# Technical Note



## Silicon Labs 社 EFM32 クイックスタートガイド

2018年3月





TecStar -

Silicon Labs 社 EFM32

クイックスタートガイド

目次	
1 はじめに	4
2 開発環境のご紹介	5
2-1 ハードウェア	5
2-1-1 EFM32 Starter Kit	5
2-1-2 EFM32 Development Kit	10
2-2 ソフトウェア	
2-2-1 Simplicity Studio	11
3 各種ドキュメントの入手方法	13
3-1 ドキュメントの入手方法(Simplicity Studio から)	13
3-1-1 情報が表示されない場合には?	15
3-1-2 欲しい情報が見つからない場合には?	15
3-1-3 表示される情報を制限したい場合には?	
3-1-4 いつも使うドキュメントに素早くアクセスしたい場合には?	
3-2 ドキュメントの入手方法(Web から)	
3-3 Starter Kit と Development Kit のユーザガイド	
3-4 EFM32のAPI 情報	
4 ソフトウェア・インストール	
4-1 Simplicity Studio のインストール	21
4-2 インストールがうまくいかない場合	22
4-2-1 シリコンラボ社アカウントの取得方法	
4-2-2 企業プロキシサーバーを介して接続している場合	23
4-2-3 オフライン・インストーラ	25
5 ハードウェア・セットアップ	
5-1 Starter Kit のセットアップ	
6 使用方法	
6-1 サンプルコードを動かしてみる	27
6-2 デバッグ機能を使ってみる(Debug)	
6-9 消弗重法を測定してれる (France Droflan)	94
0 <sup>-</sup> 5 消貨电流を測定してみる(Energy Fromer)	
6-4 ピン設定やペリフェラル設定をしてみる(Hardware Configurator)	
6-3 肩貨電流を測定してみる(Energy Promer) 6-4 ピン設定やペリフェラル設定をしてみる(Hardware Configurator) 6-5 電池寿命の見積もりをしてみる(energy Aware Battery)	
6-3 消貨電流を測定してみる(Energy Promer)	
6-3 消貨電流を測定してみる(Energy Promer) 6-4 ピン設定やペリフェラル設定をしてみる(Hardware Configurator) 6-5 電池寿命の見積もりをしてみる(energy Aware Battery) 6-6 ユーザ基板のプログラミング・デバッグを行ってみる 6-6-1 参考資料	

6-6-3 ハードウェア接続 (例)49
6-6-4 デバッグ対象の切り替え51
6-7 ユーザ基板の消費電流を測定する53
6-7-1 VMCU をどこから取るか53
6-7-2 Starter Kit 上の部品の消費電流を極限まで下げる54
6-7-3 ユーザ基板の消費電流の測定手順55
7 ソフトウェア設計
7-1 Cortex-M を初めて使う方に(ARM 社ドキュメント)56
7-2 開発用のサンプルコードやライブラリについて57
7-3 ソースコードの追い方59
7-4 割り込みハンドラ
7-5 ピン設定、ペリフェラル設定の流れ59
7-6 ピン設定
7-7 ペリフェラル設定
7-7-1 USART (Asynchronous mode)
<b>7-7-2</b> I2C
7-7-3 タイマ
7-7-4 タイマ (X 秒タイマの作り方)73
7-7-5 CMU (ペリフェラル・クロックの周波数)
改版履歴
参考文献

### 1 はじめに

この資料は、Silicon Laboratories(以下、Silicon Labs)社製 MCU EFM32 ファミリの開発環境について 簡易にまとめたものです。内容に誤りがないよう注意は払っておりますが、もし Silicon Labs 社が提供する ドキュメント等と差異がございましたら、メーカー提供のものを優先してご参照ください。

また、Silicon Labs 社の ナレッジベース(FAQ)やコミュニティフォーラム(ユーザ同士で問題解決。 Silicon Labs のエンジニアも頻繁にコメントしています)には、本資料で取り上げていない様々な情報が記 載されております。

製品をご使用頂く過程で疑問や課題が生じることもあると思いますが、他のユーザが既に解決方法を 見つけている場合も多々ございます。非常に有益ですので、ぜひご活用下さい。

### ◆ アクセス方法

Simplicity Studio から

Getting Started	Documentation	Compatible Tools	Resources	
Technical Support	⇒ Simplicity Studio 4 - A b	etter Embedded Developer Experienc	ce - from Silicon Labs	*
Presentations and Brochures		<ul> <li>Ciscolinit</li> </ul>	v Ctudio /	
Silicon Labs Community		C arbihinn	y 310010 4	

#### Web Site から

<u>https://www.silabs.com/community</u> (Silicon Labs 社製品全般) https://www.silabs.com/community/mcu/32-bit/forum (32-bit MCU に特化)

◆ 使用方法



### 2 開発環境のご紹介

EFM32 の開発環境について、ハードウェアとソフトウェアに分けてご紹介します。

### 2-1 ハードウェア

EFM32の開発環境としては、Starter KitとDevelopment Kitを用意しております。Starter Kitには周辺装置やセンサが実装されていますので、簡単に EFM32 の機能を確認できるようになっています。 Development Kit は拡張性が高く、より高度な検証を行えるようになっています。

### 2-1-1 EFM32 Starter Kit

Starter Kit は、各ファミリに1種ずつ用意されています。同一ファミリであっても、USB 有無、LCD コントローラ有無などで幾つもの製品型番が用意されていますが、Starter Kit にはフルセットの MCU が実装されていますので、これを用いて設計を進めて頂くことが可能です。

Starter Kit には、下記が同梱されています。

Starter Kit

・USB ケーブル



### Giant Gecko S1(GG11): SLSTK3701A



#### • Wonder Gecko: EFM32WG-STK3800



Pearl Gecko PG12/ Jade Gecko JG12: SLSTK3402A





ver. 2.3 2018年3月



## ♦ Happy Gecko: SLSTK3400A



Zero Gecko : EFM32ZG-STK3200



### 2-1-2 EFM32 Development Kit

Development Kit は 3 枚の基板から構成されています。EFM32 が実装された MCU プラグイン・ボード と、色々な部品を実装できるプロトタイピング・ボードの 2 つを、マザーボードに挿入して使用します。

Development Kit には、下記が同梱されています。

・EFM32 Development Kit マザーボード

- ・EFM32 MCU プラグイン・ボード
- ・EXP32 プロトタイピング・ボード
- •IAR Embedded Workbench 評価版
- Atollic TrueSTUDIO
- ・ケーブル各種

ファミリ名	CPU Core	Starter Kit	実装されている型番
Wonder Gecko	Cortex M4F	EFM32WG-DK3850	EFM32WG990F256
Giant Gecko	Cortex M3	EFM32GG-DK3750	EFM32GG990F1024
Leopard Gecko	Cortex M3	EFM32LG-DK3650	EFM32LG990F256
Gecko	Cortex M3	EFM32G-DK3550	EFM32G890F128

### Giant Gecko: EFM32GG-DK3750



## 2-2 ソフトウェア

EFM32の開発環境である Simplicity Studio を使用して設計を行うことになります。

### 2-2-1 Simplicity Studio

Simplicity Studio は、EFM32 をターゲットとしたコンパイル・デバッグ・プログラミングを1 つのプラット で提供することができるソフトウェアです。統合開発環境(IDE)を中心に、非常に便利なツール群が充 実しています。同社製の 8bit MCU や無線 MCU も同一プラットフォームで開発が可能です。



注) 画像は Simplicity Studio v3 のものです

ツール名	機能の概要
Simplicity IDE	統合開発環境(IDE)。 無償の GCC コンパイラを搭載
Energy Profiler	実機の消費電流値を測定することが可能
Hardware Configurator	ピン設定やペリフェラル設定を簡単に行うことができる
energy Aware Battery	消費電流値のシミュレータ機能。バッテリ寿命も簡単に算出
Flash Programmer	フラッシュ ROM のライト/イレース

### ◆ 消費電流が実測できます(Energy Profiler)

EFM32 Wonder Gecko St	arter Kit Board (440( +				EI 🛃 📶	
<u> </u>				PROFILER_README.txt	segmenticd.c 🔀	- 6
[emicd.out] for V	/onder Gecko 380	0 Starter Kit (pr	3.9	int data, le uint16_t bitfield uint32_t com, bit	ngth, index; ;;	
Aug Current Aug Power	Total Energy Time Span	Mode Save Reset Compare	Arg Valla	int i;		
emion Counter 204.12 µA 673.90 µW	130.14 µJ 54.84 s	Paused 1 1 1 2	*	<pre>length = strlen(s index = 0;</pre>	string);	
		4194 <b>-</b>		/* If an update i LCD_SyncBusyDelay	s in progress we must block, (@xFFFFFFF);	or there m
				/* Freeze LCD to LCD_FreezeEnable(	avoid partial updates */ (true);	
				/* Turn all segme SegmentLCD_AlphaN	<pre>ints off */ lumberOff();</pre>	
=				<pre>/* Fill out all c for (index = 0; i {</pre>	ndex < 7; index++)	
I a la l				if (index < len	igth)	
· ·······	- Sun true to	and the second		data = (int) }	*string;	
				else	/* Padding with space */	
				data = 0x20; }	/* SPACE */	
				/* Defined lett data = data - 0	ers currently starts at "SPAN x20:	CE" - ASCII
				/* Get font for	this letter */	
				Ditrield = EFM_	Alphabet[data];	
				for (i = 0; i <	: 14; i++)	
IRQ RTC_IRQHandler			1.00 s ~ + − D	bit = EFM_Dis	play.Text[index].bit[i];	
1				con = crm_ots	pray.lexc[index].com[i];	
Energy Profile (live) 🕄	_			if (bitfield	& (1 << i))	
C Function	Energy	Contribution (%)	v	/* Turn on	segment */	
HIC_CounterGet	Lm 10.08	68.831%		LCD_Segment	Set(com, bit, true);	
LCD_SyncdusyDelay	31.09 mJ	23.892%		}		
CMU_UscillatorEnable	6.96 mJ	5.36%		}		
LCD SegmentSet	390.02 ml	0.665%		string++;		
LOD_segmentset	300.03 µJ	0.292%		}		
A AC BlickTeat	165 70 µJ	0.157%		/* Enable update	*/	
delau/Tieke	100.70 µJ	0.127%		LCD_FreezeEnable(	false);	
Compati CD Number	07 10 ml	0.088%		}		
LCD SegmentSetLow	60.10 pJ	0.087%				
LCD_SegmentSetLow	86.21 µJ	0.066%				



Edit Novigate Search Run Pr	yect Window Help		
1 1 · 1 · 1 · 5 · 5	♦ 44 × 5 12 ♥ < □ PtPer		B & O F B
oject Explorer 🖾 😐 !	1 😳 trayProject/seconf 😰 🧮	C BE Outline II	-
B 🕸 '	DefaultMode Port I/O: PORTIO	DefaultMode	
3 myProject (GNU ARM v4.3.3 - Deb 8-000 Includes		B B Port 1/0	
CMS2S		PortA	
inc inc		PortC	
and		PortD	
B/ myrroject/wconn		PortF	
	OHIDAT / BRADE / L BLADE / DE BEEN AN 1000 / MARY BRADE / L	Properties	Mappine 23
		DefaultMode : PORTIO	
	190 190 190 190 190 190 190 190 190 190	ACMP0	CH0 PC0
			CH1 POI
	200 PAS PAS PAS PAS PAG		CH2 PO2
			<b>1</b> CH1
			A CH4
	75 75 10113 0700 0703 0703 7703 FG		CH5
			СН6
	FIS FO2 FO3 FAT FAT FAS STATES STATES FOS FOS FOT		CH7 PC7
		- 9	OUT PELL
		ACMP1	CHO
	ADMEDIT. XOMEDIC. 1000 100100 100100 10000 10000 1000007. 000007. 000007.		1001
	NO AG AG UNU AND BUT NOT NOT NOT NOT ADD ADD ADD ADD ADD ADD ADD ADD ADD AD		082
	u 🙂 🙂 🐱 🙂 🙂 🙂 🙂 🙂 🙂		СНЗ
	HE HO HON HON HAT HET HET HET HET HET HET HET HET HET HE	1	TUO
		ADC8	СН0
	Detau/Mode Port I/0 1     Detau/Mode Pergherals		СН1 РОТ

ピン設定やペリフェラル設定を簡単に行えます(Hardware Configurator)

◆ バッテリ寿命のシミュレーションが行えます(energy Aware Battery)



注意: 対応している製品は Gecko, Giant Gecko (S0), Leopard Gecko, Tiny Gecko (S0), Wonder Gecko, Zero Gecko の6ファミリのみです。

## 3 各種ドキュメントの入手方法

EFM32のドキュメントの入手方法について紹介します。

### 3-1 ドキュメントの入手方法 (Simplicity Studio から)

EFM32 のデータシート、リファレンス・マニュアル、エラッタ、アプリケーションノート および 評価基板 (starter kit)の回路情報などは、Simplicity Studio からご入手頂くことが可能です。

Simplicity Studio を起動し、Solutions タブ ⇒ 空欄に使用する製品型番を入力 ⇒ 候補の中から該 当する型番を選択します。



製品型番を指定すると、関連するドキュメントやサンプルコードが自動でリストアップされます。情報の 種別に応じて、Getting Started、Documentation、Compatible Tools、Resources というタブに分類されてい ます。

◆ Getting Started タブ

Getting Started	Documentation		Compatible Tools	Resources	
Demos	-+⊠≡	Software Examples	-+⊠≡	SDK Documentation	-+⊻≡
<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>Demos</li> </ul>		<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>EFM32TG11 Tiny Gecko Starter Kit</li> </ul>		<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>API References</li> </ul>	
		<ul> <li>Gecko Bootloader 1.5.0</li> <li>Gecko Bootloader Example</li> </ul>	amples	Release Notes	

#### Demos:

評価基板上で動作するデモンストレーション用のソフトです。Build することなくモジュールに書き込んで、動作を確認することができます。

#### Software Example:

評価ボード上で動作するサンプルコードです。ソフトの実装方法について学んだり、機能について理解したりするのに役立ちます。ペリフェラルごとにサンプルコードが用意されています。

