What can I do with a Parallax WX WiFi Module

First let's assume you don't have a network in your environment, sucks to be you. Not to fret if you have two WX modules on hand. You can make one an Access Point unit and the other a Station. With this combination they can talk to each other. In fact you can have several Station units connect to that one Access Point unit and they can all talk to each other. A party network.

When you first get a Parallax WX unit it is in AP or Access Point mode. This is indicated by a steady blue LED when the unit is powered on in addition to the green LED.

In order to connect to this unit you need to know the name of the unit which can be determined by scanning for network names or using a serial connection to the unit. You can also put the unit in AP mode from a serial connection. A serial connection uses a baud rate of 115200 by default but can be changed to other values.

With a serial connection you issue the following command:

```
xCHECK:module-name<cr>

x=S,wx-123456<cr>
```

This is the name that will be used to connect to the unit. Passwords will not be used for this access point connection which means anyone scanning for a network can find this unit and connect to it.

Now we will be working with the other Parallax WX module to configure it to connect to this new network playground.

Now this unit which should also be in Access Point mode needs to be switch to station mode so that we can connect to our network name determined above. Using a serial connection to this unit we will issue the following command:

```
xSET:wifi-mode,STA<cr>

x=S,0<cr>
```

Now we are in Station mode and the LED on the unit should now be off since it is no longer in AP mode and does not have a connection to a network. The next step is to issue a join command to connect to the other units name using the following command:

```
xJOIN:wx-123456,<cr>

x=S,0<cr>
```

If all goes well the blue LED will start to flash slowly and we will be able to talk to the AP and have an IP address assigned by the Access Point unit. This can be done using the following command:

```
xCHECK:station-ipaddr<cr>

x=S,192.168.4.2<cr>
```

Each station that connects to this Access Point will be assigned an IP address by this unit and will keep track of what has been assigned. So the base address of the Access Point will be 192.168.4.1. This is the address that needs to be used to talk to this unit. Let's give it a try.

What can I do with a Parallax WX WiFi Module

By default the units create a telnet port that is used to send serial data in and out of the unit when you don't use the command syntax to control the units. We can use this port to send data between the two units now.

```
xCONNECT,192.168.4.1,23<cr>

x=S,6<cr>
```

The return value of 6 is the connection number we need to use to send data over that newly created connection. To send data we need to know how much data we are going to send and then the data itself. Let's try something simple for now. How about "Hello World". This is 11 bytes long.

```
xSEND:6,11<cr>

Hello World

x=S,0<cr>
```

Now in this example we did not include the Carriage Return or Line Feed in the send data so all that was sent was "Hello World" and you should see it on the Access Point unit as such. Also since we have not closed are connection we can continue to send data such as "Did you see this".

```
xSEND:6,16<cr>

Did you see this

x=S,0<cr>
```

If your connection gets closed you will see an error return code when you try the send command.

Now we need to close the connection and free it up for use by other calls.

```
xCLOSE:6<cr>

x=S,0<cr>
```

In addition we can send data the other way by reversing the connection. On the Access Point unit you would do a connection to 192.168.4.2 instead of 1 and now you can send data to the Station unit.

So far we have not had to do anything on the receiving side to receive the data other than except the data coming from the serial port. There is one other port available and that is port 80 or HTTP. Using this port we can create URL listeners that will send data to our unit when a connection is sent using this URL.

First on the Access Point unit setup a listen URL that we are interested in receive data from. For this example let's use HELLO. The response value will be the id of the setup URL that you need to keep track of when processing.

```
xLISTEN:HTTP,/HELLO<cr>

xS,1<cr>
```

What can I do with a Parallax WX WiFi Module

Now on the station unit we try and make a connection with this unit and send some data. One would thing we would use an HTTP command but in this case we use a TCP command in that HTTP really is a TCP connection in the background with HTTP syntax on top of it.

```
xTCP:192.168.4.1,80<cr>

x=S,6<cr>

xSEND:6,23<cr>

GET /HELLO HTTP/1.1<cr><lf><cr><lf>

x=S,0<cr>

x=D,6,84<cr>
```

For this example I had command-events enabled so that it would respond as soon as an event happened. Use the following to set that option from the serial connection:

```
xSET:cmd-events,1<cr>

x=S,0<cr>
```

Here we get S,0 saying are request to send data to the remote was accepted and the D is the response from the Access Point or our application with the data. HTTP has two types of requests GET and POLL. The GET request is the simplest to use. In this example a GET was issued against URL /HELLO. We could also supply data that we want processed like this:

```
GET
/HELLO?variable=value&variable2=val
ue2 HTTP/1.1<cr><lf><cr><lf>
```

Now the two CR/LF codes must be sent to end the HTTP request otherwise it's not a valid request and will not get processed by the remote WiFi module.

