



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

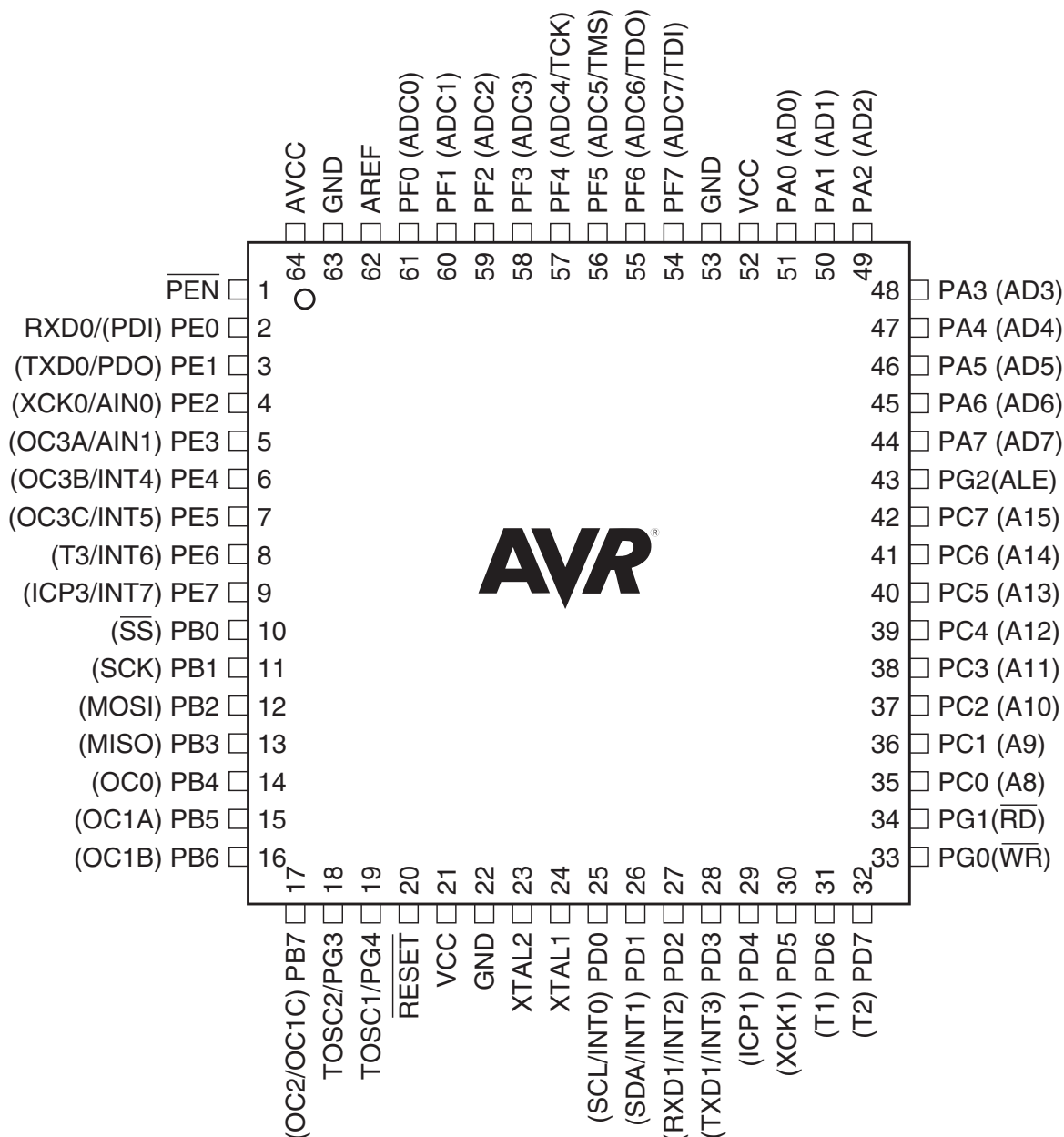
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	EBI/EMI, I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atmega128-16aj">https://www.e-xfl.com/product-detail/microchip-technology/atmega128-16aj</a>

