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	66MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	45
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 11x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3c2256c-a2ur

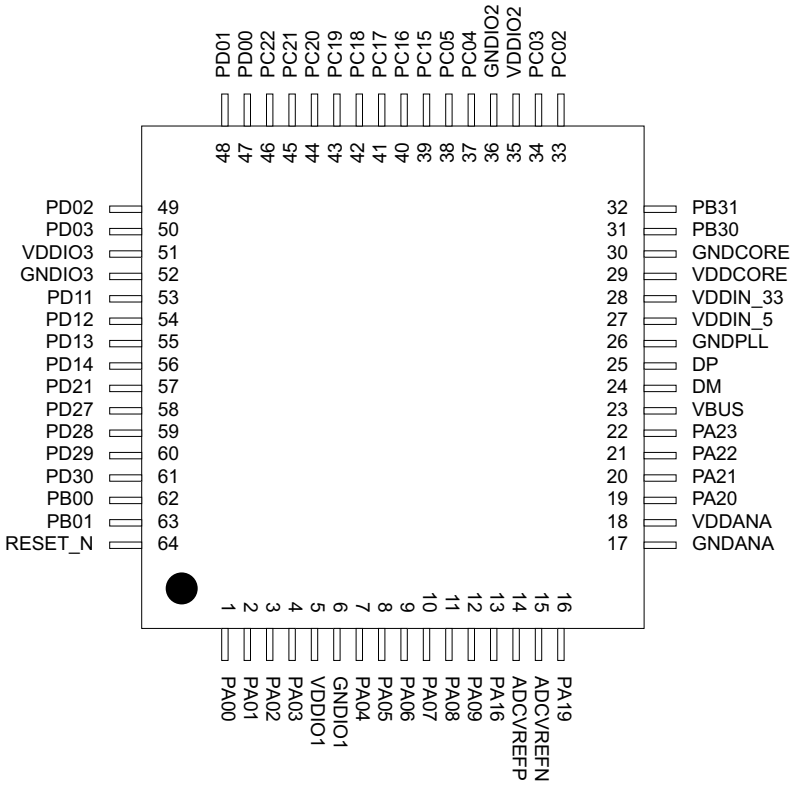
- One 4-Channel 20-bit Pulse Width Modulation Controller (PWM)
  - Complementary outputs, with Dead Time Insertion
  - Output Override and Fault Protection
- Two Quadrature Decoders
- One 16-channel 12-bit Pipelined Analog-To-Digital Converter (ADC)
  - Dual Sample and Hold Capability Allowing 2 Synchronous Conversions
  - Single-Ended and Differential Channels, Window Function
- Two 12-bit Digital-To-Analog Converters (DAC), with Dual Output Sample System
- Four Analog Comparators
- Six 16-bit Timer/Counter (TC) Channels
  - External Clock Inputs, PWM, Capture and Various Counting Capabilities
- One Peripheral Event Controller
  - Trigger Actions in Peripherals Depending on Events Generated from Peripherals or from Input Pins
  - Deterministic Trigger
  - 34 Events and 22 Event Actions
- Five Universal Synchronous/Asynchronous Receiver/Transmitters (USART)
  - Independent Baudrate Generator, Support for SPI, LIN, IrDA and ISO7816 interfaces
  - Support for Hardware Handshaking, RS485 Interfaces and Modem Line
- Two Master/Slave Serial Peripheral Interfaces (SPI) with Chip Select Signals
- One Inter-IC Sound (I2S) Controller
  - Compliant with I2S Bus Specification
  - Time Division Multiplexed mode
- Three Master and Three Slave Two-Wire Interfaces (TWI), 400kbit/s I^2C-compatible
- QTouch® Library Support
  - Capacitive Touch Buttons, Sliders, and Wheels
  - QTouch® and QMatrix® Acquisition
- On-Chip Non-intrusive Debug System
  - Nexus Class 2+, Runtime Control, Non-Intrusive Data and Program Trace
  - aWire™ single-pin programming trace and debug interface muxed with reset pin
  - NanoTrace™ provides trace capabilities through JTAG or aWire interface
- 3 package options
  - 64-pin QFN/TQFP (45 GPIO pins)
  - 100-pin TQFP (81 GPIO pins)
  - 144-pin LQFP (123 GPIO pins)
- Two operating voltage ranges:
  - Single 5V Power Supply
  - Single 3.3V Power Supply

3. Package and Pinout

3.1 Package

The device pins are multiplexed with peripheral functions as described in Table 3-1 on page 11.

Figure 3-1. QFN64/TQFP64 Pinout



Note: on QFN packages, the exposed pad is unconnected.

