SKU:DFR1063 (https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

Introduction

The Romeo mini robot controller is the ESP32 version of the Romeo BLE mini. It features an onboard ESP32-C3 module, supporting Wi-Fi and Bluetooth 5 dual-mode communication, and has two channels of 1.7A motor drivers.

Romeo mini has 9 IO interfaces (supporting analog, digital, UART, and I2C), with each interface compatible with the Gravity standard. It can be connected to various Gravity sensor modules. The servo interface can provide stable power supply through the servo power port for servos and other high-current devices. Additionally, it has a built-in GDI interface on the back for connecting an IPS display screen to show project-related information. The controller can meet the control requirements for robot motors and servos.

Features

- Main control + WiFi + Bluetooth 5 integrated motor driver development board
- Compact size
- Support for IPS TFT display screen
- Offers options for servo and motor control schemes

Specification

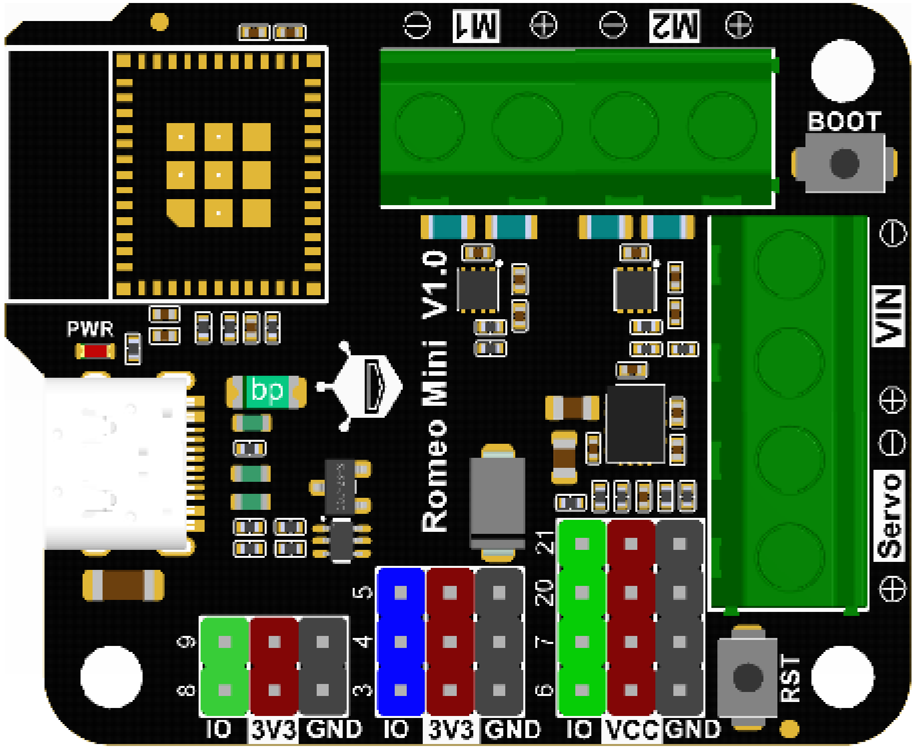
Basic Parameters:

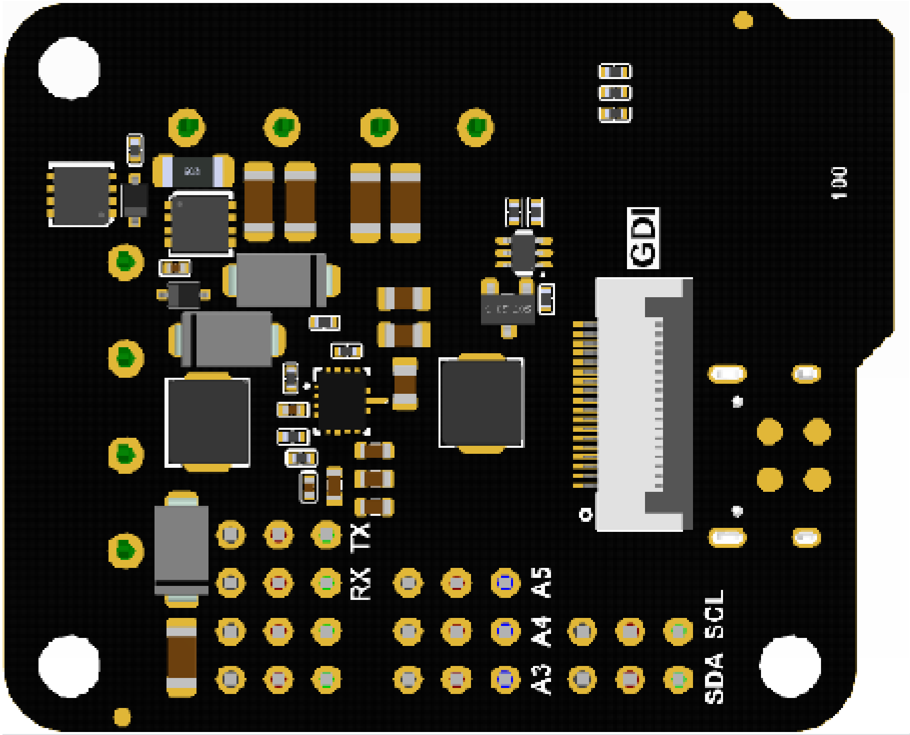
- Main control chip: ESP32-C3-MINI-1
- Type-C input voltage: 5V
- VIN input voltage: 5-15V
- Motor output current: 1.7A continuous drive current per channel
- Servo input voltage: 5-12V
- Module size: 47x38.5mm

Interface Pins:

- Digital I/O: x4
- Analog I/O: x3
- UART: x1
- I2C: x1
- Motor: x2
- GDI: x1
- Type-C: x1

Board Overview

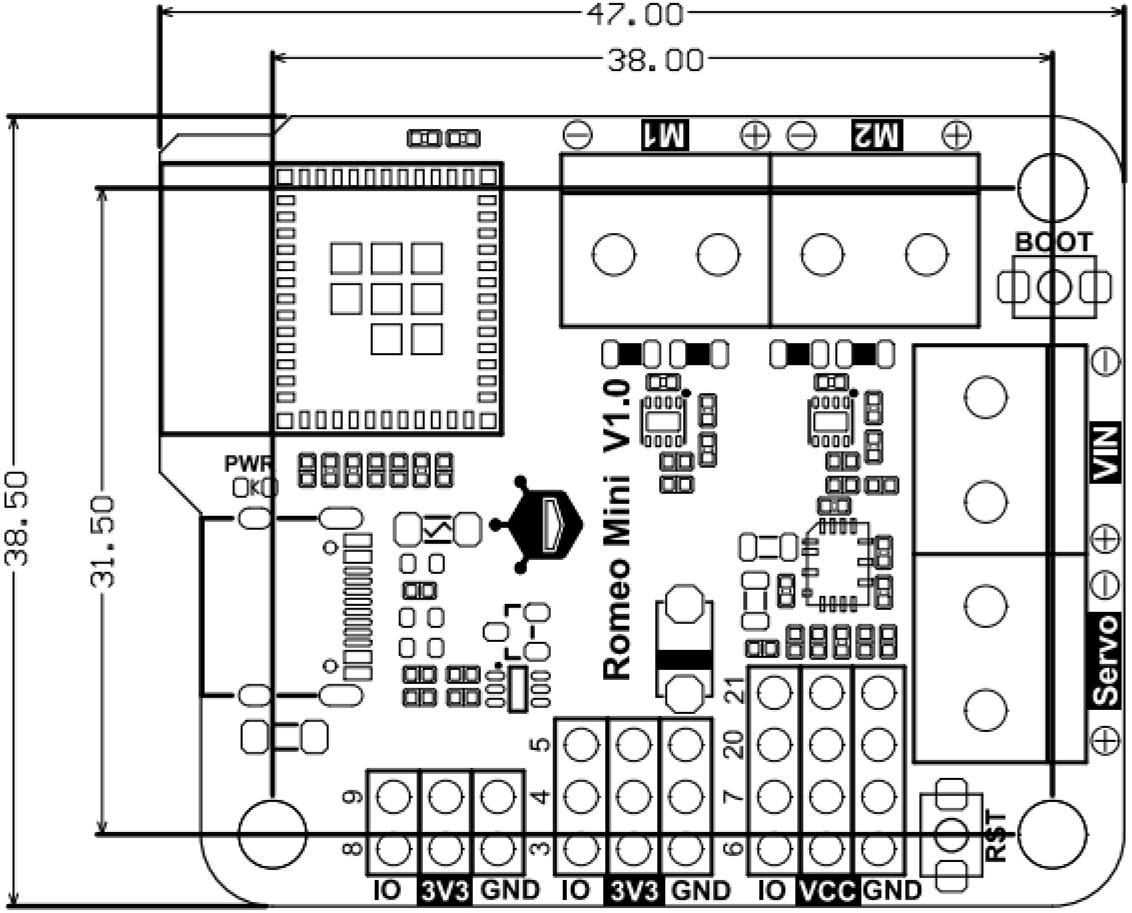




Name	Function Description
VIN	Power input for motor driver and controller: 5-15V. This port must be connected for M1/M2 motors to work.
Servo	Power input for digital ports: 5-12V. If using high-current peripherals with voltage higher than 5V on digital output port (VCC end), connect to this interface to avoid damaging the Type-C USB port.
M1/M2 Ports	Two motor control ports:
	M1: GPIO1 (direction control), GPIO0 (PWM control)
	M2: GPIO10 (direction control), GPIO2 (PWM control)

