Control Your World
"Build a Wireless Weather System"
Part 2

By Michael Simpson

In this installment of the wireless weather station I am going to show you how to build the Indoor Weather Satellite. Once assembled, we will run some tests with our PC interface Satellite as shown in the network diagram in Figure 1.

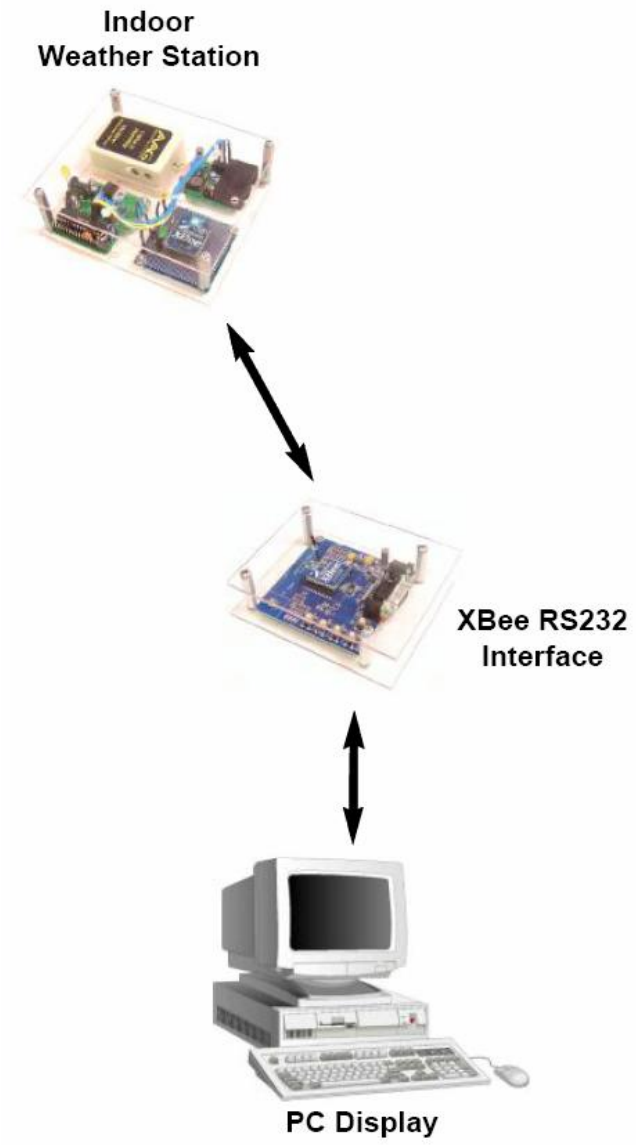


Figure 1

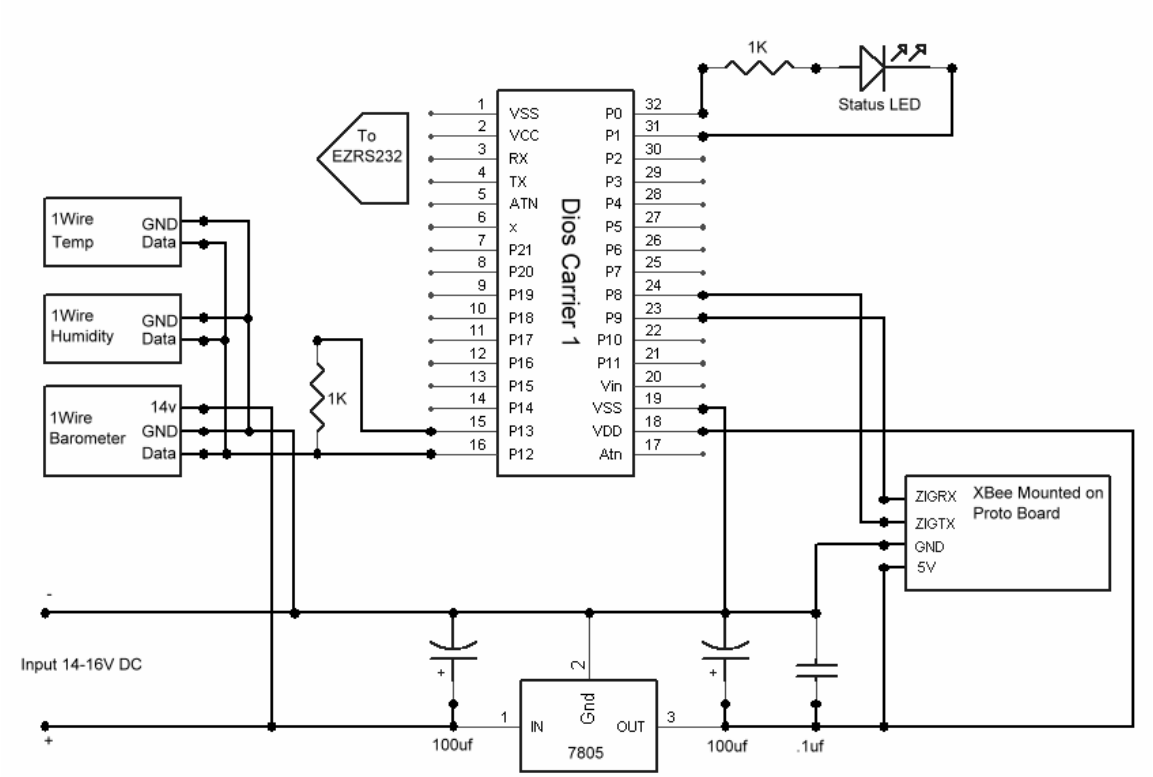
You should have the MaxStream mesh network firmware installed in each of the XBee modules you plan on using in your wireless network as outlined in Part 1 of this project. Using the Maxstream USB and RS232 development boards, you should have performed some basic tests to make sure your XBee modules are properly configured.

