A Raspberry Pi Net Server/Client for Antenna Rotor

Tom Doyle • W9KE

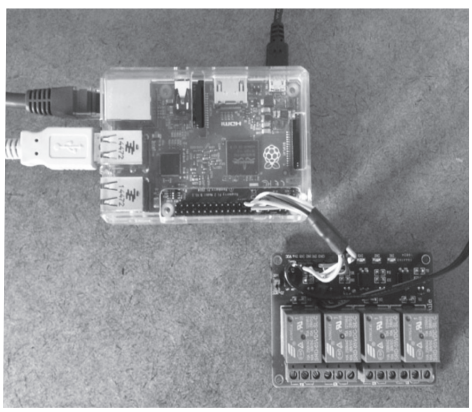
Net Server

A net server is a device used to create a network connection to an antenna rotor system. A net server can be especially useful when operating portable or from a remote location. While there are many different ways to do this, such as running GPredict software on a Raspberry Pi computer, the particular solution presented here is designed to use the Raspberry Pi with as much of your existing hardware and software as possible.

In the solution described here, the Yaesu G-5500 serves as the rotor and an LVB Tracker (USB interface) as the controller. The Raspberry Pi Net Server (Net Server) optionally can control four relays using a very high quality board currently available for about $10 on Amazon. These relays can control things like antenna polarity switches and amplifiers, or they can be used to turn equipment on and off remotely. The relay board is not required and could easily be added later. If you are interested in only the remote control relays, the programs will run without an LVB Tracker connected.

The hardware is a Raspberry Pi 2 shown in Figure 1, along with the relay board for controlling other devices. Three cables are connected to the Raspi: an RJ-45 network cable, a USB cable from the LVB Tracker, and a power cable that goes to a common USB charger. Five wires connect the Raspi to the optional relay board.

Figure 1



If you do not already own a Raspberry Pi, the Raspberry Pi 2 Model B (Raspi) is an excellent choice. If you purchase a Raspi, select the default Raspbian version of UNIX while doing the initial setup. For more information, see **www.raspberrypi.org/help/quick-start-guide/**.

The initial setup of the Raspi requires that a keyboard, mouse and monitor be connected to the device. After the initial setup, the keyboard, mouse and monitor may be removed and the device operated using what is referred to as a remote desktop connection. Since the Raspi uses a multi-tasking operating system, it can run the Net Server program and another server program to allow access from a remote device at the same time. This means that you do not need a monitor, keyboard or mouse connected to the Raspi after the initial setup. It also means the little two line LCD monitor we often see on these devices is not required. A Raspi operated this way is often referred to as "headless."

There are two parts to a remote desktop connection. The first part is a program named "xrdp" that runs on the Raspi. This program can easily be installed by entering the following command into a terminal window after the initial setup: sudo apt-get install xrdp. Once xrdp is installed and you have rebooted your Raspi, you can forget about xrdp. It will start automatically every time your Raspi boots up.

The other part of the system is the remote desktop client that runs on your device. The device can be a PC, Mac, or even an iPad.

Windows includes a remote desktop client called Remote Desktop Connection. When you run this program on your device, you should see a screen like that depicted in Figure 2.

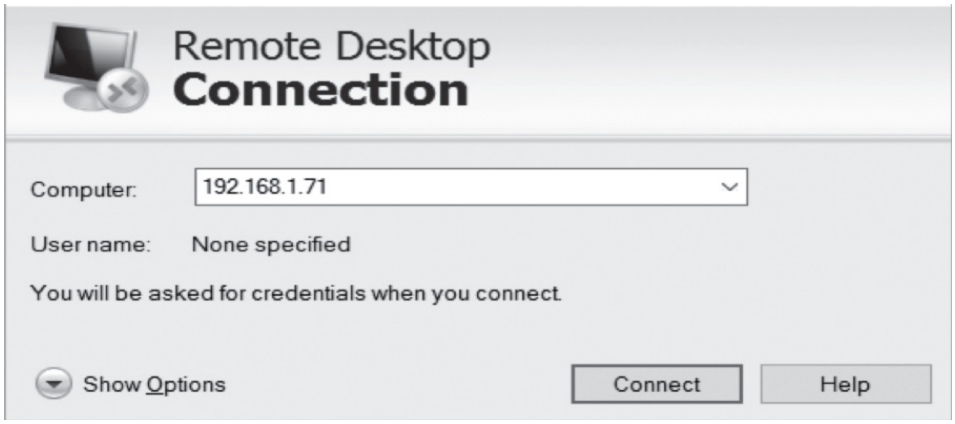
Enter the address for your Net Server in the Computer text box and click on the Connect button. Note you do not have to enter a port number; the default RDC port number is selected automatically. When the connection is made, you should see a screen that looks like the depiction in Figure 3. Leave the Xvnc Module selected, enter your Net Server username (defaults to pi) and password (defaults to raspberry), then click on the OK button. You will then be taken to a standard Raspi GUI window.

