



## 1. Getting Started

### Hardware

To set up the hardware for the EFM8BB2-SLSTK2021A kit:

1. Provide power to the board by connecting the USB connector to the PC using the provided USB cable.
2. Move the switch to the Advanced Energy Monitor (AEM) position.

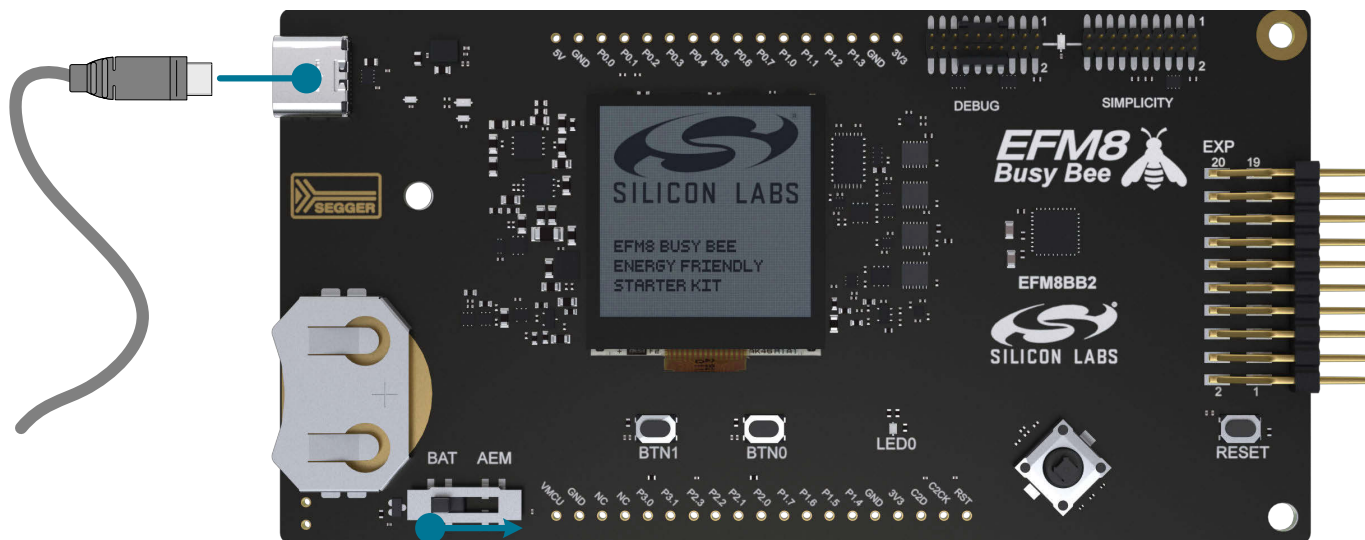


Figure 1.1. Hardware Setup

### Software

The first step to get started with your new EFM8BB2-SLSTK2021A 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 EFM8BB2 Starter Kit. The board comes pre-loaded with the default application, Space Invaders, to interact with while the software downloads.

After downloading the latest version of Simplicity Studio and installing:

1. Select the J-Link adapter for the kit under **[Devices]**.
2. Click one of the demos available under **[Getting Started]>[Demos]** or click the **[Getting Started]>[Demos]>[View All]** to view the entire list of available demos.
3. Click the **[Rainbow Blinky]** demo and click **[Start]** to download and run the demo.

Additional demos showcasing the various features of the EFM8 are also available in Simplicity Studio.

## 2. Kit Block Diagram

An overview of the EFM8BB2 Starter Kit is shown in the figure below.

