

EPC9118-20 - EPC91122

Motor Commissioning Quick Start Procedure



EFFICIENT POWER CONVERSION

OVERVIEW OF THE PROCESS

- 1. Procedure targets and requirements**
2. Measuring the motor parameters
3. Creating a ST MCW6 project
4. Generating and flashing the firmware
5. Operating the motor drive system

1. Procedure targets and requirements

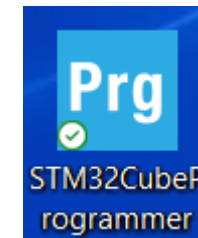
- This guide will focus on a brief description of the main steps required to drive a generic PMSM with the EPC91118-20-22 boards, both using a sensorless algorithm or exploring the magnetic encoder embedded in the inverter for a more accurate control.
- Since the microprocessor for this specific board has been chosen from the STM32 family, the motor control firmware can be easily generated using the ST motor control development suite. In particular, it is recommended to use the newest version of the STM32 MC SDK (v6.4). Refer to the following link to install the software: [X-CUBE-MCSDK - STM32 Motor Control Software Development Kit \(MCSDK\) – STMicroelectronics](#).
- If it is the first time using the ST motor control workbench (MCW6), please refer to this manual: [How to use STM32 motor control SDK v6.0 workbench - User manual](#)



1. Procedure targets and requirements

Below are listed the ST website links from which the user must install the two additional software to correctly generate, flash and run the motor control firmware:

- STM32CubeMX (initialization code generator): <https://www.st.com/en/development-tools/stm32cubemx.html>
- STM32CubeIDE (Integrated Development Environment for STM32): <https://www.st.com/en/development-tools/stm32cubeide.html>
- OPTIONAL STMCubeProgrammer (for quick programming using .elf files): [STM32CubeProg | Software - STMicroelectronics](https://www.st.com/en/development-tools/stm32cube-programmer.html)



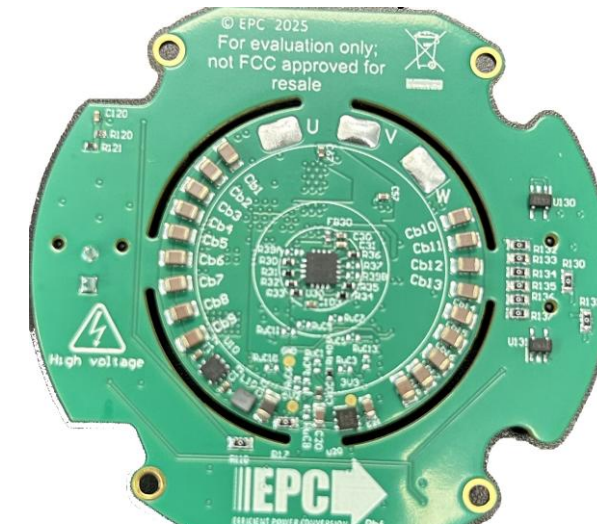
1. EPC91118-20-22 inverter board

The EPC91118-20 and EPC91122 inverter boards are similar and are 3-phase inverters for humanoid robot joints.

EPC91122 features a 3-phase module EPC33110
 EPC91118 features three EPC23104 ePower Stage IC.

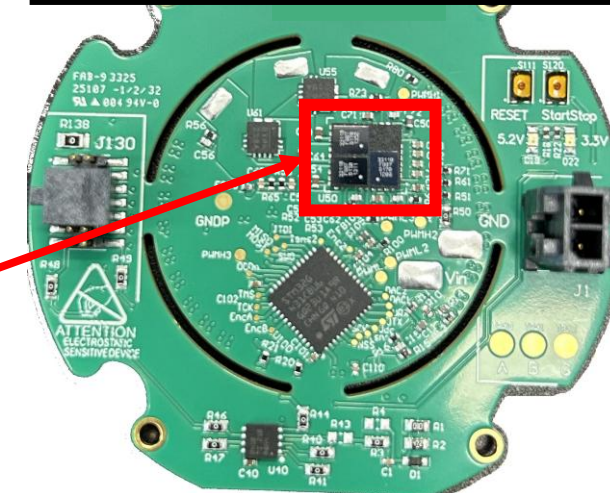
EPC91120 features three EPC23102 ePower Stage IC.

The EPC91118 Quick Start Guide is available at the link: [EPC91118_qsg.pdf](#)



top

EPC91122 board



bottom

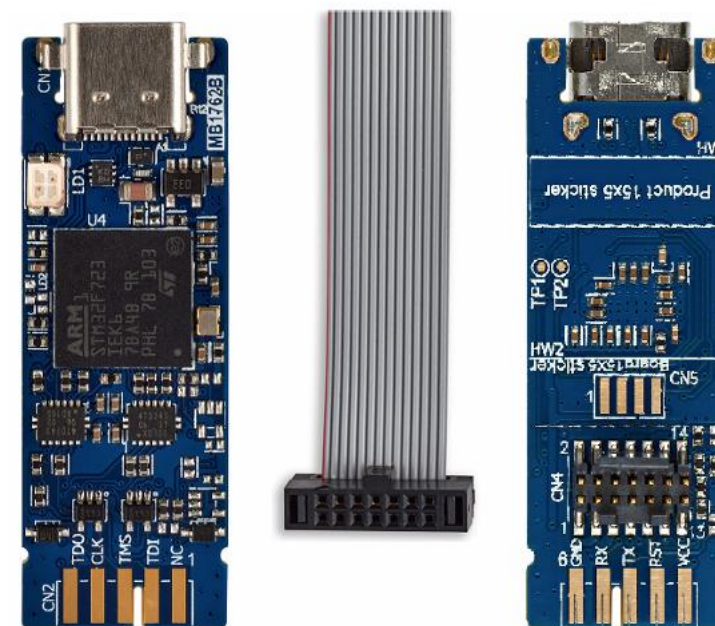
EPC33110

1. STLINK-V3MINIE

To properly communicate with the microcontroller located on the EPC91118-20-22 board, it is recommended to use the STLINK-V3MINIE debugger/programmer with the tiny probe for STM32 microcontrollers, as shown in the figure.

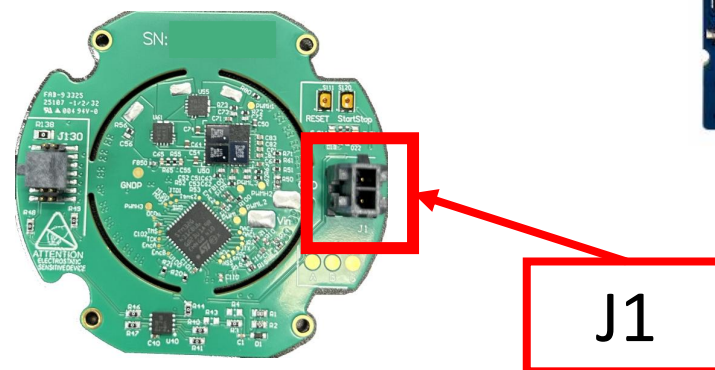
After arranging the JTAG/SWD interface between the two boards, it is essential to power on the inverter from the connector **J1** to supply the microcontroller with 3.3V (check the yellow LED on the EPC91122).

STLink-V3mini



top

bottom



OVERVIEW OF THE PROCESS

1. Procedure targets and requirements
- 2. Measuring the motor parameters**
3. Creating a ST MCW6 project
4. Generating and flashing the firmware
5. Operating the motor drive system

2. Motor specifications

Before starting the creation of a new project on the motor control workbench, it is essential to know the motor parameters, which can be obtained from the motor datasheet or through simple experimental measures. On the first pages (3-6) of the following guide, it is shown how to proceed: [Commissioning a motor using ST Motor control.pdf](#).

The main motor parameters required to insert into the project are the following:

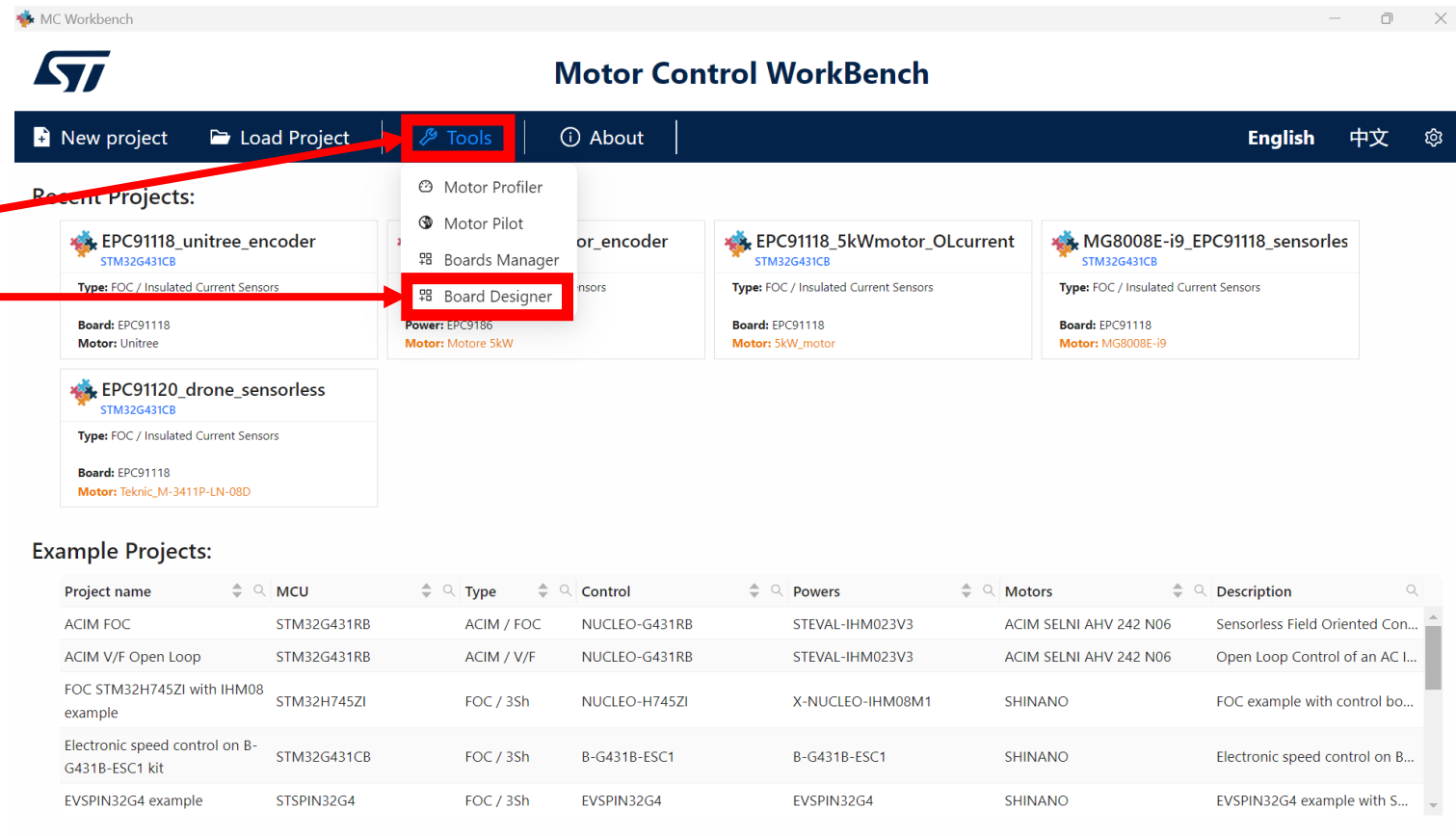
- Phase resistance (Ω)
- Phase inductance (mH)
- Pole pairs
- B-Emf constant ($V_{rms}/krpm$)

OVERVIEW OF THE PROCESS

1. Procedure targets and requirements
2. Measuring the motor parameters
- 3. Creating a ST MCW6 project**
4. Generating and flashing the firmware
5. Operating the motor drive system

