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	HCS12X
Core Size	16-Bit
Speed	40MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	88
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	8K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xhy256f0vlm

1.3.5 Internal Phase-Locked Loop (IPLL)

- Phase-locked-loop clock frequency multiplier
  - No external components required
  - Reference divider and multiplier allow large variety of clock rates
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
    - -

1.3.6 Clocks and reset generation(CRG)

- COP watchdog
- Real time interrupt
- Clock monitor
- Fast wake up from STOP in self clock mode

1.3.7 System Integrity Support

- Power-on reset (POR)
- System reset generation
- Illegal address detection with reset
- Low-voltage detection with interrupt or reset
- Real time interrupt (RTI)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Initialized out of reset using option bits located in flash memory
- Clock monitor supervising the correct function of the oscillator
- Temperature sensor

1.3.8 Timer (TIM0)

- 8x 16-bit channels for input capture
- 8x 16-bit channels for output compare
- 16-bit free-running counter with 8-bit precision prescaler
- 1 x 16-bit pulse accumulator

1.3.9 Timer (TIM1)

- 8x 16-bit channels for input capture
- 8x 16-bit channels for output compare

Table 1-7. Pin-Out Summary[1]

Package Pin		Function								Power Supply	Internal Pull Resistor		Description
LQFP 100	LQFP 64	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.	6th Func.	7th Func.	8th Func.		CTRL	Reset State	
37	-	PR2	IOC1_6	KWR2						VDDX	PERR/PPSR	Down	Port R I/O, timer1 Channel, Key wakeup
38	-	PR3	IOC1_7	KWR3						VDDX	PERR/PPSR	Down	Port R I/O, timer1 Channel, Key wakeup
39	33	PS0	PWM6	RXD0						VDDX	PERS/PPSS	Up	Port S I/O, RXD of SCI0, PWM channel6
40	34	PS1	PWM7	TXD0						VDDX	PERS/PPSS	Up	Port S I/O, TXD of SCI0, PWM channel 7
41	35	VSSX2											
42	36	VDDX2											
43	37	PS2	PWM4	RXCAN0	KWS2					VDDX	PERS/PPSS	Up	Port S I/O, PWM channel 4,RX of CAN0 , Key wakeup
44	38	PS3	PWM5	TXCAN0	KWS3					VDDX	PERS/PPSS	Up	Port S I/O,PWM channel 5, TX of CAN0 , Key wakeup
45	39	PR0	IOC0_6	RXCAN1	KWR0					VDDX	PERR/PPSR	Down	Port R I/O, timer0 Channel,RX of CAN1,Key wakeup
46	40	PR1	IOC0_7	TXCAN1	KWR1					VDDX	PERR/PPSR	Down	Port R I/O, timer0 Channel,TX of CAN1 ,Key wakeup
47	41	PS4	PWM0	SCL	MISO					VDDX	PERS/PPSS	Up	Port S I/O, MISO of SPI, SCL of IIC, PWM channel 0
48	42	PS5	PWM1	MOSI	KWS5					VDDX	PERS/PPSS	Up	Port S I/O, MOSI of SPI, PWM channel 1, key wakeup

1.7.3 Detailed Signal Descriptions

1.7.3.1 EXTAL, XTAL — Oscillator Pins

EXTAL and XTAL are the crystal driver and external clock pins. On reset all the device clocks are derived from the internal reference clock. XTAL is the oscillator output.

1.7.3.2 $\overline{\text{RESET}}$ — External Reset Pin

The $\overline{\text{RESET}}$ pin is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state, and an output when an internal MCU function causes a reset. The $\overline{\text{RESET}}$ pin has an internal pull-up device.

1.7.3.3 TEST — Test Pin

This input only pin is reserved for factory test. This pin has an internal pull-down device.

NOTE

The TEST pin must be tied to V_{SSA} in all applications.

1.7.3.4 BKGD / MODC — Background Debug and Mode Pin

The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has an internal pull-up device.

1.7.3.5 PAD[7:0] / AN[7:0] / KWAD[7:0]— Port AD Input Pins of ATD [7:0]

PAD[7:0] are a general-purpose input or output pins and analog inputs AN[7:0] of the analog-to-digital converter ATD. They can be configured as keypad wakeup inputs.

