

## アナログ信号の増幅の演習

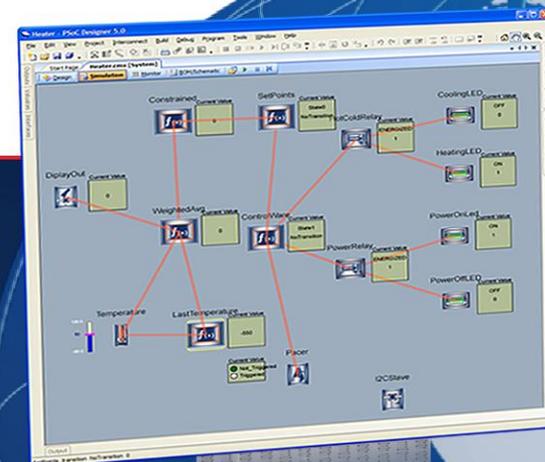
# *motor* *PSoC Experiment* *Lab*

Experiment Course Material V1.21

April 9<sup>th</sup>, 2019

motor.pptx (34Slides)

Renji Mikami





# ラボで使用するファイルのルール

Labで使用する資料,ファイルのディレクトリの作成ルールは、以下のとおりとします。絶対にまちがわないように注意してください

各自の演習ファイルの置き先:C:¥psoc\_lab¥などのユニークな名前をつけたマスタディレクトリを作成します。(前の組が演習で作成した同名のC:¥psoc\_lab¥がある場合は削除してかまいません)

各ラボのファイル群はC:¥psoc\_lab¥のサブディレクトリに作成してください。例：ラボ名 moter の場合は各自で作成するのはファイル・ディレクトリ C:¥psoc\_lab¥motor となります

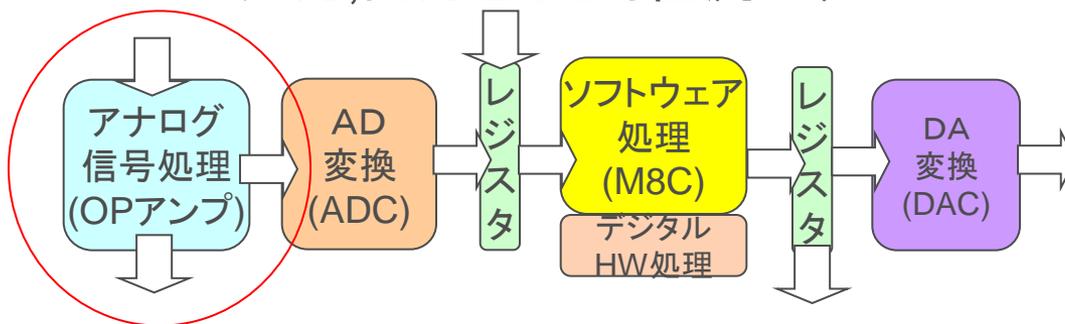
解答例となる”完成プロジェクト”は、デスクトップのpsoc\_lab\_master(またはpsoc\_lab\_master201X)というディレクトリにあります。ない場合にはWEBのサポートURLからダウンロードしてデスクトップに置いてください。



# ラボ motor

## アナログ信号の増幅

非反転増幅器PGAを使用してアナログ増幅回路を作ります。  
入力,出力どちらも直流です。





# ラボ motor 手順

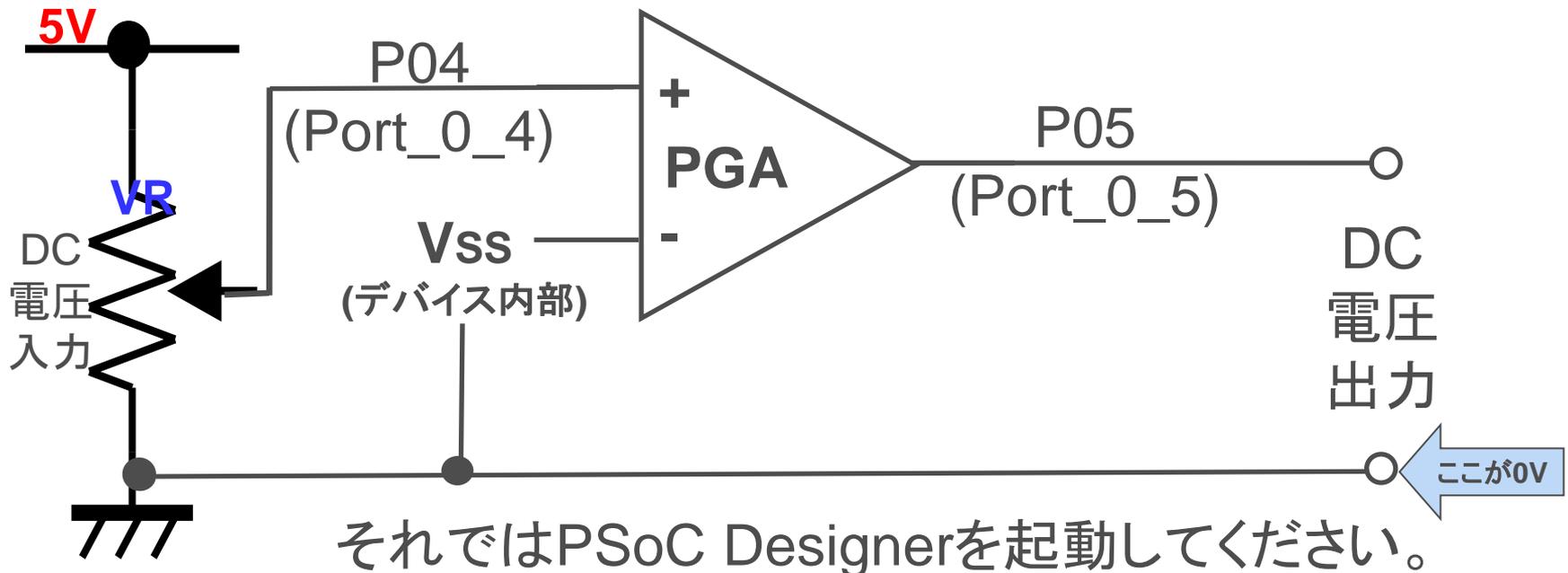
- 1.PSoC Designer でPGA ユーザーモジュールを使用して単電源非反転型オペアンプによる増幅回路を設計します
- 2.PSoC Programmerでデバイスに書き込みします
- 3.デバイスの出力で振動モーターを動かしてみます

解説：この回路では外部から加えた直流の入力信号を増幅して外部の負荷を駆動します.単電源非反転型オペアンプのリファレンス入力がVSSになっていることを考慮してください

PGA : Programmable Gain Amp

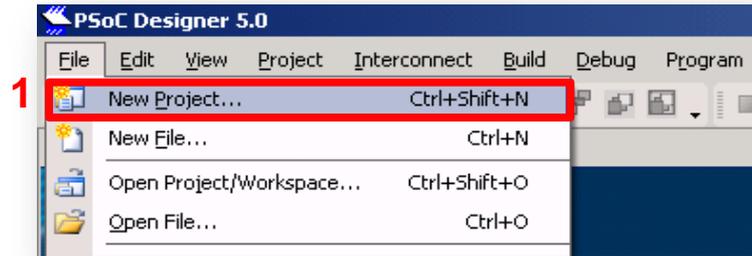
# ★プロジェクト motorの回路

- DC出力に携帯電話用のバイブレーションモーターを接続します。
- 入力に可変のアナログDC電圧を印加し出力のアナログ電圧を制御
- PGAのゲインはプロパティウインドウで設定します。
- PGAの定格、特性はデータシートウインドウで見ることができます。
- 各ピンの設定はピンアウトウインドウで行います。
- PGAのリファレンスはVssであることに注意してください。入力には**正の信号(電圧)**を加えます。出力には**正の信号(電圧)**が出ますがこの信号は電源電圧を超えることはありません



# 新規プロジェクトの作成(旧版ソフトウェアの場合)

1. File > New Project をクリック



2. Chip-level Project を選択

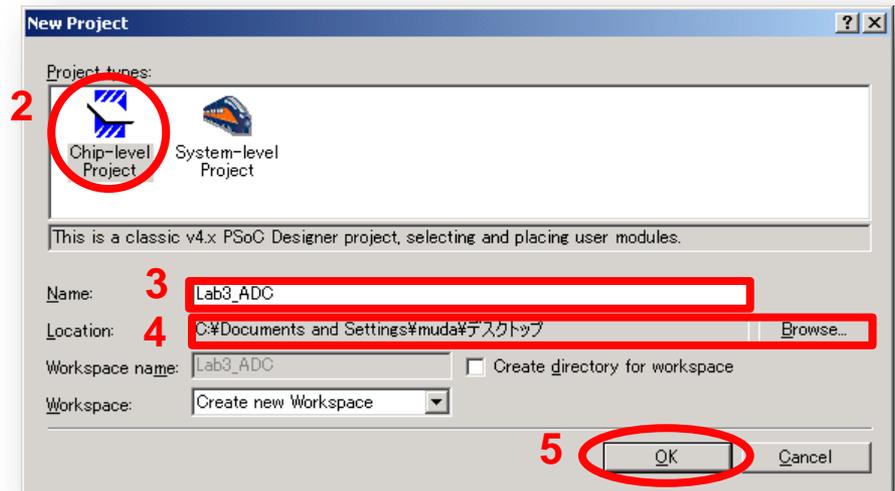
3. Name を入力

例: motor

4. Location を選択

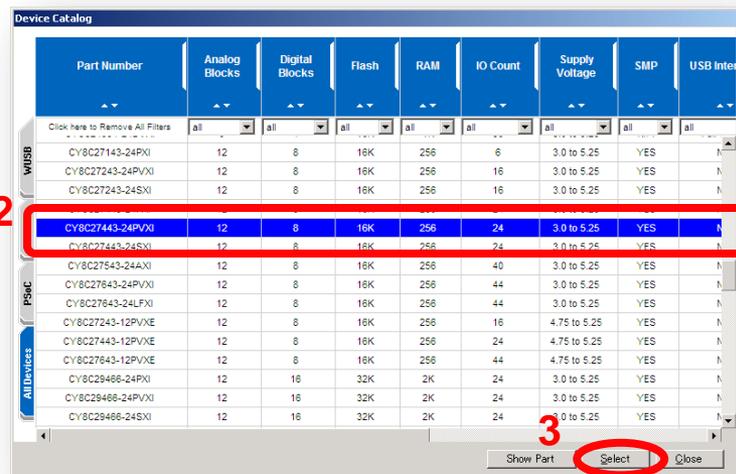
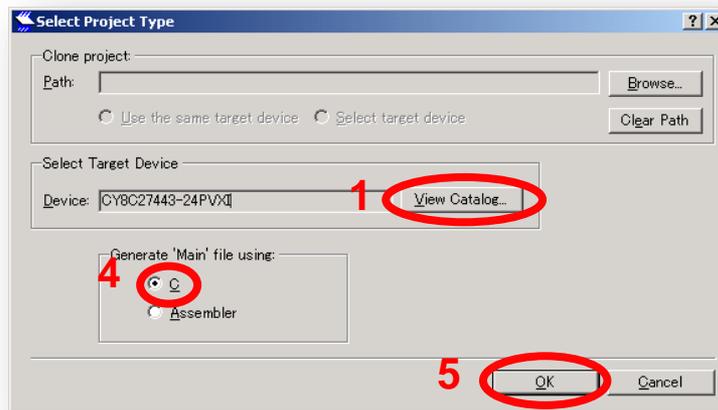
例: C:\psoc\_lab\motor

