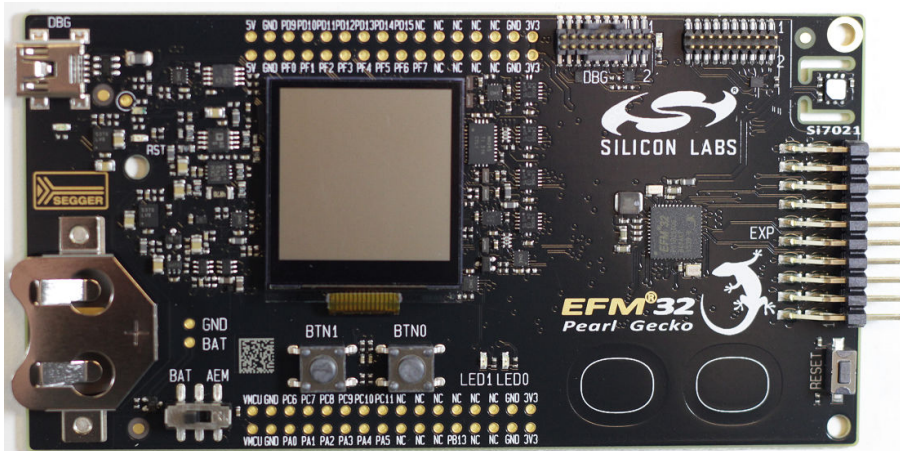# UG154: EFM32 Pearl Gecko Starter Kit User's Guide

The SLSTK3401A Starter Kit is an excellent starting point to get familiar with the EFM32™ Pearl Gecko Microcontroller.

The Starter Kit contains sensors and peripherals demonstrating some of the Pearl Gecko's many capabilities. The kit provides all necessary tools for developing an EFM32 Pearl Gecko application.

1. Introduction

1.1 Description

The SLSTK3401A is an excellent starting point to get familiar with the EFM32 Pearl Gecko Microcontrollers. The kit contains sensors and peripherals demonstrating some of the MCU's many capabilities. The kit can also serve as a starting point for application development.

In additon to supporting application development on the starter kit itself, the board is also a fully featured debugger and energy monitoring tool that can be used with external applications.

1.2 Features

* EFM32PG1B200F256GM48 MCU with 256 kB Flash and 32 kB RAM.
* Advanced Energy Monitoring system for precise current tracking.
* Integrated Segger J-Link USB debugger/emulator with the possiblity to debug external Silicon Labs devices.
* 20 pin expansion header.
* Breakout pads for easy access to I/O pins.
* Power sources including USB and CR2032 battery.
* Silicon Labs Si7021 Relative Humidity and Temperature Sensor
* Ultra low power 128x128 pixel Memory-LCD
* 2 push buttons and 2 LEDs connected to EFM32 for user interaction.
* Crystals for LFXO and HFXO: 32.768 kHz and 40.000 MHz.

1.3 Getting Started

Hardware

To set up the hardware for the SLSTK3401A kit:

1. Provide power to the board by connecting the DBG USB connector to the PC using the provided mini-USB cable.
2. Ensure that the power selector switch is in the AEM position.

Software

The first step to get started with your new SLSTK3401A is to go to

http://www.silabs.com/simplicity

The Simplicity Studio software package contains all the tools, drivers, software examples and documentation needed to use the EFM32 Pearl Gecko Starter Kit. The board comes pre-loaded with a default application, Blinky, to play with while the software downloads.

After downloading the latest version of Simplicity Studio and installing using the [**Full**] or [**EFM32**] options:

1. Click the [**Refresh detected hardware**] button and select the EFM32 Pearl Gecko Starter Kit under [**Detected Hardware**].
2. Click the [**Demos**] tile under [**Tools**] to load the available demos.
3. Click the Blinky demo and click [**Start**] to download and run the demo.

Additional demos showcasing the various features of the Pearl Gecko are also available under the same tile in Simplicity Studio.

2. Kit Block Diagram

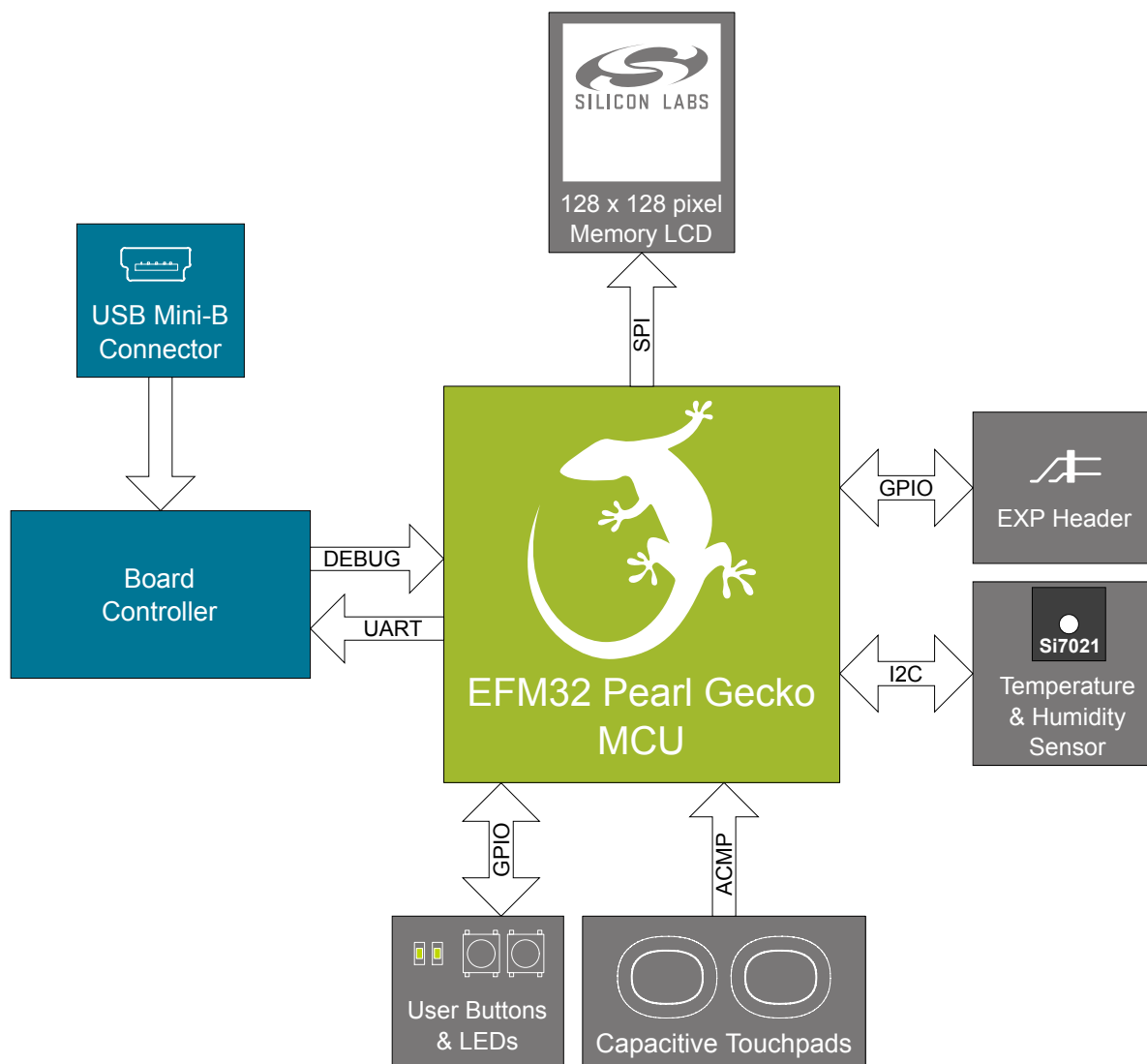An overview of the EFM32 Pearl Gecko Starter Kit is shown in the figure below.

