

1. 测试平台介绍

开发板：ESP32-WROOM-32E devKit

MCU：ESP32-32E模组

主频：240MHz

2. 引脚连接说明

本模块可以直插到ESP32-32E开发板上，如下图所示：

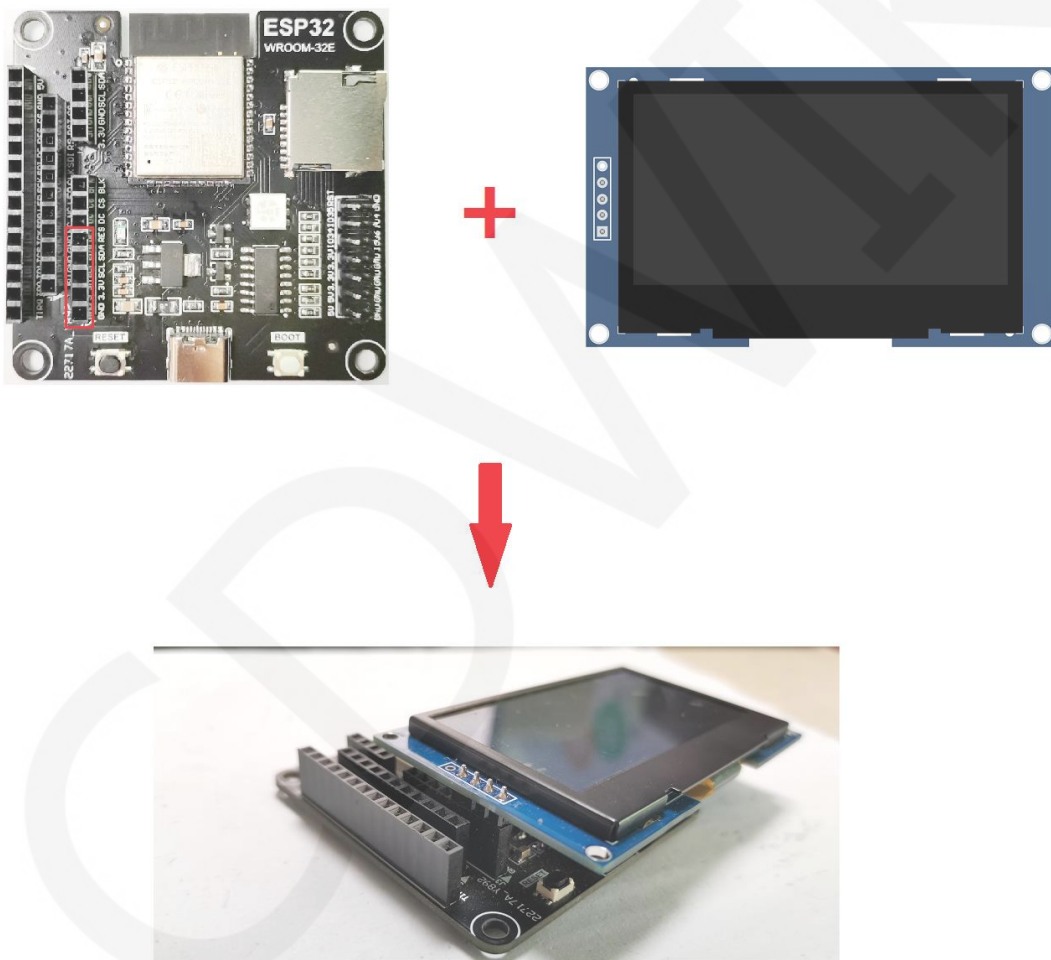


图1 模块直插ESP32-32E开发板

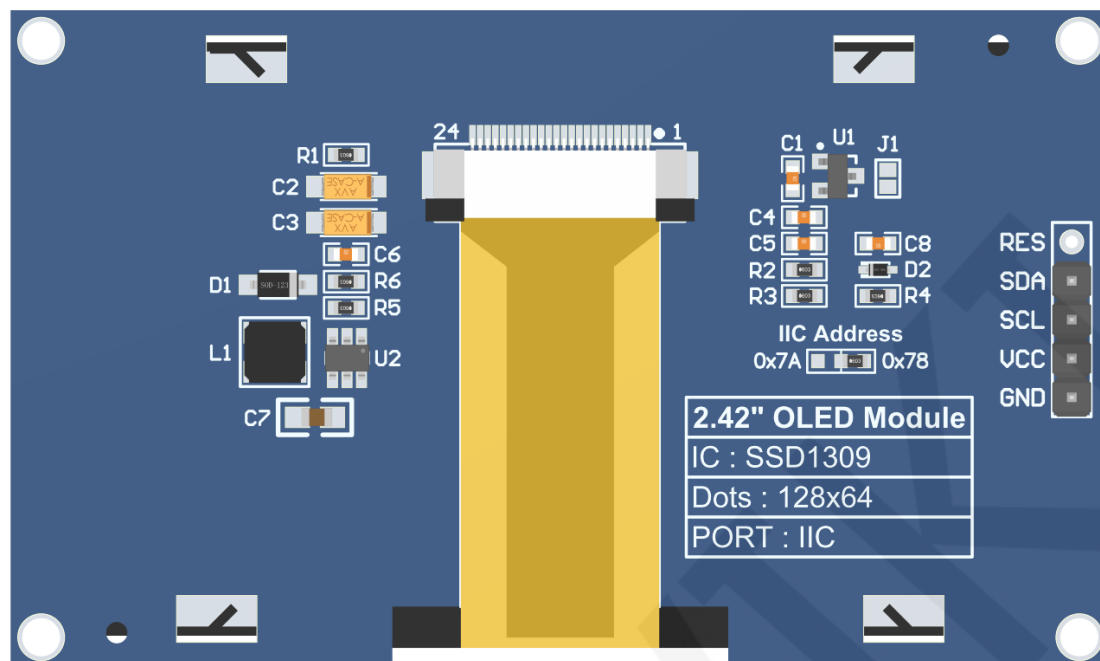


图2 模块背面引脚

注意：

- A、IIC Address电阻用来选择IIC从设备地址，如果焊接在0x78这边，则选择0x78从设备地址，如果焊接在在0x7A这边，则选择0x7A从设备地址；
- B、RES引脚排针默认不焊接，如果需要在程序里控制复位功能，则需焊接；

ESP32-32E IIC 测试程序引脚直插说明			
序号	模块引脚	对应ESP32-32E开发板接线引脚	备注
1	GND	GND	OLED屏电源地
2	VCC	5V/3.3V	OLED屏电源正
3	SCL	IO14	IIC总线时钟信号
4	SDA	IO13	IIC总线数据信号
5	RES	不焊接	该引脚排针默认不焊接，如果需要在程序里控制复位功能，则需焊接

3. 例程功能说明

本套示例程序使用ESP32硬件IIC总线，包含硬件IIC和软件IIC测试程序，其位于

Demo_ESP32目录下，如下图所示：



✧ 示例程序内容说明

本套示例程序里包含如下内容：

- A、Example01-graph_test为图形显示测试；
- B、Example02-string_test为字符显示测试；
- C、Example03-show_BMP为BMP位图显示测试；

✧ 示例程序IIC从设备地址修改说明

硬件上修改了IIC从设备地址，软件上也要相应的做修改。首先打开任意一个IIC示例程序，然后找到**setup**函数，如果使用0x7A从设备地址，则无需将**u8g2.setI2CAddress(0x7A)**这行代码注释（使其生效），如果使用0x78从设备地址，则需将**u8g2.setI2CAddress(0x7A)**这行代码注释起来（使其不生效），如下图所示：

```
void setup(void) {  
    /*When using 0x7A slave device address, please use the following definition*/  
    //u8g2.setI2CAddress(0x7A);  
    Wire.begin(/*SDA*/ SDA, /*SCL*/ SCL);  
    u8g2.begin();  
}
```

