

A P P L I C A T I O N N O T E

Doc No.: TN30-14
Date: Sep 2001
Version: 1.0
Subject: VB Examples using TrioPC OCX commands

The following example should be used in conjunction with the Trio_USB.pdf document, which explains the full instruction set for OCX control.

Connecting to the controller:



The TrioPC ActiveX control module must be selected as a VB Component and placed on the form. The following is an example of the code used to connect to and interrogate the controller type and firmware.

```

Dim Controller As Double
Dim firmware As Double
Dim PortOpen As Boolean
Dim a As Boolean
Dim b As Boolean
Dim strevent As String

' Attempt to open port and verify connection
PortOpen = TrioPC1.Open(0, 0)
If PortOpen = True Then 'Read the controller type and firmware version.
    lstconnect.AddItem "USB Connection Established"
    ' Request firmware and controller version
    Do Until (a=true and b = True)
        b = TrioPC1.GetVariable("CONTROL", Controller)
        a = TrioPC1.GetVariable("VERSION", firmware)
    Loop
    firmware = Left(firmware, 4) 'take off trailing zeros
    ' Update information in a list box
    lstconnect.AddItem "Controller Found " & "MC" & Controller
    lstconnect.AddItem "Firmware Version " & "V " & firmware
Else
    ' Cant Establish a connection. Show Message and Close program.

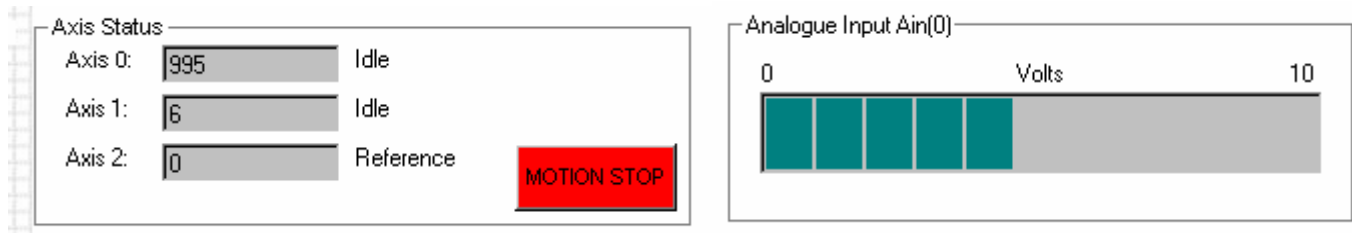
```

```

    MsgBox "Connection Failed"
    Unload frmSplash
End If

```

Setting and reading system variables



We can set and read system variables from the controller by using the GetVariable and SetVariable TrioPC1 extensions. The following example retrieves the axis encoder positions MPOS from each axis and displays them in a text box.

```
Dim position(0 To 2) As Double
```

```

With TrioPC1
    'request the value of each axis position (MPOS)
    .Base 1, 0
    .GetVariable "MPOS", position(0)
    .Base 1, 1
    .GetVariable "MPOS", position(1)
    .Base 1, 2
    .GetVariable "MPOS", position(2)
    'read value of analogue i/P
    .Ain 0, anin

End With
'Display 5 Significant figures of each position in text box
txtaxis(0).Text = Left(position(0), 5)
txtaxis(1).Text = Left(position(1), 5)
txtaxis(2).Text = Left(position(2), 5)

' Display value of Analogue as progress bar
If Abs(anin) < 2048 Then
    pbaranalogue.Value = Abs(anin)
End If

```

Reading inputs and outputs

IO Status			
Outputs		Inputs	
8	<input checked="" type="checkbox"/>	12	<input checked="" type="checkbox"/>
9	<input type="checkbox"/>	13	<input type="checkbox"/>
10	<input checked="" type="checkbox"/>	14	<input type="checkbox"/>
11	<input type="checkbox"/>	15	<input checked="" type="checkbox"/>
		0	<input checked="" type="checkbox"/>
		1	<input checked="" type="checkbox"/>
		2	<input checked="" type="checkbox"/>
		3	<input checked="" type="checkbox"/>
		4	<input type="checkbox"/>
		5	<input checked="" type="checkbox"/>
		6	<input type="checkbox"/>
		7	<input type="checkbox"/>

Since all outputs are bi-directional, we can read from them as well as write to them. There is therefore only one command needed to retrieve I/O values. In this example the first 16 I/O are read into an array called `iovalues`. These are then displayed by filling an array of shapes on the form.

```
Dim iovalues(0 To 16) As Long
Dim i As Integer
```

```
'read the I/O values from the Trio controller in the range i to i.
For i = 0 To 16
    TrioPC1.In i, i, iovalues(i)
Next i
```

```
' Display values on form
For i = 0 To 15
    If iovalues(i) = 0 Then shpled(i).FillColor = &H8000000E
    If iovalues(i) > 0 Then shpled(i).FillColor = &HFF00&
Next i
```

Motion commands.

Quick Start Panel

MoveRel	<input type="text" value="1000"/>	Axis 1	<input type="text" value="0"/>	<input type="button" value="Start"/>
MoveAbs	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="Start"/>	
MoveCircle	<input type="text" value="0"/> (Diameter)		<input type="button" value="Start"/>	

Axis(0)

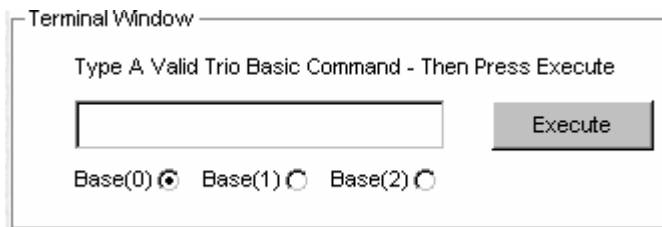
Forward	Stop	Reverse
---------	------	---------

The basic motion commands are all executable from the OCX control. The follow shows an example of a two axis relative move.

```
Dim X(2) As Double'           Define array for positions
X(0) = txtmoverel0.Text
X(1) = txtmoverel1.Text
'relative move on axis 0 and axis 1 defined by the array X
```

Executing other Trio Basic commands

When a Trio basic command needs to be executed but no specific OCX command exists, the EXECUTE command can be used.



In this example a Trio basic command is typed into the text box and sent to the controller. For ease of use 3 radio buttons have been used to direct axis specific commands.

```
' This routine sends immediate commands typed into a text box.  
' The text box is cleared if the command is recognised by the controller  
' If an invalid string / command is sent the string remains in the textbox.  
' and a Message Box warning is displayed.
```

```
Dim sent As Boolean
```

```
With TrioPC1
```

```
    If optbase(0).Value = True Then  
        .Base 1, 0  
        sent = .Execute(txtcommand.Text)  
    End If
```

```
    If optbase(1).Value = True Then  
        .Base 1, 1  
        sent = .Execute(txtcommand.Text)  
    End If
```

```
    If optbase(2).Value = True Then  
        .Base 1, 2  
        sent = .Execute(txtcommand.Text)  
    End If
```

```
    If sent = True Then  
        txtcommand.Text = vbNullString  
    Else  
        MsgBox "Command not recognised"  
    End If
```

```
End With
```