

1、開発環境の構築

1.1 STM32 用統合開発環境 STM32CubeIDE の入手とインストール

下記 Web サイトにアクセスします。

<https://www.st.com/ja/development-tools/stm32cubeide.html>

次に使用している PC の OS やソフトウェアのバージョン確認し「ダウンロード」をクリックします (図 1)。その後、ライセンス契約画面がでるので、「承諾します」をクリックします (図 2)。

ソフトウェア入手

ソフトウェア名	ECCN	OS	バージョン	アクション
STM32CubeIDE	NEC (EU) EAR99 (US)	Windows	2.1.1 Latest	
STM32CubeIDE for Visual Studio Code	NEC (EU) EAR99 (US)	-	-	 Visual Studio Codeマーケットプレイスに移動

GET-SOFTWARE-RECOMMENDS

図 1 STM32CubeIDE のダウンロード

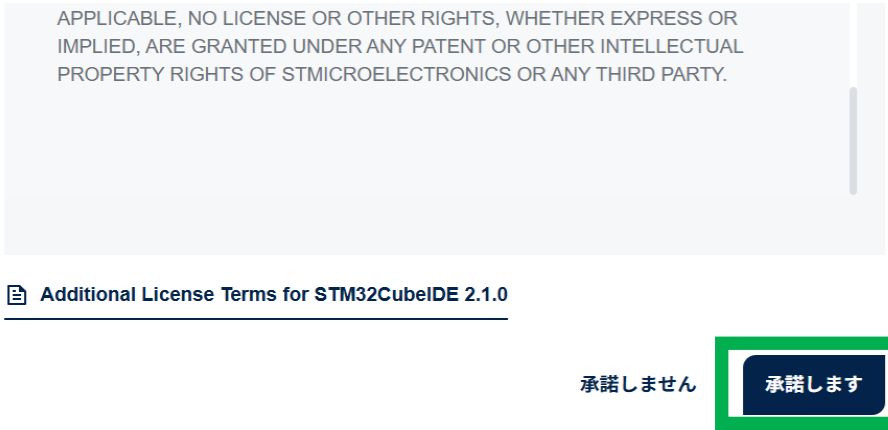


図 2 ライセンス契約画面

続いて、アカウント関係の画面に移ります (図 3)。MyST アカウントがある方はログインし、ない方は「ゲストとしてダウンロード」をクリックします。ここでアカウントを作成することもできます。

Thanks for having accepted the license.

MySTにログインするか、アカウントを作成し、
ダウンロードを開始してください。



図3 MyST アカウント有無に応じてクリック

ここでは、ゲストとしてダウンロードします。必要な情報を入力し、最後に「ダウンロード用のリンクを入手」をクリックします（図 4）。

ゲストとしてダウンロード

by filling this form you will receive an email with the link to download the software.

名

名

姓

姓

業務用メールアドレス *

業務用メールアドレス

姓名, メールアドレスを入力

STによるお客様のプロフィール情報の処理方法、またお客様がご自身の個人情報の保護に関する権利を行使する方法については、当社の [プライバシー・ステートメント](#) をご確認ください。

☐ Please keep me informed about future updates for this software or new software in the same category

ダウンロード用のリンクを入手

入力したらクリック

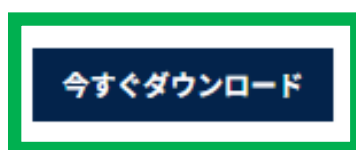
図 4 ダウンロードに必要な情報を入力

入力したメールアドレス宛にリンクが送信されてくるので、メールにある「今すぐダウンロード」をクリックします（図 5）。
これで、インストーラがダウンロードできました。



日頃は弊社製品をご愛顧いただきまして誠にありがとうございます。

メールアドレスを認証して **STM32CubeIDE-Win** をダウンロードするには、以下のボタンをクリックしてください。



よろしくお願いいたします。
STマイクロエレクトロニクス

図5 入力したメールアドレス宛にリンクが送られてくる

入手したインストーラを解凍して実行します。最初の画面では「Next」をクリックします(図6)。

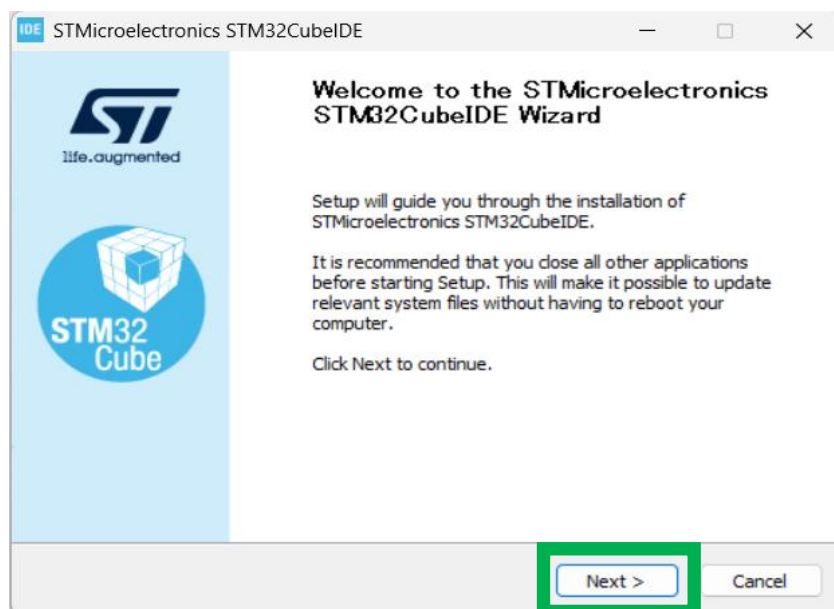


図6 インストーラ実行後の最初の画面

以降、License Agreement の画面では「I Agree」を選択し、インストール先も指定します (図 7)。最後に「Install」をクリックしインストールを実行し (図 8)、最後に「Finish」をクリックすればインストール完了です (図 9)。

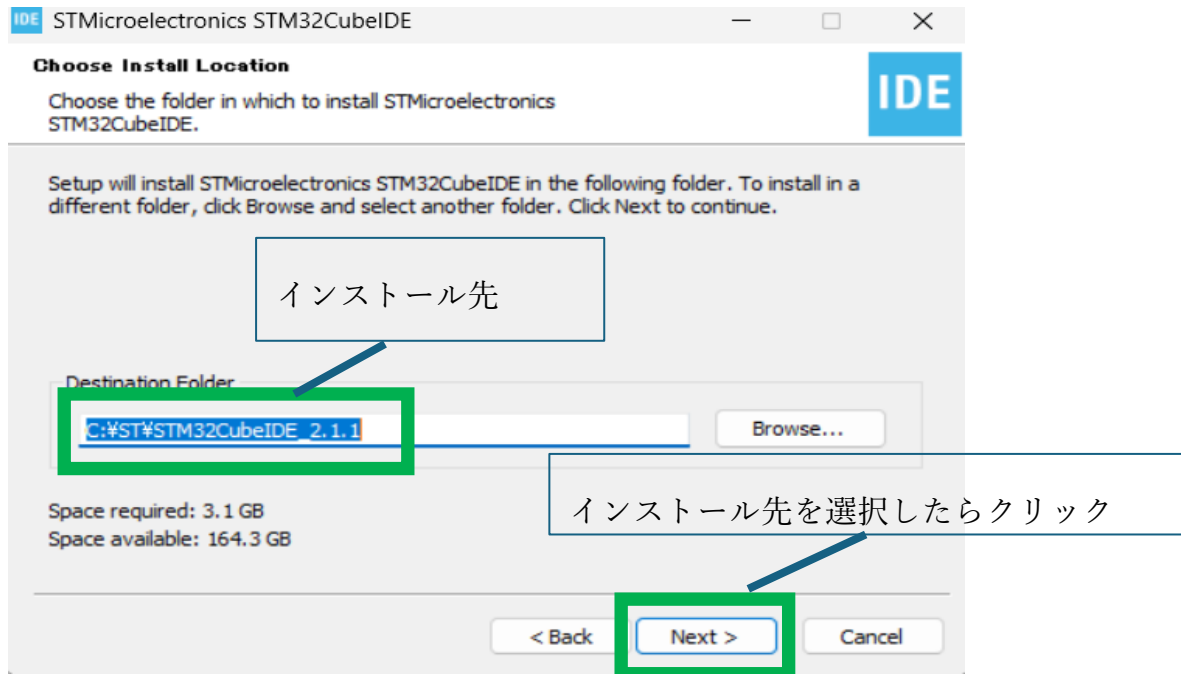


図 7 インストール先の指定 (デフォルトで問題ない)

