

プロジェクトの複製,GPIO,ポーリングと割込み

PSoC 3/5 GPIO HW Polling and Interrupt PSoC Experiment Lab

PSoC 3/5 GPIO Experiment Course Material 6 V2.02 October 15th. 2012 GPIO_35.PPT (65 Slides)



an and the state of the state o 4 mill

6



Lab GPIO_35 GPIOの使い方 プロジェクトの別名セーブ

演習ファイルと使用文字などの注意事項

- •各演習プロジェクトを置くための演習用ディレクトリを作成して ください。場所と名前は、演習中に指示します。(デフォルトは、 C:¥PSoC5_Lab)
- ・設計例(正解の例)プロジェクトは、演習で指示する場所の
 PSoC_Lab_MasterXXXにあります。
- ・使用するファイルのあるディレクトリのパス名に英数字以外の 文字が含まれていないことを確認してください。(不可例参照)
- •ファイル名、プログラムのソースには、英数字と構文で許された記号以外は使用しないでください。(不可例参照)
- •WorkspaceとProjectを1対1に対応して作成(次スライド)

•不可例:全角文字、日本語文字、半角カナ

演習プロジェクト作成のグランドルール

•設計は、プロジェクトをひとつの単位としますが、大規模設計では複数のプロジェクトをまとめて、ひとつのWorkspace内で管理することができます。

- •複数のプロジェクトを一つのWorkspaceに追加した場合、同じ名前のファイルを開いたときに、どのプロジェクトに含まれるものなのかを毎回確認する必要があります。例えば main.c などです。
- ワークショップ中は、この混乱をさけるために、一つのWorkspaceには一つのプロジェクトしか配置しないように注意して下さい。

•この方法は三つあります。

- •1.新プロジェクトを作成する前に、一度現在のPSoC Creatorを閉じる
- •2.プロジェクト作成する前に、FileメニューからClose Workspaceを実行して現在開いているWorkspaceを閉じる

•3.新規プロジェクトを作成する時、Advancedオプション内のWorkspaceを Create New Workspaceに設定する。この方法については次のページに説 明があります。 プロジェクト新規作成時の注意事項 •プロジェクトを既に開いた状態で新規プロジェクトを作成した場合、デフォルトでは、 現在開いているWorkspaceに追加されます。ウィザード内のWorkspaceをCurrent New Workspaceに切り替えて下さい。

New Project		Add to	o Current Workspaceを選択 入
Design Other PSoC Creator Inst PSoC Creator Inst DEmpty PSoC 3 Des A Creates a PSoC 3, 8 d Name: fue Location D:¥M Auvanced Workspace: Workspace Name:	talled Templates tien Empty PSoC 5 Design Advancedオプションを オープンする It, design project. YCypress#PSoC_Creator_Projects#hoge Add to Current Workspace Add to Current Workspace Create New Workspace	すると 現在開 新規プ Create と、 新規W その中 ます。	いているWorkspace内に ロジェクトが作成されます。 New Workspaceを選択する Vorkspaceが作成され、 に新規プロジェクトが作成され
Device: Sheet Template: Application Type	CY8C3866AXI-040 - (Last Used PSoC (A4 (11.7" x 8.3") Normal OK	Advanced Workspace: Workspace Name:	Add to Current Workspace Add to Current Workspace Oreate New Workspace

GPIO_35ラボの目的

- GPIOのハードウェア単独制御(MPU不使用) (GPIO_35D)
- GPIOのポーリング制御(MPU使用) (GPIO_35P)
- ・GPIOの割込み制御(MPU使用) (GPIO_351)
- プロジェクトの別名でのセーブ法 (GPIO_35D -> GPIO_35P変更)
- プロジェクトの内容
 - GPIO_35D スイッチを押すとLEDが消灯 離すと点灯
 - GPIO_35P/GPIO_35I スイッチを押すとLEDが点灯 離すと消灯



