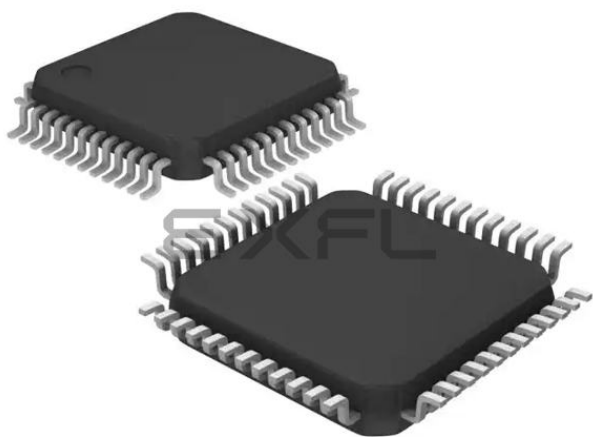




Welcome to [E-XFL.COM](https://www.e-xfl.com)

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	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/s9s12g64f0mlf

Table 1-13. 48-Pin LQFP Pinout for S12GN48

Package Pin	Function <----lowest-----PRIORITY-----highest---->					Power Supply	Internal Pull Resistor	
	Pin	2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
28	PAD9	KWAD9	AN9	ACMPO	—	V _{DDA}	PER0AD/PPS0AD	Disabled
29	PAD2	KWAD2	AN2	—	—	V _{DDA}	PER1AD/PPS1AD	Disabled
30	PAD10	KWAD10	AN10	ACMPPI		V _{DDA}	PER0AD/PPS0AD	Disabled
31	PAD3	KWAD3	AN3	—	—	V _{DDA}	PER1AD/PPS1AD	Disabled
32	PAD11	KWAD11	AN11	ACMPPI		V _{DDA}	PER0AD/PPS0AD	Disabled
33	PAD4	KWAD4	AN4	—	—	V _{DDA}	PER1AD/PPS1AD	Disabled
34	PAD5	KWAD5	AN5	—	—	V _{DDA}	PER1AD/PPS0AD	Disabled
35	PAD6	KWAD6	AN6	—	—	V _{DDA}	PER1AD/PPS1AD	Disabled
36	PAD7	KWAD7	AN7	—	—	V _{DDA}	PER1AD/PPS1AD	Disabled
37	VDDA	VRH	—	—	—	—	—	—
38	VSSA	—	—	—	—	—	—	—
39	PS0	RXD0	—	—	—	V _{DDX}	PERS/PPSS	Up
40	PS1	TXD0	—	—	—	V _{DDX}	PERS/PPSS	Up
41	PS2	RXD1	—	—	—	V _{DDX}	PERS/PPSS	Up
42	PS3	TXD1	—	—	—	V _{DDX}	PERS/PPSS	Up
43	PS4	MISO0	—	—	—	V _{DDX}	PERS/PPSS	Up
44	PS5	MOSI0	—	—	—	V _{DDX}	PERS/PPSS	Up
45	PS6	SCK0	—	—	—	V _{DDX}	PERS/PPSS	Up
46	PS7	API_EXTC LK	ECLK	$\overline{SS0}$	—	V _{DDX}	PERS/PPSS	Up
47	PM0	—	—	—	—	V _{DDX}	PERM/PPSM	Disabled
48	PM1	—	—	—	—	V _{DDX}	PERM/PPSM	Disabled

¹ The regular I/O characteristics (see [Section A.2, “I/O Characteristics”](#)) apply if the EXTAL/XTAL function is disabled

