

Welcome to [E-XFL.COM](https://www.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	60MHz
Connectivity	I ² C, IrDA, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3b1256-z1ur

1. Description

The AT32UC3B is a complete System-On-Chip microcontroller based on the AVR32 UC RISC processor running at frequencies up to 60 MHz. AVR32 UC is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high performance.

The processor implements a Memory Protection Unit (MPU) and a fast and flexible interrupt controller for supporting modern operating systems and real-time operating systems.

Higher computation capability is achieved using a rich set of DSP instructions.

The AT32UC3B incorporates on-chip Flash and SRAM memories for secure and fast access.

The Peripheral Direct Memory Access controller enables data transfers between peripherals and memories without processor involvement. PDCA drastically reduces processing overhead when transferring continuous and large data streams between modules within the MCU.

The Power Manager improves design flexibility and security: the on-chip Brown-Out Detector monitors the power supply, the CPU runs from the on-chip RC oscillator or from one of external oscillator sources, a Real-Time Clock and its associated timer keeps track of the time.

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

The PWM modules provides seven independent channels with many configuration options including polarity, edge alignment and waveform non overlap control. One PWM channel can trigger ADC conversions for more accurate close loop control implementations.

The AT32UC3B also features many communication interfaces for communication intensive applications. In addition to standard serial interfaces like USART, SPI or TWI, other interfaces like flexible Synchronous Serial Controller and USB are available. The USART supports different communication modes, like SPI mode.

The Synchronous Serial Controller provides easy access to serial communication protocols and audio standards like I²S, UART or SPI.

The Full-Speed USB 2.0 Device interface supports several USB Classes at the same time thanks to the rich End-Point configuration. The Embedded Host interface allows device like a USB Flash disk or a USB printer to be directly connected to the processor.

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 included fully debounced reporting of touch keys and includes 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.

AT32UC3B integrates a class 2+ Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control. The Nanotrace interface enables trace feature for JTAG-based debuggers.

4. Package and Pinout

4.1 Package

The device pins are multiplexed with peripheral functions as described in the Peripheral Multiplexing on I/O Line section.

Figure 4-1. TQFP64 / QFN64 Pinout

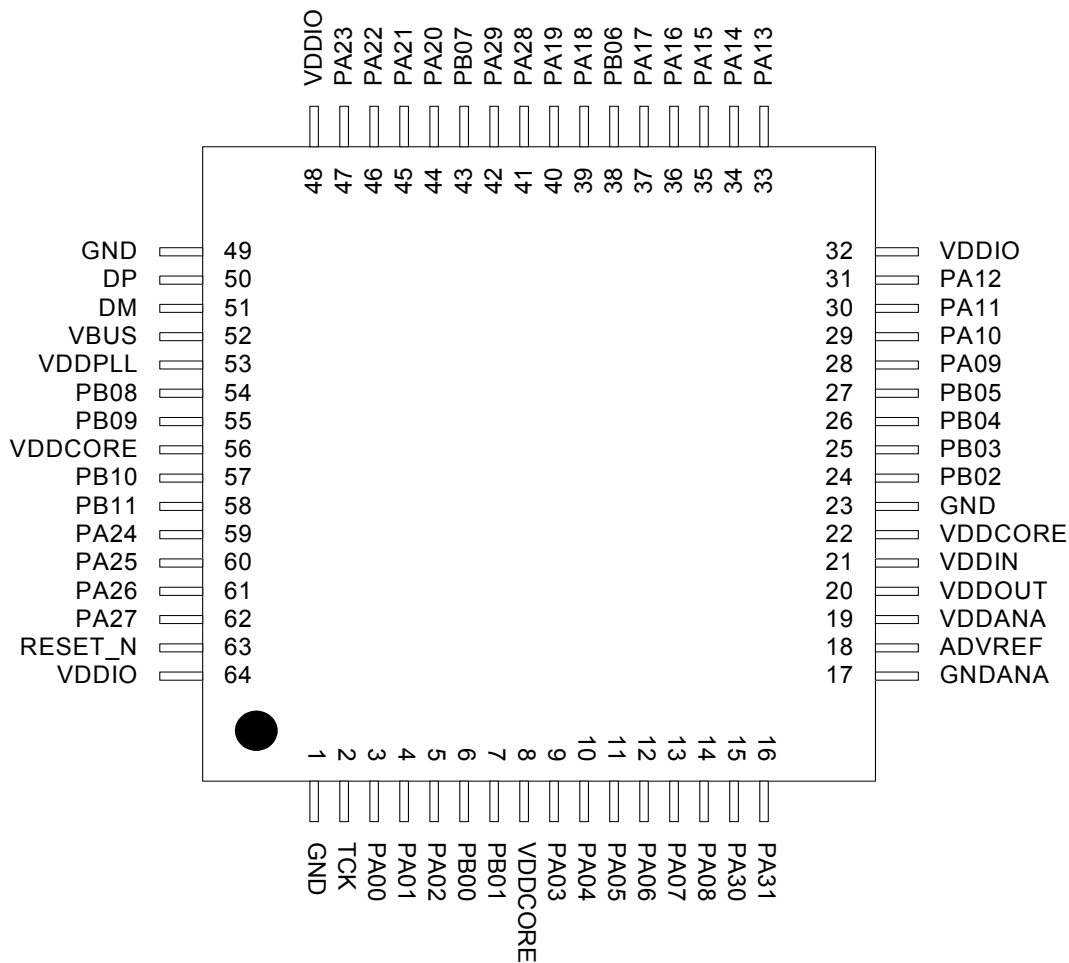
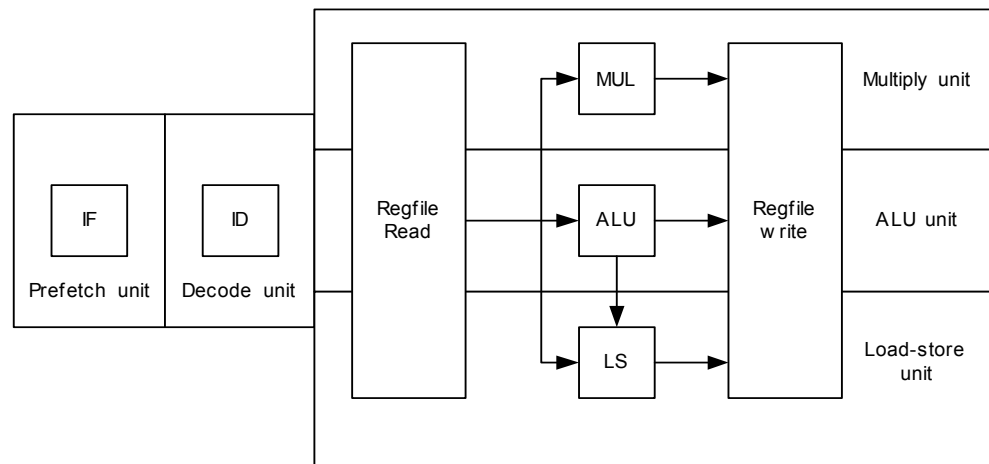


