

# MultiCom

*for LabVIEW<sup>®</sup> for Windows<sup>™</sup>*

**Windows 95/98/NT Edition**

*Copyright © 1994 - 1999, Viewpoint Software Solutions, Inc.  
All Rights Reserved*

Viewpoint Software Solutions, Inc. does not warrant that the Program will meet Customer's requirements or will operate in the combinations which may be selected by the Customer or that the operation of the Program will be uninterrupted or error free or that all Program defects will be corrected.

VIEWPOINT SOFTWARE SOLUTIONS, INC. DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS THAT MAY BE OBTAINED BY USING THIS SOFTWARE. ACCORDINGLY, THE SOFTWARE AND ITS DOCUMENTATION ARE SOLD "AS IS" WITHOUT WARRANTY AS TO THEIR PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE PROGRAM IS ASSUMED BY YOU.

NEITHER VIEWPOINT SOFTWARE SOLUTIONS, INC. NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS, RESULTING FROM THE USE OF THE PROGRAM OR ARISING OUT OF ANY BREACH OF ANY WARRANTY. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR THE ABOVE MAY NOT APPLY TO YOU.

LabVIEW® is a trademark of National Instruments Inc.

All other brand and product names are trademarks or registered trademarks of their respective companies.

© Copyright Viewpoint Software Solutions, Inc., 1994 - 1999.  
All Rights Reserved.

Reproduction or adaptation of any part of this documentation beyond that permitted by Section 117 of the 1976 United States Copyright Act without permission of the Copyright owner is unlawful.

Viewpoint Software Solutions, Inc.  
2320 Brighton Townline Road  
Rochester, NY 14623-2708

Printed in the U.S.A.  
January 1999  
Part Number: VSS-MC003

# Table Of Contents

---

Introduction .....	1
Installation and Setup .....	1
MultiCom Files .....	4
MultiCom VIs .....	8
General Information .....	17
Error Codes .....	22
How To Reach Us .....	23



# Introduction

---

MultiCom is a set of LabVIEW VIs that allow communication with the Digi Avanstar PCI Viewpoint board. The PCI Viewpoint board supports either 8 or 16 RS-232 or RS-422 communications ports. Each port is configurable to support a wide range of baud rates, handshake modes and other communications parameters.

# Installation and Setup

---

MultiCom version 5.00 supports Windows 95, Windows 98, and Windows NT environments. The following steps are required to configure your MultiCom hardware.

## 1. Install the MultiCom software.

A setup program is provided to ensure proper software installation. To install MultiCom, run the SETUP.EXE file on the installation diskette. The installation program prompts for which version (95/98 or NT) should be installed. A program group called MultiCom for Windows 95/98/NT will be added to Windows. This group will contain two icons, one for MCLOAD.EXE and an uninstall program.

The installed files are described in the next section "MultiCom Files" (page 4).

**Note:** If you are installing MultiCom over an older version make sure that the old MCOM32.DLL file is removed from your hard disk. The installation program checks your windows and windows system directories for this file but if an older version is located in other directories, problems could result.

## 2. Shut Down Your Computer and Install the Hardware

Install the MultiCom hardware, including the PCI card and the 8 or 16-port panel. Upon rebooting the PC, Windows should recognize the new hardware and set it up automatically for you. You will then need to restart your computer one more time.

If the 5.00 software is an upgrade and you are running Windows 95/98, use explorer to find `WINDOWS\INF\MULTICOM.INF`, right click on the file, and choose install. Then, go to the Device Manager tab under System in the Control Panel and under the Viewpoint MultiCom device/MultiCom PCI, choose the Driver tab and select Update Driver. Choose to not have Windows select the driver and select the MultiCom driver dated 1/19/99. Windows will then require you to reboot.

Windows NT was not supported by the PCI board under previous versions of MultiCom. Windows NT should automatically recognize the PCI board once the computer is restarted.

## 3. Run the MCLOAD program or the Avanstar Control Program Loader VI.

This step downloads the control program (the A100P.CP file) to the processor on the MultiCom board. This program or VI must be run each time the computer is booted.

Note that either the MCLOAD program or the Avanstar Control Program Loader VI can be run – you do not need to run both. Both the EXE and the VI perform the same function and user preference will determine which method is used. Also note that the EXE or VI should only need to be run once.

