

# eSCL\* Communication Reference Guide



**\*SCL over Ethernet**



## Table of Contents

Introduction .....	3
Getting Started .....	3
Connecting the Drive to Your PC.....	3
Addresses, Subnets, and Ports .....	3
Option 1: Connect a Drive to Your Local Area Network.....	5
Using DHCP .....	7
Option 2: Connect a Drive Directly to Your PC.....	8
Option 3: Use Two Network Interface Cards (NICs) .....	10
Configure the Drive for Your Motor and Application.....	11
ST Configurator .....	12
Quick Tuner .....	12
Q Programmer.....	12
Learn the SCL and Q Languages .....	12
Create Your Own Application .....	13
Packet Format .....	13
Sending Commands to a Drive .....	13
Receiving Responses from a Drive .....	13
Further Reading.....	13
Conclusion .....	14
Appendix A: Changing Your Drive's Table of IP Addresses.....	14
Appendix B: Example UDP Programs .....	15
Visual Basic 6 .....	15
C#.NET .....	16
Appendix C: Example TCP Program.....	18
Visual Basic 6 .....	18

## Introduction

eSCL (SCL over Ethernet) is Applied Motion Products' language for commanding and querying motion control products over Ethernet. It is supported by several motion control devices, including the ST5-Q-Ex, ST10-Q-Ex and SV7-Q-Ex. In addition to sending commands to a drive from a host in real time, you can also use our *Q Programmer* software to embed sequences of commands, called Q Programs, in a drive. These programs can be set to execute automatically at power up, or can be triggered by commands sent from the host.

This guide is intended to help you connect and configure your drive and to help you start writing your own eSCL host application.

## Getting Started

There are four steps required to create an eSCL application with your new Applied Motion Products motor driver. Each of these is explained in a separate section of this manual.

1. Connect the drive to your PC. This includes getting the drive physically connected to your network (or directly to the PC), setting the drive's IP address, and setting the appropriate networking properties on your PC.
2. Configure the drive for your motor and application. For step motor drives, you'll need to use a suitable version of our *Configurator* software. For servos, use *Quick Tuner*.
3. Learn the SCL and Q languages. The *eSCL Utility* is a great way to explore the features and syntax of this powerful, flexible language, with help from the *Host Command Reference Manual*.
4. Create your own application. This guide includes code examples in Visual Basic and C# to help you get started. You can download the examples in their entirety from our website, but we recommend reading the explanations in this guide first.

## Connecting the Drive to Your PC

This process requires three steps

- Get the drive physically connected to your network (or directly to the PC)
- Set the drive's IP address
- Set the appropriate networking properties on your PC.

## Addresses, Subnets, and Ports

Every device on an Ethernet network must have a unique IP address. In order for two devices to communicate with each other, they must both be connected to the network and they must have IP addresses that are on the same subnet. A subnet is a logical division of a larger network. Members of one subnet are generally not able to communicate with members of another unless they are connected through special network equipment (e.g. router). Subnets are defined by the choices of IP addresses and subnet masks.

If you want to know the IP address and subnet mask of your PC, select Start...All

Programs...Accessories...Command Prompt. Then type "ipconfig" and press Enter. You should see something like this:

```
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : 
    IP Address. . . . . : 192.168.0.22
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.254
```

If your PC's subnet mask is set to 255.255.255.0, a common setting known as a Class C subnet mask, then your machine can only talk to another network device whose IP address matches yours in the first three octets. (The numbers between the dots in an IP address are called an octet.) For example, if your PC is on a Class C subnet and has an IP address of 192.168.0.20, it can talk to a device at 192.168.0.40, but not one at 192.168.1.40. If you change your subnet mask to 255.255.0.0 (Class B) you can talk to any device whose first two octets match yours. Be sure to ask your system administrator before doing this. Your network may be segmented for a reason.

Your drive includes a 16 position rotary switch for setting its IP address. The factory default address for each switch setting is shown in the table below.

Rotary Switch	IP Address
0	10.10.10.10
1	192.168.1.10
2	192.168.1.20
3	192.168.1.30
4	192.168.0.40
5	192.168.0.50
6	192.168.0.60
7	192.168.0.70
8	192.168.0.80
9	192.168.0.90
A	192.168.0.100
B	192.168.0.110
C	192.168.0.120
D	192.168.0.130
E	192.168.0.140
F	DHCP

Settings 1 through E can be changed using the *ST Configurator* software (use *Quick Tuner* for servo drives). Setting 0 is always "10.10.10.10", the universal recovery address. If someone were to change the other settings and not write it down or tell anyone (I'm not naming names here, but you know who I'm talking about) then you will not be able to communicate with your drive. The only way to "recover" it is to use the universal recovery address.

Setting F is “DHCP”, which commands the drive to get an IP address from a DHCP server on the network. The IP address automatically assigned by the DHCP server may be “dynamic” or “static” depending on how the administrator has configured DHCP. The DHCP setting is reserved for advanced users.

Your PC, or any other device that you use to communicate with the drive, will also have a unique address. When you launch an Applied Motion Ethernet-ready software application, it will display the IP address of your PC on the title bar.

