

Using the 3499A/B/C Switch with the 34401A DMM in Visual Basic

Overview

This Technical Note describes programming in Visual Basic for the 3499A/B/C Switch, N2260A 40-Channel Multiplexer and the 34401A DMM. Using the 34401A as the Measuring instrument, and the 3499A/B or C with a N2260A Multiplexer provides up to 360 channels using the 9 slot 3499C. The instrument commands are demonstrated in Visual Basic 6.0. Additionally examples in Visual Basic 6.0 and in Excel are available for download. All are in a single .zip file.

This Technical Note demonstrate how to:

- Close or Open a relay
- Monitor a channel with the DMM
- Connect the 34401A to the 3499 trigger output to trigger the DMM when a relay is closed
- Initiate a relay closure in the 3499 that triggers the DMM
- Initiate a scan of multiple channels and read the data from the DMM

Available for download is:

- Technical Note that describes **Using the 3499A/B/C Switch with the 34401A DMM in Visual Basic** .
- Example code in Microsoft ® Visual Basic 6.0 and Microsoft ® Excel 97 or later that demonstrates fundamental measurement techniques. See the end of this document for descriptions of the example code.

Difficulty

The Visual Basic and Excel programming examples discussed are of **intermediate** difficulty. They require knowledge of the Visual Basic 6.0 development environment, the use of referenced objects, and instrument IO. Some knowledge of the 3499A/B or C and the 34401A instruments and the intended application is assumed.

Programming the 3499A/B/C and 34401A

To program these instruments requires a GPIB interface or a RS232 interface on the PC. The examples use the Agilent ISDK (Instrument Software Development Kit) in MS Visual Basic 6.0 with GPIB. Other libraries such as VISA may be substituted by using the instrument commands shown. The Agilent IntuiLink IO Object library will work for GPIB cards or USB-to-GPIB converter from either Agilent or National Instruments. The Agilent IO Objects can be loaded to your PC by loading the **IntuiLink for DMM 34401A** found at www.agilent.com/find/intuilink. Once you find the **IntuiLink** web page, select **Software, Firmware and Drivers** on the bottom of the page, then download and install **IntuiLink for DMM 34401A**. To use GPIB, you must have a GPIB interface installed and working prior to using this software. To use RS232A you must install the Agilent ISDK or any other Agilent IO Library.

Connecting

Making a connection requires a reference to **Agilent I/O Manager**. From the **Project, References...** menu in VB select **Agilent I/O Manager** to create the reference for your project. The code to establish a connection looks like this:

```
Dim DMM As AgtIOServer
Dim Switch As AgtIOServer
Dim io_mgr As AgilentIOUtilsLib.AgtIOManager

Set io_mgr = New AgtIOManager
Set DMM = io_mgr.ConnectToInstrument("GPIB::22")
Set Switch = io_mgr.ConnectToInstrument("GPIB::9")
```

Once the above code is executed the object DMM and Switch can be used to talk to the instrument. This code demonstrates how to retrieve the instrument error from the DMM:

```
Dim Reply As String
DMM.Output "Syst:Error?"
DMM.Enter Reply
```

Setting up a Function on the 34401A DMM

When setting up a function that we want to monitor at a later time use the Config command. The following code sets up the 34401A, range 10000 ohms with an integration time of 10 line cycles (6 digit resolution). Once the Configure command is sent, the Sense command can be used to refine the setting. The Sense command is optional. Without the Sense command, the settings use default values of the Configure command.

```
DMM.Output "Configure:Resistance 1E4"
DMM.Output "Sense:Res:NPLC 10"
```

Closing a Relay on the 3499A/B/C

To make a measurement the correct relay must be closed. You can close one or several relays of the MUX card with one command. Notice there is a space between the word Close and the first parenthesis and the word Route is optional.

```
Switch.Output "Route:Close (@301:305,309)"
```

The syntax 301:305 acts on channels 301-305. Open the same relays with the Route:Open (@301:305,309) command. To open all the relays send:

```
Switch.Output "Route:Open ALL"
```

. To be sure all relay operations are complete use this code and throw away the returned value.

```
Switch.Output "*OPC?"
Switch.Enter Reply
```

Reading Data with the 34401A and 3499A/B/C

There are three fundamental ways to read the data:

1. Single reading, no hardware trigger; To monitor a single channel, close the relay on the 3499 Switch to route the signal. Then take a reading with the voltmeter.
2. Single reading, hardware trigger to DMM; when the relays are closed, the 3499 triggers the DMM to take a reading. In this mode the 34401A provides a default delay dependent on the function and range.
3. Multiple readings, Bi-directional trigger: When a relay closes the 3499 triggers the DMM to take a reading. The DMM triggers the 3499 when the reading is complete which allows the 3499 to progress to the next relay and so on. This allows for successive multiple readings with the same function.

