

Getting on the Wire

by Ingo Cyliax

If you're like me, you have a multitude of test equipment at your disposal and at times wish you could centralize its use for certain projects. I often find myself setting up tests where I can automate the computer interactions, but I have to grab pencil and paper to take down voltmeter or temperature settings to go along with the test.

It would be nice to automate the whole data-collection procedure

Voltmeters and test equipment with RS-232 ports exist; however, most PCs only have two serial ports, and they are usually already taken up by programming cables and communications equipment. But by using the Ethernet, we essentially open up an unlimited potential for wiring up devices...

A serial-to-Ethernet bridge widget would be the ticket. We simply connect a serial-based device into the bridge and connect it to the Ethernet. On the PC, we can then use Telnet or other communication software capable of setting up a connection to remote host, and start using the serial-to-Ethernet bridge. The serial-to-Ethernet bridge would nominally transfer characters between the serial port and the Telnet connection.



Let's figure out how to do this. First, we need a processor with both a serial port and an Ethernet port to use as our serial-to-Ethernet bridge. Luckily something like the RCM3000 will do nicely. Simply plug it into a development board, which will provide it with regulated power and RS-232 level conversion, and we are almost ready to go...except the software of course.

The software is pretty straightforward. It initializes the RCM3000, sets up the TCP/IP protocol stack and serial ports, and waits for a network connection from an external host (PC) via Telnet. Once the network connection is made, the software simply moves bytes between a serial port's TX and RX and the network connection's socket.

Bridge Configuration

Network/IP Address Configuration

To tell the software the identity of the serial-to-Ethernet bridge, we need to assign an IP address to the bridge. IP addresses are 32-bit numbers that uniquely identify a device. Besides the IP address we also need a netmask. It's a 32-bit number that tells the TCP/IP stack what part of the IP address identifies the network the device lives on as well as which host it is.

The sample program supplied with the Serial-to-Ethernet Application Kit already configures the serial-to-Ethernet bridge with a default IP address and netmask. Use these instructions if you need to change the settings to configure the serial-to-Ethernet bridge with your own Ethernet settings.

The following are the default IP address and netmask.

IP Address: 10.10.6.230

Netmask: 255.255.255.0

When the software on the RCM3000 boots up, it allows you to change the IP address, netmask number, and other networking and serial configurations via the **RECONFIGURE SETUP** on the LCD/keypad module. If you want to use the serial-to-Ethernet bridge in a local area network (LAN), contact your network administrator to obtain an IP address, netmask, and gateway host to use.

Serial Port Configuration

In addition to the network information, you may also want to program the serial port. The default configuration, is 4800, but the LCD/keypad module allows you to set the baud rate as well as the number of stop bits, type of parity, and size of data word. Consult the manual for the device you want to connect to the serial port to establish the serial port configuration you will need.

Some serial devices communicate in protocols that are not ASCII representations of the data. This means the data that are sent over the serial port are not readily readable since the serial-to-Ethernet bridge acts as a transparent bridge. The software does provide the functionality of converting incoming data into a hexadecimal representation. It also has a translation routine to convert from a specific digital multi meter (DMM). Select this only if you have such a DMM. Many DMMs with serial ports will communicate using ASCII.

LCD/Keypad Module Configuration Menus

To select a menu option, use the **<UP>** **<DOWN>** keys to scroll between selections and hit the **<ENTER>** key to select an option. To exit a submenu and return to the menu leave above it, select the **EXIT MENU** option in each submenu.

The following are the top-level menus to be displayed.

RECONFIGURE SETUP
VIEW SERIAL SETUP
VIEW NETWORK SETUP
DISPLAY DATA PACKET

The following menus and submenus appear when you select **RECONFIGURE SETUP**.

SETUP IP PARAMS
SETUP IP ADDRESS
SETUP GATEWAY
SETUP NETMASK
SETUP SERIAL PORTS
PORT STATUS
SET BAUDRATE
SET DATA TYPE
HEX MODE
BINARY
DMM
BACKLIGHT CONTROL

Once you have selected a menu or submenu, use the **<LEFT>** and **<RIGHT>** keys to select which digit you'd like to change, and use the **<UP>** and **<DOWN>** keys to change the individual digits (0–9). The **<ENTER>** keys will save the new Address to flash memory. Once done, you exit the menu or submenu.

The IP address is the Internet address that the bridge uses to communicate over the network. The gateway address is the Internet address of a router or network access device. The netmask is a mask that tells the TCP/IP protocol stack how to parse the Internet addresses into a host part and a network part. The defaults are as follows.

IP address 10.10.6.230
Netmask 255.255.255.0
Gateway 10.10.6.1

You can also change the serial port parameters by selecting the **SETUP SERIAL PORTS** menu option. Here you are supplied with a list of possible serial ports.

Ser. A Off
Ser. B On
Ser. C Off
Ser. D Off
Ser. E Off
Ser. F Off

While the Rabbit 3000 processor has six UARTs, we use Serial Port B for this application. Select **SER B** to change this port's parameters.

