

首先，你需要先按照组装手册或视频来组装小车并正确连接线路。

1_Assembly_guide-组装指南	2023/8/16 17:45	文件夹
2_Arduino software-安装软件	2023/8/31 17:45	文件夹
3_Libraries-库文件	2023/9/4 9:55	文件夹
CH340 Driver File-MAC	2023/8/31 17:45	文件夹
CH340 Driver File-Windows	2023/8/31 17:45	文件夹

接下来按照下面教程正确创建编程环境。

1.安装 Arduino IDE

Arduino 软件

Arduino 集成开发环境（IDE）是 Arduino 平台的软件端。用于编写和上传代码到控制板。下面跟着教程安装 Arduino 软件（IDE）吧。

1.1 进入 Arduino 软件官网

在浏览器输入并点击转到 <https://www.arduino.cc/en/software> 网页，能看到以下网页位置：

HARDWARE

SOFTWARE

CLOUD

DOCUMENTATION ▼

COMMUNITY ▼

BLOG

ABOUT

Downloads



Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

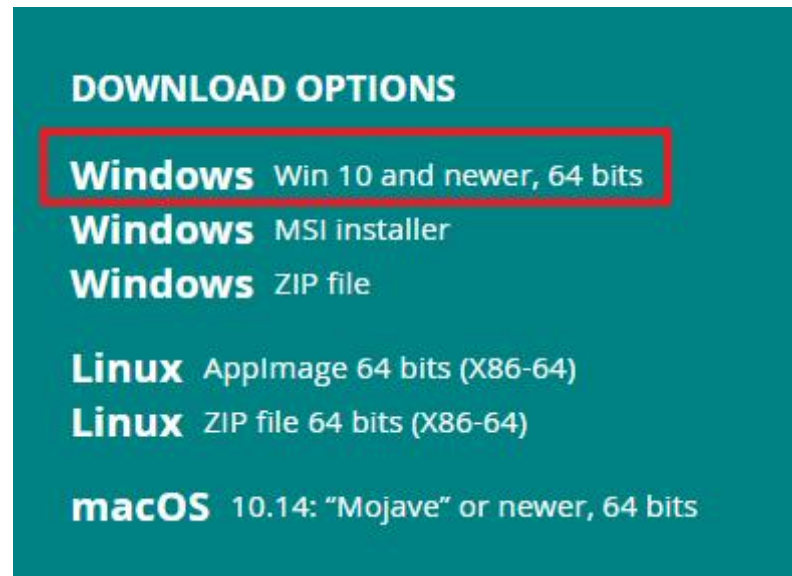
Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits

（在这里以 win10 系统安装 2.0.0 版本 IDE 为例，更低系统请将网页滑到下面安装 1.8.X 版本软件。同时，当看到本教程时可能该网站上更新的版本！）

1.2 选择软件适配的系统版本

选择与您电脑系统兼容的开发软件进行下载，这里以 Windows10 为例。



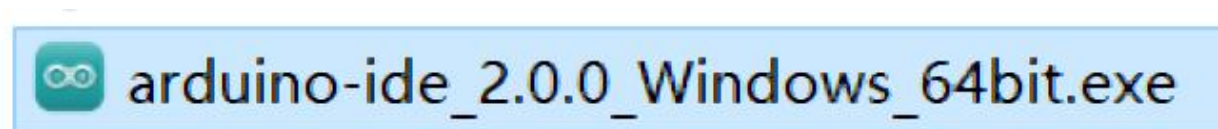
您可以在安装程序（.exe）和 Zip 包之间进行选择。我们建议您使用第一个“Windows Win10 and newer”直接安装使用 Arduino 软件（IDE）所需的一切，包括驱动程序。而使用 Zip 包，您需要手动安装驱动程序。当然如果要创建便携式安装，Zip 文件也很有用。

单击 “Windows Win10 and newer”



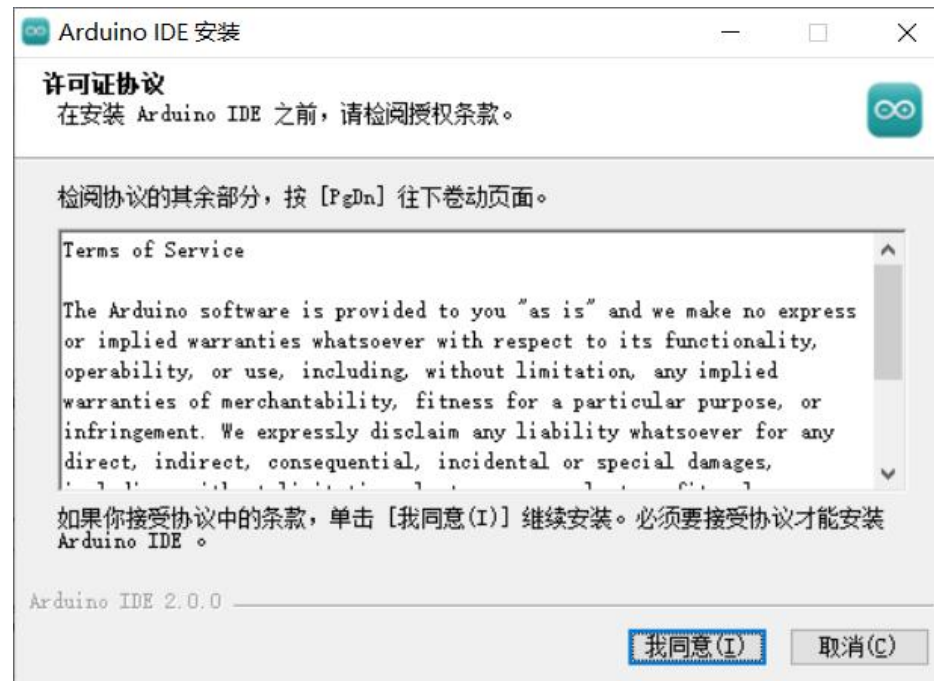
单击 “JUST DOWNLOAD” .

