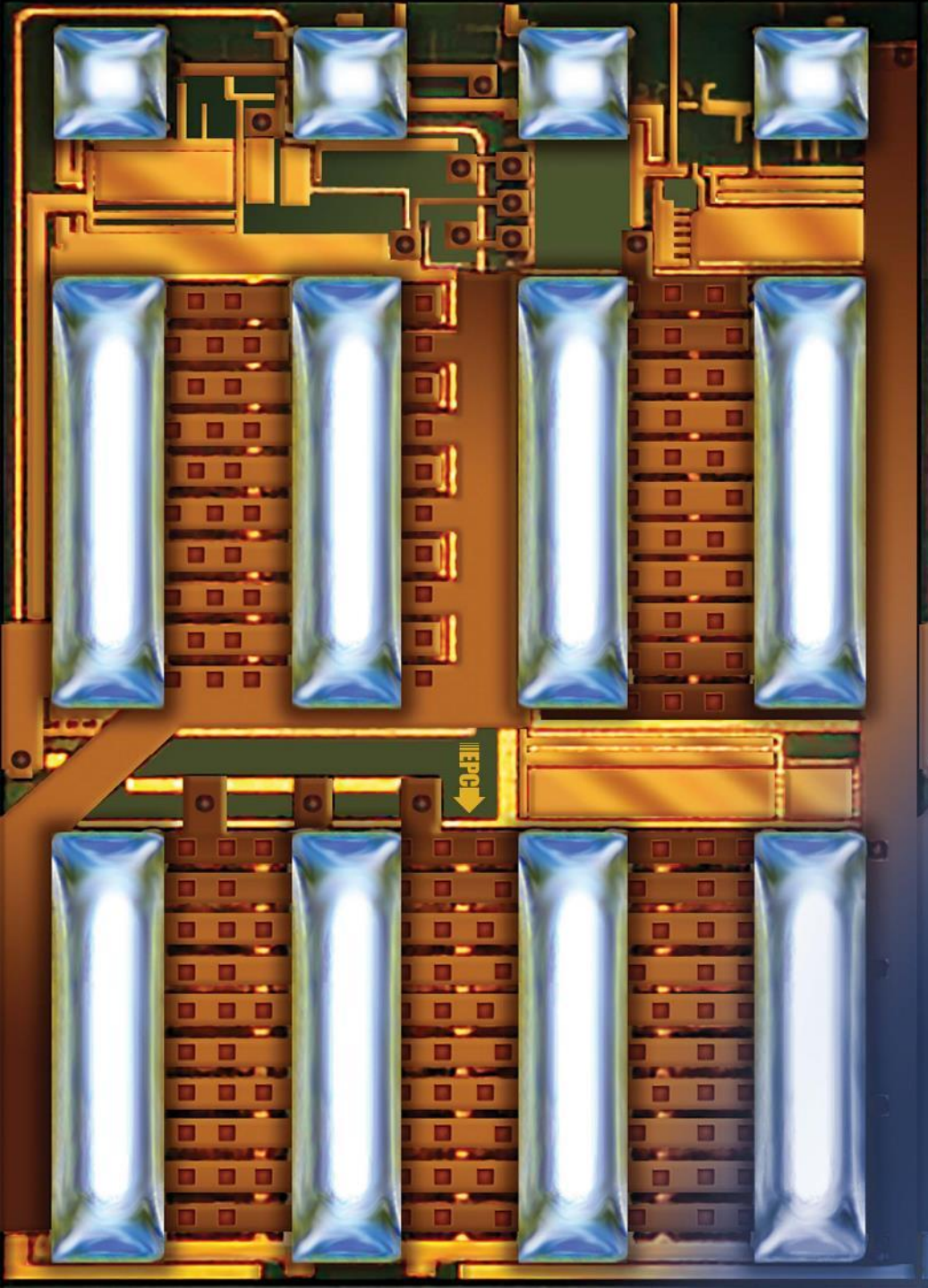


EPC91132
Motor Commissioning
Quick Start Procedure

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 EPC91132 board, either using a sensorless algorithm or exploring the magnetic encoder embedded in the inverter for 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, we recommend using the latest 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 tools to correctly generate, flash, and run the motor control firmware:

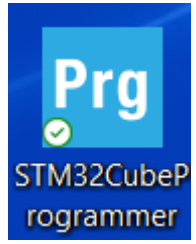


- STM32CubeMX (initialization code generator): https://www.st.com/en/development-tools/stm32cubemx.html



- STM32CubeIDE (Integrated Development Environment for STM32): https://www.st.com/en/development-tools/stm32cubeide.html

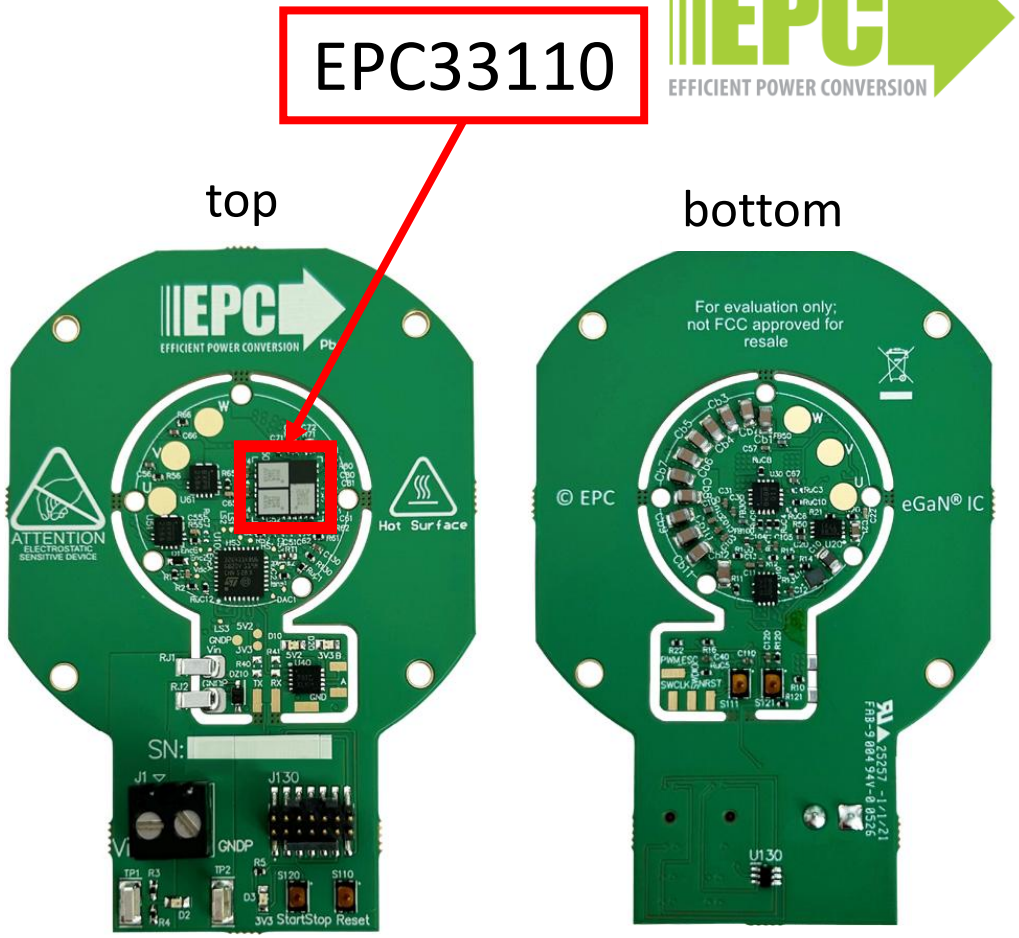
- OPTIONAL STMCubeProgrammer (for quick programming using .elf files): STM32CubeProg | Software - STMicroelectronics



1. EPC91132 inverter board

The EPC91132 evaluation board is a 3-phase BLDC motor drive inverter designed with an optimized form factor for small humanoid robot joints and 23 mm drone motors. The whole inverter is housed within the 23 mm inner circle, while the external frame can be adjusted to fit different motors.

This guide refers to the full form factor. In this configuration, the board features an external PCB ring that is meant, for mechanical reasons, to install the board into a specific humanoid joint motor and to place connectors J130 and J1. This solution is useful for debugging the firmware and for lab tests.

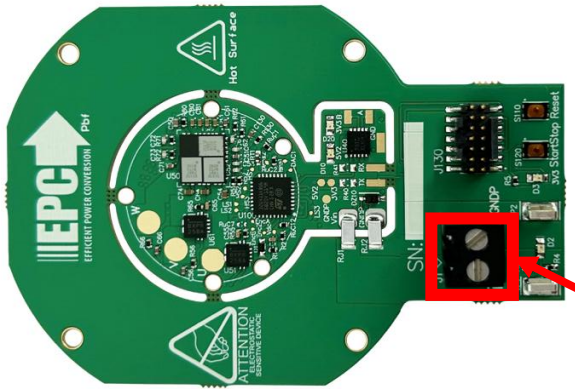


EPC91132 board

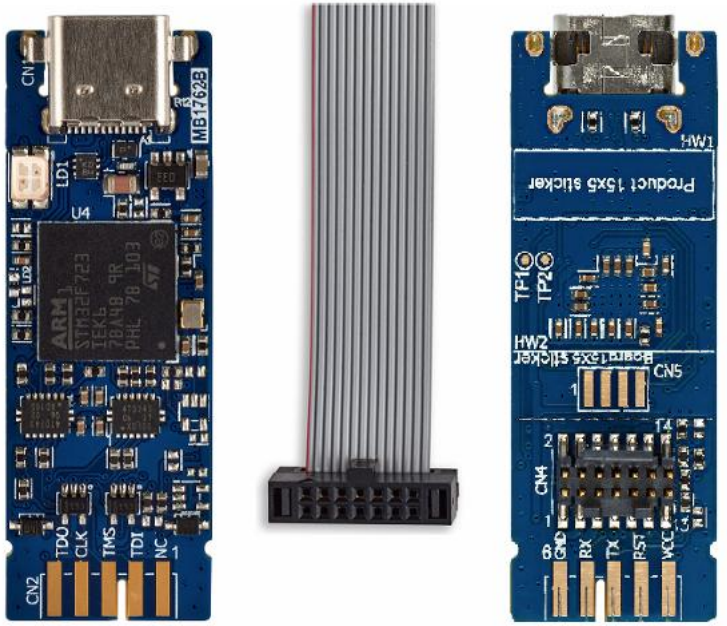
1. STLINK-V3MINIE

To communicate properly with the microcontroller on the EPC91132 board, use the STLINK-V3MINIE debugger/programmer with the tiny probe for STM32 microcontrollers, as shown in the figure.

<u>After connecting the JTAG/SWD interface between the two boards, power on the inverter via connector **J1** to supply the microcontroller with 3.3V (check the yellow LED on the EPC91132).</u>



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 measurements. 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:

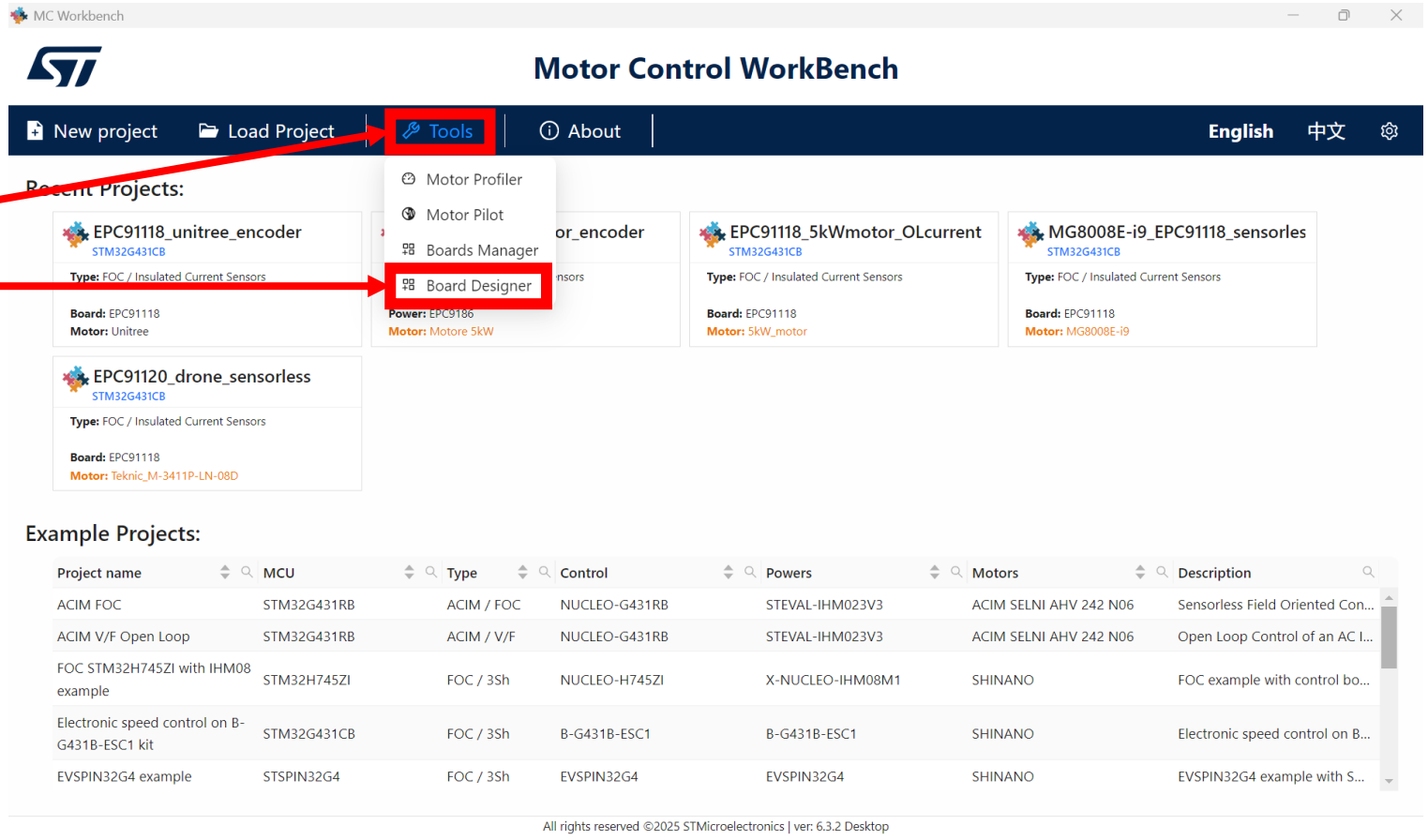
- Phase resistance (Ω)
- Phase inductance (mH)
- Pole pairs
- B-Emf constant (Vrms/krpm)

