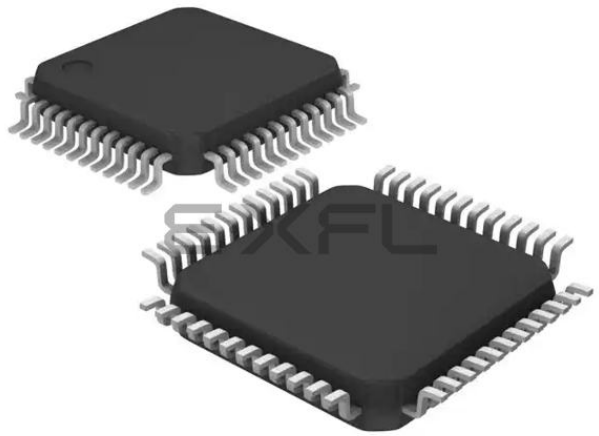


Welcome to E-XFL.COM

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, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/s912zvl96f0vlf

Port	Pin	Pin Function & Priority	I/O	Description	Routing Register Bit	Pin Function after Reset
T	PT7-6	IOC1_1:IOC1_0	I/O	TIM1 channel 1-0	T1C1RR:T1C0RR	GPIO
		PTT[7:6]	I/O	General-purpose	—	
	PT5	(TXD1)	O	SCI1 transmit	SCI1RR	
		IOC0_5	I/O	TIM0 channel 5	T0C5RR	
		PTT[5]	I/O	General-purpose	—	
	PT4	(RXD1)	I	SCI1 receive	SCI1RR	
		IOC0_4	I/O	TIM0 channel 4	T0C4RR	
		PTT[4]	I/O	General-purpose	—	
	PT3	IOC0_3[2]	I/O	TIM0 channel 3	T0C3RR, T0IC3RR1-0	
		PTT[3]	I/O	General-purpose	—	
	PT2	ACMPO0	O	ACMP0 output	—	
		IOC0_2	I/O	TIM0 channel 2	T0C2RR	
		PTT[2]	I/O	General-purpose	—	
	PT1	(LPRXD0)	O	LINPHY0 receive output	S0L0RR2-0	
		(PWM0)	O	PWM option 0	PWM0RR	
		TXD1	O	SCI1 transmit	SCI1RR	
		(SCL0)	I/O	IIC0	IIC0RR	
		IOC0_1	I/O	TIM0 channel 1	—	
		PTT[1]	I/O	General-purpose	—	
	PT0	(LPTXD0)	I	LINPHY0 transmit input	S0L0RR2-0	
		(PWM2)	O	PWM option 2	PWM2RR	
		RXD1	I	SCI1 receive	SCI1RR	
		(SDA0)	I/O	IIC0	IIC0RR	
		IOC0_0	I/O	TIM0 channel 0	—	
		PTT[0]	I/O	General-purpose	—	

Table 5-6. BDCCSRL Field Descriptions (continued)

