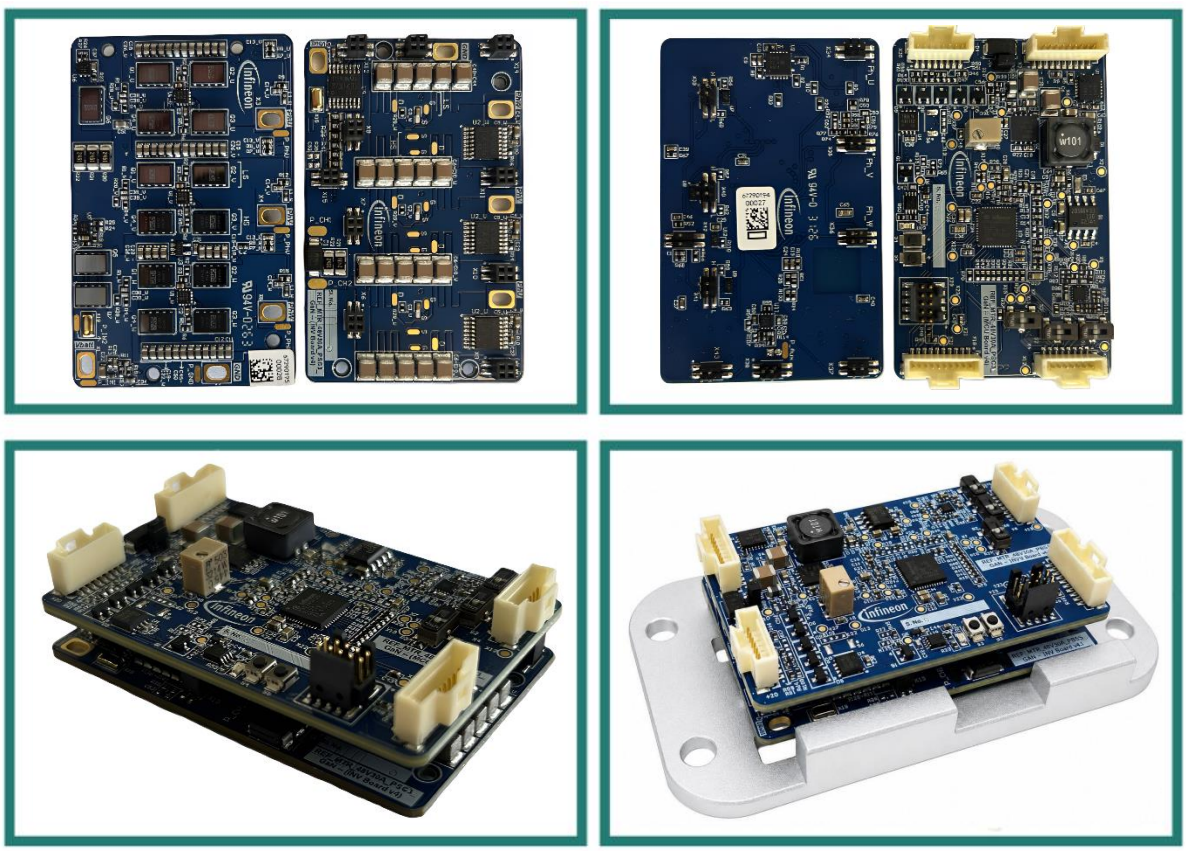
QUICK START GUIDE

# 48V GaN based Motor Drive Reference Design

## Kit contents

1. REF_MTR_48V30A_PSC3_GaN (MCU board)
2. REF_MTR_48V30A_PSC3_GaN (POWER board)

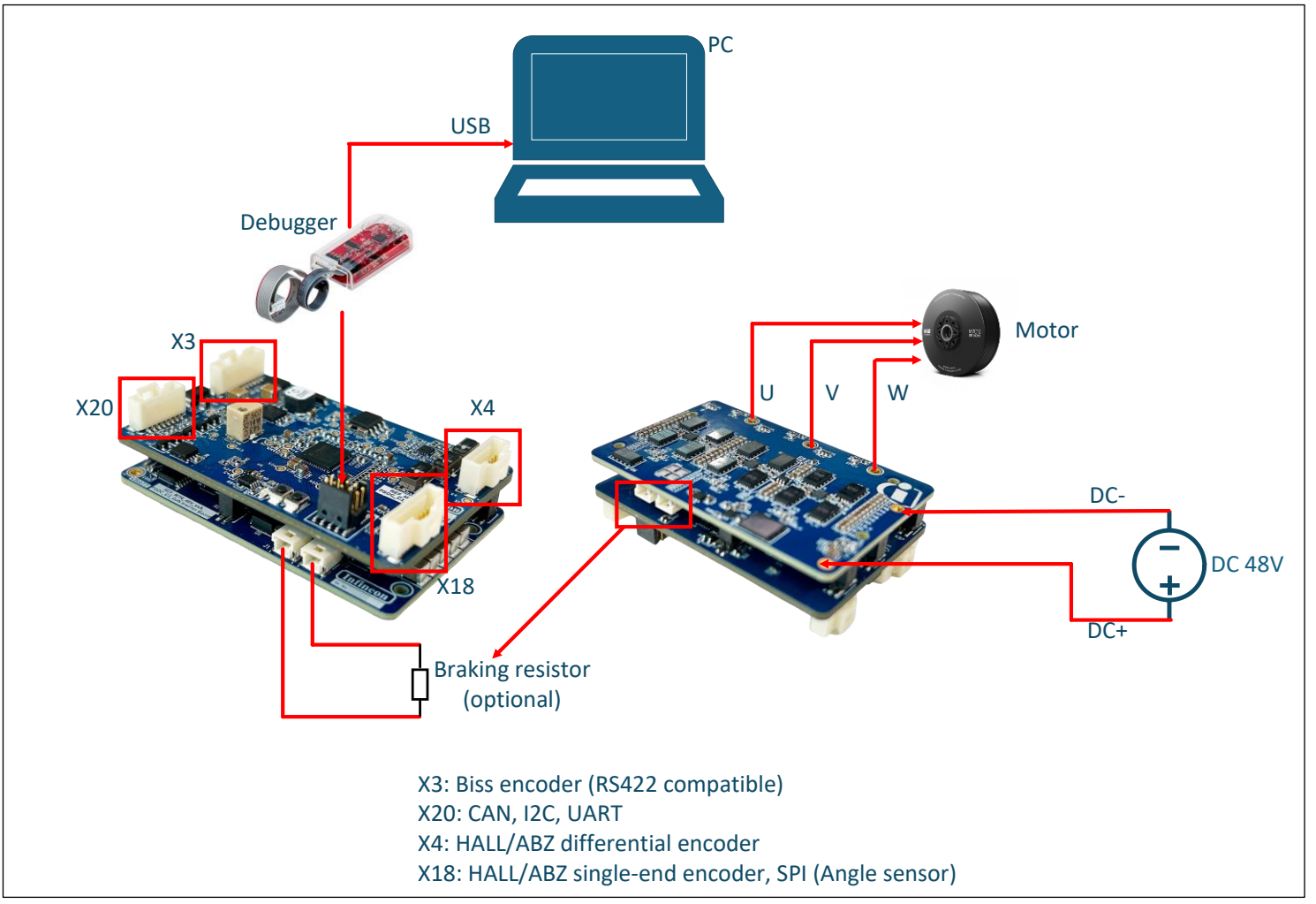


https://www.infineon.com/evaluation-board/REF-MTR-48V30A-PSC3-GAN

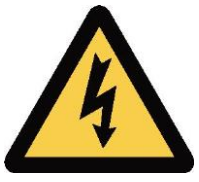
**Hardware setup**

Ensure that you have prepared the following before running a motor with this reference board.