You can run the MCLOAD program by selecting the MC Load program item in the “MultiCom 5.00 for Windows 95/98/NT” program group. Alternatively, MCLOAD can be run from a DOS box prompt (note that MCLOAD does not require any command line arguments).

MCLOAD runs in DOS mode and will report any errors in a DOS box. If there are multiple boards, the program will report any errors for each board.

MCLOAD and the Avanstar Control Program Loader.vi search for the A100P.CP file as follows:

- ▼ first in the current directory
- ▼ second in the windows directory
- ▼ third in the windows system (SYSTEM for Windows 95/98; SYSTEM32 for Windows NT) directory

**Warning:** Downloading the control program will reset the state of all MultiCom ports to their default settings. Do not download the control program while data transfer is occurring as this will likely cause data loss/corruption.

# MultiCom Files

---

## Common Files

The following files are common to both the Windows 95/98 installation and the Windows NT installation.

### **A100P.CP**

This file is the control program file for the Avanstar board. This file is used by the MCLOAD download program.

### **MCOM32.DLL**

This file is the Windows driver file for the Avanstar PCI Viewpoint board. This file is placed in the \WINDOWS\SYSTEM directory for Windows 95/98 installations and in \WINDOWS\SYSTEM32 for Windows NT installations.

### **MCLOAD.EXE**

This file downloads and initializes the control program that runs on the MultiCom board. Refer to Step 3 of the installation procedure for information on using this program.

### **README.TXT**

This file contains any last minute information not included in this manual.

## SERIAL.LLB

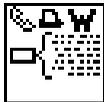
Use this set of VIs if you intend to convert existing VIs that use the computer's COM ports to use the Avanstar COM ports. This file is designed to look exactly like the SERIAL.LLB file that comes from National Instruments as a part of LabVIEW. The MultiCom SERIAL.LLB contains five VIs:

- ▼ Bytes at Serial Port.vi
- ▼ Serial Port Break.vi
- ▼ Serial Port Init.vi
- ▼ Serial Port Read.vi
- ▼ Serial Port Write.vi

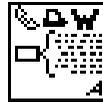
Note that the names of the VIs are exactly the same as the names of the VIs in the original SERIAL.LLB. This was done so that an existing LabVIEW application can be switched over to use the Avanstar COM ports by simply replacing the original SERIAL.LLB with the MultiCom SERIAL.LLB. Once the MultiCom SERIAL.LLB is in place, LabVIEW will automatically use the new VIs instead of the old ones and therefore will use the Avanstar ports instead of the computer's COM ports. The MultiCom VIs are wired the same as the LabVIEW serial VIs.

When replacing the original SERIAL.LLB be careful not to overwrite or delete it. It is a good idea to rename the original SERIAL.LLB file before copying the MultiCom version onto your hard disk.

Once you have replaced the serial VI library, the only noticeable difference that you will see is on the lower right corner of the icons where there will be a small "A" to signify "Avanstar" as shown below.



*Original icon*



*MultiCom icon*

## SERIALAV.LLB

This library contains 17 VIs. The first five are renamed copies of the VIs in the SERIAL.LLB.

- ▼ Avanstar Bytes at Serial Port.vi
- ▼ Avanstar Serial Port Break.vi
- ▼ Avanstar Serial Port Init.vi
- ▼ Avanstar Serial Port Write.vi
- ▼ Avanstar Serial Port Read.vi

Three VIs are permutations of the original five VIs listed above. Avanstar Serial Port Read+.vi includes EOS (End-Of-String) detection and timeout. Avanstar Serial Port Write+.vi includes a timeout value. Avanstar Serial Port Init+.vi allows ports to be configured as either RS-232 or RS-422 and allows different input and output buffer sizes to be specified.

- ▼ Avanstar Serial Port Init+.vi
- ▼ Avanstar Serial Port Read+.vi
- ▼ Avanstar Serial Port Write+.vi

The remaining VIs were included to increase the power and ease-of-use of the MultiCom software.

- ▼ Avanstar Bytes at Output Serial Port.vi
- ▼ Avanstar Comm Input Status.vi
- ▼ Avanstar Control Program Loader.vi
- ▼ Avanstar CTS/DSR Status.vi
- ▼ Avanstar EOS Status.vi
- ▼ Avanstar Max Serial Port.vi
- ▼ Avanstar Output Buf Clear.vi
- ▼ Avanstar RTS/DTR Control.vi
- ▼ Avanstar Serial 422 Port Init.vi

## Window 95/98 Files

### **MCOM95.DLL**

This file is the Windows 95/98 driver file for the Avanstar PCI Viewpoint board. This file is called from MCOM32.DLL and is installed in the \WINDOWS\SYSTEM directory.