Table 3-1. GPIO Controller Function Multiplexing

TQFP / QFN 64	TQFP 100	LQFP 144	PIN	GPIO	Supply	Pin Type (1)	GPIO function					
							A	B	C	D	E	F
		14	PB11	43	VDDIO1	x1/x2	USART1 - DSR	SPI0 - MISO	PWM - PWMH[1]			
		15	PB12	44	VDDIO1	x1/x2	USART1 - DCD	SPI0 - SCK	PWM - PWML[2]			
		16	PB13	45	VDDIO1	x1/x2	USART1 - RI	SPI0 - NPCS[0]	PWM - PWMH[2]		MACB - RX_ER	
		17	PB14	46	VDDIO1	x1/x2	USART1 - RTS	SPI0 - NPCS[1]	PWM - PWML[3]		MACB - MDC	
		18	PB15	47	VDDIO1	x1/x2	USART1 - CTS	USART1 - CLK	PWM - PWMH[3]		MACB - MDIO	
		19	PB16	48	VDDIO1	x1/x2	USART1 - RXD	SPI0 - NPCS[2]	PWM - EXT_ FAULTS[0]		CANIF - RXLINE[0]	
		20	PB17	49	VDDIO1	x1/x2	USART1 - TXD	SPI0 - NPCS[3]	PWM - EXT_ FAULTS[1]		CANIF - TXLINE[0]	
		57	PB18	50	VDDIO2	x1/x2	TC0 - CLK2		EIC - EXTINT[4]			
	42	58	PB19	51	VDDIO2	x1/x2	TC0 - A0	SPI1 - MOSI	IISC - ISDO		MACB - CRS	
	43	59	PB20	52	VDDIO2	x1/x2	TC0 - B0	SPI1 - MISO	IISC - ISDI	ACIFA1 - ACAOUT	MACB - COL	
	44	60	PB21	53	VDDIO2	x2/x4	TC0 - CLK1	SPI1 - SCK	IISC - IMCK	ACIFA1 - ACBOUT	MACB - RXD[2]	
	45	61	PB22	54	VDDIO2	x1/x2	TC0 - A1	SPI1 - NPCS[3]	IISC - ISCK	SCIF - GCLK[0]	MACB - RXD[3]	
	46	62	PB23	55	VDDIO2	x1/x2	TC0 - B1	SPI1 - NPCS[2]	IISC - IWS	SCIF - GCLK[1]	MACB - RX_CLK	
		63	PB24	56	VDDIO2	x1/x2	TC0 - CLK0	SPI1 - NPCS[1]				
		64	PB25	57	VDDIO2	x1/x2	TC0 - A2	SPI1 - NPCS[0]	PEVC - PAD_EVT [8]			
		65	PB26	58	VDDIO2	x2/x4	TC0 - B2	SPI1 - SCK	PEVC - PAD_EVT [9]		MACB - TX_EN	
		66	PB27	59	VDDIO2	x1/x2	QDEC0 - QEPA	SPI1 - MISO	PEVC - PAD_EVT [10]	TC1 - CLK0	MACB - TXD[0]	
		67	PB28	60	VDDIO2	x1/x2	QDEC0 - QEPB	SPI1 - MOSI	PEVC - PAD_EVT [11]	TC1 - B0	MACB - TXD[1]	
		68	PB29	61	VDDIO2	x1/x2	QDEC0 - QEPI	SPI0 - NPCS[0]	PEVC - PAD_EVT [12]	TC1 - A0		
31	47	69	PB30	62	VDDIO2	x1						
32	48	70	PB31	63	VDDIO2	x1						
	49	71	PC00	64	VDDIO2	x1/x2	USBC - ID	SPI0 - NPCS[1]	USART2 - CTS	TC1 - B2	CANIF - TXLINE[1]	
	50	72	PC01	65	VDDIO2	x1/x2	USBC - VBOF	SPI0 - NPCS[2]	USART2 - RTS	TC1 - A2	CANIF - RXLINE[1]	

depending on the configuration of the OCD AXS register. For details, see the AVR32UC Technical Reference Manual.

Table 3-5. Nexus OCD AUX port connections

Pin	AXS=0	AXS=1	AXS=2
EVTI_N	PA08	PB19	PA10
MDO[5]	PC05	PC31	PB06
MDO[4]	PC04	PC12	PB15
MDO[3]	PA23	PC11	PB14
MDO[2]	PA22	PB23	PA27
MDO[1]	PA19	PB22	PA26
MDO[0]	PA09	PB20	PA19
EVTO_N	PD29	PD29	PD29
MCKO	PD13	PB21	PB26
MSEO[1]	PD30	PD08	PB25
MSEO[0]	PD14	PD07	PB18

3.2.6 Other Functions

The functions listed in Table 3-6 are not mapped to the normal GPIO functions. The aWire DATA pin will only be active after the aWire is enabled. The aWire DATAOUT pin will only be active after the aWire is enabled and the 2_PIN_MODE command has been sent.