Building the Indoor Weather Satellite

The indoor satellite has a couple of functions to perform. Since the XBee module has been configured as a coordinator, its main job is to give all the other satellites a place on the network as they are added. It also serves as our main indoor data collection point. Here we will collect barometric pressure, indoor humidity, and indoor temperature. I will be using the AAG humidity sensor, HobbyBoards pressure sensor, and a DS1820 to collect temperature data. For the wireless side of things, I chose to use a standard XBee module and SparkFun's Break out board. A small regulator module and A DiosPro 28 from KronosRobotics will also be used to power and control everything. All these Items will be attached to two pieces of plastic and connected with SchmartBoard jumpers as shown in Figure 2.

As we build this satellite you can refer to Schematic 1 for detailed connection information.

Figure 2



Schematic 1

Step 1

You will need two pieces of acrylic as shown in Figure 3. The size I used was 5" x 5". A little larger size won't hurt, but I would not cut them any smaller. I also drilled a 5/32" hole into each corner of one piece. I then used that piece to mark and drill the second. This will insure that the holes line up when you add the standoffs later.

As we proceed, refer to Figure 11 for actual layout of all the assembled components and sensors. I recommend you assemble all kits and dry fit and mark them before actual assembly.

Figure 3

Step 2

You need to assemble a Dios Carrier 1 board according to the included instructions, with the exception of the 2, 16-pin headers. For this project you need to install the headers on the top of the board as shown in Figure 4. Use two layers of double stick foam tape to attach the carrier to the base, and then install the Dios Pro chip as shown.

Note: If you plan on programming the DiosPro while the top piece of acrylic is in place, you need to place this module closer to the edge of the acrylic. This will allow you to plug in the EZRS232 board.

Figure 4

Step 3

Assemble the Kronos Robotics 5 volt regulator board and attach it to the base. To do this, place the regulator on the base and mark the location of the four holes. Drill 4, 5/32 holes on the marks then install #4 machine screws and nuts as shown in Figure 6. Next, slip the regulator over the screws and add 4 more as shown in Figure 5.

Figure 5

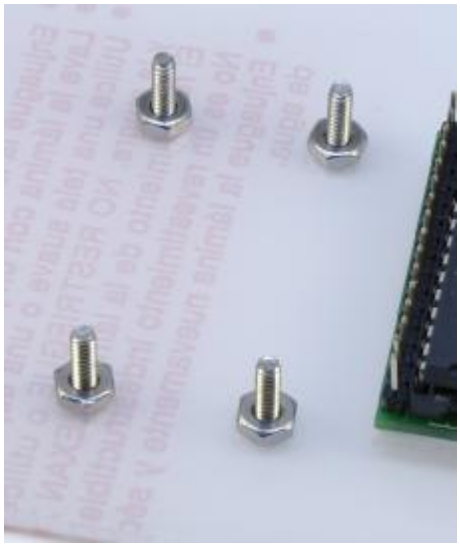


Figure 6

Step 4

You have three assembly choices for the wireless module. The purpose of this is to provide a 3.3v to 5v interface, as well as a way to attach and secure the XBee module. The actual assembly instructions for each can be found at:

SchmartBoard 2mm proto board (Figure 7a)
http://www.kronosrobotics.com/Projects/MaxStreamInterface2.shtml

SchmartBoard .1" protoboard and SparkFun breakout board (Figure 7b)
http://www.kronosrobotics.com/Projects/MaxStreamInterface3.shtml

KronosRobotics 3.3-5v interface and SparkFun breakout board (Figure 7c)
http://www.kronosrobotics.com/Projects/MaxStreamInterface4.shtml

I used the .1" proto board shown in Figure 7b, but all will work. The 2mm Schmartboard shown in Figure 7a will require that you attach a standard .1" header to the top of the 2mm headers in order to use the jumpers.

