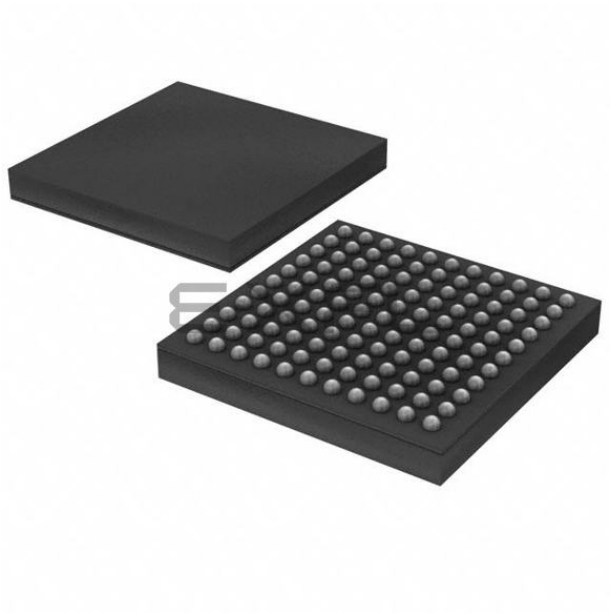


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	EBI/EMI, I²C, IrDA, SmartCard, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	93
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.98V ~ 3.8V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	120-VFBGA
Supplier Device Package	120-BGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/efm32gg995f512g-e-bga120

- **Timers/Counters**
  - 4× 16-bit Timer/Counter
    - 3 Compare/Capture/PWM channels
    - Dead-Time Insertion on TIMER0
  - 16-bit Low Energy Timer
  - 1× 24-bit and 1× 32-bit Real-Time Counter
  - 3× 8/16-bit Pulse Counter
    - Asynchronous pulse counting/quadrature decoding
  - Watchdog Timer with dedicated RC oscillator @ 50 nA
- **Backup Power Domain**
  - RTC and retention registers in a separate power domain, available in all energy modes
  - Operation from backup battery when main power drains out
- **Ultra low power precision analog peripherals**
  - 12-bit 1 Msamples/s Analog to Digital Converter
    - 8 input channels and on-chip temperature sensor
    - Single ended or differential operation
    - Conversion tailgating for predictable latency
  - 12-bit 500 ksamples/s Digital to Analog Converter
    - 2 single ended channels/1 differential channel
  - Up to 3 Operational Amplifiers
    - Supports rail-to-rail inputs and outputs
    - Programmable gain
  - 2× Analog Comparator
    - Programmable speed/current
    - Capacitive sensing with up to 8 inputs
  - Supply Voltage Comparator
- **Ultra low power sensor interface**
  - Autonomous sensor monitoring in Deep Sleep Mode
  - Wide range of sensors supported, including LC sensors and capacitive buttons

3.3.2 System Features

- **Ultra efficient Power-on Reset and Brown-Out Detector**
- **Debug Interface**
  - 2-pin Serial Wire Debug interface
    - 1-pin Serial Wire Viewer
  - Embedded Trace Module v3.5 (ETM)
- **Temperature range -40 - 85°C**
- **Single power supply 1.98 - 3.8 V**
- **Packages**
  - QFN64
  - TQFP64
  - LQFP100
  - LFBGA112
  - VFBGA120
  - Full wafer

3.4 Energy Modes

There are five different Energy Modes (EM0-EM4) in the EFM32GG, see Table 3.1 (p. 8). The EFM32GG is designed to achieve a high degree of autonomous operation in low energy modes. The intelligent combination of peripherals, RAM with data retention, DMA, low-power oscillators, and short

- Minimum 20 000 erase cycles
- More than 10 years data retention at 85°C
- Lock-bits for memory protection
- Data retention in any state

5.5 SRAM

The primary task of the SRAM memory is to store application data. Additionally, it is possible to execute instructions from SRAM, and the DMA may used to transfer data between the SRAM, Flash and peripherals.

- Up to 128 kB memory
- Bit-band access support
- 32 kB blocks may be individually powered down when not in use
- Data retention of the entire memory in EM0 to EM3

5.6 Device Information (DI) Page

The DI page contains calibration values, a unique identification number and other useful data. See the table below for a complete overview.

