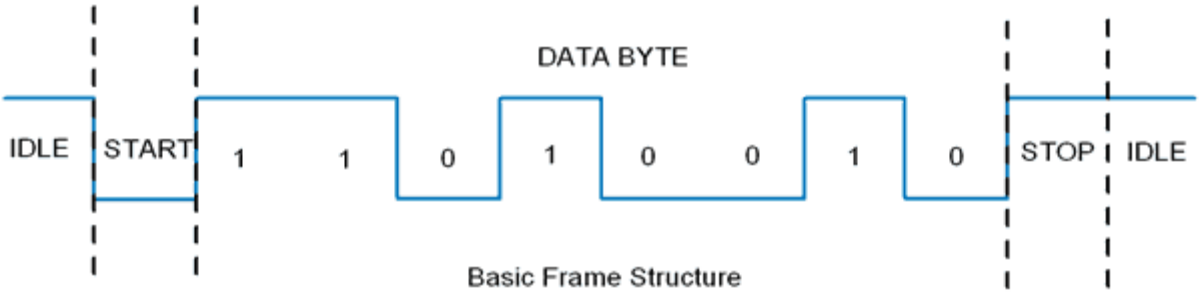
USART in AVR ATmega16/ATmega32

Khung dữ liệu nối tiếp

Trong khi gửi/nhận dữ liệu sẽ có một số bit được thêm vào dữ liệu để định nghĩa thời điểm bắt đầu và kết thúc của dữ liệu. Một chuỗi dữ liệu thông thường sẽ cấu trúc bao gồm: 8-bit dữ liệu, 1-start bit (có mức logic 0) và 1-bit stop (có mức logic 1) như hình dưới đây:



Bên cạnh đó, UART còn hỗ trợ một số định dạng khác như kiểm tra bit chẵn lẻ, các độ dài dữ liệu khác nhau (5-9 bit).

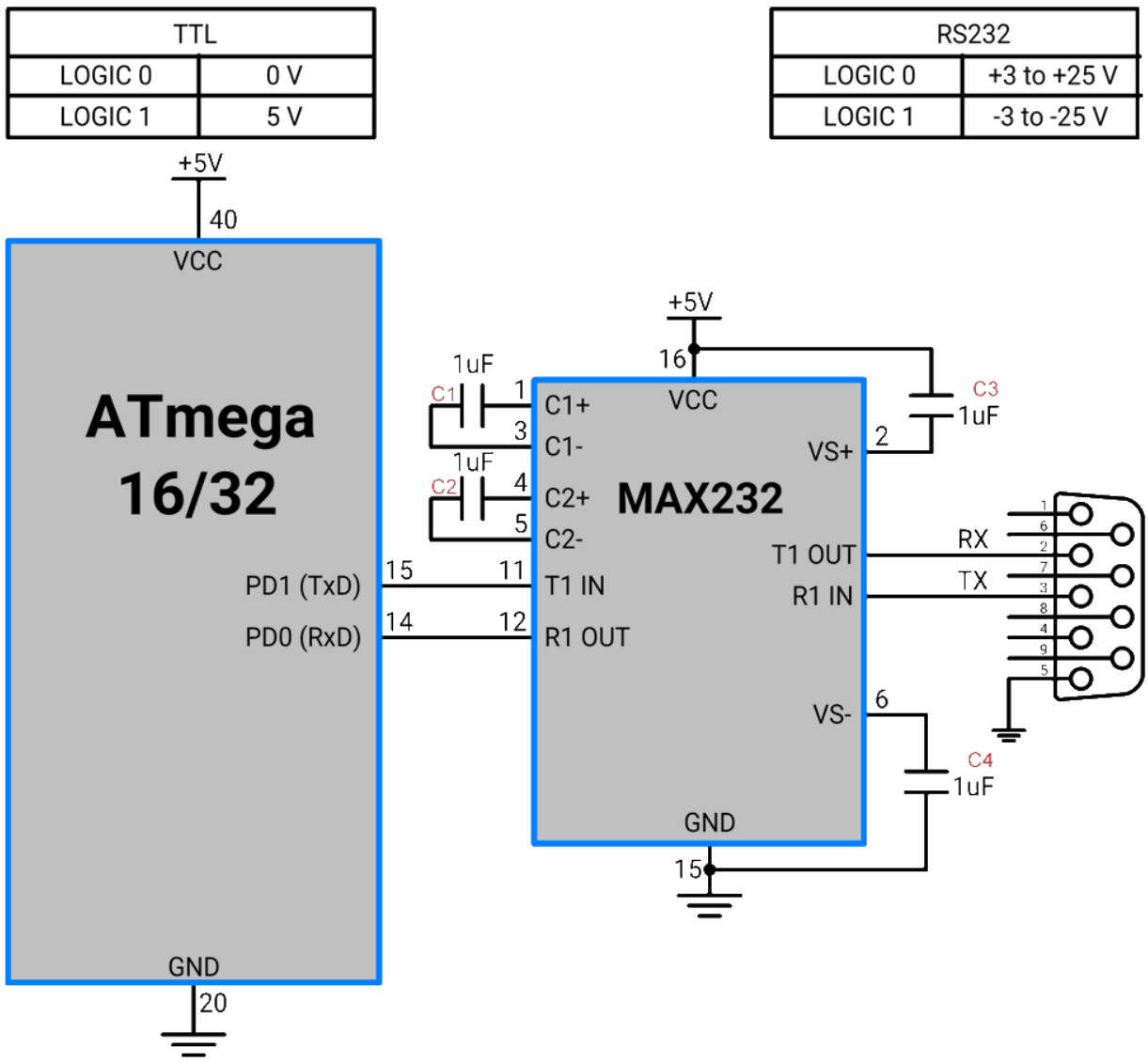
Tốc độ (Baud rate)

