

Common to all RT products

Arduino development environment construction manual

Version 1.1 RT Corporation

table of contents

Table	1
of Contents Before	2
UseComputer Requirements	2
of	3
Environment Setting Commands	3
Introduction	3
Installing Arduino IDE	3
For Windows	6
For Linux (Ubuntu) For macOS	8
	10
Arduino IDE initial settings	11
Language	11
settings	13
Appearance settings	14
Build and write sample sketches	14
Pi:Co Classic3 and ESP32-S3 microcontroller boards	14
Add ESP32 board information	14
Prepare sample sketch Build Write	17
	22
	23
Pi:Co V2	26
Add ESP32 board information	26
Prepare sample sketch Build Write	29
Training	33
tracer	34
	37
Add STM32 board information	37
Install the programming tool STM32CubeProgrammer Prepare a sample sketch	40
Build Write How to deal with errors	51
during	55
programming	56
	58
ST-Link Firmware Update	59
Tools used, OSS version revision history	67
	68
Copyright and Intellectual Property Rights	68

Before Use

Thank you for purchasing our product (hereinafter referred to as "this product"). Before using this product, please read the introductory guide that comes with the product.

Computer Requirements

When using this product, please prepare a PC that meets the operating requirements of Arduino IDE.
For the Arduino [IDE operating requirements, please refer to https://www.arduino.cc/en/software](https://www.arduino.cc/en/software).

Environment settings

This section explains the procedure for building the development environment for this product.

Command notation

In this manual, commands to be executed in the Linux terminal are written as follows. The string following "\$" represents the command (excluding "\$"). Lines beginning with "#" represent comments and do not need to be executed.

```
$ echo "command" # this  
line is a comment
```

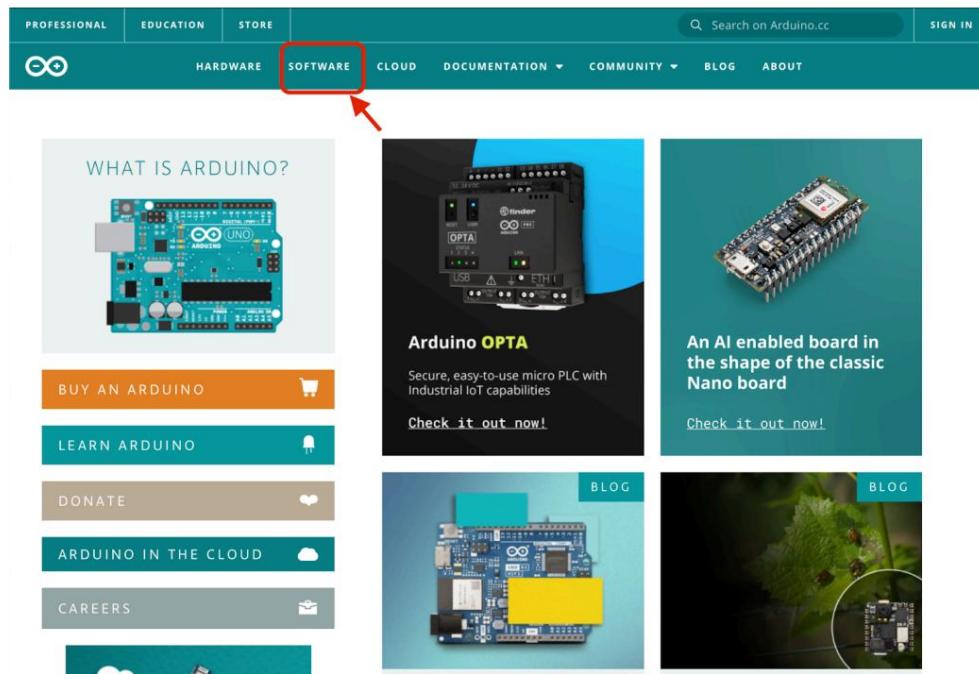
Introduction

This product uses the Arduino IDE provided by the Arduino Foundation as the development environment. The Arduino IDE supports multiple platforms, so this product can be developed on three operating systems: Windows, Linux, and macOS.

Installing Arduino IDE

This is the procedure for installing the latest version of Arduino IDE 2.3.2 as of June 2024. If your IDE is updated, please update the version numbers in this document accordingly.

Access <https://www.arduino.cc> in your web browser and click on the SOFTWARE tab.



<https://www.arduino.cc> Top page

Click the software that matches your OS from the DOWNLOAD OPTIONS.

This article explains the installation procedure assuming you have selected Windows MSI installer, Linux AppImage 64 bits (X86-64), and macOS Intel, 10.15: "Catalina" or newer, 64 bits.

The screenshot shows the Arduino IDE selection page. On the left, there's a logo and the text "Arduino IDE 2.3.2". Below it, a paragraph describes the new features of the IDE. A link leads to the "Arduino IDE 2.0 documentation". Another section links to the "SOURCE CODE" on GitHub. On the right, a teal sidebar titled "DOWNLOAD OPTIONS" lists download links for Windows (MSI installer, ZIP file), Linux (AppImage, ZIP file), and macOS (Intel, Apple Silicon). A link to "Release Notes" is also present.

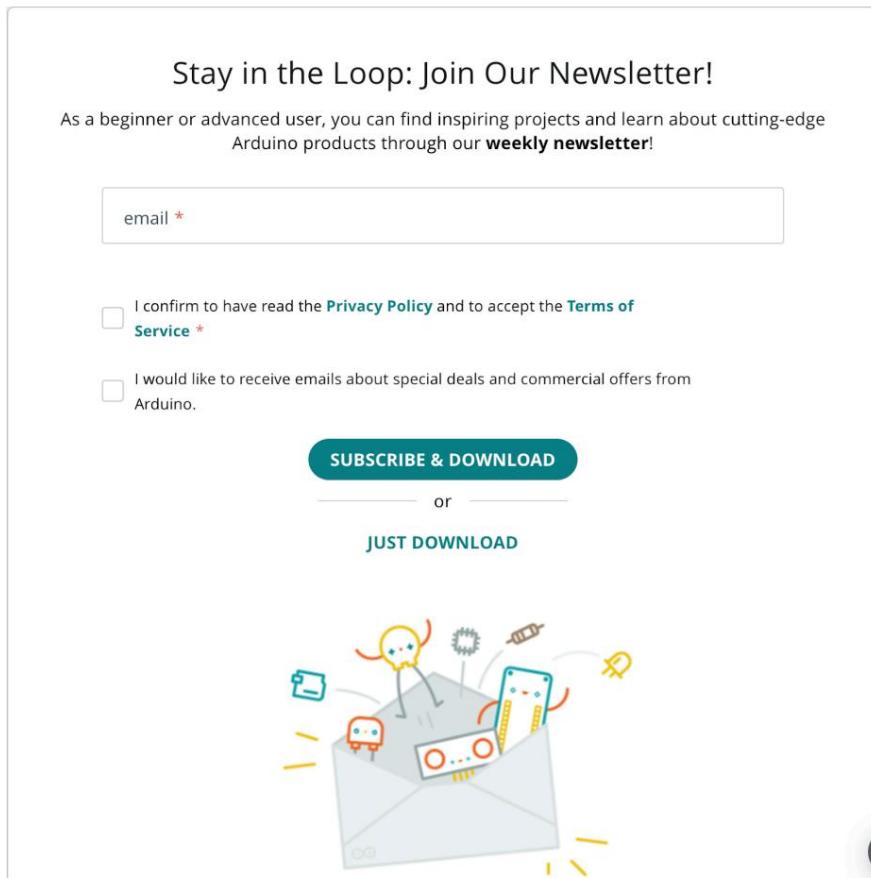
Arduino IDE selection page

If the Download Arduino IDE & support progress page below appears, **select either "CONTRIBUTE & DOWNLOAD" to donate and download the software, or "JUST DOWNLOAD" to just download the software.**

The screenshot shows the "Download Arduino IDE & support its progress" page. It features a counter for 77,959,898 downloads since March 2015. Below the counter are donation buttons for \$3, \$5, \$10, \$25, \$50, and "Other". Two large buttons at the bottom are "CONTRIBUTE AND DOWNLOAD" and "JUST DOWNLOAD". Between them is the text "or". Below the buttons is a cartoon illustration of a character standing next to a large jar labeled "Arduino" which is filled with coins. A small text link at the bottom says "Learn more about donating to Arduino."

Arduino IDE download method selection page

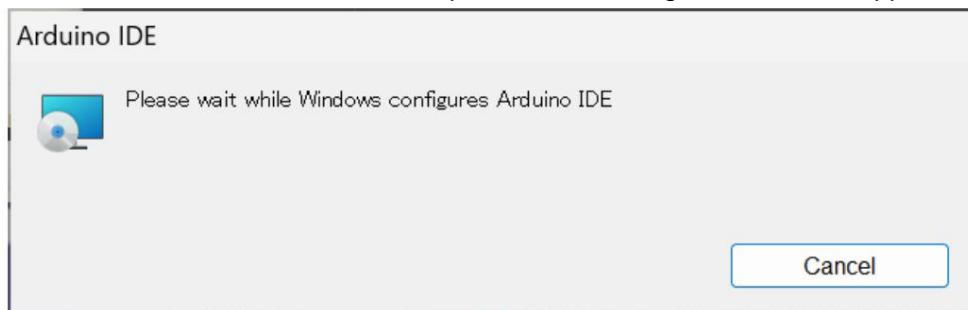
When the Stay in the loop: Join Our Newsletter page appears, if you would like to subscribe, enter your email address and check both boxes, then click SUBSCRIBE & DOWNLOAD. If you do not wish to subscribe, click JUST DOWNLOAD.



Newsletter subscription confirmation page

For Windows

Double-click the downloaded MSI installer (e.g. arduino-ide_2.3.2_Windows_64bit_msi). Click and install. When the installation is complete, the following screen will disappear.



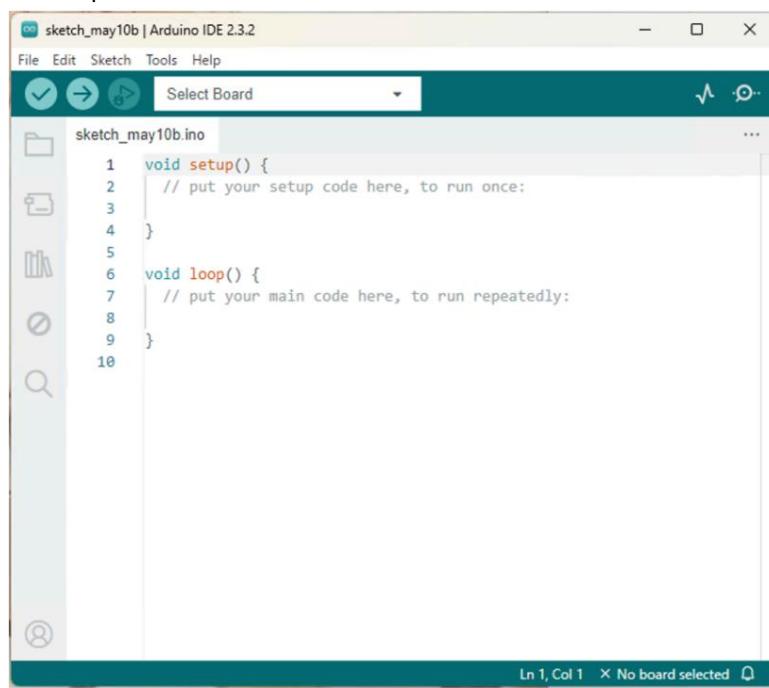
Installation screen

Start the Arduino IDE to check that it was installed correctly.
Click the Windows Start button -> enter the keyword “arduino IDE” in the search box and Arduino IDE will appear as a search result.



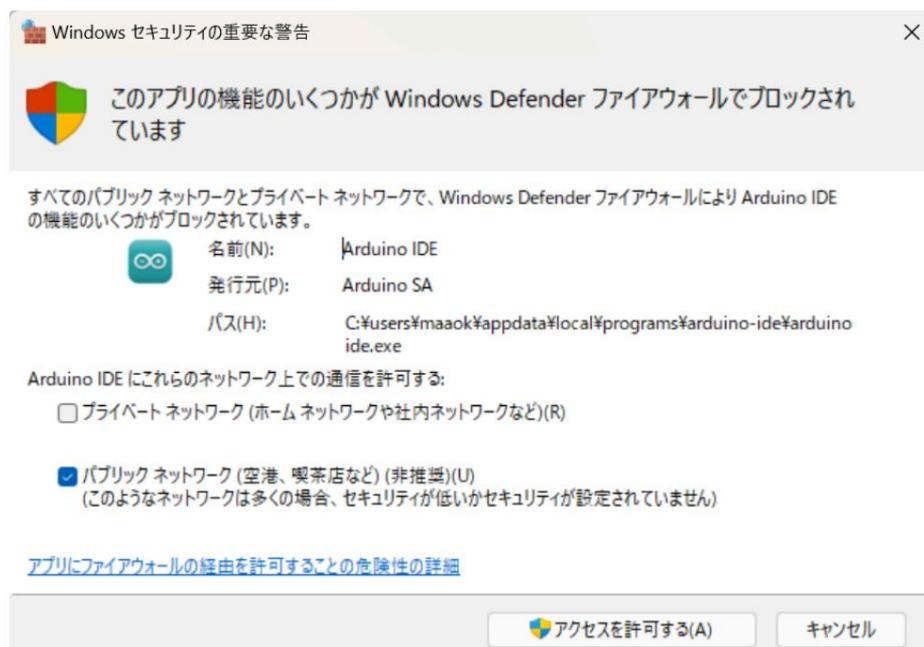
Entering "arduino IDE" in the search box

Click on the displayed Arduino IDE, and when the Arduino IDE starts, the following screen will be displayed.
The installation is now complete.



Launched Arduino IDE

If a Windows security warning message appears, allow access. If you block access, you will not be able to update the Arduino IDE or download additional libraries.



Windows Security Critical Warning Message

For Linux (Ubuntu), right-click

the downloaded file (e.g. arduino-ide_2.3.2_Linux_64bit.AppImage) and select Properties from the menu. Open the Access Rights tab in Properties and check "Can be executed as a program."

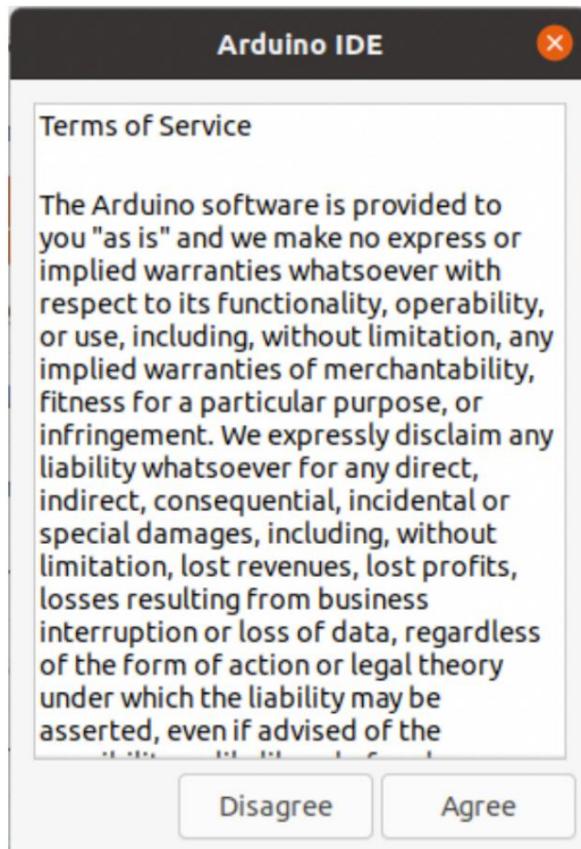


Right-click on the AppImage file



Properties Access Rights tab open

Then double-click the Appliance file. If the Terms of Service appears, Please check the contents and select "Disagree/Agree". Installation is now complete.

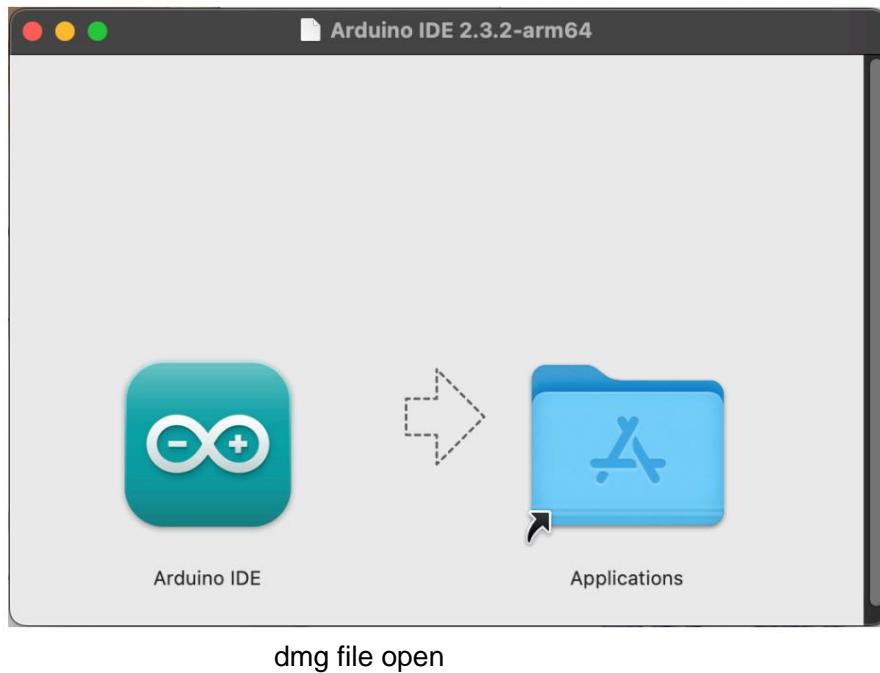


Terms of Service screen

If the Arduino IDE does not run when you double-click the Appliance file, You may not have FUSE, the tool that extracts and runs the contents of an Appliance file, installed . <https://github.com/ApplianceKit/FUSE> Refer to Install FUSE.

For macOS

Double-click the downloaded file (e.g. arduino-ide_2.3.2_macOS_64bit.dmg) and the following screen will be displayed. Move the Arduino IDE icon on the left side of the screen to the folder on the right to complete the installation. After installation, Arduino IDE can be launched from Launchpad.



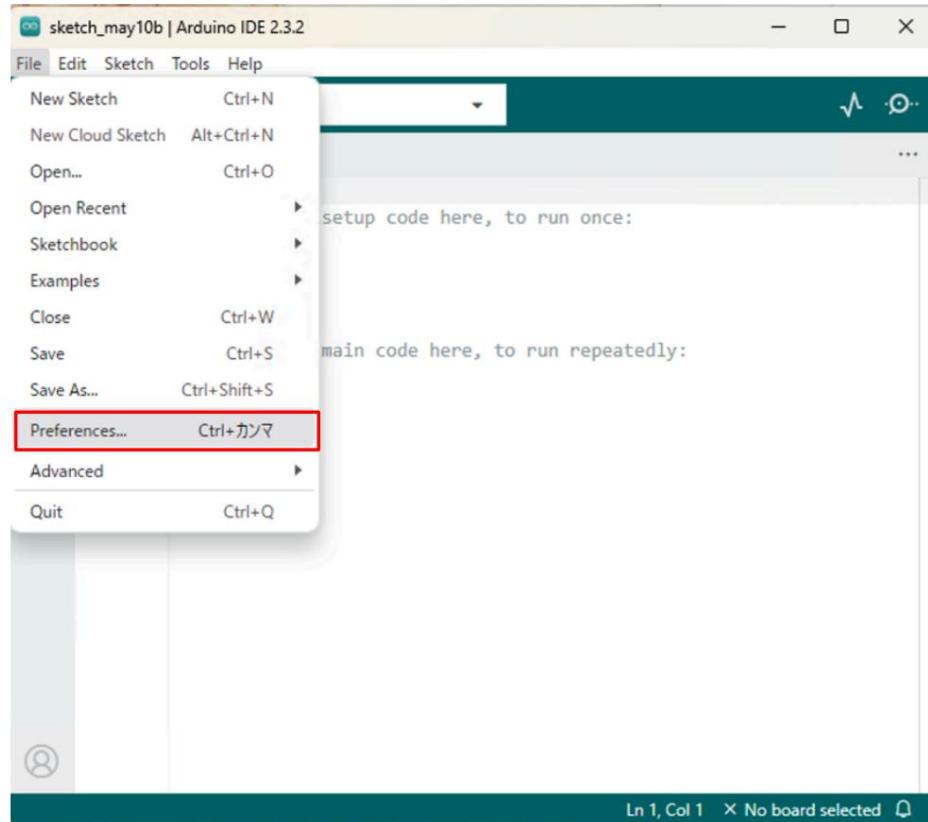
dmg file open

Initial Arduino IDE setup

Here you can set the language and appearance of the Arduino IDE. You can also add libraries and compilers to build sketch files for the ESP32-S3 included with this product.

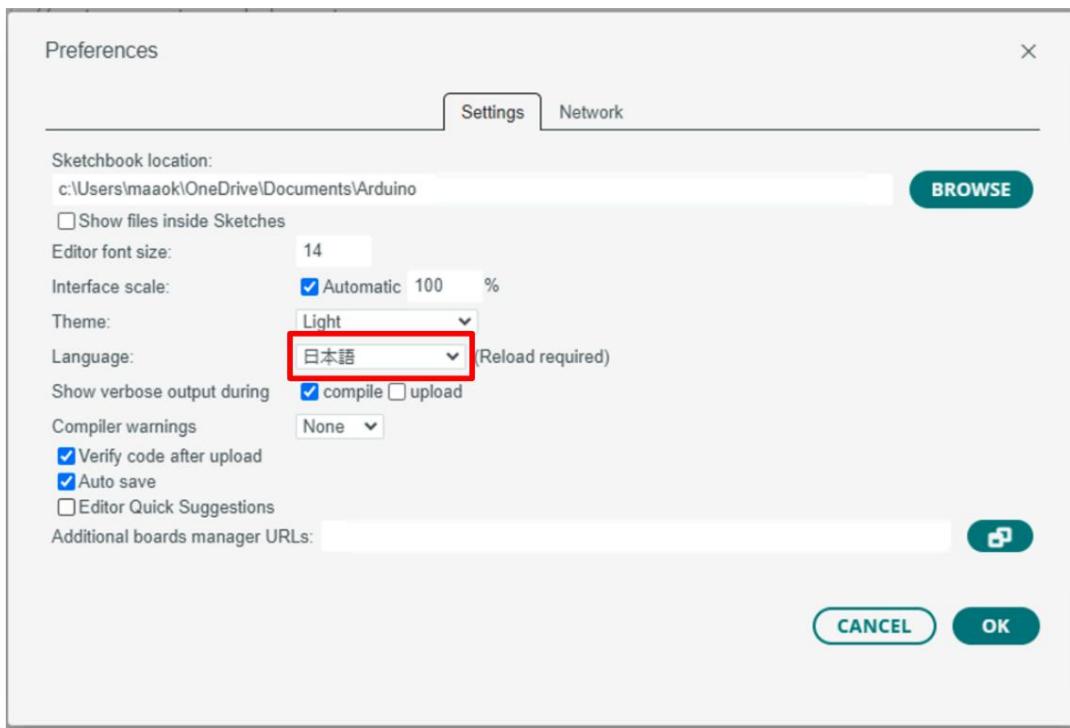
Language settings

Set the Arduino IDE language to Japanese. Select File->Preferences... on the menu bar.