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

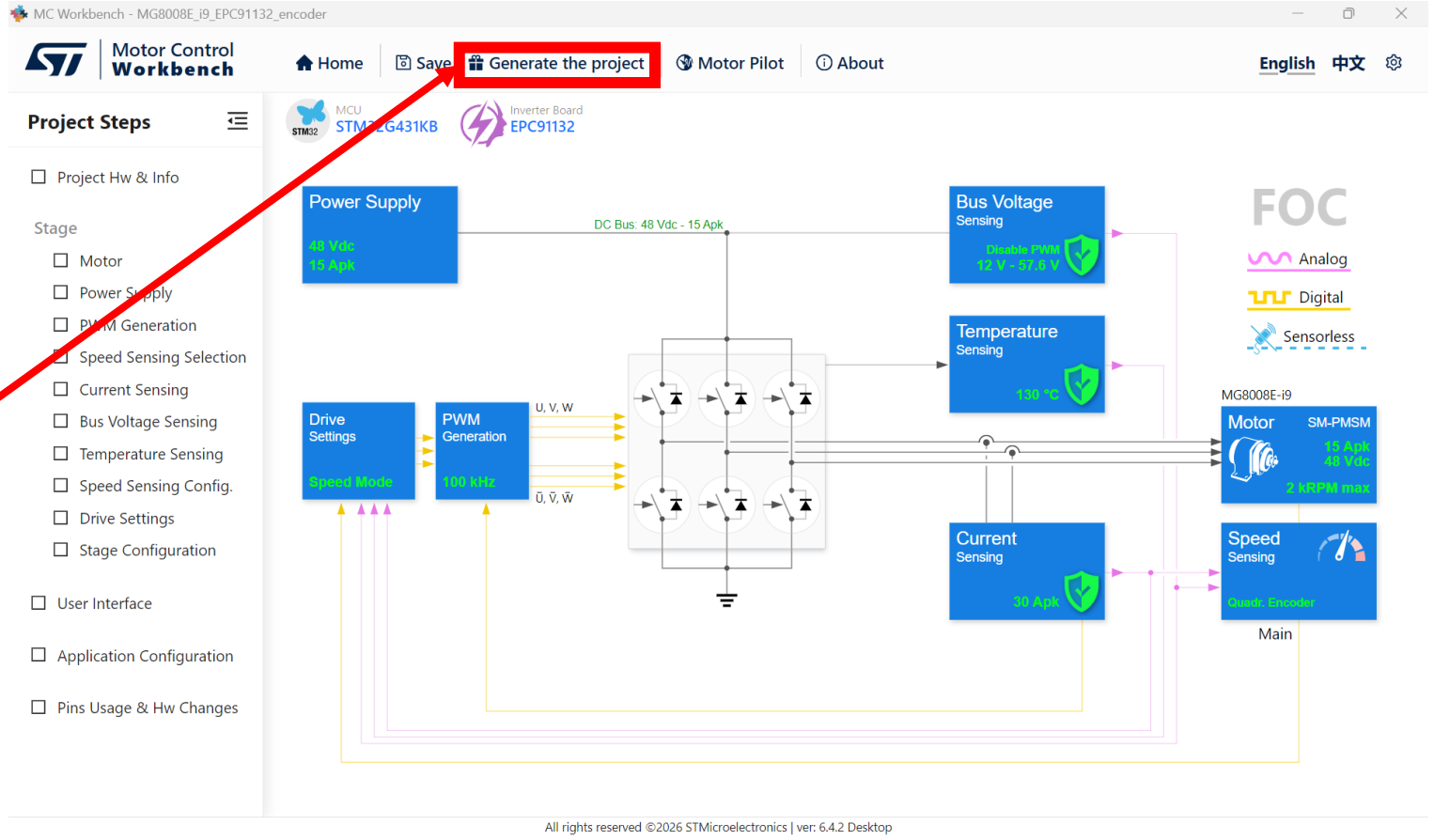
3. Opening the tools from MCW6

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



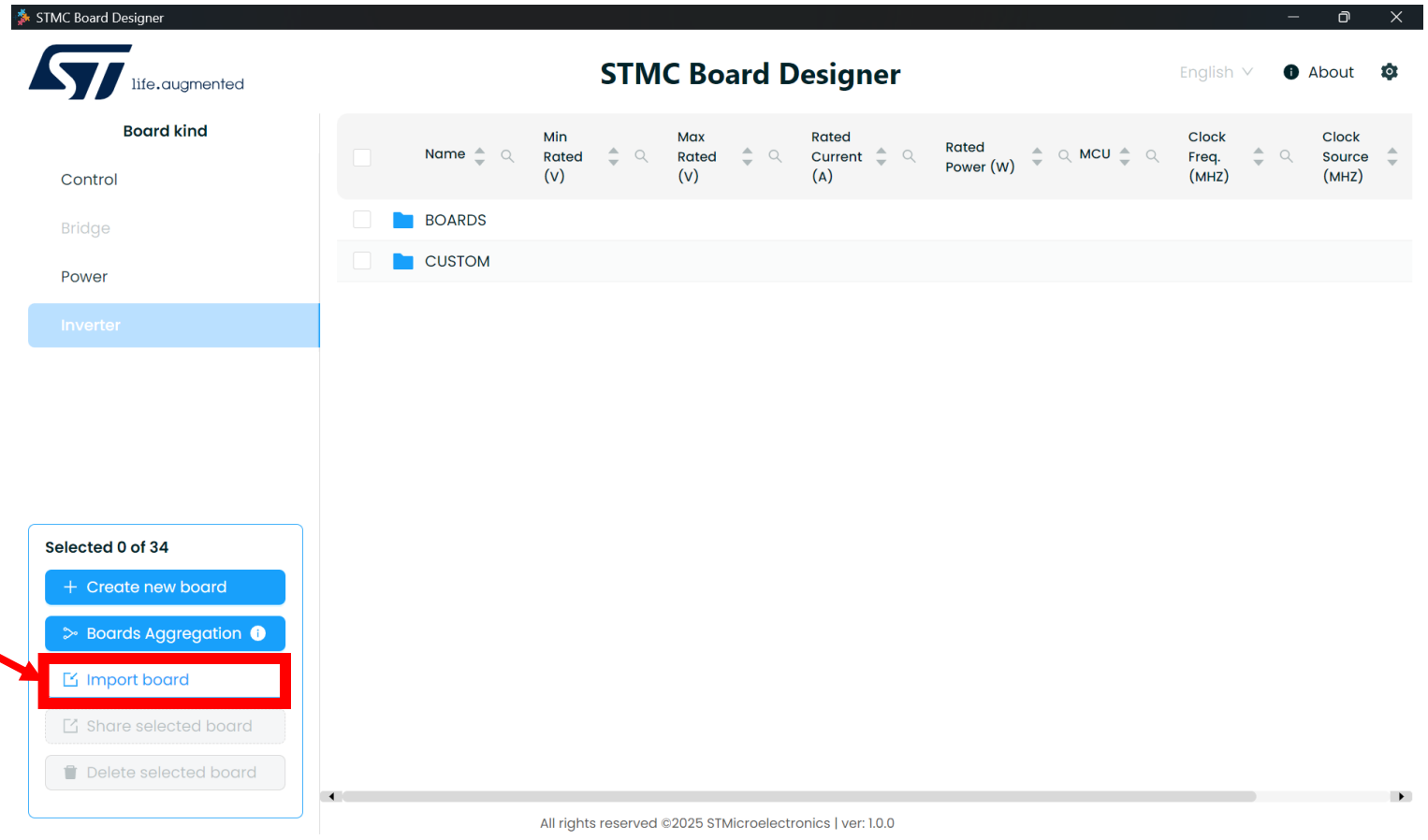
3. Opening the tools from MCW6

- 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



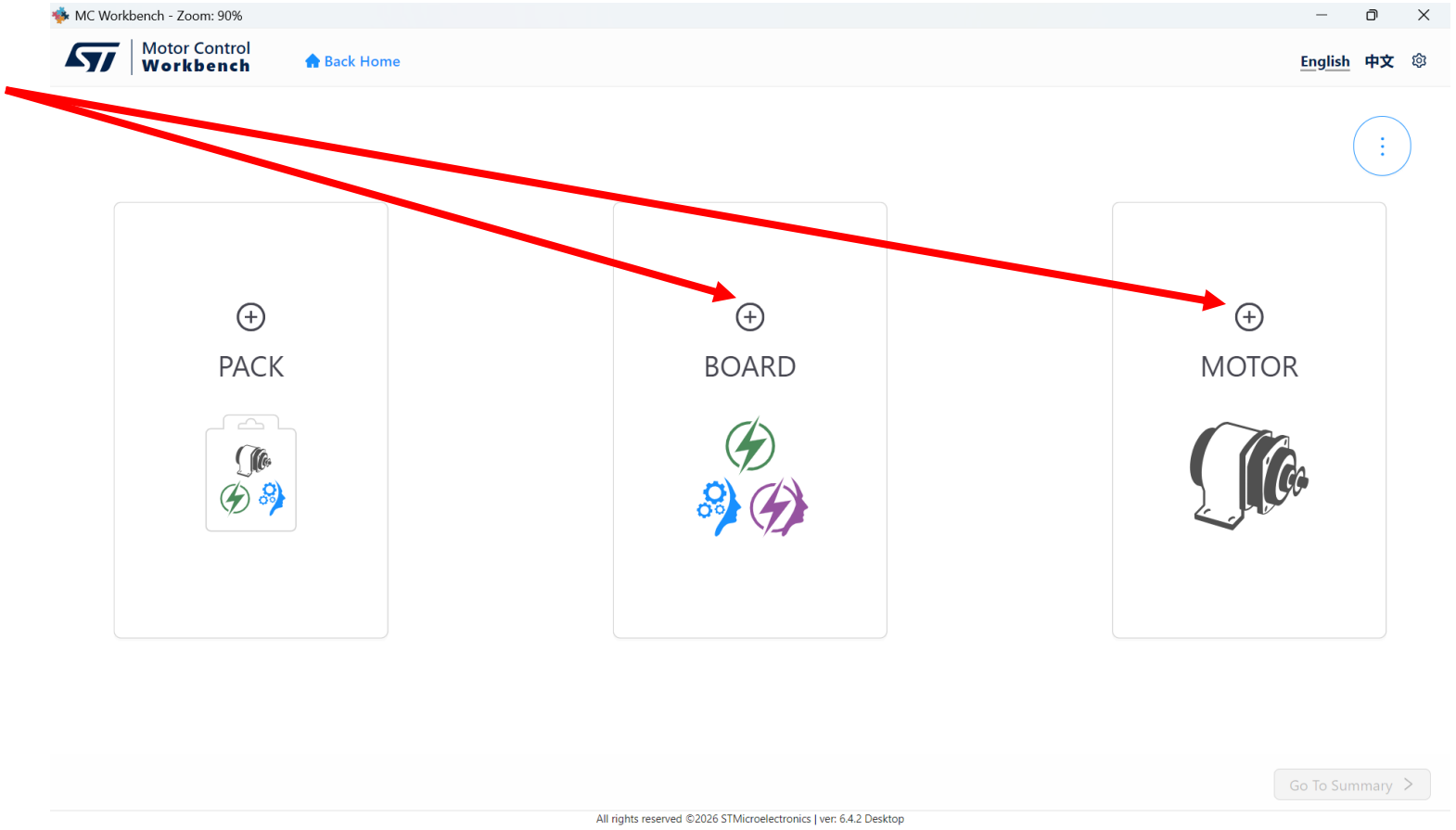
3. ST MCW6 project

- Download the .json file (EPC91132.json) of the inverter to be used in the project from this repository: GitHub - epc-co/MotorDrive-RefDesign-Firmware: EPC's Motor Drive Reference Designs Firmware repository
- Import the .json file of the board into the STMC board designer, which is a tool inside MCW6.