Figure 2.1. Kit Block Diagram

3. Kit Hardware Layout

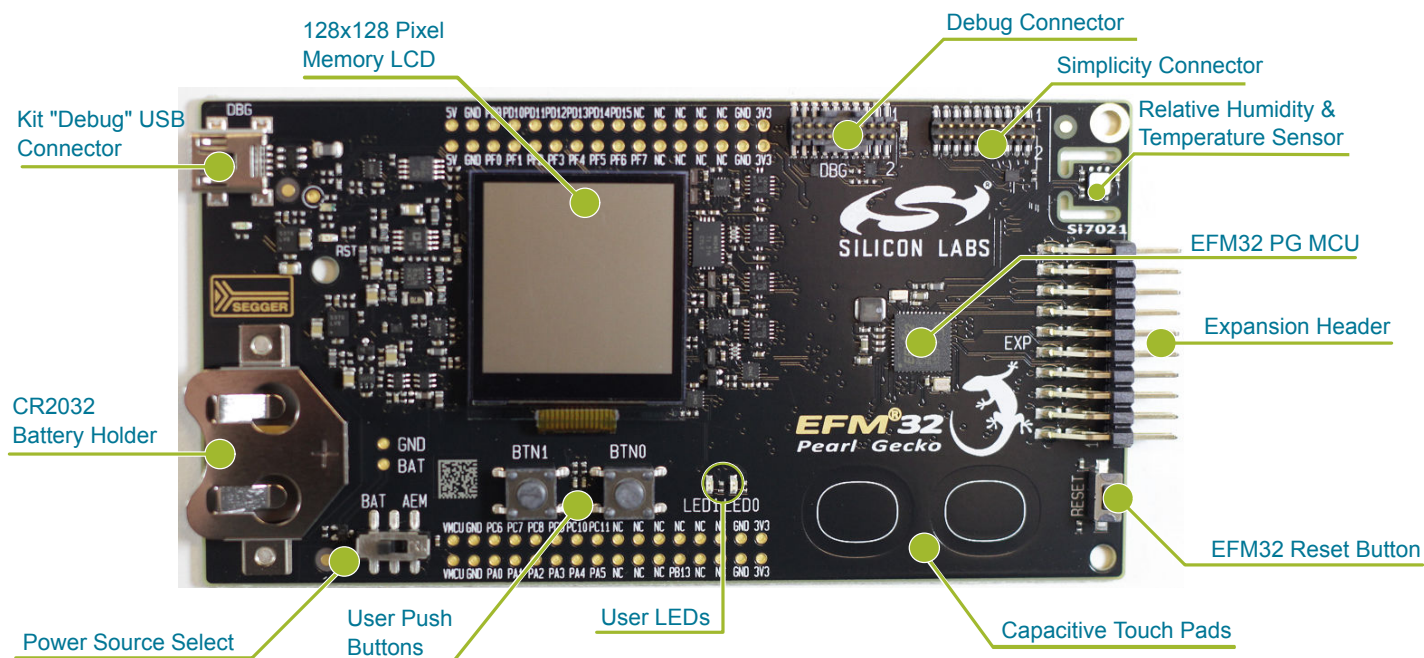The layout of the EFM32 Pearl Gecko Starter Kit is shown below.

Figure 3.1. SLSTK3401A Hardware Layout

4. Connectors

4.1 Breakout Pads

Most of the EFM32's GPIO pins are available on two pin header rows at the top and bottom edges of the board. These have a standard 2.54 mm pitch, and pin headers can be soldered in if required. In addition to the I/O pins, connections to the different power rails and ground are also provided. Note that some of the pins are used for kit peripherals or features, and may not be available for a custom application without trade-offs.

The figure below shows the pinout of these "breakout pads", as well as the pinout of the "Expansion Header" situated on the right-hand side of the board. The expansion header is further explained in the next section. The breakout pad connections are also printed in silk screen next to each pin for easy reference.
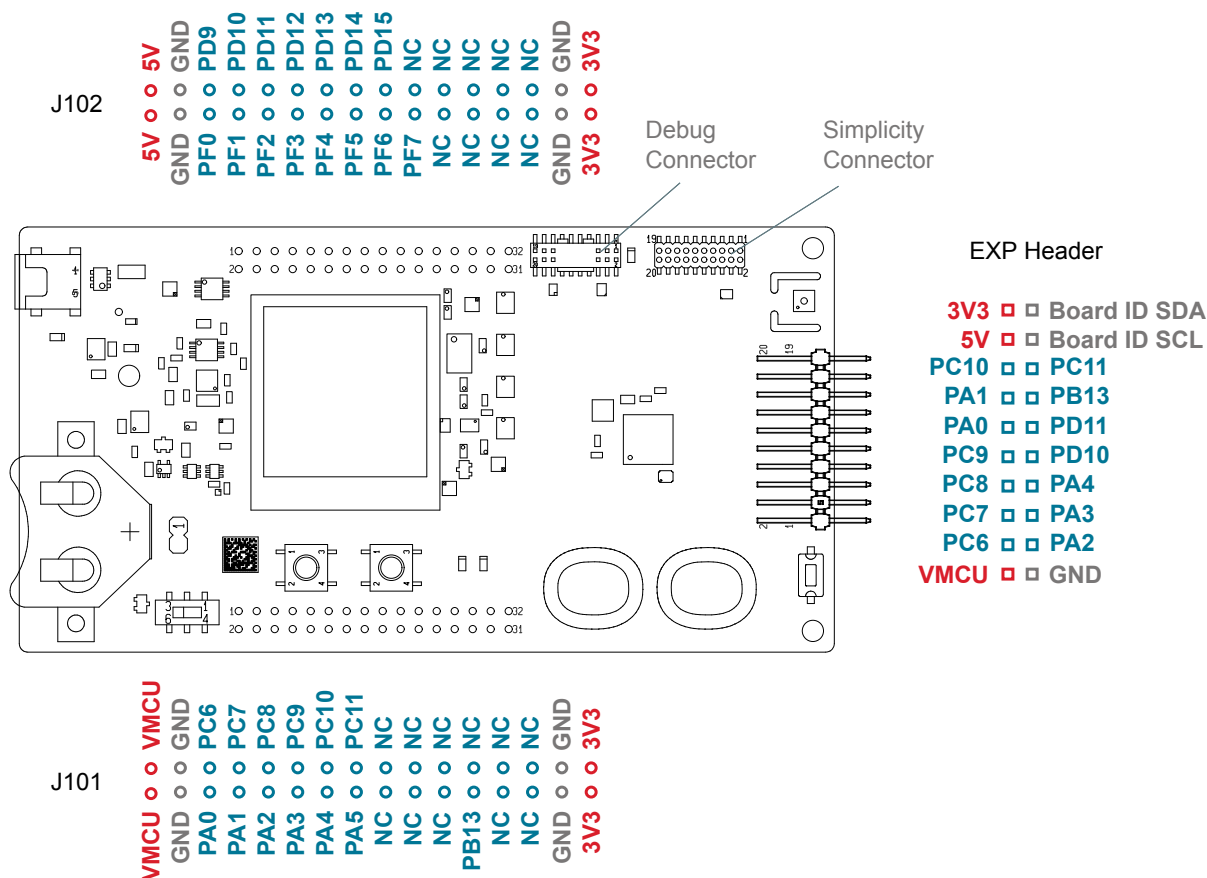

Figure 4.1. Breakout Pads and Expansion Header

The table below shows the connections of each pin of the breakout pads. They also show which kit peripherals or features are connected to the different pins.