#### SDK Documentation:

TecStar =

API リファレンス・マニュアル、リリースノートが入手できますが、後述する Documentation タブからも 入手できますので、そちらで説明します。

#### Documentation タブ **Getting Started Compatible Tools** Documentation Resources My Favorite Documents 💻 🕂 🗹 All Documents -+ 🗹 No documents have been favorited. Click the 'Favorite' icon to add a document here. Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU 5.4.0.0 EFM32TG11 Tiny Gecko Starter Kit (SLSTK3301A) API References Application Notes Data Sheets Errata Reference Manuals Release Notes Schematic and Layout Files User's Guides

#### My Favorite Documents:

お気に入り登録したドキュメントがリストアップされます。

#### All Documents:

各種ドキュメントがまとめてあります。

- API References ... API リファレンス・マニュアル。

- Application Notes ... 特定の用例について記しています。各ペリフェラル (ADC やシリアルインタフェ
- ースなど)の使用方法に関する情報も用意されています。
  - Data Sheets ... EFM32 のデータシート。
  - Errata ... EFM32 のバグ情報。
  - Reference Manual ... EFM32 の動作仕様書。
  - Release Notes ... SDK のリリースノート
  - Schematic and Layout Files ... 評価ボードの回路図・部品表・レイアウト情報。
  - User's Guide ... 評価ボードの取説。

#### 3-1-1 情報が表示されない場合には?

TecStar —

Demos, Documentation などに情報が表示されない場合には、SDK が適正に選択されていない可能 性があります。下図を参考に、Gecko SDK が選択されているか確認してみてください。SDK が選択され ていない場合には、<u>Click here</u> から SDK を選択してください。

and Launcher - soc-ibeacon_interrupt_test/main.c - Simplicity St	tudio ™		- • •
File Edit Source Refactor Navigate Search Project R	tun Window Help		
Sign In 👻 🔅 🛃 🎤	Search	😰 🔓 Launcher {	👌 Simplicity IDE 🛛 🎄 Debug
Solutions     Solution Enter product name     Enter product name     Enter Stater Kit (SLSTK3301A     EFM32TG11 Tiny Gecko Starter Kit (SLSTK3301A     EFM32TG11S1B520F128GM80	EFM32TG11 Tiny Preferred SDK Gecko SDK Suite v2.2.1 Bluet preferred SDK.	Gecko Starter Kit (SLSTK3301A)	3

また、Solutions タブで何を選ぶかで、表示される情報も変わりますので、その点も確認ください。

Solutions	Ð	Ŧ	
• New Solution			
Enter product name			
🔺 🧀 temp			*
EFM32TG11 Tiny Gecko Starter Kit (SLSTK3301A)			
EFM32TG11 Tiny Gecko Starter Kit board (BRD2102A Re	v A	<b>\04</b> )	=
EFM32TG11B520F128GM80			

#### 3-1-2 欲しい情報が見つからない場合には?

All Documents の右横にある **ビ** アイコンで、表示項目を選択したり、表示項目の並び替え (Move Up / Move Down)を行うことができます。

Favorite Documents       All Documents         odocuments have been favorited. Click the 'Favorite' icon to add a document here.       Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0,         Image: Comment Categories Dialog       Image: Comment Categories Dialog       Image: Comment Categories Dialog         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Comments in the categories in the categories.         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Comments in the categories.         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Comments in the categories.         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the categories in the categories.         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the Category         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the Category         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the Category         Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the Category         I	tting Start	ed Documentation		Compatible Tools	Resour	rces
documents have been favorited. Click the 'Favorite' icon to add a document here.       Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0,         Image: Check the categories Dialog       Image: Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.       Image: Check the category	Favorite Doo	cuments	-+0	All Documents		- 4
Document Categories Dialog All Documents Preferences Use this dialog to customize your documentation preferences Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories. Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories. Include Preferred SDK - Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU 5.4.0.0   Show Category Documents   Move Up API References 1   Application Notes 76 Move Down   Application Notes 1 Move Down   Errata 1 Select All   W Reference Manuals 1   W Reference Manuals 2   W User's Guides 2	documents have b	peen favorited. Click the 'Favorite' icon to add a doc	ument here.	Gecko SDK Suite v2.2.1: Bluetooth	2.8.0.0, EmberZNet (	5.2.1.0, Flex 2.2.1.0, MCU
All Documents Preferences         Use this dialog to customize your documentation preferences         Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.         Include Preferred SDK - Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU 5.4.0.0         Show       Category         Q       API References         N       Application Notes         P       Data Sheets         1       Move Down         Q       Reference Manuals         N       Reference Manuals         Schematic and Layout Files       1         User's Guides       2	🛃 Documer	nt Categories Dialog		F 4 6 6		
Check the categories you would like to display. Use 'Move Up' and 'Move Down' to reorder the categories.          Include Preferred SDK - Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU 5.4.0.0         Show       Category         V       API References         Application Notes       76         V       Data Sheets         V       Errata         V       Reference Manuals         V       Reference Manuals         V       Schematic and Layout Files         V       User's Guides	All Docum Use this o	nents Preferences dialog to customize your documentation preferences	5			(3)
V     API References     1       V     Application Notes     76       V     Data Sheets     1       V     Errata     1       V     Reference Manuals     1       V     Release Notes     1       V     Schematic and Layout Files     4	Check the o	categories you would like to display. Use 'Move Up' Preferred SDK - Gecko SDK Suite v2.2.1: Bluetooth Category	and 'Move Down' 2.8.0.0, EmberZN	to reorder the categories. let 6.2.1.0, Flex 2.2.1.0, MCU 5.4.0.0	uments	Move Up
V     Application Notes     76     Move Down       V     Data Sheets     1       V     Errata     1       V     Reference Manuals     1       V     Release Notes     1       V     Schematic and Layout Files     4       V     User's Guides     2		API References		1		
V     Data Sheets     1       V     Errata     1       V     Reference Manuals     1       V     Release Notes     1       V     Schematic and Layout Files     4       V     User's Guides     2		Application Notes		76		Move Down
Image: Constraint of the second se		Data Sheets		1		
Image: Constraint of the second se		Errata		1		
Image: Release Notes     1       Image: Schematic and Layout Files     4       Image: Schematic and Layout Files     2		Reference Manuals		1		Select All
Image: Schematic and Layout Files     4       Image: With the schematic and Layout Files     2		Release Notes		1		
User's Guides 2		Schematic and Layout Files		4		Unselect All
		User's Guides		2		
	•		m		•	
						L8 Silico

### 3-1-3 表示される情報を制限したい場合には?

TecStar =

Demos, Software Examples, Documentationの右横に、4つのアイコンが並んでいます。このアイコンを 使うことで、表示される情報を制限することができます。



### 3-1-4 いつも使うドキュメントに素早くアクセスしたい場合には?

各ドキュメントの右横にある☆印をクリックすると、☆の色が変わり、My Favorite Documents に追加されます。良く使うドキュメントを追加しておくと便利です。

Getting Started	Documentation		Compatible Tools	Resources
My Favorite Documents		-+ 🗹	All Documents	-+ 🖂
Data Sheets  EEM22TG11 Data Sheet			Gecko SDK Suite v2.2.1: Bluetooth 2 5.4.0.0	.8.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU
Description is unavailable			API References     Application Notes	SLSTK3301A)
			EFM32TG11 Data Sheet Description is unavailable	<b>≞</b> ★

### 3-2 ドキュメントの入手方法 (Web から)

TecStar —

EFM32 のデータシート、リファレンス・マニュアル、エラッタ、アプリケーションノート および 評価基板 (starter kit)の回路情報などは、Silicon Labs 社の Web Site からもご入手可能です。 http://www.silabs.com/support/pages/document-library.aspx

Products や Resource Type で、リストアップする対象を絞り込むこともできます。 EFM32 は、Products -> Microcontrollers -> 32-bit MCUs の下に分類されています。

Silicon Labs » Support » Technical Resource Search

## Technical Resource Search

#### Expand All / Collapse All Showing 50 of 258 Results

Narrow by:			Q,
* Products: Giant Gecko	Title 👻	Version	Resource Type
Clear All	8 Bit Tools package, Release Notes	4.90	Software
Products	8 Bits Tools Installer 🔒	4.90	Software
Analog     Audio and Radio	AN0002 Hardware Design Considerations	1.35	Example Code
<ul> <li>Additional to Radition</li> <li>Interface</li> </ul>	AN0002.0: EFM32 and EZR32 Wireless MCU Series 0 Hardware Design Considerations	1.46	Application Notes
<ul><li>Isolation</li><li>Microcontrollers</li></ul>	<ul> <li>▲ AN0002.0: EFM32 および EZR32 Wireless MCUシリーズ 0 ハード</li> <li>◆ ウェア設計上の考慮事項</li> </ul>	1.46	Application Notes
32-bit MCUs	AN0002.0: EFM32 和 EZR32 无线 MCU 系列 0 硬件设计注意事项	1.46	Application Notes
Gecko	Resource Type –	1.80	Example Code
Giant Gecko	Application Notes	1.71	Application Notes
Happy Gecko	Data Sheet Addendums	1.07	Evenuela Carda
Jade Gecko	Data Sheets	1.07	Example Code
Leopard Gecko	Errata     /ireless MCU Series 0 Clock	1.09	Application Notes
Pearl Gecko	Example Code		
	Getting Started		
	🗆 Manuals 🗸 👻		
	Miscellaneous		
	Product Change Notifications (PCN)		
	Reference Designs		
	Release Notes		
	Schematic and Layout Files		
	Software		

## 3-3 Starter Kit と Development Kit のユーザガイド

以下に Starter Kit および Development Kit のユーザガイドのリンクを掲載します。

## ◆ Starter Kit のユーザガイド

Giant S1 (GG11): https://www.silabs.com/documents/public/user-guides/ug287-stk3701.pdf Wonder: https://www.silabs.com/documents/public/user-guides/ug257-stk3402-usersguide.pdf Pearl PG12: https://www.silabs.com/documents/public/user-guides/ug154-stk3401-user-guide.pdf Giant: https://www.silabs.com/documents/public/user-guides/efm32gg-stk3700-ug.pdf Leopard: https://www.silabs.com/documents/public/user-guides/efm32lg-stk3600-ug.pdf Gecko: https://www.silabs.com/documents/public/user-guides/efm32lg-stk3600-ug.pdf Tiny: https://www.silabs.com/documents/public/user-guides/efm32-stk-documentation.zip Tiny: https://www.silabs.com/documents/public/user-guides/efm32-stk-documentation.zip Tiny S1 (TG11): https://www.silabs.com/documents/public/user-guides/efm32-stk300-ug.pdf Happy: https://www.silabs.com/documents/public/user-guides/efm32-stk300-ug.pdf

### ◆ Development Kit のユーザガイド

Wonder: <u>https://www.silabs.com/documents/public/user-guides/efm32wg-dk3850-ug.pdf</u> Giant: <u>https://www.silabs.com/documents/public/user-guides/efm32gg-dk3750-ug.pdf</u> Leopard: <u>https://www.silabs.com/documents/public/user-guides/efm32lg-dk3650-ug.pdf</u> Gecko: <u>https://www.silabs.com/documents/public/user-guides/efm32g-dk3550-ug.pdf</u>

### 3-4 EFM32のAPI 情報

ペリフェラルを制御するためライブラリ(API)が用意されており、それを使用することでソフト設計を円 滑に進めて頂くことが可能です。

「3-1 EFM32 のドキュメント」の手順に従って使用する型番を選択し、Documentation タブ ⇒ API reference ⇒ Gecko HAL and Driver API Reference Guide を選択します。

Getting Started	Documentation	Compatible Tools	Resources
My Favorite Documents	-+ 🗹	All Documents	-+ 🗹
No documents have been favorited.	Click the 'Favorite' icon to add a document have.	Gecko SDK Suite v2.2.1: Bluetooth 2 5.4.0.0 EFM32TG11 Tiny Gecko Starter Kit ( • API References	28.0.0, EmberZNet 6.2.1.0, Flex 2.2.1.0, MCU SLSTK3301A)
		Gecko HAL and Driver API Ref API reference for CMSIS, EMLIB library, middleware and board su	erence Guide 🗎 🏠 Peripheral Library, energyAware Driver upport packages.

ブラウザが起動し、製品ファミリー覧が表示されますので、使用するファミリを選択します。



なお、CMSIS や、mbed TLS(暗号化アクセラレータで使用するライブラリ)の API 情報も、ここから入 手頂けます。

ウィンドウ左で閲覧したいライブラリや APIを選択すると、右側に情報が表示されます。

EFM32 Tiny Gecko 11 Software Documentation

efm32tg11-doc-5.4

Main Page	Modules	Files	Documentation Home	silabs.com	<b>Q-</b> Search
<ul> <li>EFM32 Tiny Gecko</li> <li>Modules</li> </ul>	11 Software Documer	EFM3	2 Tiny Gecko 11 Softv	vare Documentati	on
► BSP		Welcome	e to the software documentation f	for the EFM32 Tiny Gecko 11.	Here, you will find documentation for
<ul> <li>Devices</li> <li>EMDRV</li> <li>EMLIB</li> <li>Kit Drivers</li> <li>Platform Mid</li> <li>Files</li> <li>Documentation silabs.com</li> </ul>	dleware Home	CN EN En Pla Bo Kit Please al	ISIS-CORE Device headers for th ILIB Peripheral Library ergyAware Driver Library atform Middleware aard Support Package t Driver Library so see Simplicity Studio for preco	e EFM32 Tiny Gecko 11 mpiled demo applications, a	pplication notes and software examples.
◆ GP ► CHIP	ΙΟ Φ ΑΡΙ	GPI	0		Macros   Enumerations   Functions
► CMU		EMLIB			
► COMMON	N				
▶ CORE		_ Deta	ailed Description		
▶ CRYOTIM	ER				
▶ CRYPTO		Gener	al Purpose Input/Output (GPIO)	API.	
▶ CSEN		This m	nodule contains functions to con	trol the GPIO peripheral of s	Silicon Labs 32-bit MCUs and SoCs. The GPIO
► EMU		conne	ections.	n and direct pin manipulatio	on and sensing as well as routing for peripheral pin
▶ GPCRC					
► GPIO		Mac	ros		
▶ I2C		#def	ine gpioDriveStrengthStrong	gpioDriveStrengthStrong	AlternateStrong
► INT		#def	ine gpioDriveStrengthWeak	gpioDriveStrengthWeakAl	ternateWeak
► LCD		Enu	morations		
► LDMA		Enu			
▶ LESENSE		enun	n GPIO_DriveStrength_TypeD gpioDriveStrengthWeakAl	ef { ternateWeak = GPIO_P_CT	RL DRIVESTRENGTH WEAK
► I FTIMER			GPIO_P_CTRL_DRIVESTRENG	THALT_WEAK,	
► I FUART				ternateStrong = GPIO_P_C	TRL_DRIVESTRENGTH_WEAK
► MPU		-	gpioDriveStrengthStrong	lternateWeak = GPIO_P_C	TRL_DRIVESTRENGTH_STRONG
. III		•	GPIO_P_CTRL_DRIVESTRENG	THALT_WEAK,	