Method 1, Single reading, no hardware trigger;

To monitor a single channel, close the relay on the 3499 Switch to route the signal. Then take a reading with the voltmeter. The two are independent operations. Any delay for settling is provided by the code between relay closure and when the reading on the DMM is initiated.

```
Dim Reply As String

' Close the required relay and wait until it is done
Switch.Output "Route:Close (@105)"
Switch.Output "*OPC?"
Switch.Enter Reply

' Set to resistance and take a reading
DMM.Output "Configure:Resistance 1E4"
DMM.Output "Sense:Res:NPLC 10"
DMM.Output "Read?"
' return reading as a string
DMM.Enter Reply
```

Method 2, Single reading, hardware trigger to DMM

When closing a relay and then reading the channel with a DMM, a delay for settling is required. By connecting the trigger out of the 3499 Switch to the trigger in of the DMM, the 34401A provides a default delay dependent on the function and range.

```
Dim dummy As String
Dim reading As Double

' Set DMM for DC, 10V range 51/2 digit resolution
With DMM
    .Output "Conf:volt:DC 10::Volt:DC:NPLC 1"
    .Output "Trig:Sour ext"
    .Output "*Opc?"
    .Enter
    .Output "Init"
End With

' Program the 3499 for one channel scan with DMM triggered by the 3499
With Switch
    .Output "Conf:Ext:Trigger:Output 1"
    .Output "Scan (@100)"
    .Output "Init"
End With
```

```

With DMM
    .Output "Fetch?"
    .Enter reading
End With

```

Method 3, Multiple readings, Bi-directional trigger

When there are multiple channels with the same DMM configuration the 3499 and 34401A DMM can be set to pace each other by connecting the trigger in and out of both instruments. Once these connections are made, the 3499 and 34401A will take a series of paced readings without any programming interaction.

```

Dim dummy As String
Dim readings() As Double
Dim readingCount As Long

' Program the 3499 for one scan with DMM synchronized to 3499
With Switch
    .Output "Conf:Ext:Sour 0"
    .Output "Conf:Ext:Outp 1"
    .Output "Arm:sour Imm::Arm:Count 1"
    .Output "Trig:Sour Mix"
    .Output "Scan (@100:102,105)"
    ' let the 3499 tell us how many channels
    .Output "Scan:Size?"
    .Enter readingCount
End With

With DMM
    .Output "Conf:volt:DC 10::Volt:DC:NPLC 1"
    .Output "Trig:Sour ext::Trig:count " & readingCount
    .Output "*Opc?"
    .Enter dummy
    .Output "Init"
End With

```

Once this code is executed the following sequence takes place:

1. Close the first channel in the scan list
2. 3499 triggers the DMM to take a reading
3. The DMM starts a default delay depending on range, and function
4. DMM takes a reading after the delay and stores the reading in memory
5. DMM triggers the 3499 to close the next relay on the scan list
6. 3499 closes the next relay
7. repeat from #2 until all relays in the scan list are completed.

