AVR 8-bit Microcontrollers

# AT15007: Differences between ATmega328/P and ATmega328PB

APPLICATION NOTE

## Introduction

This application note assists the users of Atmel® ATmega328 variants to understand the differences and use Atmel ATmega328PB.

ATmega328PB is not a drop-in replacement for ATmega328 variants, but a new device. However, the functions are backward compatible with the existing ATmega328 functions. Existing code for these devices will work in the new devices without changing existing configuration or enabling new functions. The code that is available for your existing ATmega328 variants will continue to work on the new ATmega328PB device.

The ATmega328PB is the first 8-bit Atmel AVR® device to feature the successful Atmel QTouch® Peripheral Touch Controller (PTC).

For differences in errata, typical, and electrical characteristics between ATmega328 variants and ATmega328PB, refer to the specific device datasheets.

For complete device details, refer to the latest version of the ATmega328PB datasheet available at www.atmel.com.

## Features

- Pin functionality difference
- Code compatibility
- Enhancement and added features
- Updated features

**Note:** Code compiled for ATmega328 variants are compatible and can be executed in the ATmega328PB device. Whereas, reverse code compatibility is not guaranteed.

# Table of Contents

# 1. Pin Functionality Difference

## 1.1. Additional Pin Functionalities

ATmega328PB supports four additional GPIOs on PORTE [3:0].

The GPIO pins PE2 and PE3 are assigned to Pin19 and Pin22. PE2 and PE3 are multiplexed with ADC6 and ADC7.

Pin3 (GND) and Pin6 (VCC) are replaced by PE0 and PE1 respectively. PE0 is multiplexed with ACO.

**Table 1-1. Pin Functionality Difference between ATmega328 Variants and ATmega328PB**

| 32-pin TQFP/MLF package | ATmega328 variants | ATmega328PB |
|---|---|---|
| Pin 3 | GND | PE0/ACO |
| Pin 6 | VCC | PE1 |
| Pin 19 | ADC6 | ADC6/PE2 |
| Pin 22 | ADC7 | ADC7/PE3 |

## 1.2. Alternate Pin Configuration

The alternate pin configurations are:

- **ADC7– Port E, Bit 3**
  PE3 can also be used as ADC input channel 7.
  **Note:** ADC input channel 7 uses analog power AVCC.
- **ADC6 – Port E, Bit 2**
  PE2 can also be used as ADC input channel 6.
  **Note:** ADC input channel 6 uses analog power AVCC.
- **None – Port E, Bit 1**
  No alternate function.
- **ACO – Port E, Bit 0**
  ACO Analog Compare Output pin is multiplexed with PE0.

# 2. Enhancement and Additional Features in ATmega328PB

Compared to existing ATmega328 variants, the following enhancements or additional features are available in ATmega328PB:

- PTC - Peripheral Touch Controller.
- CFD - Clock Failure Detection mechanism.
- OCM1C2 - Output Compare Modulator.
- USART start frame detection is available in all sleep modes
- Analog Comparator output is available on a pin. This pin is multiplexed with PE0.
- Unique device ID to identify the device
- Additional USART
- Additional SPI
- Additional TWI
- Additional Timer/Counters

## 2.1. PTC - Peripheral Touch Controller

The ATmega328PB is the first 8-bit Atmel AVR device to feature the successful Atmel QTouch Peripheral Touch Controller (PTC). The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self and mutual capacitance sensors.

The PTC supports 24 buttons in self-capacitance mode and up to 144 buttons in mutual-capacitance mode. It is possible to mix and match mutual-and self-capacitance sensors. Only one pin is required per electrode—no external components are required providing considerable savings on the BOM cost compared to competing solutions.

