



第5章 付録マイコン基板の ソフトウェアを作成する

C言語による dsPICのプログラミング

山口 晶大
Akio Yamaguchi

本章では、付録マイコン基板のC言語プログラミングについて、簡単に解説します。C言語の世界で有名な“Hello, World!”に相当する最初のプログラムとして、LED点灯プログラムを作成します。

dsPICのCプログラミング

dsPIC30/33, PIC24用のマイクロチップ社純正のCコンパイラC30はANSI規格(1989年のC89)に準拠していて、皆さんが使い慣れたパソコン用のCコンパイラとほぼ同様に扱うことができます。ここでは、dsPIC向けの仕様の拡張点やパソコン用のCコンパイラとの仕様の細かい差異など、開発に当たって注意を要する点について簡単に説明します。

● 参照すべきマニュアル類

Cコンパイラの使用に当たって不明な点がありましたら、必ずマニュアルを参照してください。主なドキュメントとして以下の三つがあります。いずれもマイクロチップ社のウェブ・サイトからダウンロード可能です。C30のユーザーズ・ガイドには日本語訳の『MPLAB C30 Cコンパイラ ユーザーズガイド』もあります。

- (1) MPLAB C30 User's Guide(DS51284)
- (2) 16-Bit Language Tools Getting Started(DS70094)
- (3) 16-Bit Language Tools Libraries(DS51456)

内蔵ペリフェラルを使用したプログラムを作成するには、次のハードウェアのマニュアルも参照する必要があります。

表1 CコンパイラC30のメモリ・モデル

プログラム・メモリとデータ・メモリのメモリ・モデルを別々に設定可能。デフォルト設定は両方ともsmallモデル

メモリ・モデル	small(スモール)	large(ラージ)
プログラム・メモリ	32 Kワード以下	32 Kワード以上
データ・メモリ	8 Kバイト以下	8 Kバイト以上

- (4) dsPIC30F Family Reference Manual(DS70046)
 - (5) dsPIC30F2011/2012/3012/3013 Data Sheet(DS70139)
- アセンブリ・ルーチンも併用する場合は、アセンブリ言語プログラミングに関する資料も必読です。
- (6) MPLAB ASM30, MPLAB LINK30 and Utilities User's Guide(DS51317)
 - (7) dsPIC30F/33F Programmer's Reference Manual(DS70157)

*

付録CD-ROMにもドキュメントが収録されています。

● パソコン用のCコンパイラとの相違点

パソコン用のCコンパイラとdsPICのCコンパイラには、ハードウェアの構成の違いによる相違点がいくつかあります。本記事で紹介している程度の簡単なサンプル・プログラム、製作例などでは意識する必要のない点もありますが、重要なことですので覚えておいてください。

▶ メモリ構成

dsPIC内蔵メモリにはフラッシュ・メモリ(プログラム・メモリ)、SRAM(データ・メモリ)、データEEPROMの3種類があります。

付録マイコン基板搭載のdsPIC30F2012はデータEEPROMを内蔵していません。プログラム・メモリとデータ・メモリは完全に分離していて、データ・メモリ上にプログラムを配置して実行することはできません。

▶ メモリ・モデル

dsPICのCコンパイラにはメモリ・モデルがあります。プログラム・メモリ、データ・メモリそれぞれについて、表1のようにメモリ・モデルを設定できます。

付録マイコン基板搭載のdsPIC30F2012はメモリ容量が少ないので、コンパイラのデフォルト設定のsmallメモリ・モデルですべてのメモリ空間を使用可能です。

浮動小数点演算 ▶ 浮動小数点は、非常に大きな値や非常に小さな値を表現するのに向いているため、精度が要求される科学分野などで多く用いられる。浮動小数点の表現形式はIEEEにより規格化されている。

表2 dsPIC用Cコンパイラとパソコン用Cコンパイラのビット幅の相違

データ型		ビット幅[ビット]	
		dsPIC(コンパイラ C30)	パソコン(32ビット・コンパイラ)
整数	char	8	8
	short	16	16
	int	16	16 または 32
	long	32	32
	long long	64	64
浮動小数点	float	32	32
	double	32	64
	long double	64	80

