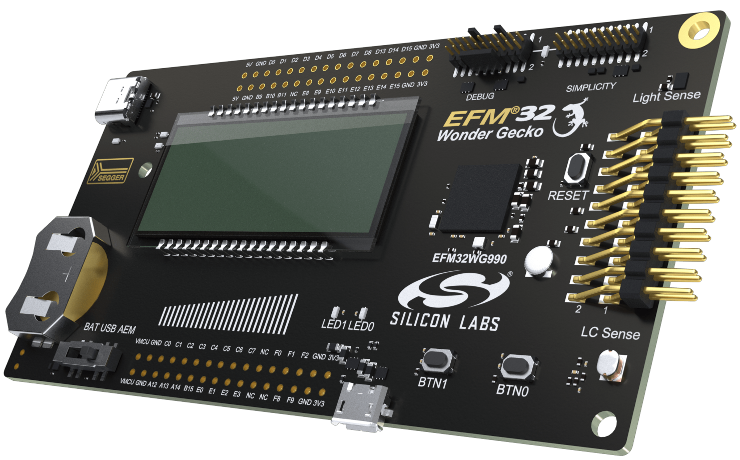
UG419: EFM32WG Gecko Starter Kit User's Guide

The EFM32WG-STK3800 is an excellent starting point to become familiar with the EFM32WG™ Gecko Microcontroller.

The starter kit contains sensors and peripherals demonstrating some of the EFM32WG's many capabilities. The kit provides all necessary tools for developing an EFM32WG Gecko application.



TARGET DEVICE

- EFM32WG Gecko Microcontroller (EFM32WG990F256-B-BGA112)
- CPU: 32-bit ARM® Cortex-M4
- Memory: 256 kB flash and 32 kB RAM

KIT FEATURES

- USB connectivity
- Advanced Energy Monitor (AEM)
- SEGGER J-Link on-board debugger
- Debug multiplexer supporting external hardware as well as on-board MCU
- 8x20 segment LCD
- Inductive LC sensor
- Photo transistor for light sensing applications
- USB Micro-AB connector supporting both host and device mode USB applications
- 2 push buttons and 2 LEDs connected to EFM32 for user interaction
- Backup battery
- 4-segment Capacitive Touch Slider
- 20-pin 2.54 mm header for expansion boards
- Breakout pads for direct access to I/O pins
- Power sources include USB and CR2032 coin cell battery

SOFTWARE SUPPORT

- Simplicity Studio™
- IAR Embedded Workbench
- Keil MDK

Table of Contents

1. Introduction

1.1 Description

The EFM32WG-STK3800 is an ideal starting point for application development on the EFM32WG Gecko Microcontrollers. The board features sensors and peripherals, demonstrating some of the many capabilities of the EFM32WG Gecko Microcontroller. Additionally, the board is a fully featured debugger and energy monitoring tool that can be used with external applications.

1.2 Features

- EFM32WG Gecko Microcontroller
  - 256 kB Flash
  - 32 kB RAM
  - BGA112 package
- Advanced Energy Monitoring system for precise current and voltage 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 include USB and CR2032 battery
- 8x20 segment LCD
- USB Micro-AB connector supporting both host and device mode USB applications
- 2 push buttons and 2 LEDs connected to EFM32 for user interaction
- LC tank circuit for inductive proximity sensing of metallic objects
- Photo transistor for light sensing applications
- Backup battery
- 4-segment capacitive touch slider
- Crystals for LFXO and HFXO: 32.768 kHz and 48.000 MHz.

1.3 Getting Started

Detailed instructions for how to get started with your new EFM32WG-STK3800 can be found on the Silicon Labs Web pages:

https://www.silabs.com/mcu/32-bit/efm32-wonder-gecko

2. Kit Block Diagram

An overview of the EFM32WG Gecko Starter Kit is shown in the figure below.

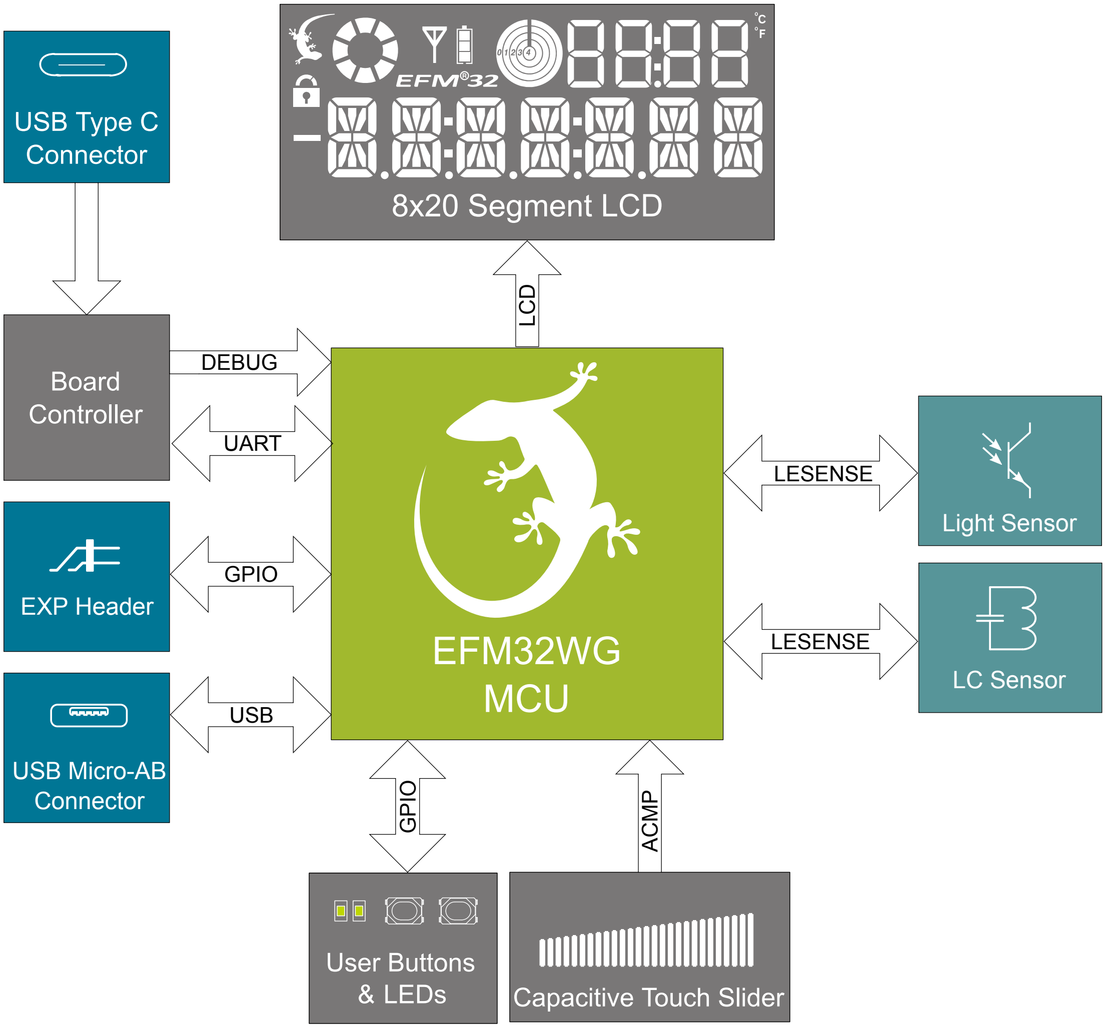


Figure 2.1. Kit Block Diagram

3. Kit Hardware Layout

The EFM32WG Gecko Starter Kit layout is shown below.