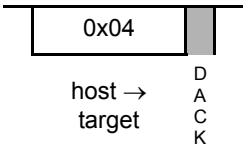
Field	Description
4 OVRUN	**Overrun Flag** — Indicates unexpected host activity before command completion. This occurs if a new command is received before the current command completion. With ACK enabled this also occurs if the host drives the BKGD pin low whilst a target ACK pulse is pending To protect internal resources from misinterpreted BDC accesses following an overrun, internal accesses are suppressed until a SYNC clears this bit. A SYNC clears the bit. 0 No overrun detected. 1 Overrun detected when issuing a BDC command.
3 NORESP	**No Response Flag** — Indicates that the BDC internal action or data access did not complete. This occurs in the following scenarios: a) If no free cycle for an access is found within 512 core clock cycles. This could typically happen if a code loop without free cycles is executing with ACK enabled and STEAL clear. b) With ACK disabled or STEAL set, when an internal access is not complete before the host starts data/BDCCSRL retrieval or an internal write access is not complete before the host starts the next BDC command. c) Attempted internal memory or SYNC_PC accesses during STOP mode set NORESP if BDCCIS is clear. In the above cases, on setting NORESP, the BDC aborts the access if permitted. (For devices supporting EWAIT, BDC external accesses with EWAIT assertions, prevent a command from being aborted until EWAIT is deasserted). d) If a BACKGROUND command is issued whilst the device is in wait mode the NORESP bit is set but the command is not aborted. The active BDM request is completed when the device leaves wait mode. Furthermore subsequent CPU register access commands during wait mode set the NORESP bit, should it have been cleared. e) If a command is issued whilst awaiting return from Wait mode. This can happen when using STEP1 to step over a CPU WAI instruction, if the CPU has not returned from Wait mode before the next BDC command is received. f) If STEP1 is issued with the BDC enabled as the device enters Wait mode regardless of the BDMACT state. When NORESP is set a value of 0xEE is returned for each data byte associated with the current access. Writing a “1” to this bit, clears the bit. 0 Internal action or data access completed. 1 Internal action or data access did not complete.
2 RDINV	**Read Data Invalid Flag** — Indicates invalid read data due to an ECC error during a BDC initiated read access. The access returns the actual data read from the location. Writing a “1” to this bit, clears the bit. 0 No invalid read data detected. 1 Invalid data returned during a BDC read access.
1 ILLACC	**Illegal Access Flag** — Indicates an attempted illegal access. This is set in the following cases: When the attempted access addresses unimplemented memory When the access attempts to write to the flash array When a CPU register access is attempted with an invalid CRN (Section 5.4.5.1, “BDC Access Of CPU Registers). Illegal accesses return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal access detected. 1 Illegal BDC access detected.

The ACK_ENABLE command is interpreted and executed in the BDC logic without the need to interface with the CPU. An ACK pulse is issued by the target device after this command is executed. This command can be used by the host to evaluate if the target supports the hardware handshake protocol. If the target supports the hardware handshake protocol, subsequent commands are enabled to execute the hardware handshake protocol, otherwise this command is ignored by the target. Table 5-8 indicates which commands support the ACK hardware handshake protocol.

For additional information about the hardware handshake protocol, refer to Section 5.4.7, “Serial Interface Hardware Handshake (ACK Pulse) Protocol,” and Section 5.4.8, “Hardware Handshake Abort Procedure.”

5.4.4.4 BACKGROUND

Enter active background mode (if enabled) **Non-intrusive**



Provided ENBDC is set, the BACKGROUND command causes the target MCU to enter active BDM as soon as the current CPU instruction finishes. If ENBDC is cleared, the BACKGROUND command is ignored.

A delay of 16 BDCSI clock cycles is required after the BACKGROUND command to allow the target MCU to finish its current CPU instruction and enter active background mode before a new BDC command can be accepted.

The host debugger must set ENBDC before attempting to send the BACKGROUND command the first time. Normally the host sets ENBDC once at the beginning of a debug session or after a target system reset. During debugging, the host uses GO commands to move from active BDM to application program execution and uses the BACKGROUND command or DBG breakpoints to return to active BDM.

A BACKGROUND command issued during stop or wait modes cannot immediately force active BDM because the WAI instruction does not end until an interrupt occurs. For the detailed mode dependency description refer to Section 5.1.3.3, “Low-Power Modes.

The host can recognize this pending BDM request condition because both NORESP and WAIT are set, but BDMACT is clear. Whilst in wait mode, with the pending BDM request, non-intrusive BDC commands are allowed.

5.4.4.5 DUMP_MEM.sz, DUMP_MEM.sz_WS

DUMP_MEM.sz

Read memory specified by debug address register, then increment address **Non-intrusive**

Address: 0x00001D

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

= Unimplemented or Reserved

Figure 6-10. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001E

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

= Unimplemented or Reserved

Figure 6-11. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001F

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

= Unimplemented or Reserved

Figure 6-12. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

Be aware that the output frequency varies with the TC trimming. A frequency trimming correction is therefore necessary. The values provided in Table 9-28 are typical values at ambient temperature which can vary from device to device.

