Shenzhen Juntong Technology Co., LTD		Document Number	Version number	Classification level
		202512171536	V1.0	External Public
Document Name	Full-Color Display Development Manual		Date	2025-12-17

Full-Color Display Development Manual

Document Revision Record

Version Number	Change (+/-) instructions	Date
V1.0	Create	2025-12-25

Catalogue

1. Introduction

1.1. Document Purposes

This document is intended to provide a systematic description of **secondary development for full-color display screens**, helping users and developers quickly understand and use the hardware, protocols, and multi-platform sample software provided by this project.

With this instruction document, readers can:

- Properly connect the flexible display to the external control device
- Control the flexible display using sample programs provided for MCU, Windows, iOS, and Android platforms
- Develop or integrate other control systems based on publicly available communication protocols

1.2. Scope of Application

This document is for the following persons:

- **Application software developers**

  A developer who uses Windows, iOS, or Android platforms and controls the content or behavior of flexible displays according to communication protocols.
- **System integration engineer**

  Engineers who need to integrate flexible display screens into existing systems or products and control them through protocols.
- **Embedded systems developer (control end)**

  Developers who implement communication protocols on external MCUS, master boards, or other control devices to interact with flexible displays.
- **Product development and technical support personnel**

  People who need to know about flexible display interfaces, communication methods, and basic usage methods.

2. Equipment Overview

2.1. Equipment Introduction

Full-color flexible display screen, with a flexible screen and drip glue process, can be directly pasted, lightweight and easy to install. It supports mobile APP and remote control. The mobile APP can set functions such as text, animation, doodle, music rhythm, etc. It can be used in the following scenarios:

- In-vehicle applications

- Window display glass

- Storefront application

- Devil's Eye App

- Creative apps

2.2. Power Supply requirements

The display supply requires DC-5V. Do not use the voltage beyond the range, otherwise the device will not work properly or even be damaged.

Note: Different resolutions of displays may have different power requirements. Please refer to the display specification sheet.

3. Quick Start Guide

3.1. Bluetooth Debugging

The flexible screen products mentioned in this document use Bluetooth, which is the Bluetooth Low Energy (BLE) protocol available only in versions 4.0 and above.

3.1.1. Device Information

1. Device Name

   CoolLEDUX, JotuPix

2. Bluetooth Service UUID and eigenvalue UUID

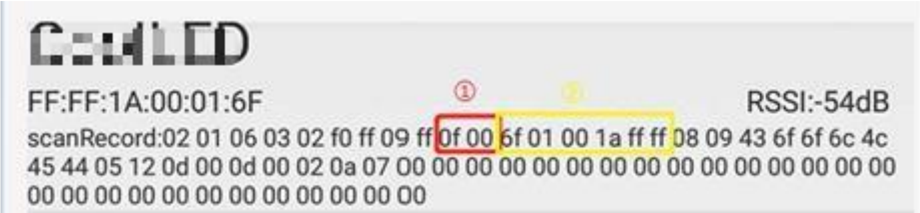
Basic UUID: 0x0000xxxx-0000-1000-8000-00805F9B34FB

Bluetooth service UUID: 0xFFF0

The Bluetooth eigenvalue UUID is: 0xFFF1

3. Device ID

The device ID is located in the first and second bytes after the Manufacturer Specific Data in BLE broadcast data, totaling two bytes. For example, data FF0F00. Where FF is Manufacturer Specific Data (manufacturer-specific data), 000F is the device ID, with the lower byte first and the higher byte second. The data in the red box in the following figure.

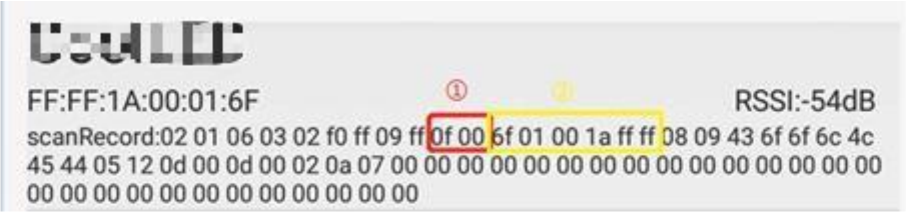

[redacted]

FF:FF:1A:00:01:6F RSSI:-54dB

scanRecord:02 01 06 03 02 f0 ff 09 ff 0f 00 6f 01 00 1a ff ff 08 09 43 6f 6f 6c 4c 45 44 05 12 0d 00 0d 00 02 0a 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

4. Device MAC address

Since IOS cannot obtain MAC addresses directly through the system interface, the MAC address of the device is also located in the third to eighth bytes after the Manufacturer Specific Data in BLE broadcast data, totaling six bytes. The data in the yellow box in the following figure. FFFF1A00016F is the MAC address, with the lower byte in front and the higher byte behind.


