SenseCAP Indicator

User Manual

Brief Introduction

SenseCAP Indicator is a 4-inch touch screen driven by ESP32 and RP2040 dual-MCU and supports Wi-Fi/BLE/LoRa communication. It is a fully open source powerful IoT development platform for developers. One-stop ODM Fusion service is also available for customization and quick scale-up.

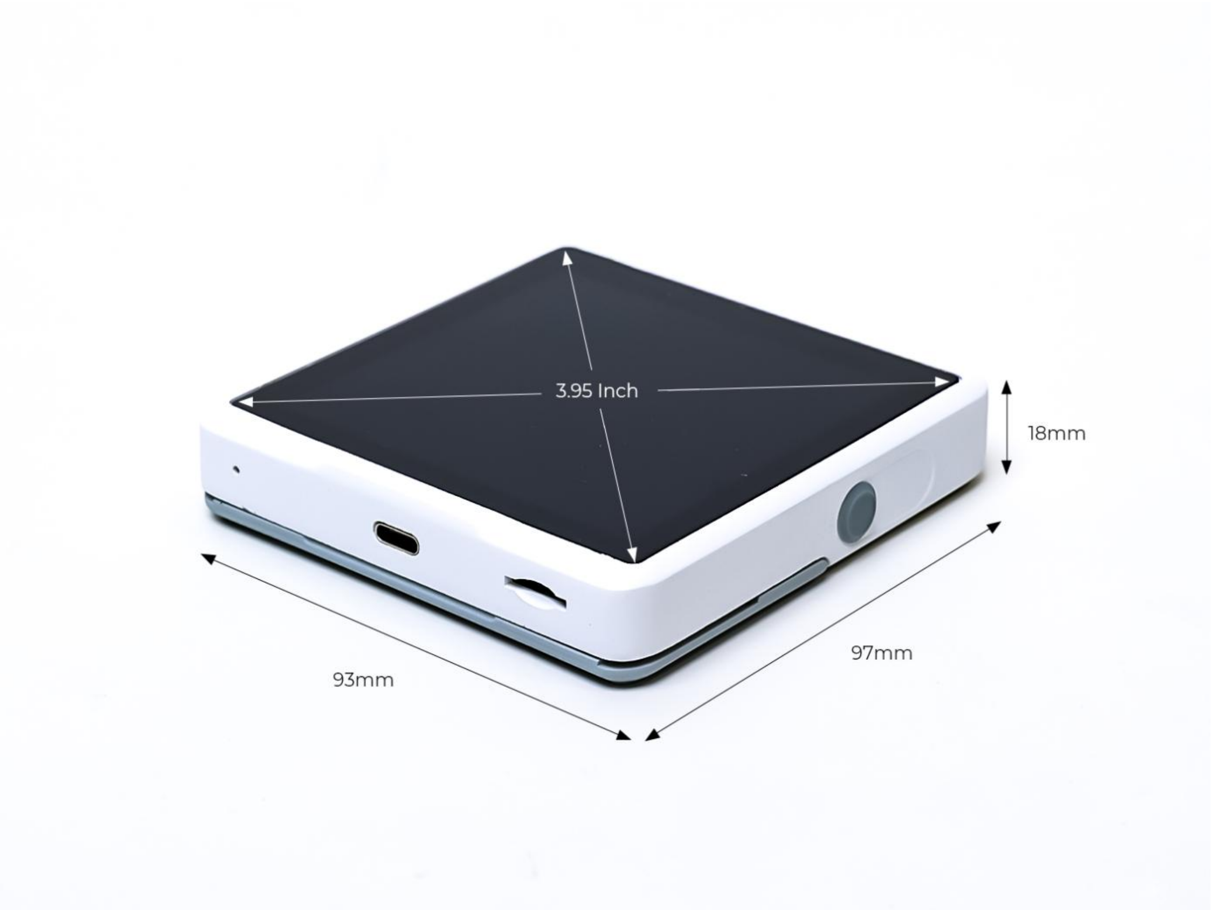
Product Video：https://youtu.be/lOdl5_MGbCw

Content

10:10
CO₂
732 ppm
tVOC
0.936 mg/m³
Temp
28.9 ℃
Humidity
78 %

Temperature
Day
Week
27.5°C
28°C
27.5°C
27.0°C
27.0°C
27.5°C
26.2°C

SENSECAP
SenseCAP Indicator
ESP32 4-Inch Open-source Touch Screen
20:23



Introduction

SenseCAP Indicator is a 4-inch touch screen driven by ESP32 and RP2040 dual-MCU and supports Wi-Fi/BLE/LoRa communication.

The board comes with Type-C and Grove interfaces, supports ADC and IIC transmission protocols, and can easily connect to other peripherals with rich GPIOs.

SenseCAP Indicator is a fully open source powerful IoT development platform for developers. One-stop ODM Fusion service is also available for customization and quick scale-up.

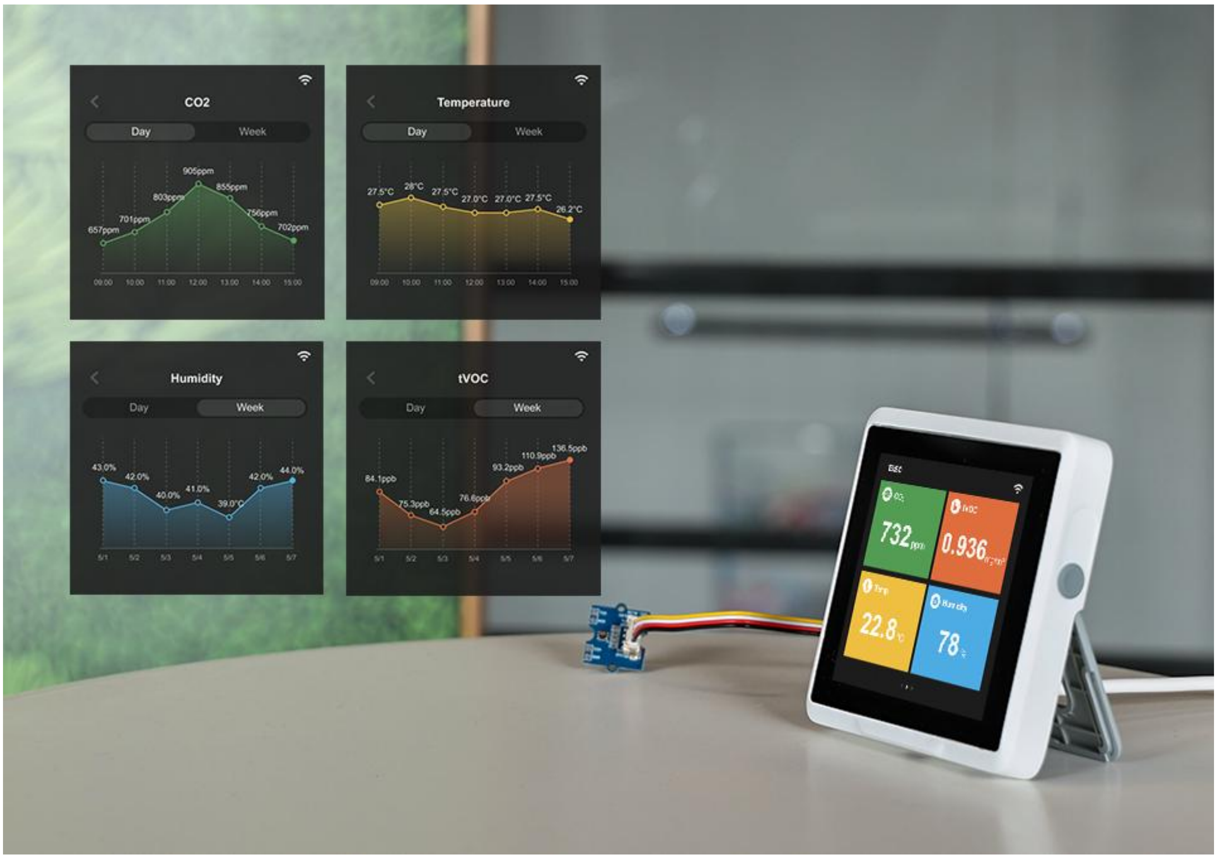
Features

- **Dual MCUs and Rich GPIOs:** Equipped with powerful ESP32S3 and RP2040 dual MCUs and over 400 Grove-compatible GPIOs for flexible expansion options.
- **Real-time Air Quality Monitoring:** Built-in tVOC and CO2 sensors, and an external

Grove AHT20 temperature and humidity sensor for more precise readings.

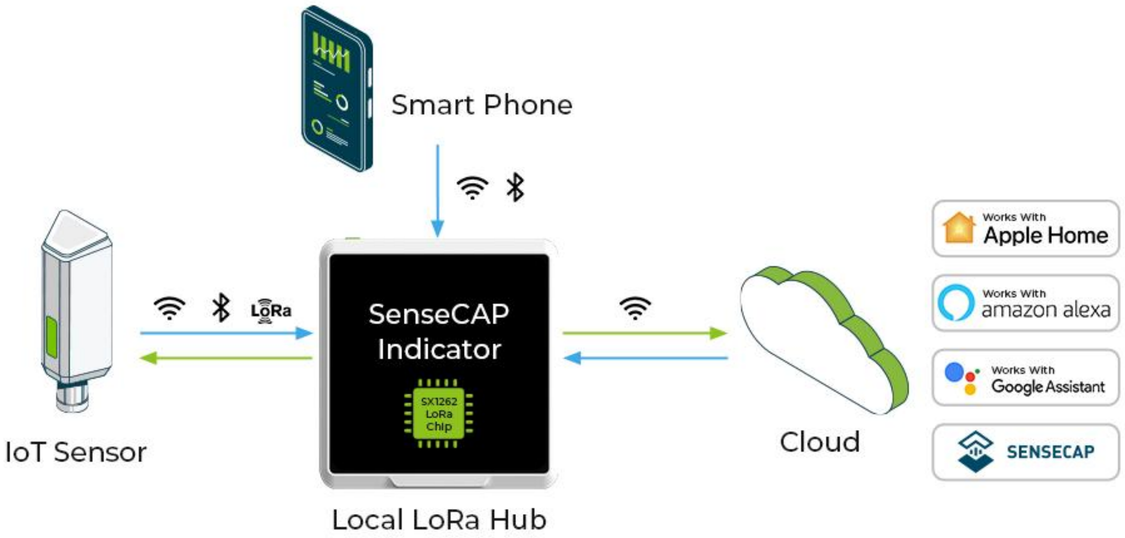
- **Local LoRa Hub for IoT Connectivity:** Integrated Semtech SX1262 LoRa chip (optional) for connecting LoRa devices to popular IoT platforms such as Matter via Wi-Fi, without the need for additional compatible devices.
- **Fully Open Source Platform:** Leverage the extensive ESP32 and Raspberry Pi open source ecosystem for infinite application possibilities.
- **Fusion ODM Service Available:** Seeed Studio also provides one-stop ODM service for quick customization and scale-up to meet various needs.

SenseCAP Indicator native firmware enables you to use it as a desktop air quality detector or you could customize one of your own styles. There are built-in tVOCs and CO2 sensors, and a Grove TH sensor is provided to connect externally to reduce heat interference to get more accurate temperature and humidity results.



The embedded SX1262 LoRa module enables you to build the LoRa application and connect your local LoRa sensors to the Cloud via Wi-Fi. For example, you could build a LoRa hub device to connect your LoRa sensors to your smart home ecosystem to implement Matter over Wi-Fi. In this way, the LoRa devices could be connected to the

Matter ecosystem via Wi-Fi, without the need to buy a new Matter-compatible device.



The software of SenseCAP Indicator is open source. Developers could create various IoT applications such as:

- Air Quality Monitor
- Weather Station Display
- Sensor Data Dashboard
- Smart Home Assistant
- Mini Sensor Hub
- Stock Price Indicator
- Digital Album
- SOP Reminder

…

Public Area Air Monitor

Weather Station Display

Asset Monitor

Grove Sensor Hub

ChatGPT Indicator

SOP Reminder

On-site Alarm

Digital Album

It can also be used to develop various exciting applications integrating with third-party platforms, such as: AWS / ChatGPT / Matter / NodeRED / IoTex ...