5. OK をクリック



# 使用するPSoC、言語の選択(旧版ソフトウェアの場合)

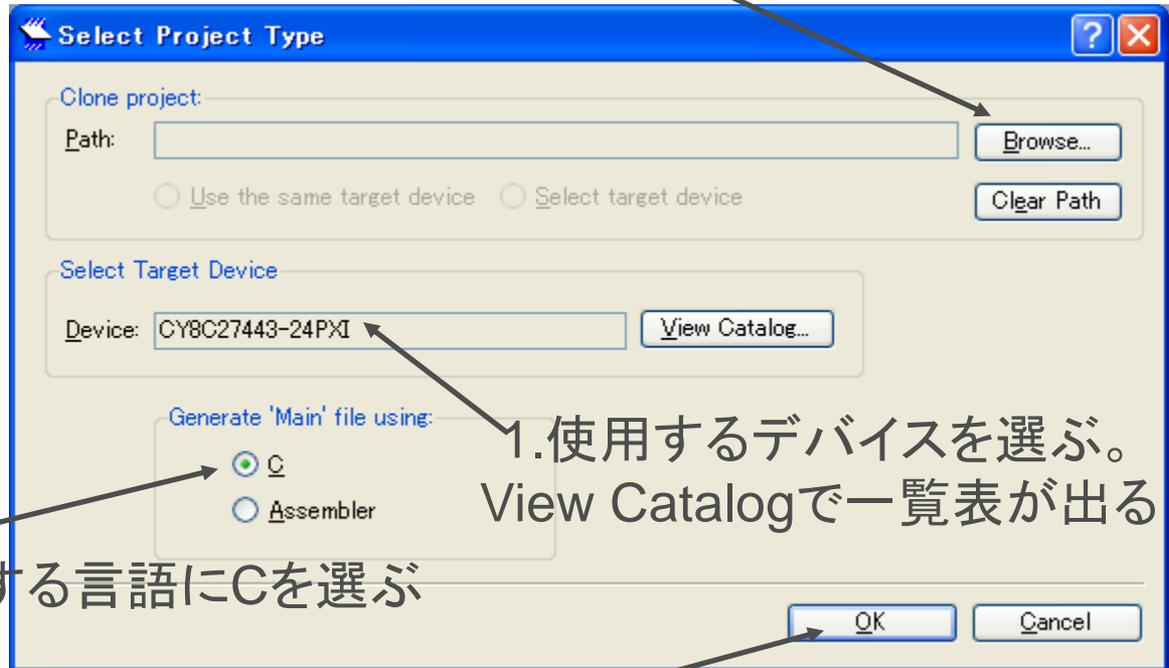
1. View Catalog をクリック
2. CY8C27443-24PXI を選択
3. Select をクリック
4. C を選択
5. OK をクリック





# クローンの作製(参考-オプション)

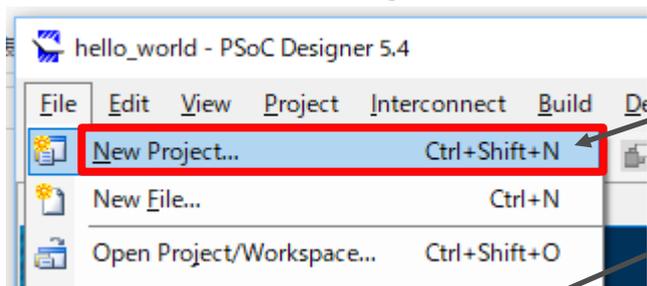
クローン化に関しては、Lab2\_pwm\_lcdを参照してください  
既存のプロジェクトをコピーして使用する場合には、クローン作製をします  
コピー元のプロジェクトの場所をBrowseするとコピーが作成されます。  
使用デバイスを変更する場合などに使います。



- 1.使用するデバイスを選ぶ。  
View Catalogで一覧表が出る
- 2.設計に使用する言語にCを選ぶ
- 3.決まったらOKをクリック



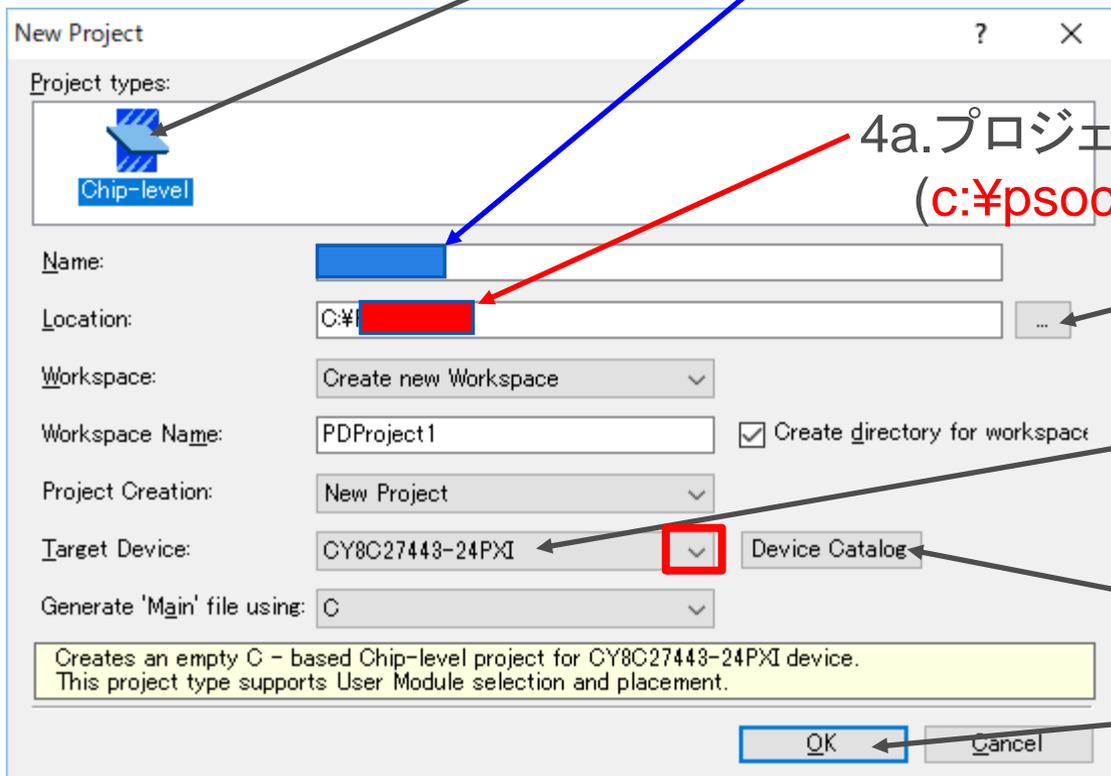
# New Project: motor の作成 (新版ソフトウェアの場合)



1. File > New Projectをクリック

2. Chip-level Project をハイライト

3. プロジェクトの名前(motor)を入力



4a. プロジェクトを保存するディレクトリ (c:¥psoc\_lab¥motor) 指定する。

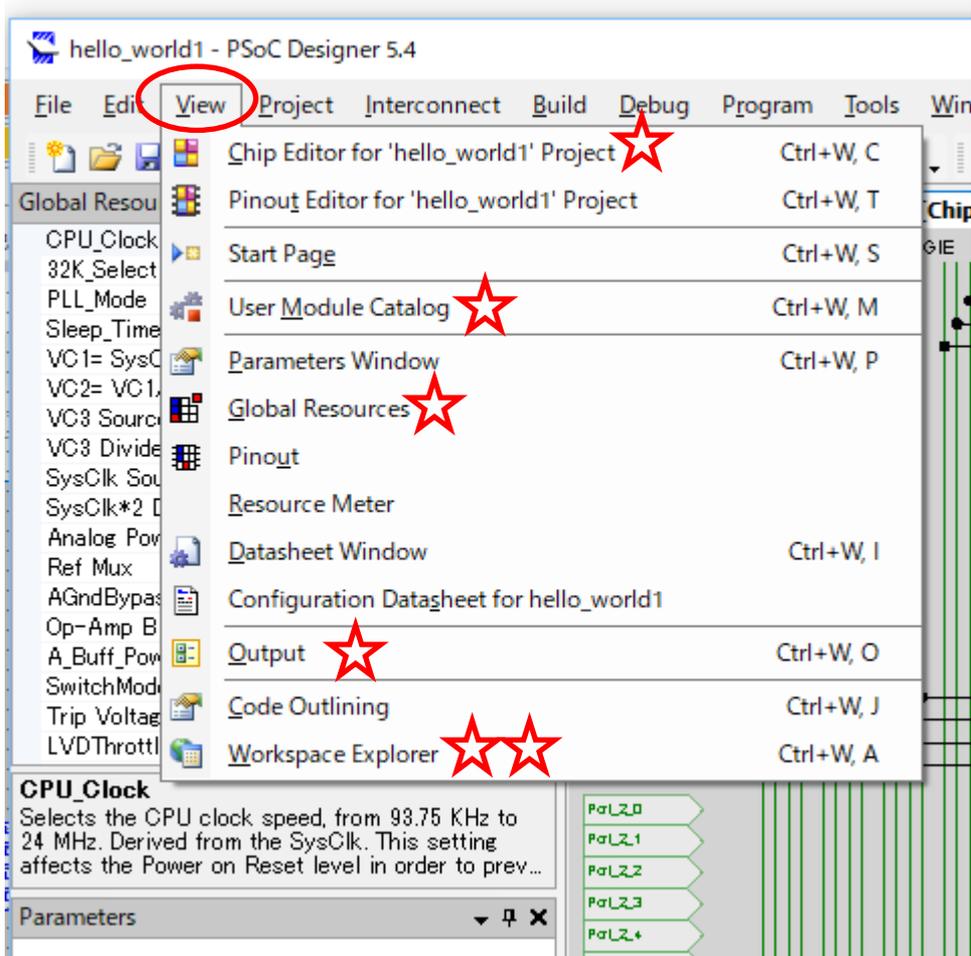
4b. 場所を選ぶときは 右の...をクリック。

5. デバイス CY8C27443-24PXIを確認 (変更はVマークかDevice Catalogをクリック)

