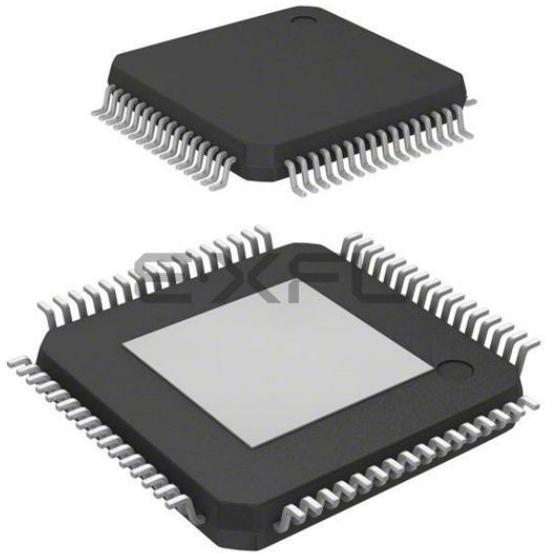


Welcome to E-XFL.COM

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	42
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 16x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvc19f0vkh

The MC9S12ZVC-Family supports BDC communication throughout the device Stop mode. During Stop mode, writes to control registers can alter the operation and lead to unexpected results. It is thus recommended not to reconfigure the peripherals during STOP using the debugger.

1.10.3 Low Power Modes

The device has two dynamic-power modes (run and wait) and two static low-power modes (stop and pseudo stop). For a detailed description refer to the CPMU section.

- Dynamic power mode: Run
  - Run mode is the main full performance operating mode with the entire device clocked. The user can configure the device operating speed through selection of the clock source and the phase locked loop (PLL) frequency. To save power, unused peripherals must not be enabled.
- Dynamic power mode: Wait
  - This mode is entered when the CPU executes the WAI instruction. In this mode the CPU does not execute instructions. The internal CPU clock is switched off. All peripherals can be active in system wait mode. For further power consumption the peripherals can individually turn off their local clocks. Asserting $\overline{\text{RESET}}$, $\overline{\text{XIRQ}}$, $\overline{\text{IRQ}}$, or any other interrupt that is not masked, either locally or globally by a CCR bit, ends system wait mode.
- Static power modes:
  Static power (Stop) modes are entered following the CPU STOP instruction unless an NVM command is active. When no NVM commands are active, the Stop request is acknowledged and the device enters either Stop or Pseudo Stop mode.
  - Pseudo-stop: In this mode the system clocks are stopped but the oscillator is still running and the real time interrupt (RTI), watchdog (COP) and Autonomous Periodic Interrupt (API) may be enabled. Other peripherals are turned off. This mode consumes more current than system STOP mode but, as the oscillator continues to run, the full speed wake up time from this mode is significantly shorter.
  - Stop: In this mode the oscillator is stopped and clocks are switched off and the VREG enters reduced performance mode (RPM). The counters and dividers remain frozen. The autonomous periodic interrupt (API) may remain active but has a very low power consumption. The KWx pins and the SCI module can be configured to wake the device, whereby current consumption is negligible.
    If the BDC is enabled, in Stop mode, the VREG remains in full performance mode. With BDC enabled and BDCCIS bit set, then all clocks remain active during Stop mode to allow BDC access to internal peripherals. If the BDC is enabled and BDCCIS is clear, then the BDCSI clock remains active to allow BDC register access, but other clocks (with the exception of the API) are switched off. With the BDC enabled during Stop, the VREG full performance mode and clock activity lead to higher current consumption than with BDC disabled.
    If the BDC is enabled in Stop mode, then the voltage monitoring remains enabled.

NOTE

The U-bit should be cleared and the S-bit (stop enable) should be cleared in the CPU condition code register (CCR) to execute the STOP instruction.Otherwise the STOP instruction is considered as a NOP.

NOTE

This document assumes the availability of all features offered in the largest package option. Refer to the package and pinout section in the device overview for functions not available in lower pin count packages.