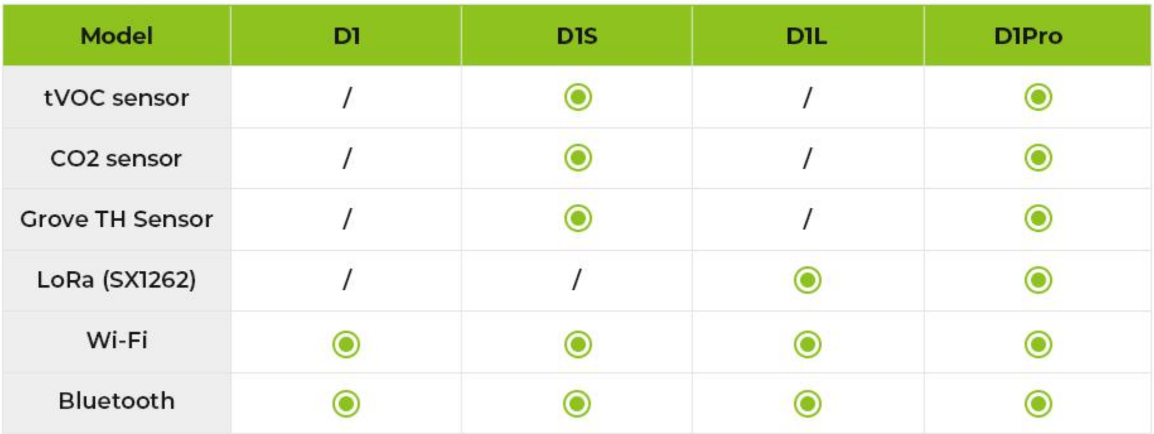
Note: This sketch shows the development possibilities for the related IoT ecosystem, developers need to develop the firmware based on need.

SenseCAP Indicator is an interactive display and control terminal for various IoT applications. It provides developers with a powerful and interface-rich development platform powered by ESP32-S3 and RP2040 dual-MCU.

Don't wait anymore, start your wonderful development journey with SenseCAP Indicator now!

Hardware Version

The SenseCAP Indicator series offers four different versions: D1, D1S, D1L, and D1Pro. Each version is designed to meet different application needs without any extra cost from unnecessary hardware. Here are the differences between the versions:



Model	D1	D1S	D1L	D1Pro
tVOC sensor	/	◉	/	◉
CO2 sensor	/	◉	/	◉
Grove TH Sensor	/	◉	/	◉
LoRa (SX1262)	/	/	◉	◉
Wi-Fi	◉	◉	◉	◉
Bluetooth	◉	◉	◉	◉

Specification

Screen	3.95 Inch, Capacitive RGB Touch Screen
Screen Resolution	480 x 480 pixels
Power Supply	5V DC, 1A
Battery	N/A
Processor	ESP32-S3: Xtensa® dual-core 32-bit LX7 microprocessor, up to 240

	MHz RP2040: Dual ARM Cortex-M0+ up to 133MHz	
Flash	ESP32S3: 8MB RP2040: 2MB	
External Storage	Support up to 32GB Micro SD Card (not included)	
Wi-Fi	802.11b/g/n, 2.4GHz	
Bluetooth	Bluetooth 5.0 LE	
LoRa(SX1262)	LoRaWAN/(G)FSK, 21dBm Max Transmitted Power Sensitivity/-136dBm@SF12 BW=125KHz RX Sensitivity, Up to 5km (depending on gateway antenna and environments)	
Sensors(Optional)	Built-in CO2 (SCD41)	Range: 0-40000ppm Accuracy: 400ppm – 5000ppm ±(50ppm+5% of reading)
	Built-in TVOC (SGP40)	Range: 1-500 VOC Index Points
	External Temperature and Humidity Sensor	Grove TH Sensor (AHT20) Sensor Range: -40 ~ + 85 ℃/± 0.3 ℃; 0 ~ 100% RH/± 2% RH (25 ℃)
Buzzer	MLT-8530, Resonant Frequency：2700Hz	

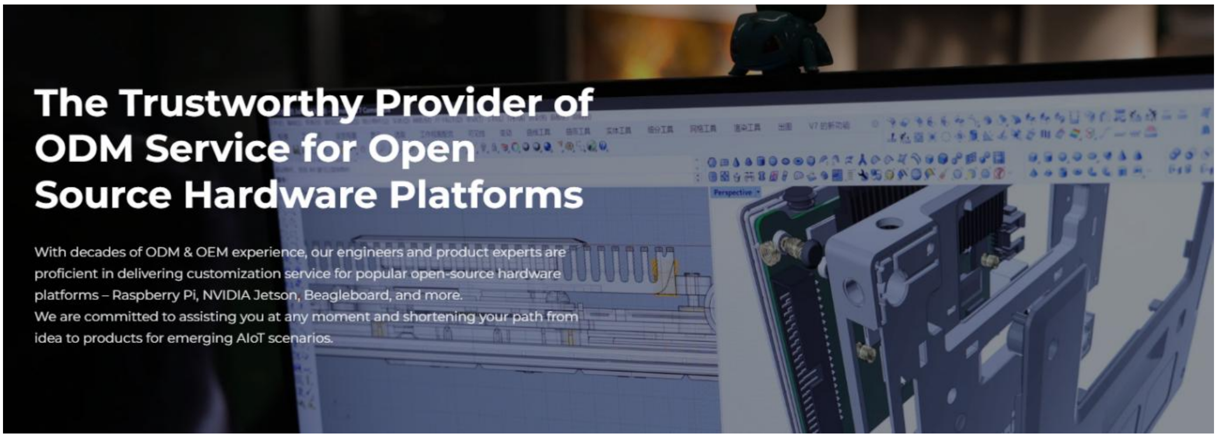
ODM Service

Note:

If you want to customize your own SenseCAP Indicator, Seeed Studio provides one-stop ODM service for quick customization and scalp-up. Please contact iot@seeed.cc if you have large quantity customization need.

With over 10 years of ODM & OEM experience, our engineers and product experts are proficient in delivering customization service for popular open-source hardware platforms, such as: Raspberry Pi, NVIDIA Jetson, Beagleboard, ESP32, and more.

We are committed to assisting you at any moment and shortening your path from idea to products for emerging AIoT scenarios.

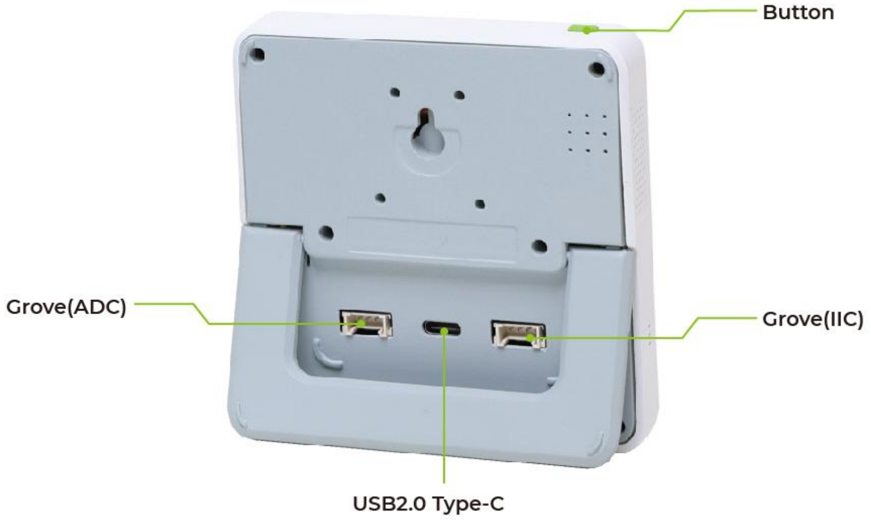


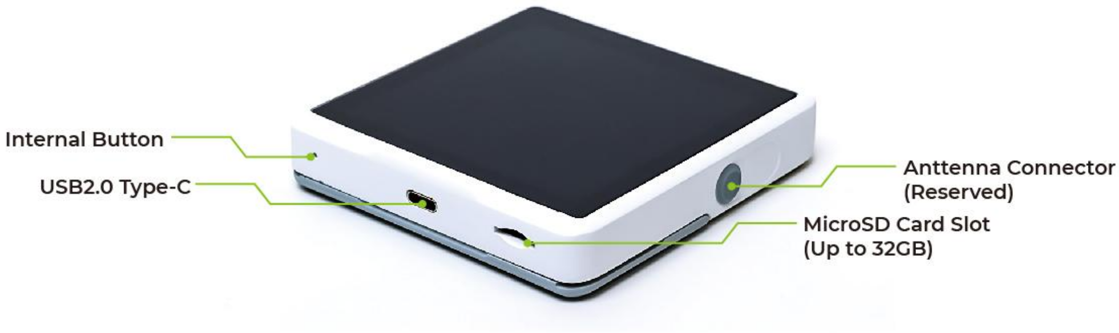
The Trustworthy Provider of ODM Service for Open Source Hardware Platforms

With decades of ODM & OEM experience, our engineers and product experts are proficient in delivering customization service for popular open-source hardware platforms – Raspberry Pi, NVIDIA Jetson, Beagleboard, and more.
We are committed to assisting you at any moment and shortening your path from idea to products for emerging AIoT scenarios.

Get Started with SenseCAP Indicator

Hardware Overview





Button Function

Short press: Turn off /wake up the screen

Long press for 3s: Switch on/switch off the device

Long press for 10s: Factory reset

Grove Interfaces

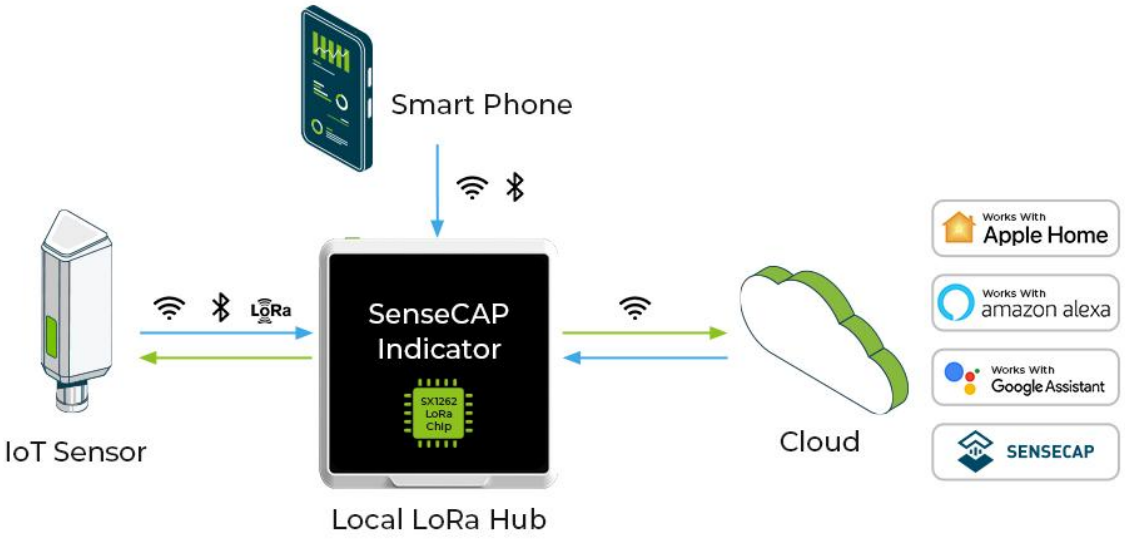
There are two Grove interfaces for connecting Grove modules, providing more possibilities for developers.

