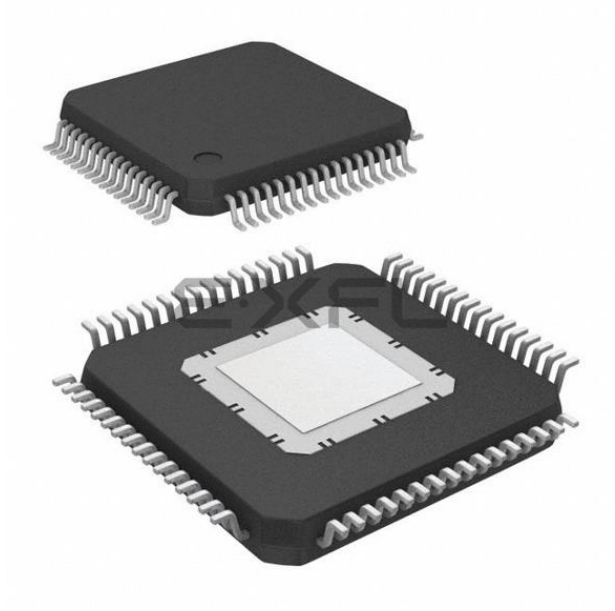


Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-HLQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvml64f2wkhr

trigger through the control loop or can prevent propagation so the static timing of the control cycle and inter-block coherency are not affected by the trigger.

At the end of the conversion sequence the first ADC command from the new sequence is loaded and the ADC*x* waits for the next **trigger_x**. The PTU continues to generate the **trigger_x** events for each trigger time from the list until a new **reload** or **async_reload** occurs.

Before the upcoming **reload** event the CPU:

- reads the ADC results from the buffered Conversion Result List
- clears the conversion complete flag
- services the **reload** by setting new duty cycle values
- sets the PTULDOK bit (corresponding to **glb_ldok**) to signal the duty cycle coherence

The CPU actions are typically performed in an ISR triggered by the conversion complete flag.

1.13.3.4 Static Timing Fault Handling

The following Faults and/or errors can occur:

- Desaturation error, Overvoltage, Undervoltage, External fault

The application run-time error is handled by the GDU without CPU interaction. Firstly the FETs are disabled and the PMF signals switched to an inactive state. To re-enable the operation first the GDU fault and then PWM fault must be cleared, to automatically re-enable the FET driving at the next PWM boundary.

- PTU reload overrun error

This is an application run-time error caused by the CPU not setting PTULDOK on time. Servicing this type of error is application dependent and may range from a further reload attempt to a total shut down.

- PTU trigger generator reload error, PTU trigger generator error

Since all timing is static, this error should only occur during application debugging. This type of error occurring in a static timing configuration indicates possible data corruption. This can be serviced by a control loop shutdown.

- PTU memory access error, Memory access double bit ECC error

This type of error occurring in an application indicates data corruption. This can be serviced by a control loop shutdown.

- ADC sequence overrun, ADC command overrun, ADC command error

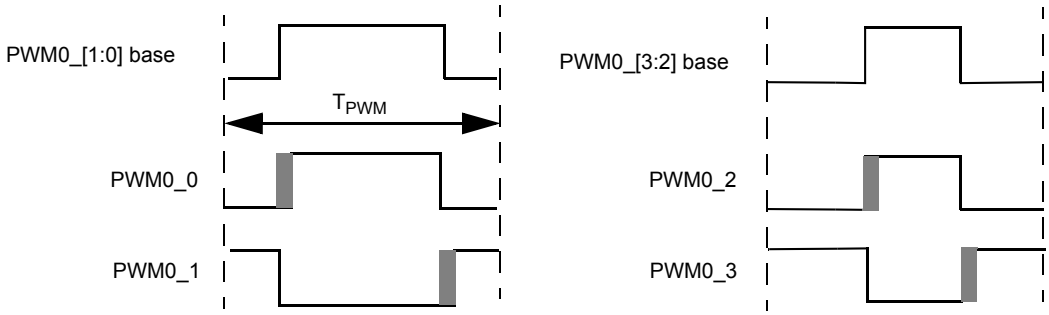
Since all timing is static, this error should only occur during application debugging. This type of error occurring in an application indicates possible data corruption. This can be serviced by a control loop shutdown.

1.13.3.5 Dynamic Timing Operation

The timing frame is dynamic if the following are modified on a cycle by cycle basis:

- PMF - duty cycle value registers (PMF_VAL*x*), modulo registers

Figure 1-13. BDCM Complementary Mode Waveform



Assuming first quadrant operation, forward accelerating operation, the applied voltage at node A must exceed the applied voltage at node B (Figure 1-11). Thus the PWM0_0 duty cycle must exceed the PWM0_2 duty cycle.

The duty cycle of PWM0_0 defines the voltage at the first power stage branch.

The duty cycle of PWM0_2 defines the voltage at the second power stage branch.

