Technical Note



Silicon Labs 社 EFM8/C8051 クイックスタートガイド

2018年3月





TecStar -

Silicon Labs 社 EFM8/C8051

クイックスタートガイド

目次	
1 はじめに	
2 開発環境のご紹介	
2-1 ハードウェア	5
2-1-1 EFM8 Starter Kit	
2-1-2 C8051 Development Kit	
2-1-3 C8051 ToolStick	
2-2 ソフトウェア	8
2-2-1 Simplicity Studio	
3 各種ドキュメントの入手方法	
3-1 ドキュメントの入手方法(Simplicity Studio から)	
3-1-1 情報が表示されない場合には?	
3-1-2 欲しい情報が見つからない場合には?	
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)	14
3-3 EFM8のAPI 情報	15
4 ソフトウェア・インストール	
4-1 Simplicity Studio のインストール	16
4-2 インストールがうまくいかない場合	17
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	
4-2-3 オフライン・インストーラ	
4-3 KEIL コンパイラのライセンス設定	21
5 ハードウェア・セットアップ	
5-1 EFM8 Starter Kit のセットアップ	22
5-2 C8051 Development Kit のセットアップ	22
5-3 C8051 ToolStick のセットアップ	23
6 使用方法	
6-1 サンプルコードを動かしてみる	24
6-2 デバッグ機能を使ってみる(Debug)	29
6-3 消費電流を測定してみる(Energy Profiler)	31
6-4 ピン設定やペリフェラル設定をしてみる (Hardware Configurator)	32
6-5 ピン設定やペリフェラル設定をしてみる(Configuration Wizard 2)	35

6-6 Simplicity Studio ver.3 から ver.4 への移行	37
7 ソフトウェア設計	38
7-1 ソースコードの追い方	38
7-2 サンプルコードにペリフェラルを実装してみる (外部割込み)	39
7-2-1 サンプルコードを理解する (EFM8BB3_Blinky)	40
7-2-2 サンプルコードを理解する (EFM8BB3_ExternalInterrupts)	42
7-2-3 ペリフェラル設定を移植する (EFM8BB3_ExternalInterrupts の設定を読み取る)	44
7-2-4 ペリフェラル設定を移植する (EFM8BB3_Blinky に設定を移植する)	47
7-2-5 アプリを実装する	50
改版履歴	52
参考文献	52

1 はじめに

この資料は、Silicon Laboratories(以下、Silicon Labs)社製 MCU EFM8/C8051 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	e - from Silicon Labs	*
Presentations and Brochures		 Complicit 	v Ctudio /	
Silicon Labs Community			y 310010 4	

Web Site から

https://www.silabs.com/community	(Silicon Labs	社製品全般)
https://www.silabs.com/community/mc	u/8-bit/forum	(8-bit MCU に特化)

◆ 使用方法



2 開発環境のご紹介

EFM8/C8051 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

開発環境としては、EFM8 用に Starter Kit を、C8051 用に Development Kit と ToolStick を用意しています。Starter Kit と ToolStick は小型サイズで、お手軽にご評価頂けます。

2-1-1 EFM8 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、ROM/RAM サイズや ペリフェラルの数に差異がありますが、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。

ファミリ名	形名	Starter Kit	実装されている型番
Busy Bee	EFM8BB1	SLSTK2020A	EFM8BB10F8G
Busy Bee	EFM8BB2	SLSTK2021A	EFM8BB22F16G
Busy Bee	EFM8BB3	SLSTK2022A	EFM8BB31F64G
Sleepy Bee	EFM8SB1	SLSTK2010A	EFM8SB10F8G
Sleepy Bee	EFM8SB2	SLSTK2011A	EFM8SB20F64G
Universal Bee	EFM8UB1	SLSTK2000A	EFM8UB10F16G
Universal Bee	EFM8UB2	SLSTK2001A	EFM8UB20F64G
Universal Bee	EFM8UB3	SLTB005A	EFM8UB31F40G
Laser Bee	EFM8LB1	SLSTK2030A	EFM8LB12F64E

• EFM8BB1 : SLSTK2020A



♦ EFM8UB3 : SLTB005A



TecStar

2-1-2 C8051 Development Kit

ベーシックな評価基板が入った開発キットです。ターゲットボード、USB Debug Adaptor (PC とターゲッ トボードとをつなぐ機材)、AC/DC アダプタが同梱されています。MCU ファミリごとに Development Kit が 用意されており、ターゲットボードに実装されている MCU が異なっています。

ターゲットボードには LED やスイッチが実装されている他、アナログ入力やタッチボタンなど、各 MCU ファミリが持つ特徴的な機能を評価できる作りになっています。また、全ての I/O ピンが引き出されてい ますので、拡張性にも富んでいます。



Development Kit

-ゲットボード

2-1-3 C8051 ToolStick

USB 給電で動作する、非常にコンパクトな評価基板です。PCに繋がるベースアダプタ、評価対象を搭 載したドーターカード、の2つで構成されています。

ドーターカードは、MCU ファミリごと、パッケージごとに多種用意しており、評価したい MCU に最適な ドーターカードをお選び頂くことができます。またソケットが載ったプログラミングアダプタも用意しており、 プログラミングにご使用頂けます。



2-2 ソフトウェア

EFM8/C8051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。C・アセンブラのコンパイラについては、KEIL 社のコンパイラ(フルライセンス)を無償提供しています。

2-2-1 Simplicity Studio

Simplicity Studio は、EFM8/C8051 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32bit 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 KEIL 社コンパイラを搭載
Energy Profiler	実機の消費電流値を測定することが可能。EFM8 に対応
Handman Confirmator	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。新しい
Hardware Configurator	製品(EFM8、C8051 の一部)に対応
Confirmation Wireard 2	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。レガシ
Configuration wizard 2	一製品(C8051)に対応
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)

Silicon Labs energyAware Profiler		
File Options Help		
J-Link Device S/N 440013344 C Giant Gecko V Grant Gecko	garithmic plot 🔽 Annotate IRQ 🔲 Plot voltage 🛛 🛨	
Code View		
To enable code view you must do two things:	Selected point	
1 Your gode must enable SWO sutput from the FEWS2 MCH. To	BTC IBOHandler Current	
enable this output, simply pasts the code below into your	Voltage	
EFM32 application as you would a function, and call the	100 mA	
function setupSWO() early in your main() program.	PC	
This code is also available in the bsp directory of your	10 mA IRQ	
kit.		
2. Within the energyAware Profiler, select the object code	1 mASelected range	
you want to debug. Click on "File->Load Object File".	Ave. current	
For more information, see the online belo	100 um_111 Time	
for more information, see the online help.	10 JA LANA NILLIIII A Energy	
	Burning average	
	1 uA	
void setupSWO(void)	Current 50.46 µA	
(100 nAPower 167.18 µW	
/* Enable GPIO Clock. */	- Energy counter	
CMU->HFPERCLKENO = CMU_HFPERCLKENO_GPIO;	10 nA	
/* Enable Serial wire output pin */	Energy 4.78 J	
#if defined(EFM32 GTANT FAMILY)	Reset counter	
defined(EFM32 WONDER FAMILY)		
defined(EFM32 LEOPARD FAMILY)	Epergy Profile	Ints
/* Set location 0 */	Exercise Exercise (a)	
GPIO->ROUTE = (GPIO->ROUTE &	Function Energy (UJ) Contribution (%)	
~(_GPIO_ROUTE_SWLOCATION_MASK))	RTCDRV_Delay 000000 7	8%
GPIO_ROUTE_SWLOCATION_LOCO;	LCD_SyncBusy*** 800881	8%



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

◆ ピン設定やペリフェラル設定を簡単に行えます(Configuration Wizard 2)



3 各種ドキュメントの入手方法

EFM8/C8051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)

EFM8/C8051 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。

Solutions		Ŧ
New Solution		
EFM8BB3		
🔺 Kits 🎽		*
EFM8BB3 Busy Bee Starter Kit (SLSTK2022A)		
▲ Boards		
EFM8BB3 Busy Bee Starter Kit Board (BRD5202A Rev A00))	Ξ
EFM8BB3 Busy Bee Starter Kit Board (BRD5202A Rev A02))	
EFM8BB3 USB Type-C 60 W Charger (BRD5204A)		
EFM8BB3 USB Type-C Rechargeable Battery Pack (BRD520)3A)	

