

# UG255: EFM32 Happy Gecko Starter Kit

## User's Guide



The SLSTK3400A Starter Kit is an excellent starting point to get familiar with the EFM32™ Happy Gecko Microcontroller.

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



### TARGET DEVICE

- EFM32 Happy Gecko Microcontroller (EFM32HG322F64G-A-QFP48)
- CPU: 32-bit ARM® Cortex-M0+®
- Memory: 64 kB flash and 8 kB RAM

### KIT FEATURES

- USB connectivity
- Advanced Energy Monitor
- SEGGER J-Link on-board debugger
- Debug Multiplexer supporting external hardware as well as on-board board MCU
- Silicon Labs' Si7021 Relative Humidity and Temperature sensor
- Ultra low power 128x128 pixel Memory LCD
- User LEDs / Pushbuttons
- 20-pin 2.54 mm header for expansion boards
- Breakout pads for direct access to I/O pins
- Power sources includes USB and CR2032 coin cell battery.

### SOFTWARE SUPPORT

- Simplicity Studio™
- Energy Profiler

## 1. Introduction

### 1.1 Description

The SLSTK3400A is an excellent starting point to get familiar with the EFM32 Happy 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 addition 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

- EFM32HG322F64G-A-QFP48 MCU with 64 kB Flash and 8 kB RAM.
- Advanced Energy Monitoring system for precise current tracking.
- Integrated Segger J-Link USB debugger/emulator with the possibility 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.
- Ultra low power 128x128 pixel Memory-LCD
- 2 user buttons, 2 user LEDs
- USB Micro-B connector.
- Secondary AEM channel to accurately measure low power USB current.
- Crystals for LFXO and HFXO: 32.768 kHz and 24.000 MHz.

### 1.3 Getting Started

#### Hardware

To set up the hardware for the SLSTK3400A 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 SLSTK3400A 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 Happy Gecko Starter Kit. The board comes pre-loaded with a default application, Space Invaders, 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 Happy 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 Happy Gecko are also available under the same tile in Simplicity Studio.