Bit rate là số lượng bit trên giây (bps), cũng được hiểu như là Baud rate trong các hệ thống nhị phân. Thông thường, định nghĩa này cho biết đường giao tiếp nối tiếp là nhanh như thế nào. Có mố số tốc độ chuẩn như 1200, 2400, 4800, 19200, 115200 bps …. Thường thì tốc độ 9600 bps hay được sử dụng.

Kết nối phần cứng

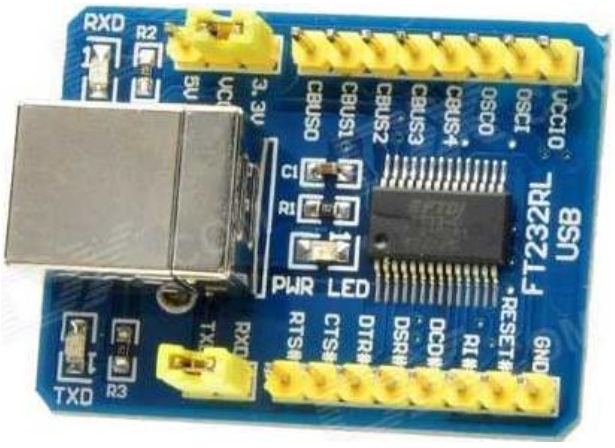
Thông thường chỉ cần các đường Tx (Truyền), Rx (Nhận) và GND (đất).

