SLC1602M Series
LCD MODULE USER MANUAL

Please click the following image to buy the sample

Business Card | WhatsAPP | 카카오톡 | LINE (ライン) | WeChat

Shenzhen Surenoo Technology Co.,Ltd.
www.surenoo.com
Skype: Surenoo365

Reference Controller Datasheet

Character LCD Selection Guide

AiP31068L

Test Code

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/5.0V

◆ Communication interface: I2C

◆ Screen type: LCD screen

◆ Control chip: AiP31068L

◆ Slave address: 0X7C

◆ Display dimensions: 64.5 x 16.0 mm

◆ Product size: 87.0 x 32.0 x 13.0(mm)

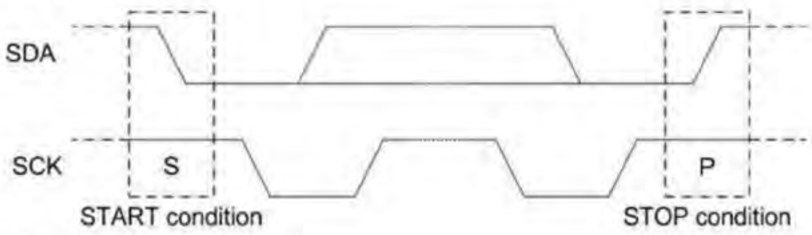
◆ Operating current: 26mA(5V), 13mA (3.3V)

>> Interface Description

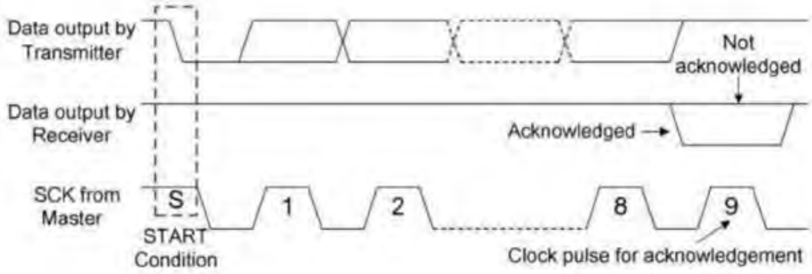
Pin	Description
VCC	3.3V/5V power input
GND	Ground
SCL	I2C clock cable
SDA	I2C data cable

>> Communication Protocol

START STOP:



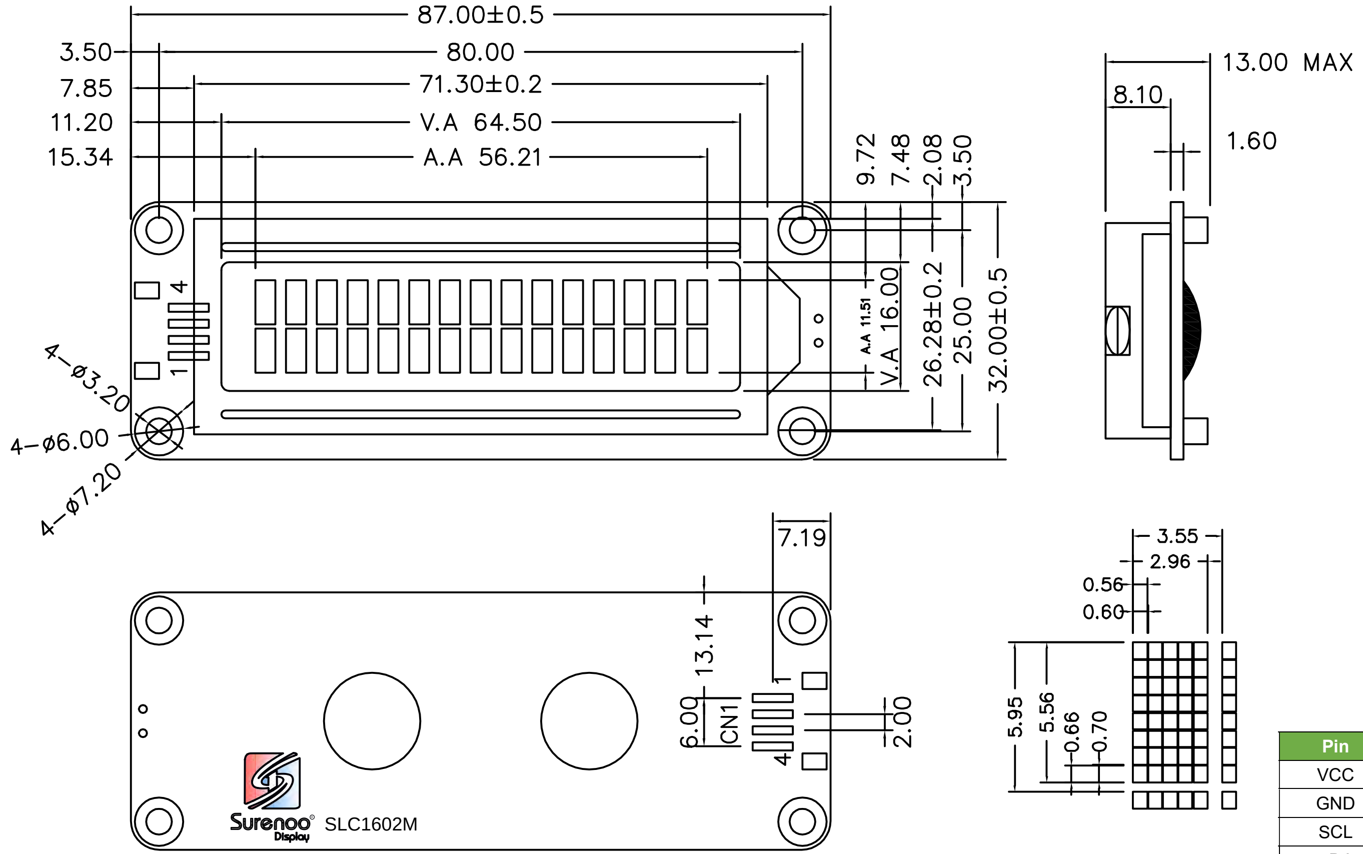
ASK :



SHENZHEN SURENOO TECHNOLOGY CO.,LTD.
深圳市襄诺科技有限公司

>> Outline Dimension

Customer approval :



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

UNIT: mm

Specifications:
1. Display Mode: STN Yellow Green / STN Blue Negative / FSTN Positive
2. Viewing Angle: 6:00 O'clock.
3. Drive Method: 1/16 Duty, 1/5 Bias.
4. Backlight Type: 1PCS LED
5. Top: −20℃~+70℃, Tst: −30℃~+80℃.
6. Drive IC: AIP31068L