1.8.5 S12GA48 and S12GA64

1.8.5.1 Pinout 48-Pin LQFP

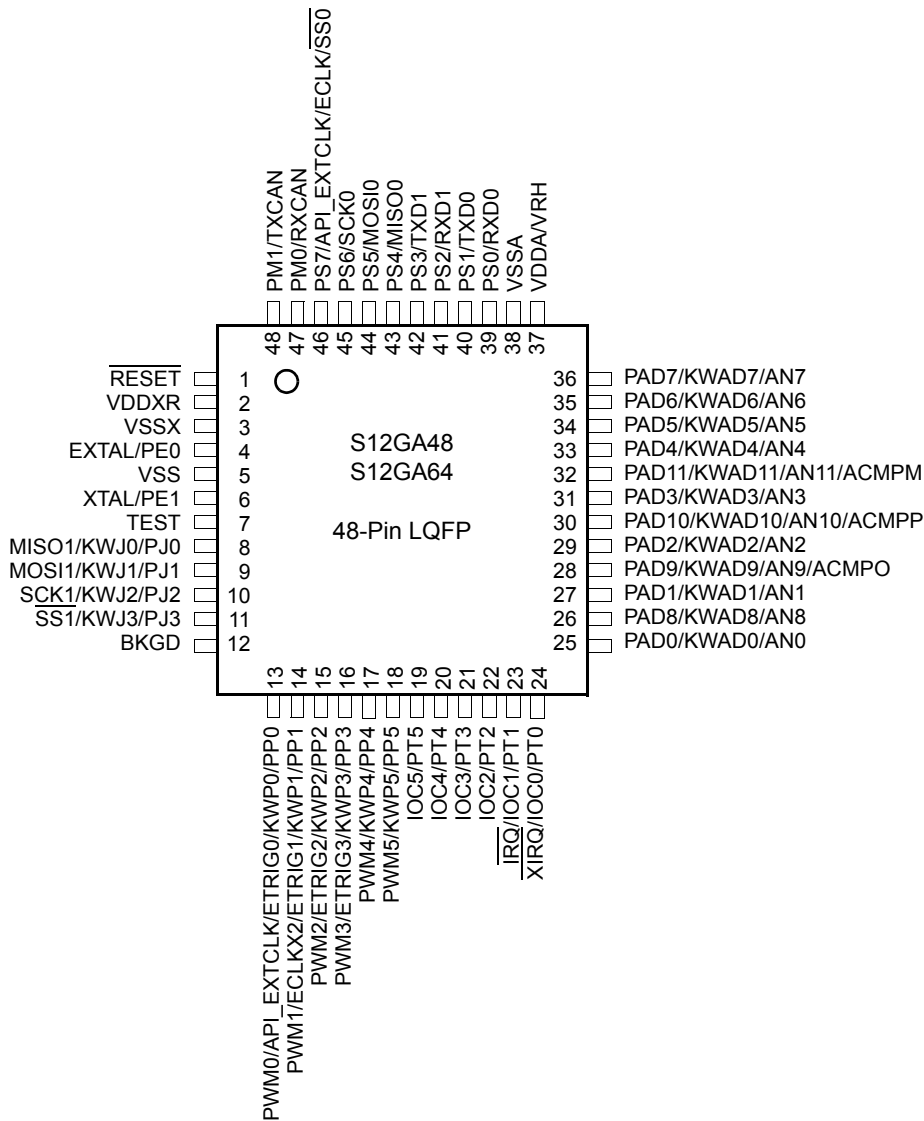


Figure 1-13. 48-Pin LQFP Pinout for S12GA48 and S12GA64

Table 1-18. 48-Pin LQFP Pinout for S12GA48 and S12GA64

Package Pin	Pin	Function <---lowest---PRIORITY---highest--->				Power Supply	Internal Pull Resistor	
		2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
1	RESET	—	—	—	—	V _{DDX}	PULLUP	

