# 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<u>https://www.arduino.cc/en/software</u>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<u>https://www.arduino.cc</u>にアクセスしSOFTWAREのタブをクリックします。



<u>https://www.arduino.cc</u> 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 AppImage 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 2.3.2

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

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

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

SOURCE CODE

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

#### DOWNLOAD OPTIONS

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

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

macOS Intel, 10.15: "Catalina" or newer, 64 bits macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

Release Notes

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**。

PROFESSIONAL	EDUCATION	STORE			Q Search or	n Arduino.cc		$\hat{\mathbf{x}}$	SIGN IN
⊙⊙	HARDWARE	SOFTWARE	CLOUD	DOCUMENTATION		BLOG	ABOUT		
	П	ownloa	d Ardui	no IDE & si	nnort its nr	ngress	:		
	Since	the 1.x releas	e in March 2	015, the Arduino ID	has been download	ded 77,959,	,898		
		times	— impressiv	ve! Help its developr	nent with a donatior	1.			
		\$3	\$5 \$	\$10 \$25	\$50 Othe	r			
			CON	ITRIBUTE AND DOW	NLOAD				
				JUST DOWNLOA	0				
					Q				
			ø		R ( · · )				
					50				
			E	$\Theta$					
			<u> </u>						
			Learr	n more about <mark>donating to</mark>	Arduino.				

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。

Arduino products through our <b>weekly newsletter</b> !
il *
nfirm to have read the <b>Privacy Policy</b> and to accept the <b>Terms of</b> vice *
ould like to receive emails about special deals and commercial offers from uino.
SUBSCRIBE & DOWNLOAD
or
JUST DOWNLOAD



#### 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。

Arduino IDE	
Please wait	while Windows configures Arduino IDE
	Cancel

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。

Q arduino IDE	
< <b>ਗ਼</b> ੑੑੑੑੑੑੑ	ドキュメント ウェブ 設定 フォルター 写
最も一致する検索結果	
Arduino IDE שלת	
アプリ	

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。

🔤 ske	tch_may10b   A	rduino IDE 2.3.2	-		×
File Ed	lit Sketch To	ools Help			
Ø	€ 🔊	Select Board 👻		∿	· <b>O</b> ··
	sketch_may1	10b.ino			
1	1 vo 2 3	<pre>pid setup() {     // put your setup code here, to run once:     // put your setup code here, to run once:</pre>			
M	4 } 5	hid loss() (			
0	7 8 0 1	// put your main code here, to run repeatedly:			
Q	10				
8			lo board	alactar	
	and the second		lo board :		<b>~</b>

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。

im Windows セキ:	ュリティの重要な警告		×
このア ていま	プリの機能のいく す	つかが Windows Defender ファイアウォールでブロックされ	
すべてのパブリック: の機能のいくつかが	ネットワークとプライベー グロックされています。	トネットワークで、Windows Defender ファイアウォールにより Arduino IDE	
$\sim$	名前(N):	Arduino IDE	
	発行元(P):	Arduino SA	
	パス(H):	C:¥users¥maaok¥appdata¥local¥programs¥arduino-ide¥arduino ide.exe	
Arduino IDE にこれ	いらのネットワーク上での	D通信を許可する:	
□ プライベート	ネットワーク (ホームネ	ットワークや社内ネットワークなど)(R)	
☑ パブリック オ (このようなオ	ヘットワーク(空港、喫 ネットワークは多くの場	を店など) (非推奨)(U) 合、 セキュリティが低いかセキュリティが設定されていません)	
<u>アプリにファイアウォ-</u>	-ルの経由を許可する	ことの危険性の詳細	
		♥アクセスを許可する(A) キャンセル	
			4

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。

ダウ	ンロード ×	
Ç	X	
Children and Child	開く	Return
arduino-ide_2. 64bit.App	別のアプリケーションで開く( <u>A</u> )	
	切り取り( <u>T</u> )	Ctrl+X
	コピー( <u>C</u> )	Ctrl+C
	指定先へ移動	
	指定先にコピー	
	ゴミ箱へ移動する(⊻)	Delete
	名前を変更( <u>M</u> )	F2
	压縮( <u>O</u> )	
	星を付ける	
	プロパティ ( <u>R</u> )	Ctrl+I

AppImageファイルを右クリックした状態

arduino-ide_264t	oit.AppImage のプロパティ 🛛 🗙
基本	アクセス権
所有者(O)	自分
アクセス	読み書き ~
グループ(G)	ubuntu ~
アクセス	読み書き ~
その他	
アクセス	読み取り専用 〜
実行	✔ プログラムとして実行可能(E)
セキュリティコンテキスト	不明

プロパティのアクセス権タブを開いた状態



その後、AppImage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

AppImage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、 AppImage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。<u>https://github.com/AppImage/AppImageKit/wiki/FUSE</u>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。

	Settings Network	
Sketchbook location:		
c:\Users\maaok\OneDrive\Do	cuments\Arduino	BROWSE
Show files inside Sketches		
Editor font size:	14	
Interface scale:	✓ Automatic 100 %	
Theme:	Light	
Language:	日本語 V (Reload required)	
Show verbose output during	🗹 compile 🗌 upload	
Compiler warnings	None 💙	
Verify code after upload		
Auto save		
Editor Quick Suggestions		
Additional boards manager UI	(LS:	64
	CANCE	с ок
Additional boards manager UI	CANCE	С ОК

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。下記手順に従ってお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。

基本設定				×
		設定 ネットワーク		
スケッチブックの場所:				
c:\Users\maaok\OneDrive\Docu	ments\Arduino		参照	照
□スケッチ内のファイルを表示				
エディターのフォントサイズ:	14			
インターフェイスのスケール:	☑ 自動 100	%		
配色テーマ:	Light	~		
エディター言語:	日本語	✓ (Reload required)		
より詳細な情報を表示する	マコンパイル(	□書き込み		
コンパイラの警告	なし 🗸			
✔ 書き込み後にコードを検証す	3			
✓ 自動保存(U)				
□エディターのクイックサジェ	スト			
追加のボードマネージャのURL:				<b>P</b>
				01
			(TF)EN ON	5

