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	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at32uc3l0256-zaur

1. Description

The Atmel® AVR® AT32UC3L0128/256 is a complete system-on-chip microcontroller based on the AVR32 UC RISC processor running at frequencies up to 50MHz. 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 and real-time operating systems. The Secure Access Unit (SAU) is used together with the MPU to provide the required security and integrity.

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

The AT32UC3L0128/256 embeds state-of-the-art picoPower technology for ultra-low power consumption. Combined power control techniques are used to bring active current consumption down to 174µA/MHz, and leakage down to 220nA while still retaining a bank of backup registers. The device allows a wide range of trade-offs between functionality and power consumption, giving the user the ability to reach the lowest possible power consumption with the feature set required for the application.

The Peripheral Direct Memory Access (DMA) controller enables data transfers between peripherals and memories without processor involvement. The Peripheral DMA controller drastically reduces processing overhead when transferring continuous and large data streams.

The AT32UC3L0128/256 incorporates on-chip Flash and SRAM memories for secure and fast access. The FlashVault technology allows secure libraries to be programmed into the device. The secure libraries can be executed while the CPU is in Secure State, but not read by non-secure software in the device. The device can thus be shipped to end customers, who will be able to program their own code into the device to access the secure libraries, but without risk of compromising the proprietary secure code.

The External Interrupt Controller (EIC) allows pins to be configured as external interrupts. Each external interrupt has its own interrupt request and can be individually masked.

The Peripheral Event System allows peripherals to receive, react to, and send peripheral events without CPU intervention. Asynchronous interrupts allow advanced peripheral operation in low power sleep modes.

The Power Manager (PM) improves design flexibility and security. The Power Manager supports SleepWalking functionality, by which a module can be selectively activated based on peripheral events, even in sleep modes where the module clock is stopped. Power monitoring is supported by on-chip Power-on Reset (POR), Brown-out Detector (BOD), and Supply Monitor (SM). The device features several oscillators, such as Phase Locked Loop (PLL), Digital Frequency Locked Loop (DFLL), Oscillator 0 (OSC0), and system RC oscillator (RCSYS). Either of these oscillators can be used as source for the system clock. The DFLL is a programmable internal oscillator from 20 to 150MHz. It can be tuned to a high accuracy if an accurate reference clock is running, e.g. the 32KHz crystal oscillator.

The Watchdog Timer (WDT) will reset the device unless it is periodically serviced by the software. This allows the device to recover from a condition that has caused the system to be unstable.

The Asynchronous Timer (AST) combined with the 32KHz crystal oscillator supports powerful real-time clock capabilities, with a maximum timeout of up to 136 years. The AST can operate in counter mode or calendar mode.

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 individually, or in interlinked mode, with multiple channels changed at the same time.

The AT32UC3L0128/256 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. All touch sensors can be configured to operate autonomously without software interaction, allowing wakeup from sleep modes when activated.