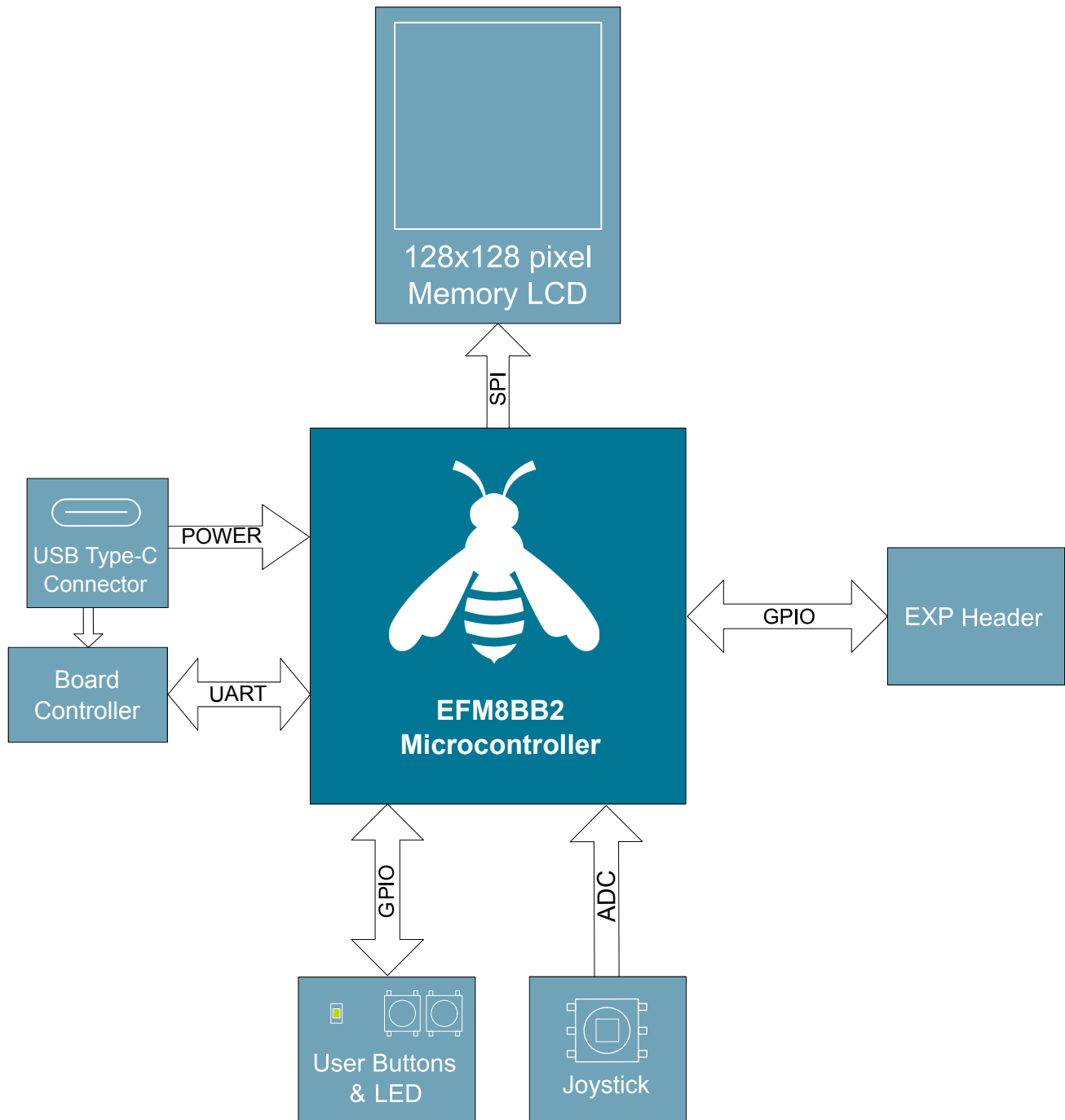


Figure 2.1. EFM8BB2-SLSTK2021A Block Diagram

### 3. Kit Hardware Layout

The layout of the EFM8BB2 Starter Kit is shown below.

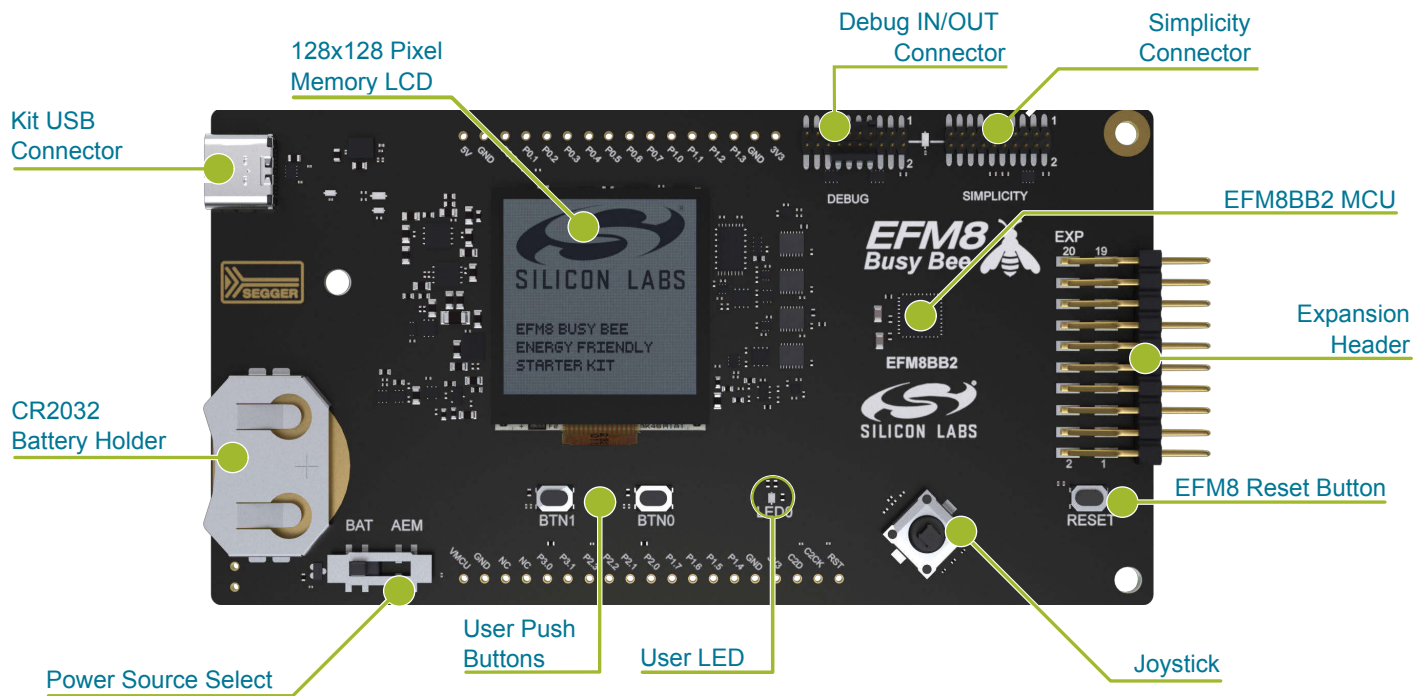


Figure 3.1. EFM8BB2-SLSTK2021A Hardware Layout

The EFM8 device on the kit is connected to several peripherals. The table below shows all of the external connections to the MCU.

Table 3.1. Kit MCU Connections

MCU Port Pin	Port Pin Assigned Function	Primary Board Connection	Secondary Board Connection	Expansion Port Connection (EXP)
P0.0	GPIO			EXP3
P0.1	GPIO	DISP - SPI CS	AGND	
P0.2	Port Match Input	Push Button 0 (PB0)		
P0.3	Port Match Input	Push Button 1 (PB1)		
P0.4	UART0 TX	BC - UART_TX		
P0.5	UART0 RX	BC - UART_RX		
P0.6	SPI0 SCK	DISP - SPI CLK <sup>1</sup>	SPI CLK	EXP8
P0.7	SPI0 MISO		SPI MISO	EXP6
P1.0	SPI0 MOSI	DISP - SPI MOSI	SPI MOSI	EXP4
P1.1	SPI0 NSS		SPI CS	EXP10
P1.2	SMBus0 SDA	I2C SDA	BC - REFCLK	EXP16
P1.3	SMBus0 SCL	I2C SCL	BC - 1 kHz SINE	EXP15
P1.4	PCA0 CEX0	RGB LED0		EXP5
P1.5	PCA0 CEX1	RGB LED1		EXP7
P1.6	PCA0 CEX2	RGB LED2		EXP9

MCU Port Pin	Port Pin Assigned Function	Primary Board Connection	Secondary Board Connection	Expansion Port Connection (EXP)
P1.7	ADC / CMP	Joystick		
P2.0	GPIO	VCOM_ENABLE		
P2.1	UART1 TX	UART1 TX		EXP12
P2.2	UART1 RX	UART1 RX		EXP14
P2.3	GPIO	DISP - Enable	DISP - SPI CS (alt)	EXP11
P3.0 / C2D	C2D (for debug)			
P3.1	GPIO			EXP13

**Note:**

1. This signal has high loading. If the MCU pin is used for signals where edge speed is critical, this signal should be disconnected from the pin.