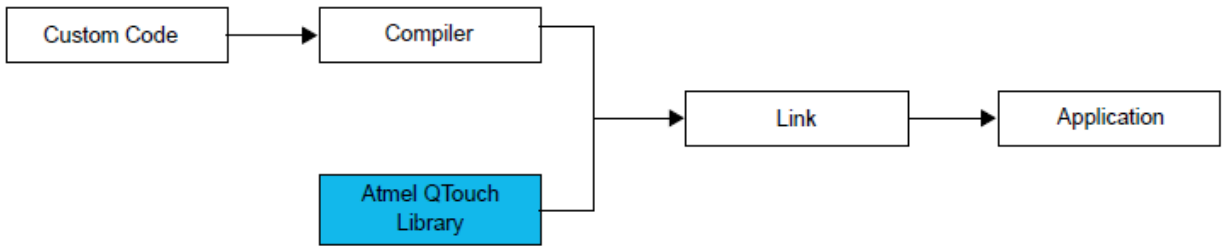
In mutual-capacitance mode, sensing is performed using capacitive touch matrices in various X-Y configurations. Whereas in self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

Refer to the chapter *I/O Multiplexing* in the ATmega328PB device datasheet for details on the pin mapping for this peripheral. A signal can be mapped on several pins.

### 2.1.1. PTC Functional Description

To access the PTC, the user must use the QTouch Composer tool to configure and link the QTouch Library firmware with the application code. QTouch Library can be used to implement buttons, sliders, wheels, and proximity sensor in a variety of combinations on a single interface.

**Figure 2-1. QTouch Library Usage**



## 2.2. CFD - Clock Failure Detection mechanism

Clock Failure Detection and Switching Mechanism is a new feature introduced in ATmega328PB. This digital logic detects the failure of the Low power crystal oscillator, Full swing crystal oscillator, and external clocks. If a failure is detected, this logic will automatically switch the clock to 1MHz internal RC system clock.

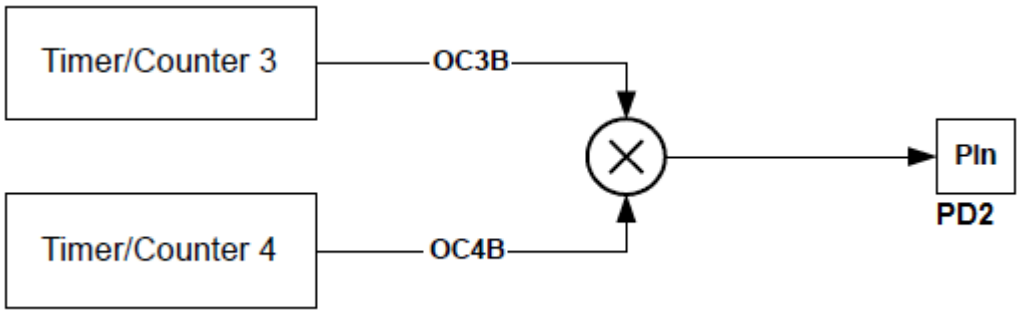
The Clock Failure Detection mechanism for the device is enabled by an active high fuse. When the CFD fuse is enabled, 128kHz oscillator will be enabled and the CFD circuit works using that clock.

CFD will be automatically disabled when the chip enters power save/down sleep mode. It will be enabled by itself when the chip returns to active mode. CFD will be enabled only when the system frequency is greater than 256kHz.

## 2.3. OCM1C2 - Output Compare Modulator

The Output Compare Modulator (OCM) allows generation of waveforms modulated with a carrier frequency. The modulator uses the outputs from the Output Compare Unit B of the 16-bit Timer/Counter3 and the Output Compare Unit of the 16-bit Timer/Counter4. When the modulator is enabled, the two output compare channels are modulated together as shown in the block diagram.

**Figure 2-2. Output Compare Modulator - Block Diagram**




The Output Comparator unit 3B and Output compare unit 4B shares the PD2 port pin for output. The outputs of the Output Compare units (OC3B and OC4B) overrides the normal PORTD2 bit when one of them is enabled (that is, when COMnx1:0 is not equal to zero). When both OC3B and OC4B are enabled simultaneously, the modulator is automatically enabled.

