EVAL_100W_DRIVE_CFD2
Getting started guide

September 2017

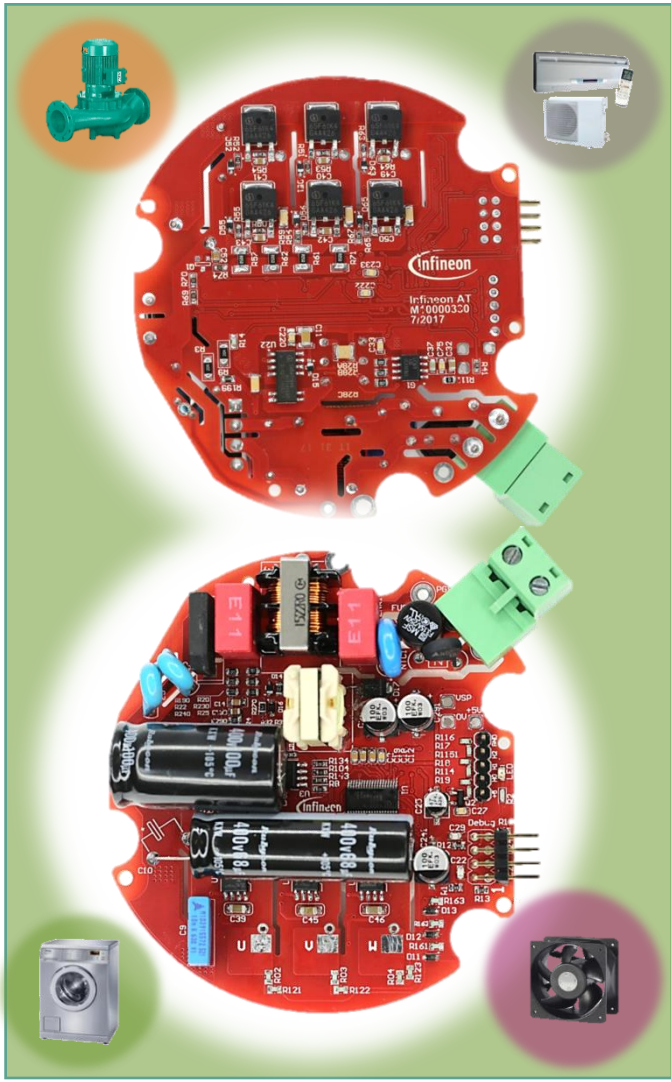
Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

Introduction



- **Compact 3-phase motor drive** system solution up to 100 W
- Designed for **sensorless or sensored** motor control
- **Spin your motor** with easy-to-use GUI for tuning of motor
- The hardware board and motor control software provides:
  - Sensorless speed controlled direct FOC startup
  - 7-segment / 5-segment SVM
  - Over-current protection by CTRAP
  - Speed controlled using µC/Probe™ GUI's slider

Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

System overview

Components

- **XMC1302 Microcontroller**
  - Based on XMC1302 ARM® Cortex®-M0 MCU enabling sensored and sensorless motor control
  - Supports HALL
  - Includes J-Link debug interface by Segger
  - µC/Probe™-based GUI for parametrization and tuning
- **Half bridge gate driver 2EDL05N06PF**
- **CoolMOS™ IPD65R1K4CFD**
- **CoolSet™ ICE5QR4770AG**
- **Software package** (downloadable from www.infineon.com/eval-100w-drive-cfd2)
  - Micrium µC/Probe™ for XMC™
  - .zip file with DAVE™ 4 project for the EVAL_100W_DRIVE_CFD2
  - kit (e.g. Eval-100W_FOC_XMC13_v1.zip)

System overview

For example the EVAL_100W_DRIVE_CFD2 Kit consists of:

Hardware

- EVAL_100W_DRIVE_CFD2 board
- XMC™ Link
- Micro-B USB cable

Software

- Micrium µC/Probe™ for XMC™
- Eval-100W_FOC_XMC13.zip file

(downloadable from https://www.infineon.com/cms/en/product/productType.html?productType=5546d4625e763904015ec84038b42ff2&redirId=59073#ispnTab12)

System overview

Software package content description:

- **Eval-100W_FOC_XMC13**

  A DAVE™ 4 project for XMC1302-TO380200 device. The user can configure the motor parameter and startup mode in the source code. The compile firmware needs to be programed to EVAL_100W_DRIVE_CFD2 board via DAVE™ 4 IDE and XMC Link

- **Eval-100W_FOC_XMC13.elf**

  A debugging file used by µC/Probe™ GUI tool

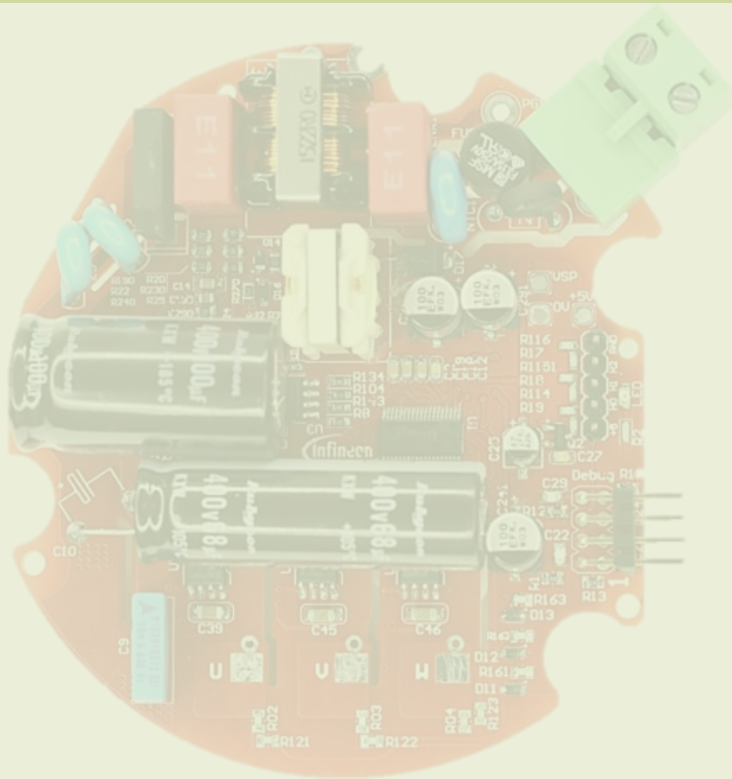
- **Eval-100W_FOC_XMC13.wspx**

  A µC/Probe™ project file for predefined GUI used for system parametrization and tuning

Hardware overview

Hardware

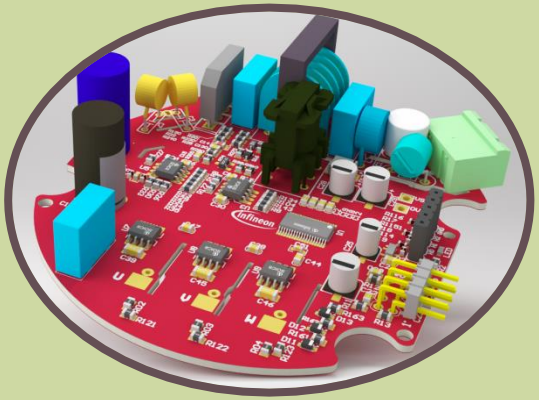
- **Kit name**
  - EVAL_100W_DRIVE_CFD2
- **Kit description**
  - EVAL_100W_DRIVE_CFD2 board
  - USB cable
  - XMC™ Link – Isolated Debug Probe
- **Order number**
  - EVAL100WDRIVECFD2TOBO1
- **Input voltage/output power**
  - 230 VAC/100 Watt max

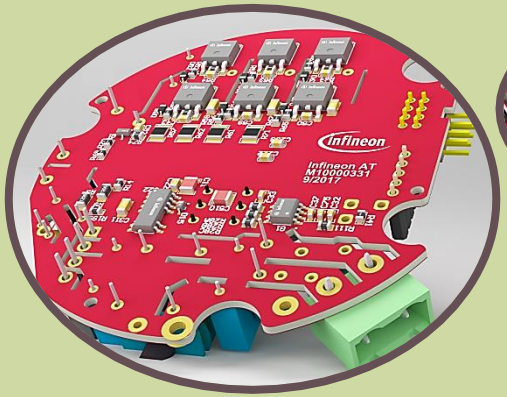


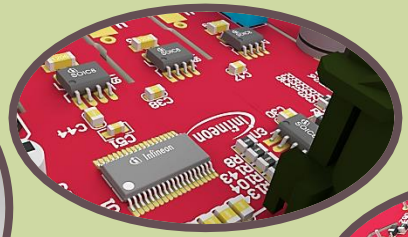
Hardware overview

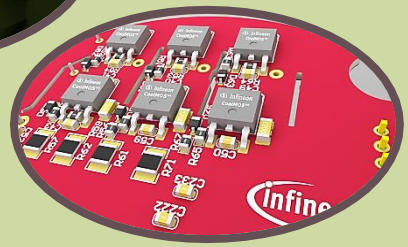
Infineon parts utilized on EVAL_100W_DRIVE_CFD2 board:

Infineon Parts	Order Number
XMC1300 Microcontroller	XMC1302-T038F0200
600 V half bridge gate driver	2EDL05N06PF
650 V CoolMOS™	IPD65R1K4CFD
Quasi-resonant CoolSET™	ICE5QR4770AG
5 V LDO regulator	IFX1763XEJV50









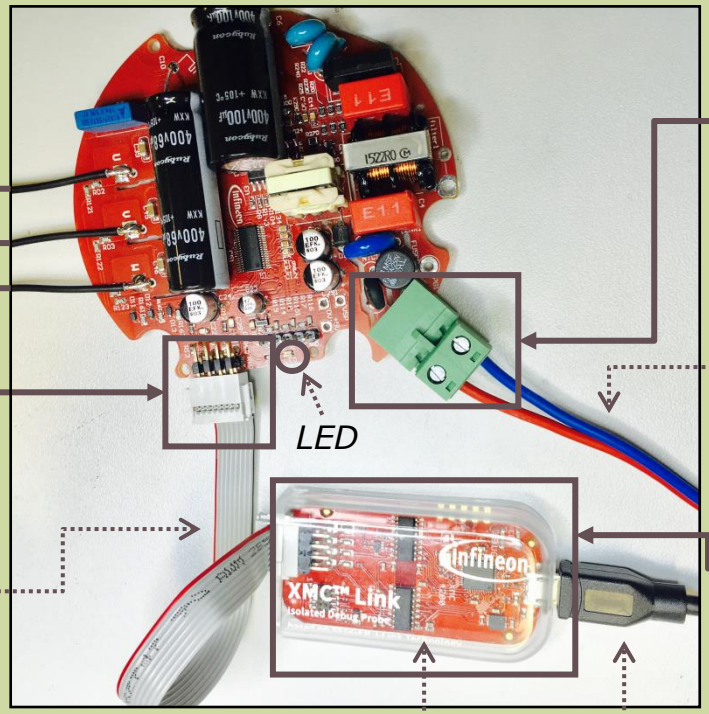
Hardware overview

To properly setup the board, follow these steps:

1 Connect 3-phase motor wiring to 'U V W' connector

2a Connect XMCTM Link's ribbon cable to pin 1 of Debug connector

3 Connect AC source to EVAL_100W_DRIVE_CF D2 board



2b Connect to PC via micro-B USB cable

Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

DAVE™ 4 setup

Micrium µC/Probe™

Download Micrium µC/Probe™ for XMC™ installer package from:

www.infineon.com/ucprobexmc

Installation requirements:

- PC with Windows Vista, Windows 7, Windows 8, Windows 10 – 32 bit & 64 bit
- RAM – 3 GB or more

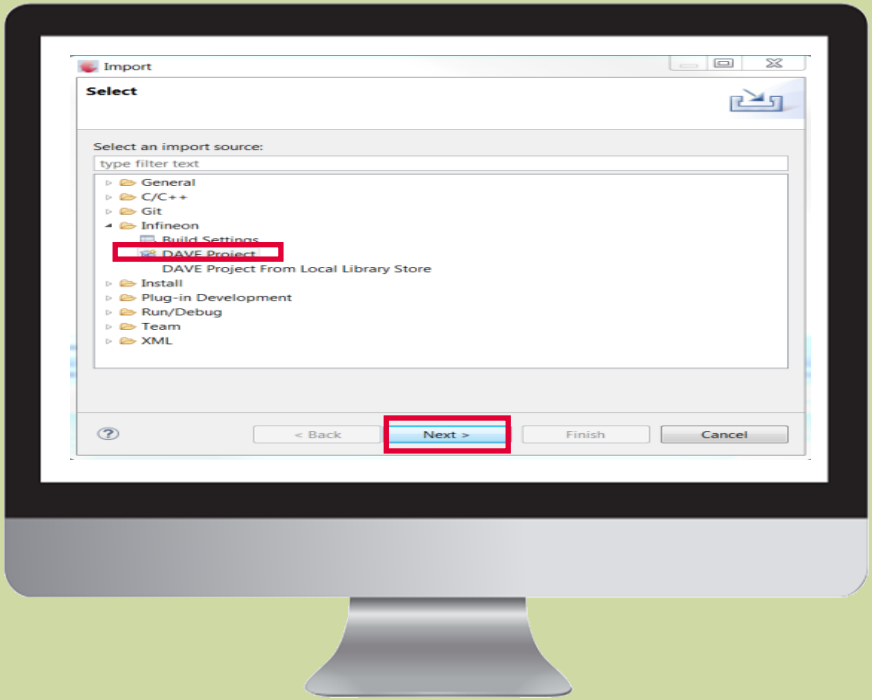
Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

Import the *.zip project

1. Import the *.zip file into DAVE™ 4 IDE by clicking on **File** > **Import**

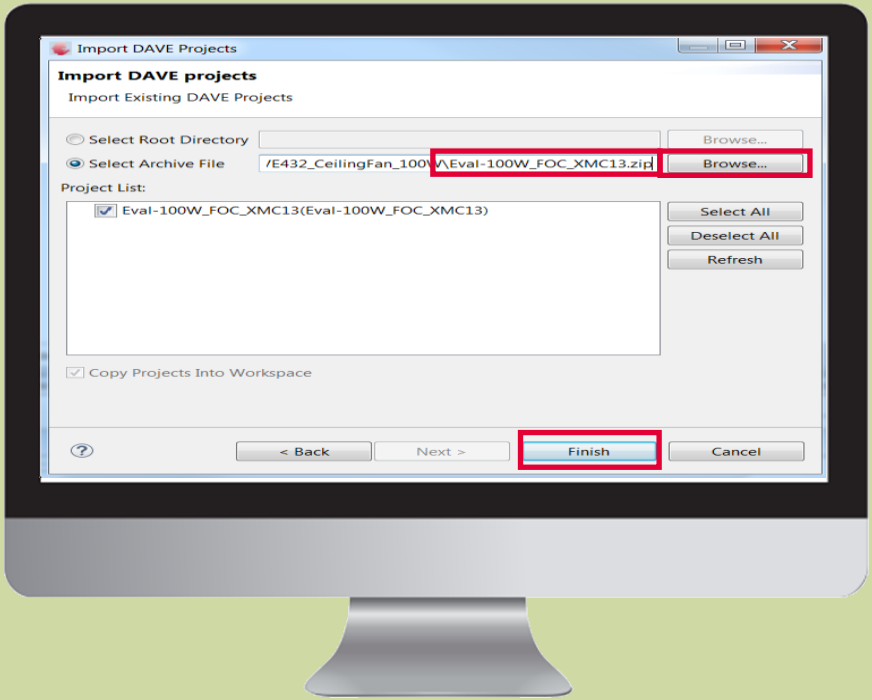
2. Select **Infineon** > **DAVE** Project and click the **Next** button



