

# アールティ製品共通 Arduino開発環境構築マニュアル

1.1版  
株式会社アールティ

# 目次

目次	1
ご使用になる前に	2
コンピュータの動作要件	2
環境設定	3
コマンドの表記について	3
はじめに	3
Arduino IDEのインストール	3
Windowsの場合	6
Linux (Ubuntu) の場合	8
macOSの場合	10
Arduino IDEの初期設定	11
言語設定	11
外観設定	13
サンプルスケッチのビルドと書き込み	14
Pi:Co Classic3およびESP32-S3マイコンボード	14
ESP32ボード情報の追加	14
サンプルスケッチの準備	17
ビルド	22
書き込み	23
Pi:Co V2	26
ESP32ボード情報の追加	26
サンプルスケッチの準備	29
ビルド	33
書き込み	34
トレーニングトレーサー	37
STM32ボード情報の追加	37
書き込みツールSTM32CubeProgrammerのインストール	40
サンプルスケッチの準備	51
ビルド	55
書き込み	56
書き込み時のエラー対処方法	58
ST-Linkファームウェアのアップデート	59
使用しているツール、OSSのバージョン	67
改訂履歴	68
Copyright・知的財産権について	68

## ご使用になる前に

この度は、弊社の製品（以下「本製品」といいます）をお買い上げいただき、誠にありがとうございます。本製品をご使用になる前に、製品に付属する**入門ガイド**をお読みいただきますようお願いいたします。

## コンピュータの動作要件

本製品を使用するに当たり、Arduino IDEの動作要件を満たしたPCをご用意ください。Arduino IDEの動作要件は<https://www.arduino.cc/en/software>をご参照ください。

## 環境設定

ここでは本製品の開発環境の構築手順について説明します。

### コマンドの表記について

本書では、Linuxのターミナルで実行するコマンドを下記のように表記します。"\$"の後に続く文字列がコマンドを表します("\$"を除く)。"#"で始まる行はコメントを表すため実行不要です。

```
$ echo "コマンド"  
# この行はコメントです
```

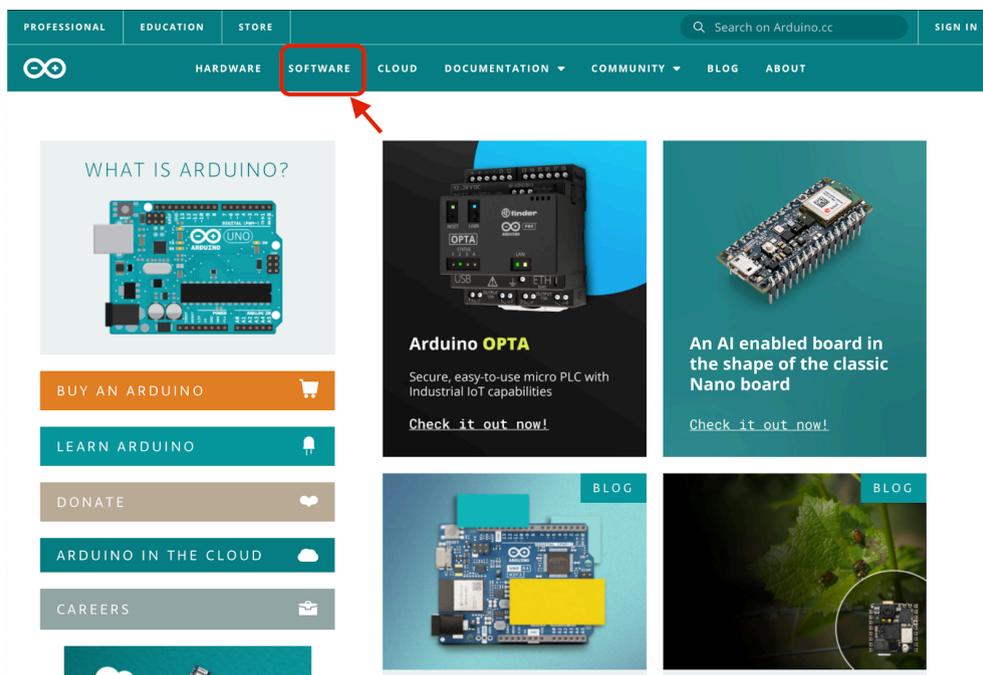
### はじめに

本製品はArduino Foundationが提供するArduino IDEを開発環境として使用します。Arduino IDEはマルチプラットフォームに対応しています。そのため、本製品は3種のOS（Windows、Linux、macOS）で開発できます。

### Arduino IDEのインストール

2024年6月時点で最新版のArduino IDE 2.3.2をインストールする手順を示します。Arduino IDEが更新された場合は、本書内のバージョン番号を適宜読み替えてください。

webブラウザで<https://www.arduino.cc>にアクセスしSOFTWAREのタブをクリックします。



<https://www.arduino.cc> のトップページ

使用するOSに合うソフトウェアをDOWNLOAD OPTIONSからクリックします。本書では、Windows MSI installer、Linux Appliance 64 bits (X86-64)およびmacOS Intel, 10.15: “Catalina” or newer, 64 bitsを選択した前提でインストール手順を説明します。

Arduino IDE 選択ページ

以下のDownload Arduino IDE & support tis progressのページが表示された場合は、寄付をしてソフトウェアをダウンロードする”CONTRIBUTE & DOWNLOAD”と、ダウンロードのみを実施する”JUST DOWNLOAD”のどちらかを選択します。

Arduino IDE ダウンロード方法選択ページ

以下のStay in the loop:Join Our Newsletterのページが表示された場合、購読を希望する方はemailの記入とチェックボックスの両方にチェックを入れ、SUBSCRIBE & DOWNLOADをクリックします。購読を希望しない方は、JUST DOWNLOADをクリックします。

**Stay in the Loop: Join Our Newsletter!**

As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

I would like to receive emails about special deals and commercial offers from Arduino.

**SUBSCRIBE & DOWNLOAD**

or

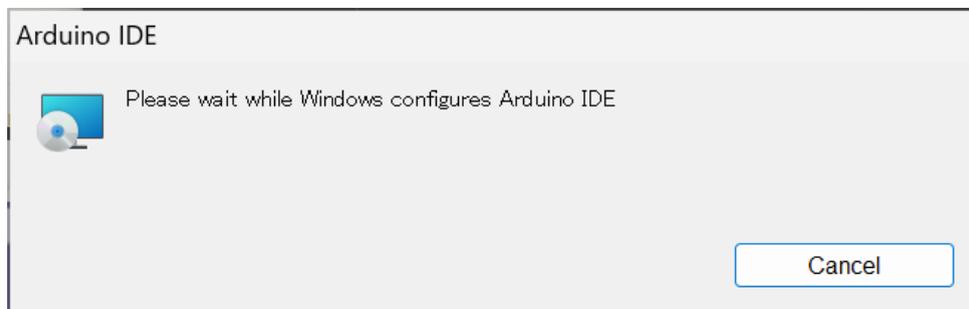
**JUST DOWNLOAD**



Newsletter購読確認ページ

## Windowsの場合

ダウンロードしたMSI installer（例：arduino-ide\_2.3.2\_Windows\_64bit\_msi）をダブルクリックしてインストールします。インストールが終わると下記の画面が消えます。



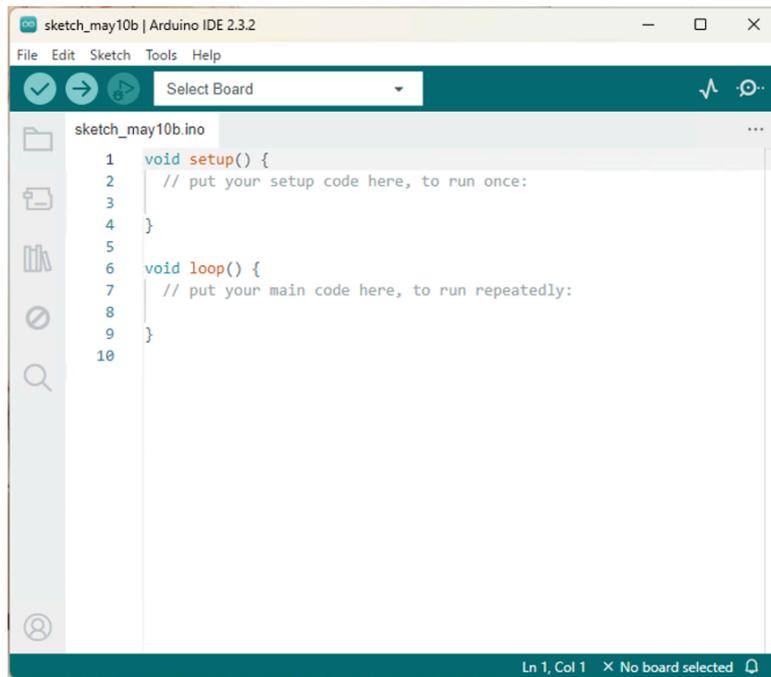
インストール中の画面

Arduino IDEが正常にインストールできたことを確認するためArduino IDEを起動します。Windowsのスタートボタン->検索ボックスに"arduino IDE"というキーワードを入力するとArduino IDEが検索結果として表示されます。



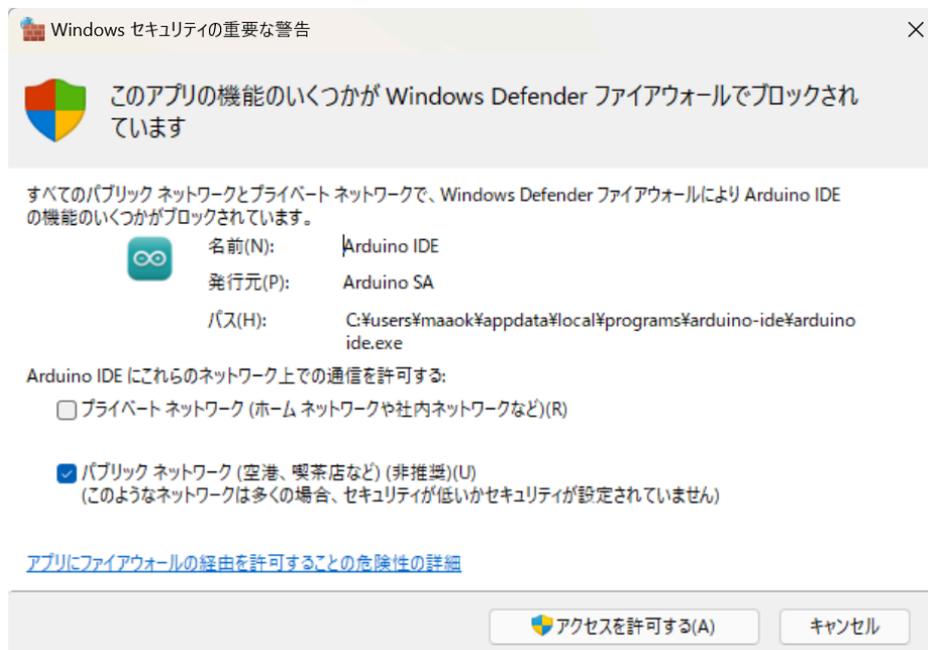
検索ボックスに"arduino IDE"を入力した状態

表示されたArduino IDEをクリックし、Arduino IDEが起動すると以下のような画面が表示されます。以上でインストール完了です。



起動したArduino IDE

Windowsセキュリティの重要な警告メッセージが表示されたときは、アクセスを許可してください。アクセスをブロックしてしまうとArduino IDEのバージョンアップや追加ライブラリのダウンロードができなくなります。



Windowsセキュリティの重要な警告メッセージ

## Linux (Ubuntu) の場合

ダウンロードしたファイル（例：arduino-ide\_2.3.2\_Linux\_64bit.AppImage）を右クリックし、メニューからプロパティを選択します。プロパティ内のアクセス権タブを開き、“プログラムとして実行可能”にチェックを入れます。

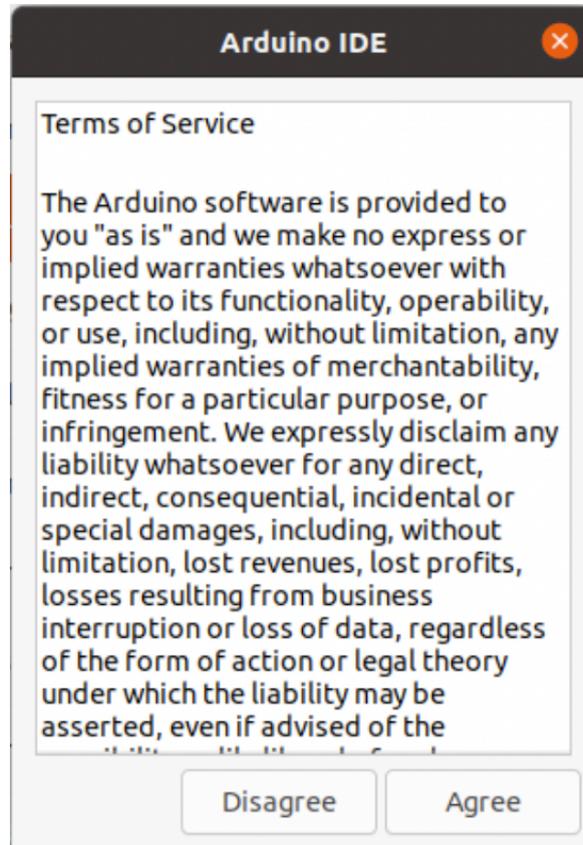


AppImageファイルを右クリックした状態



プロパティのアクセス権タブを開いた状態

その後、ApplImageファイルをダブルクリックします。Terms of Serviceが表示された場合は、中身を確認し”Disagree/Agree” を選択してください。以上でインストール完了です。



Terms of Service画面

ApplImageファイルをダブルクリックしてもArduino IDEが実行されない場合は、ApplImageファイルのコンテンツを展開し実行するツール（FUSE）がインストールされていない可能性があります。<https://github.com/ApplImage/ApplImageKit/wiki/FUSE> 参照し、FUSEをインストールしてください。

## macOSの場合

ダウンロードしたファイル（例：arduino-ide\_2.3.2\_macOS\_64bit.dmg）をダブルクリックすると以下の画面が表示されます。画面左にあるArduino IDEのアイコンを右のフォルダに移動するとインストールが完了します。インストール後のArduino IDEはLaunchpadから起動できます。



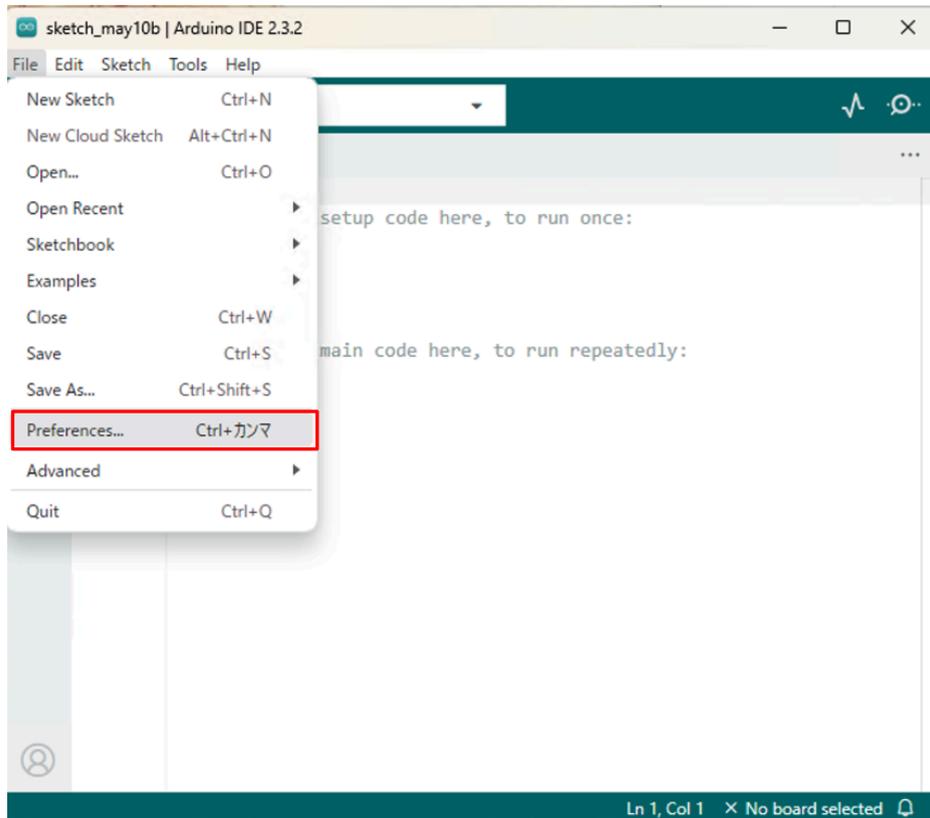
dmg ファイルを開いた状態

## Arduino IDEの初期設定

ここではArduino IDEの言語や外観を設定します。また、本製品に搭載されているESP32-S3用のスケッチファイルをビルドするために、ライブラリやコンパイラを追加します。

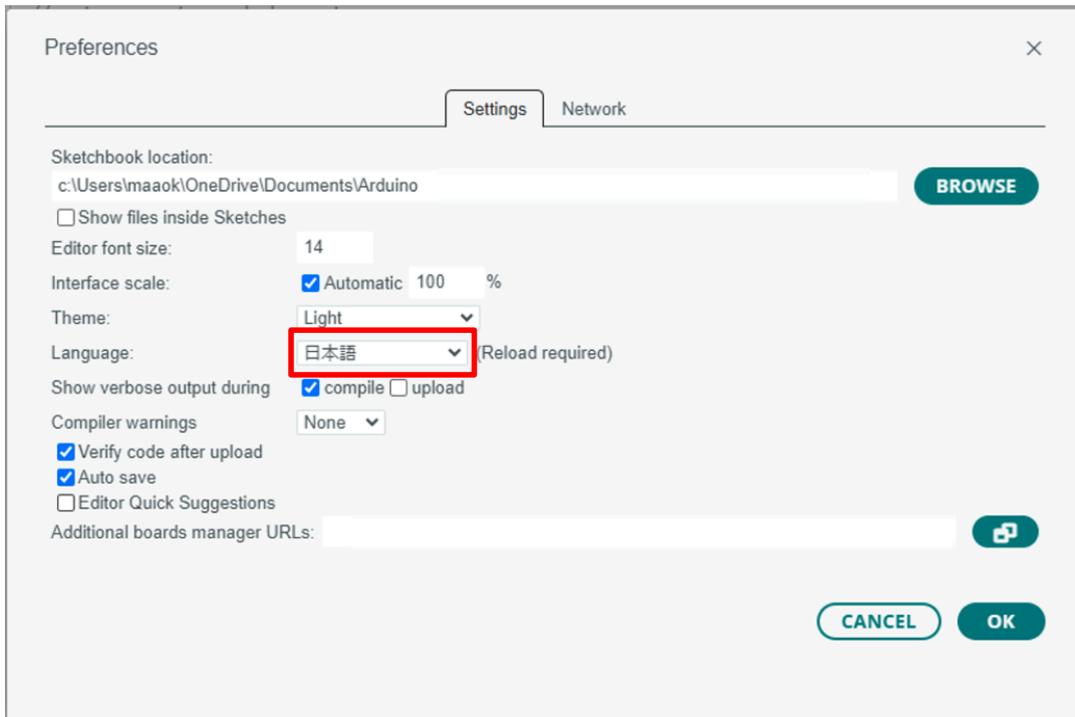
### 言語設定

Arduino IDEの言語を日本語に設定します。メニューバーにあるFile->Preferences...を選択します。



メニューバーのFile->Preferences...を選択した状態

Languageを”English”から”日本語”に変更して右下にある”OK”をクリックします。



Languageを日本語に設定した状態

”OK”をクリックしたら自動的にArduino IDEが再起動します。再起動後、メニューバーの左端にあるボタンが”File”から”ファイル”になっていれば言語設定完了です。

## 外観設定

OSの設定によっては、Arduino IDEの外観が暗い色調で表示されます。コントラストの弱いPC画面では明るい色調に設定するとスケッチ編集画面のカーソルをはっきり表示できる場合があります。下記手順に従って好みの外観を設定してください。

メニューバーのファイル->基本設定...を選択します。"配色テーマ"を"Dark"から"Light"に変更し、"OK"をクリックします。これによりArduino IDEの外観が明るい色調に変更されます。以降のページで登場するArduino IDE画面は"Light"テーマを設定したものです。