## 4 ソフトウェア・インストール

EFM32 の評価に必要なソフトウェアをインストールします。

4-1 Simplicity Studio のインストール

下記 URL よりインストーラをダウンロードしてください。

http://www.silabs.com/products/mcu/Pages/simplicity-studio.aspx

ダウンロード完了後 "install-studio-v4\_xx.exe" を起動し、インストールを開始してください。

Simplicity Studio はインターネット回線に接続した上でのインストールを想定しています。オフラインでの インストールは行えませんのでご注意ください。

インストールが進むと、Log in (サインイン) 画面が表示されますが、サインインしなくてもインストール できます。更にインストールが進むと、Install Wizard が表示されます。 Select by Product Group タブ に切り替え、EFM 32-bit Products にチェックして、Next をクリックします。

Select the devices you would like to use within Sim	plicity Studio.	
elect by Device Select by Product G	COUP	EFM 32-bit Product
Selected Products: 18	Download Size: 1.14 GB Requ	ired Disk Space: 2.63 GB

インストールを行うモジュールが自動でリストアップされます。Finish をクリックすると、残りのインスト ールが実行されます。以上でセットアップは完了です。

Install Wizard が起動しない場合には、Update Software アイコンをクリックしてください。

Sign In 👻 🕴	¢. <b>≗</b>	
<ul> <li>Devices</li> </ul>	Solutions	" 🗖

4-2 インストールがうまくいかない場合

TecStar —

#### 4-2-1 シリコンラボ社アカウントの取得方法

EFM32 SDK の入手には、シリコンラボ社 WEB サイトのアカウントは不要ですが、Bluetooth SDK などのインストールを行う際には必要となります。アカウントの作成は無料です。

① 下記 URL にアクセスし、右上の Register からアカウント作成に進んでください。

https://www.silabs.com/

				简体中文 繁體中文 日本語	Log In   Register
SILICON LABS			Parametric Search   Cros	s-Reference Search	
About 🔻	Products 🔻	Solutions 🔻	Community & Support 🔻	Search silabs.com	GO

② 必要事項を入力し、Create an Account でアカウントを作成してください。

All fields require	ed	
First Name		
Taro		名前
Last Name/Family Name		
Yamada		苗字
Company Name		
Macnica		会社名
Email		
xxxxxx@xxxxx.co.jp		メールアドレス
Password		
•••••		パスワード
Confirm Password		
•••••	]	 パスワード(再入力
Country		
Japan	•	国名
State		
Kanagawa	-	
Zip Code		
2228561		郵便番号
I would like to receive email from Silicon Labs	communications	
Create an Acc	ount	

③ アカウントが生成できたら、念のため発行されたアカウントでログインできることを確認してくださ

### い。下記 URL にアクセスし、右上の Log In からログインを行ってください。

https://www.silabs.com/

TecStar =

6	Y			简体中文 繁體中文 日本語	Log In   Register
SILICON	LABS			Parametric Search   Cros	ss-Reference Search
About 👻	Products 🔻	Solutions 🔻	Community & Support 👻	Search silabs.com	GO

④ ログインに成功すると、画面右上に「Welcome, 名前」が表示されます。

	简体中文 繁體中文 日本語	Welcome, 🔽 🔻
SILICON LABS	Parametric Search	Cross-Reference Search
About ▼ Products ▼ Solutions ▼ Commun	y & Support 🔻	GO

#### 4-2-2 企業プロキシサーバーを介して接続している場合

インストールにはインターネット接続が必要になりますが、プロキシサーバーを導入している企業ユー ザ様の場合にはプロキシ設定が必要になる場合があります。設定内容については、自社のネットワーク 管理者にご相談下さい。プロキシを介さずにインターネット回線に接続できる環境が構築できる場合に は、そちらをご利用頂くのが簡単です。(WiFi ルータや自宅など)

Simplicity Studio がアクセスする先については、シリコンラボ社のコミュニティフォーラムに関連情報があります。企業プロキシサーバーのセキュリティオプション(ホワイトリスト)で回避するような場合にご利用ください。

 $\label{eq:http://community.silabs.com/t5/Simplicity-Studio-and-Software/Simplicity-Studio-v4-installation-error-download-error/tapp/181331$ 



プロキシサーバーの設定は、以下の手順で行います。

Simplicity Studioの Settings アイコンを選択し、Network Connectionsを選択します。プロキシ設定の画面が表示されますので、Active ProviderをManualに設定変更し、Proxy entriesに必要な設定を入力してください。

Sign In 🔸 🔅 🧏			Search	📕 Tools
Devices     Solutions      New Solution     Add Devi      Preferences	ices	Welcon	ne to Sin	nplicity Studic
type filte text Capabilities > Help	Network Conne	ctions Manual 🗸		
Network Connections ▷ Security ▷ Simplicity Studio	Sch H ♥ HTTP ♥ HT ♥ SO	lost Port ↓ 頁目·設定内容	Prov Auth U Man No Man No Man No	ser Passwon Edit Clear こよって異なります
	Host V localhost V 127.0.0.1		Provider Manual Manual	Add Host Edit Remove
			ОК	Cancel

設定が終わったらログイン(Sign In)を行います。画面左上の Sign In をクリックし、シリコンラボ社 WEB サイトのアカウントを入力します。ログインに成功すると、画面左上にメールアドレスが表示されま す。

🧢 Simplicity Studio ™	🥌 Sim	plicity Studio ™	
File Help	File	Help	
Sign In 👻 🔅 🛃 😽 🗕	<b>→</b> =	:macnica.co.j	p 👻 🌣 🛃
Devices     Solutions	- De	evices • Solutions	
• New Solution • Add Devices	( <u>•</u> )	New Solution 🕢 Add [	Devices

ログインに成功したら、Update Software アイコンをクリックし、Package Manager の Install Wizard からインストールが継続できます。

🕶 Launcher	- Simplicity Studio ™	
File Edit	Navigate Search Project Run Wind	low He
@	macnica.co.jp 👻 🗧 🥕	_
Devices	🍫 🖆 📝 🗙 👌 Update Softwar	e 🗆



## 設定例: PC とプロキシサーバー間の通信に HTTP のみを使用している場合

Active Provider: Manual 👻									
Pro	Proxy entries								
	Schema	Host	Port	Provider	Auth	User	Password		Edit
	HTTP			Manual	No				Clear
	HTTPS			Manual	No				
8	SOCKS			Manual	No				
	HTTP	Dynamic	Dynamic	Native	No				
Pro	xy bypass								
	Host		Provider					Add	d Host
	☑ localhost Manual							Edit	
	V 127.0.0.1 Manual							Luit	
								R	emove

### 4-2-3 オフライン・インストーラ

オンラインでインストールすることが望ましいですが、どうしてもプロキシの設定がうまくいかない場合には、オフライン・インストーラも活用頂けます。

入手については、マクニカオンラインサービスの FAQ をご参照ください。

https://service.macnica.co.jp/support/faq/125501

## 5 ハードウェア・セットアップ

EFM32の評価に必要なハードウェアの設定を行います。

### 5-1 Starter Kit のセットアップ

以下の手順で設定していきます。

- BAT, USB, AEM(DBG)の中から、基板に給電する方法を選びます。スイッチを AEM(DBG)に 切り替えます。
- 2. DBG とPCをUSB ケーブルで接続します



### 6 使用方法

Starter Kit と Simplicity Studio を使用した評価手順をご紹介します。ここでは STK3700(Giant Gecko)を 使用しておりますが、他の Starter Kit でも手順は同じです。なお、各ツールから Simplicity Studio の Top 画面に戻るには、画面右上の Launcher アイコンを使用します。



### 6-1 サンプルコードを動かしてみる

Starter Kit 上の LED を点滅させるサンプルコードを、ダウンロードして動作を見てみます。 Starter Kit を PC に接続すると、Simplicity Studio が Starter Kit を自動認識します。 Device タブに接続 した Starter Kit および MCU の名称が表示されますので、Starter Kit を選択してください。

うまく認識してくれない場合には、Refresh アイコンを押してみてください。



Getting Started タブ  $\Rightarrow$  Software Examples 横の View All Software Examples を選択します。

Getting Started	Decume	entation	Compatible Tools	Resources	
Demos	-+⊠≡	Software Examples	-+⊻≡	SDK Documentation	-+⊻≡
<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>Demos</li> </ul>		<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>Giant Gecko 3700 Sta</li> </ul>	arter Kit	<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>API References</li> </ul>	
				Release Notes	

#### 接続した Starter Kit に合せて、Boards, Part, SDK が自動で選ばれますので、Next をクリックします。

New Silicon Labs Proj	ect							
Project setup Select the board, part	, and SDK for the proje	ct.						
Boards:								
Search								-
EFM32 Giant Gecko S	Starter Kit board (BRD2)	200A Rev A03)	<					
Part:								
								-
EFM32GG990F1024								
SDK:								
Gecko SDK Suite: Blu	Jetooth 2.3.0.0, EmberZ	ZNet 5.9.0.0, Fle	x 1.1.0.0, MCU 5	.1.2, Thread 2.2.0	.0 (v1.0.0) (C:¥	SiliconLabs¥S	implicityStudio¥v	4¥dev 🔻 🕕
							M	anage SDKs
?		< Back	N	ext >	Fin	ish	Can	cel
0			-					

Example Project で STK3700\_blink(STKxxxx\_blink)を選択し、Nextをクリックします。

New Silicon Labs Project	
Example Project Select the project template to open in Simplicity IDE.	
	iii 😽 🍝
🔺 💐 Giant Gecko 3700 Starter Kit	
STK3700 biometric	
STK3700_blink	
STK3700_burtc	
STK3700_clock	
STK3700_cpt007b_demo	
STK3700 celib	•
Blink example. Template for new projects.	Â
This example project use EFM32 CMSIS and the emlib peripheral library to	
a skeleton for new projects.	E
Roard, Silion Labs FEM3266 STR3700 Starter t	
Device: EFM32GG990F1024	~
◀	
Compared with a sector of the secto	Cancel

プロジェクト名を入力し、作業フォルダを指定します。With project files では、サンプルコードをローカ ルにコピーして使うかどうかを指定します。指定が終わったら、Finish をクリックします。



With project files	内容		
	ライブラリもソースも、オリジナルのものを使う。ライブラ		
Link to sources	リもソースも修正しない人向け。		
Tinh librarian and some sources	ライブラリはオリジナルのものを参照し、ソースコードは		
Link libraries and copy sources	ローカルにコピーして使う。		
Come contents	ライブラリもソースも、ローカルにコピーして使う。ライブ		
Copy coments	ラリを修正する可能性がある人向け。		

サンプルコードの準備が整うと、Simplicity IDE が起動します。Simplicity IDE の使い方については 「6-2 デバッグ機能を使ってみる」で詳しく紹介します。



この Simplicity IDE を使用して、サンプルコードをビルドし、Starter Kit にダウンロードします。まずはト ンカチのアイコン(Build)をクリックします。コンパイラが走り、サンプルコードがビルドされます。



ビルドが完了したら、次に虫のアイコン(Debug)をクリックし、Starter Kit にダウンロードします。

File	Edit	Source	Refactor	Navigate	Search
脊	· 🚷	• 📑 •		🛞 <del>-</del> 🐔	•

ダウンロードが完了すると、デバッグ用の画面に切り替わります。



なお、ビルド用の画面と、デバッグ用の画面の切り替えは、ウィンドウ右上のアイコンで行います。



サンプルコードを実行します。下図の実行のアイコン(Resume)をクリックしてください。

 File
 Edit
 Source
 Refactor
 Navigate
 Search
 Project
 Run
 Window
 Help

 N
 Image: Source
 Image: Source
 N</

Starter Kit 上の LED が、ゆっくりと点滅しているのが確認できます。STK3700 の場合には、LCD 下の LED0 と LED1 が点滅します。



◆ LED の点滅スピードを変更してみましょう。

Simplicity IDE アイコンをクリックして、ビルド用の画面に切り替えます。

画面左に Project Explorer があり、ソースコードの階層が表示されています。emlib や BSP など、API が整備されていますので、簡単に評価・設計を行える環境が整っています。



blink.cにmain()ループがありますが、最後あたりにDelay(1000)という記述が見つかります。この数値 が点滅時間を決定しています。

<u>c</u> t	link.c 🛛	
	<pre>/* Infinite blink loop */ while (1) {    BSP_LedToggle(0);    BSP_LedToggle(1);</pre>	*
	Delay(1000);	
	}	-
	< III	Þ.

