CONTENTS

1.Description

Lora RA-08H Development Board is a development board based on the combination of Raspberry Pi RP2040 main control chip and RA-08H, onboard TFT 1.8-inch SPI display, passive buzzer, RS485 communication module, customizable buttons, programmable indicator lights, multiple Crowtail and general IO interfaces, rich external interface resources, can be used with Crowtail series modules or modules with other interfaces. The onboard Lorawan module is connected to the Lorawan platform to realize remote module control, which can be applied to maker education, smart home, industrial control, etc.

Model: CRT01259B

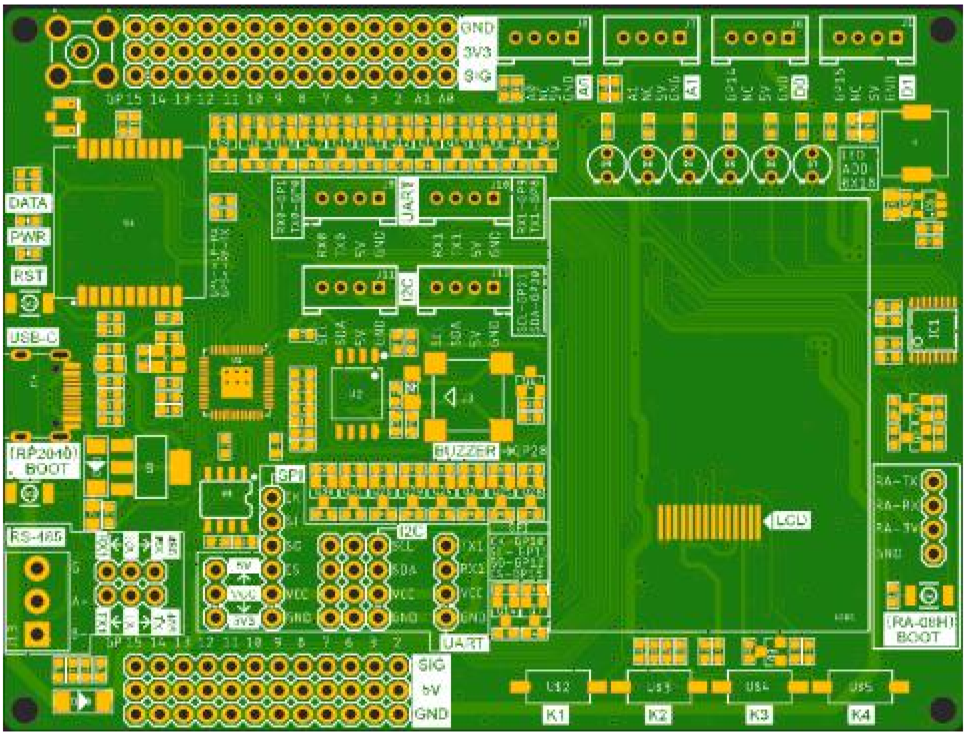
2.Feature

- Integrate RP2040 and RA-08H (Lorawan module)
- Main control Raspberry Pi chip RP2040, built-in 264KB SRAM, onboard 4MB Flash
- Dual Core Arm Cortex-M0+ @ 133MHz
- RA-08H module adopts embedded memory, 128KB Flash, 16KB SRAM
- RA-08H module supports frequency band: 803MHZ-930MHZ
- RA-08H module supports external antenna, SMA interface or IPEX first generation interface
- Compatible with Arduino/Micropython
- Onboard 1.8-inch SPI-TFT-LCD, resolution: 128*160 driver chip: ST7735S (4-wire SPI)
- 1 passive buzzer
- 4 user-defined buttons
- 6 programmable LEDs
- 1 RS485 communication interface
- 8 5V Crowtail interfaces (2 analog interfaces, 2 digital interfaces, 2 UART interfaces, 2 IIC interfaces)
- 12 5V universal pin header IO interfaces
- 14 3.3V universal pin header IO interfaces
- 1 3.3V/5V switchable SPI interface
- 1 3.3V/5V switchable UART interface
- 3 3.3V/5V switchable IIC interfaces

3.Technical Specification

- Working input voltage: USB 5V-1A
- Operating temperature range: -10°C ~ 65°C
- Dimensions: 102mm(L)x76.5mm(W)

4.Interface Function



- Onboard Ports and Functions

Name	Interface/Function Definition
DATA	RA-08H data sending and receiving indicator light, when RA-08H communicates with the outside world (need to be set by the user).
PWR	Power Indicator
RST	Reset button, press this button to reset the system.
(RP2040)BOOT	Download button, hold down the Boot button and press the RESET button to enter boot firmware download mode. Users can download firmware through the USB interface.
USB-C	USB-C interface, it is used for the power supply of the development board and the communication interface between PC and RP2040.
RS485	RS485 interface, used to communicate with external RS485 module.
485RX<-RX->RX1	3P pin header interface RS485 chip and RP2040 communication pin switching interface, when using the

	RS485 module, it is necessary to short the RX and 485RX with a jumper cap, when not using the RS485 module, the default is to short the RX and RX1.
485TX<-TX->TX1	3P pin header interface RS485 chip and RP2040 communication pin switching interface, when using the RS485 module, it needs to short-circuit its TX and 485TX with a jumper cap, and when using the RS485 module, the default is to short-circuit the TX and TX1.
3V3<-VCC->5V	3P pin header interface, used for switching the working voltage of these three interfaces: 6P-SPI interface, 3*4P-IIC interface and 4P-UART interface.
SPI	6P-SPI pin header interface, which can be used for external SPI interface communication modules and devices, and the power supply voltage is 3.3V/5V switchable.
I2C	3*4P-IIC pin header interface, can be used for external I2C interface communication modules and devices, power supply voltage 3.3V/5V switchable.
UART	4P-UART pin header interface, can be used for external UART serial communication modules and devices, power supply voltage 3.3V/5V switchable.
GND-5V-SIG	3*12PIN, 5V general-purpose IO header interface, which can be used for external 5V sensor modules or devices.
GND-3V3-SIG	3*14PIN, 3.3V general-purpose IO header interface, which can be used for external 3.3V sensor modules or devices.
Crowtail-A0	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to interface with 5V Crowtail analog signals.
Crowtail-A1	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to interface with 5V Crowtail analog signals.
Crowtail-D0	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to interface with 5V Crowtail digital signals.

Crowtail-D1	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to interface with 5V Crowtail digital signals.
Crowtail-UART	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to connect to 5V Crowtail UART serial communication.
Crowtail-I2C	External Crowtail-5V-HY-4P-2.0 port. Sensors or modules that can be used to connect to 5V Crowtail I2C interface communication.
(RA-08H)BOOT	Long press the download button before power-on. Connect the external USB-to-serial port module. Connect the cables to the external USB-to-serial port module according to (GND~GND, RA-3V~3.3V, RA-RX~TX, RA-TX~RX) and update the firmware of the RA-08H module through the TremoProgrammer_v0.8 software.
LCD	1.8 inch SPI display

- GPIO Pin Corresponding Port

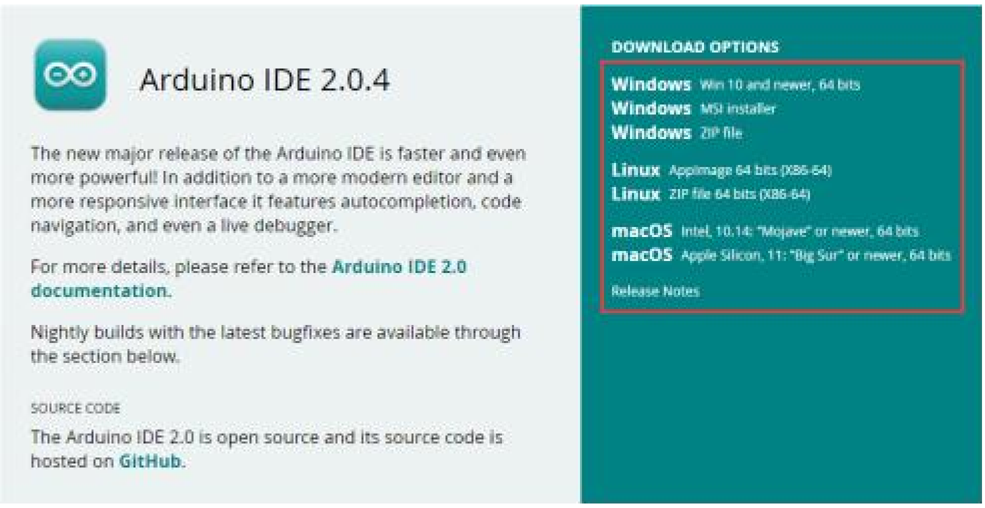
RP2040	**Lora RA-08H Development Board**
GPIO0	Crowtail-UART0-TX0
GPIO1	Crowtail-UART0-RX0
GPIO2	GPIO2
GPIO3	GPIO3
GPIO4	RA-08H-LPRXD
GPIO5	RA-08H-TXD
GPIO6	GPIO6
GPIO7	GPIO7
GPIO8	GPIO8/UART1-TX1/Crowtail-UART1-TX1/485TX
GPIO9	GPIO9/UART1-RX1Crowtail-UART1-RX1/485RX
GPIO10	GPIO10/SPI1-SCK
GPIO11	GPIO11/SPI1-MOSI

GPIO12	GPIO12/SPI1-MISO
GPIO13	GPIO13/SPI1-CS
GPIO14	GPIO14/ Crowtail-D0
GPIO15	GPIO15/ Crowtail-D1
GPIO16	LCD-DC
GPIO17	LCD-CS
GPIO18	LCD-SCK
GPIO19	LCD-DIN
GPIO20	I2C0-SDA0/Crowtail-I2C-SDA/PCA9555-SDA
GPIO21	I2C0-SCL0/Crowtail-I2C-SCL/ PCA9555-SCL
GPIO22	LCD-RST
GPIO23	LCD-BLK
GPIO24	485-EN
GPIO25	DATA
GPIO26	A0/ Crowtail-A0
GPIO27	A1/ Crowtail-A1
GPIO28	BUZZER
GPIO29	ADC-KEY

5.Usage

1. Development Environment Construction

1.1 Directly visit the Arduino official website and download the Arduino development tools supported by the device. The link is as follows: ***https://www.arduino.cc/en/software***.



