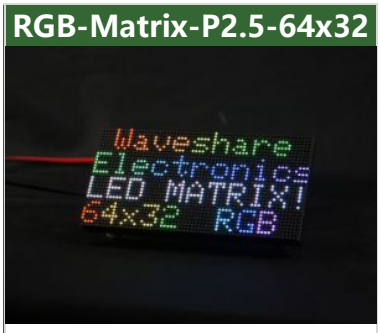
> ⚠ This product is a bare screen and needs to be displayed with the main control board such as Raspberry Pi, Pico, ESP32, and Arduino.



RGB LED, 64 x 32=2048 DOTS, 2.5mm Pitch
I/Os

Overview

Introduction

This product is a 64×32 full-color LED matrix display, with 2048 RGB LEDs on board, 2.5mm pitch, supports Raspberry Pi, Arduino, ESP32, etc., provides supporting open source demos and tutorials, suitable for makers or electronics enthusiasts getting started Learning, or DIY secondary development into other desktop or wall-mounted display applications.

Features

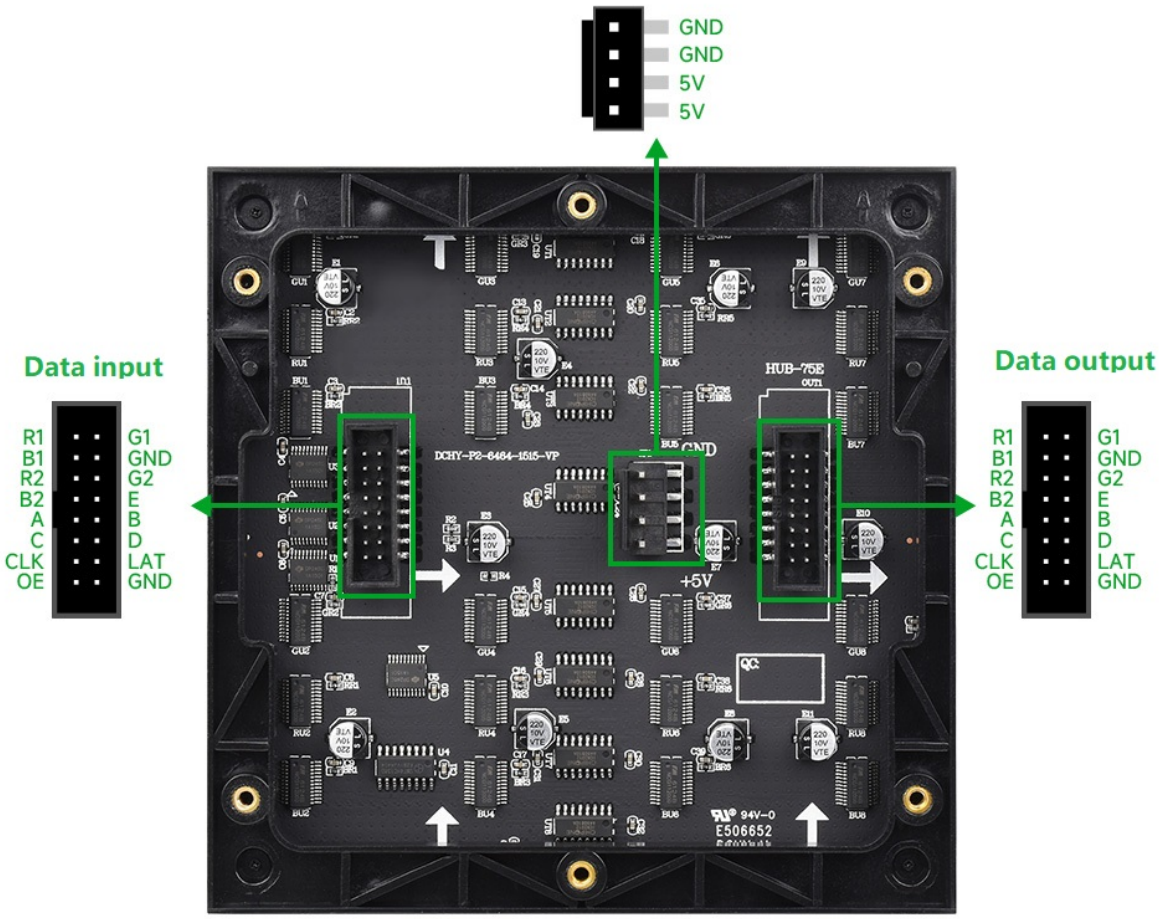
- 2048 individual RGB LEDs, full-color display, adjustable brightness.
- 64×32 pixels, 2.5mm pitch, allows displaying text, colorful images, or animation.
- 160×80mm dimensions, moderate size, suitable for DIY desktop display or wall mount display.
- Onboard two HUB75 headers, one for controller data input, one for output, and chain support.
- Provides open-source development resources (examples for Raspberry Pi / Raspberry Pi Pico / ESP32 / Arduino).

Parameters

Dimensions	160mm × 80mm
Pixel	64×32=2048 DOTS
Pitch	2.5mm
Pixel Form	1R1G1B
Viewing Angle	≥140°

Control Type	Synchronization
Driving	1/16 scan
Header	HUB75
Power Supply	5V / 2.5A (VH4 header input)
Power	≤12W

Pinout Difinition



The picture on the back is for reference only. Different batches of PCB board silk screen and layout may have minor adjustments, and the software is compatible. The actual arrival shall prevail.

PIN	Description		PIN	Description
+5V	5V power input		**GND**	Ground
R1	R higher bit data		**R2**	R lower bit data
G1	G higher bit data		**G2**	G lower bit data
B1	B higher bit data		**B2**	B lower bit data

A	A line selection		**B**	B line selection
C	C line selection		**D**	D line selection
E	E line selection		**CLK**	Clock input
LAT/STB	Latch pin		**OE**	Output enable

Note: The power port (VCC and GND) of the display is powered by 5V, do not connect to other voltages, so as not to burn out the display.

Application Scenarios

DIY maker desktop or wall display applications, billboards, environmental monitoring screens, etc.

Working With Raspberry Pi

Hardware Connection

Prepare Materials

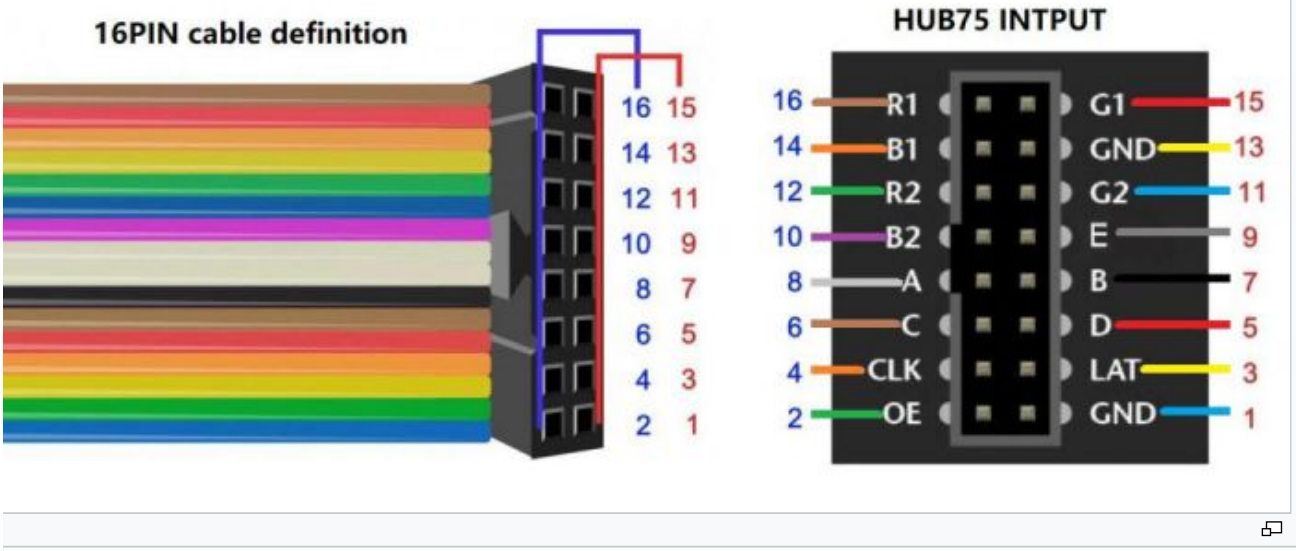
- RGB-Matrix-P2.5-64x32 (this product)

- Raspberry Pi (not included)

Hardware Connnection

The Raspberry Pi can share and connect up to three LED dot matrix screen panels. If there is only one panel, just connect the pins corresponding to the 😁 mark in the table below. [2]✹ and [3]● need to share the connection of the second and third panels at the same time.
The following table adds some icons for better visual distinction: [1]=😁, [2]=✹ and [3]=●.

