

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:



2. Set the com port:



ES ES	P8266 LuaLoader 0.90			
File	Settings Tools Help Disconnect			
0倊 Con:	Close COM10 Comm Port Settings Save Settings Load Saved Settings			RTS CTS
Nod lua >	Hide Restart Garbage Data GPIO Monitor rate V Auto Baud Rate Reset	by Lua 5.1.4	GPIO 3 GPIO0 Input	▼ Set
	Uploads at 921600 Upload as binary (requires LLbin.lua)		Read	0 1
			Heap chipID	Restart tmr.stop
			SSID	Set AP
			Password	Survey
			Get IP	Disconnect
			Upload	File Upload Text
			init.lua	J 🗐
			dofile	remove cat
			- compile	do(lc) Download
Pa	ste Text	▼ <u>S</u> end	Clear Type File Custom	Format file.list
				1

3. Select the right port about your NodeMCU:

ESP8266 LuaLoader 0.90	_ _ X
File Settings Tools Help Disconnect	
0倊???嫇侫∣OCE抋團∣O <zi:? 圐 Connected to COM10 at 9600 baud</zi:? 圐 	DTR RTS CTS
NodeMCU 0.9.6 build 20150704 powered by Lua 5.1.4 lua: cannot open init.lua >	GPIO 3 GPIO0 Set Input Floating
Serial Advanced Settings	Read () 0 1
Port COM 10 V Baudrat 9600 V Parity None (*) V Elowcontrol None (*) V	Heap Restart chipID tmr.stop
Databits 8 Bit ▼ Stop 1 Bit (*) ▼ Add Port	SSID Set AP Password Survey
Why don't I see my port listed?	Get IP 🕤 Disconnect
OK Cancel	init.lua
-	dofile remove cat compile do(lc) Download
Paste Text Send Clear Type File	Custom Format file.list

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





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.

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



LUA编辑响风菇 - [E:\Makertabs)	Viakertabs 广码\内部广码\Makertabs Main baord\NodeMCU Mini\NodeMCU Mini v1.0\测成步骤\驱动及测试工具\lua code\http serv
File Edit Search Format	View Debug Option Window Help
D 🚅 🖗 🛛 🖉 I X 🖻 🛍	🗠 🗠 // 1/ 🚧 🏥 🙀 🔯 🍳 🔍 ୩ 🎼 🖱 🕸 🗟 🎍 ! 🚿 🔂 ሽ ሽ ሽ ሽ 🕄 🖓 🎒 🦉 🍃
L版窗口	🗋 init.lua 🗙
🗆 🌄 脚本文件列表	2
🗄 🗹 🕑 init.lua	
	4 Global Variables (moolfy for your network) c ccid = "TDP.I TAW. hot".
	6 nass = "20160704"
	7
	8 Configure Wireless Internet
	<pre>9 print('\nAll About Circuits init.lua\n')</pre>
	10 wifi.setmode(wifi.STATION)
	11 print('Set mode=STATION (mode='wifi.getmode()')\n')
	12 print(nht Huress: ,Willstd.getMd())
	14 print('Heap Size: ',node.heap(),-'\n')
	15 wifi config start
	16 wifi_sta_config(ssid,pass)
	17 wifi config end
	19 Kun the main file
	20
	$22 \oplus tmr_{alarm(0, 1000, 1, function())}$
	23 ⊖ if wifi_sta_getip() == nil then
	24 print("Connecting to AP\n")
	25 else
	26 1p, nm, gw=wifi.sta.getip()
	28 print(1P inv: (ir Houress: ,1))
	<pre></pre>
	通上版工

6. Upload the code

ESP8266 LuaLoader 0.90	
File Settings Tools Help Disconnect	COM19 9600 JUR RTS CTS
查找范围 ①):	GPIO 0 GPIO16 Set Input Floating Read 0 1
<	Heap Restart chipID tmr.stop
	SSID Set AP Password Survey Get IP Sisconnect
	Upload File Upload Text init.lua
Y Send Clear Type File	donie remove cat compile do(lc) Download Custom Format file.list





7. dofile

ESP8266 LuaLoader 0.90	_					• X
<u>File</u> Settings Tools <u>H</u> elp Disconnect						
<pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	> > > > >	> > > >	> > >	-GPIO 0 GPIO16 Input Read	RTS Floating	▼ CTS Set g ▼ 0 1
All About circuits init.iuu						
set mode=STATION (mode=1) MAC Address: 60:01:94:0f:fa:51 Chip ID: 1047121 Heap Size: 27584				Heap chipID		Restart tmr.stop
All About Circuits main.lua init.lua:35: only one tcp server allowed > Connecting to AP			E	SSID Password Get IP	D is	Set AP Survey connect
Connecting to AP				Upload F	ile Upl	oad Text
IP Info: IP Address: 192.168.1.103 Netmask: 255.255.25 Gateway Addr: 192.168.1.1				init.lua dofile compile	remove do(lc) [cat
Paste Text	<u>S</u> end	Clear	Type File	Custom	Format	file.list

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



×

www.makerfabs.com

← → C 🛈 192.168.1.103

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



Preferences	X
Settings Network	
Sketchbook location:	
C:\Users\dell\Documents\Ardu	lino Browse
Editor language:	English (English) v (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 star	tup
📝 Update sketch files to ne	aw extension on save (.pde -> .ino)
Save when verifying or up	loading
Additional Boards Manager UR	Ls: http://arduino.esp8266.com/stable/package_esp8266com_index.json
More preferences can be edit	ed directly in the file
C:\Users\dell\AppData\Local\	Arduino15\preferences.txt
(edit only when Arduino is n	ot running)
	OK Cancel

