Welcome to E-XFL.COM

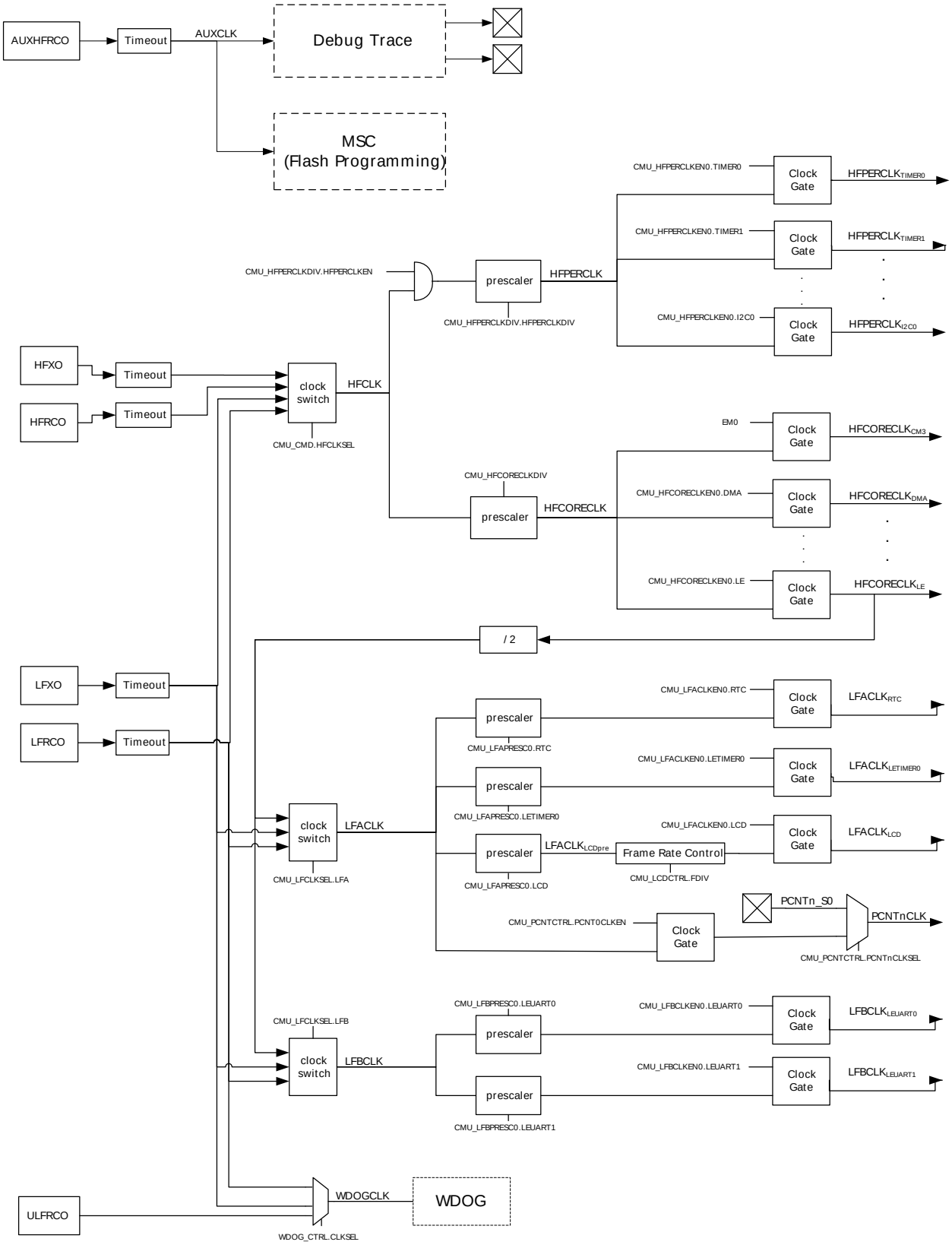
What is "Embedded - Microcontrollers"?

"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	32MHz
Connectivity	EBI/EMI, I²C, IrDA, SmartCard, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	56
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K 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	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/efm32g30f128g-e-qfn64

Figure 11.1. CMU Overview



11.3.1 System Clocks

11.3.1.1 HFCLK - High Frequency Clock

HFCLK is the selected High Frequency Clock. This clock is used by the CMU and drives the two prescalers that generate HFCORECLK and HFPERCLK. The HFCLK can be driven by a high-frequency

Bit	Name	Reset	Access	Description

Value	Mode	Description
1	PCNT0S0	External pin PCNT0_S0 is clocking PCNT0.