Arduino IDE 2.0.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the **Arduino IDE 2.0 documentation.**

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on **GitHub**.

DOWNLOAD OPTIONS

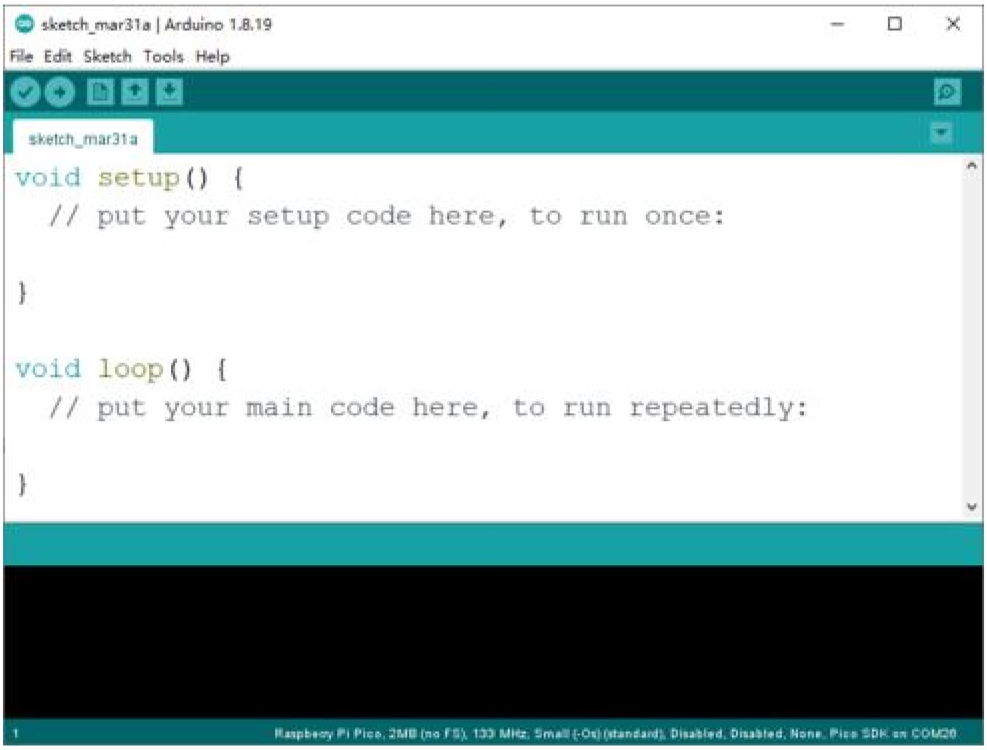
Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

Release Notes

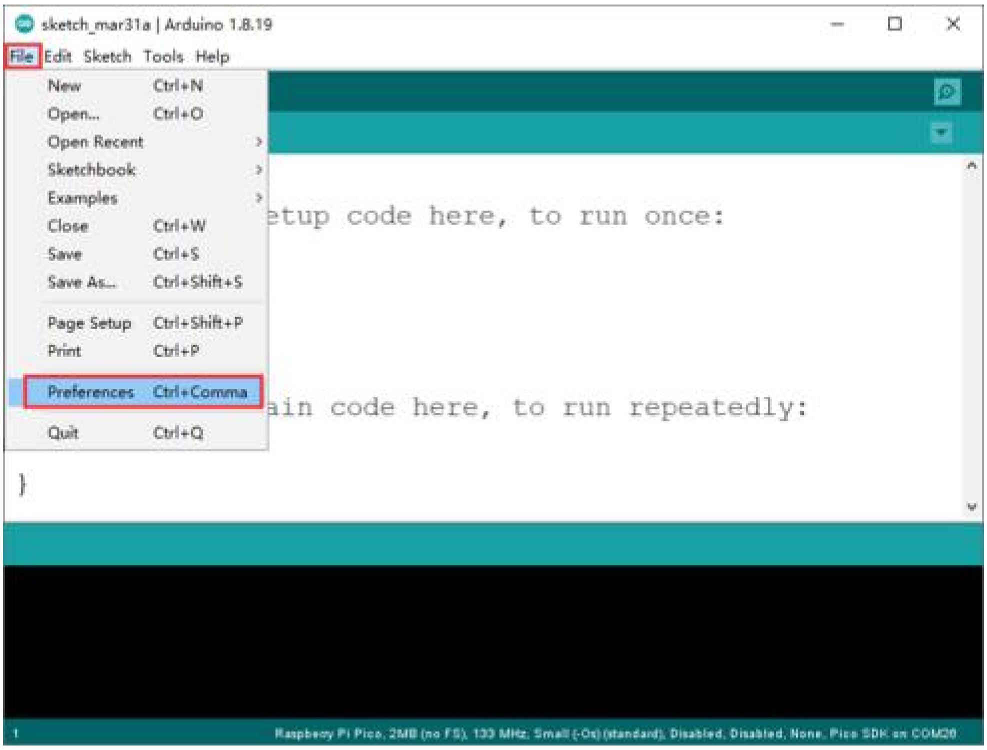
1.2 Click the downloaded Arduino development tool to install the software. After the software installation is complete, open it directly, as shown in the figure below:



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

1 Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM20

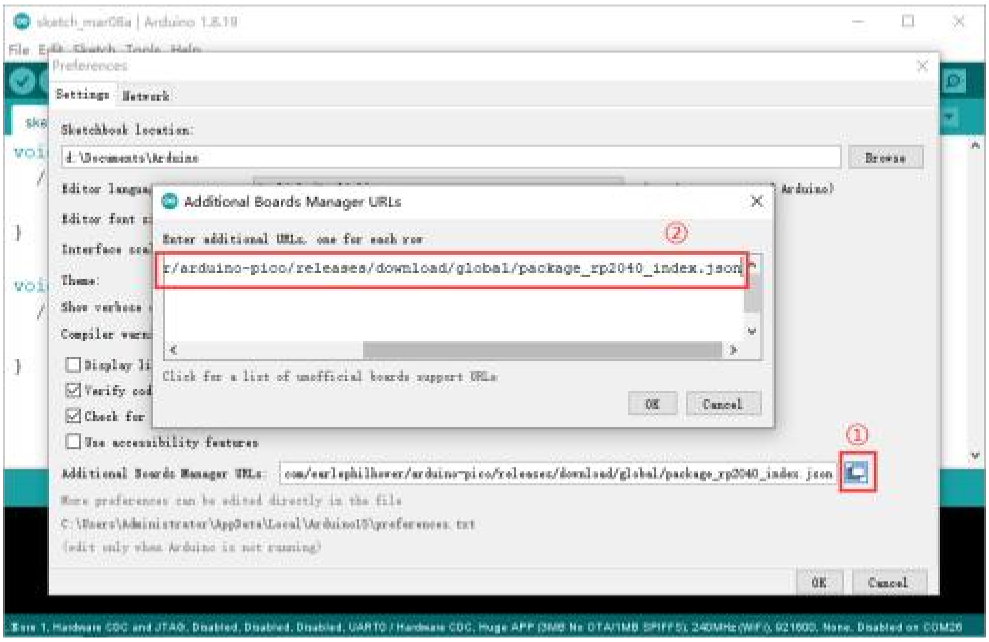
1.3 Click *"File"* -> *"Preferences"* on the menu bar to pop up the preference setting interface.



```
etup code here, to run once:

ain code here, to run repeatedly:

}
```

Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM26

1.4 Click the icon on the far right of the Additional Boards Manager URLscolumn,enter ***https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json*** in the pop-up window, and add RP2040 Arduino IDE development board URLs.



sketch_mar08a | Arduino 1.8.19
File Edit Sketch Tools Help
Preferences
Settings Network
Browse
Additional Boards Manager URLs
②
①
OK Cancel
OK Cancel

Sketchbook location: d:\Documents\Arduino

Enter additional URLs, one for each row

r/arduino-pico/releases/download/global/package_rp2040_index.json

Click for a list of unofficial boards support URLs

☐ Display li
☑ Verify cod
☑ Check for
☐ Use accessibility features

Additional Boards Manager URLs: com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

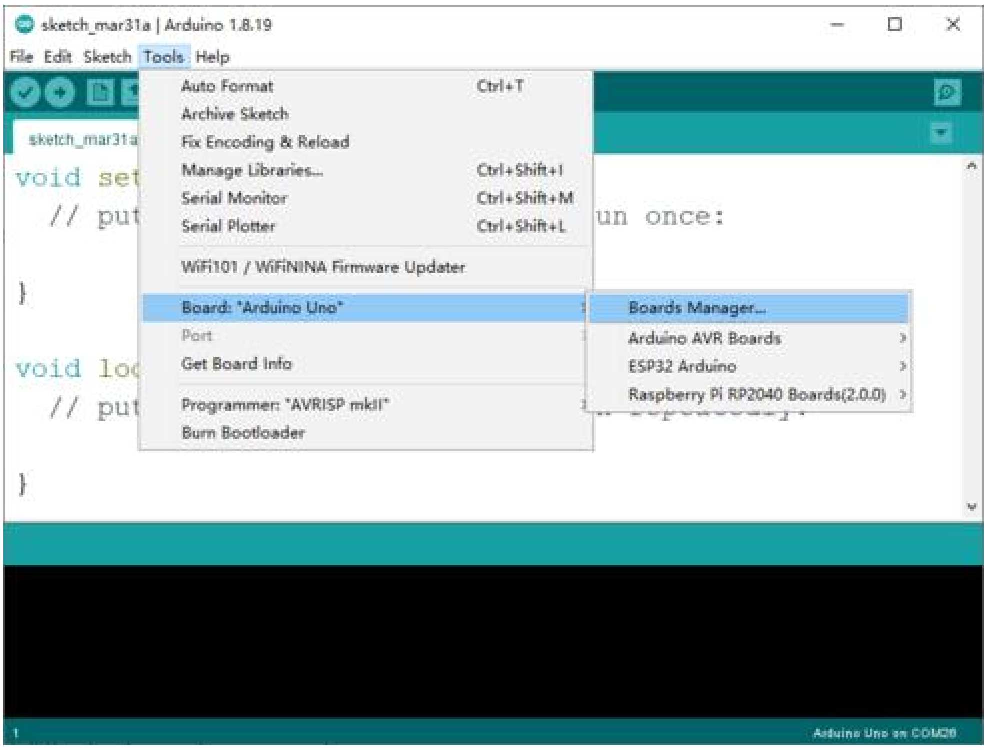
More preferences can be edited directly in the file

