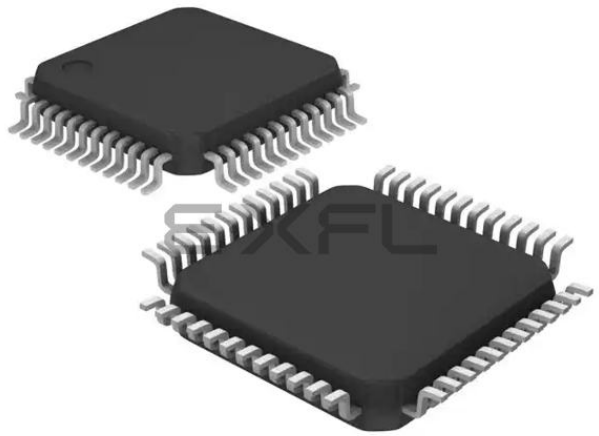


Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 10x12b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvca64f0mlf

Address	Module	Size (Bytes)
0x0718–0x077F	Reserved	104
0x0780–0x0787	SPI0	8
0x0788–0x078F	Reserved	8
0x0790–0x0797	SPI1	8
0x0798–0x07BF	Reserved	32
0x07C0–0x07C7	IIC0	8
0x07C8–0x097F	Reserved	56
0x0800–0x083F	CAN0	64
0x0840–0x098F	Reserved	336
0x0990–0x0997	CANPHY0	8
0x0998–0x099F		8
0x09A0–0x09AF	SENTTX	16
0x09B0–0x0FFF		1616

NOTE

Reserved register space shown in Table 1-2. is not allocated to any module. This register space is reserved for future use. Writing to these locations has no effect. Read access to these locations returns zero.

Table 1-3. S12ZVC-Family Memory Address Ranges

Device	Address	Memory Block	Size (Bytes)
MC9S12ZVC64 & MC9S12ZVCA64	0x00_1000 - 0x00_1FFF	SRAM	4K
	0x10_0000 - 0x10_03FF	EEPROM	1K
	0xFF_0000 - 0xFF_FFFF	Program Flash	64K
MC9S12ZVC96 & MC9S12ZVCA96	0x00_1000 - 0x00_2FFF	SRAM	8K
	0x10_0000 - 0x10_07FF	EEPROM	2K
	0xFE_8000 - 0xFF_FFFF	Program Flash	96K
MC9S12ZVC128 & MC9S12ZVCA128	0x00_1000 - 0x00_2FFF	SRAM	8K
	0x10_0000 - 0x10_07FF	EEPROM	2K
	0xFE_0000 - 0xFF_FFFF	Program Flash	128K
MC9S12ZVC192 & MC9S12ZVCA192	0x00_1000 - 0x00_3FFF	SRAM	12K
	0x10_0000 - 0x10_07FF	EEPROM	2K
	0xFD_0000 - 0xFF_FFFF	Program Flash	192K

1.7.21 Interrupt signals — $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$

$\overline{\text{IRQ}}$ is a maskable level or falling edge sensitive input. $\overline{\text{XIRQ}}$ is a non-maskable level-sensitive interrupt.

1.7.22 Oscillator and Clock Signals

1.7.22.1 4-16MHZ Oscillator Signal — EXTAL and XTAL

EXTAL and XTAL are the crystal driver. On reset, the OSC is not enabled, all the device clocks are derived from the internal reference clock. EXTAL is the oscillator input. XTAL is the oscillator output.

1.7.22.2 ECLK

This signal is associated with the output of the divided bus clock (ECLK).

NOTE

This feature is only intended for debug purposes at room temperature. It must not be used for clocking external devices in an application.

1.7.23 BDC and Debug Signals

1.7.23.1 BKGD — Background Debug Signal

The BKGD signal is used as a pseudo-open-drain signal for the background debug communication. The BKGD signal has an internal pull-up device.

1.7.23.2 DBGEEV — External Event Input

This signal is the DBG external event input. It is input only. Within the DBG module, it allows an external event to force a state sequencer transition. A falling edge at the external event signal constitutes an event. Rising edges have no effect. The maximum frequency of events is half the internal core bus frequency.

1.7.24 CAN0 Signals

1.7.24.1 RXCAN0 Signal

This signal is associated with the receive functionality of the scalable controller area network controller (MSCAN0).

1.7.24.2 TXCAN0 Signal

This signal is associated with the transmit functionality of the scalable controller area network controller (MSCAN0).

