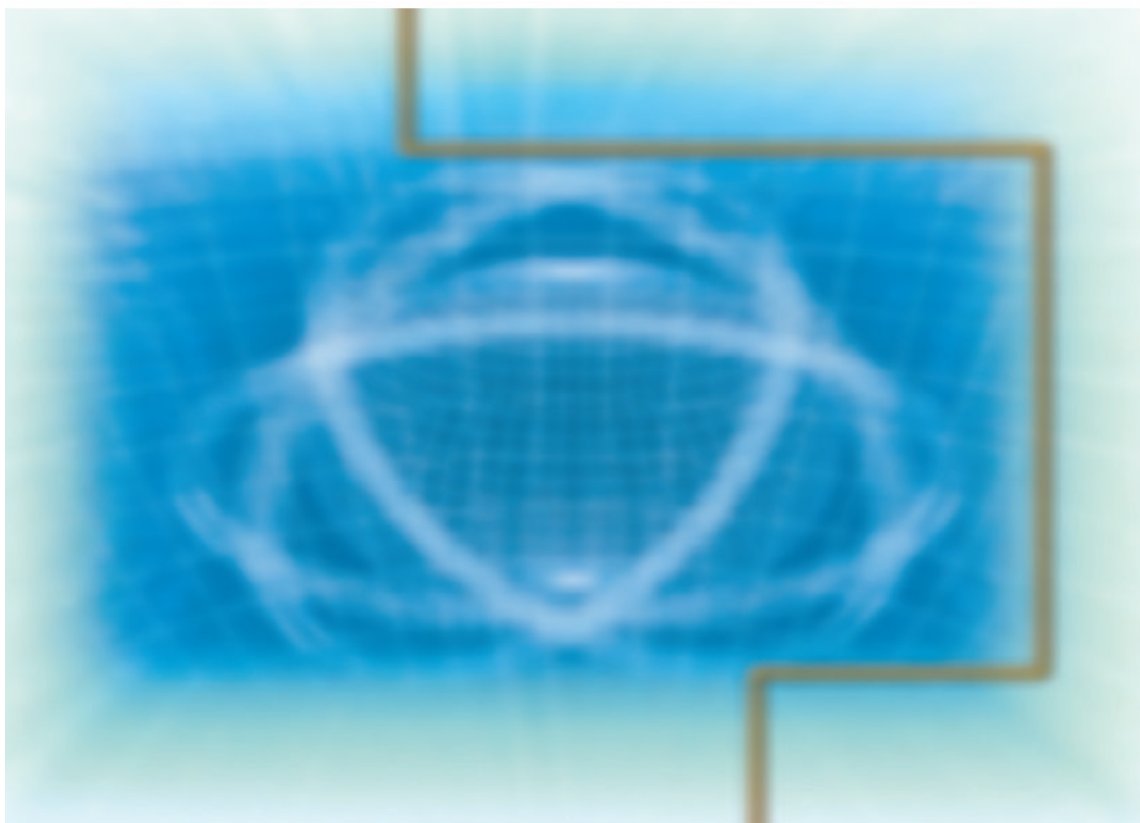


Transition Guide for the dsPIC®/PIC24 Compiler  
dsPIC®/PIC24コンパイラのための変更点ガイド



CCS PCD Compiler

revised: 25-JUL-07

# DATA

PCDコンパイラはデフォルトではintはint16、shortは8ビット、longは32ビットとなります。

更にすべてのintはデフォルトでは符号付となります。PCDで動作する変数タイプは今迄のCCSコンパイラで使用されているのと同じです。

## #TYPE CLASSIC

PCDコンパイラには整数タイプのint48とint64が追加されました。さらに2つの新しいfloat48とfloat64(aka double)も追加されました。

PCDコンパイラではマイクロチップ・フォーマットの代わりにIEEEフォーマットを使うために float32(aka float)フォーマットに変更されました。これはC30コンパイラでマイクロチップが使っているのと同じフォーマットです。

プログラム全体をとおして1つの実数タイプのみを使用することで、コード・サイズとランタイムが最も効率的に利用出来ます。

24ビット・オペコードのPICデバイスはint16を使用したときに最適に動作します。int8を使用した場合は、殆どの場合にオーバーヘッドが発生します。

他のCCSコンパイラで int8の値を返していた多くの組み込み関数は現在はint16を返します。

24ビット・オペコードのPICデバイスは他のPICのデバイスのようにRAMバンクを持っていません。RAMの最初の8Kは殆どの命令により同等にアクセスすることが出来ます。残りのRAMへのアクセスは少し効率が悪い方法でアクセスすることになります。

DSPではワークRAMは2つの領域に分割されます。(XとY)

#BANKXと#BANKYディレクティブは特定の領域に変数をおくために使用されます。DSP命令は特定の領域に確実なオペランドを要求します。

# ASSEMBLY

アセンブリ命令は他のPICの命令と同じではありません。コンパイラのインライン・アセンブリ機能はマイクロチップ・フォーマットのアセンブリを受け入れます。

PCWでは”Interpret”を選択するのにOPTIONS > PROJECT OPTIONS > FILE FORMATSを使用します。

そして、アセンブリ命令のインタープリテーションはLSTファイルに追加されます。これはアセンブリに

精通していない人には便利です。これはインプライド・レジスターを使用する命令に対して特に便利です。

同じプログラムでLSTファイルにあるSFR名を示すためにSFRオプションを使用することが出来ます。

## BUILT-IN FUNCTIONS

ペリフェラルを扱う多くの組込み関数がPCDコンパイラでは変更されました。しかし、名前は同じで、そして、使用されている定数も同様です。

PCDのヘルプ・ファイルはコンパイラ・リボンで24ビットコンパイラが選択されたときにPCWからアクセスすることが出来ます。

## PROGRAM MEMORY

各命令は24ビット長です。メモリーは16bit長(ワード)でアドレスされます。これは各命令に対して2つのアドレスが使用されることを意味していますが、これら2つのアドレスには3バイトのみがあります。

read/write \_program\_memory関数はデータをコピーしますが、4バイト目はブランクとなります。

ロム・データはこのように定義されます。:

```
rom char table[] = {"helow world"};
```

コンパイラは定数データをメモリにパックし、そして、実行されているプログラムに対してアドレスが解決されている時、偽のバイト・アドレスが使用されます。コンパイラはこのアドレスを組込み関数に移します。

CONST が使用されるとき、高速読み出しのためデータは最も効率的な方法でセーブされますが、ポインターはこのデータで使用(インデックスのみ)することは出来ません。