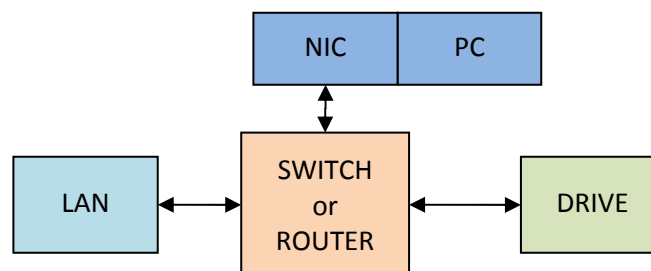
On the drive, switch settings 1 through E use the standard class B subnet mask (i.e. “255.255.0.0”). The mask for the universal recovery address is the standard class A (i.e. “255.0.0.0”).

One of the great features of Ethernet is the ability for many applications to share the network at the same time. Ports are used to direct traffic to the right application once it gets to the right IP address. The UDP eSCL port in our drives is 7775. To send and receive commands using TCP, use port number 7776. You’ll need to know this when you begin to write your own application. You will also need to choose an open (unused) port number for your application. Our drive doesn’t care what that is; when the first command is sent to the drive, the drive will make note of the IP address and port number from which it originated and direct any responses there. The drive will also refuse any traffic from other IP addresses that is headed for the eSCL port. The first application to talk to a drive “owns” the drive. This lock is only reset when the drive powers down.

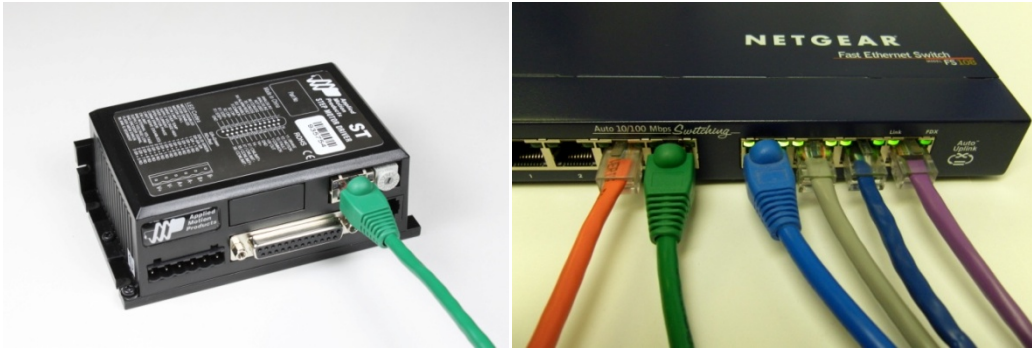
If you need help choosing a port number for your application, you can find a list of commonly used port numbers at <http://www.iana.org/assignments/port-numbers>.

One final note: Ethernet communication can use one or both of two “transport protocols”: UDP and TCP. eSCL commands can be sent and received using either protocol. UDP is simpler and more efficient than TCP, but TCP is more reliable on large or very busy networks where UDP packets might occasionally be dropped.

### Option 1: Connect a Drive to Your Local Area Network



If you have a spare port on a switch or router and if you are able to set your drive to an IP address that is compatible with your network, and not used by anything else, this is a simple way to get connected. This technique also allows you to connect multiple drives to your PC. If you are on a corporate network, please check with your system administrator before connecting anything new to the network. He or she should be able to assign you a suitable address and help you get going.



### Drive Connected to a Router or Switch

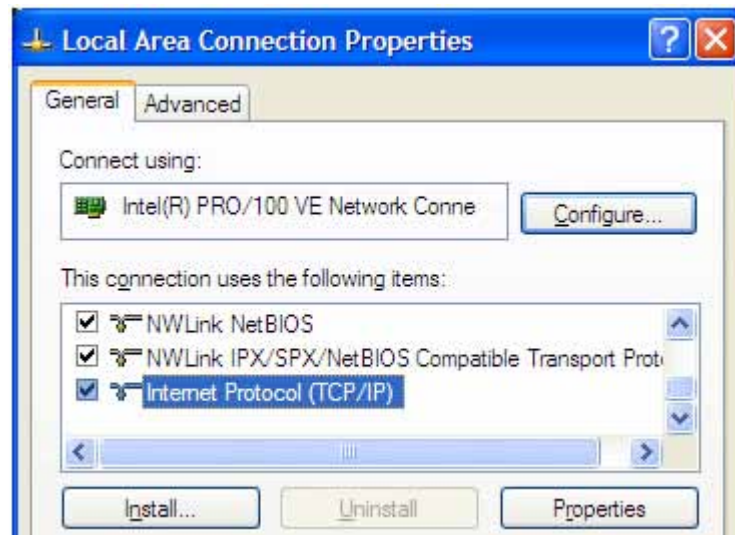
If you are not sure which addresses are already used on your network, you can find out using “Angry IP scanner”, which can be downloaded free from <http://www.angryip.org/w/Download>. But be careful: an address might appear to be unused because a computer or other device is currently turned off. And many networks use dynamic addressing where a DHCP server assigns addresses “on demand”. The address you choose for your drive might get assigned to something else by the DHCP server at another time.

