

■ 6 Analog Pin

A0 to A5 are analog inputs. A0 to A4 are located on ADC2, and A5 is located on ADC1.

■ Digital Pin

D5-D6, D9-D13 - These are digital pins. D5, D6, D9, and D10 are on ADC1. D11-D13 is on ADC2

■ SPI Pin

The SPI pin is located on the ESP32-S3 high-speed peripheral. You can set any pin to a low-speed peripheral, but you won't get a fast interface!

SCK - This is the SPI clock pin.

MOSI - This is the SPI microcontroller output/sensor input pin.

MISO - This is the SPI microcontroller input/sensor output pin.

■ UART Interface

RX - This is the UART receiving pin. TX (Transmission) pins attached to the sensor or splitter.

TX - This is the UART send pin. RX (receive) pins attached to the sensor or splitter.

■ I2C Interface

I2C interface. This is shared by the STEMMA QT connector.

SCL - This is the I2C clock pin. There is a 5k pull-up resistor on the pin.

SDA - This is the I2C data pin. There is a 5k pull-up resistor on the pin.

In CircuitPython, you can use the STEMMA connector with `board.SCL` and `board.SDA`, or with `board.STEMMA_I2C()`.

There is an I2C power supply pin that needs to be pulled high for the STEMMA QT connector, LC709203 and BME280 sensors (if available) to function properly.

CircuitPython and Arduino perform this operation automatically. It uses `TFT_I2C_POWER` in CircuitPython and Arduino.

RGB and LED

■ RGB: The RGB light refers to pin 33, which is marked 33 on the board. Used as a status indicator in the CircuitPython boot loader. Control by code. It is `board.NEOPIXEL` in CircuitPython and `PIN_NEOPIXEL` in Arduino.

■ Red LED: A red LED, labeled #13 on the board, lights up or flashes during certain operations (such as emitting a pulse in a boot loader) and can be controlled by code. It is `board.LED` in CircuitPython and `LED_BUILTIN` or 13 in Arduino.

SH1.0-4P Connector

The SH1.0 4P interface is used to connect Qwiic sensors or devices.

Button

■ RST Button: This button restarts the development board and helps enter the boot loader. You can reset the development board with one click without unplugging the USB cable or battery. Click once, and then click again when the NeoPixel state LED is purple to enter the UF2 boot loader (when CircuitPython needs to be loaded).

■ RST: Pins can be used to reset the board. Manually ground to reset the board.

■ BOOT Button: This button can be used as input or to put the board into ROM boot loader mode. It is available in CircuitPython as `board.BUTTON` and in Arduino as 0. To enter ROM BOOT loader mode, hold down the Boot button while pressing the RST button above

Debug

This is the Debug TX (DBG) pin. This is the hardware UART debug pin. You can connect it to a USB console cable in order to read the debug output of the ESP32 IDF. You can also read debugging information from the boot loader. If you are writing software and need to see low-level debugging output. This is very useful.

