Using *Networked Equipment* for Remote Station Control

Larry M. Phipps, N8LP

There has been increasing interest recently in the concept of remotely controlling a radio station. The author's situation is similar to that of many pursuing this goal. I live in a subdivision that restricts antenna height, but even if that were not the case, the size of antenna farm I eventually plan to build for DXing would be out of the question in almost any residential neighborhood. Add to that the problems of TVI, RFI and receiver noise, and it becomes attractive to move the station away from the home location. In many cases, just being able to control the station from the office, a summer home, or the family room is desirable. Another scenario is a pooled club station that members could access. How to do that with a minimum of sacrifices in terms of operating control and feedback is a challenge. I believe I have developed a system that, at least for me, gives the kind of control and feedback that I need to feel comfortable with remote operation.

The system I describe has been in use in one form or another since June 2002. I started out using commercially available components, but I have developed some of my own easily duplicated remote hardware over the period. I also offer freeware control programs that will work with my hardware or commercially available hardware.

This article is not intended to be a step-by-step construction article, but rather a template for construction of a customized remote control system based on the concepts that I have developed and will outline. If you want to duplicate my setup, either wholly or in part, details are available on my Web site.[1]

Typical Remote Control Systems

There are currently a number of clever systems in use to control a remote station, whether the link is over phone lines, VHF/UHF, the Internet or other conduits. The most popular method has been to set up a computer at the remote site, interface it to all of the station equipment through serial ports, and install a CAT control/logging program to control everything. The remote software can be controlled through remote desktop sharing programs like *pcAnywhere*, *VNC* (freeware) or the built-in Desktop Sharing in *Windows XP*.

At least one CAT program, *TRX-Manager*[2] by F6DEX, has a built-in remote control capability that uses telnet or packet radio to link a "master" copy of the program running at the home location to a "slave" copy running at the remote site. Many of my programs will work through the remote telnet facility of *TRX-Manager* in addition to the method I describe below. There is more information about these programs on my Web site as well.

Remote control your station using a combination of hardware and software solutions from N8LP.

The one thing these two methods have in common is that they require a computer at the remote site, with a lot of software running on it. This adds a level of complexity, outside of the operating requirements, for the purpose of remotely maintaining the computer and software at the remote site. Ideally, the maintenance method should not rely completely on the computer either, since it is probably the weakest link!

The N8LP System

I had several goals in mind when developing my system. The first was to have complete control of the station—of all functions and all devices. Second—have an extensive feedback system to monitor all remote parameters. Third—eliminate the computer and attendant software at the remote site. I wanted to be able to use a number of different software packages to control the remote equipment depending on the task at hand. The best program for chasing DX is not the best program for PSK31! It seemed to me that the best way to accomplish this was to keep all of the software on the home computer.

I believe I have accomplished these goals. My system uses a combination of commercial and homebrew parts. All of the homebrew parts can be duplicated with commercially available products if desired. On the software side, I have used a combination of commercially available software and my own freeware applications, written specifically for this project. More details about the actual components used, along with links to manufacturer sites and rough pricing are available on my Web site. My software is available for downloading at no cost, along with step-by-step installation and configuration advice on these and other software packages referred to in this article. If you prefer to use only off-the-shelf hardware and software components, the entire system can be duplicated for less than the cost of a mid-level transceiver.

For a clearer idea of the system, refer to the block diagram of Figure 1. Before purchasing the property and raising the antennas, I wanted to verify the system. As a result, my "remote" site currently is the basement, and my "home" site is my laptop computer in the family room. I tested it with ISDN for several months to verify its viability, but now use it over my home wireless LAN. I have also tested it over the Internet using a combination of my DSL account and a dial-up connection. The only difference in these setups is the networking hardware. Figure 1 shows ISDN modems for the connection, but these can easily be replaced with a wireless access point (WAP), a DSL router, a cable modem or other network-connected device with a built-in or stand-alone router. The "system" is very customizable. The key requirements are a network connection of

[1]Notes appear on page 43.

