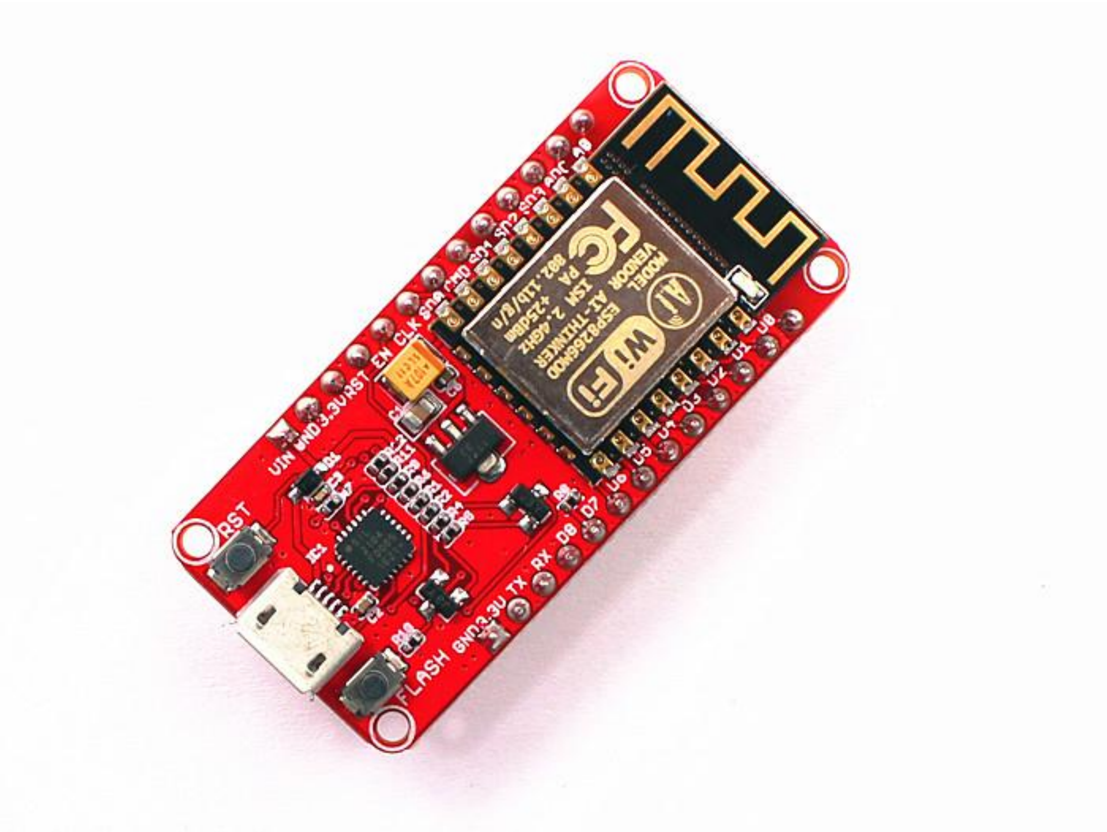
Usage of the NodeMCU Mini

NodeMCU Mini is an ESP8266 WiFi micro controller at 80MHz and at 3.3V logic. This microcontroller contains a Tensilica chip core as well as a full WiFi stack. You can program the microcontroller using the Arduino IDE for an easy-to-run Internet of Things core or using the Lua script language. We had ship with NodeMCU 0.9.5 build 20150704 powered by Lua 5.1.4, so we can directly use the Lua language to DIY the IOT project. We wired up a high-quality SiLabs CP2104 USB-Serial chip that can upload code at a blistering 921600 baud for fast development time. It also has auto-reset so no noodling with pins and reset button pressings.



Features:

Super mini dimension:23mm x48.3mm x 13mm

ESP8266 @ 80MHz with 3.3V logic/power

4MB of FLASH (32 MBit)

Built in WiFi 802.11 b/g/n

CP2104 USB-Serial converter onboard with 921600 max baudrate for speedy uploading

Auto-reset support for getting into bootload mode before firmware upload

Output all the GPIO of ESP-12F

With Reset button and Flash button

Usage:

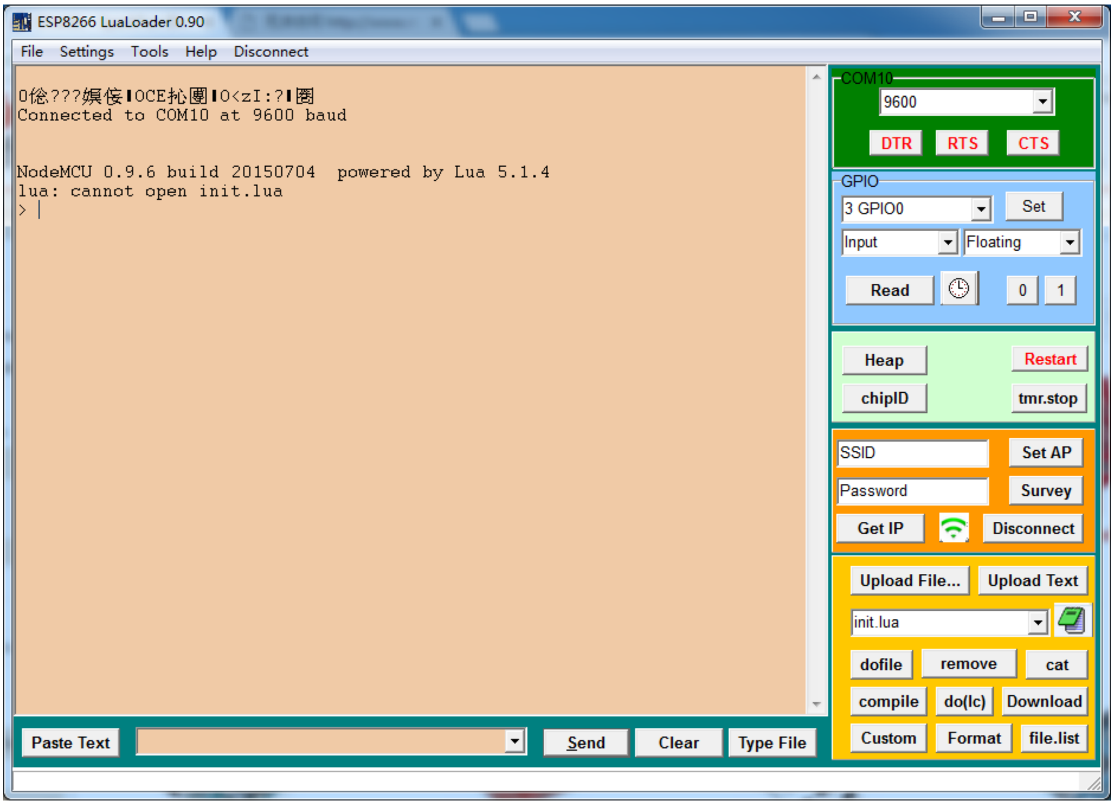
Using NodeMCU Lua

Each NodeMCU Mini comes pre-programmed with NodeMCU's Lua interpreter.