REV	REVISION RECORD	DATE	SHENZHEN SURENOO TECHNOLOGY CO.,LTD. 深圳市襄诺科技有限公司				
A	FIRST ISSUE	23/03/08					
B			MODEL NO. 模组型号	Model No.: SLC1602M Series		GENERAL TOL: 未 注 公 差	± 0.3
C			APPROVED BY 审核	CHECKED BY 检查	DRAWING BY 绘图	PAGE: 页面1 OF 1	UNIT: 单位mm
D			Liu 2023/03/08	Francis 2023/03/08	Luoxian 2023/03/08	PROJECTION 三 角 法	

Surenoo Display

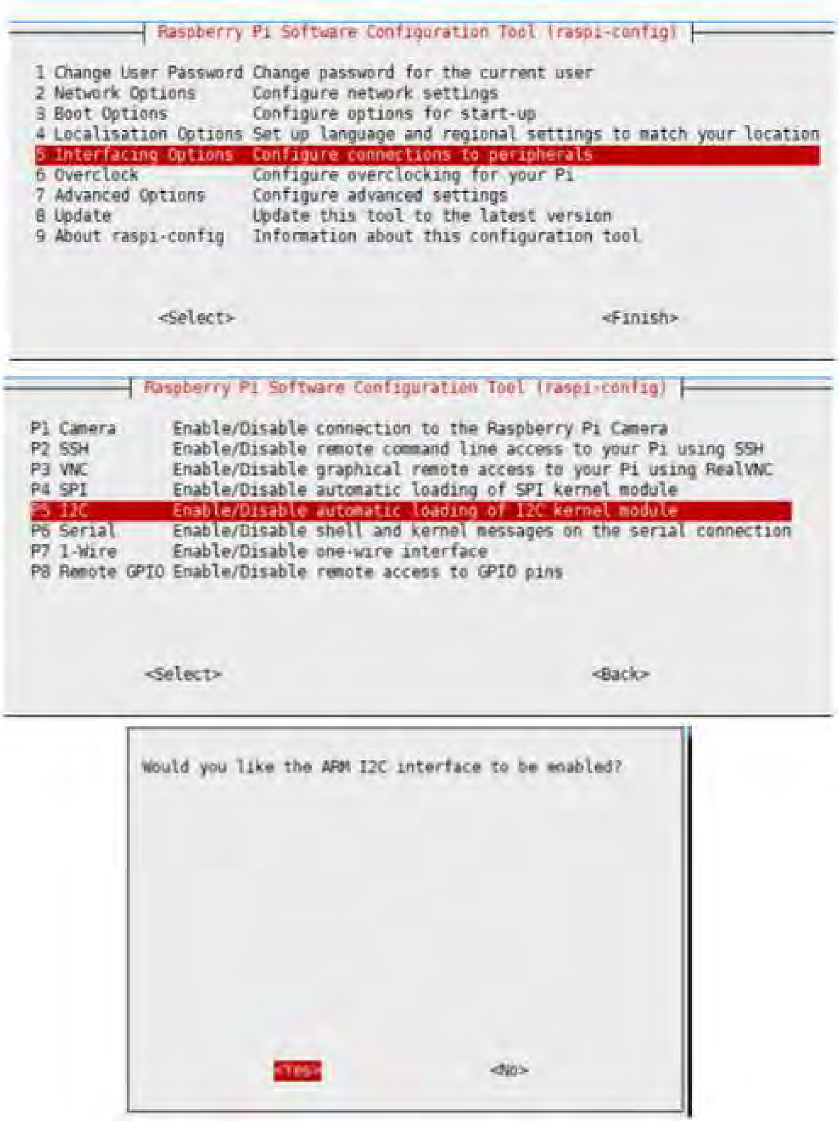
Working with Raspberry Pi

>> 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
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

          <Select>                              <Back>
```

```
Would you like the ARM I2C interface to be enabled?

          <Yes>                  <No>
```

And then reboot the Raspberry Pi.

