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"

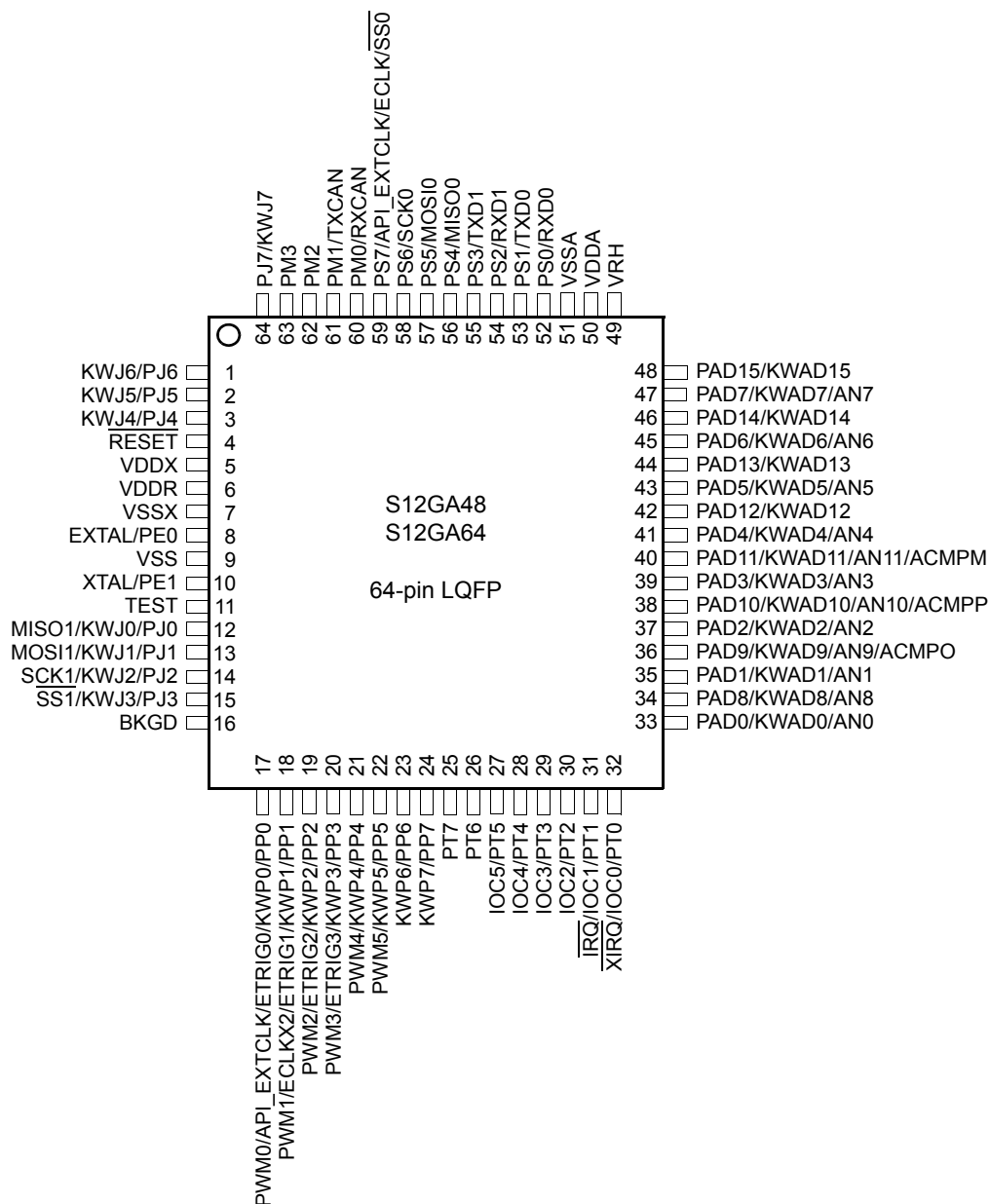| Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	40
Program Memory Size	240KB (240K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	11K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 16x10b
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/s9s12g240f0vlfr