Figure 6-2. The AVR32UC Pipeline

6.3.2 AVR32A Microarchitecture Compliance

AVR32UC implements an AVR32A microarchitecture. The AVR32A microarchitecture is targeted at cost-sensitive, lower-end applications like smaller microcontrollers. This microarchitecture does not provide dedicated hardware registers for shadowing of register file registers in interrupt contexts. Additionally, it does not provide hardware registers for the return address registers and return status registers. Instead, all this information is stored on the system stack. This saves chip area at the expense of slower interrupt handling.

Upon interrupt initiation, registers R8-R12 are automatically pushed to the system stack. These registers are pushed regardless of the priority level of the pending interrupt. The return address and status register are also automatically pushed to stack. The interrupt handler can therefore use R8-R12 freely. Upon interrupt completion, the old R8-R12 registers and status register are restored, and execution continues at the return address stored popped from stack.

The stack is also used to store the status register and return address for exceptions and *scall*. Executing the *rete* or *rets* instruction at the completion of an exception or system call will pop this status register and continue execution at the popped return address.

6.3.3 Java Support

AVR32UC does not provide Java hardware acceleration.

6.3.4 Memory Protection

The MPU allows the user to check all memory accesses for privilege violations. If an access is attempted to an illegal memory address, the access is aborted and an exception is taken. The MPU in AVR32UC is specified in the AVR32UC Technical Reference manual.

6.3.5 Unaligned Reference Handling

AVR32UC does not support unaligned accesses, except for doubleword accesses. AVR32UC is able to perform word-aligned *st.d* and *ld.d*. Any other unaligned memory access will cause an address exception. Doubleword-sized accesses with word-aligned pointers will automatically be performed as two word-sized accesses.

The following table shows the instructions with support for unaligned addresses. All other instructions require aligned addresses.

Table 6-1. Instructions with Unaligned Reference Support

Instruction	Supported alignment
ld.d	Word
st.d	Word

6.3.6 Unimplemented Instructions

The following instructions are unimplemented in AVR32UC, and will cause an Unimplemented Instruction Exception if executed:

- All SIMD instructions
- All coprocessor instructions if no coprocessors are present
- retj, incjosp, popjc, pushjc
- tlbr, tlbs, tlbw
- cache

6.3.7 CPU and Architecture Revision

Three major revisions of the AVR32UC CPU currently exist.

The Architecture Revision field in the CONFIG0 system register identifies which architecture revision is implemented in a specific device.

AVR32UC CPU revision 3 is fully backward-compatible with revisions 1 and 2, ie. code compiled for revision 1 or 2 is binary-compatible with revision 3 CPUs.

All interrupt levels are by default disabled when debug state is entered, but they can individually be switched on by the monitor routine by clearing the respective mask bit in the status register.

Debug state can be entered as described in the *AVR32UC Technical Reference Manual*.

Debug state is exited by the *retd* instruction.

6.4.4 System Registers

The system registers are placed outside of the virtual memory space, and are only accessible using the privileged *mfsr* and *mtsr* instructions. The table below lists the system registers specified in the AVR32 architecture, some of which are unused in AVR32UC. The programmer is responsible for maintaining correct sequencing of any instructions following a *mtsr* instruction. For detail on the system registers, refer to the *AVR32UC Technical Reference Manual*.