2.1.2 Features

The PIM includes these distinctive registers:

- Data registers and data direction registers for ports E, AD, T, S, P and J when used as general-purpose I/O
- Control registers to enable pull devices and select pullups/pulldowns on ports E, AD, T, S, P and J
- Control register to enable open-drain (wired-or) mode on port S and J
- Control register to enable digital input buffers on port AD and L
- Interrupt flag register for pin interrupts and key-wakeup (KWU) on port AD, S, P and L
- Control register to configure $\overline{\text{IRQ}}$ pin operation
- Control register to enable ECLK output
- Routing registers to support signal relocation on external pins and control internal routings:
  - 2 PWM1 (fast) channels to alternative pins (1 option each)
  - 4 TIM0 channels to pins (1 option each)
  - IIC0 to alternative pins (2 options)
  - SCI0 to alternative pins (1 option)
  - SCI1 to alternative pins (1 option)
  - SPI0 to alternative pins (1 option)
  - ADC0 trigger input with edge select from internal TIM output compare channel link (OC0_2) or external pins (3 options)
  - Various MSCAN0-CANPHY0 routing options for standalone use and conformance testing
  - MSCAN0 to alternative pins (1 option)
  - Internal RXD0 and RXD1 link to TIM input capture channel (IC0_3) for baud rate detection
  - Internal ACLK link to TIM input capture channel (IC0_2) for calibration and clock monitoring purposes
  - SENT_TX pin link to 2 TIM0 input capture channels (IC0_0 and IC0_1)
  - Internal ACMP0 link to TIM1 (fast) input capture channel (IC1_2)
  - Internal ACMP1 link to TIM1 (fast) input capture channel (IC1_3)

A standard port pin has the following minimum features:

- Input/output selection
- 5V output drive
- 5V digital and analog input
- Input with selectable pullup or pulldown device

Optional features supported on dedicated pins:

- Open drain for wired-or connections (ports S and J)
- Interrupt input with glitch filtering
- High current drive strength from VDDX with over-current protection
- High current drive strength to VSSX
- Selectable drive strength (port P)

2.2 External Signal Description

This section lists and describes the signals that do connect off-chip.

Table 2-2 shows all pins with the pins and functions that are controlled by the PIM. Routing options are denoted in parentheses.

NOTE

If there is more than one function associated with a pin, the output priority is indicated by the position in the table from top (highest priority) to bottom (lowest priority).

Table 2-2. Pin Functions and Priorities

Port	Pin	Pin Function & Priority	I/O	Description	Routing Register Bit	Func. after Reset
—	BKGD	MODC[1]	I	MODC input during $\overline{\text{RESET}}$	—	BKGD
		BKGD	I/O	S12ZBDC communication	—	
E	PE1	XTAL	—	CPMU OSC signal	—	GPIO
		PTE[1]	I/O	GPIO	—	
	PE0	EXTAL	—	CPMU OSC signal	—	
		PTE[0]	I/O	GPIO	—	
AD	PAD15	AN15	I	ADC0 analog input	—	GPIO
		(ETRIG0)	I	ADC0 external trigger	TRIG0RR1-0	
		PTADH[7]/ KWADH[7]	I/O	GPIO with pin-interrupt and key-wakeup	—	
	PAD14-10	AN14:AN10	I	ADC0 analog input	—	
		PTADH[6:2]/ KWADH[6:2]	I/O	GPIO with pin-interrupt and key-wakeup	—	
	PAD9	AMP	O	DAC buffered analog output	—	
		AN9	I	ADC0 analog input	—	
		PTADH[1]/ KWADH[1]	I/O	GPIO with pin-interrupt and key-wakeup	—	
	PAD8	AMPM	I	DAC standalone OPAMP inverting input	—	
		AN8	I	ADC0 analog input	—	
		PTADH[0]/ KWADH[0]	I/O	GPIO with pin-interrupt and key-wakeup	—	