Grove is a modular, standardized connector prototyping system and a strong open source hardware ecosystem. （Click Grove – Seeed Studio to learn more）

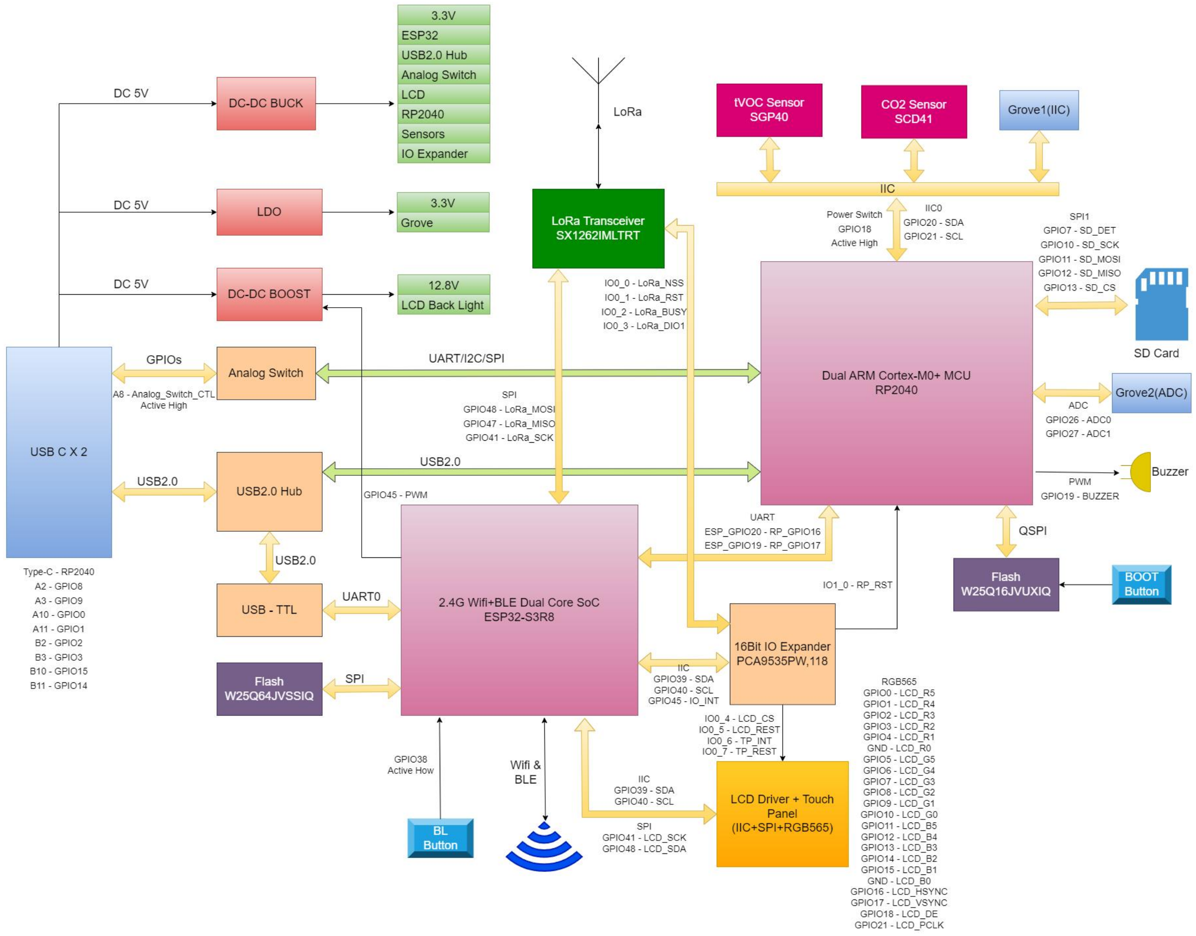
Grove
GROW YOUR IDEA
An open source, modular, and ready-to-use tool set optimized for simplicity and rapid prototyping.
With Grove, you can prove and enhance your ideas quickly and simplely.
You can even create your own Grove boards.

SX1262 LoRa Chip

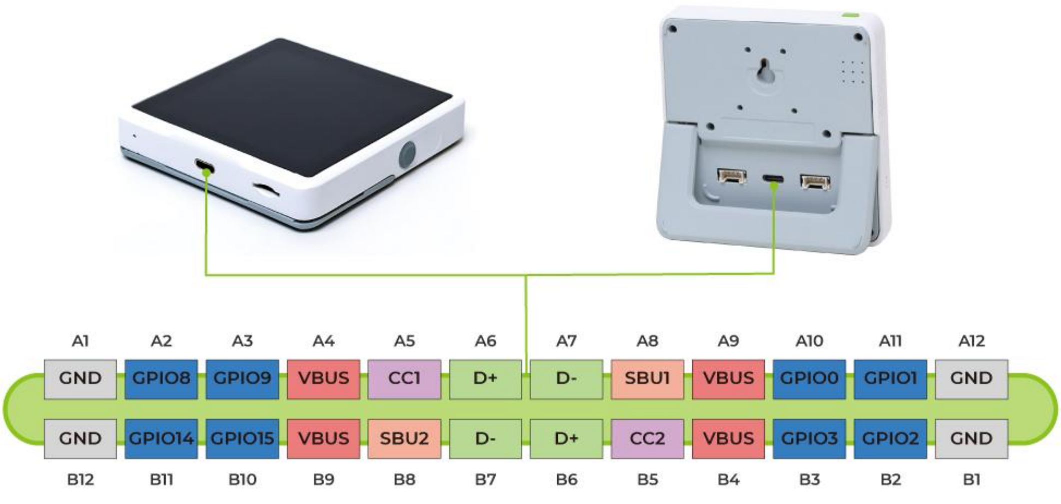
The embedded SX1262 LoRa module enables you to build the LoRa application and connect your local LoRa sensors to the Cloud via Wi-Fi. For example, you could build a LoRa hub device to connect your LoRa sensors to your smart home ecosystem to implement Matter over Wi-Fi. In this way, the LoRa devices could be connected to the Matter ecosystem via Wi-Fi, without the need to buy a new Matter-compatible device.



System Diagram



USB2.0 Type-C Pinout



A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	GPIO8	GPIO9	VBUS	CC1	D+	D-	SBU1	VBUS	GPIO0	GPIO1	GND
GND	GPIO14	GPIO15	VBUS	SBU2	D-	D+	CC2	VBUS	GPIO3	GPIO2	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

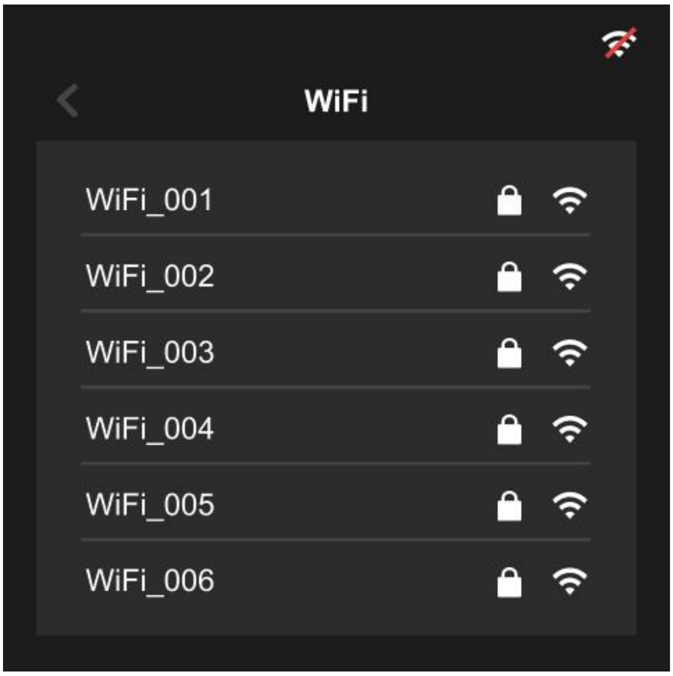
Note: A2, A3, A10, A11, B2, B3, B10, B11 connect to RP2040 GPIO8, GPIO9, GPIO0, GPIO1, GPIO2, GPIO3, GPIO15, GPIO14

Native Firmware

SenseCAP Indicator native firmware enables you to use it as a desktop air quality detector，just a few simple steps to enjoy.

Wi–Fi Configuration

Plug the 5V power adaptor into the USB type–C power connector, the device will power on automatically, then the display will show the WiFi setting page.



WiFi

WiFi_001

WiFi_002

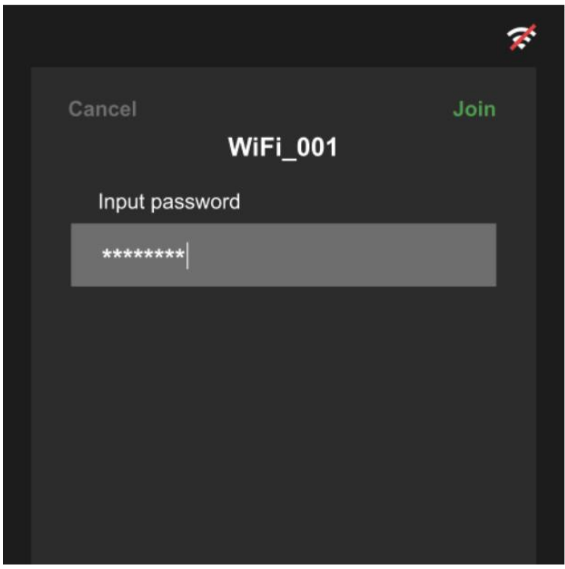
WiFi_003

WiFi_004

WiFi_005

WiFi_006

Select the WiFi you need, input the password and connect.

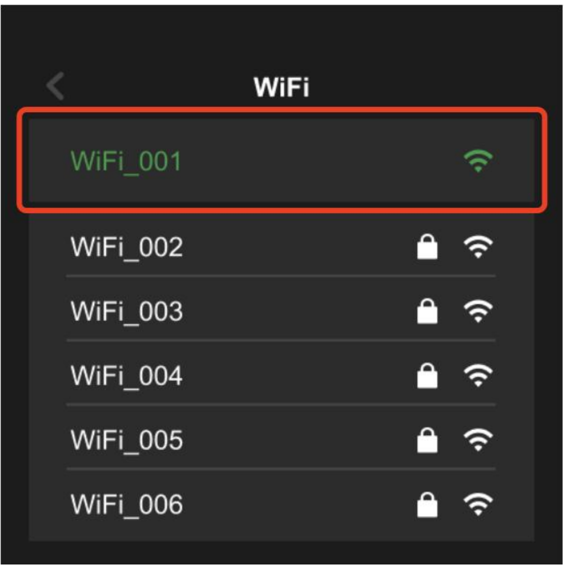


Cancel Join

WiFi_001

Input password

****



WiFi

WiFi_001

WiFi_002

WiFi_003

WiFi_004

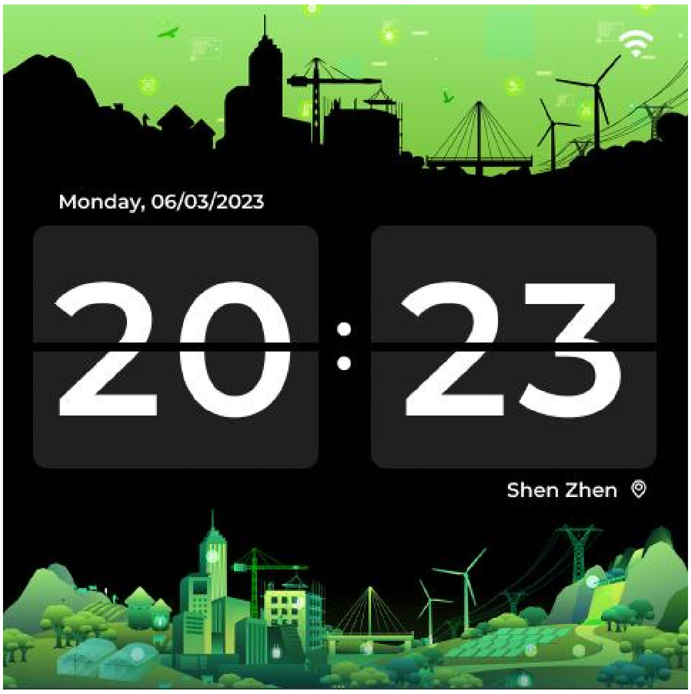
WiFi_005

WiFi_006

Data Viewing

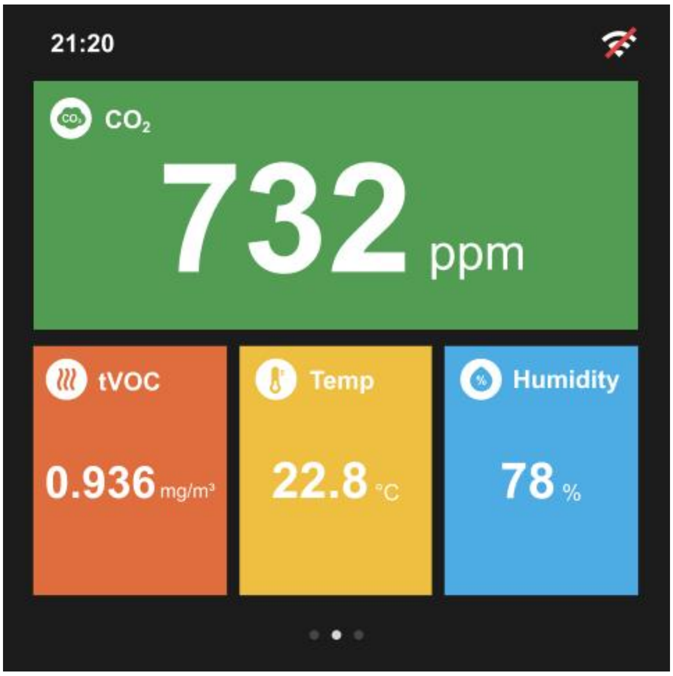
- **Home page**

Including the time, date, location info.