Import the *.zip project

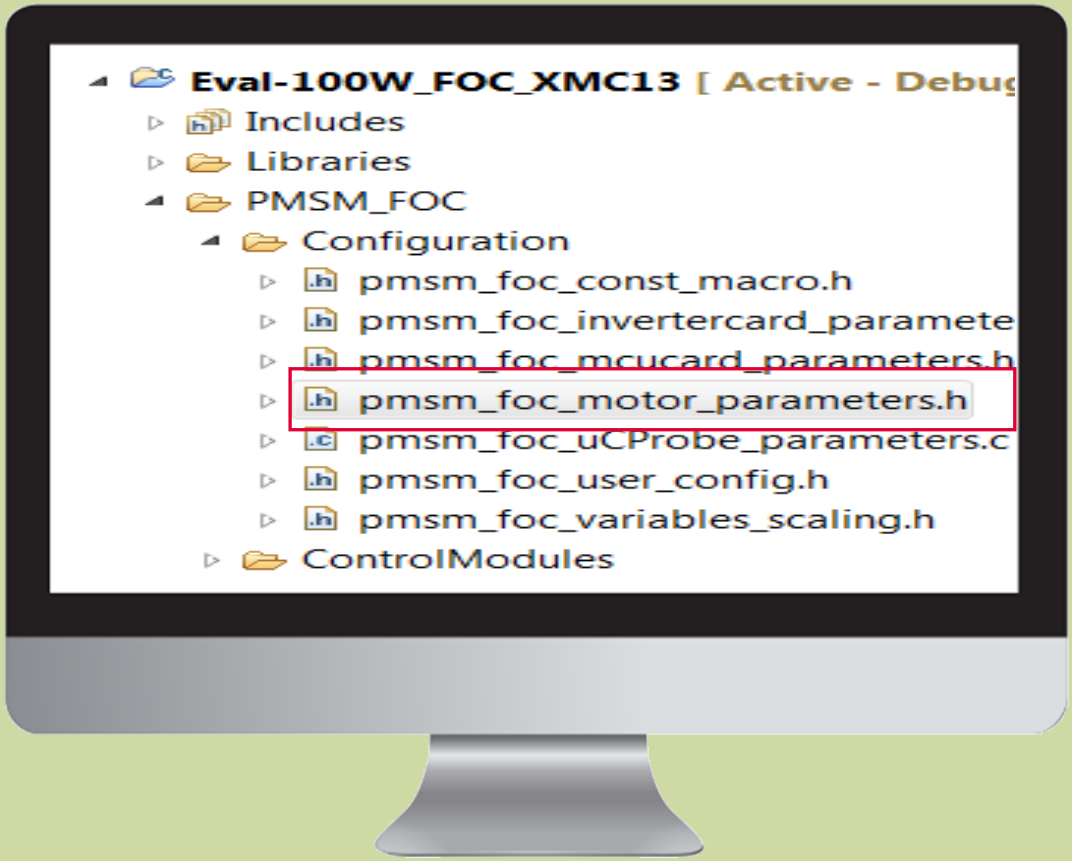
3 Click **Browse** and navigate to the *.zip file location

4 Click the **Finish** button



Input motor parameters

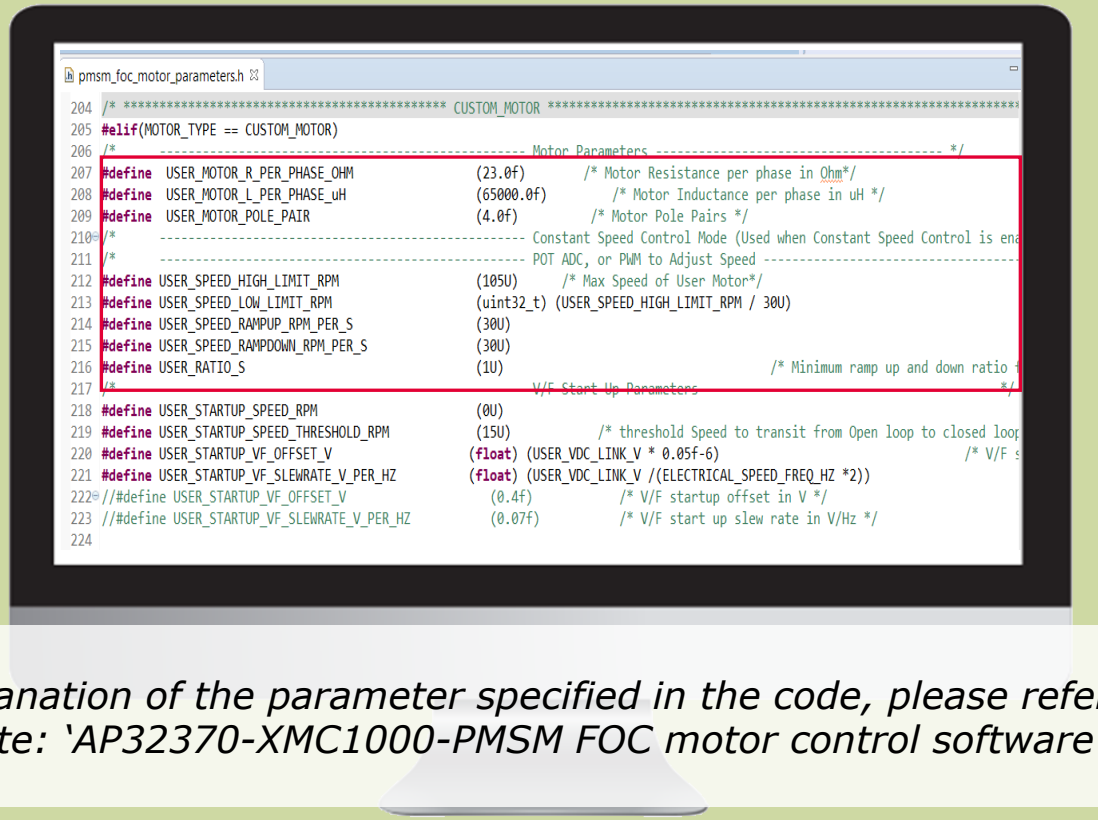
1. Double click on **pmsm_foc_motor_parameters.h**



Input motor parameters

2

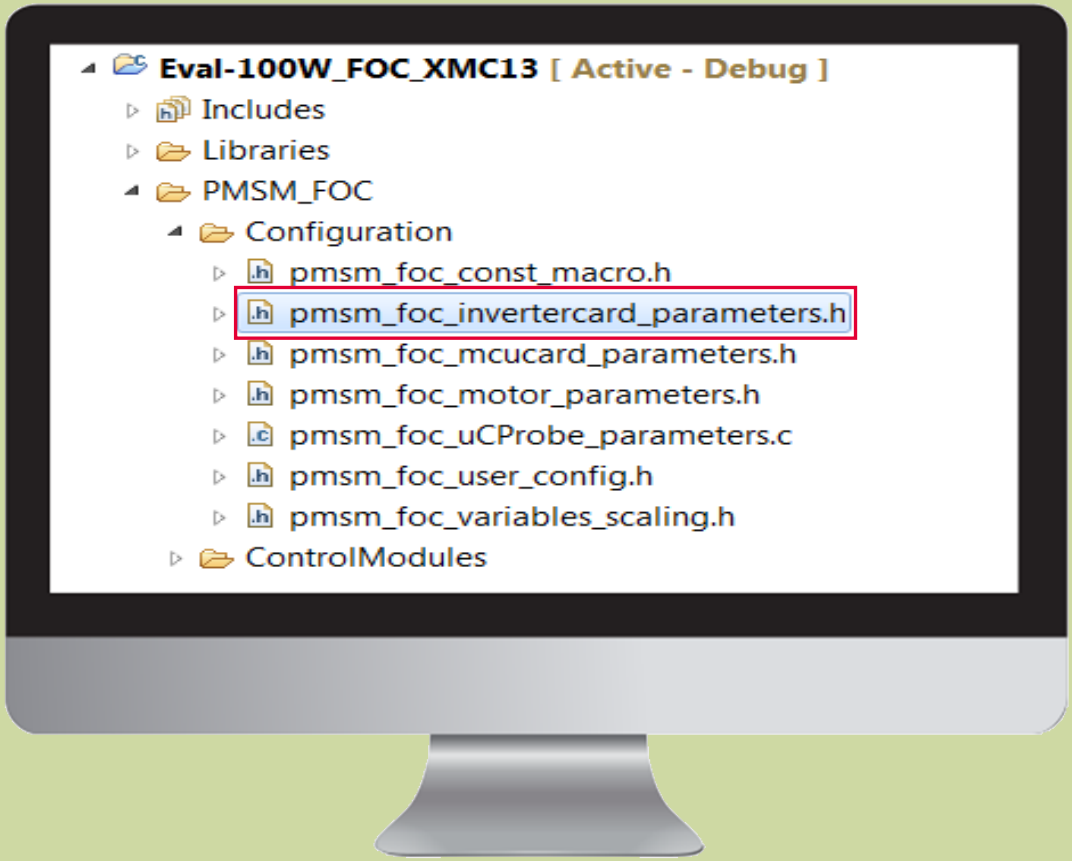
Scroll to **(MOTOR_TYPE == CUSTOM_MOTOR)** and update the parameters in the red box with your motor's data



For more explanation of the parameter specified in the code, please refer to XMC1000 application note: 'AP32370-XMC1000-PMSM FOC motor control software using XMC™'

Input inverter circuitry parameters

1. Double click on **pmsm_foc_invertercard_parameters.h**



Input inverter circuitry parameters

2

Scroll to **(INVERTERCARD_TYPE == CUSTOM_INVERTER)**

pmsm_foc_invertercard_parameters.h

```
196 #elif(INVERTERCARD_TYPE == CUSTOM_INVERTER)
197 #define INTERNAL_OP_GAIN                    DISABLED              /*1. ENABLED       2. DISABLED (Please configure C
198 #define USER_VDC_LINK_V                     (320.0f)               /* Hardware Inverter VDC link voltage in V  */
199 #define USER_CCU8_PWM_FREQ_HZ               (16000U)              /* CCU8 PWM Switching Frequency in Hz*/
200 #define USER_DEAD_TIME_US                   (1.0f)                /* deadtime, rise(left) and fall values in us  */
201 #define USER_BOOTSTRAP_PRECHARGE_TIME_MS    (20U)                 /* Initial Bootstrap precharging time in ms */
202 #define USER_DC_LINK_DIVIDER_RATIO          (float)(3.3f/(3.3f+400.0f))          /* R1/(R2+R1) ratio for DC link MCL
203 #define USER_VBEMF_RATIO                    (float)(3.3f/(3.3f+400.0f))          /* R1/(R2+R1) ratio for BEMF Voltag
204 #define USER_CURRENT_TRIP_THRESHOLD_A       (3.0f)                /* threshold current for trip detection in Ampere*/
205 #define USER_TRIP_THRESHOLD_TIME_MS         (100U)                /* threshold time for trip detection in ms */
206 #define USER_MAX_RETRY_MOTORSTARTUP_TRIP    (3U)                 /* Max retry of motor startup if trip  */
207 /*      ------------------------------------------------------ Motor Phase Current Measurement ------------------------------
208 #define USER_R_SHUNT_OHM                    (0.05f)               /* Phase shunt resistor in ohm */
209 #define USER_DC_SHUNT_OHM                   (0.5f)                /* DC link shunt current resistor in ohm */
210 #define USER_RIN_PHASECURRENT_KOHM          (1.0f)                /* R_IN (of equivalent amplifier) kohm */
211 #define USER_R_PHASECURRENT_FEEDBACK_KOHM   (10.0f)               /* R_FEEDBACK (of equivalent amplifier) kohm */
212 #define USER_RIN_DCCURRENT_KOHM             (10.0f)               /* Rf for dc current sensing */
213 #define USER_R_DCCURRENT_FEEDBACK_KOHM      (55.0f)               /* Rin for dc current sensing */
214 #define USER_MAX_ADC_VDD_V                  (5.0f)                /* VDD5, maximum voltage at ADC */
215 #define G_OPAMP_PER_PHASECURRENT            (USER_R_PHASECURRENT_FEEDBACK_KOHM / USER_RIN_PHASECURRENT_KOHM)
216
217 #if(INTERNAL_OP_GAIN == ENABLED)
218 #define OP_GAIN_FACTOR                      (3U)                  /* Different HW Board has different OP Gain fac
219 #elif(INTERNAL_OP_GAIN == DISABLED)
220 #define OP_GAIN_FACTOR                      G_OPAMP_PER_PHASECURRENT
221 #endif
222
223 #define GATE_DRIVER_INPUT_LOGIC             PASSIVE_LOW           /*1. PASSIVE_HIGH       2. PASSIVE_LOW (Plea
224
```

*The inverter parameters is already preset to the EVAL_100W_DRIVE_CFD2 board's value, e.g. USER_VDC_LINK_V is 320 V (√2*230 V) with the assumption that AC input voltage is 230 V*

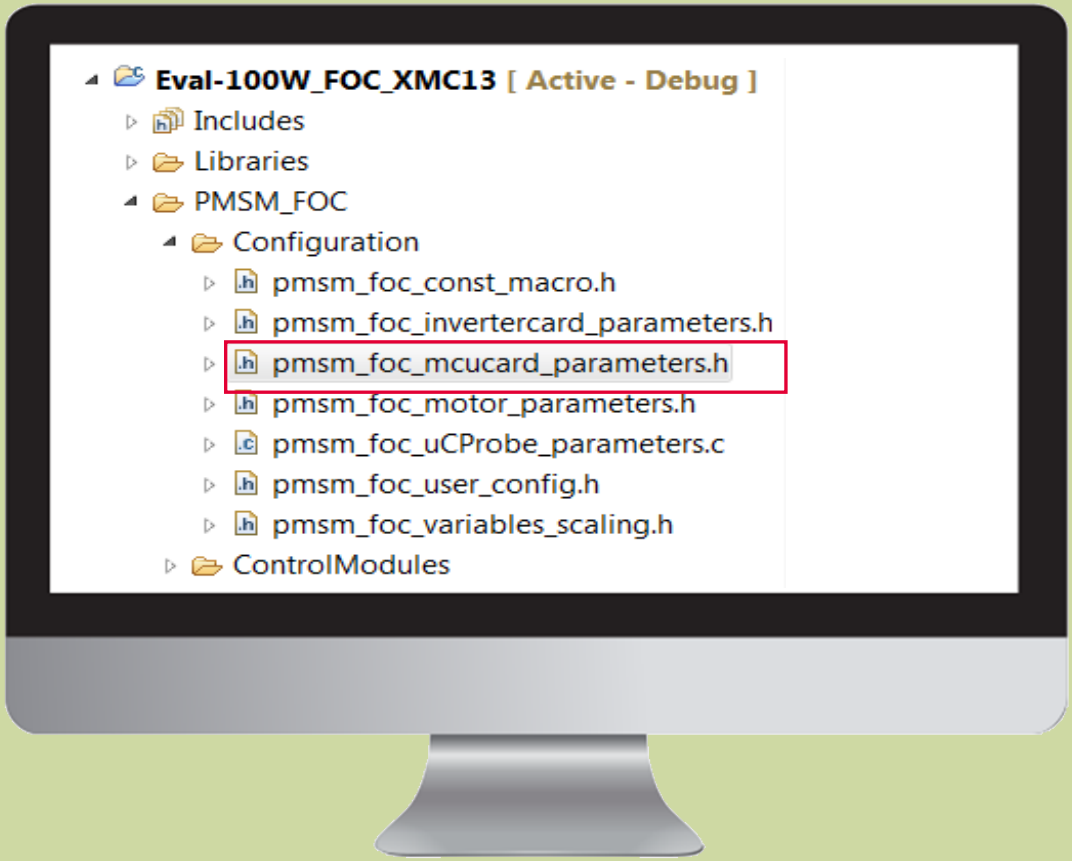
The PWM frequency driving the power MOSFET is set to 16 kHz

The minimum PWM frequency can be set is 6.6 kHz

The user does NOT need to change the value here if AC input voltage to EVAL_100W_DRIVE_CFD2 board is 230 V

Input XMC1302 pin setting

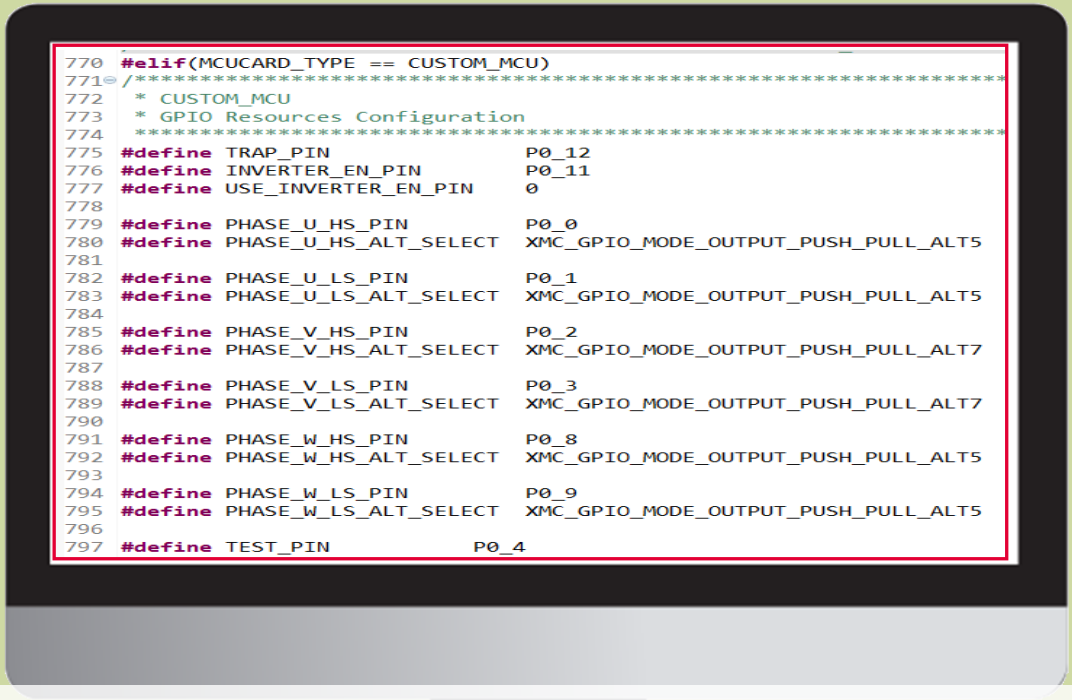
1. Double click on **pmsm_foc_mcucard_parameters.h**