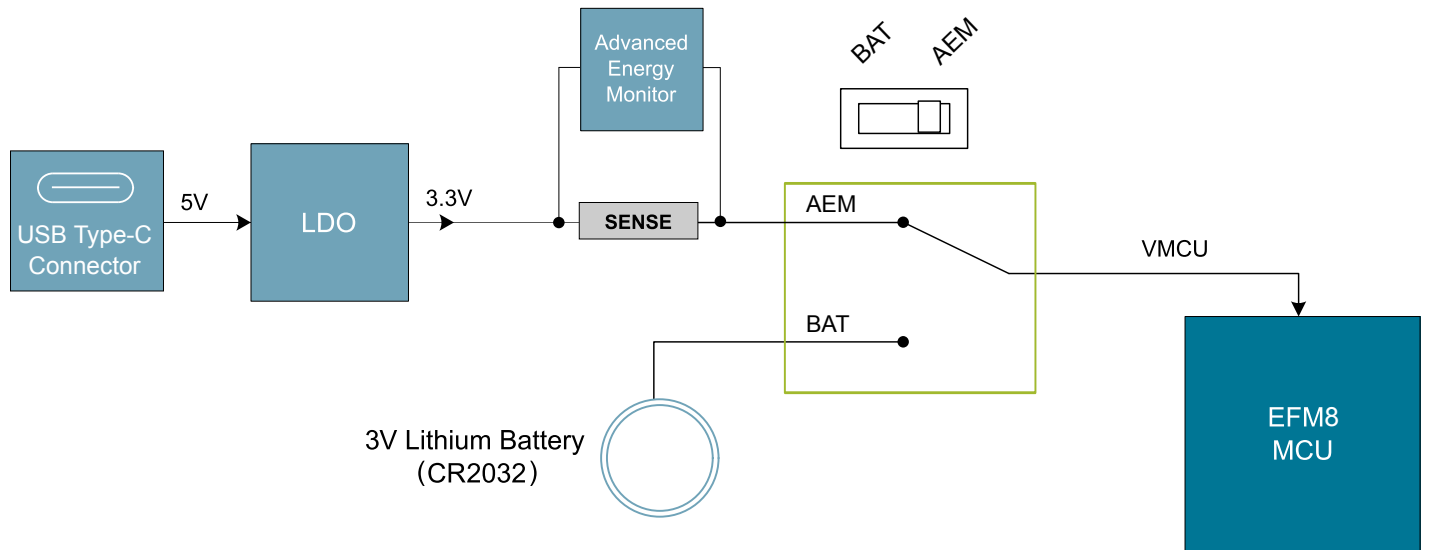
## 4. Power Supply and Reset

### 4.1 MCU Power Selection

The Busy Bee MCU on the EFM8BB2-SLSTK2021A is designed to be powered by three different sources:

- Debug USB cable
- 3 V coin cell battery
- External power source

Select the power source by using the slide switch in the lower left corner of the board. The figure below shows how to select the different power sources using the slide switch.



**Figure 4.1. EFM8BB2-SLSTK2021A Power Switch**

With the switch in the AEM position, an on-board, low-noise LDO with a fixed output voltage of 3.3 V is used to power the MCU. This LDO is powered from the "J-Link" USB cable.

With the switch in the BAT position, the device may be powered from either a 20 mm coin cell battery (in the CR2032 socket) or an external power supply (connected to the VMCU and GND pins on the expansion header).

### 4.2 MCU Reset

The EFM8 MCU can be reset by a few different sources:

- The RESET button.
- An external debugger by pulling the RSTb pin low.

## 5. Peripherals

The starter kit has a set of peripherals that showcase some of the features of the EFM8 Busy Bee microcontroller.

Be aware that most EFM8 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.

### 5.1 Push Buttons and LEDs

The kit has two user push buttons. They are connected to the EFM8, and are debounced by RC filters with a time constant of 1 ms. The buttons are connected to pins P0.2 and P0.3.

In addition to the two push buttons, the kit also features a tri-color LED that is controlled by EFM8 GPIO. The LED is connected to pins P1.4, P1.5, and P1.6 in an active-low configuration.

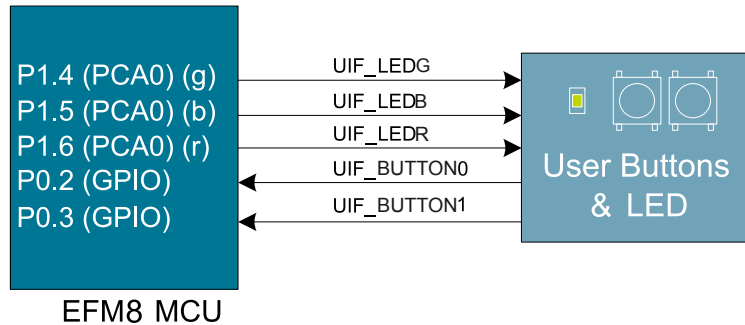


Figure 5.1. Buttons/LEDs

## 5.2 Joystick

The kit has an analog joystick with 8 measurable positions. This joystick is connected to the EFM8 on the P1.7 pin and uses different resistor values to create voltages measurable by the ADC0.

