1. Introduction to Testing Platform

Development Board : ESP32-WROOM-32E devKit

MCU : ESP32-32E module

Frequency : 240MHz

2. Pin connection instructions

The module can be directly plugged into the ESP32-32E development board, as

shown in the following figure:



Figure 1: Module Inline ESP 32-32E Development Board



Figure 2 Module Back Pins

NOTE:

- A. The **IIC Address** resistor is used to select the IIC slave device address. If it is soldered on the 0x78 side, select the 0x78 slave device address. If it is soldered on the 0x7A side, select the 0x7A slave device address;
- B. The RES pin row is not soldered by default. If the reset function needs to be controlled in the program, it needs to be soldered;

ESP32-32E IIC Test Program Pin Direct Insertion Instructions					
Number	Module pins	Corresponding ESP32-32E development board wiring pins	nt Remarks		
1	GND	GND	OLED screen power supply ground		
2	VCC	5V/3.3V	OLED screen power supply positive		
3	SCL	IO14	IIC bus clock signal		
4	SDA	IO13 IIC bus data signal			
5	RES	Not welded	The pin arrangement is not soldered by default. If the reset function needs to be controlled in the program, it needs to be soldered		

3. Demo Function Description

This sample program uses the ESP32 hardware IIC bus and includes hardware IIC and software IIC test programs, which is located in **Demo_ESP32** directory, as shown in the following figure:



♦ Description of sample program content

The testing program includes the following test items:

- A. Example01-graph_ Test is a graphical display test
- B. Example02 string_ Test is a character display test;
- C. Example03 show_ BMP is a BMP bitmap display test;

