

# Overview

## Features

- I2C interface, only two cables can be directly controlled, easy to access a variety of control boards
- Can display up to 16 x 2 characters, support screen scrolling, cursor movement, and other functions
- The appearance is exquisite and compact, and the workmanship is exquisite and beautiful. Highly integrated design, stable performance, fine workmanship, compact body with the dimensions of 87 \* 32 \* 13 mm.
- Onboard AiP31068L LCD driver chip
- Provide complete supporting information manuals (user manuals/sample demos)



## Parameters

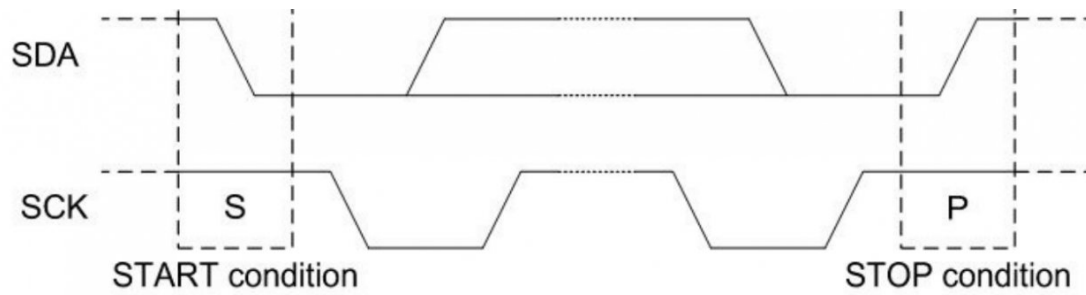
- Operating voltage: 3.3V/5V
- Communication interface: I2C
- Screen type: LCD screen
- Control chip: AiP31068L
- Display dimensions: 64.5 x 16.0 mm
- Outline dimensions: 87.0 x 32.0 x 13.0(mm)
- Operating current: 26mA(5V), 13mA (3.3V)

## Interface Description

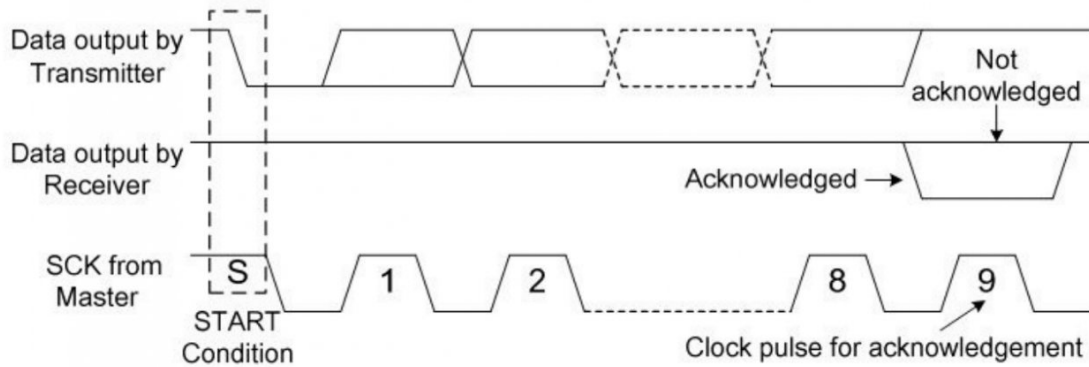
Pin	Description
VCC	3.3V/5V power input
GND	Ground
SCL	I2C clock cable
SDA	I2C data cable

## Communication Protocol

START STOP:



ASK :



Please refer to the data sheet for more details: [LCD1602\\_I2C\\_Module.pdf](#)

## Working With Raspberry Pi

### Hardware Setting

#### Enable I2C Interface

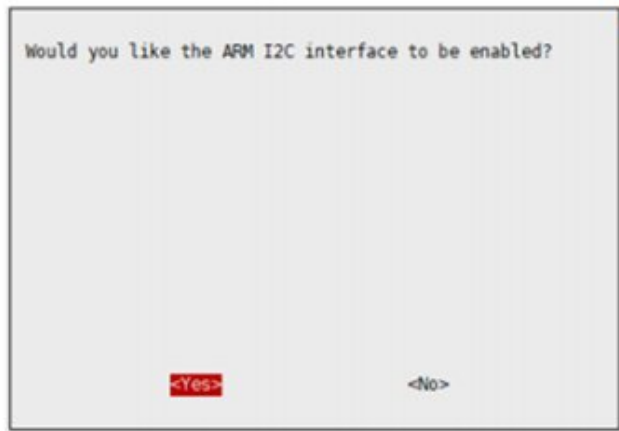
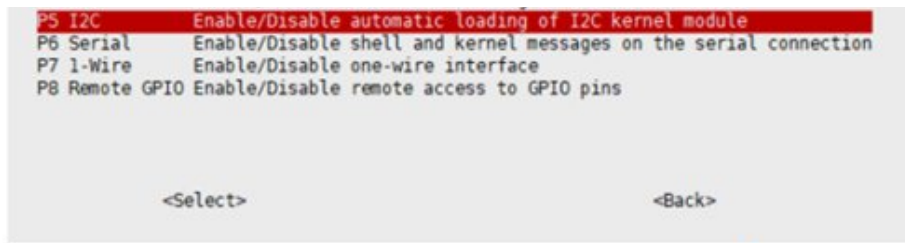
Open the Raspberry Pi terminal, enter the following command to enter the configuration interface.

```
sudo raspi-config
Select Interfacing Options -> I2C -> yes to start the i2C kernel driver
```

```
Raspberry Pi Software Configuration Tool (raspi-config)
1 Change User Password Change password for the current user
2 Network Options Configure network settings
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options Configure connections to peripherals
6 Overclock Configure overclocking for your Pi
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool

<Select> <Finish>
```

```
Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera Enable/Disable connection to the Raspberry Pi Camera
P2 SSH Enable/Disable remote command line access to your Pi using SSH
P3 VNC Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI Enable/Disable automatic loading of SPI kernel module
```



