

The Visual Basic - Fieldbus Connection

Chris Jenkins - Proces-Data (UK) Ltd.

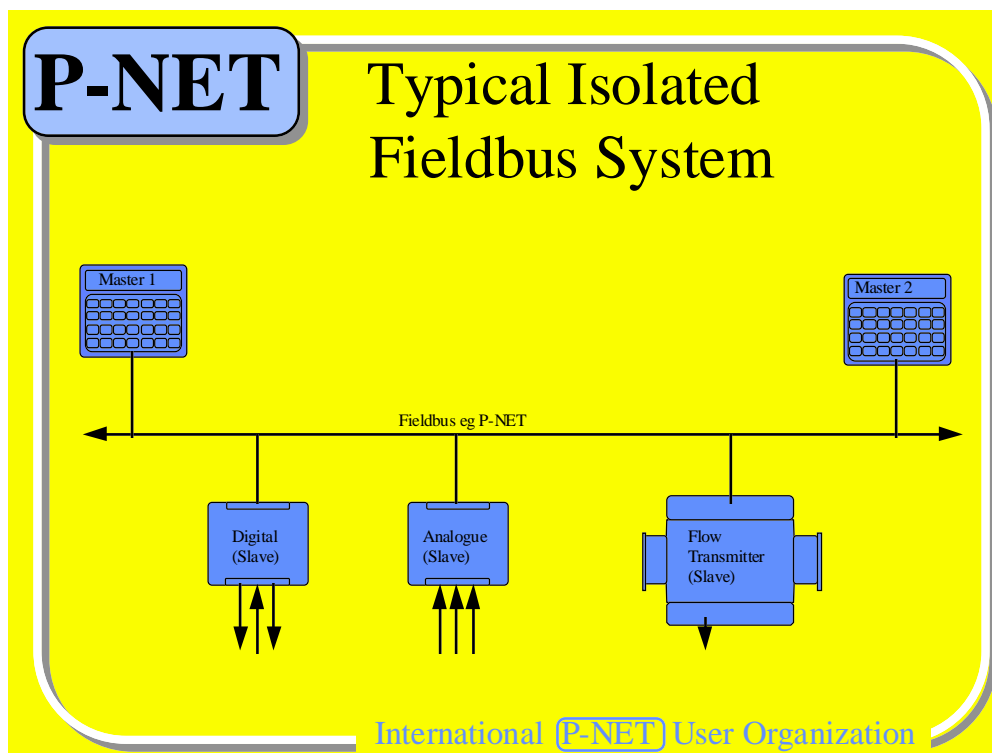
International P-NET User Organization

The paper describes a new Windows based communication package which enables standard Windows applications such as Excel and Access, and user designed applications written in object orientated programming languages, such as Visual Basic, to communicate with transducers and actuators via a fieldbus such as P-NET, together with other standardised fieldbus types.

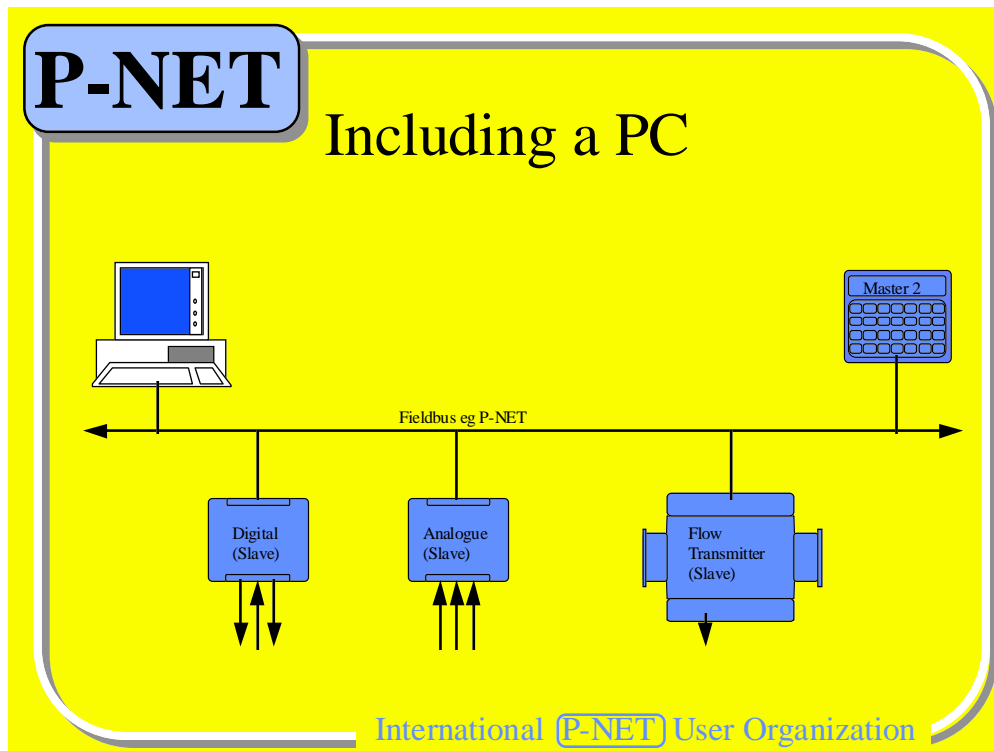
The paper includes examples of a simple Visual Basic programs, which display and control field variables on a PC screen using P-NET transducers and transmitters.