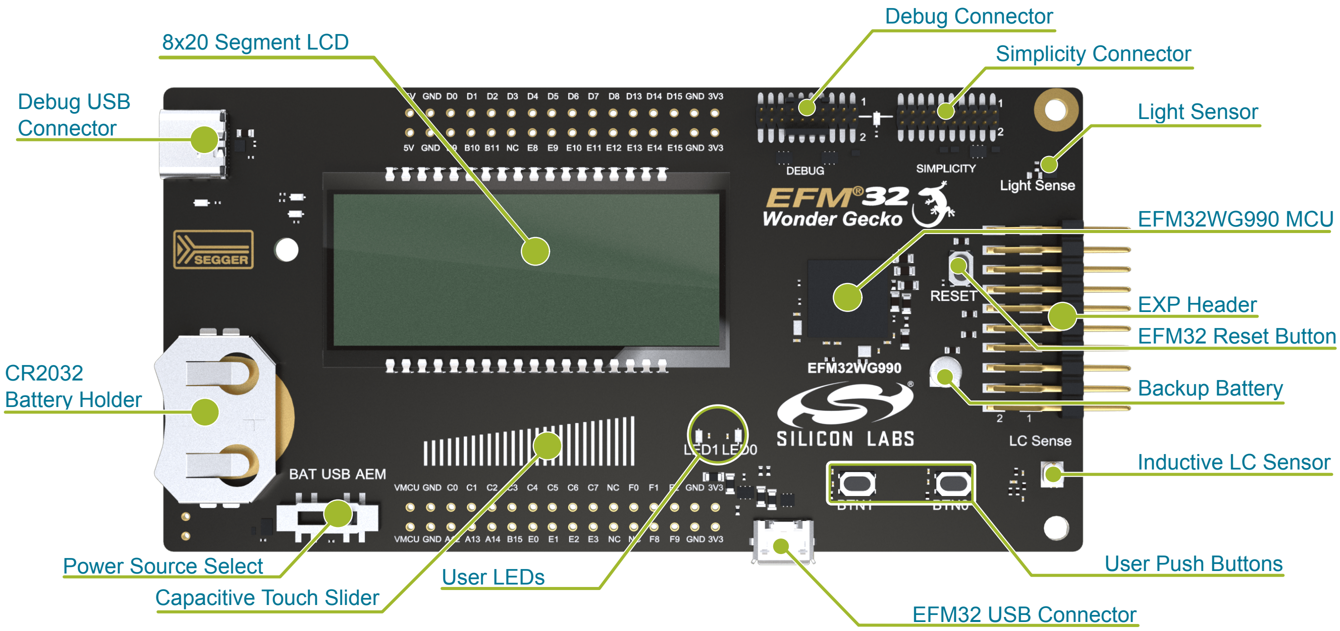


Figure 3.1. EFM32WG-STK3800 Hardware Layout

4. Connectors

4.1 Breakout Pads

Most of the EFM32WG'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 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 tradeoffs.

The figure below shows the pinout of the breakout pads and the pinout of the EXP header on the right edge of the board. The EXP header is further explained in the next section. The breakout pad connections are also printed in silkscreen next to each pin for easy reference.

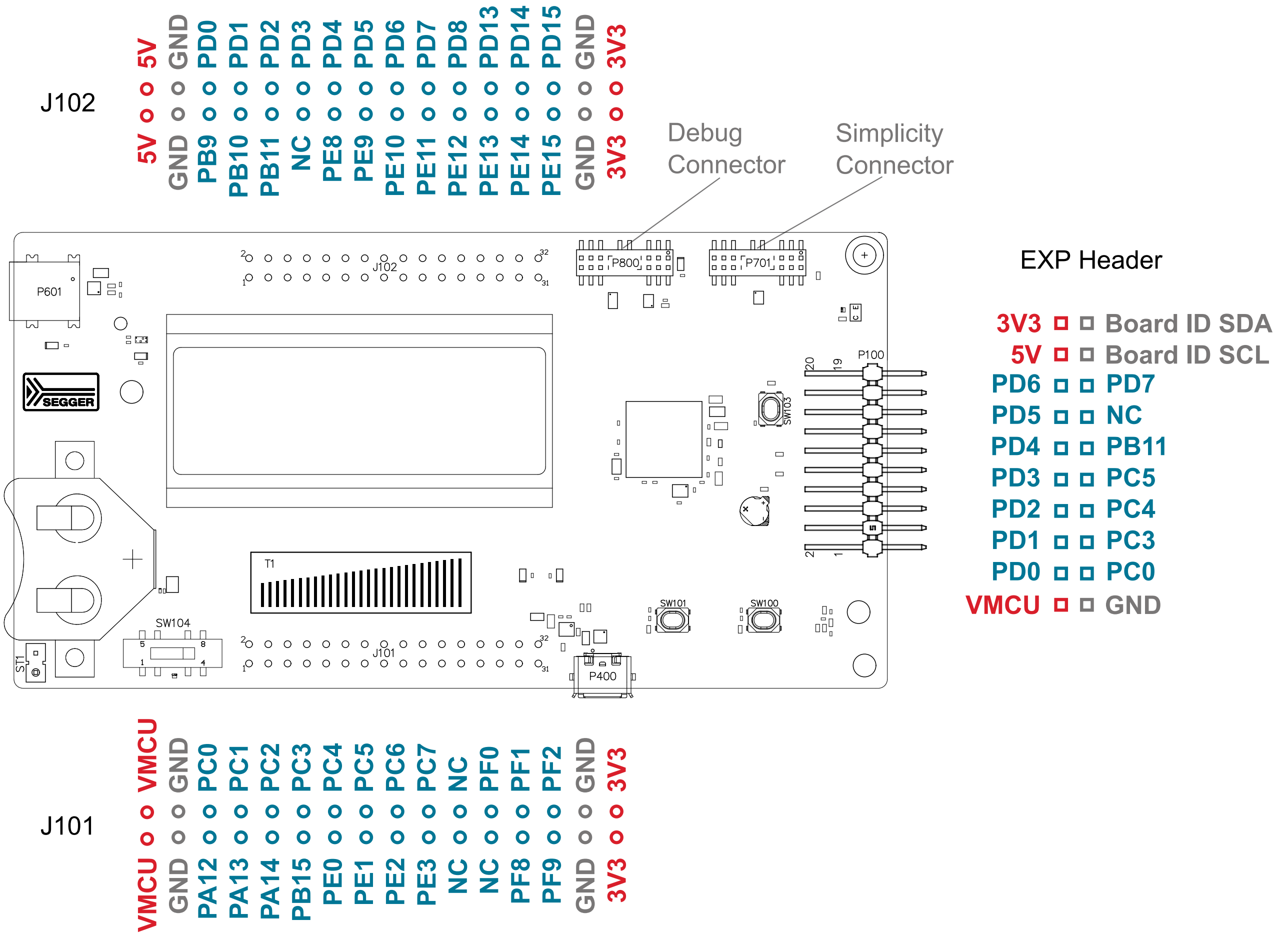


Figure 4.1. Breakout Pads and Expansion Header

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

Table 4.1. Bottom Row (J101) Pinout

Pin	EFM32WG I/O Pin	Shared Feature	Pin	EFM32WG I/O Pin	Shared Feature
1	VMCU	EFM32WG voltage domain	2	VMCU	EFM32WG voltage domain
3	GND	Ground	4	GND	Ground
5	PA12	LCD_BCAP_P	6	PC0	EXP3
7	PA13	LCD_BCAP_N	8	PC1	GPIO

