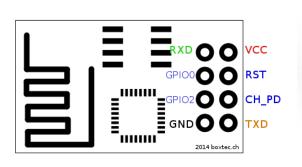
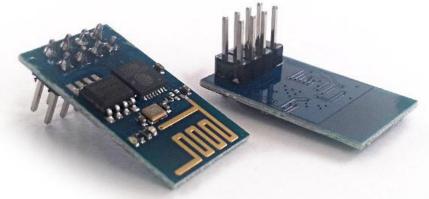
Programming ESP8266-01





Using

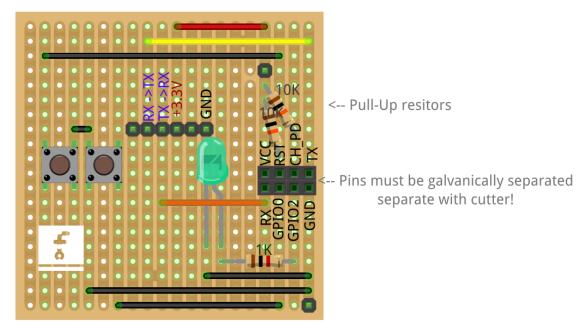
https://www.watterott.com/de/FTDI-

Breakout-Reloaded-V2



Using such a prototype board makes uploading a bit easier:

Programming Socket for ESP8266-01 with 3.3V FTDI



fritzing

Update Firmware: Nodemcu-flasher

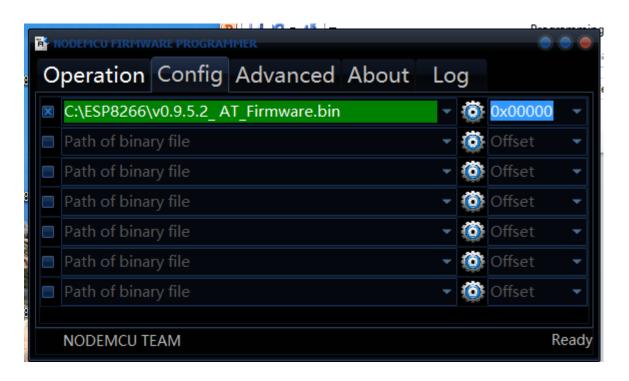
- https://github.com/nodemcu/nodemcu-flasher
- →ESP8266Flasher.exe



Get latest firmware and update!

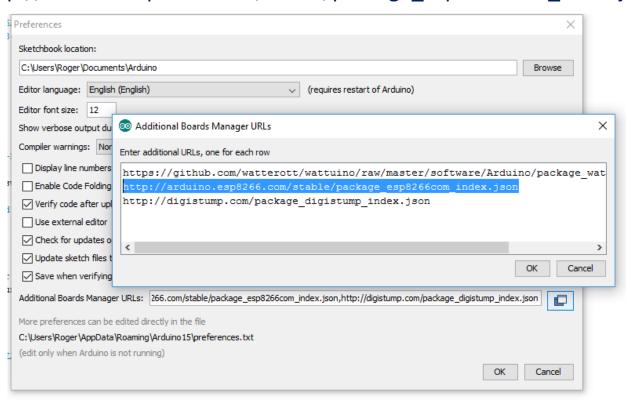
https://github.com/JhonControl/ESP8266-Flasher/tree/master/Firmware%20ESP8266/Firmware%20ESP8266%20V0.9.5.2_AT

C:\ESP8266\v0.9.5.2_AT_Firmware.bin



Arduino DIE 1.6.5

 Make sure you have the packages for ESP8266 support installed http://arduino.esp8266.com/stable/package_esp8266com_index.json



Use Arduino IDE 1.6.5 (not higher)

```
ESP8266AccessPointRequestForwarder2 | Arduino 1.6.5
File Edit Sketch Tools Help
                    Auto Format
                                                 Strg+T
                    Archive Sketch
  ESP8266Acces
                    Fix Encoding & Reload
#include <ESP
                                                 Strg+Umschalt+M
                    Serial Monitor
#include <ESP
                    Board: "Generic ESP8266 Module"
ESP8266WebSer
                                                                    igure server port
                    Flash Mode: "DIO"
int ledPin =
                    Flash Frequency: "40MHz"
const char Wi
                    Upload Using: "Serial"
const char Wi
                    CPU Frequency: "80 MHz"
                    Flash Size: "512K (64K SPIFFS)"
void setup()
                    Upload Speed: "115200"
  pinMode(led
                                                                  >PIO2 attach LED for Output
                    Port
                                                                    . and write LOW
  digitalWrit
                    Programmer: "Arduino as ISP"
  Serial.begi
                                                                   hitialize serial communication for Arc
                    Burn Bootloader
  Serial.prin
                                                               ,, write space line
  Serial.println("Starting WLAN-Hotspot" + String(WiFiSSID) + "...");
  WiFi.mode(WIFI AP);
                                                                // configure as access point
```

// Name of WIFI and Password

WiFi.softAP(WiFiSSID, WiFiPassword);

Start Upload in Arduiono IDE via FTDI

If you are using this board press the left button (RST) and the right button (CH_PD) at the same time, then release RST first and then CH_PD → The ESP8266 is now in programming mode

CH_PD (hold to enter programming mode on power-up)

Reset

CH_PD (hold to enter programming mode on power-up)

Pressing RESET while holding CH_PD makes it easy to enter programming mode...

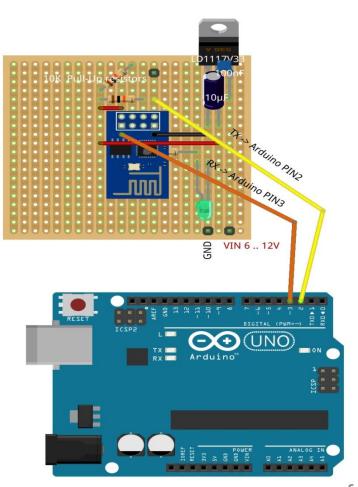
With the correct firmware on...

You should be able to use the following baud-rate (115200) and you can now send all kind of AT-Commands such as:

```
AΤ
AT+GMR
AT+CWMODE?
AT+CWMODE=1
(list of access points)
AT+CWLAP
(join network)
AT+CWJAP="YourNetworkSSID", "1234"
(return IP-Address)
AT+CIFSR
+CIFSR:STAIP,"192.168.0.19"
+CIFSR:STAMAC,"18:fe:34:81:16:23"
                                                                               Both NL & CR
                                                                                              115200 baud V
```

When software is on your ESP8266

ESP8266-01 Runtime Socket with 3.3V Voltage Regulator



Connect Pin 2 and Pin 3 to TX and RX of ESP8266

Use a Voltage Regulator 3.3 V to power your ESP8266 or any other power-source.

Arduino will not supply such amount of power!! (it's about 60mA @ 3.3V)

Larger picture of runtime-mode prototype board

