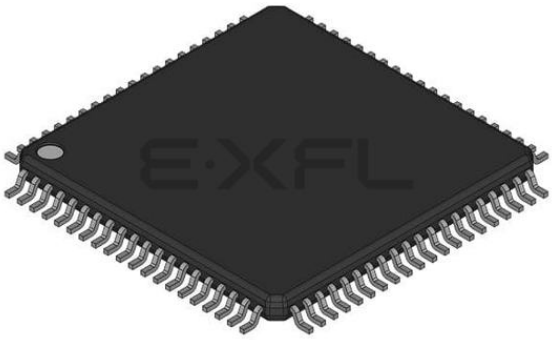


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

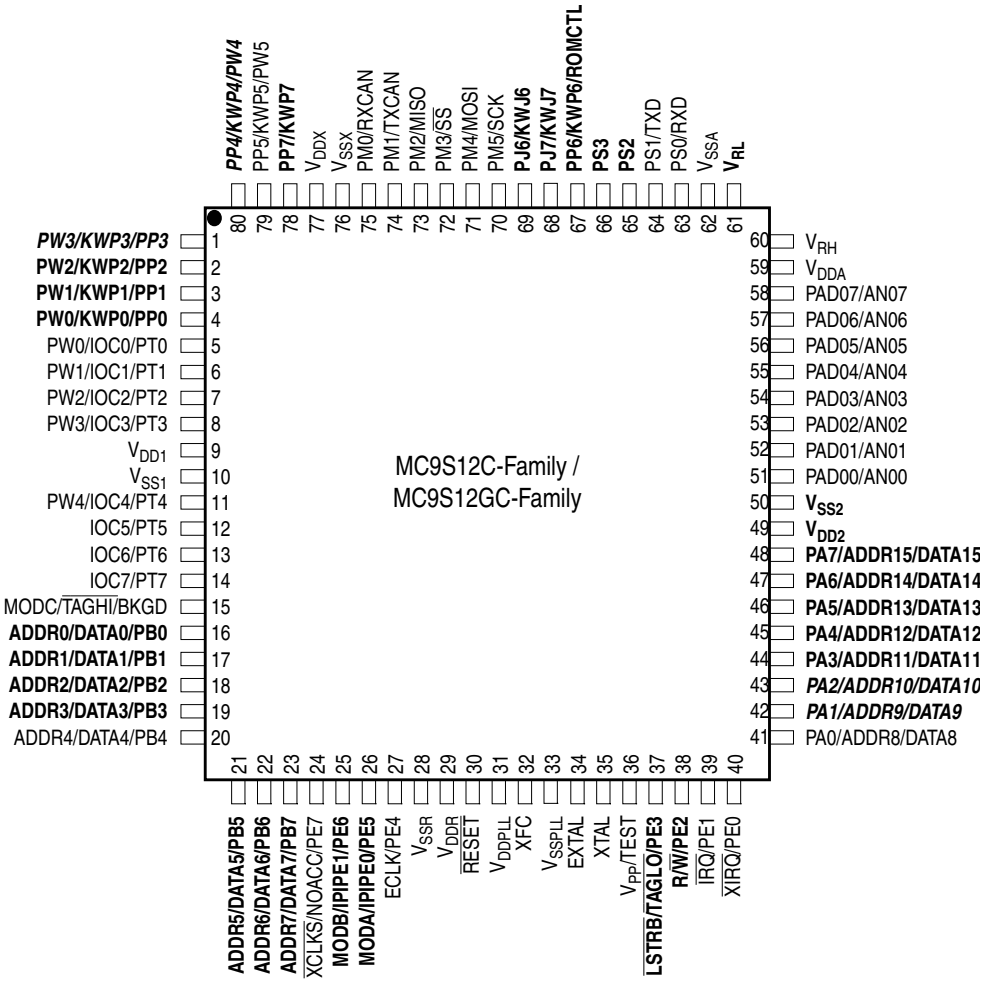
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12gc32mfue

1.3 Signal Description

1.3.1 Device Pinouts



Signals shown in **Bold** are not available on the 52- or 48-pin package
Signals shown in ***Bold Italic*** are available in the 52-pin, but not the 48-pin package