Table 5.4. Device Information Page Contents

DI Address	Register	Description
0x0FE08020	CMU_LFRCOCTRL	Register reset value.
0x0FE08028	CMU_HFRCOCTRL	Register reset value.
0x0FE08030	CMU_AUXHFRCOCTRL	Register reset value.
0x0FE08040	ADC0_CAL	Register reset value.
0x0FE08048	ADC0_BIASPROG	Register reset value.
0x0FE08050	DAC0_CAL	Register reset value.
0x0FE08058	DAC0_BIASPROG	Register reset value.
0x0FE08060	ACMP0_CTRL	Register reset value.
0x0FE08068	ACMP1_CTRL	Register reset value.
0x0FE08078	CMU_LCDCTRL	Register reset value.
0x0FE080A0	DAC0_OPACTRL	Register reset value.
0x0FE080A8	DAC0_OPAOFFSET	Register reset value.
0x0FE080B0	EMU_BUINACT	Register reset value.
0x0FE080B8	EMU_BUACT	Register reset value.
0x0FE080C0	EMU_BUBODBUVINCAL	Register reset value.
0x0FE080C8	EMU_BUBODUNREGCAL	Register reset value.
0x0FE081B0	DI_CRC	[15:0]: DI data CRC-16.
0x0FE081B2	CAL_TEMP_0	[7:0] Calibration temperature (°C).
0x0FE081B4	ADC0_CAL_1V25	[14:8]: Gain for 1V25 reference, [6:0]: Offset for 1V25 reference.
0x0FE081B6	ADC0_CAL_2V5	[14:8]: Gain for 2V5 reference, [6:0]: Offset for 2V5 reference.

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
16	PERIOD	0	RW	**Sets the timebase period**

Decides whether TIMEBASE specifies the number of AUX cycles in 1 us or 5 us. 5 us should only be used with 1 MHz AUXHFRCO band.

Value	Mode	Description
0	1US	TIMEBASE period is 1 us.
1	5US	TIMEBASE period is 5 us.

15:6	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
5:0	BASE	0x10	RW	**Timebase used by MSC to time flash writes and erases**

Should be set to the number of full AUX clock cycles in the period given by MSC_TIMEBASE_PERIOD. I.e. 1.1 us or 5.5. us with PERIOD cleared or set, respectively. The resetvalue of the timebase matches a 14 MHz AUXHFRCO, which is the default frequency of the AUXHFRCO.

7.5.17 MSC_MASSLOCK - Mass Erase Lock Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0001															
Access																	RW															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
15:0	LOCKKEY	0x0001	RW	**Mass Erase Lock**

Write any other value than the unlock code to lock access the the ERASEMAIN0 and ERASEMAIN1 commands. Write the unlock code 631A to enable access. When reading the register, bit 0 is set when the lock is enabled. Locked by default.

Mode	Value	Description
Read Operation		
UNLOCKED	0	Mass erase unlocked.
LOCKED	1	Mass erase locked.
Write Operation		
LOCK	0	Lock mass erase.
UNLOCK	0x631A	Unlock mass erase.

Bit	Name	Reset	Access	Description
5	CHPROT	0	W	**Channel Protection Control**
	Control whether accesses done by the DMA controller are privileged or not. When CHPROT = 1 then HPROT is HIGH and the access is privileged. When CHPROT = 0 then HPROT is LOW and the access is non-privileged.			
4:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	EN	0	W	**Enable DMA**
	Set this bit to enable the DMA controller.			

8.7.3 DMA_CTRLBASE - Channel Control Data Base Pointer Register

Offset	Bit Position																														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Access	RW																															
Name	CTRLBASE																															

Bit	Name	Reset	Access	Description
31:0	CTRLBASE	0x00000000	RW	**Channel Control Data Base Pointer**
	The base pointer for a location in system memory that holds the channel control data structure. This register must be written to point to a location in system memory with the channel control data structure before the DMA can be used. Note that ctrl_base_ptr[8:0] must be 0.			

8.7.4 DMA_ALTCTRLBASE - Channel Alternate Control Data Base Pointer Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000100																															
Access	R																															
Name	ALTCTRLBASE																															

Bit	Name	Reset	Access	Description
31:0	ALTCTRLBASE	0x00000100	R	**Channel Alternate Control Data Base Pointer**

8.7.10 DMA_CHREQMASKC - Channel Request Mask Clear Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0	0	0	0	0	0	0	0	0	0	0	0
Access																					W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
Name																					CH11REQMASKC	CH10REQMASKC	CH9REQMASKC	CH8REQMASKC	CH7REQMASKC	CH6REQMASKC	CH5REQMASKC	CH4REQMASKC	CH3REQMASKC	CH2REQMASKC	CH1REQMASKC	CH0REQMASKC

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
11	CH11REQMASKC	0	W1	**Channel 11 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
10	CH10REQMASKC	0	W1	**Channel 10 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
9	CH9REQMASKC	0	W1	**Channel 9 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
8	CH8REQMASKC	0	W1	**Channel 8 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
7	CH7REQMASKC	0	W1	**Channel 7 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
6	CH6REQMASKC	0	W1	**Channel 6 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
5	CH5REQMASKC	0	W1	**Channel 5 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
4	CH4REQMASKC	0	W1	**Channel 4 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
3	CH3REQMASKC	0	W1	**Channel 3 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
2	CH2REQMASKC	0	W1	**Channel 2 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
1	CH1REQMASKC	0	W1	**Channel 1 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			
0	CH0REQMASKC	0	W1	**Channel 0 Request Mask Clear**
	Write to 1 to enable peripheral requests for this channel.			