Table 6-3. System Registers

Reg #	Address	Name	Function
0	0	SR	Status Register
1	4	EVBA	Exception Vector Base Address
2	8	ACBA	Application Call Base Address
3	12	CPUCR	CPU Control Register
4	16	ECR	Exception Cause Register
5	20	RSR_SUP	Unused in AVR32UC
6	24	RSR_INT0	Unused in AVR32UC
7	28	RSR_INT1	Unused in AVR32UC
8	32	RSR_INT2	Unused in AVR32UC
9	36	RSR_INT3	Unused in AVR32UC
10	40	RSR_EX	Unused in AVR32UC
11	44	RSR_NMI	Unused in AVR32UC
12	48	RSR_DBG	Return Status Register for Debug mode
13	52	RAR_SUP	Unused in AVR32UC
14	56	RAR_INT0	Unused in AVR32UC
15	60	RAR_INT1	Unused in AVR32UC
16	64	RAR_INT2	Unused in AVR32UC
17	68	RAR_INT3	Unused in AVR32UC
18	72	RAR_EX	Unused in AVR32UC
19	76	RAR_NMI	Unused in AVR32UC
20	80	RAR_DBG	Return Address Register for Debug mode
21	84	JECCR	Unused in AVR32UC
22	88	JOSP	Unused in AVR32UC
23	92	JAVA_LV0	Unused in AVR32UC
24	96	JAVA_LV1	Unused in AVR32UC
25	100	JAVA_LV2	Unused in AVR32UC

6.6 Module Configuration

All AT32UC3B parts do not implement the same CPU and Architecture Revision.

Table 6-5. CPU and Architecture Revision

Part Name	Architecture Revision
AT32UC3Bx512	2
AT32UC3Bx256	1
AT32UC3Bx128	1
AT32UC3Bx64	1

9.2 DC Characteristics

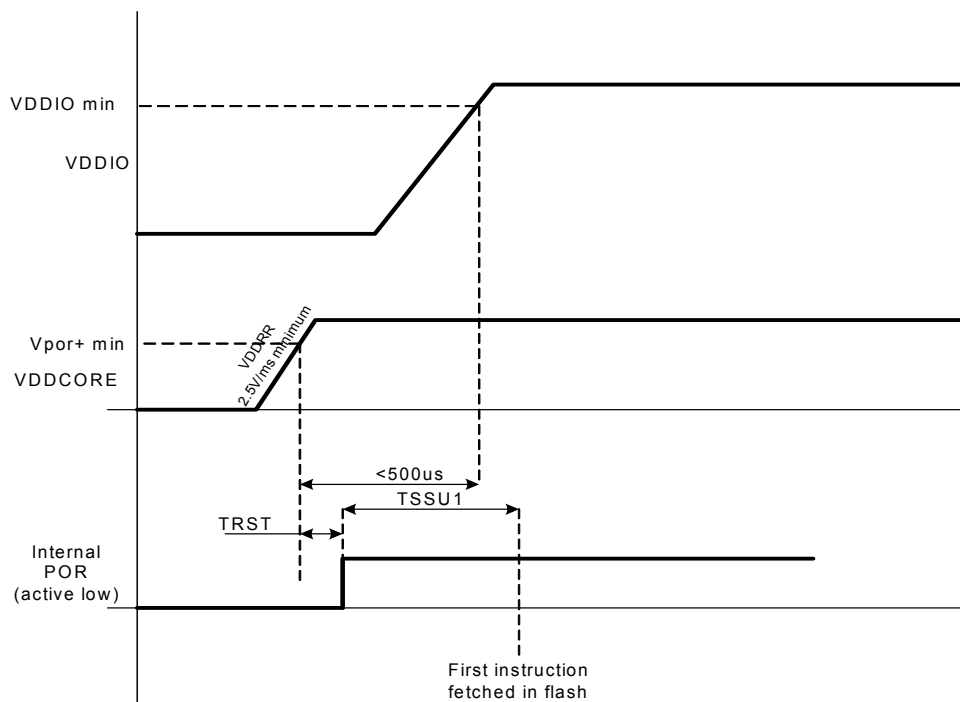
The following characteristics are applicable to the operating temperature range: $T_A = -40^{\circ}\text{C}$ to 85°C , unless otherwise specified and are certified for a junction temperature up to $T_J = 100^{\circ}\text{C}$.

Table 9-1. DC Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{VDDCORE}	DC Supply Core		1.65		1.95	V
V_{VDDPLL}	DC Supply PLL		1.65		1.95	V
V_{VDDIO}	DC Supply Peripheral I/Os		3.0		3.6	V
V_{IL}	Input Low-level Voltage		-0.3		+0.8	V
V_{IH}	Input High-level Voltage	AT32UC3B064 AT32UC3B0128 AT32UC3B0256 AT32UC3B164 AT32UC3B1128 AT32UC3B1256	All I/O pins except TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31	2.0	5.5	V
			TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31	2.0	3.6	V
		AT32UC3B0512 AT32UC3B1512	All I/O pins except TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31	2.0	5.5	V
			TCK, RESET_N	2.5	3.6	V
			PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31	2.0	3.6	V
V_{OL}	Output Low-level Voltage	$I_{\text{OL}} = -4\text{mA}$ for all I/O except PA20, PA21, PA22, PA23			0.4	V
		$I_{\text{OL}} = -8\text{mA}$ for PA20, PA21, PA22, PA23			0.4	V
V_{OH}	Output High-level Voltage	$I_{\text{OL}} = -4\text{mA}$ for all I/O except PA20, PA21, PA22, PA23	$V_{\text{VDDIO}} - 0.4$			V
		$I_{\text{OL}} = -8\text{mA}$ for PA20, PA21, PA22, PA23	$V_{\text{VDDIO}} - 0.4$			V
I_{OL}	Output Low-level Current	All I/O pins except PA20, PA21, PA22, PA23			-4	mA
		PA20, PA21, PA22, PA23			-8	mA
I_{OH}	Output High-level Current	All I/O pins except for PA20, PA21, PA22, PA23			4	mA
		PA20, PA21, PA22, PA23			8	mA
I_{LEAK}	Input Leakage Current	Pullup resistors disabled			1	μA