[redacted]

FF:FF:1A:00:01:6F RSSI:-54dB

scanRecord:02 01 06 03 02 f0 ff 09 ff 0f 00 6f 01 00 1a ff ff 08 09 43 6f 6f 6c 4c 45 44 05 12 0d 00 0d 00 02 0a 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

3.1.2. Hardware preparation

1. A set of flexible display screens

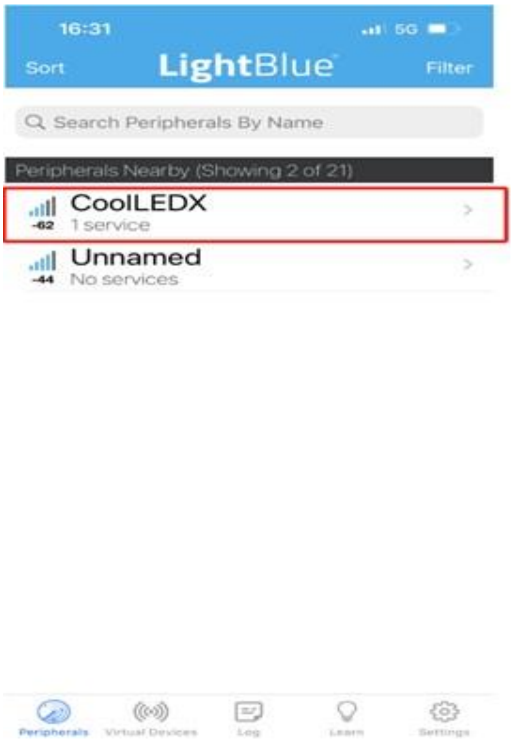
2. One mobile phone or tablet.

3.1.3. Software preparation

BLE Debug Assistant APP One. Here the iPhone app "Lightblue" is used as an illustration, and Android phones can choose the corresponding BLE Debug Assistant.

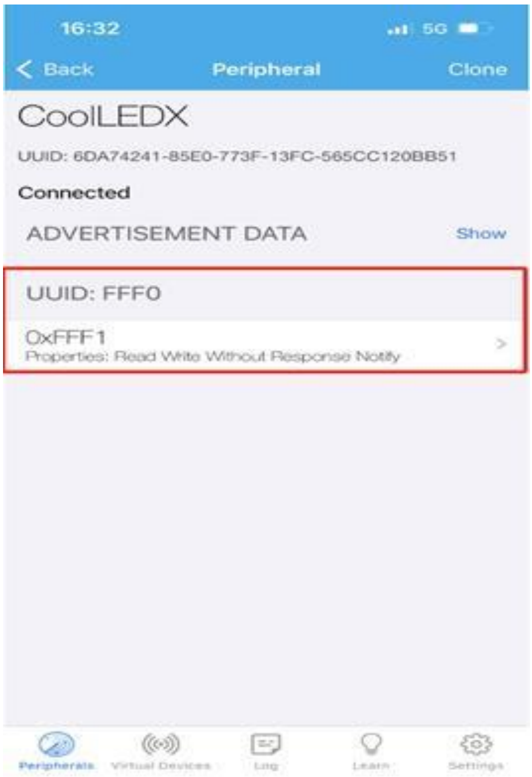
3.1.4. Debugging Steps

1. Power the flexible display, and if it's a USB-powered version, try to power it with a phone charger of 5V/2A or more.
2. Open the BLE debugging software and click on the device name to connect. The device name is CoolLEDUX.
3. After the connection is successful, select the service ID with UUID 0xFFF0 and the feature ID with UUID 0xFFF1 for communication.



CoolLEDX
1 service

Unnamed
No services



CoolLEDX

UUID: 6DA74241-85E0-773F-13FC-565CC120BB51

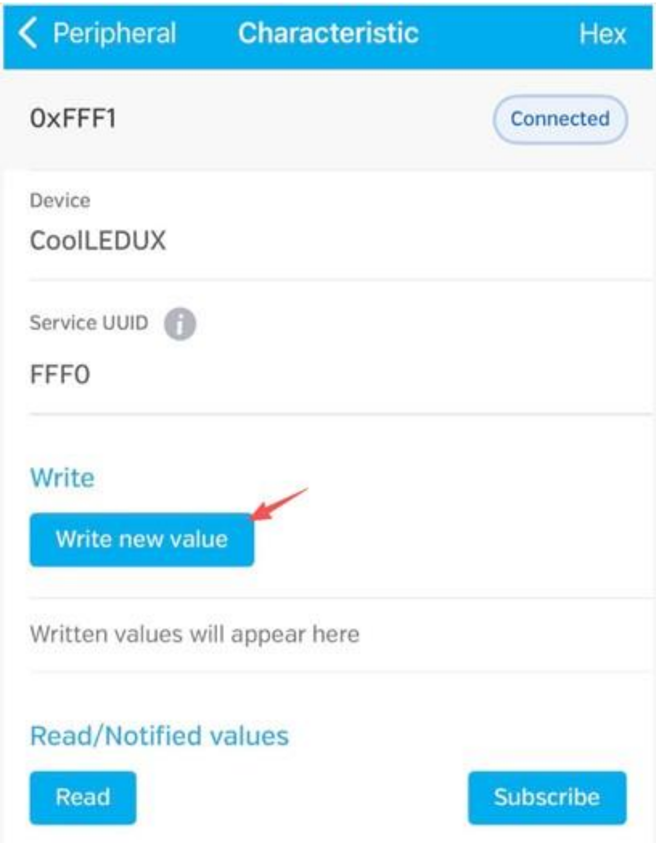
Connected

ADVERTISEMENT DATA

UUID: FFF0

0xFFF1
Properties: Read Write Without Response Notify

4. After connecting the UUID, enter the data sending interface. The data sending format is HEX. Click "Write new value" as shown in the figure.