**TecStar** 

値を200に変更してみます。

TecStar —

.c >	*blink.c 🛛	
	<pre>/* Infinite blink loop */ while (1) {     BSP_LedToggle(0);     BSP_LedToggle(1);</pre>	*
	<pre>Delay(200); } </pre>	-
	4	

あとは、ビルドして、ダウンロードして、実行します。

先ほどと同じ手順で、トンカチのアイコン(Build)  $\Rightarrow$ 虫のアイコン(Debug)  $\Rightarrow$ 実行のアイコン(Resume) の順にクリックします。

Starter Kit の LED の点滅が、先ほどよりも早くなったことを確認できるかと思います。

#### 6-2 デバッグ機能を使ってみる (Debug)

**TecStar** 

Debug では、ブレークポイント、ステップ実行などのソフトウェア・デバッグの機能がご使用になれます。 「6-1 サンプルコードを動かしてみる」でも紹介しましたが、ビルド用の画面(Simplicity IDE)と、デバッグ 用の画面(Debug)は右上のアイコンで切り替えます。



ブレークポイントを設定するには、停止させたい行の左横をダブルクリックします。設定されると、水色の小さな〇印が表示されます。再度ダブルクリックすれば解除されます。





CPU Core	ハードウェア・ブレークポイントの数
Cortex M0+	4
Cortex M3	6
Cortex M4F	6

◆ ステップ実行

各種ステップ実行に対応しています。



実機で実際に動作を見て頂くのが、判りやすいです。



#### ◆ レジスタ値の閲覧・変更

レジスタ・変数の閲覧や変更は、下のウィンドウ(Register ウィンドウなど)で行うことができます。前回の停止から、値が変化した場合には黄色で表示されます。

×)= Variables	⁰ Breakp		¤ ∰ Express	
			£ ⇒t: E [	Ĵ ĽÍ
Name	•	Value	Description	
Marcel			General Purpose Regis	ters
⊳ 👬 DMA			DMA	
AES			AES	
⊳ 👬 USB			USB	
MSC			MSC	
EMU			EMU	
RMU			RMU	
MU CMU			CMU	
Issense			LESENSE	
RTC			RTC	
Iterimero			LETIMER0	
EBI			EBI	
▷ ## USART0			USART0	

6-3 消費電流を測定してみる (Energy Profiler)

Starter Kit には電流センサが搭載されており、消費電流測定ツール(Energy Profiler)と組み合わせる ことで nA レベルでの電流測定が可能です。Starter Kit には LCD など外部部品も実装されていますが、 MCU 単体の消費電流が測定できるように配慮されています。

ここではサンプルコードを使用しながら、消費電流測定ツール(Energy Profiler)の使用方法をご紹介します。

まず、「6-1 サンプルコードを動かしてみる」を参考に、サンプルコード「STK3700\_touch」をロードし、 ビルドします。(Build までです。Debug は行いません)

このサンプルコードは、通常時は Idle(EM2)で待機しており、Starter Kit上のタッチ・スライダーに触れ ると通常動作(EM0)に移行する、というモード遷移を含んだものです。EM0⇔EM2の遷移で消費電流 が大きく変化しますので、消費電流の観点では非常に判りやすいサンプルコードになっています。



次に Project Explorer でプロジェクトを選択して右クリック  $\Rightarrow$  Profile As  $\Rightarrow$  Simplicity Energy Profiler Target を選択します。Starter Kit にコードがダウンロードされ、Energy Profiler が起動します。



電流波形が表示されない場合には、画面右上の "Current"を選択してください。なお、"Voltage"を 選択すると電圧が、"Avg"を選択すると平均電流が表示されます。



Starter Kit 上のタッチ・スライダーに触ってみます。そうすると、Energy Profiler 上の消費電流値が増減するのが判ります。



画面上部に表示される数字が、測定時間(s)、トータル平均電流(uA)、トータル平均電力(uW)、トー タルエネルギー(uWh)になります。

<b>17.10</b> s	<b>531.79</b> μW
<b>160.70</b> µA	<b>2.53</b> µWh

TecStar =

スタート/ストップ(Running/Paused)の切り替え、対数/リニア表示の切り替え、X軸・Y軸の拡大 /縮小などの機能が付いていますので、操作性を体感ください。平均電流も表示されています。



波形の上をクリックするとカーソルが現れ、特定期間の電流・電力を測定することもできます。


Energy Profiler を終了するときは、Profiler ⇒ End/Save Session を選択します。 Quick Access からも 選択できます。



◆ ソースコード連携(Code Correlation)

Code Correlation は、電流値とソースコードとを連動させる機能です。消費電流の最適化を行いたい 場合に、どの関数に手を加えるのが効率的か?の手がかりとなります。Cortex-M3/M4Fの製品でサポ ートしている SWO ピンを使用して実現しますので、M0+製品ではご使用になれません。

この機能を使用するには、ソースコード上でSWOピンの設定が必要になります。サンプルコード 「STK3700\_touch」のソースコード(touchdemo.c)では、あらかじめその設定が行われていますので、設 定内容を見てみましょう。

main()関数の中で、

**TecStar** 

BSP\_TraceProfilerSetup();

という関数が呼ばれています。これが SWO の初期化関数です。

```
*****************
386 /*******
     * @brief Main function
387
                            **
388
389⊖ int main(void)
390 {
      /* Chip errata */
391
392
      CHIP Init();
393
      /* If first word of user data page is non-zero, enable Energy Profiler trace */
394
395
      BSP_TraceProfilerSetup();
396
      /* Ensure core frequency has been updated */
397
398
      SystemCoreClockUpdate();
399
400
      /* Initialize <u>capsense</u> */
```

この関数を使用するには、

"bsp\_trace.h"

を include する必要があります。

30	/* Drivers */
31	<pre>#include "vddcheck.h"</pre>
32	<pre>#include "segmentlcd.h"</pre>
33	<pre>#include "caplesense.h"</pre>
24	"#include "stedsives b"
35	<pre>#include "bsp_trace.h"</pre>

実際に動作を見てみます。もう一度、Profile As から Energy Profiler を起動し、Starter Kit のリセットボ タンを押してください。画面下に、関数名と、その関数で消費したエネルギー量が表示されます。

(注: Simplicity Studio 旧バージョンでは、リセットなしに Code Correlation 機能が有効になっていましたので、今後そのように変更される可能性もあります)



### TecStar —

また、画面右上の IRQ 設定を有効にすると、画面下に IRQ 発生タイミングが表示されるようになります。どの IRQ が、いつ発生したかが判りますので、デバッグに非常に役立ちます。

	1 mA		
Θ	100 µА те 2 10 µА		
V EV	ents - J-Link	Silicon Labs (440028385)	
	IRQ LESENSE_IRI RTC_IRI		1101 10111 10111 10 1

### 電流波形上でクリックすると、そこに該当するソースコード箇所にジャンプします。

de ₩₩ Scope View Paused Recording 185.13 μA 136.12 μWh □	<pre>m_lcd.h X 481@/************************************</pre>
Current Voltage Avg IRQ RX/TX	485       * @return         486       * Bit fields for LCD registers which have pending updates         487       ************************************
800.23 s 612.34 μW 💙 10.00 ms 12.10 mW 185.13 μA 136.12 μWh 🛄 3.68 mA 33.60 nWh 🛄	489 { 490 return LCD->SYNCBUSY; 491 } 492
クリック	<pre>4930 /************************************</pre>
	<sup>504</sup> } <b>該当箇所にジャンブ</b> 506 /************************************

### 6-4 ピン設定やペリフェラル設定をしてみる (Hardware Configurator)

Simplicity Studio では、ピン設定やペリフェラル設定を直感的に行うことができるツールを用意しています。ここでは設定ツール(Hardware Configurator)の使用方法をご紹介します。

#### Compatible Tools の中にある Hardware Configurator を起動します。 注: 最新の Simplicity Studio では画面が多少異なっておりますが、流れは同じです。



使用する Boards, Part, SDK や作業フォルダを指定してプロジェクトを作成します。プロジェクトを作成 済みであれば、使用するプロジェクトを選択します。完了すると Hardware Configurator が起動します。

Hardware Configurator の画面中央に2つのタブがあり、Default Mode Port I/O でピン設定を、Default Mode Peripherals でペリフェラル設定を行うことができます。

🗒 EFM32GG990F1024.hwconf 🛙 🗧							
DefaultMode Port I/O: PORTIO							
DefaultMode Port I/O: PORTIO           • • • • • • • • • • • • • • • • • • •							
112-pin BGA, 10x10 - (top view)							
🛿 DefaultMode Port I/O 📲 DefaultMode Peripherals							

Default Mode Port I/O



### ピン配置状況を表示。設定変更したいピンを選択

ピンの使用有無・設定変更

• Default Mode Peripheral

Sile *myProjecthwconf 😫	- 0	E Outline 🕱	- 6	
DefaultMode Peripherals		E = = E DefaultMode		
	_	Peripherals		
	-	Brite Port DO		
		> PortA		
		PortB		
Analog A		PortC		
		🌮 PortD		
		✓ PortE ✓ PortF		
		Properties 🕅 🗰 Periphera	Manning	
		Tudouce 23	100000 C	
		Properties of ADC0		
PONT2 VOMP			. 1	
		ADO 0 Single sample mode   So	an mode	
		Property	Value	
		- ADC Settings		
		Low pace filter mode	None	
		Warmup mode	Shut down after each	
Clocking		ADC Frequency (Hz)	7000000 (0×6ACFC0)	
		Conversion tailgating	Disabled	
		Input Settings		
		Channel 0 (PD0)	Disabled	
		Channel 1 (PD1)	Disabled	
		Channel 2 (PD2)	Disabled	
		Channel 3 (PD3)	Disabled	
		Channel 4 (PD4)	Disabled	
Communications		Channel 5 (PD5)	Disabled	
		Channel 6 (PD6)	Disabled	
		Channel 7 (PD7)	Disabled	
😫 DefaultMode Port I/O 🙀 DefaultMode Peripherals 😒				
~		21		
変更したいペリフェラル	を逞	いない。	ペリフェラルの	動作

注:最新の Simplicity Studio の Hardware Configurator にはバグがあり、

◆ ADC0 を設定してみましょう ADC, コンパレータなど幾つかのペリフェラル設定が行えません。(2018/3 現在)

ADC0 を有効にし、ADC0 の動作モードを設定し、ピン設定を行う、という流れになります。 ADC0 を有効にするには、Default Mode Peripherals に切り替えて、ADC0 にチェックを入れます。 Outline ウィンドウに ADC0 が追加されます。



ADC0 の設定を変更するには、Outline に表示された ADC0 を選択します。Properties ウィンドウに ADC0 の設定画面が表示されますので、動作設定を行います。Single sample mode にして、リファレンス 電圧は 2.5V に、ADC の入力には Channel0(L11 ピン)を使用する設定にしてみます。

🗆 Properties 🕱 📲 Peripheral Mapping 📑 💀 🌣 🖶 E				
Properties of ADC0		Properties of ADC0		
ADC 0 Single sample mode Scan n	node	ADC 0 Single sample mode Scan m	node	
Property	Value	Property	Value	
ADC Settings		Single sample mode settings		
Oversampling	2x	Single sample mode	Enabled	
Low pass filter mode	None	PRS input	None	
Warmup mode	Shut down after each c.	Acquisition time (ADC clock c	1	
ADC Frequency (Hz)	7000000 (0×6ACFC0)	Reference	Internal 2.5V	
Conversion tailgating	Disabled	Resolution	12 bit	
Input Settings		Input	Channel 0 (PD0)	
Channel 0 (PD0)	Enabled	Left-adjust result	Disabled	
Channel 1 (PD1)	Disabled	Continuous conversion	Disabled	
Channel 2 (PD2)	Disabled			
Channel 3 (PD3)	Disabled			
Channel 4 (PD4)	Disabled			
Channel 5 (PD5)	Disabled			
Channel 6 (PD6)	Disabled			
Channel 7 (PD7)	Disabled			

続いてピン設定を行います。Default Mode Port I/O ウィンドウに切り替え、Outline に表示された Port I/O を選択します。Peripheral Mapping ウィンドウにペリフェラル一覧が表示されますので、そこから ADC0 を探して、ピン設定を ADC0 の入力設定に合わせます。(つまり CH0 だけを有効にする)

Properties I Periph	neral Mapping 🔀		E (	🔲 Properties 🏢 Periph	neral Mapping 😒	- 8
DefaultMode : PORT	Ю	~		DefaultMode : PORT	0	
• 0		<u> </u>	-			-
ADC0	СН0	PD0 -	-	ADC0	CH0 PD0	
	CH1	PD1			СН1	
	CH2	PD2			CH2	
	СН3	PD3			СНЗ	
	CH4	PD4			CH4	
	CH5	PD5			СН5	
	CH6	PD6				
	CH7	PD7			CH7	
D BU	STAT		-	D BU		<b>_</b>

設定が完了すると、Default Mode Port I/O ウィンドウに、ADC0 入力(L11ピン)の設定が反映されました。使用中のピンは紫色になります。ピンをクリックすると、そのピンの設定(入出力、プルアップの有無

など)も設定が行えます。

2 Dec 2	🛿 🗱 Peripheral Mapping 🛛 🛃 🛃	
0 (G11) Properties	of PD0	
ビンをクリック9 ると、計和	Value	
	gs	
設定が行える。	mode Disabled	
Pulk	up Disabled	
0) (J11) Cus	tom pin name	
a Bod Res	erve Not reserved	
ADC 入力(CH0)がL11 り (**)		
3.0 PD0		

◆ ピン配置の矛盾を解決(Resolve Pin Conflicts)

ペリフェラルごとに使用可能なピンはある程度決まっています。使用するペリフェラルが増えてくると、 ピン配置が競合してしまい、うまくピン配置が行えない場合が出てきます。それを解消してくれるのが Resolve Pin Conflicts 機能です。競合が生じないピン配置の組み合わせを探してくれます。

機能紹介のために、ピン配置が競合させてみます。LCDとUSART0をイネーブルにしてみてください。 ピン4本分の設定が競合してしまい、ペリフェラル設定とピン設定が、競合状況を示す赤色になります。



このような場合、Resolve Pin Conflict 機能を使います。Outline の Port IO を選択して(或いは Default Mode I/O ウィンドウの上で)右クリックし、Resolve Pin Conflicts を選択します。



解決案を提示してくれますので、OK をクリックすれば問題が生じないピン配置に割り当てなおしてくれます。競合を示す赤色のピンがなくなり、問題が解消されました。

	() *myProject/wconf ()	° 0
	DefaultMode Port I/O: PORTIO	
	· a a a a a a a a a a a a a a a a a a a	(AII) JPTI
		811 HE MACCO
	0         0	
🕿 Resolve Pin Conflicts 🛛 🗶	01         02         03         04         05         05         05         00         010           Max         Max         Max         100001 </th <th></th>	
Choose which modules to affect		(E11) MI
		(11) (11)
Module Change	(a)	
☑ DBG ☑ LCD	(H)         (H) <th>HIT</th>	HIT
USART0 Move from location 0 to 2	(1) (2) (3) (4) (4) (4) (4) (4) (4) (4) (4) (4) (4	JII) RN
All pin conflicts will be resolved!		(ATT) HET
OK   Cancel		(UII) RH

◆ コードに反映させる(Generate Source)

設定が完了したら、Default Mode I/O ウィンドウか、Default Mode Peripheral ウィンドウの上で右クリックし、Generate Source を選択します。ペリフェラル設定、ピン設定を含んだ C コードが生成されます。



TSC-01001

### TecStar =

注: 対応している製品は Gecko, Giant Gecko(SO), Leopard Gecko, Tiny Gecko(SO), Wonder Gecko, Zero

6-5 電池寿命の見積もりをしてみる (energy Aware Battery) Gecko の 6 ファミリのみです。

バッテリー・アプリケーションにおいて、電池寿命の見積もりは非常に重要です。しかし MCU には幾 つもの消費電力モードがあり、多数のペリフェラルがあり、電池寿命の見積もりを複雑にしています。 Simplicity Studio には、電池寿命を簡単に見積もることが可能なツールを用意しています。ここでは、電 池寿命の見積もりツール(energy Aware Battery)の使用方法を紹介します。

電池寿命を見積もるにあたって、MCUの動作を特定する必要があります。ここでは以下の動作を例 に取って、電池寿命を見積もっていきます。通常動作(EM0)で10msの作業をし、Idle(EM2)に移行して 1s 待機し、という動作を繰り返し行う、という設定です。



Compatible Tools の中にある energy Aware Battery を起動します。

o view all available tools, click the 'Tools' button in the main toolbar.		Add/Remove Tools
energyAware Battery Simplicity energyAware Battery is a tool for estimating battery lifetime based on a rough application profile based on energy modes and paripherals in use	Energy Profiler The Simplicity Profiler is a tool developed to let developers quickly visualize the energy consumption in their applications and perform optimizations to reduce power consumption	Hardware Configurator Hardware Configurator is a peripheral, pin and crossbar configuration tool that generates initialization code organized into modes
Migrate Projects	Simplicity IDE	Capacitive Sense Profiler Capacitive Sense Profiler is a tool that supports real-time analysis of sensor data from supported parts.