Name	Function Description
I2C	Standard I2C interface: SDA (8), SCL (9), VCC (3.3V), GND
A-3.3V-GND	Standard analog interface:
	A (blue): analog ports 3, 4, 5
	3.3V (red): power supply positive terminal
	GND (black): power supply negative terminal
D-VCC-GND	Standard digital interface:
	D (green): digital ports 6, 7, 20, 21
	VCC (red): power supply positive terminal (When the Servo port is not powered, VCC is 5V; when the Servo port is powered, VCC is the input power for the Servo port.)
	GND (black): power supply negative terminal
UART	RX (20), TX (21), VCC (5V), GND
	Note: The UART interface shares two IO ports with servo ports, so UART cannot be used when servos are used.
RST	Controller reset button
BOOT	Controller download button

Dimension Drawing



Tutorial

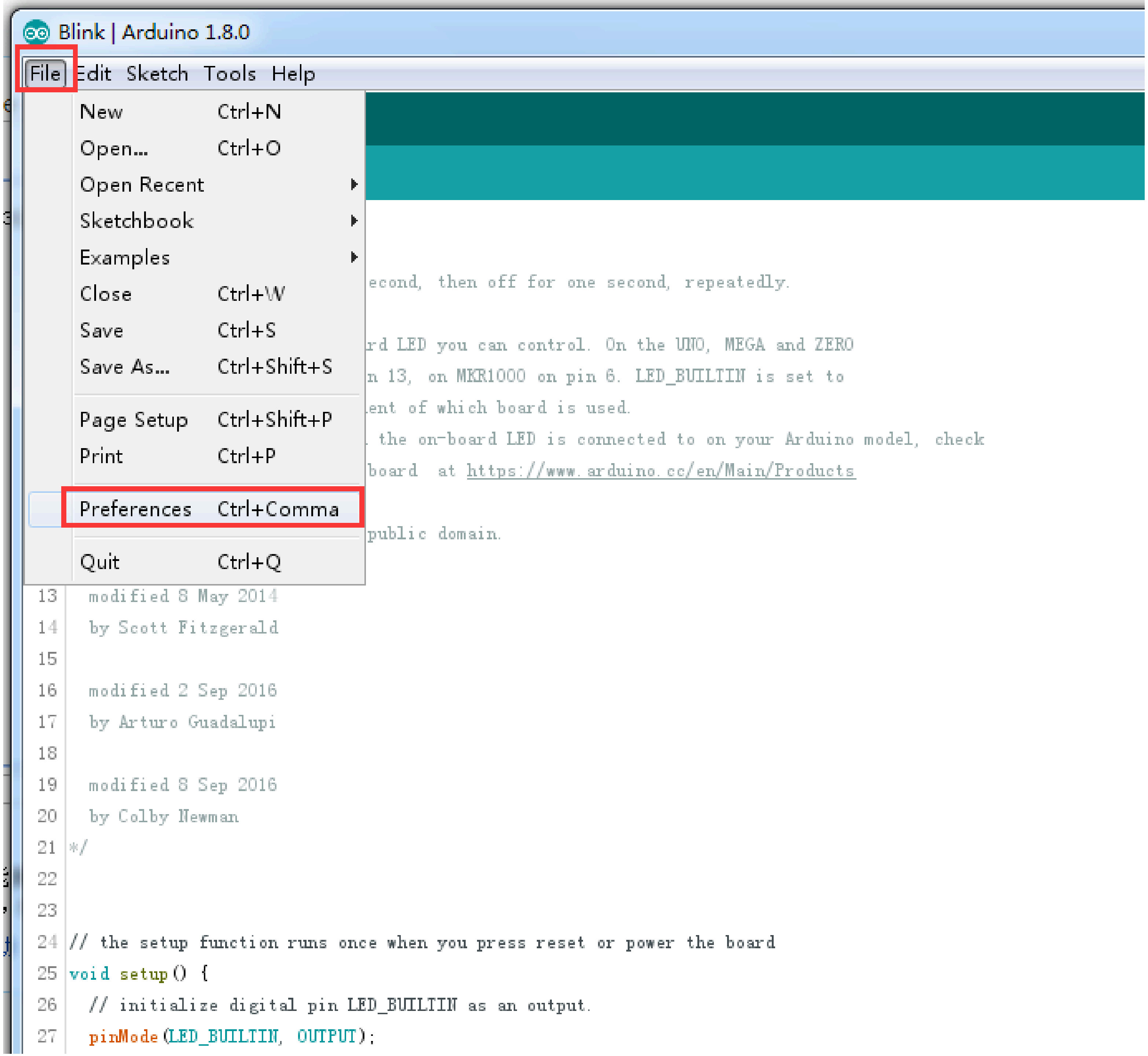
1. Arduino Environment Configuration

When using the ESP32-C3 development board for the first time, it is important to follow these steps:

Note: If your computer fails to recognize the Romeo Mini module when connected via USB, please press and hold the BOOT button, then press the RST button. Finally, release both the BOOT and RST buttons simultaneously.

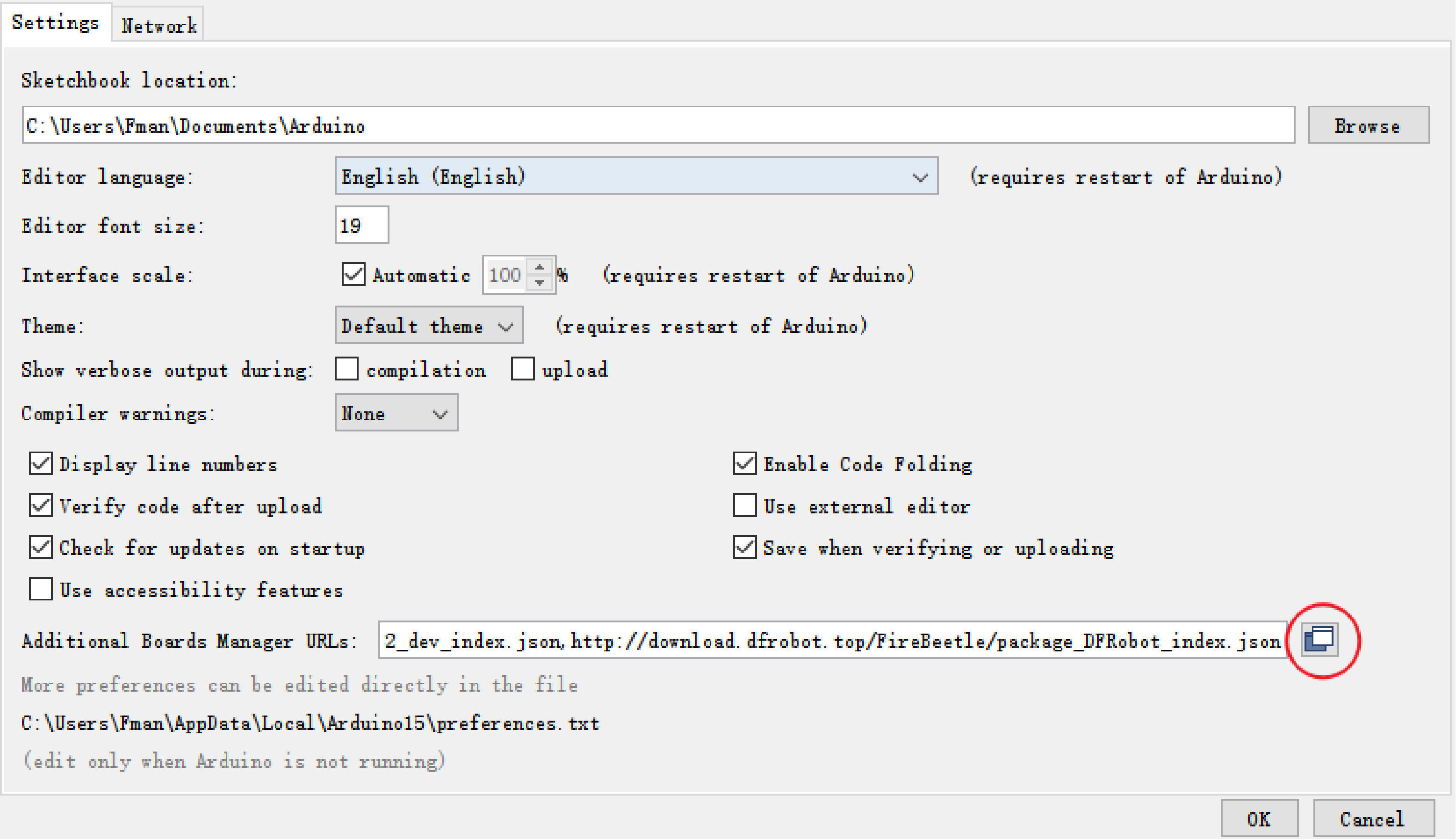
1.Add the JSON link to the IDE. 2.Download the core of the main controller. 3.Select the development board and serial port. 4.Open the sample program and upload it. 5.Familiarize yourself with the Serial Monitor.

