CONTROL YOUR WORLD

Build a Wireless Weather System: Part 1

When I first started this series, I knew I wanted to build a wireless weather station. In no way could I have imagined how well the overall system would function, nor could I have envisioned its versatility.

In this article, it all comes together.

If you have already built a PC or microcontroller-based weather station connected to a 1Wire network, you may be asking yourself why you should go to all the effort and expense of converting it to wireless.

The wired 1Wire network has five main disadvantages:

1) Your system is tethered and you have to run a cable to the location where you want to place the display. This can be problematic and time consuming.

2) You are limited to the length of the cable and the number of items you can reliably place on the cable.

3) If you want to add additional sensors to various locations, this can prove almost impossible if you don't have access to attic or basement areas.

4) With a wired system, you generally only have a single display system.

5) If lightning strikes your weather pole, you run the chance of blowing up your computer or display system.

With a wireless system, all these problems will be solved. The only cable that needs to be run is for a power source. This is simply two wires and in many cases can be a very short run. You can even power your weather pole with a set of solar cells and a rechargeable battery.

The heart of our wireless system is a Zigbee mesh network. You have probably read about mesh networks, but this is your chance to actually build one and put it to work. Once the network is built, you will be able to add sensors or displays to any location in your home.

In addition, we are going to add extra features like the ability to create as many display systems as we like without having to change the network in any way. For instance, I can add a PC interface to the system by simply adding a Zigbee unit to an RS-232 card and I can start collecting data. At the same time, I can build a small LCD display for my desktop. I can add a large LED sign display to the front entrance of my lab

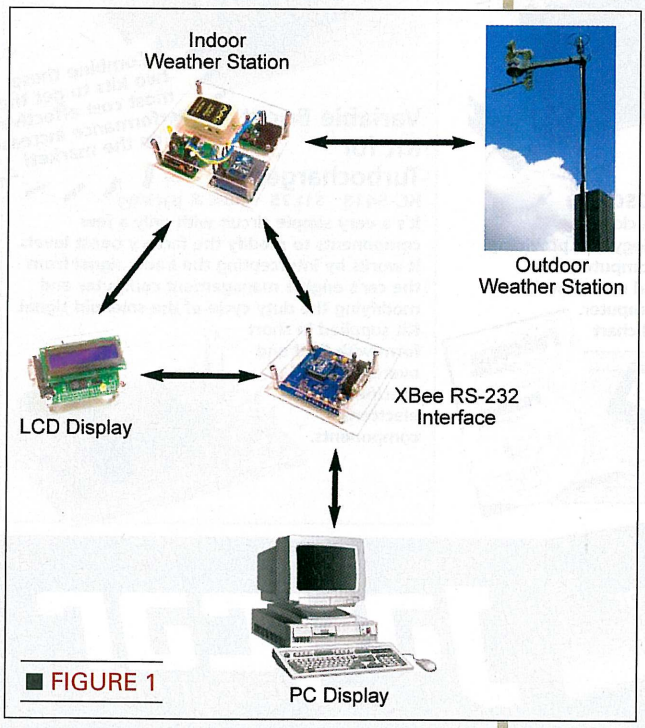


FIGURE 1

NOTE:
The parts list for this project will be included in Part 2 next month.

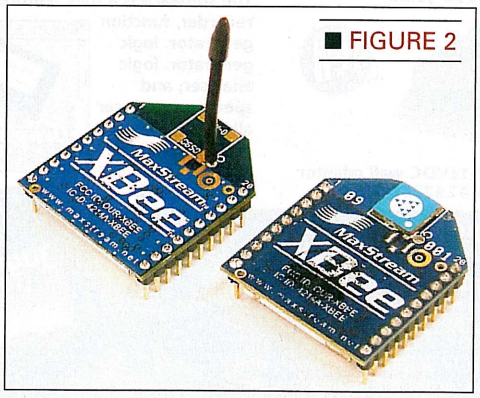


FIGURE 2

area or a voice that announces various weather or alarm data.

