



Using the ESP8266 module

by [Propellerpowered](#) on September 26, 2014

Table of Contents

Using the ESP8266 module	1
Intro: Using the ESP8266 module	2
Step 1: Obtaining and preparing your 8266 module	2
Step 2: Hooking it up	3
Step 3: Configuring the 8266 Module	4
Step 4: BASIC networking	5
Step 5: Communicating with the module	6
Step 6: Running a simple webserver in BASIC	6
Step 7: Inviting the Internet	7
Step 8: Closing Notes	8
Related Instructables	9
Advertisements	9
Comments	9

Intro: Using the ESP8266 module

This Instructable will teach you how to use those \$5.00 ESP8266 modules, as well as provide you some basic knowledge about networking. I'll be using the [Micromite Companion Kit](#) in my examples which is programmed in BASIC, however all of these instructions should be adaptable easily to your favorite micro.

In short, the ESP8266 module is a TTL "Serial to Wireless Internet" device. Providing your microcontroller has the ability to talk to a TTL serial device (most do) you'll be in business! The original instructions have been translated from Chinese into cryptic data sheets. We'll try to change that with this Instructable.

The ESP8266 module is a 3v device, but it's no wimp. It draws quite a bit of power. In fact, you'll probably need to make sure that your circuit's power supply can handle at least 1 amp of power. (In my case, I was using a simple 7.5v 500ma power supply. When I started working with this module, I switched it for a 7.5v 1amp power supply and had plenty of power.) As it turns out there is good reason for this; some Youtube videos have surfaced recently with folks seeing anything from 500 meters to a couple miles of transmission capability from this module. That's a lot of horsepower for \$5.00!



Step 1: Obtaining and preparing your 8266 module

I obtained my module from an [Ebay vendor](#) in the United States. The shipping was faster than China, but more importantly, the vendor provides the module without the pins soldered in.

The 8266 module isn't really breadboard friendly, but it's easy to convert it to a four pin module if you purchase the pin-less version. (or take a few minutes to remove the pins if you have obtained the other version)

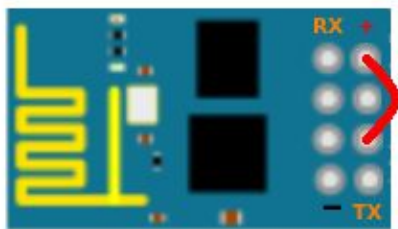
You'll need 5 pin connections to make the module work. (See image 1)

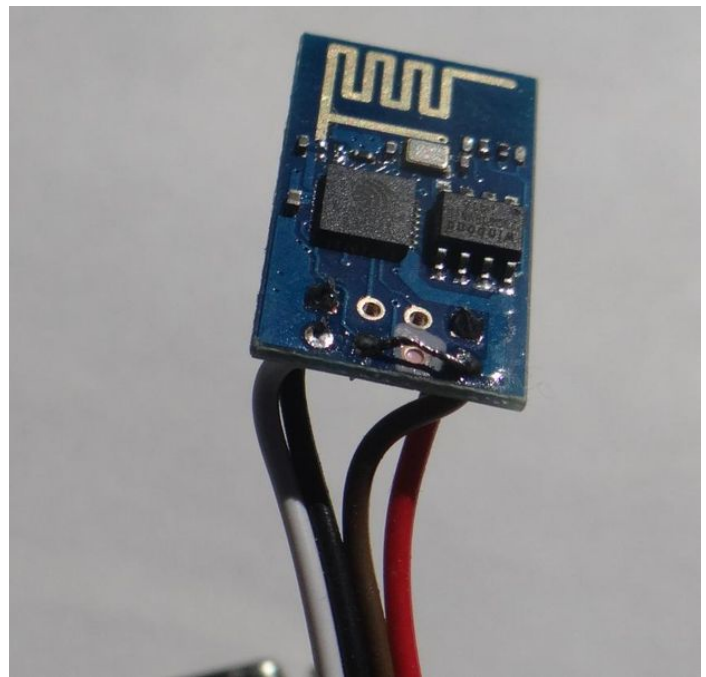
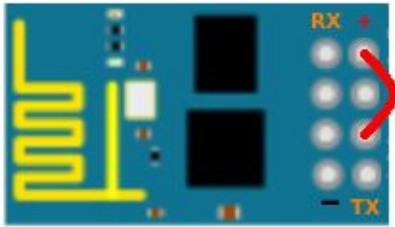
RX, TX, ground, and 3v connected to two positions on the module.