For the iPad, you will first need to download a free app from the Apple app store called Microsoft Remote Desktop (Figure 4). This is an excellent app and, in some ways, is easier to use than the PC version.

After you have downloaded and run the app, create a connection to your Net Server by tapping on the little + in the upper right corner of the app screen. This will be stored, so you only have to do it once. Tap on the connection that you created and you will see a screen that looks like Figure 5. Enter the user name and password as usual and tap on Done. This will take you directly to the Net Server GUI.

You may not realize how powerful the Net Server remote desktop connection is. Figure 6 shows a partial screen shot of the iPad running Microsoft Remote Desktop connected to the Net Server, which is currently running GPredict. This also can be done from a PC using Remote Desktop Connection. It is a bit easier on a PC than the iPad because you can use a mouse on the PC, but both work fine.

Figure 2



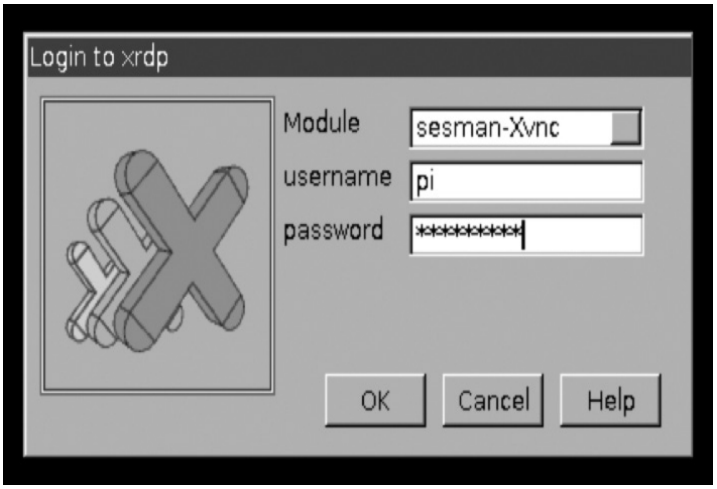


Figure 3

Figure 4

You also can use the remote desktop connection software on a PC or an iPad to connect to standard PCs. If you are using a "Pro" version of Windows, the server side software is built in. Figure 7 shows a partial screen shot from the iPad running Microsoft Remote Desktop connected to a Windows 7 Pro desktop running SatPC32. If you are not using Windows Pro, other third party remote desktop servers are available.

We have come a long way thanks to the great work done by Erich, DK1TB, with SatPC32 on the PC, Mark, N8MH, with SAEBRTrack on the Basic Stamp, and Howard, G6LVB, with LVB Tracker on the PIC. As they say, "the only reason we can see so far is we stand on the shoulders of giants."

After you have established the remote desktop connection, you can operate your Raspi from your device the same way you would operate if you were using a keyboard, mouse and monitor connected directly to the Raspi. This helps reduce clutter in the shack and also reduces the cost of the system. It is also a very important advantage when operating portable.

When Raspi is up and running, create a directory under your home directory for the Net Server program. The directory name is not critical. I used "net_server" for the directory name. You can use the Raspi GUI or a terminal window.

Download and install the Wiring Pi interface library. This library provides access to the input/output pins on the Raspi GPIO connector. For more, see **wiringpi.com/download-and-install/**.

