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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K 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/s9s12g128f0mlh

1.8 Device Pinouts

1.8.1 S12GN16 and S12GN32

1.8.1.1 Pinout 20-Pin TSSOP

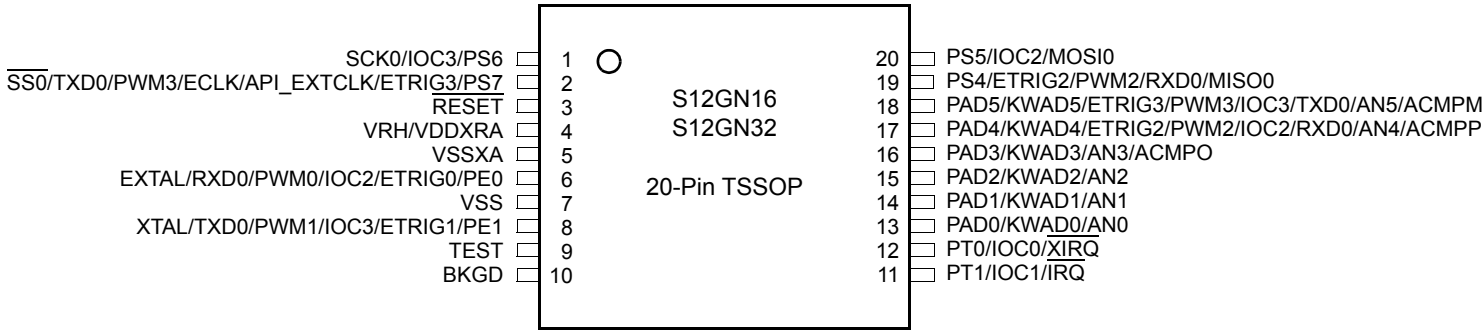


Figure 1-3. 20-Pin TSSOP Pinout for S12GN16 and S12GN32

Table 1-8. 20-Pin TSSOP Pinout for S12GN16 and S12GN32

	Function <----lowest-----PRIORITY-----highest---->								Power Supply	Internal Pull Resistor	
Package Pin	Pin	2nd Func.	3rd Func.	4th Func	5th Func	6th Func	7th Func	8th Func		CTRL	Reset State
1	PS6	IOC3	SCK0	—	—	—	—	—	V_{DDX}	PERS/PPSS	Up
2	PS7	ETRIG3	API_EXTCLK	ECLK	PWM3	TXD0	$\overline{SS0}$	—	V_{DDX}	PERS/PPSS	Up
3	RESET	—	—	—	—	—	—	—	V_{DDX}	PULLUP	
4	VDDXRA	VRH	—	—	—	—	—	—	—	—	—
5	VSSXA	—	—	—	—	—	—	—	—	—	—
6	PE0[1]	ETRIG0	PWM0	IOC2	RXD0	EXTAL	—	—	V_{DDX}	PUCR/PDPEE	Down
7	VSS	—	—	—	—	—	—	—	—	—	—
8	PE1[1]	ETRIG1	PWM1	IOC3	TXD0	XTAL	—	—		PUCR/PDPEE	Down
9	TEST	—	—	—	—	—	—	—	N.A.	$\overline{RESET}$ pin	Down
10	BKGD	MODC	—	—	—	—	—	—	V_{DDX}	Always on	Up
11	PT1	IOC1	$\overline{IRQ}$	—	—	—	—	—	V_{DDX}	PERT/PPST	Disabled
12	PT0	IOC0	$\overline{XIRQ}$	—	—	—	—	—	V_{DDX}	PERT/PPST	Disabled
13	PAD0	KWAD0	AN0	—	—	—	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled
14	PAD1	KWAD1	AN1	—	—	—	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled
15	PAD2	KWAD2	AN2	—	—	—	—	—	V_{DDA}	PER1AD/PPS1AD	Disabled

