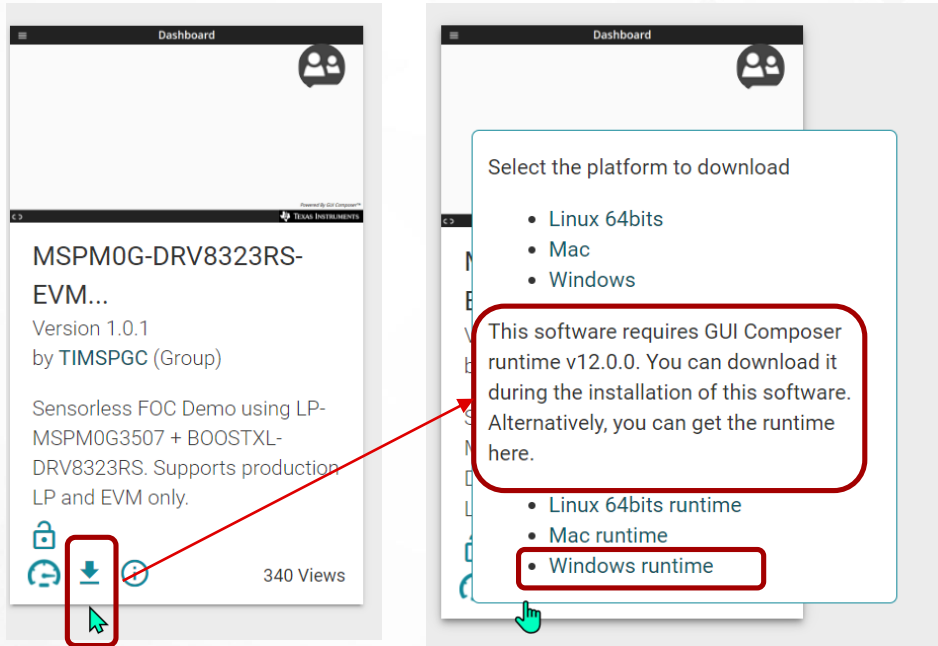


MSPM0G-DRV8323RS GUI

MSPM0G-DRV8323RS GUI

Access link:

<https://dev.ti.com/gallery/view/TIMSPGC/MSPM0G-DRV8323RS-EVM-GUI/ver/1.0.1/>



Install the software environment

MSPM0G-DRV8323RS GUI

Access link:

<https://dev.ti.com/gallery/view/TIMSPGC/MSPM0G-DRV8323RS-EVM-GUI/ver/1.0.1/>

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐ Disable Closed loop Reverse ☐

Outer Loop: Speed Motor State: IDLE

Speed Reference (Hz): 0.00 Speed (Hz):

Id Reference (A): 0.000 Id Feedback (A):

Iq Reference (A): 0.000 Iq Feedback (A):

Startup Method: IPD

IPD Thresh count: 1 IPD Freq (Hz): 50

Rampup

Rampup current (A): 0.000 Rampup speed rate (Hz/s): 0.00000

Rampup Target (Hz): 0.00

Tuning

Speed Kp: 0.000000 Speed Ki: 0.000000

Id Kp: 0.000000 Id Ki: 0.000000

Iq Kp: 0.000000 Iq Ki: 0.000000

CL

PI Speed Divider: 10

CL SpeedRef ramp (Hz/s): 0.00000

Monitors

VDC (V):

Motor & Drive Parameters

R (ohms): 0.00000 L (H): 0.000000 KE (V/Hz): 0.000000

Max Freq (Hz): 25 PWM Freq (Hz): 10000 Deadband (ns): 100

CSA Gain: 0

Protection

Over Current Limit (A): 0 Over Voltage Limit (V): 0 Under Voltage Limit (V): 0

Fault Status **CLEAR FAULTS**

Ext Fault ☐ OC ☐ OV ☐ UV ☐

Speed (Hz)



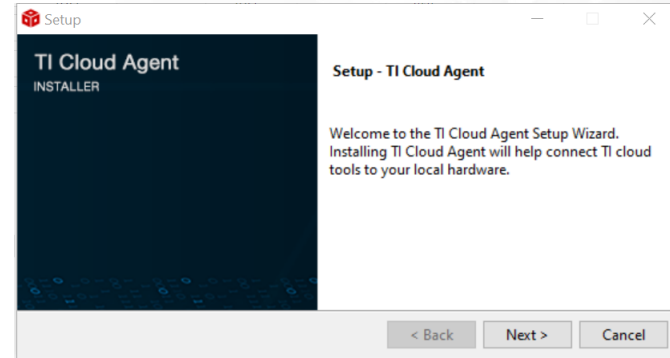
Id Current (A)



Iq Current (A)



Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)



Might need install the TI Cloud Agent if this page shows the download request.

MSPM0G-DRV8323RS GUI

Introduction

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐

Outer Loop Speed

Speed Reference (Hz) 0.00

Id Reference (A) 0.000

Iq Reference (A) 0.000

Disable Closed loop Reverse ☐

Motor State IDLE

Speed (Hz)

Id Feedback (A)

Iq Feedback (A)