1.8.5.2   Pinout 64-Pin LQFP

Figure 1-14. 64-Pin LQFP Pinout for S12GA48 and S12GA64

Table 2-17. Port AD Pins AD7-0 (continued)

PAD3	• 20 TSSOP: The ACMPO signal of the analog comparator is mapped to this pin when used with the ACMP function. If the ACMP output is enabled (ACMPC[ACOPE]=1) the I/O state will be forced to output. • The ADC analog input channel signal AN3 and the related digital trigger input are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control. • Pin interrupts can be generated if enabled in digital input or output mode. • Signal priority: 20 TSSOP: ACMPO > GPO Others: GPO
PAD2-PAD0	• The ADC analog input channel signals AN2-0 and their related digital trigger inputs are mapped to this pin. The ADC function has no effect on the output state. Refer to NOTE/2-172 for input buffer control. • Pin interrupts can be generated if enabled in digital input or output mode. • Signal priority: GPO

Table 2-31. DDRE Register Field Descriptions

Field	Description
1-0 DDRE	**Port E Data Direction**— This bit determines whether the associated pin is an input or output. 1 Associated pin configured as output 0 Associated pin configured as input

2.4.3.11 Ports A, B, C, D, E, BKGD pin Pull Control Register (PUCR)

Address 0x000C (G1)   Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	BKPUE	0	PDPEE	PUPDE	PUPCE	PUPBE	PUPAE
W		BKPUE		PDPEE	PUPDE	PUPCE	PUPBE	PUPAE
Reset	0	1	0	1	0	0	0	0

Address 0x000C (G2, G3)   Access: User read/write

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

Figure 2-12. Ports A, B, C, D, E, BKGD pin Pullup Control Register (PUCR)

[1] Read:Anytime in normal mode.
Write:Anytime, except BKPUE, which is writable in special mode only.

Table 2-32. PUCR Register Field Descriptions

Field	Description
6 BKPUE	**BKGD pin Pullup Enable**—Enable pullup device on pin This bit configures whether a pullup device is activated, if the pin is used as input. If a pin is used as output this bit has no effect. Out of reset the pullup device is enabled. 1 Pullup device enabled 0 Pullup device disabled
4 PDPEE	**Port E Pulldown Enable**—Enable pulldown devices on all port input pins This bit configures whether a pulldown device is activated on all associated port input pins. If a pin is used as output or used with the CPMU OSC function this bit has no effect. Out of reset the pulldown devices are enabled. 1 Pulldown devices enabled 0 Pulldown devices disabled
3 PUPDE	**Port D Pullup Enable**—Enable pullup devices on all port input pins This bit configures whether a pullup device is activated on all associated port input pins. If a pin is used as output this bit has no effect. 1 Pullup devices enabled 0 Pullup devices disabled

MC9S12G Family Reference Manual Rev.1.27

8.3.2.8.5 Debug Comparator Data High Register (DBGADH)

Address: 0x002C

	7	6	5	4	3	2	1	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 8-19. Debug Comparator Data High Register (DBGADH)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

Table 8-28. DBGADH Field Descriptions

Field	Description
7–0 Bits[15:8]	**Comparator Data High Compare Bits**— The Comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

8.3.2.8.6 Debug Comparator Data Low Register (DBGADL)

Address: 0x002D

	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 8-20. Debug Comparator Data Low Register (DBGADL)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

Table 8-29. DBGADL Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Compare Bits** — The Comparator data low compare bits control whether the selected comparator compares the data bus bits [7:0] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
        LDX       #SUB_1
MARK1   JMP       0,X                          ; IRQ interrupt occurs during execution of this
MARK2   NOP                                    ;

