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	64KB (64K 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/at32uc3l064-aut

The Frequency Meter (FREQM) allows accurate measuring of a clock frequency by comparing it to a known reference clock.

The device includes six identical 16-bit Timer/Counter (TC) channels. Each channel can be independently programmed to perform frequency measurement, event counting, interval measurement, pulse generation, delay timing, and pulse width modulation.

The Pulse Width Modulation controller (PWMA) provides 8-bit PWM channels which can be synchronized and controlled from a common timer. One PWM channel is available for each I/O pin on the device, enabling applications that require multiple PWM outputs, such as LCD backlight control. The PWM channels can operate independently, with duty cycles set independently from each other, or in interlinked mode, with multiple channels changed at the same time.

The AT32UC3L016/32/64 also features many communication interfaces, like USART, SPI, and TWI, for communication intensive applications. The USART supports different communication modes, like SPI Mode and LIN Mode.

A general purpose 8-channel ADC is provided, as well as eight analog comparators (AC). The ADC can operate in 10-bit mode at full speed or in enhanced mode at reduced speed, offering up to 12-bit resolution. The ADC also provides an internal temperature sensor input channel. The analog comparators can be paired to detect when the sensing voltage is within or outside the defined reference window.

The Capacitive Touch (CAT) module senses touch on external capacitive touch sensors, using the QTouch technology. Capacitive touch sensors use no external mechanical components, unlike normal push buttons, and therefore demand less maintenance in the user application. The CAT module allows up to 17 touch sensors, or up to 16 by 8 matrix sensors to be interfaced. One touch sensor can be configured to operate autonomously without software interaction, allowing wakeup from sleep modes when activated.

Atmel offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys as well as Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop, and debug your own touch applications.

The AT32UC3L016/32/64 integrates a class 2+ Nexus 2.0 On-chip Debug (OCD) System, with non-intrusive real-time trace and full-speed read/write memory access, in addition to basic run-time control. The NanoTrace interface enables trace feature for aWire- or JTAG-based debuggers. The single-pin aWire interface allows all features available through the JTAG interface to be accessed through the RESET pin, allowing the JTAG pins to be used for GPIO or peripherals.

3.2 Peripheral Multiplexing on I/O lines

3.2.1 Multiplexed signals

Each GPIO line can be assigned to one of the peripheral functions.The following table describes the peripheral signals multiplexed to the GPIO lines.

Table 3-1. GPIO Controller Function Multiplexing

48-pin	PIN	GPIO	Supply	Pin Type	GPIO Function							
					A	B	C	D	E	F	G	H
11	PA00	0	VDDIO	Normal I/O	USART0 TXD	USART1 RTS	SPI NPCS[2]		PWMA PWMA[0]		SCIF GCLK[0]	CAT CSA[2]
14	PA01	1	VDDIO	Normal I/O	USART0 RXD	USART1 CTS	SPI NPCS[3]	USART1 CLK	PWMA PWMA[1]	ACIFB ACAP[0]	TWIMS0 TWALM	CAT CSA[1]
13	PA02	2	VDDIO	High-drive I/O	USART0 RTS	ADCIFB TRIGGER	USART2 TXD	TC0 A0	PWMA PWMA[2]	ACIFB ACBP[0]	USART0 CLK	CAT CSA[3]
4	PA03	3	VDDIO	Normal I/O	USART0 CTS	SPI NPCS[1]	USART2 TXD	TC0 B0	PWMA PWMA[3]	ACIFB ACBN[3]	USART0 CLK	CAT CSB[3]
28	PA04	4	VDDIO	Normal I/O	SPI MISO	TWIMS0 TWCK	USART1 RXD	TC0 B1	PWMA PWMA[4]	ACIFB ACBP[1]		CAT CSA[7]
12	PA05	5	VDDIO	Normal I/O (TWI)	SPI MOSI	TWIMS1 TWCK	USART1 TXD	TC0 A1	PWMA PWMA[5]	ACIFB ACBN[0]	TWIMS0 TWD	CAT CSB[7]
10	PA06	6	VDDIO	High-drive I/O, 5V tolerant	SPI SCK	USART2 TXD	USART1 CLK	TC0 B0	PWMA PWMA[6]		SCIF GCLK[1]	CAT CSB[1]
15	PA07	7	VDDIO	Normal I/O (TWI)	SPI NPCS[0]	USART2 RXD	TWIMS1 TWALM	TWIMS0 TWCK	PWMA PWMA[7]	ACIFB ACAN[0]	EIC EXTINT[0]	CAT CSB[2]
3	PA08	8	VDDIO	High-drive I/O	USART1 TXD	SPI NPCS[2]	TC0 A2	ADCIFB ADP[0]	PWMA PWMA[8]			CAT CSA[4]
2	PA09	9	VDDIO	High-drive I/O	USART1 RXD	SPI NPCS[3]	TC0 B2	ADCIFB ADP[1]	PWMA PWMA[9]	SCIF GCLK[2]	EIC EXTINT[1]	CAT CSB[4]
46	PA10	10	VDDIO	Normal I/O	TWIMS0 TWD		TC0 A0		PWMA PWMA[10]	ACIFB ACAP[1]	SCIF GCLK[2]	CAT CSA[5]
27	PA11	11	VDDIN	Normal I/O					PWMA PWMA[11]			
47	PA12	12	VDDIO	Normal I/O		USART2 CLK	TC0 CLK1	CAT SMP	PWMA PWMA[12]	ACIFB ACAN[1]	SCIF GCLK[3]	CAT CSB[5]
26	PA13	13	VDDIN	Normal I/O	GLOC OUT[0]	GLOC IN[7]	TC0 A0	SCIF GCLK[2]	PWMA PWMA[13]	CAT SMP	EIC EXTINT[2]	CAT CSA[0]
36	PA14	14	VDDIO	Normal I/O	ADCIFB AD[0]	TC0 CLK2	USART2 RTS	CAT SMP	PWMA PWMA[14]		SCIF GCLK[4]	CAT CSA[6]
37	PA15	15	VDDIO	Normal I/O	ADCIFB AD[1]	TC0 CLK1		GLOC IN[6]	PWMA PWMA[15]	CAT SYNC	EIC EXTINT[3]	CAT CSB[6]
38	PA16	16	VDDIO	Normal I/O	ADCIFB AD[2]	TC0 CLK0		GLOC IN[5]	PWMA PWMA[16]	ACIFB ACREFN	EIC EXTINT[4]	CAT CSA[8]
39	PA17	17	VDDIO	Normal I/O (TWI)		TC0 A1	USART2 CTS	TWIMS1 TWD	PWMA PWMA[17]	CAT SMP	CAT DIS	CAT CSB[8]
41	PA18	18	VDDIO	Normal I/O	ADCIFB AD[4]	TC0 B1		GLOC IN[4]	PWMA PWMA[18]	CAT SYNC	EIC EXTINT[5]	CAT CSB[0]