Figure 1-7. Pin Assignments in 80-Pin QFP

The MODRR register within the PIM allows for mapping of PWM channels to Port T in the absence of Port P pins for the low pin count packages. For the 80QFP package option it is recommended not to use MODRR since this is intended to support PWM channel availability in low pin count packages. Note that when mapping PWM channels to Port T in an 80QFP option, the associated PWM channels are then mapped to both Port P and Port T

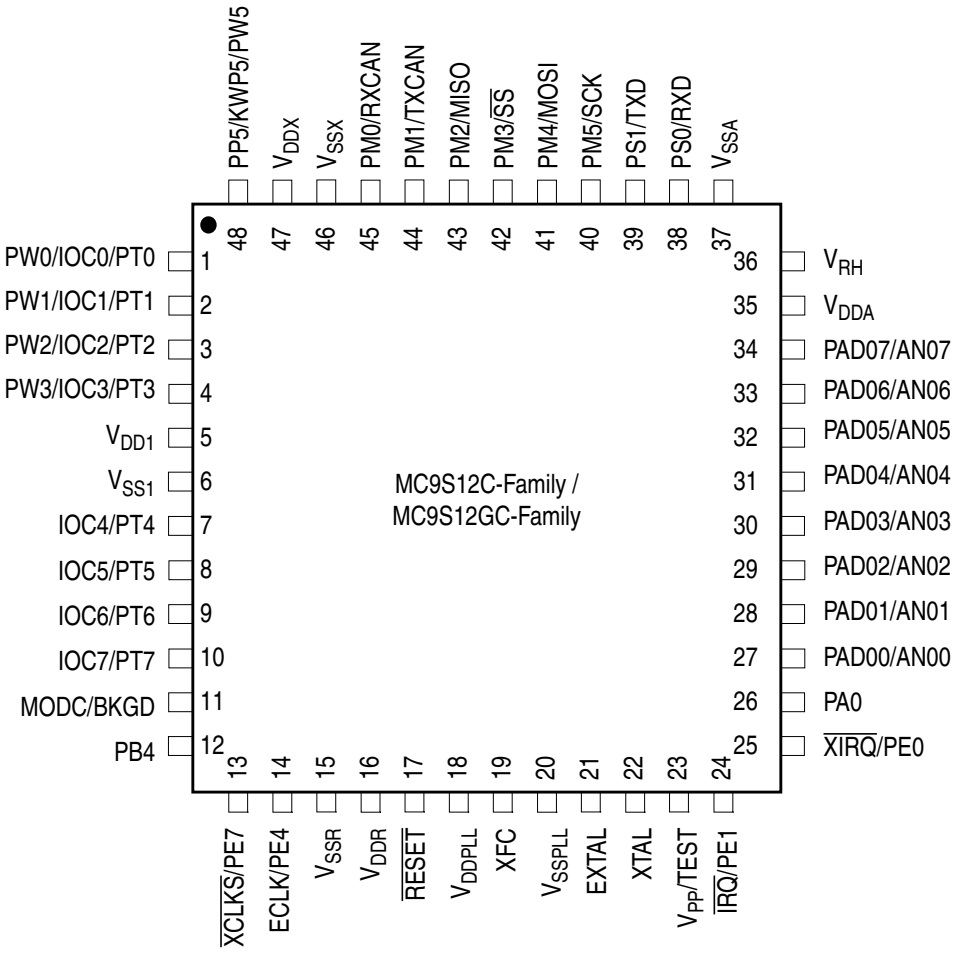


Figure 1-9. Pin Assignments in 48-Pin LQFP

4.1.2 Modes of Operation

- Normal expanded wide mode

  Ports A and B are configured as a 16-bit multiplexed address and data bus and port E provides bus control and status signals. This mode allows 16-bit external memory and peripheral devices to be interfaced to the system.
- Normal expanded narrow mode

  Ports A and B are configured as a 16-bit address bus and port A is multiplexed with 8-bit data. Port E provides bus control and status signals. This mode allows 8-bit external memory and peripheral devices to be interfaced to the system.
- Normal single-chip mode

  There is no external expansion bus in this mode. The processor program is executed from internal memory. Ports A, B, K, and most of E are available as general-purpose I/O.
- Special single-chip mode

  This mode is generally used for debugging single-chip operation, boot-strapping, or security related operations. The active background mode is in control of CPU execution and BDM firmware is waiting for additional serial commands through the BKGD pin. There is no external expansion bus after reset in this mode.
- Emulation expanded wide mode

  Developers use this mode for emulation systems in which the users target application is normal expanded wide mode.
- Emulation expanded narrow mode

  Developers use this mode for emulation systems in which the users target application is normal expanded narrow mode.
- Special test mode

  Ports A and B are configured as a 16-bit multiplexed address and data bus and port E provides bus control and status signals. In special test mode, the write protection of many control bits is lifted so that they can be thoroughly tested without needing to go through reset.
- Special peripheral mode

  This mode is intended for Freescale Semiconductor factory testing of the system. The CPU is inactive and an external (tester) bus master drives address, data, and bus control signals.

