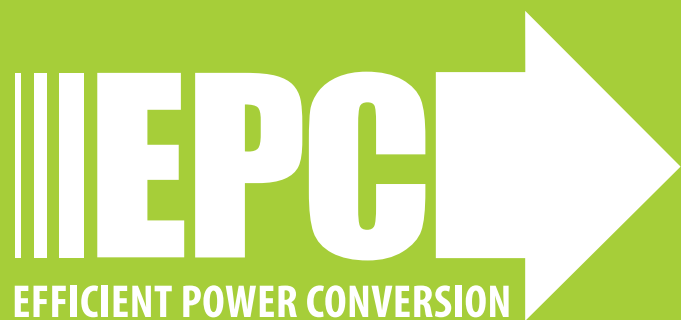


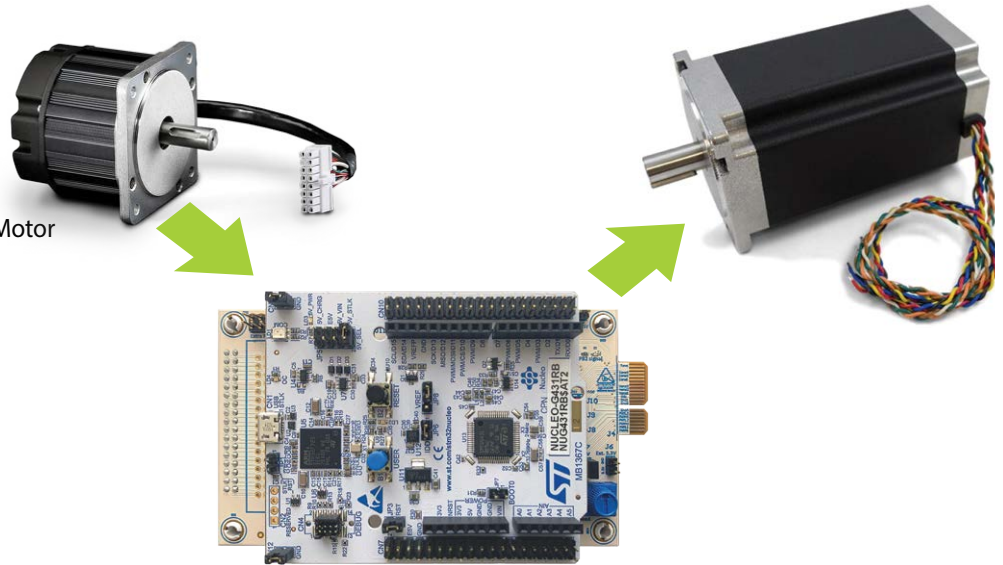
Commissioning a Motor for use with EPC motor drives that operate using ST Motor Control Workbench[®] Development Suite and EPC9147C – Rev.2.2

Revision 1.0



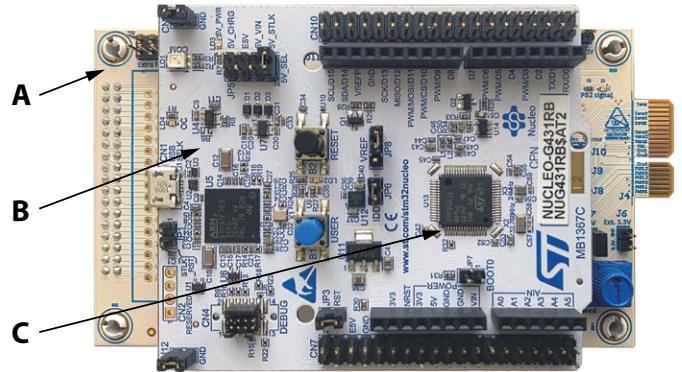
OVERVIEW OF THE PROCESS

- Background
- Equipment needed
- Measuring the motor parameters
- Inputting the motor parameters into ST Motor Control Development Suite
- Generating the control firmware:
 - Compiling
 - Build
 - Flash
- Operating the motor drive system



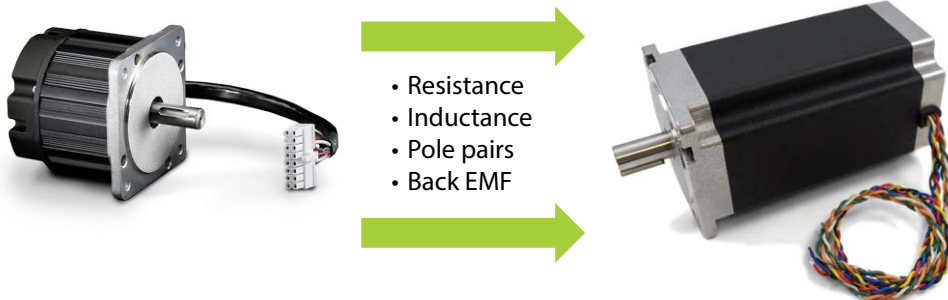
CONTROLLER BOARD BACKGROUND

- Process is for **EPC9147C Only (A)**, equipped with ST Nucleo G431RB (B) (that uses STM32G431RB (C))
 - ST Motor Control Workbench®
 - EPC9147C (Provided with motor drive KIT's)
 - Pre-programmed with a **sensor-less motor control algorithm** for a **specific motor (Teknic_M-3411P-LN-08D)**
- Note: When unboxing, **it works only with the Teknic motor (D)**



MOTOR CONTROL BACKGROUND

- For sensor-less motor control algorithms:
 - Only the three motor terminals connect to the inverter board
 - Depends on specific motor parameters (a model of the motor is used for control)
- New motor parameters **must be programmed before** operating a different motor



EQUIPMENT NEEDS, MOTOR ACCESS

Motor Access

- Direct access to the motor terminals
 - Motor terminal must be disconnected from inverter board
- Direct access to the motor shaft
 - Need to turn it by hand

Equipment

- LCR meter
 - To measure line-to-line resistance and inductance
- Oscilloscope
 - To measure line-to-line Back EMF (BEMF)



MEASURING THE MOTOR PARAMETERS

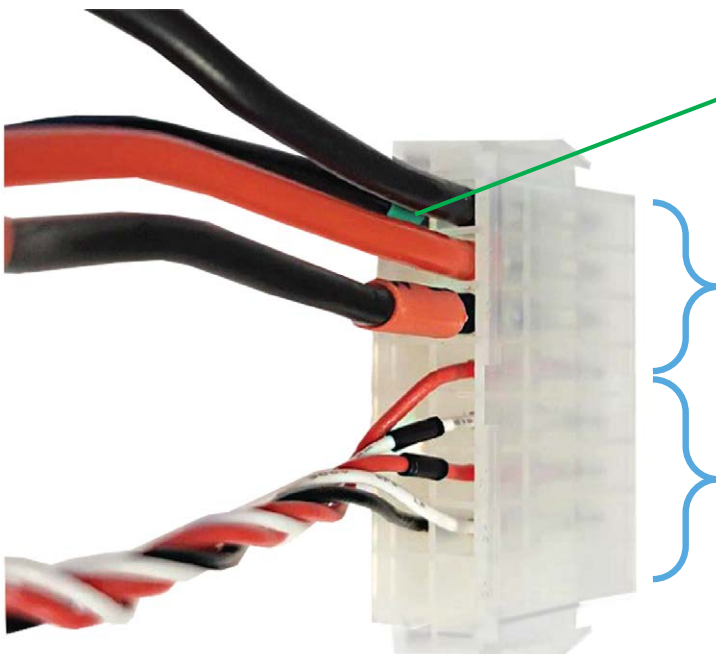
Motor Parameters Needed

- Terminal resistance (**A**)
 - Line-to-line
- Terminal inductance (**B**)
 - Line-to-line
- Pole pairs (**C**)
- Back EMF constant (**D**)



Identification of Motor Terminals

Example for Teknic Model M-3411P-LN-08D



Earth/Chassis Connection
(1x **Green** or **Clear**)



Motor Connections
(3x **thick** wires)



Shaft Encoder Connections
(multiple thin wires)
Not used for sensor-less control

Line-to-Line Resistance Measurement

1. Disconnect all three motor terminals from inverter
2. Connect **only two motor** terminals to an ohm-meter, third terminal is left floating
3. Measure the **line-to-line resistance**
4. **4-wire** resistance measurement is more accurate (if available)
5. Use line resistance by dividing by two (**R = 400 mΩ**) in ST software



This motor has $R_{L-L} = 800 \text{ m}\Omega$ line to line resistance (100 mΩ due to LCR meter leads)

Line-to-Line Inductance Measurement

1. Disconnect all three motor terminals from inverter
2. Connect **only two motor** terminals to the LCR-meter, third terminal is left floating
3. Measure the **line-to-line inductance**
4. **Note** – long leads will add inductance. Twisting the leads will help reduce inductance. More important for low inductance motors.
5. For motors with **varying inductance with shaft angle**, find the minimum and the maximum inductance values, by measuring at different angles.
6. Determine the average inductance:

$$L_{avg} = \frac{L_{min} + L_{max}}{2}$$

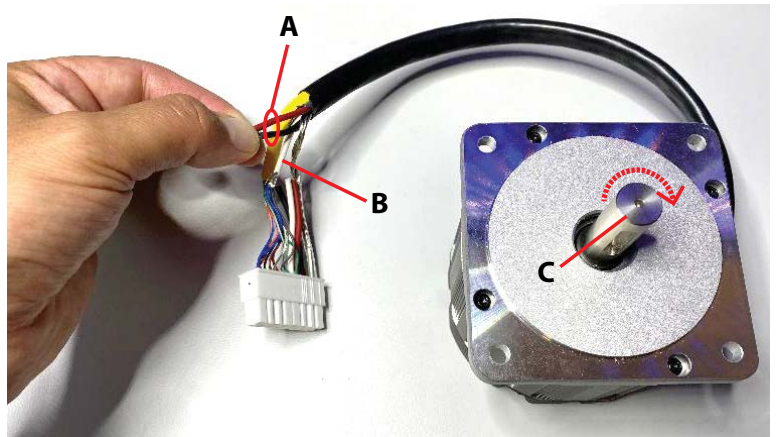
7. For the example: Rounded $932/2 \mu\text{H} = 466 \mu\text{H}$.
8. Use the same value for L_d and L_q



This motor has $L_{L-L} = 932 \mu\text{H}$ line to line inductance (LCR meter leads may also have inductance, **use autozero function if available**)

Determination of the Pole Pairs Number

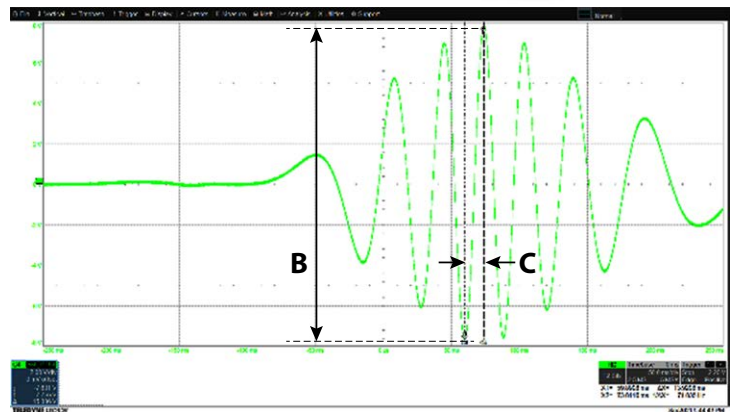
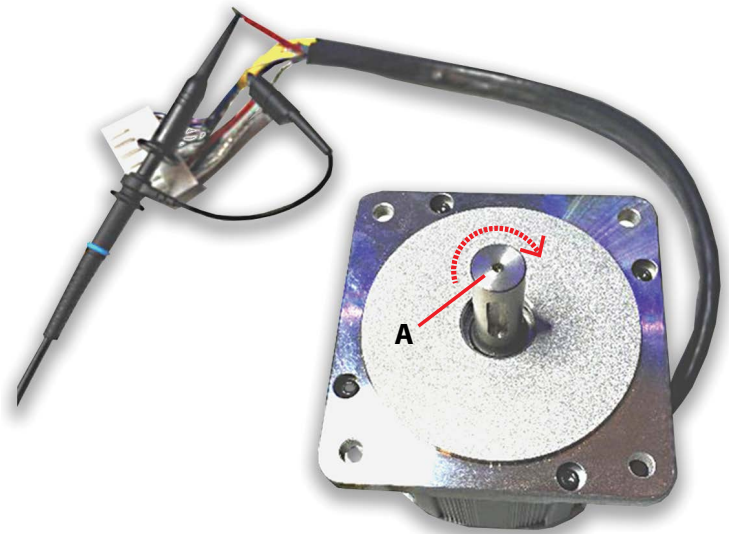
1. Disconnect all three motor terminals from inverter
2. Short **any two (A) motor** terminals, third terminal is left floating (**B**)
3. **Gently** and **slowly** hand spin the motor shaft (**C**) and make **one mechanical turn only**
 - Count the notches/steps/jumps that you feel with as the motor axle is rotated = **motor poles number**
4. Divide the **motor poles number** by 2 = **Pole Pairs number (pp)**