When File->Preferences... is selected on the menu bar

Change the Language from "English" to "Japanese" and click "OK" at the bottom right.



Language set to Japanese

After clicking "OK", the Arduino IDE will automatically restart. After restarting, click If the button on the far left changes from "File" to "File", the language setting is complete.

Appearance Settings

Depending on your OS settings, the appearance of the Arduino IDE may be displayed in dark tones. If your PC screen has low contrast, you may be able to see the cursor on the sketch editing screen more clearly by setting it to a lighter color tone. Follow the steps below to set the appearance to your liking. Select File ->

Preferences... from the menu bar. Change "Color Theme" from "Dark" to "Light".

Change the settings and click "OK". This will change the appearance of the Arduino IDE to a lighter color scheme.

The Arduino IDE screen that appears on the following pages has the "Light" theme.



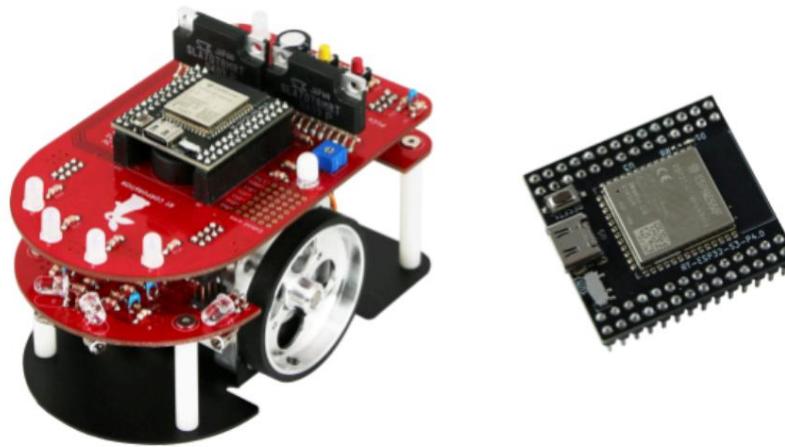
Color theme with Light selected

Build and flash the example sketch

Here you can add a board for each product, download a sample sketch, build it, and write it.

The procedure is as follows:

Pi:Co Classic3 and ESP32-S3 microcontroller boards

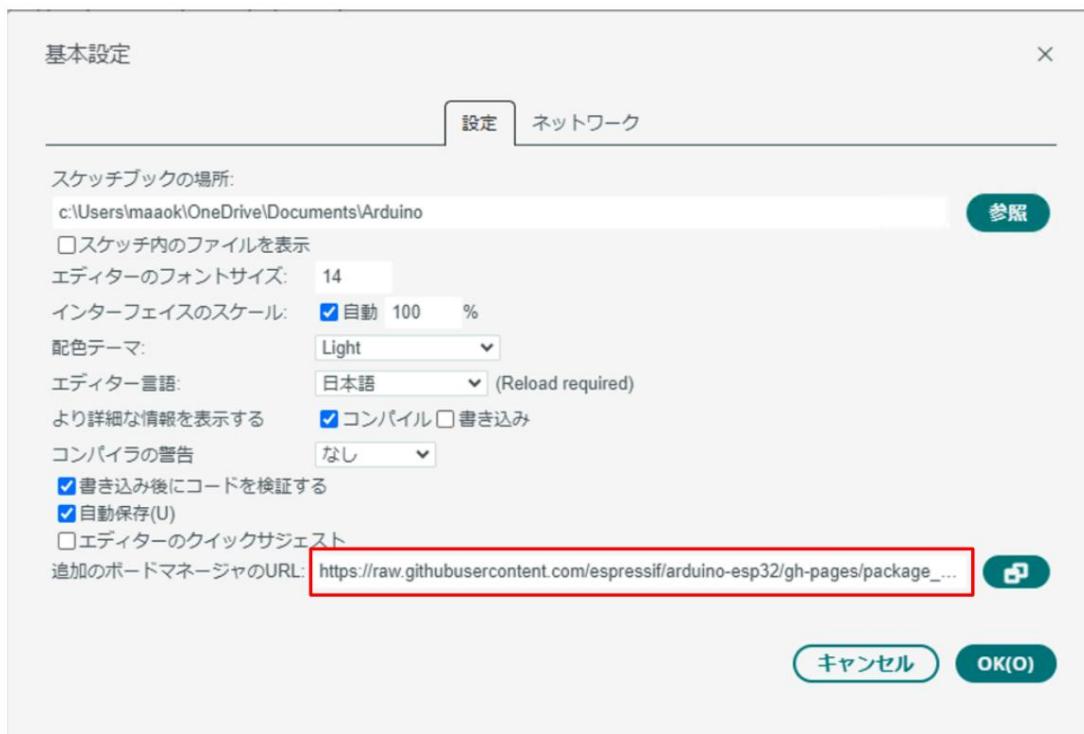


Added ESP32 board information

Add the ESP32 board information to the Arduino IDE so that you can write sketches to this product. Select File -> Preferences... from the menu bar. Enter the "Additional Boards Manager URL" at the bottom of the screen.

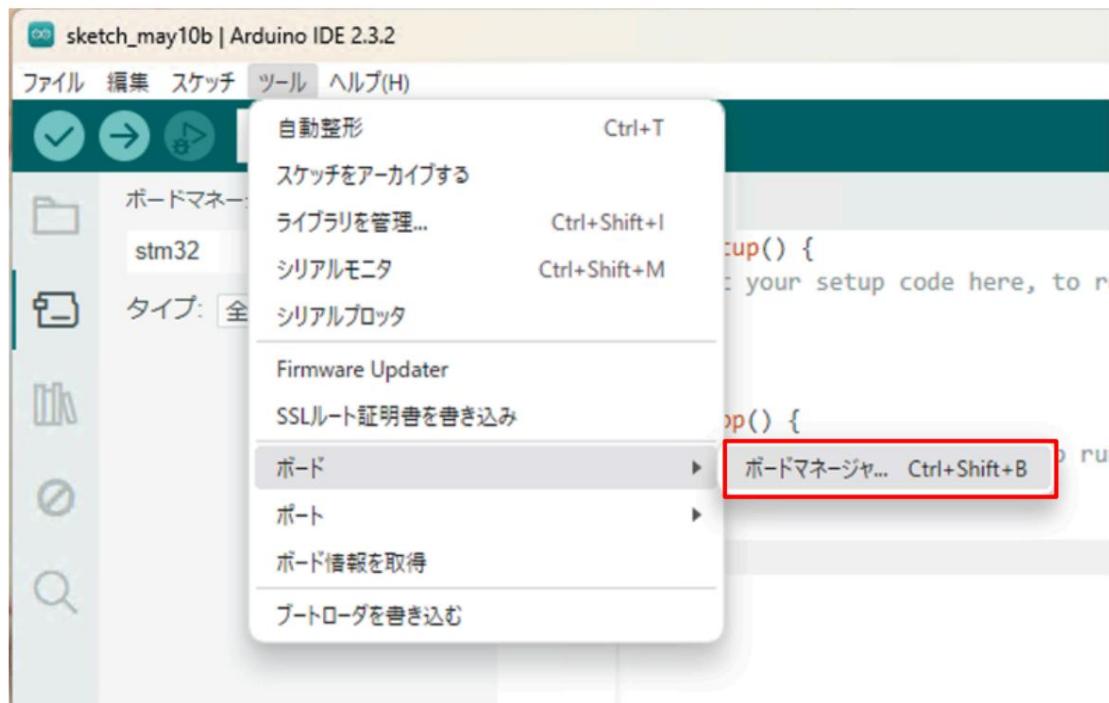
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

Add it and click "OK".



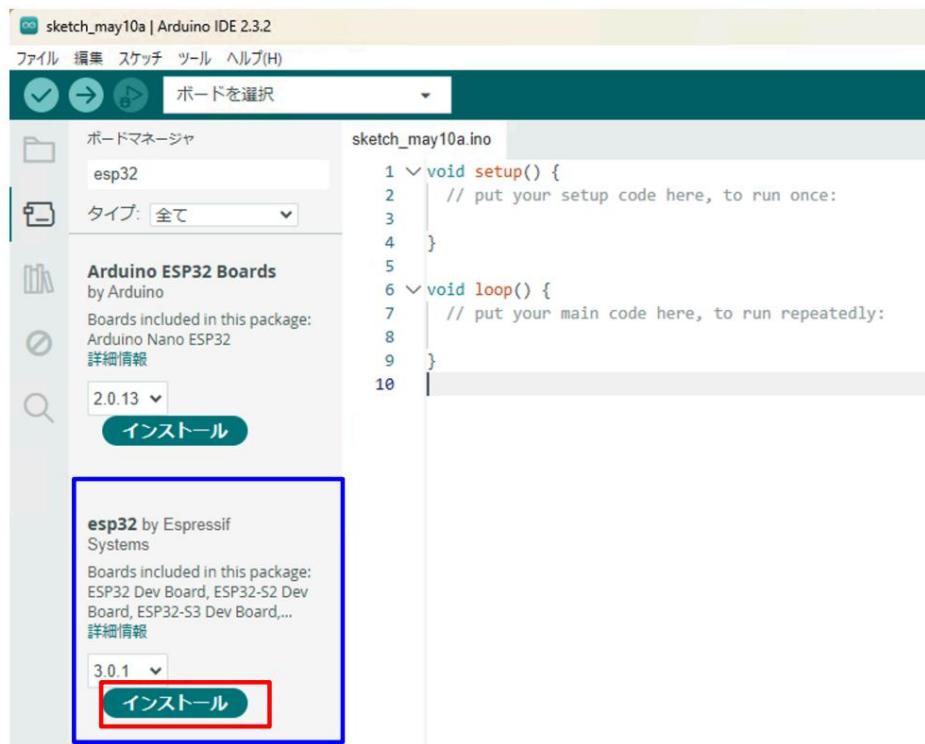
Added to the URL of the additional board manager

Once you have added the URL, download and install the ESP32 libraries and tools: from the menu bar, select Tools -> Board -> Boards Manager...



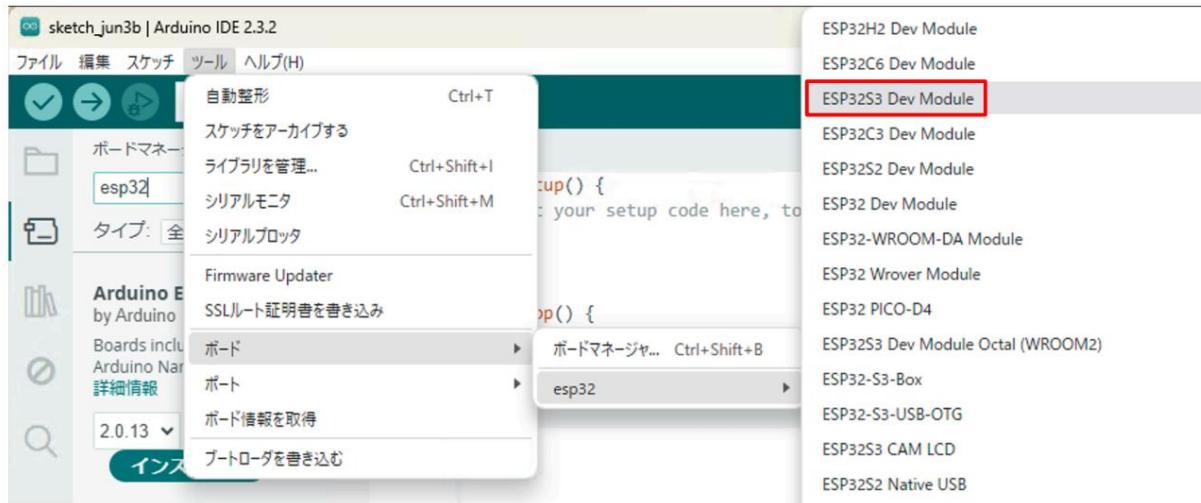
Boards Manager... selected

Enter "esp32" in the search field of the Board Manager on the left and "esp32 by Espressif Systems" will be displayed. As of June 2024, 3.0.1 is the latest version. Click Install esp32 by Espressif System.



Board Manager screen with "esp32" entered in the search field

Since the Arduino IDE supports microcontrollers other than the ESP32, select the ESP32 board when writing a sketch. Select Tools -> Board -> esp32 -> ESP32S3 Dev Module from the menu bar. The "Board" and "Port" are written in similar letters, so be careful not to select the wrong one.



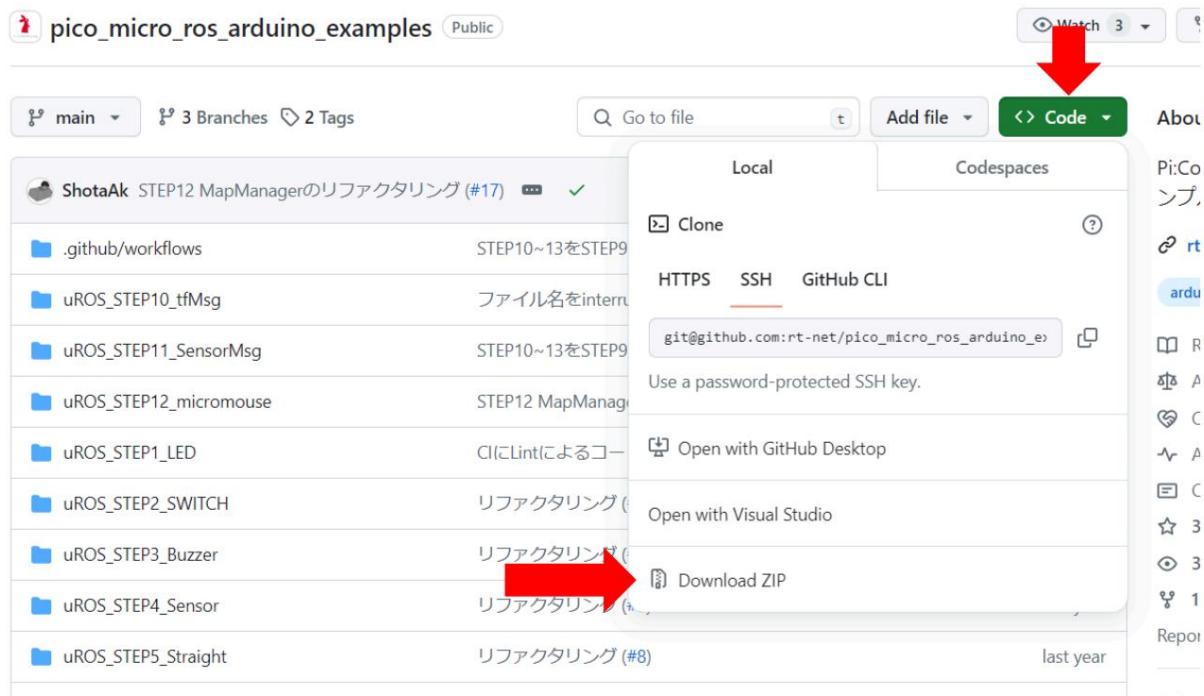
Select Tools -> Board -> esp32 -> ESP32S3 Dev Modules

This completes the build environment setup for ESP32-S3.

Prepare the sample sketch. Access

https://github.com/rt-net/pico_micro_ros_arduino_examples in a web browser.

Download the sample sketch file for this product. You can download it by clicking "Download ZIP" from "Code" on the page. Extract the downloaded zip file.

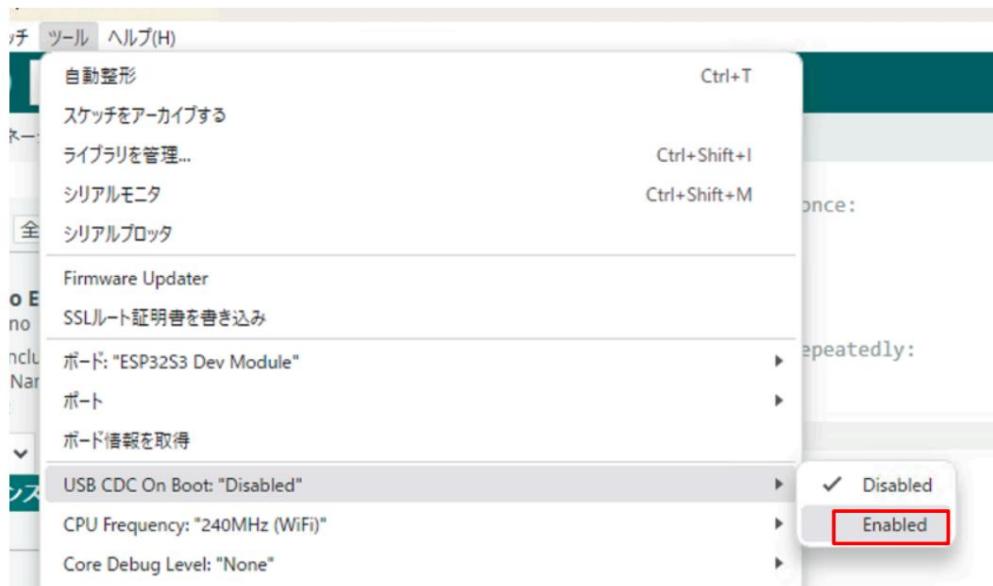


Select Code->Download ZIP to download the sample sketch.

Here is a sample sketch (uROS_STEP1_LED) that lights up the LED on the Pi Co Classic3. We will use an example to explain how to build and write a sketch.

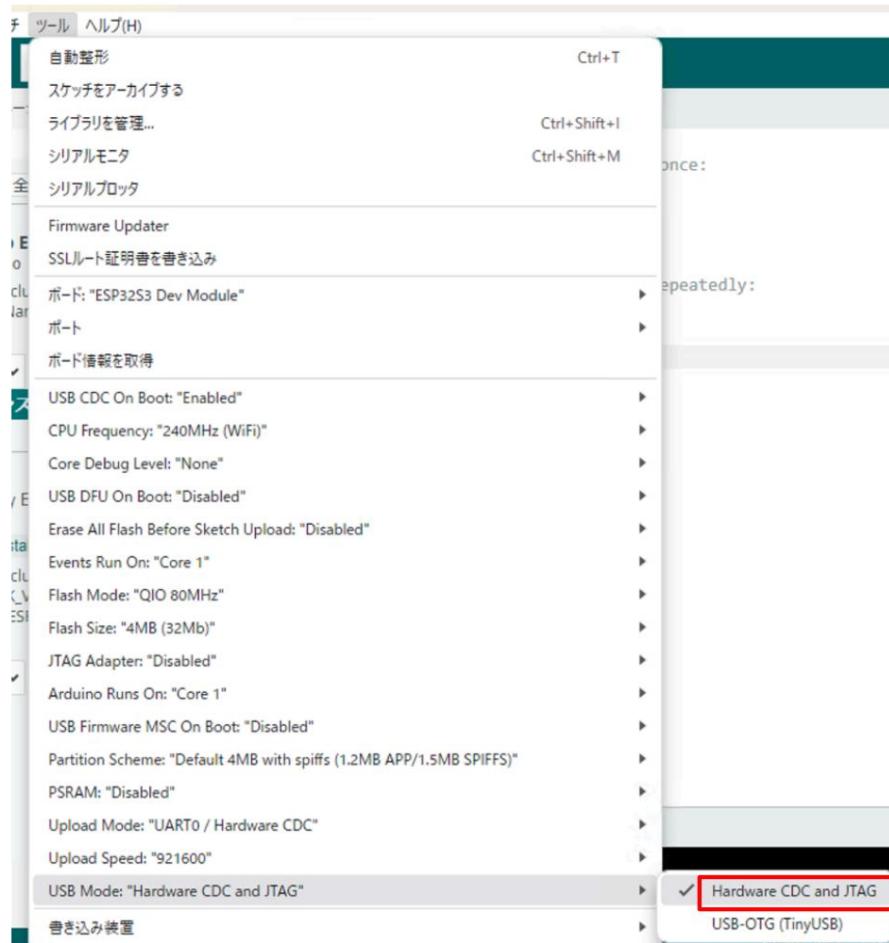
Write the sketch to this product via USB. By default, writing from USB is not possible. Therefore, you will need to make the following two changes to enable writing.

First, select Tools -> Board -> esp32 -> ESP32S3 Dev Module from the menu bar. Next
Next, open Tools -> USB CDC On Boot from the menu bar and set it to "Enabled".



Set USB CDC On Boot to Enabled

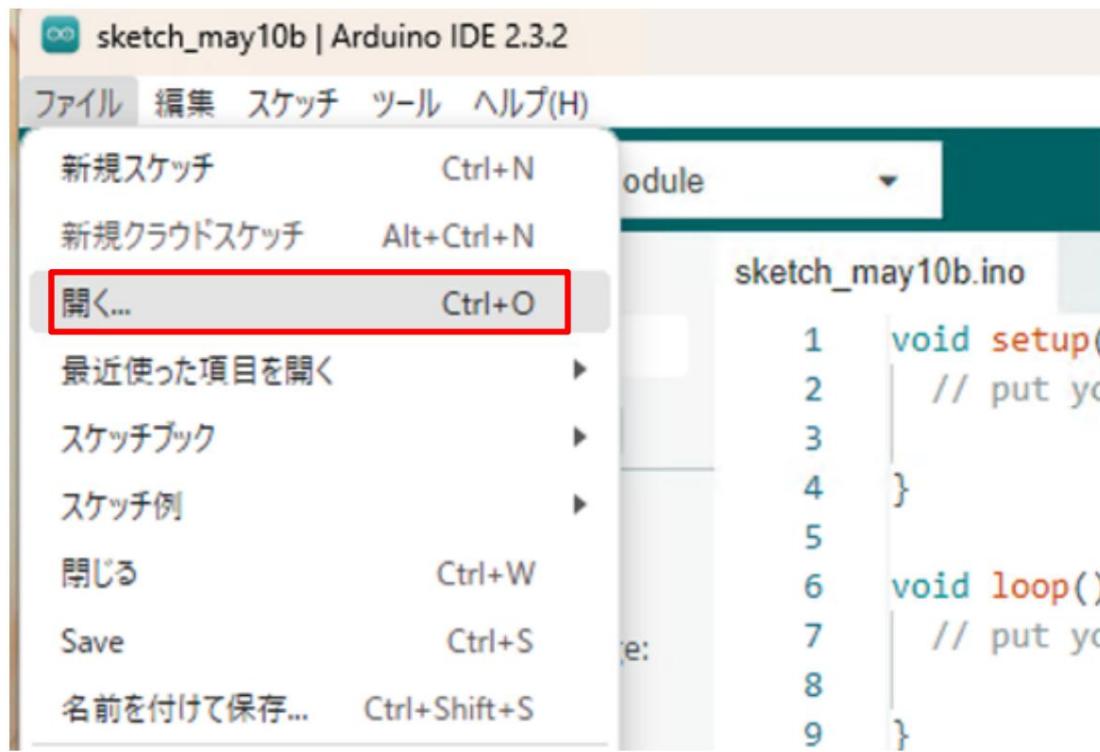
Open Tools -> USB Mode on the menu bar and select "Hardware CDC and JTAG".



Set USB Mode to Hardware CDC and JTAG

Double-click the uROS_STEP1_LED/uROS_STEP1_LED.ino file in the downloaded sample sketch to open it.