全体の処理フローの解説

Step 1: PSoC Creator Softwareの起動

Step 2: 新しいプロジェクトの作成

Step 3: コンポーネントの追加

Step 4: コンポーネントのコンフィギュレーション

Step 5: コンポーネントの接続

Step 6:共有リソースの設定

Step 7:ソースコードの記述

Step 8: Build & Debug

Step 1 PSoC Creator Softwareの起動

PSoC Creator 1 0



D Srites U.Warrangs O Notes

各ウィンドウの解説





ウィンドウ番号	機能
1,Main Window	回路の表記、ピンアサイン、コードの記述などを行います。Work space exploreから開いたものが、メイン ウィンドウに表示されます。立ち上がった状態(画面の状態)ではSchimatic Windowが表示され回路の表 記Windowが表示されています。機能ごとに TAB形式で開かれ、TABが二つ以上ある場合にはTABを 右クリックしてNew XXX windowを選択することで縦、横に分割することもできます。
2,Work Space Explore	SourceTAB にはプロジェクトに必要なファイルが表示されます。また、ここから既存のファイルを追加する なども可能です。Componentsには自分で作成したモジュールなどのファイルが表示されます。Resultに は各種ログファイルが表示されます。
3,Component Catalog	標準ではCypress社の用意したモジュールが表示されます。コンポーネントのデータシートなども、ここから 参照できます。
4,Output	現在進行中のLogファイルがここに表示されます。 コンパイル結果なども同様に、 Output Windowに表示 されます。

Step 2 新しいプロジェクトの作成

File->New->Projectを選択してください
Project名は、GPIO_35D等にしてください
Locationは、演習で指示しますが、デフォルトは、 C:¥PSoC5_Labとします。

E H	ELLO_WORLD_35 - PSoC Creator	2.0 [C:¥PSoC5_Lab¥HELLO_WC
Eile	e Edit <u>V</u> iew <u>D</u> ebu <mark>g P</mark> roject	<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp
È	New	Project
	Open •	🞦 <u>F</u> ile
	Add +	Start Page TopDesign.cysch mai
	<u>C</u> lose Ctrl+F4	10 *
ļΞ,	Close <u>W</u> orkspace	12 #include <device.h< td=""></device.h<>

Step 2.	1 新規プロジェクトの作成(続き)
New Project	🗾 2.Empty PSoC5 Design を選択(ハイライト化)
Design Ott PSoC Creator I Pa Empty PSoC 3	ner ↓↓ Installed Templates Design Palempty PSoC 5 Jesign 2 Normoの押に CDIO 25D
1.+ マークを <u>(</u>	フリック と名前をつける(名前は任意)
Creates a PSoC 5, Name: G Location: C	^{32 bit, design project} PIO_35D *PSoC5_Lab (演習で指示の全英文字パスディレクトリ デフォルトは、C:XPSoC5_Lab
Workspace Name	Create New Workspace
Device: Sheet Template:	CY8C5588AXI-060ES1 - (Last Used PSoC5 Device)
Application Type	Normal ・ 6.0Kをクリック
5 デバイフ 共	
<u>5. テハイス</u> 指	<u> (正または安史の場合は、ここから選択(次ペーシ)</u>

© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop



エディタ画面の切り替え

Workspace Explorer



© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop



© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop

Step 4 コンポーネントのコンフィギュレーション

📴 Lab1_1 - PSoC Creator 1.0 [D:¥¥PSoC_Creator_Projects¥Lab1_1¥Lab1_1.cydsn¥TopDesign¥TopDesign.cysch]					
<u>File Edit View Debug Project Build Tools Window H</u> elp					
221% 🖓 🔂 😂 属 🥔 🚳 🔍 🔺 🖻 🖎 🗙 🧐 🔍 🖕 221%	• 🔍 🤤 🖕				
Microsoft Sans Serif - 10 - B I U 📰 🚍 🛓 - 🖉 -	A B. B. B. B. A. A. A.	ू 🖾 • 🚵 🤌 👹 🔆 🖕			
Workspace Explorer + 4 X Start Page *TopDesign.cysch	• 4 ▷	x Component Catalog (84 → 쿠 X			
		^ 🖉 🖬 🖪 🛈 🗹			
Workspace 'Lab1_1' (1 Proje "]		#3 # <u>3</u>			
TopDesign.cysch		Cypress 4 D			
Pin 1 D	🔏 Cu <u>t</u> Ctrl+X	Cypress Component Catalog			
	Copy Ctrl+C	CapSense			
🖻 🧰 Source Files 🛛 💆 🖳	2 Paste Ctrl+V	🕀 🐼 Communications			
ि mainc हैं T	X Delete Del	E Solardi			
	Zoom +	 			
	Shape +	Analog Pin [v1.50]			
回路図上のPin 1を	Configure	Digital Input Pin [v1.50			
$ \pm h \parallel w h \parallel Configura 东 撄 圯 $	🕒 Open Datasheet	× m +			
ロクリックUCUIIIguleを選択	Find Example Project	Component Preview			
し設定ウインドウを開く	Open Component Web Page	Data Sheet <u>cy pins v1 50pdf</u>			
	Launch Tumer	Pins configured for digital input are used to route external digital			
 III → ♥ Output Notice List 	Output Notice List				
Ready		0 Errors 0 Warnings 1 Notes			

Step 4.1 コンポーネントの設定1

Configure 'cy_pins'	8 ×	
Configure 'cy_pins' Name: Pins Mapping Built-in Number of Pins: 1 IAll Pins] Pin_1_0 I	Image: Connection	
Datasheet	Journal 各項目の意味 Name: コンポーネントインスタンスの名称 Analog: アナログ入出力用 Digital Input: デジタル入力用 HW Connection: 内部のデジタルバスへ接続 Digital Output: デジタル出力用 HW Connection: 内部のデジタルバスへ接続 Output Enable: IOをトライステートIOに設定 Bidirectional: IOを双方向に 内部デジタルバスへ接続	

Step 4.2 コンポーネントの設定2

2

Configure 'cy_pins'

Pins Mapping Number of Pins: 1 [All Pins]	Built-in	Name を "SW" に変更 コンポーネント毎のAPIはこの名称をもとに 生成されるので、わかりやすい名前を付けること
	 Analog Digital Input HW Connection Digital Output HW Connection Output Enable Bidirectional Show Annotation Termina 	

Step 4.3 コンポーネントの設定3

Configure 'cy_pins' Name: SW Pins Mapping Number of Pins: 1	Built-in	Pin Resis	s > General の Drive Modeを stive Pull Downに設定して下さい。
[All Pins] L 🛛 SW_0	Type General Input Output Drive Mode	tial State: Configure 'cy_pins'	小蔵している アップ/ダウン抵抗は約5k です。
Datasheet		Pins Mapping Number of Pins: 1 [All Pins] L SW_0	Built-in $4 b$ Type General Input Output Drive Mode Resistive Pull Down Strong Drive Open Drain, Drives High
各項目の解説 Drive Mode : IC 詳細についてに Initial State : PC Drive Modeを Pull Downの境 Pull Upの場合	りの入出力モード はTRM 22.I/O Systemを参照 OR解除後のIO出力の初期状 Pull Up/Downで使用する場合 合は"0" は"1" に設定する必要があり	Datasheet	Open Drain, Drives Low Resistive Pull Up Resistive Pull Up/Down High Impedance Digital High Impedance Analog DE が終わったらOKをクリック

Step 4.4 コンポーネントの追加





Step 5.1 コンポーネントの設定



共有リソースエディタ(.cydwr画面から選択)

•PSoC Creatorでは、デバイスの固有リソースの割り当てを行うために共有リソース エディタが用意されています。.cydwr画面を表示して画面下のタブを選択することで、 表示する画面の切り替えを行います。



