

PSoC1(CY8C27443)搭載 Arduino ピン互換ボード

PSoC1duino (ver 4.0)

(C) 2016 Junichi Akita

(@akita11 / akita@akita11.jp)

重要説明事項(必ずご一読ください)

本キットは個人が PSoC1 を使った電子工作を楽しむことを目的としています。本キットで提供しているハードウェア、ソフトウェアの全部もしくは一部を製品に組み込んだり販売したりすることはご遠慮ください。本キットを使ったことにより直接的、間接的に被害、損害を被ったとしても一切補償しません。

組み立てに失敗した場合や使用中に故障した場合でも交換、保証、返金などには一切応じられません。

本キットは汎用的に使えるハードウェアの提供を目的としています。このキットで作ることができると例示されたものは本キットの可能性を示したものです。例示されたものすべてを実際に製作したわけではありません。

充分注意して設計をしていますが、重大な設計ミスやバグがないことの保証はありません。回路、使用部品、基板、ソフトウェアなどは予告なく変更することがあります。

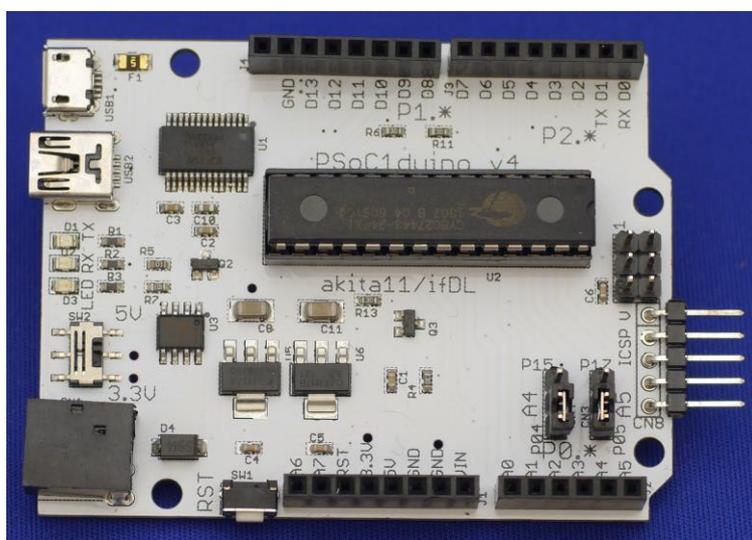
このマニュアルで説明しているボードやアプリケーションソフトウェアはこのマニュアル執筆時点のバージョンを使っていますので、ハードウェア・ソフトウェアの変更により、画面デザイン、操作方法、仕様が変わることがあります。

情報・ソフトウェア入手先

最新情報は下記サイトにて入手可能です。

- PSoC1duino 情報

<http://ifdl.jp/PSoC1duino/>



用意するもの

- PSoC1duino ボード
- USB ケーブル(A-miniB または A-microB)
- PC (Windows XP 以降)※仮想マシンでも動作します

1.準備

以下の手順で必要なファイルをダウンロードします。これらは以下のサポートページからリンクを張ってありますのでご利用ください。

サポートページ：<http://ifdl.jp/PSoC1duino/>

USB ドライバの準備とインストール

サポートページに貼ってあるリンクから、USB ドライバをダウンロードします (Windows 用。「ver 2.12.16」(2016/6/13 時点の最新版))。ダウンロード後、インストールしておきます。その後、PSoC1duino を PC に USB ケーブルで接続し、ドライバのインストールを求められた場合は、さきほど解凍したフォルダを指定します。念のため、デバイスマネージャ(コンピューター→プロパティから:Win7 の場合)を開き、COM ポートが割り当てられているのを確認します。