Three Power Consumption Modes

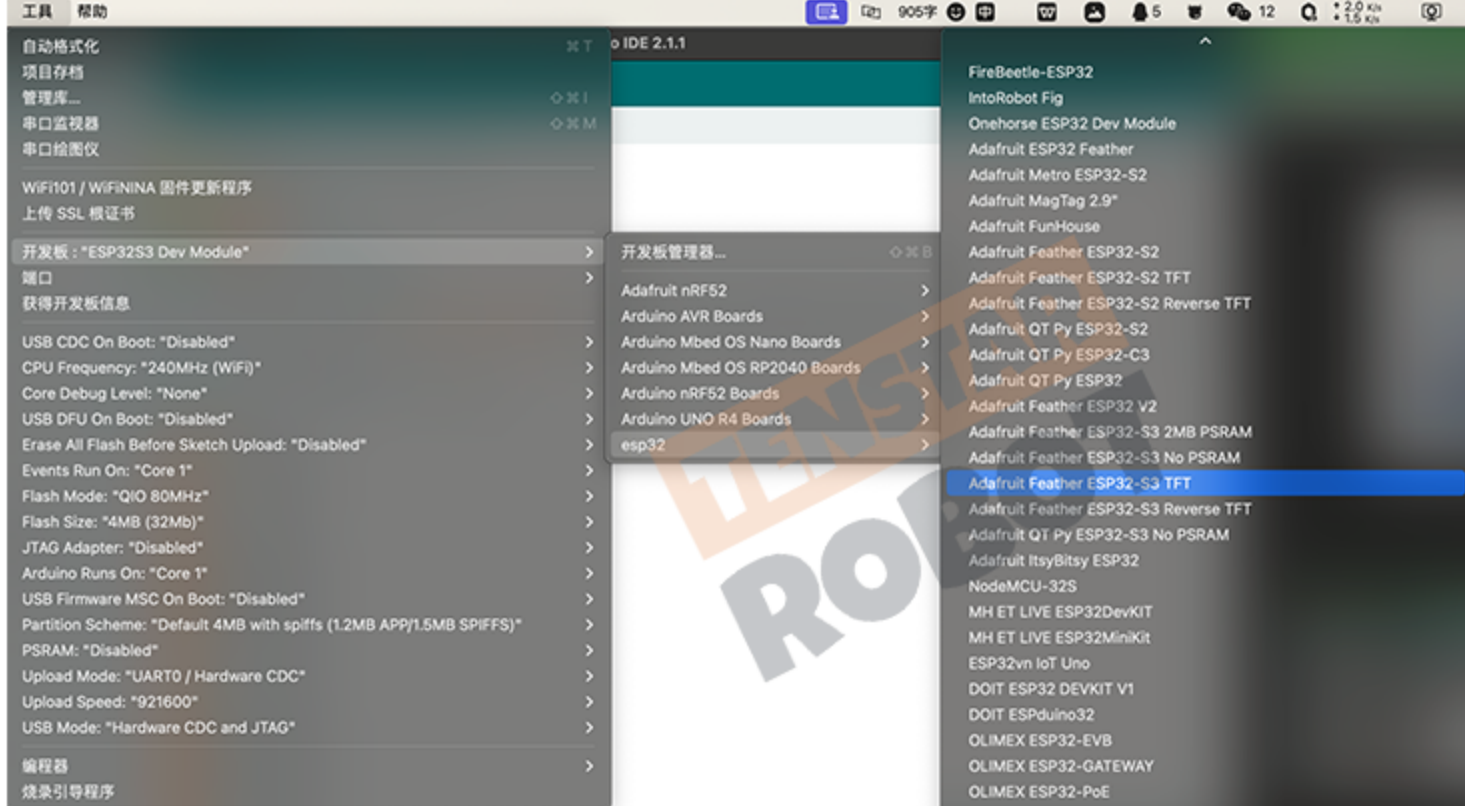
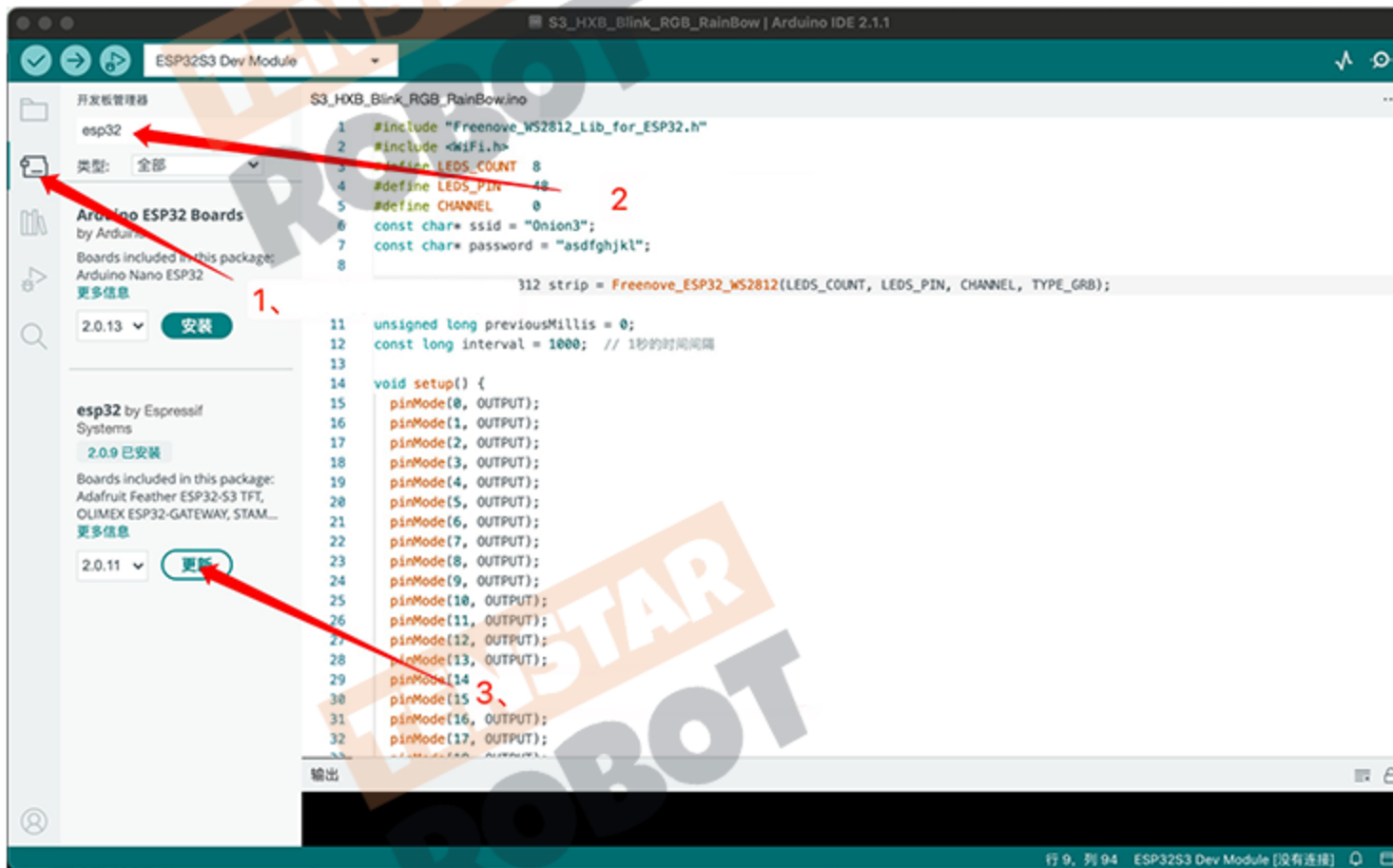
Due to the multiple sleep modes of the ESP32, the development board can be used for low power consumption. The ESP32-S3 chip has three basic operating states:

Normal, Light sleep and Deep sleep.

For Arduino

Connect the board to the computer

(hold down the BOOT button while powering on)



Note

- After you upload your code from the Arduino IDE to the ESP32-S3, be sure to press the reset button! Otherwise the program does not run.
- The ESP32-S2/S3 boot loader does not support USB serial support for Windows 7 or 8. Please update to espressif supported version Win 10!

Port Identification

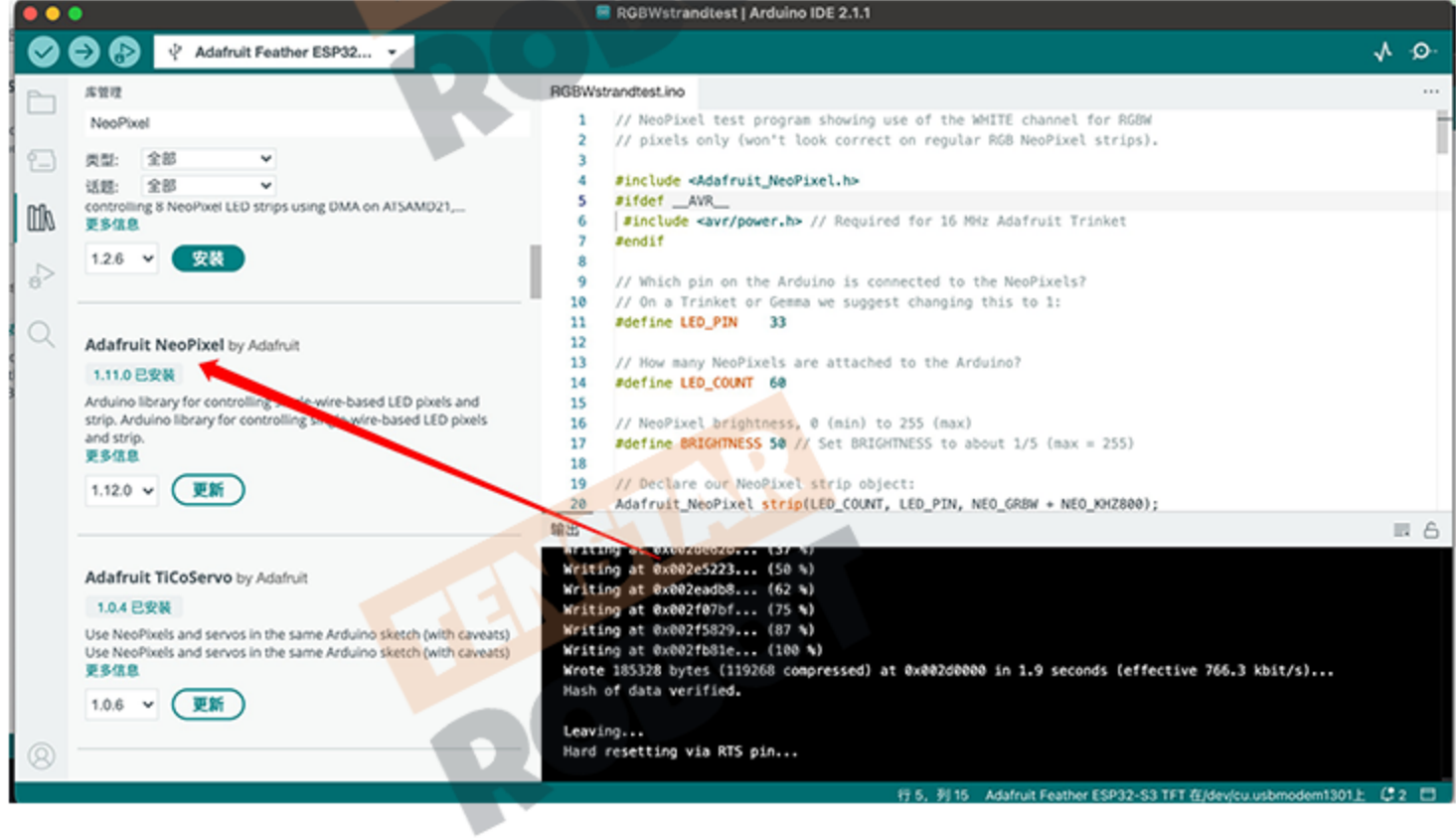
ESP32S3 built-in USB, but Arduino ESP32S3 series support is not friendly, directly inserted into the computer, you may not see the COM port, you can manually enter the boot loading mode, the boot loader is located in the ROM, is not modified, so you can use this method to enter the boot loader program at any time. After the Arduino uploads the code, you have to press the RST button to run the program.

1. Press and hold the Boot button to power on and enter the boot loading mode.

(That is, hold down the BOOT button before inserting the USB, and release the BOOT button after inserting the USB)