Table 4.1. Bottom Row (J101) Pinout

Pin	EFM32 I/O pin	Kit peripheral/feature connection		Pin	EFM32 I/O pin	Kit peripheral/feature connection
1	VMCU	EFM32 voltage domain		2	VMCU	EFM32 voltage domain
3	GND	Ground		4	GND	Ground
5	PA0	EFM_BC_TX / EXP12		6	PC6	EFM_DISP_MOSI / EXP4
7	PA1	EFM_BC_RX / EXP14		8	PC7	EXP6
9	PA2	EFM_BC_CTS / EXP3		10	PC8	EFM_DISP_SCLK / EXP8

Pin	EFM32 I/O pin	Kit peripheral/feature connection	Pin	EFM32 I/O pin	Kit peripheral/feature connection
11	PA3	EFM_BC_RTS / EXP5	12	PC9	EXP10
13	PA4	EXP7	14	PC10	SENSOR_I2C_SDA / EXP16
15	PA5	EFM_BC_EN	16	PC11	SENSOR_I2C_SCL / EXP15
17	-	Not connected	18	-	Not connected
19	-	Not connected	20	-	Not connected
21	-	Not connected	22	-	Not connected
23	PB13	EXP13	24	-	Not connected
25	-	Not connected	26	-	Not connected
27	-	Not connected	28	-	Not connected
29	GND	Ground	30	GND	Ground
31	3V3	Board controller supply	32	3V3	Board controller supply

Table 4.2. Top Row (J102) Pinout

Pin	EFM32 I/O pin	Kit peripheral/feature connection	Pin	EFM32 I/O pin	Kit peripheral/feature connection
1	5V	Board USB voltage	2	5V	Board USB voltage
3	GND	Ground	4	GND	Ground
5	PF0	DEBUG_TCK_SWCLK	6	PD9	SENSOR_ENABLE
7	PF1	DEBUG_TMS_SWDIO	8	PD10	EXP9
9	PF2	DEBUG_TDO_SWO	10	PD11	EXP11
11	PF3	DEBUG_TDI	12	PD12	
13	PF4	UIF_LED0	14	PD13	EFM_DISP_COM
15	PF5	UIF_LED1	16	PD14	EFM_DISP_CS
17	PF6	UIF_BUTTON0	18	PD15	EFM_DISP_ENABLE
19	PF7	UIF_BUTTON1	20	-	Not connected
21	-	Not connected	22	-	Not connected
23	-	Not connected	24	-	Not connected
25	-	Not connected	26	-	Not connected
27	-	Not connected	28	-	Not connected
29	GND	Ground	30	GND	Ground
31	3V3	Board controller supply	32	3V3	Board controller supply

4.2 Expansion Header

On the right hand side of the board an angled 20 pin expansion header is provided to allow connection of peripherals or plugin boards. The connector contains a number of I/O pins that can be used with most of the EFM32 Pearl Gecko's features. Additionally, the VMCU, 3V3 and 5V power rails are also exported.

The connector follows a standard which ensures that commonly used peripherals such as an SPI, a UART and an I2C bus are available on fixed locations in the connector. The rest of the pins are used for general purpose I/O. This allows the definition of expansion boards that can plug into a number of different Silicon Labs starter kits.

The figure below shows the pin assignment of the expansion header for the EFM32 Pearl Gecko Starter Kit. Because of limitations in the number of available GPIO pins, some of the expansion header pins are shared with kit features.
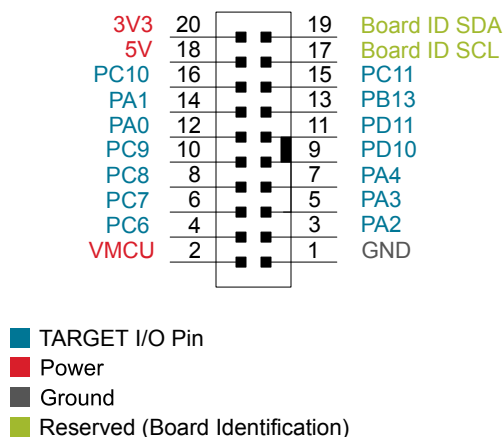

Figure 4.2. Expansion Header

Table 4.3. EXP Header Pinout

Pin	Connection	EXP Header function	Primary function	Shared feature
20	3V3	Board controller supply		
18	5V	Board USB voltage		
16	PC10	I2C_SDA	I2C0_SDA #15	SENSOR_I2C_SDA
14	PA1	UART_RX	USART0_RX #0	VCOM_RX
12	PA0	UART_TX	USART0_TX #0	VCOM_TX
10	PC9	SPI_CS	USART1_CS #11	
8	PC8	SPI_SCLK	USART1_CLK #11	DISP_SCLK
6	PC7	SPI_MISO	USART1_RX #11	
4	PC6	SPI_MOSI	USART1_TX #11	DISP_MOSI
2	VMCU	EFM32 voltage domain, included in AEM measurements.		
19	BOARD_ID_SDA	Connected to Board Controller for identification of add-on boards.		
17	BOARD_ID_SCL	Connected to Board Controller for identification of add-on boards.		
15	PC11	I2C_SCL	I2C0_SCL #15	SENSOR_I2C_SCL
13	PB13	GPIO		
11	PD11	GPIO		

Pin	Connection	EXP Header function	Primary function	Shared feature
9	PD10	GPIO		
7	PA4	GPIO		
5	PA3	GPIO		VCOM_RTS
3	PA2	GPIO		VCOM_CTS
1	GND	Ground		

4.3 Debug Connector (DBG)

The Debug Connector serves a dual purpose. Based on the "debug mode", which can be set up using Simplicity Studio. In the "Debug IN" mode this connector allows an external debug emulator to be used with the on-board EFM32. In the "Debug OUT" mode this connector allows the kit to be used as a debugger towards an external target. In the "Debug MCU" (default) mode this connector is isolated from the debug interface of both the Board Controller and the on-board target device.

Because this connector is automatically switched to support the different operating modes, it is only available when the Board Controller is powered (J-Link USB cable connected). If debug access to the target device is required when the Board Controller is unpowered, this should be done by connecting directly to the appropriate breakout pins.