Refer to the "TWI Pin Characteristics(1)" on page 49 for a description of the electrical properties of the TWI, 5V Tolerant, and SMBUS pins.

3.2.2 Peripheral Functions

Each GPIO line can be assigned to one of several peripheral functions. The following table describes how the various peripheral functions are selected. The last listed function has priority in case multiple functions are enabled on the same pin.

Table 3-2. Peripheral Functions

Function	Description
GPIO Controller Function multiplexing	GPIO and GPIO peripheral selection A to H
Nexus OCD AUX port connections	OCD trace system
aWire DATAOUT	aWire output in two-pin mode
JTAG port connections	JTAG debug port
Oscillators	OSC0, OSC32

3.2.3 JTAG Port Connections

If the JTAG is enabled, the JTAG will take control over a number of pins, irrespectively of the I/O Controller configuration.

Table 3-3. JTAG Pinout

48-pin	Pin Name	JTAG Pin
11	PA00	TCK
14	PA01	TMS
13	PA02	TDO
4	PA03	TDI

3.2.4 Nexus OCD AUX Port Connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespectively of the I/O Controller configuration. Two different OCD trace pin mappings are possible, depending on the configuration of the OCD AXS register. For details, see the AVR32 UC Technical Reference Manual.

Table 3-4. Nexus OCD AUX Port Connections

Pin	AXS=1	AXS=0
EVTI_N	PA05	PB08
MDO[5]	PA10	PB00
MDO[4]	PA18	PB04
MDO[3]	PA17	PB05
MDO[2]	PA16	PB03

Table 3-4. Nexus OCD AUX Port Connections

Pin	AXS=1	AXS=0
MDO[1]	PA15	PB02
MDO[0]	PA14	PB09
EVTO_N	PA04	PA04
MCKO	PA06	PB01
MSEO[1]	PA07	PB11
MSEO[0]	PA11	PB12

3.2.5 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the System Control Interface (SCIF). Please refer to the SCIF chapter for more information about this.

Table 3-5. Oscillator Pinout

48-pin	Pin	Oscillator Function
3	PA08	XIN0
46	PA10	XIN32
26	PA13	XIN32_2
2	PA09	XOUT0
47	PA12	XOUT32
25	PA20	XOUT32_2

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. The WAKE_N pin is always enabled. Please refer to Section 6.1.4 on page 40 for constraints on the WAKE_N pin.

Table 3-6. Other Functions

48-pin	Pin	Function
27	PA11	WAKE_N
22	RESET_N	aWire DATA
11	PA00	aWire DATAOUT

single cycle. Load and store instructions have several different formats in order to reduce code size and speed up execution.

The register file is organized as sixteen 32-bit registers and includes the Program Counter, the Link Register, and the Stack Pointer. In addition, register R12 is designed to hold return values from function calls and is used implicitly by some instructions.

4.3 The AVR32UC CPU

The AVR32UC CPU targets low- and medium-performance applications, and provides an advanced On-Chip Debug (OCD) system, no caches, and a Memory Protection Unit (MPU). Java acceleration hardware is not implemented.

AVR32UC provides three memory interfaces, one High Speed Bus master for instruction fetch, one High Speed Bus master for data access, and one High Speed Bus slave interface allowing other bus masters to access data RAMs internal to the CPU. Keeping data RAMs internal to the CPU allows fast access to the RAMs, reduces latency, and guarantees deterministic timing. Also, power consumption is reduced by not needing a full High Speed Bus access for memory accesses. A dedicated data RAM interface is provided for communicating with the internal data RAMs.