## 2. Kit Block Diagram

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

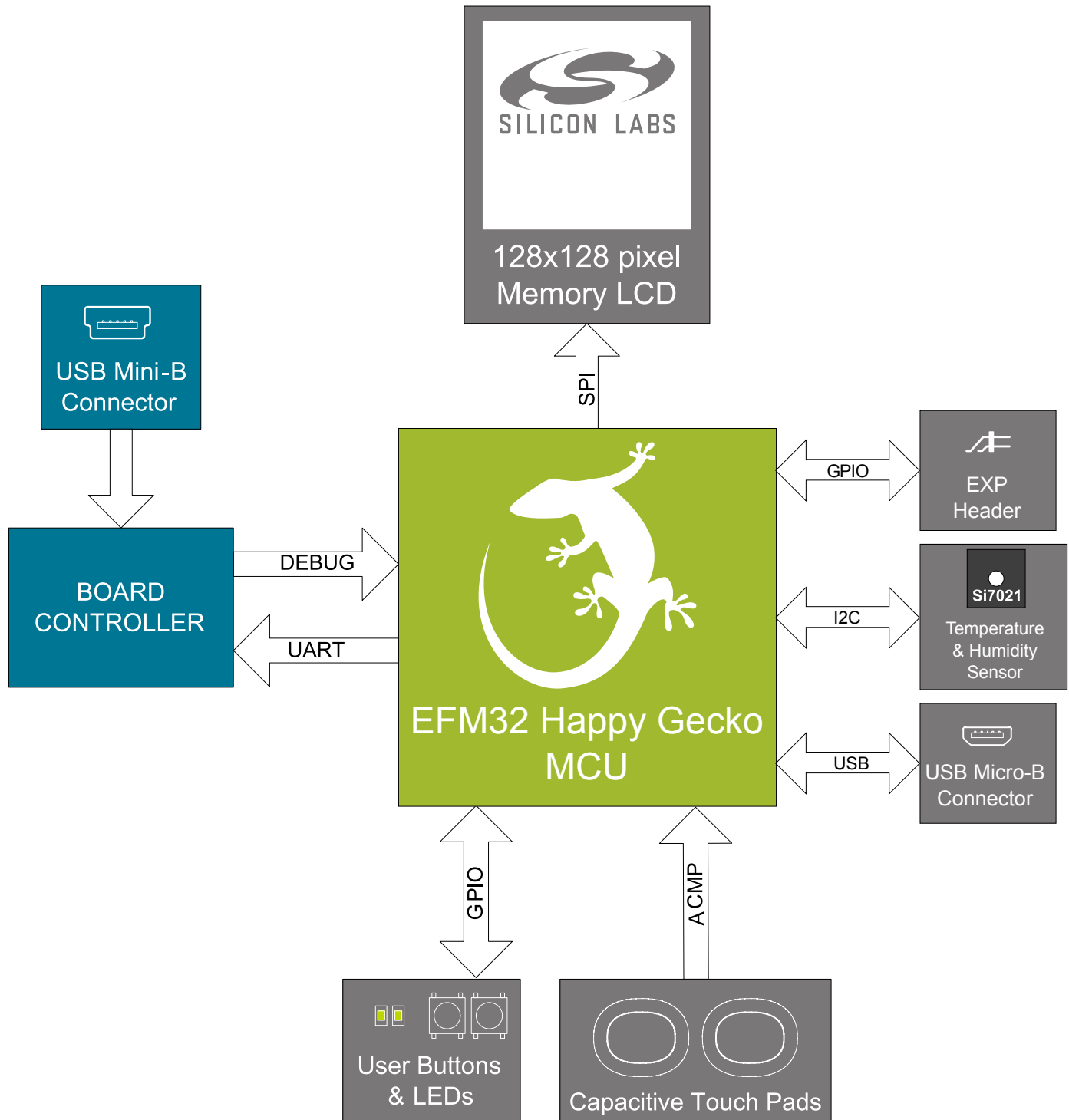


Figure 2.1. Kit Block Diagram

### 3. Kit Hardware Layout

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

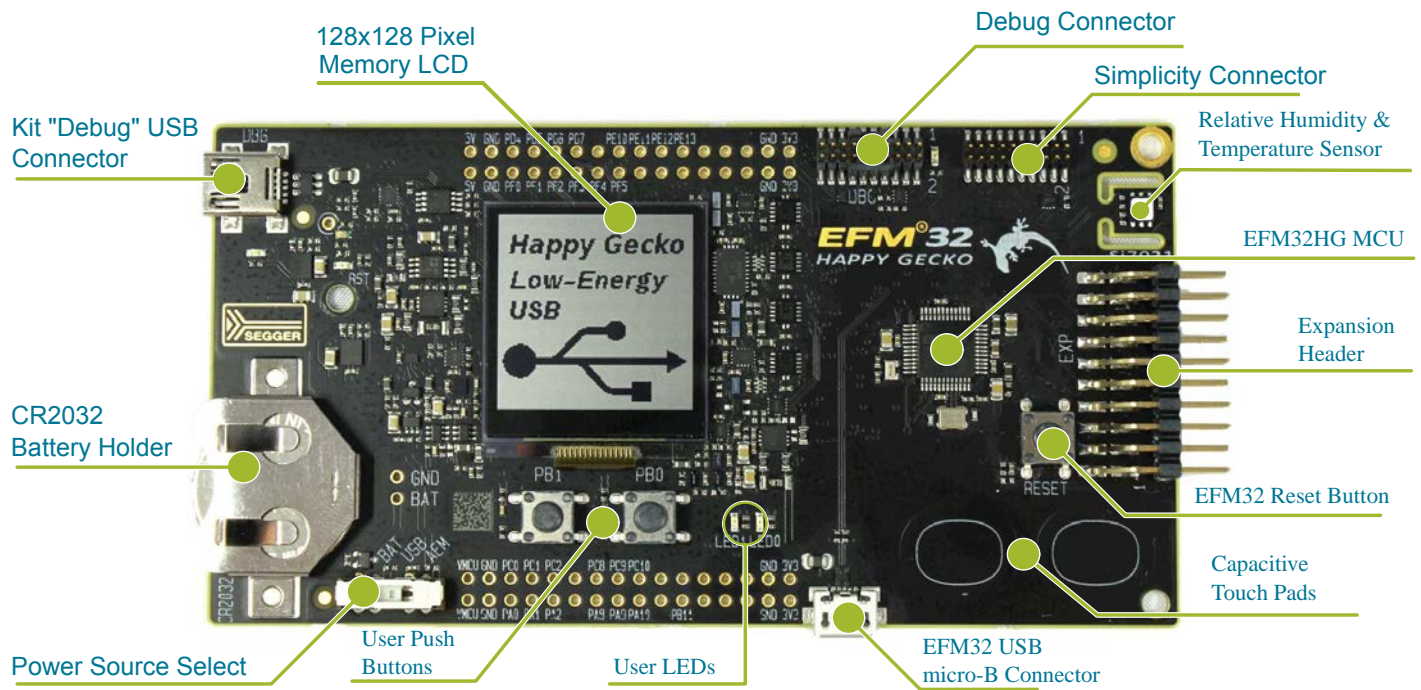


Figure 3.1. SLSTK3400A Hardware Layout

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.

