

開発環境の構築

まず、下記ソフトをインストールします。

① 統合開発環境

STM32CubeIDE

Version : 1.3.0

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

② フラッシュライター

STM32CubeProgrammer

Version : 2.3.0

URL : <https://www.st.com/ja/development-tools/stm32cubeprog.html>

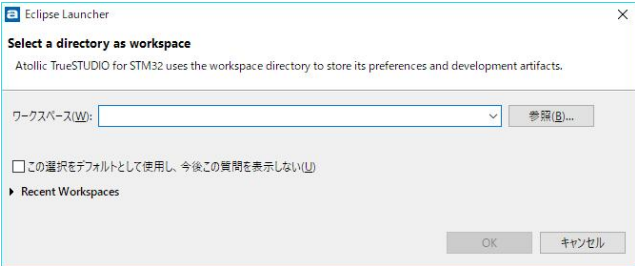


※ Versionにはサンプルプログラムを開発したときのものを記載してあります。
基本的には最新版をインストールすべきだと思いますが、
サンプルプログラムが動かない等の不具合がありましたら参考にして下さい。

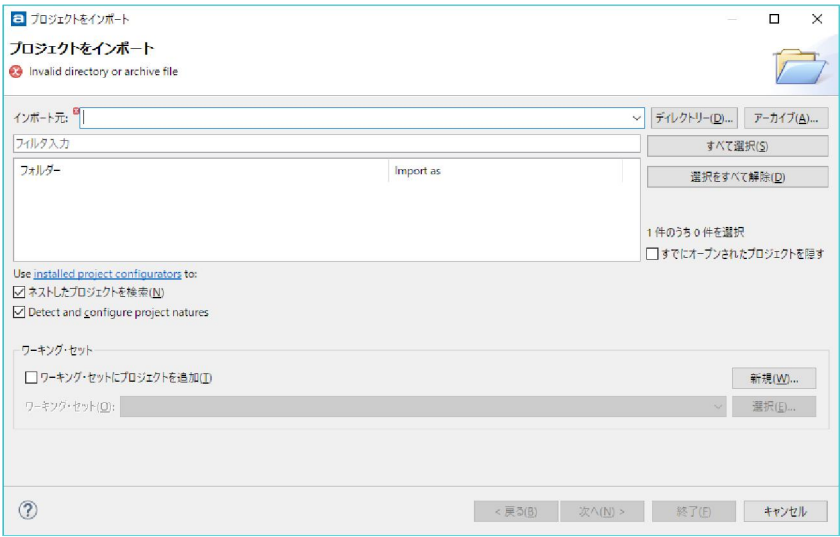
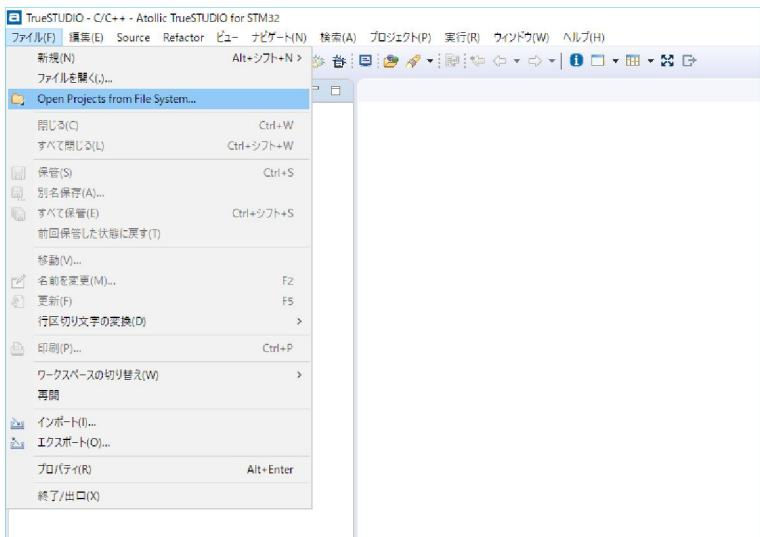
サンプルプログラムのインポート

サンプルプログラムでハードウェアが確実にできているか動作確認をします。

① TrueSTUDIOを起動すると、ワークスペースを指定画面が表示されます。
右の参照ボタンからサンプルプログラムがあるディレクトリを指定します。



② TrueSTUDIOが起動したら、左上のメニューバーから、ファイル→Open Projects from File System...を選択します。
インポート元にサンプルのフォルダを指定し、フォルダー欄にサンプルプログラム名が表示され、横のボックスにチェックが入っていることを確認したら、終了ボタンを押します。

