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	54
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	1.5K 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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g48f0mlh

Chapter 14
Analog-to-Digital Converter (ADC12B12CV2)

Chapter 15
Analog-to-Digital Converter (ADC10B16CV2)

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

	Function <----lowest-----PRIORITY-----highest---->					Power Supply	Internal Pull Resistor	
Package Pin	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	ACMPP		V_{DDA}	PER0AD/PPS0AD	Disabled
31	PAD3	KWAD3	AN3	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled
32	PAD11	KWAD11	AN11	ACMPM		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_EXTCLK	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

Table 2-19. Block Register Map (G1) (continued)

Global Address Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0255 PPSM	R	0	0	0	0	PPSM3	PPSM2	PPSM1	PPSM0
	W								
0x0256 WOMM	R	0	0	0	0	WOMM3	WOMM2	WOMM1	WOMM0
	W								
0x0257 PKGCR	R	APICLKS7	0	0	0	0	PKGCR2	PKGCR1	PKGCR0
	W								
0x0258 PTP	R	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
	W								
0x0259 PTIP	R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
	W								
0x025A DDRP	R	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
	W								
0x025B Reserved	R	0	0	0	0	0	0	0	0
	W								
0x025C PERP	R	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
	W								
0x025D PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
	W								
0x025E PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
	W								
0x025F PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
	W								
0x0260–0x0267 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0268 PTJ	R	PTJ7	PTJ6	PTJ5	PTJ4	PTJ3	PTJ2	PTJ1	PTJ0
	W								
0x0269 PTIJ	R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	PTIJ3	PTIJ2	PTIJ1	PTIJ0
	W								
0x026A DDRJ	R	DDRJ7	DDRJ6	DDRJ5	DDRJ4	DDRJ3	DDRJ2	DDRJ1	DDRJ0
	W								

= Unimplemented or Reserved

NOTE

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

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target will issue the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands will ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands will ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin and when the data bus cycle is complete. See Section 7.4.3, “BDM Hardware Commands” and Section 7.4.4, “Standard BDM Firmware Commands” for more information on the BDM commands.

The ACK_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK_ENABLE command is ignored by the target since it is not recognized as a valid command.

The BACKGROUND command will issue an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command will issue an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO_UNTIL command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

10.6 Interrupts

The interrupt/reset vectors requested by the S12CPMU are listed in Table 10-29. Refer to MCU specification for related vector addresses and priorities.

Table 10-29. S12CPMU Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
RTI time-out interrupt	I bit	CPMUINT (RTIE)
PLL lock interrupt	I bit	CPMUINT (LOCKIE)
Oscillator status interrupt	I bit	CPMUINT (OSCIE)
Low voltage interrupt	I bit	CPMULVCTL (LVIE)
Autonomous Periodical Interrupt	I bit	CPMUAPICTL (APIE)

10.6.1 Description of Interrupt Operation

10.6.1.1 Real Time Interrupt (RTI)

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop Mode with PSTP=1 (Pseudo Stop Mode), RTIOSCSEL=1 and PRE=1 the RTI continues to run, else the RTI counter halts in Stop Mode.

The RTI can be used to generate hardware interrupts at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the CPMURTI register. At the end of the RTI time-out period the RTIF flag is set to one and a new RTI time-out period starts immediately.

A write to the CPMURTI register restarts the RTI time-out period.

10.6.1.2 PLL Lock Interrupt

The S12CPMU generates a PLL Lock interrupt when the lock condition (LOCK status bit) of the PLL changes, either from a locked state to an unlocked state or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to zero. The PLL Lock interrupt flag (LOCKIF) is set to 1 when the lock condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

10.6.1.3 Oscillator Status Interrupt

When the OSCE bit is 0, then UPOSC stays 0. When OSCE = 1 the UPOSC bit is set after the LOCK bit is set.

Upon detection of a status change (UPOSC) the OSCIF flag is set. Going into Full Stop Mode or disabling the oscillator can also cause a status change of UPOSC.