2.3.3.5 Polarity Select Register

Address 0x0268 PPSE
0x0288 PPSADH
0x0289 PPSADL
0x02C4 PPST
0x02D4 PPSS
0x02F4 PPSP
0x0314 PPSJ

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSx7	PPSx6	PPSx5	PPSx4	PPSx3	PPSx2	PPSx1	PPSx0
Reset								
Ports E:	0	0	0	0	0	0	1	1
Others:	0	0	0	0	0	0	0	0

Figure 2-14. Polarity Select Register

[1] Read: Anytime
Write: Anytime

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

Table 2-15. Polarity Select Register Field Descriptions

Field	Description
7-0 PPSx7-0	**Pull Polarity Select** — Configure pull device and pin interrupt edge polarity on input pin This bit selects a pullup or a pulldown device if enabled on the associated port input pin. If a port has interrupt functionality this bit also selects the polarity of the active edge. **Note:** If MSCAN is active a pullup device can be activated on the RXCAN input; attempting to select a pulldown disables the pull-device. 1 Pulldown device selected; rising edge selected 0 Pullup device selected; falling edge selected

2.3.3.6 Port Interrupt Enable Register

Address 0x028C PIEADH
0x028D PIEADL
0x02D6 PIES
0x02F6 PIEP
0x0336 PIEL

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIEx7	PIEx6	PIEx5	PIEx4	PIEx3	PIEx2	PIEx1	PIEx0
Reset	0	0	0	0	0	0	0	0