Alternatively, start the Arduino IDE and click File -> Open... on the menu bar to open uROS_STEP1_LED.ino.



Select File -> Open...

If the following message appears when opening a sketch file, it means that the sketch file name and folder name do not match, or the folder does not exist. If you want to create a new folder, click OK.

Due to the specifications of the Arduino IDE, this message appears when the main "file name" and "folder name" are not the same. If you change the file name or folder name, make sure to give them the same name.



Confirmation screen when file name and folder name do not match

When you open the sample sketch, the following screen will be displayed.

The screenshot shows the Arduino IDE 2.0.3 interface with the title bar "uROS_STEP1_LED | Arduino IDE 2.0.3". The central area displays the code for "uROS_STEP1_LED.ino". The code defines four pins (LED0, LED1, LED2, LED3) as outputs and sets up a loop where each pin alternates between HIGH and LOW every 500ms. The output window at the bottom is currently blank.

```
// Copyright 2022 RT Corporation
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

#define LED0 1
#define LED1 2
#define LED2 42
#define LED3 41

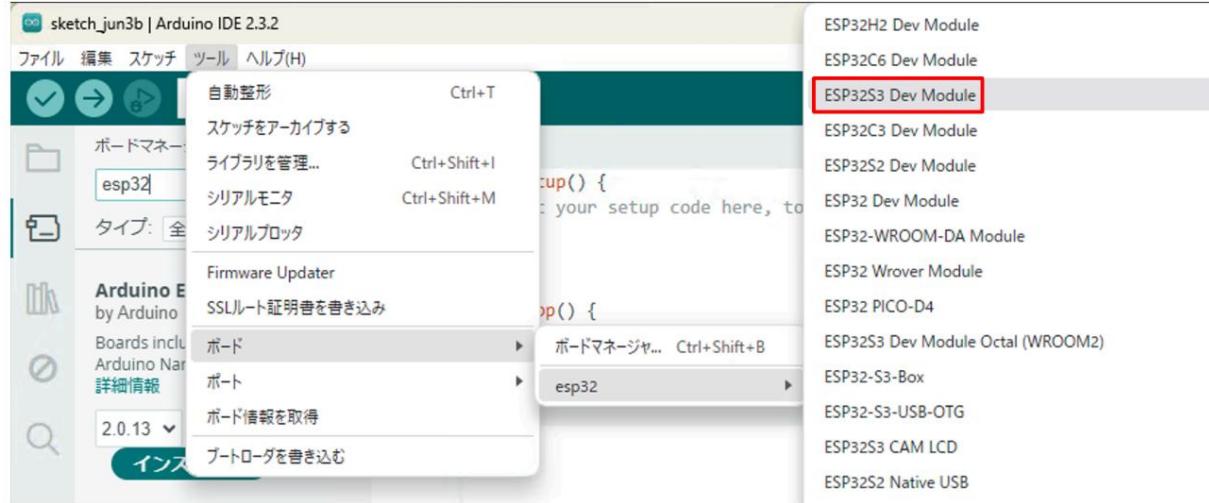
void setup() {
    // put your setup code here, to run once:
    pinMode(LED0,OUTPUT);
    pinMode(LED1,OUTPUT);
    pinMode(LED2,OUTPUT);
    pinMode(LED3,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(LED0,HIGH);
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,HIGH);
    digitalWrite(LED3,HIGH);
    delay(500);
    digitalWrite(LED0,LOW);
    digitalWrite(LED1,LOW);
    digitalWrite(LED2,LOW);
    digitalWrite(LED3,LOW);
    delay(500);
}
```

uROS_STEP1_LED.ino open

If the board is not set to “ESP32S3 Dev Module”, go to Tools -> Board in the menu bar.

Select ->esp32->ESP32S3 Dev Module.



Select Tools->Board->esp32->ESP32S3 Dev Module

Build



Click the icon in the upper left corner to run "Verify" and build the sketch.

If there are no problems, the following message will be displayed:

```
出力
最大1310720バイトのフラッシュメモリのうち、スケッチが248565バイト（18%）を使っています。
最大327680バイトのRAMのうち、グローバル変数が18456バイト（5%）を使っていて、ローカル変数で309224バイト使うことができます。
```

Message when there is no problem with the validation

If there is an error in the sketch, the error will be displayed in red.

```
出力
pinMode(LED4,OUTPUT);
          | | | | | ~~~
/Users/aokimasatake/Documents/Arduino/uROS_STEP1/uROS_STEP1.ino:12:11: note: suggested alternative:
pinMode(LED4,OUTPUT);
          | | | | | ~~~
                           LED0

exit status 1

Compilation error: 'LED4' was not declared in this scope
```

Validation error message



You can hide the message by clicking the icon at the bottom right of the Arduino IDE. Click it again to show it again. Use this function when you want to enlarge the screen to see more of the entire sketch, or when you do not need this message screen.

For Linux (Ubuntu), if the message "ModuleNotFoundError: No module named 'serial'" is displayed, the build has not been completed. Execute the following command in the terminal to install the apps required for the build.

```
$ sudo apt-get install python-pip $ pip install
pyserial $ pip install esptool
```

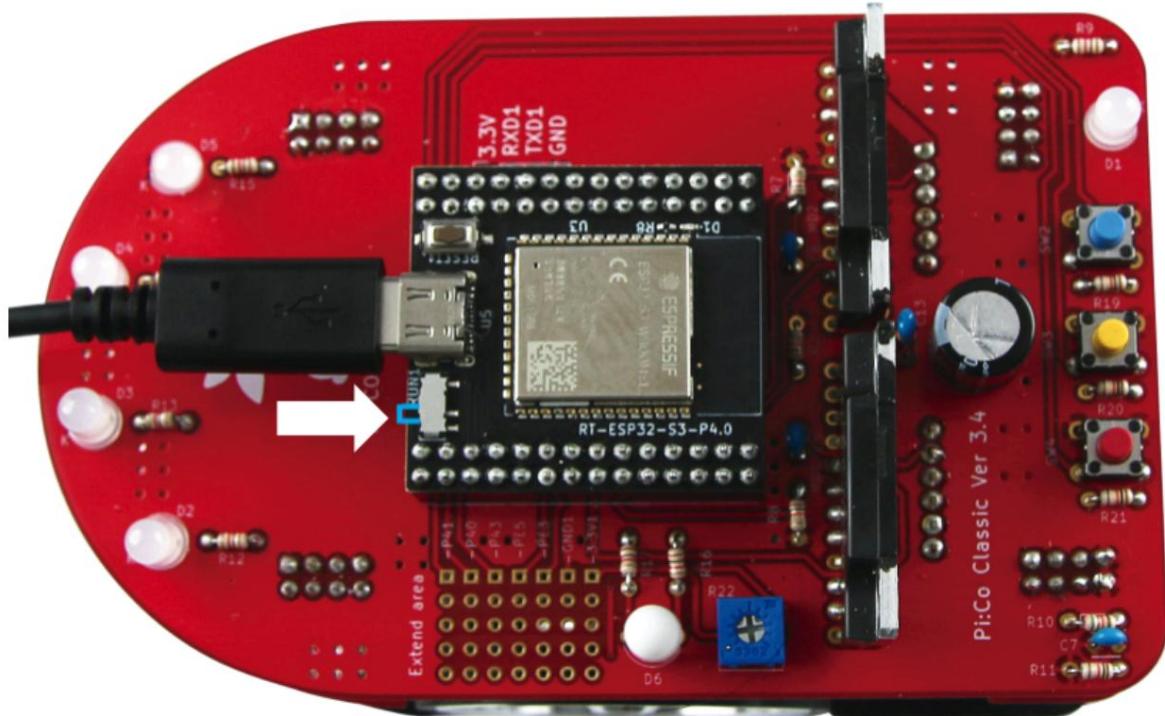


RT CORPORATION

write

Once you have confirmed that there are no errors in the sketch, prepare for writing using the following steps: 1.

1. Turn off the power switch of this product.
2. Connect the battery to this product.
3. Connect the writing terminal of this product to the PC with a USB cable.
4. Slide the mode switch of this product in the direction opposite to RUN1.
5. Turn on the power switch of this product.
6. Press the reset button of this product.

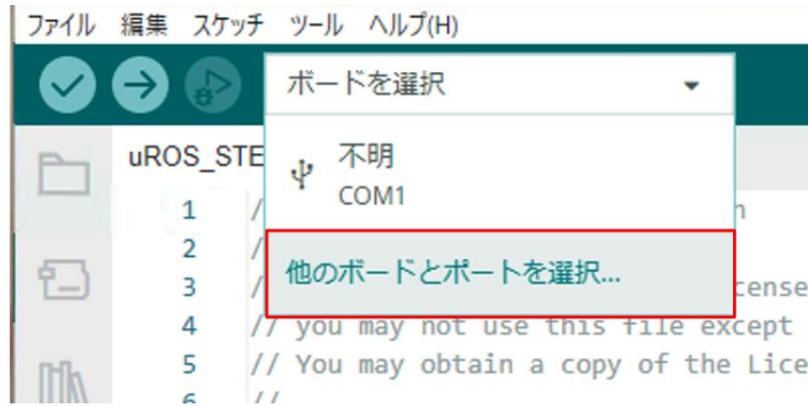


Mode switch operated and USB cable connected

After connecting this product to the PC, follow the steps below to set up the board and port.

Please set it again after turning it off or pressing the reset button.

Click the board name (ESP32S3 Dev Module) displayed under the menu bar and open "Select Other Board and Port..." at the bottom.



Open the board name -> Select other board and port...

Enter "dev" in the board search field and make sure "ESP32S3 Dev Module" is checked.

If it is not checked, click ESP32S3 Dev Module. In the Port Selection field, select the USB port to which this product is connected. For Windows, select COMX.

For Linux (Ubuntu), it is called /dev/ttyUSBX, and for macOS, it is called usbmodemXXXXX Serial Port(USB). Once you have selected it, click "OK".



Board and port settings

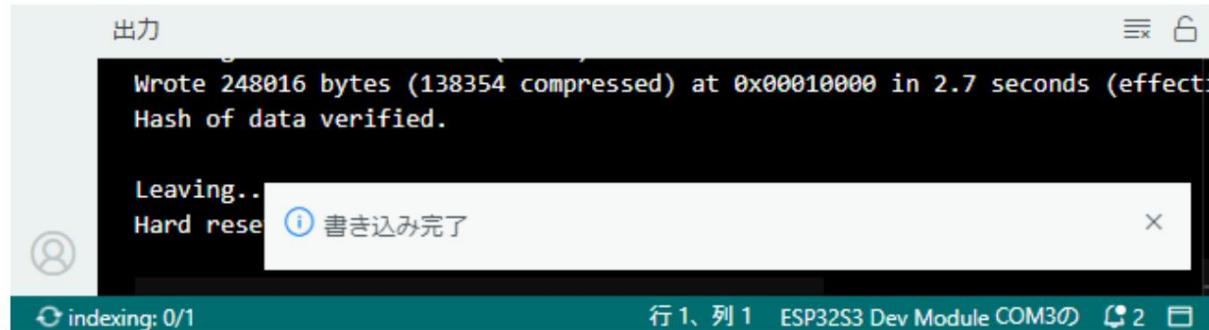


After completing the setup, a USB mark will be added to the left of the board name. In addition to writing sketches, port settings are also required when communicating with a PC via hardware serial communication.



After setting the port, click to write the sketch.

Writing is now complete.



Screen display when writing is complete

If writing fails on a Linux (Ubuntu) PC, run the following command in the terminal before writing to change the USB port permissions.

```
$ sudo chmod 777 /dev/ttyACM0
```

To change the USB port usage permissions permanently, run the following command and restart your PC.

```
# Restart the PC after executing the following: $  
sudo usermod -aG dialout $USER
```

After writing the sketch, slide the mode switch on this product to the RUN1 side and press the reset button.

The LED on the front of this product will flash for 0.5 seconds and then turn off for 0.5 seconds repeatedly.

Pi:Co V2

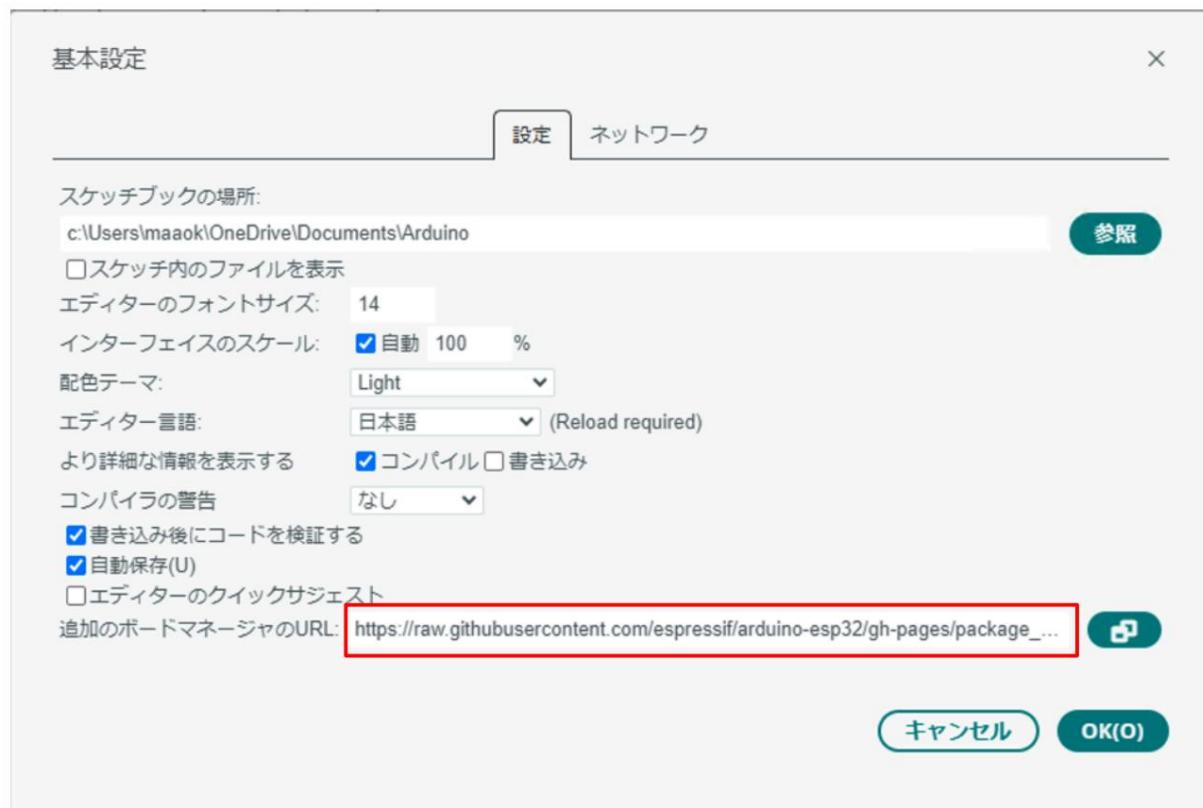


Added ESP32 board information

Add the ESP32 board information to the Arduino IDE so that you can write sketches to this product. Select File -> Preferences... from the menu bar. Enter the "Additional Boards Manager URL" at the bottom of the screen.

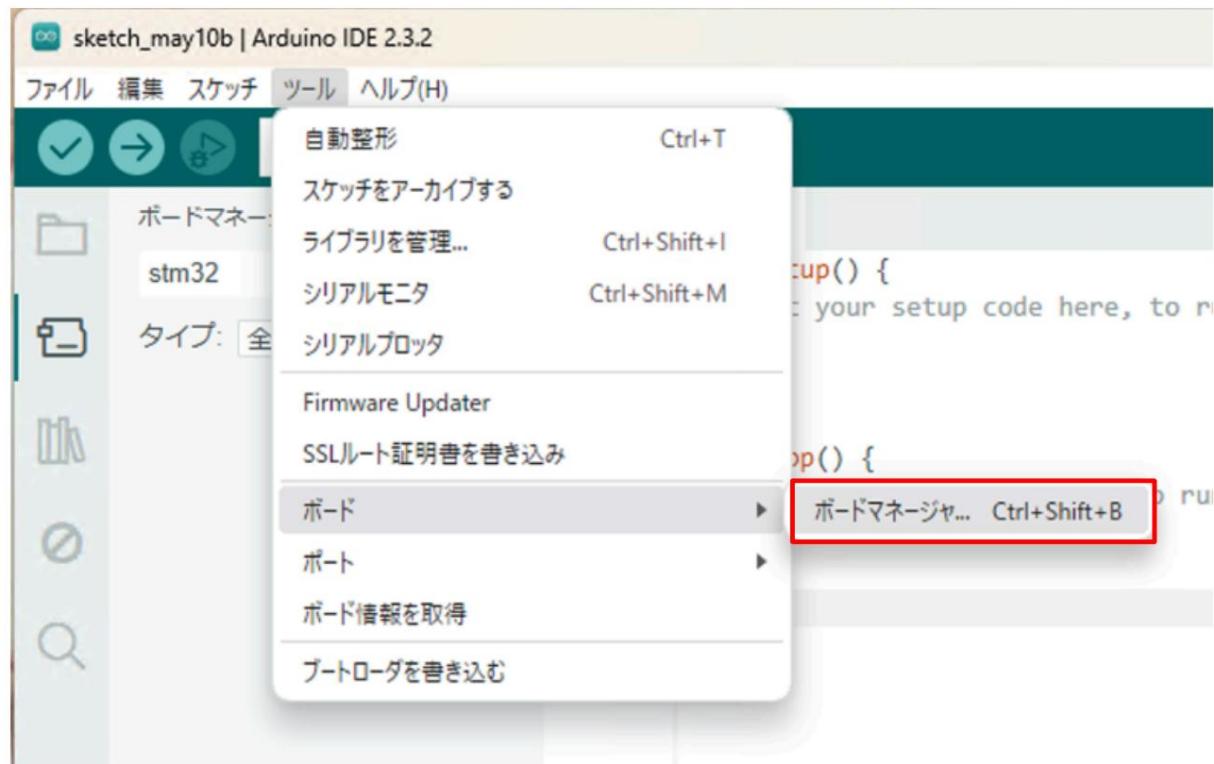
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

Add it and click "OK".



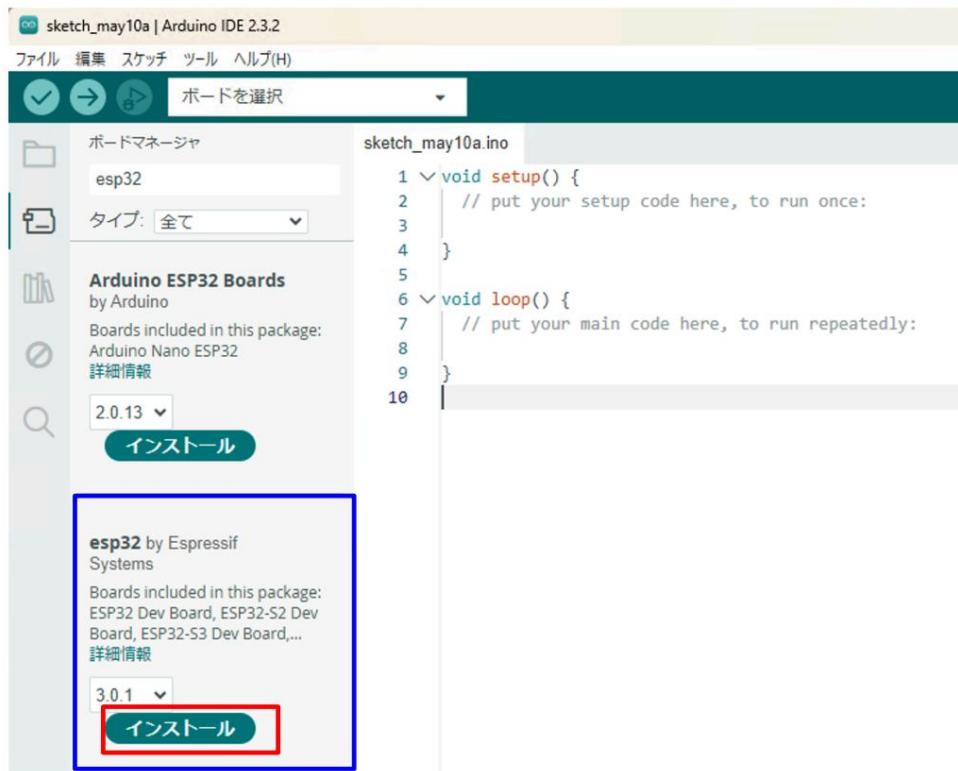
Added to the URL of the additional board manager

Once you have added the URL, download and install the ESP32 libraries and tools: from the menu bar, select Tools -> Board -> Boards Manager...



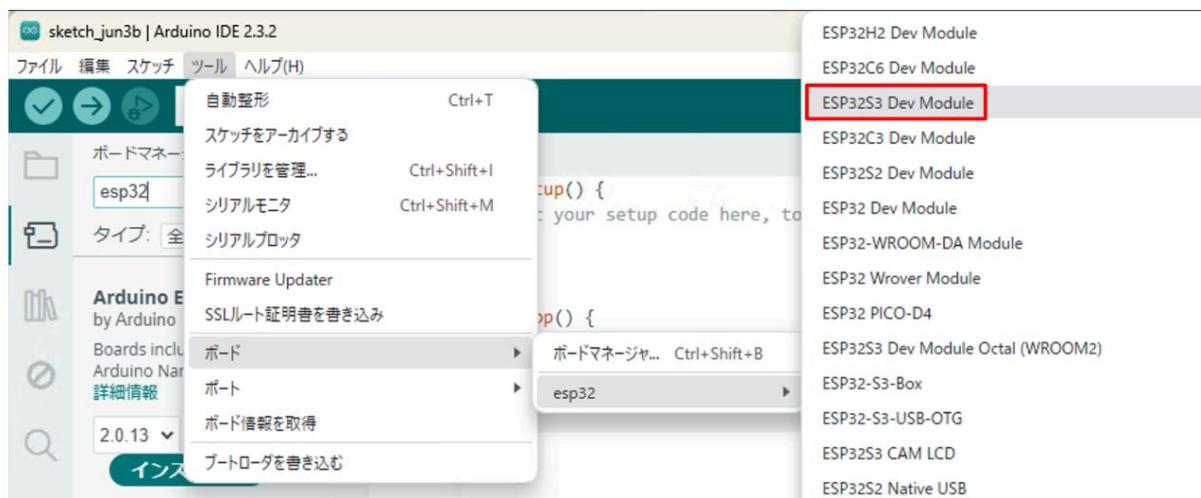
Boards Manager... selected

Enter "esp32" in the search field of the Board Manager on the left and "esp32 by Espressif Systems" will be displayed. As of June 2024, 3.0.1 is the latest version. Click Install esp32 by Espressif System.



Board Manager screen with "esp32" entered in the search field

Since the Arduino IDE supports microcontrollers other than the ESP32, select the ESP32 board when writing a sketch. Select Tools -> Board -> esp32 -> ESP32S3 Dev Module from the menu bar. The "Board" and "Port" are written in similar letters, so be careful not to select the wrong one.