Pin	EFM32WG I/O Pin	Shared Feature	Pin	EFM32WG I/O Pin	Shared Feature
9	PA14	LCD_BEXT	10	PC2	GPIO
11	PB15	GPIO	12	PC3	EXP5
13	PE0	VCOM_TX	14	PC4	EXP7
15	PE1	VCOM_RX	16	PC5	EXP9
17	PE2	UIF_LED0	18	PC6	LES_LIGHT_SENSE
19	PE3	UIF_LED1	20	PC7	LES_LC_SENSE
21	NC	–	22	NC	–
23	NC	–	24	PF0	DEBUG_SWCLK
25	PF8	GPIO	26	PF1	DEBUG_SWDIO
27	PF9	GPIO	28	PF2	DEBUG_SWO
29	GND	Ground	30	GND	Ground
31	3V3	Board controller supply	32	3V3	Board controller supply

Table 4.2. Top Row (J102) Pinout

Pin	EFM32WG I/O Pin	Shared Feature	Pin	EFM32WG I/O Pin	Shared Feature
1	5V	Board USB voltage	2	5V	Board USB voltage
3	GND	Ground	4	GND	Ground
5	PB9	UIF_BUTTON0	6	PD0	EXP4
7	PB10	UIF_BUTTON1	8	PD1	EXP6
9	PB11	EXP11	10	PD2	EXP8
11	NC	–	12	PD3	EXP10, OPAMP_N2
13	PE8	GPIO	14	PD4	EXP12, OPAMP_P2
15	PE9	GPIO	16	PD5	EXP14, OPAMP_OUT2
17	PE10	GPIO	18	PD6	EXP16, LES_LIGHT_EXCITE
19	PE11	GPIO	20	PD7	EXP15
21	PE12	GPIO	22	PD8	BU_VIN
23	PE13	GPIO	24	PD13	GPIO
25	PE14	GPIO	26	PD14	GPIO
27	PE15	GPIO	28	PD15	GPIO
29	GND	Ground	30	GND	Ground
31	3V3	Board controller supply	32	3V3	Board controller supply

4.2 EXP Header

On the right side of the board, an angled 20-pin EXP 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 EFM32WG Gecko's features. Additionally, the VMCU, 3V3, and 5V power rails are also exposed.

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

The figure below shows the EXP header pin assignment for the EFM32WG Gecko Starter Kit. Because of limitations in the number of available GPIO pins, some of the EXP header pins are shared with kit features.

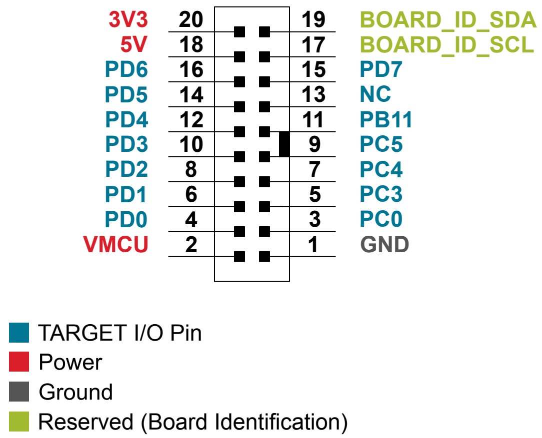


Figure 4.2. EXP Header

Table 4.3. EXP Header Pinout

Pin	Connection	EXP Header Function	Shared Feature	Peripheral Mapping
20	3V3	Board controller supply		
18	5V	Board controller USB voltage		
16	PD6	I2C_SDA	LES_LIGHT_EXCITE	I2C0_SDA #1
14	PD5	UART_RX	OPAMP_OUT2	LEUART0_RX #0
12	PD4	UART_TX	OPAMP_P2	LEUART0_TX #0
10	PD3	SPI_CS	OPAMP_N2	USART1_CS #1
8	PD2	SPI_SCLK	–	USART1_CLK #1
6	PD1	SPI_MISO	–	USART1_RX #1
4	PD0	SPI_MOSI	–	USART1_TX #1
2	VMCU	EFM32WG 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	PD7	I2C_SCL	–	I2C0_SCL #1
13	NC	–	Install R300 to connect to PB12/DAC_LC_EXCITE	DAC0_OUT1
11	PB11	DAC_OUT	–	DAC0_OUT0

Pin	Connection	EXP Header Function	Shared Feature	Peripheral Mapping
9	PC5	GPIO	–	–
7	PC4	GPIO	–	–
5	PC3	GPIO	–	–
3	PC0	GPIO	–	–
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. If the "Debug IN" mode is selected, the connector allows an external debugger to be used with the on-board EFM32WG. If the "Debug OUT" mode is selected, the connector allows the kit to be used as a debugger towards an external target. If the "Debug MCU" mode (default) is selected, the 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 pins on the breakout header.

The pinout of the connector follows that of the standard ARM Cortex Debug 19-pin connector. The pinout is described in detail below. Note that even though the connector supports JTAG in addition to Serial Wire Debug, it does not necessarily mean that the kit or the on-board target device supports these.

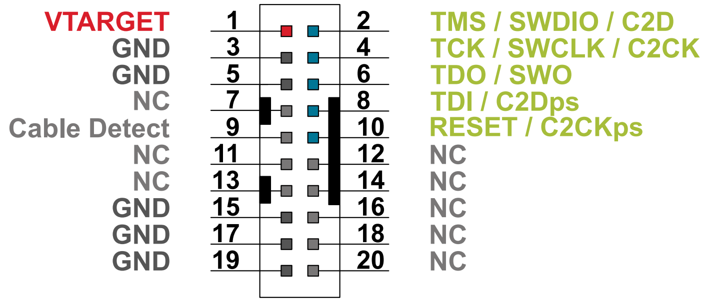


Figure 4.3. Debug Connector

Even though the pinout matches the pinout of an ARM Cortex Debug connector, these are not fully compatible as pin 7 is physically removed from the Cortex Debug connector. Some cables have a small plug that prevents 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
12	NC	TRACECLK
14	NC	TRACED0
16	NC	TRACED1
18	NC	TRACED2
20	NC	TRACED3
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 EFM32WG Gecko Starter Kit enables advanced debugging features such as the AEM and Virtual COM port to be used towards an external target. The pinout is illustrated in the figure below.