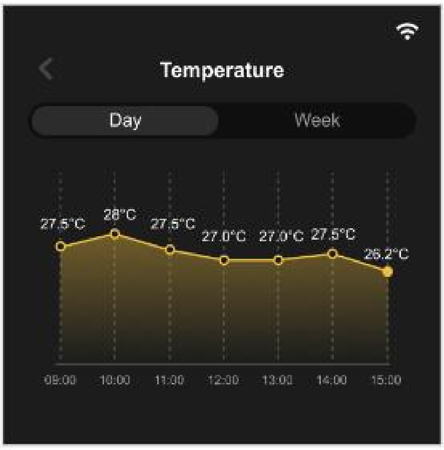


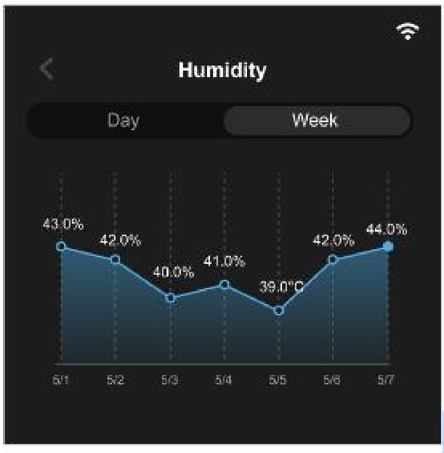
- **Sensor data**

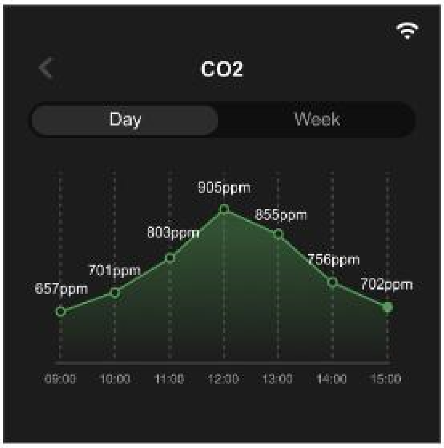
Built–in tVOC and CO2 sensors, and an external Grove AHT20 TH sensor for more precise temperature and humidity readings.

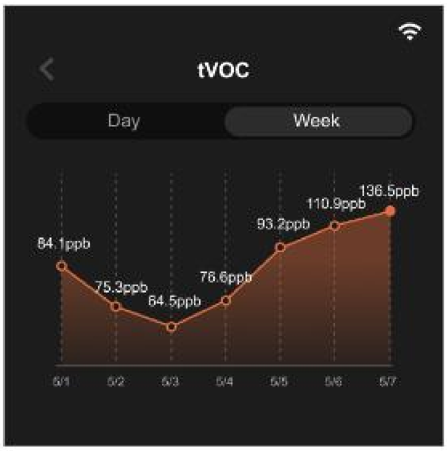


Click the specified sensor to enter the detailed information page, and you can choose to display the value of 24h or a week.

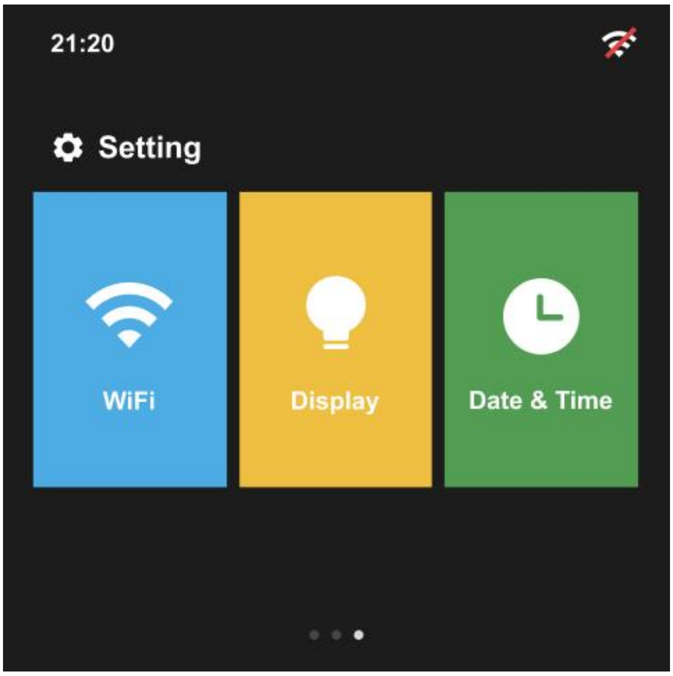








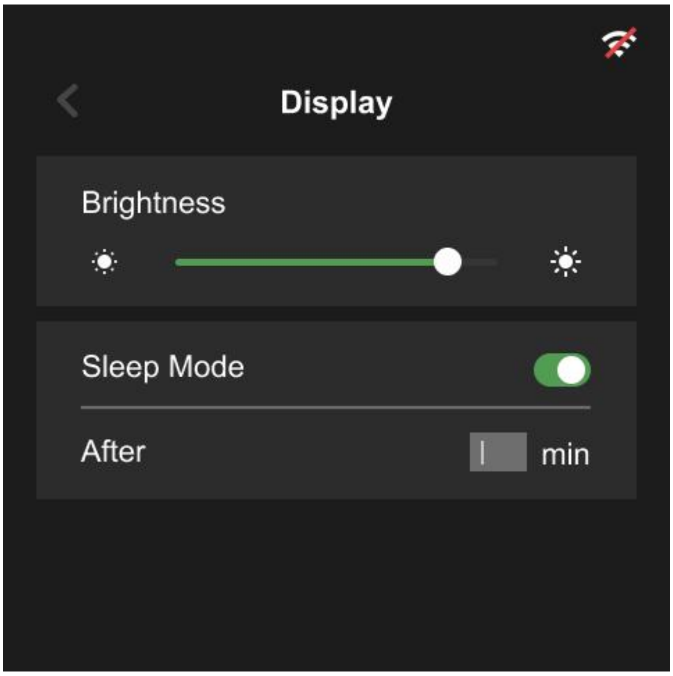
Setting



Display Setting

Brightness: Adjust screen brightness

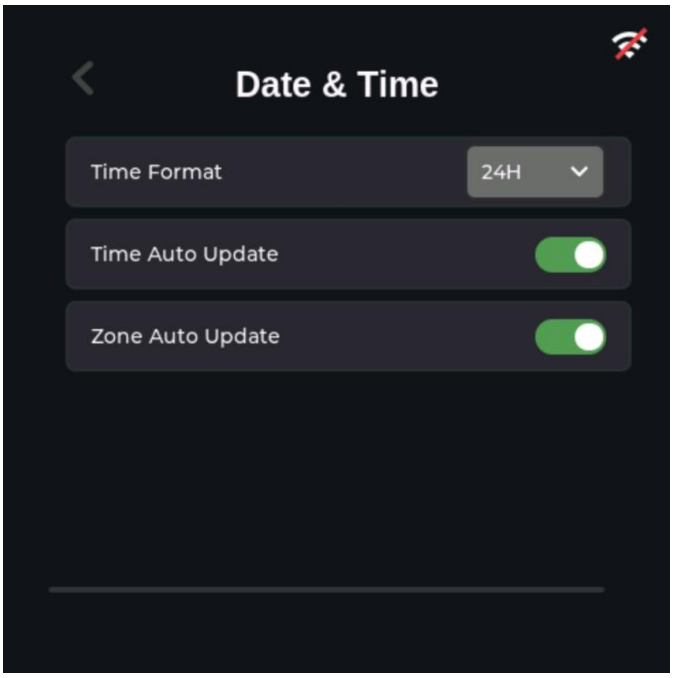
Sleep Mode: Turn off the screen according to the interval you set



Date & Time Setting

Time Format: You can set 24H or 12H format.

Time Auto Update/Zone Auto Update: When the device is connected to WiFi, it will automatically obtain the corresponding time zone and date.



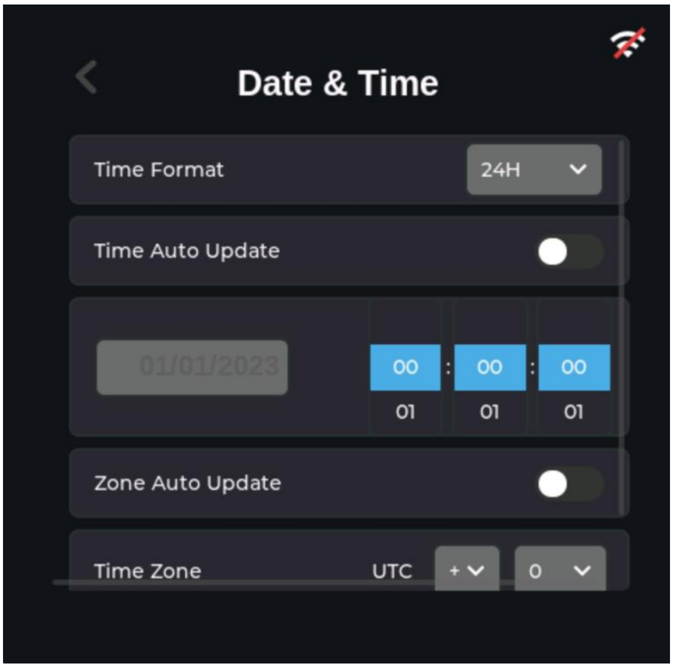
Date & Time

Time Format: 24H

Time Auto Update

Zone Auto Update

Manual setting: If the time obtained through the time zone cannot automatically identify the winter time or device is offline, then you can manually set the time zone



Date & Time

Time Format: 24H

Time Auto Update

01/01/2023

00 : 00 : 00

01 01 01

Zone Auto Update

Time Zone: UTC + 0

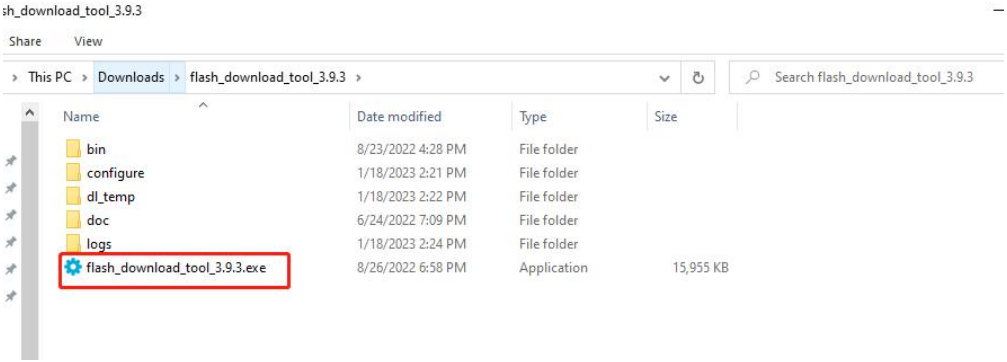
Development Tutorial

ESP32 Firmware Flashing

Flash by Flash Download Tools (For Windows only)

- **Step 1**: Connect the device to your PC with the provided USB type-C cable.
- **Step 2**:Install the Flash Download Tools

  Flash Download Tools for Windows



Name	Date modified	Type	Size
bin	8/23/2022 4:28 PM	File folder	
configure	1/18/2023 2:21 PM	File folder	
dl_temp	1/18/2023 2:22 PM	File folder	
doc	6/24/2022 7:09 PM	File folder	
logs	1/18/2023 2:24 PM	File folder	
flash_download_tool_3.9.3.exe	8/26/2022 6:58 PM	Application	15,955 KB

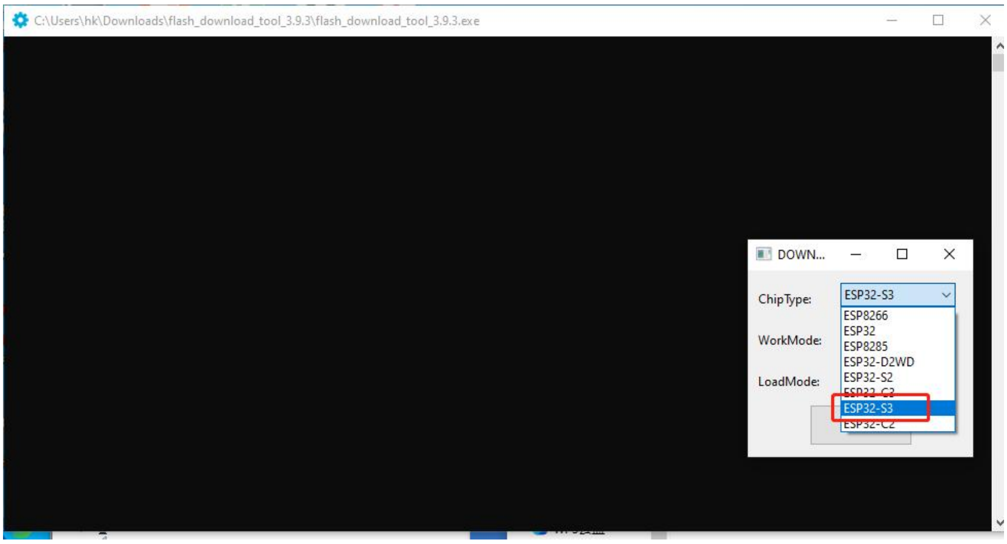
- **Step 3**: User interface setting

Double-click the .exe file to enter the main interface of the tool.