▶ 演算語長と変数のビット幅

dsPIC用のCコンパイラC30とパソコン用の32ビット・コンパイラとでは、同じC言語のデータ型でもビット幅が異なります(表2)。パソコンで開発したプログラムをdsPICに移植する場合などには注意が必要です。

▶ 浮動小数点演算関数の使用

dsPICでも浮動小数点演算ライブラリを用いて、パソコン用Cコンパイラと同様に浮動小数点演算を使用可能です。組み込み開発でも三角関数や対数関数ができるのは大変便利です。

ただし、dsPICは浮動小数点演算器を搭載していないので、**固定小数点演算と比較すると浮動小数点演算の処理効率が大幅に劣る**ことに注意してください。浮動小数点演算の使用は必要最低限にとどめてください。

▶ dsPIC内蔵レジスタのアクセス

dsPICには多数の制御用レジスタが内蔵されています。付録マイコン基板で実行するC言語プログラムを作成する場合、ヘッダ・ファイルp30f2012.hをインクルードすることにより、定義済みの変数としてdsPIC内部の制御レジスタの読み書きが可能です。

ユーザが宣言して使用する変数名は、p30f2012.hでレジスタに割り当てられている変数と重複しないようにする必要があります。詳しくはリンク・スクリプト・ファイルp30f2012.gldも参照してください。

▶ 割り込み処理関数の記述

組み込みプロセス向けのCコンパイラでは、割り込みがかかったときに実行される割り込み処理関数(割り込みハンドラ)を記述できるようになっています。

dsPIC用のCコンパイラC30では、下記のような修飾子を付けて割り込み処理関数を宣言/定義します。詳しくはコンパイラのマニュアルを参照してください。

```
void __attribute__
(( (__interrupt__ )) )_T2Interrupt(void)
```

▶ 変数・定数の記憶領域指定

dsPICはいわゆるハーバード・アーキテクチャ構成で、プログラム・メモリ(フラッシュ・メモリ)とデータ・メモリ(SRAM)の二つのメモリ空間を有してい

ます。

下記のようにして、ユーザが変数や定数の記憶領域を明示的に指定することが可能です。詳しくはCコンパイラC30のマニュアルを参照してください。

```
const __attribute__
(( (space(prog)) )) x=3;
int __attribute__
(( (space(ymemory)) )) y;
```

簡単な制御プログラムの実験

● 組み込み開発の“Hello, World!”

組み込み開発の世界で“Hello, World!”に相当するのは「LEDピカピカ」です。組み込みプロセスの開発の第一歩として、発光ダイオード(LED; Light Emitting Diode)を光らせるプログラムを作ってみましょう。

そのまえに一つだけ、LEDの電気的特性について説明します。

● LED(発光ダイオード)の特性

図1のようにLEDに直接電圧を加えたときの電圧-電流特性の一例を図2(a)に示します。LEDのカタログには一定の電流が流れるときの電圧値が順方向電圧降下特性 V_F として記載されています。図2(a)の場合、測定電流20mAでの V_F が2.2Vとなります。通常、LEDを使用する際には、図2(b)に示すような定電圧特性をもっているとみなして扱って問題ありません。

赤色LEDの V_F は2.0~2.5V程度で、オレンジ色や緑色のLEDの V_F は赤色より高めます。青色LEDや

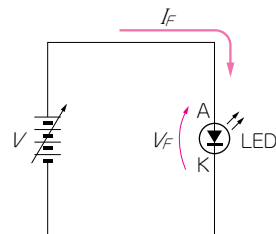


図1 LEDの順方向特性測定方法
印加電圧Vを変えたときに流れる電流 I_F を測定する

固定小数点演算▶ 固定小数点は実数の近似値の表現方式の一つ。固定小数点は、整数部分に用いるビット数と小数部分に用いるビット数をあらかじめ固定して表現する。