Modulating the duty cycle every period using the function F_{PWM} then the duty cycle is expressed as:

PWM0_0 duty-cycle = 0.5 + (0.5 * F_{PWM}); For -1<=F_{PWM} <= 1;

PWM0_2 duty-cycle = 0.5 - (0.5 * F_{PWM})

Table 2-5. Port T Pin Functions and Priorities

Port	Pin Name	ZVMC256	ZVMC128\64	ZVML128/64/32	ZVML31	ZVM32/16	Pin Function & Priority[1]	I/O	Description	Routing Register Bit	Pin Function after Reset
T	PT3	✔	✔	✔	✔	✔	($\overline{SS0}$)	I/O	SPI0 slave select	SPI0RR SPI0SSRR	GPIO
		✔					PWM1_2	O	PMF channel 2	PWM32RR PWMPRR1-0	
		✔	✔	✔	✔	✔	IOC0_3	I/O	TIM0 channel 3	T0IC3RR1-0	
		✔					PWM0_3	O	PWM0 channel 3	—	
		✔	✔	✔	✔	✔	PTT[3]	I/O	General-purpose	—	
	PT2	✔	✔	✔	✔	✔	(SCK0)	I/O	SPI0 serial clock	SPI0RR	
			✔	✔	✔	✔	(PWM1_5)	O	PMF channel 5	PWM54RR PWMPRR1-0	
		✔					(PWM1_0)	O	PMF channel 0	PWM10RR PWMPRR1-0	
		✔	✔	✔	✔	✔	IOC0_2	I/O	TIM0 channel 2	T0C2RR	
		✔					PWM0_7	O	PWM0 channel 7	—	
		✔	✔	✔	✔	✔	PTT[2]	I/O	General-purpose	—	
	PT1		✔	✔	✔	✔	PTURE	O	PTU reload event	—	
		✔	✔	✔	✔	✔	(TXD0)[2]	O	SCI0 transmit	S0L0RR2-0	
				✔	✔	✔	(LPDC0)	O	LPTXD0 direct control by LP0DR[LP0DR1]	S0L0RR2-0	
		✔	✔	✔	✔	✔	(MOSI0)	I/O	SPI0 master out/slave in	SPI0RR	
		✔	✔	✔	✔	✔	(PWM1_4)	O	PMF channel 4	PWM54RR PWMPRR1-0	
		✔	✔	✔	✔	✔	IOC0_1	I/O	TIM0 channel 1	T0C1RR T0IC1RR T0IC1RR0	
		✔	✔	✔	✔	✔	PTT[1]	I/O	General-purpose	—	
	PT0	✔	✔	✔	✔	✔	(RXD0)[2]	I	SCI0 receive	S0L0RR2-0	
		✔	✔	✔	✔	✔	(MISO0)	I/O	SPI0 master in/slave out	SPI0RR	
		✔	✔	✔	✔	✔	(PWM1_3)	O	PMF channel 3	PWM32RR PWMPRR1-0	
		✔	✔	✔	✔	✔	IOC0_0	I/O	TIM0 channel 0	—	
		✔					PWM0_5	O	PWM0 channel 5	—	
		✔	✔	✔	✔	✔	PTT[0]	I/O	General-purpose	—	

1. Signals in parentheses denote alternative module routing pins.
2. Default routing for ZVMC256

Table 2-6. Port S Pin Functions and Priorities

Port	Pin Name	ZVMC256	ZVMC128\64	ZVML128/64/32	ZVML31	ZVM32/16	Pin Function & Priority[1]	I/O	Description	Routing Register Bit	Pin Function after Reset
S	PS5		✔	✔			PDO	O	DBG profiling data output	—	GPIO
			✔	✔	✔	✔	$\overline{SS0}$	I/O	SPI0 slave select	SPI0RR SPI0SSRR	
			✔	✔	✔	✔	PTS[5]/ KWS[5]	I/O	General-purpose; with interrupt and wakeup	—	
	PS4		✔	✔			PDOCLK	O	DBG profiling clock	—	
			✔	✔	✔	✔	SCK0	I/O	SPI0 serial clock	SPI0RR	
			✔	✔	✔	✔	PTS[4]/ KWS[4]	I/O	General-purpose; with interrupt and wakeup	—	
	PS3	✔	✔	✔	✔	✔	MOSI0	I/O	SPI0 master out/slave in	SPI0RR	
		✔					IOC1_1	I/O	TIM1 channel 1	—	
		✔					(CPTXD0)	I	CANPHY0 transmit input	M0C0RR2-0	
		✔	✔	✔	✔	✔	(TXD1)	O	SCI1 transmit	SCI1RR	
		✔	✔	✔	✔	✔	DBGEEV	I	DBG external event	—	
		✔	✔	✔	✔	✔	PTS[3]/ KWS[3]	I/O	General-purpose; with interrupt and wakeup	—	
	PS2	✔	✔	✔	✔	✔	MISO0	I/O	SPI0 master in/slave out	SPI0RR	
		✔					IOC1_0	I/O	TIM1 channel 0	—	
		✔					(CPRXD0)	O	CANPHY0 receive output	M0C0RR2-0	
		✔	✔	✔	✔	✔	(RXD1)	I	SCI1 receive	SCI1RR	
		✔	✔	✔	✔	✔	PTS[2]/ KWS[2]	I/O	General-purpose; with interrupt and wakeup	—	
	PS1	✔					SCK0	I/O	SPI0 serial clock	SPI0RR	
		✔	✔	✔	✔	✔	PTUT1	O	PTU trigger 1	—	
		✔			✔	✔	(IOC0_2)	I/O	TIM0 channel 2	T0C2RR	
				✔	✔	✔	(LPTXD0)	I	LINPHY0/HVPHY0 transmit input	S0L0RR2-0	
		✔					(CPDR1)	O	CANPHY0 direct control output CP0DR[CPDR1]	M0C0RR2-0	
		✔	✔	✔			TXCAN0[2]	O	MSCAN0 transmit	M0C0RR2-0	
		✔	✔	✔	✔	✔	TXD1	O	SCI1 transmit	SCI1RR	
		✔	✔	✔	✔	✔	PTS[1]/ KWS[1]	I/O	General-purpose; with interrupt and wakeup	—	