Figure 2-15. Port Interrupt Enable Register

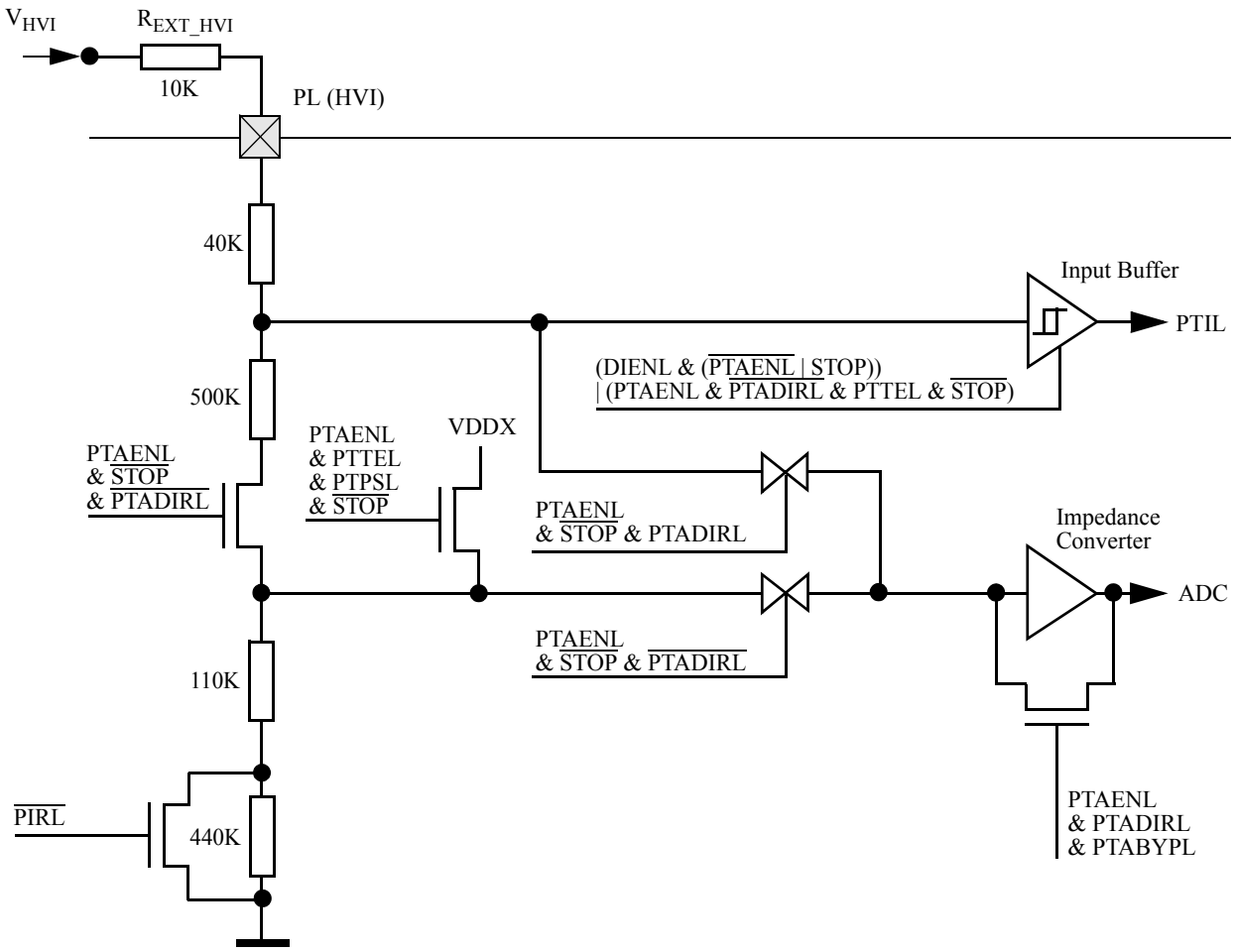


Figure 2-35. HVI Block Diagram

Voltages up to V_{HVI} can be applied to the HVI pin. Internal voltage dividers scale the input signals down to logic level. There are two modes, digital and analog, where these signals can be processed.

2.4.5.1 Digital Mode Operation

In digital mode the input buffer is enabled (DIENL=1 & PTAENL=0). The synchronized pin input state determined at threshold level V_{TH_HVI} can be read in register PTIL. An interrupt flag (PIFL) is set on input transitions of the configured edge polarity (PPSL). An interrupt (PIFL) is generated if enabled (PIEL=1) and the interrupt being set (PIFL=1). Wakeup from stop mode is supported.

2.4.5.2 Analog Mode Operation

In analog mode (PTAENL=1) the input buffer is forced off and the voltage applied to a selectable HVI pin can be measured on its related internal ADC channel (refer to device overview section for channel assignment). One of two input divider ratios ($Ratio_{H\_HVI}$, $Ratio_{L\_HVI}$) can be chosen (PIRL) on the analog

The ACK handshake protocol does not support nested ACK pulses. If a BDC command is not acknowledged by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDC command. The host can decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

Commands With-Status do not generate an ACK, thus if ACK is enabled and a With-Status command is issued, the host must use the 512 cycle timeout to calculate when the data is ready for retrieval.

3.4.7.1 Long-ACK Hardware Handshake Protocol

If a command results in an error condition, whereby a BDCCSRL flag is set, then the target generates a “Long-ACK” low pulse of 64 BDCSI clock cycles, followed by a brief speed pulse. This indicates to the host that an error has occurred. The host can subsequently read BDCCSR to determine the type of error. Whether normal ACK or Long-ACK, the ACK pulse is not issued earlier than 32 BDCSI clock cycles after the BDC command was issued. The end of the BDC command is assumed to be the 16th BDCSI clock cycle of the last bit. The 32 cycle minimum delay differs from the 16 cycle delay time with ACK disabled.

If a BDC access request does not gain access within 512 core clock cycles, the request is aborted, the NORESP flag is set and a Long-ACK pulse is transmitted to indicate an error case.

Following a STOP or WAI instruction, if the BDC is enabled, the first ACK, following stop or wait mode entry is a long ACK to indicate an exception.