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	Docume	entation	Compatible Tools	Resources	
Demos	-+⊠≡	Software Examples	-+⊠≡	Documentation	-+⊻≡
 ✓ 8051 SDK v4.1.1 → ADC 		 8051 SDK v4.1.1 ADC 		The preferred SDK does not contain of the selected device. Showing application	documents for Automatic Au
Blinky		Blinky		Application Notes	
CPT_Demo		Bootloaders			
Configurable Logic		CPT_Demo			
DAC		Comparators			
General		Configurable Logic			
		► DAC			

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。

Documentation:

Application Notes 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。



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。

- 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 ... EFM8/C8051 のデータシート。
 - Errata ... EFM8/C8051 のバグ情報。
 - Reference Manual ... EFM8 の動作仕様書。C8051 ではデータシート内に記載されています。
 - Schematic and Layout Files ... ラジオボードの回路図・部品表・レイアウト情報。
 - Uncategorized Documents ... EFM8 など特定型番向けのライブラリ情報。
 - 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、Bluetooth 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。



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		
• New Solution		
Enter product name		
🔺 🧰 temp		
4 📼 EFM8BB3 Busy Bee	Starter Kit (SLSTK2022A)	
EFM8BB3 Busy E EFM8BB31F6	Bee Starter Kit Board (BRD5202A Rev A02) 4G-B-QFN32	III

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。

Getting Starte	ed	Documentation		Compatible Tools	Reso	ources	
My Favorite Doc	uments		-+ 🗹	All Documents			-+
No documents have b	een favorited.	Click the 'Favorite' icon to add a document here	2.	8051 SDK v4.1.1			
(🗠 Documen	t Categories Dialog				EX	
	All Docum Use this d	ents Preferences ialog to customize your documentation preference	ces			3	≛ ☆
	Check the c	ategories you would like to display. Use 'Move U referred SDK - 8051 SDK v4.1.1	ip' and 'Move	e Down' to reorder the categories.			
	Show	Category			Documents	Move Up	L 🖒
		Application Notes			8	Maua Daum	
		Data Sheets			1	Move Down	
		Uncategorized Documents			1		
		User's Guides			1	Select All	
						Jelect All	
						Unselect All	
	1					x	
				ОК	Ca	ancel	

3-1-3 表示される情報を制限したい場合には?

Demos, Software Examples, SDK 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。

🧮 リストを折りたたむ

TecStar —

📍 リストを展開する

✓ 表示する大項目を選択する

📒 別の一覧表を表示する(Demos, Software Examples のみ)

PK v4.1.1 C DK v4.1.1 C The preferred SDK does not contain documents for the selected device. Showing application notes. Manufication Notes
DK v4.1.1 The preferred SDK does not contain documents for the selected device. Showing application notes. Annlication Notes ences
nky Anniication Notes
ences
ences
ences
Examples Select All
9 Unselect All
1
2
2

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 Sta	rted	Documentation		Compatible Tools	Resources
My Favorite I	Documents		-+1	All Documents	- + 🗹
✓ Data Sheets EFM8BB3 Data Description is	ta Sheet unavailable		b *	8051 SDK v4.1.1 2M8BB3 Busy Bee Starter Kit (SLSTK2022A Application Notes Data Sheets)
				EFM8BB3 Data Sheet Description is unavailable	* ۵

3-2 ドキュメントの入手方法 (Web から)

EFM8/C8051 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。 EFM8/C8051 は、Products -> Microcontrollers -> 8-bit MCUs の下に分類されています。

Silicon Labs » Support » Technical Resource Search

Technical Resource Search

Narrow by:	Apply text filter	Apply text filter Q				
* Products: 8-bit MCUs	Title 🔻		Version	Resource Type		
Clear All				Cabanatia and		
Products -	1002-TCB1 D 434 Sche	ematics and Layout		Layout Files		
Analog	1106221 E52x-E53x E0	DL and LTB std		Product Change		
Audio and Radio	1100221 F32X-F33X EC			Notifications (PCN)		
Interface	Resource Type	sheet Revision from 1.0 to 1.1		Product Change		
Isolation	Application Notes			Notifications (PCN)		
Microcontrollers	Data Sheet Addendums	b(TSMC) Site Expansion Std		Product Change		
32-bit MCUs	Data Sheets	-		Notifications (PCN)		
8-bit MCUs	🔲 Errata	and Reel Supplier std		Product Change		
C2051500y 01y	Example Code			NOUNCATIONS (PCN)		
	Getting Started	x Datasheet rev 1 4 update std		Product Change		
C8051F02x	Manuals	•		Notifications (PCN)		
C8051F04x	 Miscellaneous 			Product Change		
	Product Change Notifications (PCN)					
	Reference Designs					
	Release Notes					
	Schematic and Layout Files					
	Software					

Expand All / Collapse All Showing 50 of 956 Results

3-3 EFM8のAPI 情報

EFM8 には、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。一部 C8051 についても用意されています。

「3-1 EFM8/C8051のドキュメント」の手順に従って、使用する型番を選択してください。

Documentation タブ ⇒ 8051 SDK documentation を選択します。

Getting Started	Documentation	Compatible Tools	Resources	
My Favorite Documents	Customize	All Documents		Customize
Data Sheets		8051 SDK v4.0.3	204)	
EFM8BB1 Data Sheet	🖬 🔶	Uncategorized Documents	204)	
Description is unavailable		8051 SDK Documentation		
		8051 SDK Documentation		

ブラウザが起動し、API 情報が表示されます。

SILICON LABS EFM8BB1 Peripher	B1 Peripheral Driver Library Product logo al Driver Library Reference
Main Page Modules Files	Search
EFM88B1 Peripheral Driver Library EFM88B1 Firmware Documentation Modules I2C0 Driver	Enumerations Functions SPI0 Initialization API Sour Enumerations Functions
PCA0 Driver	Enumerations
SPI0 Driver SPI0 Runtime API SPI0 Initialization API SPI0 Butflered API UART0 Driver WDT0 Driver	enum SPI0_ClockMode_t { SPI0_CLKMODE_0 = SPI0CFG_CKPOL_IDLE_LOW SPI0CFG_CKPHA_DATA_CENTERED_FIRST, SPI0_CLKMODE_1 = SPI0CFG_CKPOL_IDLE_LOW SPI0CFG_CKPHA_DATA_CENTERED_SECOND, SPI0_CLKMODE_2 = SPI0CFG_CKPOL_IDLE_HIGH SPI0CFG_CKPHA_DATA_CENTERED_FIRST, SPI0_CLKMODE_3 = SPI0CFG_CKPOL_IDLE_HIGH SPI0CFG_CKPHA_DATA_CENTERED_SECOND }
► Files	Functions
	void SPI0_init (SPI0_ClockMode_t clockMode, bool isMasterMode, bool is4wire)
	void SPI0_reset(void) void SPI0_setClockDivider (uint16_t divider)
	Detailed Description

The following functions are part of the SPI initialization API. These functions are used to configure the SPI peripheral at run-time. It is possible to also configure the SPI peripheral as part of system initialization in which case it is not necessary to use these functions.

4 ソフトウェア・インストール

EFM8/C8051 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、EFM8 / C8051 8-bit Products にチェックして、 Next をクリックします。

🛃 Install Wizard		
Install Device Support Select the devices you would like to use within !	Simplicity Studio.	
Select by Device Select by Product	Group	
111 111 111 Xpress Products	EFM8 / C8051 8-bit Products	Image: Constraint of the second se
Wireless Products Selected Products: 15	Download Size: 533 MB	Required Disk Space: 820 MB
	< Back Next >	Finish Cancel

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。



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

EFM8/C8051 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/

5	Y	简体中文 繁體中文 日本語	Log In Register		
SILICON LABS				Parametric Search Cross	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。

Create An Ac	count	
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 emai from Silicon Labs	communications	
	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)		f	简体中文 繁體中文 日本語	Log In Register
SILICON	LABS			Parametric Search Cro	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。

S		简体中文	繁體中文 日本語	Welcome, 🗾 🔻
SILICON LABS			Parametric Search	Cross-Reference Search
About 👻 Prod	ucts Solutions	Community & Support 🔻	Search silabs.com	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			F Tools
Devices Solutions New Solution Add Dev	ices	Welcon	ne to S	implic	ity St	udic
· 🚁 Preferences						
type filte text	Network Conne	ctions			← ▼ □	⇒ ▼
Capalilities Help Instal/Undate	Active Provider	Manual 🗸				
Network Connections	Proxy entries					
Security	Sch H	lost Port	Prov Auth	User	Passwon	Edit
Simplicity Studio	📝 НТТР		Man No			Clear
	✓ HT		Man No			
	V SO		Man No			
	設定項	頁目·設定内容	は、各企業材	羨によって 身	異なります	-
	Proxy bypass					
	Host		Provider		Add	Host
	localhost		Manual		E	dit
	127.0.0.1		Manual		Re	move
	•		m		•	
			Restore Defau	lts	Apply	
			OK		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 👻 🔅 🛃 😽 🗕	→ =	: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 👻									
Proxy entries									
	Schema	Host	Port	Provider	Auth	User	Password		Edit
	HTTP			Manual	No				Clear
	HTTPS			Manual	No				Cicui
	SOCKS			Manual	No				
	HTTP	Dynamic	Dynamic	Native	No				
Pro	xy bypass								
	Host		Provider					Ade	d Host
	✓ localhost Manual							Edit	
	V 127.0.0.1 Manual							Eult	
								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