Table 15-16. ATDSTAT0 Field Descriptions (continued)

Field	Description
3–0 CC[3:0]	**Conversion Counter** — These 4 read-only bits are the binary value of the conversion counter. The conversion counter points to the result register that will receive the result of the current conversion. E.g. CC3=0, CC2=1, CC1=1, CC0=0 indicates that the result of the current conversion will be in ATD Result Register 6. If in non-FIFO mode (FIFO=0) the conversion counter is initialized to zero at the beginning and end of the conversion sequence. If in FIFO mode (FIFO=1) the register counter is not initialized. The conversion counter wraps around when its maximum value is reached. Aborting a conversion or starting a new conversion clears the conversion counter even if FIFO=1.

15.3.2.8 ATD Compare Enable Register (ATDCMPE)

Writes to this register will abort current conversion sequence.

Read: Anytime

Write: Anytime

Module Base + 0x0008

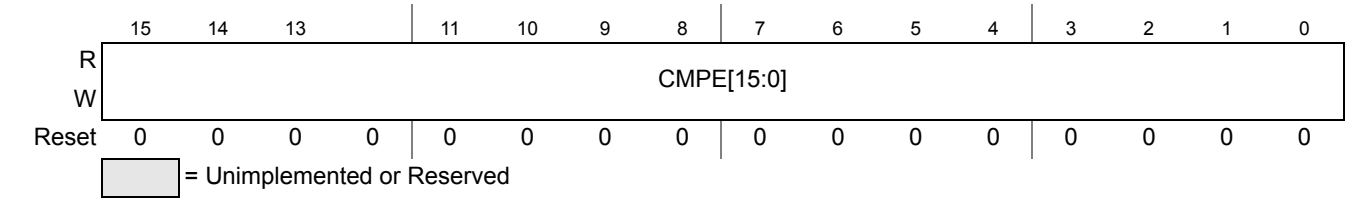

Figure 15-10. ATD Compare Enable Register (ATDCMPE)

Table 15-17. ATDCMPE Field Descriptions

Field	Description
15–0 CMPE[15:0]	**Compare Enable for Conversion Number *n* (*n*= 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0) of a Sequence (*n conversion number, NOT channel number!)*** — These bits enable automatic compare of conversion results individually for conversions of a sequence. The sense of each comparison is determined by the CMPHT[*n*] bit in the ATDCMPHT register. For each conversion number with CMPE[*n*]=1 do the following: 1) Write compare value to ATDDR*n* result register 2) Write compare operator with CMPHT[*n*] in ATDCPMHT register CCF[*n*] in ATDSTAT2 register will flag individual success of any comparison. 0 No automatic compare 1 Automatic compare of results for conversion *n* of a sequence is enabled.

16.3.2.12 ATD Conversion Result Registers (ATDDR*n*)

The A/D conversion results are stored in 16 result registers. Results are always in unsigned data representation. Left and right justification is selected using the DJM control bit in ATDCTL3.

If automatic compare of conversions results is enabled (CMPE[*n*]=1 in ATDCMPE), these registers must be written with the compare values in left or right justified format depending on the actual value of the DJM bit. In this case, as the ATDDR*n* register is used to hold the compare value, the result will not be stored there at the end of the conversion but is lost.

Attention, n is the conversion number, NOT the channel number!

Read: Anytime

Write: Anytime

NOTE

For conversions not using automatic compare, results are stored in the result registers after each conversion. In this case avoid writing to ATDDRn except for initial values, because an A/D result might be overwritten.

16.3.2.12.1 Left Justified Result Data (DJM=0)

Module Base +
0x0010 = ATDDR0, 0x0012 = ATDDR1, 0x0014 = ATDDR2, 0x0016 = ATDDR3
0x0018 = ATDDR4, 0x001A = ATDDR5, 0x001C = ATDDR6, 0x001E = ATDDR7
0x0020 = ATDDR8, 0x0022 = ATDDR9, 0x0024 = ATDDR10, 0x0026 = ATDDR11
0x0028 = ATDDR12, 0x002A = ATDDR13, 0x002C = ATDDR14, 0x002E = ATDDR15

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

= Unimplemented or Reserved

Figure 16-14. Left justified ATD conversion result register (ATDDR*n*)

Table 16-21 shows how depending on the A/D resolution the conversion result is transferred to the ATD result registers for left justified data. Compare is always done using all 12 bits of both the conversion result and the compare value in ATDDRn.