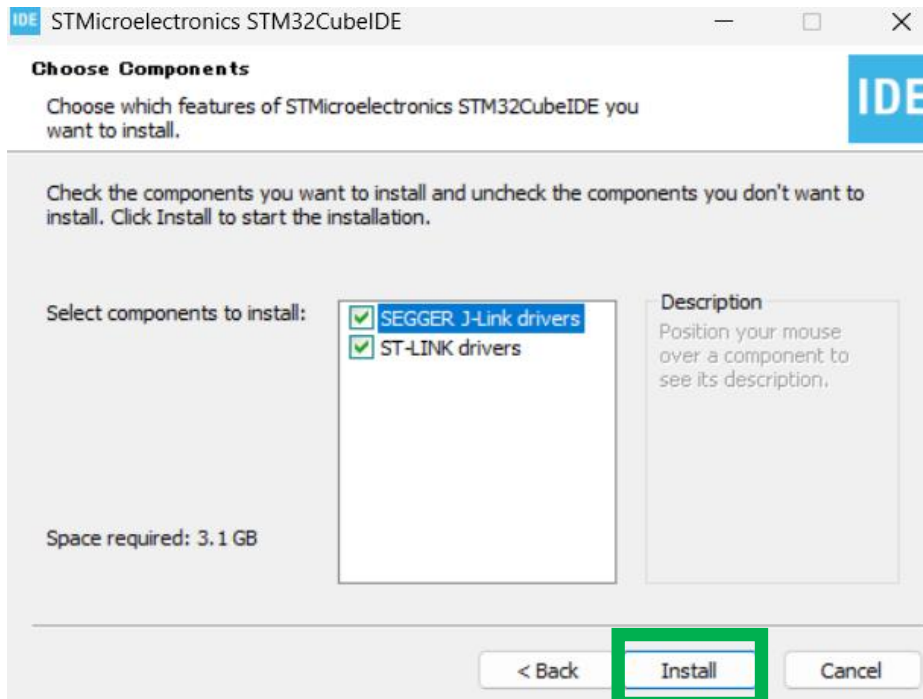


図 8 「Install」をクリックしてインストールを実行



図9 「Finish」をクリックすればインストール完了

1.2 STM32CubeMXの入手とインストール

STM32CubeMXは、STM32マイコンの各種設定をGUI画面上で行うためのツールです。GPIOの入出力設定、クロック設定、タイマ、UART、A-Dコンバータ、PWMなどの周辺機能を画面操作で設定でき、設定内容に応じた初期化コードを自動生成します。従来はレジスタ設定を手作業で記述する必要がありましたが、CubeMXを使うことで設定ミスを減らし、開発効率を大幅に向上できます。また、STM32CubeIDEと連携することで、生成したプロジェクトをそのままビルド・デバッグへ進められます（なくてもプログラムの書き込みや、パラメータの変更はできます）

まずはSTマイクロエレクトロニクス公式ウェブ・サイトへアクセスします。

<https://www.st.com/ja/development-tools/stm32cubemx.html>

STM32CubeIDEのときと同じように使用しているPCのOSやソフトウェアのバージョンを確認して「ダウンロード」をクリックします（図10）。

ソフトウェア入手

ソフトウェア名	対応ハードウェア	ECCN	OS	バージョン	アクション
STM32CubeMX	All MCU/MPU except STM32C5 Series	NEC (EU) 5D992.c (US)	Windows ▾	6.17.0 Latest ▾	 ダウンロード
STM32CubeMX2	STM32C5 Series ⓘ	NEC (EU) 5D992.c (US)	Windows ▾	1.0.1 Latest ▾	 ダウンロード

GET-SOFTWARE-RECOMMENDS

図 10 STM32CubeMX のダウンロード

以降は STM32CubeIDE の入手、インストールのときと同様にライセンス契約画面やアカウント関連の画面が出てきますので、同じように選択すれば STM32CubeMX のインストーラがダウンロードできます。

ダウンロードしたら zip ファイルを展開し exe ファイルを実行します。すると、図 11 の画面がでてくるので、推奨されている「Install for me only」をクリックします。次に画面の Starting STM32CubeMX 6.17.0 installation では「Next」をクリックし、LICENSE AGREEMENT の画面では図 12 のようにチェックを入れて「Next」をクリックします。その後、図 13 も同じようにチェックを入れて「Next」をクリックします。

