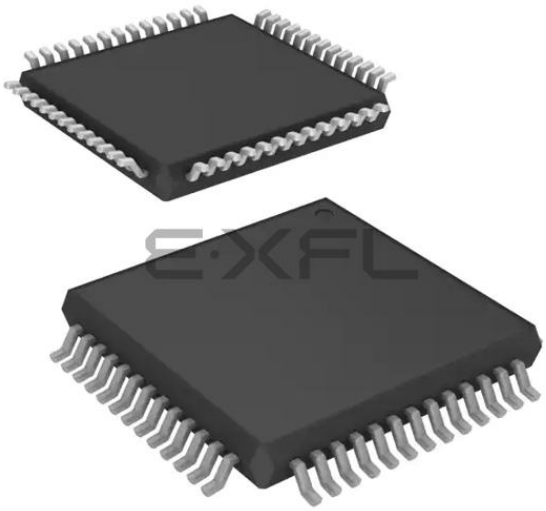


Welcome to E-XFL.COM

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	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	52-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12c64cpbe

2.3.2.4.5 Port P Pull Device Enable Register (PERP)

Module Base + 0x001C

	7	6	5	4	3	2	1	0
R W	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
Reset	0	0	0	0	0	0	0	0

Figure 2-28. Port P Pull Device Enable Register (PERP)

Read: Anytime.

Write: Anytime.

Table 2-24. PERP Field Descriptions

Field	Description
7–0 PERP[7:0]	**Pull Device Enable Port P** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.4.6 Port P Polarity Select Register (PPSP)

Module Base + 0x001D

	7	6	5	4	3	2	1	0
R W	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
Reset	0	0	0	0	0	0	0	0

Figure 2-29. Port P Polarity Select Register (PPSP)

Read: Anytime.

Write: Anytime.

Table 2-25. PPSP Field Descriptions

Field	Description
7–0 PPSP[7:0]	**Pull Select Port P** — This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 Falling edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-up device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input. 1 Rising edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-down device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

2.3.2.6.5 Port AD Pull Device Enable Register (PERAD)

Module Base + 0x0034

	7	6	5	4	3	2	1	0
R W	PERAD7	PERAD6	PERAD5	PERAD4	PERAD3	PERAD2	PERAD1	PERAD0
Reset	0	0	0	0	0	0	0	0

Figure 2-44. Port AD Pull Device Enable Register (PERAD)

Read: Anytime.

Write: Anytime.

Table 2-36. PERAD Field Descriptions

Field	Description
7–0 PERAD[7:0]	**Pull Device Enable Port AD** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled. It is not possible to enable pull devices when a associated ATD channel is enabled simultaneously. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.6.6 Port AD Polarity Select Register (PPSAD)

Module Base + 0x0035

	7	6	5	4	3	2	1	0
R W	PPSAD7	PPSAD6	PPSAD5	PPSAD4	PPSAD3	PPSAD2	PPSAD1	PPSAD0
Reset	0	0	0	0	0	0	0	0

Figure 2-45. Port AD Polarity Select Register (PPSAD)

Read: Anytime.

Write: Anytime.

Table 2-37. PPSAD Field Descriptions

Field	Description
7–0 PPSAD[7:0]	**Pull Select Port AD** — This register selects whether a pull-down or a pull-up device is connected to the pin. 0 A pull-up device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input. 1 A pull-down device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input.

2.4.2.5 Port P

The PWM module is connected to port P. Port P pins can be used as PWM outputs. Further the Keypad Wake-Up function is implemented on pins PP[7:0]. During reset, port P pins are configured as high-impedance inputs.

Port P offers 8 general purpose I/O pins with edge triggered interrupt capability in wired-or fashion. The interrupt enable as well as the sensitivity to rising or falling edges can be individually configured on per pin basis. All 8 bits/pins share the same interrupt vector. Interrupts can be used with the pins configured as inputs or outputs.

An interrupt is generated when a bit in the port interrupt flag register and its corresponding port interrupt enable bit are both set. This external interrupt feature is capable to wake up the CPU when it is in STOP or WAIT mode.

A digital filter on each pin prevents pulses (Figure 2-48) shorter than a specified time from generating an interrupt. The minimum time varies over process conditions, temperature and voltage (Figure 2-47 and Table 2-38).