Table 16-21. Conversion result mapping to ATDDRn

A/D resolution	DJM	conversion result mapping to ATDDR*n*
8-bit data	0	Result-Bit[11:4] = conversion result, Result-Bit[3:0]=0000
10-bit data	0	Result-Bit[11:2] = conversion result, Result-Bit[1:0]=00
12-bit data	0	Result-Bit[11:0] = result

18.3.3.1.2 IDR0–IDR3 for Standard Identifier Mapping

Module Base + 0x00X0

	7	6	5	4	3	2	1	0
R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
Reset:	x	x	x	x	x	x	x	x

Figure 18-30. Identifier Register 0 — Standard Mapping

Table 18-31. IDR0 Register Field Descriptions — Standard

Field	Description
7-0 ID[10:3]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 18-32.

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID2	ID1	ID0	RTR	IDE (=0)			
Reset:	x	x	x	x	x	x	x	x

= Unused; always read ‘x’

Figure 18-31. Identifier Register 1 — Standard Mapping

Table 18-32. IDR1 Register Field Descriptions

Field	Description
7-5 ID[2:0]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 18-31.
4 RTR	**Remote Transmission Request** — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)

[3] SPIDRH can be written repeatedly without any effect on SPTEF. SPTEF Flag is cleared only by writing to SPIDRL after reading SPISR with SPTEF == 1.

21.3.2.5 SPI Data Register (SPIDR = SPIDRH:SPIDRL)

Module Base +0x0004

	7	6	5	4	3	2	1	0
R	R15	R14	R13	R12	R11	R10	R9	R8
W	T15	T14	T13	T12	T11	T10	T9	T8
Reset	0	0	0	0	0	0	0	0

Figure 21-7. SPI Data Register High (SPIDRH)

Module Base +0x0005

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Figure 21-8. SPI Data Register Low (SPIDRL)

Read: Anytime; read data only valid when SPIF is set

Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 21-9).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 21-10).

24.4.6.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 24-10). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 24-4). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 24-50. Verify Backdoor Access Key Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x3_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 24-51. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 24.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

24.4.6.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

Address & Name		7	6	5	4	3	2	1	0
0x000D FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x000E FRSV3	R	0	0	0	0	0	0	0	0
	W								
0x000F FRSV4	R	0	0	0	0	0	0	0	0
	W								
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV5	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV6	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV7	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 26-4. FTMRG48K1 Register Summary (continued)

26.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field in normal mode. In special mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

26.4.6.13 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or EEPROM block.

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the

Table 26-57. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Flash block selection code [1:0]. See Table 26-34
001	Margin level setting.	

field margin level for the targeted block and then set the CCIF flag.

NOTE

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 26-58.

Table 26-58. Valid Set Field Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

1 Read margin to the erased state

2 Read margin to the programmed state

Table 26-59. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 26-27)
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied (See Table 26-34)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

26.4.6.14 Erase Verify EEPROM Section Command

The Erase Verify EEPROM Section command will verify that a section of code in the EEPROM is erased. The Erase Verify EEPROM Section command defines the starting point of the data to be verified and the number of words.

Table 26-60. Erase Verify EEPROM Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify EEPROM Section command, the Memory Controller will verify the selected section of EEPROM memory is erased. The CCIF flag will set after the Erase Verify EEPROM Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

The FPROT register, described in Section 27.3.2.9, can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0x3_8000 in the Flash memory (called the lower region), one growing downward from global address 0x3_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protection. 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 27-4.

Table 27-4. Flash Configuration Field

Global Address	Size (Bytes)	Description
0x3_FF00-0x3_FF07	8	Backdoor Comparison Key Refer to Section 27.4.6.11, “Verify Backdoor Access Key Command,” and Section 27.5.1, “Unsecuring the MCU using Backdoor Key Access”
0x3_FF08-0x3_FF0B[1]	4	Reserved
0x3_FF0C[1]	1	P-Flash Protection byte. Refer to Section 27.3.2.9, “P-Flash Protection Register (FPROT)”
0x3_FF0D[1]	1	EEPROM Protection byte. Refer to Section 27.3.2.10, “EEPROM Protection Register (EEPROT)”
0x3_FF0E[1]	1	Flash Nonvolatile byte Refer to Section 27.3.2.16, “Flash Option Register (FOPT)”
0x3_FF0F[1]	1	Flash Security byte Refer to Section 27.3.2.2, “Flash Security Register (FSEC)”

1 0x3FF08-0x3_FF0F form a Flash phrase and must be programmed in a single command write sequence. Each byte in the 0x3_FF08 - 0x3_FF0B reserved field should be programmed to 0xFF.