Motor & Drive Parameters

R (ohms) 0.00000 L (H) 0.000000 KE (V/Hz) 0.000000

Max Freq (Hz) 25 PWM Freq (Hz) 10000 Deadband (ns) 100

CSA Gain 0

Fault Status

CLEAR FAULTS

Ext Fault OC OV UV

Startup

Method IPD

IPD Thresh count 1 IPD Freq (Hz) 50

Rampup

Rampup current (A) 0.000 Rampup speed rate (Hz/s) 0.00000

Rampup Target (Hz) 0.00

Tuning

Speed Kp 0.000000 Speed Ki 0.000000

Id Kp 0.000000 Id Ki 0.000000

Iq Kp 0.000000 Iq Ki 0.000000

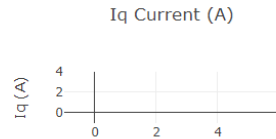
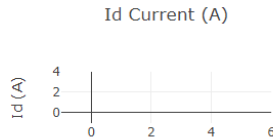
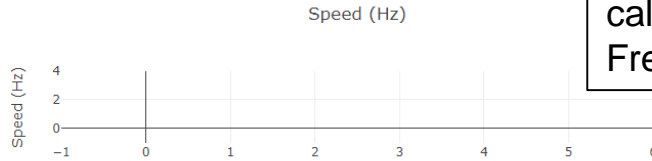
CL

PI Speed Divider 10

CL SpeedRef ramp (Hz/s) 0.00000

Monitors

VDC (V)



Motor Params.

The KE is important for the ESMO calculated paras.
Freq is the electrical frequency.

Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)

TEXAS INSTRUMENTS

MSPM0G-DRV8323RS GUI

Introduction

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐ Disable Closed loop Reverse ☐

Outer Loop

Speed Reference (Hz)

Id Reference (A)

Iq Reference (A)

Motor State

Speed (Hz)

Id Feedback (A)

Iq Feedback (A)

Motor & Drive Parameters

R (ohms) L (H)

Max Freq (Hz) PWM Freq (Hz)

Deadband (ns)

CSA Gain

Fault Status

CLEAR FAULTS

Ext Fault ☐ OC ☐ OV ☐ UV ☐

Protection

Over Current Limit (A) Over Voltage Limit (V) Under Voltage Limit (V)

Startup

Method

IPD Thresh count IPD Freq (Hz)

Rampup

Rampup current (A)

Rampup speed rate (Hz/s)

Rampup Target (Hz)

Tuning

Speed Kp Speed Ki

Id Kp Id Ki

Iq Kp Iq Ki

CL

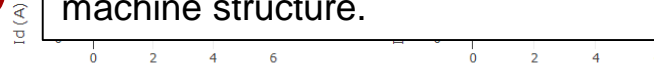
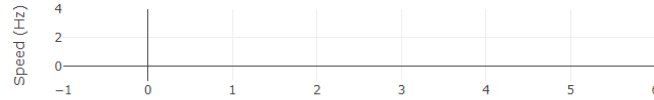
PI Speed Divider

CL SpeedRef ramp (Hz/s)

Monitors

VDC (V)

Speed (Hz)



Regulator Params.

The current loop params are related to motor params.
The speed loop params are related to motor, load, and machine structure.

Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)

TEXAS INSTRUMENTS

MSPM0G-DRV8323RS GUI

Introduction

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐

Disable Closed loop Reverse ☐

Outer Loop	Speed	Motor State	IDLE
Speed Reference (Hz)	0.00	Speed (Hz)	
Id Reference (A)	0.000	Id Feedback (A)	
Iq Reference (A)	0.000	Iq Feedback (A)	

Startup

Method IPD

IPD Threshold count 1 IPD Freq (Hz) 50

Rampup

Rampup current (A)	0.000	Rampup speed rate (Hz/s)	0.00000
Rampup Target (Hz)	0.00		

Tuning

Speed Kp	0.000000	Speed Ki	0.000000
Id Kp	0.000000	Id Ki	0.000000
Iq Kp	0.000000	Iq Ki	0.000000

CL

PI Speed Divider 10
CL SpeedRef ramp (Hz/s) 0.00000

Monitors

VDC (V)

Motor & Drive Parameters

R (ohms)	0.00000	L (H)	0.000000	KE (V/Hz)	0.000000
Max Freq (Hz)	25	PWM Freq (Hz)	10000	Deadband (ns)	100
CSA Gain	0				

Protection

Over Current Limit (A)	0	Over Voltage Limit (V)	0	Under Voltage Limit (V)	0
------------------------	---	------------------------	---	-------------------------	---

Fault Status

CLEAR FAULTS

Ext Fault OC OV UV

Control Params.

User can select the control loop and set the reference input for the loop.

User can select the start up method for different usage. Rampup stage can set the acceleration and loading current.

Id Current (A)

Iq Current (A)


User can set the speed loop divider – how many times current loop running with a speed loop running.

Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)

TEXAS INSTRUMENTS

MSPM0G-DRV8323RS GUI

Introduction

 MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐

Outer Loop

Speed

Speed Reference (Hz)

0.00

Id Reference (A)

0.000

Iq Reference (A)

0.000

Disable Closed loop Reverse ☐

Motor State

IDLE

Speed (Hz)

Id Feedback (A)

Iq Feedback (A)

Startup

Method

IPD

IPD Threshold count

1

IPD Freq (Hz)

50

Rampup

Rampup current (A)

0.000

Rampup Target (Hz)

0.00

Tuning

Speed Kp

0.000000

Id Kp

0.000000

Iq Kp

0.000000

Speed Ki

0.000000

Id Ki

0.000000

Iq Ki

0.000000

CL

PI Speed Divider

10

CL SpeedRef ramp (Hz/s)

0.00000

Monitors

VDC (V)

Motor & Drive Parameters

R (ohms)

0.00000

L (H)

0.000000

KE (V/Hz)

0.000000

Max Freq (Hz)

25

PWM Freq (Hz)

10000

Deadband (ns)

100

CSA Gain

0

Protection

Over Current Limit (A)

0

Over Voltage Limit (V)

0

Under Voltage Limit (V)

0

Fault Status

CLEAR FAULTS


Ext Fault ☐

OC ☐


OV ☐

UV ☐

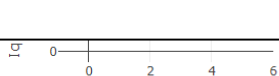
Speed (Hz)



Id Current (A)



Iq




Fault status monitor.