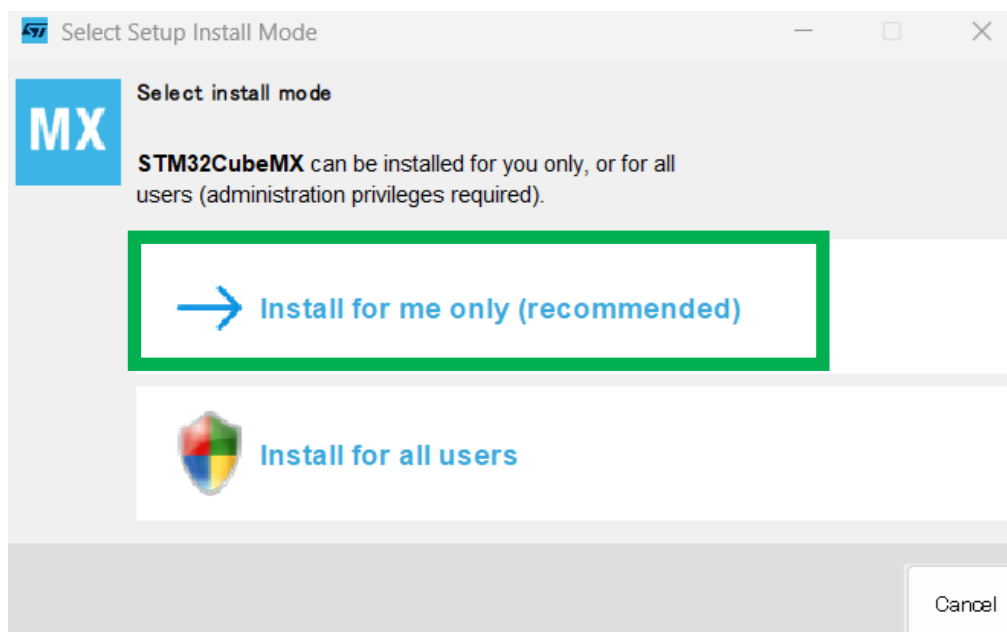


図 11 ユーザ設定画面

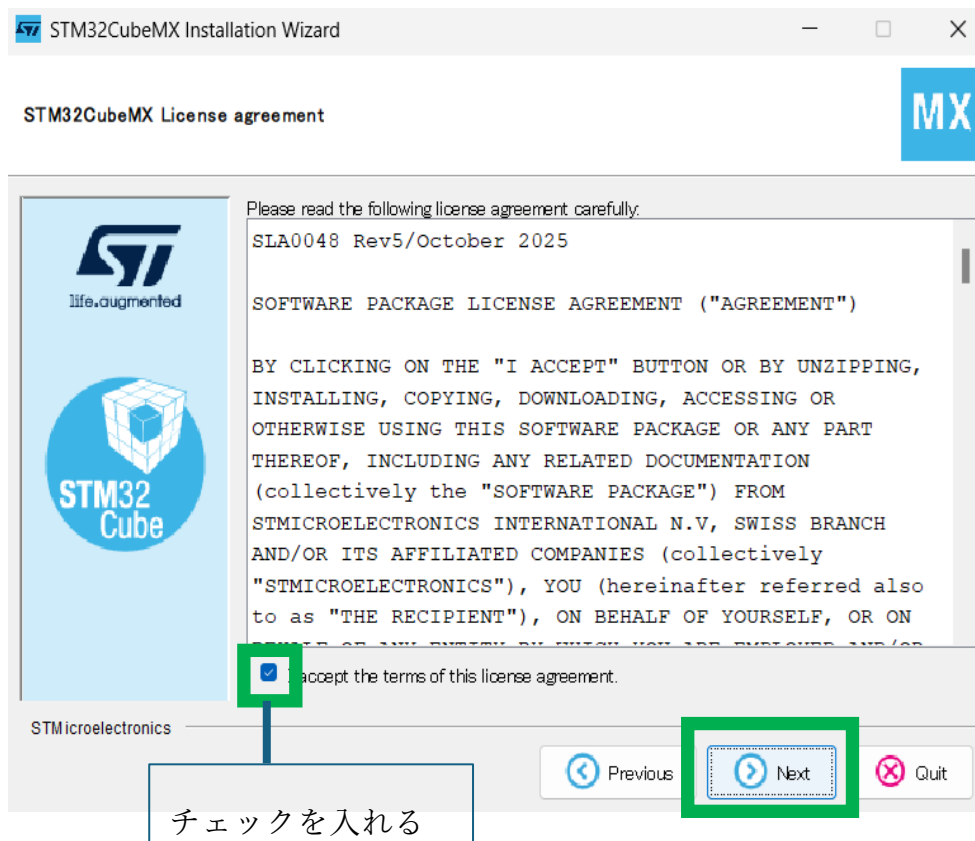


図 12 ライセンス画面

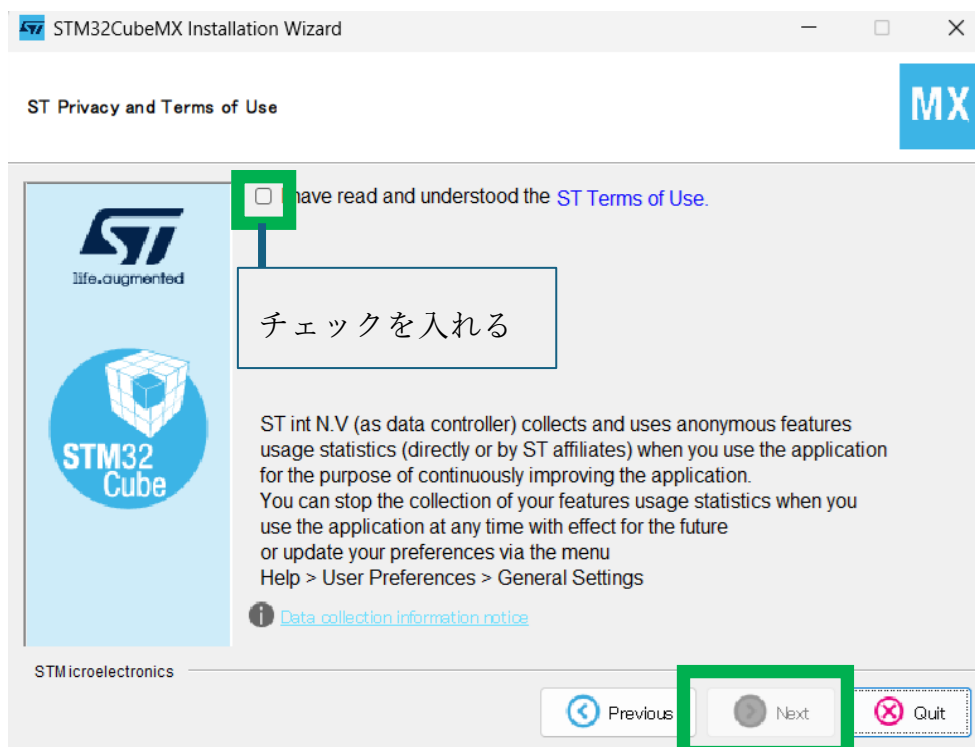


図 13 ST Privacy and Terms of Use 画面

続いて、図 14 のようにパスの設定画面がでてくるので、「Next」をクリックします。
ここで、フォルダがない場合は図 15 のようなメッセージがでるので「OK」をクリックします。

