第十二讲
ESP32
Lecture 12
ESP32

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

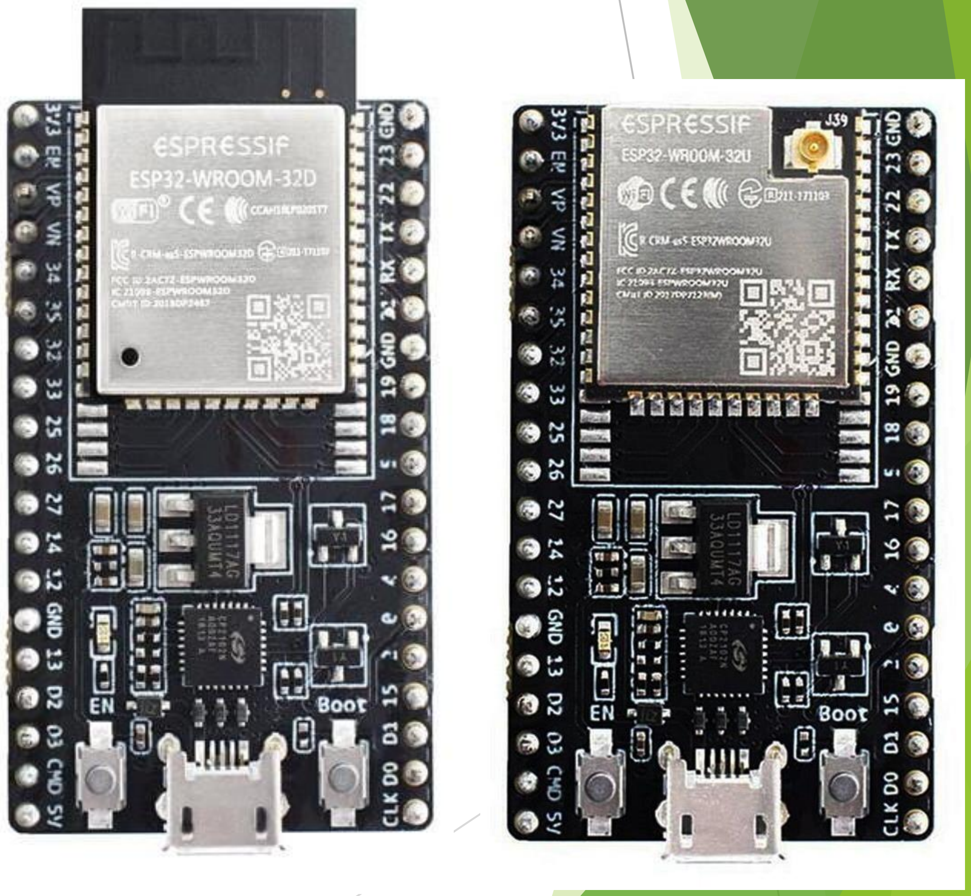
- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

ESP32

- ESP32系列由上海乐鑫科技有限公司生产。该公司主要从事ESP32系列芯片设计，然后交由其他厂家生产。我们常用的核心板以ESP-WROOM-32为主，又分为D版本与U版本。主要区别在于D版本的WIFI模块有板载天线，而U版本的需要外接天线。

  The ESP32 microprocessor is designed by Espressif based in Shanghai, China. It is a fabless company which means the ESP32 series chipsets on markets are manufactured by other downstream companies. The commonly-used core board are based on ESP-WROOM-32 version, which is further categorized into D and U subtypes. The mainly difference between these two versions is that the WIFI module comes with an onboard antenna with D type, however, for U extra independent antenna is required.



ESP-WROOM-32

双核32位微控制器模块	Dual-core, 32-bit microcontroller module
CPU内核可以单独控制	CPU cores can be individually controlled
时钟频率高达240 MHz	Clock frequency up to 240 MHz
多种电源模式	Multiple Power modes
集成 WiFi、蓝牙和 BLE	Integrated WiFi, Bluetooth and BLE
多个数字和模拟 I/O 引脚	Multiple digital and analog I/O pins
多达18个12位模数转换器	Up to eighteen 12-bit analog to digital converters
两个8位数模转换器	Two 8-bit digital to analog converters
十个电容式触摸传感器输入	Ten capacitive touch-sensor inputs
四个SPI总线通道	Four SPI bus channels
两个I2C总线连接	Two I2C bus connections
两个 I2S 总线连接	Two I2S bus connections
三个 UART	Three UARTs

ESP-WROOM-32

- SD卡主机控制器
- 红外遥控器，最多八通道
- 电机脉宽调制
- LED PWM，最多16个通道
- 霍尔效应传感器
- 超低功耗模拟前置放大器
- 实时时钟
- ESP-WROOM-32上的大多数引脚具有多种功能
- 部分功能存在冲突，并非所有功能都可以同时使用
- 并非所有开发板都暴露所有引脚
- 有些引脚不推荐使用
- 一些数字 I/O 引脚仅用于输入

- SD card host controller
- IR remote controller, up to eight channels
- Motor PWM
- LED PWM, up to sixteen channels
- Hall-effect sensor
- Ultra low-power analog preamp
- Real time clocks
- Most pins on ESP-WROOM-32 have multiple functions
- Some functions conflicts, not all can be used simultaneously
- Not all development boards expose all pins
- Some pins are not recommended for use
- Some digital I/O pins are input only

芯片系列比较
Chip Series Comparison

- 除了ESP32，常用的还有ESP32-S与ESP32-C系列。ESP32-S系列将CPU核心由Xtensa® LX6系列升级为Xtensa® LX7系列。其中ESP32-S2为单核心，ESP32-S3为双核心。而ESP32-C系列（市面上仅有ESP32-C3系列）则将CPU核换成了支持指令集RISC-V的核。

  In addition to ESP32, the commonly-used serials include ESP32-S and ESP32-C series. ESP32-S series upgrades CPU cores from the Xtensa® LX6 to the Xtensa® LX7. However, ESP32-S2 is single-core and ESP32-S3 is dual-core. The ESP32-C series (only the ESP32-C3 series are on the market) replaced the CPU core with the core that is based on RISC-V.

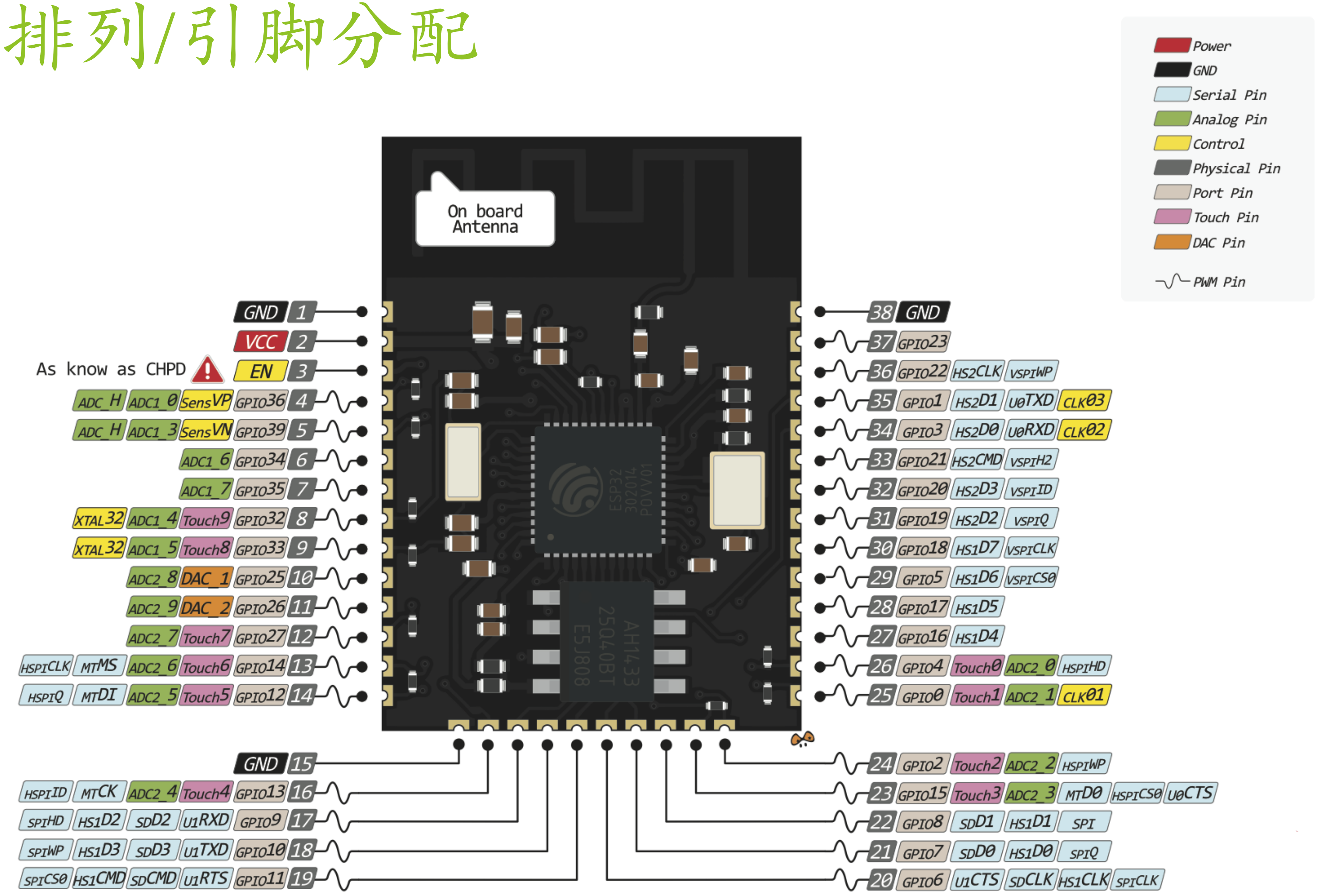
- 尽管ESP32-S2系列比ESP32运行更快，但其不支持蓝牙。尽管ESP32-C3系列不存在授权问题，但其为单核，且不存在低功耗模式。

  Although the ESP32-S2 series runs faster than ESP32, it does not support Bluetooth. Although the ESP32-C3 series does not have licensing issues, it is single-core and does not have a low-power mode.

