

STEP 2 送信機TX2-006Pのプログラミング

赤外線2チャンネル送信機TX2-006P TypeMは無事完成しましたか？

TX2-006Pには、ジョイスティック・PIC12F683のA-Dコンバータなどのプログラミングを楽しめる機能が満載です。インドアプレーンを操縦する前にプログラミングに挑戦しましょう。

準備運動として、LEDを3回点滅させるプログラムを作成しましょう。よく使われるPIC16F84AとPIC12F683は同じ「PIC」の仲間ですが、12F683は後発のために小型ながら最新機能を搭載しています。TX-006PではPICの内部クロックを使うことにより、外部発振子であるセラロックなどを省略し、回路を簡略化しています。これらの切り替えを含めたプログラミングにする必要があります。

それでは、TX2-006Pに装備しているモニタ用LED(PIC横の赤LED、赤外線LED近くの赤LEDはPICで制御できない)を使って、PIC12F683のプログラミングを楽しみましょう。

始業日

月曜日

火曜日

水曜日

木曜日

金曜日

土曜日

日曜日

放課後クラブ

1時限目・LEDを250msの間隔で3回点滅させるプログラムを作ろう



最初に行うイニシャライズ

メイン・プログラムに先立って、PIC12F683の設定をする部分をイニシャライズと呼びます。

```
#include<12f683.h>
```

このinclude文によって、12F683をコンパイルするのに必要な外部にあるヘッダ・ファイルを取り込みます。PICの型名そのものがインクルードするヘッダ・ファイル名になっています。include文はC言語では標準的に使われます。

```
#fuses INTRC_IO, NOWDT, PUT, NOPROTECT, NOMCLR
```

#fusesにより12F683の動作モードを設定します。fuses文はCCS-Cに特有な書き方です。ANSI-Cでは組み込みの記述は考慮されていませんので、マイコンの開発言語では、その言語メーカー特有の文が使われます。PIC用のほかのC言語では、このfusesは異なる文が使われることがあります。

```
INTRC_IO .....内部クロックを使う  
NOWDT .....ウオッチドック・タイマは使わない  
PUT .....パワーアップ・タイマを使う  
NOPROTECT .....書き込んだプログラムにプロテクトをかけない=書き込み、読み出しは自由にできる  
NOMCLR .....リセット端子を使わない=リセット端子をポートとして使う
```

```
#use delay(CLOCK=8000000) // 内部クロックを8MHzで駆動  
setup_oscillator(OSC_8MHZ); // 内部クロックに8MHzを使用する
```

外付けセラロックを省略する場合、内部クロックの使用宣言をここで行います。内部クロックは8MHzです。setup_oscillatorとセットで使います。この記述方法もCCS-C特有のもので、ここで、Cコンパイラに対してクロックを知らせることによって、ウェイト(ディレイ)などの時間を正確に計算してくれます。

```
#byte GP=5 // GPは0~5の6ポート  
#bit IROUT=GP.5 // 赤外線LED  
#bit LED=GP.4 // モニタ用のLED接続  
#bit SW=GP.3 // ジョイスティック付属のスイッチ
```

12F683は、GP₀~GP₅までの六つのポートを使用することができます。ポートのアドレス5番地をGPとし、各ビットをIROUT、LED、SWと定義します。