Input XMC1302 pin setting

2

Scroll to **(MCUCARD_TYPE == CUSTOM_MCU)**



```
770 #elif(MCUCARD_TYPE == CUSTOM_MCU)
771 /*******************************************************************************
772  * CUSTOM_MCU
773  * GPIO Resources Configuration
774  *******************************************************************************
775 #define TRAP_PIN                  P0_12
776 #define INVERTER_EN_PIN           P0_11
777 #define USE_INVERTER_EN_PIN       0
778
779 #define PHASE_U_HS_PIN            P0_0
780 #define PHASE_U_HS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
781
782 #define PHASE_U_LS_PIN            P0_1
783 #define PHASE_U_LS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
784
785 #define PHASE_V_HS_PIN            P0_2
786 #define PHASE_V_HS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT7
787
788 #define PHASE_V_LS_PIN            P0_3
789 #define PHASE_V_LS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT7
790
791 #define PHASE_W_HS_PIN            P0_8
792 #define PHASE_W_HS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
793
794 #define PHASE_W_LS_PIN            P0_9
795 #define PHASE_W_LS_ALT_SELECT     XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
796
797 #define TEST_PIN            P0_4
```

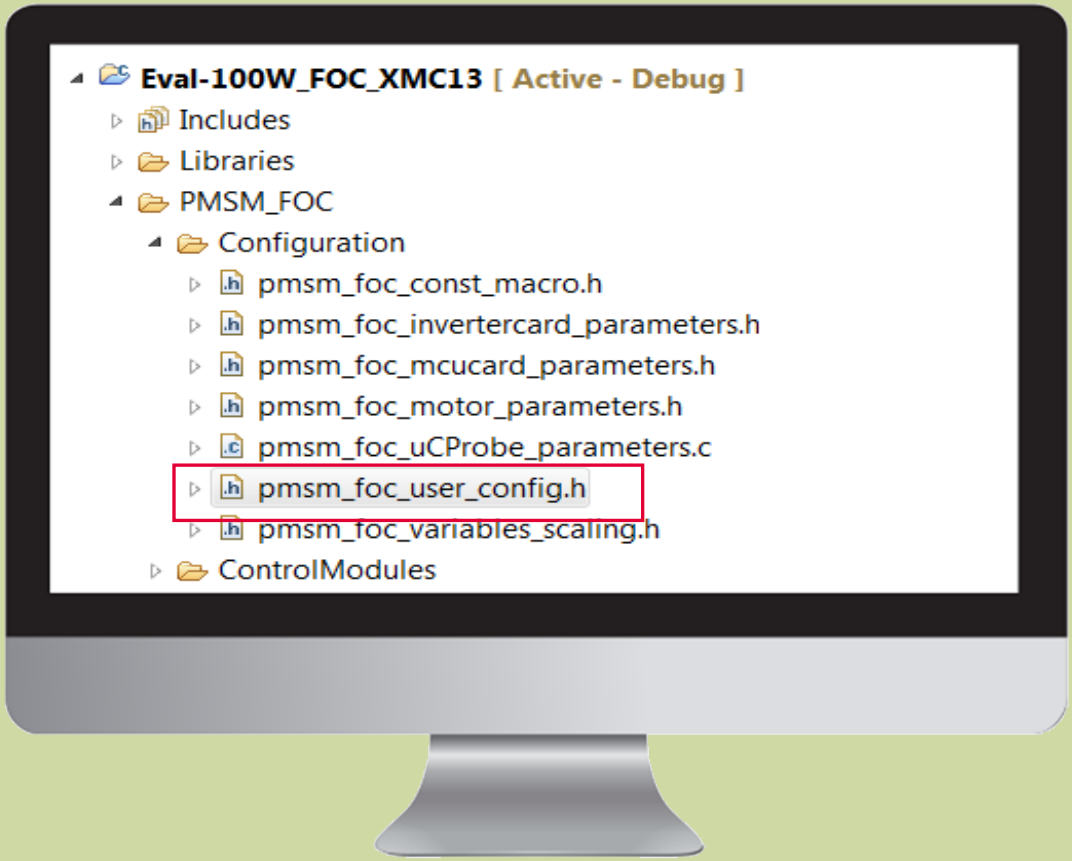
INVERTER_EN_PIN is not available on this board. Hence, the USE_INVERTER_EN_PIN is 0

The MCU pinout setting is already set according to the EVAL_100W_DRIVE_CFD2 board. The user does NOT need to change it!

Input motor startup: current sensing setting

1

Double click on **pmsm_foc_user_config.h**



Input motor startup: current sensing setting

2

Scroll to **(MCUCARD_TYPE == CUSTOM_MCU)**

pmsm_foc_user_config.h

```
78
79 /*******************************************************************************************************************
80  * MACROS
81  *******************************************************************************************************************/
82 #define  PMSM_FOC_HARDWARE_BOARD                CUSTOM_KIT                          /*1. KIT_XMC1X_AK_MOTOR_001
83                                                                                       2. KIT_XMC750WATT_MC_AK_V1
84                                                                                       3. CUSTOM_KIT*/
85
86 /*      ------------------------------------------------ Current feedback Sensing Mechanism ------------------------------
87 #define  CURRENT_SENSING                        USER_THREE_SHUNT_SYNC_CONV           /*1. USER_SINGLE_SHUNT_CONV
88                                                                                       2. USER_THREE_SHUNT_ASSYNC_CONV
89                                                                                       3. USER_THREE_SHUNT_SYNC_CONV*/
90 /*      ------------------------------------------------ FOC Control and Startup Scheme (Only Select 1 Scheme at one time)
91 #define MY_FOC_CONTROL_SCHEME                  SPEED_CONTROLLED_DIRECT_FOC          /*1. SPEED_CONTROLLED_VF_ONLY,
92                                                                                       2. SPEED_CONTROLLED_VF_MET_FOC
93                                                                                       3. SPEED_CONTROLLED_DIRECT_FOC
94                                                                                       4. TORQUE_CONTROLLED_DIRECT_FOC
95                                                                                       5. VQ_CONTROLLED_DIRECT_FOC */
96 /*      ------------------------------------------------ Micrium uC Probe Enable/Disable  ------------------------------
97 #define uCPROBE_GUI                            ENABLED                              /*1. ENABLED       2. DISABLED (
98 /*      ------------------------------------------------ Reference Speed Adjustment Method ------------------------------
99 #define SETTING_TARGET_SPEED                   MICRIUM_UC_ONLY                        /*1. MICRIUM_UC_ONLY
100                                                                                       2. BY_POT_ONLY  */
```

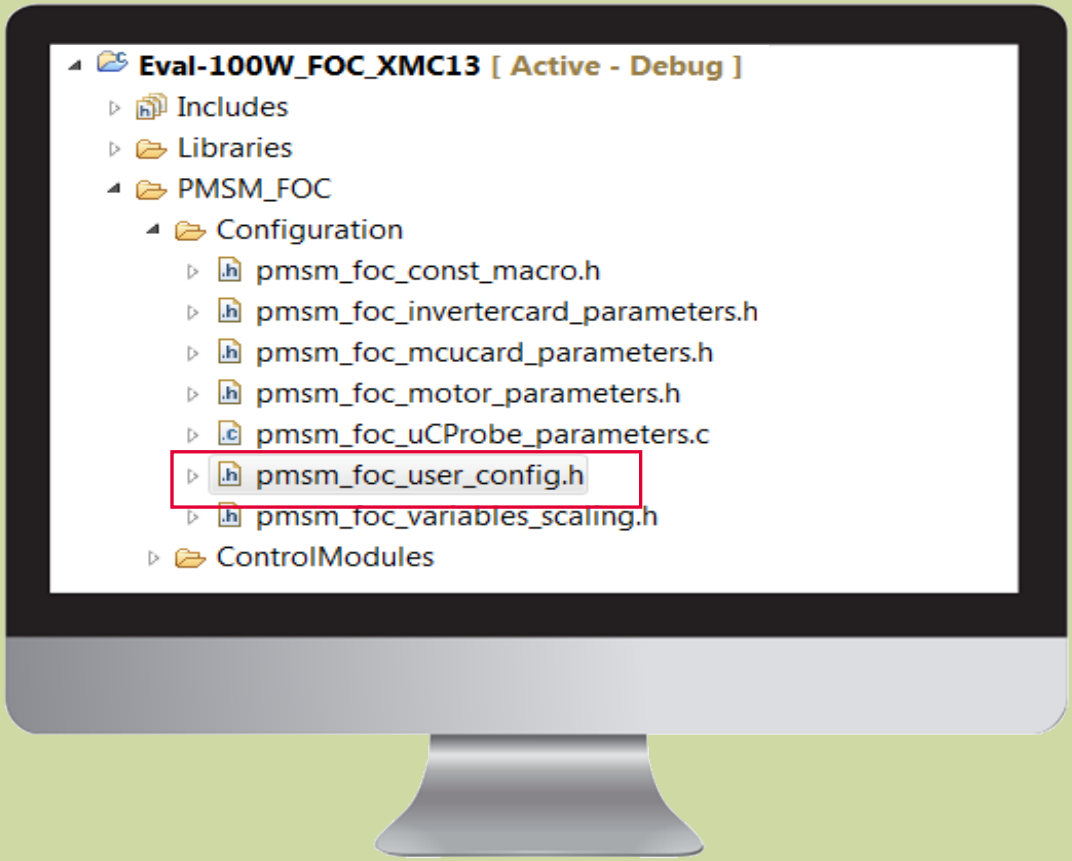
The user can select 3 types of current sensing methods (as shown in green comment lines)

The user can select 5 types of motor control schemes (as shown in green comment lines)

3 shunt synchronous current sensing and speed controlled direct FOC startups have been selected by default. The user does NOT need to change these!

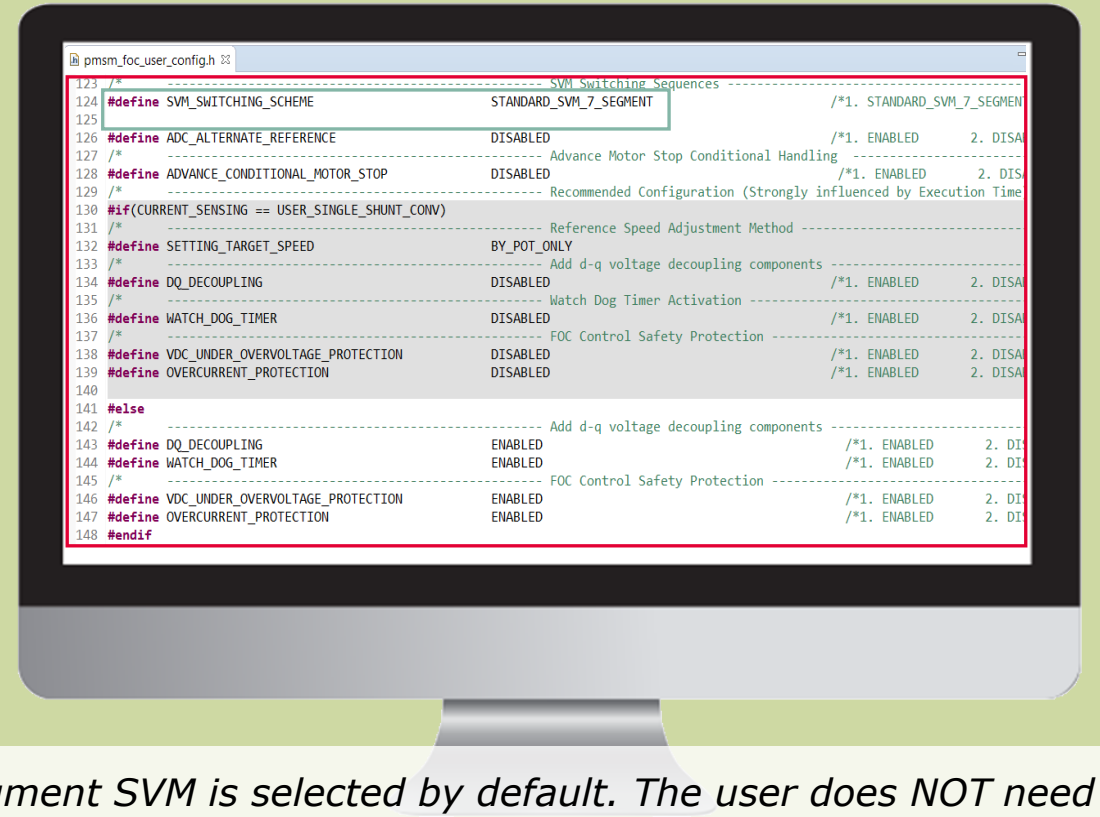
Input SVM scheme

1. Double click on **pmsm_foc_user_config.h**



Input SVM scheme

2 The user can select 2 types of SVM scheme: 7-segment or 5-segment SVM (as shown in green comment lines)

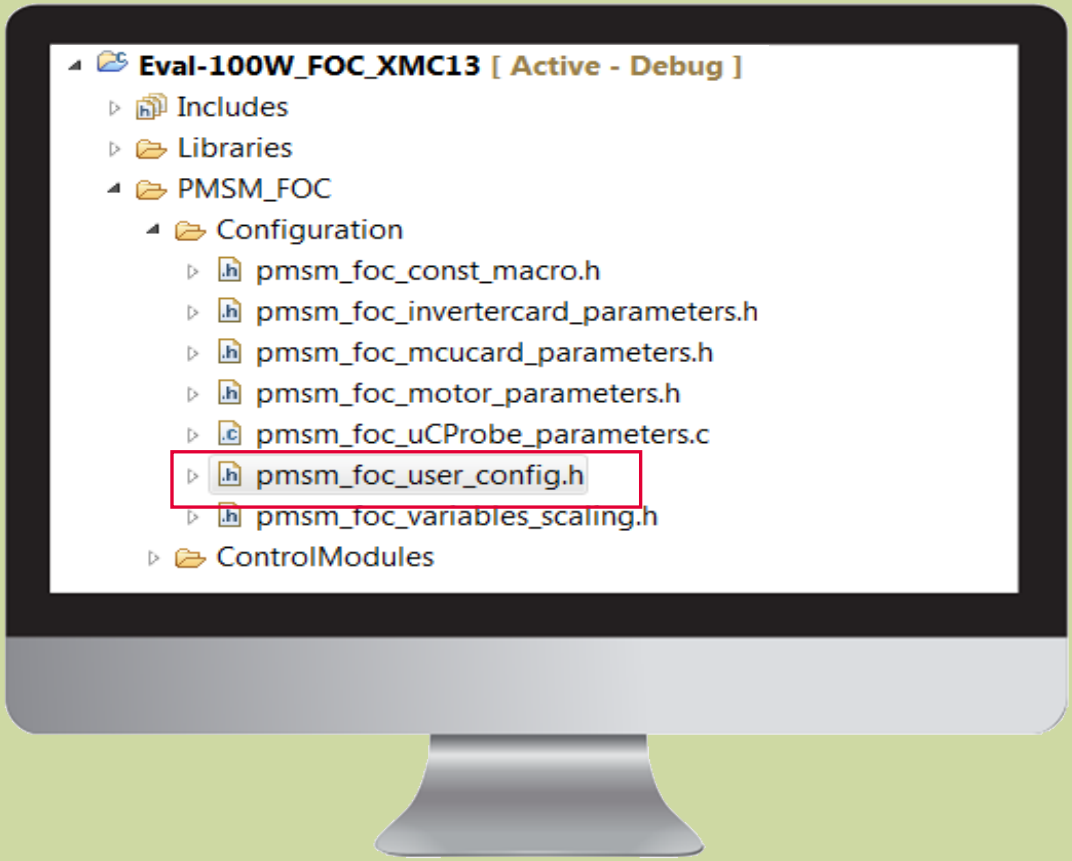


Standard 7-segment SVM is selected by default. The user does NOT need to change this!

