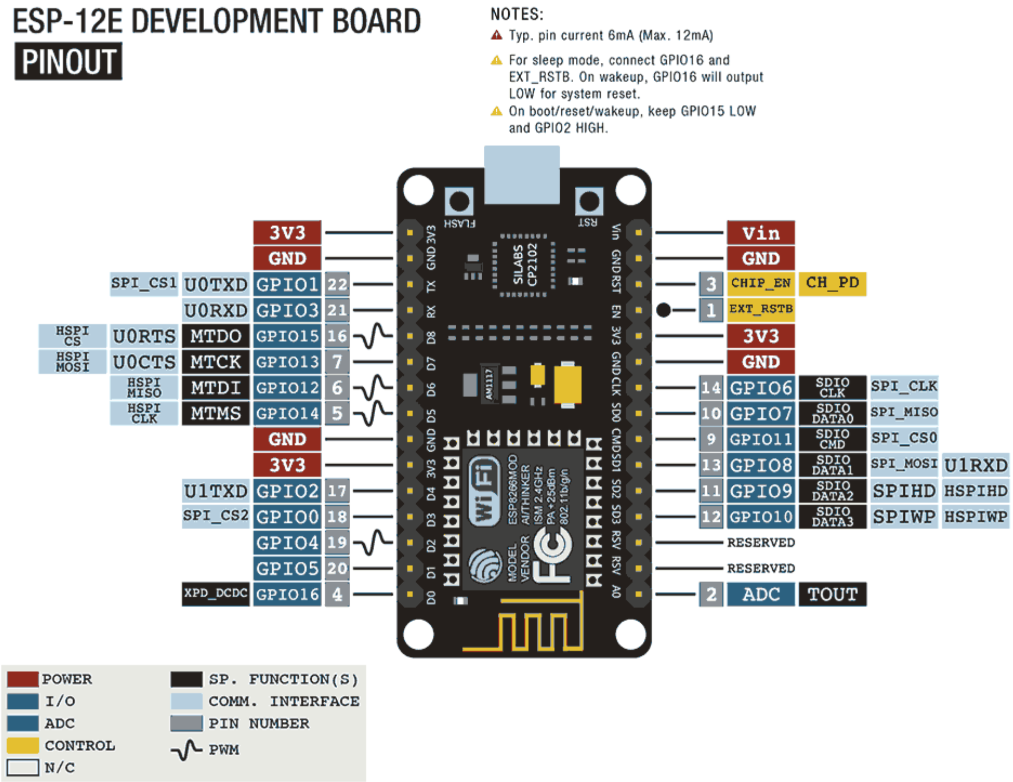
ESP8266 NodeMCU+ Electromagnetic relay for a smple IoT control

I、The function

The development environment is Arduino IDE， ESP8266wifi Module， Combined with APP (Blynk) to realize the establishment of a simple IoT control foundation. What needs to be explained here is that after operating according to this document, you can control the on/off of the relay through the Blynk APP to provide ideas for what additional functions you want to achieve in the future. The specific details can be seen in the following graphic

II、Circuit diagram

A、Module schematic

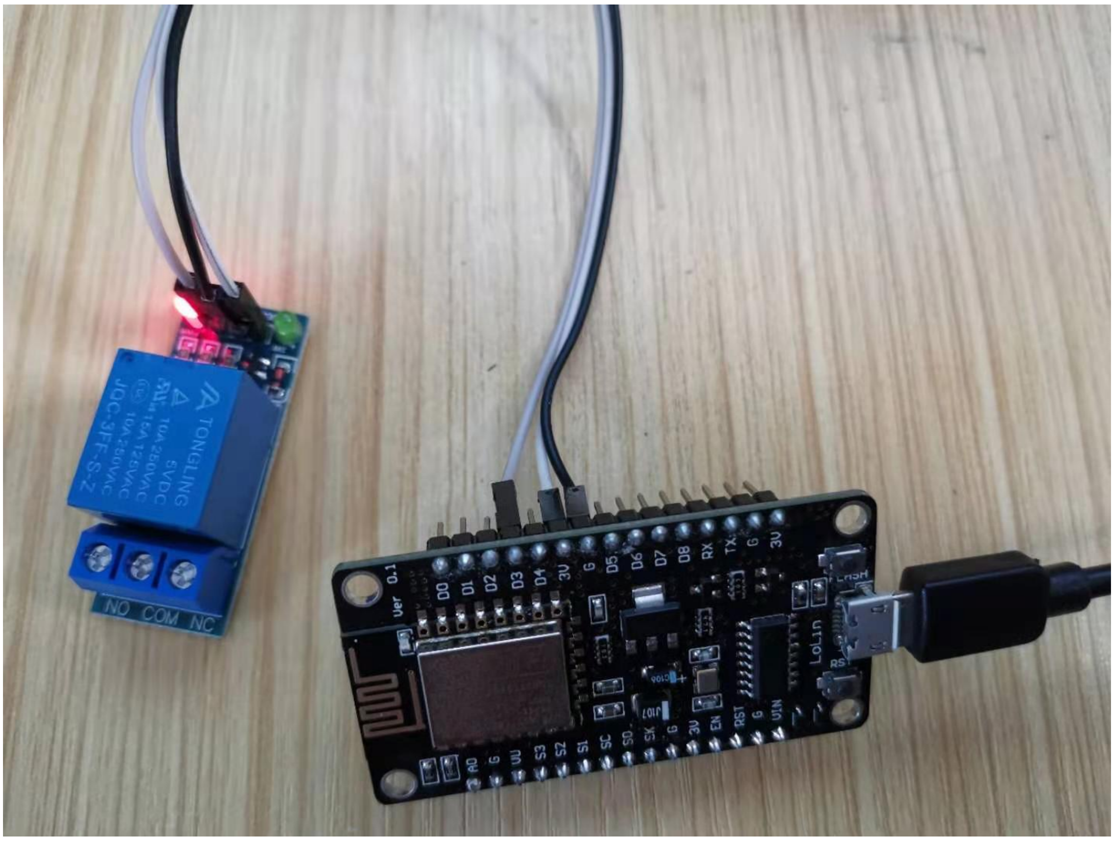


ESP-12E DEVELOPMENT BOARD
PINOUT
3V3
GND
SPI_CS1 U0TXD GPIO1 22
U0RXD GPIO3 21
HSPI CS U0RTS MTDO GPIO15 16
HSPI MOSI U0CTS MTCK GPIO13 7
HSPI MISO MTDI GPIO12 6
HSPI CLK MTMS GPIO14 5
GND
3V3
U1TXD GPIO2 17
SPI_CS2 GPIO0 18
GPIO4 19
GPIO5 20
XPD_DCDC GPIO16 4
Vin
GND
3 CHIP_EN CH_PD
1 EXT_RSTB
3V3
GND
14 GPIO6 SDIO CLK SPI_CLK
10 GPIO7 SDIO DATA0 SPI_MISO
9 GPIO11 SDIO CMD SPI_CS0
13 GPIO8 SDIO DATA1 SPI_MOSI U1RXD
11 GPIO9 SDIO DATA2 SPIHD HSPIHD
12 GPIO10 SDIO DATA3 SPIWP HSPIWP
RESERVED
RESERVED
2 ADC TOUT
FLASH
RST
SILABS CP2102
AM1117
WiFi
ESP8266MOD
AI-THINKER
FCC
3V3 GND TX RX D8 D7 D6 D5 GND 3V3 D4 D3 D2 D1 D0
Vin GND RST EN 3V3 GND CLK SD0 CMD SD1 SD2 SD3 RSV RSV A0
POWER
I/O
ADC
CONTROL
N/C
SP. FUNCTION(S)
COMM. INTERFACE
PIN NUMBER
PWM

NOTES:

- Typ. pin current 6mA (Max. 12mA)
- For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.

B、Physical connection diagram



III、 Development environment

（1）Arduino IDE 1.8.7 download

Official website download address：https://www.arduino.cn/thread-5838-1-1.html

Arduino community：https://www.arduino.cn/forum-68-1.html

1. Download for windows

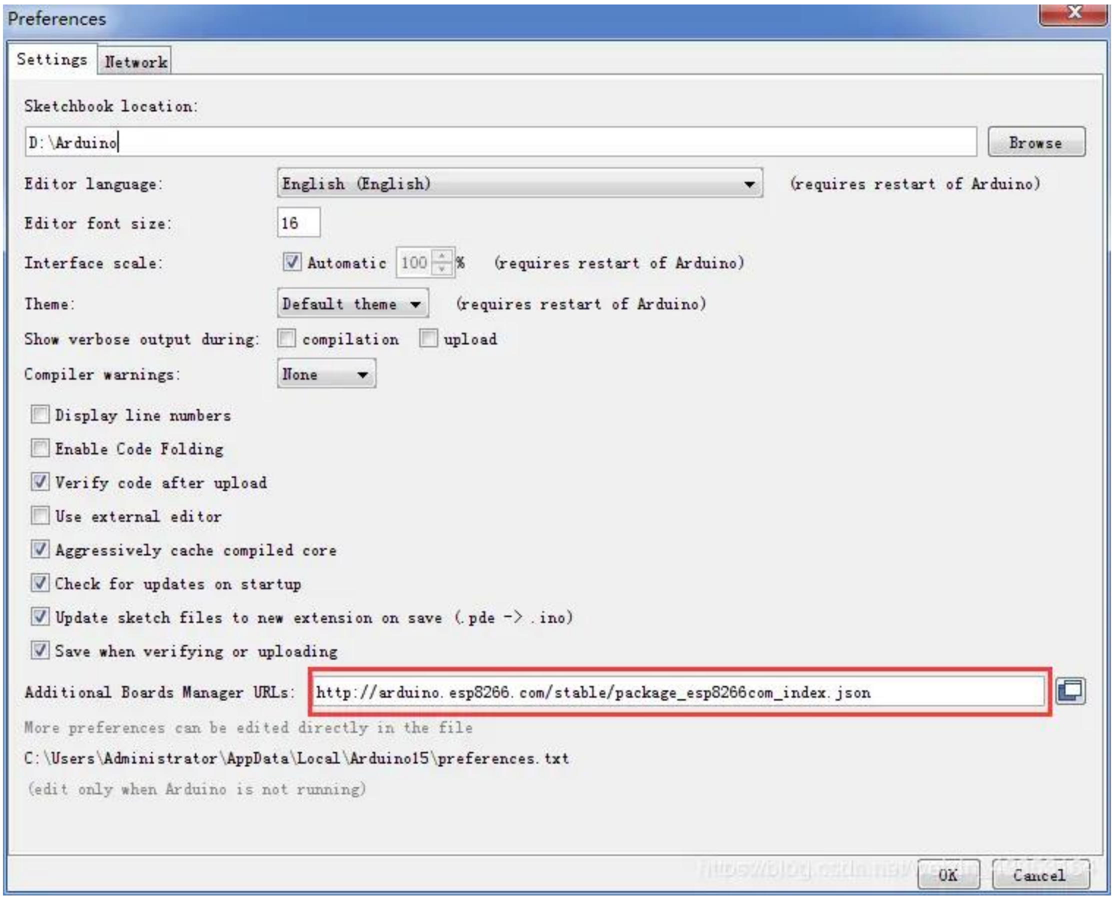
https://coding.net/u/coloz/p/arduino-installer/git/raw/master/1.8.7/arduino-1.8.7-windows.exe