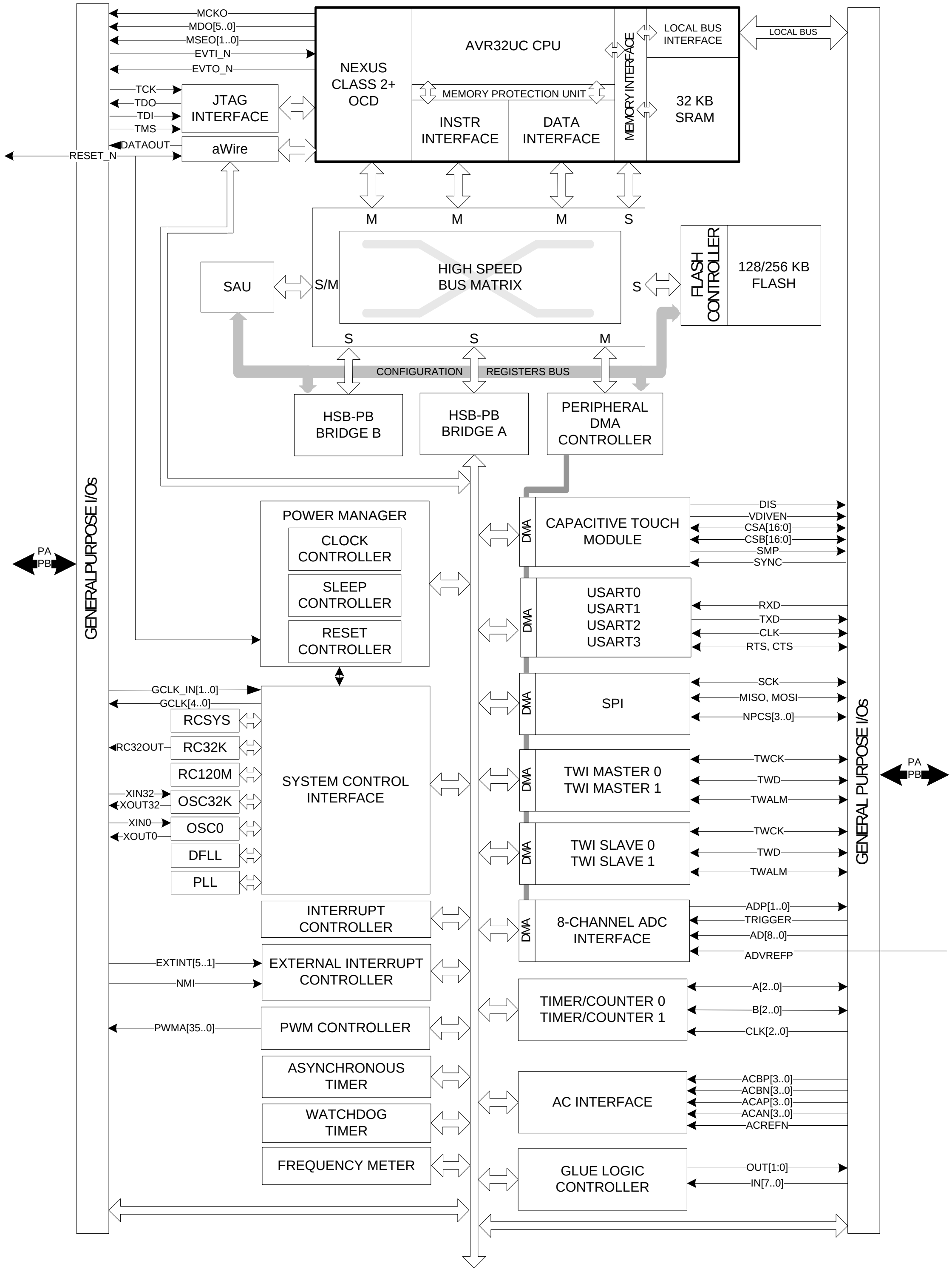
Atmel offers the QTouch library for embedding capacitive touch buttons, sliders, and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys as well as Adjacent Key Suppression® (AKS®) technology 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 AT32UC3L0128/256 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 runtime 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.

2. Overview

2.1 Block Diagram

Figure 2-1. Block Diagram



3. Package and Pinout

3.1 Package

The device pins are multiplexed with peripheral functions as described in Section 3.2.1.

