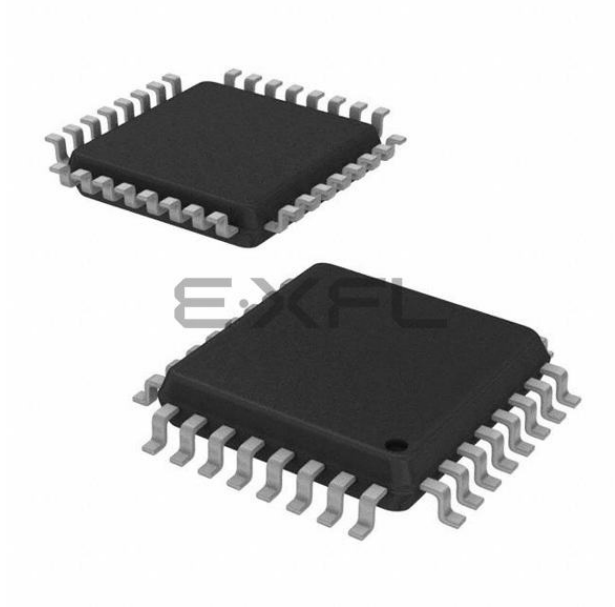


Welcome to E-XFL.COM

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	19
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 6x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvla96f0clcr

Revision History

Date	Revision Level	Description
13 May 2015	2.00 Draft E	• Added new version of Chapter 12, “Programmable Gain Amplifier (PGAV1) • Added new version of Chapter 11, “Digital Analog Converter (DAC_8B5V_V2) • added missing modules DAC, PGA, ACMP, PWM1 to Appendix O, “Detailed Register Address Map • update voltage range inside Appendix I, “ACMP Electrical Specifications and Appendix G, “DAC_8B5V Electrical Specifications
05 June 2015	2.00 Draft F	• Added new version of Chapter 1, “Device Overview MC9S12ZVL-Family” • Added new version of Appendix A, “MCU Electrical Specifications
27 October 2015	2.00 Draft G	• Added new version v0.4 of Appendix A, “MCU Electrical Specifications” • correct Order Information
25 February 2016	2.00Draft I	• change to NXP style • Added new version of Appendix A, “MCU Electrical Specifications
10 May 2016	2.00	• Added version 0.80 of Appendix A, “MCU Electrical Specifications • changed to Chapter 10, “Analog-to-Digital Converter (ADC12B_LBA) version V3
08 August 2017	2.10	• Added version 0.90 of Appendix A, “MCU Electrical Specifications • Added version 2.00 of Chapter 1, “Device Overview MC9S12ZVL-Family
12 September 2017	2.20	• Added version 1.0 of Appendix A, “MCU Electrical Specifications
10 October 2017	2.30	• Added version 1.1 of Appendix A, “MCU Electrical Specifications
19 October 2017	2.40	• Added version 1.2 of Appendix A, “MCU Electrical Specifications
24 October 2017	2.41	• Added version 1.21 of Appendix A, “MCU Electrical Specifications

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x02F4	PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
		W								
0x02F5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x02F6	PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
		W								
0x02F7	PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
		W								
0x02F8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x02F9	OCPEP	R	OCPEP7	0	OCPEP5	0	OCPEP3	0	OCPEP1	0
		W								
0x02FA	OCIEP	R	OCIEP7	0	OCIEP5	0	OCIEP3	0	OCIEP1	0
		W								
0x02FB	OCIFP	R	OCIFP7	0	OCIFP5	0	OCIFP3	0	OCIFP1	0
		W								
0x02FC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x02FD	RDRP	R	RDRP7	0	RDRP5	0	RDRP3	0	RDRP1	0
		W								
0x02FE–0x02FF	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0300–0x030F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0310	PTJ	R	0	0	0	0	0	0	PTJ1	PTJ0
		W								
0x0311	PTIJ	R	0	0	0	0	0	0	PTIJ1	PTIJ0
		W								
0x0312	DDRJ	R	0	0	0	0	0	0	DDRJ1	DDRJ0
		W								
0x0313	PERJ	R	0	0	0	0	0	0	PERJ1	PERJ0
		W								

Table 5-8. BDC Command Summary (continued)

