ESP32 Arduino IDE2 开发环境搭建

目　录

1. Arduino IDE2 软件安装包下载 3
2. Arduino IDE2 软件安装 5
3. Arduino IDE2 软件介绍 9
3.1. 菜单栏 9
3.1.1. 文件菜单 9
3.1.2. 编辑菜单 11
3.1.3. 项目菜单 12
3.1.4. 工具菜单 13
3.1.5. 帮助菜单 16
3.2. 工具栏 17
4. 安装 Arduino-ESP32 核心软件库 18
4.1. Arduino IDE 开发板管理器在线安装 18
4.2. 手动离线安装 20
5. 编译、下载并运行 ESP32 示例程序 21
5.1. 安装 USB 转串口 IC 驱动 21
5.2. 配置开发板 22
5.3. 编译、下载并运行程序 25

1. Arduino IDE2 软件安装包下载

Arduino IDE 有两个版本：Arduino IDE1 和 Arduino IDE2。Arduino IDE1 是老版本，只支持 Win7 及以下系统，Arduino IDE2 是新版本，支持 Win10 及以上系统。在这里只介绍 Arduino IDE2。

Arduino IDE2 软件安装包可以直接从官网下载，地址为：

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

进入官网软件下载页面后，找到 Arduino IDE2 软件安装包下载栏目，如下图所示：

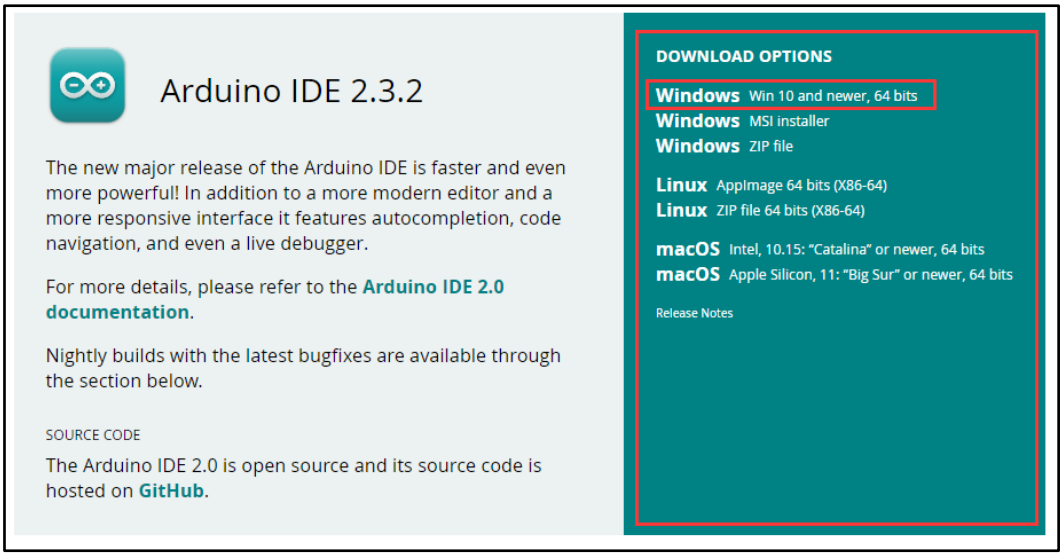

图 1.1 Arduino IDE2 软件安装包下载界面 1

从下载选项里根据自己的电脑系统选择相应的版本下载。例如使用 Windows 系统，那么直接点击“**Windows Win 10 and newer, 64 bits**”下载。还可以下载 ZIP 压缩包文件和 MSI 下载。

