WinSystems

QuickStart Guide

Datalight Sockets
TCP/IP Development Kits

WinSystems has teamed up with Datalight, the authors of ROM-DOS, to provide our customer with a cost-effective path to TCP/IP networking. By providing pre-configured systems, a lot of the misery of bringing up a TCP/IP installation from scratch has been eliminated. These preconfigured systems are nearly ready to go out of the box. Refer to the "Read Me First" document that accompanies this guide for the necessary alteration that might be necessary to make the system ready for immediate use. The purpose of this guide is to illustrate what capabilities exist on the system as delivered and how to demonstrate and/or utilize some of these capabilities.

There are two fundamental ways to utilize the TCP/IP networking provided by WinSystems and Datalight. The first method requires no knowledge of direct TCP/IP programming and the user takes advantage of pre-built clients, servers, and utilities to make his application utilize network resources.

The second method uses the documented API calls from Datalight to implement a custom application that is fully TCP/IP aware. A number of sample applications written in 'C' are included on the "Development Kit" diskette that can serve as an example for the API usage. There is also a brief programming TCP/IP tutorial in the "Datalight Sockets" manual. Technical assistance for programming the Sockets API must be addressed directly to Datalight.

Once the system has been powered up and the appropriate configuration changes have been made, the first step is to test that the system can "talk" on the network. If the system is connected to a Windows TCP/IP network, it's a simple matter to "ping" the target system from another machine in a Windows "DOS" box, try:

ping 192.168.0.7

Of course, the IP address will need to be changed to whatever is being used on the target system. If the packets are being echoed back, life is good, and it's time to go on with the next test. If not, it is necessary to verify the cabling, the IP address, the subnet mask, etc. before continuing. If you cannot "ping" the target, nothing else has a chance of working. You can also ping from the target to other computers on the network. The Datalight version on ping is called "XPING".

Once the basic ability to "talk" is confirmed using "ping" and "xping". You'll probably wan to check out the small web page that we've provided on the board. If you're working on a Windows of Linux network just bring up your favorite browser and enter the IP address of the target into the space where you would ordinarily enter a web address, i.e.

192.168.0.7

Most web browsers will automatically add the "http://" prefix to the address you entered and the WinSystems mini web page will be displayed. If the browser does not display the page, it may be necessary to enter a more complete address such as:

http://192.168.0.7

This page demonstrates some of the capabilities of the Datalight Sockets web server. The web page begins at c:\sample on the target system. You may of course replace this web page with one of your own. For another example of the Sockets web server usage, you can access the WinSystems weather station on the web at

http://66.140.41.75/

This weather page is a real-world example of the utility of ROM-DOS and Datalight Sockets. This system is based on the WinSystems PPM-520 running ROM-DOS and Sockets out of an 8MB Disk-On-Chip.

The next item to explore is the remote console Java application. Your browser must be Java enabled, and you enter the URL as:

http://192.168.0.7/remcon.htm

It will take a little bit of time for the Java applet to load and then you should be presented with s screen prompting for a login. You should enter 'admin' for the login name and 'admin' as the password. In just a moment the DOS console screen of the target should appear. You can run programs, rename, delete, copy files, etc. You can even use the screen editor "ned" that is provided with ROM-DOS. This capability can be used for remote control of web enabled applications. Be sure to change the login and password before implementing this capability outside your lab. Refer to the Datalight documentation for controlling users and security.

Another way to access the system remotely over the network is using FTP. IF you're already FTP savy, just enter the IP address of the target into your favorite FTP client, and use the user 'admin' and the password 'admin' to gain full access to the files on the target system. This is a perfect way to update web contents. This facility is used by the WinSystems weather site. Another machine that is connected to the actual

weather instrumentation serially, builds the web images and then FTP's the new contents to the Socket's web server every few minutes.

Users of the later versions of the Microsoft browsers can actually browse in the file systems on the target by entering the following URL:

ftp://192.168.0.7/

A username 'admin' and password 'admin' will be required. It is them possible to browse around the target system and cut, copy, and paste files to or from your desktop and the target. Obviously, FTP is extremely useful for moving files to or from the target system.

Another method of communication with the target without having to write to TCP/IP code is to use email. Datalight sockets provides three command line utility programs, makemail, sendmail, and getmail. These three utilities allow for the formatting of email, the transmission of email, and the reception of email, respectively. This could allow an application to email data, conditions or alarms to interested parties. Email received by an application could also direct operations or set parameters. There are also version of these three programs provided in 'C' source fomr on the "development Kit" diskette. Using these source files as templates could allow an application to "compile in" support for email rather than having to "shell out" to the command line utilities.

It is hoped that this quickstart guide in conjunction with the Datalight manuals and the preconfigures software will give you a leg-up on making your application "network aware" and "web-enabled".