

please help try to receive data from arduino with visual C# communication

system

oct. '14 #1

Hi, i want to develop my project by create some interface gui from visual C# to monitor data that receive from Arduino (i used leonardo).

problem is i cannot read data from arduino by usb port but i can send data from visual c# to arduino and arduino response with data that i send.

then how i can read data from arduino.

my code is below.

Visual C#

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

namespace usb_port_test
{
    public partial class main : Form
    {
        public main()
        {
            InitializeComponent();

            //List port available
            foreach (string iPort in SerialPort.GetPortNames())
            {
                cboPort.Items.Add(iPort);
            }

            //List baudrate
            cboBaud.Text = "9600"; //default
            int[] iBaud = new int[] {9600,14400,19200,57600,115200};
            foreach (int i in iBaud)
            {
                cboBaud.Items.Add(i);
            }

            richTextBox1.AppendText("Start ...");
        }

        public void portSetting() // usb port config
        {
            usbPort.PortName = cboPort.Text;
            usbPort.BaudRate = Convert.ToInt32(cboBaud.Text);
            usbPort.DataBits = 8;
            usbPort.Parity = Parity.None;
            usbPort.StopBits = StopBits.One;
            usbPort.Encoding = System.Text.ASCIIEncoding.Default;
            usbPort.ReceivedBytesThreshold = 1;
            usbPort.ReadTimeout = 5000;
            usbPort.WriteTimeout = 5000;
            usbPort.NewLine = "\n";
        }
    }
}
```

```

private void btnOpen_Click(object sender, EventArgs e) // button open
{
    if (usbPort.IsOpen == false)
    {
        try
        {
            portSetting();
            usbPort.Open();
            richTextBox1.AppendText(Environment.NewLine + "Open port");
        }
        catch (Exception ex)
        {
            richTextBox1.AppendText(Environment.NewLine + "Error : " + ex.Message);
        }
    }
    else
    {
        MessageBox.Show(cboPort.Text + " is already open");
    }
}

private void btnClose_Click(object sender, EventArgs e) // button close
{
    if (usbPort.IsOpen)
    {
        usbPort.Close();
        richTextBox1.AppendText(Environment.NewLine + "Close port");
    }
    else
    {
        MessageBox.Show(cboPort.Text + " is already close");
    }
}

private void btnSend0_Click(object sender, EventArgs e) // button send 0
{
    try
    {
        richTextBox1.AppendText(Environment.NewLine + "Sending 0 ...");
        usbPort.Write("0");
        richTextBox1.AppendText(" successful");
        richTextBox1.AppendText(Environment.NewLine + "Receiving Data");
        string iRead = usbPort.ReadLine();
        richTextBox1.AppendText(" successful and data is " + iRead);
        Thread.Sleep(1000);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        richTextBox1.AppendText(Environment.NewLine + "Error : " + ex.Message);
    }
}

private void btnSend1_Click(object sender, EventArgs e) // button send 1
{
    try
    {
        richTextBox1.AppendText(Environment.NewLine + "Sending 1 ...");
        usbPort.Write("1");
        richTextBox1.AppendText(" successful");
        richTextBox1.AppendText(Environment.NewLine + "Receiving Data");
        string iRead = usbPort.ReadLine();
        richTextBox1.AppendText(" successful and data is " + iRead);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        richTextBox1.AppendText(Environment.NewLine + "Error : " + ex.Message);
    }
}

```

```

private void richTextBox1_TextChanged(object sender, EventArgs e) //
{
    richTextBox1.ScrollToCaret();
}
}
}

```

Arduino Sketch

```

int getdata = 0;

void setup()
{
    pinMode(0, OUTPUT);
    pinMode(1, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    getdata = Serial.read();
    test();
}

void test()
{
    if(getdata == '0')
    {
        digitalWrite(0, HIGH);
        digitalWrite(1, LOW);
        Serial.write("5");
        delay(1000);
        digitalWrite(0, LOW);
    }
    else if(getdata == '1')
    {
        digitalWrite(1, HIGH);
        digitalWrite(0, LOW);
        Serial.write("6");
        delay(1000);
        digitalWrite(1, LOW);
    }
    delay(500);
}

```

filk
oct. '14 #2

use Serial.write writeln print, and println functions depending on the data your outputting on the arduino

to catch it with .net you will need to create a on serial recieved event to trigger your code to use serialobject.ReadLine() function to capture the sent data..

system
oct. '14 #3

filk: use Serial.write writeln print, and println functions depending on the data your outputting on the arduino

to catch it with .net you will need to create a on serial recieved event to trigger your code to use serialobject.ReadLine() function to capture the sent data..

how i can trigger ? sorry i'm newbie.

filk
oct. '14 #4

vakebi:
how i can trigger ? sorry i'm newbie.

been over a decade since i did any .net 😊 but google chuckled this up [c# serial read](#)

Robin2**oct. '14 #5**

Your PC program must also take account of how the Arduino behaves.

When the PC opens the serial port the Arduino will reset and can't do anything for a few seconds. The PC program must wait for that to complete. A simple solution is to have the Arduino send a short message from setup() and then the PC program waits until it receives that.

Also, your PC program must open the serial port once and keep it open until it is finished using the Arduino.

This [demo using Python](#) illustrates what I am saying. It may be useful learning experience to write a C# program to work with my Arduino program - that way you can be certain one part is working properly.

...R

system**oct. '14 #6**

Another thing to consider is that the UI in your application runs in one thread. Serial data sending and receiving runs in another thread. The serial thread can't directly write to the UI elements in the UI thread.

Attached is a C# application that uses threads, proper communication between threads, and a delegate to handle incoming serial data, so that you can see how the C# application should be structured.

[CommunicateWithArduino.zip](#) (54 KB)

system**nov. '14 #7**

thank all.

i can communication between Arduino and Visual C#

thank for Pauls

PaulS: Another thing to consider is that the UI in your application runs in one thread. Serial data sending and receiving runs in another thread. The serial thread can't directly write to the UI elements in the UI thread.

Attached is a C# application that uses threads, proper communication between threads, and a delegate to handle incoming serial data, so that you can see how the C# application should be structured.

for your code and i can do it !!!

and actually i try to cut your code step by step and it just add only

usbPort.DtrEnable = true; --> then can work !!!

[Back to top](#)

[Help Center](#)[Distributors](#)[FOLLOW US](#)[Contact Us](#)[Careers](#)[Trademark & Copyright](#)[Brand Guidelines](#)