4-3 KEIL コンパイラのライセンス設定

KEIL コンパイラを利用するために、ライセンス登録を行います。

Simplicity Studio を起動し、Tools ⇒Simplicity IDE ⇒ OK を選択します。



Help メニュー \Rightarrow Licensing \Rightarrow Keil を選択します。

🛩 Simplicity	IDE	- Simplicity St	udio ™					
File Edit I	Navig	jate Search	Project	Run	Window	Help		
☆ - 0	0	Help Content	s					
		Licensing	_			\rightarrow	Keil 8051 v9.53	1
Projec	89	Search						
		Dynamic Help						
		Key Assist			Ctrl+Shif	t+L		
		Tips and Tric	ks					

Licensing Helper ウィンドウの中央にある「this form」をクリックし、必要事項を入力します。メールで ライセンスキーが送られてきますので、それを Licensing Helper ウィンドウに入力してください。



5 ハードウェア・セットアップ

EFM8/C8051の評価に必要なハードウェアの設定を行います。

5-1 EFM8 Starter Kit のセットアップ

以下の手順で設定していきます。

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

DBG





BAT / USB / AEM

5-2 C8051 Development Kit のセットアップ

以下の手順で設定していきます。

- 1. USB デバッグアダプタを、ターゲットボードの DEBUG ポートに接続します。
- 2. USB ケーブルを、USB デバッグアダプタに接続します。
- 3. ターゲットボードにショートブロックがある場合には、ユーザガイドの指示に従って、正しく結線さ れているか確認してください。

例) F912DK, F930DK, F996DK の場合: J17をショート、SW5をONに設定。

- 4. USB ケーブルのもう一方を PC に接続します。
- 5. AD/DC アダプタをターゲットボード、コンセントに接続します。



5-3 C8051 ToolStick のセットアップ

以下の手順で設定していきます。

1. ベースアダプタとドーターカードとを、下の写真のように接続します。



 F912DC や F990DC など、いくつかのドーターカードでは電池でも動作するようになっています。
 USB 給電で動作するか、電池で動作するかを切り変えるスイッチがありますので、USB 給電(TS PWR)を選択して下さい。





3. USB コネクタを PC に接続します



6 使用方法

評価キットとSimplicity Studioを使用した評価手順をご紹介します。ここではSLSTK2010A(Sleepy Bee) を使用しておりますが、他の評価キットでも手順は同じです。なお、各ツールからSimplicity Studioのトップ 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	Documentation		Compatible Tools	Resources	
Demos	■+ 🗹 ≡	Software Examples	→ 🗹 🗏	Documentation	-+⊻≡
 ▼ 8051 SDK v4.1.1 ▶ ADC 		 8051 SDK v4.1.1 ADC 		The preferred SDK does not contain of selected device. Showing application	documents for the 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 Project		Child Same To Manager Die problems	1.000	
Project setup Select the board, part, and SDK for t	he project.			
Boards:				
Search				-
EFM8SB1 Sleepy Bee Starter Kit Bo	ard (BRD5101A Rev A01)	×		
Part:				
SDK:				
8051 SDK (v4.0.2) (C:¥SiliconLabs¥	SimplicityStudio¥v4¥devel	loper¥sdks¥8051¥v4.0.2¥)		• ()
				Manage SDKs
0	< Back	Next >	Finish	Cancel



Example Project で EFM8SB1 Blinky 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.	
type filter text	S
	•
GR1 Canacitive Genes for overlaw eval on overlaw This program flashes the LEDO on the EFM8581 brget board about five times a second using the interrupt handler for Timer2.	
< Back	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
Link to courses	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 liberains 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 noraries 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 contents	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」で紹介します。

Simplicity IDE - EFM6561_8knky/src/EFM6561_8k	nity c - Simplicity Studio **				(c)B=
File Edit Source Refactor Navigate Search P	reject Run Window Help				
\$ • 9. • [2] • [2] [0] [0 • % • [#	📸 🏫 Launchar 👔) Simplicity IDE			
🕒 Project Explorer 🕮 🖻 🕸 🦈 🖱 🗖	PROFILER_README.txt	EFM8SB1_Blinky.c			
Contrast, Joney (contrast, 4-53) - Goog (P) Contrast, Joney (contrast, 4-53) - Goog (P) Contrast, Joney (contrast, 4-54) Contrast, Joney (contrast, 4-54)	 Marging and Statistical Control (1998) Marging Barging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) Marging Galaxies, and an analysis of the statistical control (1998) 	An and a second	er,	,	
	Problems 12 Search	Call Hierarchy			
	0 items	a contract of the contract			
	Description		Resource	Path	
	×				

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。



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。

Debug II	Q • N · · · · ·	∞ Variables ≋	Breakpoints	III Regis	ters	& Expressio	ins	
 Silicon Labs 8051 MCU: EFM85810F8G-A-QFN24 EFM85811 Bislau conf 						80 e		📫 टो
main() at EFMESB1_Binky.c:62 0x4d		Name	туре	VB	U8	LO	Jostion	
		<						,
PROFILER_README.txt README.txt README.txt					Disass	embly a		-
#define DISP_BC_DRIVEN 0 // 0 = Board Contr	oller drives display				Enter loca	tion here	• 8 b	20
#define DISP_EPH8_DWIVEN 1 // 1 = EPH8 drives	display			-				
// mein() Routine					610000	LCALL ent	ter Default	thole
<pre>// @ int main (void)</pre>					4	DISP_EN CLR P2	<pre>= DISP_BC_B .7</pre>	DRIVEN
{ //Enter default mode					6	shile (1)	2.0	
enter Defaulthode from RESET();					0	1992CN0_1	TF2H = 0;	
DISP EN = DISP BC DEIVEN					0000054:	CLR THE	RxCN.7	
while (1) ()					0000056:	CPL P1	1	
)				1 - C	0000058:	RETI		
					.02	PD/DOUT + PORTS 1 enter	 PIMDOUT_8 Default%r 	ode fr
				•	_	<		,
Console II Memory O Executables				÷ 1) 🖾 🗳	🔉 = Uk t	3 🛛 - 🗅	• •
OT Build Console [EFM8581_Binky]	72 - Orber for and an 7000							
ike all	in project com							
ke: Nothing to be done for 'all'.								

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。

😰 🗎 🏫 Launcher	{} Simplicity IDE	bug
implicity IDE (ビルド用)		へ Debug(デバッグ用)

S

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	
6		N 🕹	• • •	.¢ i≯ @		- 🙋 🎿		- 8 0	÷ 🔶	• <> • 📖

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。SLSTK2010A の場合には、LCD 下の LED0 が点滅します。



◆ LED の点滅スピードを変更してみましょう。

TecStar =

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。

😰 🛛 🏫 Launcher 🚺 Simplicity IDE 🕇 🌾 Debug

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。

ြာ Project Explorer 🛛 🕞 🕏 🍸 🖓 🗖	
🔺 😂 EFM8SB1_Blinky [Keil 8051 v9.53 - Debugا الته	
> 🎎 Binaries 🔶	EFM8SB1 Blinky.c: メインルーチン
Includes	
⊳ 😅 bin 🔶	InitDevice.c: 主にペリフェラルの初期化
⊳ 😅 inc	
⊳ 😅 Keil 8051 v9.53 - Debug	Interrupts.c: 割り込み処理
🔺 😅 src 🔶	EFM8SB1 Blinky hwconf
▷ 📓 EFM8SB1_Blinky.c	
▷ 🛃 InitDevice.c	Hardware Configurator のプロジェクトファイル
Interrupts.c	
EFM8SB1_Blinky.hwconf	
main()を見ると LED 点滅に関する処理	は行われておらず、Interrupts.c を見ると Timer2 割り込みで