The pinout of the connector follows that of the standard ARM Cortex Debug+ETM 19-pin connector. The pinout is described in detail below.
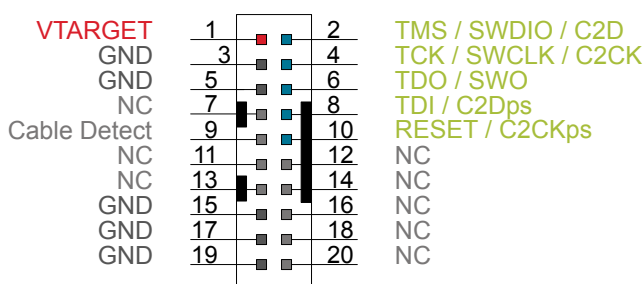

Figure 4.3. Debug Connector

Note that the pin-out matches the pin-out of an ARM Cortex Debug+ETM connector, but these are not fully compatible as pin 7 is physically removed from the Cortex Debug+ETM connector. Some cables have a small plug that prevent them from being used when this pin is present. If this is the case, remove the plug, or use a standard 2x10 1.27 mm straight cable instead.

Table 4.4. Debug Connector Pin Descriptions

Pin number(s)	Function	Note
1	VTARGET	Target reference voltage. Used for shifting logical signal levels between target and debugger.
2	TMS / SDWIO / C2D	JTAG test mode select, Serial Wire data or C2 data
4	TCK / SWCLK / C2CK	JTAG test clock, Serial Wire clock or C2 clock
6	TDO/SWO	JTAG test data out or Serial Wire Output
8	TDI / C2Dps	JTAG test data in, or C2D "pin sharing" function
10	RESET / C2CKps	Target device reset, or C2CK "pin sharing" function
9	Cable detect	Connect to ground
11, 12, 13, 14, 16, 18, 20	NC	Not connected
3, 5, 15, 17, 19	GND	

4.4 Simplicity Connector

The Simplicity Connector featured on the Starter Kit enables advanced debugging features such as the AEM and the Virtual COM port to be used towards an external target. The pinout is illustrated in the figure below.
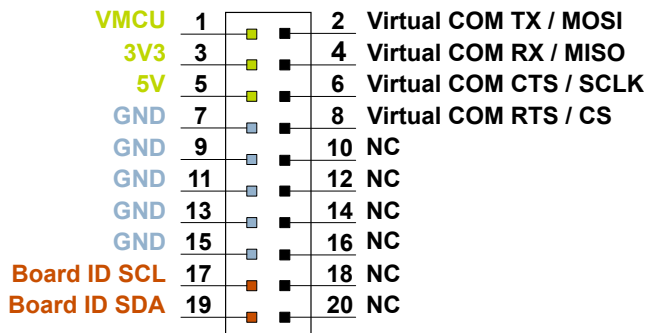

Figure 4.4. Simplicity Connector

Note: Current drawn from the VMCU voltage pin is included in the AEM measurements, while the 3V3 and 5V voltage pins are not. To monitor the current consumption of an external target with the AEM, put the on-board MCU in its lowest energy mode to minimize its impact on the measurements.

Table 4.5. Simplicity Connector Pin Descriptions

Pin number(s)	Function	Description
1	VMCU	3.3 V power rail, monitored by the AEM
3	3V3	3.3 V power rail
5	5V	5 V power rail
2	VCOM_TX_MOSI	Virtual COM Tx/MOSI
4	VCOM_RX_MISO	Virtual COM Rx/MISO
6	VCOM_CTS_SCLK	Virtual COM CTS/SCLK
8	VCOM_RTS_CS	Virtual COM RTS/CS
17	EXT_ID_SCL	Board ID SCL
19	EXT_ID_SDA	Board ID SDA
10, 12, 14, 16, 18, 20	NC	Not connected
7, 9, 11, 13, 15	GND	Ground

5. Power Supply and Reset

5.1 MCU Power Selection

The EFM32 on the Starter Kit can be powered by one of these sources:

• The debug USB cable; or
• a 3 V coin cell battery.

The power source for the MCU is selected with the slide switch in the lower left corner of the Starter Kit. shows how the different power sources can be selected with the slide switch.
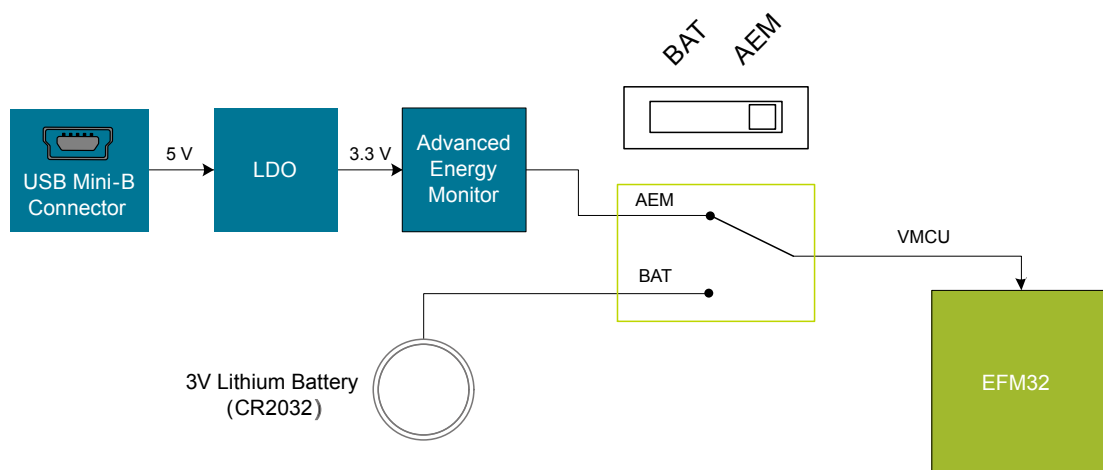

Figure 5.1. Power Switch

With the switch in the **AEM** position, a low noise 3.3 V LDO on the Starter Kit is used to power the EFM32. This LDO is again powered from the debug USB cable. The Advanced Energy Monitor is now connected in series, allowing accurate high speed current measurements and energy debugging/profiling.

With the switch in the **BAT** position, a 20 mm coin cell battery in the CR2032 socket can be used to power the device. With the switch in this position no current measurements are active. This is the recommended switch position when powering the MCU with an external power source.

Note: The Advanced Energy Monitor can only measure the current consumption of the EFM32 when the power selection switch is in the **AEM** position.

5.2 Board Controller Power

The board controller is responsible for important features such as the debugger and the Advanced Energy Monitor, and is powered exclusively through the USB port in the top left corner of the board. This part of the kit resides on a separate power domain, so a different power source can be selected for the target device while retaining debugging functionality. This power domain is also isolated to prevent current leakage from the target power domain when power to the Board Controller is removed.

The board controller power domain is exclusively supplied by the J-Link USB cable, and is not influenced by the position of the power switch.

The kit has been carefully designed to keep the board controller and the target power domains isolated from each other as one of them powers down. This ensures that the target EFM32 device will continue to operate in the **USB** and **BAT** modes.

5.3 EFM32 Reset

The EFM32 MCU can be reset by a few different sources:

- A user pressing the RESET button.
- The on-board debugger pulling the #RESET pin low.
- An external debugger pulling the #RESET pin low.