4. 例程使用说明

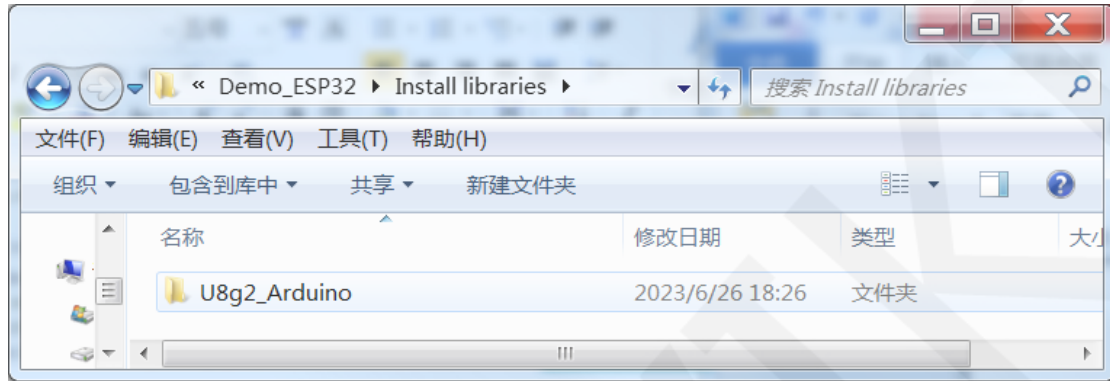
✧ 搭建开发环境

搭建开发环境的具体方法请参考本目录下

“[Arduino_development_environment_construction_for_ESP32_CN](#)”文档。

✧ 安装软件库

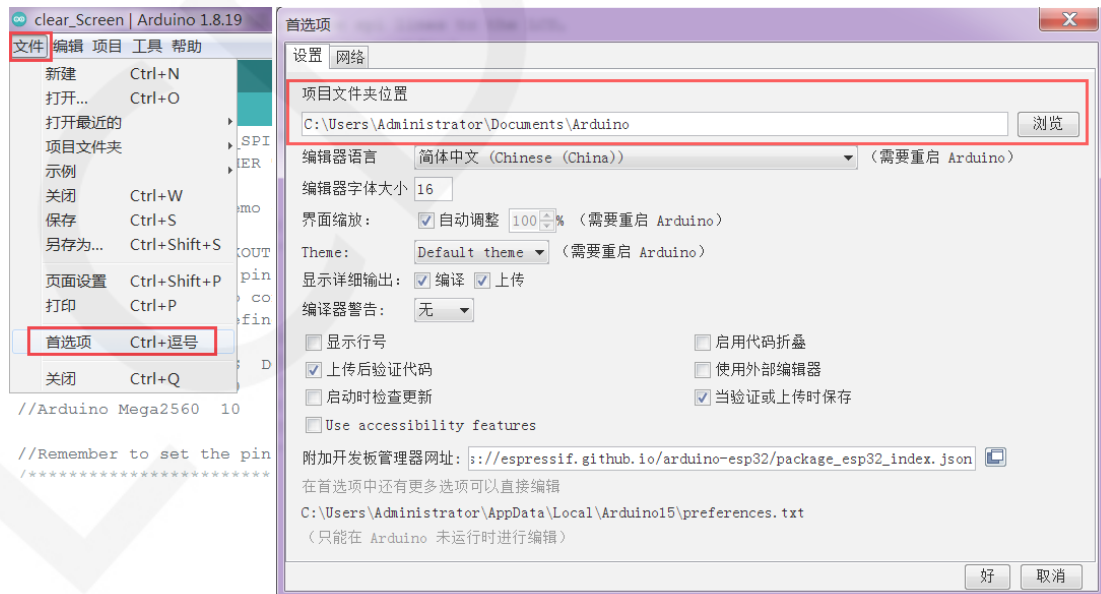
开发环境搭建好之后，需要将示例程序使用的软件库拷贝到工程库目录下，以便示例程序调用。软件库位于**Demo_ESP32\Install libraries**目录下，如下图所示：



