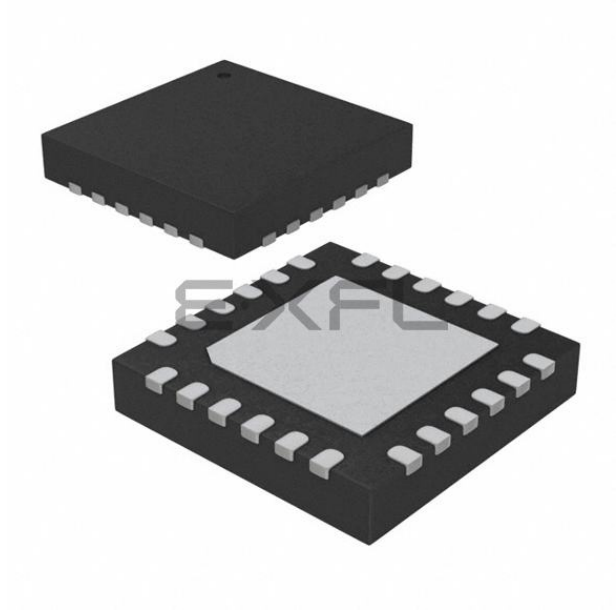


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	22
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	24-VFQFN Exposed Pad
Supplier Device Package	24-VQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny807-mnr

6.10.4.3 Oscillator Configuration

Name: OSCCFG
Offset: 0x02
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
	OSCLOCK						FREQSEL[1:0]	
Access	R						R	R
Reset	0						1	0

Bit 7 – OSCLOCK Oscillator Lock
This fuse bit is loaded to LOCK in CLKCTRL.OSC20MCALIBB during Reset.

Value	Description
0	Calibration registers of the 20 MHz oscillator are accessible
1	Calibration registers of the 20 MHz oscillator are locked

Bits 1:0 – FREQSEL[1:0] Frequency Select
These bits select the operation frequency of the 16/20 MHz internal oscillator (OSC20M) and determine the respective factory calibration values to be written to CAL20M in CLKCTRL.OSC20MCALIBA and TEMPCAL20M in CLKCTRL.OSC20MCALIBB.

Value	Description
0x1	Run at 16 MHz with corresponding factory calibration
0x2	Run at 20 MHz with corresponding factory calibration
Other	Reserved

Related Links

10.4 Register Summary - CLKCTRL
12. RSTCTRL - Reset Controller

- Event 3: Clock period = 1024 RTC clock cycles
- Event 4: Clock period = 512 RTC clock cycles
- Event 5: Clock period = 256 RTC clock cycles
- Event 6: Clock period = 128 RTC clock cycles
- Event 7: Clock period = 64 RTC clock cycles

The event users are configured by the Event System (EVSYS).

Related Links

14. EVSYS - Event System

22.6 Interrupts

Table 22-2. Available Interrupt Vectors and Sources For Devices With Up to 8 KB Flash

Offset	Name	Vector Description	Conditions
0x00	RTC	Real-time counter overflow and compare match interrupt	• Overflow (OVF): The counter has reached its top value and wrapped to zero. • Compare (CMP): Match between the counter value and the compare register.
0x02	PIT	Periodic Interrupt Timer interrupt	A time period has passed, as configured in RTC_PITCTRLA.PERIOD.

Table 22-3. Available Interrupt Vectors and Sources For Devices With More Than 8 KB Flash

Offset	Name	Vector Description	Conditions
0x00	RTC	Real-time counter overflow and compare match interrupt	• Overflow (OVF): The counter has reached its top value and wrapped to zero. • Compare (CMP): Match between the counter value and the compare register.
0x04	PIT	Periodic Interrupt Timer interrupt	A time period has passed, as configured in RTC_PITCTRLA.PERIOD.

When an interrupt condition occurs, the corresponding interrupt flag is set in the Interrupt Flags register of the peripheral (*peripheral*.INTFLAGS).

An interrupt source is enabled or disabled by writing to the corresponding enable bit in the peripheral's Interrupt Control register (*peripheral*.INTCTRL).

An interrupt request is generated when the corresponding interrupt source is enabled and the interrupt flag is set. The interrupt request remains active until the interrupt flag is cleared. See the peripheral's INTFLAGS register for details on how to clear interrupt flags.

Related Links

13. CPUINT - CPU Interrupt Controller
22.11.3 INTCTRL
22.11.13 PITINTCTRL

22.11.9 Period

Name: PER
Offset: 0x0A
Reset: 0xFF
Property: -

The RTC.PERL and RTC.PERH register pair represents the 16-bit value, 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. For more details on reading and writing 16-bit registers, refer to Accessing 16-bit Registers in the CPU chapter.