Once you’ve chosen an appropriate IP address for your drive, set the rotary switch according the address table above. **If none of the default addresses are acceptable for your network, you can enter a new table of IP addresses using *ST Configurator* (for steppers) or *Quick Tuner* (for servos). See the appendix for details.**

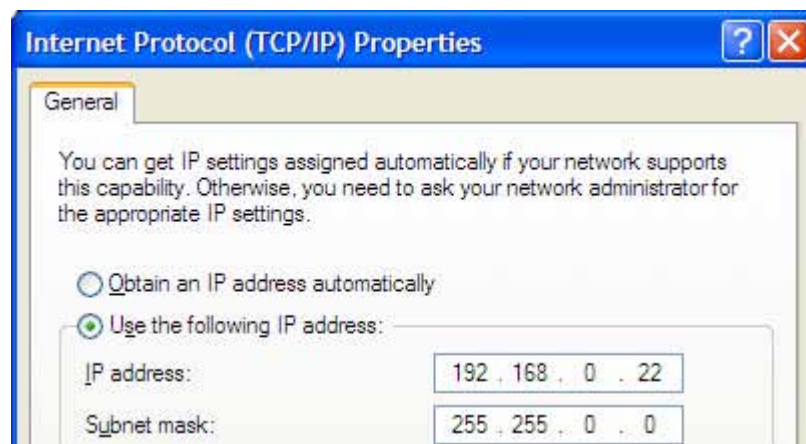
If your network uses addresses starting with 192.168.0, the most common subnet, you will want to choose an address from switch settings 4 through E. Another common subnet is 192.168.1. If your network uses addresses in this range, the compatible default selections are 1, 2 and 3.

If your PC address is not in one of the above private subnets, you will have to change your subnet mask to 255.255.0.0 in order to talk to your drive. To change your subnet mask:

1. On Windows XP, right click on “My Network Places” and select properties. On Windows 7, click Computer. Scroll down the left pane until you see “Network”. Right click and select properties. Select “Change adapter settings”
2. You should see an icon for your network interface card (NIC). Right click and select properties.
3. Scroll down until you see “Internet Properties (TCP/IP)”. Select this item and click the Properties button. On Windows 7 and Vista, look for “(TCP/IPv4)”



4. If the option “Obtain an IP address automatically” is selected, your PC is getting an IP address and a subnet mask from the DHCP server. Please cancel this dialog and proceed to the next section of this manual: “Using DHCP”.
5. If the option “Use the following IP address” is selected, life is good. Change the subnet mask to “255.255.0.0” and click OK.

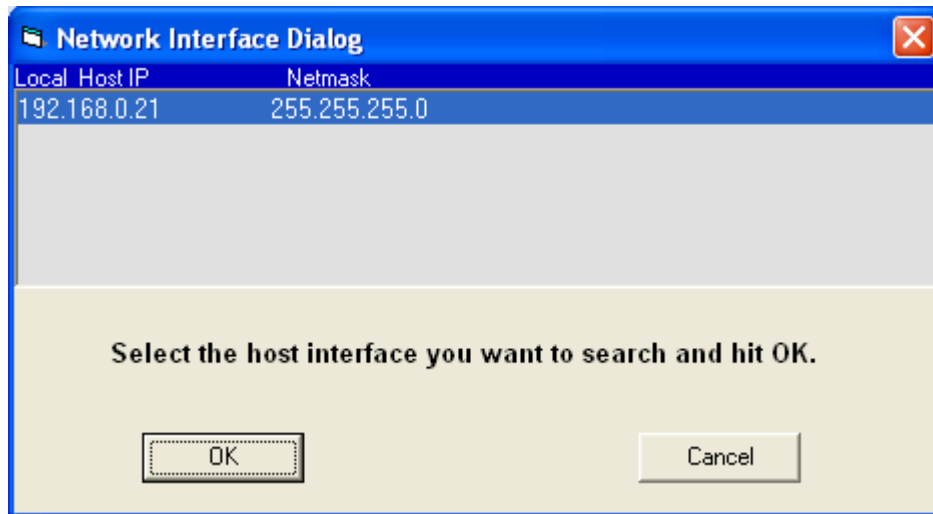


### Using DHCP

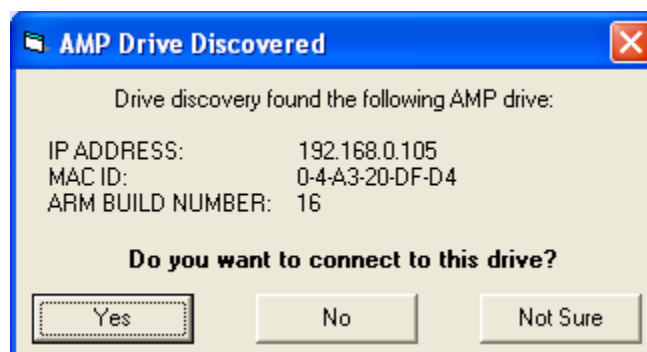
If you want to use your drive on a network that where all or most of the devices use dynamic IP addresses supplied by a DHCP server, set the rotary switch to “F”. When the drive is connected to the network and powered on, it will obtain an IP address and a subnet mask from the server that is compatible with your PC.

The only catch is that you won’t know what address the server assigns to your drive. *Ethernet Configurator* can find your drive using the Drive Discovery feature, as long as your network isn’t too large. With the drive connected to the network and powered on, select Drive Discovery from the Drive menu.

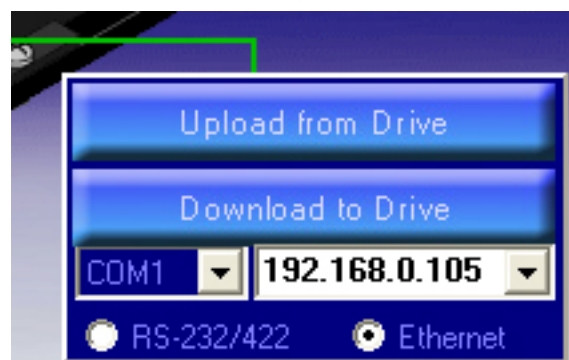
You will see a dialog such as this:



Normally, Drive Discovery will only detect one network interface card (NIC), and will select it automatically. If you are using a laptop and have both wireless and wired network connections, a second NIC may appear. Please select the NIC that you use to connect to the network to which you've connected your drive. Then click OK. Drive Discovery will notify you as soon as it has detected a drive.