Figure 3-1. TQFP48/QFN48 Pinout

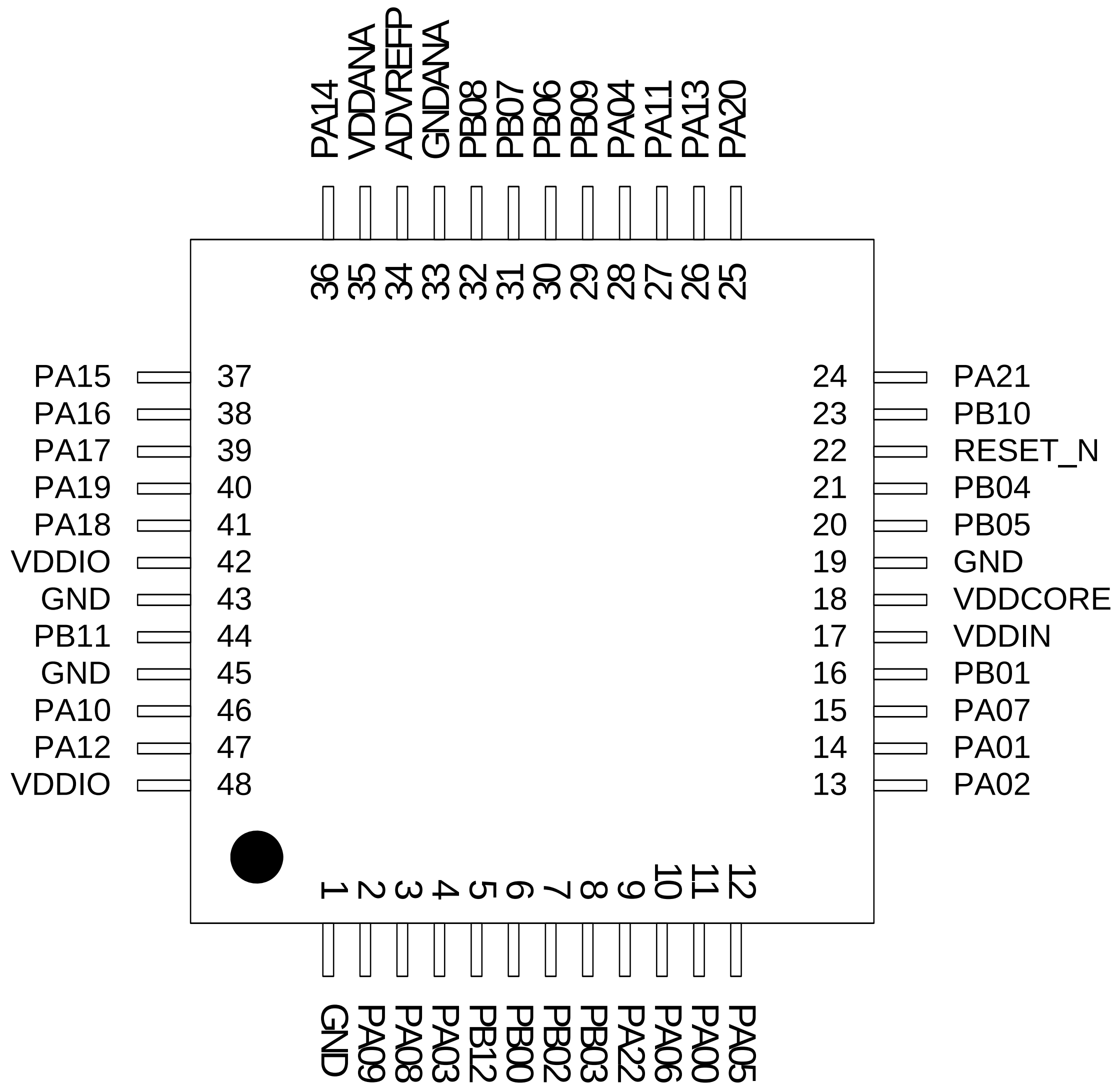
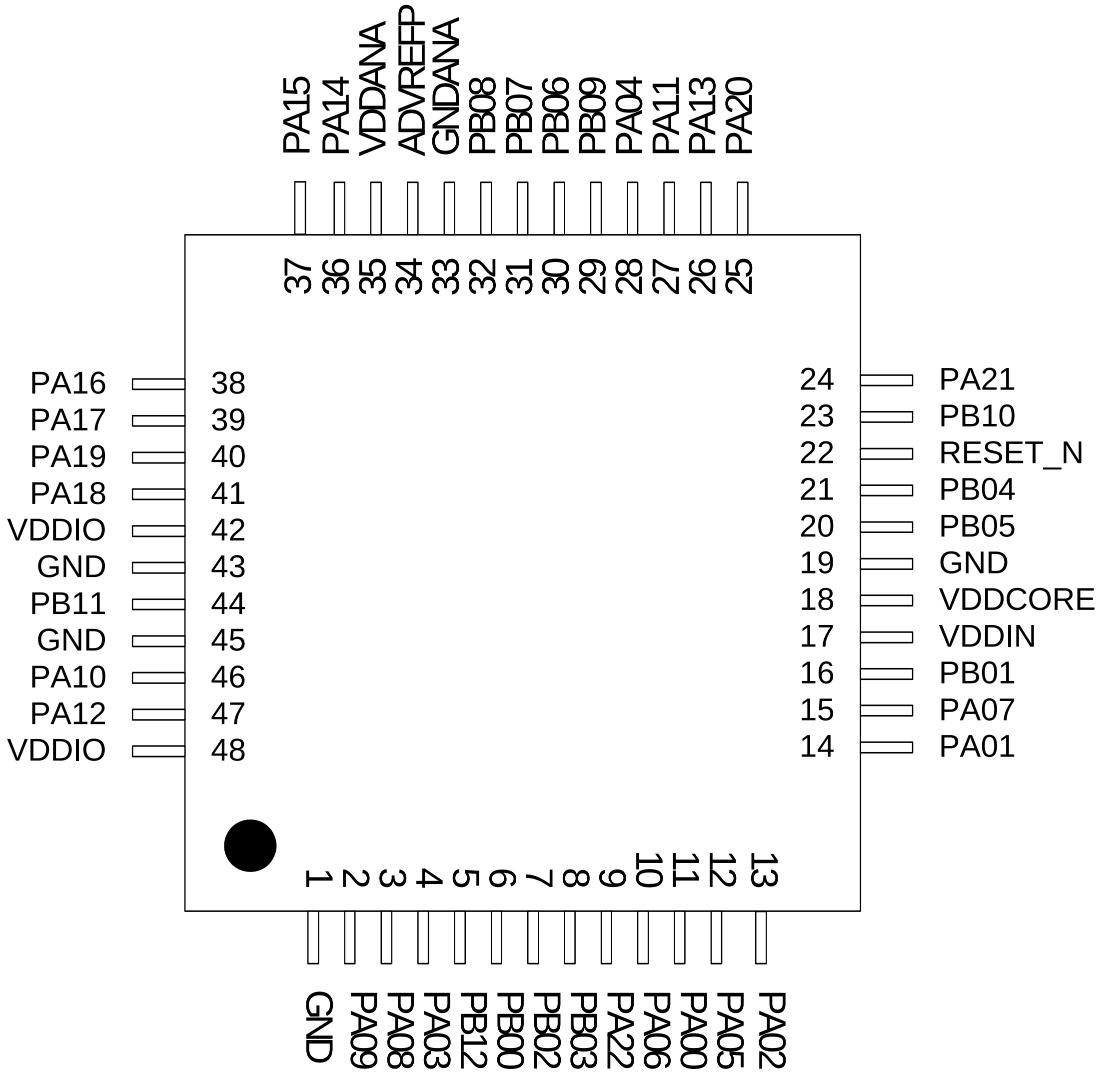