Due to synchronization between the RTC clock and system clock domains, there is a latency of two RTC clock cycles from updating the register until this has an effect. Application software needs to check that the PERBUSY flag in RTC.STATUS is cleared before writing to this register.

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] Period High Byte
These bits hold the MSB of the 16-bit Period register.

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

Table 23-8. Comparison of USART in Master SPI Mode and SPI Pins

USART	SPI	Comment
TxD	MOSI	Master out only
RxD	MISO	Master in only
XCK	SCK	Functionally identical
-	SS	Not supported by USART in Master SPI mode

Related Links

23.5.9 CTRLC

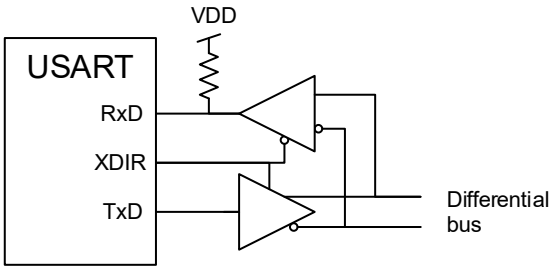
23.3.2.6 RS-485 Mode of Operation

The RS-485 feature enables the support of external components to comply with the RS-485 standard.

Either an external line driver is supported as shown in the figure below (RS-485=0x1 in USARTn.CTRLA), or control of the transmitter driving the TxD pin is provided (RS-485=0x2).

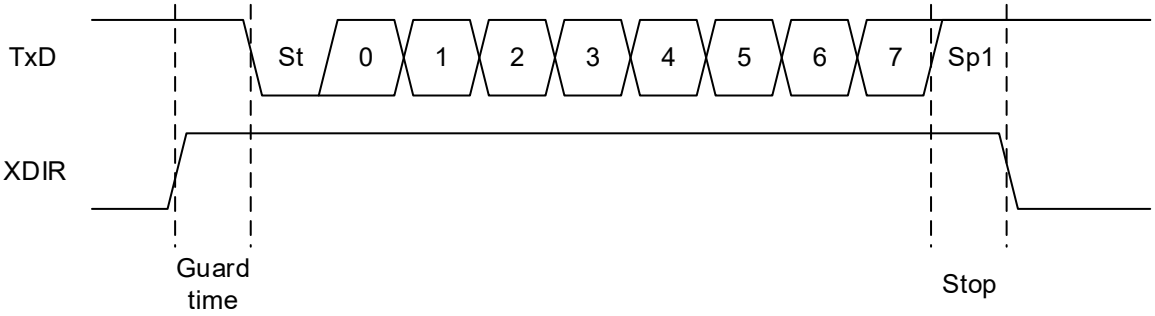
While operating in RS-485 mode, the Transmit Direction pin (XDIR) is driven high when the transmitter is active.

Figure 23-10. RS-485 Bus Connection



The XDIR pin goes high one baud clock cycle in advance of data being shifted out, to allow some guard time to enable the external line driver. The XDIR pin will remain high for the complete frame including Stop bit(s).

Figure 23-11. XDIR Drive Timing



Related Links

23.2.1 Signal Description

23.3.2.7 Start Frame Detection

The start frame detection is supported in UART mode only. The UART start frame detector is limited to Standby Sleep mode only and can wake up the system when a Start bit is detected.

23.5.5 USART Status Register

Name: STATUS
Offset: 0x04
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	RXCIF	TXCIF	DREIF	RXSIF	ISFIF		BDF	WFB
Access	R	R/W	R	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 7 – RXCIF USART Receive Complete Interrupt Flag
This flag is set to ‘1’ when there is unread data in the receive buffer and cleared when the receive buffer is empty (i.e. does not contain any unread data). When the receiver is disabled, the receive buffer will be flushed and consequently, the RXCIF will become '0'.

When interrupt-driven data reception is used, the receive complete interrupt routine must read the received data from RXDATA in order to clear the RXCIF. If not, a new interrupt will occur directly after the return from the current interrupt.

Bit 6 – TXCIF USART Transmit Complete Interrupt Flag
This flag is set when the entire frame in the Transmit Shift register has been shifted out and there are no new data in the transmit buffer (TXDATA).

This flag is automatically cleared when the transmit complete interrupt vector is executed. The flag can also be cleared by writing a ‘1’ to its bit location.