Bit	Name	Reset	Access	Description
0	PCNT0CLKEN	0	RW	**PCNT0 Clock Enable**

This bit enables/disables the clock to the PCNT.

Value	Description
0	PCNT0 is disabled.
1	PCNT0 is enabled.

11.5.26 CMU_LCDCTRL - LCD Control Register

Offset	Bit Position																															
0x07C	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																										0x2			0		0x0	
Access																										RW			RW		RW	
Name																										VBFDIV			VBOOSTEN		FDIV	

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
6:4	VBFDIV	0x2	RW	**Voltage Boost Frequency Division**

These bits control the voltage boost update frequency division.

Value	Mode	Description
0	DIV1	Voltage Boost update Frequency = LFACLK.
1	DIV2	Voltage Boost update Frequency = LFACLK/2.
2	DIV4	Voltage Boost update Frequency = LFACLK/4.
3	DIV8	Voltage Boost update Frequency = LFACLK/8.
4	DIV16	Voltage Boost update Frequency = LFACLK/16.
5	DIV32	Voltage Boost update Frequency = LFACLK/32.
6	DIV64	Voltage Boost update Frequency = LFACLK/64.
7	DIV128	Voltage Boost update Frequency = LFACLK/128.

Bit	Name	Reset	Access	Description
3	VBOOSTEN	0	RW	**Voltage Boost Enable**

This bit enables/disables the VBOOST function.

Bit	Name	Reset	Access	Description
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																														0	0	0
Access																														RW	RW	RW
Name																														LOCATION	CLKOUT1PEN	CLKOUT0PEN

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.

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 the EN bitfield in WDOG_CTRL. Make sure that the enable is registered (i.e. WDOG_SYNCBUSY_CTRL goes low), before writing other registers.

Bit	Name	Reset	Access	Description
7:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1:0	RDSETUP	0x3	RW	**Read Setup Time**
	Sets the number of cycles the address setup before REn is asserted.			

14.5.17 EBI_WRTIMING3 - Write Timing Register 3

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											0x3				0x3F											0x3	
Access			RW	RW											RW				RW											RW	
Name			WBUFDIS	HALFWE											WRHOLD				WRSTRB											WRSETUP	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
29	WBUFDIS	0	RW	**Write Buffer Disable**
	Enables or disables the write buffer.			
28	HALFWE	0	RW	**Half Cycle WEn Strobe Duration Enable**
	Enables or disables half cycle duration of the WEn strobe in the last WRSTRB cycle.			
27:18	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
17:16	WRHOLD	0x3	RW	**Write Hold Time**
	Sets the number of cycles CSn is held active after the WEn is deasserted.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
13:8	WRSTRB	0x3F	RW	**Write Strobe Time**
	Sets the number of cycles the WEn is held active. If set to 0, 1 cycle is inserted by HW.			
7:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1:0	WRSETUP	0x3	RW	**Write Setup Time**
	Sets the number of cycles the address setup before WEn is asserted.			

14.5.18 EBI_POLARITY3 - Polarity Register 3

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
Access																											RW	RW	RW	RW	RW	RW
Name																											BLPOL	ARDYPOL	ALEPOL	WEPOL	REPOL	CSPOL

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
5	BLPOL	0	RW	**BL Polarity**
	Sets the polarity of the EBI_BLn lines.			

14.5.25 EBI_TFTSTATUS - TFT Status Register

Offset	Bit Position																														
0x060	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						0x000																0x000										
Access						R																R										
Name						VCNT																HCNT										

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
26:16	VCNT	0x000	R	**Vertical Count**
	Contains the current line position within a frame (initial line in vertical back porch has VCNT = 0).			
15:11	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
10:0	HCNT	0x000	R	**Horizontal Count**
	Contains the current pixel position within a line (initial pixel in horizontal backporch has HCNT = 0).			

14.5.26 EBI_TFTFRAMEBASE - TFT Frame Base Register

Offset	Bit Position																														
0x064	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					0x0000000																											
Access					RW																											
Name					FRAMEBASE																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
27:0	FRAMEBASE	0x0000000	RW	**Frame Base Address**
	Sets the frame base address.			