点击下载选项后，会弹出一个是否提供团队资金支持的界面，如下图所示：

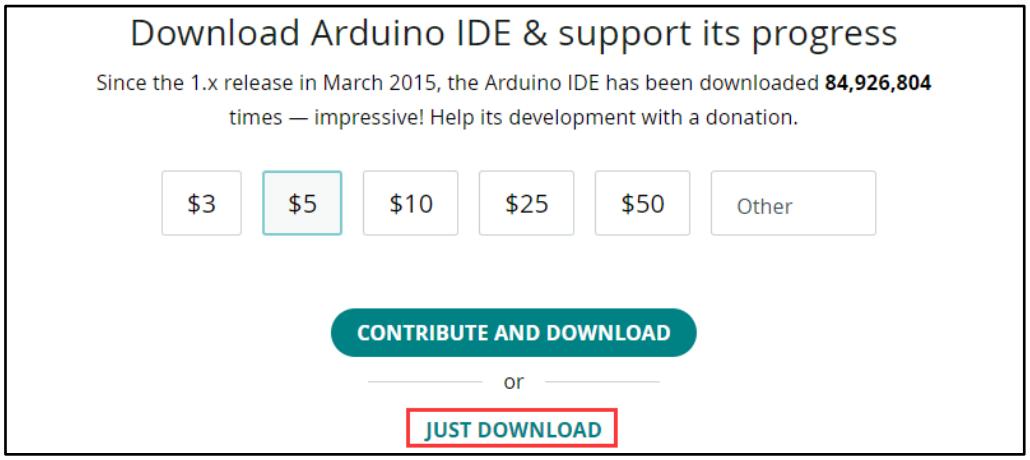

图 1.2 Arduino IDE2 软件安装包下载界面 2

可以忽略此选项，直接点击“JUST DOWNLOAD”按钮。

点击按钮后会弹出一个是否输入 email 接收 Arduino 信息的界面。可以忽略，直接点击“JUST DOWNLOAD”按钮，如下图所示：

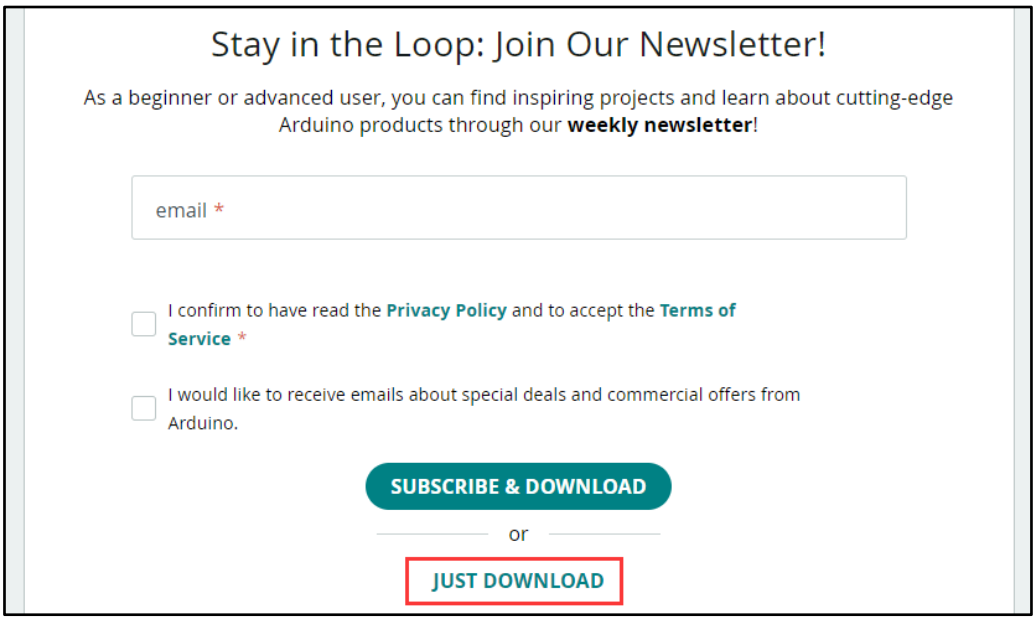

图 1.3 Arduino IDE2 软件安装包下载界面 3

点击按钮后会弹出一个“**新建下载任务**”窗口，点击“**浏览**”按钮选择软件安装包的保存路径，然后点击“**下载**”按钮，开始进入下载，如下图所示：

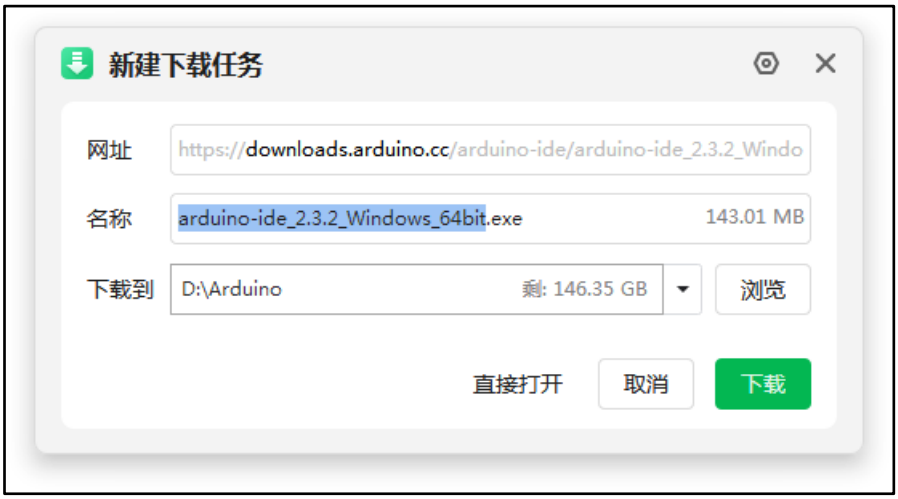

图 1.4 Arduino IDE2 软件安装包下载任务

2. Arduino IDE2 软件安装

找到 Arduino IDE2 软件安装包保存路径，然后双击 exe 文件，进入程序安装，如下图所示：

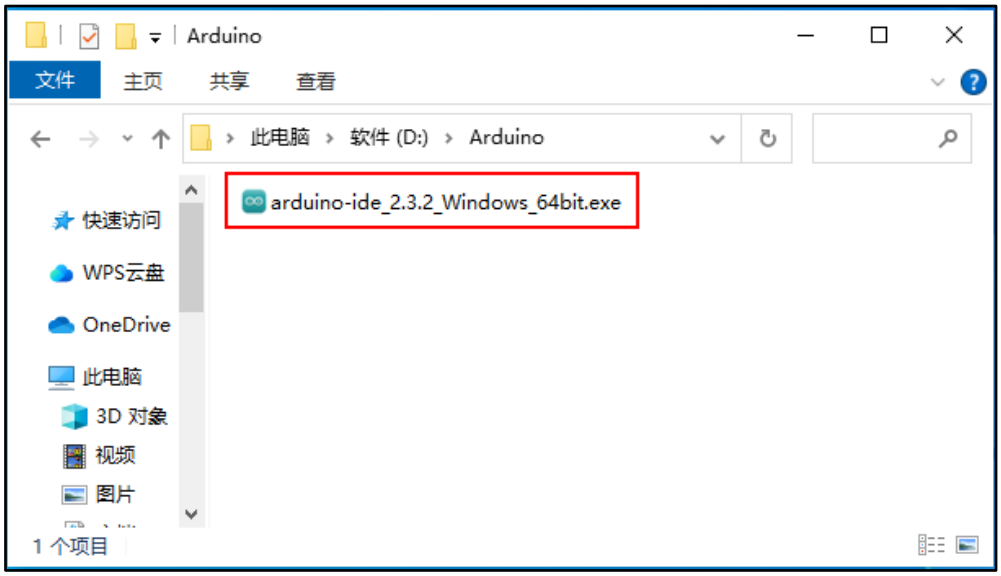

图 2.1 Arduino IDE2 软件安装包 exe 文件

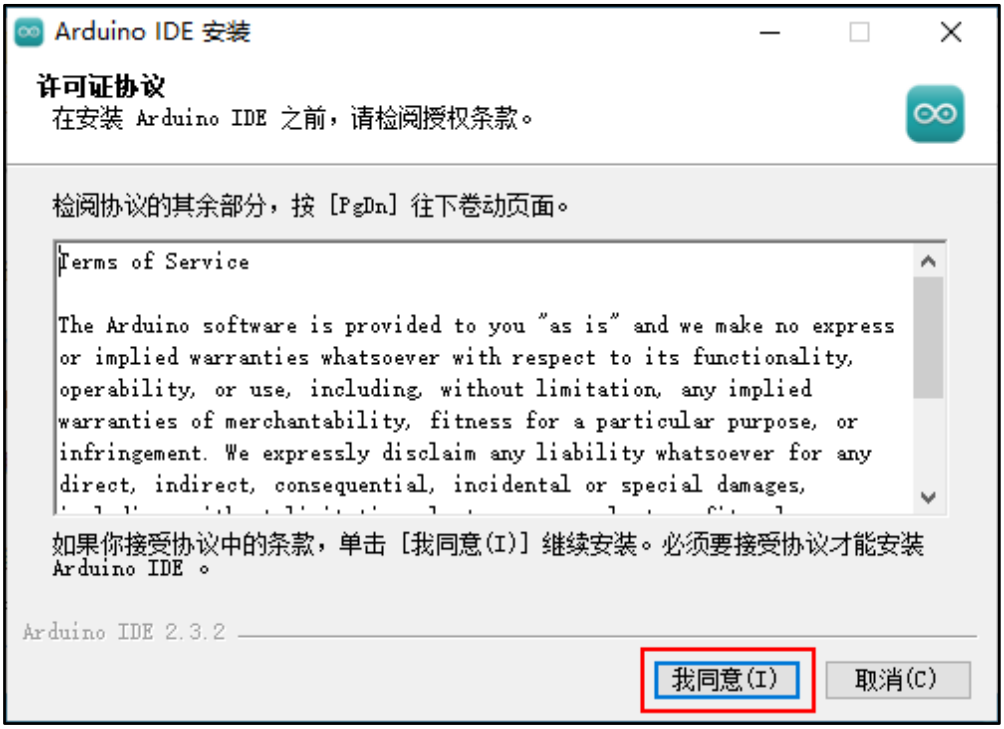

图 2.2 Arduino IDE2 软件安装许可协议

点击“**我同意**”按钮，进入安装用户选择界面，选择需要选择为当前用户安装还是所有用户安装，一般情况下选择“**为使用这台电脑的任何人安装（所有用户）**”如下图所示：

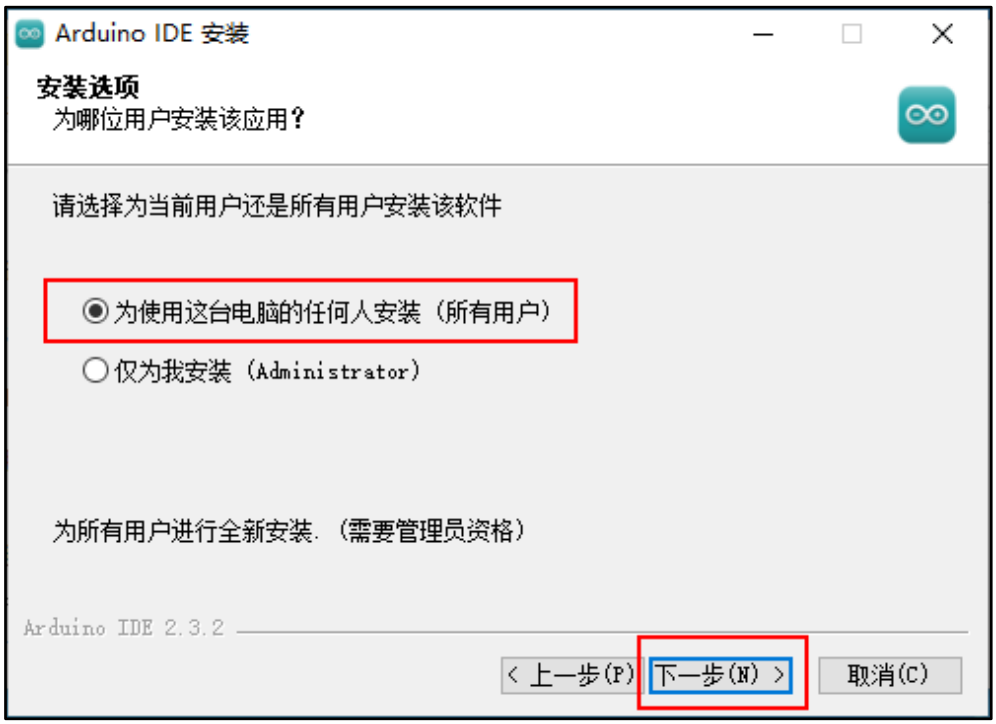

图 2.3 Arduino IDE2 软件安装用户选择

接下来点击“**下一步**”按钮，进入安装目录设置界面，点击“**浏览(B)...**”按钮选择安装目录或者直接输入安装目录，如下图所示：

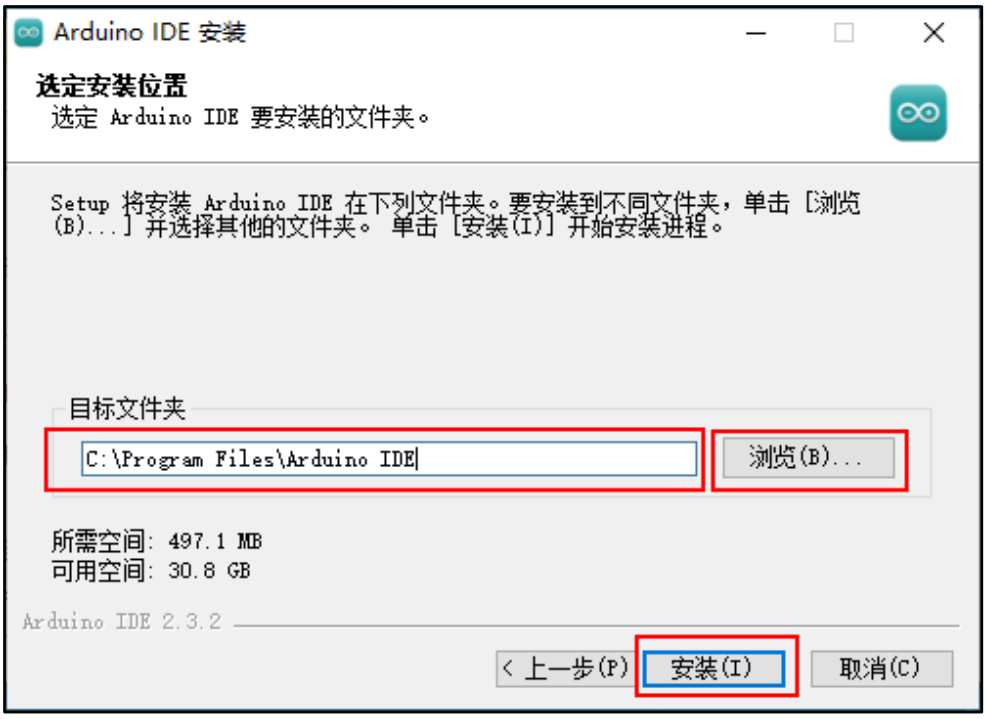

图 2.4 Arduino IDE2 软件安装目录选择

接下来点击“**安装**”按钮开始安装，可以看到安装进度条变化，如下图所示：

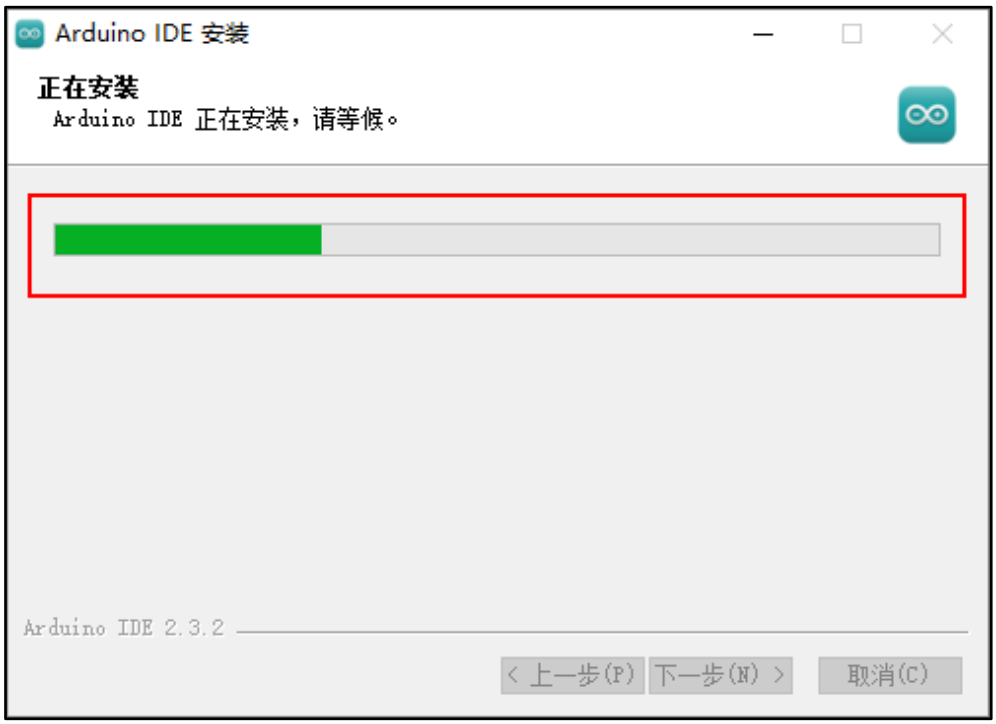

图 2.5 Arduino IDE2 软件安装过程

待进度条滚动完成，会弹出安装完成的界面，如果勾选“**运行 Arduino IDE(R)**”，那么点击“**完成(F)**”按钮关闭界面后，会自动打开 Arduino IDE2 软件，如果不勾选，则不会。如下图所示：

图 2.6 Arduino IDE2 软件安装完成

首次打开 Arduino IDE2 软件，会提示安装一些软件库和驱动，例如 Adafruit Industries LLC 端口(COM 和 LPT)、Arduino srl 的 Arduino USB Driver、Arduino SA 的 Arduino USB Driver、Genuino USB Driver，如下图所示：