1.7.3.6 PA[7:4] / FP[36:33]— Port A I/O Pins [7:4]

PA[7:4] are a general-purpose input or output pins. They can be configured as frontplane segment driver outputs FP[36:33].

1.7.3.7 PA[3:2] / API_EXTCLK / XCLKS / FP[32:31]— Port A I/O Pins [3:2]

PA[3:2] are a general-purpose input or output pins. They can be configured as frontplane segment driver outputs FP[32:31]. PA3 can also be configure as API_EXTCLK.The XCLKS is an input signal which controls whether a crystal in combination with the internal loop controlled Pierce oscillator is used or whether full swing Pierce oscillator/external clock circuitry is used (refer to Section 1.16, “Oscillator Configuration). An internal pull-down is enabled during reset.

2.3.24 Port S Data Direction Register (DDRS)

Address 0x024A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
Reset	0	0	0	0	0	0	0	0

Figure 2-22. Port S Data Direction Register (DDRS)

1 Read: Anytime.
Write: Anytime.

Table 2-19. DDRS Register Field Descriptions

Field	Description
7 DDRS	**Port S data direction**— This register controls the data direction of pin 7.This register configures pin as either input or output. If SPI is routing to PS and SPI is enabled, the SPI determines the pin direction Else If IIC is routing to PS and IIC is enabled, the IIC determines the pin direction, it will force as open-drain output Else if PWM3 is routing to PS and PWM3 is enabled it will force as output. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6 DDRS	**Port S data direction**— This register controls the data direction of pin 6.This register configures pin as either input or output. If SPI is routing to PS and SPI is enabled, the SPI determines the pin direction Else if PWM2 is routing to PS and PWM2 is enabled it will force as output. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
5 DDRS	**Port S data direction**— This register controls the data direction of pin 5.This register configures pin as either input or output. If SPI is routing to PS and SPI is enabled, the SPI determines the pin direction Else if PWM1 is routing to PS and PWM1 is enabled it will force as output. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
4 DDRS	**Port S data direction**— This register controls the data direction of pin 4.This register configures pin as either input or output. If SPI is routing to PS and SPI is enabled, the SPI determines the pin direction Else If IIC is routing to PS and IIC is enabled, it will force as open-drain output Else if PWM0 is routing to PS and PWM0 is enabled it will force as output. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

3.3.2.3 Direct Page Register (DIRECT)

Address: 0x0011

	7	6	5	4	3	2	1	0
R W	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
Reset	0	0	0	0	0	0	0	0

Figure 3-8. Direct Register (DIRECT)

Read: Anytime

Write: anytime in special modes, one time only in other modes.

This register determines the position of the 256B direct page within the memory map.It is valid for both global and local mapping scheme.

Table 3-5. DIRECT Field Descriptions

Field	Description
7–0 DP[15:8]	**Direct Page Index Bits 15–8** — These bits are used by the CPU when performing accesses using the direct addressing mode. The bits from this register form bits [15:8] of the address (see Figure 3-9).

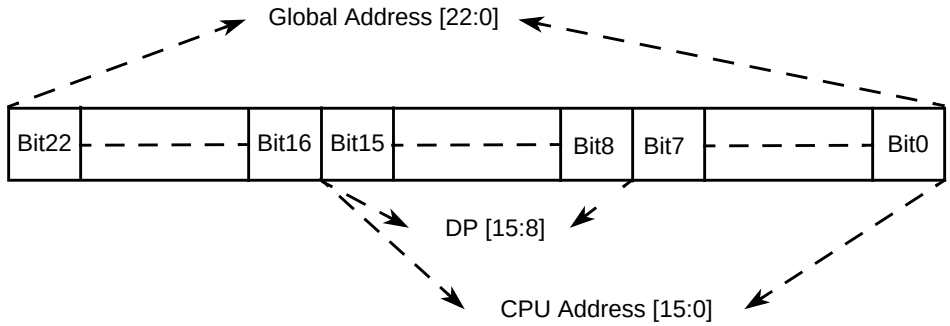

Figure 3-9. DIRECT Address Mapping

Bits [22:16] of the global address will be formed by the GPAGE[6:0] bits in case the CPU executes a global instruction in direct addressing mode or by the appropriate local address to the global address expansion (refer to Section 3.4.2.1.1, “Expansion of the Local Address Map).

