Getting started with the ESP8266 ESP-01

by JayconSystems on January 28, 2016

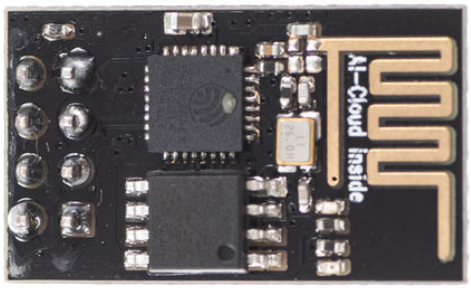
Table of Contents

Author:JayconSystems Jaycon Systems, LLC
We are tinkerers, inventors, geeks, students, teachers, hobbyist, engineers, and dreamers. We provide the building blocks to your projects!

Intro: Getting started with the ESP8266 ESP-01

The **ESP8266 ESP-01** is a Wi-Fi module that allows **microcontrollers** access to a **Wi-Fi network**. This module is a self-contained **SOC** (System On a Chip) that doesn't necessarily need a microcontroller to manipulate inputs and outputs as you would normally do with an **Arduino**, for example, because the ESP-01 acts as a small computer. Depending on the version of the ESP8266, it is possible to have up to 9 GPIOs (General Purpose Input Output). Thus, we can give a microcontroller internet access like the Wi-Fi shield does to the Arduino, or we can simply program the ESP8266 to not only have access to a Wi-Fi network, but to act as a microcontroller as well. This makes the ESP8266 very versatile, and it can save you some money and space in your projects.

In this tutorial we are going to show you how to set up the ESP-01 Wi-Fi module, configure it, and verify that there is communication established between the module and another device.



Step 1: Materials

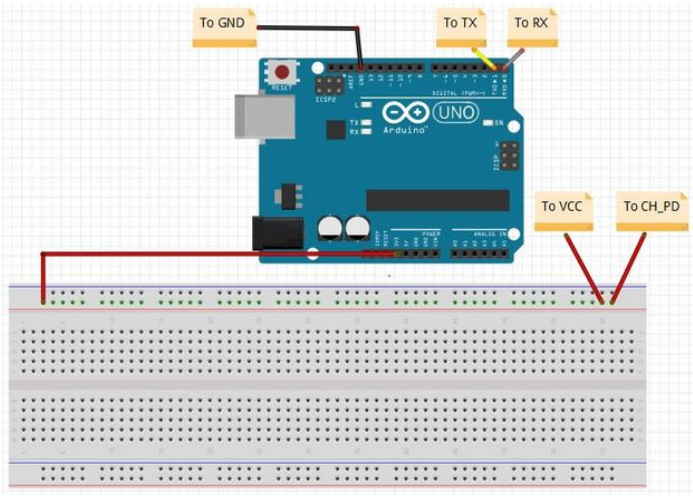
These are the components that you will need:

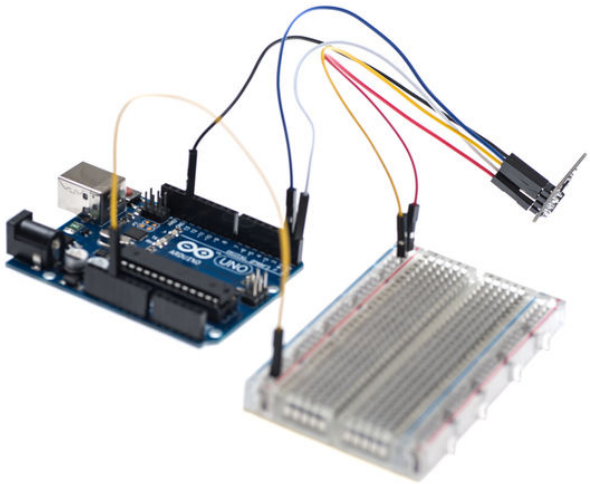
- ESP8266 Wi-Fi Module ESP-01
- Male/female jumper wires
- Breadboard
- Arduino UNO

