Implementation and Assembly Code Examples

This page contains basic implementation examples of USART serial communication using the Assembly programming language on an AVR Microcontroller (ATmega328p).

1. Printing Text to Serial Monitor

This code is used to repeatedly send the string "Programming Serial Interface!" to the Serial Monitor via the USART port.

```
;------------------------
; Assembly Code - Print Text
;------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main

main:
    CLR   R24
    STS   UCSR0A, R24               ; Clear UCSR0A register
    STS   UBRR0H, R24               ; Clear UBRR0H
    LDI   R24, 103                  ; Set UBRR value = 103 (9600 Baud Rate)
    STS   UBRR0L, R24

    LDI   R24, (1<<RXEN0) | (1<<TXEN0) ; Enable RX and TX
    STS   UCSR0B, R24

    LDI   R24, (1<<UCSZ00) | (1<<UCSZ01) ; Mode: 8-bit data, 1 stop bit, No Parity
    STS   UCSR0C, R24

print_msg:
    LDI   R30, lo8(message)
    LDI   R31, hi8(message)         ; Z points to string message
agn:
```

```
    LPM   R18, Z+                   ; Load character into R18
    CPI   R18, 0                    ; Check if end of string (null)
    BREQ  ext                       ; If yes, exit loop

l1:
    LDS   R17, UCSR0A
    SBRS  R17, UDRE0                ; Wait until buffer is empty (UDRE0=1)
    RJMP  l1
    STS   UDR0, R18                 ; Send character to Serial Monitor
    RJMP  agn                       ; Loop to next character

ext:
    RCALL delay_sec                 ; Wait for a moment
    RJMP  print_msg                 ; Repeat string transmission

message:
    .ascii "Programming Serial Interface!"
    .byte 10, 13, 0

delay_sec:                          ; Delay Subroutine (~3 seconds)
    LDI   R20, 255
l4: LDI   R21, 255
l5: LDI   R22, 255
l6: DEC   R22
    BRNE  l6
    DEC   R21
    BRNE  l5
    DEC   R20
    BRNE  l4
    RET
```

---

2. Reading Input from Serial Monitor

This code reads characters sent from the Serial Monitor and controls an LED. If the character 'H' is received, the LED turns ON; if 'L' is received, the LED turns OFF.

```
;------------------------
; Assembly Code - Input Text and Control LED
;------------------------
```

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main

main:
    CLR   R24
    STS   UBRR0H, R24
    LDI   R24, 103
    STS   UBRR0L, R24
    LDI   R24, (1<<RXEN0 | 1<<TXEN0)
    STS   UCSR0B, R24
    LDI   R24, (1<<UCSZ01 | 1<<UCSZ00)
    STS   UCSR0C, R24

    SBI   DDRB, 5               ; Set PB5 as output

wait_input:
    ; 1. Check if a byte arrived
    LDS   R17, UCSR0A
    SBRS  R17, RXC0             ; Wait for Receive Complete
    RJMP  wait_input

    ; 2. Read the character into R18
    LDS   R18, UDR0

    ; 3. Check if character is 'H'
    CPI   R18, 'H'
    BREQ  led_on

    ; 4. Check if character is 'L'
    CPI   R18, 'L'
    BREQ  led_off

    RJMP  wait_input

led_on:
    SBI   PORTB, 5              ; Turn LED ON
    RJMP  wait_input

led_off:
```

```
    CBI   PORTB, 5              ; Turn LED OFF
    RJMP  wait_input
```

Revision #1
Created 2026-02-09 14:22:19 UTC by BH
Updated 2026-02-09 14:22:35 UTC by BH