♦ Example program IIC slave device address modification instructions

The IIC slave device address has been modified on the hardware, and the software needs to be modified accordingly. First Open any IIC sample program, then locate the **setup** function. If using the 0x7A slave device address, there is no need to annotate the line of code **u8g2.setl2CAddress(0x7A)** (to make them effective). If using the 0x78 slave device address, the line of code **u8g2.setl2CAddress(0x7A)** need to be annotated (to make them ineffective), as shown in the following figure:

```
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. Demo Usage Instructions

♦ Building Development Environment

For specific methods of building a development environment, please refer to the

"Arduino_development_environment_construction_for-ESP32-EN" document in this directory.

♦ Installing software library

After the development environment is set up, the software library used by the sample program needs to be copied to the project library directory so that the sample program can be called. The software library is located in the **Install libraries** directory, as shown in the following figure:



You can also download the latest software library from Github and unzip it (for easy differentiation, you can rename the unzipped folder, as shown in the **Install libraries** directory), and then copy it to the engineering library directory. The download address is as follows:

https://github.com/olikraus/U8g2 Arduino

These software library have been configured and can be directly copied to the project library directory for use. The default path for the engineering library directory is

C:\Users\Administrator\Documents\Arduino\libraries. You can also change the project library directory: open the Arduino IDE software, click **File** ->**Preferences**, and reset the **Sketchbook location** in the pop-up interface, as shown in the following figure:

www.lcdwiki.com

LCDWIKI

💿 clear_Screen Arduino 1.8.19	Preferences			
<u>File</u> Edit <u>Sketch</u> <u>T</u> ools <u>H</u> elp	Settings Network			
New Ctrl+N Open Ctrl+O Open Recent Sketchbook I I Examples TI Close Ctrl+W Save Ctrl+S Save As Ctrl+Shift+S I I Page Setup Ctrl+Shift+P n :	Sketchbook location: C:\Users\Administrator\Documents\Arduino Editor language: English (English) C:\Users\Administrator\Documents\Arduino) Editor font size: 16 Interface scale: Automatic 100 (requires restart of Arduino) Theme: Default theme (requires restart of Arduino) Show verbose output during: V compilation V upload Compiler warnings: None			
Print Ctrl+P nec Preferences Ctrl+Comma	Display line numbers Enable Code Folding Verify code after upload Use external editor			
Quit Ctrl+Q 9 //Arduino Mega2560 10 5	Check for updates on startup ↓ Use accessibility features Additional Boards Manager URLs: s://espressif.github.io/arduino-esp32/package_esp32_index.json ₩ ore preferences can be edited directly in the file			
	C:\Users\Administrator\AppUata\Local\Ardminol>\preferences.txt (edit only when Ardmino is not running) OK Cancel			

Copy the software library to the project library directory, as shown in the following figure:

\bigcirc	- ▶ 库 ▶ 文档 ▶ Arduino ▶ libraries ▶	 搜索 libraries
文件(F)	编辑(E) 查看(V) 工具(T) 帮助(H)	
组织 ▼	☐ 打开 共享 电子邮件 新建文件夹	III - 🗌 🔞
	<mark>文档库</mark> libraries	排列方式: 文件夹▼
	名称	
	📙 U8qlib	Ξ
	📙 U8g2_Arduino	
	L TouchScreen	
- 2	· · · · ·	4

♦ Compile and Run Programs

After the library installation is completed, the sample program can be compiled and run as follows:

- Plug the display module directly into the ESP32 development board, and connect the development board to a PC to power on;
- B. Open Any sample program in the **Demo_ESP32** directory, as shown in the following figure (Here is the Example01-graph_test of the hardware IIC testing program as an example) as shown in the following figure::

	1.04	C. Dampiel's graph	Conception in a second	100 Da -	
O -	Lemo_MC242GX_ESP32-WROOM-32E_	Hardware_IIC 🕨 Exa	mple01-graph_test	▼ 4 ₇	搜索 Example01-graph_test 👂
文件(F) 编	辑(E) 查看(V) 工具(T) 帮助(H)				
组织▼	包含到库中▼ 共享▼ 新建文件夹				ii • 🗍 📀
^ چھ	名称	修改日期	类型	大小	
9 H	Example01-graph_test.ino	2023/8/11 10:22	Arduino file	9 KB	

C. After opening the sample program, select the ESP32 device, as shown in the

following figure:

colligate_test A	duino 1.8.19	
ile Edit Sketch To	ools Help	
colligate_test // IMPORTANT // CONFIGURE	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L
//This progr //of the lib //when using //the SDA pi //if you don //other pins //pin usage	Teensy 4 Security WiFi101 / WiFiNINA Firmware Updater	ESP32S3 Dev Module ESP32C3 Dev Module
	Board: "ESP32 Dev Module" Upload Speed: "921600" CPU Frequency: "240MHz (WiFi/BT)" Flash Frequency: "80MHz"	Boards Manager ESP32S2 Dev Module Arduino AVR Boards ESP32 Dev Module ESP32 Arduino ESP32-WROOM-DA Module Teensyduino ESP32 Wrover Module FSP32 Prover Module ESP32 Prover Module

D. Configure ESP32 Flash, PSRAM, ports, etc. as shown in the following figure:

💿 colligate_test Arduino 1.8.19				
File Edit Sketch To	ols Help			
	Auto Format	Ctrl+T		
colligate test	Archive Sketch			
comgate_test	Fix Encoding & Reload			
// IMPORTANT	Manage Libraries	Ctrl+Shift+I		
// CONFIGURE	Serial Monitor	Ctrl+Shift+M		
//This progr	Serial Plotter	Ctrl+Shift+L		
//of the lib	Teensy 4 Security			
	WiFi101 / WiFiNINA Firmware Updater			
//when using			L	
//the SDA pi	Board: "ESP32 Dev Module"		•	
//if you don	Upload Speed: "921600"		▶ pi	
//other pins	CPU Frequency: "240MHz (WiFi/BT)"		•	
//pin usage	Flash Frequency: "80MHz"		•	
//	Flash Mode: "QIO"		▶ G	
//ESP32-WROO	Flash Size: "4MB (32Mb)"		▶ G	
	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"		۲.	
//Remember t	Core Debug Level: "None"		F.	
/*****	PSRAM: "Disabled"		۱. *	
' * @attention	Arduino Runs On: "Core 1"		۲.	
*	Events Run On: "Core 1"		۲.	
* THE PRESEN	Erase All Flash Before Sketch Upload: "Disabled"		F.	
* WITH CODIN	JTAG Adapter: "Disabled"		F.	
* TIME. AS A	Port		F.	
* DIRECT, IN	Get Board Info			
* FROM THE C	December 11			
* CODING INF	Programmer: "Esptool"		•	

E. Click the upload button to compile and download the program, as shown in the

following figure:

🗢 colligate_test Arduino 1.8.19
<u>File Edit Sketch Tools Help</u>
colligate_test
// IMPORTANT: LCDWIKI_SPI LIBRARY MUST BE SPECIFICALLY // CONFIGURED FOR EITHER THE TFT SHIELD OR THE BREAKOUT BOARD.
<pre>//This program is a demo of how to use most of the functions //of the library with a supported display modules.</pre>
<pre>//when using the BREAKOUT BOARD only and using these hardware spi lines to the LC //the SDA pin and SCK pin is defined by the system and can't be modified. //if you don't need to control the LED pin,you can set it to 3.3V and set the pir //other pins can be defined by youself,for example</pre>
<pre>//pin usage as follow: //</pre>
//ESP32-WROOM-32E: 15 2 27 13 14 12 21 5V GM
//Remember to set the pins to suit your display module!
/*************************************

F. If the following prompt appears, it indicates that the program has been compiled and downloaded successfully, and has already been run:

🐵 GetChipID Arduino 1.8.19	x
<u>File Edit Sketch Tools Help</u>	
	ø
GetChipID	•
/* The true FSD32 chin TD is eccentially its MBC address	÷
Done uploading.	
<pre>Writing at 0x00024f8b (33 %) Writing at 0x0002a2bd (44 %) Writing at 0x0002f6ab (55 %) Writing at 0x000358af (66 %) Writing at 0x0003fe94 (77 %) Writing at 0x00045d9a (88 %) Writing at 0x0004b2ff (100 %) Wrote 261040 bytes (144184 compressed) at 0x00010000 in 2.5 seconds Wash of data warified</pre>	́(е
Hash of data verified. Leaving Hard resetting via RTS pin	₩47

G. If the display module displays content, it indicates that the program has run

successfully.