

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

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	44
Program Memory Size	256КВ (256К х 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	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3b0256-a2ur

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Figure 4-2. TQFP48 / QFN48 Pinout



Note: The exposed pad is not connected to anything internally, but should be soldered to ground to increase board level reliability.

4.2 Peripheral Multiplexing on I/O lines

4.2.1 Multiplexed signals

Each GPIO line can be assigned to one of 4 peripheral functions; A, B, C or D (D is only available for UC3Bx512 parts). The following table define how the I/O lines on the peripherals A, B,C or D are multiplexed by the GPIO.

48-pin	64-pin	PIN	GPIO Pin	Function A	Function B	Function C	Function D (only for UC3Bx512)
3	3	PA00	GPIO 0				
4	4	PA01	GPIO 1				
5	5	PA02	GPIO 2				
7	9	PA03	GPIO 3	ADC - AD[0]	PM - GCLK[0]	USBB - USB_ID	ABDAC - DATA[0]
8	10	PA04	GPIO 4	ADC - AD[1]	PM - GCLK[1]	USBB - USB_VBOF	ABDAC - DATAN[0]
9	11	PA05	GPIO 5	EIC - EXTINT[0]	ADC - AD[2]	USART1 - DCD	ABDAC - DATA[1]

 Table 4-1.
 GPIO Controller Function Multiplexing



Table 5-1. Signal Description List (Continued)

Signal Name	Function	Туре	Active Level	Comments			
Serial Peripheral Interface - SPI0							
MISO	Master In Slave Out	I/O					
MOSI	Master Out Slave In	I/O					
NPCS0 - NPCS3	SPI Peripheral Chip Select	I/O	Low				
SCK	Clock	Output					
	Synchronous Serial Cor	ntroller - SS	С				
RX_CLOCK	SSC Receive Clock	I/O					
RX_DATA	SSC Receive Data	Input					
RX_FRAME_SYNC	SSC Receive Frame Sync	I/O					
TX_CLOCK	SSC Transmit Clock	I/O					
TX_DATA	SSC Transmit Data	Output					
TX_FRAME_SYNC	SSC Transmit Frame Sync	I/O					
	Timer/Counter - 1	IMER					
A0	Channel 0 Line A	I/O					
A1	Channel 1 Line A	I/O					
A2	Channel 2 Line A	I/O					
B0	Channel 0 Line B	I/O					
B1	Channel 1 Line B	I/O					
B2	Channel 2 Line B	I/O					
CLK0	Channel 0 External Clock Input	Input					
CLK1	Channel 1 External Clock Input	Input					
CLK2	Channel 2 External Clock Input	Input					
	Two-wire Interface	e - TWI					
SCL	Serial Clock	I/O					
SDA	Serial Data	I/O					
Univ	versal Synchronous Asynchronous Receiver Tr	ransmitter -	USART0, U	SART1, USART2			
CLK	Clock	I/O					
CTS	Clear To Send	Input					



Table 5-1. Signal Description List (Continued)

Signal Name	Function	Туре	Active Level	Comments
DCD	Data Carrier Detect			Only USART1
DSR	Data Set Ready			Only USART1
DTR	Data Terminal Ready			Only USART1
RI	Ring Indicator			Only USART1
RTS	Request To Send	Output		
RXD	Receive Data	Input		
TXD	Transmit Data	Output		
	Analog to Digital Conv	erter - ADC		
AD0 - AD7	Analog input pins	Analog input		
ADVREF	Analog positive reference voltage input	Analog input		2.6 to 3.6V
	Audio Bitstream DAC	- ABDAC		
DATA0 - DATA1	D/A Data out	Output		
DATAN0 - DATAN1	D/A Data inverted out	Output		
	Pulse Width Modulat	or - PWM		
PWM0 - PWM6	PWM Output Pins	Output		
	Universal Serial Bus De	vice - USBE	3	-
DDM	USB Device Port Data -	Analog		
DDP	USB Device Port Data +	Analog		
VBUS	USB VBUS Monitor and Embedded Host Negociation	Analog Input		
USBID	ID Pin of the USB Bus	Input		
USB_VBOF	USB VBUS On/off: bus power control port	output		

5.1 JTAG pins

TMS and TDI pins have pull-up resistors. TDO pin is an output, driven at up to VDDIO, and has no pull-up resistor. These 3 pins can be used as GPIO-pins. At reset state, these pins are in GPIO mode.

TCK pin cannot be used as GPIO pin. JTAG interface is enabled when TCK pin is tied low.



5.2 RESET_N pin

The RESET_N pin is a schmitt input and integrates a permanent pull-up resistor to VDDIO. As the product integrates a power-on reset cell, the RESET_N pin can be left unconnected in case no reset from the system needs to be applied to the product.

5.3 TWI pins

When these pins are used for TWI, the pins are open-drain outputs with slew-rate limitation and inputs with inputs with spike-filtering. When used as GPIO-pins or used for other peripherals, the pins have the same characteristics as GPIO pins.

5.4 GPIO pins

All the I/O lines integrate a pull-up resistor. Programming of this pull-up resistor is performed independently for each I/O line through the GPIO Controllers. After reset, I/O lines default as inputs with pull-up resistors disabled, except when indicated otherwise in the column "Reset Value" of the GPIO Controller user interface table.

5.5 High drive pins

The four pins PA20, PA21, PA22, PA23 have high drive output capabilities.

5.6 Power Considerations

5.6.1 Power Supplies

The AT32UC3B 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 and VDDPLL. The ground pin for VDDANA is GNDANA.

Refer to Electrical Characteristics section for power consumption on the various supply pins.

The main requirement for power supplies connection is to respect a star topology for all electrical connection.





Figure 6-1. Overview of the AVR32UC CPU

6.3.1 Pipeline Overview

AVR32UC has three pipeline stages, Instruction Fetch (IF), Instruction Decode (ID), and Instruction Execute (EX). The EX stage is split into three parallel subsections, one arithmetic/logic (ALU) section, one multiply (MUL) section, and one load/store (LS) section.

Instructions are issued and complete in order. Certain operations require several clock cycles to complete, and in this case, the instruction resides in the ID and EX stages for the required number of clock cycles. Since there is only three pipeline stages, no internal data forwarding is required, and no data dependencies can arise in the pipeline.

Figure 6-2 on page 20 shows an overview of the AVR32UC pipeline stages.





Figure 6-5. The Status Register Low Halfword

6.4.3 Processor States

6.4.3.1 Normal RISC State

The AVR32 processor supports several different execution contexts as shown in Table 6-2 on page 23.

Drierity	Mede	Security	Description
Priority	Mode	Security	Description
1	Non Maskable Interrupt	Privileged	Non Maskable high priority interrupt mode
2	Exception	Privileged	Execute exceptions
3	Interrupt 3	Privileged	General purpose interrupt mode
4	Interrupt 2	Privileged	General purpose interrupt mode
5	Interrupt 1	Privileged	General purpose interrupt mode
6	Interrupt 0	Privileged	General purpose interrupt mode
N/A	Supervisor	Privileged	Runs supervisor calls
N/A	Application	Unprivileged	Normal program execution mode

 Table 6-2.
 Overview of Execution Modes, their Priorities and Privilege Levels.

Mode changes can be made under software control, or can be caused by external interrupts or exception processing. A mode can be interrupted by a higher priority mode, but never by one with lower priority. Nested exceptions can be supported with a minimal software overhead.

When running an operating system on the AVR32, user processes will typically execute in the application mode. The programs executed in this mode are restricted from executing certain instructions. Furthermore, most system registers together with the upper halfword of the status register cannot be accessed. Protected memory areas are also not available. All other operating modes are privileged and are collectively called System Modes. They have full access to all privileged and unprivileged resources. After a reset, the processor will be in supervisor mode.

6.4.3.2 Debug State

The AVR32 can be set in a debug state, which allows implementation of software monitor routines that can read out and alter system information for use during application development. This implies that all system and application registers, including the status registers and program counters, are accessible in debug state. The privileged instructions are also available.



7. Memories

7.1 Embedded Memories

Internal High-Speed Flash

- 512KBytes (AT32UC3B0512, AT32UC3B1512)
- 256 KBytes (AT32UC3B0256, AT32UC3B1256)
- 128 KBytes (AT32UC3B0128, AT32UC3B1128)
- 64 KBytes (AT32UC3B064, AT32UC3B164)
 - 0 Wait State Access at up to 30 MHz in Worst Case Conditions
 - 1 Wait State Access at up to 60 MHz in Worst Case Conditions
 - - Pipelined Flash Architecture, allowing burst reads from sequential Flash locations, hiding penalty of 1 wait state access
 - 100 000 Write Cycles, 15-year Data Retention Capability
 - - 4 ms Page Programming Time, 8 ms Chip Erase Time
 - Sector Lock Capabilities, Bootloader Protection, Security Bit
 - - 32 Fuses, Erased During Chip Erase
 - User Page For Data To Be Preserved During Chip Erase
- Internal High-Speed SRAM, Single-cycle access at full speed
 - 96KBytes ((AT32UC3B0512, AT32UC3B1512)
 - 32KBytes (AT32UC3B0256, AT32UC3B0128, AT32UC3B1256 and AT32UC3B1128)
 - 16KBytes (AT32UC3B064 and AT32UC3B164)

7.2 Physical Memory Map

The system bus is implemented as a bus matrix. All system bus addresses are fixed, and they are never remapped in any way, not even in boot. Note that AVR32 UC CPU uses unsegmented translation, as described in the AVR32UC Technical Architecture Manual. The 32-bit physical address space is mapped as follows:

Table 7-1.	AT32UC3B Physical Memory Map
------------	------------------------------

Device		Embedded SRAM	Embedded Flash	USB Data	HSB-PB Bridge A	HSB-PB Bridge B
Start Address		0x0000_0000	0x8000_0000	0xD000_0000	0xFFFF_0000	0xFFFE_0000
Size	AT32UC3B0512 AT32UC3B1512	96 Kbytes	512 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes
	AT32UC3B0256 AT32UC3B1256	32 Kbytes	256 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes
	AT32UC3B0128 AT32UC3B1128	32 Kbytes	128 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes
	AT32UC3B064 AT32UC3B164	16 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes



8. Boot Sequence

This chapter summarizes the boot sequence of the AT32UC3B. The behaviour after power-up is controlled by the Power Manager. For specific details, refer to section Power Manager (PM).

8.1 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 recieves a clock with the same frequency as the internal RC Oscillator.

8.2 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.

When powering up the device, there may be a delay before the voltage has stabilized, depending on the rise time of the supply used. The CPU can start executing code as soon as the supply

is above the POR threshold, and before the supply is stable. Before switching to a high-speed clock source, the user should use the BOD to make sure the VDDCORE is above the minimum level.



9. Electrical Characteristics

9.1 Absolute Maximum Ratings*

Operating Temperature40°C to +85°C
Storage Temperature60°C to +150°C
Voltage on GPIO Pins with respect to Ground for TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA31
Voltage on GPIO Pins with respect to Ground except for TCK, RESET_N, PA03, PA04, PA05, PA06, PA07, PA08, PA11, PA12, PA18, PA19, PA28, PA29, PA30, PA310.3 to 5.5V
Maximum Operating Voltage (VDDCORE, VDDPLL) 1.95V
Maximum Operating Voltage (VDDIO,VDDIN,VDDANA) . 3.6V
Total DC Output Current on all I/O Pinfor 48-pin packagefor 64-pin package265 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.



9.5.1 Power Consumtion for Different Sleep Modes

Table 9-10.Power Consumption for Different Sleep Modes for AT32UC3B064, AT32UC3B0128, AT32UC3B0256,
AT32UC3B164, AT32UC3B1128, AT32UC3B1256

Mode	Conditions		Тур.	Unit
Active	 - CPU running a recursive Fibonacci Algorithm from f PLL0 at f MHz. - Voltage regulator is on. - XIN0: external clock. Xin1 Stopped. XIN32 stopped. 	0.3xf(MHz)+0.443	mA/MHz	
	 All peripheral clocks activated with a division by 8. GPIOs are inactive with internal pull-up, JTAG uncor up and Input pins are connected to GND 			
	Same conditions at 60 MHz		18.5	mA
Idla	See Active mode conditions		0.117xf(MHz)+0.28	mA/MHz
lule	Same conditions at 60 MHz	7.3	mA	
Frazan	See Active mode conditions	0.058xf(MHz)+0.115	mA/MHz	
FIOZEII	Same conditions at 60 MHz	3.6	mA	
Chandhu	See Active mode conditions	0.042xf(MHz)+0.115	mA/MHz	
Standby	Same conditions at 60 MHz	2.7	mA	
Stop	 CPU running in sleep mode XIN0, Xin1 and XIN32 are stopped. All peripheral clocks are desactived. GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND. 		37.8	μA
Deepstop	See Stop mode conditions		24.9	μA
Otatia		Voltage Regulator On	13.9	μA
Static	See Stop mode conditions Voltage Regulator Off		8.9	μA

Notes: 1. Core frequency is generated from XIN0 using the PLL so that 140 MHz < f_{PLL0} < 160 MHz and 10 MHz < f_{XIN0} < 12 MHz.

Table 9-11. Power Consumption for Different Sleep Modes for AT32UC3B0512, AT32UC3B1512

Mode	Conditions	Тур.	Unit
Active	 CPU running a recursive Fibonacci Algorithm from flash and clocked from PLL0 at f MHz. Voltage regulator is on. XIN0: external clock. Xin1 Stopped. XIN32 stopped. All peripheral clocks activated with a division by 8. GPIOs are inactive with internal pull-up, JTAG unconnected with external pull-up and Input pins are connected to GND 	0.359xf(MHz)+1.53	mA/MHz
	Same conditions at 60 MHz	24	mA
Lelle	See Active mode conditions	0.146xf(MHz)+0.291	mA/MHz
luie	Same conditions at 60 MHz	9	mA



9.8 ADC Characteristics

Parameter	Conditions	Min.	Тур.	Max.	Unit
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
Track and Hold Input Resistor			350		Ω
Track and Hold Capacitor			12		pF
	ADC Clock = 5 MHz			2	μs
Conversion Time	ADC Clock = 8 MHz			1.25	μs
Three when it Dete	ADC Clock = 5 MHz			384 ⁽¹⁾	kSPS
	ADC Clock = 8 MHz			533 ⁽²⁾	kSPS

Notes: 1. Corresponds to 13 clock cycles: 3 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

2. Corresponds to 15 clock cycles: 5 clock cycles for track and hold acquisition time and 10 clock cycles for conversion.

Table 9-21.	External Voltage Reference	Input
-------------	----------------------------	-------

Parameter	Conditions	Min.	Тур.	Max.	Unit
ADVREF Input Voltage Range	(1)	2.6		VDDANA	V
ADVREF Average Current	On 13 samples with ADC Clock = 5 MHz		200	250	μA
Current Consumption on VDDANA	On 13 samples with ADC Clock = 5 MHz			1	mA

Note: 1. ADVREF should be connected to GND to avoid extra consumption in case ADC is not used.

Table 9-22.Analog Inputs

Parameter	Conditions	Min.	Тур.	Max.	Unit
Input Voltage Range		0		V _{ADVREF}	V
Input Leakage Current				1	μA
Input Capacitance			7		pF

Table 9-23. Transfer Characteristics in 8-bit Mo
--

Parameter	Conditions	Min.	Тур.	Max.	Unit
Resolution			8		Bit
	ADC Clock = 5 MHz			0.8	LSB
Absolute Accuracy	ADC Clock = 8 MHz			1.5	LSB
	ADC Clock = 5 MHz		0.35	0.5	LSB
Integral Non-linearity	ADC Clock = 8 MHz		0.5	1.0	LSB



9.11 SPI Characteristics



Figure 9-7. SPI Master mode with (CPOL = NCPHA = 0) or (CPOL= NCPHA= 1)

Figure 9-8. SPI Master mode with (CPOL=0 and NCPHA=1) or (CPOL=1 and NCPHA=0)













Compliant JEDEC Standard MD-220 variation VMMD-3

Table 10-8.	Device and Package	Maximum Weight
	Borrioo ana r aonago	maximan rroigne

MSL3
)-

Table 10-10. Package Reference

JEDEC Drawing Reference	M0-220
JESD97 Classification	e3



- 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.

7. TC

8. 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.



12.2 AT32UC3B0256, AT32UC3B0128, AT32UC3B064, AT32UC3B1256, AT32UC3B1128, AT32UC3B164

All industrial parts labelled with -UES (for engineering samples) are revision B parts.

12.2.1 Rev I, J, K

- 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.



- Processor and Architecture

- LDM instruction with PC in the register list and without ++ increments Rp For LDM with PC in the register list: the instruction behaves as if the ++ field is always set, ie the pointer is always updated. This happens even if the ++ field is cleared. Specifically, the increment of the pointer is done in parallel with the testing of R12. Fix/Workaround None.
- 2. RETE instruction does not clear SREG[L] from interrupts The RETE instruction clears SREG[L] as expected from exceptions. Fix/Workaround
 When using the STCOND instruction, clear SREC[L] in the stacked value of

When using the STCOND instruction, clear SREG[L] in the stacked value of SR before returning from interrupts with RETE.

3. Privilege violation when using interrupts in application mode with protected system stack

If the system stack is protected by the MPU and an interrupt occurs in application mode, an MPU DTLB exception will occur.

Fix/Workaround

Make a DTLB Protection (Write) exception handler which permits the interrupt request to be handled in privileged mode.

- 4. USART
- ISO7816 info register US_NER cannot be read The NER register always returns zero.
 Fix/Workaround None.
- ISO7816 Mode T1: RX impossible after any TX RX impossible after any TX.
 Fix/Workaround SOFT_RESET on RX+ Config US_MR + Config_US_CR.

7. The RTS output does not function correctly in hardware handshaking mode

The RTS signal is not generated properly when the USART receives data in hardware handshaking mode. When the Peripheral DMA receive buffer becomes full, the RTS output should go high, but it will stay low.

Fix/Workaround

Do not use the hardware handshaking mode of the USART. If it is necessary to drive the RTS output high when the Peripheral DMA receive buffer becomes full, use the normal mode of the USART. Configure the Peripheral DMA Controller to signal an interrupt when the receive buffer is full. In the interrupt handler code, write a one to the RTSDIS bit in the USART Control Register (CR). This will drive the RTS output high. After the next DMA transfer is started and a receive buffer is available, write a one to the RTSEN bit in the USART CR so that RTS will be driven low.

8. Corruption after receiving too many bits in SPI slave mode

If the USART is in SPI slave mode and receives too much data bits (ex: 9bitsinstead of 8 bits) by the SPI master, an error occurs. After that, the next reception may be corrupted



12.2.4 Rev. B

- 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. 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.

5. The following alternate C functions PWM[4] on PA16 and PWM[6] on PA31 are not available on Rev B

The following alternate C functions PWM[4] on PA16 and PWM[6] on PA31 are not available on Rev B.

Fix/Workaround

Do not use these PWM alternate functions on these pins.

6. SPI

7. 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.



- 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 been 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
 - 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
```

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.

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.

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.



- OCD

1. Stalled memory access instruction writeback fails if followed by a HW breakpoint Consider the following assembly code sequence:

A B

If a hardware breakpoint is placed on instruction B, and instruction A is a memory access instruction, register file updates from instruction A can be discarded.

Fix/Workaround

Do not place hardware breakpoints, use software breakpoints instead. Alternatively, place a hardware breakpoint on the instruction before the memory access instruction and then single step over the memory access instruction.

- Processor and Architecture

 Local Bus to fast GPIO not available on silicon Rev B Local bus is only available for silicon RevE and later. Fix/Workaround
 Do not use if silicon revision older than E

Do not use if silicon revision older than F.

2. Memory Protection Unit (MPU) is non functional Memory Protection Unit (MPU) is non functional. Fix/Workaround

Do not use the MPU.

3. Bus error should be masked in Debug mode

If a bus error occurs during debug mode, the processor will not respond to debug commands through the DINST register.

Fix/Workaround

A reset of the device will make the CPU respond to debug commands again.

4. Read Modify Write (RMW) instructions on data outside the internal RAM does not work

Read Modify Write (RMW) instructions on data outside the internal RAM does not work. **Fix/Workaround**

Do not perform RMW instructions on data outside the internal RAM.

5. Need two NOPs instruction after instructions masking interrupts

The instructions following in the pipeline the instruction masking the interrupt through SR may behave abnormally.

Fix/Workaround

Place two NOPs instructions after each SSRF or MTSR instruction setting IxM or GM in SR

6. Clock connection table on Rev B

Here is the table of Rev B



13.12 Rev. A - 05/2007

1. Initial revision.