I sourced a 4pin female cable from my parts box and cut off one end.

I used a small amount of nail polish to carefully paint over the unused pin, then looped the 3v connection from the power pin over the unused pin into the center. (See image 2)

The end result is a 4pin module that is now breadboard friendly to plug into your project.





Step 2: Hooking it up

Once you have the module adapted, now make the four connections, (RX,TX,3v,Gnd) to your microcontroller. I've breadboarded mine to my Micromite Companion which is using the Micromite chip (created by Geoff Graham) running BASIC. The Micromite has multiple serial connections, and a console which I'm using as my interface to the 8266 module. You could even connect the module directly to your PC if you have a TTL-Serial-to-USB adapter. (Don't try to connect the module to a PC serial port directly, you could cause damage to the module or the your computer!)

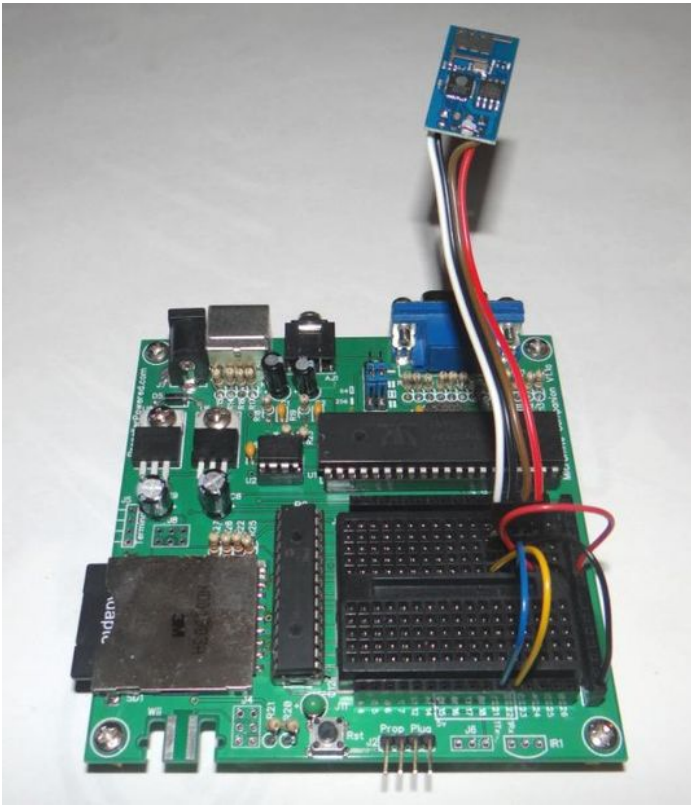
The correct connections to the Micromite Companion (Micromite) are RX to 21, TX to 22.

The default baud rate settings are 115200,N,8,1

Next, you'll need to use a terminal program to program the unit.

I've written the following BASIC terminal program for the Micromite:

- Open "Com1:115200" As #1
- terminal:
- a\$="" : b\$=""
- a\$=Inkey\$
- If a\$ <> "" Then Print #1,a\$;
- If Loc(#1) >=1 Then b\$=Input\$(1,#1)
- char = Asc(b\$)
- If char >31 Then Print b\$;
- If char =13 Then Print " "
- GoTo terminal



Step 3: Configuring the 8266 Module

You'll need to configure the module for your wireless network.

You should already know your wireless SSID and password, as we'll need those next!

From your terminal, type **AT** and press enter. If you get a cheery **OK** from the module, you have accomplished a big step in this Instructable!