The **PORT STATUS** menu option allows you to enable or disable the serial port. The **SET BAUDRATE** entry lets you can change the port's baud rate.

Select **SET DATA TYPE** to program the port's data-conversion capabilities. The **HEX MODE** will convert all incoming data on the serial port to a two-digit hex representation of the 8-bit data. The **BINARY** mode will pass all serial data straight through to the Ethernet connection. **DMM** allows our application to communicate with the DMM, which is the default.

PC Configuration (Windows 98)

Once the serial-to-Ethernet bridge has been set up, we are now ready to configure the PC that will talk to it via Ethernet. The procedure varies a bit by operating system.

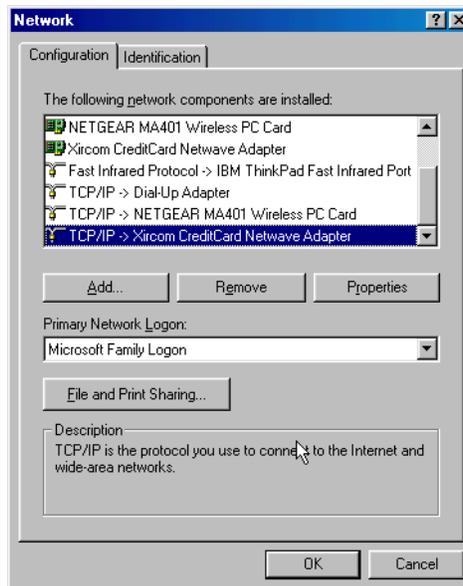
In order to configure the PC to communicate with the serial-to-Ethernet using the sample program and the crossover cable included with the Application Kit, we need to configure the PC to use a static IP address.

1. Go to control panel (**Start >Settings >Control Panel**) and select **Network**.



2. Select the Ethernet Interface that the crossover cable is plugged into (e.g., **TCP/IP Xircom Credit Card Network Adapter**) and click on the properties button.

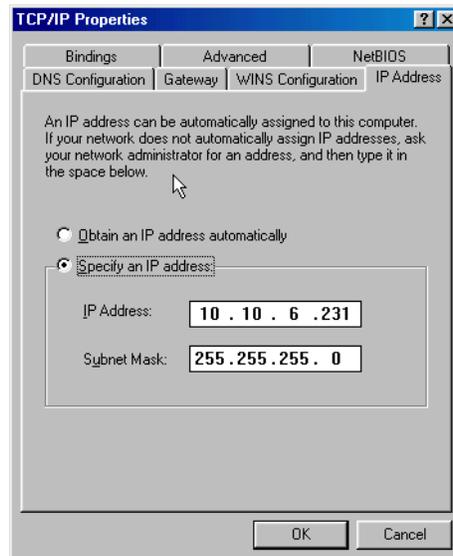
NOTE: Your Ethernet hardware may have a different name.



3. Select the **IP Address** tab, check **Specify an IP Address**, and fill in the following fields

IP Address : 10.10.6.230

Netmask : 255.255.255.0



Click:

<OK> to exit properties

<OK> to exit network

<OK> to restart computer

Once the PC is set up, we're ready to communicate. You can use Telnet, which comes with most Windows installations. Go to the **Start > Run** menu, and type in "telnet 10.10.6.230". This should connect you to the serial-to-Ethernet bridge. If you are using a serial DMM, select the function that will make it communicate, and you should see a stream of values in your Telnet window.

In order to capture data from the DMM, we will use the Tera Term terminal emulation program, which is available online at no charge from hp.vector.co.jp/authors/VA002416/teraterm.html. Download TeraTerm and follow the directions to unzip and install the software on your PC. Once you have collected the data, you can import it to a spreadsheet program such as Excel to process the data.

Further...

If you want to communicate with other serial-to-Ethernet devices, simply follow the same procedure, except that you would pick a different IP address for each device and connect the devices together through a bridge instead of using a crossover cable.

Occasionally, you may have a communication protocol for a device, which is neither ASCII or DMM22-812. The software in this kit is included in source form and allows you to modify and add conversion routines relatively easily.

You would craft a routine `convert_yourdevice()`, which will take an input buffer and length of serial characters received. The routine will convert the data and put it back in the supplied buffer. The return value, indicates the number of bytes that are in the buffer to be transmitted. Look at `convert_bin()` to see how a simple binary to hexadecimal protocol converter would be implemented. The routine `convert_dmm()` is a more complex example that illustrates looking for and converting packet-based communications.

Once you have a conversion routine for your device, you will have to give it a name and insert it into the menu structure so that it can be selected on the LCD/keypad module and used.

Well, happy communicating...

References

220-0812 Digital Multimeter Protocol Reference Manual

Bill of Material

Radio Shack 42 range DMM Cat. No. 22-812

Z-World, Inc.

2900 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-3737
Fax: (530) 757-3792

www.zworld.com

Rabbit Semiconductor

2932 Spafford Street
Davis, California 95616-6800
USA

Telephone: (530) 757-8400
Fax: (530) 757-8402

www.rabbitsemiconductor.com