった STAPLE ファイルを1度のプログラム操作で Actel デバイスにプログラムは可能ですか?

A: はい。可能です。PG4UWコントロール・プログラムは上記の様な状況に対して内蔵のマルチ・プロジェクト機能 を持っています。例としてデータ・コンテンツ(最初のSTAPLEファイル)と一緒にセキュリティ暗号化キー(2番目の STAPLファイル)をプログラムすることが出来ます。

The IspVM Virtual Machine[IspVMパーチャル・マシーン]

IspVM Virtual Machineはバウンダリー・スキャン・テストのためのIEEE 1149.1 Standardと互換性のあるプログ ラミング・デバイスのための仮想マシーンです。IspVM EMBEDDEDツールはパウンダリー・スキャン・プログラミングと テストのための工業標準シリアル・ベクター・フォーマット(SVF)言語とLattice's IspVM Virtual Machine™のパワ ーが兼ね備わっています。

IspVM システム・ソフトウェアは IEEE 1149.1 standard 規格と SVF 又は、IEEE 1532 フォーマットをサポートした ispJTAG と非ラティス JTAG ファイルの両方をサポートしている VME ファイルを生成します。 VME ファイルは IspVM システム・ウィンドウからチェーン情報を得ることが出来る hex コード・ファイルです。

デバイスはプログラマのZIFソケット、又は、ISPコネクタを介してターゲット・システムにプログラムできます。制御プログラ ムで選択されたデバイスの名前の後に、[PLCC44](VME)、又は、(ISP-VME)サフィックスで示されます。JTAGチ ェーン(JTAGチェーン(ISP-VME))を介して複数のデバイスをプログラム、及び、テストできます。

さらに詳しい情報はウェブサイト: <u>http://www.latticesemi.com</u>をご覧下さい。

ソフトウエア・ツール:

Lattice: ispLEVER, IspVM System ISP Programming Software, PAC-Designer Software, svf2vme utility (シリアル・ベクター・ファイルをVME fileに変換) をご覧ください。

Device / Device info[デバイス/デバイス情報]

このコマンドは現在選択されているデバイス、デバイスのサイズ、構成織、プログラミング・アルゴリズムとデバイスを サポートするプログラマ(ソケット・モジュールを含む)の追加情報を提供します。ここでは現在のデバイスに関するパ ッケージ情報やその他の一般的な情報も確認できます。



リザーブ・キー<Ctrl+F1>如何なるメニューの時も直ちに表示します。



Programmer[プログラマ]

メニュー/プログラマにはプログラマでの作業に使用されるコマンドが含まれています。

Programmer / Find programmer[プログラマ/プログラマ検出]

この項目は新しいタイプのプログラマと通信パラメータを選択します。このコマンドには次の項目が含まれて います。:

Programmer[プログラマ] - 検出のための新しいタイプのプログラマを設定します。Search all[すべて 検索]を選択すると制御プログラムはサポートされているすべてのプログラマを検出します。

Establish communication[通信を行う - 新しいプログラマのために手動、又は、自動により通信を確立 することができます。

Speed - 手動通信が選択された場合、どのPCがプログラマにデータを送信するか速度をセット。速度は最大速度からのパーセントで表されます。

Port - 要求されたプログラマのためにスキャンされるポートを選択。All portが選択されている場合、制御プログラムは標準のアドレスで使用可能な全てのポートをスキャンします。

キー<Enter>、又は、OKボタンを押すとセットされたパラメータでプログラマのスキャンが開始されます。制御プログ ラムの開始時と同じアクティビティがあります。このコマンドは新しく選択されたプログラマがそのデバイスをサポートし ている場合、現在のデバイスのないデフォルト・デバイスのリストをクリアします。

このセッティングはコマンド Options / Save options[オプション/セーブ・オプション]によりディスクにセーブされます。

Programmer / Refind programmer[プログラマ/プログラマ再検出]

このメニュー・コマンドは現在選択されているプログラマを再検出(通信を再確立する)ために使用されます。 他のタイプのプログラマを選択するには、プログラマの通信パラメータと新たに選択されたプログラマとの通信 を確立するためにメニューProgrammer/Find programmer**[プログラマ/プログラマ]**を使用します。

Programmer / Handler[プログラマ/ハンドラー]

ダイアログ Handler[ハンドラー]でハンドラーのタイプとハンドラーの通信パラメータを設定できます。ハンドラーは 制御プログラム内のデバイス操作を特別に制御するための外部デバイスです。None Handlerが選択されてい る場合、これは制御プログラムのデフォルト状態を意味します、即ち、ハンドラーとの協調によりデバイス動作が 自動的に制御される場合、デバイス動作はユーザーによって直接制御されます。 ダイアログ Handler[ハンドラー] は次の項目を含みます:

Selected Handler希望するハンドラー・タイプを選択Search at port要求されるハンドラーをスキャンするCOMポートを選択

キー<Enter>、又は、OKボタンを押すと設定されたパラメータによってハンドラーのスキャンが開始されます。選 択されたHandler typeがNoneの場合、Handlerスキャンは処理されません。現在のハンドラの設定は、コマン ドこ Options/Save options[オプション/セーブ・オプション]、又は、制御プログラムが閉じられたときにコンフィギュ レーション・ファイルに保存されます。 ハンドラーは販売しておりません。



Programmer / Credit box info[プログラマ/クレジット・ボックス情報]

メニュー項目'Credit box info'[クレジットボックス情報]は PC に添付されているクレジットボックスに関する全ての必 要な情報(タイプ、シリアル番号、有効化に関する情報、利用可能なクレジットに関する情報)を提供します。複 数の'クレジットボックス'が PC に接続されている場合、ソフトウェアは利用可能な全てのクレジットに関する情報も 表示します。

クレジット・ボックスの利用可能性と添付されているクレジット・ボックス内のクレジットの量に関する情報もメイン・ウィンドウに表示され、'統計情報'、及び、'カウントダウン'セクションの下にある'クレジットボックス'ボタンに表示されます。このボタンは'Paid ISP support'[有料 ISP サポート]に属するデバイスが選択され、少なくとも1つのクレジット・ボックスが存在する場合に動的に表示されます。

'Credit box'[クレジットボックス]ボタンの下部にあるハ[・]ーグラフは添付されたクレジット・ボックスのクレジットのステータ スを示します: 緑色バーグラフ - 85%のクレジットが利用可能 黄色バーグラフ - 50%のクレジットが利用可能

赤色バーグラフ - 10%のクレジットが利用可能

バーグラフなし - 0%は利用可能なクレジットがゼロ

利用可能なクレジットに関する情報は定期的に更新されます(開かれたクレジットボックス情報ウィンドウ中も)。

デバイスでの作業は'Paid ISP support'[有償 ISP サポート]カテゴリに属します。

お客様の要望やニーズに応じて、プログラマブル・デバイスはますます複雑になります。また、プログラマブル・デバイ スの範囲はますます広くなっています。結果として、私たちはデバイス・プログラマの製造元として、より多くの開発リ ソースを費やさなければなりません。新しい複雑なプログラマブル・デバイスのサポートの実装は過去と比較して難し く、実装する必要のあるデバイスの数も非常に多いためです。

オフライン・プログラミング(ZIF ソケット、プログラミング・アダプタを使用)の場合、新しいデバイス・サポートの実装に 費やされる開発コストは、プログラミングアダプタの販売によってもカバーされます。しかし、ISPモードでプログラミング する場合、顧客はプログラマのみを購入し、新しいバージョンのソフトウェアに含まれるアフター・プログラマー・サポー ト費用は何の影響も受けません。

上記の様に ISP モードでプログラムされたデバイスについても、ソフトウェア・アップデートのダウンロードを無料に保つ ため、複雑なデバイスの ISP によるプログラミングには非常に僅かな料金が掛かるシステムを取っております。 ISP サ ポートの実装には非常に時間が掛かります。

この'Paid ISP support'システムは簡単です。このようなデバイスを使用するにはデバイス・プログラマが接続されて いる PC にクレジット・ボックスを接続する必要があります。クレジット・ボックスは、技術的には USB ポートの小さな ドングルで、モデルに応じて 25 から 500,000 のクレジットが含まれています。このような各デバイス・プログラミングの マイクロペイメントはクレジット・ボックス内のクレジット額の減少によって行われます。

クレジットの実際の手数料は非常に低く、クレジット・ボックスのCB-25kバージョン(=25,000クレジット)に対して約 0.01 USドルで開始し、クレジット・ボックスのCB-500kバージョンのクレジットでは約0.003 USドルまで急激に下 がります(=500,000クレジット)。

クレジット・ボックスの有効性は regular バージョンのソフトウェアの 10 バージョンに限られます。つまり、クレジット・ボ ックスが最初の使用時にアクティブ化されると、それはその時のバージョンと次の 9 つの regular バージョンで使用す ることができます。



例:3.01,3.01 バージョンのソフトウェアで起動する場合、3.01,3.02 ... 3.10 バージョンまでのソフトウェア(全ての OnDemand バージョンを含む)で使用できます。

Credit box[クレジット・ボックス]の状態は、PG4UW、及び、PG4UWMC ソフトウェアのメイン・ウィンドウに表示 されます。

詳細については、当社 Web サイトのアプリケーションノート - 'Paid ISP support'カテゴリにあるデバイスのクレジット・ボックスの使用方法 - をご覧ください。

Programmer/Module options[プログラマ/モジュール・オプション]

このオプションはMASTERソケットと各ソケットの動作を定義するためにマルチプル・ソケットプログラマに使用されます。MASTERソケット・グループ・ボックスはユーザーがデバイスの読み取り操作に優先的に使用されるソケットを設定できます。Enable/Disable socket [ソケット有効/無効] チェック・ボックス配列はユーザーが各ソケットの有効化と無効化を個別に設定できます。無効なソケットはどのデバイス操作でも無視されます。

Programmer / Automatic YES! [プログラマ/ Automatic YES!]

このコマンドはAutomatic YES! モードの設定に使われます。このモードではプログラムされたデバイスを取り除いて新しいデバイスをZIF ソケットに装着しますと最後の操作が自動的にリピートされます。プログラムが自動的に新しいデバイスの装着を検知し、最後に行った操作をキー又は、ボタンを押すことなく実行します。ZIFへのデバイスの装着は画面に表示されます。リピート操作の実行はZIFから/への装着/取り外しを待っている間に
<Esc> キーを押すことでキャンセルされます。

デバイスで操作が実行された後、プログラマ上のOK又は ERRORのステータスLEDが操作結果により点灯します。 そして、BUSY LEDが点滅します。プログラムがデバイスが取り除かれたことを検知しますと、ステータスLEDはオフ になり、新しいデバイスが最後の操作を繰り返すためにプログラムが用意できていることを示すためにBUSY LEDが 点滅します。プログラムがプログラマのZIFソケットにある新しいデバイスの1つ又は、それ以上のピンを示した後、 BUSY LEDは連続して点灯します。ここでプログラムは新しいデバイスの残りのピンが挿入されるための要求された 時間を待ちます。もし、要求された時間(デバイス挿入完了時間)が過ぎたり、デバイスが正しく挿入されない場合、 プログラムはこのステータスをERROR LEDで示します。新しいデバイスが正しく挿入された後、プログラムはBUSY 以外のステータスLEDをオフにします。そして、新しいデバイスで操作を開始します。

このモードはAutomatic YES! モードにより有効、又は、無効にすることが出来ます。もし、新しいプログラマが Options/Find programmer[オプション・プログラマ検出]で選択されますと、このモードは無効になります。

Response time[応答時間]はZIFソケットへのチップ装着と選択されたデバイス操作の開始の間隔となります。 もし、ZIFソケットでのチップの長いポジショニングが必要な場合はelongated response time[延長した応答時 間]を選択して下さい。

Programming adapter used[使用されるプログラミング・アダプタ]は現在選択されたデバイスで使用されるアダプタ名を示します。

Pins of ZIF excluded from sensing[感知から除外されたプログラマのピン]はAutomatic YES!によるテストで無視されたピンのリストです。ピンの無視の殆どの理由はそれらのピンへのコンデンサの接続になります。

ボタンSetting Automatic YES! parameters[Automatic YES!パラメータのセッティング] は完全に接続され たピン(コンデンサーがあるピン)を検出できるウィザードを実行し、これらのピンをセンシングから除外されたピンのリス トに設定します。デバイスの選択の後, 除外されたピンのリストには選択されたデバイス・アダプタに対してデフォルト の除外されたピンを含んでいます。もし、ユーザーによりユニバーサル・プログラマーと/又は、デバイス・アダプタに他の バイパス・コンデンサが追加する場合は、デフォルト・パラメータを無視し優先するためとコンデンサーのあるその他の ピンを検出するためにAutomatic YES! parameters wizard[パラメータ・ウィザート]を実行する必要がありま す。



Device removal hold off time[デバイス・リムーバル保持時間] はZIFソケットからデバイスを取り除いた時とソ フトウエアが新しいデバイスの装着をソケットでチェックを開始する時の間の時間間隔です。この時間は秒間隔で1 ~120 (デフォルト値は2 秒)でなければいけません。

Device insertion complete time[デバイス装着完了時間] はプログラムが不正に挿入されたデバイスを検出 しない様にするために最初のピン(複数)が検知された後に全てのピンが適切に挿入されなければならない時間を セットすることが可能です。この時間は秒間隔で1~120 (デフォルト値は2 秒)でなければいけません。

Suspend on error[エラーで停止]はAutomatic YES!機能でエラーが起こった時に一時停止して操作の結果を見るか、又は、停止せずに続けるかを定義します。

このオプションはDevice/Select device[デバイス/デバイス選択]で新しいデバイスが選択された後はデフォルトに セットされます。

このセッティングはコマンドOptions/Save options[オプション/オプション保存]によりディスクに保存し、選択したデバイスをFile/Save project[ファイル/プロジェクトをセーブ]...でプロジェクト・ファイルにセーブすることが出来ます。

J-ト: 供給電源をバイバスするためにコンデンサ等、一部のpassive 部品、又は、active 部品を備えたソケット・ アダプターを使用している時、Automatic YES!機能はそれらのピンをコンデンサ・リストのピンにセットする必要があ ります。それはSetting Automatic YES! parameters[Automatic YES!パラメータのセッティング] ウィザードで行 われます。これはAutomatic YES! 機能が正しく動作するために必要です。さもないとAutomatic YES! 機能は ピンが未だ接続されていると考え、ユーザーが新しいデバイスを挿入して新しいプログラミングを開始できません。

Programmer / Selftest[プログラマ/セルフテスト]

コマンドは標準装備のAP1 PMI selftest PODを使用して現在のプログラマのセルフテストを実行します。セルフ テストの実行をする推奨頻度はメンテナンス・セクションで確認できます。

Programmer / Selftest ISP connector[プログラマー/ ISPセルフテスト・コネクター]

コマンドはAP1 ISPコネクター・セルフテストポッド使用して現在のプログラマのISPコネクターのセルフテストを実行 します。

AP1 ISP connector selftest podはプログラマの20ピンISPコネクタのテストに使用する必要があります。 AP1 ISP connector selftest podはBeeHive204AP、及び、BeeProg2APの標準アクセサリです。*注 文番号:71-2007。

20ピンISPコネクターのテストのシーケンス:

• AP1 ISP connector selftest podをプログラマのPMIコネクタに挿入します。

- ISPケーブルでプログラマのISPコネクタにAP1 ISP connector selftest podの20ピン・コネクタを相互接続します。ピンが正しく相互接続(即ち 1-1, 2-2, 20-20)されていることを確認してください。
- PG4UW(Programmer/Selftest ISP connector…[プログラマ/ISPコネクタ・セルフテスト])でISPコネクタのセ ルフテストを実行して下さい。

このテストを6カ月毎に行うことを推奨します。



Programmer / Calibration test [プログラマ/カリブレーション・テスト]

コマンドはBeeHive204APの標準アクセサリとBeeProg2APのオプションアクセサリであるAP1キャリブレーショ ン・テストポッドを使用して、プログラマのキャリブレーション値のテストを実行します。注文番号:71-2008。 PMIの各ピン上のTTLドライバー、VCCP、VPP1とVPP2電圧の電圧レベルがテストされます。キャリブレー ション・テストの結果はファイルに保存したり印刷したりすることができます(次の使用のために)。

Options[オプション]

このオプション・メニューはユーザーが各種デフォルト設定を見て、そして、変更するためのコマンドを含んでいます。

Options / General options[オプション/一般オプション]

ー般オプション・ダイアログはユーザーがさまざまなPG4UWプログラム・オプションを制御、及び、設定できます。オプションはPG4UWアプリケーションを閉じるとき、又は、いつでもコマンド**Options / Save options**でPG4UW構成ファイルに保存できます。

File options [ファイル・オプション]

ファイル・オプション]・ページは読み込み前の消去バッファー、現在のファイルの自動再読み込み、及び、読み込ま れたファイルのファイル形式の認識方法のオプションを設定できます。

Erase buffer before loading[ローディング・オプション前にバッファをイレース]はデータ・ファイルのローディングの前にイレース・バッファ(希望する値で)自動的にセットします。