Introduction

In the world of fieldbuses, the stimulus for data communication is normally associated with measurement and control. This often requires equipment for display and activation of a process, in the form of some sort of screen and keyboard.

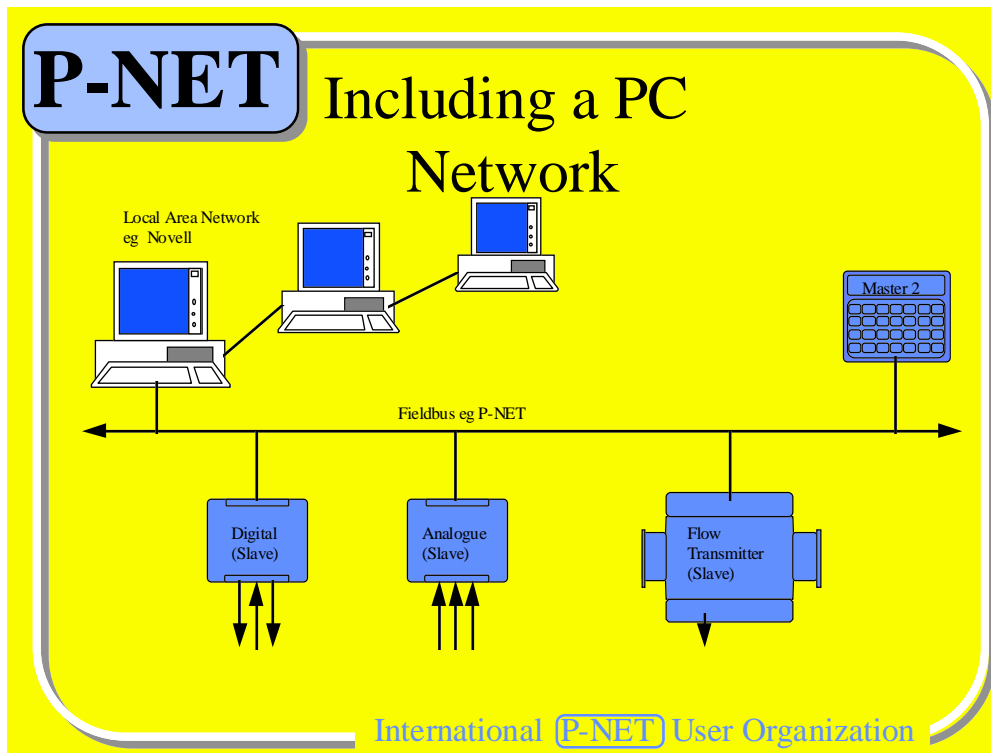


Although in less complex fieldbus systems, a single or distributed collection of field operator stations may be sufficient, there is an increasing tendency to incorporate a PC somewhere within the system.