1.8.5.2 Pinout 64-Pin LQFP

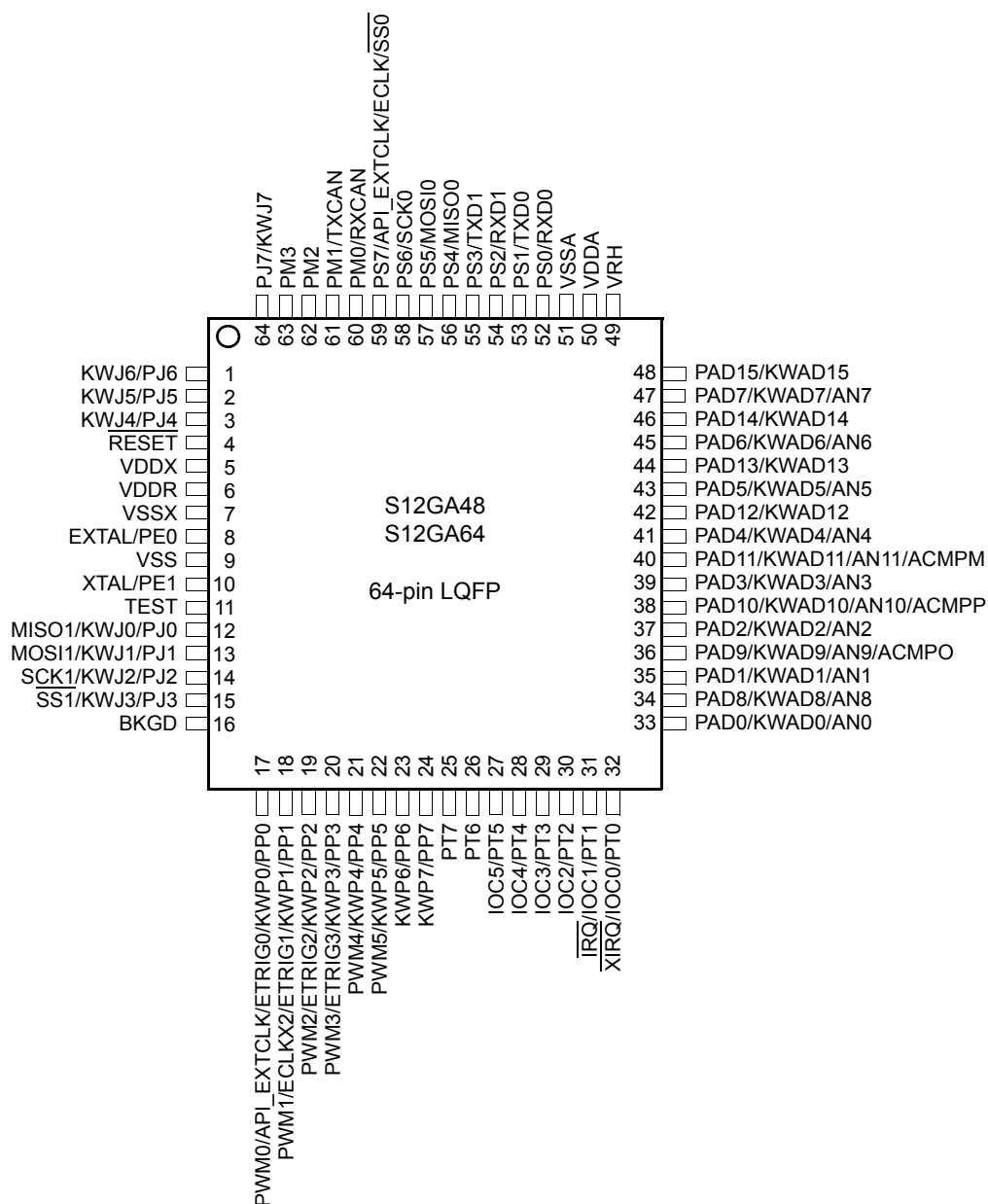


Figure 1-14. 64-Pin LQFP Pinout for S12GA48 and S12GA64

Table 1-19. 64-Pin LQFP Pinout for S12GA48 and S12GA64

Package Pin	Function <---lowest---PRIORITY---highest--->					Power Supply	Internal Pull Resistor	
	Pin	2nd Func.	3rd Func.	4th Func	5th Func		CTRL	Reset State
1	PJ6	KWJ6	—	—	—	V _{DDX}	PERJ/PPSJ	Up
2	PJ5	KWJ5	—	—	—	V _{DDX}	PERJ/PPSJ	Up
3	PJ4	KWJ4	—	—	—	V _{DDX}	PERJ/PPSJ	Up
4	RESET	—	—	—	—	V _{DDX}	PULLUP	
5	VDDX	—	—	—	—	—	—	—
6	VDDR	—	—	—	—	—	—	—
7	VSSX	—	—	—	—	—	—	—
8	PE0 ¹	EXTAL	—	—	—	V _{DDX}	PUCR/PDPEE	Down
9	VSS	—	—	—	—	—	—	—
10	PE1 ¹	XTAL	—	—	—	V _{DDX}	PUCR/PDPEE	Down
11	TEST	—	—	—	—	N.A.	$\overline{\text{RESET}}$ pin	Down
12	PJ0	KWJ0	MISO1	—	—	V _{DDX}	PERJ/PPSJ	Up
13	PJ1	KWJ1	MOSI1	—	—	V _{DDX}	PERJ/PPSJ	Up
14	PJ2	KWJ2	SCK1	—	—	V _{DDX}	PERJ/PPSJ	Up
15	PJ3	KWJ3	$\overline{\text{SS1}}$	—	—	V _{DDX}	PERJ/PPSJ	Up
16	BKGD	MODC	—	—	—	V _{DDX}	PUCR/BKPUE	Up
17	PP0	KWP0	ETRIG0	API_EXTC LK	PWM0	V _{DDX}	PERP/PPSP	Disabled
18	PP1	KWP1	ETRIG1	ECLKX2	PWM1	V _{DDX}	PERP/PPSP	Disabled
19	PP2	KWP2	ETRIG2	PWM2	—	V _{DDX}	PERP/PPSP	Disabled
20	PP3	KWP3	ETRIG3	PWM3	—	V _{DDX}	PERP/PPSP	Disabled
21	PP4	KWP4	PWM4	—	—	V _{DDX}	PERP/PPSP	Disabled
22	PP5	KWP5	PWM5	—	—	V _{DDX}	PERP/PPSP	Disabled
23	PP6	KWP6	—	—	—	V _{DDX}	PERP/PPSP	Disabled
24	PP7	KWP7	—	—	—	V _{DDX}	PERP/PPSP	Disabled
25	PT7	—	—	—	—	V _{DDX}	PERT/PPST	Disabled
26	PT6	—	—	—	—	V _{DDX}	PERT/PPST	Disabled
27	PT5	IOC5	—	—	—	V _{DDX}	PERT/PPST	Disabled

Table 2-4. Signals and Priorities

Port	Pin	Signal	Signals per Device and Package (signal priority on pin from top to bottom)																				Legend				
			GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	G64 / GA64 / G48 / GA48	GN48	GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	G64 / GA64 / G48 / GA48	GN48	GN32 / GNA32	GN16 / GNA16	G64 / G48	GN48	GN32	GN16	GN32	GN16	?	Signal available on pin		
																									?	Routing option on pin	
																									?	Routing reset location	
																										?	Not available on pin
			100			64				48								32				20		I/O	Description		
S	PS7	SS0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	SPI slave select			