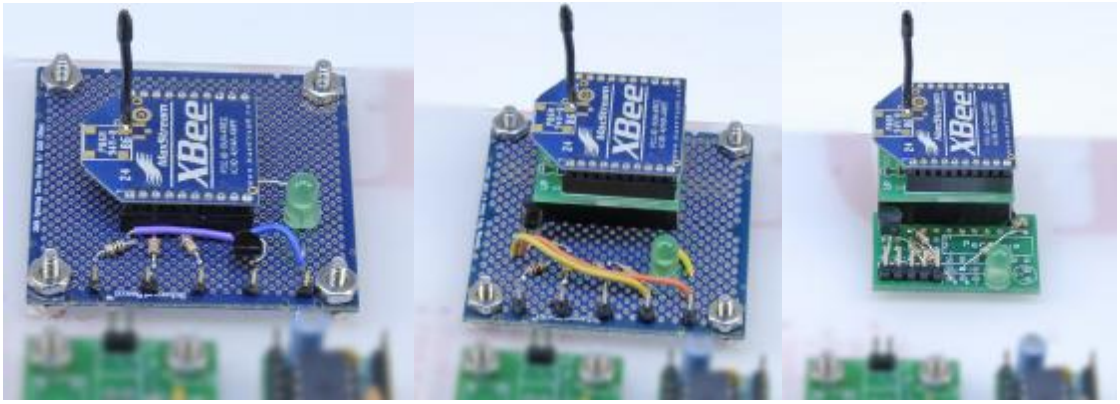


Figure 7a **Figure 7b** **Figure 7c**

Step 5

You will need to prep the HobbyBoards 1Wire Pressure sensor so that you can attach the jumpers later. To do this, break off a male header and remove the plastic from the pin. You can use straight or right angle headers for this as shown in Figure 8. Later, we will be inserting jumpers on these pins.

If you are going to connect an indoor humidity sensor, you will need to connect two 4" wires to the headers shown in Figure 8. These will route to the humidity sensor later when we install it. If you wish, you may use different colors for the wires.

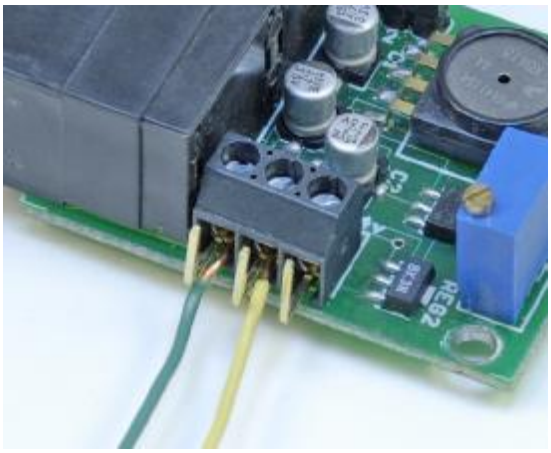


Figure 8

Step 6

The pressure sensor only has two mounting holes and the tolerance between the hole and the other components are quite close. I marked the holes and attached two, #4 machine screws and nuts to the base, then attached the pressure sensor to machine screws. Because of the tight tolerance, I used some rather thin #4 standoffs to attach the board as shown in Figure 9. You can also attach the sensor with two layers of double stick foam tape.

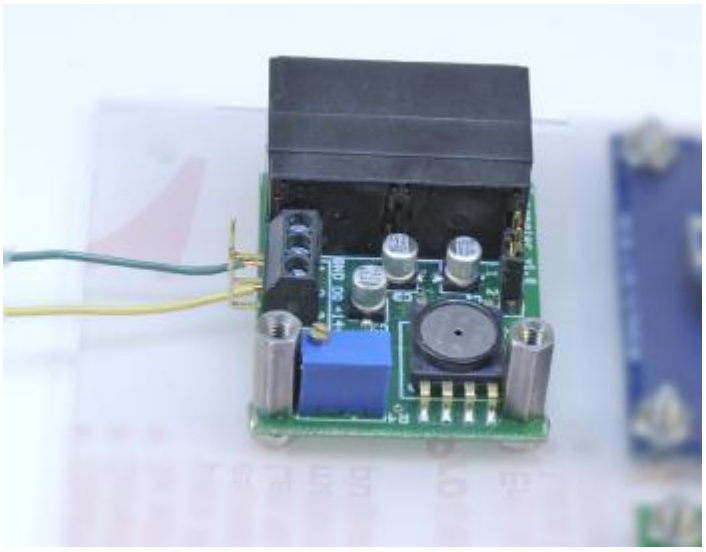


Figure 9

Step 7

If you are using the humidity sensor, you have two choices in mounting. You can use two #4 machine screws or double stick foam tape. To access the mounting holes you need to remove the PCB board first. This is held in place by the 4 small terminal screws.

Note that the HobbyBoards humidity sensor will work as well. Once mounted and connected, its operation is identical to that of the AAG sensor.