### **MULTICOM.INF**

This file provides information to Windows 95/98 that allow the board to be recognized and configured by the operating system. This file is installed in the \WINDOWS\INF directory

### **MULTICOM.VXD**

This file is a virtual device driver for MultiCom under Windows 95/98. This file is installed in the \WINDOWS\SYSTEM directory.

## Windows NT Files

### **MCOMNT.DLL**

This file is the Windows NT driver file for the Avanstar PCI Viewpoint board. This file is called from MCOM32.DLL and is installed in the \WINDOWS\SYSTEM32 directory.

### **MCPCI.SYS**

This file is a system file for MultiCom to run under Windows NT. This file is installed in the \WINDOWS\SYSTEM32\DRIVERS directory.

## MultiCom VIs

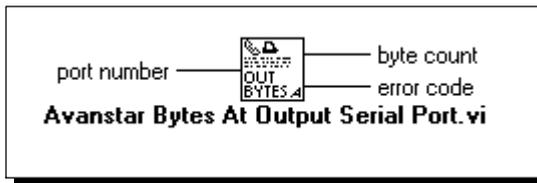
---

The VIs described in this section are the MultiCom VIs that go beyond the functionality of the serial VIs provided in LabVIEW. For information on the following VIs:

- ▼ Avanstar Bytes at Serial Port.vi
- ▼ Avanstar Serial Port Break.vi
- ▼ Avanstar Serial Port Init.vi
- ▼ Avanstar Serial Port Write.vi
- ▼ Avanstar Serial Port Read.vi

refer to the documentation on the equivalent VI in the LabVIEW “Function and VI Reference Manual.”

### Avanstar Bytes at Output Serial Port.vi



This VI returns the number of bytes waiting to be transmitted in the output buffer. Typically the only time the byte count will be a non-zero value is if some sort of output handshaking is enabled and the receiver is holding off the data transmission.

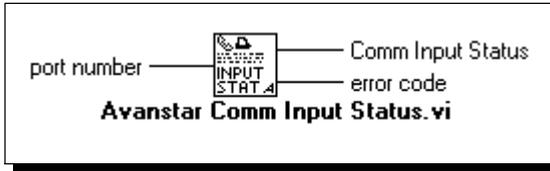


*port number* is the MultiCom port number (zero-based).



*byte count* is the number of bytes in the output buffer that have not been sent.

## Avanstar Comm Input Status.vi



**I32**

*port number* is the MultiCom port number (zero-based).

**U16**

*Comm Input Status* returns a bit field with the following data:

**Bit 0:** Input data overrun error - updated after a read; is set when both the input buffer and the FIFO (12 bytes) are full.

**Bit 1:** Parity error in input stream

**Bit 2:** Framing error in input stream

**Bit 3:** Break character received

## Avanstar Control Program Loader.vi



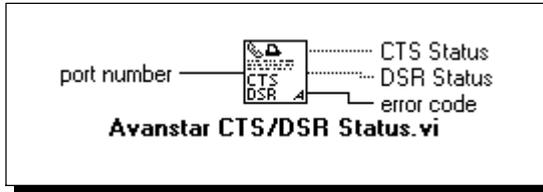
This VI downloads and initializes the control program that runs on the MultiCom board.

**[abc]**

*results* is a string array describing the download results for each MultiCom board found.

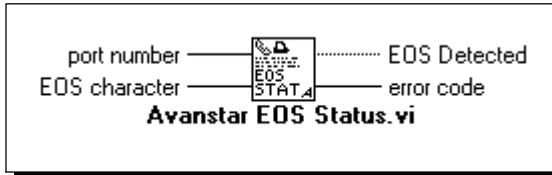
Refer to Step 3 of the installation procedure (page 2) for additional information on downloading the control program.

## Avanstar CTS/DSR Status.vi



- I32** *port number* is the zero-based MultiCom port number.
- TF** *CTS Status* returns the status of the “Clear to Send” line. A true indicates that CTS is asserted.
- TF** *DSR Status* returns the status of the “Data Set Ready” line. A true indicates that DSR is asserted.