> ! Note: If you just connect to a panel, just connect the pins of the 😁 icon.



Connection	Pin	Pin	Connection
-	1	2	-
● [3] G1	3	4	-
● [3] B1	5	6	GND 😁✹💧
😁✹💧LAT/STB	7	8	[3] R1 ●
-	9	10	E 😁✹💧
😁✹💧CLK	11	12	OE- 😁✹💧
😁 [1] G1	13	14	-
😁✹💧A	15	16	B 😁✹💧
-	17	18	C 😁✹💧
😁 [1] B2	19	20	-
😁 [1] G2	21	22	D 😁✹💧
😁 [1] R1	23	24	[1] R2 😁
-	25	26	[1] B1 😁

-	27	28	-
✹ [2] G1	29	30	-
✹ [2] B1	31	32	[2] R1 ✹
✹ [2] G2	33	34	-
✹ [2] R2	35	36	[3] G2 ●
● [3] R2	37	38	[2] B2 ✹
-	39	40	[3] B2 ●

Software Download & Run

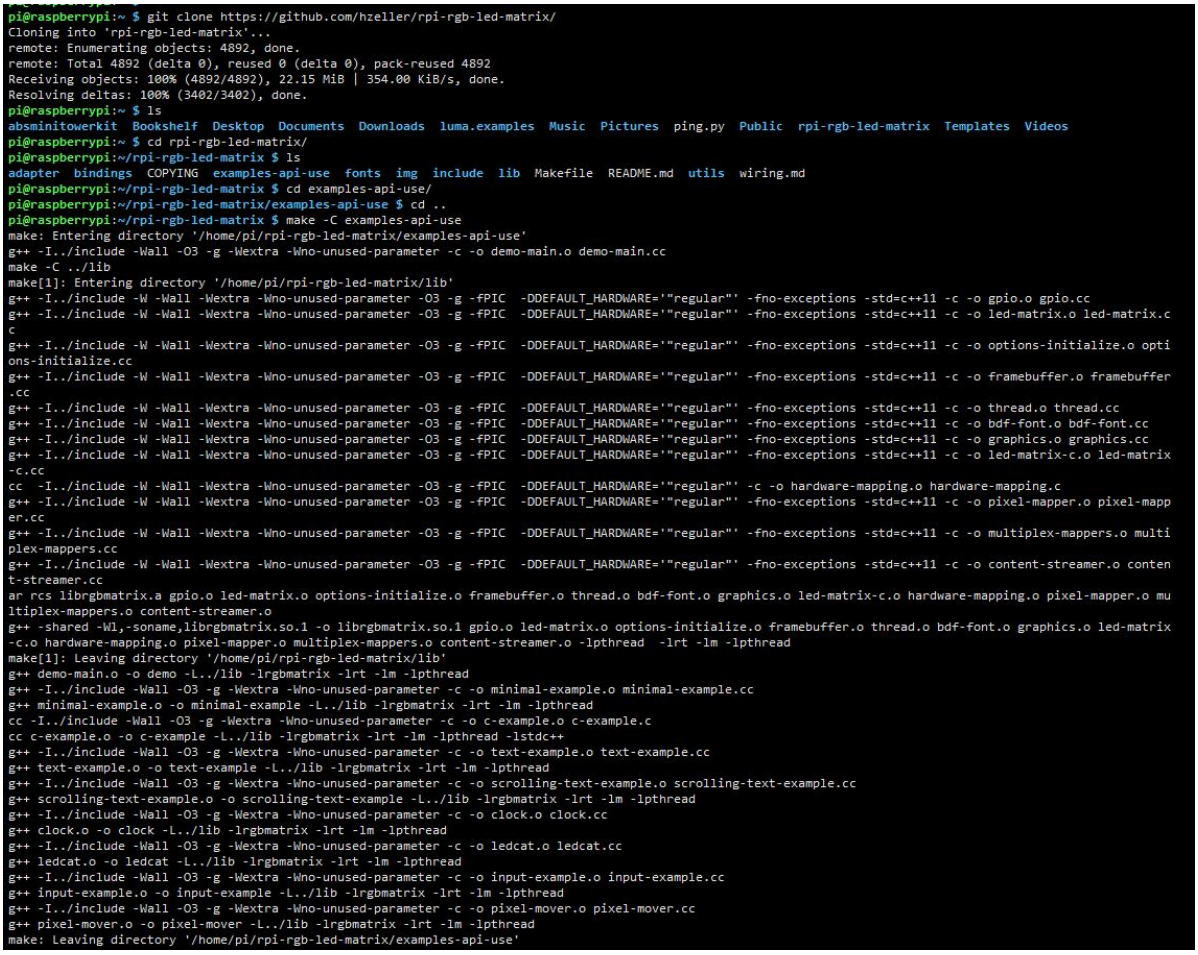
- Download the open source project on GitHub to the Raspberry Pi.

```
git clone https://github.com/hzeller/rpi-rgb-led-matrix/
```

- After the download is complete, it can be found in the `examples-api-use/` directory:

```
make -C examples-api-use
```

- Download and compile process refer to the figure below:



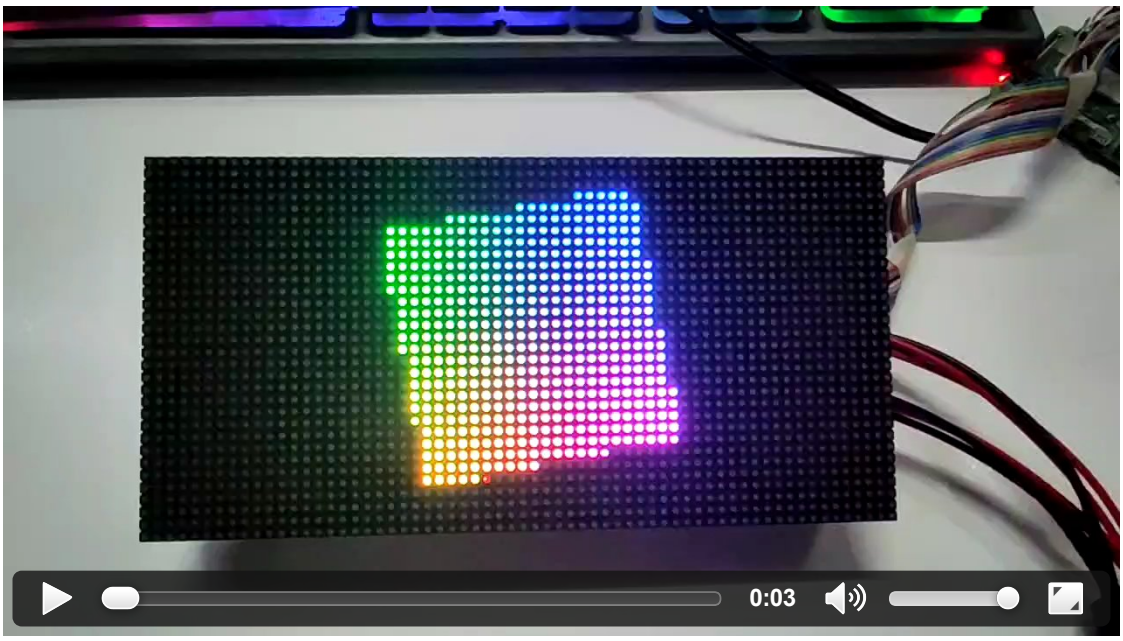
Example Running Effect

Demo

1. Execute the following command to run the demo:

```
cd examples-api-use
sudo ./demo -D0 --led-no-hardware-pulse --led-cols=64 --led-rows=32
```