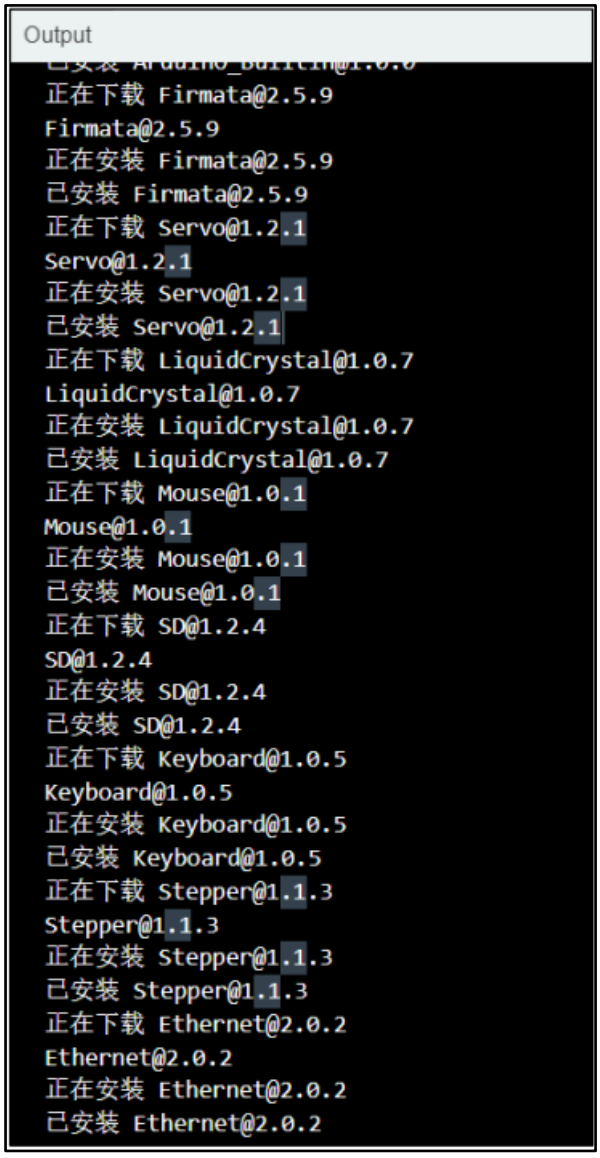

图 2.7 Arduino IDE2 软件库安装

图 2.8 Arduino IDE2 软件驱动安装

3. Arduino IDE2 软件介绍

Arduino IDE2 具有项目创建、程序代码编辑、调试、编译、上传、软件库管理、开发板管理等功能，界面如下图所示：

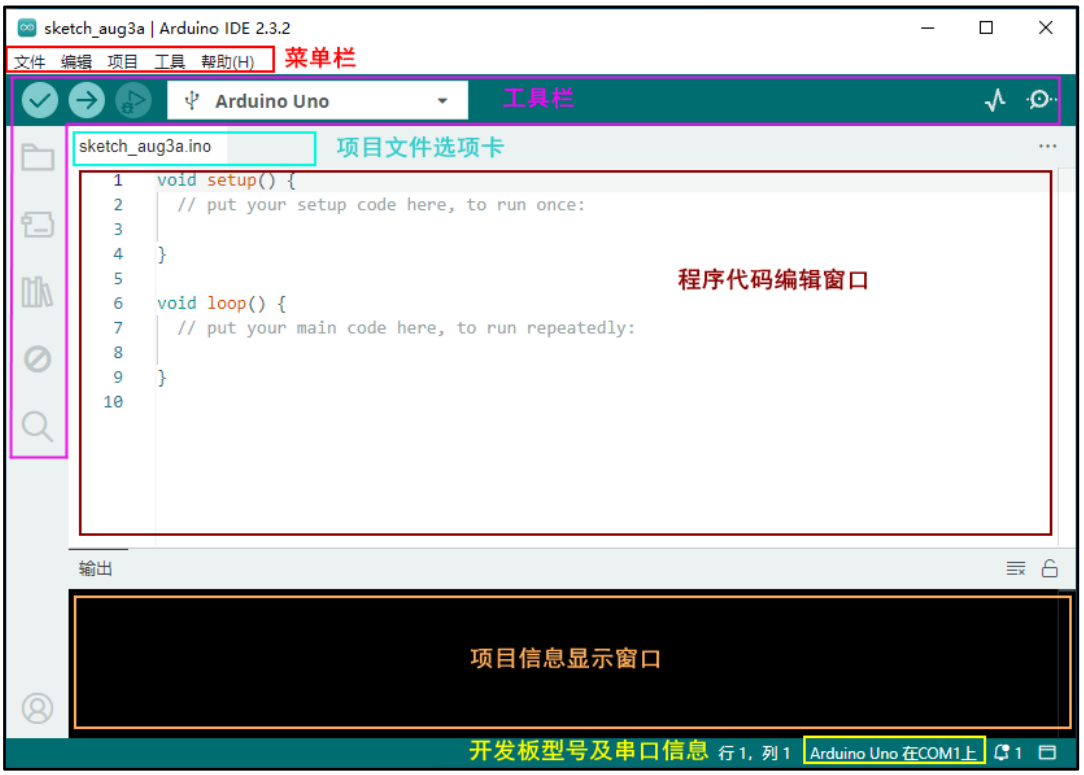

图 3.1 Arduino IDE2 界面

3.1. 菜单栏

3.1.1. 文件菜单

菜单栏文件菜单内容如下图所示：

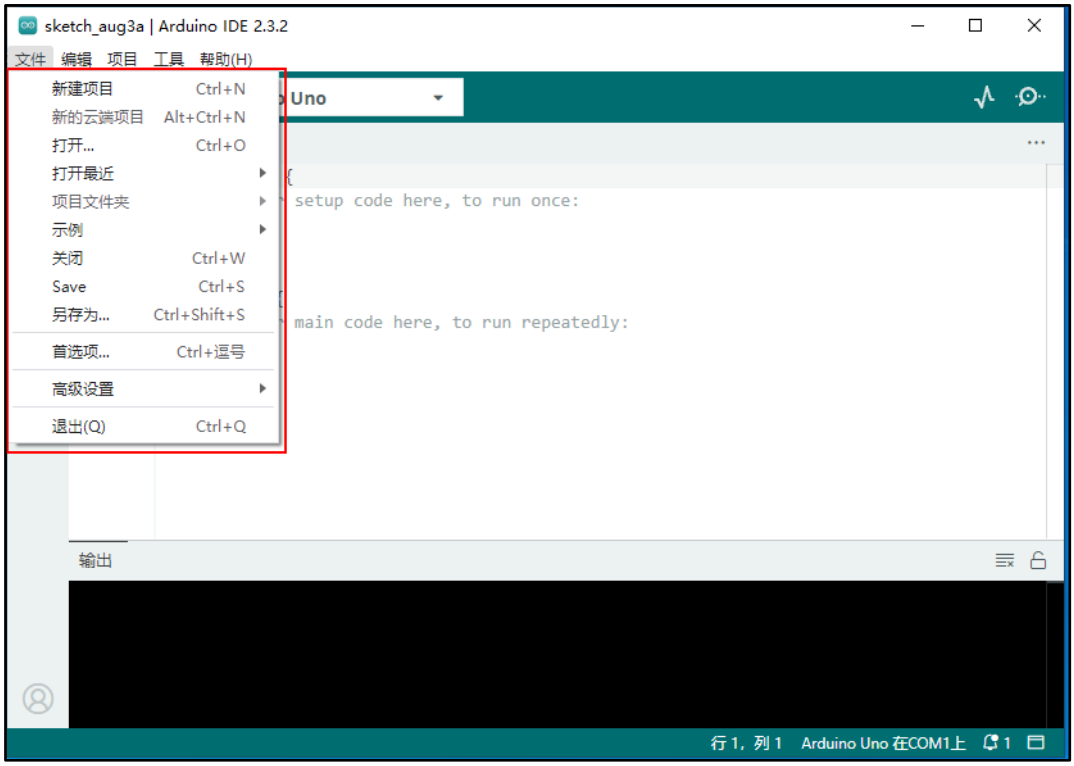

图 3.2 Arduino IDE2 文件菜单栏

里面基本上都是对项目进行新建、打开、保存、关闭等操作。其中“**示例**”选项可以选择打开第三方软件库和开发板核心库自带的示例程序。“**高级设置**”选项可以设置软件快捷键。这里重点介绍首选项菜单，点击“**首选项**”选项，如下图所示：

图 3.3 Arduino IDE2 首选项菜单

在首选项菜单里，可以进行如下方面设置：

A、项目文件夹地址，就是新建项目时，软件默认设置的项目保存位置，你可以自行修改该位置。该位置下 libraries 目录专门用于存放第三方软件库。如果勾选“**显示项目中的文件夹**”选项，则可以通过 IDE2 左边工具栏第一个按钮显示项目文件夹中项目文件。

B、编辑器字体大小和界面比例，可设置编辑器显示内容的字体尺寸以及界面大小。

C、颜色主题，可以设置编辑器的颜色显示风格，有：明亮、暗黑、明亮对比、暗黑对比四个选择。

D、编辑器语言，可设置多种语言，设置后，软件会自动重启。

E、显示详细输出，可勾选“**编译**”和“**上传**”两个选项。如果勾选“编译”，则在编译过程中信息显示窗口会输出编译信息；如果勾选“上传”，则在上传二进制文件到 MCU 时信息显示窗口会输出上传信息，为了获取详细的信息，建议两者都勾选。

F、编辑器警告，可以选择“无”、“默认”、“较多”、“全部”等，选择“无”，则不会显示任何信息，选择“更多”或者“全部”，则会显示更加全面的编译信息，不过会拖慢编译速度。