Chip Type: ESP32-S3

WorkMode: Develop

LoadMode: UART

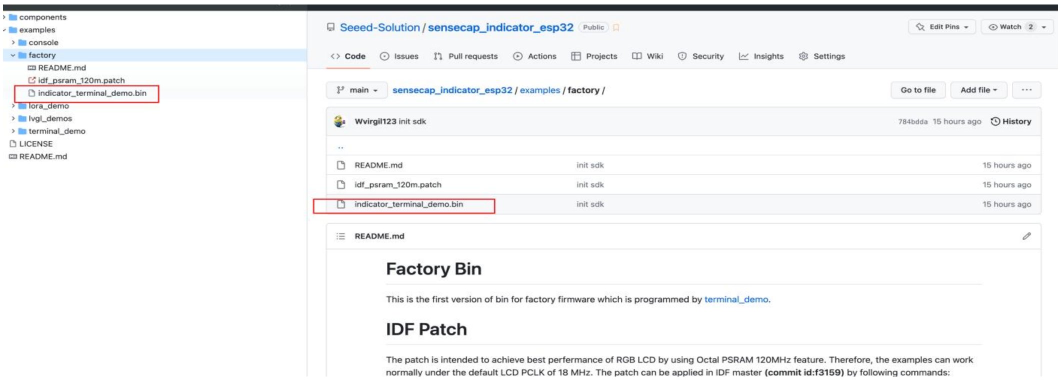


ChipType: ESP32-S3

- **Step 4**: SPI Download Tab Configure

Download the Esp32 firmware file.

Example–factory–indicator_terminal_demo.bin



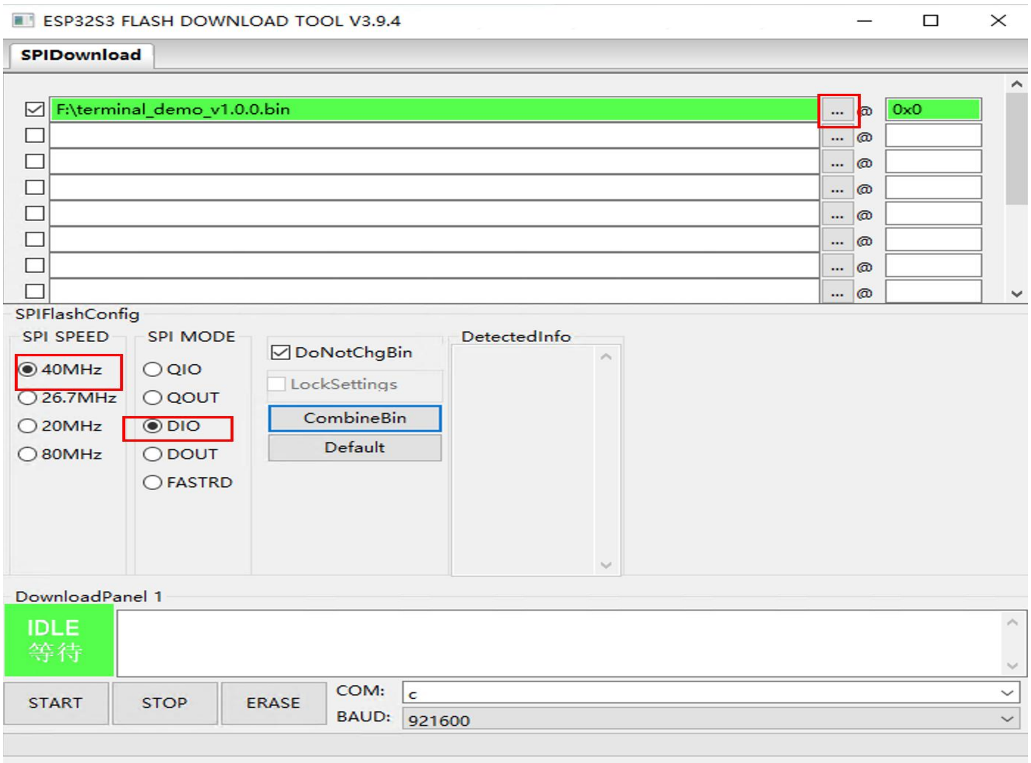
Factory Bin

This is the first version of bin for factory firmware which is programmed by terminal_demo.

IDF Patch

The patch is intended to achieve best performance of RGB LCD by using Octal PSRAM 120MHz feature. Therefore, the examples can work normally under the default LCD PCLK of 18 MHz. The patch can be applied in IDF master **(commit id:f3159)** by following commands:

Click "..." and select the .bin file, downloading addresses:0x0



☑ F:\terminal_demo_v1.0.0.bin ... @ 0x0

SPI SPEED: ● 40MHz ○ 26.7MHz ○ 20MHz ○ 80MHz

SPI MODE: ○ QIO ○ QOUT ● DIO ○ DOUT ○ FASTRD

☑ DoNotChgBin

□ LockSettings

COM: c

BAUD: 921600

- **Step 5**: SPI Flash Configure

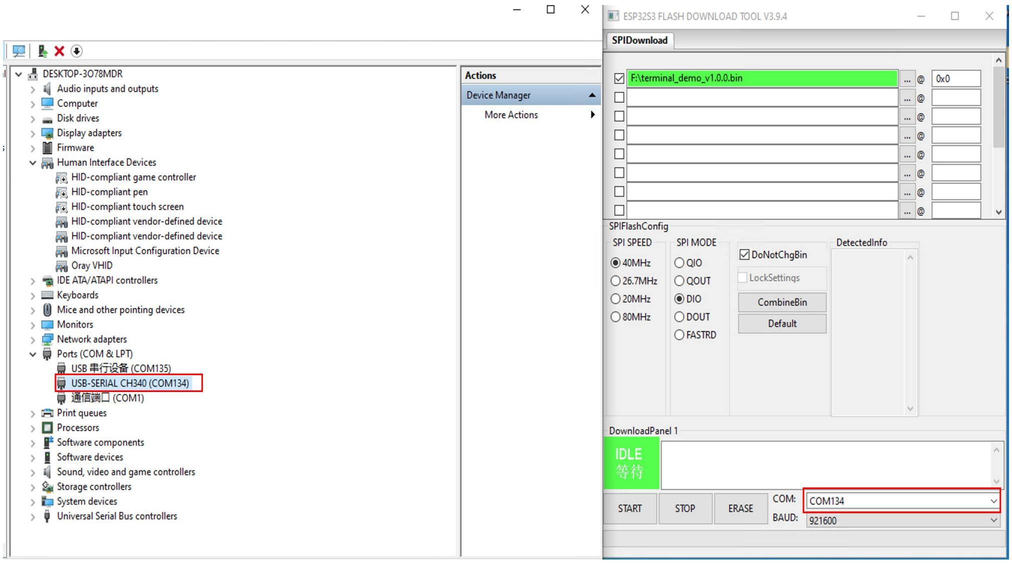
SPI SPEED: 40MHz

SPI MODE: DIO

- **Step 6**: Download Panel Configure

COM: Check the ports on your Device Manage, the USB-SERIAL is the correct one.

Baud: 921600(recommended value)



- DESKTOP-3O78MDR
  - Audio inputs and outputs
  - Computer
  - Disk drives
  - Display adapters
  - Firmware
  - Human Interface Devices
    - HID-compliant game controller
    - HID-compliant pen
    - HID-compliant touch screen
    - HID-compliant vendor-defined device
    - HID-compliant vendor-defined device
    - Microsoft Input Configuration Device
    - Oray VHID
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - USB 串行设备 (COM135)
    - USB-SERIAL CH340 (COM134)
    - 通信端口 (COM1)
  - Print queues
  - Processors
  - Software components
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

☑ F:\terminal_demo_v1.0.0.bin @ 0x0

SPIFlashConfig — SPI SPEED: ● 40MHz ○ 26.7MHz ○ 20MHz ○ 80MHz; SPI MODE: ○ QIO ○ QOUT ● DIO ○ DOUT ○ FASTRD; ☑ DoNotChgBin □ LockSettings

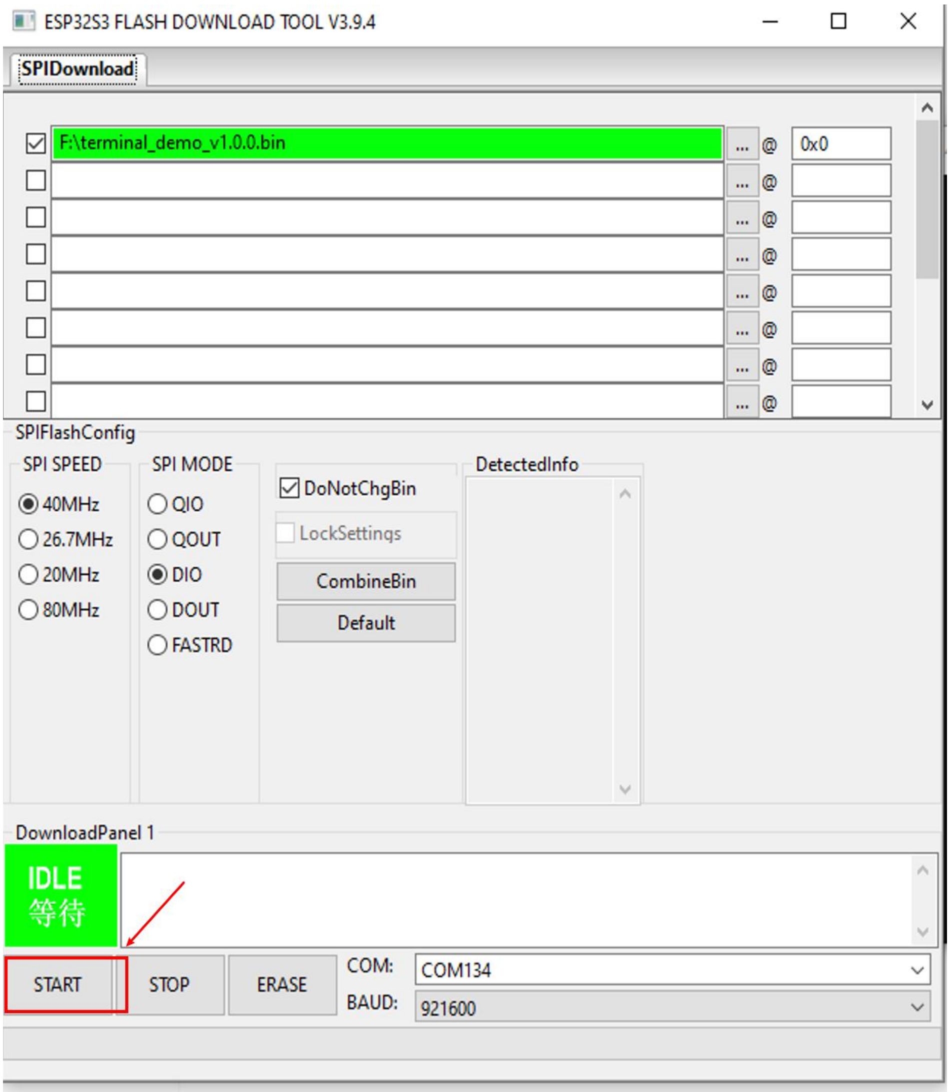
DownloadPanel 1: IDLE 等待

COM: COM134

BAUD: 921600

- **Step 6**: Start Downloading

Then click "START" to start the downloading.



