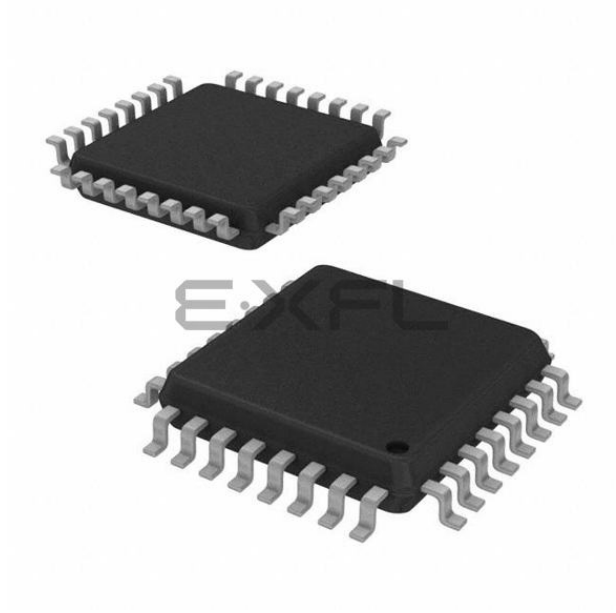


Welcome to E-XFL.COM

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	IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12vr64af0vlc

Feature	MC9S12VR16	MC9S12VR32	MC9S12VR48	MC9S12VR64	MC9S12VR48	MC9S12VR64
Chip temperature sensor	1					
Supply voltage	V_{SUP} = 6V – 18 V (normal operation) up to 40V (protected operation)					
EVDD output current	20mA @ 5V					
Maximum execution speed	25 MHz					

1.3 Chip-Level Features

On-chip modules available within the family include the following features:

- HCS12 CPU core
- 64, 48, 32 or 16 Kbyte on-chip flash with ECC
- 512 or 128 byte EEPROM with ECC
- 2 Kbyte on-chip SRAM
- Phase locked loop (IPLL) frequency multiplier with internal filter
- 1 MHz internal RC oscillator with +/-1.3% accuracy over rated temperature range
- 4-20 MHz amplitude controlled pierce oscillator
- Internal COP (watchdog) module (with separate clock source)
- Timer module (TIM) supporting input/output channels that provide a range of 16-bit input capture, output compare and counter (up to 4 channels)
- Pulse width modulation (PWM) module (up to 8 x 8-bit channels)
- 10-bit resolution successive approximation analog-to-digital converter (ADC) with up to 6 channels available on external pins
- One serial peripheral interface (SPI) module
- One serial communication interface (SCI) module supporting LIN communications (with RX connected to a timer channel for internal oscillator calibration purposes, if desired)
- Up to one additional SCI (not connected to LIN physical layer)
- One on-chip LIN physical layer transceiver fully compliant with the LIN 2.2 standard & SAE J2602-2 LIN standard
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Autonomous periodic interrupt (API) (combination with cyclic, watchdog)
- Two protected low-side outputs to drive inductive loads
- Up to two protected high-side outputs
- 4 high-voltage inputs with wake-up capability and readable internally on ADC
- Up to two 10mA high-current outputs
- 20mA high-current output for use as Hall sensor supply
- Battery voltage sense with low battery warning, internally reverse battery protected
- Chip temperature sensor

2.3.46 Port AD Interrupt Enable Register (PIE1AD)

Address 0x027D (S12VR64/48) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PIE1AD5	PIE1AD4	PIE1AD3	PIE1AD2	PIE1AD1	PIE1AD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-66. Port AD Interrupt Enable Register (PIE1AD - S12VR64/48)

1 Read: Anytime
Write: Anytime

Read: Anytime.

