

Overview

Features

- I2C interface can be directly controlled with only two wires, which is convenient to connect to various control boards.
- Incorporates character LCD panel LCD1602.
- Adjustable RGB backlight color, up to 16M (2563) backlight colors in theory.
- Can display up to 16 x 2 characters, support screen scrolling, cursor movement and other functions.
- Onboard AiP31068L LCD driver chip, PCA9633 RGB control chip.
- I2C control interface, only two signal pins are required, saving the IO resource.
- Compatible with 3.3V/5V operating voltage.
- Comes with development resources and manual (Raspberry Pi/Jetson Nano/Arduino examples).

LCD1602 RGB Module



LCD1602 RGB Module,
Adjustable RGB Backlight Color,
I2C Communication, 3.3V/5V
Compatible

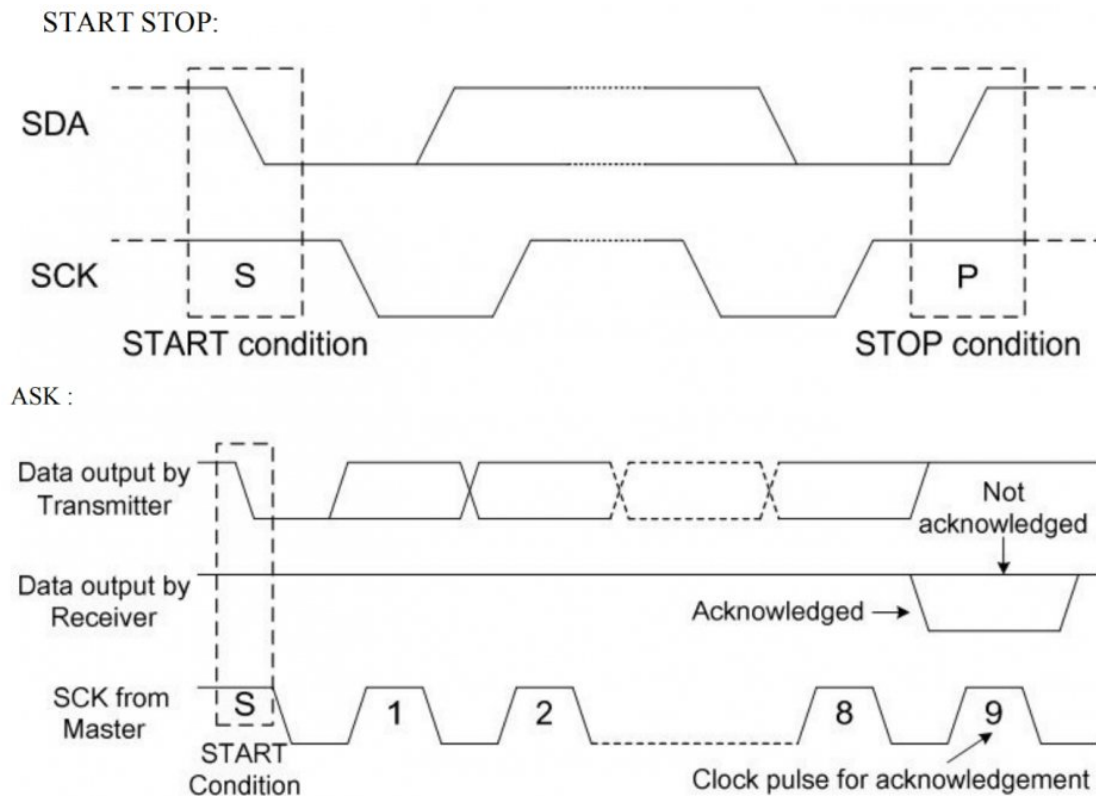
Specification

- Operating voltage: 3.3V/5V
- Interface: I2C
- LCD type: character LCD
- Controller: AiP31068L PCA9633DP2
- Display: 64.5 x 16.0 mm
- Dimension: 87.0 × 32.0 × 13.0mm
- Operating current: 26mA (5V), 13mA (3.3V)

Pinout

PIN	Description
VCC	3.3V/5V
GND	GND
SDA	I2C data pin
SCL	I2C clock pin

Working Protocol



Raspberry Pi

Hardware Configuration

Enable I2C interface

- Open the terminal, and use the command to enter the configuration page.

```
sudo raspi-config
```

Choose Interfacing Options -> I2C -> Yes to enable the I2C interface.