3. Opening the tools from MCW6

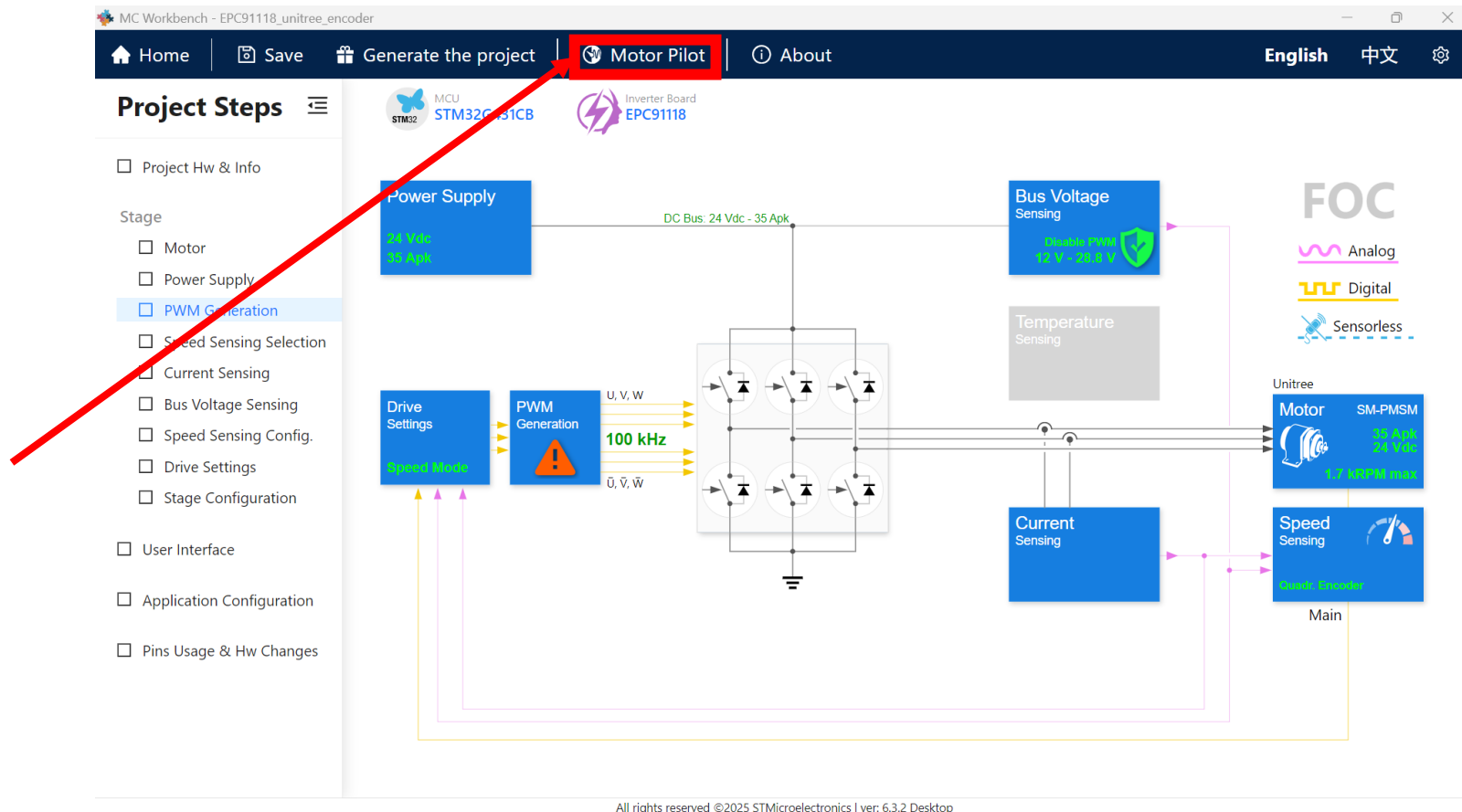
- To open the STMC board designer (application used to configure the driver board), click on 'Tools' and then select 'Board Designer'



All rights reserved ©2025 STMicroelectronics | ver: 6.3.2 Desktop

3. Opening the tools from MCW6

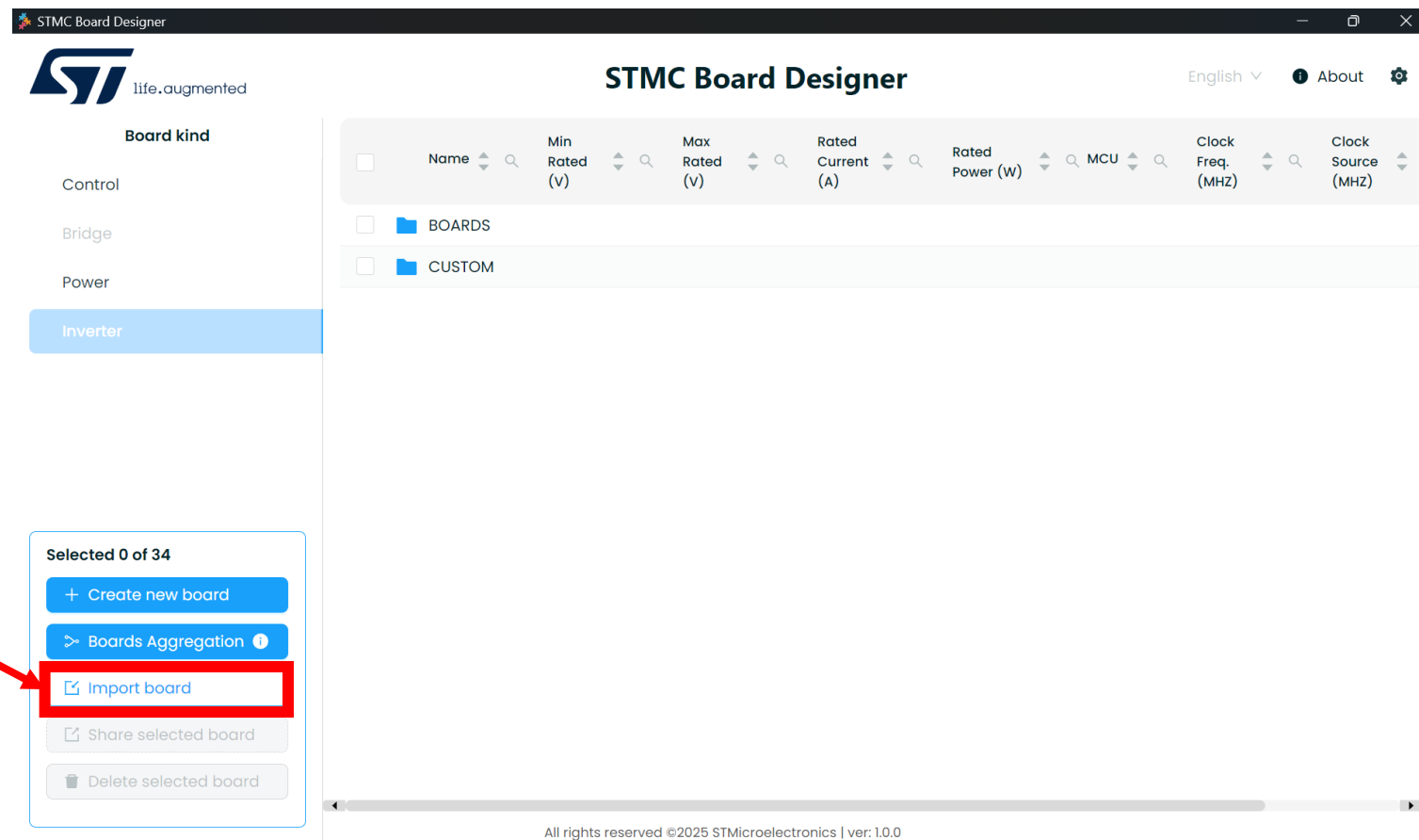
- To open the STMC board designer (application used to configure the driver board), click on 'Tools' and then select 'Board Designer'
- To open the STMC motor pilot (application used to drive the motor), click on 'Motor Pilot' inside the MCW project



All rights reserved ©2025 STMicroelectronics | ver: 6.3.2 Desktop

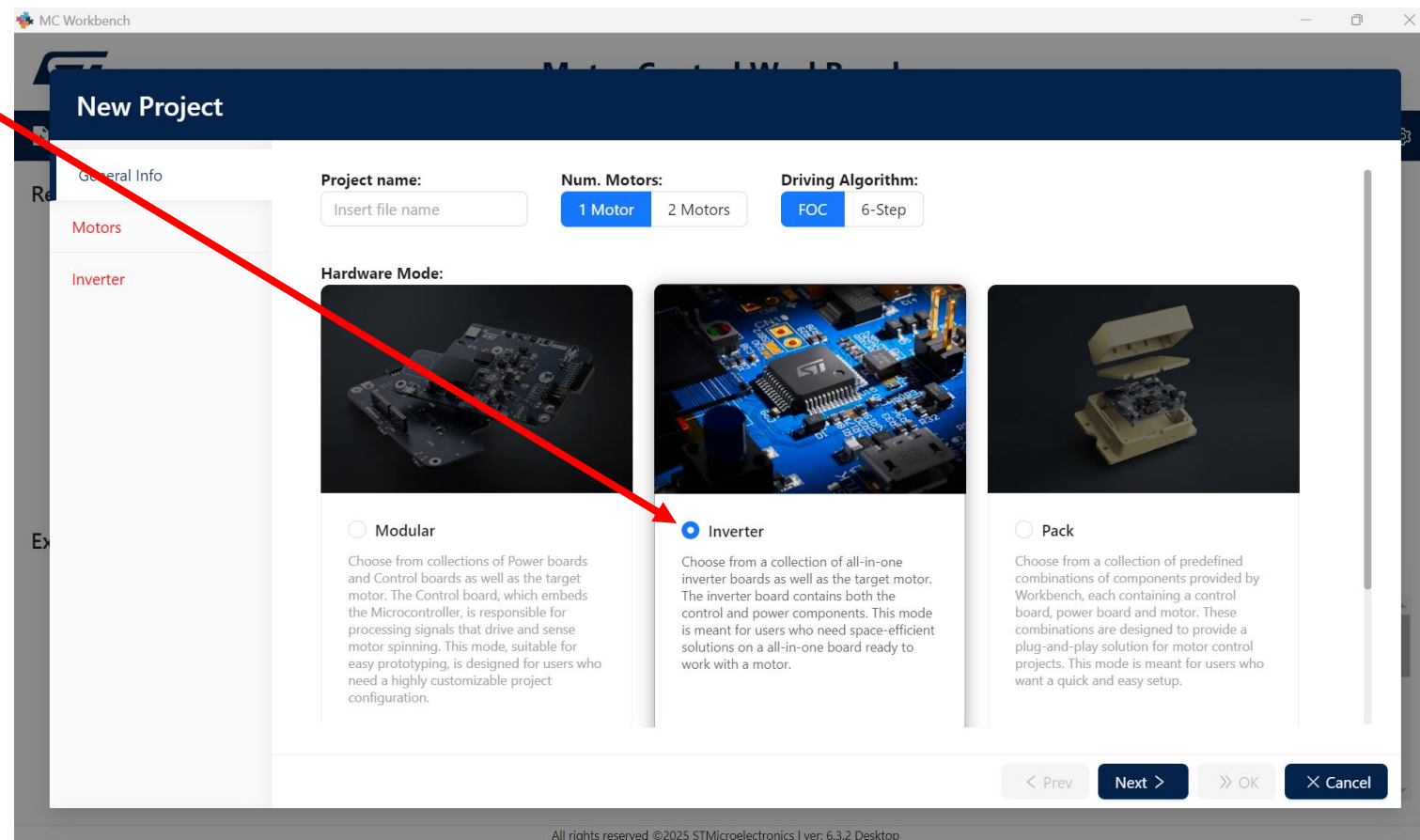
3. ST MCW6 project

- Download from this repository the .json file (EPC91118.json that is the same for EPC91118-20-22 boards) of the inverter to be used in the project: [GitHub - epc-co/MotorDrive-RefDesign-Firmware: EPC's Motor Drive Reference Designs Firmware repository](https://github.com/epc-co/MotorDrive-RefDesign-Firmware)
- Import the .json file of the board to the STMC board designer, which is a tool inside MCW6.