This motor has **pp= 4 pole pairs**

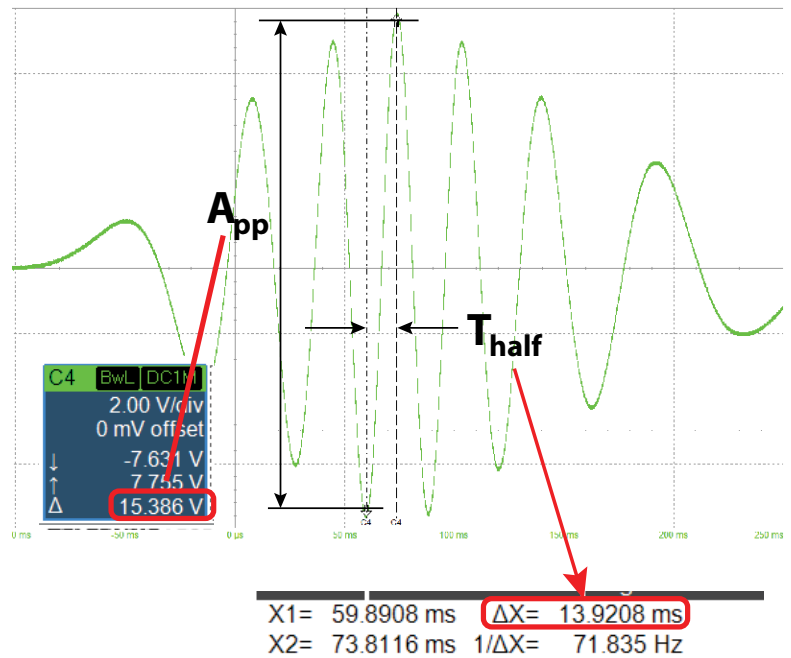
Line-to-line BEMF constant Measurement

1. Disconnect all three motor terminals from inverter
2. Connect **one** of the **motor** terminals to an oscilloscope probe **ground** lead and the **other motor** terminal to the **tip**. The third motor terminal is left floating
3. Hand **spin** the motor shaft (**A**) and record the voltage signal on the oscilloscope.
4. (**B**) Measure the **peak-to-peak** voltage of **one-half sinusoid** (details on next slide)
5. (**C**) Measure the time period between the **same two peaks** (details next slide)



Line-to-line BEMF Constant Calculation

- A_{pp} = Half-sinusoid peak-to-peak voltage amplitude
($A_{pp} = 15.836 V_{pp}$)
- T_{half} = Half sinusoid peak-to-peak period
($T_{half} = 13.92 \text{ ms}$)
- pp = Pole Pairs ($pp = 4$)
- Calculate BEMF (for 1 krpm):
 - Units: A_{pp} [V], T_{half} [s]
 - $K_e = \frac{A_{pp}}{2 \cdot \sqrt{2}} \cdot \frac{1000 \cdot pp}{60} \cdot (2 \cdot T_{half})$
 - $K_e = 11.785 pp \cdot A_{pp} \cdot T_{half}$ ← Use this
 - $K_e = 10.096 \text{ Vrms/krpm}$ for example motor (will use 10.2 in motorBench)



INSTALLING ST MOTOR CONTROL DEVELOPMENT SUITE AND INPUTTING THE MOTOR DRIVE AND MOTOR PARAMETERS INTO A PROJECT

Install STM32 Motor Control Software Development Kit (MCSDK)®

Refer to ST website to install following software, follow exactly the steps indicated in ST website

1. Register/Login at the st.com website (A).
2. Download the X-Cube-MCSDK (B)

Note: ST Motor Profiler does NOT work with EPC power boards. If you try, you may damage the EPC power board.

3. Install the X-Cube-MCSDK downloaded executable file, following the default installation settings.

Note: This guide is showing the process with the MCSDK kit. In principle same steps apply also for the MCSDK-Y kit. The only difference is that the .stmcx file needs to be created for MCSDK-Y kit. The .stmcx file for MCSDK-Y kit can be created using same parameters as they are shown in this guide.

It may be necessary to disable Antivirus and firewall programs during installation

A

B

| Part Number | General Description | Download | All versions |
|------------------|--|------------|----------------|
| X-CUBE-MCSDK | STM32 Motor Control Software Development Kit LEGACY (MCSDK) | Get latest | |
| + X-CUBE-MCSDK-Y | STM32 Motor Control Software Development Kit Rev Y (MCSDK-Y) | Get latest | Select version |

Commissioning a Motor for use with EPC motor drives operating ST Motor Control

Install STM32CubeMX

Refer to ST website to install following software, follow exactly the steps indicated in ST website

1. Register/Login at the st.com website (A)
2. Download the STM32CubeMX-Win (B)

Note: ST Motor Profiler does NOT work with EPC power boards. If you try, you may damage the EPC power board.

3. Install the STM32CubeMX-Win downloaded executable file, following the default installation settings.

Note: This procedure has been tested with version 6.2.0 and version 6.3.0

It may be better to disable Antivirus and firewall programs during installation

A

B

| Part Number | General Description | Latest version | Download | All versions |
|-------------------|---|----------------|------------|----------------|
| + STM32CubeMX-Lin | STM32Cube init code generator for Linux | 6.4.0 | Get latest | Select version |
| + STM32CubeMX-Mac | STM32Cube init code generator for macOS | 6.4.0 | Get latest | Select version |
| + STM32CubeMX-Win | STM32Cube init code generator for Windows | 6.4.0 | Get latest | Select version |

Install STM32Cube-IDE

Refer to ST website to install following software, follow exactly the steps indicated in ST website

1. Register/Login at the st.com website (A)
2. Download the STM32CubeIDE-Win (B)
3. Install the STM32CubeMX-Win downloaded executable file, following the default installation settings.

Note: This procedure has been tested with version 1.6.0 and version 1.7.0

It may be better to disable Antivirus and firewall programs during installation

A

B

| Part Number | General Description | Latest version | Download | All versions |
|--------------------|--------------------------------------|----------------|------------|----------------|
| + STM32CubeIDE-DEB | STM32CubeIDE Debian Linux Installer | 1.8.0 | Get latest | Select version |
| + STM32CubeIDE-Lnx | STM32CubeIDE Generic Linux Installer | 1.8.0 | Get latest | Select version |
| + STM32CubeIDE-Mac | STM32CubeIDE macOS Installer | 1.8.0 | Get latest | Select version |
| + STM32CubeIDE-RPM | STM32CubeIDE RPM Linux Installer | 1.8.0 | Get latest | Select version |
| + STM32CubeIDE-Win | STM32CubeIDE Windows Installer | 1.8.0 | Get latest | Select version |

Commissioning a Motor for use with EPC motor drives operating ST Motor Control

Install STM32CubeProg

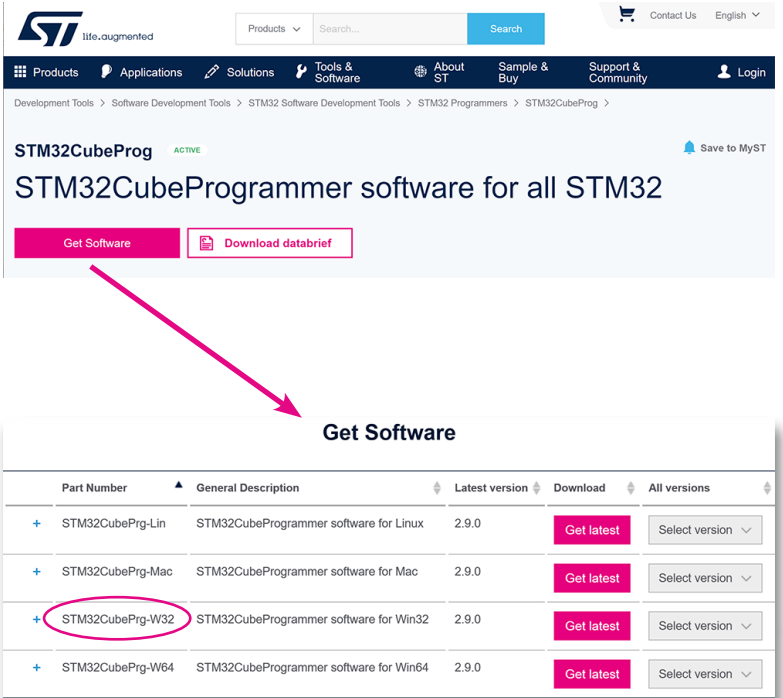
Refer to ST website to install following software, follow exactly the steps indicated in ST website

1. Register/Login at the [st.com](https://www.st.com) website (A)
2. Download the STM32CubeProg (B)
3. Install the STM32CubeProg downloaded executable file, following the default installation settings.

Note: The STM32CubeProg can be used to flash the device with a .elf file, if the .elf file is already available. The steps are explained in ECP9147C quick start guide. The aim of this guide is to show the process to generate the .elf file

It may be better to disable Antivirus and firewall programs during installation

A



B

| Part Number | General Description | Latest version | Download | All versions |
|------------------|--|----------------|----------------------------|----------------|
| STM32CubePrg-Lin | STM32CubeProgrammer software for Linux | 2.9.0 | Get latest | Select version |
| STM32CubePrg-Mac | STM32CubeProgrammer software for Mac | 2.9.0 | Get latest | Select version |
| STM32CubePrg-W32 | STM32CubeProgrammer software for Win32 | 2.9.0 | Get latest | Select version |
| STM32CubePrg-W64 | STM32CubeProgrammer software for Win64 | 2.9.0 | Get latest | Select version |

Download the specific .stmcx file for the EPC power board

Refer to [EPC website](https://www.epc-co.com) to download the proper ST Motor Control Workbench .stmcx definition file.

As an example, EPC9145's product page, the file link can be found at :

<https://epc-co.com/epc/Products/DemoBoards/EPC9145.aspx>

The filename is:

G431-EPC9145-DummyNema34_50k_100n.stmcx and needs to be loaded in ST Motor Control Workbench program.

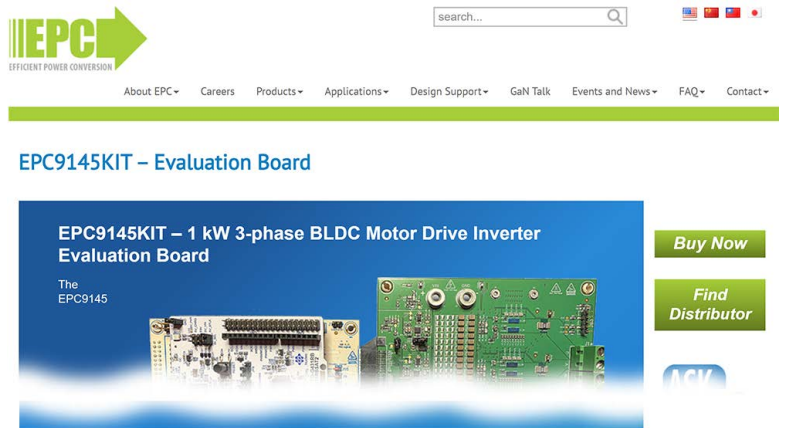
Note: This file was made for MCSDK Kit 5.4.x and it will not be loaded by the MCSDK-Y kit software.

Folder naming directions: The working directory should have short path and should not contain spaces or symbols.