アプリケーションのインストール

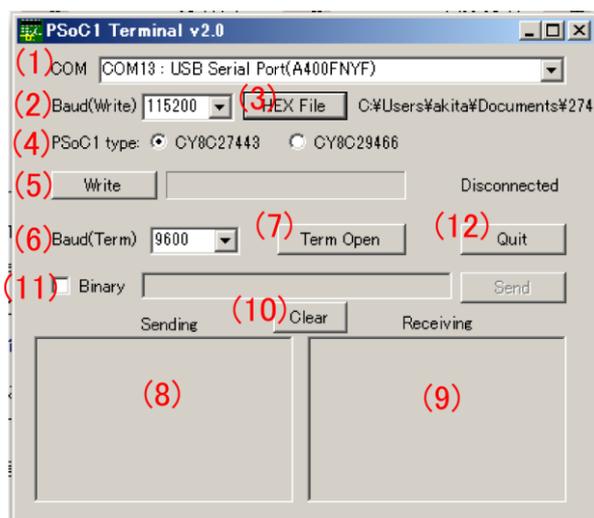
サポートページに貼ってあるリンクから Cypress 社の Web サイトにアクセスし、「PSoC Designer」をダウンロードし、インストールしておきます (2016/6/13 時点では ver5.4 が最新版)。なおダウンロードの際にユーザ登録 (無料) が必要です。

続いて、上記サポートページから「PSoC1term」(PSoC1term_v220.msi) をダウンロードして実行し、PSoC1duino 用書込・ターミナルアプリケーション PSoC1term をインストールしておきます。

同様に上記サポートページから「プログラム作成用のテンプレート」(PSoC1duino_Template.zip) をダウンロード後、作業フォルダにコピーし、展開しておきます。

2.PSoC1term の機能と操作

PSoC1term は、PSoC1term でよく使う「プログラム書き込み」と「デバッグ用シリアル端末」を統合したアプリケーションです。PSoC1term を起動すると、以下のようなウィンドウが現れます。主な操作と機能は以下の通りです。



- (1) COM: PC に接続されている COM ポートの一覧。PSoC1duino の COM ポートを選択しておく。
- (2) Baud(Write): プログラム書き込み時の通信速度(通常は変更しない)
- (3) HEX File: 書き込むプログラムの HEX ファイルを指定する
- (4) PSoC1 type: PSoC1duino に載っている PSoC1 の種類を選択(通常は変更しない)
- (5) Write: 指定したプログラムを書き込む※シリアル端末が開かれているときは自動的に閉じる
- (6) Baud(Term): シリアル端末の通信速度(標準では 9600bps のままでよい)
- (7) TermOpen: シリアル端末を開く
- (8) Sending: PC から PSoC1duino に送る文字を入力する
- (9) Receiving: PSoC1duino から PC へ送られてきた文字が表示される
- (10) Clear: Sending と Receiving の内容をクリアする
- (11) Binary: このチェックボックスをチェックした状態で、右側のテキストボックスに 16 進数表記文字列入力してから右側の Send ボタンを押すと、バイナリデータを送信する。(例:"120a"→0x12 と 0x0a の 2 バイトが送信される)
- (12) Quit: PSoC1term を終了する

3.動作テスト

出荷状態の PSoC1duino には、次のような機能のテスト用プログラムが書き込まれていますので、これを使って動作確認をします。

- 起動時に LED(オレンジ)が点灯し、"S"の 1 文字を送信(9600bps)する
- PC の COM ポートから 1 バイトのデータを受信(9600bps)し、「その値+1」を送り返す。(例えば"a"を送ると"b"が返ってくる)
- 1 バイト受信するごとに LED の点灯・消灯が切り替わる

※このテスト用プログラムのプロジェクトファイル一式は、サポートページからダウンロードできます。

1. PC に PSoC1duino を接続し、PSoC1term を起動すると「COM」のところに先ほどデバイスマネージャで確認した COM ポートが現れているのを確認する(複数の COM ポートがある場合は PSoC1duino の COM ポートを選択する)。



