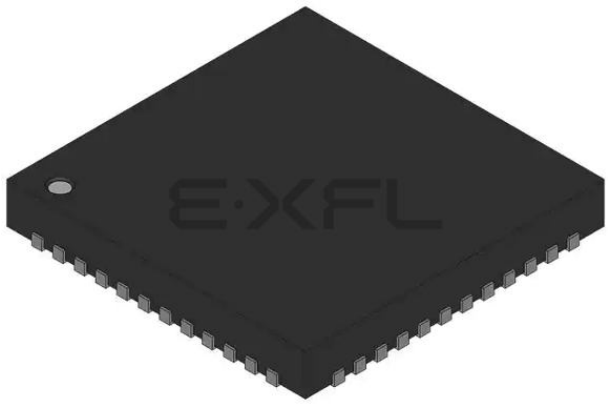


Welcome to E-XFL.COM

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, USB
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/atuc256l4u-zaur

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

3.2.5 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	64-pin	Pin Name	Oscillator Pin
3	3	PA08	XIN0
46	62	PA10	XIN32
26	34	PA13	XIN32_2
2	2	PA09	XOUT0
47	63	PA12	XOUT32
25	33	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 6.1.4.2 on page 44 for constraints on the WAKE_N pin.

Table 3-6. Other Functions

48-pin	64-pin	Pin Name	Function
27	35	PA11	WAKE_N
22	30	RESET_N	aWire DATA
11	15	PA00	aWire DATAOUT

13.6.3.1 Entering and exiting sleep modes

The sleep instruction will halt the CPU and all modules belonging to the stopped clock domains. The modules will be halted regardless of the bit settings in the mask registers.

Clock sources can also be switched off to save power. Some of these have a relatively long start-up time, and are only switched off when very low power consumption is required.

The CPU and affected modules are restarted when the sleep mode is exited. This occurs when an interrupt triggers. Note that even if an interrupt is enabled in sleep mode, it may not trigger if the source module is not clocked.

13.6.3.2 Supported sleep modes

The following sleep modes are supported. These are detailed in Table 13-2 on page 213.

- Idle: The CPU is stopped, the rest of the device is operational.
- Frozen: The CPU and HSB modules are stopped, peripherals are operational.
- Standby: All synchronous clocks are stopped, and the clock sources are running, allowing for a quick wake-up to normal mode.
- Stop: As Standby, but oscillators, and other clock sources are also stopped. 32KHz Oscillator OSC32K[2], RCSYS, AST, and WDT will remain operational.
- DeepStop: All synchronous clocks and clock sources are stopped. Bandgap voltage reference and BOD are turned off. OSC32K[2] and RCSYS remain operational.
- Static: All clock sources, including RCSYS are stopped. Bandgap voltage reference and BOD are turned off. OSC32K[2] remains operational.
- Shutdown: All clock sources, including RCSYS are stopped. Bandgap voltage reference, BOD detector, and Voltage regulator are turned off. OSC32K[2] remains operational. This mode can only be used in the "**3.3V supply mode, with 1.8V regulated I/O lines**" configuration (described in Power Considerations chapter). Refer to Section 13.6.4 for more details.

Table 13-2. Sleep Modes

Index[1]	Sleep Mode	CPU	HSB	PBx, GCLK	Clock Sources[3], SYSTIMER[4]	OSC32K[2]	RCSYS	BOD & Bandgap	Voltage Regulator
0	**Idle**	Stop	Run	Run	Run	Run	Run	On	Normal mode
1	**Frozen**	Stop	Stop	Run	Run	Run	Run	On	Normal mode
2	**Standby**	Stop	Stop	Stop	Run	Run	Run	On	Normal mode
3	**Stop**	Stop	Stop	Stop	Stop	Run	Run	On	Low power mode
4	**DeepStop**	Stop	Stop	Stop	Stop	Run	Run	Off	Low power mode
5	**Static**	Stop	Stop	Stop	Stop	Run	Stop	Off	Low power mode
6	**Shutdown**	Stop	Stop	Stop	Stop	Run	Stop	Off	Off

Notes:
1. The sleep mode index is used as argument for the sleep instruction.
2. OSC32K will only remain operational if pre-enabled.
3. Clock sources other than those specifically listed in the table.
4. SYSTIMER is the clock for the CPU COUNT and COMPARE registers.

The internal voltage regulator is also adjusted according to the sleep mode in order to reduce its power consumption.

