



Welcome to the JCZN Workshop!

.....Table of contents.....

一、 Introduction	2
二、 Installing using Arduino IDE	2
三、 sample program usage	11



Getting Started

Introduction

The objective of this post is to explain how to upload an Arduino program to the ESP32-2424S012 module, from JCZN .

<http://www.jczn1688.com/zlxz>

The ESP32-C3-MINI-1U WiFi and Bluetooth chip is the latest generation of Espressif products. It has a Single Core 32-bit MCU, which Supporting IEEE 802.11b/g/n (2.4 GHz WiFi) and Bluetooth® 5 (LE).

ESP32-C3-MINI-1U series of SoCs is an ultra-low-power and highly-integrated MCU-based solution that supports 2.4 GHz Wi-Fi and Bluetooth ® Low Energy (Bluetooth LE). The block diagram of ESP32-C3-MINI-1U is shown below.32-bit RISC-V single-core processor with a four-stage pipeline that operates at up to 160MHz. Storage capacities ensured by 400 KB ofSRAM (16 KB for cache) and 384 KB of ROM on the chip, and SPI, Dual SPI, Quad SPI, and QPI interfaces that allow connection to external flash.

Installing using Arduino IDE

Programming the ESP32-C3-MINI-1U

An easy way to get started is by using the familiar Arduino IDE. While this is not necessarily the best environment for working with the ESP32-C3-MINI-1U, it has the advantage of being a familiar application, so the learning curve is flattened.

We will be using the Arduino IDE for our experiments.

1, Installing using Arduino IDE

we first need to install version 1.8.19 of the Arduino IDE (or greater), for example, the Arduino installation was in "C/Programs(x86)/Arduino".

download release link:

<https://downloads.arduino.cc/arduino-1.8.19-windows.exe>

2, This is the way to install Arduino-ESP32 directly from the Arduino IDE.

Add Boards Manager Entry

Here is what you need to do to install the ESP32 boards into the Arduino IDE:

- (1) Open the Arduino IDE.



The screenshot shows the Arduino IDE interface with the title bar "Factory_samples | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. The code editor displays the "Factory_samples" sketch, which includes header file imports, defines for pin numbers, and a class definition for LGFX. The class constructor initializes SPI and sets up configuration parameters like SPI mode and frequency. A note in the code indicates that the configuration is for an ESP-IDF environment. The status bar at the bottom shows "ESP32 Dev Module, FTDI Adapter, Disabled, Huge APP (3MB No OTA/1MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on COM12".

```
#include "LGFx.h"
#include "Arduino.h"
#include <lvgl.h>
#include "demos/lv_demos.h"
#include <LoyanGFX.hpp>
#include <Ticker.h>
#include "CST816D.h"
#include "do_main.h"
#define I2C_SDA 4
#define I2C_SCL 5
#define TP_INT 0
#define TP_RST 1

#define off_pin 35
#define buf_size 120

class LGFX : public lgfx::LGFx_Device
{
    lgfx::Panel_GC9A01 _panel_instance;

    lgfx::Bus_SPI _bus_instance;

public:
    LGFX(void)
    {
        auto cfg = _bus_instance.config();

        // SPIバスの設定
        cfg.spi_host = SPI2_HOST; // 使用するSPIを選択 ESP32-S2,C3 : SPI2_HOST or SPI3_HOST / ESP32 : VSPI_HOST or HSPI_HOST
        // ※ ESP-IDFバージョンアップに伴い、VSPI_HOST , HSPI_HOST の記述は非推奨になるため、エラーが発生する場合は代わりにSPI2_HOST , SPI3_HOSTを使用してください。
        cfg.spi_mode = 0; // SPI通信モードを設定 (0 ~ 3)
        cfg.freq_write = 8000000; // 伝送時のSPI時钟（最高80MHz, 四舍五入为80MHz除以整数得到的值）
        cfg.freq_read = 20000000; // 受取時のSPI時钟
        cfg.spi_3wire = true; // 賽程をMOSIビンで行う場合はtrueを設定
        cfg.use_lock = true; // 使用事務锁时设置为 true
        cfg.dma_channel = SPI_DMA_CH_AUTO; // 使用するDMAチャネルを設定 (0=DMA不使用 / 1=1ch / 2=2ch / SPI_DMA_CH_AUTO=自動設定)
        // ※ ESP-IDFバージョンアップに伴い、DMAチャネルはSPI DMA CH AUTO(自動設定)が推奨になりました。1ch,2chの指定は非推奨になります。
```

- (2) Click on the File menu on the top menu bar.
- (3) Click on the Preferences menu item. This will open a Preferences dialog box.



The screenshot shows the Arduino IDE interface. The title bar reads "Factory_samples | Arduino 1.8.19". The menu bar has "File", "Edit", "Sketch", "Tools", and "Help". The "File" menu is open, showing options like "New", "Open...", "Save", "Print", "Preferences" (which is highlighted with a blue selection bar), and "Quit". Below the menu is a code editor window containing C++ code for an LGFX device. At the bottom of the screen, there is a status bar with the text "1 ESP32 Dev Module, FTDI Adapter, Disabled, Huge APP (3MB No OTA/1MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on COM12".

```
#define _ESP32_PIN_00
#define buf_size 120

class LGFX : public lgfx::LGFX_Device
{
    lgfx::Panel_GC9A01 _panel_instance;
    lgfx::Bus_SPI _bus_instance;

public:
    LGFX(void)
    {
        auto cfg = _bus_instance.config();

        // SPIバスの設定
        cfg.spi_host = SPI2_HOST; // 使用するSPIを選択 ESP32-S2,C3 : SPI2_HOST or SPI3_HOST / ESP32 : VSPI_HOST or HSPI_HOST
        // ※ ESP-IDFバージョンアップに伴い、VSPI_HOST, HSPI_HOSTの記述は非推奨になるため、エラーが表示する場合は代わりにSPI2_HOST, SPI3_HOSTを使用してください。
        cfg.spi_mode = 0;           // SPI通信モードを設定 (0 ~ 3)
        cfg.freq_write = 8000000;   // 伝送時のSPI時钟 (最高80MHz, 四倍五入法で80MHz除以整数得到的值)
        cfg.freq_read = 2000000;    // 受取時のSPI時钟
        cfg.spi_3wire = true;      // 3ワイヤーをMOSIビンで行う場合はtrueを設定
        cfg.use_lock = true;       // 使用する事務锁时设置为 true
        cfg.dma_channel = SPI_DMA_CH_AUTO; // 使用するDMAチャンネルを設定 (0=DMA不使用 / 1=1ch / 2=ch / SPI_DMA_CH_AUTO=自動設定)
        // ※ ESP-IDFバージョンアップに伴い、DMAチャンネルはSPI DMA CH AUTO(自動設定)が推奨になりました。1ch,2chの指定は非推奨になります。
```

