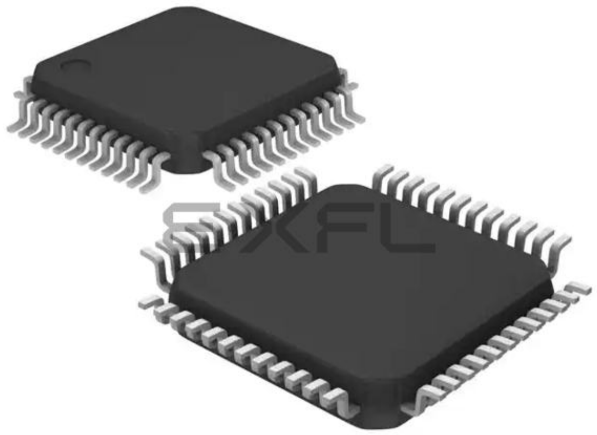
NXP USA Inc. - S912ZVL64F0VLF Datasheet



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	64KB (64K 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/s912zvl64f0vlf

Table 1-2. MC9S12ZVL-Family Comparison

Feature	MC9S12ZVL128 MC9S12ZVLA128	MC9S12ZVL96 MC9S12ZVLA96	MC9S12ZVL64 MC9S12ZVLA64	MC9S12ZVL32	MC9S12ZVL16	MC9S12ZVL8	MC9S12ZVLS32	MC9S12ZVLS16
SCI[6]	2			2			2	
SPI	1			1			1	
IIC	1			1			1	
MSCAN	1			-			-	
max SRAM_ECC access width	4 Byte			2 Byte			2 Byte	
Supported ADC option bits	yes			no			no	
General purpose I/O - pin to support 25 mA driver strength to VSSX - pin to support 20 mA driver strength from VDDX (EVDD)	34[3] / 19 3[3] / 1 1			34[3] / 19 3[3] / 1 1			18 3 1	
Interrupt capable pins[7] 5V / 12V	22[3] / 16 / 1			22[3] / 16 / 1			14 / 1	

[1] total current capability for MCU and MCU - external loads (on same PCB - board)

[2] MC9S12ZVLA device only

[3] available in 48-pin packages only

[4] to internally feed the ACMP or bonded out in 48-LQFP

[5] only 5V operation mode supported

[6] one SCI routed to the LINPHY

[7] $\overline{\text{IRQ}}$ / $\overline{\text{XIRQ}}$ and KWx pins

NOTE

After power up, the MC9S12ZVL(A)128/96/64 devices starts in 3.3V VDDX mode. Then is possible to switch to the 5.0V VDDX behavior. For more details see the “Clock, Reset and Power Management Unit” section, 9.3.2.27, “Voltage Regulator Control Register (CPMUVREGCTL)

1.3 Chip-Level Features

On-chip modules available within the family include the following features:

- S12Z CPU core
- 128, 96, 64, 32, 16 or 8 KB on-chip flash with ECC
- 2048, 1024, 128 byte EEPROM with ECC
- 8192, 4096, 1024 or 512 byte on-chip SRAM with ECC
- Phase locked loop (IPLL) frequency multiplier with internal filter
- 1 MHz internal RC oscillator with +/-1.3% accuracy over rated temperature range
- 4-20 MHz amplitude controlled pierce oscillator

- Instructions and Addressing modes optimized for C-Programming & Compiler
  - MAC unit 32bit += 32bit*32bit
  - Hardware divider
  - Single cycle multi-bit shifts (Barrel shifter)
  - Special instructions for fixed point math
- Unimplemented opcode traps
- Unprogrammed byte value (0xFF) defaults to SWI instruction

1.4.1.1 Background Debug Controller (BDC)

- Background debug controller (BDC) with single-wire interface
  - Non-intrusive memory access commands
  - Supports in-circuit programming of on-chip nonvolatile memory