1.8.9.3 Pinout 100-Pin LQFP

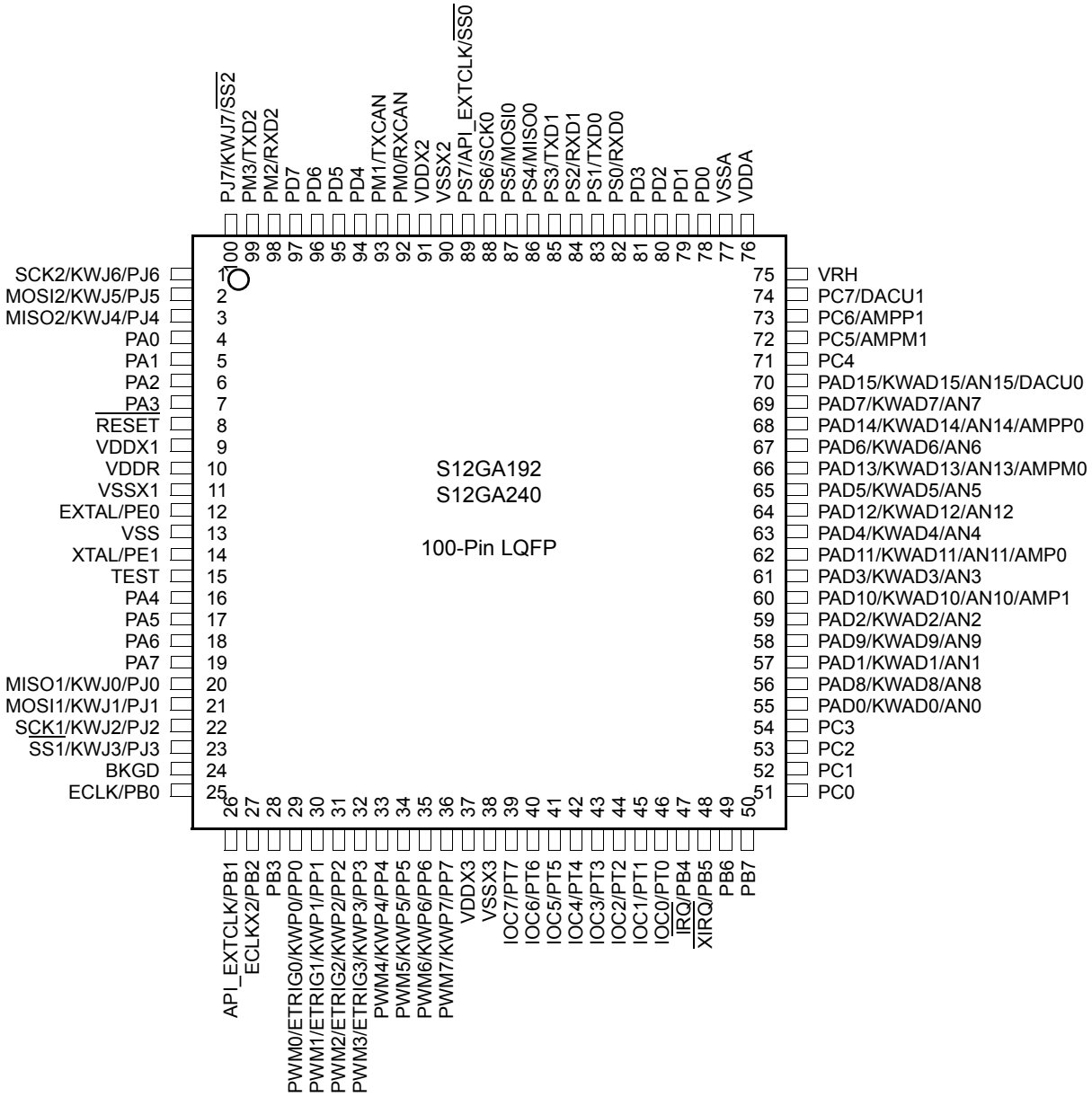


Figure 1-26. 100-Pin LQFP Pinout for S12GA192 and S12GA240

Table 1-34. Reset Sources and Vector Locations

Vector Address	Reset Source	CCR Mask	Local Enable
$FFFE	Power-On Reset (POR)	None	None
$FFFE	Low Voltage Reset (LVR)	None	None
$FFFE	External pin $\overline{\text{RESET}}$	None	None
$FFFE	Illegal Address Reset	None	None
$FFFC	Clock monitor reset	None	OSCE Bit in CPMUOSC register
$FFFA	COP watchdog reset	None	CR[2:0] in CPMUCOP register

