**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 | 8-Bit |
| Speed | 16MHz |
| Connectivity | I²C, IrDA, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 6 |
| Program Memory Size | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 64 x 8 |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 6x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 8-SOIC (0.154", 3.90mm Width) |
| Supplier Device Package | 8-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/attiny202-ssfr |

#### 6.10.2.5 OSC16 Error at 5V

**Name:** OSC16ERR5V
**Offset:** 0x23
**Reset:** [Oscillator frequency error value]
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OSC16ERR5V[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

**Bits 7:0 – OSC16ERR5V[7:0]** OSC16 Error at 5V
These registers contain the signed oscillator frequency error value when running at internal 16 MHz at 5V, as measured during production.

### 8.5.6 Accessing 16-Bit Registers

The AVR data bus is 8-bits wide, and so accessing 16-bit registers requires atomic operations. These registers must be byte accessed using two read or write operations. 16-bit registers are connected to the 8-bit bus and a temporary register using a 16-bit bus.

For a write operation, the low byte of the 16-bit register must be written before the high byte. The low byte is then written into the temporary register. When the high byte of the 16-bit register is written, the temporary register is copied into the low byte of the 16-bit register in the same clock cycle.

For a read operation, the low byte of the 16-bit register must be read before the high byte. When the low byte register is read by the CPU, the high byte of the 16-bit register is copied into the temporary register in the same clock cycle as the low byte is read. When the high byte is read, it is then read from the temporary register.

This ensures that the low and high bytes of 16-bit registers are always accessed simultaneously when reading or writing the register.

Interrupts can corrupt the timed sequence if an interrupt is triggered and accesses the same 16-bit register during an atomic 16-bit read/write operation. To prevent this, interrupts can be disabled when writing or reading 16-bit registers.

The temporary registers can be read and written directly from user software.

### 8.5.7 Configuration Change Protection (CCP)

System critical I/O register settings are protected from accidental modification. Flash self-programming (via store to NVM controller) is protected from accidental execution. This is handled globally by the Configuration Change Protection (CCP) register.

Changes to the protected I/O registers or bits, or execution of protected instructions, are only possible after the CPU writes a signature to the CCP register. The different signatures are listed in the description of the CCP register (CPU.CCP).

There are two modes of operation: one for protected I/O registers, and one for the protected self-programming.

**Related Links**

CCP

#### 8.5.7.1 Sequence for Write Operation to Configuration Change Protected I/O Registers

In order to write to registers protected by CCP, these steps are required:

1. The software writes the signature that enables change of protected I/O registers to the CCP bit field in the CPU.CCP register.
2. Within four instructions, the software must write the appropriate data to the protected register. Most protected registers also contain a write enable/change enable/lock bit. This bit must be written to '1' in the same operation as the data are written.

   The protected change is immediately disabled if the CPU performs write operations to the I/O register or data memory, if load or store accesses to Flash, NVMCTRL, EEPROM are conducted, or if the `SLEEP` instruction is executed.

#### 8.5.7.2 Sequence for Execution of Self-Programming

In order to execute self-programming (the execution of writes to the NVM controller's command register), the following steps are required:

### 10.5.4 Main Clock Status

**Name:** MCLKSTATUS
**Offset:** 0x03
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EXTS | | OSC32KS | OSC20MS | | | | SOSC |
| Access | R | | R | R | R | R | R | R |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – EXTS** External Clock Status

| Value | Description |
|---|---|
| 0 | EXTCLK has not started |
| 1 | EXTCLK has started |

**Bit 5 – OSC32KS** OSCULP32K Status
The Status bit will only be available if the source is requested as the main clock or by another module. If the oscillator RUNSTDBY bit is set but the oscillator is unused/not requested, this bit will be 0.

| Value | Description |
|---|---|
| 0 | OSCULP32K is not stable |
| 1 | OSCULP32K is stable |

**Bit 4 – OSC20MS** OSC20M Status
The Status bit will only be available if the source is requested as the main clock or by another module. If the oscillator RUNSTDBY bit is set but the oscillator is unused/not requested, this bit will be 0.