13.7.11 Interrupt Mask Register

Name: IMR

Access Type: Read-only

Offset: 0x0C8

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
AE	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	CKRDY	-	-	-	-	CFD

0: The corresponding interrupt is disabled.
1: The corresponding interrupt is enabled.
This bit is cleared when the corresponding bit in IDR is written to one.
This bit is set when the corresponding bit in IER is written to one.

14.6.32 PLL Version Register

Name: PLLVERSION

Access Type: Read-only

Offset: 0x03C4

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	VARIANT			

15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **VARIANT: Variant number**
  Reserved. No functionality associated.
- **VERSION: Version number**
  Version number of the module. No functionality associated.

14.6.41 120MHz RC Oscillator Version Register

Name: RC120MIFAVERSION

Access Type: Read-only

Offset: 0x03EC

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	VARIANT			

15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **VARIANT: Variant number**
  Reserved. No functionality associated.
- **VERSION: Version number**
  Version number of the module. No functionality associated.

.

Table 14-12. PLL Clock Sources

PLLOSC	Clock/Oscillator	Description
0	OSC0	Output clock from Oscillator0
1	GCLK8	Generic clock 8
2-3	Reserved	

Table 14-13. Generic Clock number of DIV bits

Generic Clock	Number of DIV bits
0	8
1	8
2	8
3	8
4	8
5	8
6	8
7	8
8	8
9	16

Table 14-14. Register Reset Values

Register	Reset Value
CMVERSION	0x00000100
PLLVERSION	0x00000110
OSC0VERSION	0x00000111
OSC32VERSION	0x00000110
DFLLIFVERSION	0x00000210
BODIFAVERSION	0x00000120
VREGIFBVERSION	0x00000110
RCOSCIFAVERSION	0x00000111
SM33IFAVERSION	0x00000110
TSENSEIFAVERSION	0x00000100
RC120MIFAVERSION	0x00000110
BRIFAVERSION	0x00000100

15.7 Module Configuration

The specific configuration for each AST instance is listed in the following tables.The module bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 15-3. AST Configuration

Feature	AST
Number of alarm comparators	1
Number of periodic comparators	1
Digital tuner	On

Table 15-4. AST Clocks

Clock Name	Description
CLK_AST	Clock for the AST bus interface
GCLK	The generic clock used for the AST is GCLK2
PB clock	Peripheral Bus clock from the PBA clock domain