下载完成后会得 “exe” 后缀的安装包文件



1.3 正式安装 Arduino IDE


双击运行该安装程序



点击“我同意”



您可以按“浏览...”选择安装路径或直接输入你想要的目录。然后点击“安装”进行安装。（对于 Windows 用户，在安装过程中可能会弹出驱动安装对话框，当它弹出时，请允许安装）

安装完成后，桌面会生成一个 ArduinoIDE 软件快捷方式 ，双击即可进入 Arduino 软件平台环境。安装完成后打开软件即可看到软件平台界面如下图（不同版本界面会有所不同）：



编译/上传 - 编译并上传你的代码到你的 Arduino 板

选择板类型和端口号 - 检测到的 Arduino 板以及端口号会自动显示在这里

项目草图 - 在这里你会发现所有你的草图本地存储在你的电脑。此外，您可以与 Arduino 云同步，也可以从在线环境中获取您的草图

板管理器 - 浏览可以安装的 Arduino 和第三方软件包。例如，使用 MKR WiFi 1010 板需要安装 Arduino SAMD Boards 包

库管理器 - 浏览数以千计的由 Arduino 和它的社区贡献的 Arduino 库

调试 - 实时测试和调试程序

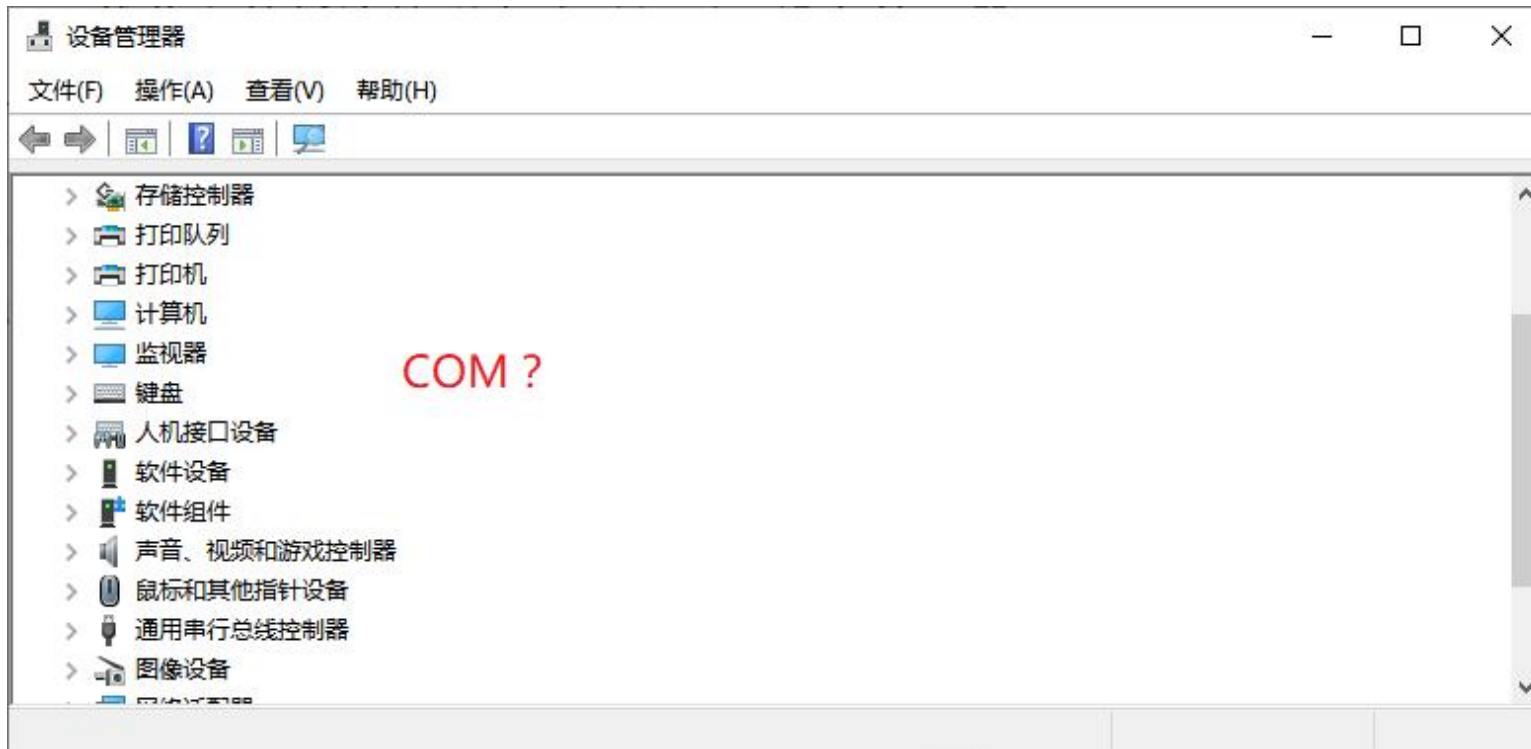
搜索 - 在代码中搜索关键字

打开串口监视器 - 作为控制台中的一个新选项卡方式打开 Serial Monitor 工具

使用 Arduino 软件（IDE）编写的程序称为“Sketch”。这些“Sketch”是在文本编辑器中编写的，并以文件扩展名为“.ino”保存。值得注意的是，对于“ino”文件必须保存于跟自己同名的文件夹中，如果不在同名文件夹时打开程序时会被强制要求自动建立一个同名文件。

1.4 安装 CH340 驱动

有时候比较新的电脑会缺少 CH340 串口驱动，用 USB 线将主控板连接到电脑，再搜索并打开“设备管理器”