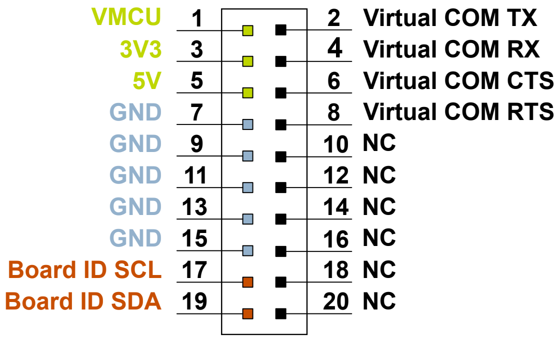


Figure 4.4. Simplicity Connector

The signal names in the figure and the pin description table are referenced from the board controller. This means that VCOM_TX should be connected to the RX pin on the external target, VCOM_RX to the target's TX pin, VCOM_CTS to the target's RTS pin, and VCOM_RTS to the target's CTS pin.

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	Virtual COM TX
4	VCOM_RX	Virtual COM RX
6	VCOM_CTS	Virtual COM CTS
8	VCOM_RTS	Virtual COM RTS
17	BOARD_ID_SCL	Board ID SCL
19	BOARD_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 EFM32WG on the starter kit can be powered by one of these sources:

- The debug USB cable; or
- the EFM32WG's own USB regulator; 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. The figure below shows how the different power sources can be selected with the slide switch.

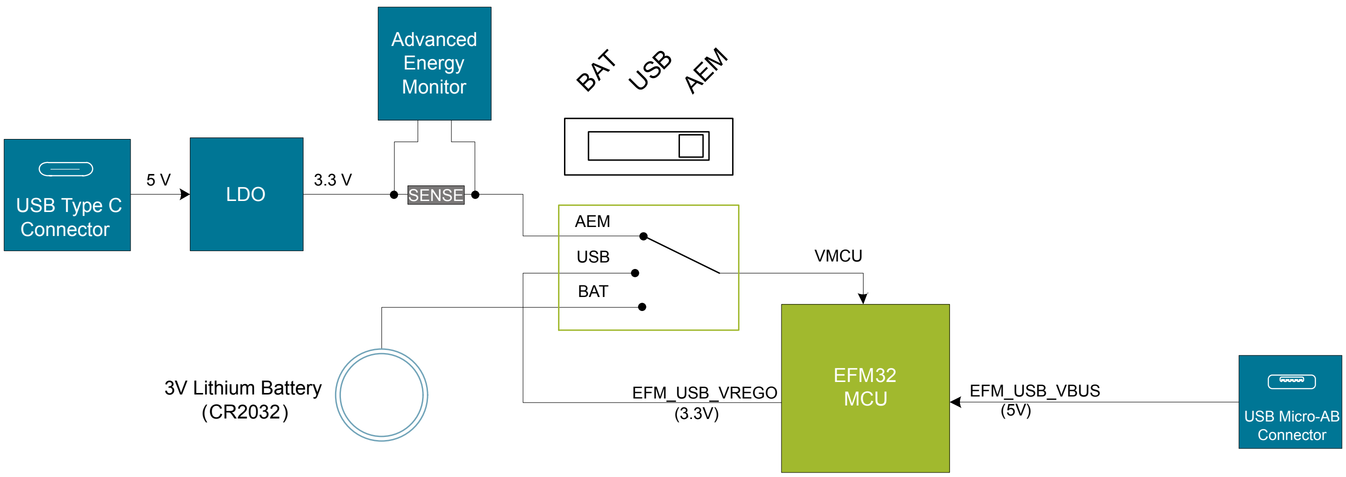


Figure 5.1. Power Switch

With the switch in the **AEM** position, a low-noise 3.3 V LDO on the starter kit is used to power the EFM32WG. 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 EFM32WG 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. Current consumption monitoring using the AEM is not available when USB is selected as power source.

Finally, 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.

5.2 Board Controller Power

The board controller is responsible for important features, such as the debugger and the AEM, 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 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 EFM32WG device will continue to operate in the **USB** and **BAT** modes.

5.3 EFM32WG Reset

The EFM32WG 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, a reset to the EFM32WG will also be issued during board controller boot-up. This means that removing power to the board controller (unplugging 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 EFM32WG features.

Note that most EFM32WG I/Os routed to peripherals are also routed to the breakout pads or the EXP header, which must be taken into consideration when using these I/Os.

6.1 Push Buttons and LEDs

The kit has two user push buttons marked BTN0 and BTN1. They are connected directly to the EFM32WG and are debounced by RC filters with a time constant of 1 ms. The buttons are connected to pins PB9 and PB10.

The kit also features two yellow LEDs marked LED0 and LED1 that are controlled by GPIO pins on the EFM32WG. The LEDs are connected to pins PE2 and PE3 in an active-high configuration.

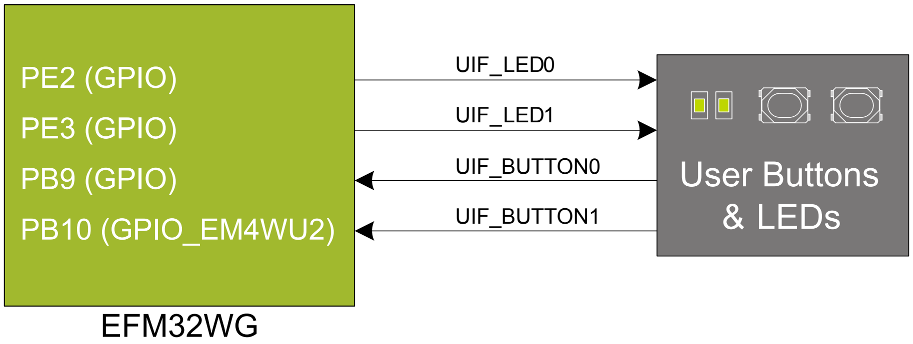


Figure 6.1. Buttons and LEDs

6.2 LCD

A 28-pin segment LCD is connected to the EFM32's LCD peripheral. The LCD has 8 common lines and 20 segment lines, giving a total of 160 segments in octaplex mode. These lines are not shared on the breakout pads. Refer to the kit schematic for information on signals to segments mapping.

A capacitor connected to the EFM32 LCD peripheral's voltage boost pin is also available on the kit.