## 2.4. USART

ATmega328PB has one additional USART with start-of-frame detection, which can wake up the MCU from all sleep modes - when a start bit is detected. Two USART modules are available in the ATmega328PB with individual configuration registers, refer ***Register Description*** section under the USART module in the ATmega328PB device datasheet for detailed description of these registers. They also have separate TX, RX, and XCK pins. For details on the pin mapping for this peripheral, refer to the ***I/O Multiplexing*** section in the ATmega328PB device datasheet.

When a high-to-low transition is detected on RxDn, the internal 8MHz oscillator is powered up and the USART clock is enabled. After start-up the rest of the data frame can be received, provided that the baud rate is slow enough to allow the internal 8MHz oscillator to start up. Start-up time of the internal 8MHz oscillator varies with supply voltage and temperature.

The USART start frame detection works both in asynchronous and synchronous modes. It is enabled by writing the Start Frame Detection Enable bit (SFDEn). If the USART Start- Interrupt Enable (RXSIE) bit is set, the USART Receive Start Interrupt is generated immediately when a start is detected.

When using the feature without the Receive Start Interrupt, the start detection logic activates the internal 8MHz oscillator and the USART clock while the frame is being received only. Other clocks remain stopped until the Receive Complete Interrupt optionally wakes up the MCU.

The maximum baud rate depends on the sleep mode the device is woken up from.

In synchronous mode:

- **Idle or ADC Noise Reduction sleep mode:** system clock frequency divided by four
- **Standby or Power-down:** 500kbps

In asynchronous mode:

- **Idle sleep mode:** the same as in active mode

## 2.5. Analog Comparator

Analog Comparator output is available on a pin. The analog comparator output is tied to PE0 when the AC output is enabled by writing a one to the Analog Comparator Output Enable bit (ACOE) in “ACSR0 – Analog Comparator Output Control Register”.

## 2.6. Serial Number

In Atmel ATmega328PB, each individual part has a specific serial number (also called unique device ID) to identify a specific part while it is in the field. The serial number consists of bytes, which can be accessed from the signature address space.

To read the Signature Row from software, load the Z-pointer with the signature byte address given in the following table and set the SIGRD and SPMEN bits in SPMCSR (SPMCSR.SIGRD and SPMCSR.SPMEN).

When an LPM instruction is executed within three CPU cycles after the SPMCSR.SIGRD and SPMCSR.SPMEN are set, the signature byte value will be loaded in the destination register.

The SPMCSR.SIGRD and SPMCSR.SPMEN will auto-clear upon completion of reading the Signature Row Lock bits, or, if no LPM instruction is executed, within three CPU cycles. When SPMCSR.SIGRD and SPMCSR.SPMEN are cleared, LPM will work as described in the Instruction set Manual.

**Table 2-1. Signature Row Addressing**

| Signature byte | Z-pointer address |
|---|---|
| Serial Number Byte 0 | 0x000E |
| Serial Number Byte 1 | 0x000F |
| Serial Number Byte 2 | 0x0010 |
| Serial Number Byte 3 | 0x0011 |
| Serial Number Byte 4 | 0x0012 |
| Serial Number Byte 5 | 0x0013 |
| Serial Number Byte 6 | 0x0014 |

| Signature byte | Z-pointer address |
|---|---|
| Serial Number Byte 7 | 0x0015 |
| Serial Number Byte 8 | 0x0016 |
| Serial Number Byte 9 | 0x0017 |

## 2.7. Additional SPI

ATmega328PB has one Additional SPI. There are two SPIs with individual configuration registers. Refer to the ***Register Description*** section in the SPI peripheral in the ATmega328PB device datasheet for detailed description of these registers. They also have a separate MOSI, MISO, SCK, and SS pins. Refer to the ***I/O Multiplexing*** section in the ATmega328PB device datasheet for details about the pin mapping for this peripheral.

