

PSoCでPWMを使う

PWMとは、Pulse Width Modulationの略で、周期的な矩形パルス波を発生させるためのモジュールです。PWMを使うと、LEDの明るさや、ブザー音などの制御ができるようになります。

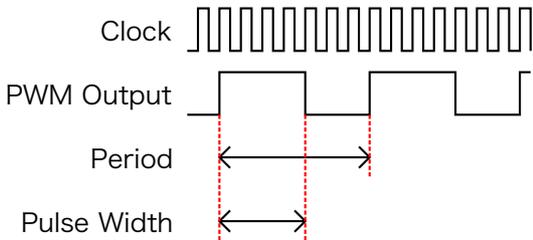


図.0

PWMを扱う上で重要なパラメータを、3つ挙げます。

(1)Clock - クロック

PWMを含むすべてのモジュールはクロックに同期して動作します。どのような周波数のクロックを入力するかで、そのモジュールの動作速度などが変化します。

(2)Period - 周期

“Period”は、PWMの出力の周期（周波数）を決定するパラメータです。クロックの立ち上がり何回分を1周期とするか、で指定します。

※注意：PSoCでは（クロックの数）-1を指定するという仕様になっています。

例えば図.0の場合、1周期中にクロックが7つあるので、Periodは6です。

(3)PulseWidth - パルス幅

“Pulse Width”は、PWMの1周期のうち、どの期間だけ“1”を出力するかを決定するパラメータです。Periodと同じように、クロックの回数でパラメータを指定します。

※注意：PSoCでは（クロックの数）-1を指定するという仕様になっています。

例えば図.0の場合、1周期中で“1”の期間はクロック4つ分なので、PulseWidthは3です。

それでは、以下の仕様の矩形パルス波を出力するための手順を解説します。

- ・周期0.1[s] (=周波数10[Hz])
- ・パルス幅0.025[s] (=デューティ比25%)

まずは、プロジェクトを新規作成します。

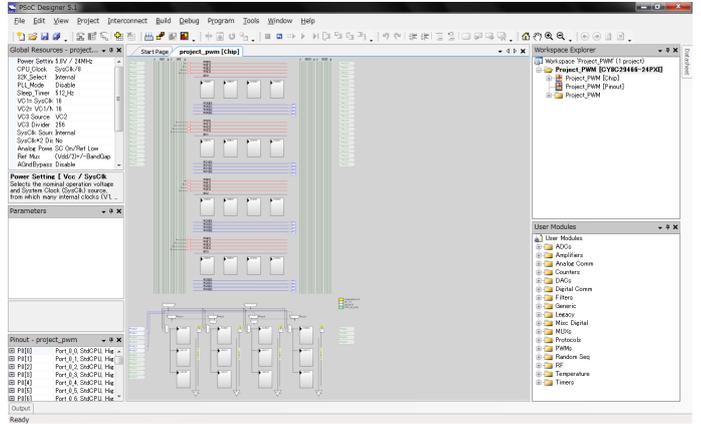


図.1

右下のツリーから、PWM8というモジュールを探して、ダブルクリックしてください。

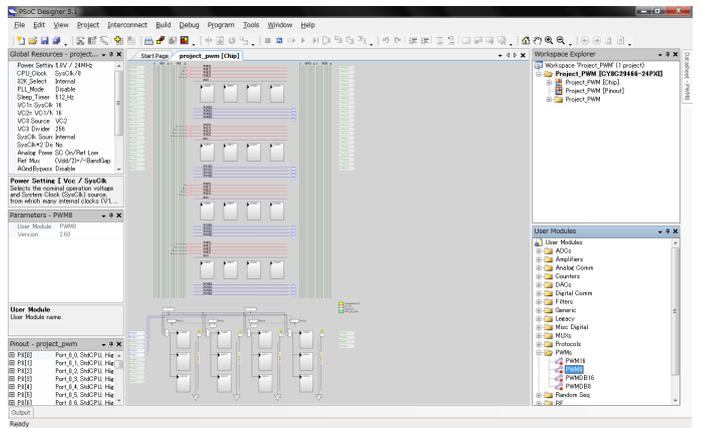


図.2

プロジェクトにPWM8モジュールが追加されました。変更点としては、

- (1)画面左中段にPWM8のパラメータが表示された。
 - (2)画面中央に青色のPWM8モジュールが設置された。
 - (3)画面右上のツリーにPWM8が追加された。
- です。