Bit 5 – DREIF USART Data Register Empty Flag
The DREIF indicates if the transmit buffer (TXDATA) is ready to receive new data. The flag is set to ‘1’ when the transmit buffer is empty and is ‘0’ when the transmit buffer contains data to be transmitted that has not yet been moved into the Shift register. DREIF is set after a Reset to indicate that the transmitter is ready. Always write this bit to ‘0’ when writing the STATUS register.

DREIF is cleared to ‘0’ by writing TXDATAL. When interrupt-driven data transmission is used, the Data Register Empty interrupt routine must either write new data to TXDATA in order to clear DREIF or disable the Data Register Empty interrupt. If not, a new interrupt will occur directly after the return from the current interrupt.

Bit 4 – RXSIF USART Receive Start Interrupt Flag
The RXSIF flag indicates a valid Start condition on RxD line. The flag is set when the system is in standby modes and a high (IDLE) to low (START) valid transition is detected on the RxD line. If the start detection is not enabled, the RXSIF will always be read as '0'. This flag can only be cleared by writing a ‘1’ to its bit location. This flag is not used in the Master SPI mode operation.

Bit 3 – ISFIF Inconsistent Sync Field Interrupt Flag
This bit is set when the auto-baud is enabled and the sync field bit time is too fast or too slow to give a valid baud setting. It will also be set when USART is set to LINAUTO mode and the SYNC character differ from data value 0x55.

Writing a ‘1’ to this bit will clear the flag and bring the USART back to Idle state.

24.5.3 Interrupt Control

Name: INTCTRL
Offset: 0x02
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	DREIE	SSIE				IE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 7 – RXCIE Receive Complete Interrupt Enable
In Buffer mode, this bit enables the receive complete interrupt. The enabled interrupt will be triggered when the RXCIF flag in the SPIn.INTFLAGS register is set. In the Non-Buffer mode this bit is zero.

Bit 6 – TXCIE Transfer Complete Interrupt Enable
In Buffer mode, this bit enables the transfer complete interrupt. The enabled interrupt will be triggered when the TXCIF flag in the SPIn.INTFLAGS register is set. In the Non-Buffer mode, this bit is zero.

Bit 5 – DREIE Data Register Empty Interrupt Enable
In Buffer mode, this bit enables the data register empty interrupt. The enabled interrupt will be triggered when the DREIF flag in the SPIn.INTFLAGS register is set. In the Non-Buffer mode, this bit is zero.

Bit 4 – SSIE Slave Select Trigger Interrupt Enable
In Buffer mode, this bit enables the Slave Select interrupt. The enabled interrupt will be triggered when the SSIF flag in the SPIn.INTFLAGS register is set. In the Non-Buffer mode, this bit is zero.

Bit 0 – IE Interrupt Enable
This bit enables the SPI interrupt when the SPI is not in Buffer mode. The enabled interrupt will be triggered when RXCIF/IF is set in the SPIn.INTFLAGS register.

Disable bit (SSD) is not '1'. The flag is cleared by writing a 1 to its bit location. In the Non-Buffer mode, this bit is always 0.

Bit 0 – BUFOVF Buffer Overflow
This flag is only used in Buffer mode. This flag indicates data loss due to a receiver buffer full condition. This flag is set if a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full (two characters) and a third byte has been received in the Shift register. If there is no transmit data the buffer overflow will not be set before the start of a new serial transfer. This flag is valid until the receive buffer (SPIn.DATA) is read. Always write this bit location to 0 when writing the SPIn.INTFLAGS register. In the Non-Buffer mode, this bit is always 0.

24.5.5 Data

Name: DATA
Offset: 0x04
Reset: 0x00
Property: -

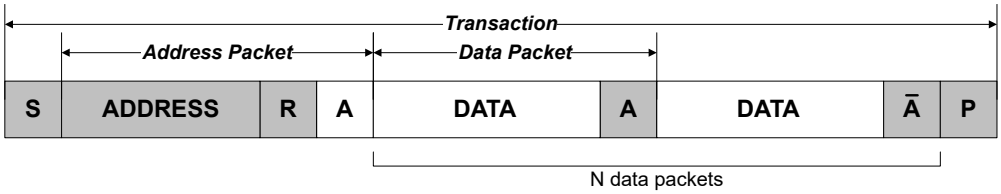
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0] SPI Data
The SPIn.DATA register is used for sending and receiving data. Writing to the register initiates the data transmission, and the byte written to the register will be shifted out on the SPI output line.

Reading this register in Buffer mode will read the second receive buffer and the contents of the first receive buffer will be moved to the second receive buffer.

Figure 25-7 illustrates the master read transaction. The master initiates the transaction by issuing a Start condition followed by an address packet with the direction bit set to '1' (ADDRESS+R). The addressed slave must acknowledge the address for the master to be allowed to continue the transaction.