Address 0x027D (S12VR32/16) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PIE1AD1	PIE1AD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-67. Port AD Interrupt Enable Register (PIE1AD - S12VR32/16)

1 Read: Anytime
Write: Anytime

Table 2-46. PIE1AD Register Field Descriptions

Field	Description
5-0 PIE1AD	**Pin Interrupt Enable register 1 port AD —** This bit enables or disables the edge sensitive pin interrupt on the associated pin. An interrupt can be generated if the pin is operating in input or output mode when in use with the general-purpose or related peripheral function. For wakeup from stop mode this bit must be set to allow activating the RC oscillator. 1 Interrupt is enabled 0 Interrupt is disabled (interrupt flag masked)

- Enable the external oscillator (OSCE bit)
- Wait for oscillator to start up (UPOSC=1)
- Select the Oscillator Clock (OSCCLK) as source of the Bus Clock (PLLSEL=0).

— The PLLCLK is on and used to qualify the external oscillator clock.

4.1.2.2 Wait Mode

For S12CPMU_UHV_V8 Wait Mode is the same as Run Mode.

4.1.2.3 Stop Mode

This mode is entered by executing the CPU STOP instruction.

The voltage regulator is in Reduced Performance Mode (RPM).

NOTE

The voltage regulator output voltage may degrade to a lower value than in Full Performance Mode (FPM), additionally the current sourcing capability is substantially reduced (see also Appendix for VREG electrical parameters). Only clock source ACLK is available and the Power On Reset (POR) circuitry is functional. The Low Voltage Interrupt (LVI) and Low Voltage Reset (LVR) are disabled.

The API is available.

The Phase Locked Loop (PLL) is off.

The Internal Reference Clock (IRC1M) is off.

Core Clock, Bus Clock and BDM Clock are stopped.

Depending on the setting of the PSTP and the OSCE bit, Stop Mode can be differentiated between Full Stop Mode (PSTP = 0 or OSCE=0) and Pseudo Stop Mode (PSTP = 1 and OSCE=1). In addition, the behavior of the COP in each mode will change based on the clocking method selected by COPOSCSEL[1:0].

- **Full Stop Mode (PSTP = 0 or OSCE=0)**
  External oscillator (XOSCLCP) is disabled.
  — If COPOSCSEL1=0:
  The COP and RTI counters halt during Full Stop Mode.
  After wake-up from Full Stop Mode the Core Clock and Bus Clock are running on PLLCLK (PLLSEL=1). COP and RTI are running on IRCCLK (COPOSCSEL0=0, RTIOSCSEL=0).
  — If COPOSCSEL1=1:
  The clock for the COP is derived from ACLK (trimmable internal RC-Oscillator clock). During Full Stop Mode the ACLK for the COP can be stopped (COP static) or running (COP active) depending on the setting of bit CSAD. When bit CSAD is set the ACLK clock source for the COP is stopped during Full Stop Mode and COP continues to operate after exit from Full Stop

4.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12CPMU_UHV_V8.

4.3.1 Module Memory Map

The S12CPMU_UHV_V8 registers are shown in Figure 4-3.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0034	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0035	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								
0x0036	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x0037	CPMUFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	ILAF	OSCIF	UPOSC
		W								
0x0038	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	**PMRF**
		W								
0x0039	CPMUCLKS	R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
		W								
0x003A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x003B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x003C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x003D	RESERVED CPMUTEST0	R	0	0	0	0	0	0	0	0
		W								
0x003E	RESERVED CPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
0x003F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F0	CPMU HTCTL	R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
		W								
0x02F1	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x02F2	CPMU APICTL	R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
		W								
			= Unimplemented or Reserved							

Figure 4-3. CPMU Register Summary[1]