Figure 3-2. TLLGA48 Pinout



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]

Table 3-4. Nexus OCD AUX Port Connections

Pin	AXS=1	AXS=0
EVTO_N	PA04	PA04
MCKO	PA06	PB01
MSEO[1]	PA07	PB11
MSEO[0]	PA11	PB12

3.2.6 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 Name	Oscillator Pin
3	PA08	XIN0
46	PA10	XIN32
26	PA13	XIN32_2
2	PA09	XOUT0
47	PA12	XOUT32
25	PA20	XOUT32_2

3.2.7 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

3.3 Signal Descriptions

The following table gives details on signal names classified by peripheral.

Table 3-7. Signal Descriptions List

Signal Name	Function	Type	Active Level	Comments
Analog Comparator Interface - ACIFB				
ACAN3 - ACAN0	Negative inputs for comparators "A"	Analog		
ACAP3 - ACAP0	Positive inputs for comparators "A"	Analog		
ACBN3 - ACBN0	Negative inputs for comparators "B"	Analog		
ACBP3 - ACBP0	Positive inputs for comparators "B"	Analog		
ACREFN	Common negative reference	Analog		
ADC Interface - ADCIFB				
AD8 - AD0	Analog Signal	Analog		
ADP1 - ADP0	Drive Pin for resistive touch screen	Output		
TRIGGER	External trigger	Input		
aWire - AW				
DATA	aWire data	I/O		
DATAOUT	aWire data output for 2-pin mode	I/O		
Capacitive Touch Module - CAT				
CSA16 - CSA0	Capacitive Sense A	I/O		
CSB16 - CSB0	Capacitive Sense B	I/O		
DIS	Discharge current control	Analog		
SMP	SMP signal	Output		
SYNC	Synchronize signal	Input		
VDIVEN	Voltage divider enable	Output		
External Interrupt Controller - EIC				
NMI (EXTINT0)	Non-Maskable Interrupt	Input		
EXTINT5 - EXTINT1	External interrupt	Input		
Glue Logic Controller - GLOC				
IN7 - IN0	Inputs to lookup tables	Input		
OUT1 - OUT0	Outputs from lookup tables	Output		
JTAG module - JTAG				
TCK	Test Clock	Input		
TDI	Test Data In	Input		
TDO	Test Data Out	Output		
TMS	Test Mode Select	Input		

4. Mechanical Characteristics

4.1 Thermal Considerations

4.1.1 Thermal Data

Table 4-1 summarizes the thermal resistance data depending on the package.

Table 4-1. Thermal Resistance Data

Symbol	Parameter	Condition	Package	Typ	Unit
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TQFP48	54.4	°C/W
θ_{JC}	Junction-to-case thermal resistance		TQFP48	15.7	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	QFN48	26.0	°C/W
θ_{JC}	Junction-to-case thermal resistance		QFN48	1.6	
θ_{JA}	Junction-to-ambient thermal resistance	Still Air	TLLGA48	25.4	°C/W
θ_{JC}	Junction-to-case thermal resistance		TLLGA48	12.7	