1.4.1.2 Debugger (DBG)

- Three comparators (A, B and D)
  - Comparator A compares the full address bus and full 32-bit data bus
  - Comparators B and D compare the full address bus only
  - Each comparator can be configured to monitor PC addresses or addresses of data accesses
  - Each comparator can select either read or write access cycles
  - Comparator matches can force state sequencer state transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax
- State sequencer control
  - State transitions forced by comparator matches
  - State transitions forced by software write to TRIG
  - State transitions forced by an external event
- The following types of breakpoints
  - CPU breakpoint entering active BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)

1.4.2 Embedded Memory

1.4.2.1 Memory Access Integrity

- Illegal address detection
- ECC support on embedded NVM and system RAM

1.4.2.2 Flash

On-chip flash memory on the MC9S12ZVL-Family

- Up to 128 KB of program flash memory
  - Automated program and erase algorithm
  - Protection scheme to prevent accidental program or erase

1.4.2.3 EEPROM

- Up to 2048 bytes EEPROM
  - 16 data bits plus 6 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 4 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

1.4.2.4 SRAM

- Up to 8 KB of general-purpose RAM with ECC
  - Single bit error correction and double bit error detection code based on 16-bit data words

1.4.3 Clocks, Reset & Power Management Unit (CPMU)

- Real time interrupt (RTI)
- Clock monitor, supervising the correct function of the oscillator (CM)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Can be initialized out of reset using option bits located in flash memory
- System reset generation
- Autonomous periodic interrupt (API) (combination with cyclic, watchdog)
- Low Power Operation
  - RUN mode is the main full performance operating mode with the entire device clocked.
  - WAIT mode when the internal CPU clock is switched off, so the CPU does not execute instructions.
  - Pseudo STOP - system clocks are stopped but the oscillator the RTI, the COP, and API modules can be enabled
  - STOP - the oscillator is stopped in this mode, all clocks are switched off and all counters and dividers remain frozen, with the exception of the COP and API which can optionally run from ACLK.

1.4.3.1 Internal Phase-Locked Loop (IPLL)

- Phase-locked-loop clock frequency multiplier

Table 2-4. MODRR1 Routing Register Field Descriptions

Field	Description
4 PWM4RR	**Module Routing Register** — PWM option 4 routing 1 PWM option 4 to PS0 0 PWM option 4 to PP4
2 PWM2RR	**Module Routing Register** — PWM option 2 routing 1 PWM option 2 to PT0 0 PWM option 2 to PP2
0 PWM0RR	**Module Routing Register** — PWM option 0 routing 1 PWM option 0 to PT1 0 PWM option 0 to PP0

2.3.2.3 Module Routing Register 2 (MODRR2)

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

	7	6	5	4	3	2	1	0
R	T1C1RR	T1C0RR	T0C5RR	T0C4RR	T0C3RR	T0C2RR	0	0
W							▨	▨
	IOC1_1	IOC1_0	IOC0_5	IOC0_4	IOC0_3	IOC0_2	—	—
Reset	0	0	0	0	0	0	0	0

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

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

Table 2-5. MODRR2 Routing Register Field Descriptions

Field	Description
7 T1C1RR	**Module Routing Register** — IOC1_1 routing 1 IOC1_1 to PP1 0 IOC1_1 to PT7
6 T1C0RR	**Module Routing Register** — IOC1_0 routing 1 IOC1_0 to PP7 0 IOC1_0 to PT6
5 T0C5RR	**Module Routing Register** — IOC0_5 routing 1 IOC0_5 to PS3 0 IOC0_5 to PT5
4 T0C4RR	**Module Routing Register** — IOC0_4 routing 1 IOC0_4 to PS2 0 IOC0_4 to PT4

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0140	DBGDCTL	R	0	0	INST	0	RW	RWE	reserved	COMPE
		W								
0x0141- 0x0144	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0145	DBGDAH	R	DBGDA[23:16]							
		W								
0x0146	DBGDAM	R	DBGDA[15:8]							
		W								
0x0147	DBGDAL	R	DBGDA[7:0]							
		W								
0x0148- 0x017F	Reserved	R	0	0	0	0	0	0	0	0
		W								

Figure 7-2. Quick Reference to DBG Registers

7.3.2 Register Descriptions

This section consists of the DBG register descriptions in address order. When ARM is set in DBGC1, the only bits in the DBG module registers that can be written are ARM, and TRIG

7.3.2.1 Debug Control Register 1 (DBGC1)

Address: 0x0100

	7	6	5	4	3	2	1	0
0x0100	ARM	0	reserved	BDMBP	BRKCPU	reserved	EEVE1	0
		TRIG						
Reset	0	0	0	0	0	0	0	0

Figure 7-3. Debug Control Register (DBGC1)

Read: Anytime

Write: Bit 7 Anytime . An ongoing profiling session must be finished before DBG can be armed again.
Bit 6 can be written anytime but always reads back as 0.
Bits 5:0 anytime DBG is not armed.