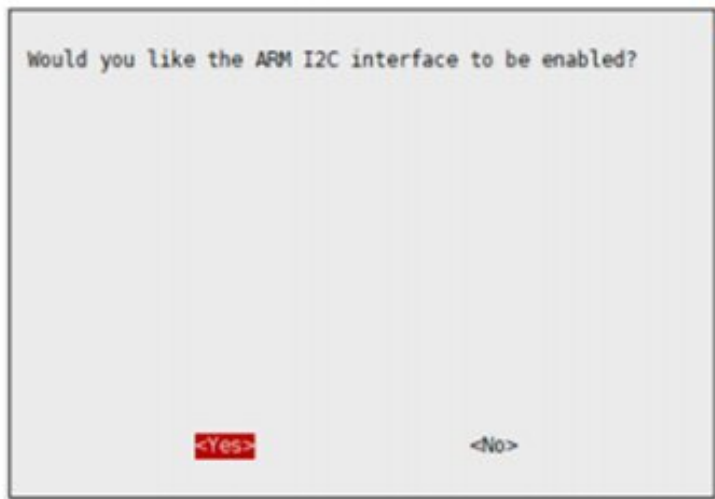
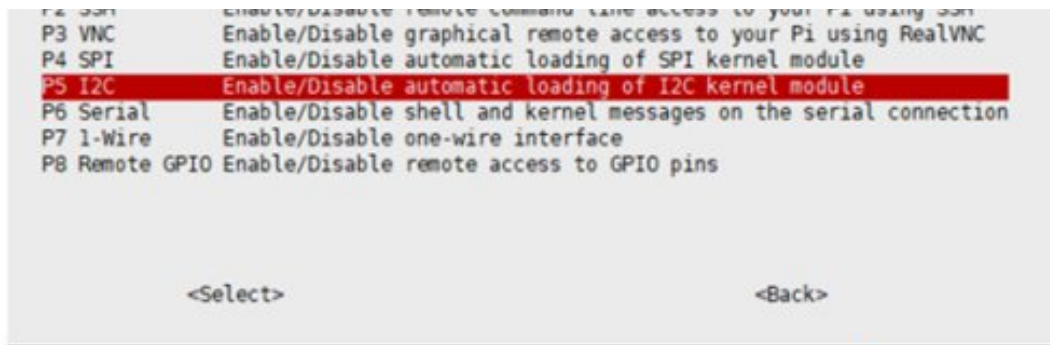
```
Raspberry Pi Software Configuration Tool (raspi-config)

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
```



And then reboot the system:

```
sudo reboot
```

Hardware Connection

You can directly attach it to the 40PIN GPIO of Raspberry Pi. Or you can wire it to Raspberry Pi with the PH2.0 4PIN interface of the Module, please refer to the Pin definition below:

LCD	Raspberry Pi	
	BCM2835	Board
VCC	3.3V	3.3V
GND	GND	GND
SDA	SDA.1	3
SCL	SCL.1	5

Take the LCD1602 RGB Module as an example, just connect it to the Raspberry Pi. **The color of the actual cable may be different from the figure here, please connect them according to the pins instead of color.**



If you are using Jetson Nano, the connection should be as bellow:



[Download the demo](#)

Open the terminal of the Raspberry Pi, execute command to download demo codes:

```
cd ~
wget https://files.waveshare.com/upload/5/5b/LCD1602-RGB-Module-demo.zip
unzip LCD1602-RGB-Module-demo.zip
cd ~/LCD1602-RGB-Module-demo/Raspberry
sudo chmod 777 *
sudo python Choose_Color.py
sudo python Discoloration.py
```