Table 4-12. RTI Frequency Divide Rates for RTDEC=1

RTR[3:0]	RTR[6:4] =							
	000 ($1x10^3$)	**001 ($2x10^3$)**	**010 ($5x10^3$)**	**011 ($10x10^3$)**	**100 ($20x10^3$)**	**101 ($50x10^3$)**	**110 ($100x10^3$)**	**111 ($200x10^3$)**
0000 (÷1)	$1x10^3$	$2x10^3$	$5x10^3$	$10x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$
0001 (÷2)	$2x10^3$	$4x10^3$	$10x10^3$	$20x10^3$	$40x10^3$	$100x10^3$	$200x10^3$	$400x10^3$
0010 (÷3)	$3x10^3$	$6x10^3$	$15x10^3$	$30x10^3$	$60x10^3$	$150x10^3$	$300x10^3$	$600x10^3$
0011 (÷4)	$4x10^3$	$8x10^3$	$20x10^3$	$40x10^3$	$80x10^3$	$200x10^3$	$400x10^3$	$800x10^3$
0100 (÷5)	$5x10^3$	$10x10^3$	$25x10^3$	$50x10^3$	$100x10^3$	$250x10^3$	$500x10^3$	$1x10^6$
0101 (÷6)	$6x10^3$	$12x10^3$	$30x10^3$	$60x10^3$	$120x10^3$	$300x10^3$	$600x10^3$	$1.2x10^6$
0110 (÷7)	$7x10^3$	$14x10^3$	$35x10^3$	$70x10^3$	$140x10^3$	$350x10^3$	$700x10^3$	$1.4x10^6$
0111 (÷8)	$8x10^3$	$16x10^3$	$40x10^3$	$80x10^3$	$160x10^3$	$400x10^3$	$800x10^3$	$1.6x10^6$
1000 (÷9)	$9x10^3$	$18x10^3$	$45x10^3$	$90x10^3$	$180x10^3$	$450x10^3$	$900x10^3$	$1.8x10^6$
1001 (÷10)	$10\ x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$	$500x10^3$	$1x10^6$	$2x10^6$
1010 (÷11)	$11\ x10^3$	$22x10^3$	$55x10^3$	$110x10^3$	$220x10^3$	$550x10^3$	$1.1x10^6$	$2.2x10^6$
1011 (÷12)	$12x10^3$	$24x10^3$	$60x10^3$	$120x10^3$	$240x10^3$	$600x10^3$	$1.2x10^6$	$2.4x10^6$
1100 (÷13)	$13x10^3$	$26x10^3$	$65x10^3$	$130x10^3$	$260x10^3$	$650x10^3$	$1.3x10^6$	$2.6x10^6$
1101 (÷14)	$14x10^3$	$28x10^3$	$70x10^3$	$140x10^3$	$280x10^3$	$700x10^3$	$1.4x10^6$	$2.8x10^6$
1110 (÷15)	$15x10^3$	$30x10^3$	$75x10^3$	$150x10^3$	$300x10^3$	$750x10^3$	$1.5x10^6$	$3x10^6$
1111 (÷16)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing over the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence after a system stop mode the handshake feature must be enabled again by sending the ACK_ENABLE command.

5.4.11 Serial Communication Time Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the time-out has occurred. This is the expected behavior if the handshake protocol is not enabled. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDM and CPU) when the hardware

a forced match, a state sequencer transition can occur immediately on a successful match of system busses and comparator registers. Whilst tagging, at a comparator match, the instruction opcode is tagged and only if the instruction reaches the execution stage of the instruction queue can a state sequencer transition occur. In the case of a transition to Final State, bus tracing is triggered and/or a breakpoint can be generated.

A state sequencer transition to final state (with associated breakpoint, if enabled) can be initiated by writing to the TRIG bit in the DBGC1 control register.

The trace buffer is visible through a 2-byte window in the register address map and must be read out using standard 16-bit word reads.

