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	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at32uc3l032-aut

The registers can also be manually reset by writing a one to the Channel Reset bit in the PCONTROL register (PCONTROL.CH0/1RES). The Performance Channel Read/Write Latency registers (PRLAT0/1 and PWLAT0/1) are saturating when their maximum count value is reached. The PRLAT0/1 and PWLAT0/1 registers can only be reset by writing a one to the corresponding reset bit in PCONTROL (PCONTROL.CH0/1RES).

A counter is enabled by writing a one to the Channel Enable bit in the Performance Control Register (PCONTROL.CH0/1EN).

7.7.22 Performance Channel 0 Write Stall Cycles

Name: PWSTALL0

Access Type: Read-only

Offset: 0x814

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
STALL[31:24]							

23	22	21	20	19	18	17	16
STALL[23:16]							

15	14	13	12	11	10	9	8
STALL[15:8]							

7	6	5	4	3	2	1	0
STALL[7:0]							

- **STALL: Stall Cycles Counted Since Last Reset**
  Clock cycles are counted using the CLK_PDCA_HSB clock

To prevent further modifications by software, the content of the register can be set as read-only by writing a one to the Store Final Value bit (SM33.SFV). When this bit is one, software can not change the SM33 register content until the device is reset.

13.5.8.2 *Operating modes*

The SM33 is disabled by default and is enabled by writing to the Supply Monitor Control field in the SM33 control register (SM33.CTRL). The current state of the SM33 can be read from the Supply Monitor On Indicator bit in SM33 (SM33.ONSM). Enabling the SM33 will disable the POR33 detector.

The SM33 can operate in continuous mode or in sampling mode. In sampling mode, the SM33 is periodically enabled for a short period of time, just enough to make a a measurement, and then disabled for a longer time to reduce power consumption.

By default, the SM33 operates in sampling mode during DeepStop and Static mode and in continuous mode for other sleep modes. Sampling mode can also be forced during sleep modes other than DeepStop and Static, and during normal operation, by writing a one to the Force Sampling Mode bit in the SM33 register (SM33.FS).

The user can select the sampling frequency by writing to the Sampling Frequency field in SM33 (SM33.SAMPFREQ). The sampling mode uses the 32kHz RC oscillator (RC32K) as clock source. The 32kHz RC oscillator is automatically enabled when the SM33 operates in sampling mode.

13.5.8.3 *Interrupt and reset generation*

If the SM33 is enabled (SM33.CTRL is one or two) and the regulator supply voltage drops below the SM33 threshold, the SM33DET bit in the Power and Clocks Status Register (PCLKSR.SM33DET) is set. This bit is cleared when the supply voltage goes above the threshold. An interrupt request is generated on a zer-to-one transition of PCLKSR.SM33DET if the Supply Monitor 3.3V Detection bit in the Interrupt Mask Register (IMR.SM33DET) is set. This bit is set by writing a one to the corresponding bit in the Interrupt Enable Register (IER.SM33DET).

If SM33.CTRL is one, a POR will be generated when the voltage drops below the threshold. If SM33.CTRL is two, the device will not be reset.

13.5.8.4 *Factory calibration*

After a reset the SM33.CALIB field is loaded with a factory defined value. This value is chosen so that the nominal threshold value is 1.75V. The flash calibration is redone after any reset, and the Flash Calibration Done bit in SM33 (SM33.FCD) is set before program execution starts in the CPU.

Although it is not recommended to override default factory settings, it is still possible to override the default value by writing to SM33.CALIB

13.5.9 Temperature Sensor

Rev: 1.0.0.0

The Temperature Sensor is connected to an ADC channel, please refer to the ADC chapter for details. It is enabled by writing one to the Enable bit (EN) in the Temperature Sensor Configuration Register (TSENS). The Temperature Sensor can not be calibrated.

Please refer to the Electrical Characteristics chapter for more details.

13.5.10 120MHz RC Oscillator (RC120M)

Rev: 1.0.1.0

The 120MHz RC Oscillator can be used as source for the main clock in the device, as described in the Power Manager chapter. The oscillator can also be used as source for the generic clocks, as described in Generic Clock section. The RC120M must be enabled before it is used as a source clock. To enable the clock, the user must write a one to the Enable bit in the 120MHz RC Oscillator Control Register (RC120MCR.EN), and read back the RC120MCR register until the EN bit reads one. The clock is disabled by writing a zero to RC120MCR.EN. The EN bit must be read back as zero before the RC120M is re-enabled. If not, undefined behavior may occur.

The oscillator is automatically disabled in certain sleep modes to reduce power consumption, as described in the Power Manager chapter.

