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	Obsolete
Core Processor	HCS12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, LINbus, SCI, SPI
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	100
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 8x10b SAR
Oscillator Type	External, Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvhl32f1vlq

Table 1-9. Security Bits

SEC[1:0]	Security State
00	1 (secured)
01	1 (secured)
10	***0 (unsecured)***
11	1 (secured)

NOTE

Please refer to the Section 21.5, “Security” for more security byte details.

1.10.3 Operation of the Secured Microcontroller

By securing the device, unauthorized access to the EEPROM and Flash memory contents is prevented. Secured operation has the following effects on the microcontroller:

1.10.3.1 Normal Single Chip Mode (NS)

- Background Debug Controller (BDC) operation is completely disabled.
- Execution of Flash and EEPROM commands is restricted (described in flash block description).

1.10.3.2 Special Single Chip Mode (SS)

- Background Debug Controller (BDC) commands are restricted
- Execution of Flash and EEPROM commands is restricted (described in flash block description).

In special single chip mode the device is in active BDM after reset. In special single chip mode on a secure device, only the BDC mass erase and BDC control and status register commands are possible. BDC access to memory mapped resources is disabled. The BDC can only be used to erase the EEPROM and Flash memory without giving access to their contents.

1.10.4 Unsecuring the Microcontroller

Unsecuring the microcontroller can be done using three different methods:

1. Backdoor key access
2. Reprogramming the security bits
3. Complete memory erase

1.10.4.1 Unsecuring the MCU Using the Backdoor Key Access

In normal single chip mode, security can be temporarily disabled using the backdoor key access method. This method requires that:

- The backdoor key has been programmed to a valid value.
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’.

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Routing Register	Pin Function after Reset
A	PA7	FP7	O	LCD FP7 signal		GPIO
		(PWM6)	O	PWM channel 6	PWM6RR	
		PTA[7]	I/O	General-purpose		
	PA6	FP6	O	LCD FP6 signal		
		(PWM4)	O	PWM channel 4	PWM4RR	
		PTA[6]	I/O	General-purpose		
	PA5	FP5	O	LCD FP5 signal		
		(PWM2)	O	PWM channel 2	PWM2RR	
		PTA[5]	I/O	General-purpose		
	PA4	FP4	O	LCD FP4 signal		
		(PWM0)	O	PWM channel 0	PWM6RR	
		PTA[4]	I/O	General-purpose		
	PA3	FP3	O	LCD FP3 signal		
		(SDA0)	I/O	SDA of IIC0 signal	IIC0RR	
		PTA[3]	I/O	General-purpose		
	PA2	FP2	O	LCD FP2 signal		
		(SCL0)	I/O	SCL of IIC0 signal	IIC0RR	
		PTA[2]	I/O	General-purpose		
	PA1	FP1	O	LCD FP1 signal		
		PTA[1]	I/O	General-purpose		
	PA0	FP0	O	LCD FP0 signal		
		PTA[0]	I/O	General-purpose		
B	PB3	BP3	O	LCD BP3 signals		GPIO
		PTB[3]	I/O	General-purpose		
	PB2	BP2	O	LCD BP2 signal		
		PTB[2]	I/O	General-purpose		
	PB1	BP1	O	LCD BP1 signal		
		PTB[1]	I/O	General-purpose		
	PB0	BP0	O	LCD BP0 signal		
		PTB[0]	I/O	General-purpose		

Table 2-21. Register availability per port(1)

Port	Data	Input	Data Direction	Pull Enable	Polarity Select	Wired-Or Mode	Slew Rate Enable	Interrupt Enable	Interrupt Flag
S	yes	yes	yes	yes	yes	yes	-	yes	yes
T	yes	yes	yes	yes	yes	-	-	yes	yes
AD	yes	yes	yes	yes	yes	-	-	yes	yes
U	yes	yes	yes	yes	yes	-	yes	-	-

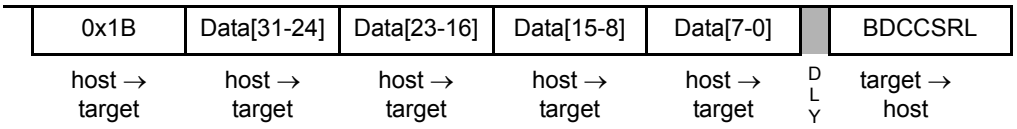
1. Each cell represents one register with individual configuration bits
2. Only PA3/PA2

Table 2-22 shows the effect of enabled peripheral features on I/O state and enabled pull devices.