Command Mnemonic	Command Classification	ACK	Command Structure	Description
READ_SAME.sz	Non-Intrusive	Yes	(0x50+4 x sz)/dack/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_SAME.sz_WS	Non-Intrusive	No	(0x51+4 x sz)/d/ss/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_BDCCSR	Always Available	No	0x2D/rd16	Read the BDCCSR register
SYNC_PC	Non-Intrusive	Yes	0x01/dack/rd24	Read current PC
WRITE_MEM.sz	Non-Intrusive	Yes	(0x10+4 x sz)/ad24/wd.sz/dack	Write the appropriately-sized (sz) memory value to the location specified by the 24-bit address
WRITE_MEM.sz_WS	Non-Intrusive	No	(0x11+4 x sz)/ad24/wd.sz/d/ss	Write the appropriately-sized (sz) memory value to the location specified by the 24-bit address and report status
WRITE_Rn	Active Background	Yes	(0x40+CRN)/wd32/dack	Write the requested CPU register
WRITE_BDCCSR	Always Available	No	0x0D/wd16	Write the BDCCSR register
ERASE_FLASH	Always Available	No	0x95/d	Mass erase internal flash
STEP1 (TRACE1)	Active Background	Yes	0x09/dack	Execute one CPU command.

1 The SYNC command is a special operation which does not have a command code.

2 The GO_UNTIL command is identical to the GO command if ACK is not enabled.

5.4.4.1 SYNC

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct speed to use for serial communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

1. Ensures that the BKGD pin is high for at least 4 cycles of the slowest possible BDCSI clock without reset asserted.
2. Drives the BKGD pin low for at least 128 cycles of the slowest possible BDCSI clock.
3. Drives BKGD high for a brief speed-up pulse to get a fast rise time. (This speedup pulse is typically one cycle of the host clock which is as fast as the maximum target BDCSI clock).
4. Removes all drive to the BKGD pin so it reverts to high impedance.
5. Listens to the BKGD pin for the sync response pulse.

In stop mode, the S12ZINTV0 module is capable of waking up the CPU if an eligible CPU exception occurs. Please refer to Section 6.5.3, “Wake Up from Stop or Wait Mode” for details.

6.1.4 Block Diagram

Figure 6-1 shows a block diagram of the S12ZINTV0 module.

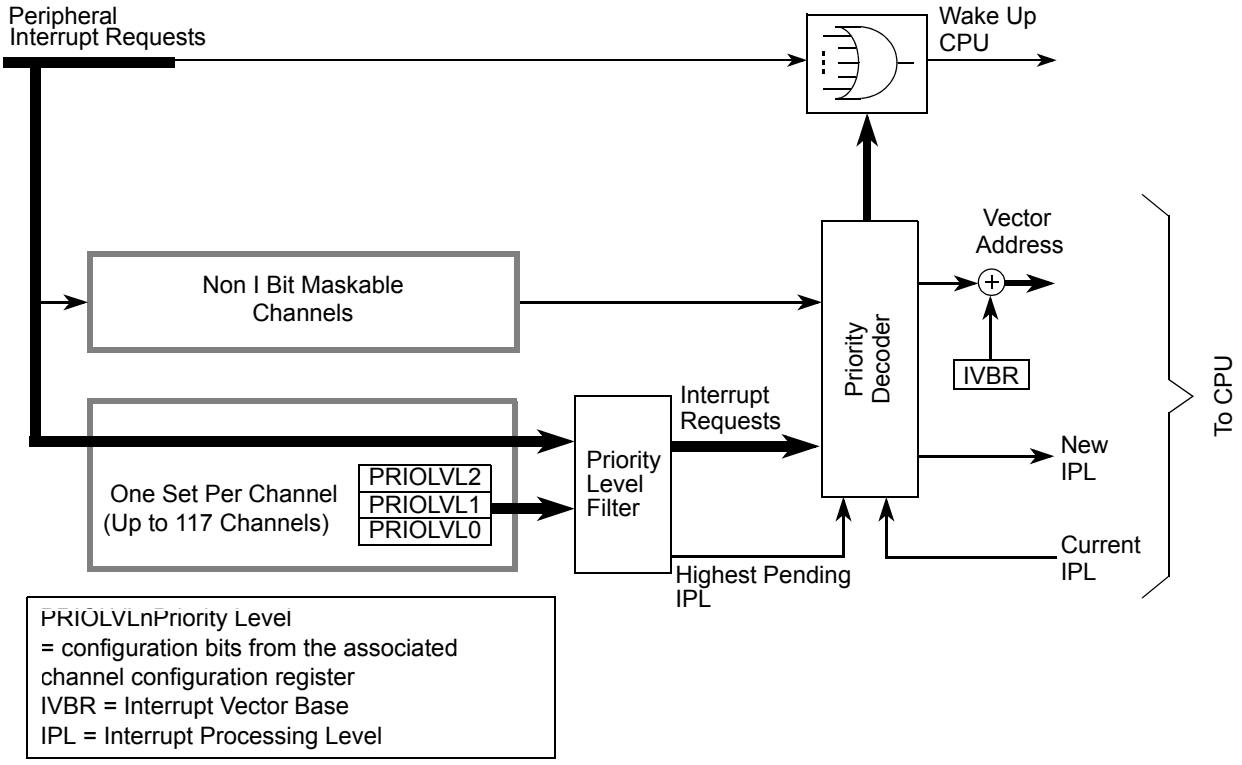


Figure 6-1. S12ZINTV0 Block Diagram

6.2 External Signal Description

The S12ZINTV0 module has no external signals.