Bit	Name	Reset	Access	Description
	Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.			
1	CH1PRIS	0	RW1	**Channel 1 High Priority Set**
	Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.			
0	CH0PRIS	0	RW1	**Channel 0 High Priority Set**
	Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.			

8.7.16 DMA_CHPRIC - Channel Priority Clear Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0	0	0	0	0	0	0	0	0	0	0	0
Access																					W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
Name																					CH11PRIC	CH10PRIC	CH9PRIC	CH8PRIC	CH7PRIC	CH6PRIC	CH5PRIC	CH4PRIC	CH3PRIC	CH2PRIC	CH1PRIC	CH0PRIC

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
11	CH11PRIC	0	W1	**Channel 11 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
10	CH10PRIC	0	W1	**Channel 10 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
9	CH9PRIC	0	W1	**Channel 9 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
8	CH8PRIC	0	W1	**Channel 8 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
7	CH7PRIC	0	W1	**Channel 7 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
6	CH6PRIC	0	W1	**Channel 6 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
5	CH5PRIC	0	W1	**Channel 5 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
4	CH4PRIC	0	W1	**Channel 4 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
3	CH3PRIC	0	W1	**Channel 3 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
2	CH2PRIC	0	W1	**Channel 2 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
1	CH1PRIC	0	W1	**Channel 1 High Priority Clear**
	Write to 1 to clear high priority for this channel.			
0	CH0PRIC	0	W1	**Channel 0 High Priority Clear**
	Write to 1 to clear high priority for this channel.			

Bit	Name	Reset	Access	Description

Value	Mode	Description
6	DIV64	Voltage Boost update Frequency = LFACLK/64.
7	DIV128	Voltage Boost update Frequency = LFACLK/128.

3	VBOOSTEN	0	RW	**Voltage Boost Enable**
	This bit enables/disables the VBOOST function.			
2:0	FDIV	0x0	RW	**Frame Rate Control**
	These bits controls the framerate according to this formula: $LFACLK_{LCD} = LFACLK_{LCDpre} / (1 + FDIV)$. Do not change this value while the LCD bit in CMU_LFACLKEN0 is set to 1.			

11.5.27 CMU_ROUTE - I/O Routing Register

Offset	Bit Position																														
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		0	0
Access																													RW		RW	RW
Name																													LOCATION		CLKOUT1PEN	CLKOUT0PEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
4:2	LOCATION	0x0	RW	**I/O Location**
	Decides the location of the CMU I/O pins.			

Value	Mode	Description
0	LOC0	Location 0
1	LOC1	Location 1
2	LOC2	Location 2

1	CLKOUT1PEN	0	RW	**CLKOUT1 Pin Enable**
	When set, the CLKOUT1 pin is enabled.			
0	CLKOUT0PEN	0	RW	**CLKOUT0 Pin Enable**
	When set, the CLKOUT0 pin is enabled.			

11.5.28 CMU_LOCK - Configuration Lock Register

Offset	Bit Position																														
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0000															
Access																	RW															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		

12.3.1 Clock Source

Three clock sources are available for use with the watchdog, through the CLKSEL field in WDOG_CTRL. The corresponding clocks must be enabled in the CMU. The SWOSCBLOCK bit in WDOG_CTRL can be written to prevent accidental disabling of the selected clocks. Also, setting this bit will automatically start the selected oscillator source when the watchdog is enabled. The PERSEL field in WDOG_CTRL is used to divide the selected watchdog clock, and the timeout for the watchdog timer can be calculated like this:

WDOG Timeout Equation

$$T_{TIMEOUT} = (2^{3+PERSEL} + 1)/f, \quad (12.1)$$

where f is the frequency of the selected clock.

It is recommended to clear the watchdog first, if PERSEL is changed while the watchdog is enabled.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

Note

Before changing the clock source for WDOG, the EN bit in WDOG_CTRL should be cleared. In addition to this, the WDOG_SYNCBUSY value should be zero.

12.3.2 Debug Functionality

The watchdog timer can either keep running or be frozen when the device is halted by a debugger. This configuration is done through the DEBUGRUN bit in WDOG_CTRL. When code execution is resumed, the watchdog will continue counting where it left off.