1.12.2 Interrupt Vectors

Table 1-35 lists all interrupt sources and vectors in the default order of priority. The interrupt module (see Chapter 6, “Interrupt Module (S12SINTV1)”) provides an interrupt vector base register (IVBR) to relocate the vectors.

Table 1-35. Interrupt Vector Locations (Sheet 1 of 2)

Vector Address[1]	Interrupt Source	CCR Mask	Local Enable	Wake up from STOP	Wakeup from WAIT
Vector base + $F8	Unimplemented instruction trap	None	None	-	-
Vector base+ $F6	SWI	None	None	-	-
Vector base+ $F4	$\overline{\text{XIRQ}}$	X Bit	None	Yes	Yes
Vector base+ $F2	$\overline{\text{IRQ}}$	I bit	IRQCR (IRQEN)	Yes	Yes
Vector base+ $F0	RTI time-out interrupt	I bit	CPMUINT (RTIE)	10.6 Interrupts	
Vector base+ $EE	TIM timer channel 0	I bit	TIE (C0I)	No	Yes
Vector base + $EC	TIM timer channel 1	I bit	TIE (C1I)	No	Yes
Vector base+ $EA	TIM timer channel 2	I bit	TIE (C2I)	No	Yes
Vector base+ $E8	TIM timer channel 3	I bit	TIE (C3I)	No	Yes
Vector base+ $E6	TIM timer channel 4	I bit	TIE (C4I)	No	Yes
Vector base+ $E4	TIM timer channel 5	I bit	TIE (C5I)	No	Yes
Vector base + $E2	TIM timer channel 6	I bit	TIE (C6I)	No	Yes
Vector base+ $E0	TIM timer channel 7	I bit	TIE (C7I)	No	Yes
Vector base+ $DE	TIM timer overflow	I bit	TSCR2 (TOI)	No	Yes
Vector base+ $DC	TIM Pulse accumulator A overflow[2]	I bit	PACTL (PAOVI)	No	Yes
Vector base + $DA	TIM Pulse accumulator input edge[3]	I bit	PACTL (PAI)	No	Yes
Vector base + $D8	SPI0	I bit	SPI0CR1 (SPIE, SPTIE)	No	Yes
Vector base+ $D6	SCI0	I bit	SCI0CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D4	SCI1	I bit	SCI1CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes

2.6 Initialization/Application Information

2.6.1 Initialization

After a system reset, software should:

1. Read the PKGCR and write to it with its preset content to engage the write lock on PKGCR[PKGCR2:PKGCR0] bits protecting the device from inadvertent changes to the pin layout in normal applications.
2. Write to PRR0 in 20 TSSOP to define the module routing and to PKGCR[APICLKS7] bit in any package for API_EXTCLK.

GA240 / GA192 devices only:

3. In applications using the analog functions on port C pins shared with AMPM1, AMPP1 or DACU1 the input buffers should be disabled early after reset by enabling the related mode of the DAC1 module. This shortens the time of potentially increased power consumption caused by the digital input buffers operating in the linear region.

2.6.2 Port Data and Data Direction Register writes

It is not recommended to write PORTx/PTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

2.6.3 Enabling IRQ edge-sensitive mode

To avoid unintended IRQ interrupts resulting from writing to IRQCR while the IRQ pin is driven to active level ($\overline{IRQ}$=0) the following initialization sequence is recommended:

1. Mask I-bit
2. Set IRQCR[IRQEN]
3. Set IRQCR[IRQE]
4. Clear I-bit

2.6.4 ADC External Triggers ETRIG3-0

The ADC external trigger inputs ETRIG3-0 allow the synchronization of conversions to external trigger events if selected as trigger source (for details refer to ATDCTL1[ETRIGSEL] and ATDCTL1[ETRIGCH] configuration bits in ADC section). These signals are related to PWM channels 3-0 to support periodic trigger applications with the ADC. Other pin functions can also be used as triggers.