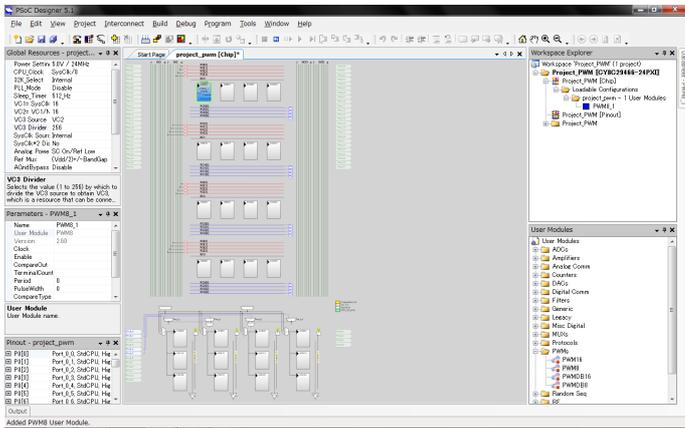


図.3

次に、PWM8モジュールに入力するクロックを作成します。クロックの作成は画面左上でおこなうことができます。

PSoCの内部クロックは、システムクロックと呼ばれる24[MHz]のクロックを分周することで作成されます。ユーザがカスタマイズできるクロックとして、VC1、VC2、VC3の3つが用意されています。

“VC1=SysClk/N”は、VC1の周波数を決定する項目です。VC1の周波数は、システムクロックの周波数（24[MHz]）を1～16で割ったものに設定できます。例えば、この項目に16を入れておくと、VC1の周波数は $VC1=24[MHz]/16=1.5[MHz]$ になります。

“VC2=VC1/N”は、VC2の周波数を決定する項目です。VC2の周波数は、VC1の周波数を更に1～16で割ったものに設定できます。例えば、“VC2=VC1/N”の項目に15を入れておくと、 $VC2=1.5[MHz]/15=100[kHz]$ になります。

VC3は他の2つと比べると少し特殊です。まず、“VC3 Source”の項目で、どのクロックを分周するかを選択することができます。そして、“VC3 Divider”の項目で、何分周にするかを決定します。“VC3 Divider”に設定できる値は1～256となっており、分周の幅が大きくなっています。ここでは、“VC3 Source”をVC2に、“VC3 Divider”を100に設定しました。すなわち、 $VC3=100[kHz]/100=1[kHz]$ です。

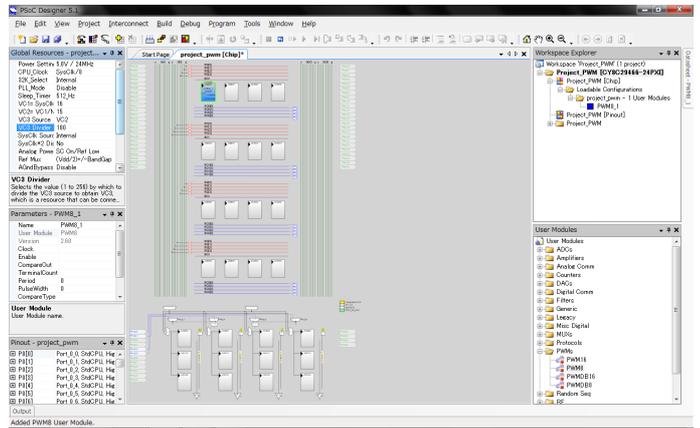


図.4

まとめると、下のようになります。

$$\begin{aligned}
 &\text{System Clock} = 24[\text{MHz}] \\
 &\quad \downarrow /16 \\
 &\text{VC1} = 1.5[\text{MHz}] \\
 &\quad \downarrow /15 \\
 &\text{VC2} = 100[\text{kHz}] \\
 &\quad \downarrow /100 \\
 &\text{VC3} = 1[\text{kHz}]
 \end{aligned}$$

PWM8モジュールのパラメータを指定します。パラメータの設定は、画面左中段でおこないます。

(1)Clock :