Therefore VDDCORE rise rate (VDDRR) must be equal or superior to 2.5V/ms and VDDIO must reach VDDIO min value before 500 us ($< TRST + TSSU1$) after VDDCORE has reached V_{POR+} min value.

Figure 9-4. Dual Supply Configuration



9.4.4 RESET_N Characteristics

Table 9-9. RESET_N Waveform Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{RESET}	RESET_N minimum pulse width		10			ns

9.7 Oscillator Characteristics

The following characteristics are applicable to the operating temperature range: $T_A = -40^{\circ}\text{C}$ to 85°C and worst case of power supply, unless otherwise specified.

9.7.1 Slow Clock RC Oscillator

Table 9-16. RC Oscillator Frequency

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
F_{RC}	RC Oscillator Frequency	Calibration point: $T_A = 85^{\circ}\text{C}$		115.2	116	KHz
		$T_A = 25^{\circ}\text{C}$		112		KHz
		$T_A = -40^{\circ}\text{C}$	105	108		KHz

9.7.2 32 KHz Oscillator

Table 9-17. 32 KHz Oscillator Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$1/(t_{CP32KHz})$	Oscillator Frequency	External clock on XIN32			30	MHz
		Crystal		32 768		Hz
C_L	Equivalent Load Capacitance		6		12.5	pF
ESR	Crystal Equivalent Series Resistance				100	K Ω
t_{ST}	Startup Time	$C_L = 6\text{pF}^{(1)}$ $C_L = 12.5\text{pF}^{(1)}$			600 1200	ms
t_{CH}	XIN32 Clock High Half-period		$0.4 t_{CP}$		$0.6 t_{CP}$	
t_{CL}	XIN32 Clock Low Half-period		$0.4 t_{CP}$		$0.6 t_{CP}$	
C_{IN}	XIN32 Input Capacitance				5	pF
I_{OSC}	Current Consumption	Active mode			1.8	μA
		Standby mode			0.1	μA

Note: 1. C_L is the equivalent load capacitance.