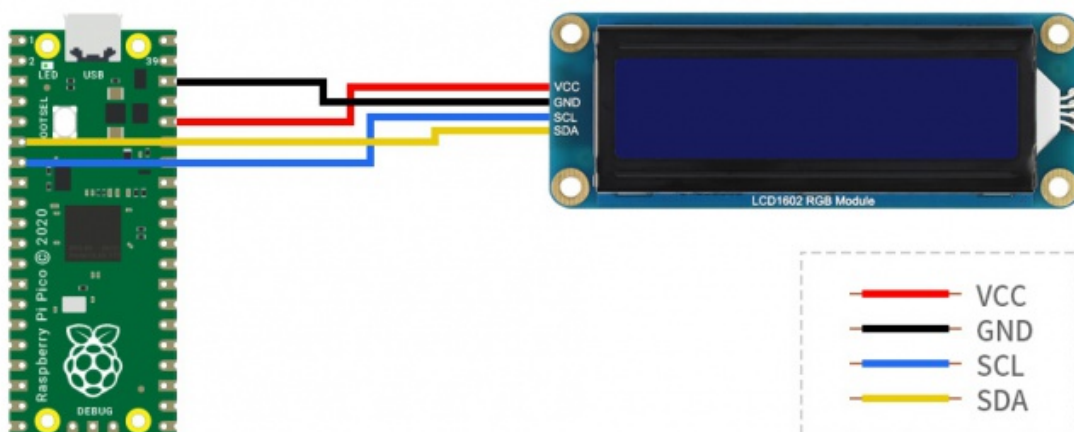
Environment Setting

1. Click [here](#) to see how to set up the Raspberry Pi environment.
2. Click [here](#) to see how to set up the Windows environment.

Pico

Hardware Configuration

LCD PIN	Pico
VCC	3.3V
GND	GND
SDA	GP4
SCL	GP5



Setup environment

Please refer to Raspberry Pi's guide:

<https://www.raspberrypi.org/documentation/pico/getting-started/>

Raspberry Pi

1. [Log in Raspberry Pi terminal by SSH](#) or press Ctrl+Alt+T at the same time while using the screen to open the terminal.
2. Download and unzip the program to the Pico C/C++ SDK directory, for those who have not yet installed the SDK, please refer to the [tutorial](#).

```
#Note that the directory of SDK may be different for different users, you need to check the actual directory. Generally, it should be ~/pico/.
wget -P ~/pico https://files.waveshare.com/upload/5/5b/LCD1602-RGB-Module-demo.zip
cd ~/pico
unzip LCD1602-RGB-Module-demo.zip
```