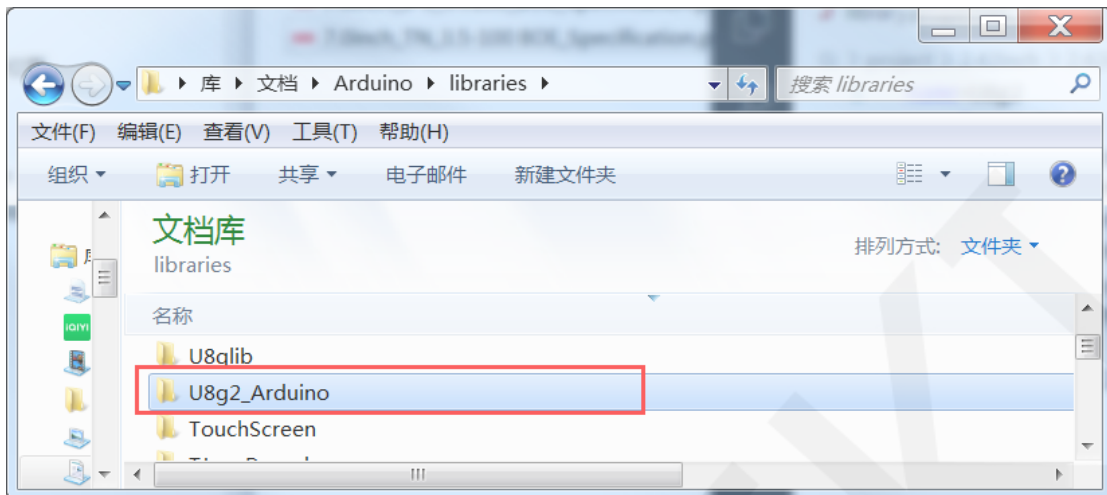
也可以从Github上下载最新的软件库，将其解压（为了便于区分，可对解压后的文件夹进行重命名，如**Install libraries**目录下所示），然后拷贝到工程库目录下，下载地址如下：

https://github.com/olikraus/U8g2_Arduino

工程库目录默认的路径为**C:\Users\Administrator\Documents\Arduino\libraries**。也可以更改工程库目录：打开Arduino IDE软件，点击**文件->首选项**，在弹出的界面里重新设置**项目文件夹位置**，如下图所示：



