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	50MHz
Connectivity	LINbus, SCI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP Exposed Pad
Supplier Device Package	48-LQFP-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvml31f1mkf

Chapter 11
Timer Module (TIM16B4CV3) Block Description

Chapter 12
Timer Module (TIM16B2CV3) Block Description

Chapter 13
Scalable Controller Area Network (S12MSCANV3)

- Configurable wake-up pulse filtering
- Over-current shutdown for CANH and CANL
- Voltage monitoring on CANH and CANL
- CPTXD-dominant timeout feature monitoring the CPTXD signal
- Fulfills the OEM “Hardware Requirements for (LIN,) CAN (and FlexRay) Interfaces in Automotive Applications” v1.3

1.4.20 Pulse Width Modulation Module (PWM) (ZVMC256 only)

- Configurable as 8 channels x 8-bit or 4 channels x 16-bit
- Programmable period and duty cycle per channel
- Center-aligned or edge-aligned outputs
- Programmable clock select logic with a wide range of frequencies

1.7.2.3 MODC — Mode C Signal

The MODC signal is used as an MCU operating mode select during reset. The state of this signal is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The signal has an internal pull-up device.

1.7.2.4 PAD[15:0] / KWAD[15:0] — Port AD, Input Pins of ADC

These are general-purpose input or output signals. The signals can be configured on per signal basis as interrupt inputs with wake-up capability (KWAD). These signals can have a pull-up or pull-down device selected and enabled on per signal basis. During and out of reset the pull devices are disabled.

1.7.2.5 PE[1:0] — Port E I/O Signals

PE[1:0] are general-purpose input or output signals. The signals can have a pull-down device, enabled by on a per pin basis. Out of reset the pull-down devices are enabled.

1.7.2.6 PL[0] — Port L Input Signal

PL[0] is a high voltage input port. The port can be configured as interrupt input with wake-up capability (KWL[0]). The input voltage is also scaled and mapped to an internal ADC channel.

1.7.2.7 PP[2:0] / KWP[2:0] — Port P I/O Signals

PP[2:0] are general-purpose input or output signals. The signals can be configured on per signal basis as interrupt inputs with wake-up capability (KWP[2:0]). They can have a pull-up or pull-down device selected and enabled on per signal basis. During and out of reset the pull devices are disabled.

1.7.2.8 PS[5:0] / KWS[5:0] — Port S I/O Signals

PS[5:0] are general-purpose input or output signals. The signals can be configured on per signal basis as interrupt inputs with wake-up capability (KWS[5:0]). They can have a pull-up or pull-down device selected and enabled on per signal basis. During and out of reset the pull-up devices are enabled.

1.7.2.9 PT[3:0] — Port T I/O Signals

PT[3:0] are general-purpose input or output signals. They can have a pull-up or pull-down device selected and enabled on per signal basis. During and out of reset the pull devices are disabled.

1.7.2.10 AN0_[7:0], AN1_[7:0]— ADC Input Signals

These are the analog inputs of the Analog-to-Digital Converters. These are mapped to PAD port pins. The number of analog input channels connected to PAD port pins is package option dependent.

1.7.2.11 VRH0_[2:0], VRL0_[1:0] — ADC0 Reference Signals

VRH0_[2:0] and VRL0_[1:0] are the reference voltage signals for the analog-to-digital converter ADC0.

1.7.2.25.4 LP0DR1

This is the LIN (or HV physical interface) LP0DR1 register bit, visible at the designated pin for debug purposes.

1.7.2.25.5 LGND — LINPHY Ground Pin

On S12ZVM(L) parts LGND is the ground pin for the LIN physical layer LINPHY. This signal must be connected to board ground, even if the LINPHY is not used.

On S12ZVM32 and S12ZVM16 parts this the ground pin for the HV physical interface. It must be connected to board ground even when the HV physical interface is not used.

1.7.2.26 CAN Physical Layer Signals (ZVMC256 Only)

1.7.2.26.1 CANH0 — CAN Bus High Pin0

The CANH0 signal either connects directly to CAN bus high line or through an optional external common mode choke.

1.7.2.26.2 CANL0 — CAN Bus Low Pin0

The CANL0 signal either connects directly to CAN bus low line or through an optional external common mode choke.

1.7.2.26.3 SPLIT0 — CAN Bus Termination Pin0

The SPLIT0 pin can drive a 2.5 V bias for bus termination purpose (CAN bus middle point). Usage of this pin is optional and depends on bus termination strategy for a given bus network.

1.7.2.26.4 CPTXD0

This is the CAN physical layer transmitter input signal.

1.7.2.26.5 CPRXD0

This is the CAN physical layer receiver output signal.

1.7.2.26.6 CPDR0

This is the CAN physical layer direct control output signal.

1.7.2.26.7 BCTLC

BCTLC provides the base current of an external bipolar that supplies an external CAN physical interface. This signal is only available on S12ZVMC versions. If not used BCTLC should be left unconnected.