Table 2-6. MODRR1 Routing Register Field Descriptions

Field	Description
2 TRIG0NEG	**Module Routing Register** — ADC0 trigger input inverted polarity 1 Falling edge active on ADC0 trigger input 0 Rising edge active on ADC0 trigger input
1-0 TRIG0RR	**Module Routing Register** — ADC0 trigger input routing 11 PP4 (ETRIG0) to ADC0 trigger input 10 PAD1 (ETRIG0) to ADC0 trigger input 01 PAD15 (ETRIG0) to ADC0 trigger input 00 TIM0 output compare channel 2 to ADC0 trigger input (output compare function on pin remains active unless disabled in timer config register TIM0OCPD[OCPD2]=1)

2.3.2.3 Module Routing Register 2 (MODRR2)

Address 0x0202 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	P0C7RR	0	0	0	P0C3RR	0	0	0
W								
	PWM0_7	—	—	—	PWM0_3	—	—	—
Reset	0	0	0	0	0	0	0	0

Figure 2-4. Module Routing Register 2 (MODRR2)

[1] Read: Anytime
Write: Once in normal, anytime in special mode

Table 2-7. MODRR2 Routing Register Field Descriptions

Field	Description
7 P0C7RR	**Module Routing Register** — PWM0_7 routing 1 PWM0_7 to PT3 0 PWM0_7 to PP3
3 P0C3RR	**Module Routing Register** — PWM0_3 routing 1 PWM0_3 to PT2 0 PWM0_3 to PP1

2.3.4.6 Port L Polarity Select Register (PPSL)

Address 0x0334 PPSL — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PPSL1	PPSL0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-26. Port L Polarity Select Register (PPSL)

[1] Read: Anytime
Write: Anytime

Table 2-26. PPSL Register Field Descriptions

Field	Description
1-0 PPSL1-0	**Polarity Select —** This bit selects the polarity of the active interrupt edge on the associated HVI pin. 1 Rising edge selected 0 Falling edge selected

2.3.4.7 Port L ADC Bypass Register (PTABYPL)

Address 0x033A — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTABYPL1	PTABYPL0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-27. Port L ADC Bypass Register (PTABYPL)

[1] Read: Anytime
Write: Anytime

Table 2-27. PTABYPL Register Field Descriptions

Field	Description
1-0 PTABYPL 1-0	**Port L ADC Connection Bypass —** This bit bypasses and powers down the impedance converter stage in the signal path from the analog input pin to the ADC channel input. This bit takes effect only if using direct input connection to the ADC channel (PTADIRL=1). 1 Impedance converter bypassed 0 Impedance converter used

After resuming application code execution, if ACK is enabled, the BDC awaits a return to active BDM before driving an ACK pulse. timeouts do not apply when awaiting a GO_UNTIL command ACK.

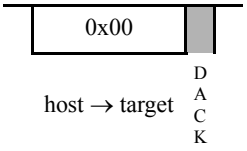
If a GO_UNTIL is not acknowledged then a SYNC command must be issued to end the pending GO_UNTIL.

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

If ACK handshaking is disabled, the GO_UNTIL command is identical to the GO command.

3.4.4.9 NOP

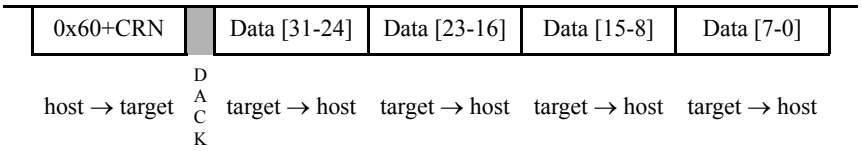
No operation **Active Background**



NOP performs no operation and may be used as a null command where required.

3.4.4.10 READ_Rn

Read CPU register **Active Background**



This command reads the selected CPU registers and returns the 32-bit result. Accesses to CPU registers are always 32-bits wide, regardless of implemented register width. Bytes that are not implemented return zero. The register is addressed through the CPU register number (CRN). See Section 3.4.5.1, “BDC Access Of CPU Registers for the CRN address decoding. If enabled, an ACK pulse is driven before the data bytes are transmitted.

If the device is not in active BDM, this command is illegal, the ILLCMD bit is set and no access is performed.

- All illegal accesses performed by the ADC module trigger error interrupts. See ADC section for details.

NOTE

Illegal accesses caused by S12ZCPU opcode prefetches will also trigger machine exceptions, even if those opcodes might not be executed in the program flow. To avoid these machine exceptions, S12ZCPU instructions must not be executed from the last (high addresses) 8 bytes of RAM, EEPROM, and Flash.

4.4.3 Uncorrectable ECC Faults

RAM and flash use error correction codes (ECC) to detect and correct memory corruption. Each uncorrectable memory corruption, which is detected during a S12ZCPU or ADC access triggers a machine exception. Uncorrectable memory corruptions which are detected during a S12ZBDC access, are captured in the RAMWF or the RDINV bit of the BDCCSRL register.

access is always a 2 byte aligned memory access, so that no ECC check is performed and no single or double bit ECC error indication is activated.