## Pin Configurations

Figure 1. Pinout ATmega128



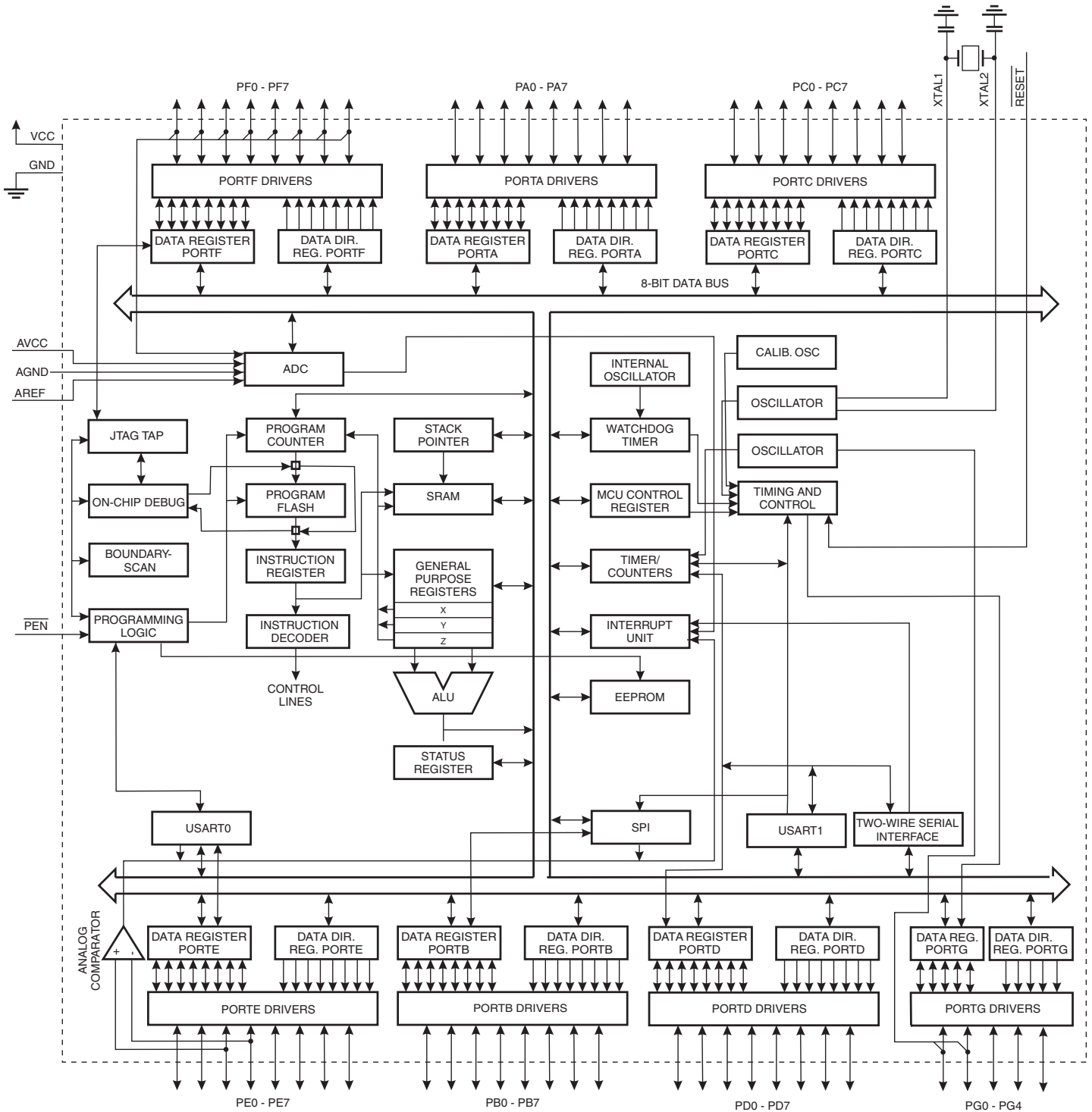
Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

## Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 2. Block Diagram



**Port C (PC7..PC0)**

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the Atmel® AVR®ATmega128 as listed on [page 76](#). In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.