Example 3-2. This example demonstrates usage of the Direct Addressing Mode

```
MOVB    #0x80,DIRECT    ;Set DIRECT register to 0x80. Write once only.
                        ;Global data accesses to the range 0xXX_80XX can be direct.
                        ;Logical data accesses to the range 0x80XX are direct.

LDY     <00             ;Load the Y index register from 0x8000 (direct access).
                        ;< operator forces direct access on some assemblers but in
                        ;many cases assemblers are "direct page aware" and can
                        ;automatically select direct mode.
```

Read: Anytime

Write: Anytime

These eight index bits are used to page 4KB blocks into the RAM page window located in the local (CPU or BDM) memory map from address 0x1000 to address 0x1FFF (see Figure 3-14). This supports accessing up to 1022KB of RAM (in the Global map) within the 64KB Local map. The RAM page index register is effectively used to construct paged RAM addresses in the Local map format.

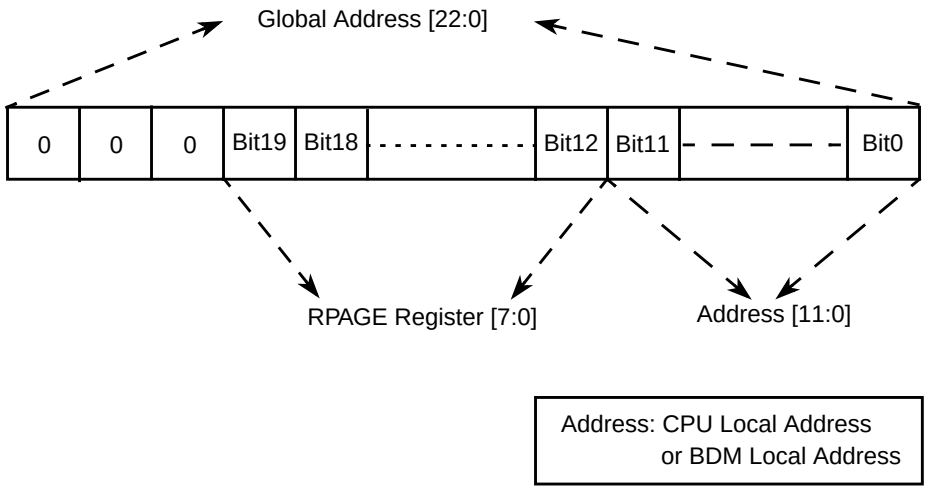

Figure 3-14. RPAGE Address Mapping

NOTE

Because RAM page 0 has the same global address as the register space, it is possible to write to registers through the RAM space when RPAGE = 0x00.

Table 3-8. RPAGE Field Descriptions

Field	Description
7–0 RP[7:0]	**RAM Page Index Bits 7–0** — These page index bits are used to select which of the 256 RAM array pages is to be accessed in the RAM Page Window.

The reset value of 0xFD ensures that there is a linear RAM space available between addresses 0x1000 and 0x3FFF out of reset.

The fixed 4K page from 0x2000–0x2FFF of RAM is equivalent to page 254 (page number 0xFE).

The fixed 4K page from 0x3000–0x3FFF of RAM is equivalent to page 255 (page number 0xFF).

NOTE

The page 0xFD (reset value) contains unimplemented area in the range not occupied by RAM if RAMSIZE is less than 12KB (Refer to Section 3.4.2.3, “Implemented Memory Map).

6.4.4.1 Final State

On entering Final State a trigger may be issued to the trace buffer according to the trace position control as defined by the TALIGN field (see Section 6.3.2.3"). If TSOURCE in the trace control register DBGTCR is cleared then the trace buffer is disabled and the transition to Final State can only generate a breakpoint request. In this case or upon completion of a tracing session when tracing is enabled, the ARM bit in the DBGC1 register is cleared, returning the module to the disarmed state0. If tracing is enabled, a breakpoint request can occur at the end of the tracing session. If neither tracing nor breakpoints are enabled then when the final state is reached it returns automatically to state0 and the debug module is disarmed.

