Setting Up a Spinneret Webserver
By Jeff Ledger

This instructional is designed to take those with limited networking knowledge through making their Spinneret accessible to the world via the Internet. I'll take you though the basics of setting up a Spinneret web server and configuring it easily and soon the world will be beating a path to your Propeller powered Spinneret.

Understanding IP addressing: (local/global)

All devices connecting to any Internet network identify themselves with a series of numbers called an IP address. (Internet Protocol Address) There are two different types of IP address which may be assigned to internet devices, "local" and "global." A global (WAN) IP address is assigned by an internet service provider and can be communicated with anywhere on the internet. Global IP addresses are unique and are assigned to only a single computer or device. A local (LAN) IP address is private address which is assigned by your router. Local IP addresses can only be used to communicate within a private or "local" network.

You can tell the difference between a "global" or "local" IP address by being able to identify the local IP addresses.

Local IP addresses:

10.0.0.0 = 10.255.255.255
172.16.0.0 = 172.31.255.255
192.168.0.0 = 192.168.255.255

Generally, home routers use local (LAN) addresses which begin with 192.168.

Typical Home Internet Configurations:

Typical home configurations generally fall into one of two categories:

1) A cable modem or DSL adapter connected directly to a single computer.

2) A cable modem or DSL adapter connected to a router, which shares the connected between multiple computers.

In configuration #1, the global IP address is usually assigned to the computer. The firewall on that computer protects the computer from outside access.

In configuration #2, the global IP address is assigned to the router. The router in turn, assigns local IP addresses to any connected computer or device.

An example of a router based configuration would look like this:

Ken's house has three computers and a router.
Ken's router is assigned a global IP address of 205.117.207.6 from his ISP.
All of Ken's computers are connected to the router.
The routers IP addresses:
Global: 205.117.207.6
Local: 192.168.0.1
Computer #1 IP address: 192.168.0.100
Computer #2 IP address: 192.168.0.101
Computer #3 IP address: 192.168.0.102

When Ken uses any of the computers for the Internet, the traffic to sent to the router which in turn is passes the information onto the Internet through its global IP address. The router is acting as the "gateway" for all connected computers regardless if they are physically plugged into the router, or are connected to it by wireless link.

Looking at configuration #1, where the Internet is connected directly to a single computer, you have probably deduced that a router will be required to connect your Spinneret to your network or the Internet. (Some cable modem or DSL adapters can actually fill this role, providing they have more than one network jack allowing for multiple computers or devices to be connected. Check with your service provider.)

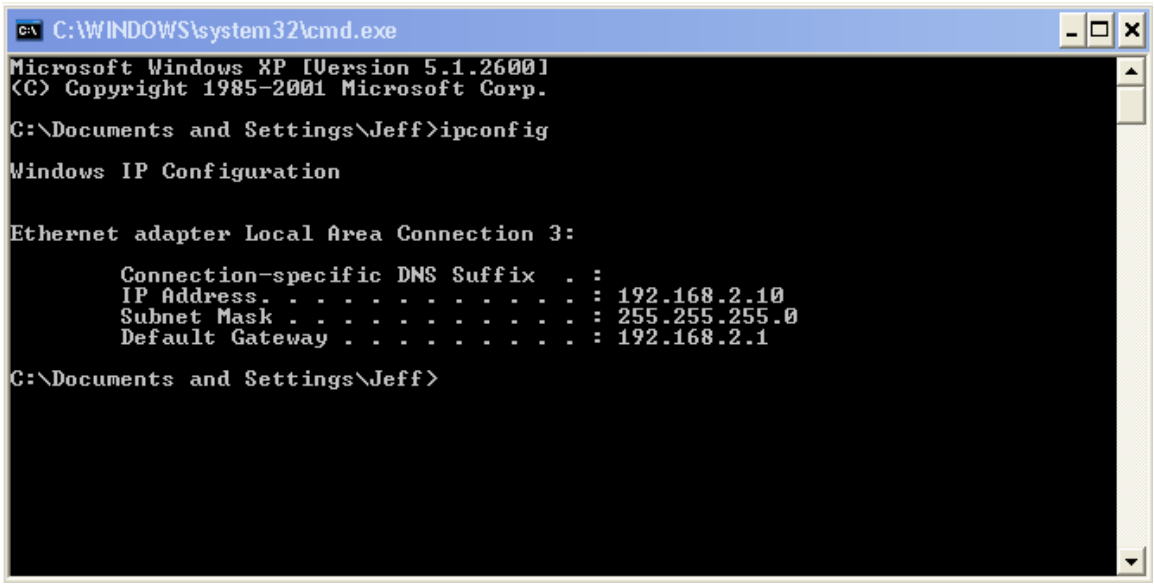
Looking back at our example, if Ken adds a Spinneret to his network, it would look something like this:

Ken's house has three computers and a router.
Ken's router is assigned a global IP address of 205.117.207.6 from his ISP.
All of Ken's computers are connected to the router.
The routers IP addresses:
Global: 205.117.207.6
Local: 192.168.0.1
Computer #1 IP address: 192.168.0.100
Computer #2 IP address: 192.168.0.101
Computer #3 IP address: 192.168.0.102
Spinneret IP address: 192.168.0.103

If the software which is running on your Spinneret is capable of auto configuring its IP address, then 192.168.0.103 would be assigned from the router. If you are configuring the IP address by hand, then it would be wise to assign it an address outside of the addresses being handed out by the router. Instead of 192.168.0.103, configure it to 192.168.0.200. This will insure that your router will never duplicate the address to another computer in the future creating an "IP address conflict". Also, by setting the address of the Spinneret by hand, you don't have to worry about it potentially changing in the future.

Determining your IP address information:

Windows users can easily determine their IP address configuration using the "Command Prompt" icon. Click on "Command Prompt" and type **ipconfig** and press enter. You should see a screen that looks something like this:



The IP address **192.168.2**.10 is the local number assigned to my computer. The default gateway **192.168.2**.1 is the local address assigned to my router.

Configuring the Spinneret to **192.168.2**.200 should allow me to access it on my local network through a web browser pointed at that address providing the Spinneret is running web server software.

There are five pieces of information required when configuring the Spinneret for Internet access. We already have three of them.

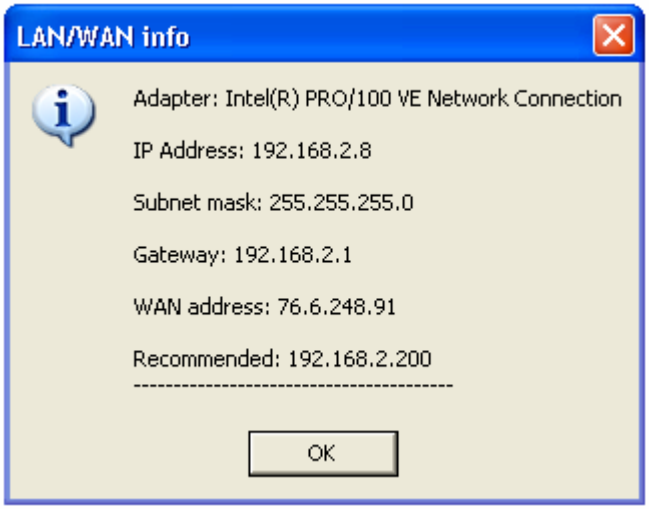
1) A local (LAN) IP address. In my case 192.168.2.200 is a good choice.
2) A subnet mask: Match what the computer is using. 255.255.255.0

3) The default gateway: Again match the computer. Mine is 192.168.2.1
4) The global (WAN) IP address assigned from the Internet provider.
5) The port which the Spinneret software is configured to answer to.

The global IP address can be found by going to http://www.whatismyip.com using any computer connected to your network. This address is the one which you would use to access your Spinneret from outside of your network.