Input current and voltage protection setting

1

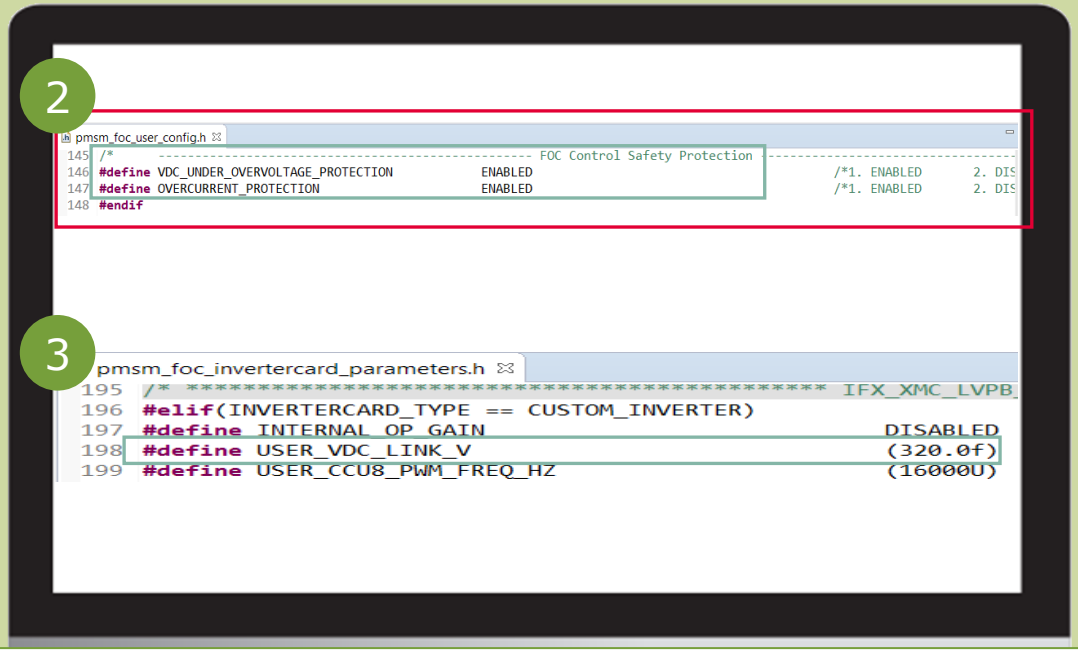
Double click on **pmsm_foc_user_config.h**



Input current and voltage protection setting

2

User can enable over-current and under-voltage protection (as shown in the green comment lines)



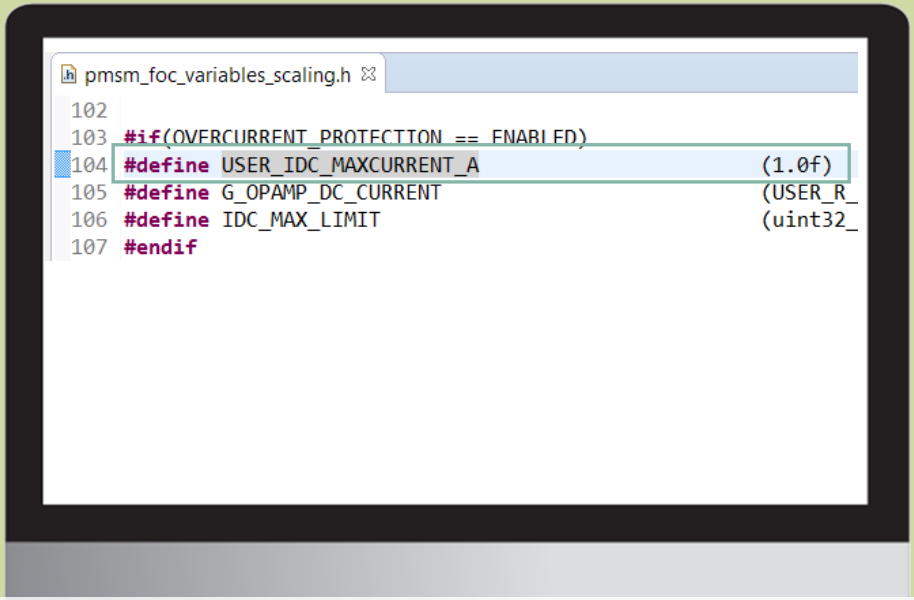
3

If the DC link voltage is <0.8 or >1.2 of USER_VDC_LINK_V, the PWM signal input to gate driver will go into high impedance and the motor will slowly stop

Input current and voltage protection setting

4

If the average current flow through the DC link shunt resistor R57 is > USER_IDC_MAXCURRENT_A, the reference speed will reduce until the current is < USER_IDC_MAXCURRENT_A

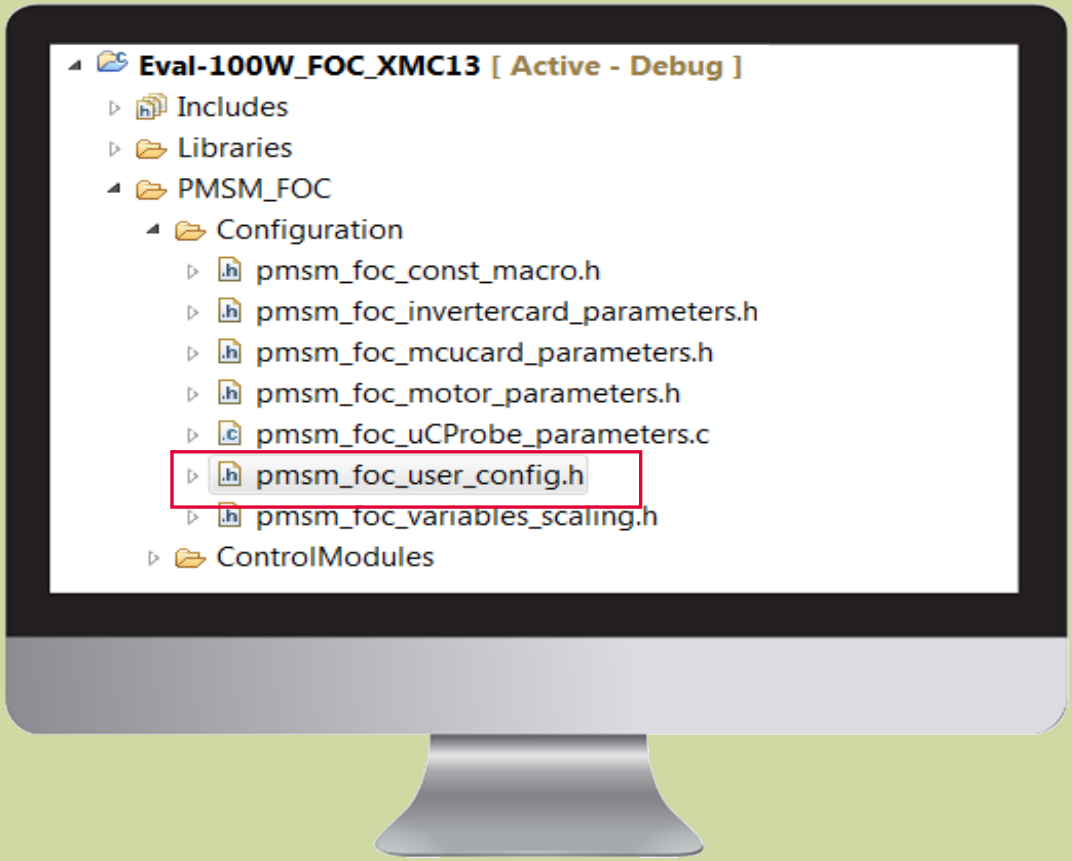


Under-voltage and over-current protection have been selected by default. The user does NOT need to change this!

USER_VDC_LINK_V and USER_IDC_MAXCURRENT_A values might need to change depending on user motor setting

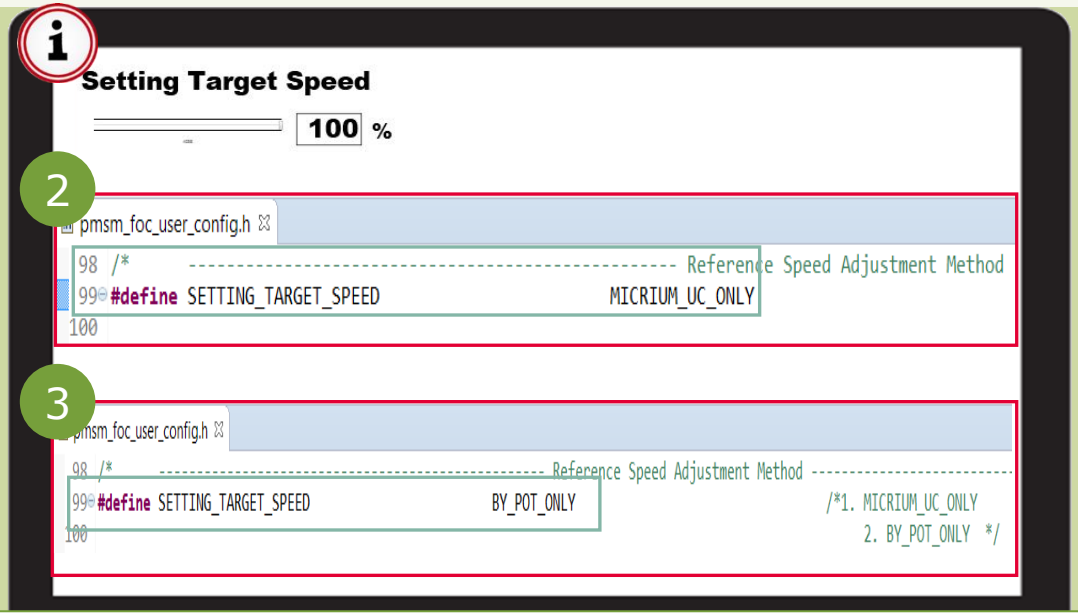
Input speed control setting

1. Double click on **pmsm_foc_user_config.h**



Input speed control setting

By default, the speed control method is using µC/Probe™ GUI's speed slider



2 The user can choose to use an external 10 kohm potentiometer as speed control method but he has to solder the potentiometer on the EVAL_100W_DRIVE_CFD2 board

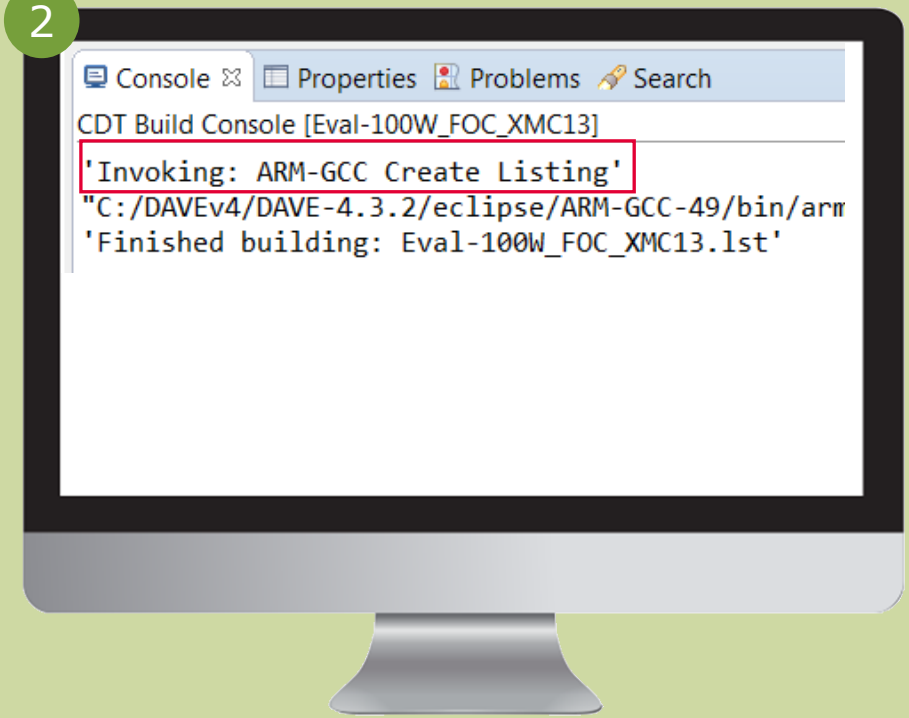
3 Change the speed control method to BY_POT_ONLY

Compile and download the code

1. Click **Build Active Project** 

2. Check **Eval-100W_FOC_XMC13.lst** created

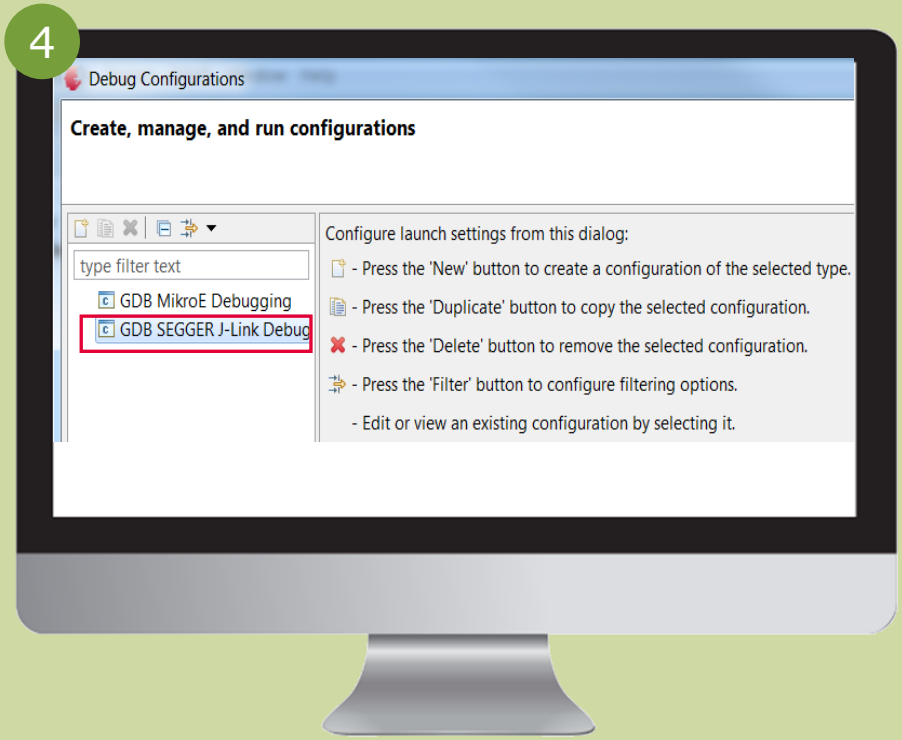
2



Compile and download the code

3 Click **Debug Configuration** 

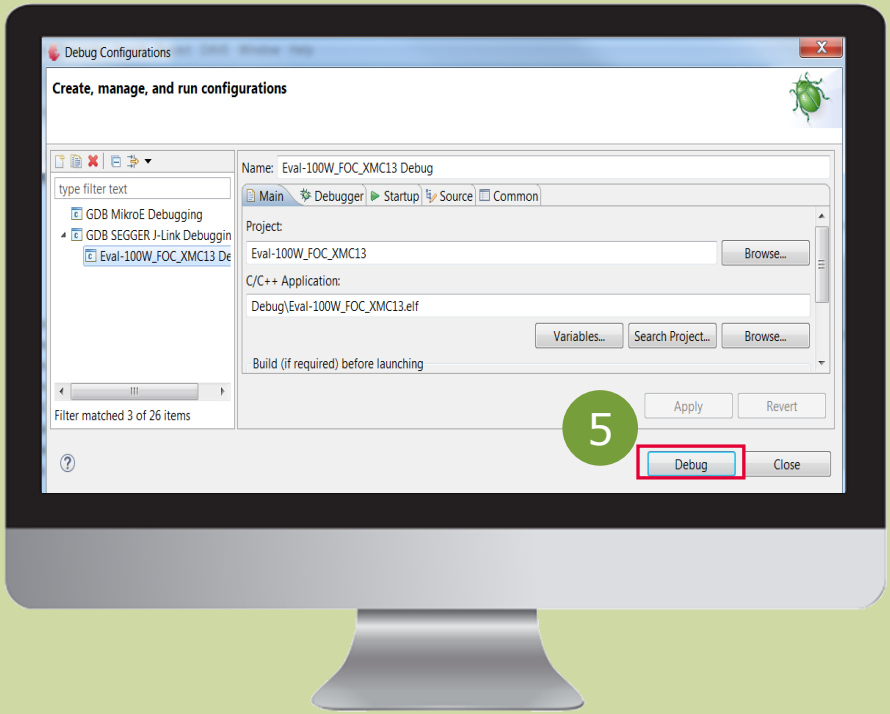
4 Select **GDB SEGGER J-Link Debugging**



Compile and download the code

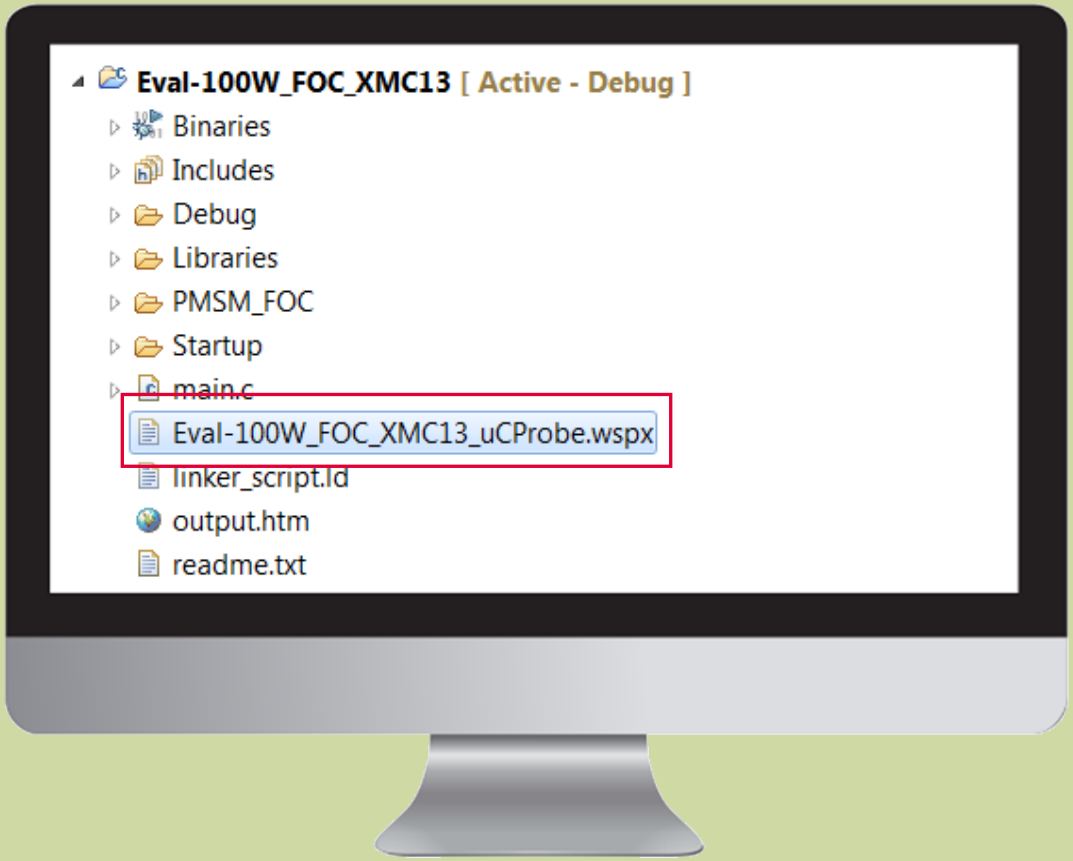
5 Click the **Debug** button to download the code