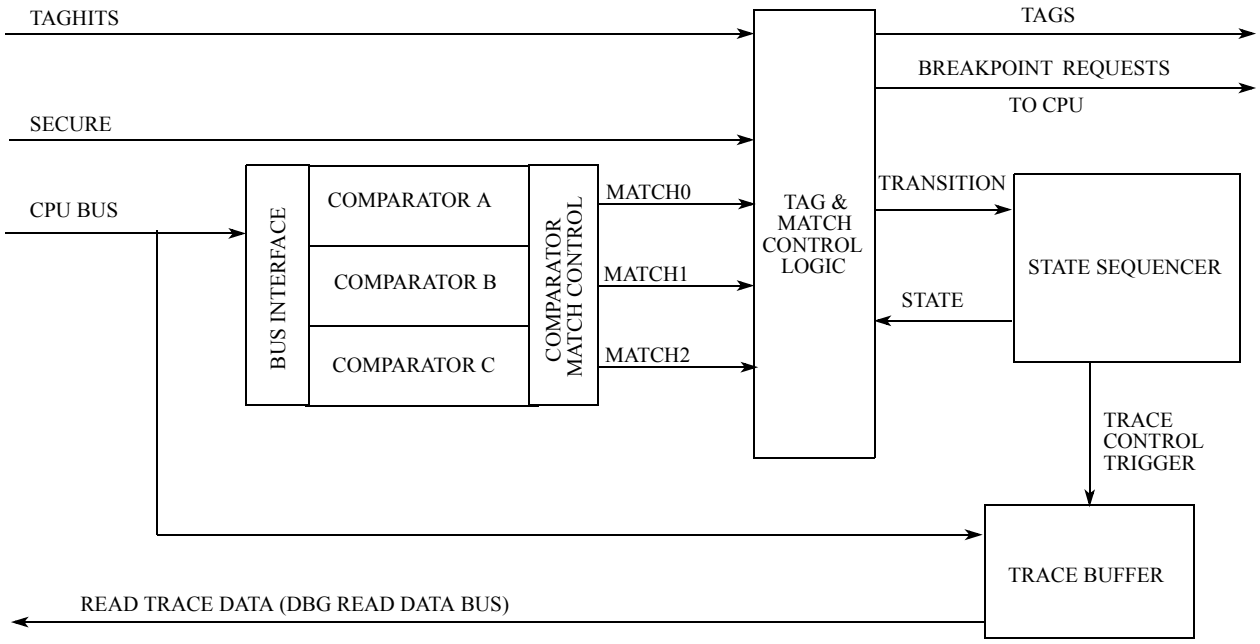


Figure 6-23. DBG Overview

6.4.2 Comparator Modes

The DBG contains three comparators, A, B and C. Each comparator compares the system address bus with the address stored in DBGXAH, DBGXAM, and DBGXAL. Furthermore, comparator A also compares the data buses to the data stored in DBGADH, DBGADL and allows masking of individual data bus bits.

All comparators are disabled in BDM and during BDM accesses.

The comparator match control logic (see Figure 6-23) configures comparators to monitor the buses for an exact address or an address range, whereby either an access inside or outside the specified range generates a match condition. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

A match can initiate a transition to another state sequencer state (see Section 6.4.4, “State Sequence Control”). The comparator control register also allows the type of access to be included in the comparison through the use of the RWE, RW, SZE, and SZ bits. The RWE bit controls whether read or write comparison is enabled for the associated comparator and the RW bit selects either a read or write access

the tagged instruction is about to be executed and the next transition is to Final State then a breakpoint is generated immediately, before the tagged instruction is carried out.

R/W monitoring, access size (SZ) monitoring and data bus monitoring are not useful if tagging is selected, since the tag is attached to the opcode at the matched address and is not dependent on the data bus nor on the type of access. Thus these bits are ignored if tagging is selected.

When configured for range comparisons and tagging, the ranges are accurate only to word boundaries.

Tagging is disabled when the BDM becomes active.

6.4.7 Breakpoints

It is possible to generate breakpoints from channel transitions to final state or using software to write to the TRIG bit in the DBGC1 register.

6.4.7.1 Breakpoints From Comparator Channels

Breakpoints can be generated when the state sequencer transitions to the Final State. If configured for tagging, then the breakpoint is generated when the tagged opcode reaches the execution stage of the instruction queue.

If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed, thus if Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 6-42). If no tracing session is selected, breakpoints are requested immediately.

If the BRK bit is set, then the associated breakpoint is generated immediately independent of tracing trigger alignment.

Table 6-42. Breakpoint Setup For CPU Breakpoints

BRK	TALIGN	DBGBRK	Breakpoint Alignment
0	0	0	Fill Trace Buffer until trigger then disarm (no breakpoints)
0	0	1	Fill Trace Buffer until trigger, then breakpoint request occurs
0	1	0	Start Trace Buffer at trigger (no breakpoints)
0	1	1	Start Trace Buffer at trigger A breakpoint request occurs when Trace Buffer is full
1	x	1	Terminate tracing and generate breakpoint immediately on trigger
1	x	0	Terminate tracing immediately on trigger

6.4.7.2 Breakpoints Generated Via The TRIG Bit

If a TRIG triggers occur, the Final State is entered whereby tracing trigger alignment is defined by the TALIGN bit. If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed, thus if Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 6-42). If no tracing session is selected, breakpoints are

11.2.3 $\overline{SS}$ — Slave Select Pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

11.2.4 SCK — Serial Clock Pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

11.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the SPI.

