



8-bit **AVR**[®]
Microcontroller

Application
Note

AVR080: Replacing ATmega103 by ATmega128

Features

- ATmega103 Errata Corrected in ATmega128
- Changes to Names
- Improvements to Timer/Counters and Prescalers
- Improvements to External Memory Interface
- Improvements to the ADC
- Improvements to SPI and UART
- Changes in EEPROM Write Timing
- Changes to Programming Interface
- Fuse Settings
- Oscillators and Selecting Start-up Delays
- Changes to Watchdog Timer
- JTAG Interface
- Features not Available in ATmega103 Compatibility Mode

Introduction

This application note is a guide to assist current ATmega103 users in converting existing designs to the ATmega128. The ATmega128 has two operating modes selected through the fuse settings. The M103C Fuse selects whether the ATmega103 compatibility mode should be used or not. By default, the M103C Fuse is programmed and the ATmega128 operates in compatibility mode. When the compatibility mode is used, only non-conflicting enhancements make the part different from the ATmega103. Additionally, the electrical characteristics of the ATmega128 are different including an increase in operating frequency because of a change in process technology. Check the data sheet for detailed information. When the M103C Fuse is unprogrammed, all new features are supported, but porting the code may require more work.

ATmega103 Errata Corrected in ATmega128

The following items from the Errata Sheets of the ATmega103 does not apply to the ATmega128. Refer to the ATmega103 Errata Sheet for a more detailed description of the errata.

Note: Some of these errata entries were corrected in the last revision of ATmega103. They are still referred, to ease converting from any ATmega103 design.

Power Consumption During Slowly Rising Supply

ATmega128 power consumption is independent of power rising time.

Releasing Reset Condition Without Clock

ATmega128 has a new reset interface in which any external reset pulse exceeding the minimum pulse width t_{RST} causes an internal reset even though the condition disappears before any valid clock is present.

Rev. 2501B-AVR-06/02





Wake-up from Power-save Executes Instructions Before Interrupt	<p>ATmega128 executes the interrupt routine as the first instruction after wake-up from Power-save mode.</p> <p>If an enabled interrupt occurs while the ATmega128 is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles, executes the interrupt routine, and resumes execution from the instruction following SLEEP.</p>
SPI Can Send Wrong Byte	<p>In ATmega128, a new byte can be written to the SPI Data Register on the same clock edge as the previous transfer finishes. There is no need to wait for the previous transfer to complete before writing the next byte into the SPI Data Register when operating in Master mode.</p>
Wrong Clearing of XTRF in MCUSR	<p>The POR and XTRF Flag can be cleared individually in ATmega128.</p>
Reset During EEPROM Write	<p>If a Reset or Power-off occurs in ATmega128 during an EEPROM write, the accessed location may be corrupted, but ATmega128 will not corrupt any other locations than the one being written.</p>
SPI Interrupt Flag Can be Undefined after Reset	<p>ATmega128 resets the SPI Interrupt Flag to zero.</p>
Skip Instruction with Interrupts	<p>ATmega128 interrupts always store the correct return address, also when interrupting a skip instruction skipping a two-word instruction.</p>
Signature Bytes	<p>The signature bytes of ATmega128 are different from the ones used in ATmega103. Hence, the errata concerning incorrect Signature in ATmega103 is not applicable. Make sure you are using the signature of ATmega128 when porting the design.</p>
Read Back Value During EEPROM Polling	<p>In ATmega128, the read-back value during EEPROM polling is always \$FF.</p>
MISO Output During In-System Programming	<p>ATmega128 tri-states the MISO output during In-System Programming. The In-System Programming interface is still using PE0 and PE1 for serial data in and serial data out, respectively.</p>
The ADC has No Free Running Mode	<p>The ATmega128 supports Free Running mode.</p>
UART Looses Synchronization if RXD Line is Low when UART Receive is Disabled	<p>The UART is replaced with a USART, which does not have this problem. The starting edge of a reception is only accepted as valid if the Receive Enable bit in the USART Control Register is set.</p>
Verifying the EEPROM at High Voltages During Programming	<p>There are no restrictions on the supply voltage or system frequency as long as operated inside the voltage and frequency range prescribed in the data sheet for the ATmega128.</p>
Verifying EEPROM In-System	<p>There are no restrictions on the supply voltage or system frequency as long as the device is operated inside the voltage and frequency range prescribed in the data sheet for the ATmega128.</p>
Serial Programming at Voltages Below 3.4V	<p>There are no restrictions on the supply voltage or system frequency as long as the device is operated inside the voltage and frequency range prescribed in the data sheet for the ATmega128.</p>

