CPE 310L
EMBEDDED SYSTEM DESIGN (CPE)

LABORATORY 10
USART

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF NEVADA, LAS VEGAS

GOAL
The goal of this lab is to transmit data to a PC using the Serial communication standard RS232.

BACKGROUND
RS232
RS232 is the name for a set of standards used for serial communications between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE). Originally the DB-25 used to be the standard connection. The DB-9 connection later replaced the DB-25. This was later replaced by USB connections because the RS232 limited speed and large voltage swings. For this lab we will be using the DB-9 connector. Normally connection is done with a DTE connecting to a DCE. For this lab we will use a DTE to DTE connection which is otherwise referred to as a null modem connection.

USART
AVR supports 4 Universal asynchronous receiver/transmitter modes: Normal asynchronous, double speed asynchronous, master synchronous and slave synchronous. For PC connections we will focus on using only the asynchronous modes. There are 5 registers that control the settings for USART in AVR.

UBRR
This register controls the baud rate at which the signal will be transmitted. The baud rates must be the same on both the receiving and transmitting end. The baud rate selected for the
microcontroller must match the baud rate of the serial port settings. This register is split into high and low components to accommodate higher baud rates.

**UDR**
This is the data register which stores the data the will be transmitted or the data that has been received.

**UCSRA/B/C**
These registers deal with controlling the various settings for the USART connection. This includes the parity bit, character size, mode (asynchronous or synchronous) and switching between transmit mode and receive mode.

**FORMULAS**
Calculating the Baud rate is determined by which USART mode the microcontroller currently has active. The table below shows the formulas for calculating the baud rate and the UBRR register value that is desired.

<table>
<thead>
<tr>
<th>Operating Mode</th>
<th>Equation for Calculating Baud Rate</th>
<th>Equation for Calculating UBRRn Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous Normal mode (U2Xn = 0)</td>
<td>( BAUD = \frac{f_{osc}}{16(UBRRn + 1)} )</td>
<td>( UBRRn = \frac{f_{osc}}{16 \cdot BAUD} - 1 )</td>
</tr>
<tr>
<td>Asynchronous Double Speed mode (U2Xn = 1)</td>
<td>( BAUD = \frac{f_{osc}}{8(UBRRn + 1)} )</td>
<td>( UBRRn = \frac{f_{osc}}{8 \cdot BAUD} - 1 )</td>
</tr>
<tr>
<td>Synchronous Master mode</td>
<td>( BAUD = \frac{f_{osc}}{2(UBRRn + 1)} )</td>
<td>( UBRRn = \frac{f_{osc}}{2 \cdot BAUD} - 1 )</td>
</tr>
</tbody>
</table>

**MICROCONTROLLER FUSES**
In the past we have spoken on internal features such as clock speed. However, most of these features cannot be accessed via code. To adjust settings such as clock speed you must access and modify the fuses of the microcontroller. You can access these fuses by opening up the AVR programming menu and clicking on the tab labeled fuses. Most of the settings are default configurations and do not need to be modified. For the purpose of this lab you should know how to change the internal clock speed. By clicking on the box marked CKDIV8 you can change the internal clock from 1MHz to 8Mz and vice versa.
PRELAB

1. Explain the function of UDRn, UCSRAR, UCSRnB and UCSRnC registers.

Code for transmitting data:

Initialize function
{
    //enable transmitter, enable receiver
    //set 8 bit, 1stop bit, disabled parity, synchronous USART
    //select UBRR0L for 8Mhz 9600 baud, check datasheet
}

Sending function
{
    //check if buffer is empty so that data can be written to transmit
    //copy data to be sent to UDR0
}

Receiving function
{
    //check if there is unread data in the buffer
    //read the data from the UDR0
}

Main function
{
    //initialize usart
    while(1)
    {
        //send data
        //read data
    }
}
1. **Transmission to the terminal**  
Modify the above code to work for the Atmega328P so the microcontroller will transmit data to the terminal. Set your Baud rate to 9600. Use the schematic below as a guide for your connections.

![Transmission connection schematic](image)

**Fig. 2. Transmission connection schematic**

2. **Terminal transmission**  
Modify your code to receive data and transmit the data back.

3. **Variable values transmission**  
Write a program where the microcontroller will take any letter and transmit the opposite case of it back to the terminal. For example, if the letter ‘T’ is sent from the terminal, the microcontroller should send back the letter ‘t’ to the terminal.
### POSTLAB

Include the following elements in the report document:

<table>
<thead>
<tr>
<th>Section</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Theory of operation&lt;br&gt;&lt;em&gt;Include a brief description of every element and phenomenon that appears during the experiments.&lt;/em&gt;</td>
</tr>
<tr>
<td>2</td>
<td>Prelab report</td>
</tr>
<tr>
<td>3</td>
<td>Results of the experiments</td>
</tr>
<tr>
<td>Experiment</td>
<td>Experiment Results</td>
</tr>
<tr>
<td>1</td>
<td>a. Code with comments&lt;br&gt;b. Picture of the terminal with transmission result</td>
</tr>
<tr>
<td>2</td>
<td>a. Code with comments&lt;br&gt;b. Picture of the terminal with transmission result</td>
</tr>
<tr>
<td>3</td>
<td>a. Code with comments&lt;br&gt;b. Picture of the terminal with transmission result</td>
</tr>
<tr>
<td>4</td>
<td>Answer the questions</td>
</tr>
<tr>
<td>Question no.</td>
<td>Question</td>
</tr>
<tr>
<td>1</td>
<td>What is baud speed? How is it different from bits/second?</td>
</tr>
<tr>
<td>2</td>
<td>What is Asynchronous and Synchronous transmission?</td>
</tr>
<tr>
<td>3</td>
<td>What is Synchronous Master mode?</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions&lt;br&gt;&lt;em&gt;Write down your conclusions, things learned, problems encountered during the lab and how they were solved, etc.&lt;/em&gt;</td>
</tr>
</tbody>
</table>