## 日本語ユーザー・マニュアル

# PIKPROG+

マイクロチップ **PICmicro** プログラマー

ELNEC s.r.o.
Presov, Slovakia
October 2003

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult manual on **www.elnec.com**.



ZLI-0017D

# Table of contents



*このマニュアルで使用れている表現*

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***
***ターゲット・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***
テンポラリーなデータのストレージに使用されるメモリー又は、ディスクの部分
***Printer port - プリンター・ポート***
プリンターの接続のために主に使用される PC のポート
双方向の場合はパラレル・ポートと呼ばれます。
***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。

# イントロダクション



## *製品の構成*

プログラマーをご使用頂く前に下記のすべてが入っていたか確認して下さい。内容が違った場合は、お買い上げ頂いた販売さきにご連絡下さい。

### *PIKPROG+ プログラマー*

- プログラマー本体
- プログラマーと PC 接続のための 25 ピン D-タイプ・コネクター
- 電源アダプター
- 自己診断のための診断 POD
- ZIF ソケット・カバー
- ユーザー・マニュアル
- コントロール・プログラムとその他のファイルの入ったフロッピー・ディスク又は、CD
- "PROGRAMMER PROBLEM REPORT" と "DEVICE PROBLEM REPORT"のリポート用紙
- レジストレーション・カード
- ペーパー・カートン・ケース

## *必要な PC の環境*

### *最低必要な PC の環境*

- PC 486 (ウィンドウズ版ソフトウエア)
- 16MB RAM 容量 (ウィンドウズ版ソフトウエア)
- CD ドライブ*インターネットでダウンロードしてご使用される場合は特に必要ではありません。
- ハードディスク：20 MB 空き容量
- OS:ウィンドウズ 3.xx, 95/98/ME/NT/2000/XP
- フリー・プリンター・ポート*他に何も接続されていないこと。

### *推奨する PC 環境*

- ペンティアム PC 100MHz 又は、以上
- 32 MB RAM 容量
- CD ドライブ
- 最低ハードディスク空き容量： 20 MB
- OS:  MS Windows 95/98/Me/NT/2000/XP
- 双方向プリンター・ポート*他に何も接続されていないこと。
- PCI bus, IEEE 1284 互換(ECP/EPP)のパラレル・ポート

***ノート:*** *便利にご使用頂くために, プリンターとプログラマーを同じ LPT ポートを使用しないですむように追加のプリンター・ポート(例えば、LPT2)をマルチ I/O カードを使って用意されることをお薦めします。*

# クイック・スタート



### *プログラマー・ハードウエアのインストール*

- PC とプログラマーのスイッチをオフにしてください。
- 添付されているケーブルを使って PC のプリンター・ポートとプログラマーを接続します。
- PC のスイッチを入れます。
- 電源ケーブルのコネクターをプログラマーに接続します。

### *プログラマー・ソフトウエアのインストール*

CD から(ウィンドウズの場合は: Setup.exe) プログラムのインストールを実行してください。下記の web サイトから最新のソフトウエアをダウンロードされることを推奨致します。 http://www.datadynamics.co.jp/download.html 又は、 http://www.elnec.com/downloads.php から最新のソフトウエアをダウンロードしてください。

### *プログラマー・ソフトウエアの使用*

コントロール・プログラマーを実行するために PG4UW.EXE (ウィンドウズの場合)を立ち上げて下さい。 メニューで **Device** はデバイス操作コマンドを含んでいます。メニュー **File** はファイルとディレクトリーのためのコマンドを含んでいます。メニューで **Buffer** はバッファ操作に使用されます。

### *デバイスのプログラミング– クイック・スタート*

ターゲット・デバイスのタイプを選択し、その名前そして/又は、マニファクチャラーを入力するためにホット・キー **<Alt+F5>** を使用します。もし、既にあるデバイスをコピーしたいときは、そのデバイスをプログラマーの ZIF ソケットに装着し、そして、**<F7>**キーを押して下さい。もし、ディスクからのデータをターゲット・デバイスにプログラムしたいときは、**<F3>**キーを押し、正しいファイルをバッファに読み込んで下さい。そして、ターゲット・デバイスを ZIF ソケットに装着して下さい。デバイスがブランクかどうかをチェックするために**<F6>**キを押してください。そこで**<F9>**キーを押して、デバイスをプログラムすることが出来ます。プログラミングの後で**<F8>**キーを押すことで、追加のベフィファイを実行することが出来ます。

# PIKPROG+



*イントロダクション*

**PIKPROG+**は Microchip™ PICmicro® シリーズのマイクロコントーラー(8-40 pins) をパラレルとシリアルのアルゴリズムでサポートしています。
また、IIC (24Cxx), Microwire (93Cxx) と SPI (25Cxx)インターフェース・タイプのシリアル EEPROM もサポートしています。
内蔵のインサーキット・シリアル・プログラミング(ISP) コネクターを使って PICmicro® ファミリーのマイクロコントローラをシリアル・アルゴリズムでプログラムすることもできます。

PIKPROG+のインターフェースは IBM 互換の PC, AT 又は、同等以上のポータブル又は、デスクトップ PC で動作します。プログラマーは IEEE1284 パラレル(ECP/EPP)・ポートを使用しますので、特別なカードは必要としません。

プログラミングのベリファイは VCCP のマージナル・レベルにより行われますのでプログラミング不良をなくし、データ保持が保障されます。

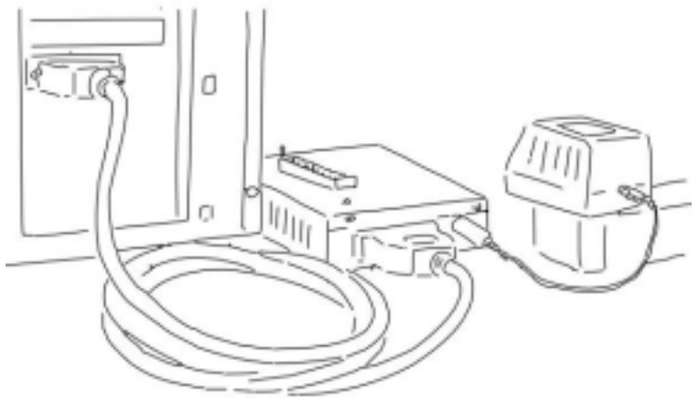
40 のパワフルな TTL ピンドライバーが H/L/プルアップ/プルダウンとソケットの各ピンを読む能力があります。ピンドライバーはオーバーシュートなしで信号をデバイスに供給します。また、1.8V まで操作することが出来ます。

PIKPROG+ はプルダウン・メニュー、ホット・キーとオンライン・ヘルプを持った**使い易いコントロール・プログラム**によりデバイスをクラス、マニファクチャラー又は、マニファクチャラー名をパーツ番号により選択することが出来ます。標準のデバイス操作機能(読み出し, ブランク・チェック, プログラム, ベリファイ) はいくつかのテスト機能と一緒に完了されます。プログラムは自動ファイル・フォーマットの検知と変換を含む、バッファとファイルの使用機能があります。

ソフトウエアは **Auto-increment 機能** を提供していますので、プログラムされるデバイスにシリアル番号を個々に割り当てることが出来ます。この機能は単にバッファ内のシリアル番号をソケットに新しいデバイスが挿入される度にインクレメントして行きます。さらに、この機能によりユーザーはシリアル番号やファイルからプログラムされたデバイスの ID 署名をを読むことが出来ます。

PIKPROG+には DIL から PLCC,SOIC,PSOP 変換ソケットがオプションで用意されております。

PC とプログラマーのスイッチをオフにして下さい。
PC のプリンター・ポートにプログラマーを付属のパラレル・ケーブルで接続してください。付属の電源ケーブルをプログラマーのラベル 12VDC と書いてあるコネクターと 100V 電源に接続して下さい。プログラマーの LED POWER が点灯し、PIKPROG+ が使用出来る状態であることを確認してください。そして、PC の電源をオンにして、コントロール・プログラムを実行してください。PIKPROG+をプリンター切り替え器等を経由して接続しないでください。



