



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 | 8MHz |
| Connectivity | SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 32 |
| Program Memory Size | 8KB (4K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 512 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4V ~ 6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 70°C |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at90s8515a-8ac |



one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The AT90S8515 provides the following features: 8K bytes of In-System Programmable Flash, 512 bytes EEPROM, 512 bytes SRAM, 32 general-purpose I/O lines, 32 general-purpose working registers, flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, programmable Watchdog Timer with internal oscillator, an SPI serial port and two software-selectable power-saving modes. The Idle Mode stops the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip In-System Programmable Flash allows the program memory to be reprogrammed In-System through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S8515 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The AT90S8515 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators and evaluation kits.

Pin Descriptions

| | |
|--------------------------|--|
| VCC | Supply voltage. |
| GND | Ground. |
| Port A (PA7..PA0) | <p>Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not active.</p> <p>Port A serves as multiplexed address/data input/output when using external SRAM.</p> |
| Port B (PB7..PB0) | <p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not active.</p> <p>Port B also serves the functions of various special features of the AT90S8515 as listed on page 66.</p> |
| Port C (PC7..PC0) | <p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers can sink 20 mA. 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 active.</p> <p>Port C also serves as address output when using external SRAM.</p> |
| Port D (PD7..PD0) | <p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source</p> |

Architectural Overview

The fast-access register file concept contains 32 x 8-bit general-purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one ALU (Arithmetic Logic Unit) operation is executed. Two operands are output from the register file, the operation is executed and the result is stored back in the register file – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing, enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function. These added function registers are the 16-bit X-, Y-, and Z-register.

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8515 AVR RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowermost Data Space addresses (\$00 - \$1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions such as Control Registers, Timer/Counters, A/D converters and other I/O functions. The I/O memory can be accessed directly or as the Data Space locations following those of the register file, \$20 - \$5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed with a two-stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Programmable Flash memory.

With the relative jump and call instructions, the whole 4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 16-bit Stack Pointer (SP) is read/write-accessible in the I/O space.

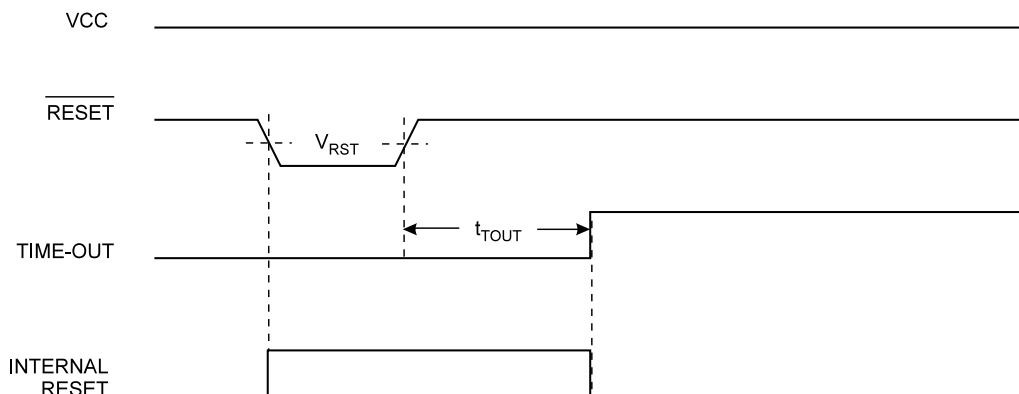
The 512-byte data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

External Reset

An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage (V_{RST}) on its positive edge, the delay timer starts the MCU after the Time-out period t_{TOUT} has expired.

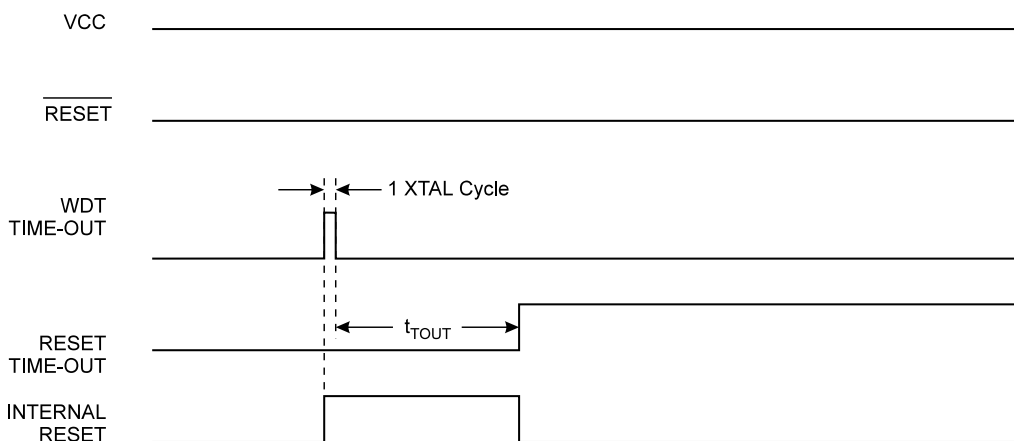
Figure 26. External Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 42 for details on operation of the Watchdog.

Figure 27. Watchdog Reset during Operation



Interrupt Handling

The AT90S8515 has two 8-bit interrupt mask control registers; GIMSK (General Interrupt Mask register) and TIMSK (Timer/Counter Interrupt Mask register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction (RETI) is executed.