☑ F:\terminal_demo_v1.0.0.bin @ 0x0

SPIFlashConfig

SPI SPEED: ● 40MHz ○ 26.7MHz ○ 20MHz ○ 80MHz

SPI MODE: ○ QIO ○ QOUT ● DIO ○ DOUT ○ FASTRD

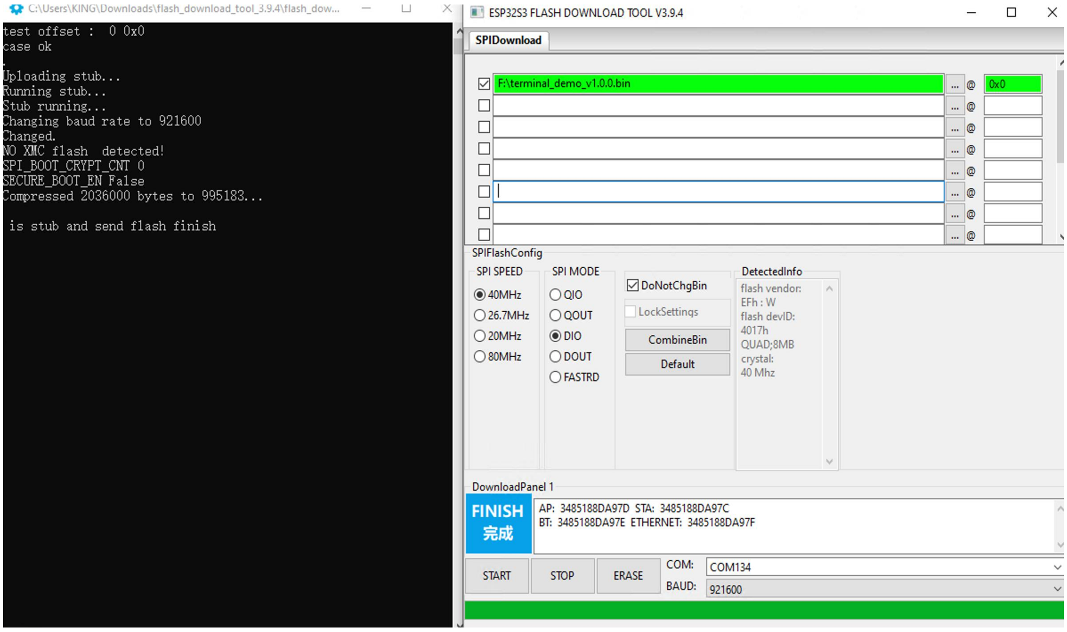
☑ DoNotChgBin □ LockSettings

DownloadPanel 1: IDLE 等待

COM: COM134

BAUD: 921600

When it shows "FINISH", the flash has been completed.



C:\Users\KING\Downloads\flash_download_tool_3.9.4\flash_dow...
ESP32S3 FLASH DOWNLOAD TOOL V3.9.4
SPIDownload
SPIFlashConfig
DownloadPanel 1
FINISH
完成
START
STOP
ERASE

```
test offset :  0 0x0
case ok
.
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
NO XMC flash  detected!
SPI_BOOT_CRYPT_CNT 0
SECURE_BOOT_EN False
Compressed 2036000 bytes to 995183...

 is stub and send flash finish
```

☑ F:\terminal_demo_v1.0.0.bin @ 0x0

SPI SPEED: ● 40MHz ○ 26.7MHz ○ 20MHz ○ 80MHz

SPI MODE: ○ QIO ○ QOUT ● DIO ○ DOUT ○ FASTRD

☑ DoNotChgBin

□ LockSettings

CombineBin

Default

DetectedInfo

flash vendor:
EFh : W
flash devID:
4017h
QUAD;8MB
crystal:
40 Mhz

AP: 3485188DA97D STA: 3485188DA97C
BT: 3485188DA97E ETHERNET: 3485188DA97F

COM: COM134

BAUD: 921600

Flash by IDF

For Linux and MacOS: Standard Toolchain Setup for Linux and macOS

For Windows: Standard Setup of Toolchain for Windows

After you have installed the IDF environment, you can flash it by IDF.

```
esptool.py write_flash 0x0 terminal_demo_v1.0.0.bin
```

also you can build, flash and monitor the project.

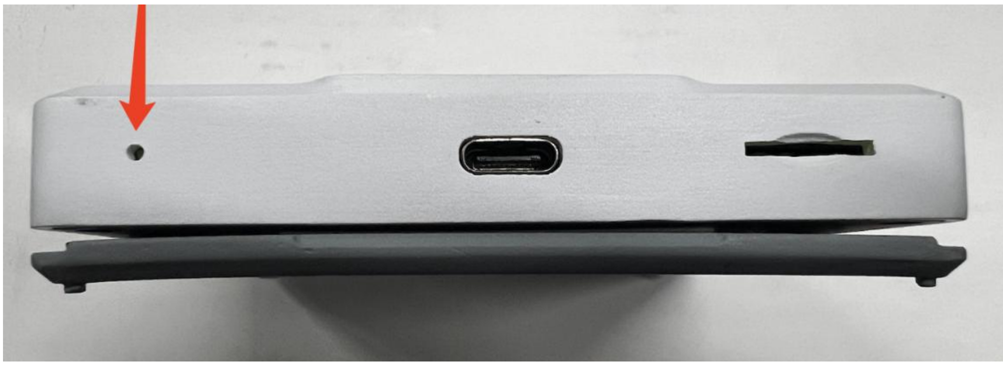
```
cd  <sdk>/examples/terminal_demo/

idf.py -p PORT build flash monitor
```

RP2040 Firmware Flashing

Flash by .uf2 file

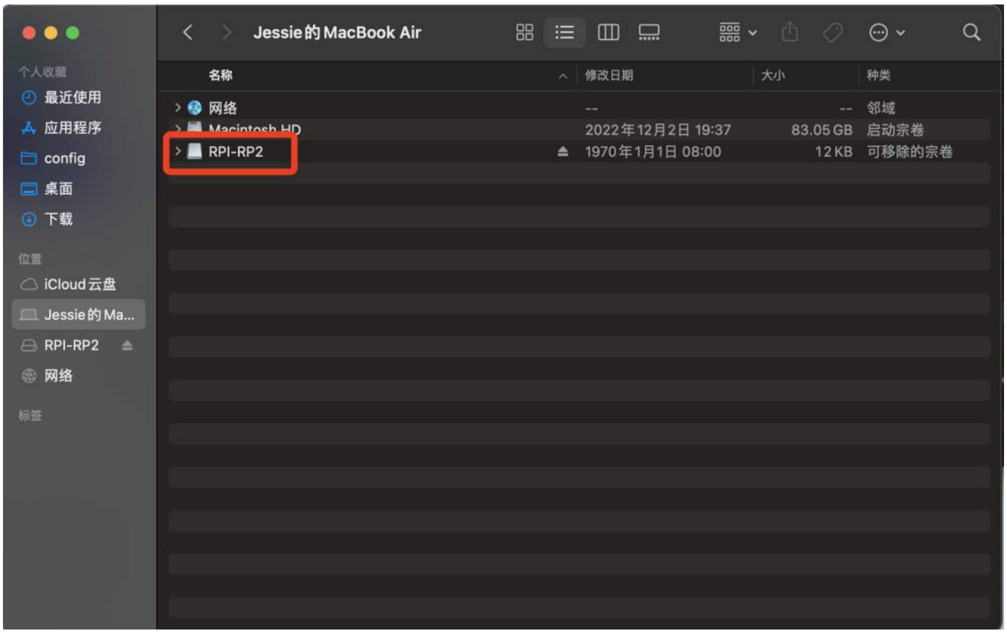
- **Step 1**: Connect the device to your PC

Long press this internal button, then connect the device to your PC by the provided USB type-C cable, release the button once connected.



- **Step 2**: Firmware Flash

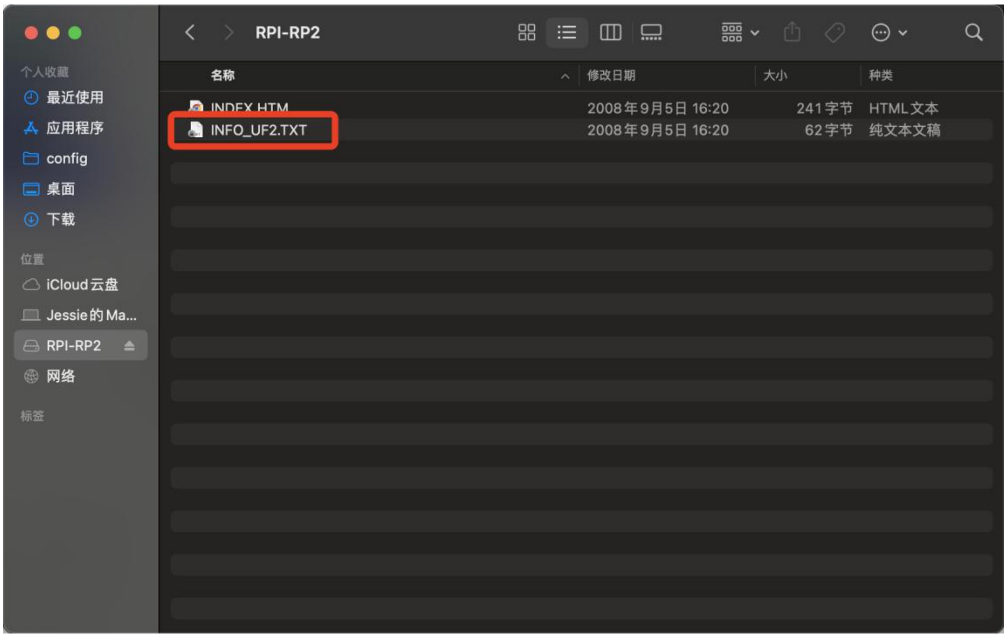
After the connection is successful, your PC will show a disk.



名称	修改日期	大小	种类
网络	--	--	邻域
Macintosh HD	2022年12月2日 19:37	83.05 GB	启动宗卷
RPI-RP2	1970年1月1日 08:00	12 KB	可移除的宗卷

Download rp2040 uf2 ,copy the .uf2 file in the firmware package to the disk, then the disk will log out.

The upgrade will run automatically.



名称	修改日期	大小	种类
INDEX.HTM	2008年9月5日 16:20	241字节	HTML文本
INFO_UF2.TXT	2008年9月5日 16:20	62字节	纯文本文稿

Flash by Arduino IDE

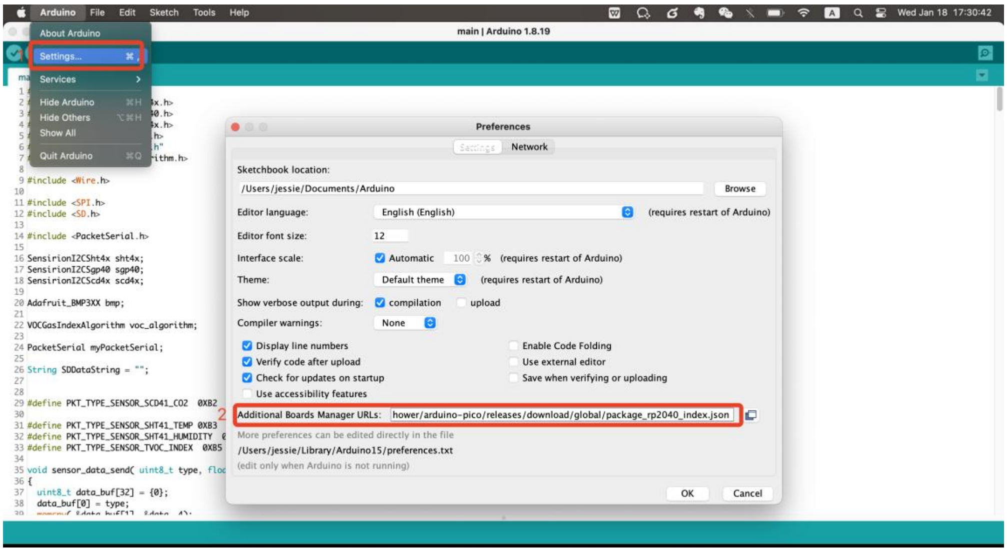
- **Step 1:** Install Arduino IDE

Arduino IDE

- **Step 2**: Add the Raspberry Pi Pico Board

Open your Arduino IDE, click on **Arduino** > **Settings**, and copy the below URL to **Additional Boards Manager URLs**:

https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json



Sketchbook location: /Users/jessie/Documents/Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation □ upload