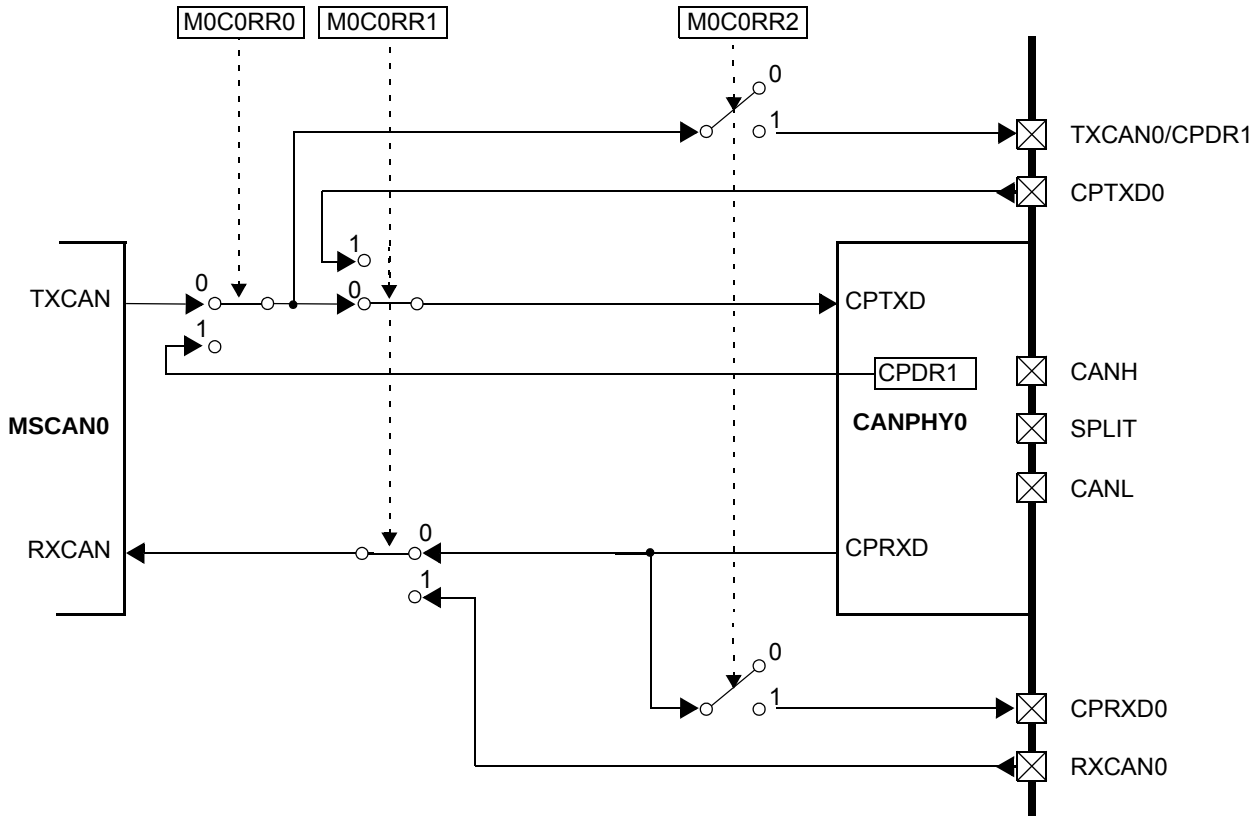


Figure 2-4. CAN Routing Options Illustration

.