G、上传后验证代码，就是二进制文件上传后验证代码是否正确，可以勾选。

H、自动保存，建议勾选，防止电脑死机，代码丢失。

I、编辑快速建议，建议勾选，勾选后，在编辑代码，调用某个函数时，会提示函数的传参，这样可以快速且正确的编写代码。

J、其他开发板管理器网址，当你要添加的开发板在 IDE 的开发板管理器中搜索不到时（非 Arduino 官方开发板），就需要在这里添加开发板的地址。

设完成后，点击“**确定**”按钮保存。

3.1.2 .编辑菜单

菜单栏编辑菜单界面如下图所示：

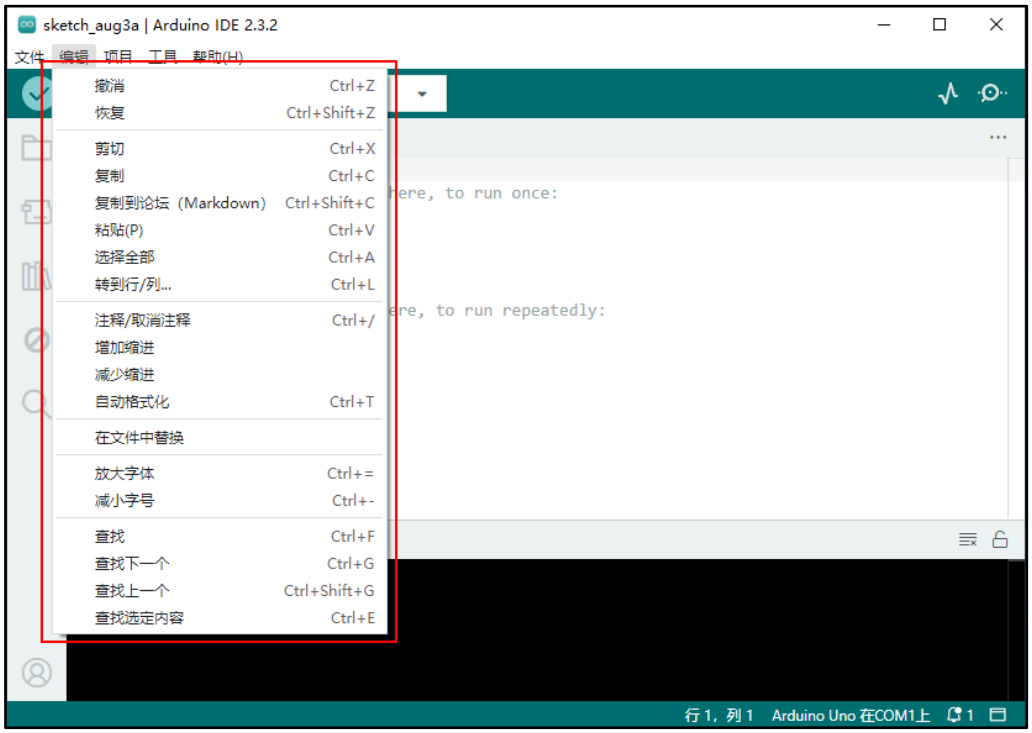

图 3.4 Arduino IDE2 编辑菜单栏

编辑菜单里主要是对项目文件内容进行复制、剪切、撤销、粘贴、查找、修改字号等编辑操作。

3.1.3. 项目菜单

菜单栏项目菜单界面如下图所示：

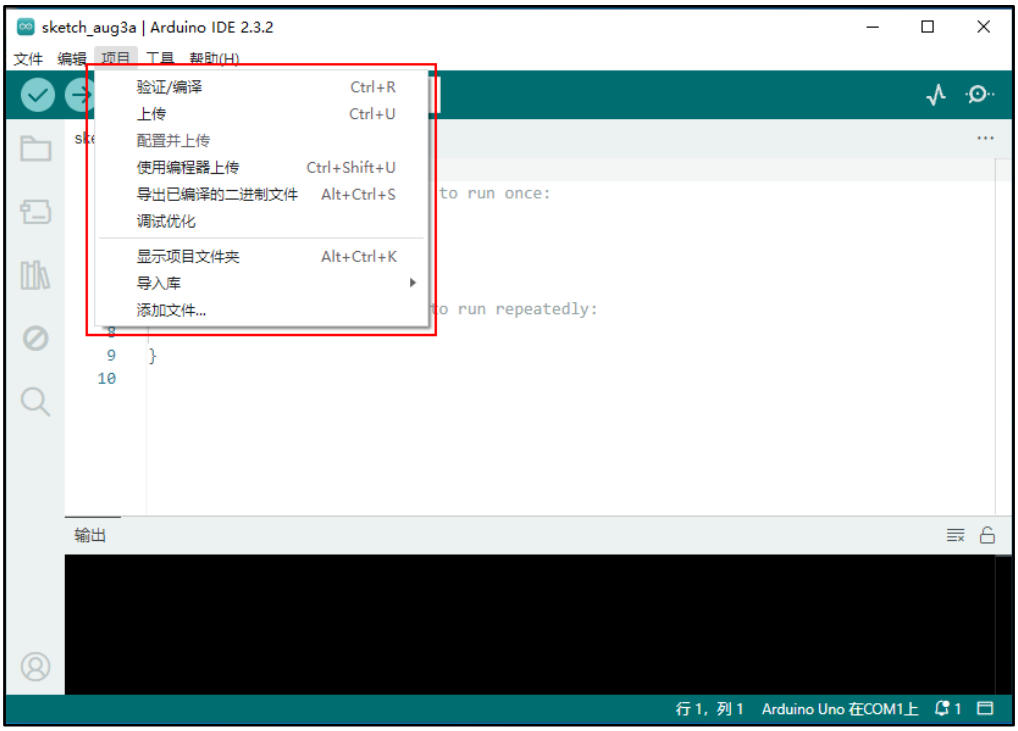

图 3.5 Arduino IDE2 项目菜单栏

项目菜单栏主要是对项目进行编译、调试、上传、导出、加载库文件等操作。

3.1.4. 工具菜单

菜单栏工具菜单界面如下图所示：

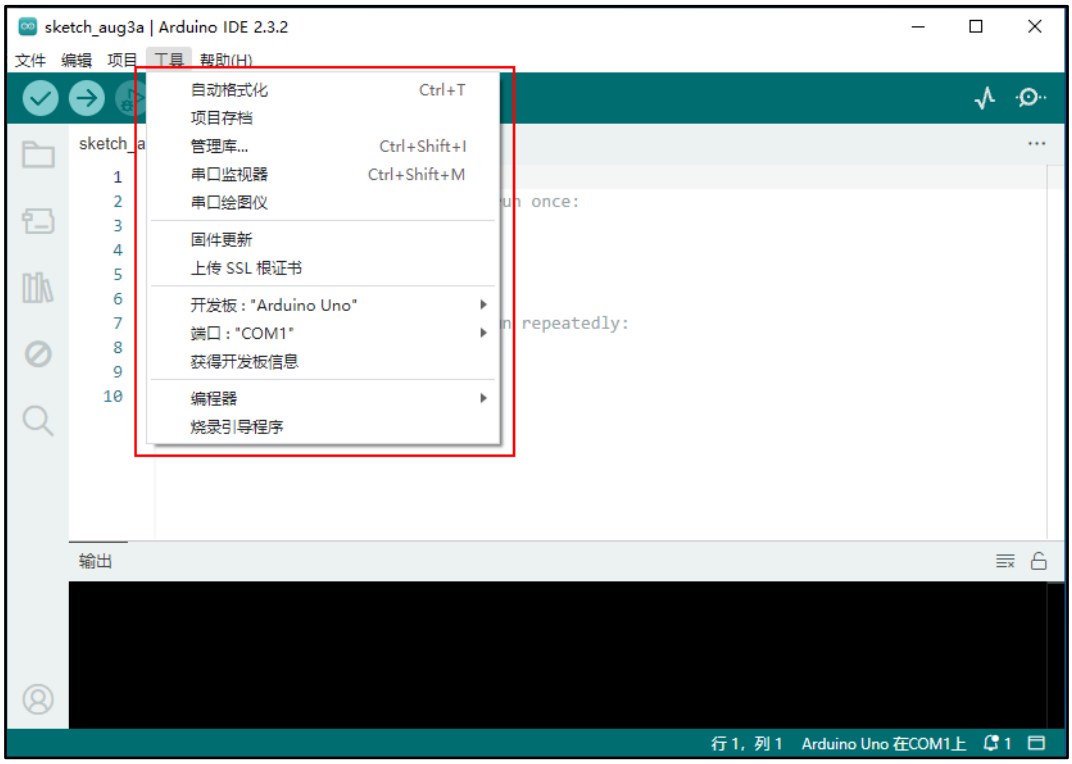

图 3.6 Arduino IDE2 工具菜单栏

在工具菜单里面可以进行如下方面设置：

A、**自动格式化**可以对项目代码格式进行自动排版，例如对齐方式等等。

B、**项目存档**可以对整个项目文件进行 ZIP 打包并保存。

C、**管理库器**可以对第三方软件库进行搜、下载并安装，点击进入，如下图所示。

在库管理器界面，可以根据软件库类型、话题对库进行过滤，也可以直接输出库名称对库进行搜索。搜索完成，选择库版本，点击“**安装**”按钮进行安装。最终软件库被安装在“C:\Users\Administrator\Documents\Arduino\libraries”目录（此为默认的目录，当然你也可以在**文件**->**首选项**界面进行修改，红色的字体为电脑实际的用户名）。当然，你也可以不通过库管理器安装软件库。你可以手动下载软件库（需要解压），然后拷贝到“C:\Users\Administrator\Documents\Arduino\libraries”目录。

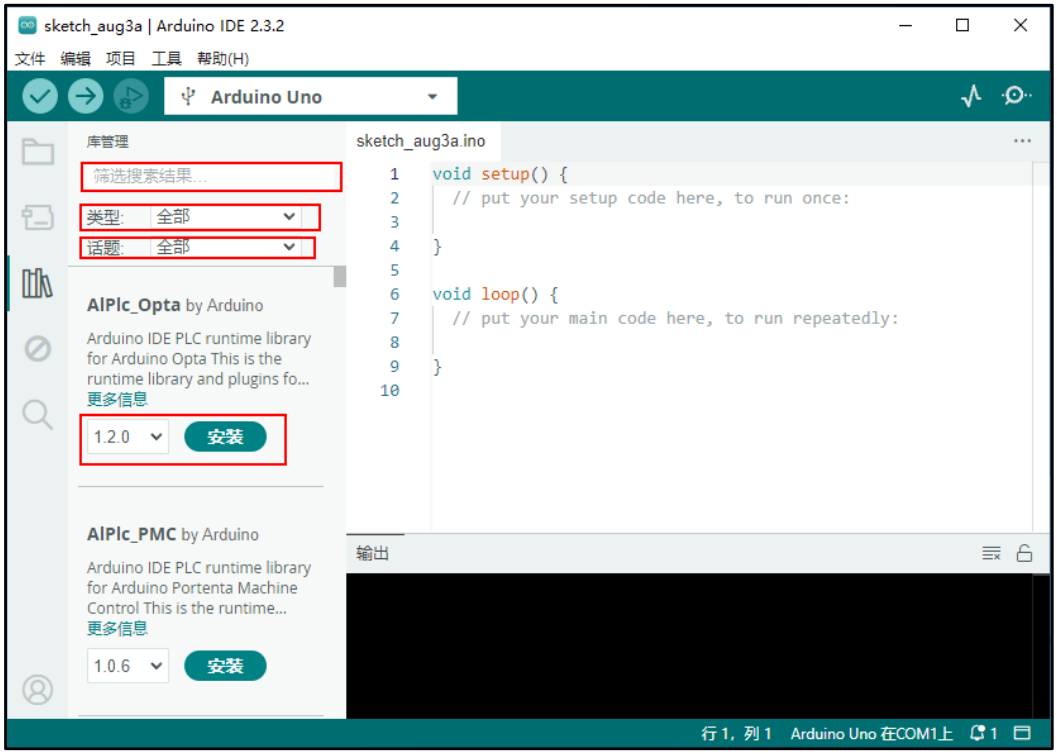

图 3.7 Arduino IDE2 库管理器

D、**串口监视器**和**串口绘图器**都是打开串口界面，设置串口波特率，显示串口输出信息，利用串口发送消息。(注意要连接开发板，正确识别串口后才能使用串口)。如下图所示：