6.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the S12ZINTV0 module.

6.3.1 Module Memory Map

Table 6-3 gives an overview over all S12ZINTV0 module registers.

Table 6-3. S12ZINTV0 Memory Map

Address	Use	Access

9.1.1 Features

The Pierce Oscillator (XOSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power and good noise immunity.

- Supports crystals or resonators from 4MHz to 20MHz.
- High noise immunity due to input hysteresis and spike filtering.
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor
- Low power consumption: Operates from internal 1.8V (nominal) supply, Amplitude control limits power
- Optional oscillator clock monitor reset
- Optional full swing mode for higher immunity against noise injection on the cost of higher power consumption and increased emission

The Voltage Regulator (VREGAUTO) has the following features:

- Input voltage range from 6 to 18V (nominal operating range)
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR)
- Low-voltage reset (LVR)
- On Chip Temperature Sensor and Bandgap Voltage measurement via internal ADC channel.
- Voltage Regulator providing Full Performance Mode (FPM) and Reduced Performance Mode (RPM)
- External ballast device support to reduce internal power dissipation
- Capable of supplying both the MCU internally plus external components
- Over-temperature interrupt

The Phase Locked Loop (PLL) has the following features:

- Highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- PLL clock monitor reset
- Reference clock either external (crystal) or internal square wave (1MHz IRC1M) based.
- PLL stability is sufficient for LIN communication in slave mode, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

10.5.2.16 ADC Command Register 1 (ADCCMD_1)

A command which contains reserved bit settings causes the error flag CMD_EIF being set and ADC cease operation. The CMD_EIF is never set for Internal_x channels, even if the channels are specified as reserved in the Device Overview section of the Reference Manual.

Module Base + 0x0015

	23	22	21	20	19	18	17	16
R W	VRH_SEL[1]	VRL_SEL[1]	CH_SEL[5:0]					
R W	VRH_SEL[1:0][2]		CH_SEL[5:0]					
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-19. ADC Command Register 1 (ADCCMD_1)

1 Only available on ADC12B_LBA V1 and V2 (see Table 10-2 for details)

2 Only available on ADC12B_LBA V3 (see Table 10-2 for details)

Read: Anytime

Write: Only writable if bit SMOD_ACC is set
(see also Section 10.5.2.2, “ADC Control Register 1 (ADCCTL_1) bit SMOD_ACC description for more details)

Table 10-23. ADCCMD_1 Field Descriptions

Field	Description
	ADC12B_LBA V1 and V2 (includes VRH_SEL/VRL_SEL)
23 VRH_SEL	**Reference High Voltage Select Bit** — This bit selects the high voltage reference for current conversion. 0 VRH_0 input selected as high voltage reference. 1 VRH_1 input selected as high voltage reference.
22 VRL_SEL	**Reference Low Voltage Select Bit** — This bit selects the low voltage reference for current conversion. 0 VRL_0 input selected as low voltage reference. 1 VRL_1 input selected as low voltage reference.
	ADC12B_LBA V3 (includes VRH_SEL[1:0])
23-22 VRH_SEL	**Reference High Voltage Select Bit** — These bits select the high voltage reference for current conversion. 00 VRH_0 input selected as high voltage reference 01 VRH_1 input selected as high voltage reference 10 VRH_2 input selected as high voltage reference 11 Reserved
21-16 CH_SEL[5:0]	**ADC Input Channel Select Bits** — These bits select the input channel for the current conversion. See Table 10-24 for channel coding information.

NOTE

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

10.9.5 List Usage — CSL double buffer mode and RVL double buffer mode

In this use case both list types are configured for double buffer mode (CSL_BMOD=1’b1) and RVL_BMOD=1’b1).

This setup is the same as Section 10.9.3, “List Usage — CSL double buffer mode and RVL double buffer mode but at the end of a CSL the CSL is not always swapped (bit LDOK not always set with bit RSTA). The Result Value List is swapped whenever a CSL is finished or a CSL got aborted.