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\_ind ex.json

を追加し"OK"をクリックします。



基本設定		×
	設定 ネットワーク	
スケッチブックの場所:		
c:\Users\maaok\OneDrive\Docu	uments\Arduino 参照	
□スケッチ内のファイルを表示	ŧ.	
エディターのフォントサイズ:	14	
インターフェイスのスケール:	☑ 自動 100 %	
配色テーマ:	Light 🗸	
エディター言語:	日本語 V (Reload required)	
より詳細な情報を表示する	✔ コンパイル □ 書き込み	
コンパイラの警告	なし 🗸	
✔ 書き込み後にコードを検証す	3	
✓ 自動保存(U)		
<ul> <li>□エティターのクイックサンコ</li> <li>追加のボードフォージャのUPL</li> </ul>	https://raw.aithubusarcaptapt.com/asprassif/arduipa.asp32/ab.pagas/pagkaga	
」」「100/10-1、くハーン1/000RL	https://aw.glinubusercontent.com/espressi/ardunio-espsz/gir-pages/package	

追加ボードマネージャーの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。

🔤 ske	tch_may10b   Ar	duino IDE 2.3.2			
ファイル	編集 スケッチ	ツール ヘルプ(H)			
	<b>→</b> 🔊 [	自動整形	Ctrl+T		
_	ボードフネー・	スケッチをアーカイブする			
	atm22	ライブラリを管理	Ctrl+Shift+I		un() {
	stm52	シリアルモニタ	Ctrl+Shift+M		your setup code here, to r
빈	タイプ: 全	シリアルプロッタ			
19.0		Firmware Updater			
ШA		SSLルート証明書を書き込み			<pre>&gt;p() {</pre>
~		ボード		٠	ボードマネージャ Ctrl+Shift+B <sup>p ru</sup>
0		ポート		•	
0		ボ−ド情報を取得			
Q		ブートローダを書き込む			

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。

🔤 sket	tch_may10a   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	•
	ボードマネージャ	sketch_may10a.ino
	esp32	1 ∨ void setup() {
<u>۴</u>	タイプ:全て 🗸	2 // put your setup code here, to run once: 3
_		4 }
ITh	Arduino ESP32 Boards	5 Characterist Jacob C
000	by Arduino	7 // put your main code here, to run repeatedly:
0	Boards included in this package: Arduino Nano ESP32	8
0	詳細情報	9 }
0	2.0.13 🗸	10
X	インストール	
	esp32 by Espressif	
	Boards included in this package:	
	ESP32 Dev Board, ESP32-S2 Dev	
	Board, ESP32-S3 Dev Board, 詳細情報	
	301 ¥	
	インストール	
		*

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<u>https://github.com/rt-net/pico\_micro\_ros\_arduino\_examples</u>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**。

main 🔹 🥲 3 Branches 🛇 2 Tags	Q	Go to file	t A	dd file  ▼	Code 👻
<b>ShotaAk</b> STEP12 MapManagerのリファクタ	リング (#17) 🚥 🗸	Local		Codespaces	
github/workflows	STEP10~13をSTEF	P Clone			?
uROS_STEP10_tfMsg	ファイル名をinte	n HTTPS SSH	GitHub CLI		
uROS_STEP11_SensorMsg	STEP10~13をSTEP	git@github.com:	rt-net/pico_mi	cro_ros_arduino_e>	C
uROS_STEP12_micromouse	STEP12 MapMana	Use a password-p g	rotected SSH k	ey.	
uROS_STEP1_LED	CI(こLint(こよるコー	_ 🖞 Open with Git	Hub Desktop		
uROS_STEP2_SWITCH	リファクタリング	Open with Visual	Studio		
uROS_STEP3_Buzzer	リファクタリング				
UROS STEPA Sansor	リファクタリング	(a) Download ZIP	·		

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。

チ	ツール ヘルプ(H)		
Γ	自動整形	Ctrl+T	
	スケッチをアーカイブする		
×-:	ライブラリを管理	Ctrl+Shift+I	
	シリアルモニタ	Ctrl+Shift+M	once:
全	シリアルプロッタ		
	Firmware Updater		_
o E no	SSLルート証明書を書き込み		
nclu	ボード: "ESP32S3 Dev Module"	)	<pre>epeatedly:</pre>
Nar	ポート	)	
~	ボード情報を取得		
27	USB CDC On Boot: "Disabled"	)	<ul> <li>Disabled</li> </ul>
	CPU Frequency: "240MHz (WiFi)"	)	Enabled
	Core Debug Level: "None"	,	
	USB CDC On BootをEnable	d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。

£	ツール ヘルプ(H)		
Г	自動整形	Ctrl+T	
h	スケッチをアーカイブする		
	ライブラリを管理	Ctrl+Shift+I	
	シリアルモニタ	Ctrl+Shift+M	once:
全	シリアルプロッタ		
	Firmware Updater		-
0 E	SSLルート証明書を書き込み		
clu	ボード: "ESP32S3 Dev Module"	•	<pre>&gt;peatedly:</pre>
lar	ポート	•	
	ボード情報を取得		
7	USB CDC On Boot: "Enabled"	•	-
	CPU Frequency: "240MHz (WiFi)"	+	
	Core Debug Level: "None"	•	
7 E	USB DFU On Boot: "Disabled"	•	
ta	Erase All Flash Before Sketch Upload: "Disabled"	•	
clu	Events Run On: "Core 1"	•	
<u>(</u> )	Flash Mode: "QIO 80MHz"	•	
ESI	Flash Size: "4MB (32Mb)"	+	
_	JTAG Adapter: "Disabled"	•	
-	Arduino Runs On: "Core 1"	•	
	USB Firmware MSC On Boot: "Disabled"	+	
	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"	•	
	PSRAM: "Disabled"	+	
	Upload Mode: "UART0 / Hardware CDC"		
	Upload Speed: "921600"	•	
	USB Mode: "Hardware CDC and JTAG"	+	✓ Hardware CDC and JTAG
	書き込み装置	+	USB-OTG (TinyUSB)