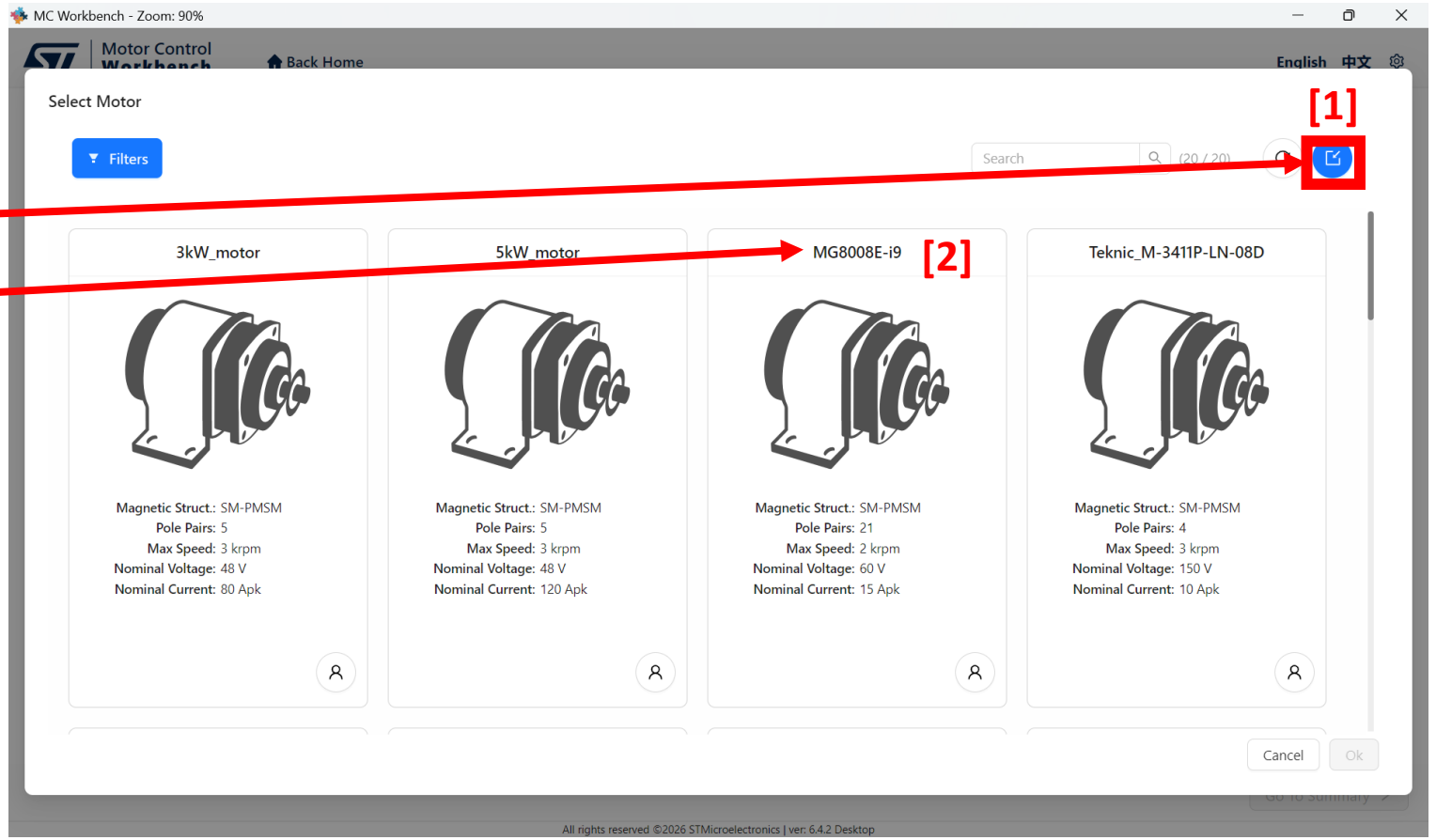
3. ST MCW6 project

- Make a new project, choosing the board and the motor



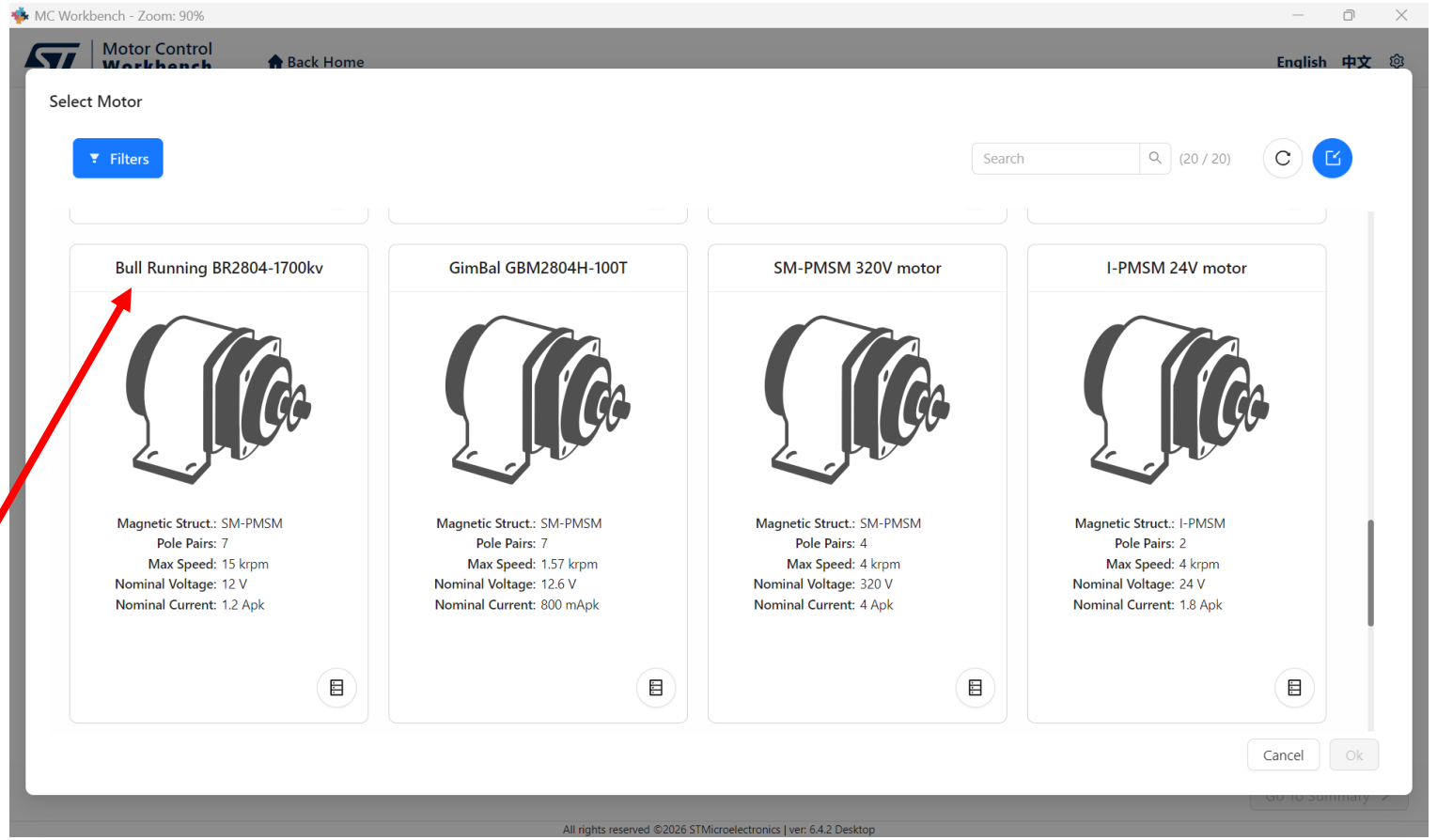
3. ST MCW6 project

- 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



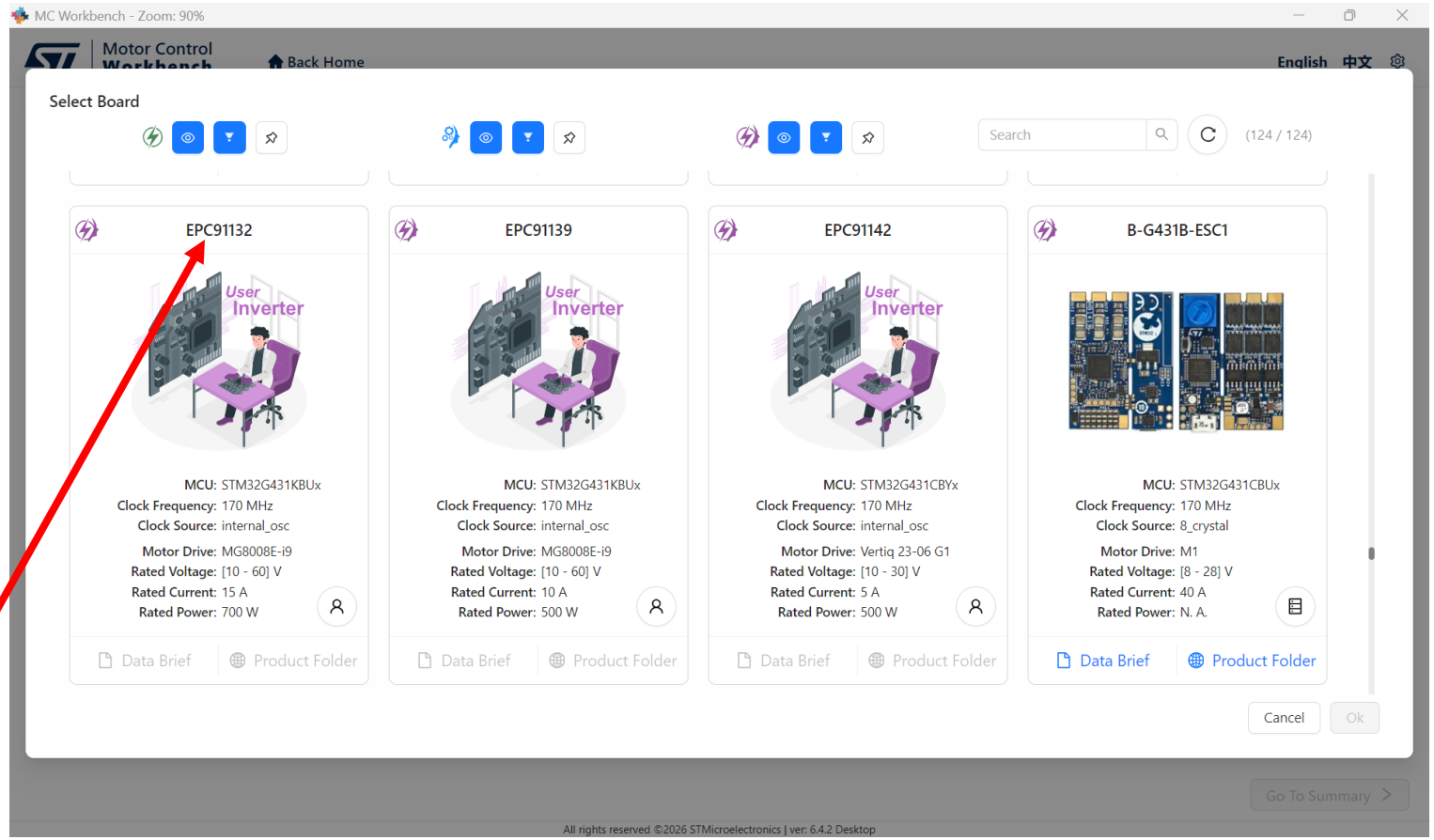
3. ST MCW6 project

- 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



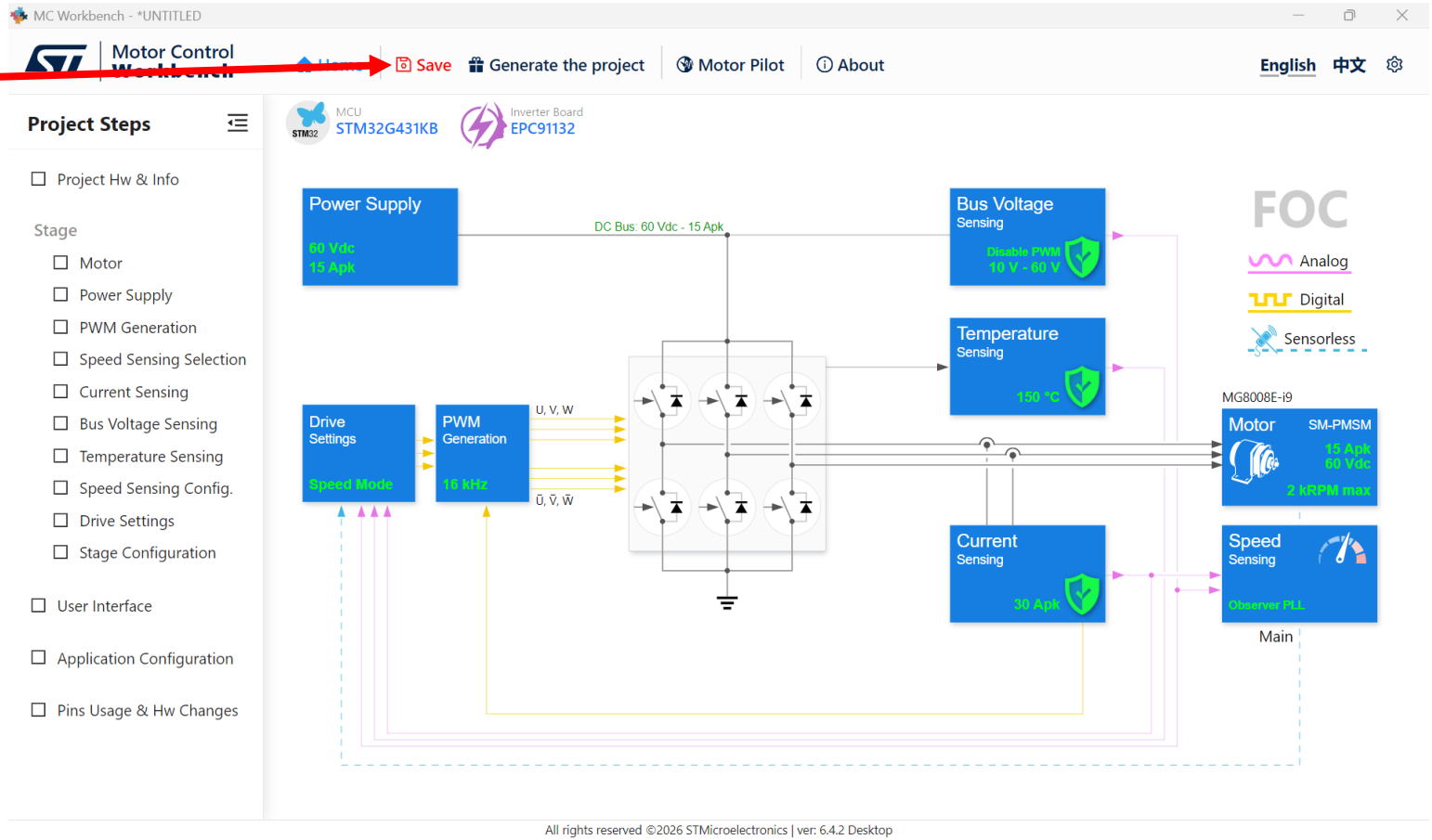
3. ST MCW6 project

- 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



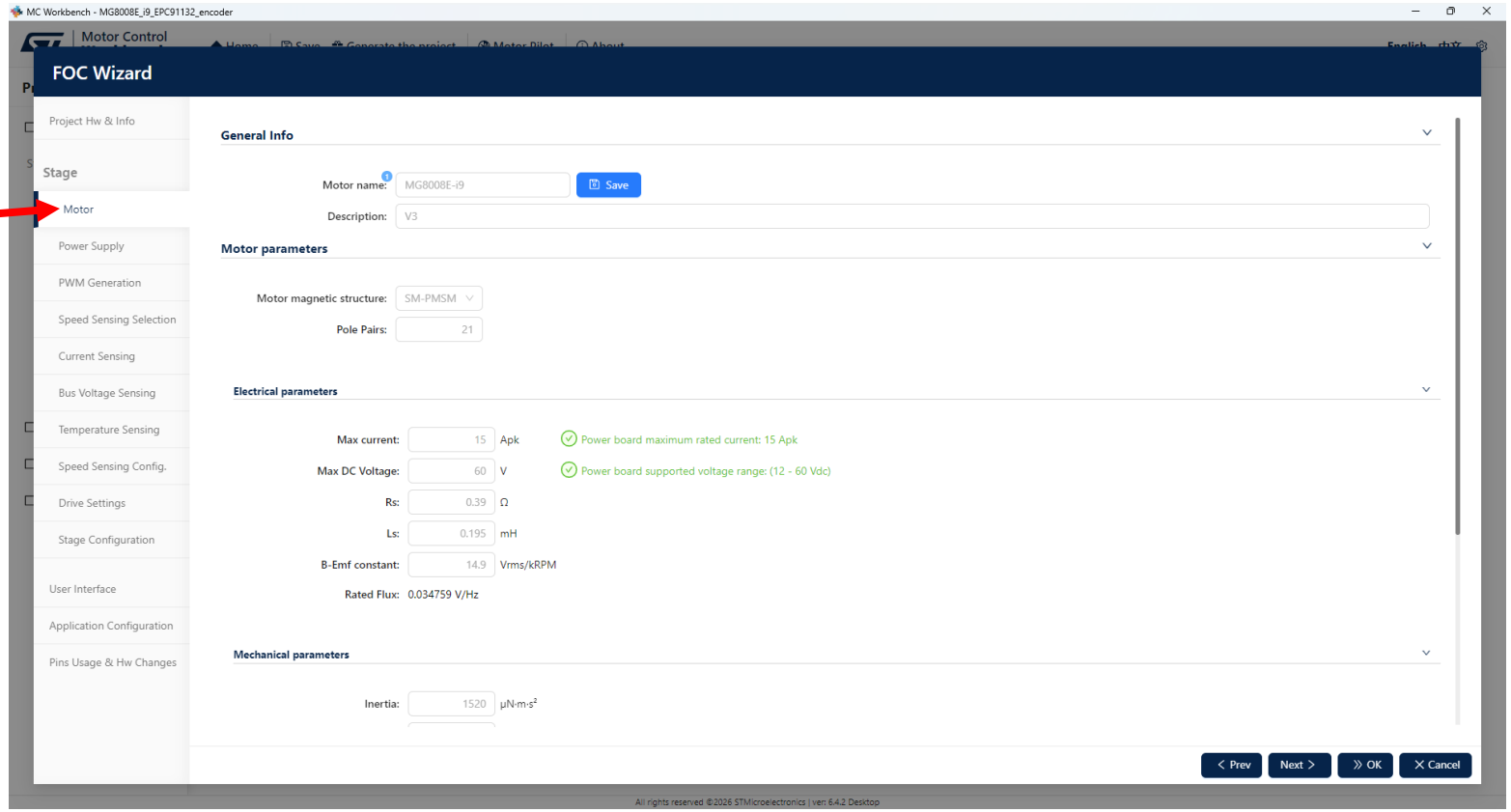
3. ST MCW6 project

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



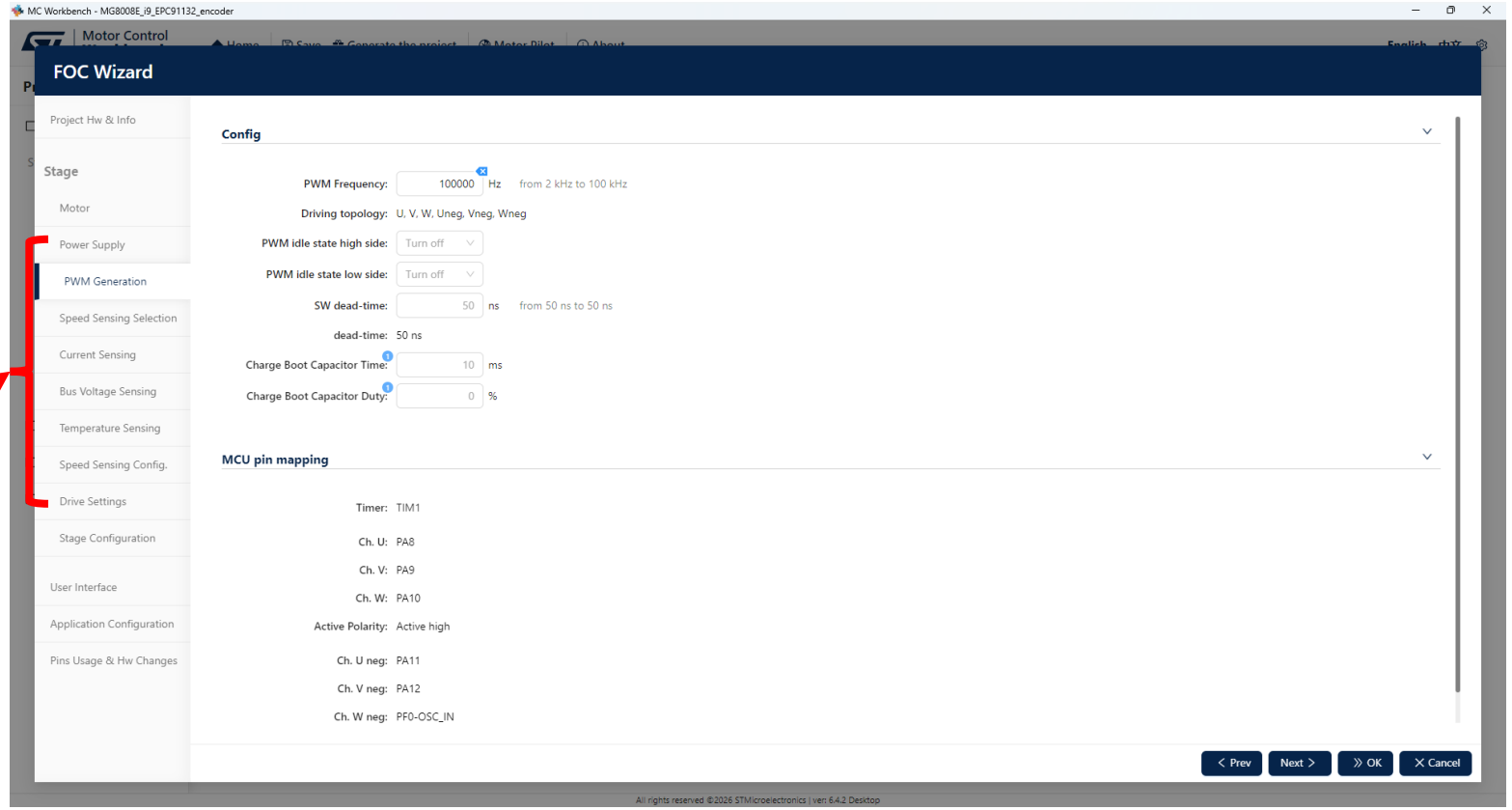
3. ST MCW6 project

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



3. ST MCW6 project

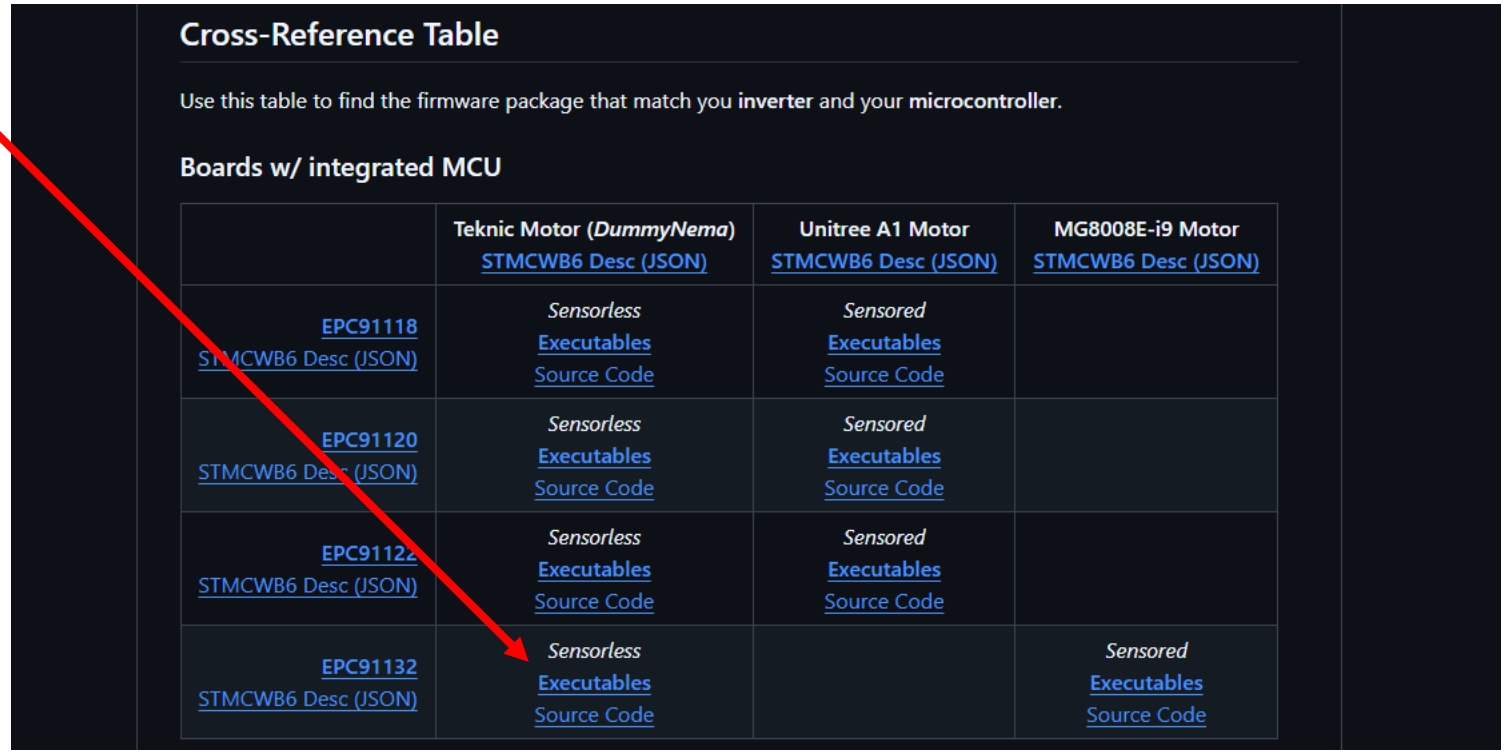
- Save the project file to be able to generate the code
