

## **CHAPTER THREE: *Control by LAN***

In this chapter, see how to

- ***Control WaveMaster by LAN***
- ***Simulate GPIB messages using LAN***

## Introduction

- The Ethernet connection (10Base-T and 100Base-T) allows you to control the instrument over a network, or through a direct connection between the oscilloscope and a computer. The connection is made through the Ethernet port located at the rear of the oscilloscope.
- This chapter introduces the basic capabilities for control of the instrument over the Ethernet interface.

This manual gives a complete description of the remote control commands. The commands apply to control of the oscilloscope via Ethernet and GPIB.

### Implementation Standard

To the greatest extent possible, these remote commands conform to the IEEE 488.2<sup>1</sup> standard, which may be considered as an extension of the IEEE 488.1 standard, dealing mainly with electrical and mechanical issues. When using LAN, the strings of data which are to be sent to the instrument must be preceded by the requisite header.

### Connections

The oscilloscope can be connected to the computer via Ethernet, using a TCP/IP network protocol.

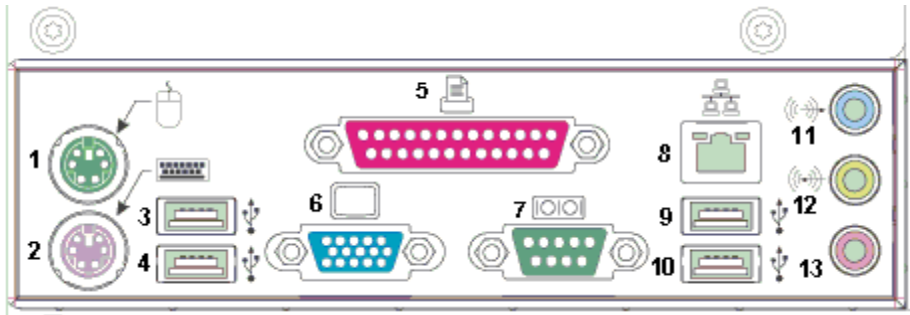
This connection can be made through a **network** (using a hub, switch, etc.) with a **straight through** network cable, or between the oscilloscope's Ethernet interface and a **computer** using a **crossover** network cable.

---

<sup>1</sup> ANSI/IEEE Std. 488.2-1987, *IEEE Standard Codes, Formats, Protocols, and Common Commands*. The Institute of Electrical and Electronics Engineers Inc., 345 East 47th Street, New York, NY 10017, USA.

### Connecting the Instrument to its Host

This section describes connecting the instrument to the host PC or network over the standard 10Base-T/100Base-T Ethernet. Windows NT and Windows 95 operating systems are supported.



#### Scope Rear Panel

The LAN connector is shown in the illustration above (item 8).

- Supports IEEE 802.3 Ethernet standards
- Supports 10BASE-T and 100BASE-T

#### Ethernet Connection

The instrument operates over a standard 10Base-T/100Base-T Ethernet connection. The instrument can be plugged into a network or operated from a direct connection to a host computer. A different type of cable is required for each of these connections. For a direct connection to the PC, a crossover cable is required, whereas the network connection is made using a straight cable.

**Headers for LAN Data Transfers**

The format of the header sent before each data block, both to and from the instrument, is set out in the following table:

Byte #	Purpose
0	Operation
1	Header Version
2	Spare (reserved for future expansion)
3	Spare (reserved for future expansion)
4	Block Length, (bytes of data), MSB
5	Block Length (bytes of data)
6	Block Length (bytes of data)
7	Block Length, (bytes of data), LSB

PART ONE: ABOUT REMOTE CONTROL

The 'Operation' bits and meanings are:

D7	D6	D5	D4	D3	D2	D1	D0
DATA	REMOTE	LOCKOUT	CLEAR	SRQ	Reserved	Reserved	EOI

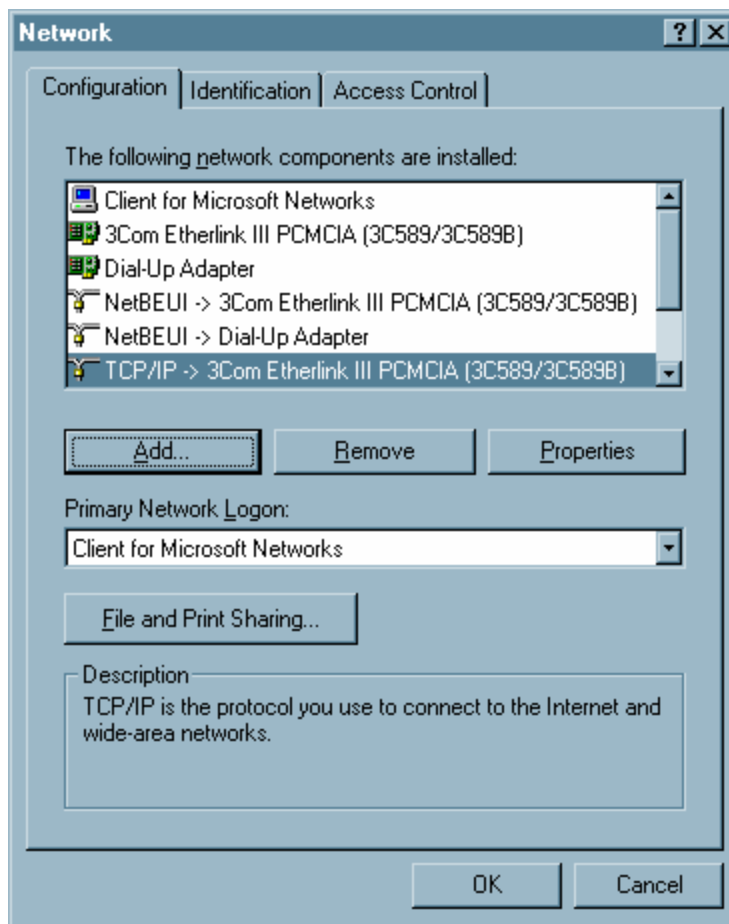
DATA BIT	MNEMONIC	PURPOSE
D7	DATA	Data block (D0 indicates termination with/without EOI)
D6	REMOTE	Remote Mode
D5	LOCKOUT	Local Lockout (Lockout front-panel)
D4	CLEAR	Device Clear (if sent with data, clear occurs before data block is passed to parser)
D3	SRQ	SRQ (Device to PC only)
D2..D1	Reserved	Reserved for future expansion
D0	EOI	Block terminated in EOI Logic "1" = use → EOI terminator Logic "0" = no EOI terminator

**Note:** The following examples assume that the host PC operates from Windows™ 95. The connection procedure for Windows NT is similar.

### Manual Setting of LAN Address


If you do need to set an address for the instrument, you need to go into Windows and perform the usual operations for setting an address. **Before establishing a direct connection between the oscilloscope and the host computer, the PC must first be properly configured. A specific TCP/IP address must be assigned — known as "static addressing." But this means that the PC *cannot* be set up to obtain its IP address from a DHCP server. To set the host PC's static address with Windows 95:**

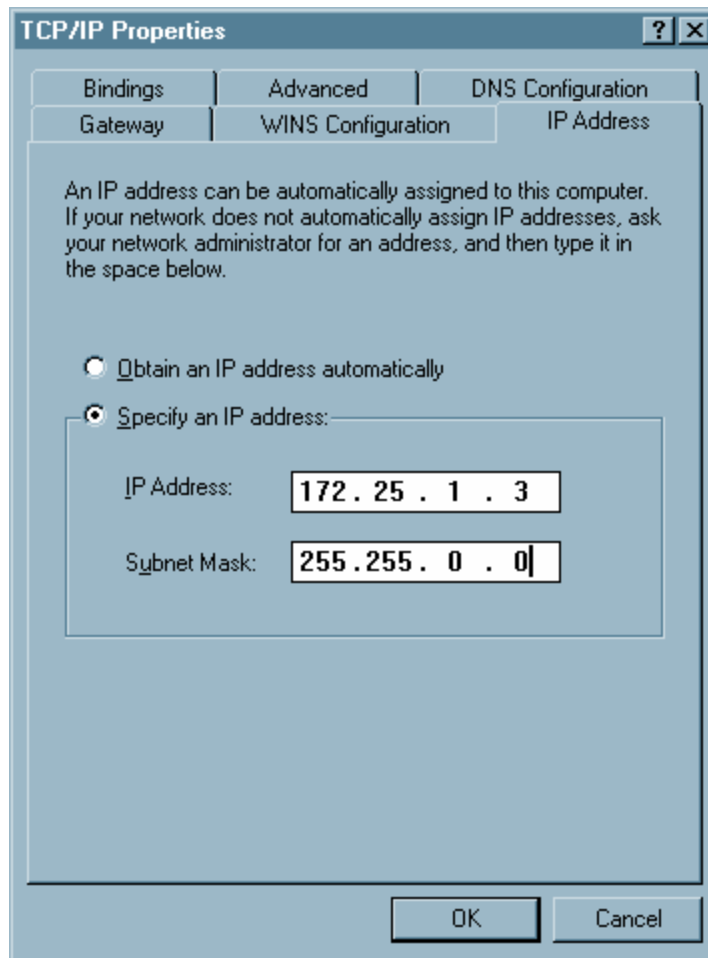
1. Select Start ? Settings ? Control Panel.
2. Double-click the Network icon in the Control Panel. A network dialog box similar to this one appears:



## PART ONE: ABOUT REMOTE CONTROL

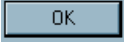
---

3. If the TCP/IP protocol is not listed, you will have to add it. Follow your operating system user guide to add the TCP/IP protocol and bind it to the Ethernet adapter.
4. Double-click the  TCP/IP -> line. A dialog box similar to the one below appears. Select ☒ Specify an IP address:



5. If this has already been selected, the computer's static address is set and nothing more needs to be done. Cancel out of the TCP/IP and network dialog boxes, and close the control panel.
6. If the address has not already been selected, fill in the IP address and subnet mask as shown above. The subnet mask for 172.25.x.x is 255.255.0.0. If the computer will not be plugged into a network, the above

address (or almost any address within the chosen subnet) will suffice. The only address that will not work is the same one as that of the oscilloscope to be controlled.

7. Now click  in the TCP/IP Properties dialog box. Depending on the operating system and version, you may need to reboot the computer. If so, a dialog box should alert you to this.

### **Making Physical Connection**

To make the physical connection between the oscilloscope and the host computer:

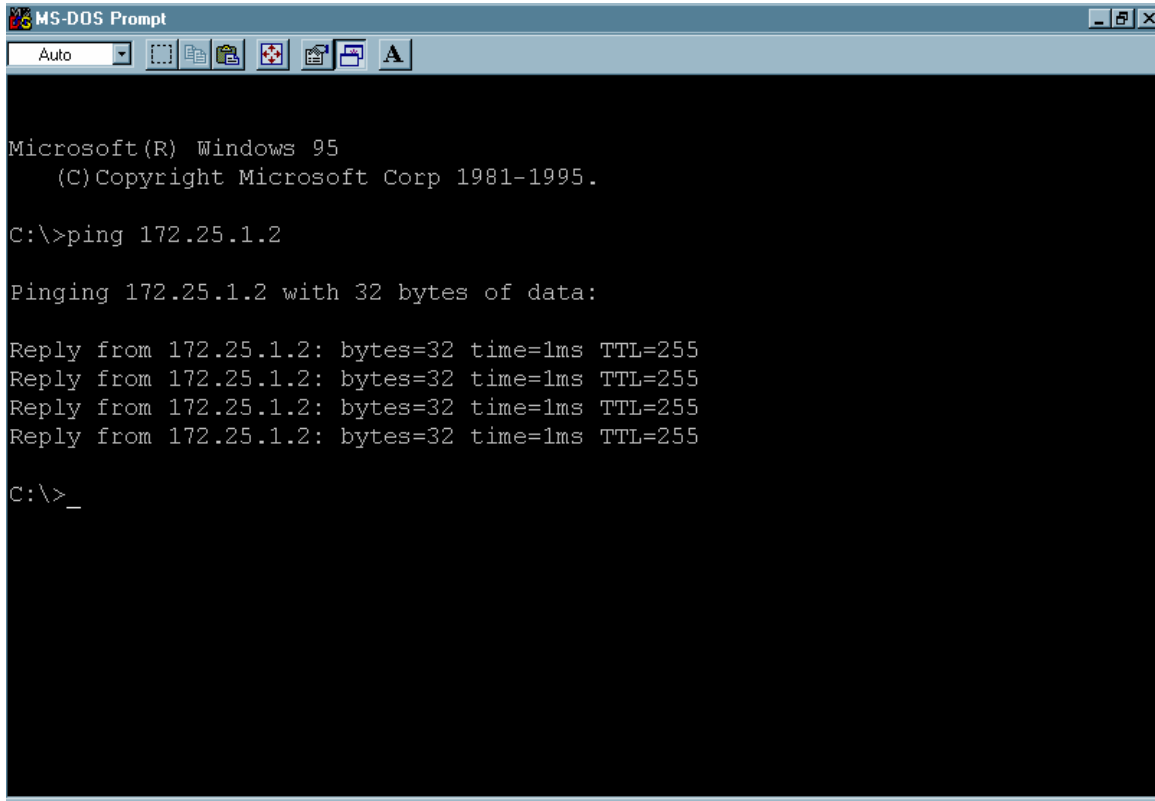
1. Connect the oscilloscope to the PC using a crossover cable (for direct connection).
2. Power the oscilloscope unit on.