12.3.3 Energy Mode Handling

The watchdog timer can be configured to either keep on running or freeze when entering EM2 or EM3. The configuration is done individually for each energy mode in the EM2RUN and EM3RUN bits in WDOG_CTRL. When the watchdog has been frozen and is re-entering an energy mode where it is running, the watchdog timer will continue counting where it left off. For the watchdog there is no difference between EM0 and EM1. The watchdog does not run in EM4, and if EM4BLOCK in WDOG_CTRL is set, the CPU is prevented from entering EM4.

Note

If the WDOG is clocked by the LFXO or LFRCO, writing the SWOSCBLOCK bit will effectively prevent the CPU from entering EM3. When running from the ULFRCO, writing the SWOSCBLOCK bit will prevent the CPU from entering EM4.

12.3.4 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3 (p. 20) for a description on how to perform register accesses to Low Energy Peripherals. note that clearing the EN bit in WDOG_CTRL will reset the WDOG module, which will halt any ongoing register synchronization.

Note

Never write to the WDOG registers when it is disabled, except to enable it by setting WDOG_CTRL_EN or when changing the clock source using WDOG_CTRL_CLKSEL. Make sure that the enable is registered (i.e. WDOG_SYNCBUSY_CTRL goes low), before writing other registers.

Bit	Name	Reset	Access	Description
	Indicates that EBI_TFTPIXEL is full.			
9	TFTPIXEL1EMPTY	0	R	**EBI_TFTPIXEL1 is empty.**
	Indicates that EBI_TFTPIXEL1 is empty.			
8	TFTPIXEL0EMPTY	0	R	**EBI_TFTPIXEL0 is empty.**
	Indicates that EBI_TFTPIXEL0 is empty.			
7:5	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
4	ECCACT	0	R	**EBI ECC Generation Active.**
	Indicates that EBI is generating ECC.			
3:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	AHBACT	0	R	**EBI Busy with AHB Transaction.**
	Indicates that EBI is busy with an AHB Transaction.			

14.5.23 EBI_ECCPARITY - ECC Parity register

Offset	Bit Position																														
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Access	R																															
Name	ECCPARITY																															

Bit	Name	Reset	Access	Description
31:0	ECCPARITY	0x00000000	R	**ECC Parity Data**
	ECC Parity Data.			

14.5.24 EBI_TFTCTRL - TFT Control Register

Offset	Bit Position																														
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0			0x0					0				0	0x0		0	0				0x0			0x0	
Access								RW			RW					RW				RW	RW		RW	RW				RW			RW	
Name								RGBMODE			BANKSEL					WIDTH				COLOR1SRC	INTERLEAVE		FBCTRIG	SHIFTDCLKEN				MASKBLEND			DD	

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
24	RGBMODE	0	RW	**TFT RGB Mode**
	This field sets TFT RGB Mode.			

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	RGB565	RGB data is 565.
1	RGB555	RGB data is 555.

23:22	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
21:20	BANKSEL	0x0	RW	**Graphics Bank**

This field sets the Memory Bank containing the Frame Buffer

Value	Mode	Description
0	BANK0	Memory bank 0 is used for Direct Drive, Masking, and Alpha Blending.
1	BANK1	Memory bank 1 is used for Direct Drive, Masking, and Alpha Blending.
2	BANK2	Memory bank 2 is used for Direct Drive, Masking, and Alpha Blending.
3	BANK3	Memory bank 3 is used for Direct Drive, Masking, and Alpha Blending.

19:17	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
16	WIDTH	0	RW	**TFT Transaction Width**

This field sets TFT tranaction width.

Value	Mode	Description
0	BYTE	TFT Data is 8 bit wide.
1	HALFWORD	TFT Data is 16 bit wide.

15:13	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
12	COLOR1SRC	0	RW	**Masking/Alpha Blending Color1 Source**

This field sets the Masking/Alpha Blending Color1 Source.

Value	Mode	Description
0	MEM	Masking/Alpha Blending color 1 is read from external memory.
1	PIXEL1	Masking/Alpha Blending color 1 is read from EBI_TFTPIXEL1.

11:10	INTERLEAVE	0x0	RW	**Interleave Mode**

This field sets the TFT Direct Drive Interleave mode.

Value	Mode	Description
0	UNLIMITED	Allow unlimited interleaved EBI accesses per EBI_DCLK period. This can cause jitter on the EBI_DCLK
1	ONEPERDCLK	Allow 1 interleaved EBI access per EBI_DCLK period.
2	PORCH	Only allow EBI accesses during TFT porches.

9	FBCTRIG	0	RW	**TFT Frame Base Copy Trigger**

Sets the trigger on which the TFTFRAMEBASE is copied into an internal buffer. Direct Drive address generation is based on the internal buffer.