2. Download for mac os

https://coding.net/u/coloz/p/arduino-installer/git/raw/master/1.8.7/arduino-1.8.7-macosx.zip

（2）Add ESP8266 development board

https://coding.net/u/coloz/p/arduinoinstaller/git/raw/master/1.8.7/arduino-1.8.7-macosx.zip



Sketchbook location:

D:\Arduino

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

Editor font size: 16

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

Theme: Default theme (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

☑ Aggressively cache compiled core

☑ 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

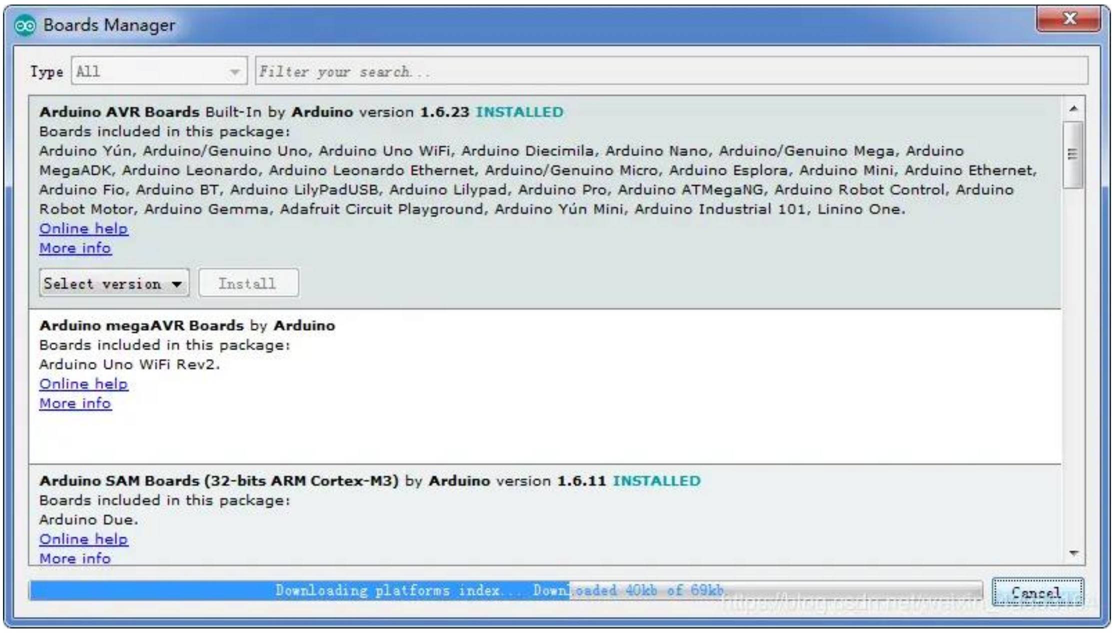
More preferences can be edited directly in the file

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

(edit only when Arduino is not running)

http://arduino.esp8266.com/stable/package_esp8266com_index.json

（3）Download ESP8266 development board

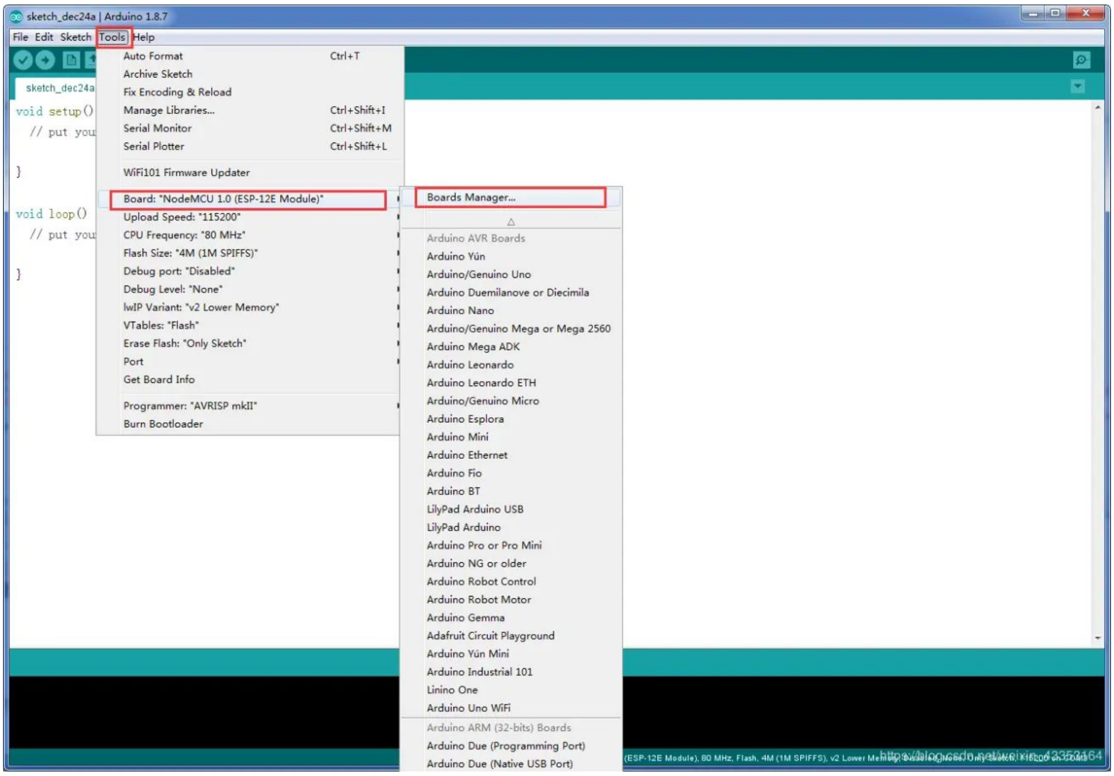


Arduino AVR Boards Built-In by **Arduino** version **1.6.23** **INSTALLED**
Boards included in this package:
Arduino Yún, Arduino/Genuino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino/Genuino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino/Genuino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.
Online help
More info

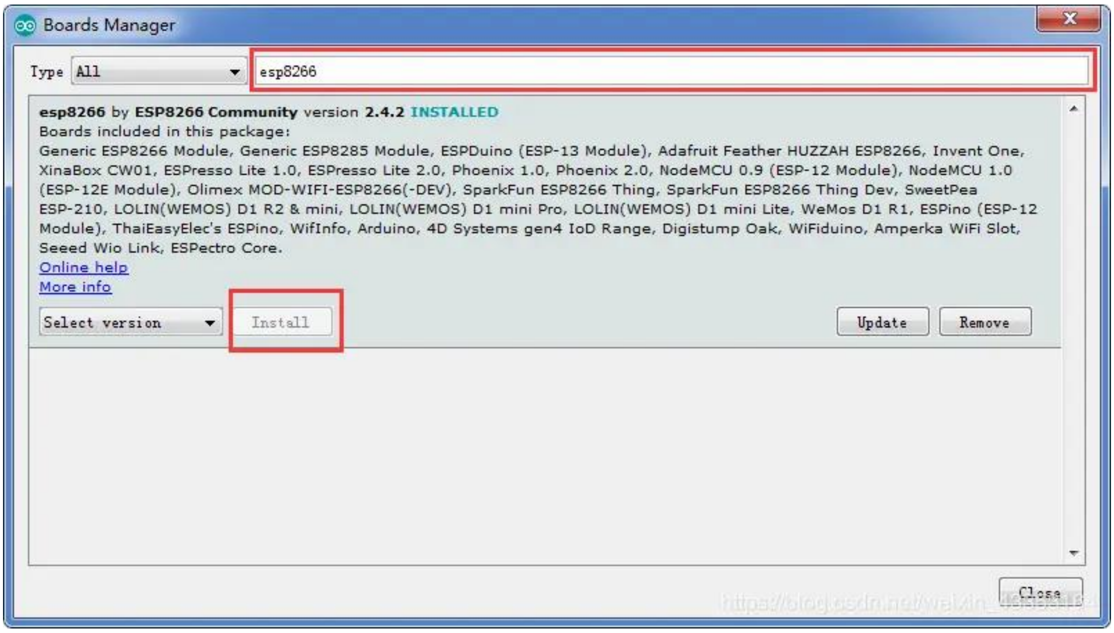
Arduino megaAVR Boards by **Arduino**
Boards included in this package:
Arduino Uno WiFi Rev2.
Online help
More info

Arduino SAM Boards (32-bits ARM Cortex-M3) by **Arduino** version **1.6.11** **INSTALLED**
Boards included in this package:
Arduino Due.
Online help
More info

Downloading platforms index... Downloaded 40kb of 69kb.

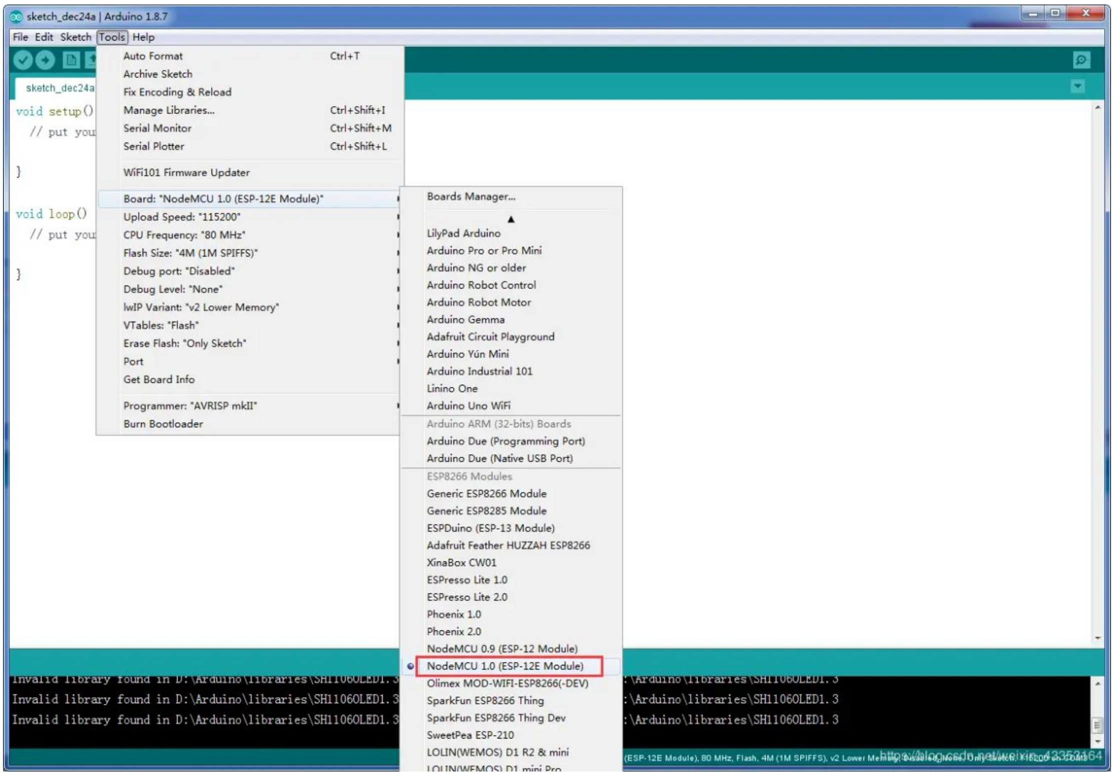


The dialog box searches for esp8266, selects the version, and downloads (the first download will be very slow, please be patient, if the download fails in the middle, just click to download again)



esp8266 by **ESP8266 Community** version **2.4.2** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

