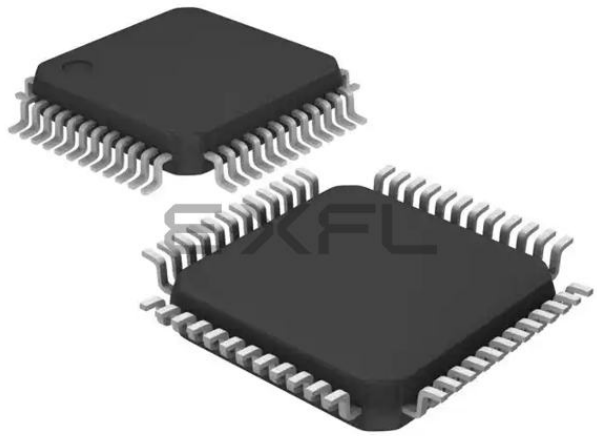


Welcome to E-XFL.COM

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, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
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/s912zvc19f0mlf

2.2.1 Internal Routing Options

The following table summarizes the internal routing options.

Table 2-3. Internal Routing Options

Internal Signal	Connects to	Routing Bits
ACMP0 out	TIM1 IC2	T1IC2RR
ACMP1 out	TIM1 IC3	T1IC3RR
ACLK	TIM0 IC2	T0IC2RR
RXD0, RXD1	TIM0 IC3	T0IC3RR1-0
TIM0 OC2	ADC0 Trigger	TRIG0RR1-0, TRIG0NEG

2.3 Memory Map and Register Definition

This section provides a detailed description of all port integration module registers.

- General-purpose data output availability depends on prioritization; input data registers always reflect the pin status independent of the use.
- Pull-device availability, pull-device polarity, wired-or mode, key-wake up functionality are independent of the prioritization unless noted differently.
- For availability of individual bits refer to Section 2.3.1, “Register Map” and Table 2-33.

2.3.3.1 Port Data Register

Address 0x0260 PTE
0x0280 PTADH
0x0281 PTADL
0x02C0 PTT
0x02D0 PTS
0x02F0 PTP
0x0310 PTJ

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTx7	PTx6	PTx5	PTx4	PTx3	PTx2	PTx1	PTx0
Reset	0	0	0	0	0	0	0	0

Figure 2-10. Port Data Register

[1] Read: Anytime. The data source is depending on the data direction value.
Write: Anytime

This is a generic description of the standard port data registers. Refer to Table 2-33 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 2-11. Port Data Register Field Descriptions

Field	Description
7-0 PTx7-0	**Port Data — General purpose input/output data** This register holds the value driven out to the pin if the pin is used as a general purpose output. When not used with the alternative function (refer to Table 2-2), these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.4.12 Port L Test Enable Register (PTTEL)

Address 0x033F Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTTEL1	PTTEL0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-32. Port L Test Enable Register (PTTEL)

[1] Read: Anytime
Write: Anytime

Table 2-32. PTTEL Register Field Descriptions

Field	Description
1-0 PTTEL1-0	**Port L Test Enable** — This bit forces the input buffer of the HVI pin active while using the analog function to support open input detection in run mode. Refer to Section 2.5.5, "Open Input Detection on PL[1:0] (HVI)"). In stop mode this bit has no effect. **Note:** In direct mode (PTADIRL=1) the digital input buffer is not enabled. 1 Input buffer enabled when used with analog function and not in direct mode (PTADIRL=0) 0 Input buffer disabled when used with analog function

2.4 Functional Description

2.4.1 General

Each pin except BKGD can act as general-purpose I/O. In addition each pin can act as an output or input of a peripheral module.

2.4.2 Registers

Table 2-33 lists the implemented configuration bits which are available on each port. These registers except the pin input registers can be written at any time, however a specific configuration might not become active. For example a pullup device does not become active while the port is used as a push-pull output.

Unimplemented bits read zero.