```
sudo reboot
```

◆ **Hardware Connection**

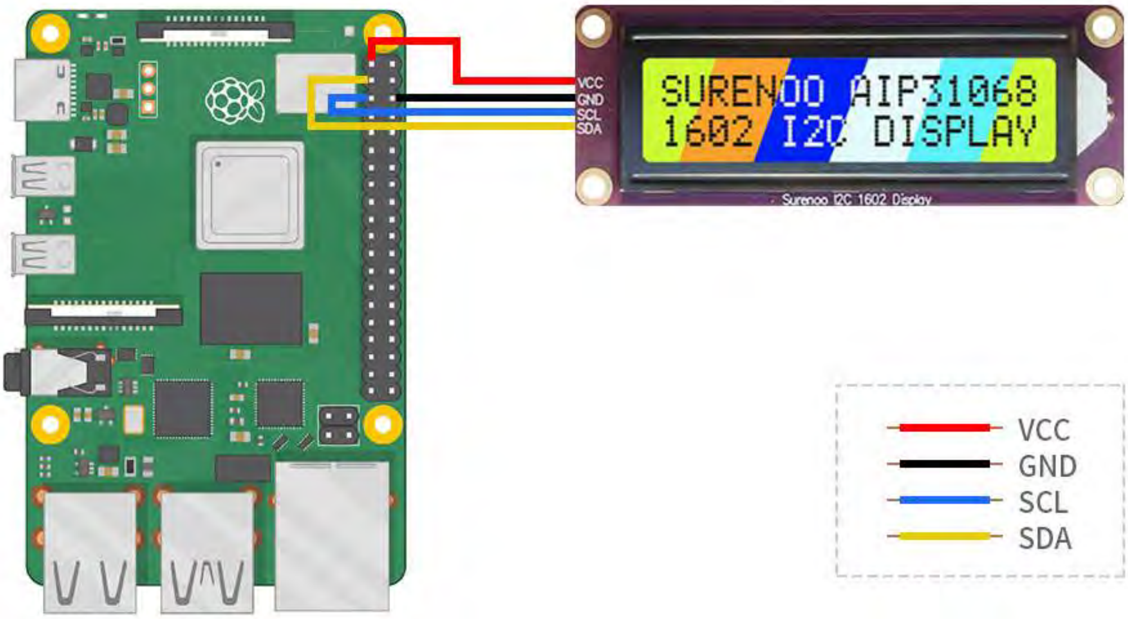
When connecting the Raspberry Pi, choose to connect with a 4PIN cable, please refer to the pin correspondence table below.

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

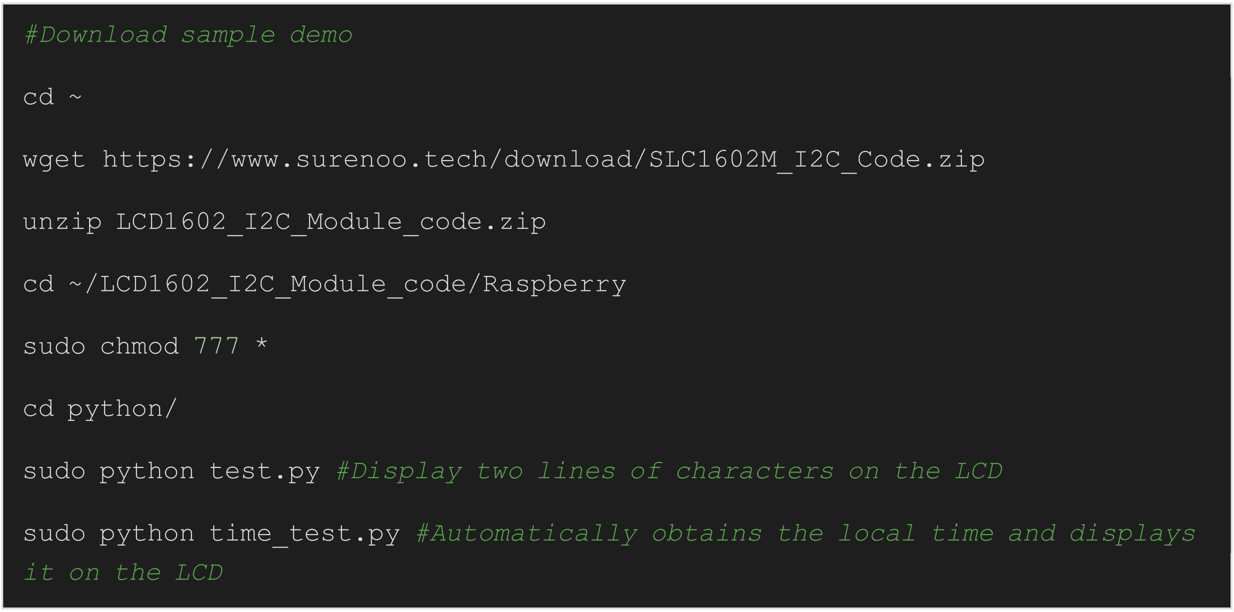
Raspberry Pi Pin Connection Correspondence		
LCD	**Raspberry Pi**	
	BCM2835	**Board Pin No.**
VCC	3.3V	3.3V
GND	GND	GND
SCL	SCL.1	5
SDA	SDA.1	3

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.)

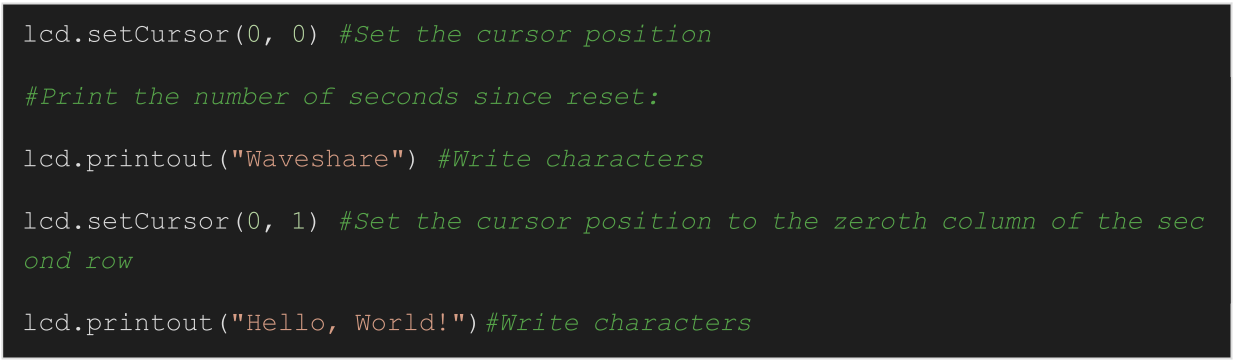


>> Example Demo



```
#Download sample demo

cd ~

wget https://www.surenoo.tech/download/SLC1602M_I2C_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)



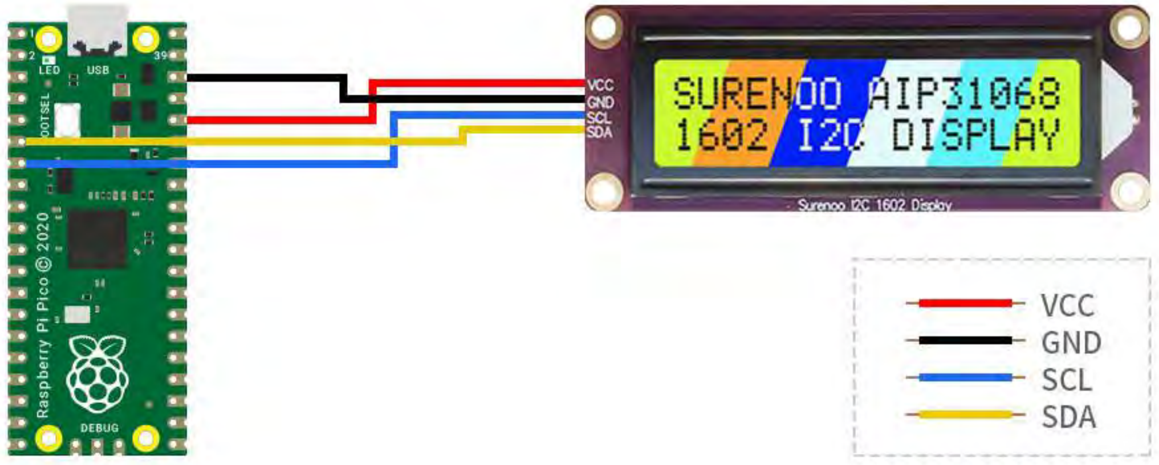
```
lcd.setCursor(0, 0) #Set the cursor position

#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 sec
ond row

lcd.printout("Hello, World!")#Write characters
```

Working with Pico

>> 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.