先に設定したVC3を入力します。これで、PWM8モジュールのクロックは1[kHz]となります。1[kHz]の周期は、1[ms]です。

(2)Period :

はじめの仕様を見ると、周期は0.1[s]、つまり100[ms]です。クロックに入力されているVC3の周期は1[ms]ですから、クロック100回分が100[ms]となります。なので、“Period”の項目は99（※注意）となります。

(3)PulseWidth :

パルス幅は0.025[s]=25[ms]です。つまり、クロック25回分が25[ms]となるので、“PulseWidth”の項目は24（※注意）とします。

これで、PWMの重要なパラメータ3つの設定はできました。

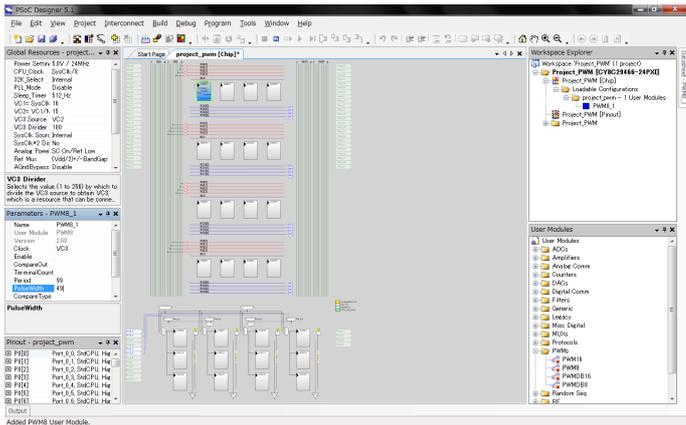


図.5

まだ、左中段のPWMのプロパティで、空欄の項目があるので埋めます。"Compare Out" (矩形パルス波の出力先) 以外の項目はあまり本質的ではないので、ここでは解説しません。スクリーンショットを真似して入力してください。

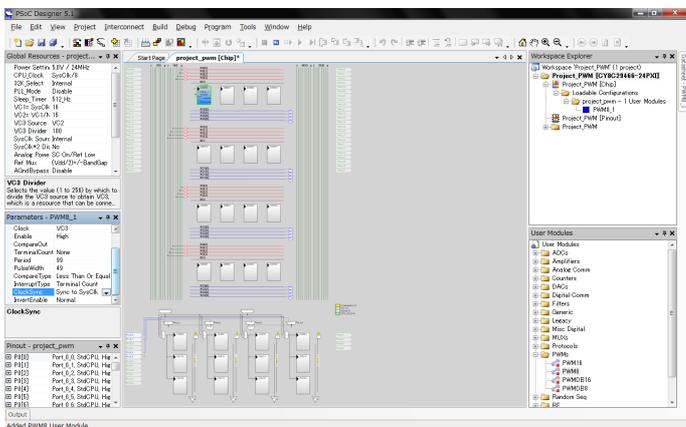


図.6

次に、矩形パルス波の出力するピンの設定をします。ピン設定の画面を開き、出力させたいピンをクリックします。今回はPort_0_7 (左上のピン) を選択しました。するとピンのプロパティが出てくるので、設定をします。

(1)Select :

内部配線をどこに繋ぐかを選択します。入力ピンであれば"GlobalIn..."、出力ピンであれば"GlobalOut..."に設定します。

今回は出力ピンなので"GlobalOutEven_7"を選択しています。

(2)Drive :

ピンの入出力のタイプを選択します。基本的に、デジタル入力の場合は"High Z"、デジタル出力の場合は"Strong"、アナログ入出力のときは"High Z Analog"です。