Table 2-12. Preferred Interface Configurations

M0C0RR[2:0]	Description
000	Default setting: MSCAN connects to CANPHY, interface internal only
001	Direct control setting: CP0DR[CPDR1] connects to CPTXD, interface internal only
100	Probe setting: MSCAN connects to CANPHY, interface visible on 2 external pins
110	Conformance test setting: Interface opened and all 4 signals routed externally

NOTE

For standalone usage of MSCAN0 on external pins set M0C0RR[2:0]=0b110 and disable CANPHY0 (CPCR[CPE]=0). This releases the CANPHY0 associated pins to other shared functions.

The DBG module provides on-chip breakpoints and trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The DBG module is optimized for the S12Z architecture and allows debugging of CPU module operations.

Typically the DBG module is used in conjunction with the BDC module, whereby the user configures the DBG module for a debugging session over the BDC interface. Once configured the DBG module is armed and the device leaves active BDM returning control to the user program, which is then monitored by the DBG module. Alternatively the DBG module can be configured over a serial interface using SWI routines.

6.1.1 Glossary

Table 6-3. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
PC	Program Counter
BDM	Background Debug Mode. In this mode CPU application code execution is halted. Execution of BDC “active BDM” commands is possible.
BDC	Background Debug Controller
WORD	16-bit data entity
Data Line	64-bit data entity
CPU	S12Z CPU module
Trigger	A trace buffer input that triggers tracing start, end or mid point

6.1.2 Overview

The comparators monitor the bus activity of the CPU. A single comparator match or a series of matches can trigger bus tracing and/or generate breakpoints. A state sequencer determines if the correct series of matches occurs. Similarly an external event can trigger bus tracing and/or generate breakpoints.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads.

6.1.3 Features

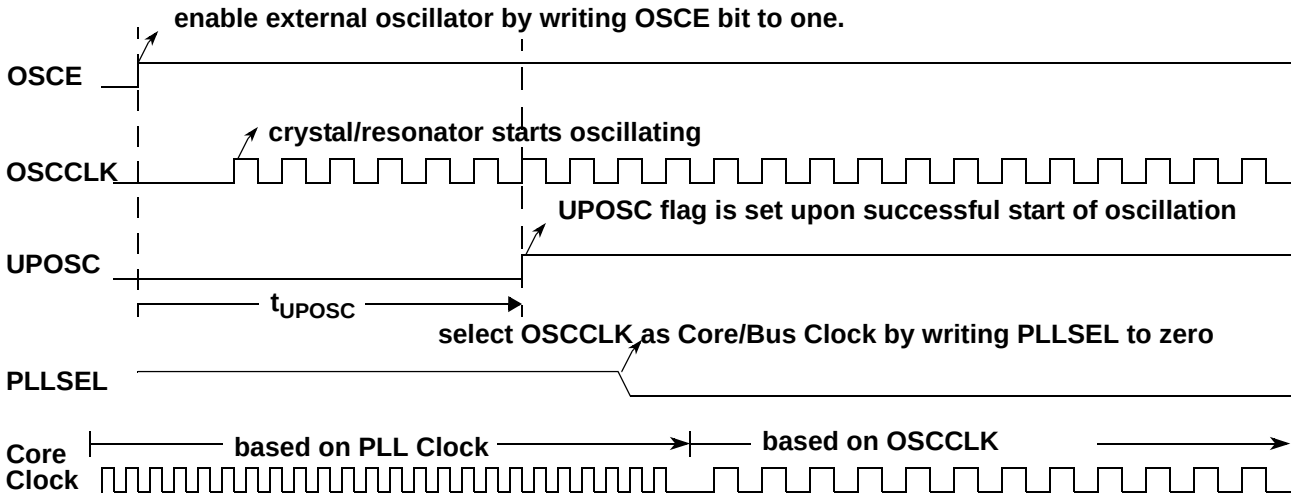
- Four comparators (A, B, C, and D)
  - — Comparators A and C compare the full address bus and full 32-bit data bus
  - — Comparators A and C feature a data bus mask register
  - — Comparators B and D compare the full address bus only
  - — Each comparator can be configured to monitor PC addresses or addresses of data accesses
  - — Each comparator can select either read or write access cycles
  - — Comparator matches can force state sequencer state transitions

8.4.5 External Oscillator

8.4.5.1 Enabling the External Oscillator

An example of how to use the oscillator as source of the Bus Clock is shown in Figure 8-44.

Figure 8-44. Enabling the external oscillator



NOTE

In principle, the MCU could stay in Wait Mode for a shorter period of time than the ADC needs to abort an ongoing conversion (range of μμμμs). Therefore in case a Sequence Abort Event is issued automatically due to MCU Wait Mode request a following Restart Event after exit from MCU Wait Mode can not be executed before ADC has finished this Sequence Abort Event. The Restart Event is detected but it is pending.
This applies in case MCU Wait Mode is exited before ADC has finished the Sequence Abort Event and a Restart Event is issued immediately after exit from MCU Wait Mode. Bit READY can be used by software to detect when the Restart Event can be issued without latency time in processing the event (see also Figure 9-1).

