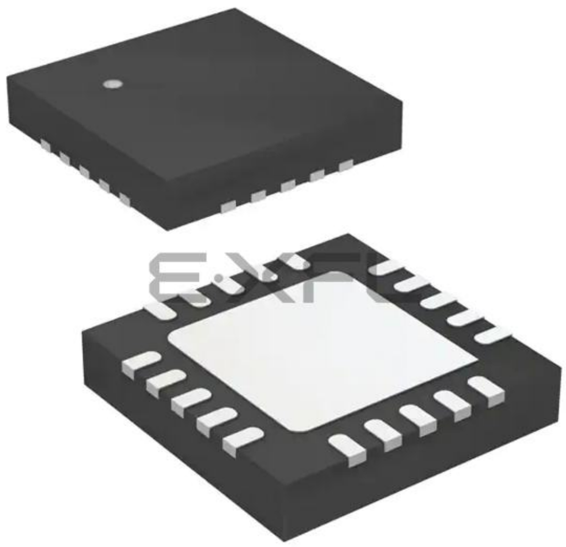




## What is "Embedded - Microcontrollers"?

"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 | Active |
| Core Processor | AVR |
| Core Size | 8-Bit |
| Speed | 12MHz |
| Connectivity | I²C, SPI, UART/USART, USI |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 18 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 12x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-WFQFN Exposed Pad |
| Supplier Device Package | 20-QFN-EP (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/attiny1634-mur |