Next, you need to connect the 2 wires from the pressure sensor to the humidity sensor. The green wire (Gnd) is connected to the green terminal on the humidity sensor. The yellow wire (DQ) is connected to the yellow terminal on the humidity sensor as shown in Figure 10. If you are using the HobbyBoards sensor it is a bit easier. Connect the Gnd and DQ terminals on the pressure sensor to the Gnd and DQ terminals on the humidity sensor.

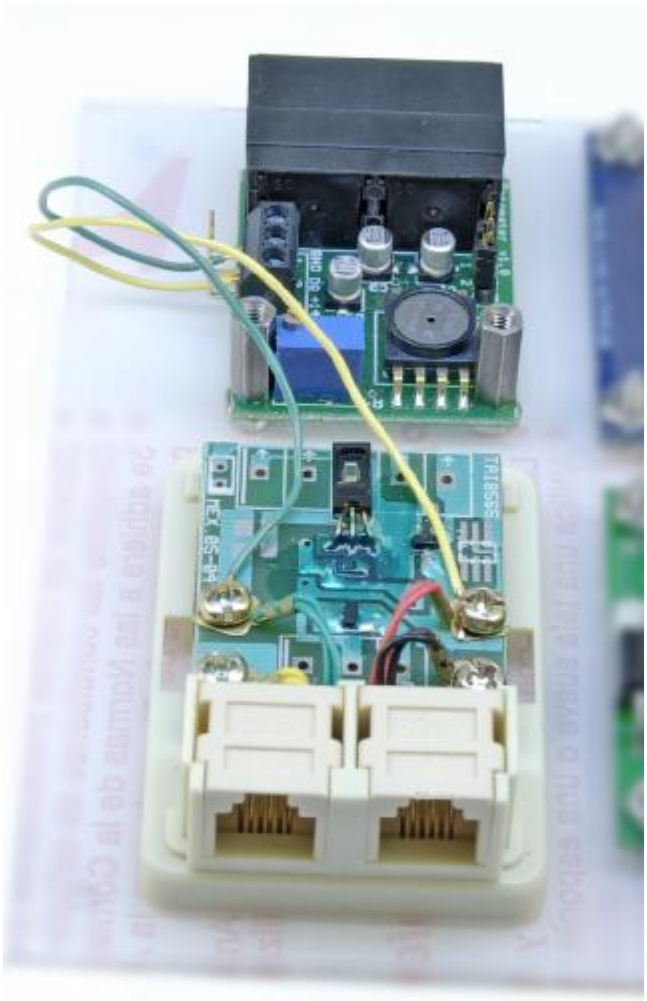


Figure 10

Step 8

All components for our indoor satellite are now in place as shown in Figure 11. Attach four 1-1/4" #4 standoffs to the bottom base as shown in Figure 11. You can use single standoffs or combinations. You may also use larger standoffs if you want more clearance. At this point, I also mounted 4 rubber feet on the bottom of my base as well. Just about any will do. I purchased these from my local home center.

Figure 11

Step 9

Using jumpers, connect the + and – terminals on the regulator to the Dios Carrier 1 board as shown in Figure 12. If you are not clear on the actual pins, refer to the included hookup sheets.

Next, connect the + and – terminals as shown in Figure 12 to the XBee module.

Figure 12

Step 10

Attach the VIN terminal on the regulator to the +14v terminal on the pressure sensor as shown in Figure 13. Then connect the Vss terminal on the regulator to the Gnd terminal on the pressure sensor as shown in Figure 13.

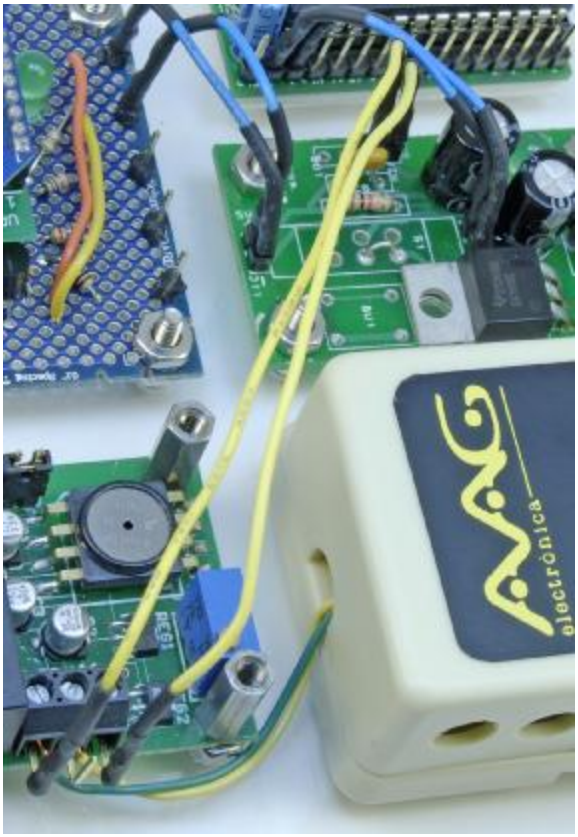


Figure 13

Step 11

Using a small 2-pin female header, attach a 1K resistor between the two pins. Then attach a single male header to one of the pins as shown in Figure 14. Slip this assembly over the carrier 1 header ports 12 and 13. Make sure the header pin that you added to the connector is on port 12. The resistor is used to hold the port 1Wire network high. Later in our software we will set port 13 to output and a high state. This will supply the voltage needed to the resistor, which will in turn supply the network.