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。

sketch_may10b   Arduino IDE 2.3.2								
ファイル 編集 スケッチ	ツール ヘルプ	(H)						
新規スケッチ	Ctrl+N	odu	ule		-			
新規クラウドスケッチ	Alt+Ctrl+N							
開く	Ctrl+O			sketch_m	ay10b.	ino		
最近使った項目を聞く		•		1	void	setup(		
				2	//	put yo		
スケッチフック		•		3				
スケッチ例		•		4	}			
				5				
閉じる	Ctrl+W			6	void	loop()		
Save	Ctrl+S	e:		7	//	put yo		
名前を付けて保存	Ctrl+Shift+S			8				
HINGETSO CINIST	carronnero			9	}			

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.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。

💿 ske	etch_jun3b   Ardu	ino IDE 2.3.2			ESP32H2 Dev Module
ファイル	編集 スケッチ	ツール ヘルプ(H)			ESP32C6 Dev Module
	€ 🔊	自動整形	Ctrl+T		ESP32S3 Dev Module
	ボードフネー・	スケッチをアーカイブする			ESP32C3 Dev Module
	//=  · <4<	ライブラリを管理	Ctrl+Shift+I	up() (	ESP32S2 Dev Module
	espoz	シリアルモニタ	Ctrl+Shift+M	your setup code here, t	O ESP32 Dev Module
也	タイプ: 全	シリアルプロッタ			ESP32-WROOM-DA Module
		Firmware Updater			ESP32 Wrover Module
	by Arduino E	SSLルート証明書を書き込み		)p() {	ESP32 PICO-D4
1	Boards inclu	ボード	÷	ボードマネージャ Ctrl+Shift+B	ESP32S3 Dev Module Octal (WROOM2)
0	Arduino Nar 詳細情報	ポート	+	esp32	ESP32-S3-Box
~	2013 ¥	ボード情報を取得			ESP32-S3-USB-OTG
Q	177	ブートローダを書き込む			ESP32S3 CAM LCD
					ESP32S2 Native USB

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。

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。

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	≣× 6
<pre>pinMode(LED4,0UTPUT);</pre>	
<pre>/Users/aokimasatake/Documents/Arduino/ukos_STEP1/ukos_STEP1.ino:12:11: note: pinMode(LED4,OUTPUT);</pre>	suggeste
exit status 1	
Compilation error: 'LED4' was not declared in this scope	

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。

Linux(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. 本製品の電源スイッチをOFFにします。
- 2. 本製品にバッテリを接続します。
- 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。

他のボードとポートを選択			×
スケッチを書き込みたい場合には、ボードと ボードのみを選択した場合、コンパイルはて	ヒポートの戸 できますが、	両方を選択してください。 スケッチの書き込みはできません。	
ボード		ポート	
dev	٩		
ESP32H2 Dev Module		COM4 Serial Port (USB)	~
ESP32S2 Dev Module	- 1	COM1 Serial Port	
ESP32S3 Dev Module	~		
ESP32S3 Dev Module Octal (WROOM2)			
Hornbill ESP32 Dev			
Lion:Bit Dev Board		□ ☆ てのポートをまこ	
		(+r>trl)	K(O)

#### ボードとポートの設定

設定完了後、ボート名の左に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。





## 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\_ind ex.json

を追加し"OK"をクリックします。



		ſ	設定	ネットワーク
スケッチブックの場所:				
c:\Users\maaok\OneDrive\Docu	ments\Arc	luino		参照
□スケッチ内のファイルを表示				_
エディターのフォントサイズ:	14			
インターフェイスのスケール:	✔ 自動	100	%	
配色テーマ:	Light		~	
エディター言語:	日本語		~	(Reload required)
より詳細な情報を表示する	✓ コン/	(่าน 🗆	書き	込み
コンパイラの警告	なし	~		
✓ 書き込み後にコードを検証す	3			
▲ 日勤休行(0) □エディターのクイックサジェ	スト			
追加のボードマネージャのURL:	https://ra	w.github	usero	ontent.com/espressif/arduino-esp32/gh-pages/package
	-	-		

追加ボードマネージャーの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。

🔤 ske	tch_may10b   A	rduino IDE 2.3.2			
ファイル	編集 スケッチ	ツール ヘルプ(H)			
		自動整形	Ctrl+T		
		スケッチをアーカイブする			
		ライブラリを管理	Ctrl+Shift+I		
	stm32	シリアルモニタ	Ctrl+Shift+M		<pre>:up() { : vour setup code here, to r</pre>
힘	タイプ: 全	シリアルプロッタ			,,,,,,
		Firmware Updater			
		SSLルート証明書を書き込み			)p() {
~		ボード		•	ボードマネージャ Ctrl+Shift+B <sup>p ru</sup>
0		ポート		۲	
0		ボード情報を取得			
X		ブートローダを書き込む			

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。

DM2)

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<u>https://github.com/rt-net/pico\_v2\_arduino\_examples</u>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**。

☐ rt-net / pico_v2_arduino_exar	mples Public	🗘 Notifi
<> Code 🕢 Issues 1 🕴 Pull requ	uests 🕞 Actions 🖽 Projects 😲 Security 🗠 Insigh	L
្ងះ main 👻 🐉 1 Branch 📀 0 Tags	Q Go to file	ode -
SusukeKato README(Carduino-esp32)	2のバージョ Clone	?
github/workflows	micrc GitHub CLI	
pico_v2_STEP1_LED	https://github.com/rt-net/pico_v2_arduino_exam,	Q
pico_v2_STEP2_SWITCH	Clone using the web URL. サン	
pico_v2_STEP3_Buzzer	サン: 🖞 Open with GitHub Desktop	
pico_v2_STEP4_Sensor	Download ZIP	
pico_v2_STEP5_Straight	micro step 1/4 -> 1/8 change (#4) 2 day	/s ago