13.5.11 Backup Registers (BR)

Rev: 1.0.0.1

Four 32-bit backup registers are available to store values when the device is in Shutdown mode. These registers will keep their content even when the VDDCORE supply and the internal regulator supply voltage supplies are removed. The backup registers can be accessed by reading from and writing to the BR0, BR1, BR2, and BR3 registers.

After writing to one of the backup registers the user must wait until the Backup Register Interface Ready bit in tne Power and Clocks Status Register (PCLKSR.BRIFARDY) is set before writing to another backup register. Writes to the backup register while PCLKSR.BRIFARDY is zero will be discarded. An interrupt can be generated on a zero-to-one transition on PCLKSR.BRIFARDY if the BRIFARDY bit in the Interrupt Mask Register (IMR.BRIFARDY) is set. This bit is set by writing a one to the corresponding bit in the Interrupt Enable Register (IER.BRIFARDY).

After powering up the device the Backup Register Interface Valid bit in PCLKSR (PCLKSR.BRIFAVALID) is cleared, indicating that the content of the backup registers has not been written and contains the reset value. After writing to one of the backup registers the PCLKSR.BRIFAVALID bit is set. During writes to the backup registers (when BRIFARDY is zero) BRIFAVALID will be zero. If a reset occurs when BRIFARDY is zero, BRIFAVALID will be cleared after the reset, indicating that the content of the backup registers is not valid. If BRIFARDY is one when a reset occurs, BRIFAVALID will be one and the content is the same as before the reset.

The user must ensure that BRIFAVALID and BRIFARDY are both set before reading the backup register values.

13.5.12 32kHz RC Oscillator (RC32K)

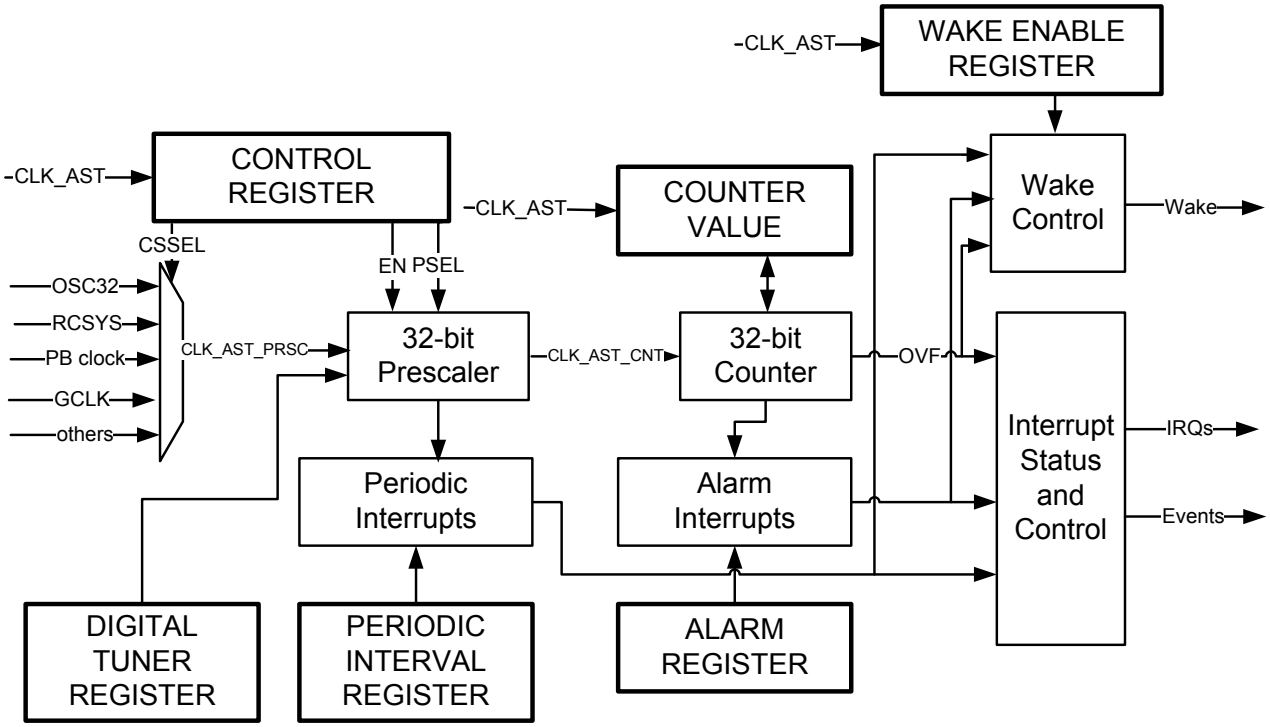
Rev: 1.0.0.0

The RC32K can be used as source for the generic clocks, as described in The Generic Clocks section.