If a PWM channel is routed to an alternative pin, the ETRIG input function will follow the relocation accordingly.

If the related PWM channel is enabled, the PWM signal as seen on the pin will drive the ETRIG input. If another signal of higher priority takes control of the pin or if on a port AD pin the input buffer is disabled,

Table 8-10. ABCM Encoding

ABCM	Description
00	Match0 mapped to comparator A match: Match1 mapped to comparator B match.
01	Match 0 mapped to comparator A/B inside range: Match1 disabled.
10	Match 0 mapped to comparator A/B outside range: Match1 disabled.
11	Reserved[1]

[1] Currently defaults to Comparator A, Comparator B disabled

8.3.2.5 Debug Trace Buffer Register (DBGTBH:DBGTBL)

Address: 0x0024, 0x0025

	15	14	13	12	11	10	9	8	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	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Other Resets	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Figure 8-7. Debug Trace Buffer Register (DBGTB)

Read: Only when unlocked AND unsecured AND not armed AND TSOURCE set.

Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents.

Table 8-11. DBGTB Field Descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the 20-bit wide data lines of the Trace Buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is set the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. Similarly reads while the debugger is armed or with the TSOURCE bit clear, return 0 and do not affect the trace buffer pointer. The POR state is undefined. Other resets do not affect the trace buffer contents.

8.3.2.6 Debug Count Register (DBGCNT)

Address: 0x0026

	7	6	5	4	3	2	1	0
R	TBF	0	CNT					
W								
Reset	—	—	—	—	—	—	—	—
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-8. Debug Count Register (DBGCNT)

Table 8-39. PCH Field Descriptions (continued)

Bit	Description
0 PC16	**Program Counter bit 16**— In Normal and Loop1 mode this bit corresponds to program counter bit 16.

8.4.5.4 Trace Buffer Organization (Compressed Pure PC mode)

Table 8-40. Trace Buffer Organization Example (Compressed PurePC mode)

Mode	Line Number	2-bits	6-bits	6-bits	6-bits
		Field 3	Field 2	Field 1	Field 0
Compressed Pure PC Mode	Line 1	00	PC1 (Initial 18-bit PC Base Address)		
	Line 2	11	PC4	PC3	PC2
	Line 3	01	0	0	PC5
	Line 4	00	PC6 (New 18-bit PC Base Address)		
	Line 5	10	0	PC8	PC7
	Line 6	00	PC9 (New 18-bit PC Base Address)		

NOTE

Configured for end aligned triggering in compressed PurePC mode, then after rollover it is possible that the oldest base address is overwritten. In this case all entries between the pointer and the next base address have lost their base address following rollover. For example in Table 8-40 if one line of rollover has occurred, Line 1, PC1, is overwritten with a new entry. Thus the entries on Lines 2 and 3 have lost their base address. For reconstruction of program flow the first base address following the pointer must be used, in the example, Line 4. The pointer points to the oldest entry, Line 2.

Field3 Bits in Compressed Pure PC Modes

Table 8-41. Compressed Pure PC Mode Field 3 Information Bit Encoding

INF1	INF0	TRACE BUFFER ROW CONTENT
0	0	Base PC address TB[17:0] contains a full PC[17:0] value
0	1	Trace Buffer[5:0] contain incremental PC relative to base address zero value
1	0	Trace Buffer[11:0] contain next 2 incremental PCs relative to base address zero value
1	1	Trace Buffer[17:0] contain next 3 incremental PCs relative to base address zero value

Each time that PC[17:6] differs from the previous base PC[17:6], then a new base address is stored. The base address zero value is the lowest address in the 64 address range

The first line of the trace buffer always gets a base PC address, this applies also on rollover.