グループ内の現在のファイルが別のプロセスによって変更された場合、実際にロード(現在の)ファイルの再ロードのモードを設定することができます。3つの選択があります::

- 再ローディング・ファイル前のプロンプト
- 自動的に再ロード
- 現在のファイルの変更スキャンを無視 ファイルの修正がテストされる時の3つのシチュエーションが有ります:
- 他のアプリケーションからコントロール・プログラムへ切り替えられた時
- デバイス操作 Vefiry、又は、Programの選択された時
- ダイアログ"Repeat?"で最後のデバイス操作のリピートが選択された時

Load file format[ロード・ファイル・フォーマット]はファイルのロードのためのファイル・フォーマット認識のモードをセッ ティング。自動ファイル形式が選択された時、プログラムで利用可能なサポートされたフォーマットの各々に対してロ ーディング・ファイルとテスト・ファイルのプログラム・フォーマットを解析します。ファイル・フォーマットがサポートされてい る形式のいずれかと一致する場合はファイルが検出された形式でバッファリングするために読み込まれます。 マニュアル・ファイル形式はユーザーがサポートされているファイル形式のリストから明示的に望んだファイル形式を選 択することができます。ファイル形式がユーザーが選択した形式と一致しない場合、ファイルは不完全、又は不正 にロードされます。

チェックボックス Show "Load recent project"ダイアログはプログラムの開始でアプリケーションPG4UW起動時に 表示されるダイアログを設定します。ダイアログ Load recent project は最近のプロジェクト(プロジェクト履歴)のリ スト含んでいます。ユーザーは直ちにリストから選択し、そして、プロジェクト・ファイルをロード、又は、プロジェクト・フ ァイルをロードせずにダイアログを閉じることが出来ます。



File extensions[ファイル拡張子]

ファイル拡張子ページはファイル・マスクをセットすることが出来ます。

File format mask[ファイル・フォーマットのマスク]は全てのファイル・フォーマットに対する File/Save[ファイル/保 存]と File/Load[ファイル/ロード] ファイル・ウィンドウでのファイル・リストのためのフィルターとしてファイル名のマスクを 設定するのに使用します。マスクは少なくともワイルドカード(*, ?)を1つを含んで、そして、構文に正しく適応してい なければいけません。

ノート: 各ファイル・フォーマットに対して複数のマスクを指定することが出来ます。

セミコロンは拡張子のための区切り文字として使用されます。

לעלוו: Motorola: *.MOT;*.S19

2つのファイル・マスクを定義 *. MOT と*. S19 を定義 - Motorola ファイル形式。

プロジェクト・ファイルのデフォルトの拡張子は File/Load project[ファイル/プロジェクトのロード]と File/Save project[ファイル/プロジェクトのセーブ]ダイアログのデフォルトの拡張子として使用されるプロジェクト・ファイル拡張を 設定するために使用されます。

Buffer[パッファ]

Erase buffer before selecting of new device[新しいデバイスの選択前にパッファをイレース]動作を選択 することが出来ます。これは特定のアドレスのデータの正確な型を必要とする特別なデバイスのいくつかの種類に便 利に使用することができ、そして、データはこのデバイスのためにバッファにロードされたデータのファイルの一部ではあ りません。

バッファは選択したデバイス、又は、カスタム定義の値を持つ"blank"値はデフォルトで消去(フィル)することが出来ま す。これはグループボックス Erase value と Custom erase value[イレース値とカスタム・イレース値] 編集フィー ルドで制御することができます。

ノート:バッファの消去を行うために多くの時間を消費しますので大規模なデバイス(8MB 以上メガバイト)の場合この機能を使用することはお薦めしません。設定は PG4UW コンフィギュレーション・ファイルにセーブされます。プロジェクト・ファイルにはセーブされません。

Language[言語]

メニュー、ボタン、ダイアログ、情報とメッセージの様なユーザー・インターフェースのために他の言語を選択することが 出来ます。

Sound: [サウンド]

プログラムのサウンド・モードを選択するために使用します。プログラムは動作、即ち、デバイス(programming, verifying, reading, 等々)上の動作終了後、サウンドを鳴らします。プログラムは警告、又は、エラー・メッセージを 表示した時にも鳴らします。ユーザーはウィンドウズ・システム・サウンド(サウンド・カードがインスールさている必要が あります)、PC スピーカーからサウンドを選択することが出来ます。

パネル Allow sound for following actionsは次のオプションを含んでいます:

・チェック・ボックス正常操作 オンにするとデバイスの操作が正常に完了した後に音声が生成されます。チェックを外すとデバイスの操作が成 功した後に音は生成されません。

・チェック・ボックス・エラーの場合 チェックするとデバイスの操作がエラーで終了した後に音が生成されます。チェックを外すとデバイスの操作がエ ラーで終了した後、音は生成されません。



パネルProgrammer internal speaker sound settings[プログラマ内部スピーカーのサウンド設定]は内 蔵スピーカーを内蔵しているプログラマーのためにサウンドオプションを設定することができます。デバイスの動作 結果を示すために各デバイスの動作後に内蔵のプログラマ・スピーカから音が鳴り結果が良いか悪いかが表示 されます。

Errors[エラー]

このオプションはデバイス・ベリファイ・エラーをファイルにセーブするように設定できます。ベリファイ・エラーが発生した 時、45個までLogウィンドウに書き込まれます。もし、ユーザーがベリファイ・エラー(データの差異)とファイルにセーブ したい場合、セクション Save device verify errors to file[デバイス・ベリファイ・エラーをファイルにセーブ]のオプ ションで2つの方法の何れかでセットすることが出来ます:同じファイルに対する全てのベリファイ動作からの累積エラ ー、又は、最後のベリファイ動作からのみファイルにエラーをセーブ。ベリファイ・エラーは指定された名前でError file name[エラー・ファイル名] 編集ボックスによりファイルにセーブされます:

- オプション No(デフォルト) ベリファイ・エラーをファイルにセーブするのを無効にします。エラーは画面に表示されるだけです。
- オプション New 最後のベリファイ・アクションからファイルにベリファイ・エラーを保存します。最初に書き 込む前に新しいベリファイ・アクションの書き込み前にファイルが削除され新しいものとして作成されます。
- オプション Append 全てのベリファイ動作からのベリファイ・エラーが同じファイルに累積されます。もし、ファイル が存在しない場合は、新しいファイルが作成されます。

ボックス Error report file size limit[エラー・レポート・ファイル・サイズ制限]はファイルにセーブされるベリファイ・ エラーの最大数をセッティング出来ます。次のオプションが含まれています:

 チェックボックス Stop verification after max. number of errors reached[最大エラー数に達した後、ベリファ イを停止]

チェックにされていますと最大数に達した後にベリファイ・アクションが完了しエラー数がファイルに書き込ま れます。チェックにされていないと、全てのベリファイ・エラーがファイルにセーブされます。

編集ボックス Max. number of errors [エラーの最大数]は1回のベリファイ操作でエラー・ファイルに書き込むことができるベリファイ・エラーの数を指定します。

Log file[ログ・ファイル]

このオプションはLogウィンドウの使用に関連付けられます。ログ・ウィンドウの全てのレポートもログ・ファイルに 書き込むことができます。ログファイル名はデフォルトで "Report.rep"です。制御プログラムはこのファイルをログ ファイル名編集ボックスで指定した名前とディレクトリで作成します。 以下のログ・ファイル・オプションが利用できます:

- No デフォルトではLogウィンドウの内容はLogファイルにコピーされません。即ち、全レポートはLogウィンドウのみに表示されます。
- New 毎コントロール・プログラム開始の間に古いログ・ファイルは削除され新しいファイルが作成されます。
- Append 既存のログ・ファイルにLOGウィンドウ・リポートを追加し、ファイルがない場合は新しいファイルを作成します。

チェックボックス Add date information to Log file name[日付の情報をログファイル名に追加]はLog file name[Logファイル名] 編集ボックスでユーザーが指定した日付情報をセットすることが出来ます。チェックボックス にチェックが入れられている時、プログラムは自動的に次のルールでユーザー指定のLogファイル名に現在の日付 を追加します:

ユーザーが指定したログ・ファイル名がフォーマット持っている場合:

<user_log_file_name>.<log_file_extension>

日付が追加された名前は:

<user_log_file_name><-yyyy-mmm-dd>.<log_file_extension>
 日付を表す新しい部分は yyyy-year、mmm-monthとdd-dayで構成されます。



サンプル: ユーザーが指定したLogファイル名:c:\logs\myfile.log 追加された日付の最後のログ·ファイル名はこのようになります(2016年11月7日の場合): c:\logs\myfile-2016-nov-07.log

日付情報の前にプリフィックス無しでログ・ファイル名を付けたい場合、ログ・ファイル名を以下として指定すること ができます:

<log_file_extension> - .(ドット)はファイル名の最初

サンプル: ユーザー指定ログ・ファイル名: c:\logs\.log

追加された日付の最後のログ・ファイル名はこのようになります(2016年11月7日の場合): c:\logs\2006-nov-07.log

アドバンスド・オプション Logファイル・サイズ制限の利用について:

- オプション Use Log file text truncating when file size limit is reached[ファイル・サイズの上限に 達したときにログ・テキスト切り捨てのファイルを使用] - チェックが入っている時、ログ・ファイルのサイズ制限が オン。これはログ・ファイルのサイズが指定された値に達した時にログファイルに含まれているテキストの一部が 切り捨てられることを意味します。オプションにチェックが入っていない時、ログ・ファイルのサイズは無制限で PCのフリー・ディスク容量のみにより制限されます。
- オプション Maximum Log file size[最大ログ・ファイル・サイズ]はkB単位でログ・ファイルの最大サイズを指定します。
- オプション Amount of truncated text[切り捨てられるテキストの量]は最大ログ・ファイル・サイズに達した後に切り捨てられるログ・ファイル・テキストの%を指定することが出来ます。大きい値はより多くのテキストがログ・ファイルから切り捨てられる(削除)されることを意味します。

ログ・ファイル設定はOptions/Save options[オプション/オプションをセーブ]によってディスクにセーブすることが 出来ます。

Job Report

ジョブ・レポートは最近のデバイスで行った操作の概要説明を表します。Jobはプロジェクト・ファイルに関連付け られ、、そして、ロード・プロジェクトで開始された操作から新しいプロジェクトのローディング、又は、プログラム PG4UWのクローズ迄の情報です。

ジョブ・レポートは次の情報を含みます:

- プロジェクト名
- プロジェクト日付
- プロテクト・モード・ステータス
- PG4UWソフトウエア・バージョンn
- プログラマー・タイプとシリアル番号
- ジョブの実行の開始時間(ロード・プロジェクト操作がおこなれた時間)
- ジョブが実行された終了時間(ジョブ・レポートの作成の時間)
- デバイス名
- デバイス・タイプ
- チェックサム
- デバイス操作オプション
- シリアライゼーション情報
- 統計情報

ジョブ・レポートは次の場合に作成されます:

- ユーザー・コマンドのロード・プロジェクトが選択された時
- 閉じられる、又は、切断されるプログラマーのサイトが選択された時
- PG4UWを終了する時



- デバイス・カウント・ダウンが0になった時(スタータス完了)
- ・ ユーザーによって、メニュー"File | Job Report"が使用された時

ジョブレポートは最近ロードされたプロジェクト・ファイルのために、合計の統計値が0より大きい時のみ生成されます。 これは少なくとも1つのデバイス操作(program, verify,...)が行われなければならないことを意味します。

次のオプションがジョブ・レポートのために利用できます

チェックボックス Enable Job Report function[ジョブ・レポート機能を有効にする] - チェックされた時、ジョブ・レポート機能がアクティブ(有効)にされます。そうでない場合はジョブレポート機能は無効にされています。

チェックボックス Automatically save Job Report file[ジョブ・レポート・ファイルを自動的にセーブする] - チェ ックされた時、ジョブレポートは編集フィールド・ジョブレポート・ディレクトリーで指定したディレクトリーに自動的に 保存され、そして、次のファイル名で作成されます:

job_report_<ordnum>_<prjname>.jrp

この場合

<ordnum>はファイルの10進数の順序です。もし、同じ名前の既存のレポート・ファイルが存在する場合、新しい レポート・ファイルの順序は既存のファイルにインクリメントされます。 <priname>は最近使用したプロジェクトのプロジェクト・ファイル名で、プロジェクト・ファイル名の拡張子はありませ

cprjname>は最近使用したノロジェクトのノロジェクト・ノアイル名で、ノロジェクト・ノアイル名の拡張士はありません。

Example 1:

プロジェクト・ファイルc .\myproject.eprjを使用し、ジョブ・レポートのディレクトリをd .\job_reports.\に設定してみま しょう。ジョブレポート・ディレクトリにはレポート・ファイルはありません。

最後のジョブ・レポートのファイル名は: d:\job_reports\job_report_000_myproject.jrp

Example 2:

Example 1からの条件を使用し、しかし、1つのレポート・ファイルが既にあると仮定します。 このファイルの名前は d:\job reports\job report 000 myproject.jrp

最終的に新しいレポートのジョブ・レポート・ファイル名は: d:\job_reports\job_report_001_myproject.jrp ノート:ファイル名に含まれている番号の順番が1つインクリメントされています。

Automatically save Job Report file[自動的にジョブ・レポート・ファイルをセーブ]にセッティングされて いる場合、新しくジョブ・レポートを生成する時にJob Reportダイアログは表示されません。新しく生成された ジョブ・レポートはダイアログやメッセージ無しでセーブされます(ファイルのセーブ中にエラーが起こらない場合)

もし、チェックボックスが Automatically save Job Report file[自動的にジョブ・レボート・ファイルをセーブ]のチ ェックが外されていますと、PG4UW は Job Report ダイアログは必要なら毎回表示されます。 Job Reportダイアログでユーザーはジョブ・レポートで行う操作を選択することができます。もし、何も選択しない場 合(ボタンを閉じる)、ジョブ・レポートはPG4UWログ・ウィンドウにのみ書かれます。

Automatic YES! [オートマチックYES!]

ユーザーはプログラマとソフトウエアがアクティブAutomatic YES!モードでプログラムされたデバイスの取り除きと新し いデバイスの装着を待つ時の状態の表示のデフォルト設定(PG4UWソフトウェアのプリセットとして)を無視することが出来ます。



ユーザーはプログラマとソフトウェアがアクティブAutomatic YES!モードでプログラムされたデバイスの取り除きと新し いデバイスの装着を待つ時の状態の表示のデフォルト設定(PG4UWソフトウェアのプリセットとして)を無視すること が出来ます。

デフォルト・セッティング(PG4UWソフトウェアのプリセットとして) - プログラマはデバイスがプログラムされた時とプログラ マが新しいデバイスに交換されるのを待っている時にその状態を表示します。マルチ-ソケット・プログラマではこの表 示はされません(quiet mode)。シングル-ソケット・プログラマはLEDビジー点滅によりこの状態を表示します。By LED Busy blinkingセッティングをご覧下さい。

表示なしク(Quite mode- クワイエット・モード) - プログラマはZIFソケットの数に関係なく(マルチ-プログラマ、シングル・ソケット・プログラマ)共にデバイスがプログラムされて新しいデバイスの装着を待つ時に状態を表示しません。デバイス操作の後、その操作の結果によりステータス LED error又は、OKの何れか1つのみが点灯します。この LEDはZIFソケットからデバイスが取り除かれるのを検知しますと直ちにオフになります.