After downloading, set the download board. If the list shown in the figure below does not appear, the download has failed. Download again.



```
void setup() {
  // put you

}

void loop() {
  // put you

}
```

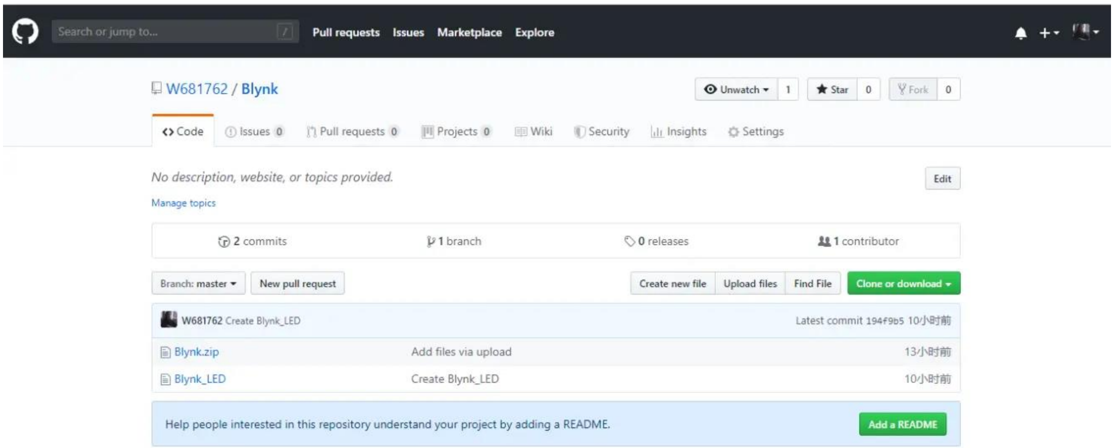
Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4M (1M SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Erase Flash: "Only Sketch"
Programmer: "AVRISP mkII"

```
Invalid library found in D:\Arduino\libraries\SH1106OLED1.3
Invalid library found in D:\Arduino\libraries\SH1106OLED1.3
Invalid library found in D:\Arduino\libraries\SH1106OLED1.3
```

(ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

（4）Download and add (Blynk) library file

Method 1: Open the link to download https://github.com/W681762/Blynk



Search or jump to...
Pull requests Issues Marketplace Explore
W681762 / Blynk
Unwatch 1 Star 0 Fork 0
Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings
Edit
Manage topics
Branch: master New pull request
Create new file Upload files Find File Clone or download
Add a README

No description, website, or topics provided.

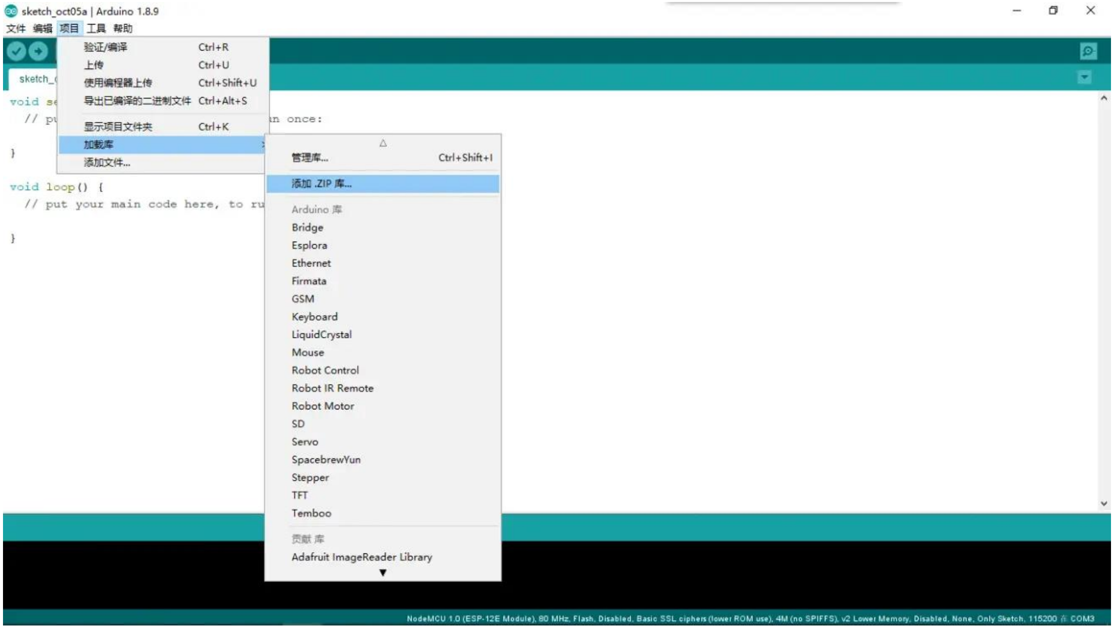
2 commits 1 branch 0 releases 1 contributor

W681762 Create Blynk_LED — Latest commit 194f9b5 10小时前

Blynk.zip	Add files via upload	13小时前
Blynk_LED	Create Blynk_LED	10小时前

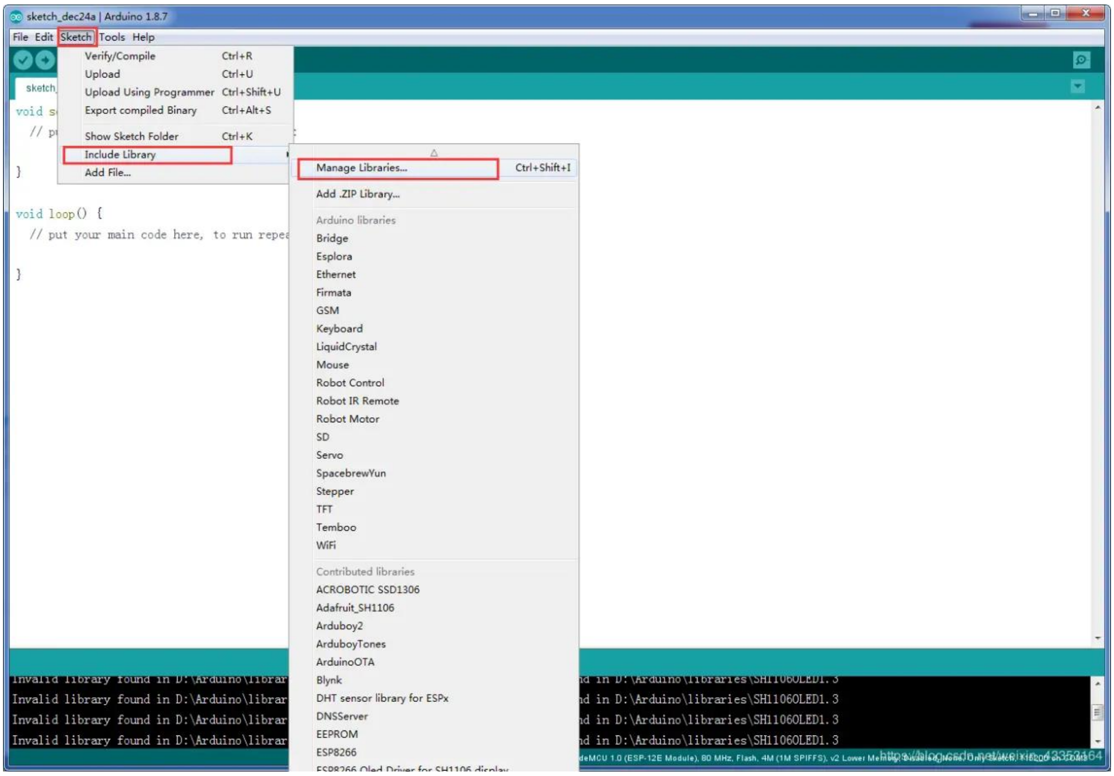
Help people interested in this repository understand your project by adding a README.

After the download is complete, get the compressed package of Blynk, choose “Add ZIP library”


sketch_oct05a | Arduino 1.8.9
文件 编辑 项目 工具 帮助
验证/编译 Ctrl+R
上传 Ctrl+U
使用编程器上传 Ctrl+Shift+U
导出已编译的二进制文件 Ctrl+Alt+S
显示项目文件夹 Ctrl+K
加载库
添加文件...
管理库... Ctrl+Shift+I
添加 .ZIP 库...
Arduino 库
Bridge
Esplora
Ethernet
Firmata
GSM
Keyboard
LiquidCrystal
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
Servo
SpacebrewYun
Stepper
TFT
Temboo
贡献 库
Adafruit ImageReader Library
sketch_
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, Basic SSL ciphers (lower ROM use), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 在 COM3

```
void se
  // pu                                            n once:

}

void loop() {
  // put your main code here, to ru

}
```

