Connect an ESP8266 to your RaspberryPi

by **osmmspex** on July 18, 2015

Table of Contents

Intro: Connect an ESP8266 to your RaspberryPi

ESP8266 boards are pretty neat, but if you just bought one (And why wouldn't you for only $5?) and have realized that you don't have any obvious means (3.3V TTL USB serial device) to communicate with it, you can talk to it directly with a Raspberry Pi. Both use 3.3V signaling, so no level converting is required.

You will need:

Raspberry Pi desktop setup (just about any Pi should do, but with power, keyboard, screen, etc. network access is a plus)

ESP8266-01 (or equivalent)

Some jumper wire and nipper/strippers

Solderless Breadboard (or You can solder this instead)

2x Pull-up/down resistors

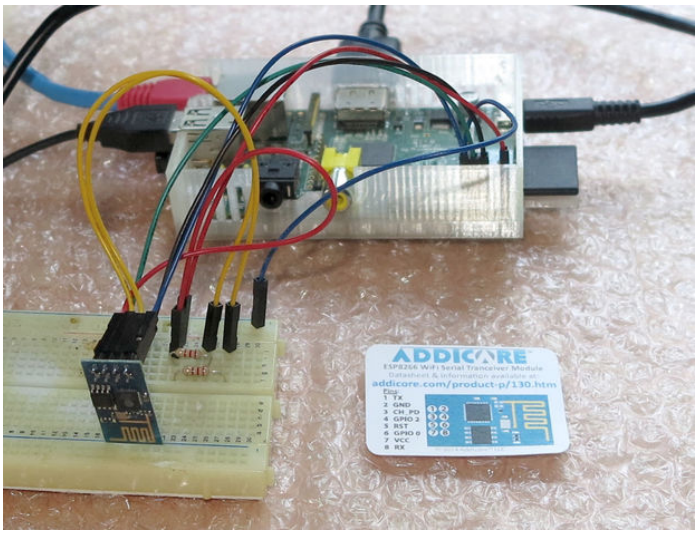
(optional) push-button switch

The process is roughly two steps,

1) Wire them together

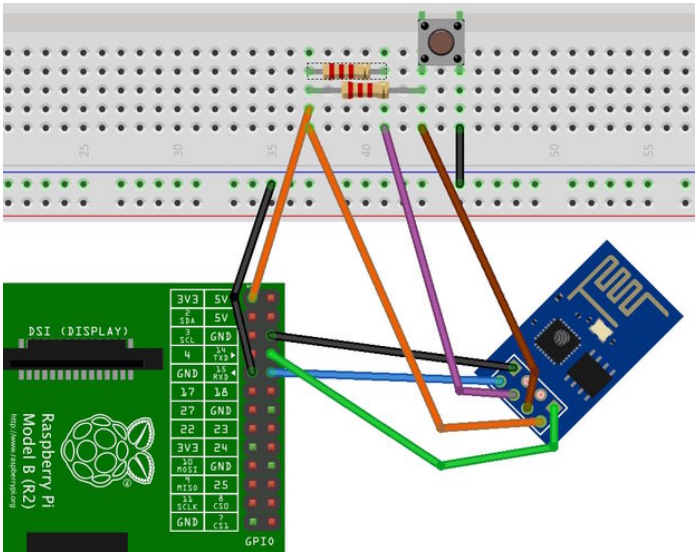
2) Configure Raspbian

3) Start talking to your ESP8266



Step 1: Wire them together

With the power off, connect the 3.3V and ground pins to one another. Similarly connect the RX to TX and vice versa. You will also want some pull down resistors and optionally a reset button. I've included a Fritzing diagram (above).



Step 2: Configure Raspbian Linux

Boot up the Pi and with super user privs make some edits (e.g. “sudo nano”).

Disable the Kernel serial console

Edit /boot/cmdline.txt to **remove** the underlined text:

dwc_otg.lpm_enable=0 rpitestmode=1 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 rootwait

Disable serial logins

Edit /etc/inittab and **remove** the (usually) last line:

2:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

Reboot

sudo shutdown -r now

Step 3: Start talking to your ESP8266

Connect to the serial port

You should now be able to interact with the ESP8266 via a terminal emulator program. I suggest GNU Screen for this (“sudo apt-get install screen”). Screen passes through your keystrokes unless you start with “CTRL-A” in which case you can then make screen do things. For example “CTRL-A” then “k” will close (kill) the session. screen /dev/ttyAMA0 115200 (the speed may vary depending on your ESP8266 board) (you may need to sudo this too, depending permissions) You can now send your ESP8266 “AT” commands. You’ll still need “return” and “CTRL-J” after each one, though.

Notes/Caveats

The Pi does not seem to recover well if you try to use the reset button. Best to shutdown the pi, then remove power, then power/boot back up rather than try the reset button.

Don’t mess with the wires while power is on. Shutdown the Pi ("sudo shutdown -h now") and wait for any blinking LEDs to stop and disconnect power before trying to connect/disconnect wires.

This linking won’t by itself give your Pi internet access via the ESP8266. But it does at least let you try out, configure, maybe even reprogram it.

Screen won’t relinquish the serial port if you just close a window or disconnect. This is on purpose, but can take some getting used to. If you think you disconnected from Screen but it is still running try “screen -ddR” to reconnect, and then you can to the kill as above. [Or reboot.]

The Pi and the ESP8266 can use a lot of power. This setup relies on the Pi’s 3.3V supply alone, which may not be up to the task if both devices are going to be going full blast. A more reliable/permanent solution would be to rig up separate power for the ESP8266.

Related Instructables

ESP8266 WiFi relay switch by EasyIoT

ESP8266 WiFi DS18B20 temperature sensor (ESP8266 Arduino IDE without Arduino) by EasyIoT

ESP8266 WiFi temperature and humidity sensor by EasyIoT

ESP8266 WiFI water leak sensor by EasyIoT

ESP8266 WiFi plant irrigation system by EasyIoT

ESP8266 WiFi touch screen thermostat by EasyIoT

Comments

1 comments **Add Comment**

DIY Hacks and How Tos says:
Nice tutorial. Thank for sharing.

Jul 18, 2015. 8:33 PM **REPLY**