Table 2-33. Bit Indices of Implemented Register Bits per Port

		Port Data Register	Port Input Register	Data Direction Register	Pull Device Enable Register	Polarity Select Register	Port Interrupt Enable Register	Port Interrupt Flag Register	Digital Input Enable Register	Reduced Drive Register	Wired-Or Mode Register
Port		**PT**	**PTI**	**DDR**	**PER**	**PPS**	**PIE**	**PIF**	**DIE**	**RDR**	**WOM**
E		1-0	1-0	1-0	1-0	1-0	-	-	-	-	-
ADH		7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	-	-
ADL		7-0	7-0	7-0	7-0	7-0	7-0	7-0	7-0	-	-
T		7-0	7-0	7-0	7-0	7-0	-	-	-	-	-
S		7-0	7-0	7-0	7-0	7-0	7-0	7-0	-	-	7-0
P		7-0	7-0	7-0	7-0	7-0	7-0	7-0	-	6-4,2,0	-
J		1-0	1-0	1-0	1-0	1-0	-	-	-	-	1-0
L		-	1-0	-	-	1-0	1-0	1-0	1-0	-	-

Table 2-34 shows the effect of enabled peripheral features on I/O state and enabled pull devices.

Table 2-34. Effect of Enabled Features

Enabled Feature[1]	Related Signal(s)	Effect on I/O state	Effect on enabled pull device
CPMU OSC	EXTAL, XTAL	CPMU takes control	Forced off
TIMx output compare y	IOCx_y	Forced output	Forced off, pulldown forced off if open-drain
TIMx input capture y	IOCx_y	None[2]	None[3]
SPIx	MISOx, MOSx, SCKx, $\overline{SSx}$	SPI takes control	Forced off if output, pulldown forced off if open-drain
SCIx transmitter	TXDx	Forced output	Forced off, pulldown forced off if open-drain
SCIx receiver	RXDx	Forced input	None[3]
IICx	SDAx, SCLx	Forced open-drain	Pulldown forced off
S12ZDBG	DBGEEV	None[2]	None[3]
PWMx channel y	PWMx_y	Forced output	Forced off
ADCx	ANy	None[2 4]	None[3]
	VRH		
ACMPx	ACMPx_0, ACMPx_1	None[2 4]	None[3]
	ACMPOx	Forced output	Forced off
DACx	AMPPx, AMPMx	None[2 4]	None[3]
	DACUx, AMPx	Digital output forced off	Forced off

7.2.2.5 ECC Debug Data (ECCDDH, ECCDDL)

Module Base + 0x000C — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDATA[15:8]							
Reset	0	0	0	0	0	0	0	0

Module Base + 0x000D — Access: User read/write

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

= Unimplemented

Figure 7-6. ECC Debug Data (ECCDDH, ECCDDL)

[1] Read: Anytime
Write: Anytime

Table 7-6. ECCDD Register Field Descriptions

Field	Description
DDATA [23:0]	**ECC Debug Raw Data** — This register contains the raw data which will be written into the system memory during a debug write command or the read data from the debug read command.

7.2.2.6 ECC Debug ECC (ECCDE)

Module Base + 0x000E — Access: User read/write[1]

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

[1] Read: Anytime
Write: Anytime

Figure 7-7. ECC Debug ECC (ECCDE)

Table 7-7. ECCDE Field Description

Field	Description
5:0 DECC[5:0]	**ECC Debug ECC** — This register contains the raw ECC value which will be written into the system memory during a debug write command or the ECC read value from the debug read command.