LED への制御ピンを反転させています。Timer2 のオーバフロー周期を変更すれば点滅スピードを変え

ることができそうです。	
//// main() Routine	// TIMER2_ISR // TIMER2_ISR
// int main (void) { //Enter default mode	// // TIMER2 ISR Content goes here. Remember to clear flag bits: // TMR2CN0::TF2H (Timer # High Byte Overflow Flag) // TMR2CN0::TF2L (Timer # Low Byte Overflow Flag)
<pre>enter_DefaultMode_from_RESET();</pre>	// //
DISP_EN = DISP_BC_DRIVEN;	SI_INTERRUPT (TIMER2_ISR, TIMER2_IRQn) {
<pre>while (1) {} // Spin forever }</pre>	LED0 = ILED0; // change state of LEDs }

もちろんソースコードを追って、設定変更する方法もありますが、ここでは Hardware Configurator を使ってみます。FM8SB1_Blinky.hwconf をダブルクリックすると Hardware Configurator が起動します。

🍐 Project Explo 🛛 🗖 🗖	PROFILER_README.txt	FM8SB1_Blinky.c	*EFM8SB1_Blinky.hw	×	Be Outline ⊠		- 0
🖻 🔹 🗢	DefaultMode Peripherals				Mode Transitions		
EFM8SB1_Blinky [Keil 8051 v9.5]					RESET → DefaultMode		
▷ ﷺ Binaries	Core			*	A Build DefaultMode		
▷ M Includes					Peripherals		
Din 🖉 Din	External Interrupts Flash C	ontrol I Interrupts	Reset Sources				
inc a discillation of the Deliver					CRUSSBARU		
Kell 8051 v9.53 - Debug							
EFM85B1_Blinky.c	Other			*			
Intervente e							
: EEM8SB1 Blinky bwconf					*Properties X # Periph	eral Map	
Er Mobbi_blinky.hwcoli							7 7
					Properties of Timers		
	(+)				TIMER Setup TIMER 0/1 TIMER	2 TIMER 3	
	Power			*	Property	Value	^
					Control		
	PMU Supply Monitor	Voltage Regulators			Clock Source	SYSCLK / 12	
					Clock Source Frequency	208.333 kHz	
					Clock Source Period	4.800 us	
					Enable Low Byte Interrupt	Disabled	
	(V)				Mode	One timer, 16-bit	
	Timers .			*	Run Control	Start	
					Timer Running State	Timer is Running	
	PCA RTC Timers				Init and Reload Value		Ξ
					Target Overflow Frequency	10	
				-	Timer Init Overflow After	4.800 us	
	A Mode Transitions 😫 Defaurement	te Peri 🖉 🖉 🖬 🖉 🛛	efaultMode Peripherals 🛛		Timer Init Value	65535 (0xFFFF)	
					Timer Reload Overflow Frequ	i€ 10.000 Hz	
Problems 🛛 📮 Console				~	Timer Reload Overflow Frequ	i€ 2.148 kHz	
tems					Timer Reload Overflow Perio	d 99.999 ms	
escription	*	Resource	Path	Locat	Timer Reload Overflow Perio	d 465.601 us	
Infoc (1 item)					Timer Reload Value	44703 (0xAE9F)	

DefaultMode Peripherals タブを選択し、Timers を選択、画面右に Timer 設定が表示されます。

Timer2の初期値設定を見てみると、Timer2のオーバフロー周期が10Hzに設定されていました。これを50Hzに変更してみます。

Init and Reload Value		□ Init and Reload Value	
Target Overflow Frequency	10 (0xA)	Target Overflow Frequency	50 (0×32)
Timer Init Overflow After	4.800 US	Timer Init Overflow After	4.000 us
Timer Init Value	65535 (0×FFFF)	Timer Init Value	65535 (0xFFFF)
Timer Reload Overflow Frequency	10.000 Hz	Timer Reload Overflow Frequency	49.996 Hz
Timer Reload Overflow Frequency(Low Byte)	2.148 kHz	Timer Reload Overflow Frequency(Low Byte)	2.934 kHz
Timer Reload Overflow Period	99,999 ms	Timer Reload Overflow Period	20.002 ms
Timer Reload Overflow Period(Low Byte)	465.601 us	Timer Reload Overflow Period(Low Byte)	340.801 us
Timer Reload Value	44703 (0×AE9F)	Timer Reload Value	61369 (0×EFB9)

設定変更をソースコードに反映させます。DefaultMode Peripherals タブに戻り、画面上で右クリックし、Generate Source を選択します。

Timers	2 🎸 🔊	Reset TIMER_SETUP_0 Undo Redo	*	-
	Ł	Generate Source		
PCA </th <th>□ ₽</th> <th>Show Properties Show Outline View</th> <th></th> <th></th>	□ ₽	Show Properties Show Outline View		
➢ Mode Transitions ☎ DefaultMode Port	I/O	📲 DefaultMode Pe	ripherals 🛛	

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) ⇒虫のアイコン(Debug)⇒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。

使用するMCUファミリによってサンプルコードの内容は異なりますが、Blinky.cのようにシンプルなサンプルコードは、制御方法を理解するのに最適です。

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。





TecStar -

◆ 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	Breakpoint	ts and Reg	isters 🛛 🕸 Expressions	-
			<u>د</u> من ا	📫 Ľí
Name		Value	Description	
▷ ## SFR Pagin	9		SFR Paging Registers	
MBus 0			SMBus 0 Registers	
M SPI 0			SPI 0 Registers	
Temperation	ure Sensor		Temperature Sensor Registers	
▷ ## TIMER 0/1	L		TIMER 0/1 Registers	
4 👬 TIMER 2			TIMER 2 Registers	
▷ 1818 TMR2		0x0	Timer 2 Word	
▷ 1810 TMR2C	N0	0x0	Timer 2 Control 0	
▷ 1810 TMR2R	iL	0x0	Timer 2 Reload Word	
TIMER 3			TIMER 3 Registers	
▷ ## TIMER Set	tup		TIMER Setup Registers	
MUART 0			UART 0 Registers	
Supply Mo	nitor		Supply Monitor Registers	
Voltage Re	eference		Voltage Reference Registers	

TIMER 2		TIMER 2 Registers
▷ 1010 TMR2	0xFE10	Timer 2 Word
▷ 1010 TMR2CN0	0x44	Timer 2 Control 0
▷ iiii TMR2RL	0xEFB9	Timer 2 Reload Word

6-3 消費電流を測定してみる (Energy Profiler)

EFM8 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。ただしC8051の評価キットは対応していません。

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

Compatible Tools の中にある Energy Profiler を起動します。



Run メニューから、Run Demo を選択します。



EFM8SB1 Capsense Power Modes を選択します。そして Start をクリックします。

Demos for EFM8SB1 Sleepy Bee Starter Kit Board (BRDS	101A Rev A01)											
elect Demo Select a demo and the mode with which to run it.												
				<u> </u>							_	_
Name	Description	- Âl	· ()	 Choos 	e Progran	1 I		•		Select one		
EFM8SB1 Blinky EFM8SB1 Capacitive Sense for eval no overlay EFM8SB1 Capsense Circle Slider	Blinky Capacitive Sense for eval no overlay Capsense Circle Slider	Е .	iession Counter	Avg Current	Avg Power 9.09 mW	Total Energy 711.04 m.l	Time Span 153.82 s	Channel Primary &FM Channel	Mode	Save Reset Compare	Į	kvg
EFM8SB1 Capsense Power Modes	Capsense Power Modes		lelected Range	2.77 mA	9.11 mW	27.61 mJ	3.06 s	,		X		1
EPH85B1 Clock EPH85B1 Clock EPH85B1 CPT07078 Demo EPH85B1 CPT1125 Demo EPH85B1 DAta Looser Competence Network Model Example This program demonstrates two different catenat low poer. Bold a finger on one of the capacitive butchs Bold a finger on one of the capacitive butchs Duttons and undates the status of the three a down Demonstrates Demonstrates and the three a	Clock CPT007B Demo CPT1125 Demo Data Loneer He power modes: responsive vs and press FBS: responsive mode and press FBS: Low power mode. so to sample the capacitive pors on the LCD.	* * *		縦軸	:電波	充						1
Enable Code Correlation	V		100 nA nA							横軸	:時間	
?	Start Cancel		IRQ.	-6.00 s	-5	00 s	-6.00 s	-3.00 s		-2.00 s	1.00 s Y + -	1

このサンプルコードは、ボタンで EFM8 のパワーモードを切り替えて、その際の消費電流をモニタできる機能です。EFM8 の Starter Kit には、消費電力のモニタ機能がついています。