9.3.2.24 S12CPMU_UHV Oscillator Register (CPMUOSC)

This registers configures the external oscillator (XOSCLCP).

Module Base + 0x001A

	7	6	5	4	3	2	1	0
R	OSCE	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-33. S12CPMU_UHV Oscillator Register (CPMUOSC)

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

NOTE.

Write to this register clears the LOCK and UPOSC status bits.

Table 9-29. CPMUOSC Field Descriptions

Field	Description
7 OSCE	**Oscillator Enable Bit** — This bit enables the external oscillator (XOSCLCP). The UPOSC status bit in the CPMIUFLG register indicates when the oscillation is stable and when OSCCLK can be selected as source of the Bus Clock or source of the COP or RTI.**If the oscillator clock monitor reset is enabled (OMRE = 1 in CPMUOSC2 register), then a loss of oscillation will lead to an oscillator clock monitor reset.** 0 External oscillator is disabled. REFCLK for PLL is IRCCLK. 1 External oscillator is enabled. Oscillator clock monitor is enabled. External oscillator is qualified by PLLCLK. REFCLK for PLL is the external oscillator clock divided by REFDIV. **If OSCE bit has been set (write “1”) the EXTAL and XTAL pins are exclusively reserved for the oscillator and they can not be used anymore as general purpose I/O until the next system reset.** **Note:** When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop Mode.

10.2 Introduction

The ADC12B_LBA is an n-channel multiplexed input successive approximation analog-to-digital converter. Refer to device electrical specifications for ADC parameters and accuracy.

The List Based Architecture (LBA) provides flexible conversion sequence definition as well as flexible oversampling. The order of channels to be converted can be freely defined. Also, multiple instantiations of the module can be triggered simultaneously (matching sampling point across multiple module instantiations).

There are four register bits which control the conversion flow (please refer to the description of register ADCFLWCTL).
The four conversion flow control bits of register ADCFLWCTL can be modified in two different ways:

- Via data bus accesses
- Via internal interface Signals (Trigger, Restart, LoadOK, and Seq_Abort; see also Figure 10-2). Each Interface Signal is associated with one conversion flow control bit.

For information regarding internal interface connectivity related to the conversion flow control please refer to the device overview of the reference manual.

The ADCFLWCTL register can be controlled via internal interface only or via data bus only or by both depending on the register access configuration bits ACC_CFG[1:0].

The four bits of register ADCFLWCTL reflect the captured request and status of the four internal interface Signals (LoadOK, Trigger, Restart, and Seq_abort; see also Figure 10-2) if access configuration is set accordingly and indicate event progress (when an event is processed and when it is finished).

Conversion flow error situations are captured by corresponding interrupt flags in the ADCEIF register.

There are two conversion flow control modes (Restart Mode, Trigger Mode). Each mode causes a certain behavior of the conversion flow control bits which can be selected according to the application needs.

Please refer to Section 10.5.2.1, “ADC Control Register 0 (ADCCTL_0) and Section 10.6.3.2.4, “The two conversion flow control Mode Configurations for more information regarding conversion flow control.

Because internal components of the ADC are turned on/off with bit ADC_EN, the ADC requires a recovery time period (t_{REC}) after ADC is enabled until the first conversion can be launched via a trigger.

When bit ADC_EN gets cleared (transition from 1’b1 to 1’b0) 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 respectively aborted, which could take up to a maximum latency time of $t_{DISABLE}$ (see device level specification for more details).