6. 決まったら OKをクリック



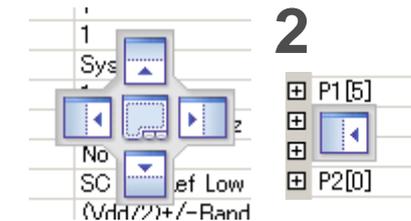
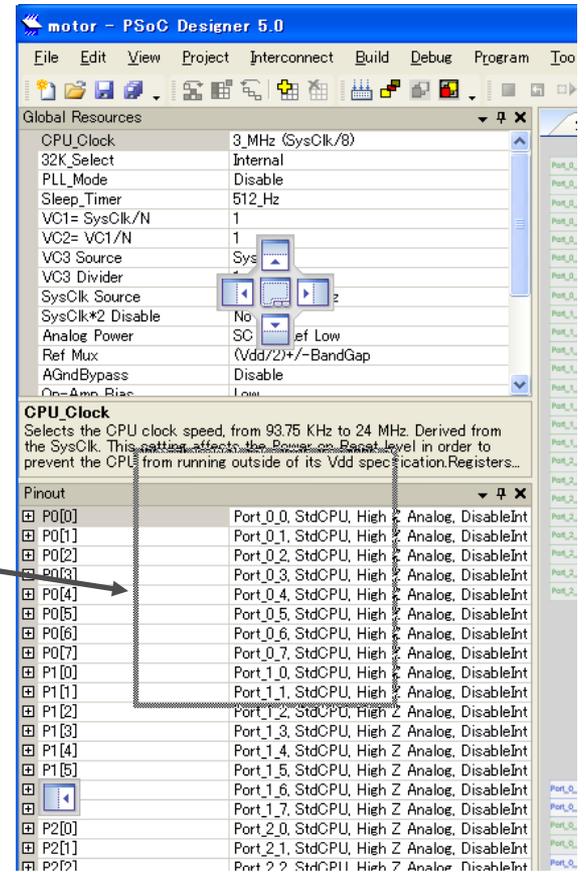
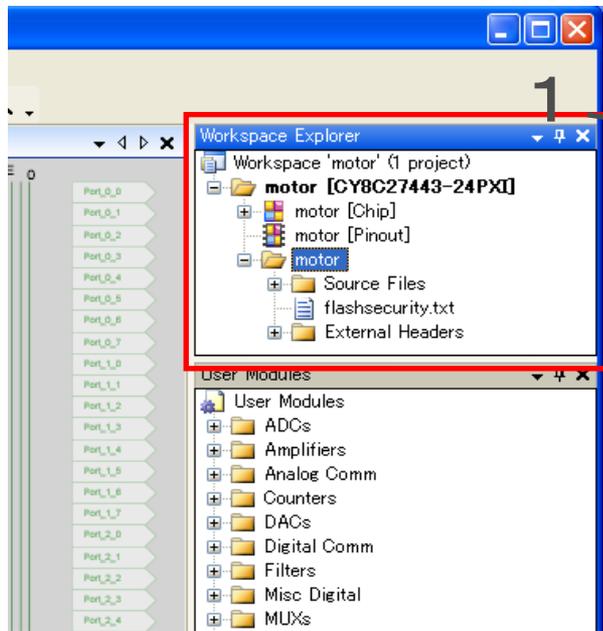
# PSoC Designerの各ウィンドウの開き方



- 必要なウィンドウを開くためには、Viewタブをクリックして選択する。以下がよく使うもの。Workspace Explorerはいつも開いておくとよい。
- Workspace Explorer
- Output Window
- Chip Editor
- User Module Catalog
- Global Resource

# ウィンドウの場所を自由に変更してみる

- ためしにWorkspace Explorer をクリックしたままで、画面の右端とか左端まで動かしてみる。
- 任意の場所でクリックを離すとその場所にウィンドウが開く(フローティング)

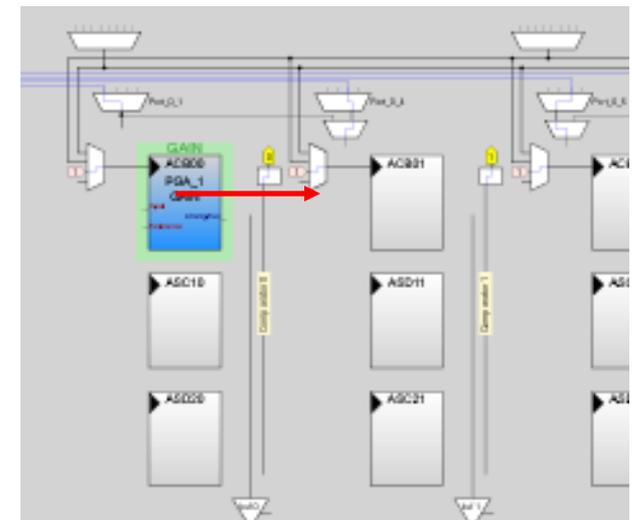
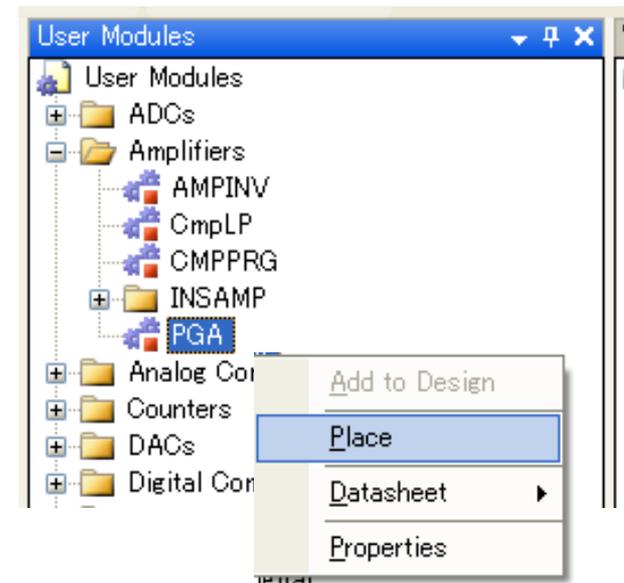


- 2.上の十文字や方向タブの場所でクリックを離すとPSoC Designerにドックした状態でウィンドウが固定される.ちょっとやってみよう

# User Module CatalogからPGAを選ぶ

View > User Module Catalogをクリック

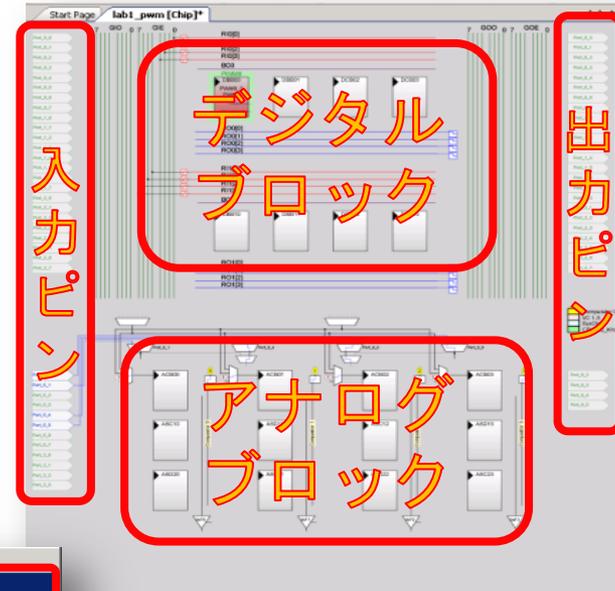
- 1.ユーザーモジュールカタログのウィンドウを開いてAmplifierグループの下にあるPGAをハイライトする。
- 2.右クリックでDatasheetを開くことができる。またPlaceを選ぶかPGAをダブルクリックすると配置が行われる
- 3.自動的にPGA\_1とリネームされ下段アナログ部の左上に配置される
- 4.配置されると下図のように色が変わって示される。配置されたPGAを左クリックしたまま移動してクリックを離すと、別の場所に配置変更ができる。
- 5.右に移動してみましよう。(結線をした後では動かせません。) **以降右に移動した位置から配線を進めます。**