Example:

C:\motor_software (acceptable: no spaces and no special characters - Note: an underscore symbol ("_") is not considered a special character)

C:\motor & software (unacceptable: uses spaces and a special character)



EPC9145KIT - Evaluation Board

[Buy Now](#)

[Find Distributor](#)

Quick Start Guide

- EPC9145 Quick Start Guide
- EPC9147C Quick Start Guide
- EPC9147E Quick Start Guide

Featured Product

EPC2206

Support Files

- EPC9145 Schematic
- EPC9147E Schematic
- EPC9145 Gerber Files
- EPC9147E Gerber Files
- EPC9145 Bill of Materials
- EPC9147E Bill of Materials
- Microchip motorBench® Development Suite
- Commissioning a Motor for use with EPC motor drives that operate using Microchip motorBench® Development Suite
- EPC9145 Firmware Hex File

Commissioning a Motor for use with EPC motor drives operating ST Motor Control

Launch Motor Control Workbench®

1. Start Motor Control Workbench 5.4.x (A)
2. Load specific sample project (B)
or click on Recent Project's filename to open it (C)
3. Once the project is loaded, it is already set with proper parameters for **specific** power board and for **specific** motor (Teknic_M-3411P-LN-08D)

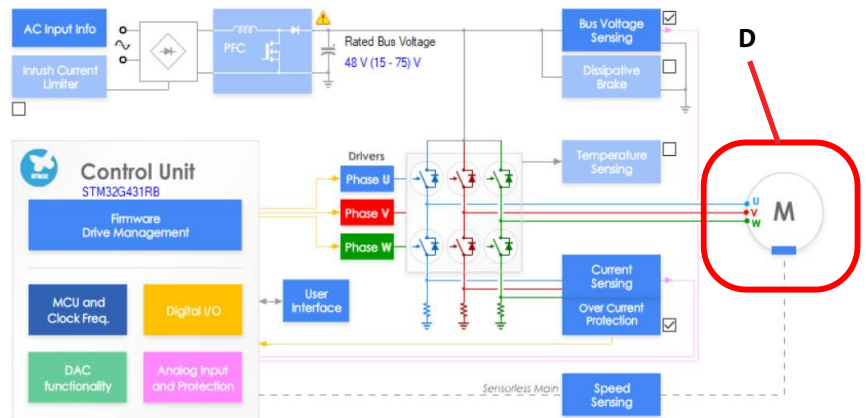


| Filename | Type | SDK | MCUs | control board | power board | motor |
|---|--------|-------|-------------|---------------|------------------|------------------|
| G431-EPC9145-DummyNema34_50k_100n.stmcx | SINGLE | 5.4.5 | STM32G431RB | NUCLEO-G431RB | X-NUCLEO-IHM08M1 | Dummy Nema34 |
| G431-EPC9145-3sh_Nema23_2.stmcx | SINGLE | 5.4.5 | STM32G431RB | NUCLEO-G431RB | X-NUCLEO-IHM08M1 | Nema23_210413_2 |
| G431-IHM08-3sh_Nema23_2.stmcx | SINGLE | 5.4.5 | STM32G431RB | NUCLEO-G431RB | X-NUCLEO-IHM08M1 | Nema23_210413_2 |
| G431-IHM08-3sh_Nema3426_2.stmcx | SINGLE | 5.4.5 | STM32G431RB | NUCLEO-G431RB | X-NUCLEO-IHM08M1 | Teknic3426p_dyno |
| G431-IHM08-3sh.stmcx | SINGLE | 5.4.5 | STM32G431RB | NUCLEO-G431RB | X-NUCLEO-IHM08M1 | Nema23_210413_2 |

Note:

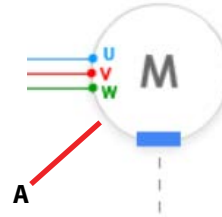
1. If you downloaded the project for the specific power board, it is already set properly to work for the **Teknic_M-3411P-LN-08D** motor.
2. To adapt to a new motor you have to change only the motor parameters by clicking on the motor icon circled in red (D)

Motor: Dummy Nema34 - Control Board: NUCLEO-G431RB - Power Board: X-NUCLEO-IHM08M1



Motor parameters dialog box

- Click on motor icon (A) to visualize the motor parameters
- Parameters shown here (B) are for the **Teknic_M-3411P-LN-08D** motor.
- To adapt to a new motor, change only the motor parameters.
- **Note:** the Rs and Ls parameters (C) are referred to phase to neutral point (i.e. the resistance and the inductance measured phase to phase must be divided by 2).



Motor - Parameters

Motor Sensors

Magnetic structure Surface Mounted PMSM

Electrical parameters

| | |
|------------------------|---------------|
| Pole Pairs | 4 |
| Max. Application Speed | 3000 rpm |
| Nominal Current | 6.00 Apk |
| Nominal DC Voltage | 48.0 V |
| Rs | 0.40 Ohm |
| Ls | 0.466 mH |
| B-Emf constant | 10.2 Vms/krpm |

C

Motor - Parameters

Motor Sensors

Sensors

| | |
|---|--------------------------------------|
| <input type="checkbox"/> Hall sensors | Sensors displacement 120 deg |
| | Placement electrical angle 300 deg |
| <input type="checkbox"/> Quadrature encoder | Pulses per mechanical revolution 400 |
| <input type="checkbox"/> Encoder Alignment | |

B

Motor Control Workbench parameters

As an example, all parameters related to the EPC9145 are shown in the next figures.

For other EPC power boards, download the corresponding **.stmcx** file from EPC website.

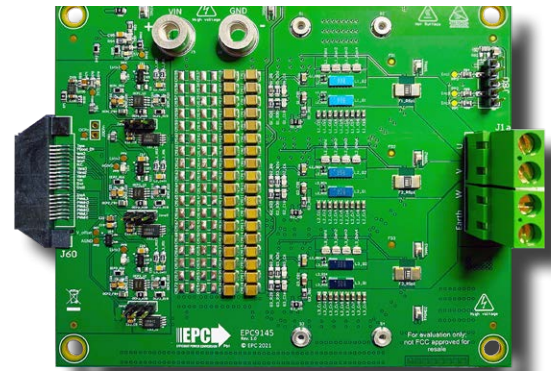
Warning: ST Motor Profiler does NOT work with EPC power boards. If you try, you may damage the EPC power board.

The next figures can be also used to modify an existing project originally made for the **ST MCSDK-Y** kit.

Note: This guide is based on ST MCSDK kit only.

The following figures are optional and are not needed if only motor parameters need to be changed. In that case you can skip and proceed to the **Code Generation** steps.

Refer to **ST Motor Control Workbench Guide** for further information.



EPC9145 board

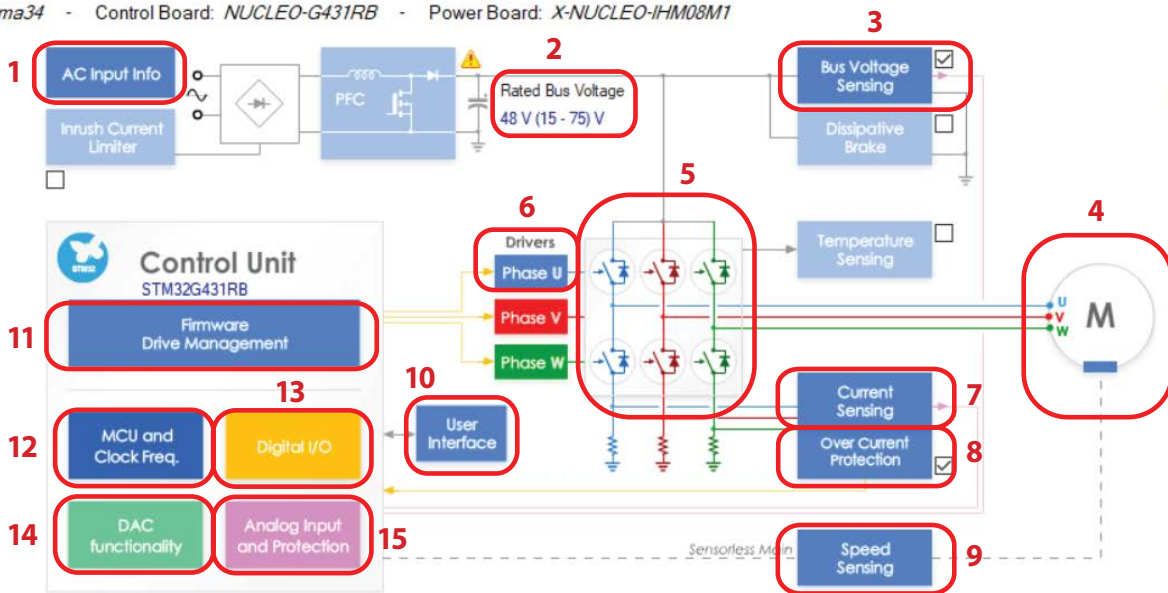
Example MCSDK project for EPC9145

Overview

Red circled boxes/areas below indicate which categories have their parameters set in the following example project.

Note: Make sure that all parameters set match exactly as shown.

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*

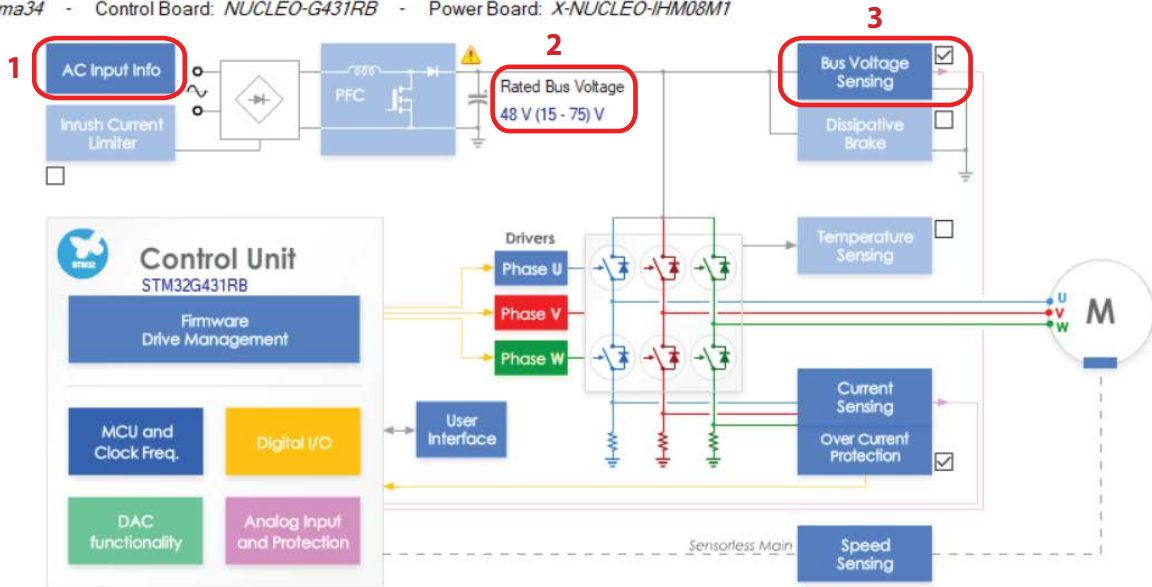


Legend:

- | | |
|--|--|
| 1. AC Input info | 9. Drive Management - Speed Position Feedback Management |
| 2. Power Stage - Rated Bus Voltage Info | 10. User Interface |
| 3. Power Stage - Bus Voltage Sensing | 11. Drive Management |
| 4. Motor Parameters | 12. MCU and Clock Frequency Selection |
| 5. Power Stage - Power Switches | 13. Control Stage - Digital I/O |
| 6. Power Stage - Driving Signals Polarity - U Driver | 14. Control Stage - DAC Functionality |
| 7. Power Stage - Current Sensing | 15. Control Stage - Analog Input and Protection |
| 8. Power Stage - Overcurrent Protection | |

Specific settings for Example MCSDK project for EPC9145

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



1 AC Input Info

Hardware Settings

Voltage: custom

Minimum: Vrms 50 Hz 60 Hz

Maximum: Vrms

Nominal: Vrms

Firmware protection

Over-voltage

Set intervention threshold to maximum power stage input voltage

Over voltage: Vrms

3 Power Stage - Bus Voltage Sensing

Power Stage | Control Stage

V_{Bus}

R1: kOhm

R2: kOhm

V_{Bus} feedback

R3:

Bus voltage divider 1/...:

2 Power Stage - Rated Bus Voltage Info

Rated Voltage

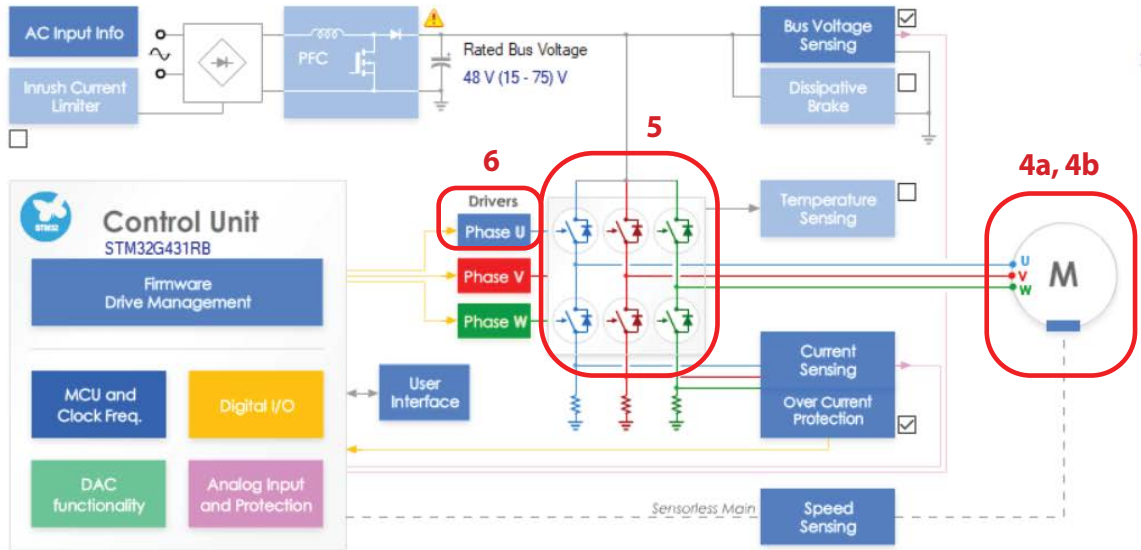
Min rated voltage: V

Max rated voltage: V

Nominal voltage: V

Specific settings for Example MCSDK project for EPC9145 (continued)

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



4a Motor - Parameters

Motor Sensors

Magnetic structure: Surface Mounted PMSM

Electrical parameters:

- Pole Pairs: 4
- Max. Application Speed: 3000 rpm
- Nominal Current: 6.00 Apk
- Nominal DC Voltage: 48.0 V
- R_s : 0.40 Ohm
- L_s : 0.466 mH
- B-Emf constant: 10.2 Vms/krpm

5 Power Stage - Power Switches

Min dead-time: 100 ns

Max switching frequency: 100 kHz

Done

4b Motor - Parameters

Motor Sensors

Hall sensors

- Sensors displacement: 120 deg
- Placement electrical angle: 300 deg

Quadrature encoder

- Pulses per mechanical revolution: 400

Encoder Alignment

6 Power Stage - Driving Signals Polarity - U Driver

High side driving signal

Polarity: Active high

Low side driving signal

Complemented from high side:

Polarity: Active high

HW inserted dead time: 500 ns

Driver enabling signal

signal:

Polarity: Active high

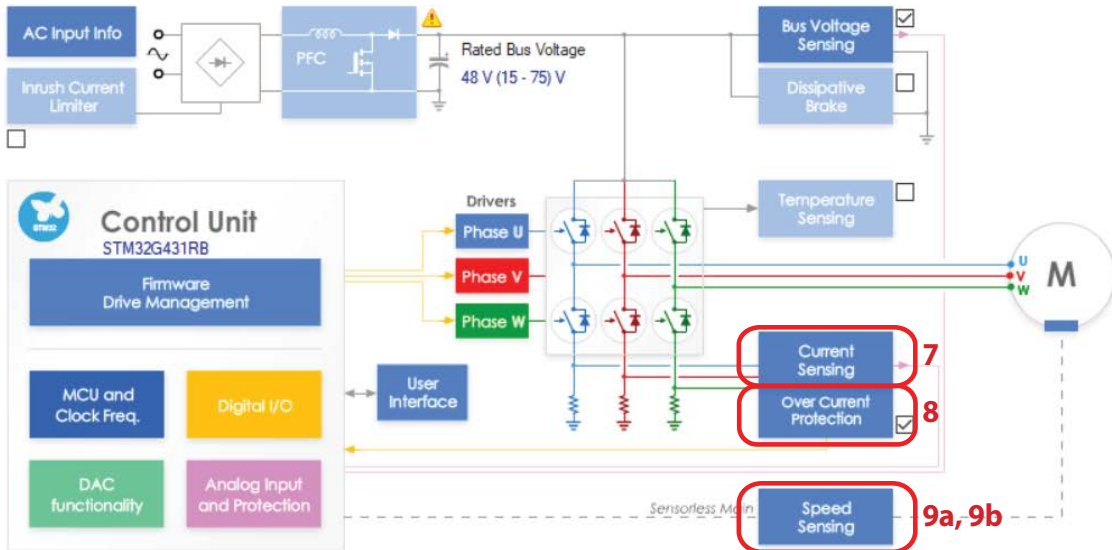
Force same values for U,V,W Driver

Share signal enable

Done

Specific settings for Example MCSDK project for EPC9145 (continued)

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



7 Power Stage - Current Sensing

Current sensor and signal conditioning

Current reading topology: Three Shunt Resistors

ICS gain: 1.000 V/A

Shunt resistor(s) value: 0.00100 ohm

Amplification on board:

Amplifying network gain: 20.00 Calculate

T-rise: 1000 ns

T-noise: 1000 ns

Max Readable Current: 82.500 A

Done

8 Power Stage - Over Current Protection

Over Current Protection

Comparator threshold: 0.30 V

Over current network offset: 0.00 V

Over current network gain: 0.0100 V/A

Expected over-current threshold: 30.0000 A

Over-current feedback signal polarity: Active low

Over-current protection disabling network

Over-current protection disabling network polarity: Active high << Less

Done

9a Drive Management - Speed Position Feedback Management

Main sensor Auxiliary sensor

Sensor selection: Sensor-less (Observer+PLL)

Max measurement errors number before fault: 30

Observer+PLL

Variance threshold: 10.00 %

Average speed depth for speed loop: 64

Average speed depth for observer equations: 64

B-emf consistency tolerance: 100.00 %

B-emf consistency gain: 100.00 %

Manual editing enabled

Observer

G1: -24154

G2: 18665

PLL

255 / 16384 P

7 / 65536 I

Back compatibility

9b Drive Management - Speed Position Feedback Management

Main sensor Auxiliary sensor

Enable auxiliary sensor

Sensor selection: Sensor-less (Observer+Cordic)

Max measurement errors number before fault: 30

Observer+Cordic

Variance threshold: 400.0 %

Average speed FIFO depth for speed loop: 64

Average speed FIFO depth for observer equations: 64

B-emf consistency tolerance: 100.00 %

B-emf consistency gain: 100.00 %

Maximum application acceleration: 6000 rpm/s

B-emf quality factor: 0.017

Manual editing enabled

Observer

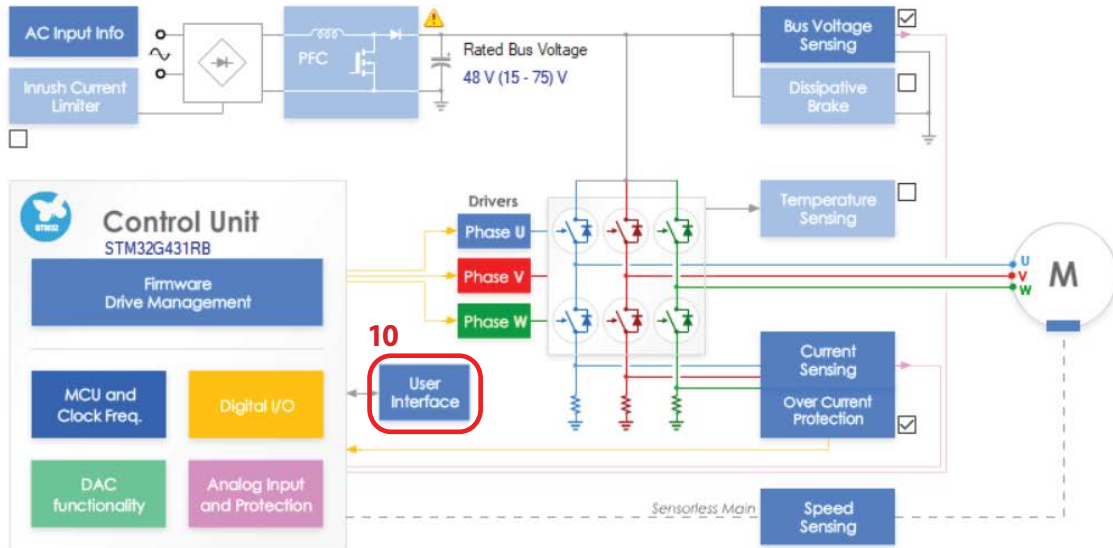
G1: -24154

G2: 18665

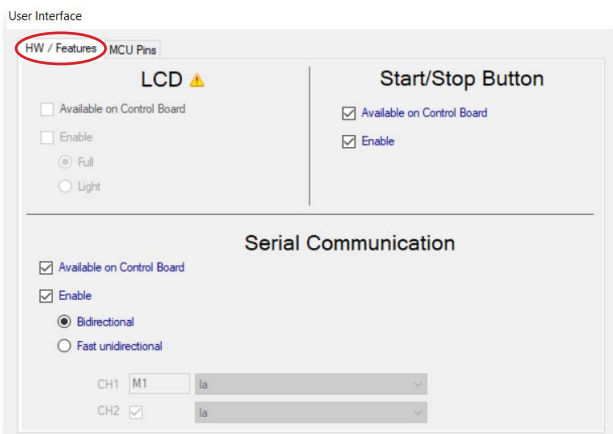
Back compatibility

Specific settings for Example MCSDK project for EPC9145 (continued)

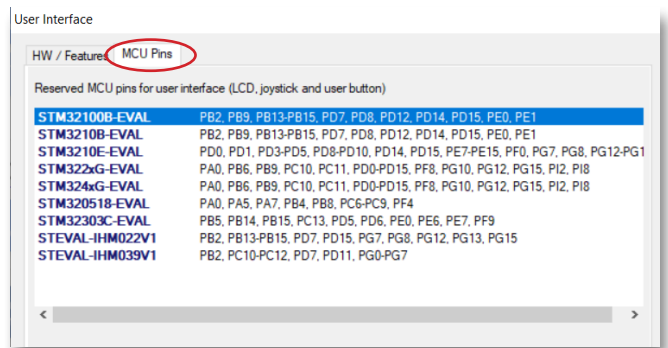
Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



