#### 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ものはファイ 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



ew Project		<u>? ×</u>
Project types:		
Chip-level S	jvstem-level	
Project	Project	
This is a classic	v4.x PSoC Designer project, selecting and placing user modules.	
Name: 3	Lab3 ADC	
Location:	C¥Documents and Settings¥muda¥デスクトップ	Browse
Workopace name:	Lab3 ADC	
workspace na <u>m</u> e.	Create new Warkesses	
Workspace:		
	<b>Ú Ú Ú</b> Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú Ú	<u>C</u> ancel

#### 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

-Clone p Path:	project	Browse
_	C Use the same target device C Select target device	Cl <u>e</u> ar Path
Select	Target Device	
<u>D</u> evice	:: [CY8C27443-24PVX]	
	Generate 'Main' file using:	
	Generate 'Main' file using:	

	Part Number	Analog Blocks	Digital Blocks	Flash	RAM	IO Count	Supply Voltage	SMP	USB Inte
			`		`				
•	Click here to Remove All Filters	al 💌	al 💌	al 💌	al 💌	al 💌	al 💌	al 💌	all
	CY8C27143-24PXI	12	8	16K	256	6	3.0 to 5.25	YES	N
	CY8C27243-24PVXI	12	8	16K	256	16	3.0 to 5.25	YES	N
	CY8C27243-24SXI	12	8	16K	256	16	3.0 to 5.25	YES	N
	CY8C27443-24PVXI	12	8	16K	256	24	3.0 to 5.25	YES	N
	CY8C27443-245XI	12	8	16K	256	24	3.0 to 5.25	YES	N
	CY8C27543-24AXI	12	8	16K	256	40	3.0 to 5.25	YES	N
	CY8C27643-24PVXI	12	8	16K	256	44	3.0 to 5.25	YES	N
	CY8C27643-24LFXI	12	8	16K	256	44	3.0 to 5.25	YES	N
	CY8C27243-12PVXE	12	8	16K	256	16	4.75 to 5.25	YES	N
	CY8C27443-12PVXE	12	8	16K	256	24	4.75 to 5.25	YES	N
	CY8C27643-12PVXE	12	8	16K	256	44	4.75 to 5.25	YES	N
	CY8C29466-24PXI	12	16	32K	2K	24	3.0 to 5.25	YES	N
	CY8C29466-24PVXI	12	16	32K	2K	24	3.0 to 5.25	YES	N
	CY8C29466-24SXI	12	16	32K	2K	24	3.0 to 5.25	YES	Ν.,
4							5 _		•



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Brouse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。

	🗯 Select	Project Type		2 🛛
	Clone pr <u>P</u> ath:	oject:  Se the same target devi	ice 🔿 <u>S</u> elect target device	<u>B</u> rowse Cl <u>e</u> ar Path
	Select T <u>D</u> evice:	arget Device CY8C27443-24PXI	iew Catalog	
		Generate 'Main' file using:- © © <u>A</u> ssembler	1.使用するデル View Catalogで	バイスを選ぶ。 で一覧表が出る
2.設計に使用す	る言言	吾にCを選ぶ		<u>OK</u> <u>C</u> ancel
	3.決ま	ったらOKをク	フリック	



## ★ PSoC Designerの各ウインドウの開き方



- 必要なウイ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

#### 🇯 motor - PSoC Designer 5.0 ためしにWorkspace Explorer をクリッ Project Interconnect Build Debug File Edit View Program Too 🎦 📂 🚽 🥥 🗶 🕿 📾 🕄 😭 🏭 💾 🗗 🗗 6 クしたままで、画面の右端とか左端ま Global Resources **-** ₽ X **CPU Clock** 3 MHz (SysClk/8) ~ 32K Select で動かしてみる. Internal PLL Mode Disable Sleep\_Timer 512\_Hz VC1= SysClk/N 1 任意の場所でクリックを離すとその場 VC2= VC1/N 1 VC3 Source Sys 🔺 VC3 Divider 所にウインドウが開く(フローティング) SysClk Source نيسي ا SvsClk\*2 Disable No SC ef Low Analog Power Ref Mux (Vdd/2)+/-BandGap AGndBypass Disable On-Amn Rise Low **CPU Clock** Selects the CPU clock speed, from 93.75 KHz to 24 MHz. Derived from the SysCik. This satting attents the Power on Reset level in order to prevent the CPL§ from running outside of its Vdd spec§ication.Registers... Pinout **- - − ×** Workspace Explorer - џ X - 4 b X Port\_0\_0, StdCPU, High 🖁 Analog, DisableInt Port\_0\_1, StdCPU, High 🖁 Analog, DisableInt 🛅 Workspace 'motor' (1 project) Port\_0\_2, StdCPU, High 🖁 Analog, DisableInt 😑 խ motor [CY8C27443-24PXI] en[3] Port\_0\_3, StdCPU, High 🖁 Analog, DisableInt - 💾 motor [Chip] ٠ Port 0\_1 Port 0 4, StdCPU, High 🖁 Analog, DisableInt 🕂 motor [Pinout] Port\_0\_2 Port 0 5, StdCPU, High 🖁 Analog, DisableInt Port 0 3 🖮 🍋 motor Port\_0\_6, StdCPU, High 🎖 Analog, DisableInt Port\_0\_7, StdCPU, High 🖁 Analog, DisableInt Port 0\_4 🗄 🫅 Source Files Port\_1\_0, StdCPU, High 🖁 Analog, DisableInt Port\_0\_5 📄 flashsecurity.txt 1 Port\_1\_1, StdCPU, High 🖁 Analog, DisableInt 🖻 🫅 External Headers Port 1 2, StdCPU, High Z Analog, DisableInt Sys Port 1 3, StdCPU, High Z Analog, DisableInt Port\_1\_0 User Modules Port\_1\_4, StdCPU, High Z Analog, DisableInt - 4 3 Port\_1\_5, StdCPU, High Z Analog, DisableInt 🔬 User Modules Ð Port 1.2 Ð Port 1 6, StdCPU, High Z Analog, DisableInt H. 🗄 🫅 ADCs Port 1.3 F Đ Port\_1\_7, StdCPU, High Z Analog, DisableInt No 🗄 🫅 Amplifiers Port\_1\_4 Port\_2\_0, StdCPU, High Z Analog, DisableInt ☑ P2[0] SC lef Low Port 2 1, StdCPU, High Z Analog, DisableInt 🗄 🫅 Analog Comm I € P2[2] Port 2.2 StdCPUL High 7 Analog DisableInt (Vdd72)+/-Band Port 1.6 😟 🦳 Counters