2. "Term Open"ボタンを押すと、PSoC1 がリセットされ(DTR リセット)、"Receiving"に"S"が表示される



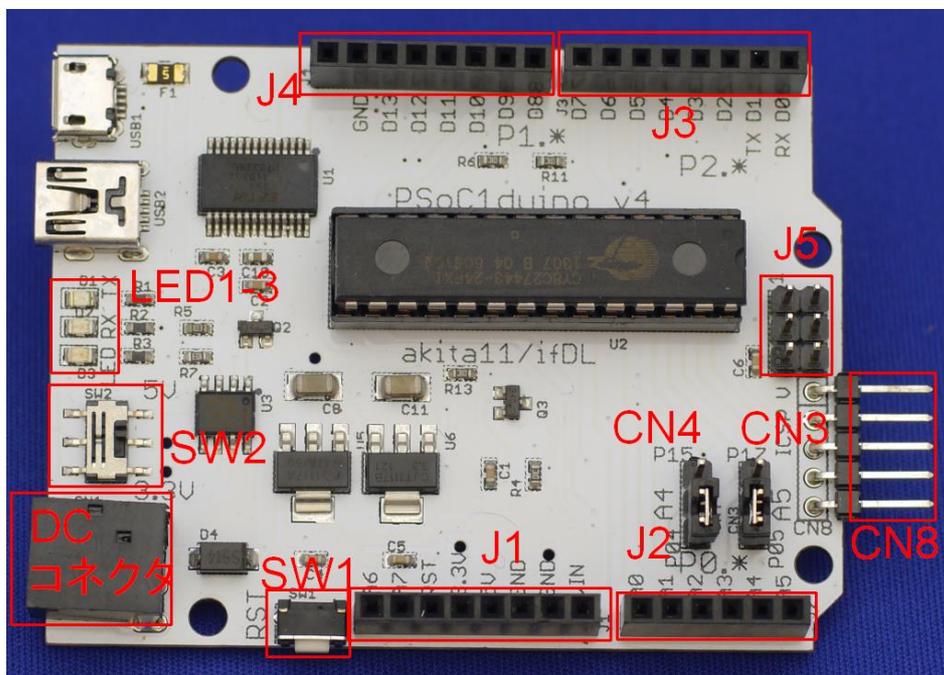
3. Sending に適当な文字を入力すると、その次の文字(例: "a"→"b")が表示される。1 文字入力するごとにオレンジ色の LED の点灯・消灯が切り替わる



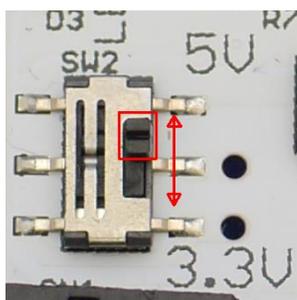
4. TermClose ボタンを押すとシリアル端末画面がグレーになって閉じられる。再度 TermOpen ボタンを押すと、PSoC1 がリセットされプログラムの実行が始まり、2.に戻る。

4. PSoC1duino ボードのピン配置・設定

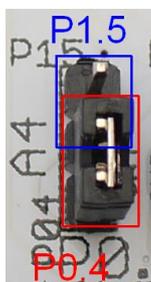
PSoC1duino には以下のようなコネクタ・設定用ジャンパピンがあります。



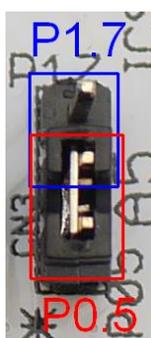
- USB コネクタ(CN8)
miniB と microB の 2 つのコネクタが並んでいます。どちらか一方を使って、PC と接続します。
- DC コネクタ: 電源供給が不足する場合はここに+7~12V (センター+) の AC アダプタ(内径 2.1mm)を接続できます。AC アダプタを接続しない場合は、USB コネクタを通して PC から電源が供給されます。
- I/O コネクタ(D0-D7)(J3): PSoC1 のピンが接続されています(表 1)
- I/O コネクタ(D8-D11)(J4): PSoC1 のピンが接続されています(表 1)
- I/O コネクタ(A0-A5)(J2): PSoC1 のピンが接続されています(表 1)
- I/O コネクタ(電源・A6-A7)(J1): PSoC1 のピン等が接続されています(表 1)
- 電源電圧設定(SW2): PSoC1 の電源電圧を 5V と 3.3V から選択できます