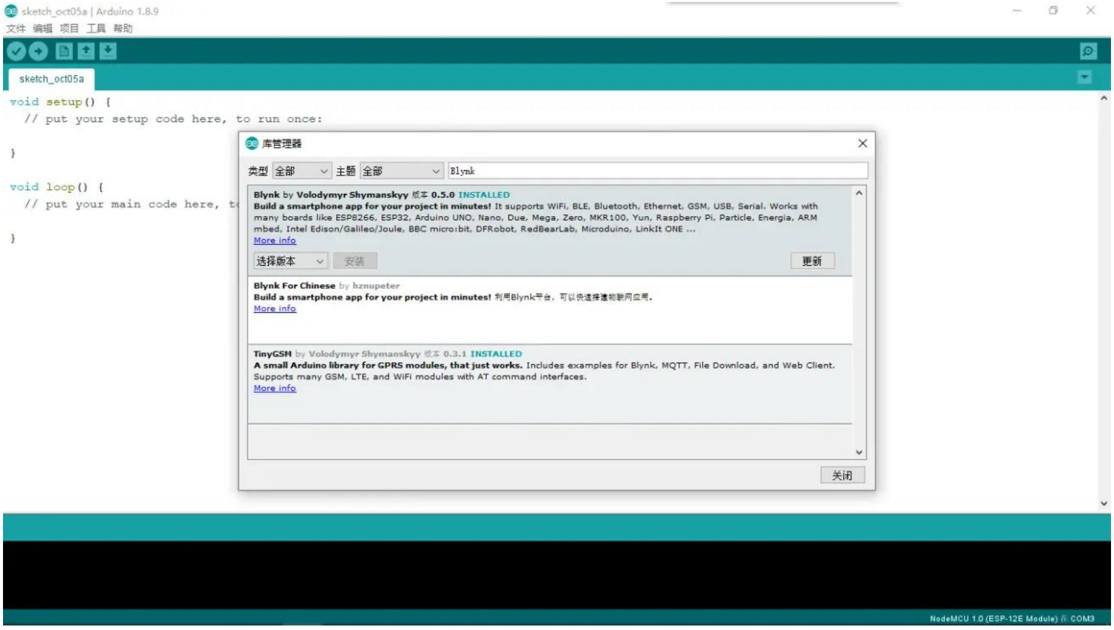
Method 2: Arduino searches for Blynk library files and adds


sketch_dec24a | Arduino 1.8.7
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Bridge
Esplora
Ethernet
Firmata
GSM
Keyboard
LiquidCrystal
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
Servo
SpacebrewYun
Stepper
TFT
Temboo
WiFi
Contributed libraries
ACROBOTIC SSD1306
Adafruit_SH1106
Arduboy2
ArduboyTones
ArduinoOTA
Blynk
DHT sensor library for ESPx
DNSServer
EEPROM
ESP8266
ESP8266 Oled Driver for SH1106 display
sketch_
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
https://blog.csdn.net/weixin_43351164

```
void s
  // pu

}

void loop() {
  // put your main code here, to run repea

}
```

```
Invalid library found in D:\Arduino\librar                              d in D:\Arduino\libraries\SH1106OLED1.3
Invalid library found in D:\Arduino\librar                              d in D:\Arduino\libraries\SH1106OLED1.3
Invalid library found in D:\Arduino\librar                              d in D:\Arduino\libraries\SH1106OLED1.3
Invalid library found in D:\Arduino\librar                              d in D:\Arduino\libraries\SH1106OLED1.3
```

Search Blynk in the dialog box, click to download



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, t
}
```

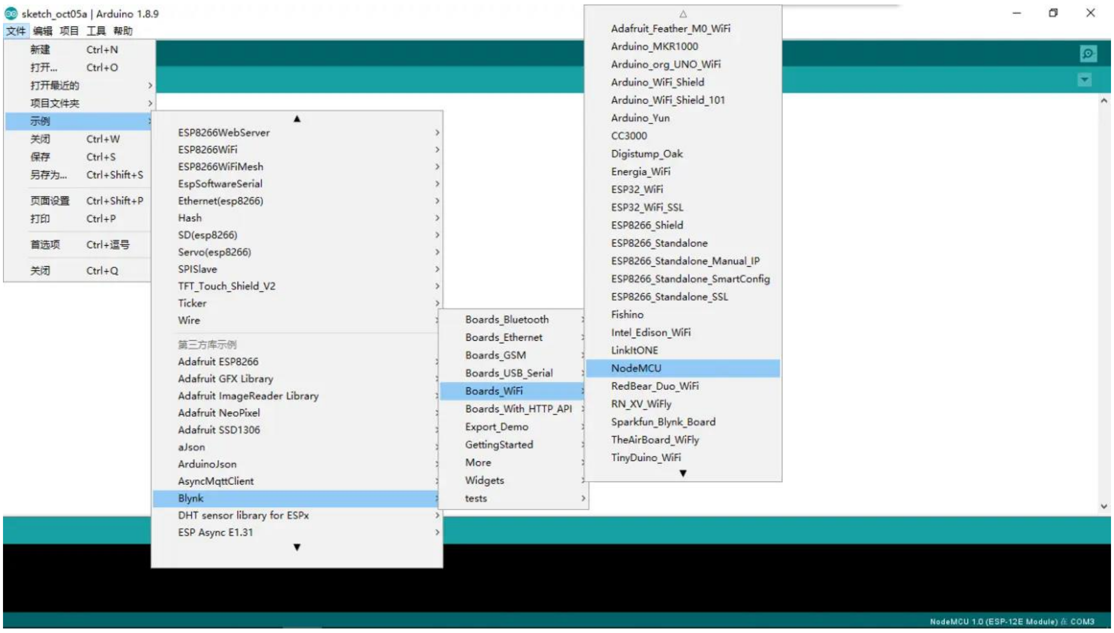
Blynk by **Volodymyr Shymanskyy** 版本 **0.5.0** INSTALLED
Build a smartphone app for your project in minutes! It supports WiFi, BLE, Bluetooth, Ethernet, GSM, USB, Serial. Works with many boards like ESP8266, ESP32, Arduino UNO, Nano, Due, Mega, Zero, MKR100, Yun, Raspberry Pi, Particle, Energia, ARM mbed, Intel Edison/Galileo/Joule, BBC micro:bit, DFRobot, RedBearLab, Microduino, LinkIt ONE ...
More info

Blynk For Chinese by hznupeter
Build a smartphone app for your project in minutes! 利用Blynk平台，可以快速搭建物联网应用。
More info

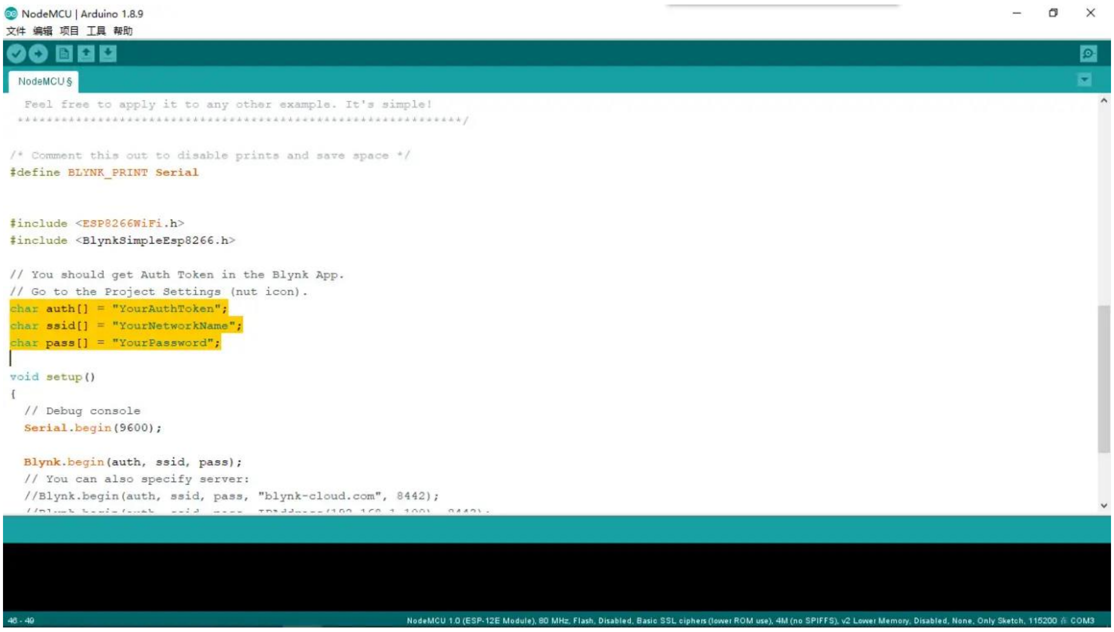
TinyGSM by Volodymyr Shymanskyy 版本 0.3.1 INSTALLED
A small Arduino library for GPRS modules, that just works. Includes examples for Blynk, MQTT, File Download, and Web Client. Supports many GSM, LTE, and WiFi modules with AT command interfaces.
More info

If the above process goes well, the development environment is now complete. Generally speaking, you may encounter various unexpected problems. Check the cause of the problem in the monitoring window at the bottom.

（5）Open the sample program in the Blynk library



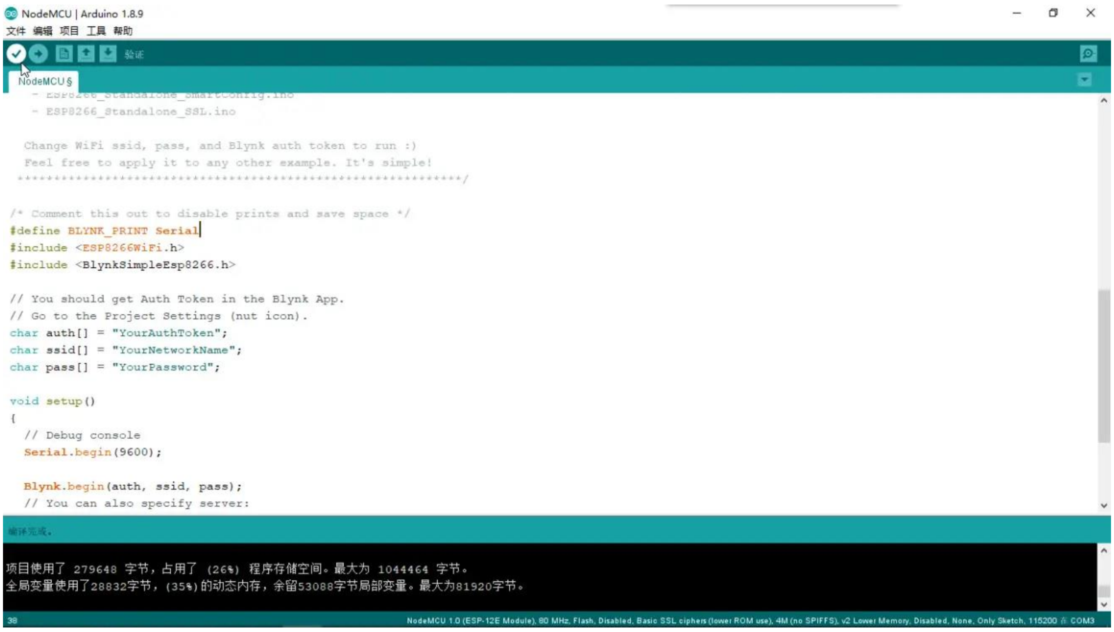
The program can modify the selected three places, 1. Blynk key (how to obtain it will be explained in detail later) 2. WIFI ssid 3. WIFI password



```
  Feel free to apply it to any other example. It's simple!
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);
```

46 - 49
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, Basic SSL ciphers (lower ROM use), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 在 COM3

Try to compile before modification, the following figure shows and compile successfully (the program is no problem)