ESP32引脚/引脚排列/引脚分配
ESP32 Pinout

- ESP32系列通过40个左右的引脚将MCU所提供的功能引出。由于成本所限，为了用有限个引脚提供尽可能多的功能，因此，ESP32的一个特点是引脚复用。

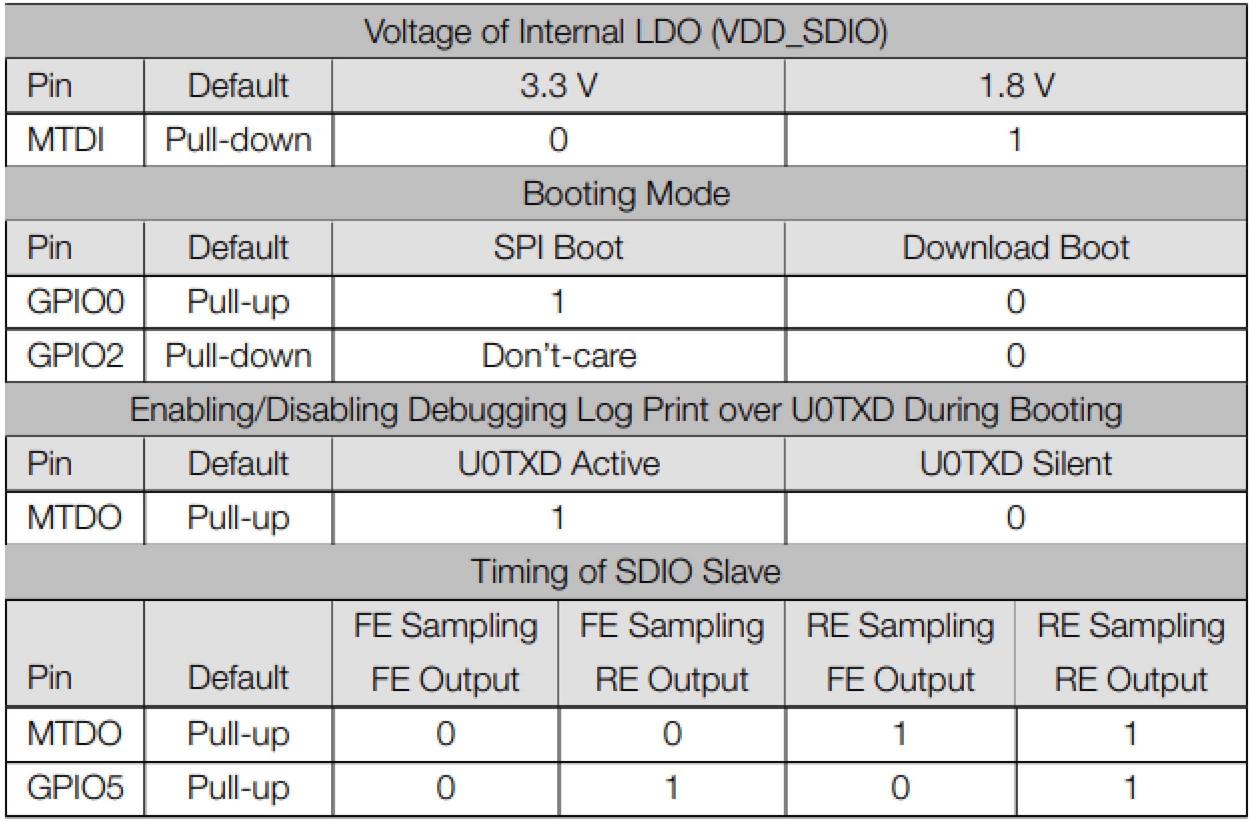
  The ESP32 series expose the various functions of MCU via around 40 pins. Due to cost constraints, to provide as many functions as possible via a limited number of pins incurs the pin multiplexing feature of ESP32.


Power
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin
PWM Pin
On board Antenna
GND 1
VCC 2
As know as CHPD EN 3
ADC_H ADC1_0 SensVP GPIO36 4
ADC_H ADC1_3 SensVN GPIO39 5
ADC1_6 GPIO34 6
ADC1_7 GPIO35 7
XTAL32 ADC1_4 Touch9 GPIO32 8
XTAL32 ADC1_5 Touch8 GPIO33 9
ADC2_8 DAC_1 GPIO25 10
ADC2_9 DAC_2 GPIO26 11
ADC2_7 Touch7 GPIO27 12
HSPICLK MTMS ADC2_6 Touch6 GPIO14 13
HSPIQ MTDI ADC2_5 Touch5 GPIO12 14
GND 15
HSPIID MTCK ADC2_4 Touch4 GPIO13 16
SPIHD HS1D2 SDD2 U1RXD GPIO9 17
SPIWP HS1D3 SDD3 U1TXD GPIO10 18
SPICS0 HS1CMD SDCMD U1RTS GPIO11 19
38 GND
37 GPIO23
36 GPIO22 HS2CLK VSPIWP
35 GPIO1 HS2D1 U0TXD CLK03
34 GPIO3 HS2D0 U0RXD CLK02
33 GPIO21 HS2CMD VSPIH2
32 GPIO20 HS2D3 VSPIID
31 GPIO19 HS2D2 VSPIQ
30 GPIO18 HS1D7 VSPICLK
29 GPIO5 HS1D6 VSPICS0
28 GPIO17 HS1D5
27 GPIO16 HS1D4
26 GPIO4 Touch0 ADC2_0 HSPIHD
25 GPIO0 Touch1 ADC2_1 CLK01
24 GPIO2 Touch2 ADC2_2 HSPIWP
23 GPIO15 Touch3 ADC2_3 MTD0 HSPICS0 U0CTS
22 GPIO8 SDD1 HS1D1 SPI
21 GPIO7 SDD0 HS1D0 SPIQ
20 GPIO6 U1CTS SDCLK HS1CLK SPICLK

管脚复用
Pin Multiplexing

- 引脚复用最典型的即所谓的strapping管脚，在系统启动或复位时，根据采样到的不同的电压值，管脚会有不同的功能。例如，有的板子第一次烧录时，需要将GPIO0对应的按键先按下，然后才上电或按复位键，目的在于使GPIO0输入电压为0，以便从下载模式启动方便烧录固件。

  The most typical pin multiplexing is the so-called strapping pins, which pick up different functions based on the sampled input voltage during system startup or reset. For example, to flash some boards for the first time, you are required to press the button corresponding to GPIO0 first, and then power on or press the reset button, in order to make the input voltage for GPIO0 become low, so as to boot from the download mode to flash the firmware.



Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Booting Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don’t-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Booting					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

通道复用
Channel Multiplexing

- 通道复用更合适的叫法为功能复用。例如，ESP32提供了两个数模转换器（ADC）。当数模转换的速度要求不高时，则同一个ADC可以服务于多个需要数模转换的场景或外设。此时，可以为多个输入管脚创建独立的通道从而形成多个通道，ADC轮流将不同通道采集的模拟量转化为数字量。

  A more appropriate term for channel multiplexing might be function multiplexing. For example, ESP32 provides two digital-to-analog converters (ADC). When the requirement of the digital-to-analog conversion speed is not high, the same ADC can serve multiple scenarios or peripherals that require digital-to-analog conversion. At this point, separate channels can be created for multiple input pins to form multiple channels, and the ADC takes turns converting the analog acquired by different channels to digital.

- 注意，当使用WiFi时，连接ADC2的所有通道可能都受影响。此时推荐使用ADC1对应通道的管脚。这在从光敏电阻读取模拟值时尤其要注意。

  When WiFi is enabled, it might affect the ADC2 channels. It is recommended to use pins associated with channels of ADC1. You have to pay attention to this, for example, try to read the analog value from a LDR.

管脚限制
Pin Constrains

- 由于技术上的原因，尽管有些管脚也是GPIO管脚，但使用起来却有一定的限制，如GPIO34、35、36和39仅能作为模拟输入，而不能用于数字或模拟输出。

  For technical reasons, although some pins are also called GPIO pins, they have certain limitations in practice. For example, GPIO34, 35, 36, and 39 can only be used as analog inputs, but not digital or analog outputs.

- 当管脚用于输出时，输出电流不超过40mA。因此需要大电流驱动的负载考虑使用放大电路，以免直接使用管脚驱动从而造成对板子的损坏。

  When pins are used for output, the output current usually does not exceed 40mA. Therefore, for loads driven by high current, an amplification circuit should be used instead of direct pin drive to avoid damage to the board.

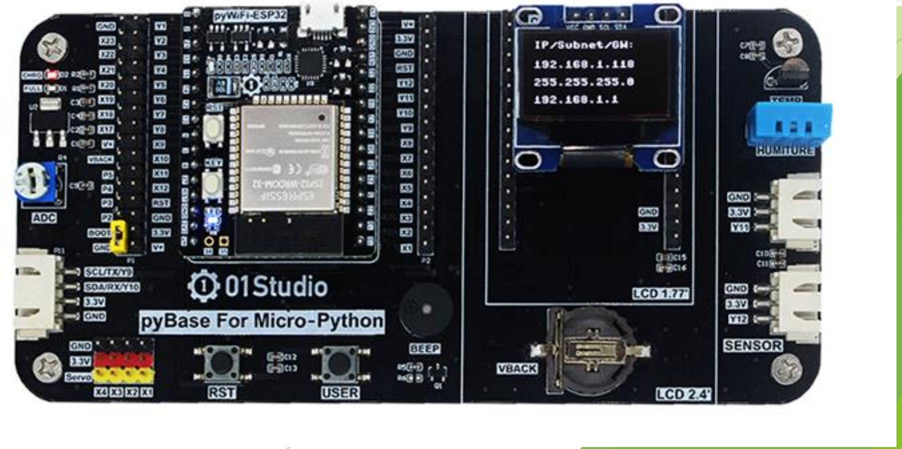
- 有些管脚尽管手册没有明确交代，如启动时GPIO12要默认低电平，管脚5与15要默认高电平；强制外加相反的电平可能会导致板子启动异常或无法启动。

  Although Some pins are not clearly specified in the data sheet, for example, voltage appearing at GPIO12 is required being low by default, GPIO5 and GPIO15 being high by default, forcing an opposite voltage level may cause the board to boot abnormally or fail to boot.