In addition to the reset sources mentioned above, the Board Controller will also issue a reset to the EFM32 when booting up. This means that removing power to the Board Controller (plugging out the J-Link USB cable) will not generate a reset, but plugging the cable back in will, as the Board Controller boots up.

6. Peripherals

The starter kit has a set of peripherals that showcase some of the features of the EFM32.

Be aware that most EFM32 I/O routed to peripherals are also routed to the breakout pads. This must be taken into consideration when using the breakout pads for your application.

6.1 Push Buttons and LEDs

The kit has two user push buttons marked PB0 and PB1. They are connected directly to the EFM32, and are debounced by RC filters with a time constant of 1 ms. The buttons are connected to pins PF6 and PF7.

The kit also features two yellow LEDs marked LED0 and LED1, that are controlled by GPIO pins on the EFM32. The LEDs are connected to pins PF4 and PF5 in an active-high configuration.
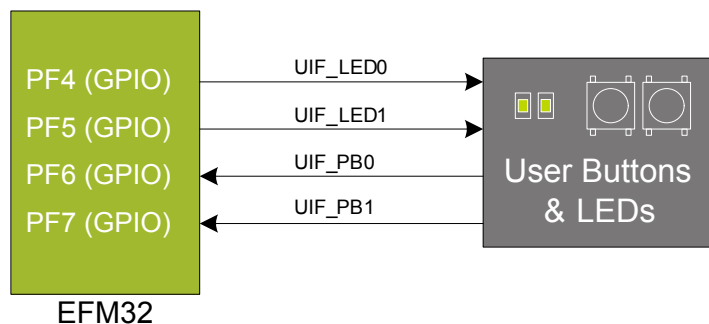
Figure 6.1. Buttons and LEDs

6.2 Memory LCD-TFT Display

A 1.28-inch SHARP Memory LCD-TFT has been added to the board to enable interactive applications to be developed. The display has a high resolution of 128 by 128 pixels, and consumes very little power. It is a reflective monochrome display, so each pixel can only be light or dark, and no backlight is needed in normal daylight conditions. Data sent to the display is stored in the pixels on the glass, which means no continous refreshing is required to maintain a static image.

The display interface consists of an SPI-compatible serial interface and some extra control signals. Pixels are not individually addressable, instead data is sent to the display one line (128 bits) at a time.

The Memory LCD-TFT display is shared with the kit Board Controller, allowing the Board Controller application to display useful information when the user application is not using the display. The user application always controls ownership of the display with the EFM_DISP_ENABLE line:
 • 0: The Board Controller has control of the display
 • 1: The user application (EFM32) has control of the display

Power to the display is sourced from the target application power domain when the EFM32 controls the display, and from the Board Controller's power domain when the EFM_DISP_ENABLE line is low. Data is clocked in on EFM_DISP_MOSI when EFM_DISP_CS is high, and the clock is sent on EFM_DISP_SCLK. The maximum supported clock speed is 1.1 MHz.

EFM_DISP_COM is the "COM Inversion" line. It must be pulsed periodically to prevent static build-up in the display itself. Please refer to the display application information for details on driving the display:

http://www.sharpmemorylcd.com/1-28-inch-memory-lcd.html
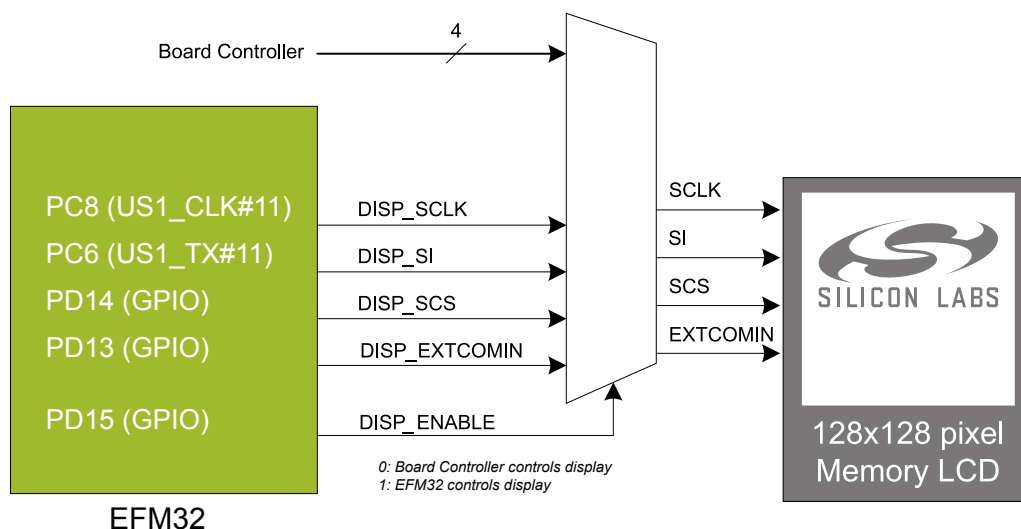
Figure 6.2. 128x128 pixel Memory LCD

6.3 Capacitive Touch Pads

Two touch pads, T1 and T2, utilizing the capacitive touch capability of the EFM32 are available. They are located on the lower right side of the board, beneath the EFM32. The pads are connected to PB11 and PB12.
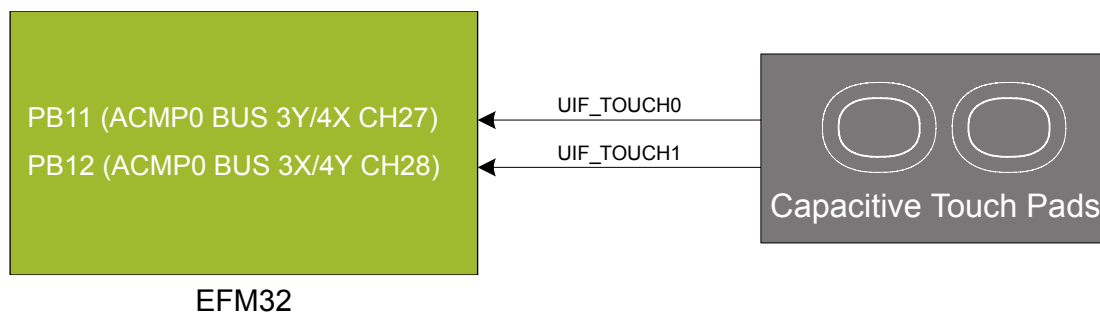

Figure 6.3. Touch Pads

The capacitive touch pads work by sensing changes in the capacitance of the pads when touched by a human finger. Sensing the changes in capacitance is done by setting up the touch pad as part of an RC relaxation oscillator using the EFM32's analog comparator, and then counting the number of oscillations during a fixed period of time.

6.4 Si7021 Relative Humidity and Temperature Sensor

The Si7021 I^2C relative humidity and temperature sensor is a monolithic CMOS IC integrating humidity and temperature sensor elements, an analog-to-digital converter, signal processing, calibration data, and an I^2C Interface. The patented use of industry-standard, low-K polymeric dielectrics for sensing humidity enables the construction of low-power, monolithic CMOS Sensor ICs with low drift and hysteresis, and excellent long term stability.

The humidity and temperature sensors are factory-calibrated and the calibration data is stored in the on-chip non-volatile memory. This ensures that the sensors are fully interchangeable, with no recalibration or software changes required.