そして、PC の電源をオンにして、コントロール・プログラムを実行してください。

**警告!** *もし、PC をスッチ・オフしたくない場合は、下記の手順に従ってください。:*

- *プログラマーが PC に**接続されている場合**: パラレル・ケーブルを**最初**に、そして、**電源ケーブル***
- *プログラマーが**PC に接続されていない場合**: **最初** に **電源ケーブル、それから、パラレル・ケーブル**をはずしてください。*

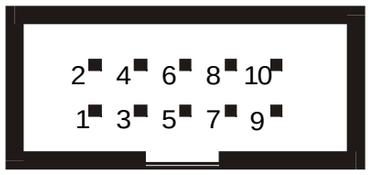
***PIKPROG+ ⇔ PC の接続とはずし方***

PIKPROG+ と PC の接続に問題がある場合は、***共通ノート***を見て下さい。

***ノートブック PC と PIKPROG+ユーザーのためのノート***

PIKPROG+ は 12V を供給していますので、この電圧を得ることが出来るどこでも車のなかでも使用することが出来ます。バッテリーから１時間以上電源を供給することが可能です。電源ケーブルの極性は下記になりますので、ユーザーの責任において安全にご使用ください。ELNEC 及び、有限会社 データダイナミクスはユーザーがプログラマーの使用により発生した如何なる事故や火災に伴う損害をへの責任は負いませんのでご承諾の上、ご使用下さい。:

### *PIKProg+ ISP コネクターの説明*




プログラマーの ISP コネクターの正面

**ISP コネクター・ピンの機能**

| ピン | 説明 |
|---|---|
| 1 | ターゲット・デバイスのための VCCP |
| 2,4, 6, 8,10 | H/L/read, GND, VPP |
| 3 | H/L/read, GND, VCCP, VPP |
| 5 | NC |
| 7,9 | GND |

ISP コネクターのピンのスペシフィケーションはプログラミング・デバイスに依存します。そして、**Device info window** (**Ctrl+F1**)に表示されます。関連するチップの ISP プログラミング方法が選択して下さい。プログラムされるチップ名の後の(ISP)サフィックスに示されております。
これらのスペシフィケーションはマイクロチップ・テクノロニー社のアプリケーション・ノートに対応しております。

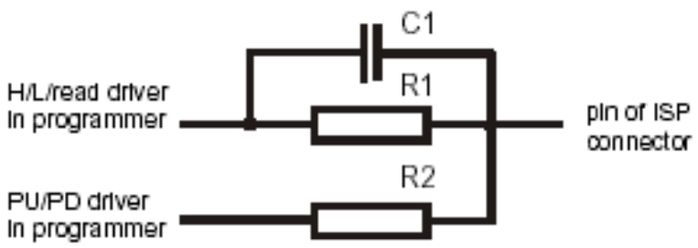
***ノート:*** *ISP ケーブル・コネクター上のピン番号 1 はトライアングル・スクラッチにより設計されております。*

PIKProg+ ISP ケーブル

**警告:**

- ***PIKprog+を ISP プログラマーとして使用する時は、 ZIF ソケットにデバイスを装着しないで下さい。***
- ***ZIF ソケットでデバイスをプログラミングする時は、ISP ケーブルを ISP コネクターに接続しないで下さい。***
- ***付属しています ISP ケーブルのみをご使用下さい。 他の ISP ケーブルを使用されますと、プログラミングが上手くいかない可能性があります。.***
- ***PIKprog+はプログラムされるデバイスにのみ電源を供給することが出来ます。しかし、ターゲット・システムへの電源を供給することが出来ません。***
- ***PIKprog+はターゲット・デバイスにプログラミング電圧を印加し、そして、その値をチェックします。(ターゲット・システムはプログラミング電圧を修正することが出来ます。)。もし、プログラミング電圧が期待したものと異なる場合は、ターゲット・デバイスに対してプログラミングは実行されません。***

*ノート: H/L/read PIKprog+ ドライバー*

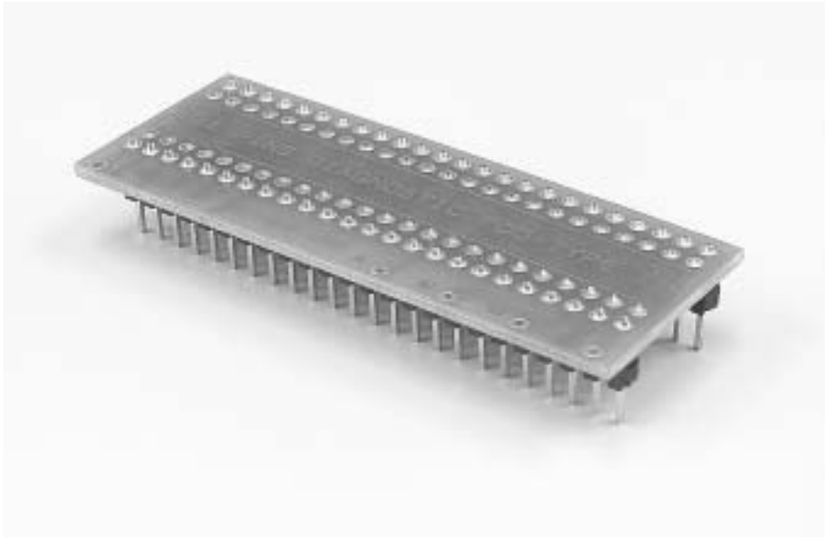



***ISP コネクター***

- 10-ピン・メス・タイプ *装着ミス・ロック
- 6 TTL ピンドライバーが H, L, CLK, プル-アップ, プル-ダウン, 1.8V～5V までのレベル H 選択可能な低電圧を含むすべてのデバイスをサポート
- 1x VCCP 電圧(範囲 2V..7V/100mA) と 1x VPP 電圧(範囲 2V..25V/50mA)
- ソース/シンク機能でのプログラム・チップ電圧(VCCP) と電圧感知

セルフ・テストとカリブレーション

もし、プログラマーが正常に動作していないと思える場合(但し、最低３ヶ月毎)は, 診断 POD を使って PIKPROG+ のセルフ・テストを行って下さい。TEST メニューを使って行って下さい。



### *ソケット, ピン・ドライバーと DAC*

- FPGA ベース IEEE 1284 スレーブ・プリンター・ポート 1MB/s までの転送レート
- 40 ピン DIL ZIF (Zero Insertion Force) が 40 ピンまでの 300/600 mil デバイスをアダプター無しでサポート
- VCCP と VPP のための２つの D/A コンバータで立ち上がりと下がり時間をコントロール
- VCCP 範囲 0..7V/250mA
- VPP 範囲 0..15V/150mA
- ピンドライバー: 40 TTL ピンドライバー, PICmicro® デバイスのための特別な GND/VCC/VPP ピンドライバー
- 将来のデバイスのために GND/VCC/VPP ピン機能のためのドライバーがスペアされております。
- FPGA ベースの TTL ドライバーがすべてのピン上(ピン・ドライバー)で H, L, CLK, プルアップ, プルダウンを供給
- 1.8V から 5V までのレベル H 選択
- イン-サーキット・プログラミング (ISP)機能を含んでいます。
- 連続テスト: 各ピンはプログラミング操作の前に毎回テストされます。
- セルフ・テスト機能
- 自動カリブレーション

### *デバイス・サポート*

- マイクロコントローラ MicrochipTM PICmicro®: 12xxx, 14xxx, 16xxx, 17xxx,18xxx, dsPIC300 シリーズ , 8 から 40 ピン (*1), パラレルとシリアル・モード
- シリアル E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx シリーズ