开发
Development

- 一般说来，单独的ESP32系列模块没有办法用来进行开发，能用于开发的基本都是基于ESP32的参考设计开发出来的各式各样的板子。这些开发板基本都将引脚拉出，有一些集成了一些常用的功能。根据集成功能的多少核心板价格在30~100元之间不等。

  Generally speaking, to develop using the ESP32 series dies are impractical. Various boards used for development are mainly based on ESP32's reference design. These boards basically expose all the pins, and some integrate certain common functions. According to the number of integrated components, the price of these core board varies between 30~100 RMB.



开发
Development

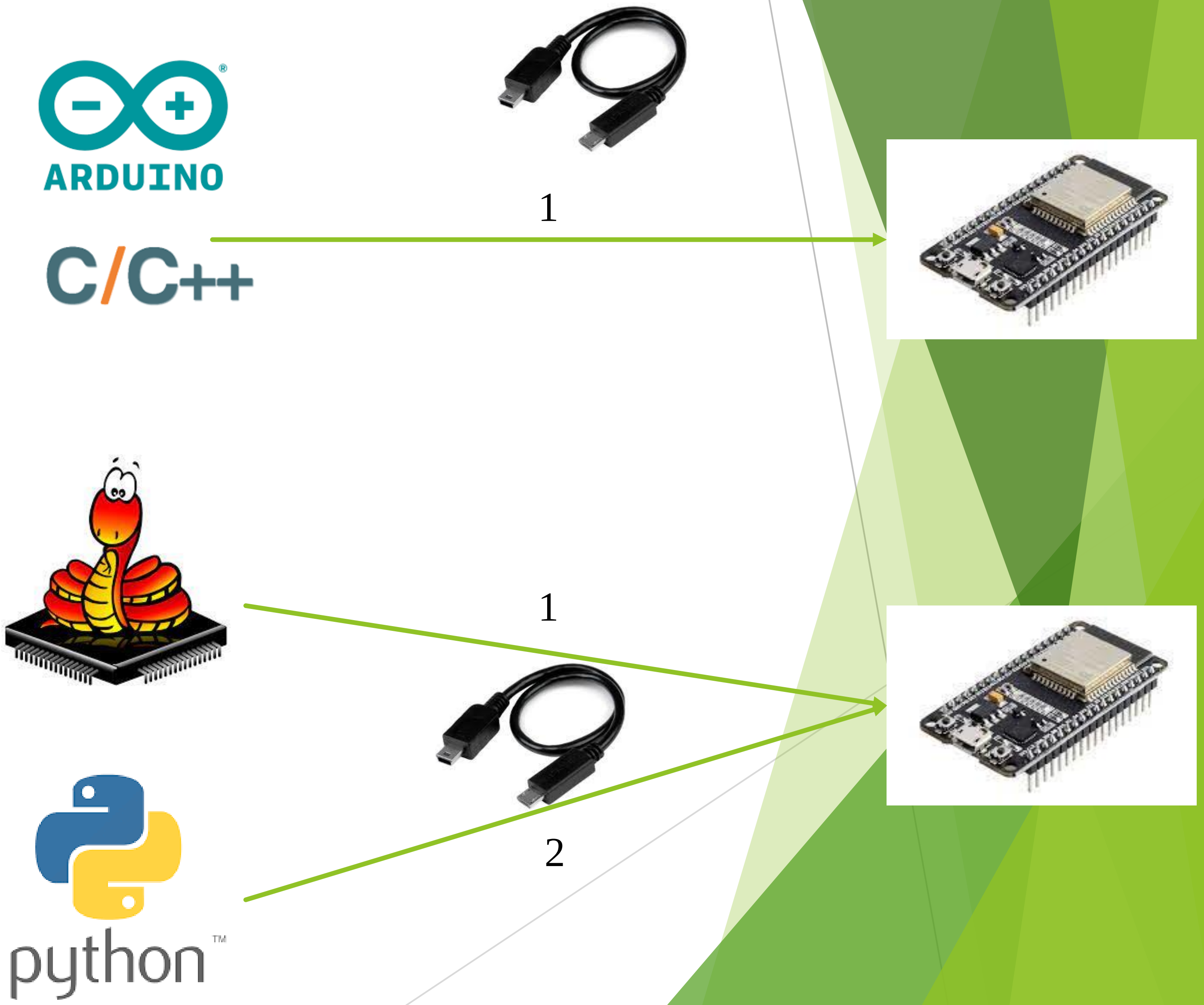
- 基于ESP32用于教学的物联网的开发一般有两种方式，一是基于Arduino IDE，一是基于MicroPython。基于Arduino IDE的开发主要基于C/C++语言。乐鑫公司已经提供了Arduino IDE的软件支持，可以方便地将相关工具集成进去后进行开发。基于MicroPython的开发基于Python语言，乐鑫公司或MicroPython官网均提供了相关的固件支持，一些图形化编程工具也集成了相关插件，方便将完成的Python代码上传至板子上执行。国内基于Python开发比较好的工具为米思齐（Mixly）开发的图形化编程环境。

  There are generally two ways to develop the education-oriented IoT applications based on ESP32. The first is based on the Arduino IDE and the other is based on MicroPython. Arduino IDE based development is mainly using the C/C++ language. Espressif has provided essential software support for the Arduino IDE, which makes it easy to launch the work after integrating related tools. The development based on MicroPython is using the Python language, Espressif or MicroPython official website has also provided firmware support, while some graphical programming tools have integrated plugins or extensions facilitating upload of Python scripts to the board for execution. The best tool for Python-based development domestically in China is the visual programming environment and tools developed by Mixly.

开发
Development

- 基于Arduino IDE的C/C++语言开发可以直接将程序代码编译后上传至开发板上运行。基于MicroPython的Python语言开发，需要首先初始化固件，然后才可以将Python脚本上传至板子上执行。

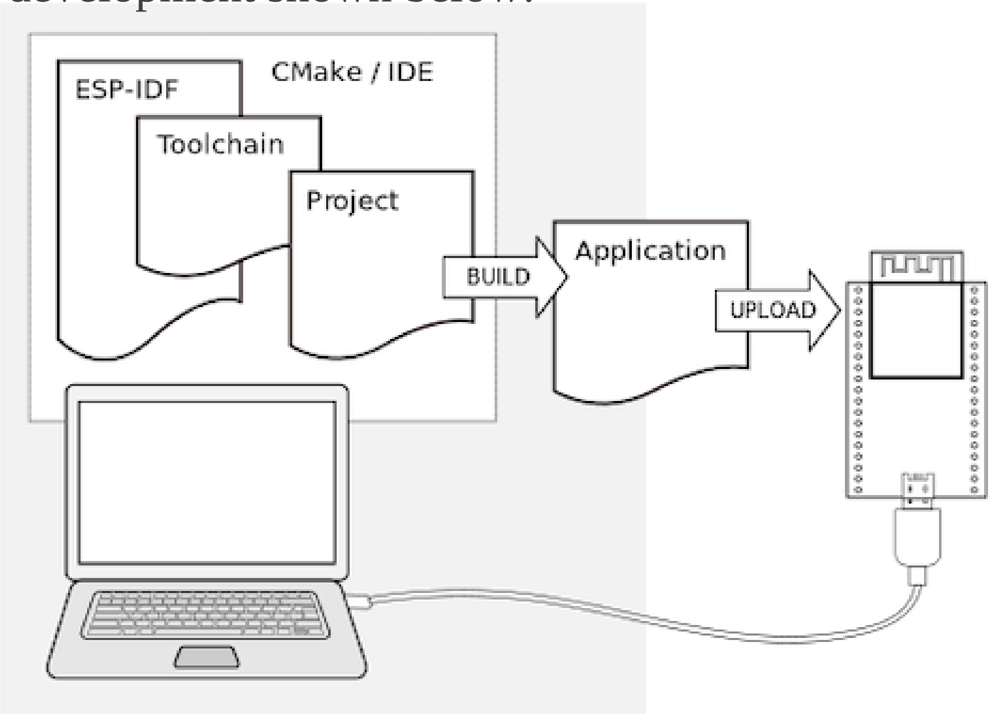
  The way of facilitating Arduino IDE based development by using the C/C++ language can directly upload/flash the compiled code into the board for executing. The development based on MicroPython by using the Python language need to initialize the board, which means flash the bin file contains the MicroPython interpreter, then upload the python script to run.