Table 3-6. Other Functions

QFN64/ TQFP64 pin	TQFP100 pin	LQFP144 pin	Pad	Oscillator pin
64	98	142	RESET_N	aWire DATA
3	3	3	PA02	aWire DATAOUT

3.3 Signals Description

The following table give details on the signal name classified by peripherals.

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Power				
VDDIO1 VDDIO2 VDDIO3	I/O Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V
VDDANA	Analog Power Supply	Power Input		4.5V to 5.5V or 3.0V to 3.6 V

Table 3-7. Signal Description List

Signal Name	Function	Type	Active Level	Comments
VDDIN_5	1.8V Voltage Regulator Input	Power Input		Power Supply: 4.5V to 5.5V or 3.0V to 3.6 V
VDDIN_33	USB I/O power supply	Power Output/ Input		Capacitor Connection for the 3.3V voltage regulator or power supply: 3.0V to 3.6 V
VDDCORE	1.8V Voltage Regulator Output	Power output		Capacitor Connection for the 1.8V voltage regulator
GNDIO1 GNDIO2 GNDIO3	I/O Ground	Ground		
GNDANA	Analog Ground	Ground		
GNDCORE	Ground of the core	Ground		
GNDPLL	Ground of the PLLs	Ground		
Analog Comparator Interface - ACIFA0/1				
AC0AN1/AC0AN0	Negative inputs for comparator AC0A	Analog		
AC0AP1/AC0AP0	Positive inputs for comparator AC0A	Analog		
AC0BN1/AC0BN0	Negative inputs for comparator AC0B	Analog		
AC0BP1/AC0BP0	Positive inputs for comparator AC0B	Analog		
AC1AN1/AC1AN0	Negative inputs for comparator AC1A	Analog		
AC1AP1/AC1AP0	Positive inputs for comparator AC1A	Analog		
AC1BN1/AC1BN0	Negative inputs for comparator AC1B	Analog		
AC1BP1/AC1BP0	Positive inputs for comparator AC1B	Analog		
ACAOUT/ACBOUT	analog comparator outputs	output		
ADC Interface - ADCIFA				
ADCIN[15:0]	ADC input pins	Analog		
ADCREF0	Analog positive reference 0 voltage input	Analog		
ADCREF1	Analog positive reference 1 voltage input	Analog		
ADCVREFP	Analog positive reference connected to external capacitor	Analog		

Table 4-3. System Registers (Continued)

Reg #	Address	Name	Function
90	360	MPUPSR2	MPU Privilege Select Register region 2
91	364	MPUPSR3	MPU Privilege Select Register region 3
92	368	MPUPSR4	MPU Privilege Select Register region 4
93	372	MPUPSR5	MPU Privilege Select Register region 5
94	376	MPUPSR6	MPU Privilege Select Register region 6
95	380	MPUPSR7	MPU Privilege Select Register region 7
96	384	MPUCRA	Unused in this version of AVR32UC
97	388	MPUCRB	Unused in this version of AVR32UC
98	392	MPUBRA	Unused in this version of AVR32UC
99	396	MPUBRB	Unused in this version of AVR32UC
100	400	MPUAPRA	MPU Access Permission Register A
101	404	MPUAPRB	MPU Access Permission Register B
102	408	MPUCR	MPU Control Register
103	412	SS_STATUS	Secure State Status Register
104	416	SS_ADRF	Secure State Address Flash Register
105	420	SS_ADRR	Secure State Address RAM Register
106	424	SS_ADR0	Secure State Address 0 Register
107	428	SS_ADR1	Secure State Address 1 Register
108	432	SS_SP_SYS	Secure State Stack Pointer System Register
109	436	SS_SP_APP	Secure State Stack Pointer Application Register
110	440	SS_RAR	Secure State Return Address Register
111	444	SS_RSR	Secure State Return Status Register
112-191	448-764	Reserved	Reserved for future use
192-255	768-1020	IMPL	IMPLEMENTATION DEFINED

4.5 Exceptions and Interrupts

In the AVR32 architecture, events are used as a common term for exceptions and interrupts. AVR32UC incorporates a powerful event handling scheme. The different event sources, like Illegal Op-code and interrupt requests, have different priority levels, ensuring a well-defined behavior when multiple events are received simultaneously. Additionally, pending events of a higher priority class may preempt handling of ongoing events of a lower priority class.