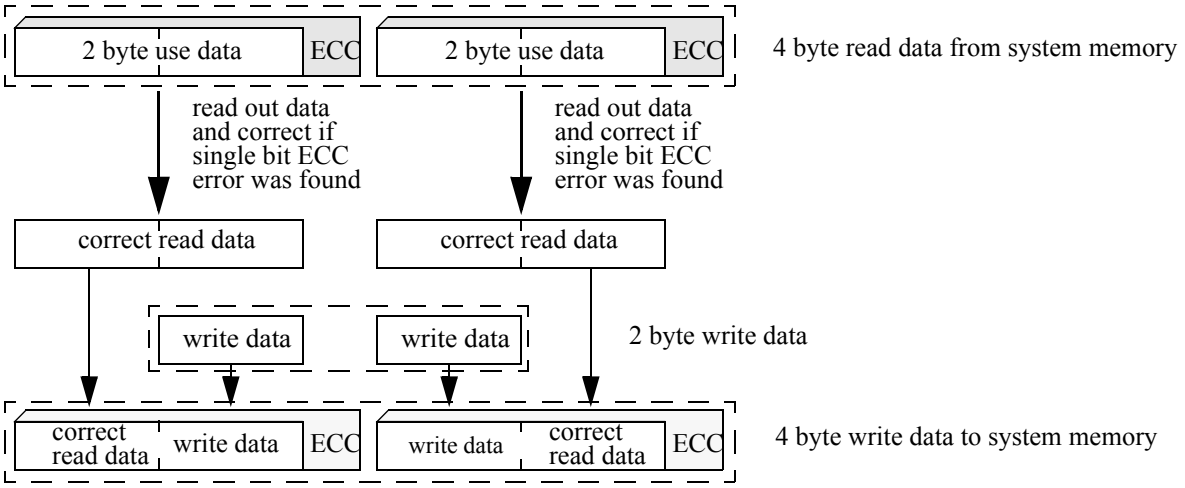


Figure 7-9. 2 byte non-aligned write access

7.3.3 Memory Read Access

During each memory read access an ECC check is performed. If the logic detects a single bit ECC error, then the module corrects the data, so that the access initiator module receives correct data. In parallel, the logic writes the corrected data back to the memory, so that this read access repairs the single bit ECC error. This automatic ECC read repair function is disabled by setting the ECCDRR bit.

If a single bit ECC error was detected, then the SBEEIF flag is set.

If the logic detects a double bit ECC error, then the data word is flagged as invalid, so that the access initiator module can ignore the data.

7.3.4 Memory Initialization

To avoid spurious ECC error reporting, memory operations that allow a read before a first write (like the read-modify-write operation of the unaligned access) require that the memory contains valid ECC values before the first read-modify-write access is performed. The ECC module provides logic to initialize the complete memory content with zero during the power up phase. During the initialization process the access to the SRAM is disabled and the RDY status bit is cleared. If the initialization process is done, SRAM access is possible and the RDY status bit is set.

7.3.5 Interrupt Handling

This section describes the interrupts generated by the SRAM_ECC module and their individual sources. Vector addresses and interrupt priority are defined at the MCU level.

Table 8-12. 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)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

8.3.2.24 Reserved Register CPMUTEST2

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU_UHV_V7's functionality.

Module Base + 0x001C

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

= Unimplemented or Reserved

Figure 8-33. Reserved Register CPMUTEST2

Read: Anytime

Write: Only in Special Mode

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

Any change in PLL configuration or any other event which causes the PLL lock status to be cleared leads to a loss of the oscillator status information as well (UPOSC=0).

Oscillator status change interrupts are locally enabled with the OSCIE bit.

NOTE

Loosing the oscillator status (UPOSC=0) affects the clock configuration of the system[1]. This needs to be dealt with in application software.

8.6.1.4 Low-Voltage Interrupt (LVI)

In FPM the input voltage VDDA is monitored. Whenever VDDA drops below level $V_{LVIA,}$ the status bit LVDS is set to 1. When VDDA rises above level V_{LVID} the status bit LVDS is cleared to 0. An interrupt, indicated by flag LVIF = 1, is triggered by any change of the status bit LVDS if interrupt enable bit LVIE = 1.

8.6.1.5 HTI - High Temperature Interrupt

In FPM the junction temperature T_J is monitored. Whenever T_J exceeds level T_{HTIA} the status bit HTDS is set to 1. Vice versa, HTDS is reset to 0 when T_J get below level T_{HTID}. An interrupt, indicated by flag HTIF = 1, is triggered by any change of the status bit HTDS, if interrupt enable bit HTIE = 1.

8.6.1.6 Autonomous Periodical Interrupt (API)

The API sub-block can generate periodical interrupts independent of the clock source of the MCU. To enable the timer, the bit APIFE needs to be set.