# Chip Editor (回路図)の使い方

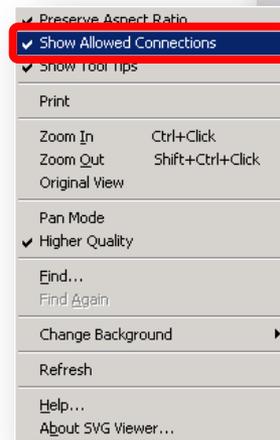
View > Chip Editor

- Alt + ドラッグで移動
- Ctrl + クリックで拡大
- Ctrl + Shift + クリックで縮小



- 回路図上で右クリック

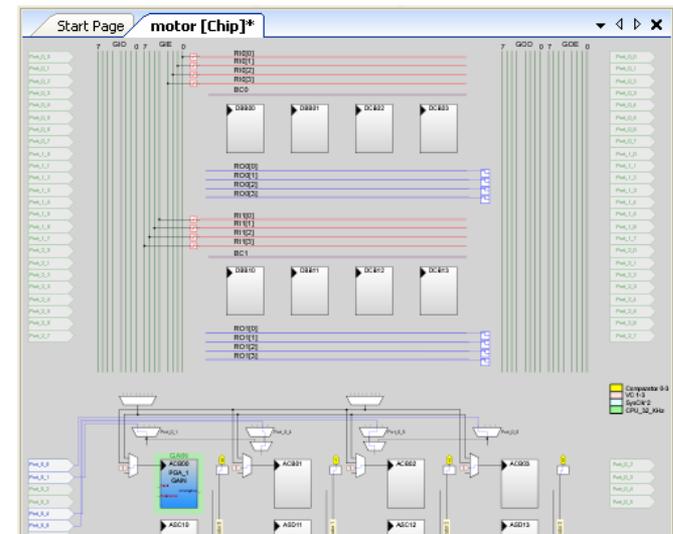
Show Allowed Connections  
で配線候補を可視化



移動	Alt+ドラッグ
拡大	Ctrl+クリック Ctrl+ドラッグ
縮小	Ctrl+shift+クリック Ctrl+shift+ドラッグ

# ズームとパンモードの切替

- 拡大、縮小アイコンでChip Editorをズームできます。任意の場所を拡大するには、Ctrlキーを押しながらマウスで拡大部分指定します。
- 手のアイコンは、矢印カーソルを出して配線を行うモードと画面を移動するパンモードを切り替えます。
- 手のマークのカーソルが出ているパンモードでは、画面の移動ができますが配線はできません。
- 矢印のカーソルが出ているときには、配線ができますが、画面は固定されます。
- 移動とズームをためしてみてください



Chip Editor ウィンドウ

# PGAピンから配線を試みる

1.配線は、ユーザー・モジュールのピンからバスなどの配線リソースに接続していきます。接続できる配線路はユーザーモジュールを配置した場所に依存します。

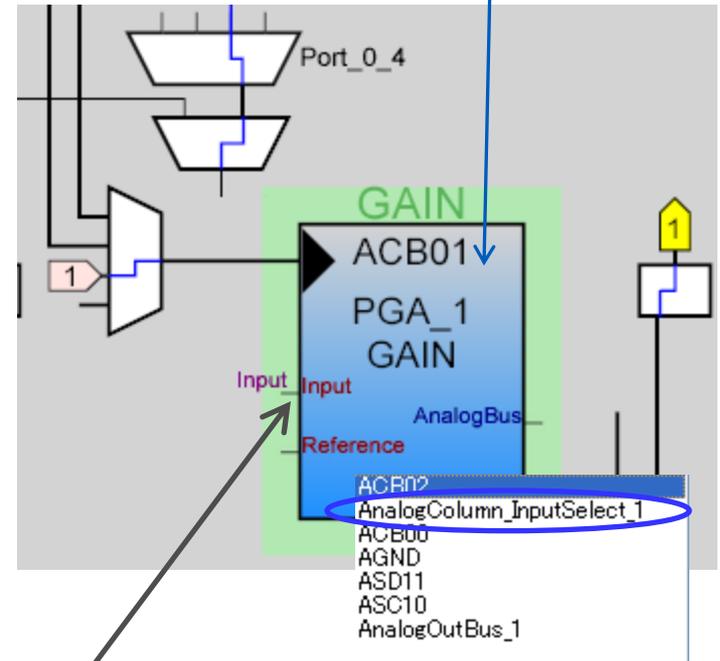
2.ピンにマウスカーソルを近づけるとピン名が現れる場所があります。そこが接続を指定できる場所です。

3.左クリックすると接続する場所を指定するボックスが現れますので、上から2番目のAnalogColumn\_InputSelect\_1をハイライトしてください

4.PGAの入カピンにつなぐ配線が現れます。

PGAの配置場所が左から2番目の

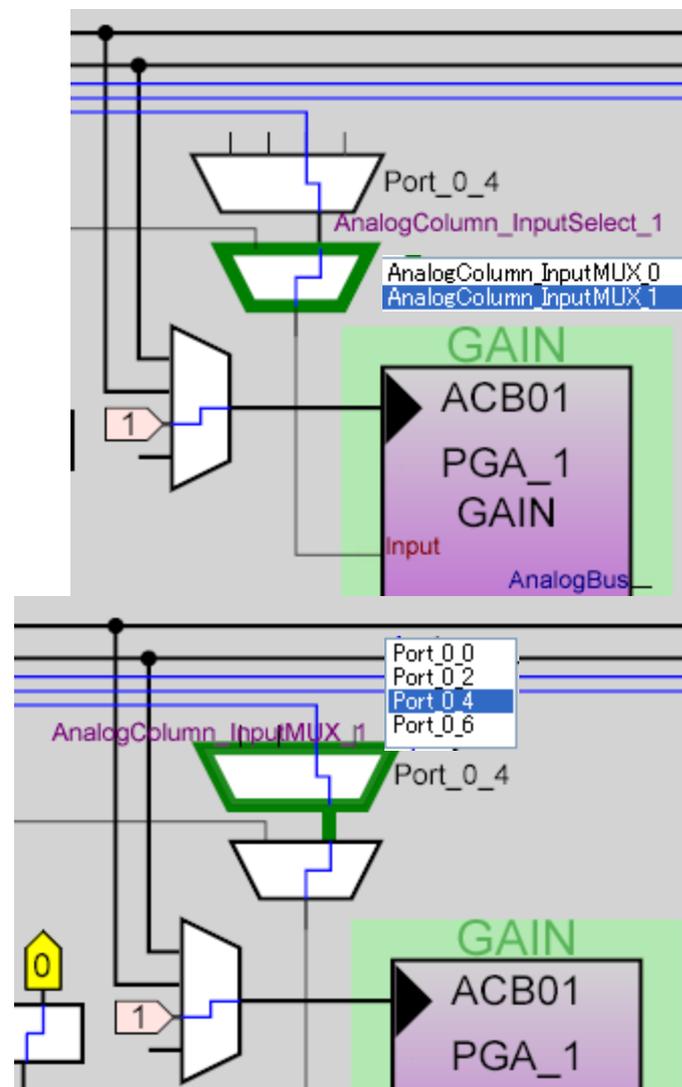
アナログブロックであることを確認してください。



Inputピンにカーソルを近づけるとInputという文字が現れる

# 配線をつないでピンまでもっていく

- 1.新たに現れた配線がつながっている先にカーソルを持っていくと緑色に表示される配線選択路が表示されます。
- 2.ここでは入出力ピンのあるポートに向かって配線していくので、上のPort\_0\_4と書かれたシンボルに向けて接続をえらびます。
- 3.さらに上のシンボルをハイライトしてピンにむけて配線を設定します。
- 4.このシンボルからはPort\_0\_0, Port\_0\_2, Port\_0\_4, Port\_0\_6に配線できます。
- 5.**Port\_0\_4**を選んで左側のどのピンの場所につながっているか確認してください。実際のデバイスの**Port\_0\_4**ピンは評価基板のボックスソケットの**P04**に配線されています。



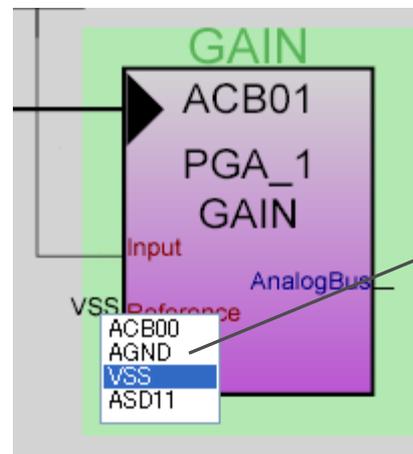
# リファレンス入力を設定

1. PGAユーザーモジュールのリファレンス・ピンをクリックしてリファレンスをVSSに設定してください。

設定されるとVSSと表示されます。

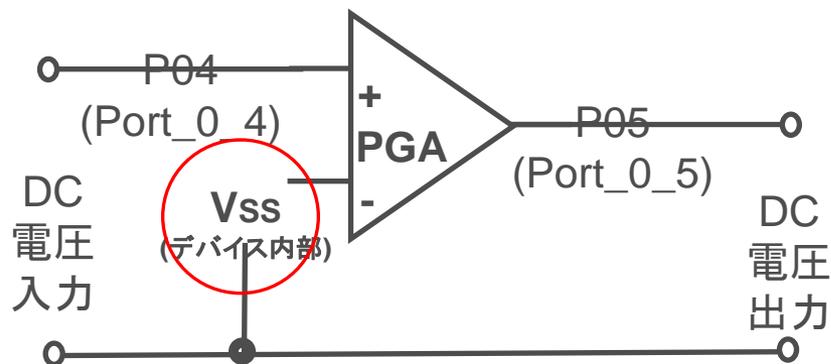
2. 差動増幅回路では入力信号の差分を増幅しますが単電源の場合は出力が0V以下にはなりません。ここではリファレンスをVSSにっていますので入力  
が正の電圧の場合のみ扱えます。出力の最大は供給電圧になります。