4.1.2 Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

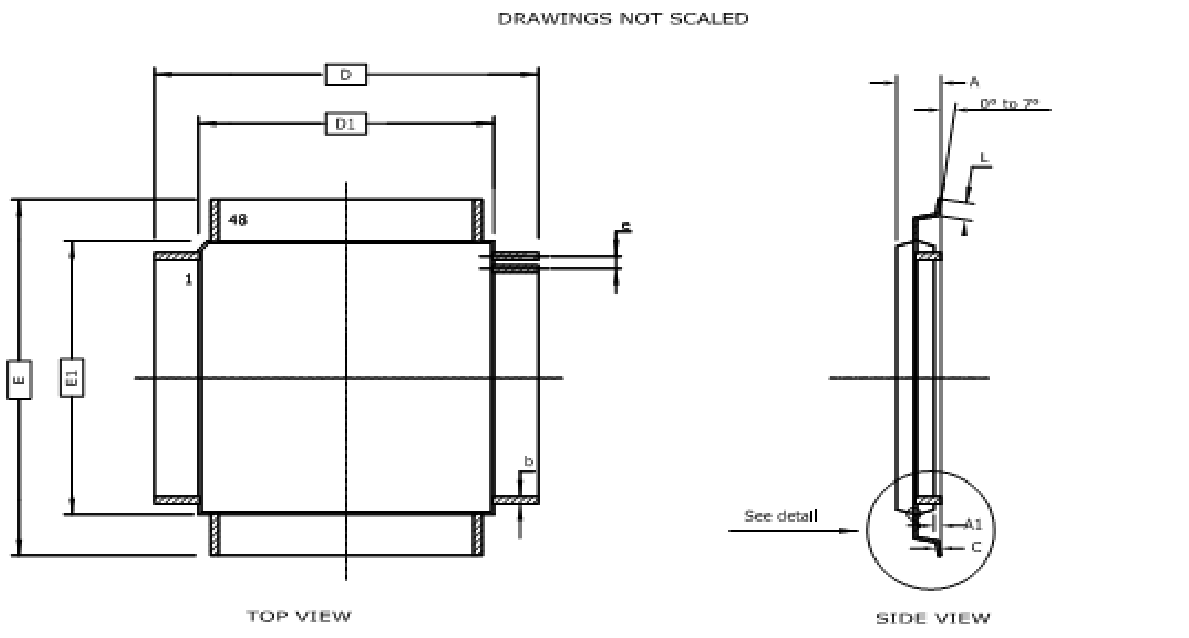
where:

- θ_{JA} = package thermal resistance, Junction-to-ambient (°C/W), provided in Table 4-1.
- θ_{JC} = package thermal resistance, Junction-to-case thermal resistance (°C/W), provided in Table 4-1.
- $\theta_{HEAT\ SINK}$ = cooling device thermal resistance (°C/W), provided in the device datasheet.
- P_D = device power consumption (W) estimated from data provided in Section 32.4 on page 792.
- T_A = ambient temperature (°C).

From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

4.2 Package Drawings

Figure 4-1. TQFP-48 Package Drawing



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.20	
A1	0.05	—	0.15	
A2	0.95	—	1.05	
C	0.09	—	0.20	
D/E	9.00 BSC			
D1/E1	7.00 BSC			
L	0.45	—	0.75	
b	0.17	—	0.27	
e	0.50 BSC			

Notes :
1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

Table 4-2. Device and Package Maximum Weight

140	mg

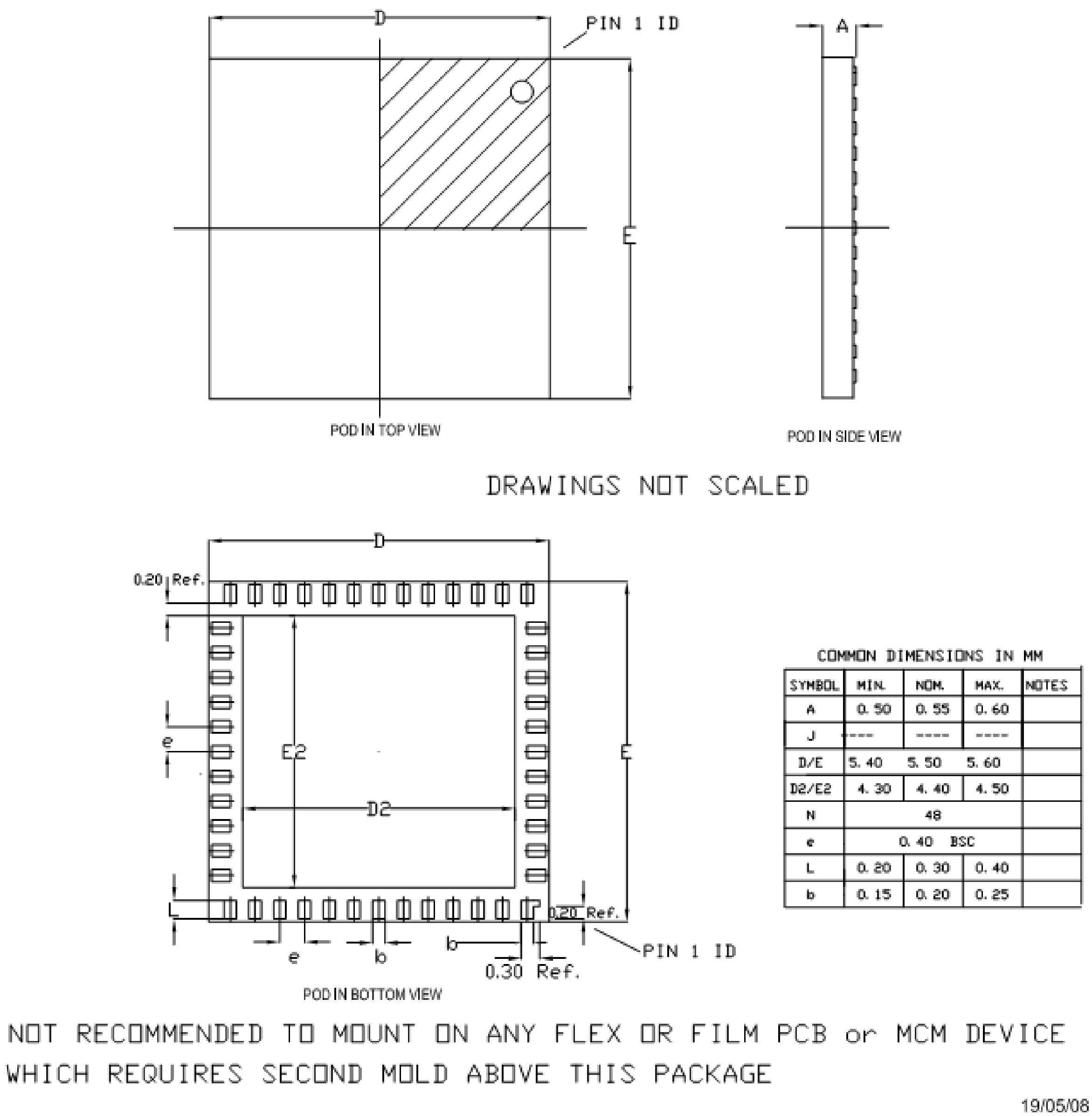
Table 4-3. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 4-4. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