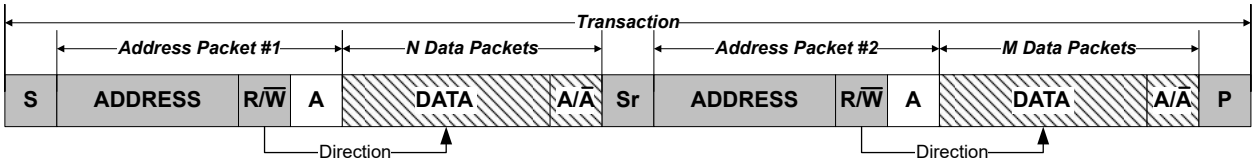
Figure 25-7. Master Read Transaction



Assuming the slave acknowledges the address, the master can start receiving data from the slave. There are no limitations to the number of data packets that can be transferred. The slave transmits the data while the master signals ACK or NACK after each data byte. The master terminates the transfer with a NACK before issuing a Stop condition.

Figure 25-8 illustrates a combined transaction. A combined transaction consists of several read and write transactions separated by repeated Start conditions (Sr).

Figure 25-8. Combined Transaction

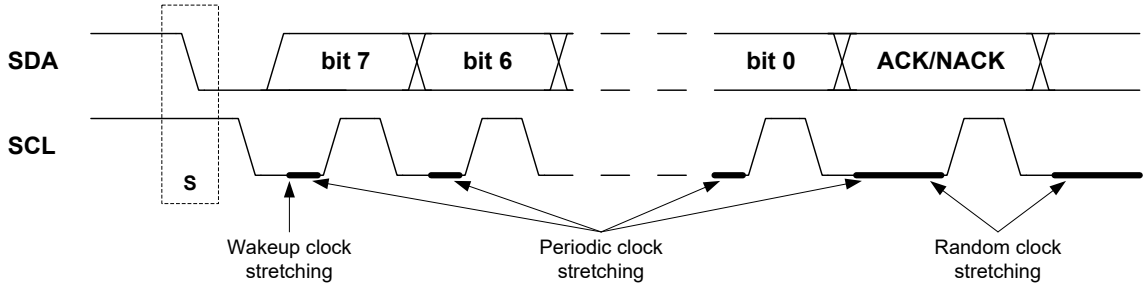


25.3.2.6 Clock and Clock Stretching

All devices connected to the bus are allowed to stretch the low period of the clock to slow down the overall clock frequency or to insert Wait states while processing data. A device that needs to stretch the clock can do this by holding/forcing the SCL line low after it detects a low level on the line.

Three types of clock stretching can be defined, as shown in Figure 25-9.

Figure 25-9. Clock Stretching (1)



Note: Clock stretching is not supported by all I^2C slaves and masters.

If a slave device is in Sleep mode and a Start condition is detected, the clock stretching normally works during the wake-up period. For AVR devices, the clock stretching will be either directly before or after the ACK/NACK bit, as AVR devices do not need to wake-up for transactions that are not addressed to it.

A slave device can slow down the bus frequency by stretching the clock periodically on a bit level. This allows the slave to run at a lower system clock frequency. However, the overall performance of the bus will be reduced accordingly. Both the master and slave device can randomly stretch the clock on a byte level basis before and after the ACK/NACK bit. This provides time to process incoming or prepare outgoing data or perform other time-critical tasks.

26.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: -

If an NMI has been triggered, this register is not writable.

Bit	7	6	5	4	3	2	1	0
	RESET						NMIEN	ENABLE
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – RESET Reset CRCSCAN
Writing this bit to '1' resets the CRCSCAN peripheral: The CRCSCAN Control registers and STATUS register (CTRLA, CTRLB, STATUS) will be cleared one clock cycle after the RESET bit was written to '1'.

If NMIEN is '0', this bit is writable both when the CRCSCAN is busy (the BUSY bit in CRCSCAN.STATUS is '1') and not busy (the BUSY bit is '0'), and will take effect immediately.

If NMIEN is '1', this bit is only writable when the CRCSCAN is not busy (the BUSY bit in CRCSCAN.STATUS is '0').

The RESET bit is a strobe bit.

Bit 1 – NMIEN Enable NMI Trigger
When this bit is written to '1', any CRC failure will trigger an NMI.

This can only be cleared by a system Reset - it is not cleared by a write to the RESET bit.

This bit can only be written to '1' when the CRCSCAN is not busy (the BUSY bit in CRCSCAN.STATUS is '0').