By LED Busy blinking[LEDビジー点滅] - プログラマはZIFソケットの数に関係なく(マルチ-プログラマ、シング ル・ソケット・プログラマ)共にデバイスがプログラムされて新しいデバイスの装着を待つ時にLEDビジーで点滅すること で状態を表示します。デバイス操作の後、その操作の結果によりステータス LED error又は、OKの何れか1つが 点灯し、そして、LEDビジーが点滅します。もし、プログラムがZIFソケットからデバイスが取り除かれるのを検知しま すとLEDはオフになりますが、新しいデバイスで操作が繰り返すためにプログラムが用意していることを示すために LEDビジーは点滅します。プログラムがZIFソケットで(新しい)デバイスの1つ以上のピンを検知しますとLEDビジー は継続的に点灯します。この時点からプログラムは新しいデバイスの残りのピンが装着されるために要求された時 間待ちます。もし、要求された時間(デバイス装着完了時間)がオーバーフローしたり、デバイスが正しく装着された時、ステータ スLEDはオフになりデバイスでの新しい操作が開始されます。

Remote control[リモート・コントロール]

PG4UW コントロール・プログラムのリモート・コントロールは他のアプリケーションによる PG4UW アプリケーションのい くつかの機能を制御することが出来ます。これはマス・プロダクション・ハンドラー、又は、他のアプリケーションにデバイ ス・プログラマを統合するために大変適した機能です。

PG4UW を制御するリモート・アプリケーションはサーバーとして動作します。プログラム PG4UW はクライアントとし て動作します。PG4UW とリモート・コントロール・プログラムは TCP プロトコル経由で通信されます。 - これは PG4UW が 1 つの PC にインストールされ、そして、リモート・コントロール・アプリケーションが他の PC にイ ンストールされ、そして、これらの PC がネットワーク経由で接続します。

リモート・コントロールのためのデフォルト TCP 通信設定は:

Port: telnet Address: 127.0.0.1 又は、ローカル・ホスト

アドレス設定は PG4UW(クライアント)のみに適用されます。 ポート設定は PG4UW(クライアント)とサーバー・アプリケーションにも適用されます。

デフォルト設定は1台のPC(アドレス・ローカルホスト)上のリモートコントロールの使用を許可します。PG4UW(クライアント)とリモート・コントロール・サーバーを同じ PC にインストールする必要があります。

ノート:もし、ファイアウォールがシステムにインストールされている場合、ファイアウォールはリモート・コントロール・サー バー、又は、クライアントがスタートする時に警告メッセージを表示することが出来ます。ファイアウォールがリモート・ サーバー、又は、クライアントのネットワーク・アクセスの強化、又は、拒否を尋ねる質問を警告表示された時、 'Allow'オプションを選択して下さい、そうでないとリモート・コントロールが動作しません。勿論、指定されたアドレス とポートのみでリモート・サーバー/クライアント・アクセスを許可するためにより厳格な権限をファイアウォール・オプシ



ョンで指定することが出来ます。

PG4UW のリモート・コントロールとデモ用リモート・コントロール・アプリケーションの詳細については PG4UW がイン ストールされているディレクトリ内の\RemoteCtrl サブ・ディレクトリにあるアプリケーション・ノート remotemanual.pdfを参照してください。

リモート・コントロールのマニュアルは PG4UW のインストール中に作成された Windows スタート/プログラム・メニュ ーリンクからリモート・マニュアルにもあります。 *Remote control of PG4UW, user's manual 5. Using remote control with multiply programmers (multiprogramming) Restrictions:を特に注意してご覧下さい。弊社でのサポートは出来ません。

Other[その他]

プログラムの処理優先レベルをセットするために使用します。

システム内のより要求の厳しいアプリケーションの実行がある場合は優先レベルの設定はプログラマのパフォーマンス (デバイス・プログラミング時間)に影響を与えます。アプリケーション優先レベルを Low にセッティングした場合書き込 み時間が非常に遅くなりますので注意して下さい。

Tool buttons[ツール・ボタン], メイン・プログラム・ウィンドウでのツール・ボタン hint display[ヒントの表示]オプション を変更することが出来ます。パネル Start-up directory[起動ディレクトリ]はプログラムの起動時にディレクトリを選 択するモードを選択することが出来ます。デフォルトの起動ディレクトリはプログラムが呼び出されるディレクトリを意 味します。プログラムは最後に終了されたディレクトリはプログラムの最後に終了した最後のカレント・ディレクトリを 意味します。このディレクトリはディレクトリ履歴リストから最初のディレクトリを前提としています。

プログラマの動作結果LEDのカラー:

- ・標準カラー・スキーム (ERROR=red, BUSY=yellow)
- 前のカラー・スキーム (ERROR=yellow, BUSY=red)

ノート:新しいタイプのプログラのみでこれらの設定を利用出来ます。もし、メニューにセッティングが無い場合、又は、 メニューで編集を有効にならない場合、ご使用のプログラマーではLEDカラー・スキームのカスタマイズはサポートさ れていません。

ソフトウエアでの動作結果表示のカラー:

- 標準カラー・スキーム (ERROR=red, BUSY=yellow)
- プログラマのLEDに従う(ERROR=yellow, BUSY=red)

ノート: これらのセッティングは古いタイプのプログラマでのみ利用出来ます。

Options / View[オプション/ピュー]

ツール・バーのようなプログラム環境で違った要素を表示又は、非表示にするためにはビュー・メニュー・コマンドを 使います。

次のツールバーが現在利用可能:

Options / View / Main toolbar[オプション/ビュー/メイン・ツールバー]

メイン・ツール・バーの表示又は、非表示はこのコマンドを選びます。



Options / View / Additional toolbar[オプション/ビュー/アディショナル・ツールバー] アッディショナル・ツール・バーの表示又は、非表示はこのコマンドを選びます。

Options / View / Device options before device operation[オプション/ビュー/デバイス操作 前のデバイス・オプション|

デバイス操作が確認される前にデバイス・オプションの表示を有効/無効にするにはこのコマンドを選択します。

Options / Protected mode[オプション/プロテクト・モード]

Protected mode[プロテクト・モード] はプログラムの特別なモードです。プログラムがProtectedモードの時はバッファ、又は、デバイス設定の変更が出来る特定のプログラム操作とコマンドが無効にされます。Protectedモードは不用意にバッファ、又は、デバイス設定を変更することを防ぐために使用されます。Protectedモードは同じ種類のデバイスの大量のプログラミングに適しています。

Protectedモード機能はシングル・プログラミング制御ソフトウェアPG4UWとマルチ・プログラミング制御ソフトウエア PG4UWMCで独立して使用可能です。

PG4UW C™Protected E−/*

プログラムをProtectedモードに切り替える2つの方法があります:

1.メニュー・コマンド Options/Protected mode[オプション/プロテクト・モード]を使用。このコマンドはパスワード・ ダイアログを表示します。ユーザーはパスワードが正しいかを確認するために2度入力しなければいけません。パスワ ード確認後にプログラムはProtectedモードに切り替えられます。パスワードの入力はProtectedモードをスイッチ・ オフするためにも使用されます。

2. Protectedモードで以前にセーブされたプロジェクトを読み込む。詳しくはFile/Save project[ファイル/セーブ・プロジェクト]をご覧ください。

チェックボックス **Keep "Load project" operation allowed** はデフォルトではインアクティブにセットされています。 - それはLoad project operation[ロード・プロジェクト操作]ボタンとメニューはProtectedモードがアクティブな時は 無効にされます。

オプションが有効(チェックされている)になっている場合、Load project operation[ロード・プロジェクト操作]ボタンと メニューはProtectedモードで許可されます。

チェックボックス **Disable view/edit buffer**はデフォルトではインアクティブにセットされています。- それは View/Edit buffer[ビュー/バッファ編集] ボタンとメニューはProtectedモードがアクティブな時有効にされます。これは バッファの内容を見ることが出来ますが、編集は出来ません(Protectedモードがアクティブなため)。

Protectedモードでバッファの内容を見れないようにしたい場合はこのオプションをアクティベートして下さい。この場合、オプション Encrypt project file (with password)[暗号アルゴリズムを使った特殊なフォーマットでプロジェクトをセーブ]もアクティベートされることをお薦めします。詳しはFile/Save project[ファイル/プロジェクトをセーブ]をご覧下さい。

プロテクト・モードのための操作モード選択

オプション "Multi operation[マルチ操作]" モード - リード以外のデバイス操作(blank, verify, program, erase)の全てが利用できるプロテクト・モードの基本を現わしています。これは偶発的、又は、意図的なリード操作によってバッファ・データを変更してしまうことを防ぐ確実性を提供します。1つのプロジェクト(マルチ・プロジェクト)で 全てのサポートされているデバイス操作を行いたいときに便利です。

オプション "One operation[シングル操作]" モード - 全ての使用可能な1つの操作のみが有効されているプロテ クト・モードの拡張フォームを表します。このモードでは使用可能なすべての操作から1つの操作のみが有効にされ ます。オペレータが間違ったタイプのデバイス操作を実行できないようにするため確実性が向上します。



マルチ-プロジェクト・ウィザードを使って"One operation[シングル操作]"モードでセーブされた複数のプロジェクトを ビルドすることでコントロールSWのデバイス操作の標準でないフロー(例えば、Program+Verify+Verify+ Verify) も一緒にすることが出来ます。

プログラムをProtectedモードからNormalモードに切り替えるためにはメニュー・コマンドOptions/Normal mode[オプション/通常モード]を使って下さい。"Password required" ダイアログが現れます。Protectedモード に切り替えるために使ったのと同じパスワードを入力しなければいけません。

Protectedモードをキャンセルする他の方法はプログラムを閉じることです。次にプログラムを起動すれば Normal[通常]モードで開始します(唯一の例外はProtectedモードでセーブされたプロジェクトの名前でコマン ド・ライン・パラメータによりプロジェクトがロードされている場合)。

Protectedモードがアクティブな時、ソフトウエアはプログラマー・アクティビティ・ログの右上に**ラベル Protected** mode[プロテクト・モード]が表示されます。

他のオプション Require project file unique ID before first programming[最初のプログラム前にプロ ジェクト・ファイルのユニークなIDが必要]に付いての情報はコントロール・プログラムの下のボタン・ステータス行に プロジェクト・ファイル名の隣にラベル(ID)により表示されています。File/Save project[ファイル/プロジェクトのセー ブ]をご覧ください。

PG4UWMC C→Protected E-F

PG4UWMCのAdministrator ModeとOperator Mode (以前のProtected mode)

プログラムPG4UWMCはデフォルトでAdministrator Modeにセットされています。これはユーザーのためにブロッ クしている操作が適用されないことを意味します。しかし生産で、幾つかのメニュー・コマンドをブロックするのに適 しており、確実にするために、ユーザーは重要なプログラムの設定や構成を変更しません。オペレーター・モード はこの目的のために使用されます。

OperatorとAdministrator Modeは付いての更に詳しい情報はOptions/Switch to Operator Mode (PG4UWMC)を参照して下さい。

プログラムPG4UWMCはプログラムPG4UWに非常によく似たProtectedモードを持っています。違いは Protectedモードはメニュー・コマンドによりアクティベートすることが出来きますがプロジェクト・ファイルによりアクテ ィベートすることが出来ません。もう1つの違いは、PG4UWMCのProtectedモード設定はPG4UWMCが閉じ られている間にPG4UWMCのコフィギュレーション .iniファイルにセーブされます。次のPG4UWMC開始の 間 .iniファイルから取得した最新のProtectedモードの設定が使用されます。

PG4UWMCで使用できる1つのメニュー・コマンドがあります - **Options/Protected mode[オプション** /**Protectedモード]** - メニューOptions/Protected mode[オプション/Protectedモード]選択後、パスワード・ダイア ログが現れます。ユーザーはパスワードが正しいかを確認するために2度入力しなければいけません。パスワード確 認後にプログラムはProtectedモードに切り替えられます。

チェックボックス Keep "Load project" operation allowed はデフォルトではインアクティブにセットされていま す。- それはLoad project operation[ロード・プロジェクト操作]ボタンとメニューはProtectedモードがアクティブ な時は無効にされます。

オプションが有効(チェックされている)になっている場合、Load project operation[ロード・プロジェクト操作]ボタンと メニューはProtectedモードで許可されます。



プログラムをProtectedモードからNormalモードに切り替えるためにはメニュー・コマンド**Options/Normal** mode[オプション/通常モード]を使って下さい。Password required" ダイアログが現れます。Protectedモ ードに切り替えるために使ったのと同じパスワードを入力しなければいけません。

Protectedモードがアクティブな時、ラベル "Protected mode"がPG4UWMCメイン・ウィンドウのLog windows のトップの近くに見えます。

Jート: 時にProtectedモードがアクティブ状態からインアクティブ状態(Normal mode)に切り替えられる時、ある 種のコマンド(例えば、"Load project")は無効のまま保持されます。これはボタン Stop ALLをクリックすることで 解決することが出来ます。

Multi-projects[マルチ-プロジェクト]

Multi-projectはsub-projectsとmulti-projectの作成中に保存された情報やマルチ・プロジェクト自体に 基づいて、任意のデバイスで任意のシーケンス操作を実行できる特別な機能です。

実際にマルチ-プロジェクトを使用すると次のことが出来ます:

- 快適にマルチ-チップ・デバイスをプログラム
- 1個のデバイスでデバイス操作の任意のシーケンスを構成、実行(即ち、Program+Verify+Verify +Verify)

更に詳しくは操作モードの説明をご覧下さい.

マルチ-プロジェクトに関する基本事項:

- Multi-project file[マルチ-プロジェクト・ファイル]は全てのMulti-project[マルチ-プロジェクト]情報を含む特別なファイルです。マルチ-プロジェクト・ファイルには1個以上のプロジェクト・ファイルを含むことが出来ます。マルチ-プロジェクト(ファイル)に含まれるプロジェクトはsub-projectsと呼ばれます。
- Sub-project[サブ-プロジェクト]はマルチ-プロジェクト・ファイルのビルド中にマルチ-プロジェクト・ファイルに入れられたクラシック・プロジェクト・ファイルです。
- Project file[プロジェクト・ファイル] バッファ・データ、デバイス操作オプション、スペシャル・オプションとある レベルの安全機能が組み合わさった特別なタイプのファイルです。デバイスをどのように扱うかを完全に定 義します。1度セーブされると、いつでも再ロード出来、そして、操作を繰り返し確実に行えます。
- Multi-chip device[マルチ-チップ・デバイス]は2つ以上の独立したチップ(同じ、又は、各種タイプ)を持ったシングル・パッケージのデバイス
- Sub-device[サブ-デバイス] マルチチップ・デバイスの独立したパート。Sub-deviceはPG4UWデバイス・リストから選択することが出来ます。1度選択されますと、デバイス毎のアルゴリズムでアクセスします。 パーシャル・チップ単独で定義、テストとプロジェクト・ファイルにセーブすることが出来ます。
- Master device[マスター・デバイス] マルチチップ・デバイス・ユニットはsub-devicesで構成されています。マスター・デバイスもPG4UWのデバイス・リストから選択可能です。1度選択しますと、マルチ・プロジェクト・ウィザードを使用して個々のプロジェクト・ファイルからマルチ・プロジェクト・ファイルをビルドし保存/ロード/実行することができます。シングル・チップ・デバイスからビルトされたマルチ・プロジェクトの場合はマスター・デバイスは定義出来ません。
- Device operation[デバイス操作] 各操作はメニューから選択して直接実行することが出来、ツール・バー・ボタンでクリックするか、又は、リモート・コマンド (Blank, Read, Verify, Program, Erase)で呼び出すことも出来ます。これらの操作の幾つか(特にProgramとErase)は組み込まれたsub-operationを含み、Menu/Device/Device[メニュー/デバイス/デバイス]オプションで編集出来ます。
- Multi-project Wizard[マルチ-プロジェクト・ウィザード] マルチ-プロジェクト・ファイルをビルドするためのア シスタント機能です。ウィザードはユーザーがマルチ-プロジェクトに含むべきプロジェクトを選択し、そして、 1つのマルチ-プロジェクト・ファイルにそれらをセーブします。選択されたプロジェクト・ファイルを1つのマルチ-プロジェクト・ファイルにセーブするプロセスはマルチ-プロジェクト・ファイル・ビルディングと呼ばれます。ウィザ ードはマルチ-プロジェクトに含まれたプロジェクト(sub-devices)に従ってデバイス操作を開始することも 出来ます。更に詳しくは後述のマルチ-プロジェクト・ウィザードをご覧下さい。



Multi-project Wizard[マルチ-プロジェクト・ウィザート]

マルチ-プロジェクト・デバイスの操作にはマルチ-プロジェクト・ファイルが必要です。マルチ-プロジェクト・ファイル にはマスター・デバイスのsub-devices(chips)に関連する部分的なサブ・プロジェクトが含まれています。マル チ-プロジェクト・ウィザードでマルチ-プロジェクト・ファイルを作成することができます。ウィザードには以下のメイン 機能があります:

- 複数のsub-projectを選択、そして、最終的なマルチ-プロジェクトをビルド
- 既存のマルチ-プロジェクト・ファイルをロード*1
- 最新のマルチ-プロジェクトのデバイス操作を開始

ノート *1: 既存のマルチ-プロジェクト・ファイルはメニューFile\Load projectを使ってPG4UWのメイン・メニューから、又は、multi-prjコマンドをロードすることでマルチ-プロジェクト・ウィザードからロードすることが出来ます。

Multi-project Wizard[マルチ・プロジェクト・ウィザード]は次の制御を含んでいます:

- ・ボタンLoad multi-prj は既存のマルチ-プロジェクト・ファイルをロードするために使用されます。
- ボタンBuild Multi-project はテーブル Sub-projectsにリストされたプロジェクトを使って新しいマルチープロジェクトのビルドのために使用されます。
- ・Table 1: Sub-projects は最新のマルチープロジェクトに含まれるプロジェクトのリストを含んでいます。
- ボタンAdd project はTable 1にあるプロジェクト・ファイルのリストに新しいプロジェクト・ファイルを追加するの に使用されます。
- ボタンRemove project はTable 1にあるプロジェクト・ファイルのリストから選択されたプロジェクト・ファイルを 削除するために使用されます。
- ボタンMove up a Move downはTable 1にある選択されたプロジェクトを1つ上、又は、下に移動するため に使用されます。プロジェクトは最も上の行(#1)を最初としてここで指定されたシーケンス順に処理されます。
- ボタンHelp はこのhelpです。
- デバイス操作Blank, Verify, Program, Erase,又は、RunのボタンはテーブルSub-projectsにリストされている全てのチップ(sub-devices)の選択された操作を実行するために使用されます。
 - "Multi operation" modeで, 全ての利用出来る操作を1度に実行(各sub-device上で同じ操作)する ことが出来ます。
 - ・"One operation" modeで、マルチ・プロジェクトが構成されているプロジェクトによっては1つの操作だけを 実行(各sub-device上で同じ操作)が実行出来ます。また、各sub-projectがその1つの操作を実行する ことが出来ます。