The exposed pad on the package bottom must be connected to a grounded contact pad on the PCB.

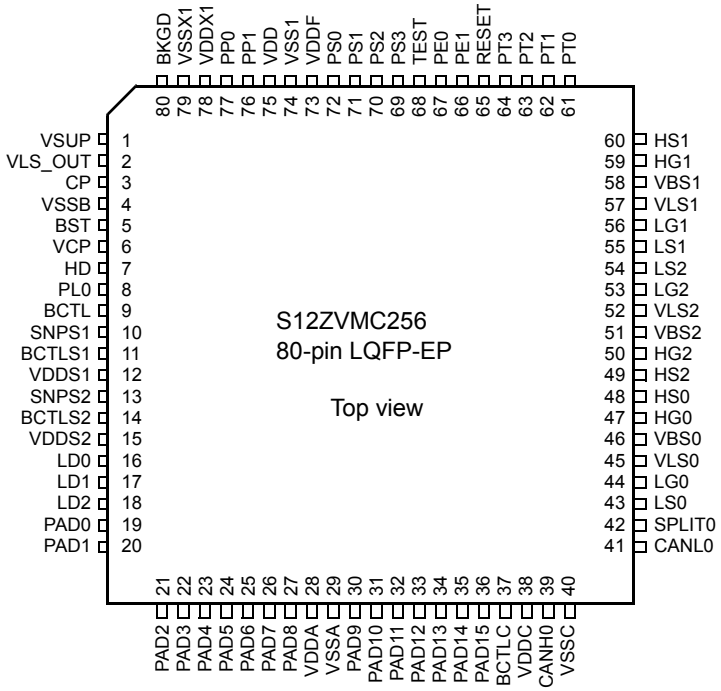
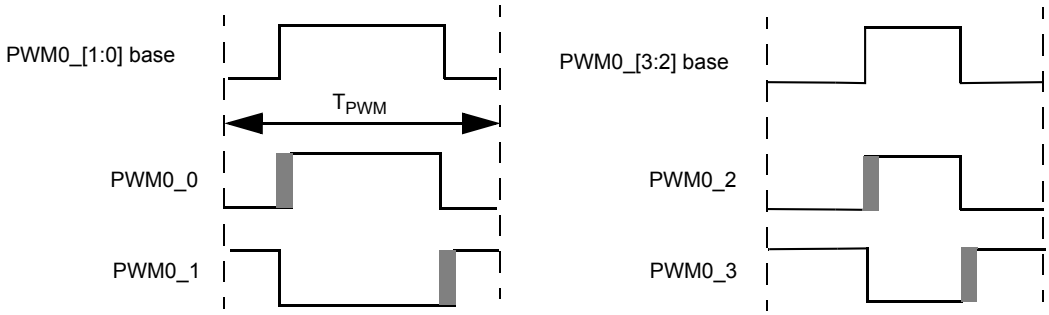


Figure 1-3. S12ZVMC256 80-pin LQFP pin out

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 2-7. Port P 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
P	PP2		✔	✔	✔	✔	(PWM1_2)	O	PMF channel 2	PWM32RR PWMPRR	GPIO
			✔	✔	✔	✔	PTP[2]/ KWP[2]	I/O	General-purpose; with interrupt and wakeup	—	
	PP1	✔	✔	✔	✔	✔	$\overline{\text{IRQ}}$	I	Maskable level- or falling edge-sensitive interrupt	—	
		✔	✔	✔	✔	✔	(PWM1_1)	O	PMF channel 1	PWM10RR PWMPRR	
		✔					PWM0_1	O	PWM0 channel 1	—	
		✔	✔	✔	✔	✔	PTP[1]/ KWP[1]	I/O	General-purpose; with interrupt and wakeup	—	
	PP0	✔	✔	✔	✔	✔	$\overline{\text{XIRQ}}$	I	Non-maskable level-sensitive interrupt[2]	—	
		✔	✔	✔	✔	✔	FAULT5	I	PMF fault	—	
		✔	✔	✔	✔	✔	ECLK	O	Free-running clock	—	
			✔	✔	✔	✔	(PWM1_0)	O	PMF channel 0 with over-current interrupt; high-current capable (20 mA)	PWM10RR PWMPRR	
		✔					(PWM1_5)	O	PMF channel 5 with over-current interrupt; high-current capable (20 mA)	PWM54RR PWMPRR	
		✔	✔	✔	✔	✔	PTP[0]/ KWP[0]/ EVDD1	I/O	General-purpose; with interrupt and wakeup Switchable external power supply output with over-current interrupt; high-current capable (20 mA)	—	

1. Signals in parentheses denote alternative module routing pins.
2. The interrupt is enabled by clearing the X mask bit in the CPU CCR. The pin is forced to input upon first clearing of the X bit and is held in this state until reset. A stop or wait recovery with the X bit set (refer to S12ZCPU reference manual) is not available.