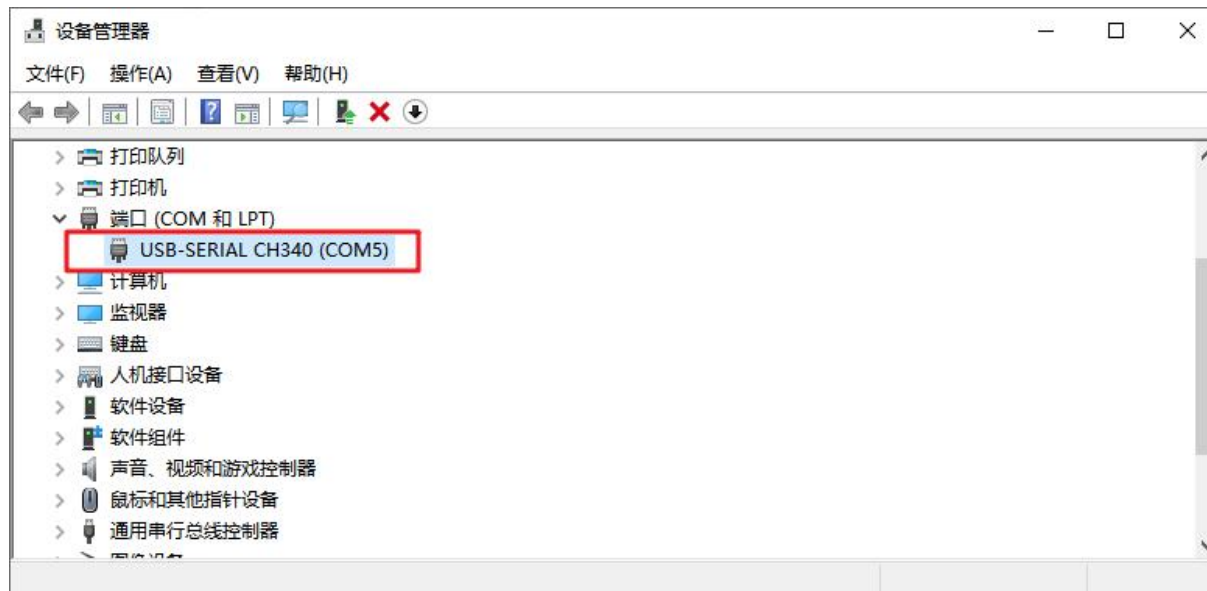


如上图没有看到 CH340 串口则需要按以下操作进行驱动安装。

打开文件夹 CH340 Driver File-Windows，双击 CH340 的 exe 程序安装包即可开始安装。

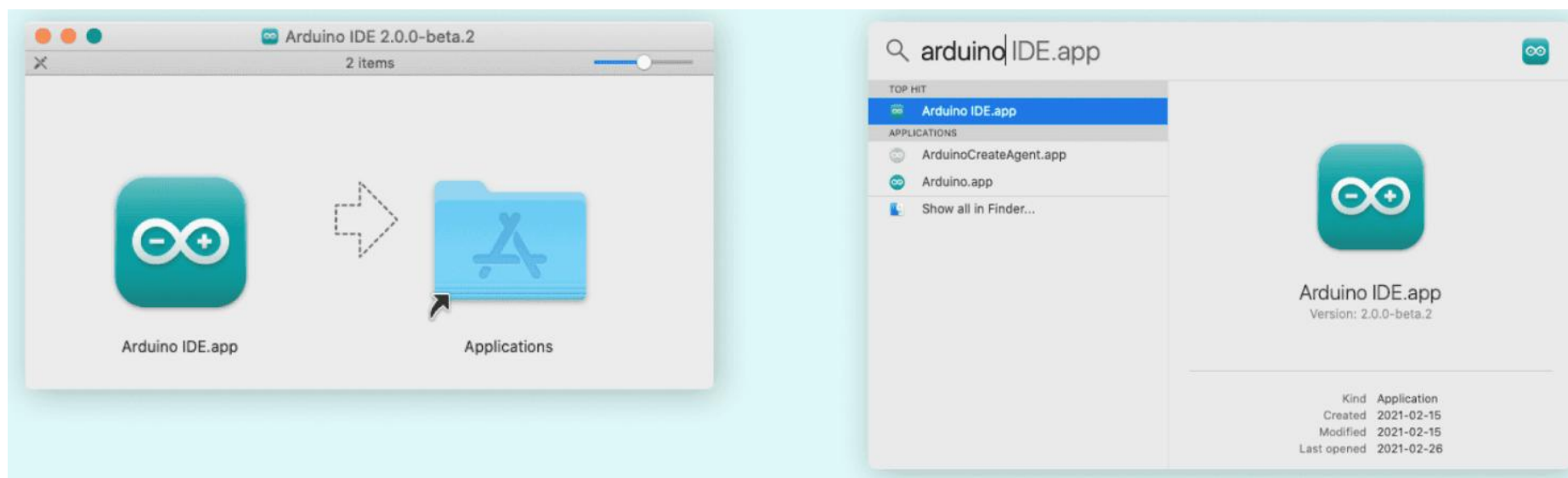
1_Assembly_robot	2023/4/27 11:02	文件夹
2_Arduino_IDE	2023/4/27 11:02	文件夹
3_Arduino_Libraries	2023/4/27 10:55	文件夹
CH340 Driver File-MAC	2023/4/27 11:23	文件夹
CH340 Driver File-Windows	2023/4/27 11:23	文件夹

安装完成后即可看到驱动已在设备管理器中显示（确保主控板与电脑正常连接）

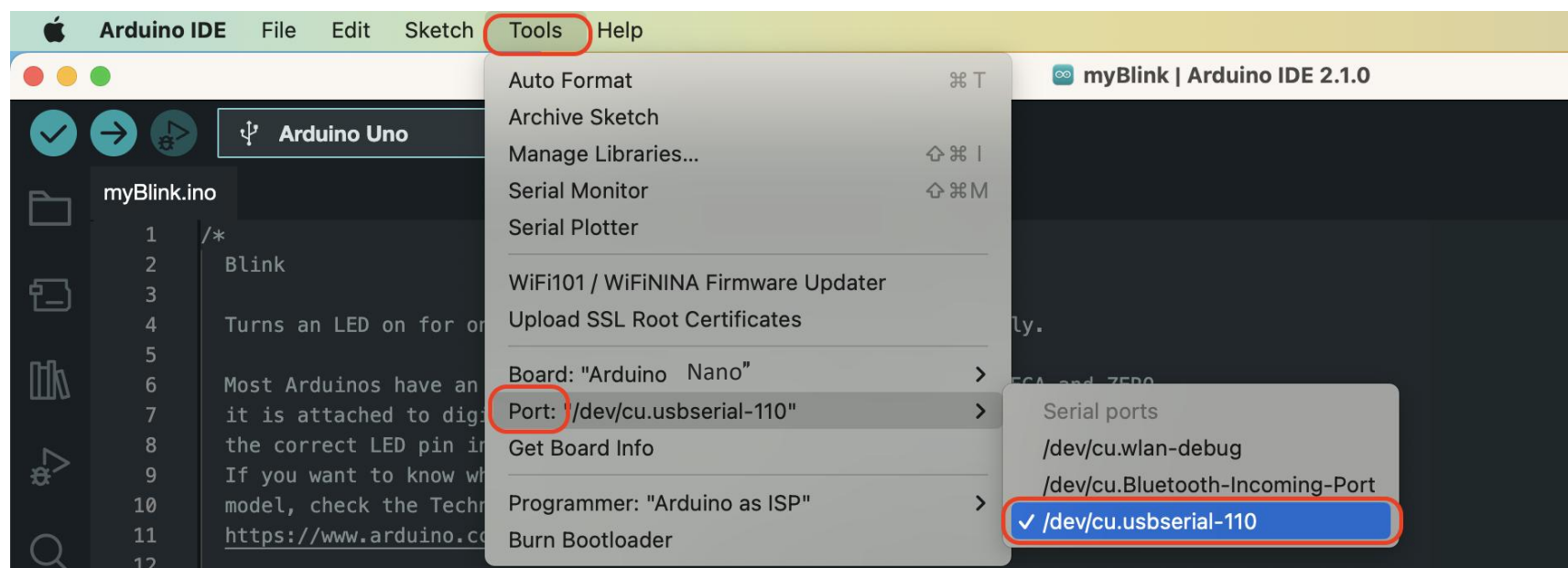


1.5 在 Mac OS X 系统安装 Arduino IDE

下载解压 zip 文件，双击 Arduino.app 进行安装；如果您的计算机中没有 Java 运行时库，系统会要求您安装它，安装完成后，您可以运行 Arduino IDE。