Note: The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.

**Port D (PD7..PD0)**

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega128 as listed on [page 77](#).

**Port E (PE7..PE0)**

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega128 as listed on [page 80](#).

**Port F (PF7..PF0)**

Port F serves as the analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

**Port G (PG4..PG0)**

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

## **RESET**

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 19 on page 50](#). Shorter pulses are not guaranteed to generate a reset.

## **XTAL1**

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

## **XTAL2**

Output from the inverting Oscillator amplifier.

## **AVCC**

AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

## **AREF**

AREF is the analog reference pin for the A/D Converter.

## **PEN**

PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. PEN has no function during normal operation.

## Resources

A comprehensive set of development tools, application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

## Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C

## About Code Examples

This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBR”, “SBRC”, “SBR”, and “CBR”.

## Capacitive touch sensing

The Atmel QTouch Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: [www.atmel.com/qtouchlibrary](http://www.atmel.com/qtouchlibrary). For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from the Atmel website.

## Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (\$FF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \$FF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
JMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBRs	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2

## Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z + 1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Inc	Rd ← (RAMPZ:Z), RAMPZ:Z ← RAMPZ:Z + 1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBI	P, b	Set Bit in I/O Register	I/O(P, b) ← 1	None	2
CBI	P, b	Clear Bit in I/O Register	I/O(P, b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z, C, N, V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1



## Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SEV		Set Twos Complement Overflow.	$V \leftarrow 1$	V	1
CLV		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
<b>MCU CONTROL INSTRUCTIONS</b>					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