9.10 JTAG Characteristics

9.10.1 JTAG Timing

Figure 9-6. JTAG Interface Signals

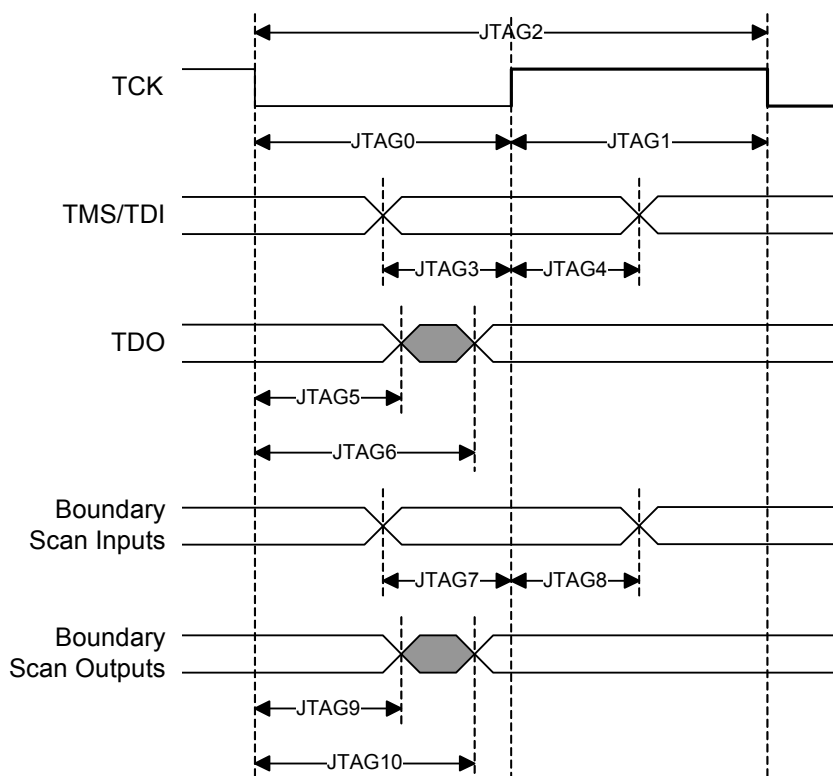


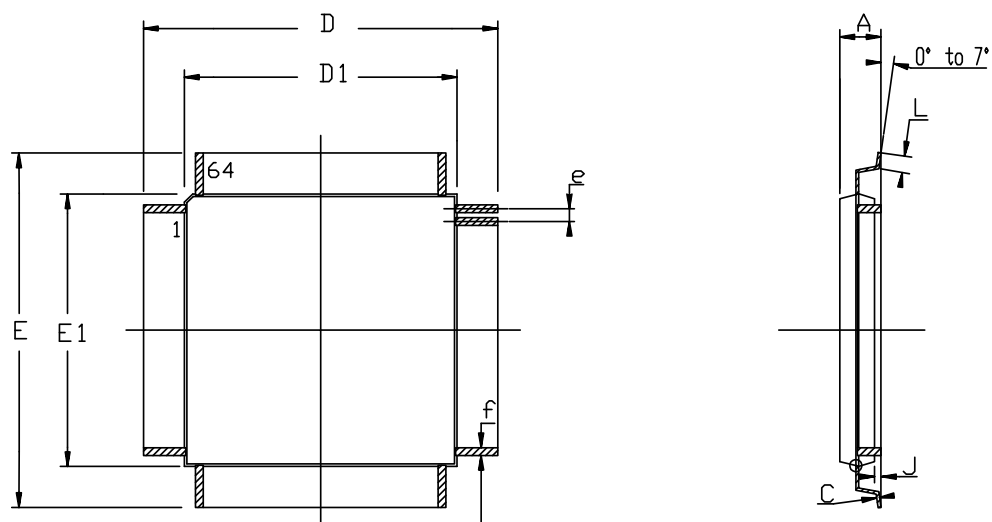
Table 9-26. JTAG Timings⁽¹⁾

Symbol	Parameter	Conditions	Min	Max	Units
JTAG0	TCK Low Half-period	V_{DDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF	23.2		ns
JTAG1	TCK High Half-period		8.8		ns
JTAG2	TCK Period		32.0		ns
JTAG3	TDI, TMS Setup before TCK High		3.9		ns
JTAG4	TDI, TMS Hold after TCK High		0.6		ns
JTAG5	TDO Hold Time		4.5		ns
JTAG6	TCK Low to TDO Valid			23.2	ns
JTAG7	Boundary Scan Inputs Setup Time		0		ns
JTAG8	Boundary Scan Inputs Hold Time		5.0		ns
JTAG9	Boundary Scan Outputs Hold Time		8.7		ns
JTAG10	TCK to Boundary Scan Outputs Valid			17.7	ns

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

10.2 Package Drawings

Figure 10-1. TQFP-64 package drawing



COMMON DIMENSIONS IN MM

SYMBOL	Min	Max	NOTES
A	----	1.20	
A1	0.95	1.05	
C	0.09	0.20	
D	12.00 BSC		
D1	10.00 BSC		
E	12.00 BSC		
E1	10.00 BSC		
J	0.05	0.15	
L	0.45	0.75	
e	0.50 BSC		
f	0.17	0.27	

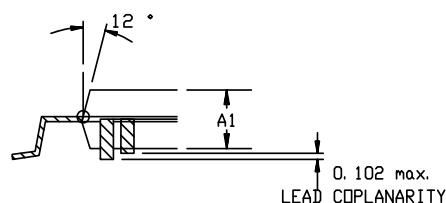


Table 10-2. Device and Package Maximum Weight

Weight	300 mg
--------	--------

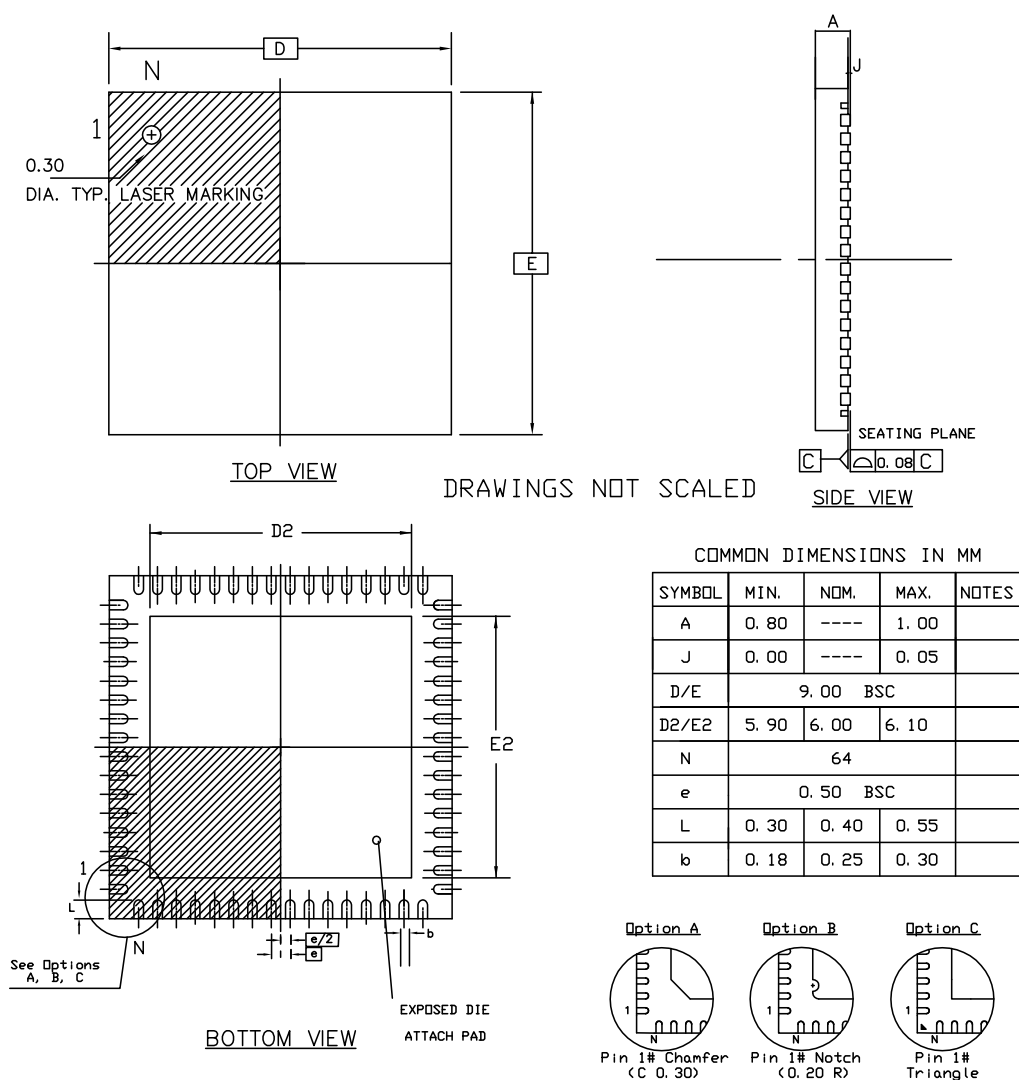
Table 10-3. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3
----------------------------	----------------------