Compiler warnings: None

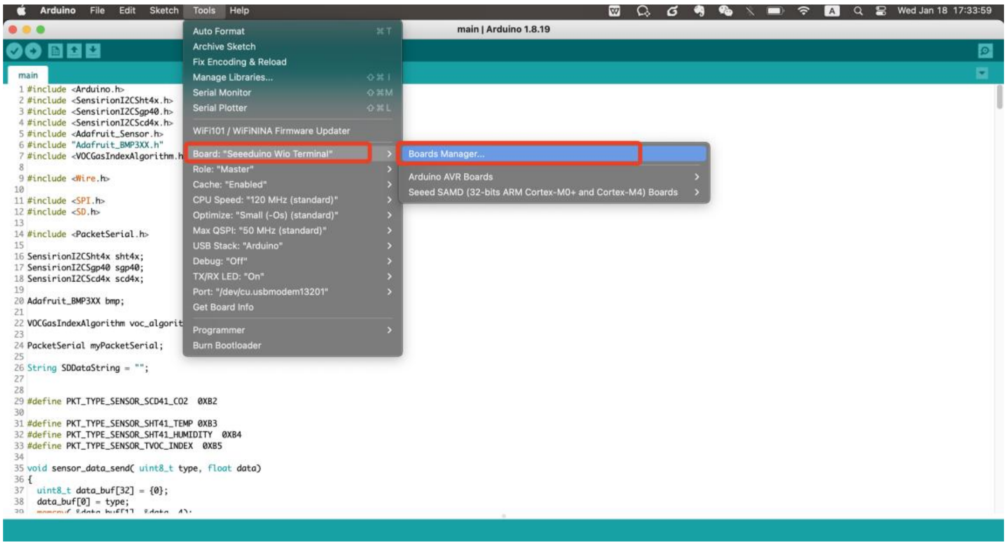
☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features
□ Enable Code Folding
□ Use external editor
□ Save when verifying or uploading

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

More preferences can be edited directly in the file
/Users/jessie/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

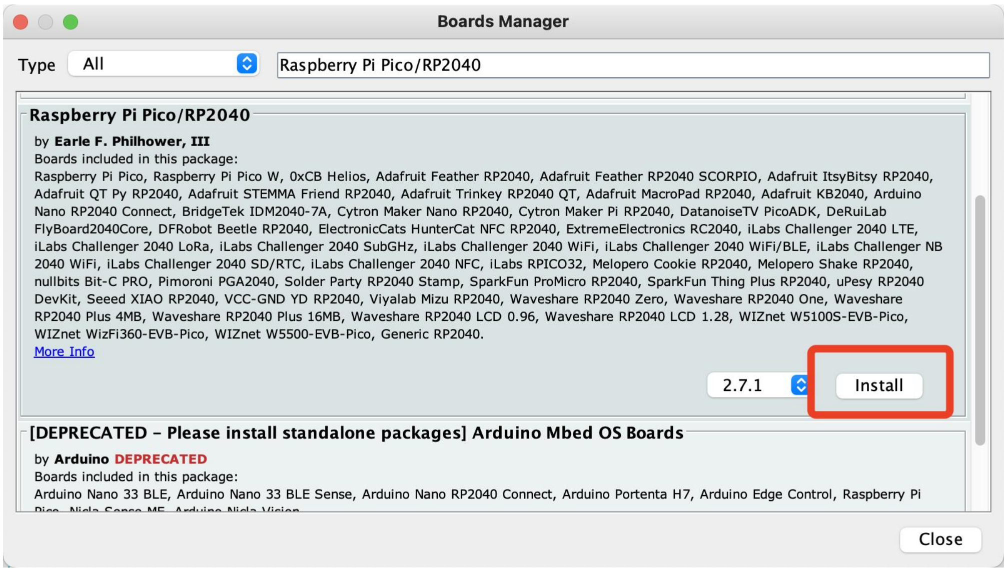
```
#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0
#define PKT_TYPE_SENSOR_TVOC_INDEX  0XB5

void sensor_data_send( uint8_t type, floa
{
  uint8_t data_buf[32] = {0};
  data_buf[0] = type;
```

Click on **Tools** > **Board** > **Board Manager**.



```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorit

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
#define PKT_TYPE_SENSOR_TVOC_INDEX  0XB5

void sensor_data_send( uint8_t type, float data)
{
  uint8_t data_buf[32] = {0};
  data_buf[0] = type;
```

Search and install "Raspberry Pi Pico/RP2040" in the Boards Manager



Raspberry Pi Pico/RP2040

by **Earle F. Philhower, III**
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
Boards included in this package:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect, Arduino Portenta H7, Arduino Edge Control, Raspberry Pi Pico, Nicla Sense ME, Arduino Nicla Vision

- **Step 3**: Add Libraries

Download these libraries we need:

Serial communication protocol

SGP40 TVOC sensor library

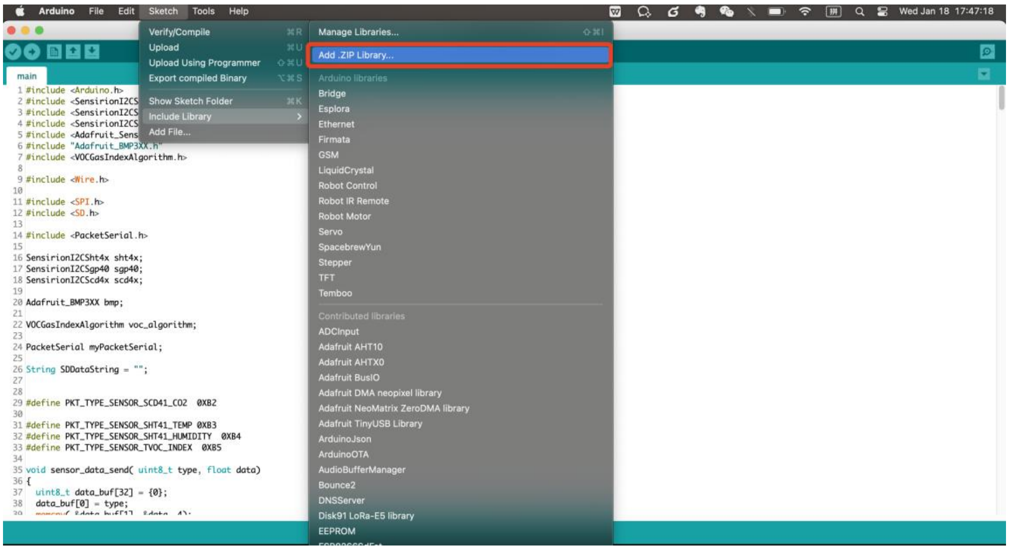
Transfer index library: Sensirion Gas Index Algorithm

SCD41 CO2 sensor library

AHT20 temperature and humidity sensor libray

Sensirion Arduino Core library

Navigate to **Sketch** –> **Include Library** –> **Add .ZIP Library**, then select the libraries you download.

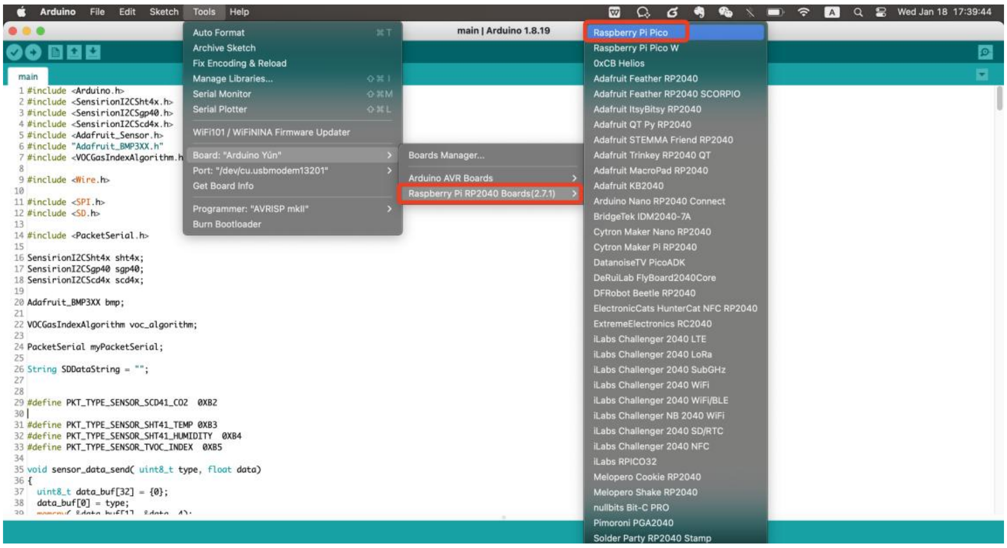


```
#include <Arduino.h>
#include <SensirionI2CS
#include <SensirionI2CS
#include <SensirionI2CS
#include <Adafruit_Sens
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h>

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
#define PKT_TYPE_SENSOR_TVOC_INDEX  0XB5

void sensor_data_send( uint8_t type, float data)
{
  uint8_t data_buf[32] = {0};
  data_buf[0] = type;
```

- **Step 4**: Connect the device to your PC with the provided USB Typc–C cable.

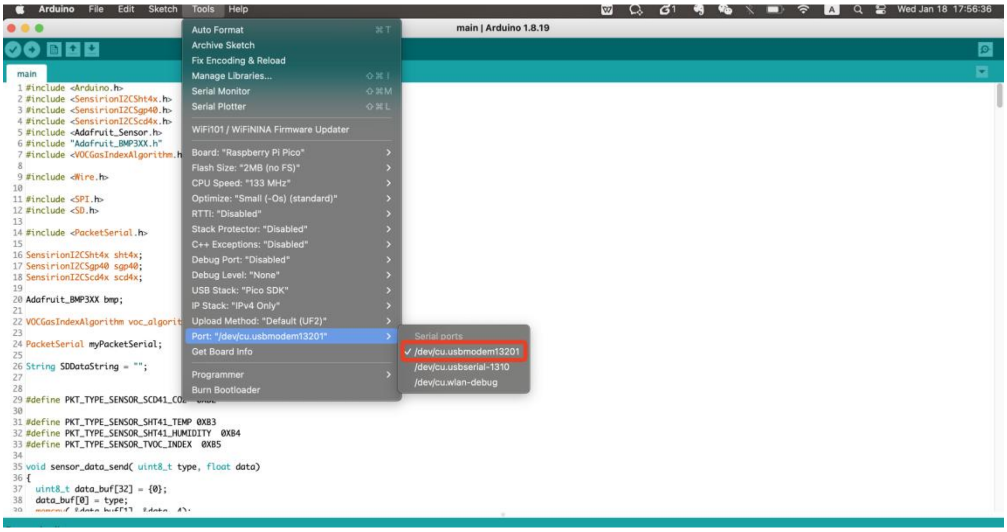
- **Step 5**: Select the board

Tools > **Board** > **Raspberry Pi PR2040 Boards** > **Raspberry Pi Pico**



```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
#define PKT_TYPE_SENSOR_TVOC_INDEX  0XB5

void sensor_data_send( uint8_t type, float data)
{
  uint8_t data_buf[32] = {0};
  data_buf[0] = type;
```

Port: Select the "usbmodem" one



Board: "Raspberry Pi Pico"
Flash Size: "2MB (no FS)"
CPU Speed: "133 MHz"
Optimize: "Small (-Os) (standard)"
RTTI: "Disabled"
Stack Protector: "Disabled"
C++ Exceptions: "Disabled"
Debug Port: "Disabled"
Debug Level: "None"
USB Stack: "Pico SDK"
IP Stack: "IPv4 Only"
Upload Method: "Default (UF2)"
Port: "/dev/cu.usbmodem13201"

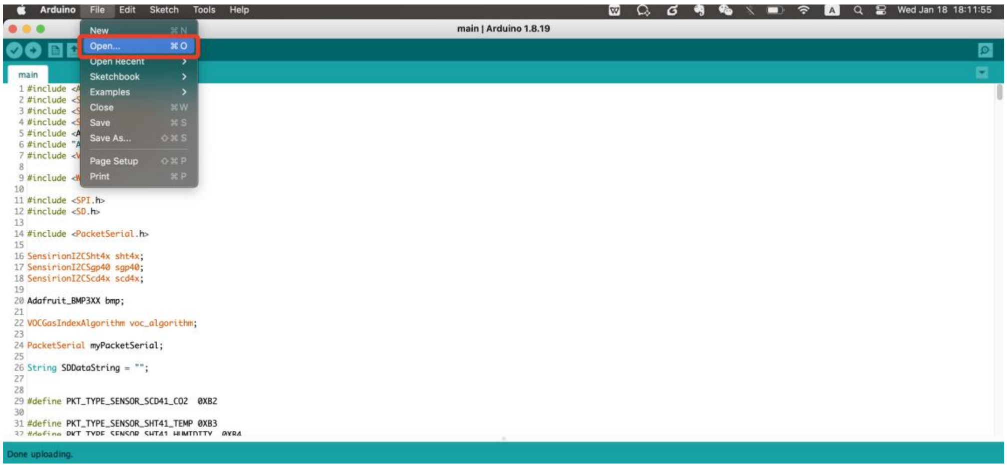
✓ /dev/cu.usbmodem13201
/dev/cu.usbserial-1310
/dev/cu.wlan-debug

```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorit

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
#define PKT_TYPE_SENSOR_TVOC_INDEX  0XB5

void sensor_data_send( uint8_t type, float data)
{
  uint8_t data_buf[32] = {0};
  data_buf[0] = type;
```

- **Step 6**: Open the example code file

File > **Open**, then select the example code file (.ino file).