Changes to Names

The following control bits have changed names, but have the same functionality and placement when accessed as in ATmega103:

Table 1. Changed Bit Names

Bit Name in ATmega103	Bit Name in ATmega128	I/O Register (ATmega103)	Comments
SRW	SRW10	MCUCR	
PWMn(0)	WGMn0	TCCRn(A)	“A” and “0” in 16-bit timers only
PWMn1	WGMn1	TCCRnA	
CTCn	WGMn2	TCCRn(B)	“B” in 16-bit timers only
WDTOE	WDCE	WDTCR	See “Changes to Watchdog Timer” on page 9.
RXCIE	RXCIE0	UCR	
TCXIE	TCXIE0	UCR	
UDRIE	UDRIE0	UCR	
RXEN	RXEN0	UCR	
TXEN	TXEN0	UCR	
CHR9	UCSZ02	UCR	
RXB	RXB0	UCR	
TXB	TXB0	UCR	
RXC	RXC0	USR	
TXC	TXC0	USR	
UDRE	UDRE0	USR	
FE	FE0	USR	
OR	DOR0	USR	

The following I/O Registers have changed names, but include the same functionality and placement when accessed as in ATmega103:

Table 2. Changed Register Names

Register Name in ATmega103	Register Name in ATmega128	Comments
EICR	EICRB	
MCUSR	MCUCSR	
UDR	UDR0	
USR	UCSR0A	
UCR	UCSR0B	
UBRR	UBRR0L	
ADCSR	ADCSRA	

Improvements to Timer/Counters and Prescalers

For details about the improved and additional features, please refer to the data sheet. The following features have been added:

- The Prescalers in ATmega128 can be reset.
- Variable top value in PWM mode.
- For Timer/Counter1, Phase and Frequency Correct PWM mode in addition to the Phase Correct PWM mode.
- Fast PWM mode.

Differences Between ATmega128 and ATmega103

Most of the improvements and changes apply to all the Timer/Counters and the description below is written in a general form. A lower case “x” replaces the output channel (A or B for Timer/Counter1, N/A for Timer/Counter0 and Timer/Counter2), while “n” replaces the Timer/Counter number (n = 0, 1, or 2).

TCNT1 Cleared in PWM Mode

In ATmega103 there are three different PWM resolutions – 8, 9, or 10 bits. Though only 8, 9, or 10 bits are compared, it is still possible to write values into the TCNT1 Register that exceed the resolution. Thus, the Timer/Counter has to complete the count to 0xFFFF before the reduced resolution becomes effective (i.e. if 8-bit resolution is selected and the TCNT1 Register contains 0x0100, the top value (0x00FF) will not be effective until the counter has counted up to 0xFFFF, turned, and counted down to 0x0000 again). In ATmega128 this has been changed so that the unused bits in TCNT1 are being cleared to zero to avoid this unintended counting up to 0xFFFF. In the ATmega128, the TCNT1 Register never exceeds the selected resolution.

ATmega128

The most significant bits in the TCNT1 Register will be cleared at the first positive edge of the prescaled clock.

- 8-bit PWM: TCNT1H7:0 = 0
- 9-bit PWM: TCNT1H7:1 = 0
- 10-bit PWM: TCNT1H7:2 = 0

ATmega103

TCNT1H not cleared.

OCR1xH Cleared in PWM Mode

Clearing OCR1xH in PWM mode is slightly different from clearing TCNT1. The ATmega103 clears the six most significant bits if 8, 9, or 10 bits PWM mode is selected. Hence, if 0xFFFF is written to OCR1x in PWM-mode and OCR1x is read back, the result is 0x03FF regardless of which PWM mode that is selected. In ATmega128 the number of cleared bits depends on the resolution.

ATmega128

The most significant bits in OCR1AH and OCR1BH are cleared when they are updated at the TOP-value of the counter.

- 8-bit PWM: OCR1xH7:0 = 0
- 9-bit PWM: OCR1xH7:1 = 0
- 10-bit PWM: OCR1xH7:2 = 0

ATmega103

The six most significant bits in the OCR1AH and OCR1BH are cleared regardless of the resolution.