When an event occurs, the execution of the instruction stream is halted, and execution is passed to an event handler at an address specified in Table 4-4 on page 38. Most of the handlers are placed sequentially in the code space starting at the address specified by EVBA, with four bytes between each handler. This gives ample space for a jump instruction to be placed there, jumping to the event routine itself. A few critical handlers have larger spacing between them, allowing the entire event routine to be placed directly at the address specified by the EVBA-relative offset generated by hardware. All interrupt sources have autovectored interrupt service routine (ISR) addresses. This allows the interrupt controller to directly specify the ISR address as an address

5.2 Physical Memory Map

The system bus is implemented as a bus matrix. All system bus addresses are fixed, and they are never remapped in any way, not even in boot. Note that AVR32UC CPU uses unsegmented translation, as described in the AVR32 Architecture Manual. The 32-bit physical address space is mapped as follows:

Table 5-1. AT32UC3C Physical Memory Map

Device	Start Address	AT32UC3 Derivatives							
		C0512C	C1512C C2512C	C0256C	C1256C C2256C	C0128C	C1128C C2128C	C064C	C164C C264C
Embedded SRAM	0x0000_0000	64 KB	64 KB	64 KB	64 KB	32 KB	32 KB	16 KB	16 KB
Embedded Flash	0x8000_0000	512 KB	512 KB	256 KB	256 KB	128 KB	128 KB	64 KB	64 KB
SAU	0x9000_0000	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB	1 KB
HSB SRAM	0xA000_0000	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB
EBI SRAM CS0	0xC000_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM CS2	0xC800_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM CS3	0xCC00_0000	16 MB	-	16 MB	-	16 MB	-	16 MB	-
EBI SRAM /SDRAM CS1	0xD000_0000	128 MB	-	128 MB	-	128 MB	-	128 MB	-
HSB-PB Bridge C	0xFFFD_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
HSB-PB Bridge B	0xFFFE_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
HSB-PB Bridge A	0xFFFF_0000	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB

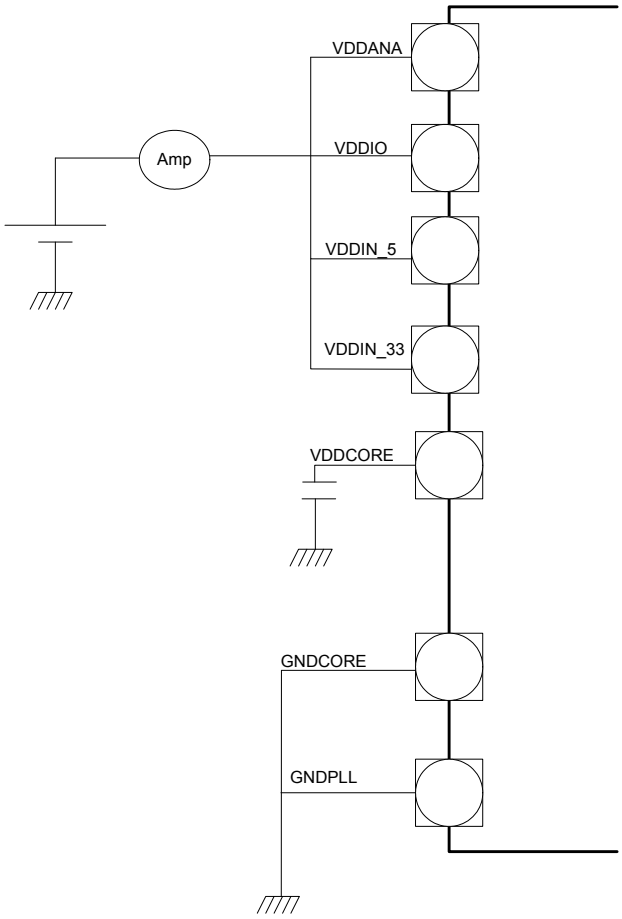
  - Internal 3.3V regulator is off
- T_A = 25°C
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
  - OSC0/1 (crystal oscillator) stopped
  - OSC32K (32KHz crystal oscillator) stopped
  - PLL0 running
  - PLL1 stopped
- Clocks
  - External clock on XIN0 as main clock source **(**10MHz**)**
  - CPU, HSB, and PBB clocks undivided
  - PBA, PBC clock divided by 4
  - All peripheral clocks running

Table 7-4. Power Consumption for Different Operating Modes

Mode	Conditions	Measured on	Consumption Typ	Unit
Active[1]	CPU running a recursive Fibonacci algorithm	Amp	512	µA/MHz
Idle[1]			258	
Frozen[1]			106	
Standby[1]			48	
Stop			73	µA
DeepStop			43	
Static	OSC32K and AST running		32	
	AST and OSC32K stopped		31	

Note: 1. These numbers are valid for the measured condition only and must not be extrapolated to other frequencies.

Figure 7-1. Measurement Schematic



7.4.1 Peripheral Power Consumption

The values in Table 7-5 are measured values of power consumption under the following conditions.