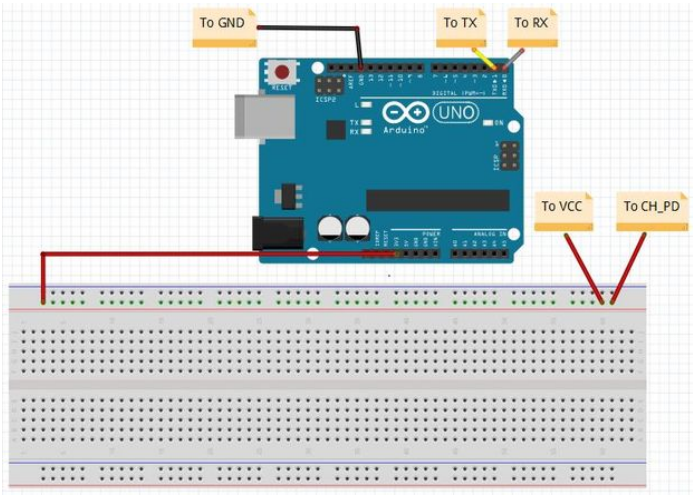
Step 2: ESP-01 Setup

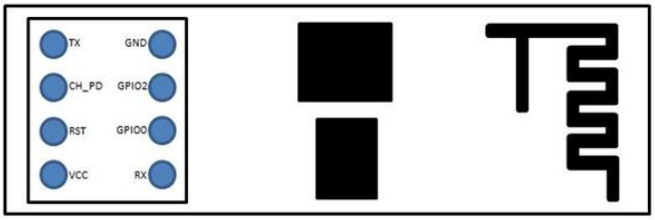
When you buy the **ESP8266 ESP-01**, it comes with a pre-installed **AT firmware**. It is possible to program the chip with another firmware such as **NodeMCU**, for example. However, AT firmware is compatible with the **Arduino IDE**, so we are going to use this firmware for this tutorial. If you want to know how to install a different firmware, then read the Miscellaneous section in this tutorial.

First use the jumper wires to connect the Wi-Fi module to the Arduino as shown in these images.









ESP8266 ESP-01	Arduino UNO
TX	Pin1
RX	Pin 0
CH_PD	3.3 V
RST	Unwired
VCC	3.3 V
GND	GND
GPIO2	Unwired
GPIO0	Unwired

Step 3: ESP-01 Setup continued

Upload the **BareMinimum** example to ensure that no previous programs are running and using the serial communication channel. Next, open the serial monitor and type the following command:

```
AT
```

You should get an “OK” response. This means that the module is working and that you are good to go. Now we are ready to test a two way communication between the module and another device.

Step 4: Basic AT Commands

The ESP8266 ESP-01 module has three operation modes:

1. **Access Point (AP)**
2. **Station (STA)**
3. **Both**

In **AP** the Wi-Fi module acts as a Wi-Fi network, or access point (hence the name), allowing other devices to connect to it. This does not mean that you will be able to check your Facebook from your device while the ESP-01 module is operating in the AP mode. It simply establishes a two way communication between the ESP8266 and the device that is connected to it via Wi-Fi.

In **STA** mode, the ESP-01 can connect to an AP such as the Wi-Fi network from your house. This allows any device connected to that network to communicate with the module.

The third mode of operation permits the module to act as both an AP and a STA.

Step 5: Basic AT Commands - STA Mode

In this tutorial, we are going to set the module to operate in **STA** mode by typing the following command:

```
AT+CWMODE=1
```

The corresponding number for each mode of operation is as follows:

- STA = 1
- AP = 2
- Both = 3

Step 6: Basic AT Commands - Check Mode

If you want to check what mode your **Wi-Fi module** is in, you can simply type the following command:

```
AT+CWMODE?
```

This will display a number (1, 2, or 3) associated with the corresponding mode of operation.