It will show if there is any out of the protection limit or DRV ext fault occurs, and then stop the motor.


Click on the red button will clear the fault status if it doesn't exist.

Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)

 TEXAS INSTRUMENTS

16

TI Information – Selective Disclosure

 TEXAS INSTRUMENTS

MSPM0G-DRV8323RS GUI

Introduction

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☐

Outer Loop

Speed Reference (Hz)

Id Reference (A)

Iq Reference (A)

Disable Closed loop Reverse ☐

Motor State

Speed (Hz)

Id Feedback (A)

Iq Feedback (A)

Motor & Drive Parameters

R (ohms)

Max Freq (Hz)

CSA Gain

L (H)

PWM Freq (Hz)

Protection

Over Current Limit (A)

Over Voltage Limit (V)

Fault Status

CLEAR FAULTS

Watch Window

User can timely get the current motor status, including the speed, d-axis current, q-axis current.

Startup Method

IPD Thresh count

IPD Freq (Hz)

Rampup

Rampup current (A)

Rampup speed rate (Hz/s)

Rampup Target (Hz)

Tuning

Speed Kp

Speed Ki

Id Kp

Id Ki

Iq Kp

Iq Ki

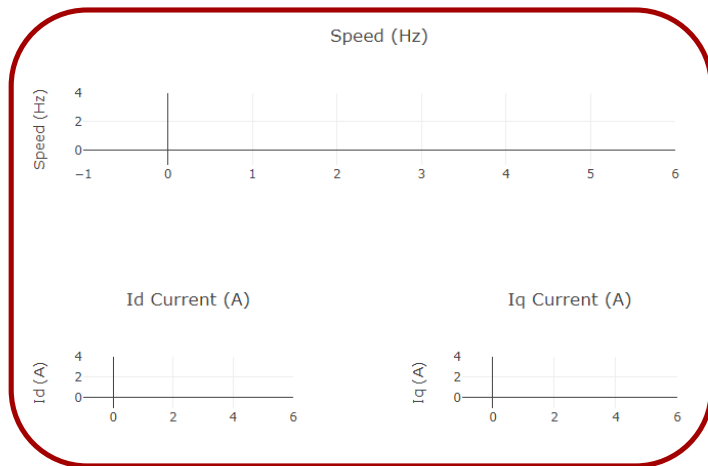
CL

PI Speed Divider

CL SpeedRef ramp (Hz/s)

Monitors

VDC (V)



Error: Failed to connect: CS_DAP_0: Error initializing emulator: (Error -260 @ 0x0)

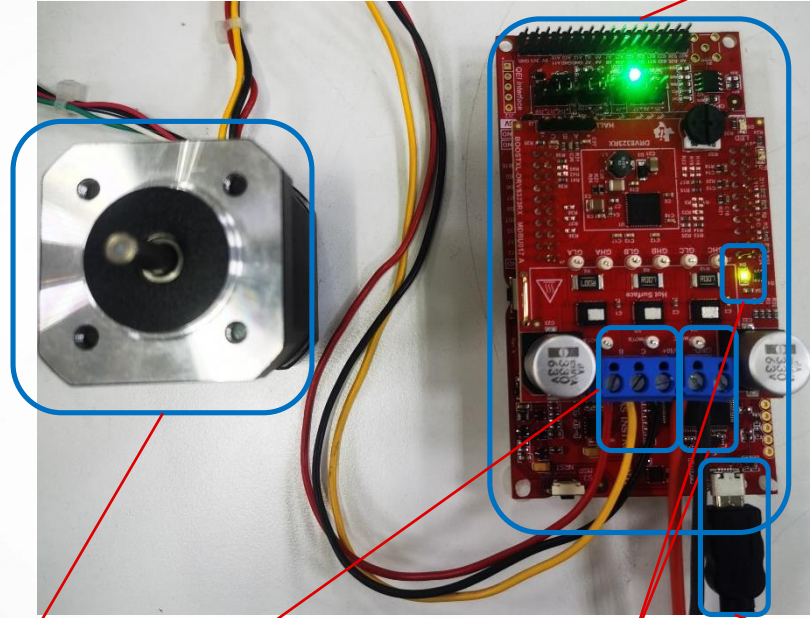
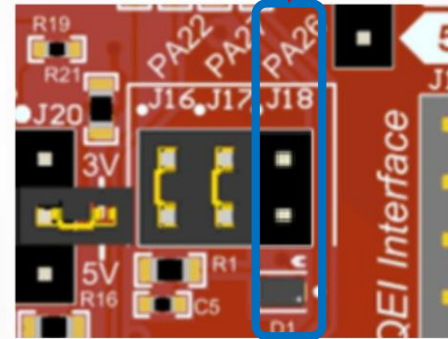
TEXAS INSTRUMENTS

Step by Step Operations - GUI

Hardware Setup

MSPM0G3507 LaunchPad +
DRV8323RS EVM

Remove the PA26 jumper J18 in the
LP-MSPM0G3507 as shown below



Motor
Motor Phase U/V/W

Power supply 4.5~34V
Turn on the power supply. The green VM LED
on the BOOSTXL-DRV8323RS should turn on.

Connect USB to the PC for
software download and debug

Step2. Software setting

2-1. Connect the USB interface of LP-3507 with PC

Note:

Please power on firstly (bus dc power) and then load the program or connect board.

If connection error, please refresh the page and it will automatically reconnect.



Hardware connected.

Connection success

Step2. Software setting

2-2. Set the correct parameters of the user motor and tuning params.