Chapter 9
Security (S12XS9SECV2)

Table 9-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
02.00	27 Aug 2004		reviewed and updated for S12XD architecture
02.01	21 Feb 2007		added S12XE, S12XF and S12XS architectures
02.02	19 Apr 2007		corrected statement about Backdoor key access via BDM on XE, XF, XS

9.1 Introduction

This specification describes the function of the security mechanism in the MC9S12G-Family (9SEC).

NOTE

No security feature is absolutely secure. However, NXP's strategy is to make reading or copying the FLASH and/or EEPROM difficult for unauthorized users.

9.1.1 Features

The user must be reminded that part of the security must lie with the application code. An extreme example would be application code that dumps the contents of the internal memory. This would defeat the purpose of security. At the same time, the user may also wish to put a backdoor in the application program. An example of this is the user downloads a security key through the SCI, which allows access to a programming routine that updates parameters stored in another section of the Flash memory.

The security features of the MC9S12G-Family (in secure mode) are:

- Protect the content of non-volatile memories (Flash, EEPROM)
- Execution of NVM commands is restricted
- Disable access to internal memory via background debug module (BDM)

9.1.2 Modes of Operation

Table 9-2 gives an overview over availability of security relevant features in unsecure and secure modes.

Table 9-2. Feature Availability in Unsecure and Secure Modes on S12XS

	Unsecure Mode						Secure Mode					
	NS	SS	NX	ES	EX	ST	NS	SS	NX	ES	EX	ST
Flash Array Access	?	?					?	?				

18.4.5.5 MSCAN Sleep Mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

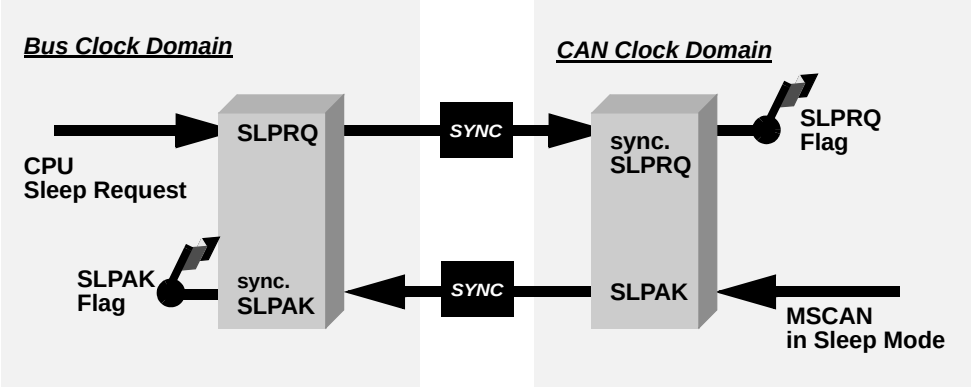


Figure 18-46. Sleep Request / Acknowledge Cycle

NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 18-46). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

18.5 Initialization/Application Information

18.5.1 MSCAN initialization

The procedure to initially start up the MSCAN module out of reset is as follows:

1. Assert CANE
2. Write to the configuration registers in initialization mode
3. Clear INITRQ to leave initialization mode

If the configuration of registers which are only writable in initialization mode shall be changed:

1. Bring the module into sleep mode by setting SLPRQ and awaiting SLPAK to assert after the CAN bus becomes idle.
2. Enter initialization mode: assert INITRQ and await INITAK
3. Write to the configuration registers in initialization mode
4. Clear INITRQ to leave initialization mode and continue

18.5.2 Bus-Off Recovery

The bus-off recovery is user configurable. The bus-off state can either be left automatically or on user request.

For reasons of backwards compatibility, the MSCAN defaults to automatic recovery after reset. In this case, the MSCAN will become error active again after counting 128 occurrences of 11 consecutive recessive bits on the CAN bus (see the Bosch CAN 2.0 A/B specification for details).

If the MSCAN is configured for user request (BORM set in MSCAN Control Register 1 (CANCTL1)), the recovery from bus-off starts after both independent events have become true:

- 128 occurrences of 11 consecutive recessive bits on the CAN bus have been monitored
- BOHOLD in MSCAN Miscellaneous Register (CANMISC) has been cleared by the user

These two events may occur in any order.