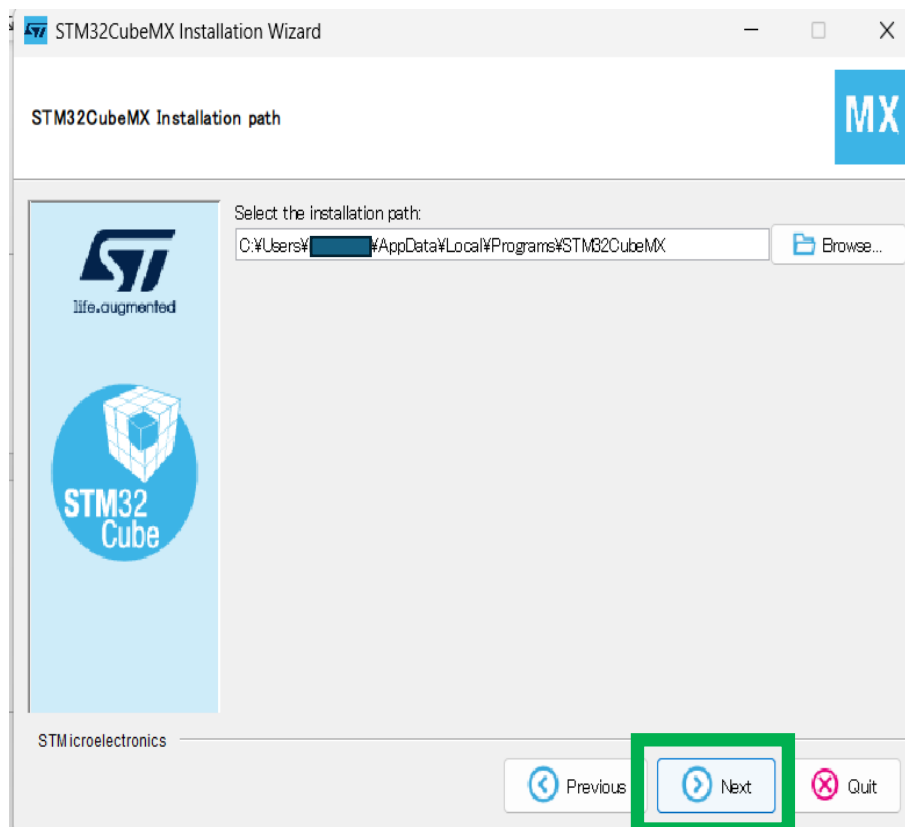


図 14 パスの設定画面

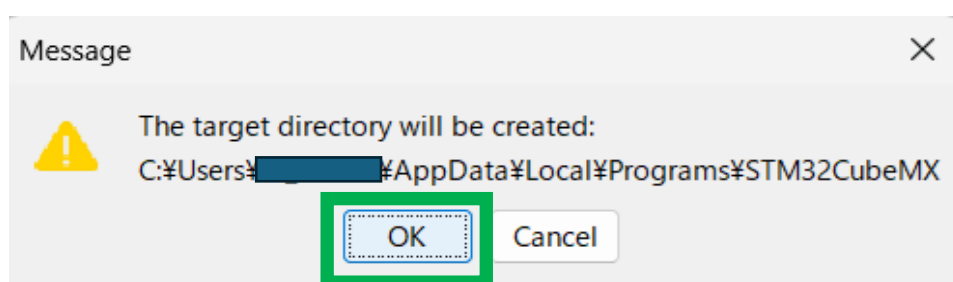


図 15 ディレクトリの新規作成

その次の画面は既定設定のままで OK です (図 16)。

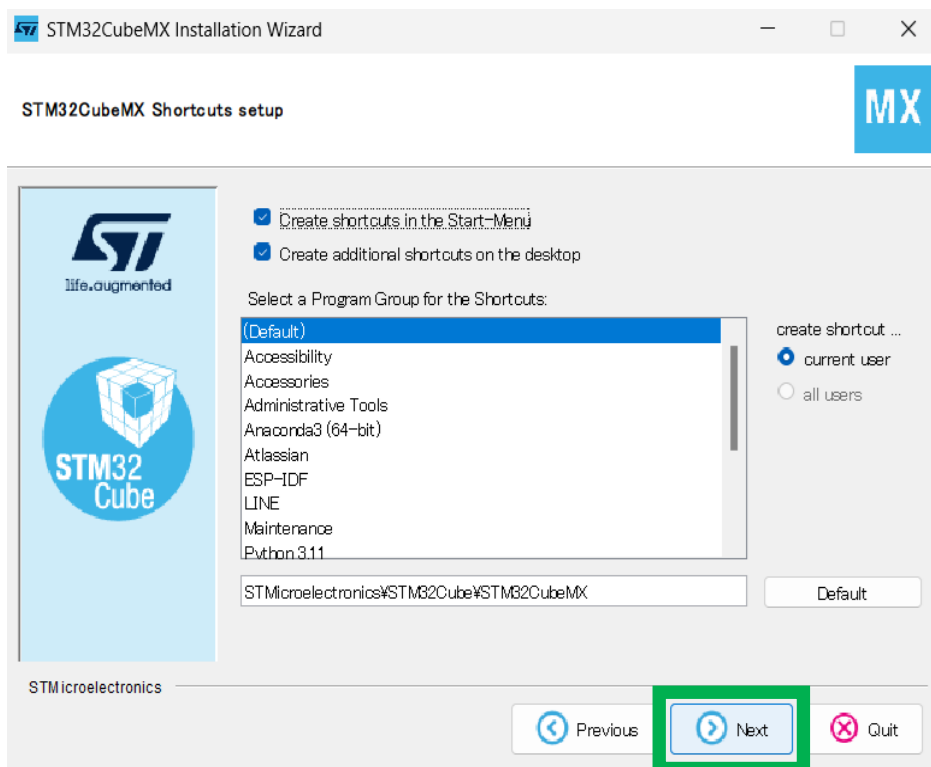


図 16 セットアップ画面

以降、「Next」をクリックしていき、最後に「Done」をクリックすればインストール完了です（図 17）。

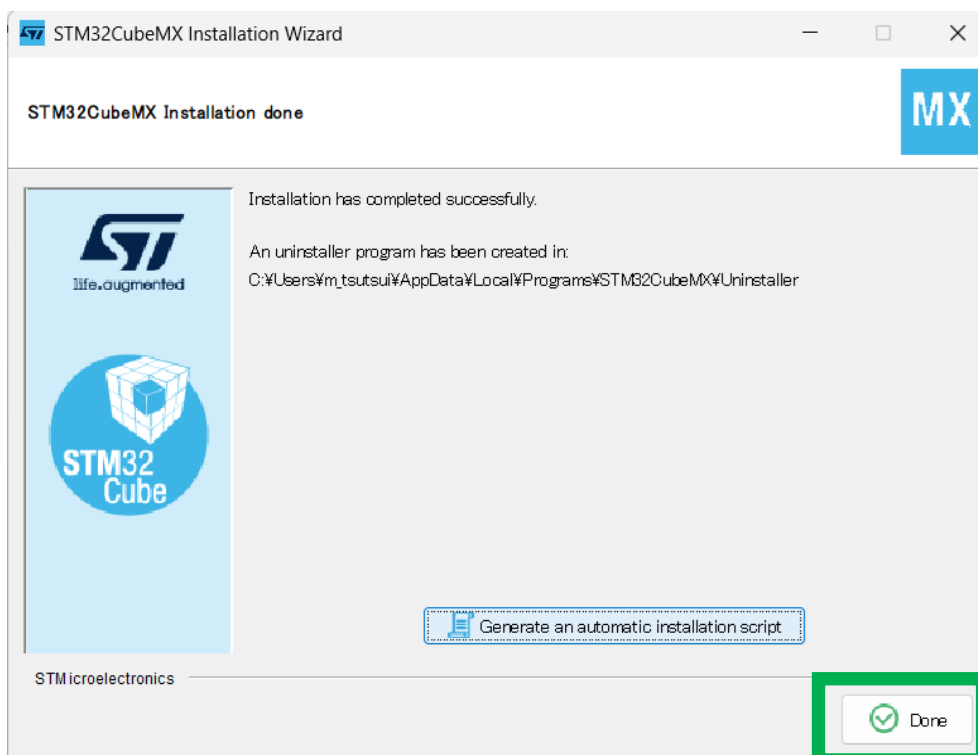


図 17 「Done」をクリックすればインストール完了

1.3 STM32CubeIDE で筆者プロジェクトを書き込む

ここでは筆者が提供するプロジェクト (CQ_Motor_CubeSimple01.zip) を Nucleo ボードに書き込む方法について説明します。

まずは、入手した CQ_Motor_CubeSimple01.zip を展開しておきます。ここでは、展開先はデスクトップとして説明します (任意の場所にする場合は適宜読み替えてください)。続いて、STM32CubeIDE を起動します。起動すると図 18 の画面が表示されます。