NOTE

On a write access to DBGC1 and simultaneous hardware disarm from an internal event, the hardware disarm has highest priority, clearing the ARM bit and generating a breakpoint, if enabled.

Table 9-28. TC trimming of the frequency of the IRC1M at ambient temperature

TCTRIM[4:0]	IRC1M Indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00000	0 (nominal TC of the IRC)	0%
00001	-0.27%	-0.5%
00010	-0.54%	-0.9%
00011	-0.81%	-1.3%
00100	-1.08%	-1.7%
00101	-1.35%	-2.0%
00110	-1.63%	-2.2%
00111	-1.9%	-2.5%
01000	-2.20%	-3.0%
01001	-2.47%	-3.4%
01010	-2.77%	-3.9%
01011	-3.04	-4.3%
01100	-3.33%	-4.7%
01101	-3.6%	-5.1%
01110	-3.91%	-5.6%
01111	-4.18%	-5.9%
10000	0 (nominal TC of the IRC)	0%
10001	+0.27%	+0.5%
10010	+0.54%	+0.9%
10011	+0.81%	+1.3%
10100	+1.07%	+1.7%
10101	+1.34%	+2.0%
10110	+1.59%	+2.2%
10111	+1.86%	+2.5%
11000	+2.11%	+3.0%
11001	+2.38%	+3.4%
11010	+2.62%	+3.9%
11011	+2.89%	+4.3%
11100	+3.12%	+4.7%
11101	+3.39%	+5.1%
11110	+3.62%	+5.6%
11111	+3.89%	+5.9%

NOTE

Since the IRC1M frequency is not a linear function of the temperature, but more like a parabola, the above relative variation is only an indication and should be considered with care.

Table 10-11. Summary of Conversion Flow Control Bit Scenarios

RSTA	TRIG	SEQA	LDOK	Conversion Flow Control Mode	Conversion Flow Control Scenario
0	0	0	0	Both Modes	Valid
0	0	0	1	Both Modes	Can Not Occur
0	0	1	0	Both Modes	Valid[5]
0	0	1	1	Both Modes	Can Not Occur
0	1	0	0	Both Modes	Valid[2]
0	1	0	1	Both Modes	Can Not Occur
0	1	1	0	Both Modes	Can Not Occur
0	1	1	1	Both Modes	Can Not Occur
1	0	0	0	Both Modes	Valid[4]
1	0	0	1	Both Modes	Valid[1,4]
1	0	1	0	Both Modes	Valid[3,4,5]
1	0	1	1	Both Modes	Valid[1,3,4,5]
1	1	0	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2,4,6]
1	1	0	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[1,2,4,6]
1	1	1	0	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[2,3,4,5,6]
1	1	1	1	“Restart Mode”	Error flag TRIG_EIF set
				“Trigger Mode”	Valid[1,2,3,4,5,6]

[1] Swap CSL buffer

[2] Start conversion sequence

[3] Prevent RSTA_EIF and LDOK_EIF

[4] Load conversion command from top of CSL

[5] Abort any ongoing conversion, conversion sequence and CSL

[6] Bit TRIG set automatically in Trigger Mode

For a detailed description of all conversion flow control bit scenarios please see also Section 10.6.3.2.4, “The two conversion flow control Mode Configurations, Section 10.6.3.2.5, “The four ADC conversion flow control bits and Section 10.6.3.2.6, “Conversion flow control in case of conversion sequence control bit overrun scenarios