1. DC power supply. 48V DC output and power range should cover the motor to be run.
2. J-link debugger or KIT_XMC_LINK_SEGGER_V1 debugger. For flashing firmware and running a motor with GUI.
3. 48V 3-phase BLDC/PMSM motor
4. PC with ModusToolbox™ and Motor Suite installed.




1. Stack the MCU board and Power board.
2. Connect DC power supply to DC+, DC- of the power board.
3. Connect motor via UVW motor connectors.
4. By default, set DC power supply to output 48V to drive a 48V PMSM/BLDC motor.



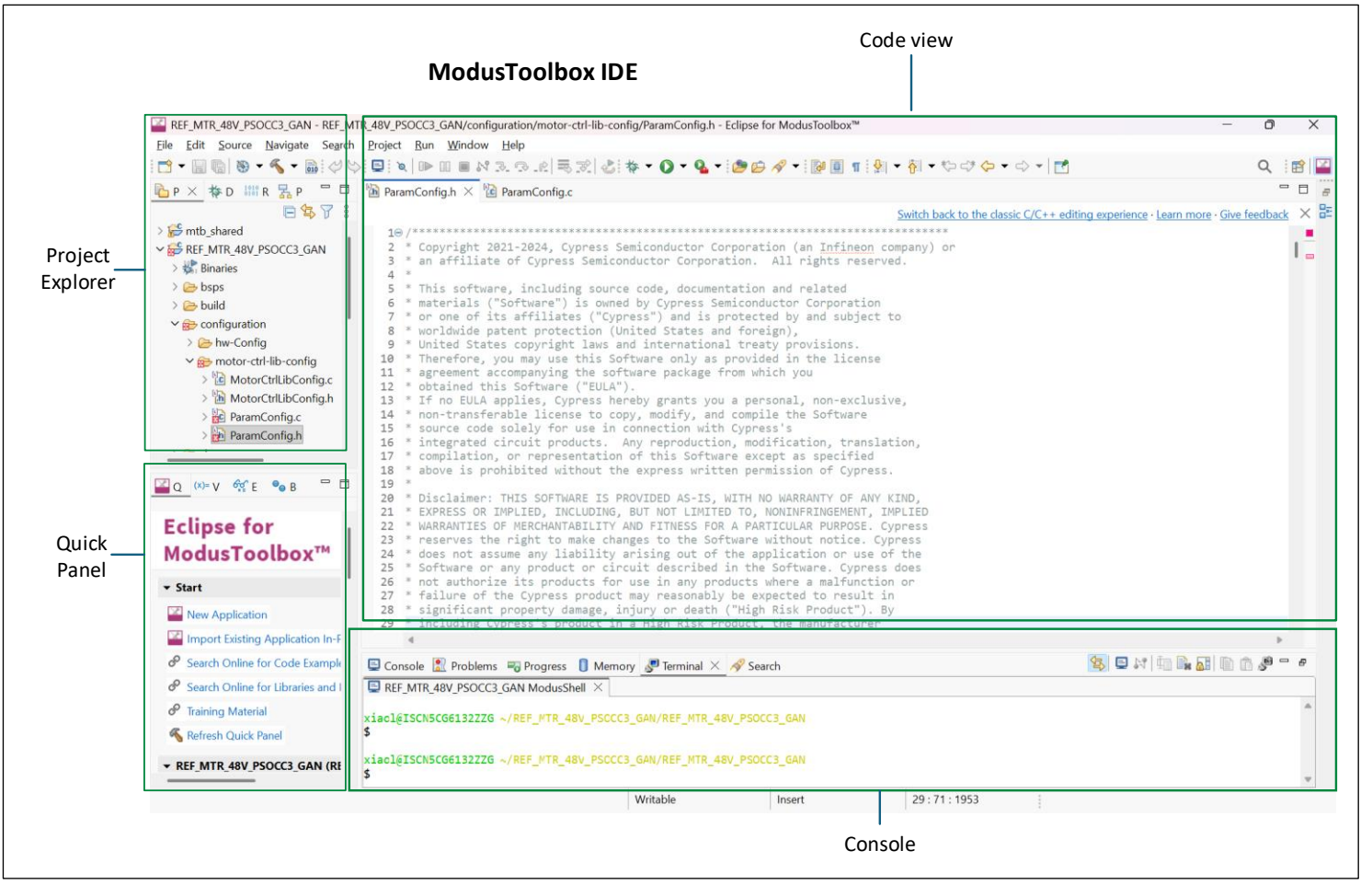
**Note:** Remove or disconnect DC power from the Power Board before you disconnect or reconnect wires or perform maintenance work. Wait for a while after removing DC power to discharge the DC link capacitors. Do not attempt to connect debugger or interface cables to MCU Board until the DC link capacitors have discharged to zero. Failure to do so may result in board damage or personal injury.

**Prerequisites**

The followings software tools are required on your PC:

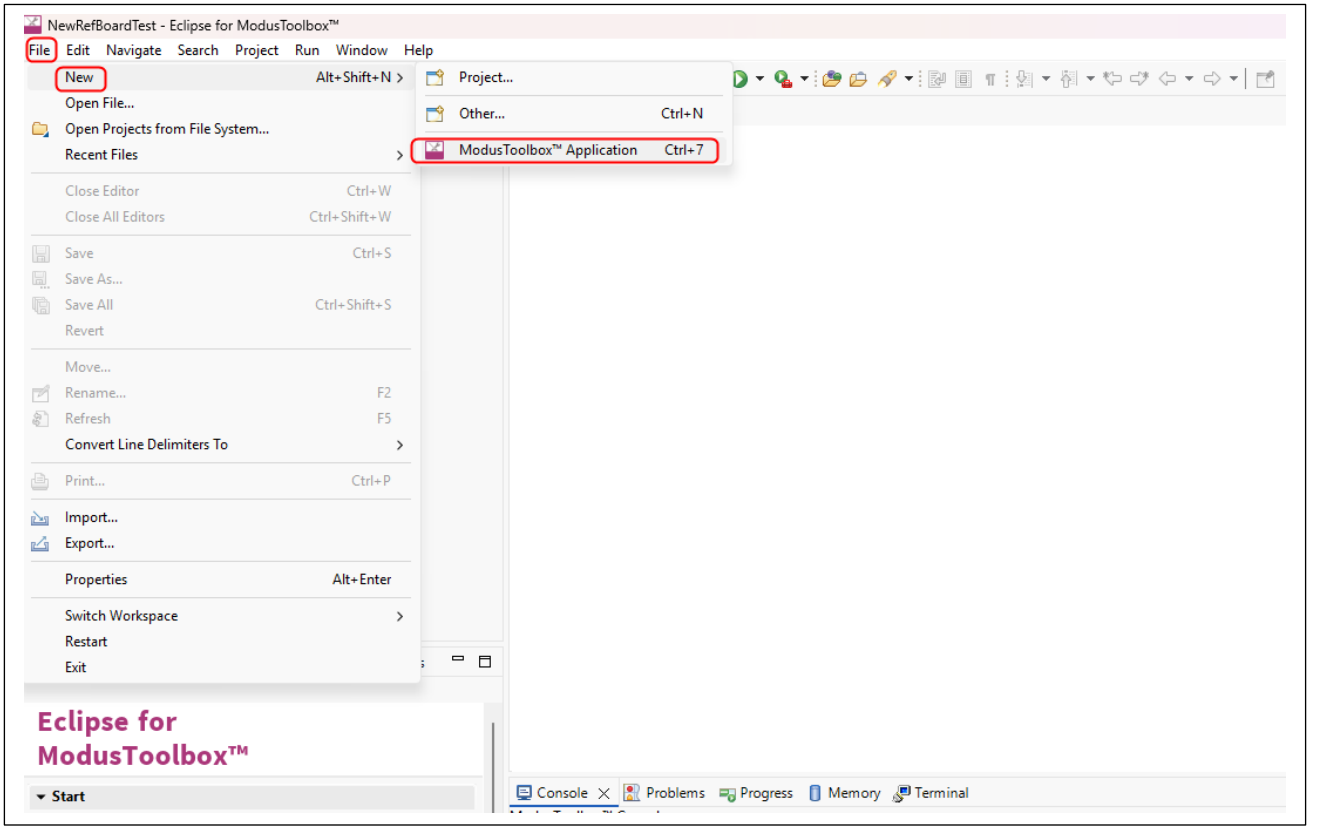
1. ModusToolbox™ Software/IDE is required to compile and debug the application project. Link to download: https://www.infineon.com/design-resources/development-tools/sdk/modustoolbox-software
2. ModusToolbox™ Motor Suite is needed for loading firmware .hex/.elf files to the target device and to provide a user interface for user to run motor and monitor real time parameters.