| Value | Description |
|---|---|
| 0 | OSC20M is not stable |
| 1 | OSC20M is stable |

**Bit 0 – SOSC** Main Clock Oscillator Changing

| Value | Description |
|---|---|
| 0 | The clock source for CLK_MAIN is not undergoing a switch |
| 1 | The clock source for CLK_MAIN is undergoing a switch, and will change as soon as the new source is stable. |

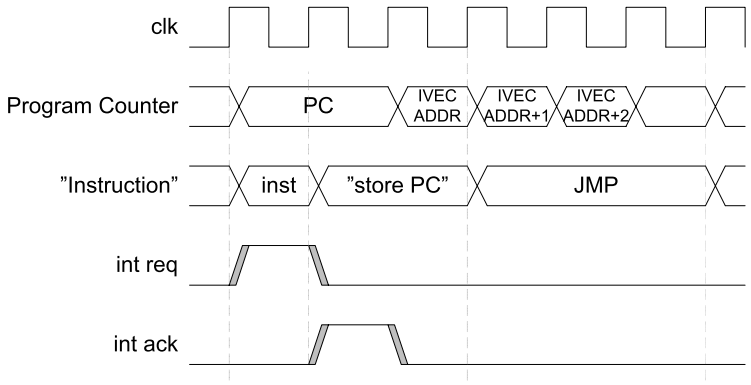
## 12.4 Register Summary - RSTCTRL

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | RSTFR | 7:0 | | | UPDIRF | SWRF | WDRF | EXTRF | BORF | PORF |
| 0x01 | SWRR | 7:0 | | | | | | | | SWRE |