6 Click the **Resume** button to start the application



Using µC/Probe™ GUI

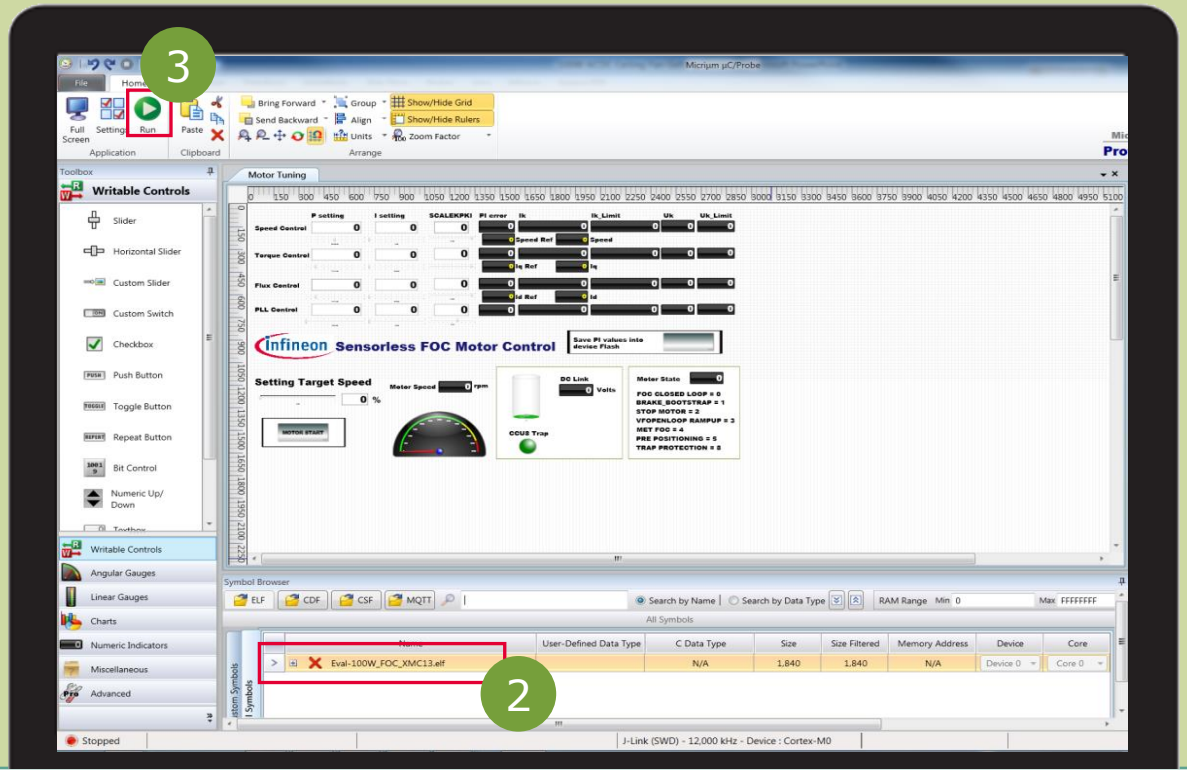
1. Open the ***.wspx µC/Probe™** project by double click on the ***.wspx** file



Using µC/Probe™ GUI

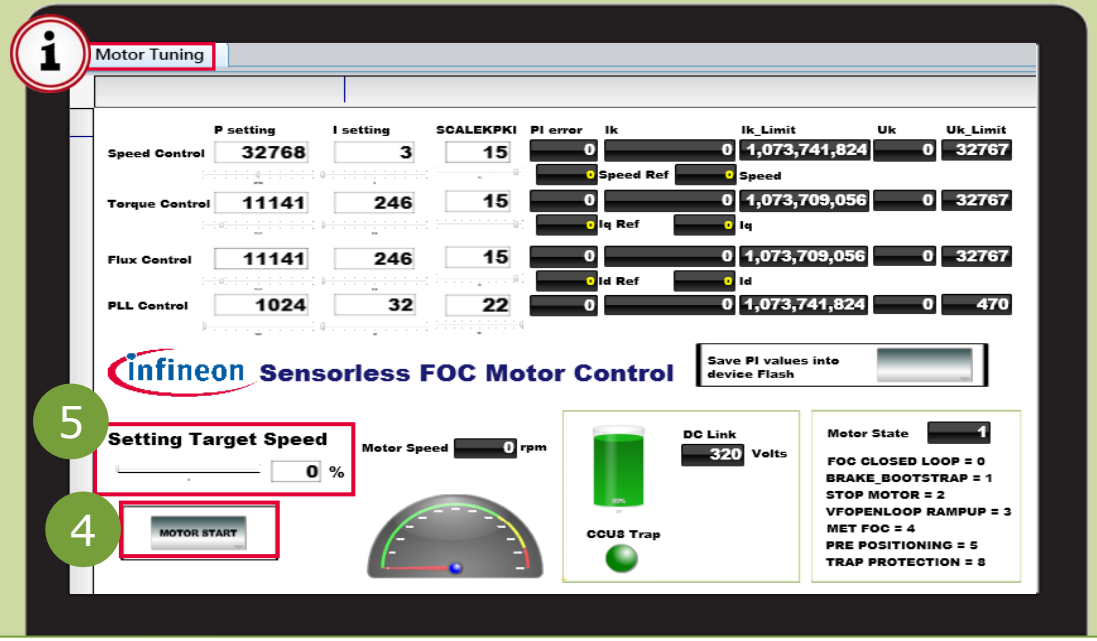
2 Check the ***.elf** file is attached to µC/Probe™ project

3 Click the **Run** button



Using µC/Probe™ GUI

The default K_P, K_I values from the pmsm_foc_motor_parameters.h file are automatically displayed in the Motor Tuning page

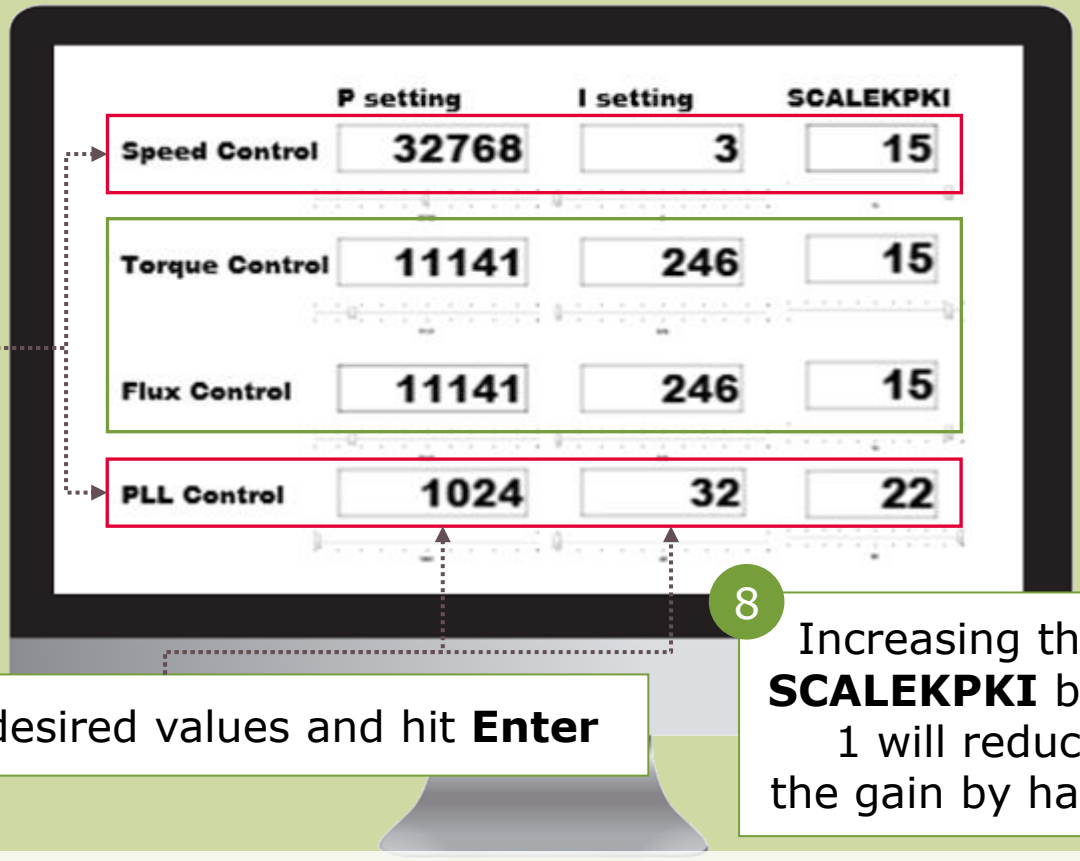


4 Click the **MOTOR START** button

5 Then use the target speed slider or key in directly to set the percentage of maximum speed desired. The motor will start to spin

Using µC/Probe™ GUI

6 Modify the **K_P**, **K_I** and **SCALEKPKI** values of the **Speed Control** and **PLL Control** loops to get optimum motor behaviour. The default K_P, K_I and SCALEKPKI values for Torque Control and Flux Control normally do not need to change



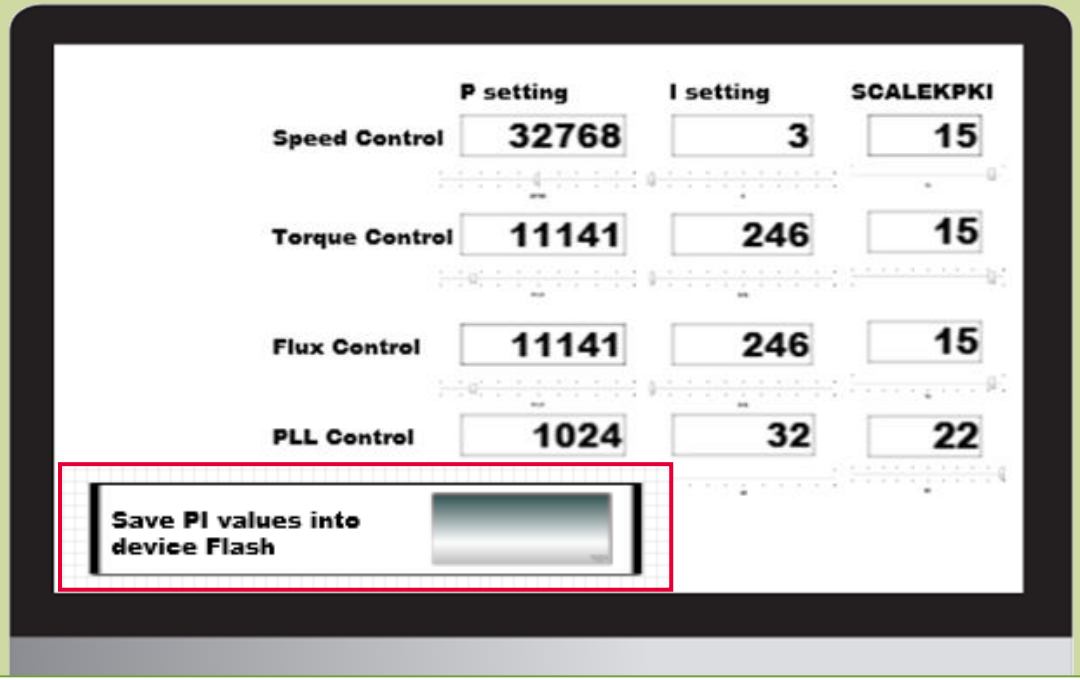
7 Type in the desired values and hit **Enter**

8 Increasing the **SCALEKPKI** by 1 will reduce the gain by half

*Under the **P setting** column are the K_P value for the 4 control loops*

*Under the **I setting** column are the K_I value for the 4 control loops*

Using µC/Probe™ GUI



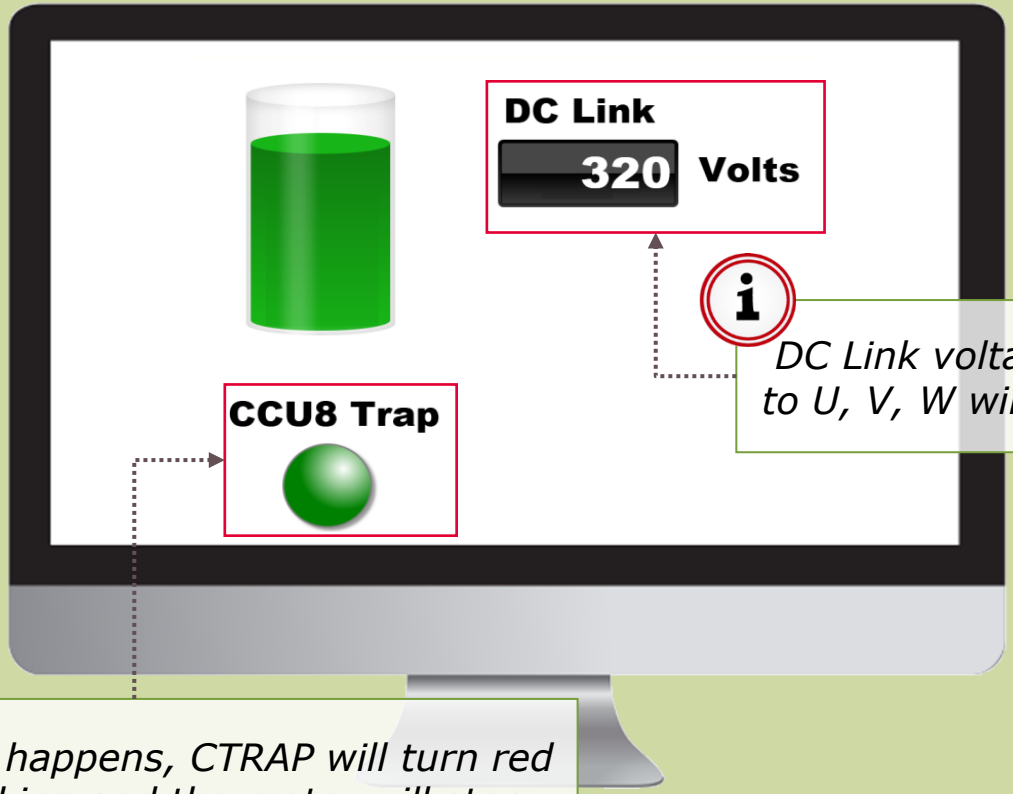
9

Click this button to save K_P, K_I, SCALEKPKI values to Flash.
The motor will stop

This saving function only works if the motor is rotating normally

Using µC/Probe™ GUI

Over-current and under-voltage protections

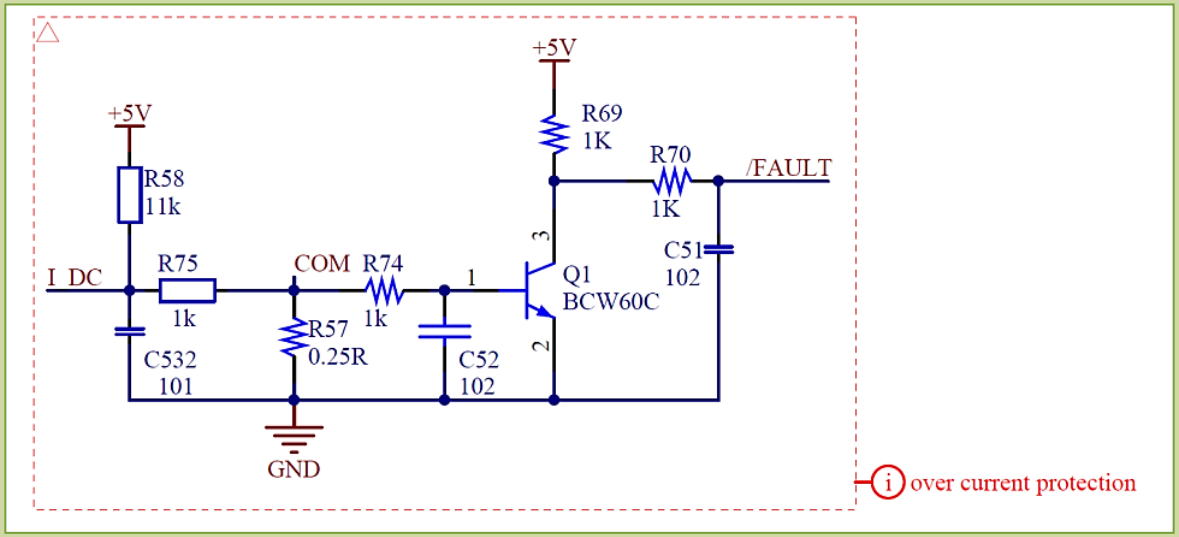


DC Link voltage, which applies to U, V, W winding of the motor

If over-current happens, CTRAP will turn red and start blinking and the motor will stop