C:\Users\Administrator\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

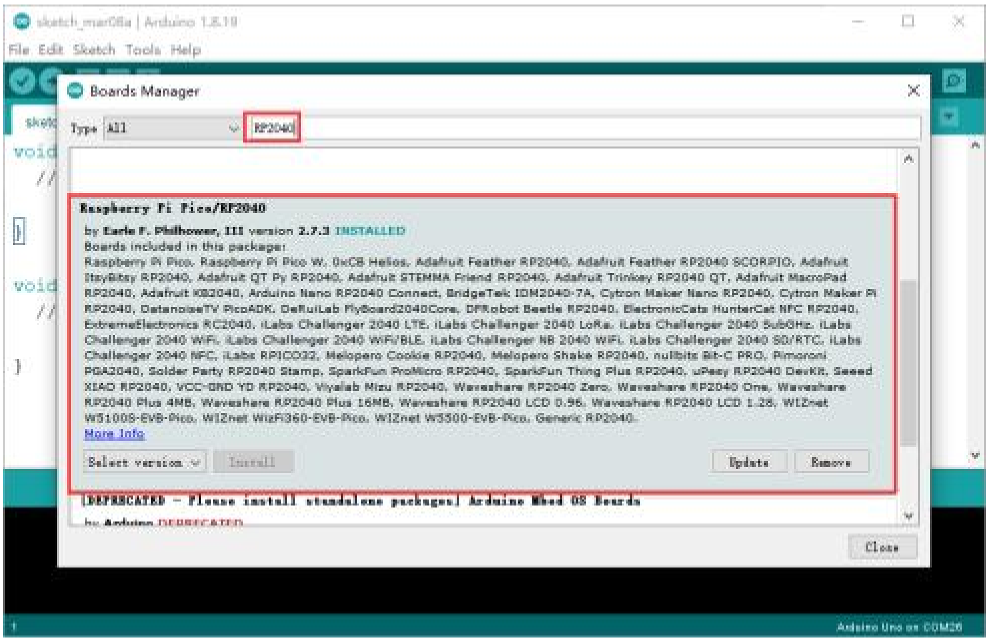
Core 1, Hardware CDC and JTAG, Disabled, Disabled, Disabled, UART0 / Hardware CDC, Huge APP (3MB No OTA/1MB SPIFFS), 240MHz (WiFi), 921600, None, Disabled on COM26

1.5 Click *"Tools" -> "Board:..." -> "Boards Manager..."* in the menu bar to enter the Boards Manager interface.


sketch_mar31a | Arduino 1.8.19
File Edit Sketch Tools Help
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
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
ESP32 Arduino
Raspberry Pi RP2040 Boards(2.0.0)
Arduino Uno on COM26

1.6 Enter "RP2040" in the search box, select "Raspberry Pi Pico/RP2040" and install the development environment.

It should be noted here that when selecting the development environment version, do not choose the latest one. It is recommended to choose version 2.6.0.

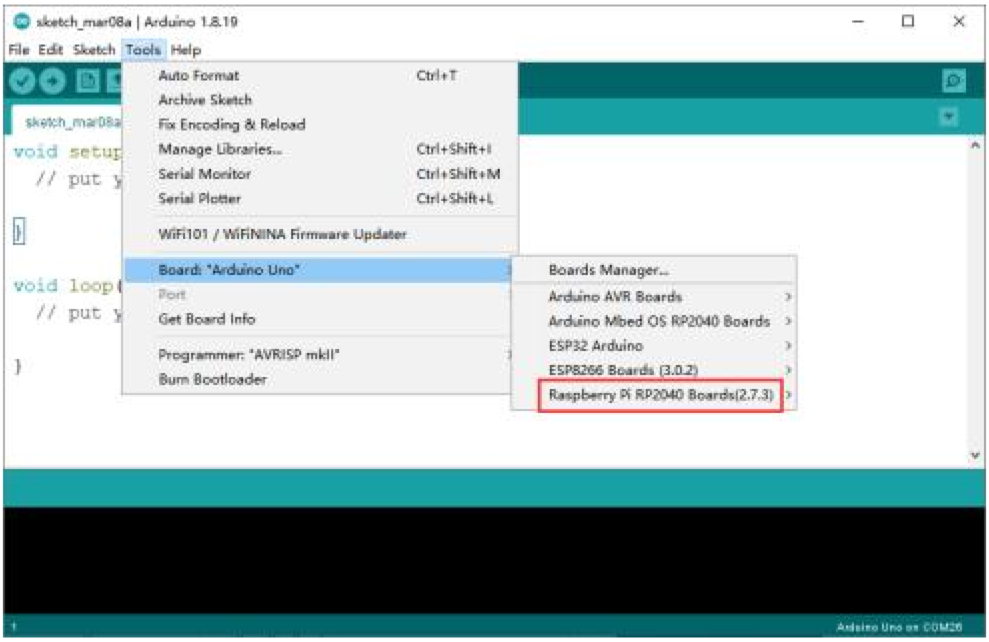


Raspberry Pi Pico/RP2040

by **Earle F. Philhower, III** version **2.7.3** INSTALLED
Boards included in this package:
Raspberry Pi Pico, Raspberry Pi Pico W, 0xCB Helios, Adafruit Feather RP2040, Adafruit Feather RP2040 SCORPIO, Adafruit ItsyBitsy RP2040, Adafruit QT Py RP2040, Adafruit STEMMA Friend RP2040, Adafruit Trinkey RP2040 QT, Adafruit MacroPad RP2040, Adafruit KB2040, Arduino Nano RP2040 Connect, BridgeTek IDM2040-7A, Cytron Maker Nano RP2040, Cytron Maker Pi RP2040, DatanoiseTV PicoADK, DeRuiLab FlyBoard2040Core, DFRobot Beetle RP2040, ElectronicCats HunterCat NFC RP2040, ExtremeElectronics RC2040, iLabs Challenger 2040 LTE, iLabs Challenger 2040 LoRa, iLabs Challenger 2040 SubGHz, iLabs Challenger 2040 WiFi, iLabs Challenger 2040 WiFi/BLE, iLabs Challenger NB 2040 WiFi, iLabs Challenger 2040 SD/RTC, iLabs Challenger 2040 NFC, iLabs RPICO32, Melopero Cookie RP2040, Melopero Shake RP2040, nullbits Bit-C PRO, Pimoroni PGA2040, Solder Party RP2040 Stamp, SparkFun ProMicro RP2040, SparkFun Thing Plus RP2040, uPesy RP2040 DevKit, Seeed XIAO RP2040, VCC-GND YD RP2040, Viyalab Mizu RP2040, Waveshare RP2040 Zero, Waveshare RP2040 One, Waveshare RP2040 Plus 4MB, Waveshare RP2040 Plus 16MB, Waveshare RP2040 LCD 0.96, Waveshare RP2040 LCD 1.28, WIZnet W5100S-EVB-Pico, WIZnet WizFi360-EVB-Pico, WIZnet W5500-EVB-Pico, Generic RP2040.
More Info

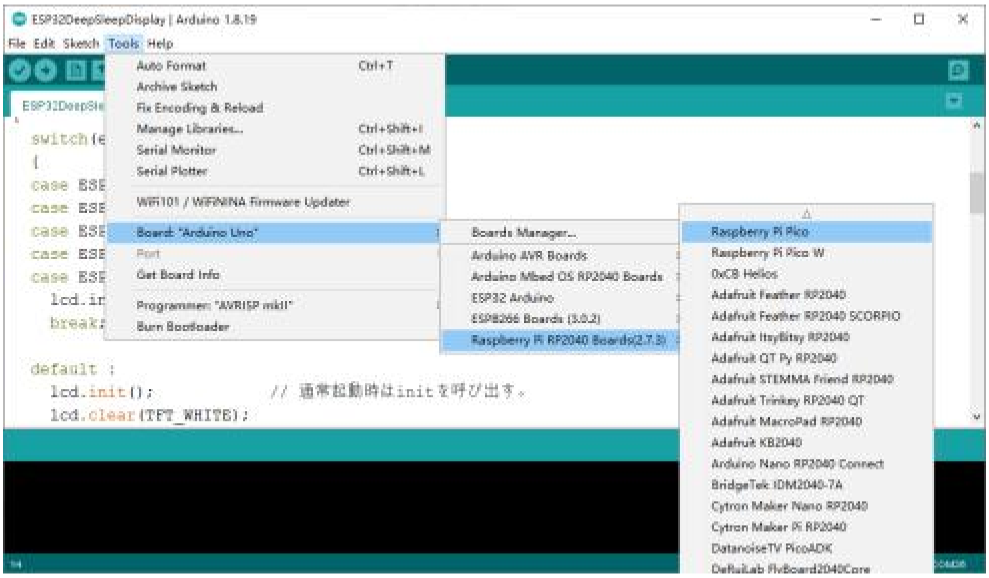
[DEPRECATED – Please install standalone packages] Arduino Mbed OS Boards
by **Arduino** DEPRECATED

1.7 After the environment is successfully installed, if you find "Rasperry Pi RP2040 Boards" in the Arduino IDE development board management, it means that the environment is successfully built.