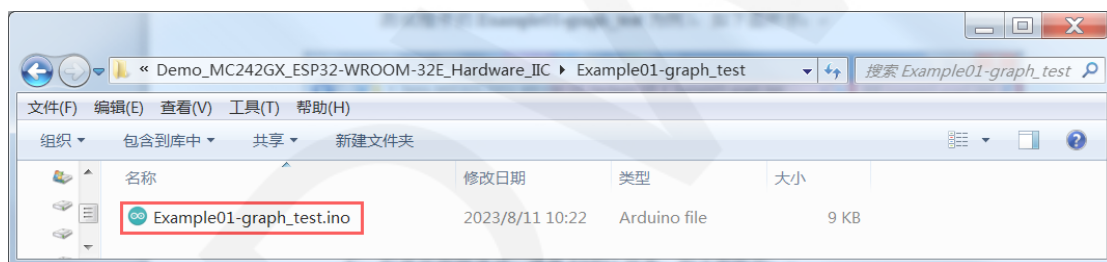
将软件库拷贝到工程库目录，如下图所示：



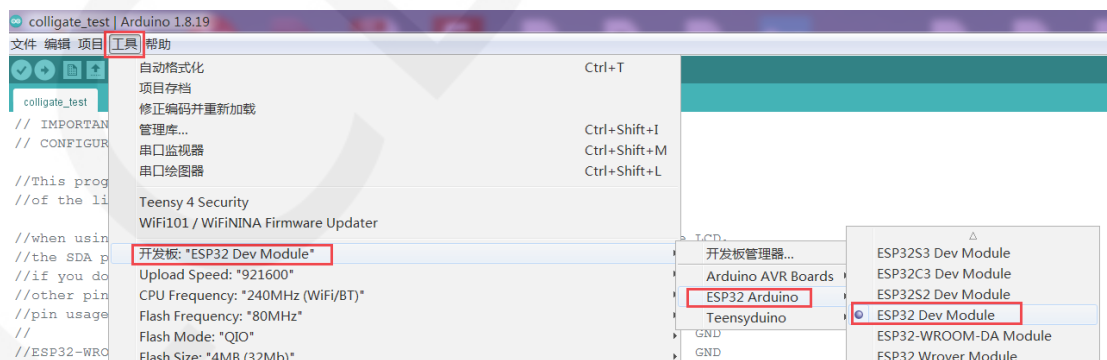
✧ 编译并运行程序

库安装完成之后，就可以进行示例程序编译及运行了，步骤如下：

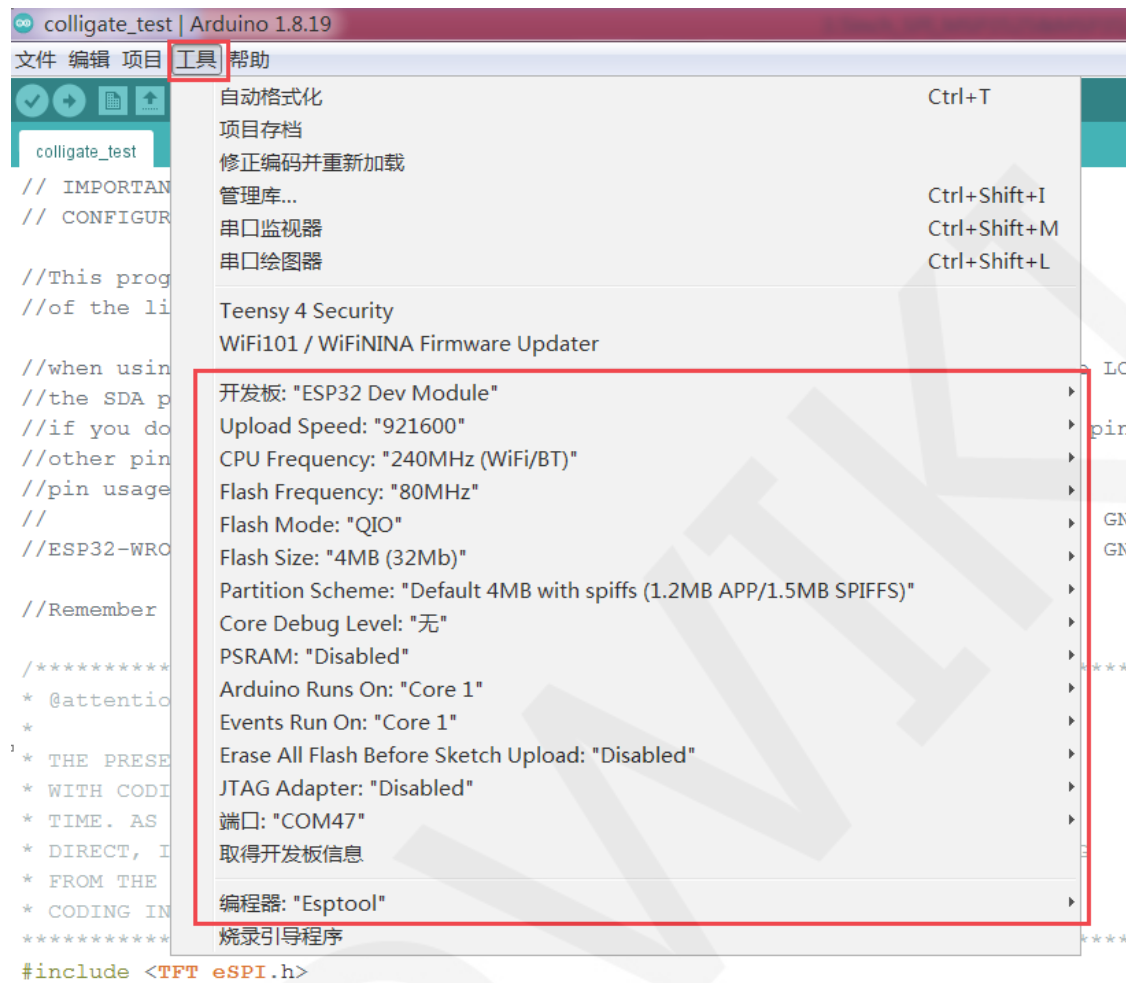
- A、将显示模块直插到 ESP32 开发板，将开发板连接 PC 机上电；
- B、打开 **Demo_ESP32** 目录下任意一个示例程序，如下图所示（这里以这里以硬件 IIC 测试程序的 Example01-graph_test 为例），如下图所示：



