

このSTEPでは、プログラムの基本構成や各部分の意味合いなどについて説明します。

次のリストは、スイッチでLEDをON/OFFするだけの単純なプログラムです(このプログラムはSTEP18の回路で動作する)。このプログラムは、プログラムを構成するための基本的な要素のみで記述されています。このプログラムの全体を見ながら、各記述部分について説明します。

L-3

```

;-----
; LEDテスト プログラム (1)
;-----
                LIST           P=PIC16F819,R=DEC ..... ②
                INCLUDE        P16F819.INC ..... ③

    _CONFIG    _BODEN_OFF & _WDT_OFF & _INTRC_IO & _PWRTE_ON &
    _LVP_OFF   & _MCLR_OFF

SW      EQU    0 } ⑤
LED     EQU    7

ORG     0 ..... ⑥

        BSF     STATUS,RP0
        MOVLW  B'01111111'
        MOVWF  TRISB
        MOVLW  B'01111111'
        MOVWF  OPTION_REG
        MOVLW  B'01100000'
        MOVWF  OSCCON
        BCF     STATUS,RP0

MAIN

        BTFSS  PORTB,SW } ⑦
        GOTO   LED_ON
        GOTO   LED_OFF

LED_ON

        BSF     PORTB,LED
        GOTO   MAIN

LED_OFF

        BCF     PORTB,LED
        GOTO   MAIN
        END ..... ⑧

```

この2行は、スペースの都合で分かれているが、1行で記述する

① タイトル

セミコロン(;)の後に続く記述は、プログラムに反映されません。したがって、タイトルやコメントなどをセミコロンの後に記述します。

② プロセッサの指定

最初の行にある下記の記述は、このプログラムで扱うプロセッサ(PICの型式)指定と数値の形式指定です。

```
LIST    P=PIC16F819, R=DEC
```

形式を指定しない数値は10進数とする

このプログラムのターゲットはPIC16F819

③ インクルード・ファイルの指定

次は、インクルード・ファイルの組み込みを指定する記述です。インクルード・ファイルは PIC のデバイスごとに存在し、各 PIC の特殊機能レジスタの名称やビット名が定義してあります。このファイルを組み込むことにより、たとえば、アドレス H'05' に配置されたポート A レジスタをプログラム中で H'05' と記述せずに PORTA と記述することが可能になります。インクルード・ファイルは実際には MPLAB IDE をインストールしたときに生成された「Microchip」というフォルダ内の「MPASM Suite」というフォルダ内にあります。一度開いて目を通しておくとよいでしょう。

```
INCLUDE P16F819.INC
```

このファイルをこのプログラムに組み込む。
P16F819.INC は PIC16F819 の特殊機能
レジスタの名称やビット名が定義してある

④ コンフィグ(コンフィギュレーション)の記述

この記述で、PIC の基本的な動作条件を設定します。__CONFIG に続けて PIC のいろいろな動作条件の設定を記述していきます。

```
__CONFIG _BODEN_OFF & _WDT_OFF & _INTRC_IO & _PWRTE_ON & _LVP_OFF & _MCLR_OFF
```

アンダバー 2 文字分

たとえば、この記述は外部からのマスタ・クリア
機能を OFF にすることを意味する

設定の種類や記述は、PIC マイコンの種類によって異なります。詳しくは各データ・シートを参照してください。また、記述法については、前述したインクルード・ファイルの中にかかれています。

⑤ ラベル定義

この記述で任意のラベル(SWやLEDなどの文字表記)にビット、アドレス、定数などの数値を定義します。このように定義すると、プログラム中の記述を数値ではなくラベルで記述することができます。

```
SW EQU 0
```

この例ではビット 0 を、プログラム中で SW と記述できるようになります。

⑥ プログラム先頭の記述

ORG は、プログラム・メモリのアドレスを示す、疑似命令です。ビルドを行うと、ORG を記述した次の行の命令は、ORG で指定したアドレスに配置されます。よって、通常プログラムの先頭では ORG でアドレス 0 指定をしておきます。

```
ORG 0 ← 電源 ON 時の開始アドレス
```

⑦ プログラム本体

この部分の記述が、プログラムの本体です。これについては、後の STEP15 以降で詳しく説明します。

⑧ プログラムの最後の記述

プログラムの最後には END を記述します。ビルドを行うとアセンブラ(MPASM)は END の記述をプログラムの最後と判断し、この記述までのプログラムが HEX ファイルに反映されます。

```
END ← プログラムの終了を宣言
```