The following sections provide the steps to use the required software tools.

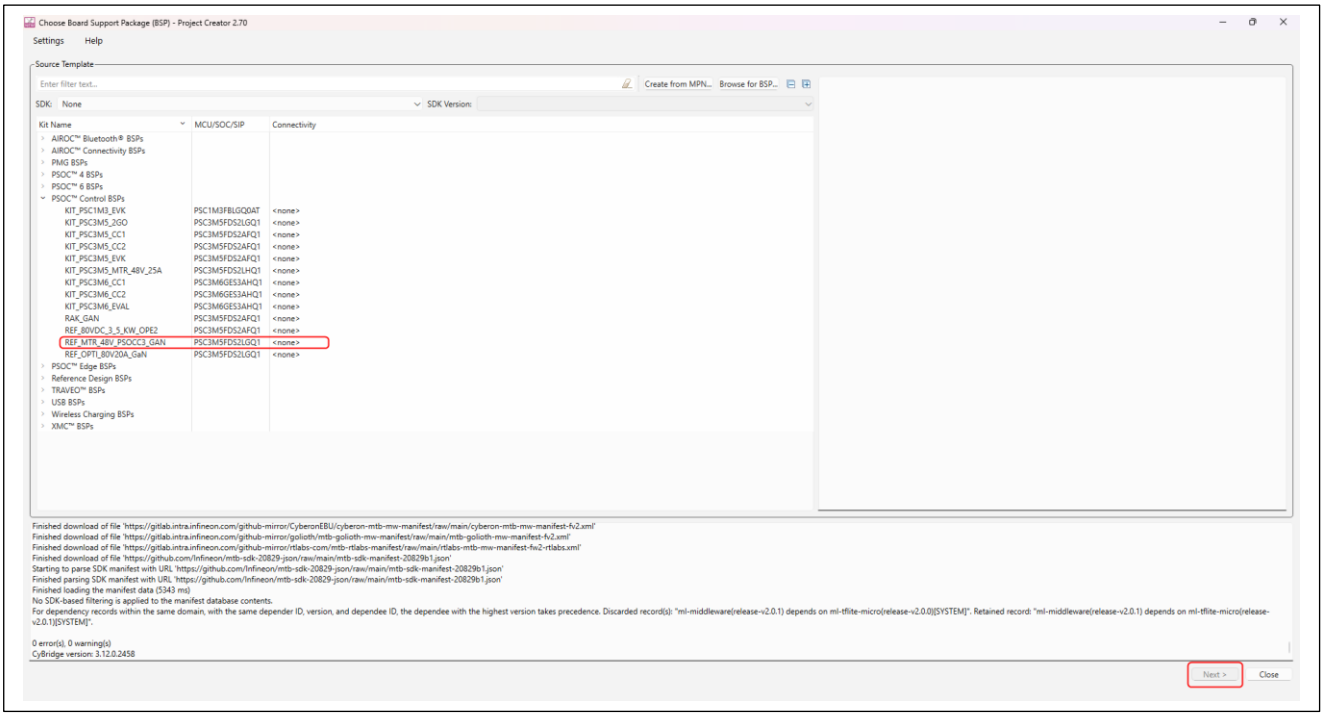


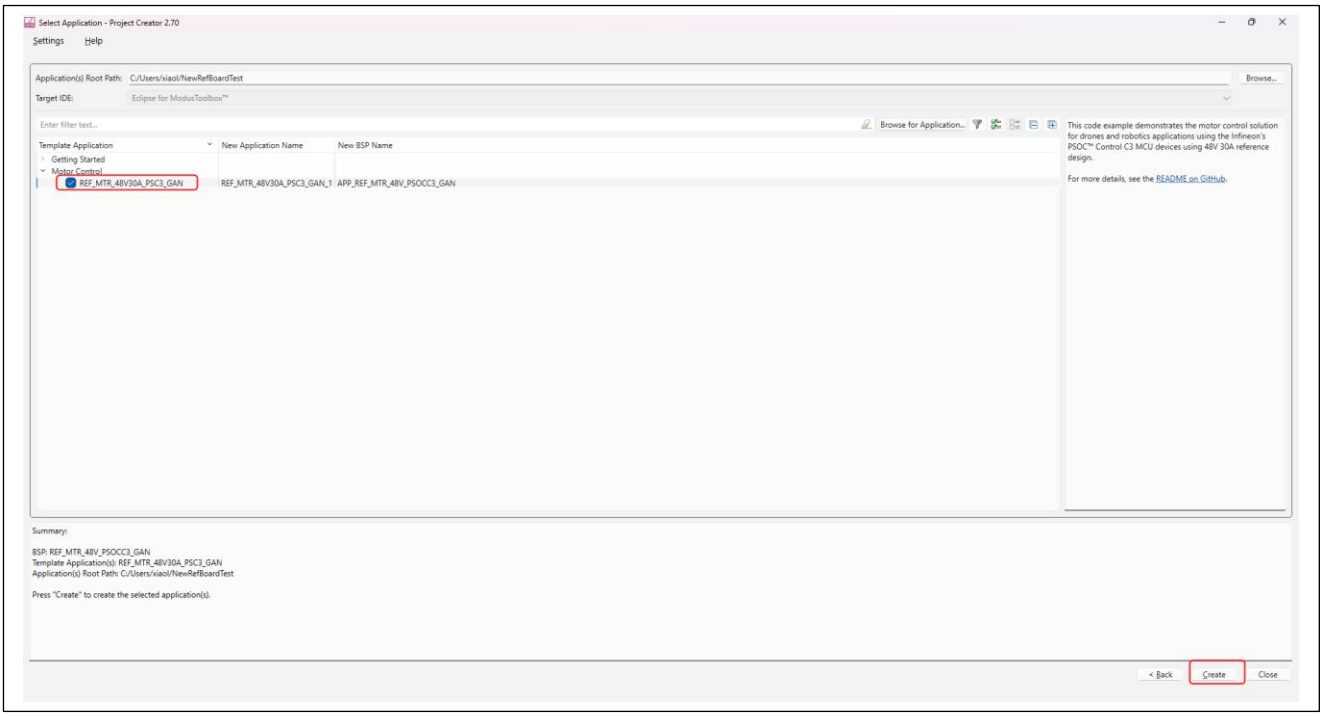

**Firmware package**

1. To get source code of this reference board REF_MTR_48V30A_PSC3_GaN, you can download the firmware package from ModusToolbox™ by creating "New application".