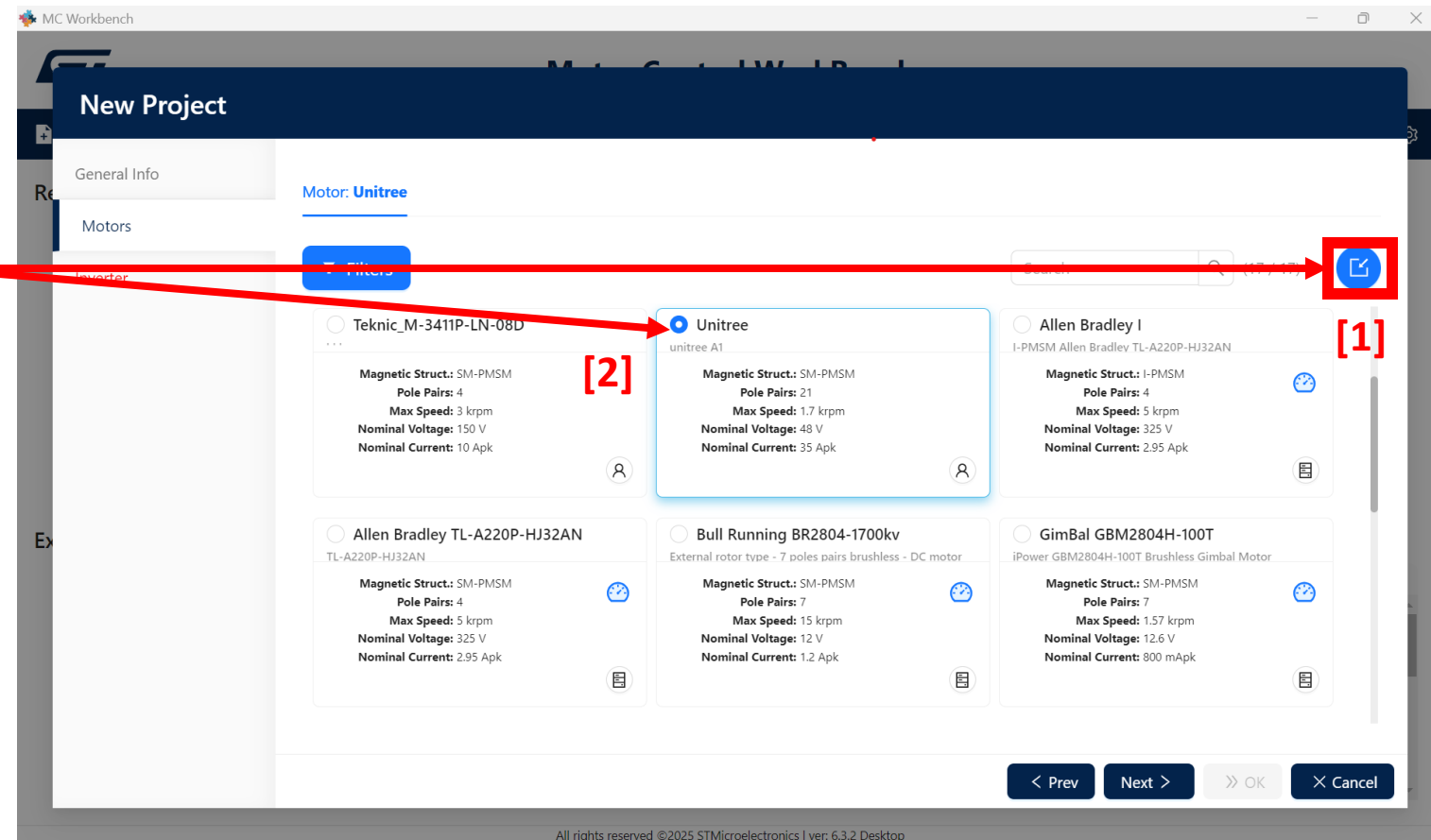
3. ST MCW6 project

- Make a new project, choosing 'Inverter' as Hardware Mode



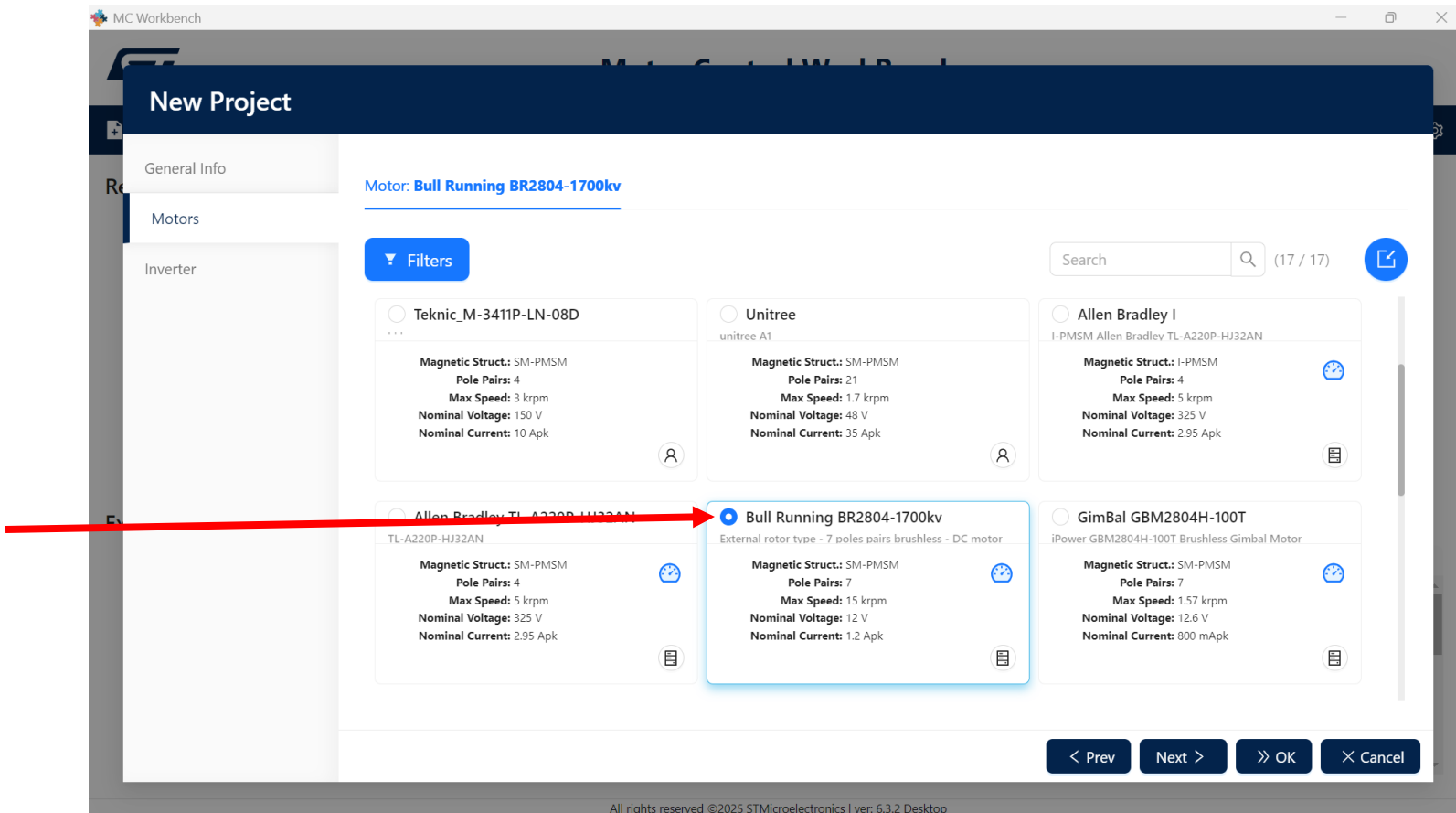
3. ST MCW6 project

- Make a new project, choosing 'Inverter' as Hardware Mode
- [1] Import the .json file of the motor and then [2] choose it in the list



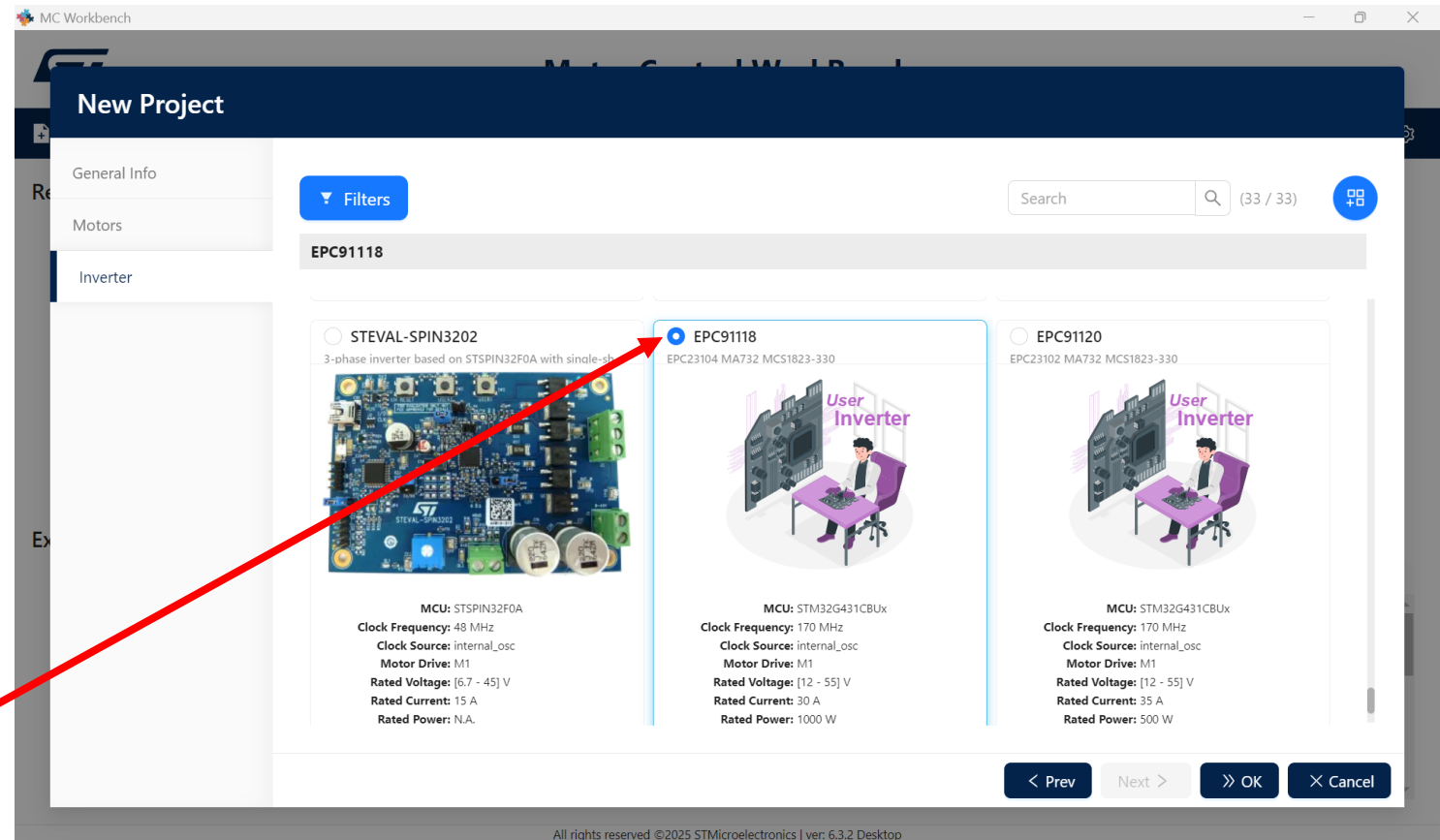
3. ST MCW6 project

- Make a new project, choosing 'Inverter' as Hardware Mode
- **[1]** Import the .json file of the motor and then **[2]** choose it in the list
- Otherwise choose a motor profile provided by ST and then you will change the parameters inside the project



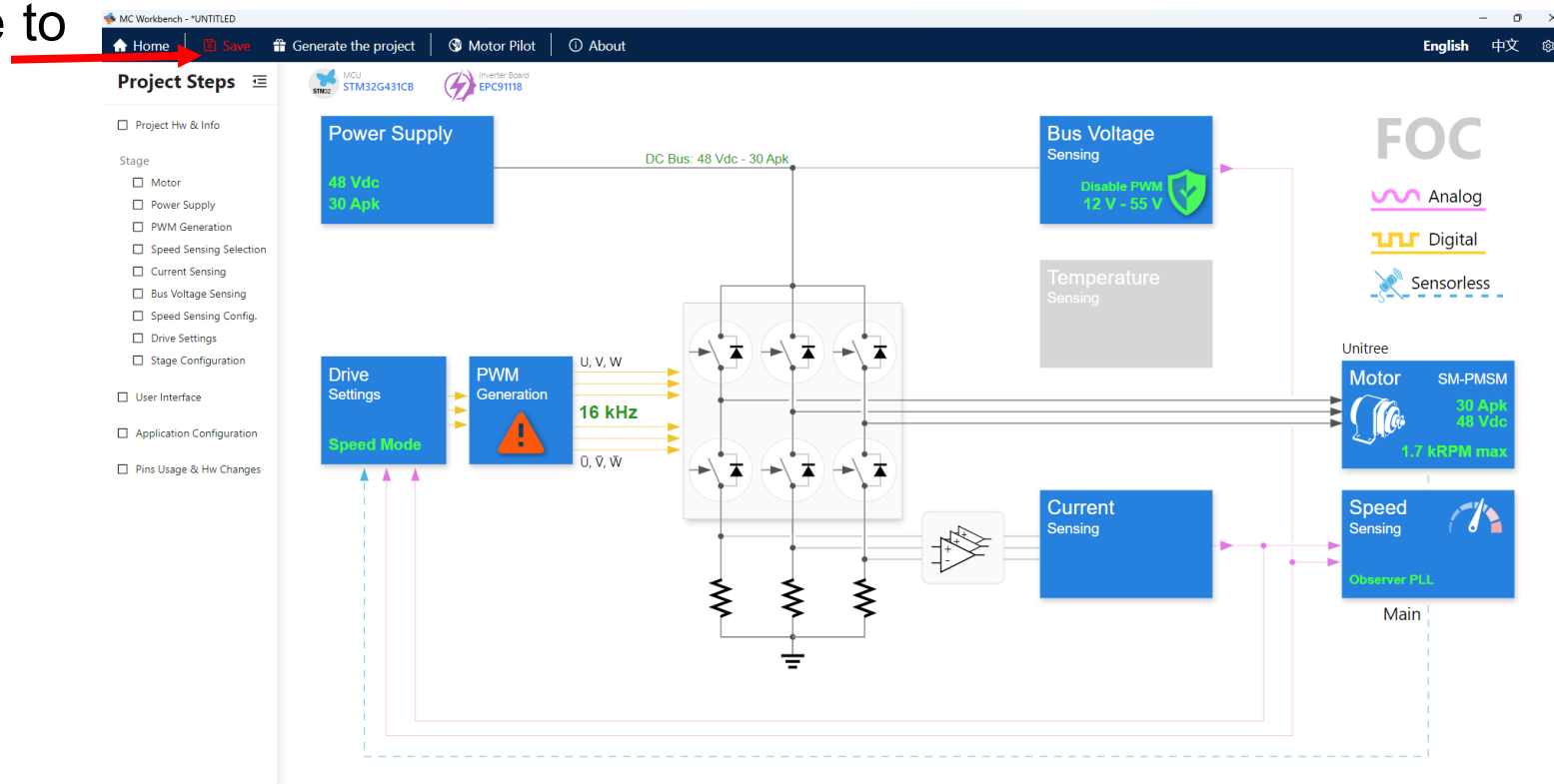
3. ST MCW6 project

- Make a new project, choosing 'Inverter' as Hardware Mode
- **[1]** Import the .json file of the motor and then **[2]** choose it in the list
- Otherwise choose a motor profile provided by ST and then you will change the parameters inside the project
- Choose the inverter board