## 2.8. Additional TWI

ATmega328PB has one additional byte-oriented 2-wire serial interface (TWI). There are two TWI peripherals with individual configuration registers. Refer the ***Register Description*** section under the TWI - 2-wire Serial Interface module in the ATmega328PB device datasheet for detailed description of these registers. Separate SDA and SDL pins are also available. Refer to section ***I/O Multiplexing*** in the ATmega328PB device datasheet for details on the pin mapping for this peripheral.

## 2.9. Additional Timer/Counters

ATmega328PB has two additional 16-bit Timer/Counters(**TC3** and **TC4**) with separate Prescaler, Compare Mode, and Capture Mode. There are three 16-bit Timer/Counters (**TC1**, **TC3**, and **TC4**) and ten PWM channels available in ATmega328PB. Refer to the ***I/O Multiplexing*** section in the ATmega328PB device datasheet for details about the pin mapping for this peripheral.

# 3. Updated Features

## 3.1. Full Swing Oscillator

Clock source options of the ATmega328 variants include full swing crystal oscillator, which can be selected by configuring the flash fuse. However, in the new ATmega328PB, the full swing crystal oscillator is removed. Refer to the "Clock Sources" chapter of the respective datasheet.

**Table 3-1. Full Swing Oscillator Removed from ATmega328PB**

| Device function | ATmega328PB | ATmega328 variants |
|---|---|---|
| Full swing crystal oscillator | No | Yes |

## 3.2. Calibrated Internal RC Oscillator Accuracy

The accuracy for the Calibrated Internal RC Oscillator has been improved in ATmega328PB.

**Table 3-2. Calibration Accuracy of Internal RC Oscillator under 8MHz Frequency**

| Test condition | Accuracy for ATmega328PB | Accuracy for ATmega328 variants |
|---|---|---|
| Factory calibration | ±2% | ±10% |
| User calibration | ±1% | ±1% |

## 3.3. Parallel Programming

Parallel programming timing in ATmega328PB has been modified, comparing with ATmega328 variants. For details, see the below table, "Parallel Programming Timing Differences".

**Table 3-3. Parallel Programming Timing Differences**

| Symbol | Parameter | ATmega328PB | | ATmega328 variants | | Units |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| $t_{WLRH}$ | WR Low to RDY/BSY High | 3.2 | 3.4 | 3.7 | 4.5 | ms |
| $t_{WLRH\_CE}$ | WR Low to RDY/BSY High for Chip Erase | 9.8 | 10.5 | 7.5 | 9 | ms |
| $t_{BVDV}$ | BS1 Valid to DATA valid | 0 | 350 | 0 | 250 | ns |
| $t_{OLDV}$ | OE Low to DATA Valid | 0 | 350 | -- | 250 | ns |

## 3.4. Power Save Mode

Power consumption in power save modes for ATmega328PB will be higher when comparing with the device ATmega328 variants. For more detail, refer to the respective datasheets.

## 3.5. NVM

Write wait delay for NVM in ATmega328PB is increased when comparing with ATmega328 variants.

**Table 3-4. Minimum Wait Delay for NVM**

| Symbol | ATmega328PB | ATmega328 variants (rev. K) | Unit |
|---|---|---|---|
| $t_{WD\_ERASE}$ | 9.0 | 10.5 | ms |

## 3.6. Signature Bytes

All Atmel microcontrollers have a 3-byte signature code which identifies the device. This code can be read in both serial and parallel mode, also when the device is locked. The three bytes reside in a separate address space. For the device signature bytes, there are differences between ATmega328PB and ATmega328 variants, see below table for more detail.