3. 正負にスイングする交流信号を増幅する場合には、リファレンス電圧を変更して入力をオフセットします。出力もこのリファレンス・レベルに対するゲインとなります。



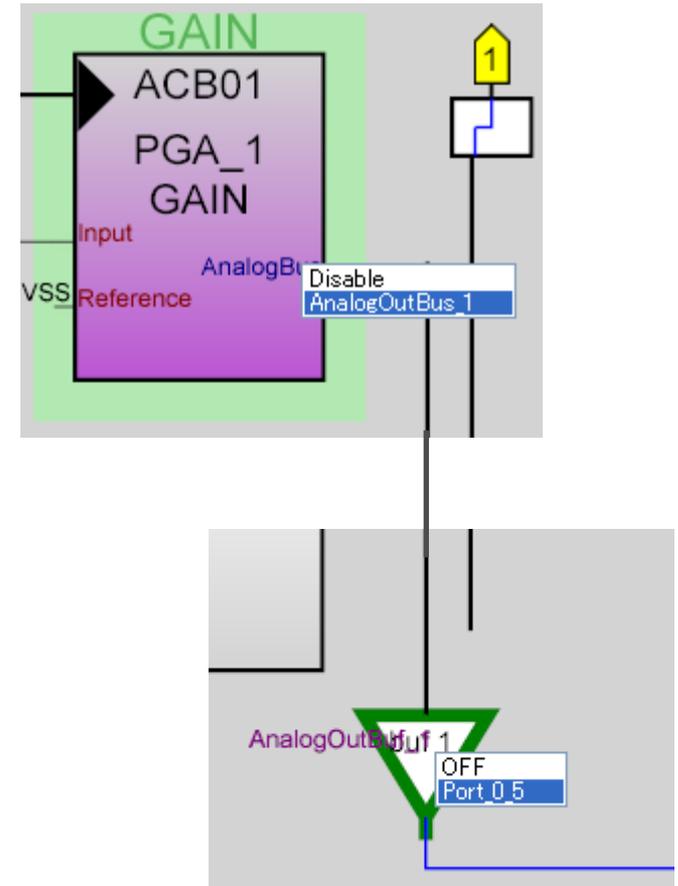
AGNDは  
アナログ  
グラウンドの  
意味です。

lab3\_adcで  
出てきます



# PGA出力のピンへの接続

1. AnalogBusピンをクリックして AnalogOutBus\_1 を選択します。  
接続されたラインを確認してください。
2. 下のほうにあるバッファ・シンボルから 接続するピン(ポート)を選択します。
3. バッファの出力をたどって Port\_0\_5ピン に出力が接続されているのを確認してください。
4. また評価基板のボックスソケットの P05 に対応していることを確認してください

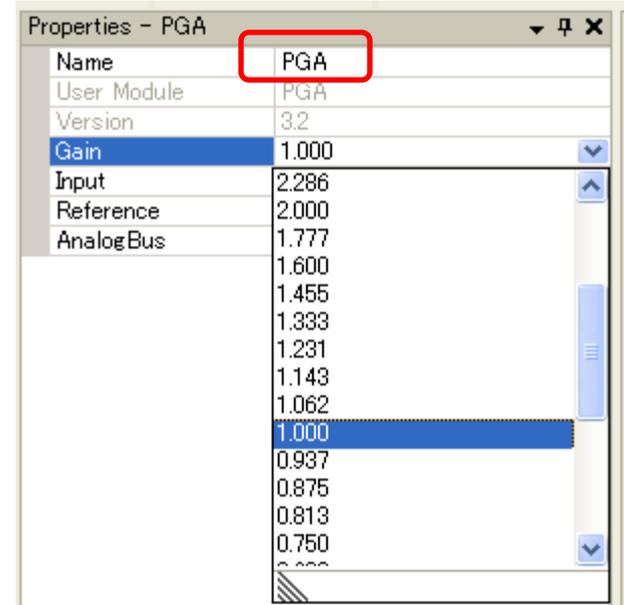


# PGAパラメータの設定

1. PGA\_1をPGAに変更(PSoC Designerではモジュールを識別するために数字の番号が自動的に割り当てられます。ここでは1つだけなのでPGAという名前にします。)

2. 続いてGainを1.000に設定します。後で電圧増幅率を自由に変えてみてください。(1.0以下はボルテージフォロワ)

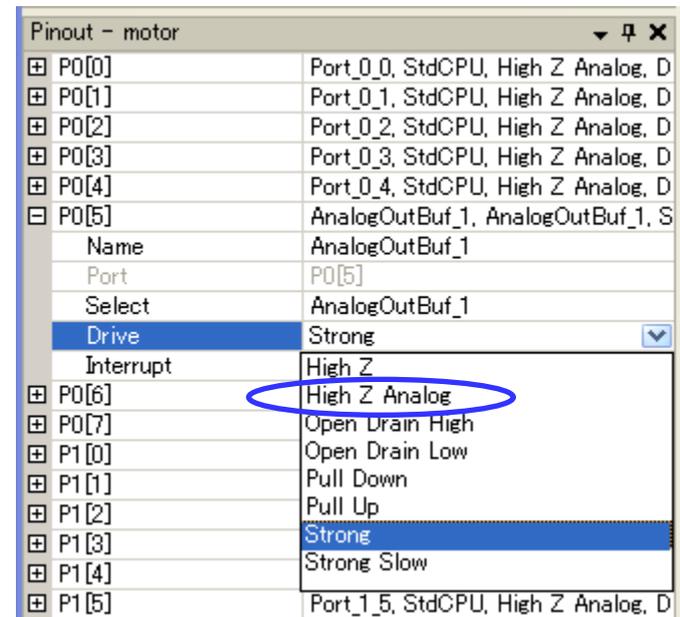
3. 今回は基板上の可変抵抗から5Vを分圧して印加しますので、詳細なコントロールが必要な場合は値を小さくします。



# ピンアウトパラメータの設定

1.PSoCのピンは入力にも出力にも設定できます。またアナログ、デジタル双方の信号を扱うことができます。またデバイスの内部でプルアップやプルダウンの設定もできます。

2.PGAの出力をアナログでP05に出力しますから**High Z Analog**に設定します。(詳細はGPIOの説明を参照してください。)

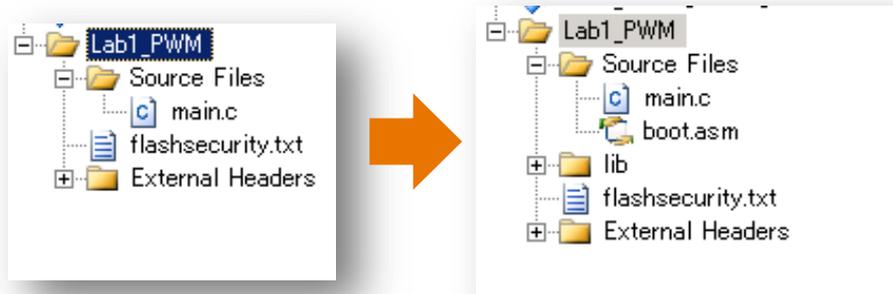
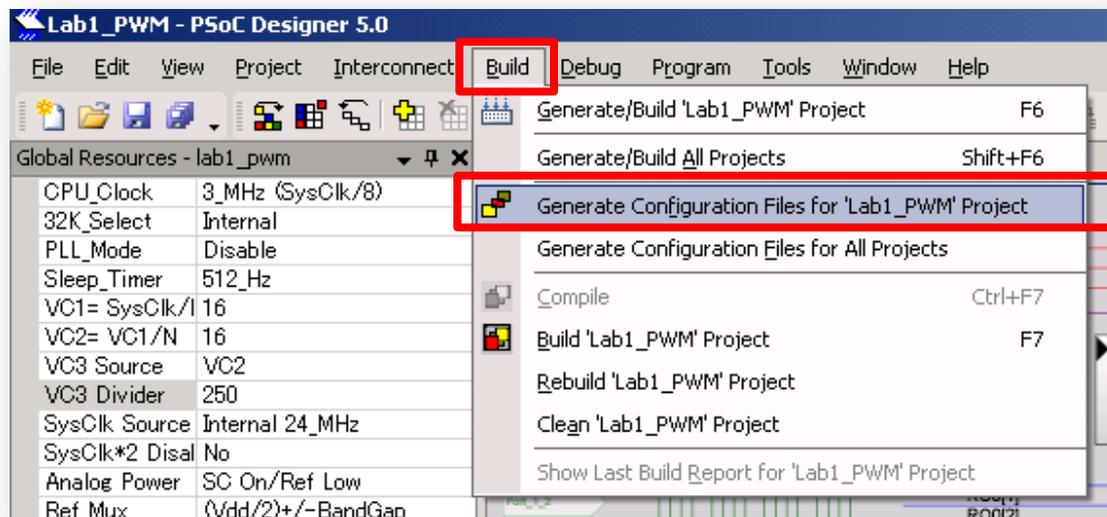


LEDやスイッチなどを接続するときは基板上の回路によって、Pull Up, Pull Downを設定することを覚えておいてください。

# GC(Generate Configuration)

1.Build >  Generate Configuration Files... をクリック

2.使用ユーザーモジュールのハードウェアをコントロールするためのAPIを自動生成します。これによりmain.cからAPIが提供する関数を使用して短いステップでプログラムが書けます。



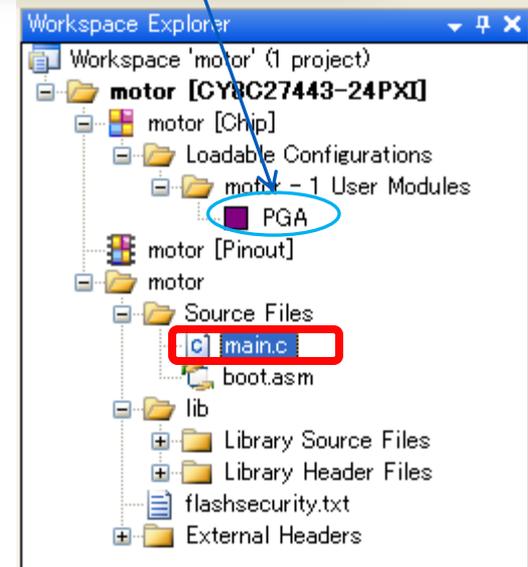
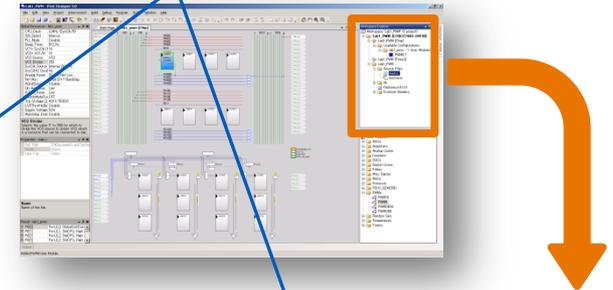
GCにより、設定ファイル、ユーザーモジュールAPIが生成される。配線、設定の変更を加えたならGCをする必要がある。