SUB_1   BRN       *                            ; JMP Destination address TRACE BUFFER ENTRY 1
                                               ; RTI Destination address TRACE BUFFER ENTRY 3
        NOP                                    ;
ADDR1   DBNE      A,PART5                      ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR LDAB      #$F0                         ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
        STAB      VAR_C1
        RTI                                    ;
```

The execution flow taking into account the IRQ is as follows

```
        LDX       #SUB_1
MARK1   JMP       0,X                          ;
IRQ_ISR LDAB      #$F0                         ;
        STAB      VAR_C1
        RTI                                    ;
SUB_1   BRN       *
        NOP                                    ;
ADDR1   DBNE      A,PART5                      ;
```

8.4.5.2.2   Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code that the DBG module is designed to help find.

8.4.5.2.3   Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the 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 the code is in error. This mode also features information bit storage to the trace buffer, for each address byte

Figure 10-33. Enabling the External Oscillator
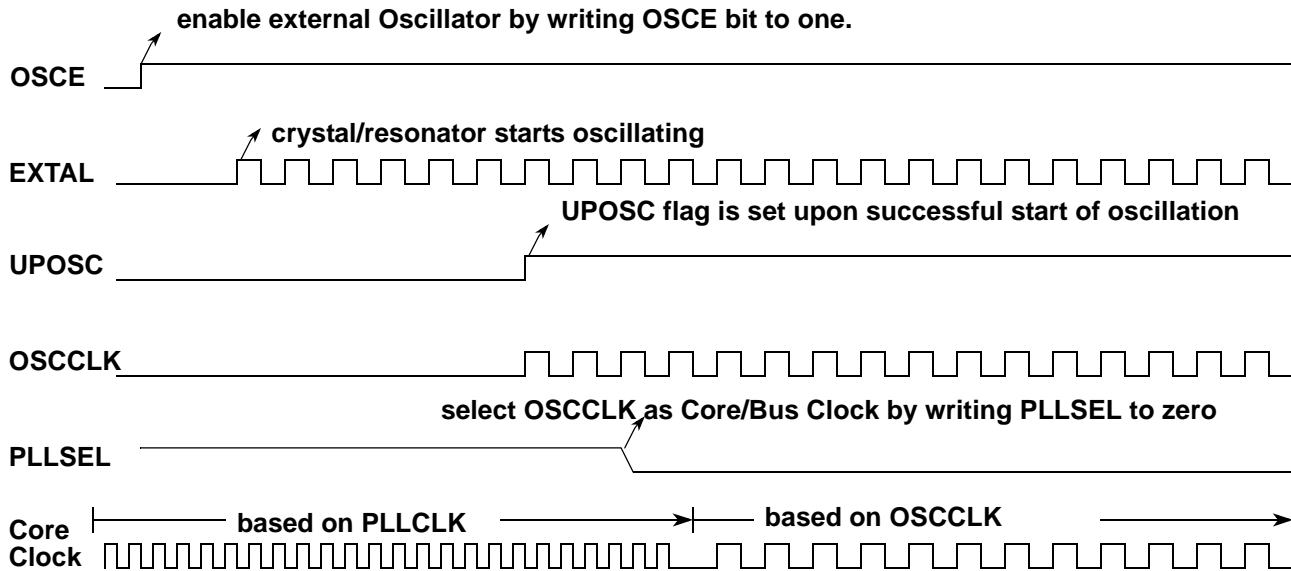
10.4.6   System Clock Configurations

10.4.6.1   PLL Engaged Internal Mode (PEI)

This mode is the default mode after System Reset or Power-On Reset.

The Bus clock is based on the PLLCLK, the reference clock for the PLL is internally generated (IRC1M). The PLL is configured to 50 MHz VCOCLK with POSTDIV set to 0x03. If locked (LOCK=1) this results in a PLLCLK of 12.5 MHz and a Bus clock of 6.25 MHz. The PLL can be re-configured to other bus frequencies.

The clock sources for COP and RTI can be based on the internal reference clock generator (IRC1M) or the RC-Oscillator (ACLK).

10.4.6.2   PLL Engaged External Mode (PEE)

In this mode, the Bus clock is based on the PLLCLK as well (like PEI). The reference clock for the PLL is based on the external oscillator.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock or the RC-Oscillator (ACLK).

This mode can be entered from default mode PEI by performing the following steps:

1. Configure the PLL for desired bus frequency.
2. Enable the external oscillator (OSCE bit).
3. Wait for oscillator to start-up and the PLL being locked (LOCK = 1) and (UPOSC =1).

[1]If only AN0 should be converted use MULT=0.

12.3.2.2 ATD Control Register 1 (ATDCTL1)

Writes to this register will abort current conversion sequence.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
W								
Reset	0	0	1	0	1	1	1	1

Figure 12-4. ATD Control Register 1 (ATDCTL1)

Read: Anytime

Write: Anytime

Table 12-3. ATDCTL1 Field Descriptions

Field	Description