A local bus interface is provided for connecting the CPU to device-specific high-speed systems, such as floating-point units and I/O controller ports. This local bus has to be enabled by writing a one to the LOCEN bit in the CPUCR system register. The local bus is able to transfer data between the CPU and the local bus slave in a single clock cycle. The local bus has a dedicated memory range allocated to it, and data transfers are performed using regular load and store instructions. Details on which devices that are mapped into the local bus space is given in the CPU Local Bus section in the Memories chapter.

Figure 4-1 on page 20 displays the contents of AVR32UC.

6.1.3.2 *1.8V Single Supply Mode*

In 1.8V single supply mode the internal regulator is not used, and VDDIO and VDDCORE are powered by a single 1.8V supply as shown in Figure 6-3. All I/O lines will be powered by the same power (VDDIN = VDDIO = VDDCORE).

Figure 6-3. 1.8V Single Supply Mode.

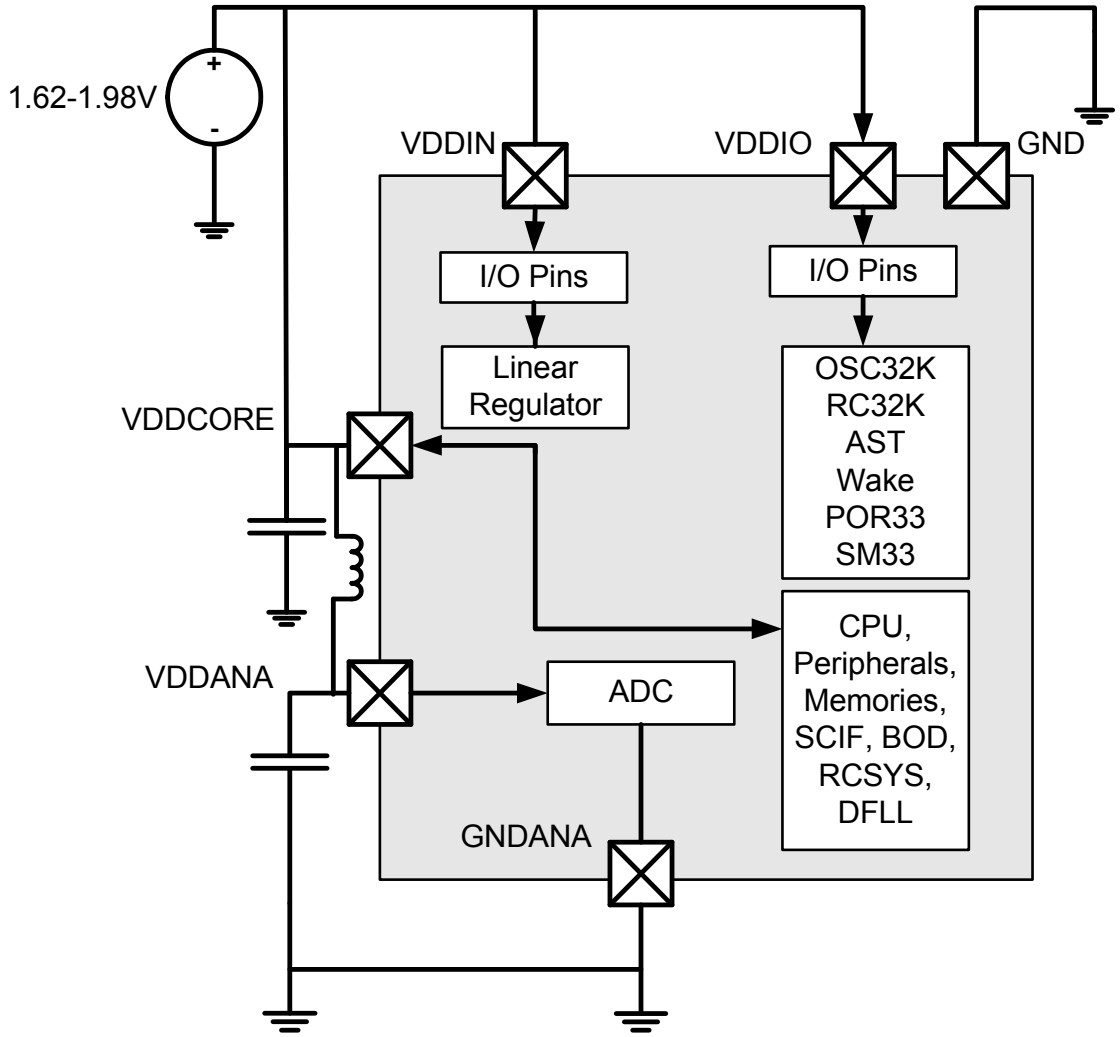


Table 7-3. Supply Rise Rates and Order[1]

Symbol	Parameter	Rise Rate			
		Min	Max	Unit	Comment
V_{VDDIO}	DC supply peripheral I/Os	0	2.5	V/µs	
V_{VDDIN}	DC supply peripheral I/Os and internal regulator	0.002	2.5	V/µs	Slower rise time requires external power-on reset circuit.
$V_{VDDCORE}$	DC supply core	0	2.5	V/µs	Rise before or at the same time as VDDIO
V_{VDDANA}	Analog supply voltage	0	2.5	V/µs	Rise together with VDDCORE

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.