- **Arduino IDE Compilation Environment Configuration**
- Configure the URL in the Arduino IDE.
- Open the Arduino IDE and click File -> Preferences, as shown in the figure below:



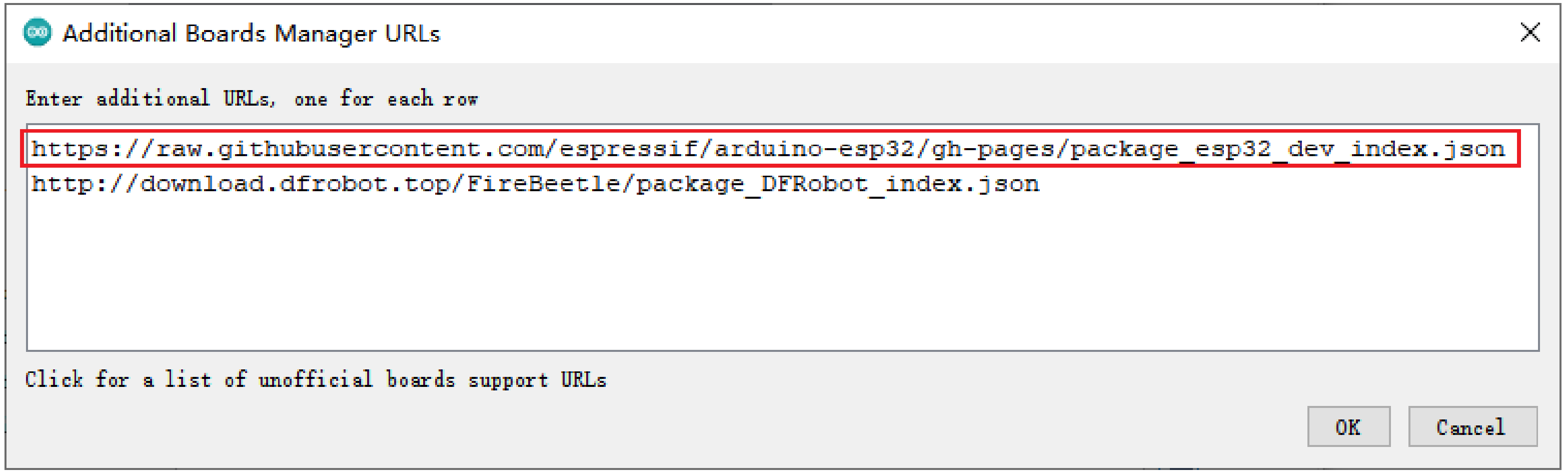
```
econd, then off for one second, repeatedly.

rd LED you can control. On the UNO, MEGA and ZERO
n 13, on MKR1000 on pin 6. LED_BUILTIN is set to
ent of which board is used.
 the on-board LED is connected to on your Arduino model, check
board  at https://www.arduino.cc/en/Main/Products

public domain.

  modified 8 May 2014
  by Scott Fitzgerald

  modified 2 Sep 2016
  by Arturo Guadalupi

  modified 8 Sep 2016
  by Colby Newman
*/


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

```

- In the new window, click the button indicated by the red circle in the figure below:



Sketchbook location:
C:\Users\Fman\Documents\Arduino
Editor language: English (English) (requires restart of Arduino)
Editor font size: 19
Interface scale: ☑ Automatic 100 % (requires restart of Arduino)
Theme: Default theme (requires restart of Arduino)
Show verbose output during: □ compilation □ upload
Compiler warnings: None
☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features
☑ Enable Code Folding
□ Use external editor
☑ Save when verifying or uploading
Additional Boards Manager URLs: 2_dev_index.json,http://download.dfrobot.top/FireBeetle/package_DFRobot_index.json
More preferences can be edited directly in the file
C:\Users\Fman\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

- Copy the following link address into the dialog box that appears: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json (https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)
- Note: If you have previously installed other environments, you can press Enter at the beginning or end of the previous link and paste the above link on any line above or below it.

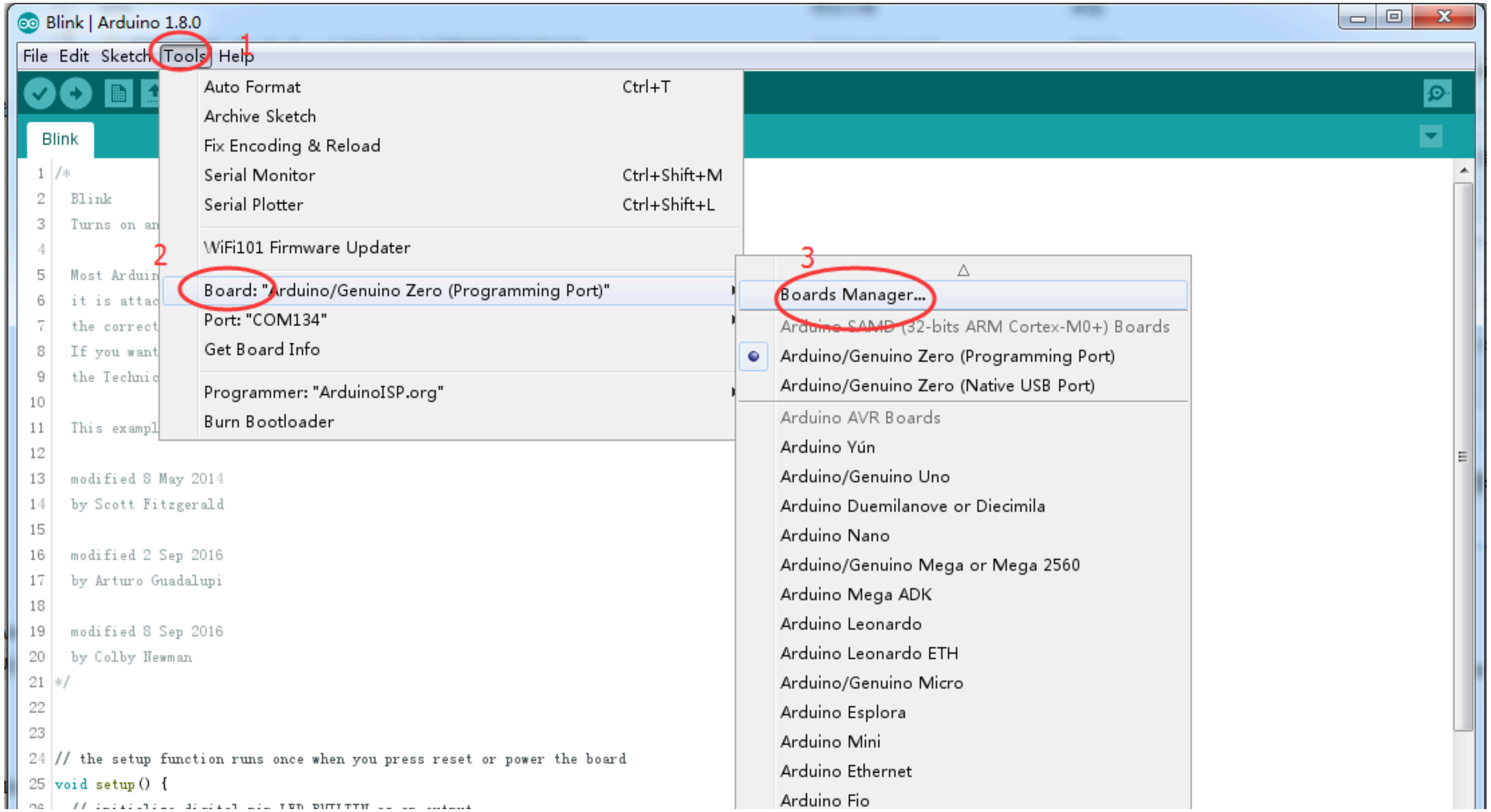


Enter additional URLs, one for each row

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json
http://download.dfrobot.top/FireBeetle/package_DFRobot_index.json
```

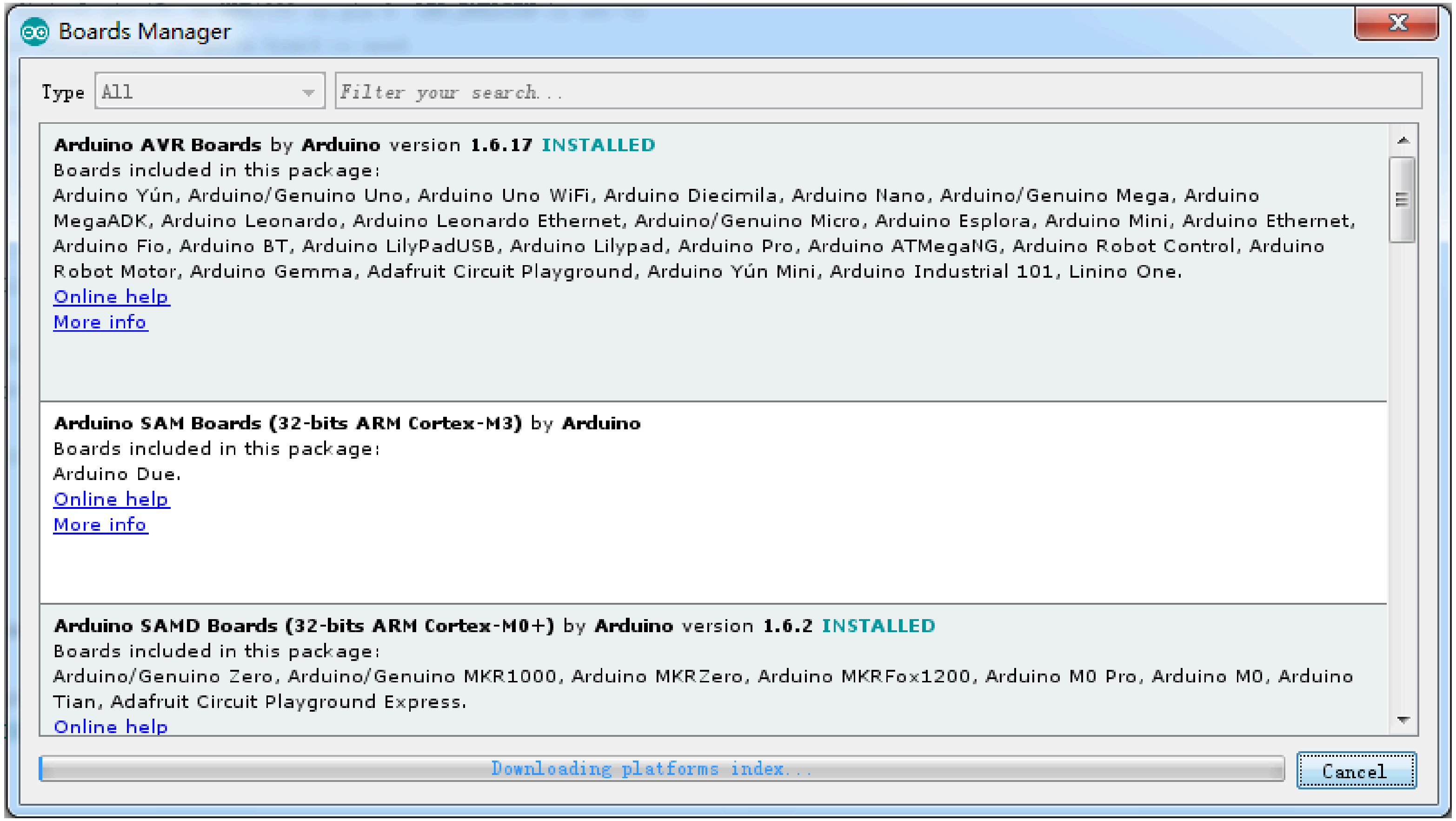
Click for a list of unofficial boards support URLs

- Click OK.
- Update the board.
- Open Tools -> Board -> Boards Manager..., as shown in the figure below:



```
/*
  Blink
  Turns on an

  Most Arduin
  it is attac
  the correct
  If you want
  the Technic

  This exampl

  modified 8 May 2014
  by Scott Fitzgerald

  modified 2 Sep 2016
  by Arturo Guadalupi

  modified 8 Sep 2016
  by Colby Newman
*/


// the setup function runs once when you press reset or power the board
void setup() {
```

- Boards Manager will automatically update the board, as shown in the figure below:



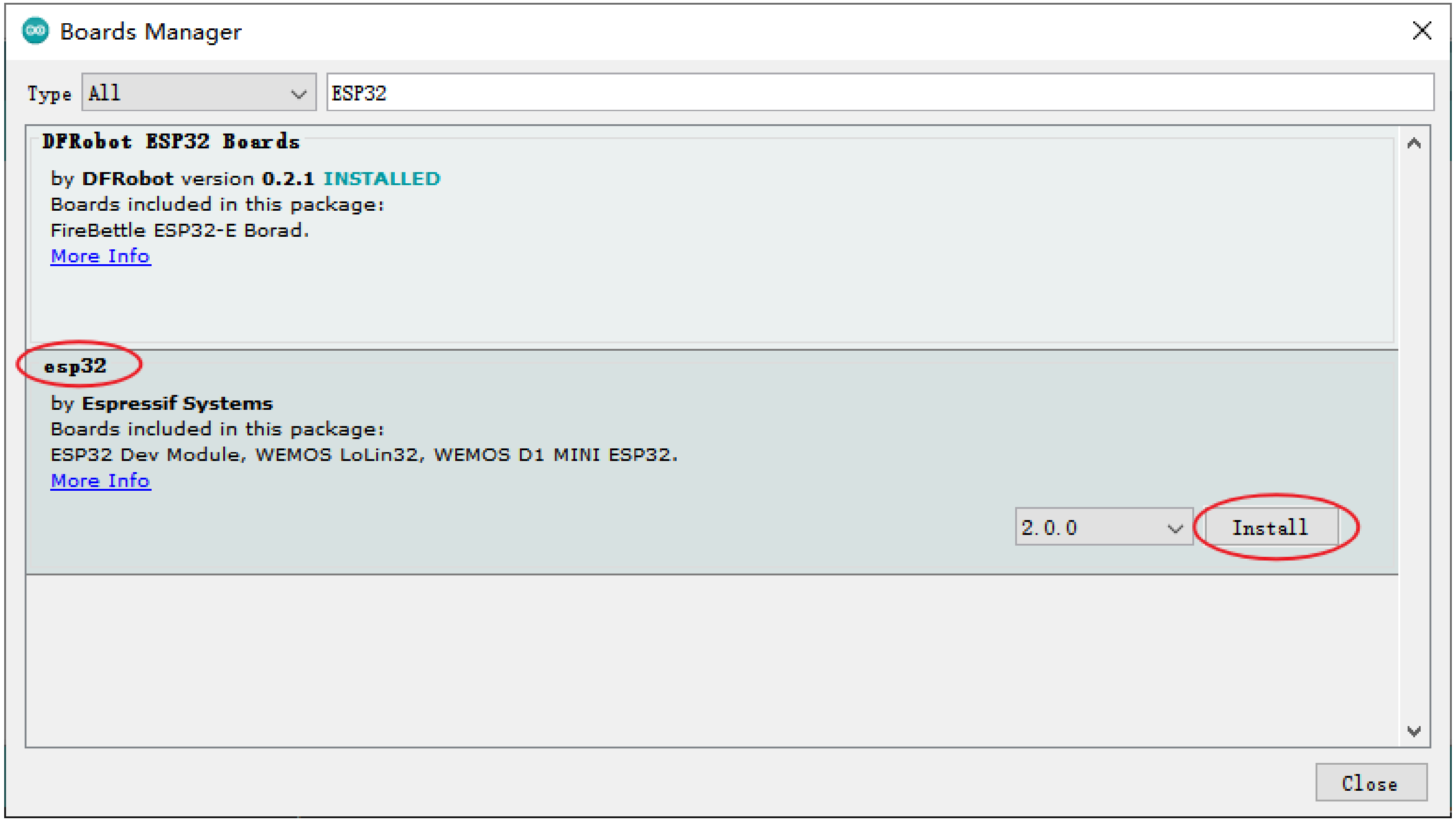
Arduino AVR Boards by **Arduino** version **1.6.17** INSTALLED
Boards included in this package:
Arduino Yún, Arduino/Genuino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino/Genuino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino/Genuino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.
Online help
More info

Arduino SAM Boards (32-bits ARM Cortex-M3) by **Arduino**
Boards included in this package:
Arduino Due.
Online help
More info

Arduino SAMD Boards (32-bits ARM Cortex-M0+) by **Arduino** version **1.6.2** INSTALLED
Boards included in this package:
Arduino/Genuino Zero, Arduino/Genuino MKR1000, Arduino MKRZero, Arduino MKRFox1200, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.
Online help

Downloading platforms index...

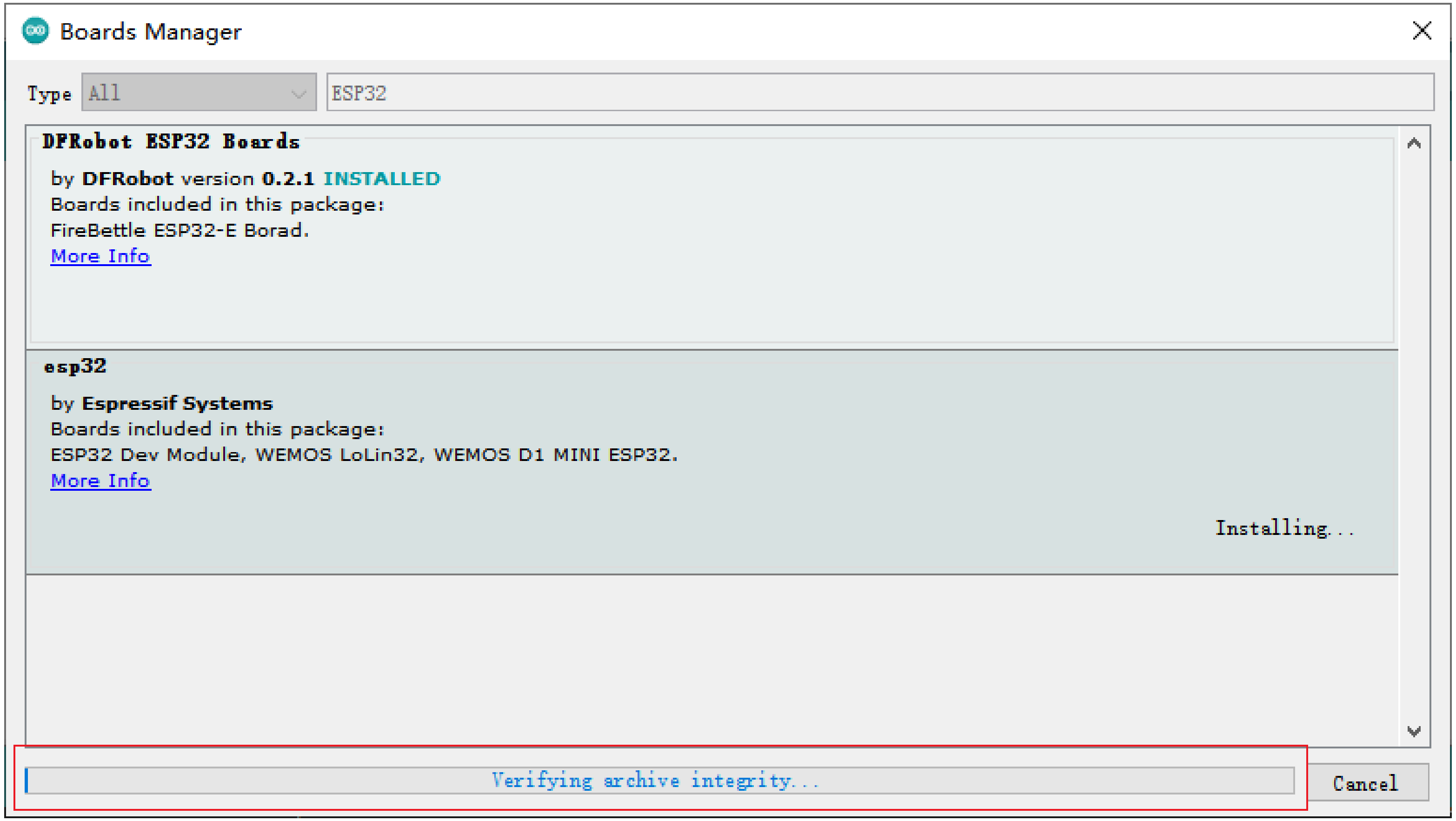