**Table 3-5. Device Signature Bytes differences**

| Part | Signature bytes address | | |
|---|---|---|---|
| | 0x000 | 0x001 | 0x002 |
| ATmega328 | 0x1E | 0x95 | 0x14 |
| ATmega328P | 0x1E | 0x95 | 0x0F |
| ATmega328PB | 0x1E | 0x95 | 0x16 |

# 4. Register Description

## 4.1. Port E Input Pins Address

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

**Name:** PINE
**Offset:** 0x2C
**Reset:** N/A
**Property:** When addressing as I/O Register: address offset is 0x0C

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | PINE6 | PINE5 | PINE4 | PINE3 | PINE2 | PINE1 | PINE0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | x | x | x |

**Bits 0, 1, 2, 3 – PINEn: Port E Input Pins Address [n = 3:0]**
Writing to the pin register provides toggle functionality for I/O.

**Bits 0, 1, 2, 3, 4, 5, 6 – PINEn: Port E Input Pins Address [n = 6:0]**
Writing to the pin register provides toggle functionality for I/O.

## 4.2. Port E Data Direction Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

**Name:** DDRE
**Offset:** 0x2D
**Reset:** 0x00
**Property:** When addressing as I/O Register: address offset is 0x0D

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | DDRE6 | DDRE5 | DDRE4 | DDRE3 | DDRE2 | DDRE1 | DDRE0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3 – DDREn: Port E Data Direction**
This bit field selects the data direction for the individual pins in the Port. When a Port is mapped as virtual, accessing this bit field is identical to accessing the actual DIR register for the Port.

**Bits 0, 1, 2, 3, 4, 5, 6 – DDREn: Port E Data Direction**
This bit field selects the data direction for the individual pins in the Port. When a Port is mapped as virtual, accessing this bit field is identical to accessing the actual DIR register for the Port.

## 4.3. Port E Data Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

**Name:** PORTE
**Offset:** 0x2E
**Reset:** 0x00
**Property:** When addressing as I/O Register: address offset is 0x0E

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | PORTE6 | PORTE5 | PORTE4 | PORTE3 | PORTE2 | PORTE1 | PORTE0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3 – PORTEn: Port E Data [n = 3:0]**

**Bits 0, 1, 2, 3, 4, 5, 6 – PORTEn: Port E Data [n = 6:0]**

## 4.4. XOSC Failure Detection Control And Status Register

**Name:** XFDCSR
**Offset:** 0x62
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | XFDIF | XFDIE |
| Access | | | | | | | R | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – XFDIF: Failure Detection Interrupt Flag**
This bit is set when a failure is detected, and it can be cleared only by reset.

It serves as status bit for CFD.

**Note:** This bit is read only.

**Bit 0 – XFDIE: Failure Detection Interrupt Enable**
Setting this bit will enable the interrupt which will be issued when XFDIF is set. This bit is enable only. Once enabled, it is not possible for the user to disable.

## 4.5. USART Control and Status Register 0 D

This register is not used in Master SPI Mode (`UMSEL0[1:0] = 11`)

**Name:** UCSR0D
**Offset:** 0xC3
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXIE | RXS | SFDE | | | | | |
| Access | R/W | R/W | R/W | | | | | |
| Reset | 0 | 0 | 0 | | | | | |

**Bit 7 – RXIE: USART RX Start Interrupt Enable**
Writing this bit to one enables the interrupt on the RXS flag. In sleep modes this bit enables start frame detector that can wake up the MCU when a start condition is detected on the RxD line. The USART RX Start Interrupt is generated only, if the RXSIE bit, the Global Interrupt flag, and RXS are set.

**Bit 6 – RXS: USART RX Start**
The RXS flag is set when a start condition is detected on the RxD line. If the RXSIE bit and the Global Interrupt Enable flag are set, an RX Start Interrupt will be generated when the flag is set. The flag can only be cleared by writing a logical one on the RXS bit location.

If the start frame detector is enabled (RXSIE = 1) and the Global Interrupt Enable flag is set, the RX Start Interrupt will wake up the MCU from all sleep modes.

**Bit 5 – SFDE: Start Frame Detection Enable**
Writing this bit to one enables the USART Start Frame Detection mode. The start frame detector is able to wake up the MCU from sleep mode when a start condition, i.e. a high (IDLE) to low (START) transition, is detected on the RxD line.