File メニューから New を選択します。



シミュレーションを行いたいファミリ名を選択し、OK をクリックします。



◆ EM0の期間(ADC 有効、10ms)を登録しましょう
 Add State をクリックします。



Energy ModeをEM0に設定し、期間を10msとします。ペリフェラルはADCを有効にしますので、ADC にチェックを付けます。設定が終わったら、Close をクリックします。

	Z eABattery	
	EM0 - 2.92mA	
	Description State 1 10.00ms	━ 期間を指定
	Energy Mode EM0  1.0με 100.0με 10.0ms 1.0ε 1 m 1 h 1 d 1 y 15 y	
消費電力モード	High Frequency Clocks	
を指定		
	Peripherals	H m + 7
		使用する
	3.000µA 3.000µA 33.000µA	<b>ペリフェラルに</b>
	DAC0         □         DMA         □         EBI         □         12C0         □         12C1           38.000µA         113.880µA         21.840µA         87.500µA         87.500µA         87.500µA	チェック
	LCD         Ε           8.550μA         Ε           150.000nA         Ε           150.000nA         Ε	
	PCNT0         PCNT1         PCNT2         PRS         RTC           1UU.UUUnA         1UU.UUUnA         39.340µA         1UU.UUUnA	
	Close	

なお、ADC の横のボタンをクリックすると、動作モードをより詳細に指定することが可能です。

Sampling Profile	10kS, 12bit, internal reference, Warmup 0b01 💌
I    🔲 I2C	10kS, 12bit, internal reference, Warmup 0b10
A 875000A 875000A	10kS, 12bit, internal reference, Warmup 0b01
	10kS, 12bit, internal reference, Warmup 0b00
	1MS, 12bit, external reference

また、MCU だけではなく、外部部品での消費電流も含めてシミュレーションを行いたい場合には、 External Current(ペリフェラルを下にスクロールしていくとあります)に数値を入力してください。

– External Cu	irrent		_					
📝 Enable e	external co	nponent curi	rent					
	1.00mA							
						[		
_ ·								
1.0nA	10.0nA	100.0nA	1.0µA	10.0µA	100.0µA	1,000.0µA	10.0mA	100.0mA

◆ EM2 の期間(RTC 有効、1s)を登録しましょう

EM0 の時と同じ作業を、EM2 についても行います。 Add State をクリックします。

Energy Mode を EM2 に設定し、期間を 1s とします。ペリフェラルは RTC を有効にしますので、RTC に チェックを付けます。設定が終わったら、Close をクリックします。 登録が完了すると、次図のように表示されます。

TecStar —

# TecStar —

EMO State 1	EM2 State 2	Add state				
•			Current	consumptio	n profile	
10.0mA =			e arrene	oono ampiro	protito	
1,000.0µA 100.0µA						
10.0µA -						
1.0µA						
100.0nA						
10.0nA -						
0.0s	200.0m	a 40	0.0ms	600.0ms	800.0ms	1.0s

### ◆ シミュレーションを行いましょう

使用する電池を決めます。今回は、CR2016を並列に2個繋いだ場合を考えます。Cell type で CR2016を選択し、Number of parallels(並列の電池数)を2、Cells in series(直列の電池数)を1にしま す。

使用する電池を選択	Battery setup							
	Cell type	÷ ()						
電池数(並列)>	Number of parallels 2							
7	Cells in series 1							
電池数(直列) 🦯								

Simulation メニューから、Start Simulation を選択します。

File	Simulation	Help
	🕨 Start Sin	nulation

シミュレーションが実行され、この使用条件の場合には電池寿命が 260 日程度であることが算出できました。



### 6-6 ユーザ基板のプログラミング・デバッグを行ってみる

Starter Kitを使用することで、ユーザ基板上の EFM32 に対して、プログラミング或いはデバッグを行う ことが可能です。その手順について紹介します。



### 6-6-1 参考資料

•AN0042: EFM32 Debug and Trace

•AN0062: Programming Internal Flash Over the Serial Wire Debug Interface

・各ファミリの Starter Kit ユーザガイド

http://community.silabs.com/t5/32-bit-MCU-Knowledge-Base/Using-an-EFM32-Starter-Kit-as-an-exter nal-debugger/ta-p/124194

### 6-6-2 ハードウェア接続

Starter Kit の Debug Connector を介して、ユーザ基板に接続します。 Debug Connector は、 Starter Kit の右上にある 20 ピンのコネクタです。



Debug Connector のピン配置情報は、Starter Kit ユーザガイドに記載されています。下図は、 EFM32GG-STK3700 ユーザガイドからの抜粋です。このうち、SWCLK、SWDIO、SWO、#RESET、 VTARGET、GND の計 6 ピンを使用します。

VTARGET	1			2	NC
#TRST	3			4	GND
TDI	5			6	GND
TMS/SWDIO	7	•		8	GND
TCK/ SWCLK	9	4 -	•	10	GND
RTCK 1	11	ᇰᇰ		12	GND
TDO/ SWO 1	13	•		14	GND
#RESET 1	15	•		16	GND
PD 1	17	•		18	Cable Detect
PD 1	19	•		20	GND

SWCLK	EFM32(ユーザ基板上)の該当ピンに接続してください。直結で結構です。
SWDIO	EFM32(ユーザ基板上)の該当ピンに接続してください。直結で結構です。
SWO	EFM32(ユーザ基板上)の該当ピンに接続してください。直結で結構です。
#RESET	EFM32(ユーザ基板上)の該当ピンに接続してください。直結で結構です。
VTARGET	EFM32(ユーザ基板上)への供給電源に接続してください。Starter Kit とユー
	ザ基板の信号レベルを合せるために使用します。接続し忘れると、Starter
	Kit からユーザ基板を認識できませんので、ご注意ください。
GND	Starter Kit の GND と、ユーザ基板の GND を接続してください。

#### 6-6-3 ハードウェア接続 (例)

ユーザ基板に載っている EFM32をプログラミング・デバッグするには、対象となる EFM32の SWCLK、 SWDIO、SWO、#RESETを、Starter Kitの Debug Connector へ接続する必要があります。また、Starter Kit とユーザ基板の GND も共通化します。Debug Connectorの VTARGET への給電も必要です。

シリコンラボ社のコミュニティフォーラムに接続例がありますので、こちらが参考になるかと思います。 http://community.silabs.com/t5/Simplicity-Studio-and-Software/Using-an-EFM32-Starter-Kit-as-an-ext ernal-debugger/ta-p/138008

♦ SWCLK, SWDIO, SWO, #RESET

これらのピンが、どの Pin Name(ピン名称)に割り当てられているかを確認してみましょう。例として、 ユーザ基板に EFM32GG990 が載っている場合を考えてみます。

まず、データシートの Alternate Functionality Pinout で、SWCLK、SWDIO、SWO がどの Pin Name (ピン名称)に割り当てられているかを確認します。下図のとおり、SWCLK は PF0、SWDIO は PF1 に割り 当てられています。SWO は PF2/PD1/PD2 のいずれかに割り当てが可能となっていますが、デフォルト は Location0 の PF2 になっています。

Alternate			LOC	ATION				
Functionality	0	1	2	3	4	5	6	Description
DBG_SWCLK	PF0	PF0	PF0	PF0				Debug-interface Serial Wire clock input. Note that this function is enabled to pin out of reset, and has a built-in pull down.
DBG_SWDIO	PF1	PF1	PF1	PF1				Debug-interface Serial Wire data input / output. Note that this function is enabled to pin out of reset, and has a built-in pull up.
DBG_SWO	PF2		PD1	PD2				Debug-interface Serial Wire viewer Output. Note that this function is not enabled after reset, and must be enabled by software to be used.

次に、データシートの Pinout で、どの Pin Name (ピン名称)が、どの Pin Number (ピン番号) に割り振られ ているかを確認します。

PF0(SWCLK)はE8ピン、PF1(SWDIO)はD8ピン、PF2(SWO#0)はC8ピン、#RESETはK6ピンに割り当てられていることが判ります。

BGA112 Pin# and Name		Pin Alternate Functionality / Description						
Pin #	Pin Name	Analog	EBI	Timers	Communication	Other		
E8	PF0			TIM0_CC0 #5 LETIM0_OUT0 #2	US1_CLK #2 LEU0_TX #3 I2C0_SDA #5	DBG_SWCLK #0/1/2/3		
D8	PF1			TIM0_CC1 #5 LETIM0_OUT1 #2	US1_CS #2 LEU0_RX #3 I2C0_SCL #5	DBG_SWDIO #0/1/2/3 GPIO_EM4WU3		
<b>C</b> 8	PF2	LCD_SEG0	EBI_ARDY #0/1/2	TIM0_CC2 #5	LEU0_TX #4	ACMP1_O #0 DBG_SWO #0 GPIO_EM4WU4		
K6	RESETn	Reset input, active low. To apply an external reset source to this pin, it is required to only drive this pin low during reset, and let the internal pull-up ensure that reset is released.						

上記を見ると、デバッグピンには、TIM\_CC0#5 や US1\_CLK#2 など、他の機能を割り当てることが可能 になっていることが判ります。デフォルトでは、デバッグ用のピンに設定されていますので、ユーザプログラ ムから設定変更を行うことになります。しかし、特別な理由がない限り、他の機能は割り当てずに、デバッ グ専用ピンとしてご使用になることを強くお勧めします。いったん設定変更を行うと、デバッグ回路にアクセ スできなくなりますので、プログラムの書き替えも行えなくなる可能性があります。

♦ VTARGET

Starter Kit 上のデバッガ(J-Link)と、ユーザ基板上の EFM32 は、Debug Connector を介して接続しま す。しかし、SWCLK、SWDIO、SWO、#RESET は直接繋がる訳ではなく、間にスイッチが入っています。 そのスイッチへの給電(イネーブル信号)に相当するのが VTARGET です。つまり、VTARGET に給電し ないと、デバッガ(J-Link)とユーザ基板上の EFM32 は接続できないことになります。

Starter Kit の Expansion Header から VMCU(3.3V)が出ていますので、VMCU からユーザ基板および VTARGET へ給電するという方法もあります。

### 6-6-4 デバッグ対象の切り替え

デバッグ対象を、Starter Kit 上の EFM32 から、ユーザ基板上の EFM32 に切り替えます。

Device タブで Starter Kit を選択すると、画面右に現在の Debug Mode について表示されます。下図では MCU の設定になっています。

● Device         %         ∴<	EFM32 Giant Gecko Starter Kit
	Preferred SDK: Gecko SDK Suite v2.2.1: Bluetooth 2.8.0.0, EmberZNet 6.2.1.0, F
	Debug Mode: MCU <u>Change</u> Adapter Firmware Version: 0v14p3b159 <u>Change</u>

Adapter Configuration タブの Debug Mode で、OUT を選択します。

💥 J-Link Silicon Labs (440028385)	
Configuration of device: J-Link Silicon Labs (440028385)	
Application images Scratchpad Packet Trace Adapter Configuration J-Link Configuration	Device hardware
Update Adapter Installation Package	
Browse	Install Package
Debug Mode: MCU V OFF MCU IN OUT	
ОК	Cancel

デバッグ対象が Starter Kit 外部に切り替わると、Starter Kit 右上の DEBUG OUT という LED が点灯 します。



次に、Device hardware タブの Target part で、ユーザ基板上で使用している型番を選択し、OK をクリックします。Simplicity Studio の旧バージョンでは型番を自動認識してくれていましたが、Simplicity Studio ver.4 では自動認識しなくなっていますので、ご注意ください。

💥 J-Link Silicon Labs (44002	8385)	- • ×
Configuration of device: J	-Link Silicon Labs (440028385)	
Application images Scrat	chpad Packet Trace Adapter Configuration J-Link Configuration Devic	ce hardware
All detected information:		
Parts:		
Boards: EFM32 Giant Gecko St Target part: TG840	arter Kit board (BR52200A Rev A03)	•
EFM32TG840F16 EFM32TG840F32 EFM32TG840F8 C Target Interface: SWE	atible with target part  Compatible with the selected target part  OK Ca	ncel

登録が完了すると、Device タブにユーザ基板上の EFM32 が追加されます。あとは、Starter Kit 上の EFM32 と同様に使用できます。

Device     Solution     S	EFM32TG840F32 Preferred SDK: Gecko SDK Suite v2.2.1: Blueto
	Debug Mode: OUT Change Adapter Firmware Version: 0v14p3b159 Change

デバッグ対象を Starter Kit 上の EFM32 に戻す場合には、Adapter Configuration タブの Debug Mode で、MCU を選択し、Device hardware タブの Target part で Starter Kit 上で使用している型番を選択して ください。MCU に切り替えただけでは自動認識しませんので、ご注意ください。