And then reboot the Raspberry Pi.

```
sudo reboot
```

## Hardware Connection

When connecting the Raspberry Pi, choose to connect with a 4PIN cable, please refer to the pin correspondence table below.

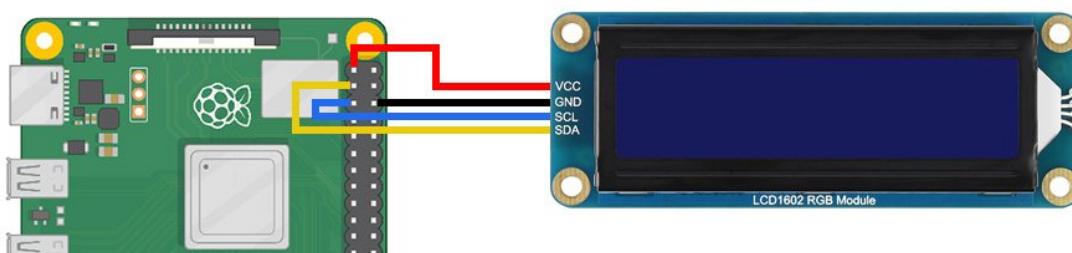
If you are using a pin header or PH2.0 4PIN interface, you need to connect according to the following table.

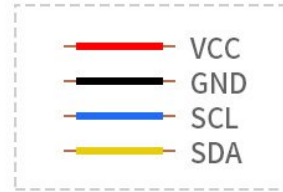
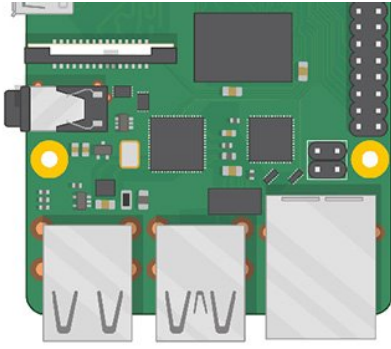
Raspberry Pi Pin Connection Correspondence

LCD	Raspberry Pi	
	BCM2835	Board Pin No.
VCC	3.3V	3.3V
GND	GND	GND
SCL	SCL.1	5
SDA	SDA.1	3

Take the LCD1602 Module using the PH2.0 4PIN interface as an example, and connect it to the Raspberry Pi according to the above table:

(Please connect according to the pin definition table. The color of the cable in the picture is for reference only, and the actual color shall prevail.)





## Example Demo

```
#Download sample demo
cd ~
wget https://files.waveshare.com/upload/d/db/LCD1602_I2C_Module_code.zip
unzip LCD1602_I2C_Module_code.zip
cd ~/LCD1602_I2C_Module_code/Raspberry
sudo chmod 777 *
cd python/
sudo python test.py #Display two lines of characters on the LCD
sudo python time_test.py #Automatically obtains the local time and displays it on
the LCD
```

- Demo analysis

test.py

```
lcd.setCursor(0, 0) #Set the cursor position
#Print the number of seconds since reset:
lcd.printout("Waveshare") #Write characters
lcd.setCursor(0, 1) #Set the cursor position to the zeroth column of the second ro
w
lcd.printout("Hello, World!")#Write characters
```

## Working With Pico

### Environment Setting

1. Raspberry Pi environment settings, please click to view [the link](#).
2. Windows environment settings, please click to view [the link](#).