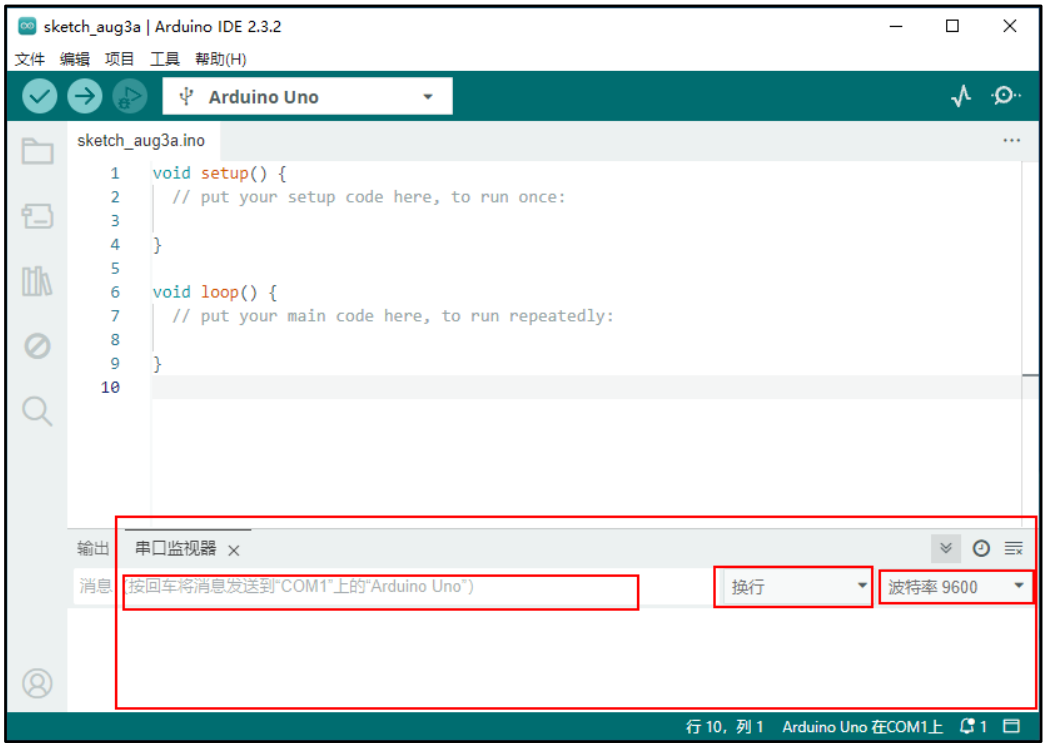

图 3.8 Arduino IDE2 串口输出窗口

E、**开发板**包含开发管理器和开发板选择两个部分，如下图所示：

图 3.9 Arduino IDE2 开发板菜单

开发板选项功能就是选择当前所使用的开发板，一旦选择成功后，在选项后面会显示。如果所使用的开发版不存在，则需对开发板的核心软件库进行安装，此情况下，要用到开发板管理器。开发板管理器界面如下图，可以对开发板进行搜索，然后选择版本，点击“**安装**”按钮对开发板核心软件库进行安装。

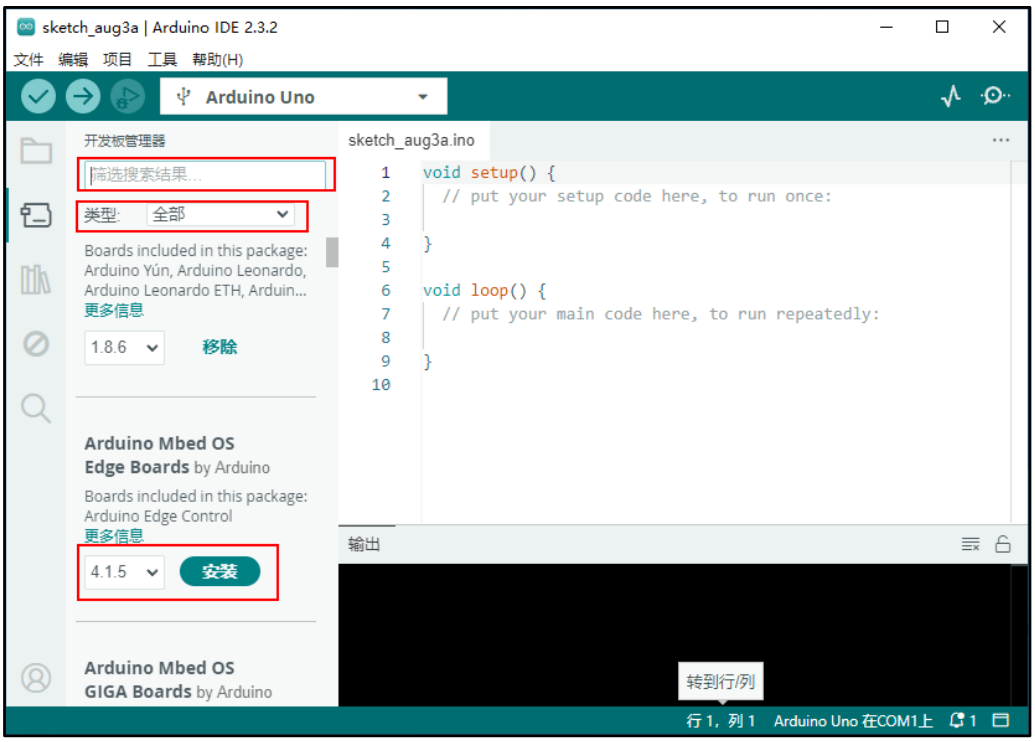

图 3.10 Arduino IDE2 开发板管理器

F、**端口**就是选择开发板连接串口，如下图所示，串口需要连接开发板后才会显示：

图 3.11 Arduino IDE2 端口选择

其他的一些选项基本用不到，保持默认设置即可。

3.1.5. 帮助菜单

帮助菜单界面如下图所示，主要是链接了官网的软件使用说明文档。

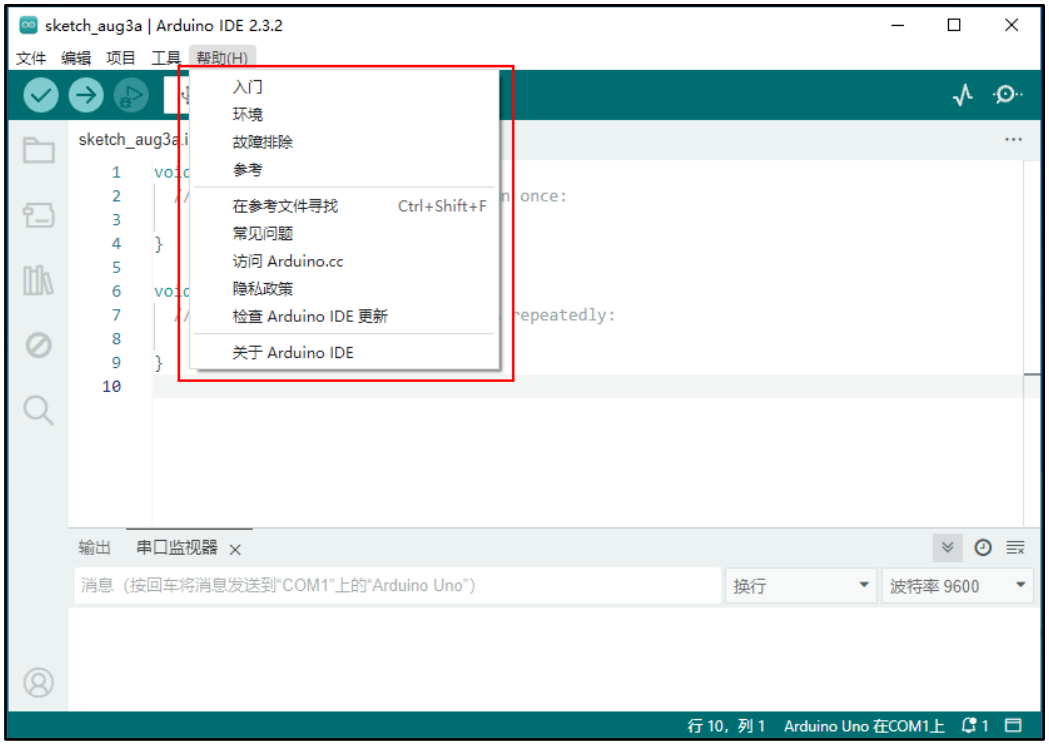

图 3.12 Arduino IDE2 帮助菜单

3.2. 工具栏

工具栏界面如下图所示：

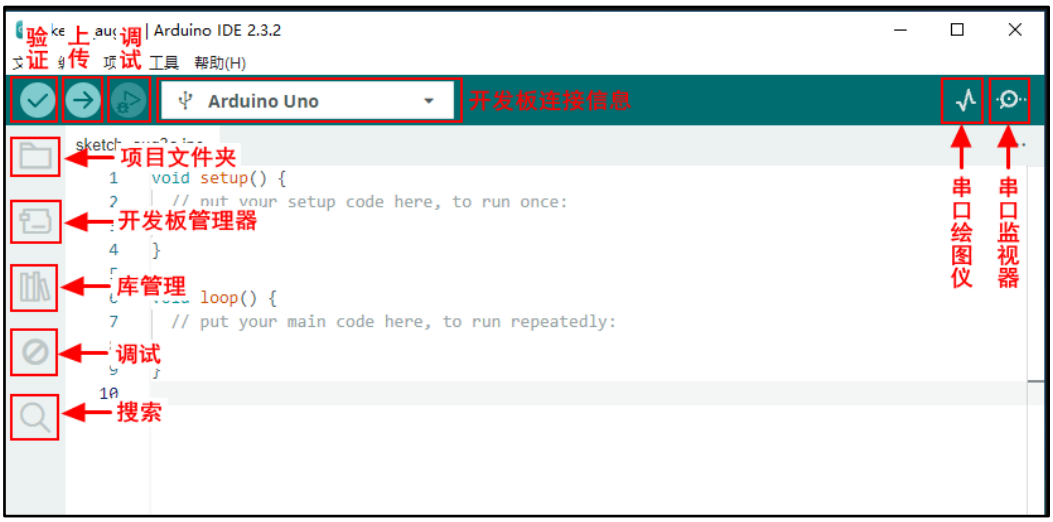

图 3.13 Arduino IDE2 工具栏

A、**验证**：编译并检查程序是否正确，如果正确则编译通过，生成二进制文件。

B、**上传**：编译程序生成二进制文件并上传到开发板 MCU 中。

C、**调试**：调试代码，可以看到 Debug 过程。Arduino IDE 不支持 ESP32 调试，所以该功能没用到。

D、**开发板连接信息**：下拉可以看到当前哪款开发板连在那个串口上。

E、**串口绘图仪**：可以更加直观的看到开发板接收到的数据及便利。

F、**串口监视器**：通过文字信息输出代码运行的情况，以及可以输入文字信息到开发板。

G、**项目文件夹**：可以显示项目文件夹里的文件情况（项目文件夹地址见首选项里面的设置）。

H、**开发板管理器**：选择已经安装的开发板和安装需要的开发板，和工具菜单栏里的开发板选项功能一致。

I、**库管理**：下载和安装第三方软件库，和工具菜单栏里的管理库选项功能一致。

J、**调试**：和上面 C 项的调试功能一致。

K、**搜索**：用于搜索函数等信息。

4. 安装 Arduino-ESP32 核心软件库

Arduino-ESP32 核心软件库是 Arduino 平台的一个插件，它为 Arduino 平台 ESP32 芯片的软件开发提供了底层支持。因为 Arduino IDE 默认不支持 ESP32，所以必须要安装 Arduino-ESP32 核心软件库。