The API timer is either clocked by the Autonomous Clock (ACLK - trimmable internal RC oscillator) or the Bus Clock. Timer operation will freeze when MCU clock source is selected and Bus Clock is turned off. The clock source can be selected with bit APICLK. APICLK can only be written when APIFE is not set.

The APIR[15:0] bits determine the interrupt period. APIR[15:0] can only be written when APIFE is cleared. As soon as APIFE is set, the timer starts running for the period selected by APIR[15:0] bits. When the configured time has elapsed, the flag APIF is set. An interrupt, indicated by flag APIF = 1, is triggered if interrupt enable bit APIE = 1. The timer is re-started automatically again after it has set APIF.

The procedure to change APICLK or APIR[15:0] is first to clear APIFE, then write to APICLK or APIR[15:0], and afterwards set APIFE.

The API Trimming bits ACLKTR[5:0] must be set so the minimum period equals 0.2 ms if stable frequency is desired.

See Table 8-20 for the trimming effect of ACLKTR[5:0].

1. For details please refer to “8.4.6 System Clock Configurations”

– *When finished:*
This bit is cleared when the first conversion command of the sequence from top of active Sequence Command List is loaded

– *Mandatory Requirement:*
- In all ADC conversion flow control modes a Restart Event causes bit RSTA to be set. Bit SEQA is set simultaneously by ADC hardware if:
* ADC not idle (a conversion or conversion sequence is ongoing and current CSL not finished) and no Sequence Abort Event in progress (bit SEQA not already set or set simultaneously via internal interface or data bus)
* ADC idle but RVL done condition not reached
The RVL done condition is reached by one of the following:
* A “End Of List” command type has been executed
* A Sequence Abort Event is in progress or has been executed (bit SEQA already set or set simultaneously via internal interface or data bus)
The ADC executes the Sequence Abort Event followed by the Restart Event for the conditions described before or only a Restart Event.
- In ADC conversion flow control mode “Trigger Mode” a Restart Event causes bit TRIG being set automatically. Bit TRIG is set when no conversion or conversion sequence is ongoing (ADC idle) and the RVL done condition is reached by one of the following:
* A “End Of List” command type has been executed
* A Sequence Abort Event is in progress or has been executed
The ADC executes the Restart Event followed by the Trigger Event.
- In ADC conversion flow control mode “Trigger Mode” a Restart Event and a simultaneous Trigger Event via internal interface or data bus causes the TRIG_EIF bit being set and ADC cease operation.

- **Restart Event + CSL Exchange (Swap)**
Internal Interface Signals: Restart + LoadOK
Corresponding Bit Names: RSTA + LDOK

– *Function:*
Go to top of active CSL (clear index register for CSL) and switch to other offset register for address calculation if configured for double buffer mode (exchange the CSL list)
Requested by:
- Internal interface with the assertion of Interface Signal Restart the interface Signal LoadOK is evaluated and bit LDOK is set accordingly (bit LDOK set if Interface Signal LoadOK asserted when Interface Signal Restart asserts).
- Write Access via data bus to set control bit RSTA simultaneously with bit LDOK.

– *When finished:*
Bit LDOK can only be cleared if it was set as described before and both bits (LDOK, RSTA) are cleared when the first conversion command from top of active Sequence Command List is loaded

– *Mandatory Requirement:*
No ongoing conversion or conversion sequence
Details if using the internal interface:

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.