		TXD0																				?	?	I/O	SCI transmit		
		PWM5																?	?	?	?			O	PWM channel		
		PWM3																				?	?	O	PWM channel		
		ECLK				?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	O	Free-running clock		
		API_EXTCLK	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	O	API Clock		
		ETRIG3																				?	?	I	ADC external trigger		
		[PTS7]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	GPIO		
	PS6	SCK0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	SPI serial clock		
		IOC5																?	?	?	?			I/O	Timer channel		
		IOC3																				?	?	I/O	Timer channel		
		[PTS6]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	GPIO		
	PS5	MOSI0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	SPI master out/slave in		
		IOC4																?	?	?	?			I/O	Timer channel		
		IOC2																				?	?	I/O	Timer channel		
		[PTS5]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	GPIO		
	PS4	MISO0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	SPI master in/slave out		
		RXD0																				?	?	I	SCI receive pin		
		PWM4																?	?	?	?			O	PWM channel		
		PWM2																				?	?	O	PWM channel		
		ETRIG2																				?	?	I	ADC external trigger		
		[PTS4]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	I/O	GPIO		
	PS3	TXD1	?	?	?	?	?	?	?	?	?	?	?	?	?									I/O	SCI transmit		
		[PTS3]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?							I/O	GPIO		
	PS2	RXD1	?	?	?	?	?	?	?	?	?	?	?	?	?									I	SCI receive		
		[PTS2]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?							I/O	GPIO		
	PS1	TXD0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	SCI transmit		
		[PTS1]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	GPIO		
	PS0	RXD0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I	SCI receive		
		[PTS0]	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	GPIO		

4.7 Functional Description

The RVA is a prescaler for the ADC reference voltage. If the attenuation is turned off the resistive divider is disconnected from VSSA, VRH_INT is connected to VRH and VRL_INT is connected to VSSA. In this mode the attenuation is bypassed and the resistive divider does not draw current.