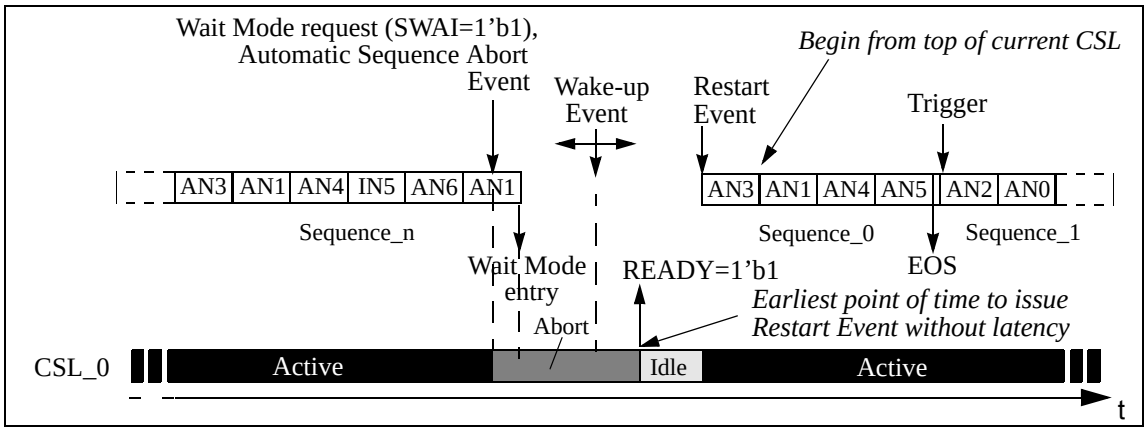


Figure 9-1. Conversion Flow Control Diagram - Wait Mode (SWAI=1'b1, AUT_RSTA=1'b0)

- **MCU Freeze Mode**
  Depending on the ADC Freeze Mode configuration bit FRZ_MOD, the ADC either continues conversion in Freeze Mode or freezes conversion at next conversion boundary before the MCU Freeze Mode is entered. After exit from MCU Freeze Mode with previously frozen conversion sequence the ADC continues the conversion with the next conversion command and all ADC interrupt flags are unchanged during MCU Freeze Mode.

9.5.2.4 ADC Timing Register (ADCTIM)

Module Base + 0x0003

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

= Unimplemented or Reserved

Figure 9-7. ADC Timing Register (ADCTIM))

Read: Anytime

Write: These bits are writable if bit ADC_EN is clear or bit SMOD_ACC is set

Table 9-7. ADCTIM Field Descriptions

Field	Description
6-0 PRS[6:0]	**ADC Clock Prescaler** — These 7bits are the binary prescaler value PRS. The ADC conversion clock frequency is calculated as follows: $f_{ATDCLK} = \frac{f_{BUS}}{2 \times (PRS + 1)}$ Refer to Device Specification for allowed frequency range of f_{ATDCLK}.

10.3.2.1 BATS Module Enable Register (BATE)

Module Base + 0x0000 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	BVHS	BVLS[1:0]		BSUAE	BSUSE	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-3. BATS Module Enable Register (BATE)

1. Read: Anytime
   Write: Anytime

Table 10-2. BATE Field Description

Field	Description
6 BVHS	**BATS Voltage High Select** — This bit selects the trigger level for the Voltage Level High Condition (BVHC). 0 Voltage level V_{HBI1} is selected 1 Voltage level V_{HBI2} is selected
5:4 BVLS[1:0]	**BATS Voltage Low Select** — This bit selects the trigger level for the Voltage Level Low Condition (BVLC). 00 Voltage level V_{LBI1} is selected 01 Voltage level V_{LBI2} is selected 10 Voltage level V_{LBI3} is selected 11 Voltage level V_{LBI4} is selected
3 BSUAE	BATS VSUP ADC Connection Enable — This bit connects the VSUP pin through the resistor chain to ground and connects the ADC channel to the divided down voltage. 0 ADC Channel is disconnected 1 ADC Channel is connected
2 BSUSE	**BATS VSUP Level Sense Enable** — This bit connects the VSUP pin through the resistor chain to ground and enables the Voltage Level Sense features measuring BVLC and BVHC. 0 Level Sense features disabled 1 Level Sense features enabled

NOTE

When opening the resistors path to ground by changing BSUSE or BSUAE then for a time T_{EN_UNC} + two bus cycles the measured value is invalid. This is to let internal nodes be charged to correct value. BVHIE, BVLIE might be cleared for this time period to avoid false interrupts.

13.4.5.5 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.