Table 15.3. I²C Clock Mode

HFPERCLK frequency (MHz)	Clock Low High Ratio (CLHR)	Sm max frequency (kHz)	Fm max frequency (kHz)	Fm+ max frequency (kHz)
32	0	93	400	1000
	1	82	400	969
	2	72	400	842
28	0	92	400	1000
	1	81	400	848
	2	71	400	736
21	0	93	400	1000
	1	83	400	954
	2	72	368	552
14	0	92	400	999
	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

15.3.5 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.

15.3.6 Buffers

15.3.6.1 Transmit Buffer and Shift Register

The I²C transmitter is double buffered through the transmit buffer and transmit shift register as shown in Figure 15.1 (p. 174) . A byte is loaded into the transmit buffer by writing to I2Cn_TXDATA. When the

Many slave-only devices operating on an I2C-bus are not capable of driving SCL low, but in the rare case that SCL is stuck LOW, the advice is to apply a hardware reset signal to the slaves on the bus. If this does not work, cycle the power to the devices in order to make them release SCL.

When SDA is stuck low and SCL is free, a master should send 9 clock pulses on SCL while tristating the SDA. This procedure is performed in the GPIO module after clearing the I2C_ROUTE register and disabling the I2C module. The device that held the bus low should release it sometime within those 9 clocks. If not, use the same approach as for when SCL is stuck, resetting and possibly cycling power to the slaves.

Lockup of SDA can be detected by keeping count of the number of continuous arbitration losses during address transmission. If arbitration is also lost during the transmission of a general call address, i.e. during the transmission of the STOP condition, which should never happen during normal operation, this is a good indication of SDA lockup.

Detection of SCL lockups can be done using the timeout functionality defined in Section 15.3.12.6 (p. 193)

15.3.12.5 Bus Idle Timeout

When SCL has been high for a significant amount of time, this is a good indication of that the bus is idle. On an SMBus system, the bus is only allowed to be in this state for a maximum of 50 µs before the bus is considered idle.

The bus idle timeout BITO in I2Cn_CTRL can be used to detect situations where the bus goes idle in the middle of a transmission. The timeout can be configured in BITO, and when the bus has been idle for the given amount of time, the BITO interrupt flag in I2Cn_IF is set. The bus can also be set idle automatically on a bus idle timeout. This is enabled by setting GIBITO in I2Cn_CTRL.

When the bus idle timer times out, it wraps around and continues counting as long as its condition is true. If the bus is not set idle using GIBITO or the ABORT command in I2Cn_CMD, this will result in periodic timeouts.

Note

This timeout will be generated even if SDA is held low.

The bus idle timeout is active as long as the bus is busy, i.e. BUSY in I2Cn_STATUS is set. The timeout can be used to get the I2C module out of the busy-state it enters when reset, see Section 15.3.7.3 (p. 183) .

15.3.12.6 Clock Low Timeout

The clock timeout, which can be configured in CLTO in I2Cn_CTRL, starts counting whenever SCL goes low, and times out if SCL does not go high within the configured timeout. A clock low timeout results in CLTOIF in I2Cn_IF being set, allowing software to take action.

When the timer times out, it wraps around and continues counting as long as SCL is low. An SCL lockup will thus result in periodic clock low timeouts as long as SCL is low.

15.3.13 DMA Support

The I2C module has full DMA support. The DMA controller can write to the transmit buffer using the I2Cn_TXDATA register, and it can read from the receive buffer using the RXDATA register. A request for the DMA controller to read from the I2C receive buffer can come from the following source:

- Data available in the receive buffer

A write request can come from one of the following sources:

Bit	Name	Reset	Access	Description

When set, the bus automatically goes idle on a bus idle timeout, allowing new transfers to be initiated.

Value	Description
0	A bus idle timeout has no effect on the bus state.
1	A bus idle timeout tells the I^2C module that the bus is idle, allowing new transfers to be initiated.

14 Reserved — To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

13:12 BITO 0x0 RW **Bus Idle Timeout**

Use to generate a timeout when SCL has been high for a given amount time between a START and STOP condition. When in a bus transaction, i.e. the BUSY flag is set, a timer is started whenever SCL goes high. When the timer reaches the value defined by BITO, it sets the BITO interrupt flag. The BITO interrupt flag will then be set periodically as long as SCL remains high. The bus idle timeout is active as long as BUSY is set. It is thus stopped automatically on a timeout if GIBITO is set. It is also stopped a STOP condition is detected and when the ABORT command is issued. The timeout is activated whenever the bus goes BUSY, i.e. a START condition is detected.