I tried several Zigbee modules and selected the Maxstream (**www.maxstream.net**) XBee modules shown in Figure 2. I recommend the whip antenna module as you get a bit more range over the small chip-based antenna module. The XBee whip antenna module will cost you about $19 each and you will need at least three to build your weather station network. Because each module's firmware must be loaded with special mesh network firmware, you will need to get a starter kit (Maxstream #XB24-DKS) which consists of the following:

- RS-232 development board
- USB development board
- Two XBee OEM RF whip antenna modules
- Various cables and connectors
- AC adapter
- Configuration software

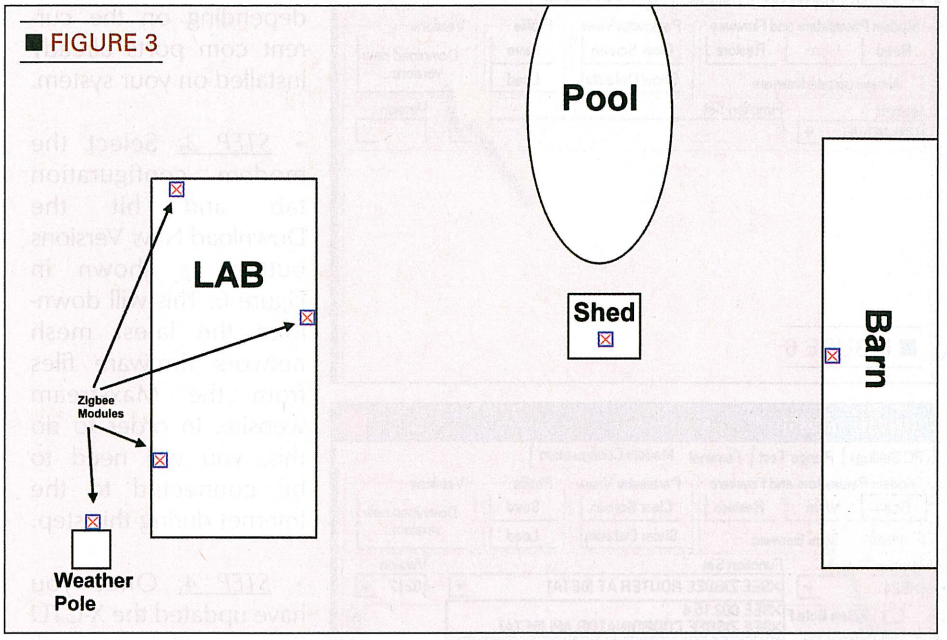
With the purchase of this starter kit, you will need at least one additional RF module; more if you want to add additional sensor networks and display systems.

Before I continue, it's important that I point out that Maxstream has just released a set of Series 2 modules and development kits. The Series 2 modules are not currently compatible with the existing XBee modules and will not work with the instructions presented here. I can assure you, however, that the current mesh network firmware for the existing XBee modules works just fine. My wireless weather station has been up and running for months now and has handled all sorts of power outages and other interruptions without a single glitch.

Just What Do You Gain by Using a Mesh Network in Your Home Weather Station?

I had three Zigbee/sensor modules located in my work area but wanted to collect data from a nearby barn. The barn is out of range of my current network. By placing a module in a small shed located between the two buildings as shown in Figure 3, I was able to add the barn to my network. This also has the benefit of allowing me to monitor the pool temperature.

FIGURE 3



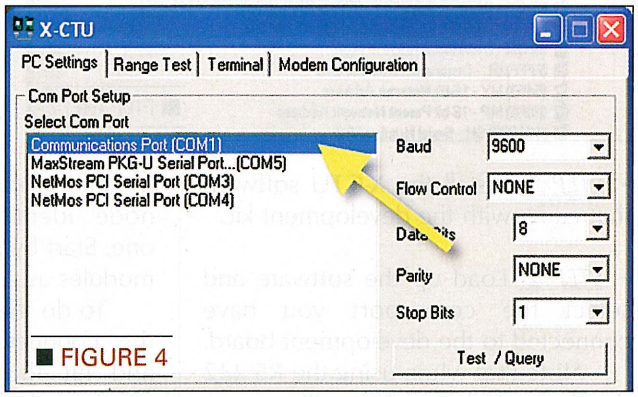


FIGURE 4

Our weather station mesh network is set up in such a way that all the modules broadcast their data. This allows any module placed in the network to be able to collect or transmit telemetry data.