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☒

Outer Loop Speed

Speed Reference (Hz) 160.00

Id Reference (A) 0.000

Iq Reference (A) 0.000

Disable Closed loop Reverse ☐

Motor State STANDBY

Speed (Hz) 0.00

Id Feedback (A) 0.000

Iq Feedback (A) 0.000

Motor & Drive Parameters

R (ohms) 0.37000

L (H) 0.000220

KE (V/Hz) 0.009867

Max Freq (Hz) 400

PWM Freq (Hz) 20000

Deadband (ns) 400

CSA Gain 0

Fault Status

CLEAR FAULTS

Ext Fault

OC

OV

UV

Startup

Method Align

AlignTime (sec) 0.500

Align Current (A) 2.000

Rampup

Rampup current (A) 2.000

Rampup speed rate (Hz/s) 20.00000

Rampup Target (Hz) 80.00

Tuning

Speed Kp 0.083000

Speed Ki 0.001000

Id Kp 0.070000

Id Ki 0.012000

Iq Kp 0.070000

Iq Ki 0.012000

CL

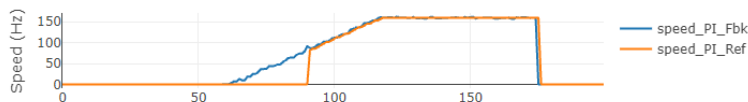
PI Speed Divider 5

CL SpeedRef ramp (Hz/s) 20.00000

Monitors

VDC (V) 11.728

Speed (Hz)



PI tuning regulator paras.

It is related to motor, load, and machine structure.



Step2. Software setting

2-3. Set the appropriate control method and reference input.

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☒

Disable Closed loop Reverse ☒

Outer Loop Motor State

Speed Reference (Hz)

Id Reference (A)

Iq Reference (A)

Speed (Hz)

Id Feedback (A)

Iq Feedback (A)

Startup

Method

AlignTime (sec)

Align Current (A)

Rampup

Rampup current (A)

Rampup speed rate (Hz/s)

Rampup Target (Hz)

Tuning

Speed Kp

Id Kp

Iq Kp

Speed Ki

Id Ki

Iq Ki

CL

PI Speed Divider

CL SpeedRef ramp (Hz/s)

Monitors

VDC (V)

Motor & Drive Parameters

R (ohms)

Max Freq (Hz)

CSA Gain

L (H)

PWM Freq (Hz)

KE (V/Hz)

Deadband (ns)

Protection

Over Current Limit (A)

Over Voltage Limit (V)

Under Voltage Limit (V)

Fault Status

CLEAR FAULTS

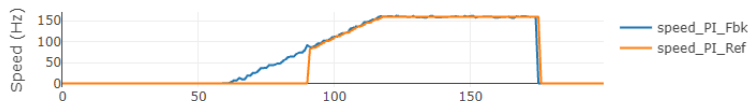
Ext Fault ☐

OC ☐

OV ☐

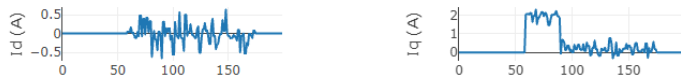
UV ☐

Speed (Hz)



PI tuning regulator paras.

It is related to motor, load, and machine structure.



Step3. Spin the motor

Here is the example select closed loop with speed loop, which means output of speed loop is the input of the current loop. Then click the **button** marked below to spin the motor

MSPM0G-DRV8323RS-EVM-GUI

Control

Enable Motor ☒ (button marked with red box and arrow)

Outer Loop: Speed

Speed Reference (Hz): 160.00

Id Reference (A): 0.000

Iq Reference (A): 0.000

Motor & Drive Parameters

Disable Closed loop ☒ Reverse ☒

Motor State: STANDBY

R (ohms): 0.37000 L (H): 0.000220 KE (V/Hz): 0.009867

Max Freq (Hz): 400 PWM Freq (Hz): 20000 Deadband (ns): 400

CSA Gain: 0

Protection

Over Current Limit (A): 12 Over Voltage Limit (V): 40 Under Voltage Limit (V): 7

Fault Status CLEAR FAULTS

Ext Fault OC OV UV

Startup Method: Align AlignTime (sec): 0.500 Align Current (A): 2.000

Rampup

Rampup current (A): 2.000 Rampup speed rate (Hz/s): 20.00000

Rampup Target (Hz): 80.00

Tuning

Speed Kp: 0.083000 Speed Ki: 0.001000

Id Kp: 0.070000 Id Ki: 0.012000

Iq Kp: 0.070000 Iq Ki: 0.012000

CL

PI Speed Divider: 5

CL SpeedRef ramp (Hz/s): 20.00000

Monitors

VDC (V): 11.728

Speed (Hz) Graph

Graph showing Speed (Hz) vs Time (s). The speed reference (speed_PI_Ref, orange line) is a step function from 0 to 160 Hz. The speed feedback (speed_PI_Fbk, blue line) follows the reference. The graph is divided into three stages: Rampup stage (0 to 80 Hz), Closed loop stage – 160Hz (80 to 160 Hz), and Align + Rampup stage (0 to 80 Hz).