The 32kHz RC oscillator (RC32K) is forced on after reset, and output on PA20. The clock is available on the pad until the PPCR.FRC32 bit in the Power Manager has been cleared or a different peripheral function has been chosen on PA20 (PA20 will start with peripheral function F by default). Note that the forcing will only enable the clock output. To be able to use the RC32K normally the oscillator must be enabled as described below.

14.3 Block Diagram

Figure 14-1. Asynchronous Timer Block Diagram



14.4 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

14.4.1 Power Management

When the AST is enabled, it will remain clocked as long as its selected clock source is running. It can also wake the CPU from the currently active sleep mode. Refer to the Power Manager chapter for details on the different sleep modes.

14.4.2 Clocks

The clock for the AST bus interface (CLK_AST) is generated by the Power Manager. This clock is turned on by default, and can be enabled and disabled in the Power Manager.

A number of clocks can be selected as source for the internal prescaler clock CLK_AST_PRSC. The prescaler, counter, and interrupt will function as long as this selected clock source is active. The selected clock must be enabled in the System Control Interface (SCIF).

The following clock sources are available:

- System RC oscillator (RCSYS). This oscillator is always enabled, except in some sleep modes. Please refer to the Electrical Characteristics chapter for the characteristic frequency of this oscillator.
- 32KHz crystal oscillator (OSC32K). This oscillator must be enabled before use.
- Peripheral Bus clock (PB clock). This is the clock of the peripheral bus the AST is connected to.

- PARDIS=0: During header transmission, the parity bits are computed and in the shift register they replace bits six and seven from IDCHR. During header reception, the parity bits are checked and can generate a LIN Identifier Parity Error (see Section 19.6.6). Bits six and seven in IDCHR read as zero when receiving.
- PARDIS=1: During header transmission, all the bits in IDCHR are sent on the bus. During header reception, all the bits in IDCHR are updated with the received Identifier.

19.6.5.9 Node Action

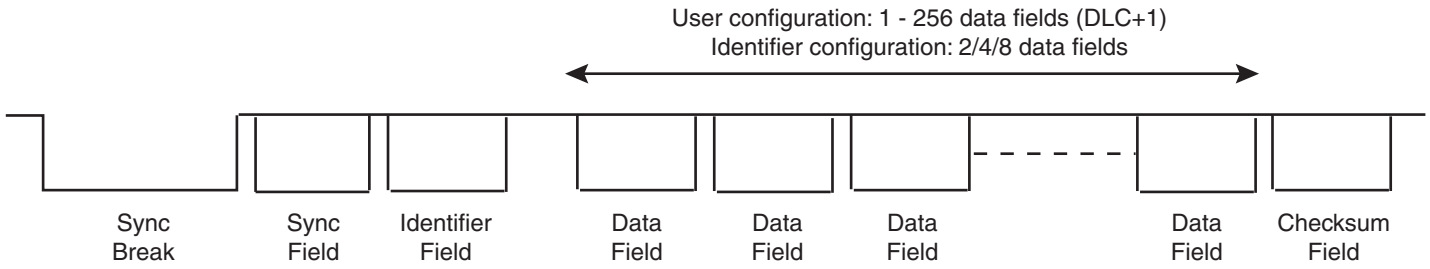
After an identifier transaction, a LIN response mode has to be selected. This is done in the Node Action field (LINMR.NACT). Below are some response modes exemplified in a small LIN cluster:

- Response, from master to slave1:
  - Master: NACT=PUBLISH
  - Slave1: NACT=SUBSCRIBE
  - Slave2: NACT=IGNORE
- Response, from slave1 to master:
  - Master: NACT=SUBSCRIBE
  - Slave1: NACT=PUBLISH
  - Slave2: NACT=IGNORE
- Response, from slave1 to slave2:
  - Master: NACT=IGNORE
  - Slave1: NACT=PUBLISH
  - Slave2: NACT=SUBSCRIBE

19.6.5.10 LIN Response Data Length

The response data length is the number of data fields (bytes), excluding the checksum.

Figure 19-25. Response Data Length



The response data length can be configured, either by the user, or automatically by bits 4 and 5 in the Identifier (IDCHR), in accordance to LIN 1.1. The user selects mode by writing to the Data Length Mode bit (LINMR.DML):

- DLM=0: the response data length is configured by the user by writing to the 8-bit Data Length Control field (LINMR.DLC). The response data length equals DLC + 1 bytes.