10a

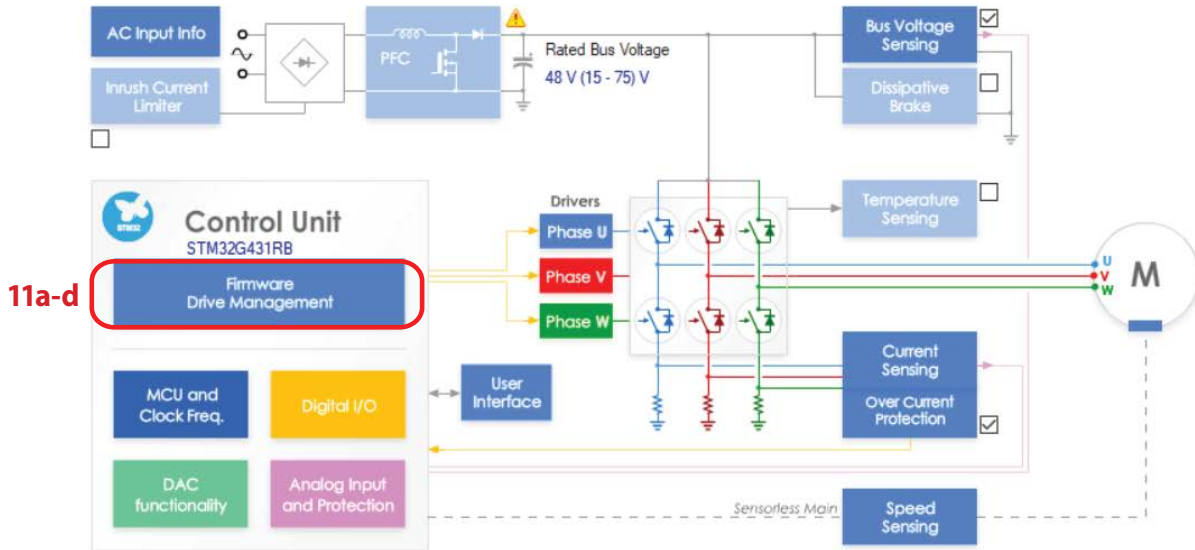


10b



Specific settings for Example MCSDK project for EPC9145 (continued)

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



11a

Drive Management - Start-up parameters

Sensorless rev-up settings
 On-the-fly startup

Profile
 Basic
 Advanced customized

Include alignment before ramp-up
 Duration: 700 ms
 Alignment electrical angle: 90 deg
 Final current ramp value: 0.60 A
 Speed ramp duration: 2000 ms
 Speed ramp final value: 1000 rpm
 Current ramp initial value: 0.60 A
 Current ramp final value: 0.70 A
 Current ramp duration: 300 ms

Consecutive successful start-up output tests: 2
 Minimum start up output speed: 250 rpm
 Estimated speed Band tolerance upper limit: 106.25 %
 Estimated speed Band tolerance lower limit: 93.75 %

Rev-up to FOC switch over
 Enable
 Duration: 25 ms

Done

11b

Drive Management - Drive Settings

PWM generation and current reading
 PWM frequency: 50000 Hz
 High sides PWM idle state: Turn-off
 Low side signals and dead time
 SW inserted dead-time: 100 ns
 Low sides PWM idle state: Turn-off

Speed regulator
 Execution rate: 1.0 ms

Default settings
 Control mode: Speed control
 Target speed: 1000 rpm
 Target stator current flux component: 0.00 A
 Target stator current torque: 0.00 A

Torque and flux regulators
 Execution rate: 2 PWM periods
 Cut-off frequency: 200 rad/s

Torque: 2625 / 8192 P, 180 / 16384 I
 Flux: 2625 / 8192 P, 180 / 16384 I

Manual editing enabled

Done

11c

Drive Management - Additional Features and PFC settings

Flux weakening
 MTPA
 Feed Forward
 Sensorless speed feedback

Inrush Current Limiter

Done

11d

Drive Management - Sensing Enabling and Firmware Protections

DC Bus voltage sensing
 Enable

Over-voltage
 Motor control
 Set intervention threshold to power stage max rated voltage
 Over-voltage threshold: 75 V
 On over voltage: Disable PWM generation
 On over-voltage, disable over-current protection by HW

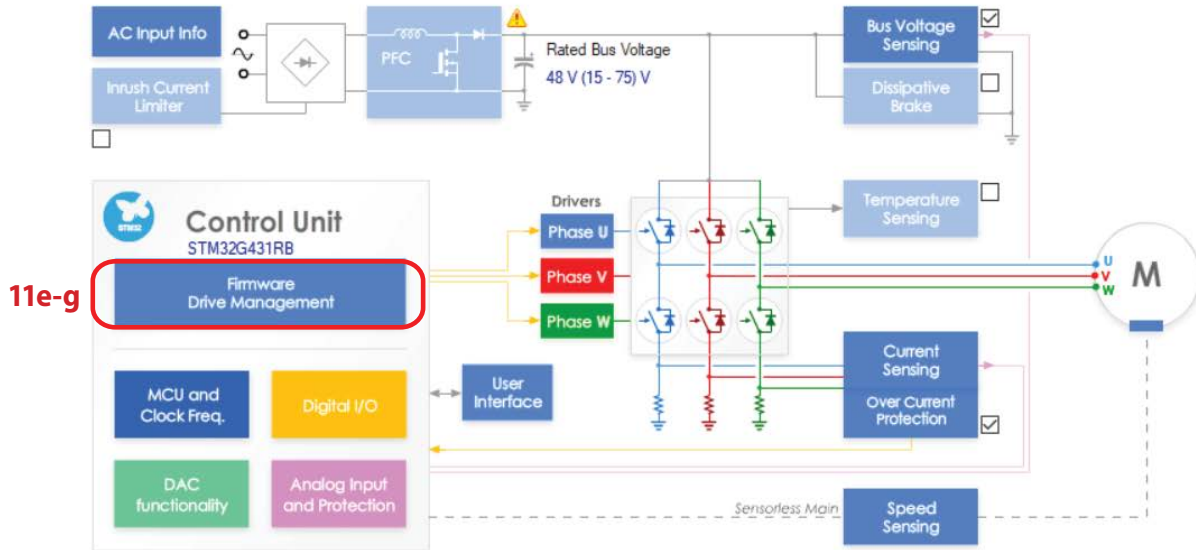
Under-voltage
 Enable
 Set intervention threshold to power stage min rated voltage
 Under-voltage threshold: 15 V

Temperature Sensing | AC Input

Done

Specific settings for Example MCSDK project for EPC9145 (continued)

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



11e

Temperature Sensing

Hardware Settings

Temperature sensing - V0: 11055 mV
 Temperature sensing - T0: 25.0 °C
 ΔV/ΔT: 22.7 mV/°C
 Max working temperature on sensor: 110 °C

Firmware sensing & protection

Enable sensing
 Enable protection

Over-Temperature

Set intervention threshold to power stage max working temperature

Overtemperature threshold: 110 °C
 Hysteresis: 10 °C

Done

11f

AC Input Info

Hardware Settings

Voltage: custom
 Minimum: 160 Vrms (50 Hz)
 Maximum: 340 Vrms (60 Hz)
 Nominal: 230 Vrms

Firmware protection

Over-voltage

Set intervention threshold to maximum power stage input voltage

Over voltage: 340 Vrms

Done

11g

FreeRTOS

enable FreeRTOS

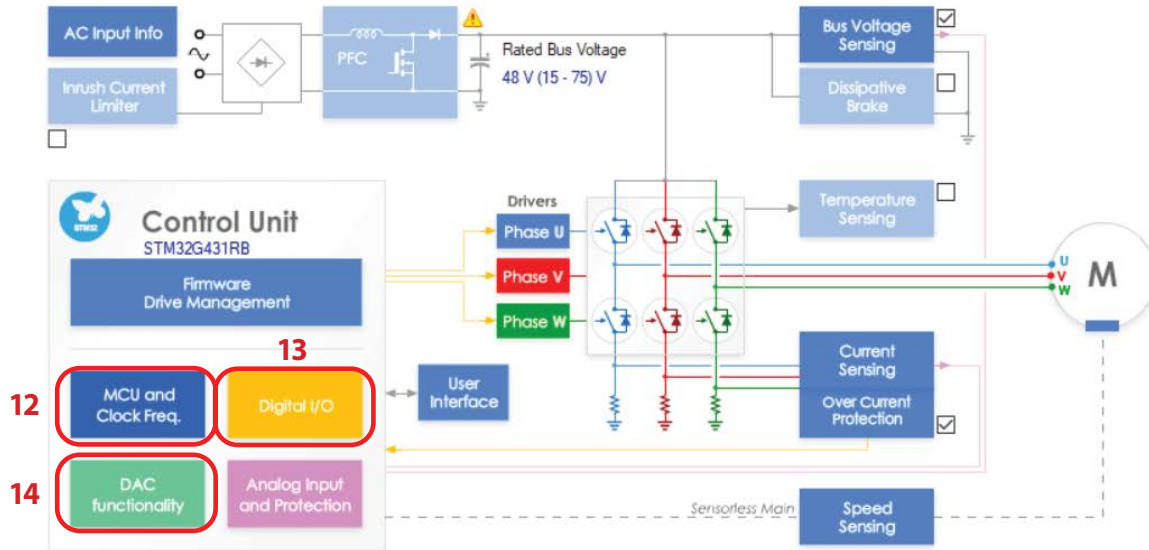
Enable MC IRQ to use FreeRTOS functions

Timer: TIM6

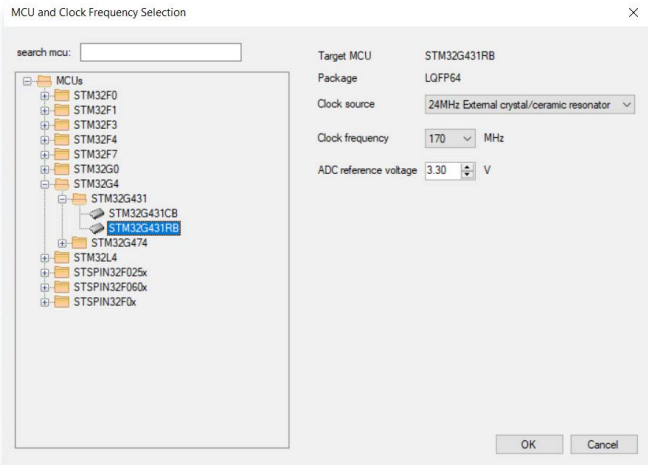
OK Cancel

Specific settings for Example MCSDK project for EPC9145 (continued)

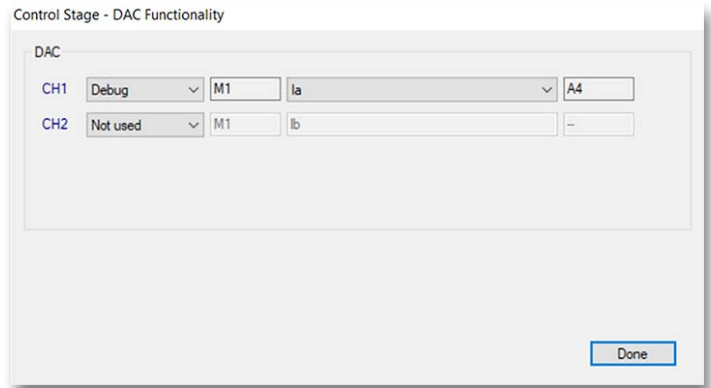
Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