Step 7: Basic AT Commands - Connecting Wi-Fi Network

Once we have the **ESP-01** operating in **STA** mode, we need to connect to a **Wi-Fi network**. First we can check if we are already connected to one by sending the command:

```
AT+CIFSR
```

This will display the station **IP address** of our ESP-01 module. If you don't get an IP address after entering the previous command, use the following command to connect to your network:

```
AT+CWJAP= "Wi-FiNetwork","Password"
```

Type the name of your Wi-Fi network and the password to connect to it. Make sure you include the quotation marks. After a couple of seconds, you should get an "OK" response. You can check again to see if you have an IP address using the AT+CIFSR command.

Step 8: Basic AT Commands - Enable Connections

Then we need to enable multiple connections before we can configure the ESP8266 ESP-01 module as a **server**. Type the next command:

```
AT+CIPMUX=1
```

Once again, each number is associated with a type of connection:

- Single = 0
- Multiple = 1

The following step is to start the server at port 80:

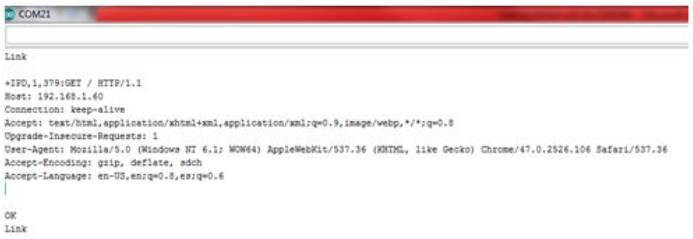
```
AT+CIPSERVER=1,80
```

The first number is used to indicate whether we want to **close server mode** (0), or **open server mode** (1). The second number indicates the port that the client uses to connect to a server. We chose port 80 because this is the default port for **HTTP protocol**.

Step 9: Basic At Commands - Response

Now, when we open a **web browser** and type the IP address of our ESP module we get the following response as shown in the image above.

This is the **HTTP** request that our computer sends to the server to fetch a file. It contains some interesting information such as what file you want to retrieve, name of the browser and version, what operating system you are using, what language you prefer to receive the file in, and more.



Step 10: Basic AT Commands - Send and Display Data

We can now use the following commands to send some data and display it in our web browser's window:

```
AT+CIPSEND=0,5
```

The "0" indicates the channel through which the data is going to be transferred; while "5" represents the number of characters that are going to be sent.

When we hit enter, the symbol ">" appears. This indicates that we can now type the characters that we want to send to the browser. In this example we chose "hello."

After a couple of seconds we get the response "SEND OK." This means that the data has been transmitted successfully to the client. However, nothing appears on the web browser's window yet. This is because it is required to close the channel first in order to display the characters. We use the following command to close the channel:

```
AT+CIPCLOSE=0
```

"0" indicates the channel that is being closed.

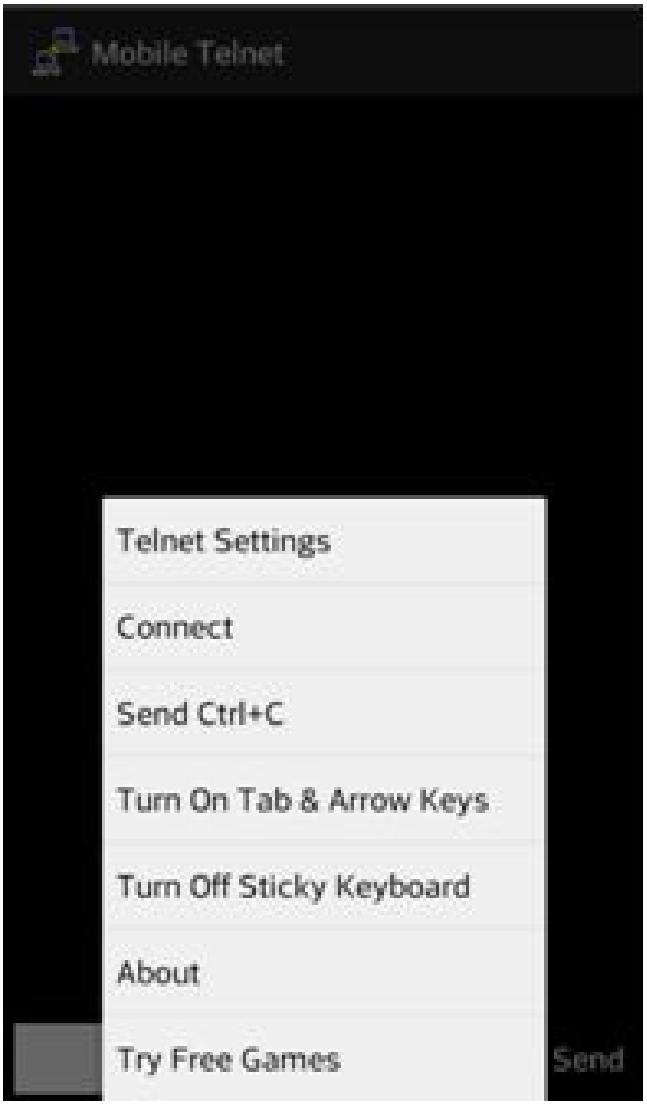
Once we hit enter, our message is displayed on the web browser's window as shown in the image above.