- Operating conditions core supply (Figure 7-1)
  - $V_{VDDIN\_5} = V_{DDIN\_33} = 3.3V$
  - $V_{VDDCORE} = 1.85V$, supplied by the internal regulator
  - $V_{VDDIO1} = V_{VDDIO2} = V_{VDDIO3} = 3.3V$
  - $V_{VDDANA} = 3.3V$
  - Internal 3.3V regulator is off.
- $T_A = 25°C$
- I/Os are configured as inputs, with internal pull-up enabled.
- Oscillators
  - OSC0/1 (crystal oscillator) stopped
  - OSC32K (32KHz crystal oscillator) stopped
  - PLL0 running

7.5 I/O Pin Characteristics

Table 7-6. Normal I/O Pin Characteristics[1]

Symbol	Parameter	Condition		Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance	V_{VDD} = 3V		5		26	kOhm
		V_{VDD} = 5V		5		16	kOhm
$R_{PULLDOWN}$	Pull-down resistance			2		16	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3V				$0.3*V_{VDDIO}$	V
		V_{VDD} = 4.5V				$0.3*V_{VDDIO}$	
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V		$0.7*V_{VDDIO}$			V
		V_{VDD} = 5.5V		$0.7*V_{VDDIO}$			
V_{OL}	Output low-level voltage	I_{OL} = -3.5mA, pin drive x1[2]				0.45	V
		I_{OL} = -7mA, pin drive x2[2]					
		I_{OL} = -14mA, pin drive x4[2]					
V_{OH}	Output high-level voltage	I_{OH} = 3.5mA, pin drive x1[2]		V_{VDD} - 0.8			V
		I_{OH} = 7mA, pin drive x2[2]					
		I_{OH} = 14mA, pin drive x4[2]					
f_{MAX}	Output frequency[3]	V_{VDD} = 3.0V	load = 10pF, pin drive x1[2]			35	MHz
			load = 10pF, pin drive x2[2]			55	
			load = 10pF, pin drive x4[2]			70	
			load = 30pF, pin drive x1[2]			15	
			load = 30pF, pin drive x2[2]			30	
			load = 30pF, pin drive x4[2]			45	
		V_{VDD} =4.5V	load = 10pF, pin drive x1[2]			50	
			load = 10pF, pin drive x2[2]			80	
			load = 10pF, pin drive x4[2]			95	
			load = 30pF, pin drive x1[2]			25	
			load = 30pF, pin drive x2[2]			40	
			load = 30pF, pin drive x4[2]			65	

Table 7-6. Normal I/O Pin Characteristics[1]

Symbol	Parameter	Condition		Min	Typ	Max	Units
t_{RISE}	Rise time[3]	V_{VDD} = 3.0V	load = 10pF, pin drive x1[2]			7.7	ns
			load = 10pF, pin drive x2[2]			3.4	
			load = 10pF, pin drive x4[2]			1.9	
			load = 30pF, pin drive x1[2]			16	
			load = 30pF, pin drive x2[2]			7.5	
			load = 30pF, pin drive x4[2]			3.8	
		V_{VDD} = 4.5V	load = 10pF, pin drive x1[2]			5.3	
			load = 10pF, pin drive x2[2]			2.4	
			load = 10pF, pin drive x4[2]			1.3	
			load = 30pF, pin drive x1[2]			11.1	
			load = 30pF, pin drive x2[2]			5.2	
			load = 30pF, pin drive x4[2]			2.7	
t_{FALL}	Fall time[3]	V_{VDD} = 3.0V	load = 10pF, pin drive x1[2]			7.6	ns
			load = 10pF, pin drive x2[2]			3.5	
			load = 10pF, pin drive x4[2]			1.9	
			load = 30pF, pin drive x1[2]			15.8	
			load = 30pF, pin drive x2[2]			7.3	
			load = 30pF, pin drive x4[2]			3.8	
		V_{VDD} = 4.5V	load = 10pF, pin drive x1[2]			5.2	
			load = 10pF, pin drive x2[2]			2.4	
			load = 10pF, pin drive x4[2]			1.4	
			load = 30pF, pin drive x1[2]			10.9	
			load = 30pF, pin drive x2[2]			5.1	
			load = 30pF, pin drive x4[2]			2.7	
I_{LEAK}	Input leakage current	Pull-up resistors disabled				1.0	µA
C_{IN}	Input capacitance	PA00-PA29, PB00-PB31, PC00-PC01, PC08-PC31, PD00-PD30			7.5		pF
		PC02, PC03, PC04, PC05, PC06, PC07			2		

Note:
1. V_{VDD} corresponds to either V_{VDDIO1}, V_{VDDIO2}, V_{VDDIO3}, or V_{VDDANA}, depending on the supply for the pin. Refer to Section 3-1 on page 11 for details.
2. drive x1 capability pins are: PB00, PB01, PB02, PB03, PB30, PB31, PC02, PC03, PC04, PC05, PC06, PC07 - drive x2 /x4 capability pins are: PB06, PB21, PB26, PD02, PD06, PD13 - drive x1/x2 capability pins are the remaining PA, PB, PC, PD pins. The drive strength is programmable through ODCR0, ODCR0S, ODCR0C, ODCR0T registers of GPIO.
3. 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.