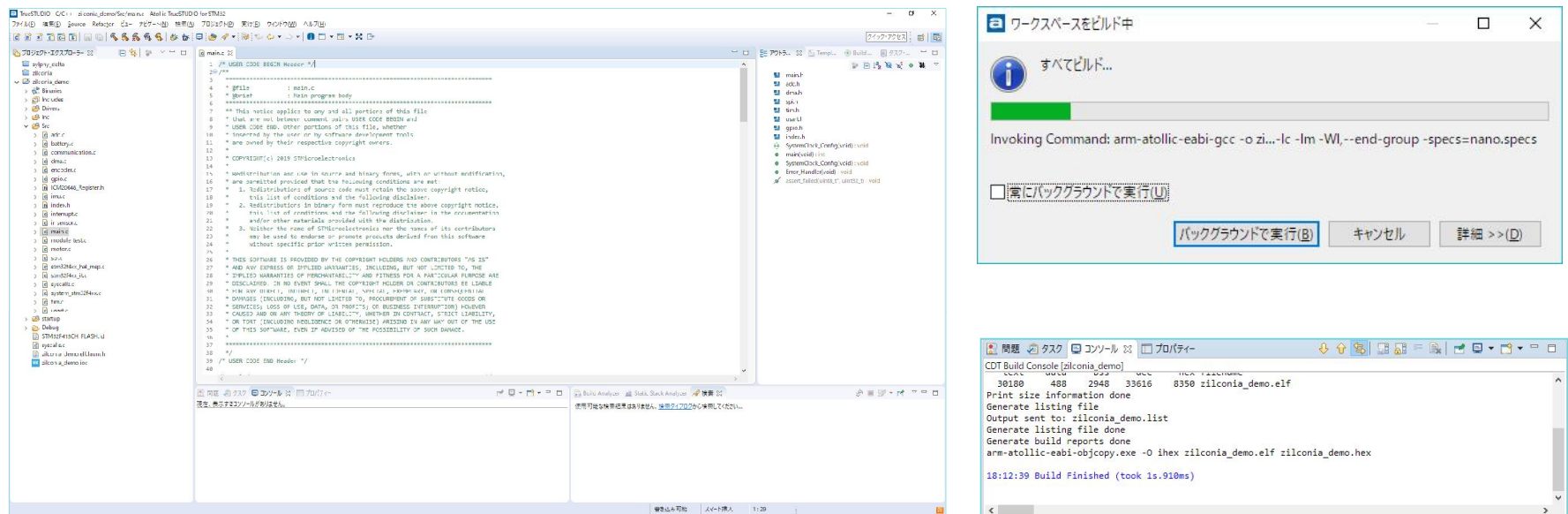


③ 左のプロジェクトツリーにサンプルプログラムがあればインポート成功です。

サンプルプログラムのビルド

① とりあえずmain.cを開いてみましょう。

左のプロジェクトツリーから、zirconia_demo→Src→main.cを開くと、真ん中の画面にソースが表示されます。
※ソースの中身については後ほど解説します。