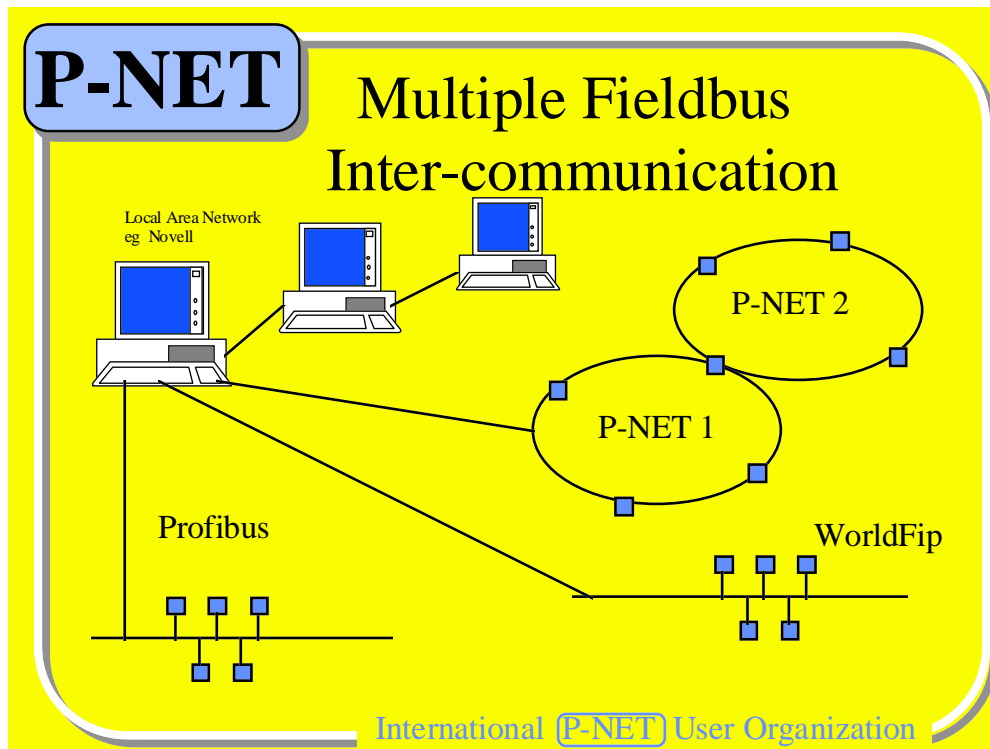


This has come about due to the PC's economic availability and escalating operator and management familiarity. It is therefore becoming increasingly important that mechanisms are provided to enable fieldbus data to be available within the PC environment.

Further, within complex process and manufacturing plants, it is highly likely that PC's will also be incorporated within a network, eg. Novell, to enable information from throughout the plant, to be available for monitoring or processing, throughout the management hierarchy.



Now that fieldbuses are coming of age (especially in Europe), with the advent of national and international standards and the ever increasing availability of equipment, leading to an accelerating node count, there is a demand for both fieldbus inter-networking, and the continued requirement to see information in a colourful and graphical form.



It is now becoming abundantly clear, that there is unlikely to be a single fieldbus standard available for some considerable time, (if ever). Therefore, there is also a requirement for a mechanism to provide inter-fieldbus communication, between those fieldbuses which have already been standardised and proved.

This paper is intended to show that practically anyone possessing a PC can make the Visual Basic to Fieldbus connection. I'm not talking about the high powered and expensive graphics packages (although they can also be linked to the fieldbus environment). This is about an ability to draw a graphical user interface and add a few lines of code, which haven't already been generated automatically, in order to produce a dynamic graphical screen which incorporates fieldbus variables, on a Windows based PC.

WINDOWS

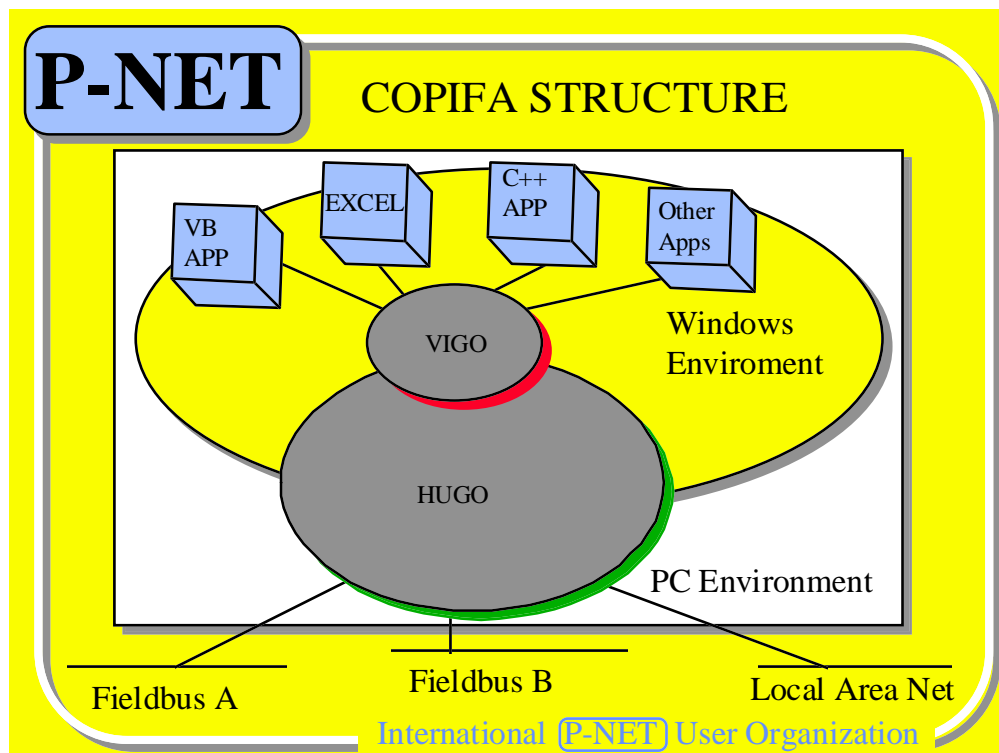
We start off by accepting that Microsoft Windows is the most widely used graphical operating system, not even considering the recent Windows '95 availability. Windows also incorporates Object Linking and Embedding Automation technology (OLE). This means that any standard or purpose written Windows application supporting OLE, can exchange data, and one application can be made to operate within another. Examples of Standard packages using OLE include Excel, Access, Powerpoint, Word, Wordperfect etc. Windows application languages that support OLE include Visual Basic and Visual C++.