Bit 0 – ENABLE Enable CRCSCAN
Writing this bit to '1' enables the CRCSCAN peripheral with the current settings. It will stay '1' even after a CRC check has completed, but writing it to ‘1’ again will start a new check.

Writing the bit to '0' will disable the CRCSCAN after the ongoing check is completed (after reaching the end of the section it is set up to check). A failure in the ongoing check will still be detected and can cause an NMI if the NMIEN bit is '1'.

The CRCSCAN can be enabled during the internal Reset initialization to verify Flash sections before letting the CPU start normal code execution (see the device data sheet fuse description). If the CRCSCAN is enabled during the internal Reset initialization, the ENABLE bit will read as '1' when normal code execution starts.

To see whether the CRCSCAN peripheral is busy with an ongoing check, poll the Busy bit (BUSY) in the STATUS register (CRCSCAN.STATUS).

Related Links
6.10 Configuration and User Fuses (FUSE)
12.3.2.2 Reset Time

27.5.6 TRUTHn

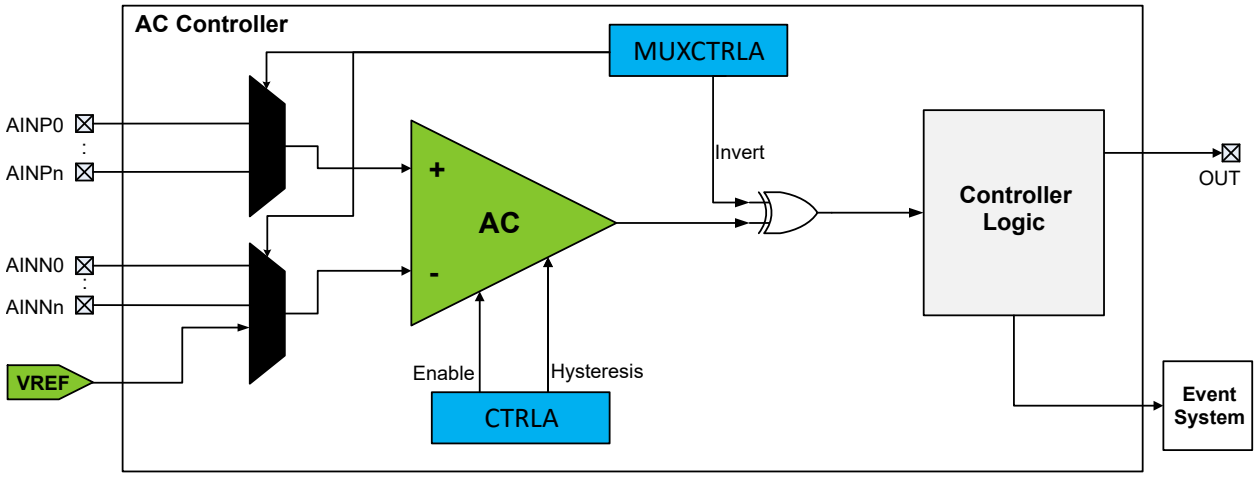
Name: TRUTH
Offset: 0x08 + n*0x04 [n=0..1]
Reset: 0x00
Property: Enable-Protected

Bit	7	6	5	4	3	2	1	0
	TRUTH[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 – TRUTH[7:0] Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

28.2.1 Block Diagram

Figure 28-1. Analog Comparator



Note: Refer to 28.2.2 Signal Description for the number of AINN and AINP.

28.2.2 Signal Description

Signal	Description	Type
AINN0	Negative Input 0	Analog
AINN1	Negative Input 1	Analog
AINP0	Positive Input 0	Analog
AINP1	Positive Input 1	Analog
OUT	Comparator Output for AC	Digital

28.2.3 System Dependencies

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

Table 28-1. AC System Dependencies

Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	Yes	PORT
Interrupts	Yes	CPUINT
Events	Yes	EVSYS
Debug	Yes	UPDI

28.2.3.1 Clocks

This peripheral depends on the peripheral clock.

28.2.3.2 I/O Lines and Connections

I/O pins AINN0-AINN1 and AINP0- AINP1 are all analog inputs to the AC.