10.6.3.2.6 Conversion flow control in case of conversion sequence control bit overrun scenarios

Restart Request Overrun:
If a legal Restart Request is detected and no Restart Event is in progress, the RSTA bit is set due to the request. The set RSTA bit indicates that a Restart Request was detected and the Restart Event is in process. In case further Restart Requests occur while the RSTA bit is set, this is defined a overrun situation. This scenario is likely to occur when bit STR_SEQA is set or when a Restart Event causes a Sequence Abort Event. The request overrun is captured in a background register that always stores the last detected overrun request. Hence if the overrun situation occurs more than once while a Restart Event is in progress, only the latest overrun request is pending. When the RSTA bit is cleared, the latest overrun request is processed and RSTA is set again one cycle later.

LoadOK Overrun:
Simultaneously at any Restart Request overrun situation the LoadOK input is evaluated and the status is captured in a background register which is alternated anytime a Restart Request Overrun occurs while Load OK Request is asserted. The Load OK background register is cleared as soon as the pending Restart Request gets processed.

Trigger Overrun:
If a Trigger occurs whilst bit TRIG is already set, this is defined as a Trigger overrun situation and causes the ADC to cease conversion at the next conversion boundary and to set bit TRIG_EIF. A overrun is also detected if the Trigger Event occurs automatically generated by hardware in “Trigger Mode” due to a Restart Event and simultaneously a Trigger Event is generated via data bus or internal interface. In this case the ADC ceases operation before conversion begins to sample. In “Trigger Mode” a Restart Request Overrun does not cause a Trigger Overrun (bit TRIG_EIF not set).

Sequence Abort Request Overrun:
If a Sequence Abort Request occurs whilst bit SEQA is already set, this is defined as a Sequence Abort Request Overrun situation and the overrun request is ignored.

Table 12-6. Amplifier Gain

PGAGAIN[3:0]	Gain A_{PGA}
0000	10x
0001	20x
0010	40x
0011	80x
others	Reserved

12.3.2.4 PGA Offset Register (PGAOFFSET)

Module Base + 0x0003 Access: User read/write[1]

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

Figure 12-6. PGA Offset Register (PGAOFFSET

[1] Read: Anytime
Write: Anytime

Table 12-7. PGAOFFSET Field Description

Field	Description
5:0 PGAOFFSET [5:0]	**PGA Offset** — These register bits select the offset correction for the PGA, see Table 12-8., “Offset Compensation and 12.4.2, “Offset Compensation.

Table 12-8. Offset Compensation

PGAOFFSET[5:3]	ΔV_{OUT}
0x011	$- 3*V_{step\_H}$
0x010	$- 2*V_{step\_H}$
0x001	$- 1*V_{step\_H}$
0x000	0
0x111	$+ 1*V_{step\_H}$
0x110	$+ 2*V_{step\_H}$
0x101	$+ 3*V_{step\_H}$
0x100	0

PGAOFFSET[2:0]	ΔV_{OUT}
0x011	$- 3*V_{step\_L}$
0x010	$- 2*V_{step\_L}$
0x001	$- 1*V_{step\_L}$
0x000	0
0x111	$+ 1*V_{step\_L}$
0x110	$+ 2*V_{step\_L}$
0x101	$+ 3*V_{step\_L}$
0x100	0

[1] Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 13-14. CANTIER Register Field Descriptions

Field	Description
2-0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

13.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
W	▨	▨	▨	▨	▨			
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 13-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)

[1] Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 13-15. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	**Abort Request** — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 13.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”) and abort acknowledge flags (ABTAK, see Section 13.3.2.10, “MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)”) are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

13.3.2.15 MSCAN Transmit Error Counter (CANTXERR)

This register reflects the status of the MSCAN transmit error counter.

Module Base + 0x000F Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 13-18. MSCAN Transmit Error Counter (CANTXERR)

[1] Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

NOTE

Reading this register when in any other mode other than sleep or initialization mode, may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

13.3.2.16 MSCAN Identifier Acceptance Registers (CANIDAR0-7)

On reception, each message is written into the background receive buffer. The CPU is only signalled to read the message if it passes the criteria in the identifier acceptance and identifier mask registers (accepted); otherwise, the message is overwritten by the next message (dropped).