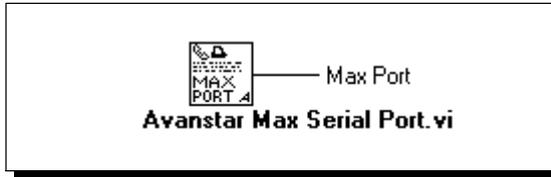
## Avanstar EOS Status.vi



This VI checks the input buffer for the specified “end of string” character. No data is removed or transferred out of the input buffer. Use this VI to check for an EOS character before the *Avanstar Serial Port Read+.vi* is called to eliminate the need for the read operation to wait for the reception of the EOS character.

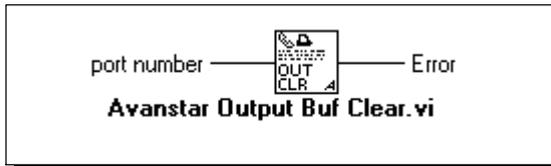
- I32** *port number* is the MultiCom port number (zero-based).
- U8** *EOS character* is the ASCII value of the character to be checked for.
- TF** *EOS Detected* returns a true if the *EOS character* is detected in the input buffer.

## Avanstar Max Serial Port.vi



**I16** *Max Port* is the value of the largest MultiCom port number (one-based) detected in hardware.

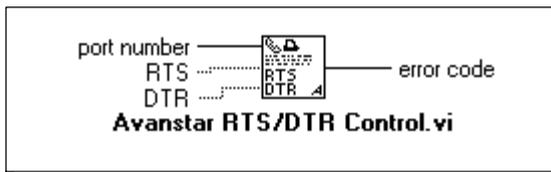
## Avanstar Output Buf Clear.vi



This VI clears (empties) the output buffer of the specified port. Use this VI if the buffer needs to be cleared because the receiving device is holding off data transfer with either the CTS line or an XOff handshake.

**I32** *port number* is the MultiCom port number (zero-based).

## Avanstar RTS/DTR Control.vi



This VI controls the RTS and DTR lines. Note that this VI will set the state of these lines but it does not guarantee that the lines will

remain at that state. If input handshaking is enabled, the RTS and DTR lines may change due to input buffer changes.



*port number* is the MultiCom port number (zero-based).

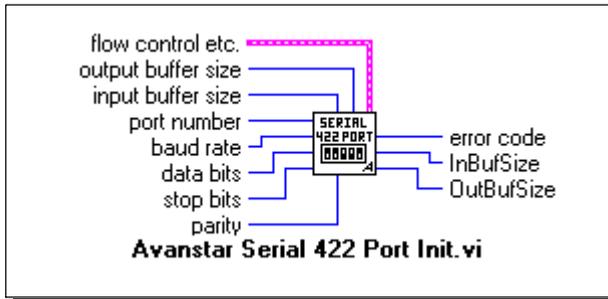


*RTS* sets the status of the "Request to Send" line. A true value asserts the RTS line.



*DTR* sets the status of the "Data Transfer Ready" line. A true value asserts the DTR line.

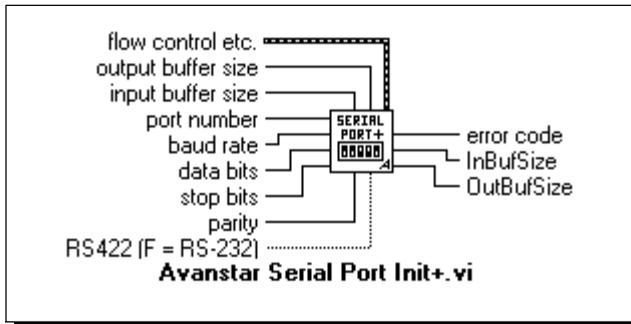
### Avanstar Serial 422 Port Init.vi



This VI configures the MultiCom port for RS-422. This VI wires identically to the Avanstar Serial Port Init+.vi with the exception of the RS-422 input. Refer to the Avanstar Serial Port Init+.vi documentation for descriptions of this VI's parameters.

**Caution:** The cable wiring for RS-232 and RS-422 are not compatible. Be sure that your configuration matches your hardware setup.

## Avanstar Serial Port Init+.vi



This VI performs the same function as the Avanstar Serial Port Init.vi and the Avanstar Serial 422 Port Init.vi with several additional inputs to increase functionality. The primary new feature of this VI is the ability to set the input buffer and output buffer to different sizes. Only the additional inputs are documented here – the functionality of the other inputs is unchanged.