- In the ‘Motor’ window, you can change the motor parameters and possibly save them into a new .json file
- For the configuration of other sections, please refer to the previously mentioned repository, which contains two projects configured for EPC91132 and two different motors



3. ST MCW6 project

- For 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



Cross-Reference Table

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

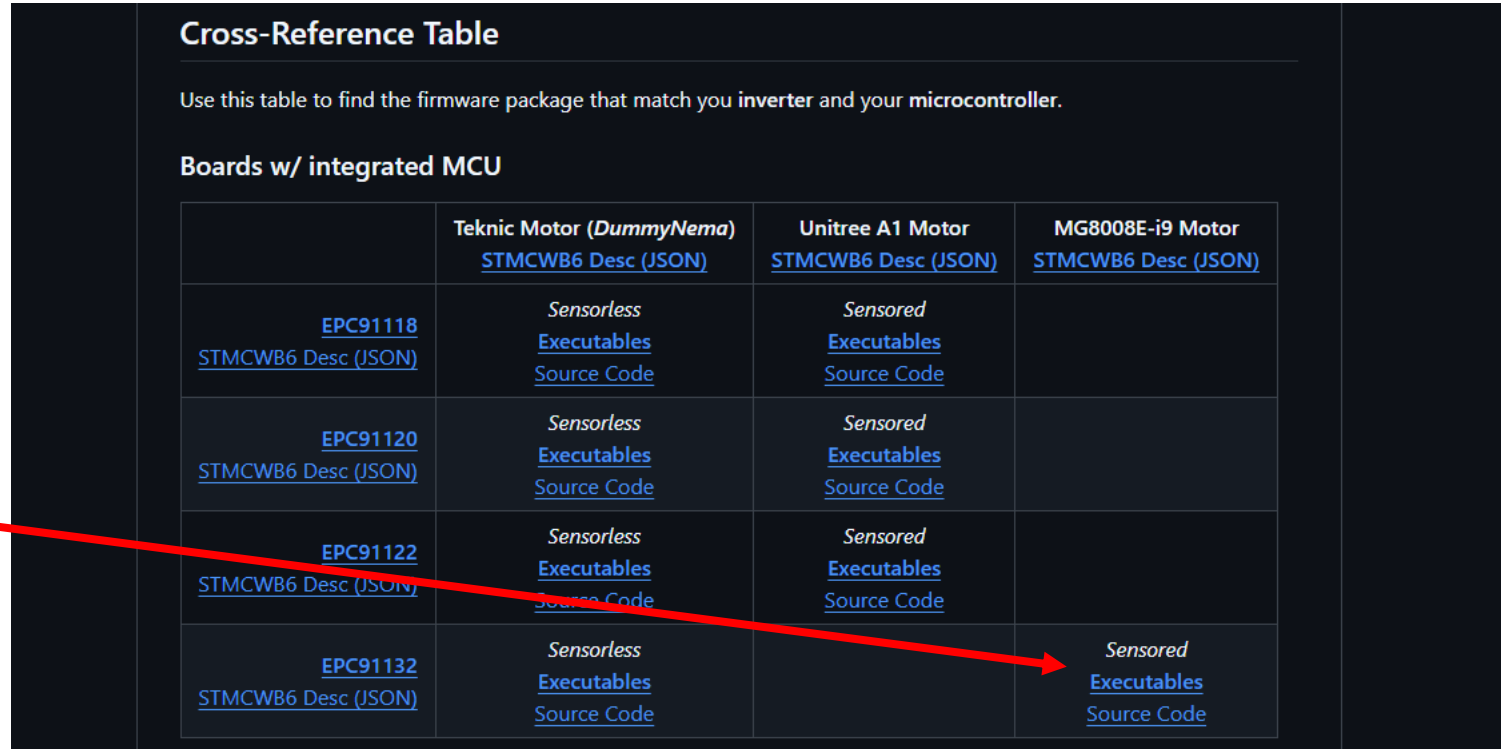
Boards w/ integrated MCU

	Teknic Motor (*DummyNema*) STMCWB6 Desc (JSON)	Unitree A1 Motor STMCWB6 Desc (JSON)	MG8008E-i9 Motor STMCWB6 Desc (JSON)
EPC91118 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91120 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91122 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91132 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code		*Sensored* Executables Source Code