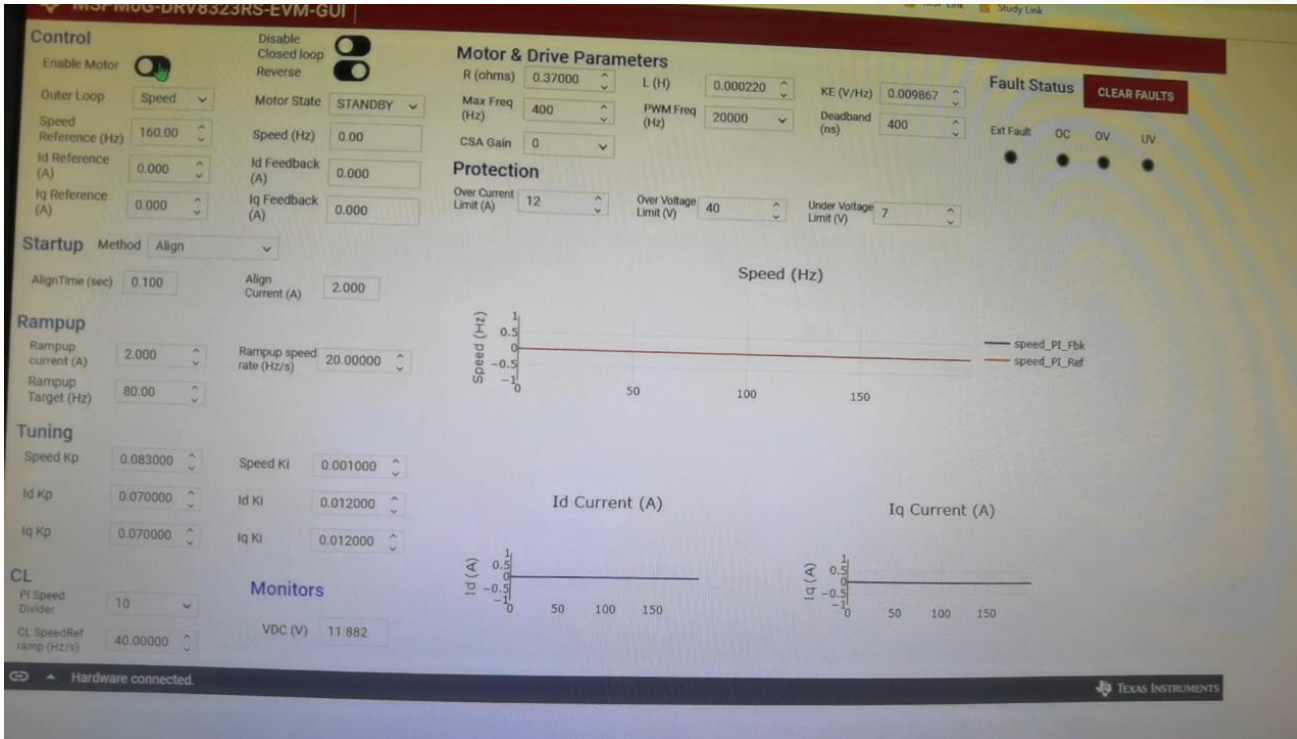
Id Current (A) Graph

Graph showing Id Current (A) vs Time (s). The current is near 0 A during the Rampup stage and increases to approximately 2 A during the Closed loop stage.

Iq Current (A) Graph

Graph showing Iq Current (A) vs Time (s). The current is near 0 A during the Rampup stage and increases to approximately 2 A during the Closed loop stage.

Example of the GUI – Video






MSPM0G-DRV8323RS CCS Project

Environment required

CCS latest version MSPM0G3507 LaunchPad

SDK latest version DRV8323RSEVM

Document for the foc sensorless solution

ti > mspm0_sdk_1_20_01_06 > docs > english > middleware > motor_control_bldc_sensorless_foc						
<input type="checkbox"/>	Name	Date modified	Type	Size		
	doc_guide	11/29/2023 10:37 AM	File folder			
	doxygen	11/29/2023 10:35 AM	File folder			
	Sensorless_FOC_Motor_Control_User_Guide...	11/20/2023 9:38 PM	Chrome HTML Docu...	1 KB		

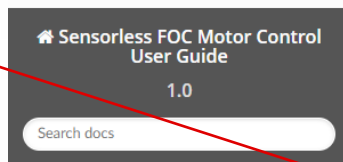
MSPM0G-DRV8323RS GUI

Introduction for GUI: Download the SDK latest version and find the file path:

MSPM0 FOC software
introduction (CCS Project)

MSPM0G-DRV8323RS GUI
Introduction

MSPM0G3507 LaunchPad and DRV8323RS
EVM hardware connection introduction



MSPM0 Sensorless FOC Software
User's Guide
Sensorless FOC Motor Control Library
Overview
DRV8323RS GUI User Guide
DRV8323RS Hardware User Guide

» Sensorless FOC Motor Control User Guide

Sensorless FOC Motor Control User Guide

• MSPM0 Sensorless FOC Software User's Guide

- CCS Project setup
- Adding Initial Parameters (optional)
- Starting the Project
- Running the Project
- 3. API Guide
- 4. Known Issues
- 5. Supported Devices

• Sensorless FOC Motor Control Library Overview

- 1. Software Overview



• DRV8323RS GUI User Guide


- 1. Overview
- 1.1 Getting Started with GUI
- 2. Using the GUI
- 3. GUI Window
















• DRV8323RS Hardware User Guide

- 1. Hardware Required
- 2. Hardware Setup

CCS Project Path

ti > mspm0_sdk_1_20_00_05 > examples > nortos > LP_MSPM0G3507 >   Search LP_MSPM0G3507

Folder 

Name	Date modified	Type	Size
 boot_manager	10/10/2023 12:55 PM	File folder	
 bsl	10/10/2023 12:55 PM	File folder	
 cmsis_dsp	10/10/2023 12:55 PM	File folder	
 demos	10/10/2023 12:54 PM	File folder	
 driverlib	10/10/2023 12:54 PM	File folder	
 drivers	10/10/2023 12:55 PM	File folder	
 eeprom	10/10/2023 12:54 PM	File folder	
 gui_composer	10/10/2023 12:54 PM	File folder	
 iqmath	10/10/2023 12:54 PM	File folder	
 lin	10/10/2023 12:54 PM	File folder	
 motor_control_bldc_sensored_trap_hall	10/10/2023 12:54 PM	File folder	
 motor_control_bldc_sensorless_foc	10/10/2023 12:55 PM	File folder	
 msp_subsystems	10/10/2023 12:55 PM	File folder	
 pmbus	10/10/2023 12:55 PM	File folder	
 smbus	10/10/2023 12:54 PM	File folder	