Value	Mode	Description
0	VSYNC	TFTFRAMEBASE is buffered on the vertical synchronization event EBI_VSYNC.
1	HSYNC	TFTFRAMEBASE is buffered on the horizontal synchronization event EBI_HSYNC.

8	SHIFTDCLKEN	0	RW	**TFT EBI_DCLK Shift Enable**

When this bit is set, EBI_DCLK edges are driven off the negative (instead of the positive) edge of the internal clock. SHIFTDCLKEN is only allowed to be set to 1 if TFTHOLD in EBI_TFTTIMING is at least 1.

7:5	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
4:2	MASKBLEND	0x0	RW	**TFT Mask and Blend Mode**

This field sets the Mask and Blend Mode.

Value	Mode	Description
0	DISABLED	Masking and Blending are disabled.
1	IMASK	Internal Masking is enabled.
2	IALPHA	Internal Alpha Blending is enabled.
3	IMASKIALPHA	Internal Masking and Alpha Blending are enabled.
5	EMASK	External Masking is enabled.
6	EALPHA	External Alpha Blending is enabled.

6. The application must service the Session Request Detected interrupt and turn on the Port Power bit by writing the Port Power bit in the Host Port Control and Status register. The PHY indicates port power-on by detecting a valid VBUS level.
7. When the USB is powered, the device connects, completing the SRP process.

15.4.5.2 B-Device Session Request Protocol

The application must set the SRP-Capable bit in the Core USB Configuration register. This enables the core to initiate SRP as a B-device. SRP is a means by which the core can request a new session from the host.

1. To save power, the host suspends and turns off port power when the bus is idle. PHY indicates port power off by detecting a not valid VBUS level.

   The core sets the Early Suspend bit in the Core Interrupt register after 3 ms of bus idleness. Following this, the core sets the USB Suspend bit in the Core Interrupt register.

   The PHY indicates the end of the B-device session by detecting a VBUS level below session valid.
2. PHY to enables the VBUS discharge function to speed up Vbus discharge.
3. The PHY indicates the session's end by detecting a session end voltage level on VBUS. This is the initial condition for SRP. The core requires 2 ms of SE0 before initiating SRP.

   The application must wait until Vbus discharges to 0.2 V after USB_GOTGCTL.BSESVLD is deasserted. This discharge time can be obtained from the datasheet.
4. The application initiates SRP by writing the Session Request bit in the OTG Control and Status register. The core perform data-line pulsing followed by Vbus pulsing.
5. The host detects SRP from either the data-line or Vbus pulsing, and turns on Vbus. The PHY indicates Vbus power-on by detecting a valid VBUS level.
6. The core performs Vbus pulsing.

   The host starts a new session by turning on Vbus, indicating SRP success. The core interrupts the application by setting the Session Request Success Status Change bit in the OTG Interrupt Status register. The application reads the Session Request Success bit in the OTG Control and Status register.
7. When the USB is powered, the core connects, completing the SRP process.

15.4.5.3 A-Device Host Negotiation Protocol

HNP switches the USB host role from the A-device to the B-device. The application must set the HNP-Capable bit in the Core USB Configuration register to enable the core to perform HNP as an A#device.

1. The core sends the B-device a SetFeature b_hnp_enable descriptor to enable HNP support. The B-device's ACK response indicates that the B-device supports HNP. The application must set Host Set HNP Enable bit in the OTG Control and Status register to indicate to the core that the B-device supports HNP.
2. When it has finished using the bus, the application suspends by writing the Port Suspend bit in the Host Port Control and Status register.
3. When the B-device observes a USB suspend, it disconnects, indicating the initial condition for HNP. The B-device initiates HNP only when it must switch to the host role; otherwise, the bus continues to be suspended.

   The core sets the Host Negotiation Detected interrupt in the OTG Interrupt Status register, indicating the start of HNP.

   The PHY turns off the D+ and D- pulldown resistors to indicate a device role. The PHY enable the D + pull-up resistor indicates a connect for B-device.

Table 16.3. I²C Clock Mode

HFPERCLK frequency (MHz)	Clock Low High Ratio (CLHR)	Sm max frequency (kHz)	Fm max frequency (kHz)	Fm+ max frequency (kHz)
48	0	92	400	1000
	1	82	400	1000
	2	72	400	842
28	0	92	400	1000
	1	81	400	848
	2	71	400	736
21	0	90	400	1000
	1	80	400	954
	2	72	368	552
14	0	92	400	1000
	1	81	400	636
	2	68	368	608
11	0	91	400	785
	1	81	333	733
	2	71	289	478
6.6	0	91	400	471
	1	81	299	439
	2	64	286	286
1.2	0	59	85	85
	1	54	79	79
	2	52	52	52

16.3.5 Arbitration

Arbitration is enabled by default, but can be disabled by setting the ARBDIS bit in I2Cn_CTRL. When arbitration is enabled, the value on SDA is sensed every time the I²C module attempts to change its value. If the sensed value is different than the value the I²C module tried to output, it is interpreted as a simultaneous transmission by another device, and that the I²C module has lost arbitration.