Select Tools -> Board -> esp32 -> ESP32S3 Dev Modules

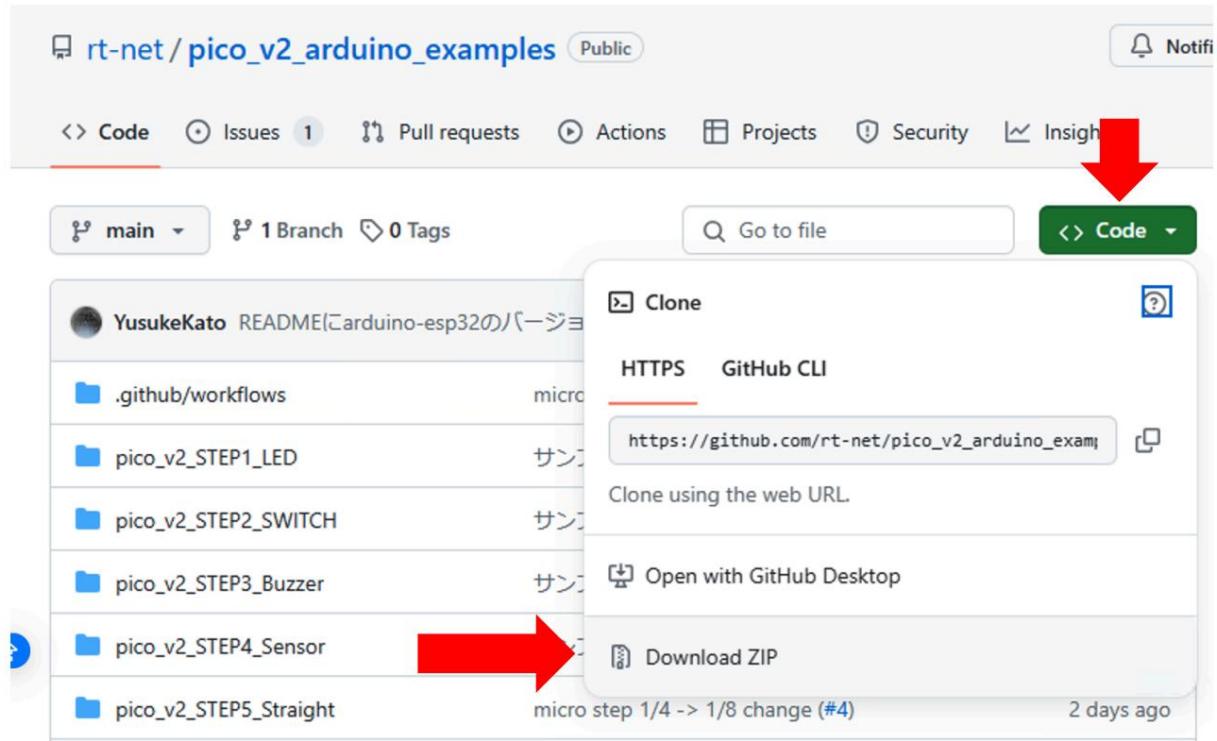
This completes the build environment setup for ESP32-S3.

Prepare the sample sketch. Access

https://github.com/rt-net/pico_v2_arduino_examples in a web browser and download the sample sketch.

Download the V2 example sketch file.

You can download it by clicking "Download ZIP" from "Code" on the page. Please unzip the downloaded zip file.



Select Code->Download ZIP to download the sample sketch.

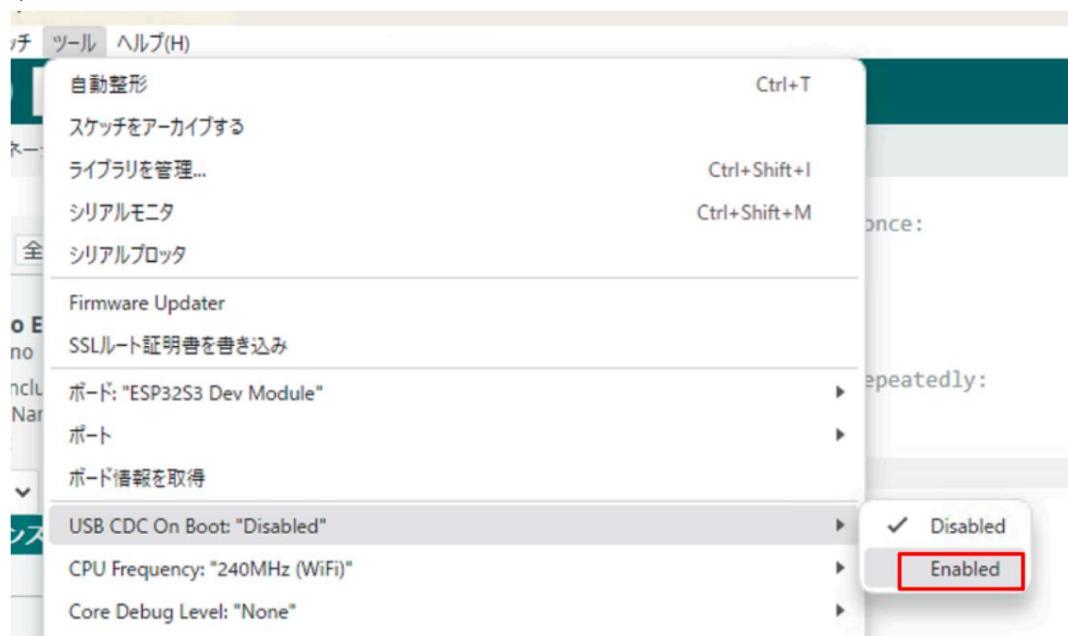
Here we will use the sample sketch (pico_v2_STEP1_LED) that lights up the LED of Pi:Co V2 as an example. This article explains how to build and write a sketch.

Write the sketch to this product via USB. By default, writing from USB is not possible.

Therefore, you will need to make the following two changes to enable writing.

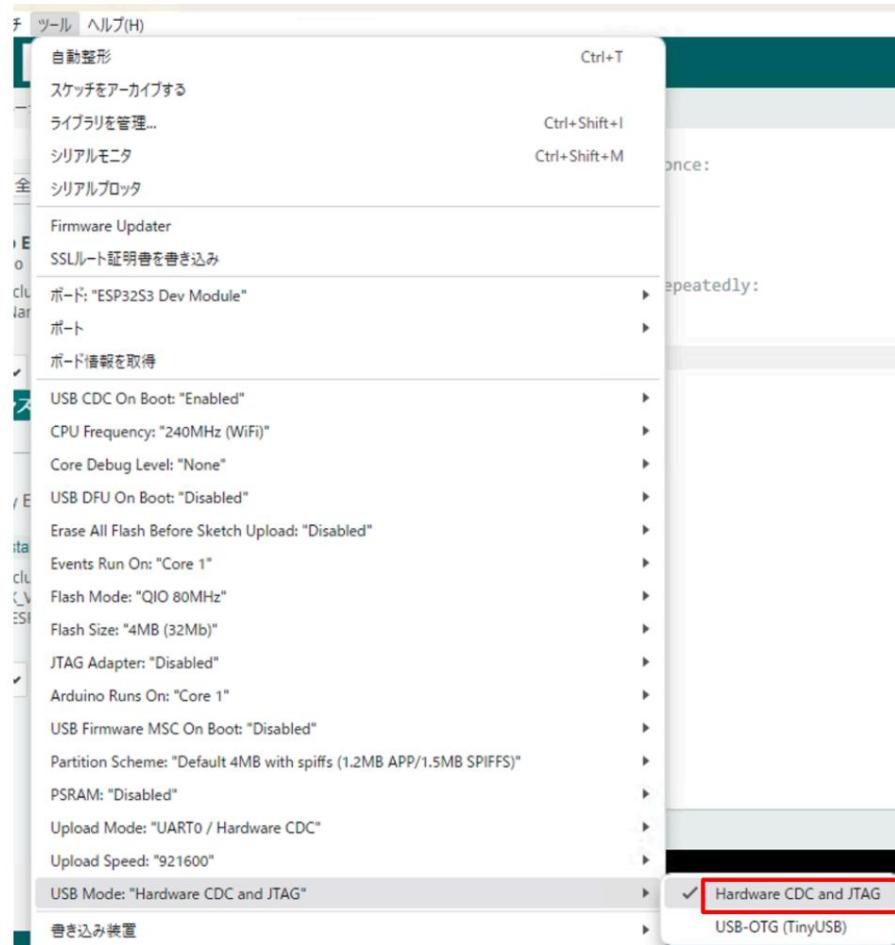
First, select Tools -> Board -> esp32 -> ESP32S3 Dev Module from the menu bar. Next

Next, open Tools -> USB CDC On Boot from the menu bar and set it to "Enabled".



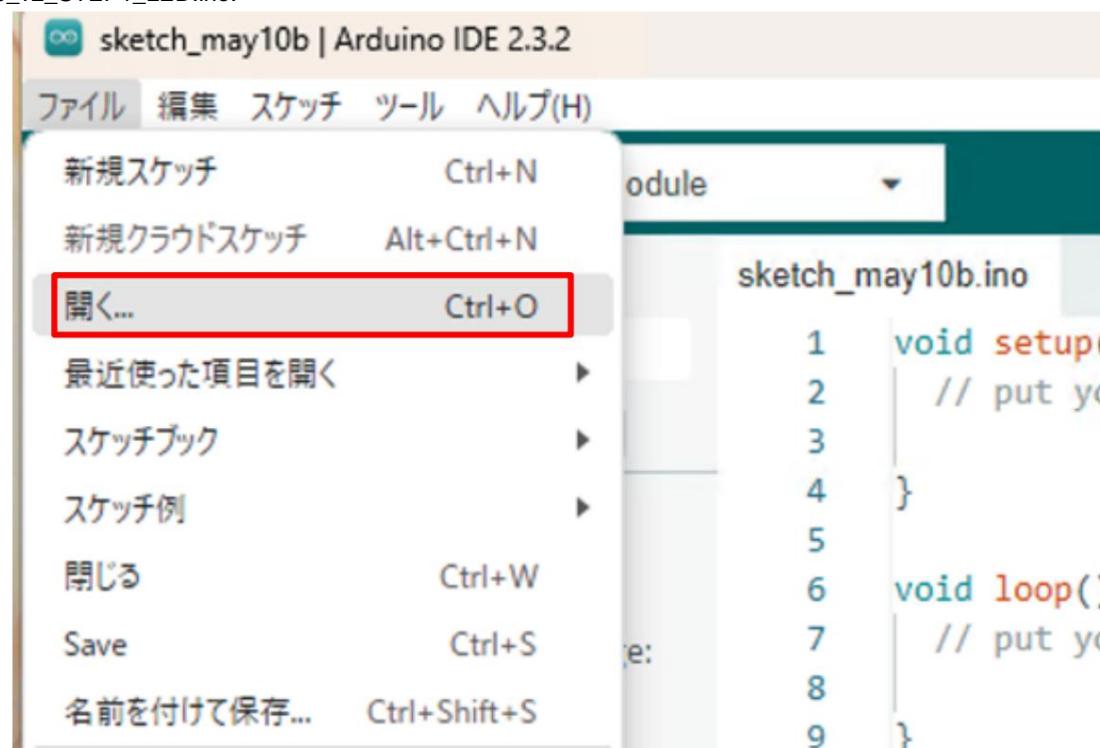
Set USB CDC On Boot to Enabled

Open Tools -> USB Mode on the menu bar and select "Hardware CDC and JTAG".



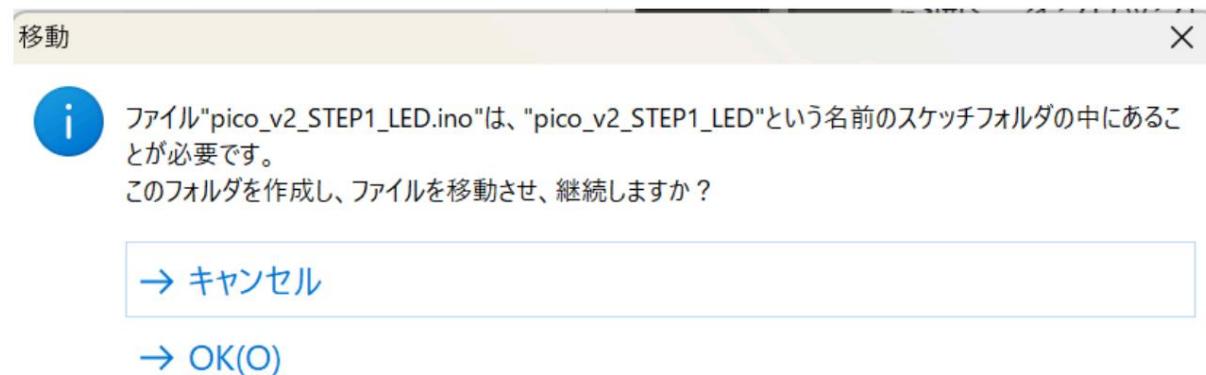
Set USB Mode to Hardware CDC and JTAG

pico_v2_STEP1_LED/pico_v2_STEP1_LED.ino in the downloaded sample sketch
 Double-click the file to open it, or start the Arduino IDE and click File -> Open... on the menu bar and open pico_v2_STEP1_LED.ino.



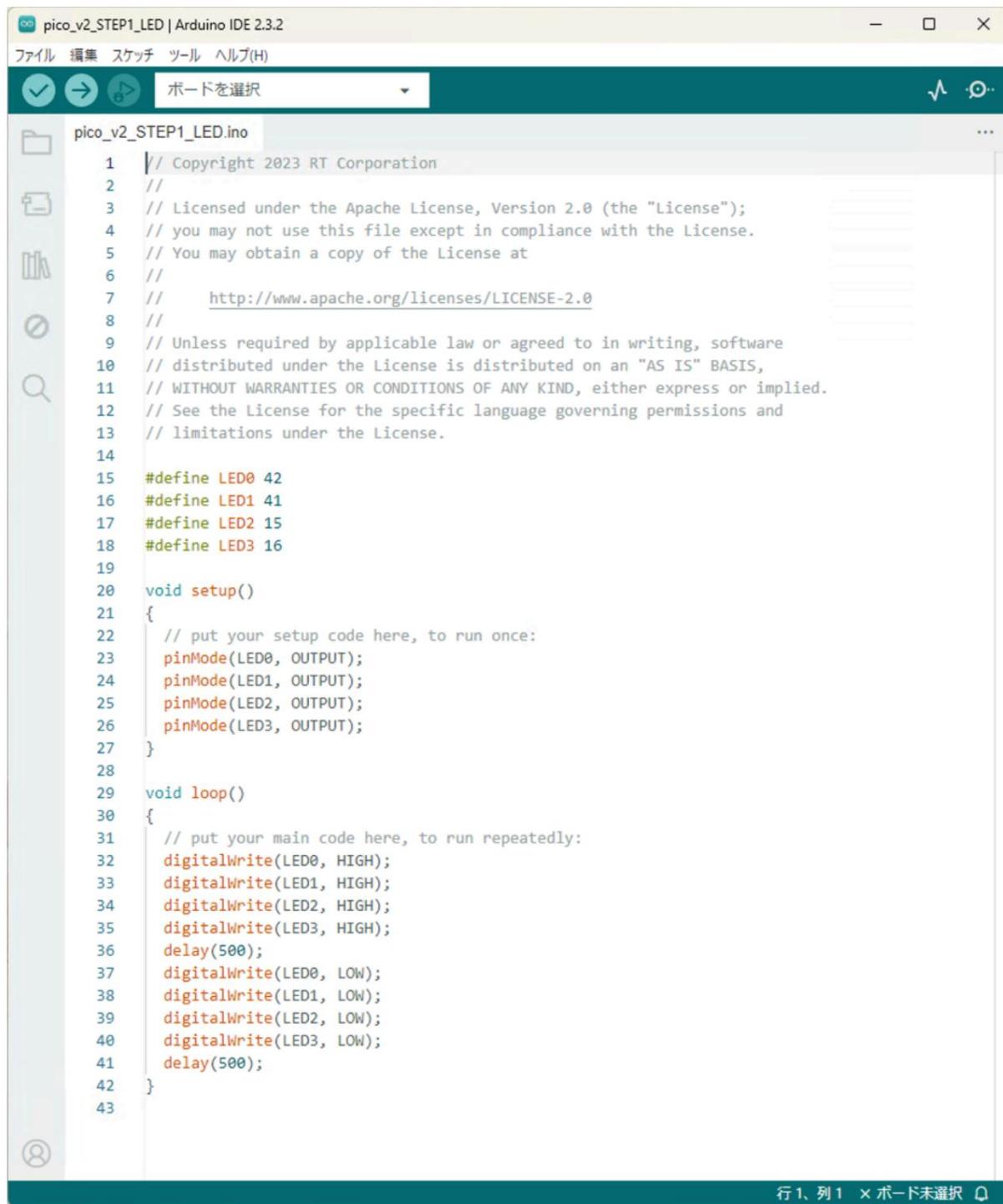
Select File -> Open...

If the following message appears when opening a sketch file, it means that the sketch file name and folder name do not match, or the folder does not exist, and you are being asked if you want to create a new folder. Due to the specifications of the Arduino IDE, this message appears when the main "file name" and "folder name" are not the same. If you change the file name or folder name, make sure to give them the same name.



Confirmation screen when file name and folder name do not match

When you open the sample sketch, the following screen will be displayed.



The screenshot shows the Arduino IDE interface with the title bar "pico_v2_STEP1_LED | Arduino IDE 2.3.2". The menu bar includes "ファイル" (File), "編集" (Edit), "スケッチ" (Sketch), "ツール" (Tools), and "ヘルプ(H)". A dropdown menu "ボードを選択" (Select Board) is open. The code editor displays the "pico_v2_STEP1_LED.ino" file, which contains the following code:

```
// Copyright 2023 RT Corporation
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

#define LED0 42
#define LED1 41
#define LED2 15
#define LED3 16

void setup()
{
    // put your setup code here, to run once:
    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}

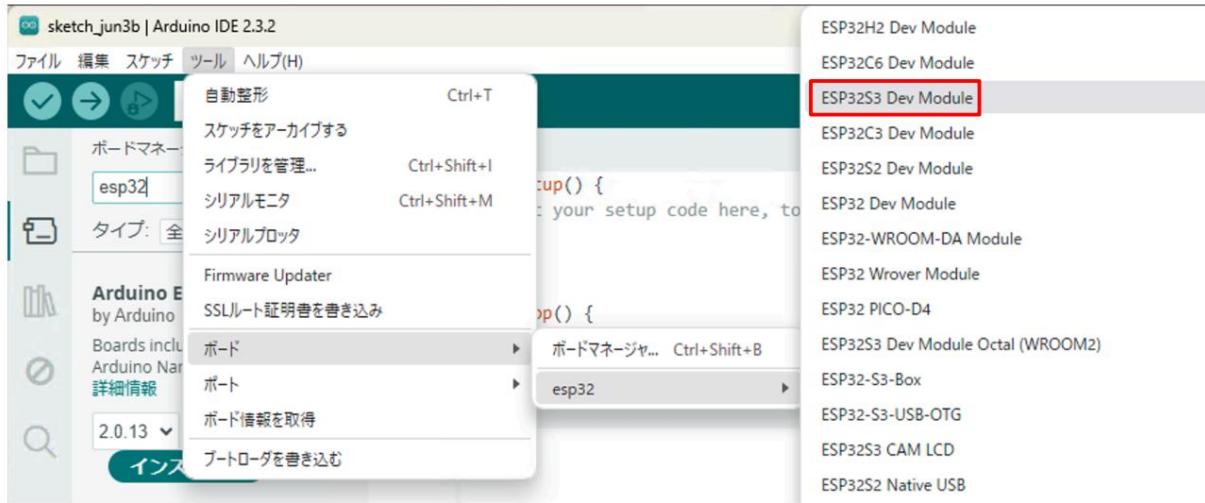
void loop()
{
    // put your main code here, to run repeatedly:
    digitalWrite(LED0, HIGH);
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    delay(500);
    digitalWrite(LED0, LOW);
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    delay(500);
}
```

The status bar at the bottom right shows "行 1、列 1 × ボード未選択" (Line 1, Column 1 × Board not selected).

pico_v2_STEP1_LED.ino open

If the board is not set to “ESP32S3 Dev Module”, go to Tools -> Board in the menu bar.

Select ->esp32->ESP32S3 Dev Module.



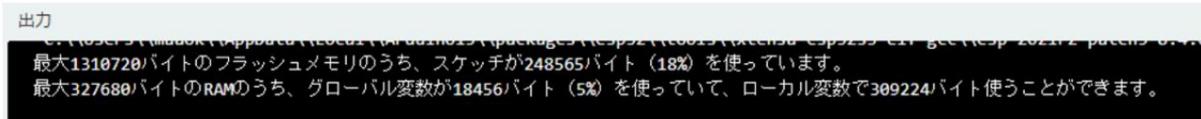
Select Tools->Board->esp32->ESP32S3 Dev Module

Build



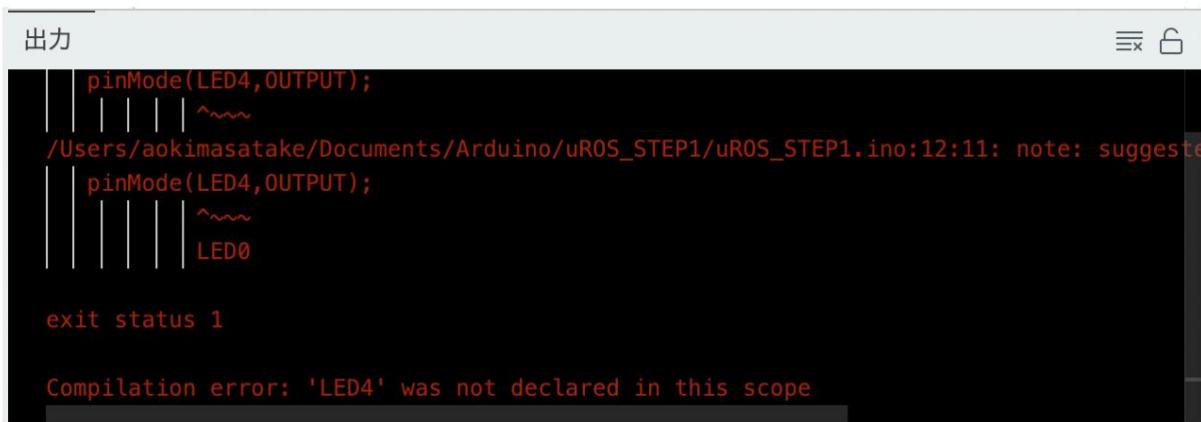
Click the icon in the upper left corner to run "Verify" and build the sketch.

If there are no problems, the following message will be displayed:



Message when there is no problem with the validation

If there is an error in the sketch, the error will be displayed in red.



Validation error message



You can hide the message by clicking the icon at the bottom right of the Arduino IDE. Click it again to show it again. Use this function when you want to enlarge the screen to see more of the entire sketch, or when you do not need this message screen.