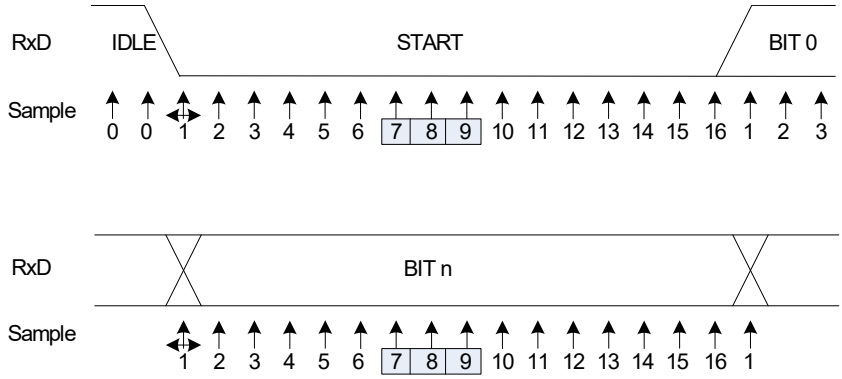
For correct operation, the pins must be configured in the port and port multiplexing peripherals.

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 30.3.3 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:

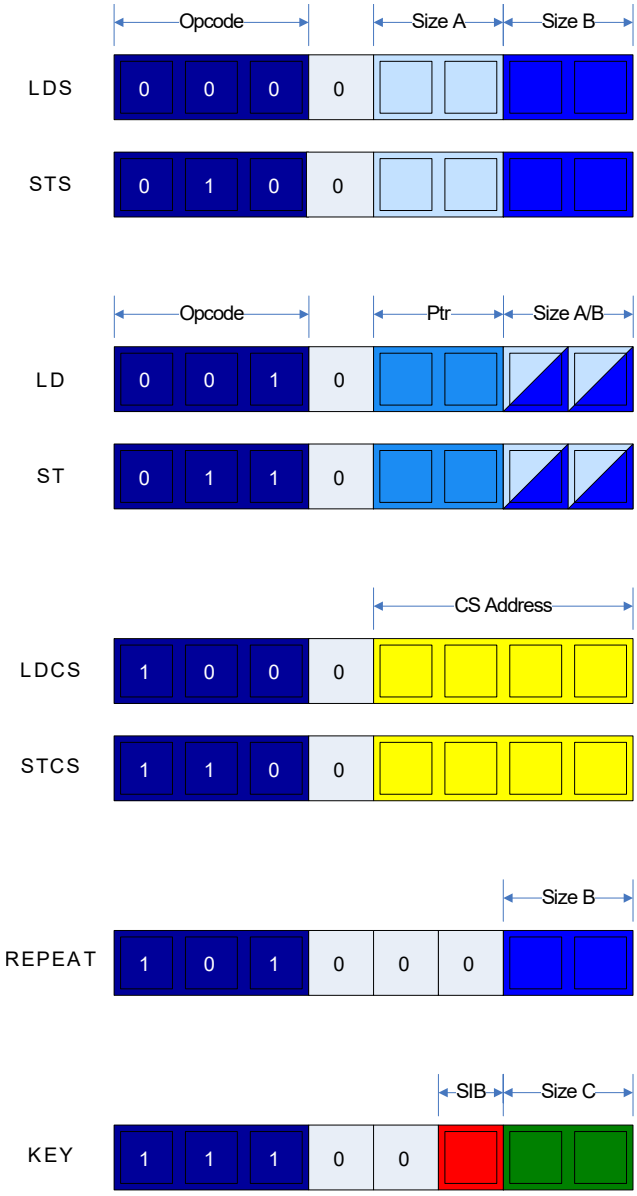
Table 30-3. Receiver Baud Rate Error

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 the synchronization between the debugger and the UPDI is lost.

Figure 30-8. UPDI Instruction Set Overview



OPCODE	
0 0 0	LDS
0 0 1	LD
0 1 0	STS
0 1 1	ST
1 0 0	LDCS (LDS Control/Status)
1 0 1	REPEAT
1 1 0	STCS (STS Control/Status)
1 1 1	KEY

Size A - Address size	
0 0	Byte
0 1	Word (2 Bytes)
1 0	Reserved
1 1	Reserved

Ptr - Pointer access	
0 0	*(ptr)
0 1	*(ptr++)
1 0	ptr
1 1	Reserved

Size B - Data size	
0 0	Byte
0 1	Word (2 Bytes)
1 0	Reserved
1 1	Reserved

CS Address (CS - Control/Status reg.)	
0 0 0 0	Reg 0
0 0 0 1	Reg 1
0 0 1 0	Reg 2
0 0 1 1	Reg 3
0 1 0 0	Reg 4 (ASI CS space)
.....	
1 1 1 1	Reserved

Size C - Key size	
0 0	64 bits (8 Bytes)
0 1	128 bits (16 Bytes)
1 0	Reserved
1 1	Reserved

SIB – System Information Block sel.	
0	Receive KEY
1	Send SIB

Table 31-6. Start-Up, Reset, and Wake-Up Time from OSC20M

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
t_{wakeup}	Start-up time from any Reset release		-	200	-	µs
	Wake-up from Idle mode	OSC20M @ 20 MHz; V_{DD}=5V	-	1	-	
		OSC20M @ 10 MHz; V_{DD}=3V	-	2	-	
		OSC20M @ 5 MHz; V_{DD}=2V	-	4	-	
	Wake-up from Standby and Power-down mode		-	10	-	

31.6 Power Consumption of Peripherals

The table below can be used to calculate the additional current consumption for the different I/O peripherals in the various operating modes.