Next, type **AT+RST** and give the module a moment to reset. You'll see a paragraph of data returned.

Type **AT+CWMODE=3** to set the module as both a client and an access point.

Don't worry if you make a typo in the process of doing these commands. (There's no backspace) Just hit the enter button and enjoy the broken English error message and retype the command.

Next, let's see if we can see your wireless router. Type **AT+CWLAP** and enter. You'll see something like this.

```
+CWLAP:(4,"Guest",-75)
```

```
+CWLAP:(4,"linksys",-80)
```

```
+CWLAP:(4,"family",-90)
```

```
+CWLAP:(4,"NETGEAR",-91)
```

See your access point? Type the following command, replacing SSID and password with your information.

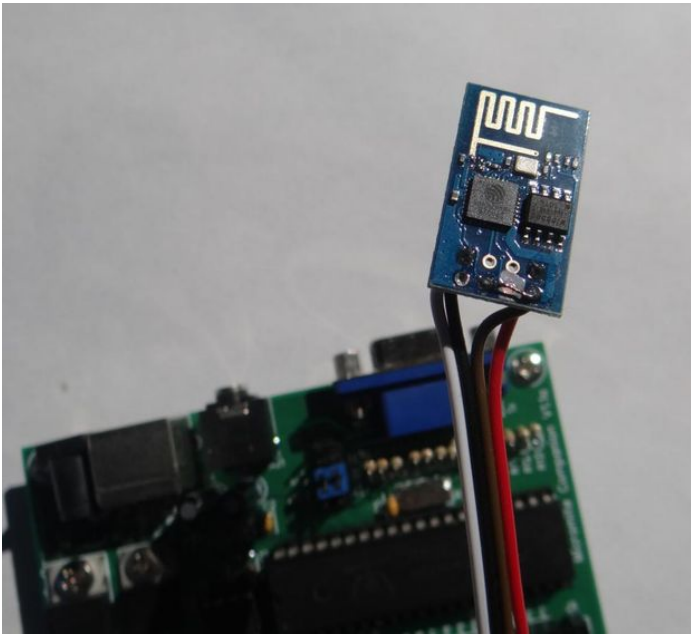
```
AT+CWJAP="SSID","password"
```

Congratulations! Your module is configured for your network.

Now we need to see what IP address has been assigned to it.

Type **AT+CIFSR** and press enter. Your module's IP address should be displayed.

```
192.168.1.20
```



Step 4: BASIC networking

Ok, we've lost about half our audience at the end of the last step. If you are still reading, it means that you have a working module, but need some guidance in the world of networking. Don't worry, you are in good hands. I'm going to condense a semester of networking classes (I used to teach CCNA) into just enough networking knowledge to be really dangerous. Sound like fun? Read on!

So you have the IP address that was displayed in the last step of the last page. (Did you write it down?)

Now what?

I'll assume you are at home with a wireless router somewhere in your home. It's probably connected to either a cable modem or DSL adapter. It's even possible that you have a single device which is doing both jobs. This device is the gateway to all of your internet travels, even the Instructable you are reading!

Your home network has a private side, and a public side. The private side of your network is all of the computers and devices which are connected to your wireless router. They can be wired to it's ports, or connected wirelessly.

You actually got a BIG CLUE to how the private side of your network is configured by the IP Address you were given to your module. Mine was 192.168.1.20.

Take a look at those first three numbers.. **192.168.1**

Those are the private side of your network. You might have 192.168.0 or even 10.0.0.

All of your computers and wireless devices on your network have an IP address that starts with those three digits.

It's that last digit (20 in my case) that determines the full address of each connection.