**U16** *output buffer size* sets the size of the output buffer. A value of zero leaves the buffer size unchanged.

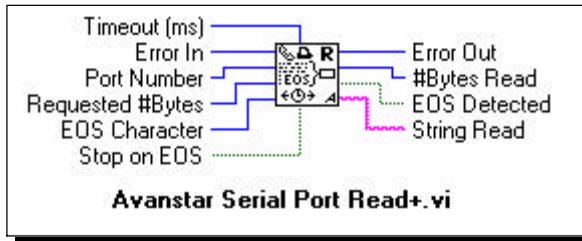
**U16** *input buffer size* sets the size of the input buffer. A value of zero leaves the buffer size unchanged.

**TF** *RS422* determines if the port is to be configured as an RS-422 port (true) or an RS-232 port (false).

**U16** *OutBufSize* returns the size that the output buffer is actually set to. If the requested size is too large, the VI will return a –10 error code and the buffer sizes will be unchanged. See “Buffer Sizes” on page 19 for more information.

**U16** *InBufSize* returns the size that the input buffer is actually set to. If the requested size is too large, the VI will return a –10 error code and the buffer sizes will be unchanged. See “Buffer Sizes” on page 19 for more information.

## Avanstar Serial Port Read+.vi



This VI performs the same function as the Avanstar Serial Port Read VI but includes options for EOS (End-Of-String) detection and timeout. Also, this VI includes an Error In terminal for easy sequencing of multiple reads.



**Timeout** is the maximum time in milliseconds that the MultiCom software will wait for a character to be received. If the *Timeout* period expires, the Avanstar Serial Port Read+ will return with *Error Out* equal to 1 (one). To turn off the timeout feature, set *Timeout* to 0 (zero).

The resolution of the *Timeout* value is one PC timer tick or 54.9 ms. (i.e., the actual timeout used will be a multiple of 54.9 ms.).



**Error In** controls whether the VI will execute. If *Error In* is 0 (zero), the VI will execute normally. If *Error In* is non-zero, the VI will simply pass the *Error In* value through to *Error Out* without executing a read.

*Error In* can be used to control the sequencing of multiple read+ VIs without using a LabVIEW sequence frame.



**Port Number** is the Avanstar port number to be read. Port numbering is zero based – note that this is different from the labeling on the Avanstar connection panel.

**U32**

*Requested # Bytes* is the maximum number of characters that will be read from the Avastar port regardless of whether an EOS is found.

**U8**

*EOS Character* is the ASCII code for the character which will cause the serial read to stop (if Stop on EOS is turned on). For example, if a serial device returns a string of the form,

```
+#.####<cr><lf>
```

setting the EOS Character to 10 (a linefeed character) will cause the read to stop after it has received the linefeed character.

**TF**

*Stop on EOS* determines whether the read will search for an EOS character. The read will be terminated on EOS if *Stop on EOS* is true.

**I32**

*Error Out* is the error value returned by the read function. If *Error Out* is a 1 (one), a timeout occurred. If *Error Out* is a negative number, a board level error occurred.

**U32**

*#Bytes Read* is the length of the *String Read*.

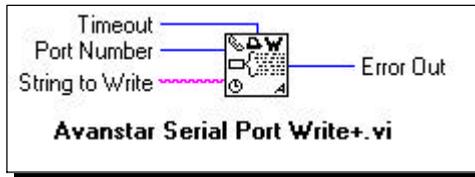
**TF**

*EOS Detected* shows whether the read stopped because of an EOS detection.

**abc**

*String Read* is the data received from the Avastar port.

## Avanstar Serial Port Write+.vi



This VI performs the same function as the Avanstar Serial Port Write VI but includes an option for timeout.

**U32** *Timeout* is the maximum time in milliseconds that the MultiCom software will wait to come back from a write operation. If the *Timeout* period expires, the Avanstar Serial Port Write+ will return with *Error Out* equal to 1 (one). To turn off the timeout feature, set *Timeout* to 0 (zero).

The resolution of the *Timeout* value is one PC timer tick or 54.9 ms. (i.e., the actual timeout used will be a multiple of 54.9 ms.).

**I32** *Port Number* is the Avanstar port number to be written to. Port numbering is zero based – note that this is different from the labeling on the Avanstar connection panel.



*String to Write* is the data that is written to the Avanstar port.