Figure 4-3. TLLGA-48 Package Drawing



COMMON DIMENSIONS IN MM

SYMBOL	MIN.	NOM.	MAX.	NOTES
A	0.50	0.55	0.60	
J	----	----	----	
D/E	5.40	5.50	5.60	
D2/E2	4.30	4.40	4.50	
N	48			
e	0.40 BSC			
L	0.20	0.30	0.40	
b	0.15	0.20	0.25	

Table 4-8. Device and Package Maximum Weight

39.3	mg

Table 4-9. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 4-10. Package Reference

JEDEC Drawing Reference	N/A
JESD97 Classification	E4

5. Ordering Information

Table 5-1. Ordering Information

Device	Ordering Code	Carrier Type	Package	Package Type	Temperature Operating Range
AT32UC3L0256	AT32UC3L0256-AUTES	ES	TQFP 48	JESD97 Classification E3	Industrial (-40°C to 85°C)
	AT32UC3L0256-AUT	Tray			
	AT32UC3L0256-AUR	Tape & Reel			
	AT32UC3L0256-ZAUTES	ES	QFN 48		
	AT32UC3L0256-ZAUT	Tray			
	AT32UC3L0256-ZAUR	Tape & Reel			
	AT32UC3L0256-D3HES	ES	TLLGA 48	JESD97 Classification E4	
	AT32UC3L0256-D3HT	Tray			
	AT32UC3L0256-D3HR	Tape & Reel			
AT32UC3L0128	AT32UC3L0128-AUT	Tray	TQFP 48	JESD97 Classification E3	
	AT32UC3L0128-AUR	Tape & Reel			
	AT32UC3L0128-ZAUT	Tray	QFN 48		
	AT32UC3L0128-ZAUR	Tape & Reel			
	AT32UC3L0128-D3HT	Tray	TLLGA 48	JESD97 Classification E4	
	AT32UC3L0128-D3HR	Tape & Reel			

6. Errata

6.1 Rev. C

6.1.1 SCIF

1. **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.

2. **PLLCOUNT value larger than zero can cause PLLEN glitch**
   Initializing the PLLCOUNT with a value greater than zero creates a glitch on the PLLEN signal during asynchronous wake up.
   Fix/Workaround
   The lock-masking mechanism for the PLL should not be used.
   The PLLCOUNT field of the PLL Control Register should always be written to zero.

3. **Writing 0x5A5A5A5A to the SCIF memory range will enable the SCIF UNLOCK feature**
   The SCIF UNLOCK feature will be enabled if the value 0x5A5A5A5A is written to any location in the SCIF memory range.
   Fix/Workaround
   None.

6.1.2 SPI

1. **SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0**
   When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.
   Fix/Workaround
   Disable mode fault detection by writing a one to MR.MODFDIS.

2. **Disabling SPI has no effect on the SR.TDRE bit**
   Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.
   Fix/Workaround
   Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

3. **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).

Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

6.2.6 TC

1. **Channel chaining skips first pulse for upper channel**
   When chaining two channels using the Block Mode Register, the first pulse of the clock between the channels is skipped.
   Fix/Workaround
   Configure the lower channel with RA = 0x1 and RC = 0x2 to produce a dummy clock cycle for the upper channel. After the dummy cycle has been generated, indicated by the SR.CPCS bit, reconfigure the RA and RC registers for the lower channel with the real values.

6.2.7 CAT

1. **CAT QMatrix sense capacitors discharged prematurely**
   At the end of a QMatrix burst charging sequence that uses different burst count values for different Y lines, the Y lines may be incorrectly grounded for up to n-1 periods of the peripheral bus clock, where n is the ratio of the PB clock frequency to the GCLK_CAT frequency. This results in premature loss of charge from the sense capacitors and thus increased variability of the acquired count values.
   Fix/Workaround
   Enable the 1kOhm drive resistors on all implemented QMatrix Y lines (CSA 1, 3, 5, 7, 9, 11, 13, and/or 15) by writing ones to the corresponding odd bits of the CSARES register.

2. **Autonomous CAT acquisition must be longer than AST source clock period**
   When using the AST to trigger CAT autonomous touch acquisition in sleep modes where the CAT bus clock is turned off, the CAT will start several acquisitions if the period of the AST source clock is larger than one CAT acquisition. One AST clock period after the AST trigger, the CAT clock will automatically stop and the CAT acquisition can be stopped prematurely, ruining the result.
   Fix/Workaround
   Always ensure that the ATCFG1.max field is set so that the duration of the autonomous touch acquisition is greater than one clock period of the AST source clock.

3. **CAT consumes unnecessary power when disabled or when autonomous touch not used**
   A CAT prescaler controlled by the ATCFG0.DIV field will be active even when the CAT module is disabled or when the autonomous touch feature is not used, thereby causing unnecessary power consumption.
   Fix/Workaround
   If the CAT module is not used, disable the CLK_CAT clock in the PM module. If the CAT module is used but the autonomous touch feature is not used, the power consumption of the CAT module may be reduced by writing 0xFFFF to the ATCFG0.DIV field.

6.2.8 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:

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

6.2.9 Flash

1. **Corrupted data in flash may happen after flash page write operations**
   After a flash page write operation from an external in situ programmer, reading (data read or code fetch) in flash may fail. This may lead to an exception or to others errors derived from this corrupted read access.
   Fix/Workaround
   Before any flash page write operation, each write in the page buffer must preceded by a write in the page buffer with 0xFFFF_FFFF content at any address in the page.

6.3 Rev. A

6.3.1 Device

1. **JTAGID is wrong**
   The JTAGID is 0x021DF03F.
   Fix/Workaround
   None.

6.3.2 FLASHCDW

1. **General-purpose fuse programming does not work**
   The general-purpose fuses cannot be programmed and are stuck at 1. Please refer to the Fuse Settings chapter in the FLASHCDW for more information about what functions are affected.
   Fix/Workaround
   None.

2. **Set Security Bit command does not work**
   The Set Security Bit (SSB) command of the FLASHCDW does not work. The device cannot be locked from external JTAG, aWire, or other debug accesses.
   Fix/Workaround
   None.

3. **Flash programming time is longer than specified**

The flash programming time is now:

Table 6-1. Flash Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{FPP}	Page programming time	f_{CLK_HSB}= 50MHz		7.5		ms
T_{FPE}	Page erase time			7.5		
T_{FFP}	Fuse programming time			1		
T_{FEA}	Full chip erase time (EA)			9		
T_{FCE}	JTAG chip erase time (CHIP_ERASE)	f_{CLK_HSB}= 115kHz		250		

Fix/Workaround
None.

4. **Power Manager**

5. **Clock Failure Detector (CFD) can be issued while turning off the CFD**
While turning off the CFD, the CFD bit in the Status Register (SR) can be set. This will change the main clock source to RCSYS.
Fix/Workaround
Solution 1: Enable CFD interrupt. If CFD interrupt is issues after turning off the CFD, switch back to original main clock source.
Solution 2: Only turn off the CFD while running the main clock on RCSYS.

6. **Sleepwalking in idle and frozen sleep mode will mask all other PB clocks**
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.

2. **Unused PB clocks are running**
Three unused PBA clocks are enabled by default and will cause increased active power consumption.
Fix/Workaround
Disable the clocks by writing zeroes to bits [27:25] in the PBA clock mask register.

6.3.3 SCIF

1. **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.

2. **PLL lock might not clear after disable**

6.3.5 GPIO

1. **Clearing Interrupt flags can mask other interrupts**
   When clearing interrupt flags in a GPIO port, interrupts on other pins of that port, happening in the same clock cycle will not be registered.
   Fix/Workaround
   Read the PVR register of the port before and after clearing the interrupt to see if any pin change has happened while clearing the interrupt. If any change occurred in the PVR between the reads, they must be treated as an interrupt.