Table 2-8. Port L Pin Functions and Priorities

Port	Pin Name	ZVMC256	ZVMC128\64	ZVML128/64/32	ZVML31	ZVM32/16	Pin Function & Priority	I/O	Description	Routing Register Bit	Pin Function after Reset
L	PL0/	✔					PTIL[0]/ KWL[0]	I	General-purpose high-voltage input (HVI); with interrupt and wakeup; optional ADC link	—	GPI (HVI)

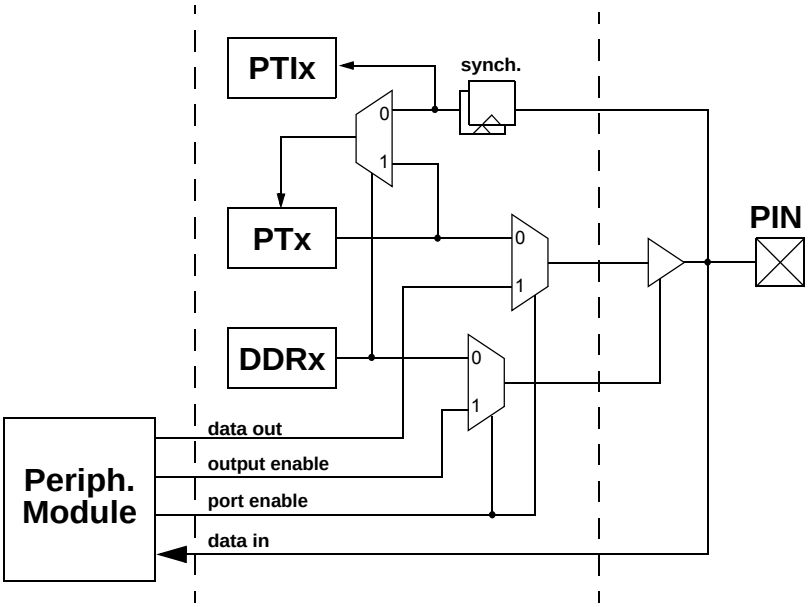


Figure 2-35. Illustration of I/O pin functionality

This section describes the interrupts generated by the PIM and their individual sources. Vector addresses and interrupt priorities are defined at MCU level.

Table 2-41. PIM Interrupt Sources

Module Interrupt Sources	Local Enable
$\overline{\text{XIRQ}}$	None
$\overline{\text{IRQ}}$	IRQCR[IRQEN]
Port AD pin interrupt	PIEADH[PIEADH7-PIEADH0] PIEADL[PIEADL7-PIEADL0]
Port S pin interrupt	PIES[PIES5-PIES0]
Port P pin interrupt	PIEP[PIEP2-PIEP0]
Port L pin interrupt	PIEL[PIEL0]
PP0 over-current interrupt	PIEP[OCIE1]

2.4.3.1 XIRQ, IRQ Interrupts

The $\overline{\text{XIRQ}}$ pin allows requesting non-maskable interrupts after reset initialization. During reset, the X bit in the condition code register is set and any interrupts are masked until software enables them.

The $\overline{\text{IRQ}}$ pin allows requesting asynchronous interrupts. The interrupt input is disabled out of reset. To enable the interrupt the IRQCR[IRQEN] bit must be set and the I bit cleared in the condition code register. The interrupt can be configured for level-sensitive or falling-edge-sensitive triggering. If IRQCR[IRQEN] is cleared while an interrupt is pending, the request will deassert.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x010C-0x010F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0110	DBGACTL	R	0	NDB	INST	0	RW	RWE	reserved	COMPE
		W								
0x0111-0x0114	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0115	DBGAAH	R	DBGAA[23:16]							
		W								
0x0116	DBGAAM	R	DBGAA[15:8]							
		W								
0x0117	DBGAAL	R	DBGAA[7:0]							
		W								
0x0118	DBGAD0	R	Bit 31	30	29	28	27	26	25	Bit 24
		W								
0x0119	DBGAD1	R	Bit 23	22	21	20	19	18	17	Bit 16
		W								
0x011A	DBGAD2	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x011B	DBGAD3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x011C	DBGADM0	R	Bit 31	30	29	28	27	26	25	Bit 24
		W								
0x011D	DBGADM1	R	Bit 23	22	21	20	19	18	17	Bit 16
		W								
0x011E	DBGADM2	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x011F	DBGADM3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0120	DBGBCTL	R	0	0	INST	0	RW	RWE	reserved	COMPE
		W								
0x0121-0x0124	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0125	DBGBAH	R	DBGBA[23:16]							
		W								
0x0126	DBGBAM	R	DBGBA[15:8]							
		W								
0x0127	DBGBAL	R	DBGBA[7:0]							
		W								