Using µC/Probe™ GUI

Over-current and under-voltage protections



On the Eval-100W_FOC board, the VBE(th) of SS8050 is 0.6 V

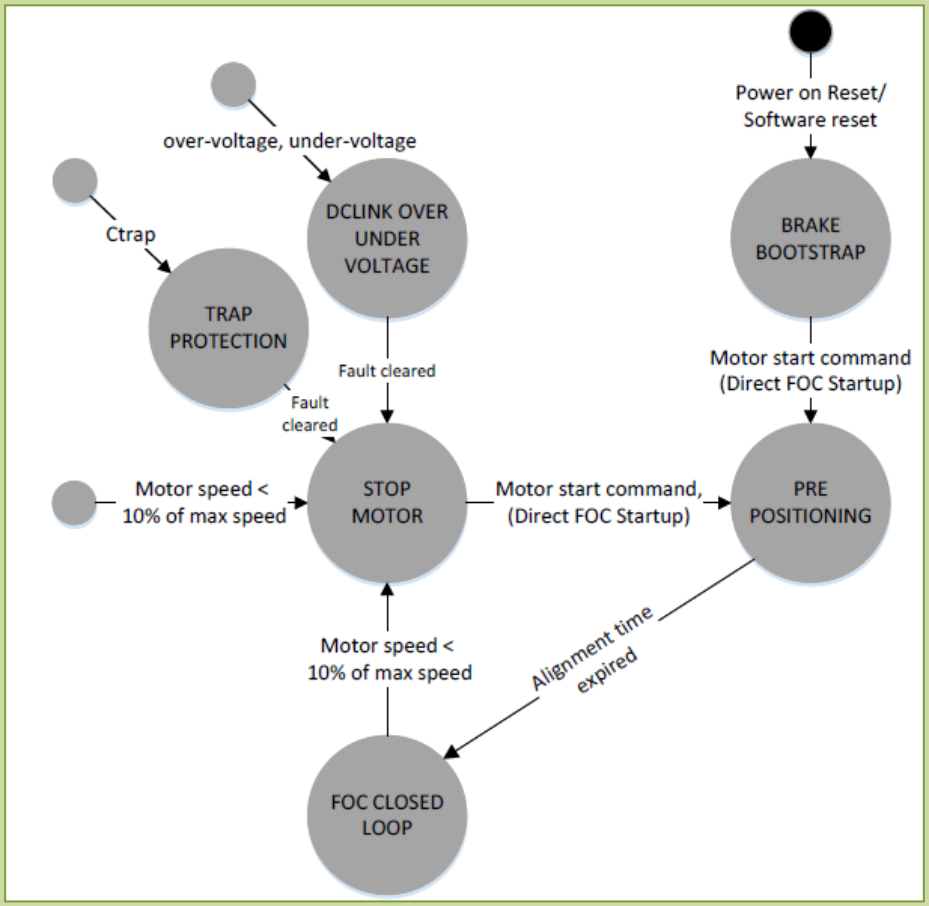
The trip current Itrip is 0.6/0.25=2.4 A

When the 3-phase motor currents exceed its trip current, the CTRAP signal is activated and over-current detected

*The power rating of the resistor R57 should be (Itrip)2*0.25*1.5=1.1 W*

Using µC/Probe™ GUI

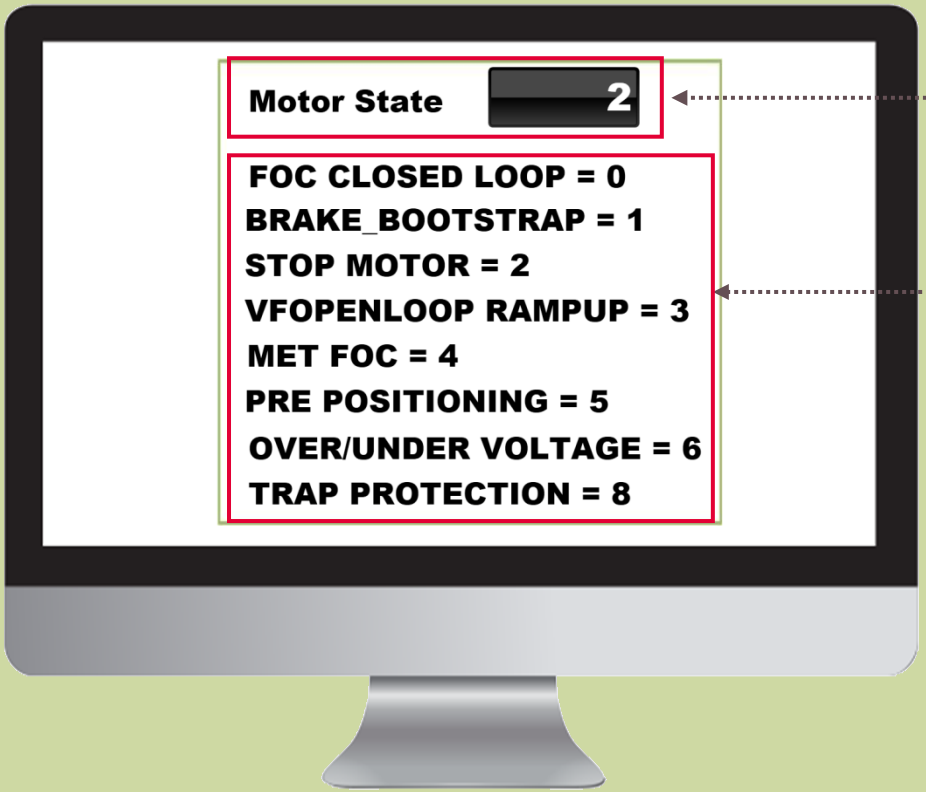
PMSM FOC state machine panel



Direct FOC startup state machine diagram

Using µC/Probe™ GUI

PMSM FOC state machine panel



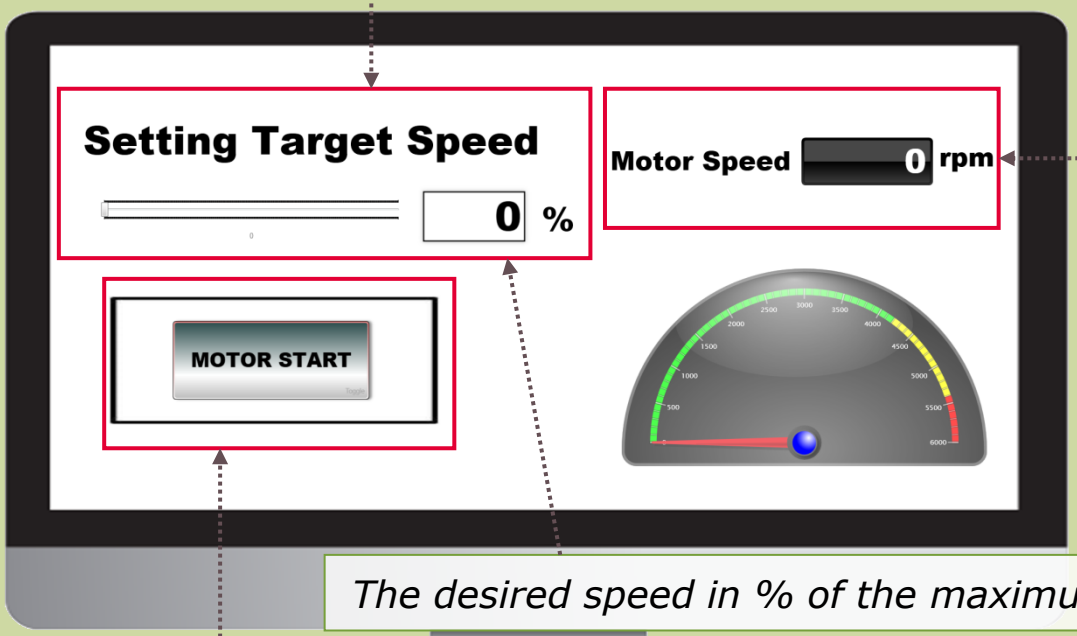
The present motor state

The various motor states

Using µC/Probe™ GUI

Motor speed, motor start/stop control panel

*Use the slider to set the desired speed (% of motor maximum speed in rpm). The motor should start to run if **MOTOR START** is clicked*

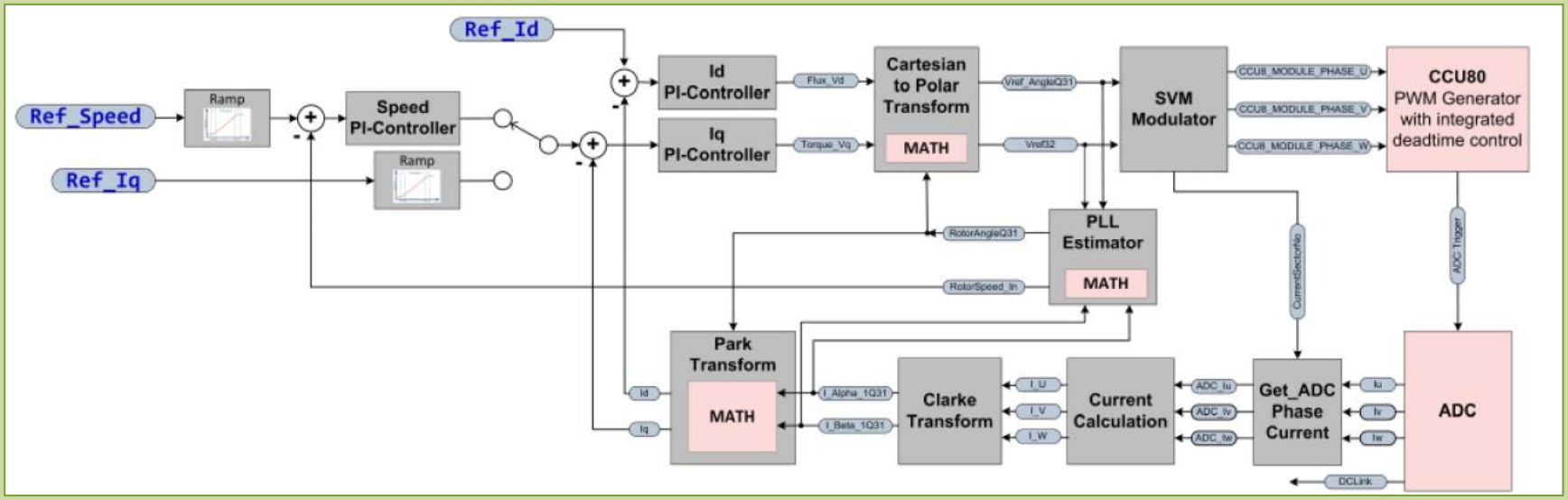


The actual speed

The desired speed in % of the maximum rpm of the motor

*1) Click this **MOTOR START** button. 2) Then, set the desired speed for the motor to start. The wording on the button will change to **MOTOR STOP**. 3) Click this button to stop the motor*

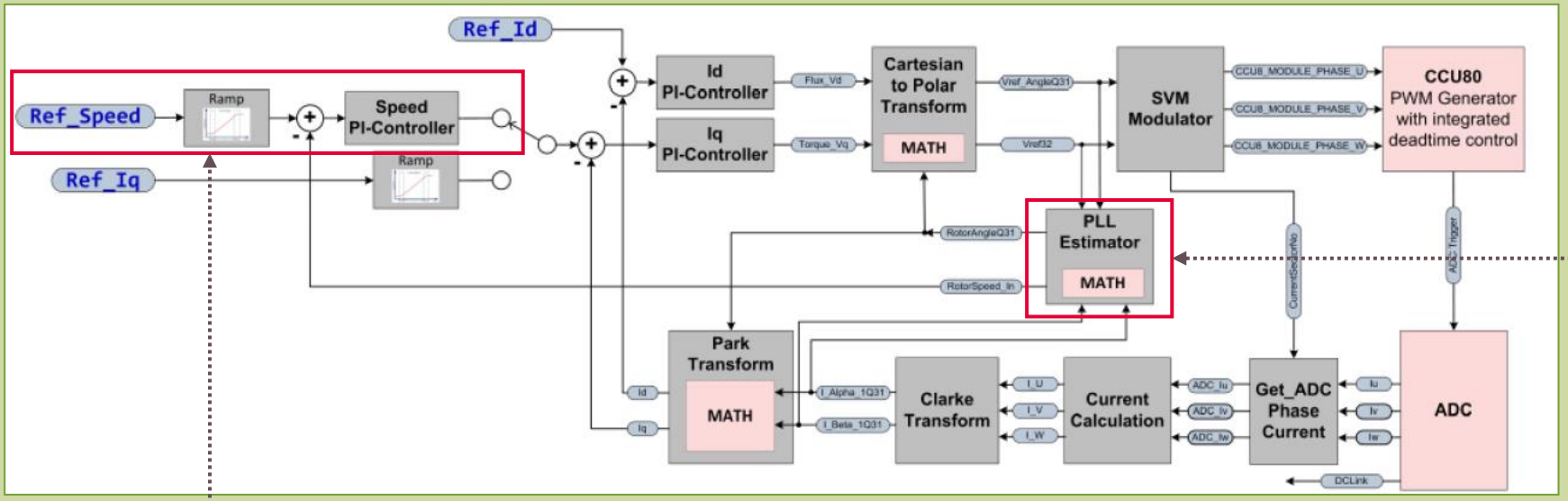
Tuning of K_P, K_I value using the µC/Probe™



PMSM FOC block diagram

*The FOC software starts the motor directly with **Speed Control Direct FOC control** scheme (by default setting)*

Tuning of K_P, K_I value using the µC/Probe™



PMSM FOC block diagram

Remove the Speed control here if using TORQUE_CONTROLLED_DIRECT_FOC scheme

Tune the PLL control loop here

*If the **MOTOR START** button was clicked and the motor does not spin after the desired speed slider is set to 30%, for example, the user should change the **My_FOC_CONTROL_SCHEME** in pmsm_foc_user_config.h to **TORQUE_CONTROLLED_DIRECT_FOC**, re-compile and reload*

So that the user can tune the K_P, K_I and SCALEKPKI of PLL, control loop first

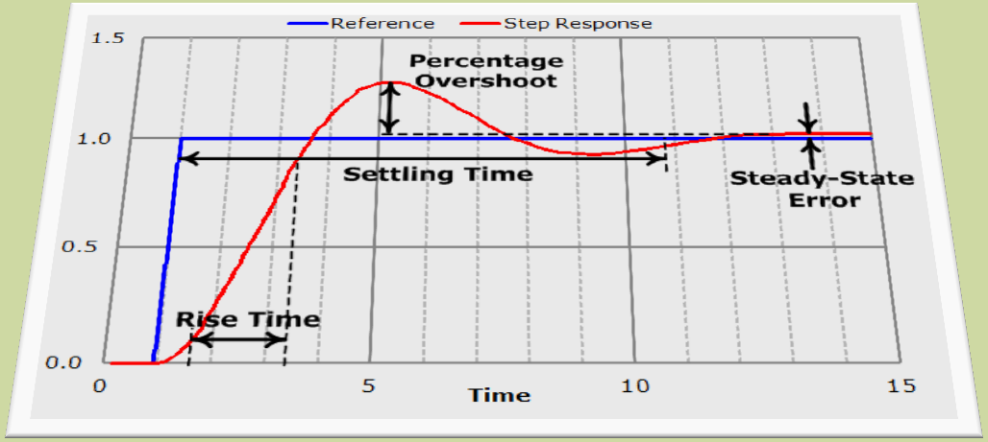
Tuning of K_P, K_I value using the µC/Probe™

Effects of increasing proportional gain K_P or integral gain K_I of PI controller independently