6.4.5 Trace Buffer Operation

The trace buffer is a 64 lines deep by 64-bits wide RAM array. The S12XDBG module stores trace information in the RAM array in a circular buffer format. The RAM array can be accessed through a register window (DBGTBH:DBGTBL) using 16-bit wide word accesses. After each complete 64-bit trace buffer line is read, an internal pointer into the RAM is incremented so that the next read will receive fresh information. Data is stored in the format shown in Table 6-40. After each store the counter register bits DBGCNT[6:0] are incremented. Tracing of CPU12X activity is disabled when the BDM is active. Reading the trace buffer whilst the DBG is armed returns invalid data and the trace buffer pointer is not incremented.

6.4.5.1 Trace Trigger Alignment

Using the TALIGN bits (see Section 6.3.2.3") it is possible to align the trigger with the end, the middle, or the beginning of a tracing session.

If End or Mid tracing is selected, tracing begins when the ARM bit in DBGC1 is set and State1 is entered. The transition to Final State if End is selected signals the end of the tracing session. The transition to Final State if Mid is selected signals that another 32 lines will be traced before ending the tracing session. Tracing with Begin-Trigger starts at the opcode of the trigger.

6.4.5.1.1 Storing with Begin-Trigger

Storing with Begin-Trigger, data is not stored in the Trace Buffer until the Final State is entered. Once the trigger condition is met the S12XDBG module will remain armed until 64 lines are stored in the Trace Buffer. If the trigger is at the address of the change-of-flow instruction the change of flow associated with the trigger will be stored in the Trace Buffer. Using Begin-trigger together with tagging, if the tagged instruction is about to be executed then the trace is started. Upon completion of the tracing session the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

6.4.5.1.2 Storing with Mid-Trigger

Storing with Mid-Trigger, data is stored in the Trace Buffer as soon as the S12XDBG module is armed. When the trigger condition is met, another 32 lines will be traced before ending the tracing session, irrespective of the number of lines stored before the trigger occurred, then the S12XDBG module is disarmed and no more data is stored. Using Mid-trigger with tagging, if the tagged instruction is about to

Table 6-42. CXINF Field Descriptions (continued)

Field	Description
5 CRW	**Read Write Indicator** — This bit indicates if the corresponding stored address corresponds to a read or write access. This bit only contains valid information when tracing CPU12X activity in Detail Mode. 0 Write Access 1 Read Access

6.4.5.4 Reading Data from Trace Buffer

The data stored in the Trace Buffer can be read using either the background debug module (BDM) module or the CPU12X provided the S12XDBG module is not armed, is configured for tracing and the system not secured. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by an aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads, any byte or misaligned reads return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT the number of valid 64-bit lines can be determined. DBGCNT will not decrement as data is read.

Whilst reading an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry, thus if no overflow has occurred, the pointer points to line0, otherwise it points to the line with the oldest entry. The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.

The least significant word of each 64-bit wide array line is read out first. This corresponds to the bytes 1 and 0 of Table 6-40. The bytes containing invalid information (shaded in Table 6-40) are also read out.

Reading the Trace Buffer while the S12XDBG module is armed will return invalid data and no shifting of the RAM pointer will occur.

6.4.5.5 Trace Buffer Reset State

The Trace Buffer contents are not initialized by a system reset. Thus should a system reset occur, the trace session information from immediately before the reset occurred can be read out. The DBGCNT bits are not cleared by a system reset. Thus should a reset occur, the number of valid lines in the trace buffer is indicated by DBGCNT. The internal pointer to the current trace buffer address is initialized by unlocking the trace buffer thus points to the oldest valid data even if a reset occurred during the tracing session. Generally debugging occurrences of system resets is best handled using mid or end trigger alignment since the reset may occur before the trace trigger, which in the begin trigger alignment case means no information would be stored in the trace buffer.

NOTE

An external pin RESET that occurs simultaneous to a trace buffer entry can, in very seldom cases, lead to either that entry being corrupted or the first entry of the session being corrupted. In such cases the other contents of the trace buffer still contain valid tracing information. The case occurs when the reset assertion coincides with the trace buffer entry clock edge.