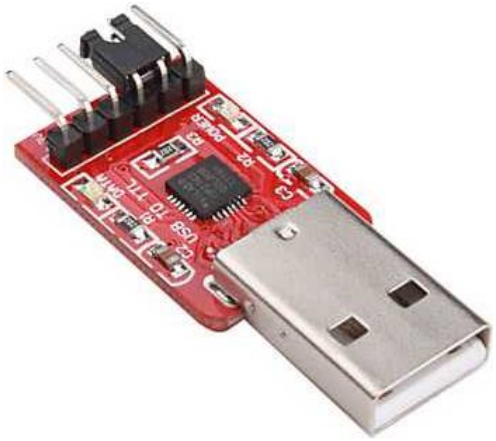
- USART trong AVR ATmega có mức điện áp TTL khi mà 0V ứng với mức logic 0 và 5V ứng với mức logic 1.
- Trong máy tính và hầu hết các thiết bị cũ, chuẩn RS232 được sử dụng cho truyền thông nối tiếp với cổng nối dạng ‘D’ có 9 chân. Truyền thông nối tiếp RS232 có các mức điện áp khác với mức điện áp quy định trên ATmega, ví dụ mức +3V tới +25V ứng với mức logic 0 và -3V tới -25V ứng với mức logic 1.
- Vì vậy, để giao tiếp theo chuẩn RS232, chúng ta cần sử dụng bộ chuyển đổi mức điện áp như MAX232 IC.



Giao tiếp nối tiếp ATmega 16/32

Ngày nay, với các máy PC và laptop loại mới sẽ không có chuẩn RS232 và cổng nối DB9. Chúng ta phải sử dụng thiết bị chuyển đổi từ USB sang nối tiếp. Có nhiều loại chuyển đổi USB sang nối tiếp như CP2102, FT232RL, CH340, ….





Bộ chuyển đổi USB sang nối tiếp

Để quan sát truyền thông nối tiếp, chúng ta có thể sử dụng các phần mềm Terminal nối tiếp như Realterm, Teraterm ... Bằng cách chọn chính xác cổng nối tiếp (cổng COM trong Windows) và tốc độ baud rate, chúng ta có thể mởi cổng nối tiếp cho truyền thông.

USART trên ATmega16 có các thuộc tính sau:

- Hoạt động theo kiểu song công (các thanh ghi truyền và nhận nối tiếp độc lập)
- Chế độ hoạt động đồng bộ hoặc bất đồng bộ
- Chế độ Master hoặc Slave với đồng bộ về xung nhịp
- Bộ tạo tốc độ baud rate có độ phân giải cao
- Hỗ trợ các khung nối tiếp có 5, 6, 7, 8, hoặc 9-bit dữ liệu và 1 hoặc 2-bit Stop
- Bộ tạo và kiểm trả chẵn lẻ được hỗ trợ bởi phần cứng
- Phát hiện tràn dữ liệu
- Phát hiện lỗi khung dữ liệu
- Bộ lọc nhiễu bao gồm phát hiện lỗi Start bit và bộ lọc thông thấp số
- Hỗ trợ 3 ngắt riêng biệt là truyền hoàn thành, nhận hoàn thành và thanh ghi dữ liệu truyền rỗng
- Chế độ giao tiếp đa xử lý
- Chế độ giao tiêp bất đồng bộ tốc độ nhân đôi

Lập trình USART trên AVR

1. UDR: Thanh ghi dữ liêu USART

Về cơ bản nó có 2 bộ đệm là Tx Byte và Rx Byte. Cả hai cùng chia sẻ thanh ghi UDR. Khi chúng ta ghi dữ lieu vào thanh ghi UDR thì bộ đệm Tx sẽ được ghi và khi chúng ta đọc dữ liệu từ thanh ghi này thì bộ đệm Rx sẽ được đọc. Bộ đệm sử dụng thanh ghi dịch FIFO để truyền dữ liệu.

2. UBRR: Thanh ghi tốc độ truyền USART

Đây là thanh ghi có độ dài 16-bit được sử dụng để cài đắt tốc độ truyền (baud rate).

3. UCSRA: Thanh ghi trạng thái và điều khiển USART

7	6	5	4	3	2	1	0
RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM

- **Bit 7 – RXC:** Bit báo hoàn thành nhận

Bit cờ báo này có giá trị 1 khi có dữ liệu chưa được đọc trong thanh ghi UDR. Bit cờ RXC có thể được sử dụng để tạo ra một ngắt nhận hoàn thành.

