Serial Communication in C# Express

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005−07/msg04525.html

- *From*: Haluk Gokmen <chemc53@xxxxxxxxxxx>
- *Date*: Wed, 27 Jul 2005 04:49:30 −0700

I had an inquiry about the use of MSComm OCX in a C# application awhile ago and I really appreciated Nicholas Paldino's help, which is listed below...
My code worked well in VS2003. I am now transfering the code into C# Express, which has a new SerialPort class in it. I came across another problem this time. My code is also listed below. I would appreciate your comments.

/******* Your previous answer to my inquiry ********/
Haluk,

When you set the output like this:

com.Output = ((char) 34) + ((char) 14);

You are actually sending a (char) type, not a string, which is what you really want to send. To do this, do the following:

com.Output = new string(new char[]{(char) 34, (char) 14});

This will create a new string composed of the two characters that will be marshaled correctly to the COM object.

Also, I suspect that the output stops when it hits the null character. To get around this, you will have to send the byte manually.

Hope this helps.

−−
− Nicholas Paldino [.NET MVP]
− nicholas.paldino@xxxxxxxxxxxxxx

"Haluk" <xxxxxxxxxxxxxx> wrote in message
news:b03001c20e13$606eb230$37ef2ecf@xxxxxxxxxxxxxx
> I am using mscommlib.ocx in my C# application to
> communicate via the serial port.
> After all the initialization code which I had already used

> in my previous VB6 applications successfully, I am using
> the following code to transmit data:
>
> com.Output = ((char) 34)+ ((char) 14);
>
> I am getting the following error code when I run this
> program:
>
> An unhandled exception of
> type 'System.Runtime.InteropServices.COMException'
> occurred in axinterop.mscommlib.dll
> Additional information: Exception from HRESULT: 0x800A017C
> (CTL_E_INVALIDPROPERTYVALUE).
>
> But also if I write the same code as follows:
>
> com.Output = "" + ((char) 34)+ ((char) 14);
>
> It runs successfully.
> This is my first problem...
>
> The second one is: instead of '34' or '14', for example if
> I decide to send '0' (zero), this character is not transmitted at all!
>
>
> com.Output = "" + ((char) 34)+ ((char) 0);
>
> I can only achieve to transmit '34', but not '0' when I
> run this code..
>
> Any explanation will be deeply appreciated..
>
> Regards,
> Haluk
>
>

/******* My new code in C# (Express) ******************/

After the usual port initialization,

```
if (com.IsOpen) com.Close();
com.Open();

// send char 34,14,192,51,0,0
com.Write(new string(new char[] { (char)34, (char)14, (char)192,
(char)51, (char)0, (char)0 }, 0, 6));

// for testing purposes I connected TX and RX pins of the port...
```

```
// receive routine:
private void com_DataReceived(object sender, SerialDataReceivedEventArgs
e)
{

// This method will be called when there is data waiting in the port's
buffer

// Obtain the number of bytes waiting in the port's buffer
bytes = com.BytesToRead;

// Create a byte array buffer to hold the incoming data
buffer = new char[bytes];

// Read the data from the port and store it in our buffer
com.Read(buffer, 0, bytes);

}
```

/****** My question about this new code *******/
After I activate the send routine, 6 bytes of data is received by pc but with some difference to the originally sent data.

I receive : 34,14,63,51,0,0 everytime instead of 34,14,192,51,0,0

I tried to transmit 34,14,127,51,0,0 and received the same data sequence...

As I try to tranmit anything bigger than 127 I receive 63 instead of the original data...

Do you have any idea why this happens?
Best Regards,
Haluk

.

- Prev by Date: ***Re: calculator features***
- Next by Date: ***Re: Convert a string to a datetime using a pattern***
- Previous by thread: ***Convert a string to a datetime using a pattern***
- Next by thread: ***Generics: Inheritance and Interface questions***
- Index(es):
  - ***Date***
  - ***Thread***