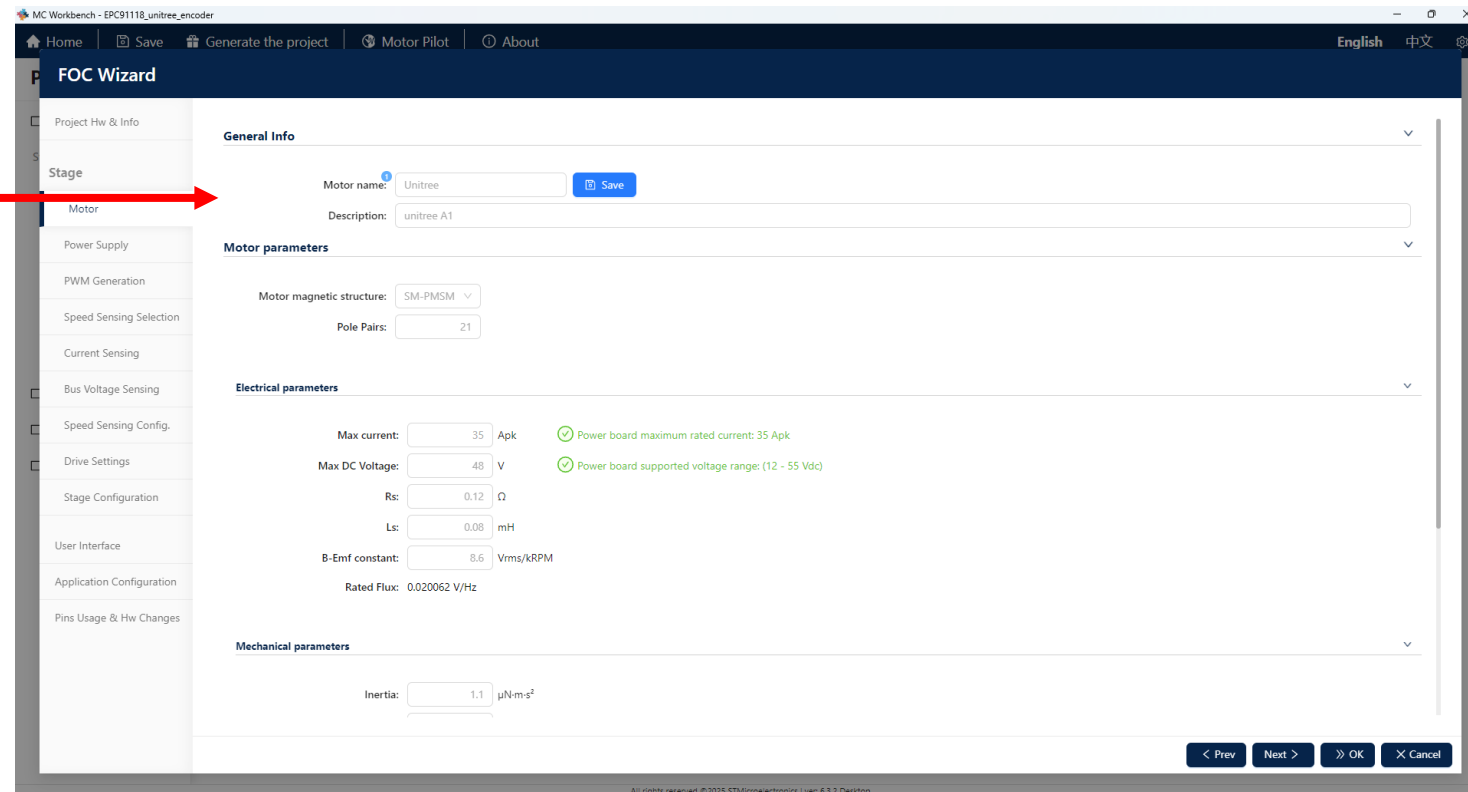
3. ST MCW6 project

- Save the file project to be able to generate the code



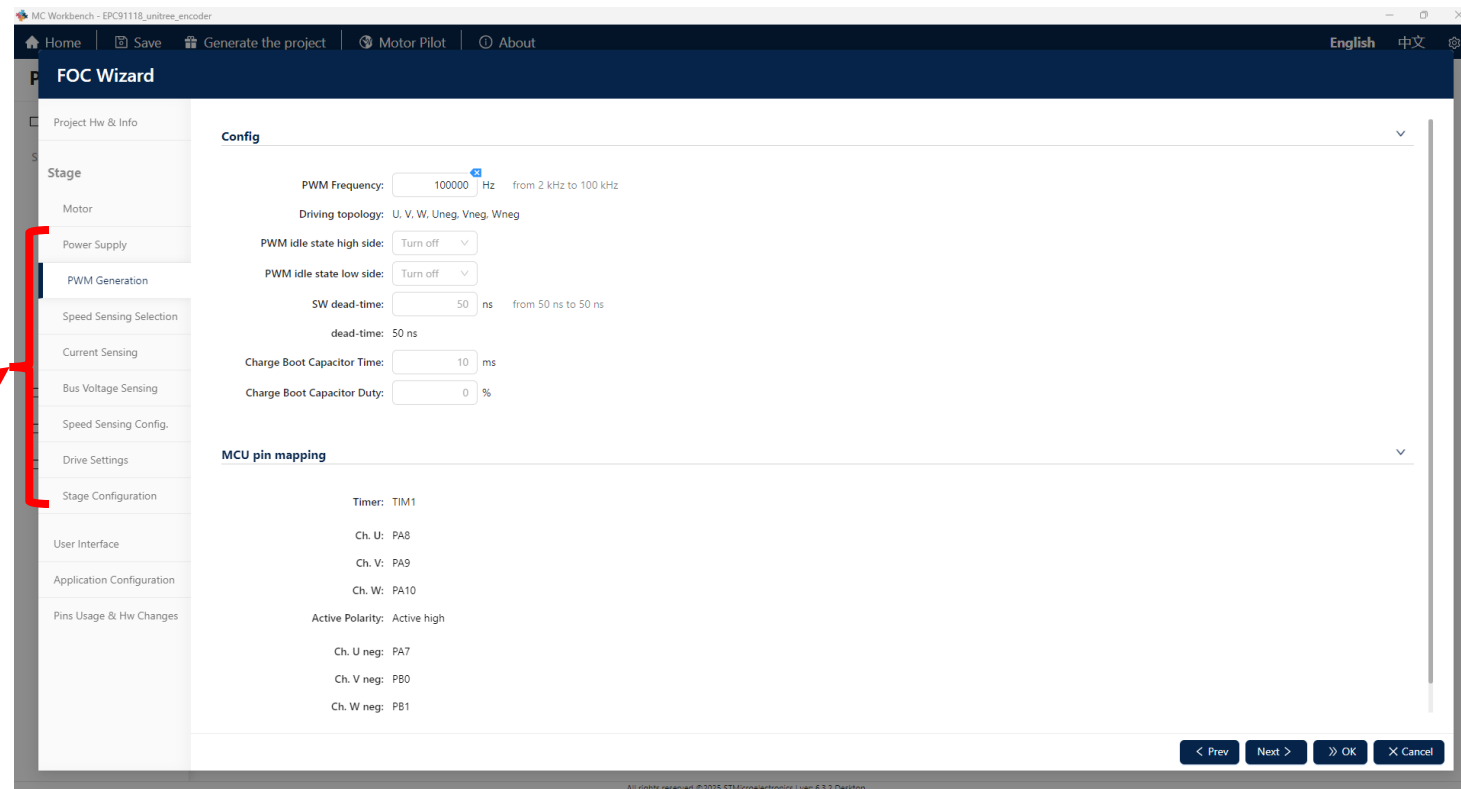
3. ST MCW6 project

- Save the file project to be able to generate the code
- In 'Motor' window you can change the motor parameters and possibly save them into a new .json file



3. ST MCW6 project

- Save the file project to be able to generate the code
- In 'Motor' window you can change the motor parameters and possibly save them into a new .json file
- For the configuration of other sections, please refer to the previously mentioned repository, in which there are two projects configured for EPC91118 and two different motors



3. ST MCW6 project

- For a sensorless motor control use the .stwb file inside this folder as a model, especially for 'speed sensing configuration' and 'drive settings'

[GitHub - epc-co/MotorDrive-RefDesign-Firmware: EPC's Motor Drive Reference Designs Firmware repository](https://github.com/epc-co/MotorDrive-RefDesign-Firmware)

Image: courtesy of Teknic, Inc.

Useful Resources

- Motor Website

Cross-Reference Table

Use this table to find the firmware package that match you inverter and your microcontroller.

Boards w/ integrated MCU

	Teknic Motor (DummyNema) STMCWB6 Desc (JSON)	Unitree A1 Motor STMCWB6 Desc (JSON)
EPC91118 STMCWB6 Desc (JSON)	Executables Source Code Sensorless	Executables Source Code Sensored

Boards w/ MCU interface connector

	Microchip MA330031-2 PIM	ST STM32 Nucleo G431RB	RENESAS Renesas RA6T2	TI TI LaunchXL-F28069M
Adapter Board →	EPC9147A	EPC9147C	EPC9147D	EPC9147B

3. ST MCW6 project

- For a sensorless motor control use the .stwb file inside this folder as a model, especially for ‘speed sensing configuration’ and ‘drive settings’
- If the motor shaft has a magnet to interface the encoder, look at this reference project

[GitHub - epc-co/MotorDrive-RefDesign-Firmware: EPC's Motor Drive Reference Designs Firmware repository](https://github.com/epc-co/MotorDrive-RefDesign-Firmware)

Image: courtesy of Teknic, Inc.

Useful Resources

- [Motor Website](#)

Cross-Reference Table

Use this table to find the firmware package that match you inverter and your microcontroller.

Boards w/ integrated MCU

	Teknic Motor (DummyNema) STMCWB6 Desc (JSON)	Unitree A1 Motor STMCWB6 Desc (JSON)
EPC91118 STMCWB6 Desc (JSON)	Source Code Sensorless	Executables Source Code Sensored

Boards w/ MCU interface connector

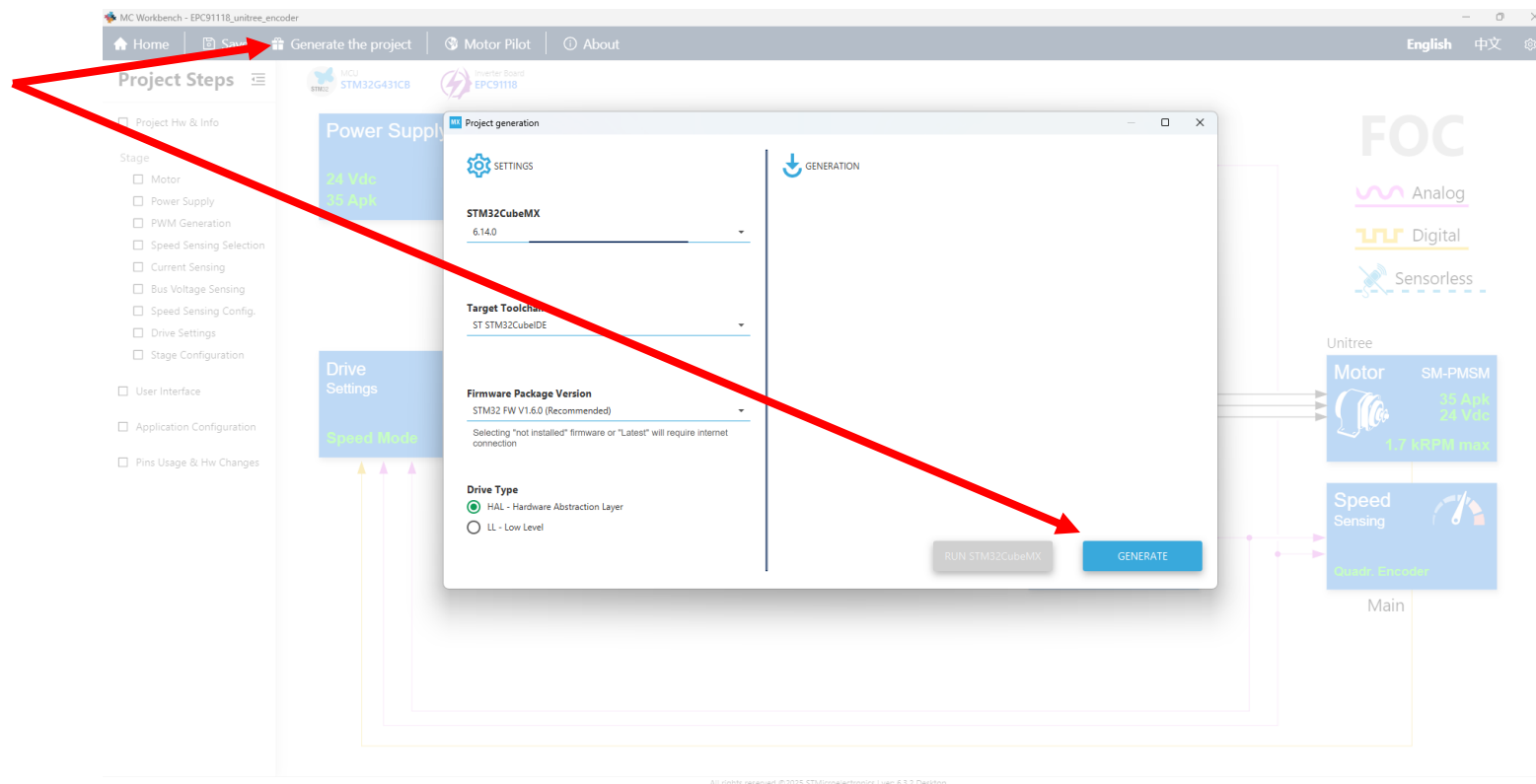
	Microchip MA330031-2 PIM	ST STM32 Nucleo G431RB	RENESAS Renesas RA6T2	TI TI LaunchXL-F28069M
Adapter Board →	EPC9147A	EPC9147C	EPC9147D	EPC9147B