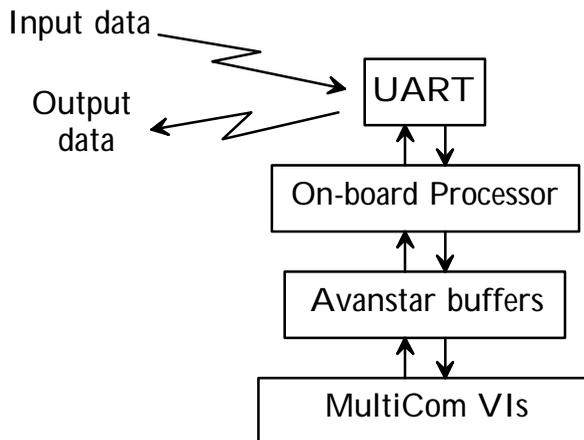
**I32** *Error Out* is the error value returned by the write function. If *Error Out* is a 1 (one), a timeout occurred. If *Error Out* is a negative number, a board level error occurred.

# General Information

---

## Understanding the Avanstar Hardware

Much of the job of sending and receiving data is taken care of by the Avanstar board. The Avanstar board contains an individual UART for each channel and an on-board microprocessor to handle data traffic between the UARTs and your PC/LabVIEW. Each UART is a circuit designed to send and receive serial data and contains two 12 byte FIFO (First In, First Out) buffers – one for sending and one for receiving data. The board's processor monitors all data channels and transfers data between the UART's FIFOs and the board's buffer area. The MultiCom VIs send/receive data by transferring data to/from the Avanstar buffers. See figure below.



## Stop Bit Support

The Avanstar board supports either 1 or 2 stop bits; the 1.5 stop bits option is not available. To configure a port for 1 stop bit, pass a 0 (zero) to the stop bits parameter in the init VI. To select 2 stop bits, pass a 2 to the init VI.

## Port Numbering

In order to remain compatible and consistent with LabVIEW, the MultiCom VIs use a zero based numbering scheme, (i.e., the first port is number zero). Note that this conflicts with the numbering on the Avanstar terminal panel which is one based.

If multiple boards are used, the ports will be numbered according to the slot location of the board. PCI slots are numbered in the computer. The lowest slot number will have the first set of ports, the next lowest number will have the next set of ports, and so on.

If the PCI slot number is unknown, a simple test can be performed. To see which panel has the first set of port numbers, load the Avanstar 422 Serial Port Init VI and initialize port 0. The panel that has the 422 status light enabled contains the first set of ports. As long as the PCI board is not put in a different slot, this configuration will be valid for subsequent boots of the PC.

## Baud Rates

The following baud rates are supported under MultiCom.

50	300	4,800	57,600
75	600	7,200	64,000
110	1,050	9,600	76,800
134.5	1,200	19,200	115,200
150	1,800	38,400	
200	2,400	56,000	

The Avanstar board has an option for 134.5 baud rate. To select this rate, pass the value 134 to the init baud rate terminal.

## Handshaking Modes

All of the handshaking modes featured in the original LabVIEW VIs are supported by MultiCom *except* "Output alt HW Handshake." If you need output hardware handshaking you may have to convert your application to use the CTS line and "Output HW Handshake" mode.

## Input Buffer Handshaking

When either hardware (RTS) or XON/XOFF handshaking is being used for receiving data, the MultiCom port signals to the sending device when it can/cannot receive additional data. The number of characters in the input buffer determines when the RTS line changes and/or when the XON and XOFF characters are sent. The point in the buffer at which the sending device will be signaled to stop transmitting data is called the "high water mark." Similarly the "low water mark" is the buffer location at which the sending device will be signaled to resume sending data.

MultiCom sets the high water mark to 32 characters less than the size of the input buffer. The low water mark is set to 64 characters less than the size of the input buffer. Consequently, the input buffer size must be greater than 64 bytes for input handshaking to work.

## Buffer Sizes

The Avanstar PCI Viewpoint board contains 64K of memory. This memory is used for board control registers as well as for input and output buffers. Out of the 64 Kbytes, approximately 63 Kbytes is available for buffers.

The default buffer size is 2000 bytes for the 16 port board (i.e. 2000 bytes for the input buffer and another 2000 bytes for the output buffer for each port.) and 4048 bytes for the 8 port board. The minimum buffer size is 2 bytes. The maximum buffer size is only limited by the amount of free space available; therefore to allocate a large buffer for a

particular port, it is necessary to reduce the size of the some of the other port buffers.