CCS Project Overall

The screenshot displays the CCS IDE interface with several key components highlighted by red boxes and annotations:

- Project Explorer (Left):** Shows the project structure for `sensorless-foc-demo_LP_MSPM0G3507_nortos_ticlang`. The `modules` folder is highlighted, containing `hal`, `iqmath_rts`, `motor`, `motor_driver`, and `sensorless_foc`. Below it, `targetConfigs` contains `ticlang`, which includes `drv8323rs-gui.c`, `drv8323rs-gui.h`, and `main.c`. The `sensorless-foc.syscfg` file is also highlighted.
- Annotation:** A red box around the `modules` folder is labeled "Module source code". Another red box around the `drv8323rs-gui.c` and `drv8323rs-gui.h` files is labeled "Used for GUI communication".
- Sysconfig Peripheral Library (Middle):** A red box highlights the list of peripherals, including Board, DMA, GPIO, MATHACL, Configuration NVM, RTC, SYSCTL, SYSTICK, WWD, ANALOG (ADC12, COMP, DAC12, GPAMP, OPA, VREF), COMMUNICATIONS (I2C, I2C-SMBUS, MCAN, SPI, UART, UART-LIN), and TIMERS (TIMER-CAPT, TIMER-COMPARE, TIMER-PWM, TIMER-QEI, TIMED).
- Configuration Panels (Right):** A large red box encompasses the `Board` configuration panel, which includes sections for **Debug Configuration** (Debug Enable On SWD Pins), **Global Pin Configuration** (Enable Global Fast-Wake, Configure Unused Pins, Generate Peripherals & Pin Assignments File), and **Initialization Priority Configuration** (a list of priorities from 0 to 12, each with a dropdown menu).

Sysconfig for peripheral setting
(automatically generate the code)

Software structure

Main.c

system initialization



main loop

Update FOC control parameters
Handle the GUI command

Polling Task

1ms timer interrupt

FOC loop (Handle the state machine)

100us PWM interrupt

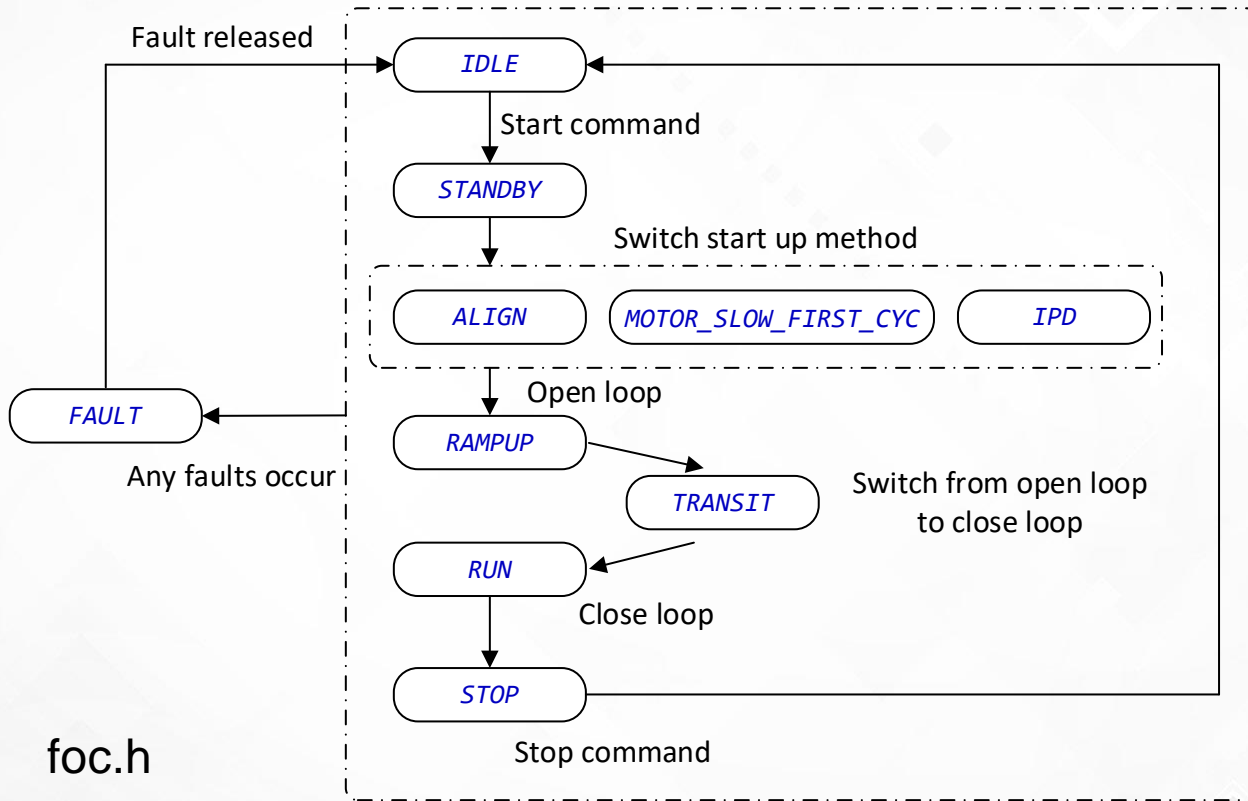
FOC run (Handle the foc algorithm)

ADC sampling interrupt

Update the adc sampling value

Interrupt Task

State Machine



foc.h

FOC sensorless flow:

Three stage start with multi-state machines.