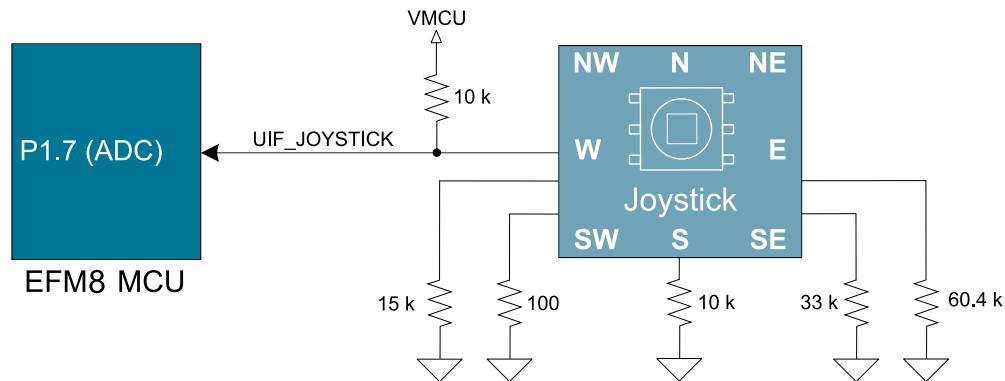


Figure 5.2. Joystick

Table 5.1. Joystick Resistor Combinations

Direction	Resistors Combinations (kΩ)	Expected UIF_JOYSTICK Voltage (V) <sup>1</sup>
Center press	$\frac{0.1}{0.1 + 10}$	0.033
Up (N)	$\frac{60.4}{60.4 + 10}$	2.831
Up-Right (NE)	$\frac{(N \parallel E)}{(N \parallel E) + 10} = \frac{21.34}{21.34 + 10}$	2.247
Right (E)	$\frac{33}{33 + 10}$	2.533
Down-Right (SE)	$\frac{(S \parallel E)}{(S \parallel E) + 10} = \frac{7.67}{7.67 + 10}$	1.433
Down (S)	$\frac{10}{10 + 10}$	1.650
Down-Left (SW)	$\frac{(S \parallel W)}{(S \parallel W) + 10} = \frac{6}{6 + 10}$	1.238
Left (W)	$\frac{15}{15 + 10}$	1.980
Up-Left (NW)	$\frac{(N \parallel W)}{(N \parallel W) + 10} = \frac{12.01}{12.01 + 10}$	1.801

**Note:**  
1. These calculated values assume a VMCU of 3.3 V.



### 5.3 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.

The display interface consists of a SPI-compatible serial interface and some extra control signals. Data are sent to the display one line (128 bits) at a time.

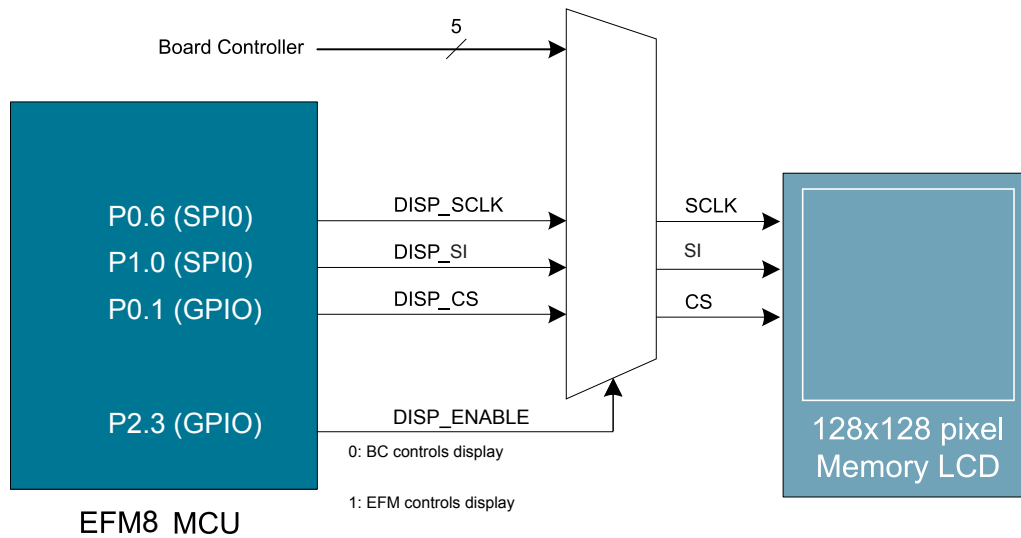
The Memory LCD-TFT display is shared with the kit board controller, allowing the board controller application to display useful information when the user application is not using the display. The EFM8 MCU always controls ownership of the display using the DISP\_ENABLE signal:

- 0: The board controller has control of the display.
- 1: The user application (EFM8) has control of the display.

Data are clocked in on DISP\_SI (P1.0) when DISP\_CS (P0.1) is high, and the clock is sent on DISP\_SCLK (P0.6). The maximum supported clock speed is 1 MHz.

Refer to the display application information for details on driving the display:

<http://www.sharpmemorylcd.com/1-28-inch-memory-lcd.html>



**Figure 5.3. 128x128 Pixel Memory LCD**

## 6. Connectors

### 6.1 Breakout Pads

Many of the EFM8's pins are routed out to "breakout pads" at the top and bottom edges of the kit. A 2.54 mm pitch pin header can be soldered in for easy access to these pins. Most I/O pins are available, with the exception of pins used to drive the LCD.

**Note:** Some of the breakout pads are shared by on-board EFM peripherals. The schematic must be consulted to make sure that it is acceptable to use a shared pin in your application.