Error downloading https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

- (4) You should be on the Settings tab in the Preferences dialog box by default.
- (5) Look for the textbox labeled “Additional Boards Manager URLs”.
- (6) If there is already text in this box add a coma at the end of it, then follow the next step.
- (7) Paste the following link into the text box :

Stable release link:

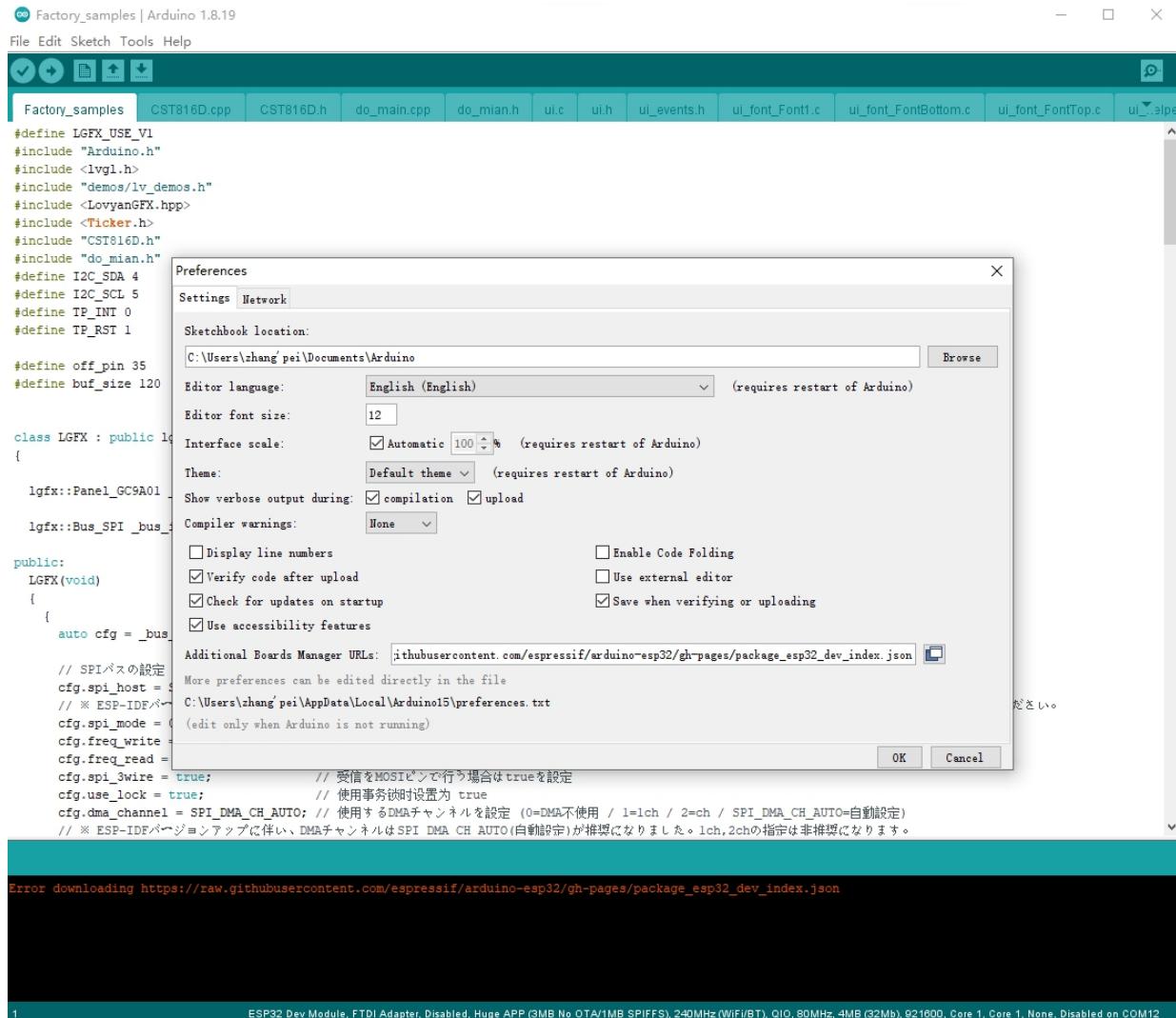
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

Development release link:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

- (8) Click the OK button to save the setting.

The textbox with the JSON link in it is illustrated here:



- (9) In the Arduino IDE click on the Tools menu on the top menu bar.
- (10) Scroll down to the Board: entry
- (11) A submenu will open when you highlight the Board: entry.
- (12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.
- (13) In the search box in the Boards Manager enter "esp32".