Arduino-ESP32 核心软件库的安装方法有两种：Arduino IDE 开发板管理器在线安装和手动离线安装。

4.1. Arduino IDE 开发板管理器在线安装

A、打开 Arduino IDE 软件，点击**工具**->**开发板**->**开发板管理器**或者直接点击左边工具栏里的开发板管理器图标，在开发板管理器界面搜索栏里输入 ESP32，就会出现 ESP32 搜索结果，如下图所示。

注意：如果搜索不到 ESP32 核心软件库，则需要点击**文件**->**首选项**，在首选项界面的**其他开发板管理器网址**里输入

“https://espressif.github.io/arduino-esp32/package_esp32_index.json”，然后再按照上述步骤进行搜索。

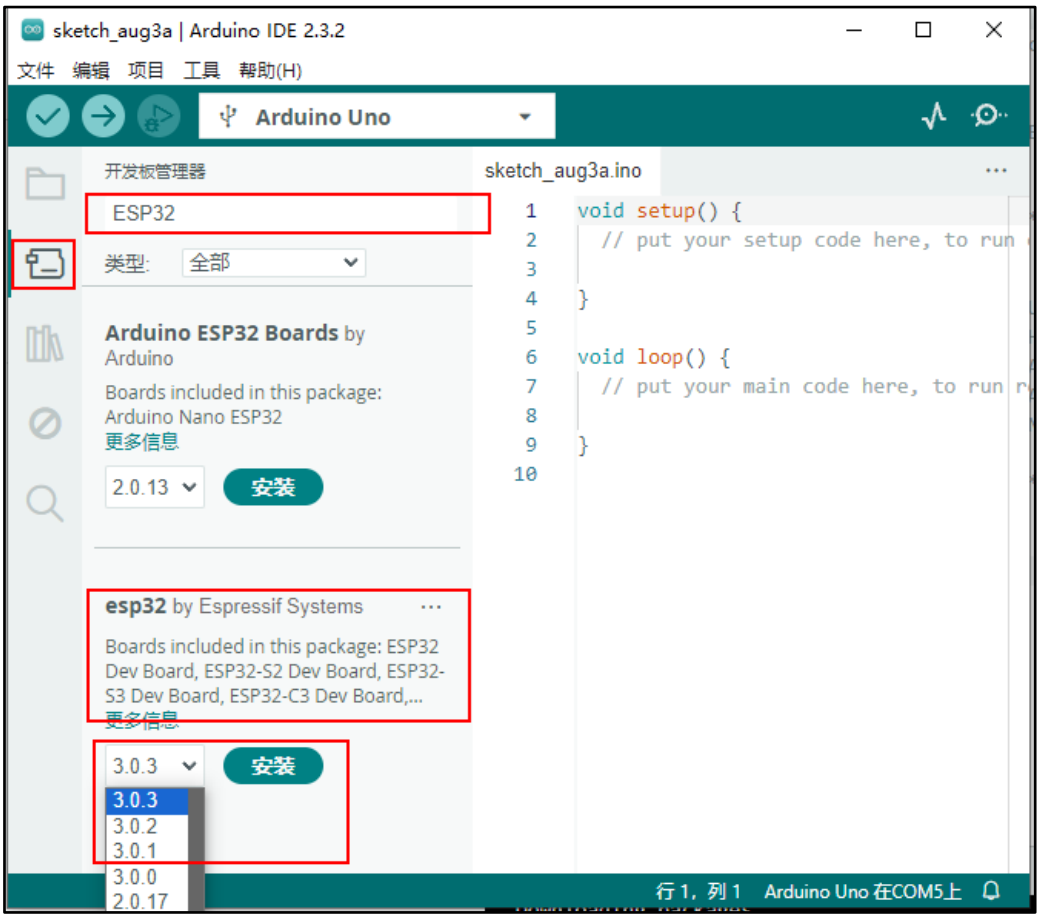

图 4.1 Arduino-ESP32 核心软件库搜索

B、选择“**esp32 by Espressif Systems**”，然后选择版本号，最后点击**安装**，如下图所示。

注意：3.0 版本是基于 ESP32 idf 5.1 开发的，2.0 版本是基于 ESP32 idf 4.4 开发的。两个版本在蓝牙、定时器、I2S 驱动、LEDC 驱动、定时器等软件上 API 有差异，所以 2.0 版本的示例中涉及到上述 API，如果使用 3.0 版本编译则会报错。需注意版本选择。

安装时间较长，安装过程可能出现下载失败的情况，需要多尝试几次安装。

下载的安装文件压缩包都保存在

“C:\Users\Administrator\AppData\Local\Arduino15\staging\packages”目录（红色部分为电脑的实际用户名，AppData 目录为隐藏目录，需要点击文件夹菜单栏的**工具->文件夹选项->查看->选择显示隐藏的文件、文件夹和驱动器**，然后点击**确定**保存）

图 4.2 Arduino-ESP32 核心软件库安装

C、安装完成后，关闭开发板管理器，点击**工具->开发板**，可以看到“**esp32**”选项，点击该选项，可以看到很多 ESP32 开发板，如下图所示：

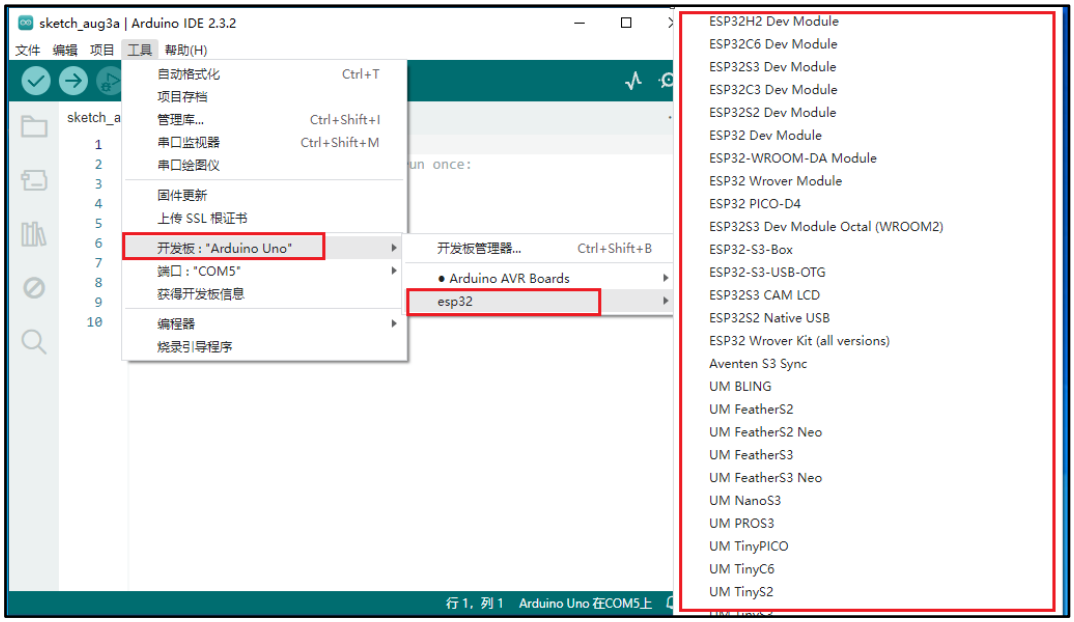

图 4.3 ESP32 开发板选择

4.2. 手动离线安装

开发板管理器在线安装是从 github 上下载安装文件压缩包，然后进行解压安装。在网络环境不好的情况下，访问 github 大概率会失败，从而导致在线安装失败。这里提供一种手动离线安装方法，步骤如下：

注意：首先得确保电脑上安装有 7-Zip 软件。

A、从以下网址下载安装文件压缩包：

https://pan.baidu.com/s/1F2tc8uJY_llqK-KFrWT2MQ?pwd=3lqe

其中 3.0.3 是 ESP32 核心软件库 3.0 版本，2.0.17 是 ESP32 核心软件库 2.0 版本，根据需求选择下载。

B、安装文件下载完成后，双击文件弹出解压缩窗口，在 Extract to 文本框里输入“C:\Users\Administrator\AppData\Local\Arduino15\packages”（红色部分是电脑的实际用户名），然后点击“Extract”按钮进行文件解压缩安装，如下图所示：

注意：如果解压的目标文件夹里已经存在 esp32 文件夹，需先将 esp32 文件夹删除，再进行文件解压缩安装。

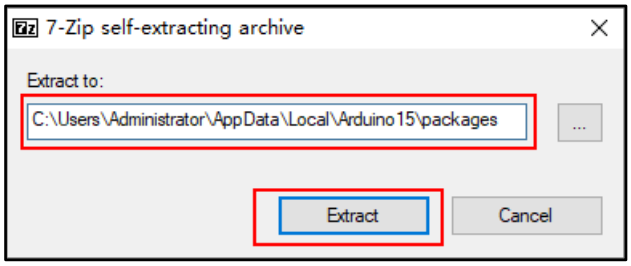

图 4.4 Arduino-ESP32 核心软件库离线安装

C、文件安装完成后，重新打开 Arduino IED 软件，点击**工具->开发板**，可以看到“esp32”选项，点击该选项，可以看到很多 ESP32 开发板，与在线安装步骤 C 一致。

5. 编译、下载并运行 ESP32 示例程序

5.1. 安装 USB 转串口 IC 驱动

使用串口下载程序时，需要在电脑上安装 USB 转串口 IC 驱动，否者电脑无法识别串口。不同的开发板使用不同的 USB 转串口 IC，需要安装相应的驱动。这里使用 CH340C USB 转串口驱动 IC，所以要安装 CH340 驱动，步骤如下（如果已经安装，可以忽略以下步骤）：

A、在资料包的“7-Tool_software”文件夹里找到“USB-SERIAL_CH340.zip”压缩包并解压。

B、进入解压后的文件夹，双击“CH341SER.EXE”可执行程序，弹出安装窗口，然后点击“**安装**”按钮继续安装，如下图所示：

图 5.1 CH340C 驱动安装

C、安装成功后，点击窗口确定按钮退出。电脑 USB 接上开发板上电，然后进入电脑设备管理器，可以看到端口下出现了被识别的 CH340 端口，如下图所示：

图 5.2 CH340 端口识别

5.2. 配置开发板

在 Arduino IDE 中新建或者打开存在的示例程序后，首先得进行开发板配置。步骤如下：

A、将开发板接电脑 USB 口上电，然后选择目标开发板型号，这里选择 ESP32，点击“**工具**”按钮，选择**开发板**->esp32->ESP32 Dev Module，