7.3.7.2 ECC Debug Memory Read Access

Writing one to the ECCDR bit performs a debug read access from the memory address defined by register DPTR. If the ECCDR bit is cleared then the register DDATA contains the uncorrected read data from the memory. The register DECC contains the ECC value read from the memory. Independent of the ECCDRR register bit setting, the debug read access will not perform an automatic ECC repair during read access. During the debug read access no ECC check is performed, so that no single or double bit ECC error indication is activated.

If the ECCDW and the ECCDR bits are set at the same time, then only the debug write access is performed.

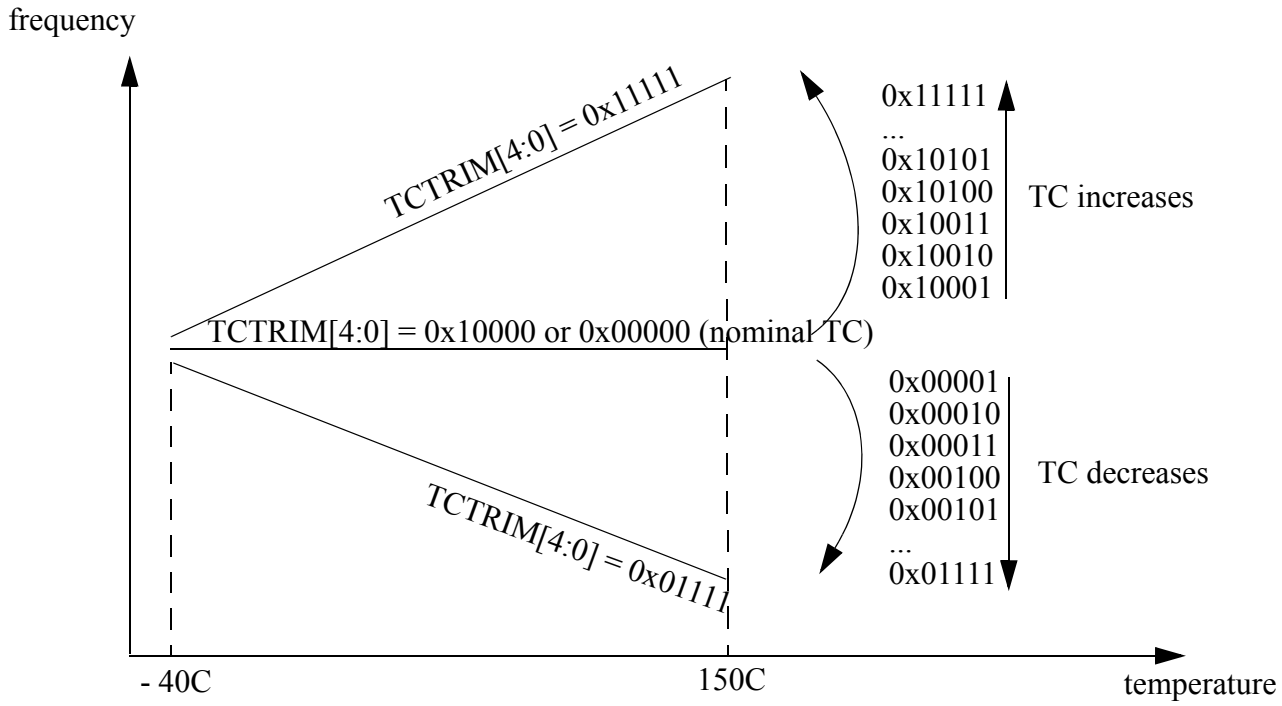


Figure 8-30. Influence of TCTRIM[4:0] on the Temperature Coefficient

NOTE

The frequency is not necessarily linear with the temperature (in most cases it will not be). The above diagram is meant only to give the direction (positive or negative) of the variation of the TC, relative to the nominal TC.

Setting TCTRIM[4:0] at 0x00000 or 0x10000 does not mean that the temperature coefficient will be zero. These two combinations basically switch off the TC compensation module, which results in the nominal TC of the IRC1M.

8.4.6.3 PLL Bypassed External Mode (PBE)

In this mode, the Bus Clock is based on the external oscillator clock. The reference clock for the PLL is based on the external oscillator.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock or the RC-Oscillator (ACLK).

This mode can be entered from default mode PEI by performing the following steps:

1. Make sure the PLL configuration is valid.
2. Enable the external Oscillator (OSCE bit)
3. Wait for the oscillator to start-up and the PLL being locked (LOCK = 1) and (UPOSC =1)
4. Clear all flags in the CPMUIFLG register to be able to detect any status bit change.
5. Optionally status interrupts can be enabled (CPMUINT register).
6. Select the Oscillator clock as source of the Bus clock (PLLSEL=0)

Loosing PLL lock status (LOCK=0) means loosing the oscillator status information as well (UPOSC=0).