Technically, your Spinneret web server will be accessible from two different IP addresses. Your LAN (local) address is how to access the Spinneret from your own computer, and computers on your "local" network. The WAN (internet) address is how computers on the Internet will have access.

Another tool which can assist in obtaining the information for configuration of your Spinneret was written by Ron Czapala (A Parallax Forums Member) called, **GetIP.vbs**, available from http://spinneret.propellerpowered.com/GetIP.vbs. It is a simple Windows script which retrieves the relevant information you need.



GetIP.vbs script

The IP Address is the one assigned to the workstation on which you are running the script. It provides us a point of reference. You'll need the **Subnet Mask**, **Gateway**, **WAN address**, and the **Recommended** address for your Spinneret.

All about port addresses:

Imagine that the Internet is a radio. The port addresses are the station numbers on the dial. On my radio, 99.3 is rock, while 104.7 is easy listening. Port addresses are the like the "station" numbers for the internet. They can range from 1 up into the thousands. Fortunately, you don't have to memorize them,

and only a handful of port numbers are used commonly for typical Internet access.

Here's a few of the common port addresses:

Web access:	port 80
FTP access:	port 21
Telnet access:	port 23
Email access:	ports 25 & 110

Since you are configuring the Spinneret to act as a web server, it would be easy to assume that you would use port 80. This will work fine on your local network. My own Spinneret would answer to http://192.168.2.200:80 from my web browser. Because most web browsers assume all web access is on port 80, I can even use http://192.168.2.200 and it works just fine. There is a catch.

Many Internet service providers do not allow these common ports incoming access from the outside world. This is done in the name of safety for typical Internet customers. My own experience is that DSL providers generally allow these ports to access, while cable providers do not. To overcome this limitation, we simply pick a higher port number and assign it to the Spinneret. Instead of using port 80, (http://192.168.2.200:80) assigning it port 5555 (http://192.168.2.200:5555) takes care of the issue.

Download and Install the Spinneret Web Server Code:

Spinneret Web Server Software has been a collaborative venture by many on the Parallax forums. At the time of this writing, I recommend the web server Spin available at agaverobotics.com.

Link to the current version: http://www.agaverobotics.com/spinneret/source/

Download and extract all of the Spin source files. Locate HTTPserver.spin.

```
DAT
  mac                   byte    $00, $08, $DC, $16, $EF, $22
  subnet                byte    255, 255 ,255, 0
  ip                    byte    192, 168, 2, 200
  gateway               byte    192, 168, 2, 1
  port                  word    80
  uport                 word    81
  remoteIp              byte    65, 98, 8, 151
  remotePort            word    80
```

In the DAT section (around line 14) adjust the **subnet**, **ip**, **gateway**, **port**, and **remotePort** settings to those you have identified in the previous section.

Use F11 to send the web server to the EEPROM.

Download and extract "**test.html**" to the microSD card. These files are a set of simple html documents to be replaced later by actual content.

Once installed, you should be able to open your web browser and see the sample webpage using your local (LAN) IP address.

Providing access to the outside world:

At this point, if you have been experimenting with the Spinneret, you may have a web server which is working fine on your computer, but are ready to take the next step, opening it to the outside world.

Remember that the router is the gateway device between your Spinneret and the outside world. By adding some simple instructions, it can direct traffic from the outside Internet to your Spinneret.