Table 10-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	e3

Figure 10-3. QFN-64 package drawing



Compliant JEDEC Standard MO-220 variation VMMD-3

Table 10-8. Device and Package Maximum Weight

Weight	200 mg
--------	--------

Table 10-9. Package Characteristics

Moisture Sensitivity Level	Jedec J-STD-20D-MSL3
----------------------------	----------------------

Table 10-10. Package Reference

JEDEC Drawing Reference	M0-220
JESD97 Classification	e3

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 the system RC oscillator is stopped, make sure that the factor between the CPU/HSB and PBx frequencies is less than or equal to 4.

15. Increased Power Consumption in VDDIO in sleep modes

If the 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

Disable the OSC0 through the Power Manager (PM) before going to any sleep mode where the OSC0 is disabled, or pull down or up XIN0 and XOUT0 with 1Mohm resistor.

16. SSC

17. Additional delay on TD output

A delay from 2 to 3 system clock cycles is added to TD output when:

TCMR.START = Receive Start,

TCMR.STTDLY = more than ZERO,

RCMR.START = Start on falling edge / Start on Rising edge / Start on any edge,

RFMR.FSOS = None (input).

Fix/Workaround

None.

18. TF output is not correct

TF output is not correct (at least emitted one serial clock cycle later than expected) when:

TFMR.FSOS = Driven Low during data transfer/ Driven High during data transfer

TCMR.START = Receive start

RFMR.FSOS = None (Input)

RCMR.START = any on RF (edge/level)

Fix/Workaround

None.

19. Frame Synchro and Frame Synchro Data are delayed by one clock cycle

The frame synchro and the frame synchro data are delayed from 1 SSC_CLOCK when:

- Clock is CKDIV

- The START is selected on either a frame synchro edge or a level

- Frame synchro data is enabled

- Transmit clock is gated on output (through CKO field)

Fix/Workaround

Transmit or receive CLOCK must not be gated (by the mean of CKO field) when START condition is performed on a generated frame synchro.

20. USB

21. UPCFGn.INTFRQ is irrelevant for isochronous pipe

As a consequence, isochronous IN and OUT tokens are sent every 1 ms (Full Speed), or every 125uS (High Speed).

Fix/Workaround

For higher polling time, the software must freeze the pipe for the desired period in order to prevent any "extra" token.

- ADC

1. Sleep Mode activation needs additional A to D conversion

If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.

Fix/Workaround

Activate the sleep mode in the mode register and then perform an AD conversion.

- PDCA

1. Wrong PDCA behavior when using two PDCA channels with the same PID

Wrong PDCA behavior when using two PDCA channels with the same PID.

Fix/Workaround

The same PID should not be assigned to more than one channel.

2. Transfer error will stall a transmit peripheral handshake interface

If a transfer error is encountered on a channel transmitting to a peripheral, the peripheral handshake of the active channel will stall and the PDCA will not do any more transfers on the affected peripheral handshake interface.

Fix/Workaround

Disable and then enable the peripheral after the transfer error.

3. TWI

4. The TWI RXRDY flag in SR register is not reset when a software reset is performed

The TWI RXRDY flag in SR register is not reset when a software reset is performed.

Fix/Workaround

After a Software Reset, the register TWI RHR must be read.

5. TWI in master mode will continue to read data

TWI in master mode will continue to read data on the line even if the shift register and the RHR register are full. This will generate an overrun error.

Fix/Workaround

To prevent this, read the RHR register as soon as a new RX data is ready.