11.3.1 Module Memory Map

The memory map for the SPI is given in Figure 11-2. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W								
SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W								
SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W								
SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
Reserved	R								
	W								
Reserved	R								
	W								

= Unimplemented or Reserved

Figure 11-2. SPI Register Summary

11.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

11.3.2.1 SPI Control Register 1 (SPICR1)

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Figure 11-3. SPI Control Register 1 (SPICR1)

Read: Anytime

Write: Anytime

Table 11-2. SPICR1 Field Descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode. 1 SPI is in master mode.
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...) of the SCK clock. 1 Sampling of data occurs at even edges (2,4,6,...) of the SCK clock.

intended for general user access. Writing to this register when in special mode can alter the module's functionality.

Table 14-7. Reserved Register Field Descriptions

Field	Description
7-0 Reserved	These reserved bits are used for test purposes. Writing to these bits can alter the module functionality.

14.3.7 HSDRV1C Status Register (HSSR)

Module Base + 0x0005 Access: User read[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	HSOL0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-6. HSDRV1C Status Register (HSSR)

[1] Read: Anytime
Write: No Write

Table 14-8. HSDRV Status Register (HSSR) Field Descriptions

Field	Description
0 HSOL0	HSDRV1C Open-Load Status Bit This bit reflects the open-load condition on the driver pin. A delay of $t_{HLROLDT}$ must be granted after enabling the high-load resistance open-load detection function in order to read valid data. 0 No open-load condition, $\lvert I_{HS} \rvert \geq \lvert I_{HLROLDC} \rvert$ 1 Open-load condition, $\lvert I_{HS} \rvert < \lvert I_{HLROLDC} \rvert$

15.3.4 LSDRV Configuration Register (LSCR)

Module Base + 0x0001 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	LSOLE1	LSOLE0	LSE1	LSE0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-3. LSDRV Configuration Register (LSCR)

[1] Read: Anytime
Write: Anytime

Table 15-5. LSCR Register Field Descriptions

Field	Description
3-2 LSOLEx	**LSDRV High-Load Resistance Open-Load Detection Enable** These bits enable the measurement function to detect an open-load condition on the related low-side driver operating on high-load resistance loads. If the low-side driver is enabled and is not being driven by the selected source, then the high-load resistance detection circuit is activated when this bit is set to '1'. 0 high-load resistance open-load detection is disabled 1 high-load resistance open-load detection is enabled
1-0 LSEx	**LSDRV Enable** These bits control the bias of the related low-side driver circuit. 0 Low-side driver is disabled. 1 Low-side driver is enabled. **NOTE** After enabling the low-side driver (write "1" to LSEx) a settling time $t_{LS\_settling}$ is required before the low-side driver is allowed to be turned on (e.g. by writing LSDRx bits).

16.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the LIN Physical Layer.

16.3.1 Module Memory Map

A summary of the registers associated with the LIN Physical Layer module is shown in Table 16-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

NOTE

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Address Offset / Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 LPDR	R	0	0	0	0	0	0	LPDR1	LPDR0
	W								
0x0001 LPCR	R	0	0	0	0	LPE	RXONLY	LPWUE	LPPUE
	W								
0x0002 Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	W								
0x0003 LPSLRM	R	LPDTDIS	0	0	0	0	0	LPSLR1	LPSLR0
	W								
0x0004 Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	W								
0x0005 LPSR	R	LPDT	0	0	0	0	0	0	0
	W								
0x0006 LPIE	R	LPDTIE	LPOCIE	0	0	0	0	0	0
	W								
0x0007 LPIF	R	LPDTIF	LPOCIF	0	0	0	0	0	0
	W								