- C、打开示例程序后，选择 ESP32 设备，如下图所示：



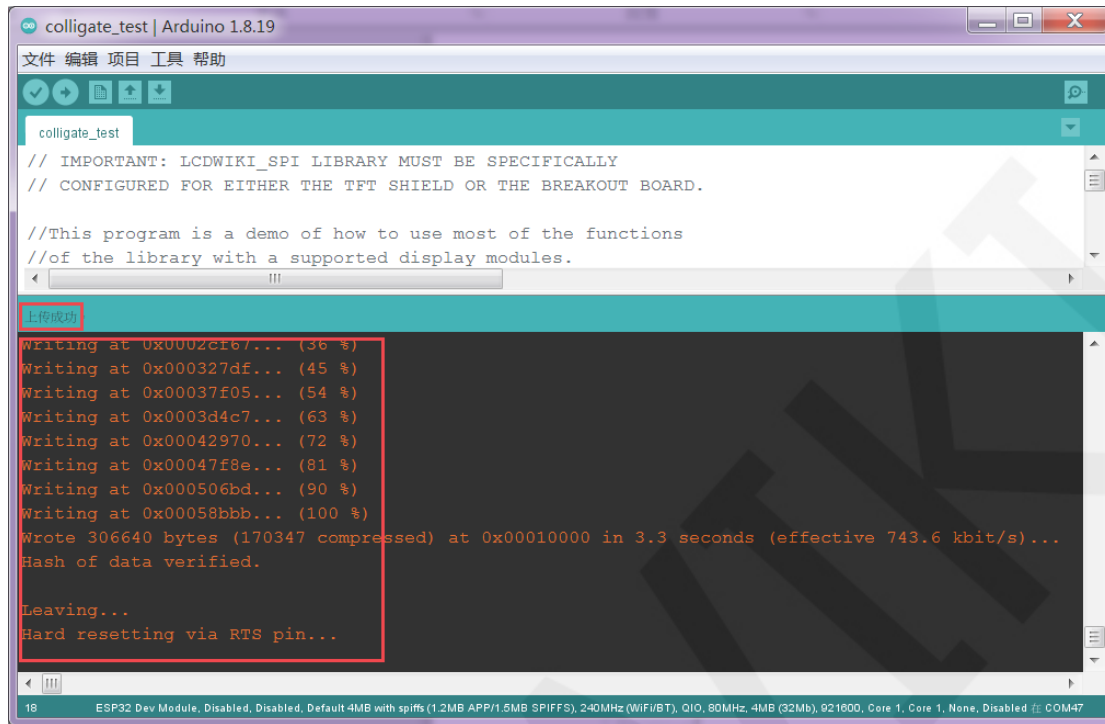
D、进行 ESP32 Flash、PSRAM、端口等配置，如下图所示：



E、点击上传按钮进行程序编译和下载，如下图所示：



F、出现如下提示则说明程序编译完成并下载成功，且已经运行：



The screenshot shows the Arduino IDE interface for the 'colligate_test' project. The top menu bar includes '文件', '编辑', '项目', '工具', and '帮助'. The main editor area contains the following code:

```
// IMPORTANT: LCDWIKI_SPI LIBRARY MUST BE SPECIFICALLY
// CONFIGURED FOR EITHER THE TFT SHIELD OR THE BREAKOUT BOARD.

//This program is a demo of how to use most of the functions
//of the library with a supported display modules.
```

Below the code, a red box highlights the '上传成功' (Upload Success) button. The serial monitor window shows the following output:

```
Writing at 0x0002cf67... (36 %)
Writing at 0x000327df... (45 %)
Writing at 0x00037f05... (54 %)
Writing at 0x0003d4c7... (63 %)
Writing at 0x00042970... (72 %)
Writing at 0x00047f8e... (81 %)
Writing at 0x000506bd... (90 %)
Writing at 0x00058bbb... (100 %)
Wrote 306640 bytes (170347 compressed) at 0x00010000 in 3.3 seconds (effective 743.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

The status bar at the bottom indicates the board is an 'ESP32 Dev Module' with various settings like 'Disabled', 'Default 4MB with spiffs', and 'COM47'.

G、如果显示模块有内容显示，则说明程序运行成功。