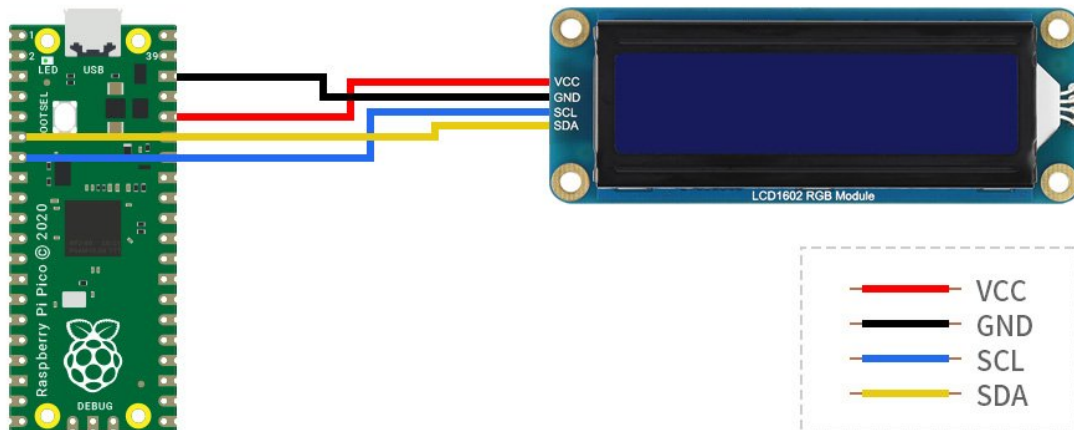
### Hardware Connection

Pico Pin Connection Correspondence

Module Interface	Pico Pin
VCC	3.3V
GND	GND

SCL	GP5
SDA	GP4

The physical connection diagram is as follows:



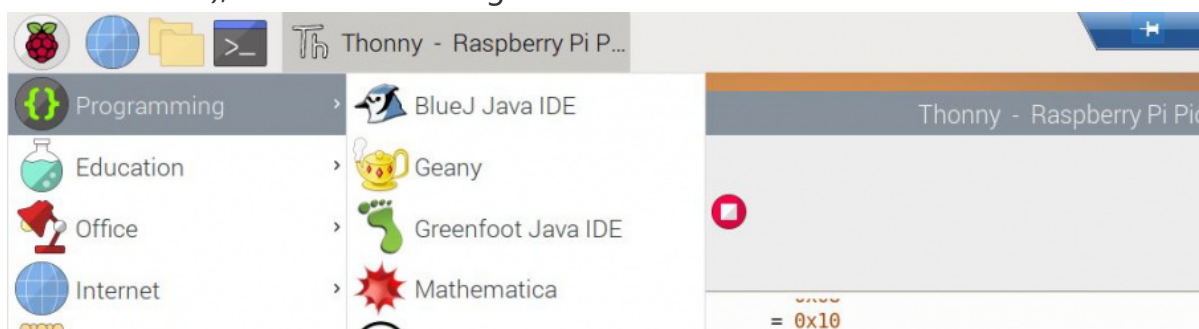
## Raspberry Pi

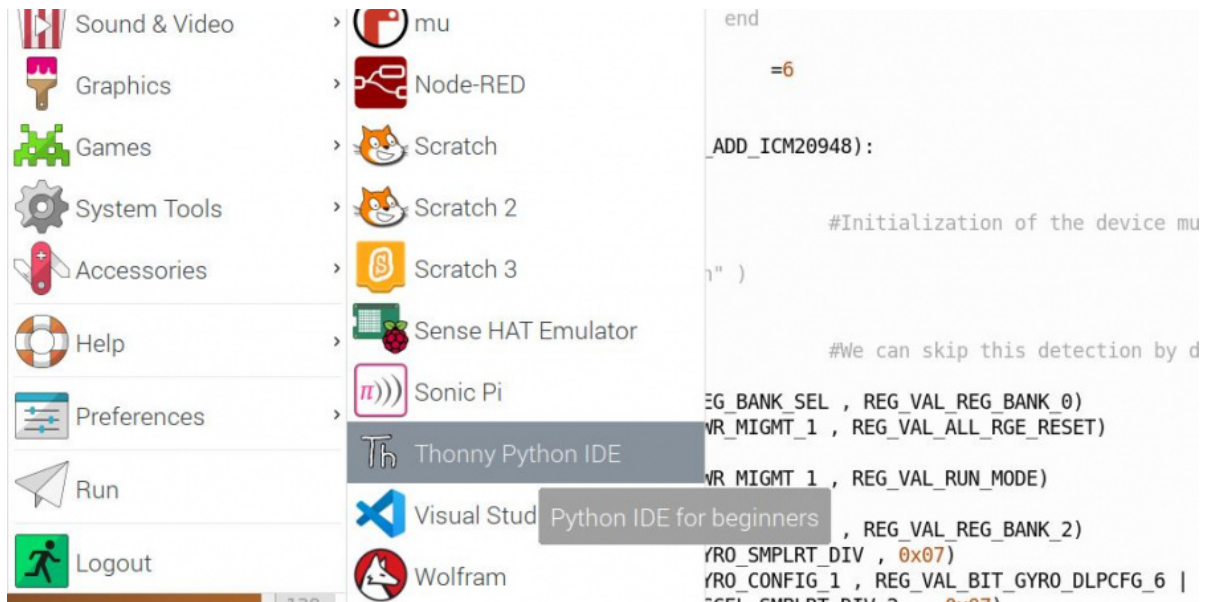
1. Use [SSH to log in to the Raspberry Pi terminal](#) or press Ctrl+Alt+T while using the screen to open the terminal.
2. Download and decompress the demo to the Pico C/C++ SDK directory, and the users who have not installed SDK can refer to [tutorial](#).

```
#Pay attention to your own directory, the default installation directory of the
SDK installation tutorial is ~/pico/, if the user customizes other directories,
change it by yourself
wget https://files.waveshare.com/upload/d/db/LCD1602_I2C_Module_code.zip
unzip LCD1602_I2C_Module_code.zip
cd LCD1602_I2C_Module_code/Pico
```