Updating the Module's Firmware

When setting up your weather station mesh network, you will need to have one coordinator. All the other modules are set up as routers. The coordinator needs to have special mesh network coordinator firmware loaded. All the other modules need to have special mesh network router firmware loaded. To load the proper firmware into the modules, you need to follow these steps:

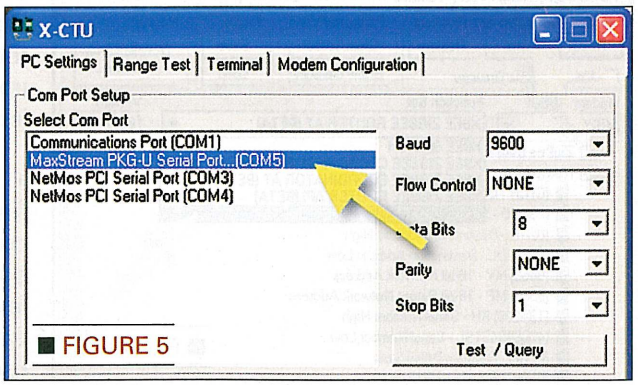


FIGURE 5

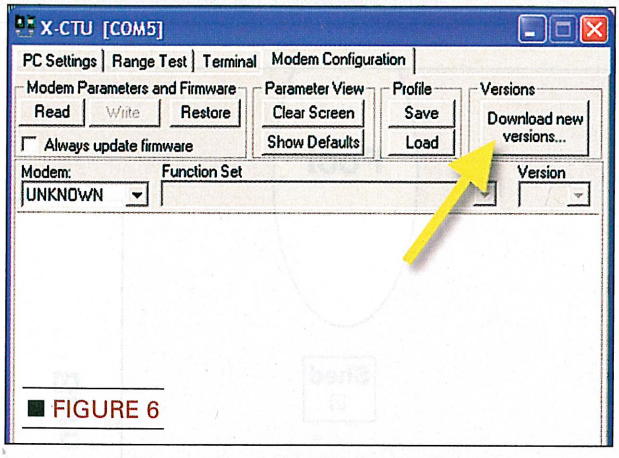


FIGURE 6

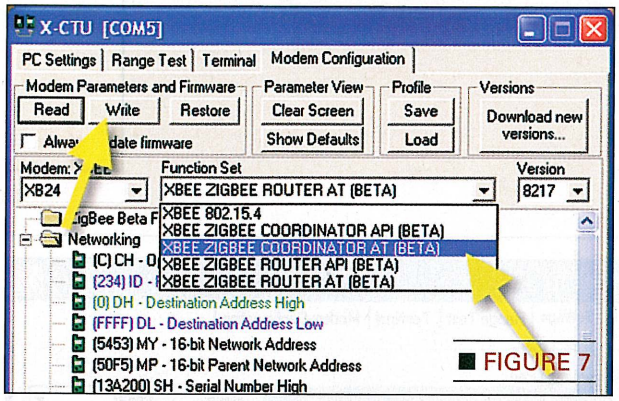


FIGURE 7

• *STEP 1:* Install the X-CTU software that came with the development kit.

• *STEP 2:* Load up the software and select the com port you have connected to the development board.

Note that when using the RS-232 development board, you will use one of your existing com ports as shown in Figure 4. The USB development board will create a new com port and will be labeled as shown in Figure 5. The actual com port assigned to the USB interface will vary depending on the current com ports already installed on your system.