4.2 External Signal Description

In typical implementations, the MEBI sub-block of the core interfaces directly with external system pins. Some pins may not be bonded out in all implementations.

Table 4-1 outlines the pin names and functions and gives a brief description of their operation reset state of these pins and associated pull-ups or pull-downs is dependent on the mode of operation and on the integration of this block at the chip level (chip dependent).

Table 7-12. DBGCC Field Descriptions

Field	Description
15:0	**Comparator C Compare Bits** — The comparator C compare bits control whether comparator C will compare the address bus bits [15:0] to a logic 1 or logic 0. See Table 7-13. 0 Compare corresponding address bit to a logic 0 1 Compare corresponding address bit to a logic 1 **Note:** This register will be cleared automatically when the DBG module is armed in LOOP1 mode.

Table 7-13. Comparator C Compares

PAGSEL	EXTCMP Compare	High-Byte Compare
x0	No compare	DBGCCH[7:0] = AB[15:8]
x1	EXTCMP[5:0] = XAB[21:16]	DBGCCH[7:0] = XAB[15:14],AB[13:8]

7.3.2.7 Debug Control Register 2 (DBGC2)

Module Base + 0x0028
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R W	BKABEN(1)	FULL	BDM	TAGAB	BKCEN(2)	TAGC2	RWCEN2	RWC2
Reset	0	0	0	0	0	0	0	0

1. When BKABEN is set (BKP mode), all bits in DBGC2 are available. When BKABEN is cleared and DBG is used in DBG mode, bits FULL and TAGAB have no meaning.
2. These bits can be used in BKP mode and DBG mode (when capture mode is not set in LOOP1) to provide a third breakpoint.

Figure 7-13. Debug Control Register 2 (DBGC2)

Table 7-14. DBGC2 Field Descriptions

Field	Description
7 BKABEN	**Breakpoint Using Comparator A and B Enable** — This bit enables the breakpoint capability using comparator A and B, when set (BKP mode) the DBGEN bit in DBGC1 cannot be set. 0 Breakpoint module off 1 Breakpoint module on
6 FULL	**Full Breakpoint Mode Enable** — This bit controls whether the breakpoint module is in dual mode or full mode. In full mode, comparator A is used to match address and comparator B is used to match data. See Section 7.4.1.2, “Full Breakpoint Mode,” for more details. 0 Dual address mode enabled 1 Full breakpoint mode enabled
5 BDM	**Background Debug Mode Enable** — This bit determines if the breakpoint causes the system to enter background debug mode (BDM) or initiate a software interrupt (SWI). 0 Go to software interrupt on a break request 1 Go to BDM on a break request

7.4.2.6.3 Detail Mode

In the detail mode, address and data for all cycles except program fetch (P) and free (f) cycles are stored in trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where his code was in error.

7.4.2.6.4 Profile Mode

This mode is intended to allow a host computer to poll a running target and provide a histogram of program execution. Each read of the trace buffer address will return the address of the last instruction executed. The DBGCNT register is not incremented and the trace buffer does not get filled. The ARM bit is not used and all breakpoints and all other debug functions will be disabled.

7.4.2.7 Storage Memory

The storage memory is a 64 words deep by 16-bits wide dual port RAM array. The CPU accesses the RAM array through a single memory location window (DBGTBH:DBGTBL). The DBG module stores trace information in the RAM array in a circular buffer format. As data is read via the CPU, a pointer into the RAM will increment so that the next CPU read will receive fresh information. In all trigger modes except for event-only and detail capture mode, the data stored in the trace buffer will be change-of-flow addresses. change-of-flow addresses are defined as follows:

- Source address of conditional branches (long, short, BRSET, and loop constructs) taken
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions
- Vector address of interrupts except for SWI and BDM vectors

In the event-only trigger modes only the 16-bit data bus value corresponding to the event is stored. In the detail capture mode, address and then data are stored for all cycles except program fetch (P) and free (f) cycles.