9.3.1 Module Memory Map

A summary of the registers associated with the VREG_3V3 sub-block is shown in Table 9-3. Detailed descriptions of the registers and bits are given in the subsections that follow

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x02F0	VREGHTCL	R	0	0	VSEL	VAE	HTEN	HTDS	HTIE	HTIF
		W								
0x02F1	VREGCTRL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x02F2	VREGAPICL	R	APICLK	0	0	APIFES	APIEA	APIFE	APIE	APIF
		W								
0x02F3	VREGAPITR	R	APITR5	APITR4	APITR3	APITR2	APITR1	APITR0	0	0
		W								
0x02F4	VREGAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
		W								
0x02F5	VREGAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
		W								
0x02F6	Reserved 06	R	0	0	0	0	0	0	0	0
		W								
0x02F7	VREGHTTR	R	HTOEN	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
		W								

Table 9-3. Register Summary

Table 11-3. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ[5]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 11.4.5.5, "MSCAN Sleep Mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 11.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see Section 11.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[6],[7]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 11.4.4.5, "MSCAN Initialization Mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 11.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). The following registers enter their hard reset state and restore their default values: CANCTL0[8], CANRFLG[9], CANRIER[10], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1. The MSCAN must be in normal mode for this bit to become set.
2. See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.
3. In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 11.4.5.2, "Operation in Wait Mode" and Section 11.4.5.3, "Operation in Stop Mode")**.**
4. The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 11.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.
5. The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).
6. The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).
7. In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.
8. Not including WUPE, INITRQ, and SLPRQ.
9. TSTAT1 and TSTAT0 are not affected by initialization mode.
10. RSTAT1 and RSTAT0 are not affected by initialization mode.

11.3.2.2 MSCAN Control Register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described below.

11.4.4.2 Special System Operating Modes

The MSCAN module behaves as described within this specification in all special system operating modes. Write restrictions which exist on specific registers in normal modes are lifted for test purposes in special modes.

11.4.4.3 Emulation Modes

In all emulation modes, the MSCAN module behaves just like in normal system operating modes as described within this specification.

11.4.4.4 Listen-Only Mode

In an optional CAN bus monitoring mode (listen-only), the CAN node is able to receive valid data frames and valid remote frames, but it sends only "recessive" bits on the CAN bus. In addition, it cannot start a transmission.

If the MAC sub-layer is required to send a "dominant" bit (ACK bit, overload flag, or active error flag), the bit is rerouted internally so that the MAC sub-layer monitors this "dominant" bit, although the CAN bus may remain in recessive state externally.

11.4.4.5 MSCAN Initialization Mode

The MSCAN enters initialization mode when it is enabled (CANE=1).

When entering initialization mode during operation, any on-going transmission or reception is immediately aborted and synchronization to the CAN bus is lost, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations, the MSCAN immediately drives TXCAN into a recessive state.

NOTE

The user is responsible for ensuring that the MSCAN is not active when initialization mode is entered. The recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before setting the INITRQ bit in the CANCTL0 register. Otherwise, the abort of an on-going message can cause an error condition and can impact other CAN bus devices.

In initialization mode, the MSCAN is stopped. However, interface registers remain accessible. This mode is used to reset the CANCTL0, CANRFLG, CANRIER, CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL registers to their default values. In addition, the MSCAN enables the configuration of the CANBTR0, CANBTR1 bit timing registers; CANIDAC; and the CANIDAR, CANIDMR message filters. See Section 11.3.2.1, "MSCAN Control Register 0 (CANCTL0)," for a detailed description of the initialization mode.

Table 12-7. IIC Divider and Hold Values (Sheet 5 of 6)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
85	112	36	44	64
86	128	40	52	72
87	152	40	64	84
88	112	28	40	60
89	128	28	48	68
8A	144	36	56	76
8B	160	36	64	84
8C	176	44	72	92
8D	192	44	80	100
8E	224	52	96	116
8F	272	52	120	140
90	192	36	72	100
91	224	36	88	116
92	256	52	104	132
93	288	52	120	148
94	320	68	136	164
95	352	68	152	180
96	416	84	184	212
97	512	84	232	260
98	320	36	152	164
99	384	36	184	196
9A	448	68	216	228
9B	512	68	248	260
9C	576	100	280	292
9D	640	100	312	324
9E	768	132	376	388
9F	960	132	472	484
A0	640	68	312	324
A1	768	68	376	388
A2	896	132	440	452
A3	1024	132	504	516
A4	1152	196	568	580
A5	1280	196	632	644
A6	1536	260	760	772
A7	1920	260	952	964
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001A PWMPER6	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001B PWMPER7	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001C PWMDTY0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001D PWMDTY1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001E PWMDTY2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001F PWMDTY3	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0010 PWMDTY4	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0021 PWMDTY5	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0022 PWMDTY6	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0023 PWMDTY7	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0024 PWMSDN	R	PWMIF	PWMIE	0	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA
	W			PWMRSTRT					