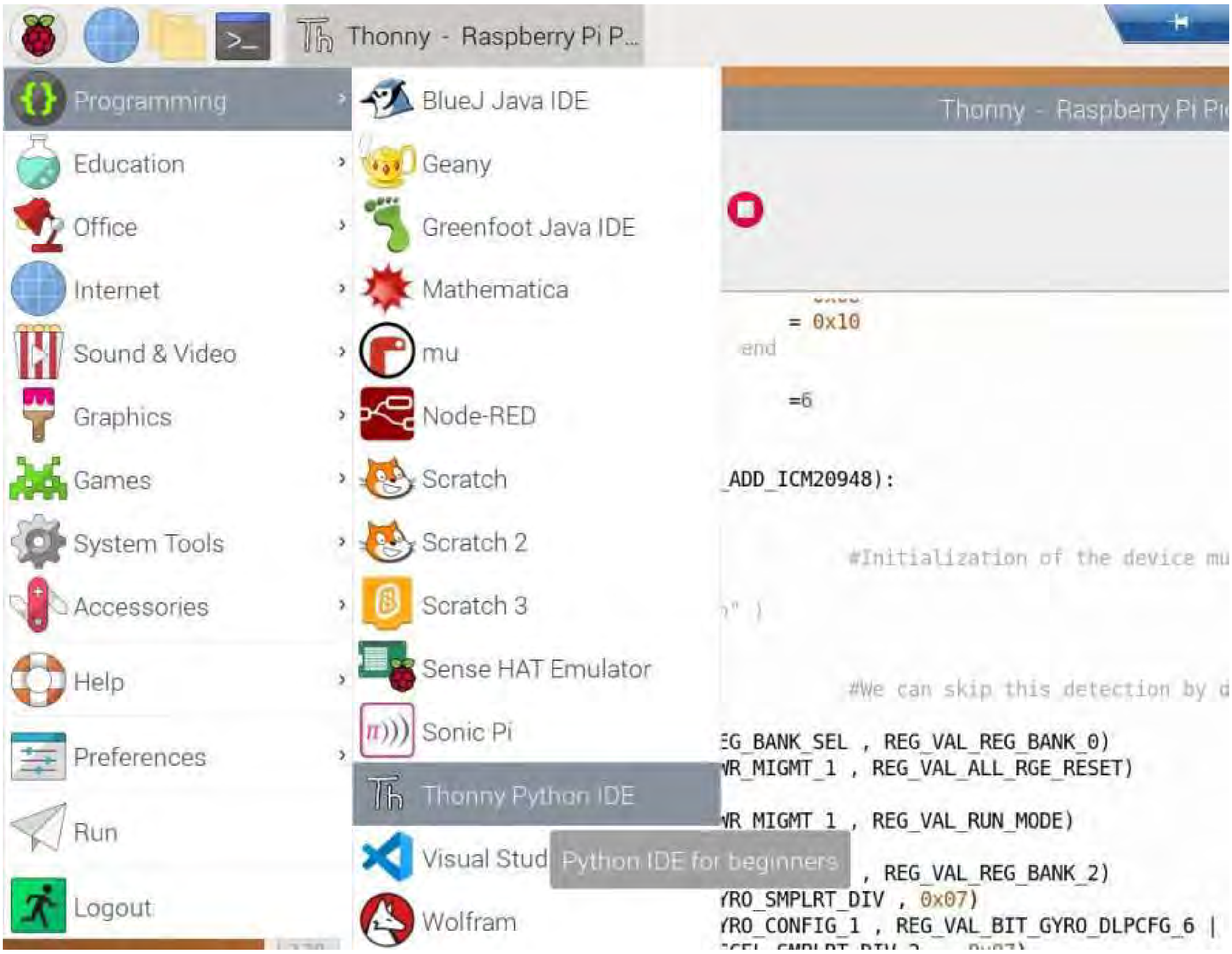
2. Download and decompress the demo to the Pico C/C++ SDK directory, and the users who have not installed SDK can refer to tutorial.

```
#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://www.surenoo.tech/download/SLC1602M_I2C_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.



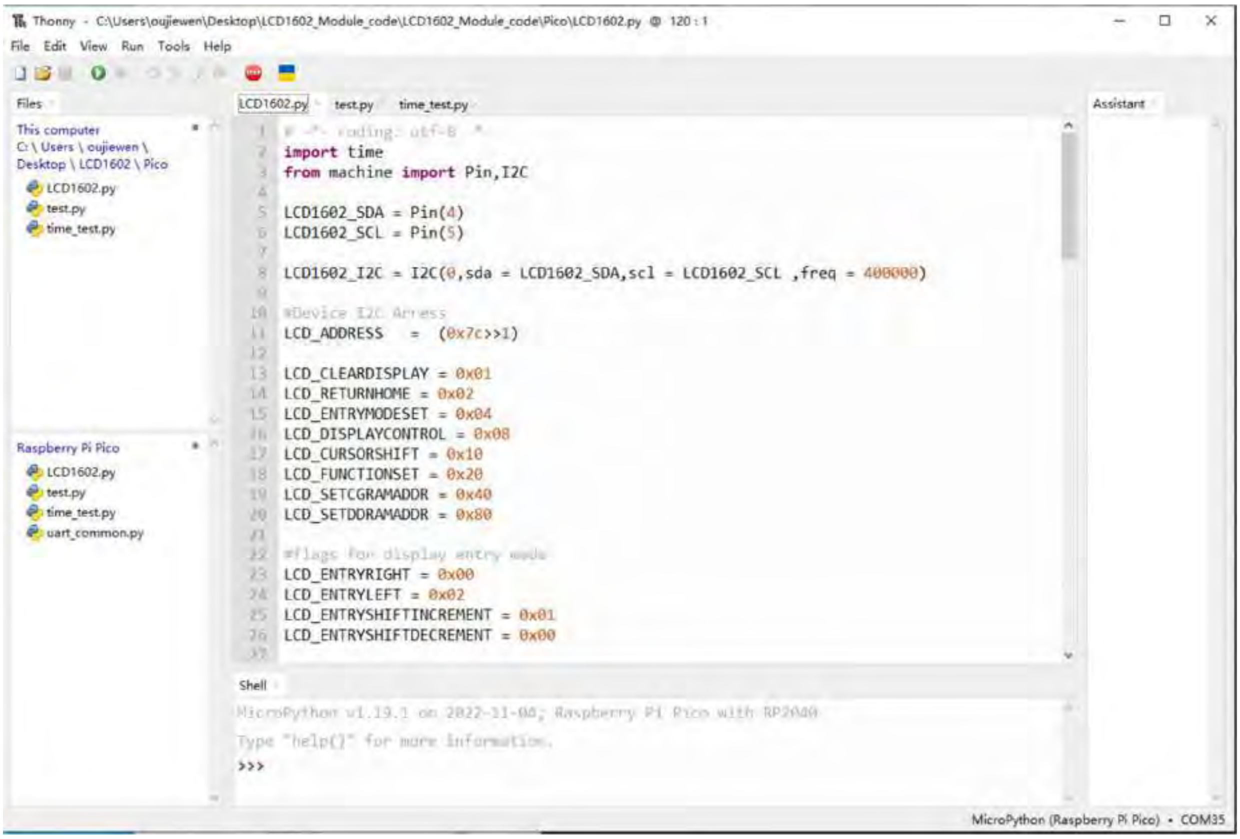
```
        = 0x10
 end
        =6
_ADD_ICM20948):
                    #Initialization of the device mu
)" )
                    #We can skip this detection by d
EG_BANK_SEL , REG_VAL_REG_BANK_0)
WR_MIGMT_1 , REG_VAL_ALL_RGE_RESET)
WR_MIGMT_1 , REG_VAL_RUN_MODE)
             , REG_VAL_REG_BANK_2)