3.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. To abort a command that has not responded with an ACK pulse, the host controller generates a sync request (by driving BKGD low for at least 128 BDCSI clock cycles and then driving it high for one BDCSI clock cycle as a speedup pulse). By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 3.4.4.1, “SYNC”, and assumes that the pending command and therefore the related ACK pulse are being aborted. After the SYNC protocol has been completed the host is free to issue new BDC commands.

The host can issue a SYNC close to the 128 clock cycles length, providing a small overhead on the pulse length to assure the sync pulse is not misinterpreted by the target. See Section 3.4.4.1, “SYNC”.

Figure 3-11 shows a SYNC command being issued after a READ_MEM, which aborts the READ_MEM command. Note that, after the command is aborted a new command is issued by the host.

Address: 0x00001D

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

= Unimplemented or Reserved

Figure 5-10. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

1 Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001E

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

= Unimplemented or Reserved

Figure 5-11. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

1 Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x00001F

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

= Unimplemented or Reserved

Figure 5-12. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

1 Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

7.2.2 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 functions follow the register diagrams, in bit order.

7.2.2.1 ECC Status Register (ECCSTAT)

Module Base + 0x00000 — Access: User read only[1]

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

1 Read: Anytime
Write: Never

Figure 7-2. ECC Status Register (ECCSTAT)

Table 7-2. ECCSTAT Field Description

Field	Description
0 RDY	**ECC Ready**— Shows the status of the ECC module. 0 Internal SRAM initialization is ongoing, access to the SRAM is disabled 1 Internal SRAM initialization is done, access to the SRAM is enabled

7.2.2.2 ECC Interrupt Enable Register (ECCIE)

Module Base + 0x00001 — Access: User read/write[1]

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

1 Read: Anytime
Write: Anytime

Figure 7-3. ECC Interrupt Enable Register (ECCIE)

Table 7-3. ECCIE Field Description

Field	Description
0 SBEEIE	**Single bit ECC Error Interrupt Enable** — Enables Single ECC Error interrupt. 0 Interrupt request is disabled 1 Interrupt will be requested whenever SBEEIF is set

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.

8.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12CPMU_UHV_V7.

8.3.1 Module Memory Map

The S12CPMU_UHV_V7 registers are shown in Figure 8-3.

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	CPMU RESERVED00	R	0	0	0	0	0	0	0	0
		W								
0x0001	CPMU RESERVED01	R	0	0	0	0	0	0	0	0
		W								
0x0002	CPMU RESERVED02	R	0	0	0	0	0	0	0	0
		W								
0x0003	CPMURFLG	R	0	PORF	LVRF	0	COPRF	0	OMRF	PMRF
		W								
0x0004	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0005	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								
0x0006	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x0007	CPMUIFLG	R	RTIF	0	0	LOCKIF	LOCK	0	OSCIF	UPOSC
		W								
0x0008	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W								
0x0009	CPMUCLKS	R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
		W								
0x000A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x000B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x000C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x000D	RESERVED CPMUTEST0	R	0	0	0	0	0	0	0	0
		W								
0x000E	RESERVED CPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
			= Unimplemented or Reserved							