From October 2005 QST © ARRL

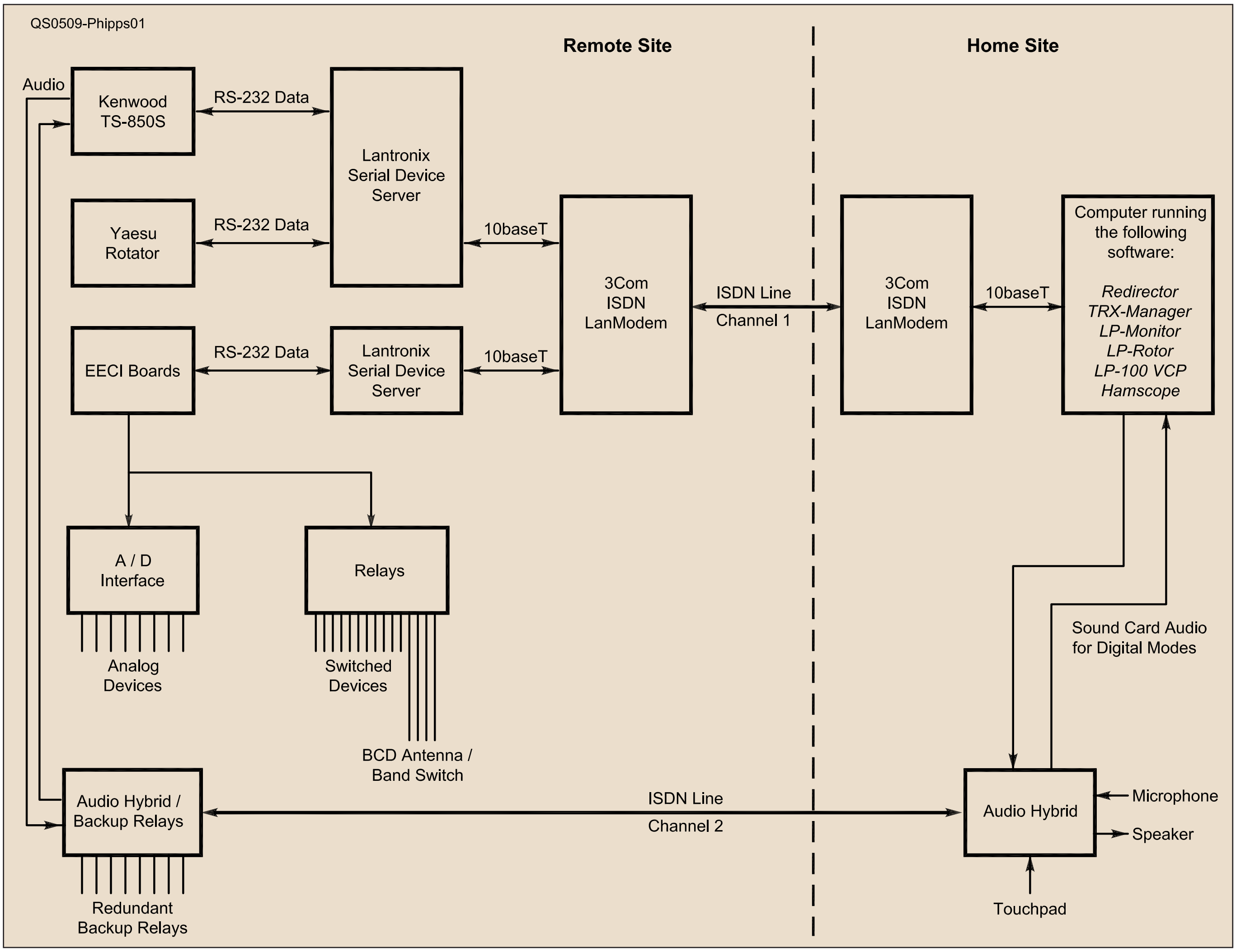

Figure 1—A block diagram of the N8LP remote base station. Notice the use of the "serial device servers" that the author discusses in the text.

almost any kind that supports TCP/IP, and a two-way audio link. The audio can use a VHF/UHF link, ISDN or even VoIP over the network connection utilizing standalone VoIP adapters like the Cisco ATA-186. In my case, I used ISDN at first, and now use a VHF link. If you plan to use PSK-31 or other digital modes, I would recommend against a VoIP audio solution.