- After the update is completed, you can type "esp32" in the input box at the top. Then, select "esp32" when it appears and click Install (current version: 2.0.0):



DFRobot ESP32 Boards
by **DFRobot** version **0.2.1** INSTALLED
Boards included in this package:
FireBettle ESP32-E Borad.
More Info

esp32
by **Espressif Systems**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

- Wait for the progress bar to finish:



DFRobot ESP32 Boards

by **DFRobot** version **0.2.1** **INSTALLED**
Boards included in this package:
FireBettle ESP32-E Borad.
More Info

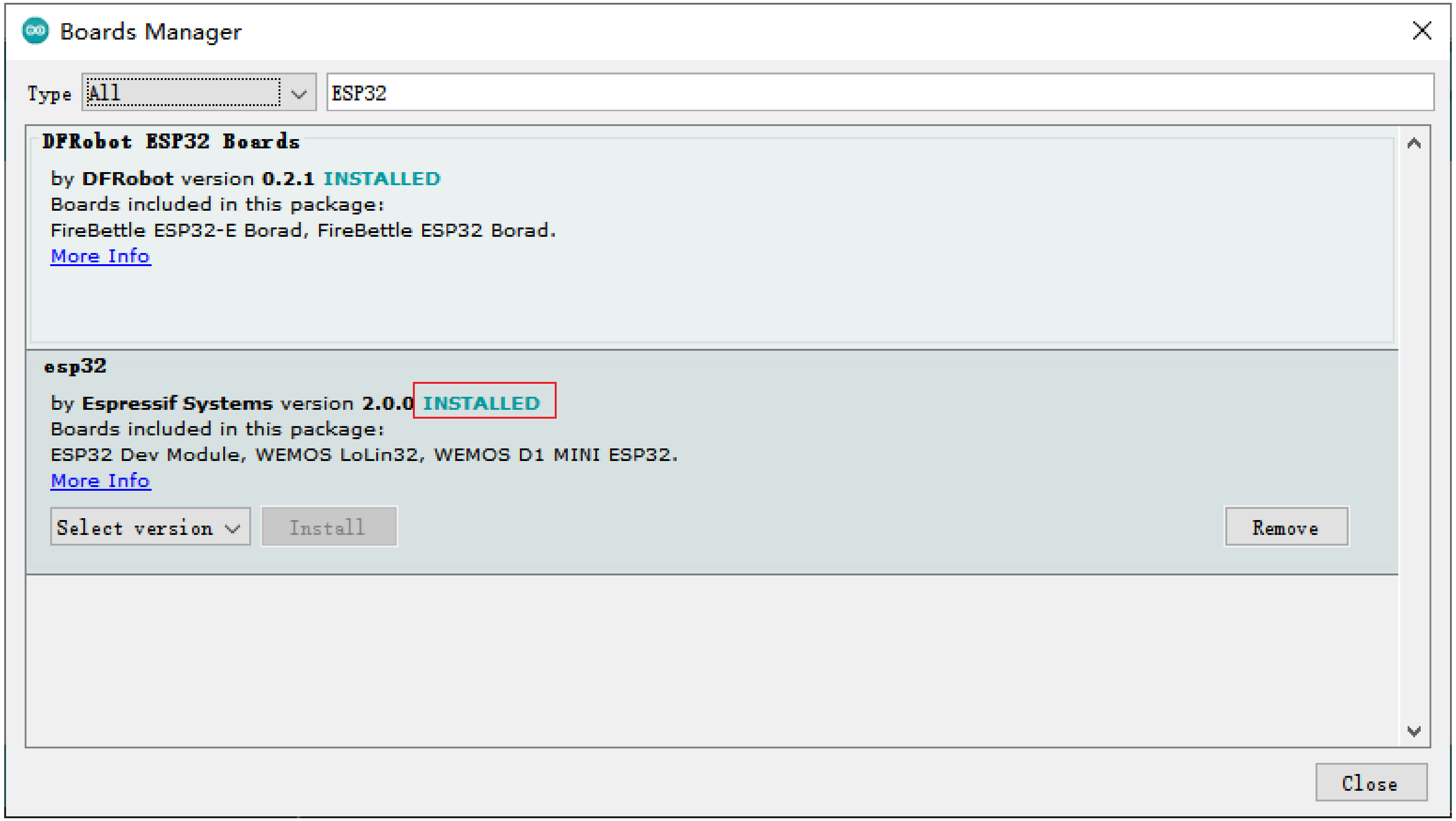
esp32

by **Espressif Systems**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

Installing...

Verifying archive integrity...

- After the installation is complete, the list will display the installed ESP32 board, as shown in the figure below:

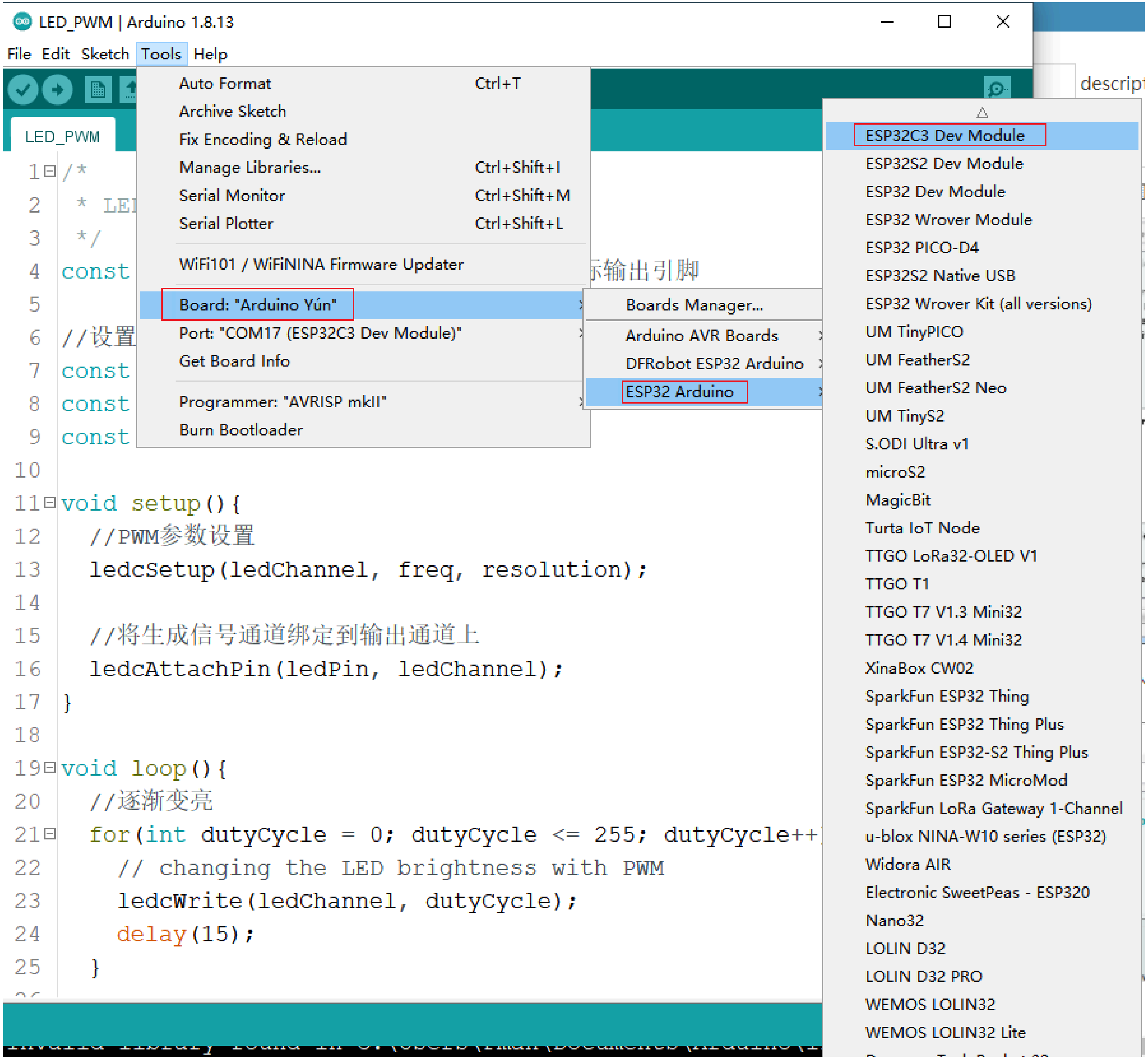


ESP32

DFRobot ESP32 Boards
by DFRobot version 0.2.1 INSTALLED
Boards included in this package:
FireBettle ESP32-E Borad, FireBettle ESP32 Borad.

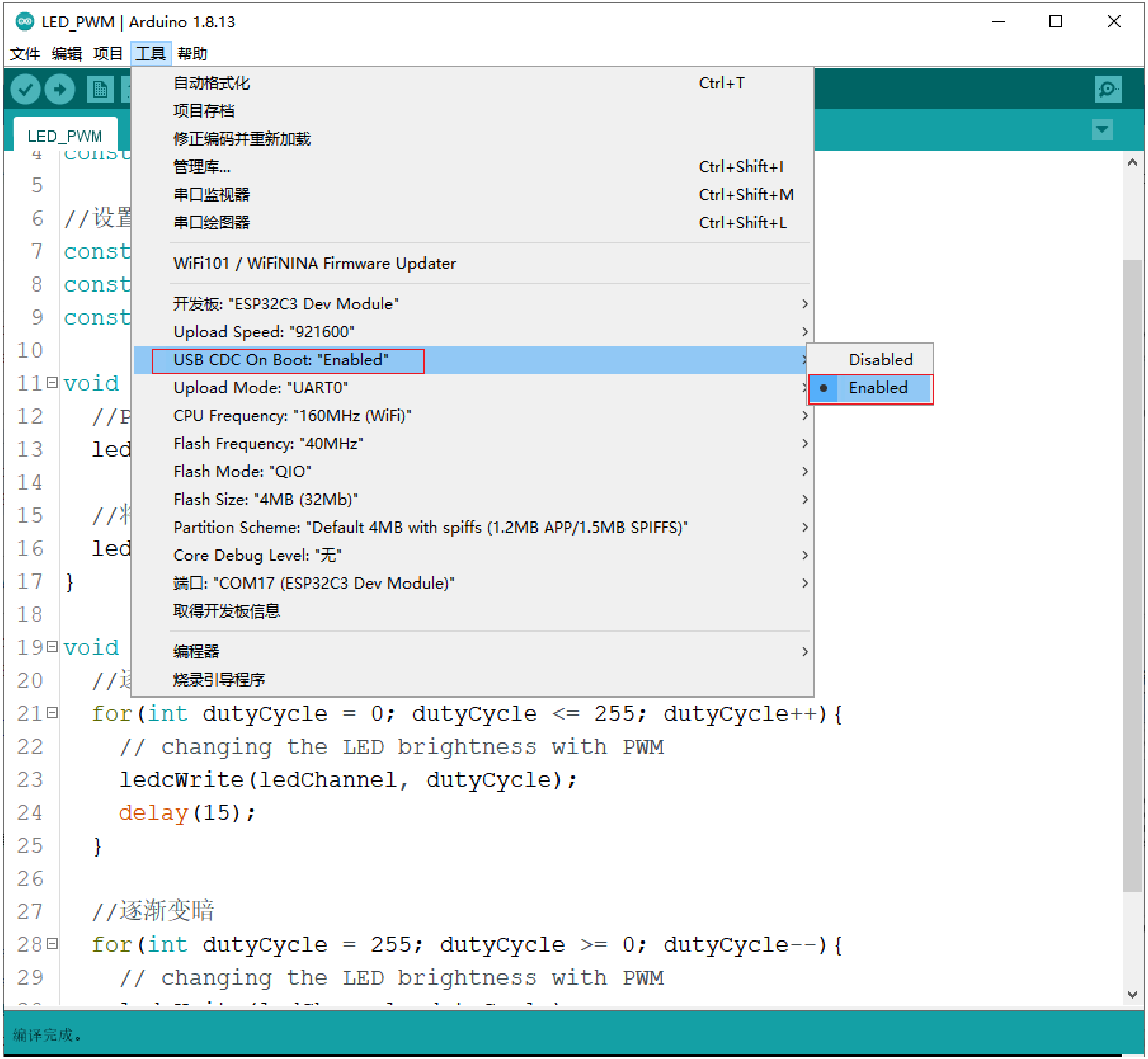
esp32
by Espressif Systems version 2.0.0 INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.

- Click Tools -> Board:, and select "ESP32C3 Dev Module" (usually the first one in the list).