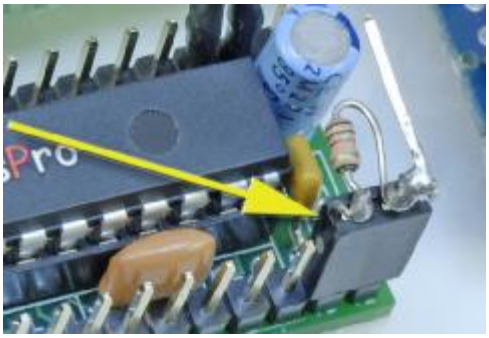


Figure 14

Step 12

Run a jumper from the pressure sensor DQ terminal to the port 12 header as shown in Figure 15.

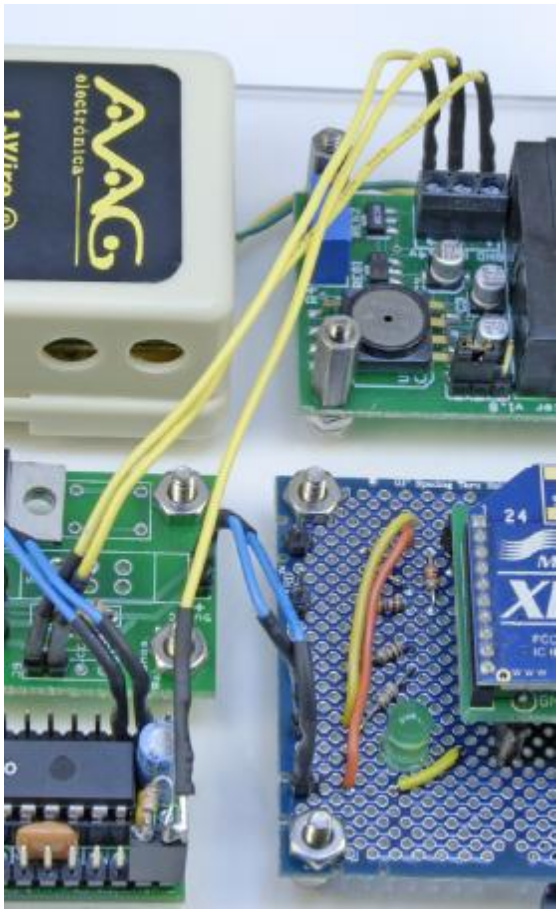


Figure 15

Step 13

The last two jumpers connect the transmit and receive leads on the XBee module to the DiosPro chip. Connect TX lead on the XBee module to port 8 on the Dios Carrier. Connect the RX lead on the XBee module to port 9 on the Dios Carrier.

Figure 16

Step 14

In order to perform tests and to supply a status LED for the satellite, we need to attach an LED to the DiosPro microcontroller. To create this LED, wire a standard LED in series with a 1K resistor to a 2-pin female header as shown in Figure 17. The location of the resistor does not matter. You may also want to add a small amount of heat shrink depending on the length of your LED leads. Attach the header to the Dios Carrier header ports 0 and 1. The flat side of the LED connects to port 0 as shown in Figure 18.

Figure 17

Figure 18

Testing the Indoor Satellite

You will need to use an AC adapter that supplies 14 volts in order to supply enough power to the pressure sensor. I used a 12 volt unregulated AC adapter for this. When you apply power to the 5v regulator, the LED on the regulator should light up. If it does not or is very dim, then you have a bad power connection somewhere. Disconnect the power immediately.

The DiosPro ships with a test program installed on the chip. When power is applied, the status LED should blink. This indicates that the chip is getting enough power.

In order to program the DiosPro chip you need an RS232 adapter to connect the chip to your PC. These are available from the Kronos Robotics website for under $10. The EZRS232 interface comes in kit form with a 5-pin male header. In order to connect the EZRS232 to our Carrier 1, we need to create a small adapter by connecting two 5-pin female headers together as shown in Figure 19.