Table 2-22. Effect of Enabled Features

Enabled Feature	Related Pin(s)	Effect on I/O state	Effect on enabled pull device
CPMU OSC	EXTAL, XTAL	CPMU takes control	Forced off
32K OSC	32K_EXTAL, 32K_XTAL	OSC takes control if CLKSRC in Section 18.4.2, “RTC Control Register 2 (RTCCTL2)“ is set	Forced off
LCD	FP[39:0], BP[3:0]	LCD takes control	Forced off
TIMx	OCx	Forced output	Forced off
	ICx	None (DDR maintains control)	None (PER/PPS maintain control)
SPIx	MISO, MOSI, SCK, $\overline{SS}$	Controlled input/output	Forced off if output
SCIx	TXD	Forced output	Forced off
	RXD	Forced input	None (PER/PPS maintain control)
CANx	TXCAN	Forced output	Forced off
	RXCAN	Forced input	Pulldown forced off
IICx	SCL, SDA	Controlled input/output	Forced off if output
S12ZDBG	PDO, PDOCLK	Forced output	Forced off
SSGx	SGA, SGT	Forced output	Forced off
PWM channel	PWMx	Forced output	Forced off
MC	MxCxM, MxCxP	Forced output	Forced off
SSDx	MxCOSM, MxCOSP, MxSINM, MxSINP	Controlled input/output	Forced off if output
API	API_EXTCLK	Forced output	Forced off
ADCx	ANx	None (DDR maintains control(1))	None (PER/PPS maintain control)
LINPHYx	LPTXD0	Forced input	None (PER/PPS maintain control)
	LPRXD0	Forced output	Forced off

FILL_MEM.sz_WS



FILL_MEM{_WS} is used with the WRITE_MEM{_WS} command to access large blocks of memory. An initial WRITE_MEM{_WS} is executed to set up the starting address of the block and write the first datum. If an initial WRITE_MEM{_WS} is not executed before the first FILL_MEM{_WS}, an illegal command response is returned. The FILL_MEM{_WS} command stores subsequent operands. The initial address is incremented by the operand size (1, 2, or 4) and saved in a temporary register. Subsequent FILL_MEM{_WS} commands use this address, perform the memory write, increment it by the current operand size, and store the updated address in the temporary register. If the with-status option is specified, the BDCCSRL status byte is returned after the write data. This status byte reflects the state after the memory write was performed. If enabled an ACK pulse is generated after the internal write access has been completed or aborted. The effect of the access size and alignment on the next address to be accessed is explained in more detail in Section 5.4.5.2”

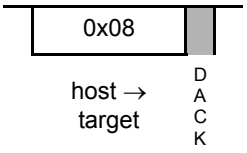
NOTE

FILL_MEM{_WS} is a valid command only when preceded by SYNC, NOP, WRITE_MEM{_WS}, or another FILL_MEM{_WS} command. Otherwise, an illegal command response is returned, setting the ILLCMD bit. NOP can be used for inter command padding without corrupting the address pointer.

The size field (sz) is examined each time a FILL_MEM{_WS} command is processed, allowing the operand size to be dynamically altered. The examples show the FILL_MEM.B{_WS}, FILL_MEM.W{_WS} and FILL_MEM.L{_WS} commands.

5.4.4.7 GO

Go **Non-intrusive**



This command is used to exit active BDM and begin (or resume) execution of CPU application code. The CPU pipeline is flushed and refilled before normal instruction execution resumes. Prefetching begins at the current address in the PC. If any register (such as the PC) is altered by a BDC command whilst in BDM, the updated value is used when prefetching resumes. If enabled, an ACK is driven on exiting active BDM.

If a GO command is issued whilst the BDM is inactive, an illegal command response is returned and the ILLCMD bit is set.

The handshake protocol is enabled by the ACK_ENABLE command. The BDC sends an ACK pulse when the ACK_ENABLE command has been completed. This feature can be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol.

Unlike the normal bit transfer, where the host initiates the transmission by issuing a negative edge on the BKGD pin, the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. Figure 5-9 specifies the timing when the BKGD pin is being driven. The host must follow this timing constraint in order to avoid the risk of an electrical conflict at the BKGD pin.