The Si7021 is available in a 3x3 mm DFN package and is reflow solderable. It can be used as a hardware- and software-compatible drop-in upgrade for existing RH/ temperature sensors in 3x3 mm DFN-6 packages, featuring precision sensing over a wider range and lower power consumption. The optional factory-installed cover offers a low profile, convenient means of protecting the sensor during assembly (e.g., reflow soldering) and throughout the life of the product, excluding liquids (hydrophobic/oleophobic) and particulates.

The Si7021 offers an accurate, low-power, factory-calibrated digital solution ideal for measuring humidity, dew-point, and temperature, in applications ranging from HVAC/R and asset tracking to industrial and consumer platforms.

The I^2C bus used for the Si7021 is shared with the Expansion Header. The temperature sensor is normally isolated from the I^2C line. To use the sensor, PD9 must be set high. When enabled, the sensor's current consumption is included in the AEM measurements.
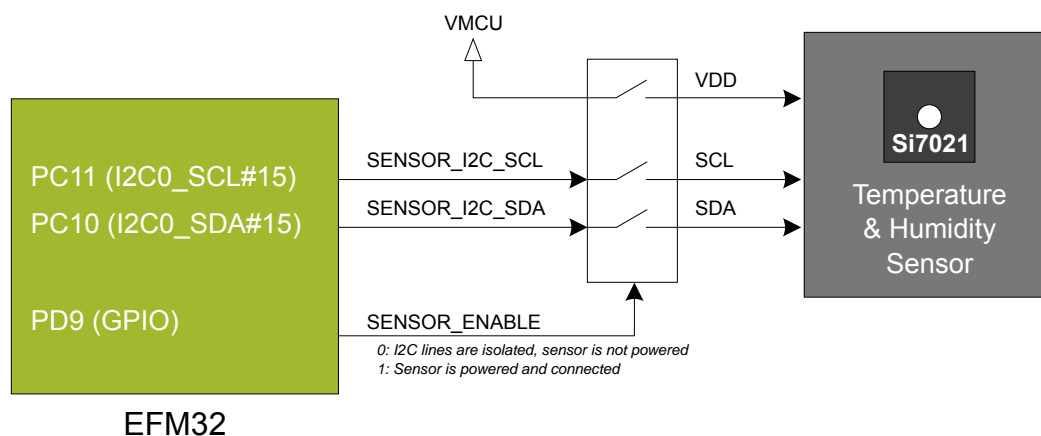

Figure 6.4. Si7021 Relative Humidity and Temperature Sensor

Please refer to the Silicon Labs web pages for more information: http://www.silabs.com/humidity-sensors

6.5 USB to UART Bridge

The board controller on the kit provides a virtual com port (CDC) interface when plugged into a computer. The target EFM32 can connect to this serial port interface and communicate directly with the host computer.

The interface is enabled by setting the EFM_BC_EN (PA5) line high, and the UART peripheral used is USART0, Location 0 (TX pin PA0, RX pin PA1). The default baud rate is 115200.
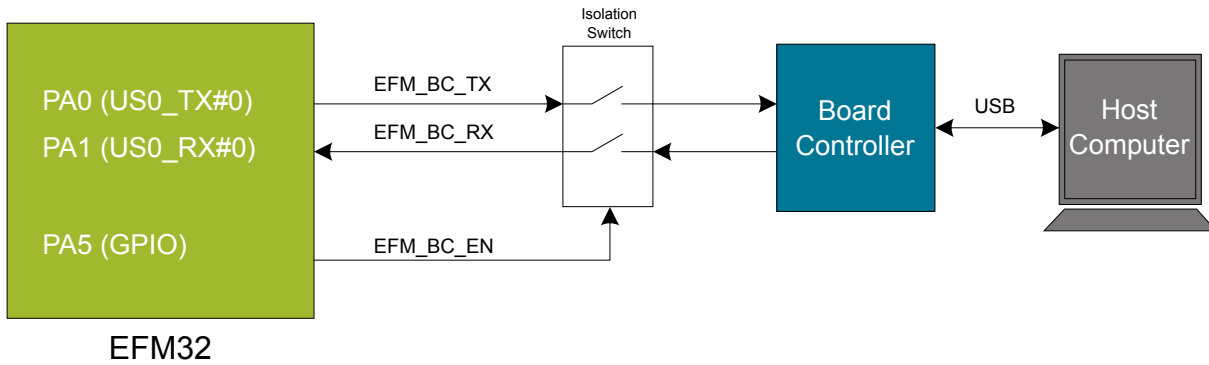

Figure 6.5. Virtual COM Port Interface

7. Advanced Energy Monitor

7.1 Usage

The AEM (Advanced Energy Monitor) data is collected by the board controller and can be displayed by the Energy Profiler, available through Simplicity Studio. By using the Energy Profiler, current consumption and voltage can be measured and linked to the actual code running on the EFM32 in realtime.

7.2 Theory of Operation

In order to be able to accurately measure current ranging from 0.1 μA to 50 mA (114 dB dynamic range), a current sense amplifier is utilized together with a dual gain stage. The current sense amplifier measures the voltage drop over a small series resistor, and the gain stage further amplifies this voltage with two different gain settings to obtain two current ranges. The transition between these two ranges occurs around 250 μA. Digital filtering and averaging is done within the Board Controller before the samples are exported to the Energy Profiler application.

During startup of the kit, an automatic calibration of the AEM is performed. This calibration compensates for the offset error in the sense amplifiers.
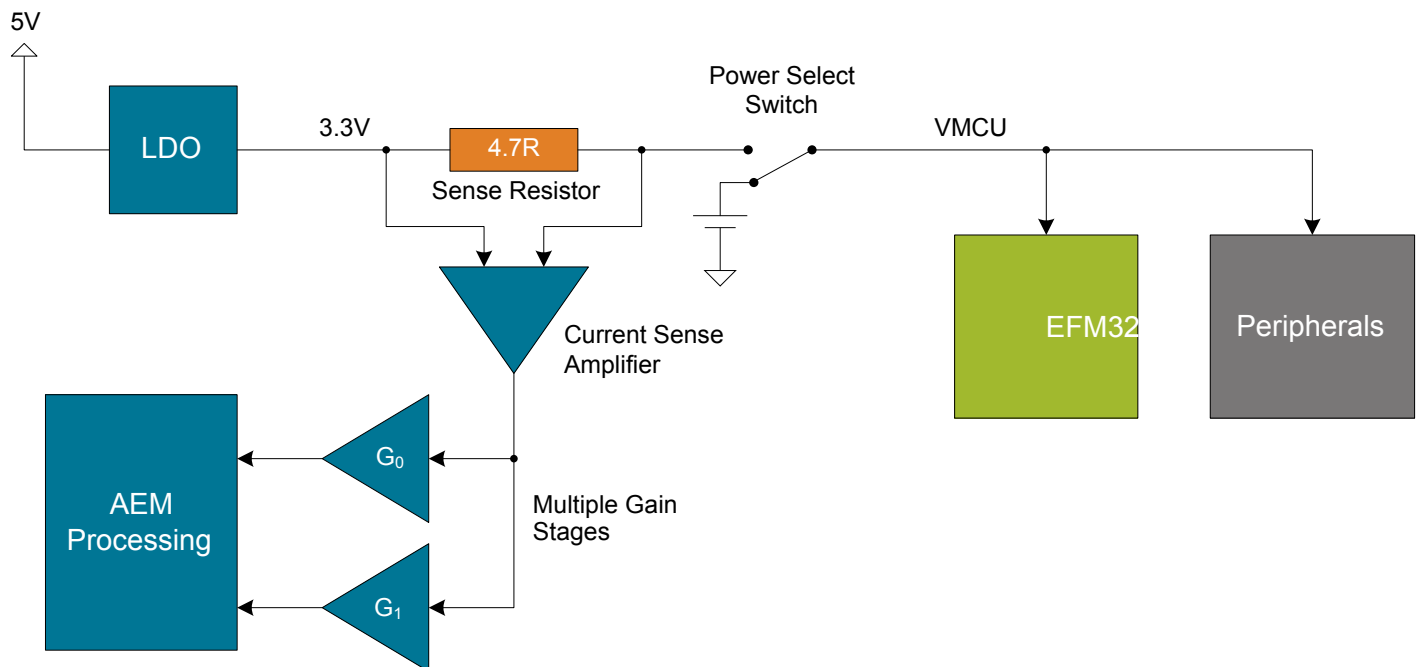
Figure 7.1. Advanced Energy Monitor