矩形パルス波は"0"か"1"のデジタル値です。今回はそれを出力させたいので"Strong"に設定します。

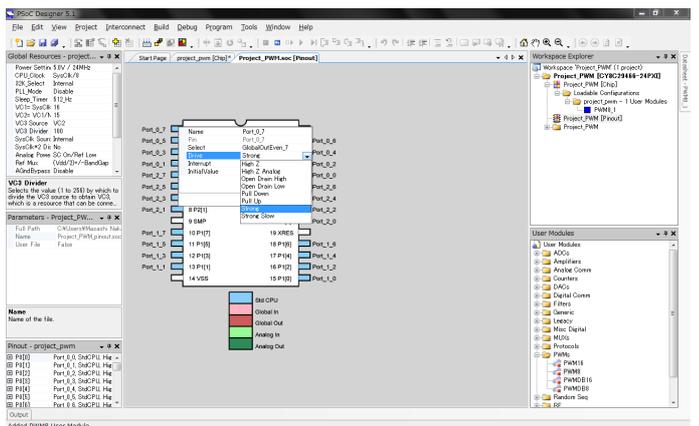


図.7

設定したら[OK]を押します。うまくいけばピンが赤色になります。

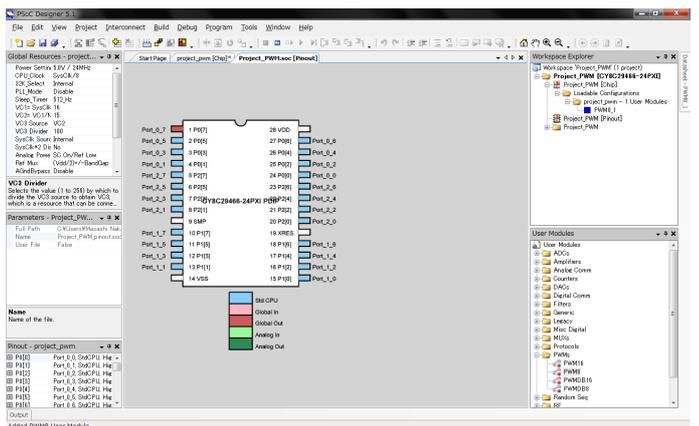


図.8

内部配線の画面に戻ります。すると、右側のPort_0_7から線が伸びています。これが先ほど設定した"Select"の"GlobalOut..."です。

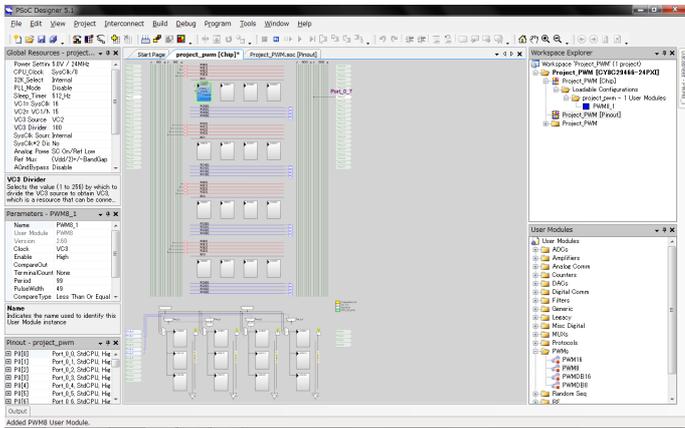


図.9

PWM8モジュールから出力ピンへ信号が渡るように、内部配線の設定をします。まず、PWM8モジュールの残った設定項目である”CompareOut”を”Row_0_Output...”にします。今回は”Row_0_Output_3”です。設定すると、PWM8モジュールから配線が伸びます。

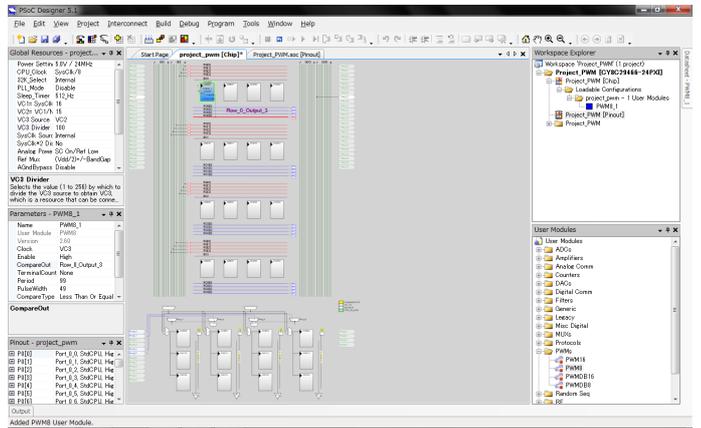


図.11

クリックするとウィンドウが開きます。ここで青色の横線と緑色の縦線の接続を設定します。出力ピンに繋がっている緑色の縦線に、ちゃんと接続されるような三角形をクリックします。クリックすると選択項目が出るので、”GlobalOut...”を選択します。