19.7.1 Control Register

Name: CR

Access Type: Write-only

Offset: 0x0

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	LINWKUP	LINABT	RTSDIS/RCS	RTSEN/FCS	–	–
15	14	13	12	11	10	9	8
RETTO	RSTNACK	–	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

- **LINWKUP: Send LIN Wakeup Signal**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will sends a wakeup signal on the LIN bus.
- **LINABT: Abort LIN Transmission**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will abort the current LIN transmission.
- **RTSDIS/RCS: Request to Send Disable/Release SPI Chip Select**
  Writing a zero to this bit has no effect.
  Writing a one to this bit when USART is not in SPI master mode drives RTS pin high.
  Writing a one to this bit when USART is in SPI master mode releases NSS (RTS pin).
- **RTSEN/FCS: Request to Send Enable/Force SPI Chip Select**
  Writing a zero to this bit has no effect.
  Writing a one to this bit when USART is not in SPI master mode drives RTS low.
  Writing a one to this bit when USART is in SPI master mode when;
  FCS=0: has no effect.
  FCS=1: forces NSS (RTS pin) low, even if USART is not transmitting, in order to address SPI slave devices supporting the CSAAT Mode (Chip Select Active After Transfer).
- **RETTO: Rearm Time-out**
  Writing a zero to this bit has no effect.
  Writing a one to this bit reloads the time-out counter and clears CSR.TIMEOUT.
- **RSTNACK: Reset Non Acknowledge**
  Writing a zero to this bit has no effect.
  Writing a one to this bit clears CSR.NACK.
- **SENDA: Send Address**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will in multidrop mode send the next character written to THR as an address.
- **STTTO: Start Time-out**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will abort any current time-out count down, and trigger a new count down when the next character has been received. CSR.TIMEOUT is also cleared.

1: The Buffer Full signal from the Peripheral DMA Controller channel is active.

- **ITER/UNRE: Max number of Repetitions Reached or SPI Underrun Error**
  If USART does not operate in SPI Slave Mode:
  ITER=0: Maximum number of repetitions has not been reached since the last RSTSTA.
  ITER=1: Maximum number of repetitions has been reached since the last RSTSTA.
  If USART operates in SPI Slave Mode:
  UNRE=0: No SPI underrun error has occurred since the last RSTSTA.
  UNRE=1: At least one SPI underrun error has occurred since the last RSTSTA.
- **TXEMPTY: Transmitter Empty**
  0: The transmitter is either disabled or there are characters in THR, or in the transmit shift register.
  1: There are no characters in neither THR, nor in the transmit shift register.
- **TIMEOUT: Receiver Time-out**
  0: There has not been a time-out since the last Start Time-out command (CR.STTTO), or RTOR.TO is zero.
  1: There has been a time-out since the last Start Time-out command.
- **PARE: Parity Error**
  0: Either no parity error has been detected, or the parity bit is a zero in multidrop mode, since the last RSTSTA.
  1: Either at least one parity error has been detected, or the parity bit is a one in multidrop mode, since the last RSTSTA.
- **FRAME: Framing Error**
  0: No stop bit has been found as low since the last RSTSTA.
  1: At least one stop bit has been found as low since the last RSTSTA.
- **OVRE: Overrun Error**
  0: No overrun error has occurred since the last RSTSTA.
  1: At least one overrun error has occurred since the last RSTSTA.
- **RXBRK: Break Received/End of Break**
  0: No Break received or End of Break detected since the last RSTSTA.
  1: Break received or End of Break detected since the last RSTSTA.
- **TXRDY: Transmitter Ready**
  0: The transmitter is either disabled, or a character in THR is waiting to be transferred to the transmit shift register, or an STTBRK command has been requested. As soon as the transmitter is enabled, TXRDY becomes one.
  1: There is no character in the THR.
- **RXRDY: Receiver Ready**
  0: The receiver is either disabled, or no complete character has been received since the last read of RHR. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
  1: At least one complete character has been received and RHR has not yet been read.

20.8.11 Chip Select Register 2

Name: CSR2

Access Type: Read/Write

Offset: 0x38

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
DLYBCT							

23	22	21	20	19	18	17	16
DLYBS							

15	14	13	12	11	10	9	8
SCBR							

7	6	5	4	3	2	1	0
BITS				CSAAT	CSNAAT	NCPHA	CPOL

- **DLYBCT: Delay Between Consecutive Transfers**

  This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.

  When DLYBCT equals zero, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.

  Otherwise, the following equation determines the delay:

  $$\text{Delay Between Consecutive Transfers} = \frac{32 \times DLYBCT}{CLKSPI}$$

- **DLYBS: Delay Before SPCK**

  This field defines the delay from NPCS valid to the first valid SPCK transition.

  When DLYBS equals zero, the NPCS valid to SPCK transition is 1/2 the SPCK clock period.