• *STEP 3:* Select the modem configuration tab and hit the Download New Versions button as shown in Figure 6. This will download the latest mesh network firmware files from the Maxstream website. In order to do this, you will need to be connected to the Internet during this step.

• *STEP 4:* Once you have updated the X-CTU firmware files, you will need to update the firmware on each module. I also suggest you use a small piece of tape and mark each module so you can keep track of the node identifiers you assign each one. Start by programming one of the modules as a coordinator.

To do this, insert the module into the connected development board and hit the Read button on the configuration tab. This will load the current settings for the installed module. Note that the modules are shipped with the baud rate set to 9600. In the Functions Set field, select the XBEE ZIGBEE COORDINATOR AT (BETA) setting, and then hit the Write button as shown in Figure 7. This will cause the X-CTU software to upload the new firmware into the module.

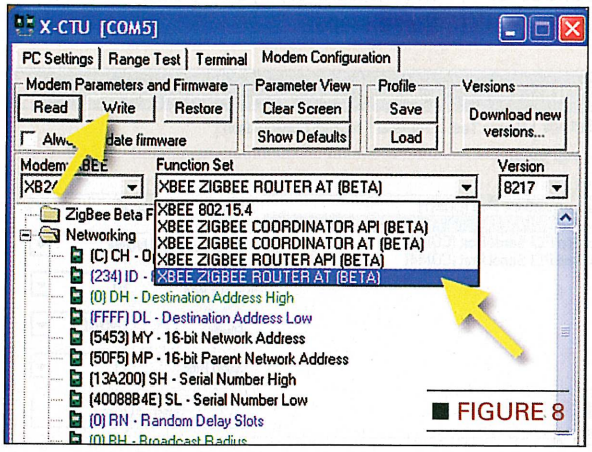


FIGURE 8

• *STEP 5:* Once the firmware update is complete, hit the Read button. You need to set a few of the parameters. You can do this manually or use the XBmeshcoord.pro file included in the available downloads (listed later).

The fields that are changed include the following:

- PAN ID = 234
- Destination Address Low = FFFF
- Node Identifier = COORDINATOR
- Packetization Timeout = 25

Once the values are changed, hit the Write button. Make sure you label the module so you can keep track.

• *STEP 6:* Now you need to load the router firmware into all the remaining modules. As before, insert the module into the development board and hit the Read button. I found that after changing the module, the X-CTU software would no longer respond. If this happens, exit the program and restart it. For the routers in the Functions Set field, select the XBEE ZIGBEE ROUTER AT (BETA) setting, and then hit the Write button as shown in Figure 8.

• *STEP 7:* Once the firmware upload is complete, hit the Read button. Make the following changes manually or use the included file called XBmeshrouter.pro.

The fields that need to be changed include the following:

- PAN ID = 234
- Destination Address Low = FFFF
- Node Identifier = R1
- Packetization Timeout = 25

Note that each router will need to have a different Node Identifier. I used R1-Rn in my network. It doesn't really matter what you use, as long as they are different. Write the changes and repeat for each router module.

You now have all your modules configured for your mesh network and are ready to build your wireless weather station.

Constructing the Network

For the basic network, there

are three components that I will call satellites:

- PC Interface Satellite
- Indoor Weather Satellite
- Outdoor Weather Satellite

PC Interface Satellite

Since the PC interface satellite is nothing more than a USB or RS-232 development board connected to the PC, we already have one of our satellites. The USB development board shown in Figure 9a is a great board for the PC interface satellite. Since it's powered directly by the USB port, it's simple to connect. The only problem with using this board is that when the PC is powered down, we lose the ability for this satellite to act as a router. If this is a problem, you can always plug it into a powered USB hub. If you decide to use the RS-232 development board shown in Figure 9b, you will need to connect the included AC adapter.

I recommend that you mount the development boards between two pieces of plastic as shown in Figures 9a and 9b. This will protect your board from shorts and will allow you to stand the board on end. Figure 10 shows my PC interface satellite sitting on one of my shelves.