- A4ピン機能切り替え(CN4): I/O コネクタの A4 ピンを、PSoC1 の P0.4 または P1.5 のどちらに接続するかを選択する。(P1.5 は I2C の SDA ピンとして使用可)



- A5ピン機能切り替え(CN3): I/O コネクタの A5 ピンを、PSoC1 の P0.5 または P1.7 のどちらに接続するかを選択する。(P1.7 は I2C の SCL ピンとして使用可)

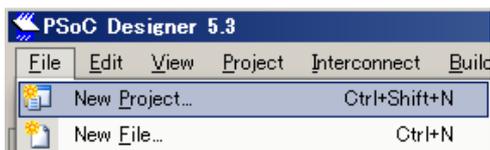


- ISP 端子(CN8): プログラム書き込みのために PSoC1 用の書き込み機である MiniProg を接続する(通常は使用しない)。基板上の「V」マークが、MiniProg の「VDD」側になるように接続する。
- Arduino ICSP 互換コネクタ(J5): Arduino Uno などにあるプログラム書き込み用端子と同一の位置に、表 2 のような信号線が引き出してある。一部の Arduino シールドにはこの端子を SPI 通信のために使うものがあるため、必要に応じて使う。
- リセットスイッチ(SW1): PSoC1 のリセットをかける。
- インジケータ LED(LED1~3): PC からの送信中(TXD:赤)、PC が受信(RXD:緑)の各状態と、D11(P1.6)に接続された LED(オレンジ)。

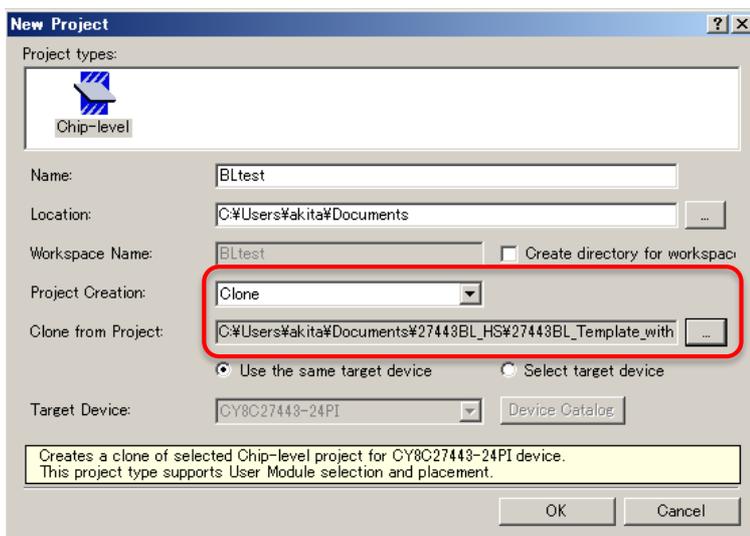
5. プログラムの開発から実行までの手順

PSoC1duino へのプログラムを USB(COM ポート)経由で書き込むためには、専用のブートローダが必要です。PSoC1 のブートローダは、Cypress 社がアプリケーションノート AN2100 として実装例を公開していますが、PSoC1duino ではこれを少し改造して転送速度を速めたブートローダをあらかじめ書き込んであります。自分でプログラムを作るときには、以下のようにブートローダを含むひな形(テンプレート)を使います。