The acceptance registers of the MSCAN are applied on the IDR0–IDR3 registers (see Section 13.3.3.1, “Identifier Registers (IDR0–IDR3)”) of incoming messages in a bit by bit manner (see Section 13.4.3, “Identifier Acceptance Filter”).

For extended identifiers, all four acceptance and mask registers are applied. For standard identifiers, only the first two (CANIDAR0/1, CANIDMR0/1) are applied.

Module Base + 0x0010 to Module Base + 0x0013 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 13-19. MSCAN Identifier Acceptance Registers (First Bank) — CANIDAR0–CANIDAR3

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

In cases of more than one buffer having the same lowest priority, the message buffer with the lower index number wins.

Module Base + 0x00XD Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PRIO7	PRIO6	PRIO5	PRIO4	PRIO3	PRIO2	PRIO1	PRIO0
Reset:	0	0	0	0	0	0	0	0

Figure 13-35. Transmit Buffer Priority Register (TBPR)

[1] Read: Anytime when TXEx flag is set (see Section 13.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 13.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)")
Write: Anytime when TXEx flag is set (see Section 13.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 13.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)")

13.3.3.5 Time Stamp Register (TSRH–TSRL)

If the TIME bit is enabled, the MSCAN will write a time stamp to the respective registers in the active transmit or receive buffer right after the EOF of a valid message on the CAN bus (see Section 13.3.2.1, "MSCAN Control Register 0 (CANCTL0)"). In case of a transmission, the CPU can only read the time stamp after the respective transmit buffer has been flagged empty.

The timer value, which is used for stamping, is taken from a free running internal CAN bit clock. A timer overrun is not indicated by the MSCAN. The timer is reset (all bits set to 0) during initialization mode. The CPU can only read the time stamp registers.

Module Base + 0x00XE Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
W								
Reset:	x	x	x	x	x	x	x	x

Figure 13-36. Time Stamp Register — High Byte (TSRH)

[1] Read: For transmit buffers: Anytime when TXEx flag is set (see Section 13.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 13.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)"). For receive buffers: Anytime when RXF is set.
Write: Unimplemented

The MSCAN facilitates a sophisticated message storage system which addresses the requirements of a broad range of network applications.

13.4.2.1 Message Transmit Background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the “local priority” concept described in Section 13.4.2.2, “Transmit Structures.”

13.4.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 13-38.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Section 13.3.3, “Programmer’s Model of Message Storage”). An additional Transmit Buffer Priority Register (TBPR) contains an 8-bit local priority field (PRIO) (see Section 13.3.3.4, “Transmit Buffer Priority Register (TBPR)”). The remaining two bytes are used for time stamping of a message, if required (see Section 13.3.3.5, “Time Stamp Register (TSRH–TSRL)”).

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see Section 13.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see Section 13.3.2.11, “MSCAN Transmit Buffer Selection Register (CANTBSEL)”). This makes the respective buffer accessible within the CANTXFG address space (see Section 13.3.3, “Programmer’s Model of Message Storage”). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0006 PWMCLKAB[1]	R	PCLKAB7	PCLKAB6	PCLKAB5	PCLKAB4	PCLKAB3	PCLKAB2	PCLKAB1	PCLKAB0
	W								
0x0007 RESERVED	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0008 PWMSCLA	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0009 PWMSCLB	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x000A RESERVED	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x000B RESERVED	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x000C PWMCNT0[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000D PWMCNT1[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000E PWMCNT2[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000F PWMCNT3[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0010 PWMCNT4[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0011 PWMCNT5[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0012 PWMCNT6[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0013 PWMCNT7[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0014 PWMPER0[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								

▨ = Unimplemented or Reserved

Figure 17-2. The scalable PWM Register Summary (Sheet 2 of 4)