You can refer to the following site to see the **ESP8266 AT Command Set**:
http://www.pridopia.co.uk/pi-doc/ESP8266ATCommands...

Step 11: Check that our ESP-01 Receives Data - Mobile Telnet

Now we want to check that our ESP-01 module receives data. We will use the **Android** application **"Mobile Telnet"** to test this.

1. Open the Android application and from the menu select **"Telnet Settings."**



Step 12: Check that our ESP-01 Receives Data - Mobile Telnet con.t'

Type in the **IP address** and the **port number**.

Click “OK” and from the menu select “Connect.”

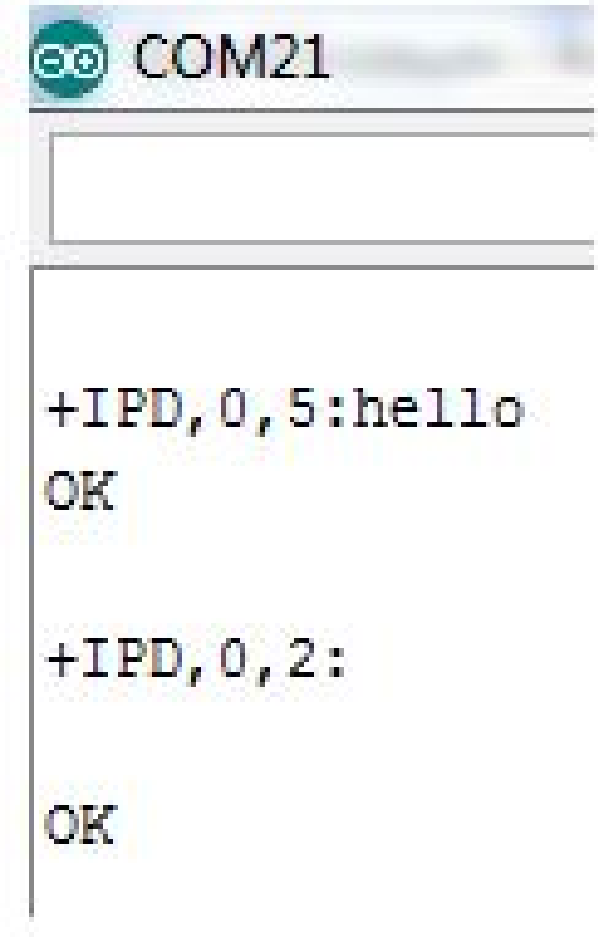
Step 13: Check that our ESP-01 Receives Data - Mobile Telnet con.t'

Type the characters that you want to send and then click the “Send” button.

Step 14: Check that our ESP-01 Receives Data - Mobile Telnet con.t'

We get the following response as shown in the image above on the **serial monitor**.

The message is successfully received and displayed.