マルチ-チップ・デバイス(シングル-チップ・デバイスと同様)をプログラムするためにマルチ-プロジェクトを使用する時は 次の2つの基本動作を行う必要があります。:

- ・マルチ-プロジェクト(又は、マルチ-プロジェクト・ファイル)の作成(ビルド)
- ・デバイス操作の実行のためにマルチ-プロジェクトを使用

Multi-project[マルチ-プロジェクト](又は、Multi-project file [マルチ-プロジェクト・ファイル])の作成(ビルト) マルチ-プロジェクト・ファイルの作成時は以下のステップを推奨します:

- "classic"プロジェクトを作成、マルチチップ・デバイスの各sub-deviceのための1つのプロジェクト。ジェネリック・デバイスのためのプロジェクトと同じ方法でプロジェクトを作成:
- ・マルチチップ・デバイスに搭載されている希望するチップに従いsub-deviceを選択 *1
- デバイスのパラメータをセット、設定し、そして、希望のデバイスのデータをPG4UWのLoad Fileコマンドでバッフ アにロード
- ・確認のためにデバイス上でデバイス操作を実行することでデバイスのテストを行って下さい。
- ・全てOKの場合、Save projectコマンドでプロジェクト・ファイルを作成することが出来ます。
- そのマルチ-プロジェクトを使用すべきマスター・マルチチップ・デバイスを選択します。マルチチップ・デバイスを選択後、マルチ-プロジェクト・ウィザードは自動的に開かれます。



- マルチ-プロジェクト・ウィザードのAdd projectボタンにより必要なプロジェクトを追加します。各プロジェクトはマル チチップ・デバイスの対応した1つのsub-deviceを表します。
- sub-project選択の完了後、最終的なマルチープロジェクト・ファイル作成のためにボタンBuild Multi-project を使用します。プログラムは新しいマルチ・プロジェクト・ファイルの名前を聞いて来ますので名前を付けて下さい。 最終的なマルチープロジェクト・ファイルはTable 1: Sub-projectsにリストされている全てのsub-projectsを含 んでいます。

Jート:全ての従来(classic)使用していたプロジェクト・ファイルからもマルチ・プロジェクトの作成が可能です。マスタ ー・デバイスとの関連付けは必須ではありません。ユーザーへの配慮のため1つのマルチ・プロジェクト内に正しいサ ブ・デバイス(sub-project)を結合する方法をサポートしているだけです。この機能は異なるプロジェクトが定義され ているJTAGチェーンのデバイスのISPプログラミングを使用する場合に特に役立ちます。.

マルチ-プロジェクト・ウィザードは次の動作の1つにより開くことが出来ます:

- PG4UWのSelect Deviceダイアログからマスター・マルチチップ・デバイスを選択
- 作成したマルチ-プロジェクト・ファイルをローディング
- PG4UWのメニューOptions | Multi-project Wizardから直接マルチ-プロジェクト・ウィザード・ダイアログを開く

*1 PG4UWデバイス・リストでのマスター・デバイスとサブ・デバイス・パーツの名前規則

Master-device: マルチチップのオリジナル部品名[パッケージ・タイプ] Sub-devices: マルチチップのオリジナル部品名[パッケージ・タイプ] (part1) マルチチップのオリジナル部品名[パッケージ・タイプ] (part2).

マルチチップのオリジナル部品名[パッケージ・タイプ] (part n)

サンプル:

Master-device: TV0057A002CAGD [FBGA107] Sub-devices #1 TV0057A002CAGD [FBGA107] (NAND) #2 TV0057A002CAGD [FBGA107] (NOR)

デバイス操作実行のためのMulti-project[マルチ-プロジェクト]の使用

既存マルチ-プロジェクト・ファイルの典型的な使用方法は次の順序です。

PG4UWでのシングル・プログラミング:

- PG4UWメイン・ウィンドウのメニュー・コマンドFile/Load project[ファイル/ロード・プロジェクト]によって作成した マルチープロジェクトをロードするか、又は、マルチープロジェクト・ウィザードでLoad multi-prjボタンを使用。マ ルチープロジェクトのローディング完了後、マルチープロジェクト・ウィザードは自動的に開きます。
- ・ウィザードで利用出来るデバイス操作ボタン(Blank, Verify, Program, Erase)の1つを使って希望するデバイス 操作を実行、最も使用されるのはプログラム・デバイス操作です。選択されたデバイス操作はマルチ・プロジェク トで定義された各sub-deviceのためのsub-projectのロードとその結果としてのsub-deviceのプログラミングの シーケンスとして実行されます。そして、これはマルチ・プロジェクトの主目的です - マルチチップ・デバイスの各チッ プのためのデバイス操作のシーケンスを自動で実行。このコンセプトのサイド・イフェクトは、そのデバイスの進捗イ ンジケーターは各sub-device操作の開始時に0にリセットされますので、マルチチップ操作の実行中にプログレ ス・バーが0数回に"ジャンプ"されているように見える動作になります。
- 全てのsub-devicesのプログラミングが完了(又は、エラー)後、標準の"Repeat"ダイアログが表示されます。プロ グラマのソケットからプログラムされたデバイスを取り除き、そして、新しいデバイスを挿着することが出来ます。
 "Repeat"ダイアログ上でYesボタンを押すか、又は、プログラマのYES!ボタンを押しますとマルチチップ・デバイスの プログラミング・シーケンスが再開されます。


 もしAutomatic YES!機能がオンされている場合、デバイス操作終了後にRepeatダイアログは表示されず Automatic YES!ウィンドウが表示されます。このウィンドウにはプログラマ・ソケット状態とプログラムされたデバ イスの取り除きと新しいデバイスの装着が表示されます。新しいデバイスの装着後、マルチチップ・デバイス操 作シーケンスが自動的に開始します。Automatic YES!の詳細はProgrammer/Automatic YES!を参 照して下さい。

PG4UWMCによるマルチ・プログラミング、又は、スタンドアローン・プログラマ

- Load projectメニューでマルチ-プロジェクトをロード
- ・利用出来るデバイス操作ボタン(Blank, Verify, Program, Erase)の1つを使って希望するデバイス操作を実行、 最も使用されるのはプログラム・デバイス操作です。選択されたデバイス操作はマルチ・プロジェクトで定義された 各sub-deviceのためのsub-projectのロードとその結果としてのsub-deviceのプログラミングのシーケンスとして 実行されます。そして、これはマルチ・プロジェクトの主目的です - マルチチップ・デバイスの各チップのためのデバイ ス操作のシーケンスを自動で実行。このコンセプトのサイド・イフェクトは、そのデバイスの進捗インジケーターは各 sub-device操作の開始時にOICリセットされますので、マルチチップ操作の実行中にプログレス・バーが0数回に "ジャンプ"されているように見える動作になります。。
- 全てのsub-devicesのプログラミングが完了(又は、エラー)後、PG4UWMCにデバイス操作の結果情報が表示 されます。プログラマのソケットからプログラムされたデバイスを取り除き、そして、新しいデバイスを挿着することが 出来ます。サイトに対する操作ボタンを押すか、又は、プログラマ・サイトのYES!ボタンを押しますとマルチチップ・ デバイスのプログラミング・シーケンスが再開されます。
- もしAutomatic YES!機能がオンされている場合、プログラマ・ソケット状態とプログラムされたデバイスの取り除き と新しいデバイスを装着後デバイス操作シーケンスが自動的に開始します。Automatic YES!の詳細は Programmer/ Automatic YES!を参照して下さい。

ノート:

- マルチ・プログラミング・モードではシリアライゼーションはサポートされていません。(シングル・プログラミングのみシリア ライゼーションをサポート)
- カウント-ダウン機能もサポートされていません。

Options / Save options[オプション/セーブ・オプション]

このコマンドはauto-save[自動保存]がオフになっていても、現在保存がサポートされているすべての 設定を保存します。次のオプションが保存されます: Optionsメニューのオプション、最後に選択され た10個のデバイス、ファイル履歴、メイン・プログラム・ウィンドウの位置とサイズ。

Help[ヘルプ(英文)]

メニューHelp[ヘルプ]にはサポートされているデバイスやプログラマ、プログラム・バージョンに関する情報を 表示するためのコマンドが含まれています。

<F1>キーを押すとヘルプにアクセスします。メニュー項目を選択して<F1>を押すと状況依存ヘルプにアクセスします。PG4UWがプログラマで操作を実行している間は<F1>は応答しません。

次のヘルプ項目が強調表示されます:

- ・現在のヘルプで参照されるキーを表す文章[英文]
- ・他の全ての重要な単語[英文]
- 現在のクロス・リファレンス[相互参照]。このクロス・リファレンスをクリックして詳細情報を入手してください。
 *英文



個々のメニュー・コマンドの詳細については統合オンライン・ヘルプを参照してください

ノート: 制御プログラムと共に継続的に更新されるため、このマニュアルに記載されていない情報が含まれている可 能性があります。ELNEC社ウェブ・サイト <u>www.elnec.com</u>又は、最新バージョンのHelpファイルを参照して下さ い。

Help / Supported devices[ヘルプ/サポートされているデバイス]

このマニュアルに記載されている情報はリリース時点での正確さを期しています が、ソフトウエア同様に全ての製品を継続的に改善しています。ヘルプシステ ムは制御プログラムと共に継続的に更新されるため、このマニュアルに記載され ていない情報が含まれている可能性があります。

このコマンドはサポートされている全てのプログラマの少なくとも1つのタイプによってサポートされている全てのデ バイスのリストを表示します。これは特に少なくとも1つのタイプのプログラマによってサポートされているデバイス を探したい場合に便利です。

デバイスの名前の前の"g_"という接頭辞が付いていればデバイスはマルチソケット・プログラマでサポートされています。

Help / Supported programmers. [ヘルプ/サポートされているプログラマ]

このコマンドはこのプログラムがサポートされているプログラマに関する情報を表示します。

Help / Device list (current programmer)) [ヘルプ/デバイス・リスト (現在使用のプロ グラマ)]

このコマンドは現在使用中のプログラマの全てのデバイス・リストを作成し、それを????? **DEV.txt**テキスト・ファ イルと????? **DEV.htm** HTMLファイルとして制御プログラムが実行されているディレクトリに保存します。キャラ クター?????は現在のプログラマの略称に置き換えてデバイス・リストが生成されます。

Help / Device list.[ヘルプ/デバイス・リスト (全プログラマ)]

このコマンドは全プログラマの全てのデバイス・リストを作成し、それを????? DEV.txtテキスト・ファイルと????? DEV.htm HTMLファイルとして制御プログラムが実行されているディレクトリに保存します。キャラクター????? は現在のプログラマの略称に置き換えてデバイス・リストが生成されます。

ノート: このコマンドが実行された後、制御プログラムは現在のデバイスに関する全ての情報を失います。 DEVICEメニューのいずれかの選択方法で希望のデバイスを再選択し直して下さい。





Help / Device list (cross reference)[ヘルプ・デバイス・リスト(クロス・リファレンス)] このコマンドは市販されておりこの制御プログラムでサポートされている全てのプログラマがサポートする全てのデ バイスの相互参照[クロス・リファレンス]リストを作成します。結果のリストはHTML形式であり以下のファイルで 構成されています:

- サポートされているデバイス製造元がリストされている1つのメインHTMLファイルTOP DEV.htm
- 各デバイス・メーカーのサポートされるデバイスのリストを含む部分的なHTMLファイル。

メインHTMLファイルはこのプログラマ向けの制御プログラムが置かれているディレクトリに置かれます。

部分HTMLファイルはプログラマのための制御プログラムが置かれているディレクトリに置かれたサブディレクト リDEV_HTMLに置かれます。

Programmer / Create problem report[ヘルプ/プロブレム・レポート作成]

コマンド プロブレム・レポート作成は特定の診断情報をログ・ウィンドウに書き込むために使用され、 結果としてログ・ウィンドウの内容をクリップボードにコピーします。ログ・ウィンドウの内容はクリップボー ドから任意のテキスト・エディタに配置できます。プロブレム・レポートは制御プログラムやプログラマに エラーが発生し、エラーの種類が自分で解決できず、プログラマの製造元に連絡しなければならない 場合に便利です。この場合、ユーザー顧客が自分の問題について製造元にメッセージを送るときに はプロブレム・レポートも送る必要があります。プロブレム・レポートはメーカーがエラーの理由をローカ ライズして早期に解決するのに必須のものです。

About[プログラムについて]

メニューからInfoコマンドを選択すると著作権とバージョン情報を示すウィンドウが表示されます。



PG4UWMC マルチ制御ソフトウェア



プログラムPG4UWMCは複数のシングル・ソケット・プログラマ、又は、同じコンピュータのUSBポートに接続された1つのマルチ・プログラミング対応プログラマでの完全並列同時デバイス・マルチ・プログラミングに使用されます。

PG4UWMCは大量生産オペレーションの容易な監視を簡単に行うことに焦点を当てています。 PG4UWMCのオペレーター・フレンドリーなユーザー・インターフェースは多くの強力な機能と使い易さを組み 合わせて重要でない詳細にオペレーターに負担をかけることなく全ての重要な動作と操作結果の概要を提 供します。

PG4UWMCはマルチ-プログラミング・システムを制御するためにプロジェクト・ファイルを使用します。プロジェ クト・ファイルはユーザー・データ、チップのプログラミング設定情報、チップ構成データ、オート・プログラミング・コ マンド・シーケンス等を含んでいます。プロジェクト・ファイルが正常に作成され、そして、技術者により確認さ れた上で全ての必要な情報が簡単な操作で行えますので、従って、オペレーターのエラーが最小化されま す。オプショナルのプロテクト・モードプロジェクト・ファイルの不要な変更を回避するためにプロジェクト・ファイ ルを設定することが出来ます。各チップはシリアル番号、設定とカリブレーション情報などの別のデータを用い てプログラムすることが出来ます。

プログラム PG4UWMCは次のメイン・ウィンドウで構成されています:

- メイン・ウィンドウ
- 設定ダイアログウィンドウ
- "Search for Programmers"ダイアログ・ウィンドウ

マルチ・プログラミングとPG4UWMCの使用方法の詳細についてはUSBインターフェースの同時マルチ・プログラミン グ・システム(プログラマ)のユーザー・マニュアルを参照してください。

PG4UWMCのメイン・パートの基本説明



PG4UWMCメイン・ウィンドウ



PG4UWMCのメイン・ウィンドウは次のパーツで構成されます:

Menu & Tools[メニューとツール]ボタン メニューとツールボタンは殆どのPG4UWMC機能にアクセスできます。

Tool「ツール」ボタン・セッティング

ボタンはPG4UWMCセッティング・ダイアログを開くために使用されます。セッティング・ダイアログは以下の通りです。

Panel[パネル] Site #1, Site #2,...

パネルは以下について表示します: ・選択されたプログラマのサイト ・プログラマ・サイトの動作状況 ・現在のデバイス操作状態と/又は、結果

各パネルはデバイス操作を開始するために使用されるボタンRun 又は、ボタンYES!を含んでいます。

ポックス Statistics

Statistics[統計]はプログラムされたデバイスと成功と失敗したデバイスの数について通知します。 ノート: 'Clear statistics' ボタンはサイト上で操作が実行されるまで無効にされます。操作を中止したい場合はボ タンStop All'を押して下さい。

Checksum

チェックサムは現在のプロジェクト・ファイルからロードされたデータのチェックサムを表示

パネル Status window

パネル Status windowは各サイトの現在の状態を知らせます。状態は Blank サイトは非アクティブ Ready サイトはアクティブで動作用意が出来ています。プログラマは接続されていますがデバイ ス操作は実行されていません。

other information[その他の情報] 現在実行されているデバイス操作、結果、プログラマ接続状態等々

Log window ステータス・ウィンドウの右側

Log windowはプログラマの接続/非接続、デバイス操作結果やその他の情報が含まれています。

ボタン Connect programmers

このボタンは選択されたプログラマ/プログラマのサイトの全てを接続するために使用されます。通常はPG4UWMC 開始後の最初のステップとして使用されます。

ボタン Disconnect programmers

このボタンは接続されている全てのプログラマ・サイトを切断し、プログラマのサイトの制御プログラムは閉じられます。 ボタンは接続されているプログラマで実行されているデバイス操作が行われていない場合にのみ適用されます。

ボタン Run <operation>

ボタンは全ての接続されているプログラマで同時にデバイス操作を開始するために使用されます。<operation>の 値は次のタイプの1つです: Program, Verify, Blank check, Erase.