VISUAL BASIC

It would be difficult to define how many Windows applications written in Visual Basic have already been developed, since the user doesn't normally need, or want to know this when using a package. However, it is worth noting that Visual Basic is also the macro language for the EXCEL spreadsheet application, and it is intended that it be the macro language for many other standard applications such as the Access database package and the Word word processor. Therefore, just on that basis, it's worth knowing a bit about the language, even if it's only to configure a spreadsheet.

The Visual Basic - Fieldbus Connection

Now that we have established that VB is a general purpose Windows application development language, how do we get a measurement on the screen? We need a link between Windows and the outside world. To Windows, this link must look like another Windows application, and to the outside world the PC must look like part of the fieldbus. Such a program package providing this mechanism is called the Common Program Interface for Fieldbus Applications or COPIFA, which has been developed by Proces-Data over the past 2 years. This is an OPEN application, in the sense that anyone can configure the package to interface between any Windows application supporting OLE automation and any Fieldbus or communications network. Inter-networking is now possible using P-NET, Novell, ARCNet, ODBC, and work is currently underway to include P-NET's partners in the proposed European standard EN50170 - Profibus and WorldFip, within the RACKS project.



COPIFA consists of two major modules, amongst others.

The first is called VIGO, which stands for Virtual Global Object, and is the initial interface between the Windows Application and the outside world.

The second is called HUGO, which in itself does not stand for anything in particular, but is the real-time communication kernel to interface with fieldbuses and other networks.

The simplified structure shown is intended to convey that Windows applications, be they written in Visual Basic or any other language which supports Object Linking and Embedding, communicates via VIGO.

HUGO then routes information to and from the fieldbuses and other networks.

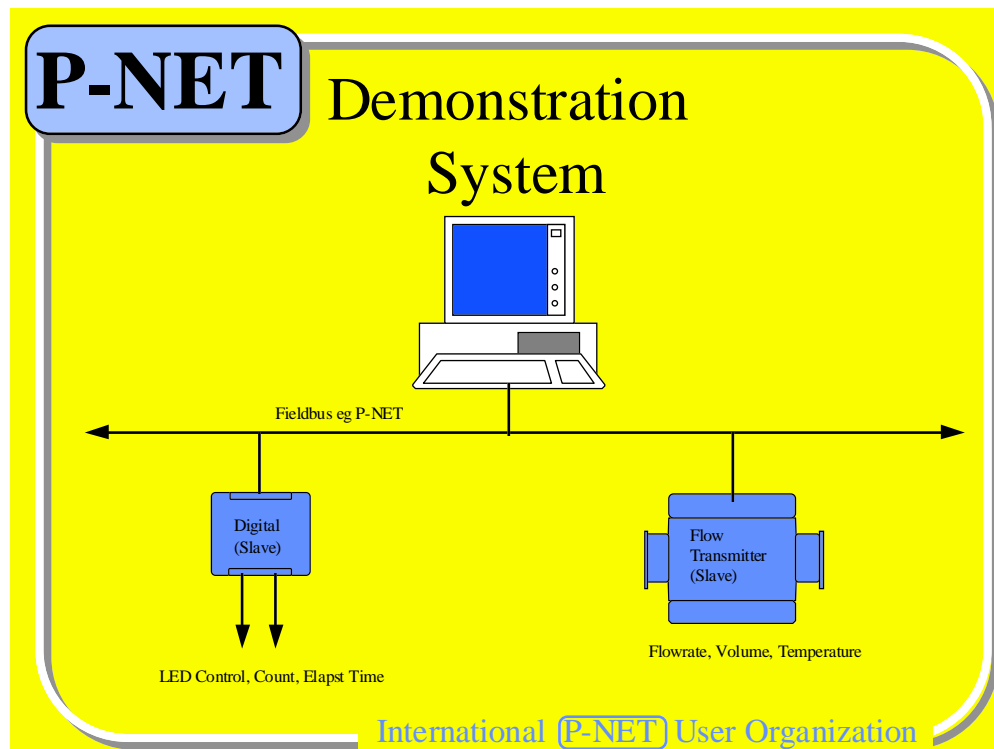
Thus, COPIFA is the enabling mechanism between the application and the fieldbus, and allows the application to act as a router, gateway, protocol converter, and/or a graphical user interface.