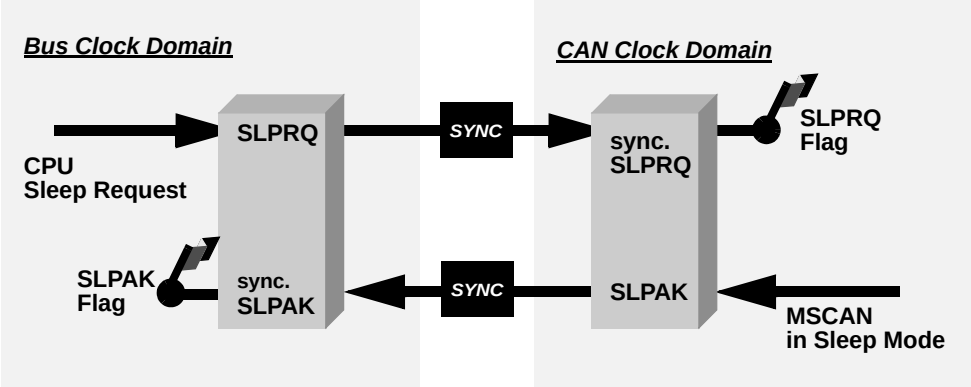


Figure 13-46. Sleep Request / Acknowledge Cycle

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 13-46). 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.

- Three complementary pairs and zero independent PWM outputs
- Zero complementary pairs and six independent PWM outputs

All PWM outputs can be generated from the same counter, or each pair can have its own counter for three independent PWM frequencies. Complementary operation permits programmable deadtime insertion, distortion correction through current sensing by software, and separate top and bottom output polarity control. Each counter value is programmable to support a continuously variable PWM frequency. Both edge- and center-aligned synchronous pulse width-control and full range modulation from 0 percent to 100 percent, are supported. The PMF is capable of controlling most motor types: AC induction motors (ACIM), both brushless (BLDC) and brush DC motors (BDC), switched (SRM) and variable reluctance motors (VRM), and stepper motors.

15.1.1 Features

- Three complementary PWM signal pairs, or six independent PWM signals
- Edge-aligned or center-aligned mode
- Features of complementary channel operation:
  - — Deadtime insertion
  - — Separate top and bottom pulse width correction via current status inputs or software
  - — Three variants of PWM output:
    - – Asymmetric in center-aligned mode
    - – Variable edge placement in edge-aligned mode
    - – Double switching in center-aligned mode
- Three 15-bit counters based on core clock
- Separate top and bottom polarity control
- Half-cycle reload capability
- Integral reload rates from 1 to 16
- Programmable fault protection
- Link to timer output compare for 6-step BLDC commutation support with optional counter restart Reload overrun interrupt
- PWM compare output polarity control Software-controlled PWM outputs, complementary or independent

15.1.2 Modes of Operation

Care must be exercised when using this module in the modes listed in Table 15-4. Some applications require regular software updates for proper operation. Failure to do so could result in destroying the hardware setup. Because of this, PWM outputs are placed in their inactive states in STOP mode, and optionally under WAIT and FREEZE modes. PWM outputs will be reactivated (assuming they were active to begin with) when these modes are exited.

15.2 Signal Descriptions

If the signals are not used exclusively internally, the PMF has external pins named PWM0–5, FAULT0–5, and $\overline{IS0}$–$\overline{IS2}$. Refer to device overview section.

15.2.1 PWM0–PWM5 Pins

PWM0–PWM5 are the output signals of the six PWM channels.

NOTE

On MCUs with an integrated gate drive unit the PWM outputs are connected internally to the GDU inputs. In these cases the PWM signals may optionally be available on pins for monitoring purposes. Refer to the device overview section for routing options and pin locations.

15.2.2 FAULT0–FAULT5 Pins

FAULT0–FAULT5 are input signals for disabling selected PWM outputs (FAULT0-3) or drive the outputs to a configurable active/inactive state (FAULT4-5).

NOTE

On MCUs with an integrated gate drive unit (GDU) either one or more FAULT inputs may be connected internally or/and available on an external pin. Refer to the device overview section for availability and pin locations.

15.2.3 $\overline{IS0}$–$\overline{IS2}$ Pins

$\overline{IS0}$–$\overline{IS2}$ are current status signals for top/bottom pulse width correction in complementary channel operation while deadtime is asserted.

NOTE

Refer to the device overview section for signal availability on pins.

15.2.4 Global Load OK Signal — glb_ldok

This device-internal PMF input signal is connected to the global load OK bit at integration level. For each of the three PWM generator time bases the use of the global load OK input can be enabled individually (GLDOKA,B,C).

15.2.5 Commutation Event Signal — async_event

This device-internal PMF input signal is connected to the source of the asynchronous event generator (preferably timer output compare channel) at integration level.

The commutation event input must be enabled to take effect (ENCE=1). When this bit is set the PMFOUTC, PMFOUT, and MSKx registers switch from non-buffered to async_event triggered double

15.3.2.17 PMF Internal Correction Control Register (PMFICCTL)

Address: Module Base + 0x001E

Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	PECC	PECB	PECA	ICCC	ICCB	ICCA
W								
Reset	0	0	0	0	0	0	0	0

Figure 15-20. PMF Internal Correction Control Register (PMFICCTL)

1. Read: Anytime
   Write: Anytime

This register is used to control PWM pulse generation for various applications, such as a power-supply phase-shifting application.