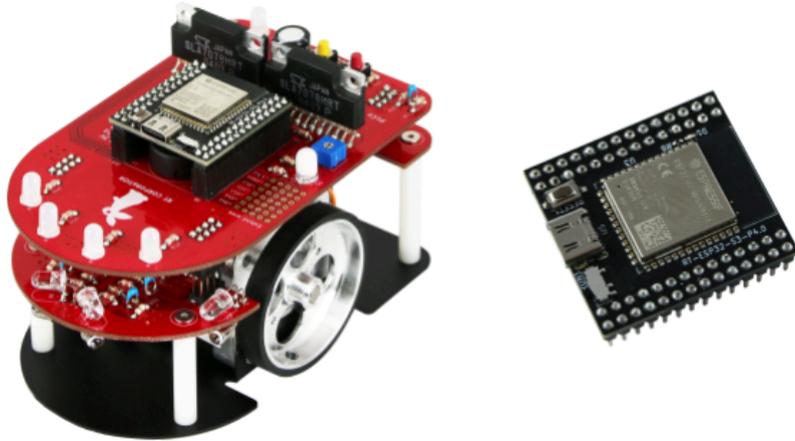
配色テーマでLightを選択した状態

## サンプルスケッチのビルドと書き込み

ここでは各製品ごとにボードの追加、サンプルスケッチのダウンロード、ビルド、書き込み手順を示します。

### Pi:Co Classic3およびESP32-S3マイコンボード

---

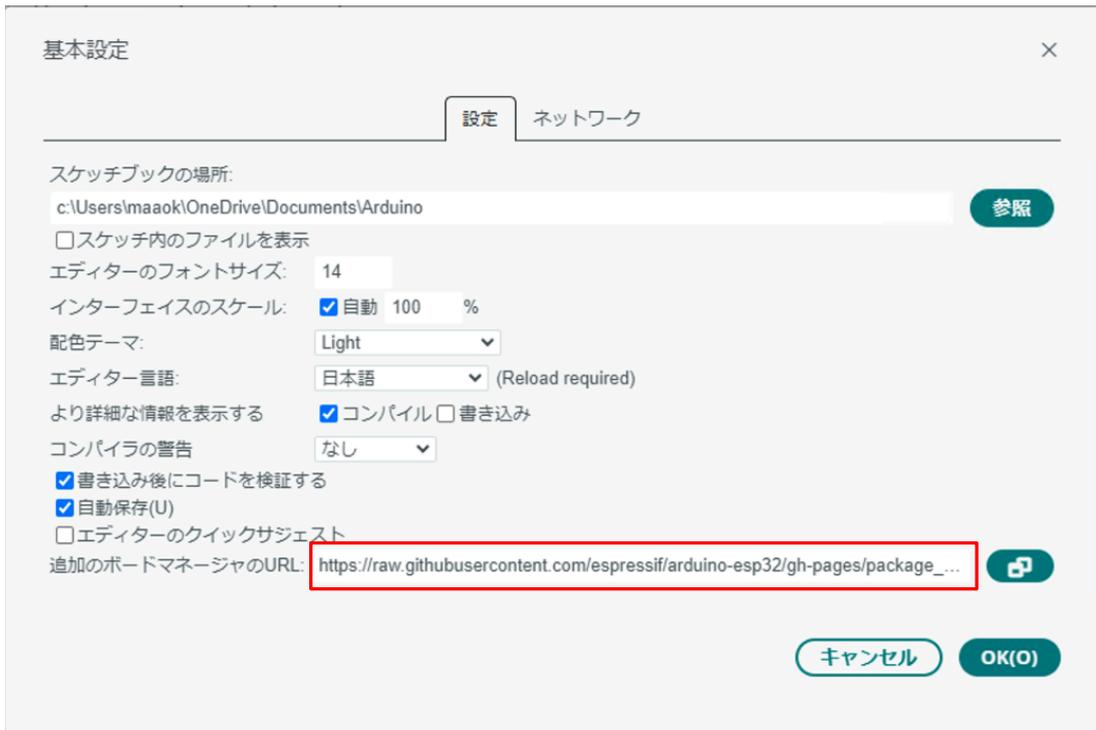


#### ESP32ボード情報の追加

スケッチを本製品に書き込めるように、ESP32のボード情報をArduino IDEに追加します。メニューバーのファイル->基本設定...を選択します。画面下部の”追加のボードマネージャのURL”に

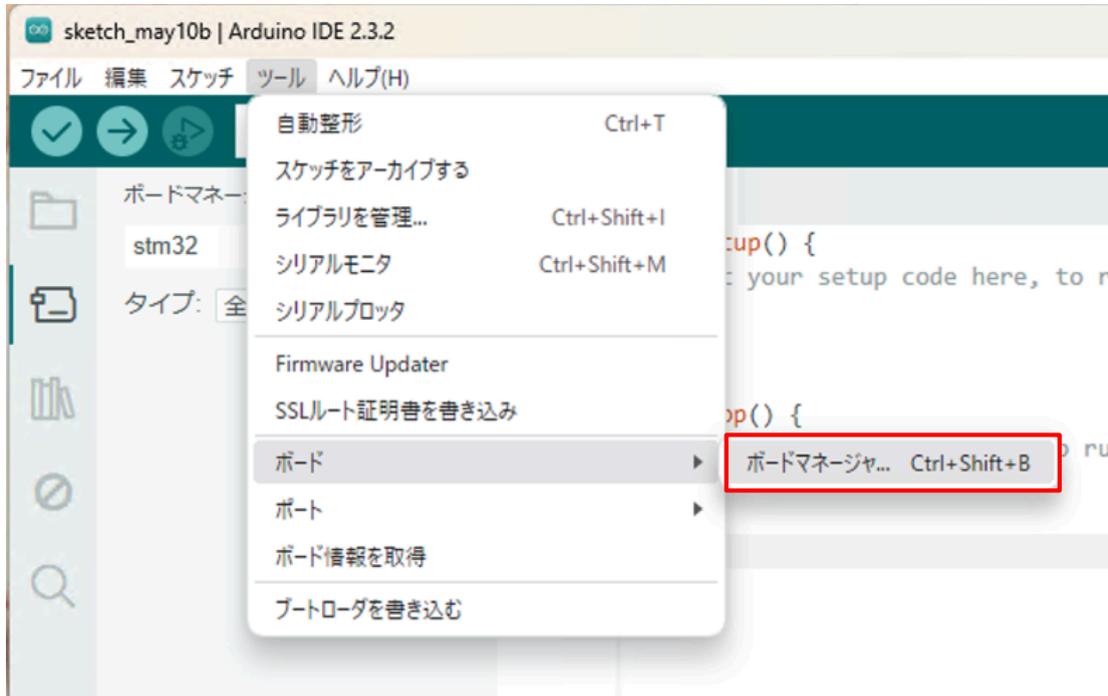
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

を追加し”OK”をクリックします。



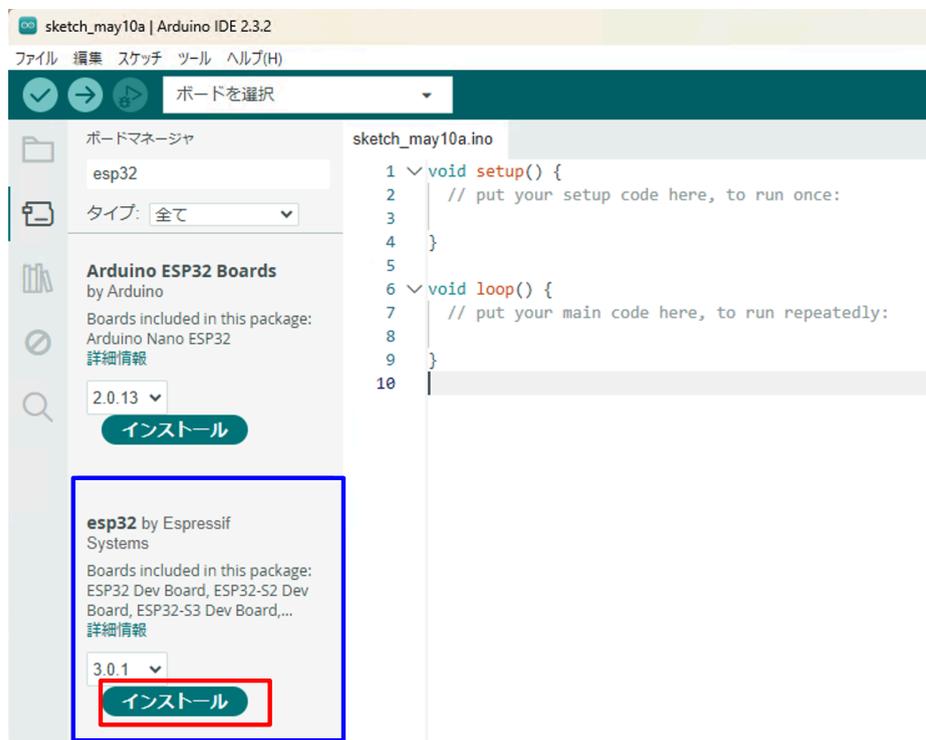
追加ボードマネージャのURLに追加した状態

URLを追加したら、ESP32ライブラリやツールをダウンロードしてインストールします。メニューバーのツール->ボード->ボードマネージャ...を選択します。



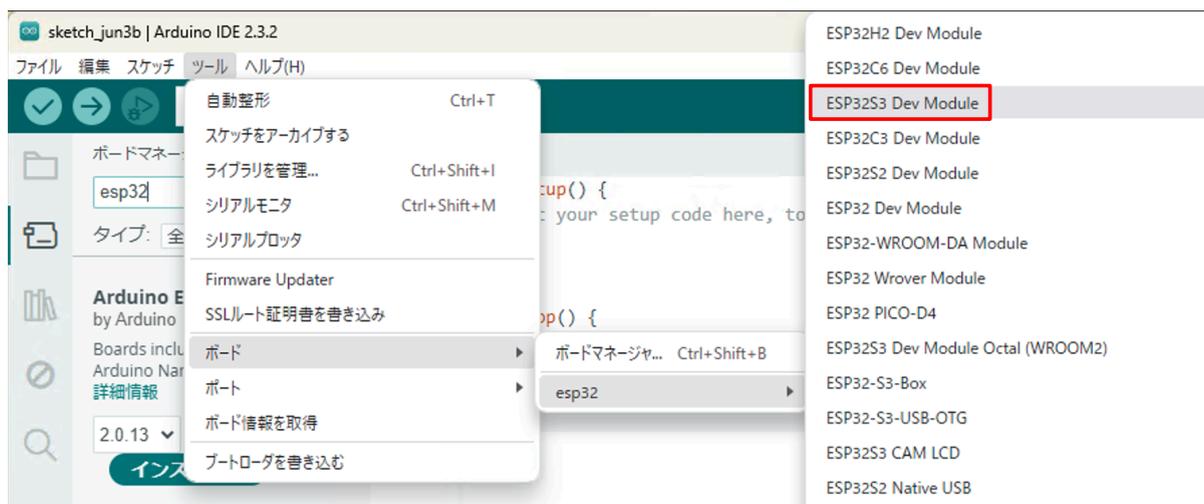
ボードマネージャ...を選択した状態

左側にあるボードマネージャの検索欄に”esp32”と入力すると”esp32 by Espressif Systems”が表示されます。2024年6月時点では3.0.1が最新です。esp32 by Espressif Systemのインストールをクリックします。



ボードマネージャ画面の検索欄に”esp32”を入力した状態

Arduino IDEはESP32以外のマイコンにも対応しているため、スケッチを書き込む際はESP32のボードを選択します。メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択します。”ボード”と”ポート”が同じような文字で上下にあるため間違えないように選択してください。



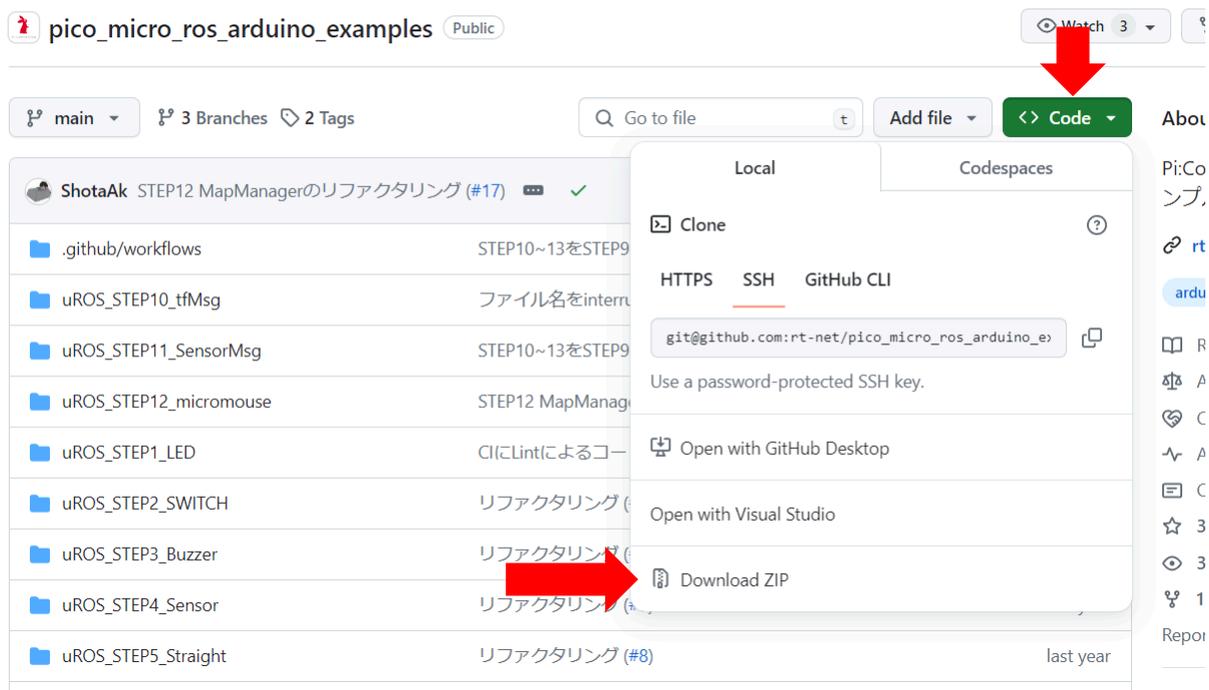
ツール->ボード->esp32->ESP32S3 Dev Modulesを選択

以上で、ESP32-S3のビルド環境設定完了です。

## サンプルスケッチの準備

webブラウザで[https://github.com/rt-net/pico\\_micro\\_ros\\_arduino\\_examples](https://github.com/rt-net/pico_micro_ros_arduino_examples)にアクセスし、本製品のサンプルスケッチファイルをダウンロードします。

ページ内の”Code”から”Download ZIP”をクリックすることでダウンロードできます。ダウンロードしたzipファイルは展開してください。



Code->Download ZIPを選択してサンプルスケッチをダウンロード

ここではPi Co Classic3のLEDを点灯させるサンプルスケッチ（uROS\_STEP1\_LED）を例にして、スケッチのビルドと書き込み方法を説明します。

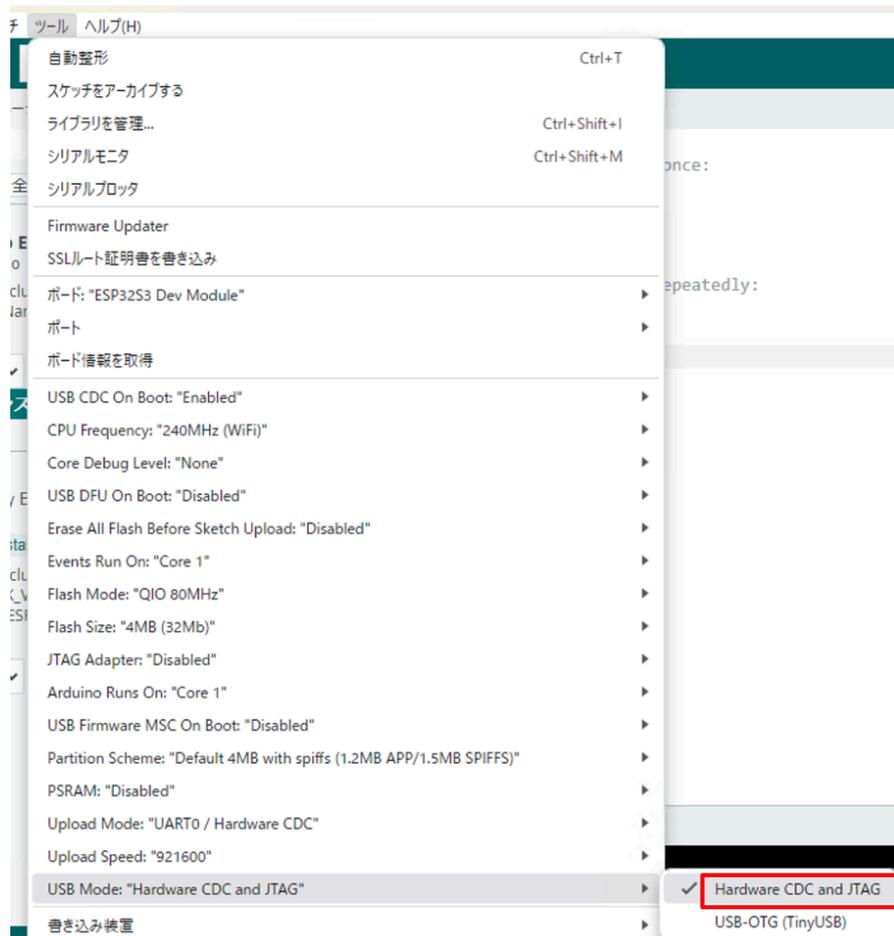
本製品にはUSB経由でスケッチを書き込みます。デフォルトではUSBからの書き込みができないため、書き込みができるように次の2箇所変更します。

まず、メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択します。次に、メニューバーのツール->USB CDC On Bootを開き、“Enabled”にします。



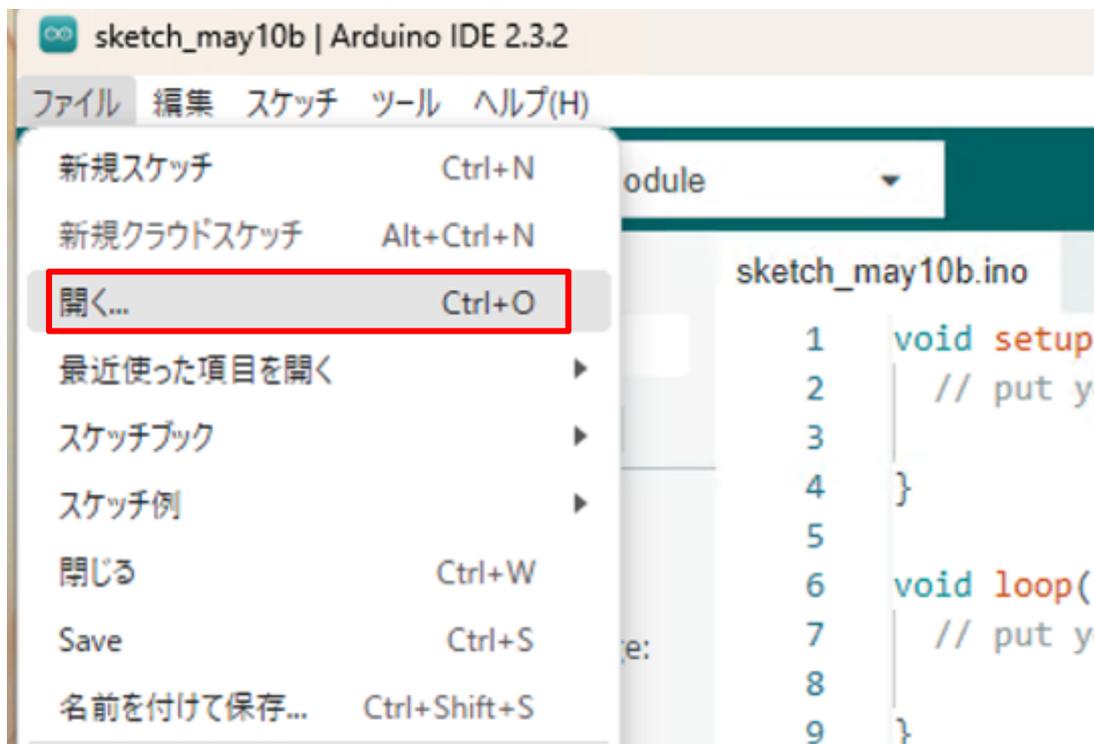
USB CDC On BootをEnabledに設定

メニューバーのツール->USB Modeを開き、“Hardware CDC and JTAG”にします。



USB ModeをHardware CDC and JTAGに設定

ダウンロードしたサンプルスケッチ内のuROS\_STEP1\_LED/uROS\_STEP1\_LED.inoファイルをダブルクリックで開きます。または、Arduino IDEを起動し、メニューバーのファイル->開く...をクリックしてuROS\_STEP1\_LED.inoを開きます。



ファイル->開く...を選択する

スケッチファイルを開く時に以下のメッセージが出た場合は、スケッチファイル名とフォルダ名が一致していない、またはフォルダが無いということを示しています。新しくフォルダを作成して良い場合はOKを押してください。

Arduino IDEの仕様上、メインとなる"ファイル名"と"フォルダ名"が同じではないときにこのメッセージが表示されます。ファイル名やフォルダ名を変更する場合はそれぞれ同じ名前にしてください。