6.3.6 SPI

1. **SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0**
   When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.
   Fix/Workaround
   Disable mode fault detection by writing a one to MR.MODFDIS.

2. **Disabling SPI has no effect on the SR.TDRE bit**
   Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.
   Fix/Workaround
   Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

3. **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).

4. **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.

5. **SPI mode fault detection enable causes incorrect behavior**
   When mode fault detection is enabled (MR.MODFDIS==0), the SPI module may not operate properly.
   Fix/Workaround
   Always disable mode fault detection before using the SPI by writing a one to MR.MODFDIS.

6. **SPI RDR.PCS is not correct**
   The PCS (Peripheral Chip Select) field in the SPI RDR (Receive Data Register) does not correctly indicate the value on the NPCS pins at the end of a transfer.
   Fix/Workaround
   Do not use the PCS field of the SPI RDR.

6.3.7 TWI

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.

2. **SMBALERT bit may be set after reset**
   The SMBus Alert (SMBALERT) bit in the Status Register (SR) might be erroneously set after system reset.
   Fix/Workaround
   After system reset, clear the SR.SMBALERT bit before commencing any TWI transfer.

3. **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.

4. **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.

5. **TWIM TWALM polarity is wrong**
   The TWALM signal in the TWIM is active high instead of active low.
   Fix/Workaround
   Use an external inverter to invert the signal going into the TWIM. When using both TWIM and TWIS on the same pins, the TWALM cannot be used.

6.3.8 PWMA

1. **The SR.READY bit cannot be cleared by writing to SCR.READY**
   The Ready bit in the Status Register will not be cleared when writing a one to the corresponding bit in the Status Clear register. The Ready bit will be cleared when the Busy bit is set.
   Fix/Workaround
   Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

6.3.9 TC

1. **Channel chaining skips first pulse for upper channel**
   When chaining two channels using the Block Mode Register, the first pulse of the clock between the channels is skipped.

6.3.12 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:

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

6.3.13 Flash

1. **Corrupted data in flash may happen after flash page write operations**
   After a flash page write operation from an external in situ programmer, reading (data read or code fetch) in flash may fail. This may lead to an exception or to others errors derived from this corrupted read access.
   Fix/Workaround
   Before any flash page write operation, each write in the page buffer must preceded by a write in the page buffer with 0xFFFF_FFFF content at any address in the page.

6.3.14 I/O Pins

1. **PA05 is not 3.3V tolerant.**
   PA05 should be grounded on the PCB and left unused if VDDIO is above 1.8V.
   Fix/Workaround
   None.

2. **No pull-up on pins that are not bonded**
   PB13 to PB27 are not bonded on UC3L0256/128, but has no pull-up and can cause current consumption on VDDIO/VDDIN if left undriven.
   Fix/Workaround
   Enable pull-ups on PB13 to PB27 by writing 0x0FFFE000 to the PUERS1 register in the GPIO.

3. **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.

7. 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.

7.1 Rev. C – 06/2013

1. Updated the datasheet with new Atmel blue logo and the last page.
2. Added Flash errata.

7.2 Rev. B – 01/2012

1. Description: DFLL frequency is 20 to 150MHz, not 40 to 150MHz.
2. Description: “One touch sensor can be configured to operate autonomously...” replaced by “All touch sensors can be configured to operate autonomously...”.
3. Block Diagram: GCLK_IN is input, not output, and is 2 bits wide (GCLK_IN[1..0]). CAT SMP corrected from I/O to output. SPI NPCS corrected from output to I/O.
4. Package and Pinout: PRND signal removed from Signal Descriptions List table and GPIO Controller Function Multiplexing table.
5. Supply and Startup Considerations: In 1.8V single supply mode figure, the input voltage is 1.62-1.98V, not 1.98-3.6V. “On system start-up, the DFLL is disabled” is replaced by “On system start-up, all high-speed clocks are disabled”.
6. ADCIFB: PRND signal removed from block diagram.
7. Electrical Characteristics: Added PLL source clock in the Clock Frequencies table in the Maximum Clock Frequencies section. Removed 64-pin package information from I/O Pin Characteristics tables and Digital Clock Characteristics table.
8. Electrical Characteristics: Removed USB Transceiver Characteristics, as the device contains no USB.
9. Mechanical Characteristics: Added notes to package drawings.
10. Summary: Removed Programming and Debugging chapter, added Processor and Architecture chapter.
11. Datasheet Revision History: Corrected release date for datasheet rev. A; the correct date is 12/2011.

7.3 Rev. A – 12/2011

1. Initial revision.

Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370