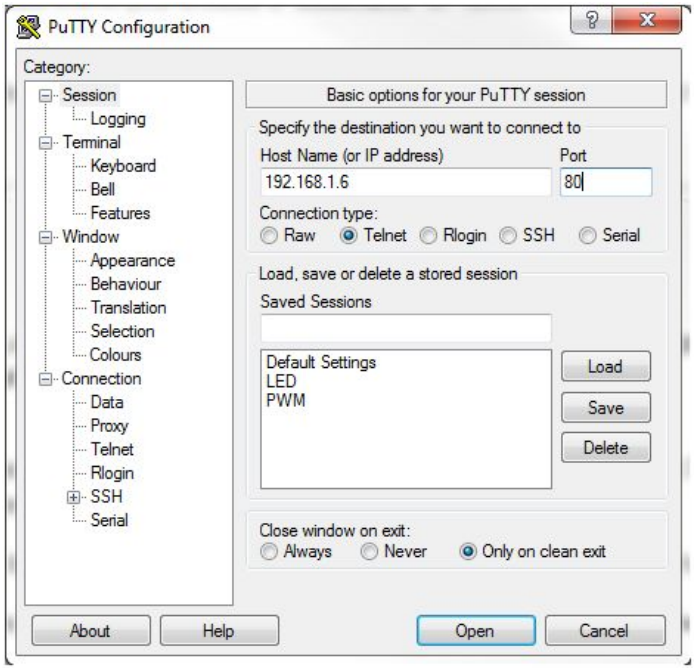


Step 15: Check that our ESP-01 Receives Data - PuTTY

Instead of **Mobile Telnet**, you can also use **PuTTY** to check that the ESP-01 is receiving data correctly. You can download PuTTY here.

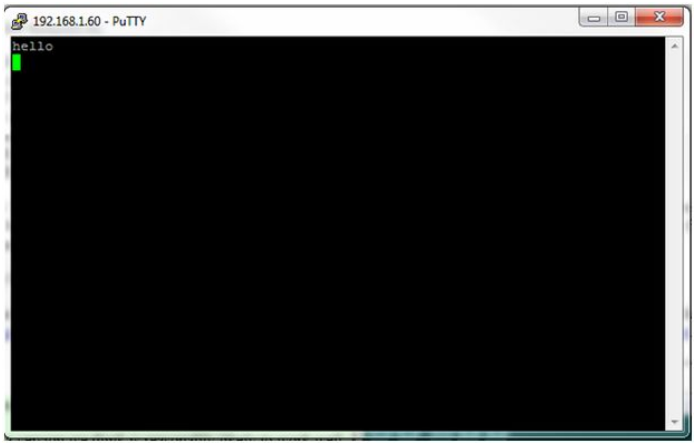
If you decide to use PuTTY follow these steps:

1. Open the program
2. Select “Telnet” as the connection type
3. Type the IP address and the port number
4. Click on “Open"



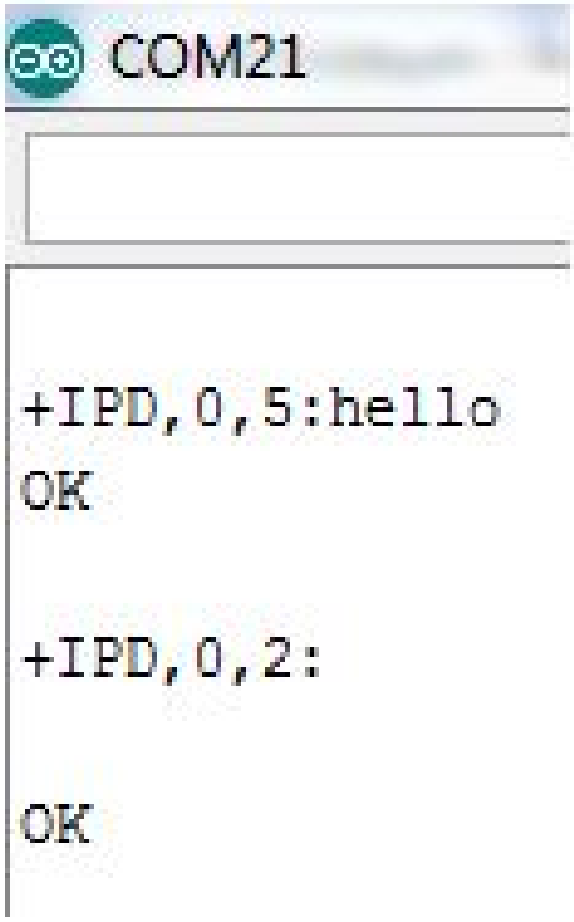
Step 16: Check that our ESP-01 Receives Data - PuTTY con.t'

5. Type the characters that you want to send and hit “Enter.”



Step 17: Check that our ESP-01 Receives Data - PuTTY con.t'

We get the same response as before.