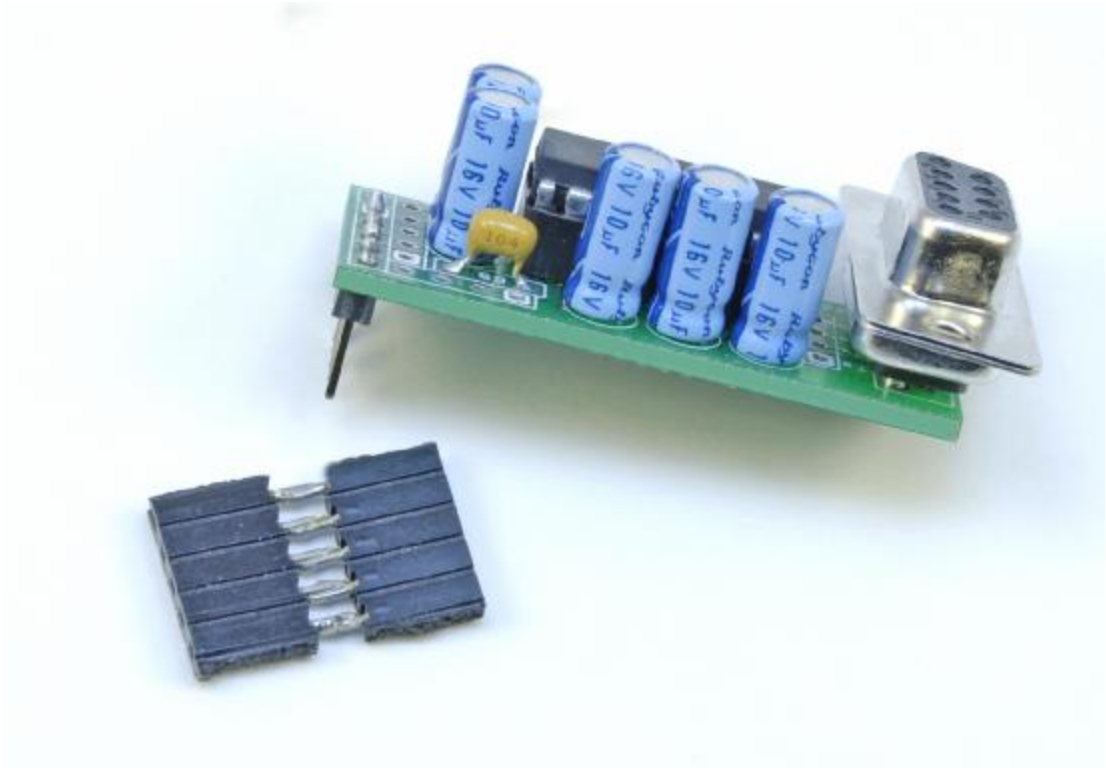


Figure 19

As an option, you may forgo the male header and attach a 5-pin female header in its place as shown in Figure 20. This is the preferred method of programming the Dios Carrier 1 when the headers are on the top of the board.

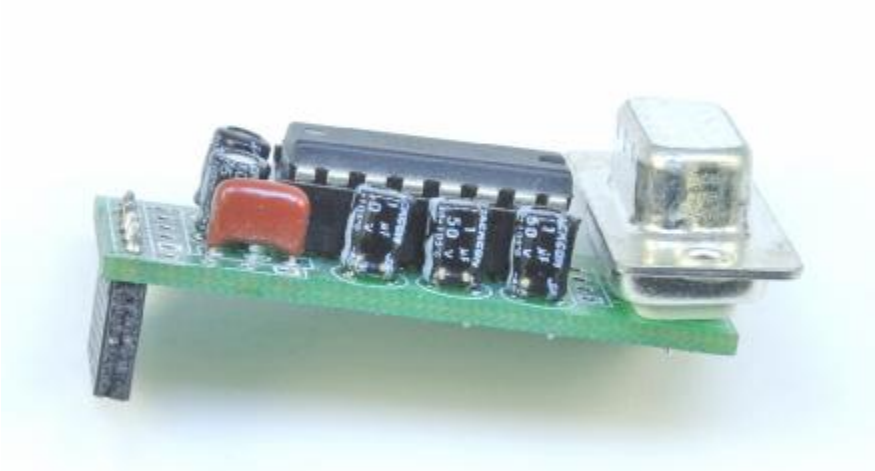


Figure 20

The EZRS232 interface board is connected to the carrier by slipping the female header over the programming leads. These are the first five leads on the right side of the carrier as shown in Figure 21. You can now connect the PC to the indoor satellite and apply power.