If you think this is the correct drive, click Yes. If you're not sure, click Not Sure and Drive Discovery will look for additional drives on your network. Once you've told Drive Discovery which drive is yours, it will automatically enter that drive's IP address in the IP address text box so that you are ready to communicate.



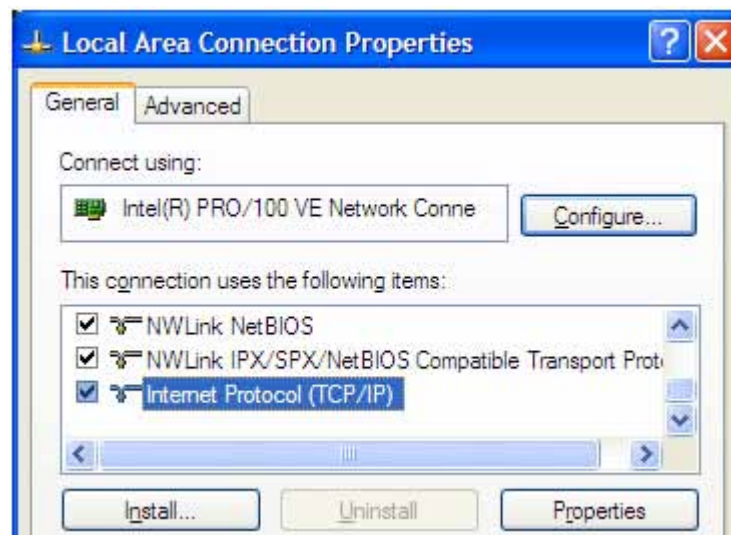
## Option 2: Connect a Drive Directly to Your PC





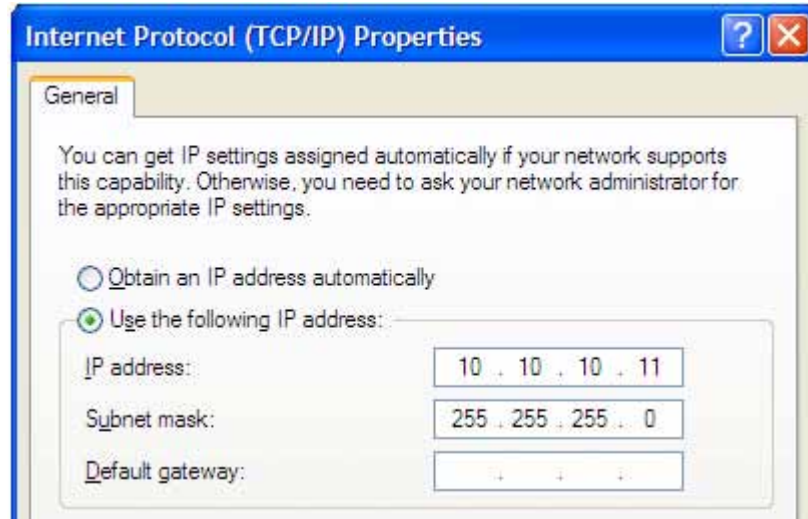
It doesn't get much simpler than this:

1. Connect one end of a CAT5 Ethernet cable into the LAN card (NIC) on your PC and the other into the drive. You don't need a special "crossover cable"; the drive will automatically detect the direct connection and make the necessary physical layer changes.
2. Set the IP address on the drive to "10.10.10.10" by setting the rotary switch at "0".
3. To set the IP address of your PC:
  - a. On Windows XP, right click on "My Network Places" and select properties.
  - b. On Windows 7, click Computer. Scroll down the left pane until you see "Network". Right click and select properties. Select "Change adapter settings"
4. You should see an icon for your network interface card (NIC). Right click and select properties.
  - a. Scroll down until you see "Internet Properties (TCP/IP)". Select this item and click the Properties button.
  - b. On Windows 7 and Vista, look for "(TCP/IPv4)"



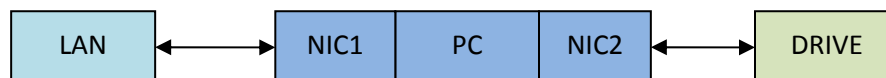
5. Select the option "Use the following IP address". Then enter the address "10.10.10.11". This will give your PC an IP address that is on the same subnet as the drive. Windows will know to direct any traffic intended for the drive's IP address to this interface card.

6. Next, enter the subnet mask as “255.255.255.0”.
7. Be sure to leave “Default gateway” blank. This will prevent your PC from looking for a router on this subnet.



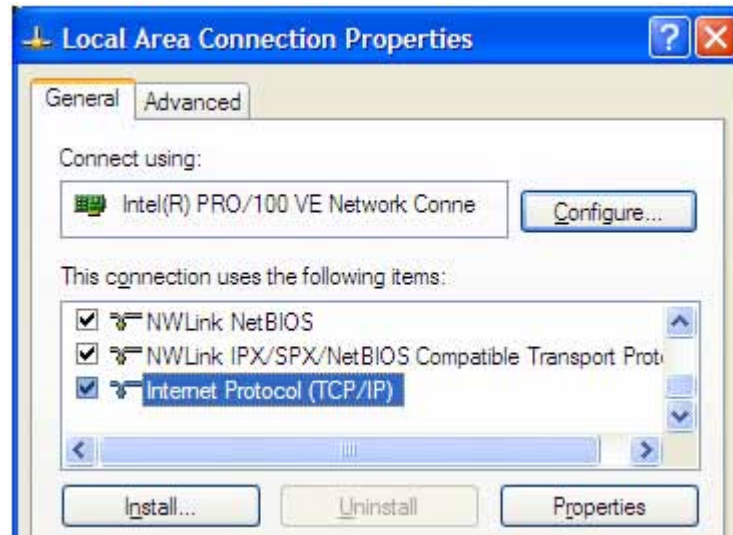
8. Because you are connected directly to the drive, anytime the drive is not powered on your PC will annoy you with a small message bubble in the corner of your screen saying “The network cable is unplugged.”