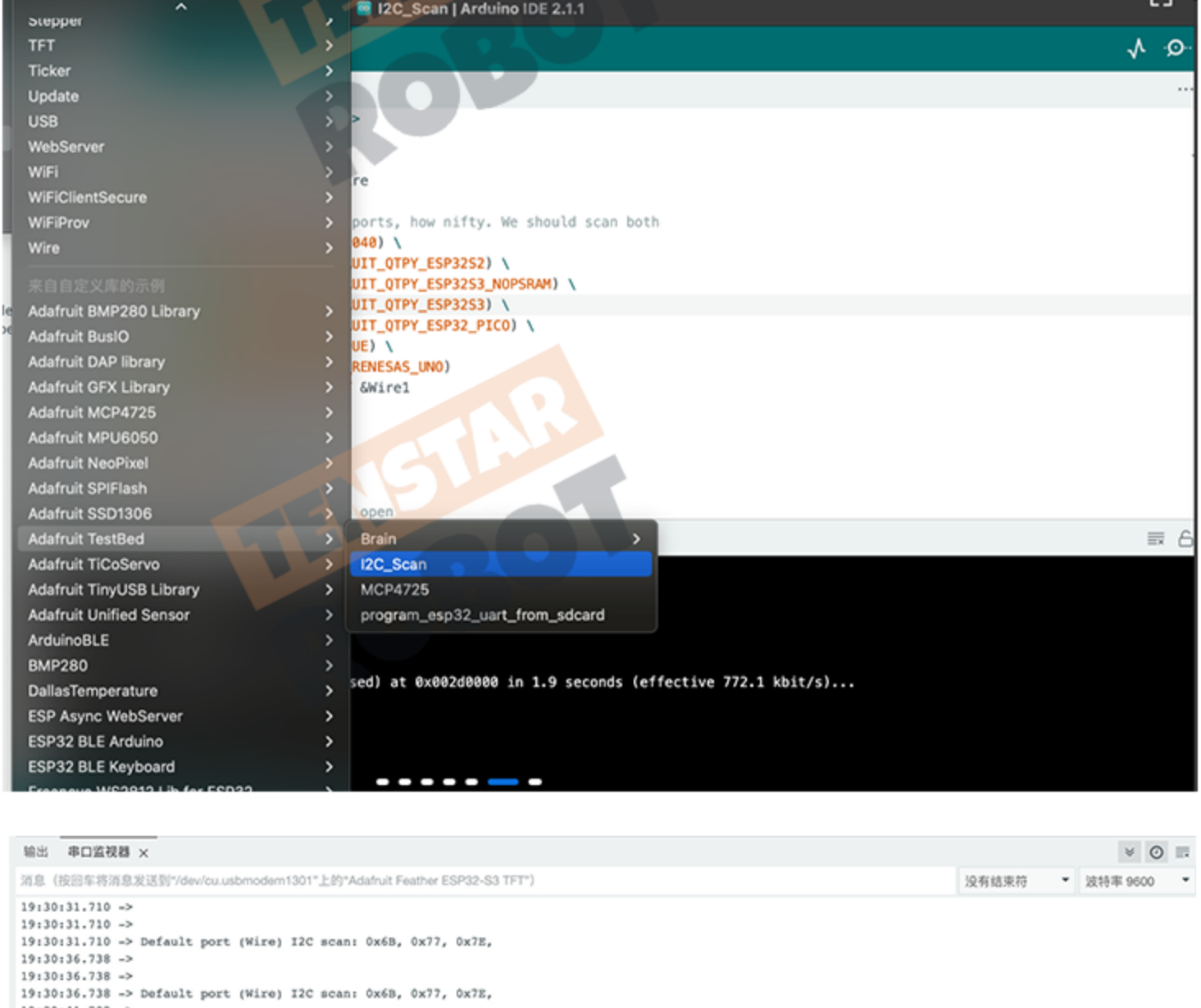
2, when the data cable has been plugged in, only need to long press Boot, and then press the RST button, you can enter the boot loading mode. (Press and hold the Boot until the RST button is released and the Boot button is released)

RGB



I2C Scan

When using the development board, you may need to connect an I2C device, which can be a little tricky the first time. The best way to debug I2C is to check the wiring and then perform an I2C scan



TFT Display

■ The front of the motherboard is a 240x135 pixel color 1.14" IPS TFT. It's a bright and colorful display with the ST7789 chip that you can view from any Angle.