Operating conditions:

- V_{DD} = 3V
- T = 25°C
- OSC20M at 1 MHz used as system clock source, except where otherwise specified
- In Idle Sleep mode, except where otherwise specified

Table 31-7. Power Consumption of Peripherals

Peripheral	Conditions	Typ.(1)	Unit
BOD	Continuous	19	µA
	Sampling @ 1 kHz	1.2	
TCA	16-bit count @ 1 MHz	12.6	µA
TCB	16-bit count @ 1 MHz	7.4	µA
RTC	16-bit count	1.2	µA
WDT		0.7	µA
OSC20M		125	µA
AC	Low-power mode(2)	45	µA
ADC	50 ksps	325	µA
	100 ksps	340	µA
USART	Enable @ 9600 Baud	13	µA
SPI (Master)	Enable @ 100 kHz	2.1	µA
TWI (Master)	Enable @ 100 kHz	23.9	µA

Peripheral	Conditions	Typ.(1)	Unit
TWI (Slave)	Enable @ 100 kHz	17.1	µA
Flash programming	Erase Operation	1.5	mA
	Write Operation	3.0	

Note:

1. Current consumption of the module only. To calculate the total power consumption of the system, add this value to the base power consumption as listed in *Power Consumption*.
2. CPU in Standby mode.

31.7 BOD and POR Characteristics

Table 31-8. Power Supply Characteristics

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
SRON	Power-on Slope		-	-	100	V/ms

Table 31-9. Power-On Reset (POR) Characteristics

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
V_{POR}	POR threshold voltage on V_{DD} falling	V_{DD} falls/rises at 0.5 V/ms or slower	0.8	-	1.6	V
	POR threshold voltage on V_{DD} rising		1.4	-	1.8	

Table 31-10. Brown-Out Detection (BOD) Characteristics

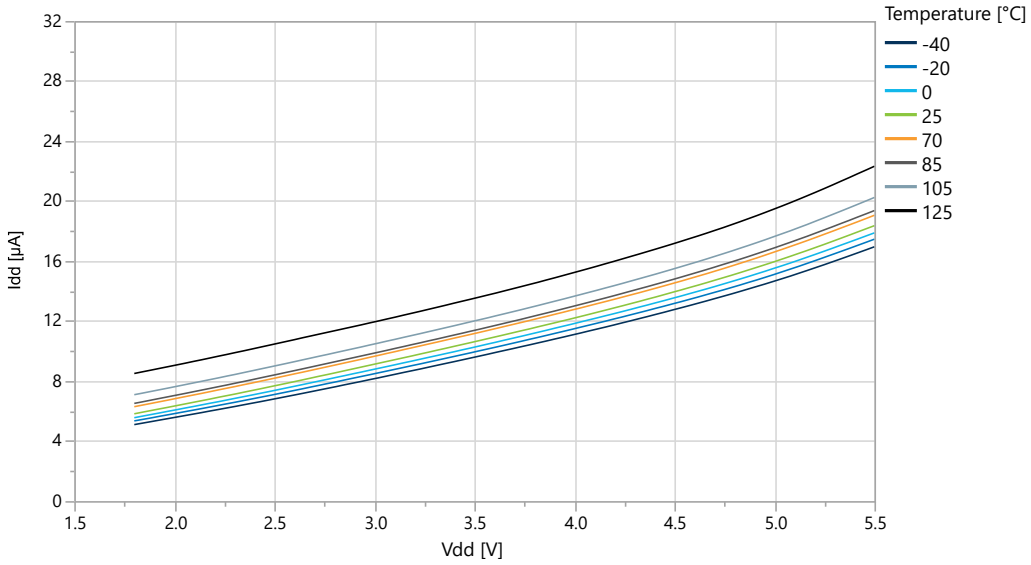
Symbol	Description	Condition	Min.	Typ.	Max.	Unit
V_{BOD}	BOD triggering level (falling/rising)	BODLEVEL7	3.9	4.2	4.5	V
		BODLEVEL2	2.4	2.6	2.9	
		BODLEVEL0	1.7	1.8	2.0	
V_{INT}	Interrupt level 0	Percentage above the selected BOD level	-	4	-	%
	Interrupt level 1		-	13	-	
	Interrupt level 2		-	25	-	
V_{HYS}	Hysteresis	BODLEVEL7	-	80	-	mV
		BODLEVEL2	-	40	-	
		BODLEVEL0	-	25	-	
T_{BOD}	Detection time	Continuous	-	7	-	µs
		Sampled, 1 kHz	-	1	-	ms
		Sampled, 125 Hz	-	8	-	
T_{start}	Start-up time	Time from enable to ready	-	40	-	µs