ICC*x* bits apply only in center-aligned operation during complementary mode. These control bits determine whether values set in the IPOL*x* bits control or the whether PWM count direction controls which PWM value register is used.

NOTE

The ICC*x* bits are buffered. The value written does not take effect until the next PWM load cycle begins regardless of the state of the LDOK bit or global load OK. Reading ICC*x* returns the value in a buffer and not necessarily the value the PWM generator is currently using.

The PEC*x* bits apply in edge-aligned and center-aligned operation during complementary mode. Setting the PEC*x* bits overrides the ICC*x* settings. This allows the PWM pulses generated by both the odd and even PWM value registers to be ANDed together prior to the complementary logic and deadtime insertion.

NOTE

The PEC*x* bits are buffered. The value written does not take effect until the related LDOK bit or global load OK is set and the next PWM load cycle begins. Reading PEC*n* returns the value in a buffer and not necessarily the value the PWM generator is currently using.

Figure 15-21. PMF Internal Correction Control Register (PMFICCTL) Descriptions

Field	Description
5 PECC	Pulse Edge Control — This bit controls PWM4/PWM5 pair. 0 Normal operation 1 Allow one of PMFVAL4 and PMFVAL5 to activate the PWM pulse and the other to deactivate the pulse
4 PECB	Pulse Edge Control — This bit controls PWM2/PWM3 pair. 0 Normal operation 1 Allow one of PMFVAL2 and PMFVAL3 to activate the PWM pulse and the other to deactivate the pulse
3 PECA	Pulse Edge Control — This bit controls PWM0/PWM1 pair. 0 Normal operation 1 Allow one of PMFVAL0 and PMFVAL1 to activate the PWM pulse and the other to deactivate the pulse

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

17.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After $2n^1$ (last) SCK edges:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Figure 17-12 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

1. n depends on the selected transfer width, please refer to Section 17.3.2.2, "SPI Control Register 2 (SPICR2)