If the attenuation is turned on the resistive divider is connected to VSSA, VRH_INT and VRL_INT are connected to intermediate voltage levels:

$$\text{VRH_INT} = 0.9 * (\text{VRH} - \text{VSSA}) + \text{VSSA} \quad \text{Eqn. 4-1}$$

$$\text{VRL_INT} = 0.4 * (\text{VRH} - \text{VSSA}) + \text{VSSA} \quad \text{Eqn. 4-2}$$

The attenuated reference voltage difference (VRH_INT - VRL_INT) equals 50% of the input reference voltage difference (VRH - VSSA). With reference voltage attenuation the resolution of the ADC is improved by a factor of 2.

NOTE

In attenuation mode the maximum ADC clock is reduced. Please refer to the conditions in appendix A “ATD Accuracy”, table “ATD Conversion Performance 5V range, RVA enabled”.

to erase the EEPROM and Flash memory without giving access to their contents. After erasing both Flash memory and EEPROM, another reset into special single chip mode will cause the blank check to succeed and the options/security byte can be programmed to “unsecured” state via BDM.

While the BDM is executing the blank check, the BDM interface is completely blocked, which means that all BDM commands are temporarily blocked.

9.1.5 Unsecuring the Microcontroller

Unsecuring the microcontroller can be done by three different methods:

1. Backdoor key access
2. Reprogramming the security bits
3. Complete memory erase (special modes)

9.1.5.1 Unsecuring the MCU Using the Backdoor Key Access

In normal modes (single chip and expanded), security can be temporarily disabled using the backdoor key access method. This method requires that:

- The backdoor key at 0xFF00–0xFF07 (= global addresses 0x3_FF00–0x3_FF07) has been programmed to a valid value.
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’.
- In single chip mode, the application program programmed into the microcontroller must be designed to have the capability to write to the backdoor key locations.

The backdoor key values themselves would not normally be stored within the application data, which means the application program would have to be designed to receive the backdoor key values from an external source (e.g. through a serial port).

The backdoor key access method allows debugging of a secured microcontroller without having to erase the Flash. This is particularly useful for failure analysis.

NOTE

No word of the backdoor key is allowed to have the value 0x0000 or 0xFFFF.

9.1.6 Reprogramming the Security Bits

In normal single chip mode (NS), security can also be disabled by erasing and reprogramming the security bits within Flash options/security byte to the unsecured value. Because the erase operation will erase the entire sector from 0xFE00–0xFFFF (0x7F_FE00–0x7F_FFFF), the backdoor key and the interrupt vectors will also be erased; this method is not recommended for normal single chip mode. The application software can only erase and program the Flash options/security byte if the Flash sector containing the Flash options/security byte is not protected (see Flash protection). Thus Flash protection is a useful means of preventing this method. The microcontroller will enter the unsecured state after the next reset following the programming of the security bits to the unsecured value.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003	ATDCTL3	R W	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
0x0004	ATDCTL4	R W	SMP2	SMP1	SMP0	PRS[4:0]				
0x0005	ATDCTL5	R W	0	SC	SCAN	MULT	CD	CC	CB	CA
0x0006	ATDSTAT0	R W	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
0x0007	Unimplemented	R W	0	0	0	0	0	0	0	0
0x0008	ATDCMPEH	R W	0	0	0	0	0	0	0	0
0x0009	ATDCMPEL	R W	CMPE[7:0]							
0x000A	ATDSTAT2H	R W	0	0	0	0	0	0	0	0
0x000B	ATDSTAT2L	R W	CCF[7:0]							
0x000C	ATDDIENH	R W	1	1	1	1	1	1	1	1
0x000D	ATDDIENL	R W	IEN[7:0]							
0x000E	ATDCMPHTH	R W	0	0	0	0	0	0	0	0
0x000F	ATDCMPHTL	R W	CMPHT[7:0]							
0x0010	ATDDR0	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x0012	ATDDR1	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x0014	ATDDR2	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x0016	ATDDR3	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x0018	ATDDR4	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x001A	ATDDR5	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x001C	ATDDR6	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x001E	ATDDR7	R W	See Section 12.3.2.12.1 , “Left Justified Result Data (DJM=0)” and Section 12.3.2.12.2 , “Right Justified Result Data (DJM=1)”							
0x0020 - 0x002F	Unimplemented	R W	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

Figure 12-2. ADC12B8C Register Summary (Sheet 2 of 2)

Table 13-11. ATD Behavior in Freeze Mode (Breakpoint)

FRZ1	FRZ0	Behavior in Freeze Mode
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze Immediately

13.3.2.5 ATD Control Register 4 (ATDCTL4)

Writes to this register will abort current conversion sequence.