Whenever arbitration is lost, the ARBLOST interrupt flag in I2Cn_IF is set, any lines held are released, and the I²C device goes idle. If an I²C master loses arbitration during the transmission of an address, another master may be trying to address it. The master therefore receives the rest of the address, and if the address matches the slave address of the master, the master goes into either slave transmitter or slave receiver mode.

Note

Arbitration can be lost both when operating as a master and when operating as a slave.

16.3.6 Buffers

16.3.6.1 Transmit Buffer and Shift Register

The I²C transmitter is double buffered through the transmit buffer and transmit shift register as shown in Figure 16.1 (p. 416) . A byte is loaded into the transmit buffer by writing to I2Cn_TXDATA. When the

See Table 16.10 (p. 433) for more information.

Table 16.10. I²C - Slave Receiver

I2Cn_STAT	Description	I2Cn_IF	Required interaction	Response
-	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x71	ADDR + W received	ADDR interrupt flag RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent, slave goes idle
			NACK + CONT + RXDATA	NACK will be sent and DATA will be received.
0xB1	Data received	RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent and slave will go idle
			NACK + CONT + RXDATA	NACK will be sent and data will be received
-	Stop received	SSTOP interrupt flag	None	The slave goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	The slave goes idle
			START	START will be sent when the bus becomes idle

16.3.10 Transfer Automation

The I²C can be set up to complete transfers with a minimal amount of interaction.

16.3.10.1 DMA

DMA can be used to automatically load data into the transmit buffer and load data out from the receive buffer. When using DMA, software is thus relieved of moving data to and from memory after each transferred byte.

16.3.10.2 Automatic ACK

When AUTOACK in I2Cn_CTRL is set, an ACK is sent automatically whenever an ACK interaction is possible and no higher priority interactions are pending.

16.3.10.3 Automatic STOP

A STOP can be generated automatically on two conditions. These apply only to the master transmitter.

If AUTOSN in I2Cn_CTRL is set, the I²C module ends a transmission by transmitting a STOP condition when operating as a master transmitter and a NACK is received.

If AUTOSE in I2Cn_CTRL is set, the I²C module always ends a transmission when there is no more data in the transmit buffer. If data has been transmitted on the bus, the transmission is ended after the (N)ACK has been received by the slave. If a START is sent when no data is available in the transmit buffer and AUTOSE is set, then the STOP condition is sent immediately following the START. Software must thus make sure data is available in the transmit buffer before the START condition has been fully transmitted if data is to be transferred.

  - Communication debugging
- PRS can trigger transmissions
- Full DMA support
- PRS RX input

18.3 Functional Description

The UART is functionally equivalent to the USART with the exceptions defined in Table 18.1 (p. 496) . The register map and register descriptions are equal to those of the USART. See the USART chapter for detailed information on the operation of the UART.

Table 18.1. UART Limitations

Feature	Limitations
Synchronous operation	Not available. SYNC, CSMA, CSINV, CPOL and CPHA in USARTn_CTRL, and MASTEREN in USARTn_STATUS are always 0.
Transmission direction	Always LSB first. MSBF in USARTn_CTRL is always 0.
Chip-select	Not available. AUTOCS in USARTn_CTRL is always 0.
SmartCard mode	Not available. SCMODE in USARTn_CTRL is always 0.
Frame size	Limited to 8-9 databits. Other configurations of DATABITS in USARTn_FRAME are not possible.
IrDA	Not available. IREN in USARTn_IRCTRL is always 0.

18.4 Register Description

The register description of the UART is equivalent to the register description of the USART except the limitations mentioned in Table 18.1 (p. 496) . See the USART chapter for complete information.

18.5 Register Map

The register map of the UART is equivalent to the register map of the USART. See the USART chapter for complete information.

When 8 data-bit frame formats are used, only the 8 least significant bits of LEUARTn_STARTFRAME are compared to incoming frames. The full length of LEUARTn_STARTFRAME is used when operating with frames consisting of 9 data bits.

Note

The receiver must be enabled for start frames to be detected. In addition, a start frame with a parity error or framing error is not detected as a start frame.

