

# PicBasic Pro Compiler

*microEngineering Labs, Inc.*

**DEBUGMON.TXT** (デバッグモニター) の

完全日本語訳 (テック・ハンゾウガネ訳)

## Copyrights and Trademarks

Copyright ©2000 microEngineering Labs, Inc.

All rights reserved.

EPIC and PicBasic Pro are trademarks of microEngineering Labs, Inc.

BASIC Stamp is a trademark of Parallax, Inc.

PICmicro is a registered trademark of Microchip Technology Inc.

## Japanese Translation Copyrights

©2003 Tech - Hanzougane Yoshiaki Morohashi

日本語訳文の著作権はテック・ハンゾウガネ諸橋義明に帰属します。

尚、日本語訳文から派生する利用者のいかなる不利益も

テック・ハンゾウガネ諸橋義明は責任を負いません。

## ***PicBasic Pro デバッグモニター (DEBUGMON.TXT の翻訳)***

PicBasic Pro バージョン 2.2 と以降はデバッグモニタプログラムを作るために使われるフックを含んでいます。モニターはユーザプログラムや停止されたり、リスタートされたり、データを表示されたりすることをユーザプログラムに可能にすることで存在するプログラムです。

PicBasic Pro の新しい ON DEBUG 命令はモニタプログラムが BASIC で記述されることを可能にします。モニターはそれぞれの基本命令間に管理が与えられます。DISABLE DEBUG と ENABLE DEBUG 前置はプログラムがどこで流れを変えられることができるかをコントロールするために使われることができます。

### **マクロ変更**

このデバッグモニターは PicBasic Pro コンパイラ (バージョン 2.20) で使われることができます

、変更がメインマクロファイル、PBPPIC14.MAC に変更されなければなりません。

若干のバージョンがすでにこの変更を含んでいます。

最初にファイルのコピーを保存してください。それからファイル PBPPIC14.MAC を DOS EDIT あるいはウインドウズワードパッドのような各種テキストエディタで編集してください。

それはこのように見えるべきである DCALL ? L マクロを搜してください:

DCALL?L macro Dlabel

local label

movlw low label

movwf DEBUG\_ADDRESS

movlw high label

movwf DEBUG\_ADDRESS + 1

L?GOTO Dlabel

label

endm

endmod

見せられ始めて RST?RP 行を加えなさい。:

DCALL?L macro Dlabel

local label

RST?RP

movlw low label

movwf DEBUG\_ADDRESS

```

        movlw    high label
        movwf    DEBUG_ADDRESS + 1
        L?GOTO   Dlabel
label
    endm
endmod

```

ファイルとあなたが使う準備ができていないデバッグモニターを保存してください。

## デバッグモニター

モニタプログラムはホストから極めて少数しかリソースを必要としません。 1つか2つの変数が定義される必要がある: 定義されるべき バンク 0 でシステムワード変数として定義される必要のある `DEBUG_ADDRESS`:

```
DEBUG_ADDRESS    var    word bank0 system
```

さらに、第二の変数を定義することによって、スタックレベルはモニターされることができます:

```
DEBUG_STACK      var    byte bank0 system
```

PicBasic Pro 文を使うことによって、モニタプログラムは使用可能です:

```
On Debug Goto label
```

どこかの `label` はデバッグモニタールーチンの名前です。 一度 PicBasic Pro コンパイラがこの命令を見つけると、それは各 PICBasic 文の前のラベルにデバッグモニタールーチンヘジャンプを置くであろう、`On Interrupt Goto` と全く同じ方法で各文の前にそのコールを置きます。

```
Disable Debug
```

はコードのクリティカル部分でモニターヘジャンプを配置するのにコンパイラを止めるために使える。

```
Enable Debug
```

はコードをもう一度再開するところにコンパイラを開始するのにジャンプを配置することを引き起こす。モニターコードそれ自身は その周りにジャンプによって BASIC プログラムの始まりに置かれるべきです。

あるいはモニタプログラムはそうであるモニターされるどんなプログラムの始めにでも含まれる別個のファイルであることができます。

## 仕組み

各命令の前に間接的に、現在プログラムのアドレスは変数 `DEBUG_ADDRESS` に格納されて、実行されたモニターへジャンプする。モニターはそのときディスプレイバッキング情報を選択できる。

これらの技術はプログラムで最もよく実証される。

PicBasic Pro プログラム `DEBUGMON.BAS` はデバッグモニターが書かれた一つの方法を示す。

それはデバッグされるプログラムの始めに含まれるべきです。もう一つのサンプルプログラム、`DBLINK.BAS` がこれを示します。`DBLINK.BAS` はデバッグモニターが挿入された `BLINK` プログラムです。

`DEBUGMON` は Hyperterm のような、PC 上で走っているターミナル通信プログラムと通信します。それは `Hserin` と `Hserout` 命令を使います、それはハードウェアシリアルポートで `PICmicro` を必要とします。デフォルトのボーレートは 2400 です、そして TX と RX ピンは `PORTC.6` と `PORTC.7` です。

プログラム例は 16F877 のためにコンパイルされました、しかし 16C74 あるいは 16C77 で使われることができます。:

```
PBP -p16f877 -c -ol dblink
```

`DBLINK` は始めに `DEBUGMON` をインクルードして、そして次に 1 秒 1 度 `PORTD.0` に接続した LED を ON・OFF にするべき単純なコードでその後に続きます。

## DEBUGMON 操作

`DEBUGMON` はいくつかの 1 つ文字コマンドを受け入れます。コマンドのための準備ができているとき、それは端末画面の上に「？」を示します。「？」が示されると、小文字がコマンドを実行するために端末キーボード上でタイプできます。`DEBUGMON` での現在のコマンドは次の通りです:

- `d` - ディスプレイスタックレベル - とすべては 16 進数で 0 のレジスタを重ねます。
- `g` - それが中止(実行)された所でプログラム実行を続けてください。
- `h` - 短いヘルプスクリーンを見せてください。
- `m` - モニタプログラムを入力してください。
- `r` - リセットアドレス(0)において動作しているプログラムをスタートしなさい。
- `s` - 1 つの PicBasic Pro 文を通して 1 ステップ
- `t` - 、それが中止したところから、それが実行されるとき、16 進数でそれぞれの BASIC 文のアドレスを示すことによって、プログラムを追跡してください。

DEBUGMON のオペレーションを理解する最も容易な方法は DBLINK プログラムでそれを試してみることです。しかしながら、「m」コマンドは少し追加の説明を必要とすることができます。

プログラムが実行あるいはトレースを使って動作 している(、そして LED が明滅しています)であるとき、あなたがレジスタを調べるか、あるいはアドレスをトレースすることができるように、コントロールをモニターに戻す方法を持つことは必要です。ターミナルから「m」を入力することはプログラムの実行を止めて、そしてモニターにコントロールを返します。モニターが BASIC で記述されるとき、それはブレークポイントあるいはハードウェアシリアルポートが不足している PICmicros 上で使われることを可能にする Hser コマンドの除去のような、他の関数を提供するために修飾することができます。