10.5.2.7 ADC Error Interrupt Enable Register (ADCEIE)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	IA_EIE	CMD_EIE	EOL_EIE	Reserved	TRIG_EIE	RSTAR_EIE	LDOK_EIE	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-10. ADC Error Interrupt Enable Register (ADCEIE)

Read: Anytime

Write: Anytime

Table 10-12. ADCEIE Field Descriptions

Field	Description
7 IA_EIE	**Illegal Access Error Interrupt Enable Bit** — This bit enables the illegal access error interrupt. 0 Illegal access error interrupt disabled. 1 Illegal access error interrupt enabled.
6 CMD_EIE	**Command Value Error Interrupt Enable Bit** — This bit enables the command value error interrupt. 0 Command value interrupt disabled. 1 Command value interrupt enabled.
5 EOL_EIE	**"End Of List" Error Interrupt Enable Bit** — This bit enables the "End Of List" error interrupt. 0 "End Of List" error interrupt disabled. 1 "End Of List" error interrupt enabled.
3 TRIG_EIE	**Conversion Sequence Trigger Error Interrupt Enable Bit** — This bit enables the conversion sequence trigger error interrupt. 0 Conversion sequence trigger error interrupt disabled. 1 Conversion sequence trigger error interrupt enabled.
2 RSTAR_EIE	**Restart Request Error Interrupt Enable Bit**— This bit enables the restart request error interrupt. 0 Restart Request error interrupt disabled. 1 Restart Request error interrupt enabled.
1 LDOK_EIE	**Load OK Error Interrupt Enable Bit** — This bit enables the Load OK error interrupt. 0 Load OK error interrupt disabled. 1 Load OK error interrupt enabled.

10.5.2.19 ADC Command Index Register (ADCCIDX)

It is important to note that these bits do not represent absolute addresses instead it is a sample index (object size 32bit).

Module Base + 0x001C

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

= Unimplemented or Reserved

Figure 10-22. ADC Command Index Register (ADCCIDX)

Read: Anytime

Write: NA

Table 10-27. ADCCIDX Field Descriptions

Field	Description
5-0 CMD_IDX [5:0]	**ADC Command Index Bits** — These bits represent the command index value for the conversion commands relative to the two CSL start addresses in the memory map. These bits do not represent absolute addresses instead it is a sample index (object size 32bit). See also Section 10.6.3.2.2, “Introduction of the two Command Sequence Lists (CSLs) for more details.

Table 13-12. 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 13.3.2.1, “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 13.3.2.5, “MSCAN Receiver Flag Register (CANRFLG)”).

13.3.2.7 MSCAN Transmitter Flag Register (CANTFLG)

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

13.5 Initialization/Application Information

13.5.1 MSCAN initialization

The procedure to initially start up the MSCAN module out of reset is as follows:

1. Assert CANE
2. Write to the configuration registers in initialization mode
3. Clear INITRQ to leave initialization mode

If the configuration of registers which are only writable in initialization mode shall be changed:

1. Bring the module into sleep mode by setting SLPRQ and awaiting SLPAK to assert after the CAN bus becomes idle.
2. Enter initialization mode: assert INITRQ and await INITAK
3. Write to the configuration registers in initialization mode
4. Clear INITRQ to leave initialization mode and continue

Table 15-5. TTOV Field Descriptions

Note: Writing to unavailable bits has no effect. Reading from unavailable bits return a zero.

Field	Description
5:0 TOV[5:0]	**Toggle On Overflow Bits** — TOVx toggles output compare pin on overflow. This feature only takes effect when in output compare mode. When set, it takes precedence over forced output compare 0 Toggle output compare pin on overflow feature disabled. 1 Toggle output compare pin on overflow feature enabled.

15.3.2.6 Timer Control Register 1/Timer Control Register 2 (TCTL1/TCTL2)

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	RESERVED	RESERVED	OM5	OL5	OM4	OL4
Reset	0	0	0	0	0	0	0	0

Figure 15-10. Timer Control Register 1 (TCTL1)

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R W	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset	0	0	0	0	0	0	0	0

Figure 15-11. Timer Control Register 2 (TCTL2)

Read: Anytime

Write: Anytime

Table 15-6. TCTL1/TCTL2 Field Descriptions

Note: Writing to unavailable bits has no effect. Reading from unavailable bits return a zero

Field	Description
5:0 OMx	**Output Mode** — These six pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** For an output line to be driven by an OCx the OCPDx must be cleared.
5:0 OLx	**Output Level** — These sixpairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare. When either OMx or OLx is 1, the pin associated with OCx becomes an output tied to OCx. **Note:** For an output line to be driven by an OCx the OCPDx must be cleared.