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

TecStar —

レジスタ設定を補助するツールとして Configurator(新しい製品向け)および Configuration Wizard2 (レガシー製品向け)が用意されています。ここでは、Configurator の使用方法を簡単にご紹介します。

Compatible Tools の中にある Hardware Configurator を起動します。

Getting Started Docum	entation	Compatible Tools	Resources
To view all available tools, click the 'Tools' button in the	main toolbar.		Add/Remove Tools
Simplicity IDE	Migrate Project	ts	Capacitive Sense Profiler Capacitive Sense Profiler is a tool that supports real-time analysis of sensor data from supported parts.
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	Flash Program Flash Programmer is a part	i mer utility to flash or erase any	Hardware Configurator Hardware Configurator is a peripheral, pin and crossbar configuration tool that generates initialization code organized into modes

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。

◆ ピン設定

ピンの設定は、DefaultMode Port I/O タブで行います。ピンを使用するペリフェラルにチェックを入れると、占有されるピンが紫色に変わります。



ピンの上でクリックすると、IO モード(Open Drain、Push-pull)、ドライブ・ストレングス、Pin Skip 設定な どを変更できます。



◆ ペリフェラル設定

ペリフェラルの設定は、DefaultMode Peripherals タブで行います。使用するペリフェラルを選択し、 Properties ウィンドウで詳細設定を行います。

*EFM8SB10F8G-A-QFN24.hwconf ×	- 0	⊞ Outline 🛛	
DefaultMode Peripherals		▲ Mode Transitions RESET → DefaultMode	
	<u>^</u>	▲ 3/2 DefaultMode	
		Peripherals	
Analog 🎓	:	4 🚑 Port I/O	
		CROSSBAR0	
Image: Comparator 0 Image: Capacitive Sensing Library Image: Current Reference Image: Current Reference		<i>F</i> ≥ <i></i>	
		🗆 *Properties 🛛 🏢 Perip	oheral Mapping
	. 11	Properties of ADC 0	
Clocking *	2	ADC 0	
	E	Property	Value
		⊿ View	
		View	Simple
	- 10	Enable ADC	Disabled
Communications *	:	Enable Burst Mode	Disabled
		Start of Conversion	Write to 1 of ADBUSY
SMBus 0 SPI 0 UART 0		Multiplexer Selection	
		Positive Input Selection	ADC0.31 (Ground)
		Configuration	
		Resolution	10-bit
Core		SARCLK (Actual)	2.500 MHz
		SAR Clock Source	2.500 MHz
Estamal Internute Estate Control Estat		SAR Clock Divider	1 (0x1)
		Enable 8-Bit Mode	Normal mode (10- or 12-bit)
		Gain Control	0.5x gain
		Result Shift and Justify	Right justified
		▲ Throughput	
Other *	2	Conversion Time	5.200 us
		Minimum Tracking Time	(1) 1.500 us
CRC	-	Maximum Throughput	149.254 ksps
Mode Transitions		Burst Mode Control	

設定が完了したら、画面上で右クリックして Generate Source を実行し、ソースコードに反映させます。

6-5 ピン設定やペリフェラル設定をしてみる (Configuration Wizard 2)

Configuration Wizard 2の使用方法を簡単にご紹介します。

Compatible Tools の中にある Configuration Wizard 2 を選択します。

Getting Started	Documentation	Compatible Tools	Resources		
To view all available tools, click the 'Tools' button in the	main toolbar.		Add/Remove Tools		
Simplicity IDE Launches the Simplicity IDE	Migrate Projects Migrate v3 projects		Configuration Wizard 2 Configuration Wizard 2 is the legacy C8051 configuration tool.		

New Project ウィンドウで MCU の型番を選択すると、Configuration Wizard 2 が起動します。

◆ ピン設定

Peripherals メニューから Port I/O を選択すると、下図のようなウィンドウが表示されます。使用する MCU の機能やピン数によって、画面が異なります。



番号	概要					
1	Crossbar の有効・無効(必ず有効にして下さい)					
1	内蔵 pull-up の有効・無効					
2	使用するペリフェラルの選択					
	アナログピン、デジタルピンの切り替え					
3	出カモード(push-pull, open-drain)の切り替え					
	ピンスキップの設定					

TecStar

I/O を使用するペリフェラル(UART, SPI など)を使用する場合、ポートはそれらペリフェラルに優先的に割り振られます。特定のポートを GPIO として使用したい場合には、ピンスキップの設定を行うことで優先的に確保できます。

下図(左)は、UART, SPI0, SMBusを使用する場合のポート設定です。P0.0~P0.7 が、これらペリフェ ラルに割り当てられています。

それに対し、下図(右)は、P0.0~P0.2をピンスキップ設定した場合です。ペリフェラルのポート割り当 てが P0.3~P1.2 へ変更されたことが判ります。



UART, SPI0, SMBus を選択。 P0.0~P0.7 が優先的に占有された。

◆ ペリフェラル設定

各ペリフェラルの設定は、Peripheralsメニューを使用して行っていきます。一つ設定が終るたびに、C コードのヘッダーファイルが更新されます。このヘッダーファイルには、設定に準じたペリフェラル初期化 ルーチンが記述されていますので、ソフトウェア設計の際にご使用頂けます。



ピンスキップ設定

6-6 Simplicity Studio ver.3 から ver.4 への移行

Simplicity Studio ver.3 から ver.4 へ簡単にプロジェクトを移行できるように、専用ツールが用意されています。

Compatible Tools の中にある Migrate Projects2 を選択し、ツールの指示に従って移行ください。

etting Started	Documentation	Compatible Tools	Resources
To view all available tools, click the 'Tools' button in the ma	ain toolbar.		Add/Remove Tools
Migrate Projects Migrate v3 projects			

TecStar

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 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



* 上記の説明では、EFM32 向けのコードを使用しています。EFM8/C8051 でも手順は同じです。

7-2 サンプルコードにペリフェラルを実装してみる (外部割込み)

EFM8/C8051 には、非常に多くのサンプルコードが用意されています。ADC のサンプルコード、 UART のサンプルコード、外部割込みのサンプルコード…といった具合に1つのペリフェラルにスポット を当てたサンプルコードが多く、ペリフェラルの機能・動作を学ぶのに非常に役立ちます。

実際のアプリケーション設計では、複数のペリフェラルを使用することがほとんどかと思いますので、 サンプルコードに別のペリフェラルを追加する手順を学ぶことは非常に有益です。

この章では、2つのサンプルコードを migration していく手順について紹介します。

題材として、「EFM8BB3 Blinky」「EFM8BB3 Enternal Interrupts」という2つのサンプルコードを使用します。どちらも EFM8BB3 STK 向けのサンプルコードです。

「EFM8BB3 Blinky」はタイマに連動して LED の色が次々と変っていくサンプルコードです。

「EFM8BB3 Enternal Interrupts」は、ボタンを押すと外部割込みが生じ、LED の色を変えるサンプルコードです。

「EFM8BB3 Blinky」に外部割込み(External Interrupts)を実装し、LED の色が次々変わるのをスター ト/ストップする機能を実装してみます。具体的には、ボタン 0 を押すと Timer2 をストップ(つまり LED の色変更も停止)、ボタン 1 を押すと Timer2 をスタート(つまり LED の色変更を再開)、という機能を実 装します。

大まかな流れとしては、

- サンプルコードを理解する(7-2-1、7-2-2)
- 「EFM8BB3 Enternal Interrupts」のペリフェラル設定を、「EFM8BB3 Blinky」に移植する(7-2-3、 7-2-4)
- アプリを実装する(7-2-5)

です。

7-2-1 サンプルコードを理解する(EFM8BB3 Blinky)

EFM8BB3_Blinkyは、初期化を行ったあと、while ループの中で延々とTimer割り込みを待ち続ける、 非常にシンプルなサンプルコードです。

初期化には、enter_DefaultMode_from_RESET() 関数を使っています。この関数は InitDevice.c の中 で定義されており、更に各ペリフェラルの初期化関数を呼び出しています。例えば、WDT_0_enter…は Watchdog Timer の初期化関数で、TIMER16_2_enter…は Timer2 の初期化関数です。

EFM8BB3_Blinky.c

TecStar =

```
67 //-----
             _____
68 // Main Routine
69 //-----
                       70⊖ void main (void)
71 {
72
    enter_DefaultMode_from_RESET();
73
74
    DISP_EN = DISP_BC_DRIVEN;
                          // Display not driven by EFM8
75
76
    IE EA = 1;
                             // Enable global interrupts
77
78
    while (1) {}
                              // Spin forever
79 }
```