19.3.5.7 Programmable Signal Frame

As well as the configurable start frame, a special signal frame can be specified. When a frame matching the frame defined in LEUARTn_SIGFRAME is detected by the receiver, the SIGF interrupt flag in LEUARTn_IF is set. As for start frame detection, the receiver must be enabled for signal frames to be detected.

One use of the programmable signal frame is to signal the end of a multi-frame message transmitted to the LEUART. An interrupt will then be triggered when the packet has been completely received, allowing software to process it. Used in conjunction with the programmable start frame and DMA, this makes it possible for the LEUART to automatically begin the reception of a packet on a specified start frame, load the entire packet into memory, and give an interrupt when reception of a packet has completed. The device can thus wait for data packets in EM2, and only be woken up when a packet has been completely received.

A signal frame with a parity error or framing error is not detected as a signal frame.

19.3.5.8 Multi-Processor Mode

To simplify communication between multiple processors and maintain compatibility with the USART, the LEUART supports a multi-processor mode. In this mode the 9th data bit in each frame is used to indicate whether the content of the remaining 8 bits is data or an address.

When multi-processor mode is enabled, an incoming 9-bit frame with the 9th bit equal to the value of MPAB in LEUARTn_CTRL is identified as an address frame. When an address frame is detected, the MPAF interrupt flag in LEUARTn_IF is set, and the address frame is loaded into the receive register. This happens regardless of the value of RXBLOCK in LEUARTn_STATUS.

Multi-processor mode is enabled by setting MPM in LEUARTn_CTRL. The mode can be used in buses with multiple slaves, allowing the slaves to be addressed using the special address frames. An addressed slave, which was previously blocking reception using RXBLOCK, would then unblock reception, receive a message from the bus master, and then block reception again, waiting for the next message. See the USART for a more detailed example.

Note

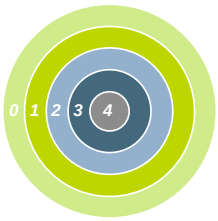
The programmable start frame functionality can be used for automatic address matching, enabling reception on a correctly configured incoming frame.

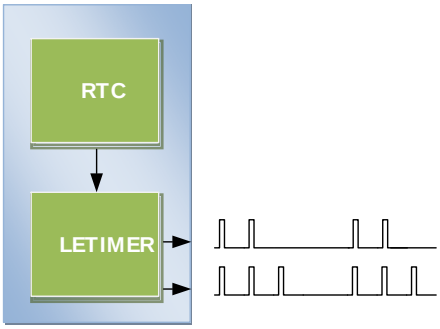
An address frame with a parity error or a framing error is not detected as an address frame.

19.3.6 Loopback

The LEUART receiver samples LEUn_RX by default, and the transmitter drives LEUn_TX by default. This is not the only configuration however. When LOOPBK in LEUARTn_CTRL is set, the receiver is connected to the LEUn_TX pin as shown in Figure 19.5 (p. 506) . This is useful for debugging, as the LEUART can receive the data it transmits, but it is also used to allow the LEUART to read and write to the same pin, which is required for some half duplex communication modes. In this mode, the LEUn_TX pin must be enabled as an output in the GPIO.

23 LETIMER - Low Energy Timer





Quick Facts

What?

The LETIMER is a down-counter that can keep track of time and output configurable waveforms. Running on a 32.768 Hz clock the LETIMER is available in EM2, while using a 1 kHz clock the LETIMER is available also in EM3, all this with sub µA current consumption.

Why?

The LETIMER can be used to provide repeatable waveforms to external components while remaining in EM2. It is well suited for e.g. metering systems or to provide more compare values than available in the RTC.

How?

With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz.
It is tightly coupled to the RTC, which allows advanced time-keeping and wake-up functions in EM2 and EM3.

23.1 Introduction

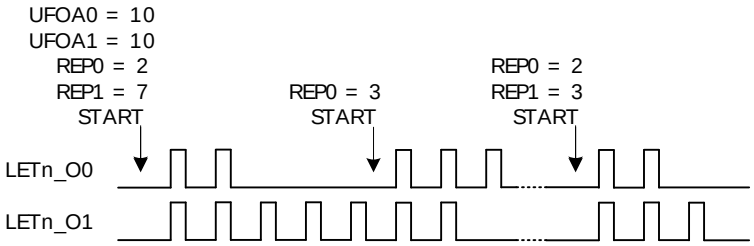
The unique LETIMER™, the Low Energy Timer, is a 16-bit timer that is available in energy mode EM2 and EM3, in addition to EM1 and EM0. Because of this, it can be used for timing and output generation when most of the device is powered down, allowing simple tasks to be performed while the power consumption of the system is kept at an absolute minimum.