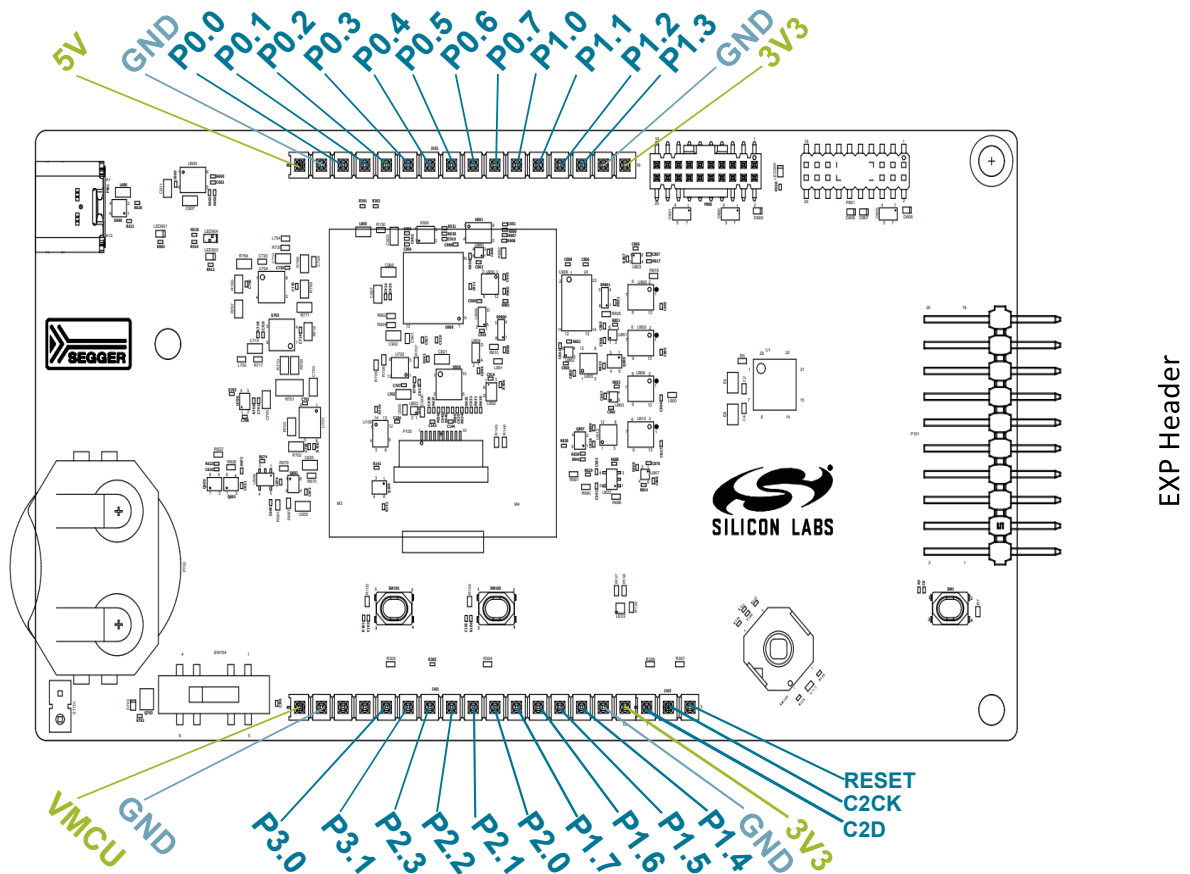
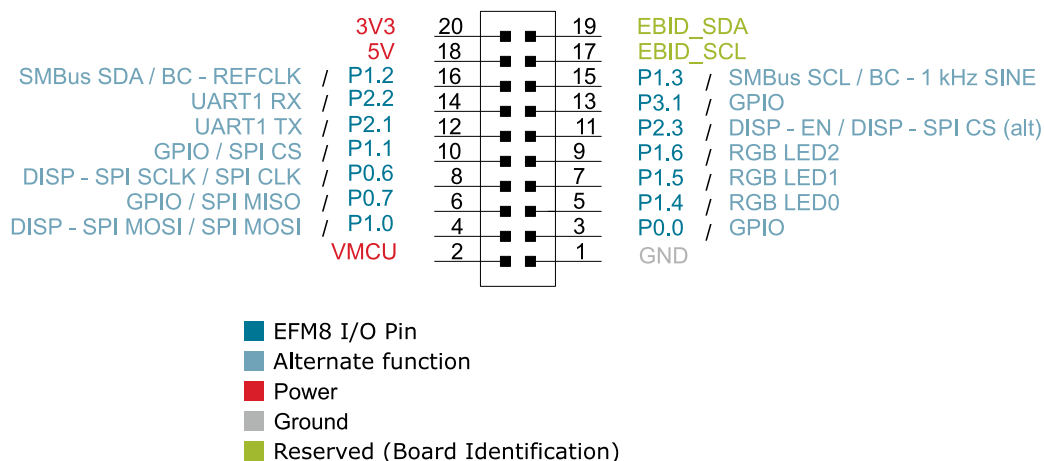


Figure 6.1. Breakout Pads and Expansion Header

## 6.2 Expansion Header

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

The figure below shows the pin assignment of the expansion header. With the exception of a few pins, most of the expansion header's pins are the same as those on the EFM32 Gecko or EFM32 Tiny Gecko starter kits.



**Figure 6.2. Expansion Header**

Some of the chip peripheral functions that are available on the expansion header are listed in the table below.

**Table 6.1. Some Peripheral Functions Available on Expansion Header**

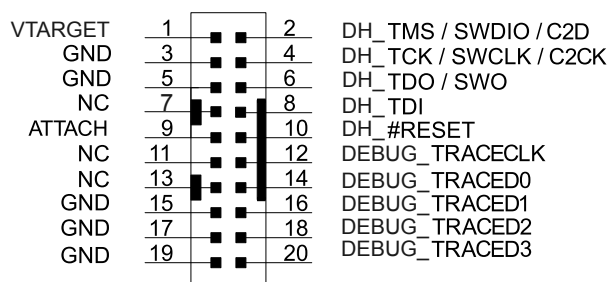
Peripheral	Peripheral Pin	MCU Pin	EXP Header Pin Number
UART1	UART1 TX	P2.1	12
	UART1 RX	P2.2	14
SPI0	SPI0 SCK	P0.6	8
	SPI0 MISO	P0.7	6
	SPI0 MOSI	P1.0	4
	SPI0 CS	P1.1	10
SMBus	SMBus0 SDA	P1.2	16
	SMBus0 SCL	P1.3	15
PCA0	PCA0 CEX0	P1.4	5
	PCA0 CEX1	P1.5	7
	PCA0 CEX2	P1.6	9
I2CSLAVE0	I2CSLAVE0 SDA	P1.5	7
	I2CSLAVE0 SCL	P1.6	9
VREF	VREF	P0.0	3
ADC0	Input	Any supported pin (see Reference Manual for more information)	Multiple
	CNVSTR	P0.6	8

Peripheral	Peripheral Pin	MCU Pin	EXP Header Pin Number
Comparator 0	CMP0P Positive Input	Any supported pin (see Reference Manual for more information)	Multiple
	CMP0N Negative Input	Any supported pin (see Reference Manual for more information)	Multiple
Comparator 1	CMP1P Positive Input	Any supported pin (see Reference Manual for more information)	Multiple
	CMP1N Negative Input	Any supported pin (see Reference Manual for more information)	Multiple

**Note:** This table only sums up some of the alternate functions available on the expansion header. Consult the EFM8BB22F16G data sheet for a complete list of alternate functions.

### 6.3 Debug Connector

This connector is used for Debug In and Debug Out (see chapter on Debugging).