InitDevice.c

```
19@//------
20 // enter_DefaultMode_from_RESET
21 //-----
22
extern void enter_DefaultMode_from_RESET(void) {
    // $[Config Calls]
23
    // Save the SFRPAGE
24
25
    uint8_t SFRPAGE_save = SFRPAGE;
    WDT_0_enter_DefaultMode_from_RESET();
26
    PORTS_1_enter_DefaultMode_from_RESET();
27
    PBCFG_0_enter_DefaultMode_from_RESET();
28
    TIMER16_2_enter_DefaultMode_from_RESET();
29
30
    INTERRUPT_0_enter_DefaultMode_from_RESET();
    // Restore the SFRPAGE
31
32
    SFRPAGE = SFRPAGE save;
33
    // [Config Calls]$
34
35 }
```

割り込み処理は Interrupts.c で行っています。Timer2 のオーバフローが発生する度に、LED の色を 次々と変えています。色は全 7 種で、割り込み \Rightarrow case0 の色 \Rightarrow 割り込み \Rightarrow case1 の色 \Rightarrow 割り込 み \Rightarrow case2 の色…といった具合に動きます。

Interrupt.c

```
_____
24 //-----
25 // TIMER2 ISR
26 //-----
27 //
28 // TIMER2 ISR Content goes here. Remember to clear flag bits:
29 // TMR2CN::TF2H (Timer # High Byte Overflow Flag)
30 // TMR2CN::TF2L (Timer # Low Byte Overflow Flag)
31 //
32 // This routine changes the state of the LED whenever Timer2 overflows.
33 //
34 //-----
35 SI_INTERRUPT (TIMER2_ISR, TIMER2_IRQn)
36 {
37
      TMR2CN0 TF2H = 0;
                                  // Clear Timer2 interrupt flag
38
39
      switch (LEDCOUNT)
40
      {
41
         case 0:
42
           LED0 = 1;
43
           LED1 = 0;
44
           LED2 = 0;
45
           break;
46
        case 1:
47
           LED0 = 0;
48
           LED1 = 1;
49
           LED2 = 0;
50
           break;
        case 2:
51
           LED0 = 0;
52
           LED1 = 0;
53
54
           LED2 = 1;
55
           break;
        case 3:
56
           LED0 = 1;
57
           LED1 = 1;
58
           LED2 = 0;
59
60
           break;
        case 4:
61
           LED0 = 0;
62
63
           LED1 = 1;
64
           LED2 = 1;
65
           break;
66
        case 5:
67
          LED0 = 1;
68
           LED1 = 0;
69
           LED2 = 1;
70
           break;
71
         case 6:
72
           LED0 = 1;
73
           LED1 = 1;
           LED2 = 1;
74
75
           break;
76
      }
77
78
      if (LEDCOUNT <= 6) LEDCOUNT++;
79
      else LEDCOUNT = 0;
80 }
0.1
```

7-2-2 サンプルコードを理解する (EFM8BB3_ExternalInterrupts)

Blinkyと同様に、初期化を行ったあと、while ループの中で延々と外部割り込みを待ち続ける、非常にシンプルなサンプルコードです。

初期化には、やはり Blinky と同様に enter_DefaultMode_from_RESET() 関数を使っています。

各ペリフェラルの初期化関数の中に EXTINT_0_enter_DefaultMode_from_RESET() という関数が ありますが、これが外部割込みの初期化関数です。EXTINT_0_enter_DefaultMode_from_RESET()は、 InitDevice.c の後半で定義されています。

EFM8BB3_ExternalInterrupts.c

```
73 //-----
                          -----
74 // Main Routine
75 //-----
                 76 void main (void)
77 {
78
    enter_DefaultMode_from_RESET();
79
80
    DISP_EN = DISP_BC_DRIVEN;
                       // EFM8 does not drive display
81
    IE_EA = 1;
82
83
    while(1);
                              // Infinite while loop waiting for
84
                              // an interrupt from /INT0 or /INT1
85
86 }
```

InitDevice.c

TecStar =

```
19@//------
20 // enter DefaultMode from RESET
21 //------
22 extern void enter_DefaultMode_from_RESET(void) {
23
     // $[Config Calls]
      // Save the SFRPAGE
24
25
     uint8 t SFRPAGE save = SFRPAGE;
26
     WDT_0_enter_DefaultMode_from_RESET();
27
     PORTS_0_enter_DefaultMode_from_RESET();
28
      PORTS_1_enter_DefaultMode_from_RESET();
29
      PORTS_2_enter_DefaultMode_from_RESET();
30
      PBCFG_0_enter_DefaultMode_from_RESET();
      RSTSRC_0_enter_DefaultMode_from_RESET();
31
      CLOCK_0_enter_DefaultMode_from_RESET();
32
      TIMER01_0_enter_DefaultMode_from_RESET();
33
34
      TIMER16_2_enter_DefaultMode_from_RESET();
35
      TIMER_SETUP_0_enter_DefaultMode_from_RESET();
36
      EXTINT_0_enter_DefaultMode_from_RESET();
      INTERRUPT_0_enter_DefaultMode_from_RESET();
37
      // Restore the SFRPAGE
38
      SFRPAGE = SFRPAGE_save;
39
40
      // [Config Calls]$
41
42 }
```

割り込み処理は Interrupts.c で行っています。INT0(外部割込み 0)が発生したら、LED_GREEN(という名称を付けた P1_4 ピン)を反転し、INT1(外部割込み 1)が発生したら、LED_BLUE(という名称を付けた P1_5 ピン)を反転する、という動作を行います。

Interrupt.c

```
10 // USER INCLUDES
11 #include <SI_EFM8BB3_Register_Enums.h>
12
13 //-----
                         -----
14 // Pin Declarations
                        15 //-----

        16
        SI_SBIT (LED_GREEN, SFR_P1, 4);
        // green LED

        17
        SI_SBIT (LED_BLUE, SFR_P1, 5);
        // blue LED

18
19@//-----
20 // INTØ ISR
21 //-----
               -----
22 //
23 // INTO ISR Content goes here. Remember to clear flag bits:
24 // TCON::IE0 (External Interrupt 0)
25 //
26 // Whenever a negative edge appears on P0.2, toggle LED_GREEN.
27 // The interrupt pending flag is automatically cleared by vectoring to the ISR
28 //
29 //-----
30 SI_INTERRUPT (INT0_ISR, INT0_IRQn)
31 {
     LED GREEN = !LED GREEN;
32
33 }
34
35@//-----
36 // INT1 ISR
37 //-----
38 //
39 // INT1 ISR Content goes here. Remember to clear flag bits:
40 // TCON::IE1 (External Interrupt 1)
41 //
42 // Whenever a negative edge appears on P0.3, toggle LED_BLUE.
43 // The interrupt pending flag is automatically cleared by vectoring to the ISR
44 //
45 //-----
46 SI_INTERRUPT (INT1_ISR, INT1_IRQn)
47 {
    LED_BLUE = !LED_BLUE;
48
49 }
50
```

7-2-3 ペリフェラル設定を移植する (EFM8BB3_ExternalInterrupts の設定を読み取る)

2 つのサンプルコードをロードします。それぞれに.hwconf ファイルが生成されますが、これが Hardware Configurator のプロジェクトファイルです。まずは EFM8BB3_ExternalInterrupts の方から見て いきます。



Default Mode Port I/O の設定から見ていきます。外部割込み用に、P0.2(INT0)とP0.3(INT1)の2ピンを使用しています。Proeprtiesを見てみると、特に変更点はなくデフォルトのままです。(変更点があれば色が変わっています)

D-foulated - D-+ 1/0, CROSSRADO	🔺 🥕 Mode Tr	ansitions			
Detaultiviode Port I/O: CKUSSBAKU	RESE	T → DefaultMode			
	4 👬 DefaultN	Mode			
	⊳ 🚮 Perip	herals			
0 7	4 🛎 Port	1/0			
	> # 0	RUSSBARU			
	9 Pi	82			
32,P0.1 34,P0.2 29,P0.4 28,P0.5 26,P0.5 25,P1.0 25,P1.0	*Properties	2		Ē	
1/P0.0	Properties of Pi	0.2			
(Top View)	PO			Value	
2/10	A Settings				
3/VDD (22/P1.3	IOMode	2		Digital OpenDrain I/O	
	Latch			High	
4/RST / 32-pin QFN (21/P1.4	Mask V	alue		Ignored	
	Match \	Value		High	
4X4 mm	Skip			Not skipped	
6/P3.4	Label				
7/P3.3 (18/P1.7					
GND					
8/P3.2 (17/P2.0					
9/P3.1 10/P3.6 13/P2.6 13/P2.6 14/P2.3 16/P2.1					
	I Port I/O Ma	pping 🛛			7 7 8
	DefaultMode	: CROSSBAR0			
	UART 0	🔲 Data	UART0_TX		*
P2.4			UART0_RX		
	SPI 0	Clock / Data	SPI0_SCK		=
A Made Tanaching A Defailth is De 110 a an 11 Defailth de Decisional			SPI0_MISO		
Mode Transitions Tar DefaultMode Port I/O			SPI0_MOSI		