0xFFF1

Device
CoolLEDUX

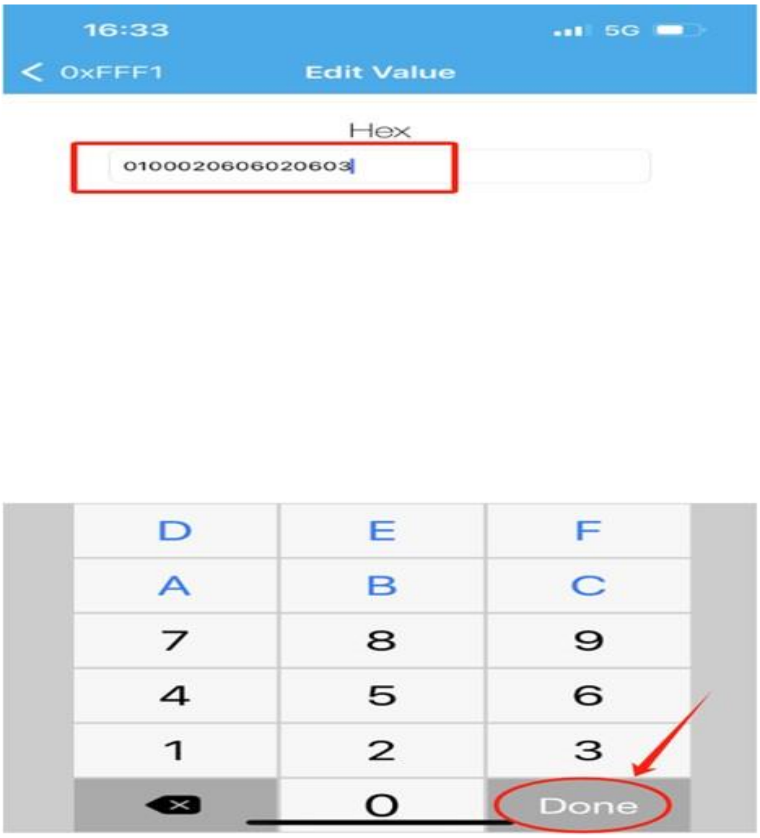
Service UUID
FFF0

Write

Written values will appear here

Read/Notified values

5. Enter the sample instruction and click Send.



0100020606020603

Sample instructions (in hexadecimal) :

Turn on the screen: 0100020605020503

Screen off: 01000206050003

3.2. Serial port communication

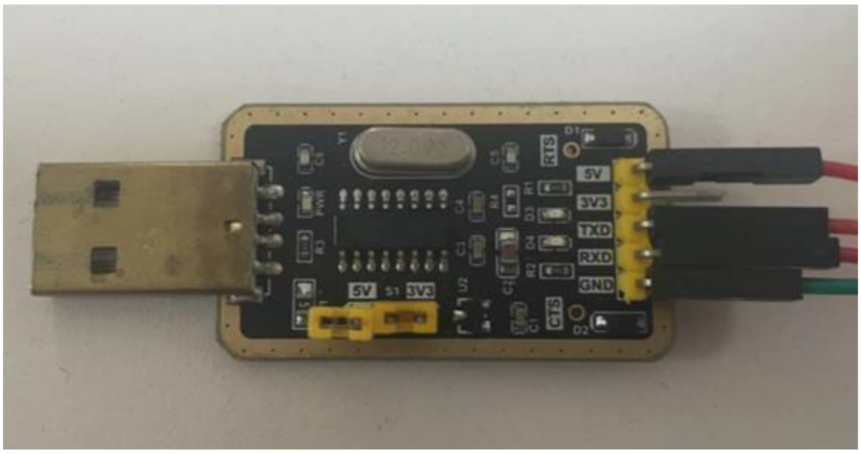
The serial port uses bidirectional communication mode, serial port baud rate: 400,000, 1 stop bit, 8 data bits, no parity.

If control is not possible, swap RX and TX.