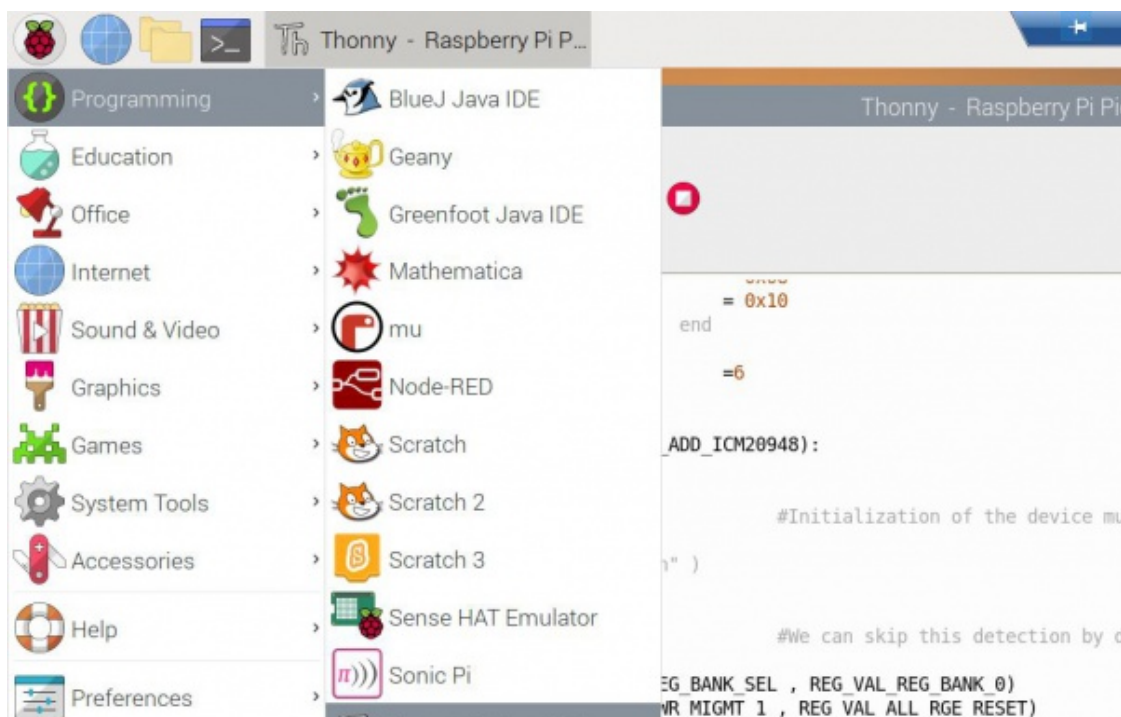
C

- Hold the BOOTSEL button of Pico, and connect the USB interface of Pico to Raspberry Pi then release the button.
- Compile and run the LCD1602-RGB-Module-demo examples:

```
cd ~/pico/LCD1602-RGB-Module-demo/Pico/c/build/
cmake ..
make
sudo mount /dev/sda1 /mnt/pico && sudo cp LCD1602_RGB_Module_demo.uf2 /mnt/pico/
&& sudo sync && sudo umount /mnt/pico && sleep 2
```

python

- Refer to Raspberry Pi's guides to setup Micropython firmware for Pico.
- Open the Thonny IDE, and drag the demo to IDE and run on Pico as below. And save RGB1602.py to Pico's file system.





Windows

C

- Download and unzip the [the demo](#) to your Windows desktop, refer to [Raspberry Pi's guides](#) to set up the Windows software environment settings.
- Find the LCD1602-RGB-Module-demo.uf2 file under the build file under the demo directory, press and hold the BOOTSEL button of Pico, connect the USB of Pico to the PC with a MicroUSB cable, and drag the uf2 file into Pico, and then the Pico will run the demo directly.
- **【Note】** If some users want to modify the demo, after debugging, you can use the following method to generate a .uf2 file:
 - 1. Copy the LCD1602-RGB-Module-demo folder from Pico/c to your pico-examples directory, and then modify the CMakeLists.txt configuration file in the pico-example directory as shown in the figure below.
 - 2. Open Visual Studio Code, open your pico-examples folder, select LCD1602_RGB_Module_demo, and click Generate, you can find the LCD1602_RGB_Module_demo.uf2 file in the build folder.