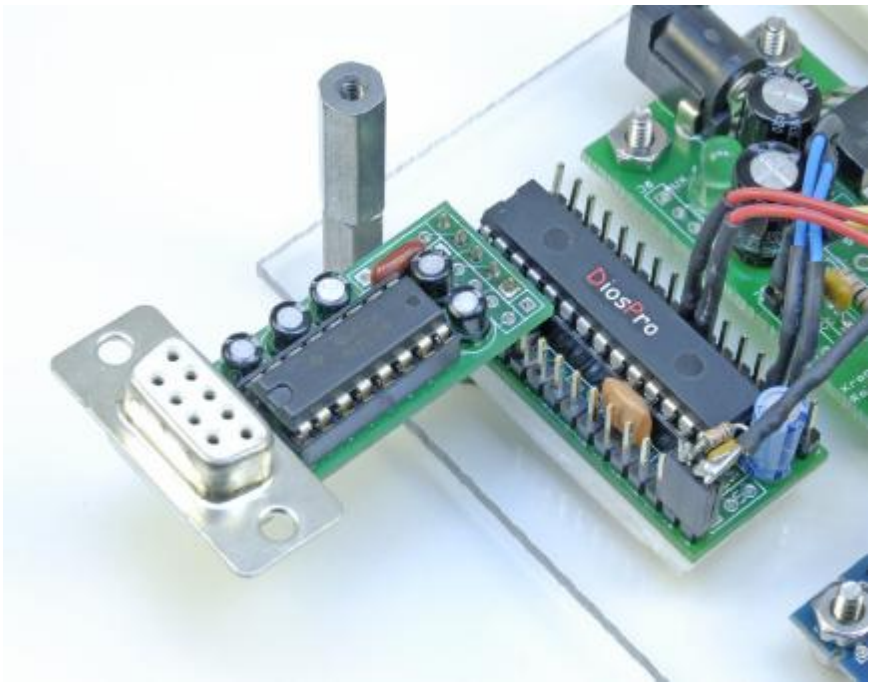


Figure 21

If you have not already done so, download and install the free Dios Compiler from the Kronos Robotics web site. You can pick up a copy at:
http://www.kronosrobotics.com/downloads/DiosSetup.exe

Once installed, start the program and load the code from Program 1 into the compiler and hit the program button. The compiler should compile the program then upload the code to your DiosPro chip. Once done, the status LED will blink and the word "test" will continue to print on the debug terminal of the program.

```
DiosPro
'LED Test Program for Indoor Weather
' Satellite module
func main()

 gconst STATLED 1
 output 0,STATLED
 low STATLED,0

Loop:
 print "Test"
 high STATLED
 pause 200
 low STATLED
 pause 200
 goto Loop

endfunc
```

Program 1

Back in March, I showed you how to build the 1Wire temperature sensor shown in figure 22. Information for this sensor is also available on the Kronos Robotics Website.

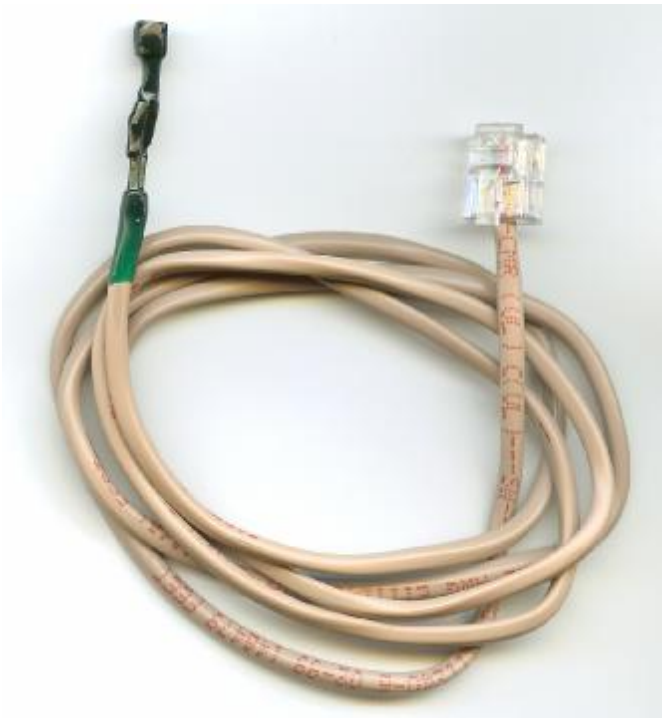


Figure 22

You will need one of these remote sensors connected to your indoor satellite. Once built, just plug the connector into one of the existing sensors mounted on the satellite.

Performing a Network Search

In order to read the 1Wire sensors, you will need to obtain the Rom serial numbers from each of the sensors on the network. Included with the downloads for this project is a program called Networksearch.txt. With the temperature sensor connected to your satellite, load and run the Networksearch.txt program. If your sensors are connected properly, the data from the three sensors should be displayed as shown in Figure 23.