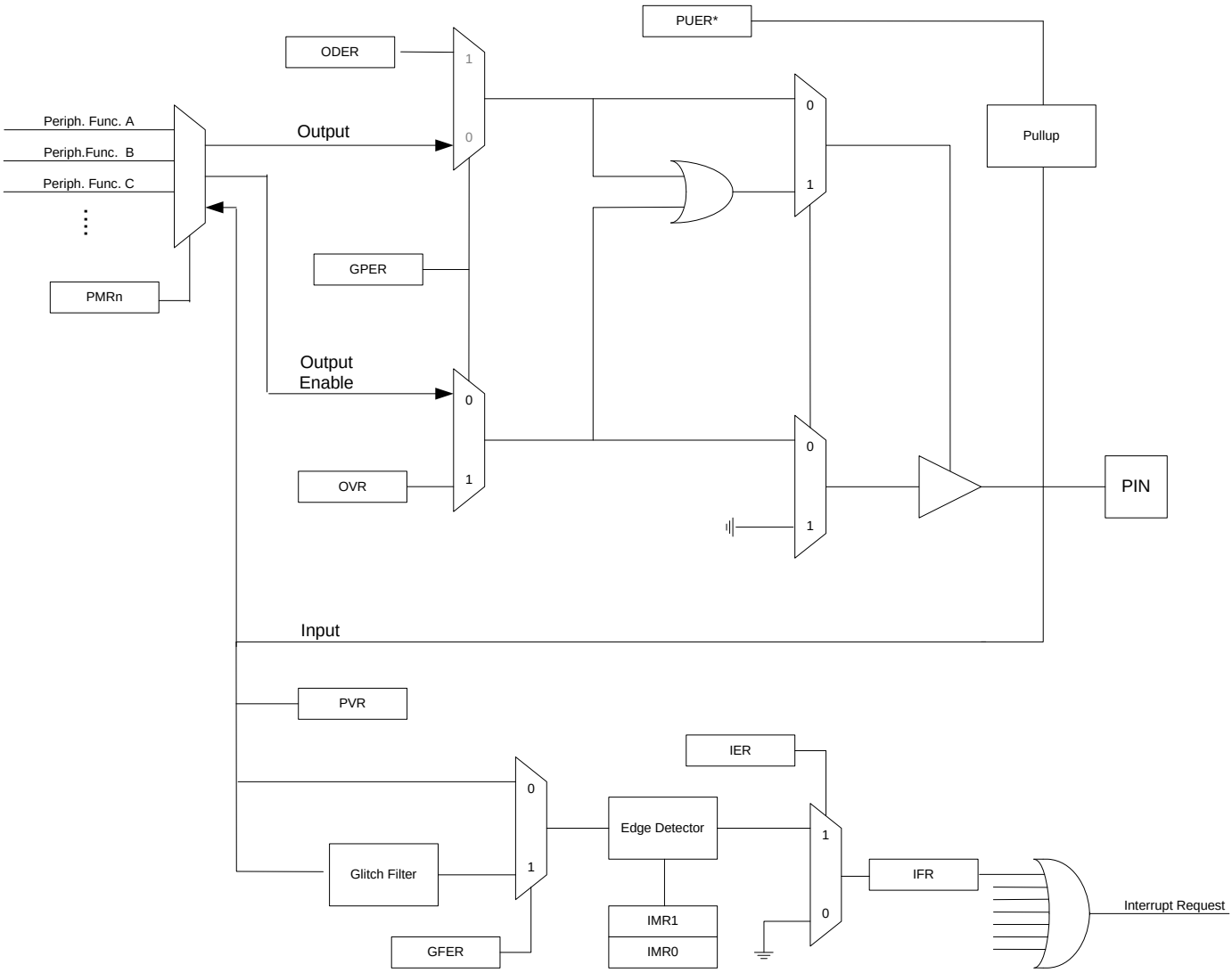
Table 15-5. Register Reset Values

Register	Reset Value
VERSION	0x00000310
PARAMETER	0x00004103

19.6 Functional Description

The GPIO controls the I/O pins of the microcontroller. The control logic associated with each pin is shown in the figure below.

Figure 19-2. Overview of the GPIO



*) Register value is overrided if a peripheral function that support this function is enabled

19.7.14 Interrupt Mode Register 1

Name: IMR1

Access: Read/Write, Set, Clear, Toggle

Offset: 0x0B0, 0x0B4, 0x0B8, 0x0BC

Reset Value: -

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0-31: Interrupt Mode Bit 1**

{IMR1, IMR0}	Interrupt Mode
00	Pin Change
01	Rising Edge
10	Falling Edge
11	Reserved

22. Two-wire Master Interface (TWIM)

Rev.: 1.1.0.1

22.1 Features

- **Compatible with I²C standard**
  - **Multi-master support**
  - **Transfer speeds of 100 and 400 kbit/s**
  - **7- and 10-bit and General Call addressing**
- **Compatible with SMBus standard**
  - **Hardware Packet Error Checking (CRC) generation and verification with ACK control**
  - **SMBus ALERT interface**
  - **25 ms clock low timeout delay**
  - **10 ms master cumulative clock low extend time**
  - **25 ms slave cumulative clock low extend time**
- **Compatible with PMBus**
- **Compatible with Atmel Two-wire Interface Serial Memories**
- **DMA interface for reducing CPU load**
- **Arbitrary transfer lengths, including 0 data bytes**
- **Optional clock stretching if transmit or receive buffers not ready for data transfer**

22.2 Overview

The Atmel Two-wire Master Interface (TWIM) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 kbit/s, based on a byte-oriented transfer format. It can be used with any Atmel Two-wire Interface bus serial EEPROM and I²C compatible device such as a real time clock (RTC), dot matrix/graphic LCD controller, and temperature sensor, to name a few. The TWIM is always a bus master and can transfer sequential or single bytes. Multiple master capability is supported. Arbitration of the bus is performed internally and relinquishes the bus automatically if the bus arbitration is lost.

A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.Table 22-1 lists the compatibility level of the Atmel Two-wire Interface in Master Mode and a full I²C compatible device.