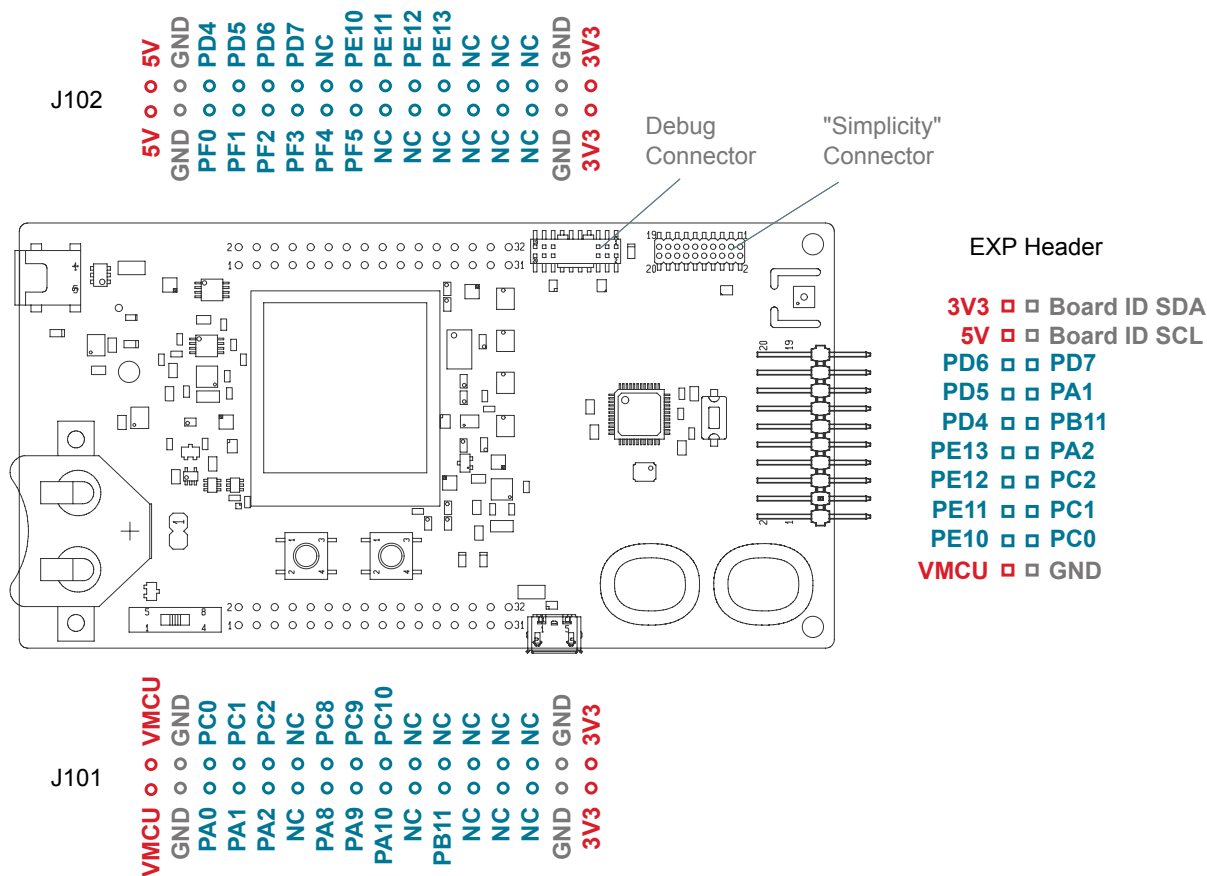


Figure 4.1. Breakout Pads and Expansion Header

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.

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 Happy 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 Happy Gecko Starter Kit. Because of limitations in the number of available GPIO pins, some of the expansion header pins are shared with kit features.

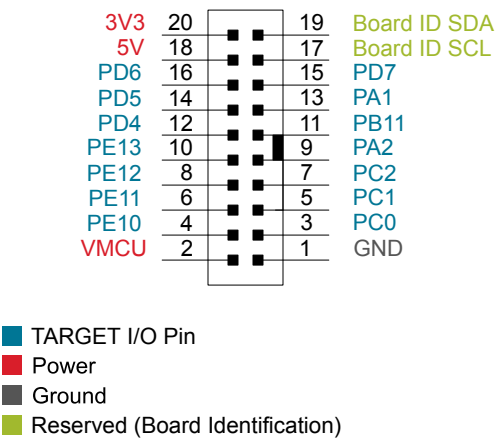


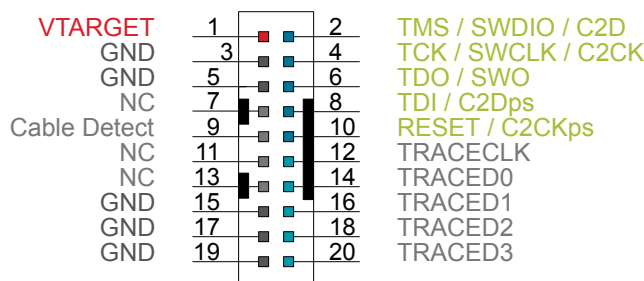
Figure 4.2. Expansion Header

### 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. Note that even though the connector has support for both JTAG and ETM Trace in addition to Serial Wire Debug, it does not necessarily mean that the kit or the on-board target device supports this.



**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 4.1. 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
12	TRACECLK	ETM Trace Clock
14	TRACED0	ETM Trace Data 0
16	TRACED1	ETM Trace Data 1
18	TRACED2	ETM Trace Data 2
20	TRACED3	ETM Trace Data 3
9	Cable detect	Connect to ground
11, 13	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.

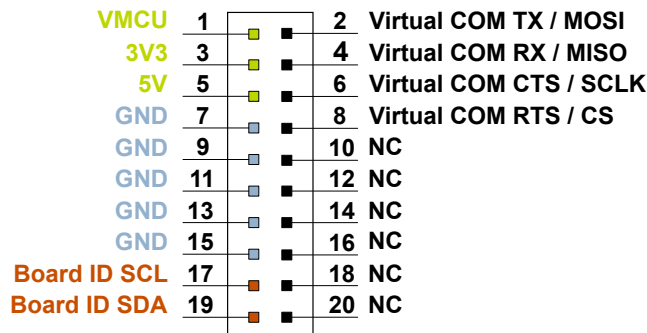


Figure 4.4. Simplicity Connector

**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 in its lowest energy mode to minimize its impact on the measurements.