OVERVIEW OF THE PROCESS

1. Procedure targets and requirements
2. Measuring the motor parameters
3. Creating a ST MCW6 project
- 4. Generating and flashing the firmware**
5. Operating the motor drive system

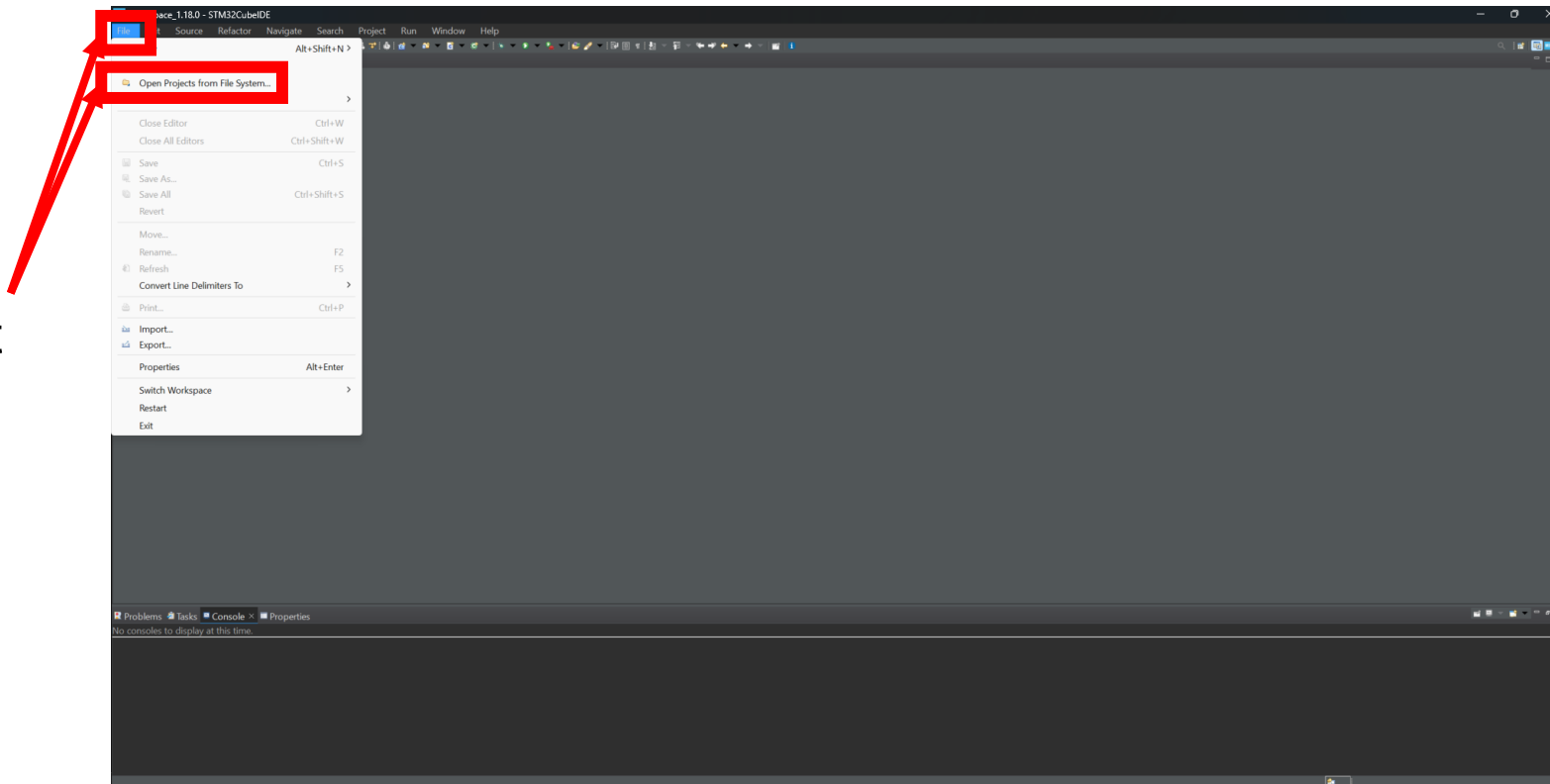
4. Generating the firmware

- Generate the code through STM32CubeMX interface inside MCW project



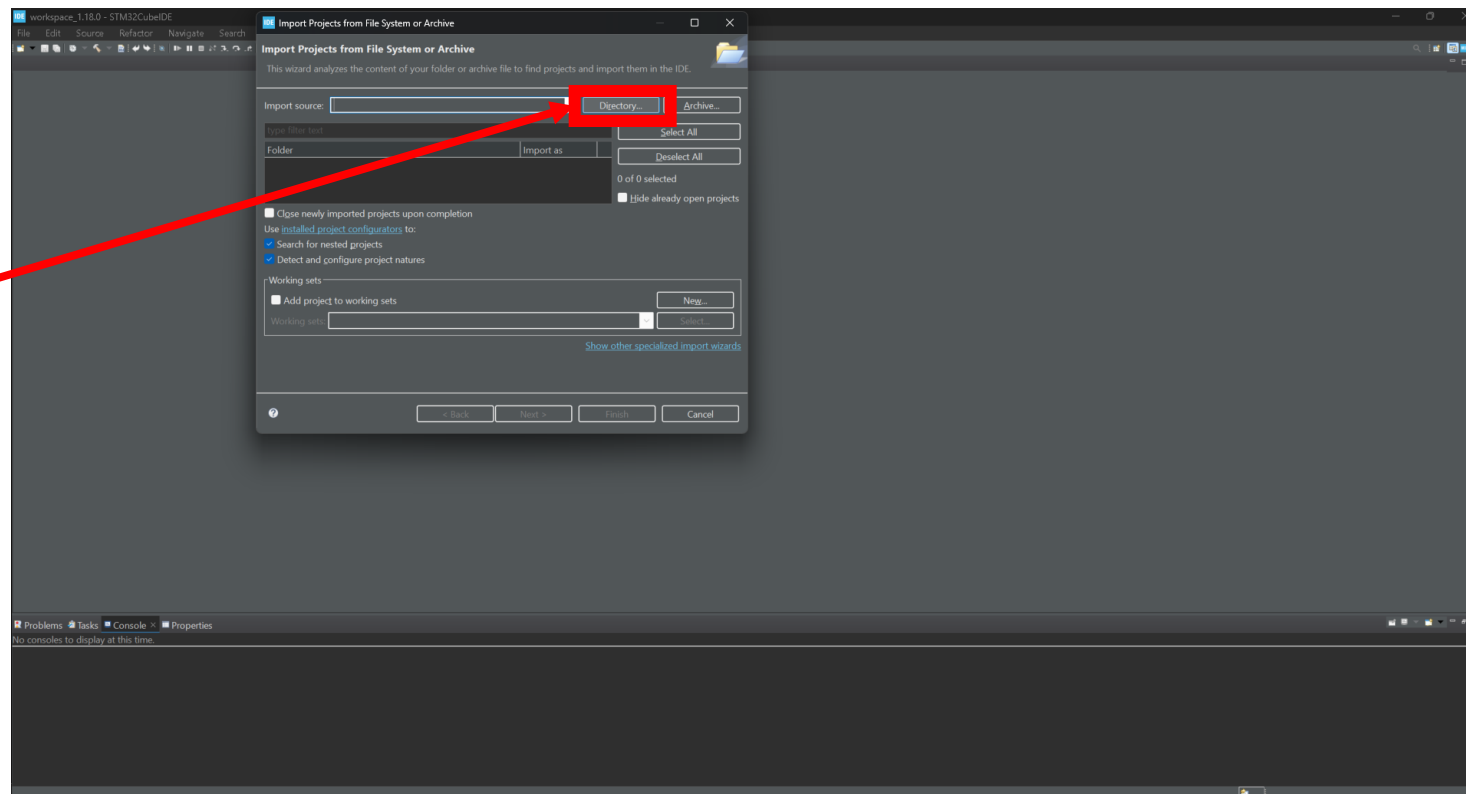
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE: click 'File' and then 'Open Project from File systems'



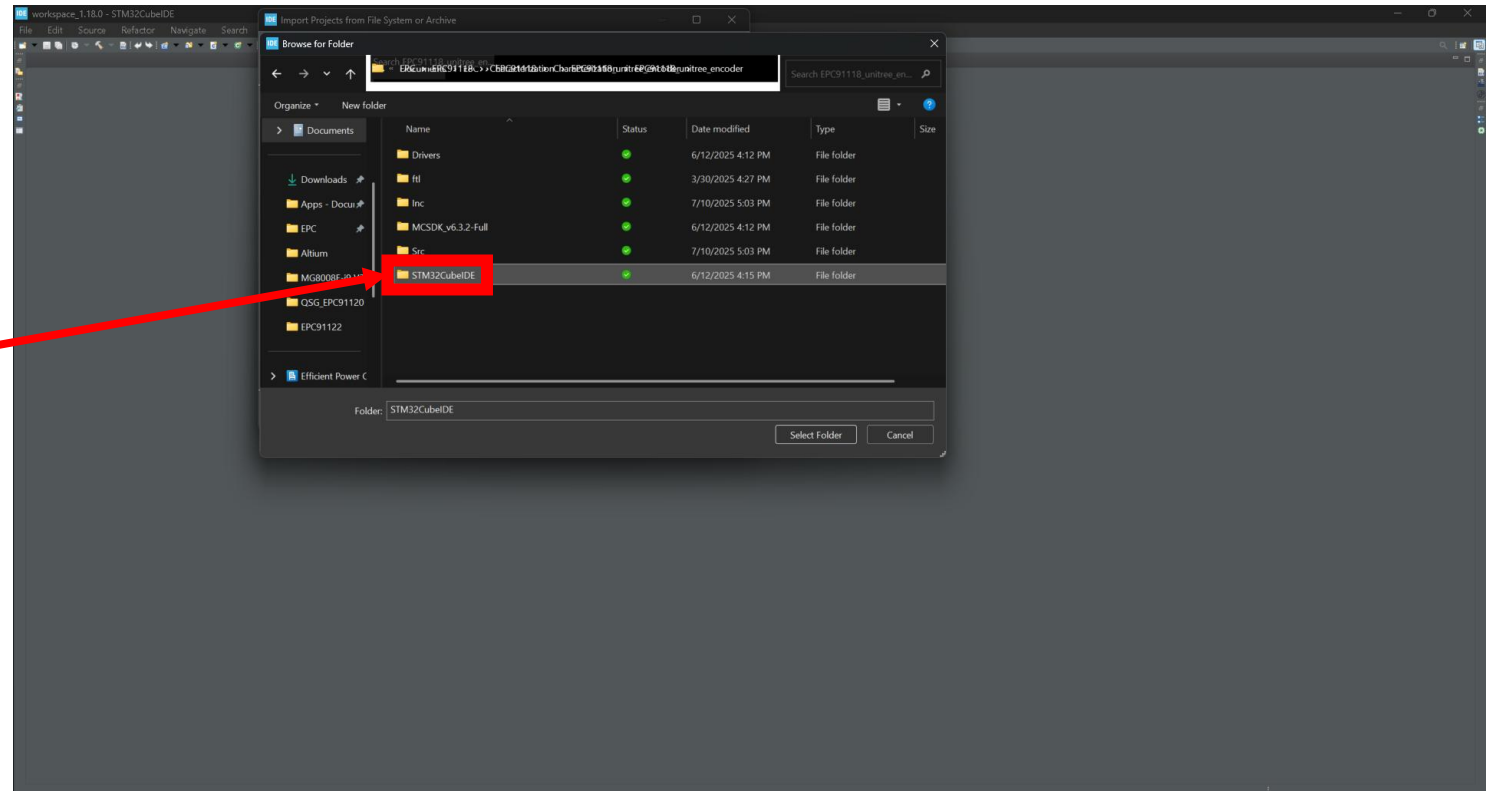
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE: click 'Directory...'