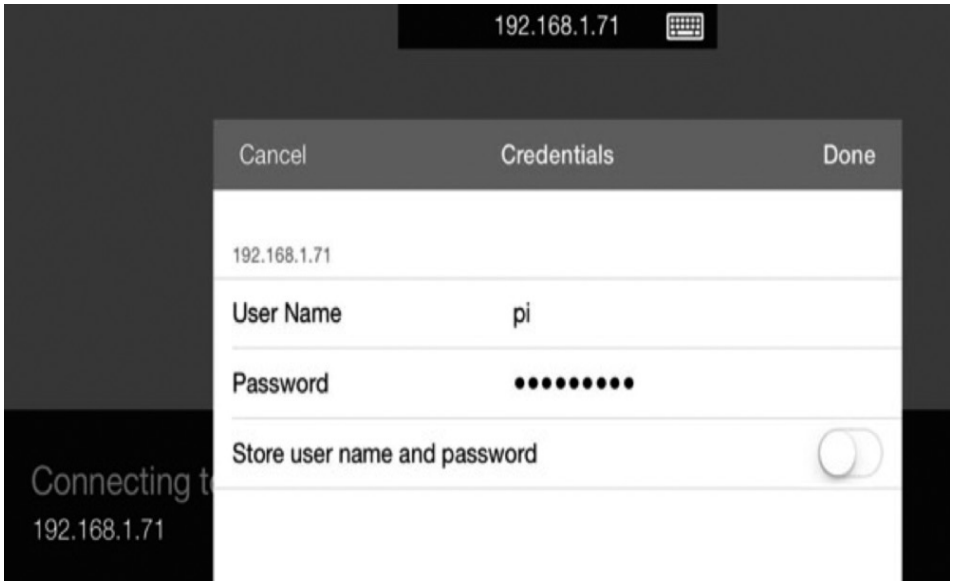


Figure 5

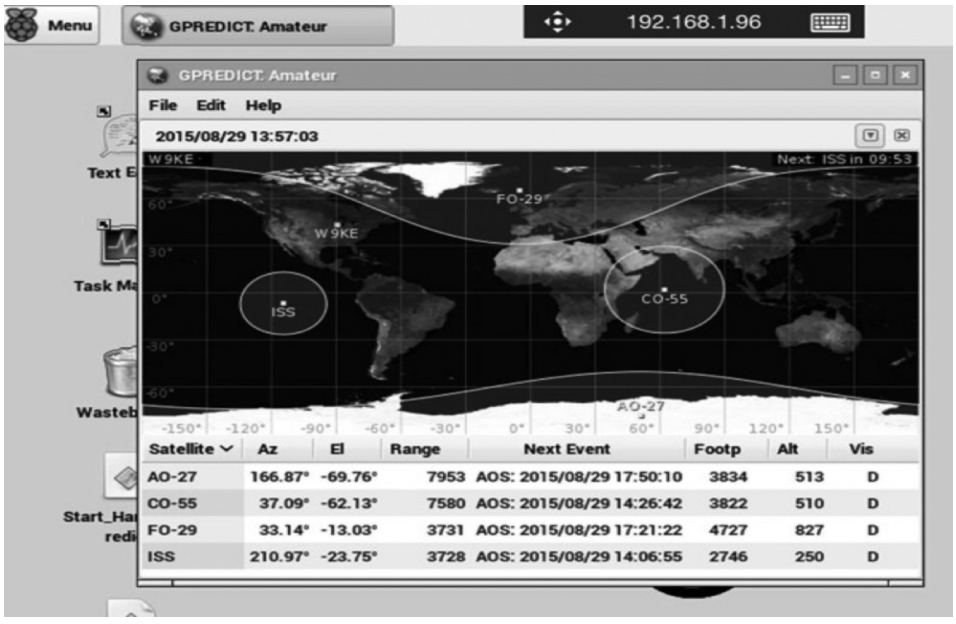


Figure 6

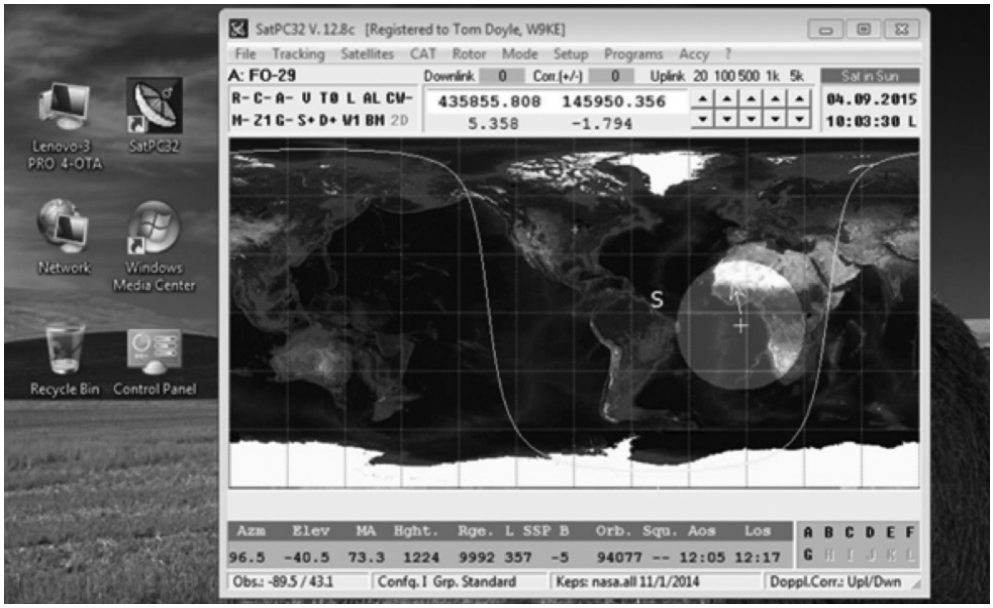


Figure 7

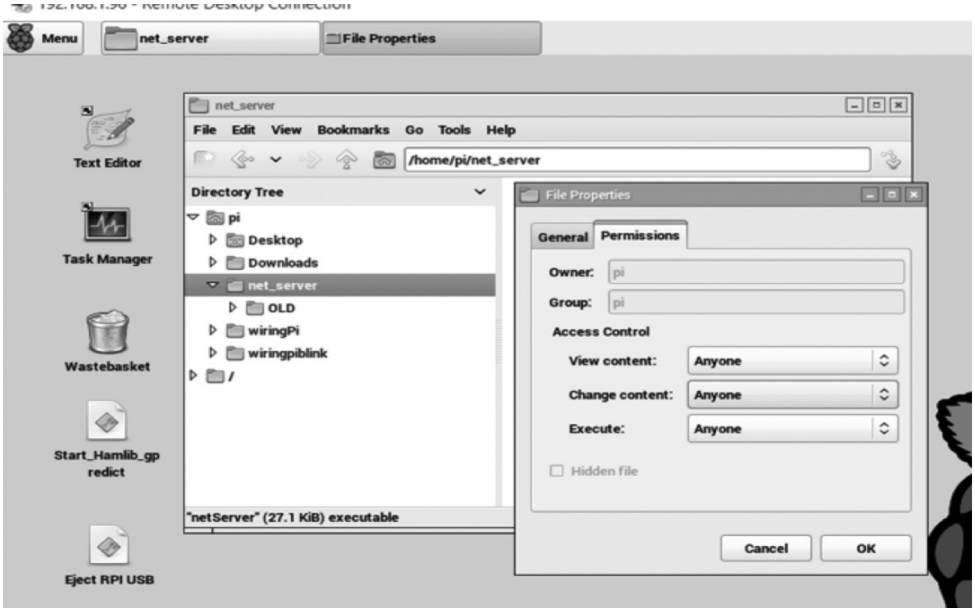


Figure 8

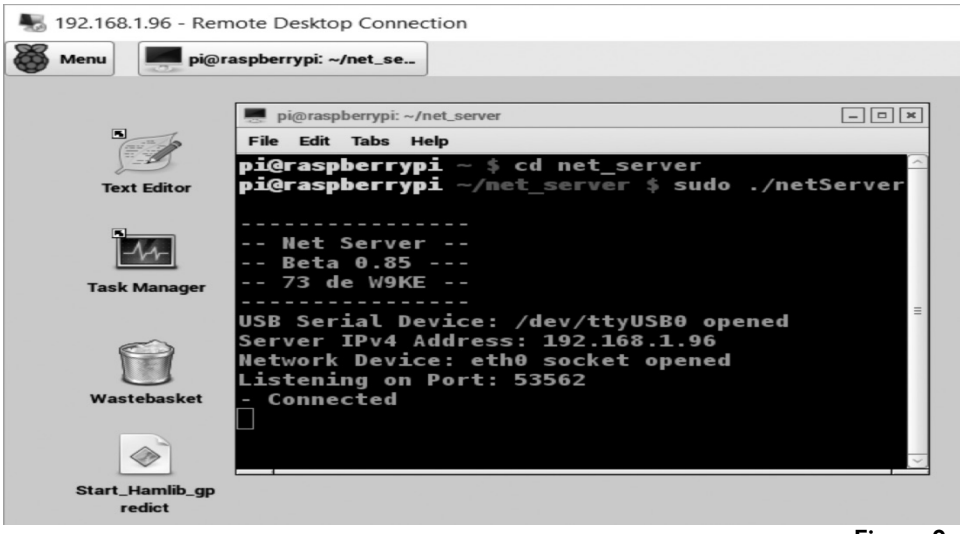


Figure 9

When you are ready to try out the program on your Raspi, send me an email, and I will send the program to you. It is a compiled binary named netServer designed to run on the Raspi, not X86 boxes.

Copy the program into the directory you created earlier. UNIX security probably will not permit you to run the program until you change the access rights. You can do this with "chmod" in a terminal window, but I find it easier to use the File Manager program available in the Raspi GUI. Note in Figure 8 that all the permissions for the netServer file have been set to "anyone."

You should now be able to run the netServer program. This is a console program designed to run from a terminal window. For this example, I opened a terminal window in the GUI. When you open a terminal window, it defaults to your home directory. The default user name is "pi," so I left it as-is for this example.

In the terminal window image (Figure 9), you can see the two commands necessary to start the program:

- 'cd net_Server' moves you into the directory we created earlier that contains the netServer program, and
- 'sudo ./netServer' causes the program to run.