# 8-bit Atmel tinyAVR Microcontroller with 16K Bytes In-System Programmable Flash

# ATtiny1634

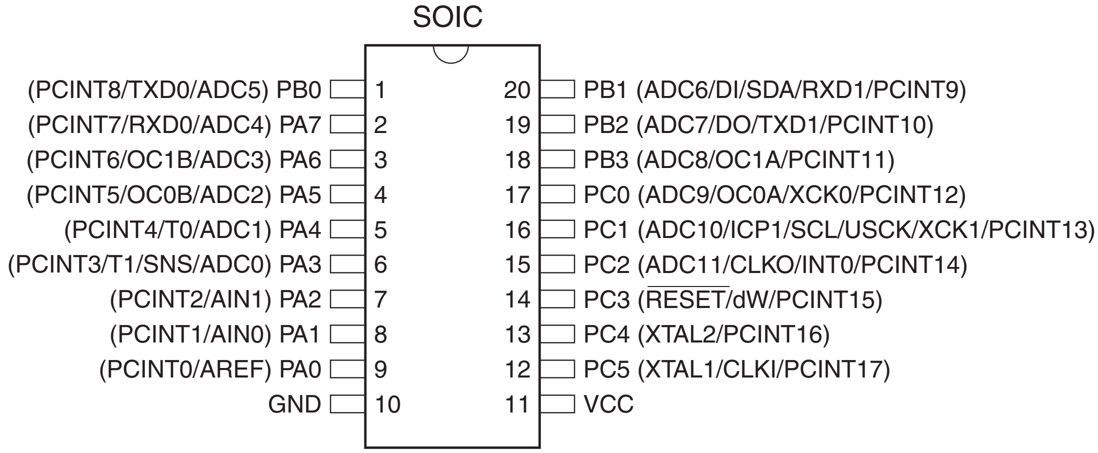
# Summary

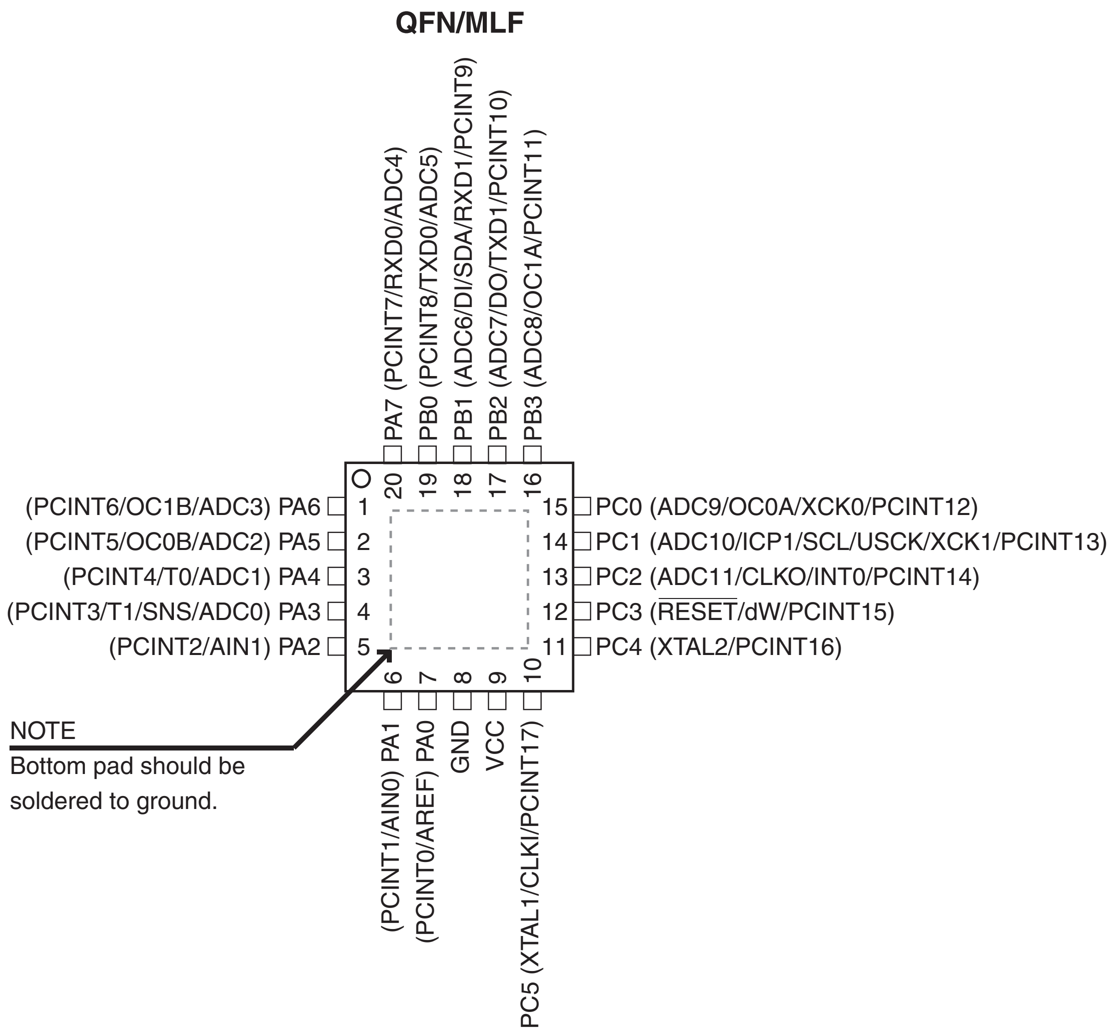
## Features

- High Performance, Low Power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 125 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
- High Endurance, Non-volatile Memory Segments
  - 16K Bytes of In-System, Self-Programmable Flash Program Memory
    - Endurance: 10,000 Write/Erase Cycles
  - 256 Bytes of In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 1K Byte of Internal SRAM
  - Data retention: 20 years at 85°C / 100 years at 25°C
  - Programming Lock for Self-Programming Flash & EEPROM Data Security
- Peripheral Features
  - Dedicated Hardware and QTouch® Library Support for Capacitive Touch Sensing
  - One 8-bit and One 16-bit Timer/Counter with Two PWM Channels, Each
  - 12-channel, 10-bit ADC
  - Programmable Ultra Low Power Watchdog Timer
  - On-chip Analog Comparator
  - Two Full Duplex USARTs with Start Frame Detection
  - Universal Serial Interface
  - Slave $I^2C$ Serial Interface
- Special Microcontroller Features
  - debugWIRE On-chip Debug System
  - In-System Programmable via SPI Port
  - Internal and External Interrupt Sources
    - Pin Change Interrupt on 18 Pins
  - Low Power Idle, ADC Noise Reduction, Standby and Power-down Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit with Supply Voltage Sampling
  - Calibrated 8MHz Oscillator with Temperature Calibration Option
  - Calibrated 32kHz Ultra Low Power Oscillator
  - On-chip Temperature Sensor
- I/O and Packages
  - 18 Programmable I/O Lines
  - 20-pad QFN/MLF, and 20-pin SOIC
- Operating Voltage:
  - 1.8 – 5.5V
- Speed Grade:
  - 0 – 2MHz @ 1.8 – 5.5V
  - 0 – 8MHz @ 2.7 – 5.5V
  - 0 – 12MHz @ 4.5 – 5.5V
- Temperature Range: -40°C to +105°C
- Low Power Consumption
  - Active Mode: 0.2mA at 1.8V and 1MHz
  - Idle Mode: 30µA at 1.8V and 1MHz
  - Power-Down Mode (WDT Enabled): 1µA at 1.8V
  - Power-Down Mode (WDT Disabled): 100nA at 1.8V

# 1. Pin Configurations

**Figure 1-1.** Pinout of ATtiny1634



## 1.1 Pin Descriptions

### 1.1.1 VCC

Supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 XTAL1

Input to the inverting amplifier of the oscillator and the internal clock circuit. This is an alternative pin configuration of PC5.

### 1.1.4 XTAL2

Output from the inverting amplifier of the oscillator. Alternative pin configuration of PC4.

### 1.1.5 $\overline{\text{RESET}}$

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 and provided the reset pin has not been disabled. The minimum pulse length is given in Table 24-5 on page 231. Shorter pulses are not guaranteed to generate a reset.

The reset pin can also be used as a (weak) I/O pin.

### 1.1.6 Port A (PA7:PA0)

This is an 8-bit, bi-directional I/O port with internal pull-up resistors (selected for each bit). Output buffers have the following drive characteristics:

- PA7, PA4:PA0: Symmetrical, with standard sink and source capability
- PA6, PA5: Asymmetrical, with high sink and standard source capability

As inputs, port pins that are externally pulled low will source current provided that pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port has alternate pin functions to serve special features of the device. See "Alternate Functions of Port A" on page 62.

### 1.1.7 Port B (PB3:PB0)

This is a 4-bit, bi-directional I/O port with internal pull-up resistors (selected for each bit).Output buffers have the following drive characteristics:

- PB3: Asymmetrical, with high sink and standard source capability
- PB2:PB0: Symmetrical, with standard sink and source capability

As inputs, port pins that are externally pulled low will source current provided that pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port has alternate pin functions to serve special features of the device. See "Alternate Functions of Port B" on page 65.