7 ETRIGSEL	**External Trigger Source Select** — This bit selects the external trigger source to be either one of the AD channels or one of the ETRIG3-0 inputs. See device specification for availability and connectivity of ETRIG3-0 inputs. If a particular ETRIG3-0 input option is not available, writing a 1 to ETRISEL only sets the bit but has no effect, this means that one of the AD channels (selected by ETRIGCH3-0) is configured as the source for external trigger. The coding is summarized in Table 12-5.
6–5 SRES[1:0]	**A/D Resolution Select** — These bits select the resolution of A/D conversion results. See Table 12-4 for coding.
4 SMP_DIS	**Discharge Before Sampling Bit** 0 No discharge before sampling. 1 The internal sample capacitor is discharged before sampling the channel. This adds 2 ATD clock cycles to the sampling time. This can help to detect an open circuit instead of measuring the previous sampled channel.
3–0 ETRIGCH[3:0]	**External Trigger Channel Select** — These bits select one of the AD channels or one of the ETRIG3-0 inputs as source for the external trigger. The coding is summarized in Table 12-5.

Table 12-4. A/D Resolution Coding

SRES1	SRES0	A/D Resolution
0	0	8-bit data
0	1	10-bit data
1	0	12-bit data
1	1	Reserved

Table 17-5. DAC Modes of Operation

Operational amplifier	001	disabled	enabled	disabled	depend on AMPP and AMPM input
Unbuffered DAC	100	enabled	disabled	unbuffered resistor output voltage	disconnected
Unbuffered DAC with Operational amplifier	101	enabled	enabled	unbuffered resistor output voltage	depend on AMPP and AMPM input
Buffered DAC	111	enabled	enabled	disconnected	buffered resistor output voltage

The DAC resistor network itself can work on two different voltage ranges:

Table 17-6. DAC Resistor Network Voltage ranges

DAC Mode	Description
Full Voltage Range (FVR)	DAC resistor network provides a output voltage over the complete input voltage range, default after reset
Reduced Voltage Range	DAC resistor network provides a output voltage over a reduced input voltage range

Table 17-7 shows the control signal decoding for each mode. For more detailed mode description see the sections below.

Table 17-7. DAC Control Signals

DACM		DAC resistor network	Operational Amplifier	Switch S1	Switch S2	Switch S3
Off	000	disabled	disabled	open	open	open
Operational amplifier	001	disabled	enabled	closed	open	open
Unbuffered DAC	100	enabled	disabled	open	open	closed
Unbuffered DAC with Operational amplifier	101	enabled	enabled	closed	open	closed
Buffered DAC	111	enabled	enabled	open	closed	open

17.5.2   Mode "Off"

The "Off" mode is the default mode after reset and is selected by DACCTL.DACM[2:0] = 0x0. During this mode the DAC resistor network and the operational amplifier are disabled and all switches are open. This mode provides the lowest power consumption. For decoding of the control signals see Table 17-7.

17.5.3   Mode "Operational Amplifier"