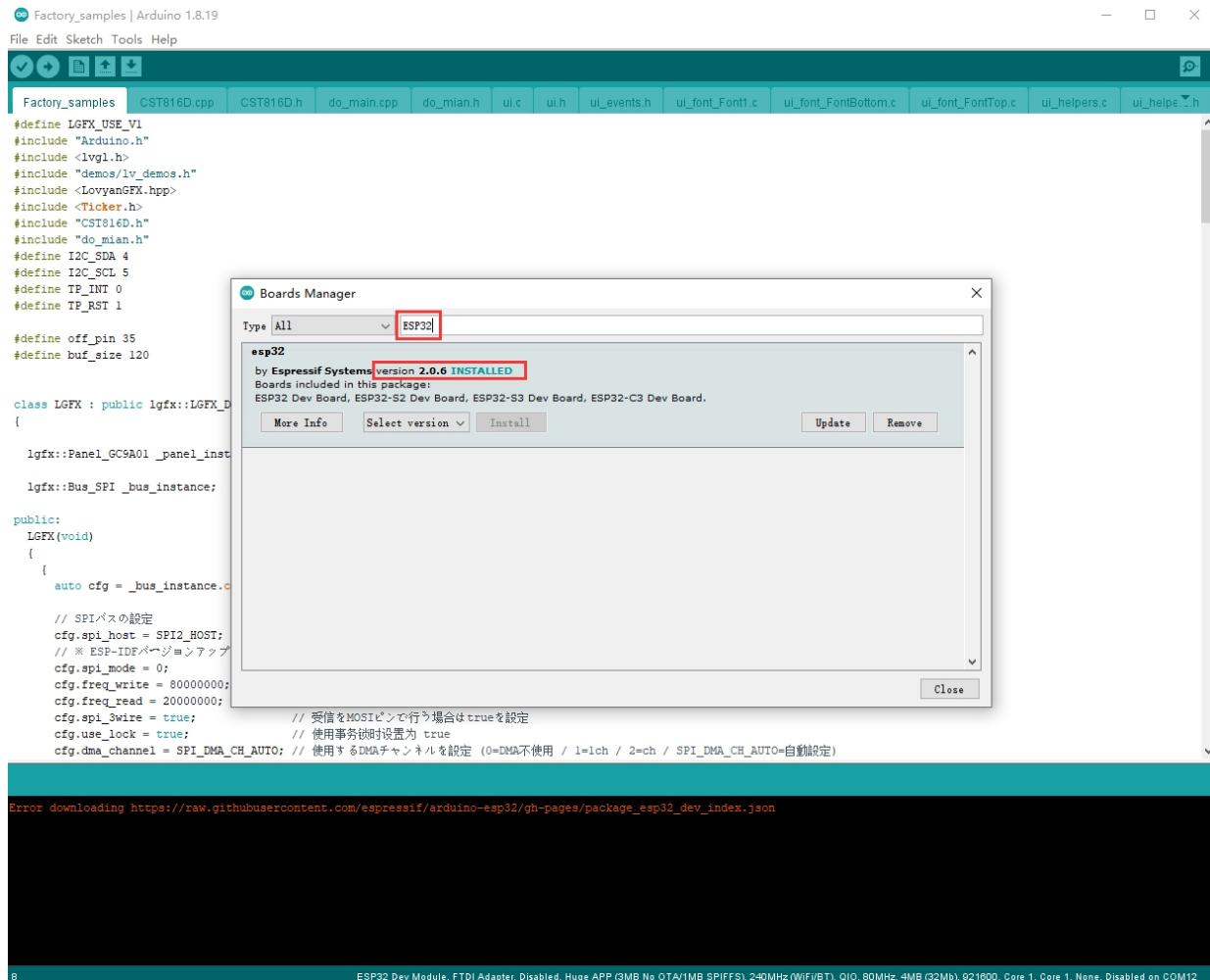
The screenshot shows the Arduino IDE interface with the following details:

- File Edit Sketch Tools Help** (Toolbar)
- Tools** menu open:
 - Auto Format
 - Archive Sketch
 - Fix Encoding & Reload
 - Manage Libraries...
 - Serial Monitor
 - Serial Plotter
 - WiFi101 / WiFiNINA Firmware Updater
 - Board: "ESP32C3 Dev Module"
 - Upload Speed: "921600"
 - USB CDC On Boot: "Disabled"
 - CPU Frequency: "160MHz (WiFi)"
 - Flash Frequency: "80MHz"
 - Flash Mode: "QIO"
 - Flash Size: "4MB (32Mb)"
 - Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
 - Core Debug Level: "None"
 - Erase All Flash Before Sketch Upload: "Disabled"
 - JTAG Adapter: "Integrated USB JTAG"
 - Port
 - Get Board Info
 - Programmer
 - Burn Bootloader
- Boards Manager...** (Submenu of Tools)
- Arduino AVR Boards** (Submenu of Boards Manager...)
- ESP32 Arduino** (Submenu of AVR Boards)
- ESP32S3 Dev Module** (Selected board)
- ESP32C3 Dev Module**
- ESP32S2 Dev Module**
- ESP32 Dev Module**
- ESP32-WROOM-DA Module**
- ESP32 Wrover Module**
- ESP32 PICO-D4**
- ESP32-S3-Box**
- ESP32-S3-USB-OTG**
- ESP32S3 CAM LCD**
- ESP32S2 Native USB**
- ESP32 Wrover Kit (all versions)**
- UM TinyPICO**
- UM FeatherS2**
- UM FeatherS2 Neo**
- UM TinyS2**
- UM RMP**
- UM TinyS3**
- UM PROS3**
- UM FeatherS3**
- S.ODI Ultra v1**
- microS2**
- MagicBit**
- Turta IoT Node**
- TTGO LoRa32-OLED**
- TTGO T1**
- TTGO T7 V1.3 Mini32**
- TTGO T7 V1.4 Mini32**
- TTGO T-OI PLUS RISC-V ESP32-C3**
- XinaBox CW02**
- SparkFun ESP32 Thing**
- SparkFun ESP32 Thing Plus**
- SparkFun ESP32 Thing_Plus_C**

Error downloading https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

(14) You should see an entry for "esp32 by Espressif Systems". Highlight this entry and click on the Install button.