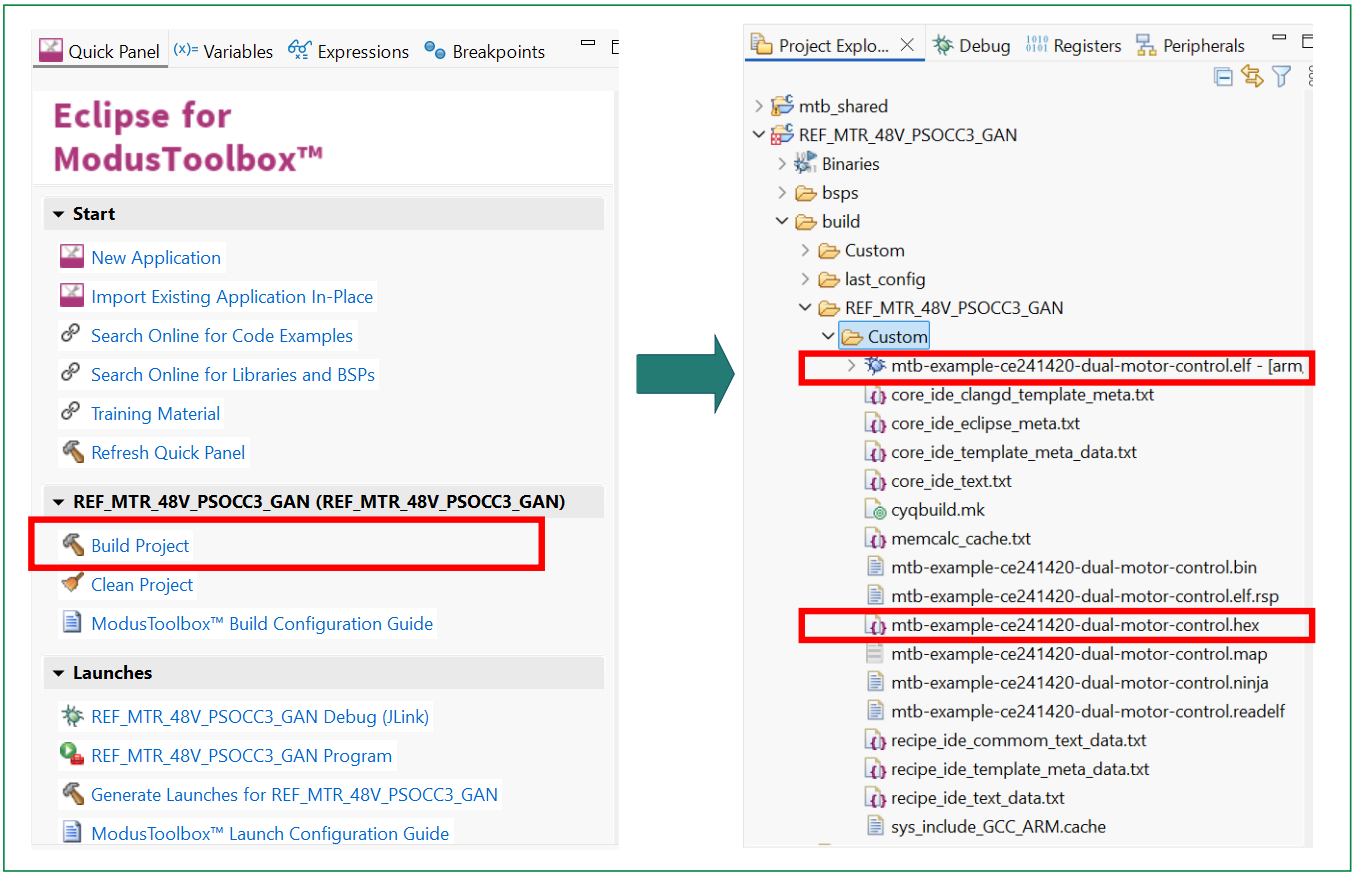



Choose the board BSP.





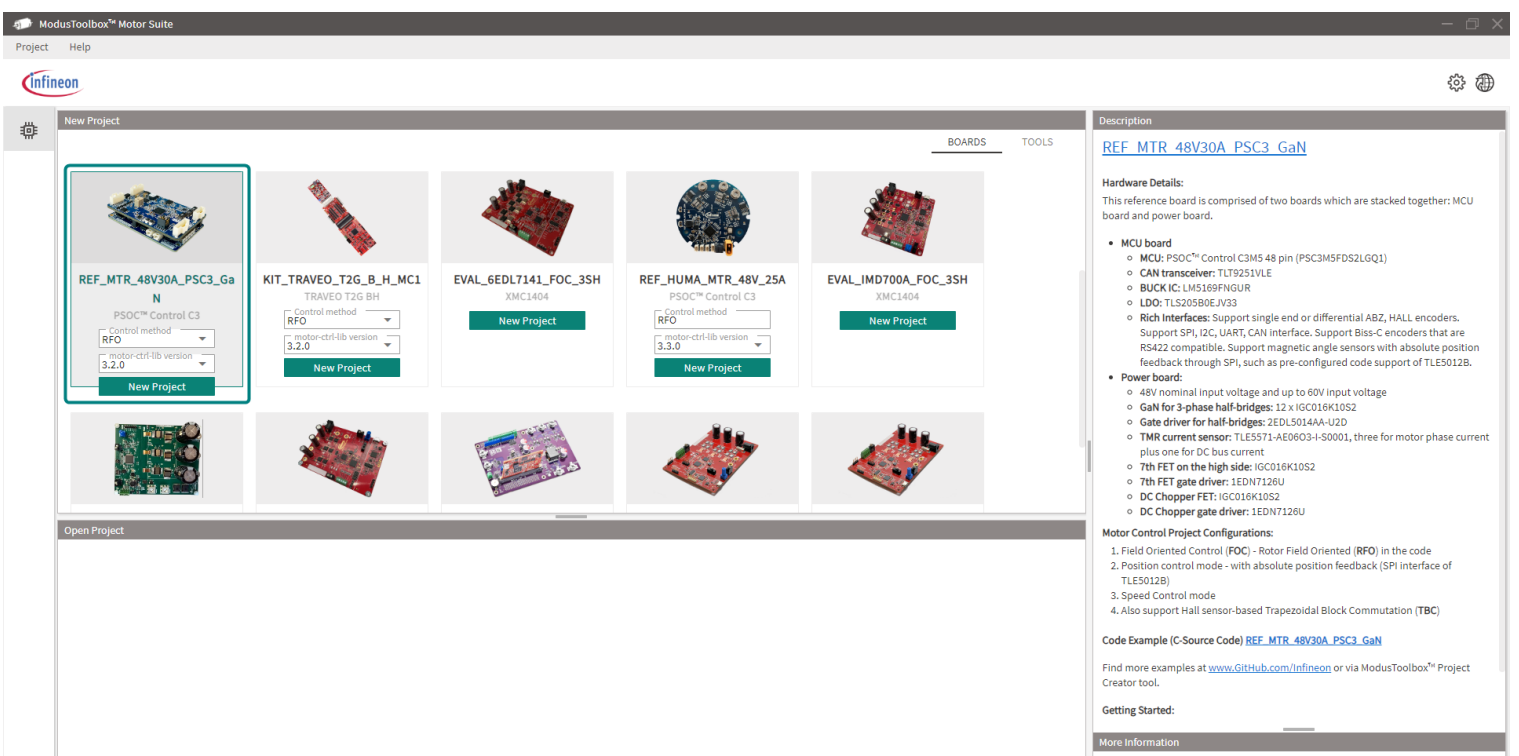
2. The created project folder and its submodule are shown in the Project Explorer Panel.
3. Build and compile the project. The result of the project built is shown in **build folder** as .elf and .hex file.