7.3 Maximum Clock Frequencies

These parameters are given in the following conditions:

- $V_{VDDCORE}$ = 1.62V to 1.98V
- Temperature = -40°C to 85°C

Table 7-4. Clock Frequencies

Symbol	Parameter	Conditions	Min	Max	Units
f_{CPU}	CPU clock frequency			50	MHz
f_{PBA}	PBA clock frequency			50	MHz
f_{PBB}	PBB clock frequency			50	MHz
f_{GCLK0}	GCLK0 clock frequency	DFLLIF main reference, GCLK0 pin		150	MHz
f_{GCLK1}	GCLK1 clock frequency	DFLLIF dithering and ssg reference, GCLK1 pin		150	MHz
f_{GCLK2}	GCLK2 clock frequency	AST, GCLK2 pin		80	MHz
f_{GCLK3}	GCLK3 clock frequency	PWMA, GCLK3 pin		110	MHz
f_{GCLK4}	GCLK4 clock frequency	CAT, ACIFB, GCLK4 pin		110	MHz
f_{GCLK5}	GCLK5 clock frequency	GLOC		80	MHz

7.4 Power Consumption

The values in Table 7-5 are measured values of power consumption under the following conditions, except where noted:

- Operating conditions internal core supply (Figure 7-1) - this is the default configuration
  - V_{VDDIN} = 3.0V
  - $V_{VDDCORE}$ = 1.62V, supplied by the internal regulator

Table 7-8. High-drive I/O Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		0.3*V_{VDD}	V
		V_{VDD} = 1.62V	-0.3		0.3*V_{VDD}	
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	0.7*V_{VDD}		V_{VDD} + 0.3	V
		V_{VDD} = 1.98V	0.7*V_{VDD}		V_{VDD} + 0.3	
V_{OL}	Output low-level voltage	V_{VDD} = 3.0V, I_{OL} = 6mA			0.4	V
		V_{VDD} = 1.62V, I_{OL} = 4mA			0.4	
V_{OH}	Output high-level voltage	V_{VDD} = 3.0V, I_{OH} = 6mA	V_{VDD}-0.4			V
		V_{VDD} = 1.62V, I_{OH} = 4mA	V_{VDD}-0.4			
f_{MAX}	Output frequency, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			45	MHz
		V_{VDD} = 3.0V, load = 30pF			23	
t_{RISE}	Rise time, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			4.7	ns
		V_{VDD} = 3.0V, load = 30pF			11.5	
t_{FALL}	Fall time, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			4.8	
		V_{VDD} = 3.0V, load = 30pF			12	
f_{MAX}	Output frequency, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			52	MHz
		V_{VDD} = 3.0V, load = 30pF			39	
t_{RISE}	Rise time, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			2.9	ns
		V_{VDD} = 3.0V, load = 30pF			4.9	
t_{FALL}	Fall time, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			2.5	
		V_{VDD} = 3.0V, load = 30pF			4.6	
I_{LEAK}	Input leakage current	Pull-up resistors disabled			1	µA
C_{IN}	Input capacitance, all High-drive I/O pins, except PA08 and PA09	TQFP48 package		2,2		pF
		QFN48 package		2.0		
		TLLGA 48 package		2.0		
C_{IN}	Input capacitance, PA08 and PA09	TQFP48 package		7.0		
		QFN48 package		6.7		
		TLLGA 48 package		6.7		

Notes: 1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2 on page 9 for details.
2. 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.

Table 7-9. High-drive I/O, 5V Tolerant, Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance		30	50	110	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		0.3*V_{VDD}	V
		V_{VDD} = 1.62V	-0.3		0.3*V_{VDD}	

7.6.5 32kHz RC Oscillator (RC32K) Characteristics

Table 7-16. 32kHz RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency[1]		20	32	44	kHz
I_{RC32K}	Current consumption			0.6		µA
$t_{STARTUP}$	Startup time			100		µs

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.

7.6.6 System RC Oscillator (RCSYS) Characteristics

Table 7-17. System RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency	Calibrated at 85°C	111.6	115	118.4	kHz

7.7 Flash Characteristics

Table 7-18 gives the device maximum operating frequency depending on the number of flash wait states and the flash read mode. The FSW bit in the FLASHCDW FSR register controls the number of wait states used when accessing the flash memory.

Table 7-18. Maximum Operating Frequency

Flash Wait States	Read Mode	Maximum Operating Frequency
1	High speed read mode	50MHz
0		25MHz
1	Normal read mode	30MHz
0		15MHz

Table 7-19. Flash Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{FPP}	Page programming time	f_{CLK_HSB} = 50MHz		5		ms
t_{FPE}	Page erase time			5		
t_{FFP}	Fuse programming time			1		
t_{FEA}	Full chip erase time (EA)			5		
t_{FCE}	JTAG chip erase time (CHIP_ERASE)	f_{CLK_HSB} = 115kHz		170		

7.8.6 Analog to Digital Converter Characteristics

Table 7-28. ADC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{ADC}	ADC clock frequency	10-bit resolution mode			6	MHz
		8-bit resolution mode			6	
$t_{STARTUP}$	Startup time	Return from Idle Mode		15		µs
t_{CONV}	Conversion time (latency)	f_{ADC} = 6MHz	11		26	cycles
	Throughput rate	V_{VDD} > 3.0V, f_{ADC} = 6MHz, 10-bit resolution mode, low impedance source			460	kSPS
		V_{VDD} > 3.0V, f_{ADC} = 6MHz, 8-bit resolution mode, low impedance source			460	
$V_{ADVREFP}$	Reference voltage range	$V_{ADVREFP}$ = V_{VDDANA}	1.62		1.98	V
I_{ADC}	Current consumption on V_{VDDANA}	ADC Clock = 6MHz		300		µA
$I_{ADVREFP}$	Current consumption on ADVREFP pin	f_{ADC} = 6MHz		250		

Note: 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.