# ソースコードの記述

View > Workspace Explorer  
1. Source Files 内 main.cをダブルクリック

```
1 // -----  
2 // C main line  
3 // -----  
4  
5 #include <m8c.h> // part specific constants and macros  
6 #include "PSoCAPI.h" // PSoC API definitions for all User Modules  
7  
8  
9 void main()  
10 {  
11     // Insert your main routine code here.  
12 }  
13
```

ハードウェアのモジュール名とCソースのモジュール名を同じにすること。ハードモジュールがPGA\_1ならCソースのモジュール名もPGA\_1\_Start()となる



```
void main()  
{  
    // Insert your main routine code here.  
    PGA_Start(PGA_HIGHPOWER);  
}
```

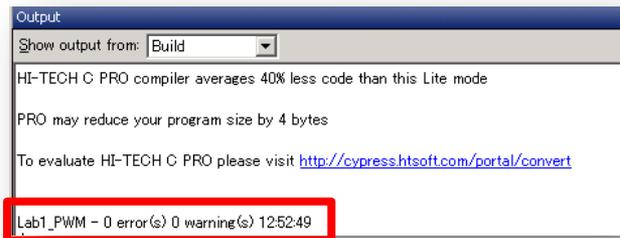
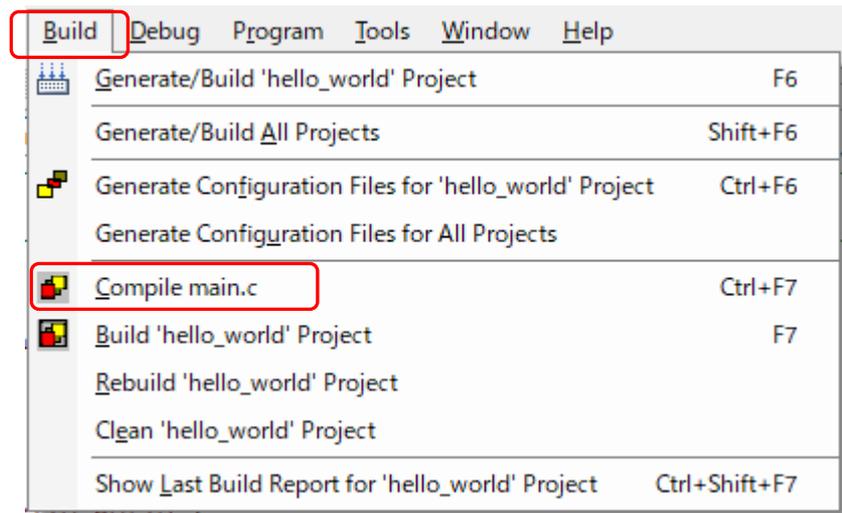
2. 斜字の部分の記述を1行追加

# コンパイル

Build > Compile main.c をクリック

1. Output Window に  
0 error(s) と出れば成功  
エラーが発生したときは、  
!Eまたは!W 表示の行を  
クリックしてCソースを修正

Output Window は表示後すぐ  
閉じるので、左下のOutput Tab  
クリックで開いてください。



# ビルド

1. Build > Build 'プロジェクト名'  
Project をクリック

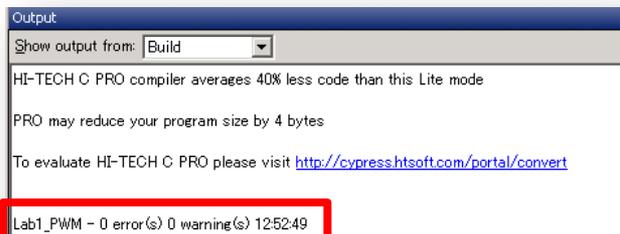
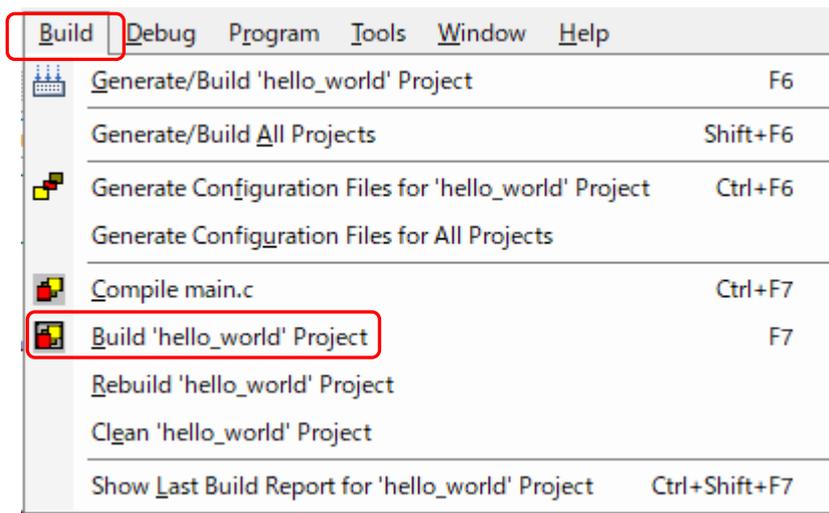
2. コンパイルを行わずに Build を行っても必要なコンパイル処理などが遡って実行されますが、慣れるまでは、

-Generate Config

-Compile

-Build

と順番に処理してください。



Output Window で 0 error(s) と出れば成功  
エラーが発生したときは、  
!E!/W 表示の行をクリックして修正

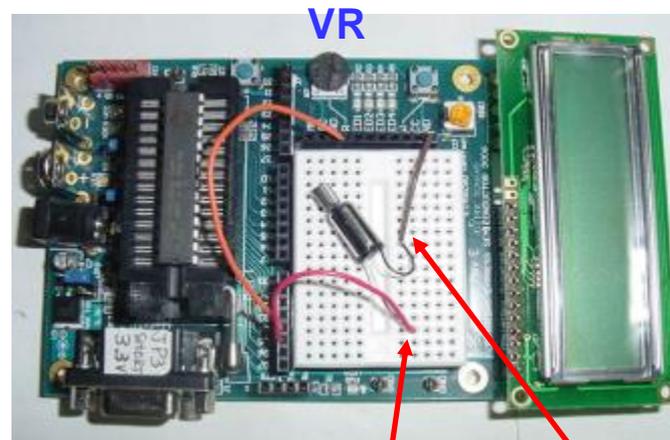
# MiniProgの接続、回路配線

1.P04 と VR をジャンプワイヤーで接続,  
VR(可変抵抗)は左に回しきっておきます。  
この状態でPGAの入力(P04)には,VRから  
印加される可変電圧は0Vです。

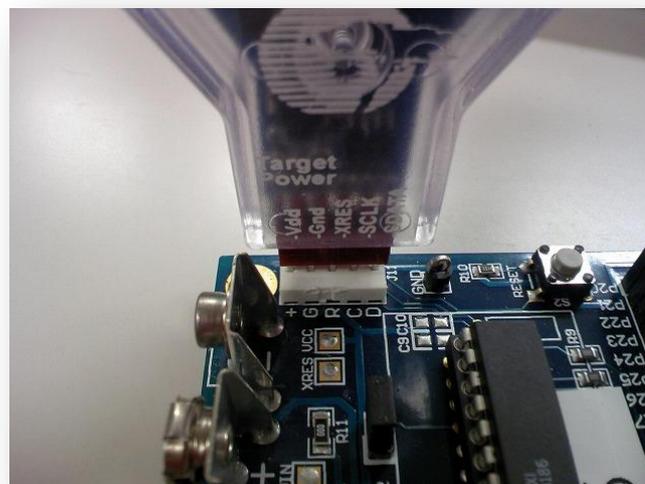
2.VRを右に回していくと少しずつ電圧が上  
がっていきます。VRは5Vを分圧しています  
から右に回しきると5Vが印加されます。

3.回転/振動モーターをP05(PGA出力)と  
GNDの間に接続します。2V,40mA程度でド  
ライブしてください。長時間動作させると焼き  
切れる場合があります。(詳細はPGAのデー  
タシート参照)

4. MiniProg を Eval1基板 に接続  
Vdd と + が一致するように注意

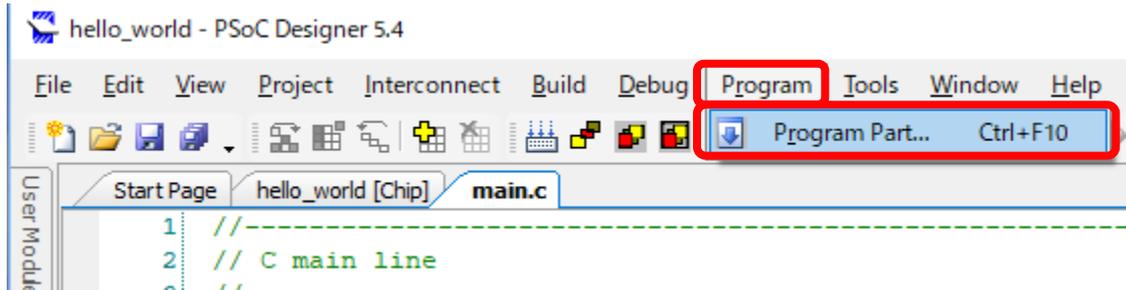


モーターのリード線を同じ穴に一緒に  
ジャンパーで押し込むようにします

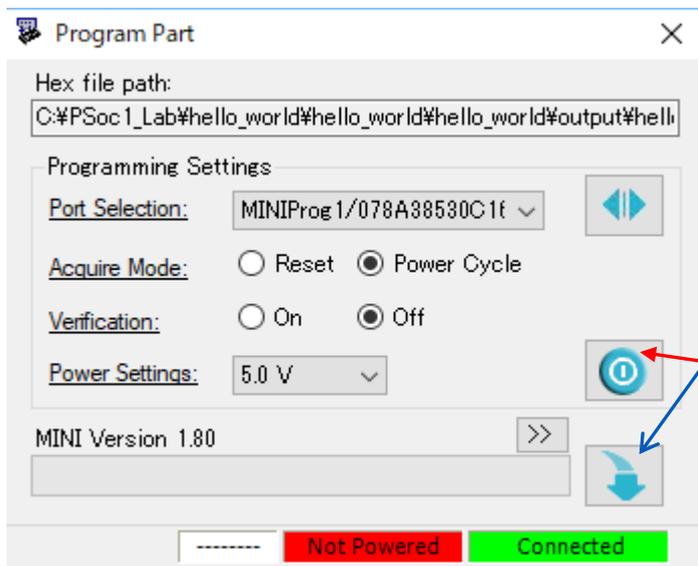


# 書き込み

Program > Program Part をクリック



1.PSoC Designer から Program > Program Part をクリックすると、PSoC Programmer が自動的に起動し 作成されたhex ファイルがロードされる。