6. TWI slave behaves improperly if master acknowledges the last transmitted data byte before a STOP condition

In I2C slave transmitter mode, if the master acknowledges the last data byte before a STOP condition (what the master is not supposed to do), the following TWI slave receiver mode frame may contain an inappropriate clock stretch. This clock stretch can only be stopped by resetting the TWI.

Fix/Workaround

If the TWI is used as a slave transmitter with a master that acknowledges the last data byte before a STOP condition, it is necessary to reset the TWI before entering slave receiver mode.

12.1.2 Rev C

- PWM

1. **PWM channel interrupt enabling triggers an interrupt**
When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.
Fix/Workaround
When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.
2. **PWN counter restarts at 0x0001**
The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
Fix/Workaround
 - The first period is 0x0000, 0x0001, ..., period.
 - Consecutive periods are 0x0001, 0x0002, ..., period.
3. **PWM update period to a 0 value does not work**
It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).
Fix/Workaround
Do not update the PWM_CUPD register with a value equal to 0.
4. **SPI**
5. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.
Fix/Workaround
For PDCA transfer: none.
6. **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.
7. **SPI Glitch on RXREADY flag in slave mode when enabling the SPI or during the first transfer**
In slave mode, the SPI can generate a false RXREADY signal during enabling of the SPI or during the first transfer.
Fix/Workaround
 1. Set slave mode, set required CPOL/CPHA.
 2. Enable SPI.
 3. Set the polarity CPOL of the line in the opposite value of the required one.
 4. Set the polarity CPOL to the required one.
 5. Read the RXHOLDING register.
Transfers can now begin and RXREADY will now behave as expected.

even if the frame is correct and the USART has been disabled, reset by a soft reset and re-enabled.

Fix/Workaround

None.

9. USART slave synchronous mode external clock must be at least 9 times lower in frequency than CLK_USART

When the USART is operating in slave synchronous mode with an external clock, the frequency of the signal provided on CLK must be at least 9 times lower than CLK_USART.

Fix/Workaround

When the USART is operating in slave synchronous mode with an external clock, provide a signal on CLK that has a frequency at least 9 times lower than CLK_USART.

10. HMATRIX

11. In the PRAS and PRBS registers, the MxPR fields are only two bits

In the PRAS and PRBS registers, the MxPR fields are only two bits wide, instead of four bits. The unused bits are undefined when reading the registers.

Fix/Workaround

Mask undefined bits when reading PRAS and PRBS.

- FLASHC

1. Reading from on-chip flash may fail after a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands).

After a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands), the following flash read access may return corrupted data. This erratum does not affect write operations to regular flash memory.

Fix/Workaround

The flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands) must be issued from internal RAM. After the write operation, perform a dummy flash page write operation (FLASHC WP). Content and location of this page is not important and filling the write buffer with all one (FFh) will leave the current flash content unchanged. It is then safe to read and fetch code from the flash.

- DSP Operations

1. Hardware breakpoints may corrupt MAC results

Hardware breakpoints on MAC instructions may corrupt the destination register of the MAC instruction.

Fix/Workaround

Place breakpoints on earlier or later instructions.

12.2.2 Rev. G

- PWM

1. **PWM channel interrupt enabling triggers an interrupt**
When enabling a PWM channel that is configured with center aligned period (CALG=1), an interrupt is signalled.
Fix/Workaround
When using center aligned mode, enable the channel and read the status before channel interrupt is enabled.
2. **PWN counter restarts at 0x0001**
The PWM counter restarts at 0x0001 and not 0x0000 as specified. Because of this the first PWM period has one more clock cycle.
Fix/Workaround
- The first period is 0x0000, 0x0001, ..., period.
- Consecutive periods are 0x0001, 0x0002, ..., period.
3. **PWM update period to a 0 value does not work**
It is impossible to update a period equal to 0 by the using the PWM update register (PWM_CUPD).
Fix/Workaround
Do not update the PWM_CUPD register with a value equal to 0.
4. **SPI**
5. **SPI Slave / PDCA transfer: no TX UNDERRUN flag**
There is no TX UNDERRUN flag available, therefore in SPI slave mode, there is no way to be informed of a character lost in transmission.
Fix/Workaround
For PDCA transfer: none.
6. **SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0**
When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.
Fix/Workaround
When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.
7. **SPI Glitch on RXREADY flag in slave mode when enabling the SPI or during the first transfer**
In slave mode, the SPI can generate a false RXREADY signal during enabling of the SPI or during the first transfer.
Fix/Workaround
 1. Set slave mode, set required CPOL/CPHA.
 2. Enable SPI.
 3. Set the polarity CPOL of the line in the opposite value of the required one.
 4. Set the polarity CPOL to the required one.
 5. Read the RXHOLDING register.
Transfers can now begin and RXREADY will now behave as expected.

8. SPI bad serial clock generation on 2nd chip_select when SCBR=1, CPOL=1, and NCPHA=0

When multiple chip selects (CS) are in use, if one of the baudrates equal 1 while one (CSRn.SCBR=1) of the others do not equal 1, and CSRn.CPOL=1 and CSRn.NCPHA=0, then an additional pulse will be generated on SCK.

Fix/Workaround

When multiple CS are in use, if one of the baudrates equals 1, the others must also equal 1 if CSRn.CPOL=1 and CSRn.NCPHA=0.