Arduino IDE

- 基于Arduino IDE的C/C++语言开发可以认为是如下流程的化简：

  The way of facilitating Arduino IDE based development by using the C/C++ language can be regarded as the simplification of the conventional development shown below:



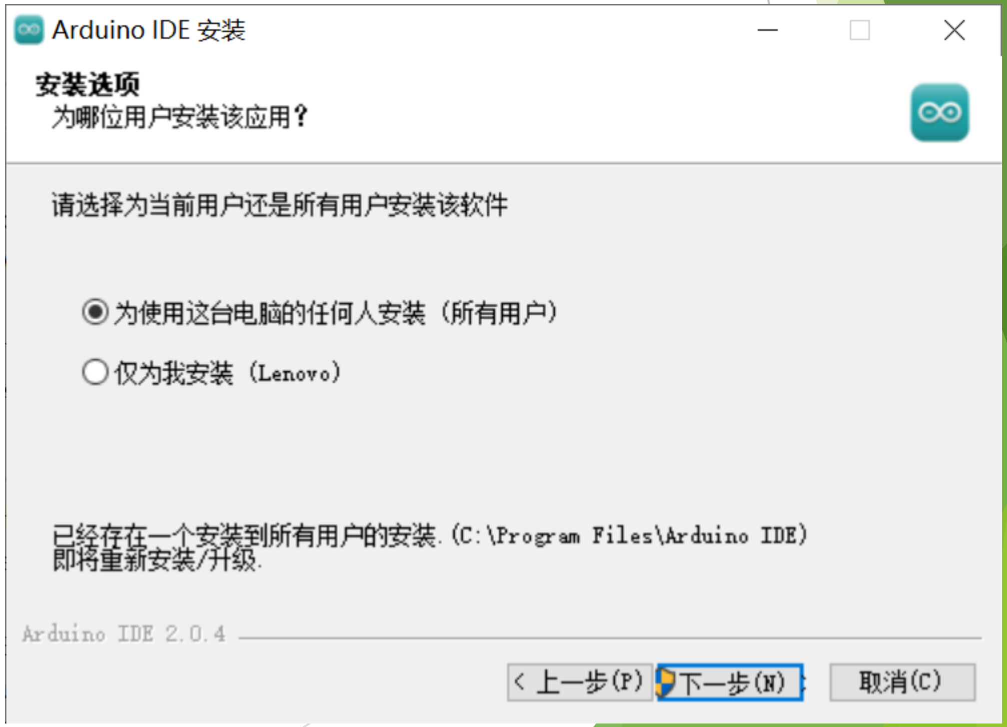
- 首先，去官网下载Arduino IDE：

  First download the Arduino IDE from the official website:

  https://www.arduino.cc/en/software

- 按照安装程序指示，安装即可：

  Follow the instruction of installer and install it:



Arduino IDE 安装

安装选项

为哪位用户安装该应用？

请选择为当前用户还是所有用户安装该软件

● 为使用这台电脑的任何人安装（所有用户）

○ 仅为我安装（Lenovo）

已经存在一个安装到所有用户的安装. (C:\Program Files\Arduino IDE)
即将重新安装/升级.

Arduino IDE 2.0.4

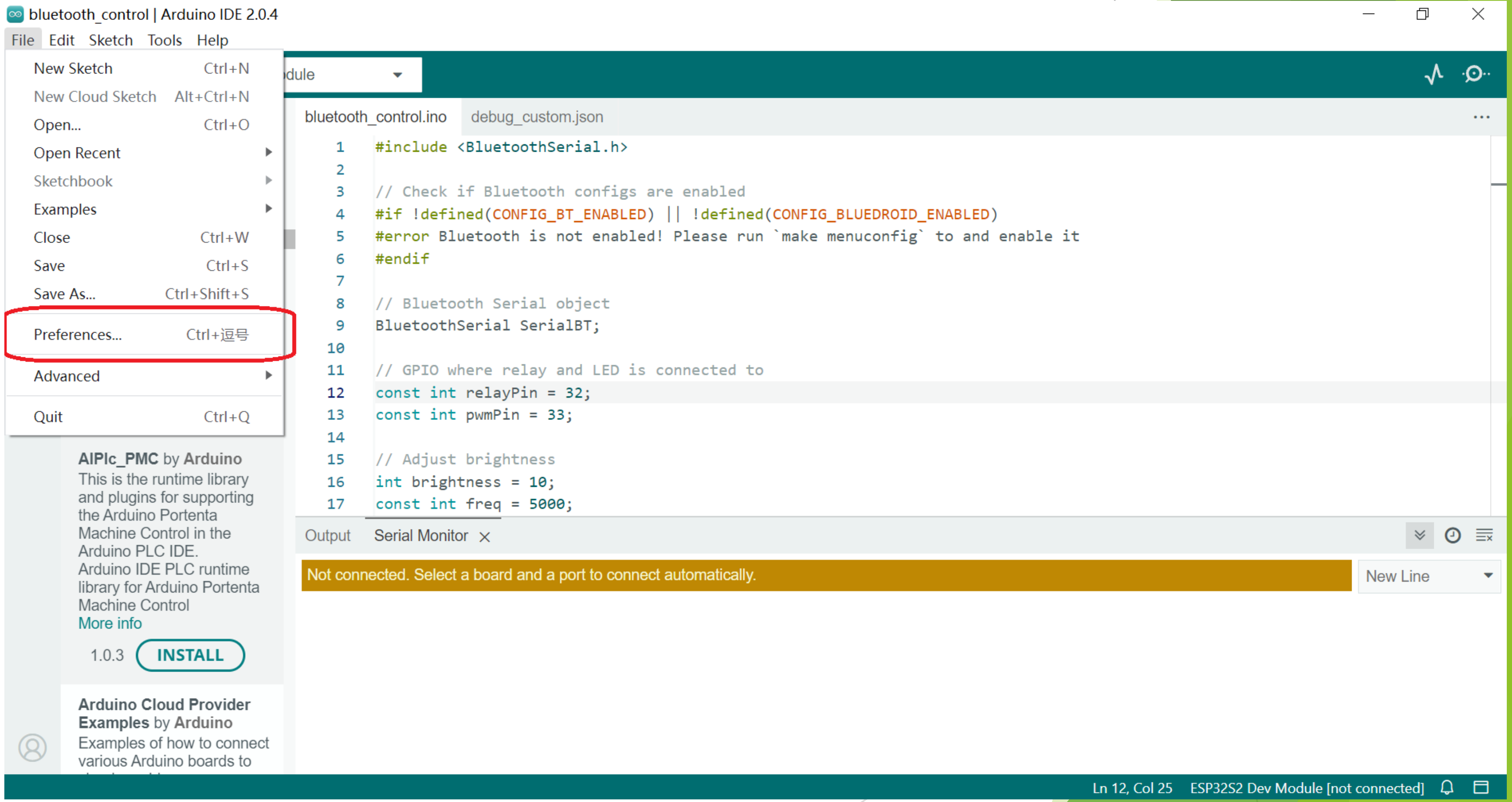
Arduino IDE

- 首先要配置Arduino IDE，获取IDF相关工具索引：

  We first need to configure the Arduino IDE to fetch the index of IDF tools:

- 选择文件→偏好：

  From the File menu, choose Preferences…:



```
#include <BluetoothSerial.h>

// Check if Bluetooth configs are enabled
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

// Bluetooth Serial object
BluetoothSerial SerialBT;

// GPIO where relay and LED is connected to
const int relayPin = 32;
const int pwmPin = 33;

// Adjust brightness
int brightness = 10;
const int freq = 5000;
```

AIPlc_PMC by Arduino
This is the runtime library and plugins for supporting the Arduino Portenta Machine Control in the Arduino PLC IDE. Arduino IDE PLC runtime library for Arduino Portenta Machine Control

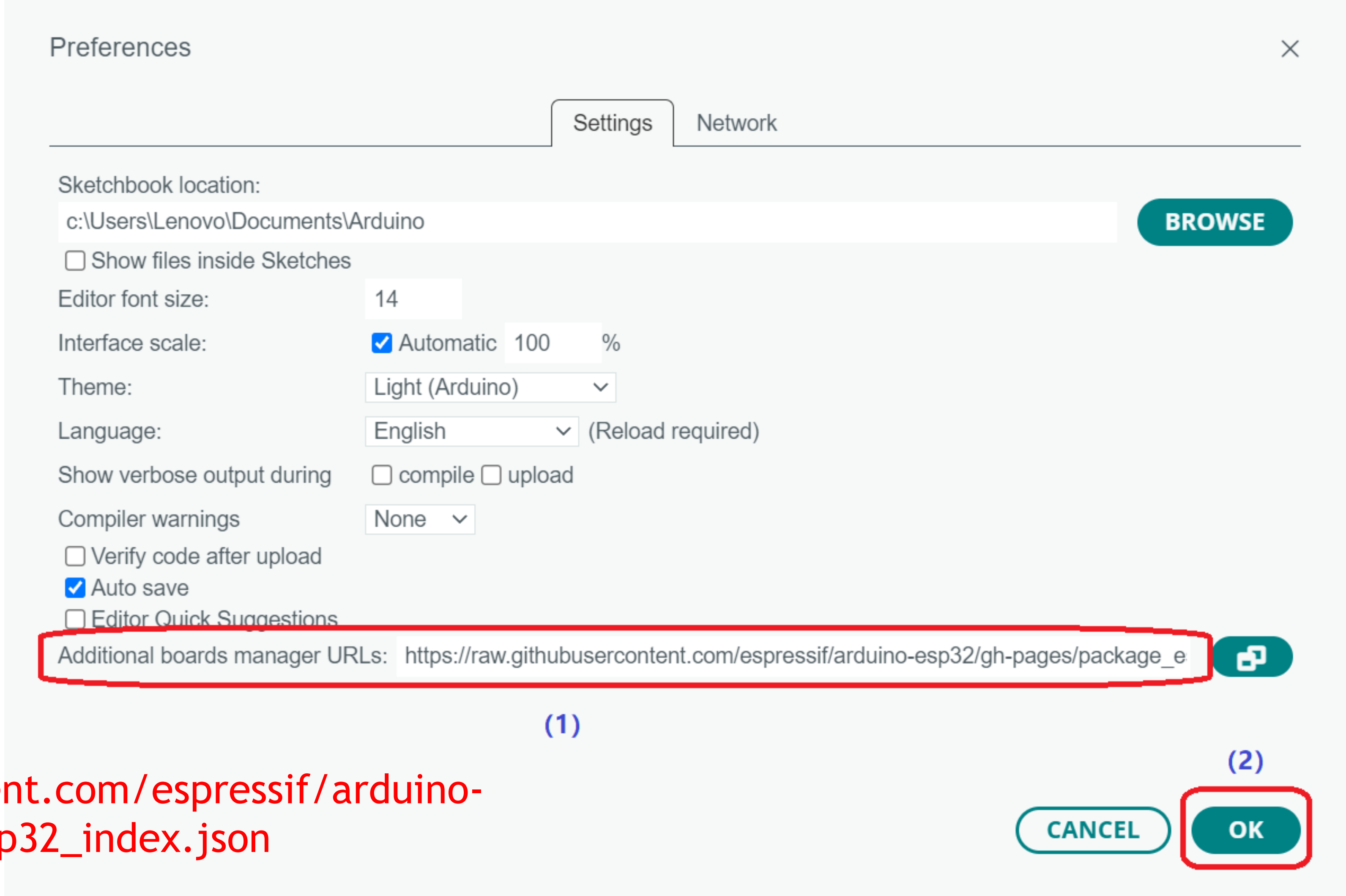
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to

Not connected. Select a board and a port to connect automatically.