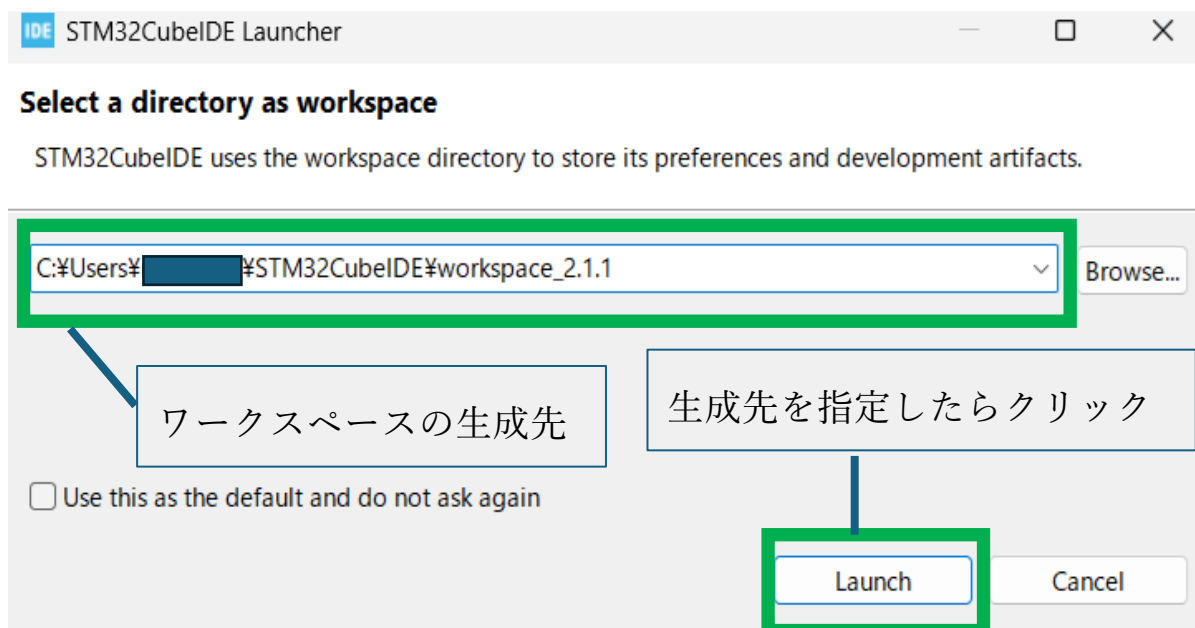


図 18 ワークスペース・ディレクトリの生成先を指定

ワークスペースの生成先を指定したら「launch」をクリックします。ここで、追加の質問はせず、この設定をデフォルトとして適用とする場合は「Use this as the default and do not ask again」にチェックを入れます。無事に起動すると図 19 の画面が表示されます。

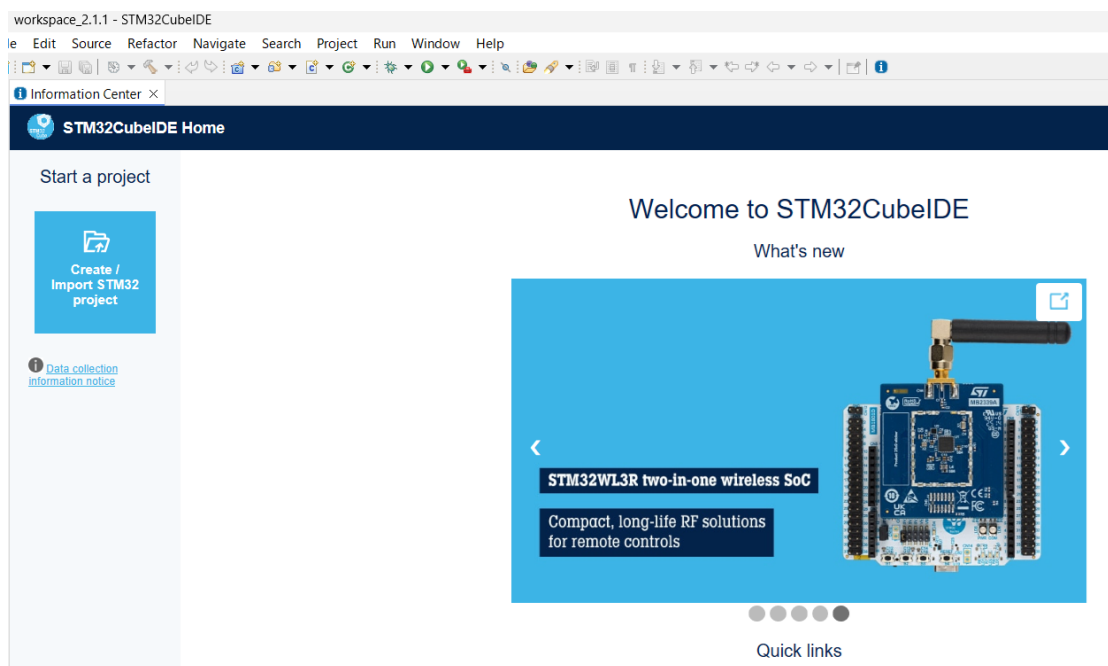


図 19 STM32CubeIDE 起動時の画面

次に、「File」→「Open Projects from File System」と進み（図 20），Import Source のところで「Directory…」をクリックします（図 21）。