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。

ツール ヘルプ(H)		
自動整形	Ctrl+T	
スケッチをアーカイブする		
ライブラリを管理	Ctrl+Shift+I	
シリアルモニタ	Ctrl+Shift+M	once:
シリアルプロッタ		
Firmware Updater		
SSLルート証明書を書き込み		
ボード: "ESP32S3 Dev Module"		<pre>epeatedly:</pre>
ポート		×
ボード情報を取得		
USB CDC On Boot: "Disabled"		► ✓ Disabled
CPU Frequency: "240MHz (WiFi)"		Enabled
Case Dahua Laush "Nasa"		

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。

£	ソール ヘルプ(H)		
Γ	自動整形	Ctrl+T	
	スケッチをアーカイブする		
	ライブラリを管理	Ctrl+Shift+I	
	シリアルモニタ	Ctrl+Shift+M	once:
全	シリアルプロッタ		
	Firmware Updater		
0	SSLルート証明書を書き込み		
clu	ボード: "ESP32S3 Dev Module"	•	epeatedly:
lar	ポート	+	
	ボード情報を取得		
7	USB CDC On Boot: "Enabled"	Þ	~
	CPU Frequency: "240MHz (WiFi)"	•	
	Core Debug Level: "None"	•	
í E	USB DFU On Boot: "Disabled"	•	
ta	Erase All Flash Before Sketch Upload: "Disabled"	•	
sta ch	Events Run On: "Core 1"	•	
	Flash Mode: "QIO 80MHz"	•	
ESI	Flash Size: "4MB (32Mb)"	•	
	JTAG Adapter: "Disabled"	•	
-	Arduino Runs On: "Core 1"	•	
	USB Firmware MSC On Boot: "Disabled"	•	
	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"	•	
	PSRAM: "Disabled"	•	10
	Upload Mode: "UART0 / Hardware CDC"	+	
	Upload Speed: "921600"		
	USB Mode: "Hardware CDC and JTAG"	+	✓ Hardware CDC and JTAG
	書き込み装置	+	USB-OTG (TinyUSB)