Value	Mode	Description
0	OFF	Timeout disabled
1	40PCC	Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
2	80PCC	Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
3	160PCC	Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.

11:10 Reserved — To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

9:8 CLHR 0x0 RW **Clock Low High Ratio**

Determines the ratio between the low and high parts of the clock signal generated on SCL as master.

Value	Mode	Description
0	STANDARD	The ratio between low period and high period counters (N_{low}:N_{high}) is 4:4
1	ASYMMETRIC	The ratio between low period and high period counters (N_{low}:N_{high}) is 6:3
2	FAST	The ratio between low period and high period counters (N_{low}:N_{high}) is 11:6

7 Reserved — To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

6 GCAMEN 0 RW **General Call Address Match Enable**

Set to enable address match on general call in addition to the programmed slave address.

Value	Description
0	General call address will be NACK'ed if it is not included by the slave address and address mask.
1	When a general call address is received, a software response is required.

5 ARBDIS 0 RW **Arbitration Disable**

A master or slave will not release the bus upon losing arbitration.

Value	Description
0	When a device loses arbitration, the ARB interrupt flag is set and the bus is released.
1	When a device loses arbitration, the ARB interrupt flag is set, but communication proceeds.

4 AUTOSN 0 RW **Automatic STOP on NACK**

Write to 1 to make a master transmitter send a STOP when a NACK is received from a slave.

Value	Description
0	Stop is not automatically sent if a NACK is received from a slave.
1	The master automatically sends a STOP if a NACK is received from a slave.

3 AUTOSE 0 RW **Automatic STOP when Empty**

Write to 1 to make a master transmitter send a STOP when no more data is available for transmission.

Value	Description
0	A stop must be sent manually when no more data is to be transmitted.
1	The master automatically sends a STOP when no more data is available for transmission.

2 AUTOACK 0 RW **Automatic Acknowledge**

Set to enable automatic acknowledges.

16.5.5 USARTn_STATUS - USART Status Register

Offset	Bit Position																															
0x010	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	1	0	0	0	0	0	0
Access																								R	R	R	R	R	R	R	R	R
Name																								RXFULL	RXDATAV	TXBL	TXC	TXTRI	RXBLOCK	MASTER	TXENS	RXENS

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
8	RXFULL	0	R	**RX FIFO Full**
	Set when the RXFIFO is full. Cleared when the receive buffer is no longer full. When this bit is set, there is still room for one more frame in the receive shift register.			
7	RXDATAV	0	R	**RX Data Valid**
	Set when data is available in the receive buffer. Cleared when the receive buffer is empty.			
6	TXBL	1	R	**TX Buffer Level**
	Indicates the level of the transmit buffer. If TXBIL is cleared, TXBL is set whenever the transmit buffer is empty, and if TXBIL is set, TXBL is set whenever the transmit buffer is half-full or empty.			
5	TXC	0	R	**TX Complete**
	Set when a transmission has completed and no more data is available in the transmit buffer and shift register. Cleared when data is written to the transmit buffer.			
4	TXTRI	0	R	**Transmitter Tristated**
	Set when the transmitter is tristated, and cleared when transmitter output is enabled. If AUTOTRI in USARTn_CTRL is set this bit is always read as 0.			
3	RXBLOCK	0	R	**Block Incoming Data**
	When set, the receiver discards incoming frames. An incoming frame will not be loaded into the receive buffer if this bit is set at the instant the frame has been completely received.			
2	MASTER	0	R	**SPI Master Mode**
	Set when the USART operates as a master. Set using the MASTEREN command and clear using the MASTERDIS command.			
1	TXENS	0	R	**Transmitter Enable Status**
	Set when the transmitter is enabled.			
0	RXENS	0	R	**Receiver Enable Status**
	Set when the receiver is enabled.			

16.5.6 USARTn_CLKDIV - Clock Control Register

Offset	Bit Position																															
0x014	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												DIV																				

Bit	Name	Reset	Access	Description
13:9	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
8:0	RXDATAP	0x000	R	**RX Data Peek**
	Use this register to access data read from the LEUART.			

18.5.10 LEUARTn_TXDATAX - Transmit Buffer Data Extended Register (Async Reg)

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

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					0x000								
Access																	W	W	W					W								
Name																	RXENAT	TXDISAT	TXBREAK					TXDATA								

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	RXENAT	0	W	**Enable RX After Transmission**
	Set to enable reception after transmission.			