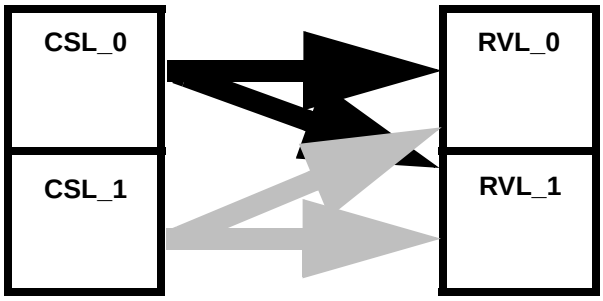


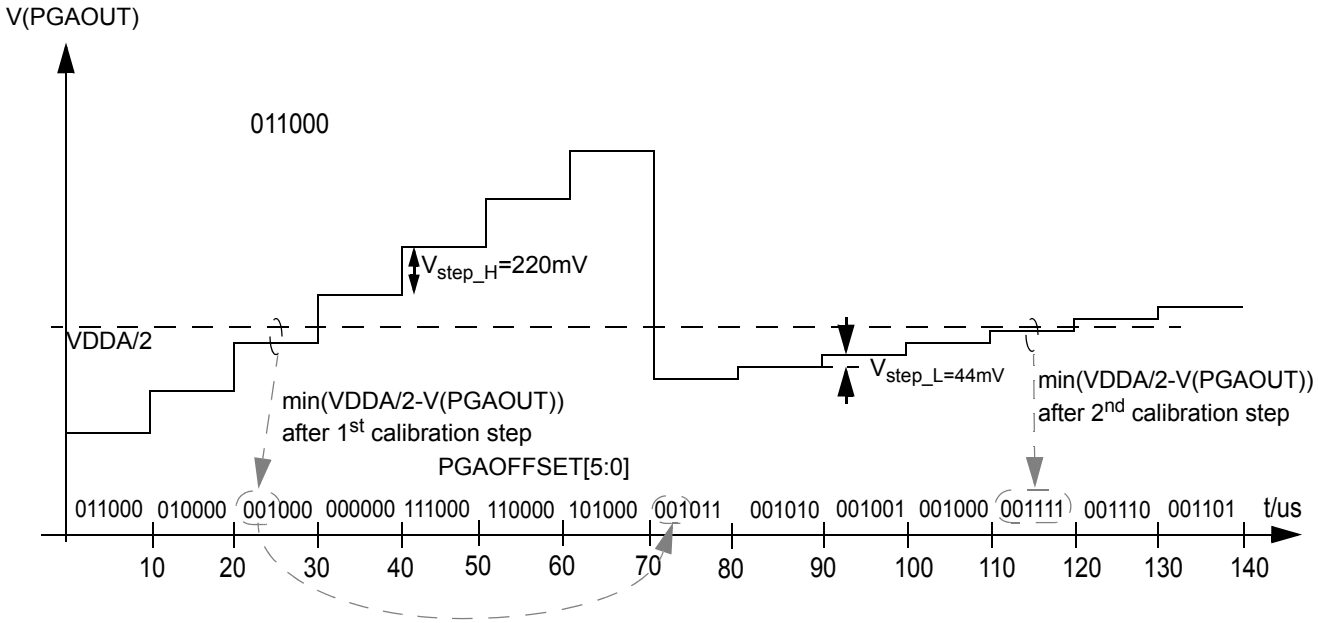
Figure 10-39. CSL Double Buffer Mode — RVL Double Buffer Mode Diagram

10.9.6 RVL swapping in RVL double buffer mode and related registers ADCIMDRI and ADCEOLRI

When using the RVL in double buffer mode, the registers ADCIMDRI and ADCEOLRI can be used by the application software to identify which RVL holds relevant and latest data and which CSL is related to this data. These registers are updated at the setting of one of the CON_IF[15:1] or the EOL_IF interrupt flags. As described in the register description Section 10.5.2.13, “ADC Intermediate Result Information Register (ADCIMDRI) and Section 10.5.2.14, “ADC End Of List Result Information Register (ADCEOLRI), the register ADCIMDRI, for instance, is always updated at the occurrence of a CON_IF[15:1] interrupt flag amongst other cases. Also each time the last conversion command of a CSL is finished and the corresponding result is stored, the related EOL_IF flag is set and register ADCEOLRI is updated. Hence application software can pick up conversion results, or groups of results, or an entire result list driven fully by interrupts. A use case example diagram is shown in Figure 10-40.

2. With the optimal PGAOFFSET[5:3] setting step through the offset compensation values PGAOFFSET[2:0]= {0x011, 0x010, 0x001, 0x000, 0x111, 0x110, 0x101} and measure the PGA_OUT value with the ADC. Select as optimal offset compensation value for the lower three bits the PGAOFFSET[2:0] which is closest to the expected ADC reading of VDDA/2.