**Figure 6.3. Debug Connector**

**Table 6.2. Debug Connector Pinout**

Pin Number	Function	Note
1	VTARGET	Target voltage on the debugged application. <b>Note:</b> This connection is required and is needed for the debug circuitry to match voltage levels with the target device.
2	DH_TMS/SWDIO/C2D	JTAG TMS, Serial Wire data I/O, or EFM8 C2 data I/O
4	DH_TCK/SWCLK/C2CK	JTAG TCK, Serial Wire clock, or EFM8 C2 clock
6	DH_TDO/SWO	JTAG TDO or Serial Wire Output
8	DH_TDI	JTAG data in
9	ATTACH	This signal must be pulled to ground by the external debugger or application for cable insertion detection.
10	DH_#RESET	Target MCU reset.
12	DEBUG_TRACECLK	Trace clock
14, 16, 18, 20	DEBUG_TRACED0-3	Trace data (4 lines)
7, 11, 13	NC	Not Connected
3, 5, 15, 17, 19	GND	

### 6.4 Direct Debug Connector

This connector (J103) is used for directly debugging the EFM8 using an external debug adapter (see chapter on Debugging). This is especially useful for debugging the MCU on the STK board when the part is battery-powered or powered by an external supply.

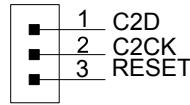


Figure 6.4. Direct Debug Connector

Table 6.3. Direct Debug Connector Pinout

Pin Number	Function	Note
1	C2D	EFM8 C2 data I/O
2	C2CK	EFM8 C2 clock
3	RESET	

### 6.5 Simplicity Connector

The top-right corner of the board includes a 20-pin Simplicity connector. The connector contains some I/O pins that can be used with some of the EFM8 Busy Bee's features. Additionally, the 3V3 and 5V power rails are also exported.

The figure below shows the pin assignment of the Simplicity connector.

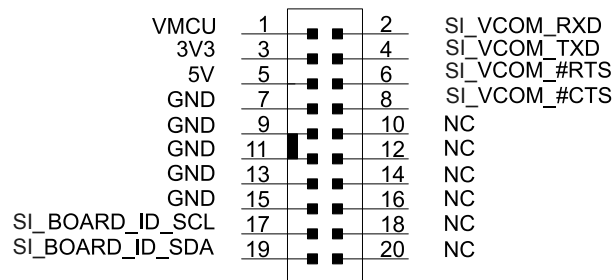


Figure 6.5. Simplicity Header

## 7. Integrated Development Environment

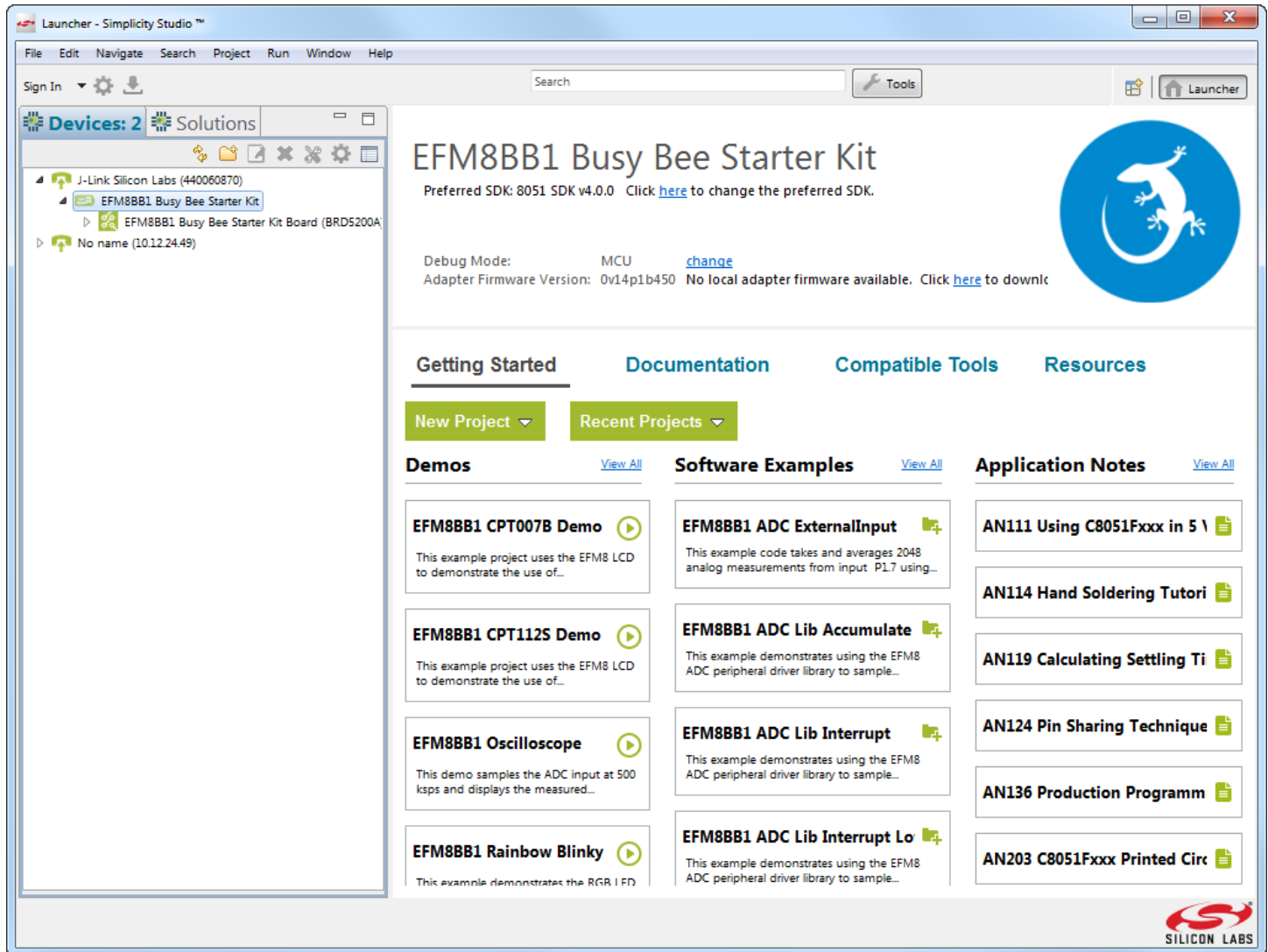


Figure 7.1. Simplicity Studio

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. Select the J-Link adapter for the kit under **[Devices]**.
4. Click the **[Getting Started]>[New Project]** button.
5. In the wizard, ensure the EFM8BB2 Busy Bee Starter Kit Board kit and click **[Next]**.
6. Select **[Example]** and click **[Next]**.
7. Select the desired example or demo from the list and click **[Next]**.
8. Click **[Finish]**.
9. Click the **[Debug]** button in the IDE to build and download the code to the hardware.
10. Follow the instructions at the top of the main example file to set up the hardware as needed.
11. Click the **[Resume]** button to start running the example.