Based on SDK 1.20.01 version

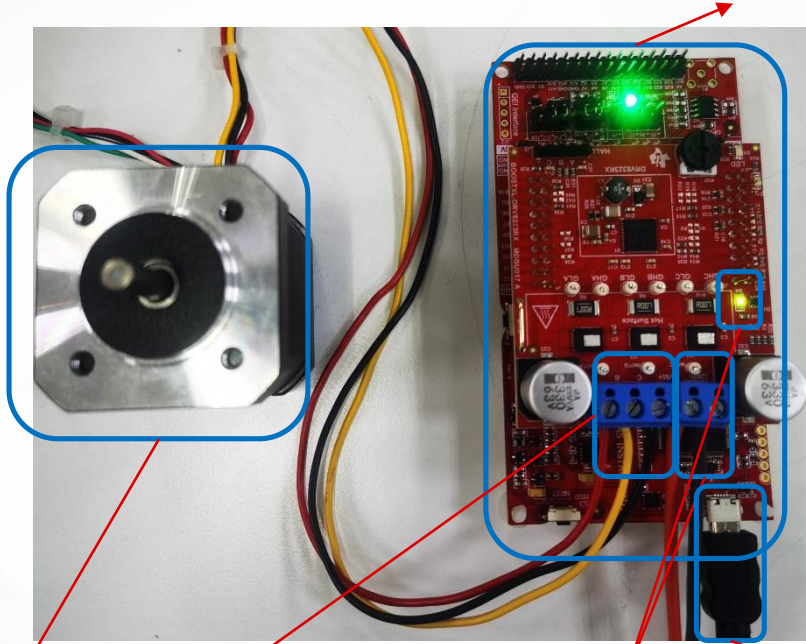
Step by Step Operations - CCS Project

Note:

The instructions are included in ***Sensorless FOC Motor Control User Guide***

Hardware Setup

MSPM0G3507 LaunchPad +
DRV8323RS EVM

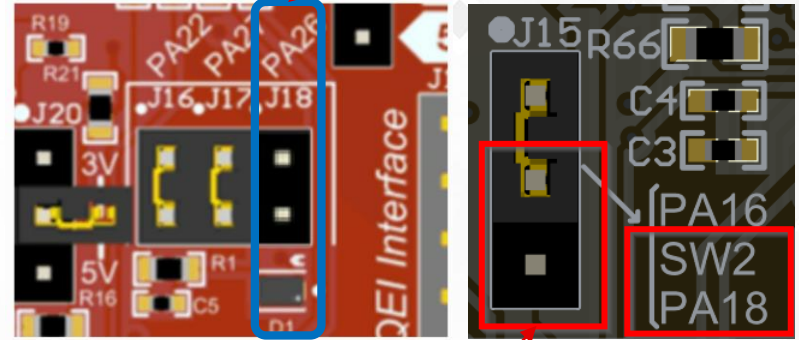


Motor
Motor Phase U/V/W

Power supply 4.5~34V
Turn on the power supply. The green VM LED
on the BOOSTXL-DRV8323RS should turn on.

Connect USB to the PC for
software download and debug

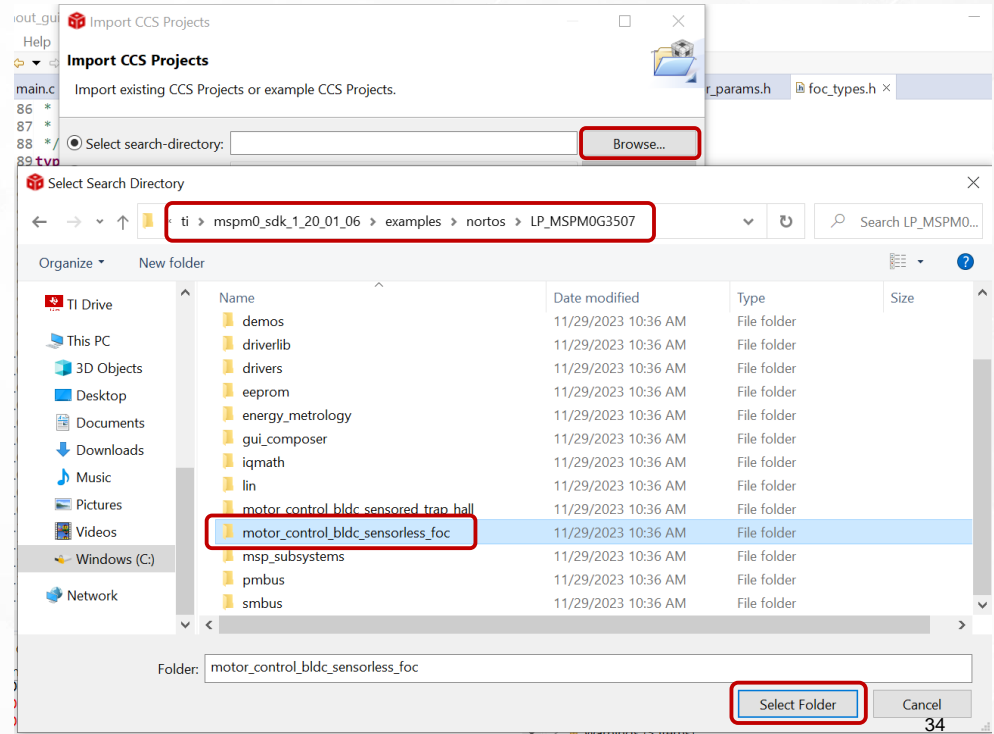
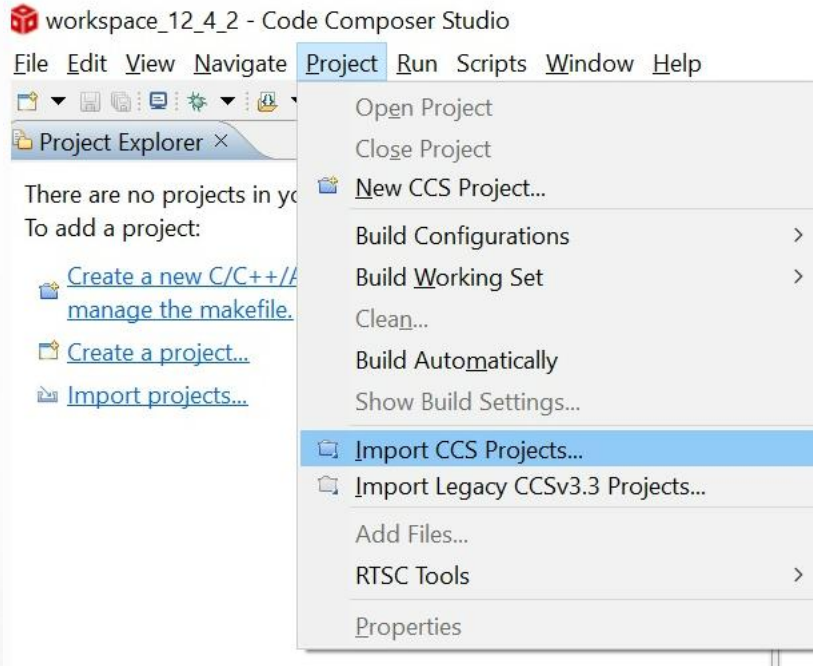
Remove the PA26 jumper J18 in the
LP-MSPM0G3507 as shown below