なお、LED を制御するために使っている P1.4、P1.5 は、IOMode を Digital Push-Pull Output に変更していました。今回は EFM8BB3_Blinky の IO 設定を流用しますので、下記情報は使用しません。

	<i>∛</i> PB3	
/P1.0	T *Properties 🕅	B 🛃 🗸 🗆 🗖
	Properties of P1.4	
24/P1.1	P1	
23/P1.2	Property	Value
(22/P1.3	IOMode	Digital Push-Pull Output
	Latch	High
(21/P1.4	Mask Value	Ignored
	Match Value	High
(20/P1.5	Skip	Not skipped
	Label	
(19/P1.6		
(18/P1.7		
(17/P2.0		
22.1		
	I Port I/O Mapping 🔀	Ÿ ▽ □ □

次に DefaultMode Peripherals の設定を見ます。External Interrupts と Interrupts にチェックが入って います。その他のペリフェラルにもチェックが入っていますが、External Interrupt と直接関係しないので 割愛します。 External Interrupts と Interrupt の 2 つを掘り下げてみていきます。

Communications *	
I2C Slave 0 SMBus 0 SPI 0 UART 0 UART 1	
Core *	ĺ
C2DEBUG_0 Core External Interrupts Interrupts Reset Sources	
SFR Paging	
Other *	
Configurable Logic	
Power *	
Mode Transitions	

External Interrupt では、INT0をP0.2に、INT1をP0.3に割り当てています。

Core 2 Co	•	Properties Properties of External Interrupts External Interrupts	
		Property	Value
	E	INTO Polarity	Active low
	- 11	INTO Port Pin Selection	P0.2
		INT1 Polarity	Active low
Other	:	INT1 Port Pin Selection	P0.3
Power	:		
Mode Transitions 😩 DefaultMode Port I/O 🛛 📲 DefaultMode Peripherals 🕱	Ŧ		m

Interrupts では、Interrupt Enable(割込みの有効・無効設定)で、2つの外部割込みを有効(Enabled)に 設定しています。

Core		T *Properties 🛛	🖾 🗾 🖂 🖂
		Properties of Interrupts	
C2DEBUG_0 Core External Interrupts Interrupts Reset Sources		[Interrupt Enables] Interrupt Priorities	
		Property	Value ^
SFR Paging	=	Enable I2C0 Slave Interrupt	Disabled
		Enable Timer 4 Interrupt	Disabled
		Enable Timer 5 Interrupt	Disabled
		Enable UART1 Interrupt	Disabled
Other		✓ Interrupt Enable	
		Enable All Interrupts	Disabled
		Enable External 0 Interrupt	Enabled
		Enable External 1 Interrupt	Enabled
		Enable SPI0 Interrupt	Disabled

7-2-4 ペリフェラル設定を移植する (EFM8BB3_Blinky に設定を移植する)

7-2-3 で読み取った設定を、今度は EFM8BB3_Blinky に反映していきます。Blinky の.hwconf ファイ ルを開きます。DefaultMode Peripherals の設定を見ると、External Interrupts は未チェックです。Interrupts にはチェックが入っていますが、外部割込みは無効(Disabled)のままです。

		🐓 PB3		
Core		■ *Properties 🕱		
C2DEBUG_0		Properties of Interrupts		
		Interrupt Enables Interrupt Priorities		
SFR Paging		Property	Value	
		Enable I2C0 Slave Interrupt	Disabled	
	-	Enable Timer 4 Interrupt	Disabled	
	=	Enable Timer 5 Interrupt	Disabled	
	. 11	Enable UART1 Interrupt	Disabled	
Uther *		▲ Interrupt Enable		
		Enable All Interrupts	Enabled	
Configurable Logic CRC		Enable External 0 Interrupt	Disabled	
		Enable External 1 Interrupt	Disabled	
		Enable SPI0 Interrupt	Disabled	
(+)		Enable Timer 0 Interrupt	Disabled	E
Power *		Enable Timer 1 Interrupt	Disabled	
		Enable Timer 2 Interrupt	Enabled	
PMU Supply Monitor Voltage Regulator		Enable UART0 Interrupt	Disabled	
Mode Transitions 💈 DefaultMode Port I/O 🛛 👖 DefaultMode Peripherals 😒		< III		

External Interrupts にチェックを入れ、INTO Port Pin Selection を P0.2 に、INTO Port Pin Selection を P0.3 に変更します。

		<i>F</i> ≥ PB3	
Core *			📑 💀 🗸 🗖
C2DEBUG_0		Properties of External Interrupts	
		External Interrupts	
SFR Paging		Property	Value
		INTO/INT1 Configuration	
	-	INT0 Polarity	Active low
	-	INTO Port Pin Selection	P0.2
Other		INT1 Polarity	Active low
		INT1 Port Pin Selection	P0.3

次に Interrupts で、Enable External 0 Interrupt と Enable External 1 Interrupt を Enabled に変更します。

		<i>y</i> PB3		
Core		Properties X	🛃 🗔 🔻	- E
C2DEBUG_0 Core External Interrupts Interrupts Reset Sources		Properties of Interrupts Interrupt Enables Interrupt Priorities		
SFR Paging		Property	Value	,
		Enable I2C0 Slave Interrupt	Disabled	
	-	Enable Timer 4 Interrupt	Disabled	
	=	Enable Timer 5 Interrupt	Disabled	
		Enable UART1 Interrupt	Disabled	
		Enable All Interrupts	Enabled	
Configurable Logic CRC		Enable External 0 Interrupt	Enabled	
		Enable External 1 Interrupt	Enabled	
		Enable SPI0 Interrupt	Disabled	
(+)		Enable Timer 0 Interrupt	Disabled	8
Power *		Enable Timer 1 Interrupt	Disabled	
		Enable Timer 2 Interrupt	Enabled	
PMU Supply Monitor Voltage Regulator	-	Enable UART0 Interrupt	Disabled	

次に Default Mode Port I/O です。今回は特に変更箇所はありませんが、INT0 と INT1 用にピンがリ ザーブされたことが確認できます。

DefaultMode Port I/O: PBCFG_0	A Mode Transitions RESET → DefaultMode All D = Could default	
/INT0 /INT1		
() ()		
1/P0.0 (Top)/(out) 24/P1.1	T *Properties 🛛	📑 🗔 🔻 🗖 🗖
2/v0 (100 View) (22/P12	Properties of P0.2	
	Draparty	Value
	- Fotback	value
(P3.7/) 4x4 mm (20/P1.5	A Settings	Digital OpenDrain I/O
	Latch	High
6/P3.4	Mask Value	Ignored
7/P3.3	Match Value	High
B/P3.2 GND (17/P2.0	Skip	Not skipped
	Label	
P3.1 P2.5 P2.1 P2.1		
9/ 11/ 11/ 11/ 11/ 11/ 11/ 11/ 11/		
Mode Transitions a DefaultMode Port I/O 🛛 📲 DefaultMode Peripherals	•	4

設定が完了したら、行った設定をソースコードに反映させます。Hardware Configurator 上で適当な場所で右クリックし、Generate Source を選択します。