Figure 12-8. Offset compensation timing diagram



12.4.3 Application Example for differential voltage measurement

For sensor applications it is often required to measure a small differential voltage V_{diff}. The PGA is not capable of amplifying a differential voltage, but an algorithm to calculate the differential voltage can be implemented. The PGA contains two input pins PGA_IN0 and PGA_IN1 which can be multiplexed by the ADC command list. By subtracting the ADC readings of the two pins the amplified differential voltage can be calculated.

For this algorithm two requirements must be met:

1. The minimum time for the input signal multiplexing is given by PGA to ADC settling time $t_{PGA\_settling}$. The rate of signal change within $t_{PGA\_settling}$ must be small.
2. The common mode input voltage range of the differential input signals must be limited that for a given gain A_{PGA} a reference voltage V_{ref} can be selected so that both amplified signals do not saturate.

If both requirements are met the algorithm can be implemented. The error calculation is the following:

$$V_{diff} = (ADC\_reading(PGA\_IN1) - ADC\_reading(PGA\_IN0))/A_{PGA} \qquad \textit{Eqn. 12-1}$$

Chapter 13
Scalable Controller Area Network (S12MSCANV2)

Table 13-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V02.14	18 Sep 2002		Added Initialization/Application information; replaced ’MCU’ with ’CPU’ in several places; cleaned up Mode descriptions; general cleanup.
V02.15	15 Jul 2004		Corrected buffer read/write access definitions; corrected bit time equation.
V02.16	3 Jan 2005		Convert to SRSLite3.2 with single-source document for V02 and V03 (controlled by conditonal text)

13.1 Introduction

Scalable controller area network (S12MSCANV2) definition is based on the MSCAN12 definition, which is the specific implementation of the MSCAN concept targeted for the S12, S12X and S12Z microcontroller families.

The module is a communication controller implementing the CAN 2.0A/B protocol as defined in the Bosch specification dated September 1991. For users to fully understand the MSCAN specification, it is recommended that the Bosch specification be read first to familiarize the reader with the terms and concepts contained within this document.

Though not exclusively intended for automotive applications, CAN protocol is designed to meet the specific requirements of a vehicle serial data bus: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness, and required bandwidth.

MSCAN uses an advanced buffer arrangement resulting in predictable real-time behavior and simplified application software.

13.1.3 Features

The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol — Version 2.0A/B
  - — Standard and extended data frames
  - — Zero to eight bytes data length
  - — Programmable bit rate up to 1 Mbps[1]
  - — Support for remote frames
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a “local priority” concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter
- Programmable loopback mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (warning, error passive, bus-off)
- Programmable MSCAN clock source either bus clock or oscillator clock
- Internal timer for time-stamping of received and transmitted messages
- Three low-power modes: sleep, power down, and MSCAN enable
- Global initialization of configuration registers

13.1.4 Modes of Operation

For a description of the specific MSCAN modes and the module operation related to the system operating modes refer to Section 13.4.4, “Modes of Operation”.

1. Depending on the actual bit timing and the clock jitter of the PLL.

13.3.3.1.1 IDR0–IDR3 for Extended Identifier Mapping

Module Base + 0x00X0

	7	6	5	4	3	2	1	0
R W	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
Reset:	x	x	x	x	x	x	x	x

Figure 13-25. Identifier Register 0 (IDR0) — Extended Identifier Mapping

Table 13-26. IDR0 Register Field Descriptions — Extended

Field	Description
7-0 ID[28:21]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
Reset:	x	x	x	x	x	x	x	x

Figure 13-26. Identifier Register 1 (IDR1) — Extended Identifier Mapping

Table 13-27. IDR1 Register Field Descriptions — Extended

Field	Description
7-5 ID[20:18]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
4 SRR	**Substitute Remote Request** — This fixed recessive bit is used only in extended format. It must be set to 1 by the user for transmission buffers and is stored as received on the CAN bus for receive buffers.
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)
2-0 ID[17:15]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

13.4.3.1 Protocol Violation Protection

The MSCAN protects the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the MSCAN cannot be modified while the MSCAN is on-line. The MSCAN has to be in Initialization Mode. The corresponding INITRQ/INITAK handshake bits in the CANCTL0/CANCTL1 registers (see Section 13.3.2.1, “MSCAN Control Register 0 (CANCTL0)”) serve as a lock to protect the following registers:
  - MSCAN control 1 register (CANCTL1)
  - MSCAN bus timing registers 0 and 1 (CANBTR0, CANBTR1)
  - MSCAN identifier acceptance control register (CANIDAC)
  - MSCAN identifier acceptance registers (CANIDAR0–CANIDAR7)
  - MSCAN identifier mask registers (CANIDMR0–CANIDMR7)
- The TXCAN is immediately forced to a recessive state when the MSCAN goes into the power down mode or initialization mode (see Section 13.4.5.6, “MSCAN Power Down Mode,” and Section 13.4.4.5, “MSCAN Initialization Mode”).
- The MSCAN enable bit (CANE) is writable only once in normal system operation modes, which provides further protection against inadvertently disabling the MSCAN.