Only bits related to implemented channels are valid.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
	W								
0x0001 CFORC	R	0	0	0	0	0	0	0	0
	W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0002 OC7M	R	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
	W								
0x0003 OC7D	R	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
	W								
0x0004 TCNTH	R	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
	W								
0x0005 TCNTL	R	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	W								
0x0006 TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
	W								
0x0007 TTOV	R	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
	W								
0x0008 TCTL1	R	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
	W								
0x0009 TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	W								
0x000A TCTL3	R	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
	W								
0x000B TCTL4	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	W								
0x000C TIE	R	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
	W								
0x000D TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	W								
0x000E TFLG1	R	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
	W								
0x000F TFLG2	R	TOF	0	0	0	0	0	0	0
	W								
0x0010–0x001F TCxH–TCxL[1]	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0020 PACTL	R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
	W								

Figure 11-5. TIM16B8CV3 Register Summary (Sheet 1 of 2)

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0015 PWMPER1[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0016 PWMPER2[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0017 PWMPER3[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0018 PWMPER4[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0019 PWMPER5[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001A PWMPER6[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001B PWMPER7[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001C PWMDTY0[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001D PWMDTY1[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001E PWMDTY2[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001F PWMDTY3[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0010 PWMDTY4[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0021 PWMDTY5[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0022 PWMDTY6[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0023 PWMDTY7[2]	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								

= Unimplemented or Reserved

Figure 13-2. The scalable PWM Register Summary (Sheet 3 of 4)

14.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Writes to a reserved register locations do not have any effect and reads of these locations return a zero. Details of register bit and field function follow the register diagrams, in bit order.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SCIBDH[1]	R	SBR15	SBR14	SBR13	SBR12	SBR11	SBR10	SBR9	SBR8
	W								
0x0001 SCIBDL[1]	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	W								
0x0002 SCICR1[1]	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
	W								
0x0000 SCIASR1[2]	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
	W								
0x0001 SCIACR1[2]	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
	W								
0x0002 SCIACR2[2]	R	IREN	TNP1	TNP0	0	0	BERRM1	BERRM0	BKDFE
	W								
0x0003 SCICR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	W								
0x0004 SCISR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	W								
0x0005 SCISR2	R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
	W								
0x0007 SCIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0

1.These registers are accessible if the AMAP bit in the SCISR2 register is set to zero.

2,These registers are accessible if the AMAP bit in the SCISR2 register is set to one.

= Unimplemented or Reserved

Figure 14-2. SCI Register Summary

Figure 14-17 shows two cases of break detect. In trace RXD_1 the break symbol starts with the start bit, while in RXD_2 the break starts in the middle of a transmission. If BRKDFE = 1, in RXD_1 case there will be no byte transferred to the receive buffer and the RDRF flag will not be modified. Also no framing error or parity error will be flagged from this transfer. In RXD_2 case, however the break signal starts later during the transmission. At the expected stop bit position the byte received so far will be transferred to the receive buffer, the receive data register full flag will be set, a framing error and if enabled and appropriate a parity error will be set. Once the break is detected the BRKDIF flag will be set.

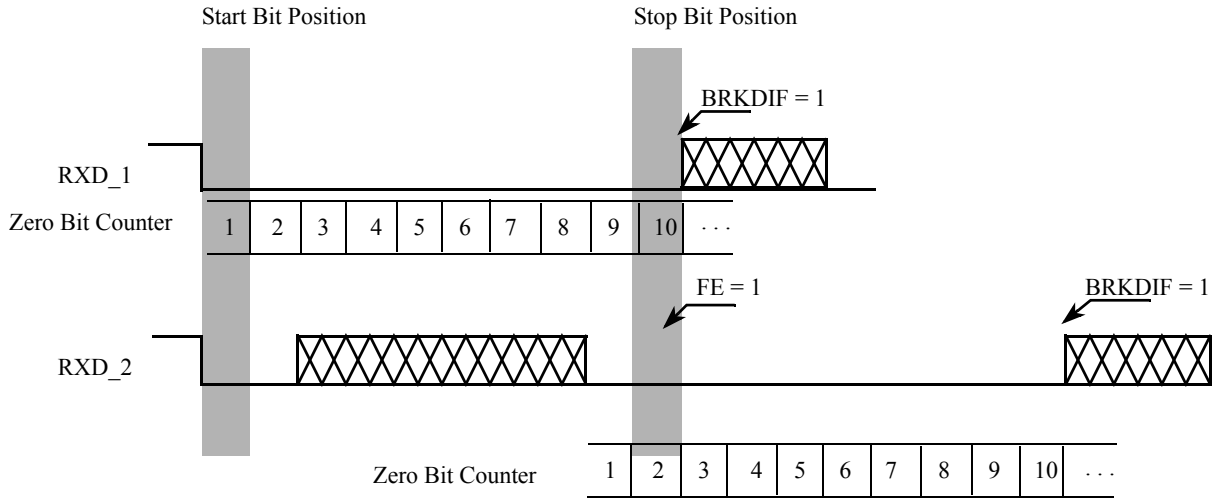


Figure 14-17. Break Detection if BRKDFE = 1 (M = 0)