Table 4.2. 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 Power Selection

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

- The debug USB cable; or
- the EFM32's own USB regulator; or
- a 3 V coin cell battery.

The power source for the is selected with the slide switch in the lower left corner of the Starter Kit. [Figure 5.1 Power Switch on page 8](#) shows how the different power sources can be selected with the slide switch.

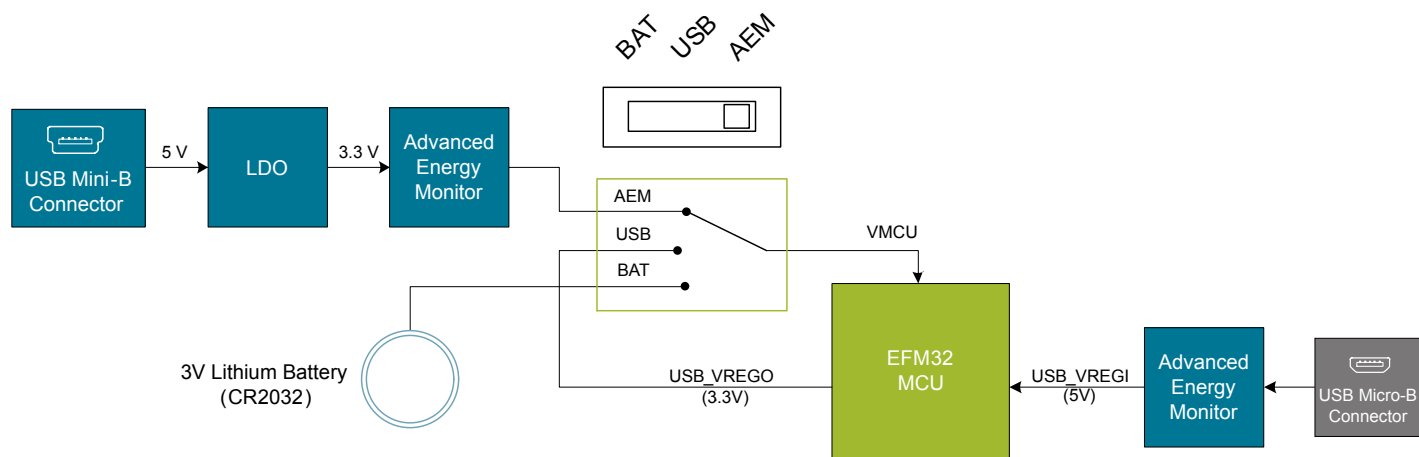


Figure 5.1. Power Switch

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 **USB** position, the integrated linear regulator in the EFM32 Happy Gecko MCU is used to power the rest of the chip as well as the USB PHY with the cable connected to the target USB connector. This allows a USB device application where the MCU acts as a bus powered device. With the switch in this position, the secondary AEM channel is automatically selected to allow accurate measurements of the current consumption of the whole application running on USB power.

Finally, with the switch in the **BAT** position, a 20mm 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 is automatically recalibrated when switching from **AEM** to **USB** or back, which causes the target EFM32 to lose power for a few seconds.

### 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

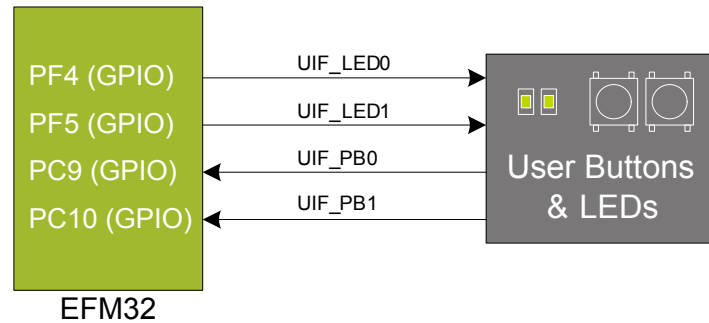
The starter kit has a set of peripherals that showcase some of the features of the EFM32.

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 PC9 and PC10.

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 continuous 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>