The "Operational Amplifier" mode is selected by DACCTL.DACM[2:0] = 0x1. During this mode the operational amplifier can be used independent from the DAC resister network. All required amplifier signals, AMP, AMPP and AMPM are available on the pins. The DAC resistor network output is disconnected from the DACU pin. The connection between the amplifier output and the negative amplifier input is open. For decoding of the control signals see Table 17-7.

Module Base + 0x0007   Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
W						TXEIE2	TXEIE1	TXEIE0
Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 18-11. MSCAN Transmitter Interrupt Enable Register (CANTIER)

[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 18-14. 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.

18.3.2.9   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						ABTRQ2	ABTRQ1	ABTRQ0
Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 18-12. 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).

18.3.2.13 MSCAN Reserved Register

This register is reserved for factory testing of the MSCAN module and is not available in normal system operating modes.

Module Base + 0x000C   Access: User read/write[1]

	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

Figure 18-16. MSCAN Reserved Register

[1] Read: Always reads zero in normal system operation modes
Write: Unimplemented in normal system operation modes

NOTE

Writing to this register when in special system operating modes can alter the MSCAN functionality.

18.3.2.14 MSCAN Miscellaneous Register (CANMISC)

This register provides additional features.

Module Base + 0x000D   Access: User read/write[1]

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

= Unimplemented

Figure 18-17. MSCAN Miscellaneous Register (CANMISC)

[1] Read: Anytime
Write: Anytime; write of '1' clears flag; write of '0' ignored

Table 18-21. CANMISC Register Field Descriptions

Field	Description
0 BOHOLD	**Bus-off State Hold Until User Request** — If BORM is set in MSCAN Control Register 1 (CANCTL1), this bit indicates whether the module has entered the bus-off state. Clearing this bit requests the recovery from bus-off. Refer to Section 18.5.2, "Bus-Off Recovery," for details. 0 Module is not bus-off or recovery has been requested by user in bus-off state 1 Module is bus-off and holds this state until user request

Chapter 19
Pulse-Width Modulator (S12PWM8B8CV2)

19.1 Introduction

The Version 2 of S12 PWM module is a channel scalable and optimized implementation of S12 PWM8B8C Version 1. The channel is scalable in pairs from PWM0 to PWM7 and the available channel number is 2, 4, 6 and 8. The shutdown feature has been removed and the flexibility to select one of four clock sources per channel has improved. If the corresponding channels exist and shutdown feature is not used, the Version 2 is fully software compatible to Version 1.