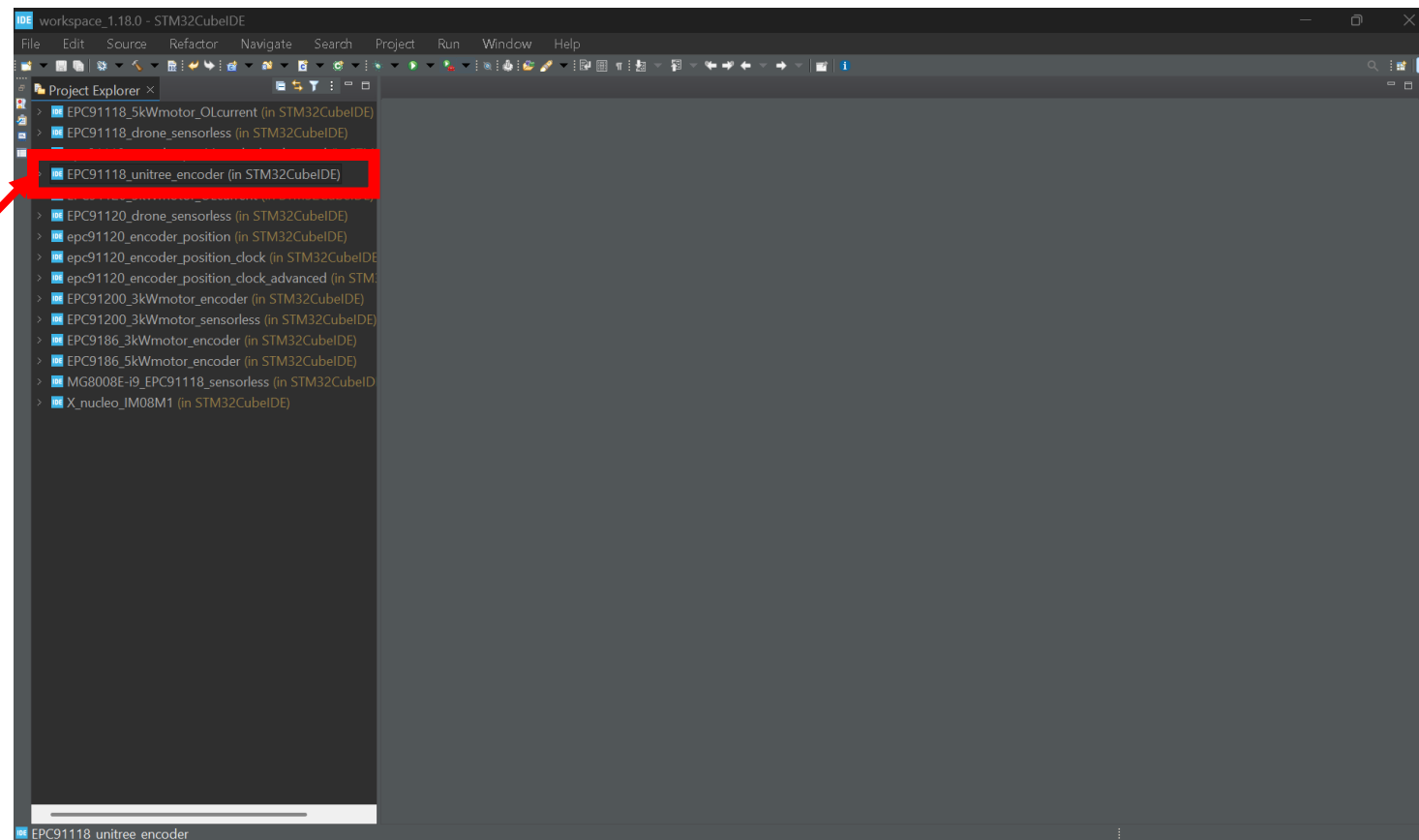
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE: select 'STM32CubeIDE' subfolder from the project folder created after the code generation



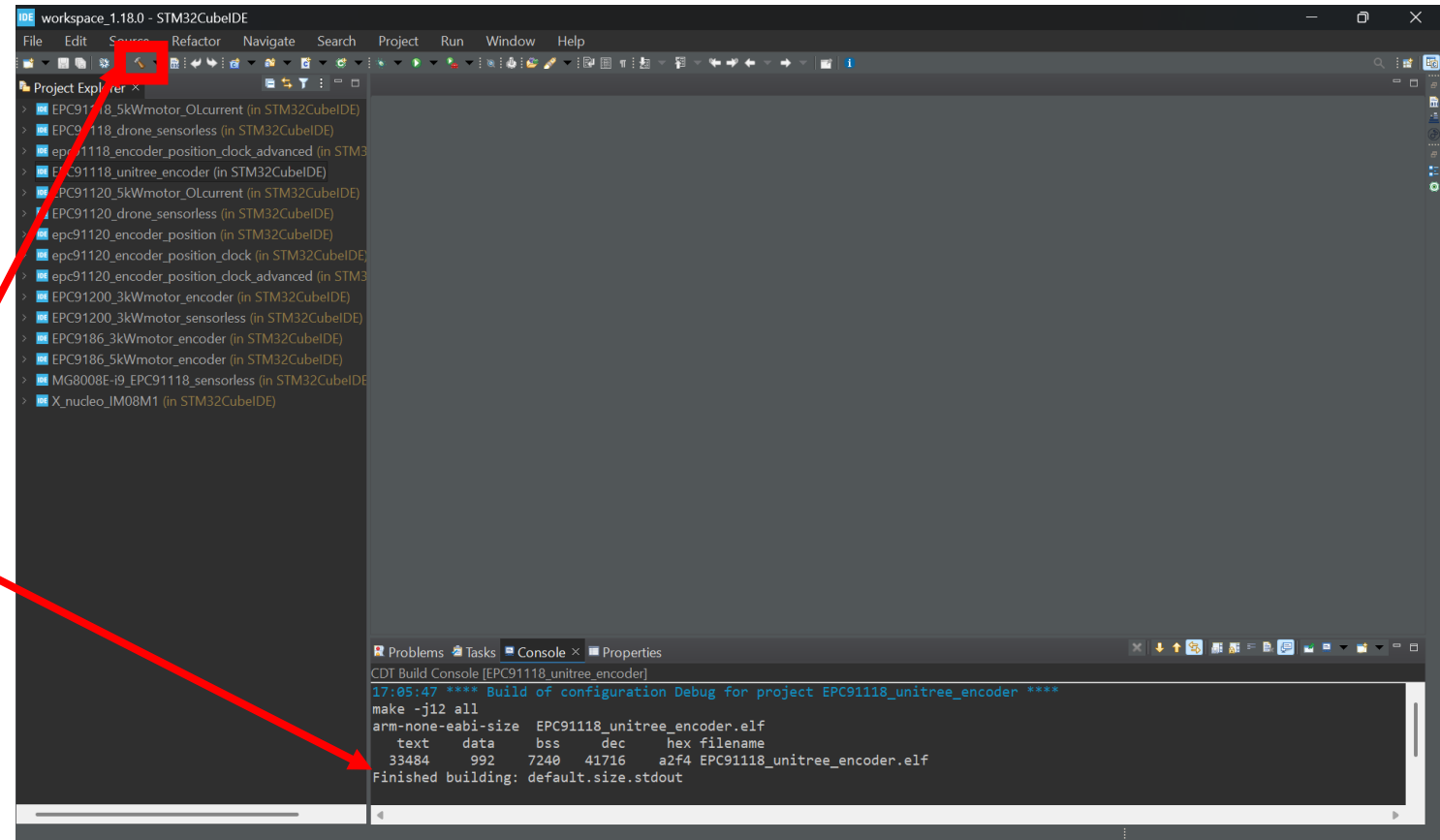
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer



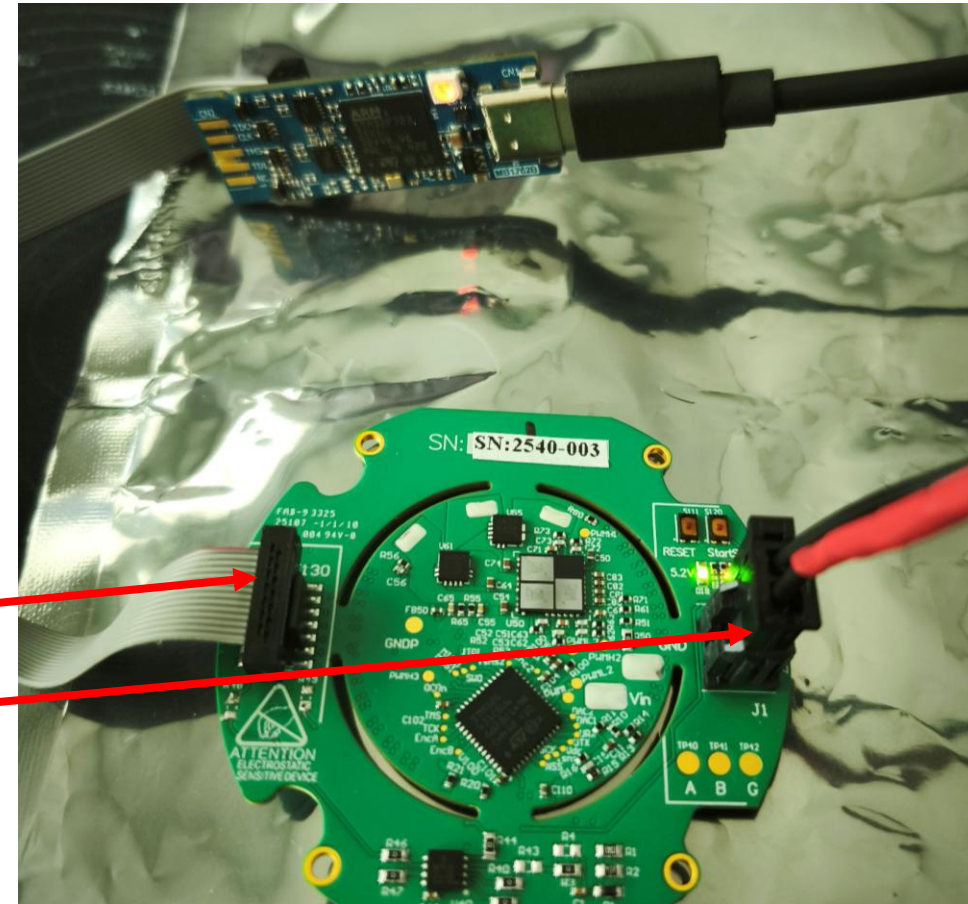
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer
- Build the code and verify the result of the process