### 1.1.8 Port C (PC5:PC0)

This is a 6-bit, bi-directional I/O port with internal pull-up resistors (selected for each bit). Output buffers have the following drive characteristics:

- PC5:PC1: Symmetrical, with standard sink and source capability
- PC0: Asymmetrical, with high sink and standard source capability

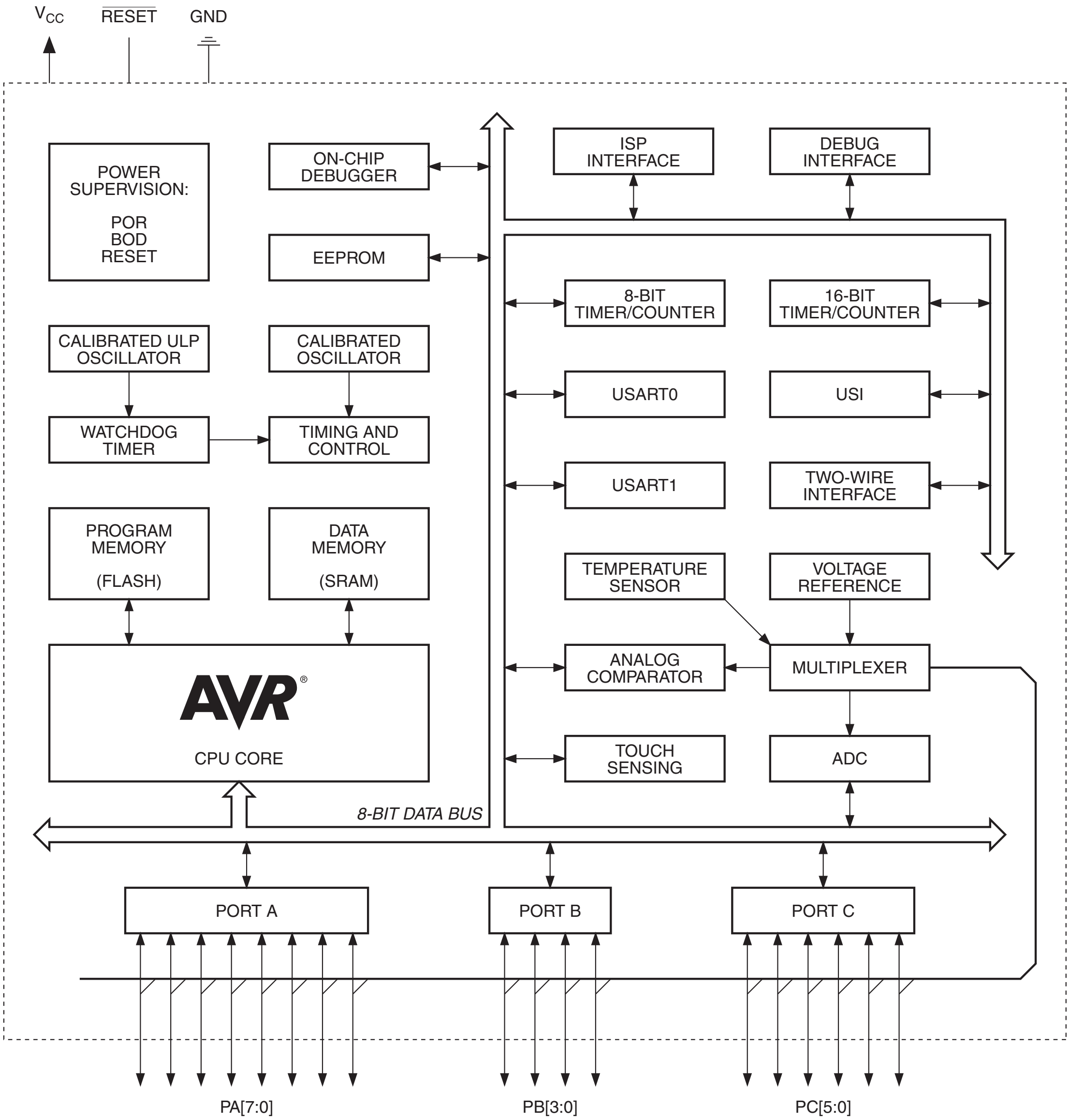
As inputs, port pins that are externally pulled low will source current provided that pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port has alternate pin functions to serve special features of the device. See "Alternate Functions of Port C" on page 67.

## 2. Overview

ATtiny1634 is a low-power CMOS 8-bit microcontrollers based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny1634 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

**Figure 2-1.** Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction, executed in one clock cycle. The resulting architecture is compact and code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

ATtiny1634 provides the following features:

- 16K bytes of in-system programmable Flash
- 1K bytes of SRAM data memory
- 256 bytes of EEPROM data memory
- 18 general purpose I/O lines
- 32 general purpose working registers
- An 8-bit timer/counter with two PWM channels
- A16-bit timer/counter with two PWM channels
- Internal and external interrupts
- A 10-bit ADC with 5 internal and 12 external channels
- An ultra-low power, programmable watchdog timer with internal oscillator
- Two programmable USART's with start frame detection
- A slave Two-Wire Interface (TWI)
- A Universal Serial Interface (USI) with start condition detector
- A calibrated 8MHz oscillator
- A calibrated 32kHz, ultra low power oscillator
- Four software selectable power saving modes.

The device includes the following modes for saving power:

- Idle mode: stops the CPU while allowing the timer/counter, ADC, analog comparator, SPI, TWI, and interrupt system to continue functioning
- ADC Noise Reduction mode: minimizes switching noise during ADC conversions by stopping the CPU and all I/O modules except the ADC
- Power-down mode: registers keep their contents and all chip functions are disabled until the next interrupt or hardware reset
- Standby mode: the oscillator is running while the rest of the device is sleeping, allowing very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash program memory can be re-programmed in-system through a serial interface, by a conventional non-volatile memory programmer or by an on-chip boot code, running on the AVR core.

The ATtiny1634 AVR is supported by a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators and evaluation kits.

# 3. General Information

## 3.1 Resources

A comprehensive set of drivers, application notes, data sheets and descriptions on development tools are available for download at http://www.atmel.com/avr.

## 3.2 Code Examples

This documentation 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 the 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, this means "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR". Note that not all AVR devices include an extended I/O map.