## Ordering Information

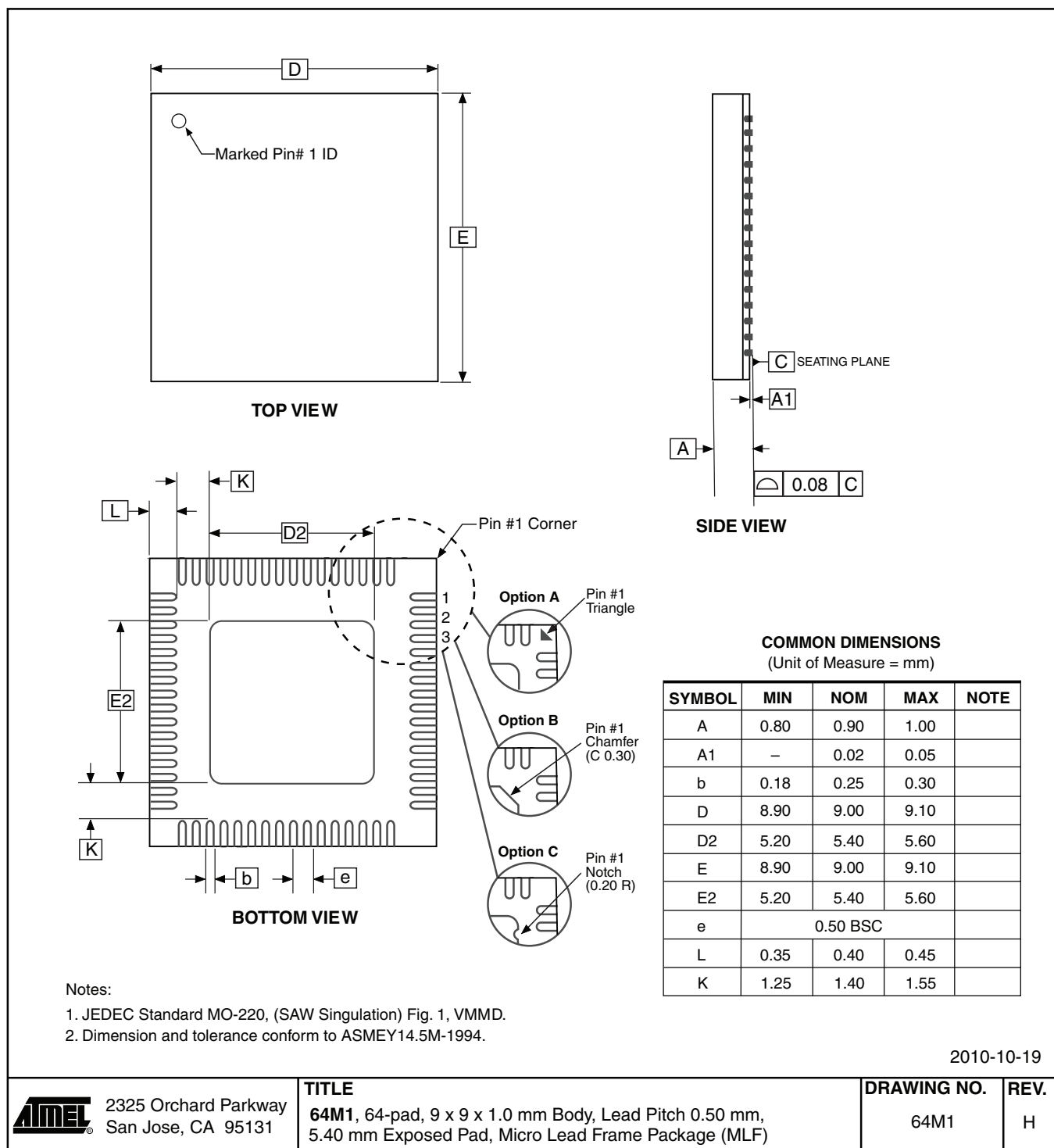
Speed (MHz)	Power Supply	Ordering Code <sup>(1)</sup>	Package <sup>(2)</sup>	Operation Range
8	2.7 – 5.5V	ATmega128L-8AU ATmega128L-8AUR <sup>(3)</sup> ATmega128L-8MU ATmega128L-8MUR <sup>(3)</sup>	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	4.5 – 5.5V	ATmega128-16AU ATmega128-16AUR <sup>(3)</sup> ATmega128-16MU ATmega128-16MUR <sup>(3)</sup>	64A 64A 64M1 64M1	
8	3.0 – 5.5V	ATmega128L-8AN ATmega128L-8ANR <sup>(3)</sup> ATmega128L-8MN ATmega128L-8MNR <sup>(3)</sup>	64A 64A 64M1 64M1	Extended (-40°C to 105°C)
16	4.5 – 5.5V	ATmega128-16AN ATmega128-16ANR <sup>(3)</sup> ATmega128-16MN ATmega128-16MNR <sup>(3)</sup>	64A 64A 64M1 64M1	

- Notes:
1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  2. The device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  3. Tape and Reel

Package Type	
<b>64A</b>	64-lead, 14 x 14 x 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP)
<b>64M1</b>	64-pad, 9 x 9 x 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)



## 64M1



## Errata

The revision letter in this section refers to the revision of the ATmega128 device.

### ATmega128 Rev. F to M

- First Analog Comparator conversion may be delayed
- Interrupts may be lost when writing the timer registers in the asynchronous timer
- Stabilizing time needed when changing XDIV Register
- Stabilizing time needed when changing OSCCAL Register
- IDCODE masks data from TDI input
- Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request

#### 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

##### Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

#### 2. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNTx) is 0x00.

##### Problem Fix/Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCRx), asynchronous Timer Counter Register (TCNTx), or asynchronous Output Compare Register (OCRx).