The "USB Serial Device" is the USB connection to the LVB Tracker. If you do not have the LVB Tracker plugged in, it will complain. That is fine. Just shutdown and, if you have one, plug the tracker into the Raspi and start the Raspi back up (you probably could do a hot swap but just to be safe, power down and reboot). You can continue even if you do not have the LVB Tracker connected.

The "Network Device" is the Ethernet connection to your network through the RJ-45 connector. You will need the IP address and the port number later, so write them down.

When the program runs, a configuration file will be created in the same directory as the program named NetServerConfig.txt (Figure 10). You can see it opened in the Raspi GUI Text Editor. This editor is very nice. Having been stuck with VI for many years back in the day, it is a real treat to use this editor. If you need to change the port, network device,

or USB device, you can do it here. Do not change any of this unless you are sure you know what you are doing.

Net Client Beta

The Net Client program, a Windows program, communicates over the network with the Net Server. When you run it, you should see something similar to what is shown in Figure 11. Ignore any message about the Serial Port not being configured; we do not need the serial port at this time. Enter the IP address and Port number you saw in the terminal window for the Raspi when you ran the netServer program. After you have entered the correct IP Address and Port, click on the Save/Restart button at the bottom of the window (the button should have turned yellow when you changed a value to remind you to save it). When you do this, updated values will be saved and the program will restart. The little button to the right of the Network label should turn green if you have a good network connection (it was most likely red when you first ran the program).

If you have an LVB Tracker connected to the Raspi, you should see the current AZ and EL for your rotor in the lower left part of the window in the section labeled "Rotor." This value is updated at an interval shown in the "Interval ms" box. Unless you have some really old slow hardware, you should not have to change it. If you make the value too low, you will have an unsatisfactory user experience.

The group of four buttons and indicators in the bottom right of the window (S1-S4) control the four relays you saw in the images of the Raspi Net Server system. For example, if you click on the S1 button, a message will be sent to the Net Server telling it to activate relay 1. When this is done, the Net Server will send back a message indicating that relay 1 is activated, and the little button indicator to the left of the S1 button will turn from red to green. This means that the indicator button will not turn green immediately after you click on the button, or it may not change at all if you do not have a network connection and a working Net Server. This system is smart in that it waits to see if the message was received by the Net Server. However, it is not brilliant because it cannot tell if you have a relay board installed.