## 3.3 Capacitive Touch Sensing

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

Touch sensing is easily added to any application by linking the QTouch Library and using the Application Programming Interface (API) of the library to define the touch channels and sensors. The application then calls the API to retrieve channel information and determine the state of the touch sensor.

The QTouch Library is free and can be downloaded from the Atmel website. For more information and details of implementation, refer to the QTouch Library User Guide – also available from the Atmel website.

## 3.4 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.

# 4. CPU Core

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

# 5. Register Summary

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| (0xFF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF9) | Reserved | – | – | – | – | – | – | – | – | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| (0x85) | Reserved | – | – | – | – | – | – | – | – | |
| (0x84) | Reserved | – | – | – | – | – | – | – | – | |
| (0x83) | Reserved | – | – | – | – | – | – | – | – | |
| (0x82) | Reserved | – | – | – | – | – | – | – | – | |
| (0x81) | Reserved | – | – | – | – | – | – | – | – | |
| (0x80) | Reserved | – | – | – | – | – | – | – | – | |
| (0x7F) | TWSCRA | TWSHE | – | TWDIE | TWASIE | TWEN | TWSIE | TWPME | TWSME | 127 |
| (0x7E) | TWSCRB | | | | | | TWAA | TWCMD[1:0] | | 127 |
| (0x7D) | TWSSRA | TWDIF | TWASIF | TWCH | TWRA | TWC | TWBE | TWDIR | TWAS | 128 |
| (0x7C) | TWSA | TWI Slave Address Register | | | | | | | | 130 |
| (0x7B) | TWSAM | TWI Slave Address Mask Register | | | | | | | | 130 |
| (0x7A) | TWSD | TWI Slave Data Register | | | | | | | | 130 |
| (0x79) | UCSR1A | RXC1 | TXC1 | UDRE1 | FE1 | DOR1 | UPE1 | U2X1 | MPCM1 | 167 |
| (0x78) | UCSR1B | RXCIE1 | TXCIE1 | UDRIE1 | RXEN1 | TXEN1 | UCSZ12 | RXB81 | TXB81 | 168 |
| (0x77) | UCSR1C | UMSEL11 | UMSEL10 | UPM11 | UPM01 | USBS1 | UCSZ11 | UCSZ10 | UCPOL1 | 169 |
| (0x76) | UCSR1D | RXSIE1 | RXS1 | SFDE1 | | | | | | 171 |
| (0x75) | UBRR1H | USART1 Baud Rate Register High Byte | | | | | | | | 172 |
| (0x74) | UBRR1L | USART1 Baud Rate Register Low Byte | | | | | | | | 172 |
| (0x73) | UDR1 | USART1 I/O Data Register | | | | | | | | 167 |
| (0x72) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | – | – | WGM11 | WGM10 | 111 |
| (0x71) | TCCR1B | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | 113 |
| (0x70) | TCCR1C | FOC1A | FOC1B | – | – | – | – | – | – | 114 |
| (0x6F) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | 114 |
| (0x6E) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | 114 |
| (0x6D) | OCR1AH | Timer/Counter1 – Compare Register A High Byte | | | | | | | | 114 |
| (0x6C) | OCR1AL | Timer/Counter1 – Compare Register A Low Byte | | | | | | | | 114 |
| (0x6B) | OCR1BH | Timer/Counter1 – Compare Register B High Byte | | | | | | | | 115 |
| (0x6A) | OCR1BL | Timer/Counter1 – Compare Register B Low Byte | | | | | | | | 115 |
| (0x69) | ICR1H | Timer/Counter1 – Input Capture Register High Byte | | | | | | | | 115 |
| (0x68) | ICR1L | Timer/Counter1 – Input Capture Register Low Byte | | | | | | | | 115 |
| (0x67) | GTCCR | TSM | – | – | – | – | – | – | PSR10 | 118 |
| (0x66) | OSCCAL1 | – | – | – | – | – | – | CAL11 | CAL10 | 33 |
| (0x65) | OSCTCAL0B | Oscillator Temperature Compensation Register B | | | | | | | | 33 |
| (0x64) | OSCTCAL0A | Oscillator Temperature Compensation Register A | | | | | | | | 33 |
| (0x63) | OSCCAL0 | CAL07 | CAL06 | CAL05 | CAL04 | CAL03 | CAL02 | CAL01 | CAL00 | 32 |
| (0x62) | DIDR2 | – | – | – | – | – | ADC11D | ADC10D | ADC9D | 200 |
| (0x61) | DIDR1 | – | – | – | – | ADC8D | ADC7D | ADC6D | ADC5D | 200 |
| (0x60) | DIDR0 | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D | AIN1D | AIN0D | AREFD | 184, 200 |
| 0x3F (0x5F) | SREG | I | T | H | S | V | N | Z | C | 14 |
| 0x3E (0x5E) | SPH | – | – | – | – | – | SP10 | SP9 | SP8 | 13 |
| 0x3D (0x5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 13 |
| 0x3C (0x5C) | GIMSK | – | INT0 | PCIE2 | PCIE1 | PCIE0 | – | – | – | 51 |
| 0x3B (0x5B) | GIFR | – | INTF0 | PCIF2 | PCIF1 | PCIF0 | – | – | – | 52 |
| 0x3A (0x5A) | TIMSK | TOIE1 | OCIE1A | OCIE1B | – | ICIE1 | OCIE0B | TOIE0 | OCIE0A | 88, 115 |
| 0x39 (0x59) | TIFR | TOV1 | OCF1A | OCF1B | – | ICF1 | OCF0B | TOV0 | OCF0A | 89, 116 |
| 0x38 (0x58) | QTCSR | QTouch Control and Status Register | | | | | | | | 6 |
| 0x37 (0x57) | SPMCSR | – | – | RSIG | CTPB | RFLB | PGWRT | PGERS | SPMEN | 207 |
| 0x36 (0x56) | MCUCR | – | SM1 | SM0 | SE | – | – | ISC01 | ISC00 | 37, 51 |
| 0x35 (0x55) | MCUSR | – | – | – | – | WDRF | BORF | EXTRF | PORF | 44 |
| 0x34 (0x54) | PRR | – | PRTWI | PRTIM0 | PRTIM0 | PRUSI | PRUSART1 | PRUSART0 | PRADC | 38 |
| 0x33 (0x53) | CLKPR | – | – | – | – | CLKPS3 | CLKPS2 | CLKPS1 | CLKPS0 | 31 |
| 0x32 (0x52) | CLKSR | OSCRDY | CSTR | CKOUT_IO | SUT | CKSEL3 | CKSEL2 | CKSEL1 | CKSEL0 | 29 |
| 0x31 (0x51) | Reserved | – | – | – | – | – | – | – | – | |
| 0x30 (0x50) | WDTCSR | WDIF | WDIE | WDP3 | – | WDE | WDP2 | WDP1 | WDP0 | 45 |
| 0x2F (0x4F) | CCP | CPU Change Protection Register | | | | | | | | 13 |
| 0x2E (0x4E) | DWDR | DWDR[7:0] | | | | | | | | 202 |
| 0x2D (0x4D) | USIBR | USI Buffer Register | | | | | | | | 144 |
| 0x2C (0x4C) | USIDR | USI Data Register | | | | | | | | 143 |