ポタン Stop ALL

ボタンは全ての接続されているプログラマ・サイトで現在行われているデバイス操作を中止するために使用されます。

ポタン Help

ヘルプを表示するために使用します。

ボタン Start remote control of Programmer(s)

このボタンは自動化プログラマでのみ利用可能で、PG4UWMCの設定 ダイアログ/マルチプログラミング/プロジェクトのオプションでUse Site #1 project for all Sites[全てのサイトのためにサイト#1のプロジェクトを使用]をチェックにされている場合。これはPG4UWMCインターフェースのリモート・コントロールを開始するために使用されます。ボタン・キャプションでプログラマは最近接続に使用したプログラマの名前に置き換えられます。オプションはサイト上でロードされたプロジェクトに応じてモジュールの検出をアクティベートします。

PG4UWMCセッティング・ダイアログ

ltiprog	ramming Log	file Job Report	Sounds Automatic YES! Other	
Elnec	multiprog	ramming syste	em	
Site	S/N	Site name	Project file	Device in project file
BeeHir	ve204 (S/N:	1177-00457)		
√ #1	1180-02637	BeeHive204 Site	D:\24LC512_SOIC8.eprj	Microchip 24LC512 [SOIC8]
v #2	1180-02638	BeeHive204 Site		
V #3	1180-02639	BeeHive204 Site		
V #4	1180-02640	BeeHive204 Site		
Startup options Auto-connect Sites Force YESI for gang programming			Project options Ive one common project for all Sites Project file: D:\24LC512_SOIC8.eptj Device used in project: Microchip 24LC512 (SOIC8)	Browse
Addition	al command lin	e parameters for site	3:	

PG4UWMC Settings ダイアログは次のオプションをセット、又は、表示するために使用されます:

- プログラマ・サイトのためのセッティング情報を含んでいます:サイト番号,サイトのシリアル番号,サイトに関連したプロジェクト・ファイル
- ・チェック・ボックス Use one common project for all Sites[全てのサイトで1つの共通プロジェクトを使用]
- ・チェック・ボックス Auto-connect sites settings[自動接続サイト・セッティング]
- ・チェック・ボックス Force gang multiprogramming mode[強制ギャング・マルチ・プログラミング・モード]
- ・パネル Log fileセッティング
- •パネル Job Reportセッティング
- •パネル Automatic YES!セッティング
- •パネル Other[その他]

パネル Multiprogramming

PG4UWMCの上のコントロール・パネルは次の3つのコラムを含んでいますs:

- コラムSitesは指定されたサイト番号で個々のプログラマ・サイトを使ってサイト番号#1,#2,#3,#4を有効/無効にするチェック・ボックスを含みます。
- ・コラムSerial numberはプログラマ・サイトのシリアル番号に付いての情報



 コラムProject fileは各PG4UWの実行後にロードする個々のプロジェクトのセッティングのための編集行 Project: #1, Project: #2, ...Project: #4を含みます。プロジェクト・ファイル名は手動で入力出来ますが、ダイ アログSelect project file によって各プロジェクト編集行の右側におかれたボタン"..."上でクリックすることで各 サイトのために開くことが出来ます。もし、プロジェクト名編集行がブランクの場合、自動プロジェクト・ロードは行 われません。

チェックボックス Use one common project for all Sites [全てのサイトで1つの共通プロジェクトを使用]がサイト番号とプロジェクト・テーブルの下に置かれます。

同じデータで同じデバイス・タイプをプログラムする必要がある時、チェックボックスはチェックにされていなければいけません。

- チェック・ボックスがチェックにされている場合、サイト#1のためのプロジェクト・ファイルが全ての他のプログラマ・サイトのために使用されます。このモードでは全てのサイトはプロジェクト・データの同じシェアされたバッファを使用し同じデバイス・タイプをプログラムします。
- チェック・ボックスがチェックにされていない場合、各サイトはコラム Project file[プロジェクト・ファイル]のサイトのテ ーブル内の名前によって定義されたそのプロジェクト・ファイルを使用します。このモードでは各サイトは各サイトで 同時に異なるタイプのデバイスに異なるデータをプログラムすることができ、プロジェクト・データの独自のバッファを 使用します。

Auto-connect sitesはPG4UWMC開始後に覚えているサイトがある場合は接続され用意された状態になります

Force YES for gang multiprogramming mode[ギャング・マルチプログラミング・モードのための強制YES]

各プログラミング・サイトが独立して動作し、他のプログラミング・サイトが実行されている間にオペレーターがプログラ ムされたデバイスを再ロードできる時、マルチプログラマでのマルチプログラミング操作の標準モードはconcurrent multiprogramming mode[同時マルチプログラミング・モード]です。gang multiprogramming mode[ギャ ング・マルチプログラミング・モード]ではあらかじめ定義された操作を任意のYES!ボタンを押すことで同時に全ての プログラミング・サイト上で操作を開始します。

- ノート:
 - ・全てのアクティブ(現存し有効な)サイトで動作
- 何れかのサイトがビジーな間、開始はブロックされます。
- ・このモードではAutomatic YES!は無効にされます。

パネル Log file settings[ログ・ファイル・セッティング]はLogファイル・レポートのモードを設定するために使用します。

PG4UWMC Settings	×	J
Multiprogramming Log file Jo	b Report Sounds Automatic YES! Other	
00110101 00110010 01010101 01010101 0101010101 000111000	Log file settings Rewrite Log file Append Log file Log file: anovak VappD ata/Roaming/Elnec/Pg4uw/reportmc.log Browse Add date information to Log file name Else size limit Else size limit V Use Log files text truncating when file size limit is reached Maximum Log file size: 20000 kB (200 kB. 200000 kB) Amount of truncated text: 90% • • • •	
	V OK X Cancel 12 Help	J

Log fileはPG4UWMC制御プログラムの操作フローに関する情報を含むテキスト・ファイルです。これはロードされ るプロジェクト・ファイル、デバイス操作のタイプとデバイス操作結果についての情報を意味します。マルチ・プログラミ ング・システムは幾つかのログ・ファイルを生成します。プログラムPG4UWMCの1つのメイン・ログ・ファイルと実行中 の各プログラマ・サイトのログ・ファイルです。各サイトは独自のログ・ファイルを1つ持っています。サイトのログ・ファイル の名前は編集ボックスのログ・ファイルで指定したログ・ファイルの名前と同じプリフィックスを持っています。ファイル名 のプリフィックスは_#<Snum>形式のサイト番号が続きます。

例えば:

ューザーにより指定されたLogファイル名は: "report.log". するとLogファイルの名前は: - PG4UWMC メイン・ロ グ・ファイル名 - "report.log"

- Site's #1 Log file name "report_#1.log"
- Site's #2 Log file name "report_#2.log"
- Site's #3 Log file name "report_#3.log" 等々...

次のオプションをLogファイル作成のために設定することが出来ます。

オプションAppend Log fileはLogファイルの使用法をセットにします。LogファイルはPG4UWMCの最初のリスタート後に作成されます。PG4UWMCの全ての他の開始では、既存のLogファイルはプリザーブされ、そして、新しいデータは既存Logファイルに追加されます。

•オプションRewrite Log fileはLogファイルの使用法をセットにします。LogファイルはPG4UWMCの最初のリスタート後に作成されます。PG4UWMCの次の起動時には、既存のログ・ファイルが書き換えられ新しいログ・ファイルが作成されます。以前のログ・ファイルのデータは削除されます。

チェック・ボックス Add date information to Log file name[日付の情報をログファイル名に追加]はLog file name [Logファイル名] 追加するログ・ファイル名の編集ボックスでユーザーが指定した日付情報をセットすることが 出来ます。このチェック・ボックスにチェックが入れられている時、現在の日付文字列をユーザーが指定したログ・ファ イル名に次の規則で追加します:

ユーザーが指定したログ・ファイル名の形式が次の場合: <user_log_file_name>.<log_file_extension>

日付が追加された名前は: <user_log_file_name><-yyyy-mmm-dd>.<log_file_extension>



日付を表す新しい部分はyyyy - year, mmm - month and dd - dayで構成されます。

例えば: ユーザー指定ログ・ファイル名: c:\logs\myfile.log

追加された日付の最後のログ・ファイル名はこのようになります(2006年11月7日の場合): c:\logs\myfile-2006-nov-07.log

日付情報の前にプリフィックス無しでログ・ファイル名を使用したい場合、次の様にログ・ファイル名をとして指定 することができます:

.<log_file_extension> - dotはファイル名の最初

例えば: ユーザー指定ログ・ファイル名: c:\logs\.log

日付の追加された最後のログ・ファイル名はこのようになります(2006年11月7日の場合): c:\logs\2006-nov-07.log

Logファイル・サイズ制限に関する詳細オプションも利用可能:

- オプション Use Log file text truncating when file size limit is reached チェックが入っている時、ログ・ファイルのサイズ制限がオン。これはログ・ファイルのサイズが指定された値に達した時にログ・ファイルに含まれているテキストの一部が切り捨てられることを意味します。オプションにチェックが入っていない時、ログ・ファイルのサイズは無制限で PC のフリー・ディスク容量のみにより制限されます。
- •オプション Maximum Log file size[最大ログ・ファイル・サイズ]は kB 単位でログ・ファイルの最大サイズを指定します。

•オプション Amount of truncated text[切り捨てられたテキストの量]は最大ログ・ファイル・サイズに達した後に切り 捨てられるログ・ファイル・テキストの%を指定することが出来ます。大きい値はより多くのテキストがログ・ファイルから 切り捨てられる(削除)されることを意味します。

ノート: '+'で開始される行はログファイルに表示されますが、画面のログには表示されません。

共通情報:

Index of Programmer Site[プログラマ・サイトのインデックス]は明確に実行中の各プログラマ・サイトを定義する1から8の整数番号です。

Serial number of Programmer Site[プログラマ・サイトのシリアル番号]は使用されているプログラマ、又は、プログラマ・サイトを定義します。実際は希望するシリアル番号を持つプログラマ(サイト)を見つけるまでUSBバス上に 接続された全てのプログラマを検索します。シリアル番号の異なるプログラマー、又は、プログラマー・サイトは無視さ れます。PG4UWMCはプログラマ(サイト)が見つからない場合は"Not found"の表示と共にDEMOモードに設定 されます。

1つのコンピューターで8つの同じタイプのプログラマをサイトとして同時に実行することが出来ます。

Job Report設定はJobレポート使用のモードをセットするために使用されます。

ジョブ・レポートは最近のデバイスで行った操作の概要説明を表します。Jobはプロジェクト・ファイルに関連付けら れロード・プロジェクトで開始された操作から新しいプロジェクトのローディング、又は、プログラムPG4UWMCのクロ ーズ迄の情報です。

Job Report は次の情報を含んでいます:

・プロジェクト名

- ・プロジェクト日付
- ・プロテクト・モードの状態
- PG4UWMCソフトウエア・バージョン
- ・プログラマのタイプとシリアル番号
- ・Job実行の開始時間(ロード・プロジェクト操作が行われた時間)



- ・Jobの実行終了時間(Jobレポートが作成された時間)
- ・デバイス名
- ・デバイス・タイプ
- ・チェックサム
- ・デバイス操作オプション
- ・シリアライゼーション情報
- •統計情報

Job Reportは次の場合に生成されます:

- ・コマンドLoad project[プロジェクトのロード]が選択された場合
- ・プログラマ・サイトを閉じる、又は、切断が選択された場合
- PG4UWMCを閉じた場合
- ・デバイス・カウント・ダウン・カウンターが0に達した場合(ステータスの完了)
- ・ユーザーにより手動でメニュー"File | Job Report[ファイル|ジョブ・レポート]"が使用された場合

Job Report は最近ロードされたプロジェクト・ファイルのために、合計の統計値が 0 より大きい時のみ生成されます。

これは少なくとも1つのデバイス操作(program, verify,...)が行われなければならないことを意味します。

Job Reportダイアログの設定はタブJob ReportのダイアログPG4UWMC Settings (メニューOptions / Settings)です。次のオプションがJob Reportで利用出来ます:t:

fultiprogramming	Log file	Job Report	Sounds	Automatic Y	SI Oth	er				
Job Report										
Automatica	illy save Jo	ob Report file								
Job Report of	- lirectory:	C:\Users\ar	novak\Ap	pData\Roamir	g\Elnec'	Pg4uv	Browse			
					9	See Job F	eport			
							_	 		
							1 011]	6	

チェックボックス Automatically save Job Report file[ジョブ・レポート・ファイルを自動的にセーブする] - チェック された時、 ジョブ・レポートは編集フィールド・ジョブ・レポート・ディレクトリーで指定したディレクトリーに自動的に 保存され、そして、次のファイル名で作成されます:

job_report_<ordnum>_<prjname>.jrp

<ordnum> はファイルの 10 進数の順序です。もし、同じ名前の既存のレポート・ファイルが存在する場合、新し いレポート・ファイルの順序は既存のファイルにインクリメントされます。

<pr/>
cprjname>は最近使用したプロジェクトのプロジェクト・ファイル名で、プロジェクト・ファイル名の拡張子はありません。



例 1: プロジェクト・ファイル c:\myproject.eprjを使用し、ジョブ・レポートのためのディレクトリは d:\job_reports\ に セットされます。

ジョブ・レポート・ディレクトリにレポート・ファイルが無い場合、最後のジョブ・レポートのファイル名は: d:\job_reports\job_report_000_myproject.jrp

例 2: Example 1 からの条件を使用し、しかし、1 つのレポート・ファイルが既にあると仮定します。 このファイルの名前は d:\job_reports\job_report_000_myproject.jrp 最終的に新しいレポートのジョブ・レポート・ファイル名は: d:\job reports\job report 001 myproject.jrp

ノート:ファイル名に含まれている番号の順番が1つインクリメントされます。

Automatically save Job Report file[自動的にジョブ・レポート・ファイルをセーブ]セッティングに設定されている時、 ジョブ・レポートを生成する時にJob Reportダイアログは表示されません。新しく生成されたジョブ・レポートはダイア ログやメッセージ無しでセーブされます(ファイルのセーブ中にエラーが起こらない場合)。

もし、チェックボックスが Automatically save Job Report file[自動的にジョブ・レポート・ファイルをセーブ]のチェッ クが外されていますと、PG4UW は Job Report ダイアログは必要なら毎回表示されます。

Job Reportダイアログでユーザーはジョブ・レポートで行う操作を選択することができます。何も選択しない場合(ボ タンを閉じる)、ジョブ・レポートはPG4UWMCログ・ウィンドウにのみ書かれます: 典型的なJob Reportダイアログの例を以下に示します::

> X Job Report: Microchip 24LC512 [SOIC8] BeeHive204 (S/N: 1177-00457), Site #1 Overall statistics Elnec Job Report, created at 2013-02-01 (year-month-day) Project Name: 24LC512_SOIC8.eprj Project Path: D:\ Project Checskum: B5C1h (CRC-16) Project Date: 2011-09-22 11:47:54 Project Size: 1,97kB Project Created In: Pg4uw 2.82/09.2011 Project Author: mkarniz Protected Mode: Enabled Software Revision: Pq4uw ver. 2.95T/01.2013 Programmer: BeeHive204 (S/N: 1177-00457), Site #1 Programming Adapter: DIL8W/SOIC8 ZIF 150mil Job Start Time: 2013-02-01 12:58:35 Job End Time: 2013-02-01 12:59:29 Job Elapsed Time: 0 Hours 0 Minutes 54 Seconds Device Name: Microchip 24LC512 [SOTC8] Save. Print. X Close 60



Automatic YES! セッティング

Itiprogramming Log file Job Heport Sounds Automatic 1ES
Itomatic YESI settings Use settings according to the last loaded project file Enabled Pesponse time: Standard Pevice removal hold off time[in seconds]; Pevice insertion complete time[in seconds]; Suspend on error: Disabled

このモードではプログラムされたデバイスを取り除いて新しいデバイスをZIF ソケットに装着しますと最後の操作が自動的にリピートされます。プログラムが自動的に新しいデバイスの装着を検知し、最後に行った操作をキー又は、ボタンを押すことなく実行します。ZIFへのデバイスの装着は画面に表示されます。リピート操作の実行はZIFから/への装着/取り外しを待っている間に<Esc> キーを押すことでキャンセルされます。

この機能はある種のタイプのプログラマでは利用できません。

Use settings according to the last loaded project file[最後にロードされたプロジェクトによるセッティング を使用] – Automatic YES!オプションはプロジェクト・ファイルのセッティングによって設定されます。Automatic YES!の設定の項目の1つは使用されるプログラミング・アダプタに依存する'Pins of programmer's ZIF excluded from sensing'[検出から除外されたプログラマのZIFのピン]です。これは別のプログラマが同じデバイス で異なるプログラミング・アダプタを使用することが可能であるため、この設定はこの場合無視されログ・ウィンドウに は次のメッセージを見つけることが出来ます:

"None connected pins setting was not accepted due to different programming adapter. Please use automatic YES wizard again."[プログラミング・アダプタが異なるため接続されていないピンの設定が受付られませんでした。Automatic YESウィザードを再度使用して下さい]。もし、これがマスター・プログラミング・サイト(オプション'Use Site #1 project for all Sites'でPG4UWMCを実行している場合)、又は、前述のメッセージがログに書かれたプログラミング・サイトで起こった場合はProgrammer/Automatic YES![プログラマ/Automatic YES!]オプションでボタン'Setting Automatic YES! parameters'[Automatic YES!パラメータのセッティング]をクリックして下さい。