I have not tested it yet, but an obvious audio solution to try with a broadband connection appears to be dedicated hardware VoIP adapters like the Cisco ATA-186 Analog Telephone Adapter.[3] One of these would plug into the router at each end of the connection and provide direct dial-up of two "phone" lines through the connection. I see these items on sale at auction sites like eBay for relatively low cost, as they have been used for Internet phone services (such as Vonage) for quite some time.

The Remote Setup

Referring to the block diagram, the equipment that is located at my remote site is:

- The station transceiver, a Kenwood TS-850S.
- The rotator, a Yaesu G-1000SDX with Idiom Press Rotorcard interface.[4]
- A set of control boards from EECI to control relays and provide analog feedback.[5]
- Lantronix Serial Device Servers for converting the equipment serial ports into network accessible TCP ports.[6]
- A 3Com ISDN LanModem for dial-in network access to my equipment ports.[7]
- My own LP-100 wattmeter design, which offers serial remote control.
- A SteppIR antenna controller.[8]
- An audio hybrid for audio isolation and interfacing to the ISDN line, with a DTMF decoder for independent control backup.
- An antenna switch that uses BCD band data. Mine is homebrew, but there are many commercial units available.

For backup purposes, almost all of the equipment is powered by a deep-cycle battery—on constant charge. A dc-dc inverter is also provided for equipment like the rotator, which requires 110 V ac. Other equipment, such as a linear amplifier, could be added, but I haven't gotten to that point yet. A solid-state amplifier in the 500-600 W class could be added easily, and I intend to do that next.

The key concept that allowed me to accomplish my goals is that of converting the RS-232 serial ports on the equipment to TCP ports which can be accessed over a 10baseT ethernet connection. The device that performs this critical task is called a Serial Device Server. These devices were originally designed for use in industrial control, and as interfaces for retail point-of-purchase devices such as card readers, scanners and cash registers. As you can see from Figure 1, each piece of equipment plugs into a Lantronix serial device server using its serial port input.

In my setup, the 10baseT output of each Lantronix Server is fed to a 3Com ISDN LanModem,[9] but as mentioned earlier, it could plug into a variety of devices depending on the network connection you are using. The 3Com device combines an ISDN modem with a router. In another iteration of my system I used a Linksys router and a DSL modem to control my equipment over the Internet. I currently use a Microsoft WAP with built-in router. A higher power WAP could also be used for wireless access over reasonably short distances, perhaps as far as 5-10 miles (line of sight), using dish antennas at each end.

For greater distances, I prefer the ISDN approach because it offers a private, interference-free dial-in connection to the remote site. It is also secure, since both the LanModem and the Hybrid/DTMF decoder box require passwords. Since ISDN provides for two "B" channel 64 kbps digital connections over one line, it allows for a high quality audio connection simultaneously with a data connection, with low latency. Since my system, with all devices connected, requires only about 25 kbps of bandwidth, this is a good match. It may also be easier to get an ISDN line in the boonies rather than a broadband cable connection or DSL, and it simplifies the audio connection.

The serial device servers can be hardware devices, or they can be implemented in software using an old computer. I have tried both approaches and they both work, but to keep things simple, I chose the "black box" approach for my final setup. Detailed information on the software approach can be found on my Web site.[10]

I chose the Lantronix[11] devices after testing a number of brands. Lantronix sells a variety of devices that could be used. I started out with a combination of a standalone model (UDS-10B), and an "embedded" model (Cobox Micro), which is a small circuit board meant to be integrated by manufacturers into existing equipment. The embedded devices have two serial ports; one with full handshaking and one with just serial T/R lines, and the second port is TTL level, not RS-232 levels. I don't recommend using an embedded device now that better alternatives are available. Most ham equipment doesn't require handshaking, but you may need the handshaking signals for PTT or cw keying depending on your rig and control/logging program.