## 8. Advanced Energy Monitor

When the EFM8BB2 Starter Kit is connected to a Silicon Labs STK, the STK's Advanced Energy Monitor (AEM) hardware also measures the slstk2021a power consumption using the VMCU connection on the EXP header. By using the [Energy Profiler] in Simplicity Studio, current consumption and voltage can be measured in real time.

More details about AEM and its operation can be found in the STK User Guide. Note that AEM will measure the current for all circuitry connected to VMCU, including the STK MCU and the CPT device.

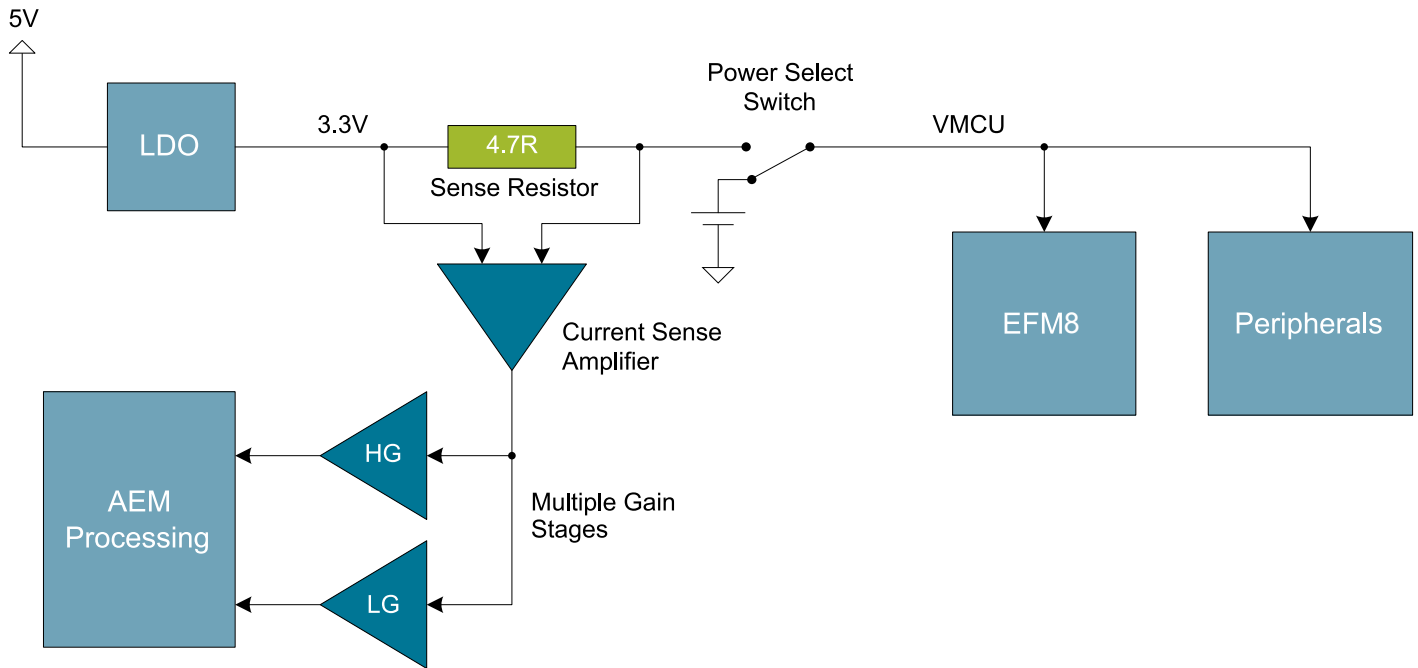


Figure 8.1. Measuring EFM8BB2-SLSTK2021A Current Using AEM

### 8.1 Usage

The AEM data is collected by the board controller and can be displayed by the energyAware Profiler, available through Simplicity Studio. By using the energyAware Profiler, current consumption and voltage can be measured in realtime.

## 8.2 AEM Theory of Operation

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.