3.2.1. Hardware readiness

1. One flexible display screen

2. Usb-to-ttl serial port clipboard

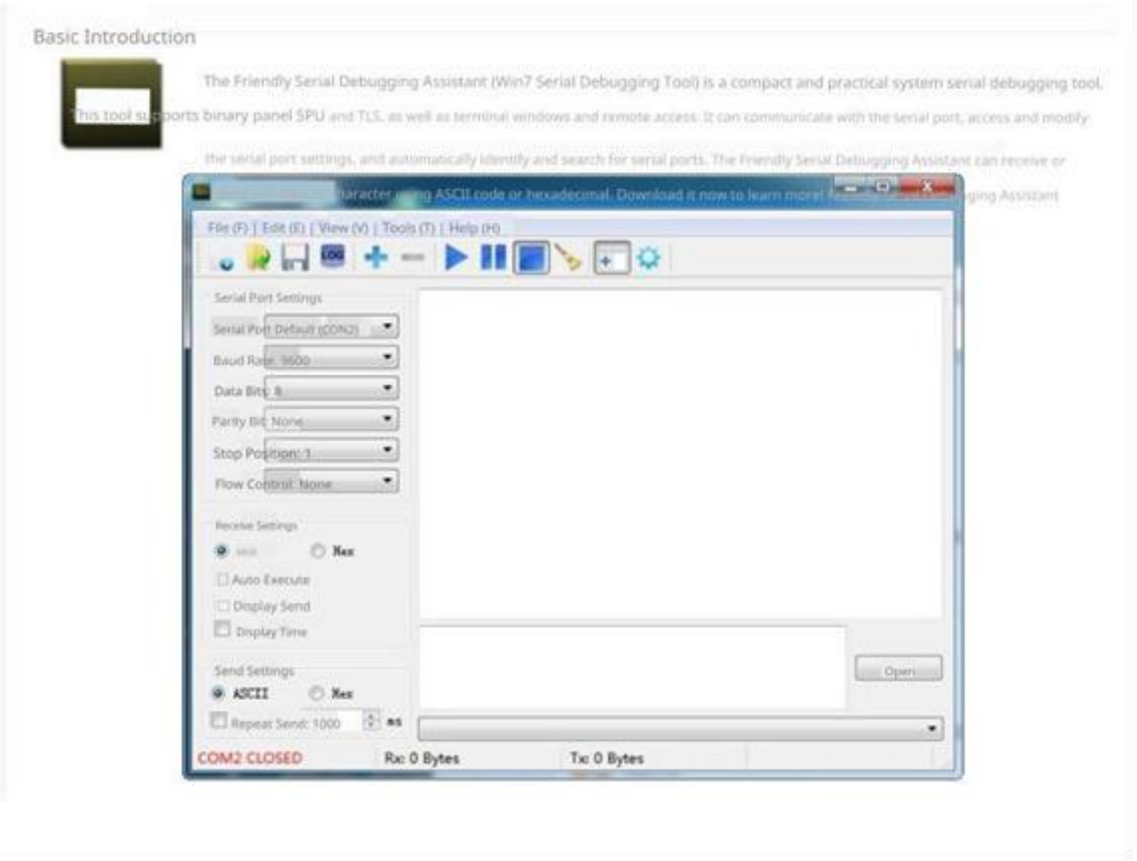


Or use UART to RS232:

UART to RS232 serial clipboard	USB to serial cable

3.2.2. Software preparation

Serial Port Debugging Assistant, which can be downloaded from the Internet, any available one will do.



Basic Introduction

The Friendly Serial Debugging Assistant (Win7 Serial Debugging Tool) is a compact and practical system serial debugging tool. This tool supports binary panel SPU and TLS, as well as terminal windows and remote access. It can communicate with the serial port, access and modify the serial port settings, and automatically identify and search for serial ports. The Friendly Serial Debugging Assistant can receive or [illegible] haracter using ASCII code or hexadecimal. Download it now to learn more! [illegible] ging Assistant

Serial Port Settings

Serial Port: Default (COM2)

Baud Rate: 9600

Data Bits: 8

Parity Bit: None

Stop Position: 1

Flow Control: None

Receive Settings

● [illegible] ○ Hex

□ Auto Execute

□ Display Send

□ Display Time

Send Settings

● ASCII ○ Hex

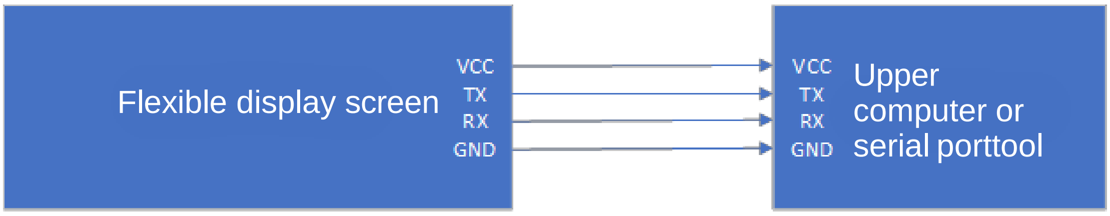
□ Repeat Send: 1000 ms

COM2 CLOSED Rx: 0 Bytes Tx: 0 Bytes

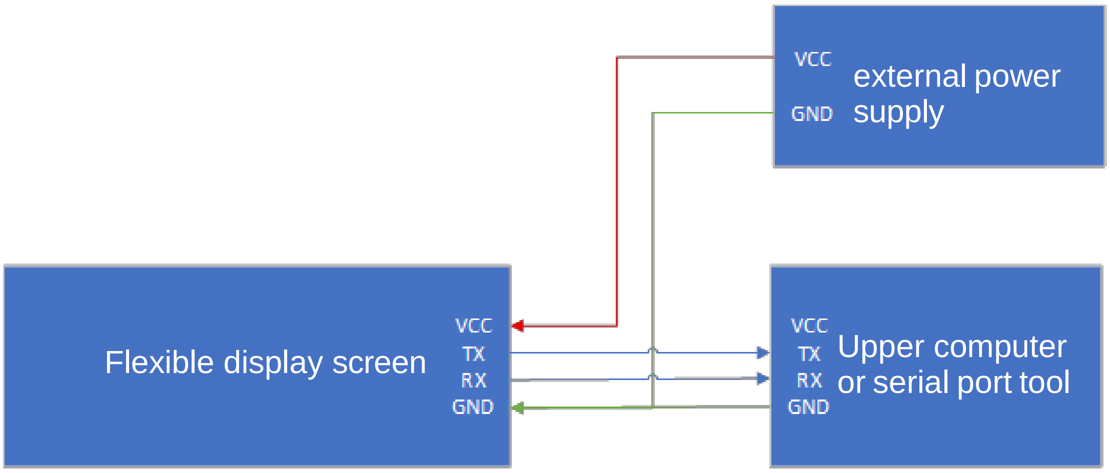
3.2.3. Hardware connection

Connect the aforementioned hardware devices according to the basic electrical characteristics of serial port connections, namely RX->TX, TX->RX. And lead out the power supply line, either an external USB cable or a DC regulated power supply of 5V.