**ModusToolbox™ Motor Suite**

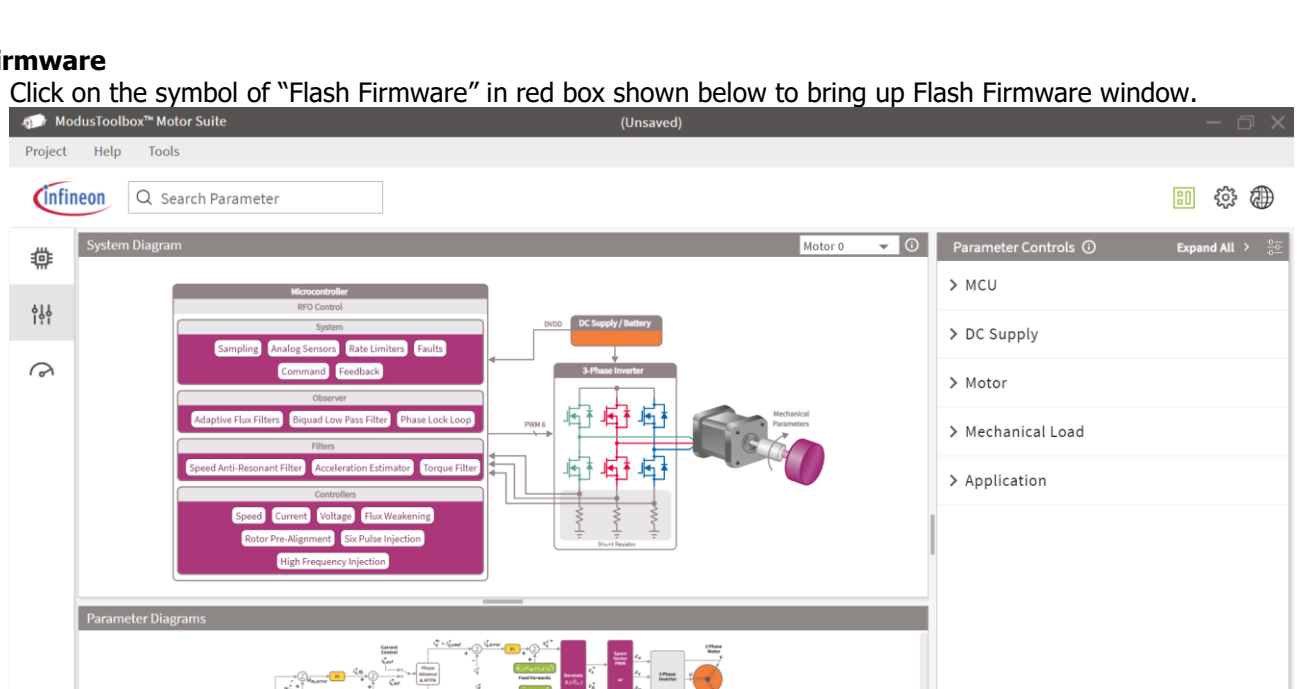
1. Search for installed ModusToolbox™ Motor Suite_v2.x.x in Windows Search panel and open the software.
2. Click drop down menu under REF_MTR_48V30A_PSC3_GaN, and select RFO as Control method.
3. Click on **New Project**.

If the USB connection is established between the debugger and user's laptop, the status at the bottom right corner will show Connected.