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。

🔤 sketch_may10b   A	rduino IDE 2.3.	2		
ファイル 編集 スケッチ	ツール ヘルプ	(H)		
新規スケッチ	Ctrl+N	odule		-
新規クラウドスケッチ	Alt+Ctrl+N	_	1.1.1.1	
開<	Ctrl+O		sketch_n	nay10b.ino
			1	<pre>void setup(</pre>
東近使つに項目を開く			2	// put yo
スケッチブック		•	3	
スケッチ例			4	}
~~~~			5	
閉じる	Ctrl+W		6	<pre>void loop()</pre>
Save	Ctrl+S	e:	7	// put yo
名前を付けて保存	Ctrl+Shift+S		8	
	car on o		9	}

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。

移動 X

ファイル"pico\_v2\_STEP1\_LED.ino"は、"pico\_v2\_STEP1\_LED"という名前のスケッチフォルダの中にあることが必要です。

このフォルダを作成し、ファイルを移動させ、継続しますか?

→ キャンセル

 $\rightarrow OK(O)$ 

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。



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。

🧧 ske	etch_jun3b   Ardu	ino IDE 2.3.2				ESP32H2 Dev Module
ファイル	編集 スケッチ	ツール ヘルプ(H)				ESP32C6 Dev Module
	€ 🔊	自動整形	Ctrl+T			ESP32S3 Dev Module
	ボードフネー・	スケッチをアーカイブする				ESP32C3 Dev Module
		ライブラリを管理	Ctrl+Shift+I	() ()		ESP32S2 Dev Module
	espoz	シリアルモニタ	Ctrl+Shift+M	: your setup code he	ere, to	ESP32 Dev Module
친	タイプ: 全	シリアルプロッタ				ESP32-WROOM-DA Module
		Firmware Updater				ESP32 Wrover Module
	by Arduino E	SSLルート証明書を書き込み		)p() {	_	ESP32 PICO-D4
-	Boards inclu	ボード	)	ボードマネージャ Ctrl+Shift	t+B	ESP32S3 Dev Module Octal (WROOM2)
0	Arduino Nar 詳細情報	ポート	)	esp32	•	ESP32-S3-Box
~	2013	ボード情報を取得			-	ESP32-S3-USB-OTG
Q	2.0.13	ブートローダを書き込む				ESP32S3 CAM LCD
						ESP32S2 Native USB

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。

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。

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 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。

TRT CORPORATION

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。

他のボードとポートを選択			×
スケッチを書き込みたい場合には、ボー ボードのみを選択した場合、コンパイルI	ドとポートのia はできますが、	両方を選択してください。 スケッチの書き込みはできませ/	ív.
ボード		ポート	
dev	Q		
ESP32H2 Dev Module		COM4 Serial Port (USB)	*
ESP32S2 Dev Module	- 1 B	COM1 Serial Port	
ESP32S3 Dev Module	~		
ESP32S3 Dev Module Octal (WROOM2	2)		
Hornbill ESP32 Dev			
Lion:Bit Dev Board		□全てのポートを表示	
		( <b>‡</b> r>t	сль Ок(о)

ボードとポートの設定



設定完了後、ボート名の左にUSBのマーク Control Control

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。





## 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。



基本設定		×
	設定 ネットワーク	
スケッチブックの場所:		
c:\Users\maaok\OneDrive\Docu	ments\Arduino	照
□ スケッチ内のファイルを表示		
エディターのフォントサイズ:	14	
インターフェイスのスケール:	☑ 自動 100 %	
配色テーマ:	Light 🗸	
エディター言語:	日本語    (Reload required)	
より詳細な情報を表示する	✓ コンパイル □書き込み	
コンパイラの警告	なし 🗸	
✔ 書き込み後にコードを検証す	3	
✓ 自動保存(U)		
しエティターのクイックサンエ 迫加のボードマネージャのUDL:	AN	-
追加のパートマネーンヤのORL.	nups.//gunub.com/sunszduino/boardivanagerriles/raw/main/package_sunicroei	γ.
	(+r>zul) OK	(0)

追加ボードマネージャーの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。

🔤 ske	tch_may10b   Ar	duino IDE 2.3.2			
ファイル	編集 スケッチ	ッール ヘルプ(H)			
	🖨 🚯 🛽	自動整形	Ctrl+T		
		スケッチをアーカイブする			
	ホートマネー:	ライブラリを管理	Ctrl+Shift+I		
	stm32	シリアルモニタ	Ctrl+Shift+M		:up() {
包	タイプ: 全	シリアルプロッタ			. your secup code here, com
		Firmware Updater			
		SSLルート証明書を書き込み			)p() {
0		ボード		•	ボードマネージャ Ctrl+Shift+B
0		ポート		۲	
0		ボード情報を取得			
X		ブートローダを書き込む			

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。

			白動整形	Ctrl+T	1				NUCCO OT
	7 8		コッシューク	curr					Nucleo-32
P	sketch_n	nar2	入り ツナを アール1 ノ 9 つ						Discovery
	1	vc	フイノフリを官理	Ctrl+Shift+I				1	Eval
-	2		シリアルモニタ	Ctrl+Shift+M		n once:			STM32MP1 series coprocessor
1_)	3	2	シリアルプロッタ						Generic STM32C0 series
-	4	}	Firmware Updater						Generic STM32F0 series
	6	vo	SSLルート証明書を書き込み						Generic STM32F1 series
	7		ボード		•	ボードマネージャ	Ctrl+Shift+B		Generic STM32E2 series
$\oslash$	8	1	ポート		F.	Arduino AVR Boards			Generic STM32F3 series
	10	1	ボード情報を取得			Arduino Mbed OS Opta Boards			Generic STM22E4 series
Q			ゴートローグを書き込む			STM32 MCII based boards		•	Conoris STM32F4 series
			7-1-1-9を書き込む		-	STWISZ WCO based boards			Generic STMSZF7 Series
									Generic STM32G0 series
									Generic STM32G4 series
									Generic STM32H5 series
									Generic STM32H7 Series
									Generic STM32L0 series
									Generic STM32L1 series
									Generic STM32L4 series
									Generic STM32L5 series
									Generic STM32U5 series

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



次にNuleo-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。

🔤 ske	etch_may10a	a   Arc	duino IDE 2.3.2					
ファイル	編集 スケ	ッチ	ツール ヘルプ(H)					
Ø	€			Ctrl+T				
	sketch n	nay1	人ケッナをアーカイノする					
	1	vo	ライブラリを管理	Ctrl+Shift+I				
	2		シリアルモニタ	Ctrl+Shift+M				
1	3		シリアルプロッタ					
-	4	}	Firmware Updater					
	6	va	SSLルート証明書を書き込み					
	7		ボード: "Nucleo-32"		►			
÷>	9	2	ポート		►			
	10	1	ボード情報を取得					
Q			Debug symbols and core logs: "None"		►			
			Optimize: "Smallest (-Os default)"		•			
			Board part number: "Nucleo F031K6"		+	~	Nucleo F031K6	
			C Runtime Library: "Newlib Nano (default)"		×		Nucleo F042K6	