Do not connect the power cord in reverse!

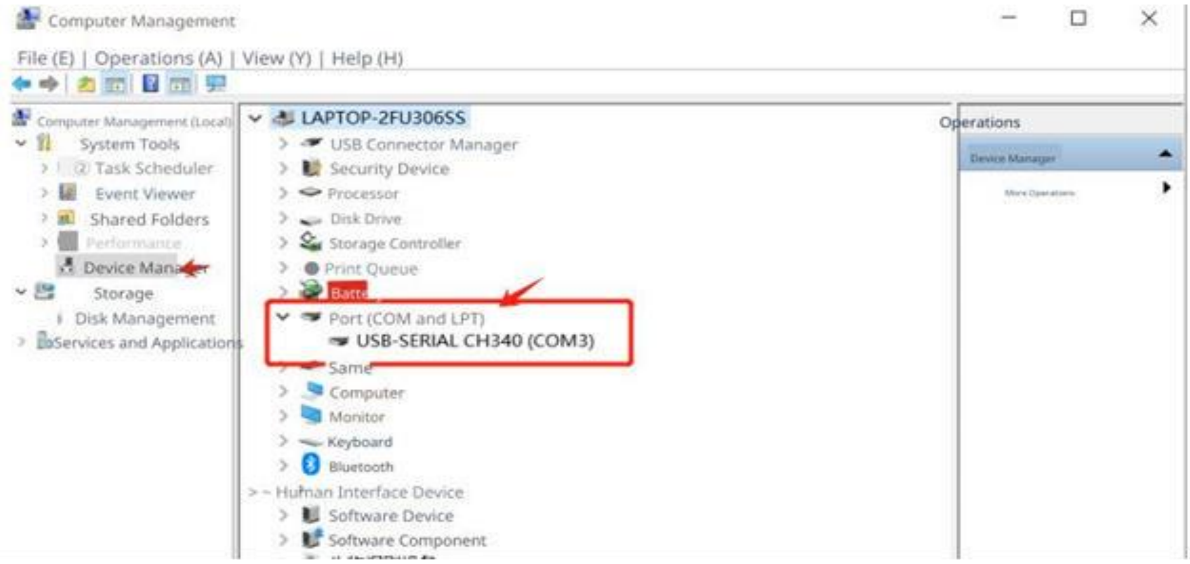


If your computer's USB power is insufficient, you need an external power supply. The external power supply is connected as follows:

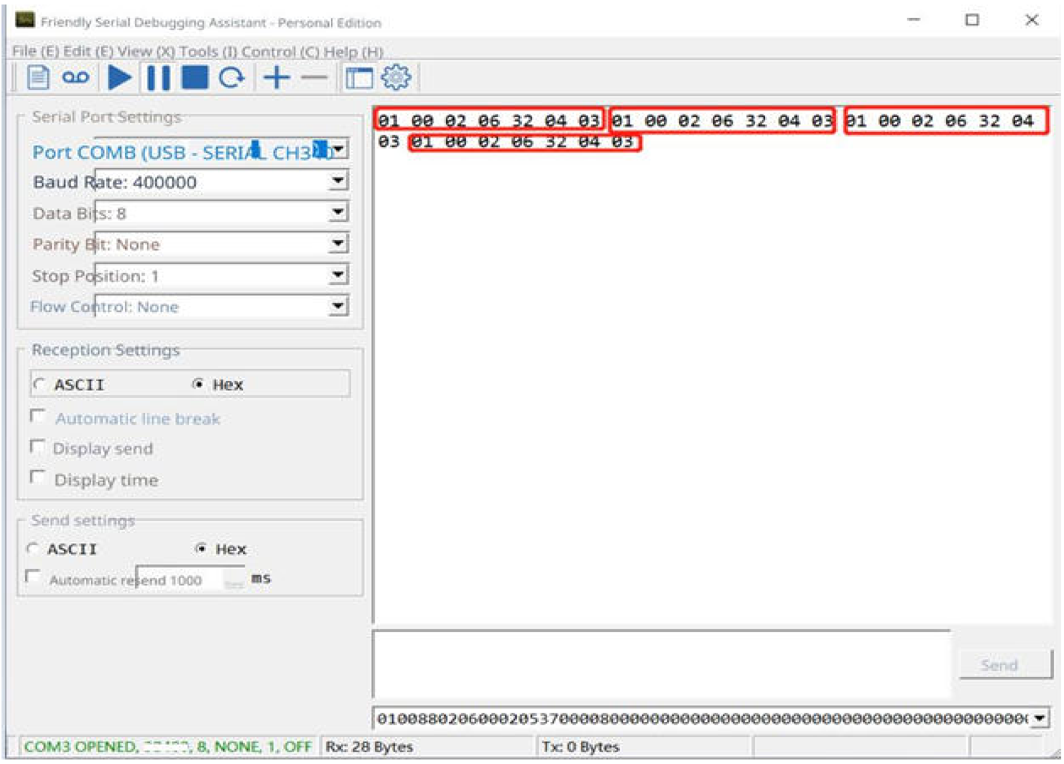


3.2.4. Serial port test

1. After the connection is successful, turn on your computer and go to "Computer" -> "Device Manager" -> "Port" to view the serial number.