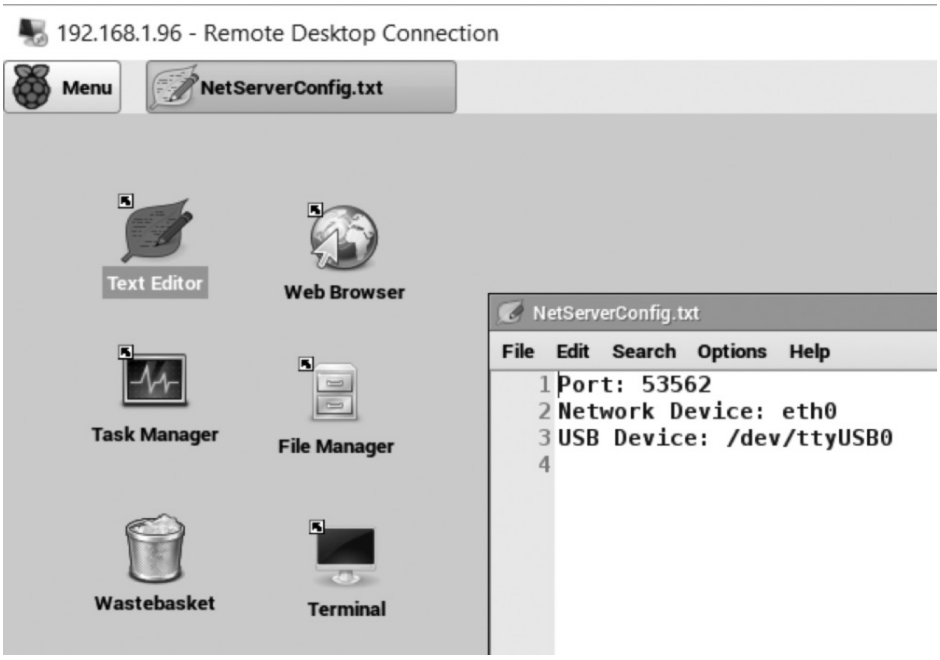


Figure 10

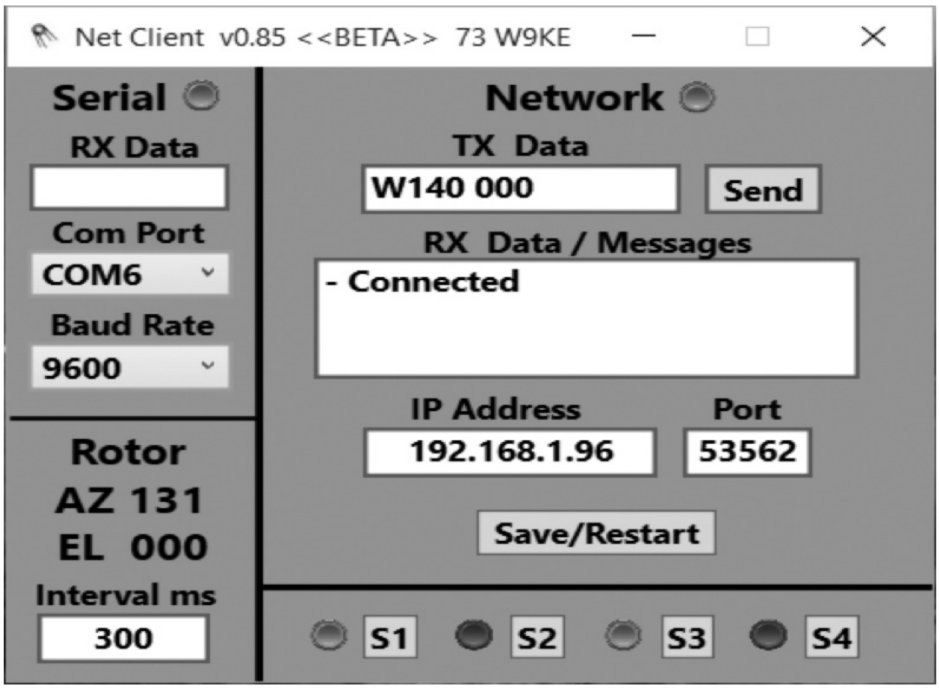


Figure 11

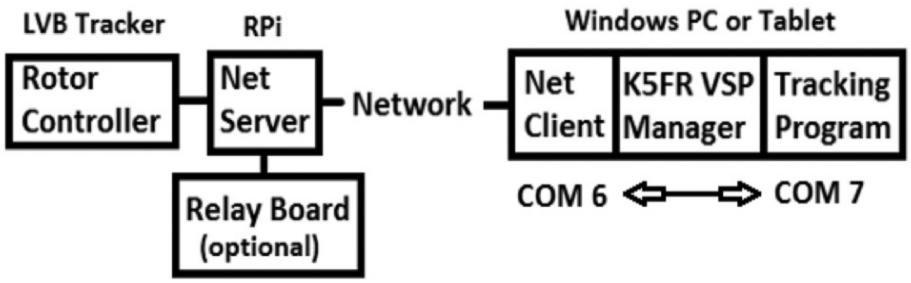


Figure 12

If you enter a valid LVB Tracker/Yaesu GS-232 command to move rotors, like "W140 000," as shown in Figure 11, and then click on the Send button, the rotor should move to the new position. The position information shown in the Rotor portion of the display will track the move with updates at an interval specified in the "Interval ms" text box.

If you make any changes to the values in the window other than the "TX Data" text box, the Save/Restart button will turn yellow to remind you to save your changes. When you click on the Save/Restart button, the values will be saved and the program will automatically restart. When deciding what to hook up to the relays, remember that the relays will all be turned to the off state whenever the Net Server starts up.

The Serial portion of the screen shows the controls used to configure the serial port that is used to connect Net Server to satellite tracker programs that have a serial interface. Use the K5FR VSP Manager program to set up a pair of virtual serial ports. Connect one of the ports to your tracker program and the other port to Net Client using the boxes in the Serial part of the Net Client window.

When the serial port connection is made, the indicator button to the right of the word Serial will turn green. For this example, I have Net Client connected to virtual port COM6 at 9600 Baud and the other virtual port of the pair COM7 connected to the tracking program. This way the tracking program can control the rotor. Figure 12 shows the path from the tracking program to the rotor controller and relay board. It is not quite as complicated as it looks.

When you are ready to try the Net Client program, send me an email at tomdoyle1948 at gmail dot com, and I will send you the program.

The only supported platform is the Raspberry Pi model 2 B with the default raspbian wheezy installed. This does not mean it will not work on other platforms; it just means if you use something else and it does not work, I probably will not be able to help.

Support AMSAT