7.4.2.8 Storing Data in Memory Storage Buffer

7.4.2.8.1 Storing with Begin-Trigger

Storing with begin-trigger can be used in all trigger modes. When DBG mode is enabled and armed in the begin-trigger mode, data is not stored in the trace buffer until the trigger condition is met. As soon as the trigger condition is met, the DBG module will remain armed until 64 words are stored in the trace buffer. If the trigger is at the address of the change-of-flow instruction the change-of-flow associated with the trigger event will be stored in the trace buffer.

7.4.2.8.2 Storing with End-Trigger

Storing with end-trigger cannot be used in event-only trigger modes. When DBG mode is enabled and armed in the end-trigger mode, data is stored in the trace buffer until the trigger condition is met. When the trigger condition is met, the DBG module will become de-armed and no more data will be stored. If

8.3.2.13.1 Left Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R W	BIT 9 MSB BIT 7 MSB	BIT 8 BIT 6	BIT 7 BIT 5	BIT 6 BIT 4	BIT 5 BIT 3	BIT 4 BIT 2	BIT 3 BIT 1	BIT 2 BIT 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-15. Left Justified, ATD Conversion Result Register, High Byte (ATDDRxH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R W	BIT 1 U	BIT 0 U	0 0	0 0	0 0	0 0	0 0	0 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-16. Left Justified, ATD Conversion Result Register, Low Byte (ATDDRxL)

8.3.2.13.2 Right Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R W	0 0	0 0	0 0	0 0	0 0	0 0	BIT 9 MSB 0	BIT 8 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-17. Right Justified, ATD Conversion Result Register, High Byte (ATDDRxH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R W	BIT 7 BIT 7 MSB	BIT 6 BIT 6	BIT 5 BIT 5	BIT 4 BIT 4	BIT 3 BIT 3	BIT 2 BIT 2	BIT 1 BIT 1	BIT 0 BIT 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-18. Right Justified, ATD Conversion Result Register, Low Byte (ATDDRxL)

Table 9-12. Outcome of Clock Loss in Pseudo-Stop Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately
1	1	0	Clock Monitor failure --> Scenario 1: OSCCLK **recovers** prior to exiting Pseudo-Stop Mode. – MCU remains in Pseudo-Stop Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* – CM no longer indicates a failure, – 4096 OSCCLK cycles later Clock Quality Check indicates clock o.k., – SCM deactivated, – PLL disabled, – VREG disabled. – MCU remains in Pseudo-Stop Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit Pseudo-Stop Mode using OSCCLK as system clock (SYSCLK), – Continue normal operation. *or an External Reset is applied.* – Exit Pseudo-Stop Mode using OSCCLK as system clock, – Start reset sequence. Scenario 2: OSCCLK **does not recover** prior to exiting Pseudo-Stop Mode. – MCU remains in Pseudo-Stop Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag, – Keep performing Clock Quality Checks (could continue infinitely) while in Pseudo-Stop Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock – Continue to perform additional Clock Quality Checks until OSCCLK is o.k. again. or an External RESET is applied. – Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock – Start reset sequence, – Continue to perform additional Clock Quality Checks until OSCCLK is o.k.again.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	CANE	CLKSRC	LOOPB	LISTEN		WUPM	SLPAK	INITAK
W								
Reset:	0	0	0	1	0	0	0	1

= Unimplemented

Figure 10-5. MSCAN Control Register 1 (CANCTL1)

Read: Anytime
Write: Anytime when INITRQ = 1 and INITAK = 1, except CANE which is write once in normal and anytime in special system operation modes when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1).