Step 18: Miscellaneous - Different Firmware

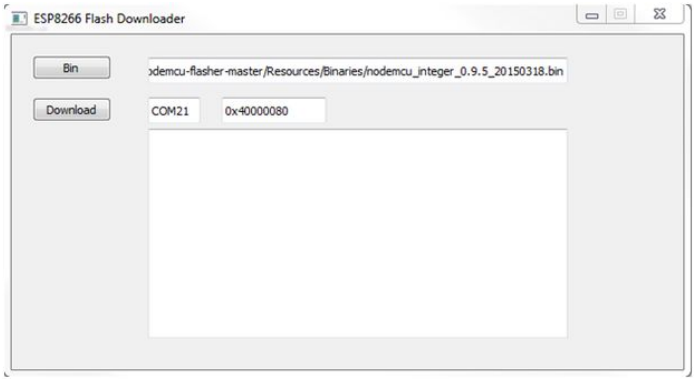
If you want to use a different firmware such as **NodeMCU**, you need to download an **ESP8266 flasher** such as this one. Then you need to download the **binary file**. You can use the following site to do it. Make sure you download just the **integer type**.

https://github.com/nodemcu/nodemcu-firmware/releases

Step 19: Different Firmware Set-up con.t'

Open the **ESP8266 flasher** and select the bin that you just downloaded. Select the serial port and type 0x40000080. Before you click on **"Download,"** make sure you **ground GPIO0**. This is required every time a new firmware is being flashed.

When you click on **"Download"** the flasher will delete the current firmware and start installing the new one.



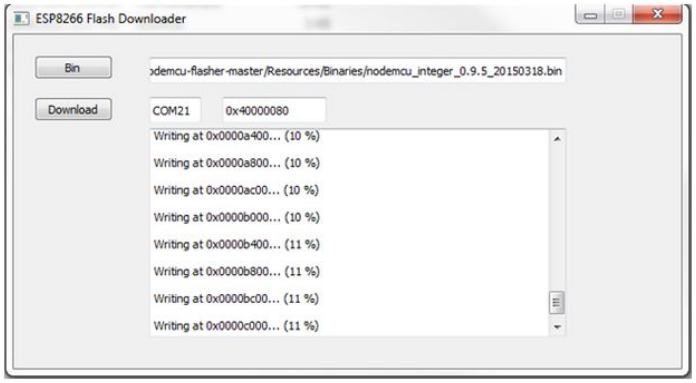
Step 20: Different Firmware Set-up con.t'

Once the **firmware** has been installed, you can disconnect **GPIO0** and use it normal.

We hope you enjoyed this **Instructable**, and you can find more tutorials about the ESP8266 and more on Jaycon System's website.

If you have any questions about this tutorial, do not hesitate to post a comment, shoot us an email, or post it in our forum.

Thanks for reading!



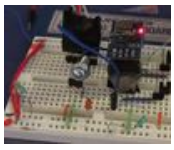
Related Instructables

Get Started with ESP8266 Using AT Commands, NodeMCU, or Arduino (ESP-12E) by acrobotic

Internet controlled electrical lamp with ESP8266 wifi (ESP01) IoT by shinteo

ESP8266-1 Enabled RC Turned Wifi Car With Browser Controlled Direction. by sumbasu

Wifi Remote Control using ESP8266 by NCKiwi



ESP8266 as Arduino by msuzuki777

ESP8266 Wifi Add on for Arduino Made Simple by drmpf

Comments