ウインドウの場所を自由に変更してみる

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 インドウが固定されるちょっとやってみよう

Port\_1\_1

Port 2 0

Port\_2\_2

🖻 🫅 DACs

🖮 🧰 Filters

🖻 🫅 Digital Comm

🛓 🚞 Misc Digital 🗄 🛅 MUXs

Part\_1

#### 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インドウ 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ょう。(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 Wditor

- 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

e Aspect Ratio			
Ctrl+Click ut Shift+Ctrl+Click View	移動	Alt+ドラッグ	
le juality	拡大	Ctrl+クリック Ctrl+ドラッグ	
Background	縮小	Ctrl+shift+クリック Ctrl+shift+ドラッグ	
Ctrl+Click ut Shift+Ctrl+Click View le juality Min Background	移動 拡大 縮小	Alt+ドラッグ Ctrl+クリック Ctrl+ドラッグ Ctrl+shift+クリック Ctrl+shift+ドラッグ	

Preserv
 Show A

Show

Print

Zoom <u>I</u>n Zoom <u>O</u> Original

Pan Moo Higher C <u>F</u>ind...

> Change Refresh <u>H</u>elp... About S

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 モート*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 ウインドウ

#### PGAピンから配線をしてみる

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AnalogColum\_InputSelect\_1をハイラ イトしてください

4.PGAの入力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。



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.ここでは入出カ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.ここではレファ レンスをVSSにと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



#### 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、詳細なコントールが必要 な場合は値を小さくします。

perties - PGA		<b>→</b> ∓ X
Name	PGA	
Jser Module	PGA	
/ersion	3.2	
Gain	1.000	×
nput	2.286	~
Reference	2.000	_
AnalogBus	1.777	
	1.600	
	1.455	
	1.333	
	1.231	
	1.143	
	1.062	
	1.000	
	0.937	
	0.875	
	0.813	
	0.750	~
	8	

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。)

Pinout - motor	<b>→</b> ∓ <b>×</b>
	Port_0_0, StdCPU, High Z Analog, D
	Port_0_1, StdCPU, High Z Analog, D
	Port_0_2, StdCPU, High Z Analog, D
	Port_0_3, StdCPU, High Z Analog, D
	Port_0_4, StdCPU, High Z Analog, D
🗆 PO[5]	AnalogOutBuf_1, AnalogOutBuf_1, S
Name	AnalogOutBuf_1
Port	P0[5]
Select	AnalogOutBuf_1
Drive	Strong 💌
Interrupt	High Z
⊞ P0[6] <	High Z Analog
	Open Drain High
	Open Drain Low
	Pull Down
	Pull Up
	Strong
	Strong Slow
	Port 1 5, StdCPU, High Z Analog, D

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 > 子 enerate 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。

Lab1_PWM - PSoC Designer 5.0							
<u>File Edit View Project Interc</u>	onnect <u>B</u> uild	<u>D</u> ebug P <u>r</u> ogram	<u>T</u> ools	<u>W</u> indow	Help	_	
🎦 🗀 🖌 🖉 🗸 🖺 🛣	🎦 🏦 🛗	Generate/Build 'Lab1	_PWM' Pro	ject	F6	1	
Global Resources - lab1_pwm	<b>→</b> ₽ <b>×</b>	Generate/Build <u>A</u> ll Pro	jects		Shift+F6		
CPU_Clock 3_MHz (SysClk/8) 32K_Select Internal	-	Generate Configurati	on Files fo	r 'Lab1_PW	/M' Project		
PLL_Mode Disable		Generate Configurati	on <u>F</u> iles fo	or All Project	ts		
Sleep_Timer 512_Hz		1_			cul : ca		
VC1= SysClk/I 16	. Bet	Compile			Ctrl+F7	- 64	
VC2= VC1/N 16	<b>5</b>	Build 'Lab1_PWM' Pro	ject		F7	- 151	
VC3 Source VC2		Debuild 'Labt, DWM' D	waiach				
VC3 Divider 250			roject			- 64	
SysClk Source Internal 24_MHz		Clean 'Lab1_PWM' Pr	oject			- 84	
SysClk*2 Disal No						-11	
Analog Power SC On/Ref Low		Show Last Build <u>R</u> epo	rt for 'Lab	1_PWM Pro	oject		
Ref Mux (Vdd/2)+/-BandGa	P	×			R00[2]	_	