16.2 External Signal Description

The TIM16B2CV3 module has a selected number of external pins. Refer to device specification for exact number.

16.2.1 IOC1 - IOC0 — Input Capture and Output Compare Channel 1-0

Those pins serve as input capture or output compare for TIM16B2CV3 channel .

NOTE

For the description of interrupts see Section 16.6, “Interrupts”.

16.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers.

16.3.1 Module Memory Map

The memory map for the TIM16B2CV3 module is given below in Figure 16-3. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the TIM16B2CV3 module and the address offset for each register.

16.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Only bits related to implemented channels are valid.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	IOS1	IOS0
	W								
0x0001 CFORC	R	0	0	0	0	0	0	0	0
	W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	FOC1	FOC0
0x0004 TCNTH	R	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
	W								
0x0005 TCNTL	R	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	W								
0x0006 TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
	W								
0x0007 TTOV	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	TOV1	TOV0
	W								
0x0008 TCTL1	R	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
	W								

Figure 16-3. TIM16B2CV3 Register Summary (Sheet 1 of 2)

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 17-2. 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.

17.3.2.1 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 17-3. PWM Enable Register (PWME)

Read: Anytime

Write: Anytime

Clock Source = bus clock, where bus clock= 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/8 = 1.25 MHz

PWMx Period = 800 ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown in Figure 17-20 is the output waveform generated.

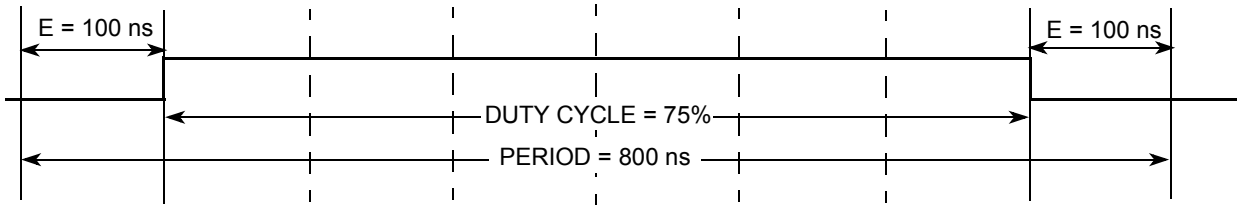


Figure 17-20. PWM Center Aligned Output Example Waveform

17.4.2.7 PWM 16-Bit Functions

The scalable PWM timer also has the option of generating up to 8-channels of 8-bits or 4-channels of 16-bits for greater PWM resolution. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains four control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 6 and 7 are concatenated with the CON67 bit, channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

NOTE

Change these bits only when both corresponding channels are disabled.

When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel, as shown in Figure 17-21. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

When using the 16-bit concatenated mode, the clock source is determined by the low order 8-bit channel clock select control bits. That is channel 7 when channels 6 and 7 are concatenated, channel 5 when channels 4 and 5 are concatenated, channel 3 when channels 2 and 3 are concatenated, and channel 1 when channels 0 and 1 are concatenated. The resulting PWM is output to the pins of the corresponding low order 8-bit channel as also shown in Figure 17-21. The polarity of the resulting PWM output is controlled by the PPOLx bit of the corresponding low order 8-bit channel as well.

from where was during the previous transmission. It is not possible for the IIC to wake up the CPU when its internal clocks are stopped.

If it were the case that the IBSWAI bit was cleared when the WAI instruction was executed, the IIC internal clocks and interface would remain alive, continuing the operation which was currently underway. It is also possible to configure the IIC such that it will wake up the CPU via an interrupt at the conclusion of the current operation. See the discussion on the IBIF and IBIE bits in the IBSR and IBCR, respectively.

20.3.1.4 IIC Status Register (IBSR)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	TCF	IAAS	IBB	IBAL	0	SRW	IBIF	RXAK
W								
Reset	1	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-7. IIC Bus Status Register (IBSR)

This status register is read-only with exception of bit 1 (IBIF) and bit 4 (IBAL), which are software clearable.