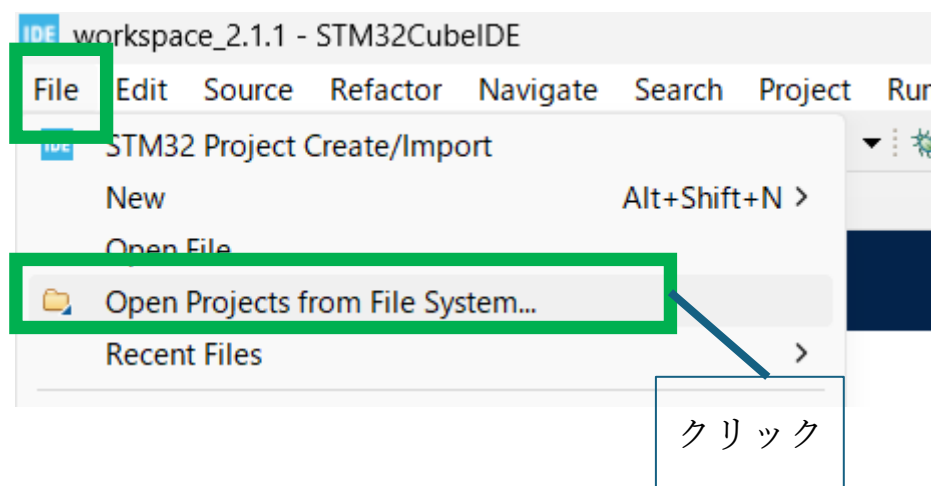


図 20 左上にある「File」から「Open Projects from File System」をクリック

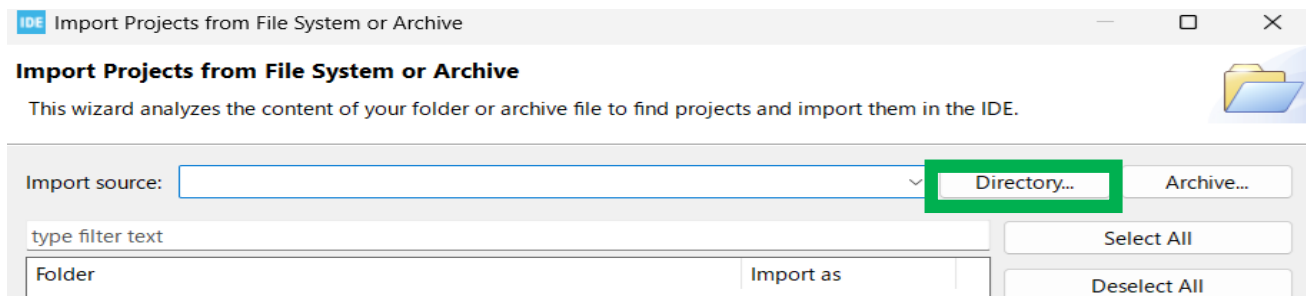


図 21 Import source の「Directory…」をクリック

クリックするとフォルダ選択の画面になるので、CQ_Motor_CubeSimple01.zip の展開したフォルダを指定します（図 22）。

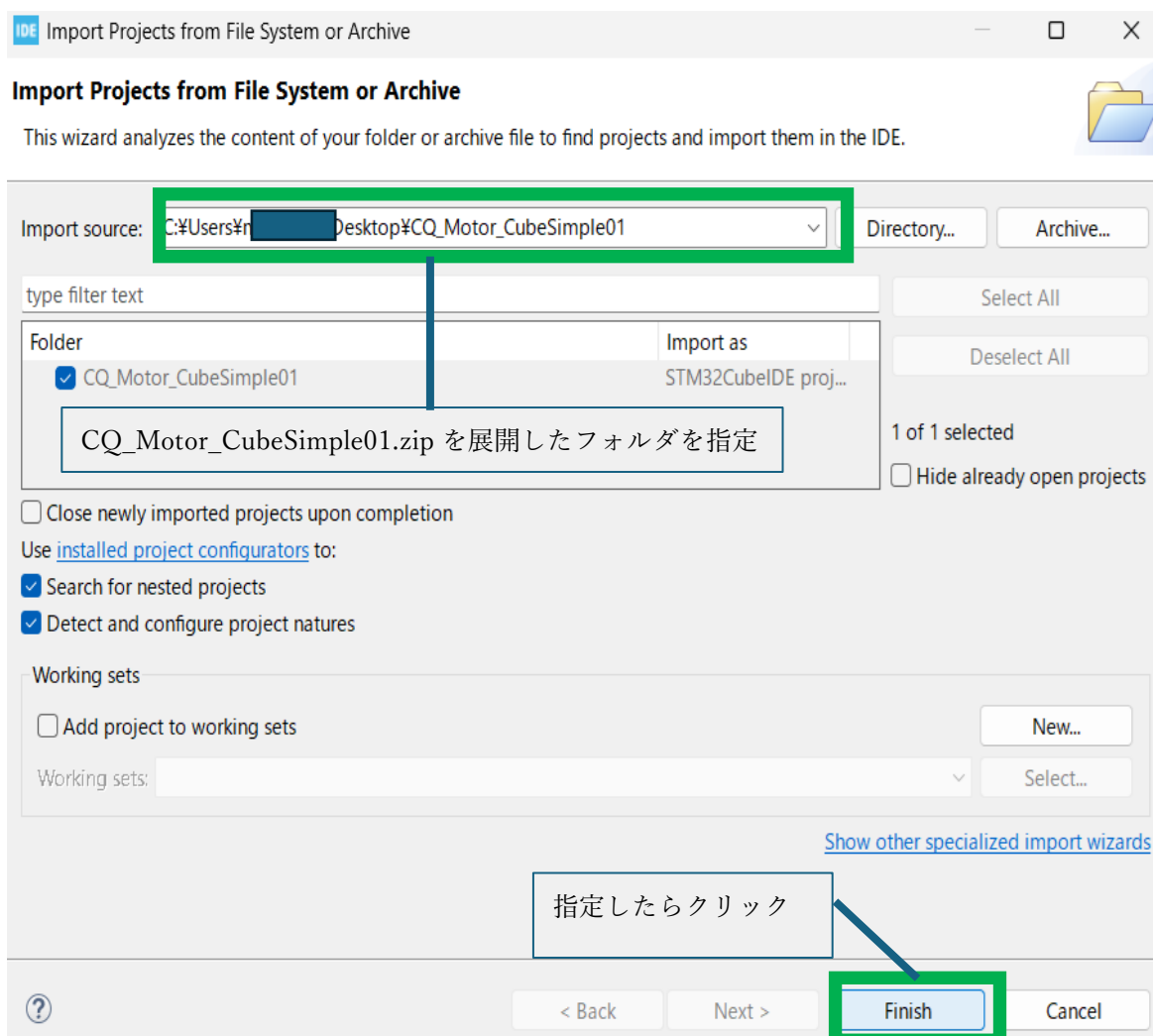


図 22 CQ_Motor_CubeSimple01.zip の展開先フォルダを指定し「Finish」をクリック

すると、図 23 のようにプロジェクト・フォルダに必要なファイルなどが表示されます。

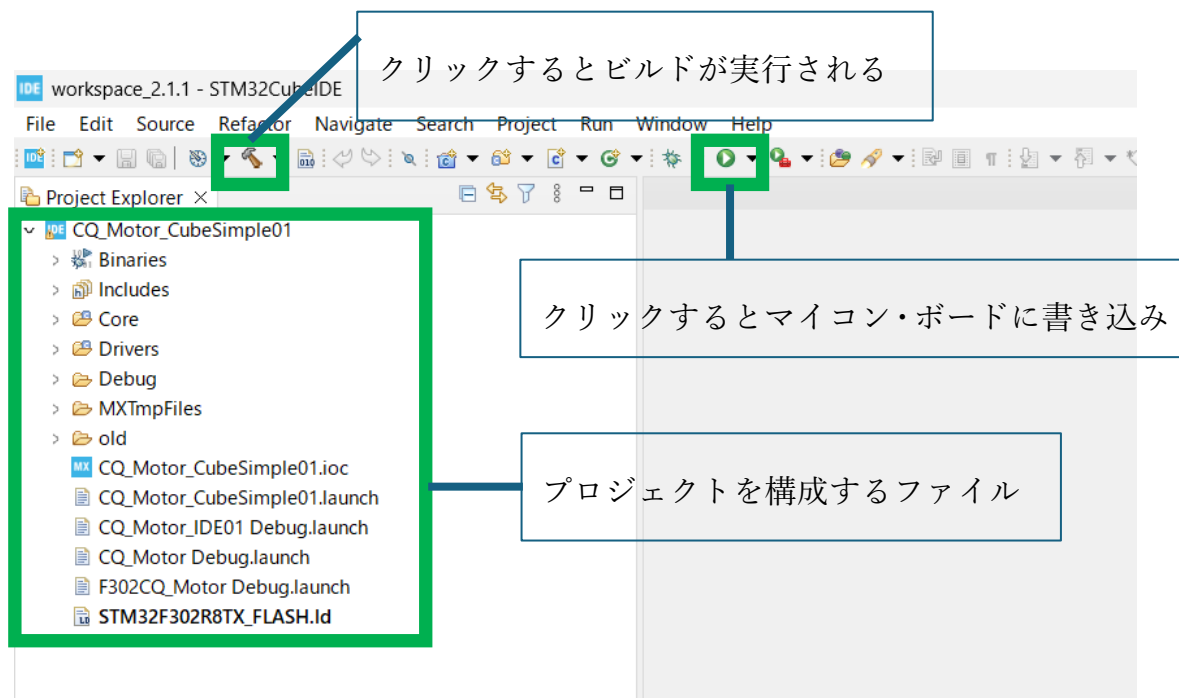



図 23 プロジェクト・フォルダの構成が表示される

プロジェクト・フォルダの中身が表示されたら、上部のハンマー（) ボタンをクリックしビルドを実行します。正常終了すると、.bin や.elf ファイルが生成されます。その後、Nucleo ボードと PC を USB 接続し、上部の▶ (Run) をクリックします。このとき、図 24 のような画面がでたら「OK」をクリックし、次の画面で「Open in update mode」をクリック後、「Upgrade」をクリックします(図 25)。Upgrade が終わると左下に「Upgrade successful」と表示されるので、表示されたら右上の×印をクリックし画面を閉じます。