  *(*1) – DIL 以外のアダプターもご使用いただけます。*

### *プログラミング速度*

**ノート** これの時間は PC の速度、LPT ポートのタイプと OS のフリー・リソースに大きく依存します。従いまして、２つの違った値は違った PC で比較されています。

| Device | Operation | Time A | Time B |
|---|---|---|---|
| PIC16C67 | programming and verify | 13 sec | 11 sec |
| PIC18F452 | Programming and verify | 11 sec | 9 sec |

**Time A conditions:** *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

**Time B conditions:** *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

*デバイス操作*

- **標準:**
  - デバイスのタイプや IC メーカー名、及び、デバイス名の断片でのデバイス選択
  - ブランク・チェック
  - 読み込み[Read]
  - プログラム
  - ベリファイ
  - イレース
  - コンフィギュレーションとセキュリティ・ビットのプログラム
  - 不正ビット・テスト
- **セキュリティ:**
  - 装着テスト
  - 接触チェック
  - ID バイト・チェック
- **スペシャル**
  - 自動デバイス連続番号インクレメント

*バッファ操作*

- ビュー/編集, 検索/置き換え
- フィル/コピー, 移動, バイト・スワップ, word/dword 分割
- チェックサム(byte, word)
- 印刷

*サポート・ファイル形式(フォーマット)*

- アンフォーマット(Raw)バイナリー
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX

***General***

- 操作電圧 12-15VDC/最大 500mA
- 消費電力 5W 最大
- サイズ 132x66x40 [mm] (5.2x2.6x1.6 [インチ])
- 重さ 200g
- 操作温度 5° ÷ 40°C
- 湿度 20%..80%　非結露



### *パッケージの含まれているもの*

- PIKPROG+ プログラマー
- PC との接続ケーブル 1.8m (6ft.)
- 12-15V DC/ 最大 500mA 電源アダプター(非安定型)
- 診断 POD
- コントロール・プログラム
- ユーザー・マニュアル
- "PROGRAMMER PROBLEM REPORT"と"DEVICE PROBLEM REPORT"フォーム用紙
- レジストレーション・カード
- ペーバー・カートン・パッケージ

### *その他のサービス*

- ご希望のアルゴリズムを追加します。
- フリー・テクニカル・サポート *e-mail 又は、ＦＡＸ)
- ウェブサイトを経由してのフリー・ライフ・タイム・ソフトウエアのアップデート

# ソフトウエア



## *プログラマー・ソフトウエア*

プログラマーのパッケージにはコントロール・プログラム、ユティリティーの入っているフロッピー又は、CD が付いています。ご使用される場合は、オリジナル・ディスクのコピーの作成をお薦めします。プログラマーがどのように動作するかをデモするためにフリーにコピーすることが許可されています。このマニュアルと違って修正された内容が README_P.TXT ファイルに含まれているかも知れません。

## *プログラマー・ソフトウエアのインストール*

**Setup セットアップ**・プログラムを実行してください。

PG4UW.EXE (Windows 用)はすべての ELNEC のプログラマーに共通して使用できるコントロール・プログラムです。

## *ソフトウエアの新しいバージョン*

常にプログラマーの機能を最大にご利用頂くために、最新のバージョンを[www.datadynamics.co.jp](http://www.datadynamics.co.jp)のソフトウエアのダウンロードからダウンロードされることをお薦めします。

## *プログラマー・ソフトウエアの使用*

*コントロール・プログラムはフロッピー又は、CD で配布されています。ウィルスの心配の無い様に検査されて出荷されておりますので、それを保つ意味からもフロッピー・ティスクの場合はライト・プロテクトしておいてください。*

### *コントロール・プログラムの実行*

ウィンドウズの場合は、PG4UW アイコンをダブル・クリックして下さい。

コントロール・プログラム(PG4UW)は自動的にすべてのポートをスキャンし、そして、接続されているプログラマー(JetProg, BeeProg+, MEMprog, 51&AVRprog, PIKprog+等)をすべてサーチします。

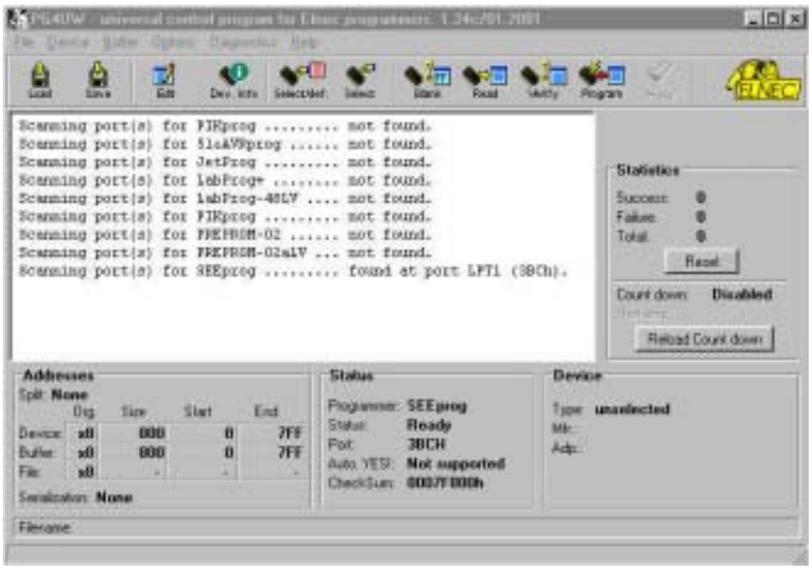
*ノート: PG4UW プログラムが開始されますと、標準のユーザー・メニューが表示されてユーザーからの指示を待ちます。*

もし、コントロール・プログラムがプログラマーと通信できないと、スクリーンにエラー・コードと考えられる理由（プログラマーの接続が外れている、間違った接続をしている、

電源アダプターの不良、間違ったプリンター・ポート）を含むエラー・メッセージが現れます。 再点検の上、問題を取り除いて、そして、いずれかのキーを押してください。
もし、エラーがまだ有る場合は、プログラムはデモ・モードで再開されますので、プログラマーへのアクセスは出来ません。もし、エラーの原因が見つからないときはトラブル・シューティングの指示に従ってください。コントロール・プログラムはデバイスのプログラムの前にプログラマー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




**Header** ヘッダー・バー
PG4U/PG4UW コントロール・プログラムの名前, copyright とバージョン
**Menu** メニュー・バー 基本機能のリスト
**File** ファイル・ウィンドウ/**ファイル名**
バッファにロードされているファイルの情報
**Status** ステータス・ウィンドウ
プログラマーと PG4U/PG4UW についての情報
**Addresses** アドレス・ウィンドウ
ターゲット・デバイス、バッファとファイルのサイズ、開始と終了アドレス
**Device** デバイス・ウィンドウ
ターゲット・デバイスについての関連情報
**Help** ヘルプ・バー
コマンドの簡単な説明



*ノート: Buffer/View/Edit[バッファ/ビュー/編集]コマンドのASCII ブロックを除き、キーボードで入力されたコマンド・データはHEX フォーマットです。*

## *ホット・キーのリスト*

| | | |
|---|---|---|
| **<F1>** | ヘルプ | ヘルプを呼ぶ |
| **<F2>** | セーブ | ファイルの保存 |
| **<F3>** | ロード | ファイルをバッファにロード |
| **<F4>** | エディット | バッファのビュー/編集 |
| **<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>** | 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 |
| **<Ctrl+F5>** | 選択/タイプ | デバイス・ファミリーによるターゲット・デバイスの選択 |
| **<Shift+F5>** | 選択/ベンダー | デバイス・マニファクチャラーによるターゲット・デバイスを選択 |
| **<F6>** | ブランク | ブランク・チェック |
| **<F7>** | リード | デバイスの内容をバッファに読み込み |
| **<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>** | プログラム | ターゲット・デバイスをプログラム |
| **<Alt+Q>** | 保存せずに終了 | プログラムを終了 |
| **<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>** | | 現在のデバイスの追加情報を表示 |
| **<Ctrl+F2>** | イレース | 与えられた値でバッファをフィル |
| **<Ctrl+Shift+F2>** | | ランダム値でバッファをフィル |