7.3 Accuracy and Performance

The Advanced Energy Monitor is capable of measuring currents in the range of 0.1 μA to 50 mA. For currents above 250 μA, the AEM is accurate within 0.1 mA. When measuring currents below 250 μA, the accuracy increases to 1 μA. Even though the absolute accuracy is 1 μA in the sub 250 μA range, the AEM is able to detect changes in the current consumption as small as 100 nA. The AEM produces 6250 current samples per second.

8. On-Board Debugger

The SLSTK3401A contains an integrated debugger, which can be used to download code and debug the EFM32. In addition to programming the EFM32 on the kit, the debugger can also be used to program and debug external Silicon Labs EFM32, EFM8, EZR32 and EFR32 devices.

The debugger supports three different debug interfaces used with Silicon Labs devices:

- Serial Wire Debug, is used with all EFM32, EFR32 and EZR32 devices
- JTAG, which can be used with some newer EFR32 and EFM32 devices
- C2 Debug, which is used with EFM8 devices

In order for debugging to work properly, make sure you have the approriate debug interface selected that works with your device. The debug connector on the board supports all three of these modes.

8.1 Debug Modes

Programming external devices is done by connecting to a target board through the provided Debug IN/OUT Connector, and by setting the debug mode to [**Out**]. The same connector can also be used to connect an external debugger to the EFM32 MCU on the kit, by setting the debug mode to [**In**].

Selecting the active debug mode is done with a drop-down menu in the Kit Manager tool in Simplicity Studio.

Debug MCU: In this mode the on-board debugger is connected to the EFM32 on the SLSTK3401A.
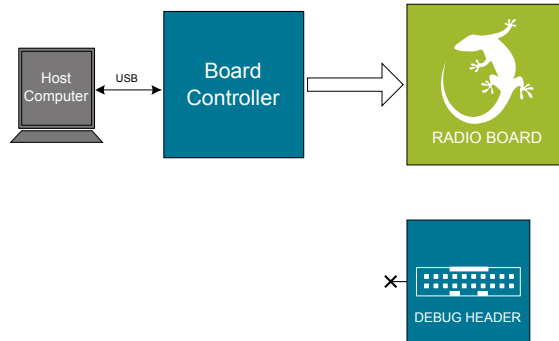
Figure 8.1. Debug MCU

Debug OUT: In this mode, the on-board debugger can be used to debug a supported Silicon Labs device mounted on a custom board.
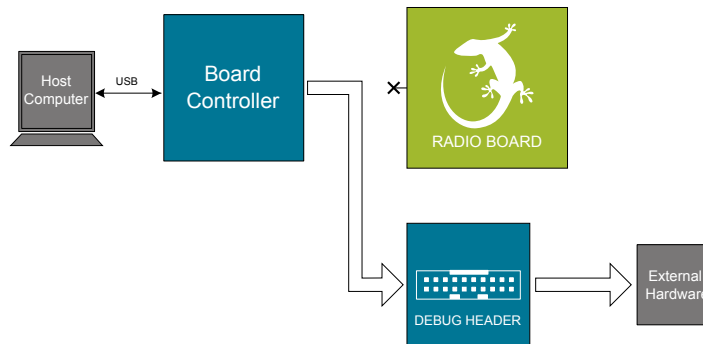
Figure 8.2. Debug OUT

Debug IN: In this mode, the on-board debugger is disconnected, and an external debugger can be connected to debug the EFM32 on the SLSTK3401A.
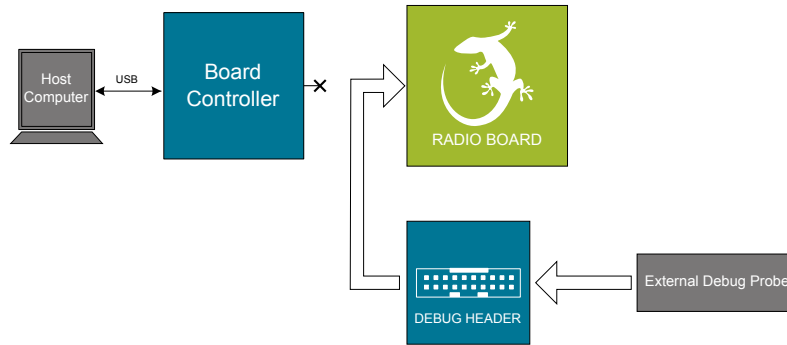
Figure 8.3. Debug IN

Note: For "Debug IN" to work, the board controller on the kit must be powered throught the USB connector.

8.2 Debugging During Battery Operation

When the EFM32 is powered by battery and the J-Link USB is still connected, the on-board debug functionality is available. If the USB power is disconnected, the Debug In mode will stop working.

If debug access is required when the target is running of another energy source, such as a battery, and the board controller is powered down, the user should make direct connections to the GPIO used for debugging. This can be done by connecting to the appropriate pins of the breakout pads. Some Silicon Labs kits provide a dedicated pin header for this purpose.

9. Board Support Package

The Board Support Package (BSP) is a set of C source and header files that enables easy access to, and control over some board-specific features. Using the BSP is not required, and the EFM32 is fully usable without the BSP.

The BSP uses EFM32 peripheral USART0, Location 0 (TX pin PA0, RX pin PA1) on baudrate 115200-8-N-1 to communicate with the board controller.

Note:

The BSP functionality is only available when the Starter Kit is USB-powered. Using these function calls with USB disconnected will give unpredictable results.

9.1 Application Programming Interface

To use the BSP, include the Starter Kit header file, like this:

```
#include "bsp.h"
```

All functions in the BSP are prefixed with BSP_. Some functions to control the user LEDs might look like:

```
int  BSP_LedClear (int ledNo)
```

Turn off a single LED.

```
int  BSP_LedSet (int ledNo)
```

Turn on a single LED.

9.2 Example Applications

Under the kits/SLSTK3401A/examples folder in your base Simplicity installation directory, you will find example programs using the BSP, with corresponding project/Makefiles for supported IDEs.