## micro python

1. Configure settings according to [the python environment](#), using the Raspberry Pi desktop system.
2. Open the Thonny IDE of the Raspberry Pi system, pull the code file into the IDE, and save LCD1602.py to the Pico file system (refer to the Window operation module below), as shown in the figure.

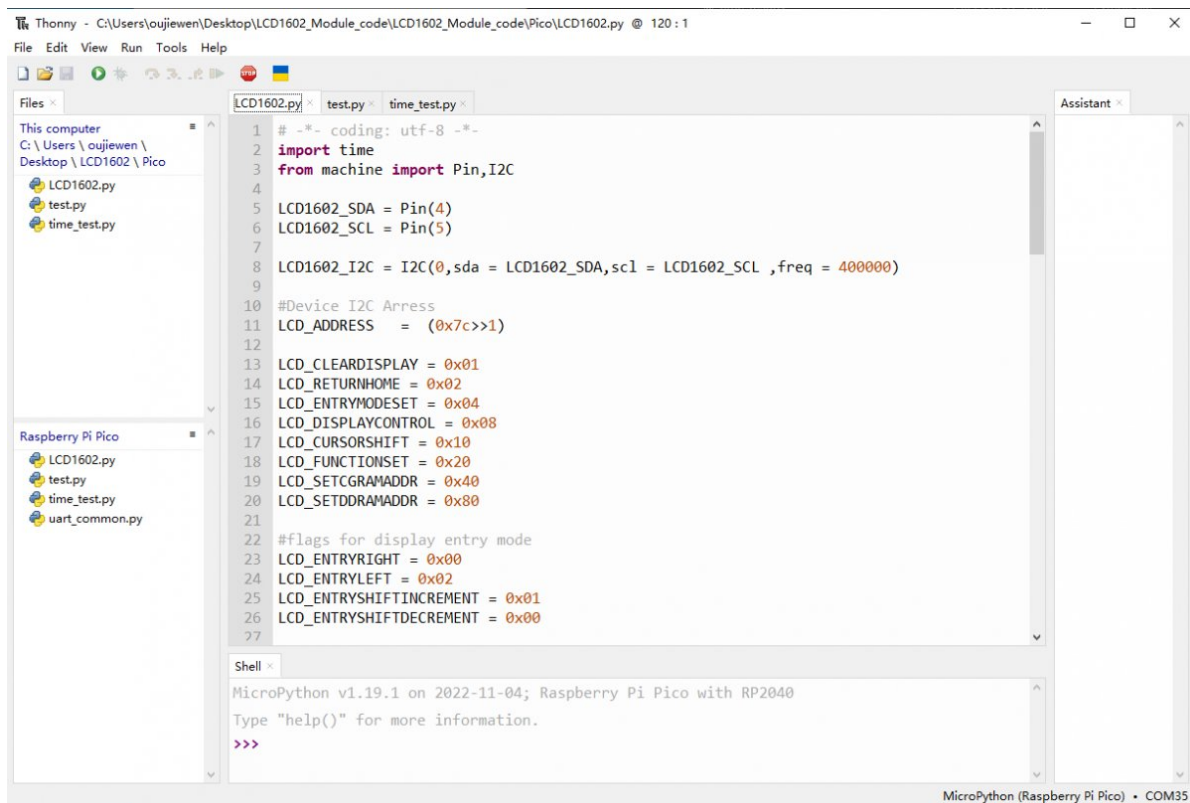




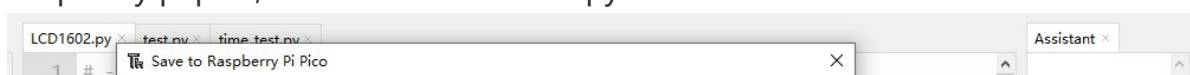
## Windows

### micro python

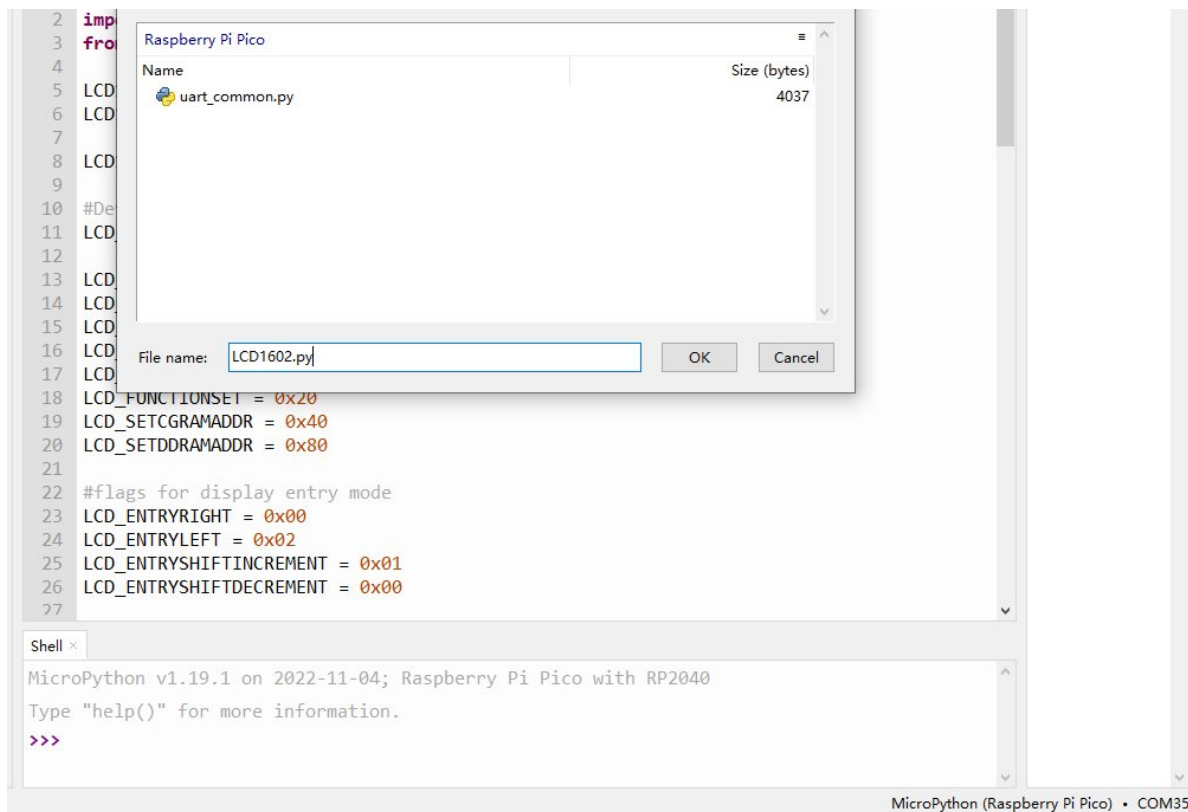
1. Download and decompress the sample program to the Windows desktop, open the pico corresponding file, refer to the [Windows software environment settings](#).
2. Open the downloaded sample program in Thonny, as shown in the figure below.