Step 6 共有リソースの設定

GPIO_35D - PSoC Creator 2.0	<pre>[C:#PSoC5_Lab#GPI0_35D#GPI0_35D.cydsn#C</pre>	SPIO_35D.cydwr]			×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>D</u> ebug <u>P</u> r	oject <u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp				
1 a 🖻 a 🖉 a 🖉 a 🕼	🌡 🖻 🖪 🗙 년 이 연 📜 🔜 - 📮 i 20% 🛛 - 🤮	🔍 🧠 🍃 j Debug			
🕮 • 适 🖉 👹 🐥 🖕					
Workspace Explorer (1 7 🗙	Start Page *TopDesign.cysch GPI0_35D.cyd	wr		• 4	Þ 🗙
🖓 🔏 🞯 Workspace 'GPIO_35D' (1 Proj		Alias Name	Pin		Lock
🗄 🔂 Project 'GPIO_35D' [CY 💡		SW		- 👻	
- A TopDesign.cysch		LED		-	
Header Files	Cricial And Annual I				
L-M deviceh					
E Cource Files					
	GPIO_35D.cydwr をタフルクリッ	クして開いてト	-20		4
S S			ecuves 🚬 🔤 riash Sec	urity	4 ۵
ज	Output				, 4 x
	Show output from: All 🔹 🔻				
	Log file for this session is locat Note: This project targets evaluat Note: The PSoC 5 FirstTouch Kit (C	ed at: C:\Users\ ion silicon (ES) Y8CKIT-014) uses	Renji Mikami\A that is not a the XTAL pins	ppData\ vailabl as GPI	Lo ^ e E O +
< ► ▼	Output Notice List				
Ready			0 Errors 0 Warning	s 0 Not	es .::

Step 6.1 IOのアサイン



Step 7 ソースコードの記述



Step 8 Build : Project のビルド



Step 8.1 ビルド結果の確認



"---- Buid Suceeded: **/**/**** **:**:** ---"と表示されたらBuild成功です

Step 8.2 デバイスへの書き込み動作の確認



<u>Step 8.3</u> デバイスへの書き込み

reau	br 1.0 [D:#MyCypress#PSoC_Creator_Project	
Del	bug <u>P</u> roject <u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp <u>W</u> indows	USBケーブルが 基板とPCに繋がっていることを確認して から、
0010	Program Ctrl+F5	Debug > Programを実行して下さい
※ ≩	Select Debug Target F5 Execute Code F5 Attach to Running Target F5	正常に書き込みが完了しますと、 Output画面に下記のメッセージが表示されます
1	Toggle Breakpoint F9 New Breakpoint • Delete All Breakpoints Ctrl+Shift+F9	Device ************ was successfuuly programmed
0	Enable All Breakpoints Output	• 4 X
	Show output from: All	• 🖹
	Device ID Check Erasing Programming of Flash Protecting Verify Checksum Device 'PSoC3 CY8C386	Starting
	ಿದವಳಿದರು. ಇದರುವರು ಬಿಡಿದರೆದರು. ಬ	ವರ್ಷದಿಂದ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಈ ಕಾರಣ ಕಾರಣ ಗ್ರಾಮಕ್ಕೆ ಸೇರಿದ್ದ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಮಾಡಿ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಈ ಸಂಗಾರ ಈ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಸಂಗಾರ ಈ ಸಂಗಾರ ಈ ಸಂಗಾರ
	* [III. •