```
/*
 * LEI
 */
const                                        示输出引脚

//设置
const
const
const

void setup(){
  //PWM参数设置
  ledcSetup(ledChannel, freq, resolution);

  //将生成信号通道绑定到输出通道上
  ledcAttachPin(ledPin, ledChannel);
}

void loop(){
  //逐渐变亮
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }
```

- Prior to getting started, you'll need to configure the following settings (if you wish to print to the Arduino monitor via USB, choose "Enable"):

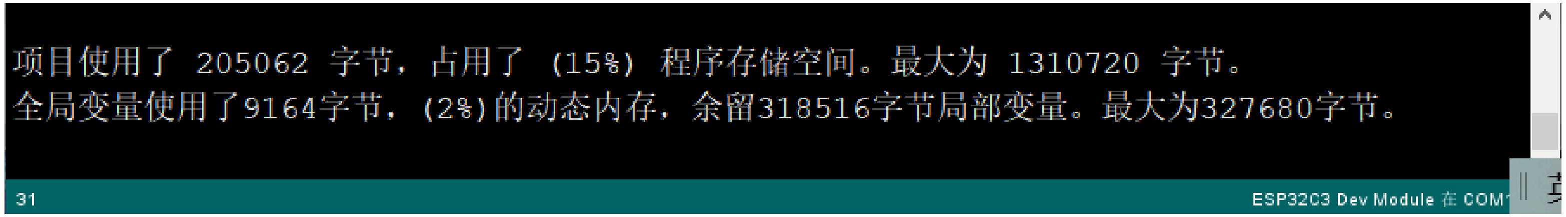


LED_PWM | Arduino 1.8.13

开发板: "ESP32C3 Dev Module"
Upload Speed: "921600"
USB CDC On Boot: "Enabled"
Upload Mode: "UART0"
CPU Frequency: "160MHz (WiFi)"
Flash Frequency: "40MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "无"
端口: "COM17 (ESP32C3 Dev Module)"

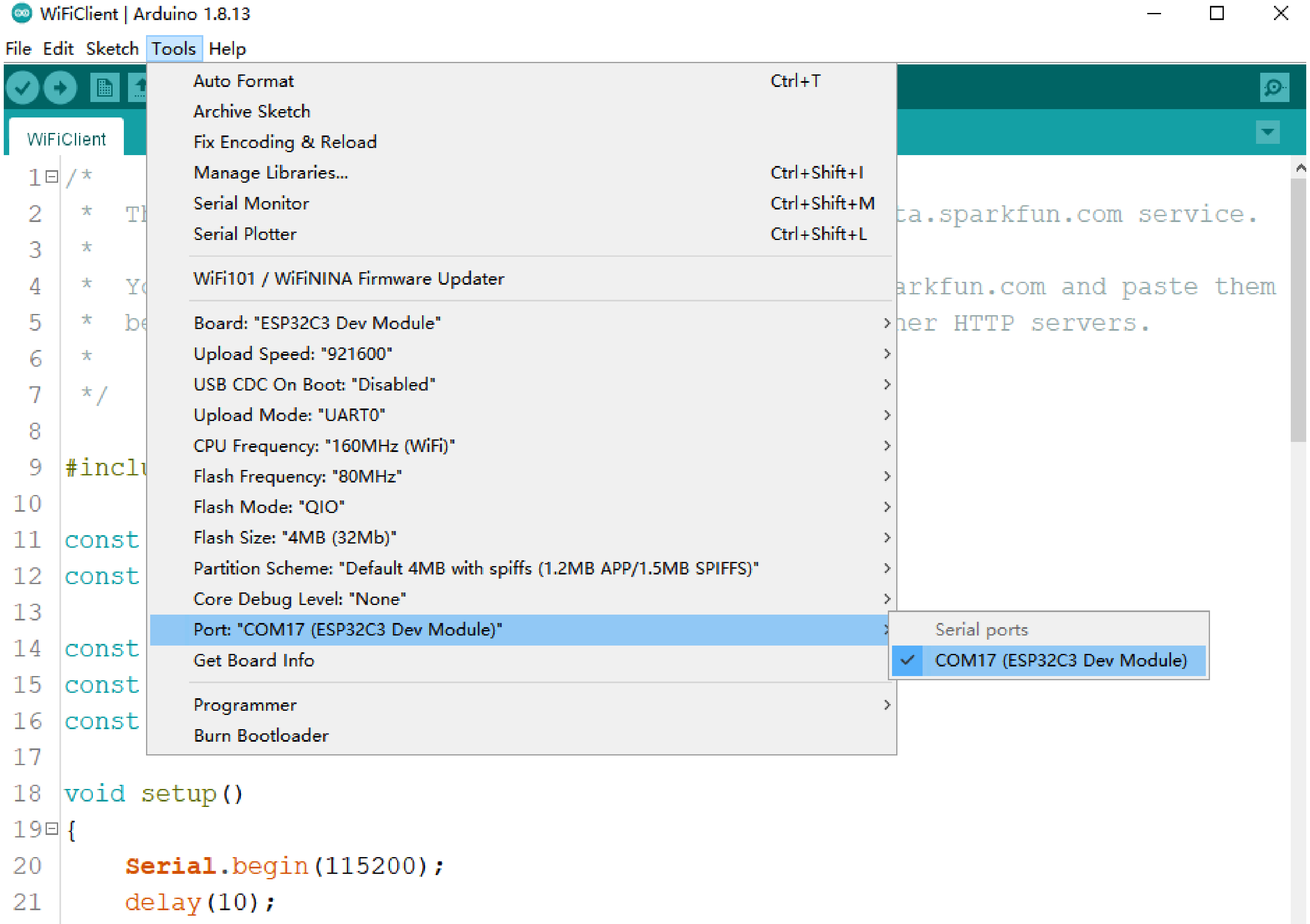
```
const