同样地，当你用 USB 线将主控板连接到电脑后发现软件识别出“USBserial”如下图,先选择板类型 Nano,



如果没有看到 USB 串口，你需要安装 CH340 驱动。

打开文件夹 CH340 Driver File-MAC，双击安装 pkg 文件

1_Assembly_robot	2023/4/27 11:02	文件夹
2_Arduino_IDE	2023/4/27 11:02	文件夹
3_Arduino_Libraries	2023/4/27 10:55	文件夹
CH340 Driver File-MAC	2023/4/27 11:23	文件夹
CH340 Driver File-Windows	2023/4/27 11:23	文件夹

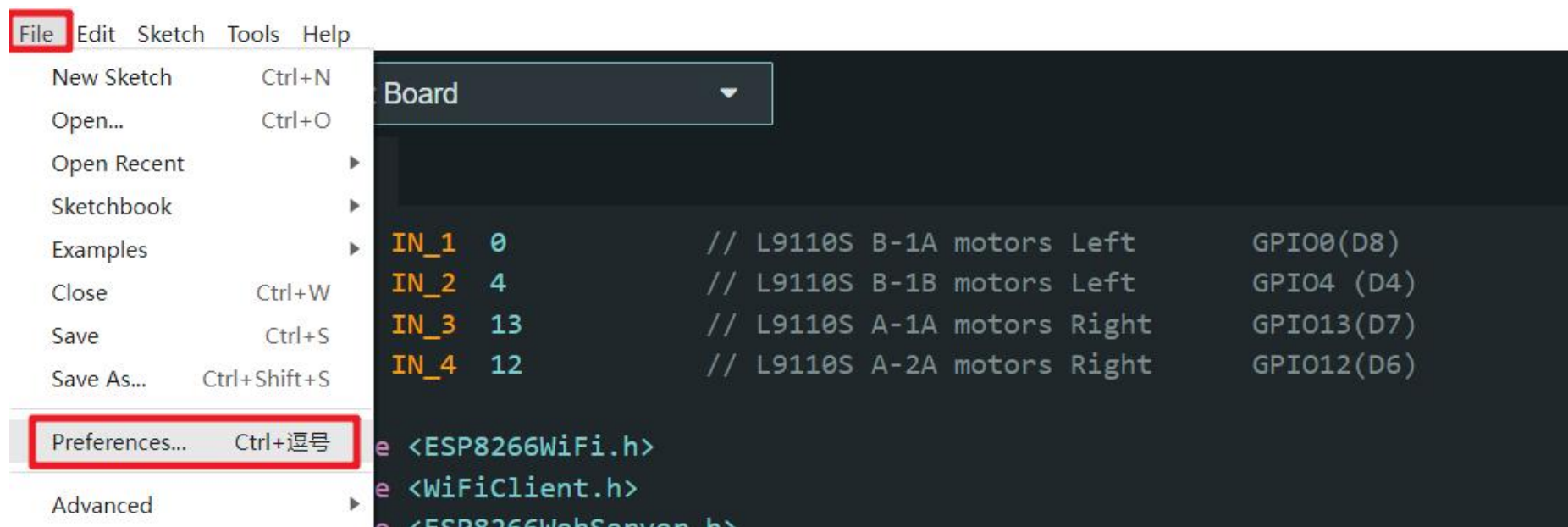
安装过程中如果电脑提示需要安装权限，就需要到“安全性与隐私”设置允许 APP 来自任何来源



2.在 Arduino IDE 中添加 ESP8266 板卡

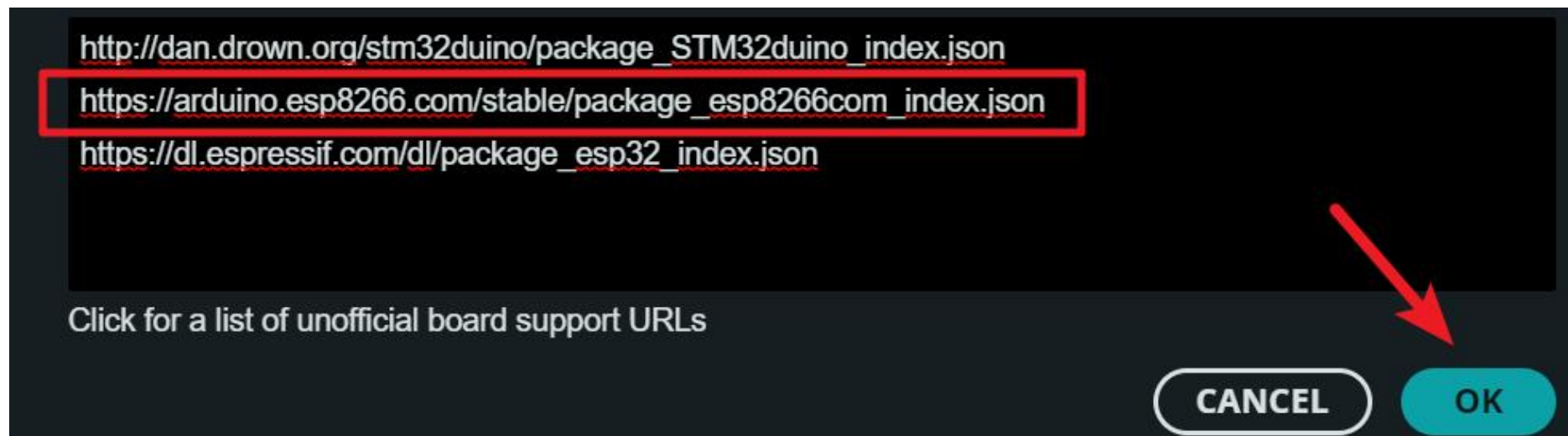
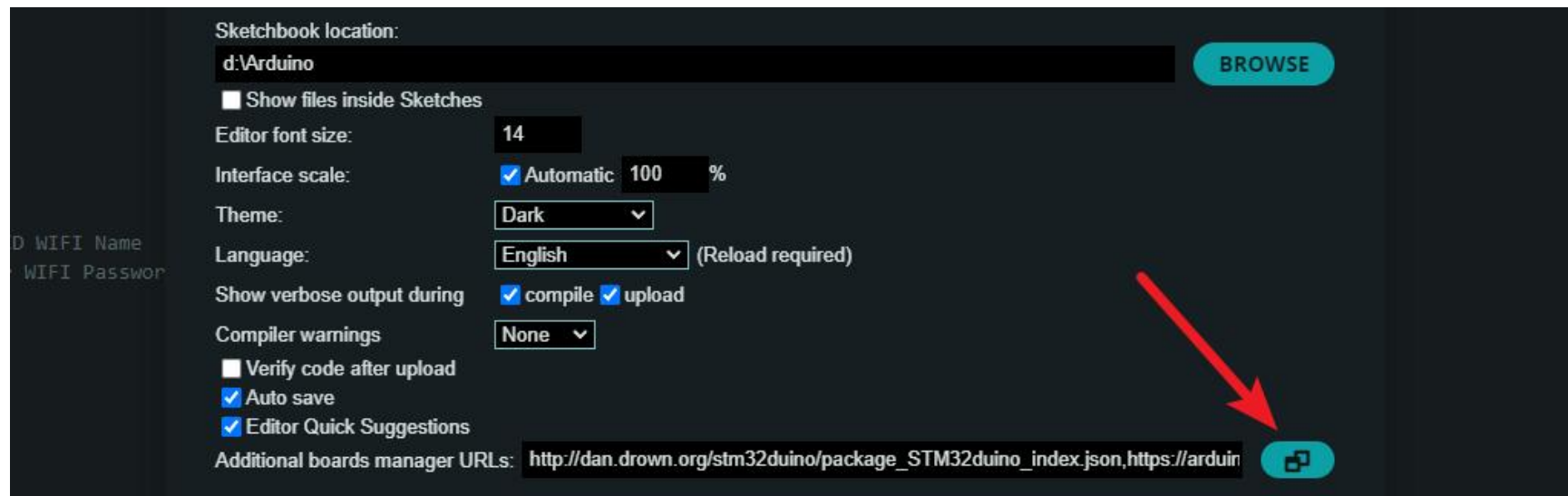
2.1 在“首选项”中添加板卡资源连接