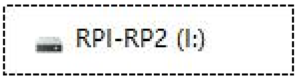

sketch_mar08a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_mar08a
void setup
// put y
void loop(
// put y
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Mbed OS RP2040 Boards
ESP32 Arduino
ESP8266 Boards (3.0.2)
Raspberry Pi RP2040 Boards(2.7.3)
Arduino Uno on COM26

2. Software Settings

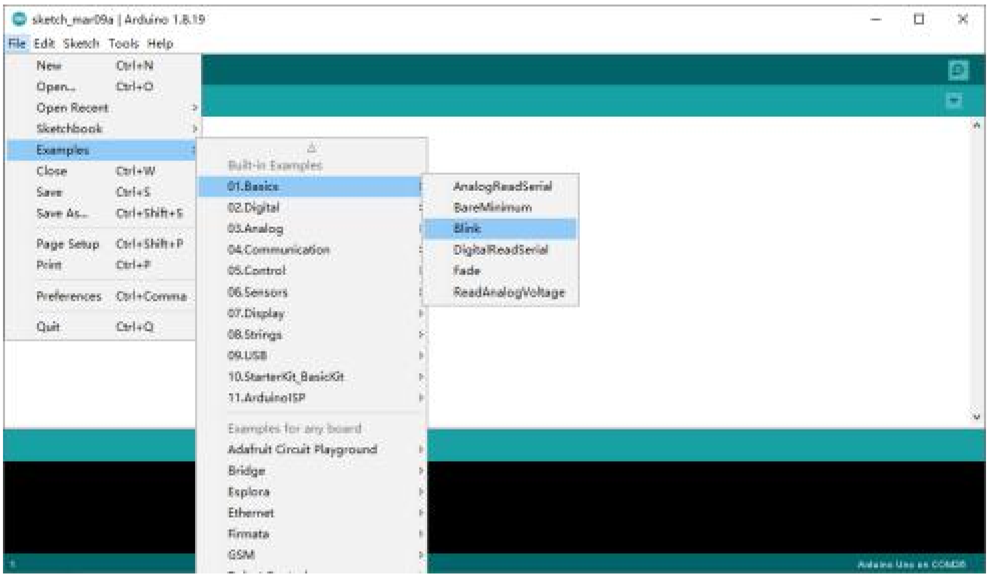
2.1 For the first use, you need to set up and select the correct development board. Click *"Tools" -> "Board:..." -> "Rasperry Pi RP2040 Boards" -> "Rasperry Pi Pico"* in the menu bar, as shown in the figure below:



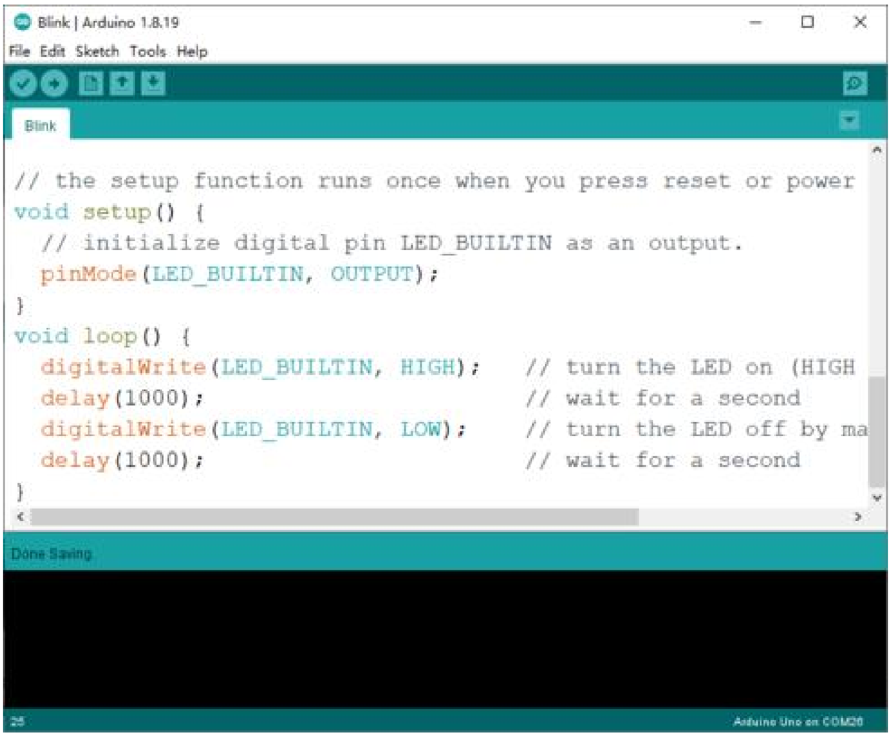
2.2 Connect the Lora RA-08H Development Board to the computer via a TYPE-C USB cable. When the development board is connected to the computer for the first time, a "RPI-RP2" USB virtual U disk will appear.



2.3 Here you need to burn a sample code in the opened Arduino IDE to convert the "RPI-RP2" USB virtual U disk into a "COM port". The sample code path is shown in the figure below:



The code after opening is as follows:



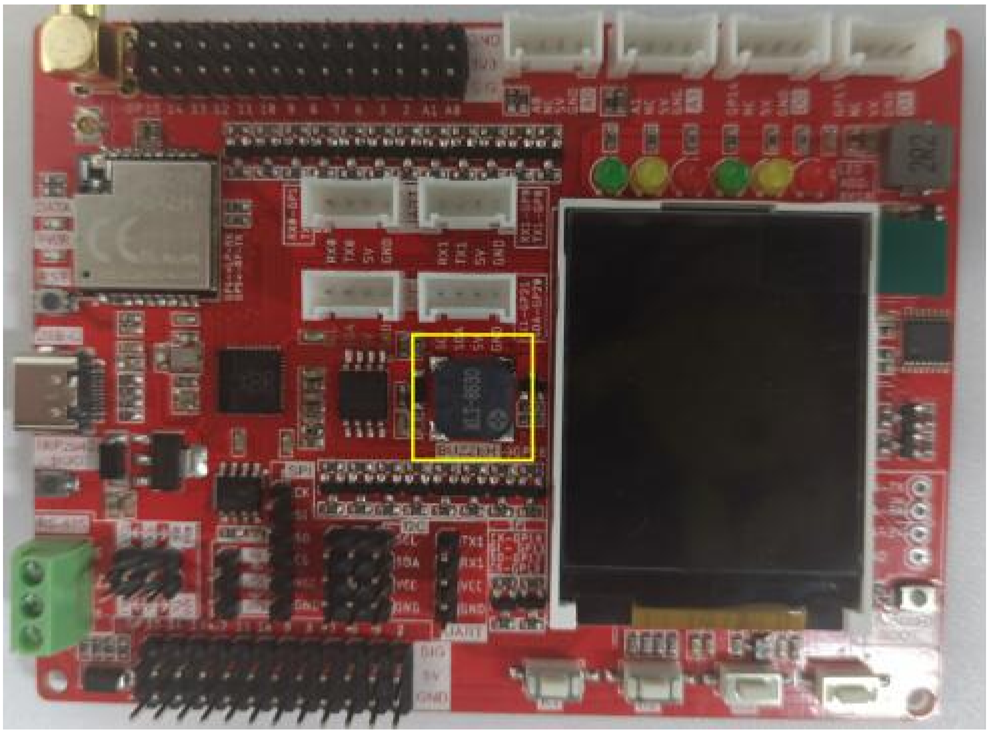
```
// the setup function runs once when you press reset or power
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by ma
  delay(1000);                       // wait for a second
}
```

2.4 Click the download icon at the bottom of the menu bar to burn the code to the Lora RA-08H Development Board. After the code is burned successfully, COMXX (Raspberry Pi Pico) will be displayed in *"Tools" -> "Port"*. Now you can program Lora RA-08H Development Board with Arduino IDE!

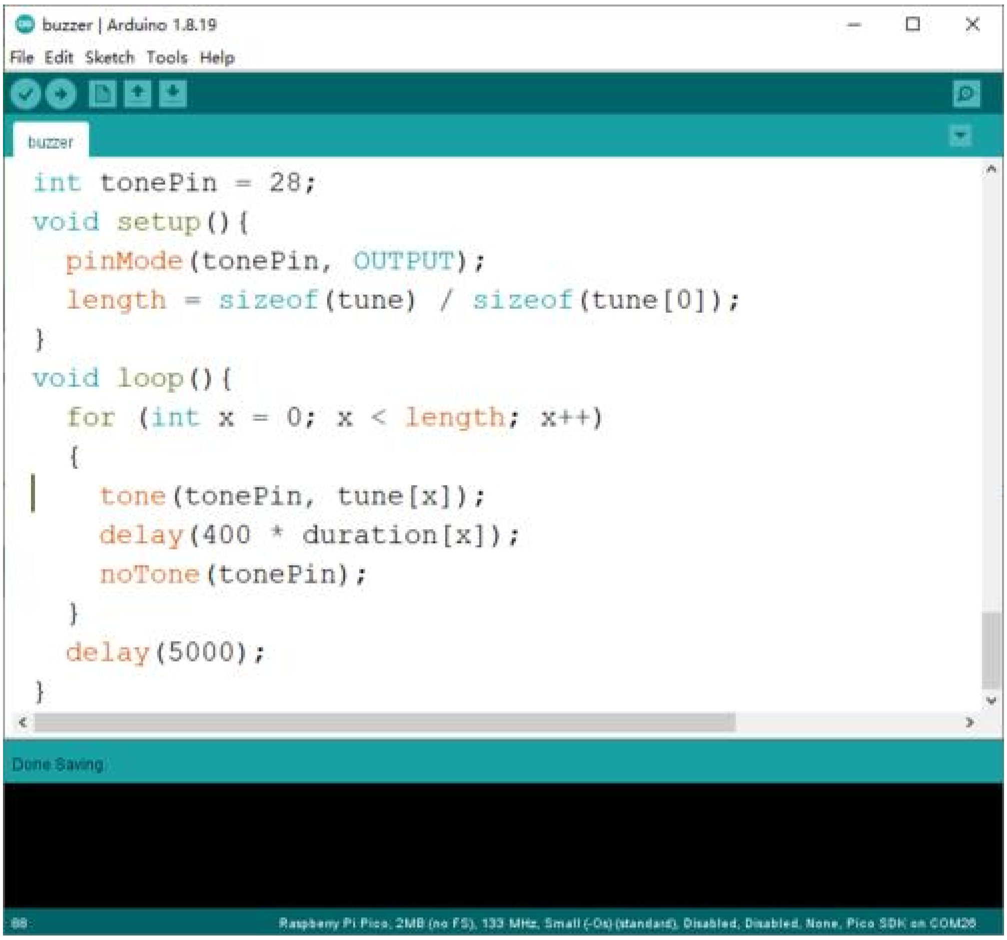
3. Sample Program

Example 1 - Driving the Onboard Passive Buzzer Let the buzzer go off!