Table 22-1. Atmel TWIM Compatibility with I²C Standard

I²C Standard	Atmel TWIM
Standard-mode (100 kbit/s)	Supported
Fast-mode (400 kbit/s)	Supported
Fast-mode Plus (1 Mbit/s)	Supported
7- or 10-bits Slave Addressing	Supported
START BYTE(1)	Not Supported
Repeated Start (Sr) Condition	Supported
ACK and NACK Management	Supported
Slope Control and Input Filtering (Fast mode)	Supported
Clock Stretching	Supported

Note: 1. START + b000000001 + Ack + Sr

25.7.2 Interlinked Single Value Duty Register

Name: ISDUTY

Access Type: Write-only

Offset: 0x04

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	DUTY[11:8]			

7	6	5	4	3	2	1	0
DUTY[7:0]							

- **DUTY: Duty Cycle Value**

  The duty cycle value written to this field is written simultaneously to all channels selected in the ISCHSETm register.

  If the value zero is written to DUTY all affected channels will be disabled. In this state the output waveform will be zero all the time.

25.7.16 Composite Waveform Generation

Name: CWG

Access Type: Read/Write

Offset: 0x3C+k*0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
XOR							

23	22	21	20	19	18	17	16
XOR							

15	14	13	12	11	10	9	8
XOR							

7	6	5	4	3	2	1	0
XOR							

- **XOR: Pair Waveform XOR'ed**

  If the bit *n* in XOR field is one, the pair of PWMA output waveforms will be XORed before output. The even number output will be the XOR'ed output and the odd number output will be reverse of it. For example, if bit 0 in XOR is one, the pair of PWMA output waveforms for channel 0 and 1 will be XORed together.

  If bit *n* in XOR is zero, normal waveforms are output for that pair. Note that

  If more than one CWG register is present, CWG0 controls the first 32 pairs, corresponding to channels 63 downto 0, and CWG1 controls the second 32 pairs, corresponding to channels 127 downto 64.

26.7.15 Version Register

Name: VERSION

Access Type: Read-only

Offset: 0xFC

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	VARIANT			

15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **VARIANT: Variant number**
  Reserved. No functionality associated.
- **VERSION: Version number**
  Version number of the module. No functionality associated.

28.7.7 Interrupt Disable Register

Name: IDR

Access Type: Write-only

Offset: 0x18

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	TXUR	TXRDY	-

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in IMR.

29.8 Operating Modes

The ADCIFB features two operating modes, each defining a separate conversion sequence:

- ADC Mode: At each trigger, all the enabled channels are converted.
- Resistive Touch Screen Mode: At each trigger, all enabled channels plus the resistive touch screen channels are converted as described in Section 29.8.3. If channels except the dedicated resistive touch screen channels are enabled, they are converted normally before and after the resistive touch screen channels are converted.

The operating mode is selected by the TSAMOD field in the Mode Register (MR).

29.8.1 Conversion Triggers

A conversion sequence is started either by a software or by a hardware trigger. When a conversion sequence is started, all enabled channels will be converted and made available in the shared Last Converted Register (LCDR).

The software trigger is asserted by writing a one to the START field in the Control Register (CR).

The hardware trigger can be selected by the TRGMOD field in the Trigger Register (TRGR). Different hardware triggers exist:

- External trigger, either rising or falling or any, detected on the external trigger pin TRIGGER
- Pen detect trigger, depending the PENDET bit in the Mode Register (MR)
- Continuous trigger, meaning the ADCIFB restarts the next sequence as soon as it finishes the current one
- Periodic trigger, which is defined by the TRGR.TRGPER field
- Peripheral event trigger, allowing the Peripheral Event System to synchronize conversion with some configured peripheral event source.

Enabling a hardware trigger does not disable the software trigger functionality. Thus, if a hardware trigger is selected, the start of a conversion can still be initiated by the software trigger.

29.8.2 ADC Mode

In the ADC Mode, the active channels are defined by the Channel Status Register (CHSR). A channel is enabled by writing a one to the corresponding bit in the Channel Enable Register (CHER), and disabled by writing a one to the corresponding bit in the Channel Disable Register (CHDR). The conversion results are stored in the Last Converted Data Register (LCDR) as they become available, overwriting old conversions.

At each trigger, the following sequence is performed:

1. If ACR.SLEEP is one, wake up the ADC and wait for the startup time.
2. If Channel 0 is enabled, convert Channel 0 and store result in LCDR.
3. If Channel 1 is enabled, convert Channel 1 and store result in LCDR.
4. If Channel N is enabled, convert Channel N and store result in LCDR.
5. If ACR.SLEEP is one, place the ADC cell in a low-power state.