The Lua interpretter runs on the ESP8266 and you can type in commands and read out the results over serial. In order to upload code to the ESP8266 and use the serial console, connect any data-capable micro USB cable to the NodeMCU Mini and the other side to your computer's USB port. Install the required CP2104 USB driver to have the COM/Serial port appear properly:
https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

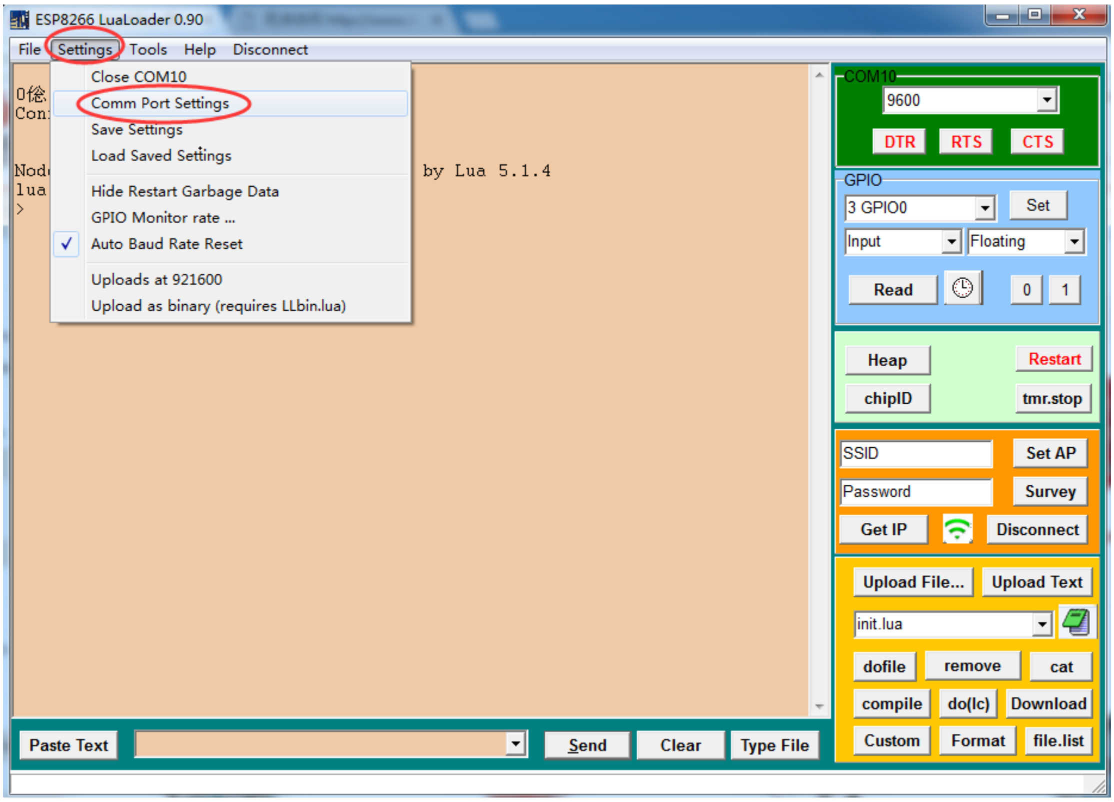
Download the Tools and firmware form our website:
http://www.makerfabs.com/fabs/index.php?route=product/product&path=90&product_id=127

1. Open up ESP8266 LuaLoader 0.90, show as below:

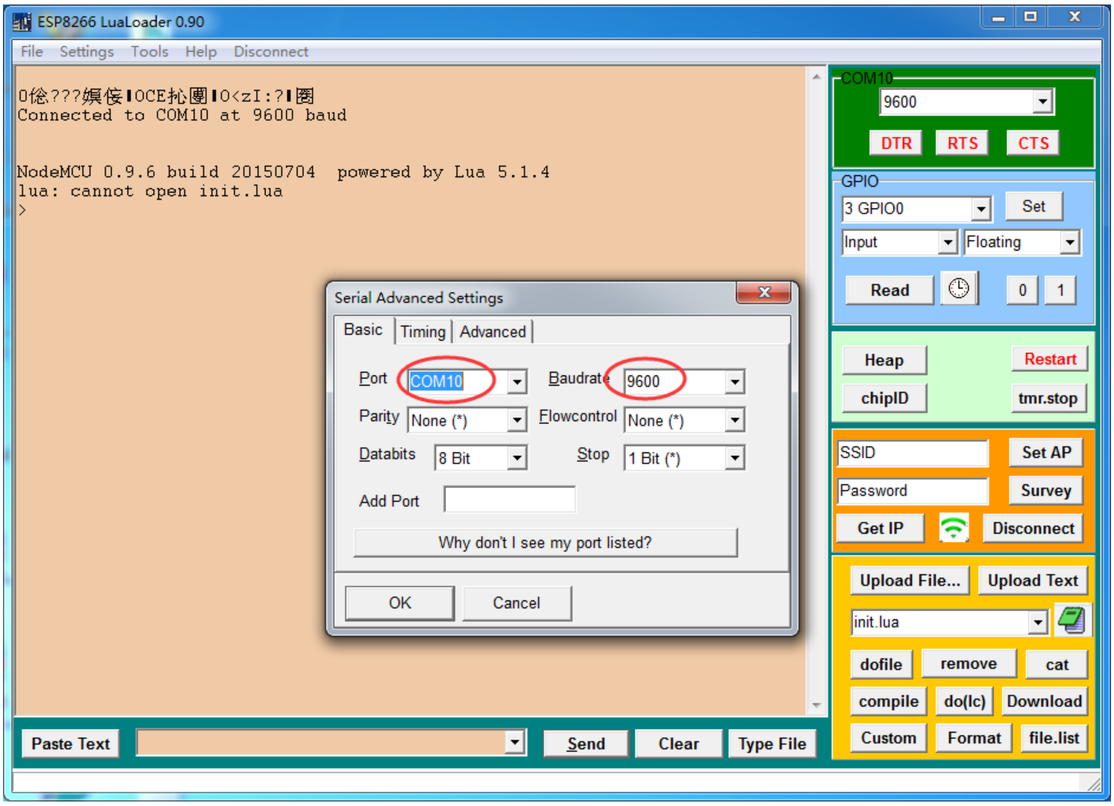


```
0俭???娯佞ǀOCE扡團ǀ0<zI:?ǀ图
Connected to COM10 at 9600 baud


NodeMCU 0.9.6 build 20150704  powered by Lua 5.1.4
lua: cannot open init.lua
> |
```

2. Set the com port:



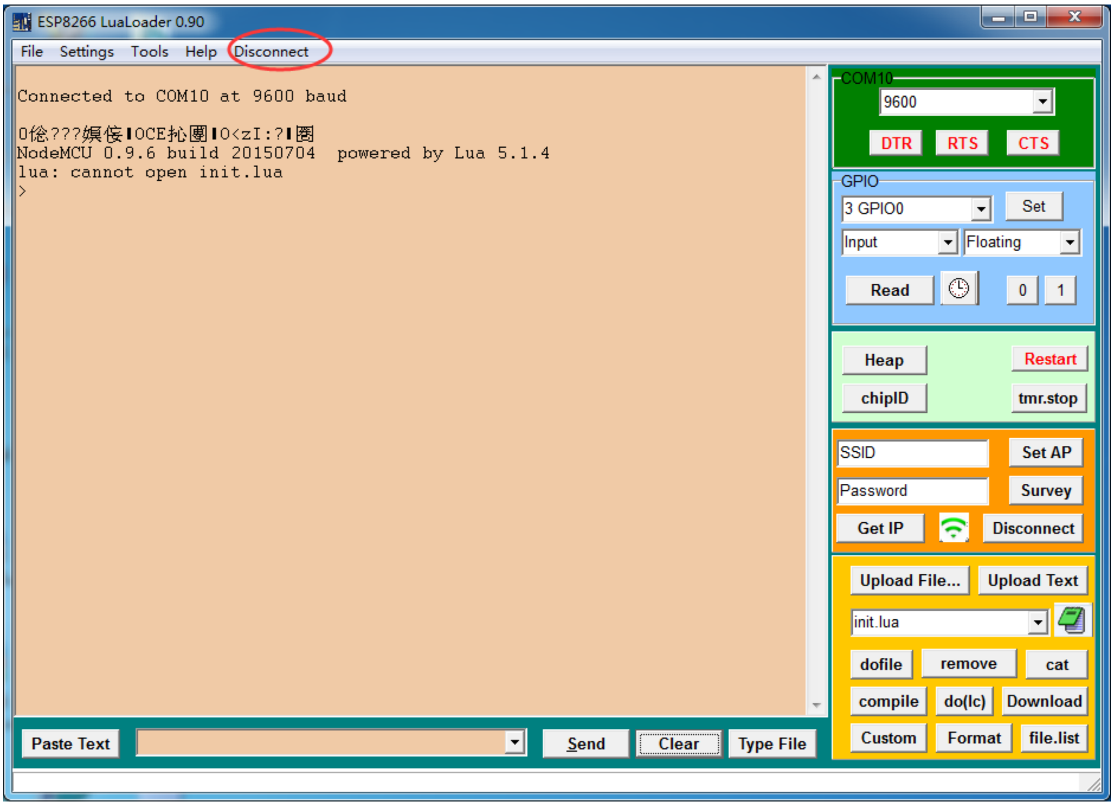
3. Select the right port about your NodeMCU:



```
0愴???緽佞ǀOCE扡團ǀ0<zI:?ǀ圏
Connected to COM10 at 9600 baud

NodeMCU 0.9.6 build 20150704  powered by Lua 5.1.4
lua: cannot open init.lua
>
```