3.1.1 The position of the onboard buzzer on the development board.



3.1.2 Open the *"Buzzer.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.



```
int tonePin = 28;
void setup(){
  pinMode(tonePin, OUTPUT);
  length = sizeof(tune) / sizeof(tune[0]);
}
void loop(){
  for (int x = 0; x < length; x++)
  {
    tone(tonePin, tune[x]);
    delay(400 * duration[x]);
    noTone(tonePin);
  }
  delay(5000);
}
```

Done Saving.

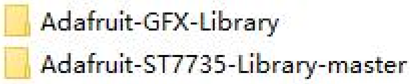
88   Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM26

3.1.3 After the program is uploaded successfully, the buzzer will sound automatically!

Example 2 - Let the onboard 1.8-inch LCD display red, green and blue respectively

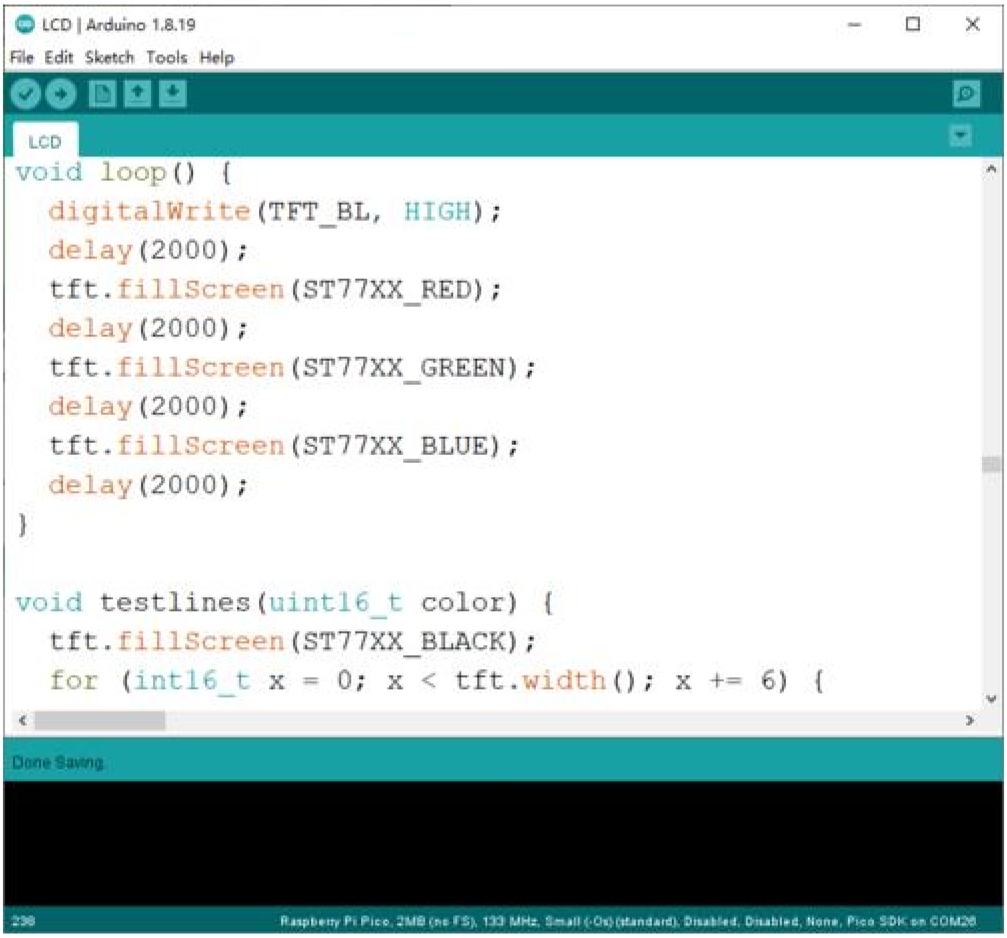
3.2.1 Copy the *"Adafruit-GFX-Library"* library file and *"Adafruit-ST7735-Library-master"* library file to the libraries folder of the Arduino IDE.

(Original path: ***"...\Lora_RA-08H_Development_Board\Library\..."****; Copy path:* ***"...\Arduino\libraries\..."*** *)*



Adafruit-GFX-Library
Adafruit-ST7735-Library-master

3.2.2 Open the *"LCD.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.



```
void loop() {
  digitalWrite(TFT_BL, HIGH);
  delay(2000);
  tft.fillScreen(ST77XX_RED);
  delay(2000);
  tft.fillScreen(ST77XX_GREEN);
  delay(2000);
  tft.fillScreen(ST77XX_BLUE);
  delay(2000);
}

void testlines(uint16_t color) {
  tft.fillScreen(ST77XX_BLACK);
  for (int16_t x = 0; x < tft.width(); x += 6) {
```

```
Done Saving
```

```
238          Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM28
```

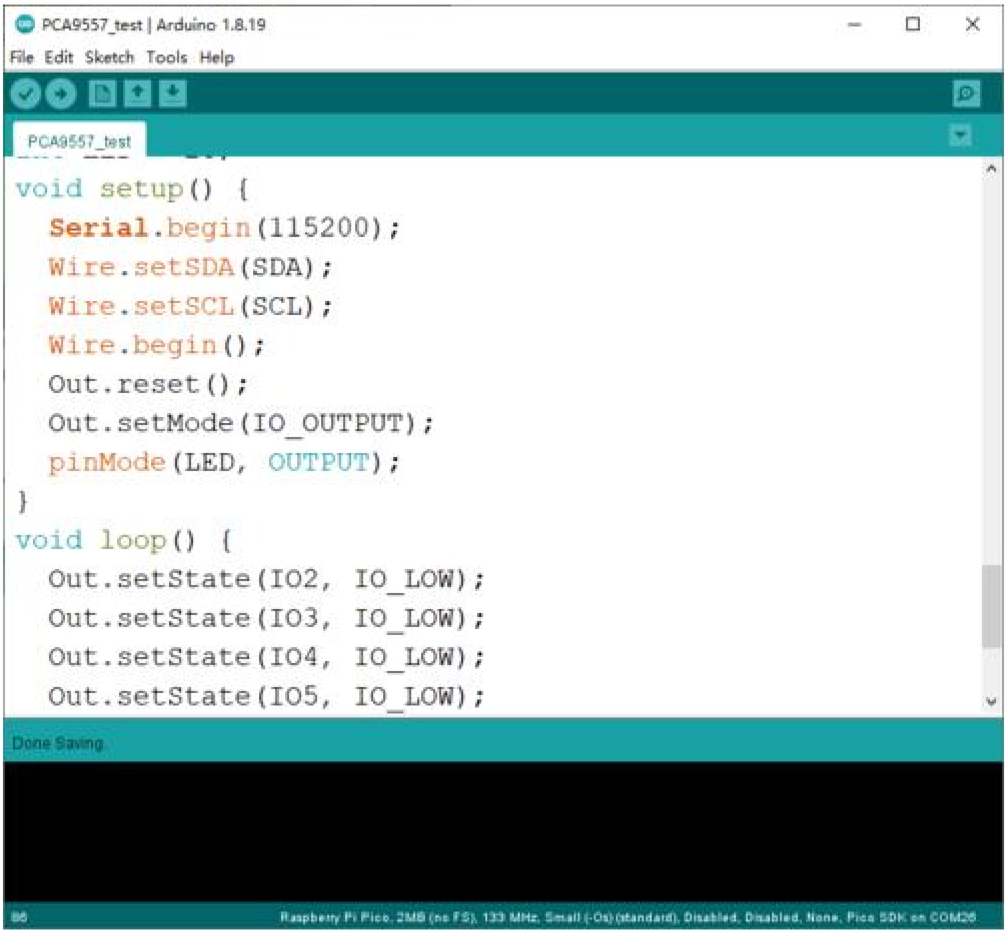
3.2.3 After the upload is successful, the LCD screen will display blue, green and red backlight in turn.

Example 3 - Light up 6 programmable LEDs on board through I2C control

3.3.1 Copy the *"PCA9557"* library file to the libraries folder of the Arduino IDE.

PCA9557

3.3.2 Open the *"PCA9557_test.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.

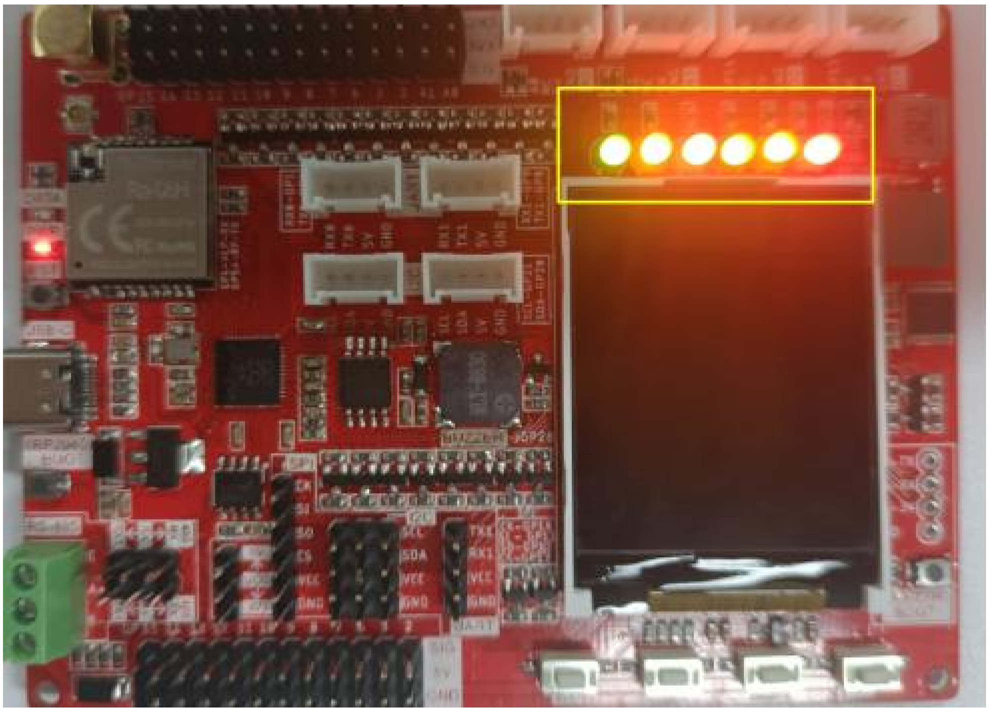


```
void setup() {
  Serial.begin(115200);
  Wire.setSDA(SDA);
  Wire.setSCL(SCL);
  Wire.begin();
  Out.reset();
  Out.setMode(IO_OUTPUT);
  pinMode(LED, OUTPUT);
}
void loop() {
  Out.setState(IO2, IO_LOW);
  Out.setState(IO3, IO_LOW);
  Out.setState(IO4, IO_LOW);
  Out.setState(IO5, IO_LOW);
```

Done Saving.