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。



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。

Output	
Show output from: Build	
HI-TECH C PRO compiler averages 40% less	code than this Lite mode
PRO may reduce your program size by 4 byte:	3
To evaluate HI-TECH C PRO please visit <u>http</u>	://cypress.htsoft.com/portal/convert
Lab1_PWM - 0 error(s) 0 warning(s) 12:52:49	

I	<u>B</u> uild	1	<u>D</u> ebug	P <u>r</u> ogram	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp			
i	<u></u>	Generate/Build 'hello_world' Project F6								
		Generate/Build <u>A</u> ll Projects Shift+F6								
	F	Generate Configuration Files for 'hello_world' Project Ctrl+F6								
		Generate Configuration Files for All Projects								
	P	<u>C</u> o	mpile ma	iin.c				Ctrl+F7		
F		<u>B</u> ui	ild 'hello	_world' Proj	ect			F7		
		<u>R</u> ebuild 'hello_world' Project								
		Cl <u>e</u>	an 'hello	_world' Pro	ject					
		Sho	ow <u>L</u> ast B	Build Report	for 'hell	o_world' Pr	oject	Ctrl+Shift+F7		
-	10-	THU:								

#### ビルド

#### 1.Build > Build 'プロジュクト名' 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<u>左に回しきっておきます。</u> 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





#### Program > Program Part をクリック

🖕 hello\_world - PSoC Designer 5.4 1.PSoC Designer から Edit View Project Interconnect Debug Program Tools Window Help <u>B</u>uild File Program > Program Part Program Part... Ctrl+F10 SIN SI **-**2: (1) 6. をクリックすると、PSoC User Modu hello\_world [Chip] Start Page main.c Programmer が自動的に 1 記 動 し 作 成 さ れ た hex C main line 2 ファイルがロードされる。 🐉 Program Part Х Hex file path: をクリックすると書き込み開始 C:¥PSoc1\_Lab¥hello\_world¥hello\_world¥hello\_world¥output¥helli Programming Settings  $\blacksquare$ MINIProg 1/078A38530C1t ~ Port Selection: 3.Actions を読んで状況を確認 O Reset 
 Power Cycle Acquire Mode: Off O 0n Verification:  $\bigcirc$ Power Settings: 5.0 V MINIProgを通じて電源を供給 >>MINI Version 1.80 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 Connected Not Powered 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。



#### ー般の正負2電源OPアンプ

入力の差を増幅して出力.出力は対グランドレベルのシングルエンド





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.



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.

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

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?)







#### アナログ・リファレンスの設定

👺 ЕМРТҮ_294	66	- PSoC	Designer 5	5.0				
<u>E</u> ile <u>E</u> dit <u>V</u> i	ew	<u>P</u> roject	Interconnect	<u>B</u> uild				
i 🐑 🖂 🗐 🗿		S B 1	⊊   <b>4</b> a %a					
Global Besources	Global Resources - empty_29466 🛛 🗸 🕂 🗙							
Global Resources Power Setting CPU_Clock 32K_Select PLL_Mode Sleep_Timer VC1= SysClk/ VC2= VC1/N VC3 Source VC3 Divider SysClk Source SysClk Source SysClk Source SysClk Source SysClk Source SysClk Source SysClk Source SysClk Source CC3 Divider SysClk Source SysClk Sou	- er [ Vc+ [ Vc+ N N ble (Vdc Ban (1.6 (2 E (2 E	mpty_29466 5.0V / 241 SysClk/8 Internal Disable 512_Hz 1 1 SysClk/1 1 Internal No SC On/Re P2[4]+/-Ea d/2)+/-Ba d/2)+/-Qa dGap+/-Ba BandGap)+/ BandGap)+/	andGap HHz andGap H/2) andGap +/-(1.6 Band( -BandGap +/-(26]	↓ ₽ ×				
Watchdog Enab	P2[4	4]+/-Band( 4]+/-D2[6]	зар					
	126	+J+/-F2[0]						
<b>Ref Mux</b> 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 - PSoCConfigTBLasm - 7 X								

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 (1.2VBG+0.1VGain) *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Vod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

#### 表 21-1. アナログリファレンスレジスタ

アドレス	名前	ピット 7	ピット 6	ピット 5	ビット 4	ピット 3	ピット 2	ピット 1	ピット 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でスイープさせてみよう.

#### 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