Module Base + 0x0004

**Figure 13-7. ATD Control Register 4 (ATDCTL4)**

Read: Anytime

Write: Anytime

Table 13-12. ATDCTL4 Field Descriptions

Field	Description
7–5 SMP[2:0]	Sample Time Select — These three bits select the length of the sample time in units of ATD conversion clock cycles. Note that the ATD conversion clock period is itself a function of the prescaler value (bits PRS4-0). Table 13-13 lists the available sample time lengths.
4–0 PRS[4:0]	ATD Clock Prescaler — These 5 bits are the binary prescaler value PRS. The ATD conversion clock frequency is calculated as follows: $f_{\text{ATDCLK}} = \frac{f_{\text{BUS}}}{2 \times (\text{PRS} + 1)}$ Refer to Device Specification for allowed frequency range of f_{ATDCLK} .

Table 13-13. Sample Time Select

SMP2	SMP1	SMP0	Sample Time in Number of ATD Clock Cycles
0	0	0	4
0	0	1	6
0	1	0	8
0	1	1	10
1	0	0	12
1	0	1	16
1	1	0	20
1	1	1	24

13.3.2.10 ATD Input Enable Register (ATDDIEN)

Module Base + 0x000C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1	1	1	1	IEN[11:0]											
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

Figure 13-12. ATD Input Enable Register (ATDDIEN)

Read: Anytime

Write: Anytime

Table 13-19. ATDDIEN Field Descriptions

Field	Description
11–0 IEN[11:0]	ATD Digital Input Enable on channel x ($x = 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0$) — This bit controls the digital input buffer from the analog input pin (AN x) to the digital data register. 0 Disable digital input buffer to AN x pin 1 Enable digital input buffer on AN x pin. Note: Setting this bit will enable the corresponding digital input buffer continuously. If this bit is set while simultaneously using it as an analog port, there is potentially increased power consumption because the digital input buffer maybe in the linear region.

13.3.2.11 ATD Compare Higher Than Register (ATDCMPHT)

Writes to this register will abort current conversion sequence.

Read: Anytime

Write: Anytime

Module Base + 0x000E

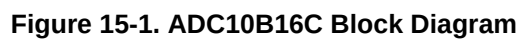
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	CMPHT[11:0]											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

Figure 13-13. ATD Compare Higher Than Register (ATDCMPHT)

Table 13-20. ATDCMPHT Field Descriptions

Field	Description
11–0 CMPHT[11:0]	Compare Operation Higher Than Enable for conversion number n ($n = 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0$) of a Sequence ($n$ conversion number, NOT channel number!) — This bit selects the operator for comparison of conversion results. 0 If result of conversion n is lower or same than compare value in ATDDR n , this is flagged in ATDSTAT2 1 If result of conversion n is higher than compare value in ATDDR n , this is flagged in ATDSTAT2



15.3.2.4 ATD Control Register 3 (ATDCTL3)

Writes to this register will abort current conversion sequence.

Module Base + 0x0003

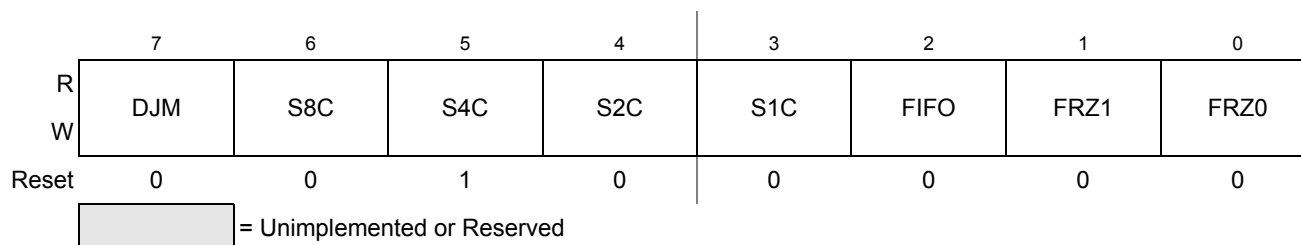


Figure 15-6. ATD Control Register 3 (ATDCTL3)