			Upload method: "Mass Storage"		+		Nucleo F303K8	
			USB support (if available): "None"		×.		Nucleo G031K8	
			U(S)ART support: "Enabled (generic 'Serial')"		×.		Nucleo G431KB	
			USB speed (if available): "Low/Full Speed"		×		Nucleo L031K6	
			ブートローダを書き込む				Nucleo L412KB	
							Nucleo L432KC	

ツール->Board part Number->Nucleo F303K8を選択 以上で、STM32 Nucle 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Nuleo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<u>https://www.st.com/ja/development-tools/stm32cubeprog.html</u> 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。

	製品型番	概要	ŀ	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

#### ソフトウェア入手



ライセンス契約ついて確認し"承諾します"をクリックします。

<ul> <li>Please indicate your acceptance or NON-acceptance by self ACCEPT" as indicated below in the media.</li> <li>BY INSTALLING COPYING, DOWNLOADING, ACCESSING SOFTWARE PACKAGE OR ANY PART THEREOF (AND TH DOCUMENTATION) FROM STMICROELECTRONICS INTE BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICR RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR EN BOUND BY THIS SOFTWARE PACKAGE LICENSE AGREI</li> <li>Under STMicroelectronics' intellectual property rights and su terms for any third-party software incorporated in this software Source Terms (as defined here below), the redistribution, rep binary forms of the software package or any part thereof, will permitted provided that the following conditions are met:</li> <li>Redistribution of source code (modified or not) must r list of conditions and the following disclaimer.</li> <li>Redistributions in binary form, except as embedded in microprocessor device manufactured by or for STMicrophysical and the following the package or for STMicrophysical and the following by more for STMicrophysical and the following disclaimer.</li> </ul>	lecting "I ACCEPT" or "I DO NOT G OR OTHERWISE USING THIS HE RELATED ERNATIONAL N.V, SWISS ROELECTRONICS), THE ON BEHALF OF ANY ENTITY IGAGED AGREES TO BE EMENT. ubject to applicable licensing are package and applicable Open production and use in source and th or without modification, are retain any copyright notice, this into microcontroller or croelectronics or a software avariable tastion, this lice of
<ol> <li>Redistribution of source code (modified or not) must r list of conditions and the following disclaimer.</li> <li>Redistributions in binary form, except as embedded in microprocessor device manufactured by or for STMic undate for such device, must reproduce the above co</li> </ol>	retain any copyright notice, this nto microcontroller or croelectronics or a software appyright notice, this list of
Additional License Terms for STM32CubeProgrammer 2.1	6.0

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	新規に登録される方
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	mySTのパーソナライズ機能
メールアドレス	<ul> <li>o イベントへの参加</li> <li>o ニュースレター配信</li> <li>o オンライン・サポート利用</li> </ul>
パスワード	<ul> <li>◦ STコミュニティへの参加</li> <li>◦ オンライン設計ツール利用</li> <li>◦ ソフトウェア・ダウンロード</li> </ul>
□ メールアドレスを記憶 ■	<ul> <li>・無償サンプル注文</li> <li>・製品情報(毎週更新)</li> <li>・ST製品 &amp; ツール購入</li> </ul>
<b>ログイン</b> パスワードを忘れた場合	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。

?
"SetupSTM32CubeProgrammer- 2.16.0" はインターネットから グロンロードさたたマプリケーション
マリンロートさ <i>れにアフリケーション</i> です。開いてもよろしいですか?
このファイルは "Safari" により今日の13:24 にダウンロードされました。 Appleによるチ ェックで悪質なソフトウェアは検出されません でした。
キャンセル 開く
"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。



#### Arduino開発環境構築マニュアル





M STM32CubeProgr	ammer Installation Wizard	- 0 X
STM32CubeProgramm Step 9 of 9	er Installation done	STM32 CubeProgrammer
Becognered	Partalistion has completed successfully. An universitier will be created in CKMrgram TilenST Microbiologica/STM320.de/STM320.de/Programme/W.binstaller	
STMicroelectronics		C Done

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ー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 Virtural 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

<u>https://www.st.com/ja/development-tools/stsw-link009.html</u>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。

ST-Link D	ebugのプロパティ		×			
全般	ドライバー 詳細	イベント				
2	ST-Link Debug					
	デバイスの種類:	ほかのデバイス				
	製造元:	不明				
	場所:	0004.0000.0003.003.000.000.000.000.000				
デバイ	スの状態					
ζσ	とのデバイスのドライバーがインストールされていません。(コード 28) ^					
ζa	デバイスと互換性のあ	るドライバーがありません。				
この。	ラデバイス用のドライバー	を検索するには、 [ドライバーの更新] をクリックしてくださ	~			
		ドライバーの更新(U)				
		OK +7	ンセル			

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。

		×
÷	■ ドライバーの更新 - ST-Link Debug	
	コンピューター上のドライバーを参照します。	
	次の場所でドライバーを検索します:	
2	C¥Users¥ ¥Downloads¥en.stsw-link009 ~ 参照(R)	
	☑ サブフォルダーも検索する(I)	
	→ コンピューター 上の利用可能なドライバーの一覧から選択します(!)	
	この一覧には、デバイスと互換性がある利用可能なドライバーと、デバイスと同じカテゴリにあるすべてのドライバーが表示されます。	
	3 次へ(N) キャンセ	216
~~~	ペューター トのドライバーを弁照します "という両面で弁照をクリック」	4

