

AN0018.1: Supply Voltage Monitoring



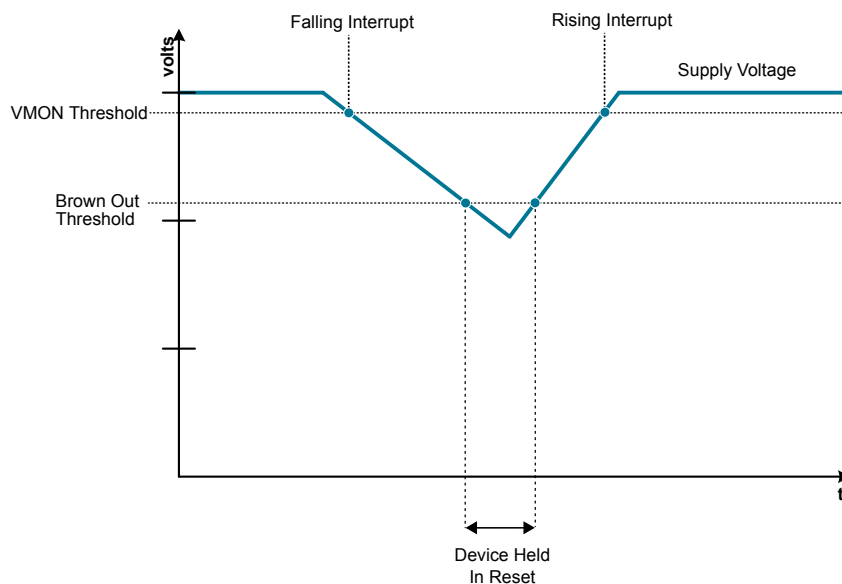
This application note will demonstrate how to use the EFM32 EMU's Voltage Monitor monitor the supply voltage and the Reset Management Unit to determine if a brown-out reset has occurred.

This application note includes:

- This PDF document
- Source files (zip)
 - Example c-code
 - Multiple IDE projects

KEY POINTS

- Observe RMU to determine if a brown-out has occurred
- Monitor power rails for low voltage conditions with VMON
- Low-current voltage monitoring down to EM4 with VMON



1. Device Compatibility

This application note supports multiple device families, and some functionality is different depending on the device.

MCU Series 1 consists of:

- EFM32 Jade Gecko (EFM32JG1/EFM32JG12)
- EFM32 Pearl Gecko (EFM32PG1/EFM32PG12)
- EFM32 Giant Gecko (EFM32GG11)

Wireless SoC Series 1 consists of:

- EFR32 Blue Gecko (EFR32BG1/EFR32BG12/EFR32BG13)
- EFR32 Flex Gecko (EFR32FG1/EFR32FG12/EFR32FG13)
- EFR32 Mighty Gecko (EFR32MG1/EFR32MG12/EFR32MG13)

2. Reset Management Unit

The Reset Management Unit (RMU) monitors the reset lines and ensures reliable operation by resetting the microcontroller if any of the supply voltages are insufficient or if certain events occur. Such events are activation of the external reset pin, watchdog timeout, kernel lockup or a system reset request. In addition, the RMU sets the reset cause register which can be used to determine the cause of the last reset at startup.

2.1 Power-on Reset (POR) and Brown-out Detection

The POR ensures that the device does not start up before the AVDD supply voltage has reached the threshold voltage $VPOR_{thr}$ (roughly 1.2V). Before the POR threshold voltage is reached, the device is kept in reset state. Once the POR threshold is reached, the device's Brown Out Detectors engage, keeping the device in reset until their thresholds are met.

If, during device operation, the supply voltage falls below the one of the brown-out detector threshold voltages, a "brown-out" has occurred. The brown-out detector that has had its threshold crossed will pull the microcontroller into reset to prevent unexpected program execution and data corruption. EFM32 and EFR32 Series 1 devices feature two brown-out detectors, one each for the DVDD and AVDD rails.

Once the RMU initiates a reset, the reset cause register is written to specify the reset source. If the supply voltage has fallen below the BOD threshold, but not below the POR threshold, the RMU will report the corresponding brown-out reset cause. If the supply voltage has fallen below the POR threshold, a POR reset cause will be reported.

2.2 Reading the Reset Cause Register

The reset cause register indicates which reset source triggered the last reset. Because several bits can be set at once and multiple reset causes can occur simultaneously, it is important to look for the reset cause in the following order to discover the main cause of reset:

1. Power-on reset
2. AVDD brown-out reset
3. DVDD brown-out reset
4. Pin reset
5. Lockup and system request reset
6. Watchdog reset

When following this list from the smallest to largest number, the first indicated reset cause will be the actual reset cause. The reset cause register needs to be cleared after it has been read, otherwise it may indicate a wrong cause of reset at the next startup. To clear the reset cause register a 1 has to be written to the RCCLR bit in RMU_CMD and the HRCCLR bit in EMU_AUXCTRL.