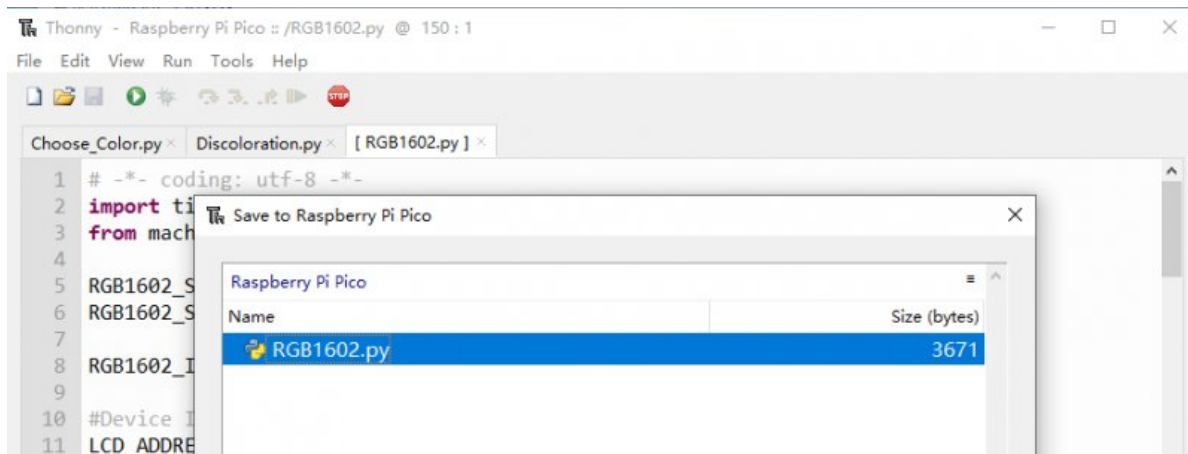


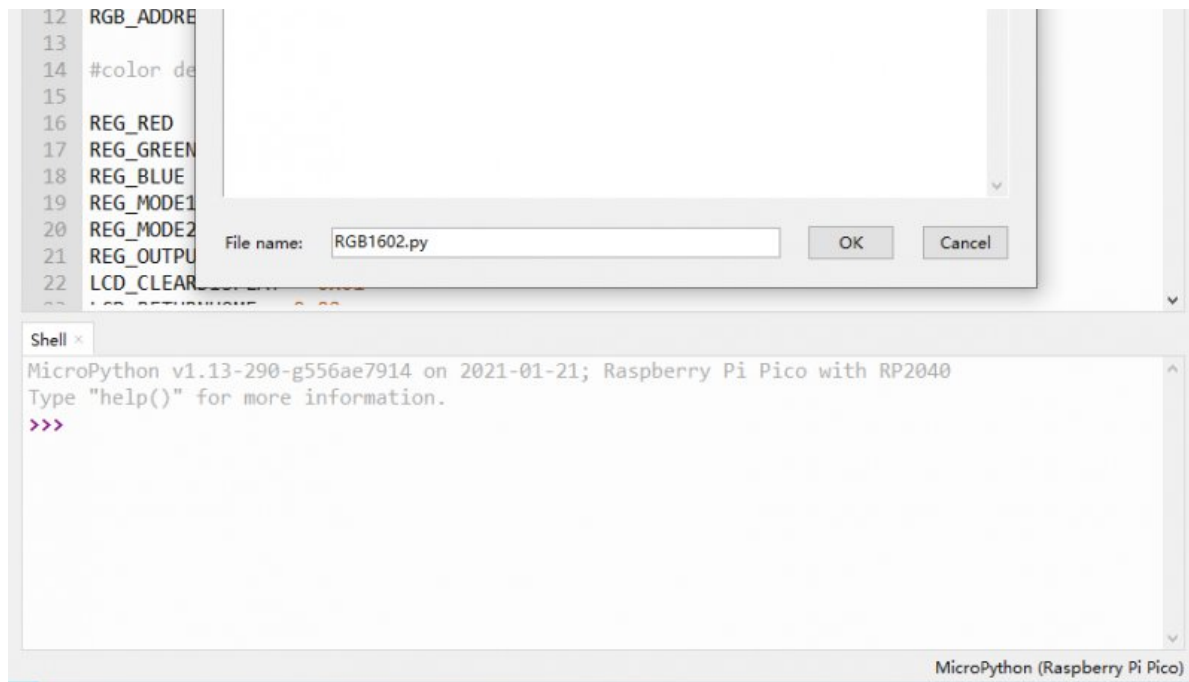
python

- Download and unzip [the demo](#) to your Windows desktop, refer to [Raspberry Pi's guides](#) to set up the Windows software environment settings.
- Open the demo program in Thonny, as shown in the figure below:



- Here you also need to save RGB1602.py to PICO, select RGB1602.py, save it as Raspberry pi pico, and name it RGB1602.py.





- And then run the demo under Choose_Color.py or Discoloration.py.
- Code analysis

Choose_Color.py

```
#define color
rgb9 = (0,255,0) #cyan'
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 second row and the zeroth column
lcd.printout("Hello, World!")#write characters
lcd.setRGB(rgb1[0],rgb1[1],rgb1[2]); #Set the backlight
```

Discoloration.py

```
t=0
while True:

    r = int((abs(math.sin(3.14*t/180)))*255); #RGB changes over time
    g = int((abs(math.sin(3.14*(t+60)/180)))*255);
    b = int((abs(math.sin(3.14*(t+120)/180)))*255);
    t = t + 3;
    lcd.setRGB(r,g,b);#Reset the value of RGB
# set the cursor to column 0, line 1
    lcd.setCursor(0, 0) #Locate to the first row and the zeroth column
# print the number of seconds since reset:
    lcd.printout("Waveshare")#write characters
    lcd.setCursor(0, 1) #Locate to the second row and the zeroth column
    lcd.printout("Hello, World!")#write characters
    time.sleep(0.3)
```


Arduino

Hardware connection

LCD PIN	Arduino
VCC	3.3V
GND	GND
SDA	SDA
SCL	SCL

The hardware connection as below:



Library file loading for Arduino

- Download the demo from the Resources, there is a Waveshare-LCD1602-RGB-master.zip library file in the Arduino folder.



- There are two methods to load the library:
 - **Method 1:** Unzip the entire .zip package to the libraries folder of the Arduino IDE.



- **Method 2:** Add this .zip package directly to the Arduino IDE.
 - Open the Arduino IDE -> Choose 'Sketch' in the taskbar -> Choose 'Include Library' -> Add .ZIP library.
 - Find Waveshare-LCD1602-RGB-mater.zip in the demo folder and click to open it.

Resources

Demo

- [LCD1602 RGB Module Demo](#)

Document

- [LCD1602 RGB Module datasheet](#)

FAQ

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

Answer:

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

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](#)