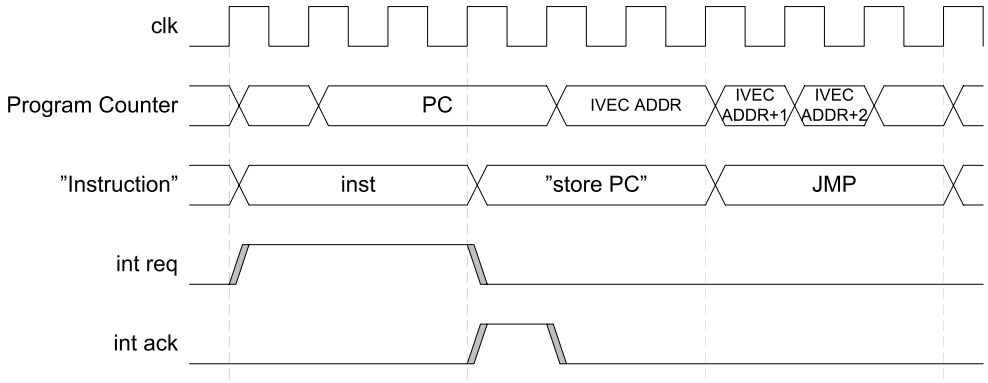
## 12.5 Register Description

**Figure 13-2. Interrupt Execution of a Single Cycle Instruction, Multicycle Instruction, and From Sleep(1)**

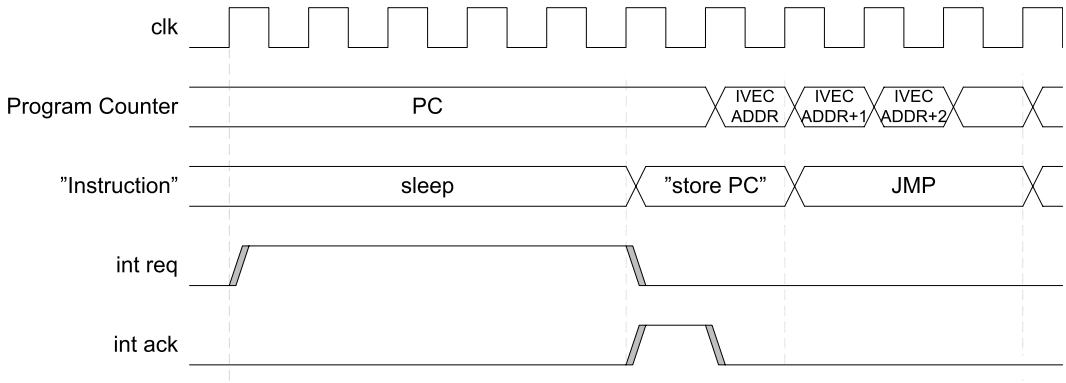
Single Cycle Instruction




Multicycle Instruction




Sleep




**Note:**

1. Devices with 8kB of Flash or less use `RJMP` instead of `JMP`, which takes only two clock cycles.

#### 13.3.2.4 Interrupt Level

The interrupt level is default on level 0 (normal) for all interrupt sources. It is possible to select one interrupt source to level 1 (high) by writing interrupt address to CPUINT.LVL1VEC register. This source will have higher priority than normal level interrupts.

An interrupt request from a level 1 source will interrupt any ongoing interrupt handler from a level 0 interrupt. When returning from the level 1 interrupt handler, the execution of the level 0 interrupt handler will continue.

#### 13.3.2.5 Interrupt Priority

**Non-Maskable Interrupts (NMI)**

An NMI will be executed regardless of the setting of the I bit in CPU.SREG, and it will never change the I bit. No other interrupt can interrupt an NMI handler. If more than one NMI is requested at the same time, priority is static according to the interrupt vector address, where the lowest address has the highest priority.

Which interrupts are non-maskable is device-dependent and not subject to configuration. Non-maskable interrupts must be enabled before they can be used. Refer to the Interrupt Vector Mapping of the device for available NMI lines.

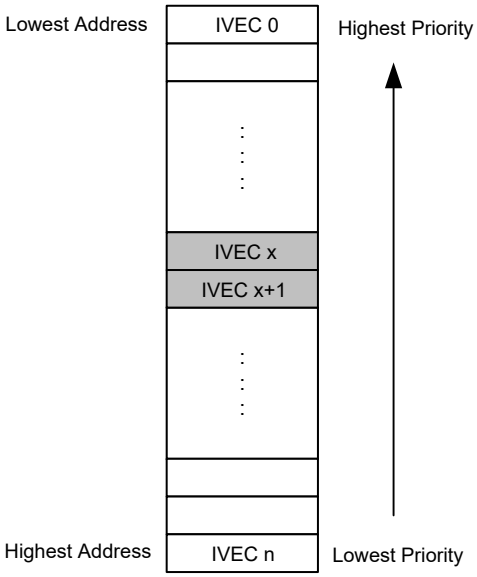
**Related Links**

Interrupt Vector Mapping

**Static Priority**

Interrupt Vectors (IVEC) are located at fixed addresses. For static priority, the interrupt vector address decides the priority within normal interrupt level, where the lowest interrupt vector address has the highest priority. Refer to the Interrupt Vector Mapping of the device for available interrupt lines and their base address offset.

**Figure 13-3. Static Priority**




**Related Links**

Interrupt Vector Mapping

**Round Robin Scheduling**

To avoid "starvation" for priority level 0 (LVL0) interrupt requests with static priority (i.e. some interrupts might never be served), the CPUINT offers round robin scheduling for LVL0 interrupts.

Round robin scheduling for LVL0 interrupt requests is enabled by writing a '1' to the Round Robin Priority Enable bit (LVL0RR) in the Control A register (CPUINT.CTRLA).