The EFM32 software library contains functions to read and clear the reset cause register.

3. Voltage Monitor

The EFM32 features an extremely low energy Voltage Monitor (VMON) capable of running down to EM4 Hibernate. Trigger points are preloaded but may be reconfigured. Voltage monitors exist for the following rails:

- AVDD X 2¹
- DVDD
- IOVDD²
- BUVDD³

Note:

1. Some devices only have one AVDD rail and, thus, one AVDD voltage monitor.
2. Separate IOVDD monitors exist for each IOVDD rail, on those devices with multiple IOVDD rails.
3. BUVDD monitor is not available on all devices.

The voltage monitor provides an easy way to warn if the supply voltage reaches a critical level. While the RMU automatically initiates a reset if the voltage level is too low, the voltage monitor compares the supply voltage to an internal production-calibrated bandgap reference. The trigger voltage level can be scaled in 20 mV increments between 1.62 V and 3.4 V. By adjusting the trigger level it is possible to take action before the microcontroller enters a brown-out reset or to verify that the supply voltage is within a specified range. Additionally, the AVDD rail has separate rising and falling threshold settings.

The voltage monitor threshold status can be read from the VMON interrupt flag bits in the EMU_IF register. This register contains separate flags for both rising and falling edges for each threshold, and can be used to interrupt the system when a threshold is crossed.

3.1 VMON Events

Table 3.1. VMON Events

Feature	Condition	AVDD	DVDD	BUVDD	IOVDD
Hysteresis (separate rise and fall triggers)	-	Yes	-	-	-
Supply switch to/from Backup	Fall/Rise	Yes	-	-	-
Interrupt	Fall or Rise	Yes	Yes	Yes	Yes
Wake-Up from EM4 Hibernate	Fall or Rise	Yes	Yes	Yes	Yes

The enable status of each VMON channel is reflected in the EMU_STATUS register.

The status of the sticky interrupt can be found at EMU_IF. These interrupt flags also serve as the wake-up source of EM4H when the associated RISEWU and FALLWU bits are set. This means that if these flags are set, EM4H entry will result in an immediate wake-up. To prevent this, these must be cleared by software before EM4H entry.

3.2 Calibration

VMON channels are calibrated at two voltages: 1.86 V and 2.98 V. The calibration results (coarse thresholds and fine thresholds for 1.86 V and 2.98 V) are placed in the VMONCAL registers in the DI page. Using these thresholds it is possible to calculate thresholds for the entire supported VMON VDD range, i.e., 1.62 V to 3.4 V. Using the values given in VMONCAL registers, one can calculate $T_{1.86}$, $T_{2.98}$, V_a and V_b .

Note: Calibration is automatically performed by the emlib functions `EMU_VmonInit()` and `EMU_VmonHystInit()`, so manual calibration is not needed if using these functions to initialize VMON channels.

$$T_{1.86} = (10 \times \text{VMONCALX_XVDD1V86THRESCOARSE}) + \text{VMONCALX_XVDD1V86THRESFINE}$$

$$T_{2.98} = (10 \times \text{VMONCALX_XVDD2V98THRESCOARSE}) + \text{VMONCALX_XVDD2V98THRESFINE}$$

$$V_a = 1.12 / (T_{2.98} - T_{1.86})$$

$$V_b = 1.86 - (V_a \times T_{1.86})$$

Figure 3.1. VMON Calibration Equations

Now if it is required to find the coarse and fine thresholds for a certain voltage Y, following equation can be used:

$$\text{Thres}_Y = (Y - V_b) / V_a$$

$$Y_{\text{calib}} = (\text{Thres}_Y \times V_a) + V_b$$

Figure 3.2. VMON Threshold Equations

Thres_Y should be rounded to the nearest integer. The least significant digit of the rounded Thres_Y gives the fine threshold and remaining digits give the coarse threshold for Y. These can now be programmed in the relevant `EMU_VMONXVDDCTRL` register as the coarse and fine thresholds. It may not be possible to set threshold exactly for Y. In that case the closest possible voltage is used. Y_{calib} gives the value of this closest possible voltage.