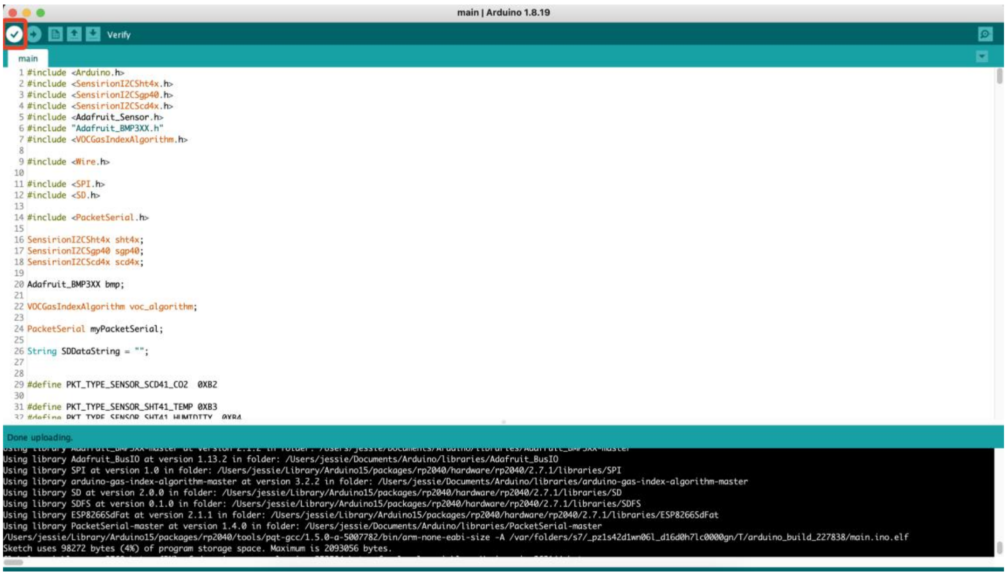
We provide an example code file, you can modify the code according to your needs.



```
#include <A
#include <S
#include <S
#include <S
#include <A
#include "A
#include <V

#include <W

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
```

Done uploading.

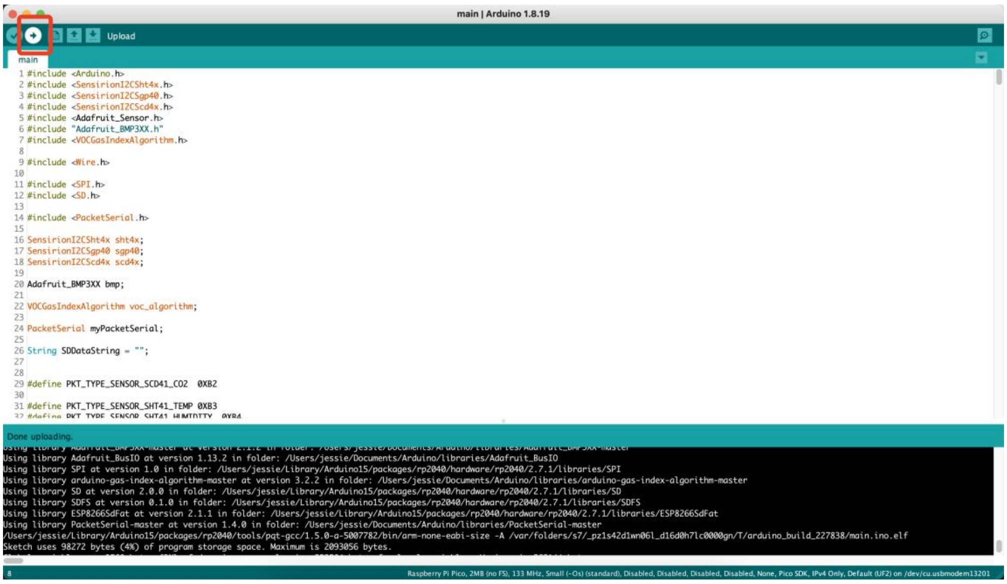
- **Step 7**: Verify and Upload the file.



```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h>

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
```

Done uploading.

```
Using library Adafruit_BMP3XX-master at version 2.1.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BMP3XX-master
Using library Adafruit_BusIO at version 1.13.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BusIO
Using library SPI at version 1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SPI
Using library arduino-gas-index-algorithm-master at version 3.2.2 in folder: /Users/jessie/Documents/Arduino/libraries/arduino-gas-index-algorithm-master
Using library SD at version 2.0.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SD
Using library SDFS at version 0.1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SDFS
Using library ESP8266SdFat at version 2.1.1 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/ESP8266SdFat
Using library PacketSerial-master at version 1.4.0 in folder: /Users/jessie/Documents/Arduino/libraries/PacketSerial-master
/Users/jessie/Library/Arduino15/packages/rp2040/tools/pqt-gcc/1.5.0-a-5007782/bin/arm-none-eabi-size -A /var/folders/s7/_pz1s42d1wn06l_d16d0h7lc0000gn/T/arduino_build_227838/main.ino.elf
Sketch uses 98272 bytes (4%) of program storage space. Maximum is 2093056 bytes.
```

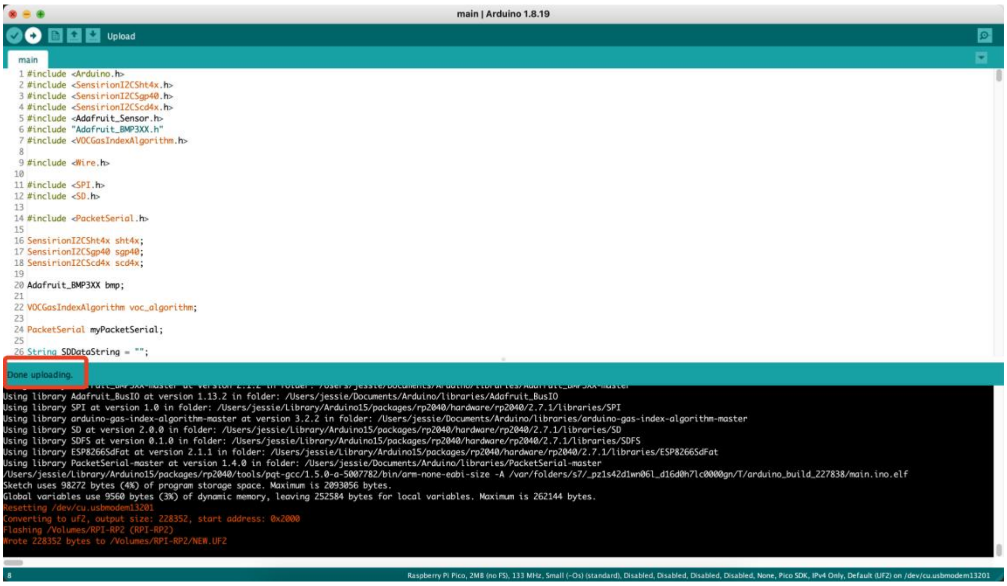


```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h>

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";


#define PKT_TYPE_SENSOR_SCD41_CO2  0XB2

#define PKT_TYPE_SENSOR_SHT41_TEMP 0XB3
#define PKT_TYPE_SENSOR_SHT41_HUMIDITY  0XB4
```

Done uploading.

```
Using library Adafruit_BMP3XX-master at version 2.1.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BMP3XX-master
Using library Adafruit_BusIO at version 1.13.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BusIO
Using library SPI at version 1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SPI
Using library arduino-gas-index-algorithm-master at version 3.2.2 in folder: /Users/jessie/Documents/Arduino/libraries/arduino-gas-index-algorithm-master
Using library SD at version 2.0.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SD
Using library SDFS at version 0.1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SDFS
Using library ESP8266SdFat at version 2.1.1 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/ESP8266SdFat
Using library PacketSerial-master at version 1.4.0 in folder: /Users/jessie/Documents/Arduino/libraries/PacketSerial-master
/Users/jessie/Library/Arduino15/packages/rp2040/tools/pqt-gcc/1.5.0-a-5007782/bin/arm-none-eabi-size -A /var/folders/s7/_pz1s42d1wn06l_d16d0h7lc0000gn/T/arduino_build_227838/main.ino.elf
Sketch uses 98272 bytes (4%) of program storage space. Maximum is 2093056 bytes.
```

8 Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, Disabled, Disabled, None, Pico SDK, IPv4 Only, Default (UF2) on /dev/cu.usbmodem13201



```
#include <Arduino.h>
#include <SensirionI2CSht4x.h>
#include <SensirionI2CSgp40.h>
#include <SensirionI2CScd4x.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BMP3XX.h"
#include <VOCGasIndexAlgorithm.h>

#include <Wire.h>

#include <SPI.h>
#include <SD.h>

#include <PacketSerial.h>

SensirionI2CSht4x sht4x;
SensirionI2CSgp40 sgp40;
SensirionI2CScd4x scd4x;

Adafruit_BMP3XX bmp;

VOCGasIndexAlgorithm voc_algorithm;

PacketSerial myPacketSerial;

String SDDataString = "";
```

Done uploading.

```
Using library Adafruit_BMP3XX-master at version 2.1.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BMP3XX-master
Using library Adafruit_BusIO at version 1.13.2 in folder: /Users/jessie/Documents/Arduino/libraries/Adafruit_BusIO
Using library SPI at version 1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SPI
Using library arduino-gas-index-algorithm-master at version 3.2.2 in folder: /Users/jessie/Documents/Arduino/libraries/arduino-gas-index-algorithm-master
Using library SD at version 2.0.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SD
Using library SDFS at version 0.1.0 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/SDFS
Using library ESP8266SdFat at version 2.1.1 in folder: /Users/jessie/Library/Arduino15/packages/rp2040/hardware/rp2040/2.7.1/libraries/ESP8266SdFat
Using library PacketSerial-master at version 1.4.0 in folder: /Users/jessie/Documents/Arduino/libraries/PacketSerial-master
/Users/jessie/Library/Arduino15/packages/rp2040/tools/pqt-gcc/1.5.0-a-5007782/bin/arm-none-eabi-size -A /var/folders/s7/_pz1s42d1wn06l_d16d0h7lc0000gn/T/arduino_build_227838/main.ino.elf
Sketch uses 98272 bytes (4%) of program storage space. Maximum is 2093056 bytes.
Global variables use 9560 bytes (3%) of dynamic memory, leaving 252584 bytes for local variables. Maximum is 262144 bytes.
Resetting /dev/cu.usbmodem13201
Converting to uf2, output size: 228352, start address: 0x2000
Flashing /Volumes/RPI-RP2 (RPI-RP2)
Wrote 228352 bytes to /Volumes/RPI-RP2/NEW.UF2
```

8 Raspberry Pi Pico, 2MB (no FS), 133 MHz, Small (-Os) (standard), Disabled, Disabled, Disabled, Disabled, None, Pico SDK, IPv4 Only, Default (UF2) on /dev/cu.usbmodem13201

Resource

IDF Download: Standard Toolchain Setup for Linux and macOS

Standard Setup of Toolchain for Windows

SenseCAP Indicator ESP32 SDK：SenseCAP Indicator SDK for esp32

SenseCAP Indicator RP2040 Examples: SenseCAP Indicator examples for rp2040