Chapter 20
Serial Communication Interface (S12SCIV5)

Table 20-1. Revision History

Version Number	Revision Date	Effective Date	Author	Description of Changes
05.03	12/25/2008			remove redundancy comments in Figure1-2
05.04	08/05/2009			fix typo, SCIBDL reset value be 0x04, not 0x00
05.05	06/03/2010			fix typo, Table 20-4,SCICR1 Even parity should be PT=0 fix typo, on page 20-674,should be BKDIF,not BLDIF

20.1 Introduction

This block guide provides an overview of the serial communication interface (SCI) module.

The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

20.1.1 Glossary

IR: InfraRed

IrDA: Infrared Design Associate

IRQ: Interrupt Request

LIN: Local Interconnect Network

LSB: Least Significant Bit

MSB: Most Significant Bit

NRZ: Non-Return-to-Zero

RZI: Return-to-Zero-Inverted

RXD: Receive Pin

SCI : Serial Communication Interface

TXD: Transmit Pin

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Re synchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

20.4.6.5.1 Slow Data Tolerance

Figure 20-28 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

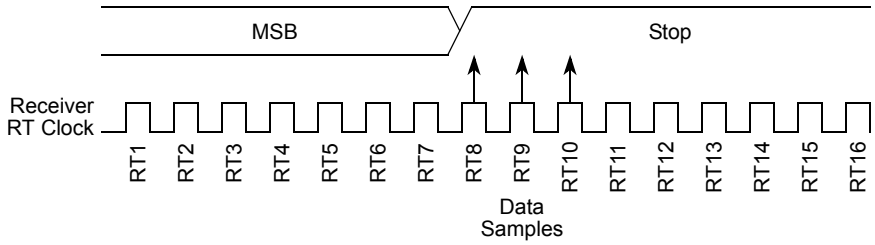


Figure 20-28. Slow Data

Let’s take RTr as receiver RT clock and RTt as transmitter RT clock.

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles +7 RTr cycles = 151 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 20-28, the receiver counts 151 RTr cycles at the point when the count of the transmitting device is 9 bit times x 16 RTt cycles = 144 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

((151 – 144) / 151) x 100 = 4.63%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 7 RTr cycles = 167 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 20-28, the receiver counts 167 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

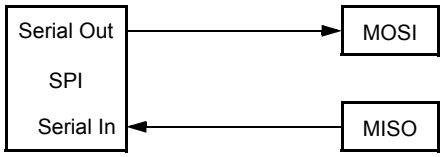
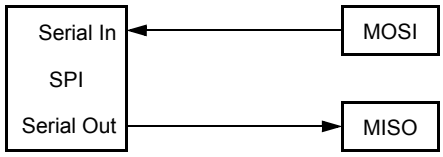
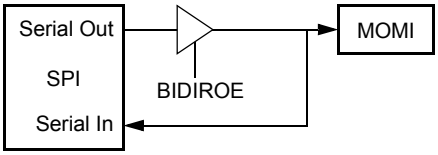
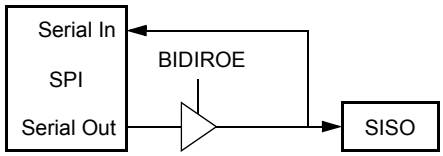
The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

((167 – 160) / 167) X 100 = 4.19%

20.4.6.5.2 Fast Data Tolerance

Figure 20-29 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

Table 21-10. Normal Mode and Bidirectional Mode

When SPE = 1	Master Mode MSTR = 1	Slave Mode MSTR = 0
Normal Mode SPC0 = 0		
Bidirectional Mode SPC0 = 1		

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

NOTE

In bidirectional master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode. In this case MISO becomes occupied by the SPI and MOSI is not used. This must be considered, if the MISO pin is used for another purpose.