//设置
const
const
const

void
  //P
  led

  //将
  led
}

void
  //逐
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }

  //逐渐变暗
  for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
```

编译完成。



```
项目使用了 205062 字节，占用了 (15%) 程序存储空间。最大为 1310720 字节。
全局变量使用了9164字节，(2%)的动态内存，余留318516字节局部变量。最大为327680字节。
```

- Click on "Port" and select the corresponding serial port. (If the port keeps appearing and disappearing, you need to press and hold the BOOT button, then press the RST button, and finally release both the BOOT and RST buttons simultaneously).



WiFiClient | Arduino 1.8.13
File
Edit
Sketch
Tools
Help
WiFiClient
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 / WiFiNINA Firmware Updater
Get Board Info
Programmer
Burn Bootloader
Serial ports

Board: "ESP32C3 Dev Module"
Upload Speed: "921600"
USB CDC On Boot: "Disabled"
Upload Mode: "UART0"
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"
Port: "COM17 (ESP32C3 Dev Module)"
✓ COM17 (ESP32C3 Dev Module)

```
/*
 * Th                                                    ta.sparkfun.com service.
 *
 * Yo                                                    arkfun.com and paste them
 * be                                                    her HTTP servers.
 *
 */

#inclu

const
const

const
const
const

void setup()
{
    Serial.begin(115200);
    delay(10);
```

2. PWM-driven DC Motors

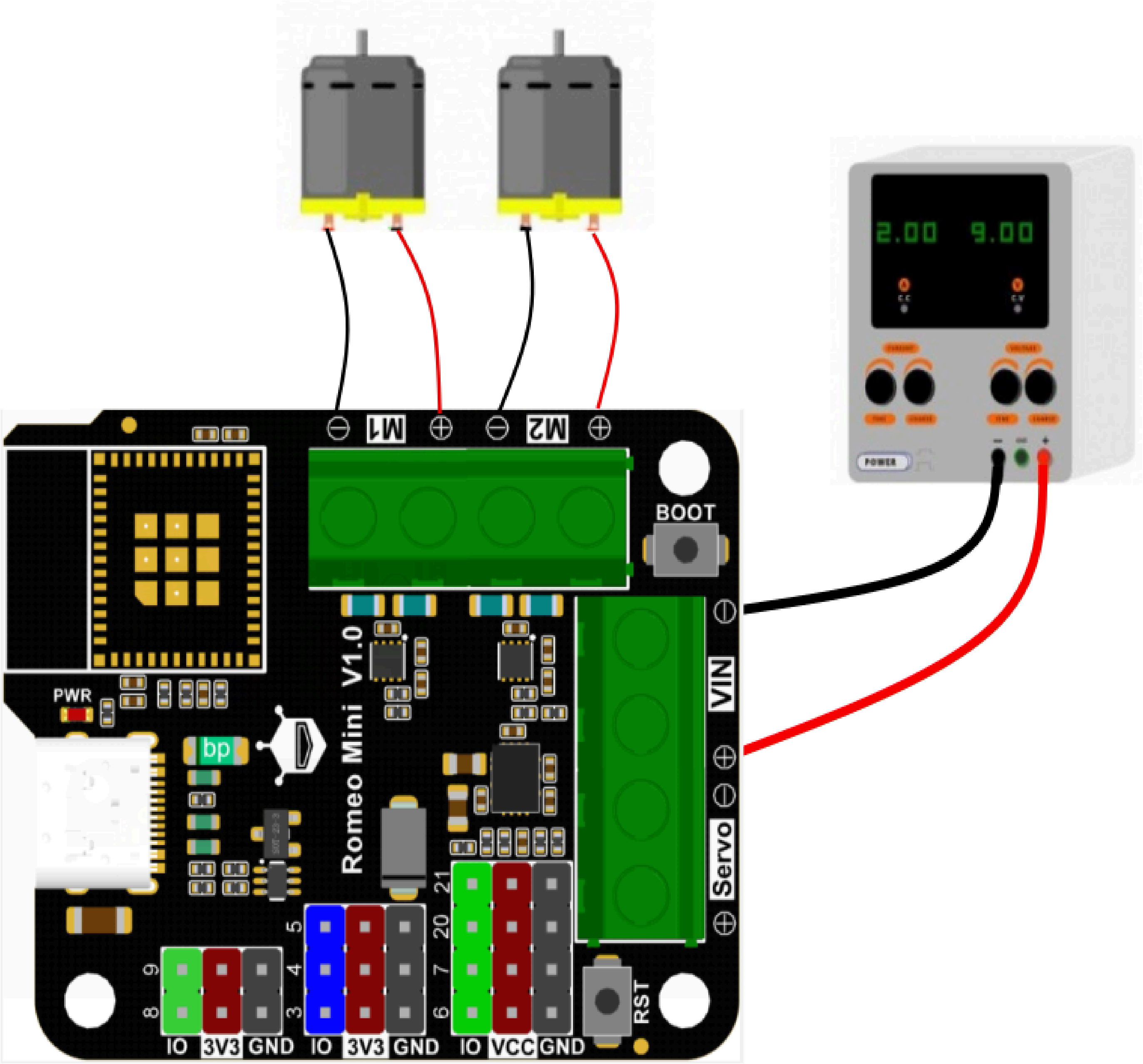
Requirements

- DC Motors: x2

- Romeo Mini: x1

Connection Diagram

The module is powered by 5-15V. Choose an appropriate power supply voltage based on the motors.



Control Modes

EN	PH	Function
H	H	Forward
H	L	Reverse
PWM	L	Forward
PWM	H	Reverse

Sample Code

```
int EN1 = 0; // PWM control for motor M1
int PH1 = 1; // Direction control for motor M1
int EN2 = 2; // PWM control for motor M2
int PH2 = 10; // Direction control for motor M2

void setup() {
  pinMode(EN1, OUTPUT);
  pinMode(PH1, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(PH2, OUTPUT);
}

void loop() {
  M1_Forward(200); // Motor M1 moves forward with adjustable PWM
  M2_Forward(200); // Motor M2 moves forward with adjustable PWM
  delay(5000);
  M1_Backward(200); // Motor M1 moves backward with adjustable PWM
  M2_Backward(200); // Motor M2 moves backward with adjustable PWM
  delay(5000);
}

void M1_Forward(int Speed1) // Forward fast decay mode for M1 motor, larger Speed1 value resu
{
  analogWrite(EN1, Speed1);
  digitalWrite(PH1, LOW);
}

void M1_Backward
```

Example Result: Execute the sample program by commanding simultaneous rapid forward rotation of motors M1 and M2, followed by simultaneous slow reverse rotation of motors M1 and M2.

3. PWM-Driven Servo Motor

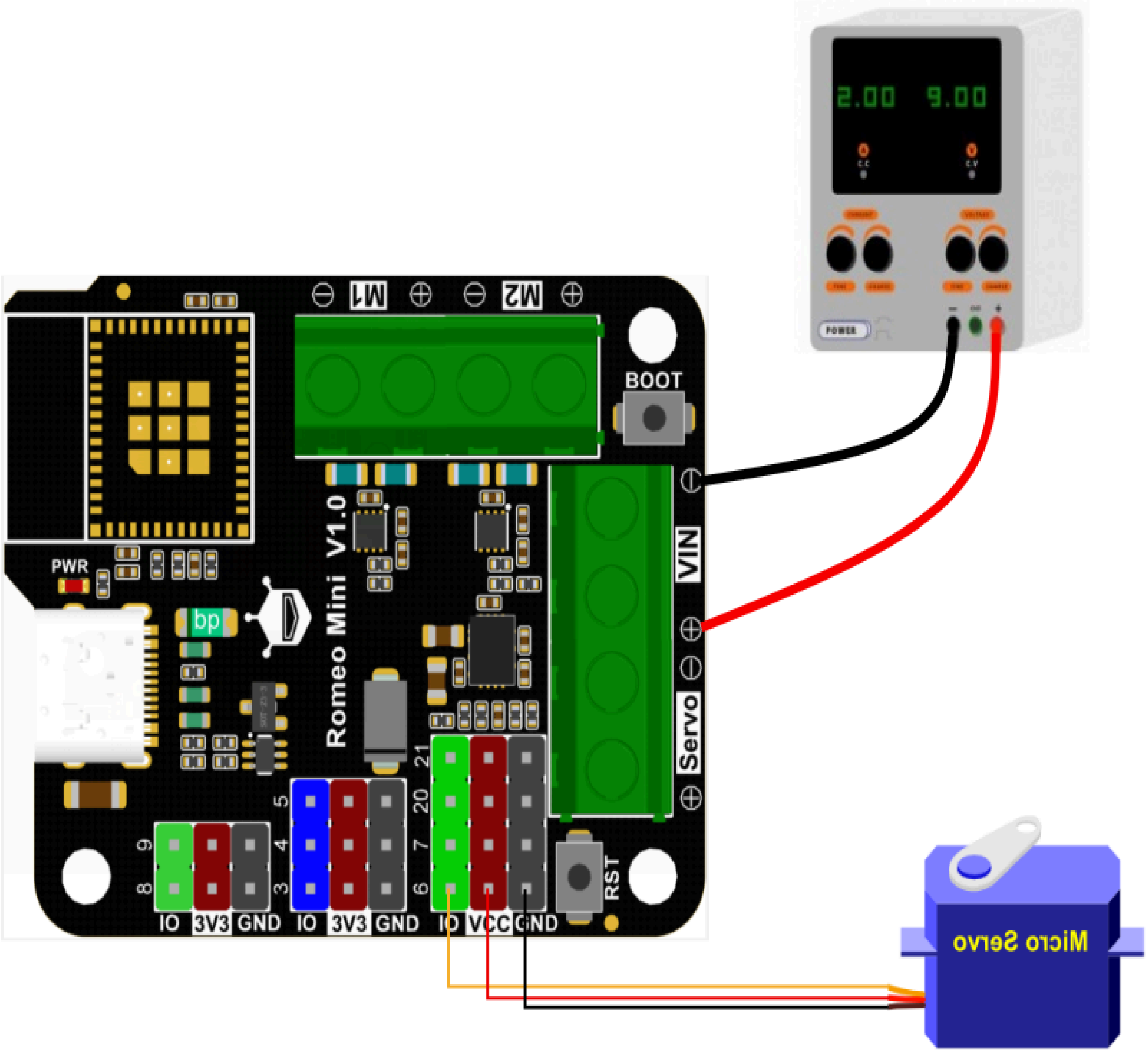
Prerequisites:

- 5V PWM servo motor x1
- Romeo mini x1

Connection Diagram:

If you need to operate a servo motor or peripheral device with a voltage higher than 5V, connect the Servon terminal to the power supply. The power supply range should be between 5-12V. The VCC terminal voltage is equal to the input voltage at the Servon terminal. When the Servon terminal is not connected, the default VCC terminal voltage is 5V.

>


M1
M2
BOOT
PWR
Romeo Mini V1.0
VIN
Servo
RST
IO 3V3 GND IO 3V3 GND IO VCC GND
8 9 3 4 5 6 7 20 21
2.00 9.00
POWER
Micro Servo

Sample Code