2. The effect is shown in the figure below:



C-example

1. Execute the following command to run the demo:

```
sudo ./c-example -D0 --led-no-hardware-pulse --led-cols=64 --led-rows=32
```

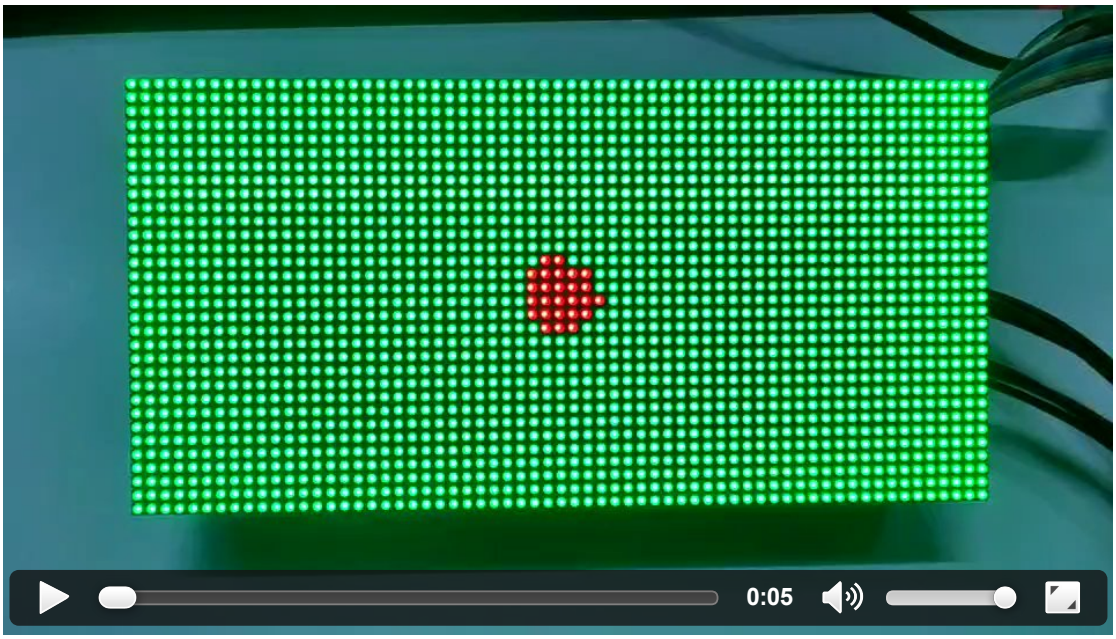
2. The effect is shown in the figure below:

Minimal-example

1. Execute the following command to run the demo:

```
sudo ./minimal-example -D0 --led-no-hardware-pulse --led-cols=64 --led-rows=32
```

2. The effect is shown in the figure below:



Text-example

1. Execute the following command to run the demo:

```
sudo ./text-example -f ../fonts/8x13.bdf --led-no-hardware-pulse --led-cols=64 --l
ed-rows=64
```

2. After running the demo, input the characters you want to display one by one, press Enter, and the corresponding output will be displayed on the display:

```
pi@raspberrypi:~/rpi-rgb-led-matrix/examples-api-use $ sudo ./text-example -f ../f
onts/8x13.bdf --led-no-hardware-pulse --led-cols=64 --led-rows=64
Suggestion: to slightly improve display update, add
        isolcpus=3
at the end of /boot/cmdline.txt and reboot (see README.md)
Enter lines. Full screen or empty line clears screen.
Supports UTF-8. CTRL-D for exit.
Hello
Wvshare
Welcome
```

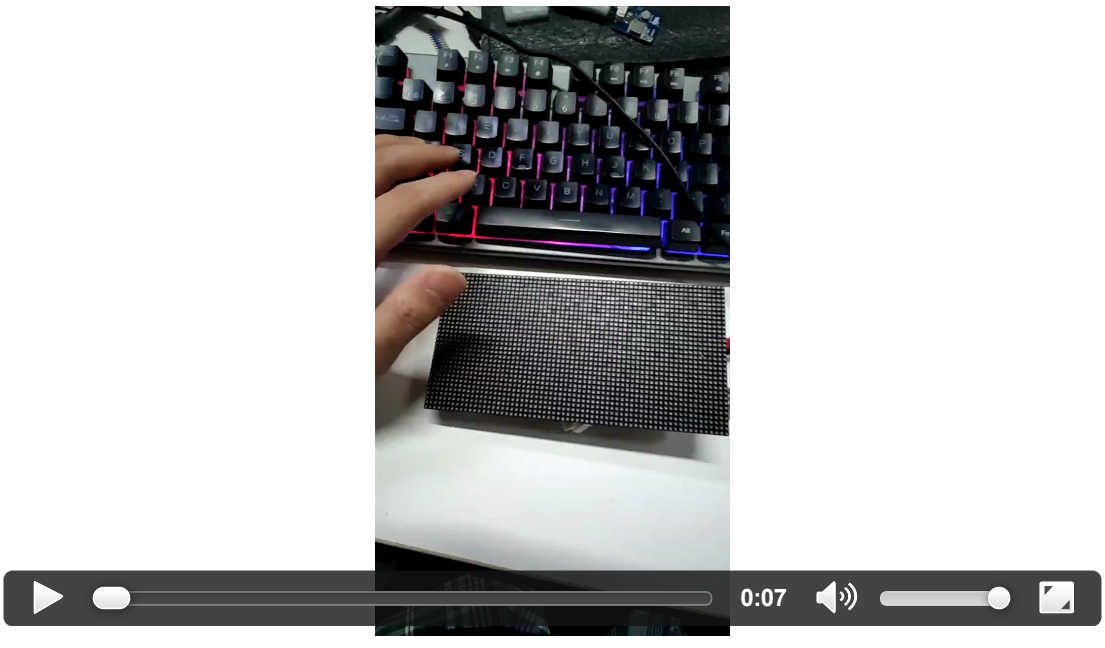
3. The effect is shown in the figure below:

Pixel-mover

1. Execute the following command to run the demo:

```
sudo ./pixel-mover --led-no-hardware-pulse --led-cols=64 --led-rows=32
```

2. The effect is as shown in the figure below: You can move the light spot on the display screen by pressing W, A, S, and D on the keyboard



Clock

1. Execute the following command to run the demo:

```
sudo ./clock -f ../fonts/7x13.bdf --led-cols=64 --led-rows=32 -d "%A" -d "%H:%M:%
S" --led-no-hardware-pulse
```

2. The effect is shown in the figure below:

For more gameplay related to this open-source project, please refer to GitHub: Demo, wiring reference.

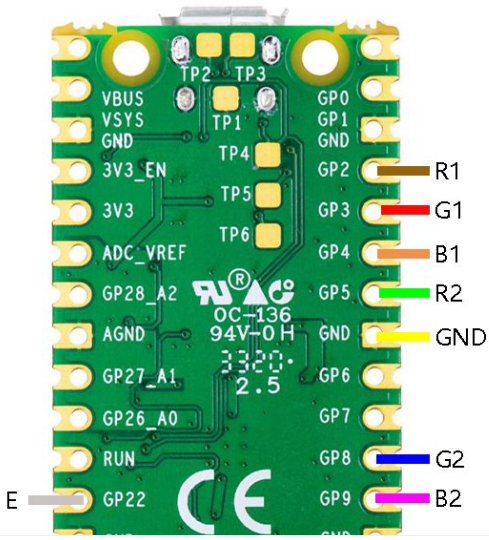
Working With Raspberry Pi Pico

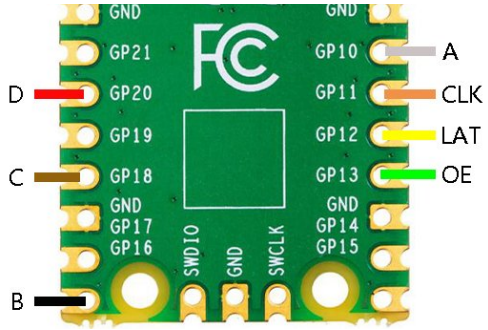
Hardware Connection

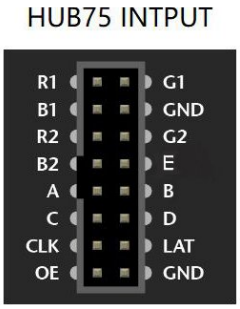
Prepare Materials

- RGB-Matrix-P2.5-64x32 (this product)
- Raspberry Pi Pico (need to be purchased separately, if not available, it is recommended to buy the version with soldered pin headers, which is convenient for wiring.)