RT CORPORATION

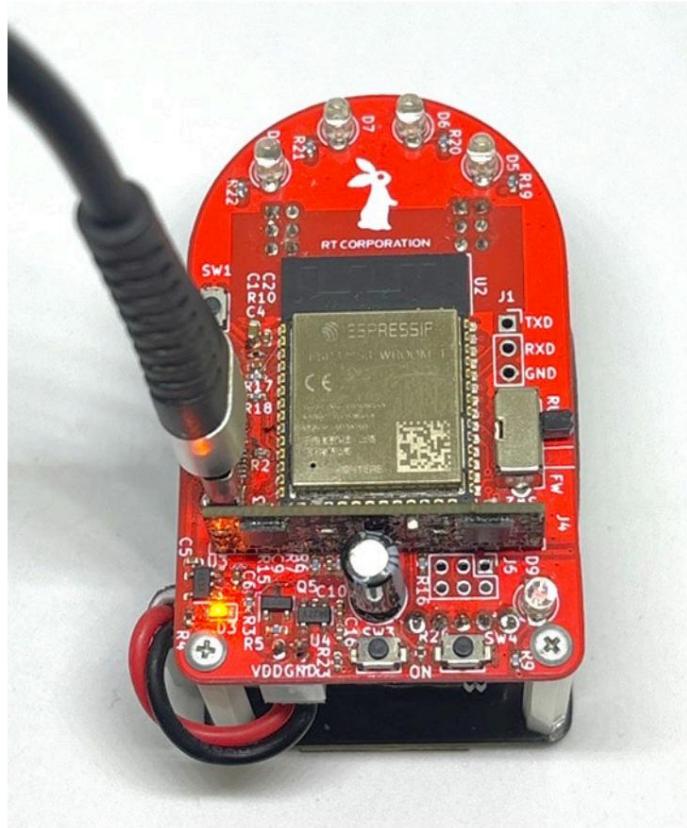
For Ubuntu, if the message "ModuleNotFoundError: No module named 'serial'" is displayed, the build has not been completed. Execute the following command in the terminal to install the apps required for the build.

```
$ sudo apt-get install python-pip $ pip install  
pyserial $ pip install esptool
```

write

Once you have confirmed that there are no errors in the sketch, follow these steps to prepare for writing:

1. Make sure the battery is connected to the device.
2. Connect the PC and this product with a USB cable.
3. Slide the mode switch of this product to the direction marked FW.
4. Slide the power switch of Pi:Co V2 to the ON side.
5. Press the reset button of this product.



Mode switch operated and USB cable connected

After connecting this product to the PC, follow the steps below to set up the board and port.

Please set it again after turning it off or pressing the reset button.

Click the board name (ESP32S3 Dev Module) displayed under the menu bar and open "Select Other Board and Port..." at the bottom.



Open the board name -> Select other board and port...

Enter "dev" in the board search field and make sure "ESP32S3 Dev Module" is checked.

If it is not checked, click ESP32S3 Dev Module. In the Port Selection field, select the USB port to which this product is connected. For Windows, select COMX.

For Linux (Ubuntu), it is called /dev/ttyUSBX, and for macOS, it is called usbmodemXXXXX Serial Port(USB). Once you have selected it, click "OK".

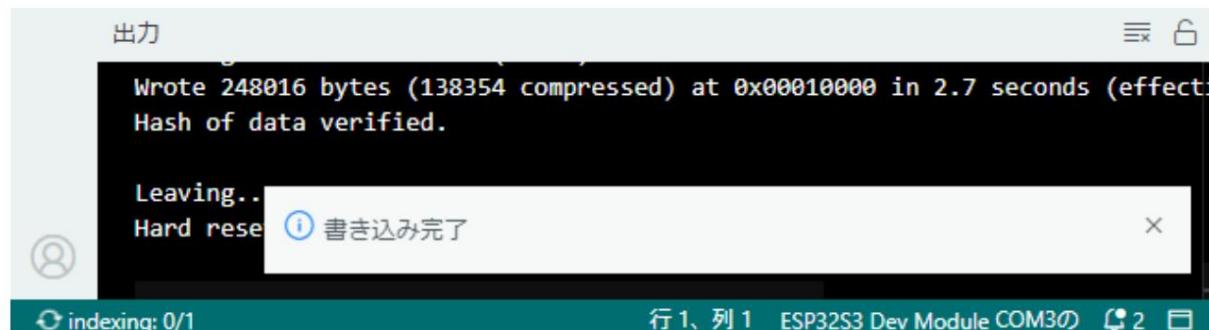


After completing the setup, a USB mark will be added to the left of the board name. In addition to writing sketches, port settings are also required when communicating with a PC via hardware serial communication.



After setting the port, click to write the sketch.

Writing is now complete.



Screen display when writing is complete

If writing fails on a Linux (Ubuntu) PC, run the following command in the terminal before writing to change the USB port permissions.

```
$ sudo chmod 777 /dev/ttyACM0
```

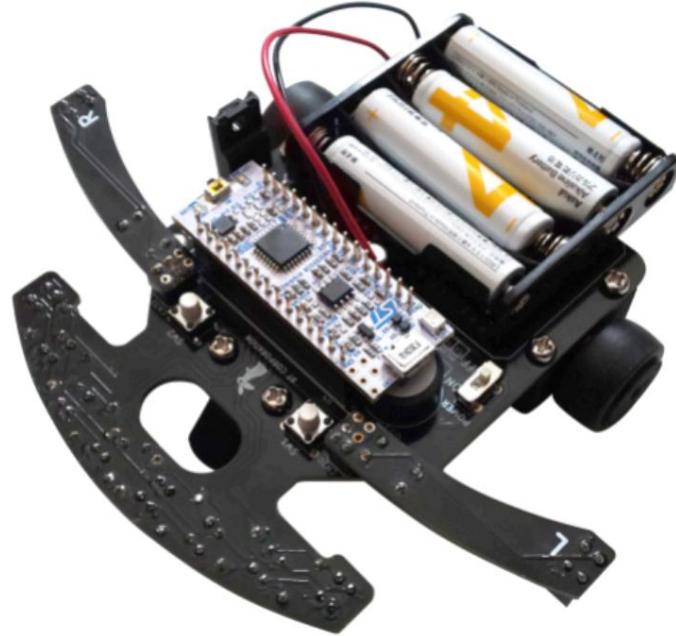
To change the USB port usage permissions permanently, run the following command and restart your PC.

```
# Restart the PC after executing the following: $  
sudo usermod -aG dialout $USER
```

After writing the sketch, slide the mode switch of this product to the RUN side and reset it.
Press the Set button.

The LED on the front of the Pi:Co V2 will blink for 0.5 seconds and then turn off for 0.5 seconds.

Training Tracer



Added STM32 board information

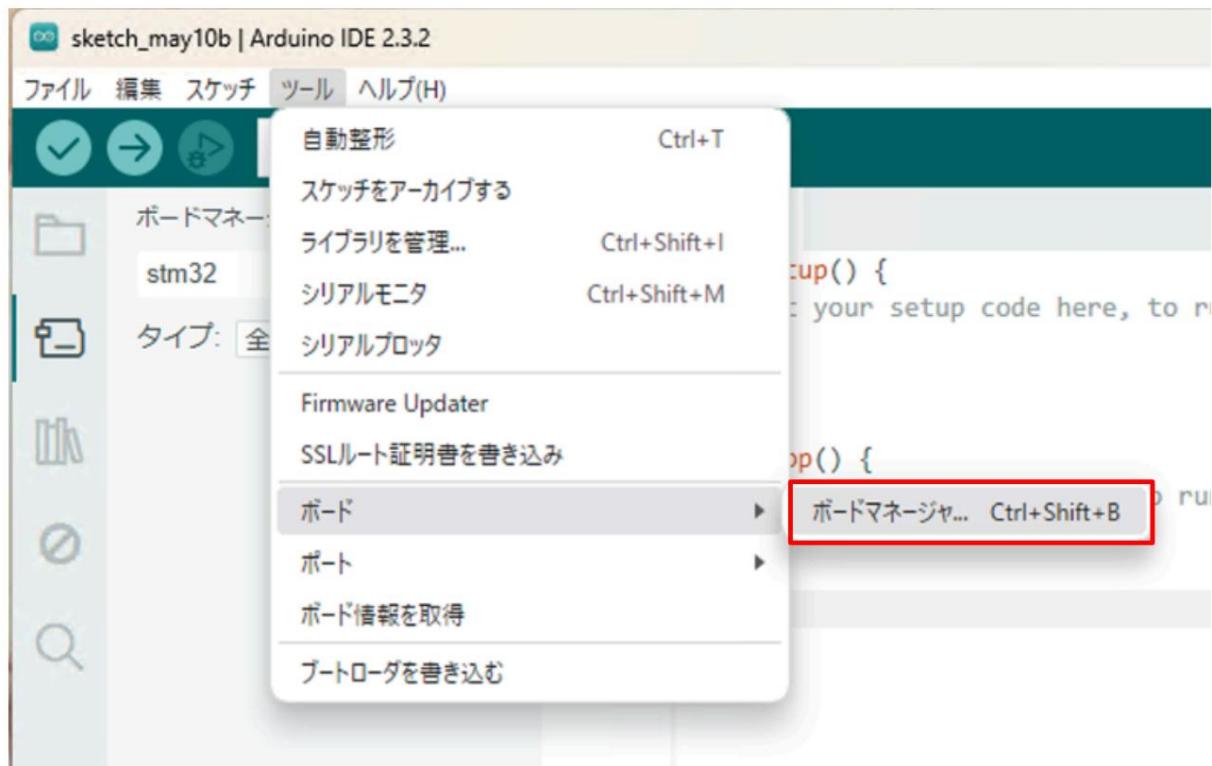
Add the STM32 board information to the Arduino IDE so that you can write sketches to this product. Select File -> Preferences... from the menu bar. Enter the "Additional Boards Manager URL" at the bottom of the screen.

Add https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json and click "OK".



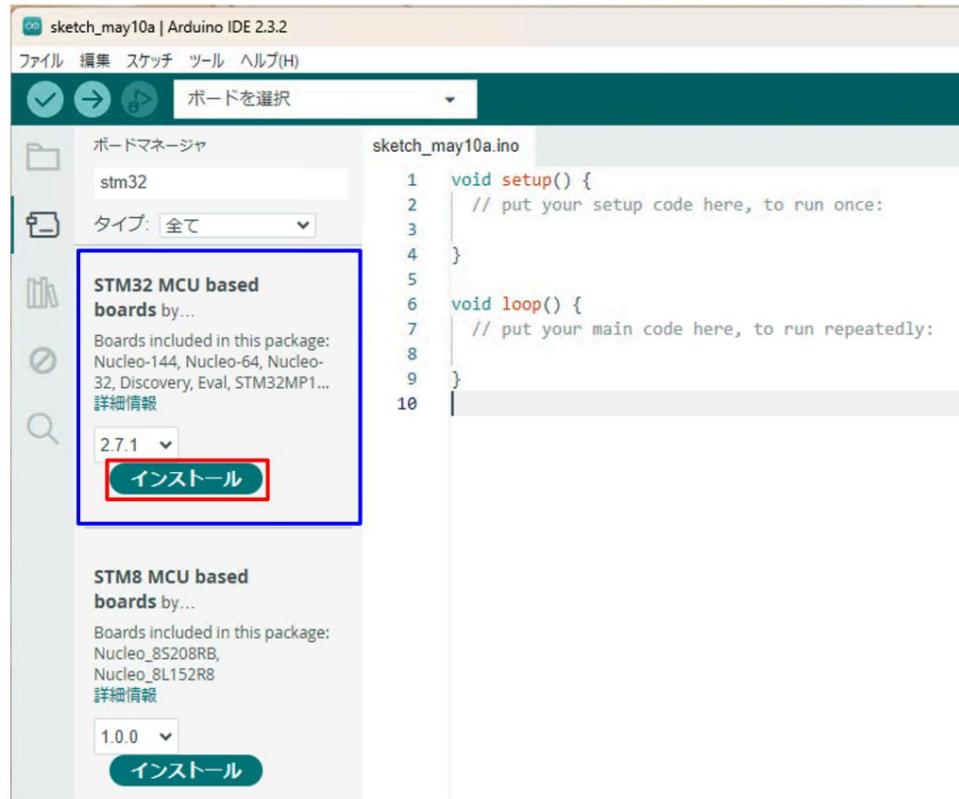
Added to the URL of the additional board manager

Once you have added the URL, download and install the STM32 libraries and tools: select Tools -> Board -> Boards Manager... from the menu bar.



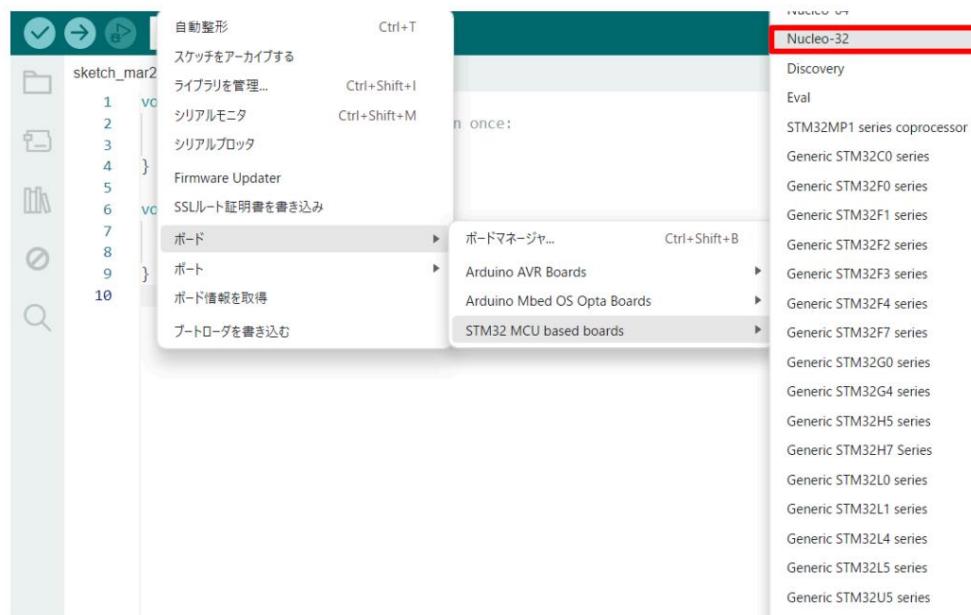
Boards Manager... selected

Enter "esp32" in the search field of the board manager on the left and "STM32 MCU Based boards by STMicroelectronics" will be displayed. As of June 2024, 2.7.1 is the latest version. Click Install STM32 MCU Based boards by STMicroelectronics.



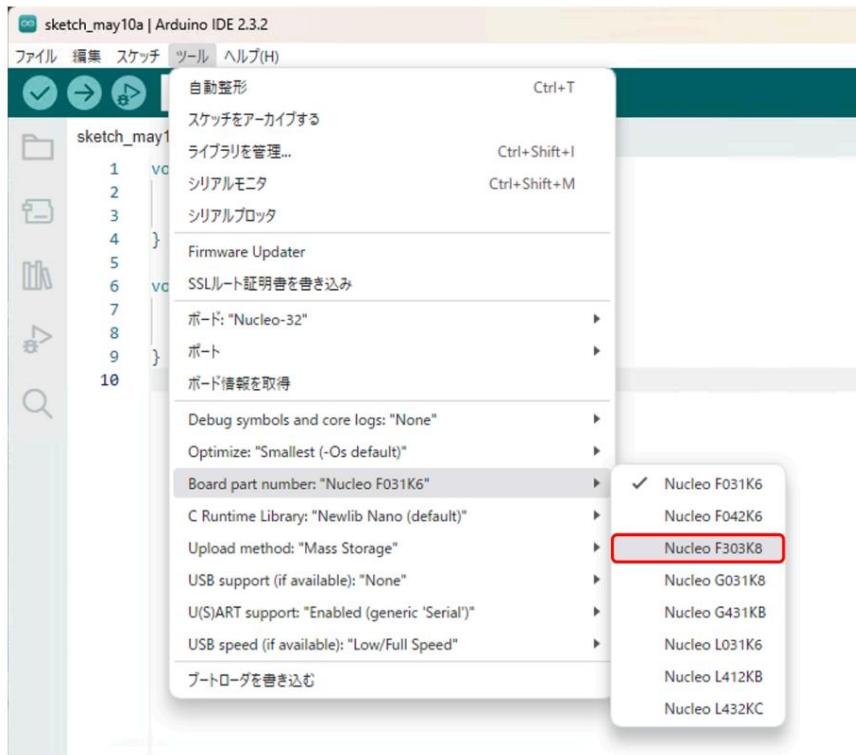
Boards Manager screen with "stm32" entered in the search field

Since Arduino IDE supports microcontrollers other than STM32, select the STM32 board when writing the sketch. Select Tools -> Board -> STM32 MCU based boards -> Nucleo-32 from the menu bar. "Board" and "Port" are written in similar letters, so be careful not to select the wrong one.



Select Tools->Boards->STM32 MCU based boards->Nucleo-32

Next, select the board you want to use from Nucleo-32. Select Nucleo
Select F303K8.



Select Tools->Board part Number->Nucleo F303K8. This completes the build environment settings for STM32 Nucle F303K8.

Installing the programming tool STM32CubeProgrammer

To write a program to the Nucleo board from the Arduino IDE, you need to use STMicroelectronics' You will need STM32_Programmer_CLI, which is included in STM32CubeProgrammer.
[STM32CubeProgrammer is available at https://www.st.com/ja/development-tools/stm32cubeprog.html](https://www.st.com/ja/development-tools/stm32cubeprog.html)
Download from. Click "Get the latest version" to download the software that matches your OS.

ソフトウェア入手

製品型番	概要	ダウンロード	すべてのバージョン
+ STM32CubePrg-Lin	STM32CubeProgrammer software for Linux	最新バージョンを取得	バージョンを選択
+ STM32CubePrg-Mac	STM32CubeProgrammer software for Mac	最新バージョンを取得	バージョンを選択
+ STM32CubePrg-W32	STM32CubeProgrammer software for Win32	最新バージョンを取得	バージョンを選択
+ STM32CubePrg-W64	STM32CubeProgrammer software for Win64	最新バージョンを取得	バージョンを選択

Read the license agreement and click "I accept".



Log in with your MyST account. If you don't have an account, create one.



Enter your registered email address and password and click Login.

mySTログイン

メールアドレスとパスワードを入力し、ログインしてください

メールアドレス
[Input field]

パスワード
[Input field]

メールアドレスを記憶 ○

ログイン

パスワードを忘れた場合

新規に登録される方

mySTのパーソナライズ機能

- イベントへの参加
- ニュースレター配信
- オンライン・サポート利用
- STコミュニティへの参加
- オンライン設計ツール利用
- ソフトウェア・ダウンロード
- 無償サンプル注文
- 製品情報（毎週更新）
- ST製品 & ツール購入

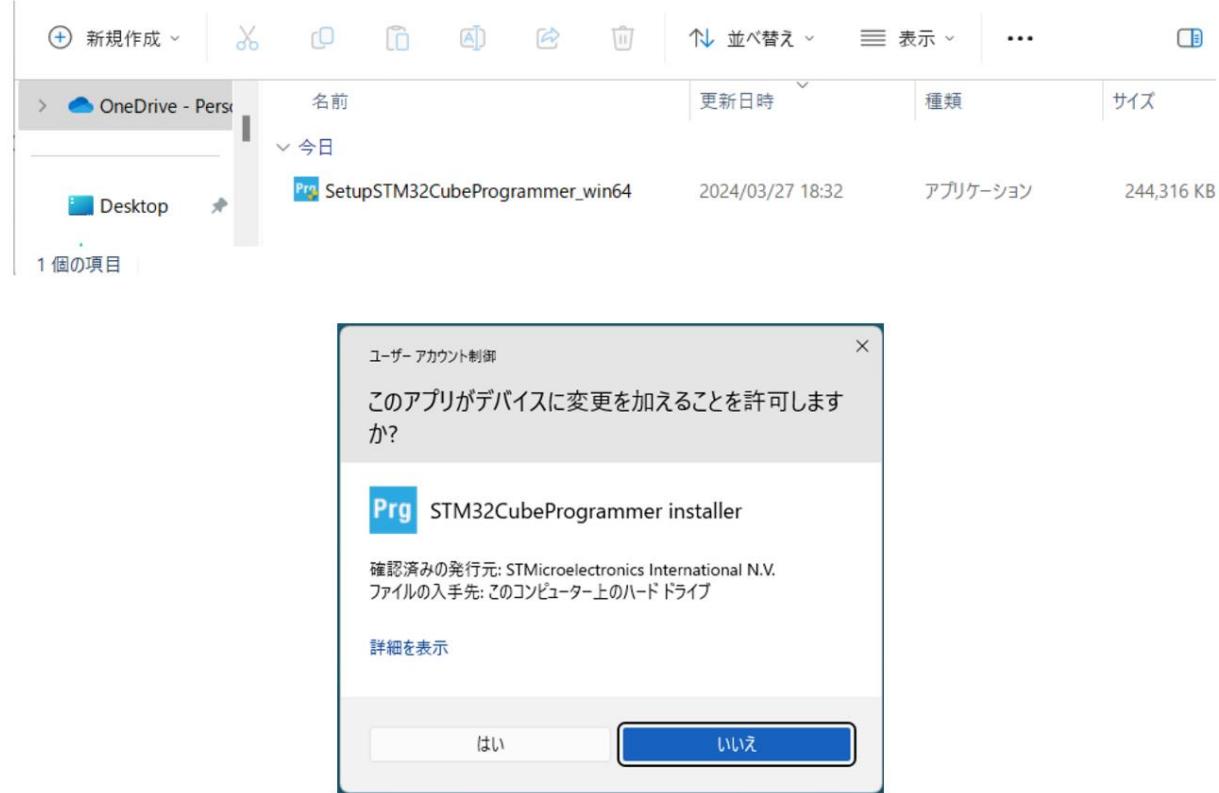
新規登録

Once you have successfully logged in, the download will begin.

The method for starting the STM32CubeProgrammer installer varies depending on the OS, but the steps are almost the same. Here, we will explain using Windows screens.

How to run the installer on Windows

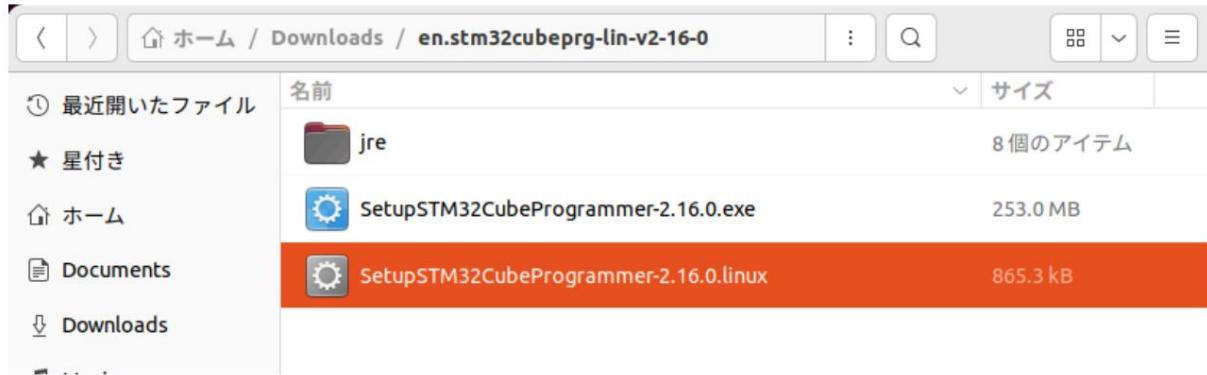
Right-click on the downloaded en.stm32cubeprog-win64-v2-16-0.zip file and select "Select All".
Select "Extract" and unzip the file. Double-click SetupSTM32CubeProgrammer_win64.exe in the extracted folder to run it.