19.1.1 Features

The scalable PWM block includes these distinctive features:

- Up to eight independent PWM channels, scalable in pairs (PWM0 to PWM7)
- Available channel number could be 2, 4, 6, 8 (refer to device specification for exact number)
- Programmable period and duty cycle for each channel
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels
- Up to eight 8-bit channel or four 16-bit channel PWM resolution
- Four clock sources (A, B, SA, and SB) provide for a wide range of frequencies
- Programmable clock select logic

19.1.2 Modes of Operation

There is a software programmable option for low power consumption in wait mode that disables the input clock to the prescaler.

In freeze mode there is a software programmable option to disable the input clock to the prescaler. This is useful for emulation.

Wait: The prescaler keeps on running, unless PSWAI in PWMCTL is set to 1.

Freeze: The prescaler keeps on running, unless PFRZ in PWMCTL is set to 1.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
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								
0x0015 PWMPER1[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								

☐ = Unimplemented or Reserved

Figure 19-2. The scalable PWM Register Summary (Sheet 1 of 4)

20.4.4　Baud Rate Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value from 0 to 8191 written to the SBR12:SBR0 bits determines the bus clock divisor. The SBR bits are in the SCI baud rate registers (SCIBDH and SCIBDL). The baud rate clock is synchronized with the bus clock and drives the receiver. The baud rate clock divided by 16 drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to one source of error:

- Integer division of the bus clock may not give the exact target frequency.

Table 20-16 lists some examples of achieving target baud rates with a bus clock frequency of 25 MHz.

When IREN = 0 then,

SCI baud rate = SCI bus clock / (16 * SCIBR[12:0])

Table 20-16. Baud Rates (Example: Bus Clock = 25 MHz)

Bits SBR[12:0]	Receiver Clock (Hz)	Transmitter Clock (Hz)	Target Baud Rate	Error (%)
41	609,756.1	38,109.8	38,400	.76
81	308,642.0	19,290.1	19,200	.47
163	153,374.2	9585.9	9,600	.16
326	76,687.1	4792.9	4,800	.15
651	38,402.5	2400.2	2,400	.01
1302	19,201.2	1200.1	1,200	.01
2604	9600.6	600.0	600	.00
5208	4800.0	300.0	300	.00

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register will be lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a tranmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

21.4.7.3   SPI in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

21.4.7.4   Reset

The reset values of registers and signals are described in Section 21.3, "Memory Map and Register Definition", which details the registers and their bit fields.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it will transmit garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset will always read zeros.