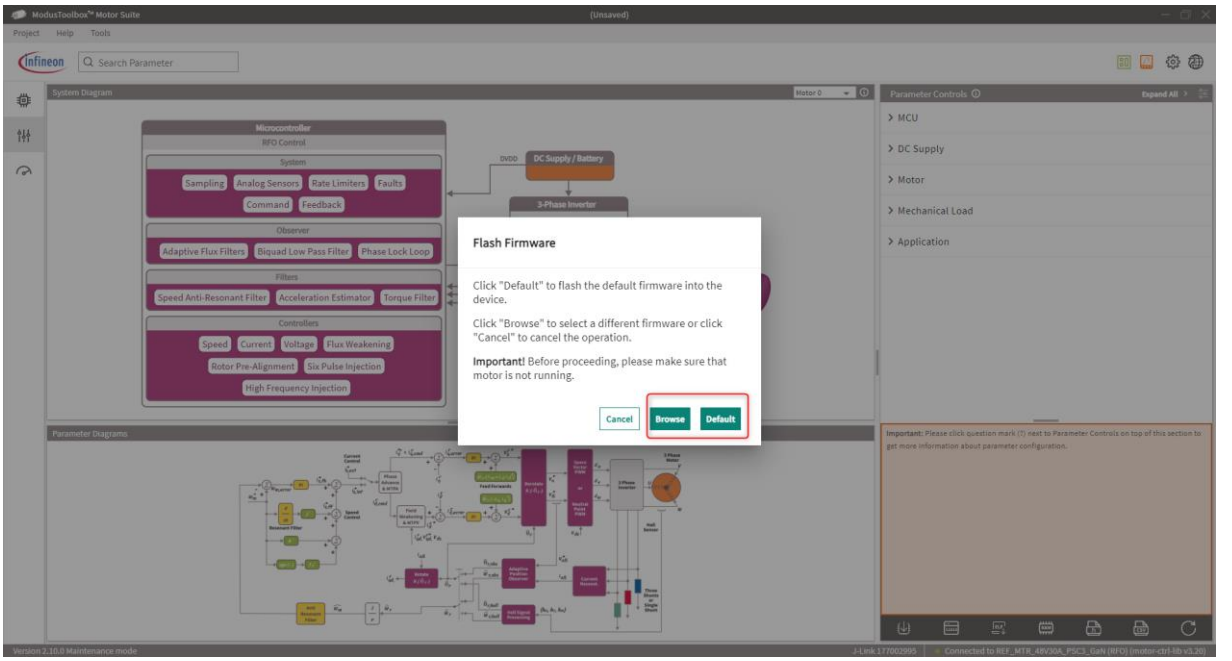



## Flash Firmware

1. Click on the symbol of "Flash Firmware" in red box shown below to bring up Flash Firmware window.

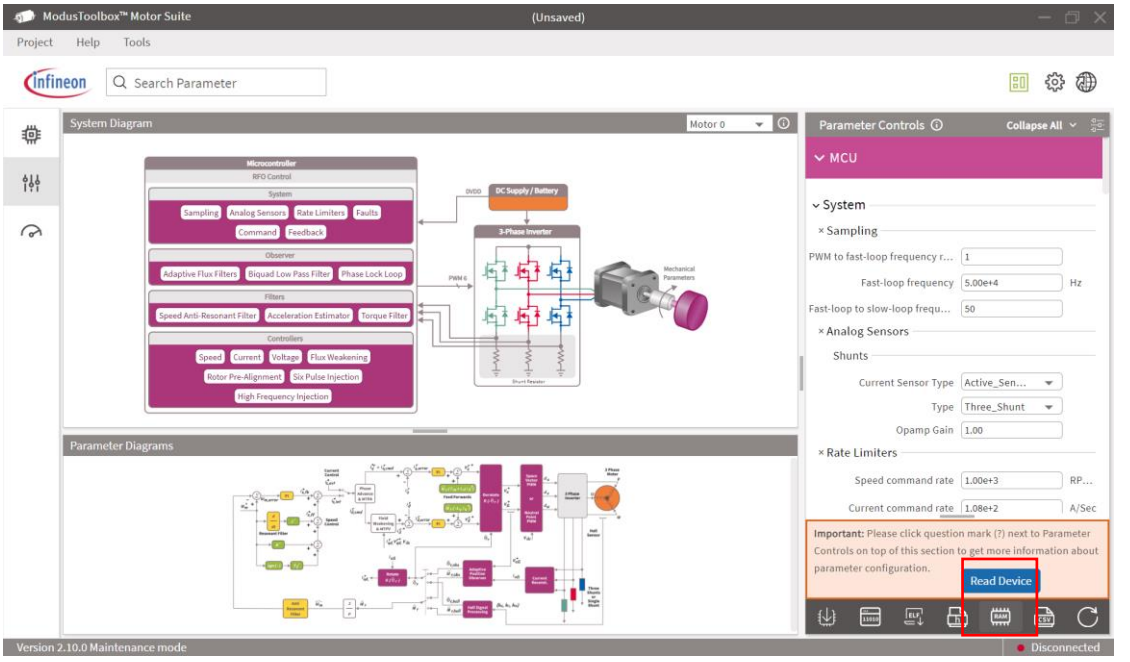


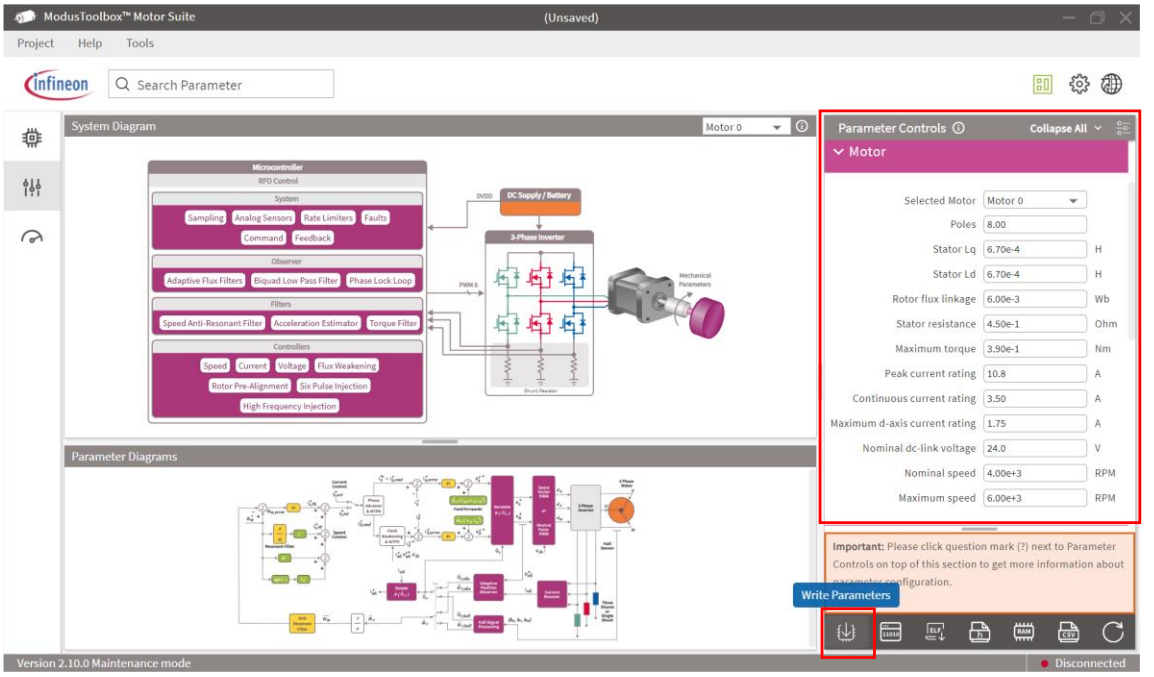

2. Click "**Default**" to flash the out-of-box firmware. If the code is changed and you want to download the new code, select "**Browse"**, then browse to the file location where it contains .elf and .hex files built from aforementioned ModusToolbox™ firmware project. For subsequent firmware flash, if the location of the files are the same as your last attempt to flash firmware, you can choose "**Last Selected"**.



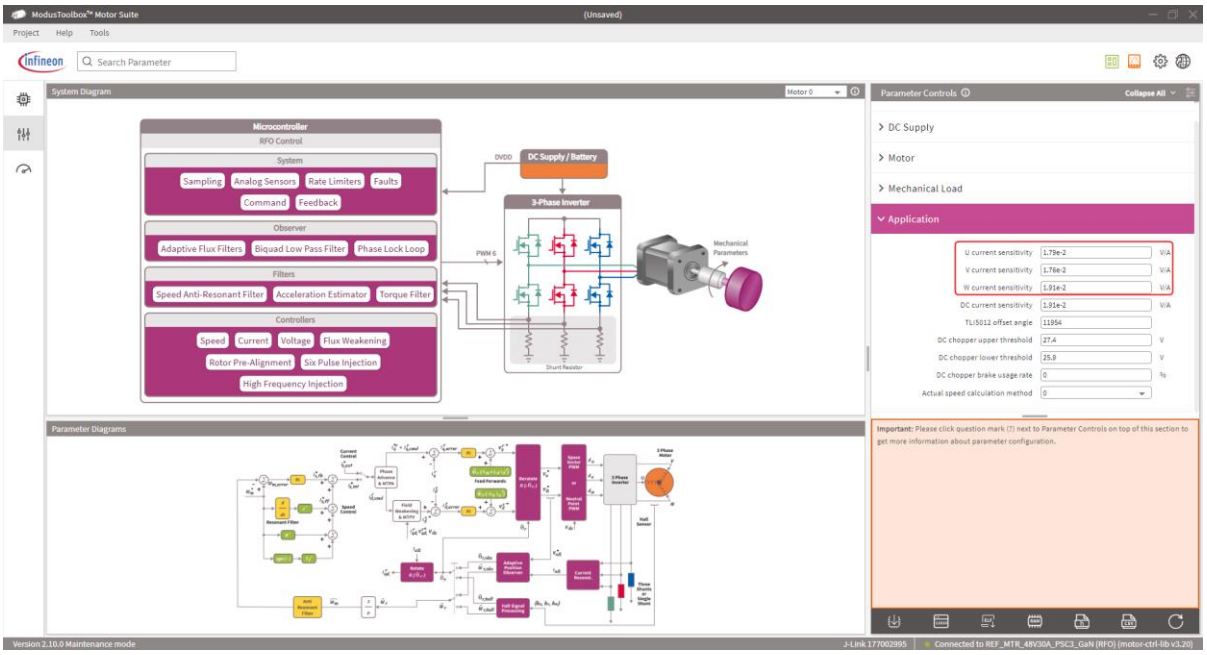

**Set parameters**

1. After firmware flashing is completed successfully, click "Read Device" to do a basic check that the parameters are correct.
2. Check and set every parameter on the right side under "Parameter Controls".
3. To run your specific motor, check that the motor parameters are correct as shown in the picture below.
4. Click "Write Parameters" to write all the modified parameters into RAM.






5. To get the best control performance, it is recommended to get **the U, V, W current sensor sensitivities specifically for this board** from the "**current sensor calibration report**" sheet provided inside the box, copy those values and set them respectively by overriding three parameters in GUI:
   U current sensitivity
   V current sensitivity
   W current sensitivity




   Alternatively, these three values can be changed in the source code of ParamConfig.h to take effective with only one time effort.