- **Bit 6 – TXC:** Bit báo truyền hoàn thành

Bit cờ báo này có giá trị 1 khi toàn bộ khung truyền từ bộ đệm Tx được dịch ra ngoài Vi điều khiển và không có dữ liệu mới xuất hiện trong bộ đệm truyền (UDR). Bit cờ TXC tự động được xóa về 0 khi ngắt truyền hoàn thành được thực hiện, hoặc nó có thể được xóa bằng cách ghi giá trị 1 tới vị trí bit tương ứng. Cờ TXC có thể tạo ra ngắt truyền hoàn thành.

- **Bit 5 – UDRE:** Bit báo thanh ghi dữ liệu rỗng

Nếu bit UDRE có giá trị 1, nghĩa là bộ đệm là rỗng và chỉ ra bộ đệm truyền (UDR) đã sẵn sàng nhận dữ liệu mới. Cờ UDRE có thể tạo ra ngắt thanh ghi dữ liệu rỗng. UDRE sẽ có giá trị 1 sau khi vi điều khiển được reset để chỉ ra bộ truyền đã sẵn sàng.

- **Bit 4 – FE:** Bit báo lỗi khung truyền

- **Bit 3 – DOR:** Bit báo tràn dữ liệu

Bit này có giá trị 1 nếu điều kiện tràn dữ liệu xảy ra. Tràn dữ liệu xảy ra khi bộ đệm nhận đầy (2 ký tự) và một ký tự mới đang đợi trong thanh ghi dịch nhận.

- **Bit 2 – PE:** Bit báo lỗi kiểm tra chẵn lẻ

- **Bit 1 – U2X:** Bit cài đặt tăng gấp đôi tốc độ truyền dữ liệu
- **Bit 0 – MPCM:** Bit cài đặt chế độ truyền thông đa chip

4. UCSRB: Thanh ghi trạng thái và điều khiển USART

7	6	5	4	3	2	1	0
RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8

- **Bit 7 – RXCIE:** Bit cho phép ngắt khi nhận hoàn thành

  Ghi giá trị 1 tới bit này cho phép ngắt nhận hoàn thành

- **Bit 6 – TXCIE:** Bit cho phép ngắt khi truyền hoàn thành

  Ghi giá trị 1 tới bit này cho phép ngắt truyền hoàn thành

- **Bit 5 – UDRIE:** Bit cho phép ngắt khi thanh ghi dữ liệu UDR rỗng

  Ghi giá trị 1 tới bit này cho phép ngắt khi thanh ghi dữ liệu UDR rỗng

- **Bit 4 – RXEN:** Bit cho phép bộ nhận hoạt động

  Ghi giá trị 1 tới bit này cho phép bộ nhận USART hoạt động

- **Bit 3 – TXEN:** Bit cho phép bộ truyền hoạt động

  Ghi giá trị 1 tới bit này cho phép bộ truyền USART hoạt động

- **Bit 2 – UCSZ2:** Bit cài đặt kích thước ký tự

Bit UCSZ2 kết hợp cùng với các bit UCSZ1 và UCSZ0 trong thanh ghi UCSRC để cài đặt số lượng bit dữ liệu (kích thước ký tự) trong một khung truyền và nhận.

- **Bit 1 – RXB8:** Bit dữ liệu thứ 9 khi chế độ nhận 9-bit dữ liệu được sử dụng
- **Bit 0 – TXB8:** Bit dữ liệu thứ 9 khi chế độ truyền 9-bit dữ liệu được sử dụng

5. UCSRC: Thanh ghi trạng thái và điều khiển USART

7	6	5	4	3	2	1	0
URSEL	UMSEL	UPM1	UPM0	USBS	USCZ1	USCZ0	UCPOL

- **Bit 7 – URSEL:** Bit lựa chọn thanh ghi

Bit này được sử dụng để lựa chọn giữa việc truy cập thanh ghi UCSRC hay UBRRH, khi mà cả 2 thanh ghi cùng sử dụng một địa chỉ. Để cài đặt được thanh ghi UCSRC thì chúng

ta phải ghi giá trị 1 tới bit URSEL nếu không dữ liệu ghi vào sẽ được ghi tới thanh ghi UBRRH.