				▲ A Mode Transitions RESET → Default
				▲ SefaultMode
				Image: Peripherals
				a 🔄 Port I/O
				▷ # CROSSBAR0
,				.
	E	Save As Picture		🐓 PB3
		Zoom To	•	
(24/P1.1				■ *Properties
23/P1.2		Skip Pins		Properties of PBCFG_0
		Unskip Pins		Port Config
(22/P1.3		IOMode	•	
21/P1.4		Reset PBCFG_0		Property
(And F	\Leftrightarrow	Undo		▲ Settings
20/11.3	\$	Redo	_	Disable Port I/O We
19/P1.6	-21	Generate Source		Enable Crossbar
(18/P1 7		Show Properties		Port 0 Drive Streng
	5	Show Outline View		Port 1 Drive Streng
(17/P2.0	-	Show outline them		Port 2 Drive Streng
		Module Configuration Report		Port 3 Drive Streng
5		Pin Configuration Report		
	24/P1.1 23/P1.2 22/P1.3 21/P1.4 20/P1.5 19/P1.6 18/P1.7 17/P2.0	24/P1.1 23/P1.2 22/P1.3 21/P1.4 20/P1.5 15/P1.6 15/P1.7 15/P1.7 15/P1.7	24/P1.1 Save As Picture 22/P1.2 Zoom To 22/P1.3 Unskip Pins 10Mode Reset PBCFG_0 11/P1.4 Wide 12/P1.5 Generate Source 13/P1.7 Show Properties 13/P1.7 Module Configuration Report 17/P2.0 Module Configuration Report	24/P1.1 Save As Picture 22/P1.2 Zoom To 22/P1.2 Unskip Pins 10Mode N 21/P1.4 Reset PBCFG_0 20/P1.5 Undo 20/P1.6 Show Properties 18/P1.7 Show Properties 17/P2.0 Module Configuration Report I Pin Configuration Report

この Generate Source で、InitDevice.cとInterrupts.cが更新されています。その更新内容を見ていきましょう。

InitDevice.c の初期化関数(enter_DefaultMode_from_RESET)を、変更前(7-2-1)のものと比較すると、 EXTINT_0_enter_DefaultMode_from_RESET() が追加されていることが確認できます。

InitDevice.c

19⊝ //===================================
20 // enter_DefaultMode_from_RESET
21 //
22⊖ extern void enter_DefaultMode_from_RESET(void) {
23 // \$[Config Calls]
24 // Save the SFRPAGE
<pre>25 uint8_t SFRPAGE_save = SFRPAGE;</pre>
<pre>26 WDT_0_enter_DefaultMode_from_RESET();</pre>
<pre>27 PORTS_1_enter_DefaultMode_from_RESET();</pre>
<pre>28 PBCFG_0_enter_DefaultMode_from_RESET();</pre>
29 TIMER16 2 enter DefaultMode from RESET();
<pre>30 EXTINT 0 enter DefaultMode from RESET();</pre>
<pre>31 INTERRUPT_0_enter_DetaultMode_trom_RESET();</pre>
32 // Restore the SFRPAGE
<pre>33 SFRPAGE = SFRPAGE_save;</pre>
34 // [Config Calls]\$
35
36 }

また、Interrupt.cでは、外部割込み(INT0、INT1)の割込みハンドラが追加されています。処理内容が何もありませんので、ここにユーザコードを追記していきます。

Interrupts.c

```
81@//-----
82 // INT0_ISR
83 //-----
84 //
85 // INTO ISR Content goes here. Remember to clear flag bits:
86 // TCON::IE0 (External Interrupt 0)
87 //
88 //-----
89 SI_INTERRUPT (INT0_ISR, INT0_IRQn)
90 {
91
92 }
93
94⊖ //-----
95 // INT1 ISR
96 //-----
97 //
98 // INT1 ISR Content goes here. Remember to clear flag bits:
99 // TCON::IE1 (External Interrupt 1)
100 //
101 //-----
102 SI_INTERRUPT (INT1_ISR, INT1_IRQn)
103 {
104
105 }
```

7-2-5 アプリを実装する

新たに追加された割込みハンドラ(7-2-4参照)の中では、タイマ2のスタート/ストップを制御したいで す。タイマ2の仕様はリファレンス・マニュアルを参照します。

https://www.silabs.com/documents/public/reference-manuals/efm8bb3-rm.pdf

21.4.13 TMR2CN0: Timer 2 Control 0

Bit	7	6	5	4	3	2	1	0			
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2XCLK				
Access	RW	RW	RW	RW	RW	RW	RW				
Reset	0	0	0	0	0	0	0x0				
SFR Page = 0x0, 0x10; SFR Address: 0xC8 (bit-addressable)											
Bit	Name	Reset	Acces	ccess Description							
7	TF2H	0	RW	Timer 2 Hig	Timer 2 High Byte Overflow Flag.						
	Set by hardwa	re when the Tin	ner 2 high byte	overflows from	0xFF to 0x00. In	n 16-bit mode, t	his will occur wi	hen Timer 2			
2	TR2	0	RW	Timer 2 Ru	n Control.						
	Timer 2 is ena split mode.	bled by setting	this bit to 1. In 8	-bit mode, this b	it enables/disab	oles TMR2H only	/; TMR2L is alw	ays enabled in			
	TOVOUV					•					

Timer2 の制御レジスタ(TMR2CN0)の説明を読むと、bit2 に TR2 というビットがあり、これを使ってス タート/ストップの制御が行えます。TR2を1に設定するとタイマ2が有効(enabled)になります。つまり、 0 に設定するとタイマ2 が無効になります。

ここで注目したいのは、"bit-addressable"の文字です。バイト単位ではなく、ビット単位でアクセスができますので、わざわざリードモディファイライトする必要がありません。

EFM8/C8051 が持つレジスタは、全て定義ファイル内で宣言されています。EFM8BB3 の場合は、 SI_EFM8BB3_Register_Enums.h および SI_EFM8BB3_Defs.h が定義ファイルになります。

<mark>Interrupt.c</mark> (EFM8BB3_Blinky.c、InitDevice.c でも使われています)

```
9 // USER INCLUDES

10 #include <SI_EFM8BB3_Register_Enums.h>

II

SI_EFM8BB3_Register_Enums.h

49

50 //Standard device includes

51 #include "SI_EFM8BB3_Defs.h"

52 //
```

使用したい TR2 ビットですが、SI_EFM8BB3_Defs.h の中で定義されており、TMR2CN0_TR2 という 名称になっています。

518	// TMR20	NØ (Timer 2 Contro	ol 0)				
519	#define	SFR_TMR2CN0 0xC8					
520	SI_SBIT	(TMR2CN0_T2XCLK0,	SFR_TMR2CN0,	0);	///<	Timer	2 External Clock Select Bit 0
521	CT CRTT	(TMDOCNO TOXCLV1	CED TMDOCNO	1).	1112	Timon	2 External Clock Select Bit 1
522	SI_SBIT	(TMR2CN0_TR2,	SFR_TMR2CN0,	2);	///<	Timer	2 Run Control
523	21_2811	(IMRZCN0_IZSPLII,	SFR_IMRZCN0,	3);	111<	limer	2 Spiit Mode Enable
524	SI_SBIT	(TMR2CN0_TF2CEN,	SFR_TMR2CN0,	4);	///<	Timer	2 Capture Enable
525	SI_SBIT	(TMR2CN0_TF2LEN,	SFR_TMR2CN0,	5);	///<	Timer	2 Low Byte Interrupt Enable
526	SI_SBIT	(TMR2CN0_TF2L,	SFR_TMR2CN0,	6);	///<	Timer	2 Low Byte Overflow Flag
527	SI_SBIT	(TMR2CN0_TF2H,	SFR_TMR2CN0,	7);	///<	Timer	2 High Byte Overflow Flag

Interrupt.c の外部割込みの割込みハンドラに、TMR2CN0_TR2の制御を盛り込みます。値を0にするとストップ、1にするとスタートです。

Interrupt.c

TecStar =

_ _ .

```
81@//-----
82 // INTØ ISR
83 //-----
          -----
84 //
85 // INTØ ISR Content goes here. Remember to clear flag bits:
86 // TCON::IE0 (External Interrupt 0)
87 //
88 //-----
89 SI_INTERRUPT (INTO_ISR, INTO_IRQn)
90 {
91
    TMR2CN0_TR2 = 0; //ボタン0を押すとストップ
92 }
93
949 //-----
95 // INT1_ISR
96 //-----
97 //
98 // INT1 ISR Content goes here. Remember to clear flag bits:
99 // TCON::IE1 (External Interrupt 1)
100 //
101 //-----
102 SI_INTERRUPT (INT1_ISR, INT1_IRQn)
103 {
104
   TMR2CN0 TR2 = 1; //ボタン1を押すとスタート
105 }
```

その後、ビルドして、ダウンロードすると、LEDの点灯スタート/ストップをボタンで制御できるようになったことが確認できます。

改版履歴

Version	改定日	改定内容
1.0	2011年07月	•新規作成
1.3	2015年06月	・EFM8 に対応。マクニカオンラインで公開
2.0	2016年12月	・Simplicity Studio ver.4 に対応
2.1	2017年03月	最新の Simplicity Studio に合わせて説明を一部変更
2.2	2018年03月	最新の Simplicity Studio に合わせて説明を一部変更。7
		章(ソフトウェア設計)追加

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