Serial servers are available with one to 32 serial ports with full handshaking. I have recently added ports using a model ETS-8P with 8 serial ports, and will probably eliminate the older boxes at some point. Although pricey when new, the ETS-8P is regularly available on the auction sites. I obtained mine for $35. The more ports on a device, the lower the cost per port. Also, a unit with 8 ports, for example, may entirely eliminate the need for a router in your installation.

The relay control and A/D converter boards I use are from EECI.[12] They are available in a "building block" style, so you can buy just what you need. I chose a 16 channel relay board, and a 16 channel A/D board. You can buy 8 channel versions, but you save very little money. The relay control board output is TTL level, and interfaces to a number of different relay cards though a ribbon cable. You can skip the relay cards and buy a terminal block card, which you can connect to with your own relay drivers and relays of your choice. EECI provides plenty of interfacing information.

The EECI hardware is old school, but well built and reliable, and the best of what was available when I built my first system three years ago. I have also breadboarded up my own hardware controller using a 16F877 PIC microcontroller chip, and developed a new control program called *LP-Remote*. If there is enough demand, I will make a PCB available for this project, as it is a much cheaper way to go than the EECI hardware. It can also be built on a development board, available from several sources, that includes all the PIC parts with room for user circuits. There is more info on this project on my Web site. My PIC board has the advantage of providing feedback status for the remote relays. The EECI hardware can be configured to do this by adding more boards, but, in my opinion, that gets cost prohibitive. In addition, there are other commercial boards available now that work like my PIC board, but they are generally expensive.

The first 12 relays on my EECI board are controlled directly from my *LP-Monitor* program by 12 customizable buttons, and the last 4 relays provide BCD band data that follows the band selector in *LP-Monitor*, which, in turn, follows *TRX-Manager*. These last 4 relays can be used to interface to an antenna switch such as those from Top-Ten Devices[13] and others, or you can build your own relay box.

The A/D boards terminate in a terminal block, which accepts analog data in the range of 0-5 V dc. You will need to "condition" any analog signal you wish to monitor to provide this range to the A/D board. Any appropriately conditioned analog signal can be monitored including power, SWR, line voltage, audio line levels, battery voltage and shack temperature. *LP-Monitor* has eight meters that are individually user customizable to read any range or parameter. Originally, I pulled FWD and REF signals out of an old wattmeter to get power and SWR, but now use my LP-100, which provides much more info over a serial connection.

I also provide a simpler program, called *LP-Relays*, in two variants. One version mimics the relay panel in *LP-Monitor*, and a second eliminates the BCD band encoder and provides 16 separate relay buttons.

The Home Setup

Referring back to the block diagram, most of the home site is implemented in software. The only hardware required is for connecting to the ISDN lines. On the audio side, a simple audio hybrid is used for interfacing transmit and receive audio, along with a sound-card interface for mic input, RTTY and the digital modes, and a powered speaker for receive audio monitoring. A DTMF touchpad is provided for sending codes to the decoder at the remote site for independent backup control functions, in case something goes wrong with the ISDN modem or serial servers.

On the control side, I am using another LanModem because I use the built-in router to feed other computers at the house, but any ISDN modem should work. Of course, if your connection is other than ISDN, different audio and networking devices would be used.

Here is a list of the software in use at my home site:

- *TRX-Manager* for rig control and logging.
- *Serial/IP* for COM port redirection.
- *LP-Monitor* for relay control and meter display of remote analog parameters.
- *LP-Rotor* for rotator control.
- *LP-100 Virtual Control Panel* for monitoring of remote LP-100 wattmeter.
- *Hamscope* and *MixW* for RTTY and the digital modes.