21.4.7.5   Interrupts

The SPI only originates interrupt requests when SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the SPI makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

21.4.7.5.1   MODF

MODF occurs when the master detects an error on the $\overline{SS}$ pin. The master SPI must be configured for the MODF feature (see Table 21-2). After MODF is set, the current transfer is aborted and the following bit is changed:

- MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in Section 21.3.2.4, "SPI Status Register (SPISR)".

Table 22-16. PTPSR Field Descriptions

Field	Description
7:0 PTPS[7:0]	**Precision Timer Prescaler Select Bits** — These eight bits specify the division rate of the main Timer prescaler. These are effective only when the PRNT bit of TSCR1 is set to 1. Table 22-17 shows some selection examples in this case. The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

The Prescaler can be calculated as follows depending on logical value of the PTPS[7:0] and PRNT bit:

PRNT = 1 : Prescaler = PTPS[7:0] + 1

Table 22-17. Precision Timer Prescaler Selection Examples when PRNT = 1

PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0	Prescale Factor
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	3
0	0	0	0	0	0	1	1	4
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
0	0	0	1	0	0	1	1	20
0	0	0	1	0	1	0	0	21
0	0	0	1	0	1	0	1	22
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	0	0	253
1	1	1	1	1	1	0	1	254
1	1	1	1	1	1	1	0	255
1	1	1	1	1	1	1	1	256

22.4 Functional Description

This section provides a complete functional description of the timer TIM16B6CV3 block. Please refer to the detailed timer block diagram in Figure 22-22 as necessary.

24.4.8 Wait Mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Section 24.4.7, "Interrupts").

24.4.9 Stop Mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation will be completed before the MCU is allowed to enter stop mode.

24.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 24-11). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0x3_FF0F. The security state out of reset can be permanently changed by programming the security byte assuming that the MCU is starting from a mode where the necessary P-Flash erase and program commands are available and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using Backdoor Key Access
- Unsecuring the MCU in Special Single Chip Mode using BDM
- Mode and Security Effects on Flash Command Availability

24.5.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x3_FF00-0x3_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 24.3.2.2), the Verify Backdoor Access Key command (see Section 24.4.6.11) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 24-11) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash memory and EEPROM memory will not be available for read access and will return invalid data.