2.  をクリックすると書き込み開始

3.Actions を読んで状況を確認

4.  をクリックすると  
MINIProgを通じて電源を供給

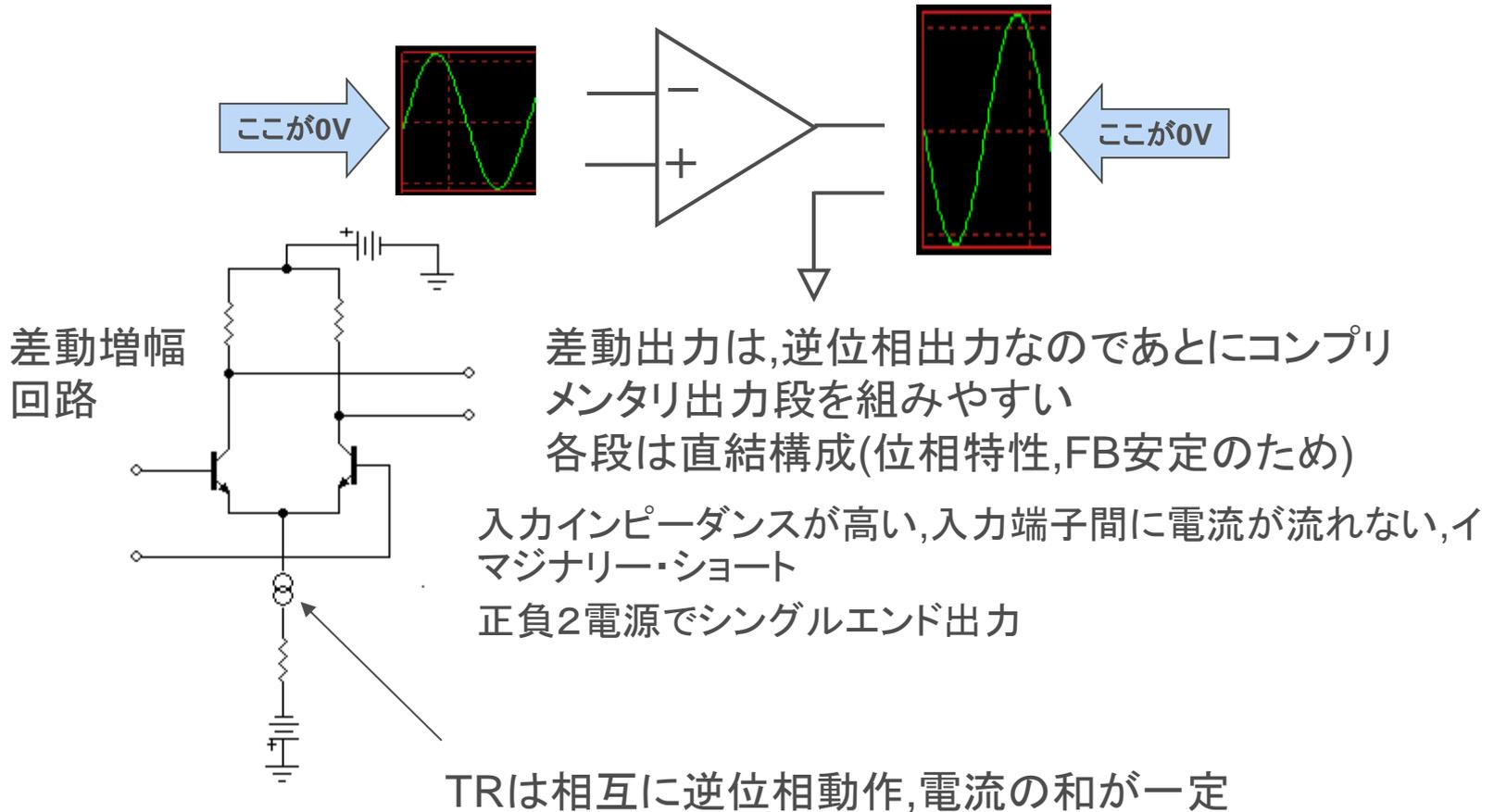
5.ゆっくりVRを右に回してください。モーターが回転を始めます。PGAのゲインを変更して(再GC、ビルドして)動作の変化を確認してみましょう。

この設計では、出力をアナログ電圧でドライブしてます。



# 一般の正負2電源OPアンプ

入力の差を増幅して出力.出力は対グラウンドレベルのシングルエンド

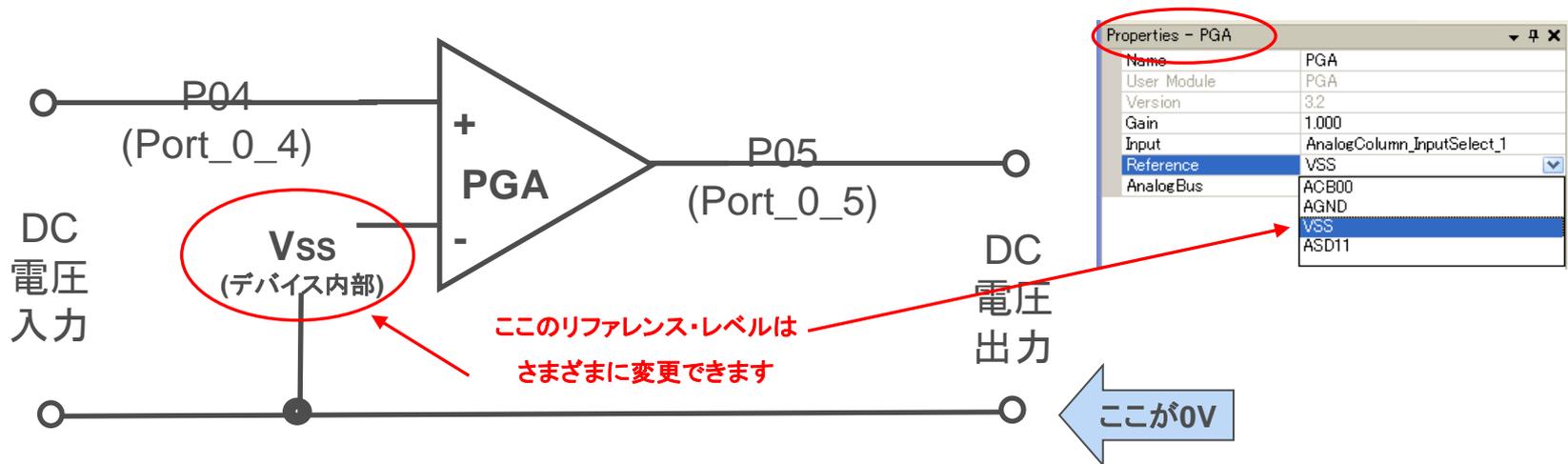




# PSoCの単電源オペアンプ

PSoCのオペアンプは単電源型ですからプラスの信号入力だけを増幅します。PGAのリファレンスはVss(0V)に設定すると入力に加えた0から5Vまでの電圧に対して設定した増幅率(ゲイン)の出力を行いますがこの出力は電源電圧を超えることはありません

マイナス側にも振幅する一般の交流信号をそのまま入力して使用するには、2電源型(プラス、マイナスの電源を供給します)オペアンプを使います



## PSoC のMotor プロジェクトの例

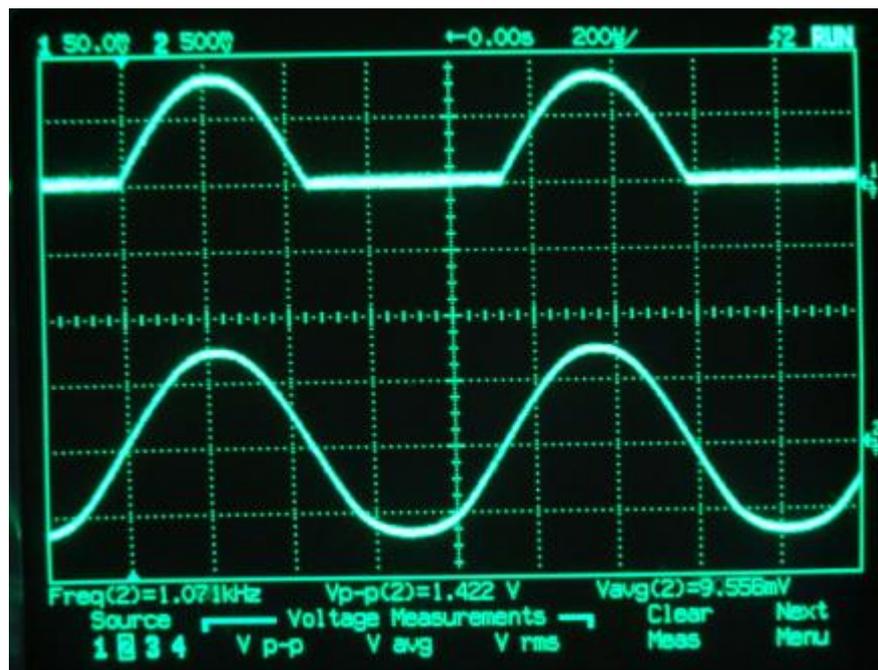


# PSoCへの交流信号入力

PSoCのオペアンプは単電源型ですからプラスの信号入力のみを増幅することができます。一般の正負に振幅する交流信号を入力してもマイナス側の振幅は増幅されません。外部から交流信号を与える場合は、PSoCのアナロググランドレベル(リファレンス・レベル 例: +2.5V)まで電圧をバイアスして上げてやる必要があります。

PGA  
出力

交流  
正弦波入力



ここが0V

ここが0V

単電源非反転オペアンプの入力に負の電圧を印加すると壊れることがありますので、注意してください

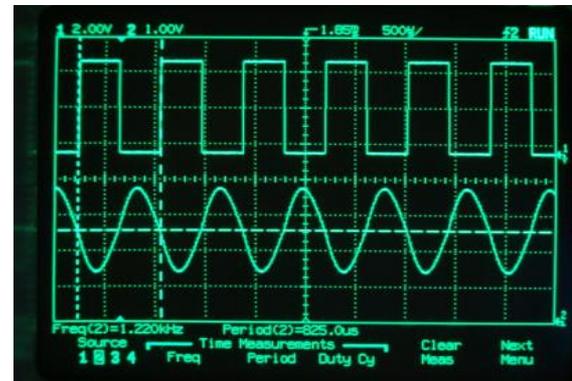
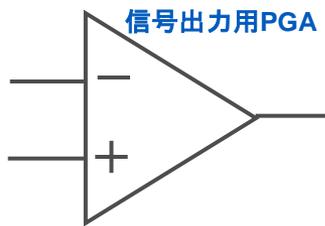


# PSoCからマイナス振幅の出力を出す

信号出力回路が+2.5VのDCバイアスがかかっている場合は別のPGAで+2.5Vのレベル生成をしてやります。

信号出力回路(0から5V振幅)とレベル生成回路(2.5V固定)間の電位を信号出力としてとり出せば±2.5V振幅の交流信号になります(確信犯的?)