Port: COM10
Baudrate: 9600
Parity: None (*)
Flowcontrol: None (*)
Databits: 8 Bit
Stop: 1 Bit (*)
Add Port:

4. Click the “connect”, it will connect the NodeMCU and print some information:



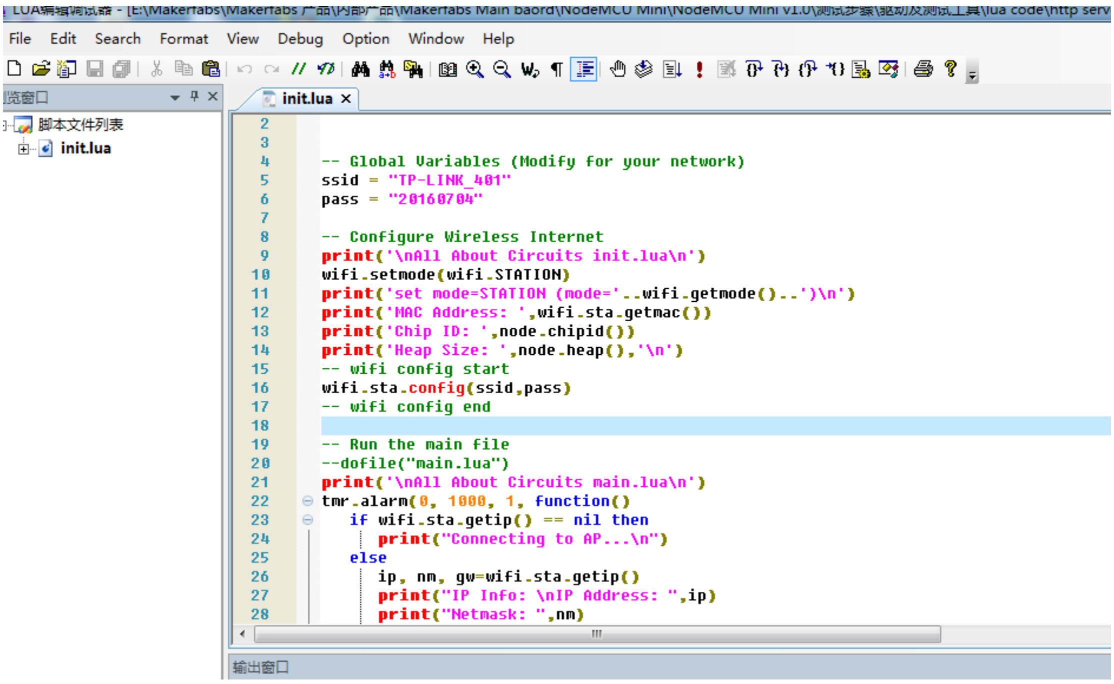
```
Connected to COM10 at 9600 baud

0倧???嫫佞▌OCE扡團▌O<zI:?▌圏
NodeMCU 0.9.6 build 20150704  powered by Lua 5.1.4
lua: cannot open init.lua
>
```

5. Write the code by any editor like luaEditor.

The demo code “http server” configured the NodeMCU mini as a server, when visit it, it will show you the “Hello, NodeMCU!!!”

With this demo code, you need to change the network of yours.

```lua
-- Global Variables (Modify for your network)
ssid = "TP-LINK_401"
pass = "20160704"
```

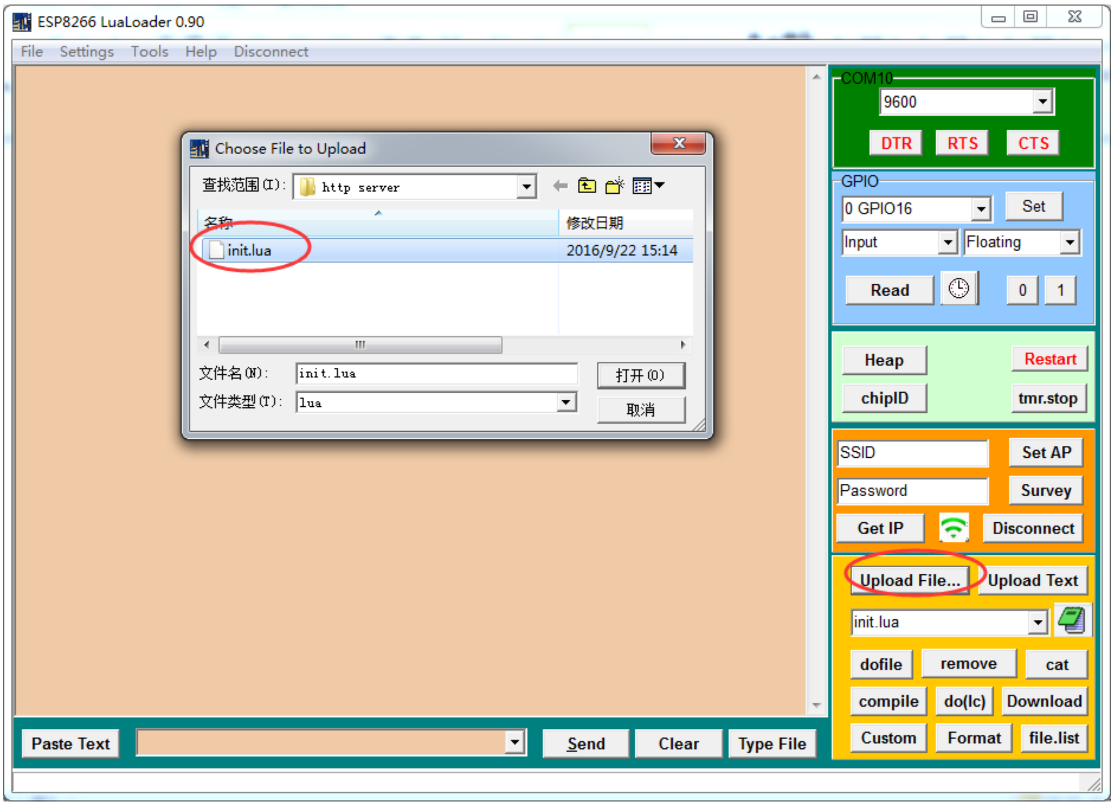


LUA编辑调试器 - [E:\Makerfabs\Makerfabs 产品\内部产品\Makerfabs Main baord\NodeMCU Mini\NodeMCU Mini v1.0\测试步骤\驱动及测试工具\lua code\http serv
File Edit Search Format View Debug Option Window Help
浏览窗口
脚本文件列表
init.lua
输出窗口

init.lua

```

-- Global Variables (Modify for your network)
ssid = "TP-LINK_401"
pass = "20160704"

-- Configure Wireless Internet
print('\nAll About Circuits init.lua\n')
wifi.setmode(wifi.STATION)
print('set mode=STATION (mode='..wifi.getmode()..')\n')
print('MAC Address: ',wifi.sta.getmac())
print('Chip ID: ',node.chipid())
print('Heap Size: ',node.heap(),'\n')
-- wifi config start
wifi.sta.config(ssid,pass)
-- wifi config end

-- Run the main file
--dofile("main.lua")
print('\nAll About Circuits main.lua\n')
tmr.alarm(0, 1000, 1, function()
   if wifi.sta.getip() == nil then
      print("Connecting to AP...\n")
   else
      ip, nm, gw=wifi.sta.getip()
      print("IP Info: \nIP Address: ",ip)
      print("Netmask: ",nm)
```

