



TCP/IP Communication for Real-Time and Embedded Systems

By Sam Mirsky

Many embedded systems in industrial automation, machinery, robotics, and other industries communicate data via Ethernet or via a real-time Fieldbus interface such as PROFIBUS or EtherCAT. Often, the embedded system is part of a network that includes sensors or other computers that provide data via TCP/IP. TCP/IP does not lend itself to real-time operation because it is a connection-based protocol that uses a client/server model in which data is retrieved via a series of requests and responses.

Using a thermal model of a house as an example, this article describes a method for communicating with other devices via TCP/IP. The method is based on using inexpensive TCP/IP-to-serial-conversion hardware. The real-time system is xPC Target™, a bootable kernel that enables real-time execution of Simulink® models on a separate PC. The target computer is an x86 processor Intel®/AMD® standard PC that can use commercial off-the-shelf input/output interface cards.

The connectivity approach described here can be used with any real-time or embedded system that supports serial communication.

Selecting a Converter

The external hardware, or converter, is the key component in the communication setup. Many types of converter are available, each offering different features and capabilities.

Select the converter that best matches your application and communication setup, keeping in mind three key features:

- Number of connections—some converters support single connections, while others provide as many as 32 multiple connections with multiple serial ports.
- Baud rate—we have seen converters with baud rates as high as 921kbaud. Converters with lower baud rates are less expensive, but their performance is slower.
- Connectivity protocol—the converter must support the protocol used by your network, either TTL serial, RS-232, RS-422, or RS-485.

For this example, we use the GW212 from Neteon.net (Figure 1). The GW212 is one of several models that would work for an application such as this one. About the size of a deck of cards, it has a RJ-45 jack on one side and a DB-9 on the other.

Products Used

- MATLAB®
- Simulink®
- xPC Target™

The GW212 supports one connection at a time, and can act as a TCP server and accept connections, or as a TCP client and request connections. It supports the RS-232 serial protocol at rates of 1200 bps up to 230 Kbps. It has status lights for Run (internal firmware), LAN activity, and COM activity.



Figure 1. GW212 TCP/IP-to-serial converter.

Configuring the Converter

Configuring the converter involves setting serial specs and TCP/IP parameters, as well as standard serial settings such as speed, parity, and stop bits. Most converters can be configured using DOS window or Telnet application, a Web browser, or the configuration software provided.

We configure the GW212 using the Telnet command. Since we will be acquiring temperature data provided by the National Oceanic and Atmospheric Administration (NOAA), we select the TCP/IP address of our Web site, which retrieves data from the NOAA Web site via a server-side script, and then set the converter to client mode.

Testing the Converter

A best practice is to test the converter in a non-real-time system before implementing commands in a real-time system.

To test the GW212, we connect it to the Internet and to the serial port on a desktop computer using Microsoft® HyperTerminal, a program for connecting to other computers, Telnet sites, online services, and host computers. We send the http commands GET and HOST. Almost immediately, the relevant data is retrieved (Figure 2).

Now that the converter is working, the final task is to parse the data returned from the server and use it in the thermal model. (In an actual application, this data might be sensor data that will be used in an algorithm.)

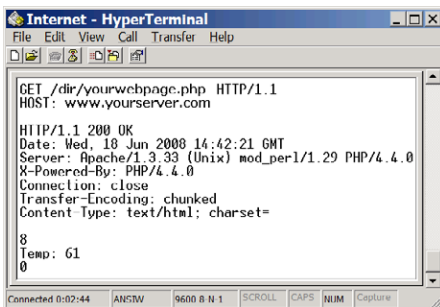


Figure 2. Testing a TCP/IP message within Windows XP. The response seen in the HyperTerminal window is the 'raw' response from the server.

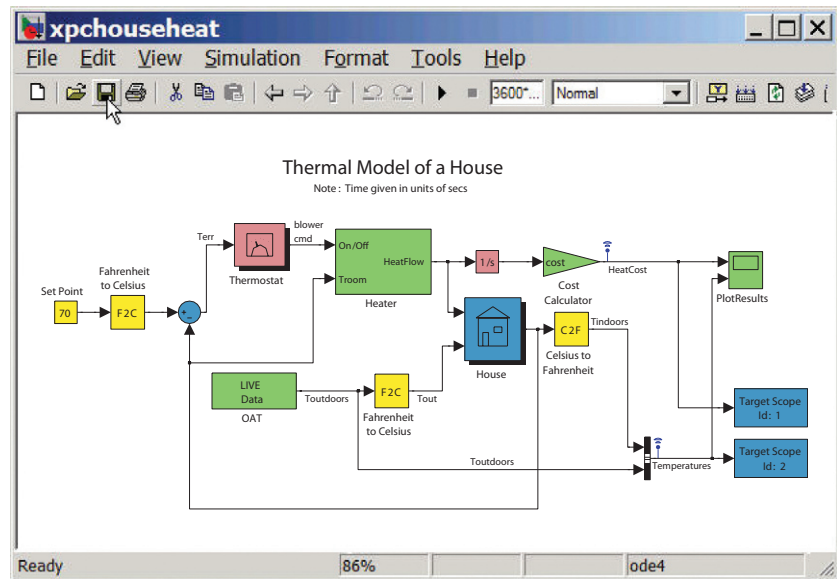


Figure 3. Thermal model of a house.