3. ST MCW6 project

- For 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 with the encoder, look at this reference project

GitHub - epc-co/MotorDrive-RefDesign-Firmware: EPC's Motor Drive Reference Designs Firmware repository



Cross-Reference Table

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

Boards w/ integrated MCU

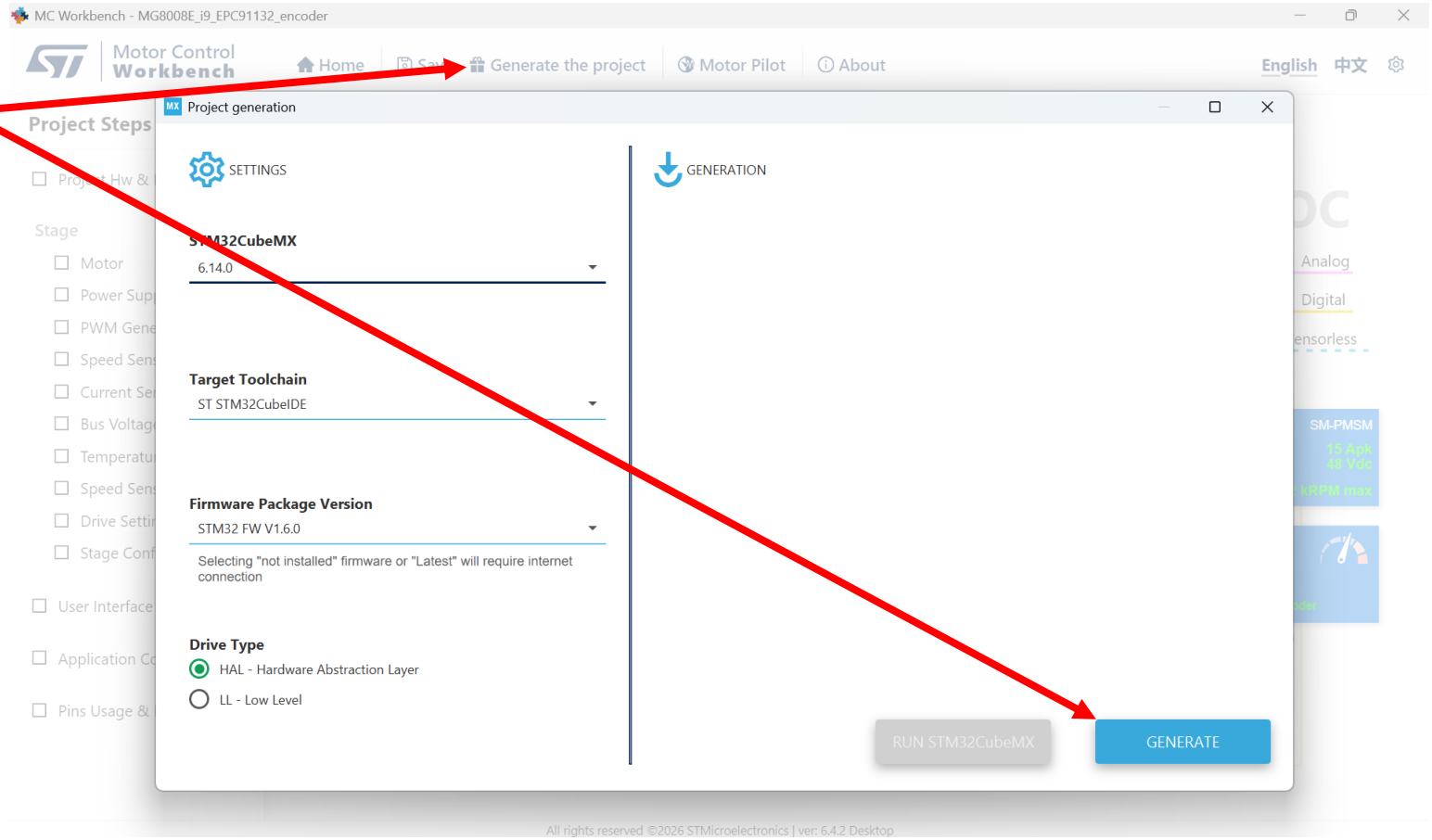
	Teknic Motor (*DummyNema*) STMCWB6 Desc (JSON)	Unitree A1 Motor STMCWB6 Desc (JSON)	MG8008E-i9 Motor STMCWB6 Desc (JSON)
EPC91118 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91120 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91122 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code	*Sensored* Executables Source Code	
EPC91132 STMCWB6 Desc (JSON)	*Sensorless* Executables Source Code		*Sensored* Executables Source Code

