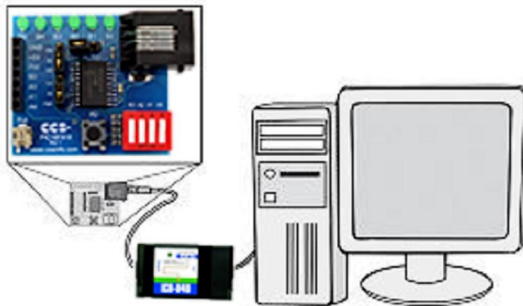


Quick C 開発キット

包装を開けてインストール

- ✓ 必要な PC と対応 OS Windows XP, Vista, Windows 7, 8, 10
PC は USB ポート、CD-ROM ドライブ、ハードディスク空容量 50MB 以上
 - ✓ **Quick C Development Kit** に含まれているもの:
 - ・CCS C コンパイラ(オプション) *PIC16F818 チップ対応のウィンドウズ版
 - ・インサーキットデバッガ/プログラマー・ユニット ・PIC16F818 開発ボード
 - ・ICD⇔PC 接続用 USB ケーブル ・ICD⇔開発ボード接続用モジュラー・ケーブル
 - ・日本語版 C 言語で何が起るか[C What Happens by David Benson]
-
- ✓ **CCS** コンパイラを同時ご購入の場合 ; *既に IDE 版コンパイラをお持ちの場合は不要
CD をコンピュータのドライブに挿入しますとプログラムのインストールが始まります。
もし、コンピュータがオートラン CD にセットされてない場合は、マイコンピュータから
CD ドライブをダブル・クリックして下さい
 - ✓ **Install** をクリック、要求に従って NEXT, OK, CONTINUE をクリックしてデフォルト
でインストールして下さい。
-
- ✓ **重要** : ICD を PC に接続する前に **USB** ドライバーをインストールしておいて下さい。
USB ケーブルを使って PC に ICD-U(*この冊子では以後 ICD とします。)を接続しま
す。モジュラー・ケーブルを使ってプロトタイプ・ボードを ICD に接続します。最初に
ICD が PC に接続されるとウィンドウズが新しいハードウェアを検知。
CD、又は、WEB サイトから USB ドライバーをダウンロードの上、インストールし ICD
の認識を完了して下さい。
如何なる場合も、デバイスが使用される前にドライバーが正しくインストールされて
いる必要があります。



- ✓ ICD の LED が PC に正しく接続されたことを示して点滅
- ✓ 次のプログラムを実行: スタート>プログラム>PIC-C>CCSLOAD。
ターゲットを PIC16F818 にセット。
- ✓ **Test Target** を選択。テストがパスしますとハードウェアは正しくインストールされて
いることとなります。

コンパイルとプログラムの実行





- ✓ アイコンをクリックして PCW IDE を起動。既にファイルが開かれている場合は **File>Close All** をクリック。 **File** アイコンは左上の隅
- ✓ **File>New>Source File** をクリック、そして、ファイル名を **blink1.C** と入れて下さい。
- ✓ コンパイラを選択するために **Compile** タブの下のドロップ・ダウン・ボックスから **PIC16F818 - PCM のための 14bit** が選択されていることを確認して下さい。
- ✓ 次のプログラムをタイプし、そして、**Compile** をクリック

```
#include <16f818.h>
#device ICD=TRUE
#use delay (internal=4mhz)

#define RIGHT_LED PIN_B0

void main ()
{ while (TRUE)
  {
    output_low (RIGHT_LED);
    delay_ms (1000);
    output_high (RIGHT_LED);
    delay_ms (1000);
  }
}
```