Figure 6-2. Quick Reference to DBG Registers

8.7.3 Application Information for PLL and Oscillator Startup

The following C-code example shows a recommended way of setting up the system clock system using the PLL and Oscillator:

```
/* Procedure proposed by to setup PLL and Oscillator */
/* example for OSC = 4 MHz and Bus Clock = 25MHz, That is VCOCLK = 50MHz */

/* Initialize */
/* PLL Clock = 50 MHz, divide by one */
CPMUPOSTDIV = 0x00;

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* configure PLL reference clock (REFCLK) for usage with Oscillator */
/* OSC=4MHz divide by 4 (3+1) = 1MHz, REFCLK range 1MHz to 2 MHz (REFFRQ[1:0] = 00) */
CPMUREFDIV = 0x03;

/* enable external Oscillator, switch PLL reference clock (REFCLK) to OSC */
CPMUOSC = 0x80;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;

/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;

/* put your code to loop and wait for the LOCKIF and OSCIF or */
/* poll CPMUIFLG register until both UPOSC and LOCK status are "1" */
/* that is CPMUIFLG == 0x1B */

/*..............continue to your main code execution here..............*/

/* in case later in your code you want to disable the Oscillator and use the */
/* 1MHz IRCCLK as PLL reference clock */

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* disable OSC and switch PLL reference clock to IRC */
CPMUOSC = 0x00;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;

/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;
```

9.9.9 Triggered Conversion — Single CSL

Applications that require the conversion of one or more groups of different channels in a periodic and timed manner can make use of a configuration in “Trigger Mode” with a single CSL containing a list of sequences. This means the CSL consists of several sequences each separated by an “End of Sequence” command. The last command of the CSL uses the “End Of List” command with wrap to top of CSL and waiting for a Trigger (CMD_SEL[1:0] =2’b11). Hence after the initial Restart Event each sequence can be launched via a Trigger Event and repetition of the CSL can be launched via a Trigger after execution of the “End Of List” command.

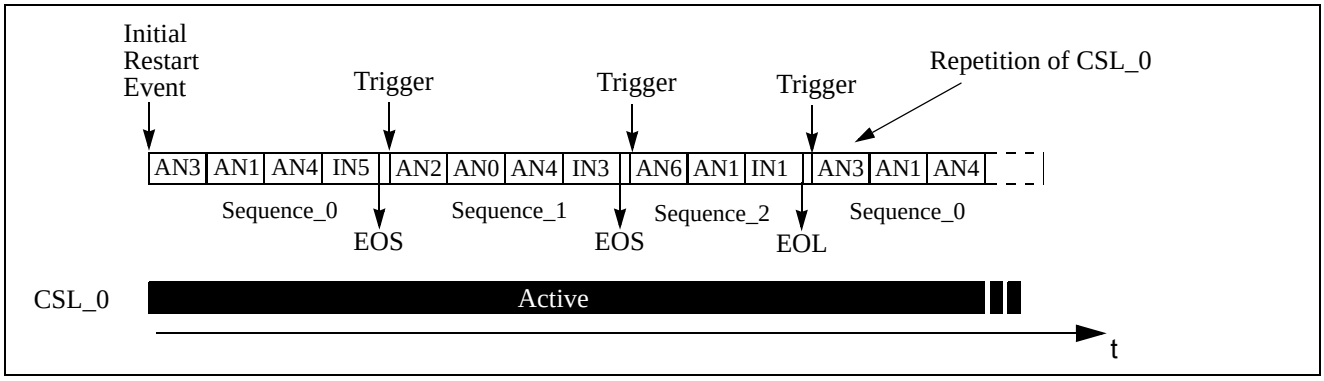


Figure 9-42. Conversion Flow Control Diagram — Triggered Conversion (CSL Repetition)