AMSAT is the North American distributor of SatPC32, a tracking program designed for ham satellite applications. For Windows 98, NT, ME, 2000, XP, Vista, Windows 7, 8/8.1 & 10.

Version 12.8c is compatible with Windows 7, 8/8.1 & 10 and features enhanced support for tuning multiple radios.

Version 12.8c features:

- SatPC32, SatPC32ISS, Wisat32 and SuM now support rotor control of the M2 RC-2800 rotor system.
- The CAT control functions of SatPC32, SatPC32ISS and Wisat32 have been expanded. The programs now provide CAT control of the new Icom transceiver IC-9100.
- The main windows of SatPC32 and SatPC32ISS have been slightly changed to make them clearer. With window size W3 the world map can be stretched (only SatPC32).
- The accuracy of the rotor positions can now be adjusted for the particular rotor controller. SatPC32 therefore can output the rotor positions with 0, 1 or 2 decimals. Corrections of the antenna positions can automatically be saved. In previous versions that had to be done manually.
- The tool "DataBackup" has been added. The tool allows users to save the SatPC32 program data via mouse click and to restore them if necessary. After the program has been configured for the user's equipment the settings should be saved with 'DataBackup'. If problems occur later, the program can easily restore the working configuration.
- The rotor interfaces IF-100, FODTrack, RifPC and KCT require the kernel driver IOPort.SYS to be installed. Since it is a 32-bit driver it will not work on 64-bit Windows systems. On such systems the driver can cause error messages. To prevent such messages the driver can now optionally be deactivated.
- SuM now outputs a DDE string with azimuth and elevation, that can be evaluated by client programs. Some demo files show how to program and configure the client.

Minimum Donation is $45 for AMSAT members, $50 for non-members, on CD-ROM.
A demo version may be downloaded from http://www.dk1tb.de/indexeng.htm
A registration password for the demo version may be obtained for a minimum donation of $40 for members and $45 for non-members.
Order by calling 1-888-322-6728.

The author DK1TB donated SatPC32 to AMSAT. All proceeds support AMSAT.