Arduino IDE

- 在额外板卡管理器文本框中，填入如下URL，然后点击确定：

  Fill in the textbox of the Additional boards manager URLs with the URL below, then press OK:



Preferences ✕

Settings | Network

Sketchbook location:

c:\Users\Lenovo\Documents\Arduino BROWSE

☐ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light (Arduino)

Language: English (Reload required)

Show verbose output during ☐ compile ☐ upload

Compiler warnings None

☐ Verify code after upload

☑ Auto save

☐ Editor Quick Suggestions

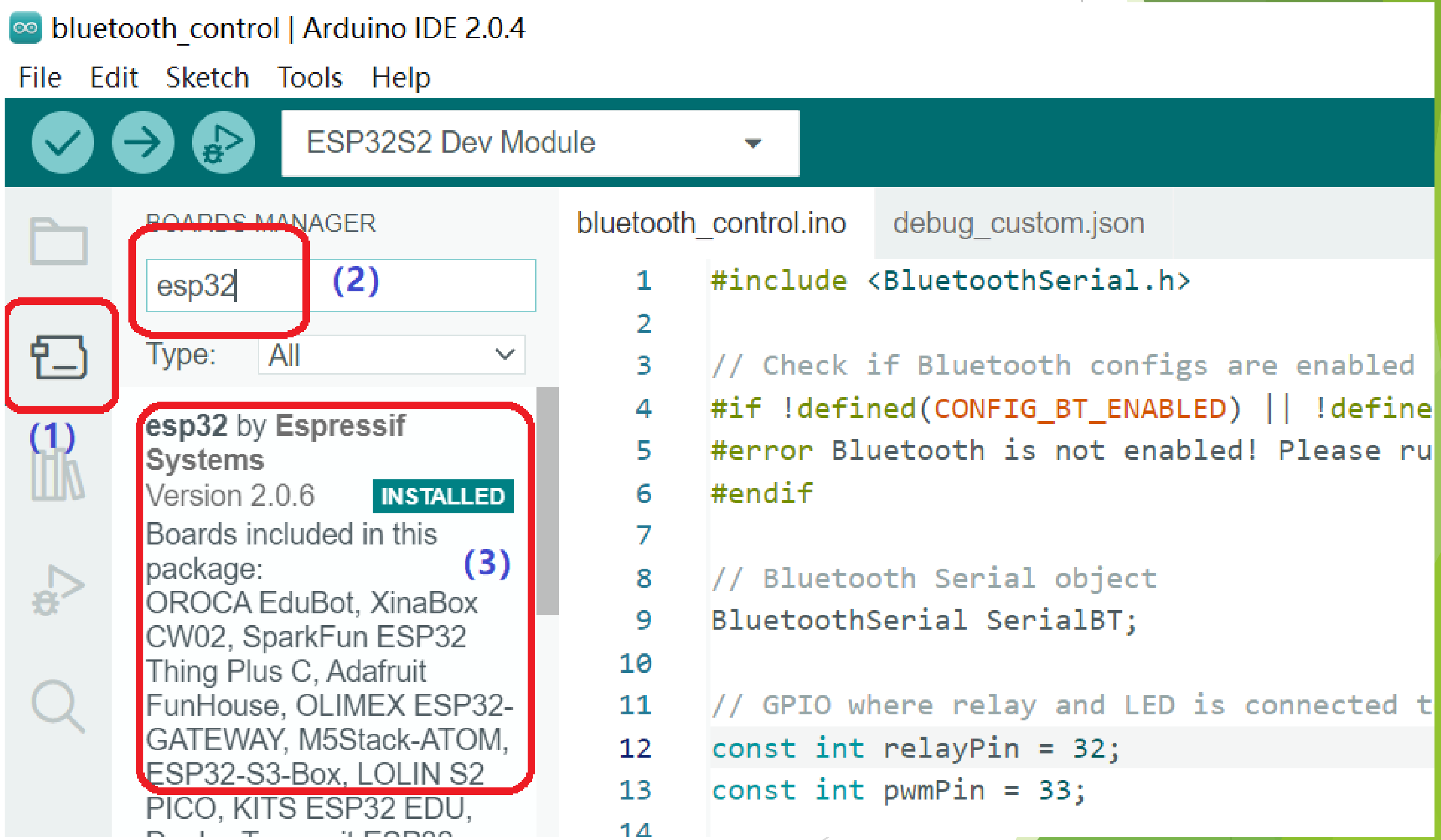
Additional boards manager URLs: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_e

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Arduino IDE

- 然后，点击板卡管理器，查找esp32板卡相关软件，然后进行安装：

  Then press the Board Manager, try to search the esp32 toolset, from the result install the esp32 tools:



bluetooth_control | Arduino IDE 2.0.4

esp32

Type: All

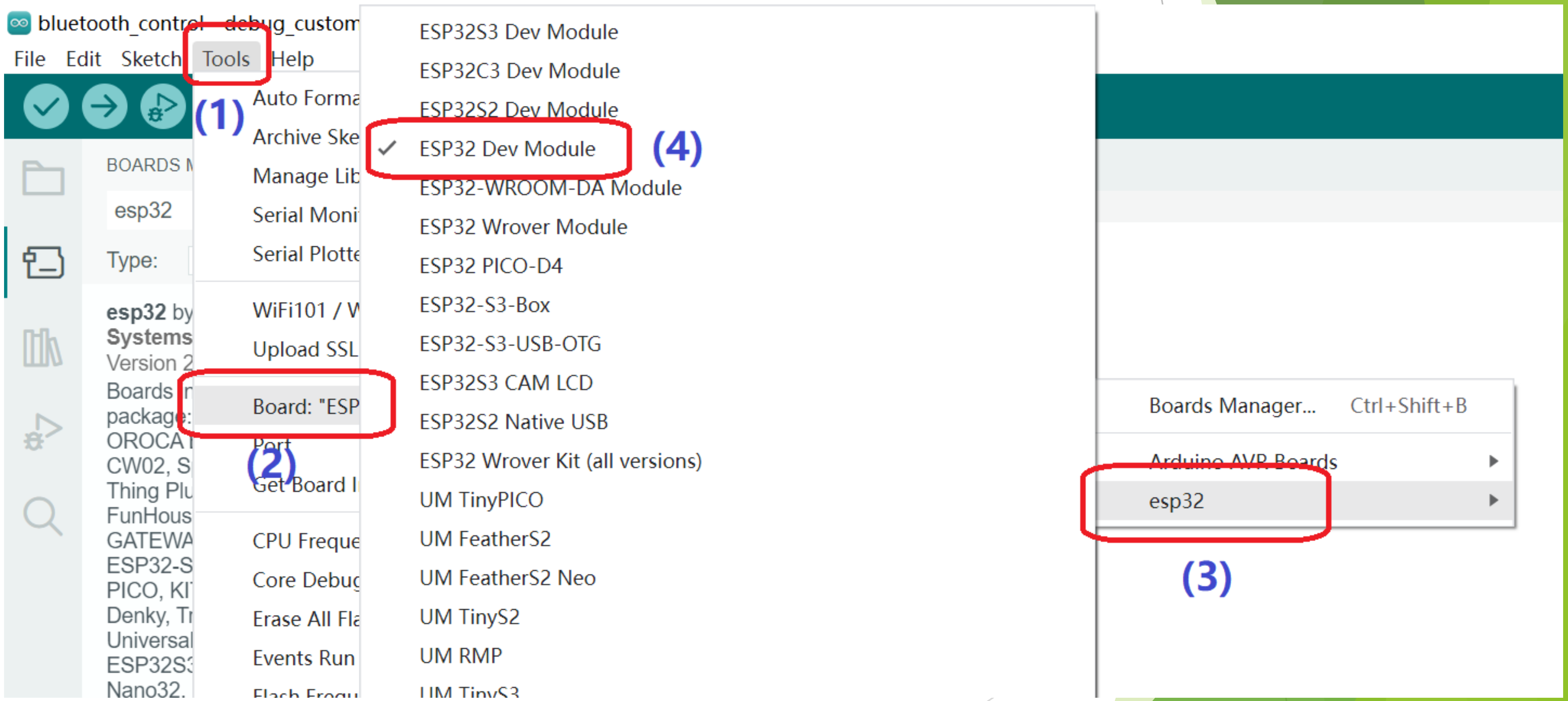
esp32 by **Espressif Systems**
Version 2.0.6 INSTALLED
Boards included in this package:
OROCA EduBot, XinaBox CW02, SparkFun ESP32 Thing Plus C, Adafruit FunHouse, OLIMEX ESP32-GATEWAY, M5Stack-ATOM, ESP32-S3-Box, LOLIN S2 PICO, KITS ESP32 EDU,

```
#include <BluetoothSerial.h>

// Check if Bluetooth configs are enabled
#if !defined(CONFIG_BT_ENABLED) || !define
#error Bluetooth is not enabled! Please ru
#endif

// Bluetooth Serial object
BluetoothSerial SerialBT;

// GPIO where relay and LED is connected t
const int relayPin = 32;
const int pwmPin = 33;
```