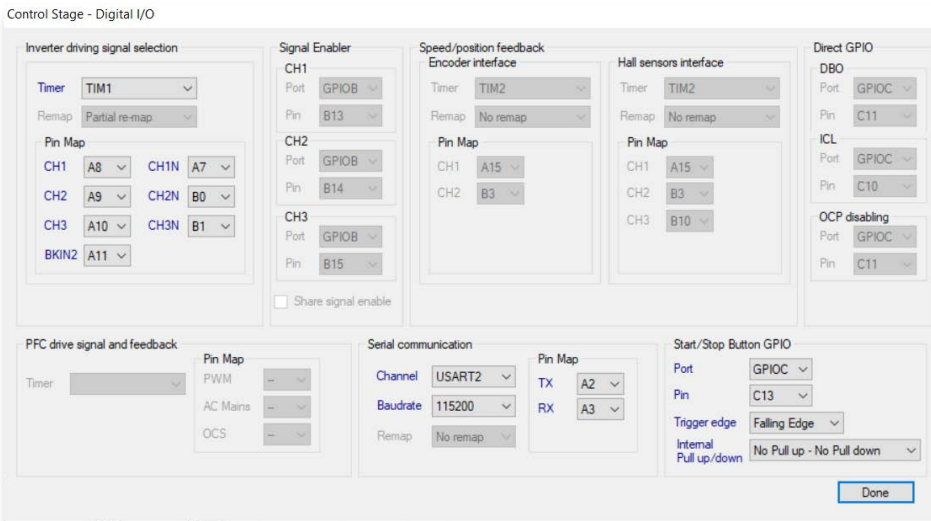
12



14

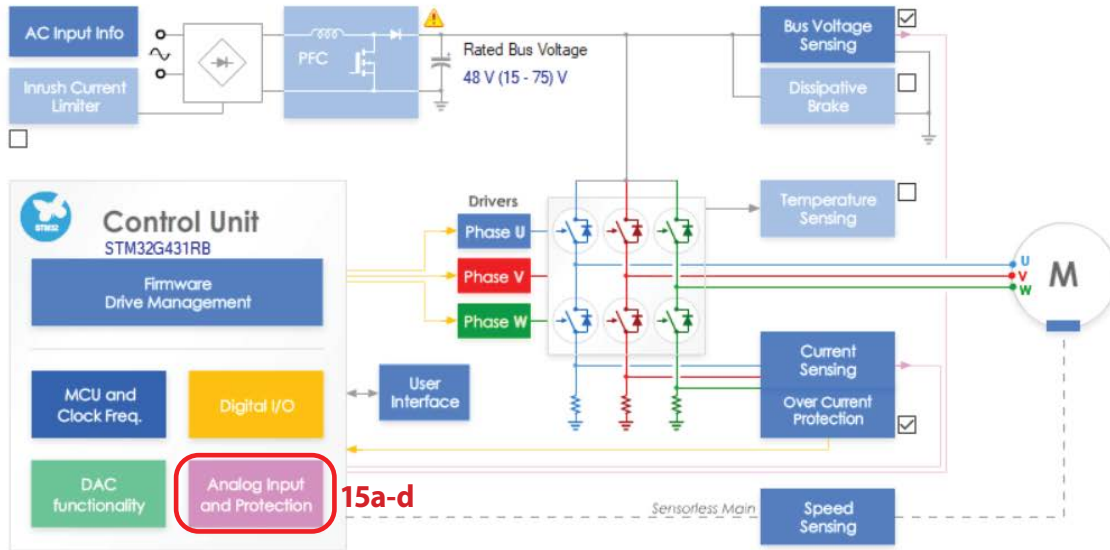


13



Specific settings for Example MCSDK project for EPC9145 (continued)

Motor: *Dummy Nema34* - Control Board: *NUCLEO-G431RB* - Power Board: *X-NUCLEO-IHM08M1*



15a

Control Stage - Analog Input and Protection

Phase current feedback **Bus voltage feedback** Temperature feedback PFC stage feedback

Current Sensing Topology: Embedded PGA External OPAMP

Over Current Protection Topology: Embedded HW OCP External Protection No protection

Sensing Setting: Sampling Time 6.5 ADC clk, 153 ns, Maximum modulation 93 %, Peripheral Selection ADC1/ADC2

Pin map: Ch phase U ADC12_IN1 (A0), Ch phase V ADC12_IN7 (C1), Ch phase W ADC12_IN6 (C0)

Sensing OPAMP: Peripheral selection OPAMP1/OPAMP2, OPAMP Gain Internal, Int gain type 2, Overall Network Gain 1.44, Vout (polarization) 1.710 V, T-rise 2550 ns

Pin map: Not inverting (A1, A7, D14), Inverting (A3, C5), Output (A2, A6, C8)

Protection Setting: Digital filter duration 8 clock, Inverting input Internal, Current threshold 5.000 Apk, Voltage Threshold 1.2 V

Pin map: CH_U Not Inv. A1 Inv. none COMP1 Output Ch_U A0, CH_V Not Inv. A7 Inv. COMP2 Output Ch_V A2, CH_W Not Inv. D14 Inv. COMP3 Output Ch_W C8

Done

15b

Control Stage - Analog Input and Protection

Phase current feedback **Bus voltage feedback** Temperature feedback PFC stage feedback

Sensing Setting: Sampling Time 47.5 ADC clk, Peripheral selection ADC2

Pin map: ADC Channel ADC12_IN2 (A1)

use Input Resistance (R3) Input Resistance 100.0 kOhm Bus Voltage Partitioning ...

15c

Control Stage - Analog Input and Protection

Phase current feedback Bus voltage feedback **Temperature feedback** PFC stage feedback

Sensing Setting: Sampling Time 47.5 ADC clk, Peripheral selection ADC2

Pin map: ADC Channel ADC12_IN8 (C2)

15d

Control Stage - Analog Input and Protection

Phase current feedback Bus voltage feedback Temperature feedback **PFC stage feedback**

Current sensing Setting: Sampling Time 1.5 ADC clk, Peripheral selection ADC2

Pin map: ADC Channel

AC voltage sensing Setting: Sampling Time 1.5 ADC clk, PFC AC/Volt Sens

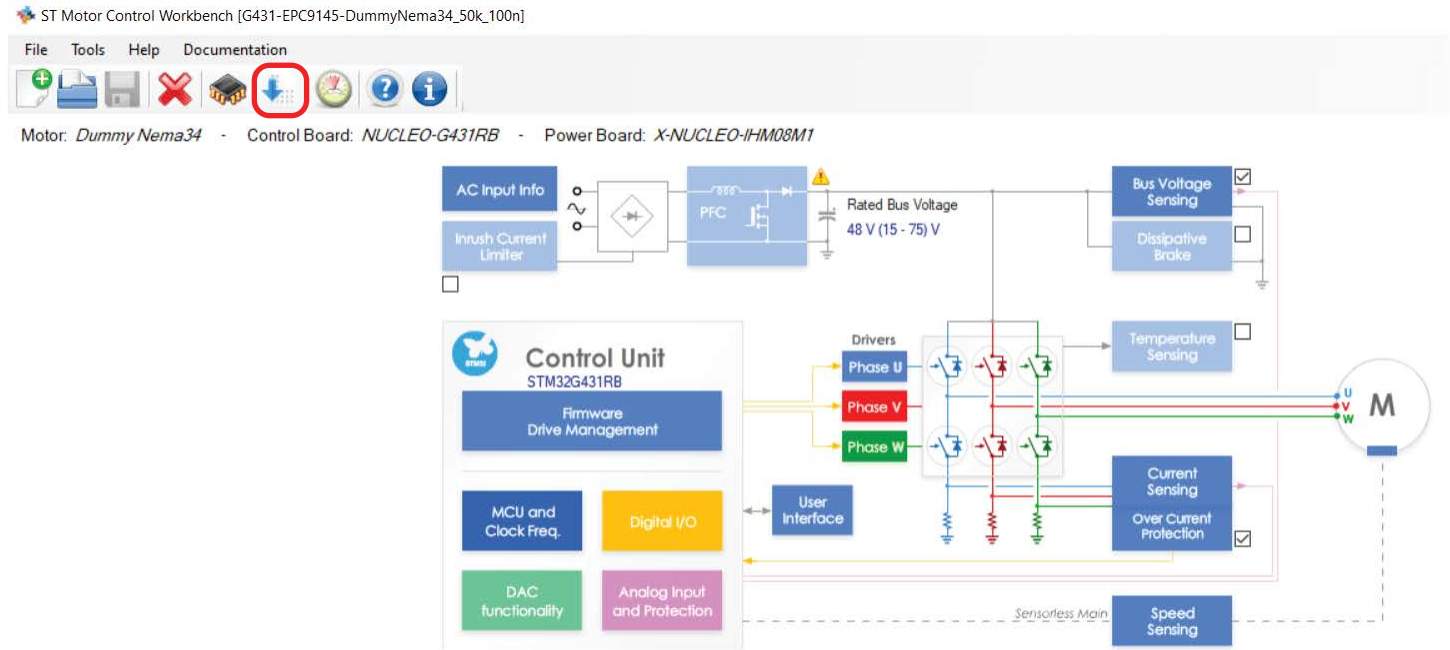
Pin map: ADC Channel ADC12_IN3 (A3)

Commissioning a Motor for use with EPC motor drives operating ST Motor Control

LAUNCH THE CODE GENERATOR

Click on the **Code Generator** button. This will invoke the **STM32CubeMX** program.

Note: depending on the installation, you may have different versions.



STM32CUBEMX – AUTOMATIC CODE GENERATION

1. Select the Recommended firmware version (A)

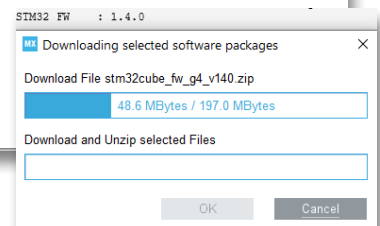
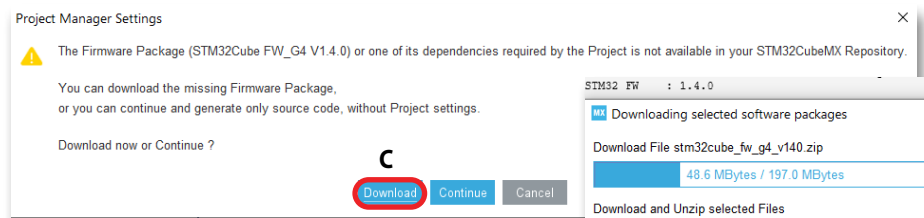
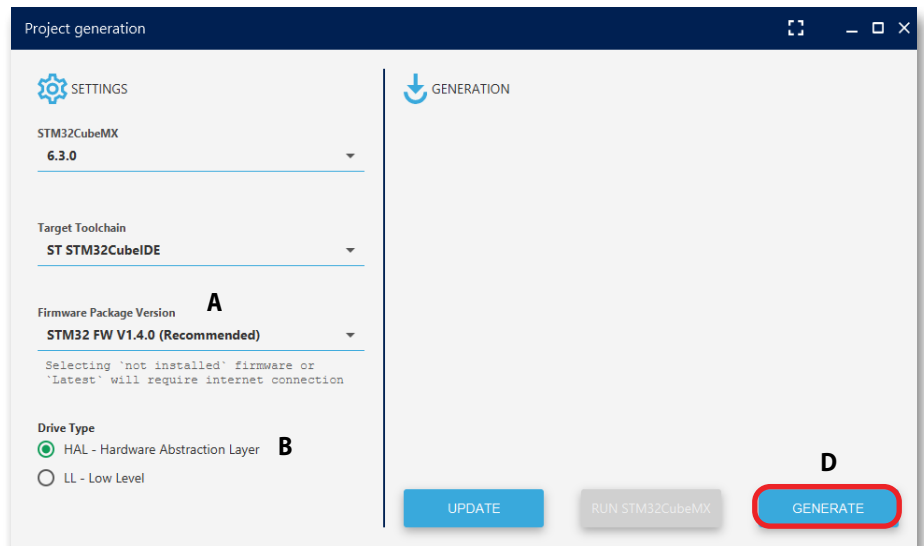
2. Select HAL drive type (B)

Note: At the first-time use, the firmware will not be available, it will be downloaded automatically. Confirm that you want to continue the download (C). This may generate Windows firewall messages. It may also be necessary to disable anti-virus software.

3. Press **Generate Button** (D)

- The first time it may happen that an error message is generated because the firmware is not yet installed completely. In this case, close the window and restart the process.
- At the end of code generation, you should not get any error message, if it happens, repeat the process. If it continues, contact a ST local FAE.