YRO_SMPLRT_DIV , 0x07)
YRO_CONFIG_1 , REG_VAL_BIT_GYRO_DLPCFG_6 |
```

>> 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.

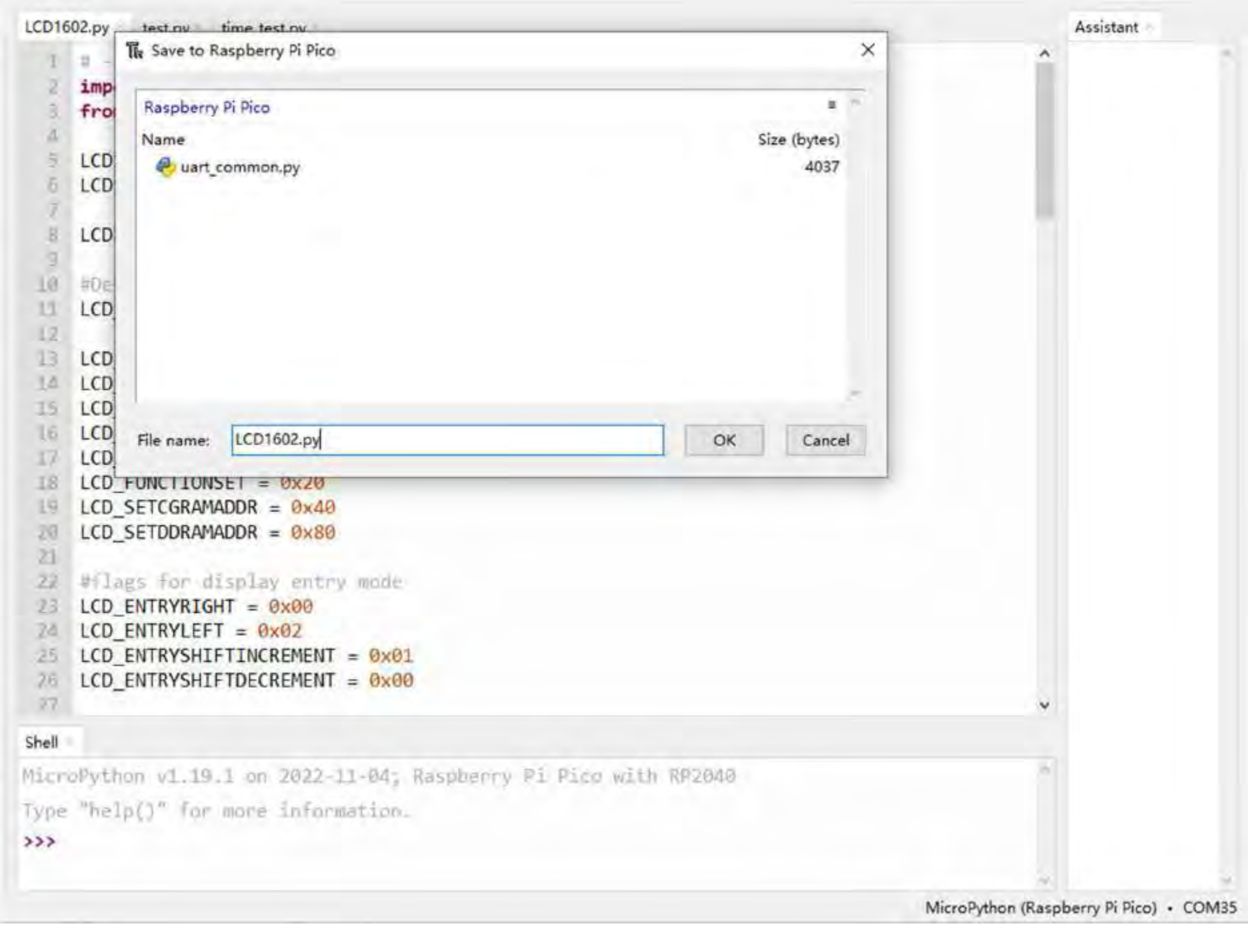
2. Open the downloaded sample program in Thonnty, as shown in the figure below.


Thonny - C:\Users\oujiewen\Desktop\LCD1602_Module_code\LCD1602_Module_code\Pico\LCD1602.py @ 120 : 1
File
Edit
View
Run
Tools
Help
Files
This computer
C: \ Users \ oujiewen \ Desktop \ LCD1602 \ Pico
LCD1602.py
test.py
time_test.py
Raspberry Pi Pico
LCD1602.py
test.py
time_test.py
uart_common.py
LCD1602.py
test.py
time_test.py
Assistant
Shell
MicroPython (Raspberry Pi Pico) • COM35