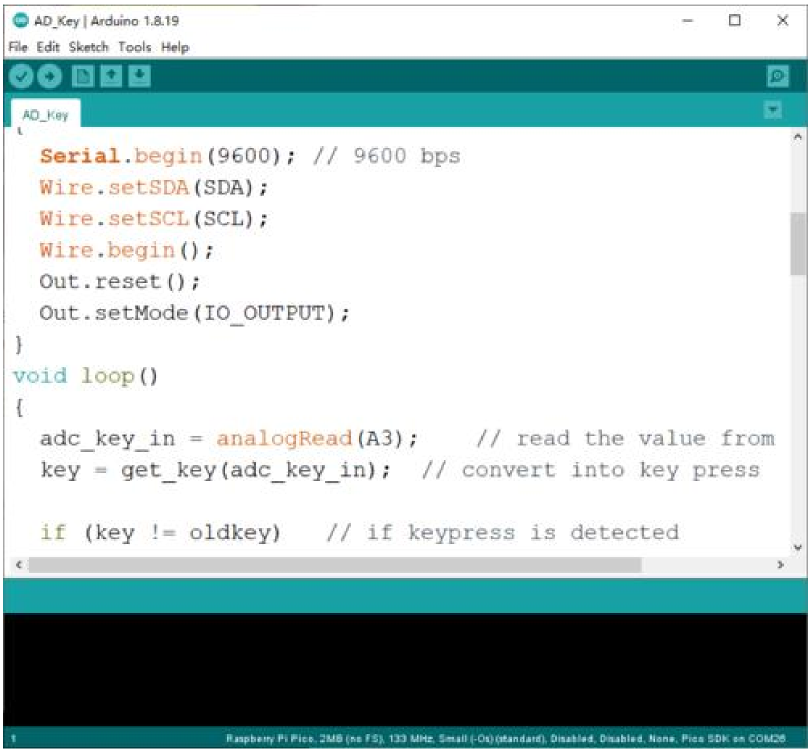
86 Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM26

3.3.3 The effect after the program is uploaded successfully.



Example 4 - Read the 4 ADC-KEYs on the board through the ADC, and print the corresponding key value in the COM port

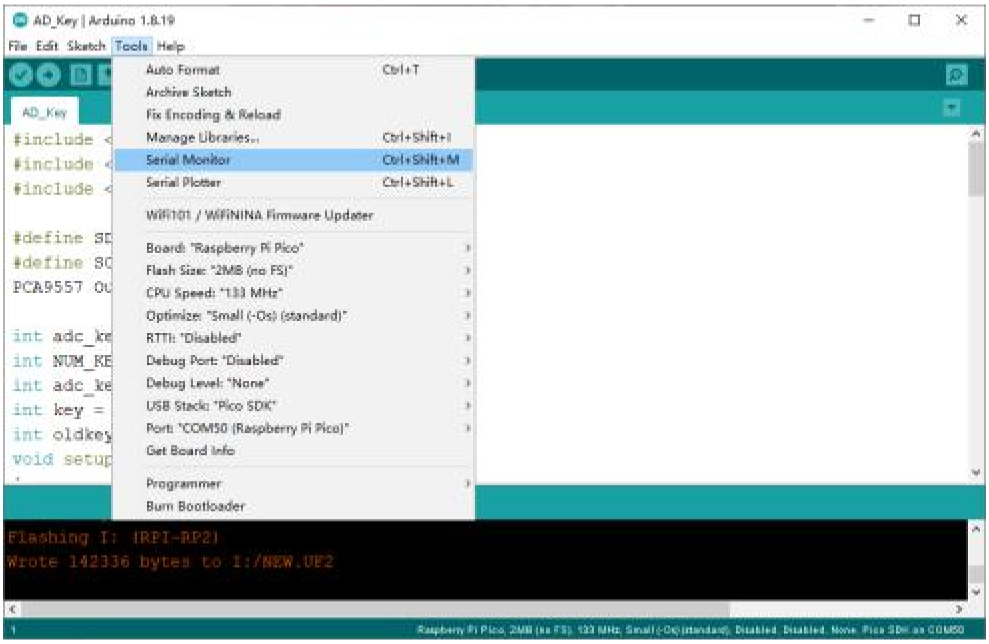
3.4.1 Open the *"AD_Key.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.



```
{
  Serial.begin(9600); // 9600 bps
  Wire.setSDA(SDA);
  Wire.setSCL(SCL);
  Wire.begin();
  Out.reset();
  Out.setMode(IO_OUTPUT);
}
void loop()
{
  adc_key_in = analogRead(A3);     // read the value from
  key = get_key(adc_key_in);  // convert into key press

  if (key != oldkey)    // if keypress is detected
```

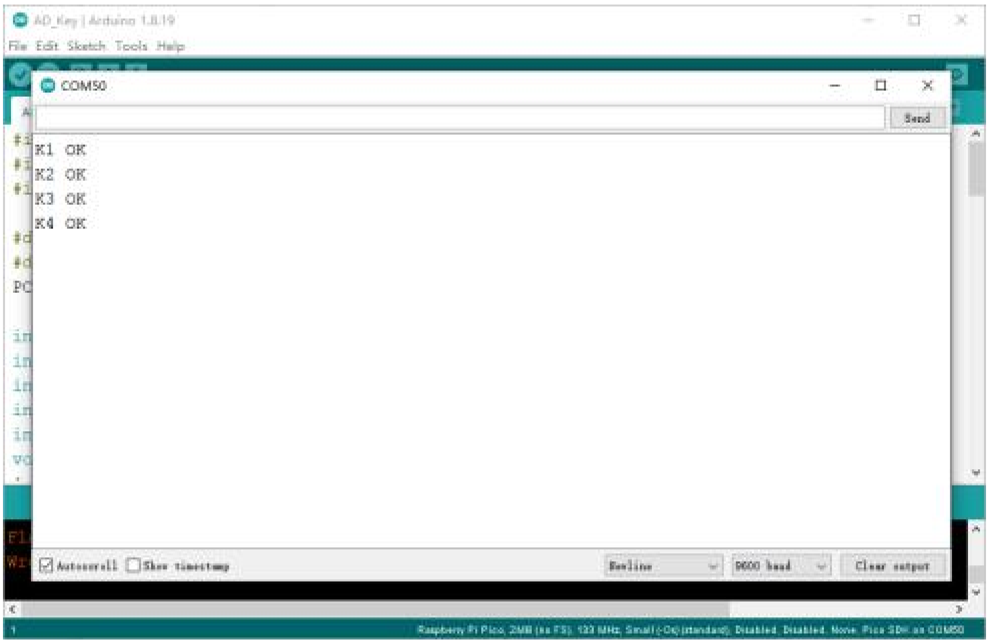
```
1        Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM26
```

3.4.2 After the program is uploaded successfully, open the Serial Monitor. After pressing the keys one by one, observe whether the serial port prints the corresponding key value.


AD_Key | Arduino 1.8.19
File Edit Sketch Tools Help
AD_Key
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
Board: "Raspberry Pi Pico"
Flash Size: "2MB (no FS)"
CPU Speed: "133 MHz"
Optimize: "Small (-Os) (standard)"
RTTI: "Disabled"
Debug Port: "Disabled"
Debug Level: "None"
USB Stack: "Pico SDK"
Port: "COM50 (Raspberry Pi Pico)"
Get Board Info
Programmer
Burn Bootloader
Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM50

```
#include <
#include <
#include <

#define SD
#define SC
PCA9557 Ou

int adc_ke
int NUM_KE
int adc_ke
int key =
int oldkey
void setup
```