② サンプルプログラムをビルドします。

左上のメニューバーから、プロジェクト→プロジェクトのビルドを押します。

③ ビルドが始まるとプログレスバーが表示され、

ビルドが正常に完了すると下のコンソール画面に「Build Finished」が表示されます。

失敗した場合は、コンソール画面にエラーが表示されるので、気合でデバッグして下さい。

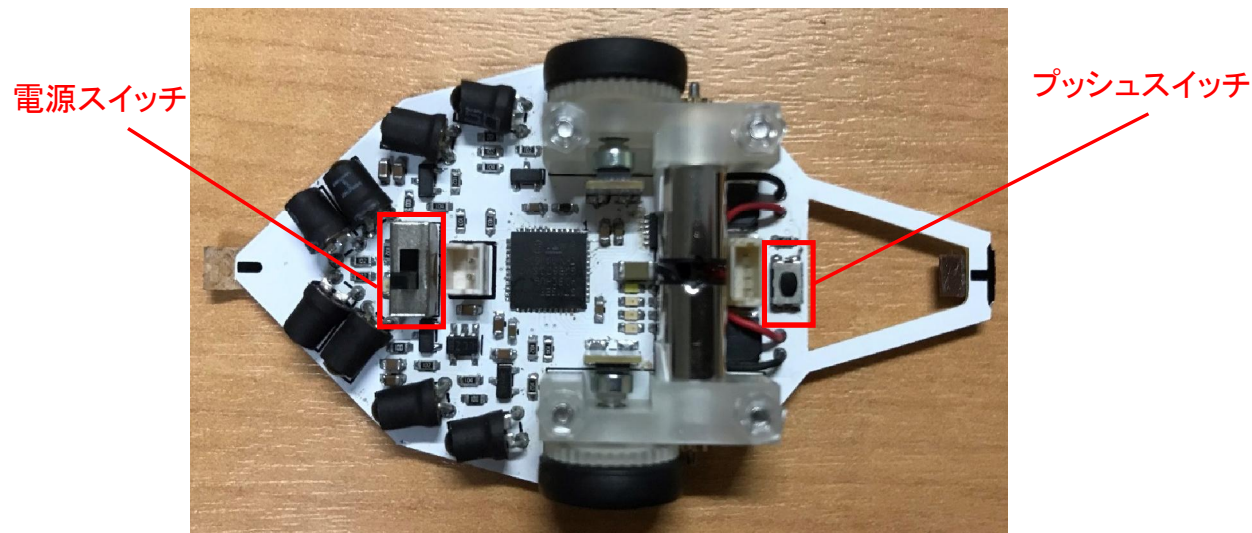
ブートモードでの起動

プログラムを書き込むためにブートモードでマウスを起動します。

① バッテリーと通信ケーブルをマウスと接続します。

※電源はまだ入れてはいけません。

② プッシュスイッチを押しながら電源を入れます。



③ プッシュスイッチを離しても赤色LEDが点灯していれば、ブートモードで起動しています。

※LEDが消灯してしまう場合には、プッシュスイッチを押しっぱなしにしてください。

フラッシュライターとの接続

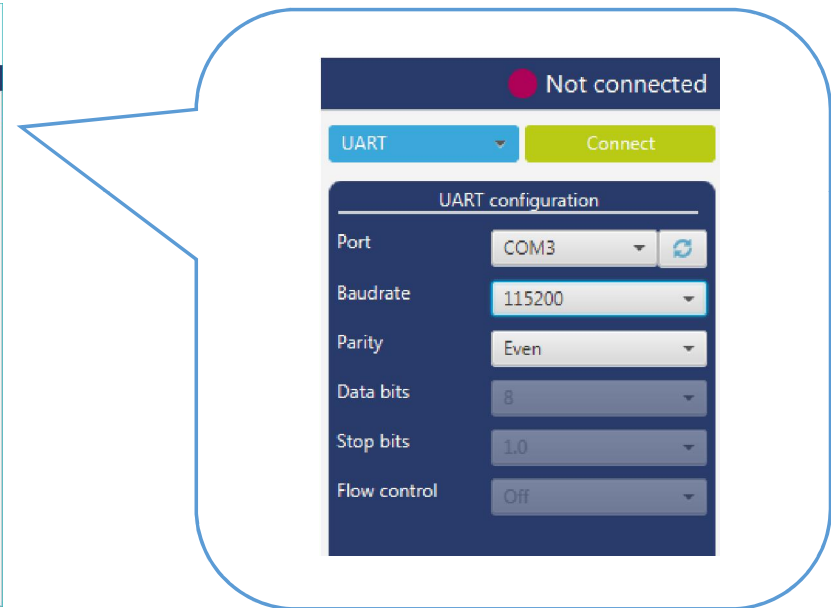
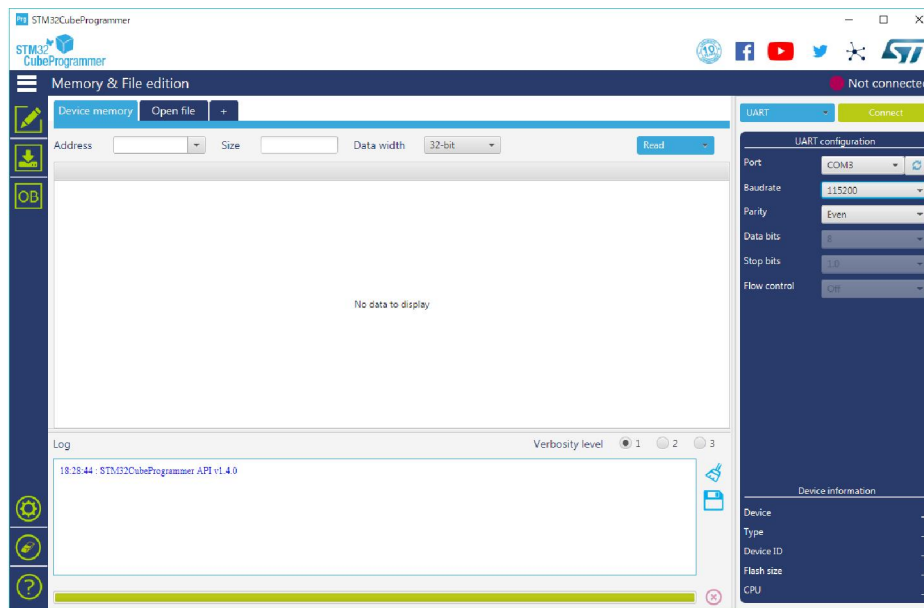
ビルドしたサンプルプログラムを書き込みます。