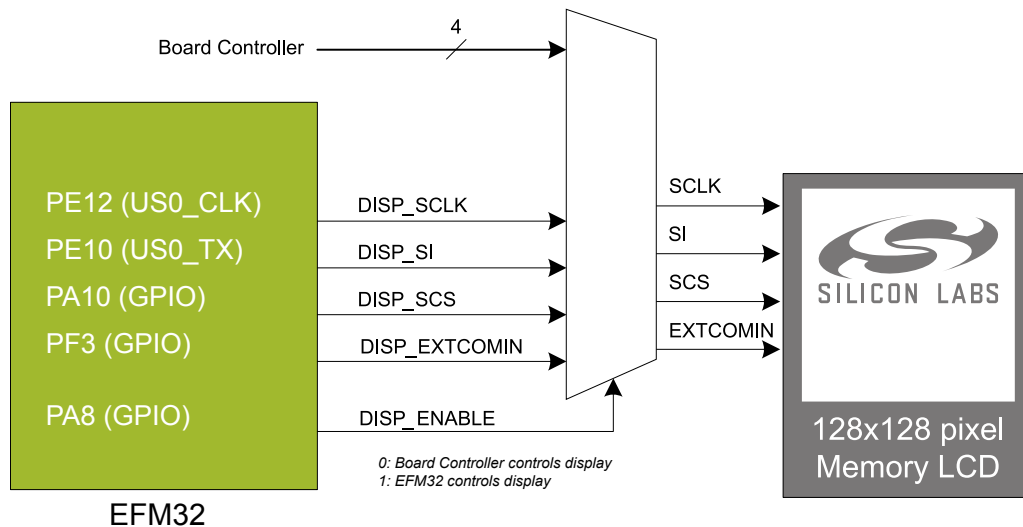
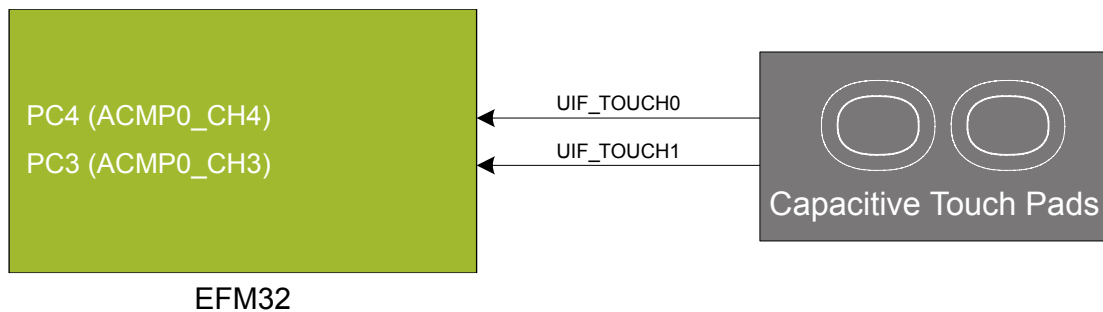


Figure 6.2. 128x128 pixel Memory LCD

### 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 and .



**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

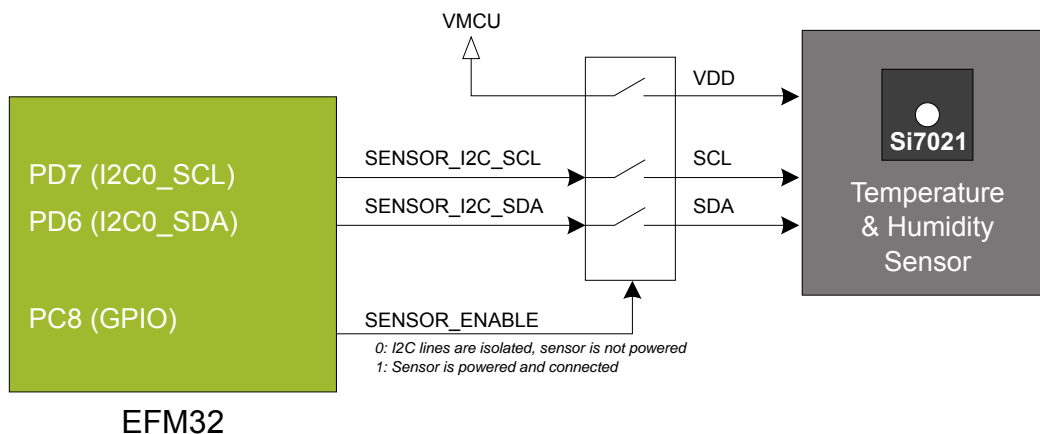
The Si7021 I<sup>2</sup>C relative humidity and temperature sensor is a monolithic CMOS IC integrating humidity and temperature sensor elements, an analog-to-digital converter, signal processing, calibration data, and an I<sup>2</sup>C Interface. The patented use of industry-standard, low-K polymeric 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<sup>2</sup>C bus used for the Si7021 is shared with the Expansion Header. The temperature sensor is normally isolated from the I<sup>2</sup>C line. To use the sensor, PC8 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 Micro-B Connector

The SLSTK3400A board is equipped with a USB Micro-B connector interfacing with the target EFM32 device. This allows the development and evaluation of applications using USB in device mode.

The secondary AEM channel (see chapter on Advanced Energy Monitoring) is placed between the USB connector and the EFM32 device, and allows monitoring of the current sourced from the USB cable. The AEM circuit includes an input filter and a sense resistor on the VBUS line, which could potentially influence USB compliance testing. If it is required, the AEM circuit can be bypassed by removing some components and shorting a footprint on the PCB. More details about this can be found in the schematic.

The EFM32 has an internal LDO regulator that powers the USB PHY inside the chip. 5 V from VBUS is applied to the USB\_VREGI pin, and the output is decoupled on the USB\_VREGO pin. When the power select switch is set to the *USB* position, USB\_VREGO is connected to the VMCU net, which powers the chip and all peripherals in the target voltage domain. When the J-Link USB cable is inserted, it is possible to monitor the current that supplies the USB PHY, the EFM32 device and all user peripherals on the board.