The *examples* folder also contains examples showing how to use the different peripherals on the SLSTK3401A.

9.3 How To Include In Your Own Applications

The easiest way to include the BSP in your application is to create it using the Simplicity IDE.

Alternatively, base your work on an example application that uses the BSP. The following items are recommended for correct configuration:

1. Make sure you define the correct part number (i.e., EFM32PG1B200F256GM48) as a preprocessor defined symbol
2. Make sure you define the correct part number (i.e., EFM32PG1B200F256GM48) for your project file
3. Add and include the EFM32 CMSIS-files (startup_efm32.s, system_efm32.c, core_cm3.c) to your project.
4. Add and include *all* BSP package .c-files, with the bsp-prefix to your project
5. Configure include paths to point at the CMSIS/CM3/CoreSupport and CMSIS/CM3/DeviceSupport/EnergyMicro/EFM32 directories
6. Configure include paths to point to the kits/SLSTK3401A/bsp directory

Make sure you call "BSP_Init()" early at startup, and you should be all set.

10. Integrated Development Environments

Figure 10.1. Simplicity Studio

Simplicity Studio includes various examples in source form to use with the Starter Kit. To run these examples:

1. Provide power to the board by connecting the DBG USB connector to the PC using the provided USB cable.
2. Move the switch to the AEM position.
3. Click the [**Refresh detected hardware**] button and select the SLSTK3401A kit under [**Detected Hardware**].
4. Click the [**Software Examples**] tile under [**Software and Kits**].
5. In the wizard, select the SLSTK3401A kit and click [**Next**].
6. Select the desired example or demo from the list and click [**Next**].
7. Click [**Finish**].
8. Click the [**Debug**] button in the IDE to build and download the code to the hardware.
9. Follow the instructions at the top of the main example file to set up the hardware as needed.
10. Click the [**Resume**] button to start running the example.

10.1 Third Party IDEs

The software examples in Simplicity Studio also include project files for several third party IDEs:

- Atollic TrueSTUDIO® for ARM.
- IAR Embedded Workbench® for ARM
- Keil® MDK (µVision®)
- Rowley CrossWorks™ for ARM
- Makefile for GNU ARM Embedded Toolchain

These project files are located in separate subfolders in each example folder.

11. Kit Manager and Upgrades

The Kit Manager is a program that comes with Simplicity Studio. It can perform various kit- and EFM32-specific tasks.

11.1 Kit Manager Operation

This utility gives the ability to program the EFM32, upgrade the kit, lock and unlock devices and more. Some of the features will only work with Silicon Labs kits, while other will work with a generic J-Link debugger connected.

Kit Manager

Adapter Information

Type	USB
Serial Number	440019219
Vendor	Segger

Kit Information

Name	EFM32 Giant Gecko Starter Kit
Board ID	BRD2200A Rev. A03
Serial Number	140502223
Firmware Version	0v9p13b391

Update Kit

Installation Package

[   ] [Browse...] [Install Package]

Debug Mode: MCU ▼

Addressing Mode: Serial number ▼

MCU Information

Name	EFM32GG990F1024
Architecture	ARM
Locked	false
Revision	1D
Production ID	20
Flash Size	1024 kB
SRAM Size	128 kB
Unique ID	0x248ab60151c79626

[Close]

Figure 11.1. Kit Manager

11.2 Firmware Upgrades

Upgrading the kit firmware is done through Simplicity Studio. Simplicity Studio will automatically check for new updates on startup.

You can also use the Kit Manager for manual upgrades. Click the [**Browse**] button in the [**Update Kit**] section to select the correct file ending in ".emz". Then, click the [**Install Package**] button.

12. Schematics, Assembly Drawings and BOM

The schematics, assembly drawings and bill of materials (BOM) for the hardware included in the EFM32 Pearl Gecko Starter Kit are available through Simplicity Studio when the kit documentation package has been installed.

13. Kit Revision History and Errata

13.1 Revision History

The kit revision can be found printed on the box label of the kit, as outlined in the figure below.

EFM32 Pearl Gecko Starter Kit

(1P) Part: SLSTK3401A   (D) Date: 30-09-15

S.nr: 115100140   (Q) Qty: 1   Rev.A00
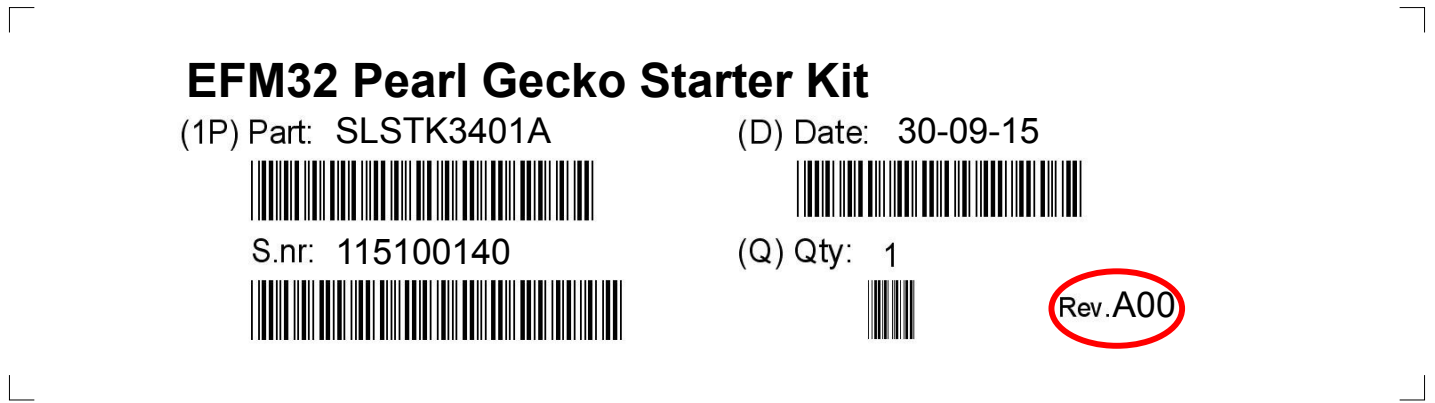
Figure 13.1. Revision Info

Table 13.1. Kit Revision History

Kit Revision	Released	Description
B00	2016-09-28	Coin-cell battery no longer included with kit due to shipping restrictions.
A01	2016-07-04	Updated board revision to A01 - Updated EFM32 Pearl Gecko to revision C0.
A00	2015-09-30	Initial Kit Revision.

13.2 Errata

There are no known errata at present.

14. Document Revision History

1.01

2016-11-21

Corrected references to radio board.

1.00

2015-10-30

Initial document version.

Table of Contents
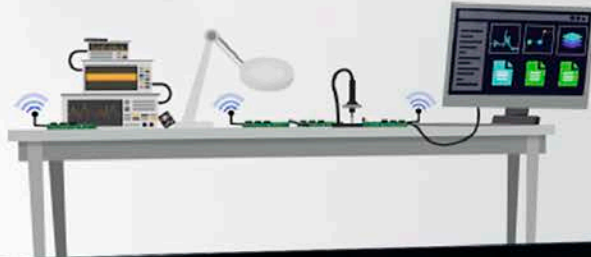
Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com