This will install the ESP32 boards into your Arduino IDE



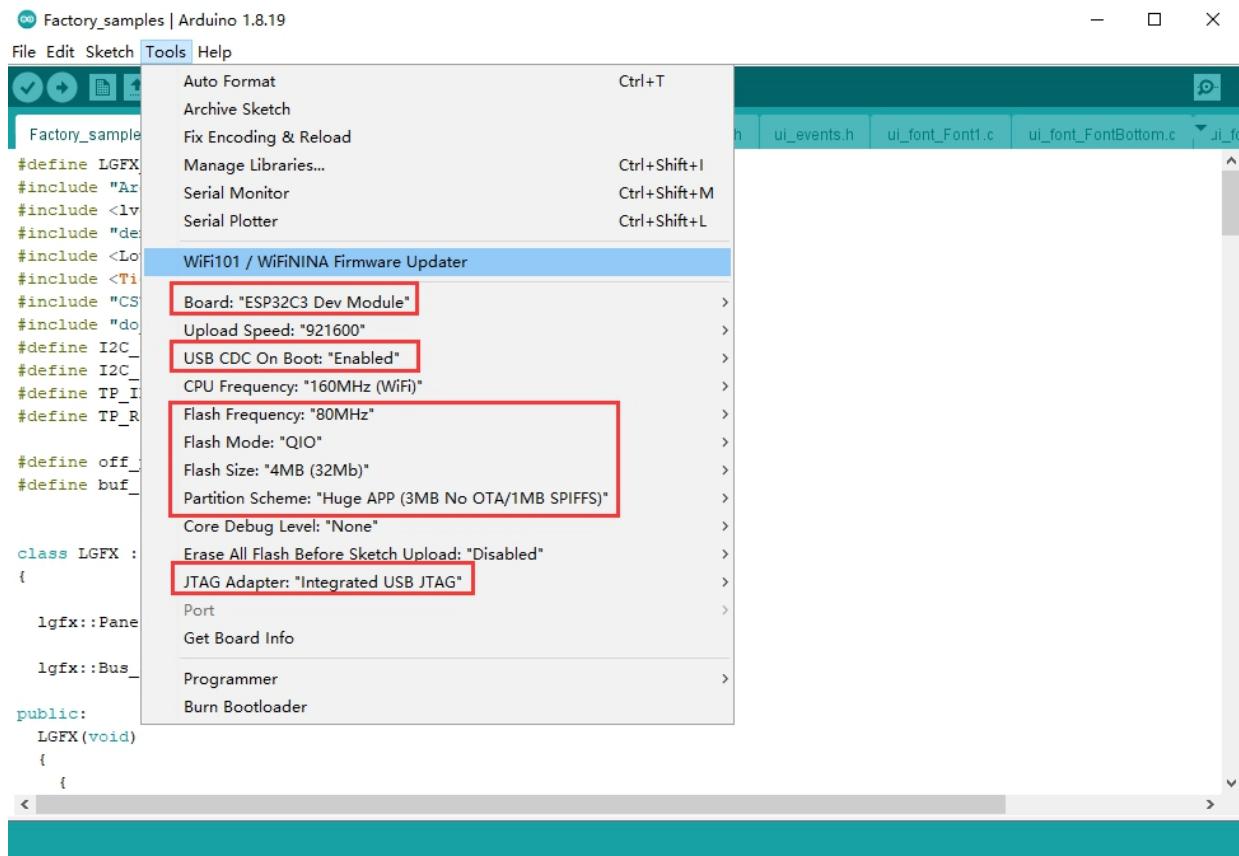
Once the installation completes, we need to select the correct board options for the "ESP32 Arduino" board. In the board type, in the tools tab, we choose "ESP32C3 Dev Module".



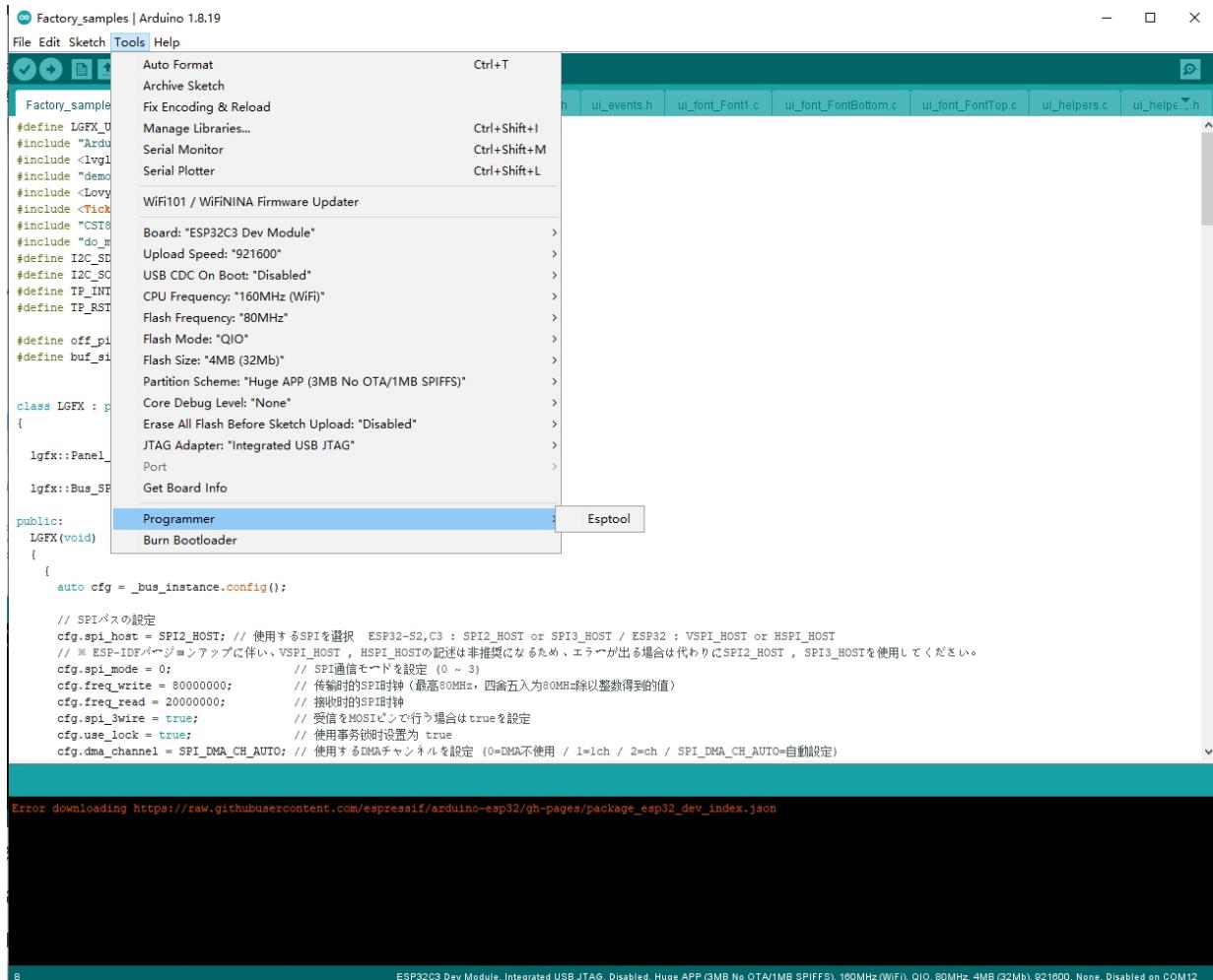
深圳市晶彩智能有限公司
Shenzhen Jingcai Intelligent Co., Ltd