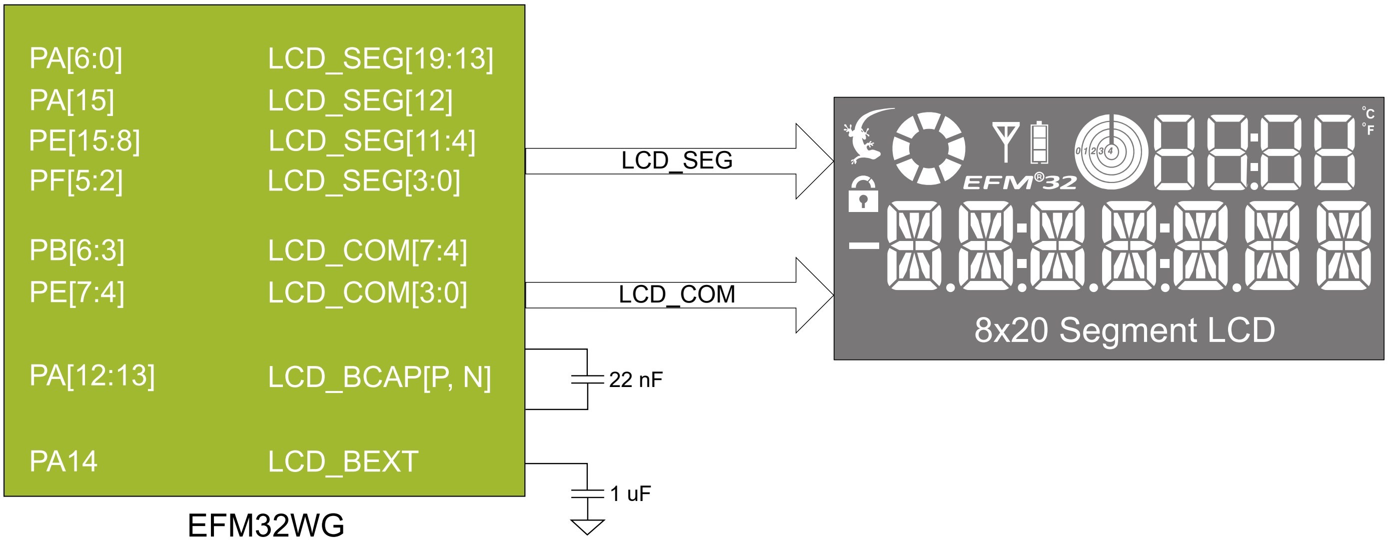


Figure 6.2. Segment LCD

6.3 Capacitive Touch Slider

A touch slider utilizing the capacitive touch capability of the EFM32WG's analog comparator (ACMP) is located on the bottom side of the board. It consists of four interleaved pads which are connected to PC8, PC9, PC10, and PC11.

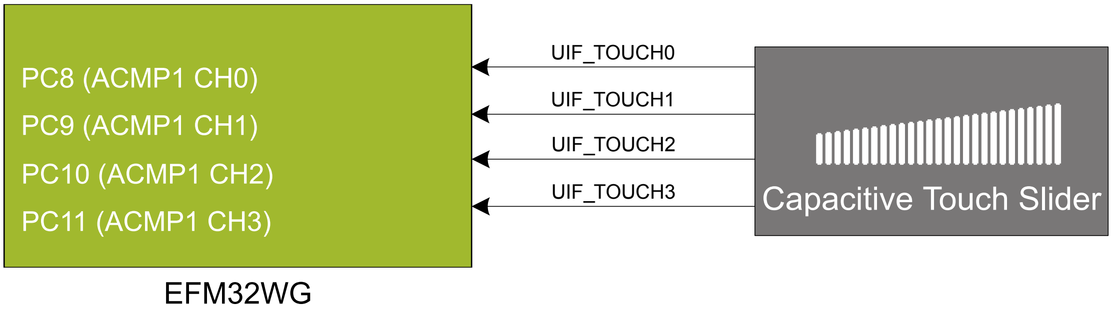


Figure 6.3. Touch Slider

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 EFM32WG's analog comparator (ACMP) in capacitive touch sensing mode. For low-power operation, the Low Energy Sensor Interface (LESENSE) can be configured to continuously scan all pads.

Sensing change in capacitance is done by setting up the touch pad as part of an RC relaxation oscillator, where the analog comparator counts the number of oscillations for a fixed period of time.

For more information about usage and theory of low-energy capacitive sensing, refer to application note, "AN0028: Low Energy Sensor Interface -- Capacitive Sense", which is available in Simplicity Studio or in the document library on the Silicon Labs website.

6.4 LC Sensor

An inductive-capacitive sensor for demonstrating the Low Energy Sensor Interface (LESENSE) is located on the bottom right of the board. The LESENSE peripheral uses the voltage digital-to-analog converter (VDAC) to set up an oscillating current through the inductor and then uses the analog comparator (ACMP) to measure the oscillation decay time. The oscillation decay time will be affected by the presence of metal objects within a few millimeters of the inductor.

The LC sensor can be used for implementing a sensor that wakes up the EFM32WG from sleep when a metal object comes close to the inductor, which again can be used as a utility meter pulse counter, door alarm switch, position indicator or other applications where one wants to sense the presence of a metal object.

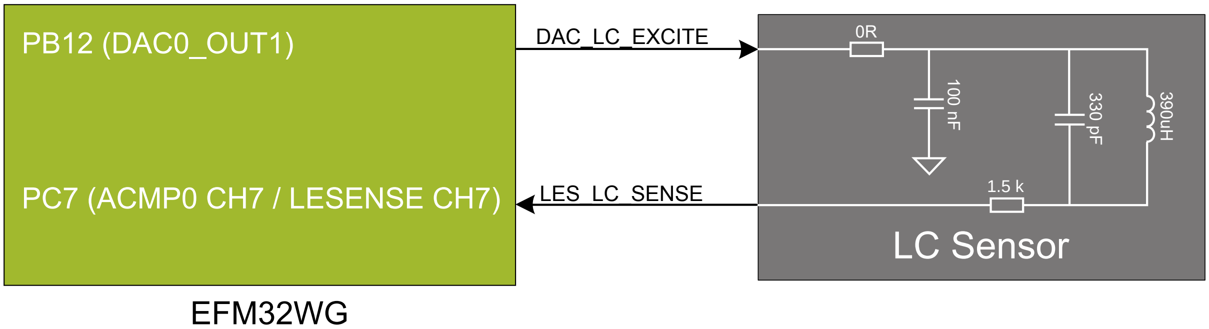


Figure 6.4. LC Metal Sensor

For more information about the LC sensor usage and operation, refer to the application note, "AN0029: Low Energy Sensor Interface -- Inductive Sense", which is available in Simplicity Studio or in the document library on the Silicon Labs website.

6.5 Ambient Light Sensor

In the top right corner of the board there is an ambient light sensor implemented using a TEMT6200FX01 photo transistor connected to the EFM32WG's Low Energy Sensor Interface (LESENSE) peripheral as a resistive sensor element. One pin is used for excitation of the transistor, while another senses the state of the sensor. LESENSE can take care of both the excitation and sensing part of the operation.