なお、Debug Mode の MCU、IN、OUT の違いは以下の通りです。



### 6-7 ユーザ基板の消費電流を測定する

ユーザ基板の消費電流を測定するためには、次の2つの処理を行う必要があります。

- ◆ ユーザ基板には、Starter Kit の VMCU ピンから給電を行う
- ◆ Starter Kit 上の部品(EFM32, その他周辺部品)の消費電流を極限まで下げる

Starter Kit には、電流モニタ用の回路が実装されています。下図は STK3700(Giant Gecko)のユーザ ガイドからの抜粋ですが、Debug 用 USB から 5V 給電を受け、LDO で 3.3V を作り、それを VMCU とし て Starter Kit 上の EFM32 やセンサ、ペリフェラルに給電しています。 Energy Profiler は VMCU の消費 電流をモニタするようになっています。



### 6-7-1 VMCU をどこから取るか

Energy Profiler でユーザ基板の電流を測定するには、VMCU からユーザ基板へ給電する必要があります。

VMCU は Starter Kit の Expansion Header から出ています。Expansion Header は、Starter Kit の右側 にある 20 ピンのコネクタです。Expansion Header のピン配置情報は、Starter Kit ユーザガイドに記載さ れていますが、2 ピンに VMCU が配置されています。



#### 6-7-2 Starter Kit 上の部品の消費電流を極限まで下げる

ユーザ基板だけでなく、Starter Kit 上の一部部品も VMCU から給電を受けています。つまり、Energy Profiler で読める電流値はユーザ基板と Starter Kit の一部部品の消費電流の合算値です。そのため、 ユーザ基板の消費電流を正確に測定するために、Starter Kit 側の消費電流値を極限まで下げる必要が あります。

Starter Kit 上の EFM32 は、EM4(消費電流がもっとも低い電力モード)に移行することで消費電流を 極限まで下げることができます。また EFM32 が EM4 にいる間は、センサやペリフェラルはほとんど電流 を消費しません。ですから EFM32 を EM4 にすれば良い事になります。

電力モードを切り替えるデモコードがありますので、それを利用します。デモコードのダウンロード手 順は以下の通りです。

1. Starter Kit を接続し、Getting Started タブの Demo 横の View All を選択します。

Getting Started	Docume	entation	Compatible Tools	Resources	
Demos	View All	Software Examples	View All	Document Index	Customize
STK3700 biometric	€	STK3700_biometric	₿Ę.	Gecko SDK Suite: Bluetooth 2.3.0.0, Er Flex 1.1.0.0, MCU 5.1.2, Thread 2.2.0.0	mberZNet 5.9.0.0, ) v1.0.0
This example project uses the EFM32 CMSIS including emlib and the board support package	2	This example project uses the Ef including emlib and the board su	M32 CMSIS pport package	Release Notes	

2. 実行可能なデモの一覧が表示されますので、STK3700 emlcd を選択し、Mode を Run に変更して Start を押します。

Demos for EFM32 Glant Gecko Starter Kit board (BF Select Demo Select a demo and the mode with which to run it.	3D2200A Rev A03)	×
Name	Description	•
STK3700 burtc	Backup power domain RTC example.	=
STK3700 clock	Wall Clock example using the segment	
STK3700 cpt007b_demo	CPT007B Demo	
STK3700 cslib	CSLIB example. Template for new proje	
STK3700 emlcd	Energy Modes with segment LCD exam	
STK3700 emode	Select a single energy mode, and stay t	Τ.
STK3700 energy	Board Support Package API demo for vo	
STK3700 freertos alink	FreeRTOS - Blink example	4
This example project use EFM32 CMSIS and th demonstrate the Jse of the LCD controller, various Energy Nodes (EM) on the starter k: EM2 is used for delays in between refreshin "eye condy" ard present to show off the MC	e emlib peripheral library to RTC (real time counter), GPIO and t. Ig the LCD display, and a lot of J module's LCD display.	H
Mode: Run	to LM2. In the demo, LM2 19 Used	
⑦ ↓	Start Cancel	

このデモは、ボタンを押すと電力モードを順々に切り替えていきます。STK3700の場合は、PB0を 押すとEM0⇒EM1⇒EM2⇒EM3の順に切り替えていき、PB1を押すとEM0⇒EM1⇒EM2⇒EM4の 順に切り替えていきます。今回は EM4 にしたいですので PB1を押すことになります。LCD に動作説 明が表示されますので、そちらも参考にしてください。

なお、Mode で Run under Energy Profiler を選択してから Start すると、モード変化に伴って電流値が 十分に下がっていることをご確認頂けます。



### 6-7-3 ユーザ基板の消費電流の測定手順

手順は以下の通りです。

- 1. 「6-7-2 Starter Kit 上の部品の消費電流を極限まで下げる」を参考に、Starter Kit 上の EFM32 に デモソフトを書き込みます。(Debug Mode は MCU)
- 「6-6 ユーザ基板のプログラミング・デバッグを行ってみる」を参考に、デバッグ経路をユーザ基板 に接続します。また「6-7-1 VMCU をどこから取るか」を参考に、VMCU もユーザ基板に接続しま す。
- 3. Debug Mode を Out に変更し、ユーザ基板の EFM32 に測定時に使用したいプログラムを書き込みます。
- Starter Kit の RESET ボタンを押し、Starter Kit 上の EFM32 のデモソフトを実行します。「6-7-2 Starter Kit 上の部品の消費電流を極限まで下げる」を参考に、PB1 を押して Starter Kit 上の EFM32 を EM4 に移行させます。
- 5. Energy Profiler を起動し、Run メニューから Profile Running Program を選択します。ユーザ基板の 消費電流がモニタできます。



7 ソフトウェア設計

ソフトウェア設計に役立つ情報をご紹介します。

7-1 Cortex-M を初めて使う方に (ARM 社ドキュメント)

ARM 社が、Cortex-M そのものに関する日本語ドキュメントを提供しております。ARM 社の Web Site からご入手頂くことができます。

◆ Cortex-M4 テクニカル・リファレンスマニュアル

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439cj/DDI0439CJ\_cortex\_m4\_r0p1\_trm\_ja-JP.pdf

◆ Cortex-M3 テクニカル・リファレンスマニュアル

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337ij/DDI0337IJ\_cortexm3\_r2p1\_trm\_japanese.pdf

◆ Cortex-M0+ テクニカル・リファレンスマニュアル

<u>http://infocenter.arm.com/help/topic/com.arm.doc.ddi0484cj/DDI0484CJ\_cortex\_m0p\_r0p0\_trm\_JP.pdf</u>
◆ Cortex-M0+ デバイス・ジェネリックユーザ・ガイド

http://infocenter.arm.com/help/topic/com.arm.doc.dui0662bj/DUI0662BJ cortex m0p r0p1 dgug JP.pdf

また、役立ちそうな日本語アプリケーションノートもご紹介します。

◆ Cortex-M3 組み込みソフトウェア開発(アプリケーションノート 179)

http://infocenter.arm.com/help/topic/com.arm.doc.dai0179bj/DAI0179BJ.pdf

◆ PIC マイコンから Cortex-M3 への置き換え(アプリケーションノート 234)

http://infocenter.arm.com/help/topic/com.arm.doc.dai0234a/DAI0234A\_migrating\_from\_pic\_to\_m3.pdf

◆ 8051 マイコンから Cortex-M への置き換え(アプリケーションノート 237)

http://infocenter.arm.com/help/topic/com.arm.doc.dai0237a/DAI0237A\_migrating\_from\_8051\_to\_Cortex\_M.pdf

# TecStar —

### 7-2 開発用のサンプルコードやライブラリについて

サンプルコードは、Getting Started  $\Rightarrow$  Software Examples もしくは、Documentation  $\Rightarrow$  Application Notes からご入手頂けます。

Getting Started	Documentation		Compatible Tools	Resources	
Demos	-+ 🗹 🚍	Software Examples	; → Ξ	SDK Documentation	-+ 🗹 🗏
<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>Demos</li> </ul>		<ul> <li>32-bit MCU SDK 5.4.0</li> <li>Giant Gecko 3700</li> </ul>	.0 Starter Kit	<ul> <li>32-bit MCU SDK 5.4.0.0</li> <li>API References</li> <li>Release Notes</li> </ul>	
Getting Started	Docume	entation	Compatible Tool	s Resources	
My Favorite Documents		-+	All Documents		-+ 🗹
No documents have been favorited. Click the 'Favorite' icon to add a document have.		Gecko SDK Suite v2.2.1: Bl 5.4.0.0 EFM32 Giant Gecko Starter • API References • Application Notes AN0002.0: EFM32 and This application note de and EZR32 Wireless MC AN0002.1: EFM32 and	uetooth 2.8.0.0, EmberZNet 6.2.1.0, Flex 2. Kit (EFM32GG-STK3700) d EZR32 Wireless MCU Series 0 Hardwar etails hardware design considerations for EF CU Series 0 devices. For hardware design d EFR32 Wireless Gecko Series 1 Hardward	2.1.0, MCU re Desi 📋 🏠 M32	

- ◆ Software Example: 1つの、或いは複数の機能を組み合わせた応用的なサンプルコードを提供 しています。
- ◆ Application Notes: アプリケーションノートには、特定の機能(各種ペリフェラルや割り込みなど) に特化した情報が記載されています。多くのアプリケーションノートには、サンプルコードが付属 しています。

ライブラリに関する情報は、「3-4 EFM32 の API 情報」を参照ください。基本的には、EMLIB および EMDRV を使用してソフト設計を行って頂くことになります。

EFM32 Giant Gecko Software Documentation				
Main Page Modules	Files	Documentation Home silabs.com		
EFM32 Giant Gecko Software Docume     Modules	<sup>tatic</sup> Module	25		
► Files	Here is a lis	t of all modules:		
Documentation Home silabs.com	BSP	Board Support Package		
	<ul> <li>Devices</li> </ul>	Silicon Laboratories CMSIS-CORE device headers		
	• EMDRV	EnergyAware drivers		
	• EMLIB	Low-level peripheral library		
	Kit Drive	rs Kit support and drivers		
	► USB	USB HOST and DEVICE protocol stacks.		

◆ EMLIB: EFM32 で共通に持つ各種機能の API を提供しています。UART, I2C, タイマ, RTC, GPIO などが用意されています。

# TecStar —

- ◆ EMDRV: EFM32 ライブラリ以外の機能の API を提供しています。内蔵 Flash、スリープモード、 SPI などが用意されています。
- ◆ Devices: CMSIS(Cortex Microcontroller System Interface Standard)用の API です。
- ◆ BSP、Kit Driver: 評価基板上で使用されている LED、センサ、液晶機能などを簡易に使用する ために用意されたライブラリです。

### 7-3 ソースコードの追い方

Simplicity IDE でソースコードを追うための方法を紹介します。

◆ 変数や関数を定義している記述を探す



③変数や関数の定義にジャンプ

### 7-4 割り込みハンドラ

EFM32の割り込みコントローラは、Cortex-M系コアに内包されるNVICいうブロックに含まれており、市場にある多くのCortex-Mコア搭載品と動作に差がありません。

EFM32 のリファレンス・マニュアルの NVIC の関連事項のほか、ARM 社のドキュメント内の NVIC の記述が参照頂けます。

参考) 日本語の Web サイトとしては、APS ACADEMY の Cortex-M 入門編 第 13 回で、非常に判り やすく説明されています。「APS 例外/割り込み処理」で検索してみてください。

7-5 ピン設定、ペリフェラル設定の流れ

ピン設定は、Configuratorを使用して行うのが簡単です。1 つのピンを複数のペリフェラルで共有し ている場合も多いですので、競合が生じないような設定を、直感的に行うことができます。ソースコード も自動生成されます。

ペリフェラル設定は、リファレンス・マニュアルとアプリケーションノート(およびソースコード)を参考に、 行って頂くことになります。

### 7-6 ピン設定

EFM32GG990を使用し、①USARTO (Asynchronous mode=UART)、②I2CO、③GPIO1 本を使用した場合の設定手順をご紹介します。もちろんゼロからコードを書くこともできますが、Hardware Configurator を使用するのが簡単です。

- 1. Hardware Configurator をクリックし、「6-4 ピン設定やペリフェラル設定してみる」と同じ手順で プロジェクトを作成します。
- Hardware Configurator が起動したら、Default Mode Peripherals でペリフェラル設定をしていき ます。まず①USART0の設定を行います。USART0にチェックを入れ、Outlineに表示された USART0を選択し、Properties ウィンドウにて USART0の設定を行います。ここでボーレートなど の設定を行うことができます。



3. ②I2C0の設定を行います。Default Mode Peripherals でI2C0 にチェックを入れ、Outline に表示さ れた I2C0 を選択し、Properties ウィンドウにて I2C0 の設定を行います。

	🔲 Properties 🗙 📜 🇮 Peripheral	Mapping 🛛 😁 🐨 🗖 🗖
	Properties of I2C0	
	I2C 0	
DefaultMode Peripherals	Property	Value
	Settings	
	Enable	Enable I2C after init
Communications *	Master/Slave	Master mode
	Low/High ratio	4:4 clock ratio
	I2C max bus frequency	1 MHz (fast mode +) 🗸
		10 kHz (low speed)
		100 kHz (standard)
UARTO UARTI <b>USARTO</b> USARTI USART2		400 kHz (fast mode)
		T Mile (last mode 17

4. ③GPIO の設定を行います。Default Mode Peripherals で GPIO にチェックを入れます。

DefaultMode Peripherals	
Other .	*

 Default Mode Port IO に移動して、①USART0 からピン設定を行っていきます。Outline で Port IO を選択し、Peripheral Mapping ウィンドウで USART0 にチェックし、使用するピン(CLK, CS, RX, TX)にチェックします。すると、Default Mode Port IO にピン配置が反映されます。次図の場合で は、パッケージ左上の4本ピンに USART0 ピンが割り振られました。よく見ると、ピン下の信号名 が赤字で表示されています。これはピン設定がまだ完了していないことを示しています。



6. ピン設定を行っていきます。例として USART0\_RX ピンをダブルクリックします。Property ウィンド ウにピンの設定画面が表示されますが、Pin mode に の表示があります。これが修正すべき 項目になります。Pin mode を選択して黒く反転させると、画面下に赤字で、どのように設定変更 すれば良いか?のアドバイスが表示されます。