点击 *File > Preferences* 打开 Preferences 窗口



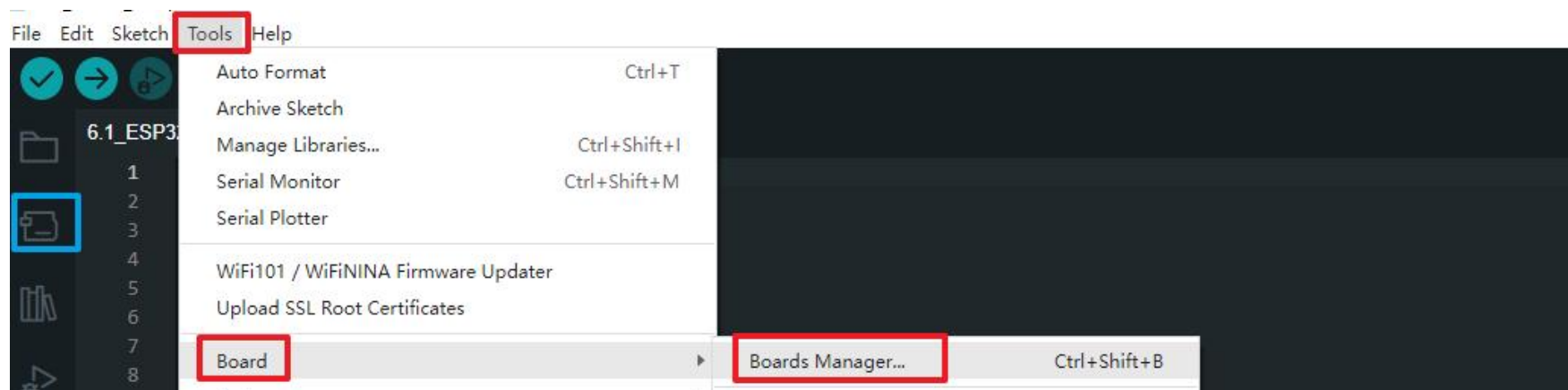
将以下资源链接复制并粘贴到“板卡管理 URLs”，完成后点击 OK 保存。

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

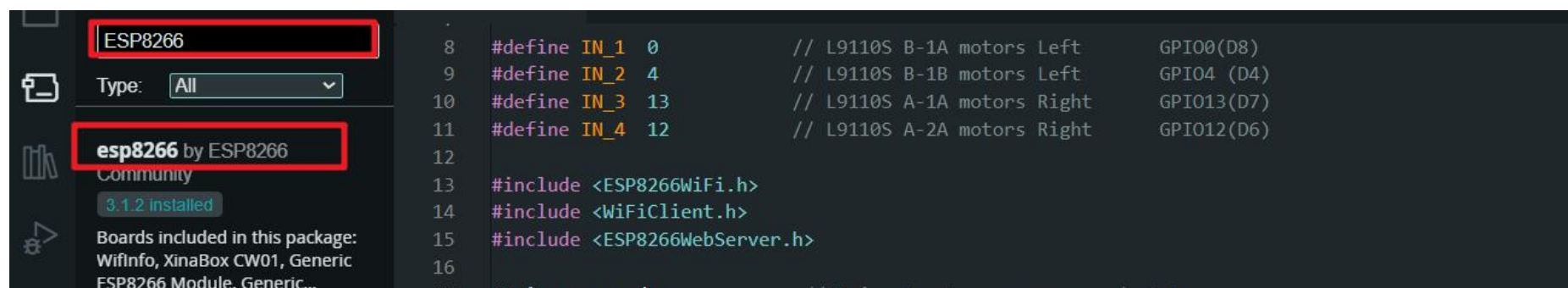


2.2 搜索并安装 ESP8266

打开“板管理”（新版 IDE 直接从左边“板管理”图标中打开）

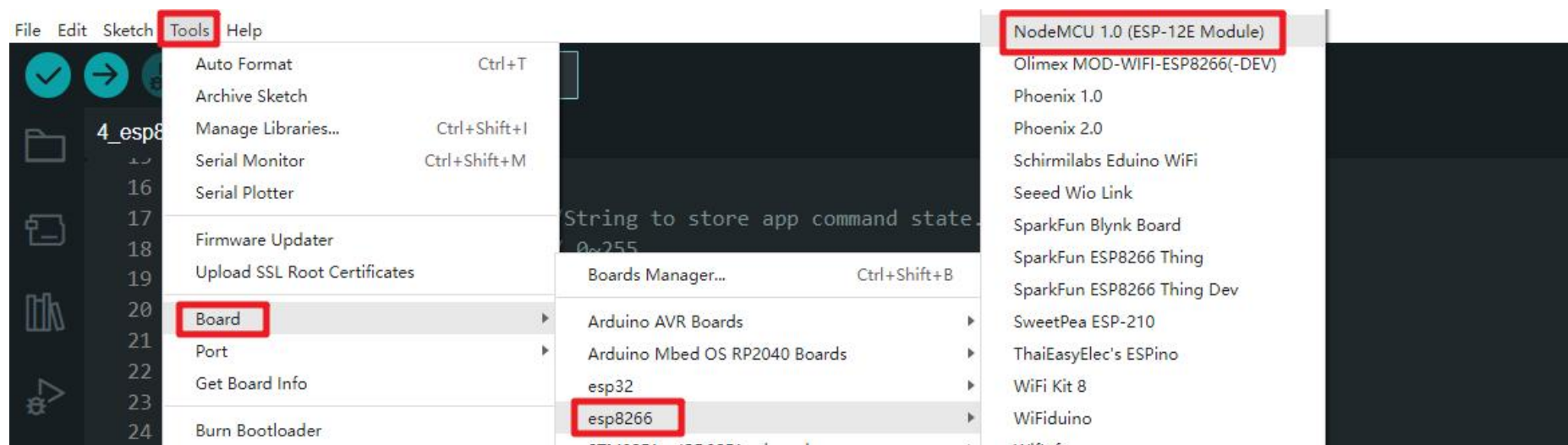


在搜索栏输入“ESP8266”，点击“install”进行安装（确保网络畅通，等待安装完成）



2.3 检查安装板卡成功

能看到“esp8266”出现在“Board”下



3.在 Arduino IDE 中添加库

3.1 如何在 Arduino IDE 中安装其他库