If you didn't setup the remote side correctly such as the wrong URL for the request you will get back bad request data like the following:

```
HTTP/1.1 404 OK
Server: esp8266-httpd/0.4
Connection: close

404 File not found.
```

On the remote side or host side where we setup the listen event we get the following with command events turned on:

```
x=!G,5,1<cr>
```

This indicates that a URL was matched and it matches URL number 1 on handle 5. So we need to call processing for that URL and we need to access the values passed in the request like so:

```
xARG:5,WORLD<cr>
x=S,25<cr>
x=S,Mike<cr>
```

In this case the request was like this “GET /HELLO?WORLD=25” or “GET /HELLO?WORLD=Mike”. Now you can format your answer and send back a reply like this:

```
xREPLY:5,200,8,8<cr>
Whatever
xS,0
```

On the remote end the response will look like the following:

```
x=D,6,92<cr>
xRECV:6,1024<cr>
x=S,92<cr>
HTTP/1.1 200 OK
Server: esp8266-httpd/0.4
Connection: close
Content-Length: 8
<cr><lf><cr><lf>
Whatever
```

The event code shows there was only 92 bytes received but I elected to receive 1024 bytes. This is ok as it only returns the 92 bytes which is more than was expected because some HTTP header data was added to the response and needs to be removed to get just the data we need to process, which is the 8 bytes of “Whatever”. You will note that the header is separated by two <cr><lf> characters so you could use that.

Make sure you close your connection on this end as it will make it not available for the next request and you will run out of connections:

```
xCLOSE:6<cr>
x=S,0<cr>
```

Now let’s try doing UDP or a connectionless send and receive. With this method whatever you send is just sent and there is no guarantee that it will be received by the other side. Here is how you set it up. On the Station side we will issue this command using port 5678. You could use a number of different ports.

What can I do with a Parallax WX WiFi Module

```
xUDP:192.168.4.1,5678<cr>

x=S,5<cr>
```

On the Access Point side we will setup the same connection only with the Station IP address:

```
xUDP:192.168.4.2,5678<cr>

x=S,5<cr>
```

Now we are setup to send data between the two Parallax WX modules using Send and Receive:

```
xSEND:5,25<cr>

Whatever you want to send

x=S,0<cr>
```

On the other side we will get this with command events turned on:

```
x=D,5,25<cr>

xRECV:5,25<cr>

x=S,25<cr>

Whatever you want to send
```

Issuing the Receive command with handle 5 and size of 25 we receive the message send by the other side. Likewise this side could turn around and send a message back to that unit just like this:

```
xSEND:5,25<cr>

Whatever you want to send

x=S,0<cr>
```

When your finished sending message to each other you just need to close the connection:

```
xCLOSE:5<cr>

x=S,0
```

You can also send the message to multiple Parallax WX modules or every computer on your network by using the broadcast address on your network. In this case it would be 192.168.4.255. Let’s give this a try with are two units:

```
xUDP:192.168.4.255,5678

x=S,5<cr>
```

This works just like the previous example. In both cases we do not know where the packet of data came from so to respond back to the other unit would be hard since we don’t know it’s IP address. With the broadcast method we don’t need to know the other IP address as the response will go to all units on the network.

It’s time to expand our network. To connect the Parallax WX unit to your network you only need to know the router name or SSID and the password. Use the following commands to connect to your home network:

```
xSET:wifi-mode,STA<cr>

x=S,0<cr>

xJOIN:SSID,password<cr>

x=S,0<cr

xCHECK:station-ipaddr<cr>

x=S,x.x.x.x<cr>
```

I found that the SSID is case sensitive and needed to be upper case for it to connect to my network. The password is always case sensitive. Even if it doesn’t connect to your network it returns the same answer. I used the “station-ipaddr” to make sure it connected. If it returns all zeros for the IP address it didn’t join your network or was not assigned an IP address. Also the blue LED will slowly flash when it is connected to a network just like before.

When a connection is made it caches the password and will automatically connect back up once power is applied to the unit.

Let’s say we have a web service out on the network and we need to get some data off of it. For this example I will build a request to parallax.com:

What can I do with a Parallax WX WiFi Module

```
xCONNECT:parallax.com,80<cr>

x=S,5<cr>

xSEND:5,18<cr>

GET / HTTP/1.1<cr><lf><cr><lf>

x=S,0<cr>

x=D,5,272<cr>

xRECV:5,272<cr>

x=S,272<cr>

HTTP/1.1 400 Bad Request
Server: awselb/2.0
Date: Sun, 21 Mar 2021
17:53:02 GMT
Content-Type: text/html
Content-Length: 122
Connection: close

<html>
<head><title>400 Bad
Request</title></head>
<body>
<center><h1>400 Bad
Request</h1></center>
</body>
</html>
```

It looks like there server requires more HTTP data than we sent and so we get back a bad response, but you get the idea of how data is send and received from the service. In some cases the service may use something other than port 80.

There are many more request that can be done with the Parallax WX module, these are just the basics.