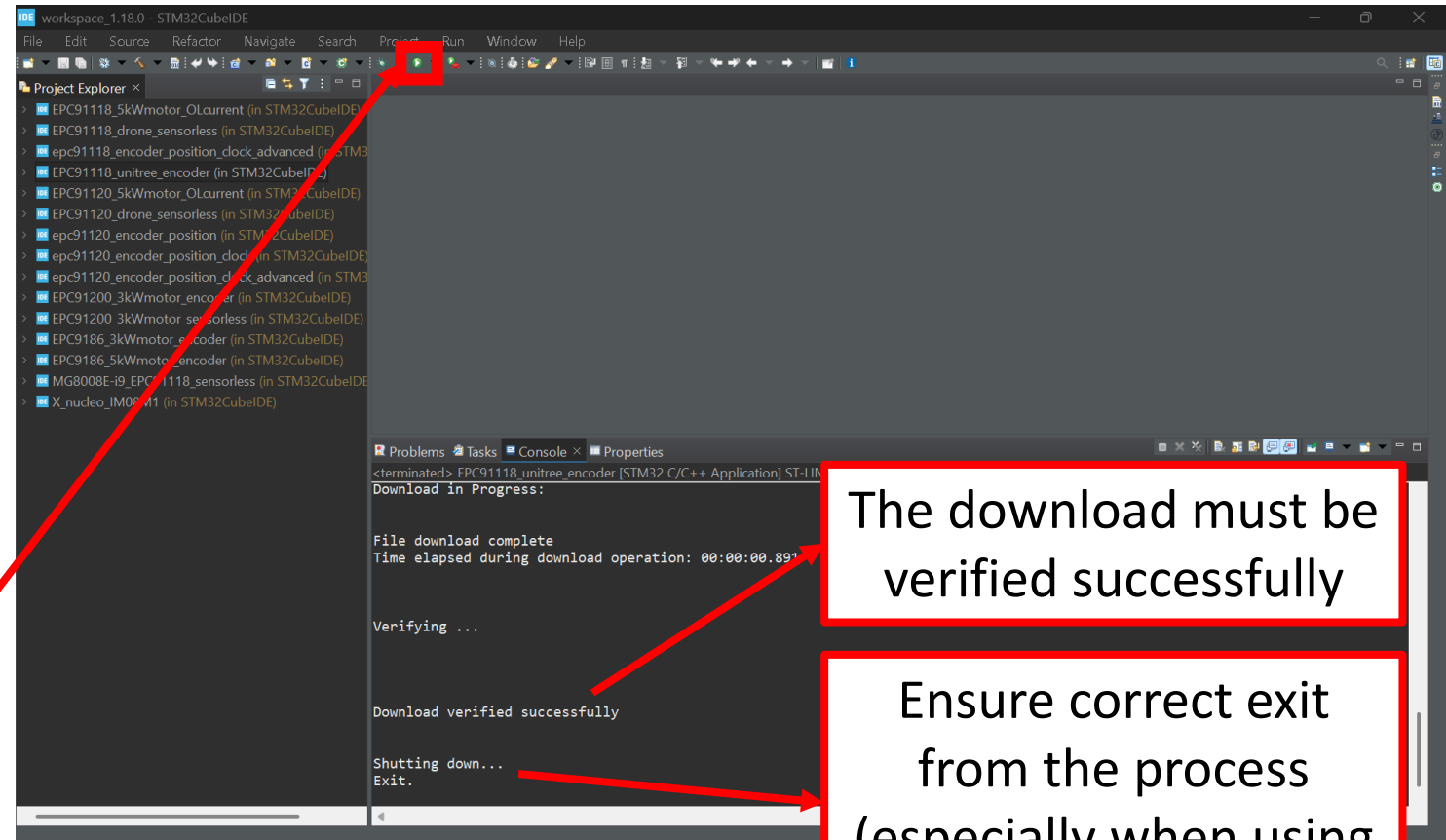
4. Flashing the firmware – hardware requirements

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer
- Build the code
- Connect the inverter to STLINK-V3MINIE and to a power supply



4. Flashing the firmware – cont.

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer
- Build the code
- Connect the inverter to STLINK-V3MINIE and to a power supply
- Flash the firmware to MCU after connecting STLINK-V3MINIE to PC

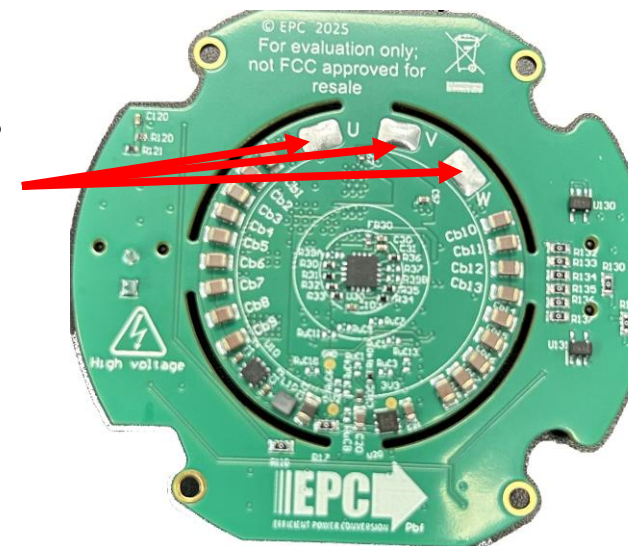


OVERVIEW OF THE PROCESS

1. Procedure targets and requirements
2. Measuring the motor parameters
3. Creating a ST MCW6 project
4. Generating and flashing the firmware
- 5. Operating the motor drive system**

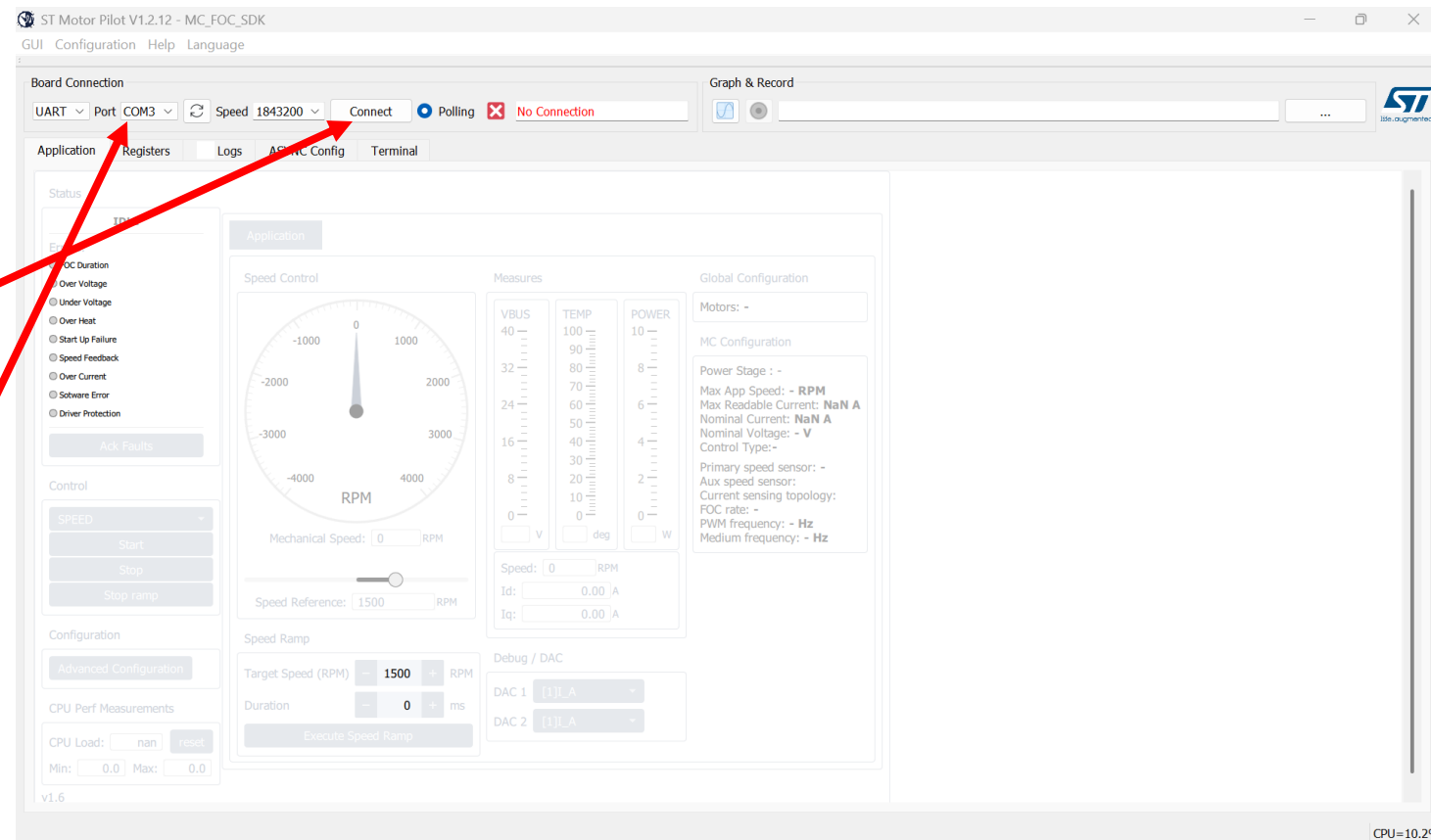
5. Driving the motor

- Solder the motor phases to the respective phase pads on the EPC91118-20-22.



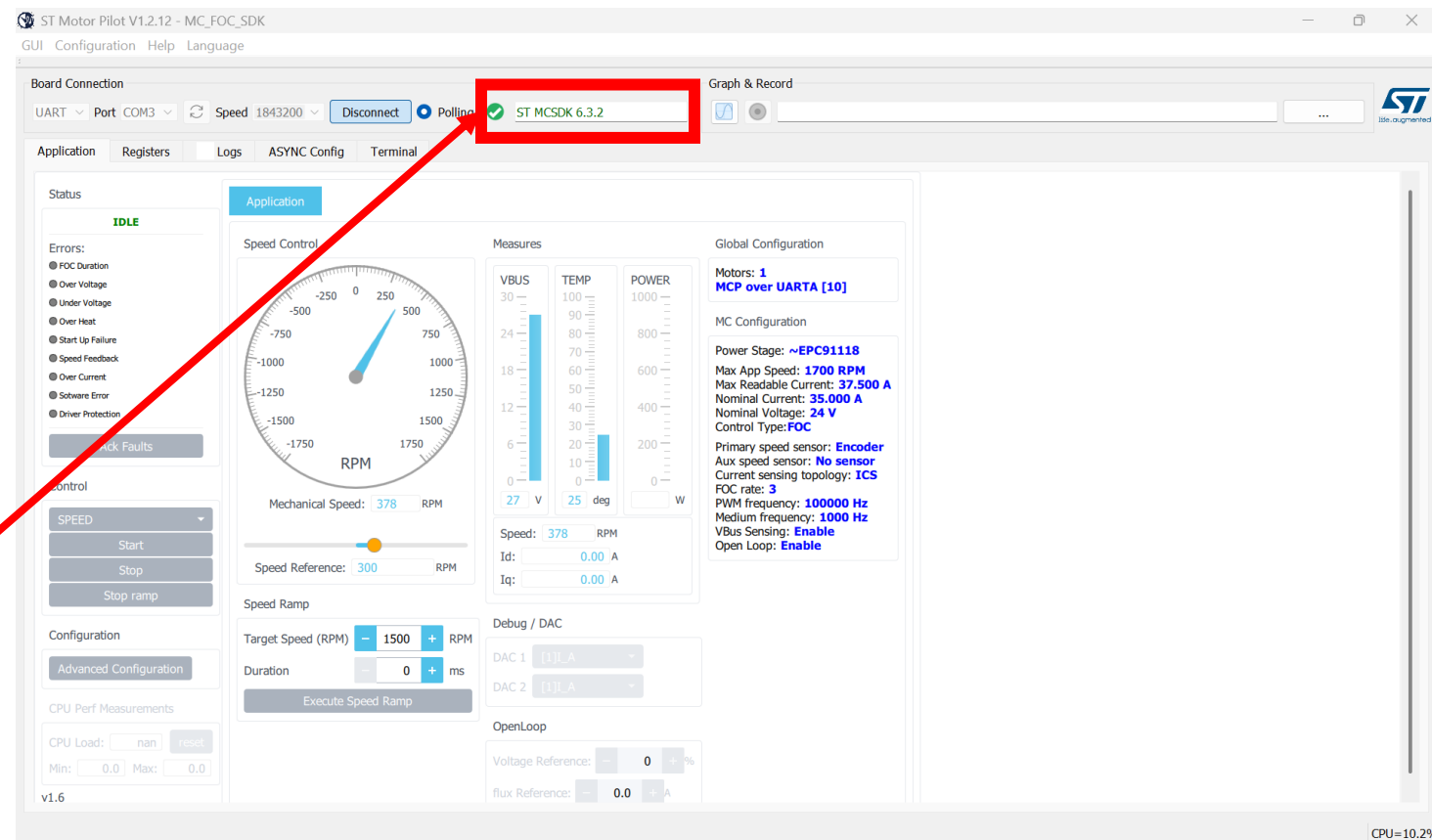
5. Driving the motor

- Solder the motor phases to the respective phase pads on the EPC91118-20-22.
- Open ST motor pilot included in STM32 MC SDK and try to connect the board, checking the correct COM port



5. Driving the motor

- Solder the motor phases to the respective phase pads on the EPC91118-20-22.
- Open ST motor pilot included in STM32 MC SDK and connect the board, checking the correct COM port
- Check for a successful connection; otherwise, check all the connections, and check if the micro is powered



5. Typical connection error

- This connection outcome will appear if the debugging process is still ongoing inside STM32CubeIDE