Consider the example where it is required to set the AVDD rise threshold to 2.2 V (so $Y=2.2$ V). This means that the `EMU_VMONAVDDCTRL_RISETHRESCOARSE` and `EMU_VMONAVDDCTRL_RISETHRESFINE` need to be programmed. Here are the steps that should be followed:

- Check VMONCAL0 register. It has the VMON AVDD channel calibrated thresholds for 1.86 V and 2.98 V. Lets assume that the following values are present in the associated bitfields:
 - AVDD1V86THRESCOARSE = 3
 - AVDD1V86THRESFINE = 5
 - AVDD2V98THRESCOARSE = 8
 - AVDD2V98THRESFINE = 7
- Using the above numbers and the VMON calibration equations:
 - $T_{1.86} = 35$
 - $T_{2.98} = 87$
 - $V_a = 21.53$ mV
 - $V_b = 1.106$ mV
- Using the VMON threshold equations (with $Y=2.2$ V), $\text{Thres}_Y = 51$ (rounded from 50.8) and $Y_{\text{calib}} = 2.204$ V

`EMU_VMONAVDDCTRL_RISETHRESCOARSE` should be programmed to 5 and `EMU_VMONAVDDCTRL_RISETHRESFINE` should be programmed to 1 (since $\text{Thres}_Y = 51$). With these programmed values, VMON AVDD rise threshold is set for $Y_{\text{calib}} = 2.204$ V, which is the closest programmable threshold.

4. Example 1: VMON Demo

This example demonstrates the usage of the Voltage Monitor in the EMU on Series 1 devices.

The VMON is enabled and monitors AVDD, DVDD, and IOVDD rails and displays their status to the STK's memory LCD. The thresholds for each rail, in millivolts, are set by the #defines at the top of the example vmon_demo.c file:

- VMON_AVDD_RISE_THRESHOLD_mV
- VMON_AVDD_FALL_THRESHOLD_mV
- VMON_DVDD_THRESHOLD_mV
- VMON_IOVDD_THRESHOLD_mV

Note: Only the AVDD rail has separate rising and falling thresholds.

When a voltage threshold is crossed, the EMU generates an interrupt which is handled by the EMU_IRQHandler function. This sets flags to indicate whether a voltage rail is 'good' (above its threshold) or 'bad' (below its threshold). In the main while loop, this status is then printed to the LCD, and the device enters EM3 to wait for the next interrupt.

To test:

1. Place the STK power switch into BAT to disconnect the debugger's power supply.
2. Attach a power supply to GND and VMCU (pins 1 and 2 on the expansion header)
3. Vary the power supply voltage above/below the defined thresholds. The device will update the status on the LCD when a voltage crosses a threshold.

Note: When applying voltages independently to supply rails, please observe the operating conditions in the device's datasheet, which maintain the following requirements:

- VREGVDD must be the highest voltage in the system
- VREGVDD = AVDD
- DVDD \leq AVDD
- IOVDD \leq AVDD

5. Revision History

Revision 1.09

November, 2017

- Split appnote into Series 0 and Series 1 versions.
- Updated to new appnote format.

Revision 1.08

May, 2014

- Updated example code to CMSIS 3.20.5.
- Changed to Silicon Labs license on code examples.
- Added example projects for Simplicity IDE.
- Removed example makefiles for Sourcery CodeBench Lite.
- Updated project files with shorter filenames.

Revision 1.07

October, 2013

- New cover layout.

Revision 1.06

May, 2013

- Added software projects for ARM-GCC and Atollic TrueStudio.

Revision 1.05

November, 2012

- Adapted software projects to new kit-driver and bsp structure.
- Added software support for Tiny and Giant Gecko STKs.

Revision 1.04

April, 2012

- Adapted software projects to new peripheral library naming and CMSIS_V3.

Revision 1.03

October, 2011

- Updated IDE project paths with new kits directory.

Revision 1.02

May, 2011

- Updated projects to align with new bsp version.

Revision 1.01

November, 2010

- Changed example folder structure, removed build and src folders.
- Added chip-init function.
- Updated register defines in code to match newest efm32lib release.

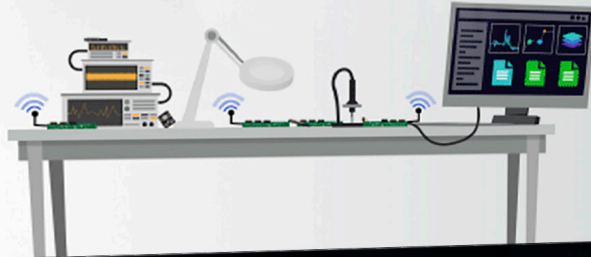
Revision 1.00

September, 2010

- Initial revision.

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



SW/HW
www.silabs.com/simplicity



Quality
www.silabs.com/quality



Support and Community
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®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri 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>