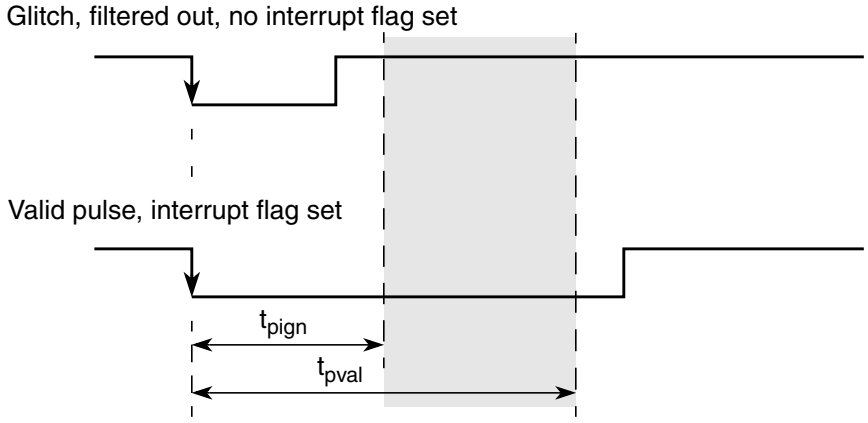


Figure 2-47. Interrupt Glitch Filter on Port P and J (PPS = 0)

Table 2-38. Pulse Detection Criteria

Pulse	$\overline{\text{STOP}}$ Mode		STOP(1) Mode	
	Value	Unit	Value	Unit
Ignored	t_{pign} <= 3	Bus clocks	t_{pign} <= 3.2	µs
Uncertain	3 < t_{pulse} < 4	Bus clocks	3.2 < t_{pulse} < 10	µs
Valid	t_{pval} >= 4	Bus clocks	t_{pval} >= 10	µs

1. These values include the spread of the oscillator frequency over temperature, voltage and process.

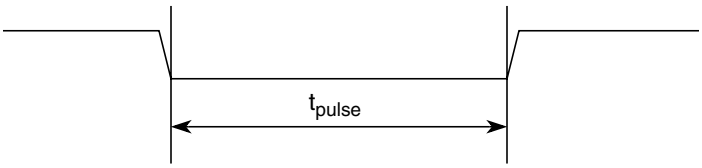


Figure 2-48. Pulse Illustration

4.3.2.13 Reserved Register

Module Base + 0x000F
Starting address location affected by INITRG register setting.

	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 4-17. Reserved Register

This register location is not used (reserved). All bits in this register return logic 0s when read. Writes to this register have no effect.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

4.3.2.14 IRQ Control Register (IRQCR)

Module Base + 0x001E
Starting address location affected by INITRG register setting.

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

= Unimplemented or Reserved

Figure 4-18. IRQ Control Register (IRQCR)

Read: See individual bit descriptions below

Write: See individual bit descriptions below

Table 4-12. IRQCR Field Descriptions

Field	Description
7 IRQE	**IRQ Select Edge Sensitive Only** Special modes: read or write anytime Normal and Emulation modes: read anytime, write once 0 IRQ configured for low level recognition. 1 IRQ configured to respond only to falling edges. Falling edges on the IRQ pin will be detected anytime IRQE = 1 and will be cleared only upon a reset or the servicing of the IRQ interrupt.
6 IRQEN	**External IRQ Enable** Normal, emulation, and special modes: read or write anytime 0 External IRQ pin is disconnected from interrupt logic. 1 External IRQ pin is connected to interrupt logic. **Note:** When IRQEN = 0, the edge detect latch is disabled.