一旦您熟悉了 Arduino 软件并使用了内置功能，您可能希望通过其他库来扩展 Arduino 的功能。

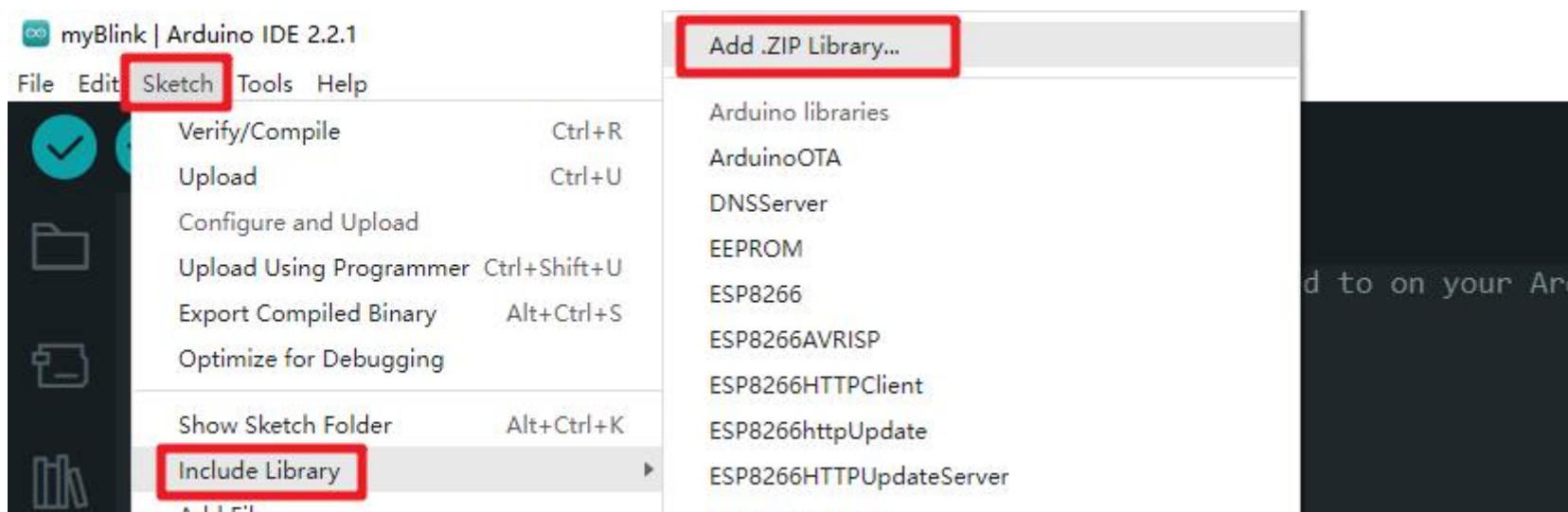
3.2 什么是 Libraries(库)?

库是一组代码，可让您轻松连接到传感器、显示器、模块等。例如，LiquidCrystal 库可让您轻松与字符 LCD 显示器交互。有数千个库可供直接通过 Arduino IDE 下载，您可以在 Library Manager 中找到所有这些库。

3.3 导入 .zip 库

库通常以 ZIP 文件或文件夹的形式分发。文件夹的名称是库的名称。该文件夹内将包含一个 .cpp 文件、一个 .h 文件，通常还有一个 keywords.txt 文件、示例文件夹和库所需的其他文件。从版本 1.0.5 开始，您可以在 IDE 中安装第 3 方库。不要解压缩下载的库，保持原样。

在 Arduino IDE 中，导航到 Sketch > Include Library > Add .ZIP Library，在下拉列表的顶部，选择“添加 .ZIP 库”选项。