#### 3. Stabilizing time needed when changing XDIV Register

After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

##### Problem Fix / Workaround

The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

1. Clear the I bit in the SREG Register.
2. Set the new pre-scaling factor in XDIV register.
3. Execute 8 NOP instructions
4. Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

##### Assembly Code Example:

```

CLI                ; clear global interrupt enable
OUT  XDIV, temp    ; set new prescale value
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation
NOP                ; no operation

```

```
SEI                      ; set global interrupt enable
```

#### 4. Stabilizing time needed when changing OSCCAL Register

After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

##### **Problem Fix / Workaround**

The behavior follows errata number 3., and the same Fix / Workaround is applicable on this errata.

#### 5. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

##### **Problem Fix / Workaround**

- If ATmega128 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega128 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega128 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega128 must be the first device in the chain.

#### 6. Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.

Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

##### **Problem Fix / Workaround**

Always use OUT or SBI to set EERE in EECR.

## Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

- |                         |  |
|-------------------------|--|
| <b>Rev. 2467X-06/11</b> | <ol style="list-style-type: none"> <li>1. Corrected typos in <a href="#">“Ordering Information” on page 12</a>.</li> </ol>   |
| <b>Rev. 2467W-05/11</b> | <ol style="list-style-type: none"> <li>1. Added Atmel QTouch Library Support and QTouch Sensing Capability Features.</li> <li>2. Updated <a href="#">“DC Characteristics” on page 318</a>. <math>R_{RST}</math> maximum value changed from 60k<math>\Omega</math> to 85k<math>\Omega</math>.</li> <li>3. Updated <a href="#">“Ordering Information” on page 12</a> to include Tape &amp; Reel devices.</li> </ol>  |
| <b>Rev. 2467V-02/11</b> | <ol style="list-style-type: none"> <li>1. Updated the literature number (2467) that accidentally changed in rev U.</li> <li>2. Editing update according to the Atmel new style guide. No more space between the numbers and their units.</li> <li>3. Reorganized the swapped chapters in rev U: 8-bit Timer/Counter 0, 16-bit TC1 and TC3, and 8-bit TC2 with PWM.</li> </ol>  |
| <b>Rev. 2467U-08/10</b> | <ol style="list-style-type: none"> <li>1. Updated <a href="#">“Ordering Information” on page 12</a>. Added Ordering information for Appendix A ATmega128/L 105°C.</li> </ol>   |
| <b>Rev. 2467T-07/10</b> | <ol style="list-style-type: none"> <li>1. Updated the <a href="#">“USARTn Control and Status Register B – UCSRnB” on page 189</a>.</li> <li>2. Added a link from <a href="#">“Minimizing Power Consumption” on page 47</a> to <a href="#">“System Clock and Clock Options” on page 35</a>.</li> <li>3. Updated use of Technical Terminology in datasheet</li> <li>4. Corrected formula in <a href="#">Table 133, “Two-wire Serial Bus Requirements,” on page 322</a></li> <li>5. Note 6 and Note 7 below <a href="#">Table 133, “Two-wire Serial Bus Requirements,” on page 322</a> have been removed</li> </ol> |
| <b>Rev. 2467S-07/09</b> | <ol style="list-style-type: none"> <li>1. Updated the <a href="#">“Errata” on page 15</a>.</li> <li>2. Updated the TOC with the newest template (version 5.10).</li> <li>3. Added note “Not recommended from new designs” from the front page.</li> <li>4. Added typical <math>I_{CC}</math> values for Active and Idle mode in <a href="#">“DC Characteristics” on page 318</a>.</li> </ol>   |
| <b>Rev. 2467R-06/08</b> | <ol style="list-style-type: none"> <li>1. Removed “Not recommended from new designs” from the front page.</li> </ol>   |