Value	Description
0	-
1	The receiver is enabled, setting RXENS after the frame has been transmitted

Bit	Name	Reset	Access	Description
14	TXDISAT	0	W	**Disable TX After Transmission**
	Set to disable transmitter directly after transmission has competed.			

Value	Description
0	-
1	The transmitter is disabled, clearing TXENS after the frame has been transmitted

Bit	Name	Reset	Access	Description
13	TXBREAK	0	W	**Transmit Data As Break**
	Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of TXDATA.			

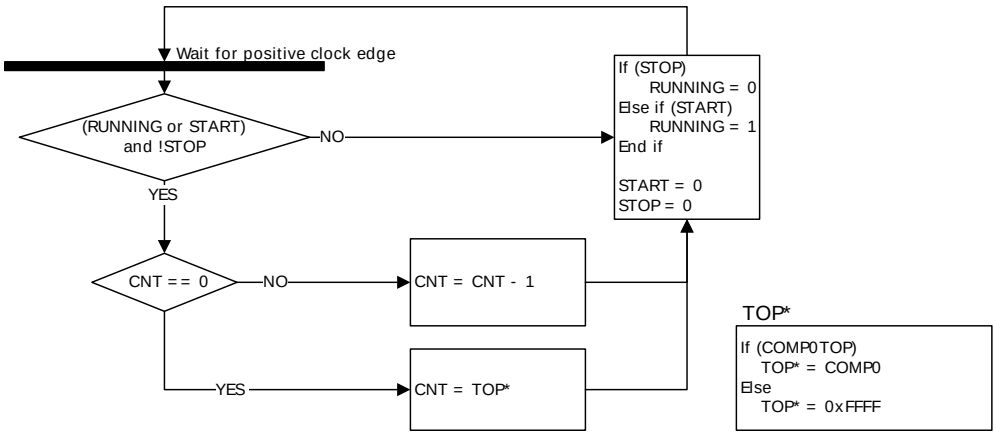
Value	Description
0	The specified number of stop-bits are transmitted
1	Instead of the ordinary stop-bits, 0 is transmitted to generate a break. A single stop-bit is generated after the break to allow the receiver to detect the start of the next frame

Bit	Name	Reset	Access	Description
12:9	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
8:0	TXDATA	0x000	W	**TX Data**
	Use this register to write data to the LEUART. If the transmitter is enabled, a transfer will be initiated at the first opportunity.			

18.5.11 LEUARTn_TXDATA - Transmit Buffer Data Register (Async Reg)

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

Figure 21.2. LETIMER State Machine for Free-running Mode



Note that the CLEAR command bit in LETIMERn_CMD always has priority over other changes to LETIMERn_CNT. When the clear command is used, LETIMERn_CNT is set to 0 and an underflow event will not be generated when LETIMERn_CNT wraps around to the top value or 0xFFFF. Since no underflow event is generated, no output action is performed. LETIMERn_REP0, LETIMERn_REP1, LETIMERn_COMP0 and LETIMERn_COMP1 are also left untouched.

21.3.3.2.2 One-shot Mode

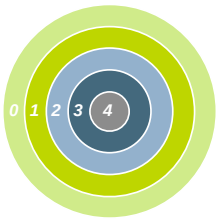
The one-shot repeat mode is the most basic repeat mode. In this mode, the repeat register LETIMERn_REP0 is decremented every time the timer underflows, and the timer stops when LETIMERn_REP0 goes from 1 to 0. In this mode, the timer counts down LETIMERn_REP0 times, i.e. the timer underflows LETIMERn_REP0 times.

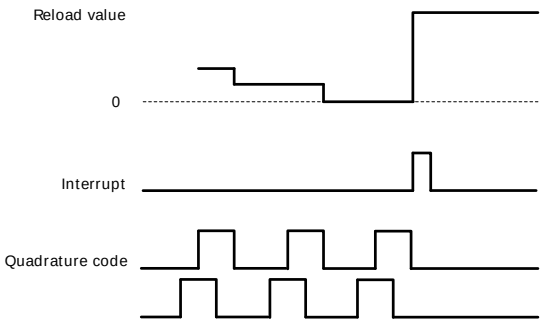
Note

Note that write operations to LETIMERn_REP0 have priority over the decrementation operation. So if LETIMERn_REP0 is assigned a new value in the same cycle it was supposed to be decremented, it is assigned the new value instead of being decremented.