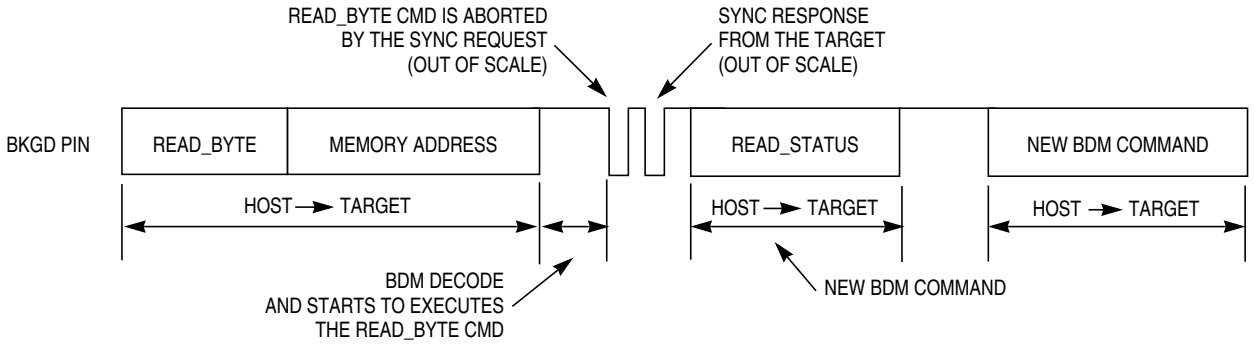


Figure 6-12. ACK Abort Procedure at the Command Level

Figure 6-13 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode. Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Because this is not a probable situation, the protocol does not prevent this conflict from happening.

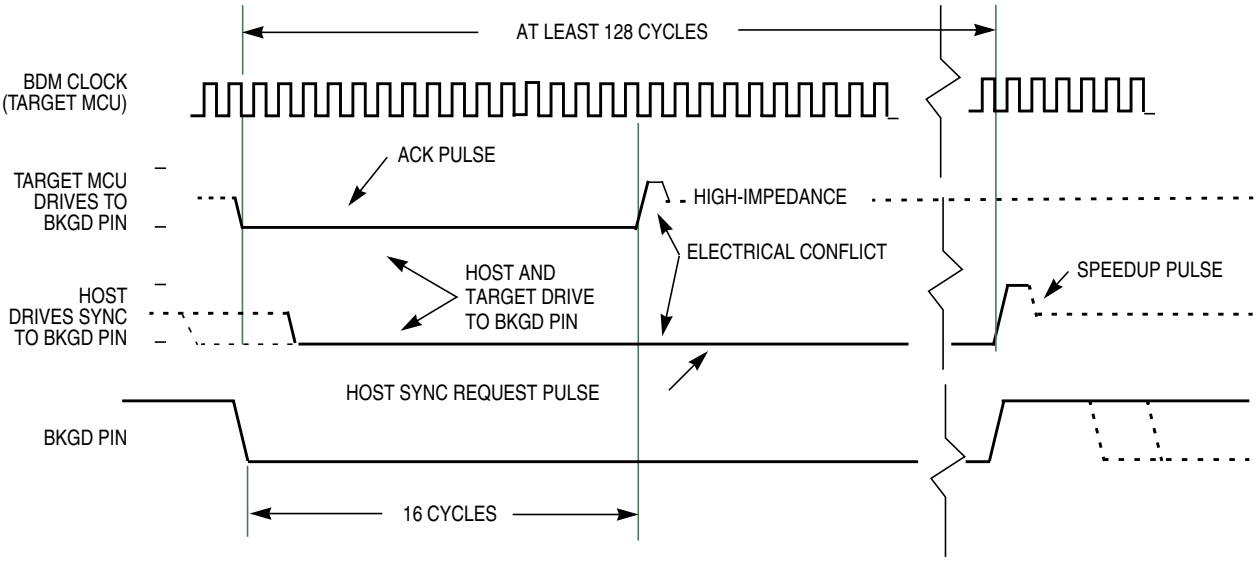


Figure 6-13. ACK Pulse and SYNC Request Conflict

NOTE

This information is being provided so that the MCU integrator will be aware that such a conflict could eventually occur.

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices which are not able to execute the hardware handshake protocol. It also allows for new POD devices, that support the hardware handshake protocol, to freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

8.3.2.10 ATD Status Register 1 (ATDSTAT1)

This read-only register contains the Conversion Complete Flags.