13.4.3.2 Clock System

Figure 13-42 shows the structure of the MSCAN clock generation circuitry.

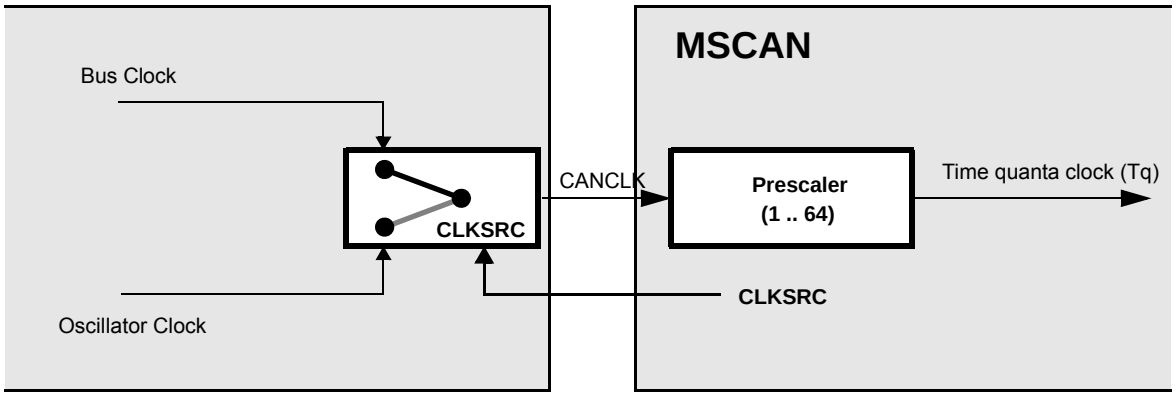


Figure 13-42. MSCAN Clocking Scheme

The clock source bit (CLKSRC) in the CANCTL1 register (13.3.2.2/13-383) defines whether the internal CANCLK is connected to the output of a crystal oscillator (oscillator clock) or to the bus clock.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met. Additionally, for high CAN bus rates (1 Mbps), a 45% to 55% duty cycle of the clock is required.

If the bus clock is generated from a PLL, it is recommended to select the oscillator clock rather than the bus clock due to jitter considerations, especially at the faster CAN bus rates.

and Table 17-6.

17.3.2.8 PWM Scale A Register (PWMSCLA)

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

Clock SA = Clock A / (2 * PWMSCLA)

NOTE

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

Module Base + 0x0008

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

Figure 17-10. PWM Scale A Register (PWMSCLA)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLA value)

17.3.2.9 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

NOTE

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

Module Base + 0x0009

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

Figure 17-11. PWM Scale B Register (PWMSCLB)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLB value).

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0007 Reserved	R								
	W								

= Unimplemented or Reserved

Figure 19-2. SPI Register Summary

19.3.2 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.

19.3.2.1 SPI Control Register 1 (SPICR1)

Module Base +0x0000

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 19-3. SPI Control Register 1 (SPICR1)

Read: Anytime

Write: Anytime

Table 19-2. SPICR1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode. 1 SPI is in master mode.

The main element of the SPI system is the SPI data register. The n-bit[1] data register in the master and the n-bit[1] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[1] register. When a data transfer operation is performed, this 2n-bit[1] register is serially shifted n[1] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 19.4.3, “Transmission Formats”).

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

19.4.1 Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{SS}$ pin

  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.

1. n depends on the selected transfer width, please refer to Section 19.3.2.2, “SPI Control Register 2 (SPICR2)

```
MASTX   TST    TXCNT          ;GET VALUE FROM THE TRANSMITING COUNTER
        BEQ    END            ;END IF NO MORE DATA
        BRSET  IBSR,#$01,END  ;END IF NO ACK
        MOVB   DATABUF,IBDR   ;TRANSMIT NEXT BYTE OF DATA
        DEC    TXCNT          ;DECREASE THE TXCNT
        BRA    EMASTX         ;EXIT
END     BCLR   IBCR,#$20      ;GENERATE A STOP CONDITION
EMASTX  RTI                   ;RETURN FROM INTERRUPT
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data which can be done by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. The following is an example showing how a STOP signal is generated by a master receiver.