Click "Yes".

How to start the installer on Linux (Ubuntu) Right-

click the downloaded en.stm32cubeprg-lin-v2-16-0.zip file to unzip it. Double-click SetupSTM32CubeProgrammer-2.16.0.linux in the unzipped folder to run it.



How to run the installer on macOS

Double-click the downloaded en.stm32cubeprg-mac-v2-16-0.zip file to unzip it, and then double-click SetupSTM32CubeProgrammer-2.16.0.

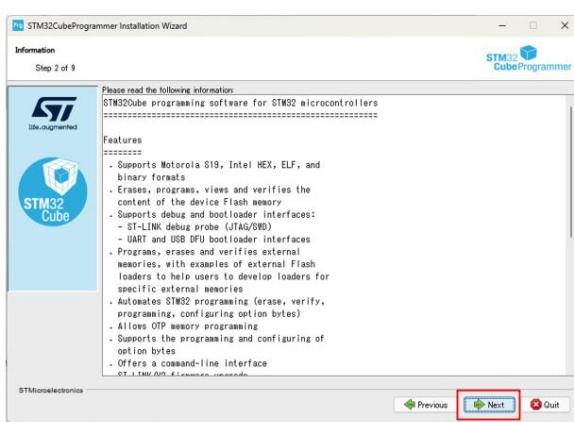


Click "Open".

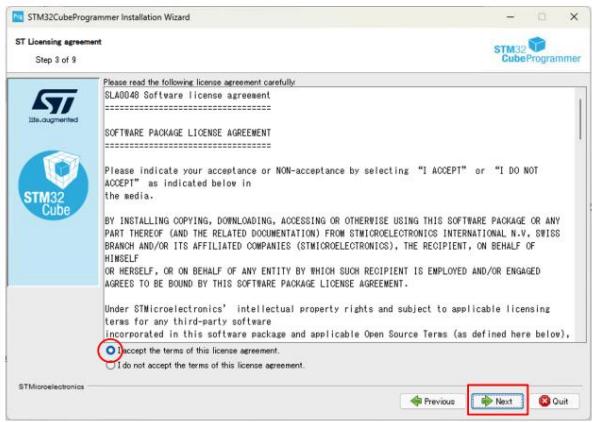
Installer operation procedure (for all OS)



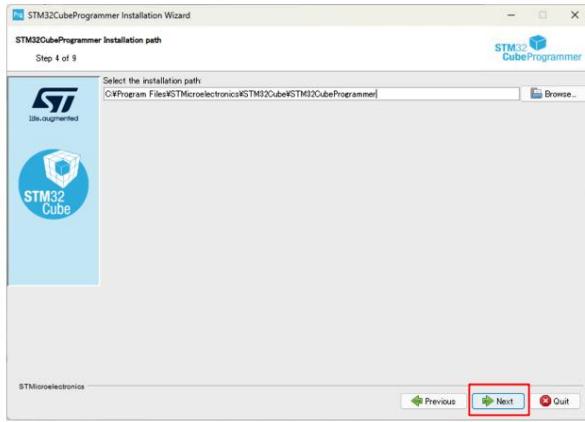
Click Next.



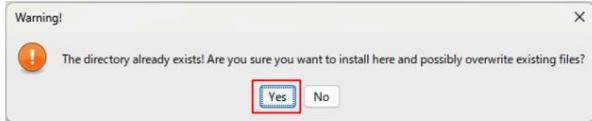
Click Next.



Select I accept and click Next.

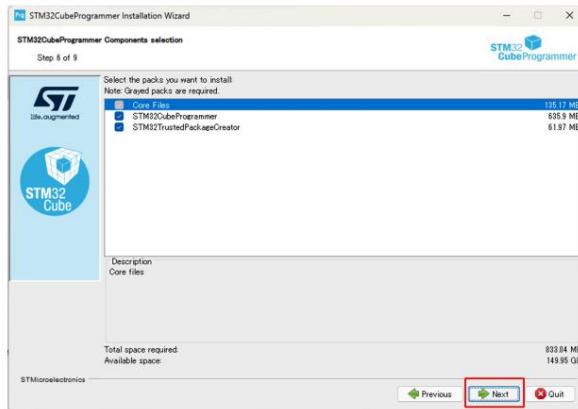


Click Next.

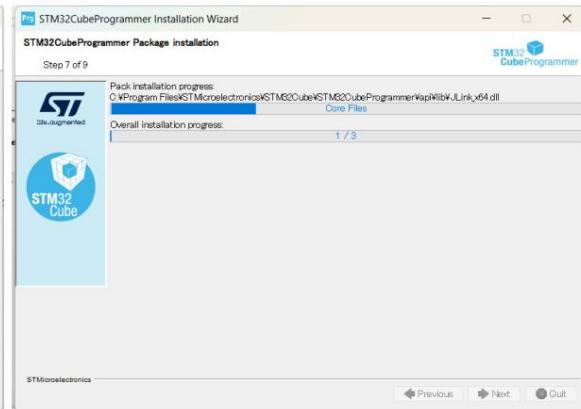


Click YES. Check I have and click Next.



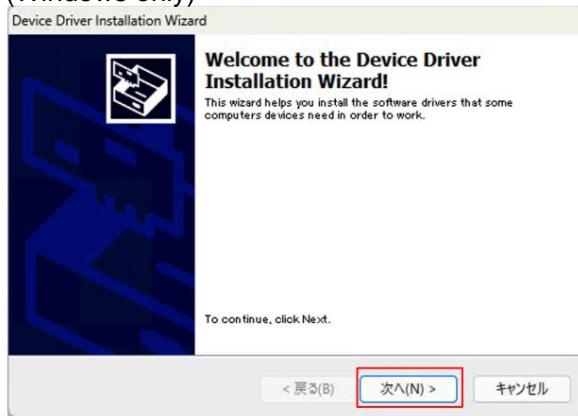


Click Next.



The installation will begin.

(Windows only)

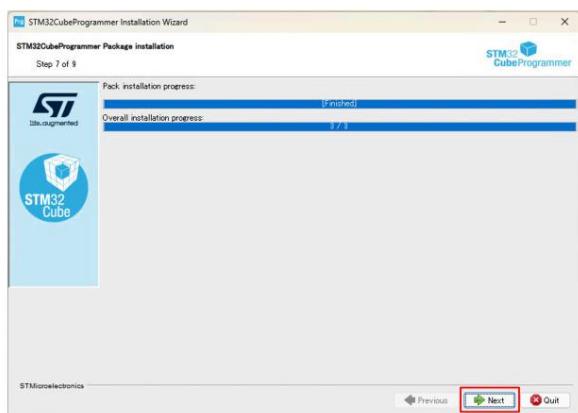


Click Next.)

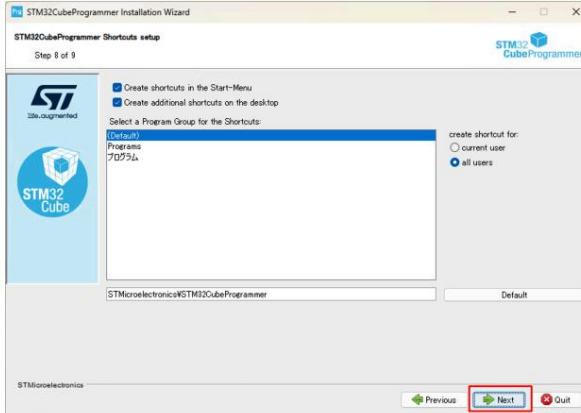
(Windows only)



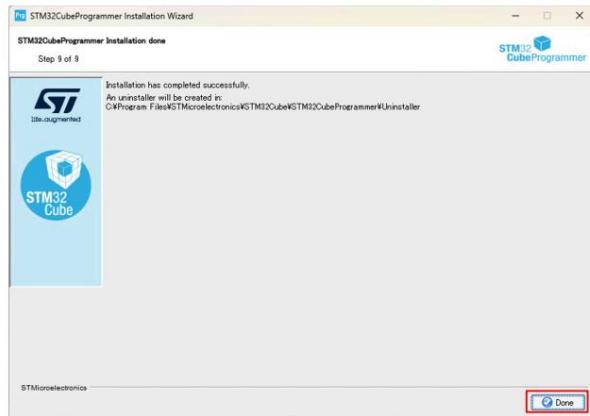
Click Finish.



Click Next.



Click Next.

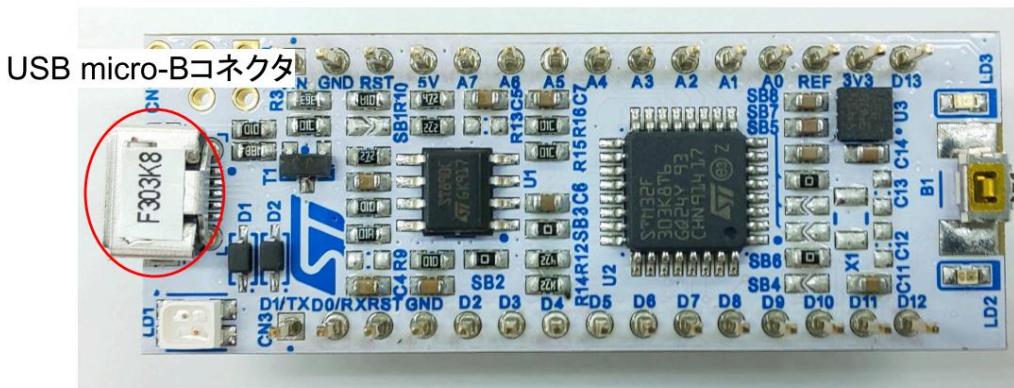


Click Done.

Installing device drivers (Windows only)

Here, the device driver used to write the program is installed on the PC.
Connect the Nucleo board to the PC with a USB cable.

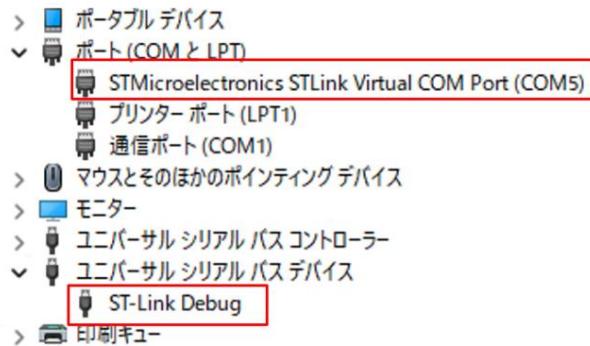
Please connect the cable while the Nucleo board is still placed on the training tracer.



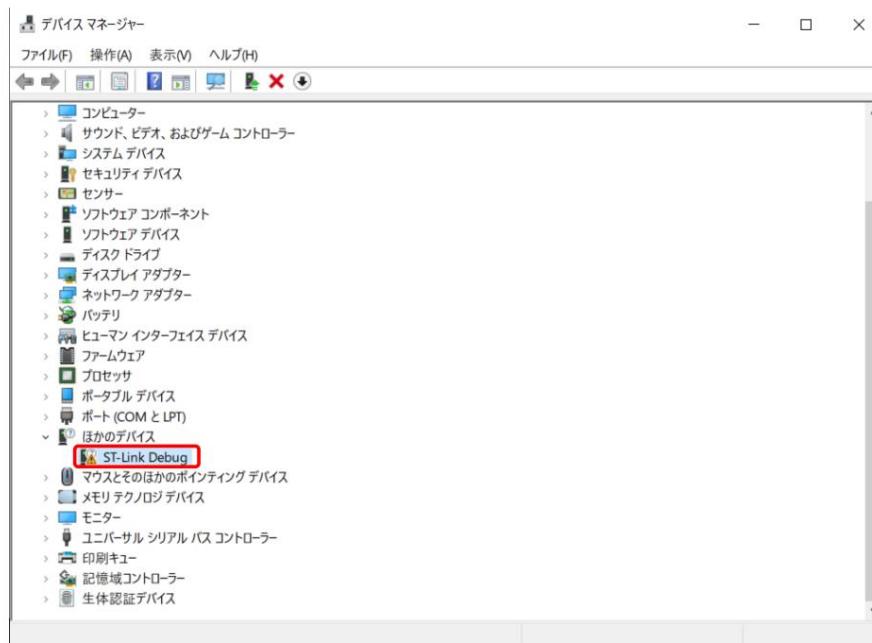
Click the Start button icon, type "Device Manager" and search
Click Device Manager in the search results.



If there is STMicroelectronics STLink Virtual COM Port (COMX) in the Ports (COM & LPT) section and ST-Link Debug in the Universal Serial Bus Devices section, you do not need to install the device driver.



There is a yellow triangle warning in Other devices or Unknown devices like the one below.
In this case, you will need to install the device driver.



To download the device driver, click on the latest version

[from STMicroelectronics' https://www.st.com/ja/development-tools/stsw-link009.html.](https://www.st.com/ja/development-tools/stsw-link009.html) and download it.

ソフトウェア入手

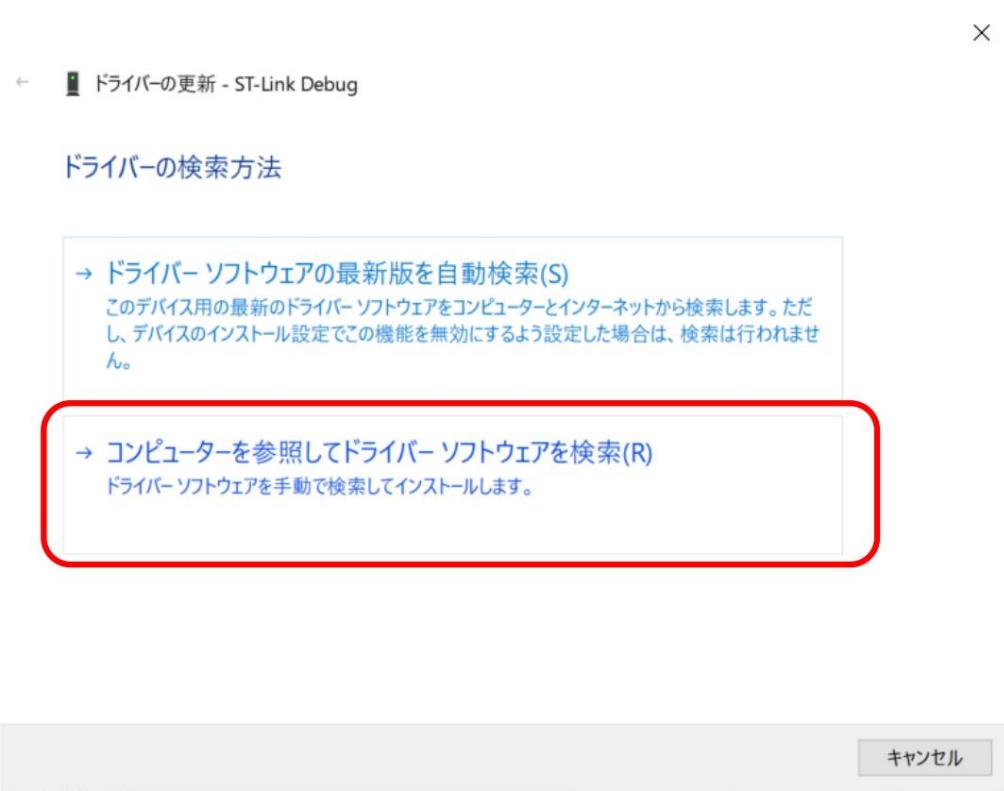
製品型番	▲ 最新バージョン	▼ ダウンロード
+ STSW-LINK009	2.0.2	最新バージョンをダウンロード

Click "Get Software". The license agreement confirmation screen will appear, so click "I Agree". Extract the downloaded

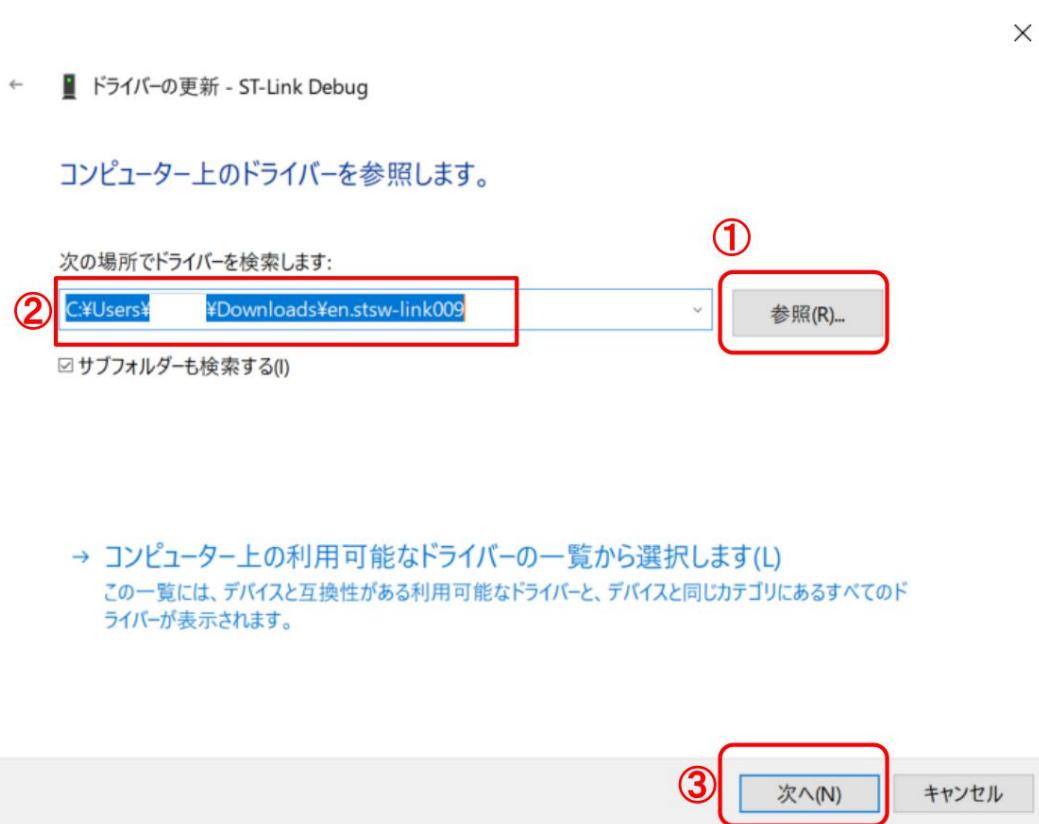
en.stsw-link009.zip. Double-click the area with the warning message in the Device Manager that you just opened to open it.



Once the Properties window opens, click "Update Driver".



When the driver search method screen appears, click "Browse my computer for driver software".



On the screen that says "Browse for drivers on your computer," click Browse, and then click the Select the folder containing the downloaded and unzipped en_stsw-link009.



On the "Windows Security" screen, click "Install".

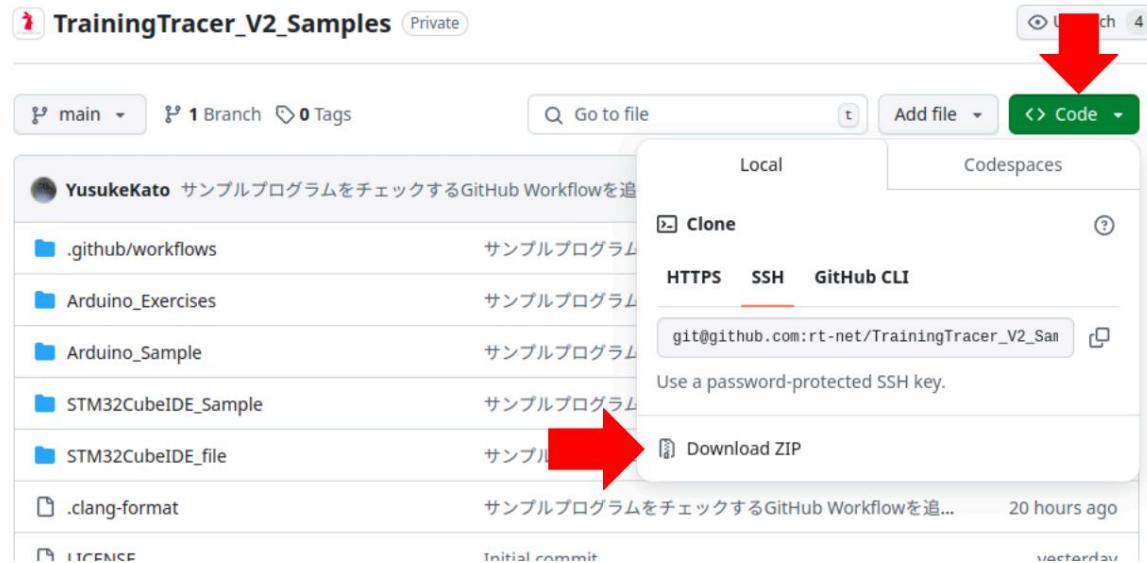


When you see the message "Updated successfully," click "Close."

Prepare sample sketch: https://github.com/rt-net/TrainingTracer_V2_Samples

in your web browser Go to
Download the training tracer example sketch file.

You can download it by clicking "Download ZIP" from "Code" on the page. Please unzip the downloaded zip file.

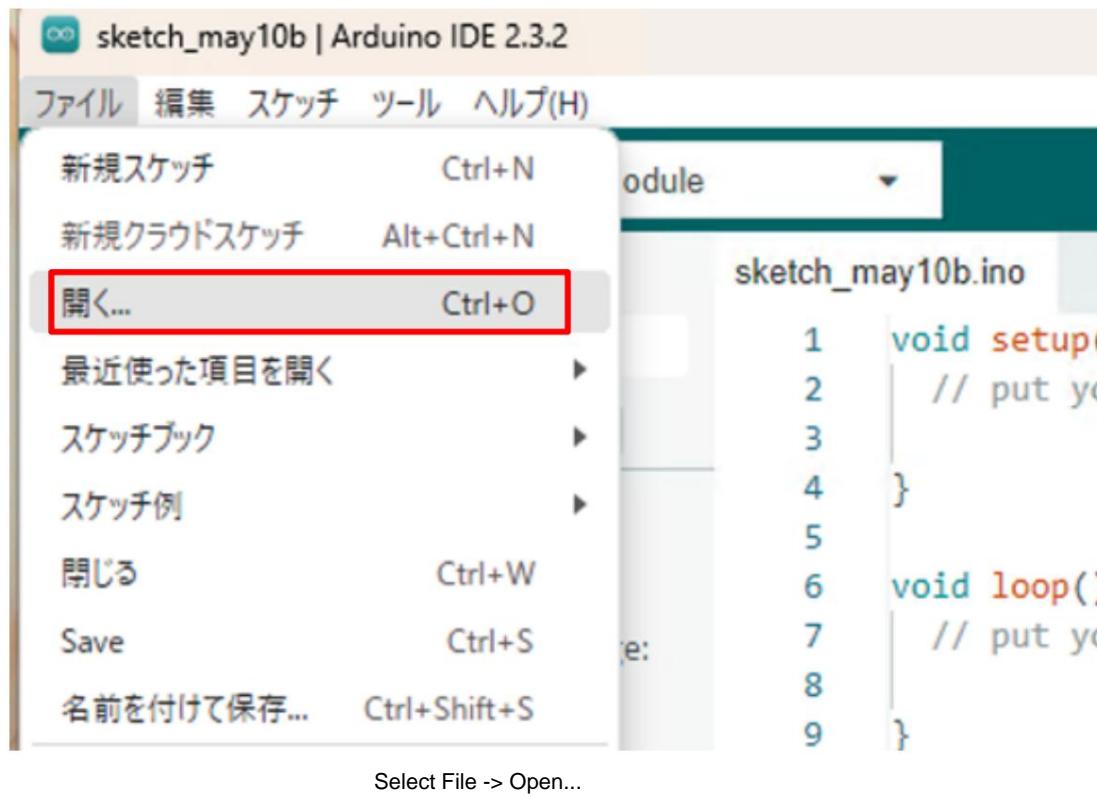


Select Code->Download ZIP to download the sample sketch.