The LETIMER can be used to output a variety of waveforms with minimal software intervention. It is also connected to the Real Time Counter (RTC), and can be configured to start counting on compare matches from the RTC.

23.2 Features

- 16-bit down count timer
- 2 Compare match registers
- Compare register 0 can be top timer top value
- Compare registers can be double buffered
- Double buffered 8-bit Repeat Register
- Same clock source as the Real Time Counter
- LETIMER can be triggered (started) by an RTC event or by software
- 2 output pins can optionally be configured to provide different waveforms on timer underflow:
  - Toggle output pin
  - Apply a positive pulse (pulse width of one $LFACLK_{LETIMER}$ period)
  - PWM

Figure 23.8. LETIMER Dual Output



23.3.5 PRS Output

The LETIMER outputs can be routed out onto the PRS system. LETn_O0 can be routed to PRS channel 0, and LETn_1O can be routed to PRS channel 1. Enabling the RRS connection can be done by setting SOURCESEL to LETIMERx and SIGSEL to LETIMERxCHn in PRS_CHx_CTRL. The PRS register description can be found in Section 13.5 (p. 169)

23.3.6 Examples

This section presents a couple of usage examples for the LETIMER.

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	DISABLE	ALTEX0 output is disabled in idle phase
1	HIGH	ALTEX0 output is high in idle phase
2	LOW	ALTEX0 output is low in idle phase

25.5.17 LESENSE_IF - Interrupt Flag Register

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access										R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name										CNTOF	BUFOF	BUFLEVEL	BUFDATAV	DECERR	DEC	SCANCOMPLETE	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
22	CNTOF	0	R	
	Set when the LESENSE counter overflows.			
21	BUFOF	0	R	
	Set when the result buffer overflows			
20	BUFLEVEL	0	R	
	Set when the data buffer is full.			
19	BUFDATAV	0	R	
	Set when data is available in the result buffer.			
18	DECERR	0	R	
	Set when the decoder detects an error			
17	DEC	0	R	
	Set when the decoder has issued and interrupt request			
16	SCANCOMPLETE	0	R	
	Set when a scan sequence is completed			
15	CH15	0	R	
	Set when channel 15 triggers			
14	CH14	0	R	
	Set when channel 14 triggers			
13	CH13	0	R	
	Set when channel 13 triggers			
12	CH12	0	R	
	Set when channel 12 triggers			
11	CH11	0	R	
	Set when channel 11 triggers			
10	CH10	0	R	
	Set when channel 10 triggers			
9	CH9	0	R	
	Set when channel 9 triggers			

Bit	Name	Reset	Access	Description
8	CH8	0	R	
	Set when channel 8 triggers			
7	CH7	0	R	
	Set when channel 7 triggers			
6	CH6	0	R	
	Set when channel 6 triggers			
5	CH5	0	R	
	Set when channel 5 triggers			
4	CH4	0	R	
	Set when channel 4 triggers			
3	CH3	0	R	
	Set when channel 3 triggers			
2	CH2	0	R	
	Set when channel 2 triggers			
1	CH1	0	R	
	Set when channel 1 triggers			
0	CH0	0	R	
	Set when channel 0 triggers			

25.5.18 LESENSE_IFC - Interrupt Flag Clear Register

Offset	Bit Position																														
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access										W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
Name										CNTOF	BUFOF	BUFLEVEL	BUFDATAV	DECERR	DEC	SCANCOMPLETE	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
22	CNTOF	0	W1	
	Write to 1 to clear CNTOF interrupt flag			
21	BUFOF	0	W1	
	Write to 1 to clear BUFOF interrupt flag			
20	BUFLEVEL	0	W1	
	Write to 1 to clear BUFLEVEL interrupt flag			
19	BUFDATAV	0	W1	
	Write to 1 to clear BUFDATAV interrupt flag			
18	DECERR	0	W1	
	Write to 1 to clear DECERR interrupt flag			
17	DEC	0	W1	
	Write to 1 to clear DEC interrupt flag			
16	SCANCOMPLETE	0	W1	

Offset	Bit Position																															
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Access	RW																															
Name	SEGD3L																															

Bit	Name	Reset	Access	Description
31:0	SEGD3L	0x00000000	RW	**COM3 Segment Data Low**
	This register contains segment data for segment lines 0-31 for COM3.			

33.5.16 LCD_SEGD0H - Segment Data High Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 20) .

Offset	Bit Position																															
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x00							
Access																									RW							
Name																									SEGD0H							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
7:0	SEGD0H	0x00	RW	**COM0 Segment Data High**
	This register contains segment data for segment lines 32-39 for COM0.			

33.5.17 LCD_SEGD1H - Segment Data High Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 20) .

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x00							
Access																									RW							
Name																									SEGD1H							