  Otherwise, the following equations determine the delay:

  $$\text{Delay Before SPCK} = \frac{DLYBS}{CLKSPI}$$

- **SCBR: Serial Clock Baud Rate**

  In Master Mode, the SPI Interface uses a modulus counter to derive the SPCK baud rate from the CLK_SPI. The Baud rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK baud rate:

  $$\text{SPCK Baudrate} = \frac{CLKSPI}{SCBR}$$

  Writing the SCBR field to zero is forbidden. Triggering a transfer while SCBR is zero can lead to unpredictable results.

  At reset, SCBR is zero and the user has to write it to a valid value before performing the first transfer.

  If a clock divider (SCBRn) field is set to one and the other SCBR fields differ from one, access on CSn is correct but no correct access will be possible on other CS.

24. Timer/Counter (TC)

Rev: 2.2.3.1.3

24.1 Features

- **Three 16-bit Timer Counter channels**
- **A wide range of functions including:**
  - **Frequency measurement**
  - **Event counting**
  - **Interval measurement**
  - **Pulse generation**
  - **Delay timing**
  - **Pulse width modulation**
  - **Up/down capabilities**
- **Each channel is user-configurable and contains:**
  - **Three external clock inputs**
  - **Five internal clock inputs**
  - **Two multi-purpose input/output signals**
- **Internal interrupt signal**
- **Two global registers that act on all three TC channels**
- **Peripheral event input on all A lines in capture mode**

24.2 Overview

The Timer Counter (TC) includes three identical 16-bit Timer Counter channels.

Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing, and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs, and two multi-purpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC block has two global registers which act upon all three TC channels.

The Block Control Register (BCR) allows the three channels to be started simultaneously with the same instruction.

The Block Mode Register (BMR) defines the external clock inputs for each channel, allowing them to be chained.

26.8 Operating Modes

The ADCIFB features two operating modes, each defining a separate conversion sequence:

- ADC Mode: At each trigger, all the enabled channels are converted.
- Resistive Touch Screen Mode: At each trigger, all enabled channels plus the resistive touch screen channels are converted as described in Section 26.8.3. If channels except the dedicated resistive touch screen channels are enabled, they are converted normally before and after the resistive touch screen channels are converted.

The operating mode is selected by the TSAMOD field in the Mode Register (MR).

26.8.1 Conversion Triggers

A conversion sequence is started either by a software or by a hardware trigger. When a conversion sequence is started, all enabled channels will be converted and made available in the shared Last Converted Register (LCDR).

The software trigger is asserted by writing a one to the START field in the Control Register (CR).

The hardware trigger can be selected by the TRGMOD field in the Trigger Register (TRGR). Different hardware triggers exist:

- External trigger, either rising or falling or any, detected on the external trigger pin TRIGGER
- Pen detect trigger, depending the PENDET bit in the Mode Register (MR)
- Continuous trigger, meaning the ADCIFB restarts the next sequence as soon as it finishes the current one
- Periodic trigger, which is defined by the TRGR.TRGPER field
- Peripheral event trigger, allowing the Peripheral Event System to synchronize conversion with some configured peripheral event source.

Enabling a hardware trigger does not disable the software trigger functionality. Thus, if a hardware trigger is selected, the start of a conversion can still be initiated by the software trigger.

26.8.2 ADC Mode

In the ADC Mode, the active channels are defined by the Channel Status Register (CHSR). A channel is enabled by writing a one to the corresponding bit in the Channel Enable Register (CHER), and disabled by writing a one to the corresponding bit in the Channel Disable Register (CHDR). The conversion results are stored in the Last Converted Data Register (LCDR) as they become available, overwriting old conversions.

At each trigger, the following sequence is performed:

1. If ACR.SLEEP is one, wake up the ADC and wait for the startup time.
2. If Channel 0 is enabled, convert Channel 0 and store result in LCDR.
3. If Channel 1 is enabled, convert Channel 1 and store result in LCDR.
4. If Channel N is enabled, convert Channel N and store result in LCDR.
5. If ACR.SLEEP is one, place the ADC cell in a low-power state.

If the Peripheral DMA Controller is enabled, all converted values are transferred continuously into the memory buffer.

26.8.3 Resistive Touch Screen Mode

Writing a one to the TSAMOD field in the Mode Register (MR) enables Resistive Touch Screen Mode. In this mode the channels TSPO+0 to TSPO+3, corresponding to the resistive touch

26.9.17 Channel Status Register

Name: CHSR

Access Type: Read-only

Offset: 0x48

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
23	22	21	20	19	18	17	16
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHn: Channel N Status**

  0: The corresponding channel is disabled.

  1: The corresponding channel is enabled.

  A bit in this register is cleared by writing a one to the corresponding bit in Channel Disable Register (CHDR).