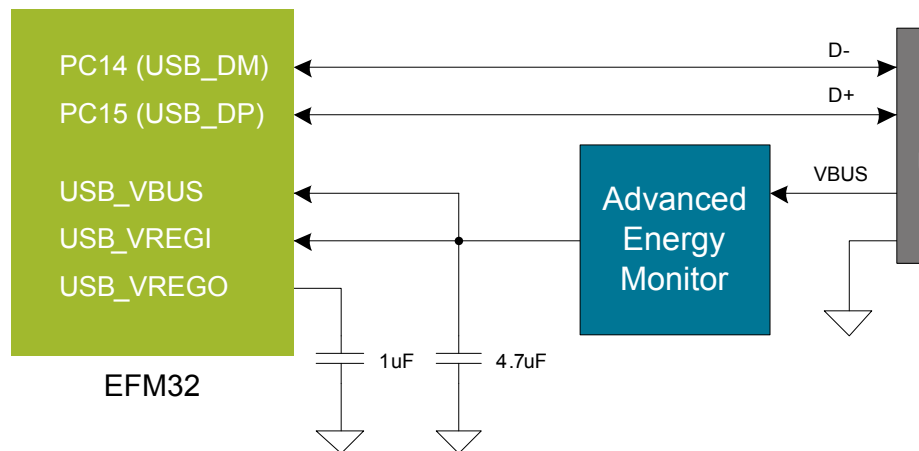


Figure 6.5. EFM32 USB Connector

### 6.5.1 Secondary AEM channel

The secondary AEM channel included in this kit features a filter and a sense resistor on the VBUS line between the connector and the EFM32. This allows the kit to accurately measure the current consumption of the whole bus-powered system when the power switch is in the "USB" position.

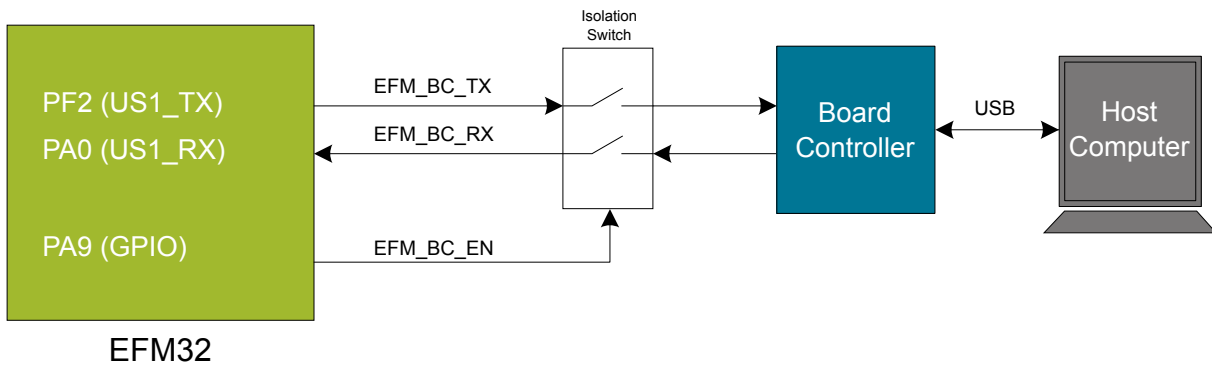
The circuit has been designed such that operation without the J-Link USB cable is also possible without powering up the whole board controller domain. In this case no energy monitoring data is available, but the target device will still operate correctly when the power switch is in the "USB" position.

The filter in the AEM circuit introduces some extra capacitance on the VBUS line, which could cause the inrush test of USB compliance testing to fail. If the kit is to be used for USB compliance testing, the secondary AEM circuit should be by-passed. This can easily be done by removing R746, R736, R748 and shorting R745 on the board. Please refer to the schematic and assembly drawings to locate these components.

## 6.6 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 (PA9) line high, and the UART peripheral used is USART1, Location 4 (TX pin PF2, RX pin PA0). The default baud rate is 115200.



**Figure 6.6. 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  $\mu\text{A}$  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  $\mu\text{A}$ . 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.