***Note: If your PC does not have TCIP/IP, see your computer's User's Manual for installation instructions.***

### **Verifying Connection**

The physical connection and the PC's TCP/IP configuration can be verified using the “ping” command, available on both Windows™ 95 and Windows™ NT with TCP/IP network protocol installed. In order to check the network connection between the PC and the oscilloscope:

1. Start MS-DOS Prompt
2. Type ping <ip\_address>, where <ip\_address> is the static address assigned to the oscilloscope. The dialog box on the next page illustrates the result of a successful “ping,” with the Ethernet connection shown established. **The ping command has sent a message to the instrument and waited for a response. If a nm timeout occurs, as is shown in the box on the next page, the IP address used for the destination (the oscilloscope) is *incorrect* or not within the subnet mask of the PC's IP.**



```
MS-DOS Prompt
Auto
Microsoft(R) Windows 95
(C) Copyright Microsoft Corp 1981-1995.

C:\>ping 172.25.1.2

Pinging 172.25.1.2 with 32 bytes of data:

Reply from 172.25.1.2: bytes=32 time=1ms TTL=255
Reply from 172.25.1.2: bytes=32 time=1ms TTL=255
Reply from 172.25.1.2: bytes=32 time=1ms TTL=255
Reply from 172.25.1.2: bytes=32 time=1ms TTL=255

C:\>_
```

### Network Connection

Check with your network administrator before connecting the oscilloscope to a network. Incorrect addresses on a network can cause both the network and the oscilloscope to behave strangely. However, a network connection ought to be as simple as plugging the oscilloscope into the network. *Proper connection can be verified by following the verification instructions in the previous section.*

**If you are concerned mainly with system throughput, network connection is *not* recommended because the network traffic will slow down the oscilloscope's data transfer rate.**

***Note: The default Gateway is assigned as "172.25.0.1". Unless your network has this Gateway available, you must ensure the computer and the oscilloscope are on the same subnet.***

**Changing IP Address** Once the IP address is changed, the unit will no longer respond to the original address.

If the network settings are unknown or accidentally set to invalid values, they can be recovered by following the procedure above.

## Introduction to Software Tools

**The instrument software tools allow you to develop your own application specific programs quickly and easily. These tools are based on *ActiveDSO*<sup>™</sup>. The files for all the software described here are to be found on the CD-ROM and on LeCroy's Web site at <http://www.lecroy.com/tm/library/software/>.**

***ActiveDSO*** Based on Microsoft's ActiveX control technology, *ActiveDSO* gives leverage to widely available Microsoft software tools, and makes programming within the Microsoft environment easier. *ActiveDSO* simplifies the computer's interface with the instrument and programming within Visual C++, Visual Basic, or any other ActiveX compatible applications. For example, Microsoft Excel can even be used to control and retrieve data directly from the instrument. This tool becomes part of the target application and provides seamless access to the full power of the instrument.

# Using ActiveDSO

**ActiveDSO is highly suitable for fast program development in the Microsoft environment. This program is a control of ActiveX, the software technology developed by Microsoft as a subset of its COM model.**

ActiveDSO facilitates programming with the instrument by providing a ready interface between the instrument and the host computer. Programs such as Visual C++, Visual Basic, or Visual Basic for Applications (VBA) can be used under remote control without concern for interfacing complications. ActiveDSO acts as the key design structure allowing effective integration of software from the different manufacturers supporting ActiveX containment.

