PdCommX ActiveX Control -Demo Excel Application

This document accompanies an Excel/VBA file which demonstrates how Nortek's PdCommX ActiveX control can be integrated with Excel using VBA code and macros. Ensure that you have a copy of the Excel file to refer to.

Overview

The application does very little, it's purpose was to ensure that the PdCommX control works with VBA and to illustrate how you might use it in conjunction with Excel. The Excel file has a single worksheet, empty except for a button and some headings. The button will open a dialog which will allow you some basic control over an instrument and allow data to be written to the spreadsheet. The VBA source code is made available and is accessed by launching the Visual Basic editor and navigating to the source code in the usual manner.

Installation

Download the PdCommX file and follow the install instructions. The next time you run Excel and enter the Visual Basic editor the PdCommX control will be available on the controls Toolbox. The control can then be placed on a form the same way as other controls. Be sure that you have macros enabled, if you do not Excel will warn you. Follow the instructions to enable macros and open the file.

Functionality

The dialog has a set of simple functions activated by buttons or check boxes. The column of edit boxes on the left display basic information from the instrument or control.

Get 'Get Config' button will populate some or all of these boxes depending upon the current mode.

Connect button connects to the instrument and enters command mode.

Disconnect button will break the connection to the instrument.

Start Data Collection button puts the instrument into measurement mode and will cause the instrument to measure and output data.

Stop Data Collection button stops measurement and returns the instrument to command mode.

Real Time Data to Spreadsheet check box will cause the data being returned from the instrument to be written to the spreadsheet, one line per measurement interval; set to 1 second for this demo.

Clear Spread Sheet check box will clear the spreadsheet, data will be written to the spreadsheet below the heading.

Download Data button will display a “Save As” dialog and then use the PdCommX DownloadData method to transfer all of the data from the recorder and save to a file.

Exit button will quit the dialog.

Source Code

Not after any prizes for the source code and I’m sure there are better ways of achieving the same results.

There are just two routines or pieces of code that are worth noting, the first is:

UserForm_Initialize()

This is called by the application when the form containing the control is first displayed. All your one-time initialization stuff should go here.

The second is:

PdCommX1_OnNewData(ByVal htype As Integer)

This is called by the control each time it receives new data from then instrument. You write the bulk of your code to handle that data in here.

The remainder of the code handles the buttons and check boxes. If you’ve used VB/VBA before then you will know how this works. If not its simple enough. Place a control on your form, double click on the control and a small function will be created for you that gets called whenever you click that control when the form is running.

Control Properties

The properties window for the control also includes a large number of properties that are also instrument parameters. These can be changed as you would any other control properties but they are part of the control and not stored on the instrument. You need to do this within your own application to ensure that the instrument is updated with any modified properties. Use the SetConfig method to do this.