COPIFA runs in the Windows multi-tasking background, and is therefore transparent to the operational application.

Communication Example

The schematic of a simple system is shown below.

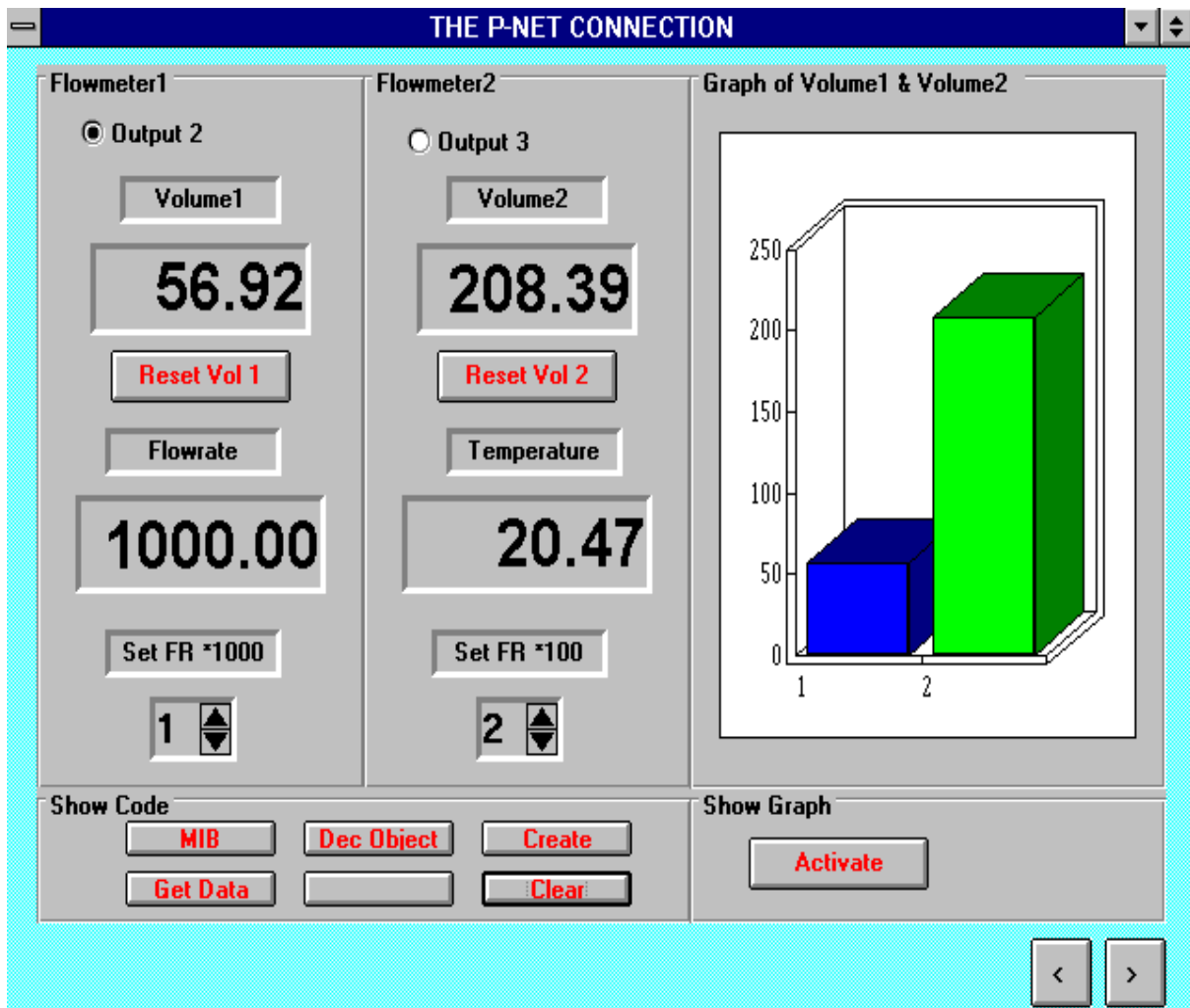
It consists of a PC, acting as a master, running a simple Windows graphical user interface, written in Visual Basic. The PC is connected to two fieldbus (P-NET) slaves.



One slave is a Flowmeter Transmitter, having the ability not only to measure flow and totalised batch and accumulative volumes, but also measure temperature, leading to internal temperature compensated flow. It has two digital outputs for pulse output or valve control and a 4-20ma output which can be used in conjunction with an in-built PI controller, for flow control.