Here you also need to save LCD1602.py to PICO, select LCD1602.py, save it as Raspberry pi pico, and name it LCD1602.py.







Finally, just run the demo under test.py or time\_test.py.

- Demo analysis:

test.py

```

lcd.setCursor(0, 0) #Set the cursor position
# print the number of seconds since reset:
lcd.printout("Waveshare") #write characters
lcd.setCursor(0, 1) #Set the cursor position to the zeroth column of the second ro
w
lcd.printout("Hello, World!")#write characters

```

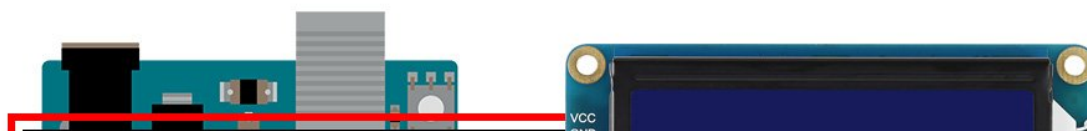
## Working With Arduino

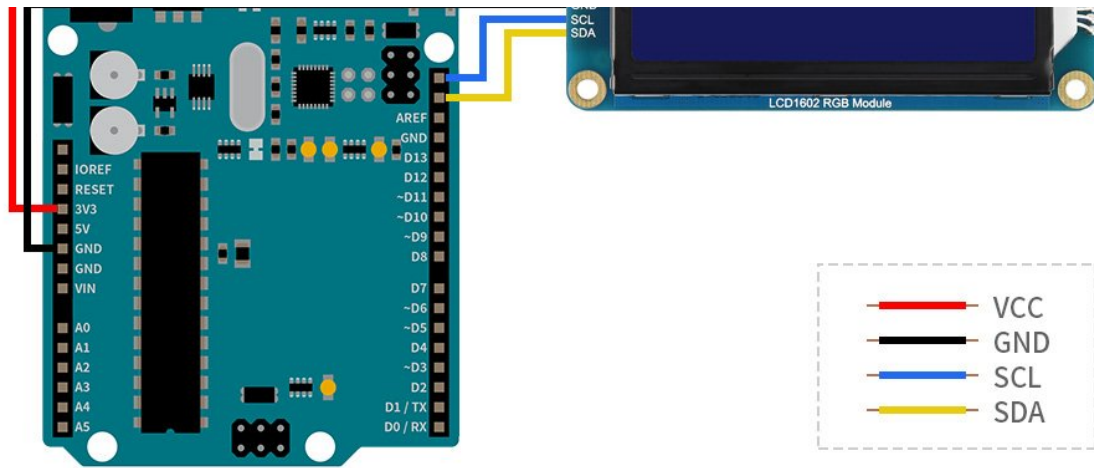
### Hardware Connection

Arduino Pin Connection Correspondence

Module Interface	Arduino Pin
VCC	5V/3.3V
GND	GND
SCL	SCL
SDA	SDA

Connection diagram:





## Arduino IDE

Download this sample demo in the Resource, enter the Arduino folder, and double-click to open the test.ino file, Arduino IDE will automatically load the driver library under the same folder, select the correct driver board model and its corresponding port number, click the upload button to automatically start compiling and uploading the demo, and the demo will run automatically after the upload is successful.



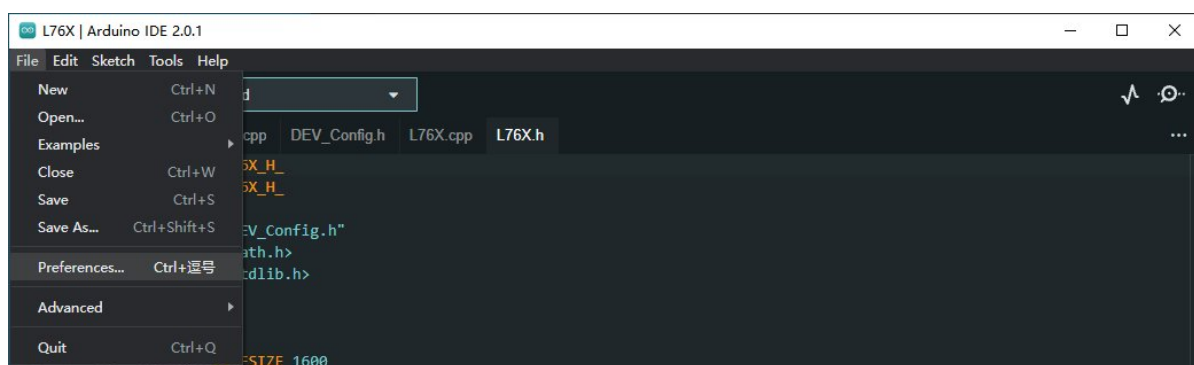
This demo has been tested and run on Arduino uno development board and Arduino mega2560 development board.