For interrupts triggered by events that can remain static (e.g., the Output Compare Register1 matching the value of Timer/Counter1), the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine and resumes execution from the instruction following SLEEP. The contents of the register file, SRAM and I/O memory are unaltered. If a reset occurs during Sleep Mode, the MCU wakes up and executes from the Reset vector.

Idle Mode

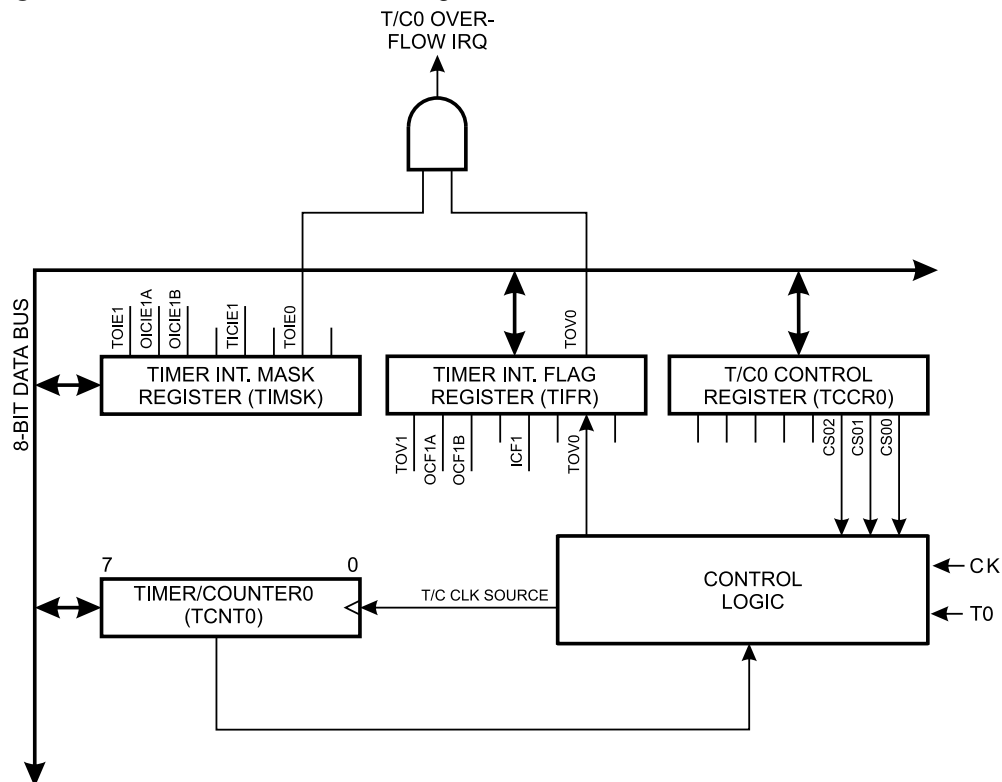
When the SM bit is cleared (zero), the SLEEP instruction forces the MCU into the Idle Mode, stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like Timer Overflow interrupt and Watchdog reset. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD-bit in the Analog Comparator Control and Status Register (ACSR). This will reduce power consumption in Idle Mode. When the MCU wakes up from Idle Mode, the CPU starts program execution immediately.

Power-down Mode

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power-down mode. In this mode, the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only an external reset, a Watchdog reset (if enabled), or an external level interrupt on INT0 or INT1 can wake up the MCU.

Note that when a level-triggered interrupt is used for wake-up from power-down, the low level must be held for a time longer than the reset delay Time-out period t_{TOU} . Otherwise, the MCU will fail to wake up.

Figure 29. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register – TCCR0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|---|---|---|---|------|------|------|-------|
| \$33 (\$53) | – | – | – | – | – | CS02 | CS01 | CS00 | TCCR0 |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read as zero.

- Bits 2, 1, 0 – CS02, CS01, CS00: Clock Select0, Bits 2, 1 and 0**

The Clock Select0 bits 2, 1 and 0 define the prescaling source of Timer/Counter0.

Table 7. Clock 0 Prescale Select

| CS02 | CS01 | CS00 | Description |
|------|------|------|--------------------------------------|
| 0 | 0 | 0 | Stop, the Timer/Counter0 is stopped. |
| 0 | 0 | 1 | CK |
| 0 | 1 | 0 | CK/8 |
| 0 | 1 | 1 | CK/64 |
| 1 | 0 | 0 | CK/256 |
| 1 | 0 | 1 | CK/1024 |
| 1 | 1 | 0 | External Pin T0, falling edge |
| 1 | 1 | 1 | External Pin T0, rising edge |

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used for Timer/Counter1, transitions on PB1/(T1) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

Timer/Counter1 – TCNT1H AND TCNT1L

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---------------|------------|-----|-----|-----|-----|-----|-----|------------|--------------------------------|
| \$2D (\$4D) | MSB | | | | | | | | TCNT1H TCNT1L |
| \$2C (\$4C) | | | | | | | | LSB | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- **TCNT1 Timer/Counter1 Write:**
When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.
- **TCNT1 Timer/Counter1 Read:**
When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

Timer/Counter1 Output Compare Register – OCR1AH AND OCR1AL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---------------|------------|-----|-----|-----|-----|-----|-----|------------|--------------------------------|
| \$2B (\$4B) | MSB | | | | | | | | OCR1AH OCR1AL |
| \$2A (\$4A) | | | | | | | | LSB | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

SPI Data Register – SPDR

| | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$0F (\$2F) | MSB | | | | | | | LSB | SPDR |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | x | x | x | x | x | x | x | x | Undefined |

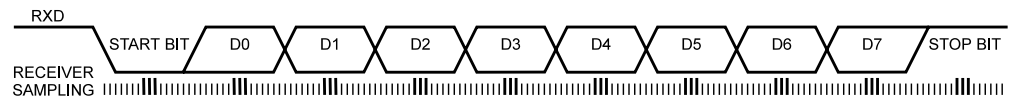
The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.