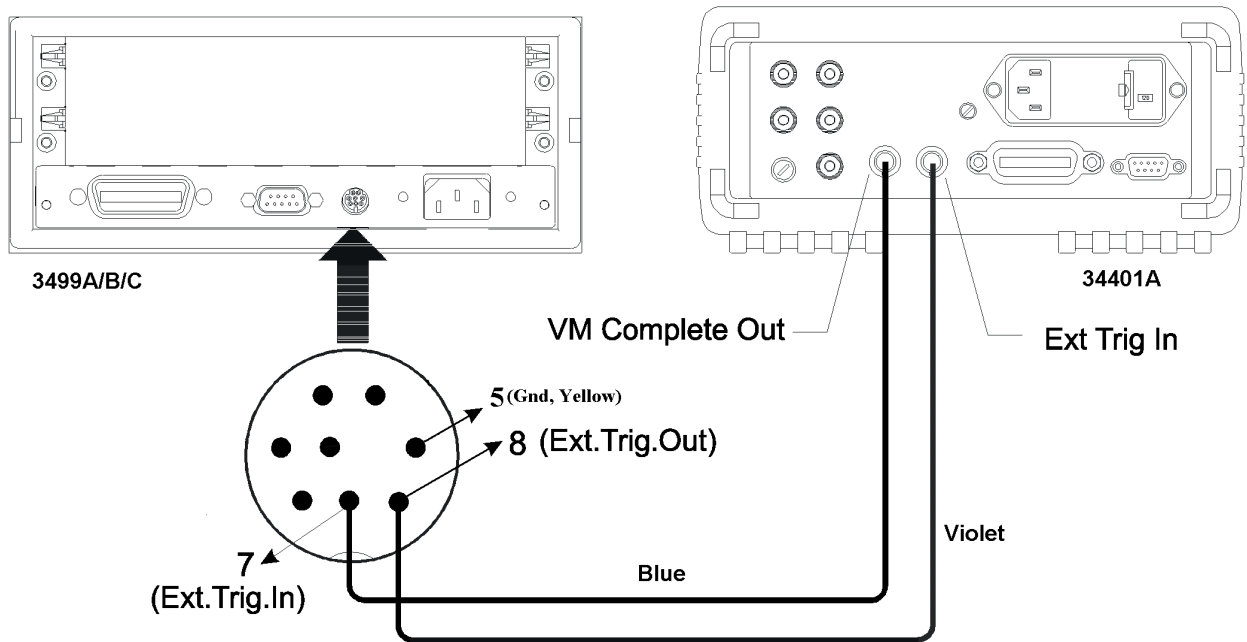
Once this sequence is complete, the readings for all the channels in the scan list are available in memory in the 34401A. Read the data array with this code. For long channel lists and DMM measurements that take a long time, a delay or a longer timeout may be required before executing this code. For best performance, avoid multiple configurations on the DMM by combining channels with the same functions and configuration.

```

With DMM
    .Output "Fetch?"
    .Enter readings ' read into array from memory
End With

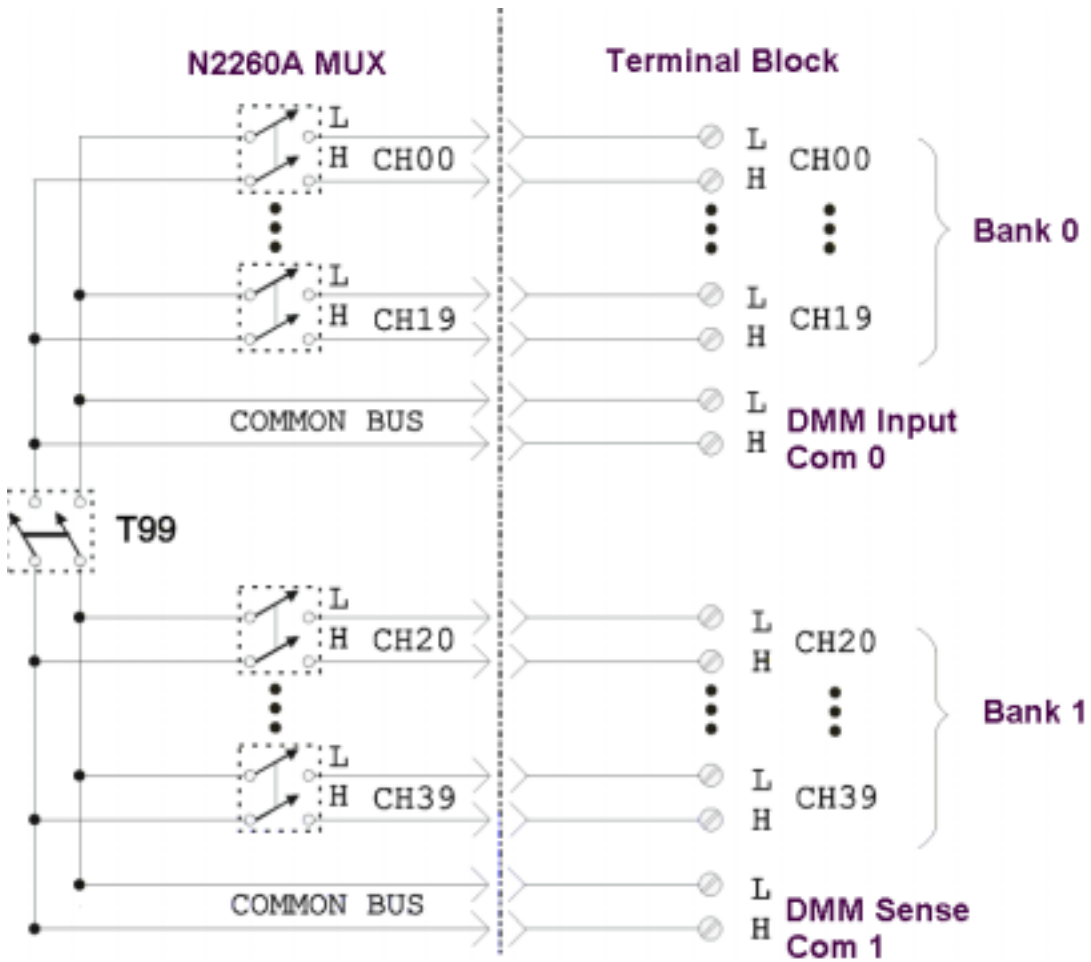
```