2. Open the Serial Port Debug Assistant, select the corresponding serial port number, baud rate 400000, data bit 8, check bit None, stop bit 1, flow control None. Both the receive setting and the send setting are Hex.



Friendly Serial Debugging Assistant - Personal Edition
File (E) Edit (E) View (X) Tools (I) Control (C) Help (H)
Send

Serial Port Settings

Port COMB (USB - SERIAL CH340

Baud Rate: 400000

Data Bits: 8

Parity Bit: None

Stop Position: 1

Flow Control: None

Reception Settings

○ ASCII ● Hex

□ Automatic line break

□ Display send

□ Display time

Send settings

○ ASCII ● Hex

□ Automatic resend 1000 ms

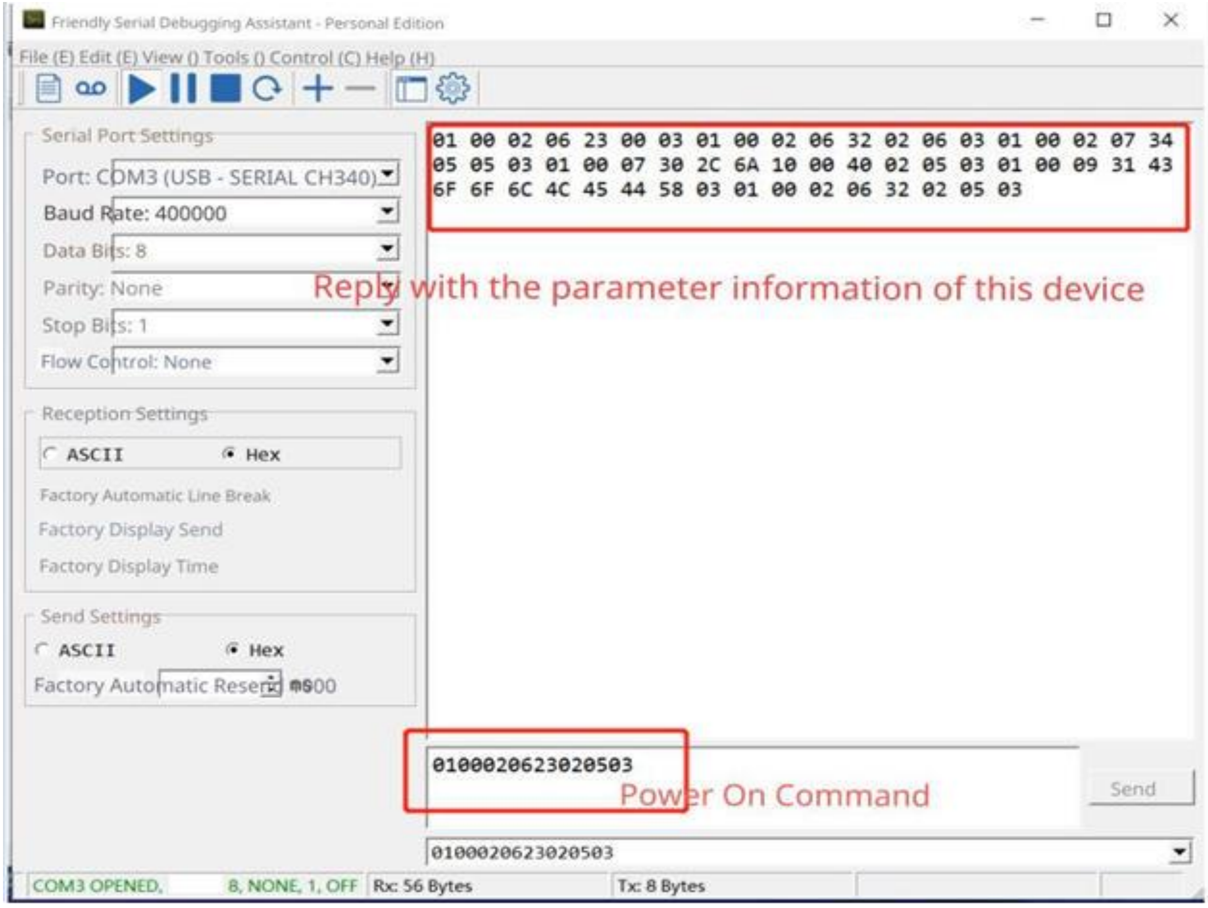
```
01 00 02 06 32 04 03 01 00 02 06 32 04 03 01 00 02 06 32 04
03 01 00 02 06 32 04 03
```

```
0100880206000205370000800000000000000000000000000000000000000000000(
```

COM3 OPENED, [illegible], 8, NONE, 1, OFF | Rx: 28 Bytes | Tx: 0 Bytes

Note: After opening the serial port, you will receive a power-on command from the device. Ignore it.