## Rev. 2467Q-05/08

1. Updated **“Preventing EEPROM Corruption” on page 24.**  
Removed sentence “If the detection level of the internal BOD does not match the needed detection level, and external low  $V_{CC}$  Reset Protection circuit can be used.”
2. Updated **Table 85 on page 196 in “Examples of Baud Rate Setting” on page 193.**  
Removed examples of frequencies above 16MHz.
3. Updated **Figure 114 on page 238.**  
Inductor value corrected from 10mH to 10μH.
4. Updated description of **“Version” on page 253.**
5. **ATmega128L removed from “DC Characteristics” on page 318.**
6. Added **“Speed Grades” on page 320.**
7. Updated **“Ordering Information” on page 12.**  
Pb-Plated packages are no longer offered, and the ordering information for these packages are removed.  
There will no longer exist separate ordering codes for commercial operation range, only industrial operation range.
8. Updated **“Errata” on page 15:**  
Merged errata description for rev.F to rev.M in **“ATmega128 Rev. F to M”.**

## Rev. 2467P-08/07

1. Updated **“Features” on page 1.**
2. Added **“Data Retention” on page 8.**
3. Updated **Table 60 on page 133 and Table 95 on page 235.**
4. Updated **“C Code Example<sup>(1)</sup>” on page 176.**
5. Updated **Figure 114 on page 238.**
6. Updated **“XTAL Divide Control Register – XDIV” on page 36.**
7. Updated **“Errata” on page 15.**
8. Updated **Table 34 on page 76.**
9. Updated **“Slave Mode” on page 166.**

## Rev. 2467O-10/06

1. Added note to **“Timer/Counter Oscillator” on page 43.**
2. Updated **“Fast PWM Mode” on page 124.**
3. Updated **Table 52 on page 104, Table 54 on page 104, Table 59 on page 133, Table 61 on page 134, Table 64 on page 156, and Table 66 on page 157.**
4. Updated **“Errata” on page 15.**

## Rev. 2467N-03/06

1. Updated note for [Figure 1 on page 2](#).
2. Updated [“Alternate Functions of Port D” on page 77](#).
3. Updated [“Alternate Functions of Port G” on page 84](#).
4. Updated [“Phase Correct PWM Mode” on page 100](#).
5. Updated [Table 59 on page 133](#), [Table 60 on page 133](#).
6. Updated [“Bit 2 – TOV3: Timer/Counter3, Overflow Flag” on page 141](#).
7. Updated [“Serial Peripheral Interface – SPI” on page 162](#).
8. Updated Features in [“Analog to Digital Converter” on page 230](#)
9. Added note in [“Input Channel and Gain Selections” on page 243](#).
10. Updated [“Errata” on page 15](#).

## Rev. 2467M-11/04

1. Removed “analog ground”, replaced by “ground”.
2. Updated [Table 11 on page 40](#), [Table 114 on page 285](#), [Table 128 on page 303](#), and [Table 132 on page 321](#). Updated [Figure 114 on page 238](#).
3. Added note to [“Port C \(PC7..PC0\)” on page 6](#).
4. Updated [“Ordering Information” on page 12](#).

## Rev. 2467L-05/04

1. Removed “Preliminary” and “TBD” from the datasheet, replaced occurrences of ICx with ICPx.
2. Updated [Table 8 on page 38](#), [Table 19 on page 50](#), [Table 22 on page 56](#), [Table 96 on page 242](#), [Table 126 on page 299](#), [Table 128 on page 303](#), [Table 132 on page 321](#), and [Table 134 on page 323](#).
3. Updated [“External Memory Interface” on page 25](#).
4. Updated [“Device Identification Register” on page 253](#).
5. Updated [“Electrical Characteristics” on page 318](#).
6. Updated [“ADC Characteristics” on page 325](#).
7. Updated [“Typical Characteristics” on page 333](#).
8. Updated [“Ordering Information” on page 12](#).