① STM32CubeProgrammerを起動します。

② 起動したら下図にあるように接続方法を変更します。

※Portはパソコンによって異なるので、Windowsならデバイスマネージャーから通信ケーブルを挿したポートを確認して下さい。

※確認したPortが表示されない場合は、右の更新ボタンを押して接続状態の更新をして下さい。



③ 右上のConnectボタンを押して、右上の赤丸が緑色になり、Connectedと表示されれば接続成功です。

※接続できない場合は、何度かブートモードで再起動させてから、接続し直して下さい。

※それでもダメならハードウェアが間違っている可能性があります。

サンプルプログラムの書き込み

先程ビルドしたサンプルプログラムをマウスに書き込みます。

① 左のアイコンからErasing&Programmingを選択します。

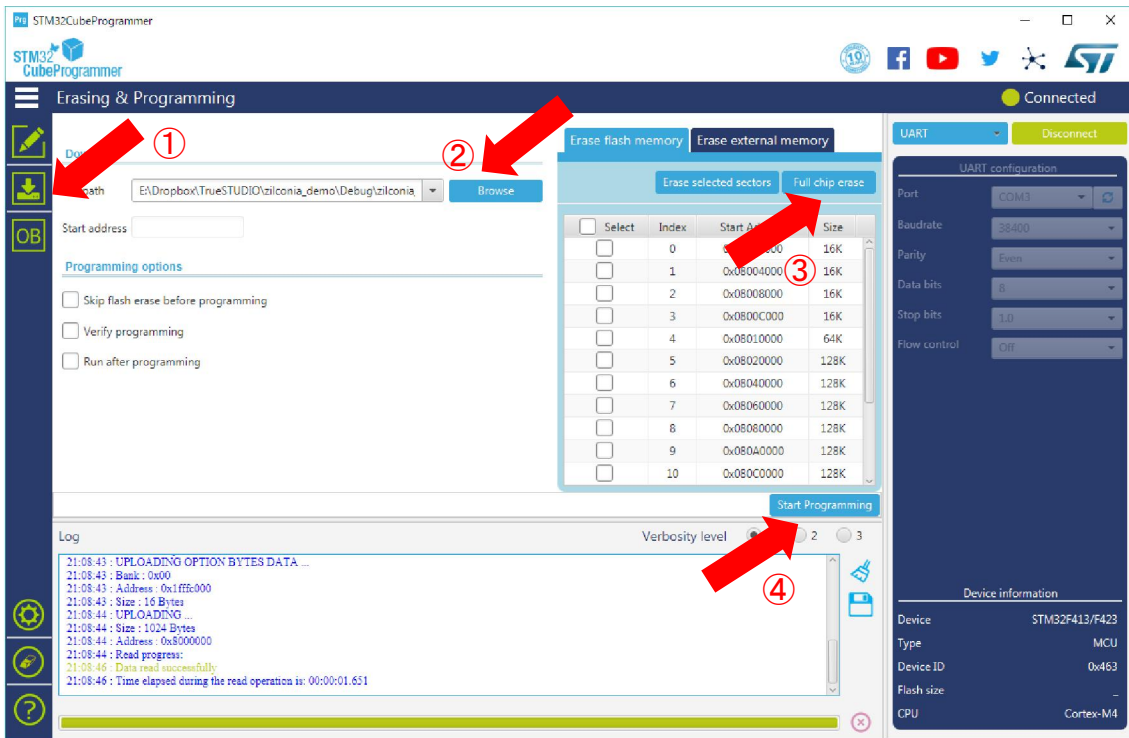
② Download欄にあるBrowseボタンから
サンプルプログラムのDebugフォルダにある
zirconia_demo.hexを指定します。