When the handshake protocol is enabled, the STEAL bit in BDCCSR selects if bus cycle stealing is used to gain immediate access. If STEAL is cleared, the BDC is configured for low priority bus access using free cycles, without stealing cycles. This guarantees that BDC accesses remain truly non-intrusive to not affect the system timing during debugging. If STEAL is set, the BDC gains immediate access, if necessary stealing an internal bus cycle.

NOTE

If bus steals are disabled then a loop with no free cycles cannot allow access. In this case the host must recognize repeated NORESP messages and then issue a BACKGROUND command to stop the target and access the data.

Figure 5-10 shows the ACK handshake protocol without steal in a command level timing diagram. The READ_MEM.B command is used as an example. First, the 8-bit command code is sent by the host, followed by the address of the memory location to be read. The target BDC decodes the command. Then an internal access is requested by the BDC. When a free bus cycle occurs the READ_MEM.B operation is carried out. If no free cycle occurs within 512 core clock cycles then the access is aborted, the NORESP flag is set and the target generates a Long-ACK pulse.

Having retrieved the data, the BDC issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the data read part of the command.

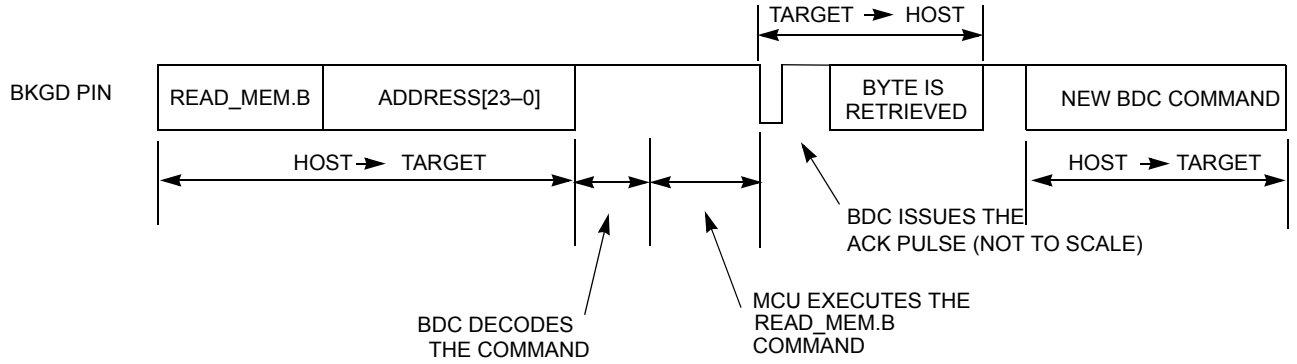


Figure 5-10. Handshake Protocol at Command Level

Alternatively, setting the STEAL bit configures the handshake protocol to make an immediate internal access, independent of free bus cycles.

Table 7-16. COP Watchdog Rates if COPOSCSEL1=1.

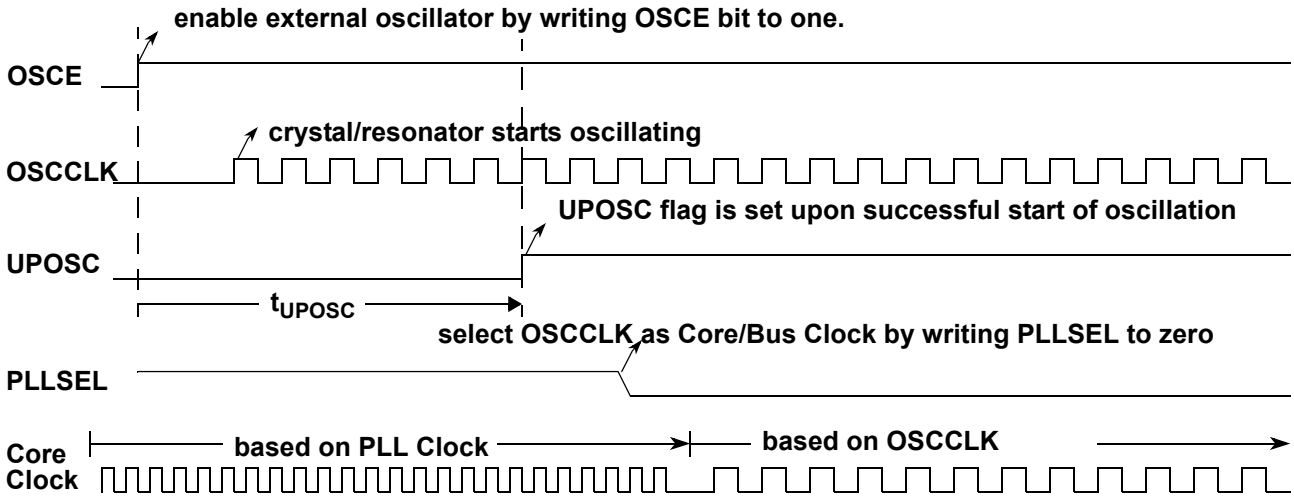
CR2	CR1	CR0	COPCLK Cycles to time-out (COPCLK is ACLK divided by 2)
0	0	0	COP disabled
0	0	1	2^{7}
0	1	0	2^{9}
0	1	1	2^{11}
1	0	0	2^{13}
1	0	1	2^{15}
1	1	0	2^{16}
1	1	1	2^{17}