```
  - ESP8266_Standalone_SmartConfig.ino
  - ESP8266_Standalone_SSL.ino

  Change WiFi ssid, pass, and Blynk auth token to run :)
  Feel free to apply it to any other example. It's simple!
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
```

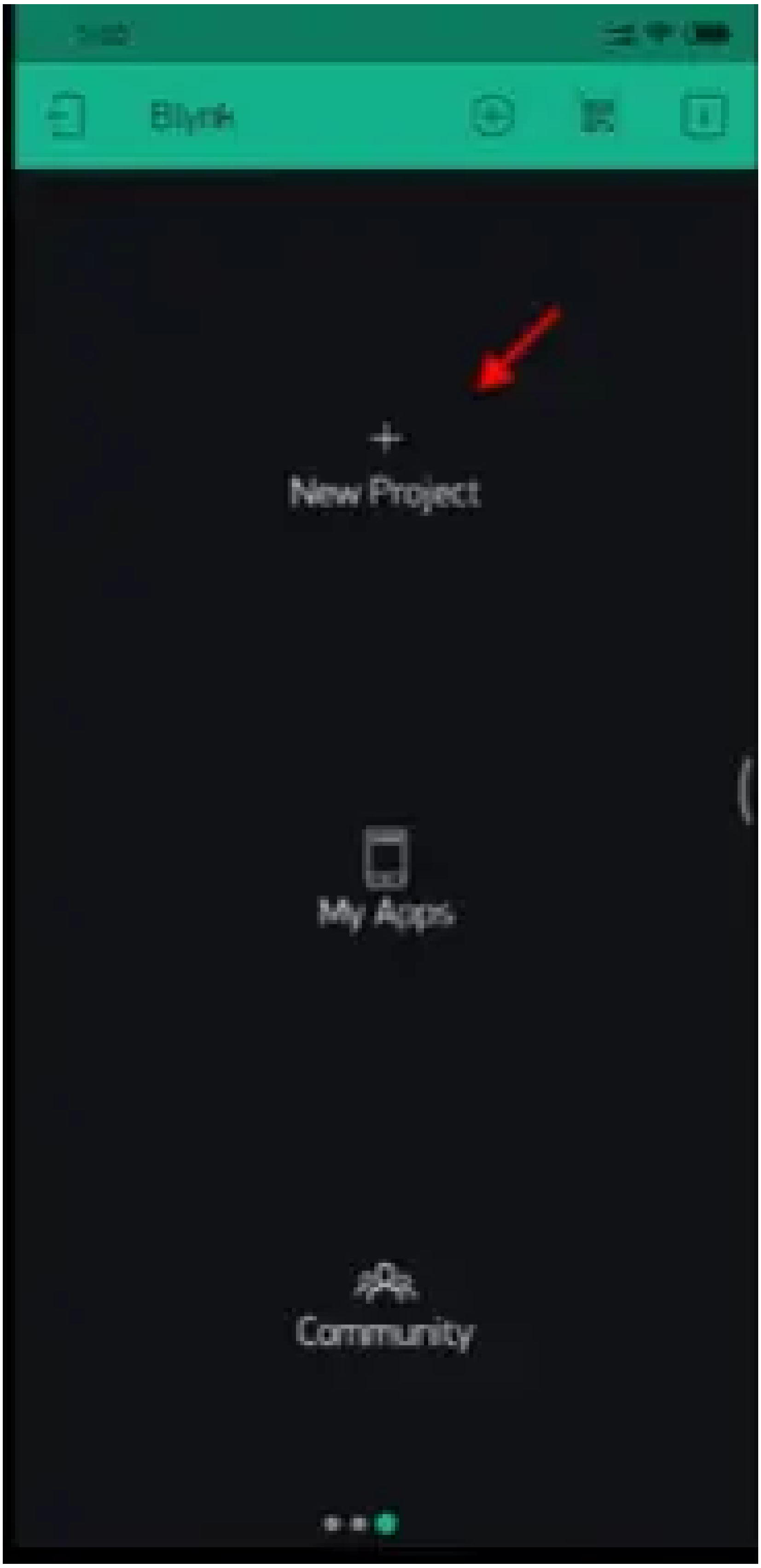
```
编译完成。

