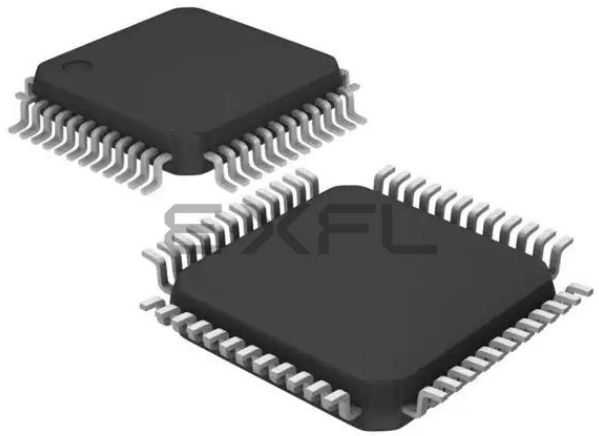

Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 10x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvla12f0mlf

- Internal COP (watchdog) module
- analog-to-digital converter (ADC) with 10 -bit or 12 -bit resolution and up to 10 channels available on external pins and V_{bg} (bandgap) result reference
- PGA module with two input channels
- One 8-bit 5V digital-to-analog converter (DAC)
- One analog comparators (ACMP) with rail-to-rail inputs
- MSCAN (1 Mbit/s, CAN 2.0 A, B software compatible) module
- One serial peripheral interface (SPI) module
- One serial communication interface (SCI) module with interface to internal LIN physical layer transceiver (with RX connected to a timer channel for frequency 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
- 6-channel timer module (TIM0) with input capture/output compare
- 2-channel timer module (TIM1) with input capture/output compare
- Inter-IC (IIC) module
- 8-channel Pulse Width Modulation module (PWM)
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Autonomous periodic interrupt (API), supports cyclic wakeup from Stop mode
- Pins to support 25 mA drive strength to VSSX
- Pin to support 20 mA drive strength from VDDX (EVDD)
- High Voltage Input (HVI)
- Supply voltage sense with low battery warning
- On-chip temperature sensor, temperature value can be measured with ADC or can generate a high temperature warning
- Up to 23 pins can be used as keyboard wake-up interrupt (KWI)

1.4 Module Features

The following sections provide more details of the integrated modules.

1.4.1 S12Z Central Processor Unit (CPU)

The S12Z CPU is a revolutionary high-speed core, with code size and execution efficiencies over the S12X CPU. The S12Z CPU also provides a linear memory map eliminating the inconvenience and performance impact of page swapping.

- Harvard Architecture - parallel data and code access
- 3 stage pipeline
- 32-Bit wide instruction and databus
- 32-Bit ALU
- 24-bit addressing (16 MByte linear address space)

1.13.3 Flash IFR Mapping

Table 1-15. Flash IFR Mapping

Target																IFR Byte Address
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
				ADC reference conversion using VDDA/VSSA												0x1F_C040 & 0x1F_C041
				ADC reference conversion using VRH/VRL												0x1F_C042 & 0x1F_C043

1.14 Application Information

1.14.1 ADC Calibration

For applications that do not provide external ADC reference voltages, the VDDA/VSSA supplies can be used as sources for VRH/VRL respectively. Since the VDDA must be connected to VDDX at board level in the application, the accuracy of the VDDA reference is limited by the internal voltage regulator accuracy. In order to compensate for VDDA reference voltage variation in this case, the reference voltage is measured during production test using the internal reference voltage VBG, which has a narrow variation over temperature and external voltage supply. VBG is mapped to an internal channel of the ADC module, see Table 1-7. The resulting 12-bit right justified ADC conversion results of VBG are stored to the flash IFR for reference, as listed in Table 1-15.