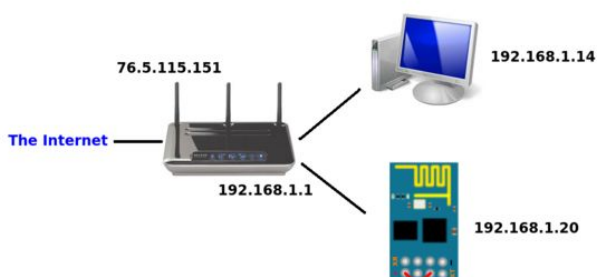
Each of your devices will have a different last number. Your wireless router probably uses 1. 192.168.1.1

The neat part about the private numbers is that typically there is room for up to 254 different devices and computers on your network right now! Talk about a LAN party!

Take a look at the image above.

Remember when I said that your wireless router has both a private side and a public side?

Your router receives a live IP address from your Internet provider. This address is unique to the entire world, and it's very important that it is! The wireless router actually contains two addresses. One is the private side, the other is the live IP address which is visible to the world. Don't worry, your router is designed to be the gatekeeper, controlling your web requests from your devices and keeping the bad guys out of your computers. The truth is, those private IP addresses are not visible from the outside world. (Unless we want them to be, keep reading!)



Step 5: Communicating with the module

Let's take a break from networking class to see if your little 8266 module is able to communicate with your network. An easy way to do this is using the PING command.

If you are using Windows:

Click on Start, Run, and type **CMD** and press enter.

Type **IPCONFIG** and press enter.

Type PING and the IP address of your module. (I typed PING 192.168.1.20)

If you are using Linux:

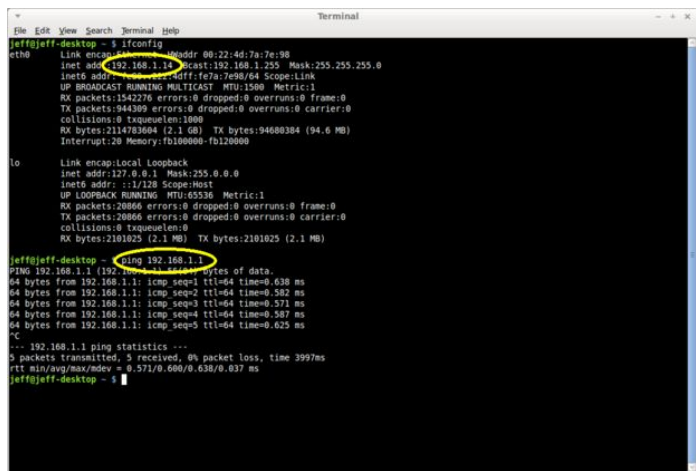
Open a terminal window

Type **IFCONFIG** and press enter

Type PING and the IP address of your module (I typed PING 192.168.1.20)

I've circle two pieces of information in my image. The first is the IP address of the computer I'm working at. (This is always good information) and the second is the IP address of what I actually PINGed. Did you catch me PINGing my wireless router? Good eyes! Ping your router as well as see if it answers. It's usually .1

A successful PING request will always return a set of numbers like mine did. If you get "Request Time Out" messages it means that something isn't communicating.



```
Terminal
jeff@jeff-desktop ~$ ifconfig
eth0: Link encap:Ethernet HWaddr 08:22:44:7a:7a:98
      inet addr:192.168.1.14 Bcast:192.168.1.255 Mask:255.255.255.0
      inet6 addr:fe80::22:44:7a:7a:98/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:1542276 errors:0 dropped:0 overruns:0 frame:0
      TX packets:944309 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2114783604 (2.1 GB) TX bytes:94680384 (94.6 MB)
      Interrupt:20 Memory:fb100000-fb120000

lo: Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr:::1::1 Scope:Host
      UP LOOPBACK RUNNING MTU:65536 Metric:1
      RX packets:20866 errors:0 dropped:0 overruns:0 frame:0
      TX packets:20866 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:2181825 (2.1 MB) TX bytes:2181825 (2.1 MB)

jeff@jeff-desktop ~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.638 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.582 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.571 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.587 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.625 ms
^C
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.571/0.600/0.638/0.037 ms
jeff@jeff-desktop ~$
```

Step 6: Running a simple webserver in BASIC

If you've gotten good PING results from your module, you are ready to start experimenting!