The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical “0” will be interpreted as the falling edge of a start bit and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the receiver samples the RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical “1”s, the start bit is rejected as a noise spike and the receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift register as they are sampled. Sampling of an incoming character is shown in Figure 40.

Figure 40. Sampling Received Data



When the stop bit enters the receiver, the majority of the three samples must be “1” to accept the stop bit. If two or more samples are logical “0”s, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect framing errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9-bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit Shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been read since the last receive, the OverRun (OR) flag in USR is set. This means that the last data byte shifted into the shift register could not be transferred to UDR and has been lost. The OR bit is buffered and is updated when the valid data byte in UDR is read. Thus, the user should always check the OR bit after reading the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PD0, which is forced to be an input pin regardless of the setting of the DDD0 bit in DDRD. When PD0 is forced to input by the UART, the PORTD0 bit can still be used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9 bits long, plus start and stop bits. The ninth data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The ninth data bit received is the RXB8 bit in the UCR register.

The FE bit is cleared when the stop bit of received data is one.

- **Bit 3 – OR: Overrun**

This bit is set if an Overrun condition is detected, i.e., when a character already present in the UDR register is not read before the next character has been shifted into the Receiver Shift register. The OR bit is buffered, which means that it will be set once the valid data still in UDRE is read.

The OR bit is cleared (zero) when data is received and transferred to UDR.

- **Bits 2..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and will always read as zero.

UART Control Register – UCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|------------|
| \$0A (\$2A) | RXCIE | TXCIE | UDRIE | RXEN | TXEN | CHR9 | RXB8 | TXB8 | UCR |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R | W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 5 – UDRIE: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 4 – RXEN: Receiver Enable**

This bit enables the UART receiver when set (one). When the receiver is disabled, the RXC, OR and FE status flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

- **Bit 3 – TXEN: Transmitter Enable**

This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

- **Bit 2 – CHR9: 9-bit Characters**

When this bit is set (one) transmitted and received characters are 9 bits long plus start and stop bits. The ninth bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The ninth data bit can be used as an extra stop bit or a parity bit.

- **Bit 1 – RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the ninth data bit of the received character.

- **Bit 0 – TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the ninth data bit in the character to be transmitted.

using the SBI or CBI instruction, ACI will be cleared if it has become set before the operation.

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the Analog Comparator interrupt is activated. When cleared (zero), the interrupt is disabled.

- **Bit 2 – ACIC: Analog Comparator Input Capture Enable**

When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the Analog Comparator. The comparator output is, in this case, directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the Analog Comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

- **Bits 1, 0 – ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events trigger the Analog Comparator interrupt. The different settings are shown in Table 18.

Table 18. ACIS1/ACIS0 Settings

| ACIS1 | ACIS0 | Interrupt Mode |
|-------|-------|---|
| 0 | 0 | Comparator Interrupt on Output Toggle |
| 0 | 1 | Reserved |
| 1 | 0 | Comparator Interrupt on Falling Output Edge |
| 1 | 1 | Comparator Interrupt on Rising Output Edge |

Note: When changing the ACIS1/ACIS0 bits, the Analog Comparator interrupt must be disabled by clearing its interrupt enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

Interface to External SRAM

The interface to the SRAM consists of:

Port A: Multiplexed low-order address bus and data bus

Port C: High-order address bus

The ALE pin: Address latch enable

The \overline{RD} and \overline{WR} pins: Read and write strobes

The external data SRAM is enabled by setting the SRE (external SRAM enable) bit of the MCUCR (MCU Control Register) and will override the setting of the Data Direction Register (DDRA). When the SRE bit is cleared (zero), the external data SRAM is disabled and the normal pin and data direction settings are used. When SRE is cleared (zero), the address space above the internal SRAM boundary is not mapped into the internal SRAM, as AVR parts do not have an interface to the external SRAM.

When ALE goes from high to low, there is a valid address on Port A. ALE is low during a data transfer. \overline{RD} and \overline{WR} are active when accessing the external SRAM only.

When the external SRAM is enabled, the ALE signal may have short pulses when accessing the internal RAM, but the ALE signal is stable when accessing the external SRAM.

Figure 42 sketches how to connect an external SRAM to the AVR using eight latches that are transparent when G is high.

Port B

Port B is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port B, one each for the Data Register – PORTB, \$18(\$38), Data Direction Register – DDRB, \$17(\$37) and the Port B Input Pins – PINB, \$16(\$36). The Port B Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20 mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in Table 20.