9. SPI Glitch on RXREADY flag in slave mode when enabling the SPI or during the first transfer

In slave mode, the SPI can generate a false RXREADY signal during enabling of the SPI or during the first transfer.

Fix/Workaround

1. Set slave mode, set required CPOL/CPHA.
2. Enable SPI.
3. Set the polarity CPOL of the line in the opposite value of the required one.
4. Set the polarity CPOL to the required one.
5. Read the RXHOLDING register.

Transfers can now begin and RXREADY will now behave as expected.

10. SPI CSNAAT bit 2 in register CSR0...CSR3 is not available

SPI CSNAAT bit 2 in register CSR0...CSR3 is not available.

Fix/Workaround

Do not use this bit.

11. SPI disable does not work in SLAVE mode

SPI disable does not work in SLAVE mode.

Fix/Workaround

Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).

- Power Manager

1. PLL Lock control does not work

PLL lock Control does not work.

Fix/Workaround

In PLL Control register, the bit 7 should be set in order to prevent unexpected behavior.

2. Wrong reset causes when BOD is activated

Setting the BOD enable fuse will cause the Reset Cause Register to list BOD reset as the reset source even though the part was reset by another source.

Fix/Workaround

Do not set the BOD enable fuse, but activate the BOD as soon as your program starts.

3. System Timer mask (Bit 16) of the PM CPUMASK register is not available

System Timer mask (Bit 16) of the PM CPUMASK register is not available.

Fix/Workaround

Do not use this bit.

- SSC

1. **SSC does not trigger RF when data is low**

The SSC cannot transmit or receive data when CKS = CKDIV and CKO = none, in TCMR or RCMR respectively.

Fix/Workaround

Set CKO to a value that is not "none" and bypass the output of the TK/RK pin with the GPIO.

- USB

1. **USB No end of host reset signaled upon disconnection**

In host mode, in case of an unexpected device disconnection whereas a usb reset is being sent by the usb controller, the UHCON.RESET bit may not be cleared by the hardware at the end of the reset.

Fix/Workaround

A software workaround consists in testing (by polling or interrupt) the disconnection (UHINT.DDISCI == 1) while waiting for the end of reset (UHCON.RESET == 0) to avoid being stuck.

2. **USBFSM and UHADDR1/2/3 registers are not available**

Do not use USBFSM register.

Fix/Workaround

Do not use USBFSM register and use HCON[6:0] field instead for all the pipes.

- Cycle counter

1. **CPU Cycle Counter does not reset the COUNT system register on COMPARE match.**

The device revision B does not reset the COUNT system register on COMPARE match. In this revision, the COUNT register is clocked by the CPU clock, so when the CPU clock stops, so does incrementing of COUNT.

Fix/Workaround

None.

- ADC

1. **ADC possible miss on DRDY when disabling a channel**

The ADC does not work properly when more than one channel is enabled.

Fix/Workaround

Do not use the ADC with more than one channel enabled at a time.

2. **ADC OVRE flag sometimes not reset on Status Register read**

The OVRE flag does not clear properly if read simultaneously to an end of conversion.

Fix/Workaround

None.

3. **Sleep Mode activation needs additional A to D conversion**

If the ADC sleep mode is activated when the ADC is idle the ADC will not enter sleep mode before after the next AD conversion.

Fix/Workaround

Activate the sleep mode in the mode register and then perform an AD conversion.

2. **The command Quick Page Read User Page(QPRUP) is not functional**
 The command Quick Page Read User Page(QPRUP) is not functional.
Fix/Workaround
 None.

3. **PAGEN Semantic Field for Program GP Fuse Byte is WriteData[7:0], ByteAddress[1:0] on revision B instead of WriteData[7:0], ByteAddress[2:0]**
 PAGEN Semantic Field for Program GP Fuse Byte is WriteData[7:0], ByteAddress[1:0] on revision B instead of WriteData[7:0], ByteAddress[2:0].
Fix/Workaround
 None.

4. **Reading from on-chip flash may fail after a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands).**
 After a flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands), the following flash read access may return corrupted data. This erratum does not affect write operations to regular flash memory.
Fix/Workaround
 The flash fuse write operation (FLASHC LP, UP, WGPB, EGPB, SSB, PGPFB, EAGPF commands) must be issued from internal RAM. After the write operation, perform a dummy flash page write operation (FLASHC WP). Content and location of this page is not important and filling the write buffer with all one (FFh) will leave the current flash content unchanged. It is then safe to read and fetch code from the flash.

- 5.

- RTC

1. **Writes to control (CTRL), top (TOP) and value (VAL) in the RTC are discarded if the RTC peripheral bus clock (PBA) is divided by a factor of four or more relative to the HSB clock**
 Writes to control (CTRL), top (TOP) and value (VAL) in the RTC are discarded if the RTC peripheral bus clock (PBA) is divided by a factor of four or more relative to the HSB clock.
Fix/Workaround
 Do not write to the RTC registers using the peripheral bus clock (PBA) divided by a factor of four or more relative to the HSB clock.

2. **The RTC CLKEN bit (bit number 16) of CTRL register is not available**
 The RTC CLKEN bit (bit number 16) of CTRL register is not available.
Fix/Workaround
 Do not use the CLKEN bit of the RTC on Rev B.