6. Upload the code



查找范围(I): http server

名称	修改日期
init.lua	2016/9/22 15:14

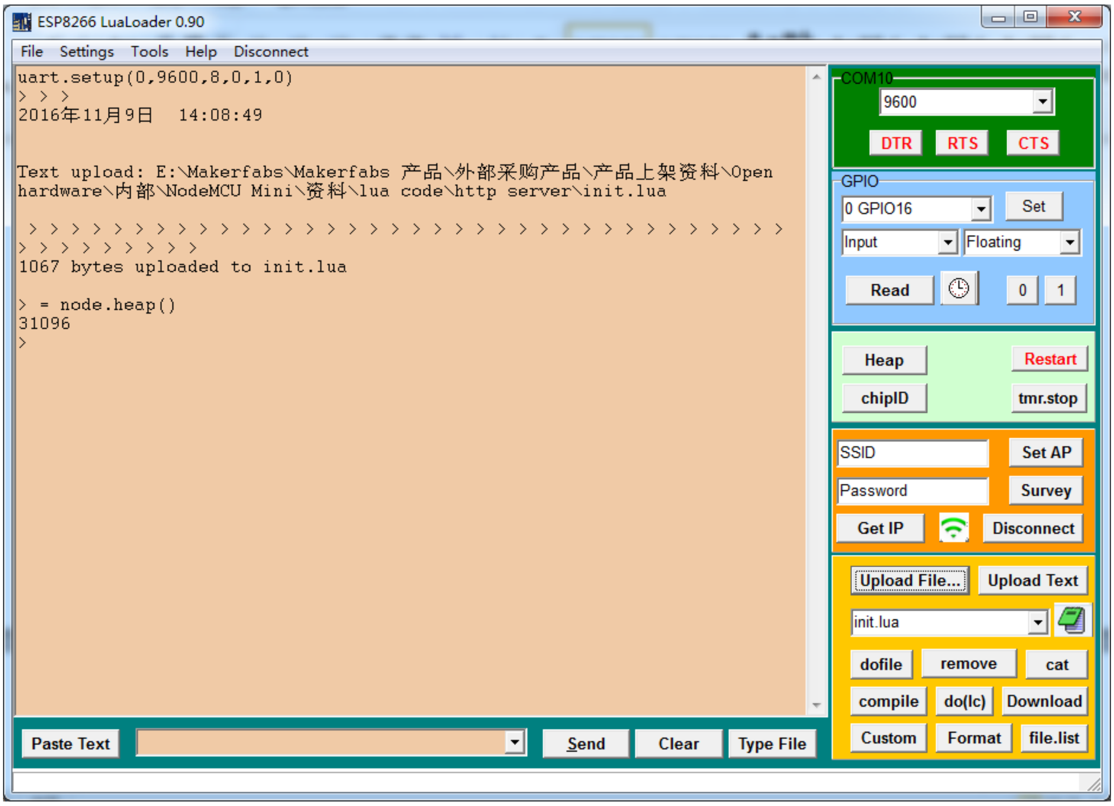
文件名(N): init.lua

文件类型(T): lua

COM10: 9600

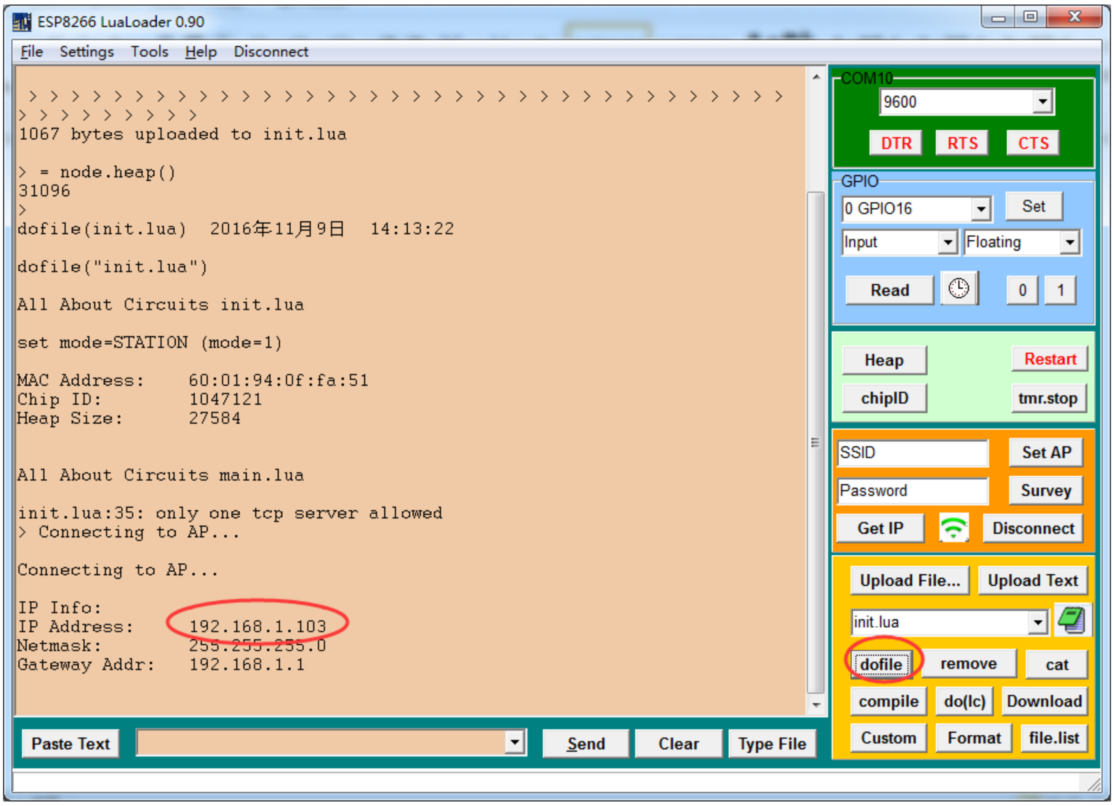
GPIO: 0 GPIO16, Input, Floating

init.lua



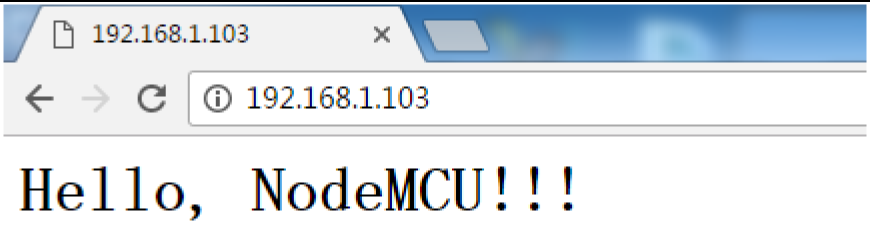
```
uart.setup(0,9600,8,0,1,0)
> > >
2016年11月9日  14:08:49


Text upload: E:\Makerfabs\Makerfabs 产品\外部采购产品\产品上架资料\Open
hardware\内部\NodeMCU Mini\资料\lua code\http server\init.lua

 > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > >
> > > > > > > > >
1067 bytes uploaded to init.lua

> = node.heap()
31096
>
```

7. dofile



```
 > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > > >
> > > > > > > > >
1067 bytes uploaded to init.lua

> = node.heap()
31096
>
dofile(init.lua)  2016年11月9日  14:13:22

dofile("init.lua")

All About Circuits init.lua

set mode=STATION (mode=1)

MAC Address:    60:01:94:0f:fa:51
Chip ID:        1047121
Heap Size:      27584


All About Circuits main.lua

init.lua:35: only one tcp server allowed
> Connecting to AP...

Connecting to AP...

IP Info:
IP Address:     192.168.1.103
Netmask:        255.255.255.0
Gateway Addr:   192.168.1.1
```

8. Open your browser and input the ip then enter. Show as below:



192.168.1.103

Hello, NodeMCU!!!