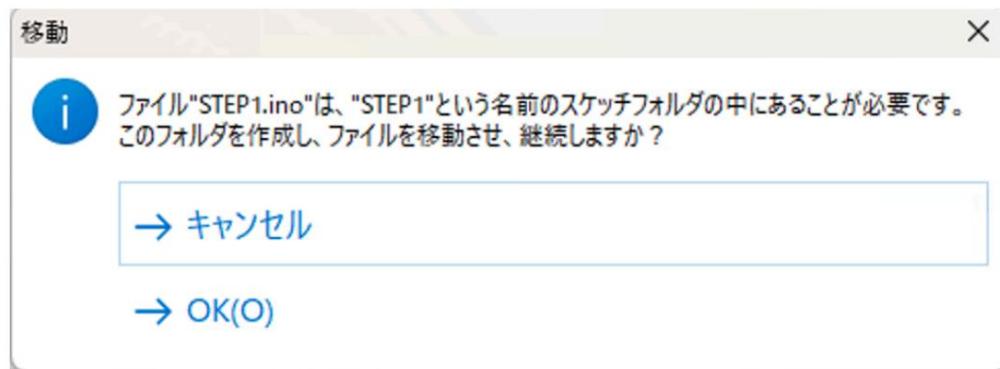
Here we will use a sample sketch (STEP 1) that lights up the LED on the Nucleo board as an example.
This explains how to build and write a sketch to this product.

Write the sketch to this product via USB.

Double-click the Arduino_Sample/STEP1/STEP1.ino file in the downloaded sample sketch to open it. Alternatively, start the Arduino IDE and click File -> Open... on the menu bar to open STEP1.ino.

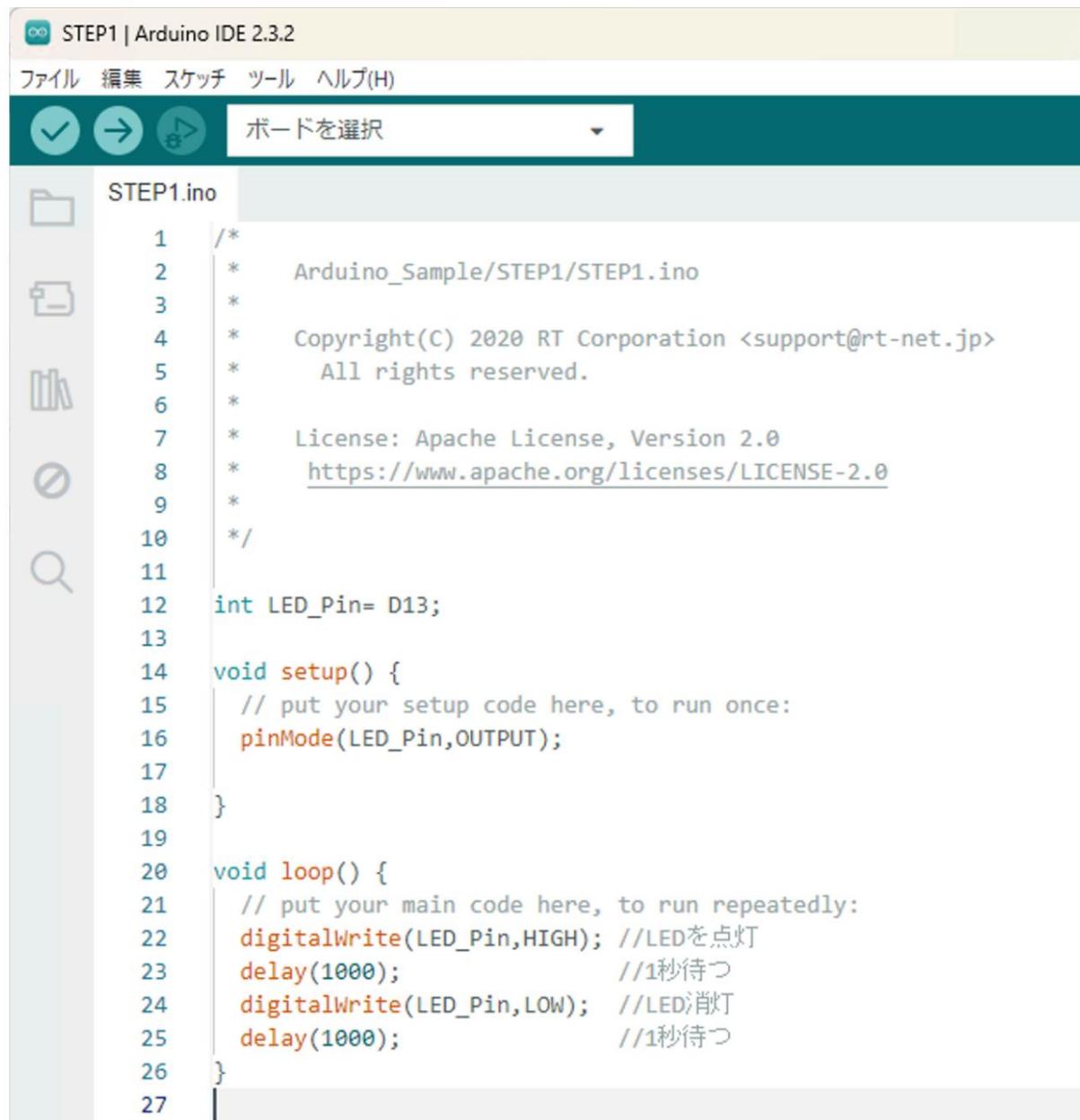


If the following message appears when opening a sketch file, it means that the sketch file name and folder name do not match, or the folder does not exist, and you are being asked if you want to create a new folder. Due to the specifications of the Arduino IDE, this message appears when the main "file name" and "folder name" are not the same. If you change the file name or folder name, make sure to give them the same name.



Confirmation screen when file name and folder name do not match

When you open the sample sketch, the following screen will be displayed.

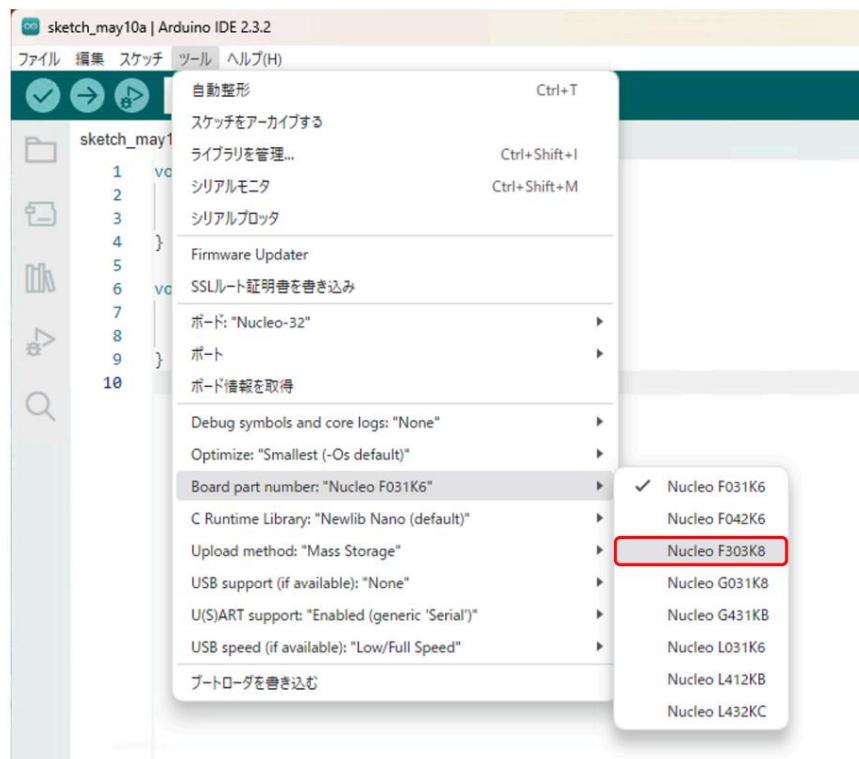


The screenshot shows the Arduino IDE interface with the title bar "STEP1 | Arduino IDE 2.3.2". The menu bar includes "ファイル" (File), "編集" (Edit), "スケッチ" (Sketch), "ツール" (Tools), and "ヘルプ(H)" (Help). A toolbar with icons for file operations is visible. The central area displays the code for "STEP1.ino". The code is as follows:

```
1  /*
2   *  Arduino_Sample/STEP1/STEP1.ino
3   *
4   *  Copyright(C) 2020 RT Corporation <support@rt-net.jp>
5   *  All rights reserved.
6   *
7   *  License: Apache License, Version 2.0
8   *  https://www.apache.org/licenses/LICENSE-2.0
9   *
10  */
11
12 int LED_Pin= D13;
13
14 void setup() {
15     // put your setup code here, to run once:
16     pinMode(LED_Pin,OUTPUT);
17 }
18
19
20 void loop() {
21     // put your main code here, to run repeatedly:
22     digitalWrite(LED_Pin,HIGH); //LEDを点灯
23     delay(1000);               //1秒待つ
24     digitalWrite(LED_Pin,LOW);  //LED消灯
25     delay(1000);               //1秒待つ
26 }
27 }
```

STEP1.ino open

If the board is not "Nucleo-32", select Tools -> Board -> STM32 MCU based boards -> Nucleo-32 from the menu bar. Also make sure that the Board part number is Nucleo F303K8. The board part number can be selected from Tools -> Board part number.



Tools -> Board part number -> Select Nucleo-F303K8

Build



Click the icon in the upper left corner to run "Verify" and build the sketch.

If there are no problems, the following message will be displayed:

```
出力
最大65536バイトのフラッシュメモリのうち、スケッチが12484バイト（19%）を使っています。
最大12288バイトのRAMのうち、グローバル変数が1244バイト（10%）を使っていて、ローカル変数で11044バイト使うことができます。
```

Message when there is no problem with the validation

If there is an error in the sketch, the error will be displayed in red.

```
出力
スケッチをコンパイルしています...
"C:\Users\maaok\AppData\Local\Arduino15\packages\STMicroelectronics\tools\xpack-arm-none-eabi-gcc\12.2.1-1.2
"C:\Users\maaok\AppData\Local\Arduino15\packages\STMicroelectronics\tools\xpack-arm-none-eabi-gcc\12.2.1-1.2
C:\Users\maaok\Downloads\TrainingTracer_Samples-master\TrainingTracer_Samples-master\Arduino_Sample\STEP1\STEP1.ino: I
C:\Users\maaok\Downloads\TrainingTracer_Samples-master\TrainingTracer_Samples-master\Arduino_Sample\STEP1\STEP1.ino:16
| 16 | pinMode(LED1_Pin,OUTPUT);
|     | ^~~~~~
|     | LED_Pin
|
次のフォルダのライブラリSrcWrapperバージョン1.0.1を使用中:C:\Users\maaok\AppData\Local\Arduino15\packages\STMicroelectron
exit status 1

Compilation error: 'LED1_Pin' was not declared in this scope; did you mean 'LED_Pin'?
```

Validation error message



You can hide the message by clicking the icon at the bottom right of the Arduino IDE. Click it again to show it again.

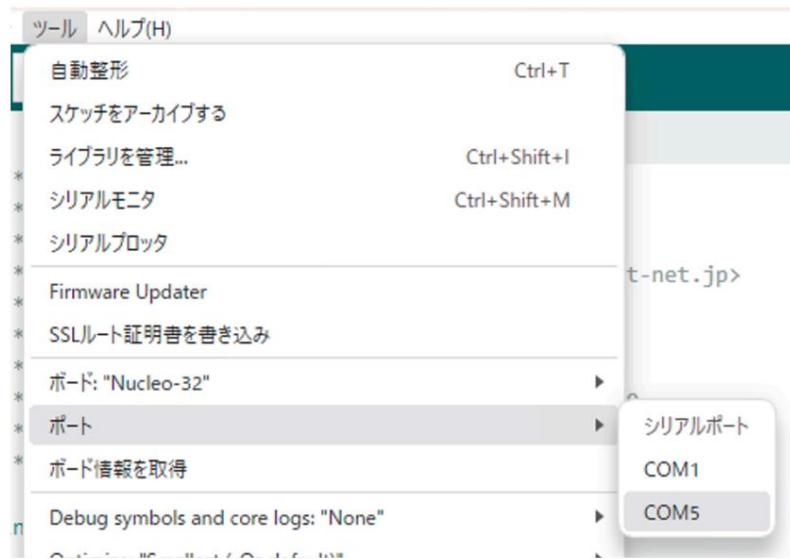
Use this function when you want to enlarge the screen to see more of the entire sketch, or when you do not need this message screen.

write

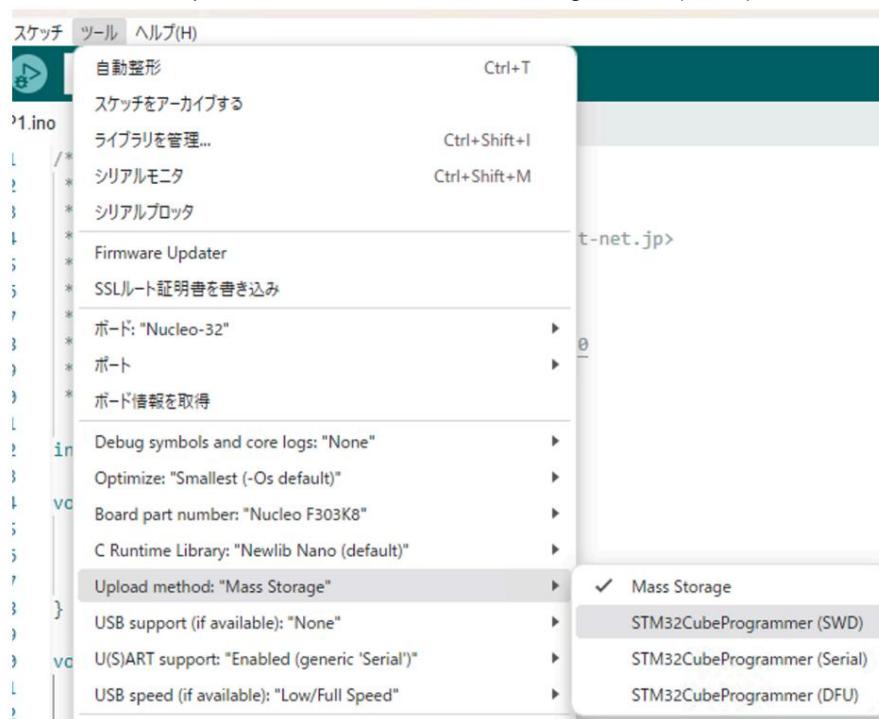
Uses COM port and SWD for writing.

The COM port can be set in Tools -> Ports. The COM number can be found in Device Manager.

For Windows, it is COMX. For Linux (Ubuntu), it is /dev/ttUSBX. For macOS, it is It is called usbmodemXXXX Serial Port(USB).

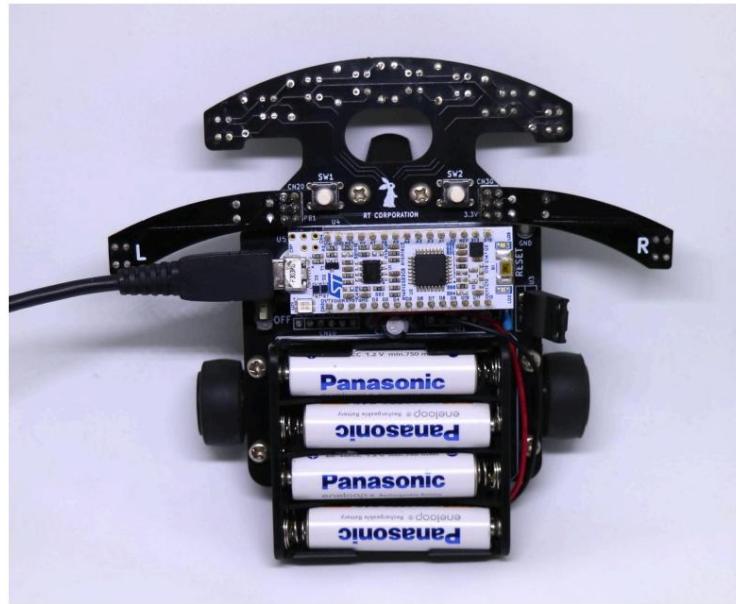


For SWD, select Tools->Upload method->STM32CubeProgrammer(SWD).



Once you have confirmed that there are no errors in the sketch, prepare for writing by following the steps below.

1. Check that the port number and upload method are correct.
2. Check that a dry cell battery is inserted in the device.
3. Connect the PC and this product with a USB cable.
4. **Raise the tire. (The left motor will rotate when writing.)**
5. Slide the power switch to the ON side.



With the USB cable connected

If it is set correctly, a USB mark will be added to the left of the board name. In addition to writing sketches, port settings are also required when communicating with a PC via hardware serial communication.



After setting the port, click to write the sketch.

Writing is now complete.

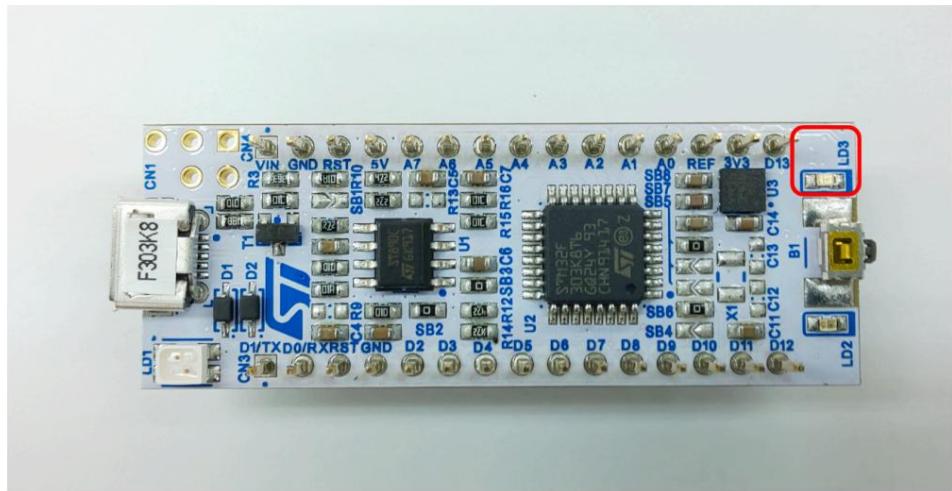
```
出力

File download complete
Time elapsed during download operation: 00:00:00.808

RUNNING Program ...
| Address:      : 0x8000000
Application is running, Please Hold on...
Start operation achieved successfully
|
```

Screen display when writing is complete

The LED on the Nucleo board will blink for 0.5 seconds and then turn off for 0.5 seconds.



How to deal with errors when writing

If the ST-Link firmware is old, the following message will appear.

If this occurs, please update the STlink firmware.

```
STM32CubeProgrammer v2.3.0

Error: Old ST-LINK firmware version. Upgrade ST-LINK firmware
Error: Old ST-LINK firmware version. Upgrade ST-LINK firmware
Error: Old ST-LINK firmware!Please upgrade it.
```

When the message "Error: No debug probe detected" appears, the PC and Nucleo board are connected with a USB cable. It is not connected properly. Reconnect the USB cable and try writing again.

```
STM32CubeProgrammer v2.16.0

Error: No debug probe detected.
Failed uploading: uploading error: exit status 1
```

If the message "Error: No STM32 target found" appears, the power to this product is not turned on. Please slide it to the ON side and then try writing again.

```
ST-LINK SN : 0667FF303032424257015845
ST-LINK FW : V2J44M29
Board      : NUCLEO-F303K8
Voltage    : 3.27V
Error: No STM32 target found! If your product embeds Debug Auth...
Failed uploading: uploading error: exit status 1
```

ST-Link Firmware Update

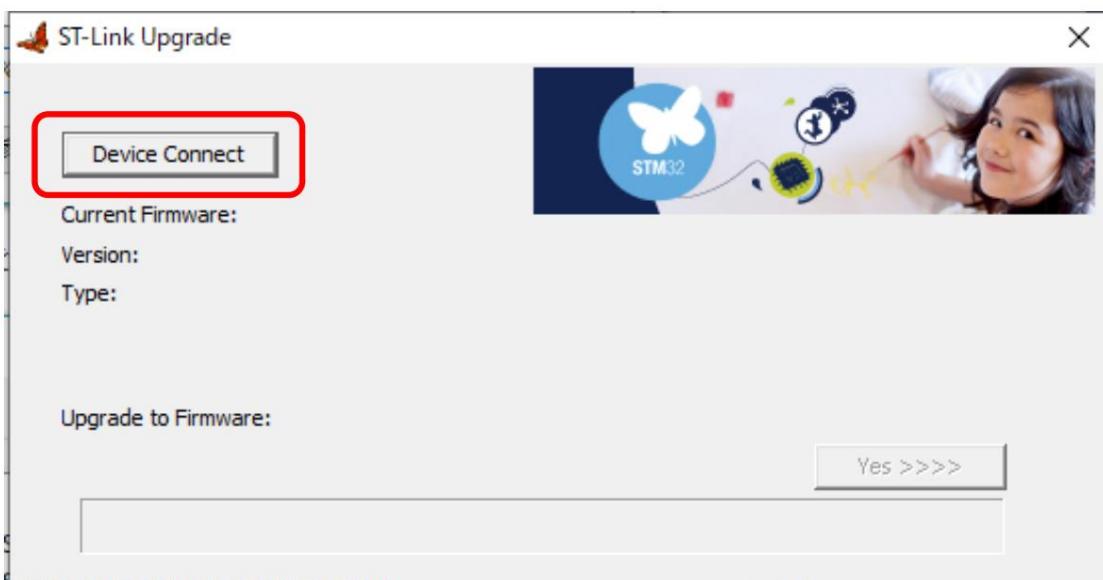
Download the ST-Link firmware from the following page.