= Unimplemented or Reserved

Figure 13-2. PWM Register Summary (Sheet 3 of 3)

[1] Intended for factory test purposes only.

13.3.2.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

NOTE

The first PWM cycle after enabling the channel can be irregular.

14.3.2.4 SCI Alternative Control Register 1 (SCIACR1)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-7. SCI Alternative Control Register 1 (SCIACR1)

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

Table 14-7. SCIACR1 Field Descriptions

Field	Description
7 RSEDGIE	**Receive Input Active Edge Interrupt Enable** — RXEDGIE enables the receive input active edge interrupt flag, RXEDGIF, to generate interrupt requests. 0 RXEDGIF interrupt requests disabled 1 RXEDGIF interrupt requests enabled
1 BERRIE	**Bit Error Interrupt Enable** — BERRIE enables the bit error interrupt flag, BERRIF, to generate interrupt requests. 0 BERRIF interrupt requests disabled 1 BERRIF interrupt requests enabled
0 BKDIE	**Break Detect Interrupt Enable** — BKDIE enables the break detect interrupt flag, BKDIF, to generate interrupt requests. 0 BKDIF interrupt requests disabled 1 BKDIF interrupt requests enabled

Table 18-23. FECCR Index Settings

ECCRIX[2:0]	FECCR Register Content		
	Bits [15:8]	**Bit[7]**	**Bits[6:0]**
000	Parity bits read from Flash block	0	Global address [22:16]
001	Global address [15:0]		
010	Data 0 [15:0]		
011	Data 1 [15:0] (P-Flash only)		
100	Data 2 [15:0] (P-Flash only)		
101	Data 3 [15:0] (P-Flash only)		
110	Not used, returns 0x0000 when read		
111	Not used, returns 0x0000 when read		

Table 18-24. FECCR Index=000 Bit Descriptions

Field	Description
15:8 PAR[7:0]	**ECC Parity Bits** — Contains the 8 parity bits from the 72 bit wide P-Flash data word or the 6 parity bits, allocated to PAR[5:0], from the 22 bit wide D-Flash word with PAR[7:6]=00.
6–0 GADDR[22:16]	**Global Address** — The GADDR[22:16] field contains the upper seven bits of the global address having caused the error.

The P-Flash word addressed by ECCRIX = 001 contains the lower 16 bits of the global address. The following four words addressed by ECCRIX = 010 to 101 contain the 64-bit wide data phrase. The four data words and the parity byte are the uncorrected data read from the P-Flash block.

The D-Flash word addressed by ECCRIX = 001 contains the lower 16 bits of the global address. The uncorrected 16-bit data word is addressed by ECCRIX = 010.

18.3.1.15 Flash Option Register (FOPT)

The FOPT register is the Flash option register.

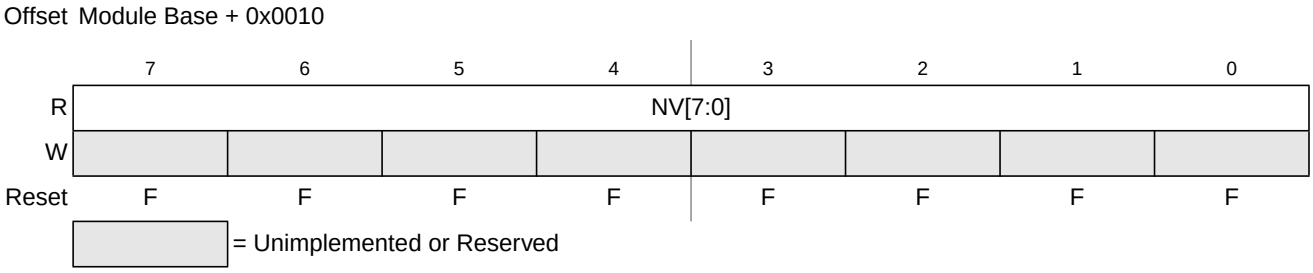

Figure 18-21. Flash Option Register (FOPT)

All bits in the FOPT register are readable but are not writable.