4. At the end of the code generation, open the folder containing the code and **close the STM32CubeMX window.**

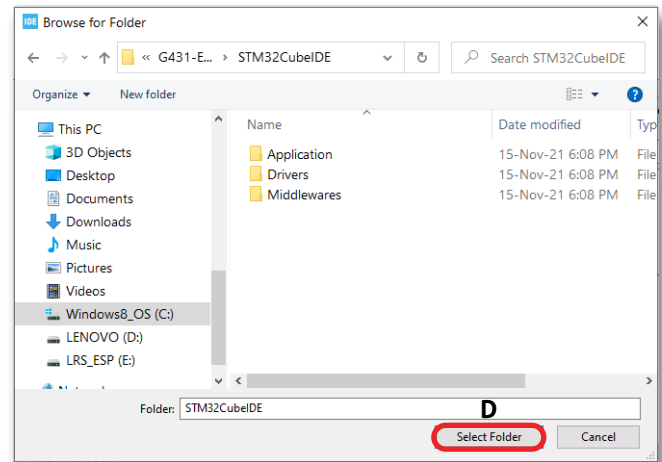
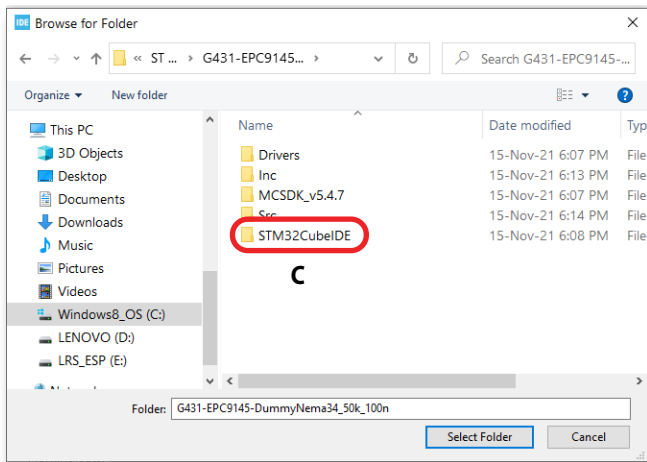
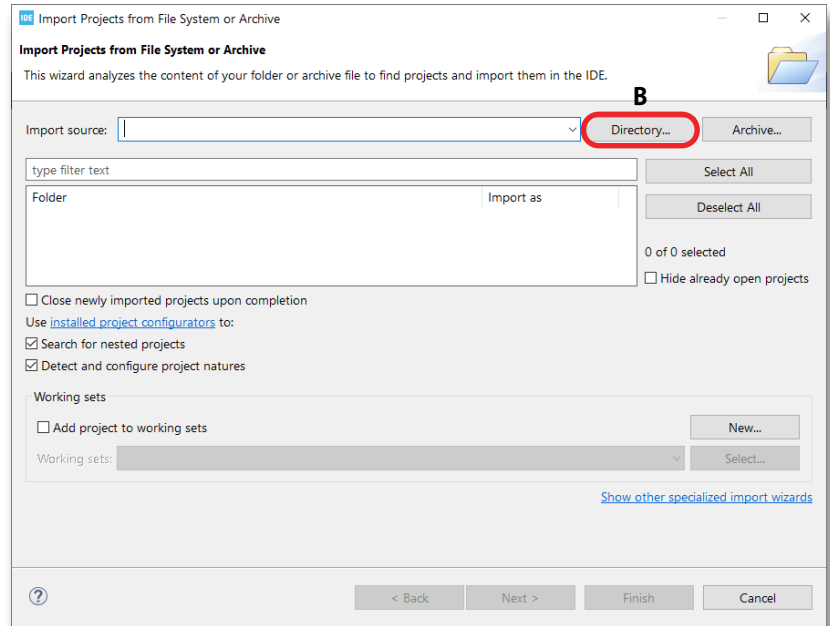
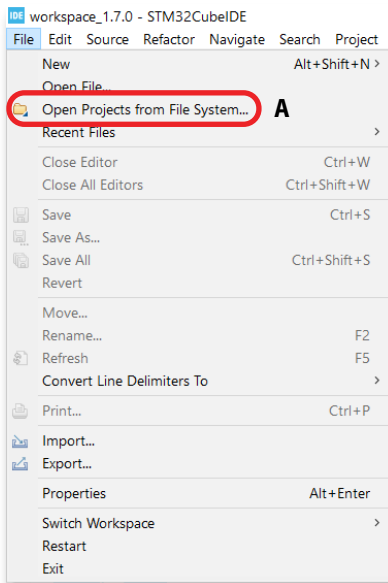


STM32CUBEIDE – IMPORT THE PROJECT

1. Open the **STM32CubeIDE** program from Windows Start Menu
2. Accept the proposed location for the Workspace and select the File menu
3. Under File menu choose **Open Projects from File System (A)**
4. Click on **Directory (B)**
5. Navigate to the generated code directory
6. Choose the **STM32CubeIDE sub-directory (C)** within the generated code directory (note: this directory was generated by STM32CubeMX program)
7. Press **Select Folder (D)**
8. Press **Finish button** on the Import Project window

Note: the project must be imported only for the first time. All subsequent code re-generations on the same project will not require to import the folder again.

Note: it is better to have separate projects for different motors



STM32CUBEIDE - BUILD THE CODE

Note: When you open STM32CubeIDE the first time you may land into the info window. In that case close the info window.

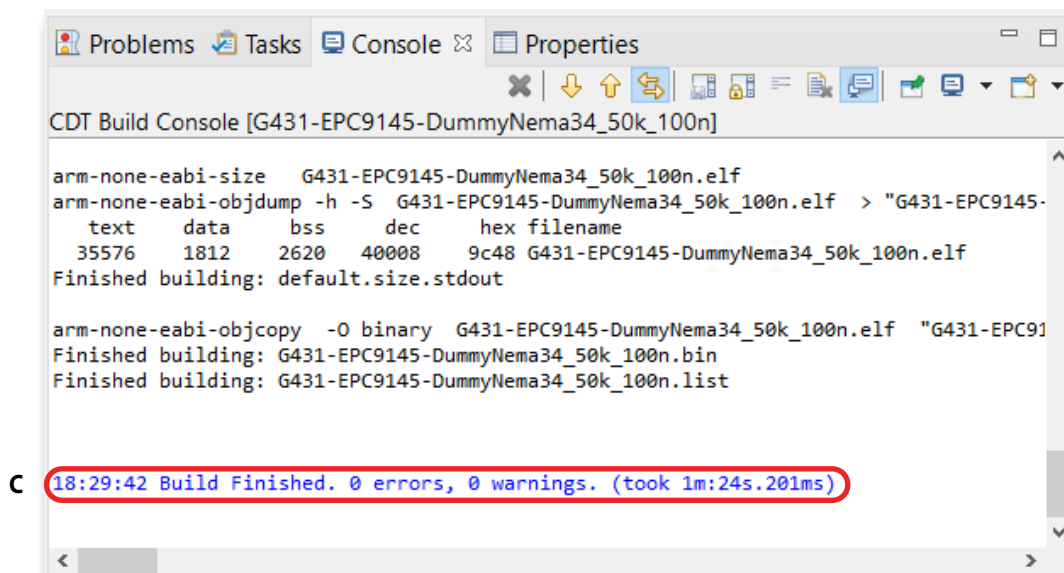
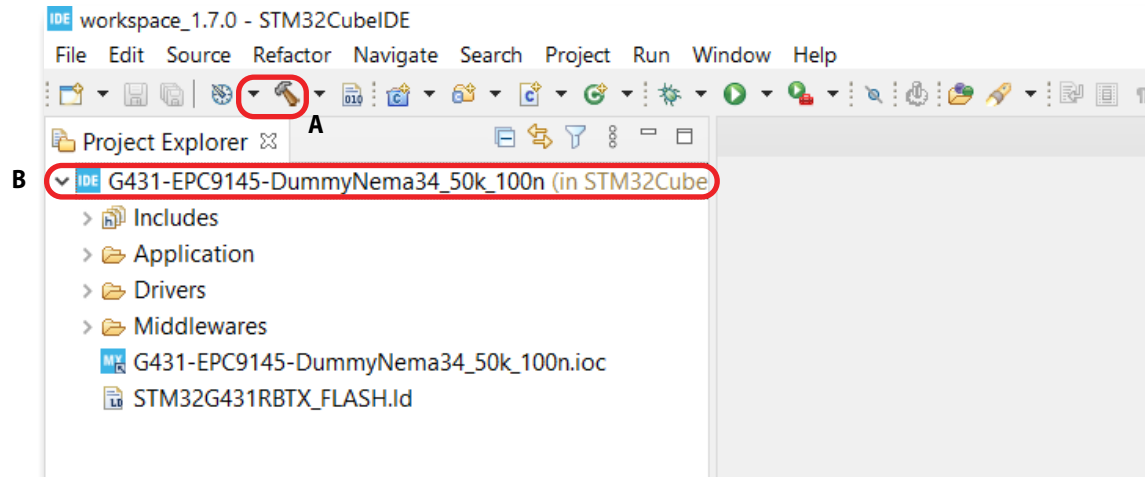
If Project Explorer window is not visible, go to the menu **Window>>Show View>>Project Explorer**

1. **Select the project (A)** on STM32CubeIDE program. In case of many projects, right click the one you want to select and set it as main project.
2. Press the **Hammer icon (B)**, it will launch the compiler and the builder.

At the end of the build process, **you must get a clean compile message in the CDT Build Console (C)**

Note: if you get any compile error, re-start the entire project from the ST Motor Control Workbench program.

Note: if you still get errors, contact a ST local FAE

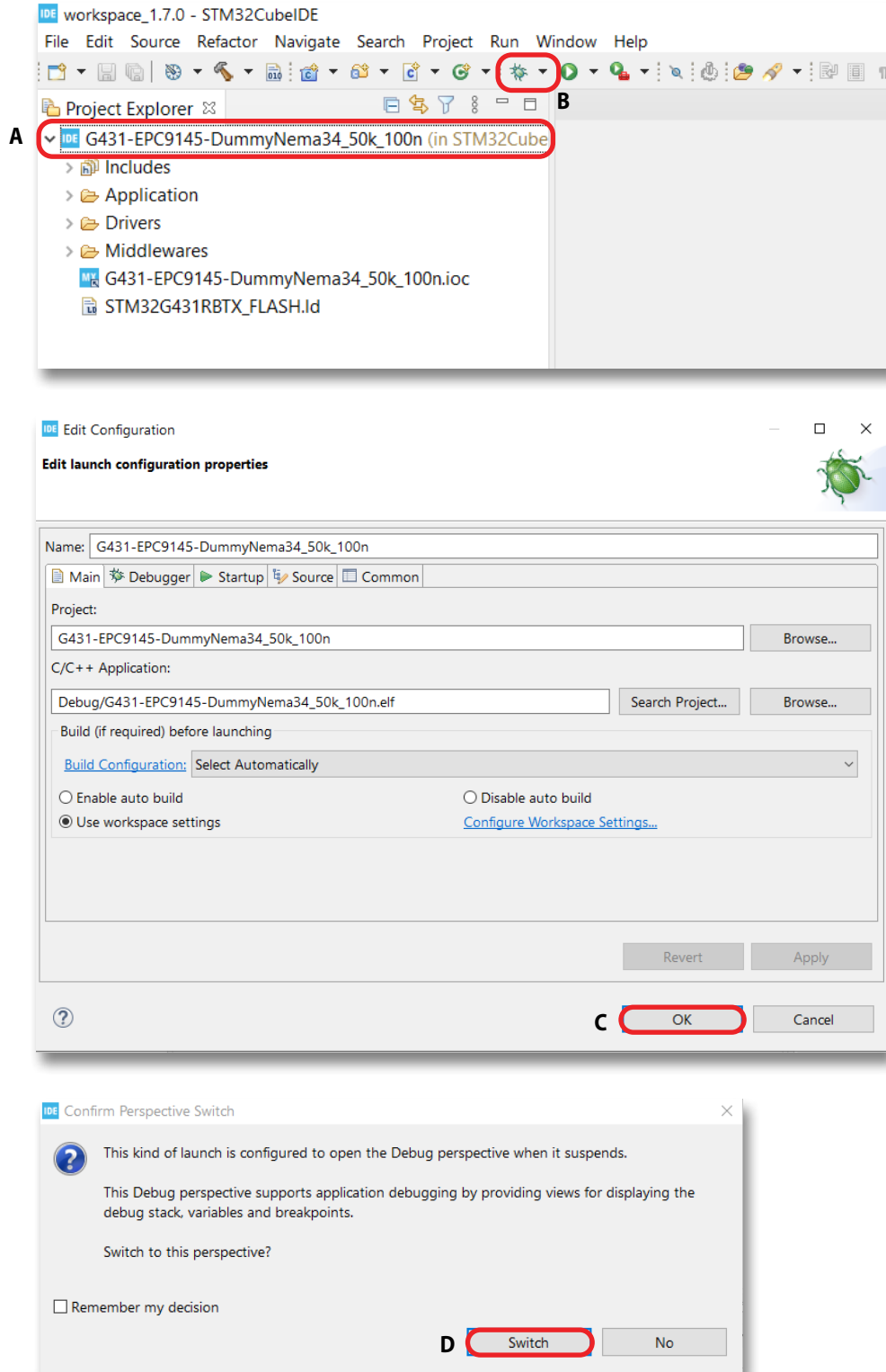


STM32CUBEIDE - FLASH AND DEBUG

Select the project (A) on STM32CubeIDE program. In case of many projects, right click the one you want to select and set it as main project. Connect the Nucleo board with an USB micro cable to an USB port on the PC. Note: debugger is onboard of the ST Nucleo board.