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x2B (0x4B) | USISR | USISIF | USIOIF | USIPF | USIDC | USICNT3 | USICNT2 | USICNT1 | USICNT0 | 142 |
| 0x2A (0x4A) | USICR | USISIE | USIOIE | USIWM1 | USIWM0 | USICS1 | USICS0 | USICLK | USITC | 140 |
| 0x29 (0x49) | PCMSK2 | – | – | PCINT17 | PCINT16 | PCINT15 | PCINT14 | PCINT13 | PCINT12 | 52 |
| 0x28 (0x48) | PCMSK1 | – | – | – | – | PCINT11 | PCINT10 | PCINT9 | PCINT8 | 53 |
| 0x27 (0x47) | PCMSK0 | PCINT7 | PCINT6 | PCINT5 | PCINT4 | PCINT3 | PCINT2 | PCINT1 | PCINT0 | 53 |
| 0x26 (0x46) | UCSR0A | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM | 167 |
| 0x25 (0x45) | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 | 168 |
| 0x24 (0x44) | UCSR0C | UMSEL01 | UMSEL00 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 | UCPOL0 | 169 |
| 0x23 (0x43) | UCSR0D | RXCIE0 | RXS0 | SFDE0 | – | – | – | – | – | 171 |
| 0x22 (0x42) | UBRR0H | – | – | – | – | USART0 Baud Rate Register High Byte | | | | 172 |
| 0x21 (0x41) | UBRR0L | USART0 Baud Rate Register Low Byte | | | | | | | | 172 |
| 0x20 (0x40) | UDR0 | USART0 I/O Data Register | | | | | | | | 167 |
| 0x1F (0x3F) | EEARH | – | – | – | – | – | – | – | – | |
| 0x1E (0x3E) | EEARL | EEAR[7:0] | | | | | | | | 22 |
| 0x1D (0x3D) | EEDR | EEPROM Data Register | | | | | | | | 22 |
| 0x1C (0x3C) | EECR | – | – | EEPM1 | EEPM0 | EERIE | EEMPE | EEPE | EERE | 22 |
| 0x1B (0x3B) | TCCR0A | COM0A1 | COM0A0 | COM0B1 | COM0B0 | – | – | WGM01 | WGM00 | 84 |
| 0x1A (0x3A) | TCCR0B | FOC0A | FOC0B | – | – | WGM02 | CS02 | CS01 | CS00 | 86 |
| 0x19 (0x39) | TCNT0 | Timer/Counter0 | | | | | | | | 88 |
| 0x18 (0x38) | OCR0A | Timer/Counter0 – Compare Register A | | | | | | | | 88 |
| 0x17 (0x37) | OCR0B | Timer/Counter0 – Compare Register B | | | | | | | | 88 |
| 0x16 (0x36) | GPIOR2 | General Purpose Register 2 | | | | | | | | 23 |
| 0x15 (0x35) | GPIOR1 | General Purpose Register 1 | | | | | | | | 24 |
| 0x14 (0x34) | GPIOR0 | General Purpose Register 0 | | | | | | | | 24 |
| 0x13 (0x33) | PORTCR | – | – | – | – | – | BBMC | BBMB | BBMA | 71 |
| 0x12 (0x32) | PUEA | PUEA7 | PUEA6 | PUEA5 | PUEA4 | PUEA3 | PUEA2 | PUEA1 | PUEA0 | 71 |
| 0x11 (0x31) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 71 |
| 0x10 (0x30) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 71 |
| 0x0F (0x2F) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 71 |
| 0x0E (0x2E) | PUEB | – | – | – | – | PUEB3 | PUEB2 | PUEB1 | PUEB0 | 72 |
| 0x0D (0x2D) | PORTB | – | – | – | – | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 72 |
| 0x0C (0x2C) | DDRB | – | – | – | – | DDB3 | DDB2 | DDB1 | DDB0 | 72 |
| 0x0B (0x2B) | PINB | – | – | – | – | PINB3 | PINB2 | PINB1 | PINB0 | 72 |
| 0x0A (0x2A) | PUEC | – | – | PUEC5 | PUEC4 | PUEC3 | PUEC2 | PUEC1 | PUEC0 | 72 |
| 0x09 (0x29) | PORTC | – | – | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 72 |
| 0x08 (0x28) | DDRC | – | – | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 72 |
| 0x07 (0x27) | PINC | – | – | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 72 |
| 0x06 (0x26) | ACSRA | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 182 |
| 0x05 (0x25) | ACSRB | HSEL | HLEV | ACLP | – | ACCE | ACME | ACIRS1 | ACIRS0 | 183 |
| 0x04 (0x24) | ADMUX | REFS1 | REFS0 | REFEN | ADC0EN | MUX3 | MUX2 | MUX1 | MUX0 | 196 |
| 0x03 (0x23) | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 197 |
| 0x02 (0x22) | ADCSRB | VDEN | VDPD | – | – | ADLAR | ADTS2 | ADTS1 | ADTS0 | 199 |
| 0x01 (0x21) | ADCH | ADC Data Register High Byte | | | | | | | | 198 |
| 0x00 (0x20) | ADCL | ADC Data Register Low Byte | | | | | | | | 198 |

Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operation the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

# 6. Instruction Set Summary

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| **ARITHMETIC AND LOGIC INSTRUCTIONS** | | | | | |
| ADD | Rd, Rr | Add two Registers | Rd ← Rd + Rr | Z,C,N,V,H | 1 |
| ADC | Rd, Rr | Add with Carry two Registers | Rd ← Rd + Rr + C | Z,C,N,V,H | 1 |
| ADIW | Rdl,K | Add Immediate to Word | Rdh:Rdl ← Rdh:Rdl + K | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Subtract two Registers | Rd ← Rd - Rr | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract Constant from Register | Rd ← Rd - K | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with Carry two Registers | Rd ← Rd - Rr - C | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with Carry Constant from Reg. | Rd ← Rd - K - C | Z,C,N,V,H | 1 |
| SBIW | Rdl,K | Subtract Immediate from Word | Rdh:Rdl ← Rdh:Rdl - K | Z,C,N,V,S | 2 |
| AND | Rd, Rr | Logical AND Registers | Rd ← Rd • Rr | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND Register and Constant | Rd ← Rd • K | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR Registers | Rd ← Rd v Rr | Z,N,V | 1 |
| ORI | Rd, K | Logical OR Register and Constant | Rd ← Rd v K | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR Registers | Rd ← Rd ⊕ Rr | Z,N,V | 1 |
| COM | Rd | One's Complement | Rd ← 0xFF − Rd | Z,C,N,V | 1 |
| NEG | Rd | Two's Complement | Rd ← 0x00 − Rd | Z,C,N,V,H | 1 |
| SBR | Rd,K | Set Bit(s) in Register | Rd ← Rd v K | Z,N,V | 1 |
| CBR | Rd,K | Clear Bit(s) in Register | Rd ← Rd • (0xFF - K) | Z,N,V | 1 |
| INC | Rd | Increment | Rd ← Rd + 1 | Z,N,V | 1 |
| DEC | Rd | Decrement | Rd ← Rd − 1 | Z,N,V | 1 |
| TST | Rd | Test for Zero or Minus | Rd ← Rd • Rd | Z,N,V | 1 |
| CLR | Rd | Clear Register | Rd ← Rd ⊕ Rd | Z,N,V | 1 |
| SER | Rd | Set Register | Rd ← 0xFF | None | 1 |
| **BRANCH INSTRUCTIONS** | | | | | |
| JMP | k | Direct Jump | PC ← k | None | 3 |
| RJMP | k | Relative Jump | PC ← PC + k + 1 | None | 2 |
| IJMP | | Indirect Jump to (Z) | PC ← Z | None | 2 |
| CALL | k | Direct Subroutine | PC ← k | None | 4 |
| RCALL | k | Relative Subroutine Call | PC ← PC + k + 1 | None | 3 |
| ICALL | | Indirect Call to (Z) | PC ← Z | None | 3 |
| RET | | Subroutine Return | PC ← STACK | None | 4 |
| RETI | | Interrupt Return | PC ← STACK | I | 4 |
| CPSE | Rd,Rr | Compare, Skip if Equal | if (Rd = Rr) PC ← 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 ← PC + 2 or 3 | None | 1/2/3 |
| SBRS | Rr, b | Skip if Bit in Register is Set | if (Rr(b)=1) PC ← PC + 2 or 3 | None | 1/2/3 |
| SBIC | P, b | Skip if Bit in I/O Register Cleared | if (P(b)=0) PC ← 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 ← PC + 2 or 3 | None | 1/2/3 |
| BRBS | s, k | Branch if Status Flag Set | if (SREG(s) = 1) then PC←PC+k + 1 | None | 1/2 |
| BRBC | s, k | Branch if Status Flag Cleared | if (SREG(s) = 0) then PC←PC+k + 1 | None | 1/2 |
| BREQ | k | Branch if Equal | if (Z = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRNE | k | Branch if Not Equal | if (Z = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRCS | k | Branch if Carry Set | if (C = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRCC | k | Branch if Carry Cleared | if (C = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRSH | k | Branch if Same or Higher | if (C = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRLO | k | Branch if Lower | if (C = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRMI | k | Branch if Minus | if (N = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRPL | k | Branch if Plus | if (N = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRGE | k | Branch if Greater or Equal, Signed | if (N ⊕ V= 0) then PC ← PC + k + 1 | None | 1/2 |
| BRLT | k | Branch if Less Than Zero, Signed | if (N ⊕ V= 1) then PC ← PC + k + 1 | None | 1/2 |
| BRHS | k | Branch if Half Carry Flag Set | if (H = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRHC | k | Branch if Half Carry Flag Cleared | if (H = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRTS | k | Branch if T Flag Set | if (T = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRTC | k | Branch if T Flag Cleared | if (T = 0) then PC ← PC + k + 1 | None | 1/2 |
| BRVS | k | Branch if Overflow Flag is Set | if (V = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if (V = 0) then PC ← PC + k + 1 | None | 1/2 |
| 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 |
| **BIT AND BIT-TEST INSTRUCTIONS** | | | | | |
| 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 |