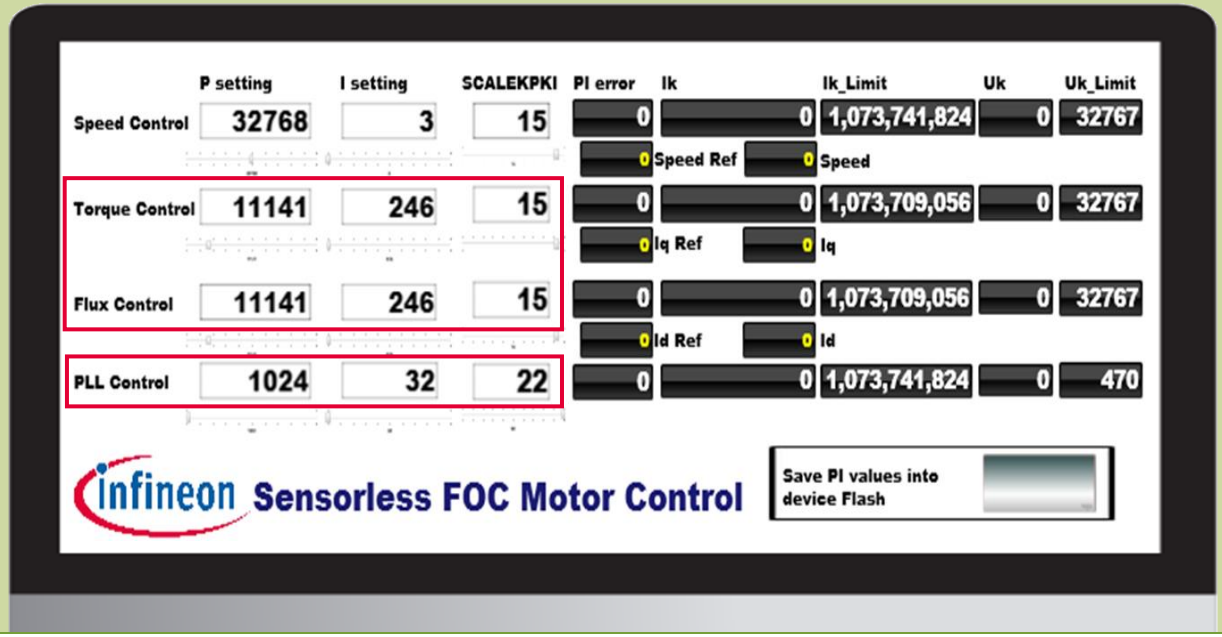
	Gain change	Effects on step response characteristics			
		Rise time	Overshoot	Settling time	Steady-state error
1	K_P ↑ K_I unchanged	↓ ☺	↑ ☹	Minor Change 😐	↓ ☺
2	K_I ↑ K_P unchanged	↓ ☺	↑ ☹	↑ ☹	Eliminate ☺

↑ Increase

↓ Decrease



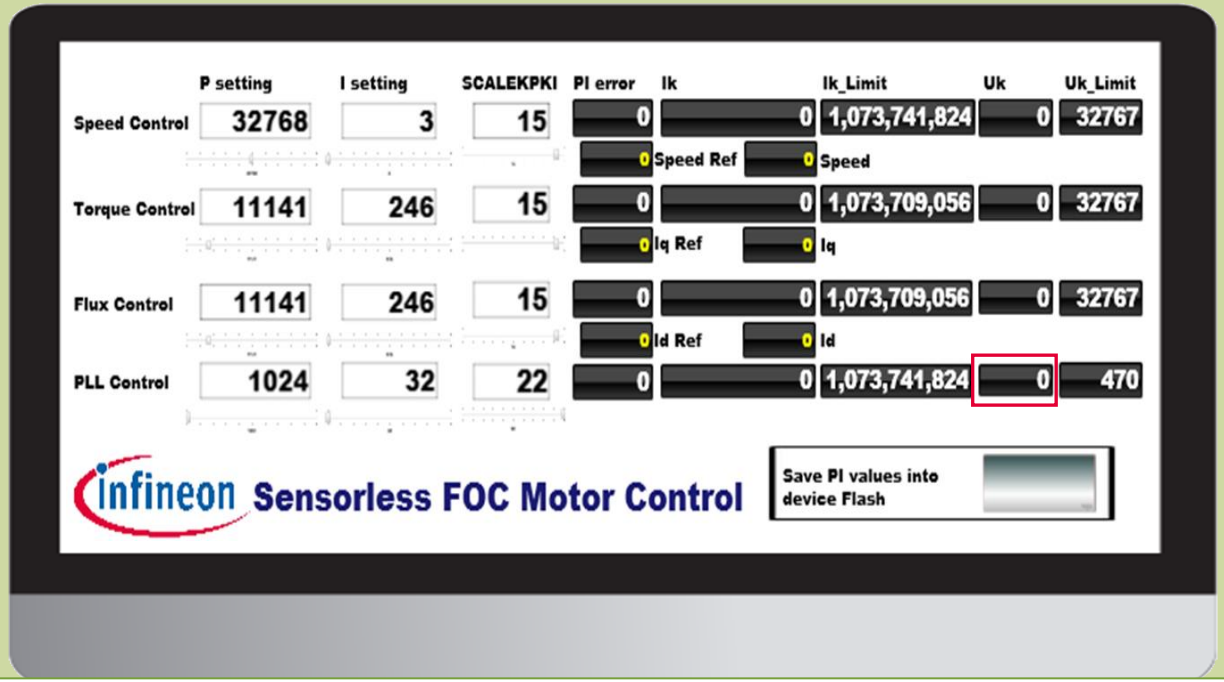
Tuning of K_P, K_I value using the µC/Probe™



1. The P setting, I setting and SCALEKPKI values for Torque and Flux PI Controllers are calculated from the physical motor and system parameters, and typically don't need to be tuned in the first iteration

2. The PLL Control parameters K_P, K_I, SCALEKPKI should start to be modified if the motor cannot startup in FOC closed-loop smoothly

Tuning of K_P, K_I value using the µC/Probe™



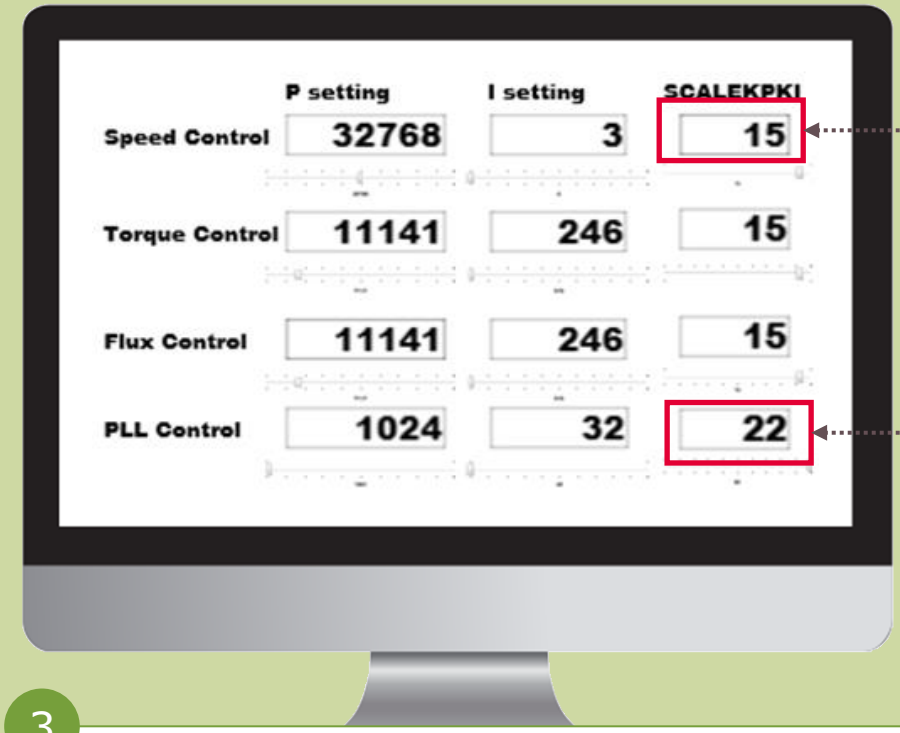
3

The target speed slider becomes the target torque slider under TORQUE_CONTROLLED_DIRECT_FOC mode

4

Make sure the Uk value of PLL control does not hit its Uk_Limit while tuning the PLL control parameters

Tuning of K_P, K_I value using the µC/Probe™



↑ *this value by 1 will* ↓ *gain of Speed controller by half*

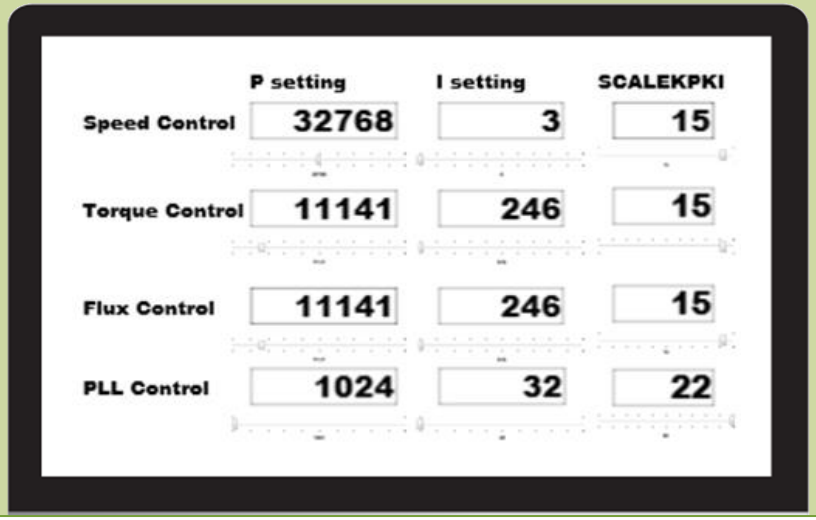
PI gains: $K_p = \frac{P\ setting}{2^{SCALEKPKI}}$, $K_i = \frac{I\ setting}{2^{SCALEKPKI}}$

↑ *this value by 1 will* ↓ *gain of PLL estimator controller by half*

3

If the motor does not spin in FOC close loop, ↑ the SCALEKPKI of PLL Control and check the motor behavior. If the motor starts to move slowly, ↑ the SCALEKPKI further. Else, ↓ the SCALEKPKI.

Tuning of K_P, K_I value using the µC/Probe™



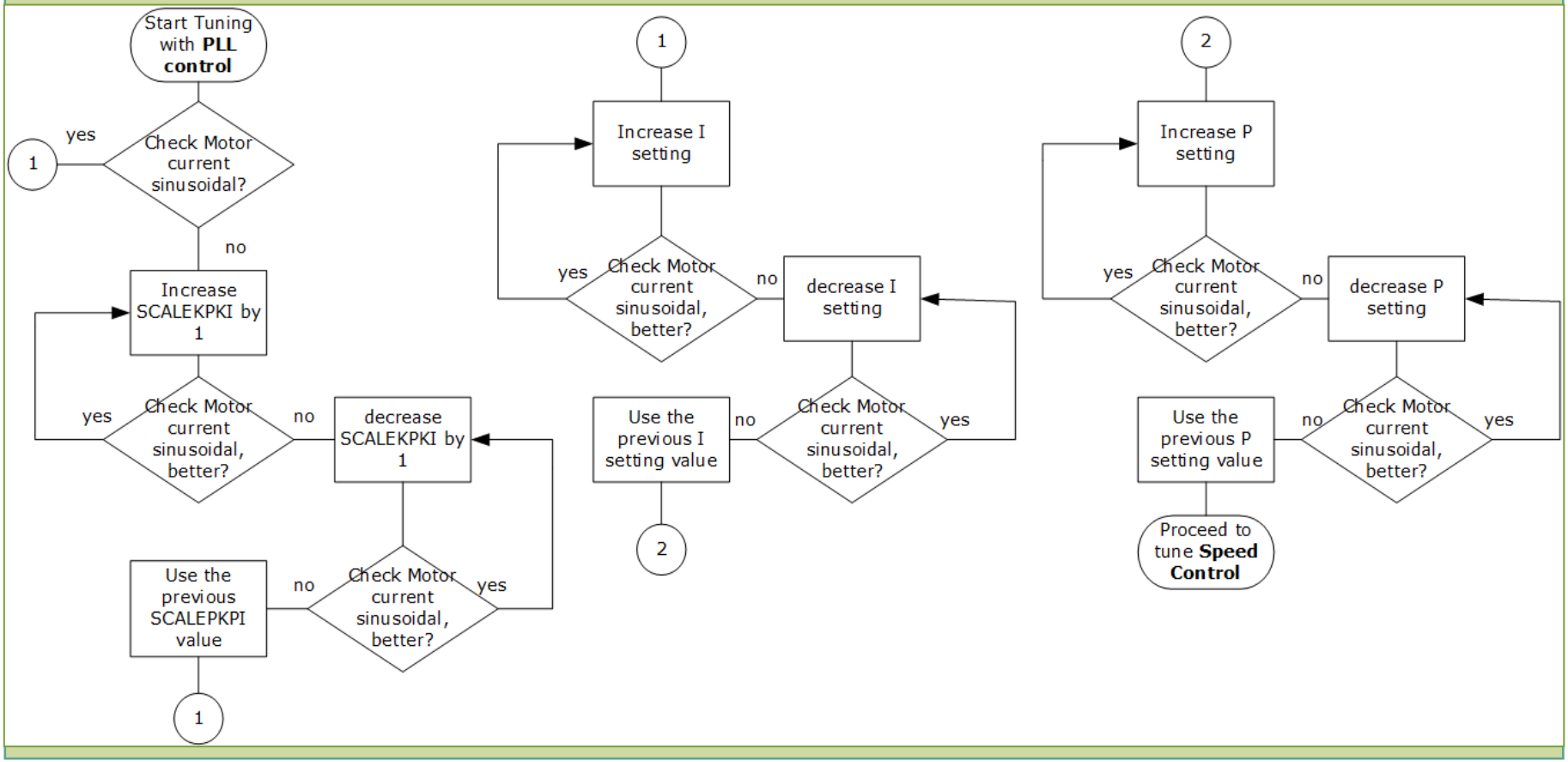
4. After the motor is able to rotate smoothly under TORQUE_CONTROLLED_DIRECT_FOC, remember to save the parameters by clicking the **Save PI** button

5. Change the My_FOC_CONTROL_SCHEME in pmsm_foc_user_config.h to SPEED_CONTROLLED_DIRECT_FOC, re-compile and reload the code

6. Use the same tactic to tune the Speed Control parameters. No need to modify the PLL control parameters

General hints on tuning of SCALEKPKI, P, I value using the µC/Probe™

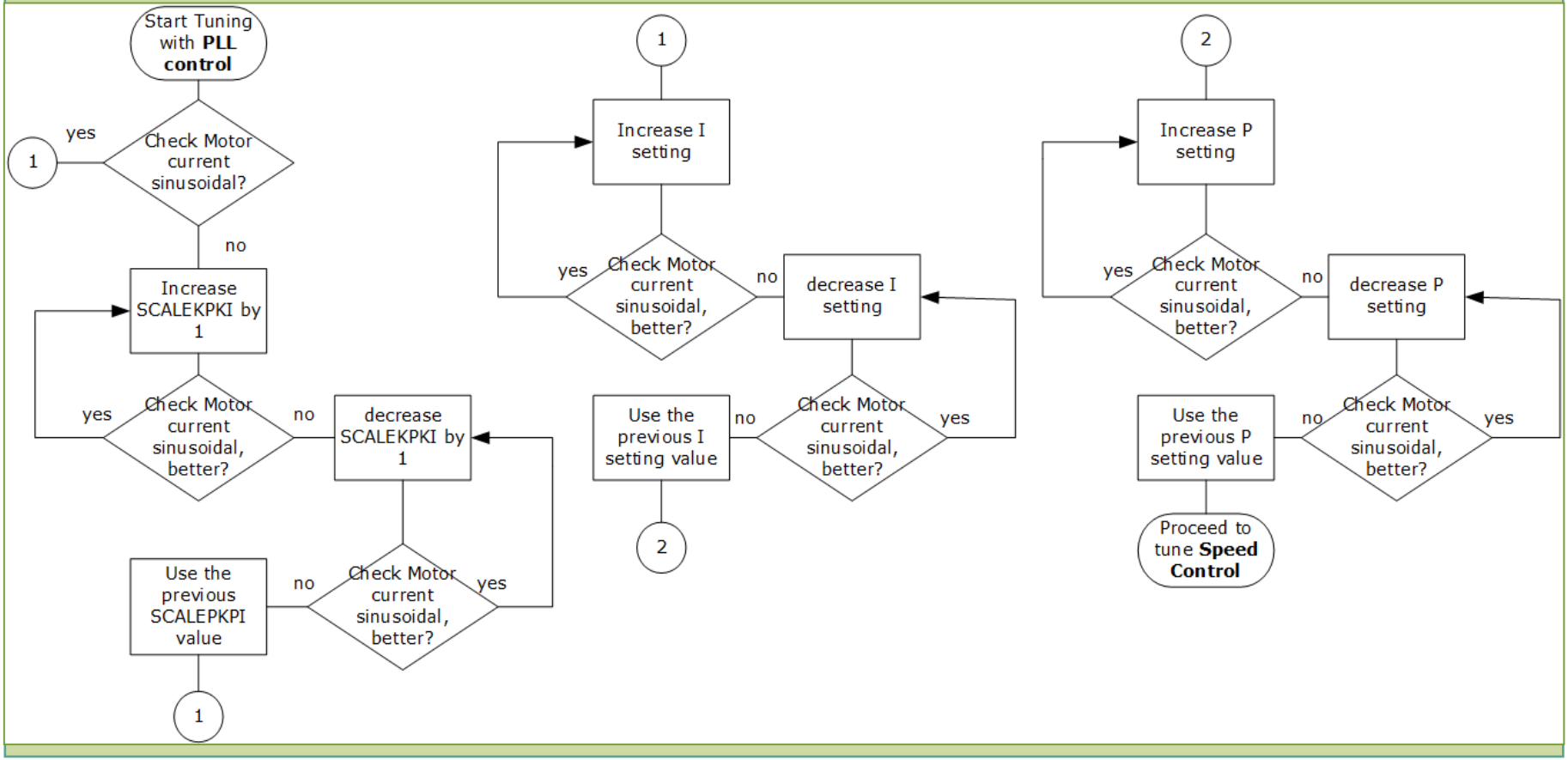
For the FOC control tuning, this is the procedure in form of an algorithm block diagram on how to tune the PLL loop for FOC control