Table 20. Port B Pin Alternate Functions

| Port Pin | Alternate Functions |
|----------|---|
| PB0 | T0 (Timer/Counter 0 External Counter Input) |
| PB1 | T1 (Timer/Counter 1 External Counter Input) |
| PB2 | AIN0 (Analog Comparator positive input) |
| PB3 | AIN1 (Analog Comparator negative input) |
| PB4 | \overline{SS} (SPI Slave Select Input) |
| PB5 | MOSI (SPI Bus Master Output/Slave Input) |
| PB6 | MISO (SPI Bus Master Input/Slave Output) |
| PB7 | SCK (SPI Bus Serial Clock) |

When the pins are used for the alternate function, the DDRB and PORTB registers have to be set according to the alternate function description.

Port B Data Register – PORTB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-------|
| \$18 (\$38) | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | PORTB |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port B Data Direction Register – DDRB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|
| \$17 (\$37) | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | DDRB |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port B Input Pins Address – PINB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|------|
| \$16 (\$36) | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | PINB |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | |

The Port B Input Pins address (PINB) is not a register; this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read and when reading PINB, the logical values present on the pins are read.

Figure 49. Port B Schematic Diagram (Pin PB5)

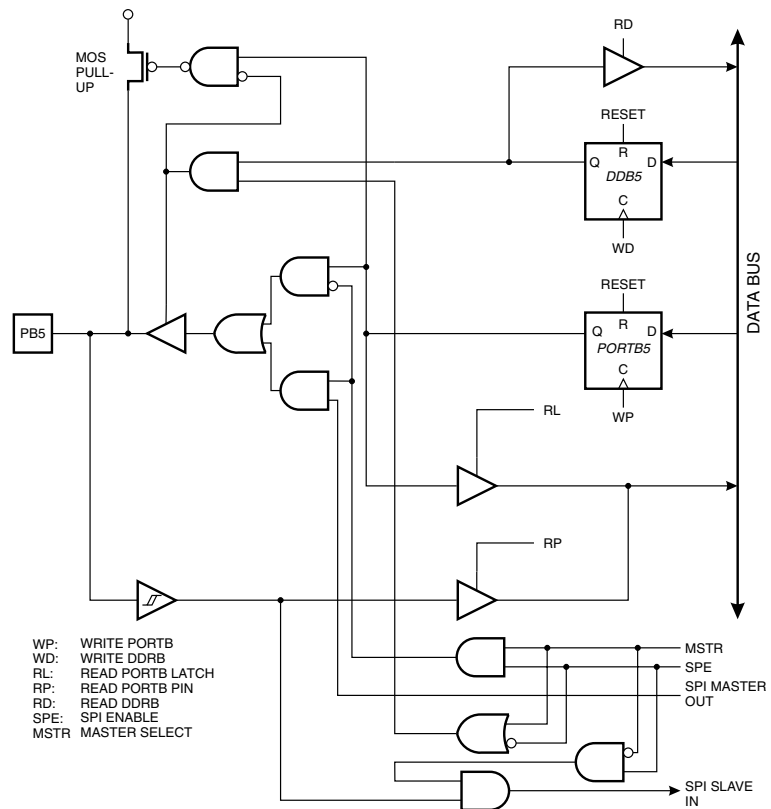


Figure 50. Port B Schematic Diagram (Pin PB6)

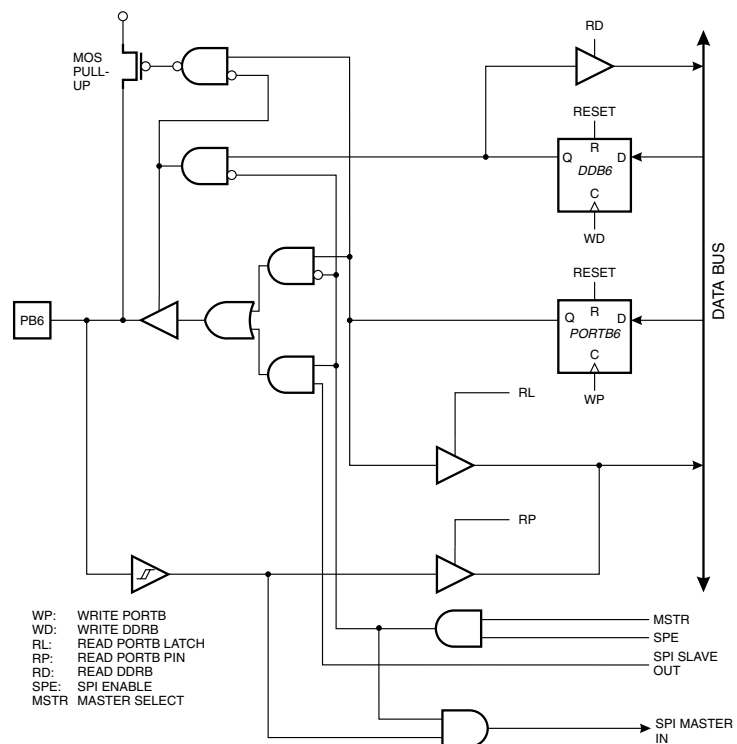


Figure 58. Port D Schematic Diagram (Pin PD6)

