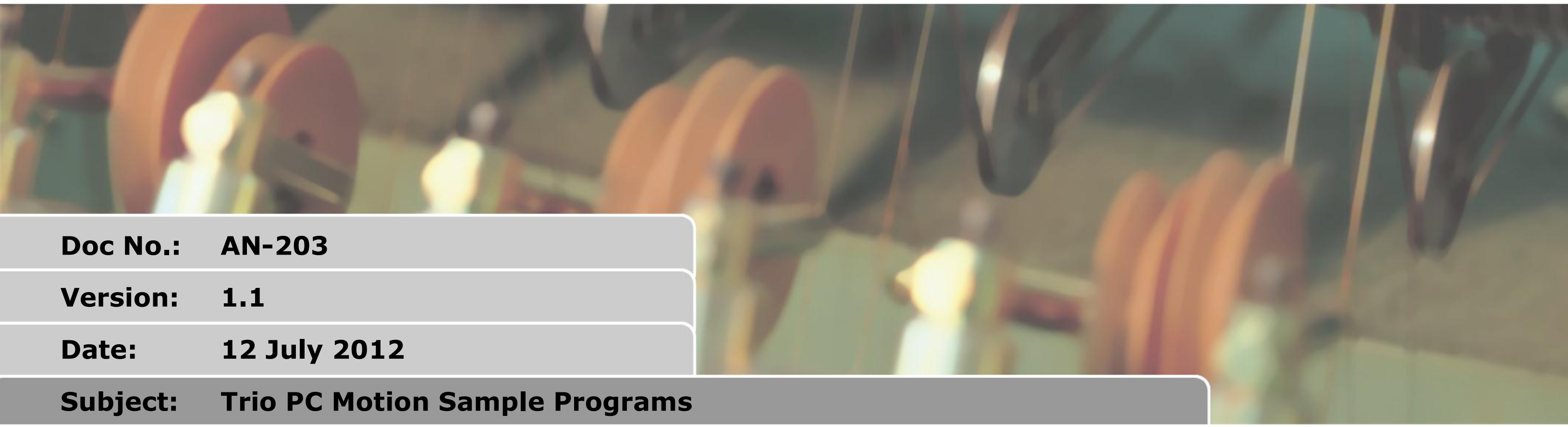
Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Tel: +1 412.968.9744
Fax: +1 412.968.9746

Tomson Centre
118 Zhang Yang Rd., B1701
Pudong New Area, Shanghai,
Postal code: 200122
P. R. CHINA
Tel/Fax: +86-21-58797659



Doc No.:	**AN-203**
Version:	**1.1**
Date:	**12 July 2012**
Subject:	**Trio PC Motion Sample Programs**

APPLICATION NOTE

www.triomotion.com

1. Trio PC Motion Sample Programs

The code samples are available as projects for the following development environments programming languages:

- Borland Delphi 7
- Microsoft Visual C++ (Visual Studio .net 2008)
- Microsoft Visual C# (Visual Studio .net 2003)
- Microsoft Visual BASIC .net (Visual Studio .net 2010)
- National Instruments LabVIEW

There are two examples per development environment:

- ActiveXTest - Shows the use of the TrioPC ActiveX control operating in synchronous (token) mode.
- AxAsyncTest - Shows the use of the TrioPC ActiveX control operating in asynchronous (stream) mode.

The examples assume that the user is familiar with the appropriate development environment and programming language. The examples were tested with version 2.9.0.0 of the TrioPC ActiveX control on a PC running Microsoft Windows 7.

2. ActiveXTest

This example shows how the control is used with different communications links using operation mode 0 (synchronous). In this mode the ActiveX control is used by calling methods which correspond to TrioBASIC commands. The commands are passed to the controller in a pre-tokenized form to improve efficiency. The code demonstrates the reading and writing of parameters, table values and global variables as well as how to program a sequence of moves. Most of the demonstrated functionality is the same for all of the programming languages.

2.1. Making a Connection

Use the combo box to set the type of communications link. Depending on the link selected enter values for host address (ethernet) or channel (PCI) in the appropriate edit box. Click on Open to open the link. The red box below the open button should turn green if the link opens correctly.

2.2. Status Tab

The Status tab shows the values of some controller parameters read from the controller.

2.3. Axis Parameters Tab

The axis parameters tab shows the values of axis parameters for a specified axis. Enter the axis number in the box then click on the refresh button the read the values from the controller.

2.4. IO Tab

The IO tab displays the value of the first 12 digital inputs.

2.5. Table Tab

The table tab allows reading and writing table values. Specify the range of table values using the start and entries edit boxes then click on refresh to read the values from the controller. To change a value click on it in the table grid and enter a new value. Clicking on the write button will then write the displayed section of table back to the controller.

2.6. VR's Tab

The VR's tab allows reading and writing of VR variables. Specify the range of VR's using the start and entries edit boxes then click on refresh to read the values from the controller. Clicking in the change VR button will bring up a dialog allowing you to specify a VR number and a value for it. When this dialog is closed the new value is written to the controller.

2.7. Directory Tab

This displays a directory listing for the controller.

2.8. Moves Tab

The moves tab runs through a fixed sequence of moves whilst displaying some axis parameters. The actual sequence of moves and the parameters displayed are differ between samples. The code demonstrates the use of timer events to control the move sequence whlie leaving the Windows application still active.

3. AxAsyncTest

This example shows the use of the ActiveX control operating in mode 1 (asynchronous). In this mode commands are sent to the controller as ASCII character streams. When operation in this mode the communications normally use the command line (channel 0) but by putting the controller into MPE(1) mode then multichannel communications becomes available with the command line on channel 0.

In order to demonstrate the difference between MPE(0) (normal) and MPE(1) (channel) modes. Type the command:

```
PRINT#5,”Hello”
```

Into the commands box then click on the send commands button after the communications link has been opened. Try this in both normal and channel modes.