
AVR32747: Compatibility Note AT32UC3Bx Revision B to Revision F or later

1 Introduction

The AT32UC3Bx revision F introduces fixes and incompatibility with the previous revision B. This document outlines the software operations to migrate software from an AT32UC3Bx revision B, also known as engineering sample (ES), to AT32UC3Bx revision F and later.

The AVR[®]32Studio, the AVR32 GNU Toolchain and the Software Framework have also been upgraded. The 2nd part of this document also outlines the software operations necessary to migrate a software project from older to newer tools.



32-bit **AVR[®]**
Microcontrollers

Application Note

Rev. 32108B-AVR32-08/08





2 Parts Concerned

Table 2-1. List of Concerned Parts

AT32UC3Bx Engineering Samples Rev B	AT32UC3Bx Production parts Rev F and later
AT32UC3B0256 Rev B AT32UC3B1256 Rev B	AT32UC3B0256 Rev F and later AT32UC3B1256 Rev F and later AT32UC3B0128 Rev F and later AT32UC3B1128 Rev F and later AT32UC3B064 Rev F and later AT32UC3B164 Rev F and later

2.1 How to differentiate Device Revisions

2.1.1 Package Label

- The packages of engineering samples are labeled with “-xES”. 32UC3Bxxx-UES means it is a Rev B.
- The packages of industrial production parts (non engineering samples) are labeled with “-U” (used for production of industrial parts).

In this document, we will use the “ES” notation to designate engineering sample parts (ie. Rev B).

2.1.2 Device Identification Register

The JTAG Device Identification register (DID) contains the device chip revision (“RN” field, revA = 0x0 revB = 0x01 etc...).

Note: Refer to device datasheet section JTAG and Boundary Scan for the JTAGID code description.

This is the C code to read the JTAG DID register in software, for IAR or GCC compiler:

For IAR compiler:

```
#include <avr32/io.h>
#include <intrinsics.h>
static inline unsigned int read_jtag_id(void) {
    return __get_debug_register(AVR32_DID);
}
```

For GCC compiler:

```
#include <avr32/io.h>
static inline unsigned int read_jtag_id(void) {
    return __builtin_mfdr(AVR32_DID);
}
```

Table 2-2. JTAG DID Register Description

R/W	Bit Number	Field Name	Init. Val.	Description
R	31:28	RN	Part specific	RN – Revision Number
R	27:12	PN	Part specific	PN – Product Number
R	11:1	MID	0x01F	Manufacturer ID 0x01F = ATMEL
R	0	Reserved	1	Reserved This bit always reads as 1.

2.1.3 Using the JTAGICEmkII or the AVR One!

Avr32program is a utility from the AVR32 GNU toolchain which can read the JTAG ID revision from a command line window (“avr32program cpuinfo”, read the “JTAG Revision” line).

Figure 2-3. JTAG Revision

```

C:\WINDOWS\system32\cmd.exe
C:\>avr32program cpuinfo
Connected to AVR ONE! 0000000015D version 1.1.
Querying file system information.
FPGA file avr32 version 1.1 loaded.
JTAG clock is configured at 40.698 MHz.

CPU information:
PartName                UC3B0256
Processor Revision      3
JTAG Revision           5
Architecture Type       0 <AVR32A>
Architecture Revision   1
Memory Management Unit Type 3 <Memory Protection Unit>
Number of Entries in the IMMU 1
Number of Entries in the DMMU 9
Floating Point Unit     No
Java Extension          No
Performance Counters    No
OnChip Debug            Yes
SIMD Instructions       No
DSP Instructions         Yes
Memory R-M-W Instructions Yes

Number of Sets in the Instruction Cache 1
Instruction Cache Line Size 1 bytes
Instruction Cache Associativity Direct Mapping
Number of Sets in the Data Cache 1
Data Cache Line Size 1 bytes
Data Cache Associativity Direct Mapping

C:\>
  
```

JTAG revision: Rev B = 0x1, revF = 0x5





3 Software Migration

The errata section in the device datasheet provides the list of known issues per revision. This migration note does not address every item of the datasheet errata. Only the items affecting software binary compatibility are addressed in the following.

3.1 CPU Cycle Counter

On Rev B:

1. The CPU Cycle Counter does not reset the COUNT system register on COMPARE match.
2. The COUNT register is clocked by the CPU clock, so when the CPU clock stops, so does incrementing of COUNT.

On Rev F:

1. The CPU cycle counter resets the COUNT system register on COMPARE match.
2. The COUNT register has a dedicated clock active only in active mode and in the following sleep modes: IDLE, FROZEN and STANDBY.

Migration from Rev B to Rev F or later:

1. Any software which relied on the fact that the COUNT system register was not reset on COMPARE match must be updated to reflect the fact that the COUNT system register is now reset on COMPARE match.
2. Any software which relied on the fact that COUNT register was not incremented in any of the sleep modes must be updated to reflect the fact that the COUNT register is now incremented in the following sleep modes: IDLE, FROZEN and STANDBY.

Software implementation examples can be found in the AVR32 UC3 software framework under the directory DRIVERS/CPU/CYCLE_COUNTER/EXAMPLE.

3.2 Timer Counter

On Rev B, the following timer counter clock inputs are different than in Rev F and later.

On Rev B:

- TIMER_CLOCK2 is connected to PBA Clock / 4
- TIMER_CLOCK4 is connected to PBA Clock / 16
- TIMER_CLOCK5 is connected to PBA Clock / 32

On Rev F:

- TIMER_CLOCK2 is connected to PBA Clock / 2
- TIMER_CLOCK4 is connected to PBA Clock / 32
- TIMER_CLOCK5 is connected to PBA Clock / 128

Migration from Rev B to Rev F or later:

Any software that was relying on the previous timer clock input frequency must be updated to match the new timer clock input frequency.

Software implementation examples can be found in the AVR32 UC3 software framework under the directory DRIVERS/TC.

3.3 Flashc

3.3.1 General Purpose Fuse Register Low Address

On Rev B the address of the Flash General Purpose Fuse Register Low (FGPFRLO, previously named FPGPFR) was 0xFFFE140C.

On Rev F the address of the Flash General Purpose Fuse Register Low (FGPFRLO) is 0xFFFE1410.

Migration from Rev B to Rev F or later:

Any software that hard coded the FGPFRLO address value must be updated with the new FGPFRLO address. If the software was relying on the flash header file definition, FPGPFR should be renamed to FGPFRLO.

3.3.2 PAGEN Semantic Field for Program GP Fuse Byte

On Rev B, the PAGEN semantic field for Program GP Fuse Byte was WriteData[7:0], ByteAddress[1:0].

On Rev F, the PAGEN semantic field for Program GP Fuse Byte is WriteData[7:0], ByteAddress[2:0].

Migration from Rev B to Rev F or later:

Any software that relied on the PAGEN Rev B semantic field must be updated with the Rev F semantic field.

Software implementation examples can be found in the AVR32 UC3 software framework under the directory DRIVERS/FLASHC.

3.4 USB

On Rev B, the UHADDR1/2/3 registers are not available. UHCON.HADDR (UHCON[6:0]) field register is used for the USB device address whatever the pipe. On Rev F, the HCON.HADDR (UHCON[6:0]) field register is not available. UHADDR1/2/3 registers are used to select one USB device address per pipe.

Migration from Rev B to Rev F or later:

Any software that relied on the UHCON.HADDR (UHCON[6:0]) field register must use the UHADDR1/2/3.UHADDR_Px instead: fill all the UHADDRx.UHADDR_Py with the value that was in the corresponding UHCON.HADDR.





Software implementation examples can be found in the AVR32 UC3 software framework under the directory DRIVERS/USBB.

3.5 Real Time Counter (RTC)

On Rev B, the RTC CLKEN bit (bit number 16) of CTRL register is not available.

On Rev F, the RTC CLKEN bit of CTRL register is available and has to be set in order to enable the RTC.

Migration from Rev B to Rev F or later:

Any software that used the RTC on Rev B must now use in Rev F the CLKEN bit of RTC CTRL register to enable the RTC.

Software implementation examples can be found in the AVR32 UC3 software framework under the directory DRIVERS/RTC.

4 Tools

4.1 AVR32Studio and GNU Toolchain 2.0 and later

The device header files set is different for ES parts compared with non-ES parts. To manage it in the GNU toolchain, the following targets are supported:

4.1.1 UC3B Part Name

With GCC compiler, the 'mpart' option should be set to one of the following:

- UC3B0256ES: Rev B.
- UC3B1256ES: Rev B.
- UC3B0256: Rev F and later.
- UC3B1256: Rev F and later.
- UC3B0128: Rev F and later.
- UC3B1128 Rev F and later.
- UC3B064: Rev F and later.
- UC3B164: Rev F and later.

4.1.2 Header Files Organization

The header files addressed in this section can be found for Windows users under the directory: C:\Program Files\Atmel\AVR Tools\AVR32 Toolchain\avr32\include\avr32. They are installed within the GNU Toolchain release.

- avr32/io.h file: this file is the entry point and is used to include the appropriate other header files according to the device chosen. This file is found in the GNU toolchain.

```
<snip>
#  elif __AVR32_UC3B0256ES__
#    include <avr32/uc3b0256es.h>
#  elif __AVR32_UC3B1256ES__
#    include <avr32/uc3b1256es.h>
<snip>
#  elif __AVR32_UC3B0256__
#    include <avr32/uc3b0256.h>
<snip>
```

- Device specific files: avr32/uc3b0256.h, avr32/uc3b0256es.h ... included by io.h according to the chosen device.
- IP specific header files: avr32/usart_319.h, avr32/usart_400.h... included by the 'device'.h specific header file.





4.1.3 Project Migration from AVR32 GNU Toolchain 1.y.z to AVR32 GNU Toolchain 2.0 or later

In AVR32 GCC toolchain version 2.0 and later, some headers files definitions have been renamed. When migrating any software from older toolchain version 1.y.z to toolchain 2.0 and later, the following definitions must be updated:

Table 4-1. Project Migration from AVR32 GNU Toolchain 1.y.z to AVR32 GNU Toolchain 2.0 or later

Old Definition in Toolchain version 1.y.z	New definition in Toolchain version 2.0 or later
Interrupt Controller header file: AVR32_INTIC_IPRx_INTLEV_OFFSET AVR32_INTIC_IPRx_INTLEV_MASK AVR32_INTIC_IPRx_INTLEV_SIZE AVR32_INTIC.icr3	AVR32_INTIC_IPR_INTLEV_OFFSET AVR32_INTIC_IPR_INTLEV_MASK AVR32_INTIC_IPR_INTLEV_SIZE AVR32_INTIC.icr
Flashc header file: AVR32_FLASHC_FGPFR_GPFx AVR32_FLASHC_FGPFR_GPFx_MASK AVR32_FLASHC_FGPFR_GPFx_SIZE AVR32_FLASHC_FGPFR_GPFx_OFFSET	AVR32_FLASHC_FGPFR_LOCKx AVR32_FLASHC_FGPFR_LOCKx_MASK AVR32_FLASHC_FGPFR_LOCKx_SIZE AVR32_FLASHC_FGPFR_LOCKx_OFFSET
Flashc header file: AVR32_FLASHC_FGPFR_GPF00_MASK	AVR32_FLASHC_FGPFR_LOCK0_MASK
PWM header file: #define AVR32_PWM_PWM_LINES_MSB 6 #define AVR32_PWM_PWM_0_0_PIN 7 #define AVR32_PWM_PWM_0_0_FUNCTION 0 #define AVR32_PWM_PWM_1_0_PIN 8 #define AVR32_PWM_PWM_1_0_FUNCTION 0 #define AVR32_PWM_PWM_2_0_PIN 21 #define AVR32_PWM_PWM_2_0_FUNCTION 0 #define AVR32_PWM_PWM_6_0_PIN 22 #define AVR32_PWM_PWM_6_0_FUNCTION 0 #define AVR32_PWM_PWM_2_1_PIN 13 #define AVR32_PWM_PWM_2_1_FUNCTION 1 #define AVR32_PWM_PWM_3_0_PIN 14 #define AVR32_PWM_PWM_3_0_FUNCTION 1 #define AVR32_PWM_PWM_4_0_PIN 15 #define AVR32_PWM_PWM_4_0_FUNCTION 1 #define AVR32_PWM_PWM_5_0_PIN 18 #define AVR32_PWM_PWM_5_0_FUNCTION 1 #define AVR32_PWM_PWM_6_1_PIN 19 #define AVR32_PWM_PWM_6_1_FUNCTION 1 #define AVR32_PWM_PWM_3_1_PIN 25 #define AVR32_PWM_PWM_3_1_FUNCTION 1 #define AVR32_PWM_PWM_4_1_PIN 28 #define AVR32_PWM_PWM_4_1_FUNCTION 1 #define AVR32_PWM_PWM_0_1_PIN 11 #define AVR32_PWM_PWM_0_1_FUNCTION 2 #define AVR32_PWM_PWM_1_1_PIN 12	#define AVR32_PWM_LINES_MSB 6 #define AVR32_PWM_0_0_PIN 7 #define AVR32_PWM_0_0_FUNCTION 0 #define AVR32_PWM_1_0_PIN 8 #define AVR32_PWM_1_0_FUNCTION 0 #define AVR32_PWM_2_0_PIN 21 #define AVR32_PWM_2_0_FUNCTION 0 #define AVR32_PWM_6_0_PIN 22 #define AVR32_PWM_6_0_FUNCTION 0 #define AVR32_PWM_2_1_PIN 13 #define AVR32_PWM_2_1_FUNCTION 1 #define AVR32_PWM_3_0_PIN 14 #define AVR32_PWM_3_0_FUNCTION 1 #define AVR32_PWM_4_0_PIN 15 #define AVR32_PWM_4_0_FUNCTION 1 #define AVR32_PWM_5_0_PIN 18 #define AVR32_PWM_5_0_FUNCTION 1 #define AVR32_PWM_6_1_PIN 19 #define AVR32_PWM_6_1_FUNCTION 1 #define AVR32_PWM_3_1_PIN 25 #define AVR32_PWM_3_1_FUNCTION 1 #define AVR32_PWM_4_1_PIN 28 #define AVR32_PWM_4_1_FUNCTION 1 #define AVR32_PWM_0_1_PIN 11 #define AVR32_PWM_0_1_FUNCTION 2 #define AVR32_PWM_1_1_PIN 12