LETIMERn_REP0 can be written while the timer is running to allow the timer to run for longer periods at a time without stopping. Figure 21.3 (p. 323) .

22 PCNT - Pulse Counter





Quick Facts

What?

The Pulse Counter (PCNT) decodes incoming pulses. The module has a quadrature mode which may be used to decode the speed and direction of a mechanical shaft. PCNT can operate in EM0-EM3.

Why?

The PCNT generates an interrupt after a specific number of pulses (or rotations), eliminating the need for timing- or I/O interrupts and CPU processing to measure pulse widths, etc.

How?

PCNT uses the LFACLK or may be externally clocked from a pin. The module incorporates an 8-bit up/down-counter to keep track of incoming pulses or rotations.

22.1 Introduction

The Pulse Counter (PCNT) can be used for counting incoming pulses on a single input or to decode quadrature encoded inputs. It can run from the internal LFACLK (EM0-EM2) while counting pulses on the PCNTn_S0IN pin or using this pin as an external clock source (EM0-EM3) that runs both the PCNT counter and register access.

22.2 Features

- 8-bit counter with reload register
- Single input oversampling up/down counter mode (EM0-EM2)
- Externally clocked single input pulse up/down counter mode (EM0-EM3)
- Externally clocked quadrature decoder mode (EM0-EM3)
- Interrupt on counter underflow and overflow
- Interrupt when a direction change is detected (quadrature decoder mode only)
- Optional pulse width filter
- Optional input inversion/edge detect select

22.3 Functional Description

An overview of the PCNT module is shown in Figure 22.1 (p. 342) .

Bit	Name	Reset	Access	Description
2	CNTDIR	0	RW	**Non-Quadrature Mode Counter Direction Control**

The direction of the counter must be set in the OVSSINGLE and EXTCLKSINGLE modes. This bit is ignored in EXTCLKQUAD mode as the direction is automatically detected.

Value	Mode	Description
0	UP	Up counter mode.
1	DOWN	Down counter mode.

Bit	Name	Reset	Access	Description
1:0	MODE	0x0	RW	**Mode Select**

Selects the mode of operation. The corresponding clock source must be selected from the CMU.

Value	Mode	Description
0	DISABLE	The module is disabled.
1	OVSSINGLE	Single input LFACLK oversampling mode (available in EM0-EM2).
2	EXTCLKSINGLE	Externally clocked single input counter mode (available in EM0-EM3).
3	EXTCLKQUAD	Externally clocked quadrature decoder mode (available in EM0-EM3).

22.5.2 PCNTn_CMD - Command Register (Async Reg)

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

Offset	Bit Position																														
0x004	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
Access																															W1	W1
Name																															LTOPBIM	LCNTIM

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1	LTOPBIM	0	W1	**Load TOPB Immediately**

This bit has no effect since TOPB is not buffered and it is loaded directly into TOP. For EFM32G revisions A and B: Load PCNTn_TOPB into PCNTn_TOP. Please see the device datasheet for a description on how to extract the chip revision.

Bit	Name	Reset	Access	Description
0	LCNTIM	0	W1	**Load CNT Immediately**

Load PCNTn_TOP into PCNTn_CNT on the next counter clock cycle.

22.5.3 PCNTn_STATUS - Status 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																																0
Access																																R
Name																																DIR

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		

23.5.3 ACMPn_STATUS - Status 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																															0	0
Access																															R	R
Name																															ACMPOUT	ACMPACT

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1	ACMPOUT	0	R	**Analog Comparator Output**
	Analog comparator output value.			
0	ACMPACT	0	R	**Analog Comparator Active**
	Analog comparator active status.			

23.5.4 ACMPn_IEN - Interrupt Enable 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																															0	0
Access																															RW	RW
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1	WARMUP	0	RW	**Warm-up Interrupt Enable**
	Enable/disable interrupt on finished warm-up.			
0	EDGE	0	RW	**Edge Trigger Interrupt Enable**
	Enable/disable edge triggered interrupt.			