- **Bit 6 – UMSEL:** Bit lựa chọn chế độ USART

  0 = Hoạt động ở chế độ bất đồng bộ

  1 = Hoạt động ở chế độ đồng bộ

- **Bit 5:4 – UPM1:0**: Bit cài đặt chế độ kiểm tra chẵn lẻ

Những bit này cho phép và cài đặt phương pháp tạo và kiểm tra chẵn lẽ. Nếu lỗi kiểm tra chẵn lẻ xảy ra thì bit cờ PE trong thanh ghi UCSRA sẽ có giá trị là 1.

UPM1	UPM0	Chế độ
0	0	Tắt chế độ
0	1	Không dùng
1	0	Cho phép và kiểu kiểm tra là chẵn
1	1	Cho pháp và kiểu kiểm tra là lẻ

- **Bit 3 – USBS**: Bit lựa chọn số lượng Stop Bit

Bit này chọn số lượng Stop Bits và được chèn bởi bộ phát. Bộ nhận bỏ qua cài đặt này.

0 = 1-bit

1 = 2-bit

- **Bit 2:1 – UCSZ1:0**: Bit cài đặt kích thước dữ liệu

Các bit UCSZ1:0 kết hợp với bit UCSZ2 trong thanh ghi UCSRB để cài đặt số lượng bit dữ liệu (kích thước ký tự) trong khung truyền và nhận.

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	**1**	**1**	**8-bit**
1	0	0	Không dùng
1	0	1	Không dùng

UCSZ2	UCSZ1	UCSZ0	Character Size
1	1	0	Không dùng
1	1	1	9-bit

- **Bit 0 – UCPOL:** Bit cài đặt cực xung đồng hồ

Bit này chỉ được sử dụng trong chế độ đồng bộ. Ghi giá trị 0 tới bit này để hoạt động ở chế độ bất đồng bộ.

6. UBRRL và UBRRH: Các thanh ghi cài đặt tốc độ truyền USART

- **Bit 15 – URSEL:** Bit lựa chọn thanh ghi

Bit này được sử dụng để lựa chọn giữa việc truy cập thanh ghi UCSRC hay UBRRH, khi mà cả 2 thanh ghi cùng sử dụng một địa chỉ. Để cài đặt được thanh ghi UCSRC thì chúng ta phải ghi giá trị 1 tới bit URSEL nếu không dữ liệu ghi vào sẽ được ghi tới thanh ghi UBRRH.

- **Bit 11:0 – UBRR11:0:** Thanh ghi cài đặt tốc độ truyền USART

Được sử dụng để định nghĩa tốc độ truyền (baud rate)

$$BaudRate = \frac{F_{osc}}{16 * (UBRR + 1)}$$

Hay:

$$UBRR = \frac{F_{osc}}{16 * BaudRate} - 1$$

Ví dụ: Giả sử tần số hoạt động của vi điều khiển Fosc=8 MHz và tốc độ truyền (baud rate) yêu cầu là 9600 bps. Khi đó, **UBRR= 51.088** hay **51.**

Chúng ta cũng có thể cài đặt giá trị này thông qua một macro như sau:

```
#define F_CPU 8000000UL                /* Define frequency here its 8MHz */
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)
```

BAUD_PRESCALE là giá trị mà chúng ta phải tải vào thanh ghi UBRR để cải đặt tốc độ truyền.