7.8 Analog Characteristics

7.8.1 1.8V Voltage Regulator Characteristics

Table 7-18. 1.8V Voltage Regulator Electrical Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{VDDIN_5}	Input voltage range	5V range	4.5		5.5	V
		3V range	3.0		3.6	
$V_{VDDCORE}$	Output voltage, calibrated value			1.85		V
I_{OUT}	DC output current				80	mA

Table 7-19. Decoupling Requirements

Symbol	Parameter	Condition	Typ	Techno.	Units
C_{IN1}	Input regulator capacitor 1		1	NPO	nF
C_{IN2}	Input regulator capacitor 2		4.7	X7R	uF
C_{OUT1}	Output regulator capacitor 1		470	NPO	pf
C_{OUT2}	Output regulator capacitor 2		2.2	X7R	uF

7.8.2 3.3V Voltage Regulator Characteristics

Table 7-20. 3.3V Voltage Regulator Electrical Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{VDDIN_5}	Input voltage range		4.5		5.5	V
V_{VDDIN_33}	Output voltage, calibrated value			3.4		V
I_{OUT}	DC output current				35	mA
I_{VREG}	Static current of regulator	Low power mode		10		µA

7.8.3 1.8V Brown Out Detector (BOD18) Characteristics

The values in Table 7-21 describe the values of the BOD.LEVEL in the SCIF module.

Table 7-21. BODLEVEL Values

BODLEVEL Value	Parameter	Min	Max	Units
0		1.29	1.58	V
20		1.36	1.63	
26	threshold at power-up sequence	1.42	1.69	
28		1.43	1.72	
32		1.48	1.77	
36		1.53	1.82	
40		1.56	1.88	

7.8.8 Analog Comparator Characteristics

Table 7-41. Analog Comparator Characteristics[1]

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Positive input voltage range		0		V_{VDDANA}	V
	Negative input voltage range		0		V_{VDDANA}	V
V_{OFFSET}	Offset	No hysteresis, Low Power mode	-29		29	mV
		No hysteresis, High Speed mode	-16		16	mV
V_{HYST}	Hysteresis	Low hysteresis, Low Power mode	7		44	mV
		Low hysteresis, High Speed mode	5		34	
		High hysteresis, Low Power mode	16		102	mV
		High hysteresis, High Speed mode	12		69	
t_{DELAY}	Propagation delay	Low Power mode			2.9	us
		High Speed mode			0.096	
$t_{STARTUP}$	Start-up time				20	µs

Note: 1. The measures are done without any I/O activity on VDDANA/GNDANA power domain.

Table 7-42. VDDANA scaled reference

Symbol	Parameter	Min	Typ	Max	Units
SCF	ACIFA.SCFi.SCF range	0		32	
V_{VDDANA} scaled			(64 - SCF) * V_{VDDANA} / 65		V
	V_{VDDANA} voltage accuracy			3.2	%

7.8.9 USB Transceiver Characteristics

7.8.9.1 Electrical Characteristics

Table 7-43. Electrical Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
R_{EXT}	Recommended external USB series resistor	In series with each USB pin with ±5%		39		Ω

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

7.9 Timing Characteristics

7.9.1 Startup, Reset, and Wake-up Timing

The startup, reset, and wake-up timings are calculated using the following formula:

$$t = t_{CONST} + N_{CPU} \times t_{CPU}$$

Where t_{CONST} and N_{CPU} are found in Table 7-44. t_{CONST} is the delay relative to RCSYS, t_{CPU} is the period of the CPU clock. If another clock source than RCSYS is selected as CPU clock the startup time of the oscillator, $t_{OSCSTART}$, must be added to the wake-up time in the stop, deepstop, and static sleep modes. Please refer to the source for the CPU clock in the "Oscillator Characteristics" on page 57 for more details about oscillator startup times.

Table 7-44. Maximum Reset and Wake-up Timing

Parameter		Measuring	Max t_{CONST} (in µs)	Max N_{CPU}
Startup time from power-up, using regulator		VDDIN_5 rising (10 mV/ms) Time from V_{VDDIN_5}=0 to the first instruction entering the decode stage of CPU. VDDCORE is supplied by the internal regulator.	2600	0
Startup time from reset release		Time from releasing a reset source (except POR, BOD18, and BOD33) to the first instruction entering the decode stage of CPU.	1240	0
Wake-up	Idle	From wake-up event to the first instruction entering the decode stage of the CPU.	0	19
	Frozen		268	209
	Standby		268	209
	Stop		268+ $t_{OSCSTART}$	212
	Deepstop		268+ $t_{OSCSTART}$	212
	Static		268+ $t_{OSCSTART}$	212

Maximum SPI Frequency, Master Output

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

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

Where $SPIn$ is the MOSI delay, USPI2 or USPI5 depending on CPOL and NCPHA. 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. f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