Read: Anytime

Write: Anytime

Table 15-8. ATDCTL3 Field Descriptions

Field	Description
7 DJM	Result Register Data Justification — Result data format is always unsigned. This bit controls justification of conversion data in the result registers. 0 Left justified data in the result registers. 1 Right justified data in the result registers. Table 15-9 gives example ATD results for an input signal range between 0 and 5.12 Volts.
6–3 S8C, S4C, S2C, S1C	Conversion Sequence Length — These bits control the number of conversions per sequence. Table 15-10 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 family.
2 FIFO	Result Register FIFO Mode — If this bit is zero (non-FIFO mode), the A/D conversion results map into the result registers based on the conversion sequence; the result of the first conversion appears in the first result register (ATDDR0), the second result in the second result register (ATDDR1), and so on. If this bit is one (FIFO mode) the conversion counter is not reset at the beginning or end of a conversion sequence; sequential conversion results are placed in consecutive result registers. In a continuously scanning conversion sequence, the result register counter will wrap around when it reaches the end of the result register file. The conversion counter value (CC3-0 in ATDSTAT0) can be used to determine where in the result register file, the current conversion result will be placed. Aborting a conversion or starting a new conversion clears the conversion counter even if FIFO=1. So the first result of a new conversion sequence, started by writing to ATDCTL5, will always be place in the first result register (ATDDDR0). Intended usage of FIFO mode is continuos conversion (SCAN=1) or triggered conversion (ETRIG=1). Which result registers hold valid data can be tracked using the conversion complete flags. Fast flag clear mode may be useful in a particular application to track valid data. If this bit is one, automatic compare of result registers is always disabled, that is ADC10B16C will behave as if ACMPIE and all CPME[n] were zero. 0 Conversion results are placed in the corresponding result register up to the selected sequence length. 1 Conversion results are placed in consecutive result registers (wrap around at end).
1–0 FRZ[1:0]	Background Debug Freeze Enable — When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. These 2 bits determine how the ATD will respond to a breakpoint as shown in Table 15-11 . Leakage onto the storage node and comparator reference capacitors may compromise the accuracy of an immediately frozen conversion depending on the length of the freeze period.

18.3.2.15 MSCAN Receive Error Counter (CANRXERR)

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

Module Base + 0x000E

Access: User read/write¹

	7	6	5	4	3	2	1	0
R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
W								
Reset:	0	0	0	0	0	0	0	0
	= Unimplemented							

Figure 18-18. MSCAN Receive Error Counter (CANRXERR)

- ¹ Read: Only when in sleep mode (SLPRQ = 1 and SLPK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

NOTE

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

18.3.2.16 MSCAN Transmit Error Counter (CANTXERR)

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

Module Base + 0x000F

Access: User read/write¹

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

Figure 18-19. MSCAN Transmit Error Counter (CANTXERR)

- ¹ Read: Only when in sleep mode (SLPRQ = 1 and SLPK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

NOTE

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

Figure 18-24. Receive/Transmit Message Buffer — Extended Identifier Mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit0
0x00X0 IDR0	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	W								
0x00X1 IDR1	R	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
	W								
0x00X2 IDR2	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	W								
0x00X3 IDR3	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	W								
0x00X4 DSR0	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X5 DSR1	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X6 DSR2	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X7 DSR3	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X8 DSR4	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X9 DSR5	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XA DSR6	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XB DSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XC DLR	R					DLC3	DLC2	DLC1	DLC0
	W								

Module Base + 0x0005

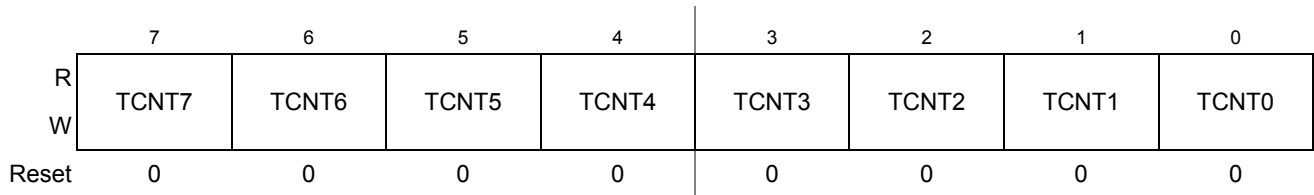


Figure 23-11. Timer Count Register Low (TCNTL)

The 16-bit main timer is an up counter.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read: Anytime

Write: Has no meaning or effect in the normal mode; only writable in special modes (test_mode = 1).