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

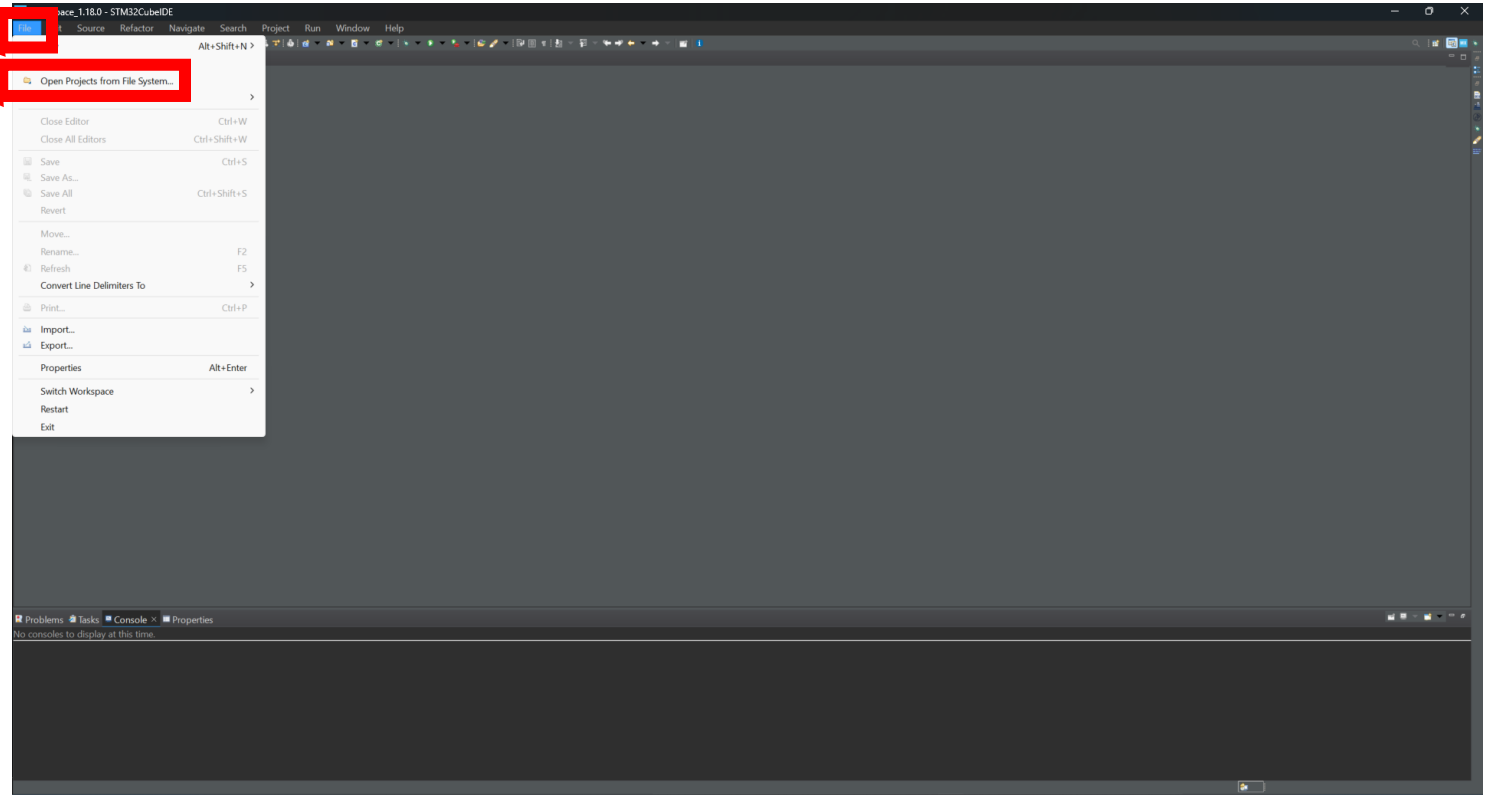
4. Generating the firmware

- Generate the code through STM32CubeMX interface inside MCW project



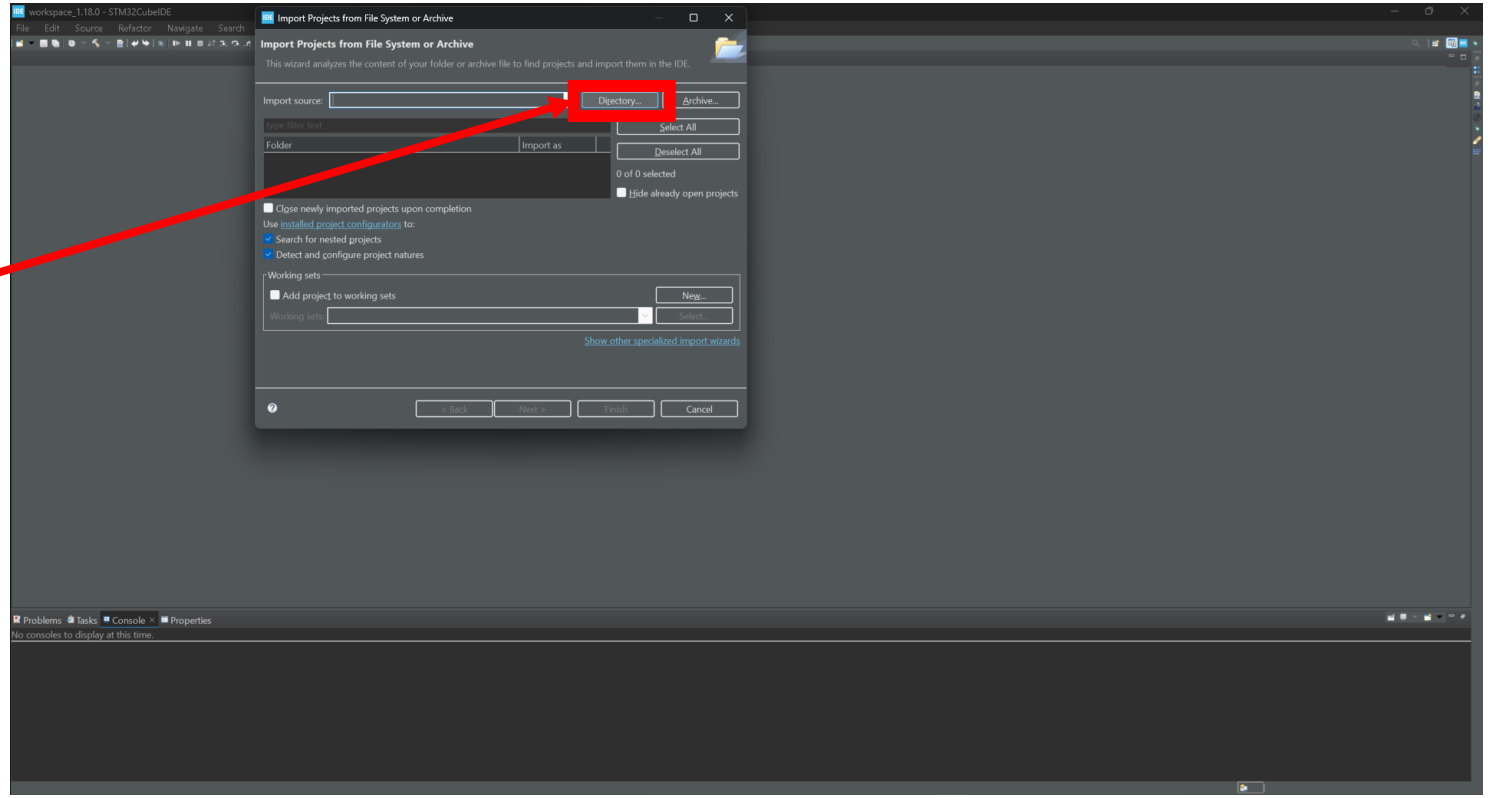
4. Building the firmware

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



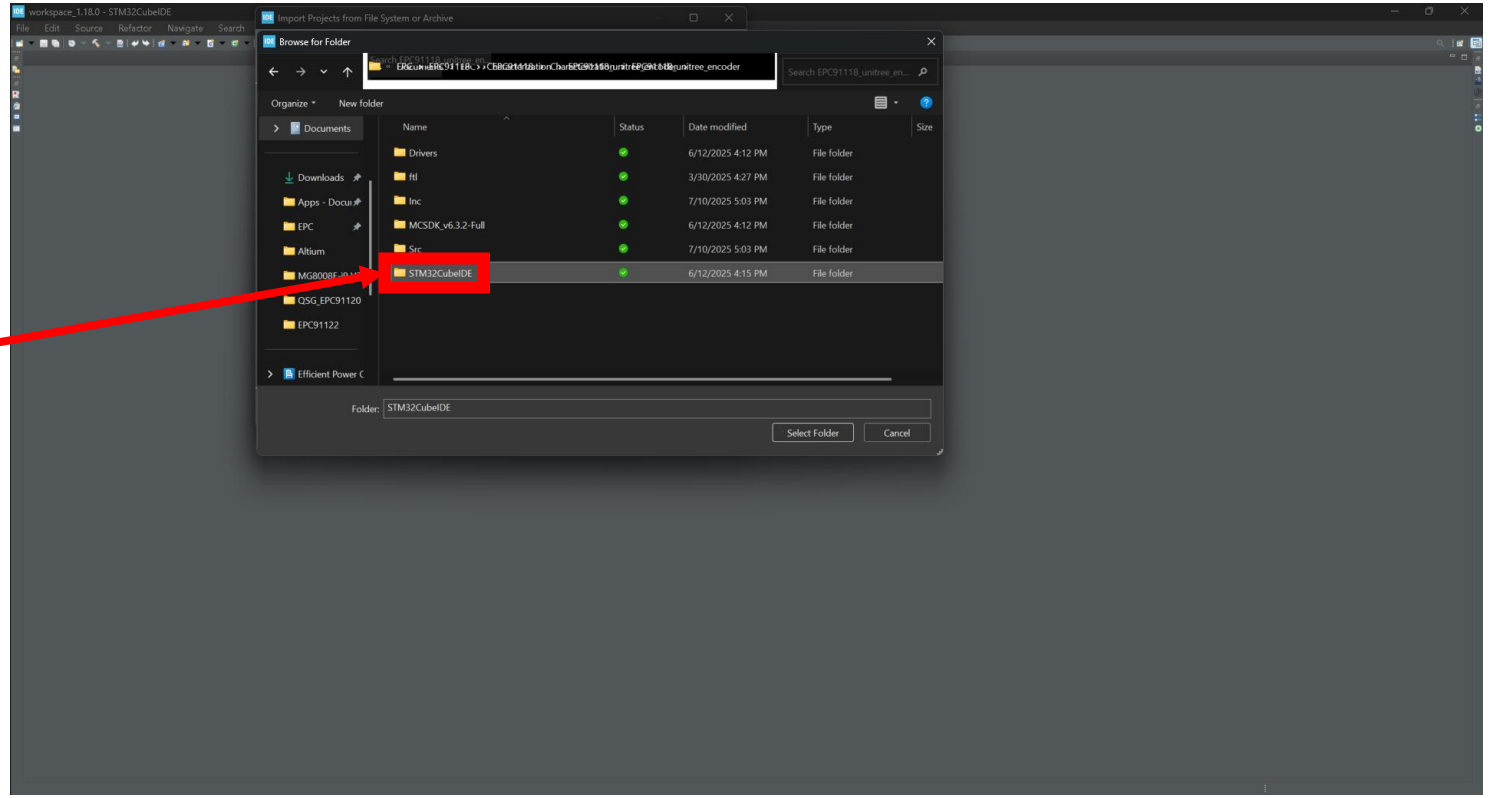
4. Building the firmware

- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE:
  click ‘Directory…’



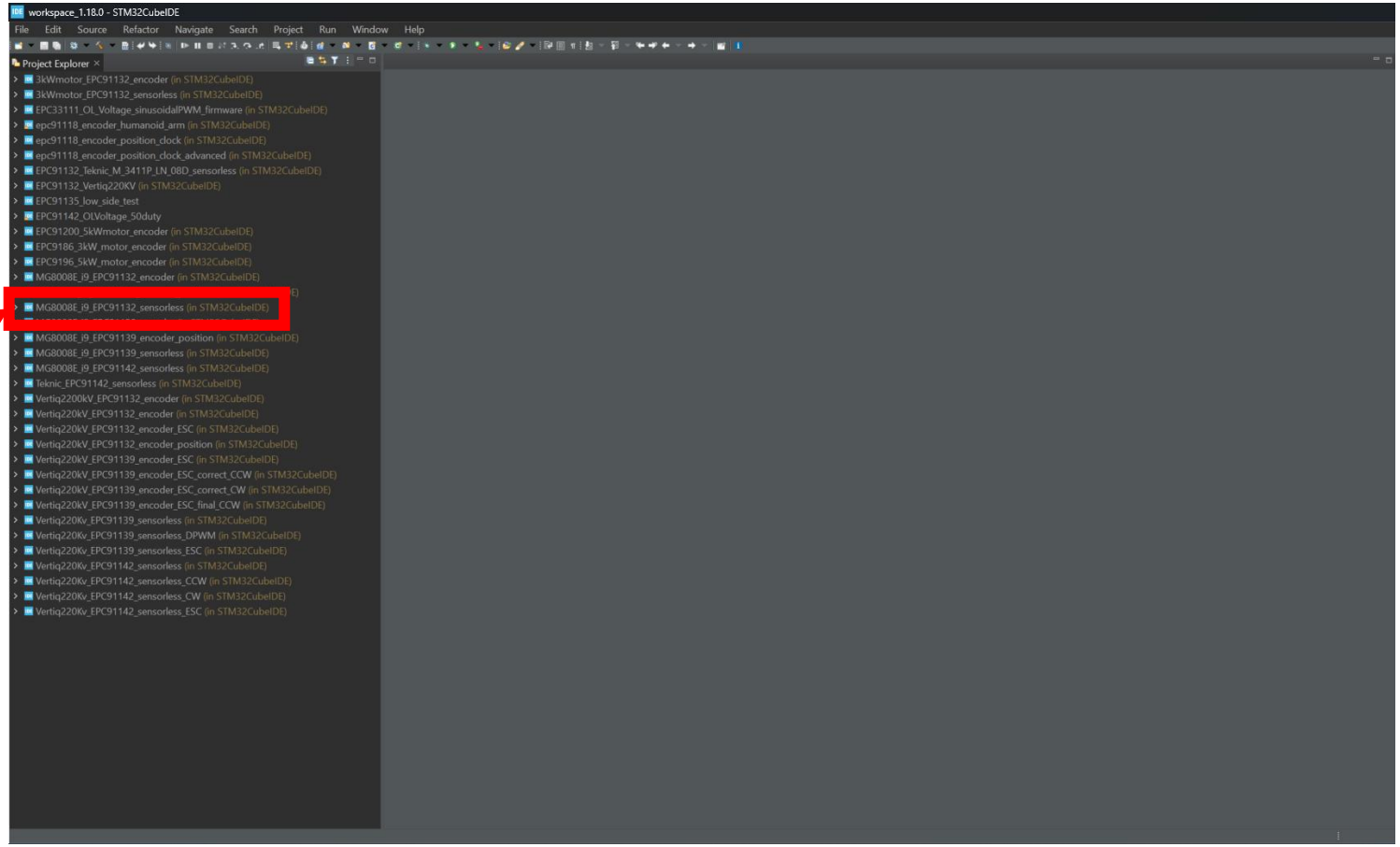
4. Building the firmware

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



4. Building the firmware

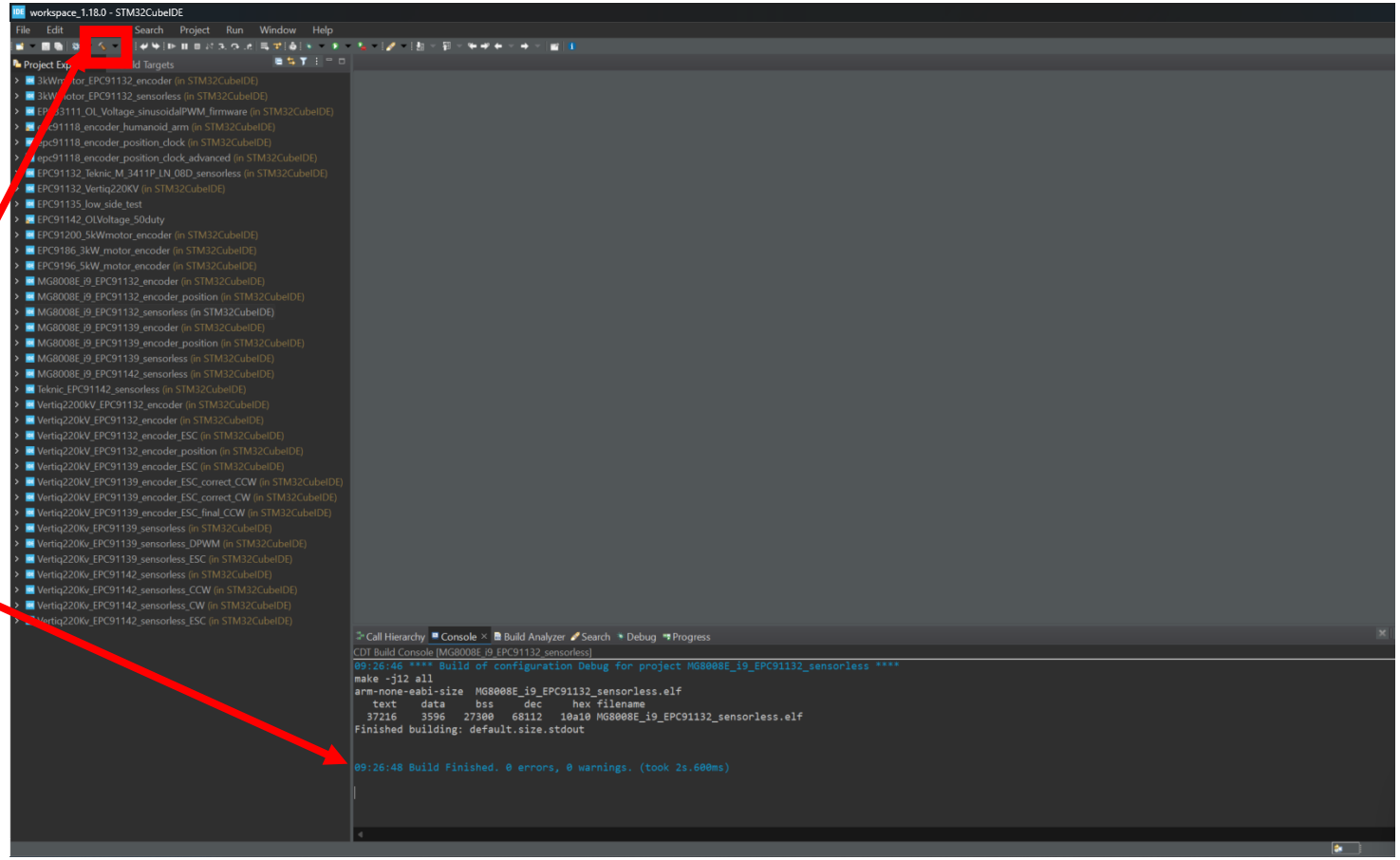
- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer



4. Building the firmware

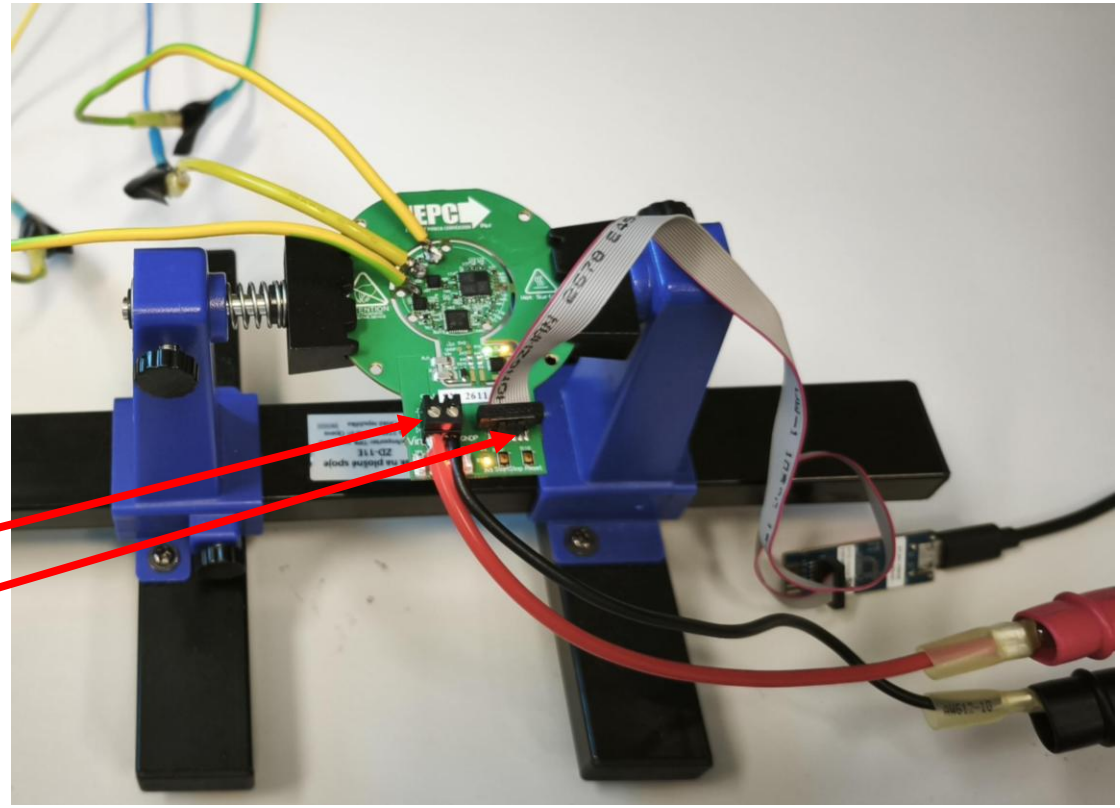
- Generate the code through STM32CubeMX interface inside MCW project
- Open the project in STM32CubeIDE
- Select the project from the explorer
- Build the code and verify the result of the process