Figure 2 is a screen capture from my laptop computer showing most of the main software packages in use. A larger screen with 1280×1024 resolution or a two-monitor setup would obviously allow more simultaneous software windows. On the left is *TRX-Manager*, the upper right shows *LP-Rotor* and the lower right shows *LP-Monitor*. I chose *TRX-Manager* as my main control program because it is a very extensive package that does just about everything I need. Laurent, F6DEX, the developer of *TRX-Manager*, is very supportive of his product and receptive to third party developers. As a result, even though I originally wrote the "LP" programs to be complete stand-alone programs, they also are able to exchange data with *TRX-Manager* through DDE and OLE linking to make for a nicely integrated package.

I usually use *Hamscope*[14] for RTTY, MFSK and PSK modes, because it also is linked to *TRX-Manager* through OLE for rig control and logging. It is interesting to note that because of the real-time tuning indicators used in *Hamscope* and its cousins, it would be very difficult to run such applications on a remote computer using a desktop sharing program for feedback.

The key software component at the client site is the "COM port redirector." It is used to convert the TCP port data (generated by the serial device servers at the remote site) into data that is accessible to programs which expect the equipment to be connected to a COM port. In other words, it is a multiple TCP port to virtual COM port converter. The client applications see these virtual COM ports as hardware devices and can connect to them as though they were real.

For my COM port redirector I use a program called *Serial/IP* from Tactical Software. This program is no longer available, and its replacement is more costly. Lantronix offers a free redirector that can be downloaded from their Web site. I am more impressed with the Tactical software, but a number of hams who have adopted my system are using the Lantronix redirector without problems, and the price is right.

In my case, I use TCP ports 14003-14007 for my equipment, with each piece of equipment dedicated to a port. At the home site, *Serial/IP* sets up a one-to-one correspondence between each TCP port and a virtual COM port. As an example, in my setup I use the following settings:

Equipment	*TCP Port*	*Virtual COM Port*
SteppIR Controller	14003	COM 3
LP-100	14004	COM 4
EECI Boards	14005	COM 5
Kenwood TS-850S	14006	COM 6
Yaesu Rotator	14007	COM 7

You can choose any ports that are not in use at either end, and can control up to 256 devices with the proper server.

Configuration of the Lantronix devices depends upon which model is being used. Most models use an interface that is accessible from a Web browser. This can be done over the network connection. Once you have set up your devices and installed the redirector software, all that is left to do is to tell your control software which virtual COM port each piece of equipment is connected to.

Summary

I have been using this system for three years now, in one form or another. The system is gaining in popularity as the switch to remote control gains momentum. I have helped numerous operators develop versions of the system worldwide.

When I started out on this project I had not even heard of serial device servers, but after extended use of alternative methods using computers at both the home and remote sites, I am sold on this system. It allows for extreme simplicity at the remote site with no worries about computer configuration and maintenance, crashes, reboots, etc. It also allows for multiple control-software choices, with quick and easy reconfiguration of the software at home.

Since the software is on the home computer, and only the port data is remote, the programs "feel" pretty much like you are controlling a local transceiver. You will notice some slight delay in the transceiver's response to some commands because of the inherent latency of the connection, but this is the case with any kind of remote control operation. One other caveat is in order. I have not tested my system with any transceiver or software other than what is described in this article. If you want to use a different transceiver or transceiver-control/logging software, for instance, you will need to run some tests. Be sure to test with the actual network connection you plan to use, since latency can make a difference with some applications.

While this system is not the only one available for remote control, and it is probably not the best one for every need, I hope I have provided food for thought in the quest for remote station control. For more details on the system, and the latest information on hardware and software additions, please check my Web site.