Table 20-9. IBSR Field Descriptions

Field	Description
7 TCF	**Data Transferring Bit** — While one byte of data is being transferred, this bit is cleared. It is set by the falling edge of the 9th clock of a byte transfer. Note that this bit is only valid during or immediately following a transfer to the IIC module or from the IIC module. 0 Transfer in progress 1 Transfer complete
6 IAAS	**Addressed as a Slave Bit** — When its own specific address (I-bus address register) is matched with the calling address or it receives the general call address with GCEN== 1,this bit is set.The CPU is interrupted provided the IBIE is set.Then the CPU needs to check the SRW bit and set its Tx/$\overline{\text{Rx}}$ mode accordingly.Writing to the I-bus control register clears this bit. 0 Not addressed 1 Addressed as a slave
5 IBB	Bus Busy Bit 0 This bit indicates the status of the bus. When a START signal is detected, the IBB is set. If a STOP signal is detected, IBB is cleared and the bus enters idle state. 1 Bus is busy
4 IBAL	**Arbitration Lost** — The arbitration lost bit (IBAL) is set by hardware when the arbitration procedure is lost. Arbitration is lost in the following circumstances: 1.SDA sampled low when the master drives a high during an address or data transmit cycle. 2.SDA sampled low when the master drives a high during the acknowledge bit of a data receive cycle. 3.A start cycle is attempted when the bus is busy. 4.A repeated start cycle is requested in slave mode. 5.A stop condition is detected when the master did not request it. This bit must be cleared by software, by writing a one to it. A write of 0 has no effect on this bit.

Table A-1. Revision History Table

Revision Number	Revision Date	Description Of Changes
0.50	14 March 2016	• added the latest characterization data, updated: Table A-11, Table A-19 • change voltage specification for MC9S12ZVL128/96/64 analog modules to VDDX ±3%:
0.60	31 March 2016	• update V_{BG} output voltage and V_{BG} voltage distribution specification: Table B-1
0.70	18 April 2016	• correct min V_{DDX} specification for MC9S12ZVL128/96/64 device: Table B-1
0.80	20 June 2016	• update Table A-19, add missing stop current for 85°C and 105°C, correct stop value for 125°C • update Table I-2, set ACMP input offset to 25mV
0.90	08 August 2017	• added 175°C parameters • update current injection consideration, section **Section C.1.1.4 Current Injection**
1.0	12 September 2017	• added 175°C Run and Wait current parameters
1.1	10 October 2017	• added Pin input leakage values for Pins PAD0 and PAD1 at 150°C < T_J < 175°C, Table A-10 • added Pin input leakage values for Pins PP1,PP3,PP5 and PP7 at 150°C < T_J < 175°C, Table A-10 • changed typical Reduced Performance Mode V_{DDX} Voltage to 5.0V, Table B-1
1.2	19 October 2017	• correct max value for Input leakage current on PP1, PP3, PP5 and PP7 for 150°C < T_J < 175°C on Table A-10
1.21	24 October 2017	• fixed minor bug in this revision history to make sure all updates are correct documented

A.1 General

This supplement contains the most accurate electrical information for the MC9S12ZVL-Family available at the time of publication.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

Table A-2. Power Supplies

Mnemonic	Nominal Voltage	Description
VSS	0V	Ground pin for 1.8V core supply voltage generated by on chip voltage regulator
VDDX	5.0 V	5V power supply output for I/O drivers generated by on chip voltage regulator if VREG5VEN is set
VDDX	3.3 V	3.3V power supply output for I/O drivers generated by on chip voltage regulator if VREG5VEN is cleared
VSSX1	0V	Ground pin for I/O drivers

Appendix C
ADC Specifications

This section describes the characteristics of the analog-to-digital converter.

C.1 ADC Operating Characteristics

The Table C-1 and Table C-2 show conditions under which the ADC operates.