项目使用了 279648 字节，占用了 (26%) 程序存储空间。最大为 1044464 字节。
全局变量使用了28832字节，(35%)的动态内存，余留53088字节局部变量。最大为81920字节。
```

38
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled, Basic SSL ciphers (lower ROM use), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 在 COM3

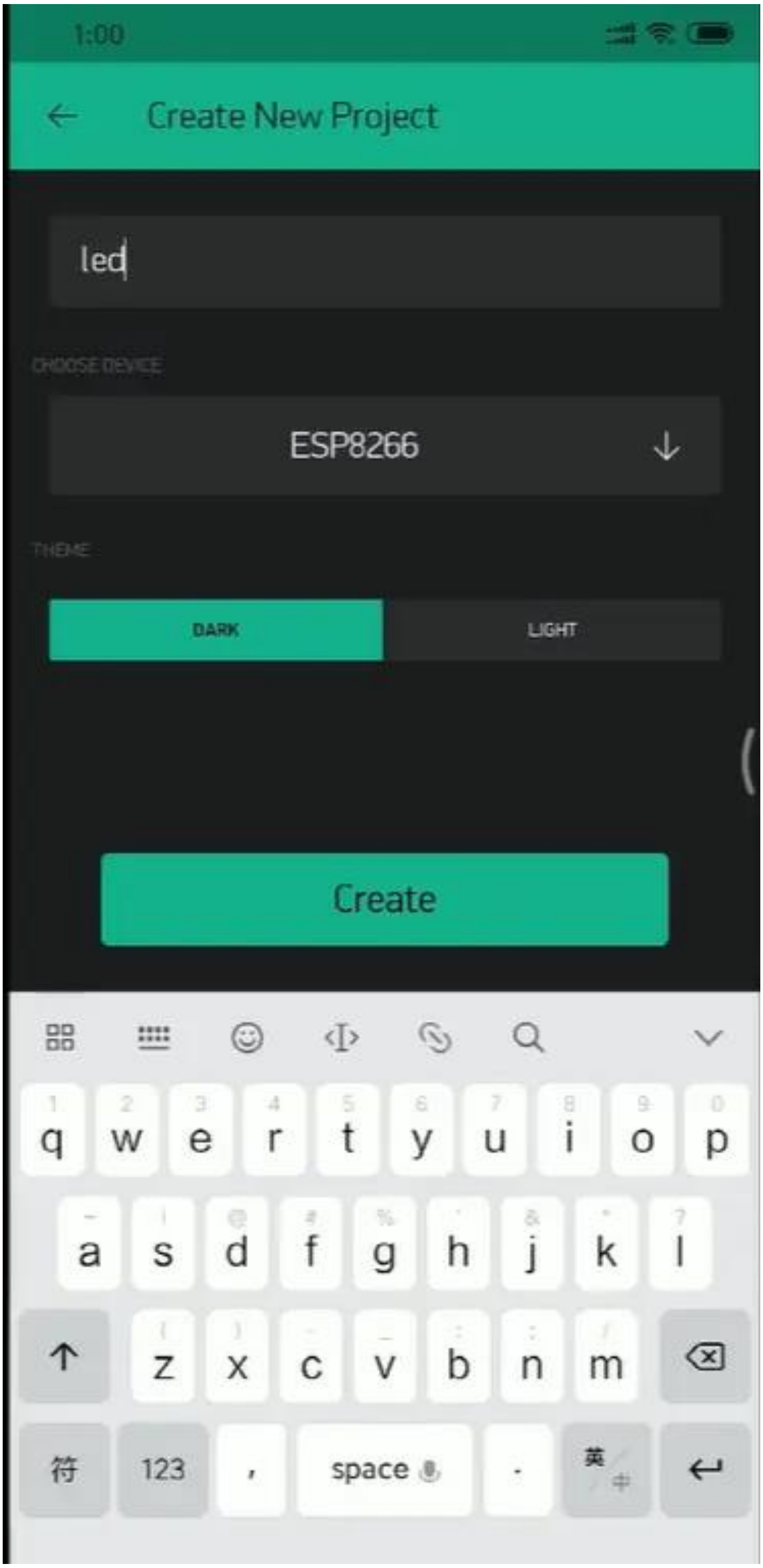
After compiling without problems, proceed to the following steps

（6）Download and open the Blynk APP to get the' Blynk key' to create the project

After downloading, register, login and Create project



Blynk
+
New Project
My Apps.
Community



Create New Project

led

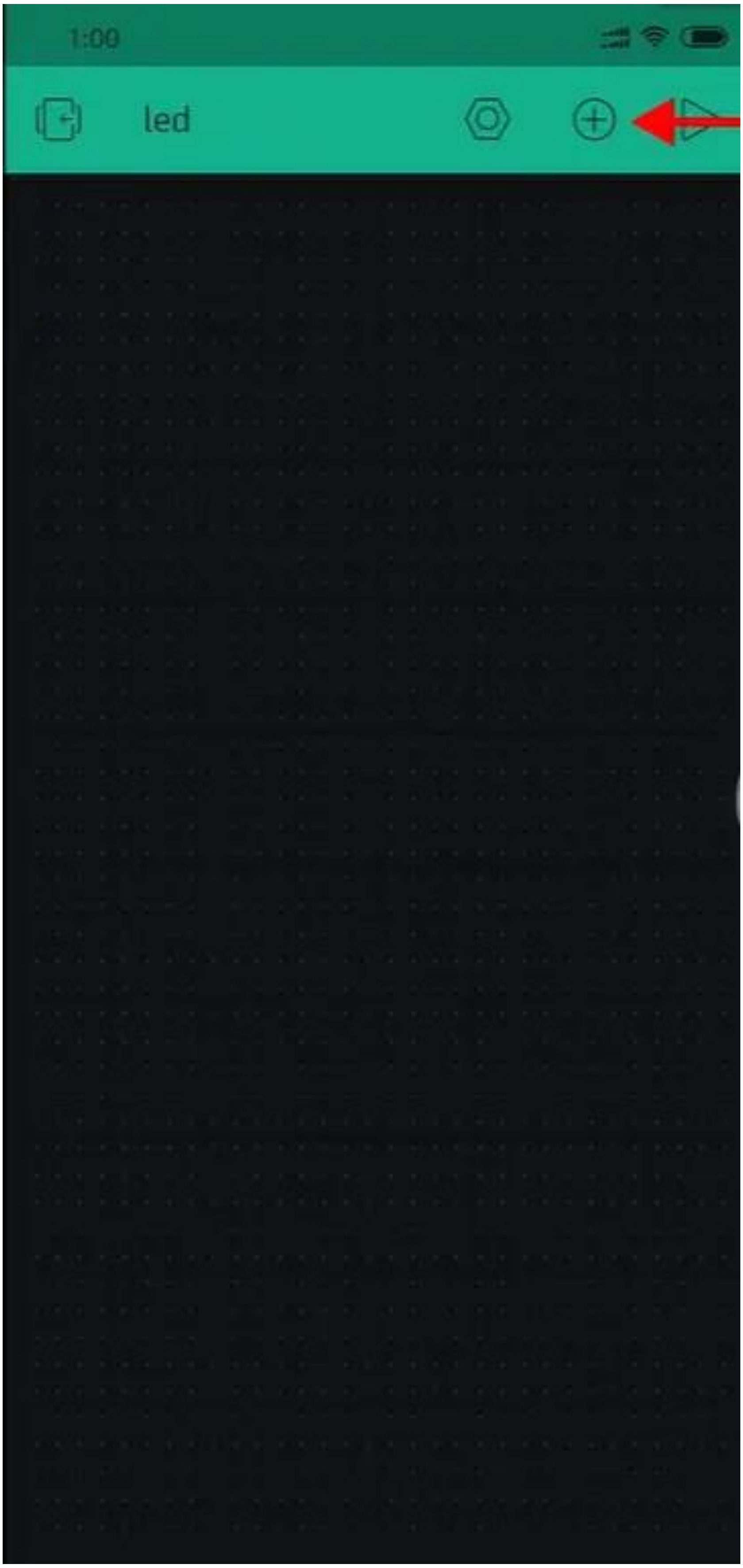
CHOOSE DEVICE

ESP8266

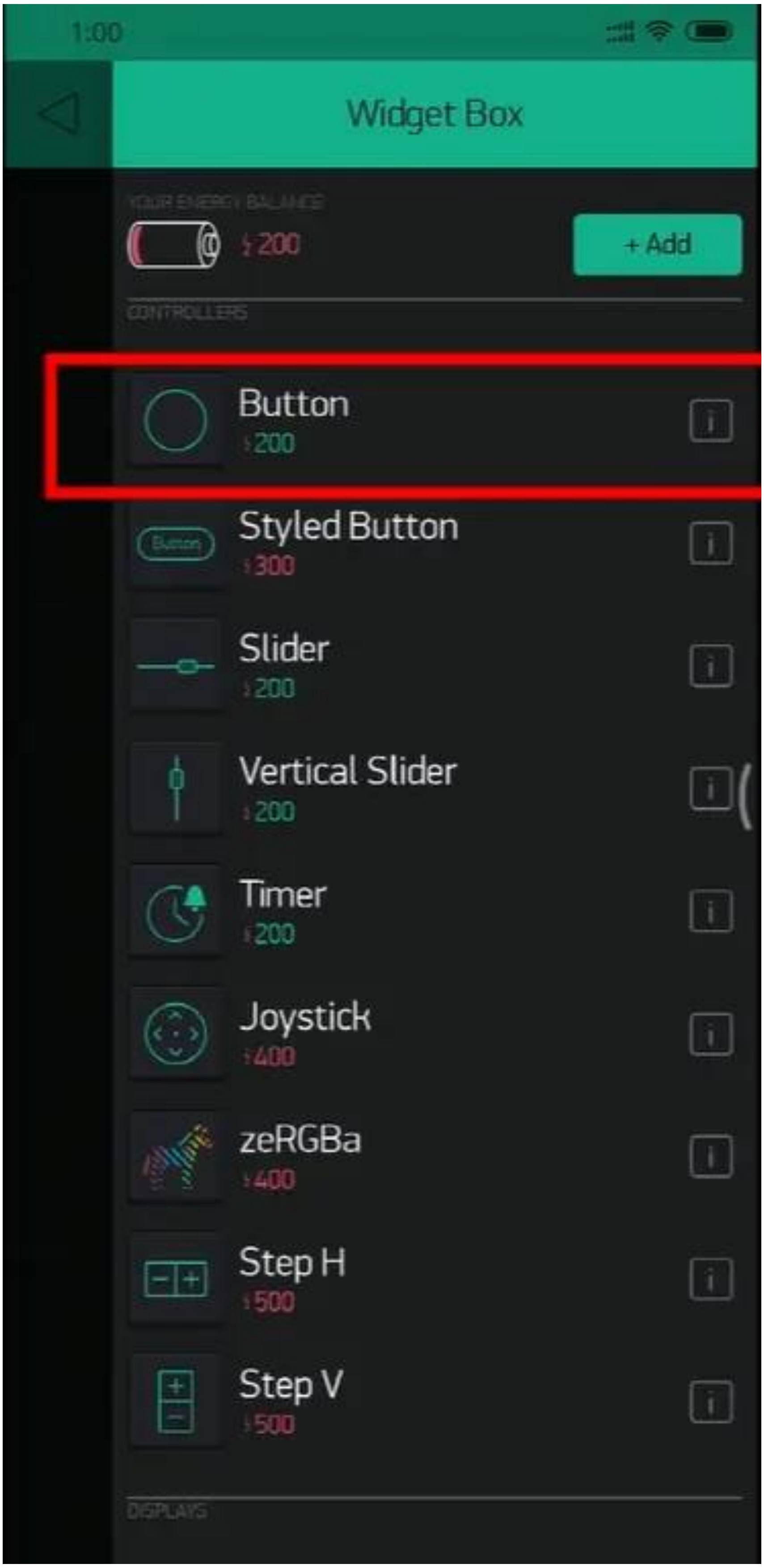
THEME

DARK | LIGHT

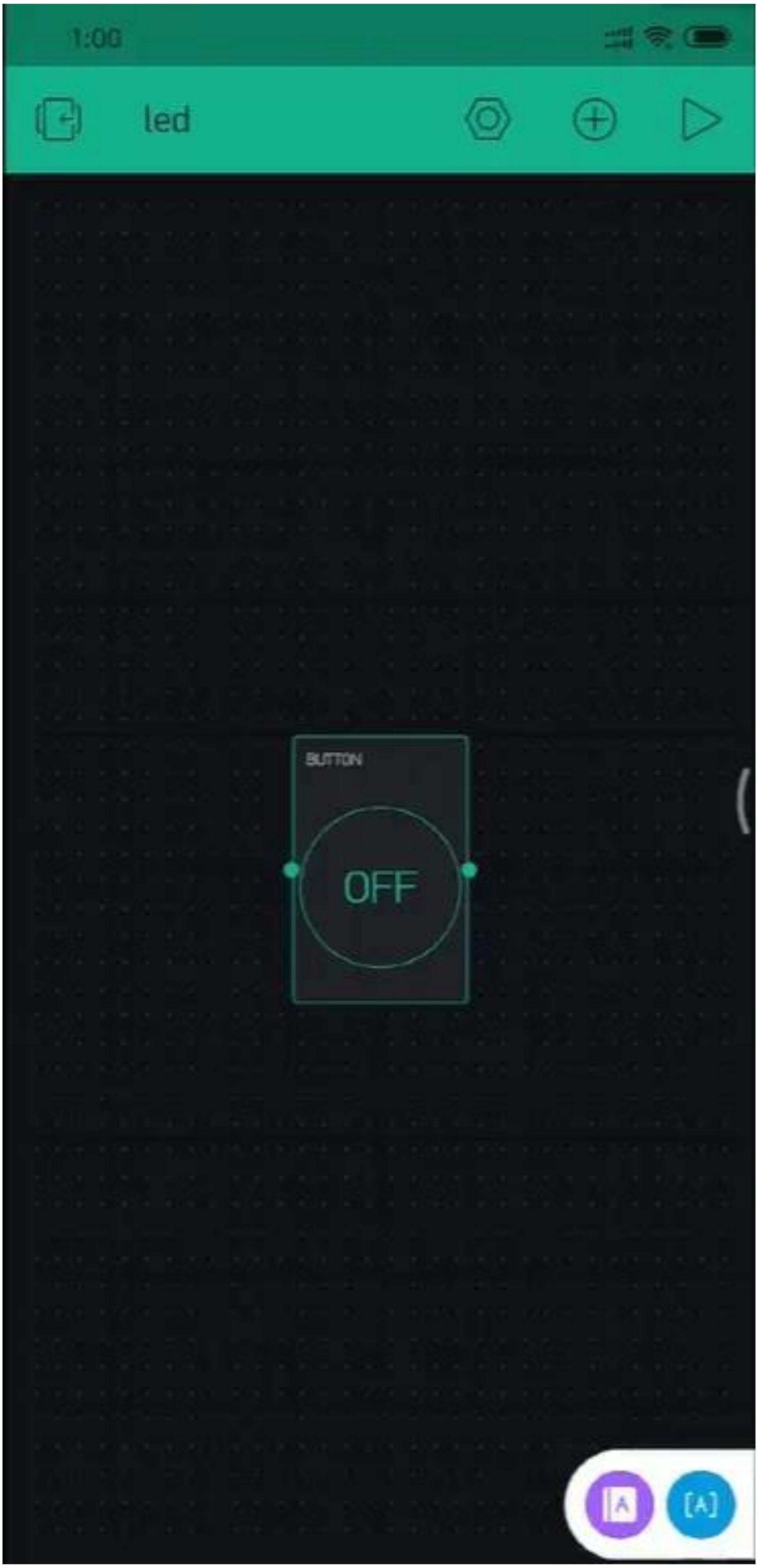
Choose ESP8266 and WIFI



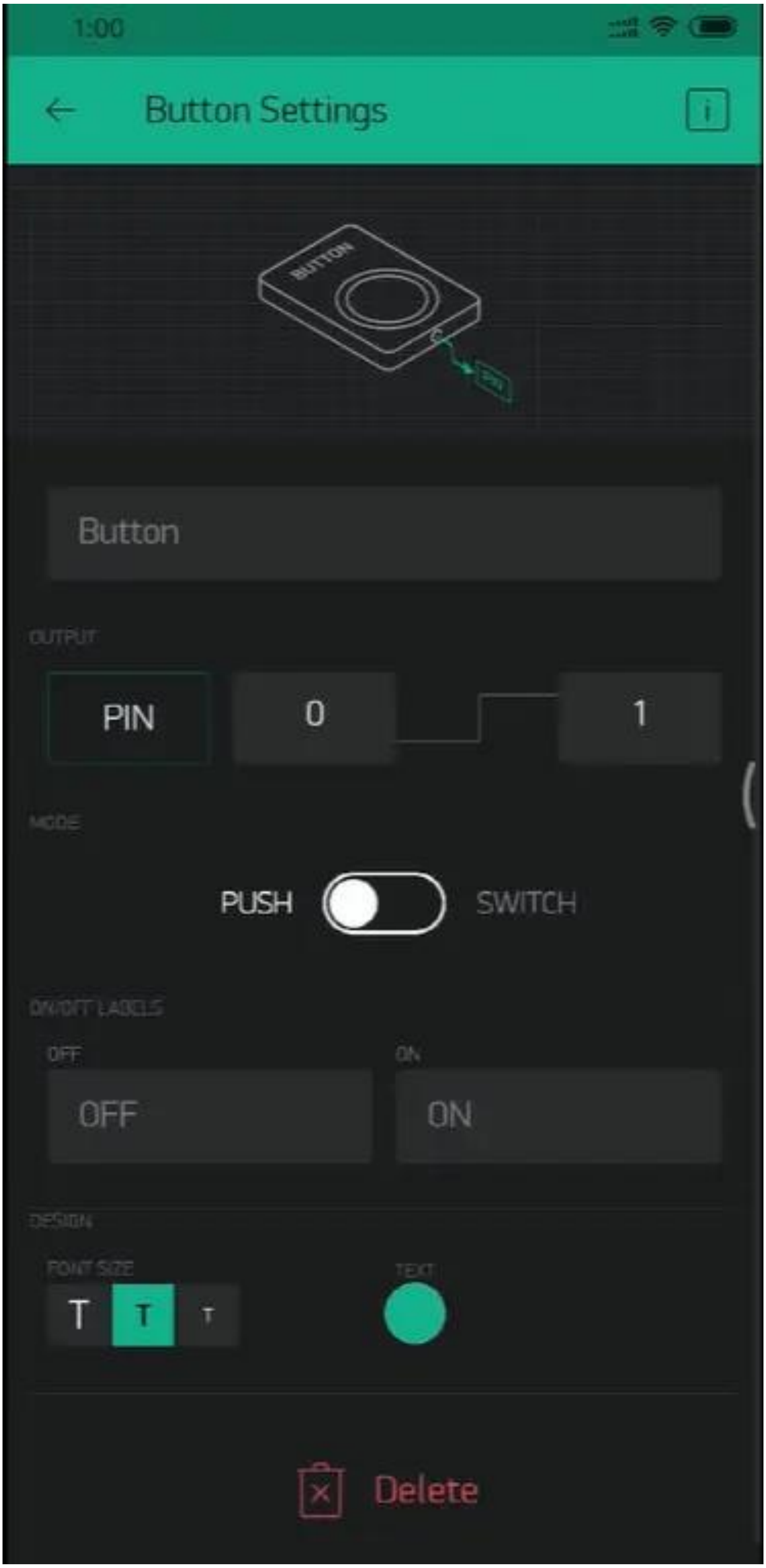
Add function buttons for new project



Add a function button, add a corresponding energy value for each consumption (the initial energy is 2000, free of charge)



1:00
led
BUTTON
OFF



Button Settings

Button

OUTPUT

PIN 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF

OFF

ON

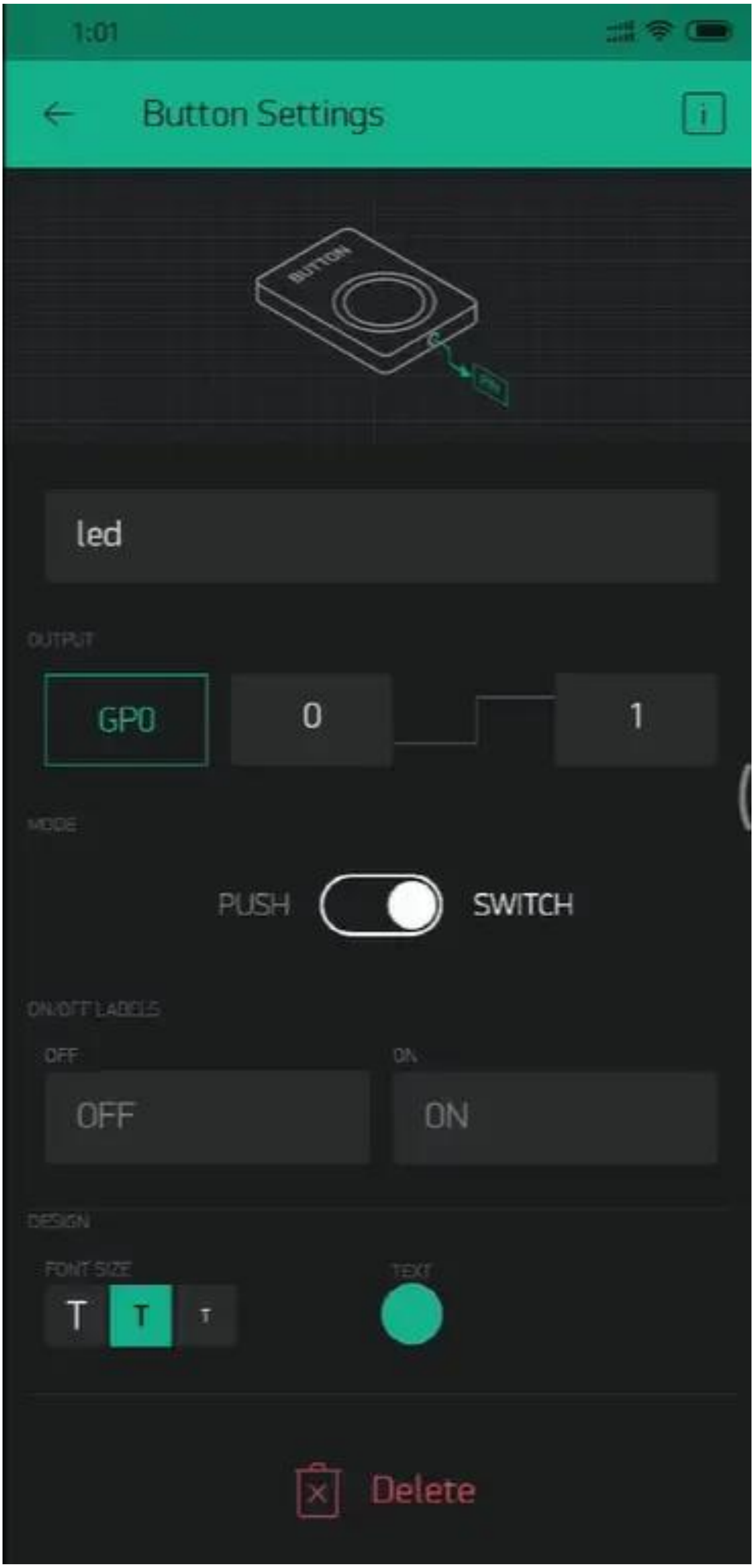
ON

DESIGN

FONT SIZE

T T T

TEXT



Button Settings

led

OUTPUT