Using Arduino IDE

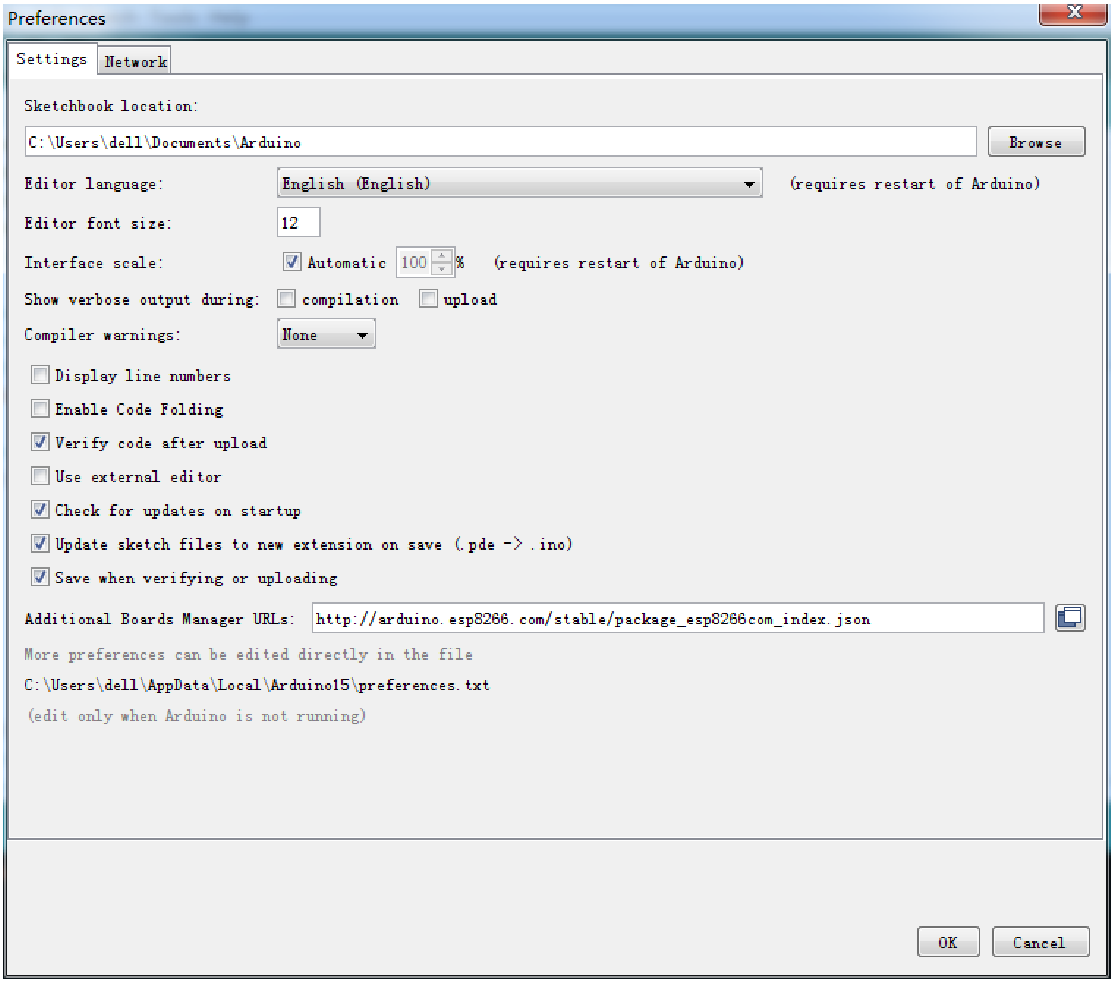
While the NodeMCU Mini comes pre-programmed with NodeMCU's Lua interpretter, you don't have to use it! Instead, you can use the Arduino IDE which may be more familar. This will write directly to the firmware, erasing the NodeMCU firmware, so if you want to go back to Lua, use the flasher to re-install it

Install the Arduino IDE 1.6.4 or greater

Download Arduino IDE from Arduino.cc (1.6.8 or greater) from Arduino.cc

1. Install the ESP8266 Board Package

Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field in the Arduino v1.6.4+ preferences.



Preferences

Sketchbook location:
C:\Users\dell\Documents\Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers

□ Enable Code Folding

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

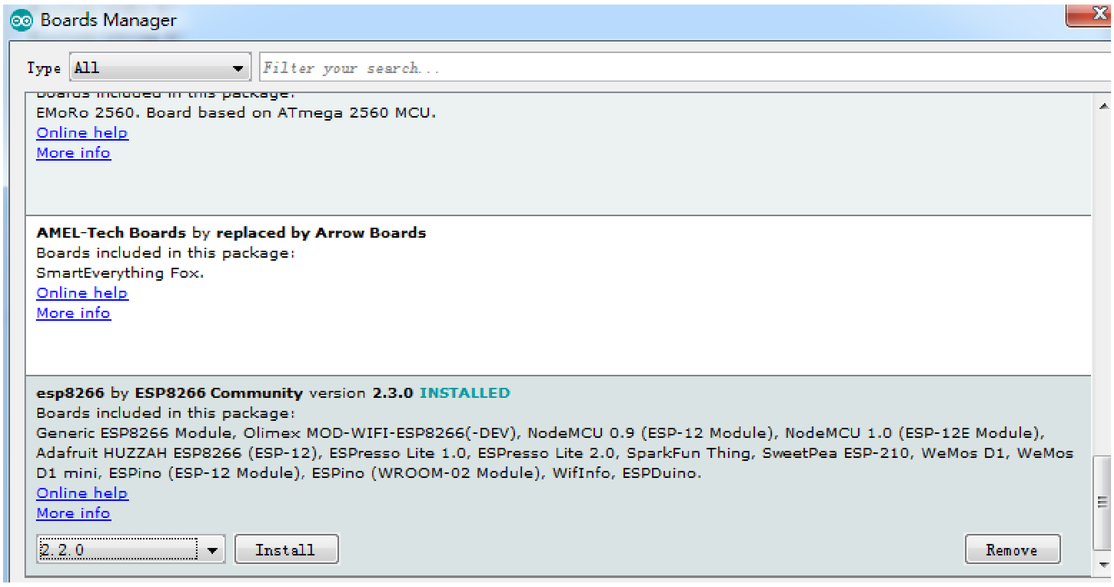
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

C:\Users\dell\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Next, use the **Board manager** to install the ESP8266 package.



Boards Manager

Type All
Filter your search...