Let's start with a really simple web server written in MMBASIC. If you are using another micro, the BASIC program should be very easy to read and convert to your language.

Type in the little program and RUN it on your Micromite Companion.

If you are using a terminal program connected to your 8266 module, take note of the following commands..

```
AT+CIPMUX=1
AT+CIPSERVER=1,80
```

These two commands set up the magic to make the module automatically answer a request from another computer or device. In my case, I've configured the module to answer web requests on port 80.

Typical ports are as follows:

- 80 = Http web requests
- 8080 = Http web requests on networks on which 80 is blocked
- 23 = Telnet (text terminal) requests

Once you've run the program, open a web browser and type the address of your device (mine was 192.168.1.20) into the web address bar. That place where you've typed www.instructables.com.

The module seems to handle all of the formatting of the required HTML headers your web browser is looking for, so you can blast data directly. (At a reasonable speed of course!)

```
*Unsaved Document 1 - pluma
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
*Unsaved Document 1 X
Open "Com1:115200" As #1
Print #1,"AT+RST"
Pause 1000
Print #1,"AT+CIPMUX=1"
Pause 1000
Print #1,"AT+CIPSERVER=1,80"

terminal:
a$=""
b$=""
a$=Inkey$
If a$ <> "" Then Print #1,a$;
If Loc(#1) >=1 Then b$=Input$(1,#1)
char = Asc(b$)
If char >31 Then Print b$;
If char =13 Then Print " "
If char =71 Then GoSub webservice
GoTo terminal

webservice:
http "<TITLE>MICROMITE WEBSERVER</TITLE>"
http "<H3>Welcome to my Micromite Companion Webserver!<H3>"
Pause 1000
Print #1,"AT+CIPCLOSE=0"
Return

Sub http output$
Print #1,"AT+CIPSEND=0,"+Str$(Len(output$)+2)
Pause 300
Print #1,output$;
Print #1,Chr$(13)+Chr$(10);
Print output$
Pause 300
End Sub

Plain Text Tab Width: 4 Ln 1, Col 1 INS
```

Step 7: Inviting the Internet

So you can communicate from your web browser, your phone, laptop, or other Internet capable device to control your projects. I'll bet the ideas are already churning!

What if you want your friend in Ireland to control your project as well?

What if you want to control your project from somewhere other than your home network?

Those private IP addresses are only good while you are inside your own network.

It's time to talk about public address and something called router "Port Forwarding".

First, you need to know your router's public Internet address. It's easy to find. Simply point your web browser at www.whatismyip.com and you will be given your live IP address. (See first image)

Next you'll need to configure your router to allow requests from the outside world into your network and provide it a "rule" to allow certain traffic to your wireless module. This is called "Port Forwarding".

Remember when I said that I PING'd my wireless router at .1 to find its address?

Open your web browser and type the address of your wireless router into the address bar.

(Usually, it's 192.168.1.1 or 192.168.0.1 depending on your network, but you should know it now.)

The router will respond with a login/password response. Unless you have re-programmed it, (Most people haven't) it will accept **admin** and **password**. (Don't worry, your router doesn't allow folks from the Internet to program it by default!)