このサブメニューはソース・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。**option / Save option**・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。

"Load file[ファイルのロード]"ダイアログから"File format recognition[ファイル・フォーマットの識別]" グループ・ボックスで、ユーザーはロードされるべきファイル・フォーマットを自動又は手動で識別するのかを選択することが出来ます。

チェック・ボックスをチェックして下さい。**Automatic file format recognition** は自動的にプログラムがファイル形式を検出します。もし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。

**Automatic file format recognition** のチェックボックスにチェックが入っていない場合は、ユーザー側が **Selected file format** のパネル上から、手動によって利用出来るファイル形式のリストから適したフォーマットを選択します。バイナリーを選択した場合は、バッファーの開始アドレスを指定することが出来ます。バッファの開始アドレスはファイルから読まれたデータがバッファ・メモリに書き込まれる時のバッファ・メモリの開始アドレスです。

もし、チェック・ボックスに **Swap bytes** が表示されていますと、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・バイト機能を有効にすることが出来ます。この機能はビッグ・エンディアンのファイルをモトローラのバイト形式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。

*ノート: ビッグ-エンディアン、リトル-エンディアンとはコンピューター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。(最下位の書き込みアドレス)。リトル-エンディアンはリトル・エンド(least significant 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ーでは、ヘキサ・デシマス値 4F52 は２バイトが要求され、ワードの書き込みアドレス 1000H には 4F52H としてストアされます。4FH はバイト・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 (バイト・アドレス 1000H が 52H でバイト・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* |

**<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 / Save[ファイル保存]***

作成や修正されたバッファ・メモリのデータや、デバイスから読み込まれたデータの保存です。希望するフォーマット(**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** 及び **POF**)を選択することが出来ます。

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

***File / Load 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(セキュリティー)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]***

設定を保存せずにプログラムを終了

***File / Exit and save[ファイル/終了と保存]***

INI ファイルに設定を保存してプログラムを終了

*デバイス*

この機能は選択されたプログラマブル・デバイスの操作に使用します。- 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 10 の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 **<Del>** 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、**<Del>**キーは受け付けられません。

***Device / Select device ...[デバイス/選択されたデバイス....]***

このウィンドウは現在のプログラマーによりサポートされているすべ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。

**Selected device[選択されたデバイス]** は自動的にバッファにデフォルトのデバイス(最大１０デバイス)がセーブされます。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、**<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 ... / All[デバイス選択.../全部]***

このウィンドウは現在のプログラマー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、サーチ・ボックス(セパレート・キャラクターとして**<Space>** を使って、そして、**<Enter>** を押すか、又は、**OK** ボタンをクリックして下さい。)で入力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。

Selected device[選択されたデバイス] は自動的にバッファにデフォルトのデバイス(最大１０デバイス)がセーブされます。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、**<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ターゲット・タイプを選択することが出来ます。最初に、デバイス・タイプ(EPROM の様に)を選択しなければいけません。そして、マウス又は、カーソル・キーを使いながら、次にデバイスのサブ・タイプ(64Kx8(27512)の様に)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、サーチ・ボックス(セパレート・キャラクターとして**<Space>** を使って、そして、**<Enter>** を押すか、又は、**OK** ボタンをクリックして下さい。)で入力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。

Selected device[選択されたデバイス] は自動的にバッファにデフォルトのデバイス(最大１０デバイス)がセーブされます。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、**<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 manufacturer[デバイス選択 .../選択されたマニファクチャラーのみ]***

このウィンドウはマニファクチャラーによりターゲット・デバイスのタイプを選択することが出来ます。最初に、マウス又は、カーソル・キーを使いながら、マニファクチャラー・ボックスでターゲット・デバイスのマニファクチャラーを選択することが出来ます。そうしますと、選択されたマニファクチャラーのデバイス・リストが表示されます。

デバイスは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、サーチ・ボックス(セパレート・キャラクターとして**<Space>** を使って、そして、**<Enter>** を押すか、又は、**OK** ボタンをクリックして下さい。)で入力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。

Selected device[選択されたデバイス] は自動的にバッファにデフォルトのデバイス(最大１０デバイス)がセーブされます。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、**<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 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。プログラマー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の 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 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 IC 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:** *プログラマーはソケット上の対応ピンに高電圧に適応します。これはデバイス ID を読むために必要です。EPROM でないデバイスをソケットに装着しないで下さい。プログラマーが高電圧に対応しているときは、ダメージを与えることがあります。*

*このコマンドを 2764 と 27128 EPROM タイプに適応させることはお薦めできません、それらのほとんどで ID がサポートされておりません。*

### *Device / Device option[デバイス/デバイス・オプション]*

このメニューのすべ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 option / Operation option[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。これはターゲット・デバイスとプログラマー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。項目はターゲット・デバイスに対しては有効ですが、現在のプログラマーではサポートされておりませんので、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。

**通常使われる項目のリスト:**

**- 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
バッファ開始アドレス (デフォルト 0)

**-インサーション・テスト・グループ:**

Insertion test
装着テスト (デフォルト ENABLE)
Check ID byte
チェック ID バイト (デフォルト ENABLE)

**- コマンド実行グループ:**

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)
プログラミング後のベリファイ (ONCE, TWICE)
ベリファイ・オプション (nominal VCC 5%, nominal VCC 10%, VCCmin VCCmax)

***Device / Device option / Serialization[デバイス/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２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

もし、新しいデバイスが選択され、シリアライゼーション機能をデフォルト状態に設定されますとディスエーブルです。
実際に選択されたデバイスに対する実際のシリアライゼーション設定は **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。

インクレメンタル・モードがアクティブなとき、次の実際の設定がコンフィギュレーション・ファイルにセーブされます。: アドレス, サイズ, シリアル, 増加ステップとモード ASCII / BIN, DEC / HEX, LS byte / MS Byte first の設定。
フロム・ファイル[ファイルから]・モードがアクティブなとき、次の実際の設定がコンフィギュレーション・ファイルにセーブされます。: インプット・シリアライゼーション・ファイル名と入力ファイルで実際のシリアル番号で表示される行の実際のラベル。

### *Device / Device option / Serialization / Incremental mode[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]*

インクレメンタル・モードは各プログラム・デバイスに個々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下記の項目があります。

**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。
Bin(バイナリー) シリアライゼーション・モードの値は 1-4 が有効です。
ASCII シリアライゼーション・モードでは 1-8 の値が S/N サイズに対する有効値です。

**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（最高又は、最低）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。

**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 32 ビット・ロング・ワードで$1FFFFFFF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 つの最も大きなビットがゼロに設定されます。このあと、番号はつねに 0..$1FFFFFFF 間の中です。(これはオーバー・フロー操作の基本スタイルです。).