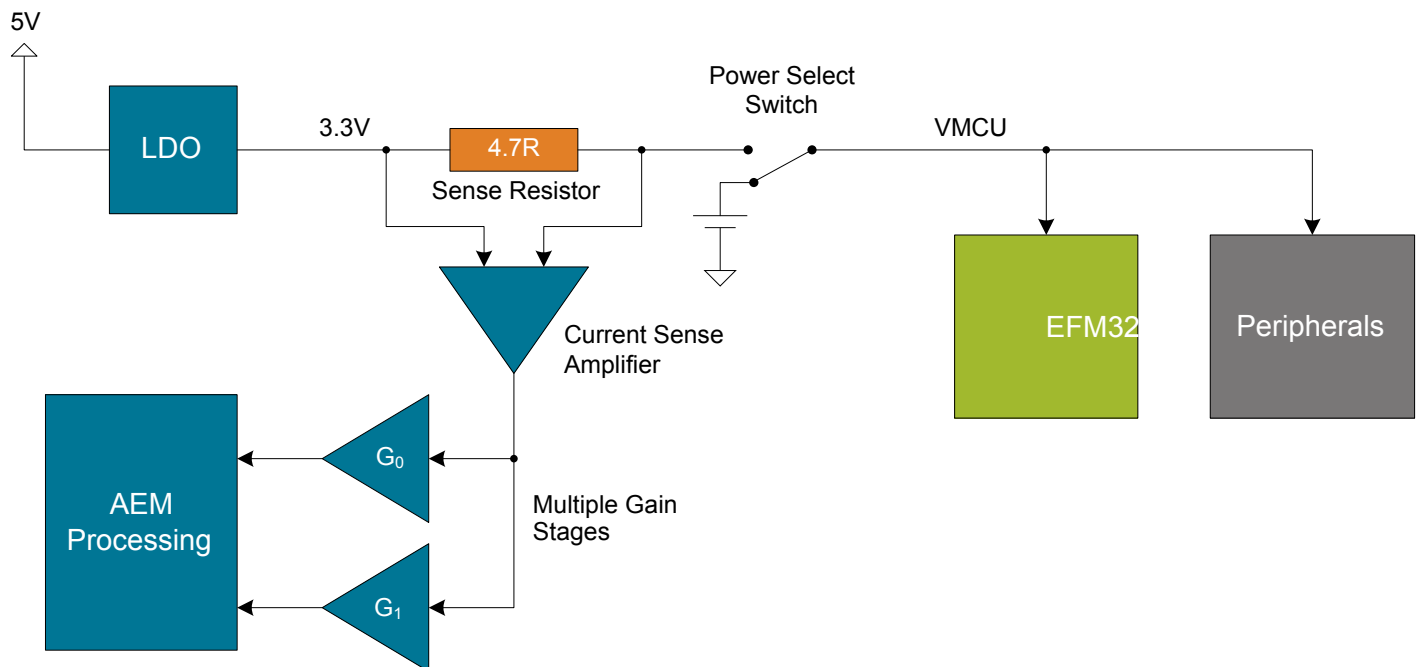


Figure 7.1. Advanced Energy Monitor

### 7.3 Secondary AEM Channel

The EFM32 Happy Gecko Starter Kit contains a secondary AEM channel in addition to the default one. This secondary channel has a sense resistor on the USB VBUS line connected to the EFM32's USB regulator input, and allows energy monitoring of a bus powered USB application, showcasing the EFM32 Happy Gecko's low energy USB peripheral.

In order to use the secondary AEM channel, the power switch must be set to the "USB" position so that the EFM32 is configured as a bus powered device with the internal USB LDO regulator powering the device's VDD rail. The board controller monitors the switch position, and performs an automatic switch-over and calibration when the USB position is selected and a valid voltage is present on the VBUS rail. Both the current and the voltage measurement will now show the VBUS supply instead of the normal 3.3 V supply.

Please note that the recalibration that occurs when switching between the "USB" and "AEM" modes on the power switch causes the target EFM32 to lose power for a few seconds.



## 7.4 Accuracy and Performance

The Advanced Energy Monitor is capable of measuring currents in the range of 0.1  $\mu\text{A}$  to 50 mA. For currents above 250  $\mu\text{A}$ , the AEM is accurate within 0.1 mA. When measuring currents below 250  $\mu\text{A}$ , the accuracy increases to 1  $\mu\text{A}$ . Even though the absolute accuracy is 1  $\mu\text{A}$  in the sub 250  $\mu\text{A}$  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 SLSTK3400A 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 appropriate 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 SLSTK3400A.

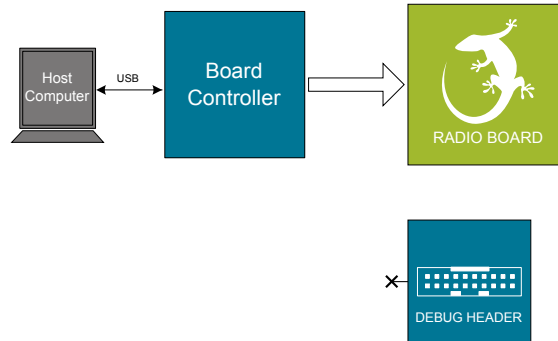


Figure 8.1. Debug MCU

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

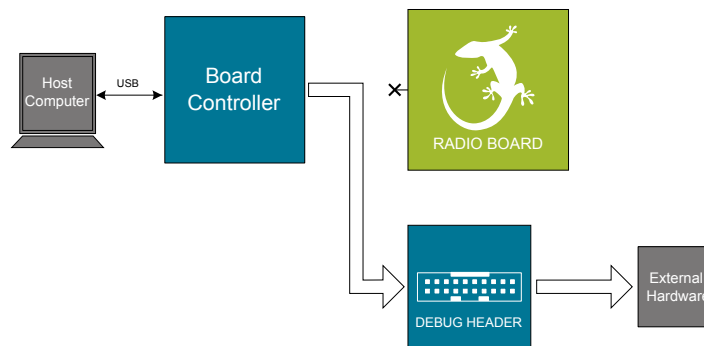
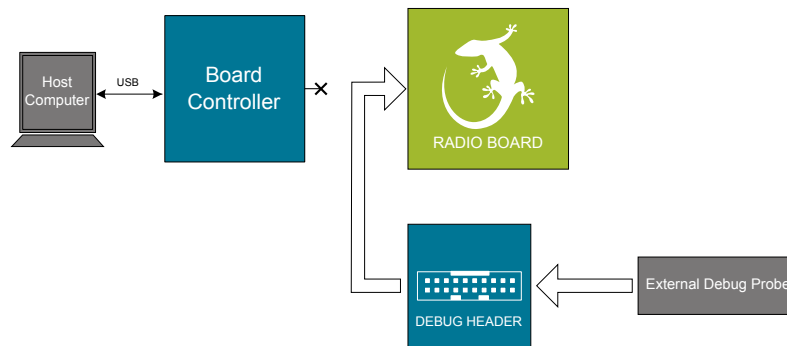


Figure 8.2. Debug OUT

**Debug IN:** In this mode, the on-board debugger is disconnected, and an external debugger can be connected to debug the EFM32 on the SLSTK3400A.