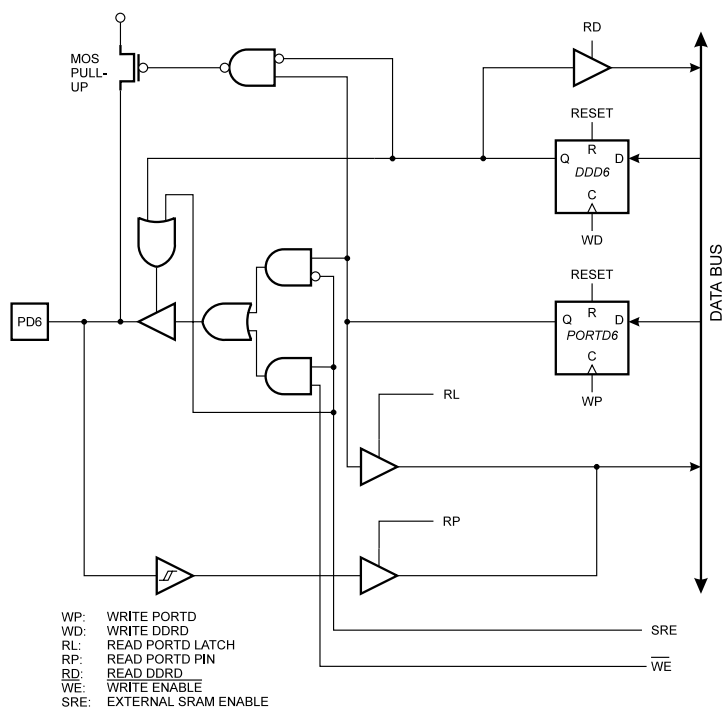
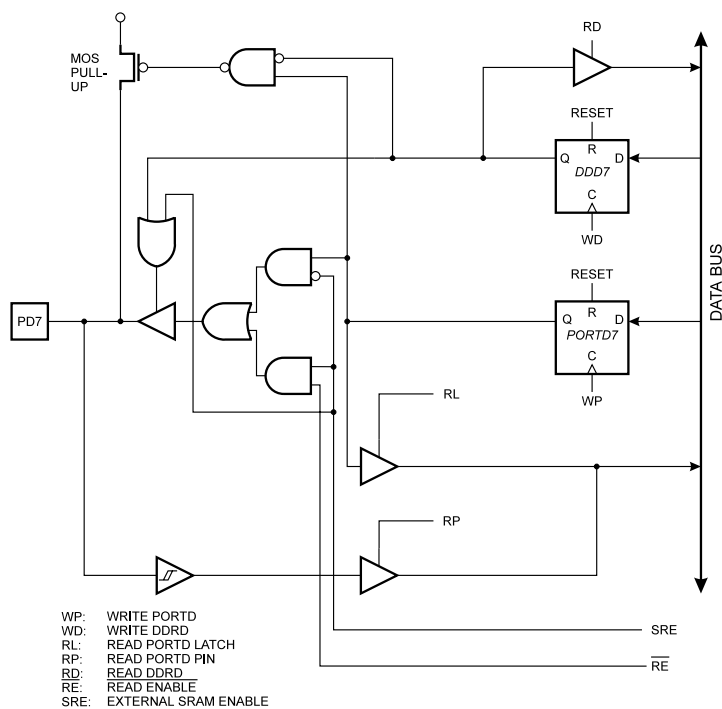


Figure 59. Port D Schematic Diagram (Pin PD7)



Memory Programming

Program and Data Memory Lock Bits

The AT90S8515 MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 25. The Lock bits can only be erased with the Chip Erase command.

Table 25. Lock Bit Protection Modes

| Memory Lock Bits | | | Protection Type |
|------------------|-----|-----|---|
| Mode | LB1 | LB2 | |
| 1 | 1 | 1 | No memory lock features enabled. |
| 2 | 0 | 1 | Further programming of the Flash and EEPROM is disabled. ⁽¹⁾ |
| 3 | 0 | 0 | Same as mode 2, and verify is also disabled. |

Note: 1. In Parallel Mode, further programming of the Fuse bits is also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The AT90S8515 has two Fuse bits, SPIEN and FSTRT.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading is enabled. Default value is programmed (“0”).
- When the FSTRT Fuse is programmed (“0”), the short start-up time is selected. Default value is unprogrammed (“1”). Parts with this bit pre-programmed (“0”) can be delivered on demand.

The Fuse bits are not accessible in Serial Programming Mode. The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a three-byte signature code that identifies the device. This code can be read in both Serial and Parallel mode. The three bytes reside in a separate address space.

For the AT90S8515⁽¹⁾ they are:

1. \$000: \$1E (indicates manufactured by Atmel)
2. \$001: \$93 (indicates 8 KB Flash memory)
3. \$002: \$01 (indicates AT90S8515 device when signature byte \$001 is \$93)

Note: 1. When both Lock bits are programmed (lock mode 3), the signature bytes cannot be read in Serial Mode. Reading the signature bytes will return: \$00, \$01 and \$02.

Programming the Flash and EEPROM

Atmel's AT90S8515 offers 8K bytes of In-System Reprogrammable Flash program memory and 512 bytes of EEPROM data memory.