```python
# -*- coding: utf-8 -*
import time
from machine import Pin,I2C

LCD1602_SDA = Pin(4)
LCD1602_SCL = Pin(5)

LCD1602_I2C = I2C(0,sda = LCD1602_SDA,scl = LCD1602_SCL ,freq = 400000)

#Device I2C Arress
LCD_ADDRESS   =  (0x7c>>1)

LCD_CLEARDISPLAY = 0x01
LCD_RETURNHOME = 0x02
LCD_ENTRYMODESET = 0x04
LCD_DISPLAYCONTROL = 0x08
LCD_CURSORSHIFT = 0x10
LCD_FUNCTIONSET = 0x20
LCD_SETCGRAMADDR = 0x40
LCD_SETDDRAMADDR = 0x80

#flags for display entry mode
LCD_ENTRYRIGHT = 0x00
LCD_ENTRYLEFT = 0x02
LCD_ENTRYSHIFTINCREMENT = 0x01
LCD_ENTRYSHIFTDECREMENT = 0x00
```

```
MicroPython v1.19.1 on 2022-11-08; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

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



uart_common.py 4037

File name: LCD1602.py

```
LCD_FUNCTIONSET = 0x20
LCD_SETCGRAMADDR = 0x40
LCD_SETDDRAMADDR = 0x80

#flags for display entry mode
LCD_ENTRYRIGHT = 0x00
LCD_ENTRYLEFT = 0x02
LCD_ENTRYSHIFTINCREMENT = 0x01
LCD_ENTRYSHIFTDECREMENT = 0x00
```

```
MicroPython v1.19.1 on 2022-11-04; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

Finally, just run the demo under test.py or time_test.py.

- **Demo analysis: (test.py)**

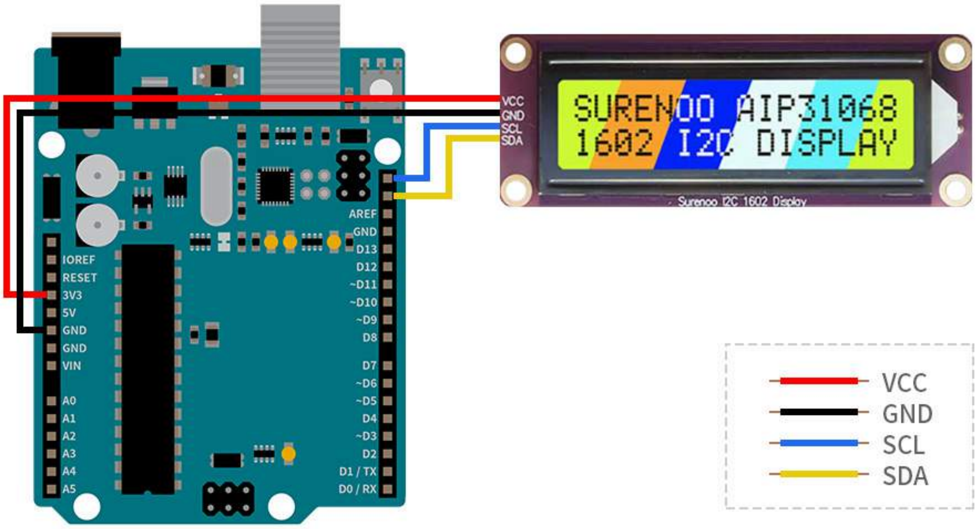
```
lcd.setCursor(0, 0) #Set the cursor position