**Table 4-1. USART Start Frame Detection Modes**

| SFDE | RXSIE | RXCIE | Description |
|---|---|---|---|
| 0 | X | X | Start frame detector disabled |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Start frame detector enabled. RXC flag wakes up MCU from all sleep modes |
| 1 | 1 | 0 | Start frame detector enabled. RXS flag wakes up MCU from all sleep modes |
| 1 | 1 | 1 | Start frame detector enabled. Both RXC and RXS wake up the MCU from all sleep modes |

## 4.6. USART Control and Status Register n D

This register is not used in Master SPI Mode (`UCSRnC.UMSEL[1:0] = 11`)

**Name:** UCSR0D, UCSR1D
**Offset:** 0xC3 + n*0x08 [n=0..1]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXIE | RXS | SFDE | | | | | |
| Access | R/W | R/W | R/W | | | | | |
| Reset | 0 | 0 | 0 | | | | | |

**Bit 7 – RXIE: USART RX Start Interrupt Enable**
Writing this bit to one enables the interrupt on the RXS flag. In sleep modes this bit enables start frame detector that can wake up the MCU when a start condition is detected on the RxD line. The USART RX Start Interrupt is generated only, if the RXSIE bit, the Global Interrupt flag, and RXS are set.

**Bit 6 – RXS: USART RX Start**
The RXS flag is set when a start condition is detected on the RxD line. If the RXSIE bit and the Global Interrupt Enable flag are set, an RX Start Interrupt will be generated when the flag is set. The flag can only be cleared by writing a logical one on the RXS bit location.

If the start frame detector is enabled (RXSIE = 1) and the Global Interrupt Enable flag is set, the RX Start Interrupt will wake up the MCU from all sleep modes.

**Bit 5 – SFDE: Start Frame Detection Enable**
Writing this bit to one enables the USART Start Frame Detection mode. The start frame detector is able to wake up the MCU from sleep mode when a start condition, i.e. a high (IDLE) to low (START) transition, is detected on the RxD line.

**Table 4-2. USART Start Frame Detection Modes**

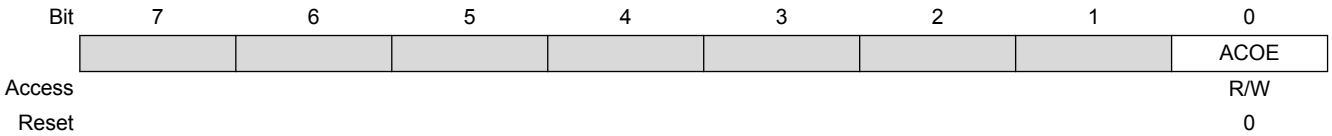
| SFDE | RXSIE | RXCIE | Description |
|---|---|---|---|
| 0 | X | X | Start frame detector disabled |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Start frame detector enabled. RXC flag wakes up MCU from all sleep modes |
| 1 | 1 | 0 | Start frame detector enabled. RXS flag wakes up MCU from all sleep modes |
| 1 | 1 | 1 | Start frame detector enabled. Both RXC and RXS wake up the MCU from all sleep modes |

## 4.7. Analog Comparator Control and Status Register B

The Store Program Memory Control and Status Register contains the control bits needed to control the Boot Loader operations.

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

**Name:** ACSRB
**Offset:** 0x4F
**Reset:** 0x00
**Property:** When addressing as I/O Register: address offset is 0x2F




**Bit 0 – ACOE: Analog Comparator Output Enable**
When this bit is set, the analog comparator output is connected to the ACO pin.

# 5. Revision History

| Doc. Rev. | Date | Comments |
|---|---|---|
| 42626C | 11/2016 | Bullet point "Additional USART" is added in chapter "Enhancement and Additional Features in ATmega328PB" |
| 42626B | 09/2016 | Added "Updated Features" |
| 42626A | 11/2015 | Initial document release |

Atmel | Enabling Unlimited Possibilities®

**Atmel Corporation** 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**