Figure 8-3. CPMU Register Summary

Table 9-9. ADCFLWCTL Field Descriptions

Field	Description
7 SEQA	**Conversion Sequence Abort Event** — This bit indicates that a conversion sequence abort event is in progress. When this bit is set the ongoing conversion sequence and current CSL will be aborted at the next conversion boundary. This bit gets cleared when the ongoing conversion sequence is aborted and ADC is idle. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. *Data Bus Control:* This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 1'b0 does not clear the flag. Writing a one to this bit does not clear it but causes an overrun if the bit has already been set. See Section 9.5.3.2.6, "Conversion flow control in case of conversion sequence control bit overrun scenarios for more details. *Internal Interface Control:* This bit can be controlled via the internal interface Signal "Seq_Abort" if access control is configured accordingly via ACC_CFG[1:0]. After being set an additional request via the internal interface Signal "Seq_Abort" causes an overrun. See also conversion flow control in case of overrun situations. *General:* In both conversion flow control modes (Restart Mode and Trigger Mode) when bit RSTA gets set automatically bit SEQA gets set when the ADC has not reached one of the following scenarios: - A Sequence Abort request is about to be executed or has been executed. - "End Of List" command type has been executed or is about to be executed In case bit SEQA is set automatically the Restart error flag RSTA_EIF is set to indicate an unexpected Restart Request. 0 No conversion sequence abort request. 1 Conversion sequence abort request.
6 TRIG	**Conversion Sequence Trigger Bit** — This bit starts a conversion sequence if set and no conversion or conversion sequence is ongoing. This bit is cleared when the first conversion of a sequence starts to sample. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. *Data Bus Control:* This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 1'b0 does not clear the flag. After being set this bit can not be cleared by writing a value of 1'b1 instead the error flag TRIG_EIF is set. See also Section 9.5.3.2.6, "Conversion flow control in case of conversion sequence control bit overrun scenarios for more details. *Internal Interface Control:* This bit can be controlled via the internal interface Signal "Trigger" if access control is configured accordingly via ACC_CFG[1:0]. After being set an additional request via internal interface Signal "Trigger" causes the flag TRIG_EIF to be set. 0 No conversion sequence trigger. 1 Trigger to start conversion sequence.

Chapter 10
Supply Voltage Sensor - (BATSV3)

Table 10-1. Revision History Table

Rev. No. (Item No.)	Data	Sections Affected	Substantial Change(s)
V01.00	15 Dec 2010	all	Initial Version
V02.00	16 Mar 2011	10.3.2.1 10.4.2.1	- added BVLS[1] to support four voltage level - moved BVHS to register bit 6
V03.00	26 Apr 2011	all	- removed Vsense
V03.10	04 Oct 2011	10.4.2.1 and 10.4.2.2	- removed BSESE

10.1 Introduction

The BATS module provides the functionality to measure the voltage of the chip supply pin VSUP.

10.1.1 Features

The VSUP pin can be routed via an internal divider to the internal Analog to Digital Converter. Independent of the routing to the Analog to Digital Converter, it is possible to route this voltage to a comparator to generate a low or a high voltage interrupt to alert the MCU.

10.1.2 Modes of Operation

The BATS module behaves as follows in the system power modes:

1. Run mode

   The activation of the VSUP Level Sense Enable (BSUSE=1) or ADC connection Enable (BSUAE=1) closes the path from VSUP pin through the resistor chain to ground and enables the associated features if selected.

2. Stop mode

   During stop mode operation the path from the VSUP pin through the resistor chain to ground is opened and the low and high voltage sense features are disabled.
   The content of the configuration register is unchanged.

Table 13-7. PWMPRCLK Field Descriptions

Field	Description
6–4 PCKB[2:0]	**Prescaler Select for Clock B** — Clock B is one of two clock sources which can be used for all channels. These three bits determine the rate of clock B, as shown in Table 13-8.
2–0 PCKA[2:0]	**Prescaler Select for Clock A** — Clock A is one of two clock sources which can be used for all channels. These three bits determine the rate of clock A, as shown in Table 13-8.

Table 13-8. Clock A or Clock B Prescaler Selects

PCKA/B2	PCKA/B1	PCKA/B0	Value of Clock A/B
0	0	0	PWM clock
0	0	1	PWM clock / 2
0	1	0	PWM clock / 4
0	1	1	PWM clock / 8
1	0	0	PWM clock / 16
1	0	1	PWM clock / 32
1	1	0	PWM clock / 64
1	1	1	PWM clock / 128

13.3.2.5 PWM Center Align Enable Register (PWMCAE)

The PWMCAE register contains eight control bits for the selection of center aligned outputs or left aligned outputs for each PWM channel. If the CAEx bit is set to a one, the corresponding PWM output will be center aligned. If the CAEx bit is cleared, the corresponding PWM output will be left aligned. See Section 13.4.2.5, “Left Aligned Outputs” and Section 13.4.2.6, “Center Aligned Outputs” for a more detailed description of the PWM output modes.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R W	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
Reset	0	0	0	0	0	0	0	0

Figure 13-7. PWM Center Align Enable Register (PWMCAE)

Read: Anytime

Write: Anytime

NOTE

Write these bits only when the corresponding channel is disabled.