### Option 3: Use Two Network Interface Cards (NICs)

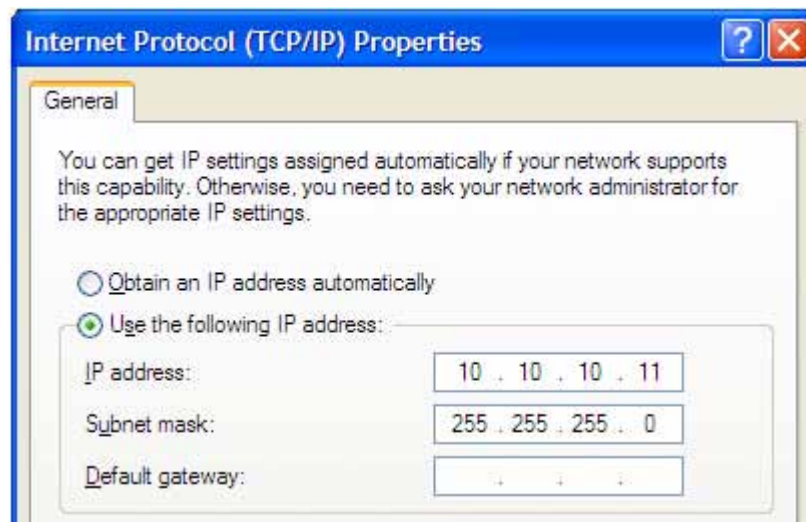


This technique allows you to keep your PC connected to your LAN, but keeps the drive off the LAN, preventing possible IP conflicts or excessive traffic.

1. If you use a desktop PC and have a spare card slot, install a second NIC and connect it directly to the drive using a CAT5 cable. You don't need a special “crossover cable”; the drive will automatically detect the direct connection and make the necessary physical layer changes.
2. If you use a laptop and only connect to your LAN using wireless networking, you can use the built-in RJ45 Ethernet connection as your second NIC.
3. Set the IP address on the drive to “10.10.10.10” by setting the rotary switch at “0”.
4. To set the IP address of the second NIC:
  - a. On Windows XP, right click on “My Network Places” and select properties.
  - b. On Windows 7, click Computer. Scroll down the left pane until you see “Network”. Right click and select properties. Select “Change adapter settings”
5. You should see an icon for your newly instated NIC. Right click again and select properties.
  - a. Scroll down until you see “Internet Properties (TCP/IP)”. Select this item and click the Properties button.
  - b. On Windows 7 and Vista, look for “(TCP/IPv4)”



6. Select the option “Use the following IP address”. Then enter the address “10.10.10.11”. This will give your PC an IP address that is on the same subnet as the drive. Windows will know to direct any traffic intended for the drive’s IP address to this interface card.
7. Next, enter the subnet mask as “255.255.255.0”. Be sure to leave “Default gateway” blank. This will prevent your PC from looking for a router on this subnet.



8. Because you are connected directly to the drive, anytime the drive is not powered on your PC will annoy you with a small message bubble in the corner of your screen saying “The network cable is unplugged.”

## Configure the Drive for Your Motor and Application

Three Windows programs are available from Applied Motion Products for use with our Ethernet drives. These programs are included on the CD that accompanies each drive and the most recent version is always available at [www.applied-motion.com](http://www.applied-motion.com).

**ST Configurator** is used to configure your stepper drive and motor. It can also be used to change the selection of drive IP addresses. *ST Configurator* includes extensive built-in help screens and manuals.

**Quick Tuner** is used to configure and tune servo drives. The *Quick Tuner Manual* is automatically installed in the Applied Motion Products program menu when you install *Quick Tuner*.

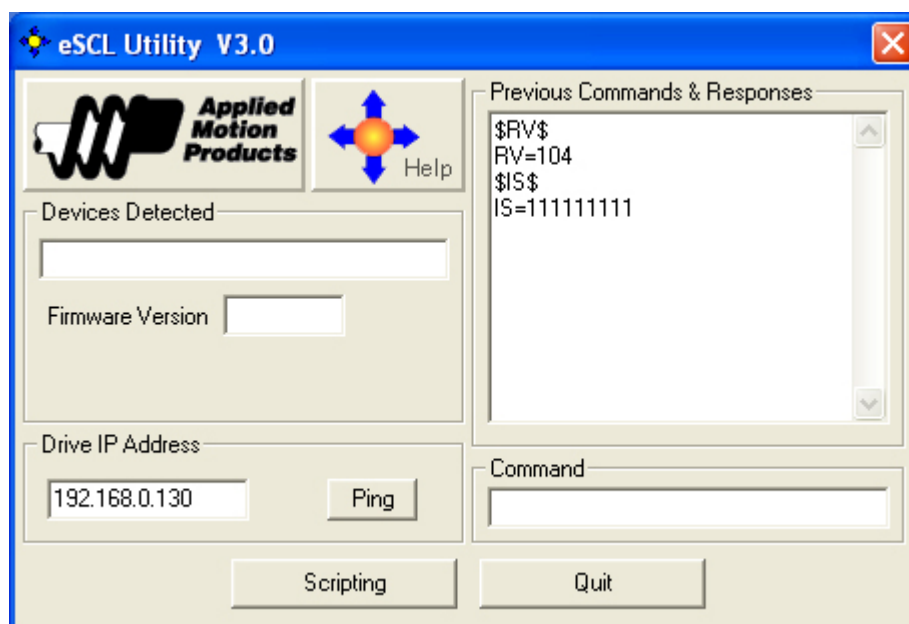
**Q Programmer** will be needed if you want to embed programs in the non-volatile memory of your drive, either to run automatically at power up or to be triggered by commands sent from a host.