'Indication of state when software waits for a new device'[ソフトウェアが新しいデバイスを待っているときの 状態の表示]と'Enable sound when ready for a new device'[新しいデバイスの準備ができたらサウンドを有効 にする]のセッティングはプロジェクト・ファイルにストアされません。

Enabled - Automatic YES! 機能はPG4UWMCによりパラメータ・セットで全ての接続されているプログラミング・ サイトを有効にされます。

Disabled - Automatic YES! 機能は全ての接続されているプログラミング・サイトが無効にされます。もし、デバイスのプログラムで次の操作を開始するためにボタンYES!を使用する必要がある場合はこのセッティングを使用して下さい。

Response time - ZIFソケットへのチップ装着と選択されたデバイス操作の開始の間隔となります。もし、ZIFソケットでのチップの長いポジショニングが必要な場合はelongated response time[延長した応答時間]を選択して下さい。



Device removal hold off time - ZIFソケットからデバイスを取り除いた時とソフトウェアが新しいデバイスの装着 をソケットでチェックを開始する時の間の時間間隔です。この時間は秒間隔で1~120(デフォルト値は2秒)でなけ ればいけません。

Device insertion complete time - プログラムが不正に挿入されたデバイスを検出しない様にするために最初のピン(複数)が検知された後に全てのピンが適切に挿入されなければならない時間をセットすることが可能です。 この時間は秒間隔で1~120 (デフォルト値は5秒)でなければいけません。

Suspend on error - Automatic YES!機能でエラーが起こった時に一時停止して操作の結果を見るか、又は、 停止せずに続けるかを定義します。

ソフトウェアが新しいデバイスを待つ時の状態表示:

Not indicated (quiet mode) - プログラマはZIFソケットの数に関係なく、デバイスがプログラムされて新しいデ バイスの装着を待つ時に状態を表示しません。デバイス操作の後、その操作の結果によりステータス LED error、又は、OKの何れか1つのみが点灯します。このLEDはZIFソケットからデバイスが取り除かれるのを検知 しますと直ちにオフになります。

By LED Busy blinking[LEDビジー点滅] - プログラマはZIFソケット、プログラマの数に関係なく(マルチ-プログラ マ、シングル・ソケット・プログラマ)共にデバイスがプログラムされて新しいデバイスの装着を待つ時にLEDビジーで点 滅することで状態を表示します。デバイス操作の後、その操作の結果によりステータス LED error又は、OKの何 れか1つのみが点灯し、そして、LEDビジーが点滅します。もし、プログラムがZIFソケットからデバイスが取り除かれ るのを検知しますとLEDはオフになりますが、新しいデバイスで操作が繰り返すためにプログラムが用意していること を示すためにLEDビジーは点滅します。プログラムがZIFソケットで(新しい)デバイスの1つ以上のピンを検知しますと、 LEDビジーは連続して点灯します。この時点からプログラムは新しいデバイスの残りのピンが装着されるために要求 された時間待ちます。もし、要求された時間(デバイス装着完了時間)がオーバーフローしたり、デバイスが正しく装 着されなかった場合、プログラムはこの状態を示すためにLED Errorを点灯します。デバイスが正しく装着された時、 ステータスLEDはオフになり、デバイスでの新しい操作が開始されます。

Enable sound when ready for a new device[新しいデバイスの用意がされた時にサウンドを有効にする] - チェックされている時、もし、SWが完全な空のZIFソケットを検出し、そして、ZIFソケットに新しいデバイスを受け入 れる準備ができている場合には音が発せられます。

前のいずれかのオプションを選択してOKボタンで確認された場合、PG4UWMCは接続されている全てのプログラ ミング・サイトに選択した設定を送信します。また、もし、マスター・プログラミング・サイトでAutomatic YES!パラメー タをセットしていますとそれらのセッティングが全ての接続されたスレーブ・プログラミング・サイトとPG4UWMCに送信 されます。

Automatic YES! 機能についての詳しくはProgrammer / Automatic YES![プログラマ/オートマチックYES!].をご 覧ください.

その他

プログラマの動作結果のLEDの色: ・標準カラー・スキーム (ERROR=red, BUSY=yellow) ・旧タイプ・カラー・スキーム (ERROR=yellow, BUSY=red)

ノート: これらのセッティングはある種のプログラマのためのみにこれらの設定を利用出来ますもし、メニューにセッティングが無い場合、又は、メニューで編集を有効にならない場合、ご使用のプログラマではLEDカラー・スキームのカスタマイズはサポートされていません。



Timer refresh rate[タイマー・リフレッシュ・レート] はPG4UWMCプログラムが実行しているプログラマ・サイトから ステータス情報を要求する頻度を定義します。ステータス情報には現在のデバイス操作タイプ、進行状況、結果 等を意味します。現在のステータス情報がPG4UWMCのメイン・ウィンドウに表示されます。デフォルトのタイマーの リフレッシュ・レート値は200msです。もし、PG4UWMCの操作パネルに表示されるステータス情報の表示をより 早くリフレッシュしたい場合、短いリフレッシュ間隔を選択して下さい。より速いリフレッシュを使用している時に、もし、 システムのパフォーマンスが低下したことがわかった場合は、高速リフレッシュを使用する場合はリフレッシュ値を高く 設定してリフレッシュを少なくします。Pentium 4コンピュータではタイマーのリフレッシュ・レートに依存するパフォーマ ンスの低下は殆どありませんが、遅いコンピュータでは時々長い(より少ない)タイマー間隔を選択することは有用で す。

PG4UWMC "Search for Programmers[プログラマのサーチ]"

ローカル・コンピューター上をサーチ

```
プログラマ検索のこのモードはデフォルトでPG4UWMCのインストール後にアクティブです。もし、ネットワーク経
由で別のコンピュータに接続されたプログラマで操作したい場合はネットワーク・モードを試してみて下さい。
```

赤色のプログラマは存在することが期待されているいくつかのサイトがあることを示していますが見つけることが出来 ない状態を示しています。これらのサイトは"Not found"列にリストされています。そうでない場合は列は非表示に されていいます。

Please sele and click of Multiprogra	rogrammers ct multiprogramming system from list bellow n button Search to start search for Programmers. mming system:						
All Programmers/BeeHive204AP-AU/BeeHive204 - Search							
Search res	ults: 2 Programmers found (max. 8 sites allowed)						
Programm	er	Sites	Not found				
#1	BeeHive204 S/N: 1177-00456	#1#4					
#2	BeeHive204 S/N: 1177-00457	#5#8	#5				
	🖌 Accept 🛛 🚫 Can	cel	? Help				

Search in defined Programmers group on network[ネットワーク上の定義されたプロ グラマ・グループでサーチ]

PG4UWMCはネットワーク・モードに切り替えるとネットワーク・コンピュータ上のPG4UWを検索、開始、制御、及び、監視することができます。PG4UWMCとPG4UW間の通信は、各コンピュータ上で動作するPG4UWMC Network Agentを介して実現されます。ネットワーク上の全てのPG4UW、PG4UWMCネットワーク・エージェント とPG4UWMCの制御は同じバージョンでなければなりません。この機能は自動プログラマにのみ使用でき、主にハンドラー・マシンで使用することを意図しています





2台のコンピュータ上で実行されるリモート制御されたマルチプログラミング・システムの典型的な構成

インストレーション

インストール中、Network Mode[ネットワーク・モード]機能はデフォルトではインストールされません。コマンド-ライン・パラメータ /networkmode を使用してインストール手順を実行することでアクティベートする必要があります (例:Start/Run/C:\pg4uwarc.exe/networkmode)。

いくつかの初期画面の後、PG4UWMC Network Agent のインストールと Programmer Group の選択を含める オプションが表示されます。このインストールされたコンピュータが属する **Programmers グループの名前を定義**し てください。PG4UWMC Network Agent は Windows で始まるように設定されます。



コマンド-ライン・パラメータ /ネットワーク・モードでのインストール手順





インストール手順 - カスタマイズド



	Additional Tasks
Whi	ch additional tasks should be performed?
Sele PG4	ct the additional tasks you would like Setup to perform while installing Elnec RUW/PG4UWMC, then click Next.
V	Create a Start menu icon
V	Create a desktop icon
V	Install Multiprogramming control support for following programmers:
	- BeeHive208S, BeeHive8S
	- ReeHive 204 ReeHive 2040P ReeHive 2040P-011 ReeHive 4+ ReeHive 4
	 BeeHive204, BeeHive204AP, BeeHive204AP-AU, BeeHive4+, BeeHive4 BeeProg2, BeeProg2C, BeeProg2AP, BeeProg+, BeeProg
	- BeeHive204, BeeHive204AP, BeeHive204AP-AU, BeeHive4+, BeeHive4 - BeeProg2, BeeProg2C, BeeProg2AP, BeeProg+, BeeProg V Install PG4UWMC Network Agent
	- Beetrive204, Beetrive204AP, Beetrive204AP-AU, Beetrive4+, Beetrive4 - Beetrog22, Beetrog2C, Beetrog2AP, Beetrog+, Beetrog // Instal PG24UMK0 Network Agent Programmers group: <u>ProgrammersGroup1</u>
	- BeeHive204, BeeHive204AP, BeeHive204AP-AU, BeeHive4+, BeeHive4 - BeeHrog2, BeeHrog2C, BeeProg2AP, BeeProg - Install PG4UWIC Network Agent Programmers group: ProgrammersGroup1 Install LPT port driver (required for programmers connected via parallel LPT port)

PG4UWMC Network AgentのInstallationがチェックにされProgrammers group名が選択されたインストール 手順

この方法はPG4UWがネットワーク上の各コンピュータにインストールされ、Programmers group[プログラマ・グル ープ]で動作すると見なされる必要があります。

Programmers group内の各コンピュータはPG4UWMC Network Agentをバック・グラウンドで実行している必要があります。PG4UWMC Network Agentがインストール後に実行されていない場合は、Start menu / All Programs...から実行して下さい。

Main Options Network card(s)	
PG4UWMC Network Agent activity log	Programmers group: ProgrammersGroup
L0001: Welcome to PG4UWMC Network Agent version: 2.99nw01/08.	2013 .
.0002:	
0003: Loading configuration done.	
0004:	
0005: >> 02.08.2013, 13:19:24	
0006: Network card(s) detected:	
0007: Local Area Connection	
D008: Description: Realtek RTL8168C(P)/8111C(P) Family PCI-E Gir	gabit Ethernet NIC (NDIS 6.20)
0009: Physical Address (MAC): 1F-D0-9A-47-C1-00	
0010: DHCP Enabled: 0	
0011: IP: 192.168.0.21, Mask: 255.255.255.0	
0012: Gateway: 192.168.0.138	
0013: VirtualBox Host-Only Network	
0014: Description: VirtualBox Host-Only Ethernet Adapter	
0015: Physical Address (MAC): 00-27-00-50-6A-00	
0016: DHCP Enabled: 1	
0017: IP: , Mask: 255.255.0.0	
0018: >> 02.08.2013, 13:19:24	
.0019: Programmers group: ProgrammersGroup1	

PG4UWMC Network AgentのInstallationがチェックにされProgrammers group名を使用したインストール手順

各コンピュータでインストールが完了したら、PG4UWMCの初期設定に進むことができます

コンフィギュレーション

プログラミング・プロセス全体を制御するコンピュータでPG4UWMCを実行します。Menu /OptionsでNetwork modeをチェックします。





PG4UWMC Network AgentのInstallationがチェックにされProgrammers group名を使用したインストール 手順

ネットワーク上にいますので、ネットワーク・パスをプロジェクト・ファイルとログ・ファイルに設定する必要があります。



ネットワークからPG4UWMCプロジェクト読み込みの設定、ネットワーク・パスへのログの保存。 ここで定義されたProgrammers groupのネットワーク上で最初に検索を実行することができます。



- プログラマを探す
- 見つかったものを評価する
- 凡例をチェックする(何をすべきかを知るためにはヘルプを見て下さい)
- 制限を満たすために問題を解決する
- 希望のプログラマを有効、無効、移動、削除
- 変更、又は、キャンセルを適用する.

rogrammers in	programming machine of Programmers group: ProgrammersGroup1
E 🙀 Program	imers Site(s)
8	-PC, IP: 192.168.0.27, SW: 2.99nw01
i 🧌	BeeHive204AP, S/N: 1202-00001
H 🕫	BeeHive204AP-AU, S/N: 2339-00007
± 🕫	/ BeeHive204AP-AU, S/N: 2339-00009
Θ 📌 💼	-PC, IP: 192.168.0.21, SW: 2.99nw01
9 . ?	/ BeeHive204AP-AU, S/N: 2339-00002
egend New (com Missing (Com	ed to previous configuration) A PG4UW running process detected pared to previous configuration)
egend > New (conf ● Missing (conf essages	ed to previous configuration) PG4UW running process detected pared to previous configuration) Incorrect version of PG4UWMC Network Agen
egend New (correspondence) Missing (correspondence) essages ly eight sites on or	ed to previous configuration) pared to previous configuration) PG4UW running process detected Incorrect version of PG4UWMC Network Agen Move up Move down K Roov the computer allowed. (Please, remove some site(s) to meet this condition)

ネットワーク上のProgrammers group をサーチ

この時点からPG4UWMCを使用する作業はいつものように行う必要があります。

トラブル・シューティング

検索プログラマが期待どおりに終了しない場合は、以下を確認してください:

- Programmers group内の各コンピュータはPG4UWMC Network Agentと同じProgrammers groupを実行する必要があります。
- ファイアウォール設定がネットワーク通信を妨げている可能性があります。ファイアウォール・ルールをチェックするか、 又は、一時的にファイアウォールを無効にして下さい(自己責任で注意して行って下さい)。

Command line parameters[コマンド・ライン・パラメータ]

プログラム PG4UWMCは以下のコマンド・ライン・パラメータをサポートしています:

/prj:<file_name>

プリフィックス /prj...を付けずにプロジェクト・ファイル名を入力することでコマンドラインからプロジェクトをロードすること もできます。

サンプル:

pg4uwmc.exe c:\projects\myproject.eprj Makes load project file "c:\projects\myproject.eprj". /autoconnectsites コマンドはPG4UWMCが最近閉じられた時に使用された全のサイトに対してPG4UWMCの起動時に PG4UWMCにプログラマ・サイト(各サイトの制御プログラムPG4UWを開始)を強制的に接続させます。



/autoconnectsites

また、PG4UWMCの"Settings"ダイアログで利用出来る"Auto-connect Sites"の同等のオプションもあります。

Programmers supported by PG4UWMC[PG4UWMCによりサポートされてい るプログラマ]

現在サポートされているプログラマのリストはメニューのヘルプ/サポートされているプログラマによってPG4UWMC に表示されます。一般的に、PG4UWMCのサポートされているプログラマはUSB、又は、LANインタフェースを 備えた48ピン・ドライバ汎用プログラマです。また、全てのUSB接続マルチ・プログラミング・システムがサポートさ れています。PG4UWMCは1から8のプログラマ・サイトを扱うことが出来ます。1つのプログラマ・サイトは1つの ZIFソケットモジュールを意味します。

Troubleshooting[トラブルシューティング]

シリアル・ナンバー

追加プログラマの使用を成功させるためにはパネルのシリアル番号で使用される各プログラマの正しいシリアル番号 を指定する必要があります。シリアル番号のフィールドが空白の場合、プログラマ・サイトに対してアプリケーション PG4UWは起動しません。

PG4UWMCアプリケーションが"Search for programmers"ダイアログで接続されているプログラマを検索している 時、プログラマのシリアル番号が自動的に検出されます。ユーザーはシリアル番号を自分で指定する必要はありま せん(また指定することも出来ません)。

プログラマをサーチ中のCommunication error(s)

何らかの通信エラーが発生した場合は、全てのPG4UWアプリケーションとPG4UWMCを終了し、そして、 PG4UWMCを起動し"Connect programmers"ボタンをクリックして各サイトのPG4UWアプリケーションを起 動しプログラマを接続してください。

全てのプログラマが正しく接続されているが動作が不安定

プログラマとの通信がデバイス操作(例えば、デバイス・プログラミング)中に無作為に失われた場合、他のプログラム、 特に大量のシステム・リソースを消費するプログラム(マルチメディア、CAD、グラフィック・アプリケーション等)を終了し てください。

Jート: また、コンピュータの背面にあるコンピュータのUSBポートを使用し、マザーボードに直接接続することを お勧めします。ケーブルを介して間接的にコンピュータのマザーボードに接続されたコンピュータのUSBポートは 高速USB 2.0転送モードを使用する際には信頼できない可能性があります。この推奨はプログラマだけでなく 他のデバイスに対しても有効です。



Common notes 共通ノート



Maintenance[メンテナンス]