<https://www.st.com/ja/development-tools/stsw-link007.html>

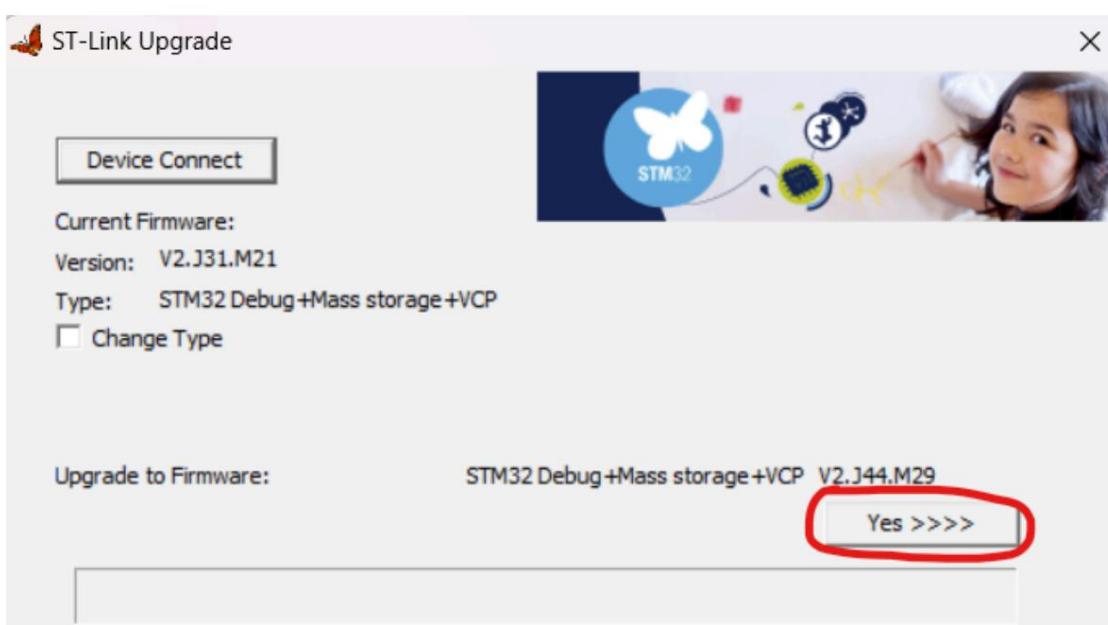
How to update the firmware on Windows

Extract the downloaded file and double-click ST-LinkUpgrade.exe in the stsw-link007/windows folder to launch it.

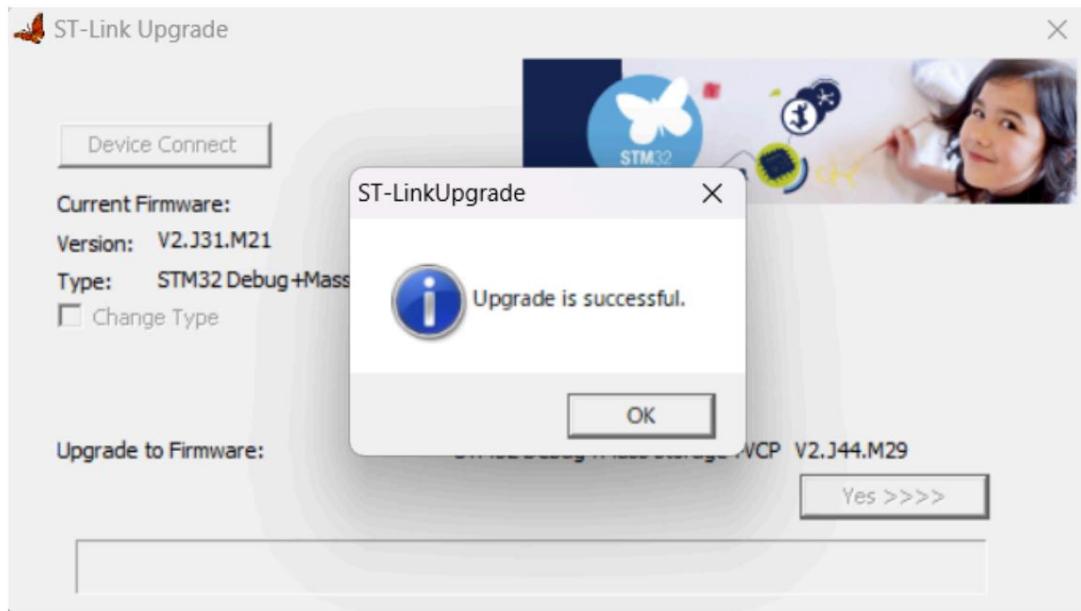
 ST-LinkUpgrade	2024/04/03 10:27	アプリケーション	1,253 KB
 STLinkUSBDriver.dll	2024/04/03 10:27	アプリケーション拡張	115 KB



Connect the Nucleo board to the PC via USB, slide the power switch, and then select "Device Connect." Click



When the Version and Type are displayed, click "Yes>>>".

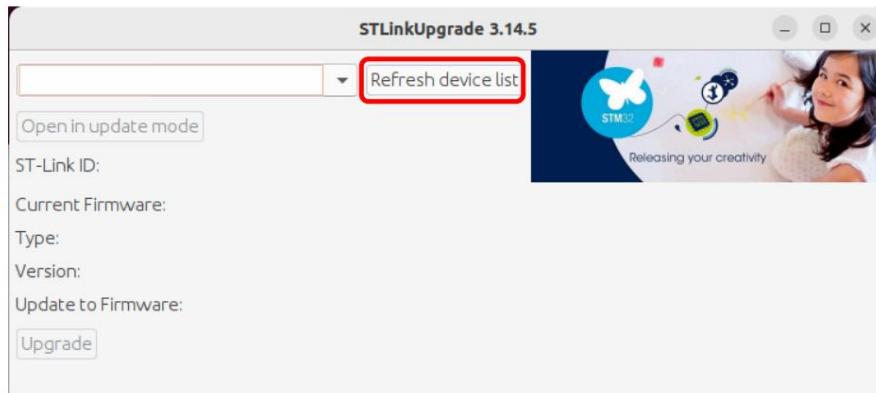


When the message "Upgrade is successful" appears, close this screen and resume writing.

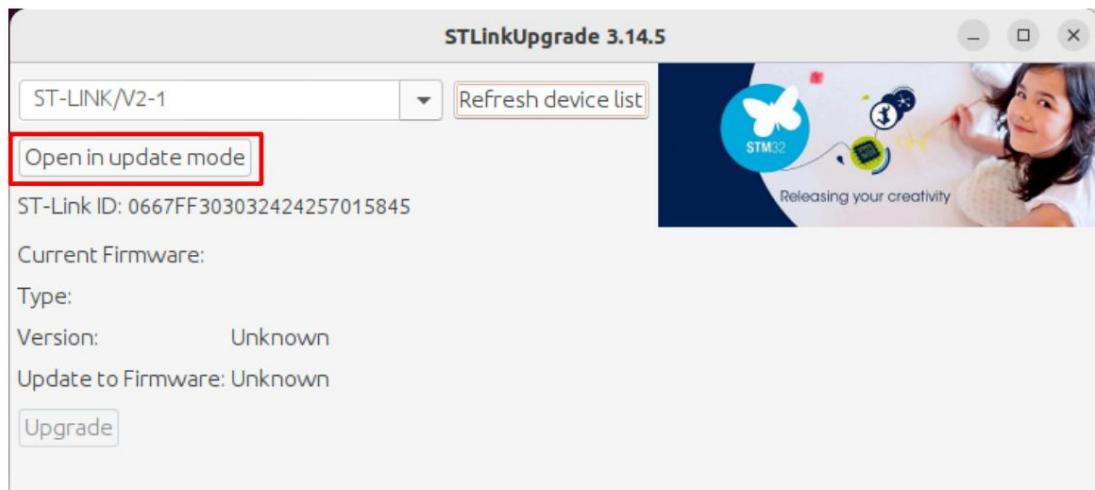
How to update the firmware on Linux (Ubuntu)

Double-click the downloaded enstsw-link007-v3-15-6.zip to unzip it. Terminal Launch and navigate to the extracted swst-link007/AllPlatforms.

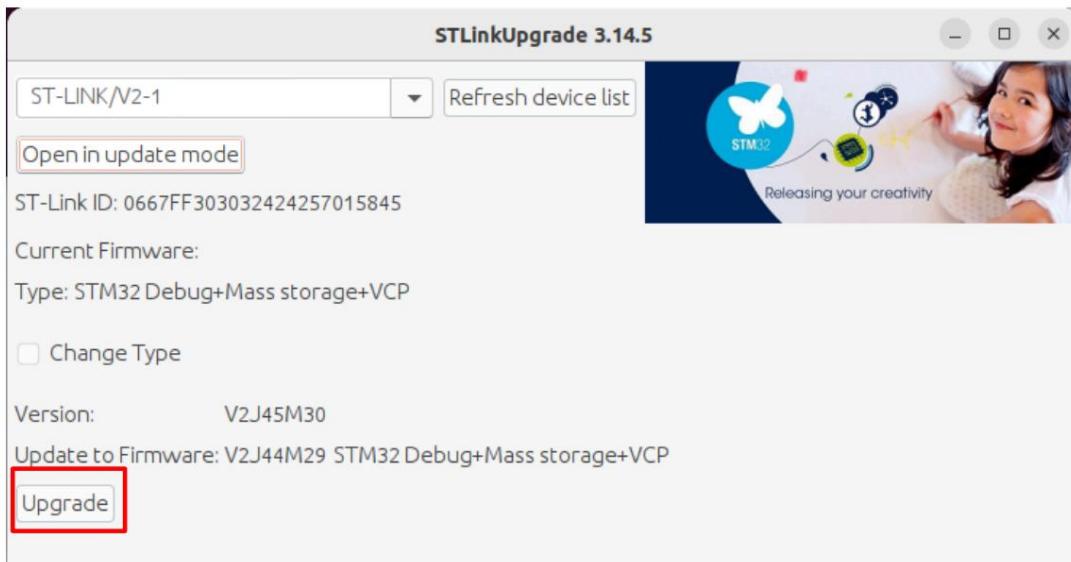
```
java -jar STLinkUpgrade.jar
```



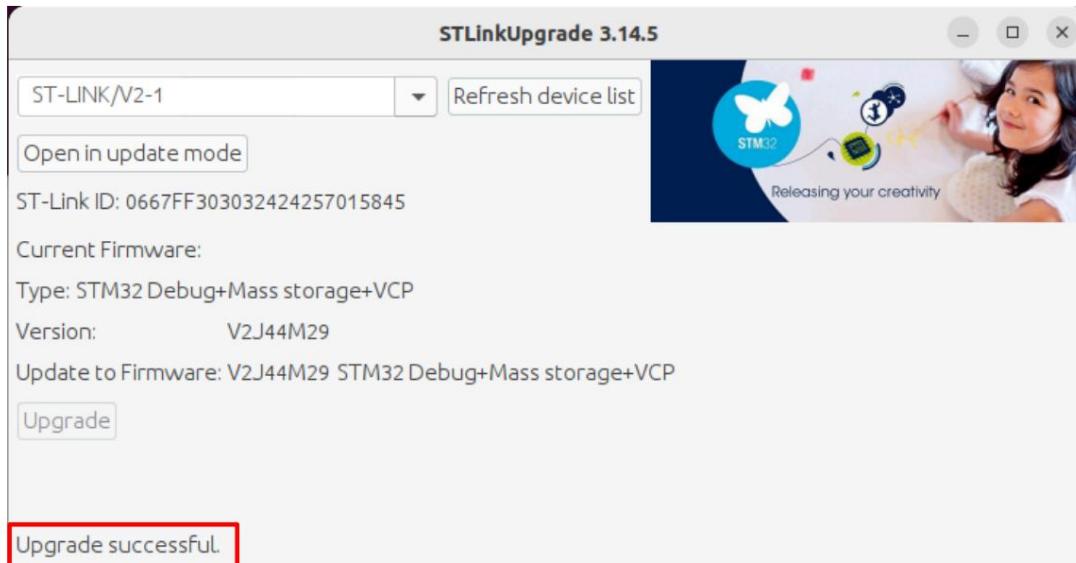
Connect the Nucleo board to the PC via USB, slide the power switch, and then press "Refresh Device Click "list".



Click "Open in update mode".



When the Version and Type are displayed, click "Upgrade".



When you see "Upgrade is successful", close this screen and resume writing.

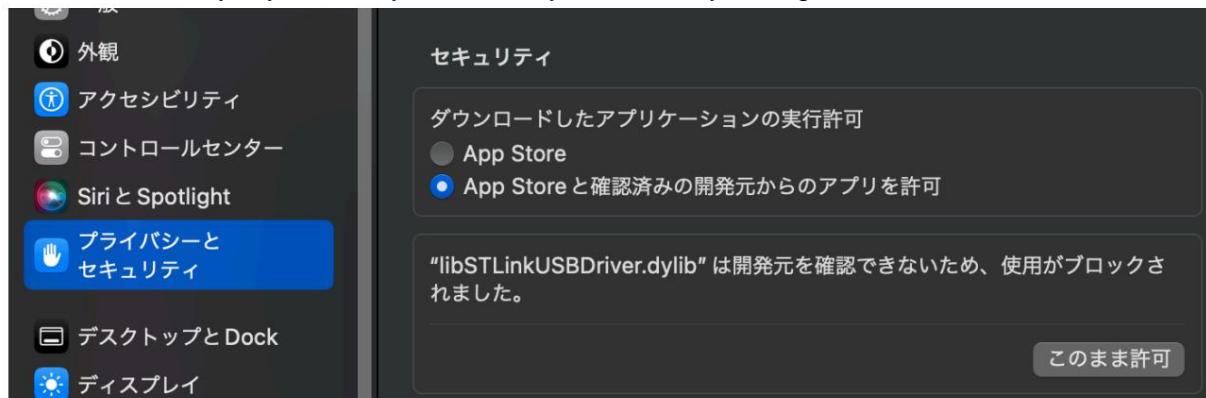
How to update firmware on macOS Double-click the downloaded enstsw-link007-v3-15-6.zip to unzip it. Terminal Launch and navigate to the extracted swst-link007/AllPlatforms.

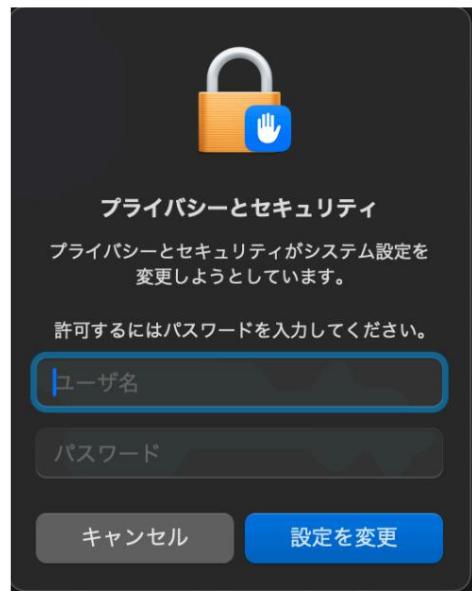
```
java -jar STLinkUpgrade.jar
```

If you see this message, press Cancel.



Click "Allow anyway" in the System Privacy and Security settings.





Enter your PC's user name and password to change the settings. Return to the terminal again, Execute the java command.

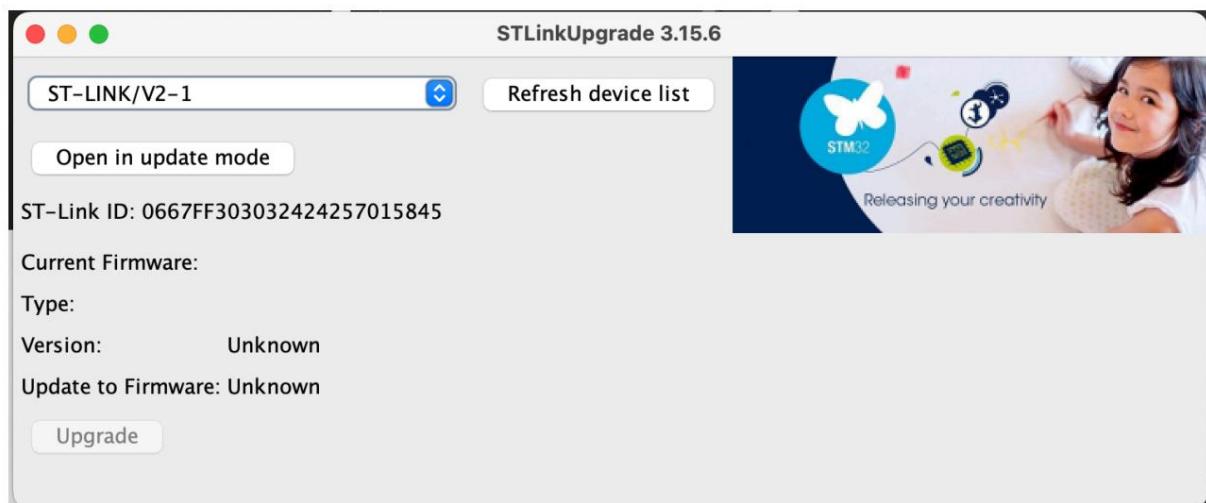


You'll see a similar message to the one before, but this time click "Open".

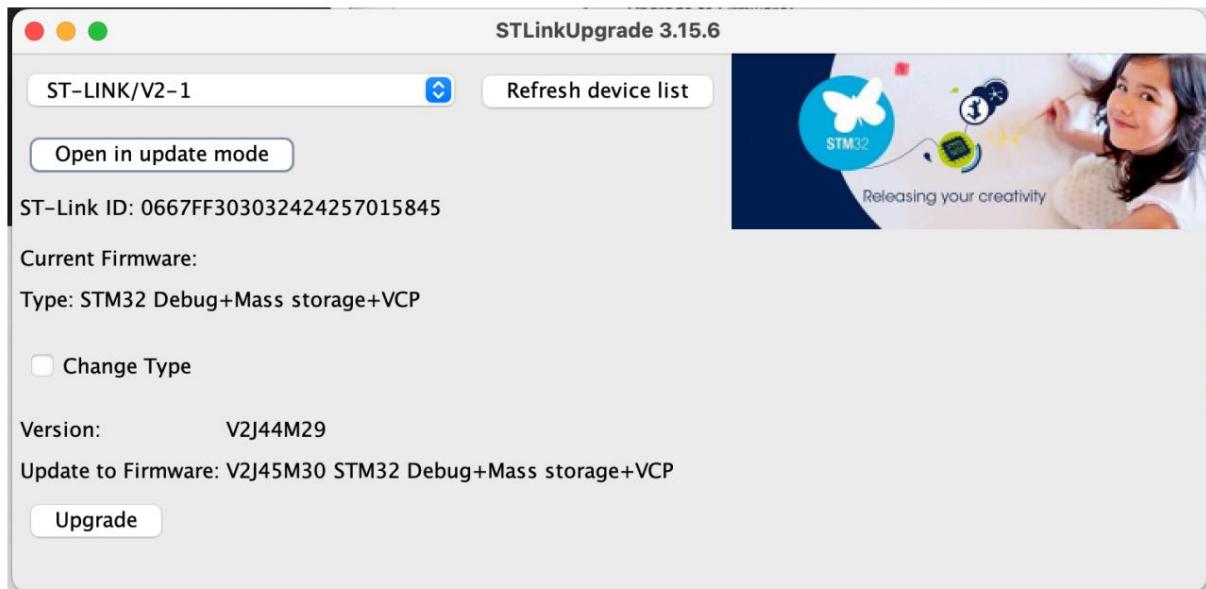


Cancel again and select "Allow anyway" in the System Privacy and Security settings.

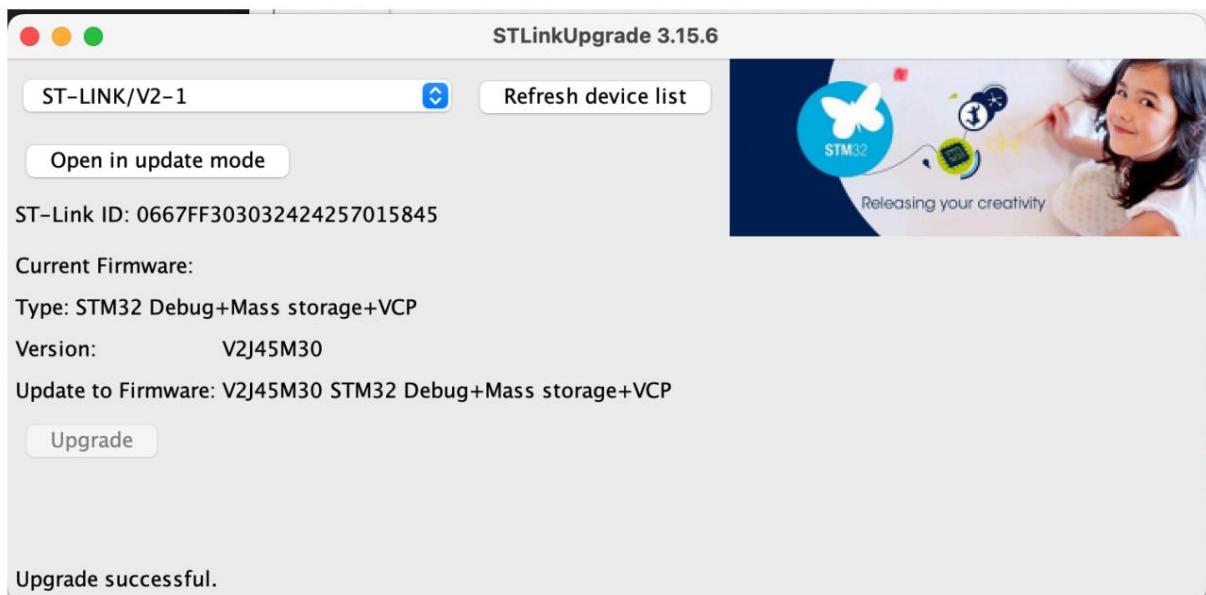
Click on the link below. Return to the terminal and run the java command.



Connect the Nucleo board to the PC via USB, slide the power switch, and then click "Open in update mode." If you cannot click "Open in update mode," click "Refresh device list" to be able to click it.



When the Version and Type are displayed, click "Upgrade".



When you see "Upgrade is successful", close this screen and resume writing.

Tools used, OSS versions

The tools and OSS versions used in this book are as follows.

Version information of tools and OSS used in this book

Tools and OSS	Version	URL
Arduino IDE	2.3.2	https://www.arduino.cc/en/software
Arduino core for the ESP32 3.0.1		https://github.com/espressif/arduino-esp32/releases
Arduino core support for STM32 based boards	2.7.1	https://github.com/stm32duino/ArduinoCore_STM32/releases
STM32CubePrograms	2.16.0	https://www.st.com/ja/development-tools/stm32cubeprog.html
STLink boards firmware upgrade	3.15.6	https://www.st.com/ja/development-tools/stsw-link007.html
STLink USB driver(windows only)	2.0.2	https://www.st.com/ja/development-tools/stsw-link009.html

Revision History

Issue date (YY/MM/DD)	Edition	Revision details
24/06/12	1.0	New Creation
24/09/20	1.1	Updated the URL of the GitHub repository for Training Tracer Ver.2

Copyright and Intellectual Property Rights

We reserve all rights to patents, utility model rights, design rights, copyrights, know-how, and other technologies and intellectual property rights related to this product and the source files, directories, executable files, data, development tools, and other materials (hereinafter referred to as "our materials") created by us in connection with this product. This document does not grant permission to use our trademarks, trade names, service marks, product names, and logos. However, this does not apply to cases where such trademarks, etc. are used to the extent reasonably necessary for the explanation or description of this product and our materials. Please do not remove any product identification numbers, trademarks, registered trademarks, copyrights, or other notices attached to this product or our materials.

All the company and product names in this document are trademarks or registered trademarks of their respective companies.

All the documents, photos, and illustrations are copyrighted and protected by the copyright law of Japan and overseas.

All the contents in this document are not allowed to be uploaded to any public or local area networks such as the Internet without permission from RT Corporation.