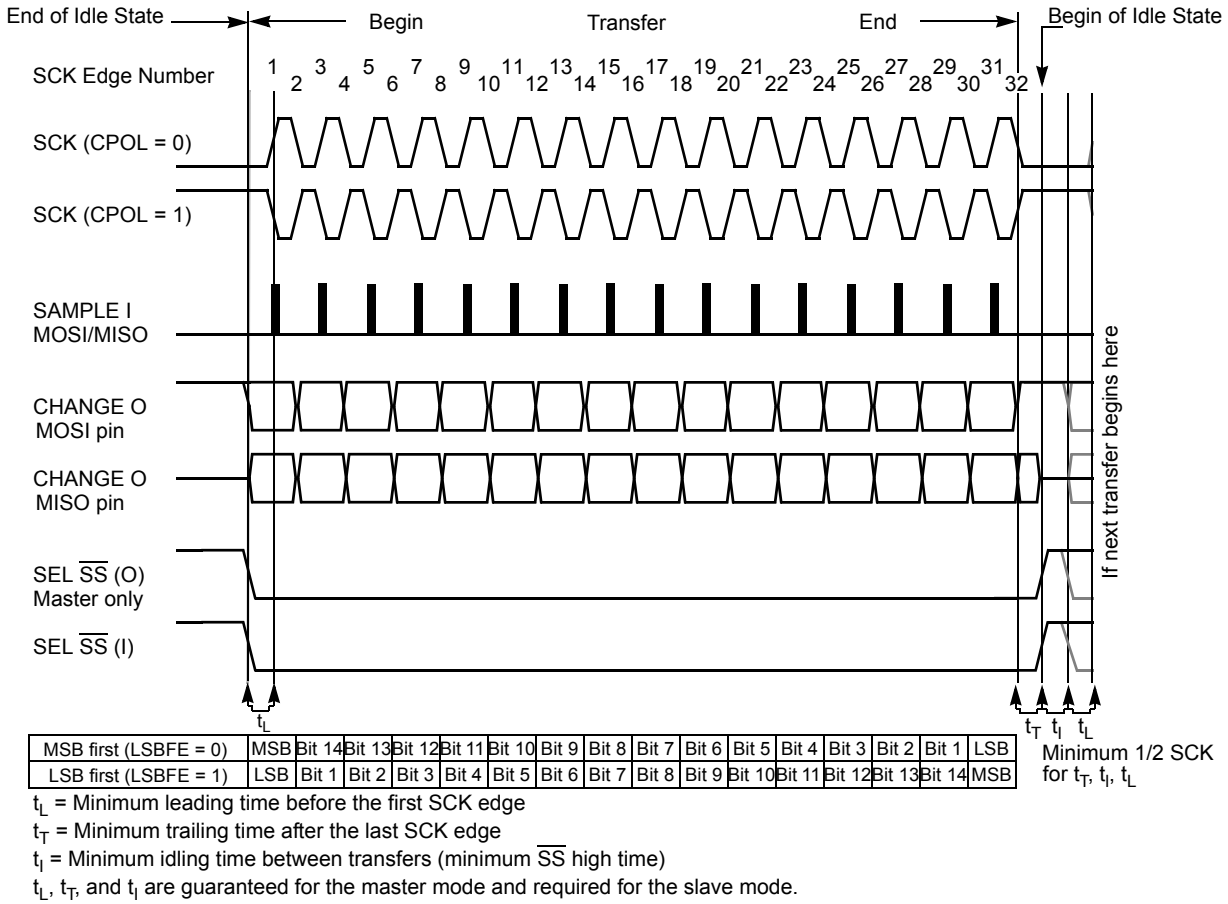


Figure 17-13. SPI Clock Format 0 (CPHA = 0), with 16-Bit Transfer Width selected (XFRW = 1)

In slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

17.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n[1]-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

1. n depends on the selected transfer width, please refer to Section 17.3.2.2, “SPI Control Register 2 (SPICR2)

2. Normal Mode

   The full functionality is available. Both receiver and transmitter are enabled.

3. Receive Only Mode

   The transmitter is disabled and the receiver is running in full performance mode.

4. Standby Mode

   The transmitter of the LIN/HV Physical Layer is disabled. If the wake-up feature is enabled, the internal pullup resistor can be selected (330 kΩ or 34 kΩ). The receiver enters a low power mode and optionally it can pass wake-up events to the Serial Communication Interface (SCI). If the wake-up feature is enabled and if the LIN Bus pin is driven with a dominant level longer than t_{WUFR} followed by a rising edge, the LIN/HV Physical Layer sends a wake-up pulse to the SCI, which requests a wake-up interrupt. (This feature is only available if the LIN/HV Physical Layer is routed to the SCI).

19.1.3 Block Diagram

Figure 19-1 shows the block diagram of the LIN/HV Physical Layer. The module consists of a receiver with wake-up control, a transmitter with slope and timeout control, a current sensor with overcurrent protection as well as a registers control block.

If LPWUE is not set, no wake up feature is available and the standby mode has the same electrical properties as the shutdown mode. This allows a low-power consumption of the device in stop mode if the wake-up feature is not needed.

If LPWUE is set, the receiver is able to pass wake-up events to the SCI (Serial Communication Interface). If the LIN/HV Physical Layer receives a dominant level longer than t_{WUFR} followed by a rising edge, it sends a pulse to the SCI which can generate a wake-up interrupt.

Once the device exits stop mode, the LIN/HV Physical Layer returns to normal or receive only mode depending on the status of the RXONLY bit.

NOTE

Since the wake-up interrupt is requested by the SCI, the wake-up feature is not available if LPRxD is not connected to the SCI..

The internal pullup resistor is selectable only if LPWUE = 1 (wake-up enabled). If LPWUE = 0, the internal pullup resistor is not selectable and remains at 330 kΩ regardless of the state of the LPPUE bit.

If LPWUE = 1, selecting the 330 kΩ pullup resistor (LPPUE = 0) reduces the current consumption in standby mode.

NOTE

The use of the LIN wake-up feature in combination with other non-LIN device wake-up features (like a periodic time interrupt) must be handled with care.

If the device leaves stop mode while the LIN bus is dominant, the LIN/HV Physical Layer returns to normal or receive only mode and the LPRxD signal is re-routed to the RxD pin of the SCI and triggers the edge detection interrupt (if the interrupt's priority of the hardware that awakes the MCU is less than the priority of the SCI interrupt, then the SCI interrupt will execute first). It is up to the software to decide what to do in this case because the LIN/HV Physical Layer may not determine whether it was a valid wake-up pulse.

Table 20-5. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x1F_C000 – 0x1F_C007	8	Reserved
0x1F_C008 – 0x1F_C0B5	174	Reserved
0x1F_C0B6 – 0x1F_C0B7	2	Version ID[1]
0x1F_C0B8 – 0x1F_C0BF	8	Reserved
0x1F_C0C0 – 0x1F_C0FF	64	Program Once Field Refer to Section 20.4.7.6, “Program Once Command”

1. Used to track firmware patch versions, see Section 20.4.2

Table 20-6. Memory Controller Resource Fields (NVM Resource Area[1])

Global Address	Size (Bytes)	Description
0x1F_4000 – 0x1F_41FF	512	Reserved
0x1F_4200 – 0x1F_7FFF	15,872	Reserved
0x1F_8000 – 0x1F_97FF	6,144	Reserved
0x1F_9800 – 0x1F_BFFF	10,240	Reserved
0x1F_C000 – 0x1F_C0FF	256	P-Flash IFR (see Table 20-5)
0x1F_C100 – 0x1F_C1FF	256	Reserved.
0x1F_C200 – 0x1F_FFFF	15,872	Reserved.

1. See Section 20.4.4 for NVM Resources Area description.

20.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.