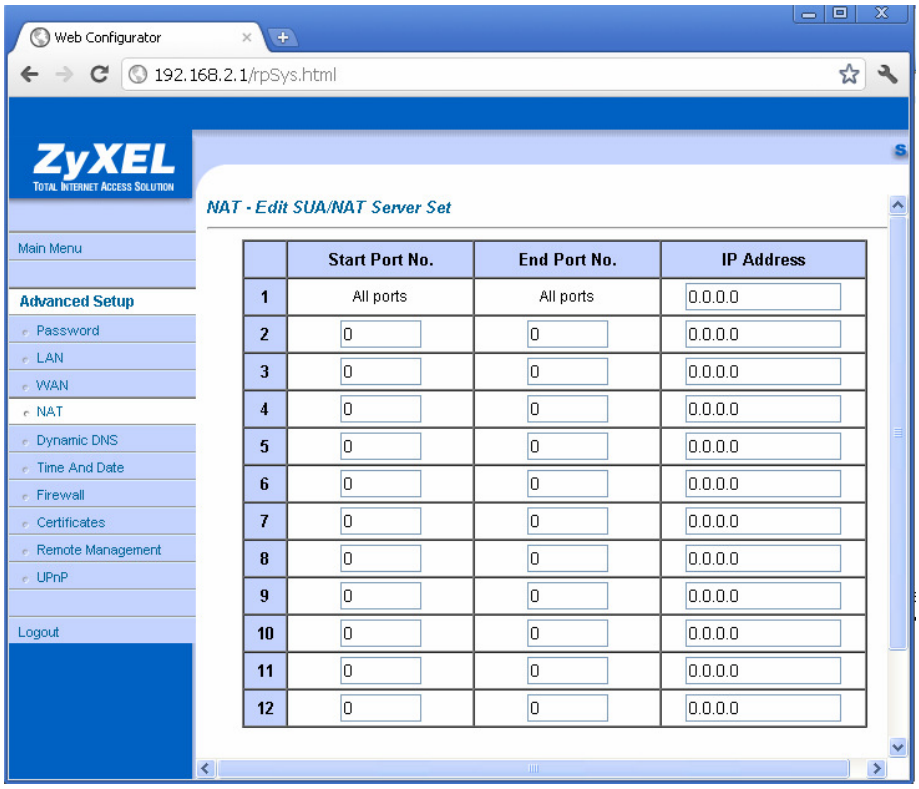
There are literally thousands of routers, all of which have variations between their configuration screens. I will give you an overview of how to program your router, but you can count on minor variations between my examples and yours.

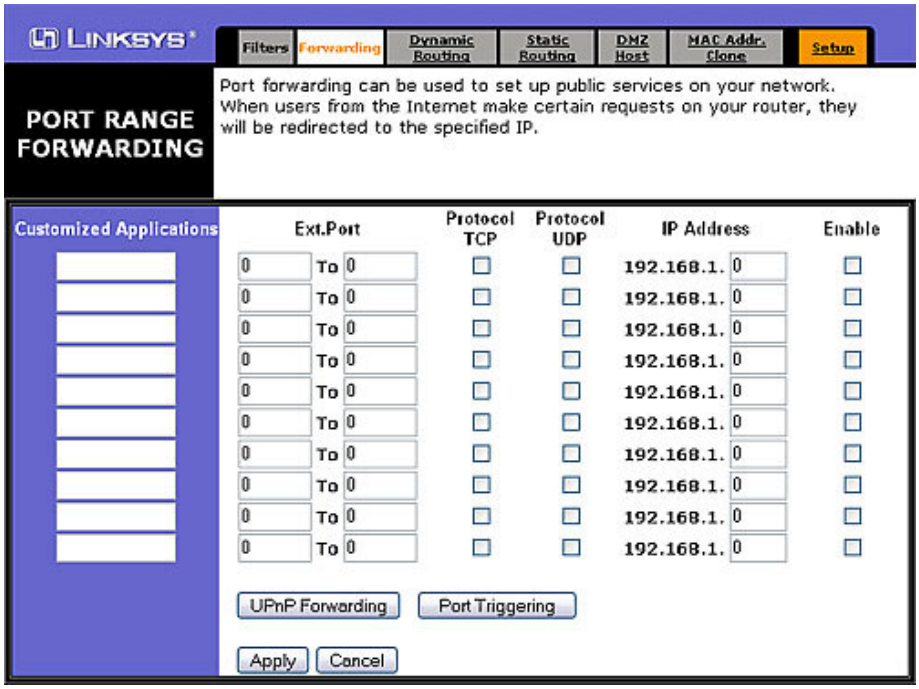
First of all, most routers can be configured using your web browser. Open your favorite browser and type the gateway address into the address bar. In my case the gateway address was 192.168.2.1.