7.8.6.1 Inputs and Sample and Hold Aquisition Time

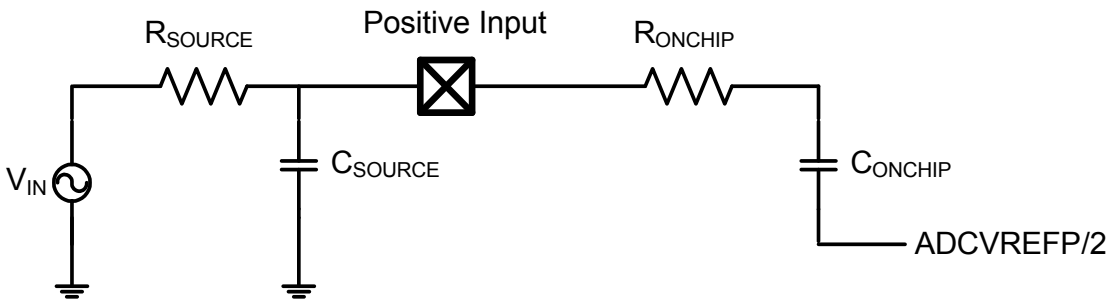
Table 7-29. Analog Inputs

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{ADn}	Input Voltage Range	10-bit mode	0		$V_{ADVREFP}$	V
		8-bit mode				
C_{ONCHIP}	Internal Capacitance[1]				21.5	pF
R_{ONCHIP}	Internal Resistance[1]	V_{VDDIO} = 3.0V to 3.6V, $V_{VDDCORE}$ = 1.8V			2.55	kOhm
		V_{VDDIO} = $V_{VDDCORE}$ = 1.62V to 1.98V			55.3	

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.

An analog voltage input must be able to charge the sample and hold (S/H) capacitor in the ADC in order to achieve maximum accuracy. Seen externally the ADC input consists of a resistor (R_{ONCHIP}) and a capacitor (C_{ONCHIP}). In addition the resistance (R_{SOURCE}) and capacitance (C_{SOURCE}) of the PCB and source must be taken into account when calculating the sample and hold time. Figure 7-7 shows the ADC input channel equivalent circuit.

Figure 7-7. ADC Input



The minimum sample and hold time (in ns) can be found using this formula:

$$t_{SAMPLEHOLD} \geq (R_{ONCHIP} + R_{OFFCHIP}) \times (C_{ONCHIP} + C_{OFFCHIP}) \times \ln(2^{n+1})$$

Where n is the number of bits in the conversion. $t_{SAMPLEHOLD}$ is defined by the SHTIM field in the ADCIFB ACR register. Please refer to the ADCIFB chapter for more information.

7.8.6.2 Applicable Conditions and Derating Data

Table 7-30. Transfer Characteristics 10-bit Resolution Mode

Parameter	Conditions	Min	Typ	Max	Units
Resolution			10		Bit
Integral non-linearity	ADC clock frequency = 6MHz		+/-2		LSB
Differential non-linearity		-0.9		1	
Offset error			+/-4		
Gain error			+/-4		

Table 7-31. Transfer Characteristics 8-bit Resolution Mode

Parameter	Conditions	Min	Typ	Max	Units
Resolution			8		Bit
Integral non-linearity	ADC clock frequency = 6MHz		+/-0.5		LSB
Differential non-linearity		-0.23		0.25	
Offset error			+/-1		
Gain error			+/-1		

7.9.3 USART in SPI Mode Timing

7.9.3.1 Master mode

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

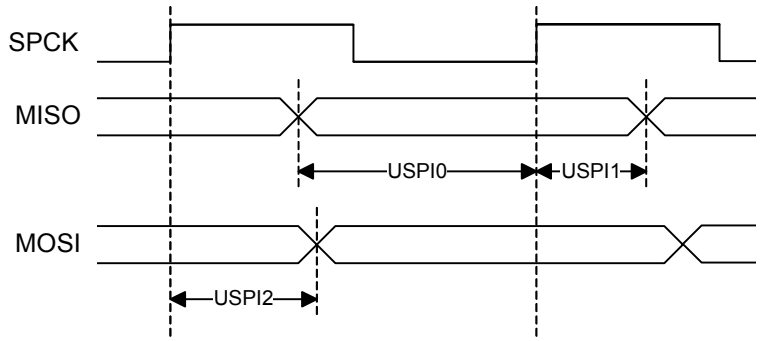


Figure 7-9. USART in SPI Master Mode With (CPOL= 0 and CPHA= 1) or (CPOL= 1 and CPHA= 0)