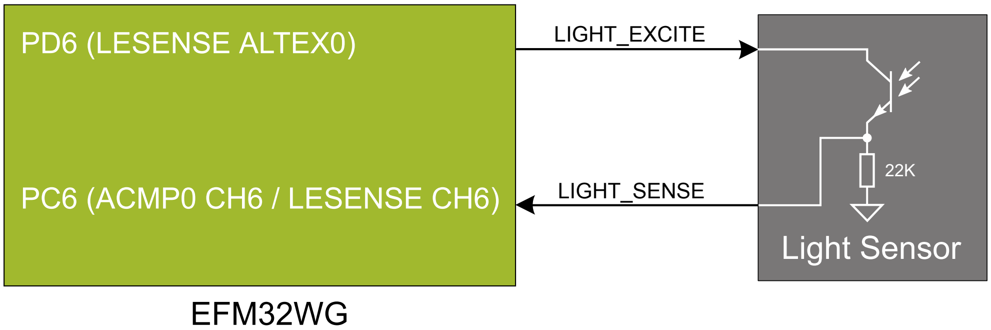


Figure 6.5. Ambient Light Sensor

For more information about using LESENSE for resistive sensor applications, refer to application note, "AN0036: Low Energy Sensor Interface -- Resistive Sense", which is available in Simplicity Studio or in the document library on the Silicon Labs website.

6.6 USB Micro-AB Connector

The EFM32WG-STK3800 board is equipped with a USB Micro-AB connector interfacing with the EFM32WG's USB peripheral. This allows the development and evaluation of applications using USB in both host and device mode. For host mode, the board can supply 5 V power to the USB VBUS if the board itself has been powered using the debug-USB connector. The VBUS power switch also provides an overcurrent flag which can be read in order to detect if a connected device draws too much current.

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.

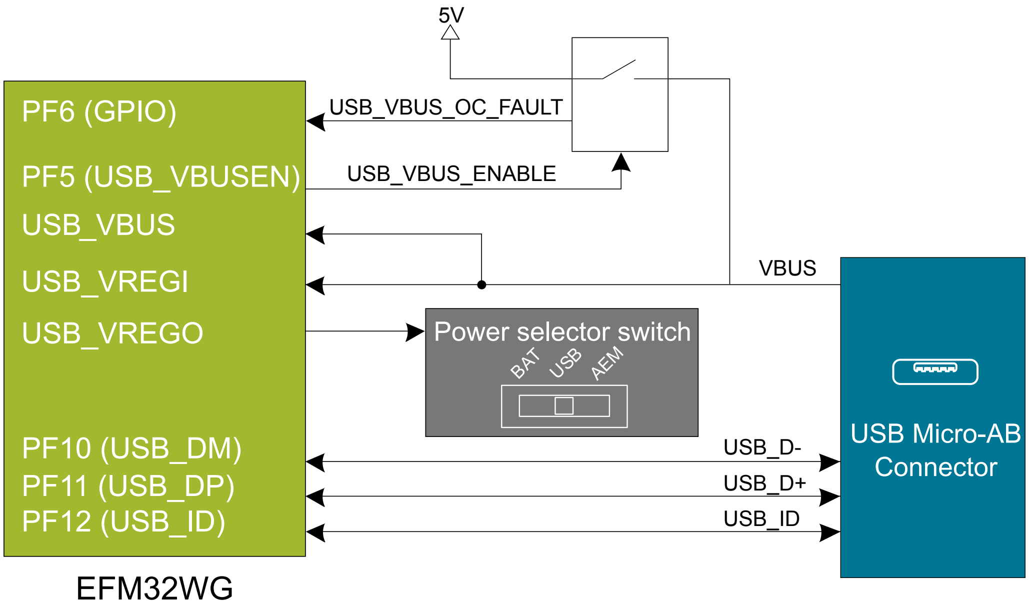


Figure 6.6. USB Connector and Power Supply

6.7 Opamp Footprint

On the board's back side there is a silk screen diagram showing and operational amplifier (opamp) with unpopulated positive and negative feedback circuits. The circuit connects to one of the EFM32WG's internal opamps, and allows the user to build an opamp circuit by installing 0603 size passive components in the available footprints. Various types of passive components (resistors, capacitors, inductors, ferrites) can be used to implement the desired amplifier circuit, such as a positive or negative feedback amplifier, buffer amplifier or active filter circuit.

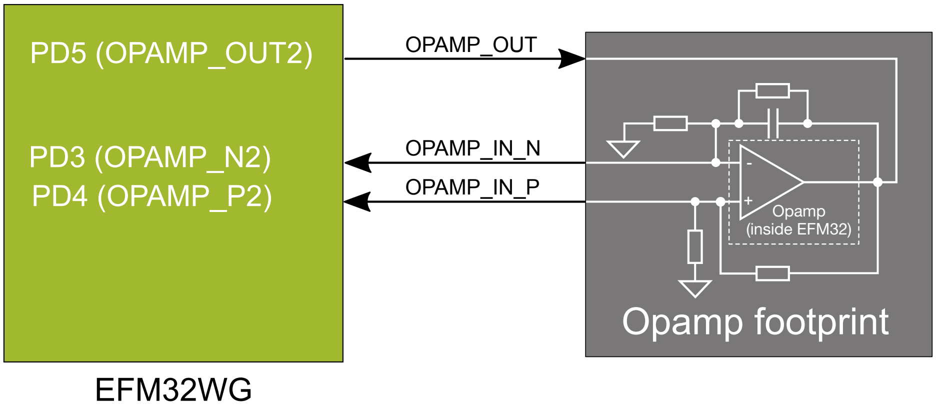


Figure 6.7. Opamp Footprint

Note: The opamp symbol in the figure represents the EFM32WG's integrated opamp peripheral. There is no discrete opamp or footprint for installing a discrete opamp on the board.

6.8 Virtual COM Port

An asynchronous serial connection to the board controller is provided for application data transfer between a host PC and the target EFM32WG, which eliminates the need for an external serial port adapter.

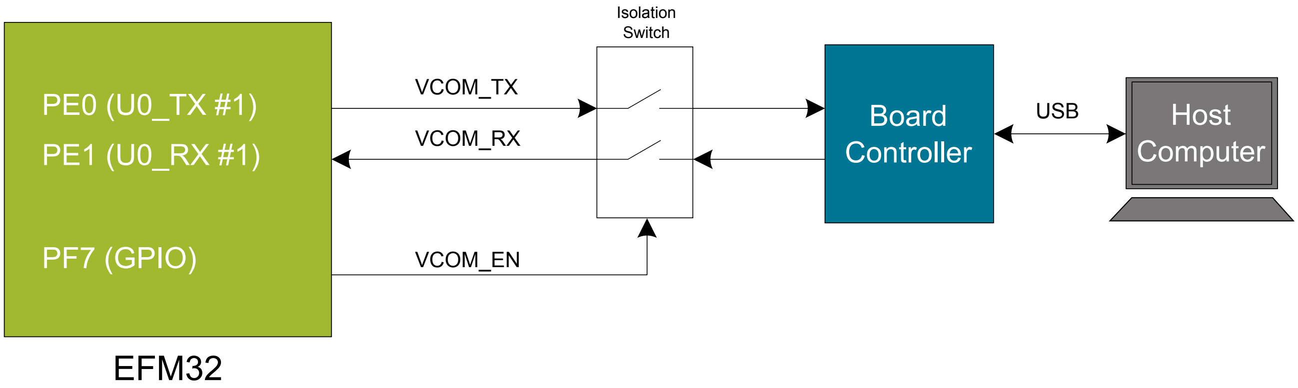


Figure 6.8. Virtual COM Port Interface

The Virtual COM port consists of a physical UART between the target device and the board controller, and a logical function in the board controller that makes the serial port available to the host PC over USB. The UART interface consists of two pins and an enable signal.