Write to OCR1x in PWM Mode, Change to Normal Mode Before OCR1x is Updated at the Top, Read OCR1x (Applies to 16-Bit Timer/Counter Only)

As described in the data sheet, the OCR1x Registers are updated at the top value when written. Thus, when writing the OCR1x in PWM mode, the value is stored in a temporary buffer. When the Timer/Counter reaches the top, the temporary buffer is transferred to the actual Output Compare Register. If PWM mode is left after the temporary buffer is written, but before the actual Output Compare Register is updated, the behavior differs between ATmega128 and ATmega103.

ATmega128

If the OCR1x Register is read before the update is done, the actual compare value is read – not the temporary OCR1x buffer.

ATmega103

If the OCR1x Register is read before the update is done, the value in the OCR1x buffer is read. For example, the value read is the one last written (to the OCR1x buffer), but since the Timer/Counter never reached the top value, it was not latched into the OCR1x Register. Hence, the value that is used for comparison is not necessarily the same as being read.

Note: This applies to 16-bit Timer/Counter only, for 8-bit Timer/Counter, the temporary buffer is read in both devices.

Memory of Previous OCnx Pin Level

In ATmega103, there are two settings of COMnx1:0 that do not update the OCnx pin in PWM mode (0b00 and 0b01), and one setting of COMnx1:0 in non-PWM mode (0b00). Assume the Timer/Counter is taken from a state that updates the OCnx pin to a state that does not, and then back again to a state that does update the OCnx pin. The following differences should be noted:

ATmega128

The level of the OCnx pin before disabling the Output Compare mode is remembered. Re-enabling the Output Compare mode will cause the OCnx pin to resume operation from the state it had when it was disabled. All Output Compare pins are initialized to zero on Reset.

ATmega103

For Timer/Counter1 in non-PWM mode, a compare match during the time when the Timer/Counter is not connected to the pin will reset the OCnx pin to the low level once enabled again. PWM mode will update the internal register for the OCnx pin, such that the state of the pin is unknown once enabled again.

For the 8-bit Timer/Counters, the state of the Output Compare pin is unknown when re-enabling the Output Compare pin.

Only the OCnx pins for the 8-bit Timer/Counters in the ATmega103 are initialized to zero on Reset.

OCR0 or OCR2 Equal Extreme Value in TCNT0 or TCNT2 Respectively (Applies to 8-bit Timer/Counter Only)

According to the values in COMn1 and COMn0, OCn will be cleared or set when changing the COMn bits. The response on the output differs from ATmega128 to ATmega103:

ATmega128

When changing COMn bits setting, the Output pin (OCn) changes accordingly to the COMn bits after a compare match has occurred.

ATmega103

When changing COMn bits setting, the Output pin (OCn) changes immediately.

For the 16-bit Timer/Counter1, neither ATmega128 nor ATmega103 change the output before a compare match occurs.

XDIV and Timer/Counter0

The settings of XTAL Divide Control Register – XDIV – do not affect the clock source for Timer/Counter0 and associated logic in ATmega128, in ATmega103 it does. As a consequence, interrupts may be lost and accessing the Timer/Counter0 Registers may fail in ATmega128. Timer/Counter0 in ATmega128 should not be used if the system clock is divided by the XDIV Register, no matter whether Timer/Counter0 is clocked synchronously or asynchronously. In other words; if Timer/Counter0 is used, the system clock divide feature should not be used.

Improvements to External Memory Interface

The combined Address/Data port in ATmega128 outputs data until a new address is set up. Refer to the ATmega128 data sheet for details on the changed timing.

Improvements to ADC

- The ADC in ATmega128 supports Free Running mode.
- A single conversion in ATmega128 takes two cycle less than in ATmega103.
- ATmega128 supports both left adjusted and right adjusted 10-bit results.
- The ADC in ATmega128 supports differential and amplified measurements.

Improvements to SPI and UART

Both SPI and USART have new Double Speed modes which allow higher communication speed.