```
void setup() {
ledcSetup(0, 5000, 10); // Configure channel 0 with a frequency of 5KHz and 10-bit resolution
ledcAttachPin(6, 0); // Assign pin 6 as the output pin for channel 0
}

void loop() {
ledcWrite(0, 125); // Set the output of channel 0 to 125, producing a PWM output of 0 to 100%
delay(1000);
ledcWrite(0, 25);
delay(1000);
}
```

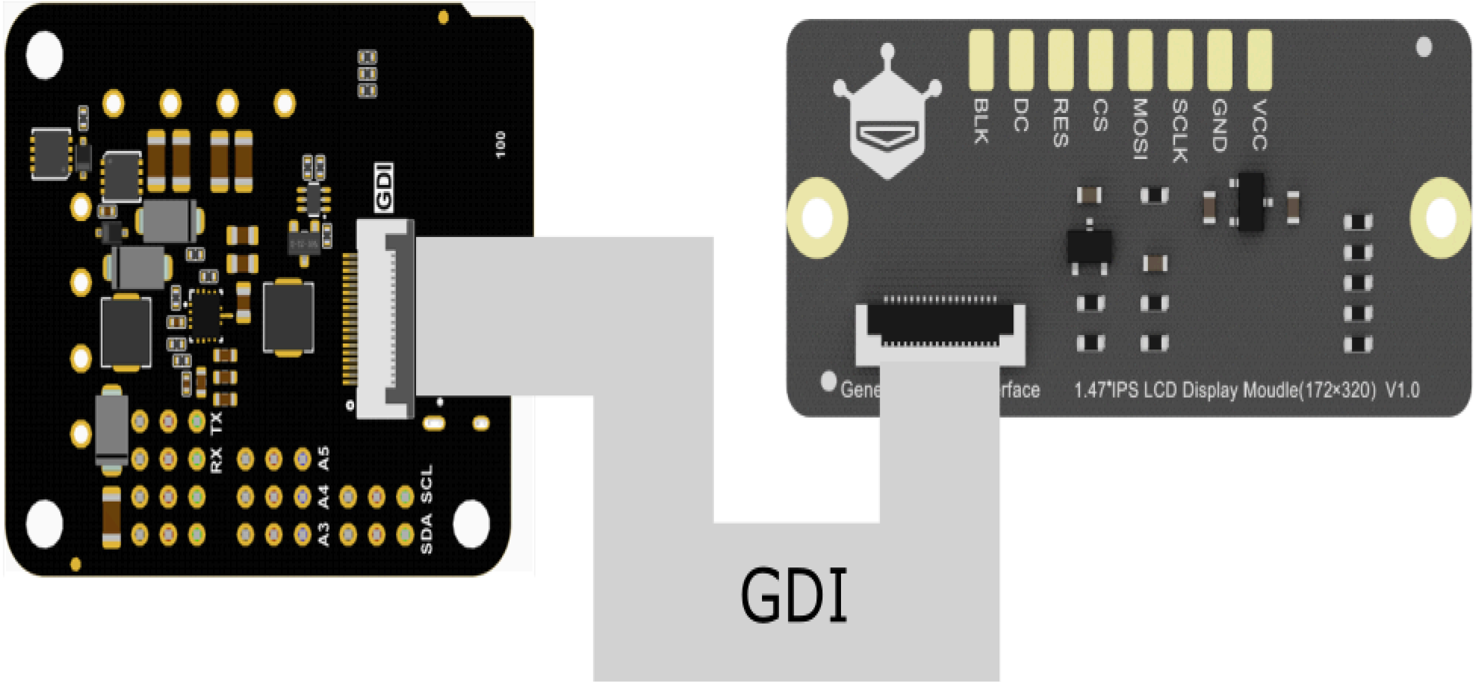
Example Result: Burn the sample program to cyclically rotate Servo 1 in the range of 0-180°.

4. Driving an SPI Display

Prerequisites:

- 1.47" 172×320 IPS LCD Display Module (https://www.dfrobot.com/product-2638.html)
- Romeo mini x1
- The DFR0664, DFR0649, DFR0847, DFR0928, and DFR0995 models are all compatible with the Romeo mini controller. For detailed usage instructions, please refer to the display's wiki page.

Connection Diagram:



GDI
100
RX
TX
A5
A4
A3
SCL
SDA
BLK
DC
RES
CS
MOSI
SCLK
GND
VCC
Gene
rface
1.47"IPS LCD Display Moudle(172×320) V1.0
GDI

Sample Code

```
#include "DFRobot_GDL.h"
#define TFT_DC 8
#define TFT_CS 7
#define TFT_RST 9

DFRobot_ST7789_172x320_HW_SPI screen(/dc=/TFT_DC,/cs=/TFT_CS,/rst=/TFT_RST);
/* M0 mainboard DMA transfer */
//DFRobot_ST7789_172x320_DMA_SPI screen(/dc=/TFT_DC,/cs=/TFT_CS,/rst=/TFT_RST);

void setup() {
screen.begin();
screen.setTextSize(2);
screen.fillScreen(COLOR_RGB565_BLACK); // Set the background color
screen.setFont(&FreeMono24pt7b); // Set the font size (9, 12, 18, 24)
screen.setCursor(/x=/32, /y=/200); // Set the text position
screen.setTextColor(COLOR_RGB565_LIGHTGRAY); // Set the text color
screen.setTextWrap(true);
screen.print("DF"); // Display the English characters
}

void loop() {
// Empty loop as no further actions are required for this example
}
```

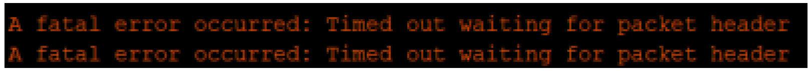
Example Result: Burn the sample program to display the gray English characters "DF" on the screen.

Frequently Asked Questions

Burn Error

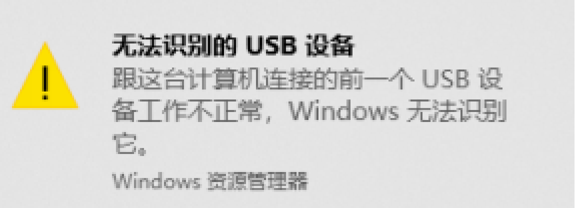
Cause

- If the delay in the loop is too short or no delay is added, it can result in a timeout error during the burning process.



```
A fatal error occurred: Timed out waiting for packet header
A fatal error occurred: Timed out waiting for packet header
```

- Incorrectly calling certain functions can cause the computer to not recognize the USB connection.



无法识别的 USB 设备

跟这台计算机连接的前一个 USB 设备工作不正常，Windows 无法识别它。

Windows 资源管理器

Solution

Press and hold the BOOT button, then click the RST button, and finally release the BOOT button.

No Serial Output

Solution

- Check if USB CDC is enabled.
- Use a different serial debugging tool to view the print information.
- For more questions and interesting applications, you can visit the forum for reference or to post your queries.

More

- esp32-c3-mini-1 datasheet (https://www.espressif.com.cn/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf)
- Motor driver chip drv8220 datasheet (https://dfimg.dfrobot.com/5ea64bf6cf1d8c7738ad2881/wiki/0d792b7309275ae76bd61cbc548219d2.pdf)

- Schematic V1.0.0 (https://dfimg.dfrobot.com/5ea64bf6cf1d8c7738ad2881/wiki/1a43b798fa4191f577a9bc8fa17dc3dc.pdf)