AVR UART Echo with Increment Program

Your Name

August 5, 2025

Overview

This document explains a simple UART communication program for an AVR microcontroller (such as ATmega328P on Arduino Uno). The program initializes UART, receives a byte, increments it by 1, and sends it back.

1 Program Code

Listing 1: UART Echo with Increment

```
#define F_CPU 16000000UL
#define BAUD 9600
#define UBRR_VALUE ((F_CPU / (16UL * BAUD)) - 1)

#include <avr/io.h>

void uart_init() {
    UBRR0H = (uint8_t)(UBRR_VALUE >> 8);
    UBRR0L = (uint8_t)UBRR_VALUE;
    UCSR0B = (1 << RXEN0) | (1 << TXEN0);        // Enable RX and TX
    UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);      // 8-bit data, 1 stop bit
}

uint8_t uart_receive() {
    while (!(UCSR0A & (1 << RXC0)));  // Wait for data to be received
    return UDR0;                      // Return received byte
}

void uart_send(uint8_t data) {
    while (!(UCSR0A & (1 << UDRE0))); // Wait for empty transmit buffer
```

```
    UDR0 = data;                        // Send data
}

int main(void) {
    uart_init();

    while (1) {
        uint8_t received = uart_receive();
        uint8_t incremented = received + 1;
        uart_send(incremented);
    }
}
```

2 Explanation of Components

ATmega328P Microcontroller

The ATmega328P is the microcontroller used in the Arduino Uno. It features a Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) module which allows serial communication.

UART Registers Used

- **UBRR0H and UBRR0L:** USART Baud Rate Registers High and Low byte. Used to set the baud rate.
- **UCSR0A (USART Control and Status Register A):** Contains flags like RXC0 (receive complete) and UDRE0 (data register empty).
- **UCSR0B (USART Control and Status Register B):** Controls enabling/disabling receiver (RXEN0) and transmitter (TXEN0).
- **UCSR0C (USART Control and Status Register C):** Configures frame format (data bits, parity, stop bits).
- **UDR0 (USART Data Register):** Used to read received data or write data to be transmitted.

UART Pins on Arduino Uno (ATmega328P)

- **RX (Pin 0)** - Receive pin for serial data input.
- **TX (Pin 1)** - Transmit pin for serial data output.

3 How it works

1. The UART is initialized with a baud rate of 9600 bps, 8 data bits, no parity, and 1 stop bit.
2. The program waits for a byte to be received via UART.
3. Once received, it increments the byte value by 1.
4. Then sends the incremented byte back through UART.
5. This process repeats indefinitely.

4 References

- AVR libc User Manual - UART Baud Rate
- ATmega328P Datasheet