ファイル名とフォルダ名が一致しない場合の確認画面

サンプルスケッチを開くと以下のような画面が表示されます。



```
uROS_STEP1_LED | Arduino IDE 2.0.3
ESP32S3 Dev Module

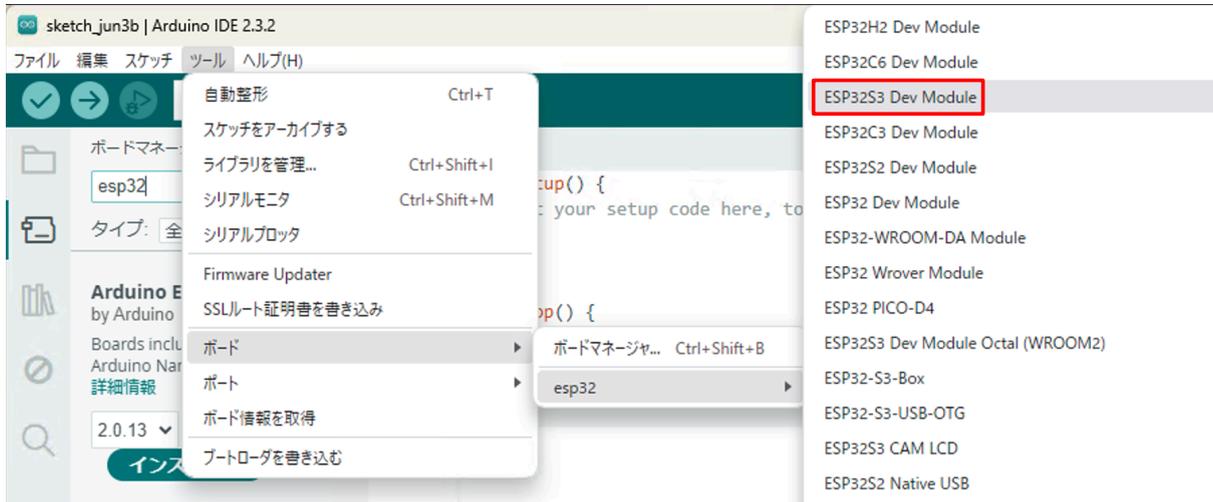
uROS_STEP1_LED.ino
1 // Copyright 2022 RT Corporation
2 //
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the License at
6 //
7 // http://www.apache.org/licenses/LICENSE-2.0
8 //
9 // Unless required by applicable law or agreed to in writing, software
10 // distributed under the License is distributed on an "AS IS" BASIS,
11 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 // See the License for the specific language governing permissions and
13 // limitations under the License.
14
15 #define LED0 1
16 #define LED1 2
17 #define LED2 42
18 #define LED3 41
19
20
21 void setup() {
22 // put your setup code here, to run once:
23 pinMode(LED0,OUTPUT);
24 pinMode(LED1,OUTPUT);
25 pinMode(LED2,OUTPUT);
26 pinMode(LED3,OUTPUT);
27
28 }
29
30 void loop() {
31 // put your main code here, to run repeatedly:
32 digitalWrite(LED0,HIGH);
33 digitalWrite(LED1,HIGH);
34 digitalWrite(LED2,HIGH);
35 digitalWrite(LED3,HIGH);
36 delay(500);
37 digitalWrite(LED0,LOW);
38 digitalWrite(LED1,LOW);
39 digitalWrite(LED2,LOW);
40 digitalWrite(LED3,LOW);
41 delay(500);
42 }
43
```

出力

行 32、列 25 UTF-8 ESP32S3 Dev Module [未接続]

uROS\_STEP1\_LED.inoを開いた状態

ボードが“ESP32S3 Dev Module”になっていないときは、メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択してください。



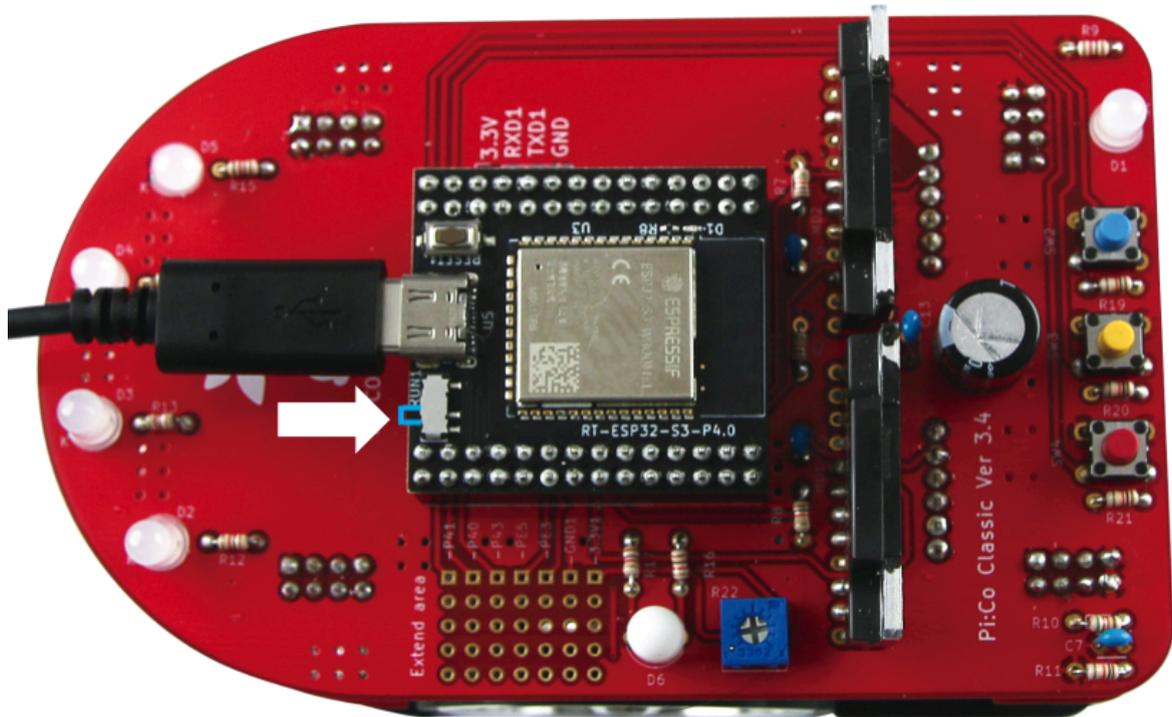
ツール->ボード->esp32->ESP32S3 Dev Moduleを選択



## 書き込み

スケッチにエラーが無いことを確認したら、次の手順で書き込み前の準備をします。

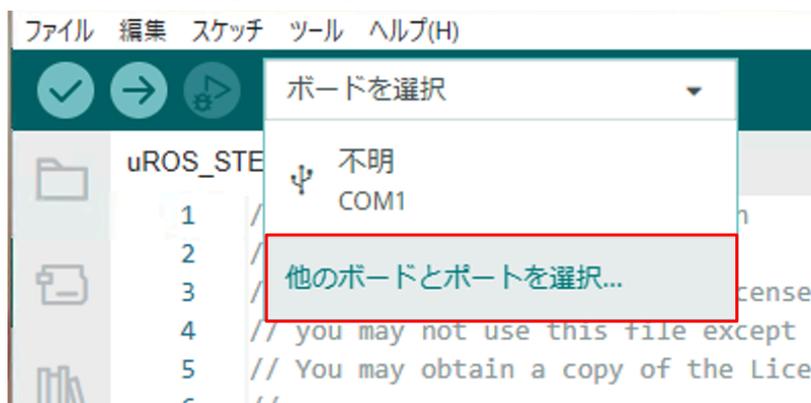
1. 本製品の電源スイッチをOFFにします。
2. 本製品にバッテリーを接続します。
3. 本製品の書き込み端子とPCをUSBケーブルで接続します。
4. 本製品のモード切替スイッチをRUN1とは逆側の方向にスライドします。
5. 本製品の電源スイッチをONにします。
6. 本製品のリセットボタンを押します。



モード切替スイッチを操作し、USBケーブルを接続した状態

本製品をPCと接続した後、以下の手順でボードとポートを設定します。本製品の電源をOFFにした後や、リセットボタンを押した後にも再度設定してください。

メニューバーの下に表示されているボード名（ESP32S3 Dev Module）をクリックし、一番下の”他のボードとポートを選択...”を開きます。



ボード名->他のボードとポートを選択...を開く

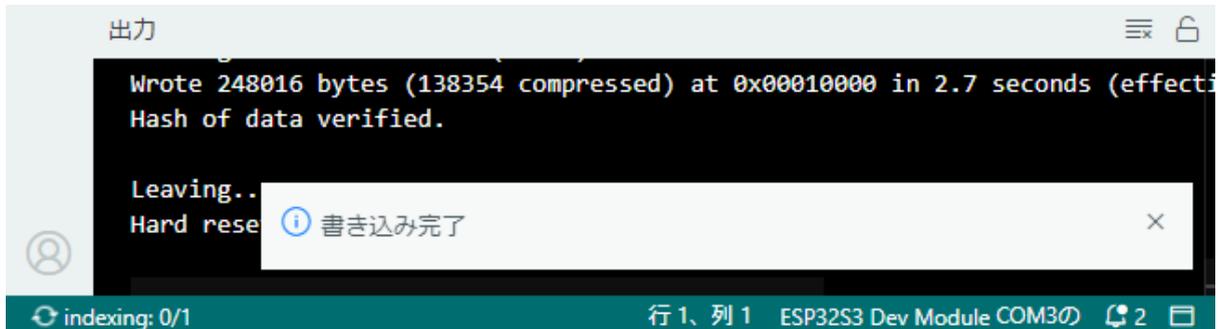
ボードの検索欄に”dev”を入力し、”ESP32S3 Dev Module”にチェックが入っていることを確認します。チェックが入っていない場合は、ESP32S3 Dev Moduleをクリックします。ポート選択欄では本製品を接続したUSBポートを選択します。Windowsの場合はCOMX、Linux(Ubuntu)の場合は/dev/ttyUSBX、macOSの場合はusbmodemXXXXX Serial Port(USB)という名称です。選択したら”OK”をクリックします。



ボードとポートの設定

設定完了後、ポート名の左にUSBのマーク  ESP32S3 Dev Module が追加されます。スケッチの書き込み以外に、PCとハードウェアシリアル通信をするときも、ポート設定が必要です。

ポート設定後、 をクリックしてスケッチを書き込みます。下記のように表示されたら書き込み完了です。



書き込み完了時の画面表示

Linux(Ubuntu) PCで書き込みに失敗する時は、書き込み前にターミナルで次のコマンドを実行し、USBポートの使用権限を変更します。

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

永続的にUSBポートの使用権限を変更する場合は、次のコマンドを実行し、PCを再起動します。

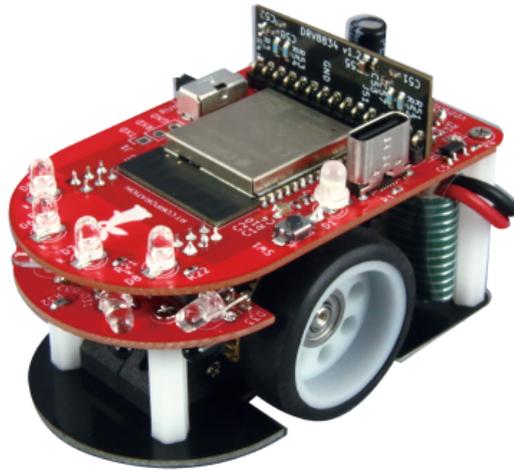
```
# 下記を実行した後にPCを再起動すること
$ sudo usermod -aG dialout $USER
```

スケッチの書き込み完了後、本製品のモード切替スイッチをRUN1側にスライドし、リセットボタンを押します。

本製品の前方のLEDが0.5秒点滅、0.5秒消灯を繰り返します。

## Pi:Co V2

---

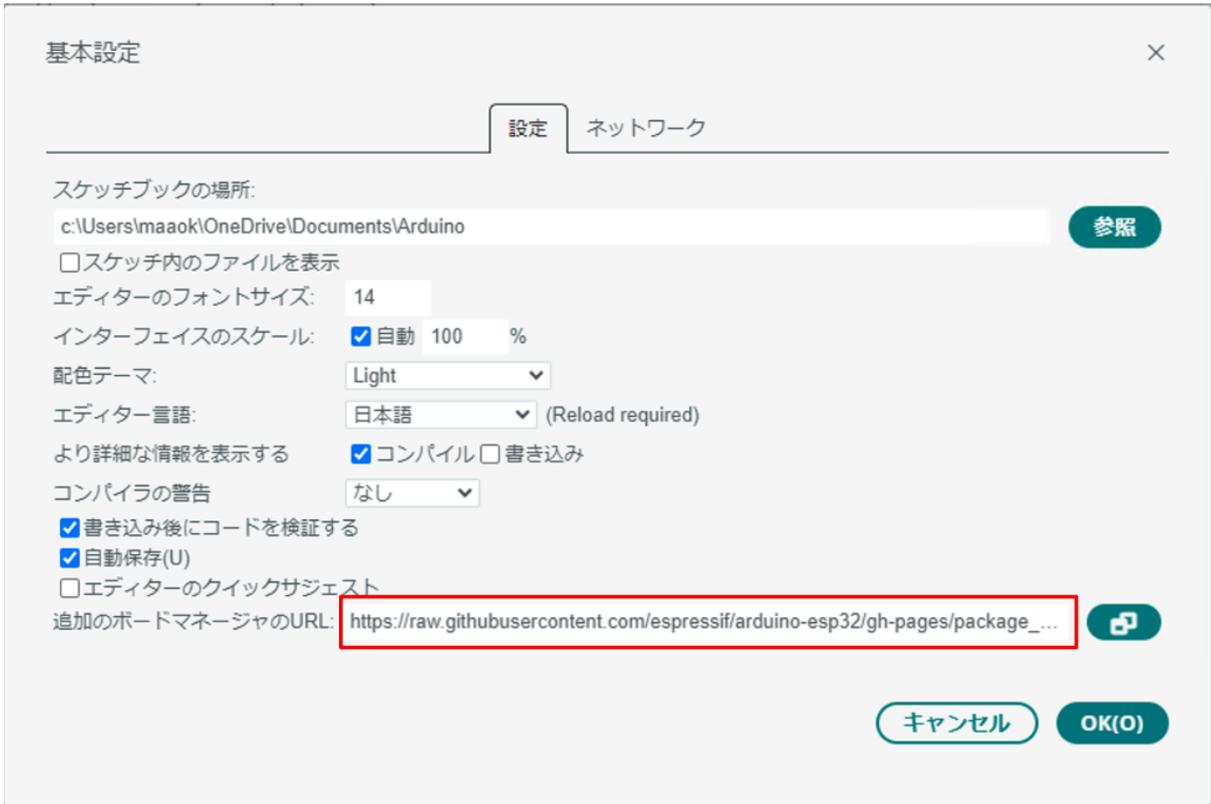


### ESP32ボード情報の追加

スケッチを本製品に書き込めるように、ESP32のボード情報をArduino IDEに追加します。メニューバーのファイル->基本設定...を選択します。画面下部の”追加のボードマネージャのURL”に

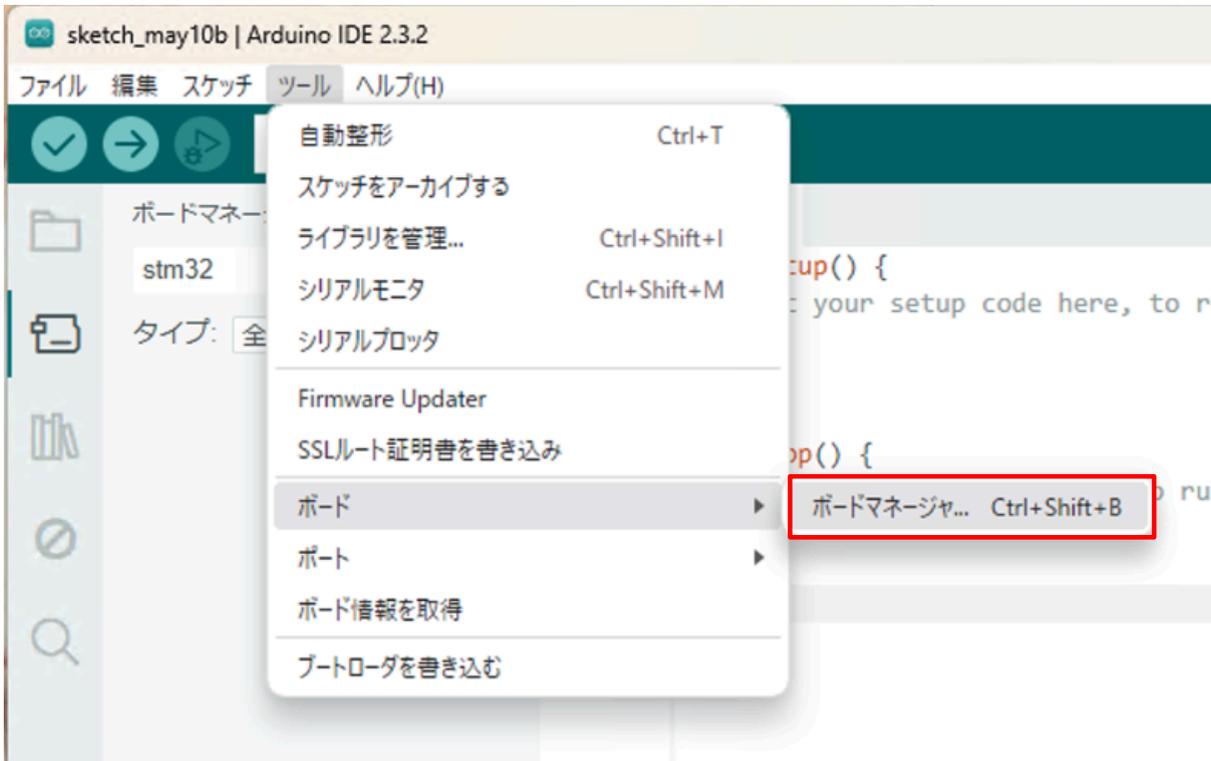
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

を追加し”OK”をクリックします。



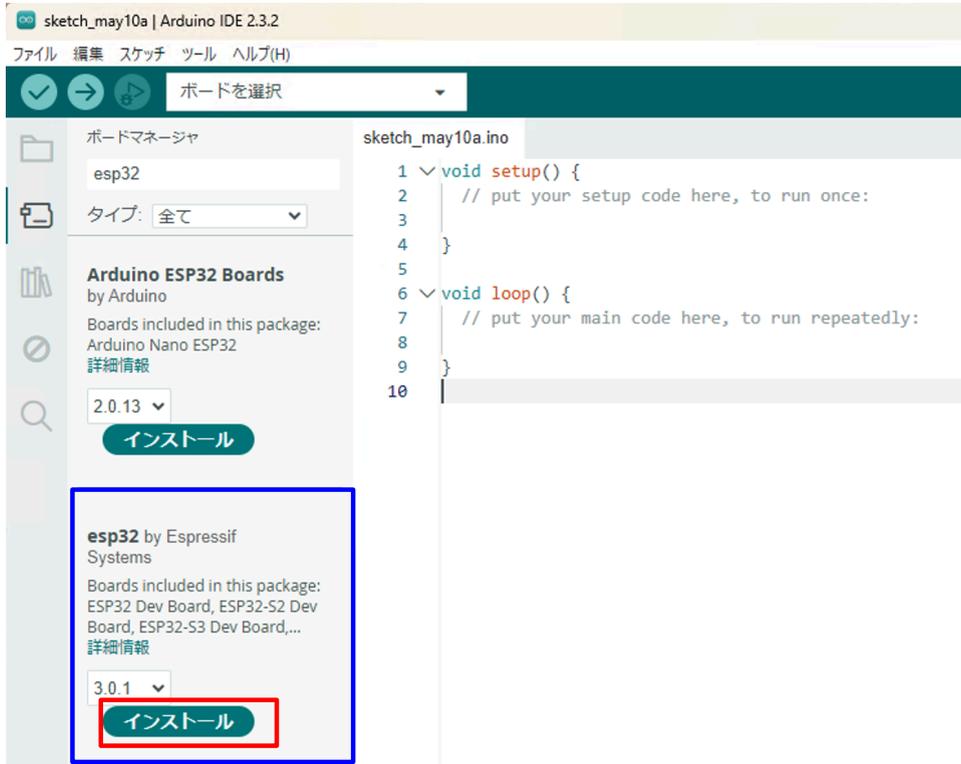
追加ボードマネージャのURLに追加した状態

URLを追加したら、ESP32ライブラリやツールをダウンロードしてインストールします。メニューバーのツール->ボード->ボードマネージャ...を選択します。



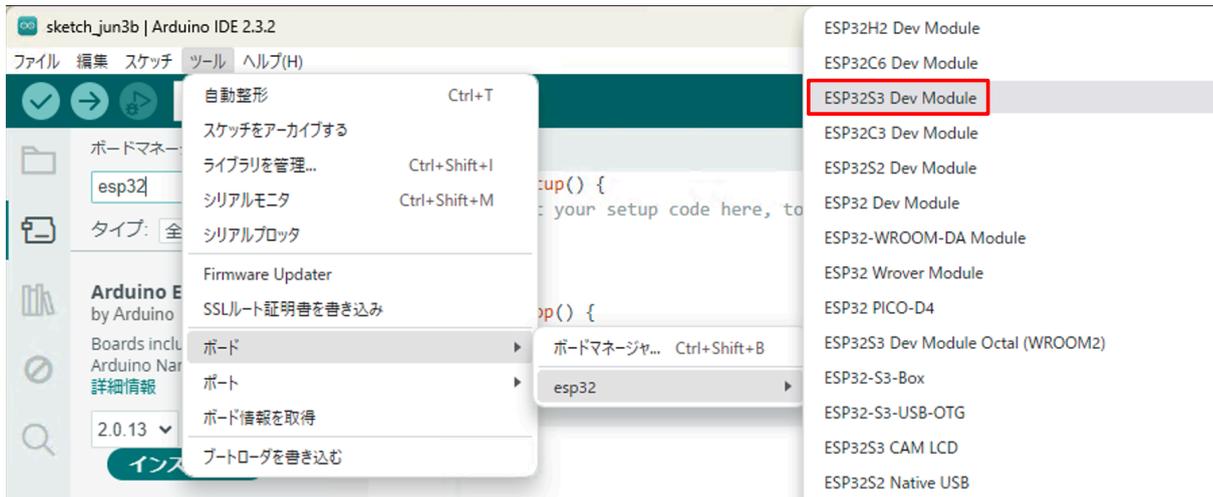
ボードマネージャ...を選択した状態

左側にあるボードマネージャの検索欄に"esp32"と入力すると"esp32 by Espressif Systems"が表示されます。2024年6月時点では3.0.1が最新です。esp32 by Espressif Systemのインストールをクリックします。