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

23.3.2.6 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

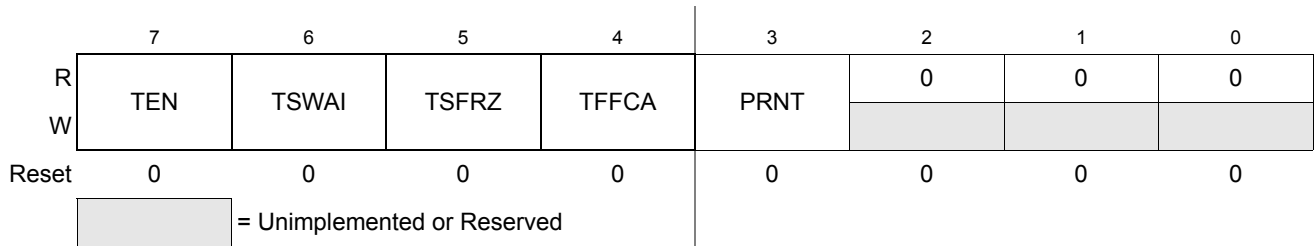


Figure 23-12. Timer System Control Register 1 (TSCR1)

Read: Anytime

Write: Anytime

Table 23-6. TSCR1 Field Descriptions

Field	Description
7 TEN	Timer Enable 0 Disables the main timer, including the counter. Can be used for reducing power consumption. 1 Allows the timer to function normally. If for any reason the timer is not active, there is no ÷64 clock for the pulse accumulator because the ÷64 is generated by the timer prescaler.
6 TSWAI	Timer Module Stops While in Wait 0 Allows the timer module to continue running during wait. 1 Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait. TSWAI also affects pulse accumulator.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0).

Table 27-7. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	Clock Divider Loaded 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset
6 FDIVLCK	Clock Divider Locked 0 FDIV field is open for writing 1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field in normal mode.
5–0 FDIV[5:0]	Clock Divider Bits — FDIV[5:0] must be set to effectively divide BUSCLK down to 1 MHz to control timed events during Flash program and erase algorithms. Table 27-8 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Please refer to Section 27.4.4, “Flash Command Operations,” for more information.

Table 27-8. FDIV values for various BUSCLK Frequencies

BUSCLK Frequency (MHz)		FDIV[5:0]	BUSCLK Frequency (MHz)		FDIV[5:0]
MIN ¹	MAX ²		MIN ¹	MAX ²	
1.0	1.6	0x00	16.6	17.6	0x10
1.6	2.6	0x01	17.6	18.6	0x11
2.6	3.6	0x02	18.6	19.6	0x12
3.6	4.6	0x03	19.6	20.6	0x13
4.6	5.6	0x04	20.6	21.6	0x14
5.6	6.6	0x05	21.6	22.6	0x15
6.6	7.6	0x06	22.6	23.6	0x16
7.6	8.6	0x07	23.6	24.6	0x17
8.6	9.6	0x08	24.6	25.6	0x18
9.6	10.6	0x09			
10.6	11.6	0x0A			
11.6	12.6	0x0B			
12.6	13.6	0x0C			
13.6	14.6	0x0D			
14.6	15.6	0x0E			
15.6	16.6	0x0F			

¹ BUSCLK is Greater Than this value.

² BUSCLK is Less Than or Equal to this value.

Table 31-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

31.3.2.12 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000C

	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 31-18. Flash Reserved1 Register (FRSV1)

All bits in the FRSV1 register read 0 and are not writable.

31.3.2.13 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000D

	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 31-19. Flash Reserved2 Register (FRSV2)

All bits in the FRSV2 register read 0 and are not writable.

31.3.2.14 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