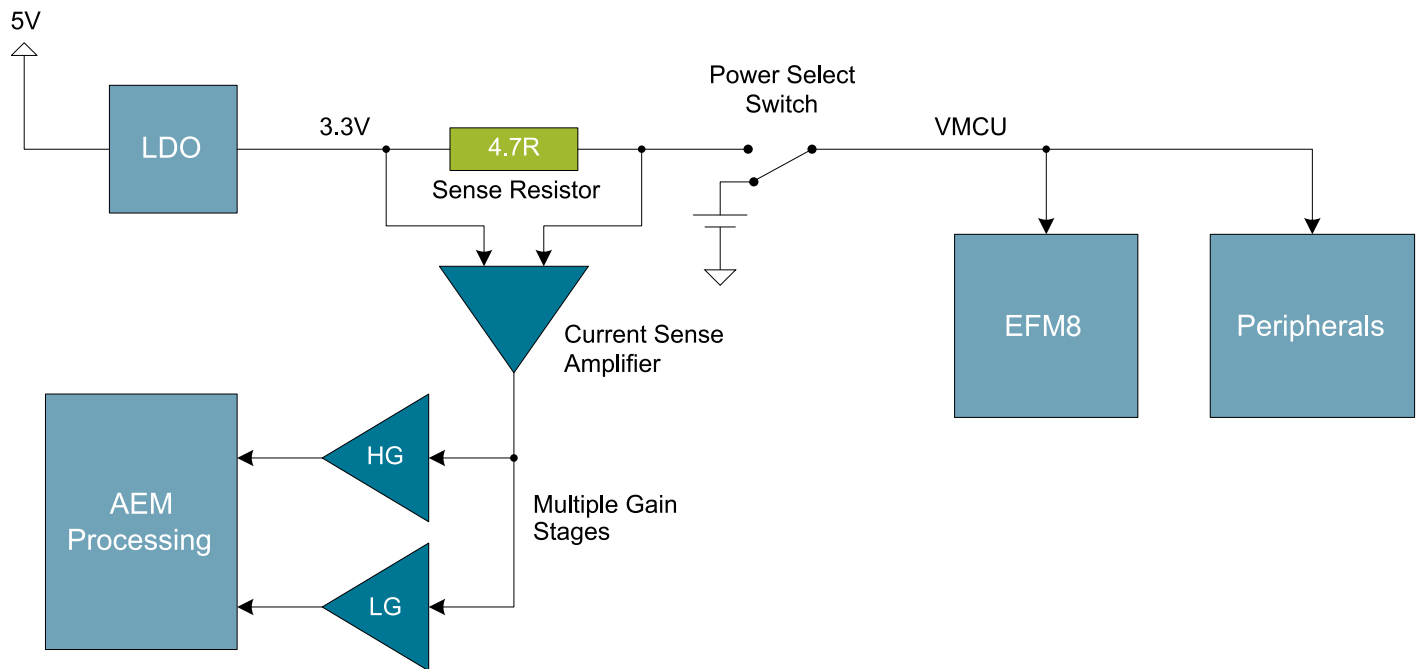


Figure 8.2. Advanced Energy Monitor

## 8.3 AEM Accuracy and Performance

The AEM 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.

**Note:** The current measurement will only be correct when powering the EFM8 from USB power through the debugger (power select switch set to DBG or AEM).



## 9. Board Controller

The kit contains a board controller that is responsible for performing various board-level tasks, such as handling the debugger and the Advanced Energy Monitor. An interface is provided between the EFM8 and the board controller in the form of a UART connection. The connection is enabled by setting the VCOM\_ENABLE (P2.0) line high, and using the lines VCOM\_TX (P0.4) and VCOM\_RX (P0.5) for communicating.

The BC enable signal connects the EFM8 to the board controller:

- 0: EFM8 UART pins are isolated from the board controller.
- 1: EFM8 UART pins are connected to the board controller (default upon reset).

**Note:** The board controller is only available when USB power is connected.

## 10. Debugging

The EFM8BB2-SLSTK2021A contains an integrated debugger, which can be used to download code and debug the Busy Bee EFM8 MCU. In addition to programming the MCU on the kit, the debugger can also be used to program and debug external Silicon Labs EFM8 devices.

### 10.1 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 EFM8 MCU on the kit by setting the debug mode to **[In]**. A summary of the different supported debug modes is given in [Table 10.1 Debug Modes on page 18](#).

**Table 10.1. Debug Modes**

Mode	Description
Debug MCU	In this mode, the on-board debugger is connected to the EFM8 on the EFM8BB2-SLSTK2021A.
Debug In	In this mode, the on-board debugger is disconnected, and an external debugger can be connected to debug the EFM8 on the EFM8BB2-SLSTK2021A.
Debug Out	In this mode, the on-board debugger can be used to debug an EFM8 mounted on a custom board.

Select the active debug mode by using the drop-down menu in the Kit Manager tool, which is available through Simplicity Studio.

When using the debug adapter in the **[Out]** mode, the end device must be manually detected before debugging and programming. To do this:

1. Right-click on the kit from the Simplicity Studio launch screen and select **[Select Target Part...]**.
2. If needed, select the appropriate **[Target Interface]** for the external device. For example, EFM8 devices will use the **[C2]** selection.
3. In the same dialog, click the **[Detect Target]** button.
4. Click **[OK]** to close the dialog. The external target can now be debugged and programmed.

### 10.2 Debugging during Battery Operation

When the EFM8 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.

To enable debugging when the USB cable is removed, connect an external debugger to the MCU Debug Header in the top right corner of the EFM8BB2-SLSTK2021A instead of the Debug IN/OUT Connector. This header is connected directly to the EFM8's debug interface. The pinout of this header is shown in the Connectors chapter.

## 11. Upgrading the Kit

Upgrading the kit firmware is done through Simplicity Studio in the **[Launcher]** perspective. Simplicity Studio will automatically check for new updates on startup. After selecting a kit in the left under **[Devices]**, the area at the top of the Launcher page displays the current kit version and Debug Mode. Use the links in this area to change the debug mode or upgrade the kit firmware.

## 12. Schematics, Assembly Drawings, and BOM

The schematics, assembly drawings, and bill of materials (BOM) for the EFM8BB2 Starter Kit board are available through Simplicity Studio when the kit documentation package has been installed.

### 12.1 Board Revision History

**Table 12.1. Board Revision History**

Kit Revision	Released	Description
Variant B Rev.A01	23 March 2022	Update to new STK platform
C01	5 April 2020	Updated kit to new platform
A04	30 October 2019	Changed joystick to L-KLS7-MT-03
A03	19 December 2017	Un-mounted D801
A02	21 August 2017	Updated MCU to revision C
A01	23 March 2015	Updated PCB revision
A00	1 December 2014	Initial Release

## 13. Revision History

### Revision 0.5

July, 2022

- Updated for Variant B, Rev A01.
- USB changed from Mini-B to Type-C, wherever applicable.
- Direct debug connector updated as per current design.
- Reference Board Connector is renamed as Simplicity Connector.

### Revision 0.4

January, 2018

- Updated with the correct pin label for EXP13.

### Revision 0.3

June, 2016

- Updated screenshots and instructions for Simplicity Studio v4.
- Added [12.1 Board Revision History](#).
- Added the UG237 document reference.
- Added a note to that VTARGET is required.
- Added the reference for the direct debug connector to .
- Added a note about high loading on the DISP CLK signal.

### Revision 0.2

February, 2015

- Updated screenshots and instructions for Simplicity Studio v3.
- Updated formatting.

### Revision 0.1

2014

- Initial revision.

# 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 & Community**

[www.silabs.com/community](http://www.silabs.com/community)

## 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 to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required, or Life Support Systems 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. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

## 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®, Gecko OS, Gecko OS Studio, ISOModem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, 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. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

<http://www.silabs.com>