NOTE

> Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

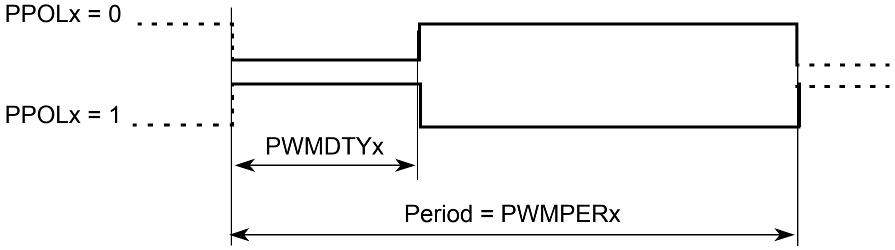


Figure 17-17. PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - — Polarity = 0 (PPOLx = 0)

    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - — Polarity = 1 (PPOLx = 1)

    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

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

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/4 = 2.5 MHz

PWMx Period = 400 ns

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

The output waveform generated is shown in Figure 17-18.

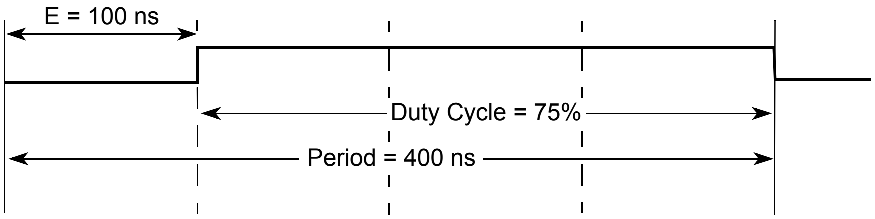


Figure 17-18. PWM Left Aligned Output Example Waveform

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

Table 17-13 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 17-13. 16-bit Concatenation Mode Summary

Note: Bits related to available channels have functional significance.

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON67	PWME7	PPOL7	PCLK7	CAE7	PWM7
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

17.4.2.8 PWM Boundary Cases

Table 17-14 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation).

Table 17-14. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always low
$00 (indicates no duty)	>$00	0	Always high
XX	$00[1] (indicates no period)	1	Always high
XX	$00[1] (indicates no period)	0	Always low
>= PWMPERx	XX	1	Always high
>= PWMPERx	XX	0	Always low

[1] Counter = $00 and does not count.

17.5 Resets

The reset state of each individual bit is listed within the Section 17.3.2, “Register Descriptions” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters do not count.

Table 19-4. SPICR2 Field Descriptions

Field	Description
6 XFRW	**Transfer Width —** This bit is used for selecting the data transfer width. If 8-bit transfer width is selected, SPIDRL becomes the dedicated data register and SPIDRH is unused. If 16-bit transfer width is selected, SPIDRH and SPIDRL form a 16-bit data register. Please refer to Section 19.3.2.4, "SPI Status Register (SPISR) for information about transmit/receive data handling and the interrupt flag clearing mechanism. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 8-bit Transfer Width (n = 8)[1] 1 16-bit Transfer Width (n = 16)[1]
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration, refer to Table 19-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 $\overline{SS}$ port pin is not used by the SPI. 1 $\overline{SS}$ port pin with MODF feature.
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled. 1 Output buffer enabled.
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in wait mode. 1 Stop SPI clock generation when in wait mode.
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 19-5. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.

[1] n is used later in this document as a placeholder for the selected transfer width.

Table 19-5. Bidirectional Pin Configurations

Pin Mode	SPC0	BIDIROE	MISO	MOSI
Master Mode of Operation				
Normal	0	X	Master In	Master Out
Bidirectional	1	0	MISO not used by SPI	Master In
		1		Master I/O
Slave Mode of Operation				
Normal	0	X	Slave Out	Slave In
Bidirectional	1	0	Slave In	MOSI not used by SPI
		1	Slave I/O	

Table 22-38. Read Once Command FCCOB Requirements

Register	FCCOB Parameters
FCCOB3	Read Once word 1 value
FCCOB4	Read Once word 2 value
FCCOB5	Read Once word 3 value

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag will set after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block will return invalid data.