ボードマネージャ画面の検索欄に"esp32"を入力した状態

Arduino IDEはESP32以外のマイコンにも対応しているため、スケッチを書き込む際はESP32のボードを選択します。メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択します。"ボード"と"ポート"が同じような文字で上下にあるため間違えないように選択してください。



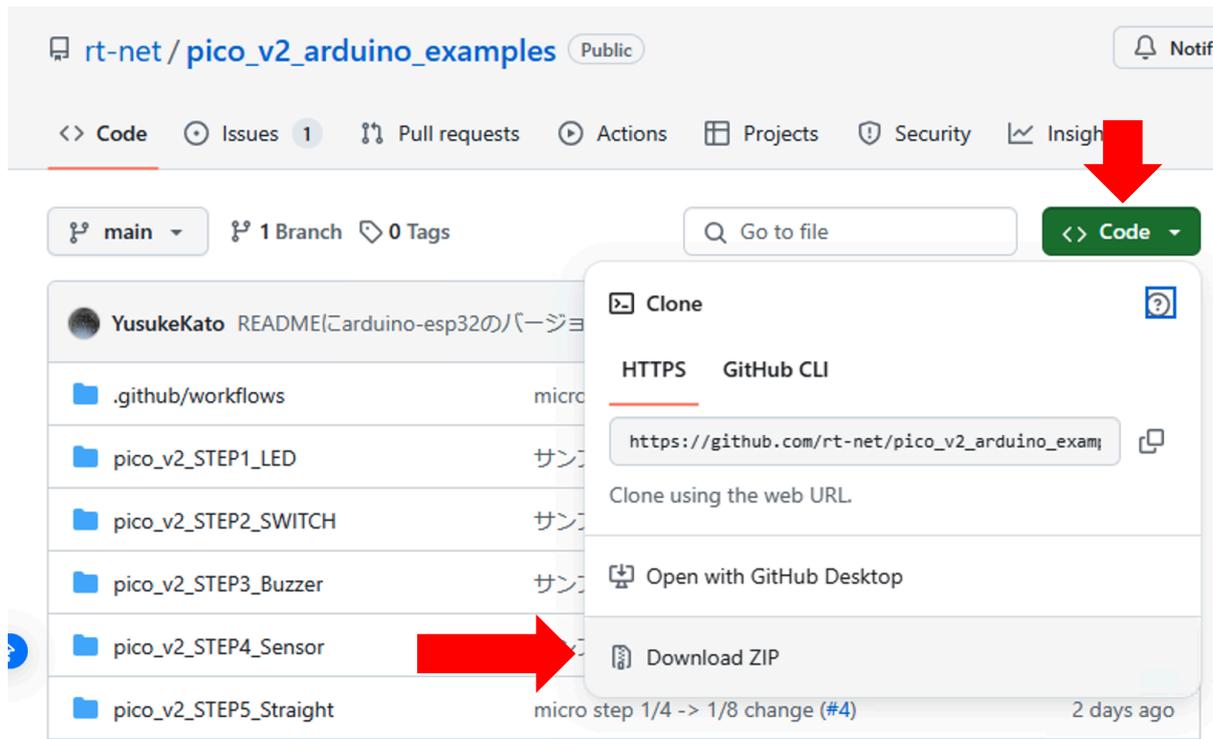
ツール->ボード->esp32->ESP32S3 Dev Modulesを選択

以上で、ESP32-S3のビルド環境設定完了です。

## サンプルスケッチの準備

webブラウザで[https://github.com/rt-net/pico\\_v2\\_arduino\\_examples](https://github.com/rt-net/pico_v2_arduino_examples)にアクセスし、Pi:Co V2のサンプルスケッチファイルをダウンロードします。

ページ内の”Code”から”Download ZIP”をクリックすることでダウンロードできます。ダウンロードしたzipファイルは展開してください。



Code->Download ZIPを選択してサンプルスケッチをダウンロード

ここではPi:Co V2のLEDを点灯させるサンプルスケッチ（pico\_v2\_STEP1\_LED）を例にとってスケッチのビルドと書き込み方法を説明します。

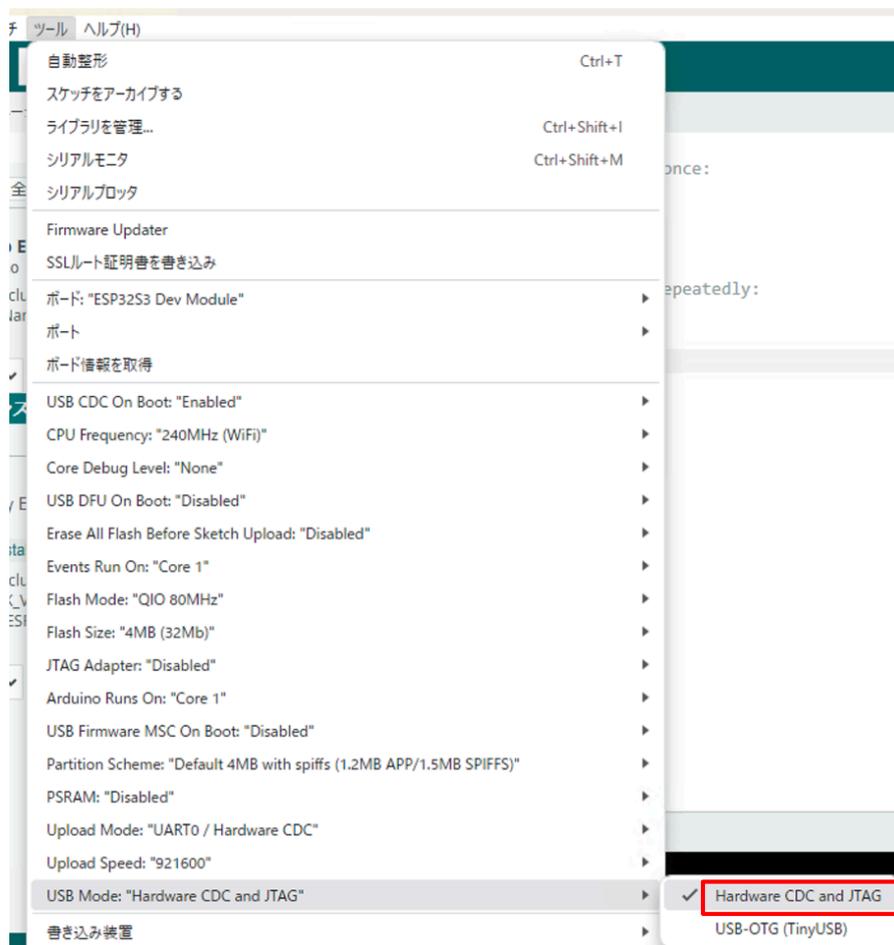
本製品にはUSB経由でスケッチを書き込みます。デフォルトではUSBからの書き込みができないため、書き込みができるように次の2箇所変更します。

まず、メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択します。次に、メニューバーのツール->USB CDC On Bootを開き、“Enabled”にします。



USB CDC On BootをEnabledに設定

メニューバーのツール->USB Modeを開き、“Hardware CDC and JTAG”にします。



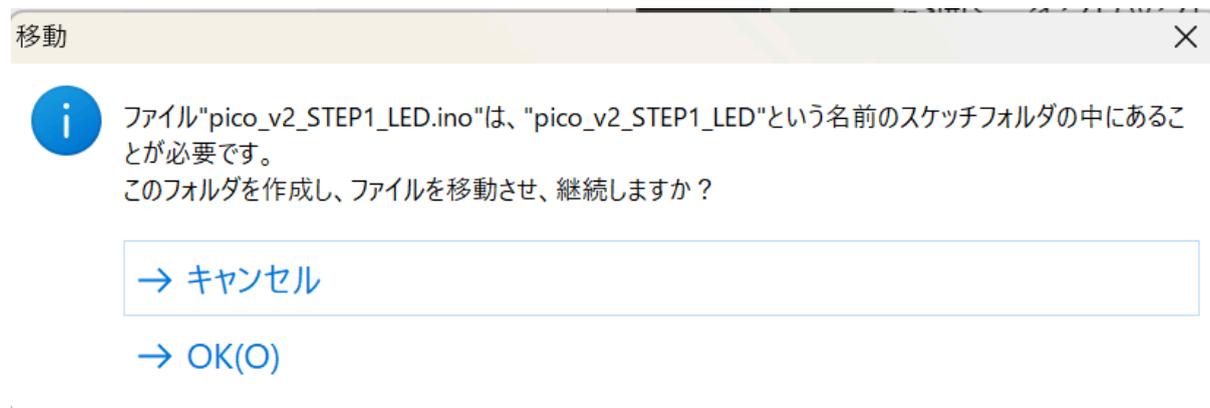
USB ModeをHardware CDC and JTAGに設定

ダウンロードしたサンプルスケッチ内のpico\_v2\_STEP1\_LED/pico\_v2\_STEP1\_LED.inoファイルをダブルクリックで開きます。または、Arduino IDEを起動し、メニューバーのファイル->開く...をクリックしてpico\_v2\_STEP1\_LED.inoを開きます。



ファイル->開く...を選択する

スケッチファイルを開く時に以下のメッセージが出た場合は、スケッチファイル名とフォルダ名が一致していない、またはフォルダが無いため新しくフォルダを作って良いか、ということを確認しています。Arduino IDEの仕様上、メインとなる”ファイル名”と”フォルダ名”が同じでないときにこのメッセージが表示されます。ファイル名やフォルダ名を変更する場合はそれぞれ同じ名前にしてください。



ファイル名とフォルダ名が一致しない場合の確認画面

サンプルスケッチを開くと以下のような画面が表示されます。

```

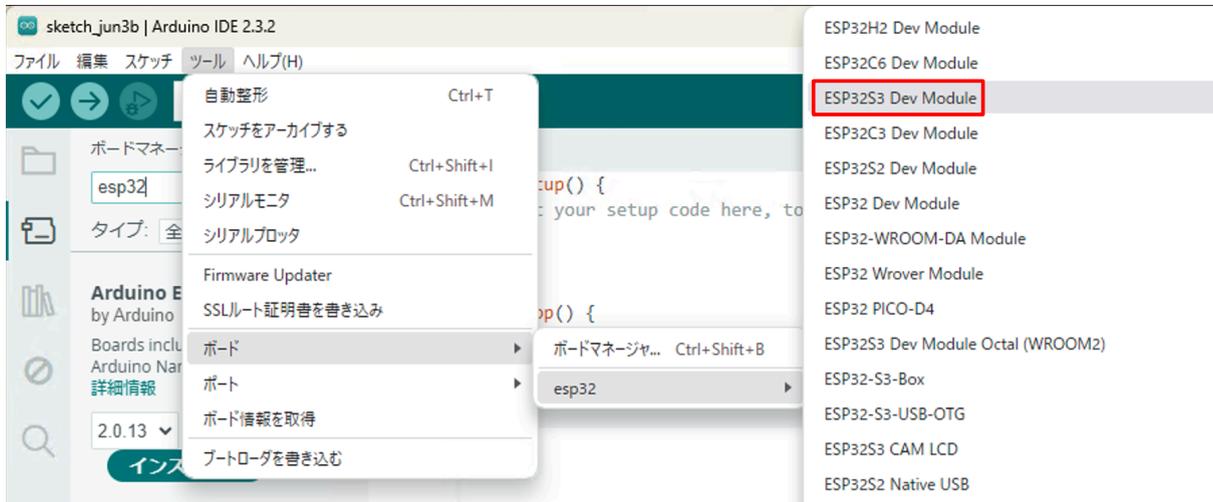
1 // Copyright 2023 RT Corporation
2 //
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the License at
6 //
7 // http://www.apache.org/licenses/LICENSE-2.0
8 //
9 // Unless required by applicable law or agreed to in writing, software
10 // distributed under the License is distributed on an "AS IS" BASIS,
11 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 // See the License for the specific language governing permissions and
13 // limitations under the License.
14
15 #define LED0 42
16 #define LED1 41
17 #define LED2 15
18 #define LED3 16
19
20 void setup()
21 {
22   // put your setup code here, to run once:
23   pinMode(LED0, OUTPUT);
24   pinMode(LED1, OUTPUT);
25   pinMode(LED2, OUTPUT);
26   pinMode(LED3, OUTPUT);
27 }
28
29 void loop()
30 {
31   // put your main code here, to run repeatedly:
32   digitalWrite(LED0, HIGH);
33   digitalWrite(LED1, HIGH);
34   digitalWrite(LED2, HIGH);
35   digitalWrite(LED3, HIGH);
36   delay(500);
37   digitalWrite(LED0, LOW);
38   digitalWrite(LED1, LOW);
39   digitalWrite(LED2, LOW);
40   digitalWrite(LED3, LOW);
41   delay(500);
42 }
43

```

行 1、列 1 × ボード未選択

pico\_v2\_STEP1\_LED.inoを開いた状態

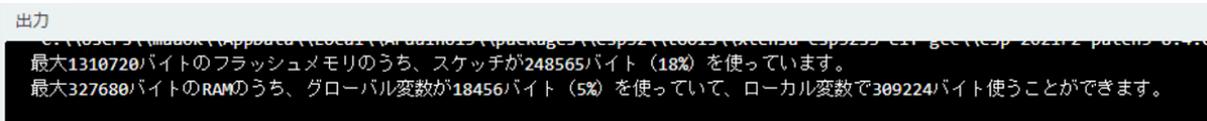
ボードが”ESP32S3 Dev Module”になっていないときは、メニューバーのツール->ボード->esp32->ESP32S3 Dev Moduleを選択してください。



ツール->ボード->esp32->ESP32S3 Dev Moduleを選択

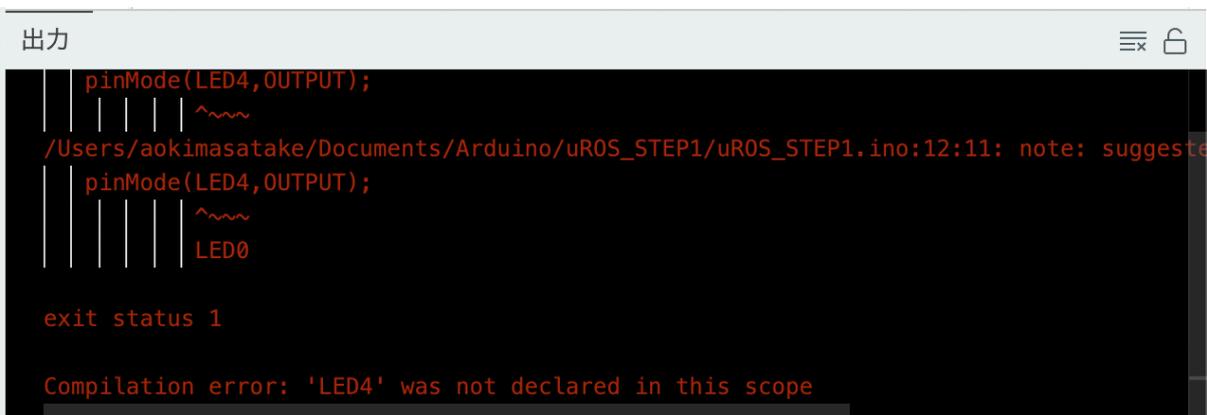
## ビルド

左上にあるアイコン  で”検証”を実行しスケッチをビルドします。スケッチの記述に問題がなければ以下のようなメッセージが表示されます。



検証に問題が無いときのメッセージ

スケッチの記述にエラーがあった場合は、赤文字でエラー箇所が表示されます。



検証エラー時のメッセージ

Arduino IDEの右下にあるアイコン  をクリックすればメッセージを非表示にできます。もう一度クリックすると表示されます。画面を大きくしてスケッチ全体を少しでも多く見たい時や、このメッセージ画面が不要なとき等に活用してください。

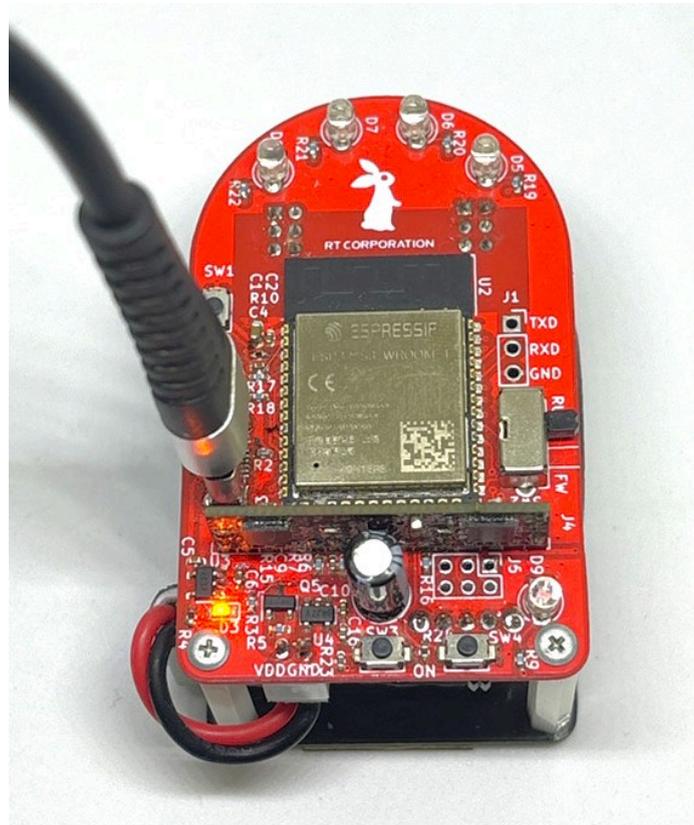
Ubuntuの場合、ModuleNotFoundError: No module named 'serial'のメッセージが出力されたときはビルドができていません。下記のコマンドをターミナルで実行し、ビルドに必要なアプリをインストールしてください。

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

## 書き込み

スケッチにエラーが無いことを確認したら、次の手順で書き込み前の準備をします。

1. 本体にバッテリーが繋がっていることを確認します。
2. PCと本製品をUSBケーブルで接続します。
3. 本製品のモード切替スイッチをFWと書かれている方向にスライドします。
4. Pi:Co V2の電源スイッチをON側にスライドします。
5. 本製品のリセットボタンを押します。



モード切替スイッチを操作し、USBケーブルを接続した状態

本製品をPCと接続した後、以下の手順でボードとポートを設定します。本製品の電源をOFFした後や、リセットボタンを押した後にも再度設定してください。

メニューバーの下に表示されているボード名（ESP32S3 Dev Module）をクリックし、一番下の”他のボードとポートを選択...”を開きます。



ボード名->他のボードとポートを選択...を開く

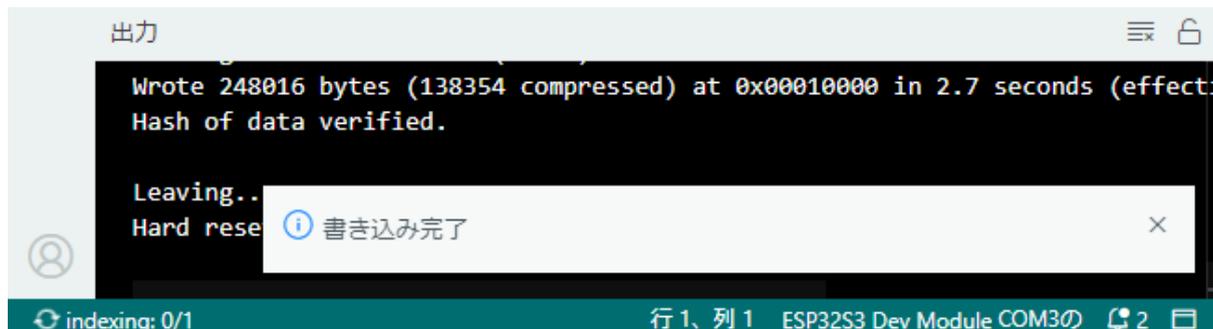
ボードの検索欄に”dev”を入力し、”ESP32S3 Dev Module”にチェックが入っていることを確認します。チェックが入っていない場合は、ESP32S3 Dev Moduleをクリックします。ポート選択欄では本製品を接続したUSBポートを選択します。Windowsの場合はCOMX、Linux(Ubuntu)の場合は/dev/ttyUSBX、macOSの場合はusbmodemXXXXX Serial Port(USB)という名称です。選択したら”OK”をクリックします。



ボードとポートの設定

設定完了後、ポート名の左にUSBのマーク     ESP32S3 Dev Module が追加されます。スケッチの書き込み以外に、PCとハードウェアシリアル通信をするときも、ポート設定が必要です。

ポート設定後、 をクリックしてスケッチを書き込みます。下記のように表示されたら書き込み完了です。



書き込み完了時の画面表示

Linux(Ubuntu) PCで書き込みに失敗する時は、書き込み前にターミナルで次のコマンドを実行し、USBポートの使用権限を変更します。

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

永続的にUSBポートの使用権限を変更する場合は、次のコマンドを実行し、PCを再起動します。

```
# 下記を実行した後にPCを再起動すること
$ sudo usermod -aG dialout $USER
```