## CONTROL INSTANTIATION

This ActiveX component can be instantiated more than once by using the Visual Basic function CreateObject. Once the object is created, invoking the connection method will initialize it. ActiveDSO enables control of the instrument from a variety of PC desktop applications. The complexities of programming with Ethernet are fully encapsulated in this control. For example, with less than 10 lines of VBA code in an Excel macro the spreadsheet can recover pre-scaled waveform data from the WaveMaster (see the on-line Excel example in ActiveDSO).

ActiveDSO control can be used in two fundamental ways:

1. As a visible object embedded in an OLE automation compatible client (PowerPoint, for example) showing a captured instrument display image. See the Embedded Control example below for more details.
2. As an invisible object accessed through a scripting language (VBA, for example) to remotely control the instrument. See VBA example below for more details.

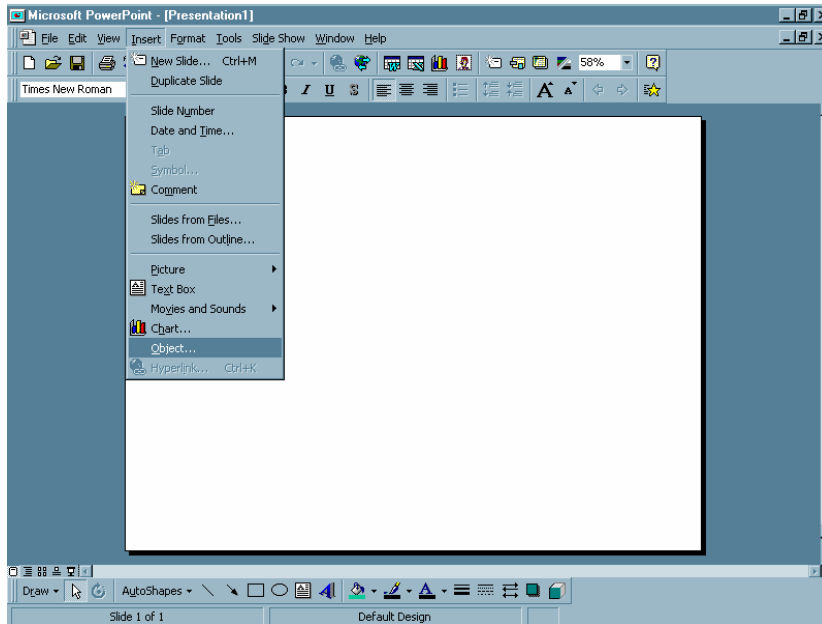
The ActiveDSO control may be embedded in any ActiveX containment-capable client, and may be used manually without need of any programming or scripting.

**Example: PowerPoint** This example shows the control being embedded in a Microsoft PowerPoint slide. The waveform captured by the instrument can be easily imported into PowerPoint with just a few mouse clicks:

1. Ensure that the ActiveDSO files from the CD-ROM are installed on the PC.
2. Verify that the PC and instrument are properly connected to the Ethernet.
3. Open a new blank presentation in PowerPoint.

**Note: This example assumes that PowerPoint 97 is being used. Earlier (or Later) versions may not behave in the same manner.**

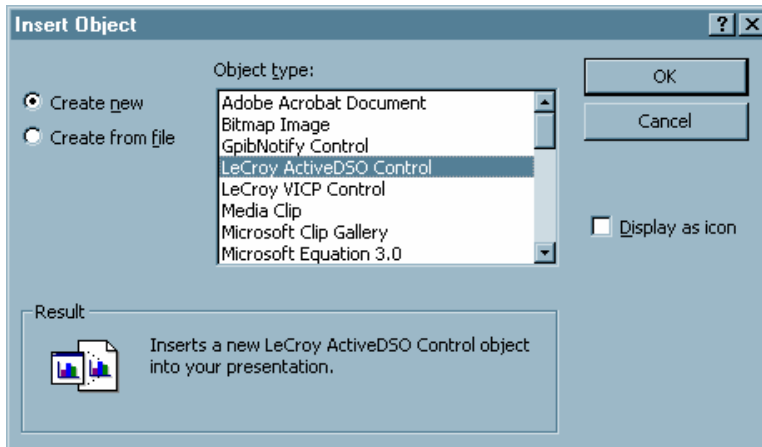
4. Select “Insert,” then **Object**, as shown here:



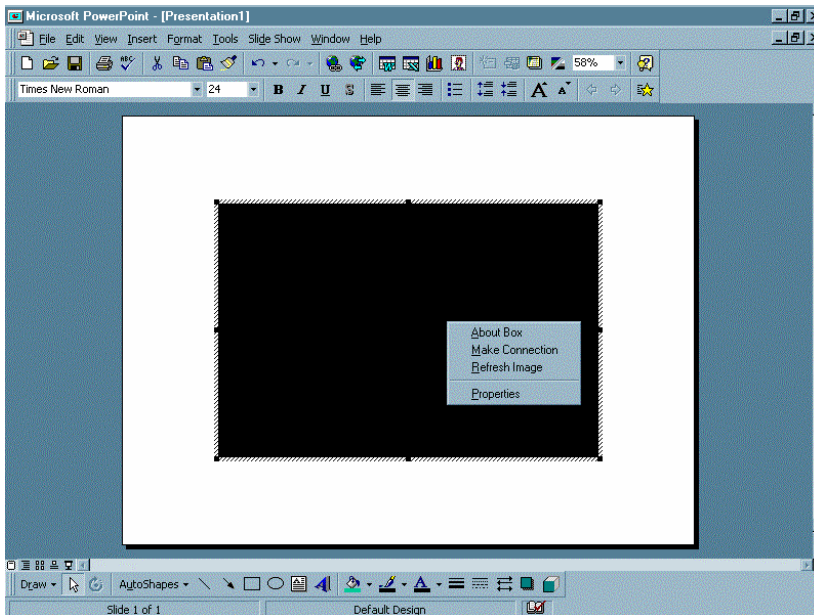
## PART ONE: ABOUT REMOTE CONTROL

---

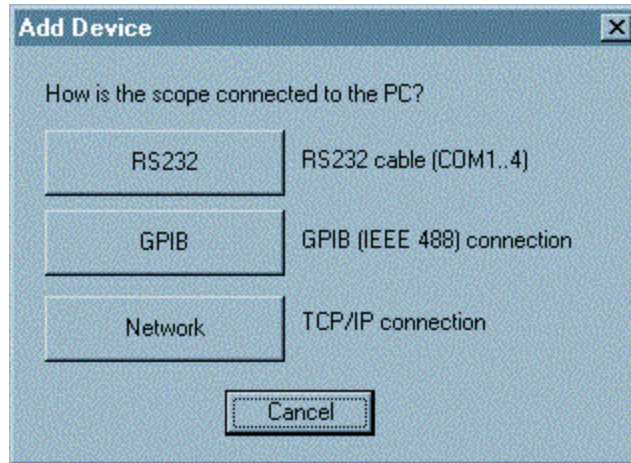
- From the pop-up window, select LeCroy ActiveDSO object as shown here:



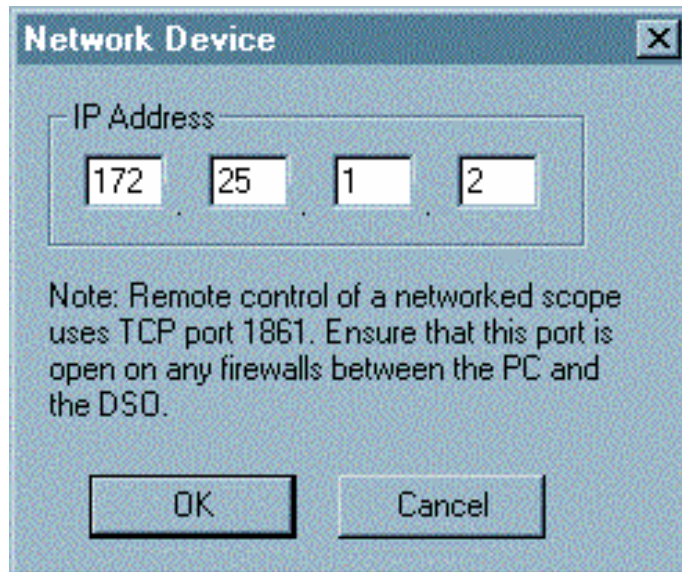
- Right-click the object and select "Make Connection."



7. Select "Network TCP/IP connection" as shown here ("scope" = WaveMaster):



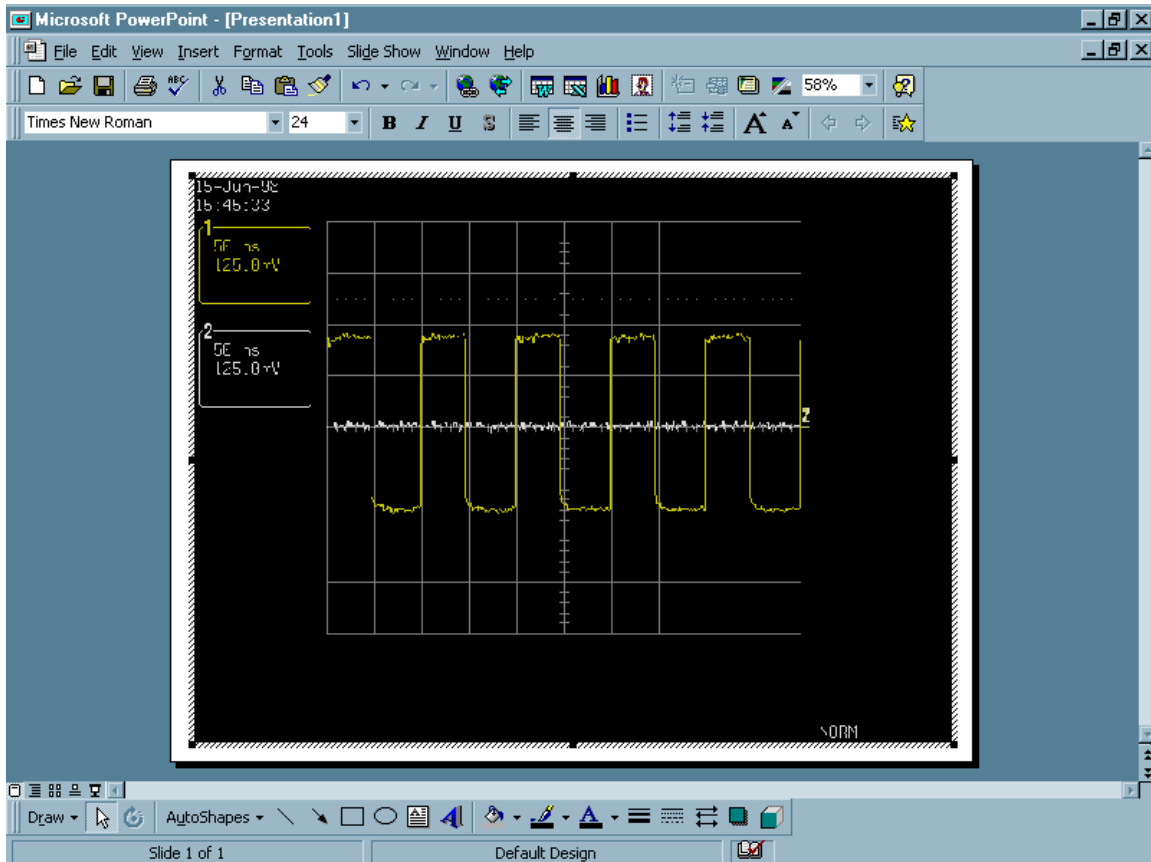
8. Enter the instrument's IP address and click "OK."



The address can also be specified in URL form, or 127.0.0.1 if you are running the controlling application on the instrument.

## PART ONE: ABOUT REMOTE CONTROL

9. Right-click the object again and select the Refresh Image menu item. A captured waveform will be displayed similar to the one shown here:



*Instrument's captured waveform imported into PowerPoint*

Once the ActiveDSO™ object has been properly set within the application, a macro script can be created, utilizing an object method such as WriteString() to send DISP ON, C1:TRA ON, TRMD. Then RefreshImage() method can be used to update the screen.