Table 22-39. Read Once Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

22.4.7.5 Program P-Flash Command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

CAUTION

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

Table 22-40. Program P-Flash Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x06	Global address [23:16] to identify P-Flash block
FCCOB1	Global address [15:0] of phrase location to be programmed[1]	
FCCOB2	Word 0 program value	
FCCOB3	Word 1 program value	
FCCOB4	Word 2 program value	
FCCOB5	Word 3 program value	

[1] Global address [2:0] must be 000

Table E-2. NVM Timing Characteristics ZVL(A)128/96/64

Num	Command	f_{NVMOP} cycle	f_{NVMBUS} cycle	Symbol	Min[1]	Typ[2]	Max[3]	Lfmax[4]	Unit
1	Bus frequency	1	—	f_{NVMBUS}	1	32	32		MHz
2	NVM Operating frequency	—	1	f_{NVMOP}	0.8	1	1.05		MHz
3	Erase Verify All Blocks[5,6]	0	34528	t_{RD1ALL}	1.08	1.08	2.16	69.06	ms
4	Erase Verify Block (Pflash)[5]	0	33323	t_{RD1BLK_P}	1.04	1.04	2.08	66.65	ms
5	Erase Verify Block (EEPROM)[6]	0	1591	t_{RD1BLK_D}	0.05	0.05	0.10	3.18	ms
6	Erase Verify P-Flash Section	0	508	t_{RD1SEC}	0.02	0.02	0.03	1.02	ms
7	Read Once	0	481	t_{RDONCE}	15.03	15.03	15.03	481.00	us
8	Program P-Flash (4 Word)	164	3133	t_{PGM_4}	0.25	0.26	0.56	12.74	ms
9	Program Once	164	3107	$t_{PGMONCE}$	0.25	0.26	0.26	3.31	ms
10	Erase All Blocks[5,6]	100066	34991	t_{ERSALL}	96.39	101.16	102.25	195.06	ms
11	Erase Flash Block (Pflash)[5]	100060	33692	t_{ERSBLK_P}	96.35	101.11	102.17	192.46	ms
12	Erase Flash Block (EEPROM)[6]	100060	1930	t_{ERSBLK_D}	95.36	100.12	100.18	128.94	ms
13	Erase P-Flash Sector	20015	924	t_{ERSPG}	19.09	20.04	20.07	26.87	ms
14	Unsecure Flash	100066	35069	t_{UNSECU}	96.40	101.16	102.26	195.22	ms
15	Verify Backdoor Access Key	0	493	t_{VFYKEY}	15.41	15.41	15.41	493.00	us
16	Set User Margin Level	0	436	t_{MLOADU}	13.63	13.63	13.63	436.00	us
17	Set Factory Margin Level	0	445	t_{MLOADF}	13.91	13.91	13.91	445.00	us
18	Erase Verify EEPROM Section	0	583	$t_{DRD1SEC}$	0.02	0.02	0.04	1.17	ms
19	Program EEPROM (1 Word)	68	1678	t_{DPGM_1}	0.12	0.12	0.28	6.80	ms
20	Program EEPROM (2 Word)	136	2702	t_{DPGM_2}	0.21	0.22	0.47	10.98	ms
21	Program EEPROM (3 Word)	204	3726	t_{DPGM_3}	0.31	0.32	0.67	15.16	ms
22	Program EEPROM (4 Word)	272	4750	t_{DPGM_4}	0.41	0.42	0.87	19.34	ms
23	Erase EEPROM Sector	5015	817	t_{DERSPG}	4.80	5.04	20.49	38.96	ms
24	Protection Override	0	475	$t_{PRTOVRD}$	14.84	14.84	14.84	475.00	us

[1] Minimum times are based on maximum f_{NVMOP} and maximum f_{NVMBUS}

[2] Typical times are based on typical f_{NVMOP} and typical f_{NVMBUS}

[3] Maximum times are based on typical f_{NVMOP} and typical f_{NVMBUS} plus aging

[4] Lowest-frequency max times are based on minimum f_{NVMOP} and minimum f_{NVMBUS} plus aging

[5] Affected by Pflash size

[6] Affected by EEPROM size

E.2 NVM Reliability Parameters

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors and burn-in to screen early life failures.

How to Reach Us:

Home Page:

nxp.com

Web Support

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Document Number: MC9S12ZVLRM
Rev. 2.41
October 24, 2017