General hints on tuning of SCALEKPKI, P, I value using the µC/Probe™

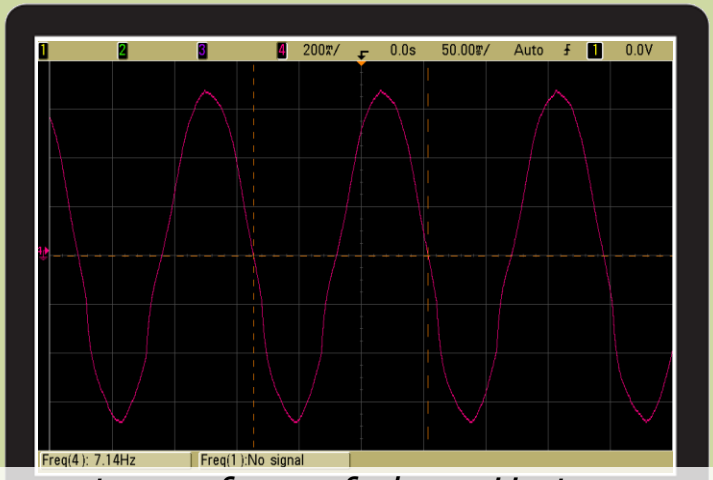
For Speed Control tuning the user should follow the same flow

Typically, only these 2 PI control loops need to be tuned with the FOC software for EVAL_100W_DRIVE_CFD2 kit

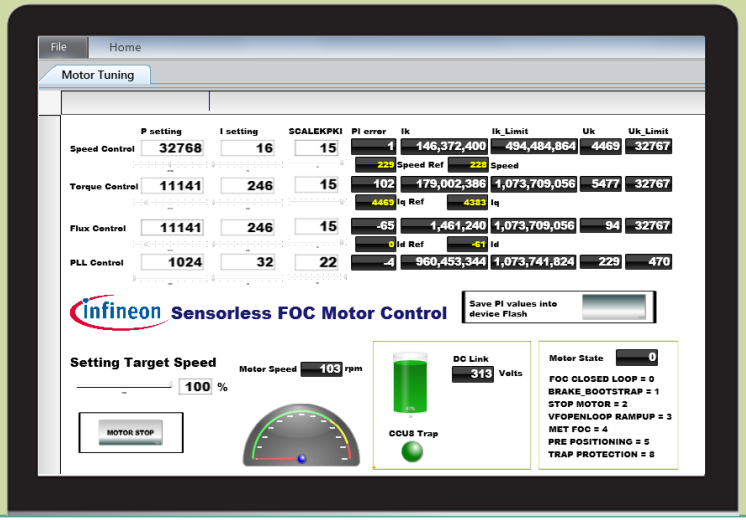


Testing result with a ceiling fan

- AC input: 230 V/50 Hz
- Ceiling Fan motor specification:
  - Resistance/phase: 23 ohms
  - Inductance/phase: 65 mH
  - Pole pairs: 4
  - Max. speed: 105 rpm
- Power consumption: 35 W @ 105 rpm



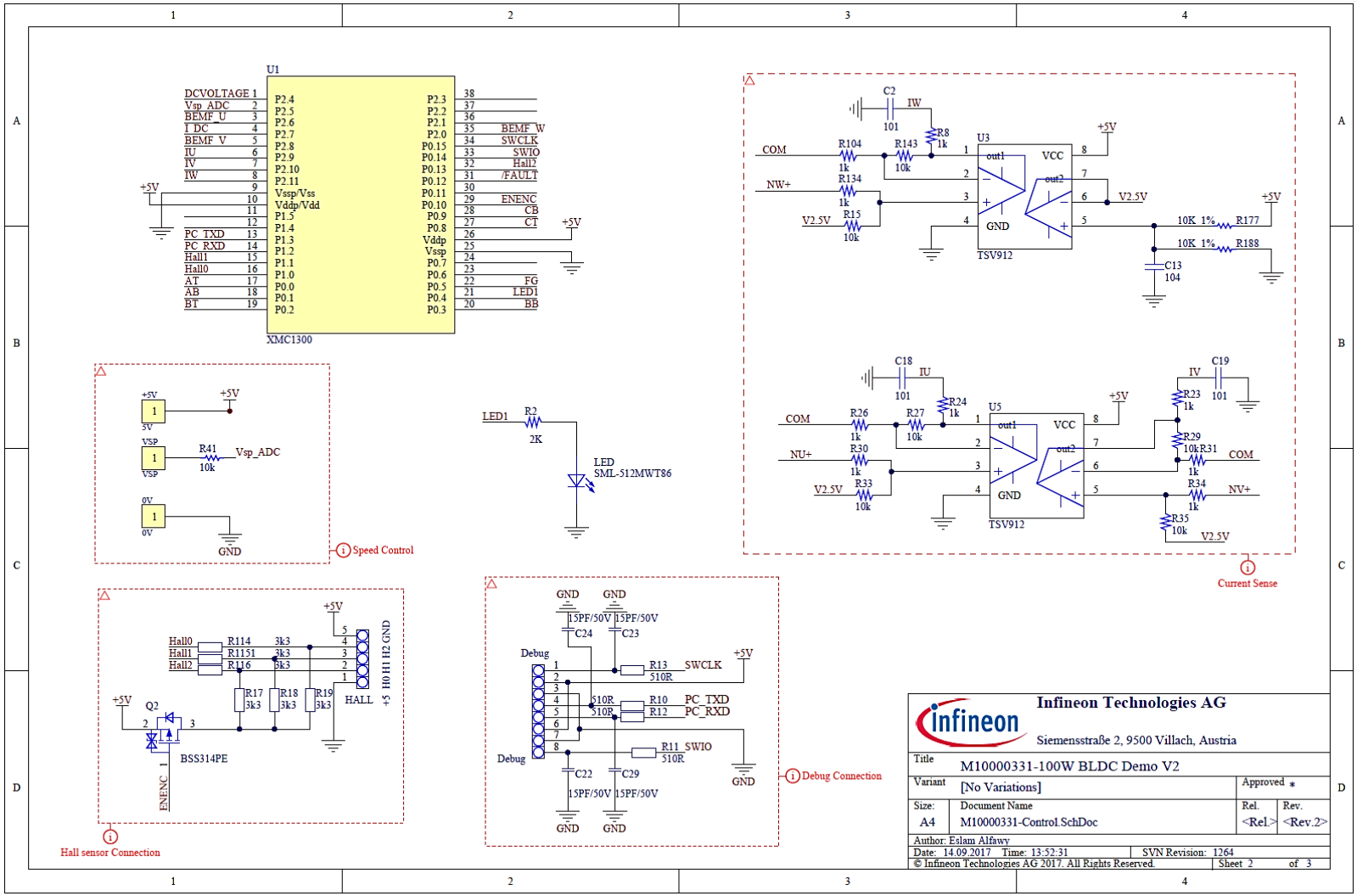
The current waveform of phase U at max. speed



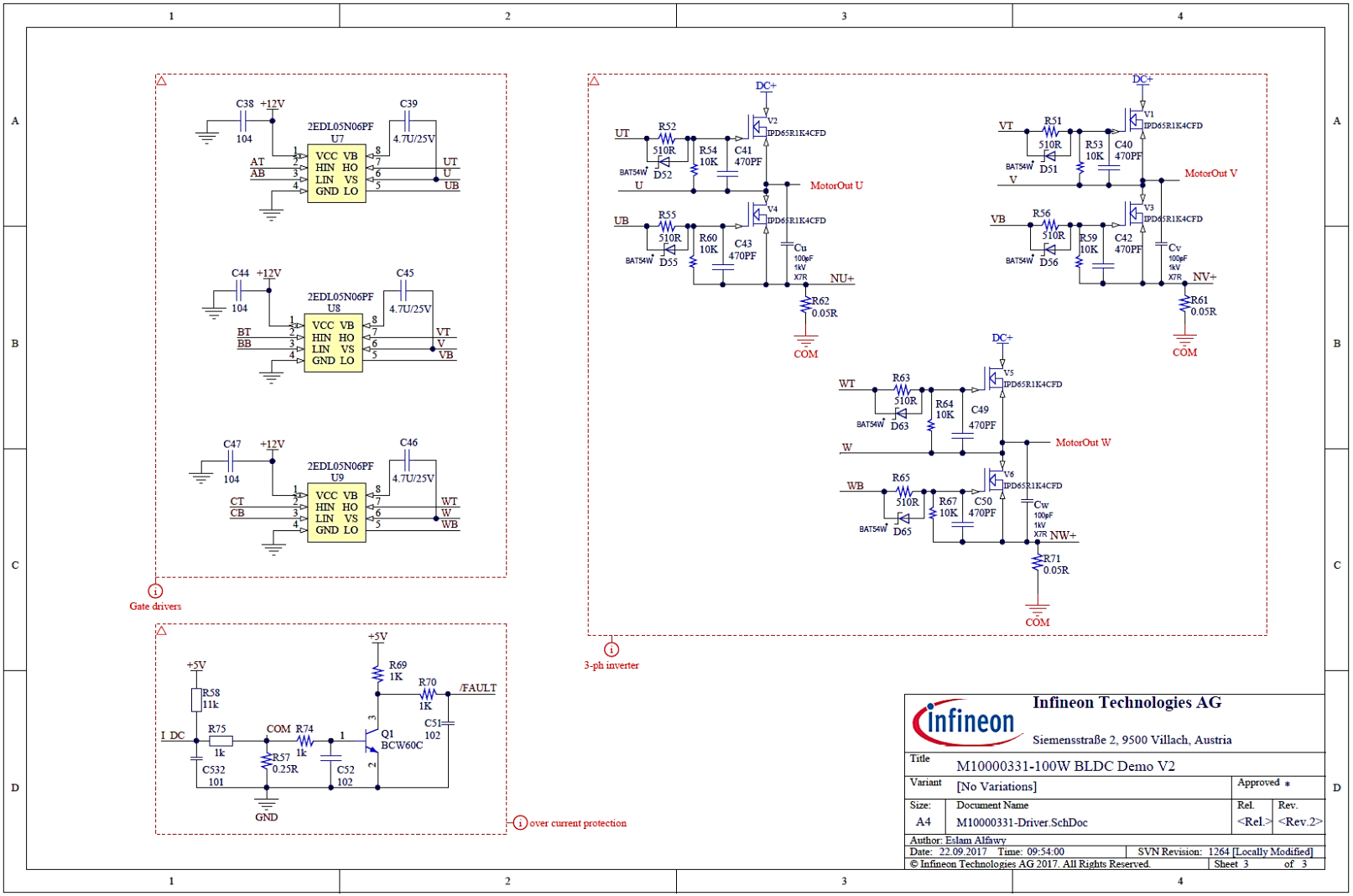
Agenda

1. 100 W motor drive evaluation board
2. Hardware and software
3. Tools
4. Getting started
5. Resources

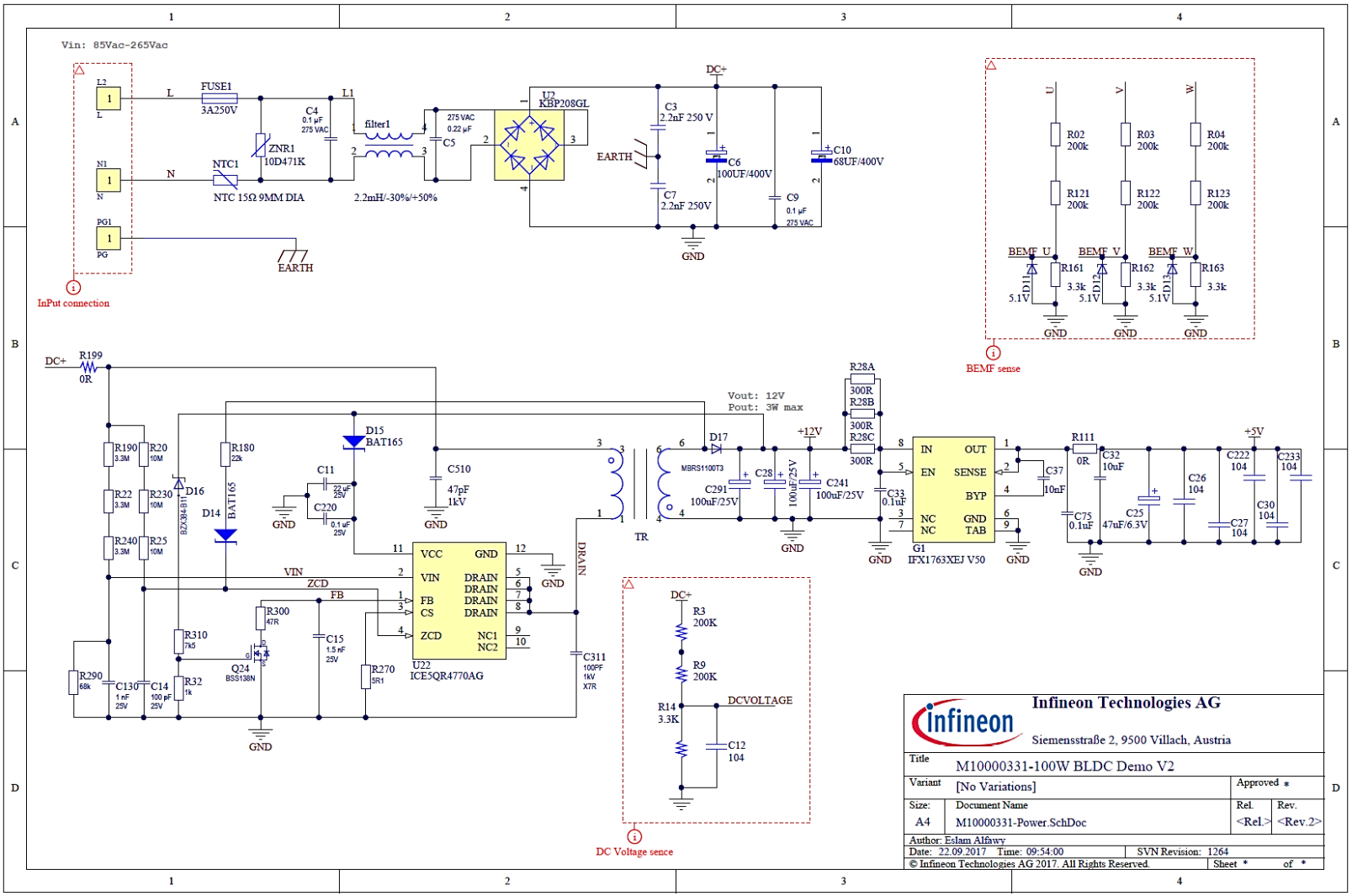
Schematic of EVAL_100W_DRIVE_CFD2 board



Schematic of EVAL_100W_DRIVE_CFD2 board



Schematic of EVAL_100W_DRIVE_CFD2 board

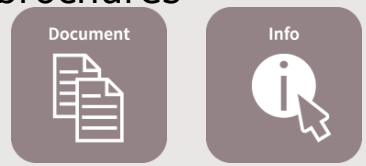
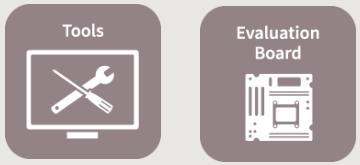


Resource Listing

- Kit documentation:
  - EVAL_100W_DRIVE_CFD2 board Appnote
  - EVAL_100W_DRIVE_CFD2 Getting Started

  https://www.infineon.com/cms/en/product/productType.html?productType=5546d4625e763904015ec84038b42ff2&redirId=59073#ispnTab4
- Application note documentation:
  - AP32370-XMC1000-PMSM FOC motor control software using XMC™

Support material:

Collaterals and brochures 	› Product briefs › Selection guides › Application brochures › Presentations › Press releases, ads	› www.infineon.com/XMC › www.infineon.com/CFD2 › www.infineon.com/COOLSET › www.infineon.com/motorcontrol
Technical material 	› Application notes › Technical articles › Simulation models › Datasheet, MCDS files › PCB design data	› www.infineon.com/XMC › Kits and Boards › DAVE™ › Software and Tool Ecosystem
Videos 	› Technical videos › Product information Videos	› Infineon Media Center › XMC Mediathek › How to measure motor parameters
Contact 	› Forums › Product support	› Infineon Forums › Technical Assistance Center (TAC)

Glossary abbreviations

Abbreviation	Full name
ADC	Analog-to-Digital Converter
FOC	Field-Oriented Control
PI Controller	Proportional–Integral Controller
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
SVM	Space Vector Modulation
XMC™	Cross-Market Microcontrollers

Disclaimer

The information given in this training materials is given as a hint for the implementation of the Infineon Technologies component only and shall not be regarded as any description or warranty of a certain functionality, condition or quality of the Infineon Technologies component.

Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this training material.

All the images used in this training are free for commercial use or free for use with attribution and were designed by Freepik.

Part of your life. Part of tomorrow.

Infineon