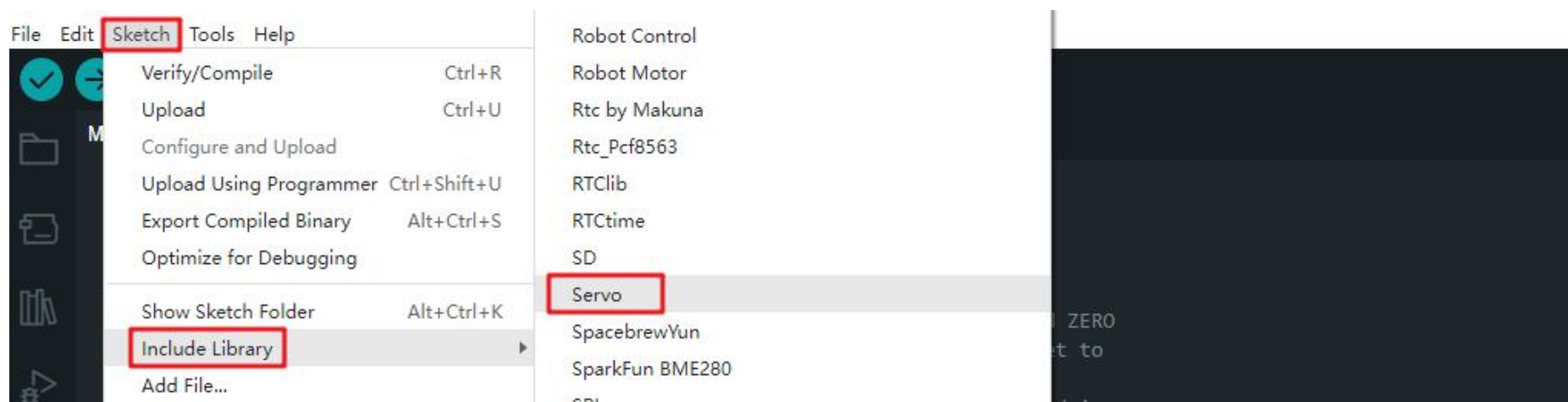


系统将提示您选择要添加的库，如下图导航到电脑中保存好的 servo.zip 文件的路径位置

(1 Get start | 3 Libraries | servo.zip) 并打开它。



打开 Sketch > Include Library 菜单。您现在应该在下拉菜单的底部看到库。它已准备好在您的草图中使用。



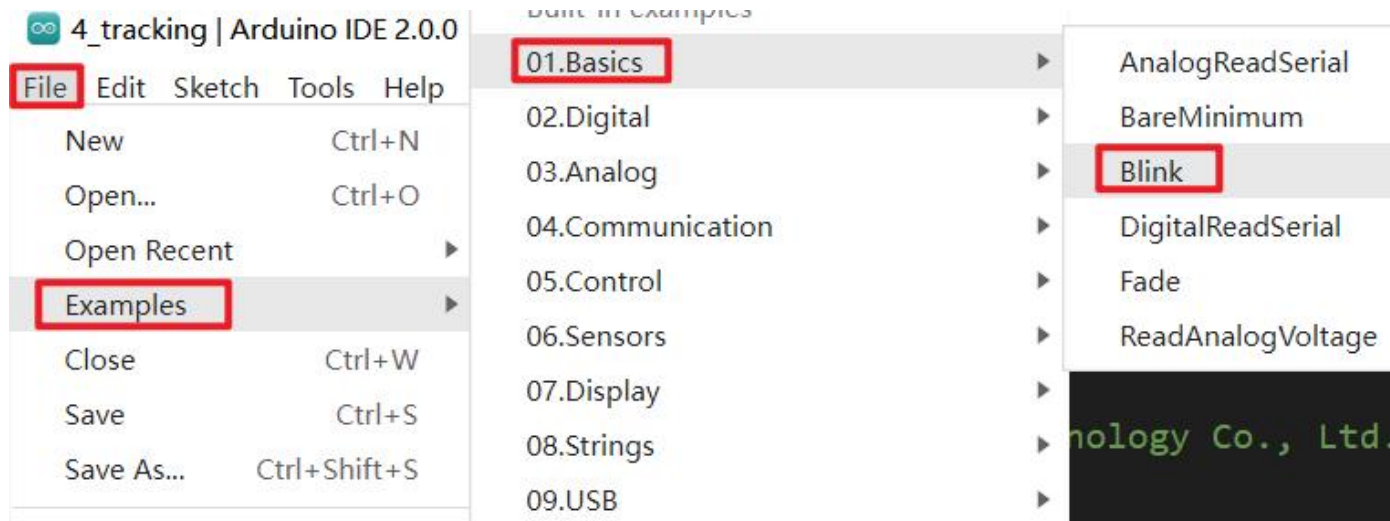
用此方法可以将所需的本地的库添加到 Arduino IDE 中。

4.开始第一个测试程序

制作属于自己的“Blink”草图

在这里我们将使用我们自己的 Blink 草图重新给主板编程, 然后更改它的闪烁速率。现在保持将主板连接到电脑, 设置 Arduino IDE 并确保您可以找到正确的串行端口, 并上传程序进行测试。

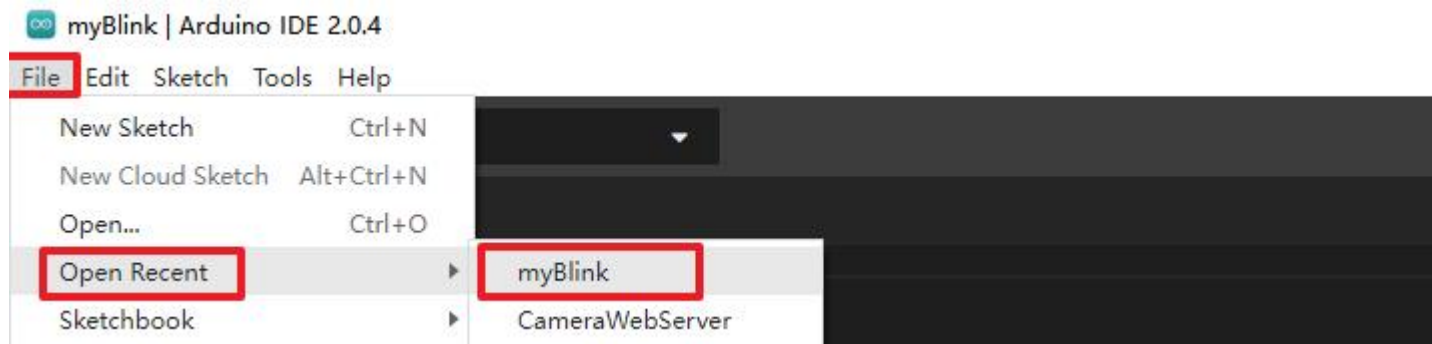
Arduino IDE 包含大量您可以加载和使用的示例草图, 其中包括一个用于制作“LED 的“眨眼”示例草图。在 IDE 中的菜单系统 File > Examples > 01.Basics 中您将找到的“Blink”草图。