プログラマの高い信頼性を長期間維持するためにここでの説明と注意事項に従うことをお勧めします。 プログラマのメンテナンスは使用者の姿勢とその使用量に依存します。何れにしましても、以下の推奨事項が 一般的に受け入れられて実行されるべきです。

- プログラマを粉塵の多い場所で使用したり保管しないで下さい。
- 湿度はZIFソケットとプログラミング・モジュール・インターフェイス(PMI)・コネクタの塵や埃の沈降を促進しますので乾燥した場所で使用して下さい。
- 使用後は付属のダストカバー付きのプログラマのZIFソケットとプログラミング・モジュール・インターフェイス・コネ クタにカバーして下さい。
- プログラマは直射日光や熱源の近くに置かないようにして下さい。ファンが故障した場合は重要なダメージに繋が ります。

集中的な日常使用(プログラミング・センタ、生産現場)

日々のメンテナンス

プログラミング・モジュールのZIFソケットの状態と装着状態を確認して下さい。清潔で乾燥した圧縮空気でZIF ソケットやモジュールからゴミや埃を取り除きます。閉じた状態と開いた状態の両方でZIFソケットを清掃します。 **週毎のメンテナンス**

全てのプログラマ、又は、プログラミング・サイトに対してセルフテストを実行します

四半期毎のメンテナンス

柔らかい布の上でイソプロピル・アルコール、又は、工業用アルコールでプログラマの表面を静かに清掃します。 清潔で乾燥した圧縮空気を使ってプログラミング・モジュール・インターフェイス(PMI)・コネクターからゴミや埃 を取り除きます。

カリブレーション・テストを実行して下さい。

日々の使用(開発研究室、オフィス)

日々のメンテナンス

作業が終了したらプログラミング・モジュールのZIFソケットにカバーします。また、プログラミング・モジュールのZIFソケ ットを埃や汚れから保護することをお勧めします。

週毎のメンテナンス

プログラミング・モジュールのZIFソケットの状態と装着状態を確認してください。清潔で乾燥した圧縮空気でZIFソ ケットからゴミやほこりを取り除きます。閉じた状態と開いた状態の両方でZIFソケットをクリーニングします。

四半期毎のメンテナンス

プログラマ、又は、プログラミング・サイトに対してセルフテストを実行します。

半年毎のメンテナンス

柔らかい布の上でイソプロピル・アルコール、又は、工業用アルコールでプログラマの表面を静かに清掃します。 清潔で乾燥した圧縮空気を使ってプログラミング・モジュール・インターフェイス(PMI)・コネクターからゴミや埃 を取り除きます。

カリブレーション・テストを実行して下さい。

時々の使用

日々のメンテナンス

作業が終了したらプログラミング・モジュールのZIFソケットにカバーします。また、プログラミング・モジュールのZIFソケットを埃や汚れから保護することをお勧めします。

四半期毎のメンテナンス

プログラミング・モジュールのZIFソケットの状態と装着状態を確認してください。清潔で乾燥した圧縮空気でZIFソ ケットからゴミやほこりを取り除きます。閉じた状態と開いた状態の両方でZIFソケットをクリーニングします。



半年毎のメンテナンス

プログラマ、又は、プログラミング・サイトに対してセルフテストを実行します。 **年間でのメンテナンス** 柔らかい布の上でイソプロピル・アルコール、又は、工業用アルコールでプログラマの表面を静かに清掃します。 清潔で乾燥した圧縮空気を使ってプログラミング・モジュール・インターフェイス(PMI)・コネクターからゴミや埃 を取り除きます。

カリブレーション・テストを実行して下さい。

警告:

プログラミング・モジュールのZIFソケットは消耗品とみなされます。ZIFソケットの製造業者によって与えられたプ ログラミング・モジュールZIFソケットの保証された機械的ライフサイクルは一般的に5.000~10,000機械的サイ クル(アクチュエーション)であり、いくつかの特殊なBGAZIFソケットのライフサイクルは約500.000機械的サイク ルまで可能な場合があります。プログラムされるデバイス、環境、及び、ZIFソケットのメンテナンスはZIFソケット の実際の電気的寿命に直接影響を与えます。ZIFソケットの接触部が指で汚れたりすると、プログラミングが 失敗する可能性があるため、ZIFソケットの接触部から指を離してください。プログラミングの失敗回数が増え ていることに気付いた場合は、ZIFソケット、又は、ソケットコンバータを交換して下さい。

摩耗や汚れがありプログラマとの作業中に大量の故障を引き起こすZIFソケットには保証は適用されません。



Software[ソフトウェア]

PG4UWは全てのElnecプログラマに共通の制御プログラムです。従って、作業中にいくつかのアイテムを見つける ことが可能です;それらは、現在選択されているプログラマでない場合もありますので、注意して作業を行って下さ い。一部の特別なデバイス(例:Philips Coolrunnerファミリ)では外部DATファイルが必要です。これはCD上の 標準のPG4UW SWには含まれていません。これらのデバイスをプログラムする必要がある場合は www.elnec.comのダウンロード・セクションをご覧ください。

Command line parameters[コマンドライン・パラメータ]

特別なユーティリティpg4uwcmd.exeを使用してPG4UWのコマンドライン・パラメータを制御することをお勧めします。下位互換性のために、いくつかのコマンドライン・パラメータをpg4uw.exeで直接 使用することもできますが、より良い方法はpg4uwcmd.exeを使用することです。これはより多くの コマンドライン・コマンドをサポートし、ExitCodeを返す機能も備えています。(又、ErrorLevel)値 はコマンドライン・パラメータの実行の成功、又は、エラーの結果を示します。 PG4UWの pg4uwcmd.exeコマンドライン・コントローラーの使用の詳細については Remote command line control of PG4UW[PG4UWのJモート・コマンドライン制御] をご覧ください。 pg4uw.exeで直接使用できるコマンドライン・パラメータ

/Prj: <file_name></file_name>	プログラムの起動時、又は、プログラムがすでに実行中であってもプロジェクトのロードを
	強制的に実行します。 <file_name>は完全、又は、相対プロジェクト・ファイルのパス</file_name>
	と名前を意味します。
/Loadfile: <file_name></file_name>	プログラムの起動時、又は、プログラムが既に実行されている場合でも、ファイルを強制
	的 にロードします。 <file_name>はロードする必要があるファイルへの絶対パス、</file_name>
	又は、相対パスを意味しファイル・フォーマットは自動的に検出されます。



/Saveproject:<file_name> このコマンドは現在選択されているデバイス・タイプ、バッファの内容、設定をプロジェ クト・ファイルに保存するために使用されます。コマンド /Saveproject ...はPG4UW 制御プログラムでユーザーが選択したコマンドSave projectと同等です。

PG4UW制御プログラムでプロジェクトを保存するファイル名はWindowsの規則を満たさなければならないことに注 意して下さい。つまり、ファイル名にスペースが含まれている場合、コマンドライン・パラメータのファイル名は引用符で 囲む必要があります。

Examples:

/prj:c:\myfile.eprj c:\myfile.eprjという名前のプロジェクトファイルをロード

/loadfile:"c:\filename with spaces.bin" バッファに"c:\filename with spaces.bin"ファイルをロード

/Program[:switch] プログラムの開始時に"Program device[プログラム・デバイス]"操作を強制的に開始 するか、又は、プログラムが既に実行されている場合でも、次のオプション・スイッチのいず れかを使用できます: switch 'noquest' 問題なくデバイス・プログラミングの開始を強制します。

switch 'noanyquest'

t' 問題なくデバイス・プログラミングの開始を強制し、デバイス上の操作が完了した後、プ ログラムは"Repeat"操作ダイアログを表示せずメイン・プログラム・ウィンドウに直接行 きます。

Examples: /Program /Program:noquest /Program:noanvguest

- /Close このパラメータは/Programパラメータと共にのみ意味を持ち、デバイス・プログラミングが正常に終了 したら自動的にプログラムを閉じるようにします。
- /Close: always このパラメータは/Programパラメータと共にのみ意味を持ち、デバイスの操作が成功したかどう かに関わらずデバイスのプログラミングが終了した後に自動的にプログラムを終了させます。

/Eprom Flash Autoselect[:xx]

プログラムが起動している時、又は、プログラムがすでに実行中であっても自動的にEPROM、 又は、FLASHをIDで強制的に選択します。xxはZIFのデバイスのピン数を表します(有効な 28ピン、又は、32ピンのみ)。これは挿入テスト機能のない古いプログラマにのみ必要です。他 のプログラマではこの値は無視されます。

エグゼクティブ・コマンドライン・パラメータを使用するための基本的なルール:

- コマンドライン・パラメータは大文字と小文字が区別されません。
- コマンドライン・パラメータはプログラムの最初の起動時、又は、プログラムがすでに実行中の時に使用できます。
- プログラムがすでに実行されている場合、プログラムがビジーでない(プログラムで操作が現在実行されていない)場合にのみ、いずれかのコマンドライン操作が処理されます。プログラムは基本的な状態でなければいけません、即ち、メイン・プログラム・ウィンドウがフォーカスされ、モーダル・ダイアログが表示されず、メニュー・コマンドがオープンされていないか、又は、実行されていない状態でなければいけません。
- 複数のパラメータを同時に使用する時コマンドライン・パラメータの処理順序は次のようにしっかりと定義されます:





- 1. プロジェクトのロード (/Prj:...)
- 2. ファイルのロード (/Load file:...)
- 3. IDによるEPROM/Flash選択
- 4. プログラム・デバイス (/Program[:switch])
- 5. コントロール・プログラムを閉じる (パラメータ /Programと一緒にのみ/Close)

デモ・モードでプログラムPG4UWを起動するために使用できるコマンドライン・パラメータ

デモ・モードはプログラマ・デバイスが使用できない状況で役立ちます。 デモ・モードを使用するには[プログラマの検 索]ダイアログの[デモ]ボタンをクリックするか、コマンドライン・パラメータ /demoを使用します。 パラメータの推奨され る使用法は次のとおりです:

pg4uw.exe /demo /<programmer name>

<programmer name>はPG4UW制御プログラムで使用されている希望のプログラマ名に置き換える必要があります。

Remote command line control of PG4UW[PG4UWのリモート・コマンドライン・ コントロール]

PG4UWはコマンドライン(コマンドライン・パラメータ)からコマンドのセットを受け入れることができます。リモート・コン トロールはこれらのコマンドライン・パラメータでも実現できますが、より効率的な方法は特別なツール pg4uwcmd.exeを使用することです。これは多くの利点があります。主な利点はpg4uwcmdのサイズで pg4uwcmdを呼び出すとPG4UWを直接呼び出すよりもはるかに高速な応答が得られます。 プログラムpg4uwcmd.exeを以下に使用出来ます:

- 1. 指定されたコマンドライン・パラメータでPG4UWアプリケーションを起動
- 2. 既に実行中のPG4UWにコマンドライン・パラメータを強制的に適用

pg4uwcmd.exeの非常に良い機能はPG4UWのコマンドライン・パラメータの操作結果に応じたそのリターンコードです。

pg4uwcmd.exeの戻り値

PG4UWで処理されたコマンドライン・パラメータが成功した場合、pg4uwcmd.exeのExitCode(又は、 ErrorLevel)はゼロです。それ以外の場合ExitCode値は1以上です。プログラムpg4uwcmd.exeの戻り値はバッ チ・ファイルでテスト出来ます。

pg4uwcmd.exeで使用できるエグゼクティブ・コマンドライン・パラメータは次の通りです。

- /Prj:<file_name> プロジェクト・ファイルを読み込みます。パラメータ<file_name>は完全、又は、相対プロジェ クト・ファイルのパスと名前を意味します。
- /Loadfile:<file_name> ファイルを読み込みます。パラメータ<file_name>はロードする必要があるファイルへの 絶対パス、又は、相対パスを示します。ファイル形式は自動的に検出されます。

/Program[:switch] プログラムの起動時に自動的に"プログラム・デバイス"の動作を強制的に開始します。また、次のオプション・スイッチのいずれかを使用できます:

switch 'noquest' 問題なくデバイス・プログラミングを開始

switch 'noanyquest' 問題なくデバイス・プログラミングを開始し、デバイス上の操作が完了した後、プログラ ムは"Repeat"操作ダイアログを表示せずメイン・プログラム・ウィンドウに直接入ります。



Examples: /Program /Program:noguest /Program:noanyquest //Close このパラメータは /Programパラメータのみと共に意味を持ち、デバイスのプログラミングが終了した後(操 作が成功したか否かにかかわらず)プログラムPG4UWを自動的に閉じます。 /Saveproject:<file name> このコマンドは現在選択されているデバイス・タイプ、バッファの内容とコンフィギュレーションをプロジェクト・ ファイルに保存するために使用されます。Command / Saveproject ...はPG4UW制御プログラムでユ ーザーが選択したコマンドSave projectと同等です。 /Eprom Flash Autoselect[:xx] プログラマのZIFソケットに現在挿入されているチップから電子IDを読み取ることによりEPROM、又は、 FLASHのタイプを自動選択します。オプションのパラメータxxはZIF(有効ピン数は28pin、又は、 32pinのみ)のデバイスのピン数を示し、そして、挿入テスト機能のない古いプログラマにのみ必要です。 他のプログラマにとっては無視されるためxxパラメータは省略することが出来ます。 Examples: /Eprom Flash Autoselect /Eprom Flash Autoselect:32 /writebuffer:ADDR1:B11,B12,B13,B14,...,B1N[::ADDR2:B21,B22,B23,B24,...,B2M]... コマンド /writebufferはPG4UWメイン・アドレスの指定されたアドレスにByteのブロックを書き込むために 使用されます。Write bufferコマンドは必要な1ブロックのデータとオプショナルな他のブロック(複数可)の データ([...]でマークされた)を持っています。コマンドにスペースやタブを使用しないでください。バッファ・アド レスは常にバイト・アドレスとして定義されます。つまり、バッファ構成x16の場合、バッファ内のアドレス AAAAx16はコマンド /writebufferで2*AAAAx(8)として指定する必要があります。 Example 1: /writebuffer:7FF800:12,AB,C5,D4,7E,80 /writebuffer:7FF800:12.AB.C5.D4.7E.80 アドレス7FF800Hのバッファに6バイトの12H ABH C5H D4H 7EH 80Hを書き込みます。 アドレッシングは次のようになります: 最下位アドレスの最初のバイト バッファ・アドレス データ 7FF800H 12H 7FF801H ABH 7FF802H C5H 7FF803H D4H 7FH 7FF805H 7FF804H 80H

Example 2: /writebuffer:7FF800:12,AB,C5,D4,7E,80::FF0000:AB,CD,EF,43,21

_____/____/ データの最初のブロック データの2

データの2番目のブロック

データ2つのブロックをバッファに書き込みます。

データの最初のブロック - 6バイト12H ABH C5H D4H 7EH 80Hはサンプル1と同じ方法でアドレス 7FF800Hのバッファに書き込まれます。

データの第2ブロック - 5バイト ABH CDH EFH 43H 21HはアドレスFF0000Hのバッファに書き込まれます。



アドレッシングは次のようになります: 最下位アドレスの最初のバイト バッファ・アドレス データ FF0000H ABH FF0001H CDH FF0002H EFH FF0003H 43H FF0004H 21H

//writebufferex:INDEX:ADDR1:B11,B12,B13,B14,...,B1N[::ADDR2:B21,B22,B23,B24,...,B2M].. コマンド /writebufferexは指定されたアドレスにバイトのブロックをPG4UWのメイン・バッファへの書き込みに使用さ れます。このコマンドはもう1つのパラメータ - INDEXを除いて、コマンド /writebufferと非常によく似ています。 INDEXパラメータはデータが送信されるバッファの順序を指定します。メイン・バッファのインデックスは '1'です。最初 の二次バッファはインデックス'2'、等を持っています。セコンダリー・バッファはいくつかの種類のデバイスのみ(例: Microchip PIC16F628)のみ利用可能なことに注意して下さい。パラメータ buffindexによってインデックスされる バッファの種類はアプリケーションPG4UWのダイアログView/Editバッファ内のバッファの順序に依存します。例えば、 デバイスMicrochip PIC16F628にはラベル "Data EEPROM"付きの追加バッファがあります。このバッファは buffindex = 2が指定されている場合、この関数によってデータ書き込みのためにアクセスできます。

Example 1:

/writebufferex:1:7FF800:12,AB,C5,D4,7E,80 そのコマンドは以下のコマンドと同等です。 /writebuffer:1:7FF800:12,AB,C5,D4,7E,80 コマンド /writebuffer.についてのセクションで説明

Example 2:

/writebufferex:2:2F:12,AB,C5,D4,7E,80 このコマンドはアドレス2FHでのインデックス"2"を持つ2次バッファに6バイト 12H ABH C5H D4H 7EH 80Hを書き込みます。 アドレッシングは次のようになります: 最下位アドレスの最初のバイト バッファ・アドレス データ 00002FH 12H

000030H	AB⊢
000031H	C5H

000032H D4H 000033H 7EH 000034H 80H

エグゼクティブ・コマンドライン・パラメータを使用するための基本的なルール:

- 1. プログラム pg4uwcmd.exeはプログラムpg4uw.exeと同じディレクトリになければなりません。
- 2. pg4uwcmd.exeが呼び出された時にpg4uw.exeが実行されていない場合、自動的に起動します。
- 3. コマンドライン・パラメータでは大文字と小文字は区別されません。
- コマンドライン・パラメータはプログラムの最初の起動時、又は、プログラムがすでに実行中のときに使用できます。
- 5. プログラムがすでに実行中の場合、プログラムがビジーでない時(プログラムで現在実行中の操作がない)にのみ コマンド行操作のいずれかが処理されます。プログラムはベーシックな状態でなければいけません。即ち、メイン・ プログラム・ウィンドウがフォーカスされモーダルダイアログが表示されずメニュー・コマンドが開かれたり実行されたり することがない状態。
- 6.より多くのパラメータを一緒に使用する場合のコマンドライン・パラメータの処理順序は次のようにしっかりと定義されています:



step1 ロード・ファイル(/ Loadfile:...) step2 プロジェクトをロード(/ Prj:...) step3 EPROM/FLASH自動選択 step4 プログラム・デバイス(/Program[:switch]) step5 制御プログラムの終了 (パラメータ /Programと一緒にのみ /Close)

Example 1:

pg4uwcmd.exe /program:noanyquest /loadfile:c:\empfile.hex 以下の操作が実行されます:

- 1. pg4uw.exeを起動します(まだ実行していない場合)
- 2. ファイル c:\empfile.hexをロード
- 3. プログラムデバイスの動作を開始
- 4. pg4uwcmd.exeはまだ実行されており、定期的にpg4uw.exeの状態を確認しています。
- 5. デバイス・プログラミングが完了すると、pg4uwcmd.exeが閉じられ、ロードファイルとpg4uw.exeのデバイス・プログラミング結果に応じてExitCodeが返されます。全ての操作が成功すると、pg4uwcmd.exeは0を返し、それ以外の場合は値1、又は、それ以上を返します。

Example 2:

pg4uwcmd.exe /program:noanyquest /prj:c:\emproject.eprj 操作はサンプル1と同じです; Load file操作はLoad project file c:\emproject.eprjコマンドに置き換えられます。

Example 3:

pg4uwcmd.exeを使ってバッチ・ファイルでpg4uwcmd.exeの戻りコードをテストします。 rem ------ beginning of batch------@echo off rem Call application with wished parameters pg4uwcmd.exe /program:noanyquest /prj:c:\emproject.eprj rem Detect result of command line execution rem Variable ErrorLevel is tested, value 1 or greater means the error occurred if ErrorLevel 1 goto FAILURE echo Command line operation was successful goto BATCHEND :FAILURE echo Command line operation error(s) :BATCHEND echo. echo This is end of batch file (or continue) pause rem ----- end of batch -----

Example 4:

PG4UW制御プログラムが動作していてユーザが選択したデバイスを持っていると仮定します。データをPG4UWデ バイス・バッファにロードし、選択したデバイスの設定とバッファの内容をプロジェクト・ファイルに保存する必要がありま す。デバイスに必要なデータはファイル c:\15001-25001\file_10.binlに保存されています。 プロジェクト・ファイルはc:\projects\project 10.eprilに保存されます。

希望の動作を実現するには次のコマンドライン・パラメータを指定する必要があります:

pg4uwcmd.exe /loadfile:c:\15001-25001\file_10.bin /saveproject:c:\projects\project_10.eprj PG4UWがコマンドを受信すると、次の手順を実行します:

1. データ・ファイル c:\15001-25001\file 10.bin をロード

2. 現在選択されているデバイス設定を保存し、データをプロジェクト・ファイルにバッファリング

c:\projects\project_10.eprj



実行された操作の結果がOKの場合、PG4UWcmdアプリケーションはExitCode(又は、ErrorLevel)値0を 返します。エラーがある場合(ファイルをロード出来ない、又は、プロジェクト・ファイルに保存できない場合)、 PG4UWcmdアプリケーションはExitCode値同等か、又は、1以上のExitCode値を返します。

ノート:上記のコマンドを使用する場合、PG4UWがデバイスの操作、例えば、デバイス・プログラミング等を実行していないことを確認する必要があります。PG4UWがビジー状態の場合、コマンドを拒否し、エラー・ステータスを返します(ExitCode同等、又は、値1以上)。[

Hardware[ハードウエア]

警告:



Class A ITE notice このマニュアルに記載されているデバイスはクラスAの製品です。家庭環境ではこの製品は 電波干渉を引き起こす可能性があり、その場合、ユーザーは適切な処置を講ずる必要 があります。

ISP (In-System Programming)[ISP (イン-システム・プログ ラミング)]

定義

イン・システム・プログラミングではエンドシステム内部に配置されたデバイスのプログラミングと再プログラミングが可能 です。シンプルなインターフェースを使用してISPプログラマはデバイスとシリアル通信しチップ上の不揮発性メモリを 再プログラミングします。イン・システム・プログラミングはシステムからのチップの物理的除去を省けます。これによりラ ボでの開発中、及び、フィールド内のソフトウェア、又は、パラメータを更新する際に時間とコストを節約できます。 ターゲット・デバイスはシステム内にあるデバイス(マイクロコントローラ、PLD等)です。ターゲット・システムは物理的 なプリント回路基板(PCB)であり、イン・システムでプログラムされるデバイスを含みます。 ISPプログラマはプログラマでありイン・システム・プログラミング能力を有する(例えば、BeeHive204AP, BeeProg2AP, BeeHive204, BeeProg2.....等)です。

General rules for in-system programming[イン-システム・プログラミングー般規則]

PC、ISPプログラマ、ターゲット・デバイス、又は、ターゲット・システムの損傷を避けるために下記のルールを守ることをお薦めします:

- ターゲット・システム、ISPプログラマ、PCの共通アースポイントを確保
- ラップトップ、又は、一般的なアースポイントに接続されていない他のPCの場合:ラップトップから一般的なアースポイントにハードワイヤード接続します(例えば、VGAコネクタを使用).
- ターゲット・システムに接続されているデバイスは全て共通接地点に接続する必要があります。





Elnec ISPプログラマをターゲット・システムに接続する方法:

イン-システム・プログラミング中にISPプログラマとターゲット・システムの2つの電気装置を接続します。非適合な接続はこれらのデバイスを損傷する可能性があります。

ノート: 指示に従わないでシステム内プログラミング中にプログラマに損害を与えた場合、それは無条件操作による プログラマの損傷であり保証対象外です。

- 1.1.ISPプログラマとターゲット・デバイスの両方のデバイスをオフにします。
- 2. 全てのデバイスに同じGND電位を割り当てます。例えば、全てのデバイスのGNDをワイヤで接続して下さい。
- 3. ISPプログラマにISPケーブルの1本のコネクタを差し込み、プログラマをオンにしてプログラムを制御します。
- 4. 制御プログラムでターゲット・デバイスと操作オプションを選択します。
- 5. ターゲット・デバイス上でアクションを開始します(読み込み、プログラム)。
- 6. 制御プログラムの指示の後、他のISPケーブル・コネクタをターゲット・システムに接続し、それをオンにします。
- 7. 制御プログラムの指示の後、他のISPケーブル・コネクタをターゲット・システムから切り離し電源を切ります。
- 8. ターゲット・デバイスで別のアクションが必要な場合はステップ5に進みます。

ISPプログラム・デバイスでのターゲット・システムの設計に関する推奨事項

ターゲット・システムはイン-システム・プログラミングに使用される全ての信号をISPコネクタ経由でISPプログラマに 直接接続できるように設計する必要があります。ターゲット・システムがこれらの信号を他の機能に使用する場合 は、これらの信号を分離する必要があります。ターゲット・システムはイン-システム・プログラミング中にこれらの信号 に影響してはいけません。

イン-システム・プログラマブル・デバイスの場合、メーカーはアプリケーション・ノートを発行しています。これらのアプリ ケーション・ノートを考慮したElnecプログラマの設計は、適切なインシステム・プログラミングを可能にします。条件 はこれらのアプリケーション・ノートを厳密に尊重しています。ElnecがISPプログラマで使用するアプリケーション・ノー トはwww.elnec.comのセクションSupport / Application Notes. 以下の推奨される回線に関する注意事項をお読みください。

- D1ダイオードの目的はISPプログラマによって提供されるより高い電圧からターゲット回路を保護することです。
- ・ターゲットボードの電源が上記の5Vと異なる場合は、この電源電圧に応じてツェナー・ダイオード(D1) 電圧を選 択してください。
- ・ターゲット・デバイスとターゲット・システムを分離するために抵抗R1, R2, (R3)を使用することを推奨します。ISP プログラミングに必要なピンがターゲット・システムの入力である場合、抵抗による分離で十分であり抵抗でもロー パス・フィルタが可能です。ピンが出力の場合、抵抗を使用するとプログラミング時間が節約されます。勿論、分 離抵抗器R1, R2, (R3)は必要に応じてスイッチ、又は、ジャンパで置き換えることが出来ます。この場合、ター ゲット・デバイスのISPプログラミング中にスイッチ(ジャンパー)がオープンしていなければいけません。しかし、スイッチ (ジャンパー)を使用すると次の操作時間がプログラミング手順に追加されます。



アプリケーションノートの例

マイクロコントローラAtmel AVR。及び、AT89Sxxxシリーズ

このインタフェースはAtmelアプリケーション・ノートAVR910:イン-システム・プログラミングに対応しています。このア プリケーション・ノートではターゲット・システムの推奨ISPインターフェイス・コネクタ・レイアウトについて説明します。 (トップ・ビュー).

ATMEL AVR対応のエルネック推奨回路:



AT89Sxxx対応のエルネック推奨回路:





PICmicro® マイクロコントローラ

このインタフェースはマイクロチップのアプリケーション・ノートTB013, TB017, TB016:PIC16CXXX OTP(PIC12C5XX OTP)(PIC16F8Xフラッシュ)MCUを使用したICSP™の実装方法に対応しています。 このアプリケーション・ノートではイン-システム・プログラミング・デバイスとISPプログラマを使用するターゲット・システム の要件について説明します。以下の信号はPICmicro®マイクロ・コントローラのインシステム・プログラミングに使用 されます。

- MCLR\ / VPP リセット / プログラミング・モードへの切り替え
- RB6 (GP1) クロック
- RB7 (GP0) データ入力 / 出力
- VDD 電圧供給
- GND グラウンド

PICmicro® デバイスがプログラムされる時; ピン MCLR\ / VPPは約12Vに駆動されます。

従って、プログラマが提供するこの電圧からターゲット・システムを分離する必要があります。RB6、及び、RB7信 号はイン-システム・プログラミング用にPICmicro® によって使用されます。従って、ターゲット・システムはイン・システ ム・プログラミング中にこれらの信号に影響を与えてはなりません。限界値検証はプログラミング後に使用されます。 プログラマは最小、及び、最大電源の両方でプログラム・メモリの内容を確認する必要があるためプログラミング中 はPICmicro®のVDDピンをターゲット・システムの残りから絶縁する必要があります。

PICmicro対応のエルネック推奨回路:



Jート: 外部リセット回路が必要なのはVDDパワーアップ勾配が低すぎる場合のみです。

その他

BUSY LEDが点灯している間、情報ウィンドウを移動しないで下さい - 監視回路を起動して通信中のPCプログラマ・エラーの場合はプログラマを安全な状態に切り替えることができます。







Troubleshooting[トラブルシューティング]

Elnec社はユーザーが弊社製品を快適に使用されることを望んでいますが、それにも拘わらず、問題が発生する 可能性があります。

• プログラマ、又は、その制御プログラムPG4UWを正しく操作していないかも知れません。

同梱のマニュアルをすべて慎重にお読み下さい。恐らく、直ぐに必要な答えを見つけることが出来ると思います。

 上手く動作しない場合はプログラマとPG4UWを別のコンピュータにインストールしてみて下さい。システムが他の コンピュータで正常に動作している場合、最初の1台のPCに問題がある可能性がある場合もあります。両方のコ ンピュータの違いを比較します。

• 社内のより知識の豊富な担当者に聞いて下さい。.

• 既にプログラマをインストールした経験を持った人に尋ねて下さい。

 問題が解消されない場合は、先ず、作業中のLOGをセーブして下さい。その上でプログラマを購入した販売店に連絡するかElnecに直接メールでお問い合わせ下さい。ほとんどの問題は電子メールで解決出来ております。 連絡したい場合:

 E-mail - インターネットのウェブ・サイト Support -> Problem Reportからフォームを使用してフォームの最後に 記載されている指示に従って下さい。*重要: 1如何なる場合もLOGファイル(メニューのヘルプからドロップダウンで" プロブレム・レポート作成"をクリックすることでデスクトップ上に作成されます)を添付して下さい。プログラマのモデル 名、ソフトウェアのバージョン、及び、ターゲット・デバイスに関連すると考えられる全ての情報が必要ですので、上記 のLOGと共にお送り下さい。Eメールで地域の販売店に送付するか、又は、elnecドットコムで(迷惑メールになら ないようにし)に送付してください

プログラマが不良品であると診断された場合は、お住まいの国のElnecの代理店、又は、Elnecにご相談ください。パッケージに次のアイテムを慎重に入れて下さい:

・ 問題の製品

- ・プロブレム・レポートを事前にメール添付するか、又は、"DEVICE PROBLEM REPORT" フォームに入力
- 購入日証明になる何らかのコピー

上記項目なしではプログラマの修理を受け付けることは出来ません。

ノート:

"**DEVICE PROBLEM REPORT**" フォームはインターネット(<u>www.elnec.com</u>) で Support -> Problem Reportからフォームを使用出来ます。

サポートされていないターゲット・デバイスがある場合

プログラマ用の制御プログラムでサポートされていないターゲット・デバイスを使用する必要がある場合は、次の 手順に従って下さい:

• インターネットサイトの最新バージョンのコントロール・プログラムのデバイス・リストを見て下さい(ダウンロード、プログラマに対応するファイルを参照)。新しいターゲット・デバイスが既に日々の最新OnDemandバージョンに含まれている可能性があります。

•記載のない場合は、インターネットのElnecのウェブサイトのSupport-> **AlgOR**(New device support)フォームに入力して下さい。ターゲット・デバイスの詳細なデータシートとサンプルが必要な場合があります。


Warranty terms[保証期間]

製造元、Elnec s.r.o. Presov, Slovakiaは購入日から3年間(BeeHive304、BeeProg3)プログラマ、及び、 その部品、材料、及び、ワークマンシップの全ての無故障動作を保証します。製品が欠陥品であると診断され た場合、Elnec s.r.o. が欠陥部品を無償で修理または交換します。交換、及び/又は、プログラマ全体に使 用される部品は、元の保証期間内として保証されます。

プログラミング・モジュールの保証はZIFソケットの機械的寿命によって制限されます。特定のアダプ タでZIFソケットの寿命は異なります;特定のアダプタのZIFソケットの機械的寿命についての情報 はEInec社のWebサイトで見つけることができます:製品 - アダプタのプログラミング... - 希望の種 類のアダプタをクリックします。この保証はZIFソケットの機械的寿命のみを対象としています。この 保証は損耗には適用されません。例えば 埃によるより高い/未定義/接触抵抗の変動、又は、プロ グラムされたデバイスの鉛からの擦れによる接触抵抗。

保証期間内の修理の場合、顧客は購入日を証明しなければいけません。

保証期間はElnec社から直接プログラマを購入するお客様に有効です。Elnecの代理店の保証条件は対象 となる国の法律、又は、ディストリビューターの保証ポリシーによって異なる場合があります。

摩耗や機械的損傷を受けた製品には保証は適用されません。同様にElnecの許可を受けていない人物によって開封、改造された製品、又は、誤用、乱用されたり誤ったインストールによって故障した場合には保証は適用されません。

不当な修理の場合、材料、サービス時間、貨物の交換費用に応じて請求されます。エルネック、又は、その代 理店は不良品を修理、又は、交換するかどうかを決定し保証が適用されるかどうかを判断します。

Elnecは安定した信頼性の高いハードウェアとソフトウェアの開発に最善を尽くしています。Elnecはハードウェア及 び、ソフトウェアに"バグ"、エラー、又は、欠陥がないことを保証するものではありません。エルネックの責任は購入者 が支払った契約の正味価値に常に制限されます。

Elnecは以下の責任を負いません:

- ・製品の不適切な使用、又は、操作ミスに起因する損失
- ユーザー、又は、第三者が製品を変更、又は、改造しようとすることによる損害
- ・ ウイルス、ルートキット等による損害
- ・ハードウェアのエラー、又は、ソフトウェアの"バグ"によって引き起こされるそれ以上の損害、又は、結果的損害

例えば:利益の損失、第三者からのクライアントに対する請求、記録されたデータやファイルの破損、又は、使用 不可能による損失等

製造元:

- Elnec s. r. o., Jana Bottu 5, SK 08001 Presov, Slovakia
- **2**: +42151/77 34 328, 77 31 007, fax 77 32 797

www.elnec.com, e-mail (nospam version): elnec at elnec dot com