3. Send startup instructions to test the device. Type: 0100020623020503 (hexadecimal) in the serial tool send box, and the device will boot up, display the ID, and then show the preset content.


Friendly Serial Debugging Assistant - Personal Edition
File (E) Edit (E) View () Tools () Control (C) Help (H)
Send

Serial Port Settings

Port: COM3 (USB - SERIAL CH340)

Baud Rate: 400000

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

Reception Settings

○ ASCII ● Hex

Factory Automatic Line Break

Factory Display Send

Factory Display Time

Send Settings

○ ASCII ● Hex

Factory Automatic Reserid 0600

```
01 00 02 06 23 00 03 01 00 02 06 32 02 06 03 01 00 02 07 34
05 05 03 01 00 07 30 2C 6A 10 00 40 02 05 03 01 00 09 31 43
6F 6F 6C 4C 45 44 58 03 01 00 02 06 32 02 05 03
```

Reply with the parameter information of this device

```
0100020623020503
```

Power On Command

```
0100020623020503
```

COM3 OPENED, 8, NONE, 1, OFF | Rx: 56 Bytes | Tx: 8 Bytes

Sample instruction (hexadecimal format) :

Turn on the screen: 0100020605020503

Screen off: 01000206050003

4. Interaction Protocol

For the interaction protocol, please refer to the documentation "Interaction Protocol for Secondary Development of Full-Color Display Screens"

5. Sample Program

The sample program is licensed under the MIT Open Source LICENSE. See the License file in the project for details.

The sample programs related to this project are uniformly hosted in the following code repositories:

https://github.com/jotupix/demo

5.1. Feature Overview

The sample program provides for demonstrating how to control and modify content on a flexible display through an interactive protocol.

The sample program is primarily used as a reference implementation to help developers quickly understand the communication process, data organization, and control logic implementation, and to provide a basic code reference for secondary development, enabling the rapid construction of a runnable project.

The sample program mainly has the following functions:

- Data transformation and parsing
- Data subcontracting sent
- Data compression processing
- Program data construction
- Basic instructions and program sending
- Provide a basic operation interface for testing and validating control functions
- Scan and connect the flexible screen device

5.2. Basic Usage

5.2.1. Code acquisition

Clone the code repository using the GIT tool

```
    git clone https://github.com/jotupix/demo.git
```

When the clone is completed, the code repository will contain directories of iOS, Android, Windows, MCU and related sample code.

5.2.2. Directory Structure description

The directory structure of the sample program in the code repository is as follows:

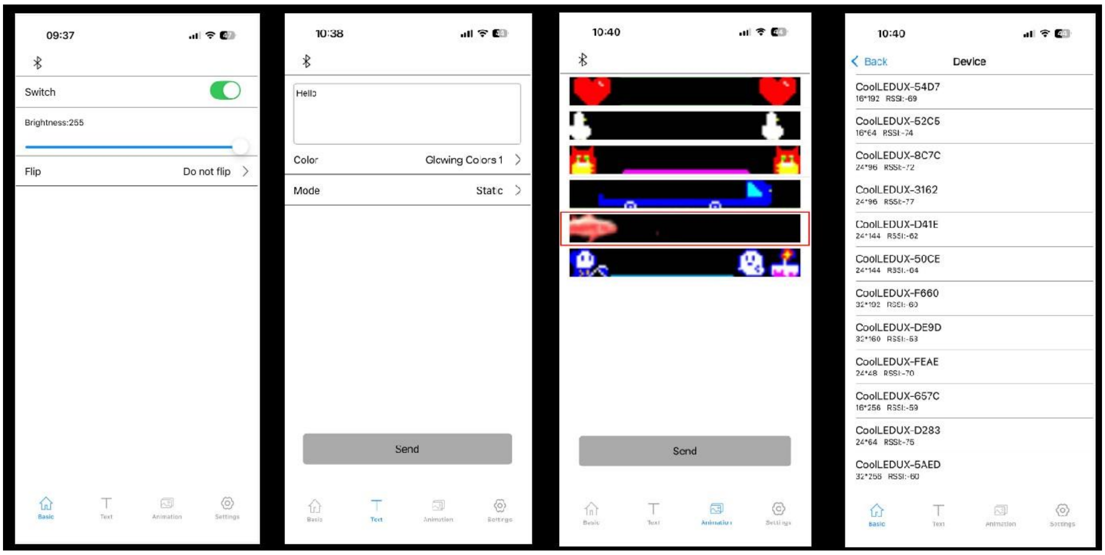
```
    JotuPix/
    ├── doc/ # documentation
    ├── ios/ # iOS sample program
    ├── android/ # Android sample program
    ├── windows/ # windows Sample program
    └ -- mcu/ # mcu examples such as stm32, esp32
```

5.3. Compile and run

For details, see the "README" of the sample program in each platform directory.