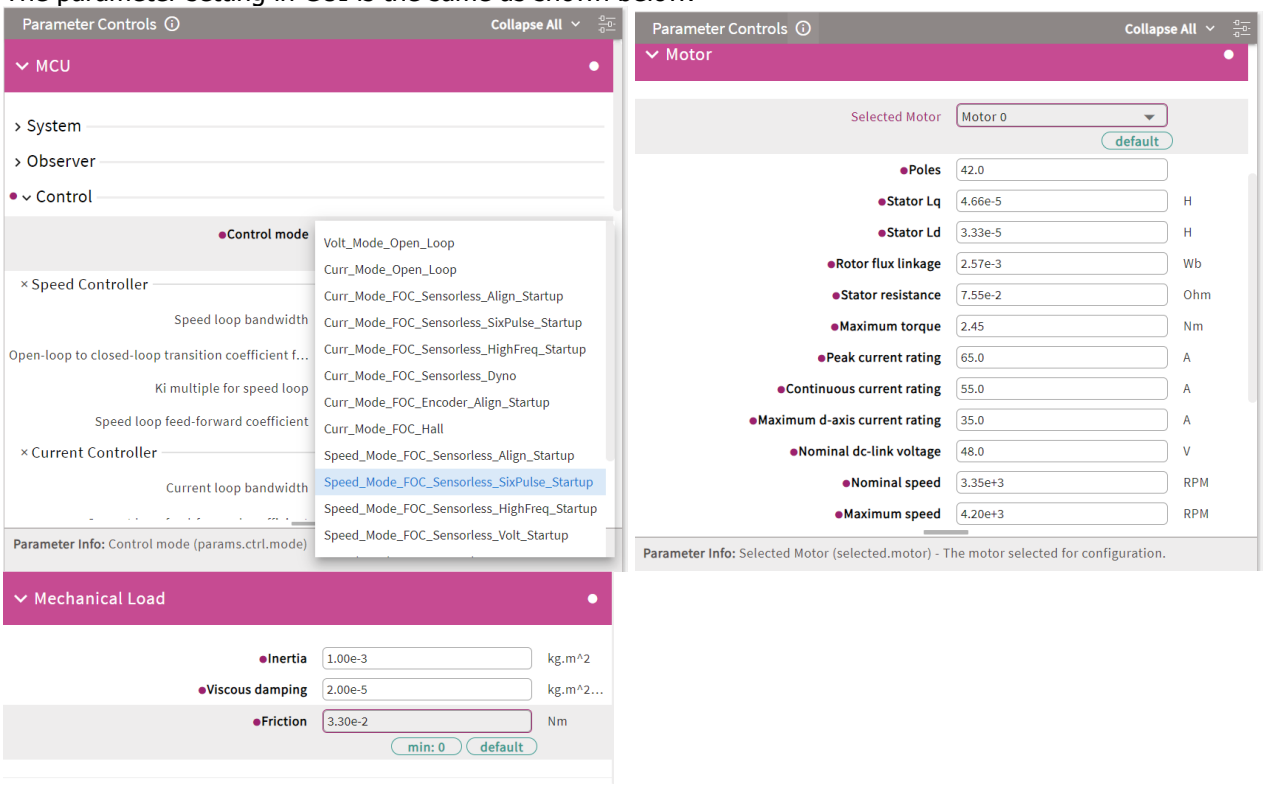
```
#define ADC_CS_U_CURRENT_SENSITIVITY          (17.756E-3f)
#define ADC_CS_V_CURRENT_SENSITIVITY          (17.457E-3f)
#define ADC_CS_W_CURRENT_SENSITIVITY          (18.042E-3f)
```

**Typical parameters to run a motor**

1. To run a motor, typical parameters are as follows. The example motor given is [MAD M7C10 IPE] with below parameter setting in the firmware source code (ParamConfig.h).

```
#elif (MOTOR_TYPE == MC710_100KV)
#define MOTOR_POLE                      (42.0f)                 /*[],  motor poles*/ //BLY172S
#define MOTOR_LQ                        (46.62E-6f)            /*[H], Stator q-axis inductance*/
#define MOTOR_LD                        (33.27E-6f)             /*[H], Stator d-axis inductance*/
#define MOTOR_I_AM                      (2.57E-3f)              /*[Wb],  Rotor flux linkage*/ //to confirm
#define MOTOR_R                         (75.5E-3f)             /*{Ohm],  stator resistance*/ //to confirm
#define MOTOR_TORQUE_MAX                (2.45f)                 /*[Nm],  maximum torque*/
#define MOTOR_CURRENT_PEAK              (65.0f)                  /*[A],  peak current rating*/
#define MOTOR_CURRENT_CONT              (55.0f)                  /*[A],  continuous current rating*/
#define MOTOR_ID_MAX                    (35.0f)                  /*[A], maximum d-axis current*/ //to confirm
#define MOTOR_VOLTAGE                   (48.0f)                 /*[V], motor voltage*/
#define MOTOR_NORM_SPEED                (3350.0f)                /*[RPM], nominal speed*/
#define MOTOR_MAX_SPEED                 (4200.0f)                /*[RPM],  maximum no load speed*/
/*******Mechanical Load*******/
#define MECH_INERTIA                    (1.0E-3f)              /*[kg.m^2],  Inertia*/
#define MECH_VISCOUS                    (2.0E-5f)               /*[kg.m^2/sec], Viscous Damping*/
#define MECH_FRICTION                   (3.3E-2f)                  /*[kg.m^2/sec^2], Frittion*/
/*******V/F offset and ratio*******/
#define MOTOR_CTRL_VOLT_VF_OFFSET       (4.0E-2f * 2.0f)          /*[V], V/F ramp voltage offset*/ //(0.15f)
#define MOTOR_CTRL_VOLT_VF_RATIO        (2.8E-3f)                 /*[V/(Rad/secz], V/f ramp slope*/ //(7.5E-3f)
```