23.5.5 ACMPn_IF - Interrupt Flag Register

Offset	Bit Position																													
0x010	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
Access																															R	R
Name																															WARMUP	EDGE

27.5.7 AES_IFC - Interrupt Flag Clear Register

Offset	Bit Position																													
0x018	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
Access																																W1
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	DONE	0	W1	**Encryption/Decryption Done Interrupt Flag Clear**
	Write to 1 to clear encryption/decryption done interrupt flag			

27.5.8 AES_DATA - DATA Register

Offset	Bit Position																													
0x01C	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	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x00000000	RW	**Data Access**
	Access data through this register.			

27.5.9 AES_XORDATA - XORDATA Register

Offset	Bit Position																													
0x020	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	XORDATA																															

Bit	Name	Reset	Access	Description
31:0	XORDATA	0x00000000	RW	**XOR Data Access**
	Access data with XOR function through this register.			

27.5.10 AES_KEYLA - KEY Low Register

Offset	Bit Position																													
0x030	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	KEYLA																															

Bit	Name	Reset	Access	Description
31:0	KEYLA	0x00000000	RW	**Key Low Access A**
	Access the low key words through this register.			

27.5.11 AES_KEYLB - KEY Low Register

Offset	Bit Position																													
0x034	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	KEYLB																															

Bit	Name	Reset	Access	Description
31:0	KEYLB	0x00000000	RW	**Key Low Access B**
	Access the low key words through this register.			

28.5.9 GPIO_Px_PINLOCKN - Port Unlocked Pins Register

Offset	Bit Position																													
0x020	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																	0xFFFF														
Access																	RW														
Name																	PINLOCKN														

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	PINLOCKN	0xFFFF	RW	**Unlocked Pins**
	Shows unlocked pins in the port. To lock pin n, clear bit n. The pin is then locked until reset.			

28.5.10 GPIO_EXTIPSELL - External Interrupt Port Select Low Register

Offset	Bit Position																													
0x100	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				0x0				0x0				0x0				0x0				0x0				0x0				0x0		
Access		RW				RW				RW				RW				RW				RW				RW				RW		
Name		EXTIPSEL7				EXTIPSEL6				EXTIPSEL5				EXTIPSEL4				EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0		

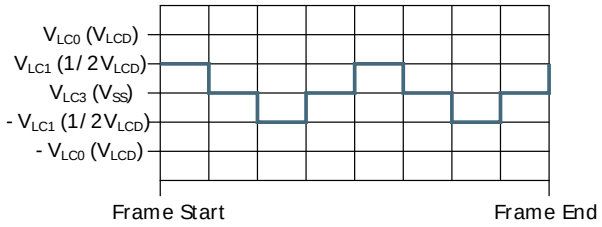
Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
30:28	EXTIPSEL7	0x0	RW	**External Interrupt 7 Port Select**
	Select input port for external interrupt 7.			

Value	Mode	Description
0	PORTA	Port A pin 7 selected for external interrupt 7
1	PORTB	Port B pin 7 selected for external interrupt 7
2	PORTC	Port C pin 7 selected for external interrupt 7
3	PORTD	Port D pin 7 selected for external interrupt 7
4	PORTE	Port E pin 7 selected for external interrupt 7
5	PORTF	Port F pin 7 selected for external interrupt 7

27	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
26:24	EXTIPSEL6	0x0	RW	**External Interrupt 6 Port Select**
	Select input port for external interrupt 6.			

Value	Mode	Description
0	PORTA	Port A pin 6 selected for external interrupt 6
1	PORTB	Port B pin 6 selected for external interrupt 6
2	PORTC	Port C pin 6 selected for external interrupt 6
3	PORTD	Port D pin 6 selected for external interrupt 6
4	PORTE	Port E pin 6 selected for external interrupt 6
5	PORTF	Port F pin 6 selected for external interrupt 6

Figure 29.10. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM1



29.3.3.3 Waveforms with 1/3 Bias and Duplex Multiplexing

In this mode, each frame is divided into 4 periods. LCD_COM[1:0] lines can be multiplexed with all segment lines. Figures show 1/3 bias and duplex multiplexing (waveforms show two frames).