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| 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 |
| SEV | | Set Twos Complement Overflow. | V ← 1 | V | 1 |
| CLV | | Clear Twos Complement Overflow | V ← 0 | V | 1 |
| SET | | Set T in SREG | T ← 1 | T | 1 |
| CLT | | Clear T in SREG | T ← 0 | T | 1 |
| SEH | | Set Half Carry Flag in SREG | H ← 1 | H | 1 |
| CLH | | Clear Half Carry Flag in SREG | H ← 0 | H | 1 |
| **DATA TRANSFER INSTRUCTIONS** | | | | | |
| 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 |
| 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 |
| **MCU CONTROL INSTRUCTIONS** | | | | | |
| 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 |

# 7. Ordering Information

## 7.1 ATtiny1634

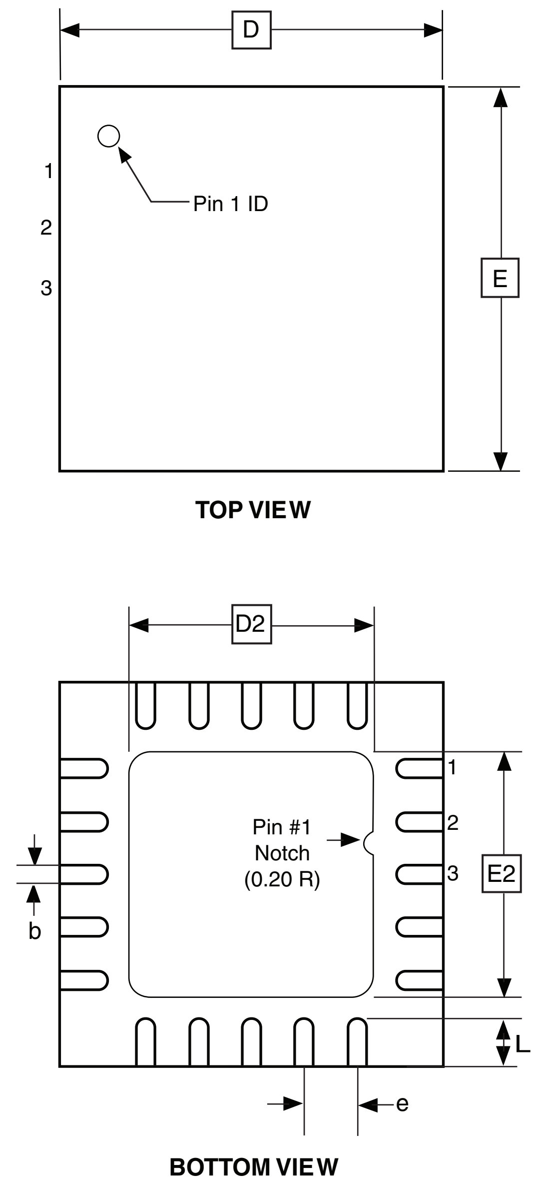
| Speed (MHz) [1] | Supply Voltage (V) | Temperature Range | Package [2] | Accuracy [3] | Ordering Code [4] |
|---|---|---|---|---|---|
| 12 | 1.8 – 5.5 | Industrial (-40°C to +85°C)[5] | 20M1 | ±10% | ATtiny1634-MU |
| | | | | ±2% | ATtiny1634R-MU |
| | | | | ±10% | ATtiny1634-MUR |
| | | | | ±2% | ATtiny1634R-MUR |
| | | | 20S2 | ±10% | ATtiny1634-SU |
| | | | | ±2% | ATtiny1634R-SU |
| | | | | ±10% | ATtiny1634-SUR |
| | | | | ±2% | ATtiny1634R-SUR |
| | | Extended (-40°C to +105°C)[5] | 20M1 | ±10% | ATtiny1634-MN |
| | | | | ±10% | ATtiny1634-MNR |

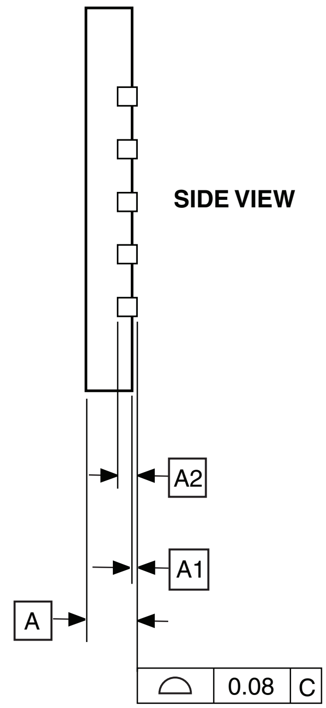
Notes:
1. For speed vs. supply voltage, see section 24.3 “Speed” on page 229.
2. All packages are Pb-free, halide-free and fully green, and they comply with the European directive for Restriction of Hazardous Substances (RoHS).
3. Denotes accuracy of the internal oscillator. See Table 24-2 on page 230.
4. Code indicators:
   - U: matte tin
   - R: tape & reel
5. Can also be supplied in wafer form. Contact your local Atmel sales office for ordering information and minimum quantities.