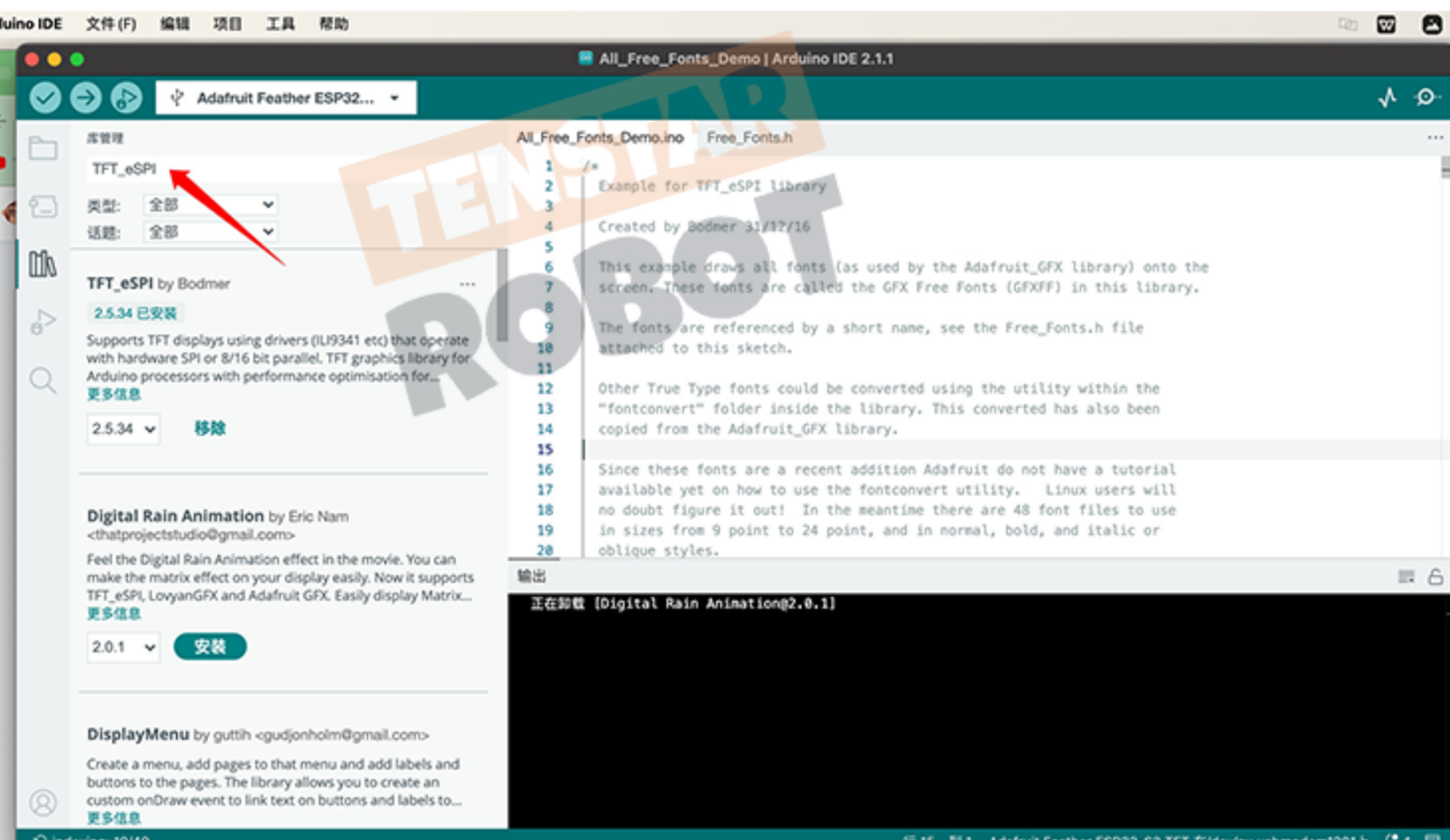
■ There is a power pin that must be pulled high for the display to work. This is done automatically by CircuitPython and Arduino. This pin is available in CircuitPython and Arduino in the format TFT_I2C_POWER

Note:

If you are having problems with I2C or TFT display on the Arduino, make sure you are using the latest Espressif board support pack. If you are still having problems, you may need to manually pull the pin up in your code.

