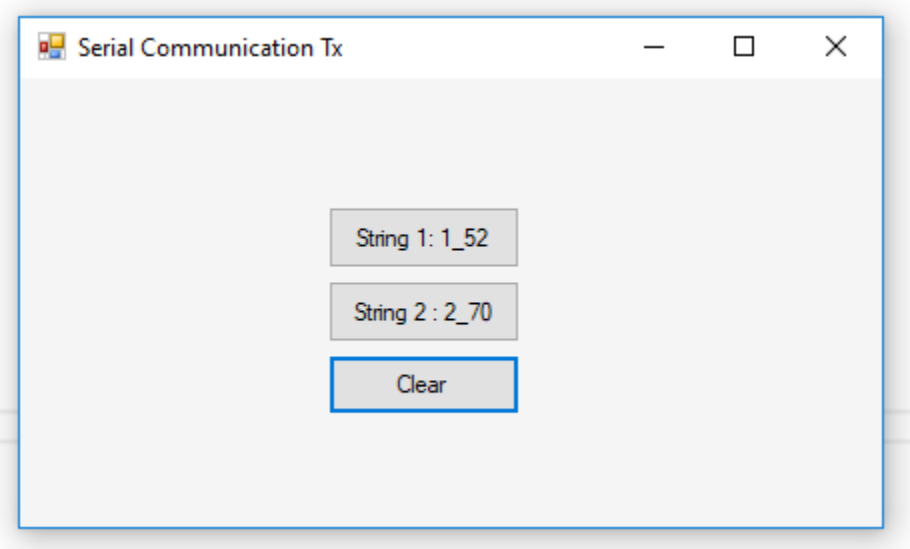
Serial Communication Tx (String)

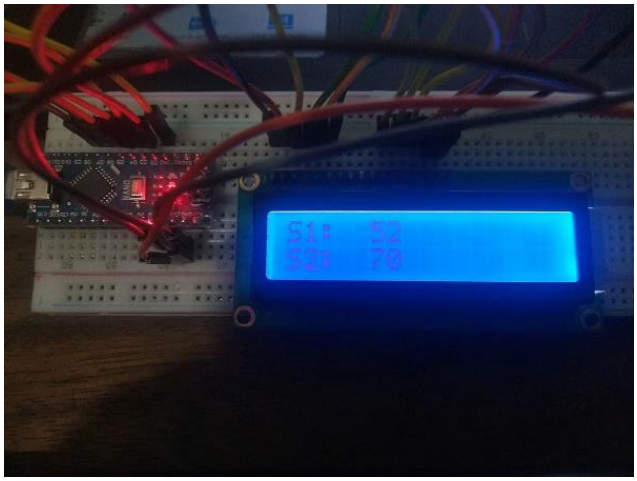
In this program, two numbers 52 and 70 will be transmitted. As in serial communication string is transmitted and, in the receiver, single byte is received at each time, so, these two numbers will be embedded in a string envelop and transmitted. The receiver will decode the received character set and bring out the data from the envelop.

.NET Framework C# Code:




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO.Ports;
using System.IO;
using System.Windows.Forms;

namespace Topic_02___Serial_Communication_TX
{

    public partial class Form1 : Form
    {
        char[] String_1 = {'\x02','1','5','2','\x03'};  // See ASCII Chart
        char[] String_2 = {'\x02','2','7','0','\x03'};  // See ASCII Chart
        char[] String_3 = {'\x27'};
        SerialPort Myport = new SerialPort("COM3", 9600);
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
```

```
        {
            try
            {
                Myport.Open();
                Myport.Write(String_1,0,5);
                Myport.Close();
            }
            catch(IOException)
            {
                MessageBox.Show("Error");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                Myport.Open();
                Myport.Write(String_2, 0, 5);
                Myport.Close();
            }
            catch (IOException)
            {
                MessageBox.Show("Error");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                Myport.Open();
                Myport.Write(String_3, 0, 1);
                Myport.Close();
            }
            catch (IOException)
            {
                MessageBox.Show("Error");
            }
        }
    }
}
```



Arduino code:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to

const int rs = 8, en = 7, d4 = 6, d5 = 5, d6 = 4, d7 = 3;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

volatile int Data_Array[5];
volatile int S1[2],S2[2];
volatile int x,i;

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("S1: ");
  lcd.setCursor(0, 1);
  lcd.print("S2: ");
  Serial.begin(9600);

}

void loop() {
  if(Serial.available() > 0 )
  {
    x = Serial.read();
    if(x=='\x02')i=0;
```

```
  else if (x=='\x03')
    {
     if(Data_Array[0] == '1')
       {S1[0] = Data_Array[1]-48; S1[1] = Data_Array[2]-48;}
     else if(Data_Array[0] == '2')
       {S2[0] = Data_Array[1]-48; S2[1] = Data_Array[2]-48;}
    }
  else if(x== '\x27')
  {
    lcd.setCursor(5, 0);S1[0]=0;S1[1]=0;
    lcd.setCursor(5, 1);S2[0]=0;S2[1]=0;
    i=0;
  }
  else {Data_Array[i]=x;i=i+1;}
 }

 lcd.setCursor(5, 0);lcd.print(S1[0]);lcd.print(S1[1]);
 lcd.setCursor(5, 1);lcd.print(S2[0]);lcd.print(S2[1]);
}
```