Figure 29.11. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM0

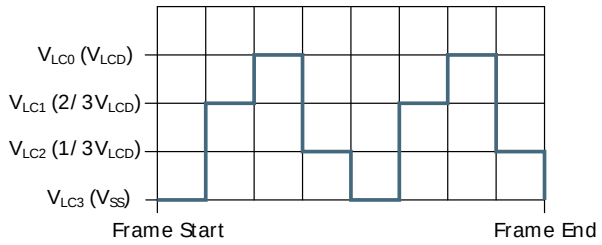
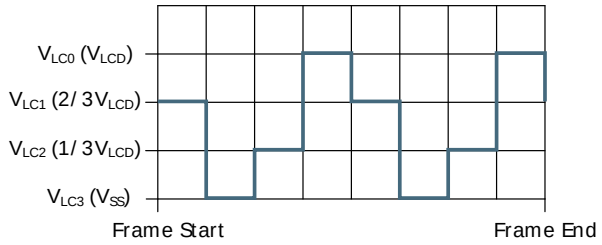


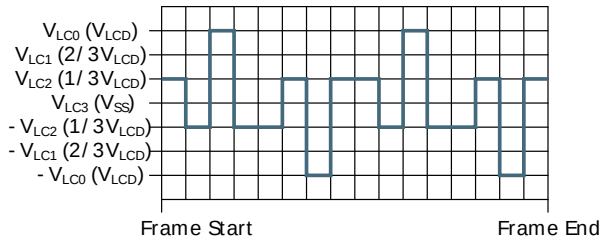
Figure 29.12. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM1



1/3 bias and duplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform on the left is just an example to illustrate how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in the figures below, this waveform will turn ON pixels connected to LCD_COM0, while pixels connected to LCD_COM1 will be turned OFF.

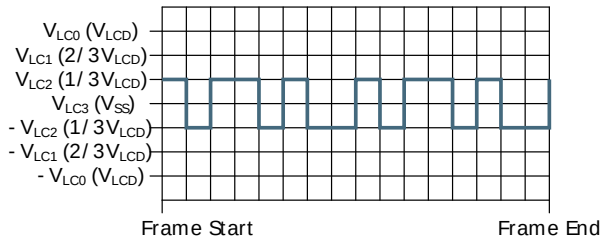
Figure 29.41. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM2



1/3 bias and quadruplex multiplexing - LCD_SEG0-LCD_COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD\_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM3 will be OFF with this waveform

Figure 29.42. LCD 1/3 Bias and Quadruplex Multiplexing- LCD_SEG0-LCD_COM3



29.3.4 LCD Contrast

Different LCD panels have different characteristics and also temperature may affect the characteristics of the LCD panels. To compensate for such variations, the LCD driver has a programmable contrast that adjusts the V_{LCD_OUT}. The contrast is set by CONLEV in LCD_DISPCTRL, and can be adjusted relative to either V_{DD} (V_{LCD}) or Ground using CONCONF in LCD_DISPCTRL. See Table 29.4 (p. 460) and Table 29.5 (p. 461) , Table 29.5 (p. 461) and Table 29.6 (p. 461) .

Table 29.4. LCD Contrast

BIAS	CONLEV	Equation	Range
00	00000-11111	$V_{LCD\_OUT} = V_{LCD}$ x (0.61 x (1 + CONLEV/(2^5 - 1)))	CONLEV = 0 => $V_{LCD\_OUT} = 0.61V_{LCD}$ CONLEV = 31 => $V_{LCD\_OUT} = V_{LCD}$
01	00000-11111	$V_{LCD\_OUT} = V_{LCD}$ x (0.53 x (1 + CONLEV/(2^5 - 1)))	CONLEV = 0 => $V_{LCD\_OUT} = 0.53V_{LCD}$ CONLEV = 31 => $V_{LCD\_OUT} = V_{LCD}$
10	00000-11111	$V_{LCD\_OUT} = V_{LCD}$ x (0.61 x (1 + CONLEV/(2^5 - 1)))	CONLEV = 0 => $V_{LCD\_OUT} = 0.61V_{LCD}$ CONLEV = 31 => $V_{LCD\_OUT} = V_{LCD}$

Note

Reset value is maximum contrast

Bit	Name	Reset	Access	Description
	Set when Frame Counter is zero.			

29.5.9 LCD_IFS - Interrupt Flag Set Register

Offset	Bit Position																															
0x020	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
Access																																W1
Name																																FC

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	FC	0	W1	**Frame Counter Interrupt Flag Set**
	Write to 1 to set FC interrupt flag.			

29.5.10 LCD_IFC - Interrupt Flag 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
Access																																W1
Name																																FC

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	FC	0	W1	**Frame Counter Interrupt Flag Clear**
	Write to 1 to clear FC interrupt flag.			

29.5.11 LCD_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x028	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
Access																																RW
Name																																FC

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		