Boards included in this package:
EMoRo 2560. Board based on ATmega 2560 MCU.
Online help
More info

AMEL-Tech Boards by **replaced by Arrow Boards**
Boards included in this package:
SmartEverything Fox.
Online help
More info

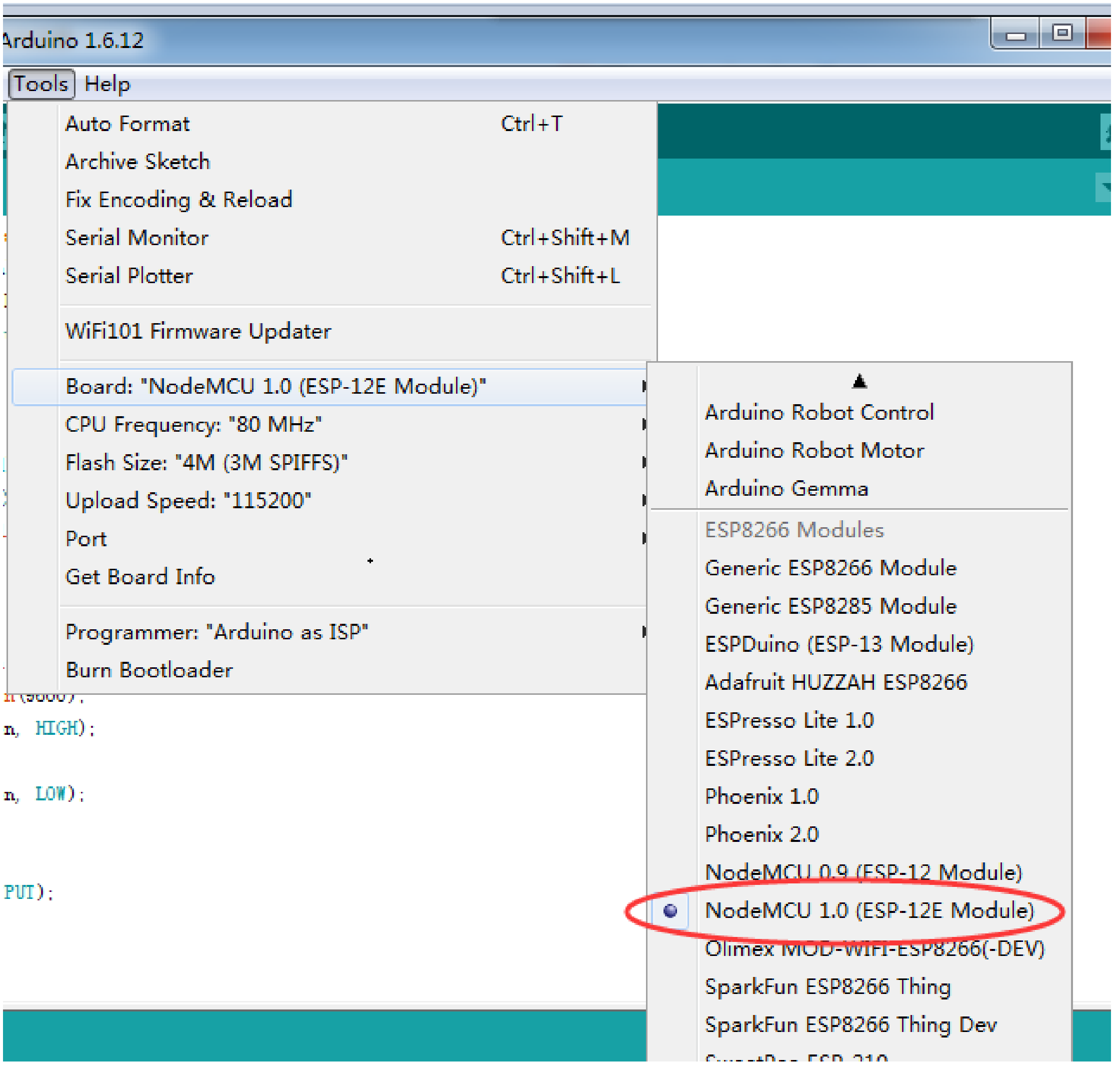
esp8266 by **ESP8266 Community** version **2.3.0** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

2.2.0

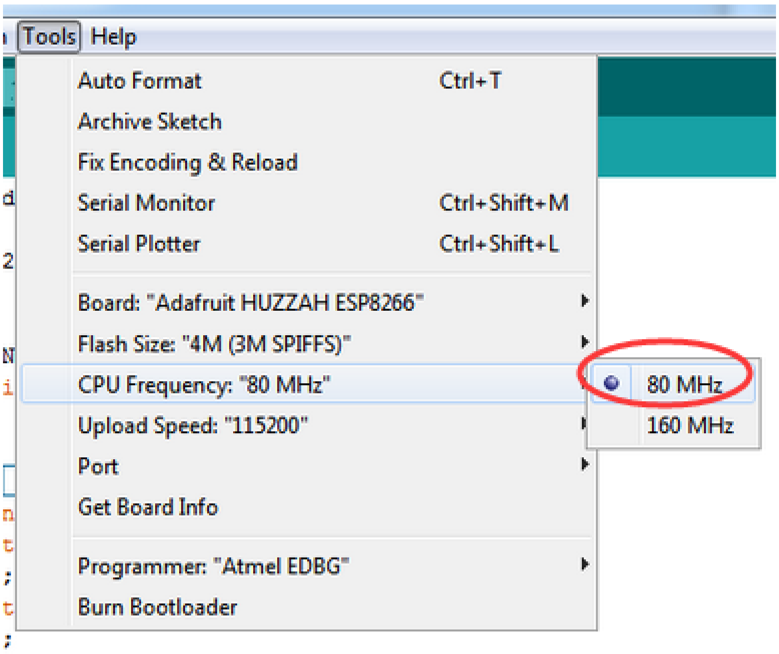
After the install process, you should see that esp8266 package is marked INSTALLED. Close the Boards Manager window once the install process has completed.

2. Setup ESP8266 Support

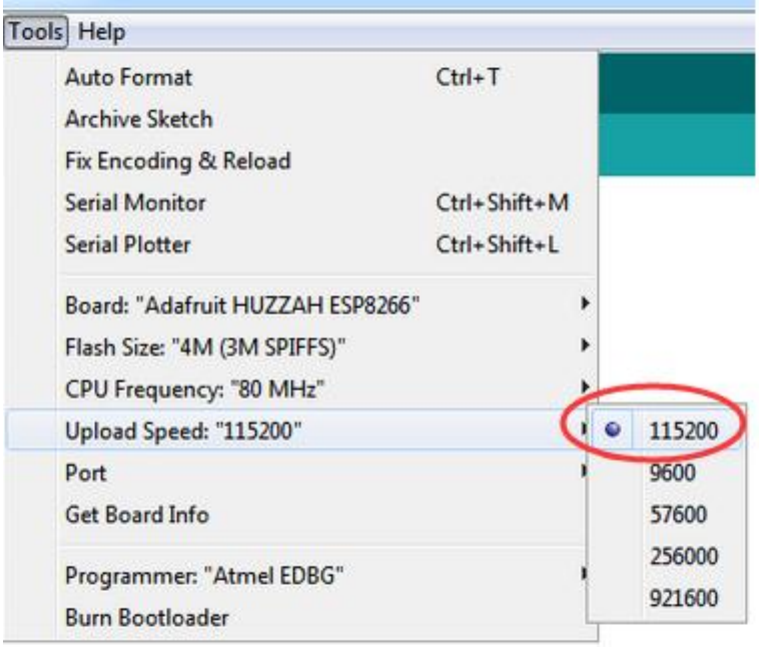
When you've restarted, select **NodeMCU 1.0** from the Tools->Board dropdown



80 MHz as the CPU frequency

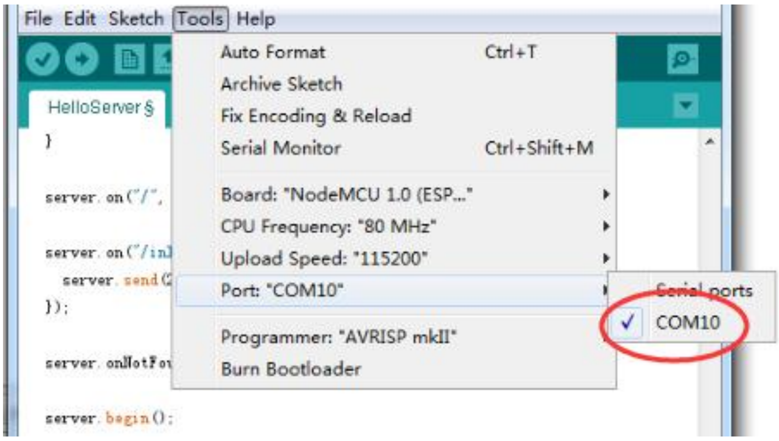


You can keep the **Flash Size**at "4M (3M SPIFFS),.For **Upload Speed**, select 115200 baud .