## Learn the SCL and Q Languages

The best guide for learning the SCL and Q languages is the *Host Command Reference*, which is installed with *ST Configurator* and the *eSCL Utility*. The latest version is also available online at [www.applied-motion.com](http://www.applied-motion.com).

*eSCL Utility* is an excellent companion to the *Host Command Reference Manual*. It allows you to type in commands and observe the behavior of the drive. A scripting window can be used to execute sequences of commands, either one time or in a continuous loop.

To connect the *eSCL Utility* to your drive, enter the drive's IP address in the box provided. To test communication, click the Ping button.



To try an SCL command, enter the command in the Command box. Do not type 07, the opcode; the eSCL Utility will add the opcode and terminator automatically when you press Enter. The opcode is necessary to indicate to the drive that the packet is an SCL command. Other packets, such as Configurator downloads, use other opcodes.

## Create Your Own Application

To create your own application, you will need to choose a programming language, learn how SCL commands and responses are encapsulated in UDP or TCP packets, and learn to use your programming language's interface to the network. See Appendix B for example programs in Visual Basic 6 and C#.NET.

### Packet Format

eSCL is based on Applied Motion's Serial Command Language (SCL), an ASCII-based language with roots in RS-232 and RS-485 communication. eSCL drives support the full SCL and Q command sets, and utilize the speed and reliability of Ethernet. Commands and responses are encapsulated in the payload of User Datagram Protocol (UDP) packets, and are transmitted using standard Ethernet hardware and standard TCP/IP stacks.

For details on the features and capabilities of SCL and Q, please refer to the *Host Command Reference*.

### Sending Commands to a Drive

An eSCL UDP packet consists of three parts, the header (binary 07), the SCL string (a sequence of ASCII encoded characters) and the SCL terminator (ASCII carriage return, 13)

header	SCL string	<cr>
--------	------------	------

Example: Sending "RV"

- SCL Header = 07 (two bytes)
- R = ASCII 82
- V = ASCII 86
- <cr> (ASCII carriage return) = 13

header	"RV"		<cr>	
0	7	82	86	13

### Receiving Responses from a Drive

A typical response to "RV" would be "RV=103<cr>" which would be formatted as

header	"RV=103"							<cr>
0	7	82	86	61	49	48	51	13

## Further Reading

The following materials can be downloaded from [www.applied-motion.com](http://www.applied-motion.com).

- The *eSCL Utility* will help you get familiar with the SCL language.
- The *Host Command Reference* contains detailed information about the SCL and Q languages.

- *ST Configuration* is needed to configure the ST5-Q-Ex and ST10-Q-Ex step motor drives. This application also includes extensive help screens.
- *QuickTuner* is used to configure and tune SV7 servo drives. *Quick Tuner* also includes extensive help screens.
- Visual Basic 6 and C#.NET example projects can be downloaded from the software page.

To learn more about networking using Ethernet, we recommend reading *Sams Teach Yourself TCP/IP in 24 Hours*, available from Amazon.com and other fine booksellers.

## Conclusion

We hope your experience with Applied Motion Products is pleasant and productive. If you get stuck and need help, or wish to suggest ways to improve our products or documentation, please contact our applications engineering department by calling (800) 525-1609 or by writing to [support@applied-motion.com](mailto:support@applied-motion.com).

## Appendix A: Changing Your Drive's Table of IP Addresses

If you need to change the IP addresses that are selected by the rotary switch in your drive, you can do so using *Configurator* (for stepper drives) or *Quick Tuner* (for servo drives). Please follow the steps below.

**Step 1: If your network includes a DHCP server**, set your drive to rotary switch position "F" (DHCP). Connect it to a spare port on a switch or router and power up the drive. When the drive powers on, your network's DHCP server will assign a suitable address to your drive. This process may take a few minutes.

**If your network does not have a DHCP server** then you must connect the drive directly to your PC in order to change the table. Please read the Section "Connecting a Drive Directly to a PC", then return to Step 3 below.

**Step 2:** In *Configurator*, or *Quick Tuner*, select "Drive Discovery". A list of network adapter cards will appear. If you have more than one network adapter, select the one that connects to the same network as the drive. If you only have one network adapter, it will be automatically selected. Select OK. If Drive Discovery finds an Applied Motion Ethernet drive on your network, a message box will appear asking you if this is the correct drive. If it is, select Yes. If you are not sure (because there are multiple drives on your network), select Not Sure and the search will continue. You will be able to choose from all discovered drives once they are found.