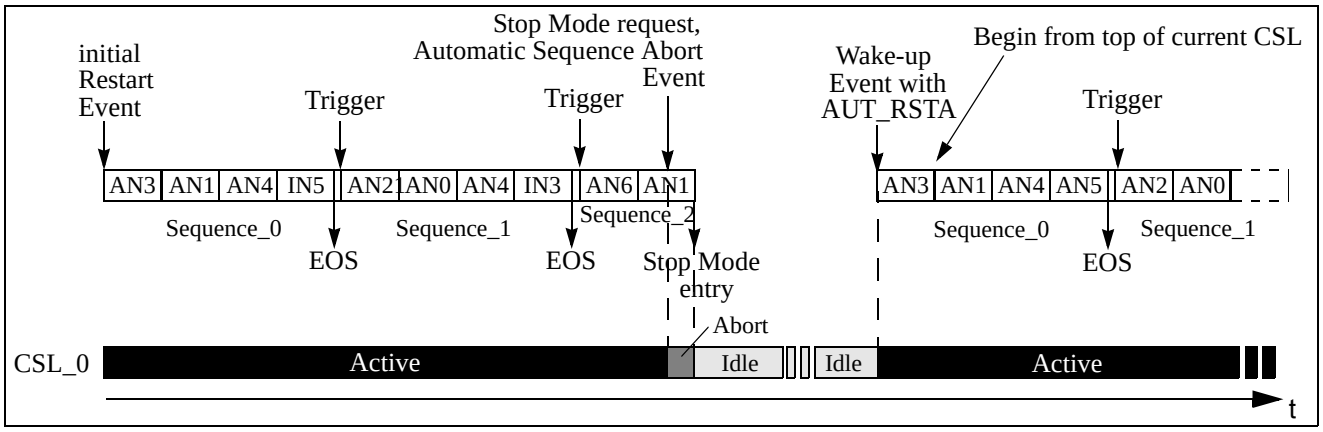


Figure 9-43. Conversion Flow Control Diagram — Triggered Conversion (with Stop Mode)

In case a Low Power Mode is used:
If bit AUT_RSTA is set before Low Power Mode is entered, the conversion continues automatically as soon as a low power mode (Stop Mode or Wait Mode with bit SWAI set) is exited.

The comparator outputs BVLC and BVHC are forced to zero if the comparator is disabled (configuration bit BSUSE is cleared). If the software disables the comparator during a high or low Voltage condition (BVHC or BVLC active), then an additional interrupt is generated. To avoid this behavior the software must disable the interrupt generation before disabling the comparator.

The BATS interrupt vector is named in Table 10-6. Vector addresses and interrupt priorities are defined at MCU level.

The module internal interrupt sources are combined into one module interrupt signal.

Table 10-6. BATS Interrupt Sources

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
BATS Interrupt (BATI)	BATS Voltage Low Condition Interrupt (BVLI)	BVLIE = 1
	BATS Voltage High Condition Interrupt (BVHI)	BVHIE = 1

10.4.2.1 BATS Voltage Low Condition Interrupt (BVLI)

To use the Voltage Low Interrupt the Level Sensing must be enabled (BSUSE =1).

If measured when

a) V_{LBI1} selected with BVLS[1:0] = 0x0
$V_{measure} < V_{LBI1\_A}$ (falling edge) or $V_{measure} < V_{LBI1\_D}$ (rising edge)

or when

b) V_{LBI2} selected with BVLS[1:0] = 0x1 at pin VSUP
$V_{measure} < V_{LBI2\_A}$ (falling edge) or $V_{measure} < V_{LBI2\_D}$ (rising edge)

or when

c) V_{LBI3} selected with BVLS[1:0] = 0x2
$V_{measure} < V_{LBI3\_A}$ (falling edge) or $V_{measure} < V_{LBI3\_D}$ (rising edge)

or when

d) V_{LBI4} selected with BVLS[1:0] = 0x3
$V_{measure} < V_{LBI4\_A}$ (falling edge) or $V_{measure} < V_{LBI4\_D}$ (rising edge)

then BVLC is set. BVLC status bit indicates that a low voltage at pin VSUP is present. The Low Voltage Interrupt flag (BVLIF) is set to 1 when the Voltage Low Condition (BVLC) changes state . The Interrupt flag BVLIF can only be cleared by writing a 1. If the interrupt is enabled by bit BVLIE the module requests an interrupt to MCU (BATI).

10.4.2.2 BATS Voltage High Condition Interrupt (BVHI)

To use the Voltage High Interrupt the Level Sensing must be enabled (BSUSE=1).