Notes

[1]**www.telepostinc.com/n8lp.html**.
[2]**www.trx-manager.com**.
[3]**www.cisco.com/warp/public/cc/pd/as/180/186/**.
[4]**www.idiompress.com**.
[5]**www.eeci.com**. Details are available at the author's Web site. These boards are used and supported in freeware applications that are also discussed in this article.
[6]**www.lantronix.com**.
[7]**www.3com.com/products/en_US/detail.jsp?tab=support&pathtype= support&sku=3C891A-US**. This device is now discontinued, but datasheets are still available at the Web site. Used units can be found on the auction sites (like eBay).
[8]**www.steppir.com**.
[9]See Note 7.
[10]See Note 1.
[11]See Note 6.
[12]See Note 5.
[13]**www.qth.com/topten/**.
[14]**www.qsl.net/hamscope/**.

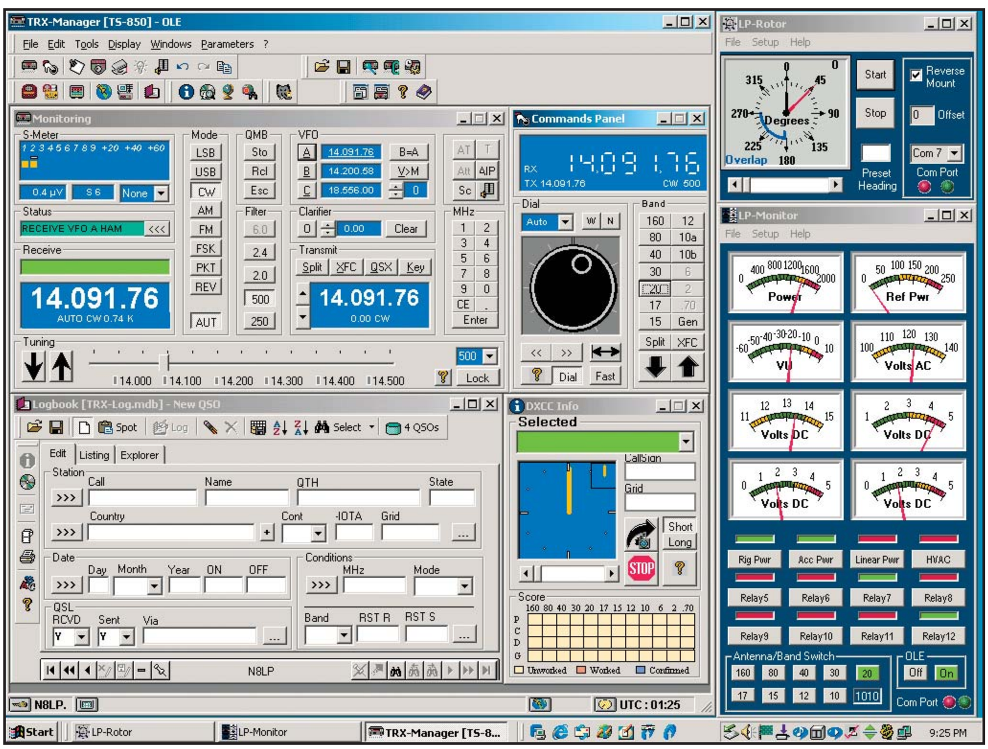

Figure 2—A screen capture of the N8LP remote base station software.

Larry Phipps, N8LP, has been licensed since 1965, first as WN8SPD, later as WA8SPD, and finally as N8LP, when he upgraded to Amateur Extra. Larry also earned an FCC First Class Radiotelephone license in 1967. He is retired from TelePost, Inc, a television post-production facility that he and his wife Janet founded and owned for 16 years. Larry worked for WJBK-TV, Detroit, while attending the University of Michigan College of Engineering, leaving school in 1971 to work full-time at WJBK as a broadcast engineer. He then went to NET Television Inc (a subsidiary of WNET-TV, New York) where he was instrumental in launching one of the first computer-controlled videotape editing systems in the country. Larry is now devoting time to his house, his wife and ham radio (not necessarily in that order!), all sorely ignored while he and his wife operated the company. You can reach him at 49100 Pine Hill Dr, Plymouth, MI 48170; **larry@telepostinc.com**.