Arduino IDE

- 安装完毕后，从工具菜单，选择板卡子菜单，然后从下级子菜单中选择esp32，然后根据开发板选择具体类型：

  After installation of the tools, from the Tools menu choose the Board submenu, then choose the esp32 submenu, following by selecting the most appropriate board type corresponding to what you have:



Arduino IDE

- 我们通常用Micro-USB或USB Type-C将开发板与主机连接。一般开发板上均有USB转串口模块，方便主机与开发板通信。常用的USB转串口有两款，相应的驱动如下所示：

  We usually use Micro-USB or USB Type-C to connect the board with the host. Generally, there is a USB-to-serial module on the board to facilitate the communication between the host and the board. There are two commonly used USB to serial modules, and the corresponding drivers are as follows:

  - 一类是Silicon Lab CP210x系列，读者可以从以下网站找到驱动：

    One is the Silicon Lab CP210x series, users can find the driver from the following website:

    https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers

  - 一类是WCH CH340系列，读者可以从以下网站找到驱动：

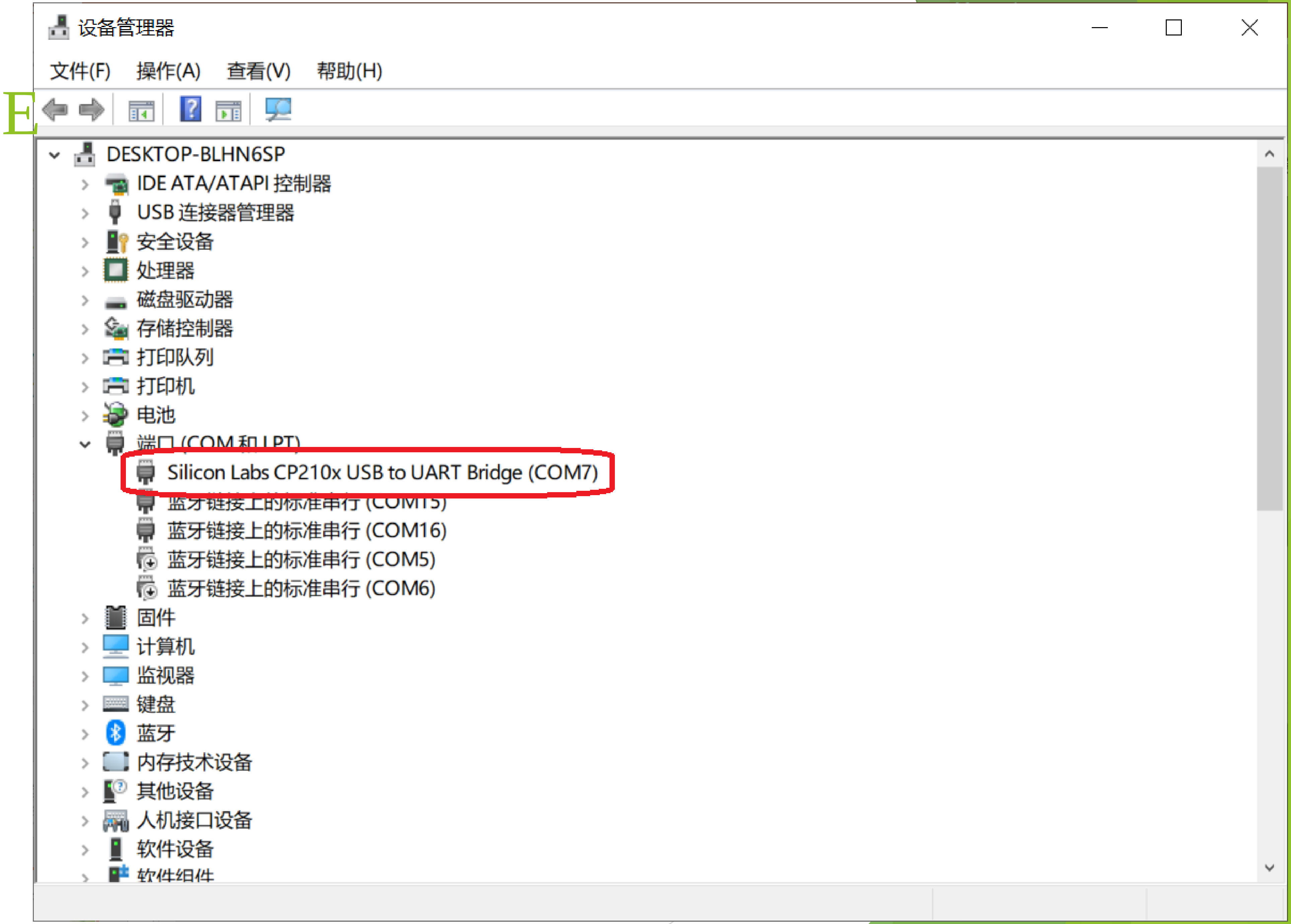
    One is WCH CH340 series, readers can find the driver from the following website:

    https://www.wch.cn/downloads/USBSER_ZIP.html

Arduino IDE

- 在安装完驱动之后，我们可以连接开发板与主机，正常情况下会出现正确识别的COM端口：

  After installing the driver, we can connect the board to the host, and in the usual case, a correctly identified COM port will appear in the device manager:

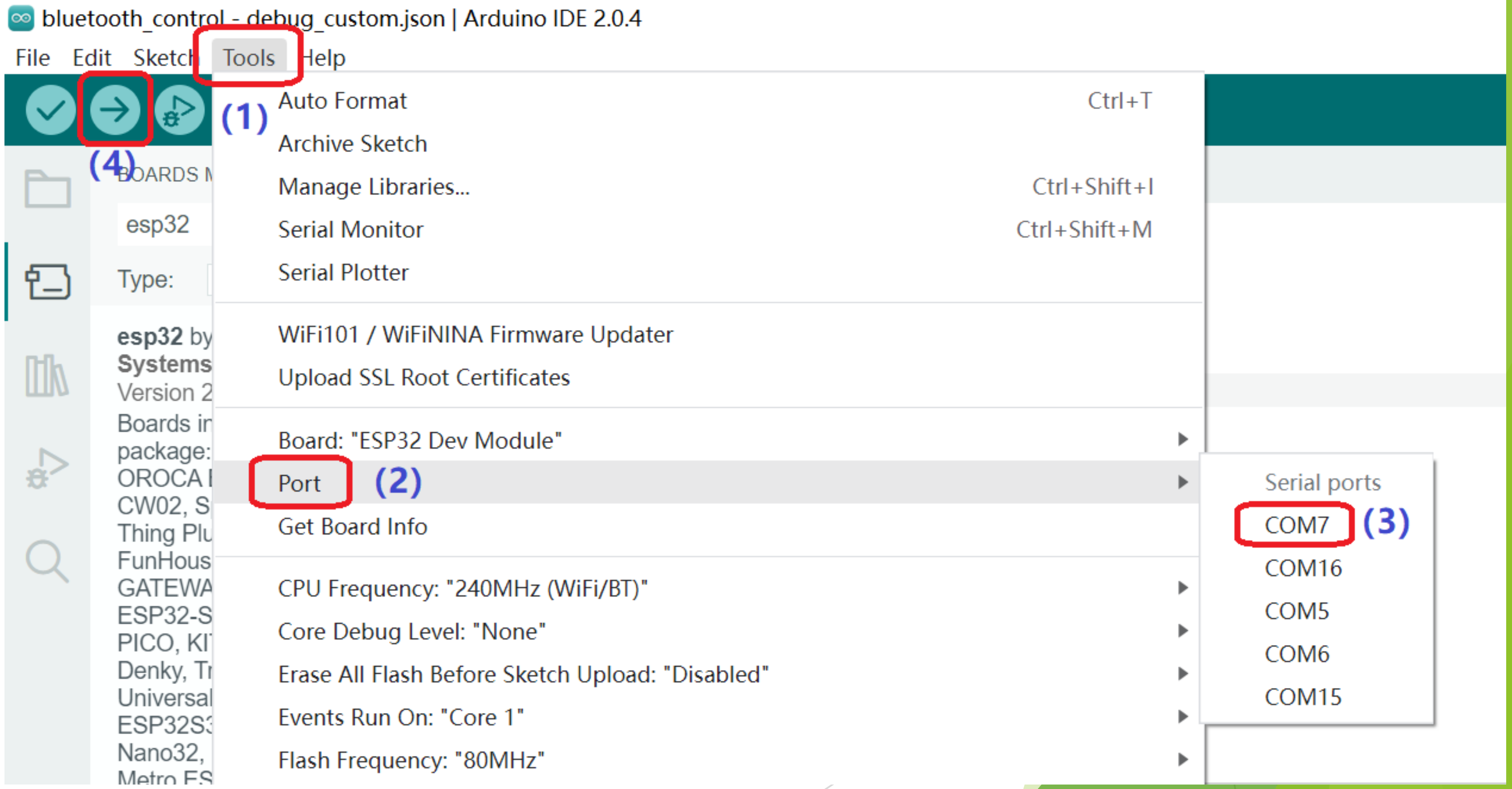


DESKTOP-BLHN6SP
- IDE ATA/ATAPI 控制器
- USB 连接器管理器
- 安全设备
- 处理器
- 磁盘驱动器
- 存储控制器
- 打印队列
- 打印机
- 电池
- 端口 (COM 和 LPT)
  - Silicon Labs CP210x USB to UART Bridge (COM7)
  - 蓝牙链接上的标准串行 (COM15)
  - 蓝牙链接上的标准串行 (COM16)
  - 蓝牙链接上的标准串行 (COM5)
  - 蓝牙链接上的标准串行 (COM6)
- 固件
- 计算机
- 监视器
- 键盘
- 蓝牙
- 内存技术设备
- 其他设备
- 人机接口设备
- 软件设备
- 软件组件

Arduino IDE

- 当在设备管理器中识别出COM端口之后，选择工具菜单，从工具菜单中选择端口子菜单，在端口子菜单中选择对应的COM口。当选定之后编写代码，当所有完成之后，选择如图所示箭头按钮，编译后上传至板子上运行。

  After identifying the COM port in the device manager, select the Tools menu, following by the Port submenu, then choose the corresponding COM port. After complete the code, select the arrow button as shown in the figure to compile and upload to the board to run.