The UART in ATmega103 has been replaced by a USART in ATmega128. The ATmega128 USART is compatible with the ATmega103 UART with one exception: The two-level Receive Register acts as a FIFO. The FIFO is disabled when the M103C Fuse is programmed. Still the following must be kept in mind when the M103C Fuse is programmed:

- The UDR must only be read once for each incoming data.
- The Error Flags (FE and DOR) and the ninth data bit (RXB8) are buffered with the data in the receive buffer. Therefore the status bits must always be read before the UDR Register is read. Otherwise, the error status will be lost.

Another minor difference is the initial value of RXB8, which is “1” in the UART in ATmega103 and “0” in the USART in ATmega128

Changes to EEPROM Write Timing

In ATmega103, the EEPROM write time is dependent on supply voltage, typically 2.5 ms @ $V_{CC} = 5V$ and 4 ms @ $V_{CC} = 2.7V$. In ATmega128, the EEPROM write time takes 8,448 cycles of the calibrated RC Oscillator (regardless of the clock source and frequency for the system clock). The calibrated RC Oscillator is assumed to be calibrated to 1.0 MHz regardless of V_{CC} , i.e., typical write time is 8.4 ms.

Note: Changing the value in the OSCCAL Register affects the frequency of the calibrated RC Oscillator and hence the EEPROM write time.

Programming Interface

Some changes have been done to the programming interface, especially in the In-System Programming interface. This has been done to support all the additional fuses in ATmega128. The timing requirements are unchanged. See the ATmega128 data sheet for details.

The Parallel Programming algorithm is changed. The most significant change is that the PAGEL pin on ATmega128 is located on PD7, while BS2 is located on PA0. On ATmega103 the opposite pin-mapping was chosen (PAGEL pin on PA0, while BS2 was mapped to PD7). This change has been done to make it possible to use the same programmer for all new AVR devices. In Parallel mode, the ATmega128 supports page programming of the EEPROM. Note that the additional Fuses and Lock bits also require a change in the fuse writing algorithm. The timing requirements for parallel programming have been changed. See the ATmega128 data sheet for details.

The STK500 supports both In-System Programming and Parallel Programming of the ATmega128.

Fuse Settings

ATmega128 contains more fuses than ATmega103. Table 3 shows the recommended ATmega103 compatible fuse settings of ATmega128. Some of the fuses are described further in the following sections.

Table 3. Comparing Fuses in ATmega103 and ATmega128Table 1

Fuse	Default ATmega103 Setting	Default ATmega128 Setting	ATmega103 Compatible Setting
M103C	-	0	0
WDTON	-	1	1
OCDEN	-	1	1
JTAGEN	-	0	1 ⁽²⁾
SPIEN	0	0	0
CKOPT	-	1	0 ⁽³⁾
EESAVE	1	1	1
BOOTSZ1	-	0	0 (N/A) ⁽⁴⁾
BOOTSZ0	-	0	0 (N/A) ⁽⁴⁾
BOOTRST	-	1	1
BODLEVEL	-	1	1
BODEN	-	1	1
SUT1	1	1	See note ⁽⁵⁾
SUT0	1	0	See note ⁽⁵⁾
CKSEL3	-	0	See note ⁽⁵⁾
CKSEL2	-	0	See note ⁽⁵⁾
CKSEL1	-	0	See note ⁽⁵⁾
CKSEL0	-	1	See note ⁽⁵⁾

- Notes:
1. A dash indicates that the Fuse is not present in ATmega103.
 2. See "JTAG Interface" on page 9.
 3. See "Oscillators and Selecting Start-up Delays" on page 9.

4. SPM and Self-programming is not available in ATmega103 compatibility mode. The default factory setting of BOTT SZ1:0 is OK.
5. The CKSEL Fuses are available in both ATmega103 and ATmega128. However, the SUT and CKSEL setting should be reconsidered when moving to ATmega128. See “Oscillators and Selecting Start-up Delays” on page 9.

Oscillators and Selecting Start-up Delays

ATmega128 provides more Oscillators and Start-up time selections than ATmega103. During wake-up from Power-down mode and Power-save mode, the ATmega128 uses the CPU frequency to determine the wake-up delay, while ATmega103 determines the delay from the WDT Oscillator frequency (except SUT = 0b00).

Follow the guidelines from the section “System Clock and Clock Options” in the ATmega128 data sheet to find appropriate start-up values.

Special attention must be paid when changing the fuses in In-System Programming mode. In-System Programming is dependent on a system clock. If wrong Oscillator setting is programmed, it may be impossible to re-enter In-System Programming mode due to missing system clock (Parallel Programming mode must then be used).