- ✓ モジュラー・ケーブルを使って ICD をプロトタイプ・ボードに接続
USB ケーブルを使って ICD を PC に接続
これでボードに電源供給され ICD の LED が点滅することを確認して下さい。
- ✓ **Debug>Enable Debugger** をクリックし、そしてプログラムがロードされるのを待ちます。
- ✓ デバッガが ICD と通信出来ていない時は、**Debug Configure** タブのリスト・ボックスで **ICD-USB** が選択されているか確認して下さい。ジャンパーがピン B0 と LED に接続されていることを確認して下さい。
- ✓  アイコンをクリック:
- ✓ デバッガ・ウィンドウ・ステータス・ブロックが黄色に変わってプログラムが実行されていることを示します。
- ✓ プロトタイピング・ボードの右の LED がフラッシュします。1 秒オン、そして、1 秒オフ
- ✓  アイコンをクリックしてプログラムを止めるか、又は、RESET します。:

アドバンスド・デバッグ

- ✓ RS-232 printf 表記はプログラムのデバッグを助ける良いツールとなります。しかし、それを使うには特別なハードウェアのセットアップが必要です。もし、ICD がデバッガとして使用されている場合は、コンパイラはデバッグ画面に `putc()` と `getc()` をデバッガ・インタフェースを通してデバッグ画面に導くことができます。
- ✓ PCW IDE を開きます、もしファイルが開かれている場合、**File>Close All** をクリック
- ✓ **File>New>Source File** をクリックして、ファイル名 **debug1.C** と入れて下さい。
- ✓ 次のプログラムをタイプして、**Compile** をクリック



```
#include <16F818.h>
#include <delay.h>
#include <RS232.h>

void main(void)
{
    char key;

    delay_ms(72);

    output_b(0);

    printf("Running\r\n");
    while(TRUE) {
        printf("LED to toggle (0-5): ");
        key=getc();
        printf("%c\r\n",key);
        switch(key) {
            case '0' : output_toggle(PIN_B0); break;
            case '1' : output_toggle(PIN_B1); break;
            case '2' : output_toggle(PIN_B2); break;
            case '3' : output_toggle(PIN_B3); break;
            case '4' : output_toggle(PIN_B4); break;
            case '5' : output_toggle(PIN_B5); break;
        }
    }
}
```

- ✓ **Debug>Enable Debugger** をクリックし、プログラムがロードされるのを待ちます。
- ✓  をクリックして、**Monitor** タブをクリック
- ✓ プロンプトが現れます。プログラムが動作しているかを確認するために何かデータを入れて下さい。
- ✓  アイコンをクリックしてプログラムを止めるか、又は、RESET します。

Hints

コンパイラ

- ✓ コンパイルされたメイン・プログラムは常に IDE の下に表示されます。もし、そのファイルがコンパイルしたいファイルで無い場合は、コンパイルしたいファイルのタブをクリックして下さい。右クリックでエディターに入り、そして Make file project を選択します。

IDE ビューワ

- ✓ Compiles>Symbol Map をクリック。このファイルは使用されているマイクロコントローラで RAM がどれだけ消費されているかを示します。@ で始まる識別子はコンパイラが生成した変数です。
注意: ある位置では1つ以上が使用されていますが、これはそれらの変数が同時にアクティブでないためです。
- ✓ Compile>C/ASM List をクリック。このファイルはオリジナルの C コードと C から生成されたアセンブリ・コードを示します。
- ✓ マイクロチップ社のデータシートをよく見ておいて下さい。
- ✓ PCW では project>Open all files をクリックすることで、全てのプロジェクト・ファイルを見ることが出来ます。

ヘルプ

- ✓ コードの関数の上にカーソルを置いて F1 を押しますとヘルプが現われます。
- ✓ エラー・メッセージの上で右クリックしますとエラーの情報が現われます。

PCW は IDE であり、PCW, PCWH, PCWHD の共通の総称です。

ICD ユニット

- ✓ ICD-U ユニットは内部にジャンパーがインストールされており、ターゲット・ボードに 5V を供給します。他の 5V が外部から供給されているボードでご使用になる場合はジャンパは外して下さい。



<http://www.datadynamics.co.jp/ccscc/CWorkshop.html>