スケッチの書き込み完了後、本製品のモード切替スイッチをRUN側にスライドし、リセットボタンを押します。

Pi:Co V2前方のLEDが0.5秒点滅、0.5秒消灯を繰り返します。

## トレーニングトレーサー

---



### STM32ボード情報の追加

スケッチを本製品に書き込めるように、STM32のボード情報をArduino IDEに追加します。メニューバーのファイル->基本設定...を選択します。画面下部の”追加のボードマネージャのURL”に

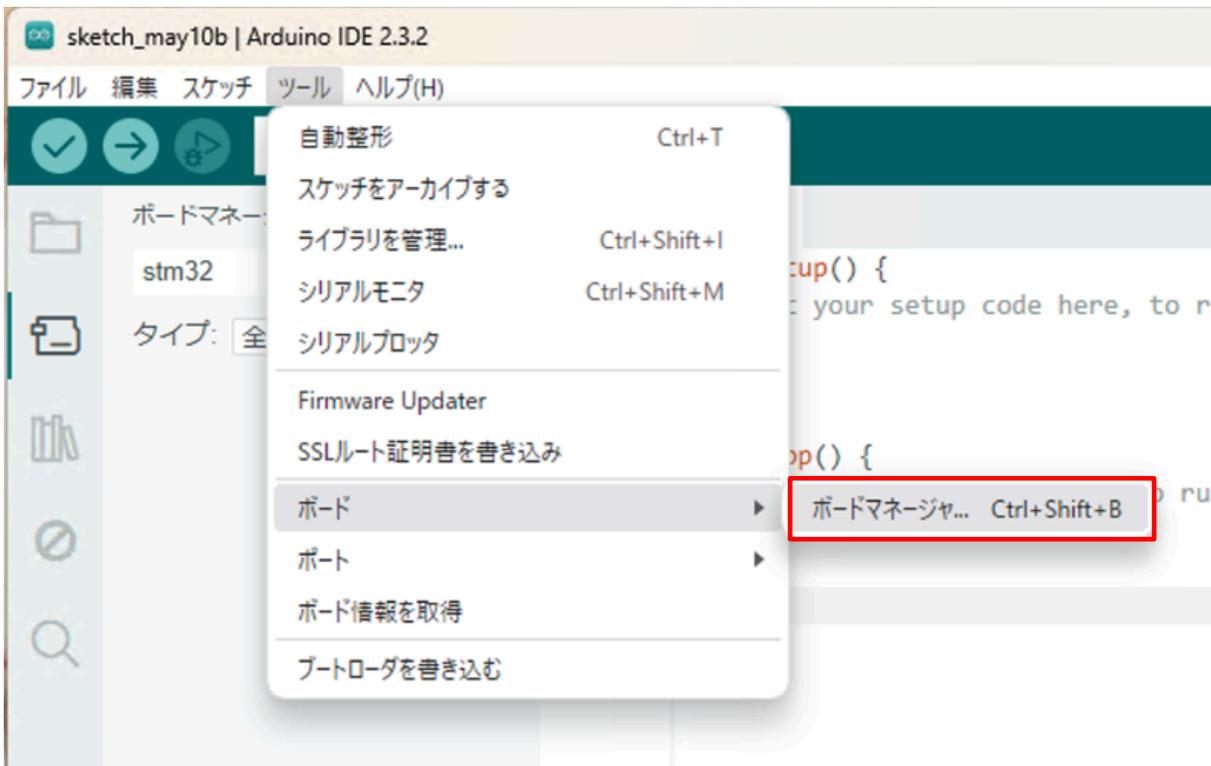
```
https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json
```

を追加し”OK”をクリックします。



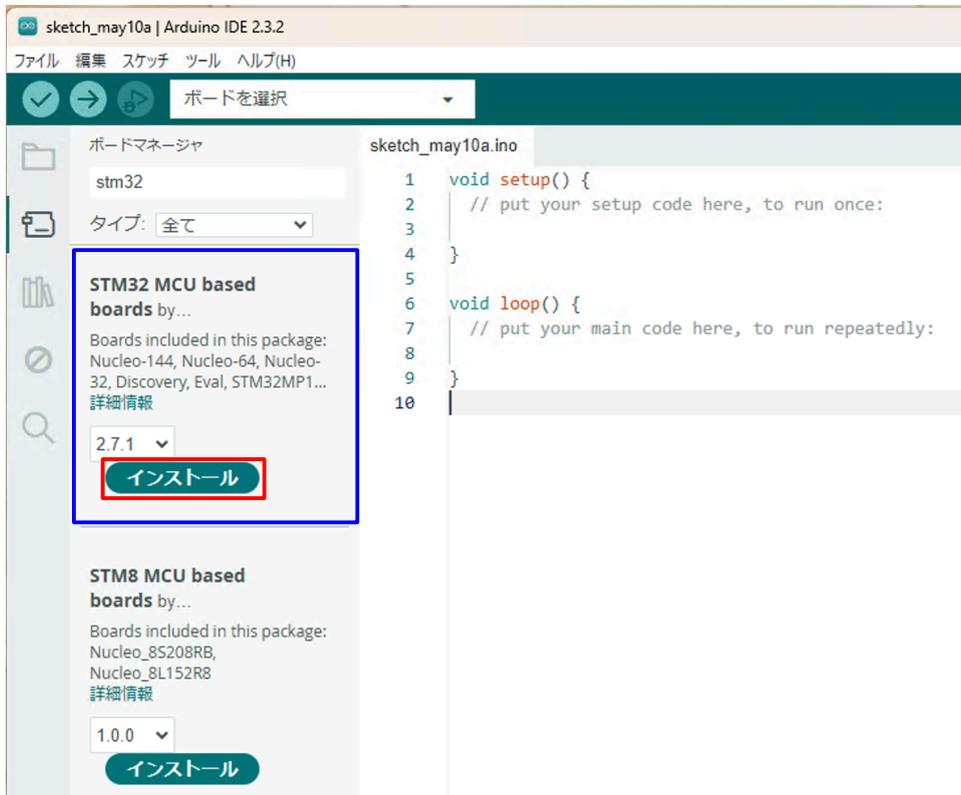
追加ボードマネージャのURLに追加した状態

URLを追加したら、STM32ライブラリやツールをダウンロードしてインストールします。メニューバーのツール->ボード->ボードマネージャ...を選択します。



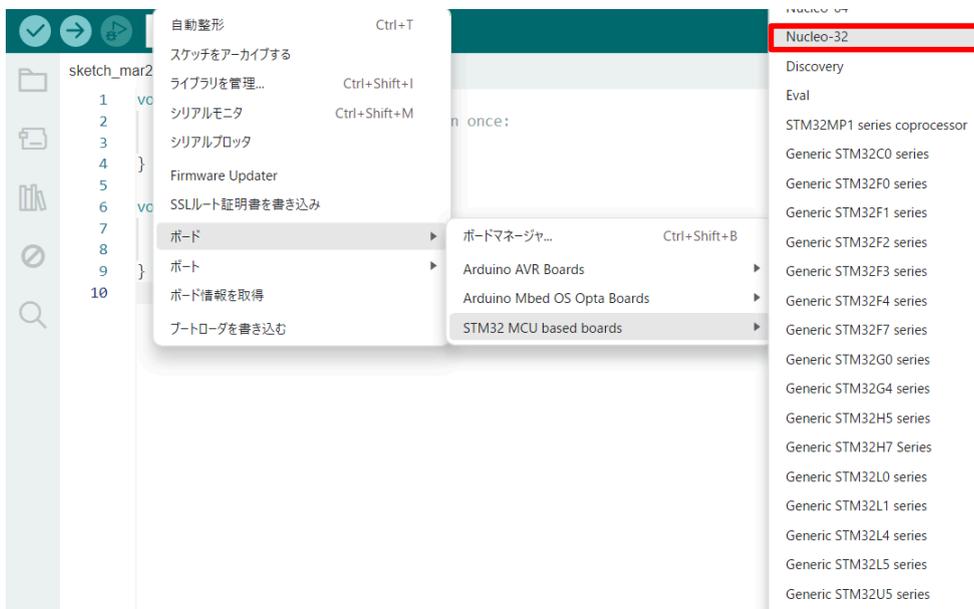
ボードマネージャ...を選択した状態

左側にあるボードマネージャの検索欄に"esp32"と入力すると"STM32 MCU Based boards by STMicroelectronics"が表示されます。2024年6月時点では2.7.1が最新です。STM32 MCU Based boards by STMicroelectronicsのインストールをクリックします。



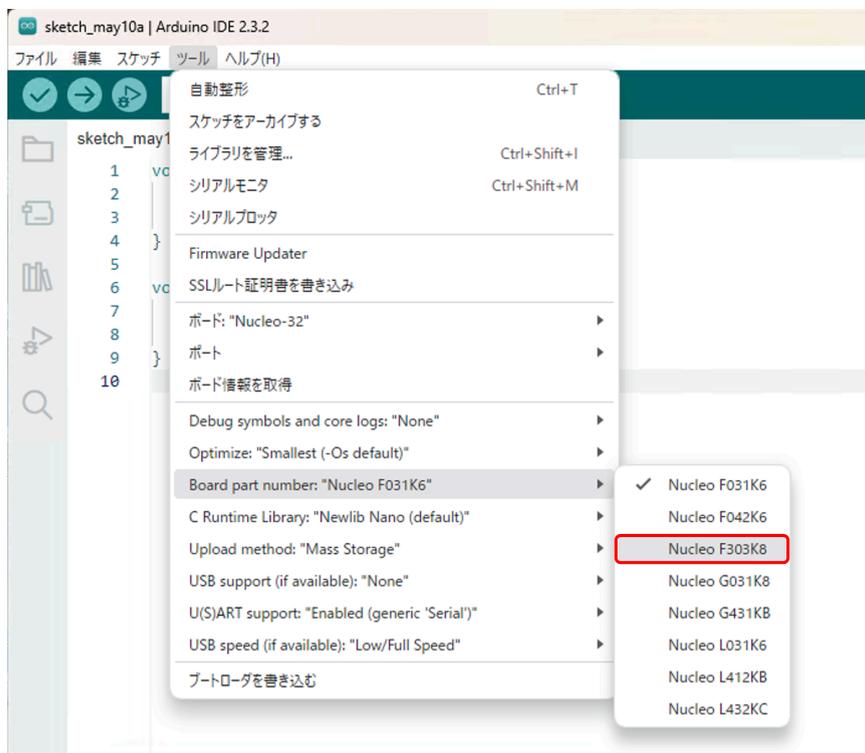
ボードマネージャ画面の検索欄に"stm32"を入力した状態

Arduino IDEは STM32以外のマイコンにも対応しているため、スケッチを書き込む際は STM32のボードを選択します。メニューバーのツール->ボード->STM32 MCU based boards ->Nucleo-32を選択します。"ボード"と"ポート"が同じような文字で上下にあるため間違えないように選択してください。



ツール->ボード->STM32 MCU based boards->Nucleo-32を選択

次にNucleo-32から使用するボードを選択します。ツール -> Board part numberでNucleo F303K8を選択します。



ツール->Board part Number->Nucleo F303K8を選択  
 以上で、STM32 Nucleo F303K8のビルド環境設定完了です。

## 書き込みツールSTM32CubeProgrammerのインストール

Arduino IDEからNucleoボードにプログラムを書き込むにはSTMicroelectronics社のSTM32CubeProgrammerに含まれているSTM32\_Programmer\_CLIが必要になります。

STM32CubeProgrammerは<https://www.st.com/ja/development-tools/stm32cubeprog.html>からダウンロードします。

使用するOSに合うソフトウェアを”最新バージョンの取得”をクリックしてダウンロードします。

### 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

ライセンス契約について確認し”承諾します”をクリックします。



MySTアカウントでログインします。アカウントを所持していない場合は新規作成してください。



登録してあるメールアドレスとパスワードを入力しログインをクリックします。

### 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

パスワード

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**

ログインが成功するとダウンロードが開始されます。

STM32CubeProgrammerインストーラの起動方法はOS毎に異なりますが、インストール手順はほとんど同じです。ここではWindowsの画面で説明します。

### Windowsでのインストーラ起動方法

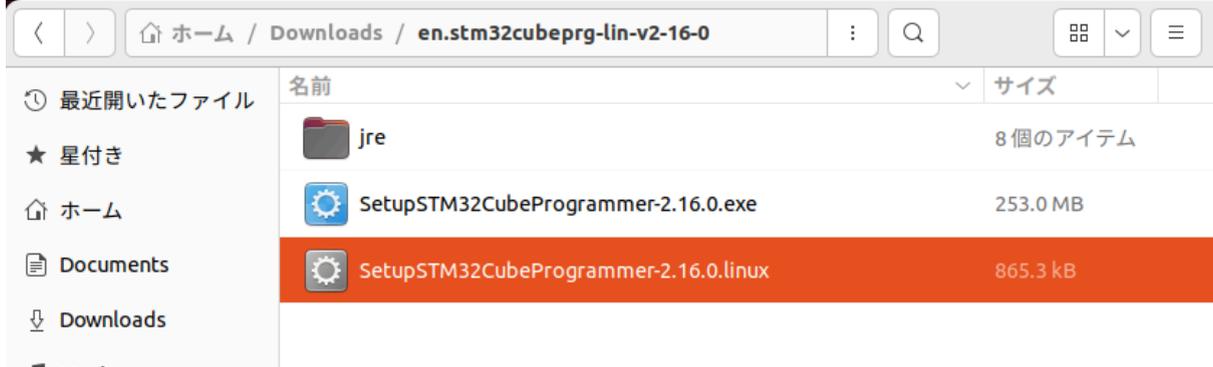
ダウンロードしたen.stm32cubeprog-win64-v2-16-0.zipファイルを右クリックし”すべてを展開”を選択し展開します。展開されたフォルダにあるSetupSTM32CubeProgrammer\_win64.exeをダブルクリックで実行します。



“はい”をクリックします。

### Linux (Ubuntu) でのインストーラ起動方法

ダウンロードしたen.stm32cubeprg-lin-v2-16-0.zipファイルを右クリックで展開します。展開されたフォルダにあるSetupSTM32CubeProgrammer-2.16.0.linuxをダブルクリックで実行します。



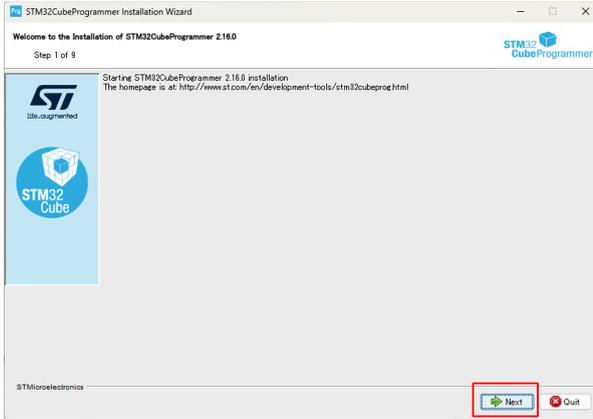
### macOSでのインストーラ起動方法

ダウンロードしたen.stm32cubeprg-mac-v2-16-0.zipファイルをダブルクリックで展開し、SetupSTM32CubeProgrammer-2.16.0をダブルクリックします。

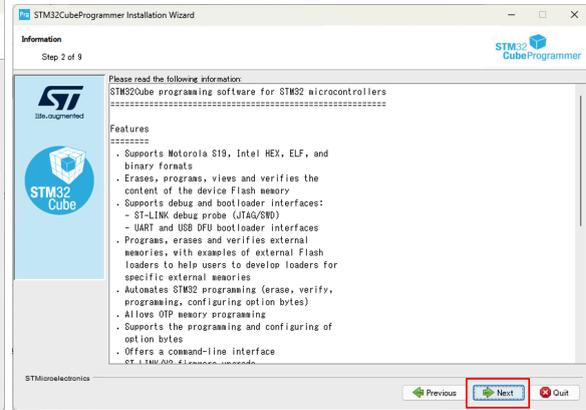


“開く”をクリックします。

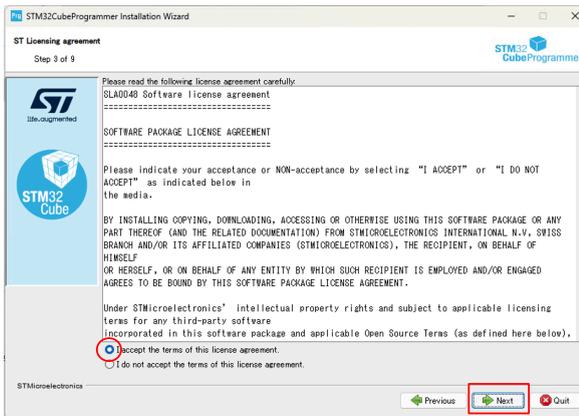
インストーラの操作手順 (OS共通)



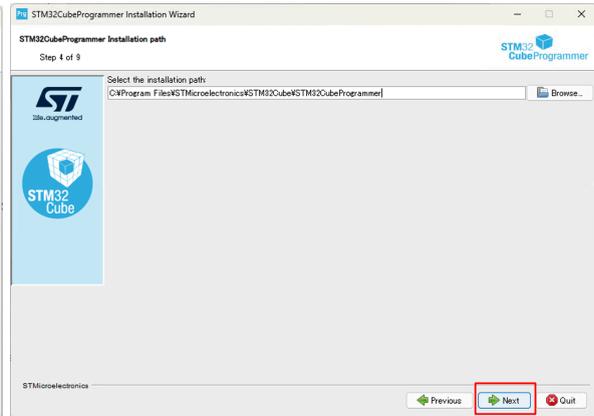
Nextをクリックします。



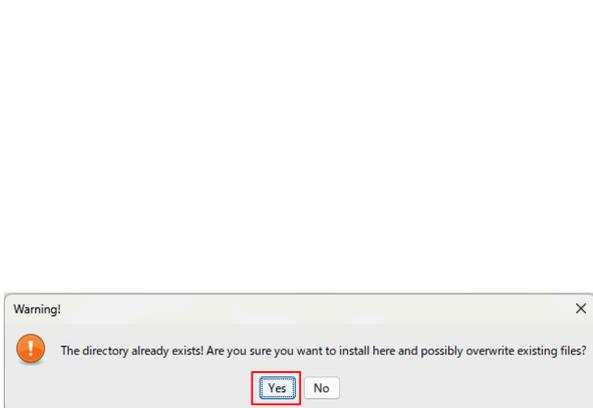
Nextをクリックします。



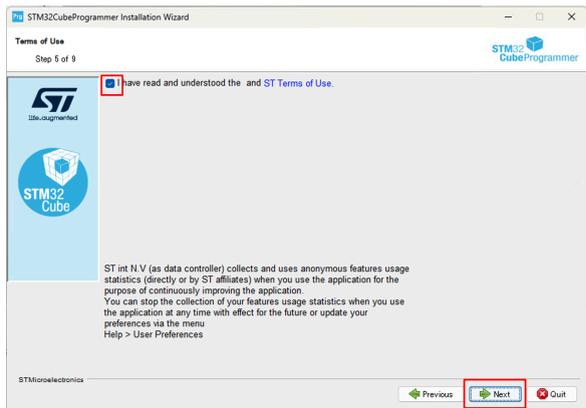
I accept を選択してNextをクリックします。



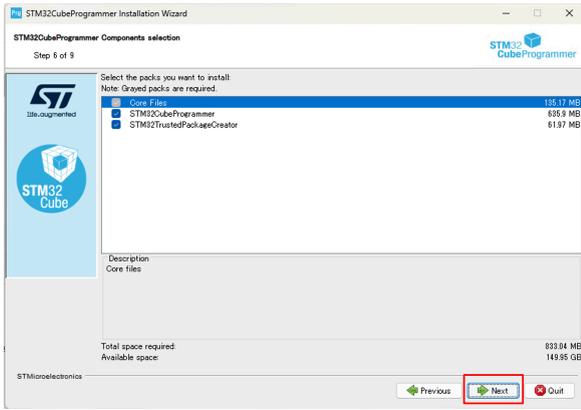
Nextをクリックします。



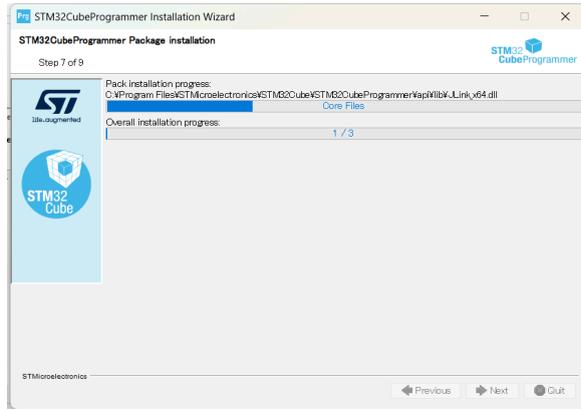
YESをクリックします。



I haveにチェックを入れてNextをクリックします。

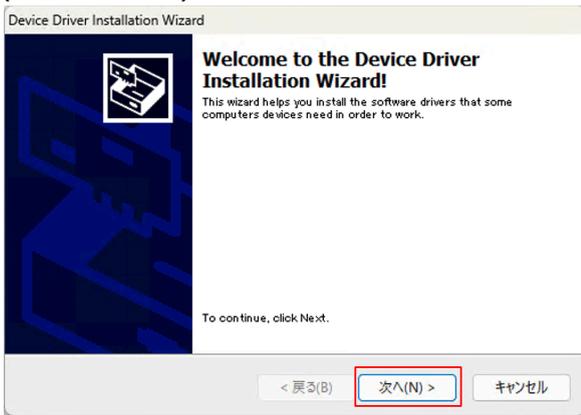


Nextをクリックします。



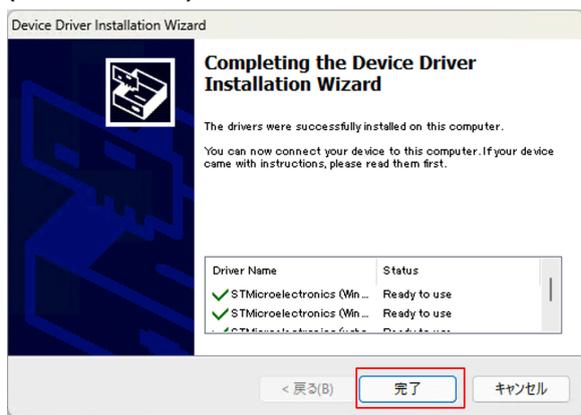
インストールが始まります。

(windowsのみ)