### 17.5.2 Control B

**Name:** CTRLB
**Offset:** 0x01
**Reset:** Loaded from fuse
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | LVL[2:0] | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | x | x | x |

**Bits 2:0 – LVL[2:0]** BOD Level
These bits select the BOD threshold level.

The Reset value is loaded from the BOD Level bits (LVL) in the BOD Configuration Fuse (FUSE.BODCFG).

| Value | Name | Description |
|---|---|---|
| 0x0 | BODLEVEL0 | 1.8V |
| 0x1 | BODLEVEL1 | 2.15V |
| 0x2 | BODLEVEL2 | 2.60V |
| 0x3 | BODLEVEL3 | 2.95V |
| 0x4 | BODLEVEL4 | 3.30V |
| 0x5 | BODLEVEL5 | 3.70V |
| 0x6 | BODLEVEL6 | 4.00V |
| 0x7 | BODLEVEL7 | 4.30V |

### 19.5.2 Status

**Name:** STATUS
**Offset:** 0x01
**Reset:** 0x00
**Property:** Configuration Change Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LOCK | | | | | | | SYNCBUSY |
| Access | R/W | | | | | | | R |
| Reset | 0 | | | | | | | 0 |

**Bit 7 – LOCK** Lock
Writing this bit to '1' write protects the WDT.CTRLA register.

It is only possible to write this bit to '1'. This bit can only be cleared in Debug mode.

If the PERIOD bits in WDT.CTRLA are different from zero after boot code, the lock will automatically be set.

This bit is under CCP.

**Bit 0 – SYNCBUSY** Synchronization Busy
This bit is set after writing to the WDT.CTRLA register while the data is being synchronized from the system clock domain to the WDT clock domain.

This bit is cleared by the system after the synchronization is finished.

This bit is not under CCP.

**Related Links**
Synchronization
Configuration Change Protection

### 20.5.15 Period Register - Normal Mode

**Name:** PER
**Offset:** 0x26
**Reset:** 0xFFFF
**Property:** -

TCAn.PER contains the 16-bit TOP value in the timer/counter.

The TCAn.PERL and TCAn.PERH register pair represents the 16-bit value, TCAn.PER. The low byte [7:0] (suffix L) is accessible at the original offset. The high byte [15:8] (suffix H) can be accessed at offset + 0x01.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PER[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 15:8 – PER[15:8]** Periodic High Byte
These bits hold the MSB of the 16-bit period register.

**Bits 7:0 – PER[7:0]** Periodic Low Byte
These bits hold the LSB of the 16-bit period register.

### 20.7.11 High Byte Timer Counter Register - Split Mode

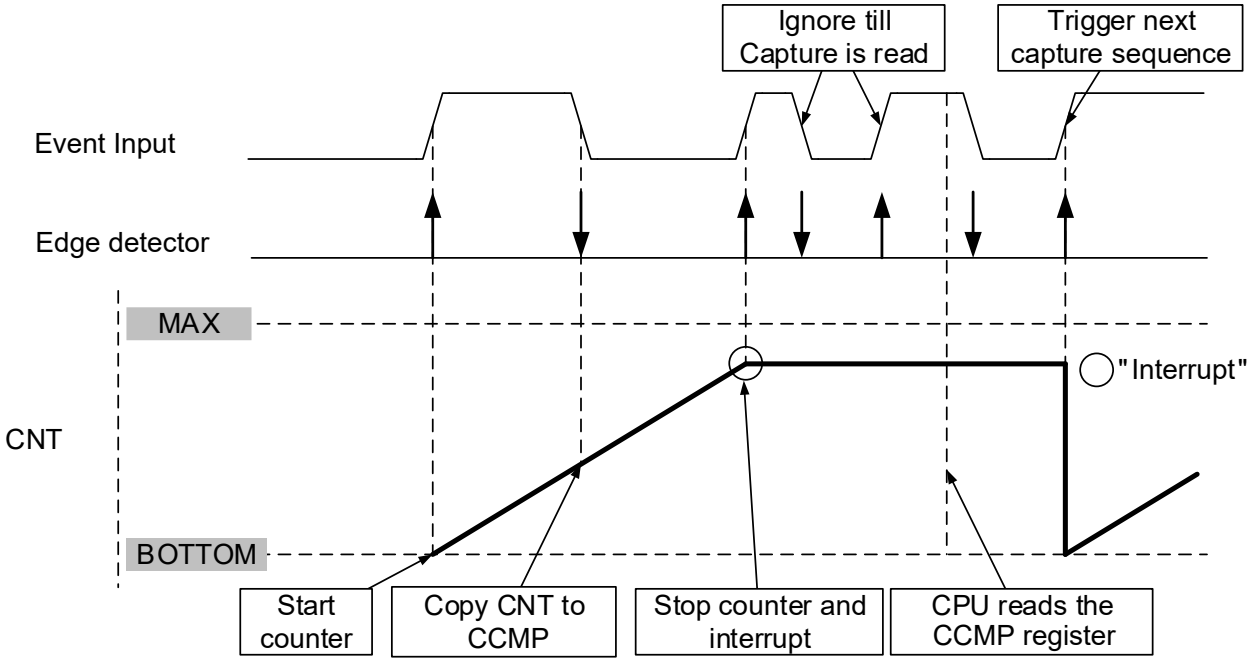
**Name:** HCNT
**Offset:** 0x21
**Reset:** 0x00
**Property:** -