The second slave is a general purpose digital input/output module, consisting of 16 channels. Each channel has the ability to store the state of the input or output, hold a count of I/O transitions, measure the time the output is active, and many other functions. In addition the module incorporates a programming channel, which allows the unit to act like a PLC.

For the purposes of demonstration, the flowmeter has been switched to test mode to simulate a selectable flowrate, and the digital module has been programmed to flash a couple of LED's on and off at a selectable frequency.



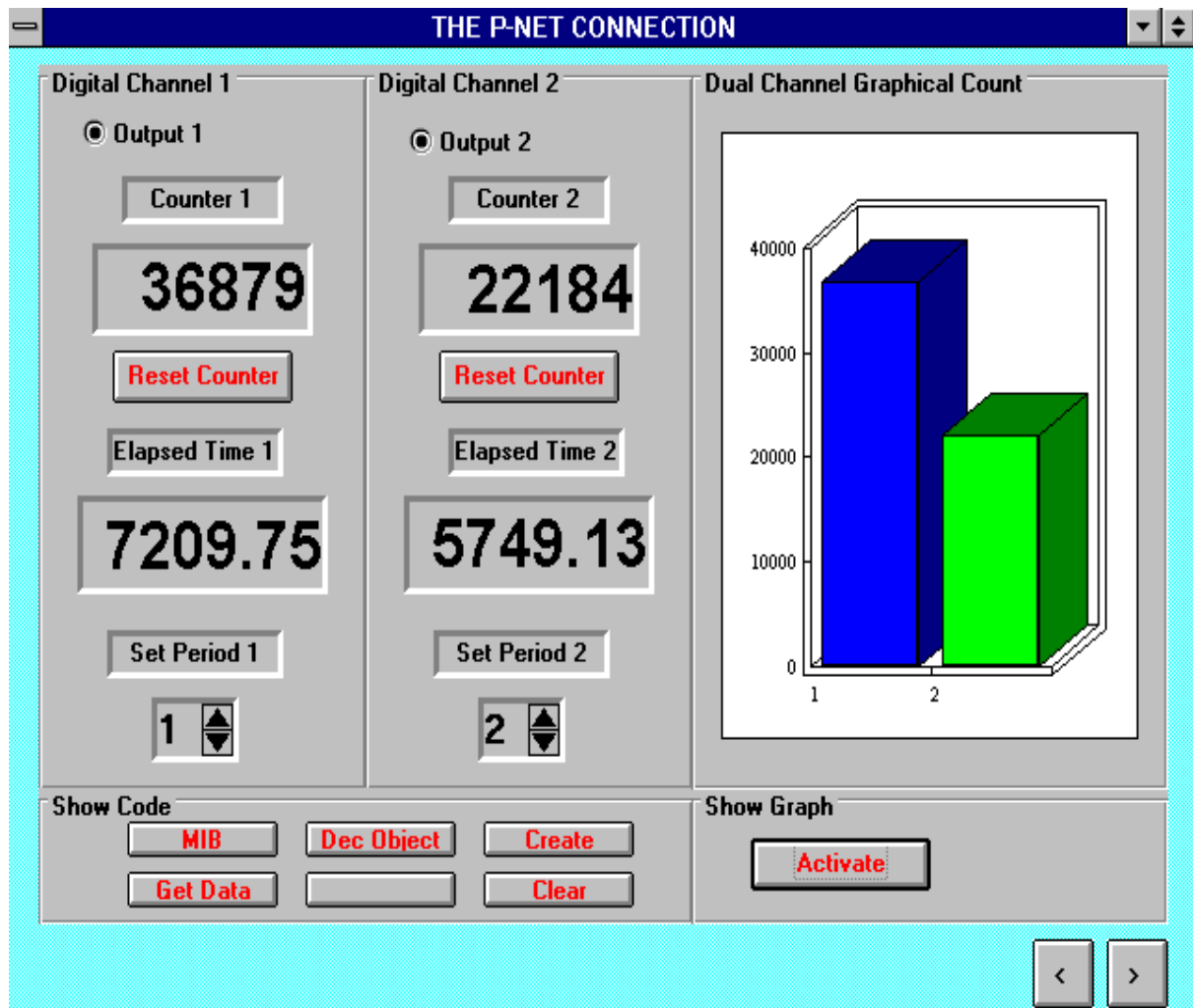
This Flowmeter display has been created using the graphical selection tools within Visual Basic.

It can be seen that the screen is being updated with the variables we are interested in.

It is also possible to control the fieldbus variables from the screen, such as resetting an external volume register.

