

---

Subject: [SOLVED] Managing String buffer  
Posted by [Giorgio](#) on Fri, 20 Jan 2017 08:56:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi there,  
as it seems that there is no way to use the boost library with u++, I digged into the forum and found a thread that helped me, but now I have a problem treating a buffer of the String class.

In this message [http://www.ultimatepp.org/forums/index.php?t=msg&th=5401&goto=27851&#msg\\_27851](http://www.ultimatepp.org/forums/index.php?t=msg&th=5401&goto=27851&#msg_27851) there is a u++ project containing a working serial library. With some modifications it works with my device. My problem is when I read data from the device and try to print them on screen. The library has a method (ReadDataWaiting) that tells me how many characters the device sent out as response. Using that method I get 7, that is what I expect. Problems come when I try to actually read those data.

There is this method: `int SerialPort::ReadData( void *buffer, int limit )` that, as far as I understood, puts the read data in a buffer.

In the CommPak sources there is also this:  
`String buffer[1024]; // Rx buffer`

...

```
CommPort.ReadData(buffer, sizeof(buffer));  
So I wrote the following code:  
if (x = commPort.ReadDataWaiting())  
{  
    commPort.ReadData(buffer, sizeof(buffer));  
    std::cout << "buffer" << std::endl;  
    std::cout << buffer << std::endl;  
}
```

But the result on screen is "011C3DE8" while it should be a string like "TAV 45", so my problem now is just how to treat that buffer to get results in plain English.

This is a very important and urgent matter for me and any help is very appreciated.  
Thanks,  
Gio

---

Subject: [SOLVED]Re: Managing String buffer  
Posted by [Giorgio](#) on Fri, 20 Jan 2017 12:48:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Changing

```
String buffer[1024]; // Rx buffer
```

to

```
char buffer[20]; // Rx buffer
```

fixed the issue.

---