  A bit in this register is set by writing a one to the corresponding bit in Channel Enable Register (CHER).

  The number of available channels is device dependent. Please refer to the Module Configuration section at the end of this chapter for information regarding how many channels are implemented.

Table 31-1. SAB Slaves, Addresses and Descriptions

Slave	Address [35:32]	Description
HSB	0x5	Alternative mapping for HSB space, for compatibility with other 32-bit AVR devices.
Memory Service Unit	0x6	Memory Service Unit registers
Reserved	Other	Unused

31.2.2 SAB Security Restrictions

The Service Access bus can be restricted by internal security measures. A short description of the security measures are found in the table below.

31.2.2.1 Security measure and control location

A security measure is a mechanism to either block or allow SAB access to a certain address or address range. A security measure is enabled or disabled by one or several control signals. This is called the control location for the security measure.

These security measures can be used to prevent an end user from reading out the code programmed in the flash, for instance.

Table 31-2. SAB Security Measures

Security Measure	Control Location	Description
Secure mode	FLASHCDW SECURE bits set	Allocates a portion of the flash for secure code. This code cannot be read or debugged. The User page is also locked.
Security bit	FLASHCDW security bit set	Programming and debugging not possible, very restricted access.
User code programming	FLASHCDW UPROT + security bit set	Restricts all access except parts of the flash and the flash controller for programming user code. Debugging is not possible unless an OS running from the secure part of the flash supports it.

Below follows a more in depth description of what locations are accessible when the security measures are active.

Table 31-3. Secure Mode SAB Restrictions

Name	Address Start	Address End	Access
Secure flash area	0x580000000	0x580000000 + (USERPAGE[15:0] << 10)	Blocked
Secure RAM area	0x500000000	0x500000000 + (USERPAGE[31:16] << 10)	Blocked
User page	0x580800000	0x581000000	Read
Other accesses	-	-	As normal

Note: 1. Second Word of the User Page, refer to the Fuses Settings section for details.

1. 0x55 (sync)
2. 0xC1 (command)
3. 0x00 (length MSB)
4. 0x07 (length LSB)
5. 0xCA (Data MSB)
6. 0xFE
7. 0xBA
8. 0xBE (Data LSB)
9. 0x00 (Status byte)
10. 0x00 (Bytes remaining MSB)
11. 0x00 (Bytes remaining LSB)
12. 0xXX (CRC MSB)
13. 0xXX (CRC LSB)

The status is 0x00 and all data read are valid. An unsuccessful four byte read can look like this:

1. 0x55 (sync)
2. 0xC1 (command)
3. 0x00 (length MSB)
4. 0x07 (length LSB)
5. 0xCA (Data MSB)
6. 0xFE
7. 0xXX (An error has occurred. Data read is undefined. 5 bytes remaining of the Data field)
8. 0xXX (More undefined data)
9. 0x02 (Status byte)
10. 0x00 (Bytes remaining MSB)
11. 0x05 (Bytes remaining LSB)
12. 0xXX (CRC MSB)
13. 0xXX (CRC LSB)

The error occurred after reading 2 bytes on the SAB. The rest of the bytes read are undefined. The status byte indicates the error and the bytes remaining indicates how many bytes were remaining to be sent of the data field of the packet when the error occurred.

Table 31-53. MEMDATA Status Byte

status byte	Description
0x00	Read successful
0x01	SAB busy
0x02	Bus error (wrong address)
Other	Reserved

Table 31-54. MEMDATA Details

Response	Details
Response value	0xC1
Additional data	Data read, status byte, and byte count (2 bytes)

31.6.8.5 MEMORY_READWRITE_STATUS

After a MEMORY_WRITE command this response is sent by AW. The response can also be sent after a MEMORY_READ command if AW encountered an error when receiving the address. The response contains 3 bytes, where the first is the status of the command and the 2 next contains the byte count when the first error occurred. The first byte is encoded this way:

Table 31-55. MEMORY_READWRITE_STATUS Status Byte

status byte	Description
0x00	Write successful
0x01	SAB busy
0x02	Bus error (wrong address)
Other	Reserved

Table 31-56. MEMORY_READWRITE_STATUS Details

Response	Details
Response value	0xC2
Additional data	Status byte and byte count (2 bytes)

31.6.8.6 BAUD_RATE

The current baud rate in the AW. See Section 31.6.6.7 for more details.

Table 31-57. BAUD_RATE Details

Response	Details
Response value	0xC3
Additional data	Baud rate

31.6.8.7 STATUS_INFO

A status message from AW.