**ステップ**
ステップ・オプションはシリアル値の増加の増加ステップを指定します。

**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 ストリング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 – シリアル番号を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の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 進法]
- Hexadecimal[ヘキサデシマル - 16 進法]

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。
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。 すなわち、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[バッファにセーブ]**
Save to buffer[バッファにセーブ]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。) はシリアル番号のリースト・シグニフィカント・バイトを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。) はモースト・シグニフィカント・バイトをバッファの最下位アドレスに置きます。

***Device / Device・option / Serialization / 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・モード]***
From-file[フロム・ファイル]を使ってユーザーが指定した入力ファイルからシリアル値が読まれます。そして、入力ファイルで指定されたアドレスでバッファに書込まれます。

２つのユーザー・オプションがあります。: ファイル名と開始ラベル

**ファイル名**
ファイル名オプションはシリアル・アドレスと値が読まれるファイル名を指定します。フロム・ファイル・シリアライゼーションのための入力ファイルは、下記に説明されるフロム・ファイル・シリアライゼーション・ファイル・フォーマットのような特別なフォーマットでなければいけません。

**開始ラベル**
スタート・ラベルは入力ファイルでスタート・ラベルを定義します。ファイルから読まれたシリアル値が定義されたスタート・ラベルから開始します。

**フロム・ファイル・シリアライゼーション・ファイル・フォーマット[From file serialization file format]**
フロム・ファイル・シリアライゼーション入力ファイルはアドレスとバッファに書込むためのバッファ・アドレスとデータを定義するバイトの列を含んでいます。入力ファイ

ルはテキスト・タイプのフォーマットを持っています。その構成は:

```
[label1] addr byte0 byte1 .. byten
...
[labeln] addr byte0 byte1 .. bytem , addr byte0 byte1 ... bytek
\_____________________/  \_____________________________/
          |                          |
      基本部分                   オプション部分

; Comment
```

**基本部分**
基本部分はバッファに書込むためのバッファ・アドレスとバイトの列を定義します。基本部分はつねに行でラベルの後に定義しなければいけません。

**オプション部分**
オプション部分はバイトの２番目のアレイとバッファに書込むバッファ・アドレスを定義します。オプション部分はデータのベーシック部分の後に定義することが出来ます。

label1, labeln - labels
ラベルは入力ファイルの各行を識別するものです。それらはファイルの各行を呼ぶために使用されます。従って、ラベルはユニークでなければいけません。
ファイルのアドレス行の意味は、ユーザーにより入力された要求される開始ラベルがシリアル値が開始を読まれる入力ファイルでの行を定義します。

addr -
Addr はアドレスに従ってデータを書込むためにバッファ・アドレスを定義します。

byte0..byten, byte0..bytem, byte0..bytek -
バイト・アレイ byte0..byten, byte0..bytem と byte0..bytek はバッファに書込むために割り当てられるデータを定義します。アドレスに従う１つのデータ・フィールドのバイトの最大カウントは 64 バイトです。バッファに書込まれるデータ・バイトはアドレス addr から addr+n です。
特定のバイトをバッファに書きこむプロセスは:

```
byte0  ~  addr
byte1  ~  addr + 1
byte2  ~  addr + 2
 ....
byten  ~  addr + n
```

オプション部分はキャラクター“ , “ (カンマ)により最初のデータ部分から制限され、そして、その構成は最初のデータ部分と同じです。すなわち、アドレスとデータ・バイトのアレイ。

特別に使用するキャラクター:

[ ] – ラベルは角カッコ内に定義しなければいけません。

',' – はデータのベーシック部分とオプション部分を区別します。

‘;’ – セミコロンはコメントの始まりを意味します。„;„ から行の終わりまでのすべてのキャラクターは無視されます。コメントは個々の行又は、定義行の終わりに入れることが出来ます。

***ノート:*** *ラベル名は‘[‘ と ‘]’を除くすべてのキャラクターを含めることが出来ます。ラベル名は非ケース・センシティブとして扱われます。すなわち、キャラクター ‘a’ は ‘A’と同じ, ‘b’ は ‘B’と同じです。*

入力ファイルのすべてのアドレスとバイト番号値はヘキサデシマルです。
許容されるアドレス値のサイズは 1 から 4 バイトです。
許容される１行のデータ・アレイのサイズは 1 から 64 バイトの範囲です。１行に２つのデータ・アレイがあるとき、バイトでのサイズの合計は最大 80 バイトにすることが出来ます。
正しいアドレスを設定するように注意して下さい。アドレスはデバイスの開始と終了アドレス範囲内に定義されなければいけません。アドレスが範囲の外にある場合は、警告のウィンドウが現われ、そして、シリアライゼーションがディスエーブルに設定されます。

**例:**

```
[nav1] A7890 78 89 56 02 AB CD ; コメント 1
[nav2] A7890 02 02 04 06 08 0A
[nav3] A7890 08 09 0A 0B A0 C0 ; コメント 2
[nav4] A7890 68 87 50 02 0B 8D
[nav5] A7890 A8 88 59 02 AB 7D

;next line contains also second definition
[nav6] A7890 18 29 36 42 5B 6D , FFFF6  44 11 22 33 99 88 77 66 55 16
```

; this is last line - end of file

サンプルのファイルでは各ラベルに６つのシリアル値 „nav1“, „nav2“, ...“nav6“ が定義されています。各値はアドレス $A7890 上のバッファに書込まれます。すべての値がサイズ 6 バイトを持っています。„nav6“ のラベルを持つ行はアドレス $FFFF6 とサイズ 10 バイトのバッファに書込まれる２つ目の値の定義を持っています。すなわち、この値の最後のバイトはアドレス$FFFFF に書かれます。

***Device / Device・option / 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、すなわち、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[ベリファイ]**, **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。

チェック・ボックス- **Count down[カウント・ダウン]** はカウント・ダウンのイネーブル又は、ディスエーブルを設定します。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 パネルを右マウス・ボタンを押すことで、そして、項目はクリックすることでオープンされます。

実際のスタティクス値は Statistics[スタティクス]パネルのコントロール・プログラムのメイン・ウィンドウに表示されます。

スタティクス・パネルには３つのスタティクス値– **Success[サクセス]**, **Failure[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[リメイン]**を含みます。

値の意味は:

**Success** 成功して完了した操作の数
**Failure** 失敗した操作の数
**Total** 全ての操作の数
**Count down** カウント・ダウン(イネーブル又は、 ディスエーブル)についての情報
**Remains** 残っているデバイス操作の数についての情報

サクセスフル操作はエラーなしで完了した下記のデバイス操作を意味します。:

- プログラム
- ベリファイ
- ブランク・チェック
- イレース
- リード

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。
**スタティクス**・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。
**スタティクス**・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**に初期値を再ロードします。

***Device / Device option / Associated file[デバイス/デバイス・オプション/アソシエーティド・ファイル(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イネーブル又は、ディスエーブルも変更出来ます。

**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イネーブルをディスクにセーブ出来ます。

***Device / Blank check[デバイス/ブランク・チェック]***

このコマンドは、もし、可能な場合は、全てのデバイス又は、そのパーツのブランク・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]ウィンドウに警告のメッセージを書くことにより、このアクションの結果を報告します。

メニュー・コマンド **Device / Device option / Operation option[デバイス/デバイス・オプション/操作オプション]** で標準としてその他のワーキング・エリアを設定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てのデバイス又は、その一部分を読み込むことが出来ます。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 option / Operation option[デバイス/デバイス・オプション/操作オプション]** で標準としてその他のワーキング・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ベリファイ]** 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]***

このコマンドはバッファにあるデータと全てのデバイス又は、その一部分のプログラムされたデータを比較照合します。コントロール・プログラムは INFO[情報]ウィンドウにエラー・メッセージを書くことにより、このアクションの結果を報告します。

メニュー・コマンド **Device / Device option / Operation option[デバイス/デバイス・オプション/操作オプション]** で標準としてその他のワーキング・エリアを設定することが出来ます。

メニューの **option / Display errors[オプション/エラーの表示]** を設定することにより、このコマンドは見つかったエラーを表示したり、又は、VERIFY.ERR ファイルに書きます。Display errors[エラー表示]モードで、それらを起こしたアドレスの位置で、プログラムを最大 45 まで最初に見つかった違いを画面に表示します。

### *Device / Program[デバイス/プログラム]*

このコマンドはバッファにあるデータを全てのデバイス又は、その一部分にプログラムすることが出来ます。コントロール・プログラムは INFO[情報]ウィンドウにエラー・メッセージを書くことにより、このアクションの結果を報告します。

メニュー・コマンド **Device / Device option / Operation option[デバイス/デバイス・オプション/操作オプション]** で標準としてその他のワーキング・エリアを設定し、そして、その他の操作オプションをプログラミング・プロセス・コントロールのために設定することが出来ます。