The screenshot shows the Arduino IDE interface with the following details:

- Menu Bar:** File, Edit, Sketch, Tools, Help
- Toolbar:** Auto Format, Archive Sketch, Fix Encoding & Reload, Manage Libraries..., Serial Monitor, Serial Plotter, WiFi101 / WiFiINA Firmware Updater.
- Code Area:** The code editor displays the `Factory_samples` sketch. A red box highlights the line `Board: "ESP32C3 Dev Module"`. Below it is a list of board configuration parameters:
 - Upload Speed: "921600"
 - USB CDC On Boot: "Disabled"
 - CPU Frequency: "160MHz (WiFi)"
 - Flash Frequency: "80MHz"
 - Flash Mode: "QIO"
 - Flash Size: "4MB (32Mb)"
 - Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
 - Core Debug Level: "None"
 - Erase All Flash Before Sketch Upload: "Disabled"
 - JTAG Adapter: "Integrated USB JTAG"
- Toolbars:** Top right toolbars include `Ctrl+T`, `Ctrl+Shift+I`, `Ctrl+Shift+M`, `Ctrl+Shift+L`.
- Bottom Status Bar:** Shows the message: `ESP32C3 Dev Module, Integrated USB JTAG, Disabled, Default 4MB with spiffs (1.2MB APP/1.5M`
- Right Sidebar:** A floating window titled "Boards Manager..." lists available ESP32 boards. A red box highlights the selected option: `ESP32 Arduino`. Other options listed include:
 - ESP32S3 Dev Module
 - ESP32C3 Dev Module** (selected)
 - ESP32S2 Dev Module
 - ESP32 Dev Module
 - ESP32-WROOM-DA Module
 - ESP32 Wrover Module
 - ESP32 PICO-D4
 - ESP32-S3-Box
 - ESP32-S3-USB-OTG
 - ESP32S3 CAM LCD
 - ESP32S2 Native USB
 - ESP32 Wrover Kit (all versions)
 - UM TinyPICO
 - UM FeatherS2
 - UM FeatherS2 Neo
 - UM TinyS2
 - UM RMP
 - UM TinyS3
 - UM PROS3
 - UM FeatherS3
 - S.ODI Ultra v1
 - microS2
 - MagicBit
 - Turta IoT Node
 - TTGO LoRa32-OLED
 - TTGO T1
 - TTGO T7 V1.3 Mini32
 - TTGO T7 V1.4 Mini32
 - TTGO T-OI PLUS RISC-V ESP32-C3
 - XinaBox CW02
 - SparkFun ESP32 Thing
 - SparkFun ESP32 Thing Plus
 - SparkFun ESP32 Thing Plus C
 - SparkFun ESP32-S2 Thing Plus
 - SparkFun ESP32 MicroMod



Set and In the programmer entry of the same tab, we choose “esptool”.



It's important to note that after the code is uploaded, the device will start to run it. So, if we want to upload a new program, we need to reset the power of the device, in order to guarantee that it enters flashing mode again.