| Package Type | |
|---|---|
| **20M1** | 20-pad, 4 x 4 x 0.8 mm Body, Quad Flat No-Lead / Micro Lead Frame Package (QFN/MLF) |
| **20S2** | 20-lead, 0.300" Wide Body, Plastic Gull Wing Small Outline Package (SOIC) |

# 8. Packaging Information

## 8.1 20M1




**COMMON DIMENSIONS**
(Unit of Measure = mm)

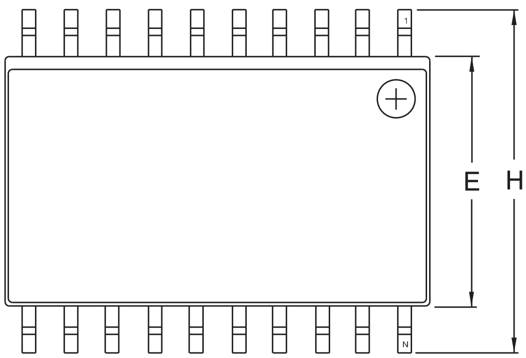
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.70 | 0.75 | 0.80 | |
| A1 | – | 0.01 | 0.05 | |
| A2 | | 0.20 REF | | |
| b | 0.18 | 0.23 | 0.30 | |
| D | | 4.00 BSC | | |
| D2 | 2.45 | 2.60 | 2.75 | |
| E | | 4.00 BSC | | |
| E2 | 2.45 | 2.60 | 2.75 | |
| e | | 0.50 BSC | | |
| L | 0.35 | 0.40 | 0.55 | |

Note: Reference JEDEC Standard MO-220, Fig. 1 (SAW Singulation) WGGD-5.

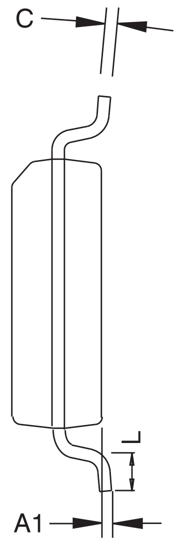
12/02/2014

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| Atmel 2325 Orchard Parkway San Jose, CA 95131 | **20M1**, 20-pad, 4 x 4 x 0.8 mm Body, Lead Pitch 0.50 mm, 2.6 mm Exposed Pad, Micro Lead Frame Package (MLF) | 20M1 | B |

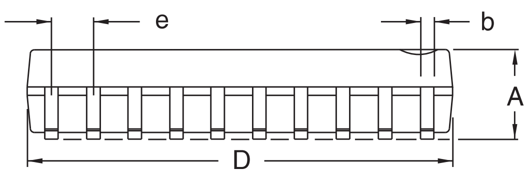
## 8.2 20S2




Top View




End View




Side View

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 2.35 | | 2.65 | |
| A1 | 0.10 | | 0.30 | |
| b | 0.33 | | 0.51 | 4 |
| C | 0.23 | | 0.32 | |
| D | 12.60 | | 13.00 | 1 |
| E | 7.40 | | 7.60 | 2 |
| H | 10.00 | | 10.65 | |
| L | 0.40 | | 1.27 | 3 |
| e | 1.27 BSC | | | |

Notes.
1. This drawing is for general information only; refer to JEDEC Drawing MS-013, Variation AC for additional information.
2. Dimension 'D' does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006') per side.
3. Dimension 'E' does not include inter-lead Flash or protrusion. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010') per side.
4. 'L' is the length of the terminal for soldering to a substrate.
5. The lead width 'b', as measured 0.36 mm (0.014') or greater above the seating plane, shall not exceed a maximum value of 0.61 mm (0.024') per side.

11/6/06

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| Atmel 2325 Orchard Parkway San Jose, CA 95131 | **20S2**, 20-lead, 0.300' Wide Body, Plastic Gull Wing Small Outline Package (SOIC) | 20S2 | B |

# 9. Errata

The revision letters in this section refer to the revision of the corresponding ATtiny1634 device.

## 9.1 ATtiny1634

### 9.1.1 Rev. C

- **Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

**1. Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

Port pin PB3 is not guaranteed to perform as a reliable input when the Ultra Low Power (ULP) oscillator is not running. In addition, the pin is pulled down internally when ULP oscillator is disabled.

**Problem Fix / Workaround**

The ULP oscillator is automatically activated when required. To use PB3 as an input, activate the watchdog timer. The watchdog timer automatically enables the ULP oscillator.

### 9.1.2 Rev. B

- **Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

**1. Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

Port pin PB3 is not guaranteed to perform as a reliable input when the Ultra Low Power (ULP) oscillator is not running. In addition, the pin is pulled down internally when ULP oscillator is disabled.

**Problem Fix / Workaround**

The ULP oscillator is automatically activated when required. To use PB3 as an input, activate the watchdog timer. The watchdog timer automatically enables the ULP oscillator.

### 9.1.3 Rev. A

- **Flash / EEPROM Can Not Be Written When Supply Voltage Is Below 2.4V**
- **Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

**1. Flash / EEPROM Can Not Be Written When Supply Voltage Is Below 2.4V**

When supply voltage is below 2.4V write operations to Flash and EEPROM may fail.

**Problem Fix / Workaround**

Do not write to Flash or EEPROM when supply voltage is below 2.4V.

**2. Port Pin Should Not Be Used As Input When ULP Oscillator Is Disabled**

Port pin PB3 is not guaranteed to perform as a reliable input when the Ultra Low Power (ULP) oscillator is not running. In addition, the pin is pulled down internally when ULP oscillator is disabled.

**Problem Fix / Workaround**

The ULP oscillator is automatically activated when required. To use PB3 as an input, activate the watchdog timer. The watchdog timer automatically enables the ULP oscillator.

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**