Hardware Connection







Environment Setting

With Pico, we use CircuitPython as an example of RGB-Matrix. If you are not familiar with CircuitPython, you can first learn the official recommended guide "Raspberry Pi Pico CircuitPython Getting Started Tutorial (English Version)".
This guide covers the basics of getting started with CircuitPython and the use of the editor.

CircuitPython Development Environment Setting

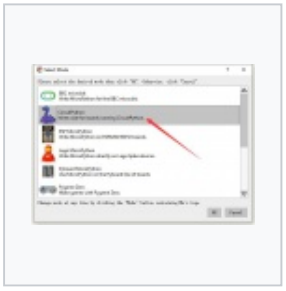
In order to facilitate the programming, development and debugging of CircuitPython, it is recommended to use the "Mu Editor" development software.
You can use Mu Editor for Pico's CircuitPython development on Windows.
The following describes the development and use of Mu Editor under Windows.

Build and Use Windows Development Environment (Mu Editor)

- Download Mu editor and install it by steps.
- After the installation is complete, configure the language and select the mode for the first time. Since we are using CircuitPython, pay attention to the mode and select the CircuitPython option.
- After the configuration is complete, it will display that the device cannot be found, because Pico has not downloaded the CircuitPython firmware library.
- Download the CircuitPython firmware library and program it into Pico.

1. Download the CircuitPython UF2 file.
2. Hold down the BOOTSEL button, then plug the Pico into a USB port on your Raspberry Pi or another computer.
3. Release the BOOTSEL button after connecting the Pico.
4. It will install as a mass storage device named "RPI-RP2".
5. Drag and drop the CircuitPython UF2 file onto the "RPI-RP2" volume. Your Pico will reboot, a new disk drive will appear named CIRCUITPY, and you're done flashing.
6. There will be a default code.py file in the new disk drive, you open it with Mu Editor, and the content in it is: "print("Hello World!")", the specific opening steps are shown in the last figure.

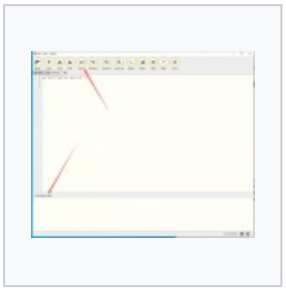
- Open the serial port, click the blank space and press Ctrl+C, then press Ctrl+D or click the blank space of the code interface and press Ctrl+S to run the demo, and you can observe the running effect in the CircuitPython REPL window.



Select Mode

Load

Select file



Open the serial port

Run the demo

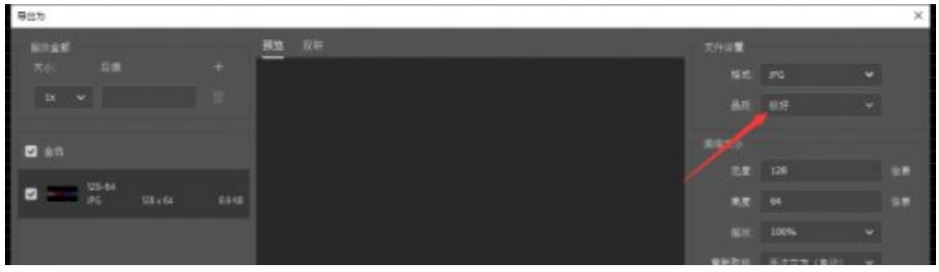
Software Download

- Download the sample demo.

> Note: After copying the sample demo to pico, you need to delete the original code.py code in Pico.

Example

- After setting up the CircuitPython environment, copy all the contents under the CircuitPython directory in the downloaded Pico example to the recognized USB flash drive, and then you can run the example (there are 16 demos in this code).
- The PSD folder is a file in .psd format, which can be used to modify the text and pictures that need to be moved. The modified pictures need to be saved as "excellent" and the format needs to be converted to BMP format.



【Function Description】

- Display text
- Set scrolling effect

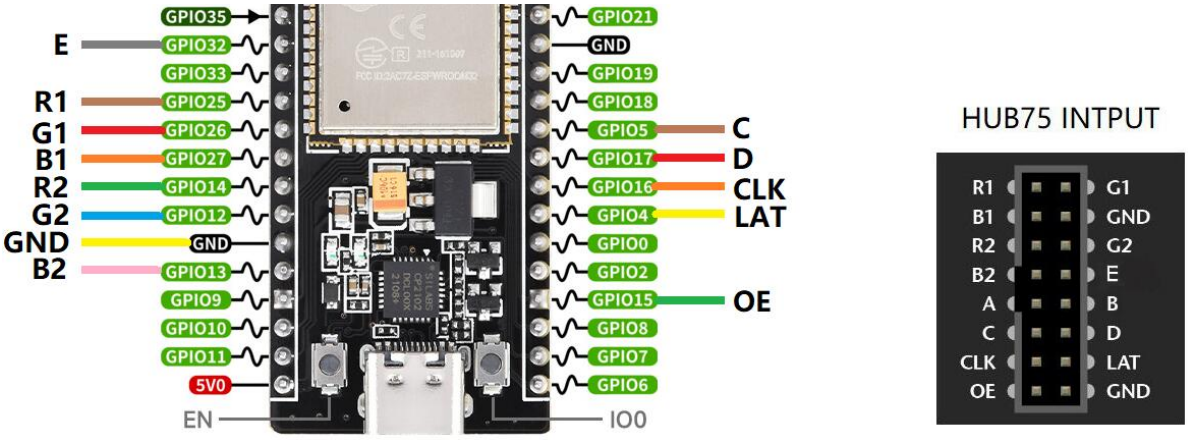
Working With ESP32

Hardware Connection

Prepare Materials

- RGB-Matrix-P2.5-64x32 (this product)
- NodeMCU-32S (not included)