```
MASR    DEC    RXCNT          ;DECREASE THE RXCNT
        BEQ    ENMASR         ;LAST BYTE TO BE READ
        MOVB   RXCNT,D1       ;CHECK SECOND LAST BYTE
        DEC    D1             ;TO BE READ
        BNE    NXMAR          ;NOT LAST OR SECOND LAST
LAMAR   BSET   IBCR,#$08      ;SECOND LAST, DISABLE ACK
                              ;TRANSMITTING
        BRA    NXMAR
ENMASR  BCLR   IBCR,#$20      ;LAST ONE, GENERATE ‘STOP’ SIGNAL
NXMAR   MOVB   IBDR,RXBUF     ;READ DATA AND STORE
        RTI
```

20.7.1.5 Generation of Repeated START

At the end of data transfer, if the master continues to want to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. A program example is as shown.

```
RESTART BSET   IBCR,#$04      ;ANOTHER START (RESTART)
        MOVB   CALLING,IBDR   ;TRANSMIT THE CALLING ADDRESS;D0=R/W
```

20.7.1.6 Slave Mode

In the slave interrupt service routine, the module addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the transmit/receive mode select bit (Tx/Rx bit of IBCR) according to the R/W command bit (SRW). Writing to the IBCR clears the IAAS automatically. Note that the only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred, interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer may now be initiated by writing information to IBDR, for slave transmits, or dummy reading from IBDR, in slave receive mode. The slave will drive SCL low in-between byte transfers, SCL is released when the IBDR is accessed in the required mode.

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. Setting RXAK means an 'end of data' signal from the master receiver, after which it must be switched from transmitter mode to receiver mode by software. A dummy read then releases the SCL line so that the master can generate a STOP signal.

Table B-1. Voltage Regulator Electrical Characteristics

-40°C ≤T_J ≤ 175°C unless noted otherwise, V_{DDA} and V_{DDX} must be shorted on the application board.						
Num	**Characteristic**	**Symbol**	**Min**	**Typical**	**Max**	**Unit**
19a	V_{BG} voltage Distribution over temperature T_J V_{SUP} = 12V, -40°C ≤ T_J ≤ 150°C, VDDX = 5V	Δ_{VBGV5V}	-20		20	mV
19b	V_{BG} voltage Distribution over temperature T_J V_{SUP} = 12V, -40°C ≤ T_J ≤ 150°C, VDDX= 3.3V	$\Delta_{VBGV3V3}$	-25		25	mV
20	Base Current For External PNP (V_{DDX})[10] -40°C ≤ T_J ≤ 150°C 150°C <T_J < 175°C	$I_{BCTLMAX}$	 2.3 1.5	 — —	 — —	mA
21	Recovery time from STOP	t_{STP_REC}	—	23	—	µs

1 3% Vreg tolerance

2 Please note that the core current is derived from V_{DDX}

3 Further limitation may apply due to maximum allowable T_J