GP0 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF: OFF

ON: ON

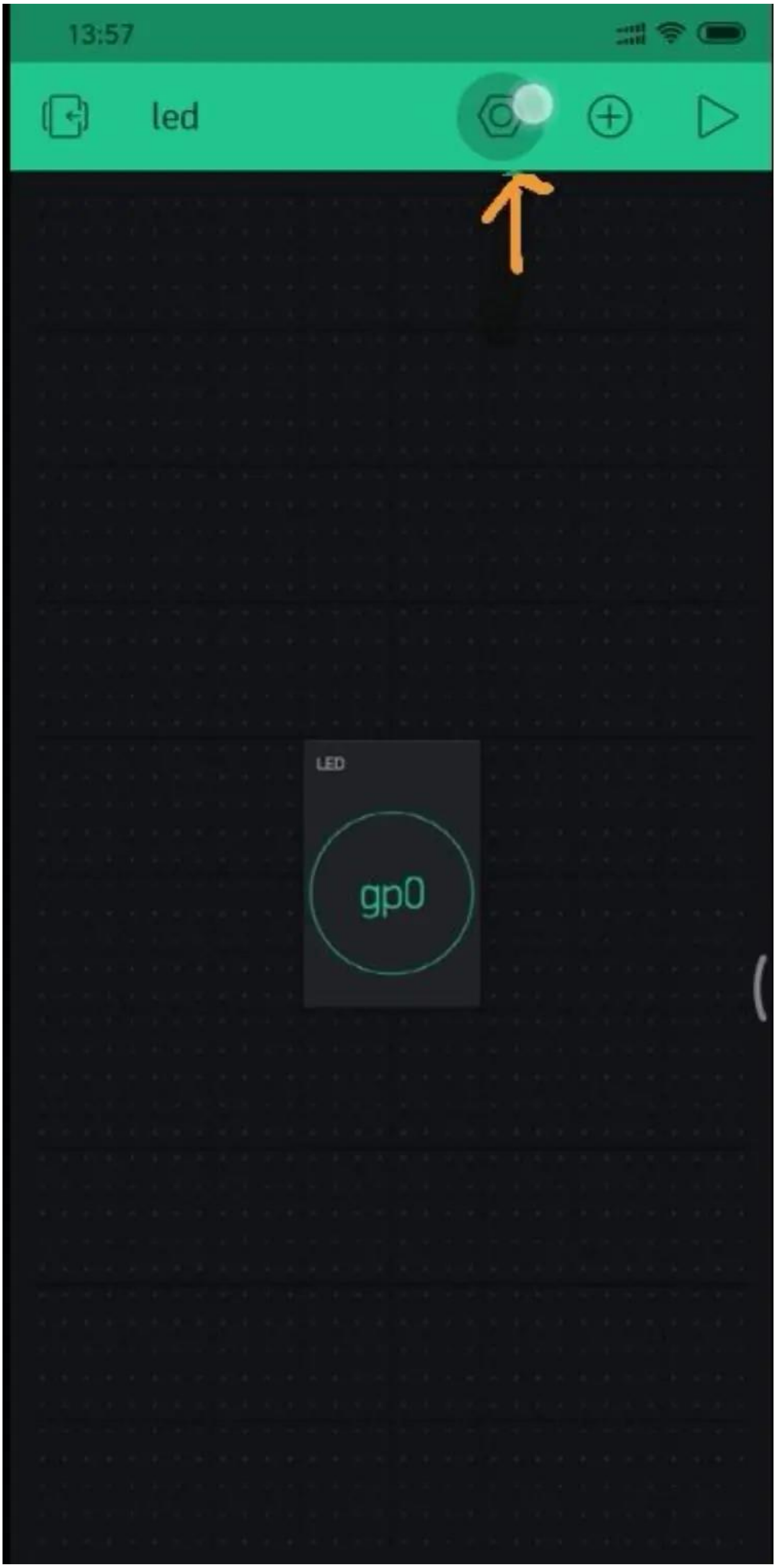
DESIGN

FONT SIZE

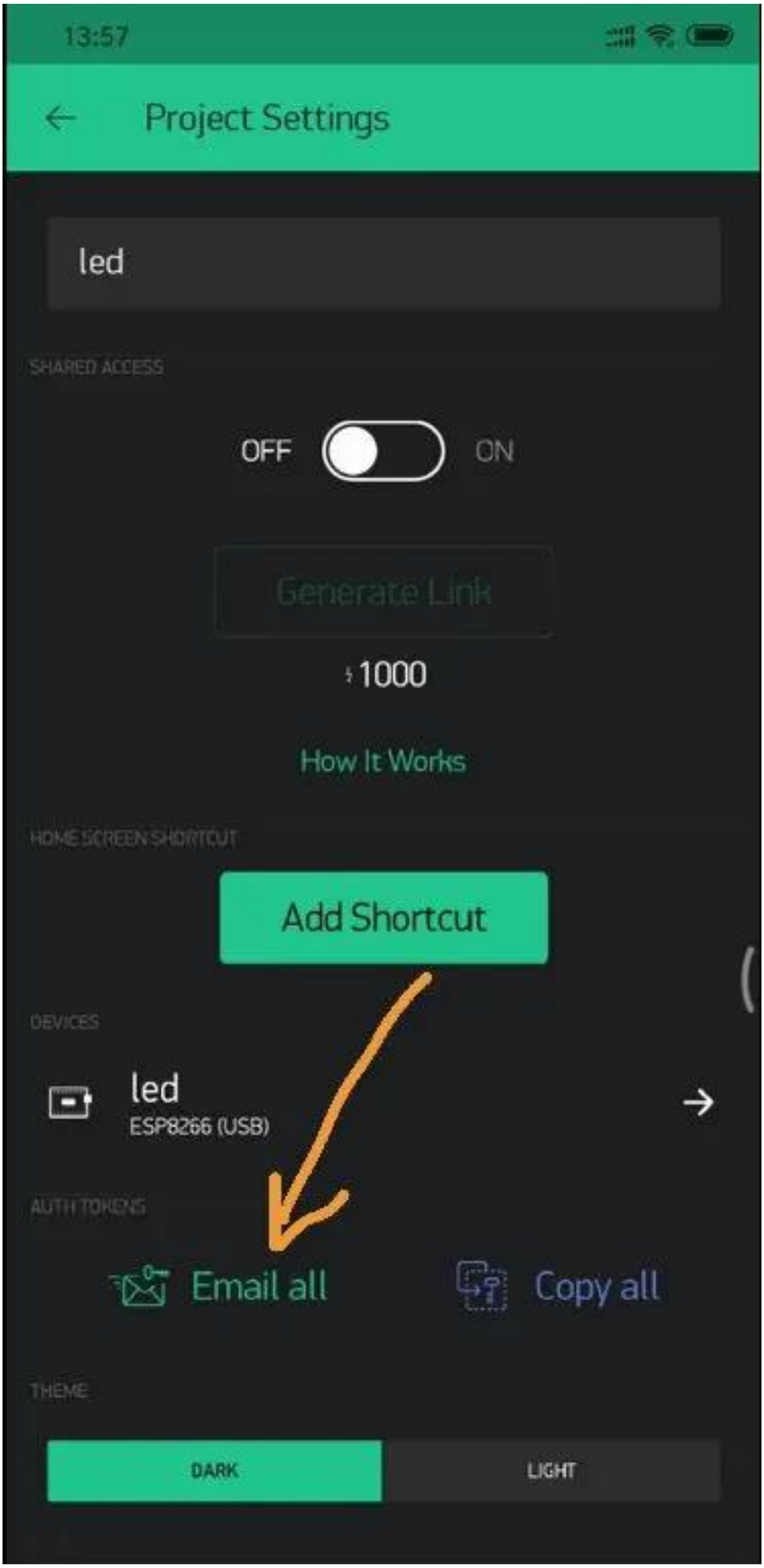
TEXT

Here choose GP0 as the output pin to control the high ground level to trigger the relay action, **GP0--the D3 pin** of the development board.

Send the Blynk "key" to your registered mailbox, just open the mailbox and copy it to your program, as shown in the figure below



13:57
led
LED
gp0



13:57

Project Settings

led

SHARED ACCESS

OFF ON

1000

HOME SCREEN SHORTCUT

DEVICES

led
ESP8266 (USB)

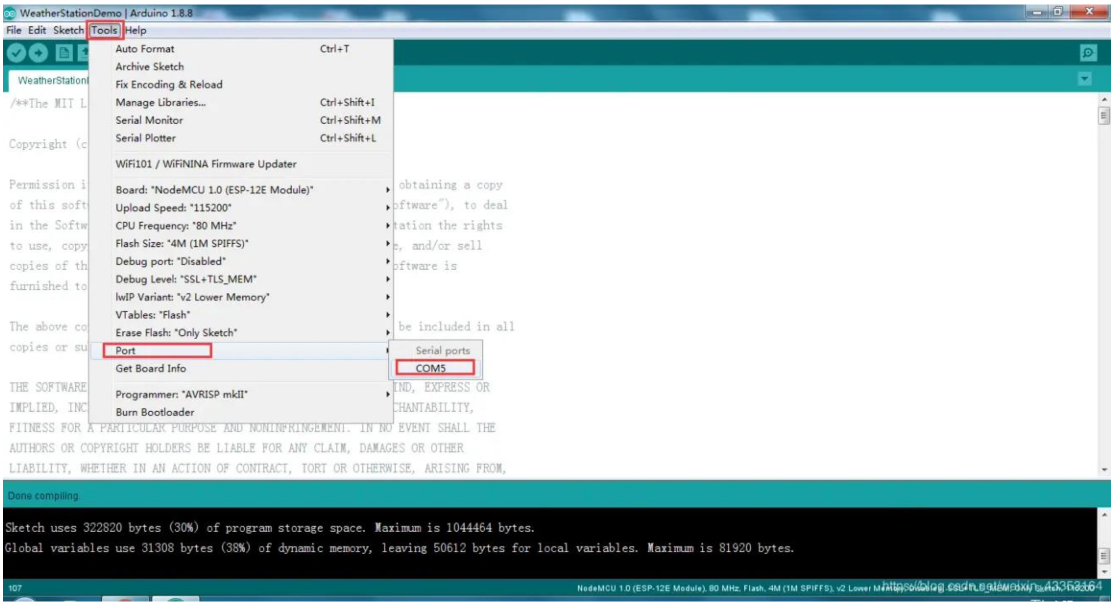
AUTH TOKENS

THEME

DARK LIGHT

(7）Modify the program and upload it to the ESP8266 development board

Upload to your development board, select the port (COM), depending on the actual situation, connect your esp8266 with a USB cable, and it will appear (the program in the picture shown is not the program in this tutorial, just to show how to operate)

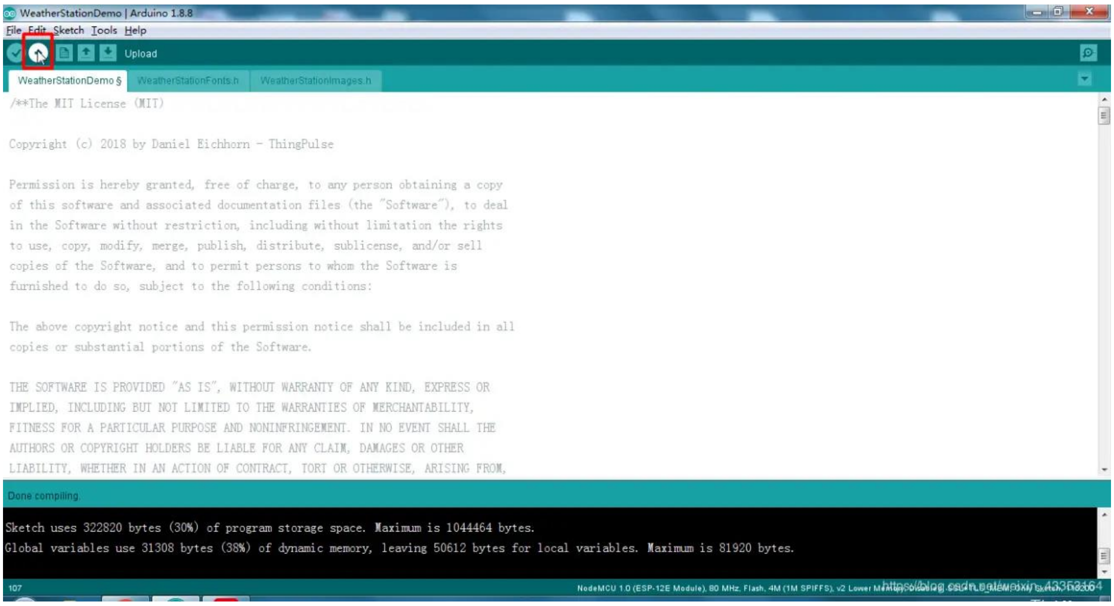


Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4M (1M SPIFFS)"
Debug port: "Disabled"
Debug Level: "SSL+TLS_MEM"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Erase Flash: "Only Sketch"
Programmer: "AVRISP mkII"

