



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.

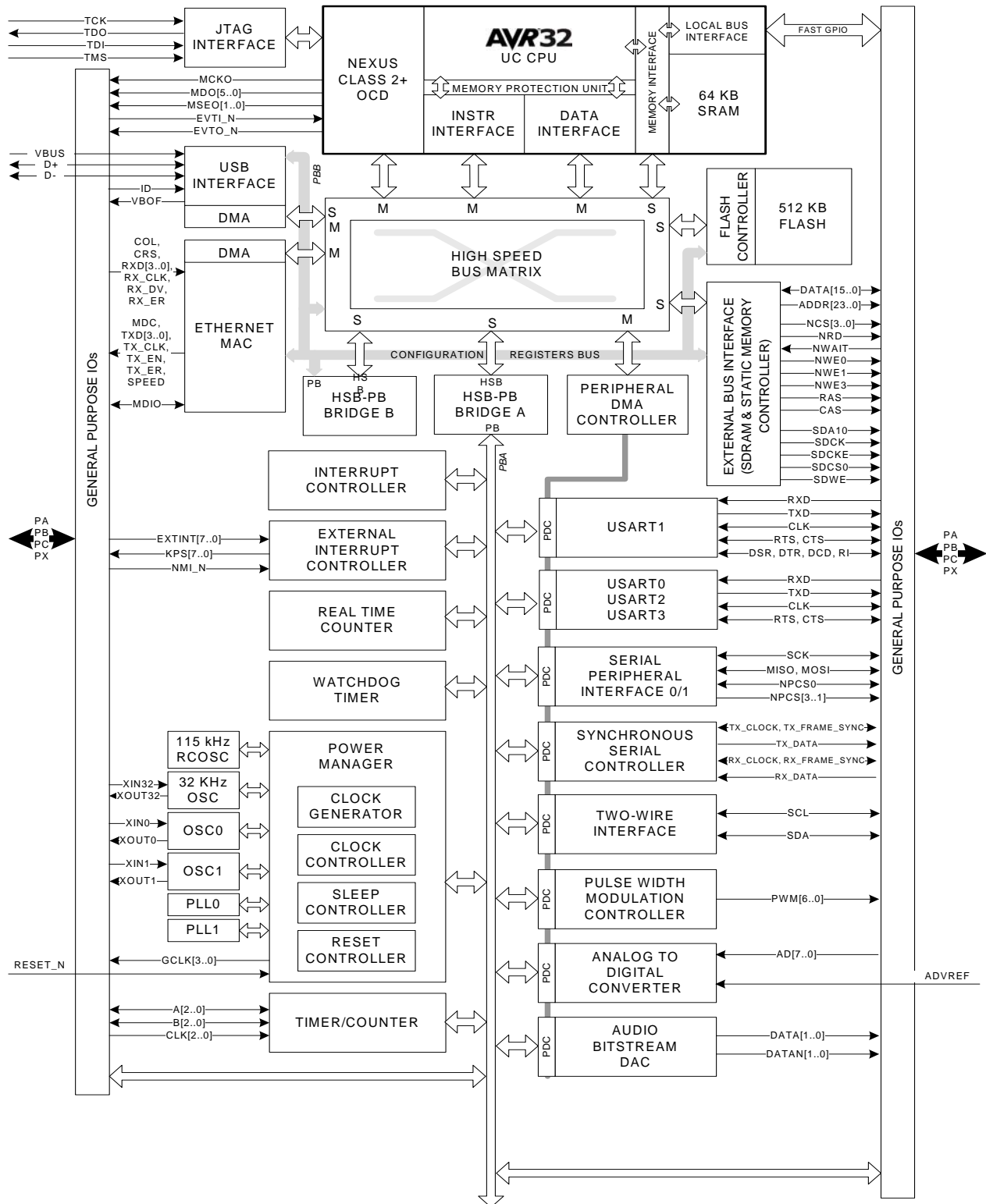
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, Ethernet, I <sup>2</sup> C, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	109
Program Memory Size	128KB (128K 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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-TFBGA
Supplier Device Package	144-FFBGA (11x11)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a0128-ctut">https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a0128-ctut</a>

## 4. Blockdiagram

Figure 4-1. Blockdiagram



## 4.1 Processor and architecture

### 4.1.1 AVR32 UC CPU

- 32-bit load/store AVR32A RISC architecture.
  - 15 general-purpose 32-bit registers.
  - 32-bit Stack Pointer, Program Counter and Link Register reside in register file.
  - Fully orthogonal instruction set.
  - Privileged and unprivileged modes enabling efficient and secure Operating Systems.
  - Innovative instruction set together with variable instruction length ensuring industry leading code density.
  - DSP extension with saturating arithmetic, and a wide variety of multiply instructions.
- 3 stage pipeline allows one instruction per clock cycle for most instructions.
  - Byte, half-word, word and double word memory access.
  - Multiple interrupt priority levels.
- MPU allows for operating systems with memory protection.

### 4.1.2 Debug and Test system

- IEEE1149.1 compliant JTAG and boundary scan
- Direct memory access and programming capabilities through JTAG interface
- Extensive On-Chip Debug features in compliance with IEEE-ISTO 5001-2003 (Nexus 2.0) Class 2+
  - Low-cost NanoTrace supported.
- Auxiliary port for high-speed trace information
- Hardware support for 6 Program and 2 data breakpoints
- Unlimited number of software breakpoints supported
- Advanced Program, Data, Ownership, and Watchpoint trace supported

### 4.1.3 Peripheral DMA Controller

- Transfers from/to peripheral to/from any memory space without intervention of the processor.
- Next Pointer Support, forbids strong real-time constraints on buffer management.
- Fifteen channels
  - Two for each USART
  - Two for each Serial Synchronous Controller
  - Two for each Serial Peripheral Interface
  - One for each ADC
  - Two for each TWI Interface

### 4.1.4 Bus system

- High Speed Bus (HSB) matrix with 6 Masters and 6 Slaves handled
  - Handles Requests from the CPU Data Fetch, CPU Instruction Fetch, PDCA, USBB, Ethernet Controller, CPU SAB, and to internal Flash, internal SRAM, Peripheral Bus A, Peripheral Bus B, EBI.
  - Round-Robin Arbitration (three modes supported: no default master, last accessed default master, fixed default master)
  - Burst Breaking with Slot Cycle Limit
  - One Address Decoder Provided per Master

**Table 5-1.** Signal Description List

Signal Name	Function	Type	Active Level	Comments
MSEO0 - MSEO1	Trace Frame Control	Output		
EVTI_N	Event In	Output	Low	
EVTO_N	Event Out	Output	Low	
<b>Power Manager - PM</b>				
GCLK0 - GCLK3	Generic Clock Pins	Output		
RESET_N	Reset Pin	Input	Low	
<b>Real Time Counter - RTC</b>				
RTC_CLOCK	RTC clock	Output		
<b>Watchdog Timer - WDT</b>				
WDTEXT	External Watchdog Pin	Output		
<b>External Interrupt Controller - EIC</b>				
EXTINT0 - EXTINT7	External Interrupt Pins	Input		
KPS0 - KPS7	Keypad Scan Pins	Output		
NMI_N	Non-Maskable Interrupt Pin	Input	Low	
<b>Ethernet MAC - MACB</b>				
COL	Collision Detect	Input		
CRS	Carrier Sense and Data Valid	Input		
MDC	Management Data Clock	Output		
MDIO	Management Data Input/Output	I/O		
RXD0 - RXD3	Receive Data	Input		
RX_CLK	Receive Clock	Input		
RX_DV	Receive Data Valid	Input		
RX_ER	Receive Coding Error	Input		
SPEED	Speed			
TXD0 - TXD3	Transmit Data	Output		
TX_CLK	Transmit Clock or Reference Clock	Output		
TX_EN	Transmit Enable	Output		
TX_ER	Transmit Coding Error	Output		

## 7. Power Considerations

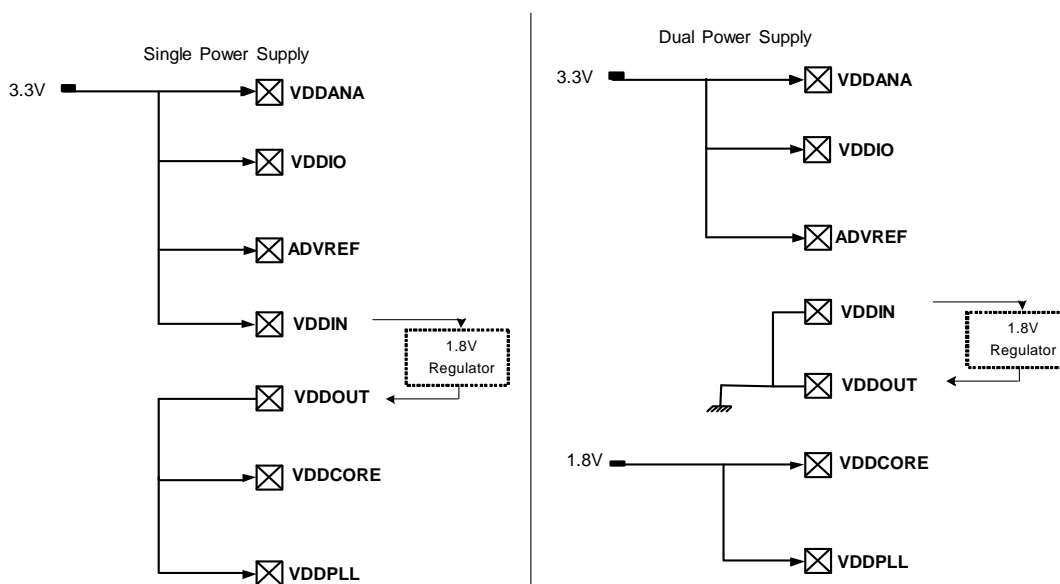
### 7.1 Power Supplies

The AT32UC3A has several types of power supply pins:

- **VDDIO:** Powers I/O lines. Voltage is 3.3V nominal.
- **VDDANA:** Powers the ADC. Voltage is 3.3V nominal.
- **VDDIN:** Input voltage for the voltage regulator. Voltage is 3.3V nominal.
- **VDDCORE:** Powers the core, memories, and peripherals. Voltage is 1.8V nominal.
- **VDDPLL:** Powers the PLL. Voltage is 1.8V nominal.

The ground pins GND are common to VDDCORE, VDDIO, VDDPLL. The ground pin for VDDANA is GNDANA.

Refer to ["Power Consumption" on page 44](#) for power consumption on the various supply pins.



**Table 10-8.** PDC Handshake Signals

PID Value	Peripheral module & direction
4	USART2 - RX
5	USART3 - RX
6	TWI - RX
7	SPI0 - RX
8	SPI1 - RX
9	SSC - TX
10	USART0 - TX
11	USART1 - TX
12	USART2 - TX
13	USART3 - TX
14	TWI - TX
15	SPI0 - TX
16	SPI1 - TX
17	ABDAC

## 10.7 Peripheral Multiplexing on I/O lines

Each GPIO line can be assigned to one of 3 peripheral functions; A, B or C. The following table define how the I/O lines on the peripherals A, B and C are multiplexed by the GPIO.

**Table 10-9.** GPIO Controller Function Multiplexing

TQFP100	VQFP144	PIN	GPIO Pin	Function A	Function B	Function C
19	25	PA00	GPIO 0	USART0 - RXD	TC - CLK0	
20	27	PA01	GPIO 1	USART0 - TXD	TC - CLK1	
23	30	PA02	GPIO 2	USART0 - CLK	TC - CLK2	
24	32	PA03	GPIO 3	USART0 - RTS	EIM - EXTINT[4]	DAC - DATA[0]
25	34	PA04	GPIO 4	USART0 - CTS	EIM - EXTINT[5]	DAC - DATAN[0]
26	39	PA05	GPIO 5	USART1 - RXD	PWM - PWM[4]	
27	41	PA06	GPIO 6	USART1 - TXD	PWM - PWM[5]	
28	43	PA07	GPIO 7	USART1 - CLK	PM - GCLK[0]	SPI0 - NPCS[3]
29	45	PA08	GPIO 8	USART1 - RTS	SPI0 - NPCS[1]	EIM - EXTINT[7]
30	47	PA09	GPIO 9	USART1 - CTS	SPI0 - NPCS[2]	MACB - WOL
31	48	PA10	GPIO 10	SPI0 - NPCS[0]	EIM - EXTINT[6]	
33	50	PA11	GPIO 11	SPI0 - MISO	USB - USB_ID	
36	53	PA12	GPIO 12	SPI0 - MOSI	USB - USB_VBOF	
37	54	PA13	GPIO 13	SPI0 - SCK		
39	56	PA14	GPIO 14	SSC - TX_FRAME_SYNC	SPI1 - NPCS[0]	EBI - NCS[0]
40	57	PA15	GPIO 15	SSC - TX_CLOCK	SPI1 - SCK	EBI - ADDR[20]

**Table 10-9. GPIO Controller Function Multiplexing**

41	58	PA16	GPIO 16	SSC - TX_DATA	SPI1 - MOSI	EBI - ADDR[21]
42	60	PA17	GPIO 17	SSC - RX_DATA	SPI1 - MISO	EBI - ADDR[22]
43	62	PA18	GPIO 18	SSC - RX_CLOCK	SPI1 - NPCS[1]	MACB - WOL
44	64	PA19	GPIO 19	SSC - RX_FRAME_SYNC	SPI1 - NPCS[2]	
45	66	PA20	GPIO 20	EIM - EXTINT[8]	SPI1 - NPCS[3]	
51	73	PA21	GPIO 21	ADC - AD[0]	EIM - EXTINT[0]	USB - USB_ID
52	74	PA22	GPIO 22	ADC - AD[1]	EIM - EXTINT[1]	USB - USB_VBOF
53	75	PA23	GPIO 23	ADC - AD[2]	EIM - EXTINT[2]	DAC - DATA[1]
54	76	PA24	GPIO 24	ADC - AD[3]	EIM - EXTINT[3]	DAC - DATAN[1]
55	77	PA25	GPIO 25	ADC - AD[4]	EIM - SCAN[0]	EBI - NCS[0]
56	78	PA26	GPIO 26	ADC - AD[5]	EIM - SCAN[1]	EBI - ADDR[20]
57	79	PA27	GPIO 27	ADC - AD[6]	EIM - SCAN[2]	EBI - ADDR[21]
58	80	PA28	GPIO 28	ADC - AD[7]	EIM - SCAN[3]	EBI - ADDR[22]
83	122	PA29	GPIO 29	TWI - SDA	USART2 - RTS	
84	123	PA30	GPIO 30	TWI - SCL	USART2 - CTS	
65	88	PB00	GPIO 32	MACB - TX_CLK	USART2 - RTS	USART3 - RTS
66	90	PB01	GPIO 33	MACB - TX_EN	USART2 - CTS	USART3 - CTS
70	96	PB02	GPIO 34	MACB - TXD[0]	DAC - DATA[0]	
71	98	PB03	GPIO 35	MACB - TXD[1]	DAC - DATAN[0]	
72	100	PB04	GPIO 36	MACB - CRS	USART3 - CLK	EBI - NCS[3]
73	102	PB05	GPIO 37	MACB - RXD[0]	DAC - DATA[1]	
74	104	PB06	GPIO 38	MACB - RXD[1]	DAC - DATAN[1]	
75	106	PB07	GPIO 39	MACB - RX_ER		
76	111	PB08	GPIO 40	MACB - MDC		
77	113	PB09	GPIO 41	MACB - MDIO		
78	115	PB10	GPIO 42	MACB - TXD[2]	USART3 - RXD	EBI - SDCK
81	119	PB11	GPIO 43	MACB - TXD[3]	USART3 - TXD	EBI - SDCKE
82	121	PB12	GPIO 44	MACB - TX_ER	TC - CLK0	EBI - RAS
87	126	PB13	GPIO 45	MACB - RXD[2]	TC - CLK1	EBI - CAS
88	127	PB14	GPIO 46	MACB - RXD[3]	TC - CLK2	EBI - SDWE
95	134	PB15	GPIO 47	MACB - RX_DV		
96	136	PB16	GPIO 48	MACB - COL	USB - USB_ID	EBI - SDA10
98	139	PB17	GPIO 49	MACB - RX_CLK	USB - USB_VBOF	EBI - ADDR[23]
99	141	PB18	GPIO 50	MACB - SPEED	ADC - TRIGGER	PWM - PWM[6]
100	143	PB19	GPIO 51	PWM - PWM[0]	PM - GCLK[0]	EIM - SCAN[4]
1	3	PB20	GPIO 52	PWM - PWM[1]	PM - GCLK[1]	EIM - SCAN[5]
2	5	PB21	GPIO 53	PWM - PWM[2]	PM - GCLK[2]	EIM - SCAN[6]
3	6	PB22	GPIO 54	PWM - PWM[3]	PM - GCLK[3]	EIM - SCAN[7]
6	9	PB23	GPIO 55	TC - A0	USART1 - DCD	

- Supports Mobile SDRAM Devices
- Error Detection
  - Refresh Error Interrupt
- SDRAM Power-up Initialization by Software
- CAS Latency of 1, 2, 3 Supported
- Auto Precharge Command Not Used

## 10.11.4 USB Controller

- USB 2.0 Compliant, Full-/Low-Speed (FS/LS) and On-The-Go (OTG), 12 Mbit/s
- 7 Pipes/Endpoints
- 960 bytes of Embedded Dual-Port RAM (DPRAM) for Pipes/Endpoints
- Up to 2 Memory Banks per Pipe/Endpoint (Not for Control Pipe/Endpoint)
- Flexible Pipe/Endpoint Configuration and Management with Dedicated DMA Channels
- On-Chip Transceivers Including Pull-Ups

## 10.11.5 Serial Peripheral Interface

- Supports communication with serial external devices
  - Four chip selects with external decoder support allow communication with up to 15 peripherals
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD Controllers, CAN Controllers and Sensors
  - External co-processors
- Master or slave serial peripheral bus interface
  - 8- to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delays between consecutive transfers and between clock and data per chip select
  - Programmable delay between consecutive transfers
  - Selectable mode fault detection
- Very fast transfers supported
  - Transfers with baud rates up to Peripheral Bus A (PBA) max frequency
  - The chip select line may be left active to speed up transfers on the same device

## 10.11.6 Two-wire Interface

- High speed up to 400kbit/s
- Compatibility with standard two-wire serial memory
- One, two or three bytes for slave address
- Sequential read/write operations

## 10.11.7 USART

- Programmable Baud Rate Generator
- 5- to 9-bit full-duplex synchronous or asynchronous serial communications
  - 1, 1.5 or 2 stop bits in Asynchronous Mode or 1 or 2 stop bits in Synchronous Mode
  - Parity generation and error detection
  - Framing error detection, overrun error detection
  - MSB- or LSB-first
  - Optional break generation and detection
  - By 8 or by-16 over-sampling receiver frequency
  - Hardware handshaking RTS-CTS
  - Receiver time-out and transmitter timeguard
  - Optional Multi-drop Mode with address generation and detection

## 11. Boot Sequence

This chapter summarizes the boot sequence of the AT32UC3A. The behaviour after power-up is controlled by the Power Manager. For specific details, refer to [Section 13. "Power Manager \(PM\)" on page 53](#).

### 11.1 Starting of clocks

After power-up, the device will be held in a reset state by the Power-On Reset circuitry, until the power has stabilized throughout the device. Once the power has stabilized, the device will use the internal RC Oscillator as clock source.

On system start-up, the PLLs are disabled. All clocks to all modules are running. No clocks have a divided frequency, all parts of the system receives a clock with the same frequency as the internal RC Oscillator.

### 11.2 Fetching of initial instructions

After reset has been released, the AVR32 UC CPU starts fetching instructions from the reset address, which is 0x8000\_0000. This address points to the first address in the internal Flash.

The code read from the internal Flash is free to configure the system to use for example the PLLs, to divide the frequency of the clock routed to some of the peripherals, and to gate the clocks to unused peripherals.

## 12. Electrical Characteristics

### 12.1 Absolute Maximum Ratings\*

Operating Temperature .....	-40°C to +85°C
Storage Temperature .....	-60°C to +150°C
Voltage on Input Pin with respect to Ground except for PC00, PC01, PC02, PC03, PC04, PC05.....	-0.3V to 5.5V
Voltage on Input Pin with respect to Ground for PC00, PC01, PC02, PC03, PC04, PC05.....	-0.3V to 3.6V
Maximum Operating Voltage (VDDCORE, VDDPLL) .....	1.95V
Maximum Operating Voltage (VDDIO, VDDIN, VDDANA).....	3.6V
Total DC Output Current on all I/O Pin for TQFP100 package .....	370 mA
for LQGP144 package .....	470 mA

\*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

These figures represent the power consumption measured on the power supplies.

**Table 12-9.** Power Consumption for Different Modes

Mode	Conditions		Typ.	Unit
Active	Typ : Ta = 25 °C CPU running from flash <sup>(1)</sup> . VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. <sup>(1)</sup> XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	9	mA
		f = 24 MHz	15	mA
		f = 36MHz	20	mA
		f = 50 MHz	28	mA
		f = 66 MHz	36.3	mA
Idle	Typ : Ta = 25 °C CPU running from flash <sup>(1)</sup> . VDDIN=3.3 V. VDDCORE =1.8V. CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	5	mA
		f = 24 MHz	10	mA
		f = 36MHz	14	mA
		f = 50 MHz	19	mA
		f = 66 MHz	25.5	mA
Frozen	Typ : Ta = 25 °C CPU running from flash <sup>(1)</sup> . CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	3	mA
		f = 24 MHz	6	mA
		f = 36MHz	9	mA
		f = 50 MHz	13	mA
		f = 66 MHz	16.8	mA
Standby	Typ : Ta = 25 °C CPU running from flash <sup>(1)</sup> . CPU clocked from PLL0 at f MHz Voltage regulator is on. XIN0 : external clock. XIN1 stopped. XIN32 stopped PLL0 running All peripheral clocks activated. GPIOs on internal pull-up. JTAG unconnected with ext pull-up.	f = 12 MHz	1	mA
		f = 24 MHz	2	mA
		f = 36MHz	3	mA
		f = 50 MHz	4	mA
		f = 66 MHz	4.8	mA

**Table 12-9.** Power Consumption for Different Modes

Mode	Conditions		Typ.	Unit
Stop	Typ : Ta = 25 °C. CPU is in stop mode GPIOs on internal pull-up. All peripheral clocks de-activated. DM and DP pins connected to ground. XIN0,Xin1 and XIN2 are stopped	on Amp0	47	uA
		on Amp1	40	uA
Deepstop	Typ : Ta = 25 °C.CPU is in deepstop mode GPIOs on internal pull-up. All peripheral clocks de-activated. DM and DP pins connected to ground. XIN0,Xin1 and XIN2 are stopped	on Amp0	36	uA
		on Amp1	28	uA
Static	Typ : Ta = 25 °C. CPU is in static mode GPIOs on internal pull-up. All peripheral clocks de-activated. DM and DP pins connected to ground. XIN0,Xin1 and XIN2 are stopped	on Amp0	25	uA
		on Amp1	14	uA

1. Core frequency is generated from XIN0 using the PLL so that 140 MHz < fpll0 < 160 MHz and 10 MHz < fxin0 < 12MHz

**Table 12-10.** Power Consumption by Peripheral in Active Mode

Peripheral	Typ.	Unit
GPIO	37	μA/MHz
SMC	10	
SDRAMC	4	
ADC	18	
EBI	31	
INTC	25	
TWI	14	
MACB	45	
PDCA	30	
PWM	36	
RTC	7	
SPI	13	
SSC	13	
TC	10	
USART	35	
USB	45	

## 12.6 Clock Characteristics

These parameters are given in the following conditions:



## 12.8 ADC Characteristics

**Table 12-17.** Channel Conversion Time and ADC Clock

Parameter	Conditions	Min	Typ	Max	Units
ADC Clock Frequency	10-bit resolution mode			5	MHz
ADC Clock Frequency	8-bit resolution mode			8	MHz
Startup Time	Return from Idle Mode			20	μs
Track and Hold Acquisition Time		600			ns
Conversion Time	ADC Clock = 5 MHz			2	μs
Conversion Time	ADC Clock = 8 MHz			1.25	μs
Throughput Rate	ADC Clock = 5 MHz			384 <sup>(1)</sup>	kSPS
Throughput Rate	ADC Clock = 8 MHz			533 <sup>(2)</sup>	kSPS

Notes: 1. Corresponds to 13 clock cycles at 5 MHz: 3 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.  
2. Corresponds to 15 clock cycles at 8 MHz: 5 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

**Table 12-18.** External Voltage Reference Input

Parameter	Conditions	Min	Typ	Max	Units
ADVREF Input Voltage Range		2.6		VDDANA	V
ADVREF Average Current	On 13 samples with ADC Clock = 5 MHz		200	250	μA
Current Consumption on VDDANA				1.25	mA

Note: ADVREF should be connected to GND to avoid extra consumption in case ADC is not used.

**Table 12-19.** Analog Inputs

Parameter	Min	Typ	Max	Units
Input Voltage Range	0		V <sub>ADVREF</sub>	
Input Leakage Current			1	μA
Input Capacitance		17		pF

**Table 12-20.** Transfer Characteristics in 8-bit mode

Parameter	Conditions	Min	Typ	Max	Units
Resolution			8		Bit
Absolute Accuracy	f=5MHz			0.8	LSB
	f=8MHz			1.5	LSB
Integral Non-linearity	f=5MHz		0.35	0.5	LSB
	f=8MHz		0.5	1.0	LSB
Differential Non-linearity	f=5MHz		0.3	0.5	LSB
	f=8MHz		0.5	1.0	LSB
Offset Error	f=5MHz	-0.5		0.5	LSB
Gain Error	f=5MHz	-0.5		0.5	LSB

Table 12-24. SMC Read Signals with no Hold Settings

Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC <sub>19</sub>	Data Setup before NRD High	13.7	ns
SMC <sub>20</sub>	Data Hold after NRD High	1	
NRD Controlled (READ_MODE = 0)			
SMC <sub>21</sub>	Data Setup before NCS High	13.3	ns
SMC <sub>22</sub>	Data Hold after NCS High	0	

Table 12-25. SMC Write Signals with Hold Settings

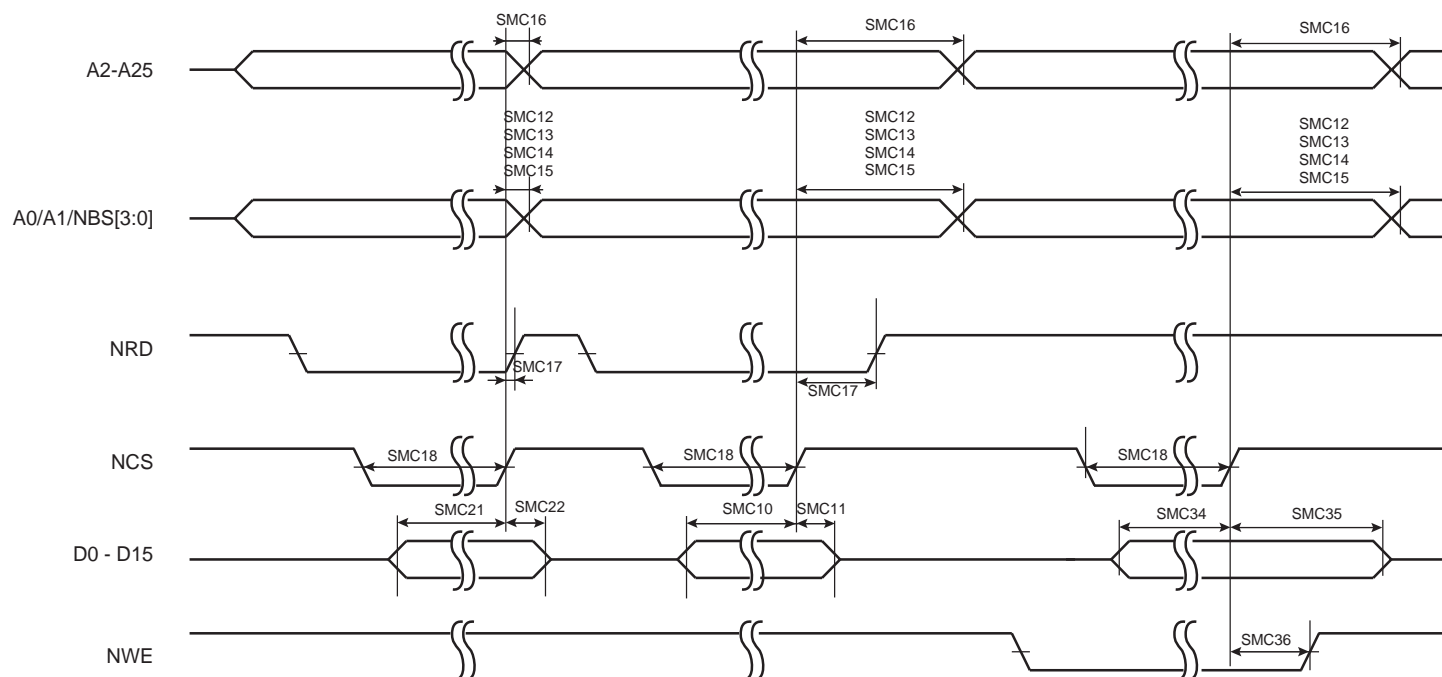
Symbol	Parameter	Min	Units
NRD Controlled (READ_MODE = 1)			
SMC <sub>23</sub>	Data Out Valid before NWE High	(nwe pulse length - 1) * t <sub>CPSMC</sub> - 0.9	ns
SMC <sub>24</sub>	Data Out Valid after NWE High <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 6	
SMC <sub>25</sub>	NWE High to NBS0/A0 Change <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 1.9	
SMC <sub>26</sub>	NWE High to NBS1 Change <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 1.9	
SMC <sub>29</sub>	NWE High to NBS2/A1 Change <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 1.9	
SMC <sub>30</sub>	NWE High to NBS3 Change <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 1.9	
SMC <sub>31</sub>	NWE High to A2 - A25 Change <sup>(1)</sup>	nwe hold length * t <sub>CPSMC</sub> - 1.7	
SMC <sub>32</sub>	NWE High to NCS Inactive <sup>(1)</sup>	(nwe hold length - ncs wr hold length)* t <sub>CPSMC</sub> - 2.9	
SMC <sub>33</sub>	NWE Pulse Width	nwe pulse length * t <sub>CPSMC</sub> - 0.9	
NRD Controlled (READ_MODE = 0)			
SMC <sub>34</sub>	Data Out Valid before NCS High	(ncs wr pulse length - 1)* t <sub>CPSMC</sub> - 4.6	ns
SMC <sub>35</sub>	Data Out Valid after NCS High <sup>(1)</sup>	ncs wr hold length * t <sub>CPSMC</sub> - 5.8	
SMC <sub>36</sub>	NCS High to NWE Inactive <sup>(1)</sup>	(ncs wr hold length - nwe hold length)* t <sub>CPSMC</sub> - 0.6	

Note: 1. hold length = total cycle duration - setup duration - pulse duration. "hold length" is for "ncs wr hold length" or "nwe hold length"

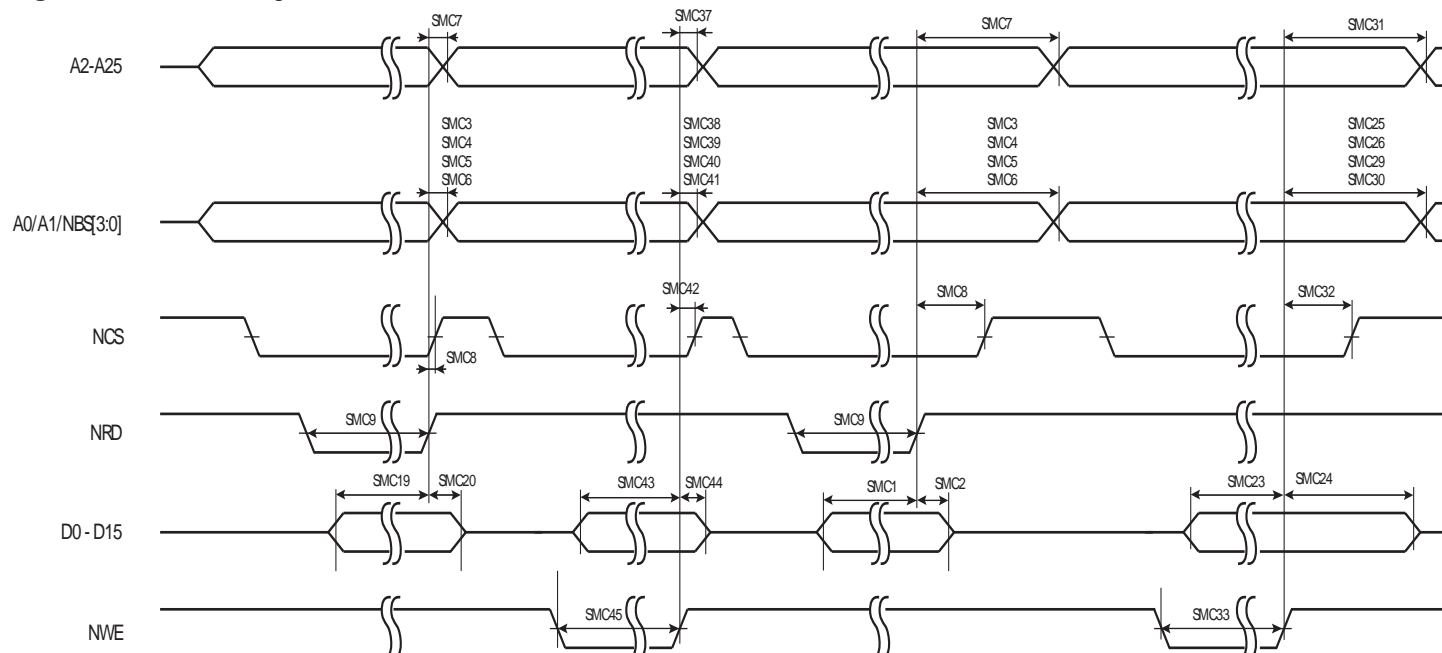
**Table 12-26.** SMC Write Signals with No Hold Settings (NWE Controlled only).

Symbol	Parameter	Min	Units
SMC <sub>37</sub>	NWE Rising to A2-A25 Valid	5.4	ns
SMC <sub>38</sub>	NWE Rising to NBS0/A0 Valid	5	
SMC <sub>39</sub>	NWE Rising to NBS1 Change	5	
SMC <sub>40</sub>	NWE Rising to A1/NBS2 Change	5	
SMC <sub>41</sub>	NWE Rising to NBS3 Change	5	
SMC <sub>42</sub>	NWE Rising to NCS Rising	5.1	
SMC <sub>43</sub>	Data Out Valid before NWE Rising	$(nwe \text{ pulse length} - 1) * t_{CPSMC} - 1.2$	
SMC <sub>44</sub>	Data Out Valid after NWE Rising	5	
SMC <sub>45</sub>	NWE Pulse Width	$nwe \text{ pulse length} * t_{CPSMC} - 0.9$	

**Figure 12-2.** SMC Signals for NCS Controlled Accesses.



**Figure 12-3.** SMC Signals for NRD and NRW Controlled Accesses.



## 12.9.1 SDRAM Signals

These timings are given for 10 pF load on SDCK and 40 pF on other signals.

**Table 12-27.** SDRAM Clock Signal.

Symbol	Parameter	Max <sup>(1)</sup>	Units
$1/(t_{CPSDCK})$	SDRAM Controller Clock Frequency	$1/(t_{cpcpu})$	MHz

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

**Table 12-28.** SDRAM Clock Signal.

Symbol	Parameter	Min	Units
SDRAMC <sub>1</sub>	SDCKE High before SDCK Rising Edge	7.4	ns
SDRAMC <sub>2</sub>	SDCKE Low after SDCK Rising Edge	3.2	
SDRAMC <sub>3</sub>	SDCKE Low before SDCK Rising Edge	7	
SDRAMC <sub>4</sub>	SDCKE High after SDCK Rising Edge	2.9	
SDRAMC <sub>5</sub>	SDCS Low before SDCK Rising Edge	7.5	
SDRAMC <sub>6</sub>	SDCS High after SDCK Rising Edge	1.6	
SDRAMC <sub>7</sub>	RAS Low before SDCK Rising Edge	7.2	
SDRAMC <sub>8</sub>	RAS High after SDCK Rising Edge	2.3	
SDRAMC <sub>9</sub>	SDA10 Change before SDCK Rising Edge	7.6	
SDRAMC <sub>10</sub>	SDA10 Change after SDCK Rising Edge	1.9	

### 3. SPI Bad Serial Clock Generation on 2nd chip\_select when SCBR = 1, CPOL=1 and NCPHA=0

When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an additional pulse will be generated on SCK.