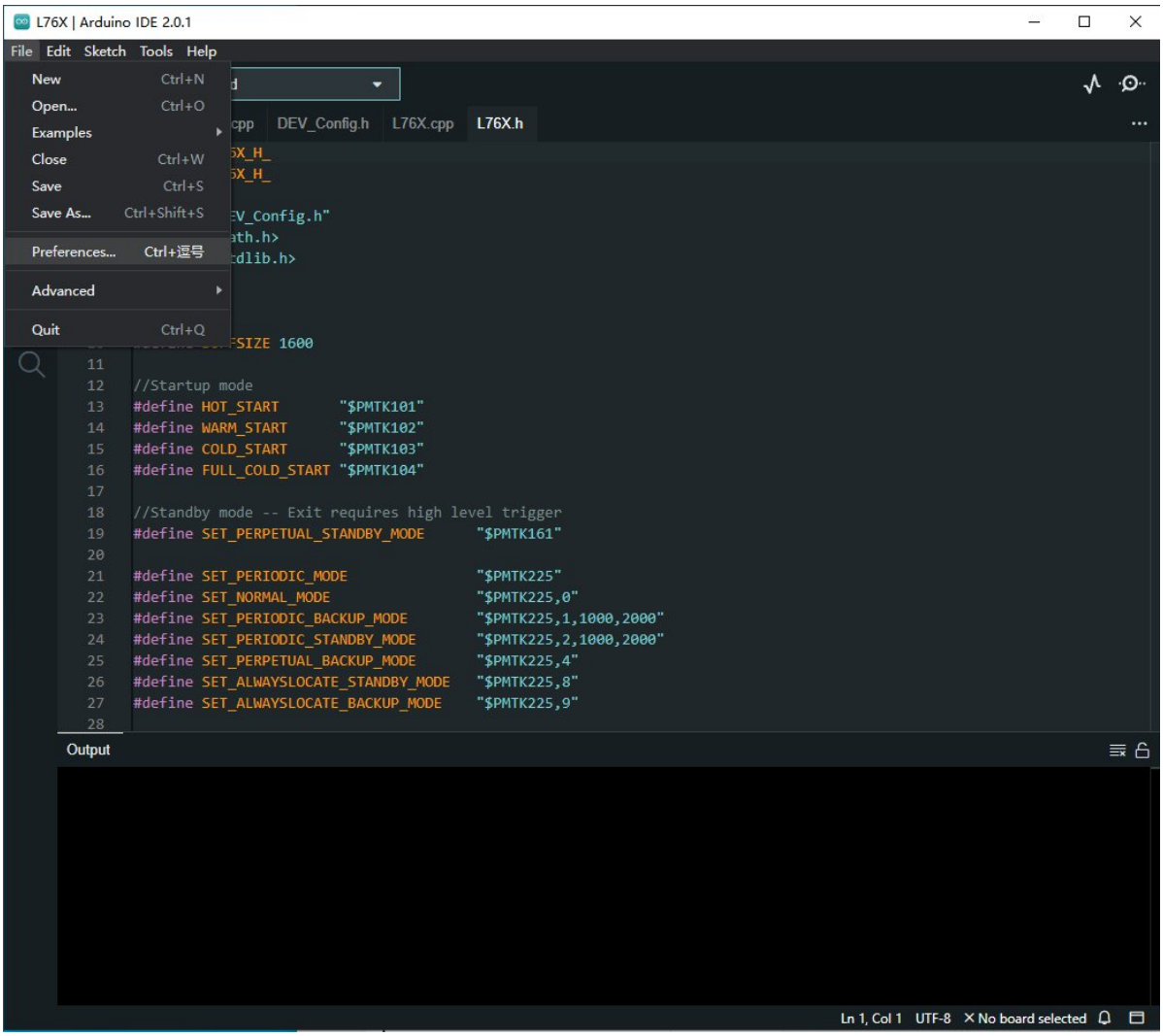
Hardware Connection Diagram



- Install the Arduino IDE (version 1.8.15 is available, or you can download the new version from the Arduino official website).

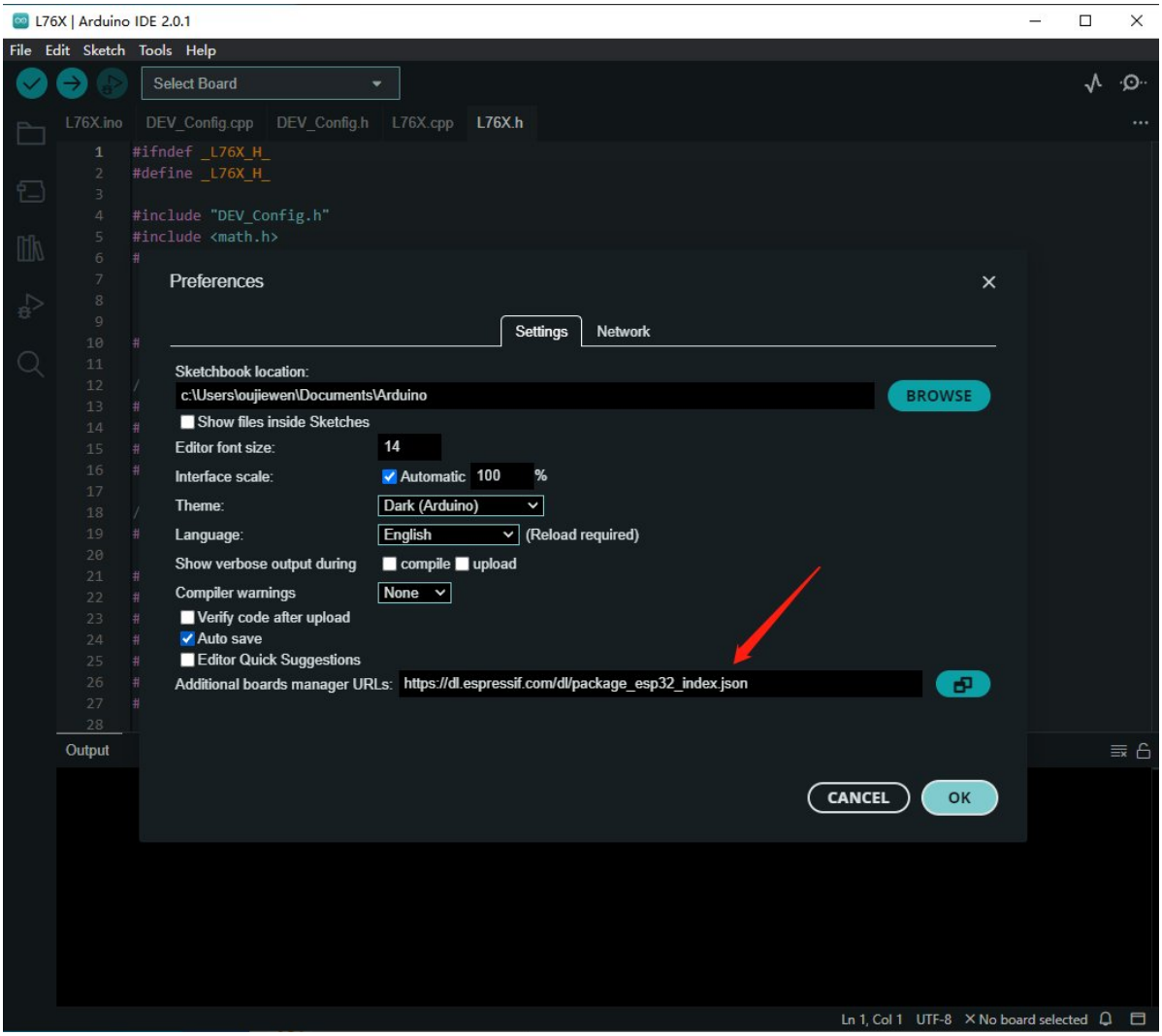
Install the ESP32 PLug-in in the Arduino IDE

1. Open the Arduino IDE, click on the file in the upper left corner, and select Preferences.



2. Add the following link in the additional development board manager URL, then click OK.

```
https://dl.espressif.com/dl/package_esp32_index.json
```