TCAn.HCNT contains the counter value in high byte timer. CPU and UPDI write access has priority over count, clear, or reload of the counter.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HCNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – HCNT[7:0]** Counter Value for High Byte Timer
These bits define the counter value in high byte timer.

**Figure 21-7. Input Capture Frequency and Pulse-Width Measurement**




**Single-Shot Mode**

This mode can be used to generate a pulse with a duration that is defined by the Compare register (TCBn.CCMP), every time a rising or falling edge is observed on a connected event channel.

When the counter is stopped, the output pin is driven to low. If an event is detected on the connected event channel, the timer will reset and start counting from zero to TOP while driving its output high. The RUN bit in the STATUS register can be read to see if the counter is counting or not. When the counter register reaches the CCMP register value, the counter will stop and the output pin will go low for at least one prescaler cycle. If a new event arrives during this time, that event will be ignored. The following figure shows an example waveform. There is a two clock cycle delay from when the event is received until the output is set high. If the ASYNC bit in TCBn.CTRLB is written to '1', an asynchronous edge detector is used for input events to give immediate action. When the EDGE bit of the TCBn.EVCTRL register is written to '1', any edge can trigger the start of the counter. If the EDGE bit is '0', only positive edges will trigger the start.

The counter will start as soon as the module is enabled, even without triggering event. This is prevented by writing TOP to the counter register.

Similar behavior is seen if the EDGE bit in the TCBn.EVCTRL register is '1' while the module is enabled. Writing TOP to the Counter register prevents this as well.

It is not recommended to change configuration while the module is enabled.

### 22.11.11 Periodic Interrupt Timer Control A

**Name:** PITCTRLA
**Offset:** 0x10
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | PERIOD[3:0] | | | | | | PITEN |
| Access | | R/W | R/W | R/W | R/W | | | R/W |
| Reset | | 0 | 0 | 0 | 0 | | | 0 |

**Bits 6:3 – PERIOD[3:0]** Period
Writing this bit field selects the number of RTC clock cycles between each interrupt.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | No interrupt |
| 0x1 | CYC4 | 4 cycles |
| 0x2 | CYC8 | 8 cycles |
| 0x3 | CYC16 | 16 cycles |
| 0x4 | CYC32 | 32 cycles |
| 0x5 | CYC64 | 64 cycles |
| 0x6 | CYC128 | 128 cycles |
| 0x7 | CYC256 | 256 cycles |
| 0x8 | CYC512 | 512 cycles |
| 0x9 | CYC1024 | 1024 cycles |
| 0xA | CYC2048 | 2048 cycles |
| 0xB | CYC4096 | 4096 cycles |
| 0xC | CYC8192 | 8192 cycles |
| 0xD | CYC16384 | 16384 cycles |
| 0xE | CYC32768 | 32768 cycles |
| 0xF | - | Reserved |

**Bit 0 – PITEN** Periodic Interrupt Timer Enable
Writing a '1' to this bit enables the Periodic Interrupt Timer.

### 23.5.3 Transmit Data Register Low Byte

**Name:** TXDATAL
**Offset:** 0x02
**Reset:** 0x00
**Property:** R/W

The Transmit Data Buffer (TXB) register will be the destination for data written to the USARTn.TXDATAL register location.