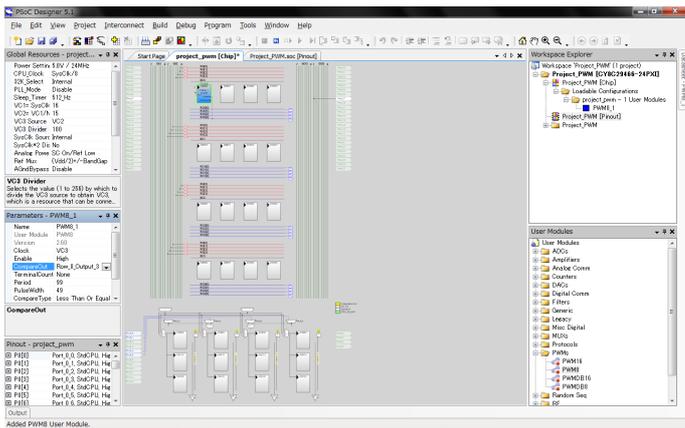


図.10

次に、PWM8モジュールから伸びた配線の先の、青色の横線をクリックします。

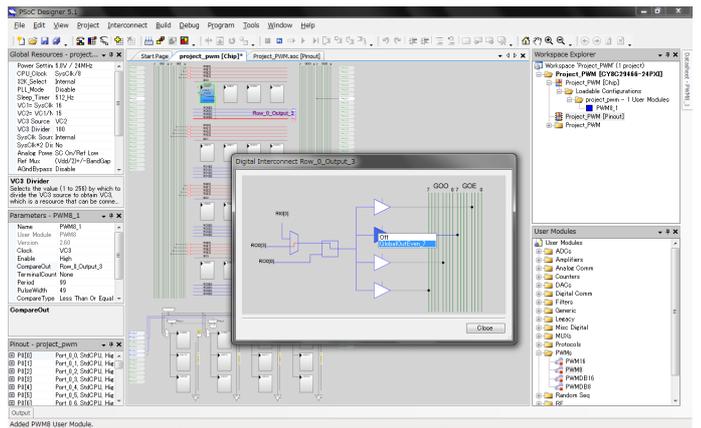


図.12

これで内部配線ができました。PWM8モジュールから、出力ピンまで、配線で繋がれていることを確認してください。

ハードウェアの設定が終わったら、次はソフトウェアの設定（プログラム）です。プログラム記述の画面を開きます。

main関数の中に、以下の記述をします。

```
PWM8_1_Start();
while(1);
```

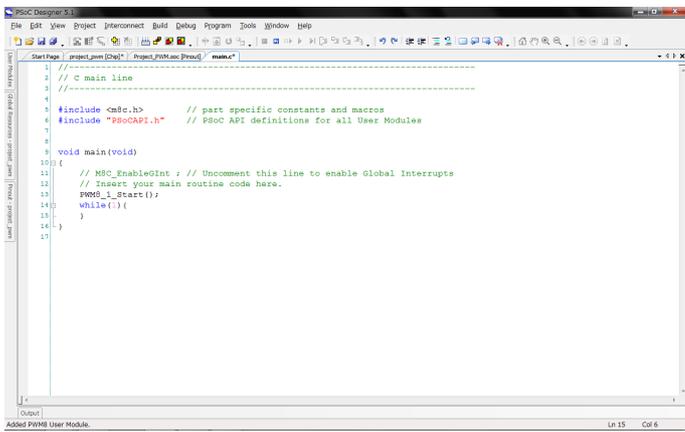


図.13

PWM8_1_Start(void)は、PWM8モジュールを起動する関数です。これをしなければPWM8モジュールから矩形パルス波は出力されません。while(1)は、無限ループです。