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller will verify that the entire Flash memory space is erased. The CCIF flag will set after the Erase Verify All Blocks operation has completed.

Table 19-29. Erase Verify All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

19.3.2.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or D-Flash block has been erased. The FCCOB upper global address bits determine which block must be verified.

Table 19-30. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Global address [22:16] of the Flash block to be verified.

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or D-Flash block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.

Table 19-31. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid global address [22:16] is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

19.3.2.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases. The section to be verified cannot cross a Kbyte boundary in the P-Flash memory space.

Table 19-39. Program Once 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 19-25)
		Set if an invalid phrase index is supplied
		Set if the requested phrase has already been programmed[1]
	FPVIOL	None
	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

1 If a Program Once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the Program Once command will be allowed to execute again on that same phrase.

19.3.2.7 Erase All Blocks Command

The Erase All Blocks operation will erase the entire P-Flash and D-Flash memory space.

Table 19-40. Erase All Blocks Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x08	Not required

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller will erase the entire Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag will set after the Erase All Blocks operation has completed.

Table 19-41. Erase All Blocks Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 19-25)
	FPVIOL	Set if any area of the P-Flash or D-Flash 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

19.3.2.8 Erase Flash Block Command

The Erase Flash Block operation will erase all addresses in a P-Flash or D-Flash block.

21.4.3 Operation in Low Power Modes

The SSD block can be configured for lower MCU power consumption in three different ways.

- Stop mode powers down the sigma-delta converter and halts clock to the modulus counter. Exit from Stop enables the sigma-delta converter and the clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Wait mode with SSDWAI bit set powers down the sigma-delta converter and halts the clock to the modulus counter. Exit from Wait enables the sigma-delta converter and clock to the modulus counter but due to the converter recovery time, the integration result should be ignored.
- Clearing SDCPU bit powers down the sigma-delta converter.

21.4.4 Stall Detection Flow

Figure 21-15 shows a flowchart and software setup for stall detection of a stepper motor. To control a second stepper motor, the SMS bit must be toggled during the SSD initialization.

A.8 MSCAN

Table A-24. MSCAN Wake-up Pulse Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	MSCAN wakeup dominant pulse filtered	t_{WUP}	—	—	1.5	µs
2	P	MSCAN wakeup dominant pulse pass	t_{WUP}	5	—	—	µs

0x0140–0x017F MSCAN (CAN0) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x015C-0x015F	CAN0IDMR4-CAN0IDMR7	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x0160-0x016F	CAN0RXFG	R	FOREGROUND RECEIVE BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								
0x0170-0x017F	CAN0TXFG	R	FOREGROUND TRANSMIT BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								

Detailed MSCAN Foreground Receive and Transmit Buffer Layout

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xXXX0	Extended ID	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	Standard ID	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	CANxRIDR0	W								
0xXXX1	Extended ID	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
	Standard ID	R	ID2	ID1	ID0	RTR	IDE=0			
	CANxRIDR1	W								
0xXXX2	Extended ID	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	Standard ID	R								
	CANxRIDR2	W								
0xXXX3	Extended ID	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	Standard ID	R								
	CANxRIDR3	W								
0xXXX4-0xXXXB	CANxRDSR0-CANxRDSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		W								
0xXXXC	CANRxDLR	R					DLC3	DLC2	DLC1	DLC0
		W								
0xXXXD	Reserved	R								
		W								
0xXXXE	CANxRTSRH	R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
		W								
0xXXXF	CANxRTSRL	R	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
		W								
0xXX10	Extended ID CANxTIDR0	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
		W								
	Standard ID	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
		W								
0xXX0xXX10	Extended ID CANxTIDR1	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
		W								
	Standard ID	R	ID2	ID1	ID0	RTR	IDE=0			
		W								
0xXX12	Extended ID CANxTIDR2	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
		W								
	Standard ID	R								
		W								