

Re: Serial COM port communication problems

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2007-05/msg00206.htm>

- *From:* an0047@xxxxxxxxxx
 - *Date:* 21 May 2007 11:14:00 -0700
-

On 21 Mai, 03:21, "Scott McPhillips [MVP]" <org-dot-mvps-at-scottmcp> wrote:

It is not really practical to use the rx flag to detect end of message. The problem is that your program is subject to latency. It may not resume execution for 50–100 msec after the CR is received, and in that time more characters can arrive.

But this should not cause a loss of data, assuming you are reading the actual number of characters returned by WaitCommEvent. If you are losing characters you have a bug in your program, or perhaps ClearCommError will show a hardware timing error.

You are right, I could experience this behavior today. The rx flag is used to detect the subpackages of the whole package. As soon as the CR is detected an int event counter (or subpackage counter) is incremented. The event is always very accurately detected. So at least I have some kind of signal to know how many subpackages arrived. Then the queue size is requested by ClearCommError, as some new bytes already arrived the size returned is bigger than the actual subpackage, including the CR. Later on ReadFile is called. Non OVL implementation clears the receiving queue, the OVL implementation doesn't yet.

I suggest that you use overlapped I/O and the SetCommTimeouts to cause it to return when the buffer is full or a brief silence (a few character times) is detected. Then parse the returned buffer to find the CR characters yourself. With this approach I have successfully stress tested the COM ports at extreme speeds and loads without ever losing a character.
Scott McPhillips [MVP VC++]

Re: Serial COM port communication problems

How can this be done? (SetCommTimeout association to buffer size?)...I will try to understand the overlapped method better. Do you mean the Readfile Buffer that receives the data read? I'm free to set the size of that buffer, I will need to find the best size for this buffer, subpackages vary among them as much as 20 bytes. I'm a bit afraid of adding timing parameters to the problem but it seems an elegant approach, hope doesn't get more complex.

The problem by this type of signalization (the new timeouted buffers) is that there will be CRs embedded on them. So the CR is first to be found, and regard the left part as a subpackage, the right part as a part of a subpackage an so on...? Is there some type of generic algorithm to build the subpackages together?

Thank you very much for your advice

.