### *Device / Erase[デバイス/イレース]*

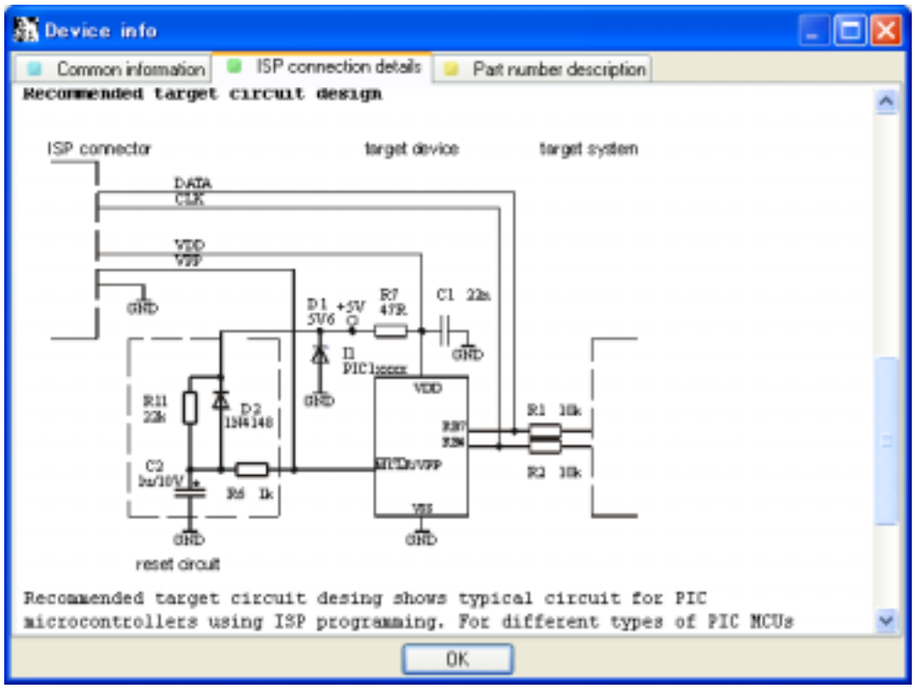
このコマンドはすべてのプログラマブル・デバイスを消去することが出来ます。プログラムはエラーなしで終了、又は、エラーで終了したかを警告のための報告を表示します。

### *Device / Test[デバイス/テスト]*

このコマンドはサポート・デバイスのリストから選択されたプログラマー(このテストをサポートしている)上のデバイス(すなわち、スタティクス RAM)のテストを実行します。

### *Device / Device info[デバイス/デバイス情報]*

コマンドはターゲット・デバイスについての追加情報を提供します。 - デバイスのサイズ, オーガナイゼーション, プログラミング・アルゴリズムと、そのデバイスをサポートしているプログラマーのリスト(モジュールを含む)。
ここでパッケージ情報とその他の一般情報も見つけることが出来ます。ISP プログラミングに関する情報も見ることが出来ます。




**<Ctrl+F1>** キーでいつでも、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。

このサブ・メニューはバッファ操作、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[バッファ/ビュー/エディット]***

このコマンドはバッファ(ビューは DUMP モードのみ)のデータを見たり(ビュー・モード) 又は、編集(エディット・モード) するのに使用します。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表示されます。

**<F4>** ホット・キーでも使用出来ます。

***View/Edit Buffer[ビュー/バッファの編集]***

| | |
|---|---|
| **F1** | ヘルプの表示 |
| **F2** | フィルのための開始と終了ブロックと要求されるヘキサ又は、ASCII ストリングをセットして下さい。 |
| **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 |
| **F3** | ブロックのコピー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・アドレスの外側である必要はありません。 |
| **F4** | ブロックの移動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・ブロック(又は、一部分)はブランク・キャラクターによりフィルされます。 |
| **F5** | スワップ・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スワップします。<br>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。<br>もし、この条件が満たされないと、プログラムの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、終了アドレスが高い奇数アドレスに移動されます。) |
| **F6** | プリント・バッファ |
| **F7** | ストリング検索(最大長 16 ASCII キャラクター) |
| **F8** | ストリングを検索し、置き換えます。(最大 16 ASCII キャラクター) |
| **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 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(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。 |
| **矢印キー** | カーソル移動 |
| **Home/End** | 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** | ターゲット・デバイスの開始/終了へジャンプ |
| **Backspace** | カーソルを１つ左へバック |

*ノート: キャラクター 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・デバイスへの ASCII キャラクターの編集は出来ません。

***Print buffer[プリンター・バッファ]***

このコマンドはバッファの選択された部分をプリンター又は、ファイルに書きます。プログラムは外部テキスト・エディターを使用します。デフォルトでは **Notepad.exe** に設定されています。

Print buffer[プリント・バッファ]ダイアログに次のオプションがあります。:

**Block start[ブロック開始]**
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[ブロック終了]**
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

**外部エディター**
バッファの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。ユーザーはどのテキスト・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 text[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ストリングスを入力し検索します。検索を始めるために**<OK>** 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(エディット・モード). **Forward** (現在の位置、又は、バッファの最初からバッファの最後へ) がデフォルトです。**Backward** 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。

***Replace text[テキストの置換]ダイアログ・ボックス***
**Text to find[**検索のためのテキスト**]**ストリング・入力ボックスに検索のためのストリングスを入力し、そして、**Replace with[置換]** 入力ボックスに置換のための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。

**Origin[オリジン]** は検索をどこから開始するか指定します。

**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(エディット・モード). **Forward** (現在の位置、又は、バッファの最初からバッファの最後へ) がデフォルトです。**Backward** 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>** を押すか、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** 置換と次を検索
**No** 置換せずに次を検索
**Replace All** すべてを置換
**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** アドレスへ...
**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。
**矢印 keys** カーソル移動
**Home/End** 現在行の開始/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** エディット・エリアの開始/終了へジャンプ
**Backspace** カーソルを１つ左へバック

*ノート: 0 と 1 のキャラクターがエディット・エリア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、要求したヘキサ(又は、ASCII)ストリングによりバッファの選択されたブロックをフィルします。フィルのためのブロックの開始と終了とヘキサ(又は、ASCII)ストリングを設定して下さい。

***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・アドレスの外側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・ブロック(又は、一部分)はブランク・キャラクターによりフィルされます。

***Buffer/ Swap block[バッファ]***

この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スワップします。
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、プログラムを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、終了アドレスが高い奇数アドレスに移動されます。)

***Buffer / Erase[バッファ]***

このコマンドを選択しますと、バッファの内容をブランクでフィルします。

**<Ctrl+F2>** キーでいつでも、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[バッファ]***

もし、このコマンドが選択されますと、バッファの内容がランダム・データでフィルされます。

**<Shift+Ctrl+F2>** キーでいつでも、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 Checksum[バッファ・チェックサム]***



チェックサム・ダイアログは選択されたバッファのブロックのチェックサムを計算するのに使用されます。チェックサムは次の方法で計算されます。:

**Byte** バイトによるワードの合計。CY フラッグは無視されます。
**Word** ワードによるワードの合計。 CY フラッグは無視されます。
**Byte (CY)** バイトによるワードの合計。CY フラッグは結果に追加されます。
**Word (CY)** ワードによるワードの合計。 CY フラッグは結果に追加されます。
**CRC-CCITT** RESULT[結果]=PREVIOUS[前回] + (x^16 + x^12 + x^5 +1) を使ったバイトによるワードの合計。
**CRC-XModem** RESULT[結果]=PREVIOUS[前回] + (x^16 + x^15 + x^2 +1) を使ったワードによるワードの合計。

**Neg.** とマークされたコラムはチェックサムの相殺です。
従って、Sum + Neg. = FFFFH.
**Suppl.** とマークされたコラムはチェックサムを補うものです。従って、Sum + Suppl. = 0 (+ carry).