If the Peripheral DMA Controller is enabled, all converted values are transferred continuously into the memory buffer.

29.8.3 Resistive Touch Screen Mode

Writing a one to the TSAMOD field in the Mode Register (MR) enables Resistive Touch Screen Mode. In this mode the channels TSPO+0 to TSPO+3, corresponding to the resistive touch

31.7.4 Autonomous/DMA Touch Configuration Register 0

Name: ATCFG0

Access Type: Read/Write

Offset: 0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
DIV[15:8]							
23	22	21	20	19	18	17	16
DIV[7:0]							
15	14	13	12	11	10	9	8
CHLEN							
7	6	5	4	3	2	1	0
SELEN							

- **DIV: Clock Divider**
  The prescaler is used to ensure that the CLK_CAT clock is divided to around 1 MHz to produce the sampling clock.The prescaler uses the following formula to generate the sampling clock:
  Sampling clock = CLK_CAT / (2(DIV+1))
- **CHLEN: Charge Length**
  For the autonomous QTouch sensor and DMATouch sensors, specifies how many sample clock cycles should be used for transferring charge to the sense capacitor.
- **SELEN: Settle Length**
  For the autonomous QTouch sensor and DMATouch sensors, specifies how many sample clock cycles should be used for settling after charge transfer.

Table 35-44. SPI Timing, Slave Mode[1]

Symbol	Parameter	Conditions	Min	Max	Units
SPI6	SPCK falling to MISO delay	V_{VDDIO} from 3.0V to 3.6V, maximum external capacitor = 40pF		29.4	ns
SPI7	MOSI setup time before SPCK rises		0		
SPI8	MOSI hold time after SPCK rises		6.0		
SPI9	SPCK rising to MISO delay			29.0	
SPI10	MOSI setup time before SPCK falls		0		
SPI11	MOSI hold time after SPCK falls		5.5		
SPI12	NPCS setup time before SPCK rises		3.4		
SPI13	NPCS hold time after SPCK falls		1.1		
SPI14	NPCS setup time before SPCK falls		3.3		
SPI15	NPCS hold time after SPCK rises		0.7		

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.

Maximum SPI Frequency, Slave Input Mode

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

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

Where $SPIn$ is the MOSI setup and hold time, SPI7 + SPI8 or SPI10 + SPI11 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:

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

Where $SPIn$ is the MISO delay, SPI6 or SPI9 depending on CPOL and NCPHA. t_{SETUP} is the SPI master setup time. Please refer to the SPI master datasheet for t_{SETUP} . 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.

35.10.5 TWIM/TWIS Timing

Figure 35-45 shows the TWI-bus timing requirements and the compliance of the device with them. Some of these requirements (t_r and t_f) are met by the device without requiring user intervention. Compliance with the other requirements (t_{HD-STA}, t_{SU-STA}, t_{SU-STO}, t_{HD-DAT}, $t_{SU-DAT-TWI}$, $t_{LOW-TWI}$, t_{HIGH}, and f_{TWCK}) requires user intervention through appropriate programming of the relevant

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.

38.3.6 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.

38.3.7 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 1 kOhm 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.

38.3.8 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

38.5.12 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:

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

38.5.13 Flash

5. **Corrupted data in flash may happen after flash page write operations**
   After a flash page write operation from an external 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.

38.5.14 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.

39. 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.

39.1 Rev. D – 06/2013

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

39.2 Rev. C – 01/2012

1. Description: DFLL frequency is 20 to 150MHz, not 40 to 150MHz.
2. Block Diagram: GCLK_IN is input, not output. CAT SMP corrected from I/O to output. SPI NPCS corrected from output to I/O.
3, Package and Pinout: EXTINT0 in Signal Descriptions table is NMI.
4, 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”.
5, ADCIFB: PRND signal removed from block diagram.
6, Electrical Charateristics: Added 64-pin package information to I/O Pin Characteristics tables and Digital Clock Characteristics table.
7, Mechanical Characteristics: QFN48 Package Drawing updated. Note that the package drawing for QFN48 is correct in datasheet rev A, but wrong in rev B. Added notes to package drawings.
8. Summary: Removed Programming and Debugging chapter, added Processor and Architecture chapter.

39.3 Rev. B – 12/2011

1. JTAG Data Registers subchapter added in the Programming and Debugging chapter, containing JTAG IDs.

39.4 Rev. A – 12/2011

1. Initial revision.