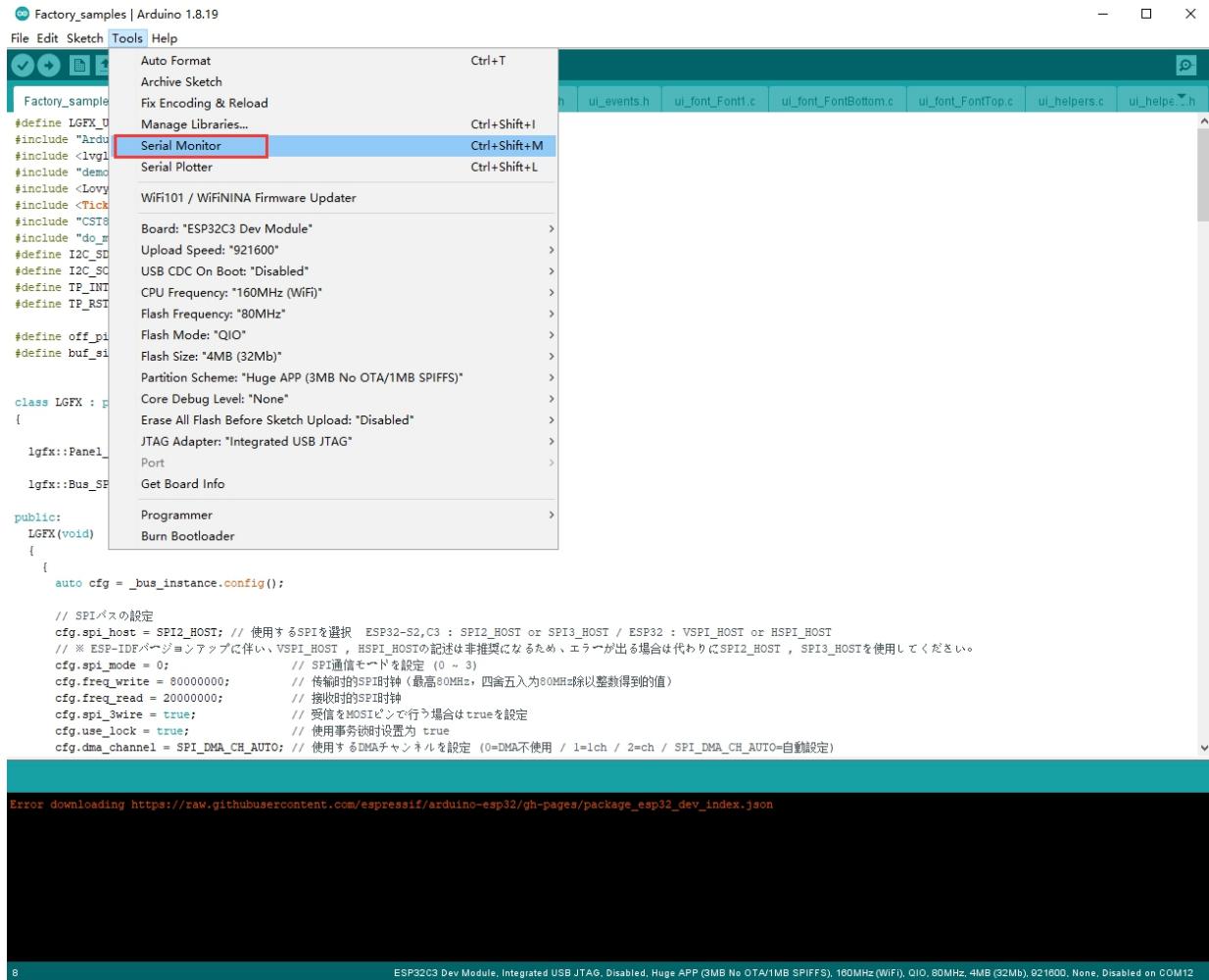
First program

Since this platform is based on Arduino, we can use many of the usual functions. As an example for the first program, the code below starts the Serial port and prints "hello from ESP32" every second.

```
void setup() {
    Serial.begin(115200);
}

void loop() {
    Serial.println("hello from ESP32");
    delay(1000);
}
```

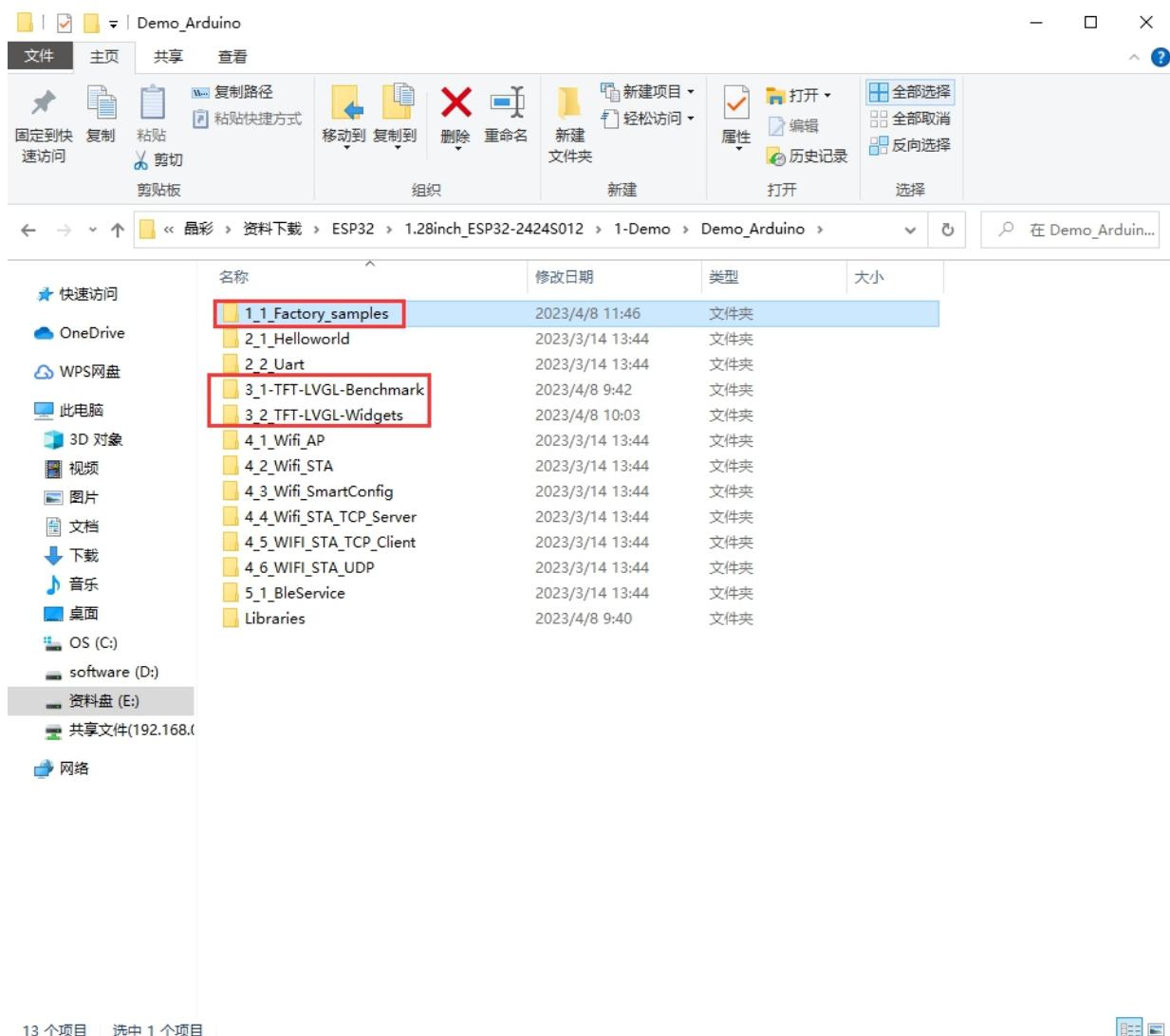
If everything is working fine, we will see the output in the serial console shown.