*Note: if your subnet is too large (larger than Class C) drive discovery is not available because the discovery process would take too long.*

Once you have discovered a drive and acknowledged that it is the correct one, the IP address of this drive will automatically be set in the main window, and you will be able to communicate with the drive from the *Configurator* or *Quick Tuner* software.

**Step 3:** To change the table of IP addresses that will be stored in your drive, select "IP Addresses" from the Drive menu. Selecting "Read From Drive" will allow you to see which IP addresses correspond with the rotary



switch settings of your drive. You can edit the addresses and click “Save to Drive” to install a new set of addresses in your drive. You can also save a copy of the IP address table to a file by selecting “Save to File”.

After saving a new set of IP addresses to your drive, the drive will continue to communicate using its existing IP address until you cycle power.

## Appendix B: Example UDP Programs

All example programs are available for download at [www.applied-motion.com/ampinfo/software/samplecode/](http://www.applied-motion.com/ampinfo/software/samplecode/). You should still read this section so that you understand the key elements of the code and what tradeoffs you may encounter.

### Visual Basic 6

Even though VB6 is an older language, its refreshing simplicity makes it a compelling choice for quickly developing an Ethernet application.

To communicate over Ethernet UDP from VB6, you’ll need the Winsock control (MSWINSCK.OCX), which is included in the Professional and Enterprise editions of the language. To configure an instance of Winsock, you must specify the protocol as UDP, choose a local port number, and set the remote IP address and port number to match the drive. In the code example below, 7775 is the port of the drive. *driveIPAddress* is the IP address of the drive (“10.10.10.10” or “192.168.0.130” for example). 7777 is the port of the PC.

```
Winsock1.RemotePort = 7775
Winsock1.RemoteHost = driveIPAddress
Winsock1.Protocol = sckUDPProtocol
Winsock1.Bind 7777
// if port 7777 is in use by another application, you will get an error.
// that error should be trapped using the On Error statement
// and an alternate port should be chosen.
```

Sending “RV” command:

```
Dim myPacket(0 to 4) as Byte ' declare a byte array just large enough
myPacket(0) = 0 ' first byte of SCL opcode
myPacket(1) = 7 ' second byte of SCL opcode
myPacket(2) = "R" ' R
myPacket(3) = "V" ' V
myPacket(4) = vbCR ' carriage return
Winsock1.SendData myPacket
```

To receive a response, you will need to place some code in the Winsock\_DataArrival event. This event is automatically declared as soon as you add a Winsock control to your form. The DataArrival event will automatically trigger each time a packet is received. The code below extracts the SCL response from the UDP payload and displays it in a message box.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim udpData() As Byte, n As Integer
    Dim hexbyte As String, packetID As Long, SCLrx As String
    Winsock1.GetData udpData
    ' remotehost gets clobbered when packet rec'd,
```

```

' next line fixes it
Winsock1.RemoteHost = Winsock1.RemoteHostIP
' first 16 bits of packet are the ID (opcode)
If UBound(udpData) >= 1 Then
    packetID = 256 * udpData(0) + udpData(1)
    If packetID = 7 then ' SCL response
        SCLrx = ""
        For n = 2 To UBound(udpData)
            SCLrx = SCLrx & Chr(udpData(n))
        Next n
        MsgBox SCLrx
    End If
End If
End Sub

```

## C#.NET

The .NET languages are Microsoft's modern, object oriented Windows application building tools and include robust Ethernet support. We present this example in C#.

Make sure your project includes this line, providing access to an Ethernet socket:

```
using System.Net.Sockets;
```

In your form header you must declare a `UdpClient` object and create an instance, which can be done in the same line. The local port number is included in the "new `UdpClient`" call. This is the port number that will be reserved on the PC for your application.

```
static UdpClient udpClient = new UdpClient(7777);
```

To open the connection, invoke the `Connect` method, specifying the drive's IP address and port number:

```
udpClient.Connect("192.168.0.130", 7775);
```

To send "RV" to the drive:

```

//create a string loaded with the SCL command
Byte[] SCLstring = Encoding.ASCII.GetBytes("RV");
// create a byte array that will be used for the actual
// transmission
Byte[] sendBytes = new Byte[SCLstring.Length + 3];
// insert opcode (07 is used for all SCL commands)
sendBytes[0] = 0;
sendBytes[1] = 7;
// copy string to the byte array
System.Array.Copy(SCLstring, 0, sendBytes, 2, SCLstring.Length);
// insert terminator
sendBytes[sendBytes.Length - 1] = 13; // CR
// send it to the drive
udpClient.Send(sendBytes, sendBytes.Length);

```

Getting responses back from the drive in C# is a more complicated than VB6. You have two choices: poll for a response or create a callback function that will provide a true receive event.



Polling is easier to code but less efficient because you must either sit in a loop waiting for an expected response or run a timer to periodically check for data coming in. Since the choice depends on your programming style and the requirements of your application, we present both techniques.

### Polling for an incoming packet

The same `UdpClient` object that you use to send packets can be used to retrieve incoming responses from the drive. The `Available` property will be greater than zero if a packet has been received. To retrieve a packet, assign the `Receive` property to a `Byte` array. You must create an `IPEndPoint` object in order to use the `Receive` property.