```
Sketch uses 322820 bytes (30%) of program storage space. Maximum is 1044464 bytes.
Global variables use 31308 bytes (38%) of dynamic memory, leaving 50612 bytes for local variables. Maximum is 81920 bytes.
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower M

Click Upload to download to esp8266

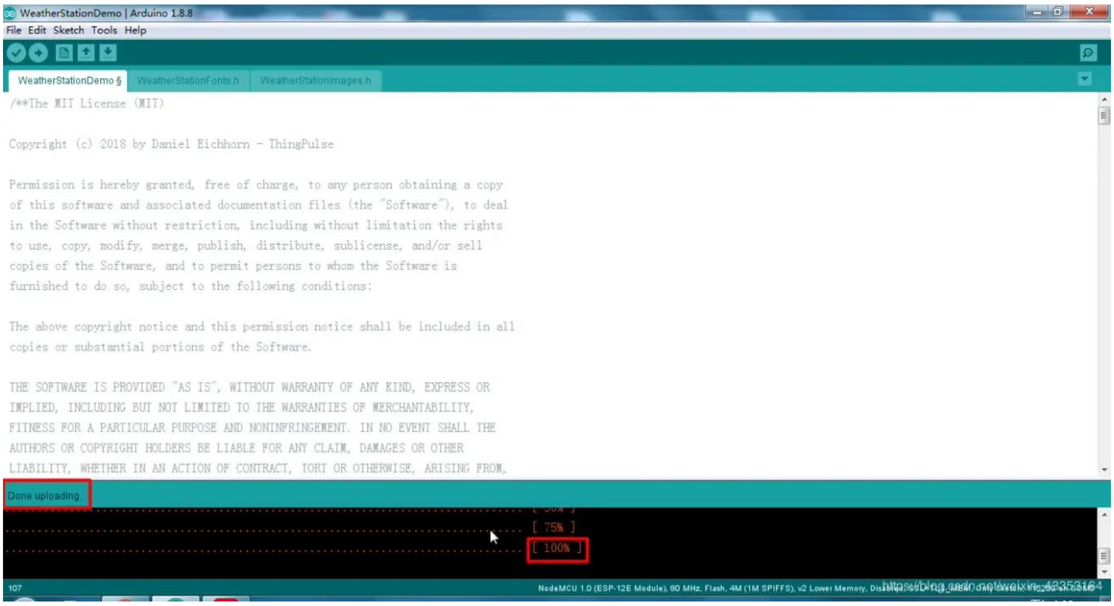


```
/**The MIT License (MIT)

Copyright (c) 2018 by Daniel Eichhorn - ThingPulse

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```

```
Sketch uses 322820 bytes (30%) of program storage space. Maximum is 1044464 bytes.
Global variables use 31308 bytes (38%) of dynamic memory, leaving 50612 bytes for local variables. Maximum is 81920 bytes.
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower M

Burning is complete



```
/**The MIT License (MIT)

Copyright (c) 2018 by Daniel Eichhorn - ThingPulse

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```

Done uploading.

```
................................................................................ [ 75% ]
................................................................................ [ 100% ]
```

107 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Dis

(8）Appendix: Program (you can use it after you modify the mentioned three places)

```
/* Comment this out to disable prints and save space */

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "*********";   //Blynk   “key”

char ssid[] = "*********";   //wifi   “ssid”

char pass[] = "*********";    //wifi   “password”

void setup()

{

// Debug console

Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:

  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);

  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8442);

}

void loop()

{

  Blynk.run();

}
```