Table 27-41. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 27-27)
		Set if an invalid global address [17:0] is supplied see Table 27-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

27.4.6.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 27.4.6.4. The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 27-42. Program Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field in normal mode. In special mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

CAUTION

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

Table 29-7. 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 29-8 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Please refer to Section 29.4.4, “Flash Command Operations,” for more information.

Table 29-17. FPROT Field Descriptions (continued)

Field	Description
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x3_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 29-20. The FPLS bits can only be written to while the FPLDIS bit is set.

Table 29-18. P-Flash Protection Function

FPOPEN	FPHDIS	FPLDIS	Function[1]
1	1	1	No P-Flash Protection
1	1	0	Protected Low Range
1	0	1	Protected High Range
1	0	0	Protected High and Low Ranges
0	1	1	Full P-Flash Memory Protected
0	1	0	Unprotected Low Range
0	0	1	Unprotected High Range
0	0	0	Unprotected High and Low Ranges

1 For range sizes, refer to Table 29-19 and Table 29-20.

Table 29-19. P-Flash Protection Higher Address Range

FPHS[1:0]	Global Address Range	Protected Size
00	0x3_F800–0x3_FFFF	2 Kbytes
01	0x3_F000–0x3_FFFF	4 Kbytes
10	0x3_E000–0x3_FFFF	8 Kbytes
11	0x3_C000–0x3_FFFF	16 Kbytes

Table 29-20. P-Flash Protection Lower Address Range

FPLS[1:0]	Global Address Range	Protected Size
00	0x3_8000–0x3_83FF	1 Kbyte
01	0x3_8000–0x3_87FF	2 Kbytes
10	0x3_8000–0x3_8FFF	4 Kbytes
11	0x3_8000–0x3_9FFF	8 Kbytes

All possible P-Flash protection scenarios are shown in Figure 29-14 . Although the protection scheme is loaded from the Flash memory at global address 0x3_FF0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

30.4.6.15 Program EEPROM Command

The Program EEPROM operation programs one to four previously erased words in the EEPROM block. The Program EEPROM operation will confirm that the targeted location(s) were successfully programmed upon completion.

CAUTION

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

Table 30-62. Program EEPROM Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x11	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	
101	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program EEPROM command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program EEPROM command launch determines how many words will be programmed in the EEPROM block. The CCIF flag is set when the operation has completed.

Table 30-63. Program EEPROM Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] < 010 at command launch
		Set if CCOBIX[2:0] > 101 at command launch
		Set if command not available in current mode (see Table 30-27)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested group of words breaches the end of the EEPROM block
	FPVIOL	Set if the selected area of the EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

30.4.6.16 Erase EEPROM Sector Command

The Erase EEPROM Sector operation will erase all addresses in a sector of the EEPROM block.

Table 31-5. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x0_40B8 – 0x0_40BF	8	Reserved
0x0_40C0 – 0x0_40FF	64	Program Once Field Refer to Section 31.4.6.6, “Program Once Command”

1 Used to track firmware patch versions, see Section 31.4.2

Table 31-6. Memory Controller Resource Fields (NVMRES[1]=1)

Global Address	Size (Bytes)	Description
0x0_4000 – 0x040FF	256	P-Flash IFR (see Table 31-5)
0x0_4100 – 0x0_41FF	256	Reserved.
0x0_4200 – 0x0_57FF		Reserved
0x0_5800 – 0x0_5AFF	768	Reserved
0x0_5B00 – 0x0_5FFF	1,280	Reserved
0x0_6000 – 0x0_67FF	2,048	Reserved
0x0_6800 – 0x0_7FFF	6,144	Reserved