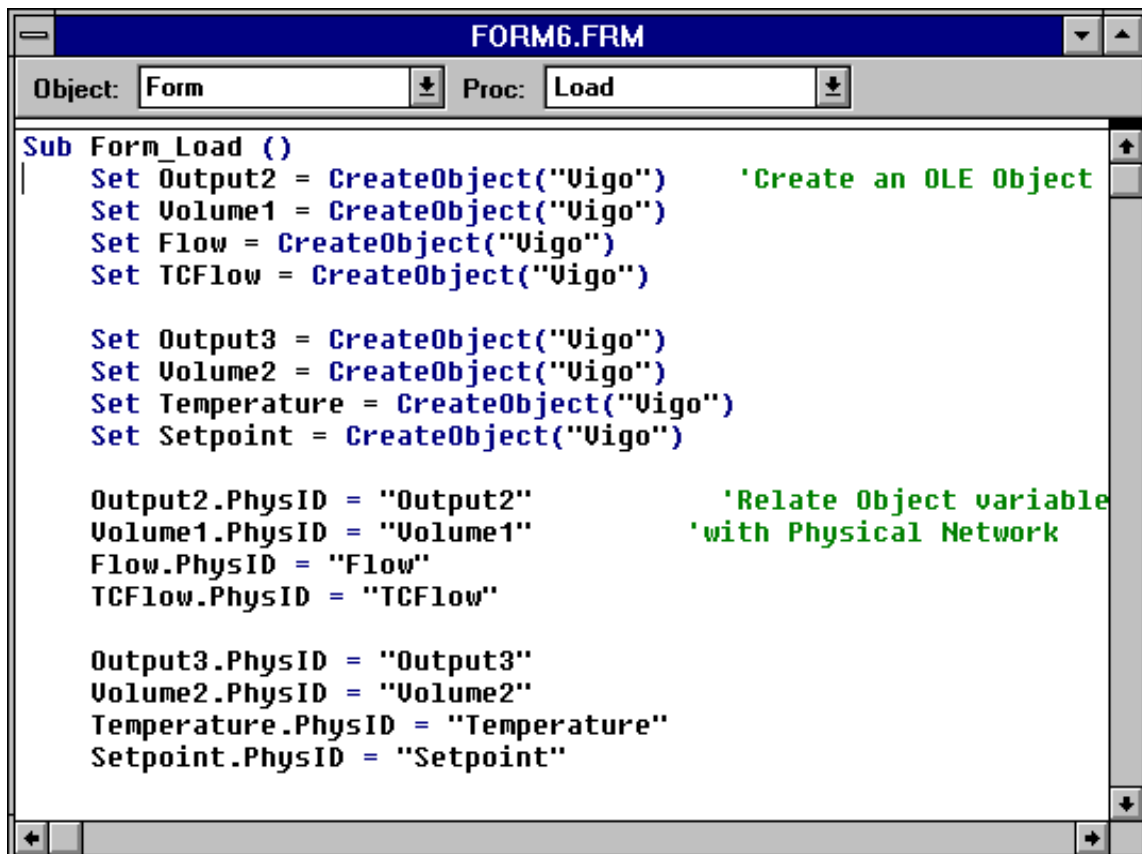
Data can also be displayed in a graphical form, by using a control available within Visual Basic.

The display of data from the Digital module has been generated by slightly modifying the



previous display, in terms of captions, and bringing in a different set of fieldbus variables. Of course, the flowmeter and digital data could have been displayed on the same screen.

The Visual Basic code required to make the connection, consists merely of specifying the variables which are to be used on the screen, which will be obtained via VIGO