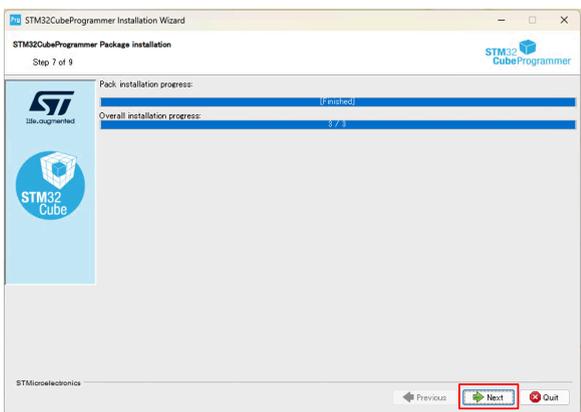


次へをクリックします。)

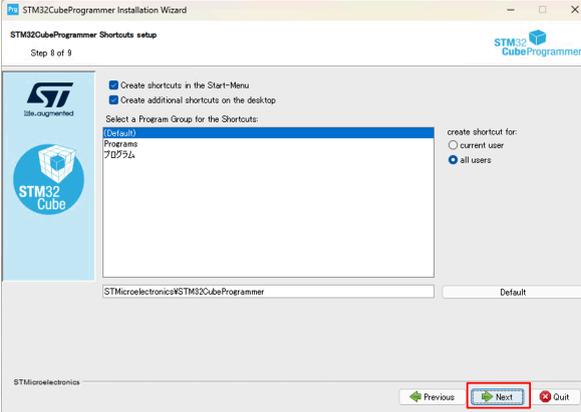
(windowsのみ)



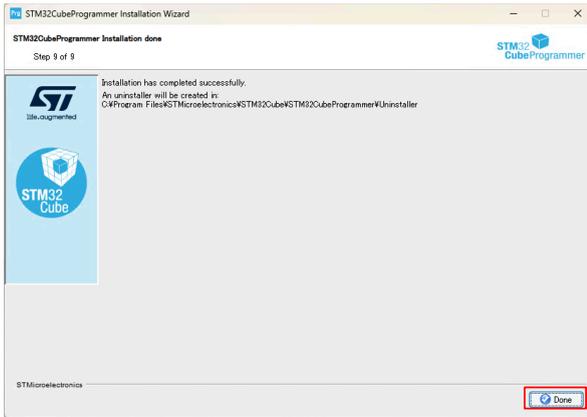
完了をクリックします。



Nextをクリックします。



Nextをクリックします。

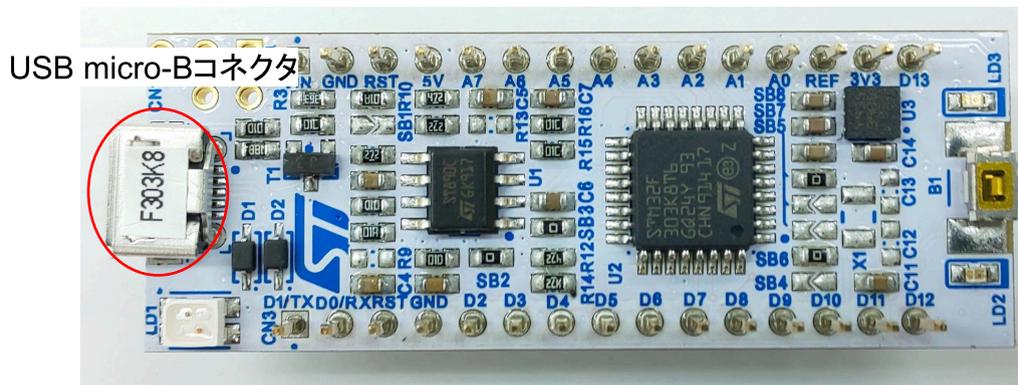


Doneをクリックします。

### デバイスドライバのインストール(Windowsのみ)

ここでは、プログラムの書き込みに使用するデバイスドライバがPCにインストールされているかを確認します。USBケーブルでNucleoボードとPCを接続します。

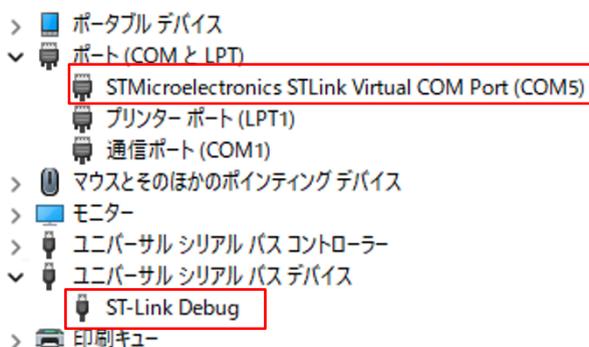
Nucleoボードはトレーニングトレーサに載せたままケーブルを繋いでください。



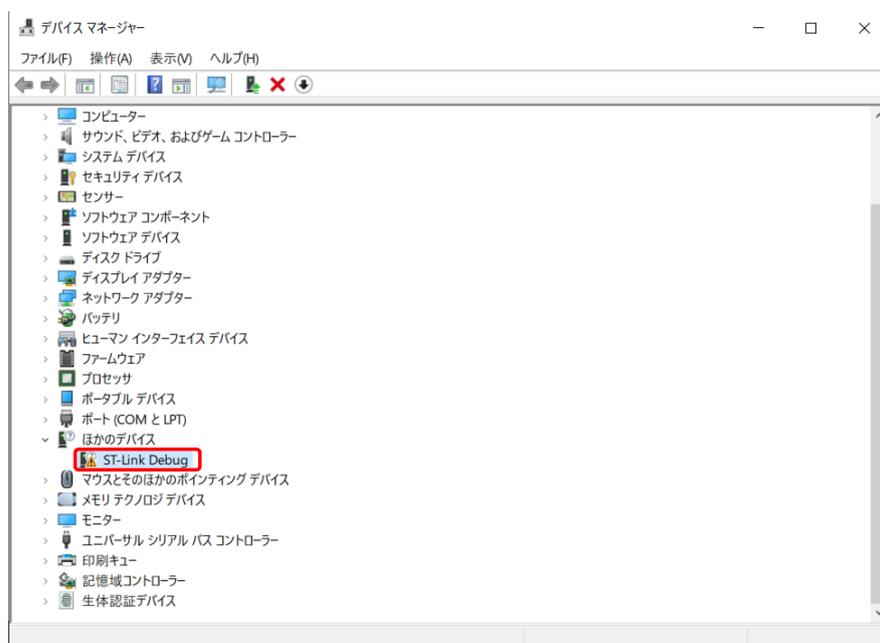
スタートボタンの🔍アイコンをクリックして、「デバイスマネージャー」と入力し、検索結果にあるデバイスマネージャーをクリックします。



ポート(COMとLPT)の項目にSTMicroelectronics STLink Virtual COM Port(COMX)があり、ユニバーサルシリアルバスデバイスの項目にST-Link Debugがあればデバイスドライバをインストールする必要はありません。



ほかのデバイスまたは不明なデバイスの中に下図のような黄色の三角形で警告が出ている場合があります。この場合はデバイスドライバのインストール作業が必要です。



デバイスドライバはSTMicroelectronics社の  
<https://www.st.com/ja/development-tools/stsw-link009.html>から最新バージョンをクリックしてダウンロードします。

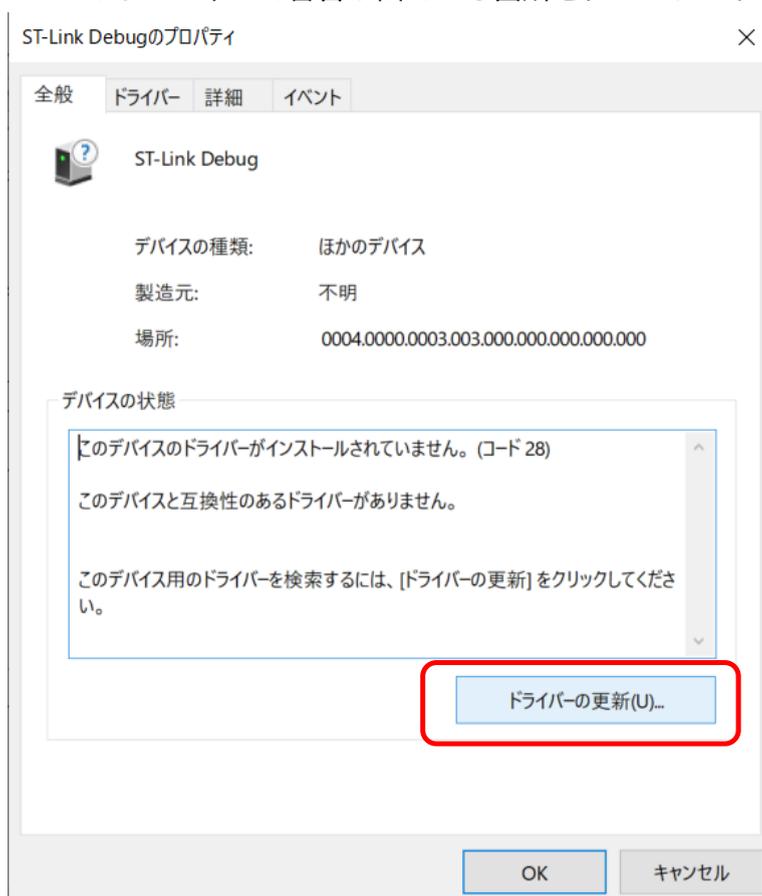
## ソフトウェア入手

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

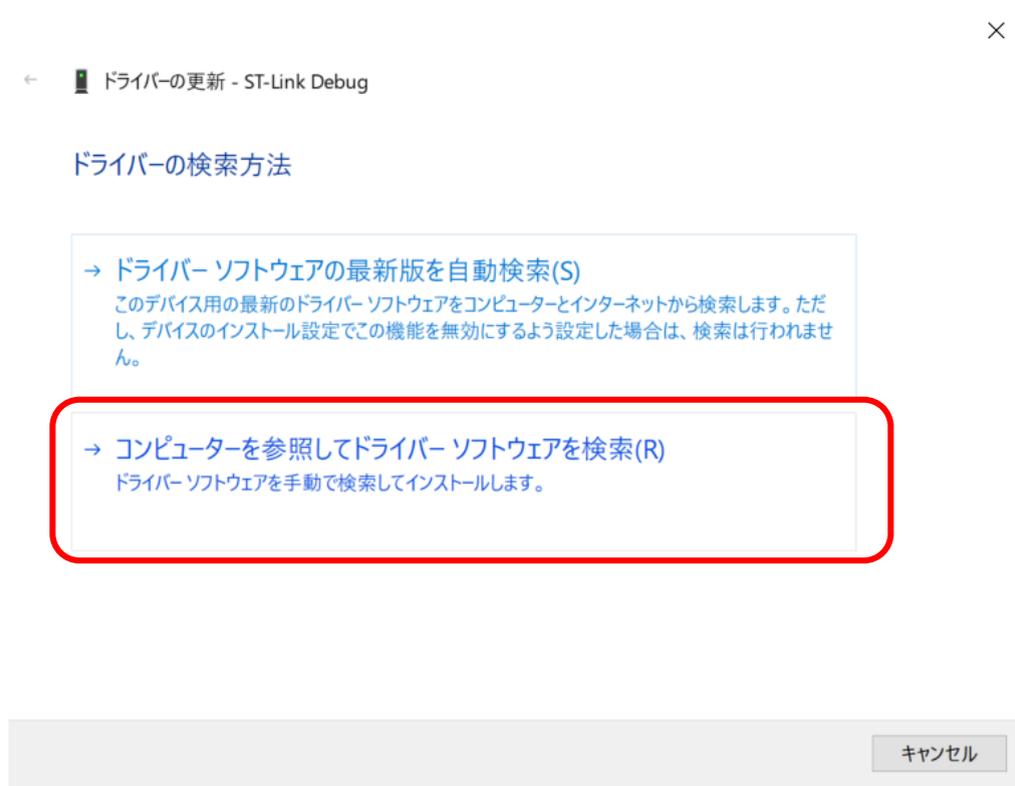
"ソフトウェア入手"をクリックします。ライセンス契約の確認画面が表示されるので"同意"をクリックします。

ダウンロードしたen.stsw-link009.zipを展開します。

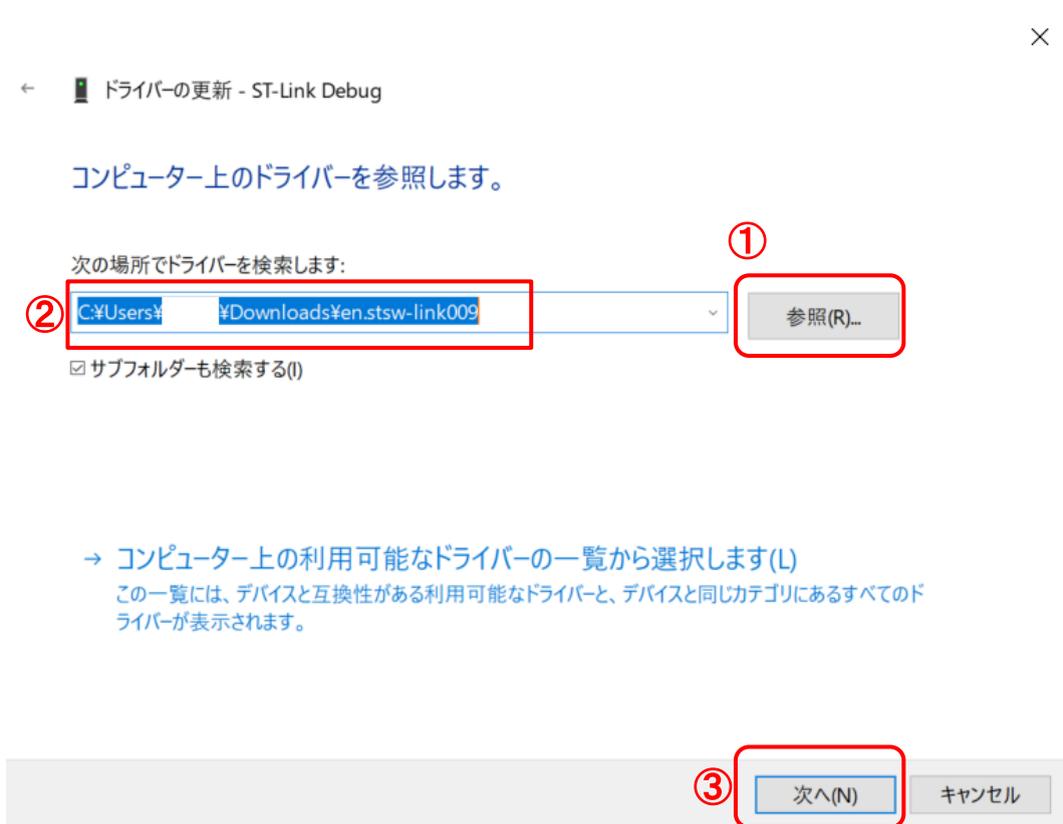
先程開いたデバイスマネージャーで警告が出ている箇所をダブルクリックして開きます。



プロパティが開いたら"ドライバーの更新"をクリックします。



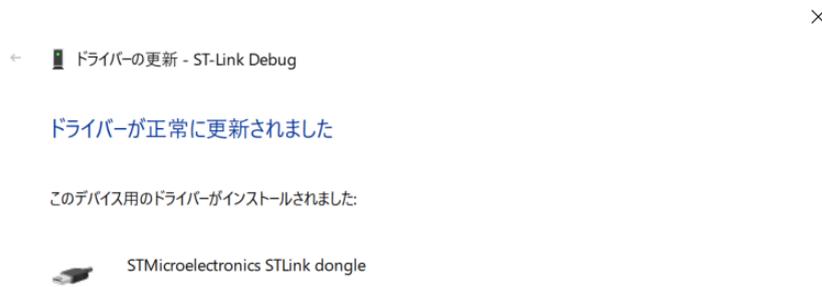
ドライバーの検索方法という画面が開いたら、"コンピューターを参照してドライバーソフトウェアを検索"をクリックします。



"コンピューター上のドライバーを参照します。"という画面で参照をクリックし、先程ダウンロードして展開したen\_stsw-link009があるフォルダを選択します。



"Windowsセキュリティ"という画面で"インストール"をクリックします。



"正常に更新されました"と表示されたら"閉じる"をクリックします。

## サンプルスケッチの準備

webブラウザで[https://github.com/rt-net/TrainingTracer\\_V2\\_Samples](https://github.com/rt-net/TrainingTracer_V2_Samples) にアクセスし、トレーニングトレーサーのサンプルスケッチファイルをダウンロードします。

ページ内の”Code”から”Download ZIP”をクリックすることでダウンロードできます。ダウンロードしたzipファイルは展開してください。

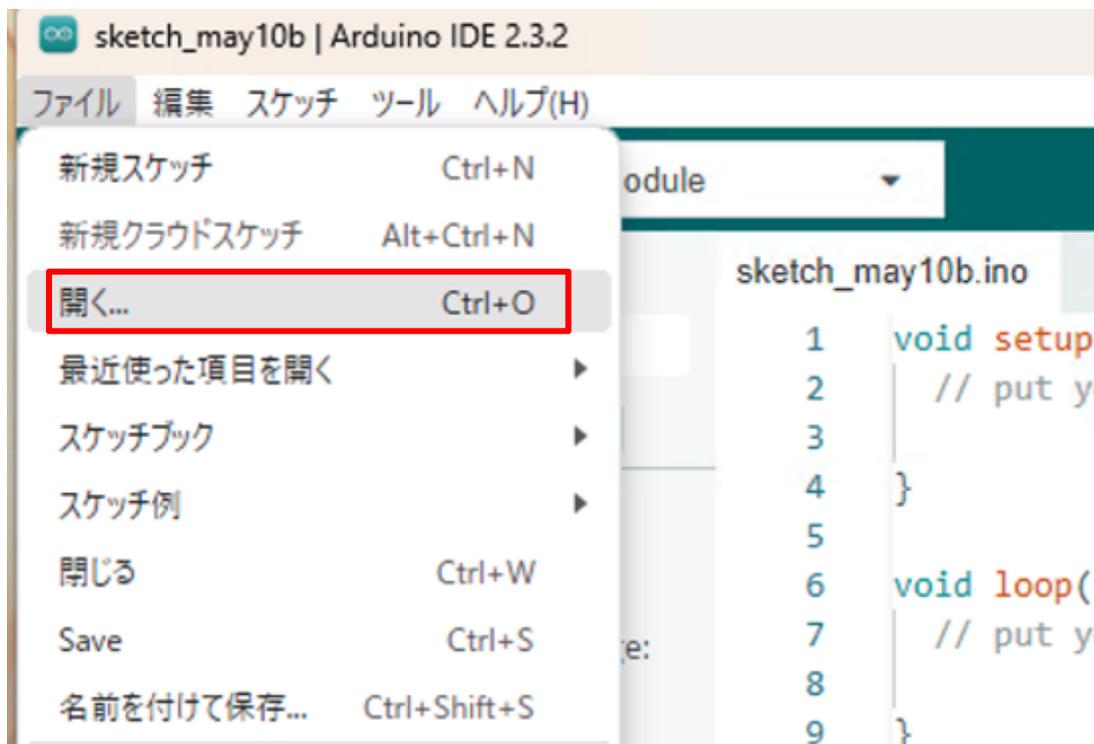


Code->Download ZIPを選択してサンプルスケッチをダウンロード

ここではNucleoのボード上にあるLEDを点灯させるサンプルスケッチ（STEP1）を例にとってスケッチのビルドと書き込み方法を説明します。

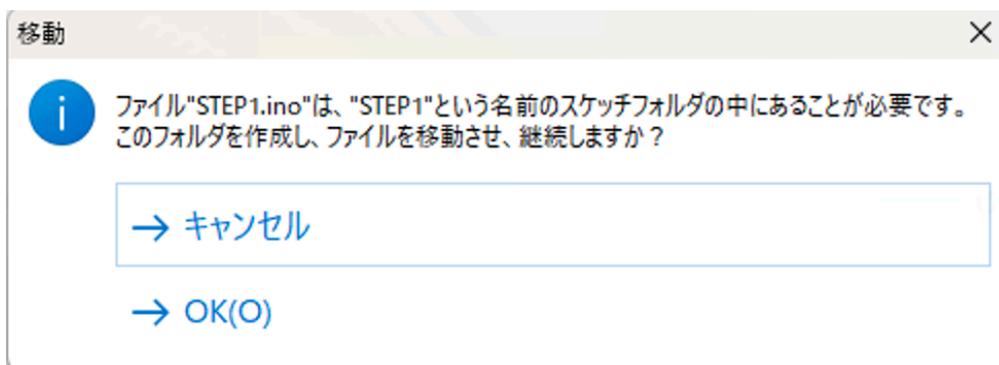
本製品にはUSB経由でスケッチを書き込みます。

ダウンロードしたサンプルスケッチ内のArduino\_Sample/STEP1/STEP1.inoファイルをダブルクリックで開きます。または、Arduino IDEを起動し、メニューバーのファイル->開く...をクリックしてSTEP1.inoを開きます。