チェックサム・ダイアログは下記の項目を含みます。:

**From address[アドレスから]:** これはバッファ内のチェックサムを計算するために選択されたブロックの開始アドレスです。アドレスはバイト・アドレスにより定義されます。

**To address[アドレスへ]:** これはバッファ内のチェックサムを計算するために選択されたブロックの終了アドレスです。アドレスはバイト・アドレスにより定義されます。

**Insert checksum[チェックサムの挿入]:** これは **Calculate & insert[計算と挿入]** が実行されたときバッファに書込まれるチェックサムの種類を選択するのに使用される特別な項目です。

**Insert address[アドレスの挿入]:** これは **Calculate & insert[計算と挿入]** が実行されたときに、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を指定するための特別な項目です。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。アドレスはバイト・アドレスとして定義されます。

**Size[サイズ]:** この項目は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を設定するために使用されます。

チェックサム結果のサイズは 8(バイト)又は、16(ワード)ビット長です。 もし、ワード・サイズが選ばれますと、全チェックサム値がバッファに書込まれます。それ以外は、チェックサム値のロー・バイトのみが書込まれます。
**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。*

**Calculate[計算]: 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を計算します。バッファへの書込みは行われません。

**Calculate & insert[計算と挿入]: 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。

*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。

***Option / General option[オプション/ゼネラル・オプション]***

ゼネラル・オプション・ダイアログはユーザーがプログラムの下記のオプションをコントロールすることが出来ます。

**File option[ファイル・オプション]**

ファイル・オプション・ページは認識されるファイル・フォーマットのロードを選択し、そして、ファイル・マスクを設定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]**

このコマンドは、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、シンタックスに正しく適応していなければいけません。

***Option / View[オプション/ビュー]***

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。

***Option / 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。

***Option / 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。

***Option / Display errors[オプション/表示エラー]***

このオプションはプログラムされたデータのベリファイの結果としてのエラー表示の形式をセットすることが出来ます。エラーをスクリーンに表示(最大 45)したり、カレント・ディレクトリーの **VERIFY.ERR** ファイルに保存したり、又は、表示をオフにすることが出来ます。エラー表示がターン・オフにされますと、コントロール・プログラム

は INFO ウィンドウのみに警告メッセージをリポートします。

この設定は **option / Save option[オプション/セーブ・オプション]** コマンドでディスクにセーブすることが出来ます。

***Option / Find programmer[オプション/プログラマー検索]***

新しいプログラマーのタイプとコミュニケーション・パラメータを選択します。このコマンドは下記の項目を含んでいます。:

**Programmer[プログラマー]** – 新しいプログラマーのタイプをセットします。もし、Search all[すべてを検索]が選択されますと、コントロール・プログラムはサポートされているすべてのプログラマーを検索します。
**Establish communication[通信の確立]** – 新しいプログラマーのための手動又は、自動通信の設定を行います。
**Speed[スピード]** – 通信速度を設定。もし、マニュアル通信設定が選択されますと、速度は最大速度からパーセンテージとして表示されます。

通信速度修正は PC<->プログラマーのケーブルに対して十分なパワーを持っていない、遅い LPT ポートを持った PC に対しては重要です。
もし、PC の LPT ポートとプログラマーの接続で通信の問題がある場合は、このコマンドを使用してください。コントロール・プログラムはプログラムがない、プログラマーとの通信が不安定等をリポートします。

Automatic establishing communication[自動通信設定]を選択しますと、コントロール・プログラムが最大通信速度を設定します。

**ポート** –LPT ポートを選択しますと、プログラマーが接続されているポートをスキャンします。もし、すべてのポートが選択されますと、コントロール・プログラムは標準アドレスとして利用出来るすべてのポートをスキャンします。

**特別なポートのためのアドレス** – もし、特別なポートが選ばれた場合、LPT ポートのアドレスを設定します。

設定パラメータによるプログラマーのスキャンを開始するときは**<Enter>**キー、又は、**OK** ボタンを押して下さい。

***Option / Automatic YES![オプション/オートマチック YES]***

このコマンドは **Automatic YES!** モードを設定するために使用されます。このモードでは、ユーザーはデバイスを ZIF ソケットに装着するだけで、キーやボタンを押すことなく最後に行った操作が自動的に繰り返されます。
ZIF ソケットへのデバイスの装着がスクリーンに表示されます。

***ノート:*** *新しいデバイスを ZIF ソケットに装着するのを待っている間、プログラマーの LED BUSY が点滅します。*

このモードは **Automatic YES!** モードによりイネーブル又は、ディスエーブルにすることが出来ます。もし、新しいプログラマーが **option / Find programmer[オプション/プログラマー検索]**を選択していますと、このモードはディスエーブルになります。

**Response time[応答時間]** は ZIF ソケットが新しいデバイスの装着を受け付けて、検知するタイム・インターバルをセットすることが出来ます。デフォルトは標準インターバルにセットされています。ソケット・アダプターが使用されている場合は少し長く設定して下さい。

**Pins with capacitors[コンデンサーを持ったピン]** バーでコンデンサー(例: もし、VCC と GND の間がコンデンサーで接続されているコンバーターが使用されている場合)による内部接続されたピンのリストを入力することが出来ます。この場合、新しいデバイスの装着を検知するときに問題があるかも知れません。
デバイスのピンのリストの形式:
pinA, pinB, pinC....
**例:** 4,6,17

このリストは、もし、新しいデバイスが **Device / Select default[デバイス/デフォルト選択]** 又は、**Device / Select device[デバイス/デバイス選択]** により選択されていますと、消去されます。

### *Option /Log file[オプション/ログ・ファイル]*

このオプションは **Log window[ログ・ウィンドウ]**の使用と関連しています。このウィンドウに関するすべてのリポートは **Log file[ログ・ファイル]** にも書込まれます。ログ・ファイル名は **REPORT.REP** として、コントロール・プログラムがカレント・ディレクトリーにこのファイルを作成します。

**New[新規]**にセットしますと、もし、古いログ・ファイルがある場合は、削除され、新しいファイルが作成されます。**Append[追加]** をセットしますと、古いログ・ファイルに全リポートを追加します。ファイルがない場合は、新しいファイルを作成します。設定はプログラムのスタート時のみ適応されます。

### *Option / Save option[オプション/セーブ・オプション]*

このコマンドは、オート・セーブがターン・オフになっていても、保存に対する現在サポートされている全ての設定をセーブします。

*診断*

このコマンドはプログラマーのためのセルフテストと IC テストを含んでいます。

***Diagnostics / Selftest[診断 / 自己テスト]***

このコマンドは診断 POD なしでプログラマーのセルフテストを実行します。プログラマーの **Diagnostics / Selftest plus[診断/セルフテスト・プラス]** もお薦めします。

***Diagnostics / Selftest plus[診断 / 自己テスト・プラス]***

このコマンドは診断 POD を使ってプログラマーのセルフテストを実行します。出来るだけ１ヶ月に一度位の頻度で行って頂く事をお薦めします。

***Diagnostics / IC test[診断 / IC テスト]***

このコマンドは IC のためのテスト・セクションをアクティベートします。最初に、正しいライブラリー、希望のデバイスとテスト・ベクターを実行(**Loop[ループ]**, **Single step[シングル・ステップ]**)させるためのモードを選んで下さい。コントロール・シーケンスとテスト結果が Log[ログ]ウィンドウに表示されます。

**一般情報:**
インターゲット・システムでデバイスと接続する場合は、ISP ケーブルがターゲット・システムと ISP プログラマーに正しく接続されていることを確認して下さい。そして、プログラマーの 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。
ISP プログラミングを使用されるに際して、マイクロチップ社のイン-サーキット・シリアル・プログラミング(ICSP) ガイド (http://www.microchip.comで "ICSP guide"で検索して下さい。) は必ずお読み下さい。