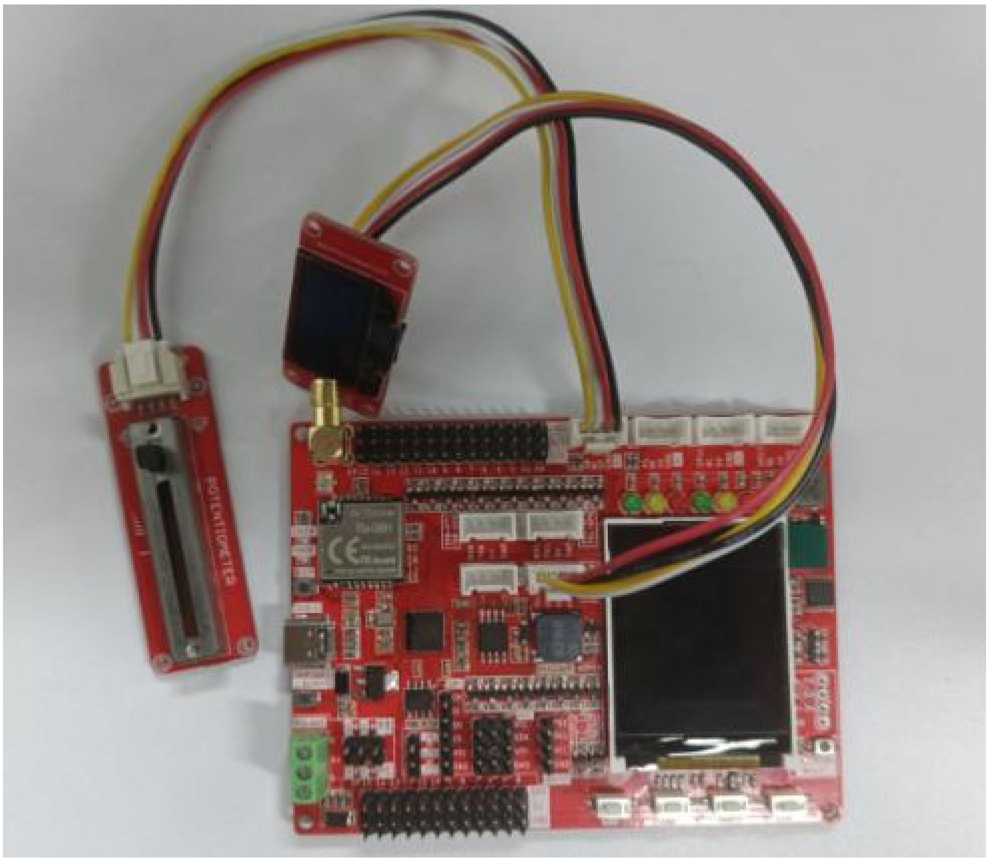
```
Flashing I: (RPI-RP2)
Wrote 142336 bytes to I:/NEW.UF2
```


AD_Key | Arduino 1.8.19
File Edit Sketch Tools Help
COM50
Send
☑ Autoscroll ☐ Show timestamp
Newline
9600 baud
Clear output
Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM50

```
K1 OK
K2 OK
K3 OK
K4 OK
```

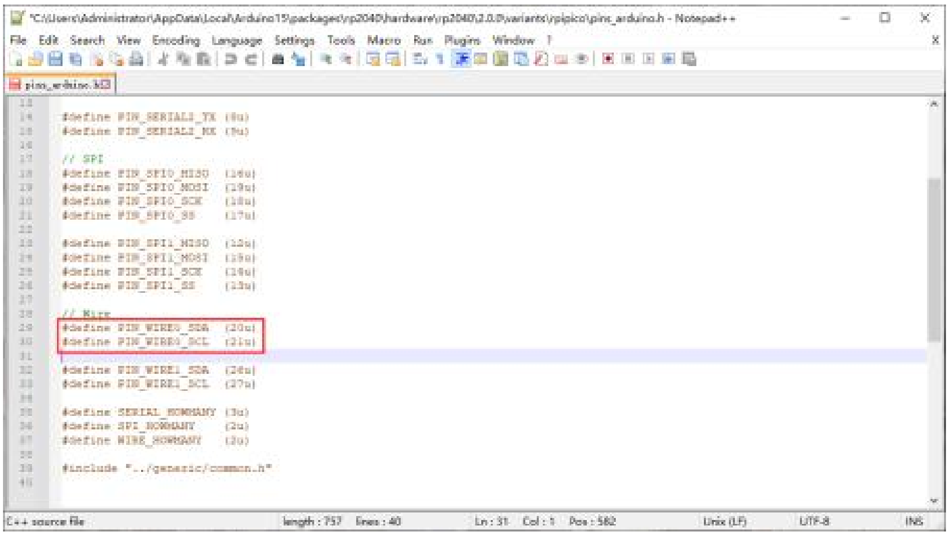
Example 5 - Light up 6 programmable LEDs on board through I2C control

3.5.1 As shown in the figure below, connect the Crowtail OLED module and the Crowtail Potentiometer module to the Crowtail I2C interface of the Lora RA-08H Development Board.



3.5.2 Find the Pins arduino.h file under the *"... \Users \Administrator\AppData\Local\Arduino15\packages\rp2040\hardware\rp2040\2.0.0\variants\rpipico\..."* path.

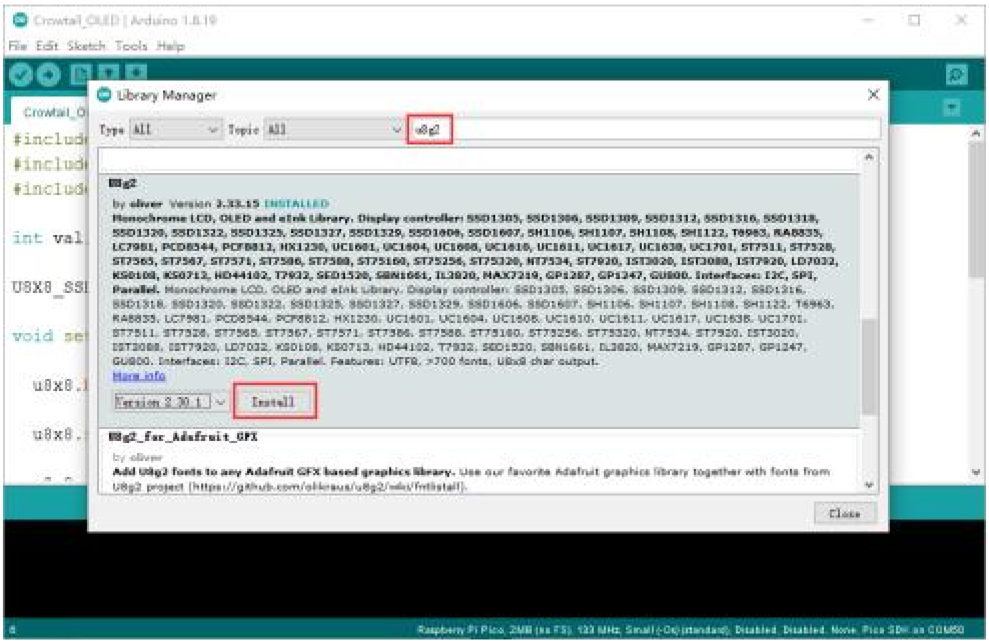
Opening the file, and modify the default definition *"define PIN_WIRE0_SDA (4u)"* and *"define PIN_WIRE0_SDA (5u)"* to *"define PIN_WIRE0_SDA (20u)"* and *"define PIN_WIRE0_SDA (21)"*.



*C:\Users\Administrator\AppData\Local\Arduino15\packages\rp2040\hardware\rp2040\2.0.0\variants\rpipico\pins_arduino.h - Notepad++

```
#define PIN_SERIAL2_TX (8u)
#define PIN_SERIAL2_RX (9u)

// SPI
#define PIN_SPI0_MISO  (16u)
#define PIN_SPI0_MOSI  (19u)
#define PIN_SPI0_SCK   (18u)
#define PIN_SPI0_SS    (17u)

#define PIN_SPI1_MISO  (12u)
#define PIN_SPI1_MOSI  (15u)
#define PIN_SPI1_SCK   (14u)
#define PIN_SPI1_SS    (13u)

// Wire
#define PIN_WIRE0_SDA  (20u)
#define PIN_WIRE0_SCL  (21u)

#define PIN_WIRE1_SDA  (26u)
#define PIN_WIRE1_SCL  (27u)

#define SERIAL_HOWMANY (3u)
#define SPI_HOWMANY    (2u)
#define WIRE_HOWMANY   (2u)

#include "../generic/common.h"
```

3.5.3 Install u8g2 library in Arduino IDE. Click *"Sketch" -> "Include Library" -> "Manage Libraries"*, then enter "u8g2" in the pop-up interface, find "u8g2" and click "Install" to complete the installation of the u8g2 library.



u8g2

U8g2

by oliver Version 2.33.15 INSTALLED

Monochrome LCD, OLED and eInk Library. Display controller: SSD1305, SSD1306, SSD1309, SSD1312, SSD1316, SSD1318, SSD1320, SSD1322, SSD1325, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, SH1107, SH1108, SH1122, T6963, RA8835, LC7981, PCD8544, PCF8812, HX1230, UC1601, UC1604, UC1608, UC1610, UC1611, UC1617, UC1638, UC1701, ST7511, ST7528, ST7565, ST7567, ST7571, ST7586, ST7588, ST75160, ST75256, ST75320, NT7534, ST7920, IST3020, IST3088, IST7920, LD7032, KS0108, KS0713, HD44102, T7932, SED1520, SBN1661, IL3820, MAX7219, GP1287, GP1247, GU800. Interfaces: I2C, SPI, Parallel. Monochrome LCD, OLED and eInk Library. Display controller: SSD1305, SSD1306, SSD1309, SSD1312, SSD1316, SSD1318, SSD1320, SSD1322, SSD1325, SSD1327, SSD1329, SSD1606, SSD1607, SH1106, SH1107, SH1108, SH1122, T6963, RA8835, LC7981, PCD8544, PCF8812, HX1230, UC1601, UC1604, UC1608, UC1610, UC1611, UC1617, UC1638, UC1701, ST7511, ST7528, ST7565, ST7567, ST7571, ST7586, ST7588, ST75160, ST75256, ST75320, NT7534, ST7920, IST3020, IST3088, IST7920, LD7032, KS0108, KS0713, HD44102, T7932, SED1520, SBN1661, IL3820, MAX7219, GP1287, GP1247, GU800. Interfaces: I2C, SPI, Parallel. Features: UTF8, >700 fonts, U8x8 char output.