Indoor Weather Satellite

You will need a couple of sensors indoors as part of your weather station. As a minimum, I recommend a barometric pressure gauge and a temperature sensor. I also added a humidity gauge to mine. Since we need a coordinator module on our network, the indoor satellite is the perfect candidate. Its job is to set up the network routing for all the other modules (routers) on the network.

The whole unit shown in Figure 11 is controlled by a DiosPro 28. The DiosPro can talk directly to one or more 1Wire networks, as well as to various other sensors. Since I am using a Hobby Boards pressure gauge, a 14 VAC adapter is needed to power this satellite. As an option, you may also use the SparkFun pressure sensor that

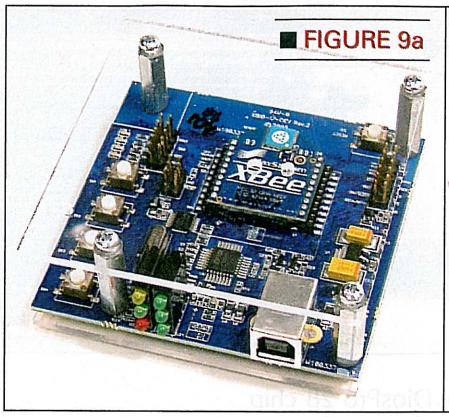


FIGURE 9a

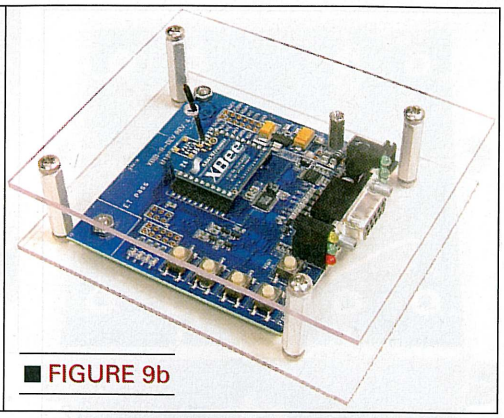


FIGURE 9b

was featured in the July issue of this series.

The components needed for this satellite are:

- DiosPro 28 chip
- Dios Carrier 1 kit (Note: This is built with the headers facing up.)
- One amp regulator kit
- XBee module
- XBee interface board
- Hobby Boards or SparkFun pressure sensor
- Humidity gauge (optional)
- AC adapter
- PlexiGlass and standoffs
- Double sided foam tape
- SchmartBoard jumpers

Notice that I listed an XBee interface board. This is a board that you will need to build. It will provide the 5 to 3.3 volt converter needed to connect the microcontroller. On the Kronos Robotics website, there are four application notes showing how to build various interface boards:

www.kronosrobotics.com/Projects/MaxStreamInterface1.shtml

www.kronosrobotics.com/Projects/MaxStreamInterface2.shtml

www.kronosrobotics.com/Projects/MaxStreamInterface3.shtml

www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml

In my opinion, the type 4 interface is the smallest and easiest to use, but types 2 and 3 will work just as well. Next month in Part 2, we will go step-by-step in the overall assembly