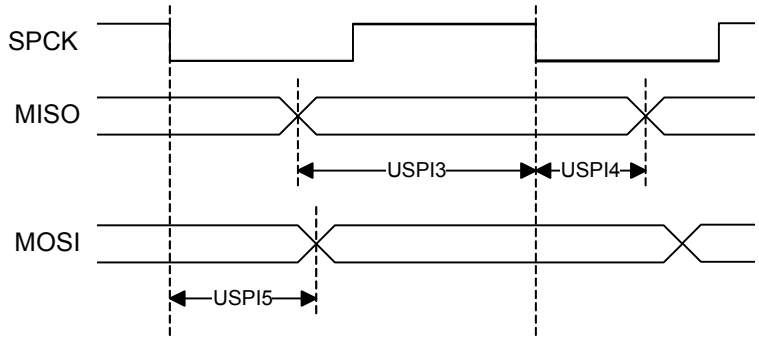


Table 7-38. USART in SPI Mode Timing, Master Mode[1]

Symbol	Parameter	Conditions	Min	Max	Units
USPI0	MISO setup time before SPCK rises	V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF	30.0+ t_{SAMPLE}[2]		ns
USPI1	MISO hold time after SPCK rises		0		
USPI2	SPCK rising to MOSI delay			8.5	
USPI3	MISO setup time before SPCK falls		25.5 + t_{SAMPLE}[2]		
USPI4	MISO hold time after SPCK falls		0		
USPI5	SPCK falling to MOSI delay			13.6	

Notes: 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. Where: $t_{SAMPLE} = t_{SPCK} - \left(\left\lfloor \frac{t_{SPCK}}{2 \times t_{CLKUSART}} \right\rfloor \frac{1}{2} \right) \times t_{CLKUSART}$

Figure 7-12. USART in SPI Slave Mode NPCS Timing

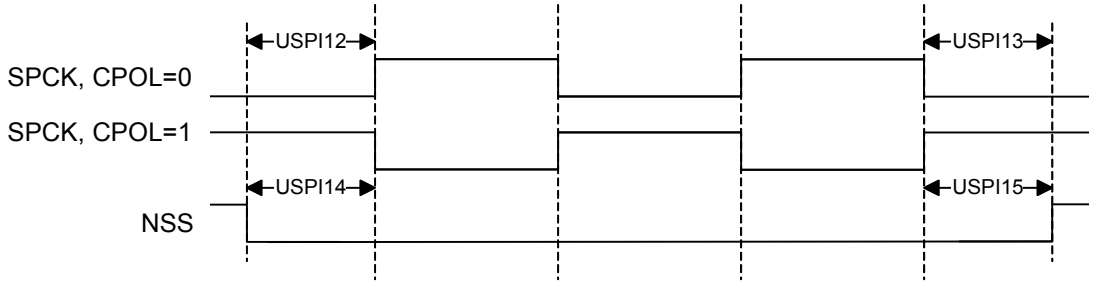


Table 7-39. USART in SPI mode Timing, Slave Mode[1]

Symbol	Parameter	Conditions	Min	Max	Units
USPI6	SPCK falling to MISO delay	V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF		27.6	ns
USPI7	MOSI setup time before SPCK rises		t_{SAMPLE}[2] + t_{CLK_USART}		
USPI8	MOSI hold time after SPCK rises		0		
USPI9	SPCK rising to MISO delay			27.2	
USPI10	MOSI setup time before SPCK falls		t_{SAMPLE}[2] + t_{CLK_USART}		
USPI11	MOSI hold time after SPCK falls		0		
USPI12	NSS setup time before SPCK rises		25.0		
USPI13	NSS hold time after SPCK falls		0		
USPI14	NSS setup time before SPCK falls		25.0		
USPI15	NSS hold time after SPCK rises		0		

Notes: 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. Where: $t_{SAMPLE} = t_{SPCK} - \left(\left\lfloor \frac{t_{SPCK}}{2 \times t_{CLKUSART}} \right\rfloor + \frac{1}{2} \right) \times t_{CLKUSART}$

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 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.

7.9.4 SPI Timing

7.9.4.1 Master mode

Figure 7-13. SPI Master Mode With (CPOL= NCPHA= 0) or (CPOL= NCPHA= 1)

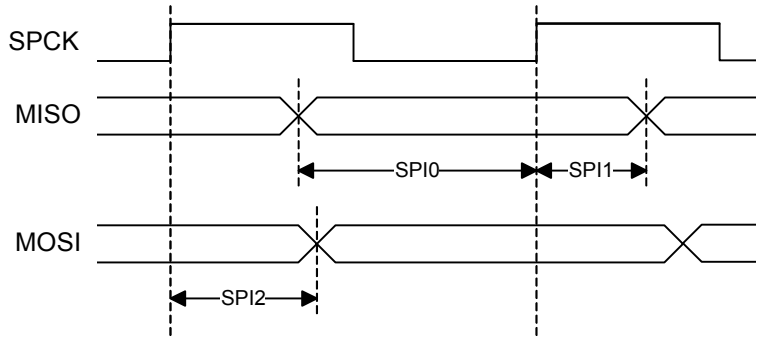
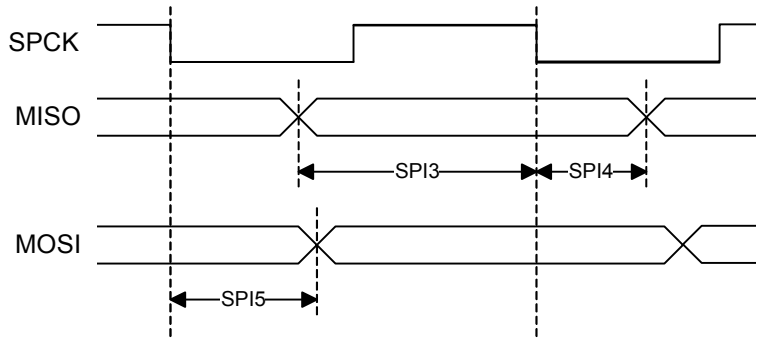


Figure 7-14. SPI Master Mode With (CPOL= 0 and NCPHA= 1) or (CPOL= 1 and NCPHA= 0)



If the CPU is in idle or frozen sleep mode and a module is in a state that triggers sleep walking, all PB clocks will be masked except the PB clock to the sleepwalking module.
Fix/Workaround
Mask all clock requests in the PM.PPCR register before going into idle or frozen mode.