```
private void UDPpoll()
{
    // you can call this from a timer event or a loop
    if (udpClient.Available > 0) // is there a packet ready?
    {
        IPEndPoint RemoteIpEndPoint = new IPEndPoint(IPAddress.Any, 0);
        try
        {
            // Get the received packet. Receive method blocks
            // until a message returns on this socket from a remote host,
            // so always check .Available to see if a packet is ready.
            Byte[] receiveBytes = udpClient.Receive(ref RemoteIpEndPoint);
            // strip opcode
            Byte[] SCLstring = new byte[receiveBytes.Length - 2];
            for (int i = 0; i < SCLstring.Length; i++)
                SCLstring[i] = receiveBytes[i + 2];
            string returnData = Encoding.ASCII.GetString(SCLstring);
            AddToHistory(returnData);
        }
        catch (Exception ex)
        {
            // put your error handler here
            Console.WriteLine(ex.ToString());
        }
    }
}
```

### Creating a receive event using a call back function

First, create a function to handle incoming packets. This function must contain two local objects: a `UdpClient` and an `IPEndPoint`. The call back function will be passed an `IAsyncResult` object that contains a reference to the UDP connection. The local `IPEndPoint` object is passed to the `UdpClient`'s `EndReceive` property to retrieve the packet.

```
public void ReceiveCallback(IAsyncResult ar)
{
    int opcode;
    UdpClient u = (UdpClient)((UdpState)(ar.AsyncState)).u;
    IPEndPoint e = (IPEndPoint)((UdpState)(ar.AsyncState)).e;
    Byte[] receiveBytes = u.EndReceive(ar, ref e);
    // get opcode
    opcode = 256 * receiveBytes[0] + receiveBytes[1];
    if (opcode == 7) // SCL response
    {
        string receiveString = Encoding.ASCII.GetString(receiveBytes);
    }
}
```

```

Byte[] SCLstring = new Byte[receiveBytes.Length - 2];
// remove opcode
System.Array.Copy(receiveBytes, 2, SCLstring, 0, SCLstring.Length);
receiveString = Encoding.ASCII.GetString(SCLstring);
AddToHistory(receiveString);
}
else if (opcode == 99) // ping response
{
    MessageBox.Show("Ping!", "eSCL Utility", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
}

```

The call back function will not be called unless it is “registered” with the `UdpClient` object using the `BeginReceive` method, as shown below. `StartRecvCallback` can be called from the Form Load event. It must also be re-registered each time it is called (this is to prevent recursion), which is most easily accomplished by making a call to `StartRecvCallback` each time you send a packet.

```

private void StartRecvCallback()
{
    UdpState s = new UdpState();
    s.e = new IPEndPoint(IPAddress.Any, 0);
    s.u = udpClient;
    udpClient.BeginReceive(new AsyncCallback(ReceiveCallback), s);
}

```

This example requires that you declare a class called `UdpState` as described below.

```

class UdpState
{
    public UdpClient u;
    public IPEndPoint e;
}

```

As if this event driven technique wasn’t quirky enough, it also creates a threading error unless the following statement is included in the form load event:

```

CheckForIllegalCrossThreadCalls = false; // this must be so for callbacks which
operate in a different thread

```

## Appendix C: Example TCP Program

All example programs are available for download at [www.applied-motion.com/ampinfo/software/samplecode/](http://www.applied-motion.com/ampinfo/software/samplecode/). You should still read this section so that you understand the key elements of the code and what tradeoffs you may encounter.

### Visual Basic 6

Even though VB6 is an older language, its refreshing simplicity makes it a compelling choice for quickly developing an Ethernet application.

To communicate over Ethernet TCP from VB6, you’ll need the Winsock control (MSWINSOCK.OCX), which is included in the Professional and Enterprise editions of the language. To configure an instance of Winsock, you

must specify the protocol as TCP, choose a local port number, and set the remote IP address and port number to match the drive. In the code example below, 7776 is the port of the drive. *driveIPAddress* is the IP address of the drive ("10.10.10.10" or "192.168.0.130" for example).

```
Winsock1.RemotePort = 7776
Winsock1.RemoteHost = driveIPAddress
Winsock1.Protocol = sckTCPProtocol
```

Sending "RV" command:

```
Dim myPacket(0 to 4) as Byte ' declare a byte array just large enough
myPacket(0) = 0 ' first byte of SCL opcode
myPacket(1) = 7 ' second byte of SCL opcode
myPacket(2) = "R" ' R
myPacket(3) = "V" ' V
myPacket(4) = vbCR ' carriage return

' after 20 seconds of inactivity, the drive will close the connection
' resulting in a Winsock state error
' check for the error and close this end of the connection
If Winsock1.State = sckError Then Winsock1.Close
' if necessary, open a new connection
If Winsock1.State = sckClosed Then
    Winsock1.Connect
    Delay 0.1 ' 0.1 second delay
End If

If Winsock1.State = sckConnected Then
    Winsock1.SendData myPacket
End If
```

To receive a response, you will need to place some code in the Winsock\_DataArrival event. This event is automatically declared as soon as you add a Winsock control to your form. The DataArrival event will automatically trigger each time a packet is received. The code below extracts the SCL response from the payload and displays it in a message box.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim udpData() As Byte, n As Integer
    Dim hexbyte As String, packetID As Long, SCLrx As String
    Winsock1.GetData udpData
    ' remotehost gets clobbered when packet rec'd,
    ' next line fixes it
    Winsock1.RemoteHost = Winsock1.RemoteHostIP
    ' first 16 bits of packet are the ID (opcode)
    If UBound(udpData) >= 1 Then
        packetID = 256 * udpData(0) + udpData(1)
        If packetID = 7 then ' SCL response
            SCLrx = ""
            For n = 2 To UBound(udpData)
                SCLrx = SCLrx & Chr(udpData(n))
            Next n
            MsgBox SCLrx
        End If
    End If
End Sub
```