The measurement conditions of the reference conversion are listed in the device electrical parameters appendix. By measuring the voltage VBG in the application environment and comparing the result to the reference value in the IFR, it is possible to determine the current ADC reference voltage V_{RH}:

$$V_{RH} = \frac{StoredReference}{ConvertedReference} \bullet 5V$$

The exact absolute value of an analog conversion can be determined as follows:

$$Result = ConvertedADInput \bullet \frac{StoredReference \bullet 5V}{ConvertedReference \bullet 2^{n}}$$

With:

ConvertedADInput:	Result of the analog to digital conversion of the desired pin
ConvertedReference:	Result of internal channel conversion
StoredReference:	Value in IFR location
n:	ADC resolution (12 bit)

3.6.2 Register Descriptions

3.6.2.1 ACMP Control Register 0 (ACMPC0)

Module Base + 0x0000 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	ACE	ACOPE	ACOPS	ACDLY	ACHYS1-0		ACMOD1-0	
Reset	0	0	0	0	0	0	0	0

Figure 3-2. ACMP Control Register (ACMPC0)

[1] Read: Anytime
Write: Anytime

Table 3-2. ACMPC0 Register Field Descriptions

Field	Description
7 ACE	**ACMP Enable —** This bit enables the ACMP module. When set the related input pins are connected with the low pass input filters. **Note:** After setting ACE to 1 an initialization delay of 127 bus clock cycles must be accounted for. During this time the comparator output path to all subsequent logic (ACO, ACIF, timer link) is held at its current state. When setting ACE to 0 the current state of the comparator will be maintained.For ACMPO a delay of $t_{ACMP\_delay\_en}$ must be accounted for. 0 ACMP disabled 1 ACMP enabled
6 ACOPE	**ACMP Output Pin Enable —** This bit enables the ACMP output on external ACMPO pin. 0 ACMP output pin disabled 1 ACMP output is driven out to ACMPO
5 ACOPS	**ACMP Output Polarity Select** — This bit selects the output polarity on ACMPO. 0 ACMPO is ACMP output 1 ACMPO is ACMP output inverted
4 ACDLY	**ACMP Input Filter Select for Inputs ACMP_0 and ACMP_1** — This bit selects the analog input filter characteristics resulting in a signal propagation delay of t_{ACMP_delay}. 0 Select input filter with low speed characteristics 1 Select input filter with high speed characteristics

Table 8-10. SRAM_ECC Interrupt Sources

Module Interrupt Sources	Local Enable
Single bit ECC error	ECCIE[SBEEIE]

8.3.6 ECC Algorithm

The table below shows the equation for each ECC bit based on the 16 bit data word.

Table 8-11. ECC Calculation

ECC bit	Use data
ECC[0]	~ (^ (data[15:0] & 0x443F))
ECC[1]	~ (^ (data[15:0] & 0x13C7))
ECC[2]	~ (^ (data[15:0] & 0xE1D1))
ECC[3]	~ (^ (data[15:0] & 0xEE60))
ECC[4]	~ (^ (data[15:0] & 0x3E8A))
ECC[5]	~ (^ (data[15:0] & 0x993C))

8.3.7 ECC Debug Behavior

For debug purposes, it is possible to read and write the uncorrected use data and the raw ECC value directly from the memory. For these debug accesses a register interface is available. The debug access is performed with the lowest priority; other memory accesses must be done before the debug access starts. If a debug access is requested during an ongoing memory initialization process, then the debug access is performed if the memory initialization process is done.

If the ECCDRR bit is set, then the automatic single bit ECC error repair function for all read accesses is disabled. In this case a read access from a system memory location with single bit ECC error will produce correct data and the single bit ECC error is flagged by the SBEEIF, but the data inside the system memory are unchanged.

By writing wrong ECC values into the system memory the debug access can be used to force single and double bit ECC errors to check the software error handling.

It is not possible to set the ECCDW or ECCDR bit if the previous debug access is ongoing (ECCDW or ECCDR bit active). This ensures that the ECCDD and ECCDE registers contains consistent data. The software should read out the status of the ECCDW and ECCDR register bit before a new debug access is requested.

8.3.7.1 ECC Debug Memory Write Access

Writing one to the ECCDW bit performs a debug write access to the memory address defined by register DPTR. During this access, the raw data DDATA and the ECC value DECC are written directly into the system memory. If the debug write access is done, the ECCDW register bit is cleared. The debug write

Table 9-13. 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)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

9.3.2.17 Low Voltage Control Register (CPMULVCTL)

The CPMULVCTL register allows the configuration of the low-voltage detect features.

Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LVDS	LVIE	LVIF
W								
Reset	0	0	0	0	0	U	0	U

The Reset state of LVDS and LVIF depends on the external supplied VDDA level

= Unimplemented or Reserved

Figure 9-21. Low Voltage Control Register (CPMULVCTL)

Read: Anytime

Write: LVIE and LVIF are write anytime, LVDS is read only

Table 9-18. CPMULVCTL Field Descriptions

Field	Description
2 LVDS	**Low-Voltage Detect Status Bit** — This read-only status bit reflects the voltage level on VDDA. Writes have no effect. 0 Input voltage VDDA is above level V_{LVID} or RPM. 1 Input voltage VDDA is below level V_{LVIA} and FPM.
1 LVIE	**Low-Voltage Interrupt Enable Bit** 0 Interrupt request is disabled. 1 Interrupt will be requested whenever LVIF is set.
0 LVIF	**Low-Voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request. 0 No change in LVDS bit. 1 LVDS bit has changed.

NOTE

Each conversion flow control bit (SEQA, RSTA, TRIG, LDOK) must be controlled by software or internal interface according to the requirements described in Section 10.6.3.2.4, “The two conversion flow control Mode Configurations and overview summary in Table 10-11.

10.5.2.19 ADC Command Index Register (ADCCIDX)

It is important to note that these bits do not represent absolute addresses instead it is a sample index (object size 32bit).

Module Base + 0x001C

	7	6	5	4	3	2	1	0
R	0	0	CMD_IDX[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-22. ADC Command Index Register (ADCCIDX)

Read: Anytime

Write: NA

Table 10-27. ADCCIDX Field Descriptions

Field	Description
5-0 CMD_IDX [5:0]	**ADC Command Index Bits** — These bits represent the command index value for the conversion commands relative to the two CSL start addresses in the memory map. These bits do not represent absolute addresses instead it is a sample index (object size 32bit). See also Section 10.6.3.2.2, “Introduction of the two Command Sequence Lists (CSLs) for more details.

10.5.2.21 ADC Result Index Register (ADCRIDX)

It is important to note that these bits do not represent absolute addresses instead it is a sample index (object size 16bit).

Module Base + 0x0020

	7	6	5	4	3	2	1	0
R	0	0	RES_IDX[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-24. ADC Result Index Register (ADCRIDX)

Read: Anytime

Write: NA

Table 10-29. ADCRIDX Field Descriptions

Field	Description
5-0 RES_IDX[5:0]	**ADC Result Index Bits** — These read only bits represent the index value for the conversion results relative to the two RVL start addresses in the memory map. These bits do not represent absolute addresses instead it is a sample index (object size 16bit). See also Section 10.6.3.2.3, “Introduction of the two Result Value Lists (RVLs) for more details.

Table 13-37. CPU vs. MSCAN Operating Modes

CPU Mode	MSCAN Mode			
	Normal	Reduced Power Consumption		
		Sleep	Power Down	Disabled (CANE=0)
RUN	CSWAI = X[1] SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = 1 SLPAK = 1		CSWAI = X SLPRQ = X SLPAK = X
WAIT	CSWAI = 0 SLPRQ = 0 SLPAK = 0	CSWAI = 0 SLPRQ = 1 SLPAK = 1	CSWAI = 1 SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X
STOP			CSWAI = X SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X

1 'X' means don't care.

13.4.5.1 Operation in Run Mode

As shown in Table 13-37, only MSCAN sleep mode is available as low power option when the CPU is in run mode.

13.4.5.2 Operation in Wait Mode

The WAI instruction puts the MCU in a low power consumption stand-by mode. If the CSWAI bit is set, additional power can be saved in power down mode because the CPU clocks are stopped. After leaving this power down mode, the MSCAN restarts and enters normal mode again.

While the CPU is in wait mode, the MSCAN can be operated in normal mode and generate interrupts (registers can be accessed via background debug mode).

13.4.5.3 Operation in Stop Mode

The STOP instruction puts the MCU in a low power consumption stand-by mode. In stop mode, the MSCAN is set in power down mode regardless of the value of the SLPRQ/SLPAK and CSWAI bits (Table 13-37).

13.4.5.4 MSCAN Normal Mode

This is a non-power-saving mode. Enabling the MSCAN puts the module from disabled mode into normal mode. In this mode the module can either be in initialization mode or out of initialization mode. See Section 13.4.4.5, "MSCAN Initialization Mode".