Board: "ESP32 Dev Module"

CPU Frequency: "240MHz (WiFi/BT)"

Core Debug Level: "None"

Erase All Flash Before Sketch Upload: "Disabled"

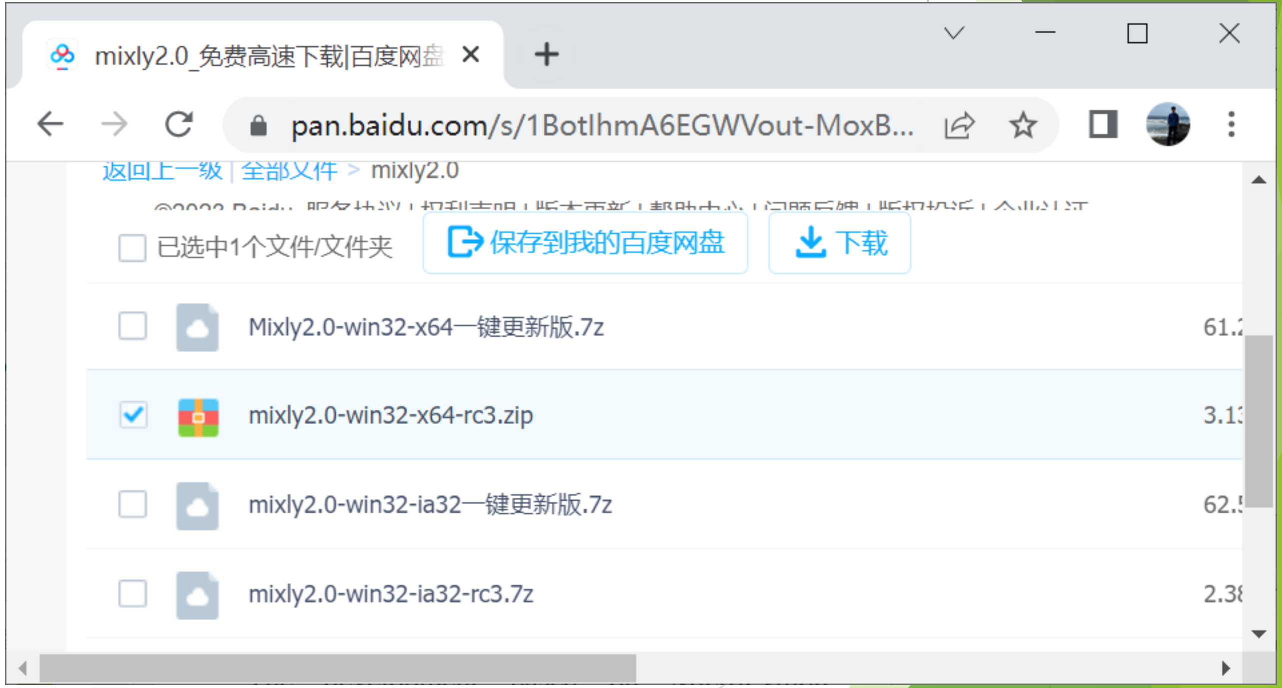
Events Run On: "Core 1"

Flash Frequency: "80MHz"

MicroPython

- 基于MicroPython的开发基本是在主机上进行基于Python开发的范式移到了开发板上，因此第一步需要找一个适配于Python开发的环境，这里推荐使用Mixly，最新的为Mixly2.0 RC3，相关资料地址为https://mixly.org/bnu-maker/mixl2.0rc。

  The development based on MicroPython is basically by migrating the paradigm of Python-based development on the host to the board, so the first step is to find an IDE suitable for Python development. Mixly is recommended here, and the latest is Mixly2.0 RC3. The information can be accessed here: https://mixly.org/bnu-maker/mixl2.0rc.



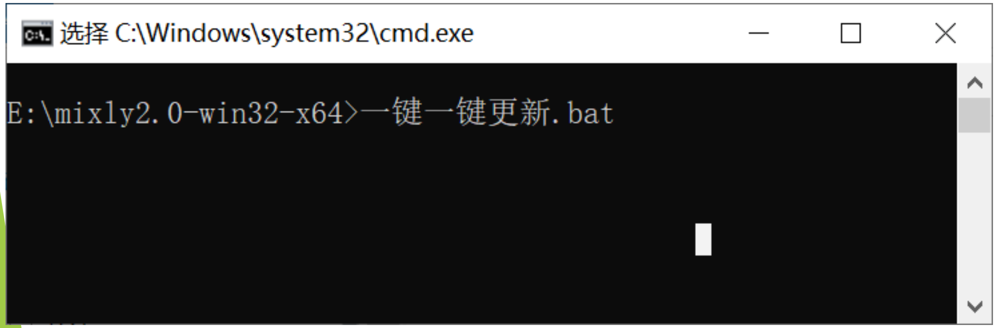
mixly2.0_免费高速下载|百度网盘
pan.baidu.com/s/1BotlhmA6EGWVout-MoxB...
返回上一级 | 全部文件 > mixly2.0
©2023 Baidu 服务协议 | 权利声明 | 版本更新 | 帮助中心 | 问题反馈 | 版权投诉 | 企业认证
保存到我的百度网盘
下载

□	已选中1个文件/文件夹	
□	Mixly2.0-win32-x64一键更新版.7z	61.2
☑	mixly2.0-win32-x64-rc3.zip	3.1
□	mixly2.0-win32-ia32一键更新版.7z	62.5
□	mixly2.0-win32-ia32-rc3.7z	2.3

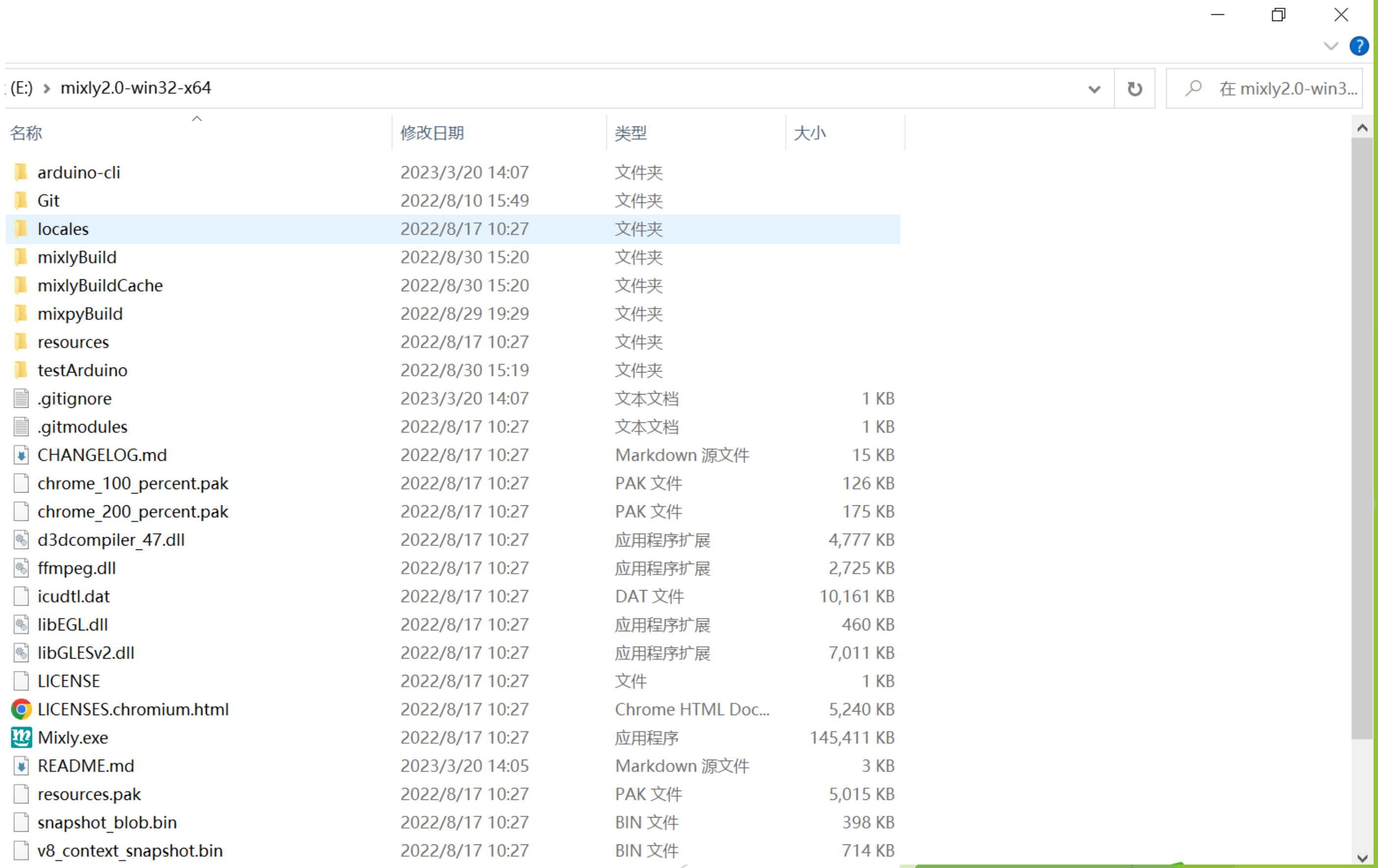
MicroPython

- 当将压缩包下载下来之后，解压到特定目录，然后进行一键升级工作：

  After downloading the compressed package, unzip it to a specific directory, and then perform a one-click upgrade:



```
E:\mixly2.0-win32-x64>一键一键更新.bat
```