Figure 16-2. Register Summary

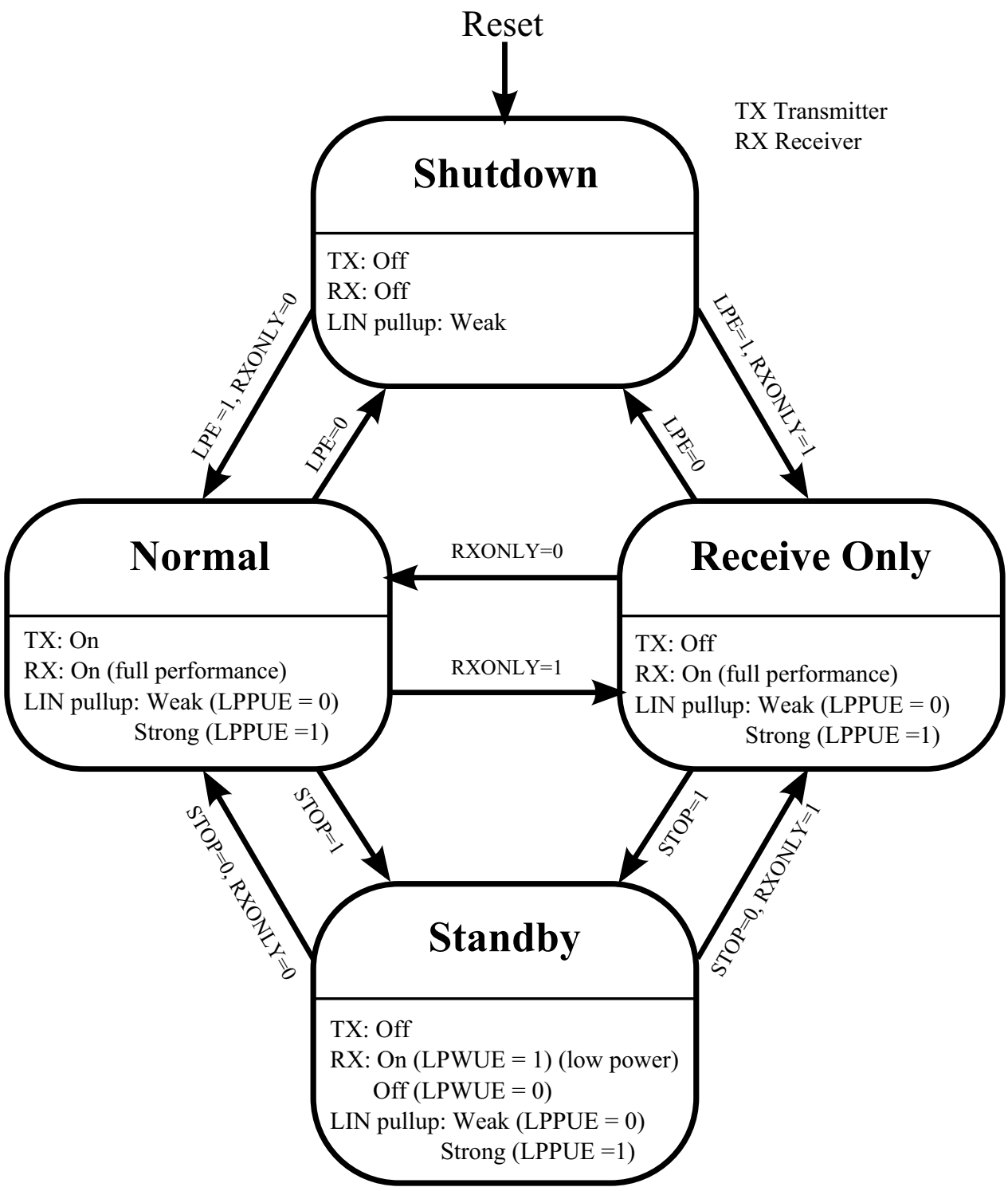


Figure 16-11. LIN Physical Layer Mode Transitions

18.3 Memory Map and Registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

CAUTION

Writing to the Flash registers while a Flash command is executing (that is indicated when the value of flag CCIF reads as ’0’) is not allowed. If such action is attempted the write operation will not change the register value.