15.3.2.12 Timer Input Capture/Output Compare Registers High and Low 0–5(TCxH and TCxL)

Module Base + 0x0010 = TC0H
0x0012 = TC1H
0x0014=TC2H
0x0016=TC3H
0x0018=TC4H
0x001A=TC5H
0x001C=RESERVD
0x001E=RESERVD

	15	14	13	12	11	10	9	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 15-18. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L
0x0013 = TC1L
0x0015 =TC2L
0x0017=TC3L
0x0019 =TC4L
0x001B=TC5L
0x001D=RESERVD
0x001F=RESERVD

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

Figure 15-19. Timer Input Capture/Output Compare Register x Low (TCxL)

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should take place before low byte otherwise it will give a different result.

The Prescaler can be calculated as follows depending on logical value of the PTPS[7:0] and PRNT bit:

PRNT = 1 : Prescaler = PTPS[7:0] + 1

Table 15-17. Precision Timer Prescaler Selection Examples when PRNT = 1

PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0	Prescale Factor
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	3
0	0	0	0	0	0	1	1	4
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
0	0	0	1	0	0	1	1	20
0	0	0	1	0	1	0	0	21
0	0	0	1	0	1	0	1	22
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
1	1	1	1	1	1	0	0	253
1	1	1	1	1	1	0	1	254
1	1	1	1	1	1	1	0	255
1	1	1	1	1	1	1	1	256

15.4 Functional Description

This section provides a complete functional description of the timer TIM16B6CV3 block. Please refer to the detailed timer block diagram in Figure 15-22 as necessary.

the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

NOTE

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte will be lost.

19.4.7 Low Power Mode Options

19.4.7.1 SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

19.4.7.2 SPI in Wait Mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of wait mode (i.e., if the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

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 21-2. Register Summary

21.3.2 Register Descriptions

This section describes all the S12LINPHYV2 registers and their individual bits.

21.4.4 Interrupts

The interrupt vector requested by the S12LINPHYV2 is listed in Table 21-10. Vector address and interrupt priority is defined at the MCU level.

The module internal interrupt sources are combined into a single interrupt request at the device level.

Table 21-10. Interrupt Vectors

Module Interrupt Source	Module Internal Interrupt Source	Local Enable
LIN Interrupt (LPI)	LIN Txd-Dominant Timeout Interrupt (LPDTIF)	LPDTIE = 1
	LIN Overcurrent Interrupt (LPOCIF)	LPOCIE = 1

21.4.4.1 Overcurrent Interrupt

The transmitter is protected against overcurrent. In case of an overcurrent condition occurring within a time frame called t_{OCLIM} starting from LPTxD falling edge, the current through the transmitter is limited (the transmitter is not shut down). The masking of an overcurrent event within the time frame t_{OCLIM} is meant to avoid “false” overcurrent conditions that can happen during the discharging of the LIN bus. If an overcurrent event occurs out of this time frame, the transmitter is disabled and the LPOCIF flag is set.

In order to re-enable the transmitter again, the following prerequisites must be met:

1) Overcurrent condition is over

2) LPTxD is recessive or the LIN Physical Layer is in shutdown or receive only mode for a minimum of a transmit bit time.

To re-enable the transmitter then, the LPOCIF flag must be cleared (by writing a 1).

NOTE

Please make sure that LPOCIF=1 before trying to clear it. It is not allowed to try to clear LPOCIF if LPOCIF=0 already.

After clearing LPOCIF, if the overcurrent condition is still present or the LPTxD pin is dominant while being in normal mode, the transmitter remains disabled and the LPOCIF flag is set again after a time to indicate that the attempt to re-enable has failed. This time is equal to:

- minimum 1 IRC period (1 us) + 2 bus periods
- maximum 2 IRC periods (2 us) + 3 bus periods

If the bit LPOCIE is set in the LPIE register, an interrupt is requested.

Figure 21-12 shows the different scenarios for overcurrent interrupt handling.

21.5 Application Information

21.5.1 Module Initialization

The following steps should be used to configure the module before starting the transmission:

1. Set the slew rate in the LPSLRM register to the desired transmission baud rate.
2. When using the LIN Physical Layer for other purposes than LIN transmission, de-activate the dominant timeout feature in the LPSLRM register if needed.
3. In most cases, the internal pullup should be enabled in the LPCR register.
4. Route the desired source in the PIM module to the LIN Physical Layer.
5. Select the transmit mode (Receive only mode or Normal mode) in the LPCR register.
6. If the SCI is selected as source, activate the wake-up feature in the LPCR register if needed for the application (SCI active edge interrupt must also be enabled).
7. Enable the LIN Physical Layer in the LPCR register.
8. Wait for a minimum of a transmit bit.
9. Begin transmission if needed.