10.1.4 SCIF

1. **PCLKSR.OSC32RDY bit might not be cleared after disabling OSC32K**
In some cases the OSC32RDY bit in the PCLKSR register will not be cleared when OSC32K is disabled.
Fix/Workaround
When re-enabling the OSC32K, read the PCLKSR.OSC32RDY bit. If this bit is:
0: Follow normal procedures.
1: Ignore the PCLKSR.OSC32RDY and ISR.OSC32RDY bit. Use the Frequency Meter (FREQM) to determine if the OSC32K clock is ready. The OSC32K clock is ready when the FREQM measures a non-zero frequency.

2. **The RC32K output on PA20 is not always permanently disabled**
The RC32K output on PA20 may sometimes re-appear.
Fix/Workaround
Before using RC32K for other purposes, the following procedure has to be followed in order to properly disable it:
- Run the CPU on RCSYS
- Disable the output to PA20 by writing a zero to PM.PPCR.RC32OUT
- Enable RC32K by writing a one to SCIF.RC32KCR.EN, and wait for this bit to be read as one
- Disable RC32K by writing a zero to SCIF.RC32KCR.EN, and wait for this bit to be read as zero.

10.1.5 AST

1. **Reset may set status bits in the AST**
If a reset occurs and the AST is enabled, the SR.ALARM0, SR.PER0, and SR.OVF bits may be set.
Fix/Workaround
If the part is reset and the AST is used, clear all bits in the Status Register before entering sleep mode.

2. **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.

10.1.6 WDT

1. **Clearing the Watchdog Timer (WDT) counter in second half of timeout period will issue a Watchdog reset**
If the WDT counter is cleared in the second half of the timeout period, the WDT will immediately issue a Watchdog reset.
Fix/Workaround

None.

2. **The aWire debug interface is reset after leaving Shutdown mode**
   If the aWire debug mode is used as debug interface and the program enters Shutdown mode, the aWire interface will be reset when the part receives a wakeup either from the WAKE_N pin or the AST.
   Fix/Workaround
   None.

10.2.14 CHIP

1. **In 3.3V Single Supply Mode, the Analog Comparator inputs affects the device's ability to start**
   When using the 3.3V Single Supply Mode the state of the Analog Comparator input pins can affect the device's ability to release POR reset. This is due to an interaction between the Analog Comparator input pins and the POR circuitry. The issue is not present in the 1.8V Supply Mode or the 3.3V Supply Mode with 1.8V Regulated I/O Lines.
   Fix/Workaround
   ACREFN (pin PA16) must be connected to GND until the POR reset is released and the Analog Comparator inputs should not be driven higher than 1.0V until the POR reset is released.

2. **Increased Power Consumption in VDDIO in sleep modes**
   If OSC0 is enabled in crystal mode when entering a sleep mode where the OSC0 is disabled, this will lead to an increased power consumption in VDDIO.
   Fix/Workaround
   Solution 1: Disable OSC0 by writing a zero to the Oscillator Enable bit in the System Control Interface (SCIF) Oscillator Control Register (SCIF.OSC0CTRL.OSCEN) before going to a sleep mode where OSC0 is disabled.
   Solution 2: Pull down or up XIN0 or XOUT0 with 1MOhm resistor.

10.2.15 I/O Pins

1. **PA17 has low ESD tolerance**
   PA17 only tolerates 500V ESD pulses (Human Body Model).
   Fix/Workaround
   Care must be taken during manufacturing and PCB design.

10.3 Rev. C

Not sampled.

10.4 Rev. B

10.4.1 Processor and Architecture

1. **RETS behaves incorrectly when MPU is enabled**
   RETS behaves incorrectly when MPU is enabled and MPU is configured so that system stack is not readable in unprivileged mode.
   Fix/Workaround
   Make system stack readable in unprivileged mode, or return from supervisor mode using rete instead of rets. This requires:
   1. Changing the mode bits from 001 to 110 before issuing the instruction. Updating the mode bits to the desired value must be done using a single mtsr instruction so it is done

2. **It is not possible to mask the request clock requests**
   It is not possible to mask the request clock requests using PPCR.
   Fix/Workaround
   None.

3. **Static mode cannot be entered if the WDT is using OSC32**
   If the WDT is using OSC32 as clock source and the user tries to enter Static mode, the Deepstop mode will be entered instead.
   Fix/Workaround
   None.

4. **Clock Failure Detector (CFD) does not work**
   Clock Failure Detector (CFD) does not work.
   Fix/Workaround
   None.

5. **WCAUSE register should not be used**
   The WCAUSE register should not be used.
   Fix/Workaround
   None.