You can be expect to be greeted by a login and password prompt, or just a password prompt like the one shown. When initially installing the router on the network, this login and password were assigned. (Hopefully, you recorded that information.) It is common for this password to be left as "factory default" and the entries, "admin" for either login, password, or both tend to go a long way on most routers which never had their access passwords configured.

If at first your login/password doesn't work, there is no harm to keep trying combinations until you get access. A quick Google search for "default" login/name and password for your router can also be profitable.

Once you have obtained access, you'll be presented with a webpage-like group options. This is where routers will differ from brand to brand, but you are looking for a screen like these examples.





These two examples share important similarities. Both ask for started and ending port numbers. With my Spinneret, I would put 80 in both boxes. The final bit of information is the IP address of the Spinneret. This is where I would put the 192.168.2.200, and in the case of the second example, there are a few

checkboxes to enable to get things working. Don't forget to Hit "Apply" before closing your connection to the router.

This configuration tells the router that any incoming Internet data directed at port 80 should be *forwarded* to 192.168.2.200. We call this "Port Forwarding."

Here's final check list to see if your Spinneret should work on the Internet.

1) First, obtain the local IP address data from a computer on your network, making note of the first part of the IP address, the subnet mask, and the default gateway.
2) Assign a local (LAN) IP address and port number to your Spinneret. If you are running web server software on the Spinneret. You can check to see it is working from your web browser at that address and port number.
3) Discover your global (WAN) IP address by going to http://whatismyip.com with your computer. This is the address you would give someone else attempting to access your Spinneret from the outside world. For example, if you used 192.168.2.200:5555 on your end, they might use 204.117.207.5:5555 from outside of your network.
4) Configuring your router to "forward" incoming traffic based on the port assigned to your Spinneret's software.

Post your global address and port number to the Spinneret forums! There is always someone ready to help test out your Spinneret server.

Addendum #1
Using a Dynamic DNS service

Residential Internet services use "dynamic" IP addresses which have the tendency to change from time to time. "Static" (unchanging) IP addresses can be purchased from most service providers at an additional monthly fee.

Another way that this problem can be tacked is through the use of a Dynamic DNS service. These websites act as a central hub, keeping track of the when your IP address has changed through the use of a special program you leave running on your PC.

A couple of these services are:

http://www.no-ip.com
http://www.dyn.com

These companies offer the use of free subdomains. (The subdomain is the "www" part of an Internet address.) Your address might look like: http://spinneret.no-ip.com. Additional features including full domain hosting can also be purchased from these companies at a cost.

Addendum #2
PropBASIC on the Spinneret

Terry Hitt (Bean of Parallax forums) has created a version of embedded BASIC which you can also run on the Spinneret, accessible by TELNET.

You'll need to obtain the following files:

Embedded BASIC for Spinneret: (Top thread at: http://goo.gl/mkvkI)
Brad's Spin Tool Compiler: (http://www.fnarfbargle.com/bst.html)

Once you extract the Embedded BASIC, you will have three folders. Copy the Brad's Spin Tool Compiler executable to the folder named, "Code".