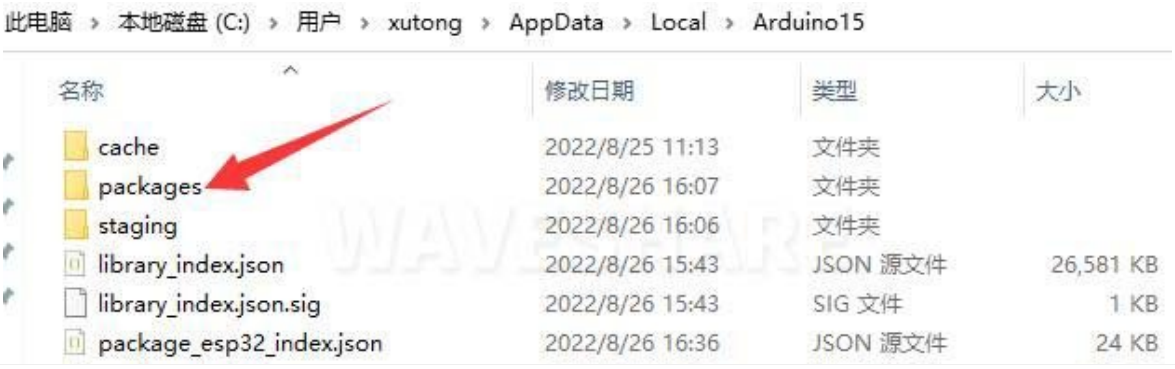


Note: If you already have the ESP8266 board URL, you can separate the URLs with commas like this:

```
https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

3. Download the packages compressed package and copy the decompressed packages file to the following path:

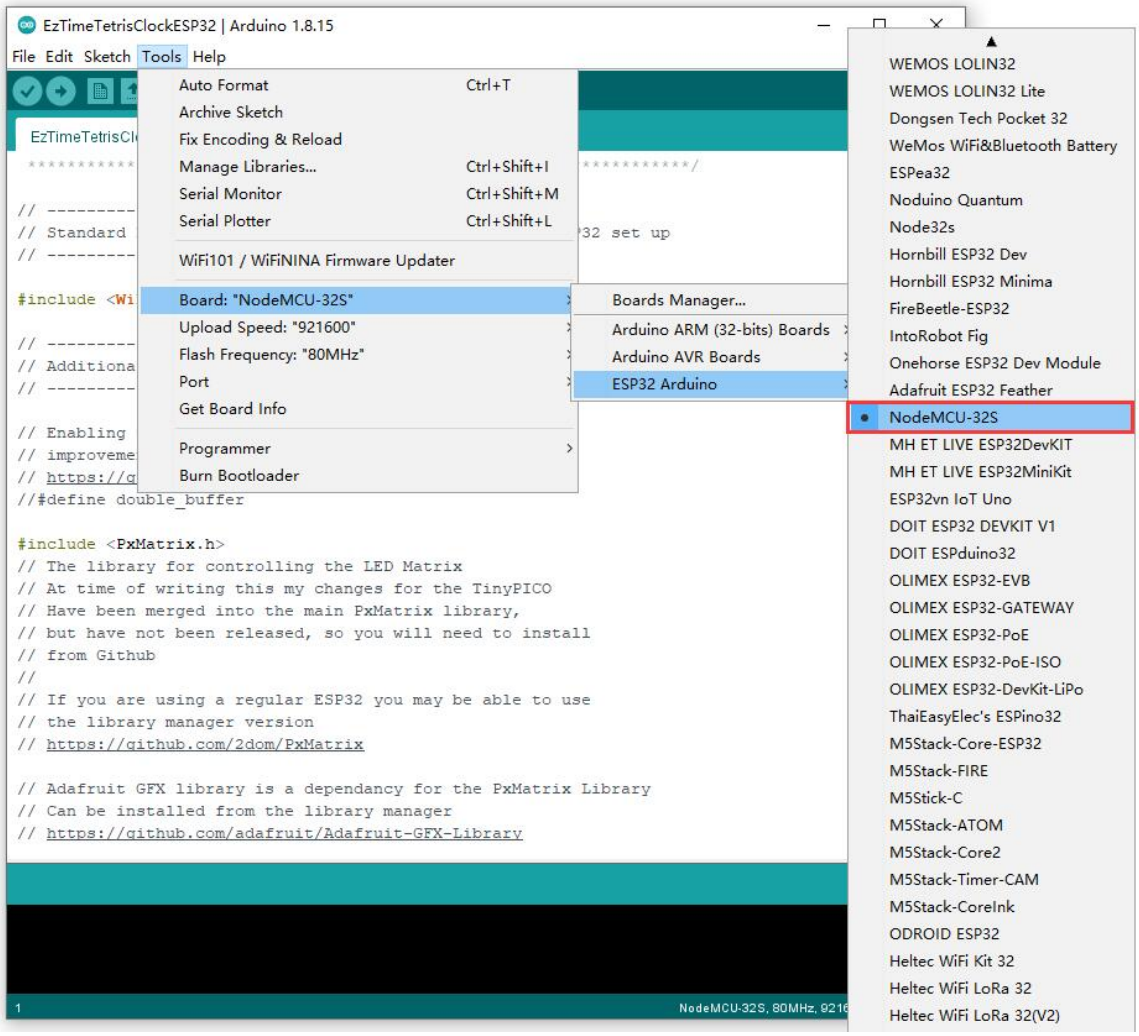
```
C:\Users\xutong\AppData\Local\Arduino15
```



Note: Replace the username: xutong with your own username.

Software Download And Run

- Download the sample demo.
- Copy the files under `..\ESP32\libraries` to the libraries under the Arduino IDE installation directory
  (Arduino IDE 2.0 or above, the libraries path is general C:\Users\"your_user_name"\AppData\Local\Arduino15\libraries)
- After connecting the wires according to the hardware connection diagram, the software settings are as follows:



- Open the demo through File, see the relative path: `RGB-Matrix-P2-64x64-Demo\ESP32`