Module Base + 0x000B

	7	6	5	4	3	2	1	0
R	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-12. ATD Status Register 1 (ATDSTAT1)

Read: Anytime

Write: Anytime, no effect

Table 8-16. ATDSTAT1 Field Descriptions

Field	Description
7–0 CCF[7:0]	**Conversion Complete Flag x (x = 7, 6, 5, 4, 3, 2, 1, 0)** — A conversion complete flag is set at the end of each conversion in a conversion sequence. The flags are associated with the conversion position in a sequence (and also the result register number). Therefore, CCF0 is set when the first conversion in a sequence is complete and the result is available in result register ATDDR0; CCF1 is set when the second conversion in a sequence is complete and the result is available in ATDDR1, and so forth. A flag CCFx (x = 7, 6, 5, 4, 3, 2, 1, 0) is cleared when one of the following occurs: A) Write to ATDCTL5 (a new conversion sequence is started) B) If AFFC = 0 and read of ATDSTAT1 followed by read of result register ATDDRx C) If AFFC = 1 and read of result register ATDDRx 0 Conversion number x not completed 1 Conversion number x has completed, result ready in ATDDRx

The MSCAN then schedules the message for transmission and signals the successful transmission of the buffer by setting the associated TXE flag. A transmit interrupt (see Section 10.4.7.2, "Transmit Interrupt") is generated[1] when TXEx is set and can be used to drive the application software to re-load the buffer.

If more than one buffer is scheduled for transmission when the CAN bus becomes available for arbitration, the MSCAN uses the local priority setting of the three buffers to determine the prioritization. For this purpose, every transmit buffer has an 8-bit local priority field (PRIO). The application software programs this field when the message is set up. The local priority reflects the priority of this particular message relative to the set of messages being transmitted from this node. The lowest binary value of the PRIO field is defined to be the highest priority. The internal scheduling process takes place whenever the MSCAN arbitrates for the CAN bus. This is also the case after the occurrence of a transmission error.

When a high priority message is scheduled by the application software, it may become necessary to abort a lower priority message in one of the three transmit buffers. Because messages that are already in transmission cannot be aborted, the user must request the abort by setting the corresponding abort request bit (ABTRQ) (see Section 10.3.2.9, "MSCAN Transmitter Message Abort Request Register (CANTARQ)".) The MSCAN then grants the request, if possible, by:

1. Setting the corresponding abort acknowledge flag (ABTAK) in the CANTAAK register.
2. Setting the associated TXE flag to release the buffer.
3. Generating a transmit interrupt. The transmit interrupt handler software can determine from the setting of the ABTAK flag whether the message was aborted (ABTAK = 1) or sent (ABTAK = 0).

10.4.2.3 Receive Structures

The received messages are stored in a five stage input FIFO. The five message buffers are alternately mapped into a single memory area (see Figure 10-38). The background receive buffer (RxBG) is exclusively associated with the MSCAN, but the foreground receive buffer (RxFG) is addressable by the CPU (see Figure 10-38). This scheme simplifies the handler software because only one address area is applicable for the receive process.

All receive buffers have a size of 15 bytes to store the CAN control bits, the identifier (standard or extended), the data contents, and a time stamp, if enabled (see Section 10.3.3, "Programmer's Model of Message Storage").

The receiver full flag (RXF) (see Section 10.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)") signals the status of the foreground receive buffer. When the buffer contains a correctly received message with a matching identifier, this flag is set.

On reception, each message is checked to see whether it passes the filter (see Section 10.4.3, "Identifier Acceptance Filter") and simultaneously is written into the active RxBG. After successful reception of a valid message, the MSCAN shifts the content of RxBG into the receiver FIFO[2], sets the RXF flag, and generates a receive interrupt (see Section 10.4.7.3, "Receive Interrupt") to the CPU[3]. The user's receive handler must read the received message from the RxFG and then reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message, which can follow immediately after the IFS field of the CAN frame, is received into the next available RxBG. If the MSCAN receives an invalid

1. The transmit interrupt occurs only if not masked. A polling scheme can be applied on TXEx also.
2. Only if the RXF flag is not set.
3. The receive interrupt occurs only if not masked. A polling scheme can be applied on RXF also.

  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection

13.1.3 Modes of Operation

The SCI operation is the same independent of device resource mapping and bus interface mode. Different power modes are available to facilitate power saving.

13.1.3.1 Run Mode

Normal mode of operation.

13.1.3.2 Wait Mode

SCI operation in wait mode depends on the state of the SCISWAI bit in the SCI control register 1 (SCICR1).