Writing to the Flash registers is allowed when the Flash is not busy executing commands (CCIF = 1) and during initialization right after reset, despite the value of flag CCIF in that case (refer to Section 18.6 Initialization for a complete description of the reset sequence).

.

Table 18-2. FTMRG Memory Map

Global Address (in Bytes)	Size (Bytes)	Description
0x0_0000 - 0x0_03FF	1,024	Register Space
0x0_0400 – 0x0_05FF	512	EEPROM Memory
0x0_4000 – 0x0_7FFF	16,284	NVMRES[1]=1 : NVM Resource area (see Figure 18-2)

[1] See NVMRES description in Section 18.4.3 Internal NVM resource (NVMRES)

18.3.1 Module Memory Map

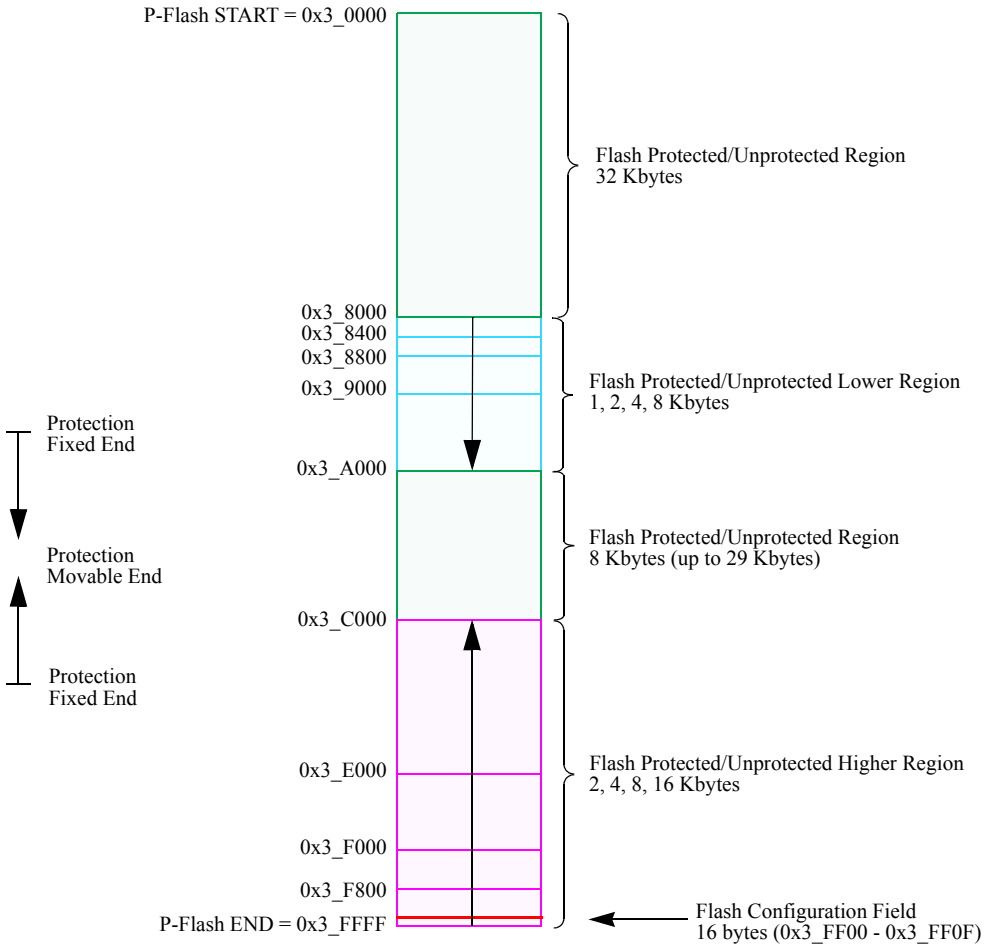
The S12 architecture places the P-Flash memory between global addresses 0x3_0000 and 0x3_FFFF as shown in Table 18-3.The P-Flash memory map is shown in Figure .