Table 31-58. STATUS_INFO Contents

Bit number	Name	Description
15-9	Reserved	
8	Protected	The protection bit in the internal flash is set. SAB access is restricted. This bit will read as one during reset.
7	SAB busy	The SAB bus is busy with a previous transfer. This could indicate that the CPU is running on a very slow clock, the CPU clock has stopped for some reason or that the part is in constant reset.
6	Chip erase ongoing	The Chip erase operation has not finished.
5	CPU halted	This bit will be set if the CPU is halted. This bit will read as zero during reset.
4-1	Reserved	
0	Reset status	This bit will be set if AW has reset the CPU using the RESET command.

The maximum SPI slave output frequency is given by the following formula:

$$f_{SPCKMAX} = MIN(\frac{f_{CLKSPI} \times 2}{9}, f_{PINMAX}, \frac{1}{SPIn + t_{SETUP}})$$

Where $SPIn$ is the MISO delay, USPI6 or USPI9 depending on CPOL and NCPHA. T_{SETUP} is the SPI master setup time. Please refer to the SPI master datasheet for T_{SETUP} . f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock. f_{PINMAX} is the maximum frequency of the SPI pins. Please refer to the I/O Pin Characteristics section for the maximum frequency of the pins.

32.9.4 SPI Timing

32.9.4.1 Master mode

Figure 32-13. SPI Master Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

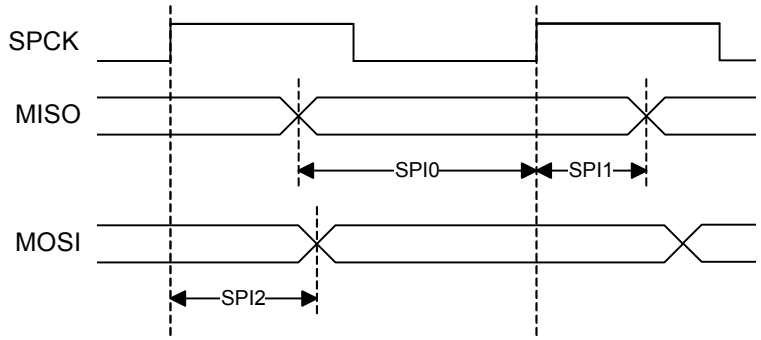


Figure 32-14. SPI Master Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)

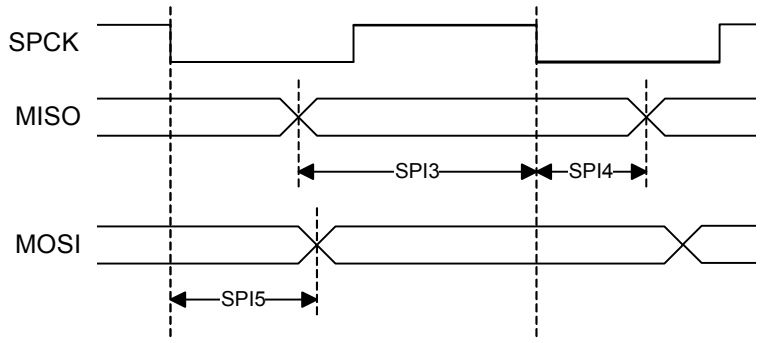


Figure 32-16. SPI Slave Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

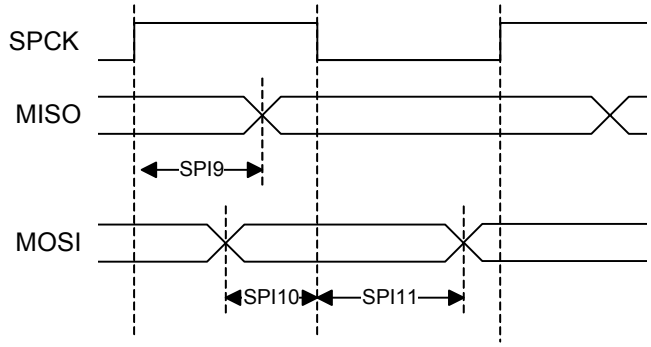


Figure 32-17. SPI Slave Mode NPCS Timing

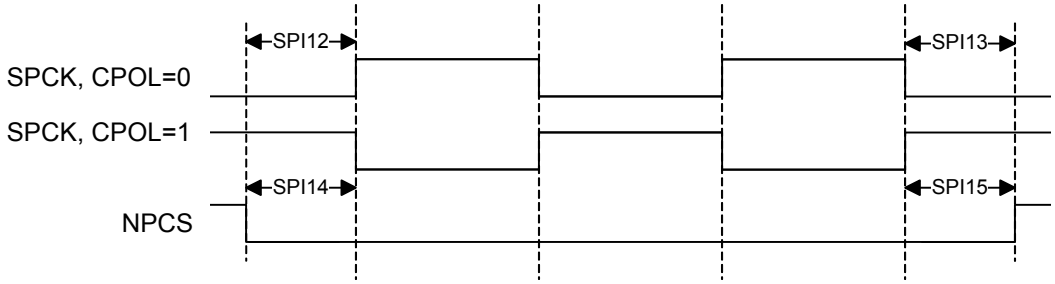


Table 32-41. SPI Timing, Slave Mode[1]