7.4.5 External Oscillator

7.4.5.1 Enabling the External Oscillator

An example of how to use the oscillator as source of the Bus Clock is shown in Figure 7-39.

Figure 7-39. Enabling the external oscillator



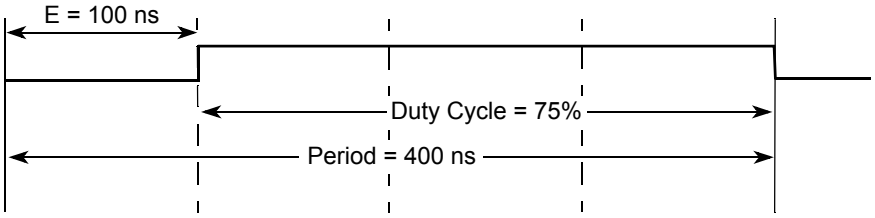


Figure 9-18. PWM Left Aligned Output Example Waveform

9.4.2.6 Center Aligned Outputs

For center aligned output mode selection, set the CAEx bit (CAEx = 1) in the PWMCAE register and the corresponding PWM output will be center aligned.

The 8-bit counter operates as an up/down counter in this mode and is set to up whenever the counter is equal to $00. The counter compares to two registers, a duty register and a period register as shown in the block diagram in Figure 9-16. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register changes the counter direction from an up-count to a down-count. When the PWM counter decrements and matches the duty register again, the output flip-flop changes state causing the PWM output to also change state. When the PWM counter decrements and reaches zero, the counter direction changes from a down-count back to an up-count and a load from the double buffer period and duty registers to the associated registers is performed, as described in Section 9.4.2.3, “PWM Period and Duty”. The counter counts from 0 up to the value in the period register and then back down to 0. Thus the effective period is PWMPERx*2.

NOTE

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

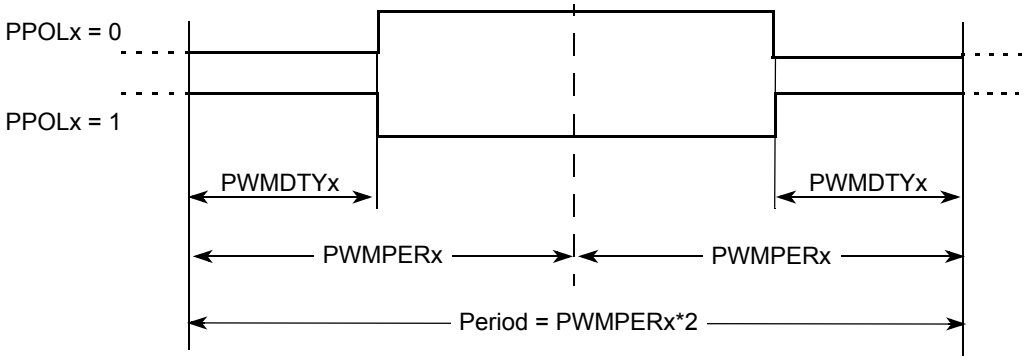


Figure 9-19. PWM Center Aligned Output Waveform

Table 10-10. Summary of Conversion Flow Control Bit Scenarios

RSTA	TRIG	SEQA	LDOK	Conversion Flow Control Mode	Conversion Flow Control Scenario
0	0	0	0	Both Modes	Valid
0	0	0	1	Both Modes	Can Not Occur
0	0	1	0	Both Modes	Valid[5.]
0	0	1	1	Both Modes	Can Not Occur
0	1	0	0	Both Modes	Valid[2.]
0	1	0	1	Both Modes	Can Not Occur
0	1	1	0	Both Modes	Can Not Occur
0	1	1	1	Both Modes	Can Not Occur
1	0	0	0	Both Modes	Valid[4.]
1	0	0	1	Both Modes	Valid[1. 4.]
1	0	1	0	Both Modes	Valid[3. 4. 5.]
1	0	1	1	Both Modes	Valid[1. 3. 4. 5.]
1	1	0	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2. 4. 6.]
1	1	0	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[1. 2. 4. 6.]
1	1	1	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2. 3. 4. 5. 6.]
1	1	1	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[(1) (2) (3) (4) (5) (6)]

1. Swap CSL buffer
2. Start conversion sequence
3. Prevent RSTA_EIF and LDOK_EIF
4. Load conversion command from top of CSL
5. Abort any ongoing conversion, conversion sequence and CSL
6. Bit TRIG set automatically in Trigger Mode

For a detailed description of all conversion flow control bit scenarios please see also Section 10.5.3.2.4, “The two conversion flow control Mode Configurations, Section 10.5.3.2.5, “The four ADC conversion flow control bits and Section 10.5.3.2.6, “Conversion flow control in case of conversion sequence control bit overrun scenarios

10.4.2.8 ADC Interrupt Enable Register (ADCIE)

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	SEQAD_IE	CONIF_OIE	Reserved	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-11. ADC Interrupt Enable Register (ADCIE)

Read: Anytime

Write: Anytime

Table 10-12. ADCIE Field Descriptions

Field	Description
7 SEQAD_IE	**Conversion Sequence Abort Done Interrupt Enable Bit** — This bit enables the conversion sequence abort event done interrupt. 0 Conversion sequence abort event done interrupt disabled. 1 Conversion sequence abort event done interrupt enabled.
6 CONIF_OIE	**ADCCONIF Register Flags Overrun Interrupt Enable** — This bit enables the flag which indicates if an overrun situation occurred for one of the CON_IF[15:1] flags or for the EOL_IF flag. 0 No ADCCONIF Register Flag overrun occurred. 1 ADCCONIF Register Flag overrun occurred.

10.4.2.17 ADC Command Register 2 (ADCCMD_2)

A command which contains reserved bit settings causes the error flag CMD_EIF being set and ADC cease operation.

Module Base + 0x0016

	15	14	13	12	11	10	9	8
R	SMP[4:0]					0	0	Reserved
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-20. ADC Command Register 2 (ADCCMD_2)

Read: Anytime

Write: Only writable if bit SMOD_ACC is set
(see also Section 10.4.2.2, “ADC Control Register 1 (ADCCTL_1) bit SMOD_ACC description for more details)

Table 10-24. ADCCMD_2 Field Descriptions

Field	Description
15-11 SMP[4:0]	**Sample Time Select Bits** — These four bits select the length of the sample time in units of ADC conversion clock cycles. Note that the ADC conversion clock period is itself a function of the prescaler value (bits PRS[6:0]). Table 10-25 lists the available sample time lengths.

NOTE

If bit SMOD_ACC is set modifying this register must be done carefully - only when no conversion and conversion sequence is ongoing.

Table 10-25. Sample Time Select

SMP[4]	SMP[3]	SMP[2]	SMP[1]	SMP[0]	Sample Time in Number of ADC Clock Cycles
0	0	0	0	0	4
0	0	0	0	1	5
0	0	0	1	0	6
0	0	0	1	1	7
0	0	1	0	0	8
0	0	1	0	1	9
0	0	1	1	0	10
0	0	1	1	1	11
0	1	0	0	0	12
0	1	0	0	1	13

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 11-13. CANTIER Register Field Descriptions

Field	Description
2-0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

11.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 11-14. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	**Abort Request** — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and abort acknowledge flags (ABTAK, see Section 11.3.2.10, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)") are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

LIN: Local Interconnect Network

LSB: Least Significant Bit

MSB: Most Significant Bit

NRZ: Non-Return-to-Zero

RZI: Return-to-Zero-Inverted

RXD: Receive Pin

SCI : Serial Communication Interface

TXD: Transmit Pin

12.1.2 Features

The SCI includes these distinctive features:

- Full-duplex or single-wire operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse widths
- 16-bit baud rate selection
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable polarity for transmitter and receiver
- Programmable transmitter output parity
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation with eight flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
  - Receive wakeup on active edge
  - Transmit collision detect supporting LIN
  - Break Detect supporting LIN

12.3.2.5 SCI Alternative Control Register 2 (SCIACR2)

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	IREN	TNP1	TNP0	0	0	BERRM1	BERRM0	BKDFE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-8. SCI Alternative Control Register 2 (SCIACR2)

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

Table 12-7. SCIACR2 Field Descriptions

Field	Description
7 IREN	**Infrared Enable Bit** — This bit enables/disables the infrared modulation/demodulation submodule. 0 IR disabled 1 IR enabled
6:5 TNP[1:0]	**Transmitter Narrow Pulse Bits** — These bits enable whether the SCI transmits a 1/16, 3/16, 1/32 or 1/4 narrow pulse. See Table 12-8.
2:1 BERRM[1:0]	**Bit Error Mode** — Those two bits determines the functionality of the bit error detect feature. See Table 12-9.
0 BKDFE	**Break Detect Feature Enable** — BKDFE enables the break detect circuitry. 0 Break detect circuit disabled 1 Break detect circuit enabled

Table 12-8. IRSCI Transmit Pulse Width

TNP[1:0]	Narrow Pulse Width
11	1/4
10	1/32
01	1/16
00	3/16

Table 12-9. Bit Error Mode Coding

BERRM1	BERRM0	Function
0	0	Bit error detect circuit is disabled
0	1	Receive input sampling occurs during the 9th time tick of a transmitted bit (refer to Figure 12-19)
1	0	Receive input sampling occurs during the 13th time tick of a transmitted bit (refer to Figure 12-19)
1	1	Reserved

Table 13-4. SPICR2 Field Descriptions

Field	Description
6 XFRW	**Transfer Width —** This bit is used for selecting the data transfer width. If 8-bit transfer width is selected, SPIDRL becomes the dedicated data register and SPIDRH is unused. If 16-bit transfer width is selected, SPIDRH and SPIDRL form a 16-bit data register. Please refer to Section 13.3.2.4, “SPI Status Register (SPISR) for information about transmit/receive data handling and the interrupt flag clearing mechanism. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 8-bit Transfer Width (n = 8)(1) 1 16-bit Transfer Width (n = 16)1
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the SS port pin is not used by the SPI. In slave mode, the SS is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the SS port pin configuration, refer to Table 13-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 SS port pin is not used by the SPI. 1 SS port pin with MODF feature.
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled. 1 Output buffer enabled.
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in wait mode. 1 Stop SPI clock generation when in wait mode.
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 13-5. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.

1. n is used later in this document as a placeholder for the selected transfer width.

Table 13-5. Bidirectional Pin Configurations

Pin Mode	SPC0	BIDIROE	MISO	MOSI
Master Mode of Operation				
Normal	0	X	Master In	Master Out
Bidirectional	1	0	MISO not used by SPI	Master In
		1		Master I/O
Slave Mode of Operation				
Normal	0	X	Slave Out	Slave In
Bidirectional	1	0	Slave In	MOSI not used by SPI
		1	Slave I/O	

14.7.1.7 Arbitration Lost

If several masters try to engage the bus simultaneously, only one master wins and the others lose arbitration. The devices which lost arbitration are immediately switched to slave receive mode by the hardware. Their data output to the SDA line is stopped, but SCL continues to be generated until the end of the byte during which arbitration was lost. An interrupt occurs at the falling edge of the ninth clock of this transfer with IBAL=1 and MS/SL=0. If one master attempts to start transmission while the bus is being engaged by another master, the hardware will inhibit the transmission; switch the MS/SL bit from 1 to 0 without generating STOP condition; generate an interrupt to CPU and set the IBAL to indicate that the attempt to engage the bus is failed. When considering these cases, the slave service routine should test the IBAL first and the software should clear the IBAL bit if it is set.

17.3 Memory Map and Register Definition

This section provides a detailed description of all registers of the stepper stall detector (SSD) block.

17.3.1 Module Memory Map

Table 17-3 gives an overview of all registers in the SSDV2 memory map. The SSDV2 occupies eight bytes in the memory space. The register address results from the addition of *base address* and *address offset*. The *base address* is determined at the MCU level and is given in the Device Overview chapter. The *address offset* is defined at the block level and is given here.

Table 17-3. SSDV2 Memory Map

Address Offset	Use	Access
0x0000	RTZCTL	R/W
0x0001	MDCCTL	R/W
0x0002	SSDCTL	R/W
0x0003	SSDFLG	R/W
0x0004	MDCCNT (High)	R/W
0x0005	MDCCNT (Low)	R/W
0x0006	ITGACC (High)	R
0x0007	ITGACC (Low)	R

17.3.2 Register Descriptions

This section describes in detail all the registers and register bits in the SSDV2 block. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

the variation in the RTC clock output and determine the correct calibration point. Refer to Device spec for more information on timer channel connection.

18.6.2 RTC compensation

To reach the high precision RTC clock, the compensation need following steps.

- Characterization of the OSCCLK or OSCCLK_32K crystal, get the lookup table for the crystal frequency versus to temperature. This step is just required one time per crystal type.
- Perform the RTC calibration at production for each device, get the base frequency offset.
- Perform the RTC compensation periodically after enabling the RTC.
  - Measure the external temperature
  - Lookup compensation value in lookup table
  - Load the compensation value into RTC

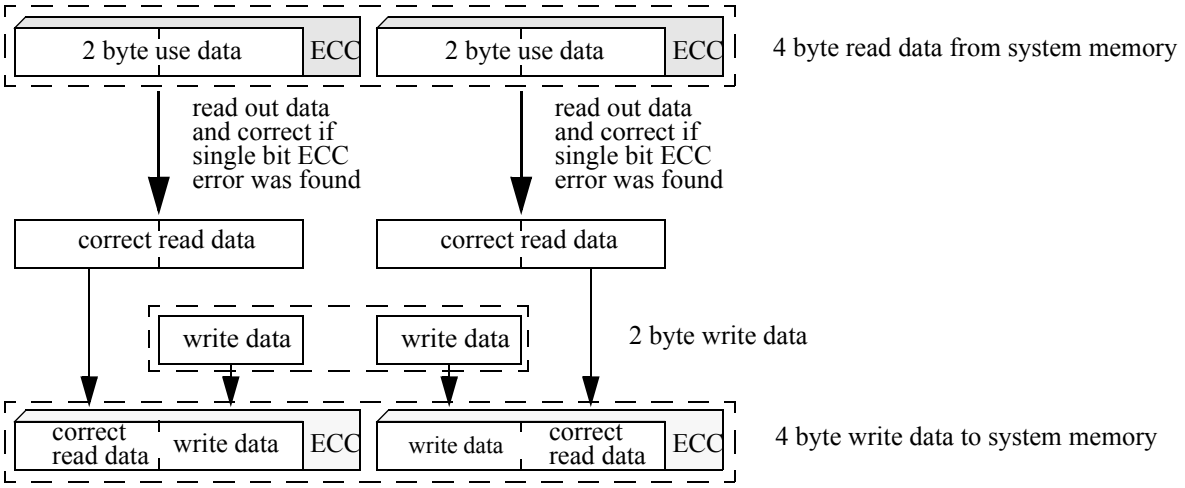


Figure 20-9. 2 byte non-aligned write access

20.3.3 Memory read access

During each memory read access an ECC check is performed. If the logic detects a single bit ECC error then the module corrects the data, so that the access initiator module receives correct data. In parallel, the logic writes the corrected data back to the memory, so that this read access repairs the single bit ECC error. This automatic ECC read repair function is disabled by setting the ECCDRR bit.

If a single bit ECC error was detected then the SBEEIF flag is set.

If the logic detects a double bit ECC error, then the data word is flagged as invalid, so that the access initiator module can ignore the data.

20.3.4 Memory initialization

Memory operation which allows a read before a first write, like the read-modify-write operation of the un-aligned access, requires that the memory contains valid ECC values before the first read-modify-write access is performed to avoid spurious ECC error reporting. The ECC module provides logic to initialize the complete memory content with zero during the power up phase. During the initialization process the access to the SRAM is disabled and the RDY status bit is cleared. If the initialization process is done, the SRAM access is possible and the RDY status bit is set.

20.3.5 Interrupt handling

This sections describes the interrupts generated by the SRAM_ECC module and their individual sources, Vector addresses and interrupt priority are defined by MCU level.

Table 21-9. FSEC Field Descriptions (continued)

Field	Description
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 21-11. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 21-10. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED(1)
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable backdoor key access.

Table 21-11. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED(1)
10	UNSECURED
11	SECURED

1. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 21.5.

21.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to indicate the amount of parameters loaded into the FCCOB registers for Flash memory operations.

Offset Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 21-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.