Các bước lập trình

Cài đặt USART

1. Cho phép bộ truyền/ nhận hoạt động sử dụng thanh ghi UCSRB.
2. Cài đặt kích thước ký tự là 8-bit sử dụng thanh ghi UCSRC.
3. Cài đặt tốc độ truyền (baud rate) sử dụng thanh ghi UBRR.

```
void UART_init(long USART_BAUDRATE)
{
     UCSRB |= (1 << RXEN) | (1 << TXEN);   /* Bộ bộ truyền và bộ nhân */
     UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1);  /* Sử dụng 8-bit */
     UBRRL = BAUD_PRESCALE;                /* 8-bit thấp của baud rate */
        UBRRH = (BAUD_PRESCALE >> 8);      /* 8-bits cao của baud rate*/
}
```

Nhận ký tự

Giám sát bit RXC trong thanh ghi UCSRA. Bit RXC chỉ ra trạng thái nhận ký tự hoàn thành hay chưa.

```
unsigned char UART_RxChar()
{
      while ((UCSRA & (1 << RXC)) == 0);   /* Đợi cho tới khi dữ liệu được nhận */
      return(UDR);                         /* Trả lại byte dữ liệu nhận được */
}
```

Truyền ký tự

Monitor the UDRE bit from UCSRA. When UDRE flag becomes one, which indicates transmitting buffer is empty and ready to accept another byte.

Giám sát bit UDRE trong thanh ghi UCSRA. Khi cờ UDRE có giá trị 1 thì nghĩa là bộ đệm truyền rỗng và sẵn sàng để nhận 1 byte dữ liệu khác.

```
void UART_TxChar(char ch)
```

```
{
    while (! (UCSRA & (1<<UDRE)));   /* Wait for empty transmit buffer */
    UDR = ch ;
}
```

Ví dụ

Chương trình sau sẽ nhận 1 ký tự và gửi lại ký tự đó. Chúng ta có thể sử dụng PC Terminal để quan sát.

```
/*
    ATmega 16 UART echo program
*/

#define F_CPU 8000000UL             /* Define frequency here its 8MHz */

#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdio.h>

//#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

void UART_init(long USART_BAUDRATE)
{
    UCSRB |= (1 << RXEN) | (1 << TXEN);    /* Cho phép truyền, nhận */
    UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1); /* Sử dụng 8-bit dữ liệu */
    UBRRL = BAUD_PRESCALE;                  /* 8-bit thấp của baud rate */
    UBRRH = (BAUD_PRESCALE >> 8);           /* 8-bit cao của baud rate */
}

unsigned char UART_RxChar()
{
    while ((UCSRA & (1 << RXC)) == 0);  /* Đợi tới khi nhận được dữ liệu */
    return (UDR);                       /* Trả lại dữ liệu */
}

void UART_TxChar(char ch)
{
    while (! (UCSRA & (1<<UDRE)));      /* Đợi tới khi bộ đệm truyền rỗng */
    UDR = ch;
}

void UART_SendString(char *str)
{
    unsigned char j=0;

    while (str[j] != 0)             /* Gửi chuỗi ký tự cho tới khi kết thúc */
    {
```