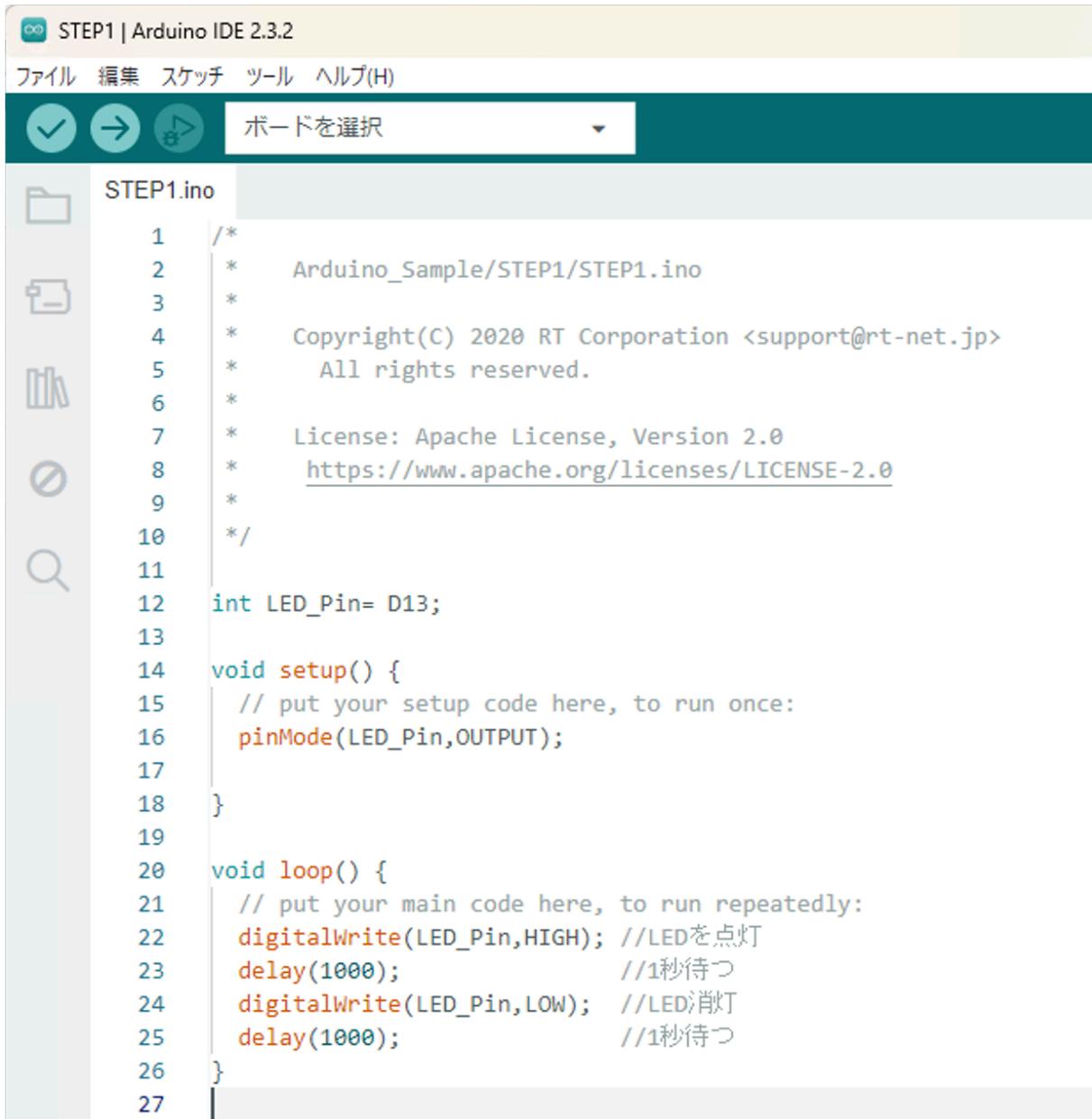
ファイル->開く...を選択する

スケッチファイルを開く時に以下のメッセージが出た場合は、スケッチファイル名とフォルダ名が一致していない、またはフォルダが無いため新しくフォルダを作って良いか、ということを確認しています。Arduino IDEの仕様上、メインとなる”ファイル名”と”フォルダ名”が同じでないときにこのメッセージが表示されます。ファイル名やフォルダ名を変更する場合はそれぞれ同じ名前にしてください。



ファイル名とフォルダ名が一致しない場合の確認画面

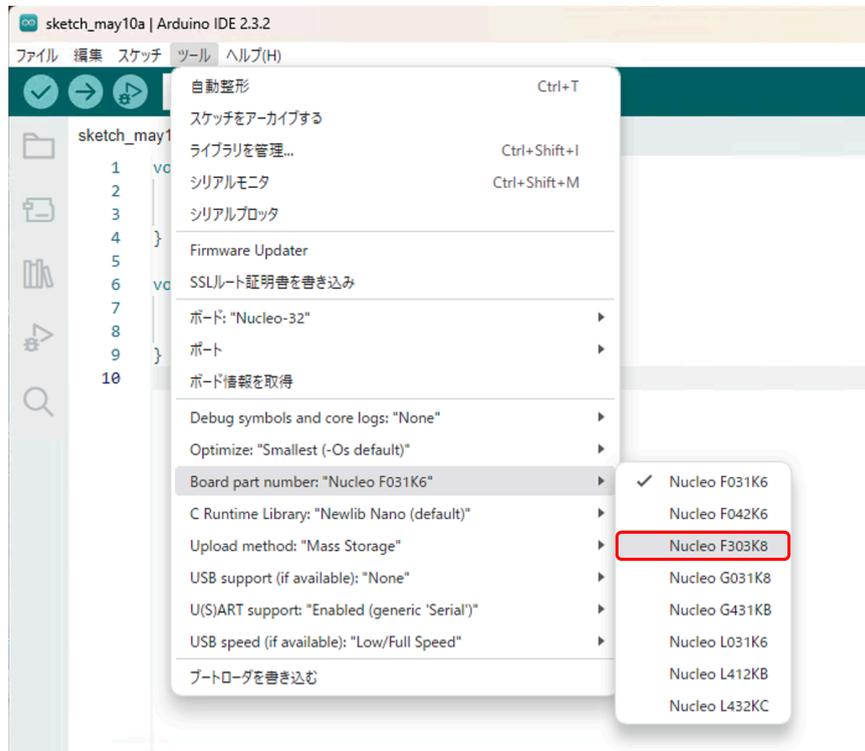
サンプルスケッチを開くと以下のような画面が表示されます。



```
STEP1 | Arduino IDE 2.3.2
ファイル 編集 スケッチ ツール ヘルプ(H)
ボードを選択
STEP1.ino
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を開いた状態

ボードが”Nucleo-32”になっていないときは、メニューバーのツール->ボード->STM32 MCU based boards ->Nucleo-32を選択してください。また Board part numberがNucleo F303K8になっていることも確認してください。Board part numberはツール-> Board part numberで選択できます。



ツール-> Board part number->Nucleo-F303K8を選択

## ビルド

左上にあるアイコン  で”検証”を実行しスケッチをビルドします。スケッチの記述に問題がなければ以下のようなメッセージが表示されます。

```
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。
```

### 検証に問題が無いときのメッセージ

スケッチの記述にエラーがあった場合は、赤文字でエラー箇所が表示されます。

```
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\xtensa-esp32-none-elf-gcc\12.2.1-1.2
"C:\Users\maaok\AppData\Local\Arduino15\packages\STMicroelectronics\tools\xtensa-esp32-none-elf-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'?
```

### 検証エラー時のメッセージ

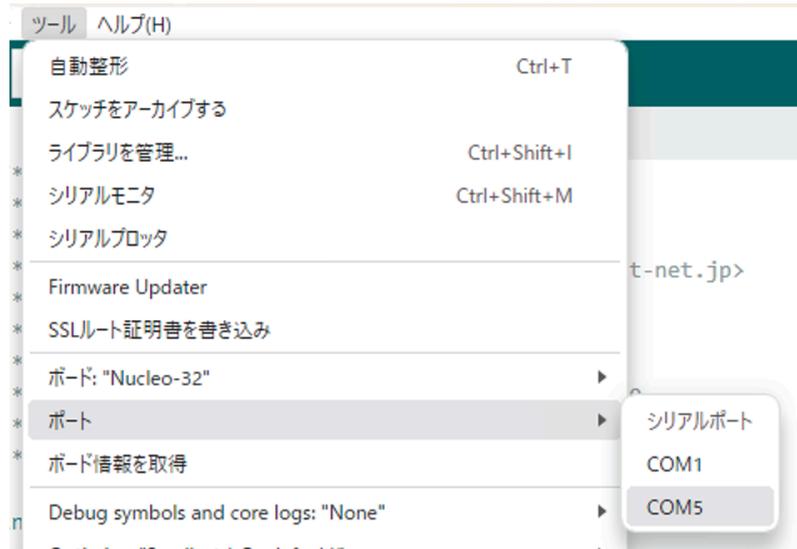
Arduino IDEの右下にあるアイコン  をクリックすればメッセージを非表示にできます。もう一度クリックすると表示されます。画面を大きくしてスケッチ全体を少しでも多く見たい時や、このメッセージ画面が不要なとき等に活用してください。

## 書き込み

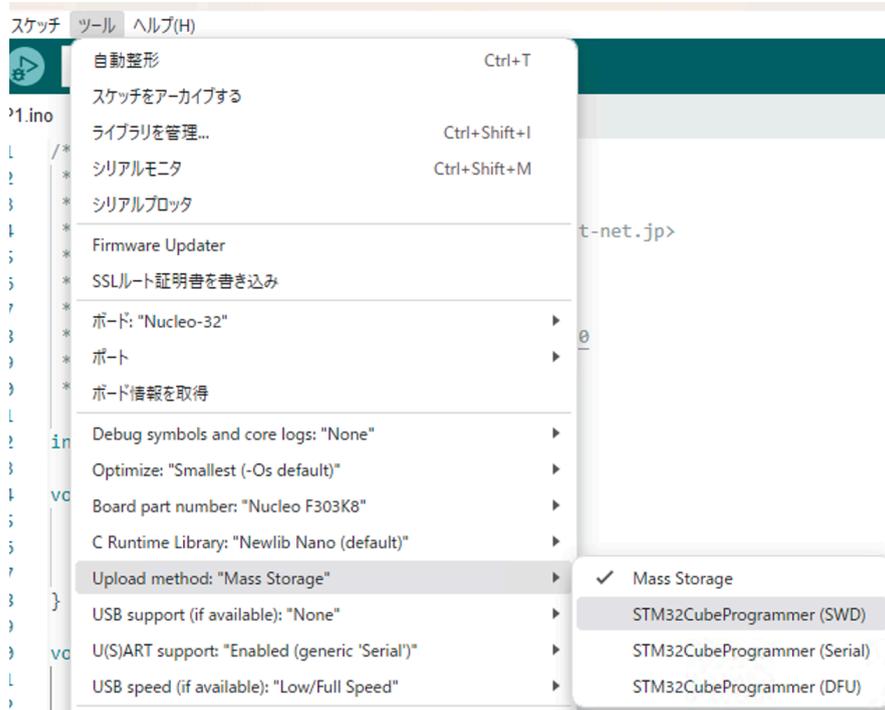
書き込みにはCOMポートとSWDを使用します。

COMポートはツール->ポートで設定します。COM番号はデバイスマネージャーで確認できます。

Windowsの場合はCOMX、Linux(Ubuntu)の場合は/dev/ttyUSBX、macOSの場合はusbmodemXXXXX Serial Port(USB) という名称です。

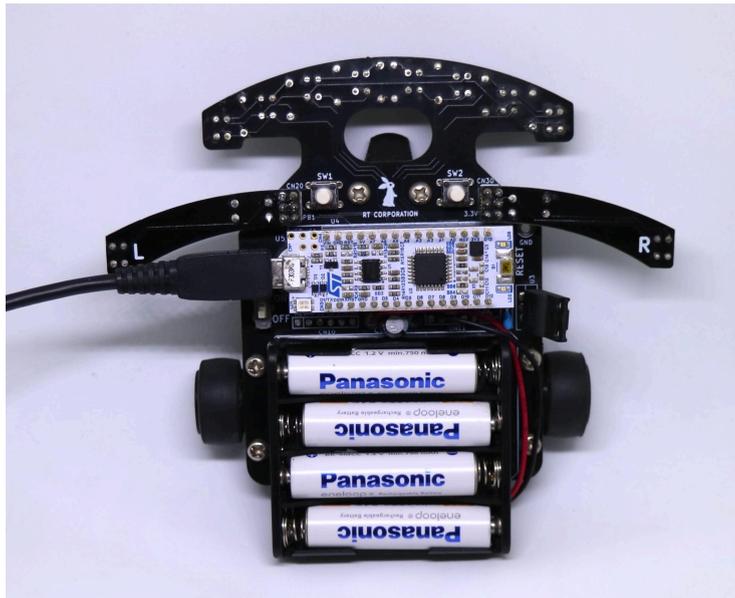


SWDはツール->Upload method->STM32CubeProgrammer(SWD)を選択します。



スケッチにエラーが無いことを確認したら、次の手順で書き込み前の準備をします。

1. ポート番号とUpload methodが正しいか確認します。
2. 本体に乾電池をセットしていることを確認します。
3. PCと本製品をUSBケーブルで接続します。
4. **タイヤを浮かせます。**（書き込み実行時に左のモータが回転するため）
5. 電源スイッチをON側にスライドします。



USBケーブルを接続した状態

正しく設定されているとポート名の左にUSBのマーク  Nucleo-32  が追加されます。スケッチの書き込み以外に、PCとハードウェアシリアル通信をするときも、ポート設定が必要です。

ポート設定後、 をクリックしてスケッチを書き込みます。下記のように表示されたら書き込み完了です。

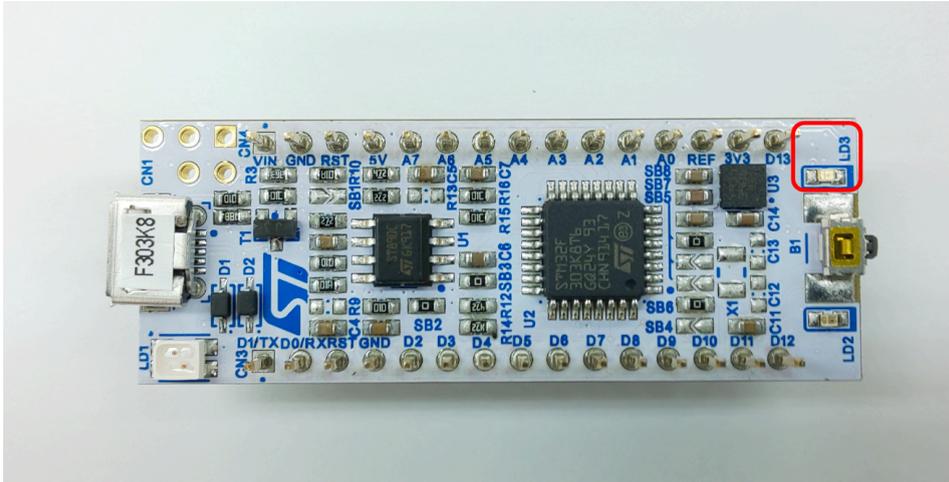
#### 出力

```
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
```

書き込み完了時の画面表示

Nucleoボード上にあるLEDが0.5秒点滅、0.5秒消灯を繰り返します。



## 書き込み時のエラー対処方法

ST-Linkのファームウェアが古いと以下のメッセージが出ます。このメッセージが表示された時は、STlinkのファームウェアのアップデートを行ってください。

```
-----
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.
```

Error: No debug probe detectedの表示が出た時は、PCとNucleoボードがUSBケーブルで正常に接続されていません。USBケーブルを接続し直し、再度書き込みをしてください。

```
-----
| | | | | | | | | | STM32CubeProgrammer v2.16.0
-----
Error: No debug probe detected.
Failed uploading: uploading error: exit status 1
```

Error: No STM32 target foundの表示が出た時は、本製品の電源が入っていません。電源をON側にスライドしたあと再度書き込みをしてください。

```
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ファームウェアのアップデート

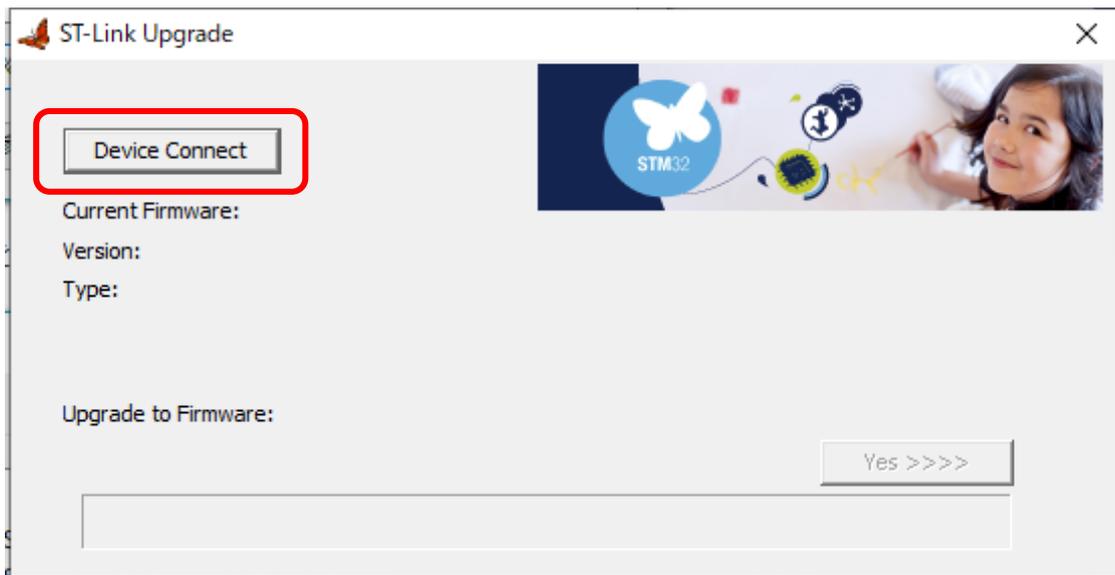
以下のページからST-Linkのファームウェアをダウンロードします。

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

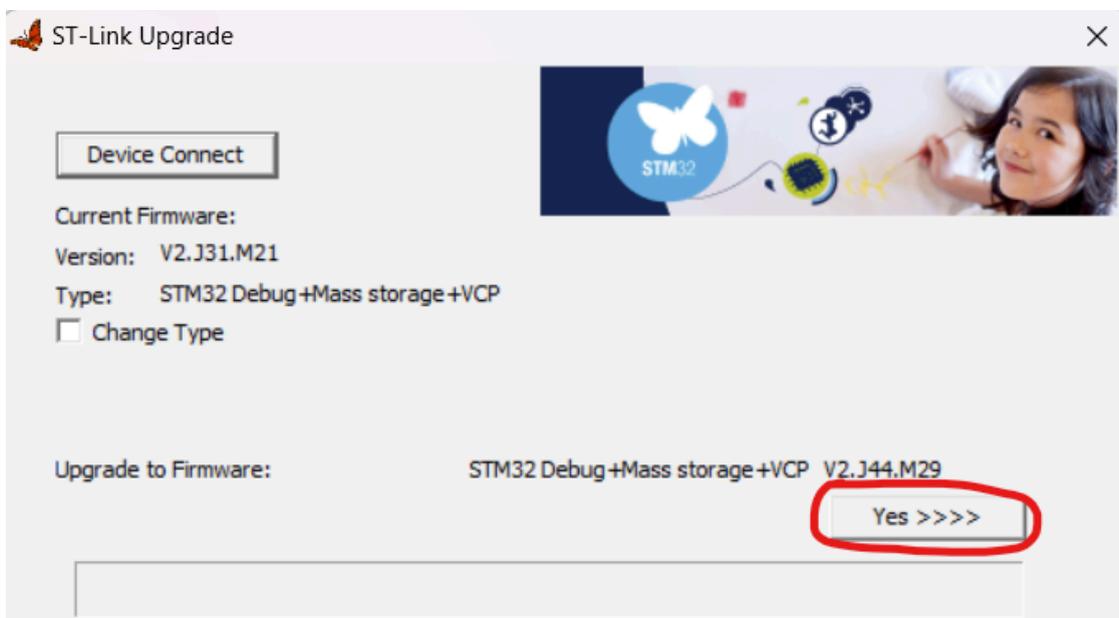
### Windowsでのファームウェアアップデート方法

ダウンロードしたファイルを展開し、stsw-linl007/windowsフォルダにあるST-LinkUpgrade.exeをダブルクリックで起動します。

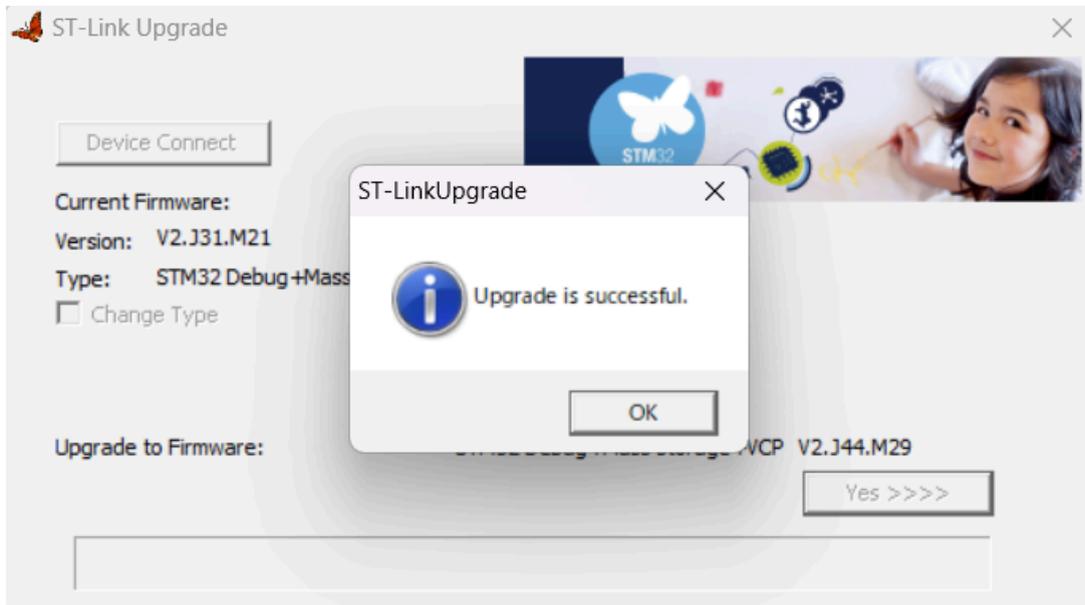
 ST-LinkUpgrade	2024/04/03 10:27	アプリケーション	1,253 KB
 STLinkUSBDriver.dll	2024/04/03 10:27	アプリケーション拡張	115 KB