The FPROT register, described in Section 18.3.2.9 P-Flash Protection Register (FPROT), can be set to

Table 18-3. P-Flash Memory Addressing

Global Address	Size (Bytes)	Description
0x3_0000 – 0x3_FFFF	64 K	P-Flash Block Contains Flash Configuration Field (see Table 18-4)

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 protectio. Two separate memory


P-Flash START = 0x3_0000
Flash Protected/Unprotected Region
32 Kbytes
0x3_8000
0x3_8400
0x3_8800
0x3_9000
Flash Protected/Unprotected Lower Region
1, 2, 4, 8 Kbytes
Protection
Fixed End
0x3_A000
Flash Protected/Unprotected Region
8 Kbytes (up to 29 Kbytes)
Protection
Movable End
0x3_C000
Protection
Fixed End
0x3_E000
Flash Protected/Unprotected Higher Region
2, 4, 8, 16 Kbytes
0x3_F000
0x3_F800
P-Flash END = 0x3_FFFF
Flash Configuration Field
16 bytes (0x3_FF00 - 0x3_FF0F)

18.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

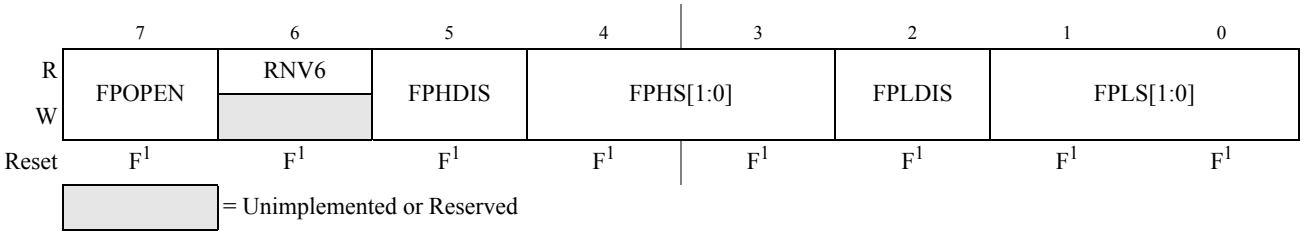


Figure 18-12. Flash Protection Register (FPROT)

1 Loaded from IFR Flash configuration field, during reset sequence.

The (unreserved) bits of the FPROT register are writable in Normal Single Chip Mode with the restriction that the size of the protected region can only be increased (see Section 18.3.2.9.1, "P-Flash Protection Restrictions," and Table 18-21).All (unreserved) bits of the FPROT register are writable without restriction in Special Single Chip Mode .

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 18-4) as indicated by reset condition 'F' in Figure 18-12. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 18-17. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 18-18 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x3_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 18-19. The FPHS bits can only be written to while the FPHDIS bit is set.

18.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skeeped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

18.5.3 Mode and Security Effects on Flash Command Availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 18-27.

18.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and EEPROT protection registers, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

P.27 0x0160-0x0167 LIN Physical Layer (LINPHY)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0164	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0165	LPSR	R	LPDT	0	0	0	0	0	0	0
		W								
0x0166	LPIE	R	LPDTIE	LPOCIE	0	0	0	0	0	0
		W								
0x0167	LPIF	R	LPDTIF	LPOCIF	0	0	0	0	0	0
		W								

P.28 0x0168-0x016F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0168-0x016F	Reserved	R	0	0	0	0	0	0	0	0
		W								

P.29 0x0170-0x0177 Supply Voltage Sense (BATS)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0170	BATE	R	0	0	BVHS	BVLS	BSUAE	BSUSE	0BSEAE	BSESE
		W								
0x0171	BATSR	R	0	0	0	0	0	0	BVHC	BVLC
		W								
0x0172	BATIE	R	0	0	0	0	0	0	BVHIE	BVLIE
		W								
0x0173	BATIF	R	0	0	0	0	0	0	BVHIF	BVLIF
		W								
0x0174	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0175	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0176	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0177	Reserved	R	0	0	0	0	0	0	0	0
		W								

P.30 0x0178-023F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0178-0x023F	Reserved	R	0	0	0	0	0	0	0	0
		W								