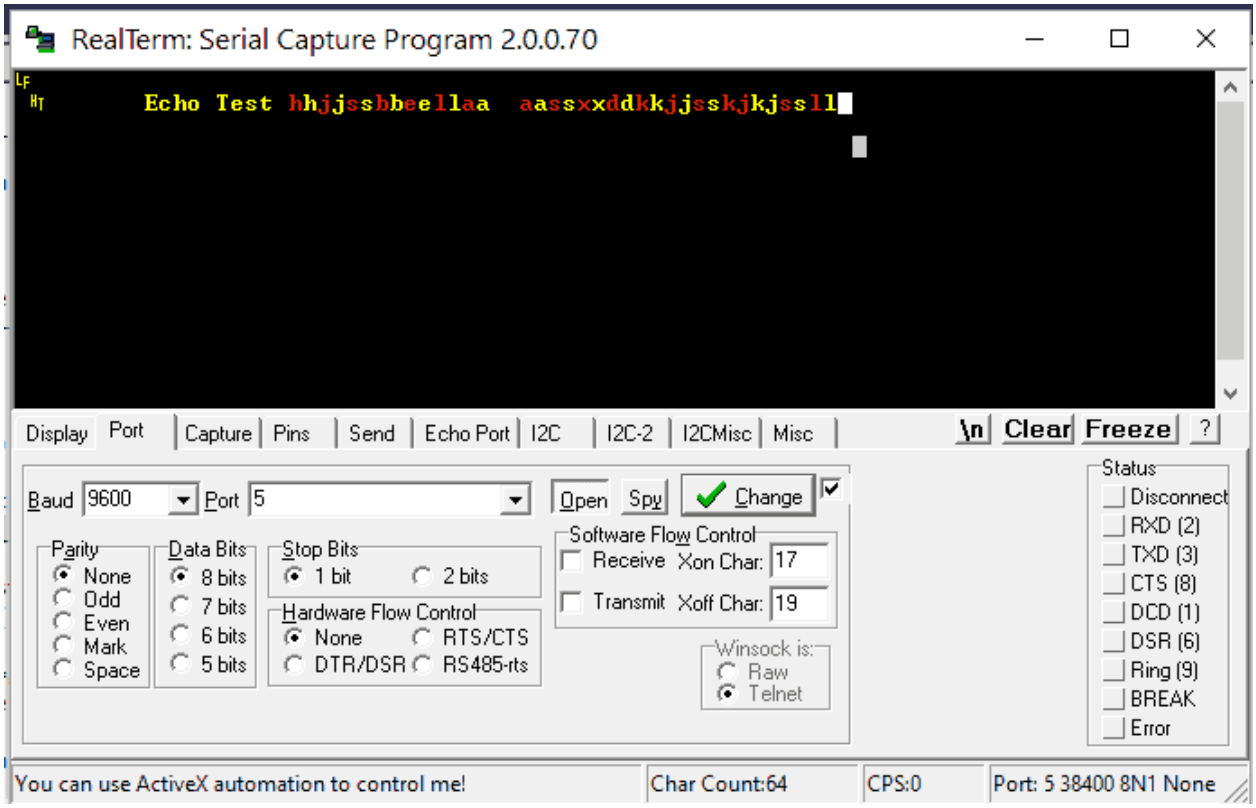
```
        UART_TxChar(str[j]);
        j++;
    }
}

int main(void)
{
    char c;

    UART_init(9600);
    UART_SendString("\n\t Echo Test ");     /* Gửi chuỗi ký tự */

    while(1)
    {
        c = UART_RxChar();    /* Nhận ký tự*/
        UART_TxChar(c);       /* Gửi ký tự vừa nhận được */
    }
}
```

Giao diện quan sát được trên máy tính



Lập trình USART sử dụng ngắt

Để cho phép ngắt truyền, chúng ta phải ghi giá trị 1 tới bit UDRIE.

Để cho phép ngắt nhận, chúng ta phải ghi giá trị 1 tới bit RXCIE trong thanh ghi UCSRB.

Ví dụ:

Nhận dữ liệu sử dụng ngắt: Nhận dữ liệu và gửi tới PORTB

```
/*
     ATmega 16 UART using interrupt
*/

#define F_CPU 8000000UL             /* Define frequency here its 8MHz */

#include <avr/io.h>
#include <avr/interrupt.h>

#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)


void UART_init(long USART_BAUDRATE)
{
    UCSRB |= (1 << RXEN) | (1 << RXCIE);   /* Bật bộ nhận, bộ truyền */
    UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1); /* Sử dụng 8-bit  */

    UBRRL = BAUD_PRESCALE;                 /* 8-bit thấp của baud rate */
    UBRRH = (BAUD_PRESCALE >> 8);          /* 8-bit cao của baud rate */
}


ISR(USART_RXC_vect)
{
    PORTB = UDR;
}


int main(void)
{
    DDRB = 0xFF;

    UART_init(9600);
    sei();    /* Cho phép ngắt toàn cục */

    while(1);
}
```