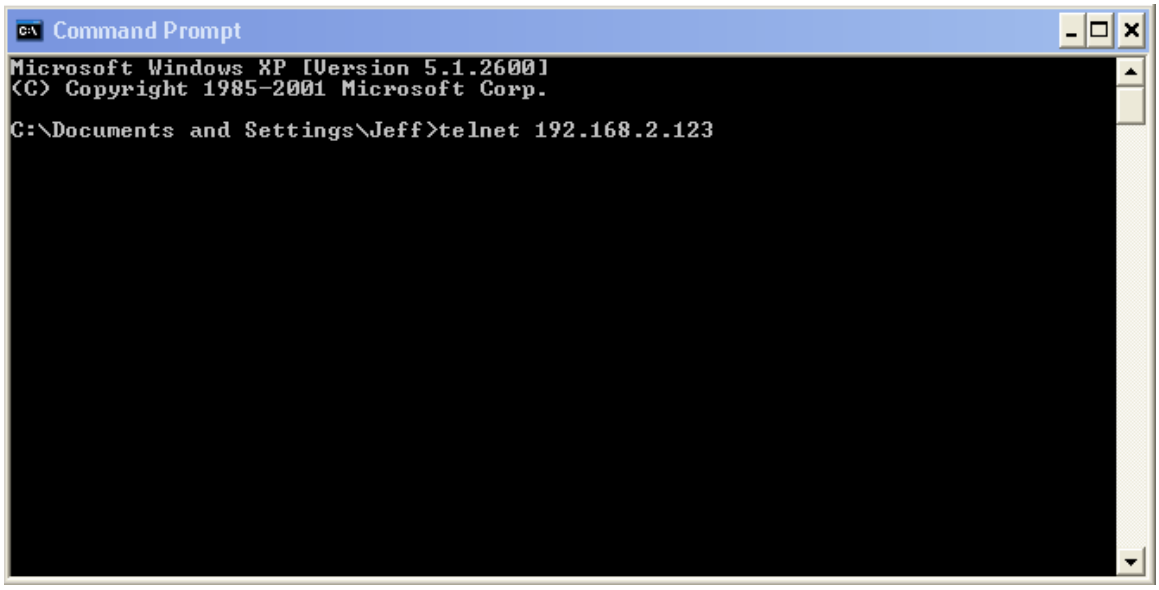
Using Brad's Spin Tool, open the file named, "PE-Basic Compiled (Use BST).spin". Scroll down to line 538, and change both "SetGateway" and "SetIP" lines to match the settings for your network. (As discussed earlier in this document.)

```
537
538   SetGateway(192, 168, 2, 1)
539
540   SetSubnet(255, 255, 255, 0)
541
542   SetIP(192, 168, 2, 123)
543
```

Example: I've changed my Gateway to 192.168.2.1, and IP to 192.168.2.123.

Once you have compiled Spinneret PropBASIC, you should be able to open a Command Prompt and telnet to the Spinneret in BASIC.

Type: **TELNET (followed by the IP address you assigned on line #542)**



If everything works as it should, the window should clear with a flashing cursor. Type this little BASIC program:

```
10 HIGH 23
20 PAUSE 100
30 LOW 23
40 PAUSE 100
50 GOTO 10
```

Type RUN after entering the last line. The ORANGE light (on I/O 23) on your Spinneret should be flashing off and on.

In addition to controlling the other I/O lines on your Spinneret, you can also share your Spinneret-loaded basic with the rest of the Internet by adjusting the port number, (lines #544, & #545) and creating a "Port forward" in your router. Once you change the port number of the software, you'll need to use the following to connect to your Spinneret.

TELNET (IP ADDRESS) (PORT#)

Additional documentation on both PropBASIC and the Spinneret PE Project is located in the "Documents" folder included in the Embedded BASIC download.

Addendum #3
Email Notification from your Spinneret

The Spinneret is capable of sending email as well.

Mike Gebhard (MikeG of Parallax forums) has also written a simple SMTP email program which will transmit email from your Spinneret. An example usage might be connecting a motion sensor to the Spinneret and sending an email when it is triggered.

Download the email server example from: http://goo.gl/8tdde

Before we jump into the configuration of the software, let's talk a bit about email servers and security. Most Internet email severs use authenticated login which prevents them from being used as "open relays" for email spam. The Spinneret code is expecting an open email server.

There are two ways to use Mike's Spinneret code.

1) You can add the proper authenticated login/password to the Spinneret code. Conversion of your login name and password to BASE64 can be done with this online tool: http://www.webpan.com/Customers/Email/base64_conversion.htm You will also need to add a few lines of code to **PUB SendTestEmail()** to create an authenticated session. (See *Understanding Email Transmission* below.)