14.4.5.4 Idle Characters

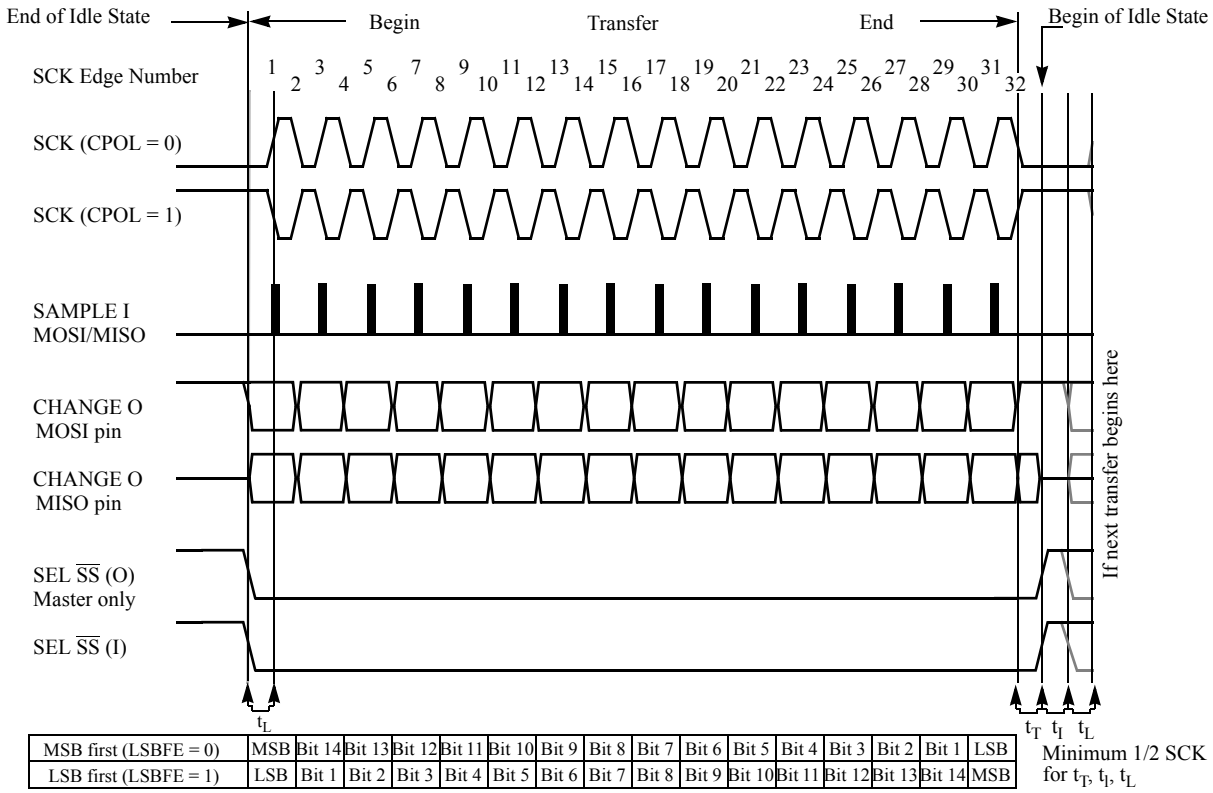
An idle character (or preamble) contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCI control register 1 (SCICR1). The preamble is a synchronizing idle character that begins the first transmission initiated after writing the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the TXD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the frame currently being transmitted.