Example:

**VBA** VBA is the programming language built in to many of the more recent Windows applications. It is a subset of Visual Basic that makes using OLE Automation Servers and ActiveX Controls very simple. The following VBA subroutine demonstrates how easy it is to connect to an instrument and send remote commands to it.

```
Sub LeCroyDSOTest()  
  
Dim dso As Object  
  
    Set dso = CreateObject("LeCroy.ActiveDSO.1")  
  
    Call dso.AboutBox                ' Present the control's About box  
    Call dso.MakeConnection("IP:172.25.1.2") ' Connect to the unit  
    Call dso.WriteString("C1:VDIV 2", 1)    ' Setup C1 for 2 Volts/Div  
    Call dso.WriteString("TRMD AUTO", 1)    ' Set the trigger mode to AUTO  
  
End Sub
```

---

To enter the VBA editor in members of the Microsoft Office suite:

1. Select Tools ? Macro ? Visual Basic Editor menu item.
2. When the VBA window appears, select the Insert ? Module menu item.
3. Copy the above example into the editor window that appears.

**To execute:**

4. Position the text cursor within the subroutine.
5. Either select the Run ? Run Sub/UserForm or press function key F5.

***Note: For more information, see the ActiveDSO on-line Help. On-line Help contains VisualC++ examples, and explanations of ActiveDSO Methods and Properties.***

**ActiveDSO** This **ActiveX™** control enables LeCroy oscilloscopes to be **controlled by**, and to **exchange data with**, a variety of Windows applications that support the ActiveX standard. MS Office programs, Internet Explorer, Visual Basic, Visual C++, Visual Java, and MATLAB (V5.3 and later) are a few of the many applications that support ActiveX controls. ActiveDSO is available on CD-ROM or on the internet at [www.lecroy.com](http://www.lecroy.com)

- With **ActiveDSO** you can develop your test program using standard GPIB commands. For easy integration of your scope data with your Windows Application (through GPIB or Ethernet 10Base-T or 100Base-T), ActiveDSO helps you with the following tasks:
- Generate a report by importing scope data right into Excel or Word.
- Archive measurement results on the fly in a Microsoft Access Database.
- Automate tests using Visual Basic, Java, C++, Excel (VBA).

## PART ONE: ABOUT REMOTE CONTROL

---

- The ActiveDSO control hides the intricacies of programming and provides a simple and consistent interface to the controlling application. With less than 10 lines of VBA (Visual Basic for Applications) code in an Excel macro the spreadsheet can recover pre-scaled waveform data from a remote instrument.
- The ActiveDSO control can also be embedded visually in any OLE automation compatible client, and can be used manually without any need for programming. It will run on any PC running Windows 95, Windows 98, or Windows NT.
- There are two fundamental ways to use the control:
- As a visible object embedded in an OLE Automation compatible Client (PowerPoint for example) showing a captured display image. See Embedded Control Example for more details.
- As an invisible object accessed via a scripting language (Visual Basic for Applications, for example) to remotely control an instrument. See Accessing from VBA for more details.
- VBA (Visual Basic for Applications) is the programming language built into many of the more recent Windows applications. It is a subset of Visual Basic that makes it very simple to utilize the services of OLE Automation Servers and ActiveX Controls.
- The following VBA subroutine demonstrates how easy it is to connect to an instrument and send remote commands to it.

```
Sub LeCroyDSOTest()  
    Dim o As Object  
        Set o = CreateObject("LeCroy.ActiveDSOCtrl.1")  
    Call o.AboutBox           ' Present the control's About box  
    Call o.MakeConnection("IP: 172.28.11.26) 'Connect to device on LAN  
    Call o.WriteString("BUZZ BEEP", True) ' Make the DSO beep  
End Sub
```

### Example Syntax:

- Boolean controlName.WriteString
- The WriteString method has the following arguments.

### Argument Description

- controlname The name of the ActiveDSO control object.
- textString String, Text string to send to the device.
- EOI Boolean, TRUE = terminate with EOI
- **Returns:** True on success, False on failure.
- **Remarks:** This method sends a string command to the instrument.

- If EOI is set to TRUE, the device will start to interpret the command immediately. This is normally the desired behavior.
- If EOI is set to FALSE, a command may be sent in several parts with the device starting to interpret the command only when it receives the final part which should have EOI set TRUE.

§ § §