2) Run a local open SMTP server program on a computer on your network which the Spinneret can use to relay mail onto the Internet. A program for Windows called, "Free SMTP Server" can be downloaded from: http://www.softstack.com/freesmtp.html

We'll use the latter (and simplest) solution in this example.

Open the "HTTPServer.spin" and change lines 53 and 54 (gateway and IP) to correct gateway and IP for your network. Change line 60 (emailIp) to the IP address of the computer running the "Free SMTP Server" software.

```
53  gateway             byte    192, 168, 2, 1
54  ip                  byte    192, 168, 2, 123
55  port                word    5000
56  port2               word    5010
57  remoteIp            byte    65, 98, 8, 151 {65.98.8.151}
58  remotePort          word    80
59  uport               word    5050
60  emailIp             byte    192,168,2,8
61  emailPort           word    25
```

Scroll down to lines 306, 310 and 314, changing the HELO, MAIL FROM and RCPT TO email addresses.

```
305   'Send greeting
306   StringSend(id, string("HELO agavejoe@cox.net", 13, 10))
307   pause(wait)
308
309   ' From Address
310   StringSend(id, string("MAIL FROM: <agavejoe@cox.net>", 13, 10))
311   pause(wait)
312
313   'To Address
314   StringSend(id, string("RCPT TO: <agavejoe@cox.net>", 13, 10))
315   pause(wait)
```

You may also want to alter the subject and message body of the text which can be found a few lines further down.

```
321   'Subject line
322   StringSend(id, string("SUBJECT: Email from the Spinneret", 13, 10))
323   pause(wait)
324
325   'Email body
326   StringSend(id, string("Hello from the Spinneret.", 13, 10))
327   pause(wait)
```

My own email server initially saw emails from my Spinneret relay as spam, so I added a filter in my email which identified the messages by their SUBJECT line and kept them from go to my SPAM folder.

If you need additional message lines, simply duplicate lines 326 and 327 as many times as needed.

Did you know that you can also send email messages to most cell phones? Contact your cell phone carrier for the email address which acts as a gateway between email and text messaging. (Verison is cellphone#@vtext.com)

Understanding Email Transmission:

PUB SendTestEmail() is a conversation with the email server which looks something like this: The bold text is the Spinneret's responses.

220 ESMTP server ready
HELO your@email.net
250-hello [0.0.0.0], pleased to meet you
250-HELP
250 OK
MAIL FROM: your@email.com
250 2.1.0 sender ok
RCPT TO: your@email.com
250 2.1.5 recipient ok
DATA
354 enter mail, end with "." on a line by itself
SUBJECT: Email from the Spinneret
Hello from the Spinneret.
.
250 mail accepted for delivery
quit
221 closing connection

By inserting a few additional lines of code, you could also create an authenticated login. This time I have converted my transmitted login name and password to BASE64. An example of the conversion might look something like this:

220 ESMTP server ready
EHLO your@email.net
250-hello [0.0.0.0], pleased to meet you
250-HELP
250 OK
AUTH LOGIN
334 o12a4kfoka
amVmZmxlZGdlcg==
334 prwejrknfd093492
TmljZSBUcnkh
235 Authentication succeeded
MAIL FROM: your@email.com
250 2.1.0 sender ok
RCPT TO: your@email.com
250 2.1.5 recipient ok
DATA
354 enter mail, end with "." on a line by itself
SUBJECT: Email from the Spinneret
Hello from the Spinneret.
.
250 mail accepted for delivery
quit
221 closing connection

Addendum #4
Controlling the world from a Spinneret Webpage

The Spinneret is capable of controlling "real world" items from its available I/O. With a few modifications of the Mike's Spinneret webserver we can add a custom webpage allow us to control one of the LEDs on the Spinneret.

While the actual code itself has been provided with this documentation, I'd encourage you to make the changes yourself as it will help you understand the requirements necessary to expand this example to controlling multiple items.