"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。

TRT CORPORATION

 $\times$ 



"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。

ドライバーの更新 - ST-Link Debug
 ドライバーが正常に更新されました
 このデバイス用のドライバーがインストールされました:





RT CORPORATION

### サンプルスケッチの準備

webブラウザで<u>https://github.com/rt-net/TrainingTracer\_V2\_Samples</u> 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**。

main 👻 🥲 🕈 🕈 🖓 🕈 🖓 main 🗸	Q Go to fil	le t	Add file 👻 <> Code
YusukeKato サンプルプログラムをチェック	フするGitHub Workflowを追	Local	Codespaces
aithub/workflows	サンプルプログラム	▶ Clone	(
Arduino_Exercises	サンプルプログラム	HTTPS SSH GitHu	b CLI
Arduino_Sample	サンプルプログラム	git@github.com:rt-net/	CYTrainingTracer_V2_San
STM32CubeIDE_Sample	サンプルプログラム	Use a password-protected	d SSH key.
STM32CubeIDE_file	サンプル	Download ZIP	
		t T h t Z Citlach Mard	-0

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。

🔤 sketch_may10b   A	rduino IDE 2.3.	2					
ファイル 編集 スケッチ	ツール ヘルプ	(H)					
新規スケッチ	Ctrl+N	(	odule		•		
新規クラウドスケッチ	Alt+Ctrl+N	- 1			4.01		
開<	Ctrl+O			sketch_n	1ay10b.	ino	
長近体。も何日も明く				1	void	setup	(
東近使つに項目を開く		1		2	//	put y	C
スケッチブック		->		3			
スケッチ例		ъ I		4	}		
~~~~~		·		5			
閉じる	Ctrl+W			6	void	loop(	)
Save	Ctrl+S		e:	7	11	put y	C
名前を付けて保存	Ctrl+Shift+S			8			
A MCNO CMT.	Curt Shirt+S			9	}		

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。

💿 ste	EP1   Arduin	o IDE 2.3.2
ファイル	編集 スケ	ッチ ツール ヘルプ(H)
Ø	€ 🗧	ボードを選択・
Ph	STEP1.ir	0
	1	/*
_	2	<pre>* Arduino_Sample/STEP1/STEP1.ino</pre>
Ľ_)	3	*
	4	* Copyright(C) 2020 RT Corporation <support@rt-net.jp></support@rt-net.jp>
ITh	5	* All rights reserved.
	6	*
	7	* License: Apache License, Version 2.0
$\oslash$	8	<pre>* https://www.apache.org/licenses/LICENSE-2.0 *</pre>
	10	* /
$\bigcirc$	10	
$\mathcal{A}$	12	int LED Pin= D13:
	13	
	14	<pre>void setup() {</pre>
	15	// put your setup code here, to run once:
	16	<pre>pinMode(LED_Pin,OUTPUT);</pre>
	17	
	18	}
	19	
	20	<pre>void loop() {</pre>
	21	<pre>// put your main code here, to run repeatedly:</pre>
	22	digitalWrite(LED_Pin,HIGH); //LEDを点灯
	23	delay(1000); //1秒待つ
	24	digitalWrite(LED_Pin,LOW); //LED)削灯
	25	delay(1000); //1秒待つ
	26	}
	27	

STEP1.inoを開いた状態



ボードが"Nucle-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。

		Ξ.	1 71 75 7 (1)	_				
	⇒ ₽		目動整形	Ctr	1+T			
	sketch r	nav1	スケッチをアーカイブする					
	1		ライブラリを管理	Ctrl+Shi	ft+l			
	2		シリアルモニタ	Ctrl+Shift	+M			
	3		シリアルプロッタ					
_	4	}	Firmware Updater					
Ŋ	6	vo	SSLルート証明書を書き	込み				
	7		ボード: "Nucleo-32"		•			
>	8	2	ポート		•			
	10	1	ボ−ド情報を取得					
2			Debug symbols and c	ore logs: "None"	•			
			Optimize: "Smallest (-	Os default)"	►			
			Board part number: "I	Nucleo F031K6"	Þ	~	Nucleo F031K6	
			C Runtime Library: "N	ewlib Nano (default)"	Þ		Nucleo F042K6	
			Upload method: "Mas	s Storage"	•		Nucleo F303K8	
			USB support (if availa	ble): "None"	•		Nucleo G031K8	
			U(S)ART support: "Ena	abled (generic 'Serial')"	•		Nucleo G431KB	
			USB speed (if availabl	e): "Low/Full Speed"	•		Nucleo L031K6	
			ブートローダを書き込む			1	Nucleo L412KB	
							Nucleo L432KC	

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
 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。
шл ☴ 읍
"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); [1]
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 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。

2	ソール ヘルプ(H)			
T	自動整形	Ctrl+T		
1	スケッチをアーカイブする			
*	ライブラリを管理	Ctrl+Shift+I		
*	シリアルモニタ	Ctrl+Shift+M		
*	シリアルプロッタ			
*	Firmware Updater			t-net.jp>
*	SSLルート証明書を書き込み			
*	ボード: "Nucleo-32"		۲	0
*	ポート		٠	シリアルポート
*	ボード情報を取得			COM1
n	Debug symbols and core logs: "None"		Þ	COM5

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

e動整形 Ctrl+T スケッチをアーカイブする ライブラリを管理 Ctrl+Shift+l シリアルモニタ Ctrl+Shift+M シリアルブロッタ Firmware Updater SSLルート証明書を書き込み ボード: "Nucleo-32" ポート ボード 情報を取得 Debug symbols and core logs: "None" のptimize: "Smallest (-Os default)" Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" USB support (if available): "None" USB support (if available): "Low/Full Speed" USB speed (if available): "Low/Full Speed"	ケッチ	ツール ヘルプ(H)			
Aケッチをアーカイブする ライブラリを管理、Ctrl+Shift+1 シリアルブロック Firmware Updater SSLルート証明書を書き込み ボート: "Nucleo-32" ボート: "Nucleo-32" ボート: "Nucleo-32" レebug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" C Runtime Library: "Nucleo F303K8" C Runtime Library: "Nucleo F303K8" C Runtime Library: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" USB support (if available): "None" USB speed (if available): "Low/Full Speed" Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karlow Karl		自動整形	Ctrl+T		
1.100       ライブラリを管理       Ctrl+Shift+I         /*       シリアルモニタ       Ctrl+Shift+M         シリアルブロッタ       Firmware Updater         *       SSLルート証明書を書き込み         ボード: "Nucleo-32"       ・         ボード: "Smallest (-Os default)"       ・         Optimize: "Smallest (-Os default)"       ・         VC       Board part number: "Nucleo F303K8"       ・         C Runtime Library: "Newlib Nano (default)"       ・         Upload method: "Mass Storage"       ✓       Mass Storage         VC       U(S)ART support: "Enabled (generic 'Serial)"       STM32CubeProgrammer (SWD)         VC       U(S)ART support: "Enabled (generic 'Serial)"       STM32CubeProgrammer (SWD)		スケッチをアーカイブする			
<pre>/* シリアルモニタ Ctrl+Shift+M  * シリアルブロック  Firmware Updater SSLルート証明書を書き込み * ボード: "Nucleo-32" * ボード: "Nucleo-32" * ボード * ボード * ボード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" VC Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" VC USB support (if available): "None" VC USB support (if available): "None" VC USB speed (if available): "Low/Full Speed" </pre>	.ino	ライブラリを管理	Ctrl+Shift+I		
* シリアルブロッタ Firmware Updater SSLルート証明書を書き込み ボード: "Nucleo-32" ボード: "Nucleo-32" ボート ボード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" USB support (if available): "None" USB support (if available): "None" STM32CubeProgrammer (SWD) STM32CubeProgrammer (SWD) STM32CubeProgrammer (Serial)" USB speed (if available): "Low/Full Speed"	*	シリアルモニタ	Ctrl+Shift+M		
Firmware Updater Firmware Updater SSL/L-ト証明書を書き込み ボード: "Nucleo-32" ボード: "Nucleo-32" ボート ホード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Vo Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" Vo U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" Firmware Updater Firmware Updater STM32CubeProgrammer (SWD) STM32CubeProgrammer (SWD) STM32CubeProgrammer (SPIA)	*	シリアルプロッタ			
* SSLルート証明書を書き込み ボード: "Nucleo-32" ホード: "Nucleo-32" ホート ボード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Ootimize: "Smallest (-Os default)" Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" VC U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (SWD) STM32CubeProgrammer (Serial)	*	Firmware Updater		t-ne	t.jp>
* ボード: "Nucleo-32" * ボード: "Nucleo-32" * ボード ボード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" VO Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" VO U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (SWD) STM32CubeProgrammer (SVD)	*	SSLルート証明書を書き込み			