Table 10-2. CANCTL1 Register Field Descriptions

Field	Description
7 CANE	**MSCAN Enable** 0 MSCAN module is disabled 1 MSCAN module is enabled
6 CLKSRC	**MSCAN Clock Source** — This bit defines the clock source for the MSCAN module (only for systems with a clock generation module; Section 10.4.3.2, "Clock System," and Section Figure 10-42., "MSCAN Clocking Scheme,"). 0 MSCAN clock source is the oscillator clock 1 MSCAN clock source is the bus clock
5 LOOPB	**Loopback Self Test Mode** — When this bit is set, the MSCAN performs an internal loopback which can be used for self test operation. The bit stream output of the transmitter is fed back to the receiver internally. The RXCAN input pin is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 0 Loopback self test disabled 1 Loopback self test enabled
4 LISTEN	**Listen Only Mode** — This bit configures the MSCAN as a CAN bus monitor. When LISTEN is set, all valid CAN messages with matching ID are received, but no acknowledgement or error frames are sent out (see Section 10.4.4.4, "Listen-Only Mode"). In addition, the error counters are frozen. Listen only mode supports applications which require "hot plugging" or throughput analysis. The MSCAN is unable to transmit any messages when listen only mode is active. 0 Normal operation 1 Listen only mode activated
2 WUPM	**Wake-Up Mode** — If WUPE in CANCTL0 is enabled, this bit defines whether the integrated low-pass filter is applied to protect the MSCAN from spurious wake-up (see Section 10.4.5.4, "MSCAN Sleep Mode"). 0 MSCAN wakes up on any dominant level on the CAN bus 1 MSCAN wakes up only in case of a dominant pulse on the CAN bus that has a length of T_{wup}

10.3.2.10 MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)

The CANTAAK register indicates the successful abort of a queued message, if requested by the appropriate bits in the CANTARQ register.

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-13. MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)

NOTE

The CANTAAK register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1).

Read: Anytime
Write: Unimplemented for ABTAKx flags

Table 10-14. CANTAAK Register Field Descriptions

Field	Description
2:0 ABTAK[2:0]	**Abort Acknowledge** — This flag acknowledges that a message was aborted due to a pending abort request from the CPU. After a particular message buffer is flagged empty, this flag can be used by the application software to identify whether the message was aborted successfully or was sent anyway. The ABTAKx flag is cleared whenever the corresponding TXE flag is cleared. 0 The message was not aborted. 1 The message was aborted.

"MSCAN Receiver Flag Register (CANRFLG)" and Section 10.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)").

10.4.7.6 Interrupt Acknowledge

Interrupts are directly associated with one or more status flags in either the Section 10.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)" or the Section 10.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)." Interrupts are pending as long as one of the corresponding flags is set. The flags in CANRFLG and CANTFLG must be reset within the interrupt handler to handshake the interrupt. The flags are reset by writing a 1 to the corresponding bit position. A flag cannot be cleared if the respective condition prevails.

NOTE

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

10.4.7.7 Recovery from Stop or Wait

The MSCAN can recover from stop or wait via the wake-up interrupt. This interrupt can only occur if the MSCAN was in sleep mode (SLPRQ = 1 and SLPAK = 1) before entering power down mode, the wake-up option is enabled (WUPE = 1), and the wake-up interrupt is enabled (WUPIE = 1).

10.5 Initialization/Application Information

10.5.1 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 and enter normal mode

If the configuration of registers which are writable in initialization mode needs to be changed only when the MSCAN module is in normal mode:

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 in normal mode

12.4 Functional Description

12.4.1 PWM Clock Select

There are four available clocks called clock A, clock B, clock SA (scaled A), and clock SB (scaled B). These four clocks are based on the bus clock**.**

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the bus clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8, ..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of two clocks, either the pre-scaled clock (clock A or B) or the scaled clock (clock SA or SB).

The block diagram in Figure 12-34 shows the four different clocks and how the scaled clocks are created.

12.4.1.1 Prescale

The input clock to the PWM prescaler is the bus clock. It can be disabled whenever the part is in freeze mode by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This is useful for emulation in order to freeze the PWM. The input clock can also be disabled when all six PWM channels are disabled (PWME5–PWME0 = 0) This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the bus clock. The value selected for clock A is determined by the PCKA2, PCKA1, and PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, and PCKB0 bits also in the PWMPRCLK register.