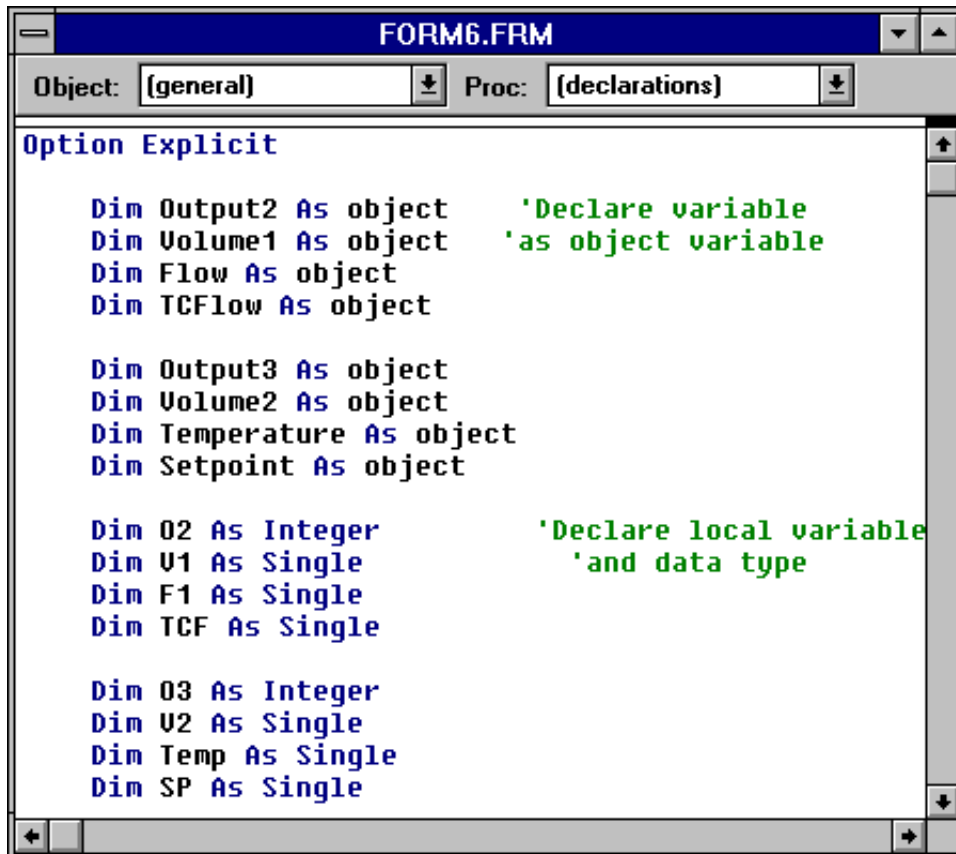
```
Sub Form_Load ()
    Set Output2 = CreateObject("Vigo") 'Create an OLE Object
    Set Volume1 = CreateObject("Vigo")
    Set Flow = CreateObject("Vigo")
    Set TCFflow = CreateObject("Vigo")

    Set Output3 = CreateObject("Vigo")
    Set Volume2 = CreateObject("Vigo")
    Set Temperature = CreateObject("Vigo")
    Set Setpoint = CreateObject("Vigo")

    Output2.PhysID = "Output2" 'Relate Object variable
    Volume1.PhysID = "Volume1" 'with Physical Network
    Flow.PhysID = "Flow"
    TCFflow.PhysID = "TCFlow"

    Output3.PhysID = "Output3"
    Volume2.PhysID = "Volume2"
    Temperature.PhysID = "Temperature"
    Setpoint.PhysID = "Setpoint"
```

Then using these as local variables within your Visual Basic Program



```
Option Explicit

Dim Output2 As object      'Declare variable
Dim Volume1 As object     'as object variable
Dim Flow As object
Dim TCFflow As object

Dim Output3 As object
Dim Volume2 As object
Dim Temperature As object
Dim Setpoint As object

Dim O2 As Integer        'Declare local variable
Dim U1 As Single         'and data type
Dim F1 As Single
Dim TCF As Single

Dim O3 As Integer
Dim U2 As Single
Dim Temp As Single
Dim SP As Single
```

Standard Application

By using the Visual Basic Macro language within Excel, it is a simple matter to bring data into a spreadsheet for display in a cell, or use graphs to display it. Here is the flowmeter batch volume in Litres, as used before, where the spreadsheet itself calculates the equivalent volume in Gallons.

P-NET

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Worksheet in PNET2.PPT". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The active cell is B2, containing the formula $=A2/4.54608$. The spreadsheet has two columns: "Volume Ltrs" (A) and "Volume Gals" (B). Row 2 shows a value of 123.456 in A2 and 27.15658325 in B2. A 3D bar chart is embedded in the spreadsheet, comparing the values in A2 and B2. The chart has two bars, labeled 1 and 2, with a legend for "Series1". The y-axis ranges from 0 to 200. The status bar at the bottom indicates "Ready".

International P-NET User Organization

Summary

I have shown in this short paper, the ability of Windows based applications to interact with devices connected to a fieldbus (P-NET in this example).

This has been possible due to a user configurable communications package called COPIFA.

The use of Visual Basic as a Windows application development language, allows a user to generate specific real time screens to suit his requirements. Further, the use of the Visual Basic macro language in some standard and widely available Windows applications such as EXCEL, means that the Fieldbus connection can be made even easier.

The fact that COPIFA utilises OLE technology modules in the form of a Virtual Global Object (VIGO) and a configurable real time communications kernel (HUGO), means that specialist graphics display applications which support OLE automation, can also make the Fieldbus connection.

Because COPIFA is also designed for inter-networking, it means that using the currently available Novell Network abilities, a whole network of PC's can display and control fieldbus data.

Finally, due to the open nature of COPIFA and the activation of co-operative projects such as RACKS, means that multi-fieldbus inter-communication can now take place, ensuring a truly global Visual Basic to Fieldbus Connection.