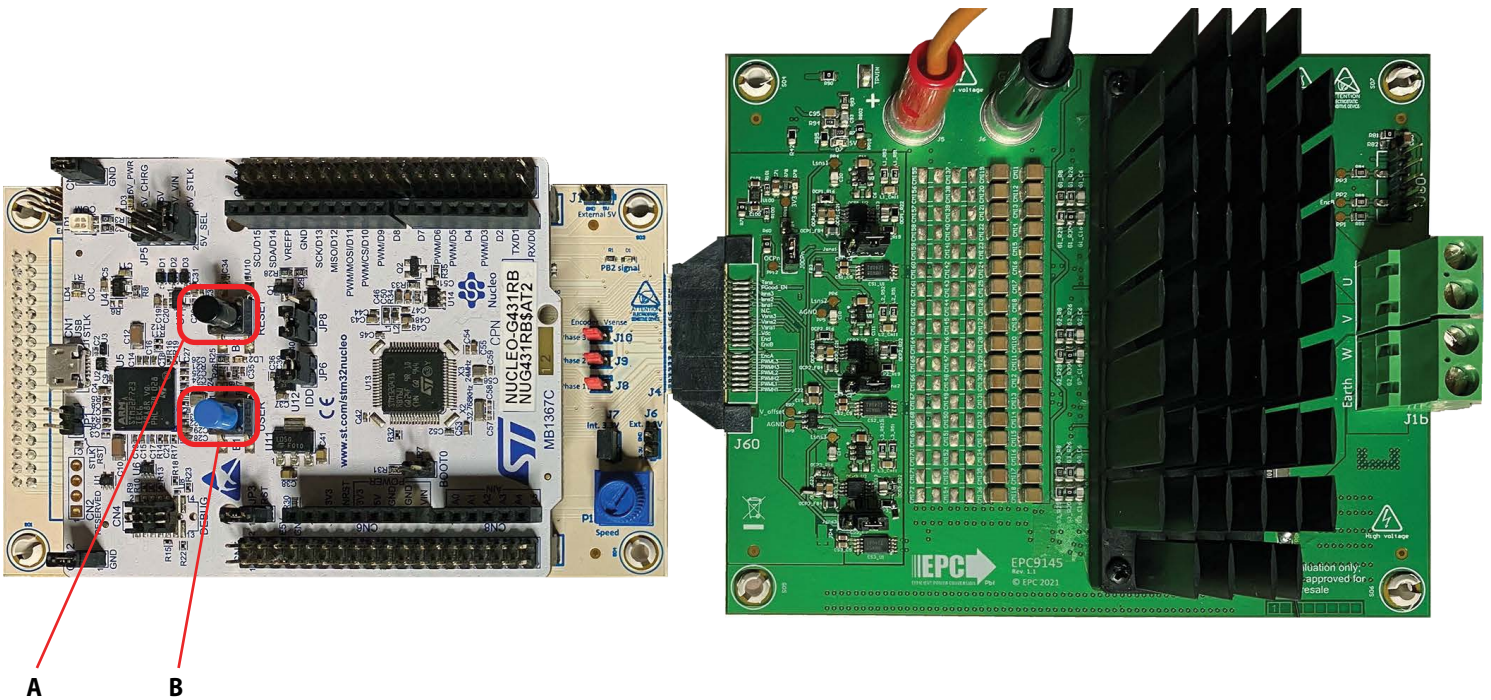
1. Press the **Bug icon (B)**, it will launch first the flash downloader process and then the debugger.
2. The first time you will get two dialog windows as the one shown here, **press OK (C)** and then **Switch (D)**

When the download to the Flash is finished, and the controller is in Halt mode (for debug), close STM32CubeIDE and detach the controller from USB port.



OPERATING THE MOTOR DRIVE SYSTEM DISCONNECTED FROM PC

1. Connect the EPC9147C to a compatible inverter board; e.g. EPC9145
2. Connect the motor to the inverter board. Follow QSG instructions.
3. With power **OFF**, connect the power supply to the inverter board. Make sure the jumpers are set as in the picture.
4. Set the power supply to the correct operating voltage for the inverter board. Make sure the current limit setting is sufficient to operate the motor drive system. For EPC9145 $V_{sup} = 48\text{ V}$ and $I_{lim} \geq 2.5\text{ A}$
5. Power on
6. **Press the black button (A)** on Nucleo board
7. **Press the blue button (B)** on Nucleo board to start and stop the motor



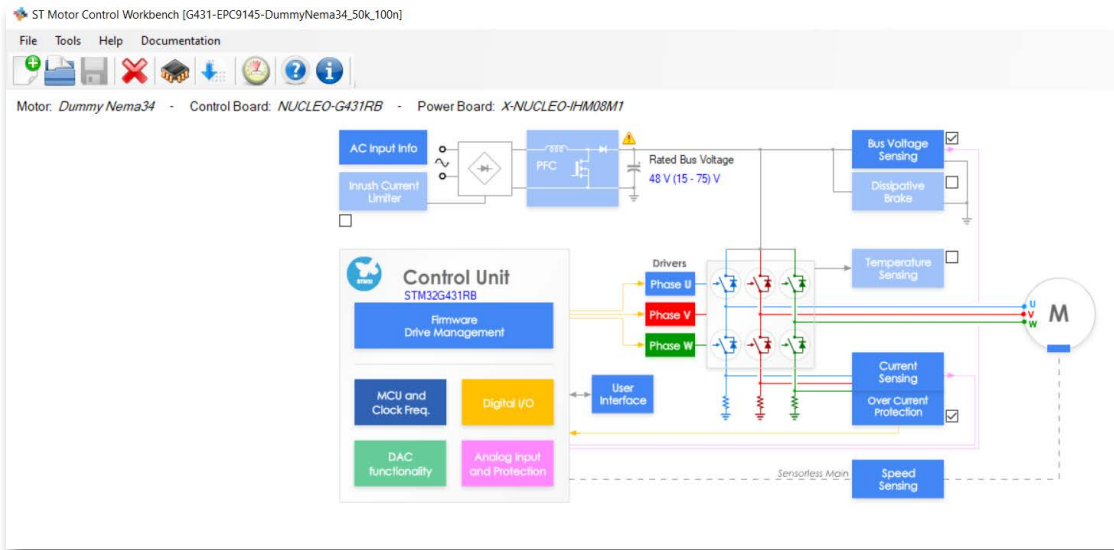
OPERATING THE MOTOR DRIVE SYSTEM CONNECTED TO THE PC AND USING ST MOTOR CONTROL WORKBENCH MONITOR OPTION

Launch the Monitor

When the ST Nucleo board is properly programmed and connected via the EPC9147C to the proper power board, it is also possible to use the ST Motor Control Workbench GUI to change the speed and the direction of the motor.

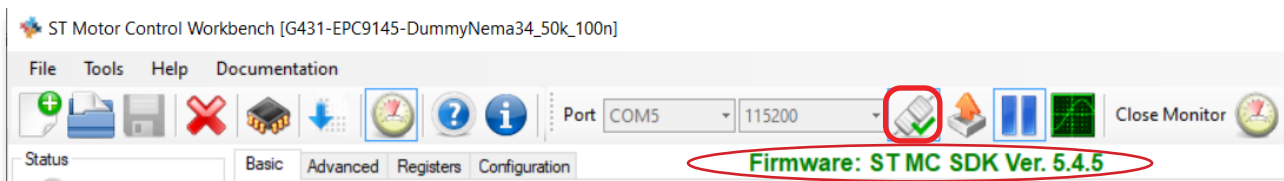
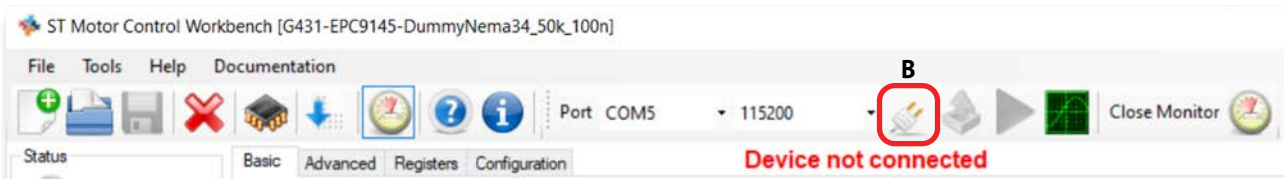
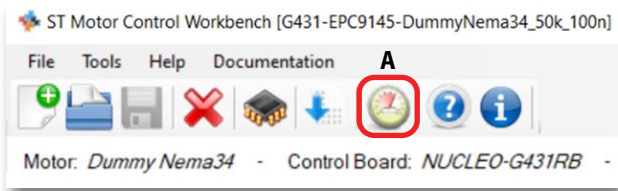
Open the ST Motor Control Workbench and load the proper .stmcx file that is relevant to the project you are working at (e.g. G431-EPC9145-DummyNema34_50k_100n.stmcx).

Note: Depending on the installation, you may have different versions.



Monitor

1. Click on the **Open Monitor Button (A)**.
2. Connect the USB cable to the PC and power up the 48 V to the power board. Click on the **Connect button (B)**.
3. Wait for the **successful connection message (C)**.



C

Commissioning a Motor for use with EPC motor drives operating ST Motor Control

Press Fault Ack button (A) if any fault was detected. Then **press Start Motor button (B)**. Motor should spin. It is now possible to move the graphic potentiometer on the GUI to change the speed of the motor and to change the motor direction. Refer to ST user guide manual for more details on how to work with the ST Motor Control Workbench for further customization.

Refer to [ST Motor Control Workbench Guide](#) for further information

Note: ST Motor Profiler does NOT work with EPC power boards. If you try, you may damage the EPC power board.

The screenshot displays the ST Motor Control Workbench interface. The main window shows several gauges: Bus Voltage (Volt) at 47, Motor Power (W) at 0, Measured speed (rpm) at 0, and Final ramp speed (rpm) at 1278. A thermometer gauge shows Heatsink temperature at 25. The interface includes a menu bar (File, Tools, Help, Documentation), a toolbar, and a status bar. The 'Generic' panel on the right contains control buttons: 'Start Motor' (labeled B), 'Stop Motor', 'Stop Ramp', 'Fault Ack' (labeled A), 'Encoder Align', and 'All motors'. The 'PFC' panel includes 'PFC Enable', 'PFC Disable', and 'PFC Fault Ack'. The bottom panel shows a log of messages:

| Time | Motor | Id | Message |
|----------|-------|----|-----------------------------|
| 11:29:20 | | | Command "Fault Ack" done! |
| 11:29:21 | | | Command "Start Motor" done! |
| 11:29:34 | | | Command "Stop Motor" done! |

For More Information:

Please contact info@epc-co.com
or your local sales representative

Visit our website:
www.epc-co.com

Sign-up to receive
EPC updates at
bit.ly/EPCupdates
or text "EPC" to 22828



EPC Products are distributed through Digi-Key.
www.digikey.com

Demonstration Board Notification

The EPC9147A board is intended for product evaluation purposes only. It is not intended for commercial use nor is it FCC approved for resale. Replace components on the Evaluation Board only with those parts shown on the parts list (or Bill of Materials) in the Quick Start Guide. Contact an authorized EPC representative with any questions. This board is intended to be used by certified professionals, in a lab environment, following proper safety procedures. Use at your own risk.

As an evaluation tool, this board is not designed for compliance with the European Union directive on electromagnetic compatibility or any other such directives or regulations. As board builds are at times subject to product availability, it is possible that boards may contain components or assembly materials that are not RoHS compliant. Efficient Power Conversion Corporation (EPC) makes no guarantee that the purchased board is 100% RoHS compliant.

The Evaluation board (or kit) is for demonstration purposes only and neither the Board nor this Quick Start Guide constitute a sales contract or create any kind of warranty, whether express or implied, as to the applications or products involved.

Disclaimer: EPC reserves the right at any time, without notice, to make changes to any products described herein to improve reliability, function, or design. EPC does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, or other intellectual property whatsoever, nor the rights of others.