Figure 13-18 shows a burst of noise near the beginning of the start bit that resets the RT clock. The sample after the reset is low but is not preceded by three high samples that would qualify as a falling edge. Depending on the timing of the start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

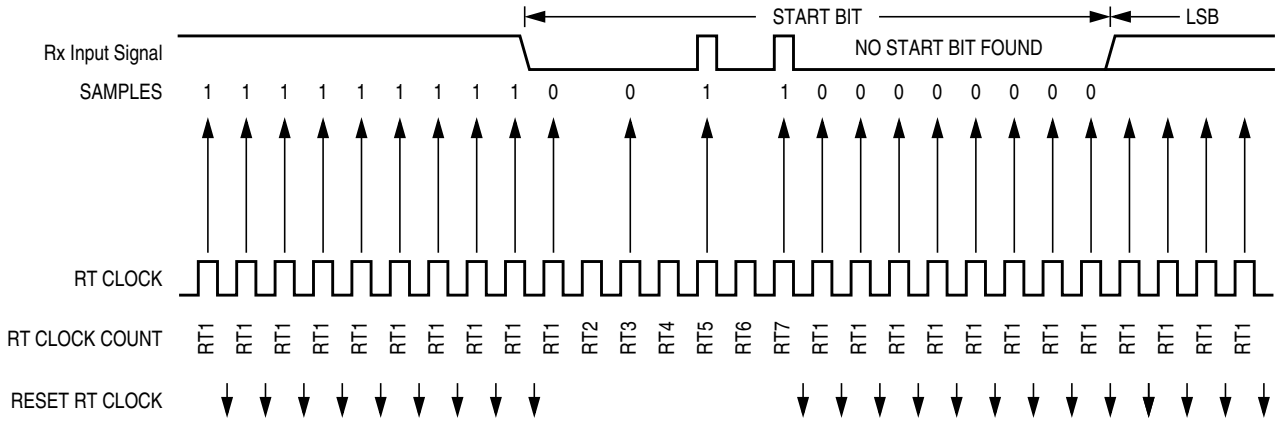


Figure 13-18. Start Bit Search Example 5

In Figure 13-19, a noise burst makes the majority of data samples RT8, RT9, and RT10 high. This sets the noise flag but does not reset the RT clock. In start bits only, the RT8, RT9, and RT10 data samples are ignored.

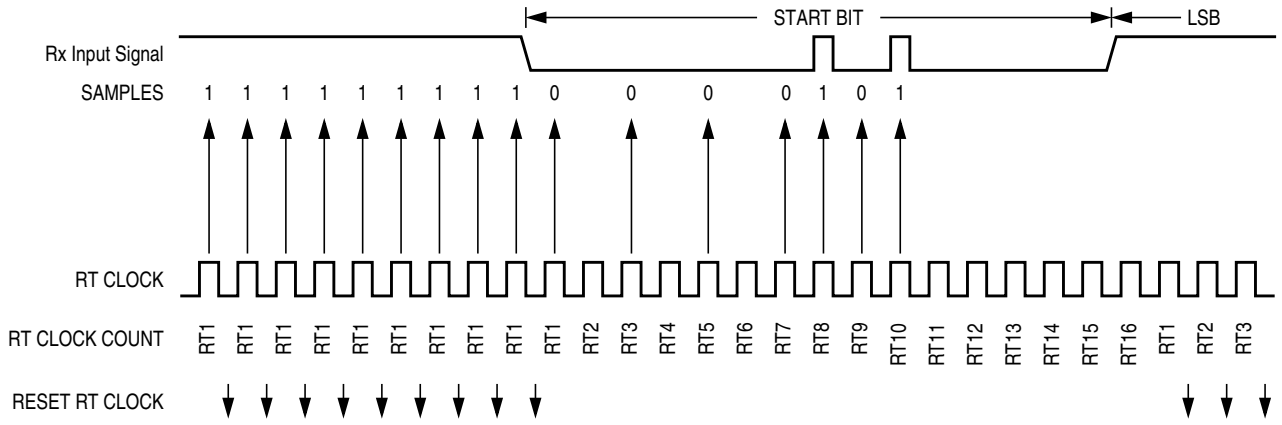


Figure 13-19. Start Bit Search Example 6

14.3.2.2 SPI Control Register 2 (SPICR2)

Module Base 0x0001

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-4. SPI Control Register 2 (SPICR2)

Read: anytime

Write: anytime; writes to the reserved bits have no effect

Table 14-4. SPICR2 Field Descriptions

Field	Description
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure being 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 14-3. 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 14-5. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state