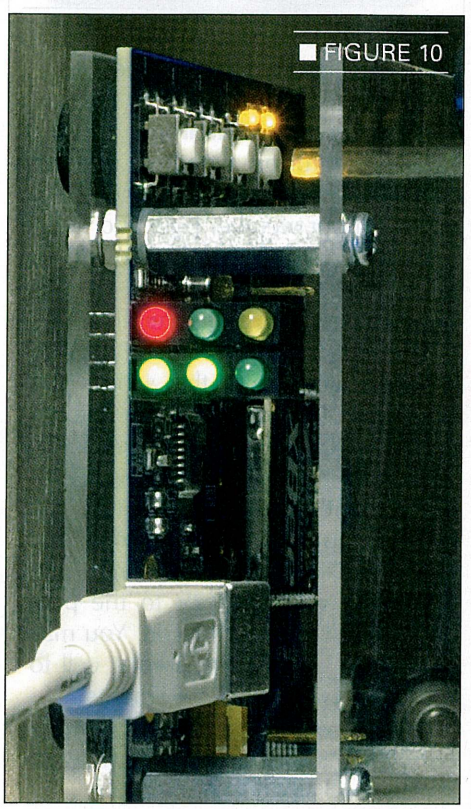


FIGURE 10

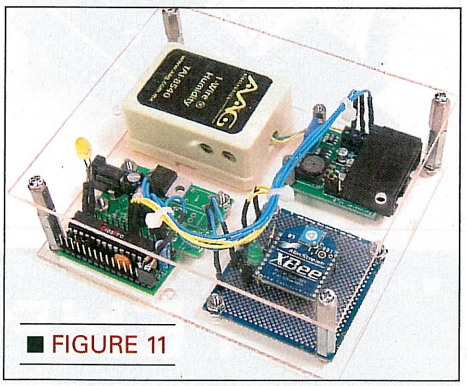


FIGURE 11

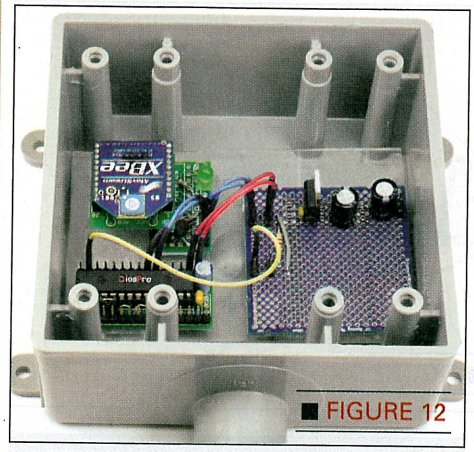


FIGURE 12

and testing process of this satellite.

Outdoor Weather Satellite

The outdoor weather satellite collects all of the data from the instruments on our weather pole and transmits the data to our mesh network. All the components that make up this satellite must fit in a weather-tight box like the one shown in Figure 12. If you followed along with this series, you saw how I built a weather station using various 1-Wire components. We will utilize this interface as well with our outdoor weather satellite. The main advantage that we gain is that once the DiosPro is programmed, you need only supply 7-12V to the pole and the rest is done for you. You may also use a battery and a solar cell to remove even these requirements.

You will need the following components for the outdoor weather satellite:

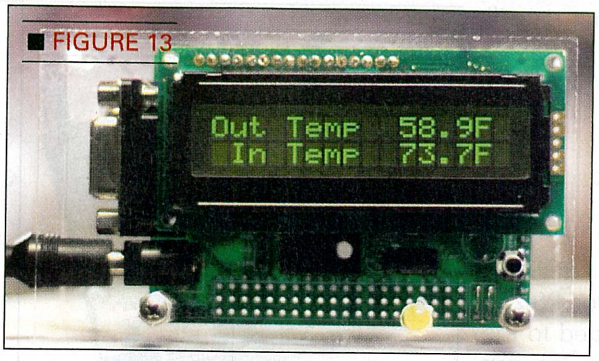


FIGURE 13

- DiosPro 28 chip
- Dios Carrier 1 kit (Note: This is built with the headers facing up.)
- Xbee module
- XBee interface board
- SchmartBoard jumpers
- SchmartBoard .1" prototype board
- 5V regulator chip
- Two 100 µF capacitors
- .1 µF capacitor
- 1K resistor

Since space is at a premium with the outdoor satellite, you need to build the Type 4 XBee interface board. You can find a complete application note explaining the process at **www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml**.

I also used a SmartBoard prototype board to build a one amp regulator, as well as the 1-Wire bus interface. There is plenty of room on this board for any other circuitry you might want to add to this satellite. For instance, you may want to use the built-in 10-bit A-to-D (analog-to-digital) ports on the DiosPro to create some sort of moisture meter.

In Part 3, I will take you step-by-step in building this satellite.