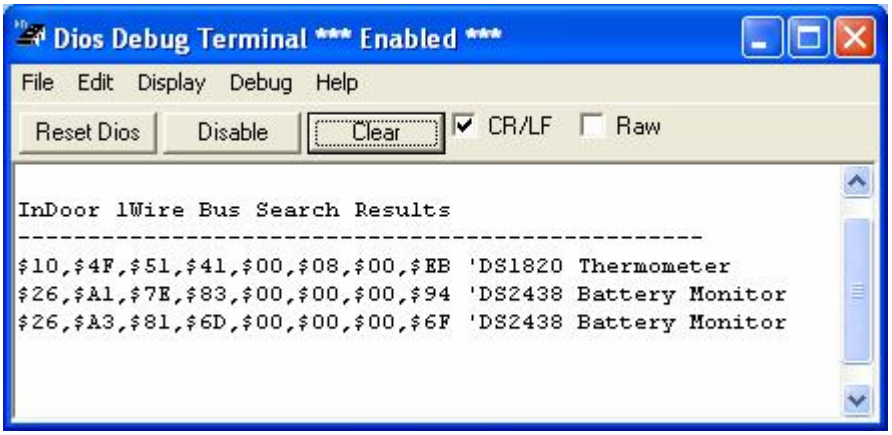


Figure 23

Unfortunately, both the humidity sensor and the pressure sensor use the same 1Wire interface chips so it is up to you to determine which Rom number is associated with which sensor. To do this, you can temporarily disconnect the humidity sensor and run the search program.

To test the sensors run the program called SensorTest.txt. This program will take readings from the temperature sensor, barometer sensor, and the humidity sensor and display them on the debug terminal. At the very beginning of the program are three entries that assign the Rom addresses foe the sensors. You will need to change these entries to match your own sensors.

The sensor data will also be sent over the network via the XBee module. If you connect your PC to one of your development boards and run the X-CTU software in terminal mode, you should be able to see the network data.

Final Thoughts

I wanted to get into the protocol I developed for this project but just did not have the space to do it this month. So it will have to wait until the end of the series, which is where it is needed anyway.

We used the HobbyBoard 1Wire pressure sensor in this project. It has a built-in thermometer, but due to the heat generated by the sensor it is unreliable. This also applies to the humidity sensor as well. On the other hand, you could have used the SparkFun pressure sensor that I presented last week in the barometer plotter. This sensor has both a reliable pressure sensor and a temperature gauge so the need for an external sensor is not required. The use of an indoor humidity sensor is really not required. I like to have one because I do a lot of work with wood, and I like to know what the humidity is in my lab/shop area.

What's Next

Next month we will build the Outdoor Weather Satellite. Again, we will be using a maxstream XBee module as well as a DiosPro 28 to interface with the various sensors.

Be sure to check for updates at:
http://www.kronosrobotics.com/Projects/wirelessweather.shtml

Parts

The following is a break down of the source for all the components needed for Parts 2 and 3 of this project.

MaxStream

Starter Kit #XB24-DKS
http://store.maxstream.net/index.cfm?fuseaction=product.display&Product_ID=1

Also available from Mouser at:
http://www.mouser.com/search/ProductDetail.aspx?R=XB24-DKSvirtualkey61440000virtualkey888-XB24-DKS

Or do a search for XB24-DKS

Hobby Boards

Barometer Module (B1-R1-A)
http://www.hobby-boards.com/catalog/product_info.php?cPath=22&products_id=36

Humidity Module (H3-R1-A)
http://www.hobby-boards.com/catalog/product_info.php?cPath=22&products_id=46

Spark Fun Electronics

SCP1000
http://www.sparkfun.com/commerce/product_info.php?products_id=8161

XBee Breakout Board (Used to build various interface boards)
http://www.sparkfun.com/commerce/product_info.php?products_id=8276

2mm connectors (You need 2 for each Breakout board)
http://www.sparkfun.com/commerce/product_info.php?products_id=8272

Kronos Robotics

DiosPro 28 chip
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16429

Dios Carrier 1
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16170

3.3v to 5v Interface Kit
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16537

40-Pin Male Header
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16290

T220 Heat Sink
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16250

1K resistors
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16178

Regulator Kit
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16304

5v Regulator Chip (Used in Outdoor Satellite)
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16208

100uf Capacitors
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16202

.1uf Capacitors
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16198

14V DC Adapter
http://www.kronosrobotics.com/xcart/customer/product.php?productid=16500

Free Dios Compiler (Includes 1-Wire libraries)
http://www.kronosrobotics.com/downloads/DiosSetup.exe

SchmartBoard

Prototyping board (.1”)
http://www.schmartboard.com/index.asp?a=11&id=24

Jumpers 5” Yellow
http://www.schmartboard.com/index.asp?a=11&id=42

Jumpers 3” Red
http://www.schmartboard.com/index.asp?a=11&id=41

Jameco Electronics

Standoffs
http://www.jameco.com/webapp/wcs/stores/servlet/CategoryDisplay?storeId=10001&catalogId=10001&langId=-1&categoryId=355050

Local Hardware or Home Center

Double sided Tape
PlexiGlass (Or equivalent)

Links

Hobby Boards
http://www.hobby-boards.com

Spark Fun Electronics
www.sparkfun.com

Kronos Robotics
http://www.kronosrobotics.com/xcart/customer/home.php

ShmartBoard
http://www.schmartboard.com/index.asp?a=11&page=a_products

Jameco Electronics
http://www.jameco.com