Table 14-5. 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

Table 15-15. TSCR2 Field Descriptions

Field	Description
7 TOI	**Timer Overflow Interrupt Enable** 0 Interrupt inhibited. 1 Hardware interrupt requested when TOF flag set.
3 TCRE	**Timer Counter Reset Enable** — This bit allows the timer counter to be reset by a successful output compare 7 event. This mode of operation is similar to an up-counting modulus counter. 0 Counter reset inhibited and counter free runs. 1 Counter reset by a successful output compare 7. **Note:** If TC7 = 0x0000 and TCRE = 1, TCNT will stay at 0x0000 continuously. If TC7 = 0xFFFF and TCRE = 1, TOF will never be set when TCNT is reset from 0xFFFF to 0x0000. **Note:** TCRE=1 and TC7!=0, the TCNT cycle period will be TC7 x "prescaler counter width" + "1 Bus Clock", for a more detail explanation please refer to Section 15.4.3, “Output Compare
2 PR[2:0]	**Timer Prescaler Select** — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 15-16.

Table 15-16. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	Bus Clock / 1
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

NOTE

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

15.3.2.12 Main Timer Interrupt Flag 1 (TFLG1)

Module Base + 0x000E

	7	6	5	4	3	2	1	0
R W	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
Reset	0	0	0	0	0	0	0	0

Figure 15-20. Main Timer Interrupt Flag 1 (TFLG1)

Read: Anytime

Module Base + 0x0009

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

Figure 17-14. Flash Address Low Register (FADDRLO)

In normal modes, all FABHI and FABLO bits read 0 and are not writable. In special modes, the FABHI and FABLO bits are readable and writable. For sector erase, the MCU address bits [8:0] are ignored. For mass erase, any address within the Flash array is valid to start the command.

17.3.2.10 Flash Data Register (FDATA)

FDATAHI and FDATALO are the Flash data registers.

Module Base + 0x000A

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

Figure 17-15. Flash Data High Register (FDATAHI)

Module Base + 0x000B

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

Figure 17-16. Flash Data Low Register (FDATALO)

In normal modes, all FDATAHI and FDATALO bits read 0 and are not writable. In special modes, all FDATAHI and FDATALO bits are readable and writable when writing to an address within the Flash address range.

17.3.2.11 RESERVED3

This register is reserved for factory testing and is not accessible to the user.

Table 18-9. Flash Protection Function

FPOPEN	FPHDIS	FPHS[1]	FPHS[0]	FPLDIS	FPLS[1]	FPLS[0]	Function(1)
1	1	x	x	1	x	x	No protection
1	1	x	x	0	x	x	Protect low range
1	0	x	x	1	x	x	Protect high range
1	0	x	x	0	x	x	Protect high and low ranges
0	1	x	x	1	x	x	Full Flash array protected
0	0	x	x	1	x	x	Unprotected high range
0	1	x	x	0	x	x	Unprotected low range
0	0	x	x	0	x	x	Unprotected high and low ranges

1. For range sizes refer to Table 18-10 and or Table 18-11.

Table 18-10. Flash Protection Higher Address Range

FPHS[1:0]	Address Range	Range Size
00	0xF800–0xFFFF	2 Kbytes
01	0xF000–0xFFFF	4 Kbytes
10	0xE000–0xFFFF	8 Kbytes
11	0xC000–0xFFFF	16 Kbytes

Table 18-11. Flash Protection Lower Address Range

FPLS[1:0]	Address Range	Range Size
00	0x4000–0x41FF	512 bytes
01	0x4000–0x43FF	1 Kbyte
10	0x4000–0x47FF	2 Kbytes
11	0x4000–0x4FFF	4 Kbytes

Figure 18-9 illustrates all possible protection scenarios. Although the protection scheme is loaded from the Flash array after reset, it is allowed to change in normal modes. This protection scheme can be used by applications requiring re-programming in single chip mode while providing as much protection as possible if no re-programming is required.

Table 20-5. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of the backdoor key access to the Flash module as shown in Table 20-6.
5–2 NV[5:2]	**Nonvolatile Flag Bits** — The NV[5:2] bits are available to the user as nonvolatile flags.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 20-7. If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

Table 20-6. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01(1)	DISABLED
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable Backdoor Key Access.

Table 20-7. Flash Security States

SEC[1:0]	Status of Security
00	Secured
01(1)	Secured
10	Unsecured
11	Secured

1. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 20.4.3, “Flash Module Security”.

20.3.2.3 RESERVED1

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 20-8. RESERVED1

All bits read 0 and are not writable.

20.4.1.3 Valid Flash Commands

Table 20-17 summarizes the valid Flash commands along with the effects of the commands on the Flash array.