6. **PB writes via debugger in sleep modes are blocked during sleepwalking**
   During sleepwalking, PB writes performed by a debugger will be discarded by all PB modules except the module that is requesting the clock.
   Fix/Workaround
   None.

7. **Clock sources will not be stopped in Static mode if the difference between CPU and PBx division factor is larger than 4**
   If the division factor between the CPU/HSB and PBx frequencies is more than 4 when entering a sleep mode where the system RC oscillator (RCSYS) is turned off, the high speed clock sources will not be turned off. This will result in a significantly higher power consumption during the sleep mode.
   Fix/Workaround
   Before going to sleep modes where RCSYS is stopped, make sure the division factor between CPU/HSB and PBx frequencies is less than or equal to 4.

8. **Disabling POR33 may generate spurious resets**
   Depending on operating conditions, POR33 may generate a spurious reset in one of the following cases:
   - When POR33 is disabled from the user interface
   - When SM33 supply monitor is enabled
   - When entering Shutdown mode while debugging the chip using JTAG or aWire interface

   In the listed cases, writing a one to the bit VREGCR.POR33MASK in the System Control Interface (SCIF) to mask the POR33 reset will be ineffective
   Fix/Workaround
   - Do not disable POR33 using the user interface
   - Do not use the SM33 supply monitor
   - Do not enter Shutdown mode if a debugger is connected to the chip

9. **Instability when exiting sleep walking**
   If all the following operating conditions are true, exiting sleep walking might lead to instability:

Fix/Workaround
If the target frequency is below 30MHz, use a max step size (DFLL0MAXSTEP.MAXSTEP) of seven or lower.

7. **Generic clock sources are kept running in sleep modes**
If a clock is used as a source for a generic clock when going to a sleep mode where clock sources are stopped, the source of the generic clock will be kept running. Please refer to Power Manager chapter for details about sleep modes.
Fix/Workaround
Disable generic clocks before going to sleep modes where clock sources are stopped to save power.

8. **DFLL clock is unstable with a fast reference clock**
The DFLL clock can be unstable when a fast clock is used as a reference clock in closed loop mode.
Fix/Workaround
Use the 32KHz crystal oscillator clock, or a clock with a similar frequency, as DFLLIF reference clock.

9. **DFLLIF indicates coarse lock too early**
The DFLLIF might indicate coarse lock too early, the DFLL will lose coarse lock and regain it later.
Fix/Workaround
Use max step size (DFLL0MAXSTEP.MAXSTEP) of 4 or higher.

10. **DFLLIF dithering does not work**
The DFLLIF dithering does not work.
Fix/Workaround
None.

11. **DFLLIF might lose fine lock when dithering is disabled**
When dithering is disabled and fine lock has been acquired, the DFLL might lose the fine lock resulting in up to 20% over-/undershoot.
Fix/Workaround
Solution 1: When the DFLL is used as main clock source, the target frequency of the DFLL should be 20% below the maximum operating frequency of the CPU. Don't use the DFLL as clock source for frequency sensitive applications.
Solution 2: Do not use the DFLL in closed loop mode.

12. **GCLK5 is non-functional**
GCLK5 is non-functional.
Fix/Workaround
None.

13. **BRIFA is non-functional**
BRIFA is non-functional.
Fix/Workaround
None.

14. **SCIF VERSION register reads 0x100**
SCIFVERSION register reads 0x100 instead of 0x102.
Fix/Workaround
None.

11. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

11.1 Rev. I - 01/2012

1. Overview - Block diagram: CAT SMP corrected from I/O to output. SPI NPCS corrected from output to I/O.
2. Package and Pinout: PRND signal removed from Signal Descriptions List table and GPIO Controller Function Multiplexing table
3. ADCIFB: PRND signal removed from block diagram.
4. Electrical Characteristics: Added pin input capacitance C_{IN} for TLLGA package for “all normal I/O pins except PA05 , PA07, PA17, PA20, PA21, PB04, PB05”.
5. Errata: Added more errata for TWI and CAT modules.

11.2 Rev. H - 12/2011

1. Mechanical Characteristics: Updated the note related to the QFN48 Package Drawing.Updated thermal reistance data for TLLGA48 package. Updated package drawings for TQFP48 and QFN48 packages. Soldering profile updated (Time maintained above 217°C.)
2. Memories: Local bus address map corrected: The address offset for port 1 registers is 0x100, not 0x200.
3. SCIF DFLL: Removed “not” from “DFLLnSTEP.CSTEP and DFLLnSTEP.FSTEP should not be lower than 50% of the maximum value of DFLLnCONF.COARSE and DFLLnCONF.FINE”.
4. SCIF VREG POR descriptions updated. POR33 bits added in VREGCR.
5. SCIF VREG: Removed reference to flash fuses for CALIB field, user is recommended not writing to SELVDD field. Removed references to Electrical Characteristics.
6. SCIF: Fuses text removed from some submodules (SM33, VREG).
7. SCIF VREG: Flash recalibration is always done at POR.
8. SCIF SM33: Enabling SM33 will disable the POR33 detector.
9, Erratum regarding OSC32 disabling is not valid for RevB.
10. Flash Controller: Serial number address updated.
11. Block diagram and ADCIFB: Removed PRND signal from block diagram and ADCIFB block diagram.
12. USART: Added CTSIC bit description.
13. Power Manager: Updated Clock division and Clock ready flag sections.
14. ADCIFB: Added DMA section in Product Dependencies.