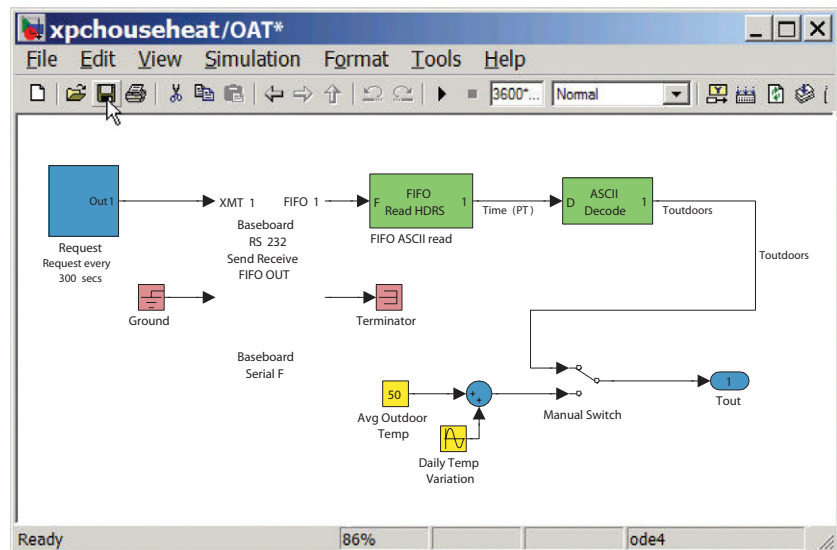


Figure 4. Serial communications with xPC Target.

Using TCP/IP Data in a Real-Time Simulation

After connecting the converter to the serial port of the real-time machine, we run the thermal model (Figure 3) using live outside air temperature (OAT) data from NOAA. For efficiency, we create a server-side script that parses the NOAA data to simplify the real-time task. xPC Target then requests the

data from the script on our server. The real-time model retrieves the data of interest by making the FIFO read header "Temp:" and the terminating string CR and LF (ASCII [13 10]). Then ASCII Decode reads the format string "Temp: %f\r\n" to obtain the OAT.

The outside air temperature is obtained from the Internet with the subsystem shown in Figure 4.

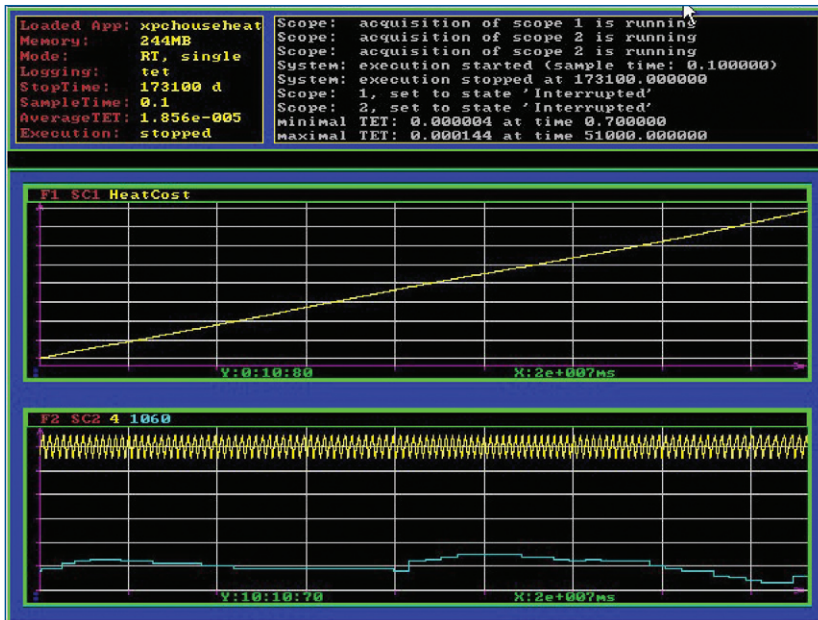


Figure 5. Bottom plot: The inside and outside temperature, in degrees Fahrenheit, for Worcester, MA from 9:00 a.m. on December 4, 2007 to 9:00 a.m. on December 6, 2007. Top plot: The estimated cost of heating the house (in U.S.dollars).

Simulation Results

In the two days during which temperature data was captured, the temperature never exceeded 25 degrees Fahrenheit (3.9 degrees Celsius). As the simulation demonstrates (Figure 5), heating the house with electricity under these conditions would be prohibitive—costing more than \$40 a day, or \$1200 a month.

Using the Converter in a Real-World Application

Because serial communication has a slower data rate than Ethernet communications, the connectivity solution described in this article might not be appropriate for transferring large amounts of data. It does, however, work well for basic tasks, such as communicating command or status data from sensors or devices on a TCP/IP network.

It can be used with any real-time device that supports serial but not TCP/IP, such as Real-Time Windows Target™ or embedded processors (such as the TI C2000 and the MPC55x series), as long as the serial electrical interface is matched. It will also work for bi-directional communications on any TCP/IP network.

Acknowledgement

This idea was inspired by an article by Tom Igoe that describes a method for acquiring air quality data from the Internet using an Arduino microcontroller and a Lantronics TCP/IP to TTL serial converter. The microcontroller then drives an analog meter with pulse-width modulated (PWM) output. ■

Resources

VISIT

www.mathworks.com

TECHNICAL SUPPORT

www.mathworks.com/support

ONLINE USER COMMUNITY

www.mathworks.com/matlabcentral

DEMOS

www.mathworks.com/demos

TRAINING SERVICES

www.mathworks.com/training

THIRD-PARTY PRODUCTS AND SERVICES

www.mathworks.com/connections

Worldwide CONTACTS

www.mathworks.com/contact

E-MAIL

info@mathworks.com

© 2008 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

91580v00 06/08

For More Information

- Tom Igoe "Net Data Meter." *Make*, volume 11, pp. 133-137.
- Webinar: Real-Time Execution Using Simulink and xPC Target
www.mathworks.com/tcpip_wbnr