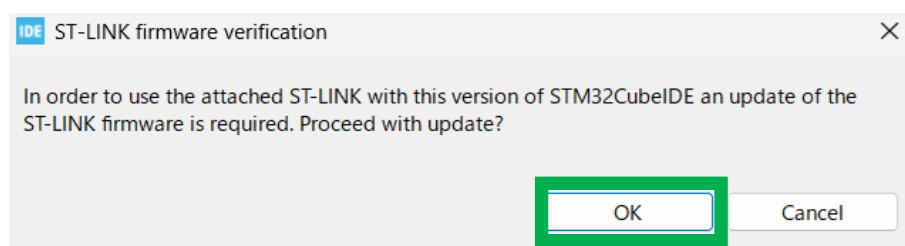


図 24 ST-LINK のファームウェアを更新

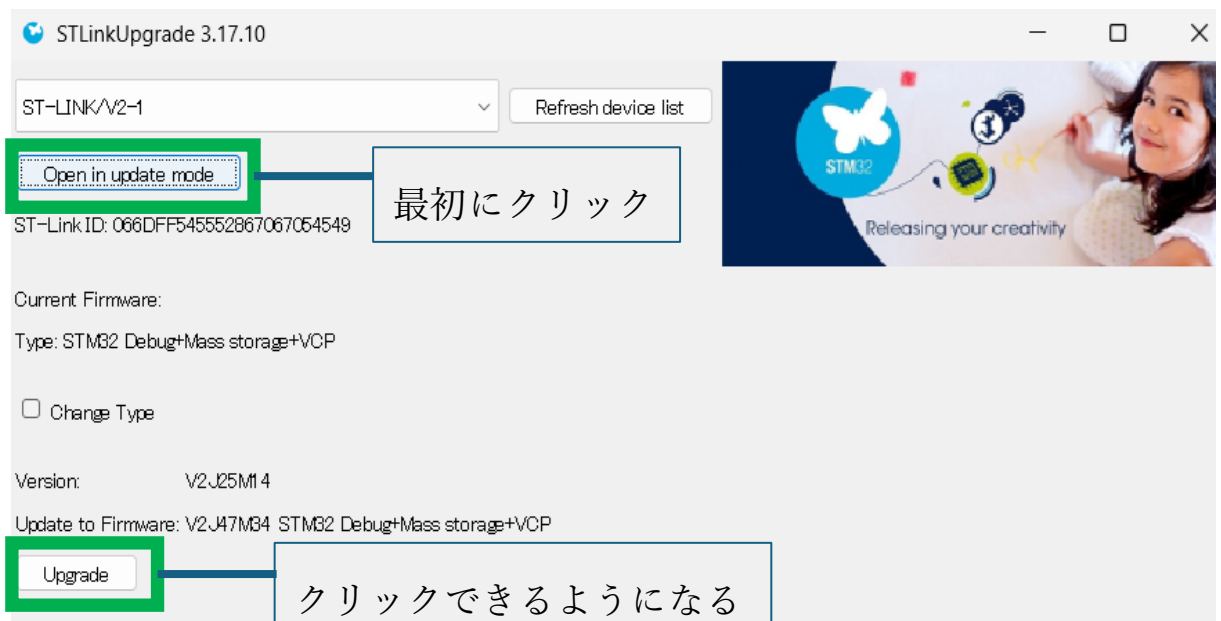
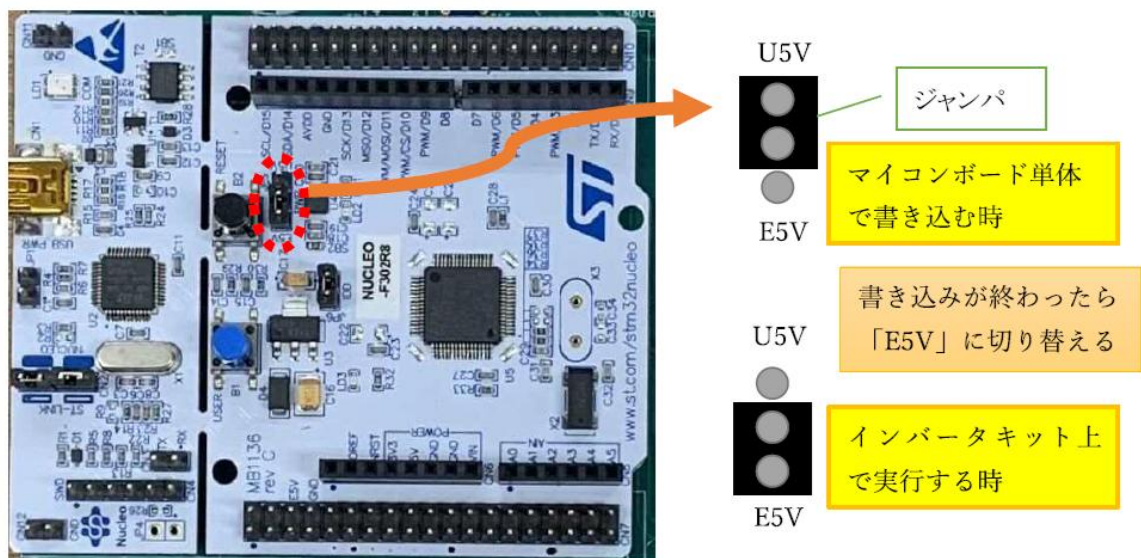


図 25 「Open in update mode」をクリック後「Upgrade」をクリック

書き込みが正常に終わると、図 26 のように Console に「Download verified successfully」と表示されます。これは STM32 への転送成功、書き込み検証成功を意味します。書き込みが終わったら、ジャンパ・ピン JP5 を E5V に戻すことを忘れないでください。



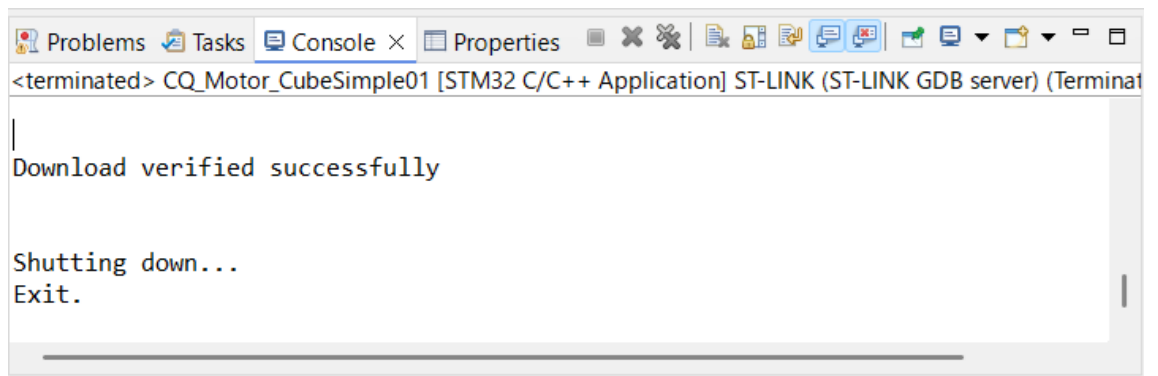


図 26 ボードへの書き込み

ここで、注意点として、Drivers 以下は巨大ライブラリです。誤って Drivers/以下を開くと、「Editor Scalability」と警告が出ることがあります。これは巨大ファイル警告であり、異常ではありません。基本は、「No」を選択します。