Again thank you for so much concern.. Hopefully, it's the beginning of a wonderful relationship!

Sample program usage

At present, only a preliminary explanation and introductory use are given to the samples displayed on the screen, and the corresponding examples in the data center are found, as shown in the figure:



13 个项目 选中 1 个项目



The examples in the red circle are all based on the LovyanGFX library as the basic application. This library supports various commonly used driver chips, such as ST7735, ST7789, ILI9341, etc., and has good compatibility.

LovyanGFX library file installation:

Open the library manager in Arduino, search for LovyanGFX, and click instal .



Factory_samples | Arduino 1.8.19

File Edit Sketch Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 / WiFi/NINA Firmware Updater

Board: "ESP32C3 Dev Module" >

Upload Speed: "921600" >

USB CDC On Boot: "Disabled" >

CPU Frequency: "160MHz (WiFi)" >

Flash Frequency: "80MHz" >

Flash Mode: "QIO" >

Flash Size: "4MB (32Mb)" >

Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)" >

Core Debug Level: "None" >

Erase All Flash Before Sketch Upload: "Disabled" >

JTAG Adapter: "Integrated USB JTAG" >

Port >

Get Board Info >

Programmer >

Burn Bootloader >

LGFX(void)

{

auto cfg = _bus_instance.config();

// SPIバスの設定

cfg.spi_host = SPI2_HOST; // 使用するSPIを選択 ESP32-S2,C3 : SPI2_HOST or SPI3_HOST / ESP32 : VSPI_HOST or HSPI_HOST
// ※ ESP-IDFページに伴い、VSPI_HOST, HSPI_HOSTの記述は非推奨になるため、エラーが出る場合は代わりにSPI2_HOST, SPI3_HOSTを使用してください。

cfg.spi_mode = 0; // SPI通信モードを設定 (0 ~ 3)

cfg.freq_write = 8000000; // 传输侧的SPI时钟 (最高80MHz, 四舍五入为80MHz除以整数得到的值)

cfg.freq_read = 2000000; // 接收侧的SPI时钟

cfg.spi_3wire = true; // 受信をMOSIビンで行う場合はtrueを設定

cfg.use_lock = true; // 使用事务锁时设置为 true

cfg.dma_channel = SPI_DMA_CH_AUTO; // 使用するDMAチャネルを設定 (0=DMA不使用 / 1=lch / 2=ch / SPI_DMA_CH_AUTO=自動設定)

Error downloading https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

8 ESP32C3 Dev Module, Integrated USB JTAG, Disabled, Huge APP (3MB No OTA/1MB SPIFFS), 160MHz (WiFi), QIO, 80MHz, 4MB (32Mb), 921600, None, Disabled on COM12



The screenshot shows the Arduino IDE interface with the 'Factory_samples' sketch open. A 'Library Manager' window is displayed, showing search results for 'LovyanGFX'. The 'ESPDisplay' library is listed above it. The 'LovyanGFX' library entry is highlighted with a red box. It shows the library version as 'Version 1.1.5 INSTALLED' and provides a brief description: 'TFT LCD Graphics driver with touch for ESP32, ESP8266, SAMD21, SAMD51, RP2040 Supports TFT LCD displays using drivers that operate with hardware SPI: ESP32, ESP8266, SAMD21, SAMD51, RP2040, M5Stack, M5StickCore2, M5StickCPlus, ODRDIO-GO, TTGO T-Watch, TTGO T-Wristband, ESP-WROVER-KIT, WiFiBoY, MakePython, HX8357, ILI9163, ...'. The code editor at the bottom shows the start of the 'Factory_samples' sketch.

```
#include "LGFx.h"
#include <lvgl.h>
#include "demos/lv_demos.h"
#include <LovyanGFX.hpp>
#include <Ticker.h>
#include "CST816D.h"
#include "do_mian.h"
#define I2C_SDA 4
#define I2C_SCL 5
#define TP_INT 0
#define TP_RST 1

#define off_pin 35
#define buf_size 120

class LGFx : public lgfx::LGFX_I
{
    lgfx::Panel_GC9A01 _panel_inst;

    lgfx::Bus_SPI _bus_instance;

public:
    LGFx(void)
    {
        auto cfg = _bus_instance.*;

        // SPIバスの設定
        cfg.spi_host = SPI2_HOST;
        // ※ ESP-IDF版マジックアップ
        cfg.spi_mode = 0;
        cfg.freq_write = 8000000;
        cfg.freq_read = 20000000;
        cfg.spi_3wire = true;
    }
}
```