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the transmitter and set to zero by the receiver.

The transmit buffer can only be written when the DREIF flag in the USARTn.STATUS register is set. Data written to DATA when the DREIF flag is not set will be ignored by the USART transmitter. When data is written to the transmit buffer, and the transmitter is enabled, the transmitter will load the data into the Transmit Shift register when the Shift register is empty. The data is then transmitted on the TxD pin.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – DATA[7:0]** Transmit Data Register

### 23.5.10 Baud Register

**Name:** BAUD
**Offset:** 0x08
**Reset:** 0x00
**Property:** -

The USARTn.BAUDL and USARTn.BAUDH register pair represents the 16-bit value, USARTn.BAUD. The low byte [7:0] (suffix L) is accessible at the original offset. The high byte [15:8] (suffix H) can be accessed at offset + 0x01.

Ongoing transmissions of the transmitter and receiver will be corrupted if the baud rate is changed. Writing this register will trigger an immediate update of the baud rate prescaler. For more information on how to set the baud rate, see Table 23-2.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:8 – BAUD[15:8]** USART Baud Rate High Byte
These bits hold the MSB of the 16-bit Baud register.

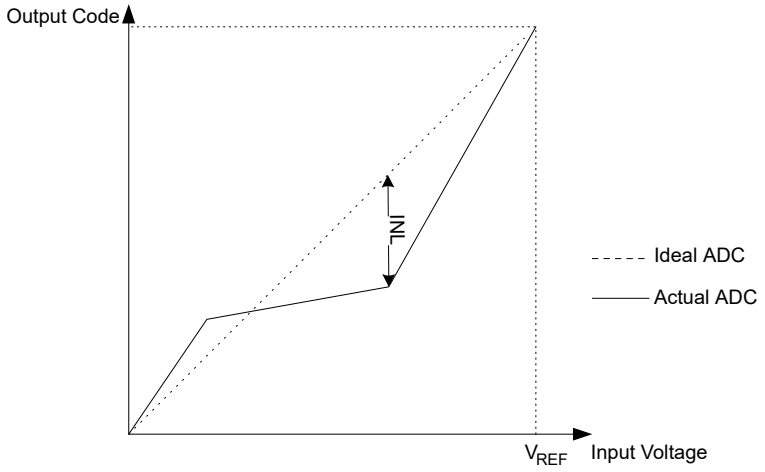
**Bits 7:0 – BAUD[7:0]** USART Baud Rate Low Byte
These bits hold the LSB of the 16-bit Baud register.

If the Clock Source bit (CLKSRC) in the LUT n Control A register (CCL.LUTnCTRLA) is written to '1', the LUT input 2 (IN[2]) will always clock the Filter, Edge Detector, and Sequential block. The availability of the IN[2] clock in sleep modes will depend on the sleep settings of the peripheral employed.

### 27.3.5 Configuration Change Protection

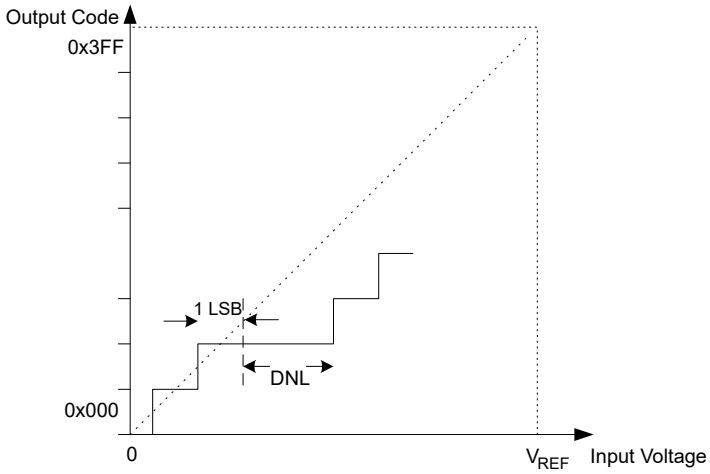
Not applicable.