NOTE

It is not allowed to try to clear LPOCIF or LPDTIF if they are already cleared. Before trying to clear an error flag, always make sure that it is already set.

21.5.2 Interrupt handling in Interrupt Service Routine (ISR)

Both interrupts (TxD-dominant timeout and overcurrent) represent a failure in transmission. To avoid more disturbances on the transmission line, the transmitter is de-activated in both cases. The interrupt subroutine must take care of clearing the error condition and starting the routine that re-enables the transmission. For that purpose, the following steps are recommended:

1. First, the cause of the interrupt must be cleared:
   — The overcurrent will be gone after the transmitter has been disabled.
   — The TxD-dominant timeout condition will be gone once the selected source for LPTxD has turned recessive.
2. Clear the corresponding enable bit (LPDTIE or LPOCIE) to avoid entering the ISR again until the flags are cleared.
3. Notify the application of the error condition (LIN Error handler) and leave the ISR.

In the LIN Error handler, the following sequence is recommended:

1. Disable the LIN Physical Layer (LPCR) while re-configuring the transmission.
   — If the receiver must remain enabled, set the LIN Physical Layer into receive only mode instead.
2. Do all required configurations (SCI, etc.) to re-enable the transmission.
3. Wait for a transmit bit (this is needed to successfully re-enable the transmitter).

Upon clearing CCIF to launch the Erase EEPROM Sector command, the Memory Controller will erase the selected Flash sector and verify that it is erased. The CCIF flag will set after the Erase EEPROM Sector operation has completed.

Table 22-66. Erase EEPROM Sector Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is supplied see Table 22-2
		Set if a misaligned word address is supplied (global address [0] != 0)
	FPVIOL	Set if the selected area of the EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

22.4.7.17 Protection Override Command

The Protection Override command allows the user to temporarily override the protection limits, either decreasing, increasing or disabling protection limits, on P-Flash and/or EEPROM, if the comparison key provided as a parameter loaded on FCCOB matches the value of the key previously programmed on the Flash Configuration Field (see Table 22-3). The value of the Protection Override Comparison Key must not be 16'hFFFF, that is considered invalid and if used as argument will cause the Protection Override feature to be disabled. Any valid key value that does not match the value programmed in the Flash Configuration Field will cause the Protection Override feature to be disabled. Current status of the Protection Override feature can be observed on FPSTAT FPOVRD bit (see Section 22.3.2.4, "Flash Protection Status Register (FPSTAT)).

Table 22-67. Protection Override Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x13	Protection Update Selection [1:0] See Table 22-68
FCCOB1	Comparison Key	
FCCOB2	reserved	New FPROT value
FCCOB3	reserved	New DFPROT value

Table 22-68. Protection Override selection description

Protection Update Selection code [1:0]	Protection register selection
bit 0	Update P-Flash protection 0 - keep unchanged (do not update) 1 - update P-Flash protection with new FPROT value loaded on FCCOB

Table A-17. Run and Wait Current Characteristics for ZVL(A)128/96/64

Conditions are: V_{SUP} = 18V, see Table A-14 and Table A-15						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	Run Current, -40°C < T_J ≤150°C, f_{bus}=32MHz	I_{SUPR}	—	21	27	mA
2	Run Current, 150°C < T_J < 175°C, f_{bus}=25MHz	I_{SUPR}	—	21	27	mA
3	Wait Current, -40°C < T_J ≤ 150°C, f_{bus}=32MHz	I_{SUPW}	—	13	20	mA
4	Wait Current, 150°C < T_J < 175°C, f_{bus}=25MHz	I_{SUPW}	—	13	20	mA

Table A-18. Stop Current Characteristics for ZVL(S)32/16/8

Conditions are: V_{SUP} = 12V						
Num	**Rating**[1]	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Stop Current all modules off						
1	T_J= -40°C	I_{SUPS}	—	20	28	µA
2	T_J= 25°C	I_{SUPS}	—	23	33	µA
3	T_J= 85°C	I_{SUPS}	—	44	55	µA
4	T_J= 105°C	I_{SUPS}	—	63	85	µA
5	T_J= 125°C	I_{SUPS}	—	115	156	µA
Stop Current API enabled & LINPHY in standby						
6	T_J=25°C	I_{SUPS}	—	38	—	µA