Board: "Adafruit HUZZAH ESP8266"
Flash Size: "4M (3M SPIFFS)"
CPU Frequency: "80 MHz"
Upload Speed: "115200"

The matching COM port for your FTDI or USB-Serial cable

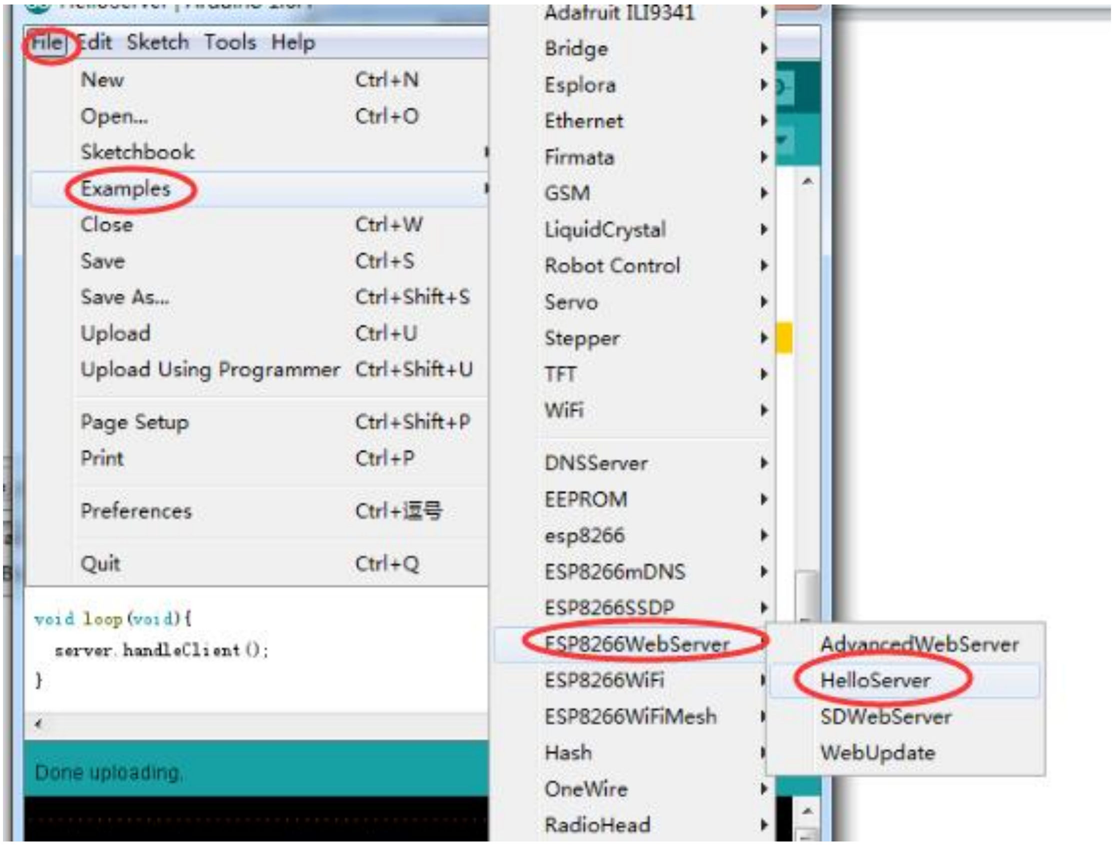


Board: "NodeMCU 1.0 (ESP..."
CPU Frequency: "80 MHz"
Upload Speed: "115200"
Port: "COM10"

```
}

server.on("/",

server.on("/inl
  server.send(2
});

server.onNotFou

server.begin();
```

HelloServer test:

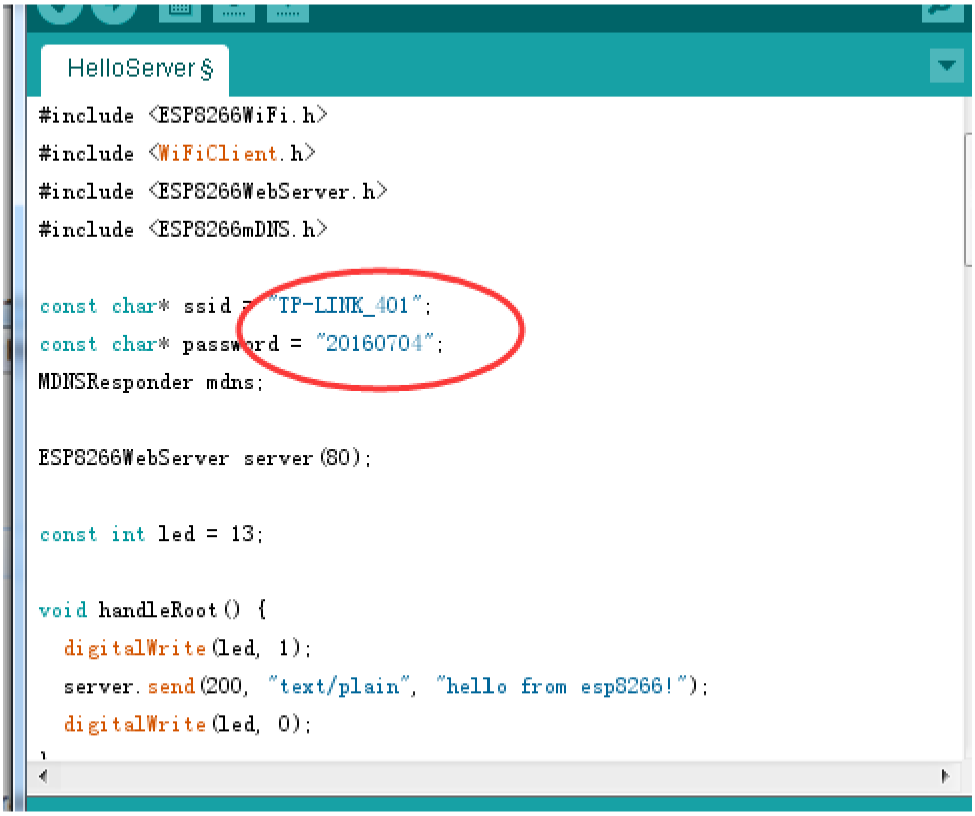
Connect the NodeMCU Mini to your PC via the micro USB cable, and open the demo. Show as below:



```
void loop(void){
  server.handleClient();
}
```