## Working With ESP32

### Environment Setting

#### Install the ESP32 Plug-in in the Arduino IDE

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





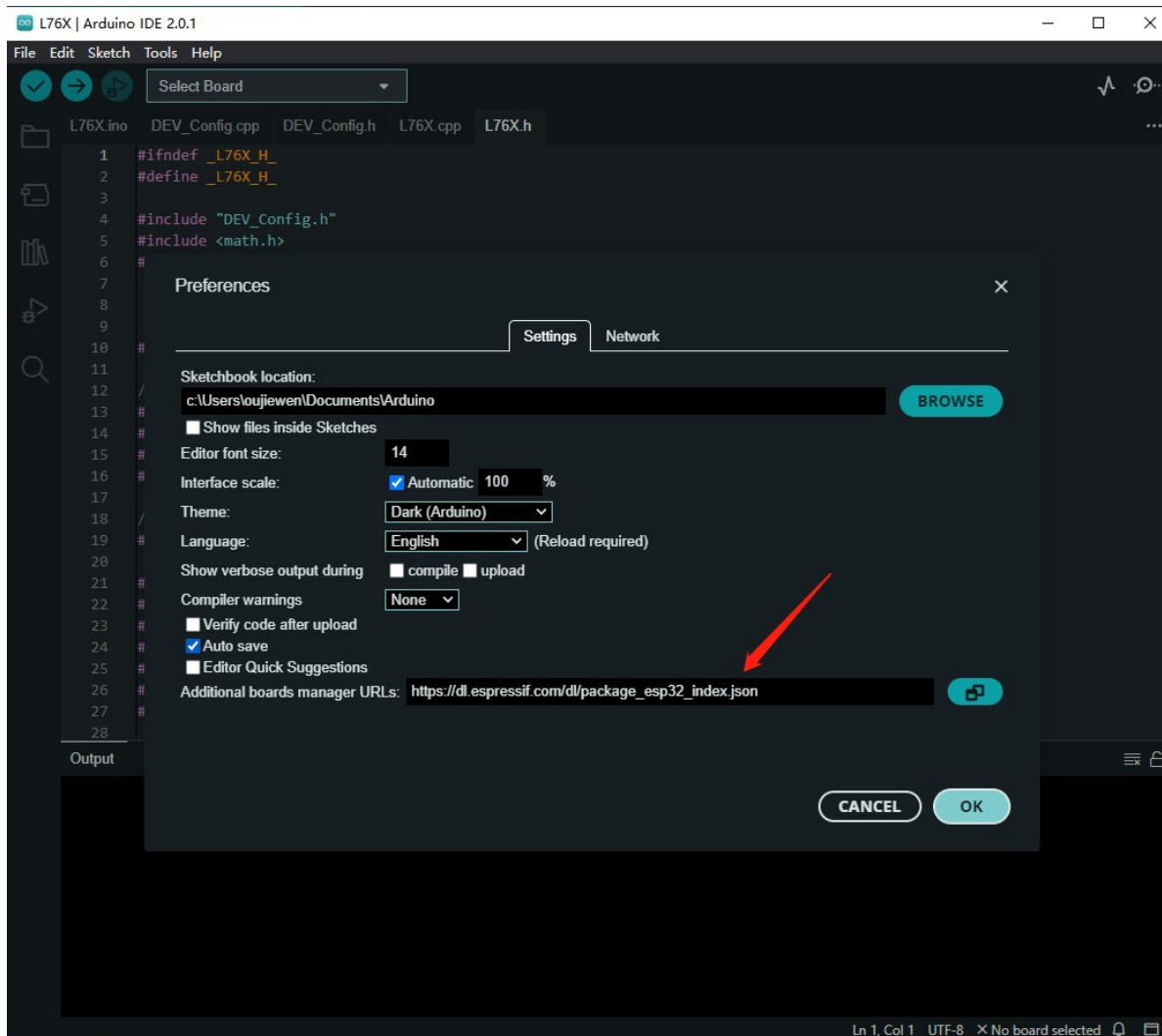
```
11
12 //Startup mode
13 #define HOT_START "$PMTK101"
14 #define WARM_START "$PMTK102"
15 #define COLD_START "$PMTK103"
16 #define FULL_COLD_START "$PMTK104"
17
18 //Standby mode -- Exit requires high level trigger
19 #define SET_PERPETUAL_STANDBY_MODE "$PMTK161"
20
21 #define SET_PERIODIC_MODE "$PMTK225"
22 #define SET_NORMAL_MODE "$PMTK225,0"
23 #define SET_PERIODIC_BACKUP_MODE "$PMTK225,1,1000,2000"
24 #define SET_PERIODIC_STANDBY_MODE "$PMTK225,2,1000,2000"
25 #define SET_PERPETUAL_BACKUP_MODE "$PMTK225,4"
26 #define SET_ALWAYSLOCATE_STANDBY_MODE "$PMTK225,8"
27 #define SET_ALWAYSLOCATE_BACKUP_MODE "$PMTK225,9"
28
```