NucleoボードとPCをUSBで接続し、電源スイッチをスライドした後、"Device Connect"をクリックします。



VersionやTypeが表示されたら"Yes>>>>"をクリックします。

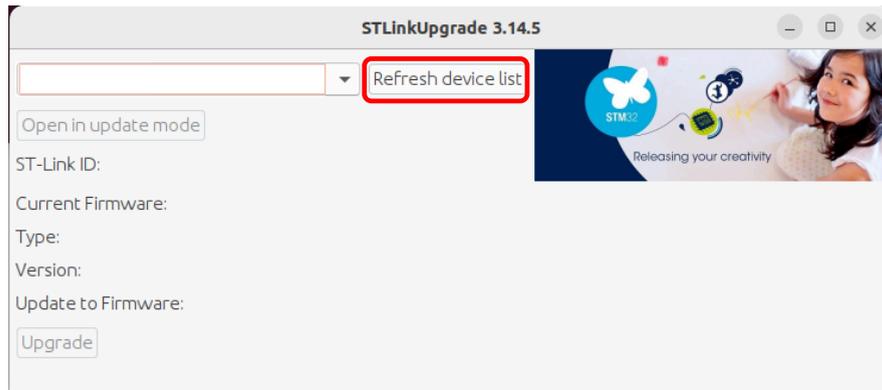


Upgrade is successfulと表示されたら、この画面を閉じて書き込みを再開します。

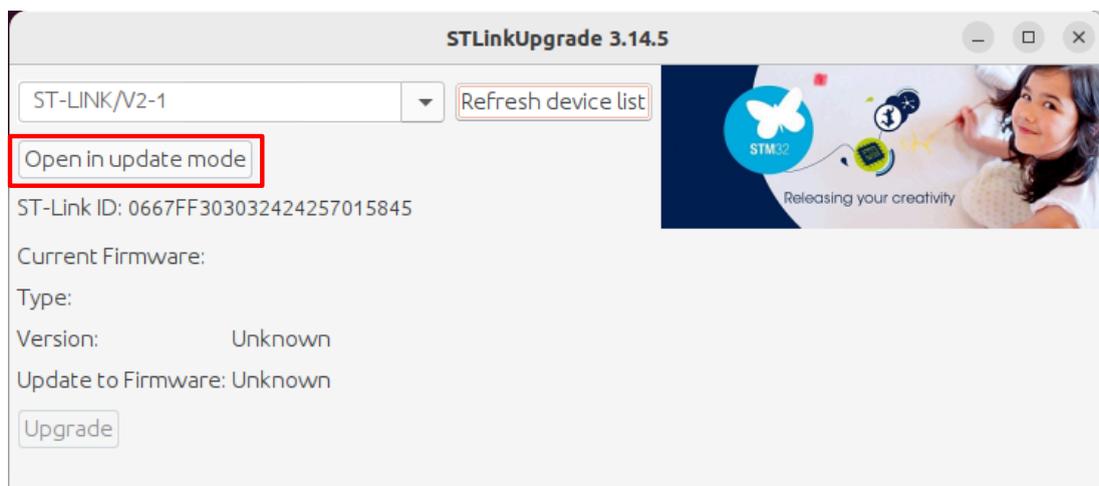
## Linux (Ubuntu) でのファームウェアアップデート方法

ダウンロードしたenstsw-link007-v3-15-6.zipをダブルクリックで展開します。ターミナルを起動し、展開したswst-link007/AllPlatformsに移動します。

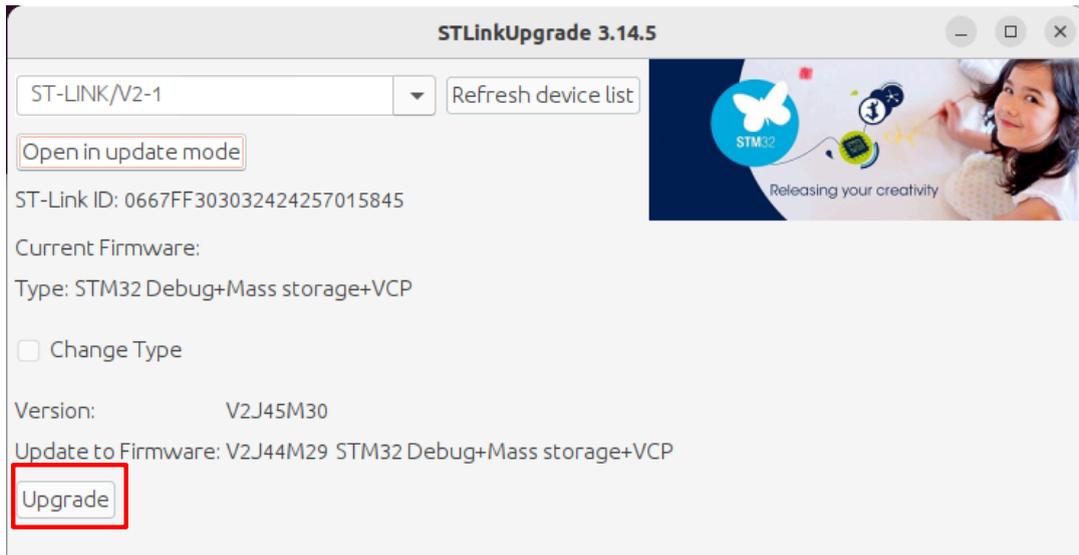
```
java -jar STLinkUpgrade.jar
```



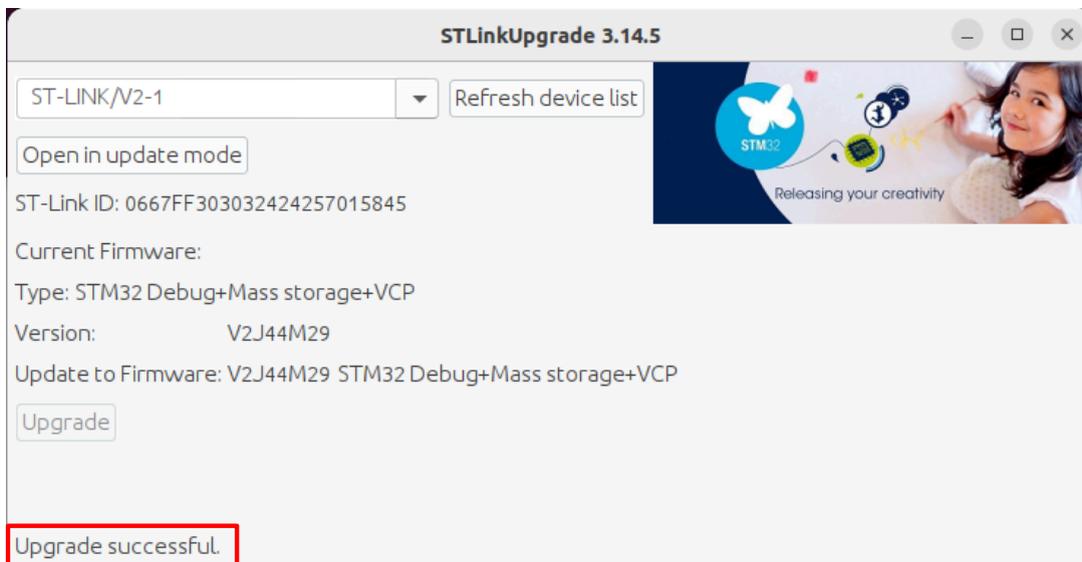
NucleoボードとPCをUSBで接続し、電源スイッチをスライドした後、"Refresh Device list"をクリックします。



"Open in update mode"をクリックします。



VersionやTypeが表示されたら"Upgrade"をクリックします。



"Upgrade is successful"と表示されたら、この画面を閉じて、書き込みを再開します。

## macOSでのファームウェアアップデート方法

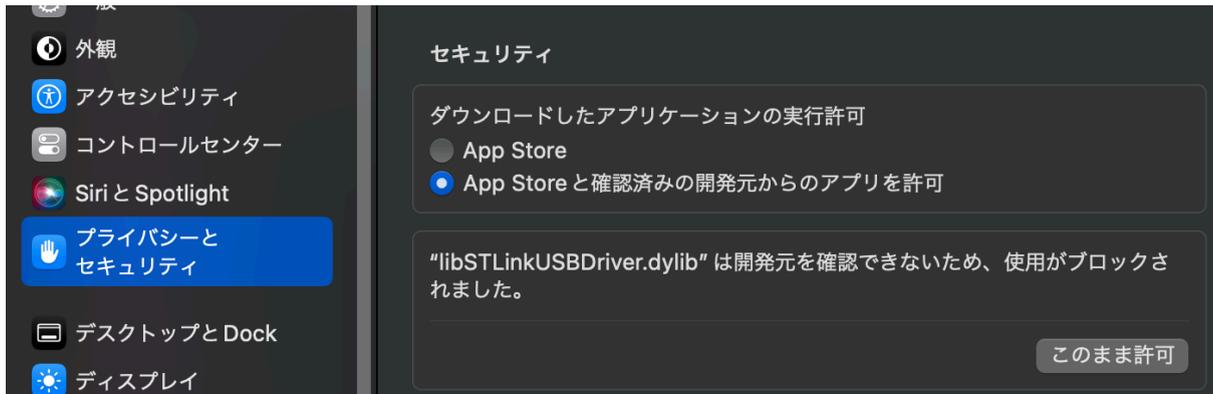
ダウンロードしたenstsw-link007-v3-15-6.zipをダブルクリックで展開します。ターミナルを起動し、展開したswst-link007/AllPlatformsに移動します。

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

このメッセージが表示された場合は、キャンセルを押します。



システムのプライバシーとセキュリティの設定項目で”このまま許可”をクリックします。





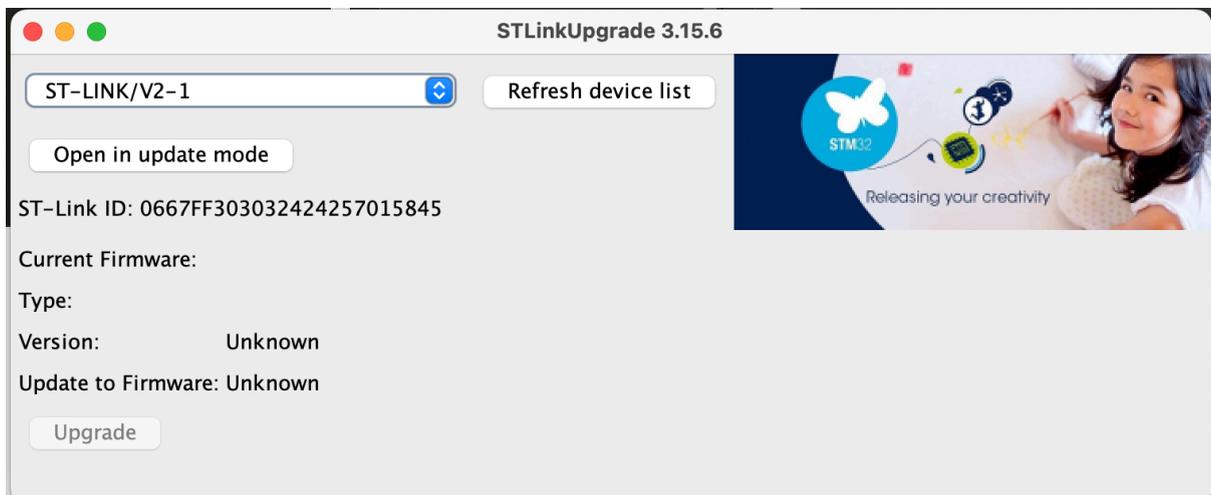
PCのユーザ名とパスワードを入力して設定を変更します。再度、ターミナルに戻り、javaのコマンドを実行します。



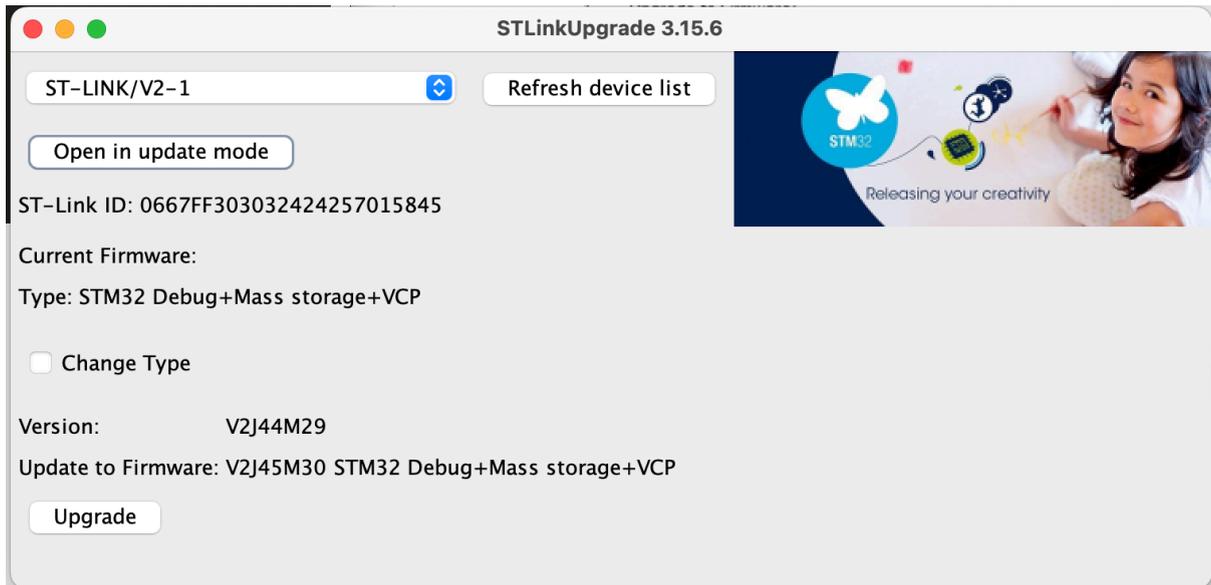
先程と似たようなメッセージが表示されますが、今度は“開く”をクリックします。



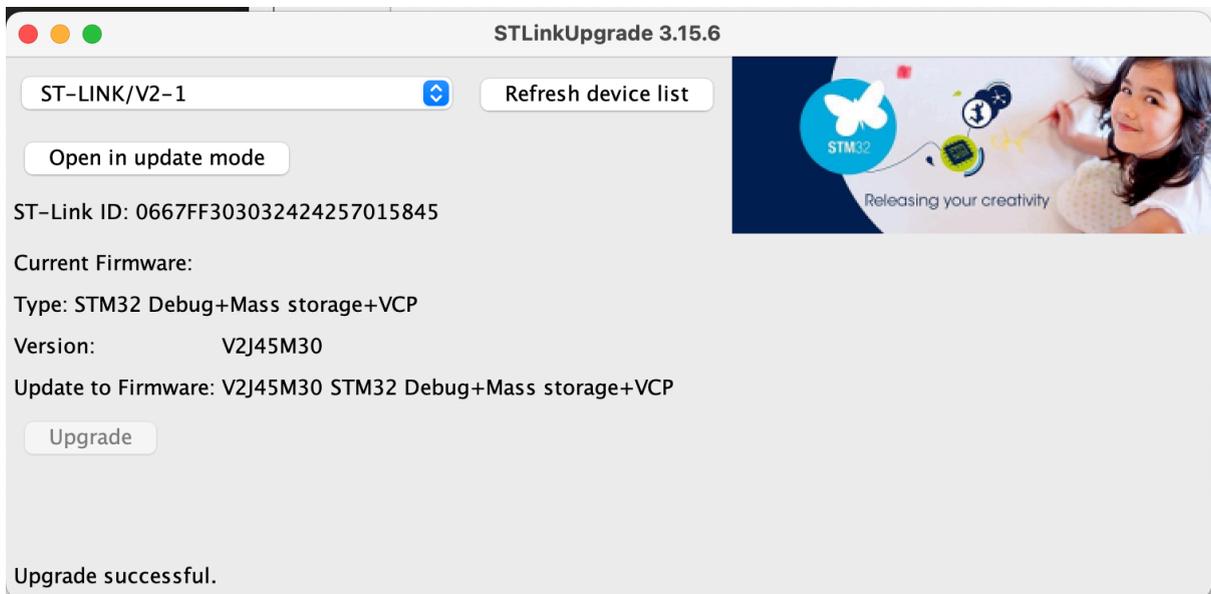
再びキャンセルし、システムのプライバシーとセキュリティの設定項目で”このまま許可”をクリックします。再度、ターミナルに戻り、javaのコマンドを実行します。



NucleoボードとPCをUSBで接続し、電源スイッチをスライドした後、"Open in update mode"をクリックします。"Open In update mode"をクリックできない時は、"Refresh device list"をクリックするとクリックできるようになります。



VersionやTypeが表示されたら"Upgrade"をクリックします。



"Upgrade is successful"と表示されたら、この画面を閉じて、書き込みを再開します。

## 使用しているツール、OSSのバージョン

本書で使用するツールやOSSのバージョンは以下のとおりです。

本書で使用するツールやOSSのバージョン情報

ツールやOSS	バージョン	URL
Arduino IDE	2.3.2	<a href="https://www.arduino.cc/en/software">https://www.arduino.cc/en/software</a>
Arduino core for the ESP32	3.0.1	<a href="https://github.com/espressif/arduino-esp32/releases">https://github.com/espressif/arduino-esp32/releases</a>
Arduino core support for STM32 based boards	2.7.1	<a href="https://github.com/stm32duino/Arduino_Core_STM32/releases">https://github.com/stm32duino/Arduino_Core_STM32/releases</a>
STM32CubeProgrammer	2.16.0	<a href="https://www.st.com/ja/development-tools/stm32cubeprog.html">https://www.st.com/ja/development-tools/stm32cubeprog.html</a>
STLink boards firmware upgrade	3.15.6	<a href="https://www.st.com/ja/development-tools/stsw-link007.html">https://www.st.com/ja/development-tools/stsw-link007.html</a>
STLink USB driver(windows only)	2.0.2	<a href="https://www.st.com/ja/development-tools/stsw-link009.html">https://www.st.com/ja/development-tools/stsw-link009.html</a>

## 改訂履歴

発行日 (YY/MM/DD)	版数	改訂内容
24/06/12	1.0	新規作成
24/09/20	1.1	Training Tracer Ver.2のGitHubリポジトリのURLに更新

## Copyright・知的財産権について

弊社は、本製品および本製品に関連して弊社が制作したソースファイル、ディレクトリ、実行可能ファイル、データ、開発ツールおよびその他の資料（以下「弊社資料」といいます）にかかる特許権、実用新案権、意匠権、著作権、ノウハウ、その他の技術および知的財産に関する一切の権利を留保するものとします。本書は、弊社の商標、商号、役務商標、商品名、ロゴの使用を許諾するものではありません。ただし、本製品および弊社資料の説明または記述に合理的に必要な範囲において当該商標等を使用する場合は、この限りでないものとします。なお、本製品および弊社資料に付された商品識別番号、商標、登録商標、コピーライト、その他の注意事項は除去しないようお願いします。

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.