如下图所示：

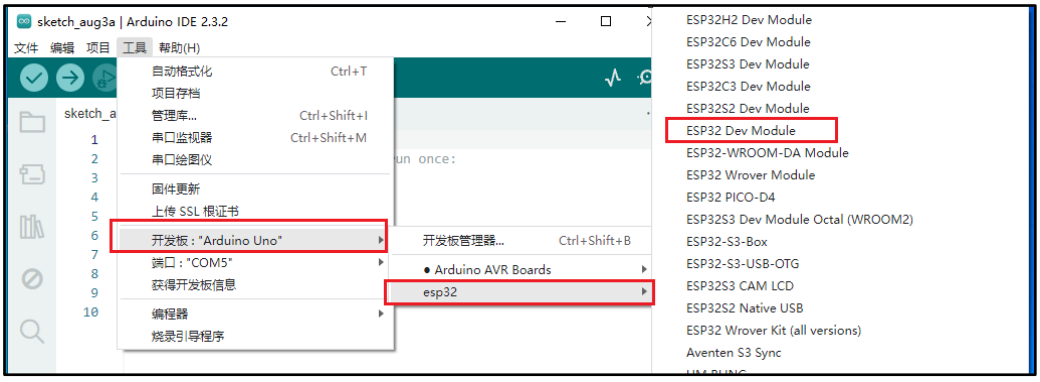

图 5.3 选择 ESP32 开发板型号

B、点击“**工具**”按钮，可以看到 ESP32 开发板的默认配置，如下图所示：

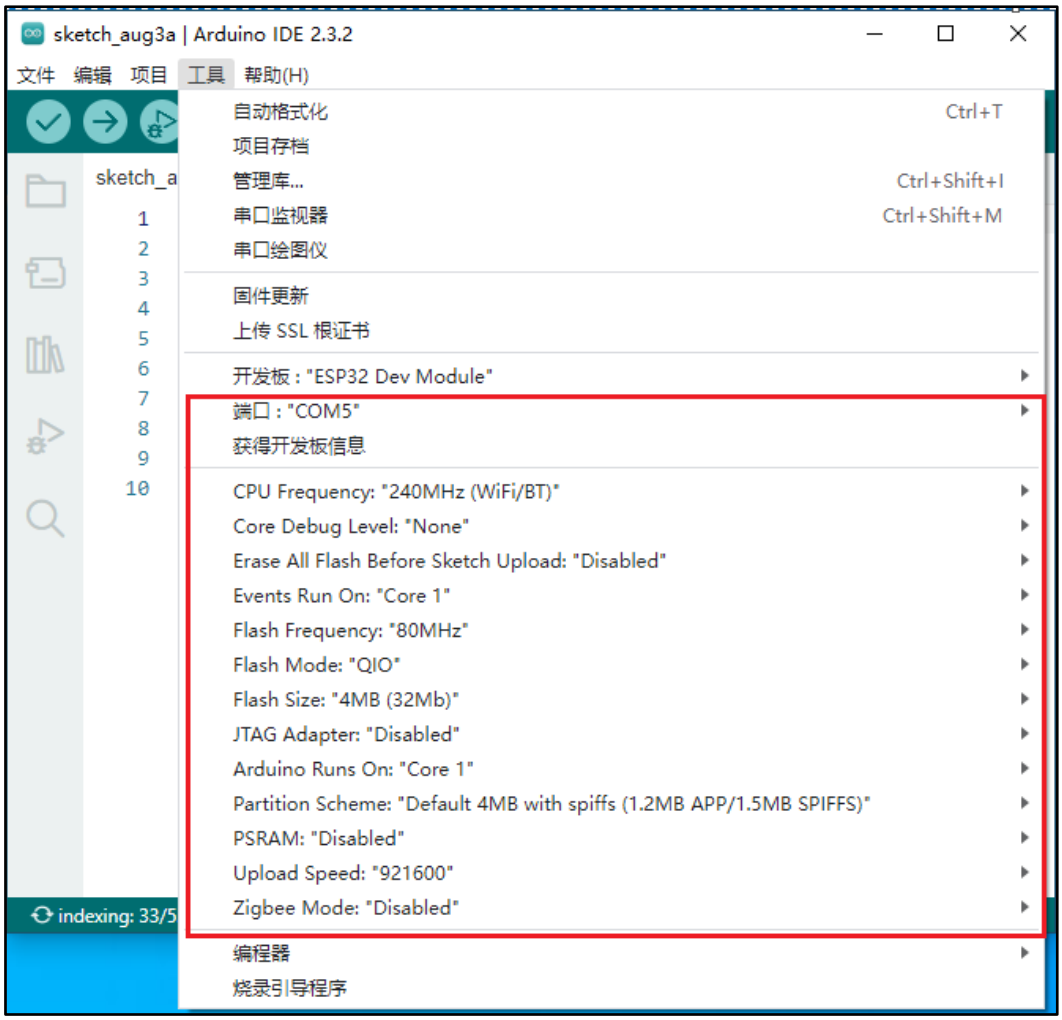

图 5.4 ESP32 开发板配置

这里介绍一下各配置参数：

端口：选择 ESP32 开发板所连接的串口号，一般情况下会自动识别出来。

CPU Frequency：CPU 时钟频率，可选参数有：240MHz(WiFi/BT)、160MHz(WiFi/BT)、80MHz(WiFi/BT)、40MHz、26MHz、20MHz、13MHz、10MHz。一般来说，频率越高，功耗越大，可以根据需求选择。这里不考虑功耗，直接选择最大 240MHz 频率，发挥最佳性能。需要注意的地方，240MHz、160MHz、80MHz 这三个频率可以保证 WiFi 和 BT 的正常运行，其他频率不能保证 WiFi 和 BT 正常运行，只能保证 CPU 运行基本功能。

Core Debug Level：Arduino 内核调试日志等级，通过串口输出，可选参数有：None、Error、Warn、Info、Debug、Verbose。其中，

None：不输出任何调试日志；

Error：仅输出错误级别的调试日志；

Warn：仅输出警告及以上级别的调试日志；

Info：仅输出信息及以上级别的调试日志；

Debug：仅输出调试及以上级别的调试日志；

Verbose：输出内核调试中所有级别的调试日志；

一般情况下，无需关注内核调试日志，除非开发一些和内核相关的功能。所以这里选择无就可以了。

Erase ALL Flash Before Sketch Upload：配置上传代码时是否将整个 Flash 全面擦除，可选参数有：Disabled、Enabled。选择“Disabled”就是不需要全面擦除，选择“Enabled”就是需要全面擦除。选择全面擦除，那么上传代码时，速率会变慢，此外经常全面擦除 Flash，会影响 Flash 的使用寿命，所以这里选择“Disabled”。

Event Runs On：配置 Arduino 中断事件所运行的 ESP32 内核，可选参数有：Core0、Core1。可以根据情况自行选择，这里默认选择 Core1。

这里配置的内核可以和 **Arduino Runs On** 中配置的一样，也可以不一样，当配置成一样时，可以降低 ESP32 功耗；当配置成不一样时，可以提高程序运行效率。

Flash Frequency：ESP32 上挂载的 Flash SPI 总线时钟频率，可选参数有：80MHz、40MHz。为了提高 Flash 的读写速度，一般选择 80MHz 高频率。

Flash Mode：ESP32 上挂载的 Flash 通信模式，可选参数有：QIO、DIO。其中 QIO 为使用 4 根 SPI 数据线用于 Flash 写入和读取。DIO 为使用 2 根 SPI 数据线用于 Flash 写入和读取。根据 Flash 的实际连接方式进行选择。这里使用使用 4 根 SPI 数据线用于 Flash 写入和读取，所以选择 QIO。

Flash Size：ESP32 上挂载的 Flash 容量，可选参数有：4MB(32Mb)、8MB(64Mb)、2MB(16Mb)、16MB(128Mb)。根据 Flash 的实际容量进行选择，这里使用的 Flash 为 4MB，所以选择 4MB(32Mb)。

JTAG Adapter：配置 JTAG Adapter，可选参数有：Disabled、FTDI Adapter、ESP USB Bridge。使用 JTAG 调试代码更加方便，但是 Arduino IDE 不支持 ESP32 调试。所以这里选择“Disabled”。

Arduino Runs On：配置 Arduino Core 任务代码所运行的 ESP32 内核，可选参数有：Core0、Core1。ESP32 有两个内核，分别为 Core0 和 Core1，每个内核可以运行不同的代码任务。可以根据情况自行选择，这里默认选择 Core1。

Partition Scheme：ESP32 上挂载的 Flash 空间分区方法。为了更合理的利用 Flash 的空间，Arduino IDE 设计了十几种分区方法，这里就不一一介绍了，有兴趣，可以自行去学习。这里使用的 Flash 为 4MB，一般情况选择“Default 4MB with spiffs(1.2MB APP/1.5MB SPIFFS)”,如果项目文件多，编译生成的二进制文件比较大，可以选择“Huge APP(3MB No OTA/1MB SPIFFS)”。

PSRAM：配置 ESP32 外挂的 PSRAM，可选参数有：Disabled、Enabled。有些 ESP32 除了内置的 SRAM 外，还外挂了 PSRAM 进行内存扩容，此种情况需选择“Enabled”。有些 ESP32 只有内部 SRAM，此种情况需选择“Disabled”。这里使用的 ESP32 没有挂载 PSRAM，所以选择“Disabled”。

Upload Speed：代码上传的速率，可选参数有：51200、230400、256000、115200、921600。根据开发板上 USB 转串口所支持的最大速率选择，例如这里所用的 CH340C 支持的最大速率为 2Mbps，所以选择最大值 921600。