When the transmit shift register is not transmitting a frame, the TXD pin goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

If software clears TE while a transmission is in progress (TC = 0), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first byte of the second message to SCIDRH/L.

14.4.5.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCICR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCICR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

The SCI recognizes a break character when there are 10 or 11(M = 0 or M = 1) consecutive zero received. Depending if the break detect feature is enabled or not receiving a break character has these effects on SCI registers.

If the break detect feature is disabled (BKDFE = 0):

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see 3.4.4 and 3.4.5 SCI Status Register 1 and 2)

If the break detect feature is enabled (BKDFE = 1) there are two scenarios[1]

The break is detected right from a start bit or is detected during a byte reception.

- Sets the break detect interrupt flag, BKDIF
- Does not change the data register full flag, RDRF or overrun flag OR
- Does not change the framing error flag FE, parity error flag PE.
- Does not clear the SCI data registers (SCIDRH/L)
- May set noise flag NF, or receiver active flag RAF.

1. A Break character in this context are either 10 or 11 consecutive zero received bits

Table 14-20. SCI Interrupt Sources

RXEDGIF	SCIASR1[7]	RXEDGIE	Active high level. Indicates that an active edge (falling for RXPOL = 0, rising for RXPOL = 1) was detected.
BERRIF	SCIASR1[1]	BERRIE	Active high level. Indicates that a mismatch between transmitted and received data in a single wire application has happened.
BKDIF	SCIASR1[0]	BRKDIE	Active high level. Indicates that a break character has been received.

15.3.2.3 SPI Baud Rate Register (SPIBR)

Module Base +0x0002

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-5. SPI Baud Rate Register (SPIBR)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 15-5. SPIBR Field Descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 15-6. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 15-6. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 15-1}$$

The baud rate can be calculated with the following equation:

$$\text{Baud Rate} = \text{BusClock} / \text{BaudRateDivisor} \qquad \textit{Eqn. 15-2}$$

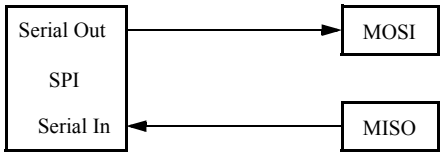
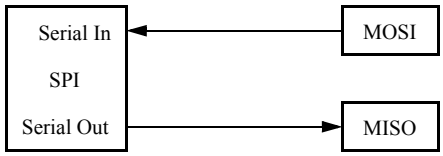
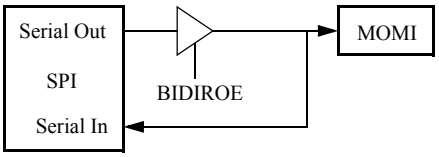
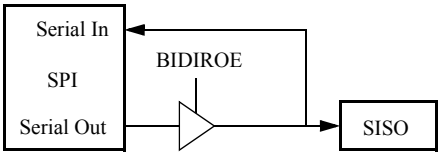
NOTE

For maximum allowed baud rates, please refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

Table 15-6. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 1 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s