Step 8.5 動作確認



プロジェクト・ワークスペースの複製 1 ^{プロジェクト}GPIO_35Dを選択、File>Save XXXX As

GPIO_35D - PSoC Creator 2.0 [C:¥PSoC5_Lab¥G	PIO_35D¥GPIO	×		
<u>File Edit View D</u> ebug <u>P</u> roject <u>B</u> uild <u>T</u> ools	<u>W</u> indow <u>H</u> elp	🗄 G	PIO_35D - PSoC Creator 2.0 [C:	¥PSoC5_Lab¥GPI
10000000000000000000000000000000000000	🖞 🔝 🗸 🍦 Debug	Eile	<u>E</u> dit <u>V</u> iew <u>D</u> ebug <u>P</u> roject	<u>B</u> uild <u>T</u> ools
圖•ǎ彡噹 ♥★。 宇宇三일□。	11-0		New •	BX 9 C
Workspace Explorer 🗸 🗸 🗶	O_35D.cydwr main.c		Open •	330,
· · · · · · · · · · · · · · · · · · ·	1 /*		Add	→ ∓ ×
B Project 'GPIO_35D' [CY8C5588AXI-060ES1	3 * Copyr:		Close Ctrl+F4	
TopDesigneysch	4 * All R 5 * UNPUBI	, đaj	Close <u>W</u> orkspace	1-060EST
E C Header Files		Я	Save Ctrl+S	ource
deviceh	Output	-	Save GPIO_35D As	
	w output from: All		Save All Ctrl+Shift+S	ompor
続いてGPIO 35D0	e: The PSoC 5		Create Workspace Bundle	Inents
- プロジェクトを 復制リ			Page Setup	Re
ノロシェクトを複表し	Notice List	4	Print Ctrl+P	sults
GPIO_35Pの	0 Warnings 0 f	Q	Print Pre <u>v</u> iew	
🗏 プロジェクトを作りま	d		Recent Files	E
	·)		Recent Projects	
			E <u>x</u> it	

© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop

プロジェクト・ワークスペースの複製2 プロジェクトをGPIO_35Pという名前でセーブしてください



File > Close Workspace でワークスペースを閉じてください。



BSoC Creator 2.0		
File Edit View Debug Projec	t <u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp	新しい
·西·古乡信 學系 _		, GPIO_35P
Workspace Explorer - 7 ×	Start Page 🔹 4 🗙	プロジェクトが
4 a		
s	PSoC [®] Creator	できていることを
urce		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
	Recent Projects	
	HELLO Jywrk	
ファイル	THELLO_WORLD_35.cywrk	
が	Open Existing Project	
たいこと	Output 🗸 🕈 🗙	
	Show output from: All	
を確認	Note: This project tar * Note: The PSoC 5 First	
	4 III)	
▼	Output Notice List	
Ready	0 Errors 0 Warnings 0 Notes .:	

File >Open						
Eile	<u>E</u> dit <u>V</u> iew <u>D</u> ebug <u>P</u> roject	<u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> e				
	New •	BX 90 B.	>Project/workspace C			
	Open	Project/Workspace	複製されたGPIO 35P			
	Add +	🚰 <u>F</u> ile Ctrl+0	プロジェクトを問きます			
	Close Ctrl+F4	Example Project				
φî	Close Workspace	PSoC				
9	Save Ctrl+S	2				
	Save As					
1	Save All Ctrl+Shift+S	S S S S S S S S S S S S S S S S S S S	C5_Lat GPIO_35P.cydsn GPIO_35P.cydsnの検索 。			
	Create Workspace Bundle	整理 ▼ 新しいフォ	オルダー 📰 🕶 🚺 🔞			
	Page Setup	🔋 ドキュメン ^	名前 更新日時 種類			
6	Print Ctrl+P	■ ピクチャ	→ CortexM3 2012/01/07 13:38 ファイルフォル			
٩	Print Pre <u>v</u> iew	ビデオ	Generated_Source 2012/01/07 13:38 ファイルフォル			
	Recent Files	<i>∂</i> ≋⊐-ジッ	KopDesign 2012/01/07 13:38 ファイルフォル CPTO 255 cmmi			
	Recent Projects		GPI0_35HCypr] 1012/01/07 13:38 Offkribwn Proj			
	Exit					
-	8271	7	7ァイル名(N): GPIO_35P.cypr All Project Files (*.cyprj;*.c) -			
			開く(0) マ キャンセル			

© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop

<u>F</u> ile <u>E</u> dit <u>V</u> ie <u>H</u> elp	w <u>D</u> ebug <u>P</u>	roject	<u>B</u> uild	Tools	<u>W</u> indow		
Workspace Explo	rer	•	Ψ×	Start Pa	a _t ∓∢⊁×		
■ *Workspace 'GP → ▶ Project GH → ₩ TopDesig → ♥ GPIO_351	10_35P-000'(1 Pi 10_35P [CY8C Incysch ⁹ cydwr	ojects) 5588AXI	Source	PSo	C [®] Cre		
⊨ 👝 Header Files └──ोì deviceh ⊨ — ☐ Source Files └──ⓒ main.c			Components	Output Show o	+ + × utput from:		
⊟- Cenerate ⊟- Con PSoC ⊟- Con Con	d_Source 5 ◀ y_boot) cm3gcc.ld		Results	Note Note	This +		
<	Cm3RealView	scat 🕴	▼ △ ▽	Output	Notice List		
20 C		22	244	aminaa	O Mataz		