Table 6.1. Virtual COM Port Interface Pins

Signal	Description
VCOM_TX	Transmits data from the EFM32WG to the board controller
VCOM_RX	Receives data from the board controller to the EFM32WG
VCOM_ENABLE	Enables the VCOM interface, allowing data to pass through to the board controller

Note: The VCOM port is only available when the board controller is powered, which requires the J-Link USB cable to be inserted.

7. Advanced Energy Monitor

7.1 Usage

The Advanced Energy Monitor (AEM) 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 EFM32WG in realtime.

7.2 Theory of Operation

To accurately measure current ranging from 0.1 µA to 47 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. 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 kit startup, an automatic calibration of the AEM is performed, which compensates for the offset error in the sense amplifiers.

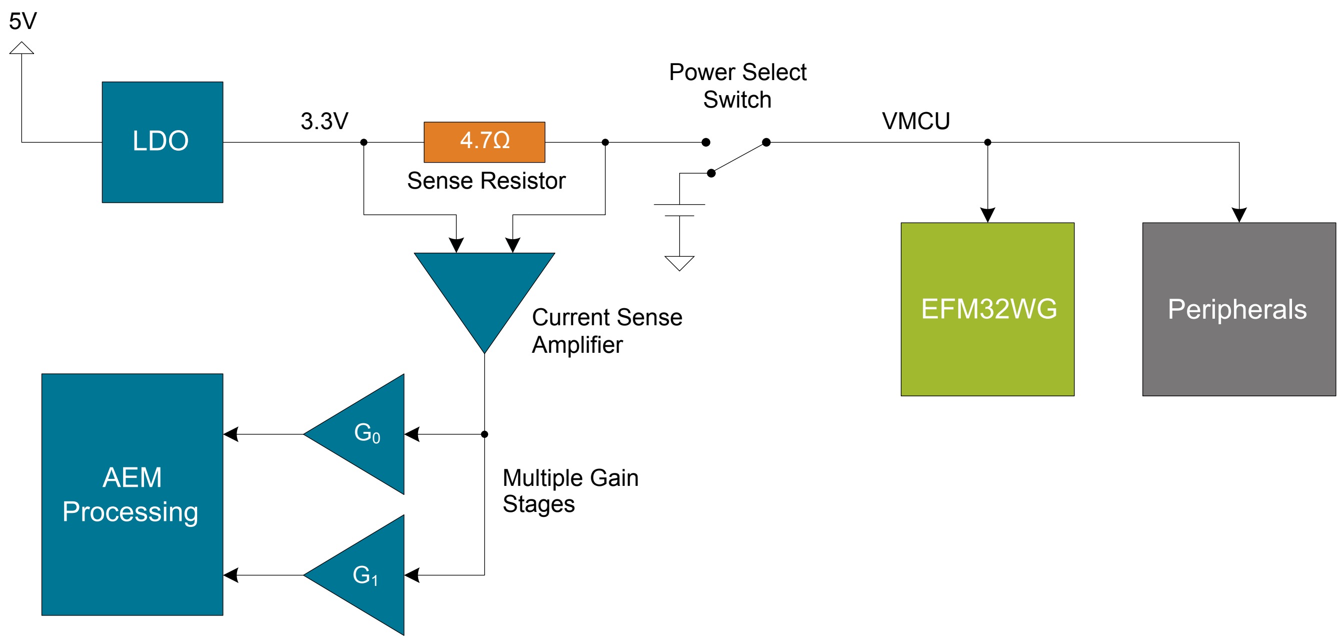


Figure 7.1. Advanced Energy Monitor

7.3 Accuracy and Performance

The AEM is capable of measuring currents in the range of 0.1 µA to 47 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. Although 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 EFM32WG-STK3800 contains an integrated debugger, which can be used to download code and debug the EFM32WG. In addition to programming the EFM32WG 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, which is used with all EFM32, EFR32, and EZR32 devices
- JTAG, which can be used with EFR32 and some EFM32 devices
- C2 Debug, which is used with EFM8 devices

To ensure accurate debugging, use the appropriate debug interface for your device. The debug connector on the board supports all three of these modes.

8.1 Debug Modes

To program external devices, use the debug connector to connect to a target board and set the debug mode to [**Out**]. The same connector can also be used to connect an external debugger to the EFM32WG MCU on the kit by setting debug mode to [**In**].

Selecting the active debug mode is done in Simplicity Studio.

Debug MCU: In this mode, the on-board debugger is connected to the EFM32WG on the kit.

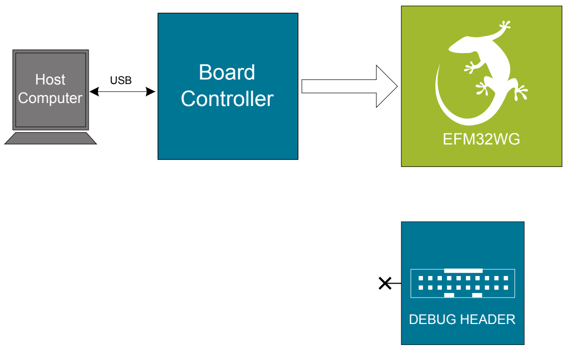


Figure 8.1. Debug MCU

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

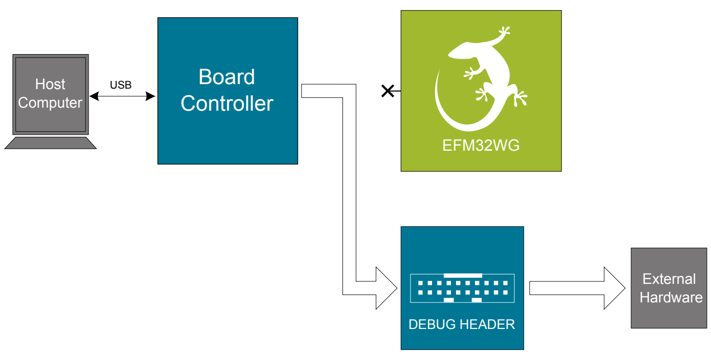


Figure 8.2. Debug OUT

Debug IN: In this mode, the on-board debugger is disconnected and an external debugger can be connected to debug the EFM32WG on the kit.

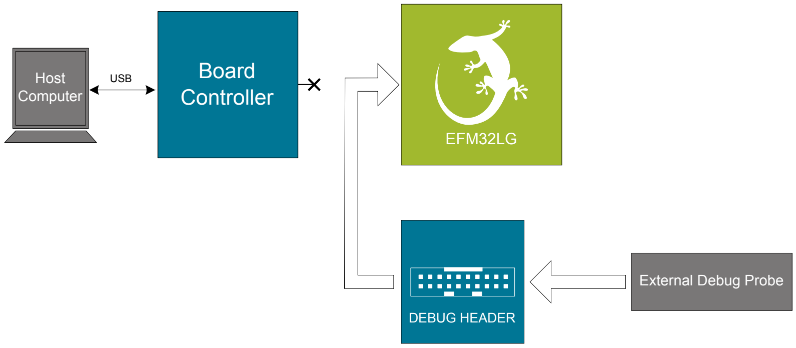


Figure 8.3. Debug IN

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

8.2 Debugging During Battery Operation

When the EFM32WG is battery-powered 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 off another energy source, such as a battery, and the board controller is powered down, make direct connections to the GPIOs used for debugging, which are exposed on the breakout pads.