# 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 row

lcd.printout("Hello, World!")#write characters
```

Working with Arduino

>> Hardware Connection

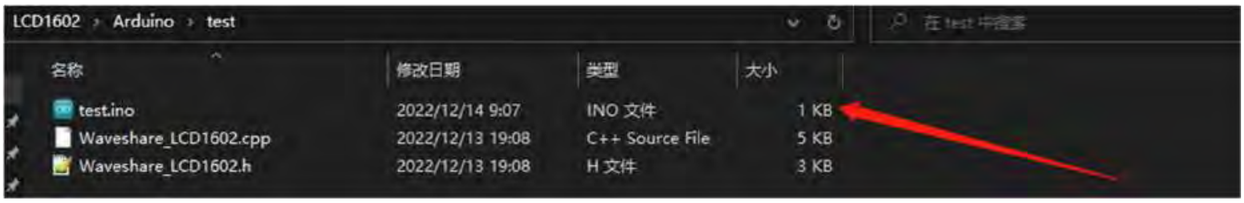
Arduino Pin Connection Correspondence	
Module Interface	Arduino Pin
VCC	5V/3.3V
GND	GND
SCL	SCL
SDA	SDA

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.



名称	修改日期	类型	大小
test.ino	2022/12/14 9:07	INO 文件	1 KB
Waveshare_LCD1602.cpp	2022/12/13 19:08	C++ Source File	5 KB
Waveshare_LCD1602.h	2022/12/13 19:08	H 文件	3 KB

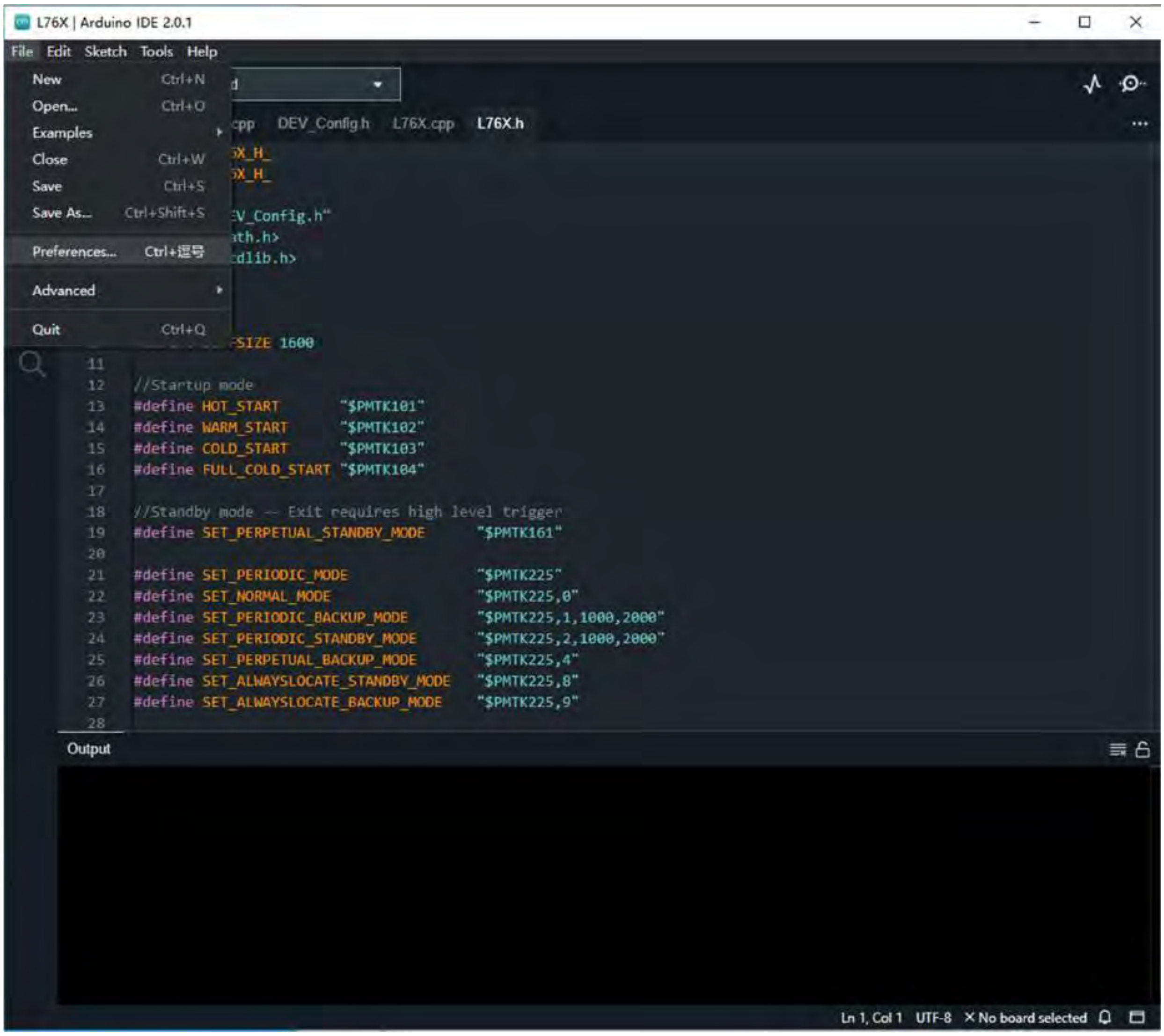
This demo has been tested and run on Arduino uno development board and Arduino mega2560 development board.

Working with ESP32

>> Environment Setting

◆ **Install the ESP32 Plug-in in the Arduino IDE**

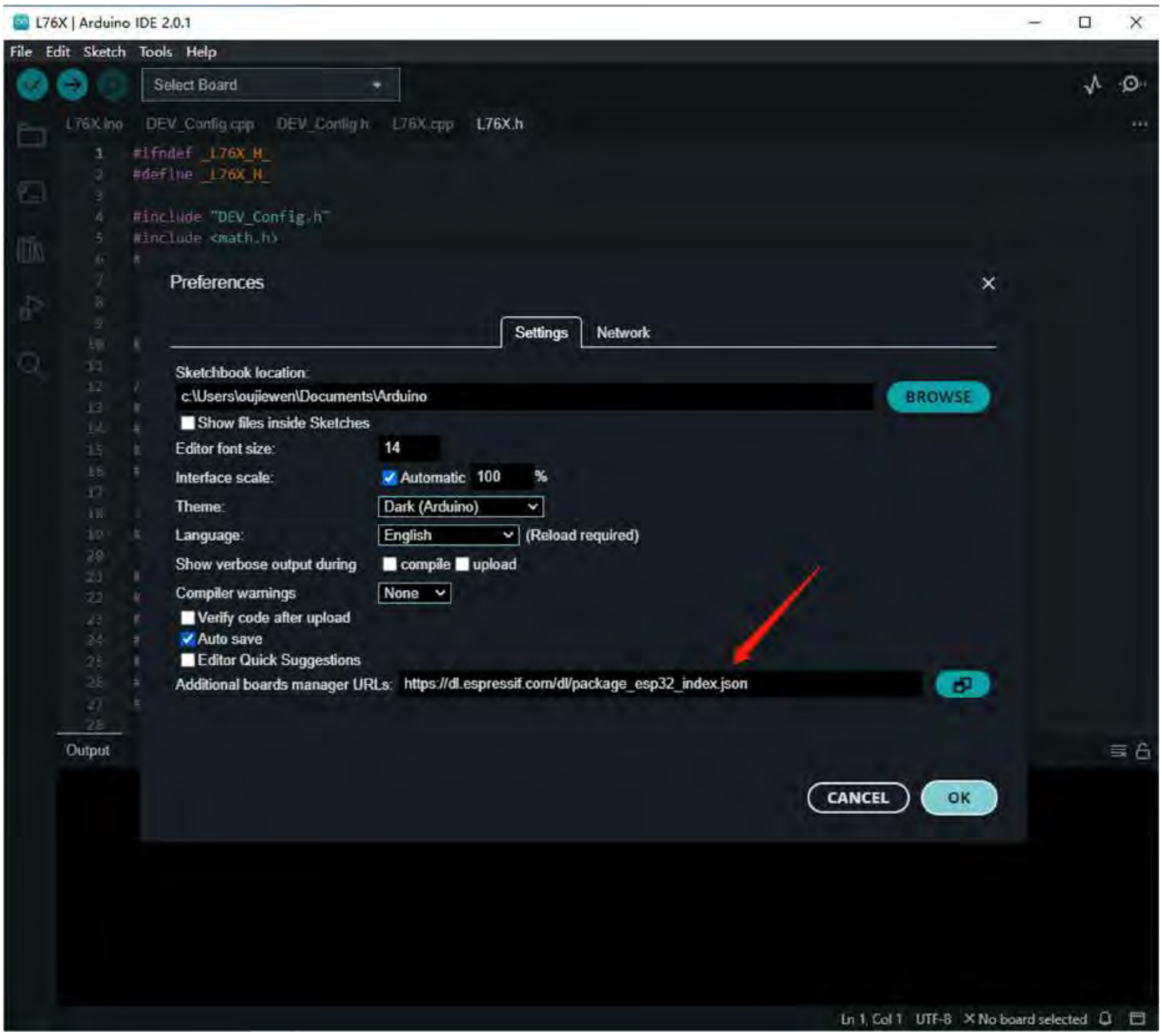
1. Open the Arduino IDE, click on the file in the upper left corner, select Preferences:



```
X_H_
X_H_

V_Config.h"
th.h>
dlib.h>



SIZE 1600

//Startup mode
#define HOT_START       "$PMTK101"
#define WARM_START      "$PMTK102"
#define COLD_START      "$PMTK103"
#define FULL_COLD_START "$PMTK104"

//Standby mode -- Exit requires high level trigger
#define SET_PERPETUAL_STANDBY_MODE      "$PMTK161"

#define SET_PERIODIC_MODE               "$PMTK225"
#define SET_NORMAL_MODE                 "$PMTK225,0"
#define SET_PERIODIC_BACKUP_MODE        "$PMTK225,1,1000,2000"
#define SET_PERIODIC_STANDBY_MODE       "$PMTK225,2,1000,2000"
#define SET_PERPETUAL_BACKUP_MODE       "$PMTK225,4"
#define SET_ALWAYSLOCATE_STANDBY_MODE   "$PMTK225,8"
#define SET_ALWAYSLOCATE_BACKUP_MODE    "$PMTK225,9"
```

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

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


L76X | Arduino IDE 2.0.1
File Edit Sketch Tools Help
Select Board
L76X.ino DEV_Config.cpp DEV_Config.h L76X.cpp L76X.h
Output
Ln 1, Col 1 UTF-8 × No board selected
BROWSE
CANCEL
OK