NOTE

When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current frame shifts out through the TXD pin. Setting TE after the stop bit appears on TXD causes data previously written to the SCI data register to be lost. Toggle the TE bit for a queued idle character while the TDRE flag is set and immediately before writing the next byte to the SCI data register.

If the TE bit is clear and the transmission is complete, the SCI is not the master of the TXD pin



t_L = Minimum leading time before the first SCK edge

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 15-13. SPI Clock Format 0 (CPHA = 0), with 16-Bit Transfer Width selected (XFRW = 1)

In slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

15.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n[1]-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

1. n depends on the selected transfer width, please refer to Section 15.3.2.2, “SPI Control Register 2 (SPICR2)

16.4.1.11 General Call Address

To broadcast using a general call, a device must first generate the general call address($00), then after receiving acknowledge, it must transmit data.

In communication, as a slave device, provided the GCEN is asserted, a device acknowledges the broadcast and receives data until the GCEN is disabled or the master device releases the bus or generates a new transfer. In the broadcast, slaves always act as receivers. In general call, IAAS is also used to indicate the address match.

In order to distinguish whether the address match is the normal address match or the general call address match, IBDR should be read after the address byte has been received. If the data is $00, the match is general call address match. The meaning of the general call address is always specified in the first data byte and must be dealt with by S/W, the IIC hardware does not decode and process the first data byte.

When one byte transfer is done, the received data can be read from IBDR. The user can control the procedure by enabling or disabling GCEN.

16.4.2 Operation in Run Mode

This is the basic mode of operation.

16.4.3 Operation in Wait Mode

IIC operation in wait mode can be configured. Depending on the state of internal bits, the IIC can operate normally when the CPU is in wait mode or the IIC clock generation can be turned off and the IIC module enters a power conservation state during wait mode. In the later case, any transmission or reception in progress stops at wait mode entry.

16.4.4 Operation in Stop Mode

The IIC is inactive in stop mode for reduced power consumption. The STOP instruction does not affect IIC register states.

16.5 Resets

The reset state of each individual bit is listed in Section 16.3, “Memory Map and Register Definition,” which details the registers and their bit-fields.

16.6 Interrupts

IICV3 uses only one interrupt vector.

Table 16-11. Interrupt Summary

Interrupt	Offset	Vector	Priority	Source	Description
IIC Interrupt	—	—	—	IBAL, TCF, IAAS bits in IBSR register	When either of IBAL, TCF or IAAS bits is set may cause an interrupt based on arbitration lost, transfer complete or address detect conditions

22.4.5.3 Valid Flash Module Commands

Table 22-28. present the valid Flash commands, as enabled by the combination of the functional MCU mode (Normal SingleChip NS, Special Singlechip SS) with the MCU security state (Unsecured, Secured).

Table 22-28. Flash Commands by Mode and Security State

FCMD	Command	Unsecured		Secured	
		NS[1]	SS[2]	NS[3]	SS[4]
0x01	Erase Verify All Blocks	*	*	*	
0x02	Erase Verify Block	*	*	*	
0x03	Erase Verify P-Flash Section	*	*	*	
0x04	Read Once	*	*	*	
0x06	Program P-Flash	*	*	*	
0x07	Program Once	*	*	*	
0x08	Erase All Blocks		*		
0x09	Erase Flash Block	*	*	*	
0x0A	Erase P-Flash Sector	*	*	*	
0x0B	Unsecure Flash		*		
0x0C	Verify Backdoor Access Key	*		*	
0x0D	Set User Margin Level	*	*	*	
0x0E	Set Field Margin Level		*		
0x10	Erase Verify EEPROM Section	*	*	*	
0x11	Program EEPROM	*	*	*	
0x12	Erase EEPROM Sector	*	*	*	
0x13	Protection Override	*	*	*	

1 Unsecured Normal Single Chip mode

2 Unsecured Special Single Chip mode.

3 Secured Normal Single Chip mode.

4 Secured Special Single Chip mode.Please refer to <st-blue>Section 22.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM.

Figure L-2. 64 LQFP Exposed Pad Package