Errata

- **Verifying the EEPROM at High Voltages During Programming (ATmega103)**
- **Wake-up from Power Save Executes Instructions Before Interrupt**
- **SPI can Send Wrong Byte**
- **Wrong Clearing of XTRF in MCUSR**
- **Reset During EEPROM Write**
- **SPI Interrupt Flag can be Undefined After Reset**
- **Verifying EEPROM In-System**
- **Serial Programming at Voltages Below 3.4 Volts (ATmega103L)**
- **Skip Instruction with Interrupts**
- **Signature Bytes**
- **Read Back Value during EEPROM polling**
- **MISO Output during In-System Programming**
- **The ADC has no Free-Running Mode**

Note: ATmega103L operating range: 2.7V - 3.6V.
ATmega103 operating range: 4.0V - 5.5V

13. Verifying the EEPROM at High Voltages During Programming

Reading the EEPROM in the serial and the parallel programming modes is not guaranteed for supply voltages above 5.1V. This applies for the programming procedures only. Thus, for normal operation, including EEPROM access from the AVR itself, the ATmega103 device works in its full voltage range from 4.0V to 5.5V.

Problem Fix/Workaround

The programming voltage should be controlled not to exceed the 5.1V limit. If the programming voltage is above 5.1V, the verification of the contents of the EEPROM may fail. Alternatively, the timing specifications and voltage range for the ATmega103L device can be used for the ATmega103 during programming

Note: In the past, the manufacturing test did not screen this programming issue properly. For devices marked with ‘5V’ on the top side of the device, the programming is guaranteed functional up to 5.1V.

12. 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, 2 instruction cycles are executed. If it is woken up by the asynchronous timer, 3 instructions are executed before the interrupt.

Problem Fix/Workaround

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

11. 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.

8-bit AVR® Microcontroller with 64K/128K Bytes In-System Programmable Flash

ATmega103L Rev. F/G, ATmega103 Rev. G Errata Sheet

Rev. 1197B–02/99

10. Wrong Clearing of XTRF in MCUSR

The XTRF 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 XTRF in MCUCR.

9. 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.

8. 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 be 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.

7. Verifying EEPROM In-System

EEPROM verify In-System Programming mode cannot operate with maximum clock frequency. This is independent of the SPI clock frequency.

Problem Fix/Workaround

Reduce the clock speed, or avoid using the EEPROM verify feature.

6. Serial Programming at Voltages Below 3.4 Volts

At voltages below 3.4 Volts, serial programming might fail.

Note: Applies only to ATmega103L.

Problem Fix/Workaround

Keep V_{CC} above 3.4 Volts during in-system programming.

5. 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 C-programs, use the IAR compiler version 1.40b or later. The compiler will never generate the sequence.

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

4. Signature Bytes

The signature bytes of the first few lots of the ATmega103/L have been shipped with wrong signature bytes. Also in the datasheet, the wrong signature bytes has 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.

3. 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 1) 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.

Table 1. Address Location

Revision	P1	P2
F	$7F	$7F
G	$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 neither 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.

2. MISO Active during In-System Programming

During In-System Programming, the Miso line of the ATmega103 is active, even if the UART pins are used for programming. If the pin 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 a 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

1. The ADC has no Free-Running Mode

Early versions of the ATmega603/103(L) data sheet 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

Always use Single-Conversion Mode, and please use the latest revision of the data sheet.