#### Fix/workaround

When multiple CS are in use, if one of the baudrate equals 1, the other must also equal 1 if CPOL=1 and CPHA=0.

### 4. 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.

### 5. SPI Disable does not work in Slave mode

#### Fix/workaround

Read the last received data then perform a Software reset.

## 15.1.4 Power Manager

### 1. If the BOD level is higher than VDDCORE, the part is constantly under reset

If the BOD level is set to a value higher than VDDCORE and enabled by fuses, the part will be in constant reset.

#### Fix/Workaround

Apply an external voltage on VDDCORE that is higher than the BOD level and is lower than VDDCORE max and disable the BOD.

## 15.1.5 PDCA

### 1. 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.

## 15.1.6 TWI

### 1. 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.

## 15.1.7 USART

### 1. ISO7816 info register US\_NER cannot be read

The NER register always returns zero.

#### Fix/Workaround

None

## 15.1.8 Processor and Architecture

### 1. LDM instruction with PC in the register list and without ++ increments R<sub>p</sub>

When multiple CS are in use, if one of the baudrate equals 1, the other must also equal 1 if CPOL=1 and CPHA=0.

#### 4. **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.

#### 5. **SPI Disable does not work in Slave mode**

##### **Fix/workaround**

Read the last received data then perform a Software reset.

### 15.3.4 Power Manager

#### 1. **If the BOD level is higher than VDDCORE, the part is constantly under reset**

If the BOD level is set to a value higher than VDDCORE and enabled by fuses, the part will be in constant reset.

##### **Fix/Workaround**

Apply an external voltage on VDDCORE that is higher than the BOD level and is lower than VDDCORE max and disable the BOD.

### 15.3.5 Flashc

#### 1. **On AT32UC3A0512 and AT32UC3A1512, corrupted read in flash after FLASHC WP, EP, EA, WUP, EUP commands may happen**

- After a FLASHC Write Page (WP) or Erase Page (EP) command applied to a page in a given half of the flash (first or last 256 kB of flash), reading (data read or code fetch) the other half of the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

- After a FLASHC Erase All (EA) command, reading (data read or code fetch) the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

- After a FLASHC Write User Page (WUP) or Erase User Page (EUP) command, reading (data read or code fetch) the second half (last 256 kB) of the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

##### **Fix/Workaround**

Flashc WP, EP, EA, WUP, EUP commands: these commands must be issued from RAM or through the EBI. After these commands, read twice one flash page initialized to 00h in each half part of the flash.

### 15.3.6 PDCA

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

specific case.

2. Execute the RETE instruction.

## 4. **SPI Bad Serial Clock Generation on 2nd chip\_select when SCBR = 1, CPOL=1 and NCPHA=0**

When multiple CS are in use, if one of the baudrate equals to 1 and one of the others doesn't equal to 1, and CPOL=1 and CPHA=0, then an additional pulse will be generated on SCK.

### **Fix/workaround**

When multiple CS are in use, if one of the baudrate equals 1, the other must also equal 1 if CPOL=1 and CPHA=0.

## 5. **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.

## 6. **SPI Disable does not work in Slave mode**

### **Fix/workaround**

Read the last received data then perform a Software reset.

## 15.4.4 Power Manager

### 1. **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.

### 2. **If the BOD level is higher than VDDCORE, the part is constantly under reset**

If the BOD level is set to a value higher than VDDCORE and enabled by fuses, the part will be in constant reset.

### **Fix/Workaround**

Apply an external voltage on VDDCORE that is higher than the BOD level and is lower than VDDCORE max and disable the BOD.

## 15.4.5 FLASHC

### 1. **On AT32UC3A0512 and AT32UC3A1512, corrupted read in flash after FLASHC WP, EP, EA, WUP, EUP commands may happen**

- After a FLASHC Write Page (WP) or Erase Page (EP) command applied to a page in a given half of the flash (first or last 256 kB of flash), reading (data read or code fetch) the other half of the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

- After a FLASHC Erase All (EA) command, reading (data read or code fetch) the flash may fail. This may lead to an exception or to other errors derived from this corrupted read access.

- After a FLASHC Write User Page (WUP) or Erase User Page (EUP) command, reading

## Fix/workaround

When multiple CS are in use, if one of the baudrate equals 1, the other must also equal 1 if CPOL=1 and CPHA=0.

### 15.5.2 PWM

#### 1. PWM 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

#### 2. 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.

#### 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. PWM channel status may be wrong if disabled before a period has elapsed

Before a PWM period has elapsed, the read channel status may be wrong. The CHIDx-bit for a PWM channel in the PWM Enable Register will read '1' for one full PWM period even if the channel was disabled before the period elapsed. It will then read '0' as expected.

## Fix/Workaround

Reading the PWM channel status of a disabled channel is only correct after a PWM period has elapsed.

### 15.5.3 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 PIO.

#### 2. SSC Data is not sent unless clock is set as output

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 PIO.