Although the Arduino_GFX library has many advantages, it may also have a troublesome place for ordinary users, that is, after the installation

About the use of Factory_samples:

Find the data center 1_1_Factory_samples

As shown:



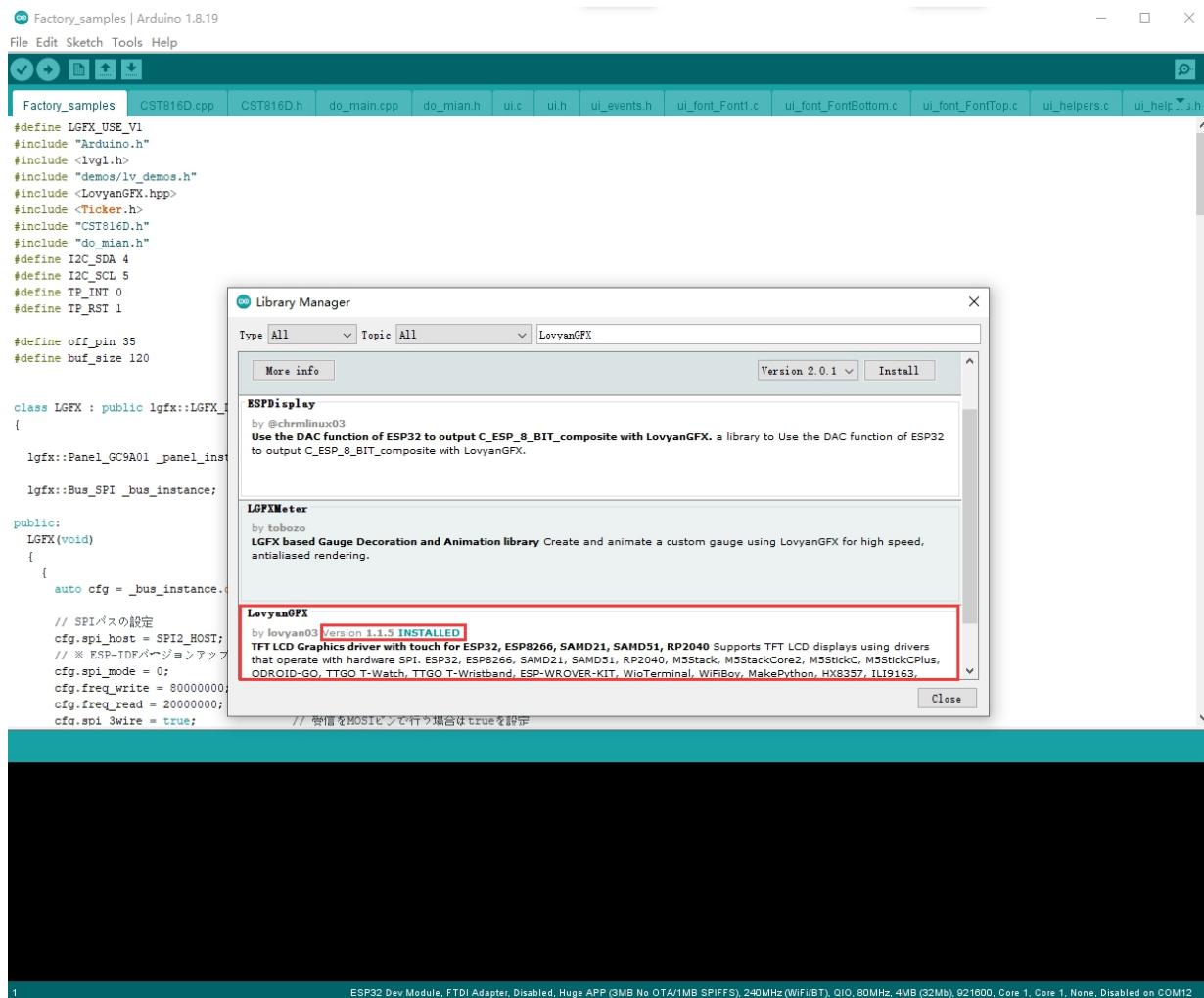
The screenshot shows a Windows File Explorer window with the following details:

- Toolbar:** Includes icons for file operations like Cut, Copy, Paste, Move, Delete, Rename, and New.
- Address Bar:** Shows the path: 资料盘 (E:) > 晶彩 > 资料下载 > 1.28inch_ESP32-2424S012 > 1-Demo > Demo_Arduino > Demo_Arduino.
- Search Bar:** A search field with placeholder text "在 Demo_Arduin...".
- File List:** A table showing the contents of the selected folder:

名称	修改日期	类型	大小
1_1_Factory_samples	2023/4/8 11:46	文件夹	
2_1_Helloworld	2023/3/14 13:44	文件夹	
2_2_Uart	2023/3/14 13:44	文件夹	
3_1-TFT-LVGL-Benchmark	2023/4/8 9:42	文件夹	
3_2-TFT-LVGL-Widgets	2023/4/8 10:03	文件夹	
4_1_Wifi_AP	2023/3/14 13:44	文件夹	
4_2_Wifi_STA	2023/3/14 13:44	文件夹	
4_3_Wifi_SmartConfig	2023/3/14 13:44	文件夹	
4_4_Wifi_STA_TCP_Server	2023/3/14 13:44	文件夹	
4_5_WIFI_STA_TCP_Client	2023/3/14 13:44	文件夹	
4_6_WIFI_STA_UDP	2023/3/14 13:44	文件夹	
5_1_BleService	2023/3/14 13:44	文件夹	
Libraries	2023/4/8 9:40	文件夹	
- Left Sidebar:** Shows a tree view of local drives and network locations, with "资料盘 (E:)" currently selected.
- Status Bar:** Displays "13 个项目 选中 1 个项目".

Download library files .

LovyanGFX library



After compiling, you can run 1_1_Factory_samples normally.