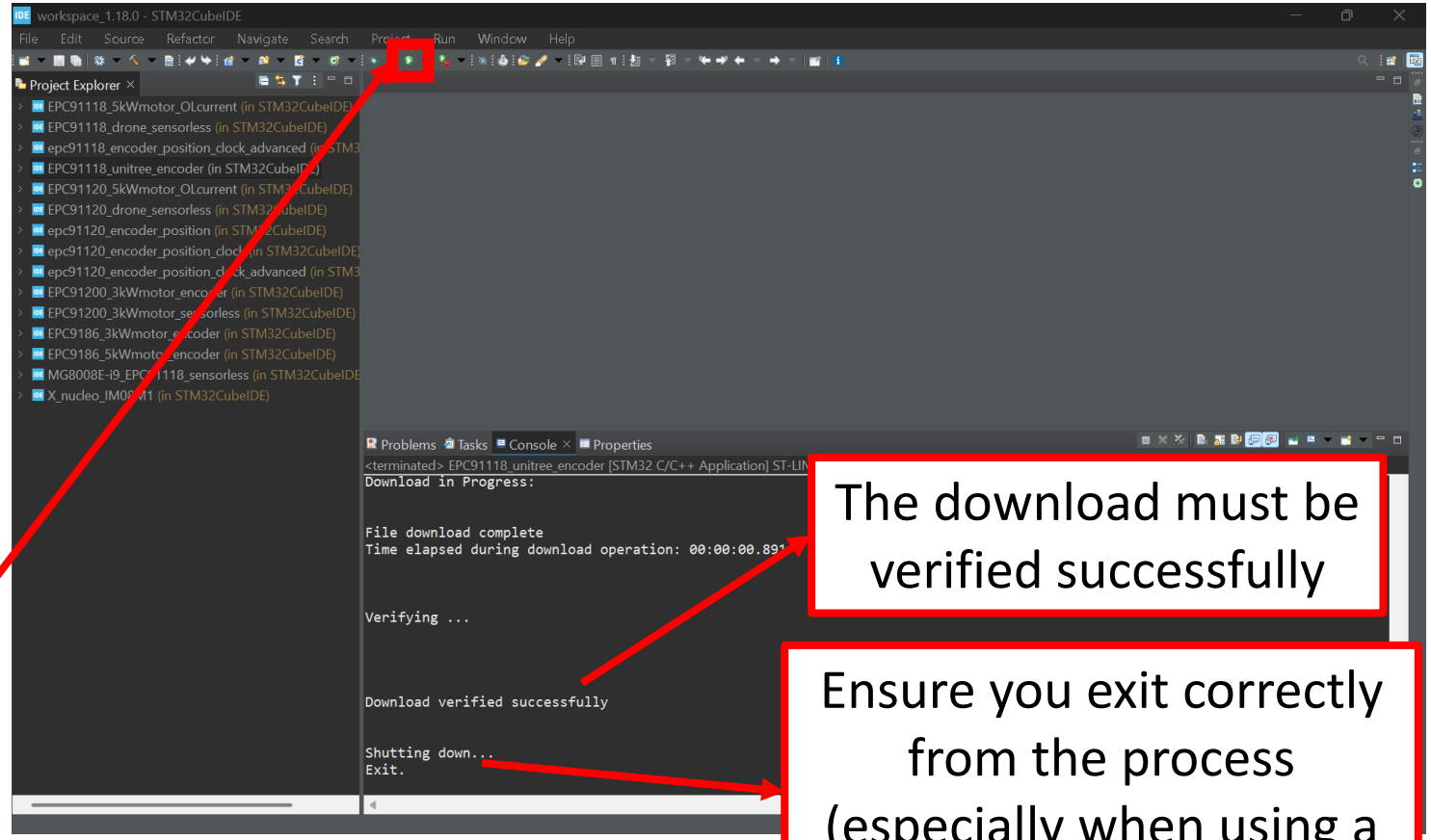
4. Flashing the firmware – hardware requirements

- 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 a power supply and to the STLINK-V3MINIE



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 the MCU after connecting STLINK-V3MINIE to PC

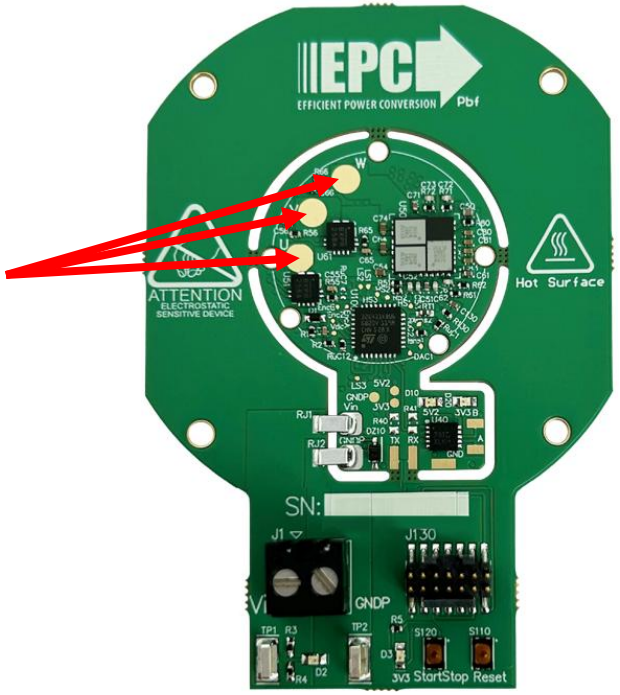


OVERVIEW OF THE PROCESS

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**

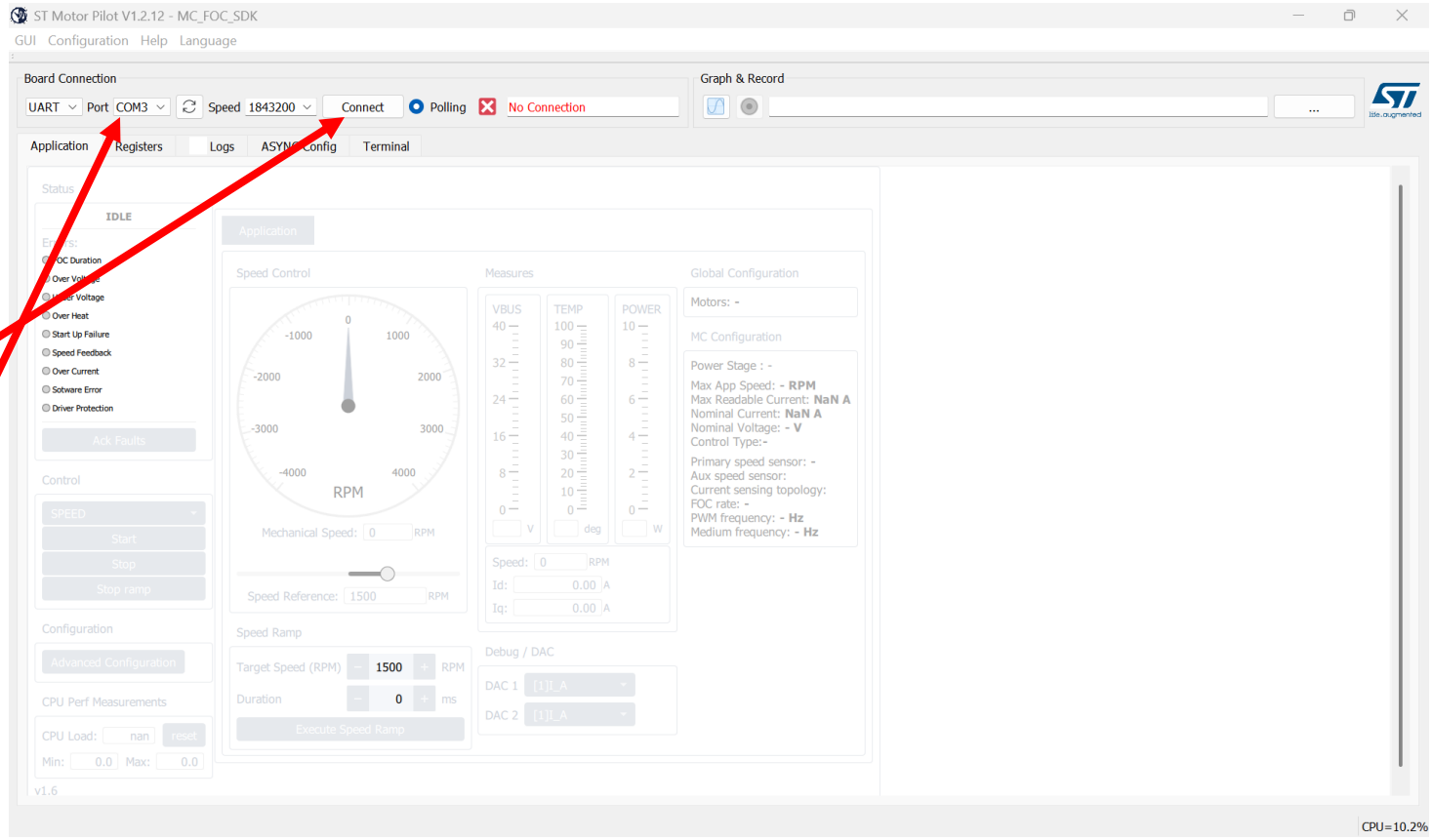
5. Driving the motor

- Solder the motor phases to the respective phase pads on the EPC91132. Pay attention to the phases' order (U, V, W).



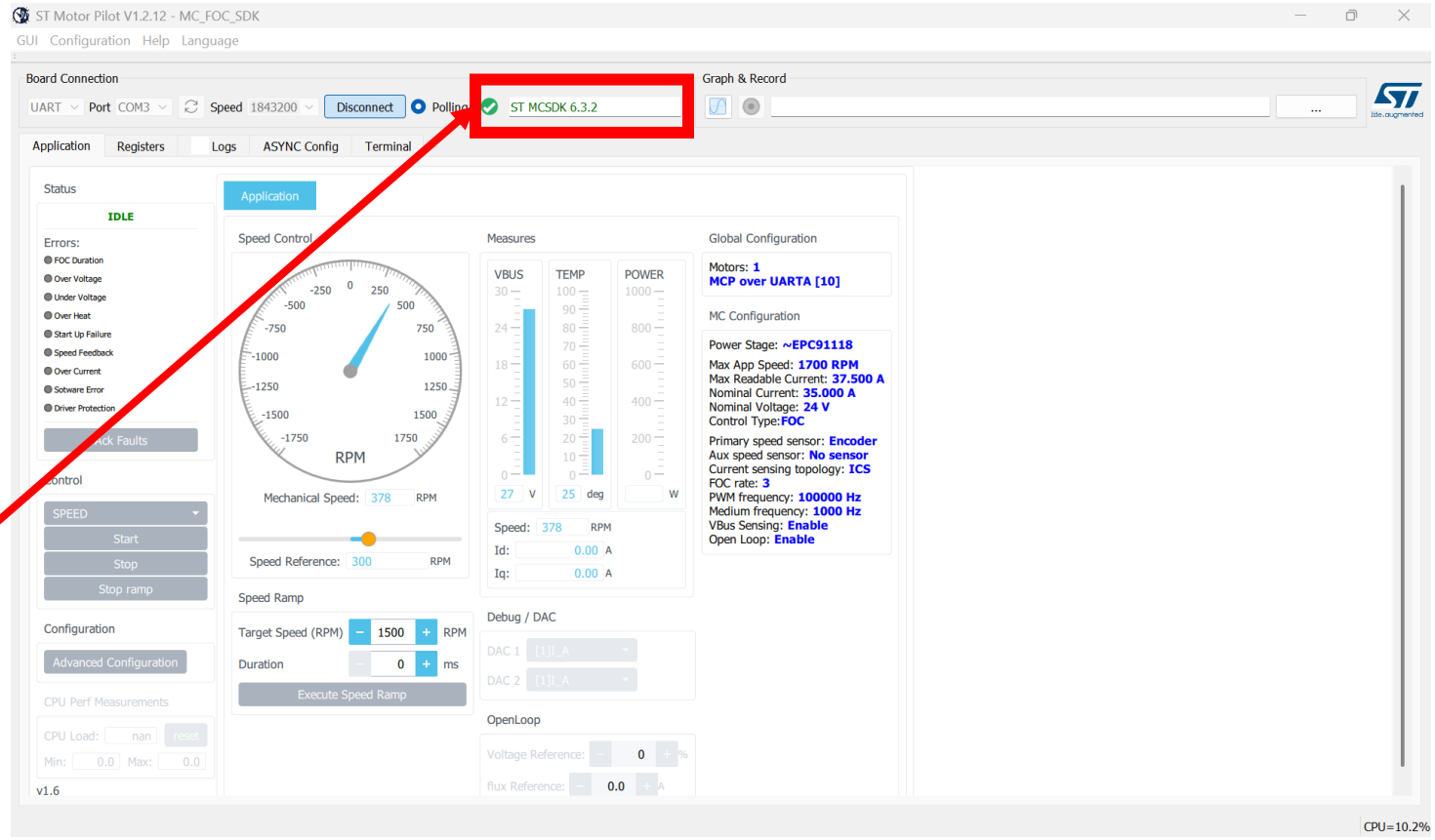
5. Driving the motor

- Solder the motor phases to the respective phase pads on the EPC91132. Pay attention to the phases' order.
- Open the ST motor pilot included in the STM32 MC SDK and try to connect the board, checking the correct COM port