5.4. Sample Program Instructions

5.4.1. IOS & Android



Switch

Brightness:255

Flip Do not flip

Hello

Color Glowing Colors 1

Mode Static

CoolLEDUX-54D7 16*192 RSSI:-69

CoolLEDUX-52C5 16*64 RSSI:-74

CoolLEDUX-8C7C 24*96 RSSI:-72

CoolLEDUX-3162 24*96 RSSI:-77

CoolLEDUX-D41E 24*144 RSSI:-62

CoolLEDUX-50CE 24*144 RSSI:-64

CoolLEDUX-F660 32*192 RSSI:-60

CoolLEDUX-DE9D 32*160 RSSI:-53

CoolLEDUX-FEAE 24*48 RSSI:-70

CoolLEDUX-657C 16*256 RSSI:-59

CoolLEDUX-D283 24*64 RSSI:-75

CoolLEDUX-5AED 32*256 RSSI:-60

- Basic Features Page

This page mainly provides examples of some basic functions such as: turning devices on and off, adjusting device brightness, setting device flip status.

- Text Features Page

This page provides basic Settings for displaying text on the screen. You can type in any text and send it to the device for display. And offer the ability to select text color and display mode.

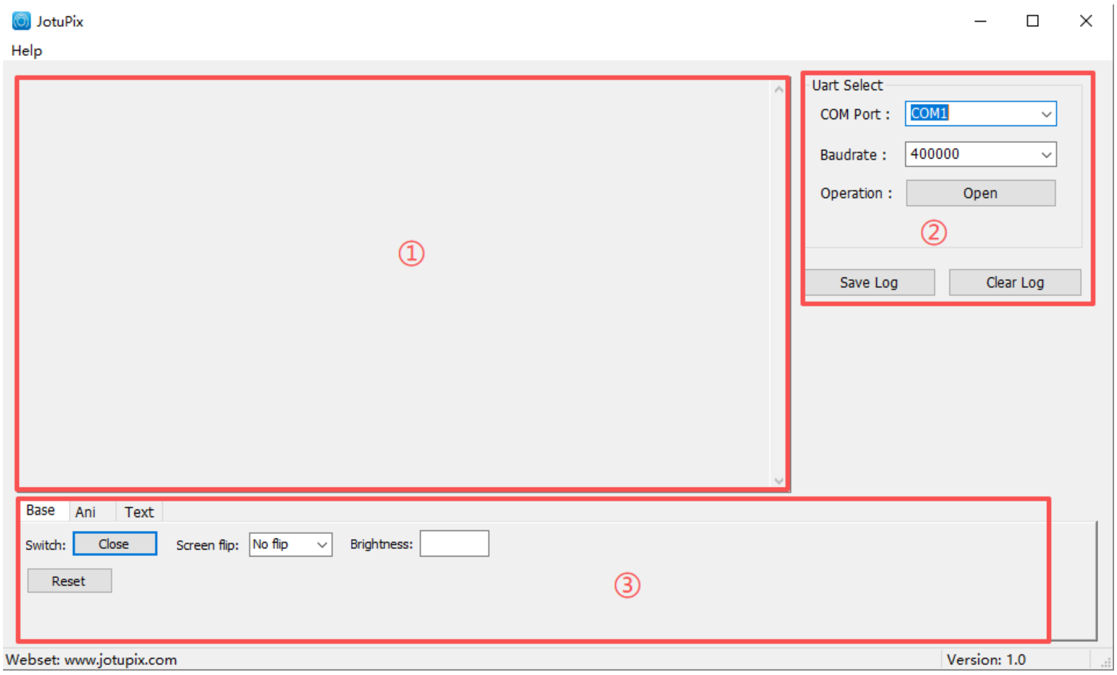
- Animation Function page

This page provides examples of the feature to display GIF animations, with several built-in GIF sample animations that can be sent to the device for display.

- Device List page

This page can search for display devices using BLE and make device connections.

5.4.2. Windows Instructions



COM Port : COM1

Baudrate : 400000

Switch: Close

Screen flip: No flip

Brightness:

Webset: www.jotupix.com

Version: 1.0

We provide Windows sample programs that communicate with devices via serial port and control device functions.

The Windows sample program interface is mainly divided into three parts: the log output area, the serial port configuration area, and the functionality area.

- Log Output Area

This area provides program log output functionality. Runtime data and error prompts are output to this area, and the data from this section can be used for comparison during development.

- Serial port configuration area

This area allows you to select serial numbers, configure serial baud rates, and connect and disconnect serial ports.

- Ribbon

  This area provides examples of various features of the display device, such as: turning on/off, adjusting brightness, configuring flip, sending GIF animations, sending text, etc.

5.4.3. MCU Description

We provide an STM32 host computer sample program to operate and control the device via serial port.

The following features are provided: doodle, animation, border, glitter text, custom color text, GIF animation demonstration.

Each demonstration function is automatically switched every 5 seconds.

5.5. Notes

The IOS & Android sample program relies on BLE to communicate with the device.

Windows & MCU sample programs rely on serial ports to communicate with devices.

The sample program is just a demonstration of the relevant functions of our display screen. If you need to use the actual project, please modify the open-source code yourself.

5.6. Update Instructions

We will update the sample program from time to time for new feature demonstrations. Please keep an eye on our code repository address or website.

Website: www.jotupix.com

Warehouse address: https://github.com/jotupix/demo

6. Frequently Asked questions for troubleshooting

7. Issue feedback and suggestions

If you have any comments, suggestions or need assistance with our products, please contact us by email.

Technical support: jotupix@gmail.com

If you have any needs for product purchase or customized development, please contact us by email.

Sales: jotupix@gmail.com