

## Assembly Code Example<sup>(1)</sup>

```

USART_Receive:
    ; Wait for data to be received
    sbis UCSRnA, RXCn
    rjmp USART_Receive
    ; Get status and 9th bit, then data from buffer
    in    r18, UCSRnA
    in    r17, UCSRnB
    in    r16, UDRn
    ; If error, return -1
    andi r18, (1<<FEn) | (1<<DORn) | (1<<UPEn)
    breq USART_ReceiveNoError
    ldi   r17, HIGH(-1)
    ldi   r16, LOW(-1)
USART_ReceiveNoError:
    ; Filter the 9th bit, then return
    lsr   r17
    andi r17, 0x01
    ret

```

## C Code Example<sup>(1)</sup>

```

unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXCn)) )
        ;
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRnA;
    resh = UCSRnB;
    resl = UDRn;
    /* If error, return -1 */
    if ( status & (1<<FEn) | (1<<DORn) | (1<<UPEn) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}

```

Note: 1. See "Code Examples" on page 7.

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

The receive function example reads all the I/O Registers into the Register File before any computation is done. This gives an optimal receive buffer utilization since the buffer location read will be free to accept new data as early as possible.