11.1.3 Block Diagrams

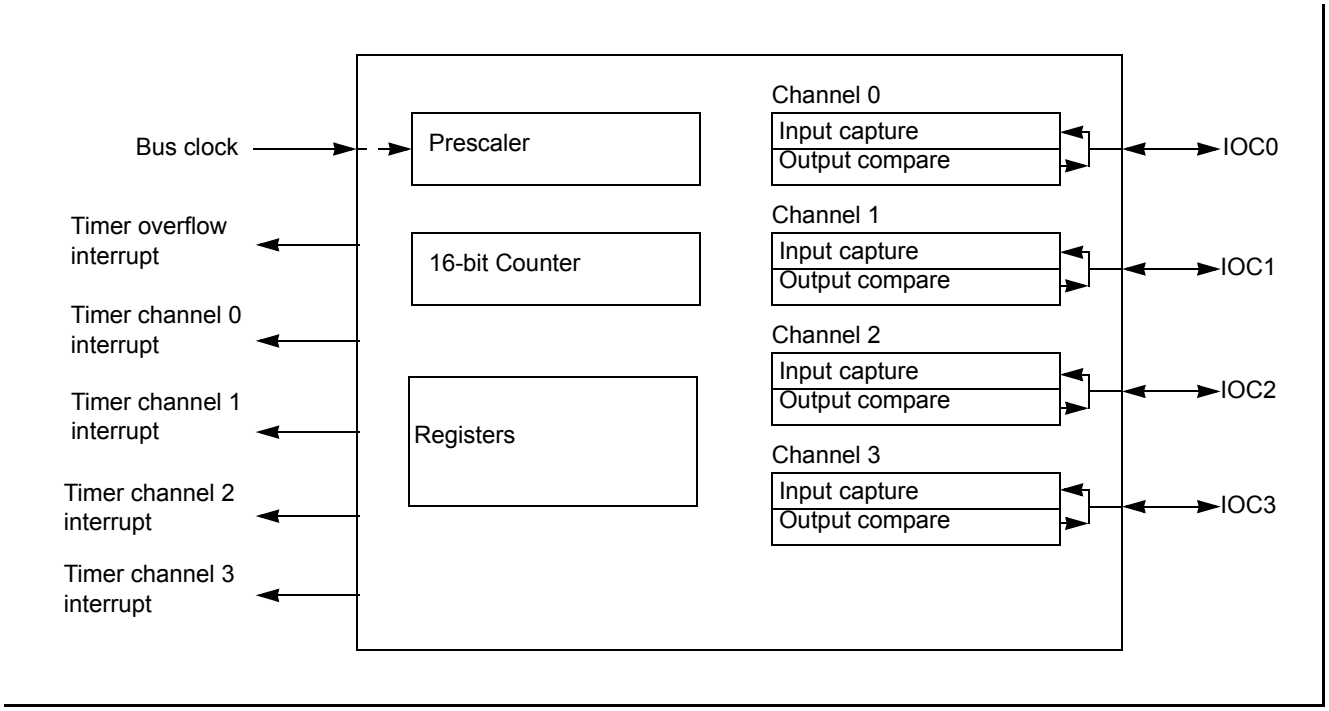


Figure 11-1. TIM16B4CV3 Block Diagram

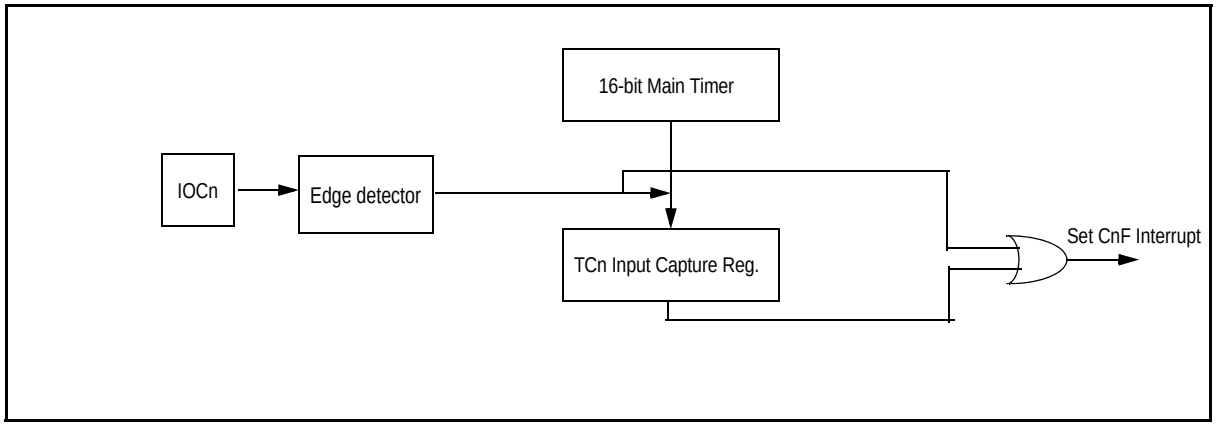


Figure 11-2. Interrupt Flag Setting

Table 13-21. CANIDAR0–CANIDAR3 Register Field Descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

Module Base + 0x0018 to Module Base + 0x001B — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset	0	0	0	0	0	0	0	0

Figure 13-21. MSCAN Identifier Acceptance Registers (Second Bank) — CANIDAR4–CANIDAR7

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 13-22. CANIDAR4–CANIDAR7 Register Field Descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

13.3.2.18 MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)

The identifier mask register specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering. To receive standard identifiers in 32 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1 and CANIDMR5 to “don’t care.” To receive standard identifiers in 16 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1, CANIDMR3, CANIDMR5, and CANIDMR7 to “don’t care.”

