PC-BASED TEST

Making a Strip Chart Recorder in Excel

Werner Haussmann
Agilent Technologies

As I collect data, I like to import it into Excel for analysis. Fortunately, you can use Excel’s macro language, Visual Basic for Applications (VBA), to set up an instrument control program to transfer data directly into a worksheet. For instrument control, it’s convenient to use a program called TMW Control, which was explained in earlier articles in Footnotes 1 & 2. You can conveniently download the TMW control program at www.tmworld.com/57jun99.zip. I have designed two Excel spreadsheets that make use of the TMW control. One, called dmm.xls, turns a DMM into a temperature logger.

I used the other, called dmm_scope.xls, to plot both the DC voltage output of a power supply measured with a DMM and the supply’s ripple measured with an oscilloscope. Figure 1 shows data I captured with Excel and plotted in a chart. (Both spreadsheets are available at www.tmworld.com/software_from_articles.htm.) If you need to measure with more than one instrument, you’ll need a copy of the TMW control for each instrument. Once you set up the control, use the following procedure to test it before you use the spreadsheets.

Get Started
First, connect a DMM or a scope to the IEEE-488 port of your computer. Then, open a new Excel spreadsheet. If the Control Toolbox isn’t visible, click View and select Toolbars. Click on Control Toolbox to make it visible. To add the TMW control to your spreadsheet, click the More Controls button on the Control Toolbox (the button showing the hammer and wrench). That will bring up a list of additional controls registered with Windows. Scroll down the list and click on T&MW Instrument I/O Control.

Listing 1

Private Sub CommandButton1_Click()
    Dim reply As String
    dmm.output "Measure?"
    dmm.enter reply
    Cells(1,2) = "Voltage"
    Cells(2,2) = reply
End Sub
Your cursor will change to a crosshatch. Position the cursor somewhere in the Excel worksheet and then click and drag it; this will add the TMW control to the worksheet and slightly change its size to make it visible.

You can test IEEE-488 communication without writing any VBA code. To do that, you first need to open the custom Properties page for the control. The custom Properties page lets you set properties such as the IEEE-488 (GPIB) port and the instrument address. Different versions of Excel let you access the window differently.

Click the TMW control on your worksheet to highlight it, and then right-click on the control. If you see an option called **TMW Instrument I/O Control Object**, select it. You'll see a pop-up menu that says “Properties”. Click on it, and the custom Properties page will open.

If you don't see the **TMW Instrument I/O Control Object** option, press Esc to clear the window. Click on the TMW control to highlight it, and then click on the Properties option in the Control Toolbox. You’ll see a standard VB properties window. You can then click on the word “Custom” in the list, which will cause an ellipsis (...) to appear. Click on the ellipsis to open the custom properties page.

Once the custom Properties page appears, set the GPIB address. Then, click on the Instrument tab. You'll see an Output-Command field, where you can enter a command such as “**IDN?”** Send this command to the instrument by clicking the Output button. To view the instrument's response, click the Enter button. If data appear in the Enter-Response field, you’re communicating with the instrument. (The article in Footnote 1 explains this process in more detail.)

Once you've verified that Excel communicates with your instrument, you're ready to add some code to get data into a worksheet cell. Assume you have a DMM you want to control from Excel. Click on the TMW control to select it. Next, click on the Properties tool in the Control Toolbox to open the standard VBA properties page. Here, you'll use the selections you made from the custom Properties sheet. In the (Name) field, rename the control “dmm”.

Now, you need to add a command button to the worksheet. Click the Command button on the Control Toolbox; your cursor will turn into a crosshatch. Then, move your cursor onto the worksheet and click again. A button labeled CommandButton1 will appear. After you place the button on your worksheet, double-click on it. You'll see a VBA routine to which you add the code in **Listing 1**. This code will then run when you click on the button.

To test the code and the button, click the “Exit Design Mode” button on the Control Toolbox. Click the Command-
Listing 2

Sub update()
    ' Using the TMW control on this sheet, get
    ' the frequency
    ' and place in cell 'A1'
    ' set the timer to take the next reading

    If continue Then ' take the reading
        ' set the time for the next reading
        nextTime = Now() + TimeValue(strInterval)
        Application.OnTime nextTime, "Sheet1.Update"
        Cells(Row, 1) = Format$(Now, "hh:mm:ss")

    measureDC Row, 2
    measureRipple Row, 3

    Row = Row + 1
    End If
    DoEvents
End Sub

Sub measureDC(Row, col)
    Dim reply As Double

    ' query meter and get reading
dmm.Output "Measure?"
dmm.Enter reply

    ' put reading into cell
    Cells(Row, col) = reply

End Sub

Sub measureRipple(Row, col)
    Dim reply As Double

    ' query scope and get reading
scope.Output "Measure:VPP?"
scope.Enter reply

    ' convert reading to mV and put into cell
    reply = reply * 1000
    Cells(Row, col).Value = Format$(reply, 
        "##0.0")

End Sub

Button1, and you should see the word “Voltage” appear in cell B1 and see a measurement in cell B2. Note that when programming, you must refer to cells using the syntax Cells(row, column) in which you refer to columns using numbers, not letters; thus, cell B1 becomes Cells(1,2) in the code.

At this point, you can transfer one data sample into Excel, but you probably need to analyse many measurements. Listing 2 contains the code I wrote to continuously capture data from a DMM and a scope and then store each pair of samples in columns. To obtain the complete code, download the dmm_scope.xls file from www.tmworld.com/sftw_files/software_from_articles.htm. The dmm.xls file (available at the same Web address) contains the code for capturing data from my DMM, but you can modify it to work with other instruments.
Once you use the control to transfer data from the DMM or scope into Excel, the spreadsheet automatically plots the data in a chart, similar to the way in which a strip-chart recorder works. If you aren’t interested in the automatic plotting, you can download the spreadsheet file you need, delete the chart, and save the file under a different name.

Because setting up charts requires an in-depth understanding of Excel, I haven’t provided instructions for creating them. You should keep a version of the spreadsheet with the chart in case you eventually decide you want it.

If you do want to use the chart, you can use the spreadsheet to set parameters for it. Column E in Figure 1 defines parameters for the chart. In Cell E2, you enter the number of most recent samples you want the chart to plot. Cell E1 counts the number of samples taken starting when you click on the Start button. As Excel gets the measurements, it will add them to the chart until the sample count equals the number of samples you entered in cell E2. (Until then, cell E3 displays “False” because the chart isn’t yet a strip-chart recorder.)

When the sample count in cell E1 reaches the number of samples in cell E2, cell E3 changes to “True” and the chart scrolls, plotting only the most recent number of samples entered in cell E2. In Figure 1, the system has taken 91 pairs of samples, but it displays only the latest 60 pairs.

The spreadsheets I’ve developed let you capture and plot data right away without having to learn the details of how Excel charts work. You can use the code as is, but you’ll probably want to modify it to suit your needs. For example, you might need to add conversion factors to convert a voltage output from a sensor to engineering units.

Footnotes


Werner Haussmann is an R&D project manager at Agilent Technologies. E-mail: werner_haussmann@agilent.com.

Get the Code

You can find the Excel spreadsheets and code listings for this article at:

www.tmworld.com/sfttw_files/software_from_articles.htm