## Rev. 2467K-03/04

1. Updated [“Errata” on page 15](#).



## Rev. 2467J-12/03

1. Updated [“Calibrated Internal RC Oscillator”](#) on page 41.

## Rev. 2467I-09/03

1. Updated note in [“XTAL Divide Control Register – XDIV”](#) on page 36.
2. Updated [“JTAG Interface and On-chip Debug System”](#) on page 48.
3. Updated values for  $V_{BOT}$  (BODLEVEL = 1) in [Table 19](#) on page 50.
4. Updated [“Test Access Port – TAP”](#) on page 246 regarding JTAGEN.
5. Updated description for the JTD bit on [page 255](#).
6. Added a note regarding JTAGEN fuse to [Table 118](#) on page 288.
7. Updated  $R_{PU}$  values in [“DC Characteristics”](#) on page 318.
8. Added a proposal for solving problems regarding the JTAG instruction IDCODE in [“Errata”](#) on page 15.

## Rev. 2467H-02/03

1. Corrected the names of the two Prescaler bits in the SFIOR Register.
2. Added Chip Erase as a first step under [“Programming the Flash”](#) on page 315 and [“Programming the EEPROM”](#) on page 316.
3. Removed reference to the [“Multipurpose Oscillator”](#) application note and the [“32kHz Crystal Oscillator”](#) application note, which do not exist.
4. Corrected OCn waveforms in [Figure 52](#) on page 125.
5. Various minor Timer1 corrections.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Various minor TWI corrections.
8. Added reference to [Table 124](#) on page 291 from both SPI Serial Programming and Self Programming to inform about the Flash Page size.
9. Added note under [“Filling the Temporary Buffer \(Page Loading\)”](#) on page 280 about writing to the EEPROM during an SPM Page load.
10. Removed ADHSM completely.
11. Added section [“EEPROM Write During Power-down Sleep Mode”](#) on page 24.
12. Updated drawings in [“Packaging Information”](#) on page 13.

## Rev. 2467G-09/02

1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

## Rev. 2467F-09/02

1. Added 64-pad QFN/MLF Package and updated [“Ordering Information”](#) on page 12.

2. Added the section **“Using all Locations of External Memory Smaller than 64 Kbyte”** on page 32.
3. Added the section **“Default Clock Source”** on page 37.
4. Renamed SPMCR to SPMCSR in entire document.
5. When using external clock there are some limitations regards to change of frequency. This is descried in **“External Clock”** on page 42 and **Table 131, “External Clock Drive,”** on page 320.
6. Added a sub section regarding OCD-system and power consumption in the section **“Minimizing Power Consumption”** on page 47.
7. Corrected typo (WGM-bit setting) for:  
**“Fast PWM Mode”** on page 98 (Timer/Counter0).  
**“Phase Correct PWM Mode”** on page 100 (Timer/Counter0).  
**“Fast PWM Mode”** on page 151 (Timer/Counter2).  
**“Phase Correct PWM Mode”** on page 152 (Timer/Counter2).
8. Corrected **Table 81 on page 191 (USART)**.
9. Corrected **Table 102 on page 259 (Boundary-Scan)**
10. Updated V<sub>il</sub> parameter in **“DC Characteristics”** on page 318.

## Rev. 2467E-04/02

1. Updated the Characterization Data in Section **“Typical Characteristics”** on page 333.
2. Updated the following tables:  
**Table 19 on page 50, Table 20 on page 54, Table 68 on page 157, Table 102 on page 259,**  
**and Table 136 on page 328.**
3. **Updated Description of OSCCAL Calibration Byte.**  
 In the data sheet, it was not explained how to take advantage of the calibration bytes for 2MHz, 4MHz, and 8MHz Oscillator selections. This is now added in the following sections:  
 Improved description of **“Oscillator Calibration Register – OSCCAL”** on page 41 and **“Calibration Byte”** on page 289.