- If SCISWAI is clear, the SCI operates normally when the CPU is in wait mode.
- If SCISWAI is set, SCI clock generation ceases and the SCI module enters a power-conservation state when the CPU is in wait mode. Setting SCISWAI does not affect the state of the receiver enable bit, RE, or the transmitter enable bit, TE.
- If SCISWAI is set, any transmission or reception in progress stops at wait mode entry. The transmission or reception resumes when either an internal or external interrupt brings the CPU out of wait mode. Exiting wait mode by reset aborts any transmission or reception in progress and resets the SCI.

13.1.3.3 Stop Mode

The SCI is inactive during stop mode for reduced power consumption. The STOP instruction does not affect the SCI register states, but the SCI module clock will be disabled. The SCI operation resumes from where it left off after an external interrupt brings the CPU out of stop mode. Exiting stop mode by reset aborts any transmission or reception in progress and resets the SCI.

If software clears TE while a transmission is in progress (TC = 0), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first byte of the second message to SCIDRH/L.

13.4.3.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCICR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCICR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

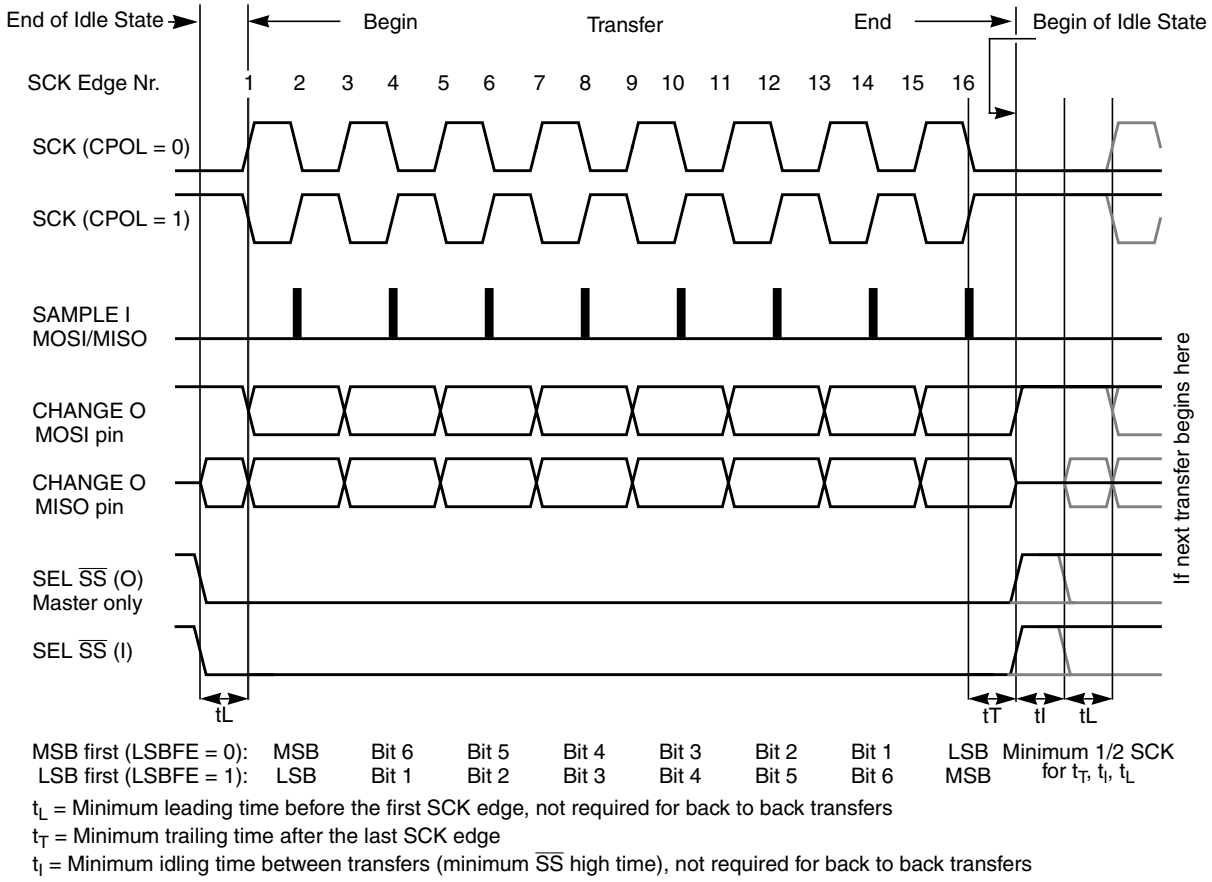
The SCI recognizes a break character when a start bit is followed by eight or nine logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on SCI registers:

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see Section 13.3.2.4, “SCI Status Register 1 (SCISR1)” and Section 13.3.2.5, “SCI Status Register 2 (SCISR2)”

13.4.3.4 Idle Characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCI control register 1 (SCICR1). The preamble is a synchronizing idle character that begins the first transmission initiated after writing the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the **Tx output** signal becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the frame currently being transmitted.



t_L = Minimum leading time before the first SCK edge, not required for back to back transfers
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back to back transfers

Figure 14-10. SPI Clock Format 1 (CPHA = 1)

14.4.4 SPI Baud Rate Generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI Baud Rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in Figure 14-11

When all bits are clear (the default condition), the SPI module clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the module clock divisor becomes 4. When the selection bits are 010, the module clock divisor becomes 8 etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 14-7 for baud rate calculations for all bit conditions, based on a 25-MHz bus clock. The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

15.3.2.14 Timer Input Capture/Output Compare Registers High and Low 0–7 (TCxH and TCxL)

Module Base + 0x0010 = TC0H, 0x0012 = TC1H, 0x0014 = TC2H, 0x0016 = TC3H, 0x0018 = TC4H, 0x001A = TC5H, 0x001C = TC6H, 0x001E = TC7H

	15	14	13	12	11	10	9	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 15-22. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L, 0x0013 = TC1L, 0x0015 = TC2L, 0x0017 = TC3L, 0x0019 = TC4L, 0x001B = TC5L, 0x001D = TC6L, 0x001F = TC7L

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

Figure 15-23. Timer Input Capture/Output Compare Register x Low (TCxL)

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should takes place before low byte otherwise it will give a different result.

15.6.2 Pulse Accumulator Input Interrupt (PAOVI)

This active high output will be asserted by the module to request a timer pulse accumulator input interrupt to be serviced by the system controller.

15.6.3 Pulse Accumulator Overflow Interrupt (PAOVF)

This active high output will be asserted by the module to request a timer pulse accumulator overflow interrupt to be serviced by the system controller.

15.6.4 Timer Overflow Interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt to be serviced by the system controller.

17.4.1.3.4 Mass Erase Command

The mass erase operation will erase all addresses in a Flash array using an embedded algorithm.

An example flow to execute the mass erase operation is shown in Figure 17-25. The mass erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the mass erase command. The address and data written will be ignored.
2. Write the mass erase command, 0x41, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the mass erase command.

If a Flash array to be erased contains any protected area, the PVIOL flag in the FSTAT register will set and the mass erase command will not launch. Once the mass erase command has successfully launched, the CCIF flag in the FSTAT register will set after the mass erase operation has completed unless a new command write sequence has been buffered.

18.4.1.3.3 Sector Erase Command

The sector erase operation will erase all addresses in a 512 byte sector of the Flash array using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 18-24. The sector erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while MCU address bits [8:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

If a Flash sector to be erased is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. Once the sector erase command has successfully launched, the CCIF flag in the FSTAT register will set after the sector erase operation has completed unless a new command write sequence has been buffered.

20.3.2 Register Descriptions

The Flash module contains a set of 16 control and status registers located between module base + 0x0000 and 0x000F. A summary of the Flash module registers is given in Figure 20-5. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 FCLKDIV	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
	W								
0x0002 RESERVED1 (1)	R	0	0	0	0	0	0	0	0
	W								
0x0003 FCNFG	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
	W								
0x0004 FPROT	R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0005 FSTAT	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
	W								
0x0006 FCMD	R	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
	W								
0x0007 RESERVED2[1]	R	0	0	0	0	0	0	0	0
	W								
0x0008 FADDRHI[1]	R	FABHI							
	W								
0x0009 FADDRLO[1]	R	FABLO							
	W								
0x000A FDATAHI[1]	R	FDHI							
	W								
0x000B FDATALO[1]	R	FDLO							
	W								
0x000C RESERVED3[1]	R	0	0	0	0	0	0	0	0
	W								
0x000D RESERVED4[1]	R	0	0	0	0	0	0	0	0
	W								
0x000E RESERVED5[1]	R	0	0	0	0	0	0	0	0
	W								
0x000F RESERVED6[1]	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 20-5. Flash Register Summary

1. Intended for factory test purposes only.

Table 20-10. 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 20-11 and Table 20-12 or .