Satellite Display Units

One of the advantages of building a wireless weather station is that you can tap into any of the data being transmitted. I have created a very simple protocol and various DiosPro and PC routines to parse the network data for just about any display system you might want to include.

The LED display shown in Figures 13 and 14 is a Dios LCD board with a XBee module mounted on the bottom, using one of the various interfaces. This particular display uses a single button to toggle through the many different display items. It even counts lightning strikes collected by the outdoor satellite and keeps track of rain fall totals.

The display in Figure 15 is the same LCD carrier board, but it has a graphic LCD attached. In this particular display, I collect the data given by the indoor satellite and display a 12 hour forecast.

Figure 16 shows a BetaBrite LED sign connected to a Dios Carrier board that scrolls real-time weather data across the screen.

The beauty of these displays is

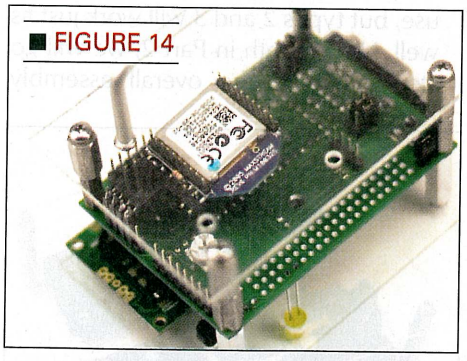


FIGURE 14

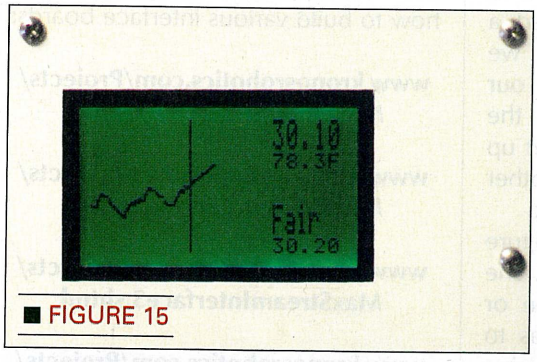


FIGURE 15

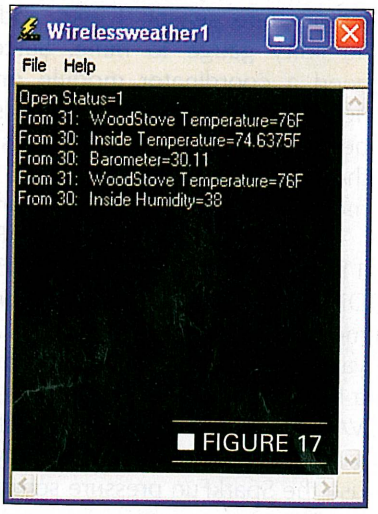


FIGURE 17

FIGURE 16

that they are not tethered to the rest of the system and can be placed anywhere in the house. You can even build a portable display that runs off of a battery.

For the PC interface, I have created several Zeus routines for parsing the network data using the USB or RS-232 development boards from Maxstream. Figure 17 shows one of the very simple display applications I use for displaying various pieces of data. You can also use some of my gauge routines from some of the previous articles to plot the data. This particular display will be used next month in Part 2 as we test our indoor satellite.

Final Thoughts

This portion of the Control Your World series is probably one of the most complicated projects to-date. However, I plan on taking you step-by-step through the various sections as we proceed. As we move forward, you will be able to take the knowledge gained from these and the previous weather articles to further enhance your weather station and home automation system. If you plan on building this system, I urge you to purchase the Maxstream starter kit and an extra XBee module or two. You should go ahead and set them up for a mesh network. Since you get two development boards, you can open two instances of the X-CTU software and run some simple tests.

What's Next

Next month, we will build the indoor weather satellite. Using one of the Maxstream development boards, we will test and display the results from this system. I will also break down the protocol that I use for transmitting data over the network. By understanding this protocol, you should be able to build your own display systems using any language or controller.

Be sure to check for updates at **www.kronosrobotics.com/Projects/wirelessweather.shtml**. NV