```
#ifndef _L76X_H_
#define _L76X_H_

#include "DEV_Config.h"
#include <math.h>
```

Preferences

Settings Network

Sketchbook location: c:\Users\oujiewen\Documents\Arduino

□ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Dark (Arduino)

Language: English (Reload required)

Show verbose output during □ compile □ upload

Compiler warnings None

□ Verify code after upload

☑ Auto save

□ Editor Quick Suggestions

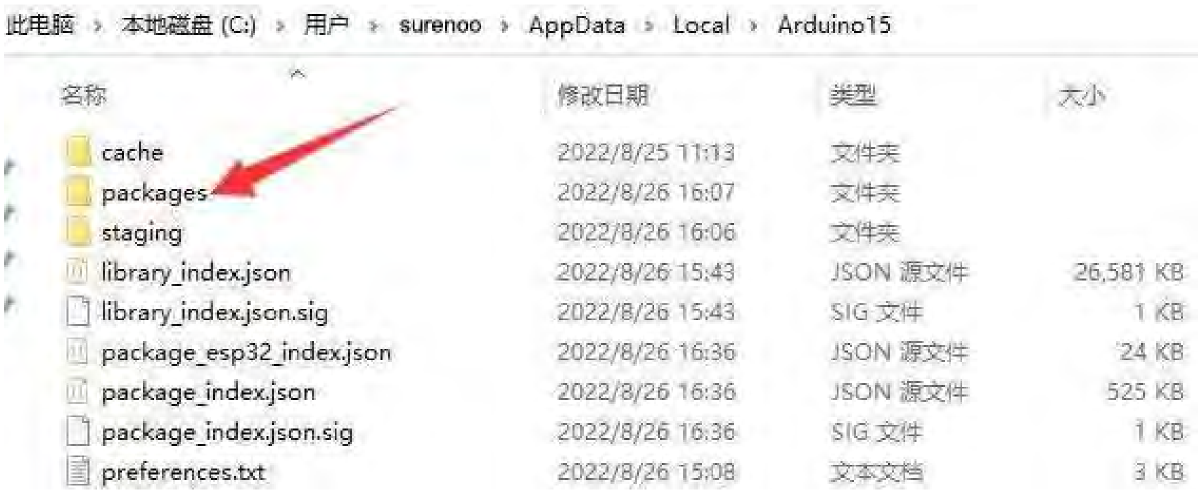
Additional boards manager URLs: https://dl.espressif.com/dl/package_esp32_index.json

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\surenoo\AppData\Local\Arduino15
```



此电脑 › 本地磁盘 (C:) › 用户 › surenoo › 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: surenoo 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:

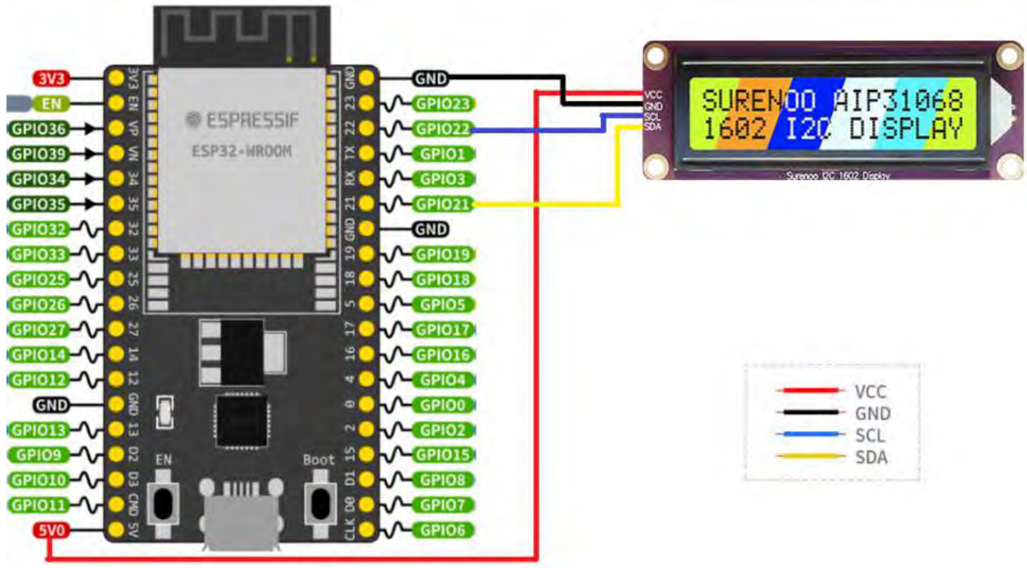
If you are using a pin header or PH2.0 4PIN interface, you need to connect according to the following table.

ESP32 Pin Connection Correspondence	
LCD Interface	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:

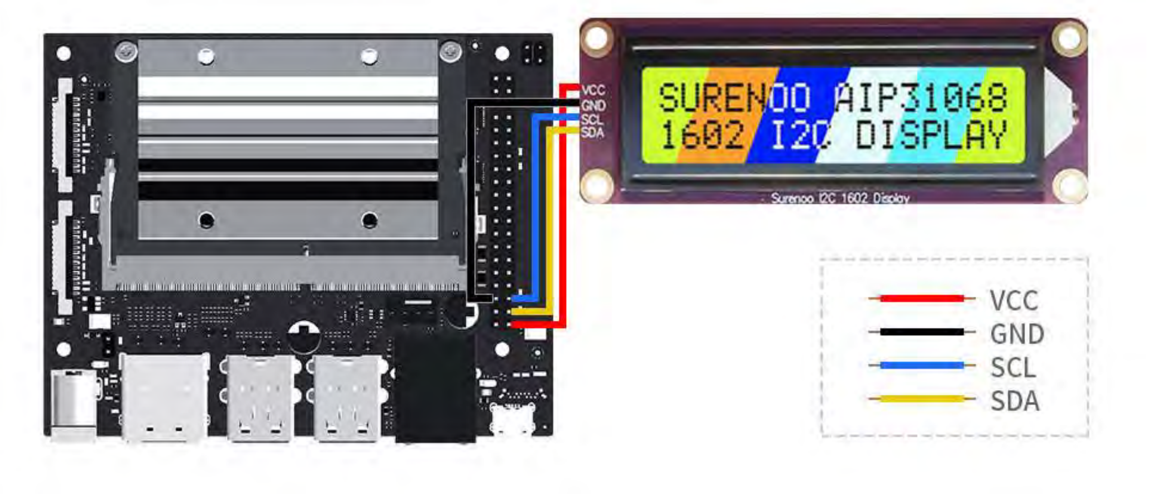
If you are using a pin header or PH2.0 4PIN interface, you need to connect according to the following table:

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://www.surenoo.tech/download/SLC1602M_I2C_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 on the LCD
```