以上ができれば、Generate（コンパイルみたいなもの）をします。メニューバー下のアイコンをクリックします。

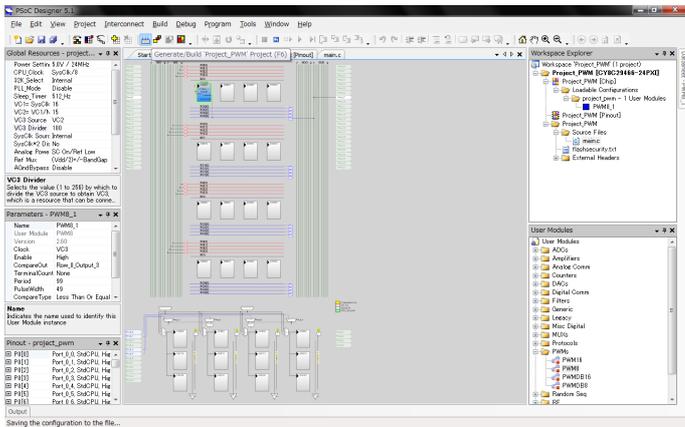


図.14

問題が無ければ、0 error 0 warningと出ます。

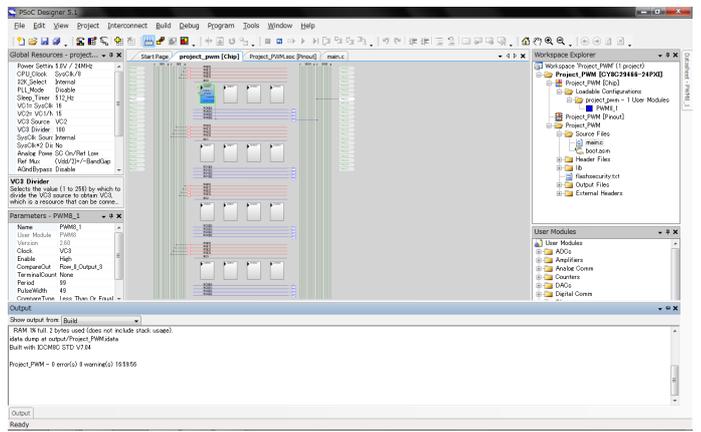


図.15

最後に、マイコンに書き込みをおこないます。メニューバーから、[Program]→[ProgramPart]をクリックします。

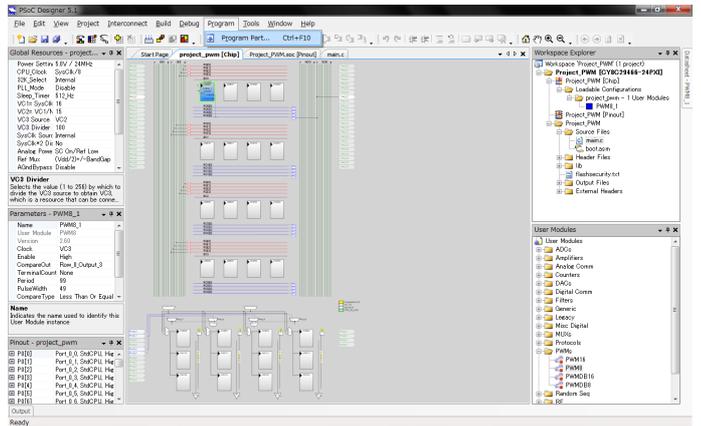


図.16

マイコンをUSB接続して、ダウンロードのボタンを押すと書き込みが始まります。

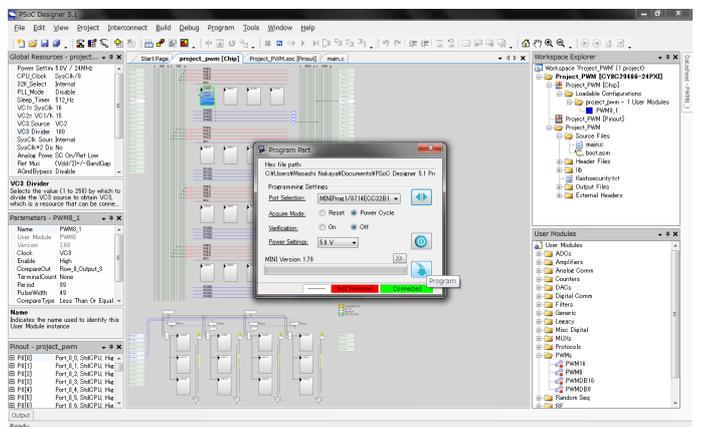


図.17

おわり