Ĭ	🗖 Properties 🕱 🛛 🏢 Peripheral Mapping	🛃 🖪 🗸 🗖
	Properties of PE11	
	Port Pin	
	Property	Value
	Settings	
	Pin mode	🔞 Disabled
	Pullup	Disabled
	Custom pin name	
<b></b>	Reserve	Not reserved
	- T	
ect) • USART0_RX (PE11) pin mode must be Inpu	t	

USART0\_RX の Pin mode を Input にするように指示されていますので、その通りに設定すると、 Default Mode Port IO の信号名 (USART0\_RX) が赤字から黒字に変わりました。 同じ要領で、他 の 3 つのピンの Pin mode を push-pull に変更します。



 ②I2C0 のピン設定を行います。Outline で Port IO を選択し、Peripheral Mapping ウィンドウで I2C0 にチェックし、使用するピン(SCL, SDA)にチェックします。Default Mode Port IO のピン下の 信号名を見ると赤字で表示されていますので、まだ設定が完了していません。USART0と同じ要 領で、2 つのピンの Pin mode を Wired-and pullup filter に変更します。



 ③GPIO のピン設定を行います。今回は、A1(PE15)に push pull 出力を割り当ててみます。
 Default Mode Port IO にて、A1(PE15)をダブルクリックします。Property ウィンドウにピンの設定 画面で Pin mode を Push-pull に変更し、Custom pin name に信号名(下図では TEST)を入力しま す。Default Mode Port IO に反映されました。

main: 🔅 *myProjecthwcont 🕴		🗖 maino 🛛 🗐 🛤	rojecthwconf 83
DefaultMode Port I/O: PORTIO		DefaultMode Por	rt I/O: PORTIO
A1         A2         A3           PEIS         PEIA         PEI2           USARTO,OLK         B3         PEI3           PAIS         PEI3         B3           PAIS         PEI3         PEI3           PAIS         PEI3         B3           PAIS         PA0         B10           ECOSOL         ECOSOL         USARTO,TX	Properties & ## Peripheral Mapping      Properties of PE15      Port Pin      Property     Value     Settings     Pin mode     Push-pull     Data output     Test     Reserve     Not reserved	A1 PEIS (TEST) B1 PATS C1 PAA ECO.SOL	A2         A3           PE14         PE12           USARTO,CLK         B3           PE03         PE01           LEANTO,CS         USARTO,RX           C2         C3           PE00         PE00           PE00         C3           PE00         PE00           PE00         PE00           PE00         PE00

9. 設定が完了したら、Default Mode I/O ウィンドウか、Default Mode Peripheral ウィンドウの上で右 クリックし、Generate Sourceを選択します。ペリフェラル設定、ピン設定を含んだCコードが生成さ れます。

左下図はプロジェクトツリーですが、InitDevice.c に行った設定が反映されています。

Enter\_DefaultMode\_from\_RESET という関数の中で、USART0, I2C0, PortI/O の初期化関数を呼んでいます。それぞれの初期化関数も参考のために掲載します。



Hardware Configurator が生成したコードをベースに、リファレンス・マニュアルとアプリケーションノート (およびサンプルコード)を参考に、必要に応じて追記・修正を行ってください。

<InitDevice.c の一部>

```
@ //-----
// enter_DefaultMode_from_RESET
//------
@ extern void enter_DefaultMode_from_RESET(void) {
    // $[Config Calls]
    CMU_enter_DefaultMode_from_RESET();
    USART0_enter_DefaultMode_from_RESET();
    I2C0_enter_DefaultMode_from_RESET();
    PORTIO_enter_DefaultMode_from_RESET();
    // [Config Calls]$
  }
}
```

```
⊜//-----
 // USART0_enter_DefaultMode_from_RESET
 extern void USART0_enter_DefaultMode_from_RESET(void) {
    // $[USART_InitAsync]
    USART_InitAsync_TypeDef initasync = USART_INITASYNC_DEFAULT;
    initasync.baudrate = 115200;
    initasync.databits = usartDatabits8;
    initasync.parity = usartNoParity;
    initasync.stopbits = usartStopbits1;
    initasync.oversampling = usartOVS16;
 #if defined( USART_INPUT_RXPRS ) && defined( USART_CTRL_MVDIS )
    initasync.mvdis = 0;
    initasync.prsRxEnable = 0;
    initasync.prsRxCh = 0;
 #endif
    USART_InitAsync(USART0, &initasync);
    // [USART_InitAsync]$
    // $[USART InitSync]
    // [USART_InitSync]$
    // $[USART_InitPrsTrigger]
    USART_PrsTriggerInit_TypeDef initprs = USART_INITPRSTRIGGER_DEFAULT;
    initprs.rxTriggerEnable = 0;
    initprs.txTriggerEnable = 0;
    initprs.prsTriggerChannel = usartPrsTriggerCh0;
    USART_InitPrsTrigger(USART0, &initprs);
    // [USART_InitPrsTrigger]$
 }
// I2C0_enter_DefaultMode_from_RESET
 //-----
extern void I2C0_enter_DefaultMode_from_RESET(void) {
    // $[I2C0 initialization]
    I2C_Init_TypeDef init = I2C_INIT_DEFAULT;
    init.enable = 1;
    init.master = 1;
    init.freq = I2C_FREQ_FAST_MAX;
init.clhr = i2cClockHLRAsymetric;
    I2C Init(I2C0, &init);
    // [I2C0 initialization]$
 }
```

### TSC-01001

### **TecStar**

```
// PORTIO_enter_DefaultMode_from_RESET
 //=====
                                    _____
extern void PORTIO_enter_DefaultMode_from_RESET(void) {
     // $[Port A Configuration]
     /* Pin PA1 is configured to Open-drain with pull-up and filter */
     GPIO->P[0].MODEL = (GPIO->P[0].MODEL & ~ GPIO_P_MODEL_MODE1_MASK)
            GPIO_P_MODEL_MODE1_WIREDANDPULLUPFILTER;
     // [Port A Configuration]$
     // $[Port B Configuration]
     // [Port B Configuration]$
     // $[Port C Configuration]
     // [Port C Configuration]$
     // $[Port D Configuration]
     // [Port D Configuration]$
     // $[Port E Configuration]
     /* Pin PE10 is configured to Push-pull */
     GPIO->P[4].MODEH = (GPIO->P[4].MODEH & ~_GPIO_P_MODEH_MODE10_MASK)
            GPIO_P_MODEH_MODE10_PUSHPULL;
     /* Pin PE11 is configured to Input enabled */
     GPIO->P[4].MODEH = (GPIO->P[4].MODEH & ~_GPIO_P_MODEH_MODE11_MASK)
            | GPIO_P_MODEH_MODE11_INPUT;
     /* Pin PE12 is configured to Push-pull */
     GPIO->P[4].MODEH = (GPIO->P[4].MODEH & ~_GPIO_P_MODEH_MODE12_MASK)
            GPIO P MODEH MODE12 PUSHPULL;
     /* Pin PE13 is configured to Push-pull */
     GPIO->P[4].MODEH = (GPIO->P[4].MODEH & ~_GPIO_P_MODEH_MODE13_MASK)
            GPIO_P_MODEH_MODE13_PUSHPULL;
     /* Pin PE15 is configured to Push-pull */
     GPIO->P[4].MODEH = (GPIO->P[4].MODEH & ~ GPIO P MODEH MODE15 MASK)
            GPIO_P_MODEH_MODE15_PUSHPULL;
     // [Port E Configuration]$
     // $[Port F Configuration]
// [Port F Configuration]$
    // $[Route Configuration]
    /* Enable signals SCL, SDA */
    I2C0->ROUTE |= I2C_ROUTE_SCLPEN | I2C_ROUTE_SDAPEN;
    /* Enable signals CLK, CS, RX, TX */
    USART0->ROUTE |= USART_ROUTE_CLKPEN | USART_ROUTE_CSPEN | USART_ROUTE_RXPEN
           USART_ROUTE_TXPEN;
    // [Route Configuration]$
}
```

### TecStar -

### 7-7 ペリフェラル設定

ペリフェラル設定の流れをご紹介します。

#### 7-7-1 USART (Asynchronous mode)

◆ 参考にするソースコード

AN0045「USART or UART Asynchronous mode」にサンプルコードが付属しています。Documentation タブの All Documents から参照したいアプリケーションノートを選択します。一覧に表示されない場合に は Filter by selected product line のチェックを外してみてください。ご使用になる型番以外のサンプルコー ドも表示されます。ご参考にはなると思います。

Open をクリックすると、アプリケーションノート(pdf)が開きます。Open Folder をクリックすると、アプリ ケーションノートやサンプルコードが格納されたフォルダが開きます。Import Project をクリックすると、サ ンプルコードを使って新規プロジェクトを作成します。Import Project を使うと、Simplicity IDE からソース コードが開けますので、ソースコードを追うのが楽になります。



### ◆ ヘッダファイル

"em\_gpio.h" と"em\_usart.h"を include してください。

◆ 初期化

以下は、初期化の例です。 void usart1\_init(void) 任意の関数名 { USART\_InitAsync\_TypeDef uart\_init = { usartEnable, // Enable RX/TX when init completed 0, // Use current configured reference clock for configuring baudrate 115200, // 115200 bits/s usartOVS16, // 16x oversampling usartDatabits8, // 8 databits usartNoParity, // No parity usartStopbits1, // 1 stopbit false, // Not USART PRS input mode usartPrsRxCh0 // PRS channel 0 }; // Using "TX" and "RX" GPIO PinModeSet(gpioPortD, 7, gpioModePushPull, 1); GPIO PinModeSet(gpioPortD, 6, gpioModeInput, 0); // Configure USART for basic async operation uart init enable = usartDisable; USART IntClear(USART1, &uart\_init); // Prepare UART Rx and Tx interrupts USART IntClear(USART1, USART IFC MASK); USART IntEnable(USART1, USART IFC MASK); NVIC ClearPendingIRQ(USART1 TX IRQn); NVIC TenableIRQ(USART1 TX IRQn); NVIC TenableIRQ(USART1 RX IRQn); // Finally enable it USART T\_Enable(USART1, usartEnable); }

#### ◆ 割り込み処理

割り込みハンドラは"system efm32gg.h"(EFM32GGの場合)でプロトタイプ宣言されています。

¥¥SiliconLabs¥SimplicityStudio¥v4¥developer¥sdks¥exx32¥v5.0.0.0¥platform¥Device¥SiliconLabs¥EFM32GG¥Include

#### 以下は、割り込み処理の例です。



◆ 制御方法

"em\_usart.h"の API を使用して制御します。API ドキュメントの閲覧方法については、7-2「開発用のソースコードについて」をご参照ください。



7-7-2 I2C

◆ 参考にするソースコード

AN0011「I2C Master and Slave Operation」にサンプルコードが付属しています。Documentation タブの All Documents から参照したいアプリケーションノートを選択します。一覧に表示されない場合には Filter by selected product line のチェックを外してみてください。ご使用になる型番以外のサンプルコードも表示 されます。ご参考にはなると思います。

Open をクリックすると、アプリケーションノート(pdf)が開きます。Open Folder をクリックすると、アプリ ケーションノートやサンプルコードが格納されたフォルダが開きます。Import Project をクリックすると、サ ンプルコードを使って新規プロジェクトを作成します。Import Project を使うと、Simplicity IDE からソース コードが開けますので、ソースコードを追うのが楽になります。



#### 注: 最新の Simplicity Studio では画面が多少異なっておりますが、流れは同じです。

### ◆ ヘッダファイル

"em\_gpio.h" と"em\_i2c.h"を include してください。

◆ 初期化

```
以下は、初期化の例です。
void i2c0 init(void)
                                      任意の関数名
ł
       // default settings is 100KHz
       I2C_Init_TypeDef i2cInit = {
              true,
                                                       // Enable when init done
              true,
                                                       // Set to master mode
              // Use currently configured reference clock
I2C FREQ STANDARD MAX, // Set to standard rate assuring being within I2C spec
i2cClockHLRStandard, // Set to use 4:4 low/high duty cycle
       };
       // Using "SDA" and "SCL" pin
GPIO_PinModeSet(gpioPortE, 12, gpioModeWiredAndPullUpFilter, 1);
GPIO_PinModeSet(gpioPortE, 13, gpioModeWiredAndPullUpFilter, 1);
      // Initializing the I2C
I2C Init(I2C0, &i2cInit);
      // Clear and enable interrupt
NVIC_ClearPendingIRQ(12C0_IRQn);
NVIC_EnableIRQ(12C0_IRQn);
       I2C Enable(I2C0, 1);
```

```
}
```

#### ◆ 割り込み処理

割り込みハンドラは"system\_efm32gg.h" (EFM32GG の場合)でプロトタイプ宣言されています。

以下は、割り込み処理の例です。

```
void I2C0_IRQHandler (void) 割り込みハンドラ
{
// 独自の処理
NVIC_DisableIRQ(I2C0_IRQn);
}
```

```
◆ 制御方法
```

"em\_i2c.h"の APIを使用して制御します。APIドキュメントの閲覧方法については、7-2「開発用のソースコードについて」をご参照ください。

```
以下は、I2C ライト、I2C リードの例になります。

int i2c_write(I2C_TypeDef *i2c, uint8_t addr, uint8_t wval) I2C ライトの制御例

{

    uint8 t wdata[1];

    I2C TransferSeq TypeDef seq;

    I2C_TransferReturn_TypeDef I2C_Status;

    seq.addr = addr;

    seq.flags = I2C_FLAG_WRITE_WRITE;

    // write buffer setting

    wdata[0] = (uint8_t) wval;

    seq.buf[0].len = 1;

    // Do a polled transfer

    I2C_Status = I2C_TransferInit(i2c, &seq);

    while (I2C_Status == i2cTransferInProgress) {

        I2C_Status = I2C_Transfer(i2c);

        }

        return(I2C_Status);

    }
```

int i2c\_read(I2C\_TypeDef \*i2c, uint8\_t addr, uint8\_t \*p\_rval) I2Cリードの制御例

uint8 t rdata[1] = {0}; I2C TransferSeq\_TypeDef seq; I2C<sup>\_</sup>TransferReturn\_TypeDef I2C\_Status; seq.addr = addr;

ł

seq.flags = I2C\_FLAG\_READ;