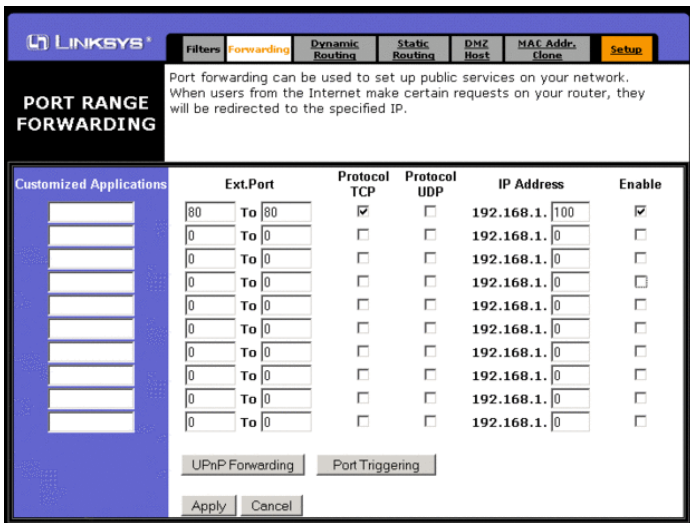
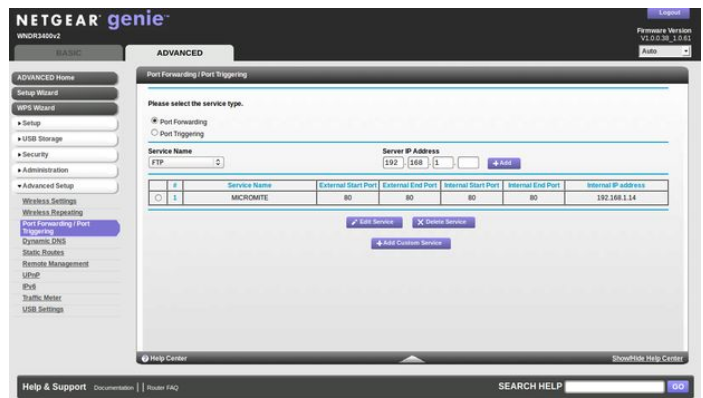
Here's the tricky part. You'll need to dig, (usually in the "advanced" menus) for something called "Port Forwarding" or "Forwarding". All routers are a little different, but don't be afraid to poke around. You aren't going to hurt anything.

Take a look at the 2nd and 3rd images. They are great samples of some common routers.

Once you found it, you'll need to add a rule with the following information:

The External Port# you want to use with your device. Most of the time, you'll use either 80 (if you want to provide web access) or 23 (if you want to provide telnet "text" access). Just use the same number twice as you see in my examples. Some routers will also ask for an Internal Port# as well. Again you can use the same numbers twice again. Finally, give the IP address of your device. (Mine was 192.168.1.20)

Once you've established this rule in your router, your device is now accessible from the world! From outside of your network, you can use your "live" IP address to access your 8266 module.



Step 8: Closing Notes

Remember when I said you'd get enough networking information to be dangerous? Welcome to the fun.

A few notes:

First, some Internet providers, in particular cable providers don't like to give you the ability to use the common lower port numbers (like port 80, or 23). They will claim that doing this is a violation of their service (nonsense!) or that they are protecting you by blocking these ports. (hog wash!)

If this is the case, just use higher port numbers, like 8080 or 2323 (or just make up a higher number you can remember easily.) Just add it at the end of the web or telnet request to make it work.

Also, from time to time your "live" IP address can change making it impossible for you to reach your project until you go back home and look up the new address with www.whatismyip.com. There is a great, free service which you can subscribe to called DuckDNS (www.duckdns.org) which will give you a name on their server and a little tool to run on your PC which will keep track of the changes. Instead of using the IP address, you'll be able to use {yourname}.duckdns.org. It really works well!

Need more help?

Drop over to our friendly forums at Propellerpowered and post up!

<http://forums.propellerpowered.com>



Related Instructables



Building your own Micromite Companion Minicomputer
by Propellerpowered



Getting started with the Pocket Mini Computer
by Propellerpowered



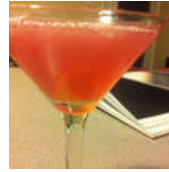
Calling Bulletin Board Systems (BBS) by Propellerpowered



Robotics on a budget by Propellerpowered



How to make a Winterfell Manhattan by BarNotes



How to make a Cosmopolitan by BarNotes

Comments

2 comments [Add Comment](#)



rpotts2 says:
very nicely done! so this is straight tcpip?

Sep 26, 2014. 2:02 PM [REPLY](#)



Propellerpowered says:
Yes, it's TCP. I believe the unit can also be set for UDP packets, but I really don't have a need for that in my projects.

Sep 26, 2014. 2:38 PM [REPLY](#)