25.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register.
Byte wide reads and writes are allowed to the FCCOB register.
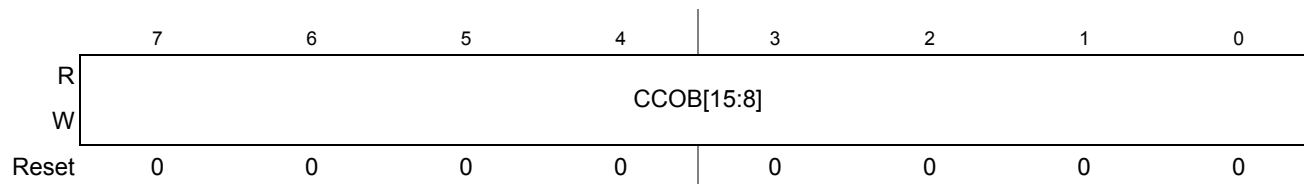
Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R				CCOB[15:8]				
W								
Reset	0	0	0	0	0	0	0	0

Figure 25-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

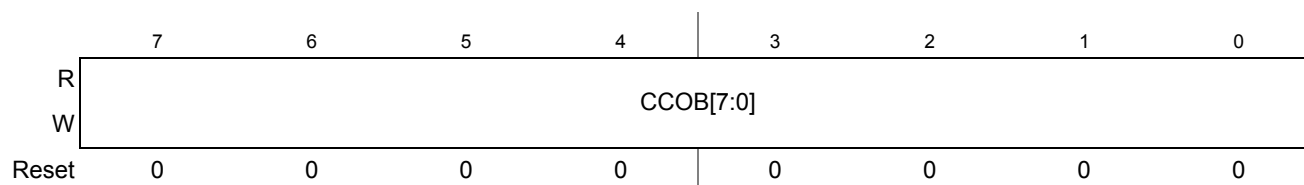|	7	6	5	4	3	2	1	0
R				CCOB[7:0]				
W								
Reset	0	0	0	0	0	0	0	0

Figure 25-17. Flash Common Command Object Low Register (FCCOBLO)

25.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant
parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates
the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears
the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all
FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as
evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the
FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 25-24.
The return values are available for reading after the CCIF flag in the FSTAT register has been returned to
1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX
= 111) are ignored with reads from these fields returning 0x0000.

Table 25-24 shows the generic Flash command format. The high byte of the first word in the CCOB array
contains the command code, followed by the parameters for this specific Flash command. For details on
the FCCOB settings required by each command, see the Flash command descriptions in Section 25.4.6.

Table 25-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]

Table 26-3. P-Flash Memory Addressing

Global Address	Size (Bytes)	Description
0x3_4000 – 0x3_FFFF	48 K	P-Flash Block Contains Flash Configuration Field (see Table 26-4).

The FPROT register, described in Section 26.3.2.9, can be set to protect regions in the Flash memory from accidental program or erase. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 26-4.

indicated by reset condition F in Figure 29-6. If a double bit fault is detected while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 29-9. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 29-10.
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 29-11. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 29-10. Flash KEYEN States

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

[1] Preferred KEYEN state to disable backdoor key access.

Table 29-11. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED[1]
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 29.5.

29.3.2.3   Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.
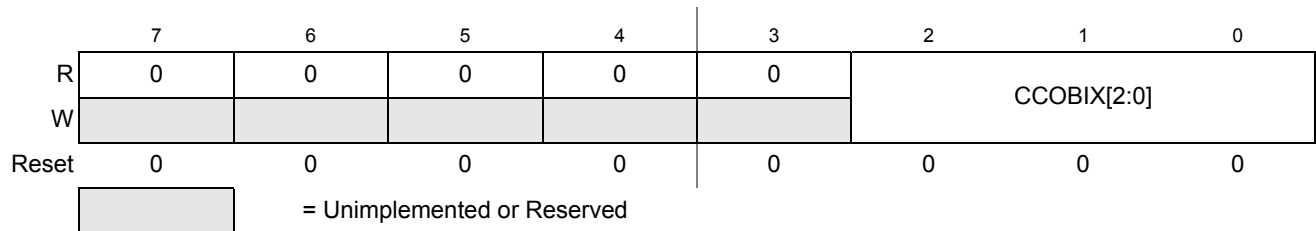
Offset Module Base + 0x0002

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

▨	= Unimplemented or Reserved

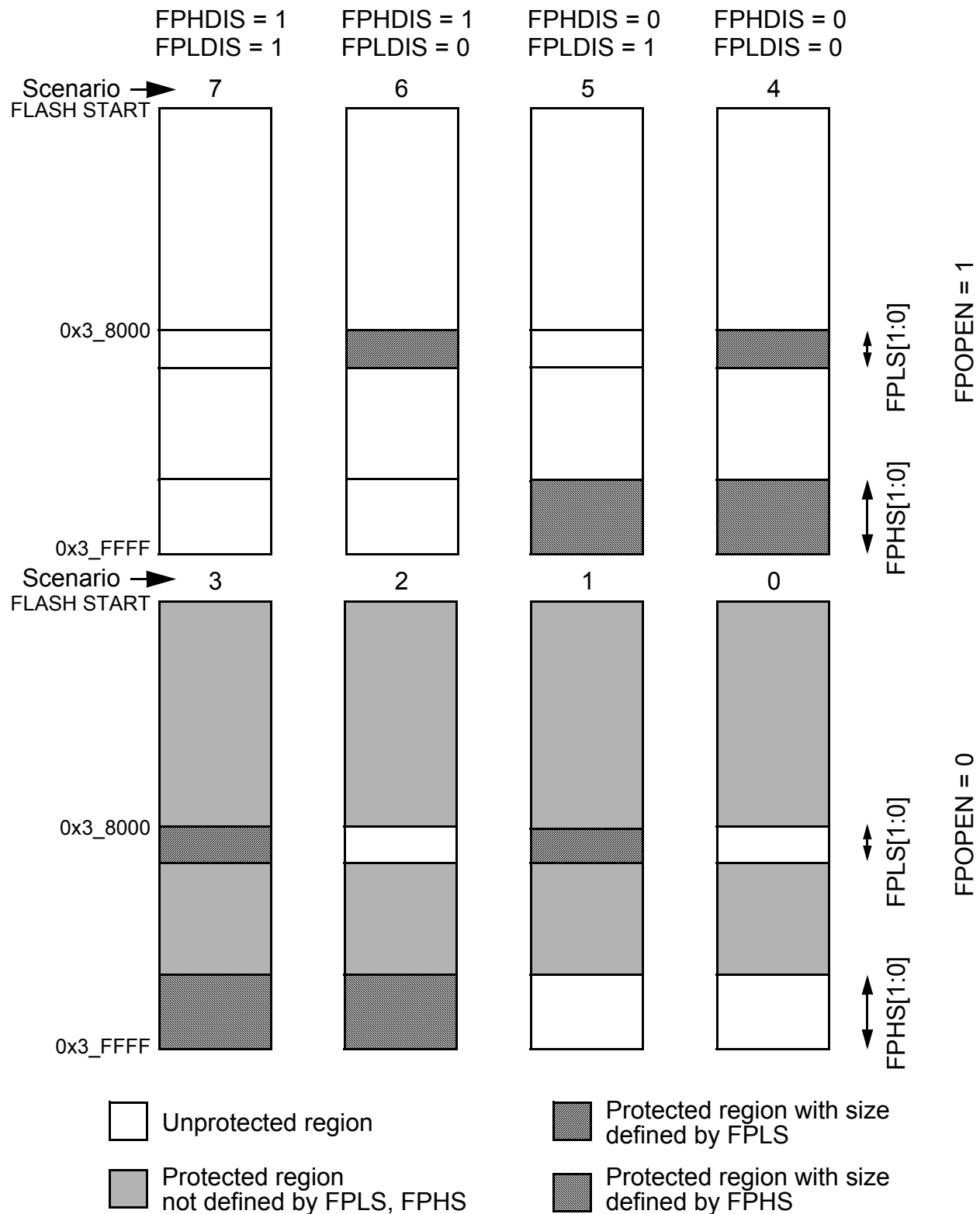**Figure 29-7. FCCOB Index Register (FCCOBIX)**

MC9S12G Family Reference Manual Rev.1.27

Figure 31-14. P-Flash Protection Scenarios