Symbol	Parameter	Conditions	Min	Max	Units
SPI6	SPCK falling to MISO delay	V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF		30.8	ns
SPI7	MOSI setup time before SPCK rises		0		
SPI8	MOSI hold time after SPCK rises		4.1		
SPI9	SPCK rising to MISO delay			29.9	
SPI10	MOSI setup time before SPCK falls		0		
SPI11	MOSI hold time after SPCK falls		3.5		
SPI12	NPCS setup time before SPCK rises		1.9		
SPI13	NPCS hold time after SPCK falls		0.2		
SPI14	NPCS setup time before SPCK falls		2.2		
SPI15	NPCS hold time after SPCK rises		0		

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Maximum SPI Frequency, Slave Input Mode

34. Ordering Information

Table 34-1. Ordering Information

Device	Ordering Code	Carrier Type	Package	Package Type	Temperature Operating Range
AT32UC3L064	AT32UC3L064-AUTES	ES	TQFP 48	JESD97 Classification E3	Industrial (-40°C to 85°C)
	AT32UC3L064-AUT	Tray			
	AT32UC3L064-AUR	Tape & Reel			
	AT32UC3L064-ZAUES	ES	QFN 48		
	AT32UC3L064-ZAUT	Tray			
	AT32UC3L064-ZAUR	Tape & Reel			
	AT32UC3L064-D3HES	ES	TLLGA 48	JESD97 Classification E4	
	AT32UC3L064-D3HT	Tray			
	AT32UC3L064-D3HR	Tape & Reel			
AT32UC3L032	AT32UC3L032-AUT	Tray	TQFP 48	JESD97 Classification E3	
	AT32UC3L032-AUR	Tape & Reel			
	AT32UC3L032-ZAUT	Tray	QFN 48		
	AT32UC3L032-ZAUR	Tape & Reel			
	AT32UC3L032-D3HT	Tray	TLLGA 48	JESD97 Classification E4	
	AT32UC3L032-D3HR	Tape & Reel			
AT32UC3L016	AT32UC3L016-AUT	Tray	TQFP 48	JESD97 Classification E3	
	AT32UC3L016-AUR	Tape & Reel			
	AT32UC3L016-ZAUT	Tray	QFN 48		
	AT32UC3L016-ZAUR	Tape & Reel			
	AT32UC3L016-D3HT	Tray	TLLGA 48	JESD97 Classification E4	
	AT32UC3L016-D3HR	Tape & Reel			

Use twice as long timeout period as needed and clear the WDT counter within the first half of the timeout period. If the WDT counter is cleared after the first half of the timeout period, you will get a Watchdog reset immediately. If the WDT counter is not cleared at all, the time before the reset will be twice as long as needed.

2. **WDT Control Register does not have synchronization feedback**
   When writing to the Timeout Prescale Select (PSEL), Time Ban Prescale Select (TBAN), Enable (EN), or WDT Mode (MODE) fieldss of the WDT Control Register (CTRL), a synchronizer is started to propagate the values to the WDT clcok domain. This synchronization takes a finite amount of time, but only the status of the synchronization of the EN bit is reflected back to the user. Writing to the synchronized fields during synchronization can lead to undefined behavior.
   Fix/Workaround
   -When writing to the affected fields, the user must ensure a wait corresponding to 2 clock cycles of both the WDT peripheral bus clock and the selected WDT clock source.
   -When doing writes that changes the EN bit, the EN bit can be read back until it reflects the written value.

35.1.7 GPIO

1. **Clearing GPIO interrupt may fail**
   Writing a one to the GPIO.IFRC register to clear an interrupt will be ignored if interrupt is enabled for the corresponding port.
   Fix/Workaround
   Disable the interrupt, clear it by writing a one to GPIO.IFRC, then enable the interrupt.

35.1.8 SPI

1. **SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0**
   When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.
   Fix/Workaround
   Disable mode fault detection by writing a one to MR.MODFDIS.

2. **Disabling SPI has no effect on the SR.TDRE bit**
   Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.
   Fix/Workaround
   Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

3. **SPI disable does not work in SLAVE mode**
   SPI disable does not work in SLAVE mode.
   Fix/Workaround
   Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

4. **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
   When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.