1 NVMRES - See Section 31.4.3 for NVMRES (NVM Resource) detail.

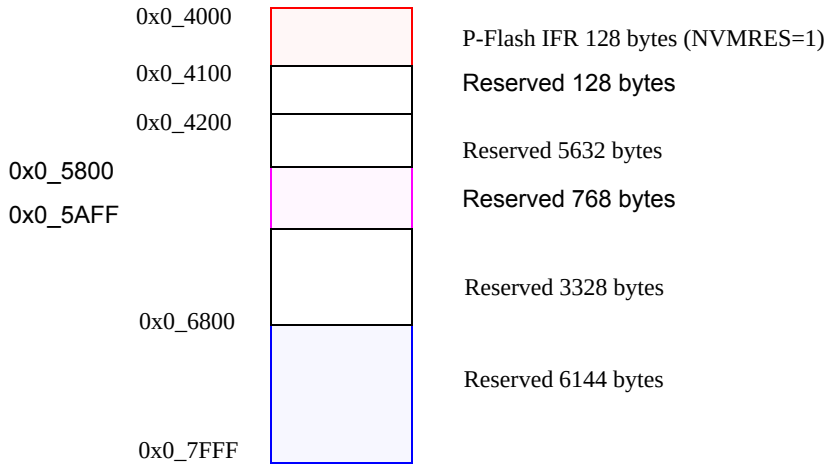

Figure 31-3. Memory Controller Resource Memory Map (NVMRES=1)

Table A-5. Thermal Package Characteristics[1]

Num	C	Rating	Symbol	S12GN32, S12GNA32, S12GN16, S12GNA16	S12G64, S12GA64, S12G48, S12GN48, S12GA64	S12G128, S12GA128, S12G96, S12GA96	S12G240, S12GA240, S12G192, S12GA192	Unit
20-pin TSSOP								
1	D	Thermal resistance single sided PCB, natural convection[2]	θ_{JA}	91				°C/W
2	D	Thermal resistance single sided PCB @ 200 ft/min[3]	θ_{JMA}	72				°C/W
3	D	Thermal resistance double sided PCB with 2 internal planes, natural convection[3]	θ_{JA}	58				°C/W
4	D	Thermal resistance double sided PCB with 2 internal planes @ 200 ft/min[3]	θ_{JMA}	51				°C/W
5	D	Junction to Board[4]	θ_{JB}	29				°C/W
6	D	Junction to Case[5]	θ_{JC}	20				°C/W
7	D	Junction to Package Top[6]	Ψ_{JT}	4				°C/W
32-pin LQFP								
8	D	Thermal resistance single sided PCB, natural convection[2]	θ_{JA}	81	84			°C/W
9	D	Thermal resistance single sided PCB @ 200 ft/min[3]	θ_{JMA}	68	70			°C/W
10	D	Thermal resistance double sided PCB with 2 internal planes, natural convection[3]	θ_{JA}	57	56			°C/W
11	D	Thermal resistance double sided PCB with 2 internal planes @ 200 ft/min[3]	θ_{JMA}	50	49			°C/W
12	D	Junction to Board[4]	θ_{JB}	35	32			°C/W
13	D	Junction to Case[5]	θ_{JC}	25	23			°C/W
14	D	Junction to Package Top[6]	Ψ_{JT}	8	6			°C/W
48-pin LQFP								
15	D	Thermal resistance single sided PCB, natural convection[2]	θ_{JA}	81	80	79	75	°C/W
16	D	Thermal resistance single sided PCB @ 200 ft/min[3]	θ_{JMA}	68	67	66	62	°C/W
17	D	Thermal resistance double sided PCB with 2 internal planes, natural convection[3]	θ_{JA}	57	56	56	51	°C/W
18	D	Thermal resistance double sided PCB with 2 internal planes @ 200 ft/min[3]	θ_{JMA}	50	50	49	45	°C/W
19	D	Junction to Board[4]	θ_{JB}	35	34	33	30	°C/W
20	D	Junction to Case[5]	θ_{JC}	25	24	21	19	°C/W
21	D	Junction to Package Top[6]	Ψ_{JT}	8	6	4	N/A	°C/W