**ISP コネクター・ピンの機能**

| ピン | 説明 |
|---|---|
| 1 | デバイスへの VDD 電源供給 |
| 2 | PGM – 低電圧プログラミング・ピン |
| 3 | NC |
| 4 | NC |
| 5 | NC *1 |
| 6 | シリアル・データ I/O (RB7) |
| 7 | GND (VSS) |
| 8 | シリアル・データ・クロック(RB6) |
| 9 | GND (VSS) |
| 10 | デバイスへの MCLR/VPP |

*1 BeeProg プログラマーはターゲット・システムに電源供給することが出来ます。 ヘルプの"Device options / Operation options" メニューを見て下さい。

イン-サーキット・プログラミングを行う時は、ISP コネクターの VDD 電源ピンが PIC マイコンのすべての VDD と AVDD ピンに接続されていること、及び、ISP コネクターの GND が PIC マイコンのすべての VSS と AVSS ピンに接続されていることを確認して下さい。

**マイクロチップ PIC18Fxxx** デバイスはすべて 0 の状態で出荷されている場合が有ります。この状態のデバイスではデバイス操作(書き込み、読み出し、ベリファイ、ID チェック)で問題が起きることがありますので"Device operation options"ダイアログでプログラミング前に自動的にイレースするように設定するようにして下さい。ID チェック・エラーが起こった場合は、ID チェック機能を OFF にして下さい。最初のイレース操作の後は、デバイスはノーマル(ブランク)状態にセットされ、そして、デバイス操作が可能になります。

**低電圧プログラミング:**
ISP プログラマーはデバイスの低電圧プログラミング操作機能を使ってプログラミングすることが出来ます。

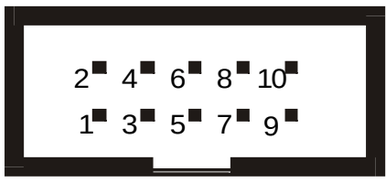
低電圧プログラミング・オプションは"Device operation options"ダイアログでセット又は、ディスエーブルすることが出来ます。低電圧プログラミングを使用するにはデバイス・コンフィギュレーション・ワードが'1' (enabled)で LVP にセットされていなければいけません。

そうでなければ、デバイスは低電圧モードでプログラム操作することが出来ません。低電圧プログラム・ピン(PGM ピン)は ISP プログラマーの ISP コネクターの PGM ピンに接続されている必要が有ります。これは他のピン(RB6,RB7,VDD,VSS,MCLR/VPP)に加えて必要です。PGM ピンは標準のプログラミング電圧モードでも PGM ピンは 'L'にセットされていなければいけませんので、低電圧プログラミングが使用されない時でも、通常のプログラミングモードのときこのピンがＬにする必要があるため、ISP の PGM ピンに正しく接続されている必要が有ります。

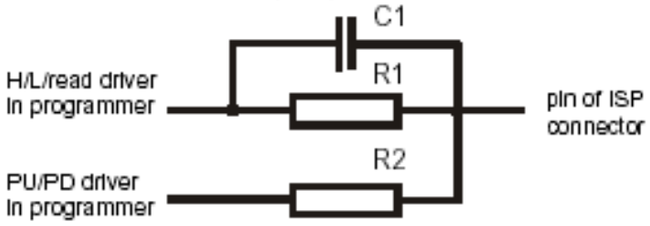
**ISP ノート:** プログラマーが ISP モードで動作していることはデバイス操作が ISP コネクターのみを経由して行われていることを意味します。プログラマーの ZIF ソケットは空でなくてはいけませんので、デバイスが ZIF ソケットに忘れられてる場合は、ダメージを受ける恐れがありますので注意して下さい。

***PIKprog+ ISP コネクター***

プログラマーの ISP コネクターの正面




*ノート: H/L/read PIKprog+ ドライバー*




C1 1nF, R1 1k3, R2 22k

上記の値はプログラムされるチップとターゲット・システムをアイソレートする抵抗の値を示したものです。

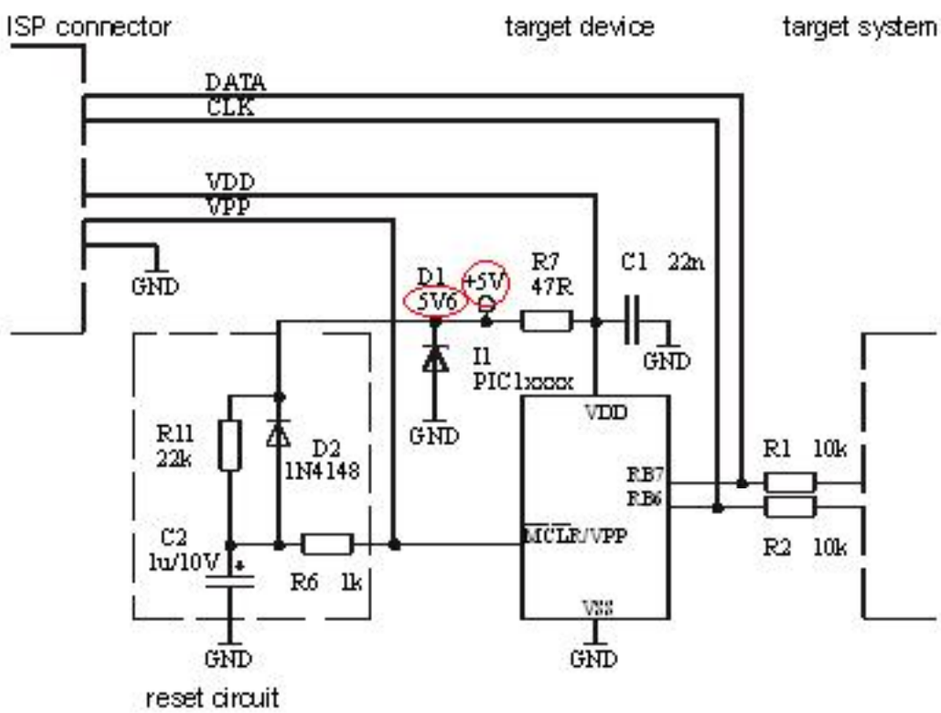
推奨ターゲット回路設計：
次の信号がPICmicro® マイコンのイン-システム・プログラミングに使用されます。

| | |
|---|---|
| MCLR\ / VPP | プログラミング・モードのリセット/スイッチ |
| RB6 (GP1) | クロック |

| | |
|---|---|
| RB7 (GP0) | データ入力/出力 |
| VDD | 電源供給 |
| GND | グラウンド |

PICmicro® デバイスがプログラムされる時、MCLR\ / VPPピンは約12Vで駆動されます。従って、ターゲット・システムはプログラマーから供給されるこの電圧からアイソレートされていなければいけません。RB6とRB7信号はイン-システム・プログラミングのためにPICmicro® によって使用されます。従って、ターゲット・システムはプログラミング・エラーを避けるためにイン-システム・プログラミング中はこれらの信号に影響を与えてはいけません。
プログラミングの後、マージナル・ベリファイが使用されます。プログラマーは公称値と最大値の両方でプログラム・メモリーの内容をベリファイしなければいけません。従って、PICmicro® のVDDピンはプログラミング中はターゲット・システムの他の部分からアイソレートされていなければいけません。

ELNEC社のPICmicro®推奨回路:




上記の回路設計はISPプログラミングを使った典型的なPICマイコンの回路を示しています。PICマイコンの種類によって多少の違いが有ります。例えば、低電圧プログラミング機能を持ったデバイスではPGMピンと呼ばれる１つのエクストラ・ピンを使用します。PIC17C7xxファミリーは２つのエクストラ・ピン- TESTピンとT0CKIピンを使用します。

ノート: ターゲット・デバイスとターゲット・システムをセパレートするR1, R2 (R3)抵抗はスイッチ又は、ジャンパーで置き換えること



も出来ます。ターゲット・デバイスの ISP プログラミングの間はスイッチ(ジャンパー)はオープンでなければいけません。

*ノート: VDDパワーアップ・スロープが遅すぎる場合のみ外部リセット回路が必要です。*