1 If MCU is in STOP long enough then $T_A = T_J$. Die self heating due to stop current can be ignored.

Table A-19. Stop Current Characteristics for ZVL(A)128/96/64

Conditions are: V_{SUP} = 12V						
Num	**Rating**[1]	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Stop Current all modules off						
1	T_J= -40°C	I_{SUPS}	—	20	40	µA
2	T_J= 25°C	I_{SUPS}	—	25	50	µA
3	T_J= 85°C	I_{SUPS}	—	60	107	µA
4	T_J= 105°C	I_{SUPS}	—	78	176	µA
5	T_J= 125°C	I_{SUPS}	—	130	301	µA
Stop Current API enabled & LINPHY in standby						
6	T_J=25°C	I_{SUPS}	—	53	—	µA

1 If MCU is in STOP long enough then $T_A = T_J$. Die self heating due to stop current can be ignored.

Table A-20. Pseudo Stop Current Characteristics for ZVL(S)32/16/8

Conditions are: V_{SUP} = 12V, API, COP & RTI enabled						
Num	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	T_J = 25°C	I_{SUPPS}	—	155	350	µA

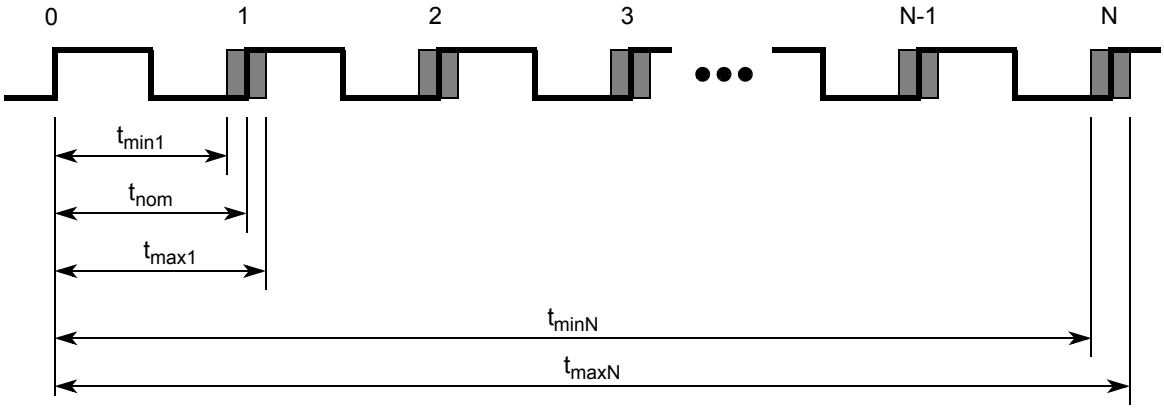

Figure B-1. Jitter Definitions

The relative deviation of t_{nom} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

$$J(N) = \max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

The following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}}$$

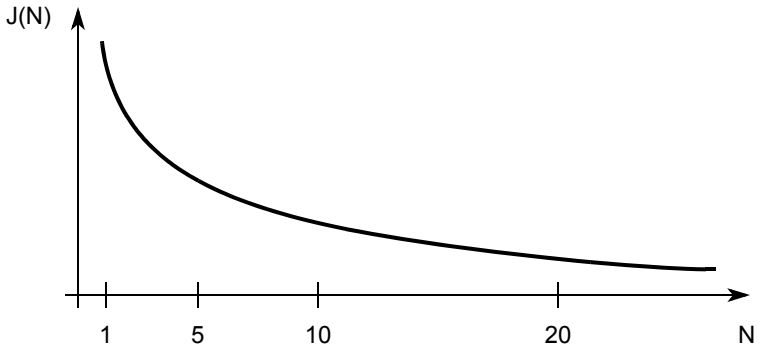

Figure B-2. Maximum Bus Clock Jitter Approximation

O.2 0x0010–0x001F S12ZINT

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001A	INT_CFDATA2	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001B	INT_CFDATA3	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001C	INT_CFDATA4	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001D	INT_CFDATA5	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001E	INT_CFDATA6	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x001F	INT_CFDATA7	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								