21.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

21.4.6.1 Mode Fault Error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case

Chapter 25
32 KByte Flash Module (S12FTMRG32K1V1)

Table 25-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	17 Jun 2010	25.4.6.1/25-846 25.4.6.2/25-847 25.4.6.3/25-847 25.4.6.14/25-857	Clarify Erase Verify Commands Descriptions related to the bits MGSTAT[1:0] of the register FSTAT.
V01.05	20 aug 2010	25.4.6.2/25-847 25.4.6.12/25-854 25.4.6.13/25-856	Updated description of the commands RD1BLK, MLOADU and MLOADF
Rev.1.27	31 Jan 2011	25.3.2.9/25-829	Updated description of protection on Section 25.3.2.9

25.1 Introduction

The FTMRG32K1 module implements the following:

- 32Kbytes of P-Flash (Program Flash) memory
- 1 Kbytes of EEPROM memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

CAUTION

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

The Flash memory may be read as bytes and aligned words. Read access time is one bus cycle for bytes and aligned words. For misaligned words access, the CPU has to perform twice the byte read access command. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

During the reset sequence, fields DPOPEN and DPS of the EEPROT register are loaded with the contents of the EEPROM protection byte in the Flash configuration field at global address 0x3_FF0D located in P-Flash memory (see Table 25-4) as indicated by reset condition F in Table 25-23. To change the EEPROM protection that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 25-22. EEPROT Field Descriptions

Field	Description
7 DPOPEN	EEPROM Protection Control 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
4–0 DPS[4:0]	**EEPROM Protection Size** — The DPS[4:0] bits determine the size of the protected area in the EEPROM memory as shown inTable 25-23 .

Table 25-23. EEPROM Protection Address Range

DPS[4:0]	Global Address Range	Protected Size
00000	0x0_0400 – 0x0_041F	32 bytes
00001	0x0_0400 – 0x0_043F	64 bytes
00010	0x0_0400 – 0x0_045F	96 bytes
00011	0x0_0400 – 0x0_047F	128 bytes
00100	0x0_0400 – 0x0_049F	160 bytes
00101	0x0_0400 – 0x0_04BF	192 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
11111 - to - 11111	0x0_0400 – 0x0_07FF	1,024 bytes

Table 29-25. FOPT Field Descriptions

Field	Description
7–0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits.

29.3.2.17 Flash Reserved5 Register (FRSV5)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0011

	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 29-23. Flash Reserved5 Register (FRSV5)

All bits in the FRSV5 register read 0 and are not writable.

29.3.2.18 Flash Reserved6 Register (FRSV6)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

	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 29-24. Flash Reserved6 Register (FRSV6)

All bits in the FRSV6 register read 0 and are not writable.

29.3.2.19 Flash Reserved7 Register (FRSV7)

This Flash register is reserved for factory testing.

P-Flash memory (see Table 31-4) as indicated by reset condition F in Table 31-23. To change the EEPROM protection that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 31-22. EEPROT Field Descriptions

Field	Description
7 DPOPEN	EEPROM Protection Control 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
6–0 DPS[6:0]	**EEPROM Protection Size** — The DPS[6:0] bits determine the size of the protected area in the EEPROM memory, this size increase in step of 32 bytes, as shown in Table 31-23 .

Table 31-23. EEPROM Protection Address Range

DPS[6:0]	Global Address Range	Protected Size
0000000	0x0_0400 – 0x0_041F	32 bytes
0000001	0x0_0400 – 0x0_043F	64 bytes
0000010	0x0_0400 – 0x0_045F	96 bytes
0000011	0x0_0400 – 0x0_047F	128 bytes
0000100	0x0_0400 – 0x0_049F	160 bytes
0000101	0x0_0400 – 0x0_04BF	192 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
1111111	0x0_0400 – 0x0_13FF	4,096 bytes

Table A-35. NVM Timing Characteristics)