The AT90S8515 is shipped with the On-chip Flash program and EEPROM data memory arrays in the erased state (i.e., contents = \$FF) and ready to be programmed. This device supports a high-voltage (12V) Parallel Programming Mode and a low-voltage Serial Programming Mode. The +12V is used for programming enable only, and no current of significance is drawn by this pin. The Serial Programming Mode provides a convenient way to download program and data into the AT90S8515 inside the user's system.

The program and data memory arrays on the AT90S8515 are programmed byte-by-byte in either programming mode. For the EEPROM, an auto-erase cycle is provided within

Table 27. Pin Name Mapping

| Signal Name in Programming Mode | Pin Name | I/O | Function |
|---------------------------------|----------|-----|---|
| RDY/BSY | PD1 | O | 0: Device is busy programming, 1: Device is ready for new command |
| \overline{OE} | PD2 | I | Output Enable (Active low) |
| \overline{WR} | PD3 | I | Write Pulse (Active low) |
| BS | PD4 | I | Byte Select ("0" selects low byte, "1" selects high byte) |
| XA0 | PD5 | I | XTAL Action Bit 0 |
| XA1 | PD6 | I | XTAL Action Bit 1 |
| DATA | PB7-0 | I/O | Bi-directional Data Bus (Output when \overline{OE} is low) |

Table 28. XA1 and XA0 Coding

| XA1 | XA0 | Action when XTAL1 is Pulsed |
|-----|-----|--|
| 0 | 0 | Load Flash or EEPROM Address (High or low address byte determined by BS) |
| 0 | 1 | Load Data (High or low data byte for Flash determined by BS) |
| 1 | 0 | Load Command |
| 1 | 1 | No Action, Idle |

Table 29. Command Byte Bit Coding

| Command Byte | Command Executed |
|--------------|-------------------------|
| 1000 0000 | Chip Erase |
| 0100 0000 | Write Fuse Bits |
| 0010 0000 | Write Lock Bits |
| 0001 0000 | Write Flash |
| 0001 0001 | Write EEPROM |
| 0000 1000 | Read Signature Bytes |
| 0000 0100 | Read Lock and Fuse Bits |
| 0000 0010 | Read Flash |
| 0000 0011 | Read EEPROM |

Enter Programming Mode

The following algorithm puts the device in Parallel Programming Mode:

1. Apply supply voltage according to Table 26, between V_{CC} and GND.
2. Set the \overline{RESET} and BS pin to "0" and wait at least 100 ns.
3. Apply 11.5 - 12.5V to \overline{RESET} . Any activity on BS within 100 ns after +12V has been applied to \overline{RESET} will cause the device to fail entering programming mode.

Chip Erase

The Chip Erase command will erase the Flash and EEPROM memories and the Lock bits. The Lock bits are not reset until the Flash and EEPROM have been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash or EEPROM is reprogrammed.

Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to "1000 0000". This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give \overline{WR} a t_{WLWH_CE} -wide negative pulse to execute Chip Erase. See Table 30 on page 85 for t_{WLWH_CE} value. Chip Erase does not generate any activity on the RDY/ \overline{BSY} pin.

Programming the Flash

A: Load Command "Write Flash"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to "0001 0000". This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

B: Load Address High Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "1". This selects high byte.
3. Set DATA = Address high byte (\$00 - \$0F).
4. Give XTAL1 a positive pulse. This loads the address high byte.

C: Load Address Low Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "0". This selects low byte.
3. Set DATA = Address low byte (\$00 - \$FF).
4. Give XTAL1 a positive pulse. This loads the address low byte.

D: Load Data Low Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data low byte (\$00 - \$FF).
3. Give XTAL1 a positive pulse. This loads the data low byte.

E: Write Data Low Byte

1. Set BS to "0". This selects low data.
2. Give \overline{WR} a negative pulse. This starts programming of the data byte. RDY/ \overline{BSY} goes low.
3. Wait until RDY/ \overline{BSY} goes high to program the next byte.

(See Figure 61 for signal waveforms.)

F: Load Data High Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data high byte (\$00 - \$FF).
3. Give XTAL1 a positive pulse. This loads the data high byte.

G: Write Data High Byte

Bit 5 = SPIEN Fuse bit

Bit 0 = FSTRT Fuse bit

Bit 7 - 6, 4 - 1 = "1". These bits are reserved and should be left unprogrammed ("1").

3. Give \overline{WR} a t_{WLWH_PFB} -wide negative pulse to execute the programming, t_{WLWH_PFB} is found in Table 30. Programming the Fuse bits does not generate any activity on the RDY/ \overline{BSY} pin.

Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" on page 81 for details on command and data loading):

1. A: Load Command "0010 0000".
2. D: Load Data Low Byte. Bit n = "0" programs the Lock bit.
Bit 2 = Lock Bit2
Bit 1 = Lock Bit1
Bit 7 - 3, 0 = "1". These bits are reserved and should be left unprogrammed ("1").
3. E: Write Data Low Byte.

The Lock bits can only be cleared by executing Chip Erase.

Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 81 for details on Command loading):

1. A: Load Command "0000 0100".
2. Set \overline{OE} to "0", and BS to "1". The status of the Fuse and Lock bits can now be read at DATA ("0" means programmed).
Bit 7 = Lock Bit1
Bit 6 = Lock Bit2
Bit 5 = SPIEN Fuse bit
Bit 0 = FSTRT Fuse bit
3. Set \overline{OE} to "1".

Observe that BS needs to be set to "1".