**Figure 29-4. Integral Non-Linearity**




**Differential Non-Linearity (DNL)** The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

**Figure 29-5. Differential Non-Linearity**




**Quantization Error** Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5 LSB.

**Absolute Accuracy** The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of all aforementioned errors. Ideal value: ±0.5 LSB.

**Related Links**

ADC Characteristics

## 29.3 Functional Description

### 29.3.1 Initialization

The following steps are recommended in order to initialize ADC operation:

1. Configure the resolution by writing to the Resolution Selection bit (RESSEL) in the Control A register (ADC.CTRLA).

## 29.4 Register Summary - ADCn

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTBY | | | | | RESSEL | FREERUN | ENABLE |
| 0x01 | CTRLB | 7:0 | | | | | | SAMPNUM[2:0] | | |
| 0x02 | CTRLC | 7:0 | | SAMPCAP | REFSEL[1:0] | | | PRESC[2:0] | | |
| 0x03 | CTRLD | 7:0 | INITDLY[2:0] | | | ASDV | SAMPDLY[3:0] | | | |
| 0x04 | CTRLE | 7:0 | | | | | | WINCM[2:0] | | |
| 0x05 | SAMPCTRL | 7:0 | | | | SAMPLEN[4:0] | | | | |
| 0x06 | MUXPOS | 7:0 | | | | MUXPOS[4:0] | | | | |
| 0x07 | Reserved | | | | | | | | | |
| 0x08 | COMMAND | 7:0 | | | | | | | | STCONV |
| 0x09 | EVCTRL | 7:0 | | | | | | | | STARTEI |
| 0x0A | INTCTRL | 7:0 | | | | | | | WCOMP | RESRDY |
| 0x0B | INTFLAGS | 7:0 | | | | | | | WCOMP | RESRDY |
| 0x0C | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0D | TEMP | 7:0 | TEMP[7:0] | | | | | | | |
| 0x0E ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | RES | 7:0 | RES[7:0] | | | | | | | |
| | | 15:8 | RES[15:8] | | | | | | | |
| 0x12 | WINLT | 7:0 | WINLT[7:0] | | | | | | | |
| | | 15:8 | WINLT[15:8] | | | | | | | |
| 0x14 | WINHT | 7:0 | WINHT[7:0] | | | | | | | |
| | | 15:8 | WINHT[15:8] | | | | | | | |
| 0x16 | CALIB | 7:0 | | | | | | | | DUTYCYC |

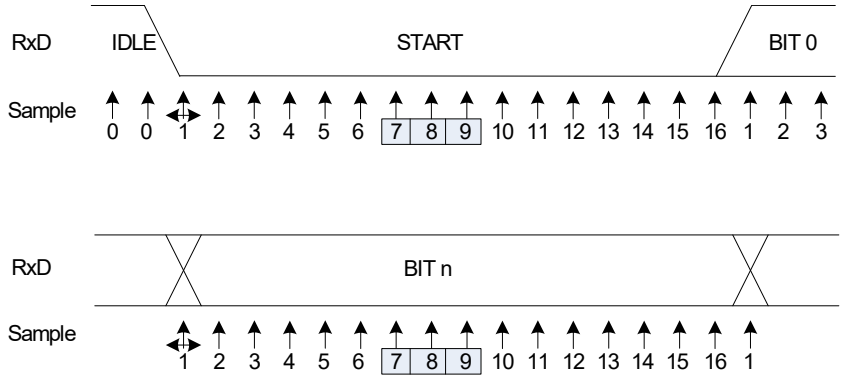
## 29.5 Register Description

### 30.3.1.1 UPDI UART

All transmission and reception of serial data on the UPDI is achieved using the UPDI frames presented in Figure 30-3. Communication is initiated from the master (debugger) side, and every transmission must start with a SYNCH character upon which the UPDI can recover the transmission baud rate, and store this setting for the coming data. The baud rate set by the SYNCH character will be used for both reception and transmission for the instruction byte received after the SYNCH. See UPDI Instruction Set for details on when the next SYNCH character is expected in the instruction stream.