1.4 プログラムの編集

主に編集するのは、

Core

└ Src

└ main.c

にある main.c です (図 27)。Drivers 以下は STM 公式ライブラリなので、初心者は基本触らないでください。

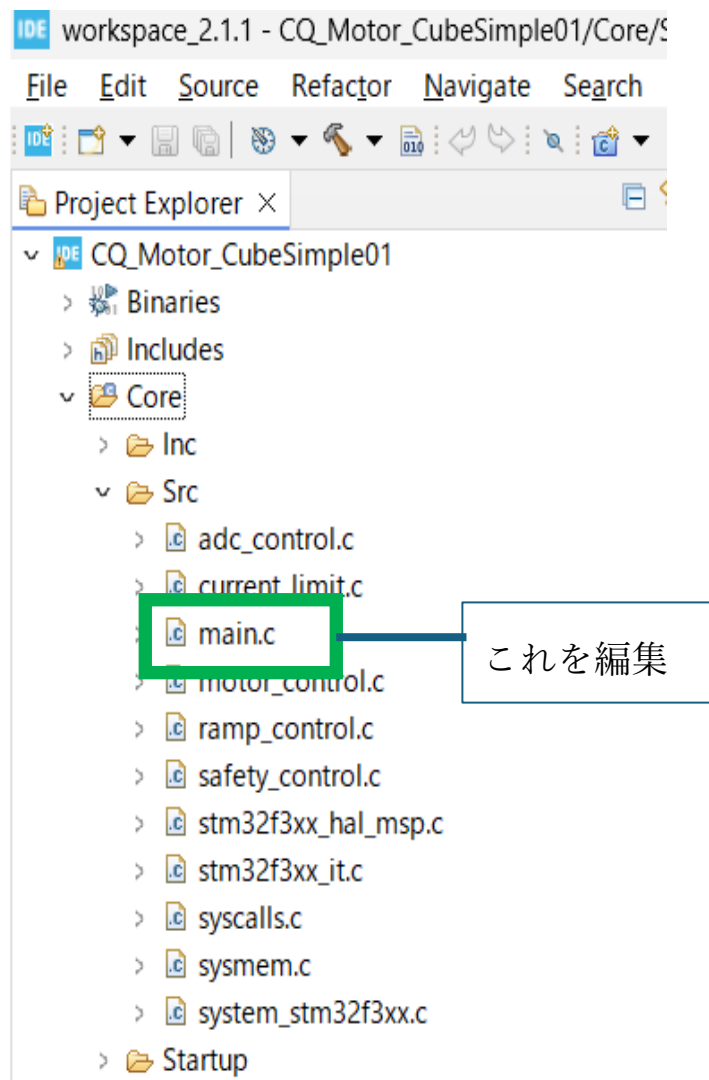


図 27 プログラム編集するのは main.c

1.5 STM32 開発の流れ

STM32 開発は概ね図 28 の流れで進みます。

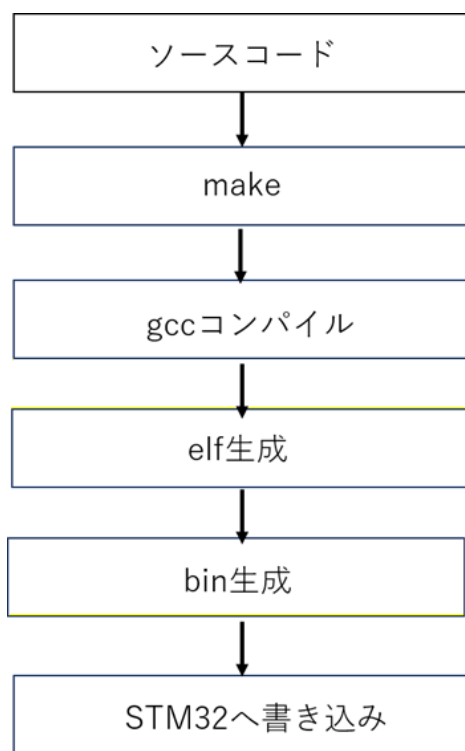


図 28 STM32 開発の流れ

ここで、それぞれの用語について説明しておきます。

項目	内容
ビルド	ソースコードを機械語へ変換する作業
make	ビルド手順を管理するツール
elf	デバッグ用完成ファイル
bin	STM32 へ書き込む実行ファイル
ST-LINK	STM32 書き込み器


1.6 main.c を書き換えた後の手順

STM32 では、main.c を修正→ビルド→STM32 へ再書き込みという流れになります。

1. main.c を保存

編集後 [Ctrl] + [S] を押して保存する。


2. ビルドする

画面上部の  (ハンマー) ボタンをクリックする。または「Project」→「Build Project」を選択する。

3. ビルド成功確認

Console に「Build Finished」などが表示されれば成功。この時点で、bin, elf が更新される。

4. STM32 へ書き込む

画面上部の  (Run) をクリックする。

5. 書き込み成功確認

Console に「Download verified successfully」と表示されれば成功。STM32 は新しいプログラムで動作を開始する。

1.7 Src フォルダ下のプログラムについて

筆者提供のプログラムは、BLDC（ブラシレス DC）モータを STM32 で制御するためのソフトウェアです。特徴は、機能ごとにファイルを分割している点にあります。大学のプログラミング演習では main.c にすべてを書く例も多いですが、本プログラムでは役割分担を明確にしています。これは実際の組み込み開発でよく使われる構成です。

プログラム全体の流れは、main.c が中心となります。main.c では、STM32 の初期化を行ったあと、

- モータ制御初期化
- A-D コンバータ初期化
- タイマ開始

などを実行します。その後、while(1)ループの中で、5ms ごとに制御処理を繰り返しています。

まず、adc_control.c ではアクセル値とモータ電流を A-D コンバータで取得しています。例えばアクセル電圧を 0～1.0 の範囲に正規化し、そこからデューティ比へ変換しています。デューティ比とは PWM の ON 時間割合であり、モータ速度を決める重要な値です。しかし、アクセル値をそのままモータへ与えると急加速して危険です。そこで ramp_control.c が使われます。このファイルではデューティを徐々に変化させています。例えば、アクセルを急に全開にしても、デューティを少しずつ増加させることで滑らかな加速を実現しています。これは電気自動車や電動工具でも使われる考え方です。

さらに `current_limit.c` では過電流保護を行っています。モータ電流が設定値を超えた場合、デューティを減少させます。さらに重大な過電流が発生すると、異常状態を保持し、アクセルを戻すまで再起動できないようにしています。これはモータや MOSFET の破損防止に重要です。

実際に PWM 信号を出力しているのが `motor_control.c` です。このプログラムでは TIM1 を用いて三相 PWM を生成しています。また、ホール・センサの状態を読み取り、6 ステップ整流を行っています。BLDC モータでは回転位置に応じて励磁相を切り替える必要がありますが、その切り替えテーブルが `drive_table_forward` です。つまり、全体としては図 29 の流れで動作しています。

このように機能分割を行うことで、速度制御だけ変更したい、保護機能を追加したいといった場合でも、他の部分へ影響を与えにくくなります。これは大規模な組み込みソフトウェアで特に重要な考え方です。

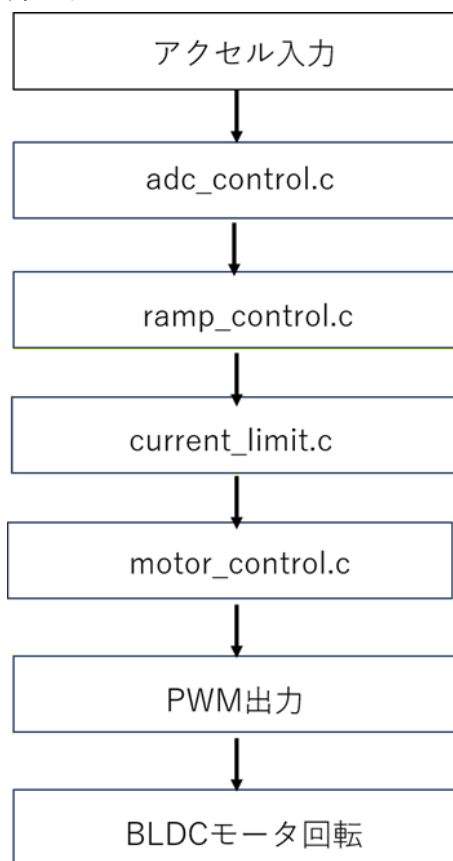


図 29 STM32 でモータ制御する流れ