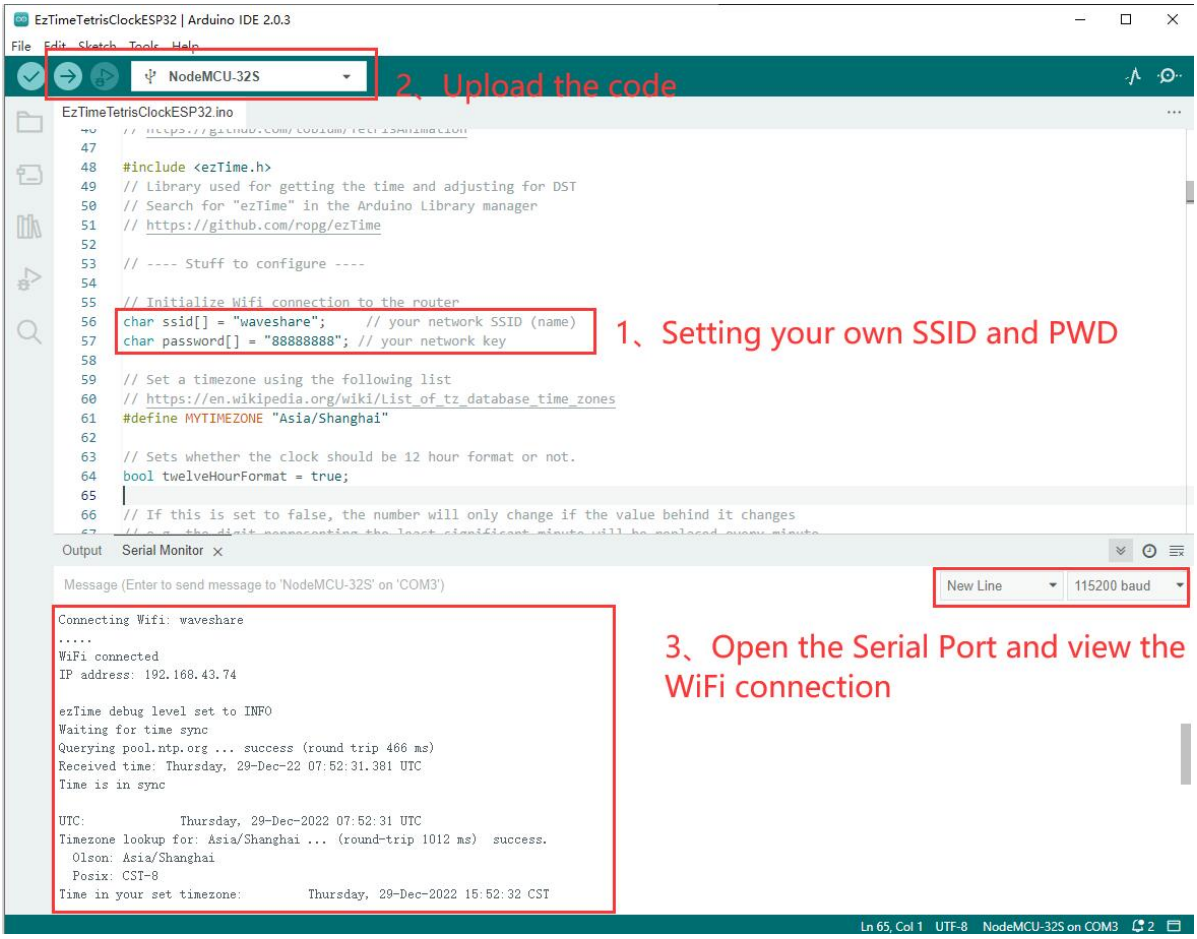
Example Running Effect

Building Block Digital Clock (With Network Calibration)

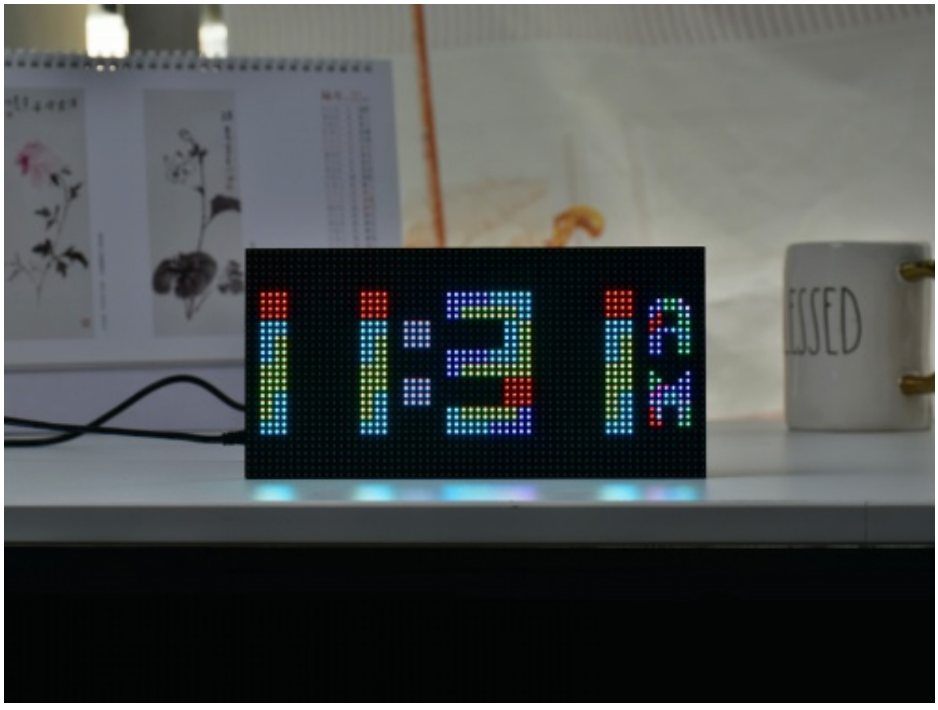
- Hardware connection diagram reference:

ESP32-connected.png

- Arduino IDE demo download (note that you need to change the WiFI corresponding to your home WIFI account password, if you cannot connect to the wifi, it will not be displayed normally):



The effect of running the example is shown in the figure below:



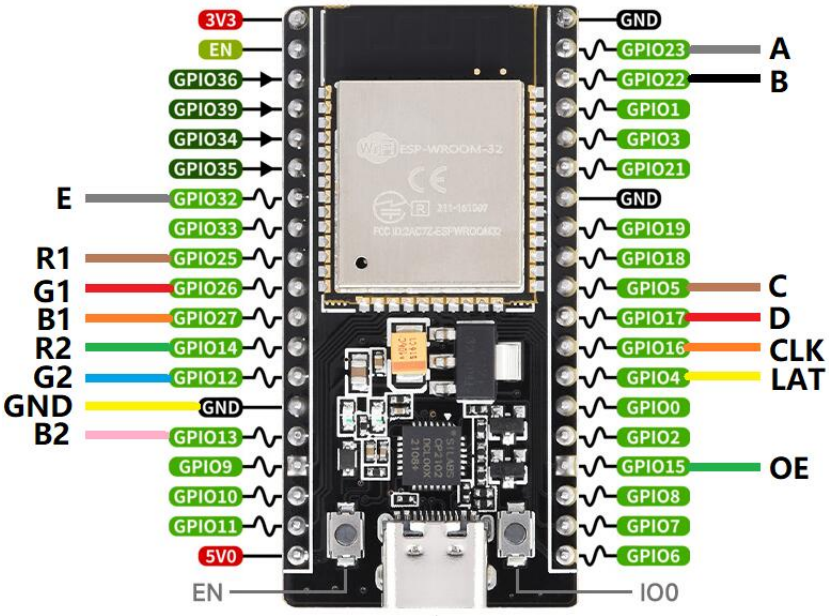
【Function Description】

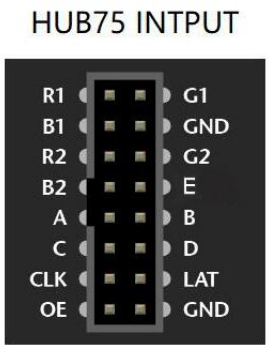
- Time display interface:
  - Calibrate time via ESP32 network
  - Display time
  - The animation effect of building blocks is displayed

> For more information about this GitHub open-source project, please see: WiFi-Tetris-Clock

Other Demos

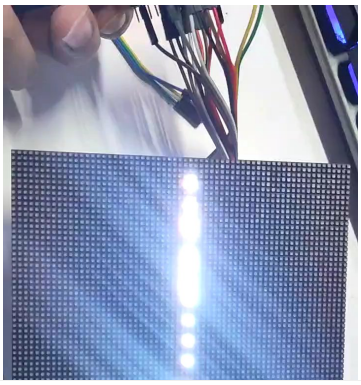
Hardware Connection Diagram



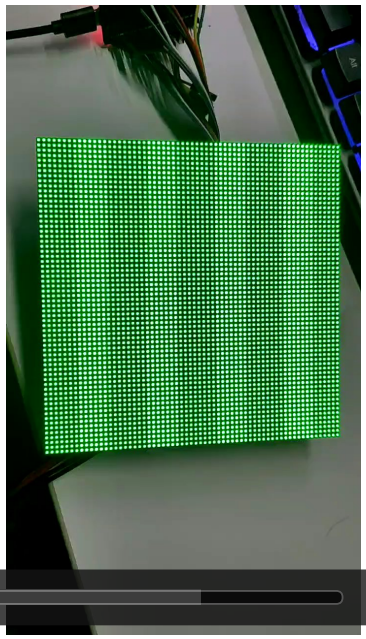