Output

Ln 1, Col 1 UTF-8 X No board selected

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

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](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
```

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

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

此电脑 > 本地磁盘 (C:) > 用户 > xutong > AppData > Local > Arduino15



名称	修改日期	类型	大小
cache	2022/8/25 11:13	文件夹	
packages	2022/8/26 16:07	文件夹	
staging	2022/8/26 16:06	文件夹	
library_index.json	2022/8/26 15:43	JSON 源文件	26,581 KB
library_index.json.sig	2022/8/26 15:43	SIG 文件	1 KB
package_esp32_index.json	2022/8/26 16:36	JSON 源文件	24 KB
package_index.json	2022/8/26 16:36	JSON 源文件	525 KB
package_index.json.sig	2022/8/26 16:36	SIG 文件	1 KB
preferences.txt	2022/8/26 15:08	文本文档	3 KB

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

## Hardware Connection

When connecting to ESP32, choose to connect with a 4PIN cable, please refer to the pin correspondence table below:

If you are using a pin header or PH2.0 4PIN interface, you need to connect according to the following table.

ESP32 Pin Connection Correspondence

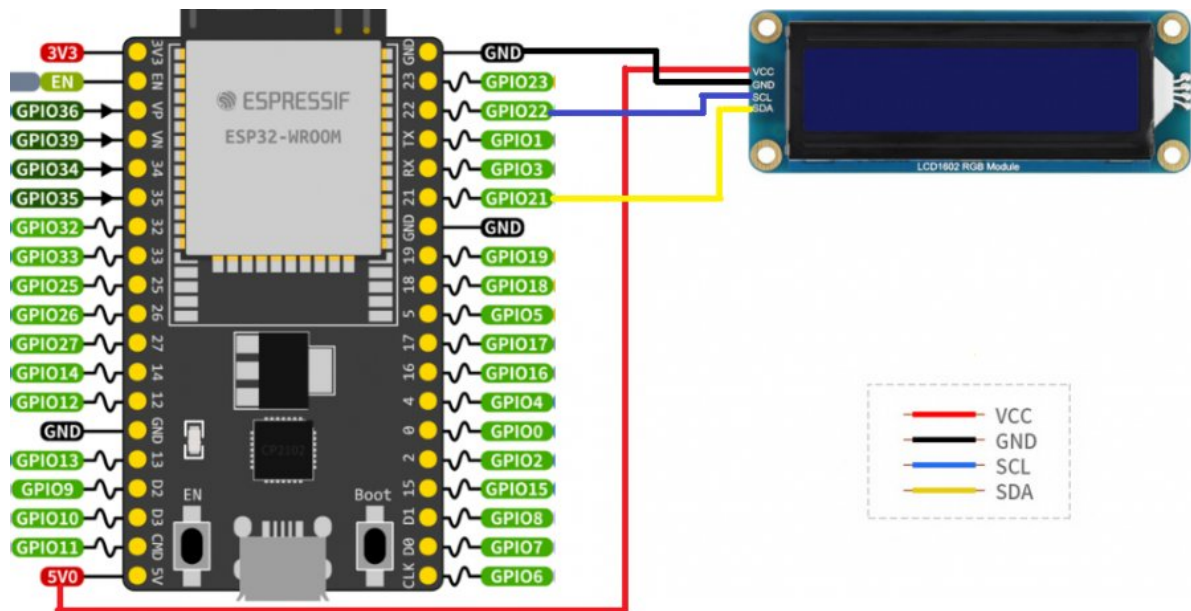
LCD	ESP32
	Pin No.
VCC	5V/3.3V
GND	GND
SCL	GPIO22
SDA	GPIO21

Take the LCD1602 Module using the PH2.0 4PIN interface as an example, and connect it to the ESP32 according to the above table:

**(Please connect according to the pin definition table. The color of the wiring in the picture is for reference only, and the actual color shall prevail.)**

Connect to ESP32 as shown in the figure below.





## Use With Arduino IDE

Download this sample demo in the Resource, enter the ESP32 folder, double-click to open the test.ino file, Arduino IDE will automatically load the driver library under the same folder, select the correct driver board model and its corresponding port number, click the upload button to automatically Start compiling and uploading the demo, and the demo will run automatically after the upload is successful.

## Working With Jetson Nano

### Hardware Connection

When connecting Jetson nano, choose to connect with 4PIN cable, please refer to the pin correspondence table below:

If you are using a pin header or PH2.0 4PIN interface, you need to connect according to the following table:

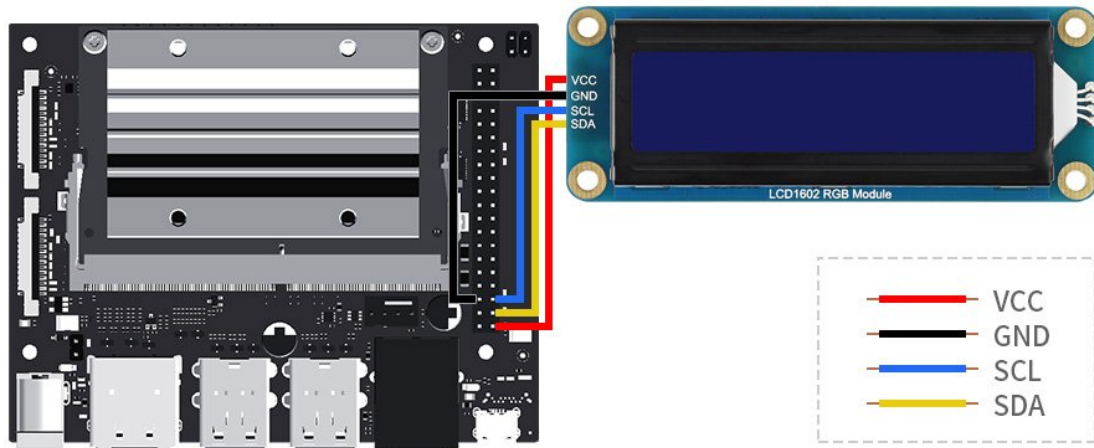
Jetson nano Pin Connection Correspondence

LCD	Jetson nano	
	BCM2835 Pin	Board Pin No.
VCC	3.3V	3.3V/5V
GND	GND	GND
SCL	SCL.1	5
SDA	SDA.1	3

Take the LCD1602 Module using the PH2.0 4PIN interface as an example, and connect it to the Jetson nano according to the above table:

(Please connect according to the pin definition table. The color of the wiring in the picture is for reference only, and the actual color shall prevail.)

Connect to Jetson nano as shown below:



## Sample Demo

```
cd ~
wget https://files.waveshare.com/upload/d/db/LCD1602_I2C_Module_code.zip
unzip LCD1602_I2C_Module_code.zip
cd ~/LCD1602_I2C_Module_code/Jetson\ Nano
sudo chmod 777 *
cd python/
sudo python time_test.py #Automatically obtains the local time and displays it o
n the LCD
```

## Resource

### Demo

- [Demo](#)

### Datasheet

- [LCD1602\\_I2C\\_Module.pdf](#)
- [AIP31068L](#)

## FAQ

**Question:**When using the python sample demo, why only the second line is displayed but the first line is not displayed?

**Answer:**

It may be a problem of python firmware incompatibility, you can use the firmware version given by our sample demo.

**Question:**Do SDA and SCL of LCD1602 have pull-up resistors?

**Answer:**

Yes.

## 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](#)