Open the file "**HTTPServer.spin**" and make the following additions:

Around line 70, just under **PUB Initialize**, add the following line:

```
dira[23]~~
```

Jump down the line 660. You should see a block of code which looks like this:

```
PRI Dispatcher(id)
{    }
  if(strcomp(Request.GetName(id), string("index")))
    Index(id)
  if(strcomp(Request.GetName(id), string("post")))
    Post(id)
  if(strcomp(Request.GetName(id), string("upload")))
    Upload(id)
  if(strcomp(Request.GetName(id), string("ajax")))
    Ajax(id)
  if(strcomp(Request.GetName(id), string("getdir")))
    GetDir(id)
    return
  if(strcomp(Request.GetName(id), string("gettime")))
    GetTime(id)
    return
  if(strcomp(Request.GetName(id), string("firmware")))
    Firmware(id)
    return


  StaticFileHandler(id)
  return
```

Make the following changes:

```
PRI Dispatcher(id) | bypass

  bypass := 0

  '' Checks for commandline entries
  if(strcomp(Request.GetName(id), string("hello")))
    Index(id)
    DisplayStuff(id)
    bypass:=1     '' Don't look for actual file.


  if(strcomp(Request.GetName(id), string("post")))
    Post(id)
  if(strcomp(Request.GetName(id), string("upload")))
    Upload(id)
  if(strcomp(Request.GetName(id), string("ajax")))
    Ajax(id)
  if(strcomp(Request.GetName(id), string("getdir")))
    GetDir(id)
    return

  if(strcomp(Request.GetName(id), string("gettime")))
    GetTime(id)
    return
  if(strcomp(Request.GetName(id), string("firmware")))
    Firmware(id)
    return

  if bypass == 0
    StaticFileHandler(id)
  return
```

When the Spinneret webserver encounters a request for "hello.html" it will bypass it's usual file retrieval routine, jumping to a PUB called "DisplayStuff" instead. The "**hello**" could be replaced with any other appropriate word for your application. Be sure and make the changes to the last three lines as well.

Scroll down to next section:

```
PRI Index(id)
  pst.str(string(13, "Index Processor", 13))
  pst.str(string("led := "))
  pst.str(Request.Get(id, string("led")))
  pst.char(13)
  pst.str(string("this := "))
  pst.str(Request.Get(id, string("this")))
  pst.char(13)
  return
```

Make the following changes:

```
PRI Index(id)
  pst.str(string(13, "Index Processor", 13))

  led:=str2dec((Request.Get(id, string("led"))))

  if led == 208
      pst.str(string("LED is OFF"))
      outa[23]:=0
  if led == 218
      pst.str(string("LED is ON"))
      outa[23]:=1

  pst.char(13)
  pst.str(string("this := "))
  pst.str(Request.Get(id, string("this")))
  pst.char(13)
  return
```

The section of code executes the actual changes on the Spinneret's I/O 23 which has an orange LED. This could be adjusted easily to an open I/O pin.

Insert the following two code blocks beneath the section we have been working on to finish our LED control.

```
PRI DisplayStuff(id)   | headerLen

''Build HTML Header
headerLen := Response.BuildHeader(Request.GetExtension(id), 200, false)
Socket.txTCP(id, @txdata, headerLen)

StringSend(id, string("<form>",13,10))
StringSend(id, string("<input type=radio name=led value=1 /> Led On<br /> ",13,10))
StringSend(id, string("<input type=radio name=led value=0 /> Led Off ",13,10))
StringSend(id, string("<form>",13,10))
StringSend(ID, string("<input type=submit /> Change the State of LED."))


PUB str2dec(stri) | index2
  result := 0
  repeat index2 from 0 to strsize(stri)
    result *= 10
    result += byte[stri][index2] - "0"
```

Once you've added these code blocks, you should be able to access the LED control from your web browser using http://{spinneret address}/hello.html

The controls can be expanded providing that you keep the actual code in "DisplayStuff" reasonably short so that a buffer overflow is never created.