Errata

- **Some instructions are not interruptible when they are followed by a two-word instruction**
- **Interrupt return fails when stack pointer addresses the external memory**
- **Writing UBBRH affects both UART0 and UART1**
- **Store Program Memory instruction may fail**

1. Some instructions are not interruptible when they are followed by a two-word instruction

This is the case for the following instructions:

- all one-word multi-cycle instructions except skip (adiw, sbiw, rjmp, rcall, ijmp, icall, ret, reti, mul, branch that branches, ld, st, push, pop, lpm, elpm, spm, sbi, cbi)

- all two-word instructions where the second word is the opcode of a two-word instruction

- all skips that skip the following two-word instruction, and where the second word of the following two-word instruction is the opcode of a two-word instruction

In most cases it will only increase the interrupt latency with a few cycles. However, in assembly code constructs where the program waits for an interrupt the bug gives a deadlock:

```
wait:   rjmp wait
<any two-word instruction>
```

There is only one way to make the C-compiler generate the rjmp -1 instruction, and that is by the following construction:

```
for(;;)
;
```

This construction will make the compiler delete the rest of the function (unreachable code). If the function at link time is linked next to a module starting with the JMP instruction, then you have the hazard.

Problem fix/Workaround

Work-around in assembly: insert nop before the two-word instruction

For C programmers: The workaround is to use one of the following constructions instead:

```
while(1)
;
```

or

```
do
{
}
while(1);
```

2. Interrupt return fails when stack pointer addresses the external memory

When stack pointer addresses external memory (SPH:SPL > $45F), returning from interrupt will fail. The program counter will be updated with a wrong value and thus the program flow will be corrupted.

Problem fix/Workaround

Address the stack pointer to internal SRAM or disable interrupts while stack pointer addresses external memory.

8-Bit AVR® Microcontroller with 16K Bytes Flash

ATmega161 Rev. E, ATmega161L Rev. E

Errata Sheet

Rev. –03/01

3. Writing UBBRH affects both UART0 and UART1

Writing UBRRHI updates baud rate generator for both UART0 and UART1. The baud rate generator's counter is updated each time either UBRR or UBRRHI are written. Since the UBRRHI register is shared by UART0 and UART1, changing the baud rate for one UART will affect the operation of the other UART.

Problem fix/Workaround

Do not update UBRRHI for one UART when transmitting/receiving data on the other.

4. Store Program Memory instruction may fail

At certain frequencies and voltages, the store program memory (SPM) instruction may fail.

Problem fix/Workaround

Avoid using the SPM instruction