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



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



Ardui	no 1.6.12			
Тоо	ls Help			
	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T		
	Serial Monitor Serial Plotter	Ctrl+Shift+M Ctrl+Shift+L		
	WiFi101 Firmware Updater			
	Board: "NodeMCU 1.0 (ESP-12E Module)" CPU Frequency: "80 MHz" Flash Size: "4M (3M SPIFFS)"			Arduino Robot Control Arduino Robot Motor Arduino Gemma
	Port Get Board Info			ESP8266 Modules Generic ESP8266 Module Generic ESP8285 Module
	Burn Bootloader			ESPDuino (ESP-13 Module) Adafruit HUZZAH ESP8266
n, HI	GH) :			ESPresso Lite 1.0 ESPresso Lite 2.0
n, 110	();			Phoenix 1.0 Phoenix 2.0 NodeMCL 0.9 (ESP. 12 Module)
?VT);		<	•	NodeMCU 1.0 (ESP-12 Module) Olimex MOD-WIFI-ESP8266(-DEV)
				SparkFun ESP8266 Thing SparkFun ESP8266 Thing Dev

80 MHz as the CPU frequency

Т	ools Help			
	Auto Format	Ctrl+T		
	Archive Sketch			
	Fix Encoding & Reload			
d	Serial Monitor	Ctrl+Shift+M		
2	Serial Plotter	Ctrl+Shift+L		
	Board: "Adafruit HUZZAH ESP8266"		×	
N	Flash Size: "4M (3M SPIFFS)"		-	
i	CPU Frequency: "80 MHz"		6	80 MHz
	Upload Speed: "115200"		1	160 MHz
	Port		•	
n	Get Board Info			
t ;	Programmer: "Atmel EDBG"		•	
t	Burn Bootloader			



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

Too	ls] Help				
	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T			
	Serial Monitor	Ctrl+Shift+N	Ν		
	Serial Plotter	Ctrl+Shift+L			
	Board: "Adafruit HUZZAH ESP8266"		•		
	Flash Size: "4M (3M SPIFFS)"		•		
	CPU Frequency: "80 MHz"		1	-	-
	Upload Speed: "115200"		Q	۲	115200
	Port		1		9600
	Get Board Info				57600
	Programmer: "Atmel EDBG"		,		256000 921600

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

HelloServer §	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T	<u>@</u>
3	Serial Monitor	Ctrl+Shift+M	*
server. on ("/",	Board: "NodeMCU 1.0 (ESP" CPU Frequency: "80 MHz"	*	
server. on ("/in]	Upload Speed: "115200"		
server. send C	Port: "COM10"		Ceriel p
server onNotFou	Programmer: "AVRISP mkII" Burn Bootloader	C	COM10

HelloServer test:

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



el Edit Sketch Tools Help	_	Adatruit ILI9341	1
New Open Sketchbook Examples Close Save Save Save As Upload Upload Using Programmer Page Setup	Ctrl+N Ctrl+O Ctrl+W Ctrl+S Ctrl+Shift+S Ctrl+Shift+S Ctrl+U Ctrl+Shift+U Ctrl+Shift+P	Bridge Esplora Ethernet Firmata GSM LiquidCrystal Robot Control Servo Stepper TFT WiFi	
Print Preferences Quit	Ctrl+P Ctrl+逗号 Ctrl+Q	DNSServer EEPROM esp8266 ESP8266mDNS	• • • • • • • • • • • • • • • • • • •
<pre>void loop(void) { server.handleClient();</pre>		ESP8266SSDP ESP8266WebServer	AdvancedWebServer
Done uploading.		ESP8266WiFi ESP8266WiFiMesh Hash OneWire	BDWebServer WebUpdate

Don't forget to update you network

	~
HelloServer§	•
#include <esp8266wifi.h></esp8266wifi.h>	
#include <wificlient.h></wificlient.h>	- [
#include <esp8266webserver.h></esp8266webserver.h>	
#include <esp8266mdns.h></esp8266mdns.h>	
const char* ssid 7"TP-LINK_401";	
const char* password = "20160704";	
MDHSResponder mdns;	
ESP8266WebServer server(80);	
const int led = 13;	
void handleRoot() {	
digitalWrite(led, 1);	
<pre>server.send(200, "text/plain", "hello from esp8266!");</pre>	
digitalWrite(led, 0);	
	Þ

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





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



Send rl 1溃 窘鄭 1 b 巶 抮抌 b 騨n lnn浼 b p寧lrlrlp騨 1 寽 ▲ Connected to IP-LINK_401 IP address: 192.168.1.104 MDNS responder started HIIP server started E
rl 1嵗 宕鄭 1 b)邵 珍抌 b 騨n lnn逸 b p寧lrlrlp騨 1 寽 ^ Connected to TP-LINK_401 IP address: 192.168.1.104 MDNS responder started HTIP server started
Connected to IP-LINK_401 IP address: 192.168.1.104 MDNS responder started HIIP server started
IP address: 192.168.1.104 MDNS responder started HIIP server started E
MDNS responder started HTIP server started
HIIP server started
E
4
✓ Autoscroll No line ending ▼ 115200 baud ▼

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



That's it, pretty easy!