S12GN16, S12GNA16, S12GN32, S12GNA32									
Num	**Command**	**f_{NVMOP} cycle**	**f_{NVMBUS} cycle**	**Symbol**	**Min[1]**	**Typ[2]**	**Max[3]**	**Lfmax[4]**	**Unit**
1	Erase Verify All Blocks[5,6]	0	9233	t_{RD1ALL}	0.37	0.37	0.74	18.47	ms
2	Erase Verify Block (Pflash)[5]	0	8737	t_{RD1BLK_P}	0.35	0.35	0.7	17.47	ms
3	Erase Verify Block (EEPROM)[6]	0	1000	t_{RD1BLK_D}	0.04	0.04	0.08	2	ms
4	Erase Verify P-Flash Section	0	486	t_{RD1SEC}	19.44	19.44	38.88	972	ms
5	Read Once	0	445	t_{RDONCE}	17.8	17.8	17.8	445	µs
6	Program P-Flash (4 Word)	164	2935	t_{PGM_4}	0.27	0.28	0.63	11.95	ms
7	Program Once	164	2888	$t_{PGMONCE}$	0.27	0.28	0.28	3.09	ms
8	Erase All Blocks[5,6]	100066	9569	t_{ERSALL}	95.68	100.45	100.83	144.22	ms
9	Erase Flash Block (Pflash)[5]	100060	8975	t_{ERSBLK_P}	95.65	100.42	100.78	143.03	ms
10	Erase Flash Block (EEPROM)[6]	100060	1296	t_{ERSBLK_D}	95.35	100.11	100.16	127.67	ms
11	Erase P-Flash Sector	20015	875	t_{ERSPG}	19.1	20.05	20.09	26.77	ms
12	Unsecure Flash	100066	9647	t_{UNSECU}	95.69	100.45	100.84	144.38	ms
13	Verify Backdoor Access Key	0	481	t_{VFYKEY}	19.24	19.24	19.24	481	µs
14	Set User Margin Level	0	404	t_{MLOADU}	16.16	16.16	16.16	404	µs
15	Set Factory Margin Level	0	413	t_{MLOADF}	16.52	16.52	16.52	413	µs
16	Erase Verify EEPROM Section	0	546	$t_{DRD1SEC}$	0.02	0.02	0.04	1.09	ms
17	Program EEPROM (1 Word)	68	1565	t_{DPGM_1}	0.13	0.13	0.32	6.35	ms
18	Program EEPROM (2 Word)	136	2512	t_{DPGM_2}	0.23	0.24	0.54	10.22	ms
19	Program EEPROM (3 Word)	204	3459	t_{DPGM_3}	0.33	0.34	0.76	14.09	ms
20	Program EEPROM (4 Word)	272	4406	t_{DPGM_4}	0.44	0.45	0.98	17.96	ms
21	Erase EEPROM Sector	5015	753	t_{DERSPG}	4.81	5.05	20.57	37.88	ms

1 Minimum times are based on maximum f_{NVMOP} and maximum f_{NVMBUS}

2 Typical times are based on typical f_{NVMOP} and typical f_{NVMBUS}

3 Maximum times are based on typical f_{NVMOP} and typical f_{NVMBUS} plus aging

4 Lowest-frequency max times are based on minimum f_{NVMOP} and minimum f_{NVMBUS} plus aging

5 Affected by Pflash size

6 Affected by EEPROM size

Table A-53. Measurement Conditions

Description	Symbol	Value	Unit
NVM activity		none	

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. DATUMS A, B AND D TO BE DETERMINED AT DATUM PLANE H.
4. DIMENSIONS TO BE DETERMINED AT SEATING PLANE C.
5. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE UPPER LIMIT BY MORE THAN 0.08 mm AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD SHALL NOT BE LESS THAN 0.07 mm.
6. THIS DIMENSION DOES NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. THIS DIMENSION IS MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
7. EXACT SHAPE OF EACH CORNER IS OPTIONAL.
8. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.1 mm AND 0.25 mm FROM THE LEAD TIP.

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE: 64LD LQFP, 10 X 10 X 1.4 PKG, 0.5 PITCH, CASE OUTLINE	DOCUMENT NO: 98ASS23234W	REV: E
	CASE NUMBER: 840F–02	11 AUG 2006
	STANDARD: JEDEC MS–026 BCD	