The crystal Oscillator in ATmega103 is capable of driving an addition clock buffer from the XTAL2 output. In ATmega128, this is only possible when the CKOPT Fuse is programmed. In this mode the Oscillator has a rail-to-rail swing at the output, but at the expense of higher power consumption. Hence, do only program this fuse when rail-to-rail swing is required.

Changes to Watchdog Timer

The Watchdog Timer in ATmega128 is improved compared to the one in ATmega103. In ATmega103, the Watchdog Timer is either enabled or disabled, while ATmega128 supports two safety levels selected by the WDTON Fuse. See description in ATmega128 data sheet for further information.

The combination of programming the M103C Fuse and having the WDTON Fuse unprogrammed makes the Watchdog Timer behave exactly as in ATmega103.

The frequency of the Watchdog Oscillator in ATmega128 is close to 1.0 MHz for all supply voltages. The typical frequency of the Watchdog Oscillator in ATmega103 is close to 1.0 MHz at 5V, but the Time-out period increases with decreasing V_{CC} . This means that the selection of Time-out period for the Watchdog Timer (in terms of number of WDT Oscillator cycles) must be reconsidered when porting the design to ATmega128. Refer to the data sheet for ATmega128 for further information.

JTAG Interface

The ATmega128 provides a JTAG interface, which can be used for programming, Boundary-scan, and On-chip debug. Refer to data sheet for details. Note that the JTAG interface is also available in ATmega103 compatibility mode. The device is shipped with the JTAGEN Fuse programmed in order to allow programming through the JTAG interface. This fuse must be erased to be ATmega103 compatible (If not, four pins are dedicated to the Test Access Port – TAP instead of being I/O pins). The M103C Fuse does *not* override the JTAGEN Fuse.

Features not Available in ATmega103 Compatibility Mode

The M103C Fuse makes the ATmega128 compatible to ATmega103. However, with the M103C Fuse programmed, some of the new features in ATmega128 become unavailable. The following features are not supported when the ATmega128 is used in the ATmega103 compatibility mode:

- TWI – Two Wire Interface module.
- USART1 – The additional second USART.
- The FIFO operation of the USART, synchronous mode and advanced functions like parity and double stop bit.
- Boot Loader capabilities (SPM – Self Programming Memories).
- Advanced External Memory Control (more wait-states, configurable number of bits are assigned to address high byte, different wait-states settings for different pages of external memory).
- Access to the status bits JTD, JTRF, WDRF, and BORF in MCUCSR.
- A timed sequence to change Watchdog Timer prescaler settings by software.
- Additional Output Compare Register (OCR1C) for Timer/Counter1.
- Timer/Counter3 (16-bit Timer/Counter identical to Timer/Counter1).
- Edge interrupts on INT0 to INT3.
- Port C as general I/O (Digital output only in ATmega103 compatibility mode).
- Port F as general I/O (Analog/digital input only in ATmega103 compatibility mode).
- Port G (Dedicated functions only in ATmega103 compatibility mode).

If any of the features above are needed or wanted and the M103C Fuse is unprogrammed, this introduces several differences between ATmega128 and ATmega103 which do not exist as long as the compatibility fuse is programmed:

- Address space 0x0060-0x00FF is dedicated extended I/O, not internal SRAM.
- Address space 0x0100-0x10FF is dedicated internal SRAM (ATmega128 supports 4,096 locations of internal SRAM compared to 4000 in ATmega103), thus the external memory starts at address 0x1100 (external memory on ATmega103 starts at address 0x1000).
- Port C is not initialized upon reset to drive 0x00, but it is tri-stated as all other ports.
- The ALE, RD, and WR pins (PG2:0) are not configured as output until the XRAM is enabled.
- The TOSC1 and TOSC2 (PG4:3) pins are configured as digital input pins after reset, not 32 kHz Oscillator unless AS0 bit is written.
- A timed sequence must be followed to change Watchdog Timer prescaler settings by software.
- In the MCUCSR Register, all RESET Flags are present in the register, not only EXTRF and PORF as in ATmega103.
- The UART will have an extra input buffer which allows one more data byte to be received before the Data OverRun Flag (DOR) is set.



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

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

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 436-4314

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

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

© Atmel Corporation 2002.

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.