Table 15-10. Normal Mode and Bidirectional Mode

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

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

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

NOTE

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

15.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

15.4.6.1 Mode Fault Error

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

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

Table 17-7. CPIE Register Field Descriptions

Field	Description
4 CPVFIE	**CAN Physical Layer Voltage-Failure Interrupt Enable** If enabled, the CAN Physical Layer generates an interrupt if any of the CAN Physical Layer voltage failure interrupt flags assert. 0 Voltage failure interrupt is disabled 1 Voltage failure interrupt is enabled
3 CPDTIE	**CPTXD-Dominant Timeout Interrupt Enable** If enabled, the CAN Physical Layer generates an interrupt if the CPTXD-dominant timeout interrupt flag asserts. 0 CPTXD-dominant timeout interrupt is disabled 1 CPTXD-dominant timeout interrupt is enabled
0 CPOCIE	**CAN Physical Layer Over-current Interrupt Enable** If enabled, the CAN Physical Layer generates an interrupt if any of the CAN Physical Layer over-current interrupt flags assert. 0 Over-current interrupt is disabled 1 Over-current interrupt is enabled

17.4.2.8 CAN Physical Layer Interrupt Flag Register (CPIF)

Module Base + 0x0007 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	CHVHIF	CHVLIF	CLVHIF	CLVLIF	CPDTIF	0	CHOCIF	CLOCIF
W								
Reset	0	0	0	0	0	0	0	0

Figure 17-9. CAN Physical Layer Interrupt Flag Register (CPIF)

[1] Read: Anytime
Write: Anytime, write 1 to clear

If any of the flags is asserted an error interrupt is pending if enabled. A flag can be cleared by writing a logic level 1 to the corresponding bit location. Writing a 0 has no effect.

Table 17-8. CPIF Register Field Descriptions

Field	Description
7 CHVHIF	**CANH Voltage Failure High Interrupt Flag** This flag is set to 1 when the CPCHVH bit in the CAN Physical Layer Status Register (CPSR) changes. 0 No change in CPCHVH 1 CPCHVH has changed
6 CHVLIF	**CANH Voltage Failure Low Interrupt Flag** This flag is set to 1 when the CPCHVL bit in the CAN Physical Layer Status Register (CPSR) changes. 0 No change in CPCHVL 1 CPCHVL has changed

Figure 18-24. Receive/Transmit Message Buffer — Extended Identifier Mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit0
0x00X0 IDR0	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	W								
0x00X1 IDR1	R	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
	W								
0x00X2 IDR2	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	W								
0x00X3 IDR3	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	W								
0x00X4 DSR0	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X5 DSR1	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X6 DSR2	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X7 DSR3	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X8 DSR4	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X9 DSR5	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XA DSR6	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XB DSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XC DLR	R					DLC3	DLC2	DLC1	DLC0
	W								

N.24 0x0990-0x0997 CANPHY

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0990	CPDR	R	CPDR7	0	0	0	0	0	CPDR1	CPDR0
		W								
0x0991	CPCR	R	CPE	SPE	WUPE1-0		0	SLR2-0		
		W								
0x0992	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0993	CPSR	R	CPCHVH	CPCHVL	CPCLVH	CPCLVL	CPDT	0	0	0
		W								
0x0994	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0995	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0996	CPIE	R	0	0	0	CPVFIE	CPDTIE	0	0	CPOCIE
		W								
0x0997	CPIF	R	CHVHIF	CHVLIF	CLVHIF	CLVLIF	CPDTIF	0	CHOCIF	CLOCIF
		W								

= Unimplemented or Reserved

N.25 0x09A0-0x09AF SENTTX

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x09A0	STTICKRATE	R	0	0	PRE[13:8]					
		W								
		R	PRE[7:0]							
		W								
0x09A2	STPPULSE	R	PPEN	PPFIXED	0	0	0	PPCOUNT[10:8]		
		W								
		R	PPCOUNT[7:0]							
		W								
0x09A4	STCONFIG	R	TXINIT	TXEN	0	0	0	DNIBBLECOUNT[2:0]		
		W								
		R	0	0	0	OPTEDGE	SINGLE	CRCSCN	CRCLEG	CRCBYP
		W								
0x09A6	STINTEN	R	0	0	0	PPREIE	TUIE	CSIE	TCIE	TBEIE
		W								

= Unimplemented or Reserved