Table 20-17. Valid Flash Commands

FCMD	Meaning	Function on Flash Array
0x05	Erase Verify	Verify all bytes in the Flash array are erased. If the Flash array is erased, the BLANK bit will set in the FSTAT register upon command completion.
0x20	Program	Program a word (2 bytes) in the Flash array.
0x40	Sector Erase	Erase all 1024 bytes in a sector of the Flash array.
0x41	Mass Erase	Erase all bytes in the Flash array. A mass erase of the full Flash array is only possible when FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register are set prior to launching the command.

CAUTION

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

addresses sequentially staring with 0xFF00-0xFF01 and ending with 0xFF06–0xFF07. The values 0x0000 and 0xFFFF are not permitted as keys. When the KEYACC bit is set, reads of the Flash array will return invalid data.

The user code stored in the Flash array must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If KEYEN[1:0] = 1:0 in the FSEC register, the MCU can be unsecured by the backdoor key access sequence described below:

1. Set the KEYACC bit in the FCNFG register
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00
3. Clear the KEYACC bit in the FCNFG register
4. If all four 16-bit words match the backdoor key stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and bits SEC[1:0] in the FSEC register are forced to the unsecure state of 1:0

The backdoor key access sequence is monitored by the internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following illegal operations will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor key programmed in the Flash array
2. If the four 16-bit words are written in the wrong sequence
3. If more than four 16-bit words are written
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF
5. If the KEYACC bit does not remain set while the four 16-bit words are written

After the backdoor key access sequence has been correctly matched, the MCU will be unsecured. The Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the four word backdoor key by programming bytes 0xFF00–0xFF07 of the Flash configuration field.

The security as defined in the Flash security/options byte at address 0xFF0F is not changed by using the backdoor key access sequence to unsecure. The backdoor key stored in addresses 0xFF00–0xFF07 is unaffected by the backdoor key access sequence. After the next reset sequence, the security state of the Flash module is determined by the Flash security/options byte at address 0xFF0F. The backdoor key access sequence has no effect on the program and erase protection defined in the FPROT register.

It is not possible to unsecure the MCU in special single chip mode by executing the backdoor key access sequence in background debug mode.

Table A-17. PLL Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Self Clock Mode frequency	f_{SCM}	1	—	5.5	MHz
2	D	VCO locking range	f_{VCO}	8	—	50	MHz
3	D	Lock Detector transition from Acquisition to Tracking mode	$\|\Delta_{trk}\|$	3	—	4	%(1)
4	D	Lock Detection	$\|\Delta_{Lock}\|$	0	—	1.5	%[1]
5	D	Un-Lock Detection	$\|\Delta_{unl}\|$	0.5	—	2.5	%[1]
6	D	Lock Detector transition from Tracking to Acquisition mode	$\|\Delta_{unt}\|$	6	—	8	%[1]
7	C	PLLON Total Stabilization delay (Auto Mode) (2)	t_{stab}	—	0.5	—	ms
8	D	PLLON $_{Acquisition}$ mode stabilization delay [2]	t_{acq}	—	0.3	—	ms
9	D	PLLON Tracking mode stabilization delay [2]	t_{al}	—	0.2	—	ms
10	D	Fitting parameter VCO loop gain	K_1	—	-100	—	MHz/V
11	D	Fitting parameter VCO loop frequency	f_1	—	60	—	MHz
12	D	Charge pump current acquisition mode	$\| i_{ch} \|$	—	38.5	—	µA
13	D	Charge pump current tracking mode	$\| i_{ch} \|$	—	3.5	—	µA
14	C	Jitter fit parameter 1[2]	j_1	—	—	1.1	%
15	C	Jitter fit parameter 2[2]	j_2	—	—	0.13	%

1. % deviation from target frequency
2. f_{OSC} = 4MHz, f_{BUS} = 25MHz equivalent f_{VCO} = 50MHz: REFDV = #$03, SYNR = #$018, Cs = 4.7nF, Cp = 470pF, Rs = 10KΩ.

A.5 NVM, Flash, and EEPROM

A.5.1 NVM Timing

The time base for all NVM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The NVM modules do not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the NVM modules at a lower frequency a full program or erase transition is not assured.

The Flash program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV and ECLKDIV registers respectively. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in Table A-18 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2MHz.