There is no writable baud rate register in the UPDI, so the baud rate sampled from the SYNCH character is used for data recovery by sampling the Start bit, and performing a majority vote on the middle samples. This process is repeated for all bits in the frame, including the parity bit and two Stop bits. The baud generator uses 16 samples, and the majority voting is done on sample 7, 8, and 9.

**Figure 30-4. UPDI UART Start Bit and Data/Parity/Stop Bit Sampling**




The transmission baud rate must be set up in relation to the selected UPDI clock, which can be adjusted by UPDICLKSEL in UPDI.ASI_CTRLA. See Table 30-2 for recommended maximum and minimum baud rate settings.

**Table 30-2. Recommended UART Baud Rate Based on UPDICLKSEL Setting**

| UPDICLKSEL[1:0] | MAX Recommended Baud Rate | MIN Recommended Baud Rate |
|---|---|---|
| 0x1 (16 MHz) | 0.9 Mbps | 0.300 kbps |
| 0x2 (8 MHz) | 450 kbps | 0.150 kbps |
| 0x3 (4 MHz) - Default | 225 kbps | 0.075 kbps |

The UPDI Baud Rate Generator utilizes fractional baud counting to minimize the transmission error. With the fixed frame format used by the UPDI, the maximum and recommended receiver transmission error limits can be seen in the following table:

**Table 30-3. Receiver Baud Rate Error**

| Data + Parity Bits | $R_{slow}$ | $R_{fast}$ | Max. Total Error [%] | Recommended Max. RX Error [%] |
|---|---|---|---|---|
| 9 | 96.39 | 104.76 | +4.76/-3.61 | +1.5/-1.5 |

### 30.3.1.2 BREAK Character

The BREAK character is used to reset the internal state of the UPDI to the default setting. This is useful if the UPDI enters an error state due to a communication error, or when synchronization between the debugger and the UPDI is lost.

**Figure 32-55. Propagationdelay vs. $V_{CM}$ Falling Positive input, $V_{OD}$ = 25mV ( T=25°C)**

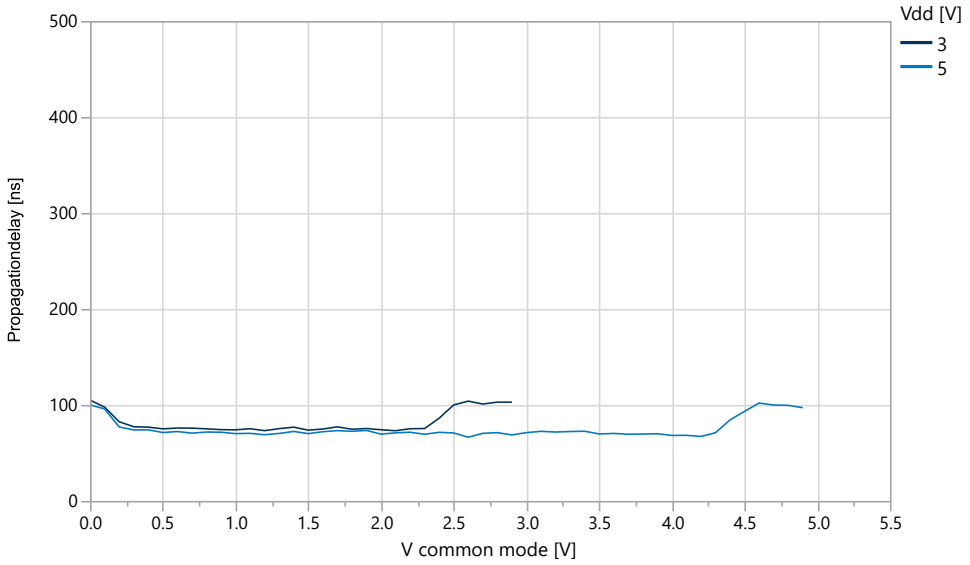



**Figure 32-56. Propagationdelay vs. $V_{CM}$ Rising Positive input, $V_{OD}$ = 30mV ( T=25°C)**