4 2% Vreg tolerance, MC9S12ZVLA device only

5 LVI is monitored on the V_{DDA} supply domain

6 LVRX is monitored on the V_{DDX} supply domain only active during full performance mode. During reduced performance mode (stopmode) voltage supervision is solely performed by the POR block monitoring core V_{DD}.

7 the ACLK is not factory trimmed, the customer can use the CPMUACLKTR register to trim the ACLK, see also 9.3.2.19, “Autonomous Clock Trimming Register (CPMUACLKTR)

8 The ACLK trimming must be set that the minimum period equals to 0.2ms

9 CPMUHTTR=0x88

10 This is the minimum base current that can be guaranteed when the external PNP is delivering maximum current.

NOTE

The LVR monitors the voltages VDD, VDDF and VDDX. If the voltage drops on these supplies to a level which could prohibit the correct function (e.g. code execution) of the micro controller, the LVR triggers.

Table B-2. Recommended Capacitor Values

V_{DDA} and V_{DDX} must be shorted on the application board.				
Num	**Characteristic**	**Symbol**	**Typical[1]**	**Unit**
1	VDDX capacitor[2]	C_{VDDX}	100-220	nF
2	VDDA capacitor[3]	C_{VDDA}	100-220	nF
3	Stability capacitor[4,5]	C_{VDD5}	4.7-10	µF

[1]Values are nominal component values

[2]X7R ceramics

[3]X7R ceramics

[4]Can be placed anywhere on the 5V supply node (VDDA, VDDX)

[5]4.7µF X7R ceramics or 10µF tantalum

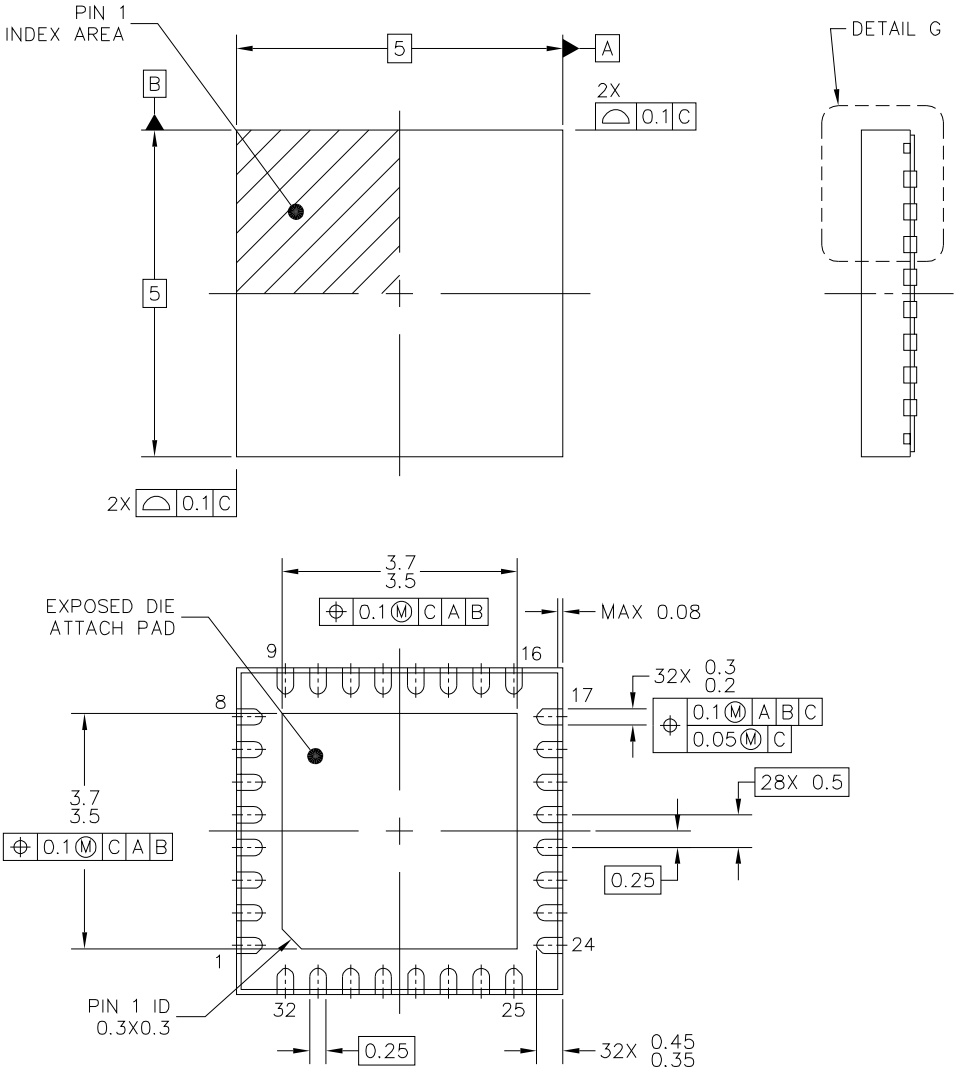
In Table K-3 the timing characteristics for slave mode are listed.

Table K-3. SPI Slave Mode Timing Characteristics

Num	Characteristic	Symbol	Min	Typ	Max	Unit
1	SCK Frequency	f_{sck}	DC	—	1/4	f_{bus}
1	SCK Period	t_{sck}	4	—	∞	t_{bus}
2	Enable Lead Time	t_L	4	—	—	t_{bus}
3	Enable Trail Time	t_T	4	—	—	t_{bus}
4	Clock (SCK) High or Low Time	t_{wsck}	4	—	—	t_{bus}
5	Data Setup Time (Inputs)	t_{su}	8	—	—	ns
6	Data Hold Time (Inputs)	t_{hi}	8	—	—	ns
7	Slave Access Time (time to data active)	t_a	—	—	20	ns
8	Slave MISO Disable Time	t_{dis}	—	—	22	ns
9	Data Valid after SCK Edge	t_{vsck}	—	—	$28 + 0.5 \cdot t_{bus}$ [1]	ns
10	Data Valid after $\overline{SS}$ fall	t_{vss}	—	—	$28 + 0.5 \cdot t_{bus}$ [1]	ns
11	Data Hold Time (Outputs)	t_{ho}	20	—	—	ns
12	Rise and Fall Time Inputs	t_{rfi}	—	—	8	ns
13	Rise and Fall Time Outputs	t_{rfo}	—	—	8	ns

[1] $0.5t_{bus}$ added due to internal synchronization delay

M.3 32 QFN, MC9S12ZVLS device only



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QFN, THERMALLY ENHANCED, 5 X 5 X 0.85, 0.5 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA00656D	REV: O
	STANDARD: NON-JEDEC	
	10 DEC 2013	