Symbol	Description	Conditions		Min.	Typ.	Max.	Unit
		REFSEL = INTERNAL or VDD 1.1V≤V_{REF}	f_{ADC}=77 ksps	-	0.4	1.0	
		REFSEL = INTERNAL 1.1V≤V_{REF}	f_{ADC}=115 ksps	-	0.5	1.6	
		REFSEL = VDD 1.8V≤V_{REF}	f_{ADC}=115 ksps	-	0.9	2.0	
EABS	Absolute accuracy	REFSEL = INTERNAL V_{REF} = 1.1V	T=[0-105]°C V_{DD} = [1.8V-3.6V]	-	3	30	LSB
			V_{DD} = [1.8V-3.6V]	-	3	40	
		REFSEL = V_{DD}		-	2	5	
		REFSEL = INTERNAL		-	-	65	
EGAIN	Gain error	REFSEL = INTERNAL V_{REF} = 1.1V	T=[0-105]°C V_{DD} = [1.8V-3.6V]	-25	3	25	LSB
			V_{DD} = [1.8V-3.6V]	-35	3	35	
		REFSEL = V_{DD}		-1	2	4	
		REFSEL = INTERNAL		-60	-	60	
EOFF	Offset error	REFSEL = INTERNAL V_{REF} = 0.55V		-5	-0.5	2	LSB
		REFSEL = INTERNAL 1.1V ≤ V_{REF}		-4	-0.5	2	LSB

Note:

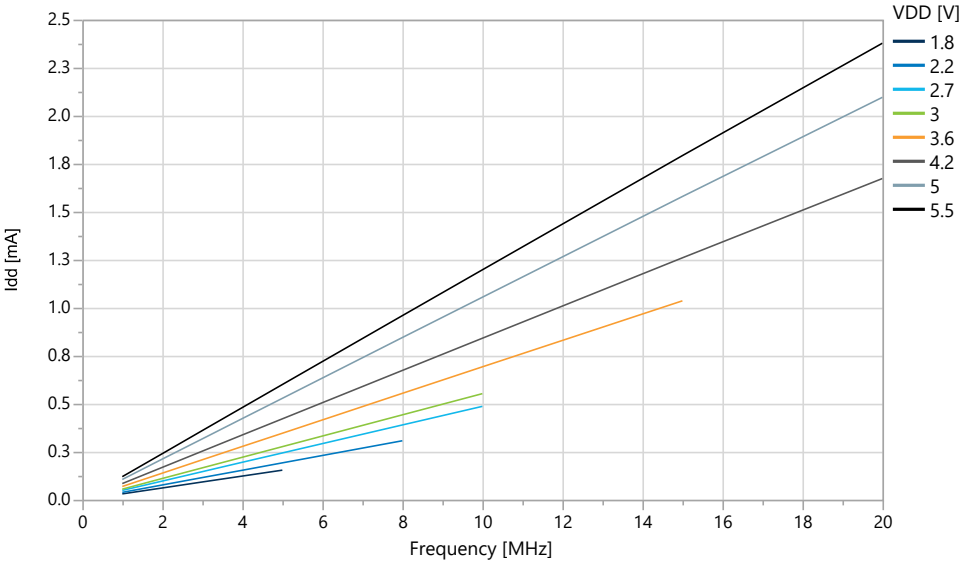
1. A DNL error of less than or equal to 1 LSB ensures a monotonic transfer function with no missing codes.
2. These values are based on characterization and not covered by production test limits.
3. Reference setting and f_{ADC} must fulfill the specification in Clock and Timing Characteristics, and Power supply, Reference, and Input Range tables.

Figure 32-5. Active Supply Current vs. V_{DD} (f=32 KHz OSCULP32K)



32.1.2 Supply Currents in Idle Mode

Figure 32-6. Idle Supply Current vs. Frequency (1-20 MHz) at T=25°C



- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks