# 8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

# ATmega103(L) Rev. L Errata Sheet

## Errata

- **Power Consumption when Using Slowly Rising Power Supply**
- **Releasing Reset Condition without Clock**
- **Wake-up from Power-save Executes Instructions before Interrupt**
- **SPI Can Send Wrong Byte**
- **Wrong Clearing of EXTRF in MCUSR**
- **Reset during EEPROM Write**
- **SPI Interrupt Flag Can be Undefined after Reset**
- **Skip Instruction with Interrupts**
- **Signature Bytes**
- **Read Back Value during EEPROM Polling**
- **MISO Active during In-System Programming**
- **The ADC Has no Free-running Mode**
- **UART Loses Synchronization if RXD Line is Low when UART Receive is Disabled**

### 13. Power Consumption when Using Slowly Rising Power Supply

If the power supply rises slowly upon start-up (slower than 10 ms rise time), the power consumption in sleep modes may exceed the specification.

**Problem Fix/Workaround**

The device behaves functionally correct, and no actions need to be taken if power consumption is acceptable. To reduce power consumption, make sure the power supply has a sufficiently fast rise time.

### 12. Releasing Reset Condition without Clock

If an external reset or a watchdog reset occurs while the clock is stopped and the reset is released before the clock is restarted, the internal reset will time-out after the start-up delay which is independent of the external clock. If no external clock pulses are present in the period when internal reset is active, the reset does correctly cause tri-stating of the I/O while the reset is held. However, if the internal reset is released before the clock starts running, the part does not clear I/O registers, nor set PC to 0x00. Here, stopping the clock refers to gating the external clock input. Power-down and Power-save mode do not have this issue.

**Problem Fix/Workaround**

Make sure the clock is running whenever an external reset can be expected. If the watchdog is used, never stop an external clock.

### 11. Wake-up from Power-save Executes Instructions before Interrupt

When waking up from Power-save, some instructions are executed before the interrupt is called. If the device is woken up by an external interrupt, two instruction cycles are executed. If it is woken up by the asynchronous timer, three instructions are executed before the interrupt.

**Problem Fix/Workaround**

Make sure that the first two or three instructions following sleep are not dependent on the executed interrupt.

### 10. The SPI Can Send Wrong Byte

If the SPI is in Master mode, it will restart the old transfer if new data is written on the same clock edge as the previous transfer is finished.

**Problem Fix/Workaround**

When writing to the SPI, first wait until it is ready, then write the byte to transmit.

Rev. 1436C–09/01

### 9. Wrong Clearing of EXTRF in MCUSR

The EXTRF flag in MCUSR will be cleared when clearing the PORF-flag. The flag does not get cleared by writing a “0” to it.

**Problem Fix/Workaround**

Finish the test of both flags before clearing any of them. Clear both flags simultaneously by writing “0” to both PORF and EXTRF in MCUCR.

### 8. Reset during EEPROM Write

If reset is activated during EEPROM write, the result is not what should be expected. The EEPROM write cycle completes as normal, but the address registers are reset to 0. The result is that both the address written and address 0 in the EEPROM can be corrupted.

**Problem Fix/Workaround**

Avoid using address 0 for storage, unless you can guarantee that you will not get a reset during EEPROM write.

### 7. SPI Interrupt Flag Can be Undefined after Reset

In certain cases when there are transitions on the SCK pin during reset, or the SCK pin is left unconnected, the start-up value of the SPI interrupt flag is unknown. If the flag is not reset before enabling the SPI interrupt, a pending SPI interrupt may be executed.

**Problem Fix/Workaround**

Clear the SPI interrupt flag before enabling the interrupt.

### 6. Skip Instruction with Interrupts

A skip instruction (SBRS, SBRC, SBIS, SBIC, CPSE) that skips a two-word instruction needs three clock cycles. If an interrupt occurs during the first or second clock cycle of this skip instruction, the return address will not be stored correctly on the stack. In this situation, the address of the second word in the two-word instruction is stored. This means that on return from interrupt, the second word of the two-word command will be decoded and executed as an instruction. The ATmega103 has four two-word instructions: LDS, STS, JMP, and CALL.

**Notes:**

1. This can only occur if all of the following conditions are true:
   - A skip instruction is followed by a two-word instruction.
   - The skip instruction is actually skipping the two-word instruction.
   - Interrupts are enabled, and at least one interrupt source can generate an interrupt.
   - An interrupt arrives in the first or second cycle of the skip instruction.
2. This will only cause problems if the address of the following LDS or STS command points to an address beyond 400 Hex.

**Problem Fix/Workaround**

For assembly program, avoid skipping a two-word instruction if interrupts are enabled.

The following C-compilers handles this sequence correctly:

- IAR Compiler, version 1.40b or higher
- Image Craft compiler, all versions
- Codevision Compiler, version 1.0.0.5 or higher

### 5. Signature Bytes

The signature bytes of the first few lots of the ATmega103 have been shipped with wrong signature bytes. Also in the datasheet, the wrong signature bytes have been given. The correct signature bytes are: $1E $97 $01.

**Problem Fix/Workaround**

Programmers must allow both $1E $97 $01 and $1E $01 $01 as valid signature bytes.

### 4. Read Back Value during EEPROM Polling

When a new EEPROM byte is being programmed into the EEPROM with In-System Programming, reading the address location being programmed will give the value P1 (see table below) until the Auto-erase is finished. Then the value P2 will follow until programming is finished. At the time the device is ready for a new EEPROM byte, the programmed value will read correctly.

| Revision | P1 | P2 |
|---|---|---|
| F | $7F | $7F |
| G | $80 | $7F |
| J | $80 | $7F |
| K | $80 | $7F |
| L | $80 | $7F |

**Note:** This is only a problem for In-System Programmers. Reading and writing the EEPROM during normal operation is not affected by this.

#### Problem Fix/Workaround

Programmers must allow both $80 and $7F as read back values if data polling is used for the EEPROM. Polling will not work for either of the values P1 and P2, so when programming these values, the user will have to wait the prescribed time $t_{WD\_EEPROM}$ before programming the next byte.

### 3. MISO Active during In-System Programming

During In-System Programming, the MISO line (pin 13) of the ATmega103 is active, although the UART pins are used for programming. If pin 13 is used as an input in the application, a collision may occur on this line.

#### Problem Fix/Workaround

- If the MISO pin is used as an input, make sure that there is a current-limiting resistor in series with the line.
- If the pin is used as an output, make sure that whatever is connected to the line can accept that the pin is toggling during programming.

### 2. The ADC Has no Free-running Mode

Early versions of the ATmega603/103 datasheet described an ADC Free-running mode. This mode is not available in this device, and bit number 5 in the ADCSR register must always be written as “0”.

#### Problem Fix/Workaround

Use Single-conversion mode and always use the latest revision of the datasheet.

### 1. UART Loses Synchronization if RXD Line is Low when UART Receive is Disabled

The UART will detect a UART start bit and start reception even if the UART is not enabled. If this occurs, the first byte after re-enabling the UART will be corrupted.

#### Problem Fix/Workaround

Make sure that the RX line is high at start-up and when the UART is disabled. An external RS-232 level converter keeps the line high during start-up.

## Atmel Headquarters

*Corporate Headquarters*
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

*Europe*
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

*Asia*
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

*Japan*
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

## Atmel Product Operations

*Atmel Colorado Springs*
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

*Atmel Grenoble*
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

*Atmel Heilbronn*
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

*Atmel Nantes*
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

*Atmel Rousset*
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

*Atmel Smart Card ICs*
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

*e-mail*
literature@atmel.com

*Web Site*
http://www.atmel.com

*BBS*
1-(408) 436-4309

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.