**Figure 8.3. Debug IN**

**Note:** For "Debug IN" to work, the board controller on the kit must be powered through 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 USART1, Location 4 (TX pin PF2, RX pin PA0) 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/SLSTK3400A/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 SLSTK3400A.

### 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., EFM32HG322F64G-A-QFP48) as a preprocessor defined symbol
  2. Make sure you define the correct part number (i.e., EFM32HG322F64G-A-QFP48) 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/SLSTK3400A/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 SLSTK3400A kit under **[Detected Hardware]**.
4. Click the **[Software Examples]** tile under **[Software and Kits]**.
5. In the wizard, select the SLSTK3400A 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.

The screenshot shows the Kit Manager application window with the following sections:

- 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

Debug Mode:

Addressing Mode:
- MCU Information**

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

At the bottom left is a help icon (?) and at the bottom right is a  button.

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 Happy 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.



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.
A04	2015-11-20	Updated board revision to B01 - Added input filter for secondary AEM channel.
A03	2015-04-30	Updated board revision to B01 - Added input filter for secondary AEM channel.
A02	2015-03-20	Updated board revision to B00 - Pinout changes related to internal operation of the Board Controller
A01	2015-02-02	Initial Kit Revision.

### 13.2 Errata

There are no known errata at present.

## 14. Document Revision History

### 1.01

2016-11-21

Updated document title.

### 1.00

2015-04-30

Initial document version.

---

# Table of Contents

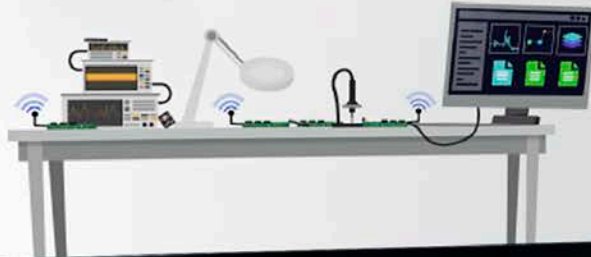
<b>1. Introduction . . . . .</b>	<b>1</b>
1.1 Description . . . . .	1
1.2 Features. . . . .	1
1.3 Getting Started . . . . .	1
<b>2. Kit Block Diagram . . . . .</b>	<b>2</b>
<b>3. Kit Hardware Layout . . . . .</b>	<b>3</b>
<b>4. Connectors . . . . .</b>	<b>4</b>
4.1 Breakout Pads . . . . .	4
4.2 Expansion Header . . . . .	5
4.3 Debug Connector (DBG) . . . . .	6
4.4 Simplicity Connector. . . . .	7
<b>5. Power Supply and Reset . . . . .</b>	<b>8</b>
5.1 Power Selection . . . . .	8
5.2 Board Controller Power. . . . .	8
5.3 EFM32 Reset . . . . .	9
<b>6. Peripherals . . . . .</b>	<b>10</b>
6.1 Push Buttons and LEDs . . . . .	10
6.2 Memory LCD-TFT Display. . . . .	11
6.3 Capacitive Touch Pads . . . . .	12
6.4 Si7021 Relative Humidity and Temperature Sensor . . . . .	12
6.5 USB Micro-B Connector . . . . .	13
6.5.1 Secondary AEM channel . . . . .	13
6.6 USB to UART Bridge . . . . .	14
<b>7. Advanced Energy Monitor . . . . .</b>	<b>15</b>
7.1 Usage . . . . .	15
7.2 Theory of Operation . . . . .	15
7.3 Secondary AEM Channel . . . . .	15
7.4 Accuracy and Performance . . . . .	16
<b>8. On-Board Debugger. . . . .</b>	<b>17</b>
8.1 Debug Modes . . . . .	18
8.2 Debugging During Battery Operation . . . . .	19
<b>9. Board Support Package . . . . .</b>	<b>20</b>
9.1 Application Programming Interface . . . . .	20
9.2 Example Applications . . . . .	20
9.3 How To Include In Your Own Applications . . . . .	20

---

<b>10. Integrated Development Environments . . . . .</b>	<b>21</b>
10.1 Third Party IDEs . . . . .	.22
<b>11. Kit Manager and Upgrades . . . . .</b>	<b>23</b>
11.1 Kit Manager Operation . . . . .	.23
11.2 Firmware Upgrades . . . . .	.23
<b>12. Schematics, Assembly Drawings and BOM . . . . .</b>	<b>24</b>
<b>13. Kit Revision History and Errata . . . . .</b>	<b>25</b>
13.1 Revision History. . . . .	.25
13.2 Errata . . . . .	.25
<b>14. Document Revision History . . . . .</b>	<b>26</b>
<b>Table of Contents . . . . .</b>	<b>27</b>

Silicon Labs

# Simplicity Studio™4



## 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](http://www.silabs.com/IoT)



**SW/HW**  
[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

### Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

### Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



**SILICON LABS**

Silicon Laboratories Inc.  
400 West Cesar Chavez  
Austin, TX 78701  
USA

<http://www.silabs.com>