GPIO_35P-0000 / という名前のワーク スペースが自動生成 されます

> その中にGPIO_35P 名のプロジェクトが複製 されます

デバイスをPSoC5から3に変える場合では、 Generated_Source内のサブディレクトリが PSoC5のままになります。 PSoC3への変更は、Generate Configを 実行してください

GPIO_35Pプロジェクトの目的

- ハードウェア直結からMPU制御に変更(GPIO_35P)
- GPIO_35Pは、ポーリング方式
- プロジェクトの内容
 - スイッチを押すとLEDが点灯
 - スイッチを離すとLEDが消灯
- 複製したGPIO_35Pプロジェクトをロードして開始します



Step 1 回路図エディタの操作1



Step 2 回路図エディタの操作2



配線の削除ではエラーが表示されます

🖽 GPIO_35P-000 - PSoC Creator 2.0. [C:¥PSoC5_Lab¥GPIO_35P.cydsn¥TopDesign¥TopDesign.cysch]									
<u>File Edit View Debug Project Build Tools Window H</u> elp									
[2] 한 금 😝 및 의 은 👗 🗈 🗈 🗙 🥑 🔍 🔍 - 🗍 198% - 🔍 🔍 Debug -									
Start Page *TopDesign.cysch									
Project 'GP 👦 💑 SW 6[1]									
TopDesie E	ation Concept Cypress								
	- Cypress Component Catalog								
L device g Page 1	↓ D GapSense								
Source F S Notice List	→ n ★ Communications								
Generate	aw Dotails								
PSoC 7	Error Looption								
E C S State	B Analog Pin [v1.60]								
terminal "y_0".	sysch Signal: Net_4,								
	Digital Output Pin [v1								
Shape_1.2 TopDesigno	cysch Custom A								
when it is visible.	sysch Signal: Net_4 Component Preview								
Signal: Net_4 TopDesign.c	Datasheet								
Shape_1.2 TopDesign.c	sysch								
 III → ✓ Output Notice List 									
Ready 2 Errors 0 Warnings 0 Notes									

© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop

Step 3 SWコンポーネントの設定変更





設定変更でエラーが消えます

GPIO_35P-000 - PSoC Creator 2.0 [C:¥PSoC5_Lab¥GPIO_35P.cydsn¥TopDesign¥TopDesign.cysch]									
<u>File Edit View Debug Project Build Tools Window H</u> elp									
🚼 🖞 🔂 🗃 🕼 🕼 🕼 🛝 🖄 🛝 🗶 🌱 🔍 🖕 198% 🕞 🍳 🖕 Debug 🗸 🖕									
Ш • Ă 参 嗜 🦉 条 , Microsoft Sans Serif • 10 • B I 및 📄 書 🗒 🖄 💁 🐁 42. 43 🖨 44 🖓 囝 📜									
Workspace Ex	Component Catalog (1 👻 🛱 🗙								
	計 計 中								
	tion Concept Cypress								
	Cypress Component Catalog								
	⊕ <mark>oog</mark> Analog ⊕ oog CapSense								
	⊕ 🐼 Communications								
	🕀 💑 Display								
Philip Generates Notice List + 7 ×									
다 다 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이 이	-B Analog Pin [v1.60]								
Description File Error Location	Digital Bidirectional F								
	<u>۲ </u>								
	Component Preview Datasheet								
 III → □ Output Notice List 									
Ready 0 Errors 0 Warnings 0 Notes									