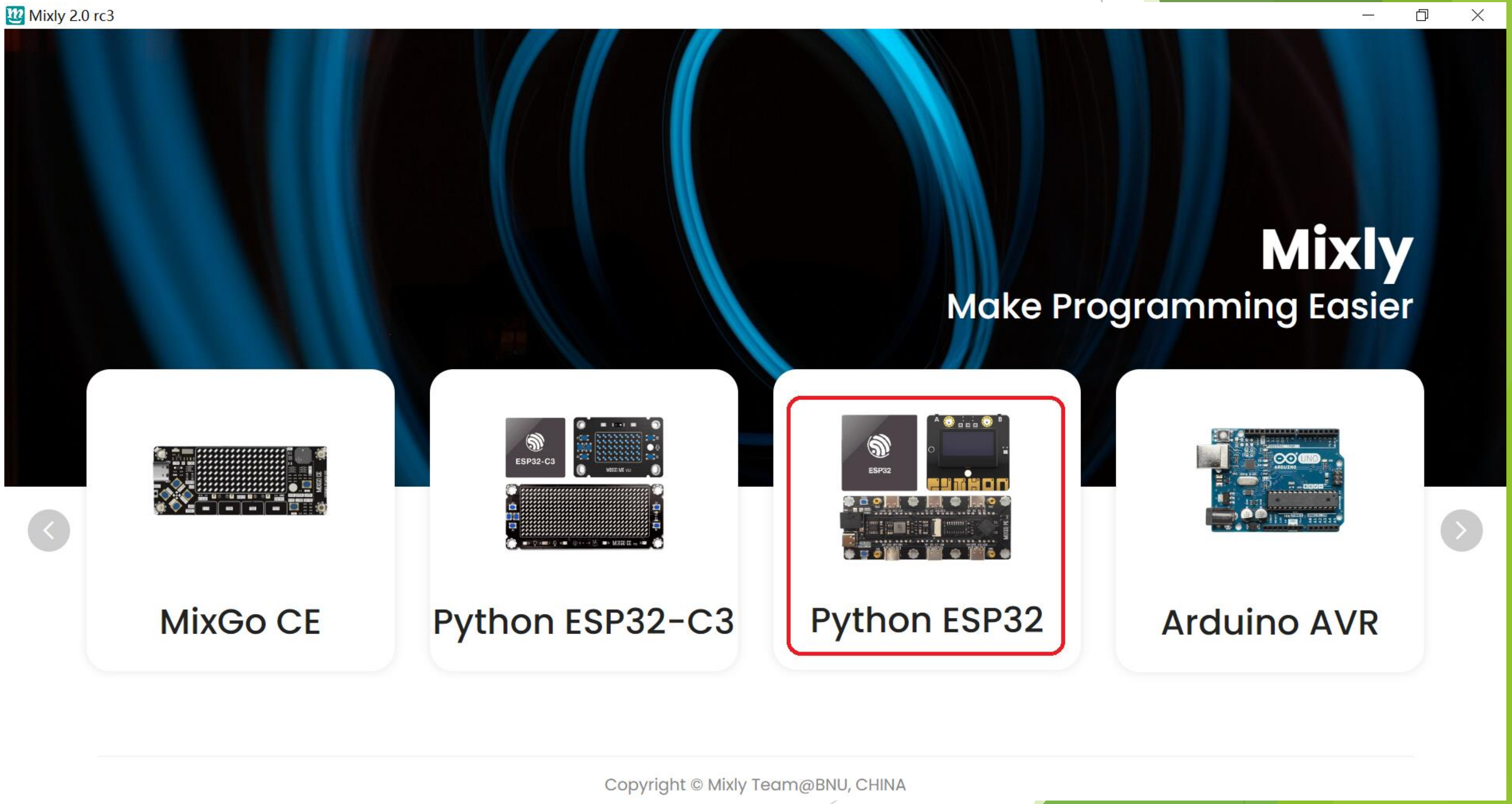


名称	修改日期	类型	大小
arduino-cli	2023/3/20 14:07	文件夹	
Git	2022/8/10 15:49	文件夹	
locales	2022/8/17 10:27	文件夹	
mixlyBuild	2022/8/30 15:20	文件夹	
mixlyBuildCache	2022/8/30 15:20	文件夹	
mixpyBuild	2022/8/29 19:29	文件夹	
resources	2022/8/17 10:27	文件夹	
testArduino	2022/8/30 15:19	文件夹	
.gitignore	2023/3/20 14:07	文本文档	1 KB
.gitmodules	2022/8/17 10:27	文本文档	1 KB
CHANGELOG.md	2022/8/17 10:27	Markdown 源文件	15 KB
chrome_100_percent.pak	2022/8/17 10:27	PAK 文件	126 KB
chrome_200_percent.pak	2022/8/17 10:27	PAK 文件	175 KB
d3dcompiler_47.dll	2022/8/17 10:27	应用程序扩展	4,777 KB
ffmpeg.dll	2022/8/17 10:27	应用程序扩展	2,725 KB
icudtl.dat	2022/8/17 10:27	DAT 文件	10,161 KB
libEGL.dll	2022/8/17 10:27	应用程序扩展	460 KB
libGLESv2.dll	2022/8/17 10:27	应用程序扩展	7,011 KB
LICENSE	2022/8/17 10:27	文件	1 KB
LICENSES.chromium.html	2022/8/17 10:27	Chrome HTML Doc...	5,240 KB
Mixly.exe	2022/8/17 10:27	应用程序	145,411 KB
README.md	2023/3/20 14:05	Markdown 源文件	3 KB
resources.pak	2022/8/17 10:27	PAK 文件	5,015 KB
snapshot_blob.bin	2022/8/17 10:27	BIN 文件	398 KB
v8_context_snapshot.bin	2022/8/17 10:27	BIN 文件	714 KB

MicroPython

- 升级完成后，双击Mixly.exe启动程序。为方便起见，也可以发送到桌面快捷方式。我们通常选择Python ESP32。

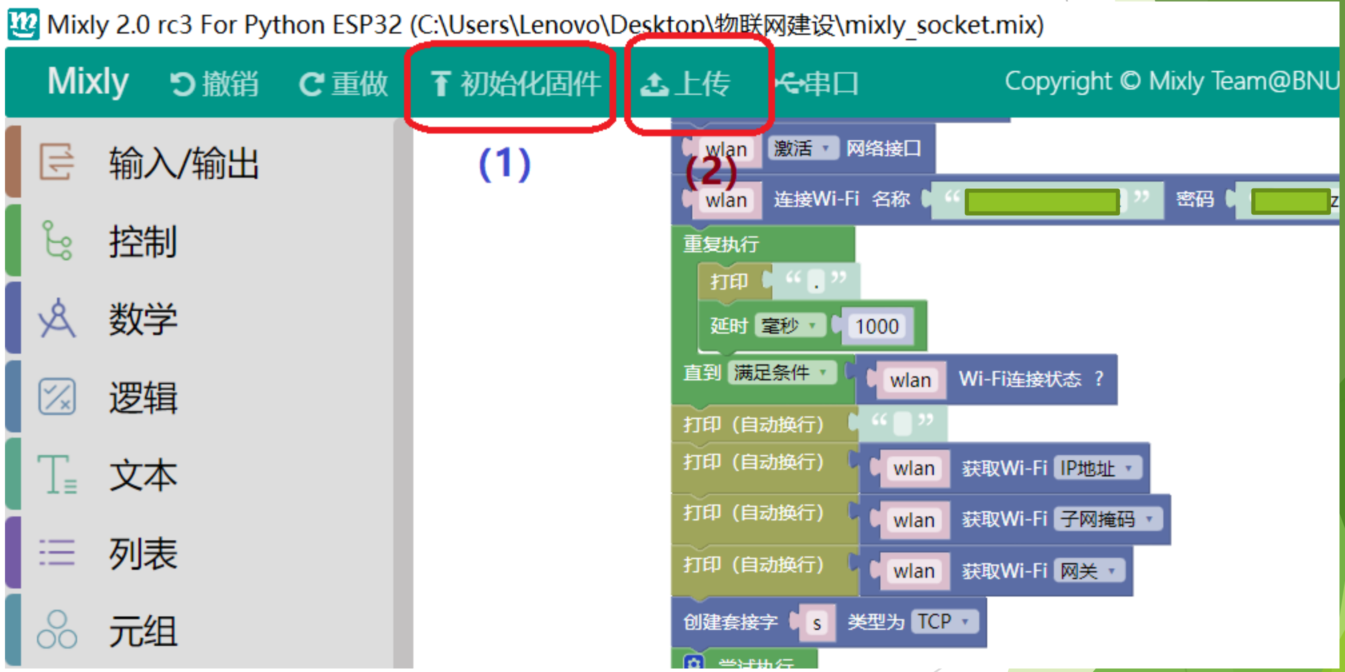
  After finish the upgrade, double-click Mixly.exe to launch the program. For convenience, you can also create a shortcut on the desktop. Generally, we just choose Python ESP32.



MicroPython

- 对于第一次使用开发板的情况，我们可能需要初始化固件。在初始化固件之后，便可以利用米思齐的可视化方式拖曳模块进行编程。在代码完成之后，便可以将代码上传到开发板上运行，完成相关作品制作。

  If you use the board for the first time, we may need to flash the firmware with MicroPython first. After flash the firmware, you can use Mixly's visual editor to program by dragging code blocks. After finish the code, you can upload the code to the board to run to make your design work.



Mixly 2.0 rc3 For Python ESP32 (C:\Users\Lenovo\Desktop\物联网建设\mixly_socket.mix)

```
wlan 激活 网络接口
wlan 连接Wi-Fi 名称 “[redacted]” 密码 [redacted]
重复执行
    打印 “.”
    延时 毫秒 1000
直到 满足条件 wlan Wi-Fi连接状态 ?
打印（自动换行） “ ”
打印（自动换行） wlan 获取Wi-Fi IP地址
打印（自动换行） wlan 获取Wi-Fi 子网掩码
打印（自动换行） wlan 获取Wi-Fi 网关
创建套接字 s 类型为 TCP
尝试执行
```