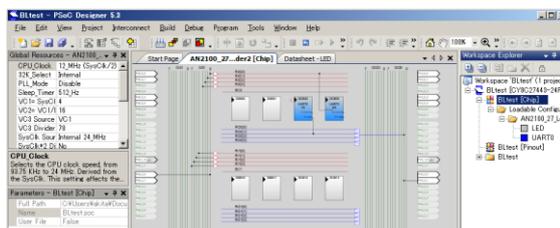
1. PSoC Designer で新規プロジェクトを作成する。



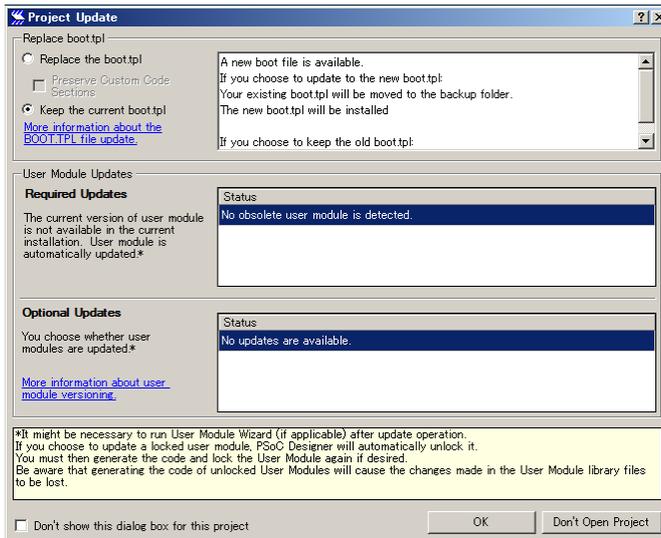
2. プロジェクト名などを入力後、プロジェクトの作成方法を"Clone"とし、「Clone from Project」で以下のいずれかのプログラム作成用テンプレートを選ぶ。



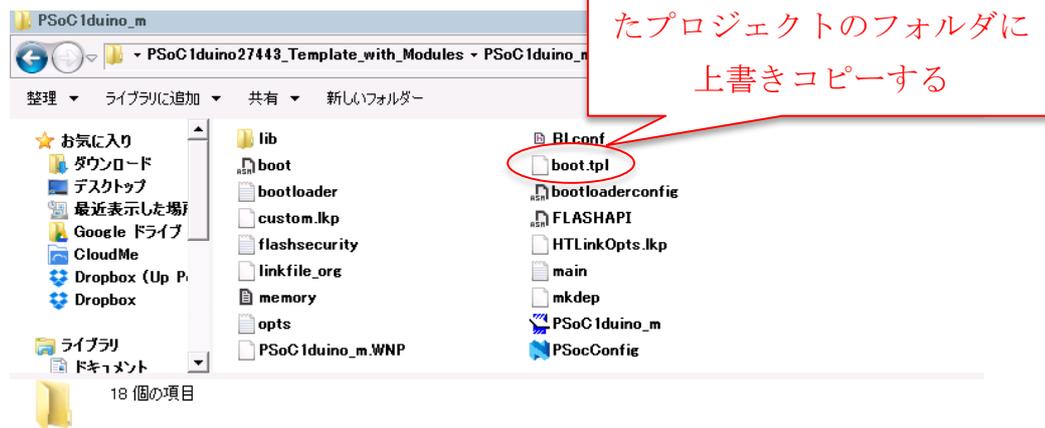
- 「PSoC1duino27443_Template_with_Modules」：ボード上の UART や LED のためのユーザモジュールが配置済みなので通常はこちらを使う。



- 「PSoC1duino27443_Template」: PSoC1 のユーザモジュール等を何も配置していない真っ白な状態からプログラム開発を始める場合はこちらを使う。
3. ユーザモジュールの更新などを尋ねられますが、とりあえず OK を押して更新しておきます。