#define AVR32_PWM_PWM_1_1_FUNCTION	2	#define AVR32_PWM_PWM_1_1_FUNCTION	2
#define AVR32_PWM_PWM_5_1_PIN	37	#define AVR32_PWM_PWM_5_1_PIN	37
#define AVR32_PWM_PWM_5_1_FUNCTION	2	#define AVR32_PWM_PWM_5_1_FUNCTION	2
Power Manager header file:			
AVR32_PM_GCLK_USB		AVR32_PM_GCLK_USBB	
RTC:			
AVR32_RTC_RTC_IRQ		AVR32_RTC_IRQ	

Note 1: The UC3 software framework version 1.1.1 is implemented with definitions from the left column.

Note 2: The UC3 software framework version 1.2.z and 1.3.z or later are implemented with definition from the right column.

Project Migration from Toolchain 1.y.z to Toolchain 2.0 or later:

Any software that relied on the Toolchain 1.y.z must be updated with the header definition from the Toolchain 2.0 or later version.

4.1.4 Project Migration from UC3Bx256ES Target (Rev B) to UC3Bx256 Target (Rev F or later) in Toolchain 2.x

To migrate a project from UC3B0256ES (Rev B) to UC3B0256 (Rev F or later), the following defines must be updated:

Table 4-2. Project Migration from UC3Bx256ES Target (Rev B) to UC3Bx256 Target (Rev F or later) in Toolchain 2.x

Old Definition in Toolchain version 1.y.z	New definition in Toolchain version 2.0 or later
Interrupt Controller header file: AVR32_INTC_IPRx_INTLEV_OFFSET AVR32_INTC_IPRx_INTLEV_MASK	AVR32_INTC_IPR_INTLEV_OFFSET AVR32_INTC_IPR_INTLEV_MASK
Flashc header file: AVR32_FLASHC_FGPFR_x	AVR32_FLASHC_FGPFRLO_x

To migrate a project from UC3Bx256ES (Rev B) target to UC3Bx256 (Rev F or later), the definitions described above must be updated.





4.2 AVR32 UC3 Software Framework

4.2.1 Overview

Table 4-3. Supported Frameworks per Parts

UC3Bx Rev B		UC3Bx Rev F or later	
Toolchain 1.y.z	Toolchain 2.0 or later	Toolchain 1.y.z	Toolchain 2.0 or later
Software framework 1.1.1	Software framework 1.2.0ES	Not Available	Software framework 1.3.0

4.2.2 Version 1.1.1-AT32UC3B

The AVR32 software framework version 1.1.1 was dedicated to ES parts.

All drivers provided in this framework are implemented for the Rev B parts, and include workarounds for the Rev B errata.

This framework only supports the Toolchain version 1.y.z.

Since the Toolchain 2.0 or later introduced header files definition changes, the software framework 1.1.1 does not straight away compile with these toolchains.

4.2.3 Version 1.2.zES-AT32UC3B

The AVR32 software framework version 1.2.zES is dedicated to ES parts. All drivers provided in this framework are implemented for the Rev B parts, and include workarounds for the Rev B errata.

This framework only supports the Toolchain version 2.0 or later.

4.2.4 Version 1.3.z-AT32UC3B

The AVR32 software frameworks 1.3.z and later versions support UC3B Rev F and later revisions. It does not support the Rev B version (ES).

All drivers provided in this framework are implemented for the Rev F parts, and include workarounds for the Rev F errata.

This framework only supports the Toolchain version 2.0 or later.

4.3 USB DFU Bootloader

The pre-programmed Atmel USB DFU Bootloader implementation is different for Rev B and Rev F, the interface remains the same.

The bootloader binary and its programming script can be found:

- On Rev B, in software framework 1.2.zES :
\\SERVICES\USB\CLASS\DFU\EXAMPLES\ISP\AT32UC3BES\Releases\
- On Rev F and later, in software framework 1.3.z:
\\SERVICES\USB\CLASS\DFU\EXAMPLES\ISP\AT32UC3B\Releases\

5 Suggested reading

5.1 Device datasheet

The device datasheet contains block diagrams of the peripherals and details about implementing firmware for the device. The datasheet is available on <http://www.atmel.com/AVR32> in the *Datasheets* section.

5.2 AVR32 Software Framework

This framework provides software drivers and libraries to build any application for AVR32 UC3 devices. It is available on <http://www.atmel.com/AVR32> in the *Tools & Software* section.





Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® is a registered trademark of Microsoft Corporation in the US and/or other countries. Other terms and product names may be trademarks of others.