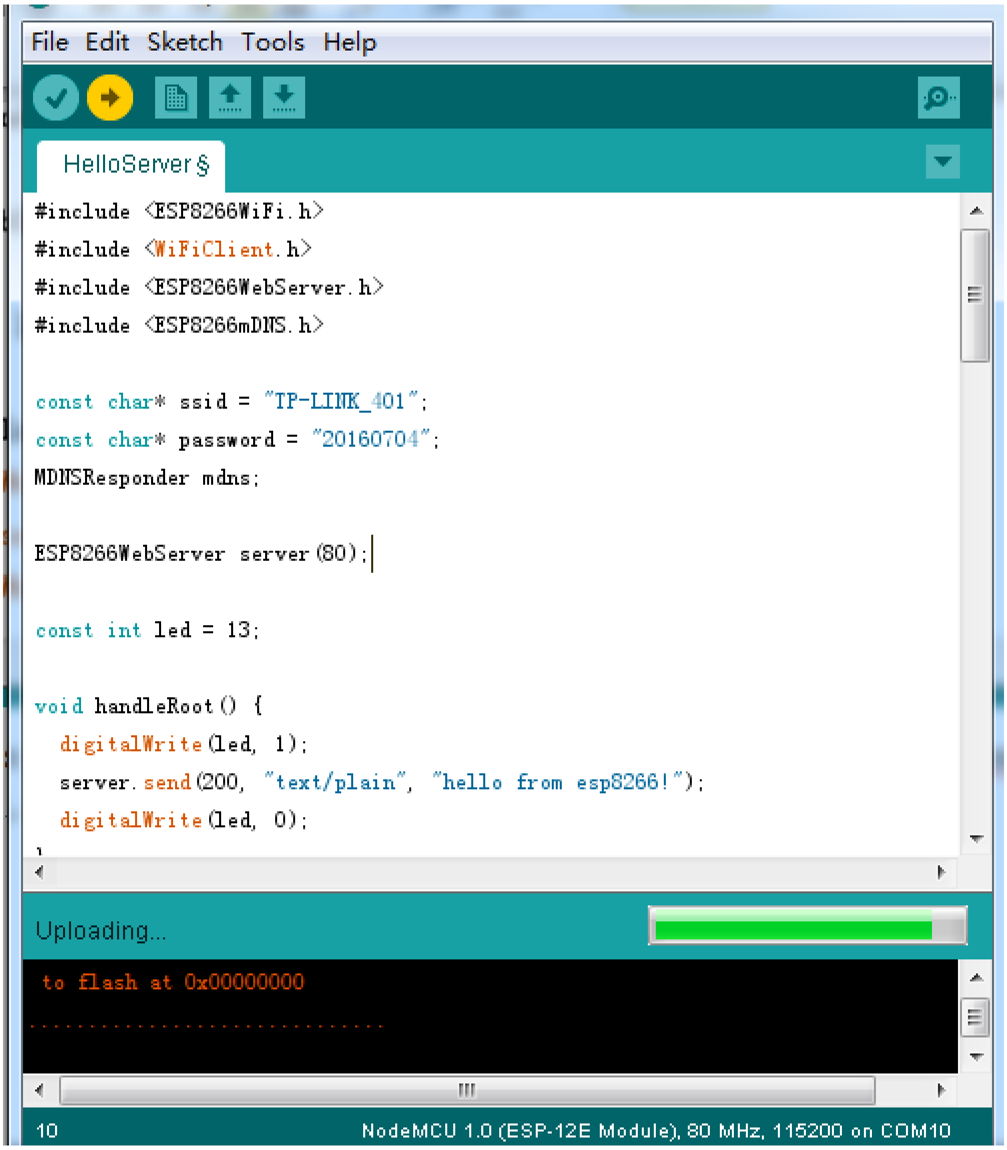
Done uploading.

Don’t forget to update you network



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "TP-LINK_401";
const char* password = "20160704";
MDNSResponder mdns;

ESP8266WebServer server(80);

const int led = 13;

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "hello from esp8266!");
  digitalWrite(led, 0);
}
```

to your access point and password, then upload the same way: get into bootload mode, then upload code via IDE

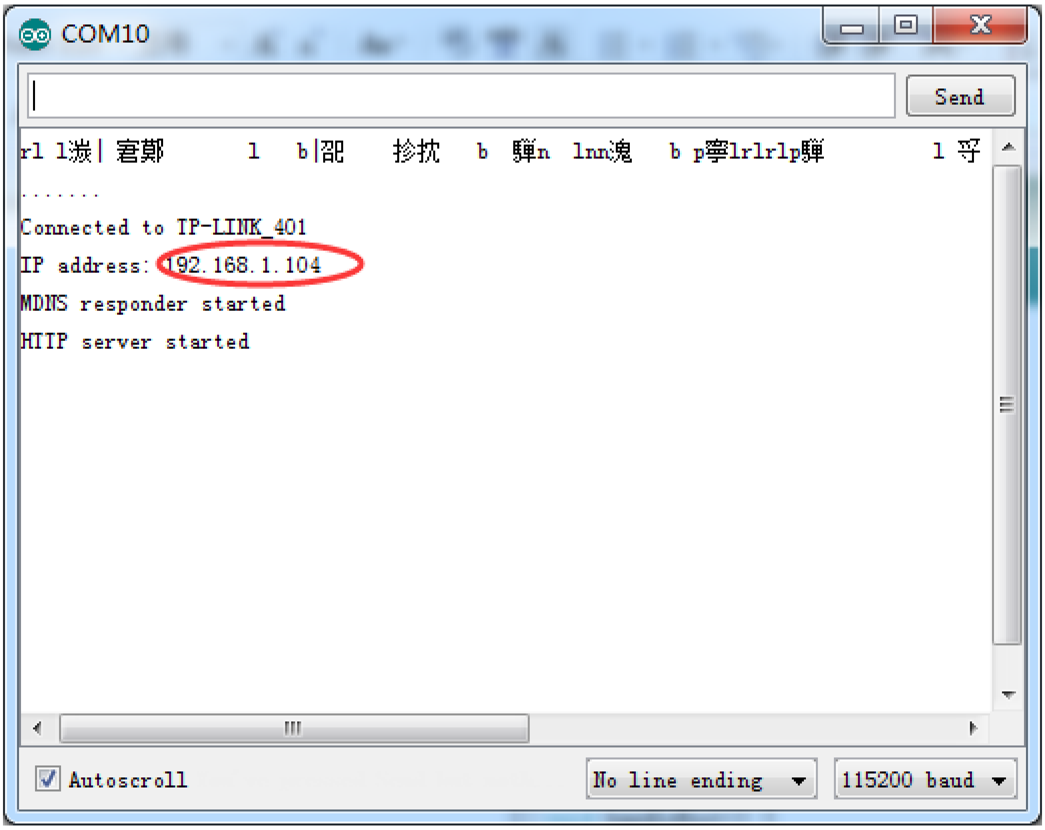


```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "TP-LINK_401";
const char* password = "20160704";
MDNSResponder mdns;

ESP8266WebServer server(80);

const int led = 13;

void handleRoot() {
  digitalWrite(led, 1);
  server.send(200, "text/plain", "hello from esp8266!");
  digitalWrite(led, 0);
}
```

```
 to flash at 0x00000000
..............................
```

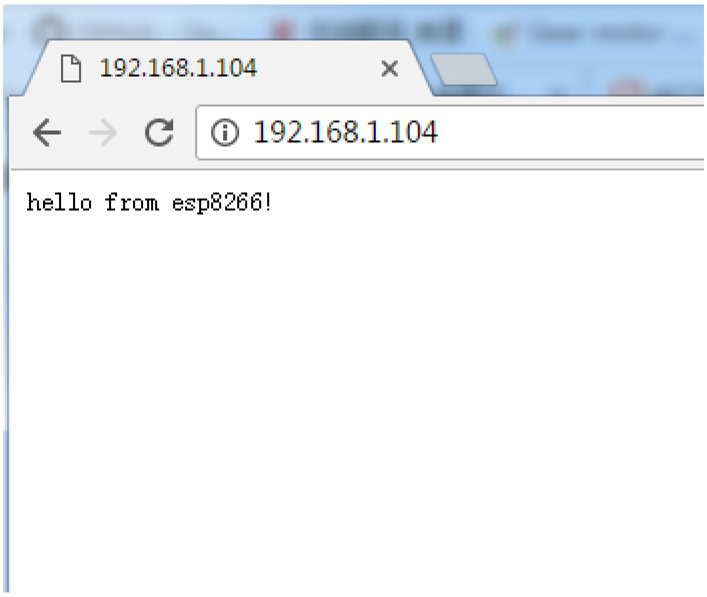
10 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200 on COM10

Open up the IDE serial console at 115200 baud to see the connection and Server IP printout!



```
rl 1渋| 言鄭      1   b|郘    拎抗   b  鞸n  lnn滬   b p寧lrlrlp鞸        1 孚
.......
Connected to TP-LINK_401
IP address: 192.168.1.104
MDNS responder started
HTTP server started
```

Open your browser and input the IP, show as below:



```
hello from esp8266!
```

That's it, pretty easy!