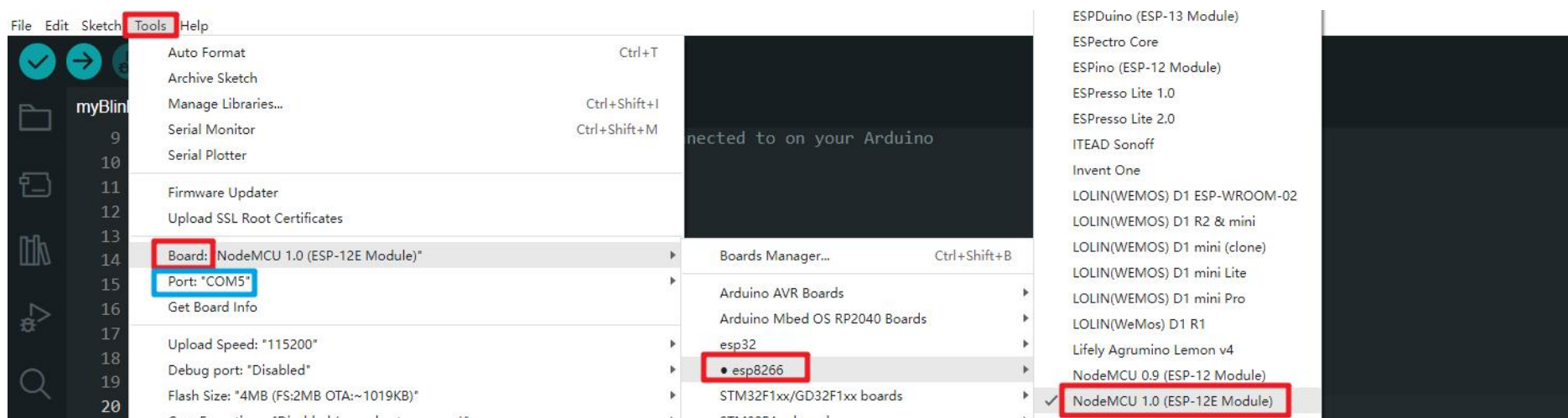
Arduino IDE 中包含的示例草图是“只读的”。也就是说, 您可以将它们上传到主控板上, 但如果您更改它们, 您不能将它们保存为同一个文件。因此您需要做的第一件事是保存自己的副本。

从 Arduino IDE 的文件菜单中, 选择“另存为..”, 然后将草图保存为“myBlink”。

您已将“闪烁”副本保存在速写本中, 如果您想再次找到它, 只需使用“File >Open Recent”菜单选项打开它。



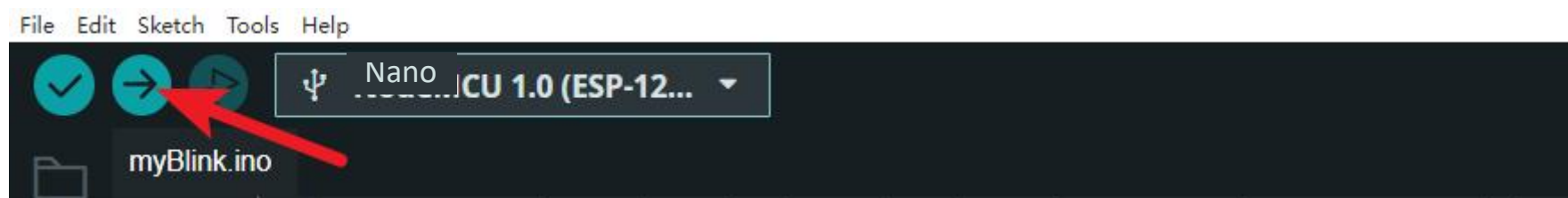
使用 USB 连接线将 Arduino 板连接到计算机并检查 Board Type 和 Serial Port 设置正确。



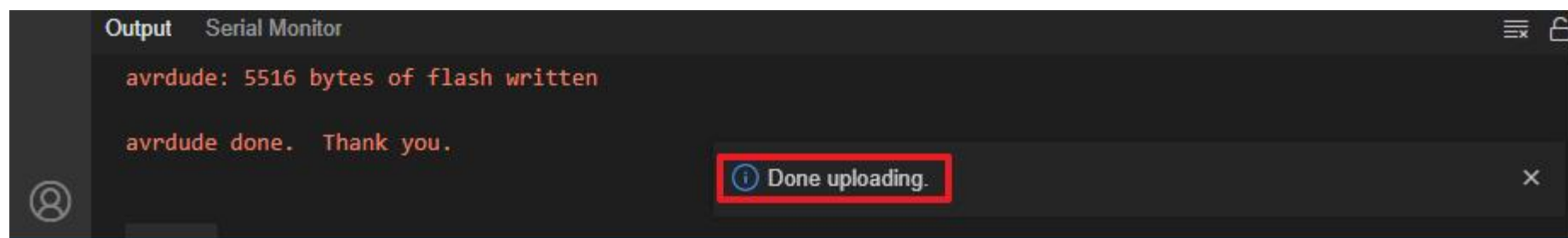
注意: 这里的板卡类型选择 **NodeMCU1.0** 和串口选择 **COM5**。

实际上串行端口显示为每个人都不一样, 尽管这里选择了 COM5, 但它可能是您计算机上的 COM3 或 COM4。

点击“上传”按钮后程序开始上传，此时，Arduino 上的 LED 会随着草图传输而开始闪烁。



传输完成，出现“上传完成”



在“编译草图..”过程中，您也许可能会收到错误消息：

这可能意味着您的板根本没有连接，或者 CH340 驱动程序没有安装（如果需要）或者串行端口选择错误。如果您遇到这种情况，请检查您的 IDE 设置和主板连接情况，截屏后寻求业务人员帮助。

上传完成后，电路板 LED 应重新启动并开始闪烁。请注意，此草图的很大一部分是由注释组成的。这些不是实际

的程序指令；相反，他们只是解释如何让该程序有效。他们在那里是为了您的易于阅读。草图顶部 “/*” 和 “*/” 之间的所有内容都是块注释，它解释了草图的用途。

单行注释以 “//” 开头，直到该行末尾的所有内容都被视为注释。

第一部分代码是：

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

每个草图都需要“设置”功能，也就是“Void setup()”函数，这是在按下重置按钮时执行的。每当电路板因任何原因而重置时，例如首次通电或上传草图后它就被执行。

接下来是给引脚命名和设置输出，这里设置“LED_BUILTIN”为输出端口。在大多数 Arduinos 上，包括 UNO，13 引脚是 LED 对应的引脚，而为了方便编程，程序已经将 LED_BUILTIN 变量设置为该引脚，所以直接使用不需要重新命名为 13 引脚。

草图还必须具有“循环”功能。与只运行一次的“设置”功能不同，在重置后，“循环”功能将在完成命令运行后立

即重新开始。

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

在循环函数内部, 命令首先打开 LED 引脚(高电平), 然后“延迟 1000 毫秒(1 秒)”, 然后关闭 LED 引脚并暂停一秒。

你现在要让你的 LED 闪烁得更快。正如您可能已经猜到的那样, 关键在于更改“delay()”中的参数。

```
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

此延迟时间以毫秒为单位, 因此如果您希望“LED”以两倍的速度闪烁, 请将值从“1000”更改为“500”。这将在每次延迟时暂停半秒, 而不是一秒。再次上传草图, 您应该会看到“LED”开始更快地闪烁。