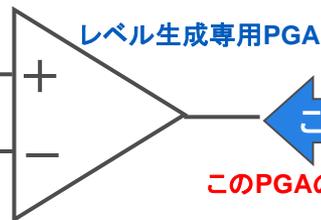
Global Resources - motor	
CPU_Clock	24_MHz (SysClk/1)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	16
VC2= VC1/N	16
VC3 Source	SysClk/1
VC3 Divider	1
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-(Vdd/2)
AGndBypass	(Vdd/2)+/-BandGap
Op-Amp Bias	(Vdd/2)+/-(Vdd/2)
A_Buff_Power	BandGap+/-BandGap
SwitchModePump	(1.6 BandGap)+/-(1.6 BandGap)
Trip Voltage [LVD (SM	(2 BandGap)+/-(BandGap)
LVDThrottleBack	(2 BandGap)+/-(P2[6]
Supply Voltage	P2[4]+/-BandGap
Watchdog Enable	P2[4]+/-(P2[6]



ここが5V

ここが0V

Properties - PGA_3	
Name	PGA_3
User Module	PGA
Version	3.2
Gain	1.000
Input	AGND
Reference	AGND
Analog Bus	ACB02
	AGND
	VSS
	ASD13



このPGAの出力は増幅しないので常にAGND電位(設定はVdd/2ですから2.5V)になります

このPGAは入力もリファレンス・レベルもAGND(アナログ・グランド)に設定しています



# アナログ・リファレンスの設定

EMPTY\_29466 - PSoC Designer 5.0

File Edit View Project Interconnect Build

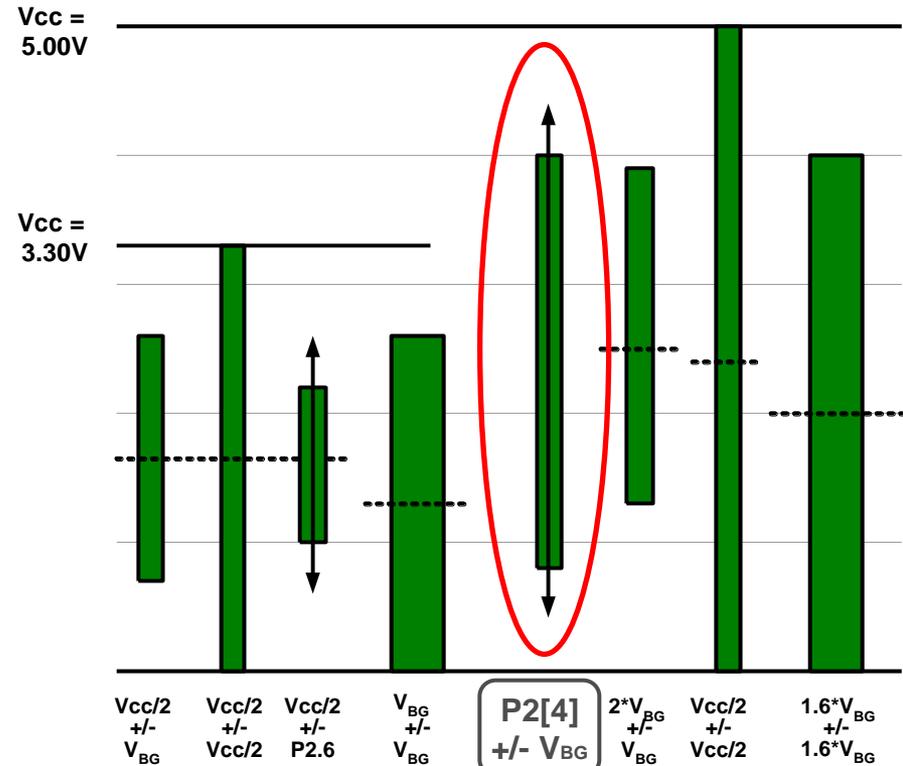
Global Resources - empty\_29466

Power Setting [ Vcc]	5.0V / 24MHz
CPU_Clock	SysClk/8
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	1
VC2= VC1/N	1
VC3 Source	SysClk/1
VC3 Divider	1
SysClk Source	Internal
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	P2[4]+/-BandGap
AGndBypass	(Vdd/2)+/-BandGap
Op-Amp Bias	(Vdd/2)+/-(Vdd/2)
A_Buff_Power	BandGap+/-BandGap
SwitchModePu	(1.6 BandGap)+/-(1.6 BandGap)
Trip Voltage [L	(2 BandGap)+/-BandGap
LVDThrottleBa	(2 BandGap)+/-P2[6]
Watchdog Enab	P2[4]+/-BandGap
	P2[4]+/-P2[6]

**Ref Mux**  
Selects the range and accuracy of various analog references. This sets the analog ground and peak-to-peak voltage limits of the analog system.  
Registers Affected:  
ARF\_CR

Properties - PSoCConfigTBL.asm

アナログのリファレンス電圧を設定する  
ここではP2[4]+/- BandGap としてみる  
BandGap電圧は1.3V ( $1.2V_{BG} + 0.1V_{Gain}$ )  
レジスタはARF\_CR: 0,63h REF[2:0]の3ビット





# アナログ・リファレンスの構造

PSoC オペアンプは単一電源型のためAGNDをV<sub>DD</sub>の中間付近に設定します。AGND電位は各ブロック毎にバッファされるので各ブロック間で若干のオフセットが発生します。RefHi/LoはDACのコンパレータ・スパンを設定します。これは 0,63h ARF\_CRレジスタのREF[2:0]値で設定します

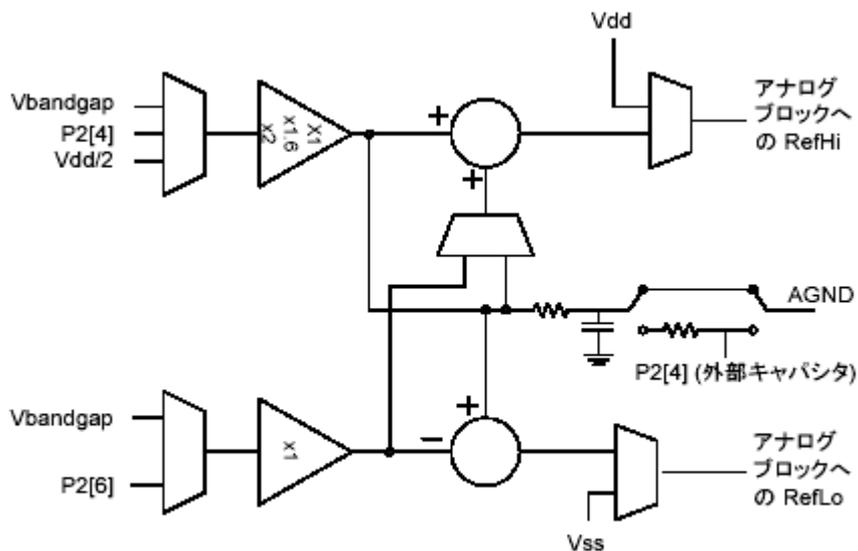


図 21-1. アナログリファレンスコントロール模型

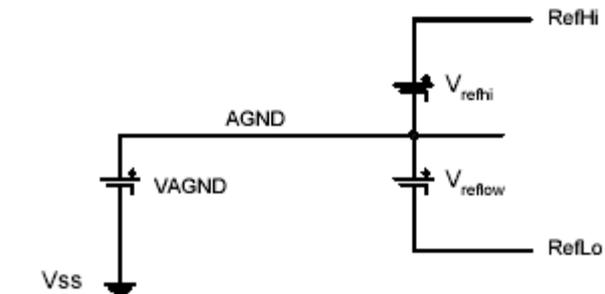


図 21-2. リファレンス構造

表 21-1. アナログリファレンスレジスタ

アドレス	名前	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0	アクセス
0,63h	ARF_CR		HBE	REF[2:0]			PWR[2:0]			RW : 00



## 自由課題

p3\_1200hzの出力信号をmotorの回路に入力して増幅してWSで信号を観測するにはどうしたらよいだろうか？

WGの信号をmotorの回路に入力して波形を歪ませず(マイナス側がつぶれた山形にならないように,サイン波の形で)増幅させるにはどうしたらいいだろうか

できたらWGでスweepさせてみよう.

# Memo

フォローアップURL

<http://mikami.a.la9.jp/meiji/MEIJI.HTM>



担当講師

三上廉司(みかみれんじ)

Renji\_Mikami(at\_mark)nifty.com (Default - Recommended)

mikami(at\_mark)meiji.ac.jp (Alternative)

[http://mikami.a.la9.jp/\\_edu.htm](http://mikami.a.la9.jp/_edu.htm)