Reading the Signature Bytes

The algorithm for reading the signature bytes is as follows (refer to "Programming the Flash" on page 81 for details on command and address loading):

1. A: Load Command "0000 1000".
2. C: Load Address Low Byte (\$00 - \$02).
Set \overline{OE} to "0", and BS to "0". The selected signature byte can now be read at DATA.
3. Set \overline{OE} to "1".

Analog Comparator offset voltage is measured as absolute offset.

Figure 76. Analog Comparator Offset Voltage vs. Common Mode Voltage

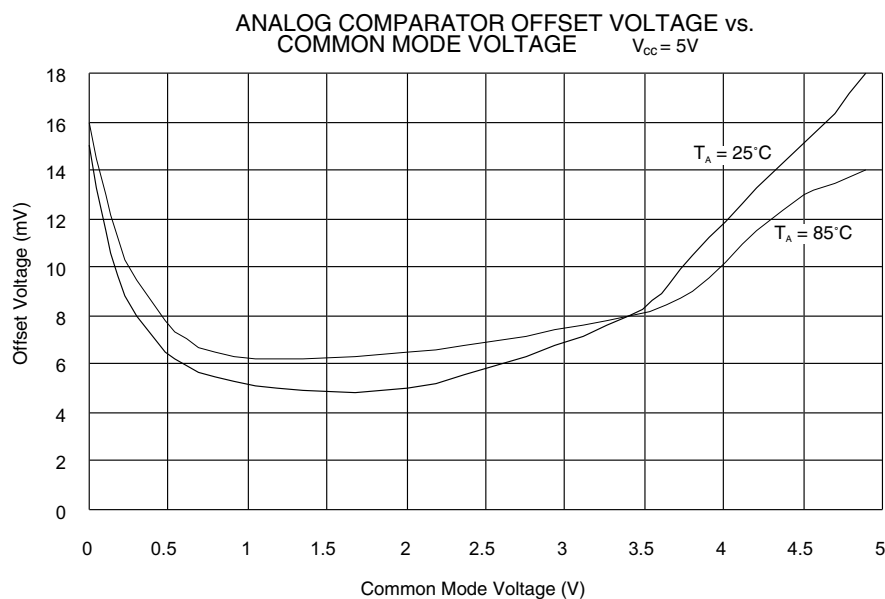