// register setting
seq.buf[0].data = rdata;
seq.buf[0].len = 1;

**TecStar** 

}

// Do a polled transfer I2C\_Status = I2C\_TransferInit(i2c, &seq); while (I2C\_Status == i2cTransferInProgress) { I2C\_Status = I2C\_Transfer(i2c); }

}
\*p\_rval = rdata[0];

return (I2C\_Status);

# Main Page Modules Files Documentation Home silabs.com Oc Search <td

▶ COMMON	
▶ CORE	Datailed Description
▶ DAC	
▶ DBG	Inter-integrated Circuit (I2C) Peripheral API.
▶ DMA	This module contains functions to control the I2C peripheral of Silicon Labs 32-bit MCUs and SoCs. The I2C Interface allows communication on I2C buses with the lowest energy
► EBI	consumption possible.
► EMU	Data Structures
► GPIO	struct I2C Init TypeDef
	struct I2C_TransferSeq_TypeDef
▶ INT	Master mode transfer message structure used to define a complete I2C transfer sequence (from start to stop). More
► LCD	Macros

### 7-7-3 タイマ

◆ 参考にするソースコード

AN0014「TIMER」にサンプルコードが付属しています。Documentation タブの All Documents から参照したいアプリケーションノートを選択します。一覧に表示されない場合には Filter by selected product line のチェックを外してみてください。ご使用になる型番以外のサンプルコードも表示されます。ご参考にはなると思います。

Open をクリックすると、アプリケーションノート(pdf)が開きます。Open Folder をクリックすると、アプリ ケーションノートやサンプルコードが格納されたフォルダが開きます。Import Project をクリックすると、サ ンプルコードを使って新規プロジェクトを作成します。Import Project を使うと、Simplicity IDE からソース コードが開けますので、ソースコードを追うのが楽になります。



◆ ヘッダファイル

"em\_timer.h"を include してください。

◆ 初期化

以下は、初期化の例です。

```
static void start timer1(void)
                                    任意の関数名
                                                                            enable
                                                                                         = true,
ł
                                                                            .debugRun
                                                                                        = false,
= timerPrescale1024,
= timerClkSelHFPerClk,
     // Enable clock for TIMER1 module
                                                                            .prescale
.clkSel
     CMU_ClockEnable(cmuClock_TIMER1, true);
                                                            初期設定
                                                                            .fallAction = timerInputActionNone,
     // Select TIMER1 parameters
                                                                            .riseAction = timerInputActionNone,
     TIMER Init TypeDef timerInit = TIMER INIT DEFAULT;
                                                                            .mode
.dmaClrAct
                                                                                          = timerModeUp,
                                                                                         = false,
     // Enable overflow interrupt
TIMER_IntEnable(TIMER1, TIMER_IF_OF);
                                                                            .quadModeX4 = false,
                                                                                         = false,
                                                                            .oneShot
                                                                                         = false,
                                                                            .sync
};
    // Enable TIMER1 interrupt vector in NVIC NVIC_EnableIRQ(TIMER1_IRQn);
     // Set TIMER1 Top value
TIMER_TopSet(TIMER1, CMU_ClockFreqGet(cmuClock_HFPER) / 1000);
      / Configure TIMER1
     TIMER_Init(TIMER1, &timerInit);
}
    割り込み処理
割り込みハンドラは"system efm32gg.h" (EFM32GG の場合)でプロトタイプ宣言されています。
```

以下は、割り込み処理の例です。

void TIMER1\_IRQHandler(void) 割り込みハンドラ

// 独自の処理

// Clear flag for TIMER1 overflow interrupt TIMER\_IntClear(TIMER1, TIMER\_IF\_OF);

### ◆ 制御方法

}

"em\_timer.h"の APIを使用して制御します。APIドキュメントの閲覧方法については、7-2「開発用のソースコードについて」をご参照ください。


### 7-7-4 タイマ (X 秒タイマの作り方)

**TecStar** 

TIMER\_Init\_TypeDef 型のメンバーである".prescale"と".clkSel"と、"TIMER\_TopSet()"の値を変えて 任意の時間のタイマを作ります。

AN0014 の"main\_timer\_up\_count.c"では2秒タイマを作っており、これを例に説明します。

タイマへ供給されるクロックは、下図のような構成になっています。(AN0014 Fig.2.1)



"main\_timer\_up\_count.c"では、".prescale"と".clkSel"は以下の設定になっています。

TIMER Init TypeDef timerInit = .enable = true, .prescale で 1024 分の 1 を選択 = true, = timerPrescale1024 .debugRun prescale. = timerClkSelHFPerClk, clkSel .fallAction = timerInputActionNone, .riseAction = timerInputActionNone, .mode .dmaClrAct = timerModeUp, = false, .clkSel で HFPERCLK(14MHz)を選択 .quadModeX4 = false, .oneShot = false, = false, .sync };

つまりタイマへの供給クロックは、14 MHz(周波数) ÷ 1024(prescale) = 13671 Hz となります。 2 秒タイマを作るには、13671 Hz × 2 = 27342 のカウントが必要です。"TIMER\_TopSet()"を使 って、タイマの上限値を 27342 に設定します。



## 7-7-5 CMU (ペリフェラル・クロックの周波数)

TecStar =

ペリフェラルへの供給クロックの周波数を、どのように設定するか紹介します。

リファレンスマニュアルのCMU - Clock Management Unitの章に、CMU Overview という図があります。 この図で、EFM32 のクロックツリーを確認する事ができます。下図は CMU Overview の一部抜粋になり ますが、元になるクロックソース(下図の青枠)が、セレクタや分周回路(下図の緑枠)を経て、ペリフェラ ル・クロック(下図の赤枠)になります。



シンプルな図にすると下図になります。色づけした箇所が、周波数に影響するブロックです。



クロックソースとして、HFRCO(内蔵の高速クロック)を選択した場合を例に説明します。

◆ HFRCO の 周波数 設定 (クロックソース)

HFRCOの周波数は、CMU\_HFRCOCTRLのBANDビットで設定されています。

下図は Leopard Gecko のリファレンスマニュアルからの抜粋ですが、1 (実際には 1.2),7 (実際には 6.6)、11、14、21、28MHz から選択ができます。デフォルトでは 14MHz に設定されています。

11.5.4 CMU\_HFRCOCTRL - HFRCO Control Register

	_																															
Offset															в	it Po	ositi															
0x00C	31	30	29	28	27	26	25	24	23	3	21	20	19	18	17	16	15	14	13	12	÷	10	თ	ω	7	9	5	4		N	-	0
Reset																		0×00					0X3	)					0×80			
Access																		RW					RW						RW			
Name																		SUDELAY					BAND						TUNING			
															ess Description																	
вп	Na	me						Re	set			,	100	ess	s Description																	
10:8	BA	ND						0x3	5			F	w		HFRCO Band Select																	
	Write this field to set the frequency band in which the HFRCO is to operate. When changing this setting there will be n the HFRCO output, hence it is safe to change this setting even while the system is running on the HFRCO. To ensure frequency, the HFTUNING value should also be written when changing the frequency band. The calibrated tuning different bands can be read from the Device Information page.							no gl e an valu	litch acc ie fo	es on curate or the																						
	Va	lue			M	lode								(	Desci	riptior	n															
	0				11	MHZ	2							1	1 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.																	
	1				7	MHZ	2							7	7 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.																	
	2	2 11MH7										1	11 M	Hz ba	and M		· AI	50.5	et the		JING	valu	e (hi	ts 7	0) w	hen	char	naina	hand	4		



14MHZ

21MHZ

28MHZ

14 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.

21 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band

28 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.

# TecStar -

◆ クロックソースの選択 (セレクタ)



クロックソースは、CMU\_CMD の HFCLKSEL で設定します。

11.5.10 CMU\_CMD - Command Register

Offset															в	it Po	siti	on														
0x024	3	30	29	28	27	26	25	24	53	52	5	20	19	9	1	16	15	4	9	13	÷	9	6	80	2	ø	2	4	e	2	-	0
Reset																											0X0	•	0		0×0	
Access																											۲1	W1	W1		W1	
Name																											USBCCLKSEL	CALSTOP	CALSTART		HFCLKSEL	
2:0	HF	CLK	SEL					0x0	)			۷	V1			HF	CLK	Sel	ect													
	Selects the clock source for HFCLK. Note that selecting an oscillator that is disabled will cause the system clock to stop. Check the status register and confirm that oscillator is ready before switching.						the																									
	Value Mode									Description																						
	1	1 HFRCO									Select HFRCO as HFCLK.																					
	2 HFXO Select HFXO as HFCLK.																															
	3 LFRCO Sele									Selec	t LFF	CO a	is HF	FCLI	К.																	

Select LFXO as HFCLK

◆ HFCLK の決定 (分周 1)

LFXO



クロックソース(HFRCO)を分周して HFCLK にします。何分の1にするかは、CMU\_CTRL の HFCLKDIV で決定します。

HFCLK は、クロックソース(HFRCO)の 1/(HFCLKDIV+1) なります。1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 から選択できます。HFCLKDIV = 0であればHFCLK = HFRCO、HFCLKDIV = 7であればHFCLK = HFRCO \* (1/8) になります。

この HFCLK は、CPU コアの動作周波数にも利用します。

Offset															Bi	t Po	ositi	on													
0x000	3	8	29	28	27	26	25	24	23	3	2	20	9	9	17	16	15	4	13	12	÷	9	ი	 2	9	2	4	e	~	-	0
Reset		0		0				0X0			0×0		2	ŝ	0		0×0		-	2	2	3	ŝ	0		X1		2	200	2	2
Access		RV		RV				RW			RV			۶ ۲	RV		RW		ΝŇ			ł	A Y	RW	i	N N			A L	Md	~
Name		HFLE		DBGCLK				CLKOUTSEL1			CLKOUTSEL0		I POTIMEOUT	LFAUIIMEOUI	LFXOBUFCUR		HFCLKDIV		LFXOBOOST			TICLINECOLI	HFAUIIMEOUI	HFXOGLITCHDETEN		HFXOBUFCUR		TOCOCOL	HFAUBOUSI	HEXOMODE	
16:14	HF	CLK	DIV					0x	0			F	w			HF	CLK	Div	/isi	on											
	Use to divide HECLK frequency by (HECLKDIV + 1).																														

#### 11.5.1 CMU\_CTRL - CMU Control Register

◆ HFPERCLK の決定 (分周 2)



HFCLK を分周して、HFPERCLK にします。何分の1にするかは、CMU\_HFPERCLKDIVの HFPERCLVDIV ビットで決定します。

分周は、1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 から選択できます。1/512 を選択した 場合には、HFPERCLK = HFCLK \* (1/512) になります。

# 11.5.3 CMU\_HFPERCLKDIV - High Frequency Peripheral Clock Division Register

						_	_																									
Offset															Bi	it Po	ositi															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	8	17	16	15	4	13	5	÷	9	6	8	7	9	5	4	e	8	-	0
Reset																								-							, X	
Access																								ΝŇ							۸ ۲	
Name																								HFPERCLKEN							HFFERCLAUIV	
3:0	HFF	PER	CLK	DIV		0x0 RW										HF	PER	RCLI	K D	Divid	er											
	Spe	cifie	s the	e clo	ock d	livid	er fo	or the	e HF	PEF	CLF	ς.																				
	Val	ue			M	lode								Description																		
	0				н	FCL	ĸ							-	HFPERCLK = HFCLK.																	
	1				н	FCL	.K2							-	HFPERCLK = HFCLK/2.																	
	2				н	FCL	.K4							-	HFPERCLK = HFCLK/4.																	
	3				н	HFCLK8								ŀ	HFPERCLK = HFCLK/8.																	
	4				н	HFCLK16								ŀ	HFPERCLK = HFCLK/16.																	
	5				н	HFCLK32								ŀ	HFPERCLK = HFCLK/32.																	
	6				н	FCL	.K64							ŀ	HFPERCLK = HFCLK/64.																	
	7				н	FCL	K12	в						ŀ	HFPERCLK = HFCLK/128.																	
	8				н	FCL	K25	6						ŀ	HFPERCLK = HFCLK/256.																	
	9 HFCLK512										ŀ	IFPE	RCL	K = H	IFCL	.K/5	512.															

◆ まとめ

**TecStar** 

以上をまとめますと、ペリフェラルに供給されるクロックの周波数は、下記の組み合わせになります。



HFRCO を分周し、通常ペリフェラルで使用した場合の事例を紹介しましたが、HFRCO ではなく HFXO(外付け高速クロック)/LFRCO(内蔵の低速クロック)/LFXO(外付け低速クロック)を使用した場 合や、通常ペリフェラルではなく Low Energy ペリフェラルを使用した場合にも、CMU Overview のクロッ クツリーから読み解いていくという考え方は共通です。

# TecStar =

### 改版履歴

Version	改定日	改定内容
1.0	2014年07月	·新規作成
1.13	2015年06月	・マクニカオンラインで公開
1.14	2015年08月	・誤植訂正(6-2章 M0+の breakpoint数)、4-1, 6-3, 6-6-4,
		6-7-3 章追記
1.15	2016年02月	・Pearl Gecko/Jade Gecko 追加。最新の Simplicity Studio
		に合わせて説明を一部変更。
2.0	2017年1月	・Simplicity Studio ver.4 に対応
2.1	2017年3月	・Pearl Gecko PG12/Jade Gecko JG12 追加。最新の
		Simplicity Studio に合わせて説明を一部変更。
2.2	2017年8月	・Giant Gecko GG11 追加。オフライン・インストーラの入手
		方法について追加。
2.3	2018年3月	・Tiny Gecko S1(TG11)追加。最新の Simplicity Studio に
		合わせて説明を一部変更。

### 参考文献

- Silicon Labs 社 各種ドキュメント
- Silicon Labs 社 ナレッジベース、コミュニティフォーラム

# 免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を 一読いただいた上でご使用ください。

- 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
- 2. 本資料は予告なく変更することがあります。
- 3. 本資料の作成には万全を期していますが、万一ご不審な点や誤り、記載漏れなどお 気づきの点がありましたら、弊社までご一報いただければ幸いです。
- 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的なものとしてかかれたものです。製品をご使用 になる場合は、メーカーリリースの資料もあわせてご利用ください。

〒222-8561 横浜市港北区新横浜 1-6-3 TEL 045-470-9841 FAX 045-470-9844

本社