Table 20-11. 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 20-12. Flash Protection Lower Address Range

FPLS[1:0]	Address Range	Range Size
00	0x4000–0x43FF	1 Kbyte
01	0x4000–0x47FF	2 Kbytes
10	0x4000–0x4FFF	4 Kbytes
11	0x4000–0x5FFF	8 Kbytes

Figure 20-11 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.

A.2.5 ATD Accuracy (3.3V Range)

Table A-13 specifies the ATD conversion performance excluding any errors due to current injection, input capacitance and source resistance.

Table A-13. ATD Conversion Performance

Conditions are shown in Table A-4 unless otherwise noted
$V_{REF} = V_{RH} - V_{RL}$ = 3.328V. Resulting to one 8 bit count = 13mV and one 10 bit count = 3.25mV
f_{ATDCLK} = 2.0MHz

Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	P	10-Bit Resolution	LSB	—	3.25	—	mV
2	P	10-Bit Differential Nonlinearity	DNL	–1.5	—	1.5	Counts
3	P	10-Bit Integral Nonlinearity	INL	–3.5	±1.5	3.5	Counts
4	P	10-Bit Absolute Error(1)	AE	–5	±2.5	5	Counts
5	P	8-Bit Resolution	LSB	—	13	—	mV
6	P	8-Bit Differential Nonlinearity	DNL	–0.5	—	0.5	Counts
7	P	8-Bit Integral Nonlinearity	INL	–1.5	±1	1.5	Counts
8	P	8-Bit Absolute Error[1]	AE	–2.0	±1.5	2.0	Counts

1. These values include the quantization error which is inherently 1/2 count for any A/D converter.

For the following definitions see also Figure A-1.

Differential Non-Linearity (DNL) is defined as the difference between two adjacent switching steps.

$$DNL(i) = \frac{V_i - V_{i-1}}{1LSB} - 1$$

The Integral Non-Linearity (INL) is defined as the sum of all DNLs:

$$INL(n) = \sum_{i=1}^{n} DNL(i) = \frac{V_n - V_0}{1LSB} - n$$

B.1.1 PK[2:0] / XADDR[16:14]

PK2-PK0 provide the expanded address XADDR[16:14] for the external bus.

Refer to the S12 Core user guide for detailed information about external address page access.

Pin Name Function 1	Pin Name Function 2	Power Domain	Internal Pull Resistor		Description
			CTRL	Reset State	
PK[2:0]	XADDR[16:14]	V_{DDX}	PUPKE	Up	Port K I/O Pins

The reset state of DDRK in the S12_CORE is $00, configuring the pins as inputs.

The reset state of PUPKE in the PUCR register of the S12_CORE is "1" enabling the internal pullup resistors at PortK[2:0].

In this reset state the pull-up resistors provide a defined state and prevent a floating input, thereby preventing unnecessary current consumption at the input stage.