Light the TFT display

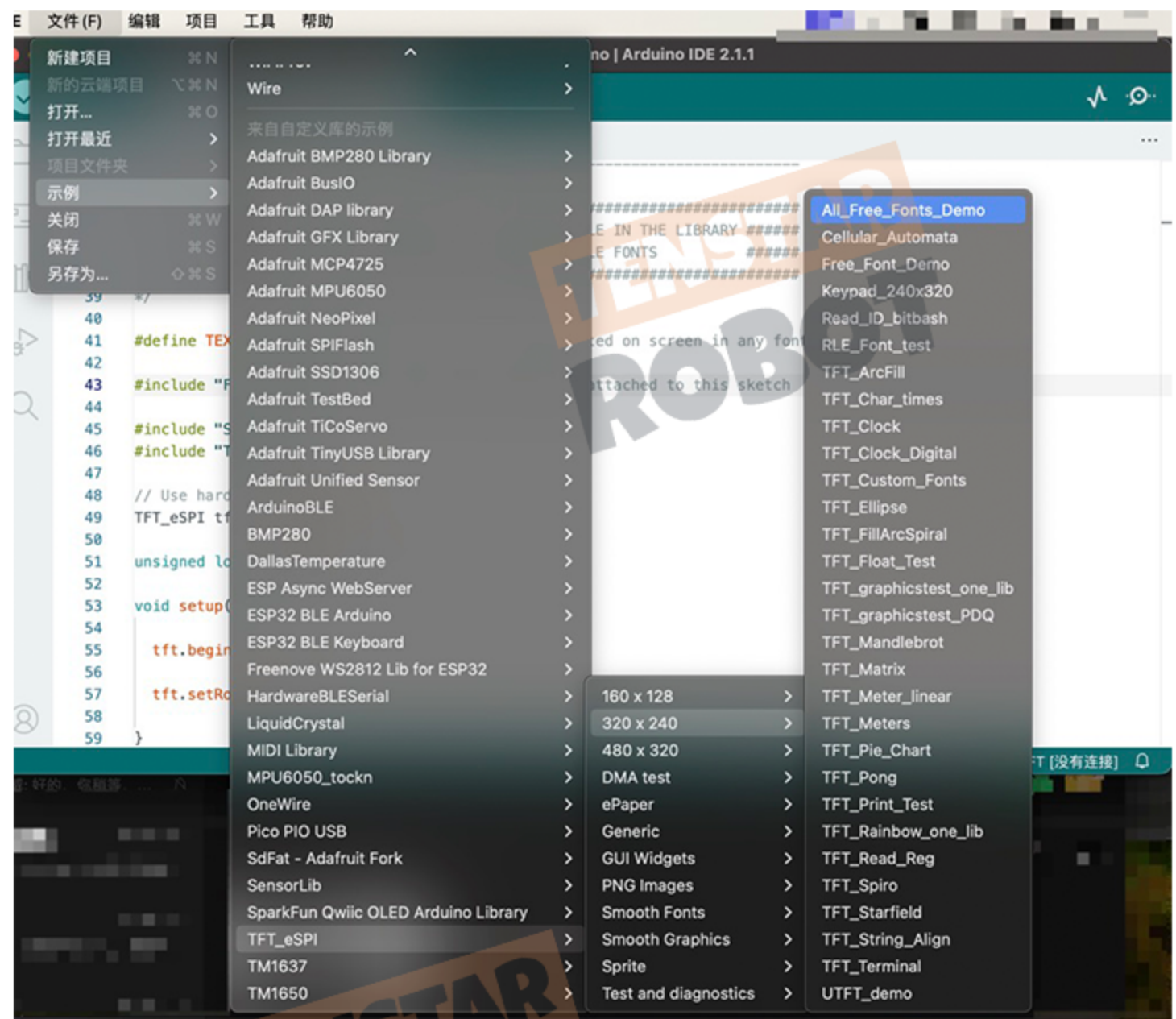
1. Install the TFT_eSPI library



2.Modify the TFT configuration file

```
#define ST7789_DRIVER
...
#define TFT_WIDTH 135
#define TFT_HEIGHT 240
...
#define TFT_BL 45
...
#define TFT_MISO 37
#define TFT_MOSI 35
#define TFT_SCLK 36
#define TFT_CS 7 // Chip select control pin (library pulls permanently low
#define TFT_DC 39// Data Command control pin - must use a pin in the range 0-31
#define TFT_RST 40 // Reset pin, toggles on startup
...
#define SMOOTH_FONT
```

3.Display content



NOTE

After downloading the program, you need to press the RST button to run the program.