```
SimpleTestShapes: Displays basic shapes
PatternPlasma: Display cool plasma patterns
BouncingSquares: Displays bouncing squares
AuroraDemo: A simple example showing various animation effects
```

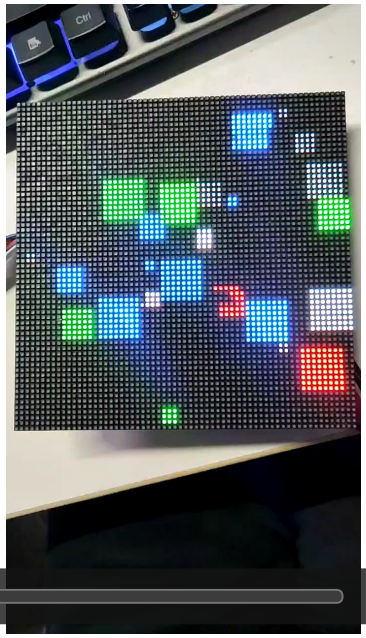
> There is an open source project on Github: ESP32-HUB75-MatrixPanel-I2S-DMA, which has a more detailed introduction.



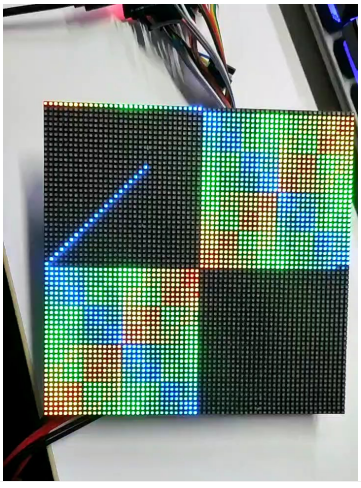
0:04

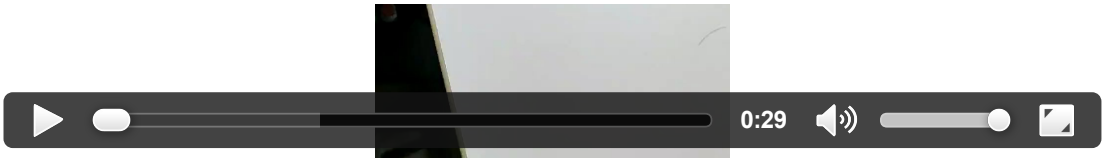


0:10



0:04





Working With Arduino Mega

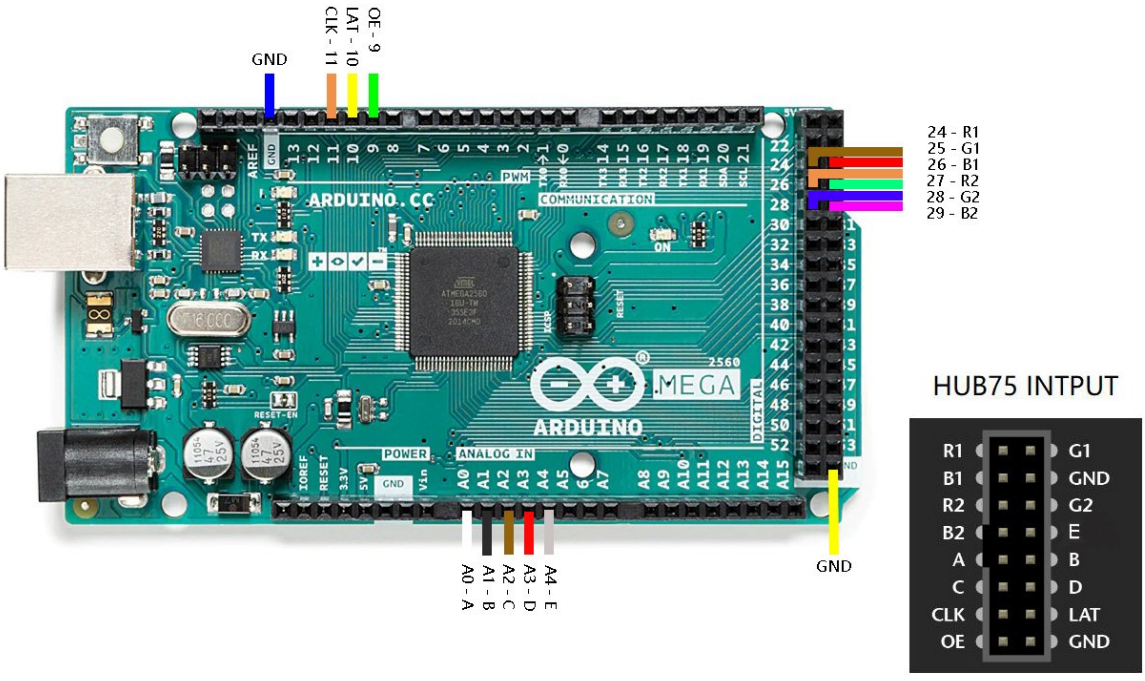
Hardware Connection

Prepare Material

- RGB-Matrix-P2.5-64x32 (this product)
- Arduino Mega2560 (not included)

> Because Arduino Uno has only 32KB of memory, it is not suitable for displaying too many bitmaps and information on the LED matrix, so it is recommended to use Arduino Mega2560 with 256KB of memory.

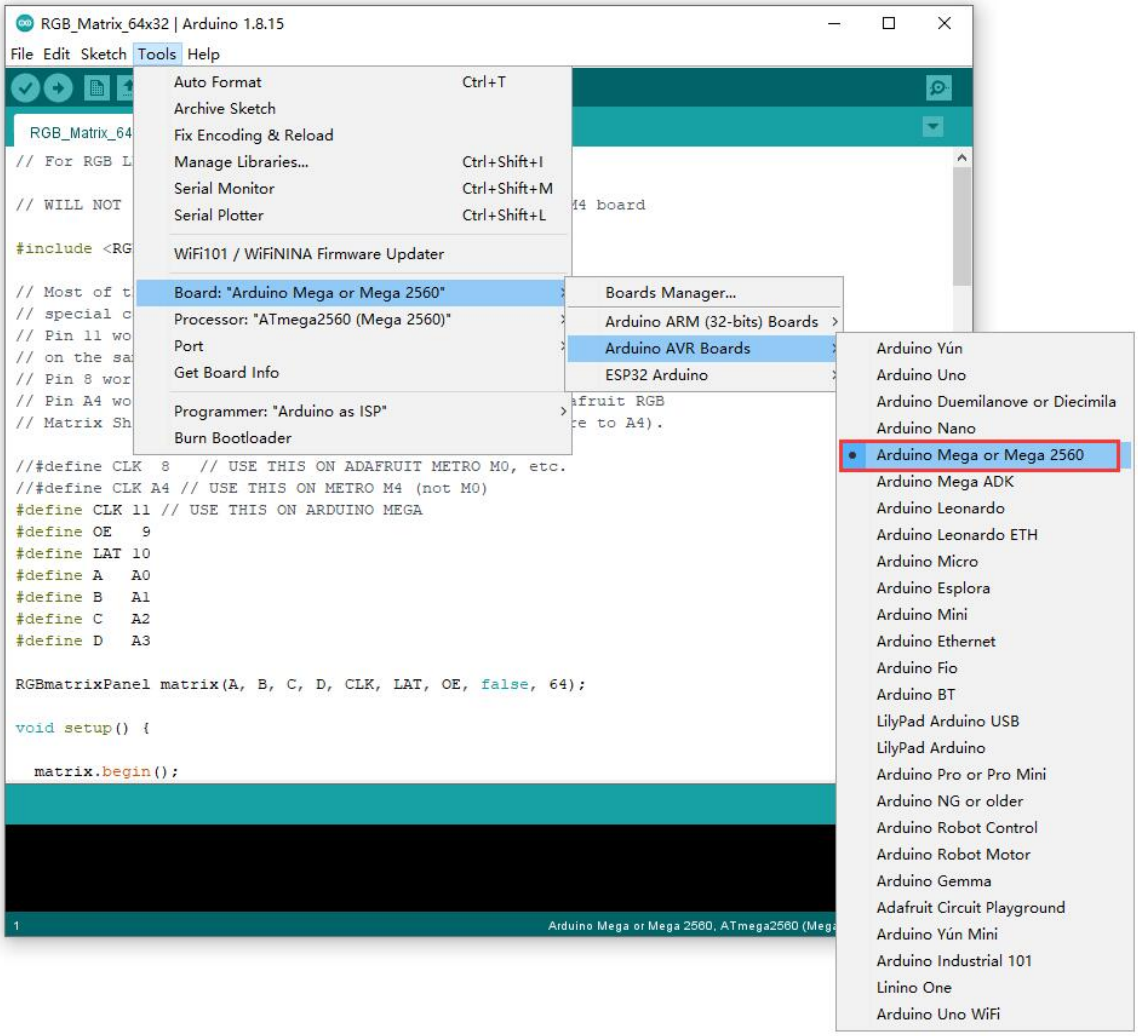
Hardware Connection Diagram



Software Setting

- Install Arduino IDE (Available version 1.8.15, you can also download a new version through Arduino official website).
- Download sample demo.

- Software selection: `Tools` -> `Board` -> `Arduino Mega or Mega 2560`, as shown below:



- Open the demo through File, see the relative path: `RGB-Matrix-P2-64x64-Demo\Arduino MEGA\RGB_Matrix_64x64_P2`

Example Running Effect

Part of the screen of the example running is shown in the figure below:

【Function Description】

- UI:
  - Draw lines, shapes, and patterns.
  - icon can be displayed.
  - Text content such as text and numbers can be displayed.
  - Can display Chinese characters and pictures.
  - Users can customize the displayed text, drawings or animation pictures, etc.

Resource

Document

- Note: Currently only supports font sizes of 16*16, 32*32, and 64*64.
- English Character Display Principle.pdf

Demo

- Sample demo

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 AM GMT+8 (Monday to Friday)

Submit Now