5. Driving the motor

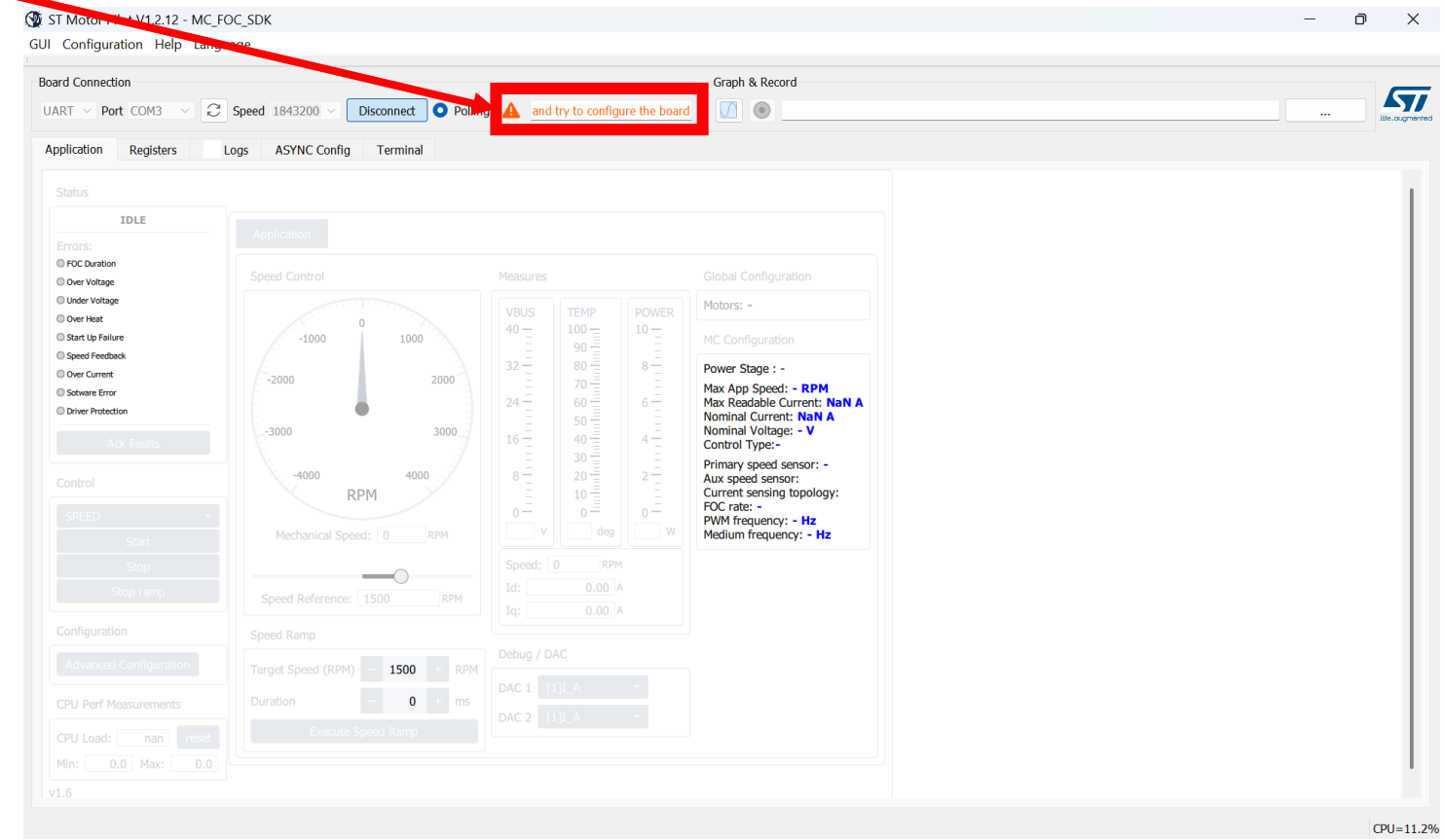
- Solder the motor phases to the respective phase pads on the EPC91132. Pay attention to the phases' order.
- Open the ST motor pilot included in the STM32 MC SDK and connect the board, checking the correct COM port
- Check for a successful connection; otherwise, check all the connections, and check if the micro is powered



5. Typical connection error

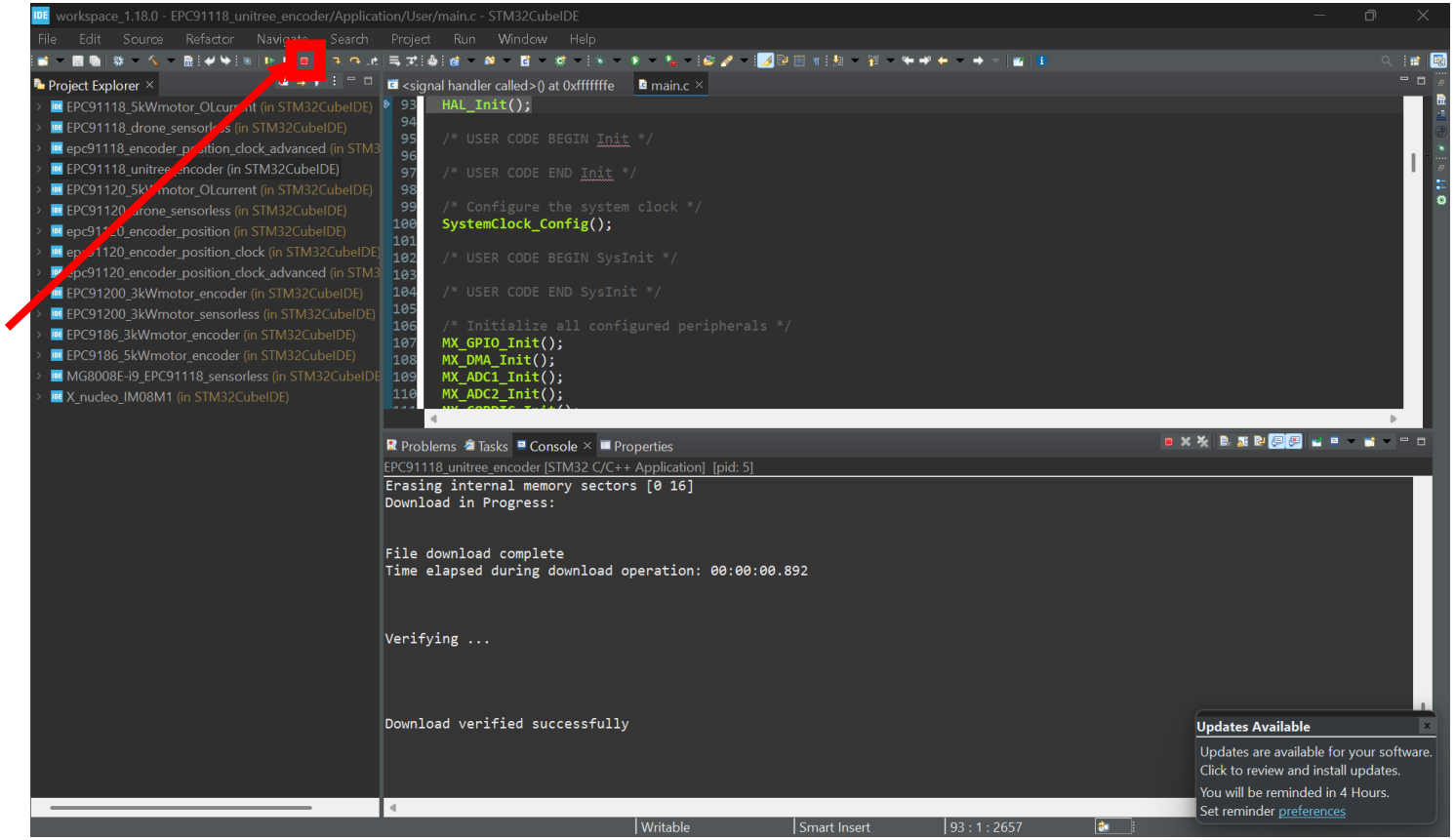
- This connection notification will appear if the debugging process is still running inside STM32CubeIDE





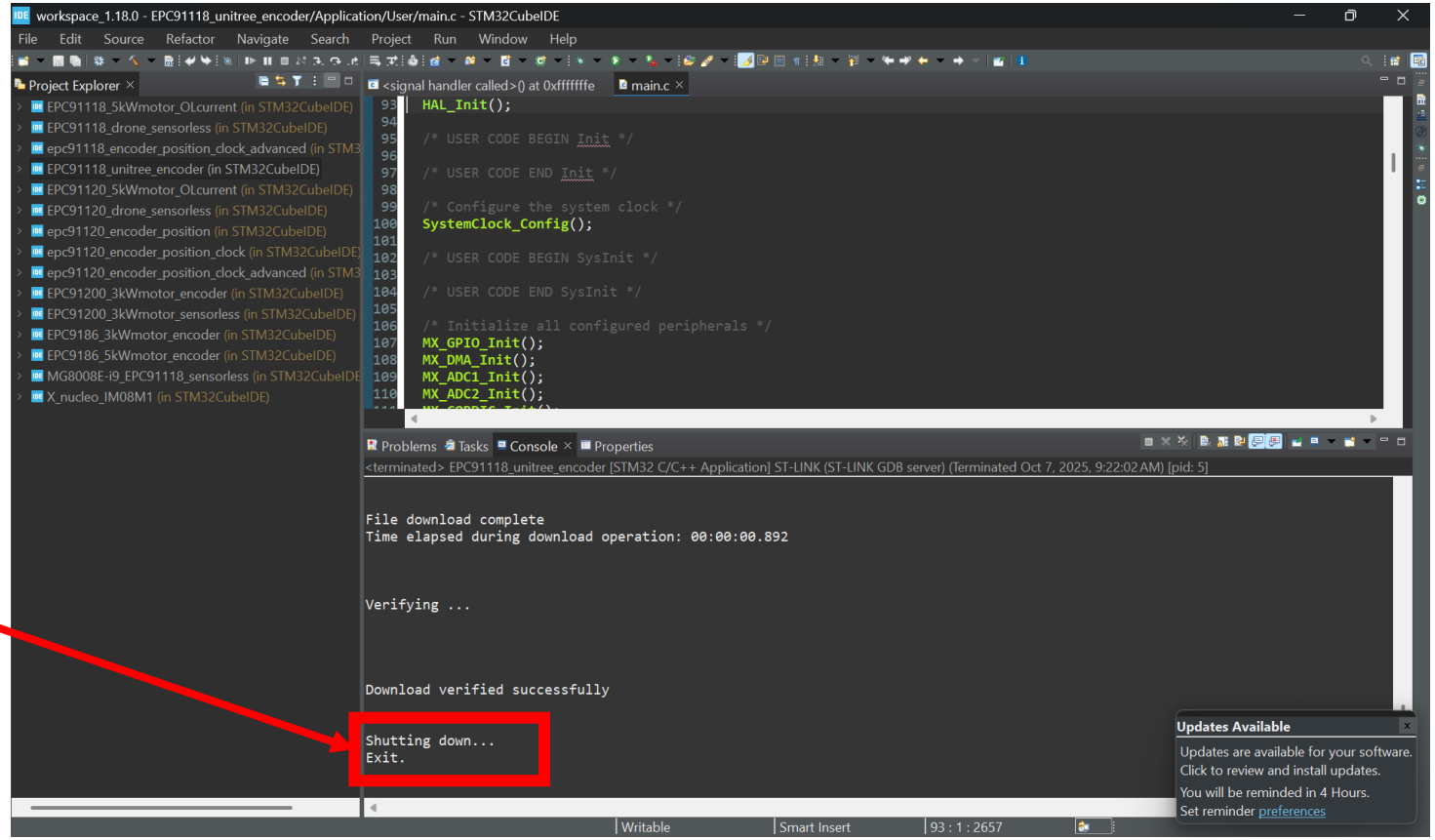
5. Typical connection error

- This connection notification will appear if the debugging process is still running inside STM32CubeIDE
- Click on the ‘Terminate’ button to interrupt the debug connection



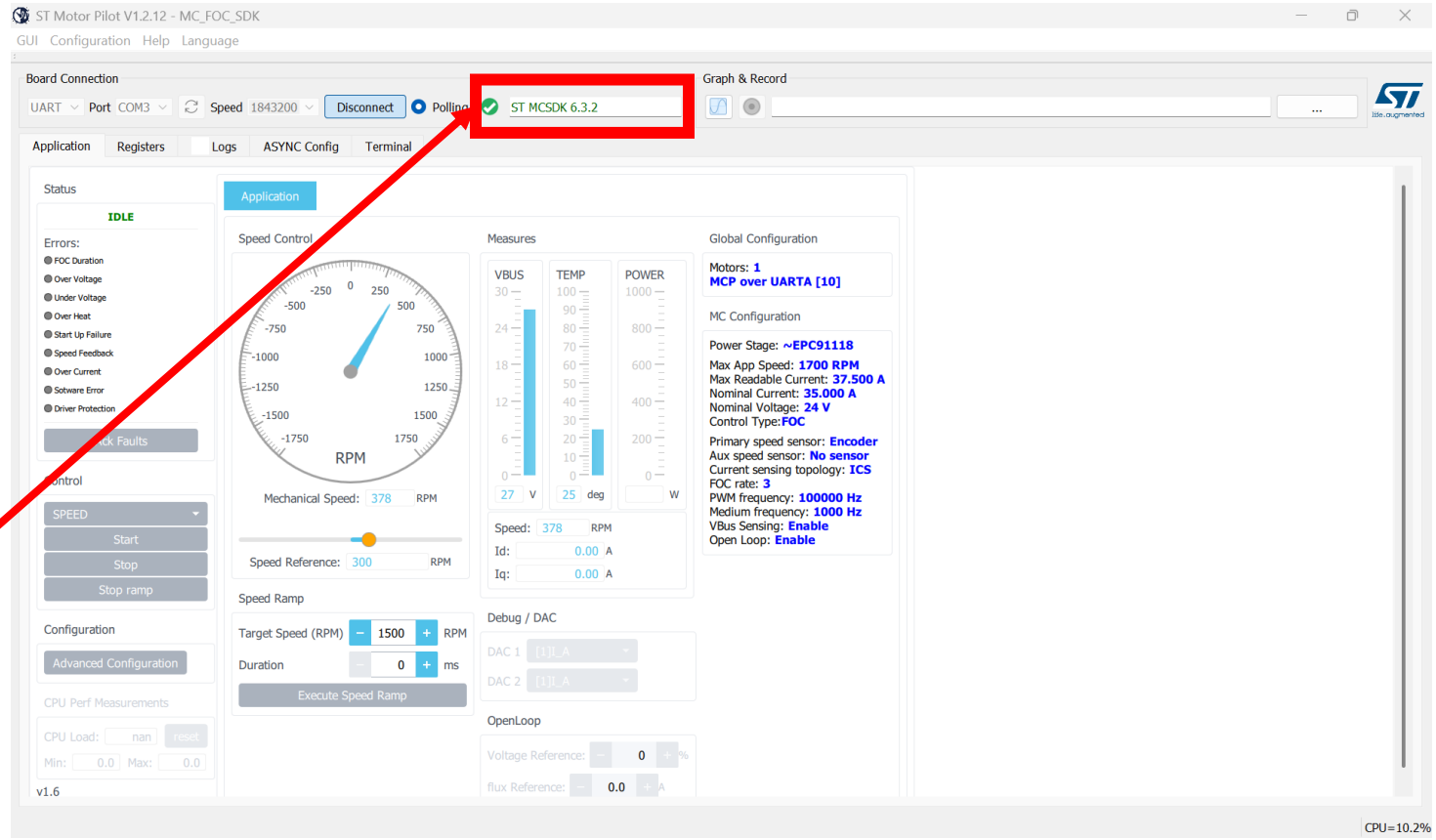
5. Typical connection error

- This connection notification will appear if the debugging process is still running inside STM32CubeIDE
- Click on the ‘Terminate’ button to interrupt the debug connection
- Ensure a proper exit from the process



5. Typical connection error

- This connection notification will appear if the debugging process is still running inside STM32CubeIDE
- Click on the ‘Terminate’ button to interrupt the debug connection
- Ensure a proper 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]