The following constraints exist to obtain full-scale, full range results:

$$V_{SSA} \le V_{RL} \le V_{IN} \le V_{RH} \le V_{DDA}.$$

This constraint exists since the sample buffer amplifier can not drive beyond the power supply levels that it ties to. If the input level goes outside of this range it will effectively be clipped.

Table C-1. ADC Operating Characteristics

Supply voltage: MC9S12ZVL(S)32\16\8: 3.13V ≤ V_{DDX} ≤ 5.5V, MC9S12ZVL(A)128\96\64: 3.2V ≤ V_{DDX} ≤ 5.15V, -40°C < T_J < 175°C						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Reference potential Low High	 V_{RL} V_{RH}	 V_{SSA} $V_{DDA}/2$	 — —	 $V_{DDA}/2$ V_{DDA}	 V V
2	Voltage difference V_{DDX} to V_{DDA}	Δ_{VDDX}	–0.1	0	0.1	V
3	Voltage difference V_{SSX} to V_{SSA}	Δ_{VSSX}	–0.1	0	0.1	V
4	Differential reference voltage[1]	V_{RH}-V_{RL}	3.13	5.0	5.5	V
5	ATD Clock Frequency (derived from bus clock via the prescaler bus)	f_{ATDCLk}	0.25		8.34	MHz
6	Buffer amplifier recovery time (turn on delay after module start/recovery from stop)	t_{REC}			1	µs
7	ATD Conversion Period[2] 12 bit resolution: 10 bit resolution: 8 bit resolution:	 N_{CONV12} N_{CONV10} N_{CONV8}	 20 18 16		 40 38 36	 ATD clock Cycles

1 The accuracy is reduced if the differential reference voltage is less than 3.13V when using the ATD in the 3.3V range or if the differential reference voltage is less than 4.5V when using the ATD in the 5V range

2 The minimum time assumes a sample time of 4 ATD clock cycles. The maximum time assumes a sample time of 24 ATD clock cycles.

C.1.1 Factors Influencing Accuracy

Source resistance, source capacitance and current injection have an influence on the accuracy of the ADC, see Figure C-1 . A further factor is that port AD pins that are configured as output drivers switching.

O.4 0x0100-0x017F S12ZDBG (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0107	DBGSCR1	R	C3SC1	C3SC0	0	0	C1SC1	C1SC0	C0SC1	C0SC0
		W								
0x0108	DBGSCR2	R	C3SC1	C3SC0	0	0	C1SC1	C1SC0	C0SC1	C0SC0
		W								
0x0109	DBGSCR3	R	C3SC1	C3SC0	0	0	C1SC1	C1SC0	C0SC1	C0SC0
		W								
0x010A	DBGEFR	R	0	TRIGF	0	EEVF	ME3	0	ME1	ME0
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								

O.12 0x0600-0x063F ADC0 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0620	ADC0RIDX	R	0	0	RES_IDX[5:0]					
		W								
0x0621	ADC0RBP_0	R	0	0	0	0	RES_PTR[19:16]			
		W								
0x0622	ADC0RBP_1	R	RES_PTR[15:8]							
		W								
0x0623	ADC0RBP_2	R	RES_PTR[7:2]						0	0
		W								
0x0624	ADC0CROFF0	R	0	CMDRES_OFF0[6:0]						
		W								
0x0625	ADC0CROFF1	R	0	CMDRES_OFF1[6:0]						
		W								
0x0626	Reserved	R	0	0	0	0	Reserved			
		W								
0x0627	Reserved	R	Reserved							
		W								
0x0628	Reserved	R	Reserved						0	0
		W								
0x0629	Reserved	R	Reserved	0	Reserved					
		W								
0x062A-0x063F	Reserved	R	0	0	0	0	0	0	0	0
		W								

O.13 0x0680-0x0687 DAC

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0680	DACCTL	R	FVR	DRIVE	0	0	0	DACM[2:0]		
		W								
0x0681	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0682	DACVOL	R	VOLTAGE[7:0]							
		W								
0x0683-0x0686	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0687	Reserved	R	0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								