Maximum SPI Frequency, Master Input

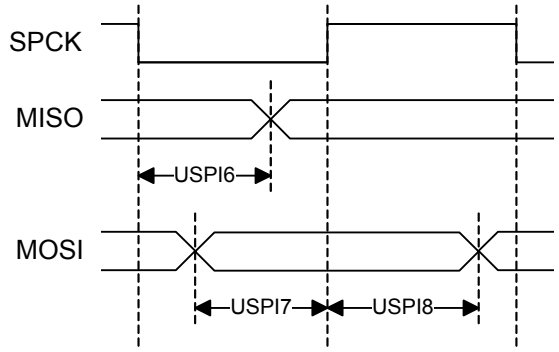
The maximum SPI master input frequency is given by the following formula:

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

Where $SPIn$ is the MISO setup and hold time, USPI0 + USPI1 or USPI3 + USPI4 depending on CPOL and NCPHA. T_{VALID} is the SPI slave response time. Please refer to the SPI slave datasheet for T_{VALID} . f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

7.9.3.2 Slave mode

Figure 7-8. USART in SPI Slave Mode With (CPOL= 0 and CPHA= 1) or (CPOL= 1 and CPHA= 0)



Maximum SPI Frequency, Slave Input Mode

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

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

Where $SPIn$ is the MOSI setup and hold time, USPI7 + USPI8 or USPI10 + USPI11 depending on CPOL and NCPHA. f_{CLKSPI} is the maximum frequency of the CLK_SPI. Refer to the SPI chapter for a description of this clock.

Maximum SPI Frequency, Slave Output Mode

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

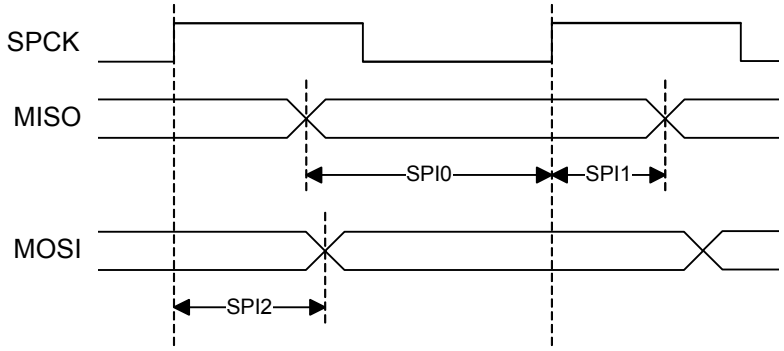
$$f_{SPCKMAX} = MIN(\frac{f_{CLKSPI} \times 2}{9}, f_{PINMAX}, \frac{1}{SPIn + t_{SETUP}})$$

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 masterdatasheet 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.

7.9.4 SPI Timing

7.9.4.1 Master mode

Figure 7-11. SPI Master Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)



7.9.7 EBI Timings

See EBI I/O lines description for more details.

Table 7-52. SMC Clock Signal.

Symbol	Parameter	Max[1]	Units
$1/(t_{CPSMC})$	SMC Controller clock frequency	f_{cpu}	MHz

Note: 1. The maximum frequency of the SMC interface is the same as the max frequency for the HSB.

Table 7-53. SMC Read Signals with Hold Settings[1]

Symbol	Parameter	Conditions	Min	Units
NRD Controlled (READ_MODE = 1)				
SMC_1	Data setup before NRD high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	32.5	ns
SMC_2	Data hold after NRD high		0	
SMC_3	NRD high to NBS0/A0 change[2]		nrd hold length * t_{CPSMC} - 1.5	
SMC_4	NRD high to NBS1 change[2]		nrd hold length * t_{CPSMC} - 0	
SMC_5	NRD high to NBS2/A1 change[2]		nrd hold length * t_{CPSMC} - 0	
SMC_7	NRD high to A2 - A25 change[2]		nrd hold length * t_{CPSMC} - 5.6	
SMC_8	NRD high to NCS inactive[2]		(nrd hold length - ncs rd hold length) * t_{CPSMC} - 1.3	
SMC_9	NRD pulse width		nrd pulse length * t_{CPSMC} - 0.6	
NRD Controlled (READ_MODE = 0)				
SMC_{10}	Data setup before NCS high	V_{VDD} = 3.0V, drive strength of the pads set to the lowest, external capacitor = 40pF	34.1	ns
SMC_{11}	Data hold after NCS high		0	
SMC_{12}	NCS high to NBS0/A0 change[2]		ncs rd hold length * t_{CPSMC} - 3	
SMC_{13}	NCS high to NBS0/A0 change[2]		ncs rd hold length * t_{CPSMC} - 2	
SMC_{14}	NCS high to NBS2/A1 change[2]		ncs rd hold length * t_{CPSMC} - 1.1	
SMC_{16}	NCS high to A2 - A25 change[2]		ncs rd hold length * t_{CPSMC} - 7.2	
SMC_{17}	NCS high to NRD inactive[2]		(ncs rd hold length - nrd hold length) * t_{CPSMC} - 2.2	
SMC_{18}	NCS pulse width		ncs rd pulse length * t_{CPSMC} - 3	

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.

2. hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs rd hold length” or “nrd hold length”.

10. Errata

10.1 rev E

10.1.1 ADCIFA

1 **ADCREFP/ADCREFN can not be selected as an external ADC reference by setting the ADCIFA.CFG.EXREF bit to one**
Fix/Workaround
A voltage reference can be applied on ADCREFP/ADCREFN pins if the ADCIFA.CFG.EXREF bit is set to zero, the ADCIFA.CFG.RS bit is set to zero and the voltage reference applied on ADCREFP/ADCREFN pins is higher than the internal 1V reference.

10.1.2 AST

1 **AST wake signal is released one AST clock cycle after the BUSY bit is cleared**
After writing to the Status Clear Register (SCR) the wake signal is released one AST clock cycle after the BUSY bit in the Status Register (SR.BUSY) is cleared. If entering sleep mode directly after the BUSY bit is cleared the part will wake up immediately.
Fix/Workaround
Read the Wake Enable Register (WER) and write this value back to the same register. Wait for BUSY to clear before entering sleep mode.

10.1.3 aWire

1 **aWire MEMORY_SPEED_REQUEST command does not return correct CV**
The aWire MEMORY_SPEED_REQUEST command does not return a CV corresponding to the formula in the aWire Debug Interface chapter.
Fix/Workaround
Issue a dummy read to address 0x100000000 before issuing the MEMORY_SPEED_REQUEST command and use this formula instead:

$$f_{sab} = \frac{7 f_{aw}}{CV - 3}$$

10.1.4 Power Manager

1 **TWIS may not wake the device from sleep mode**
If the CPU is put to a sleep mode (except Idle and Frozen) directly after a TWI Start condition, the CPU may not wake upon a TWIS address match. The request is NACKed.
Fix/Workaround
When using the TWI address match to wake the device from sleep, do not switch to sleep modes deeper than Frozen. Another solution is to enable asynchronous EIC wake on the TWIS clock (TWCK) or TWIS data (TWD) pins, in order to wake the system up on bus events.

10.2.10 TWIS

1. **Clearing the NAK bit before the BTF bit is set locks up the TWI bus**
When the TWIS is in transmit mode, clearing the NAK Received (NAK) bit of the Status Register (SR) before the end of the Acknowledge/Not Acknowledge cycle will cause the TWIS to attempt to continue transmitting data, thus locking up the bus.
Fix/Workaround
Clear SR.NAK only after the Byte Transfer Finished (BTF) bit of the same register has been set.

2. **TWIS stretch on Address match error**
When the TWIS stretches TWCK due to a slave address match, it also holds TWD low for the same duration if it is to be receiving data. When TWIS releases TWCK, it releases TWD at the same time. This can cause a TWI timing violation.
Fix/Workaround
None.

3. **TWALM forced to GND**
The TWALM pin is forced to GND when the alternate function is selected and the TWIS module is enabled.
Fix/Workaround
None.

10.2.11 USBC

1. **UPINRQx.INRQ field is limited to 8-bits**
In Host mode, when using the UPINRQx.INRQ feature together with the multi-packet mode to launch a finite number of packet among multi-packet, the multi-packet size (located in the descriptor table) is limited to the UPINRQx.INRQ value multiply by the pipe size.
Fix/Workaround
UPINRQx.INRQ value shall be less than the number of configured multi-packet.

2. **In USB host mode, downstream resume feature does not work (UHCON.RESUME=1).**
Fix/Workaround
None.

3. **In host mode, the disconnection during OUT transition is not supported**
In USB host mode, a pipe can not work if the previous USB device disconnection has occurred during a USB transfer.
Fix/Workaround
Reset the USBC (USBCON.USB=0 and =1) after a device disconnection (UHINT.DDISCI).

4. **In USB host mode, entering suspend mode can fail**
In USB host mode, entering suspend mode can fail when UHCON.SOFE=0 is done just after a SOF reception (UHINT.HSOFI).
Fix/Workaround
Check that UHNUM.FLENHIGH is below 185 in Full speed and below 21 in Low speed before clearing UHCON.SOFE.

5. **In USB host mode, entering suspend mode for low speed device can fail when the USB freeze (USBCON.FRZCLK=1) is done just after UHCON.SOFE=0.**
Fix/Workaround
When entering suspend mode (UHCON.SOFE is cleared), check that USBFSM.DRDSTATE is not equal to three before freezing the clock (USBCON.FRZCLK=1).

Table of Contents