The impact of loosing the oscillator status (UPOSC=0) in PBE mode is as follows:

- PLLSEL is set automatically and the source of the Bus clock is switched back to the PLL clock.
- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

8.5 Resets

8.5.1 General

All reset sources are listed in Table 8-33. There is only one reset vector for all these reset sources. Refer to MCU specification for reset vector address.

Table 8-33. Reset Summary

Reset Source	Local Enable
Power-On Reset (POR)	None
Low Voltage Reset (LVR)	None
External pin $\overline{\text{RESET}}$	None
PLL Clock Monitor Reset	None
Oscillator Clock Monitor Reset	OSCE Bit in CPMUOSC register and OMRE Bit in CPMUOSC2 register
COP Reset	CR[2:0] in CPMUCOP register

9.4.2 Register Descriptions

This section describes in address order all the ADC12B_LBA registers and their individual bits.

9.4.2.1 ADC Control Register 0 (ADCCTL_0)

Module Base + 0x0000

	15	14	13	12	11	10	9	8
R W	ADC_EN	ADC_SR	FRZ_MOD	SWAI	ACC_CFG[1:0]		STR_SEQA	MOD_CFG
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-4. ADC Control Register 0 (ADCCTL_0)

Read: Anytime

Write:

- Bits ADC_EN, ADC_SR, FRZ_MOD and SWAI writable anytime
- Bits MOD_CFG, STR_SEQA and ACC_CFG[1:0] writable if bit ADC_EN clear or bit SMOD_ACC set

Table 9-2. ADCCTL_0 Field Descriptions

Field	Description
15 ADC_EN	**ADC Enable Bit** — This bit enables the ADC (e.g. sample buffer amplifier etc.) and controls accessibility of ADC register bits. When this bit gets cleared any ongoing conversion sequence will be aborted and pending results or the result of current conversion gets discarded (not stored). The ADC cannot be re-enabled before any pending action or action in process is finished or aborted, which could take up to a maximum latency time of $t_{DISABLE}$ (see device reference manual for more details). Because internal components of the ADC are turned on/off with this bit, the ADC requires a recovery time period (t_{REC}) after ADC is enabled until the first conversion can be launched via a trigger. 0 ADC disabled. 1 ADC enabled.
14 ADC_SR	**ADC Soft-Reset** — This bit causes an ADC Soft-Reset if set after a severe error occurred (see list of severe errors in Section 9.4.2.9, "ADC Error Interrupt Flag Register (ADCEIF) that causes the ADC to cease operation). It clears all overrun flags and error flags and forces the ADC state machine to its idle state. It also clears the Command Index Register, the Result Index Register, and the CSL_SEL and RVL_SEL bits (to be ready for a new control sequence to load new command and start execution again from top of selected CSL). A severe error occurs if an error flag is set which cause the ADC to cease operation. In order to make the ADC operational again an ADC Soft-Reset must be issued. Once this bit is set it can not be cleared by writing any value. It is cleared only by ADC hardware after the Soft-Reset has been executed. 0 No ADC Soft-Reset issued. 1 Issue ADC Soft-Reset.
13 FRZ_MOD	**Freeze Mode Configuration** — This bit influences conversion flow during Freeze Mode. 0 ADC continues conversion in Freeze Mode. 1 ADC freezes the conversion at next conversion boundary at Freeze Mode entry.
12 SWAI	**Wait Mode Configuration** — This bit influences conversion flow during Wait Mode. 0 ADC continues conversion in Wait Mode. 1 ADC halts the conversion at next conversion boundary at Wait Mode entry.

9.4.2.2 ADC Control Register 1 (ADCCTL_1)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	CSL_BMOD	RVL_BMOD	SMOD_ACC	AUT_RSTA	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-5. ADC Control Register 1 (ADCCTL_1)

Read: Anytime

Write:

- Bit CSL_BMOD and RVL_BMOD writable if bit ADC_EN clear or bit SMOD_ACC set
- Bit SMOD_ACC only writable in MCU Special Mode
- Bit AUT_RSTA writable anytime

Table 9-4. ADCCTL_1 Field Descriptions

Field	Description
7 CSL_BMOD	**CSL Buffer Mode Select Bit** — This bit defines the CSL buffer mode. This bit is only writable if ADC_EN is clear. 0 CSL single buffer mode. 1 CSL double buffer mode.
6 RVL_BMOD	**RVL Buffer Mode Select Bit** — This bit defines the RVL buffer mode. 0 RVL single buffer mode 1 RVL double buffer mode
5 SMOD_ACC	**Special Mode Access Control Bit** — This bit controls register access rights in MCU Special Mode. This bit is automatically cleared when leaving MCU Special Mode. Note: When this bit is set also the ADCCMD register is writeable via the data bus to allow modification of the current command for debugging purpose. But this is only possible if the current command is not already processed (conversion not started). Please see access details given for each register. Care must be taken when modifying ADC registers while bit SMOD_ACC is set to not corrupt a possible ongoing conversion. 0 Normal user access - Register write restrictions exist as specified for each bit. 1 Special access - Register write restrictions are lifted.
4 AUT_RSTA	**Automatic Restart Event after exit from MCU Stop and Wait Mode (SWAI set)** — This bit controls if a Restart Event is automatically generated after exit from MCU Stop Mode or Wait Mode with bit SWAI set. It can be configured for ADC conversion flow control mode “Trigger Mode” and “Restart Mode” (anytime during application runtime). 0 No automatic Restart Event after exit from MCU Stop Mode. 1 Automatic Restart Event occurs after exit from MCU Stop Mode.

Chapter 13
Pulse-Width Modulator (S12PWM8B8CV2)

Table 13-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
v02.00	Feb. 20, 2009	All	Initial revision of scalable PWM. Started from pwm_8b8c (v01.08).

13.1 Introduction

The Version 2 of S12 PWM module is a channel scalable and optimized implementation of S12 PWM8B8C Version 1. The channel is scalable in pairs from PWM0 to PWM7 and the available channel number is 2, 4, 6 and 8. The shutdown feature has been removed and the flexibility to select one of four clock sources per channel has improved. If the corresponding channels exist and shutdown feature is not used, the Version 2 is fully software compatible to Version 1.

13.1.1 Features

The scalable PWM block includes these distinctive features:

- Up to eight independent PWM channels, scalable in pairs (PWM0 to PWM7)
- Available channel number could be 2, 4, 6, 8 (refer to device specification for exact number)
- Programmable period and duty cycle for each channel
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels
- Up to eight 8-bit channel or four 16-bit channel PWM resolution
- Four clock sources (A, B, SA, and SB) provide for a wide range of frequencies
- Programmable clock select logic

13.1.2 Modes of Operation

There is a software programmable option for low power consumption in wait mode that disables the input clock to the prescaler.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0024 RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x0025 RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x0026 RESERVED	R	0	0	0	0	0	0	0	0
	W								
0x0027 RESERVED	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 13-2. The scalable PWM Register Summary (Sheet 4 of 4)

1 The related bit is available only if corresponding channel exists.

2 The register is available only if corresponding channel exists.

13.3.2.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

NOTE

The first PWM cycle after enabling the channel can be irregular.

An exception to this is when channels are concatenated. Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit. In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output lines are disabled.

While in run mode, if all existing PWM channels are disabled (PWMEx–0 = 0), the prescaler counter shuts off for power savings.

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R W	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
Reset	0	0	0	0	0	0	0	0

Figure 13-3. PWM Enable Register (PWME)

Read: Anytime

Write: Anytime

14.4.5.5 LIN Transmit Collision Detection

This module allows to check for collisions on the LIN bus.

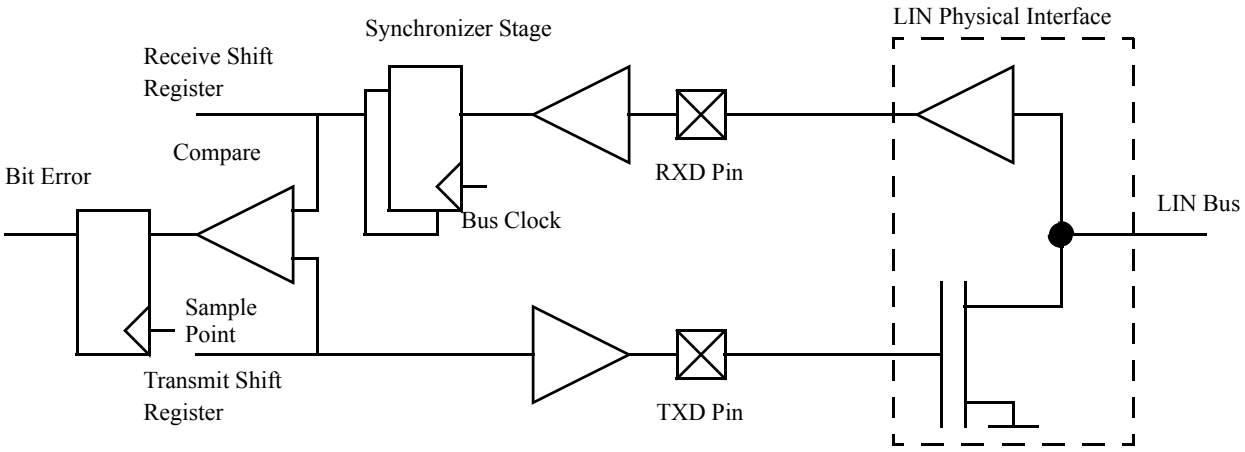


Figure 14-18. Collision Detect Principle

If the bit error circuit is enabled (BERRM[1:0] = 0:1 or = 1:0]), the error detect circuit will compare the transmitted and the received data stream at a point in time and flag any mismatch. The timing checks run when transmitter is active (not idle). As soon as a mismatch between the transmitted data and the received data is detected the following happens:

- The next bit transmitted will have a high level (TXPOL = 0) or low level (TXPOL = 1)
- The transmission is aborted and the byte in transmit buffer is discarded.
- the transmit data register empty and the transmission complete flag will be set
- The bit error interrupt flag, BERRIF, will be set.
- No further transmissions will take place until the BERRIF is cleared.

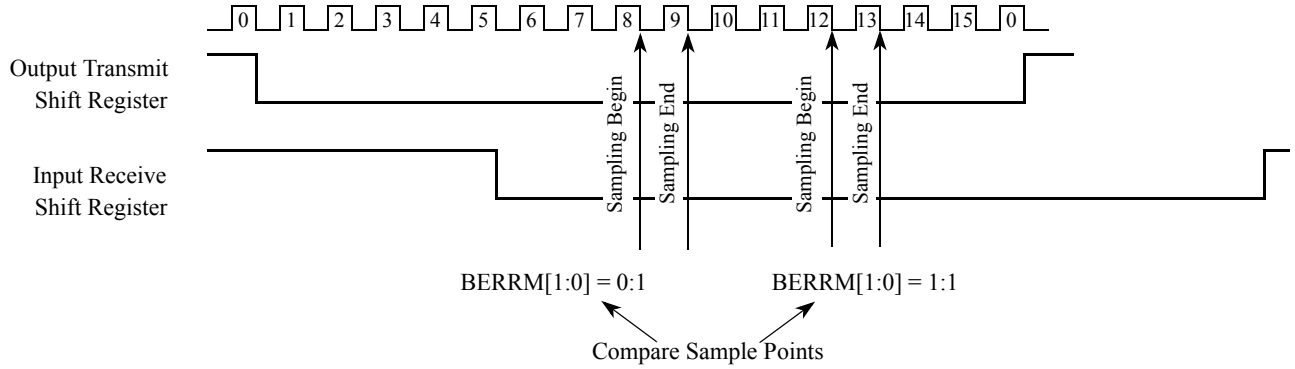


Figure 14-19. Timing Diagram Bit Error Detection

If the bit error detect feature is disabled, the bit error interrupt flag is cleared.

NOTE

The RXPOL and TXPOL bit should be set the same when transmission collision detect feature is enabled, otherwise the bit error interrupt flag may be set incorrectly.

Table 18-11. CANRIER Register Field Descriptions

Field	Description
7 WUPIE[1]	**Wake-Up Interrupt Enable** 0 No interrupt request is generated from this event. 1 A wake-up event causes a Wake-Up interrupt request.
6 CSCIE	**CAN Status Change Interrupt Enable** 0 No interrupt request is generated from this event. 1 A CAN Status Change event causes an error interrupt request.
5-4 RSTATE[1:0]	**Receiver Status Change Enable** — These RSTAT enable bits control the sensitivity level in which receiver state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level the RSTAT flags continue to indicate the actual receiver state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by receiver state changes. 01 Generate CSCIF interrupt only if the receiver enters or leaves “bus-off” state. Discard other receiver state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the receiver enters or leaves “RxErr” or “bus-off”[2] state. Discard other receiver state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
3-2 TSTATE[1:0]	**Transmitter Status Change Enable** — These TSTAT enable bits control the sensitivity level in which transmitter state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level, the TSTAT flags continue to indicate the actual transmitter state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by transmitter state changes. 01 Generate CSCIF interrupt only if the transmitter enters or leaves “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the transmitter enters or leaves “TxErr” or “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
1 OVRIE	**Overrun Interrupt Enable** 0 No interrupt request is generated from this event. 1 An overrun event causes an error interrupt request.
0 RXFIE	**Receiver Full Interrupt Enable** 0 No interrupt request is generated from this event. 1 A receive buffer full (successful message reception) event causes a receiver interrupt request.

1 WUPIE and WUPE (see Section 18.3.2.1, “MSCAN Control Register 0 (CANCTL0)”) must both be enabled if the recovery mechanism from stop or wait is required.

2 Bus-off state is only defined for transmitters by the CAN standard (see Bosch CAN 2.0A/B protocol specification). Because the only possible state change for the transmitter from bus-off to TxOK also forces the receiver to skip its current state to RxOK, the coding of the RXSTAT[1:0] flags define an additional bus-off state for the receiver (see Section 18.3.2.5, “MSCAN Receiver Flag Register (CANRFLG)”).

18.3.2.7 MSCAN Transmitter Flag Register (CANTFLG)

The transmit buffer empty flags each have an associated interrupt enable bit in the CANTIER register.

generates a receive interrupt[1] (see Section 18.4.7.3, “Receive Interrupt”) to the CPU. The user’s receive handler must read the received message from the RxFG and then reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message, which can follow immediately after the IFS field of the CAN frame, is received into the next available RxBG. If the MSCAN receives an invalid message in its RxBG (wrong identifier, transmission errors, etc.) the actual contents of the buffer will be over-written by the next message. The buffer will then not be shifted into the FIFO.

When the MSCAN module is transmitting, the MSCAN receives its own transmitted messages into the background receive buffer, RxBG, but does not shift it into the receiver FIFO, generate a receive interrupt, or acknowledge its own messages on the CAN bus. The exception to this rule is in loopback mode (see Section 18.3.2.2, “MSCAN Control Register 1 (CANCTL1)”) where the MSCAN treats its own messages exactly like all other incoming messages. The MSCAN receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN must be prepared to become a receiver.

An overrun condition occurs when all receive message buffers in the FIFO are filled with correctly received messages with accepted identifiers and another message is correctly received from the CAN bus with an accepted identifier. The latter message is discarded and an error interrupt with overrun indication is generated if enabled (see Section 18.4.7.5, “Error Interrupt”). The MSCAN remains able to transmit messages while the receiver FIFO is being filled, but all incoming messages are discarded. As soon as a receive buffer in the FIFO is available again, new valid messages will be accepted.

18.4.3 Identifier Acceptance Filter

The MSCAN identifier acceptance registers (see Section 18.3.2.12, “MSCAN Identifier Acceptance Control Register (CANIDAC)”) define the acceptable patterns of the standard or extended identifier (ID[10:0] or ID[28:0]). Any of these bits can be marked ‘don’t care’ in the MSCAN identifier mask registers (see Section 18.3.2.18, “MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)”).

A filter hit is indicated to the application software by a set receive buffer full flag (RXF = 1) and three bits in the CANIDAC register (see Section 18.3.2.12, “MSCAN Identifier Acceptance Control Register (CANIDAC)”). These identifier hit flags (IDHIT[2:0]) clearly identify the filter section that caused the acceptance. They simplify the application software’s task to identify the cause of the receiver interrupt. If more than one hit occurs (two or more filters match), the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes:

- Two identifier acceptance filters, each to be applied to:
  - The full 29 bits of the extended identifier and to the following bits of the CAN 2.0B frame:
    - Remote transmission request (RTR)
    - Identifier extension (IDE)
    - Substitute remote request (SRR)
  - The 11 bits of the standard identifier plus the RTR and IDE bits of the CAN 2.0A/B messages. This mode implements two filters for a full length CAN 2.0B compliant extended identifier. Although this mode can be used for standard identifiers, it is recommended to use the four or

1. The receive interrupt occurs only if not masked. A polling scheme can be applied on RXF also.

18.4.5.5 MSCAN Sleep Mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

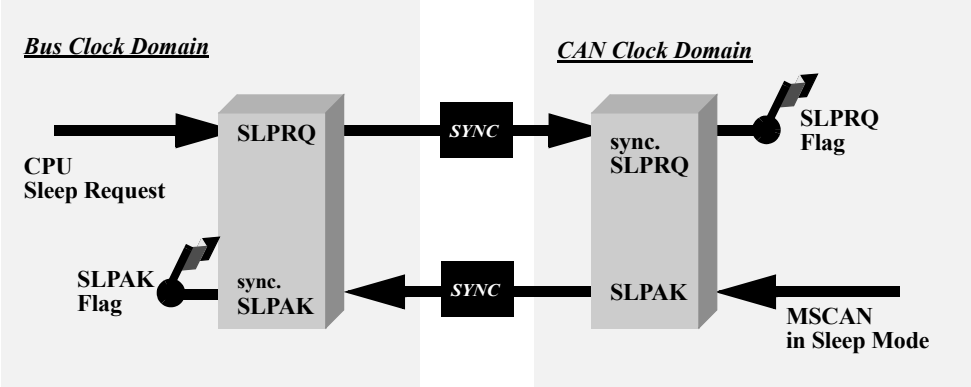


Figure 18-46. Sleep Request / Acknowledge Cycle

NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 18-46). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

negative amplifier input is open. The unbuffered analog voltage from the DAC resistor network is available on the internal connection DACI. For decoding of the control signals see Table 19-7.

19.5.7 Mode “Buffered DAC”

The “Buffered DAC” mode is selected by DACCTL.DACM[2:0] = 0x7. During this is mode the DAC resistor network and the operational amplifier are enabled. The analog output voltage from the DAC resistor network output is buffered by the operational amplifier and is available on the AMP output pin.

The DAC resistor network output is disconnected from the DACU pin. The unbuffered analog voltage from the DAC resistor network is available on the internal connection DACI. For the decoding of the control signals see Table 19-7.

19.5.8 Analog output voltage calculation

The DAC can provide an analog output voltage in two different voltage ranges:

- FVR = 0, reduced voltage range

  The DAC generates an analog output voltage inside the range from 0.1 x (VRH - VRL) + VRL to 0.9 x (VRH-VRL) + VRL with a resolution ((VRH-VRL) x 0.8) / 256, see equation below:

analog output voltage = VOLATGE[7:0] x ((VRH-VRL) x 0.8) / 256) + 0.1 x (VRH-VRL) + VRL ***Eqn. 19-1***

- FVR = 1, full voltage range

  The DAC generates an analog output voltage inside the range from VRL to VRH with a resolution (VRH-VRL) / 256, see equation below:

analog output voltage = VOLTAGE[7:0] x (VRH-VRL) / 256 +VRL ***Eqn. 19-2***

See Table 19-8 for an example for VRL = 0.0 V and VRH = 5.0 V.

Table 19-8. Analog output voltage calculation

FVR	min. voltage	max. voltage	Resolution	Equation
0	0.5V	4.484V	15.625mV	**VOLTAGE[7:0] x (4.0V) / 256) + 0.5V**
1	0.0V	4.980V	19.531mV	**VOLTAGE[7:0] x (5.0V) / 256**

Offset Module Base + 0x0016

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 22-28. Flash Common Command Object 5 High Register (FCCOB5HI)

Offset Module Base + 0x0017

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

Figure 22-29. Flash Common Command Object 5 Low Register (FCCOB5LO)

22.3.2.13.1 FCCOB - NVM Command Mode

NVM command mode uses the FCCOB registers to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 22-26. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. The value written to the FCCOBIX field must reflect the amount of CCOB words loaded for command execution.

Table 22-26 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in <st-blue>Section 22.4.7 Flash Command Description.

Table 22-26. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Register	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	FCCOB0	HI	FCMD[7:0] defining Flash command
		LO	Global address [23:16]
001	FCCOB1	HI	Global address [15:8]
		LO	Global address [7:0]

Table 22-45. Erase All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 22-28)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

22.4.7.7.1 Erase All Pin

The functionality of the Erase All Blocks command is also available in an uncommanded fashion from the *soc_erase_all_req* input pin on the Flash module. Refer to the Reference Manual for information on control of *soc_erase_all_req*.

The erase-all function requires the clock divider register FCLKDIV (see <st-blue>Section 22.3.2.1 Flash Clock Divider Register (FCLKDIV)) to be loaded before invoking this function using *soc_erase_all_req* input pin. Please refer to the Reference Manual for information about the default value of FCLKDIV in case direct writes to register FCLKDIV are not allowed by the time this feature is invoked. If FCLKDIV is not properly set the erase-all operation will not execute and the ACCERR flag in FSTAT register will set. After the execution of the erase-all function the FCLKDIV register will be reset and the value of register FCLKDIV must be loaded before launching any other command afterwards.

Before invoking the erase-all function using the *soc_erase_all_req* pin, the ACCERR and FPVIOL flags in the FSTAT register must be clear. When invoked from *soc_erase_all_req* the erase-all function will erase all P-Flash memory and EEPROM memory space regardless of the protection settings. If the post-erase verify passes, the routine will then release security by setting the SEC field of the FSEC register to the unsecure state (see <st-blue>Section 22.3.2.2 Flash Security Register (FSEC)). The security byte in the Flash Configuration Field will be programmed to the unsecure state (see Table 22-8.). The status of the erase-all request is reflected in the ERSAREQ bit in the FCNFG register (see <st-blue>Section 22.3.2.5 Flash Configuration Register (FCNFG)). The ERSAREQ bit in FCNFG will be cleared once the operation has completed and the normal FSTAT error reporting will be available as described inTable 22-46..

At the end of the erase-all sequence Protection will remain configured as it was before executing the erase-all function. If the application requires programming P-Flash and/or EEPROM after the erase-all function completes, the existing protection limits must be taken into account. If protection needs to be disabled the user may need to reset the system right after completing the erase-all function.

Table 22-46. Erase All Pin Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if command not available in current mode (see Table 22-28)
	MGSTAT1	Set if any errors have been encountered during the erase verify operation, or during the program verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the erase verify operation, or during the program verify operation