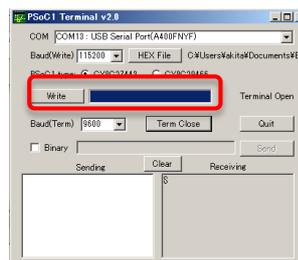


4. テンプレート内にある「boot.tpl」というファイルを、新規作成したプロジェクト内の「boot.tpl」に上書きしコピーして置き換える。(これを必ず行ってください)



5. 通常通り PSoC1 のプログラムを作成する。Device Editor でユーザモジュールを配置したり、プログラムを記述したりする。完成したら Generate や Build を行う。※一般的な PSoC Designer での PSoC1 のプログラム開発については、インターネット上の情報や書籍を参照してください

6. PSoC1term を起動し、できあがった HEX ファイルを指定して「Write」ボタンを押すと、書き込みが行われる。プログレスバーが右端まで達したら書き込み終了。



7. 書き込みが終了すると転送したプログラムが実行される。PSoC1term の TerminalOpen ボタンを押すと、シリアルターミナルが開き、リセットがかかってプログラムが実行される。ここではテキストまたはバイナリデータの送受信ができる。

6. 実践例

例 1: LED チカチカ

「PSoC1duino27443_Template_with_Modules」をテンプレートとして新規プロジェクトを作成し、main.c の main 関数に以下のようなプログラムを記述し、PSoC1duino に書き込んで実行させます。

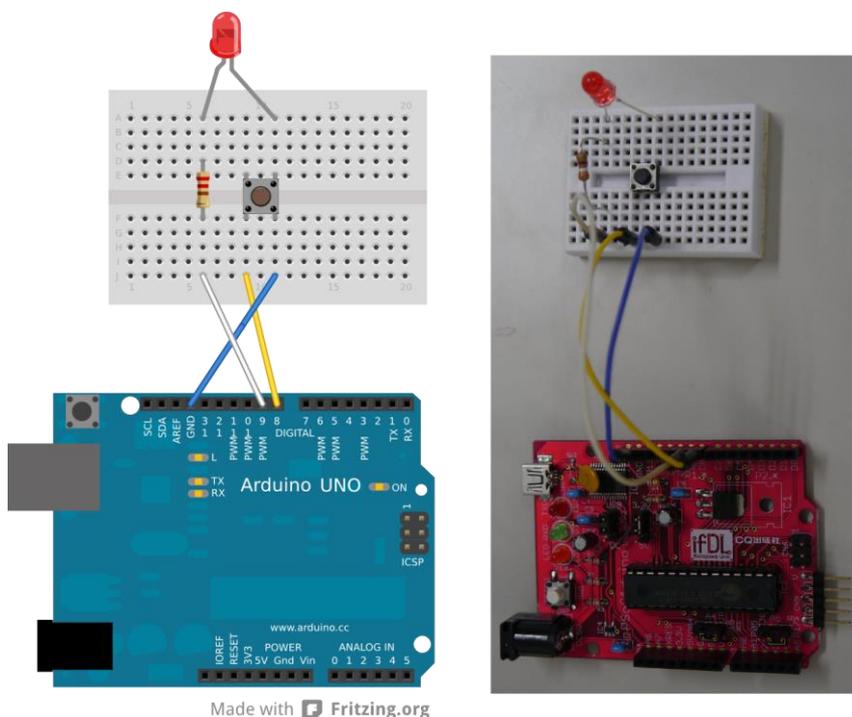
```
void main(void) {  
    WORD w;  
    LED_Start();  
    UART0_Start(UART_PARITY_NONE);  
    LED_On();  
    while(1) {  
        for (w = 0; w < 50000; w++);  
        LED_Invert();  
        UART0_PutChar('.');  
    }  
}
```

タイマを使った正確な「LED チカチカ」ではありませんが、PSoC1duino 上のオレンジ色 LED が点滅します。(おまけとして LED の点滅にあわせて PC へ"."を送信しています)