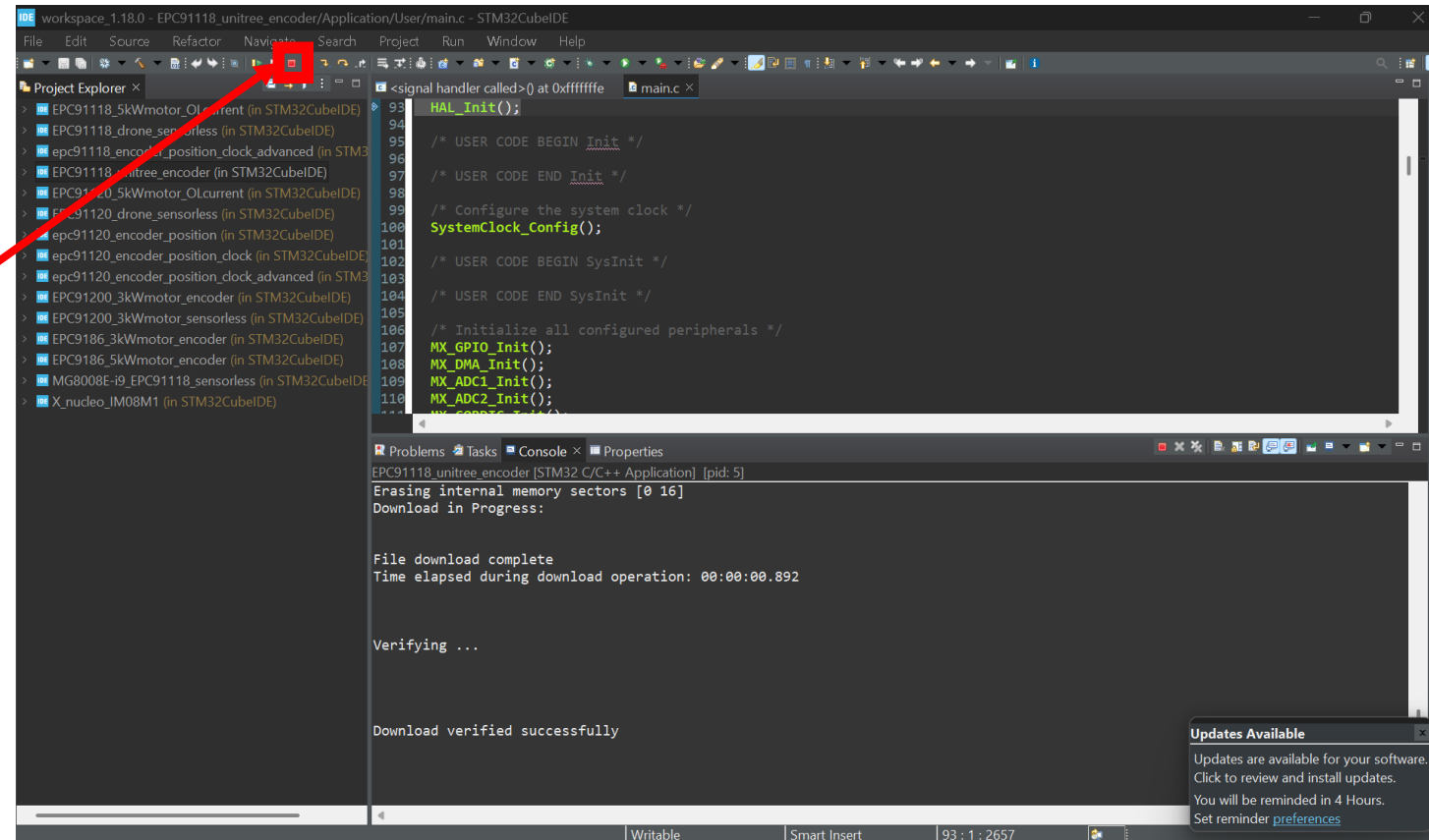
The screenshot shows the ST Motor Control GUI with the following details:

- Status:** IDLE
- Errors:** FOC Duration, Over Voltage, Under Voltage, Over Heat, Start Up Failure, Speed Feedback, Over Current, Software Error, Driver Protection.
- Control:** Start, Stop, Stop ramp buttons.
- Configuration:** Advanced Configuration button.
- CPU Perf Measurements:** CPU Load: nan, reset, Min: 0.0, Max: 0.0.
- Speed Control:** RPM gauge showing 0 RPM, Mechanical Speed: 0 RPM, Speed Reference: 1500 RPM.
- Measures:** VBUS (0-40), TEMP (0-100), POWER (0-10) gauges.
- Global Configuration:** Motors: -, MC Configuration, Power Stage: -, Max App Speed: - RPM, Max Readable Current: NaN A, Nominal Current: NaN A, Nominal Voltage: - V, Control Type: -, Primary speed sensor: -, Aux speed sensor: -, Current sensing topology: -, FOC rate: -, PWM frequency: - Hz, Medium frequency: - Hz.
- Debug / DAC:** DAC 1 [X]_A, DAC 2 [X]_A.

A red box highlights the error message "and try to configure the board" with a warning icon. A red arrow points from the text in the slide to this error message.

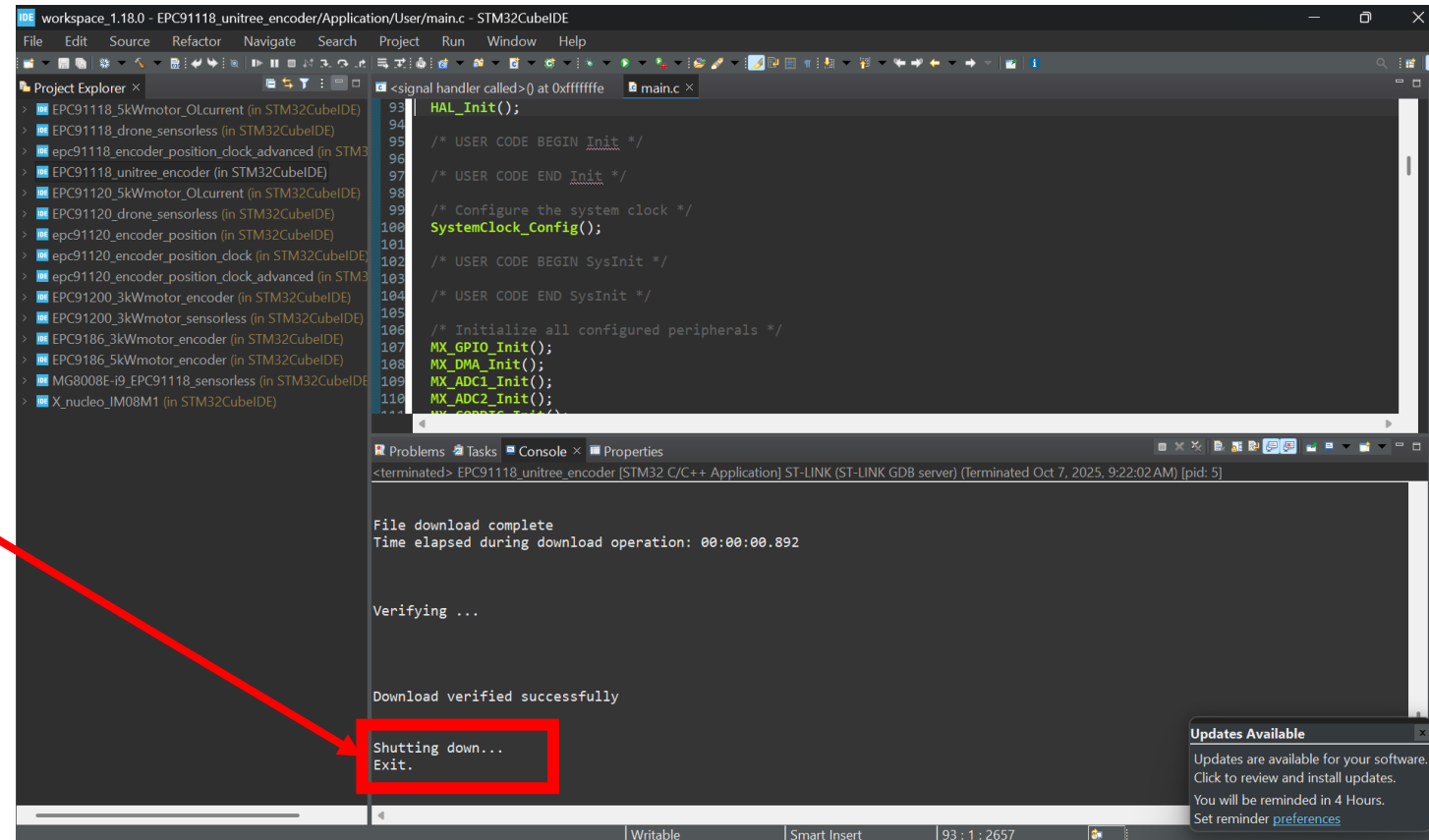
5. Typical connection error

- This connection outcome will appear if the debugging process is still ongoing inside STM32CubeIDE
- Click on 'Terminate' button to interrupt the debug connection



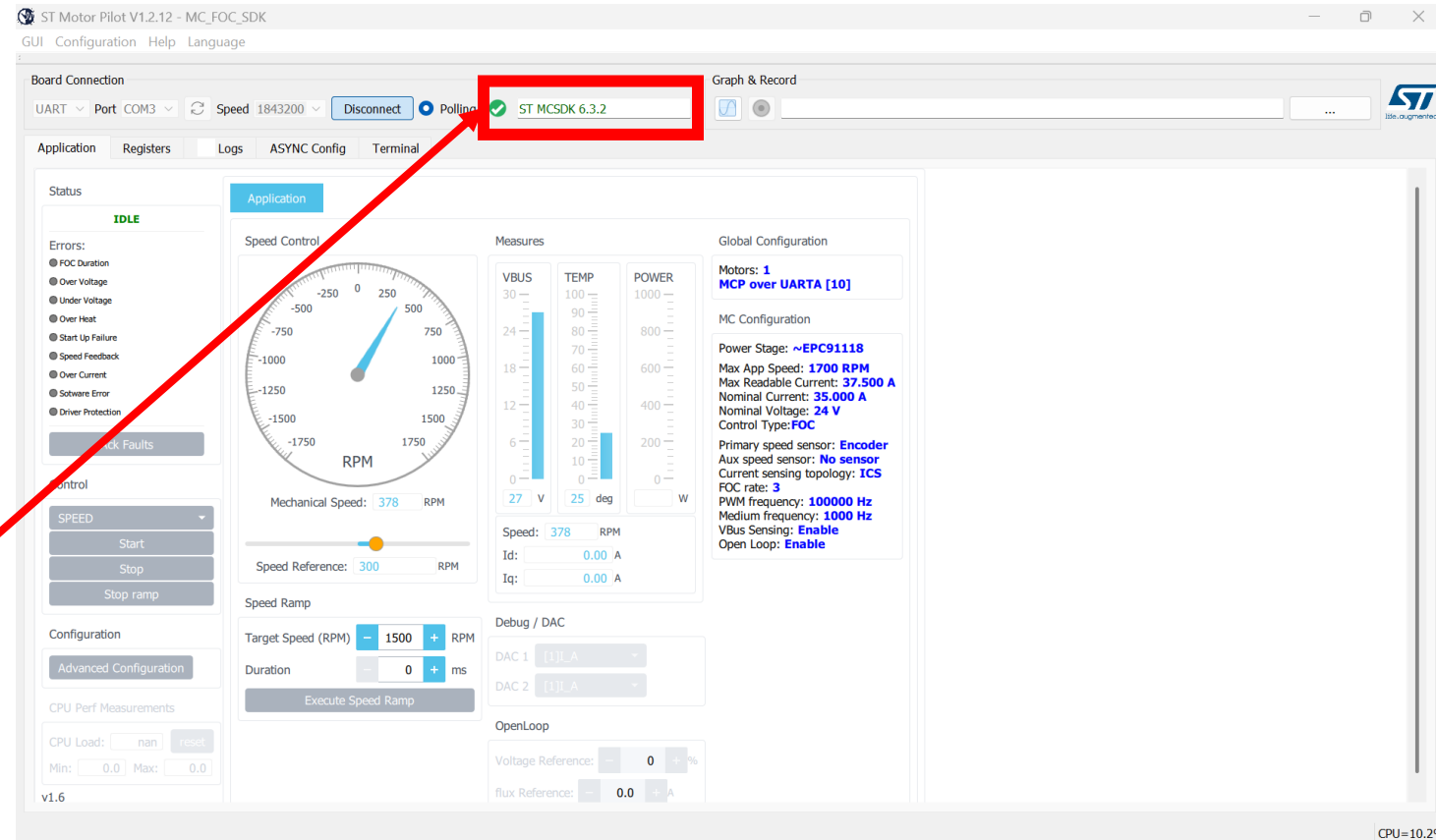
5. Typical connection error

- This connection outcome will appear if the debugging process is still ongoing inside STM32CubeIDE
- Click on 'Terminate' button to interrupt the debug connection
- Ensure correct exit from the process



5. Typical connection error

- This connection outcome will appear if the debugging process is still ongoing inside STM32CubeIDE
- Click on 'Terminate' button to interrupt the debug connection
- Ensure correct exit from the process
- Retry the connection with the MCU and verify the correct execution



5. Driving the motor

For a detailed guide on how to use the motor pilot tool, please refer to this link:
[STM32 MC Motor Pilot - Start-up guide - stm32mcu](https://www.st.com/en/motor-control/stm32-mc-motor-pilot-start-up-guide.html)

STM32 MC Motor Pilot - Start-up guide

1. Overview of the STM32 MC Motor Pilot tool

The STM32 MC Motor Pilot is a monitoring tool for STM32 motor-control applications that:

- connects to MC applications built with the UI module through the serial port
- uses the new Monitor and Control Protocol (MCPV2)
- allows for controlling, monitoring, and tuning MC applications
- replaces the monitor part of STM32 MC workbench 4.X

The added value features of the STM32 MC Motor Pilot are indicated below.