The init VI default for buffer size is 0 (zero). The effect of this (or any value less than 2) is to leave the buffer size unaltered from its current value. Therefore leaving the buffer size terminal unwired on any of the MultiCom init VIs will not change the buffer size.

**Note:** The original Serial Port Init VI will default the input and output buffer sizes to 1024 bytes if the buffer size terminal is left unwired. Keep this difference in mind if you replace an original VI with one of the MultiCom init VIs; the operation of your application may be affected.

**Caution:** Do not change the size of any MultiCom buffer while any port is active. Changing buffer sizes moves the locations of the buffers of many MultiCom ports and could result in lost data.

## Reads/Writes May Hang LabVIEW

The MultiCom software is not interrupt driven and therefore once LabVIEW enters the MultiCom code, no Windows (LabVIEW or otherwise) can occur. Consequently, if you request more characters to be read than have been transmitted, the computer will wait indefinitely until the requested number of characters are read (unless a timeout is specified or an EOS character is detected). Likewise, if a write is held off due to a handshake situation, the computer will wait indefinitely until the hold off condition is released. To avoid this problem it is important to follow these guidelines:

### Reads

- ▼ Use the *Bytes at Serial Port* VI to determine the number of bytes that can be read. The proper use of this VI will also increase the

execution efficiency of your code since no time will be spent in the MultiCom software waiting to data to be received.

- ▼ Use the *Avanstar Serial Port Read+* VI with timeout enabled to prevent LabVIEW from hanging if no data is available.

## Writes

- ▼ Use the *Bytes at Output Port* VI to determine if there is room in the output buffer for the new output string. The proper use of this VI will also increase the execution efficiency of your code since no time will be spent in the MultiCom software waiting for the output buffer to empty.
- ▼ Use the *Avanstar Serial Port Write+* VI with timeout enabled to prevent LabVIEW from hanging due to handshaking.

If LabVIEW hangs in one of these wait loops, press <Ctrl><Alt><Del> followed by <Enter> to shut down LabVIEW and return to Windows.

## Serial Port Break VI

The break generated by the Avanstar board is a fixed time length. Although the MultiCom Serial Port Break VI includes a delay terminal for compatibility with the original LabVIEW VI, the value passed is ignored by the Avanstar board.

## Parity Error Byte in the Flow Control Cluster

If the high-order byte of the Parity Error Byte field in the Flow Control cluster in the serial initialization routines is non-zero, MultiCom will insert an FFh into the input buffer (not the low order byte of the Parity Error Byte field as in LabVIEW) when a parity error is detected. The low-order byte of the Parity Error Byte field is not used by MultiCom.

## Error Codes

---

Listed below are the error codes returned by the MultiCom VIs. The VIs are listed under the VI names in SERIALAV.LLB. The VIs in SERIAL.LLB return the same error codes as their equivalent functions in SERIALAV.LLB.

All MultiCom VIs return the following error codes

- 0 – no error
- 1 – unable to access board or control program
- 2 – unable to access board or control program

Additionally, the following VIs return the error codes listed below:

### **Avanstar Serial Port Init.vi**

### **Avanstar Serial 422 Port Init.vi**

### **Avanstar Serial Port Init+.vi**

- 3 – illegal baud rate specified
- 4 – illegal number of data bits specified
- 5 – illegal parity specified
- 6 – illegal number of stop bits specified
- 7 – board timeout - board did not respond
- 8 – RS-422 not supported
- 9 – External connection panel not installed
- 10 – Not enough room for requested buffer size

### **Avanstar Serial Port Read+.vi**

### **Avanstar Serial Port Write+.vi**

- 1 – timeout exceeded

## Avanstar Control Program Loader.vi

-11 – error downloading the MultiCom control program

-12 – no boards configured

## How to Reach Us

---

Viewpoint Software Solutions, Inc.  
2320 Brighton Townline Road  
Rochester, NY 14623-2708

phone: (716) 475-9555  
fax: (716) 475-9645  
e-mail: [support@ViewpointUSA.com](mailto:support@ViewpointUSA.com)

Technical support is available by e-mail, fax or voice. Phone support is available business days, 9:00 a.m. to 5:00 p.m. Eastern time.

Before calling for technical support please double check your work. When calling, it is important to have all relevant information on hand.