9. Kit Configuration and Upgrades

The kit configuration dialog in Simplicity Studio allows you to change the J-Link adapter debug mode, upgrade its firmware, and change other configuration settings. To download Simplicity Studio, go to silabs.com/simplicity.

In the main window of the Simplicity Studio's Launcher perspective, the debug mode and firmware version of the selected J-Link adapter are shown. Click the [**Change**] link next to any of these settings to open the kit configuration dialog.

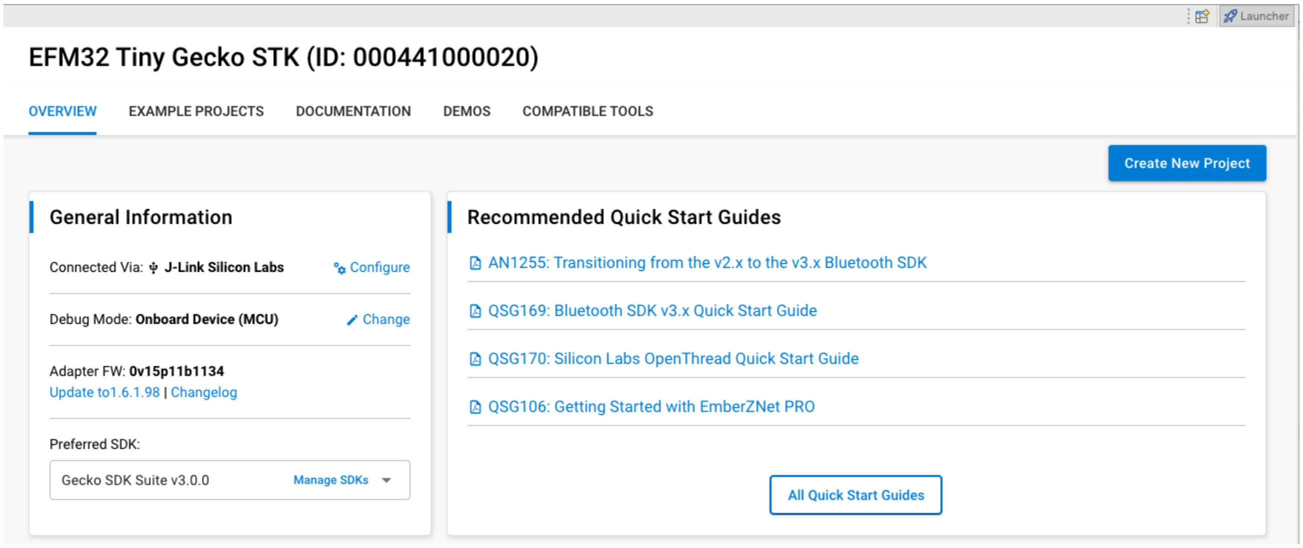


Figure 9.1. Simplicity Studio Kit Information

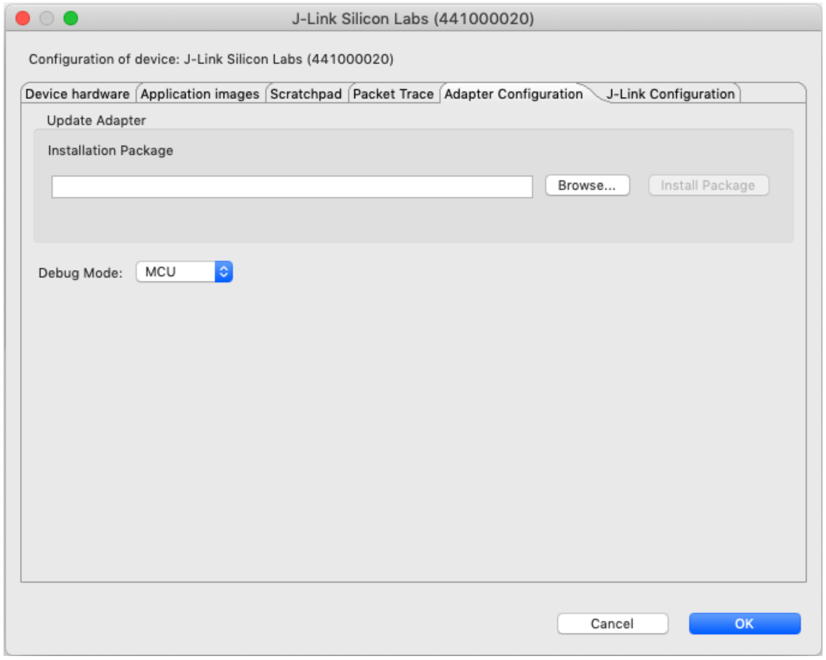


Figure 9.2. Kit Configuration Dialog

9.1 Firmware Upgrades

You can upgrade the kit firmware through Simplicity Studio. Simplicity Studio will automatically check for new updates on startup.

You can also use the kit configuration dialog for manual upgrades. Click the [**Browse**] button in the [**Update Adapter**] section to select the correct file ending in `.emz`. Then, click the [**Install Package**] button.

10. Schematics, Assembly Drawings, and BOM

Schematics, assembly drawings, and bill of materials (BOM) are available through Simplicity Studio when the kit documentation package has been installed. They are also available from the kit page on the Silicon Labs website: silabs.com.

11. Kit Revision History and Errata

11.1 Revision History

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

Figure 11.1. Revision Info

Table 11.1. Kit Revision History

Kit Revision	Released	Description
D02	21 November 2023	Updated Kit as per latest board revision (BRD2400B_A01).
D01	5 june 2023	Kit revised due to the removal of USB Enumeration cables from the Kit contents.
D00	3 February 2023	Kit revised due to variant changed from BRD2400A to BRD2400B and changed packaging.
C01	24 April 2020	Kit revised due to BRD2400A upped to C01.
C00	18 December 2016	Kit revision up to C00.
B01	18 December 2016	Kit revised due to BRD2400A upped to C00.
B00	28 Spetember 2016	Removal of coin cell GPCR2032 C1.
A01	12 September 2013	Update in kit content.
A00	20 December 2012	Initial kit revision

11.2 Errata

There are no known errata at present.

12. Document Revision History

2.00

July, 2024

Updated user guide to reflect new board revision (BRD2400B A01).

1.00

May, 2020

Updated in conjunction with release of new kit revision.

0.11

January, 2014

Updated description of the integrated debugger in the feature list. Added missing section on debugging.

0.10

January, 2013

Initial document version.

Simplicity Studio

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

IoT Portfolio	SW/HW	Quality	Support & Community
www.silabs.com/IoT	www.silabs.com/simplicity	www.silabs.com/quality	www.silabs.com/community

SILICON LABS

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

www.silabs.com