Step 5 ソースコードの編集



Step 6 ソースコードの解説

```
APIの解説
#include <device.h>
                                 uint8 Pin_1_Read( void )

    ピンコンポーネントの現在の状態を読み込む

void main()
                                 void Pin_1_Write( uint8 value )
{
                                 •ピンコンポーネントに値を書き込む
       for(;;)
        {
               if( SW_Read() ){
                       LED_Write(0);
               } el se {
                       LED_Write(1);
                                              追加分
                }
        }
}
```

Step 7 プロジェクトのビルド



Step 8 ビルド結果の確認



"---- Buid Suceeded: **/**/**** **:**:** ---"と表示されたらBuild成功です

Step 9 デバイスへの書き込み





GPIO_35l プロジェクトの目的 このプロジェクトでは、割り込みを用いたIO制御を行います プロジェクトの内容 •GPIO_35PプロジェクトをGPIO_35Iに複製します (GPIO_35Pのプロジェクトの資料を参考にしてください) •複製したGPIO_35Iプロジェクトをロードして開始します



Step 1 回路図エディタの操作1



Step 2 SWコンポーネントの設定変更



Step 3 コンポーネントの追加



Step 4 ソースコードの編集



Step 5 ソースコードの解説 #include <device.h> 追加分 CY_ISR(sw_isr){ CY ISR(MyISR)について •PSoC Creator では CY ISR というマクロが割り込みル SW_ClearInterrupt(); ーチン用に用意されています。左のように記述することで、 if(SW_Read()){ 割り込みルーチンを生成できます。 LED Write(0); void Pin 1 ClearInterrupt(void) } el se { •Pinコンポーネントの割り込み出力はレベル出力なので、 LED_Write(1); 割り込みが発生するたびにクリアする必要があります } } void main() 追加分 { ISR_StartEx(cyisraddress addrss) isr_1_StartEx(sw_isr); ●ISRコンポーネントの初期化、引数で指定した割り込みル CYGI obal IntEnable; ーチンのセット、及びその割り込みをEnableします CYGLobal IntEnable for(;;) •デバイス全体の割り込みをEnableします ł }

Step 6 プロジェクトのビルド



Step 7 ビルド結果の確認



"---- Buid Suceeded: **/**/**** **:**:** ---"と表示されたらBuild成功です

Step 8 デバイスへの書き込み





GPIO – I/O割り込み

From GPIO Pin

- GPIOポート毎に保有:
- Port Interrupt Control Unit (PICU)
- 専用の割り込みベクタ
- Interrupt on:
- Rising edge
- Falling edge
- Any edge
- Status Register
- どのピンによってI/O割り込みが発 生したかをラッチする
- ファームウェアによる読み出し可能
- 読み出しによるクリア





GPIO – アナログI/O機能

- 全てのピンがアナログ入出力に
 利用可能
- ピン毎に2種の異なる内部接続
 - Analog Global Bus
 - Analog Mux Bus
- いくつかのピンは特別な機能を 実現するための内部接続を持つ (P0[0], P0[1], P3[6], P3[7])
 - Opamps
 - High Current DAC mode
- CapSense Touch Sensing
- LCD char/segment drive
- Hardware controlled analog mux at pin



SIO (Special I/O) 機能

•入出力電圧とドライブ能力の異なるGPIO

• Vddaに左右されない5.5V tolerant機能

- ホットスワップ
- 過電圧耐性
- 入出力電圧レベルを調整可能
 DAC出力
 - High Speedコンパレータ入力
- Logic level max current
 - 25 mA sink
 - 4 mA source
- Pin max current
 - •~50 mA sink
 - •~25 mA source
- No Analog
- No LCD char/segment drive
- No CapSense touch sensing



課題演習

演習時に指示します

Lab GPIO_35



© Renji Mikami – 2012 PSoC 5 / 3 CUA Workshop

Memo

- フォローアップURL
- http://mikamir.web.fc2.com/?/?.htm
- ?に入る文字列は、講義中に示します。
- 担当講師
- ミカミ設計コンサルティング
- 〒142-0042 東京都品川区豊町 2-17-8
- 三上廉司(みかみれんじ)
- <u>Renji_Mikami@nifty.com</u>
- http://homepage3.nifty.com/western/mikamiconsult.htm
- 電話 080-5422-2503(au)