2. The parameter setting in GUI is the same as shown below.




3. Click "Write Parameters" before running this motor.

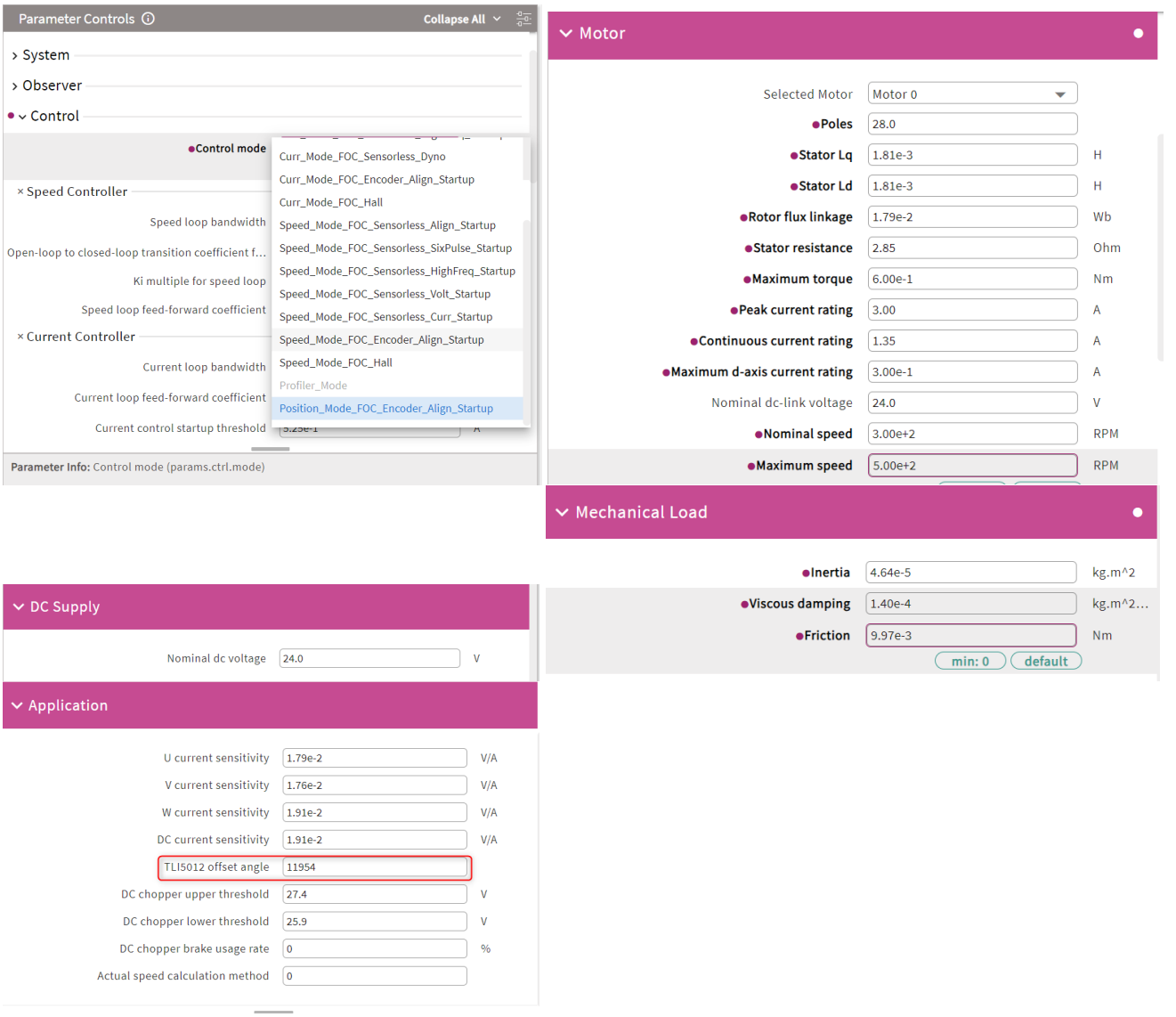
**Robotics motor typical parameters**

1. To run a robotics motor in **position control**, typical parameters are as follows.
   The example motor given is GL60 KV25 with below motor parameter setting in the firmware source code (ParamConfig.h). The example encoder is a TLE5012 angle sensor by SPI communication with MCU board.
   **Note: This motor is 24V motor, so DC supply should output 24V instead of 48V !**

```
/*******DC Supply*******/
#if (MOTOR_TYPE == GL60_KV25)
#define MOTOR_CTRL_VDC_NOM_VOLT         (24.0f)                 /*[V], Nominal DC bus voltage*/
```

```
#if (MOTOR_TYPE == GL60_KV25)
#define MOTOR_POLE                      (28.0f)                 /*[],  motor poles*/  //24N28P
#define MOTOR_LQ                        (1.815E-3f)              /*[H], Stator q-axis inductance*/
#define MOTOR_LD                        (1.815E-3f)              /*[H], Stator d-axis inductance*/
#define MOTOR_I_AM                      (17.92E-3f)             /*[Wb],  Rotor flux linkage*/
#define MOTOR_R                         (2.85f)                 /*[Ohm],  stator resistance*/
#define MOTOR_TORQUE_MAX                (0.60f)                 /*[Nm],  maximum torque*/
#define MOTOR_CURRENT_PEAK              (3.0f)                  /*[A],  peak current rating*/
#define MOTOR_CURRENT_CONT              (1.35f)                  /*[A],  continuous current rating*/
#define MOTOR_ID_MAX                    (0.3f)                  /*[A], maximum d-axis current*/
#define MOTOR_VOLTAGE                   (24.0f)                 /*[V], motor voltage*/
#define MOTOR_NORM_SPEED                (300.0f)                /*[RPM], nominal speed*/
#define MOTOR_MAX_SPEED                 (500.0f)                /*[RPM],  maximum no load speed*/
/*******Mechanical Load*******/
#define MECH_INERTIA                    (4.64E-5f)              /*[kg.m^2],  Inertia*/
#define MECH_VISCOUS                    (1.4E-4f)               /*[kg.m^2/sec], Viscous Damping*/
#define MECH_FRICTION                   (9.97E-3f)              /*[kg.m^2/sec^2], Frittion*/
/*******V/F offset and ratio*******/
#define MOTOR_CTRL_VOLT_VF_OFFSET       (0.2f)                   /*[V], V/F ramp voltage offset*/ //(0.15f)
#define MOTOR_CTRL_VOLT_VF_RATIO        (30.0E-3f)               /*[V/(Rad/secz], V/f ramp slope*/ //(7.5E-3f)
```

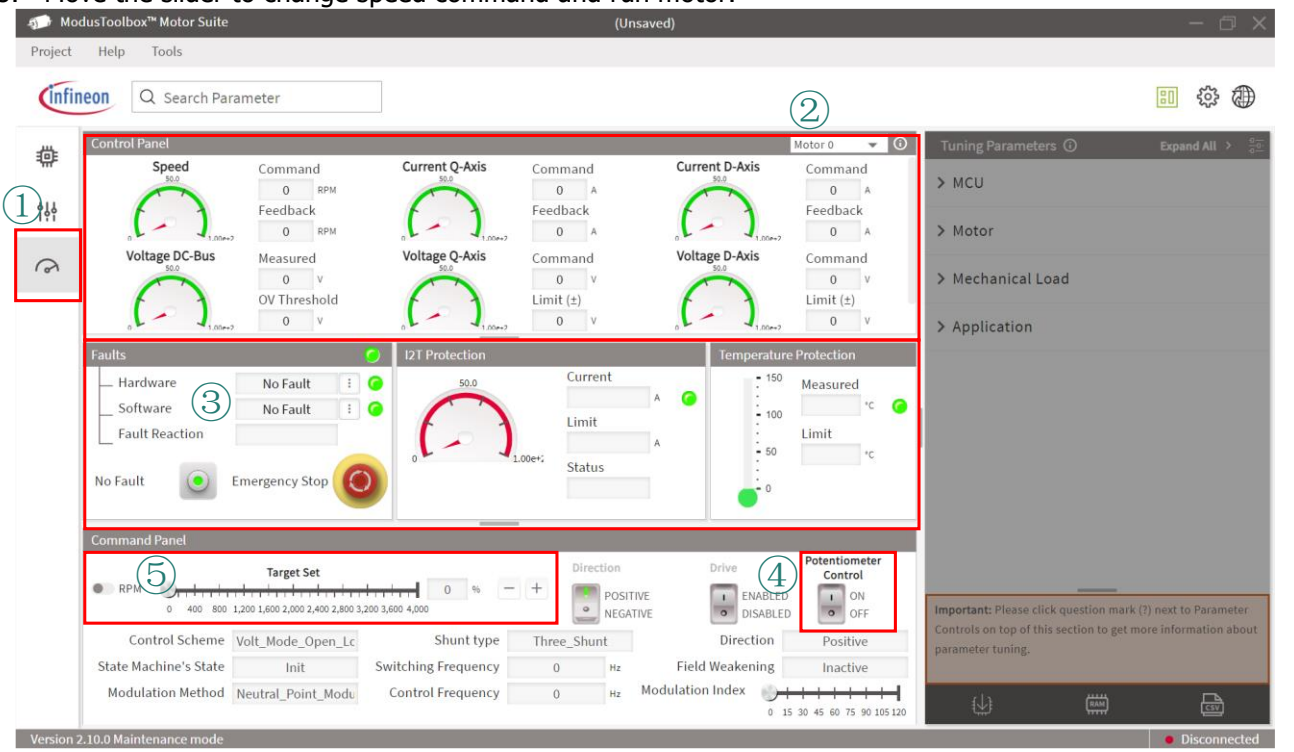
2. The parameter setting in GUI is the same as in source code, as shown below.
Please refer to the "Offset angle calibration" chapter in User guide of this board about how to set the TLE5012 offset angle.




3. Click "Write Parameters" before running a robotics motor.

**Run motor**

1. Choose the Test Bench tab to control/run a motor.
2. Indicators show real time parameters value.
3. These are the indicators showing any active faults.
4. Toggle Potentiometer Control to OFF in order to use Motor Suite to control the motor's target value/speed.
5. Move the slider to change speed command and run motor.




Document number: 002-39405 Rev. **
Date: 07 / 2026

All rights reserved. © 2026
Infineon Technologies AG