例 2: スイッチと LED

PSoC1duino に LED とスイッチをつなぐ例をみてみます。

1. ブレッドボードなど使って以下のように PSoC1duino と LED、タクトスイッチを接続します。



2. 「PSoC1duino27443_Template_with_Modules」をテンプレートとして新規プロジェクトを作成します。
3. Device Editor 画面で、ユーザモジュール LED を配置し、そのプロパティを右のように設定します。

Name	LED_1
User Module	LED
Version	1.40
Port	Port_1
Pin	Port_1_1
Drive	Active High

4. 同じく Device Editor 画面で、P1.0 (D8) ピンの Drive Mode を"Pull Down"に設定します。

P1[0]	Port_1_0, StdCPU, Pull Up
Name	Port_1_0
Port	P1[0]
Select	StdCPU
Drive	Pull Up
Interrupt	DisableInt
InitialValue	1

5. main.c の main 関数に以下のようなプログラムを記述し、実行させます。

```
void main(void){
  LED_1_Start();
  while(1){
    PRT1DR &= 0x01; // set P1.0="1"
    if (!(PRT1DR & 0x01))
      LED_1_On(); // if SW pressed, turn LED on
    else
      LED_1_Off(); // otherwise, turn LED off
  }
}
```

スイッチを押すと LED が点灯、離すと消灯します。

例 3: Touch Shield の使用

市販の Arduino 用シールドの例として、Sparkfun で販売されている Touch Shield を使う例をみてみます。Touch Shield は、Freescale 社のタッチセンサ IC の MPR121 と 9 個のタッチ電極が載った Arduino 用シールドで、通信方式は I2C、電源電圧は 3.3V です。

1. PSoC1duino の設定スイッチとジャンパピンのうち、以下の 3 つの設定を変更します。

- PSoC1 電源電圧: 3.3V を選択
- A4 ピン機能: P1.5 を選択
- A5 ピン機能: P1.7 を選択

2. PSoC1duino に Touch Shield を続します。



3. サポートページから”PSoC1duino_Touch.zip”をダウンロードして展開し、その中の Touch.app プロジェクトを開きます。

4. Generate と Build をしたあと、プログラムを書き込み実行させます。

5. Touch Shiled の数字ボタンに触れると、その数字の回数だけ LED が点滅します。

※このプロジェクト内の touchpad.c と touchpad.h は、Sparkfun で公開されて

いる Touch Shield 用のサンプルプログラムを整理したものです。低レベルの I2C 送受信関数を PSoC1 用書き換えたほかは、大半をそのまま流用しています。

表 1: PSoC1duino のピン配置

(サポートページに、名刺サイズにまとめたピン配置表がありますのでご利用ください)

D0-D7 コネクタ

D0(RX)	P2.0
D1(TX)	P2.1
D2	P2.2
D3	P2.3
D4	P2.4
D5	P2.5
D6	P2.6
D7	P2.7

D8-D13 コネクタ

D8	P1.0
D9	P1.1
D10	P1.2
D11	P1.3
D12	P1.4
D13	P1.6
GND	GND
AREF	-

A0-A5 コネクタ

A0	P0.0
A1	P0.1
A2	P0.2
A3	P0.3
A4(SDA)	P0.4/P1.5 (ショートピンで切替)
A5(SCL)	P0.5/P1.7 (ショートピンで切替)

電源/A6-A7 コネクタ

A6	P0.6
A7	P0.7
RESET	RST(負論理:Arduinoと同じ)
3.3V	3.3V(レギュレータ出力)
5V	5V(供給された5Vそのまま)
GND	GND
GND	GND
Vin	ACアダプタからの供給電圧

表 2: Arduino ICSP 互換コネクタのピン配置

	PSoC1 のピン	ArduinoICSP の機能
1	P1.4(D12)	SPI_MISO(D12)
2	VCC(PSoC1 電源)	VCC
3	P1.6(D13)	SPI_SCK(D13)
4	P1.3(D11)	SPI_MOSI(D11)
5	RES	RES
6	GND	GND