Figure 77. Analog Comparator Offset Voltage vs. Common Mode Voltage

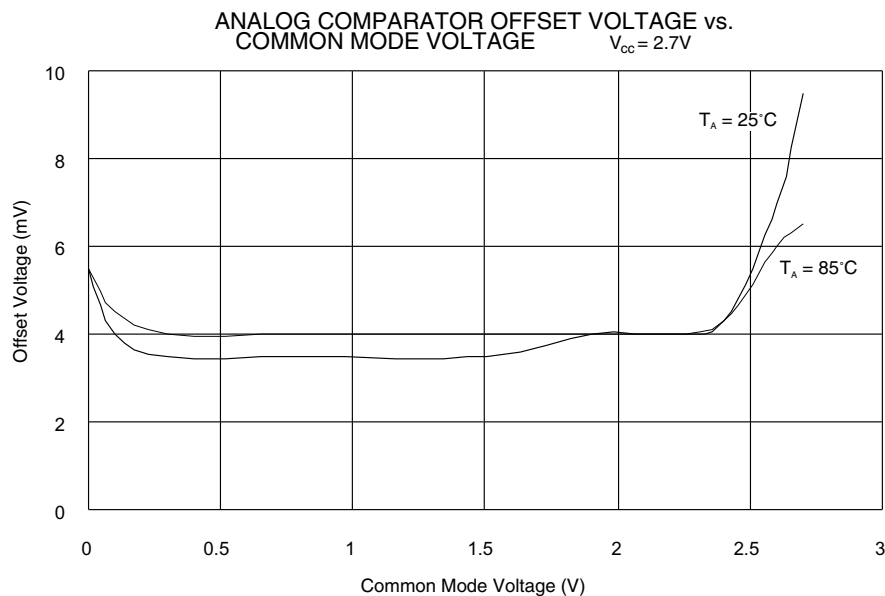


Figure 82. I/O Pin Sink Current vs. Output Voltage

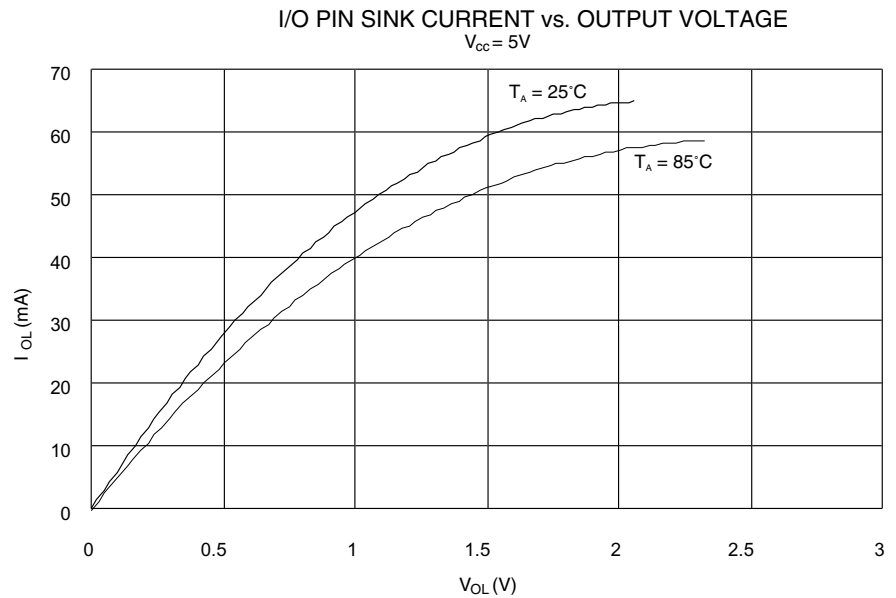
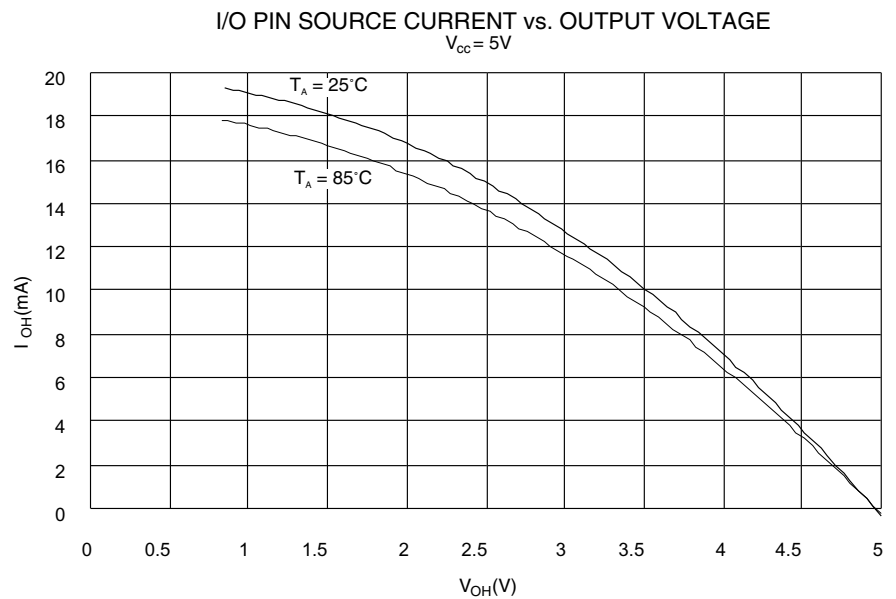


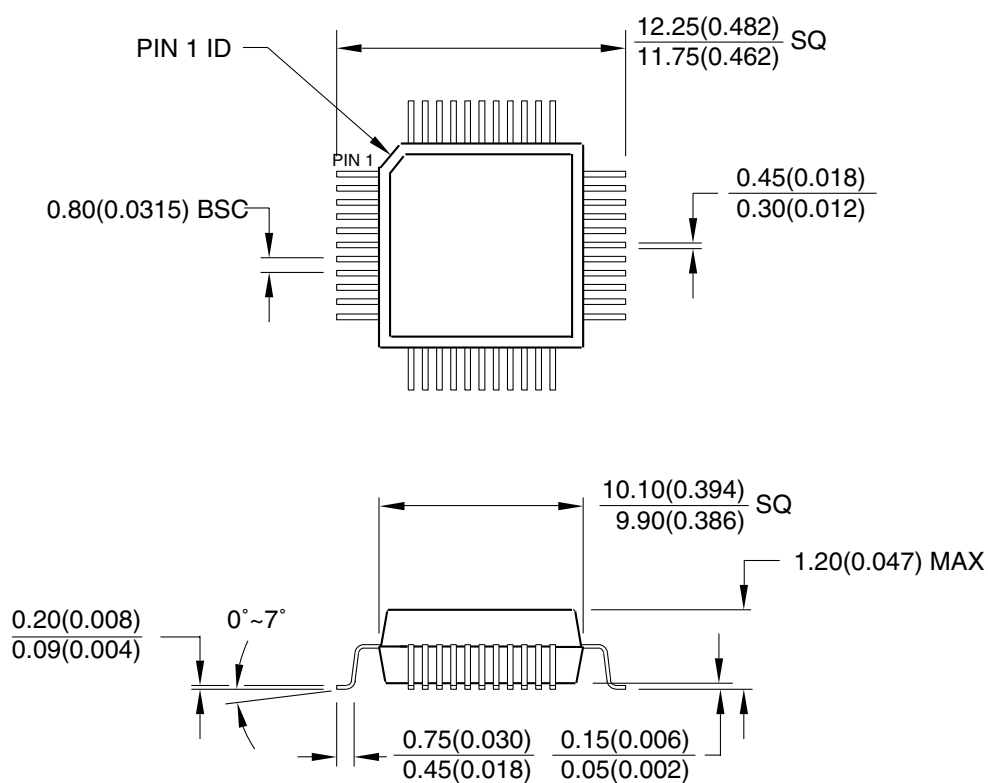
Figure 83. I/O Pin Source Current vs. Output Voltage



Packaging Information

44A

44-lead, Thin (1.0mm) Plastic Quad Flat Package
(TQFP), 10x10mm body, 2.0mm footprint, 0.8mm pitch.
Dimension in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB



*Controlling dimension: millimeter

REV. A 04/11/2001