* ポート * ポード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" VC Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" VC U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" VC USB speed (if available): "Low/Full Speed"	*	ボード: "Nucleo-32"		•	
* ボード情報を取得 in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" VC U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (SWD) STM32CubeProgrammer (SPIal)	*	ポート		•	
in Debug symbols and core logs: "None" Optimize: "Smallest (-Os default)" Board part number: "Nucleo F303K8" C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" VC U(S)ART support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (DFU)	*	ボード情報を取得			
Optimize: "Smallest (-Os default)"       >         Board part number: "Nucleo F303K8"       >         C Runtime Library: "Newlib Nano (default)"       >         Upload method: "Mass Storage"       >         USB support (if available): "None"       >         VC       U(S)ART support: "Enabled (generic 'Serial')"       >         USB speed (if available): "Low/Full Speed"       >	in	Debug symbols and core logs: "None"		•	
Vd       Board part number: "Nucleo F303K8"         C Runtime Library: "Newlib Nano (default)"         Upload method: "Mass Storage"         VG       USB support (if available): "None"         VC       U(S)ART support: "Enabled (generic 'Serial')"         USB speed (if available): "Low/Full Speed"    STM32CubeProgrammer (SWD) STM32CubeProgrammer (DFU)		Optimize: "Smallest (-Os default)"		•	
C Runtime Library: "Newlib Nano (default)" Upload method: "Mass Storage" USB support (if available): "None" USB support: "Enabled (generic 'Serial')" USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (SWD) STM32CubeProgrammer (DFU)	vo	Board part number: "Nucleo F303K8"		•	
Upload method: "Mass Storage"       Mass Storage         USB support (if available): "None"       STM32CubeProgrammer (SWD)         VG       U(S)ART support: "Enabled (generic 'Serial')"       STM32CubeProgrammer (Serial)         USB speed (if available): "Low/Full Speed"       STM32CubeProgrammer (DFU)		C Runtime Library: "Newlib Nano (default)"		۲ I	
}       USB support (if available): "None"       STM32CubeProgrammer (SWD)         vc       U(S)ART support: "Enabled (generic 'Serial')"       STM32CubeProgrammer (Serial)         USB speed (if available): "Low/Full Speed"       STM32CubeProgrammer (DFU)		Upload method: "Mass Storage"		+ -	Mass Storage
VC     U(S)ART support: "Enabled (generic 'Serial')"     Image: STM32CubeProgrammer (Serial)       USB speed (if available): "Low/Full Speed"     STM32CubeProgrammer (DFU)	}	USB support (if available): "None"		•	STM32CubeProgrammer (SWD)
USB speed (if available): "Low/Full Speed" STM32CubeProgrammer (DFU)	va	U(S)ART support: "Enabled (generic 'Serial')"		F	STM32CubeProgrammer (Serial)
		USB speed (if available): "Low/Full Speed"		F .	STM32CubeProgrammer (DFU)



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ボート名の左に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、 ら書き込み完了です。

#### 出力



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 ∨2.3.0
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。



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 Authe
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。

4 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

📣 ST-Link Upgrade	×
Device Connect Current Firmware: Version: Type:	STM82
Upgrade to Firmware:	Yes >>>>

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。

ST-Link Upgrade Device Connect Current Firmware: Version: V2.J31.M21 Type: STM32 Debug +Mass s Change Type	κ torage+VCP
Upgrade to Firmware:	STM32 Debug +Mass storage +VCP V2. J44.M29 Yes >>>>



VersionやTypeが表示されたら"Yes>>>>"をクリックします。

ST-Link Upgrade	×
Device Connect	ST-LinkUpgrade
Version: V2.J31.M21 Type: STM32 Debug+Mas	
Change Type	
Upgrade to Firmware:	OK
	Yes >>>>

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

	STLinkUpgrade 3.14.5	- • ×
Open in update mode	Refresh device list	<b>S</b>
ST-Link ID:		Releasing your creativity
Current Firmware:		
Туре:		
Version:		
Update to Firmware:		
Upgrade		

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。

	STLinkUpgrade 3.14.5	- • ×
ST-LINK/V2-1	Refresh device list	
Open in update mode		STM22
ST-Link ID: 0667FF303032424257015845		Releasing your creativity
Current Firmware:		
Туре:		
Version: Unknown		
Update to Firmware: Unknown		
Upgrade		

"Open in update mode"をクリックします。



STLinku	Jpgrade 3.14.5	- • ×
ST-LINK/V2-1	sh device list	
Open in update mode	STM32	
ST-Link ID: 0667FF303032424257015845	Re	eleasing your creativity
Current Firmware:		
Type: STM32 Debug+Mass storage+VCP		
Change Type		
Version: V2J45M30		
Update to Firmware: V2J44M29 STM32 Debug+Ma Upgrade	ass storage+VCP	

VersionやTypeが表示されたら"Upgrade"をクリックします。

STLinkUpgrade 3.14.5		×
ST-LINK/V2-1 Refresh device list	-	-
Open in update mode		1
ST-Link ID: 0667FF303032424257015845	ry y	
Current Firmware:		
Type: STM32 Debug+Mass storage+VCP		
Version: V2J44M29		
Update to Firmware: V2J44M29 STM32 Debug+Mass storage+VCP		
Upgrade		
Upgrade successful.		

"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。





プライバシーとセキュリティ
プライバシーとセキュリティがシステム設定を 変更しようとしています。
許可するにはパスワードを入力してください。
ローザ名
パスワード
キャンセル 設定を変更

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。

? ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
このアプリケーションを開くことによって、 システムのセキュリティが無効になり、コン ピュータと個人情報がマルウェアにさらされ る場合があります。その結果、マルウェアに よって、Mac やプライバシーに損害を受ける 可能性があります。
このファイルは "Safari" により今日の16:25 にダウンロードされました。
開く
ゴミ箱に入れる
キャンセル

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。

• • •	STLinkUpgrade 3.15.6
ST-LINK/V2-1	Refresh device list
Open in update mode	STM32
ST-Link ID: 0667FF303032424257015845	Releasing your creativity
Current Firmware:	
Туре:	
Version: Unknown	
Update to Firmware: Unknown	
Upgrade	

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。



#### Arduino開発環境構築マニュアル

•••	STLinkUpgrade 3.15.6
ST-LINK/V2-1	Refresh device list
Open in update mode	STM32
ST-Link ID: 0667FF303032424257015845	Releasing your creativity
Current Firmware:	
Type: STM32 Debug+Mass storage+VCP	
Change Type	
Version: V2J44M29	
Update to Firmware: V2J45M30 STM32 Debug+M	Mass storage+VCP
Upgrade	

VersionやTypeが表示されたら"Upgrade"をクリックします。

STLinkUpgrade 3.15.6
ST-LINK/V2-1 © Refresh device list
Open in update mode
ST-Link ID: 0667FF303032424257015845
Current Firmware:
Type: STM32 Debug+Mass storage+VCP
Version: V2J45M30
Update to Firmware: V2J45M30 STM32 Debug+Mass storage+VCP
Upgrade
Upgrade successful.

'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	https://www.arduino.cc/en/software
Arduino core for the ESP32	3.0.1	https://github.com/espressif/arduino-esp 32/releases
Arduino core support for STM32 based boards	2.7.1	https://github.com/stm32duino/Arduino_ Core_STM32/releases
STM32CubeProgrammer	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



# 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.