Module Base + 0x0014 to Module Base + 0x0017 — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 13-22. MSCAN Identifier Mask Registers (First Bank) — CANIDMR0–CANIDMR3

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 15-18. Software Output Control

OUT*n* Bit	Complementary Channel Operation	Independent Channel Operation
OUT0	1 — PWM0 is active 0 — PWM0 is inactive	1 — PWM0 is active 0 — PWM0 is inactive
OUT1	1 — PWM1 is complement of PWM0 0 — PWM1 is inactive	1 — PWM1 is active 0 — PWM1 is inactive
OUT2	1 — PWM2 is active 0 — PWM2 is inactive	1 — PWM2 is active 0 — PWM2 is inactive
OUT3	1 — PWM3 is complement of PWM2 0 — PWM3 is inactive	1 — PWM3 is active 0 — PWM3 is inactive
OUT4	1 — PWM4 is active 0 — PWM4 is inactive	1 — PWM4 is active 0 — PWM4 is inactive
OUT5	1 — PWM5 is complement of PWM4 0 — PWM5 is inactive	1 — PWM5 is active 0 — PWM5 is inactive

15.3.2.12 PMF Deadtime Sample Register (PMFDTMS)

Address: Module Base + 0x000E

Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	DT5	DT4	DT3	DT2	DT1	DT0
W								
Reset	0	0	0	0	0	0	0	0

Figure 15-15. PMF Deadtime Sample Register (PMFDTMS)

1. Read: Anytime
   Write: Never

Table 15-19. PMFDTMS Field Descriptions

Field	Description
5–0 DT[5:0]	**DT*n* Bits** — The DT*n* bits are grouped in pairs, DT0 and DT1, DT2 and DT3, DT4 and DT5. Each pair reflects the corresponding $\overline{IS}$ input value as sampled at the end of deadtime. *n* is 0, 1, 2, 3, 4 and 5.

15.3.2.13 PMF Correction Control Register (PMFCCTL)

Address: Module Base + 0x000F

Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	ISENS		0	IPOLC	IPOLB	IPOLA
W								
Reset	0	0	0	0	0	0	0	0

Figure 15-16. PMF Correction Control Register (PMFCCTL)

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

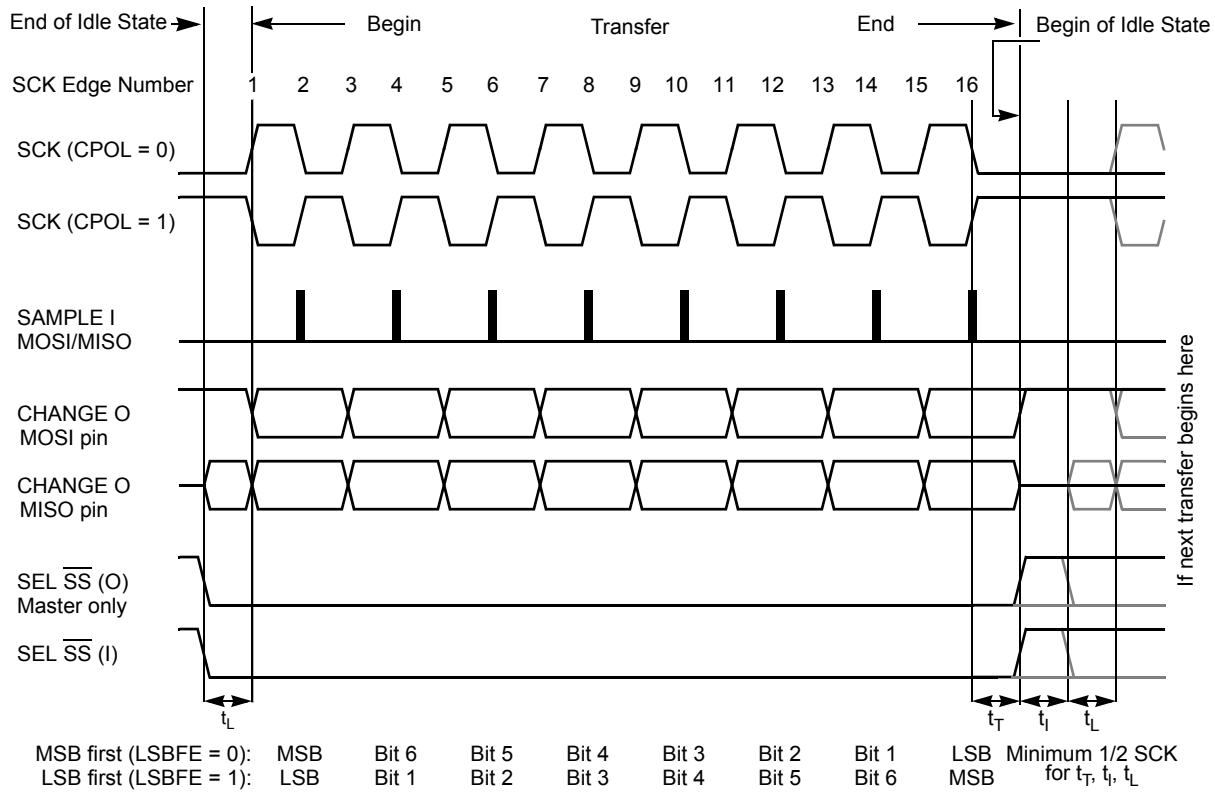
This process continues for a total of n^1 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted 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 $2n^1$ SCK edges:

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

Figure 17-14 shows two clocking variations for CPHA = 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 $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.



t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 17-14. SPI Clock Format 1 (CPHA = 1), with 8-Bit Transfer Width selected (XFRW = 0)

18.3.2.16 GDU Overcurrent Register 1 (GDUOC1)

Module Base + 0x000F — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	GOCA1	GOCE1	0	GOCT1[4:0]				
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-18. GDU Overcurrent Register 1 (GDUOC1)

1. Read: Anytime
   Write: Only if GWP=0

Table 18-20. GDUOC1 Register Field Descriptions

Field	Description
7 GOCA1	GDU Overcurrent Action — This bit cannot be modified after GWP bit is set. This bit controls the action in case of an overcurrent event or overvoltage event. See Table 18-24 and Table 18-23
6 GOCE1	GDU Overcurrent Enable — This bit cannot be modified after GWP bit is set. 0 Overcurrent Comparator 1 is disabled 1 Overcurrent Comparator 1 is enabled
GDUV4 (includes GOCT1 bits 3:0)	
3:0 GOCT1[3:0]	GDU Overcurrent Comparator Threshold — These bits cannot be modified after GWP bit is set. The overcurrent comparator threshold voltage is the output of a 6-bit digital-to-analog converter. The upper two bits of the digital inputs are tied to one. The other bits of the digital inputs are driven by GOCT1. The overcurrent comparator threshold voltage can be calculated from equation below. $$Voct1 = (48 + GOCT1) \cdot \frac{VDDA}{64}$$
GDUV5 and V6 (includes GOCT1 bits 4:0)	
4:0 GOCT1[4:0]	GDU Overcurrent Comparator Threshold — These bits cannot be modified after GWP bit is set. The overcurrent comparator threshold voltage is the output of a 6-bit digital-to-analog converter. The upper bit of the digital inputs is tied to one. The other bits of the digital inputs are driven by GOCT1. The overcurrent comparator threshold voltage can be calculated from equation below. $$Voct1 = (32 + GOCT1) \cdot \frac{VDDA}{64}$$

18.5.3 Calculation of Bootstrap Capacitor

The size of the bootstrap capacitor C_{BS} depends on the total gate charge Q_G needed to turn on the power FET used in the application. If the bootstrap capacitor is too small there can be a large voltage drop due to charge sharing between bootstrap capacitor C_{BS} and the total gate capacitance of the power FET C_G. The resulting voltage on the gate of the power FET can be calculated as follow:

Eqn. 18-1

$$V_G = \frac{Q_{BS}}{C_{BS} + C_G} = \frac{V_{BS}}{1 + \frac{C_G}{C_{BS}}}$$

For example if $C_{BS} = 20\ C_G$ then the resulting gate voltage is $V_G = 0.95\ V_{BS}$.

18.5.4 On Chip GDU t_{delon} and t_{deloff} Measurement

The S12ZVM256 provides the capability to measure the GDU t_{delon} and t_{deloff} delays of the high-side and low-side drivers with the on chip timer. The timing diagram Figure 18-32 shows the basic concept. The high-side and low-side drivers provide the feedback signals hs0_fb and ls0_fb which indicate that the drivers are turned on or off. The feedback signals and the related pwm signals are used to generate the gdu_del_on_off output signal. (see Figure 18-32) This signal can be routed to TIM1 input capture channel IOC1_0 for pulse width measurement.

Following below are the steps to do the delay measurement:

- 1. Route gdu_del_on_off signal to TIM1 IOC1_0 in PIM routing register MODRR2.T1ICORR
- 2. Setup TIM1 IOC1_0 for pulse width measurement
- 3. Use software control of PWM output feature PMFOUTC and PMFOUTB to assert PWM0
- 4. Store measured pulse width (t_{delon} of high-side driver 0) in RAM
- 5. Use software control of PWM output feature PMFOUTC and PMFOUTB to deassert PWM0
- 6. Store measured pulse width (t_{deloff} of high-side driver 0) in RAM
- repeat 3 to 6 for all PWM channels

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

22.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram Figure 22-16 as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

22.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

22.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 22.4.1, “PWM Clock Select” for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 22-16. 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 behaves differently depending on what output mode is selected as shown in Figure 22-16 and described in Section 22.4.2.5, “Left Aligned Outputs” and Section 22.4.2.6, “Center Aligned Outputs”.

M.9 0x0480-0x04AF PWM0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0489	PWMSCLB	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x048A - 0x048B	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x048C	PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048D	PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048E	PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048F	PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0490	PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0491	PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0492	PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0493	PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0494	PWMPER0	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0495	PWMPER1	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0496	PWMPER2	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0497	PWMPER3	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0498	PWMPER4	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0499	PWMPER5	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x049A	PWMPER6	R W	Bit 7	6	5	4	3	2	1	Bit 0