Appendix E
Ordering Information

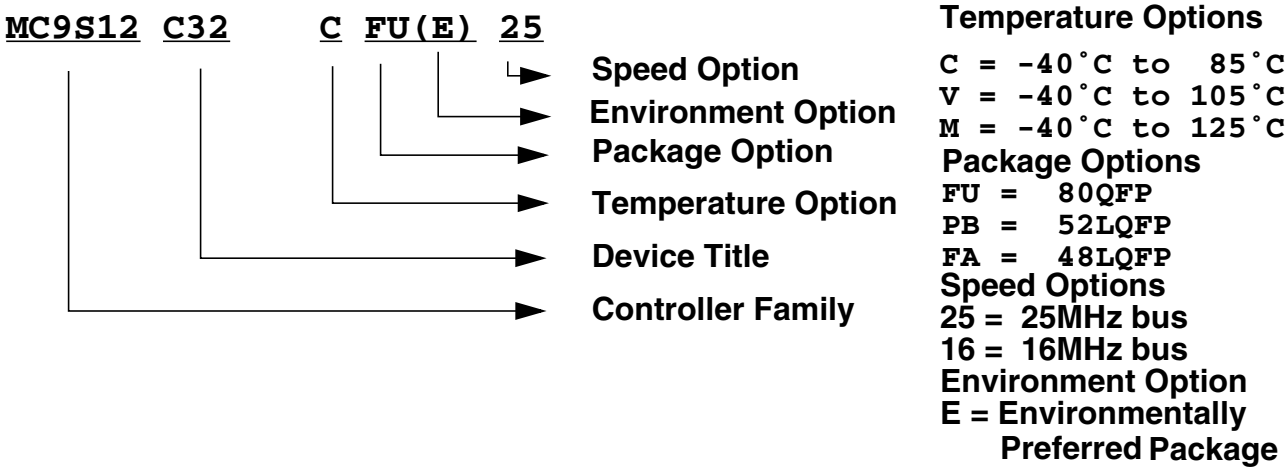


Figure E-1. Order Part number Coding

Table E-1. lists C-family part number coding based on package, speed and temperature and die options.

Table E-2. lists CG-family part number coding based on package, speed and temperature and die options.

Table E-1. MC9S12C-Family / MC9S12GC-Family Part Number Coding

Part Number	Mask[1] set	Temp.	Package	Speed	Die Type	Flash	RAM	I/O[2], [3]
MC9S12C128CFA	XL09S/0M66G	-40˚C, 85˚C	48LQFP	25MHz	C128 die	128K	4K	31
MC9S12C128CPB	XL09S/0M66G	-40˚C, 85˚C	52LQFP	25MHz	C128 die	128K	4K	35
MC9S12C128CFU	XL09S/0M66G	-40˚C, 85˚C	80QFP	25MHz	C128 die	128K	4K	60
MC9S12C128VFA	XL09S/0M66G	-40˚C,105˚C	48LQFP	25MHz	C128 die	128K	4K	31
MC9S12C128VPB	XL09S/0M66G	-40˚C,105˚C	52LQFP	25MHz	C128 die	128K	4K	35
MC9S12C128VFU	XL09S/0M66G	-40˚C, 105˚C	80QFP	25MHz	C128 die	128K	4K	60
MC9S12C128MFA	XL09S/0M66G	-40˚C,125˚C	48LQFP	25MHz	C128 die	128K	4K	31
MC9S12C128MPB	XL09S/0M66G	-40˚C,125˚C	52LQFP	25MHz	C128 die	128K	4K	35
MC9S12C128MFU	XL09S/0M66G	-40˚C, 125˚C	80QFP	25MHz	C128 die	128K	4K	60
MC9S12C96CFA	XL09S/0M66G	-40˚C, 85˚C	48LQFP	25MHz	C128 die	96K	4K	31
MC9S12C96CPB	XL09S/0M66G	-40˚C, 85˚C	52LQFP	25MHz	C128 die	96K	4K	35
MC9S12C96CFU	XL09S/0M66G	-40˚C, 85˚C	80QFP	25MHz	C128 die	96K	4K	60
MC9S12C96VFA	XL09S/0M66G	-40˚C,105˚C	48LQFP	25MHz	C128 die	96K	4K	31
MC9S12C96VPB	XL09S/0M66G	-40˚C,105˚C	52LQFP	25MHz	C128 die	96K	4K	35
MC9S12C96VFU	XL09S/0M66G	-40˚C, 105˚C	80QFP	25MHz	C128 die	96K	4K	60
MC9S12C96MFA	XL09S/0M66G	-40˚C,125˚C	48LQFP	25MHz	C128 die	96K	4K	31
MC9S12C96MPB	XL09S/0M66G	-40˚C,125˚C	52LQFP	25MHz	C128 die	96K	4K	35
MC9S12C96MFU	XL09S/0M66G	-40˚C, 125˚C	80QFP	25MHz	C128 die	96K	4K	60
MC9S12C64CFA	XL09S/0M66G	-40˚C, 85˚C	48LQFP	25MHz	C128 die	64K	4K	31
MC9S12C64CPB	XL09S/0M66G	-40˚C, 85˚C	52LQFP	25MHz	C128 die	64K	4K	35