③ マイコンにプログラムを書き込むために、
フラッシュ領域を初期化する必要があります。
真ん中上にあるFull Chip Eraseボタンを押して、
初期化が完了するまで待ちます。

④ 初期化後にStart Programmingボタンを押して、
マイコンにプログラムを書き込みます。

⑤ File download completeと
メッセージが表示されれば、
プログラムの書き込み成功です。

⑥ 右上のDisconnectボタンを押して、フラッシュライターとの接続を解除し、マウスの電源を落とします。



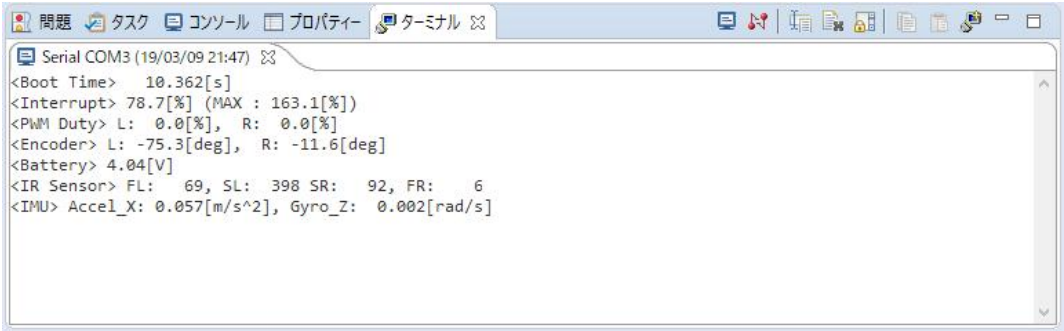
サンプルプログラムの実行

サンプルプログラムを実行してハードウェアの動作確認をします。
※ここで動作がおかしいものがあればハードウェア不良の可能性があります。

- ① TrueSTUDIOのターミナルを起動します。
メニューバーのOpen a Terminalボタンを押すか、Ctrl+Alt+Shift+Tで起動します。
※TeraTermなどのターミナルソフトでも問題ありません。
- ② ターミナルの設定は右図のように設定します。
- ③ ターミナルが起動したら、マウスの電源を入れると下図のように表示されます。



- ・Boot Timeは起動してからの経過時間
- ・Interruptは割り込み時間内の処理割合
- ・PWM Dutyはモータに加わっているDuty比
※モータはキーボードの入力で回転できます
キーボードのRでストップ
- ・Encoderはタイヤの回転角度
- ・Batteryはバッテリーの電圧
- ・IR Sensorは各壁センサのAD値
- ・IMUは前後加速度とヨー方向の角速度



- ④ キーボードのEscでプログラムを終了します。

	左モータ			右モータ		
Duty	1%	10%	100%	1%	10%	100%
正転	1	2	3	7	8	9
逆転	Q	W	E	U	I	O

サンプルプログラムの使い方

このサンプルプログラムはマウスに必要な初期設定をブラックボックス化して、上位のアプリケーション開発に集中できるように関数化してあります。

① STM32CubeMXの注意点

このサンプルプログラムは、STM32CubeMXの自動コード生成機能を利用して書かれているため、マイコンの各ペリフェラルの設定を変更する際にはSTM32CubeMX上から変更して下さい。

また、オリジナルのコードは/* USER CODE BEGIN ○○○ */から
/* USER CODE END ○○○ */と書かれたコメントの間に記述して下さい。
それ以外の場所にコードを記述した場合、STM32CubeMXで再生成した際に削除されてしまいます。

② index.h

サンプルプログラムで使われているマクロや関数はすべて上記ヘッダーファイルに宣言されているので、サンプルプログラムを利用する際にはindex.hをインクルードして下さい。
また、各関数の使い方についてコメントしてありますので、ご活用下さい。

③ メイン関数と割り込み関数

サンプルプログラムでは、main.cにメイン関数、interrupt.cに割り込み関数を記載しています。
オリジナルコードを記述する場所をコメントにて指示してあります。

④ printfとscanf

このサンプルプログラムでは、studio.hをインクルードすることで、
ターミナルと通信するための関数として、printfとscanf関数を使用することができます。