Version 2.30.1 Install

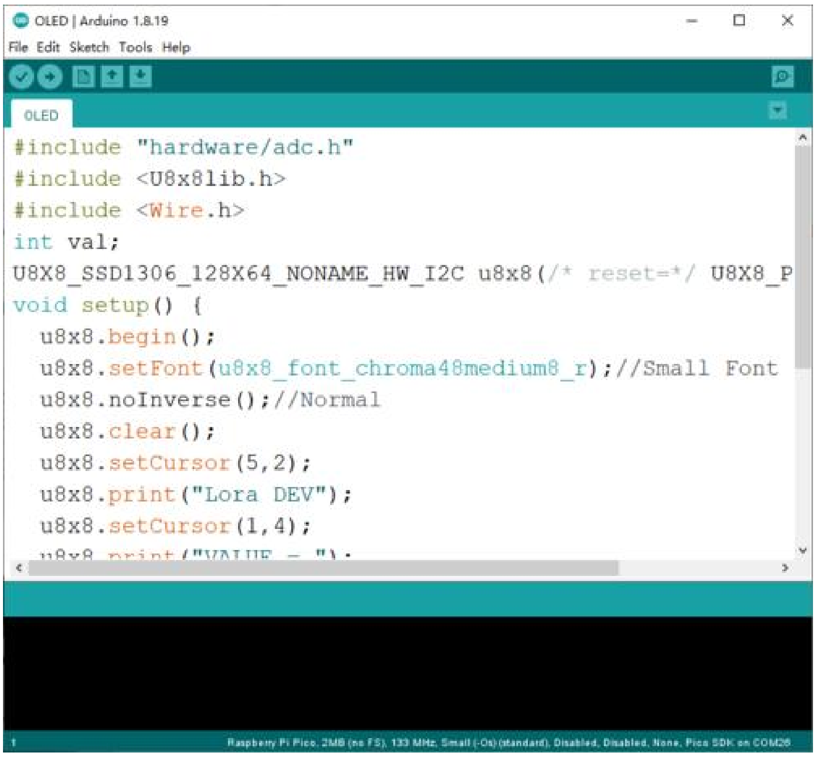
U8g2_for_Adafruit_GFX

by oliver

Add U8g2 fonts to any Adafruit GFX based graphics library. Use our favorite Adafruit graphics library together with fonts from U8g2 project (https://github.com/olikraus/u8g2/wiki/fntlistall).

Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM50

3.5.4 Open the *"Crowtail_OLED.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.



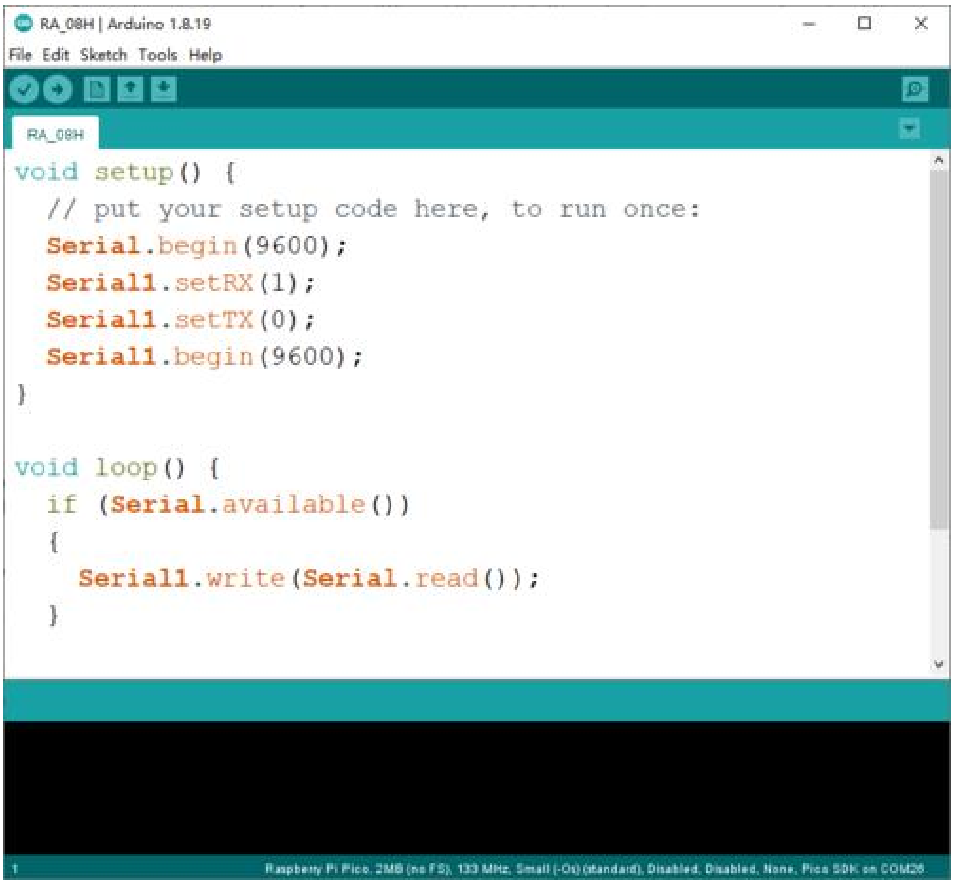
```
#include "hardware/adc.h"
#include <U8x8lib.h>
#include <Wire.h>
int val;
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_P
void setup() {
  u8x8.begin();
  u8x8.setFont(u8x8_font_chroma48medium8_r);//Small Font
  u8x8.noInverse();//Normal
  u8x8.clear();
  u8x8.setCursor(5,2);
  u8x8.print("Lora DEV");
  u8x8.setCursor(1,4);
  u8x8.print("VALUE = ");
```

Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM26

3.5.5 After power on, the value on the sliding potentiometer will be displayed on the OLED screen.

Example 6 - Communication between RP2040 and RA-08H

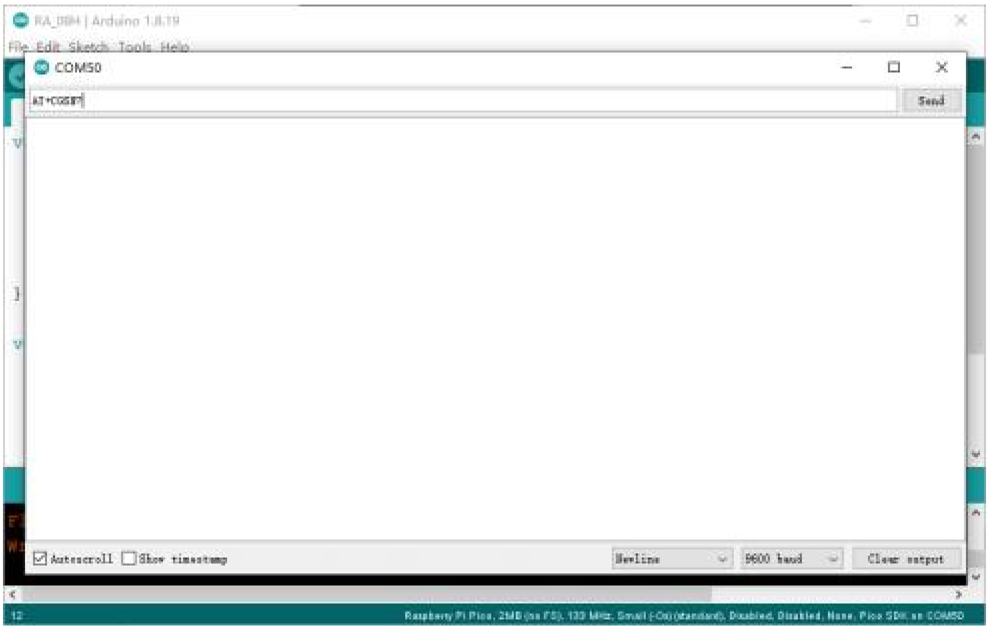
3.6.1 Open the *"RA_08H.ino"* sample program in the Arduino IDE, and then click the upload program icon to upload the program to the development board.



```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial1.setRX(1);
  Serial1.setTX(0);
  Serial1.begin(9600);
}

void loop() {
  if (Serial.available())
  {
    Serial1.write(Serial.read());
  }
```

```
1                Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os)(standard), Disabled, Disabled, None, Pico SDK on COM26
```

3.6.2 After the program is successfully uploaded, open the serial monitor and send the AT command "AT+CGSN?" to RA_08H.



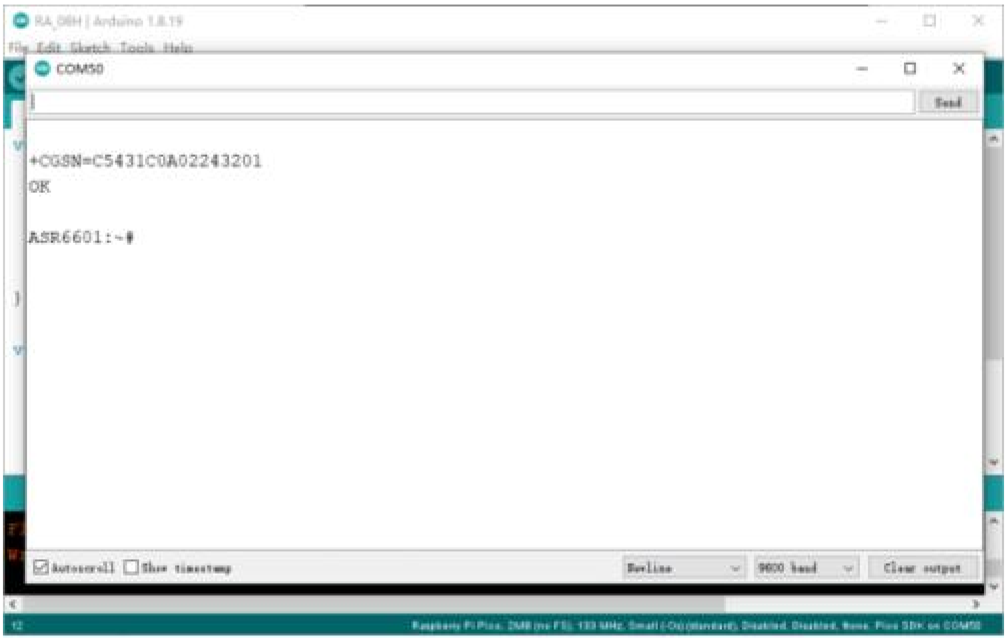
```
AT+CGSN?
```

☑ Autoscroll □ Show timestamp

Newline | 9600 baud

Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM50

3.6.3 After sending the command, if the monitor can display the returned "+CGSN=85521C0A02243201 OK" (the value returned by different RA-08H firmware is different), it proves that you can send AT commands through the Arduino IDE to control and set the RA_08H module.



```
+CGSN=C5431C0A02243201
OK

ASR6601:~#
```

☑ Autoscroll □ Show timestamp

Newline | 9600 baud

Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, None, Pico SDK on COM50