Connecting Trigger for 3499 Switch and 34401A DMM



To perform a scan with the 3499 Switch and the 34401A DMM connect the two instruments as shown. You can create the cable shown by purchasing the Agilent N2289A cable, cut the DB9 connector on one end and connect BNC connectors to the Blue and Violet wire. Connect all the shields together and connect to ground. The connector that fits into the 3499A/B/C is a Miniature DIN in-line plug, 8 positions (Digi-Key PN CP-2080-ND).

Connecting the MUX to the DMM



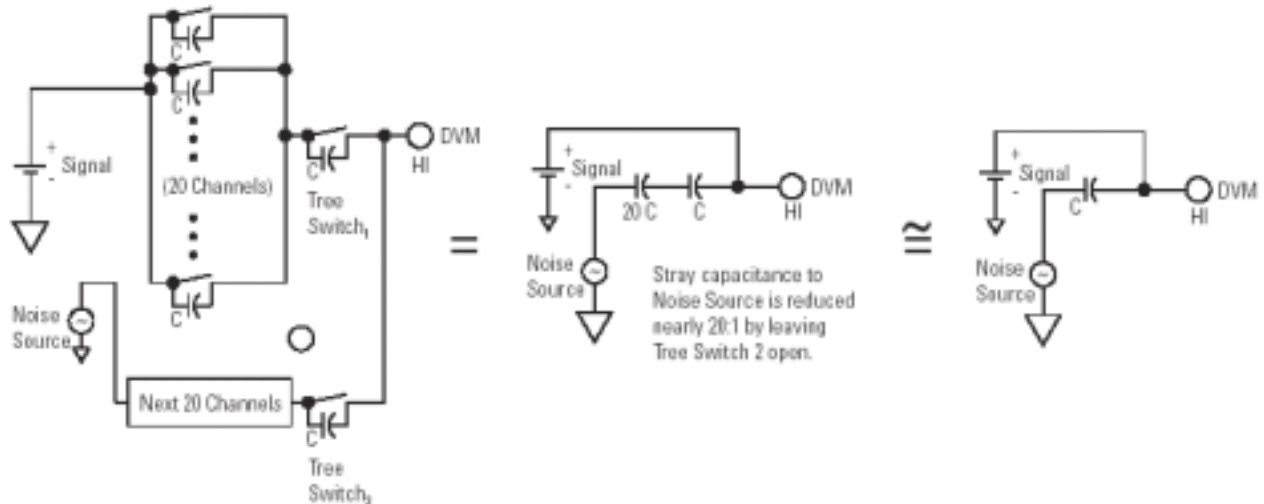
Connect the DMM Input to the Common Bus 0 of the N2260A module. If 4 wire ohms measurements or RTD temperature measurements are to be made, also connect the DMM Sense terminals to the Common Bus 1 of the N2260A module.

Other Considerations

Tree switching

Tree switching is a method of organizing the channels of a scanner into groups, each with its own main switch.