## Rev. 2467D-03/02

1. Added more information about **“ATmega103 Compatibility Mode”** on page 5.
2. Updated **Table 2, “EEPROM Programming Time,”** on page 22.
3. Updated typical Start-up Time in **Table 7 on page 37, Table 9 and Table 10 on page 39,**  
**Table 12 on page 40, Table 14 on page 41, and Table 16 on page 42.**
4. Updated **Table 22 on page 56** with typical WDT Time-out.
5. Corrected description of ADSC bit in **“ADC Control and Status Register A – ADCSRA”** on page 244.

6. Improved description on how to do a polarity check of the ADC differential results in [“ADC Conversion Result” on page 241](#).
7. Corrected JTAG version numbers in [“JTAG Version Numbers” on page 256](#).
8. Improved description of addressing during SPM (usage of RAMPZ) on [“Addressing the Flash During Self-Programming” on page 278](#), [“Performing Page Erase by SPM” on page 280](#), and [“Performing a Page Write” on page 280](#).
9. Added note regarding OCDEN Fuse below [Table 118 on page 288](#).
10. Updated Programming Figures:  
[Figure 135 on page 290](#) and [Figure 144 on page 301](#) are updated to also reflect that AVCC must be connected during Programming mode. [Figure 139 on page 297](#) added to illustrate how to program the fuses.
11. Added a note regarding usage of the PROG\_PAGELOAD and PROG\_PAGEREAD instructions on [page 307](#).
12. Added Calibrated RC Oscillator characterization curves in section [“Typical Characteristics” on page 333](#).
13. Updated [“Two-wire Serial Interface”](#) section.  
 More details regarding use of the TWI Power-down operation and using the TWI as master with low TWBRR values are added into the data sheet. Added the note at the end of the [“Bit Rate Generator Unit” on page 203](#). Added the description at the end of [“Address Match Unit” on page 204](#).
14. Added a note regarding usage of Timer/Counter0 combined with the clock. See [“XTAL Divide Control Register – XDIV” on page 36](#).

## Rev. 2467C-02/02

1. Corrected Description of Alternate Functions of Port G  
 Corrected description of TOSC1 and TOSC2 in [“Alternate Functions of Port G” on page 84](#).
2. Added JTAG Version Numbers for rev. F and rev. G  
 Updated Table 100 on page 256.
3. Added Some Preliminary Test Limits and Characterization Data  
 Removed some of the TBD's in the following tables and pages:  
[Table 19 on page 50](#), [Table 20 on page 54](#), [“DC Characteristics” on page 318](#), [Table 131 on page 320](#), [Table 134 on page 323](#), and [Table 136 on page 328](#).
4. Corrected [“Ordering Information” on page 12](#).
5. Added some Characterization Data in Section [“Typical Characteristics” on page 333](#).
6. Removed Alternative Algorithm for Leaving JTAG Programming Mode.  
 See [“Leaving Programming Mode” on page 315](#).
7. Added Description on How to Access the Extended Fuse Byte Through JTAG Programming Mode.

See [“Programming the Fuses”](#) on page 317 and [“Reading the Fuses and Lock Bits”](#) on page 317.

**Atmel Corporation**

2325 Orchard Parkway  
San Jose, CA 95131  
USA

**Tel:** (+1)(408) 441-0311

**Fax:** (+1)(408) 487-2600

[www.atmel.com](http://www.atmel.com)

**Atmel Asia Limited**

Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG

**Tel:** (+852) 2245-6100

**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY

**Tel:** (+49) 89-31970-0

**Fax:** (+49) 89-3194621

**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

© 2011 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, AVR®, QTouch®, QMatrix®, AVR Studio® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® and others are registered trademarks of Microsoft Corporation in U.S. and other countries. Other terms and product names may be trademarks of others.

**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 THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, 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 AND 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 products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.