5.3.编译、下载并运行程序

这里使用 Arduino-ESP32 核心软件库自带的示例作为演示。也可以自己新建一个项目来编译、下载并运行，还可以打开已经完成的项目来操作。

A、打开 Arduino IDE，点击**“文件”**按钮，选择**示例->ESP32->ChipID->GetChipID**，如下图所示：

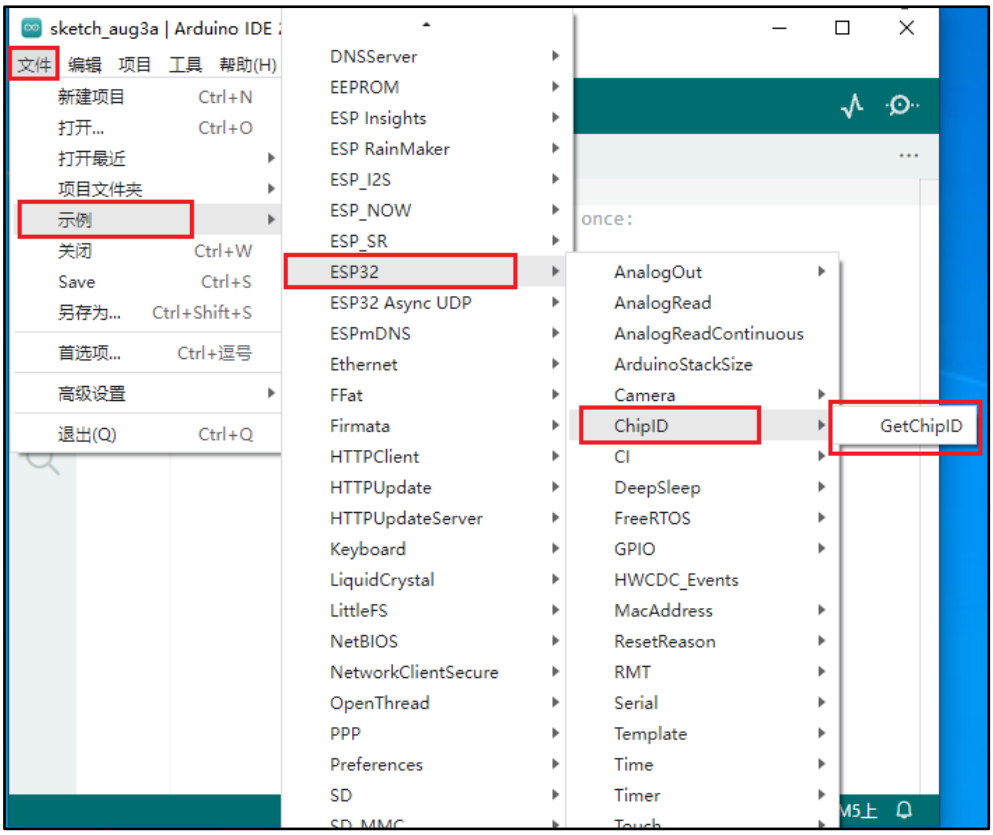

图 5.5 ESP32 示例程序

B、点击“**上传**”按钮，可以看到“**正在编译项目...**”提示，如下图所示：

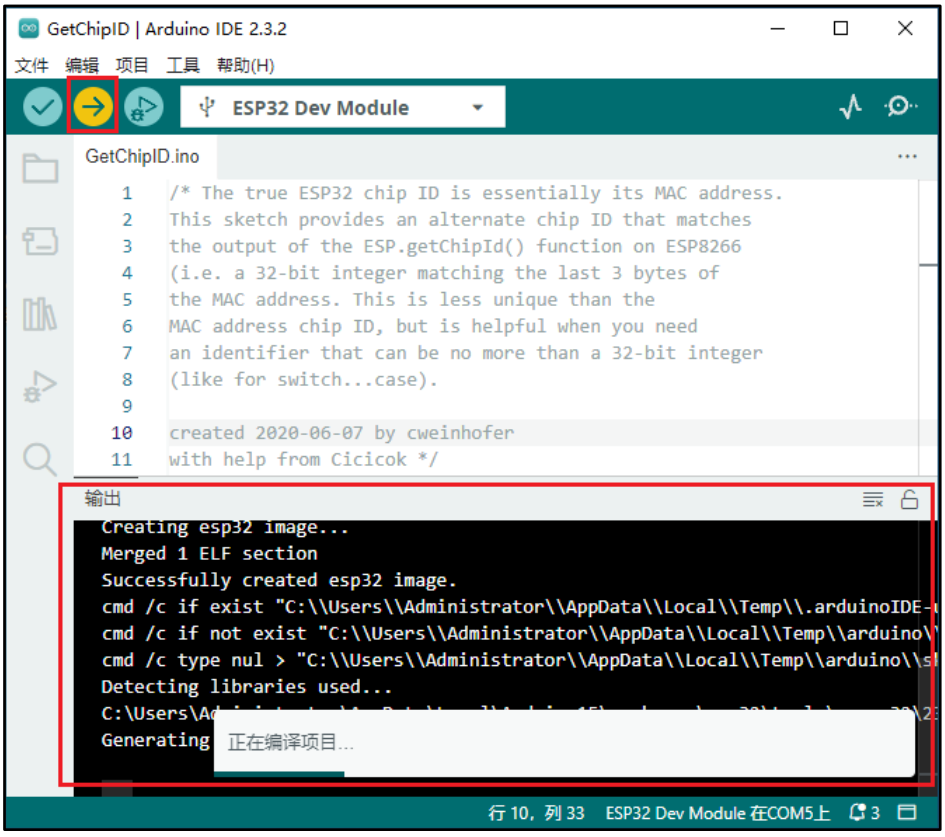

图 5.6 ESP32 项目编译

C、编译成功后，会显示“**正在上传...**”提示，同时信息输出窗口会输出编译成功信息，如下图所示：

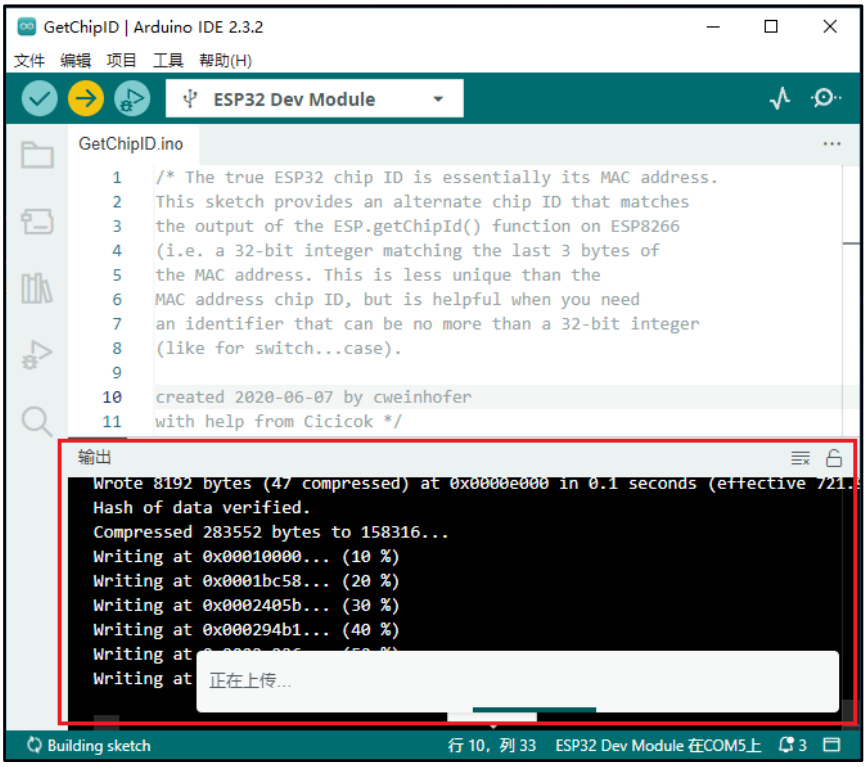

图 5.7 ESP32 项目编译成功

D、上传成功后，信息输出窗口会输出上传的信息和程序运行的提示，如下图所示：

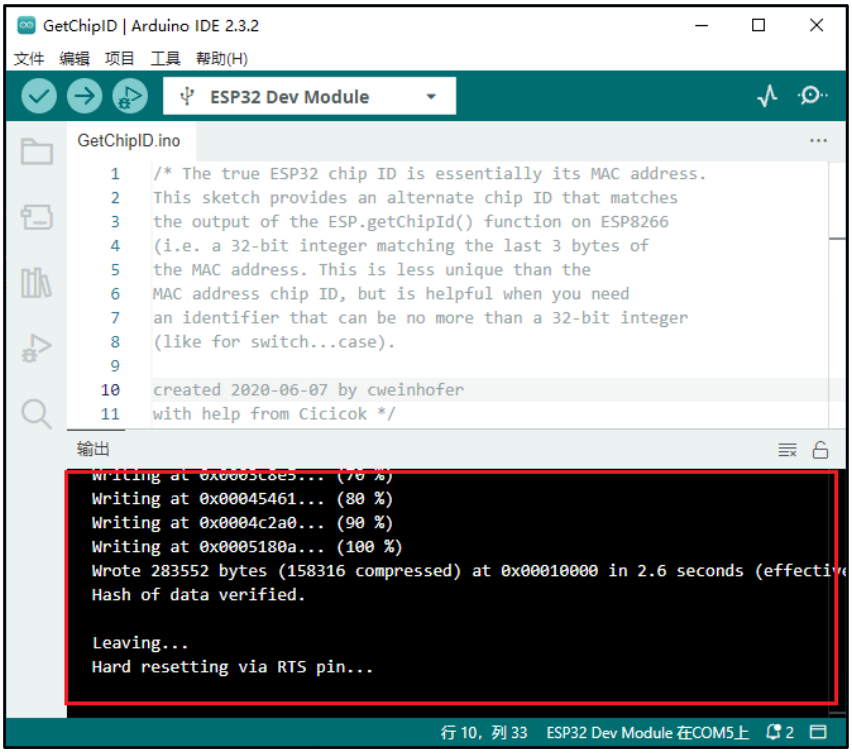

图 5.8 ESP32 项目上传成功并运行

E、点击菜单栏**工具**->**串口监视器**，弹出串口界面，设置波特率为 115200，就可以看到串口终端有信息输出，此时程序运行成功，如下图所示：

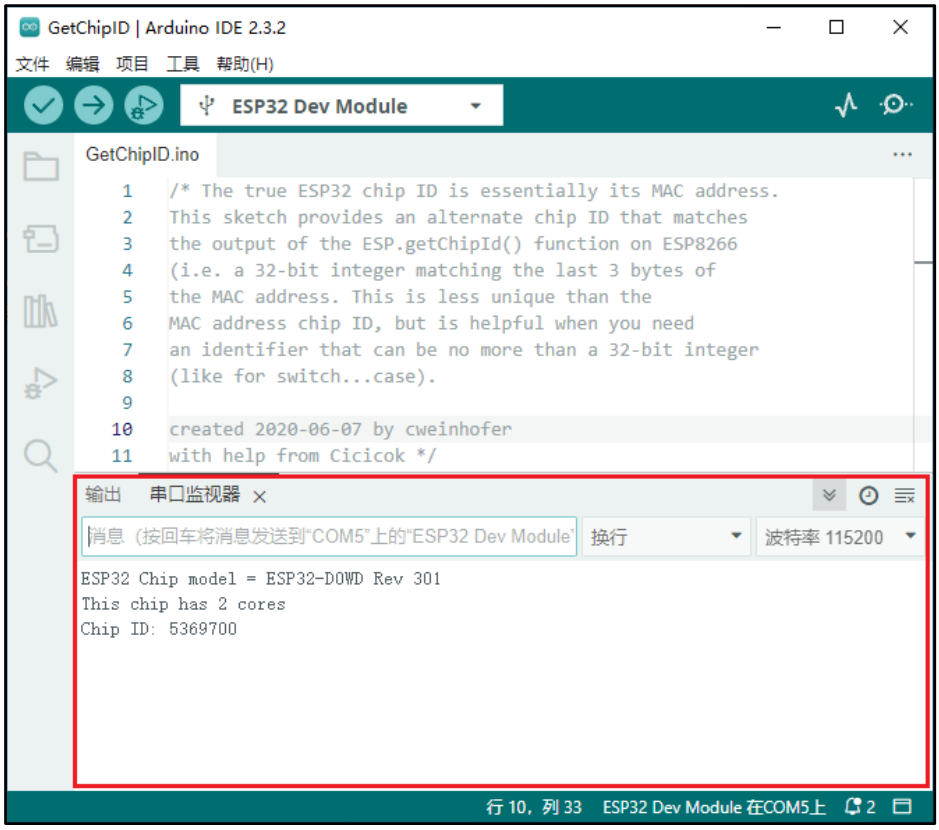

图 5.9 ESP32 程序串口输出