Without tree switching, every channel can contribute noise directly through its stray capacitance. With tree switching, groups of parallel channel capacitances are in series with a single tree switch capacitance. The result is greatly reduced crosstalk in a large data acquisition system, due to the reduced inter-channel capacitance.



Temperature

When measuring temperature, the 34401A is used to measure the resistance for RTD and thermistor sensors and voltage for thermocouples. These values are then used to calculate the temperature in software. In the example software, routines for 5k ohm thermistor, 100 ohm Ro85 RTD, and J type thermocouples are included.

For thermocouple measurements the thermocouples must be terminated at an isothermal terminal block. The temperature of the isothermal terminal block is measured using a thermistor or other temperature device. This reference temperature value is used in the calculation of the thermocouple temperatures. One Agilent E1586A isothermal terminal block is suitable for up to 39 channels of thermocouples plus one reference channel when used with the 3499 Switch and one N2260A MUX relay card. For more information on temperature measurements go to the <http://agilent.com> web site and search on **temperature** and from the **Application and Technical Paper** category (scroll down) select **Practical Temperature Measurements**

SRQ

When making measurements with a large scan list and functions that require long settling or take an appreciable amount of time, using a delay or increasing the timeout of the IO will capture the bus and the PC screen. Under these circumstances a more suitable way to take the reading is to set the DMM to give an SRQ when the reading are complete. This frees up the PC and GPIB bus resources for other tasks while the 3499 Switch and 34401A DMM are busy with scanning and measurements. If an Agilent GPIB card is installed, this can be done with an SRQ event in Visual Basic. When an SRQ event is enabled, the event will fire when the scan is complete, and the readings can be read from the DMM. Alternately the DMM can be polled on GPIB every fraction of a second and asked if the SRQ bit is set. Once the SRQ bit is set, the data can be read from the DMM. The examples in the **34401A SRQ** directory demonstrate both techniques using the Agilent ISDK objects.

Wait or Delay

When using the DMM and a MUX Switch card it is desirable to add delays for settling. Visual Basic does not implement a delay or wait function. You can implement a delay by declaring the windows Sleep API and then calling the sleep function. This code will cause a delay of 500 msec.

```
Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Sleep 500
```

Example Visual Basic Code

The .zip file available along with this Technical Note contains example software in Visual Basic 6.0. When extracting the examples into the root of the C:\ directory the examples will be placed into the directory:

C:\Program Files\Agilent\IntuiLink\3499\Examples.

Unless otherwise indicated, these examples use the General Purpose set of ActiveX control and IO objects installed with the **IntuiLink for Meters** www.agilent.com/find/IntuiLink. These objects are part of the Agilent Instrument Software Development Kit (**ISDK**) for VB 6.0 that is also available from the Professional Subscription of ADN at www.agilent.com/find/adn

EZ_relay – Demonstrates how to open and close a relay

MatrixSwitch – A graphical user interface for a Matrix Card that demonstrates how to open and close a relay, how to read the state of a relay and how to read the module type installed in the 3499

MUX_Switch – A graphical user interface for a MUX Card that demonstrates how to open and close a relay, how to read the state of a relay and how to read the module type installed in the 3499. Requires N2265A matrix module.

Switch_DMM_scanExample – This directory contains several different programs. All are functionally the same, except they are implemented using different IO in **Visual Basic 6.0**. The Example program demonstrates the three methods described above in **Programming the 34999A/B/C and 34401A**

ISDK – uses the General Purpose set of ActiveX control and IO objects installed with the **IntuiLink for Meters** www.agilent.com/find/intuilink and provided on the CD shipped with the 34401A.

NI_488 – Uses the National Instrument NI-488 syntax that comes with the GPIB interface device.

SICL – Uses the Agilent SICL syntax that comes with the Agilent GPIB interface device.

VISA – Uses the industry standard VISA syntax that comes with some GPIB interface cards

Temperature – Demonstrates making temperature measurements using a DMM. Contains the equations/routines for Thermistor (5k, 10k ohm), RTD, and thermocouple (type J,K,T)

Excel – The **Switch_DMM_scanExample** is duplicated in Excel using the ISDK for VB 6.0. Load the Excel file Switch_DMM_scanExample to see the three buttons. To see the example code click on Alt-F11. The module modScanExample is the same as in the ISDK Visual Basic Example in the directory Switch_DMM_scanExample.