Connect PA18 with SW2 for ISENC
In Jumper J15

CCS Project setup

1. Import CCS Projects as shown below



CCS Project setup

2. Setting the Motor Parameters

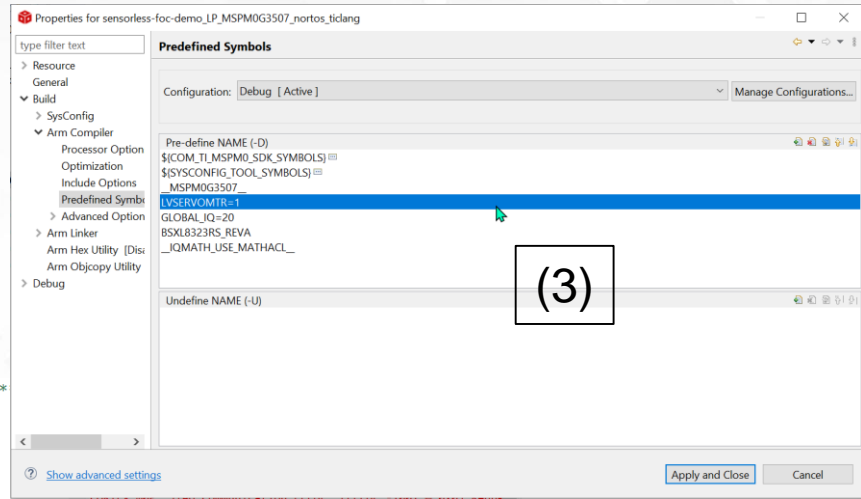
Project Explorer x

sensorless-foc-demo_LP_MSPM0G3507_nor

(1)

```
43 *****
44 /** @defgroup MOTOR_MODULE Motor_Module
45 * @{
46 */
47 #ifndef MOTOR_PARAMS_H
48 #define MOTOR_PARAMS_H
49
50 #ifdef __cplusplus
51 extern "C" {
52 #endif
53
54 #if (LVSERVOMTR)
55 /* Define the electrical motor parameters (Teknic 2310 Servomotor) */
56 /* @brief Stator resistance (ohm) */
57 #define MOTOR_PARA_RS (0.38)
58 /* @brief Stator inductance (H) */
59 #define MOTOR_PARA_LS (0.000169)
60 /* @brief Number of poles */
61 #define MOTOR_PARA_POLES (8)
62 /* @brief Back Emf constant in V/Hz */
63 #define MOTOR_PARA_KE (0.069)
64 /* @brief Maximum Frequency */
65 #define MOTOR_PARA_MAX_FREQ (400)
66
```

(2)

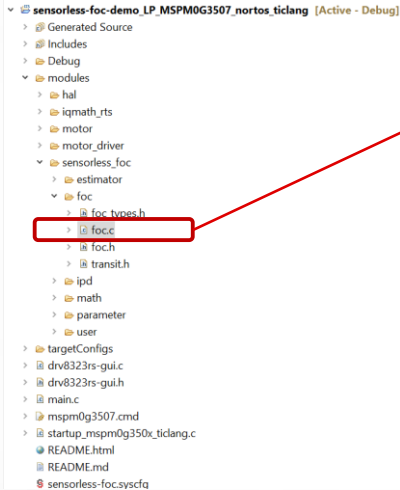


The motor parameter can be found in the motor_params.h file. These are the motor parameters required for Sensorless FOC:

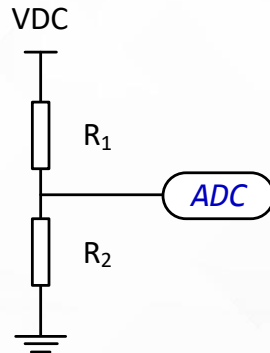
- Stator resistance, Rs (ohm)
- Stator inductance, Ls (H)
- Number of motor poles (optional)
- Back Emf constant in V/Hz
- Maximum Frequency of the motor (Hz)

CCS Project setup

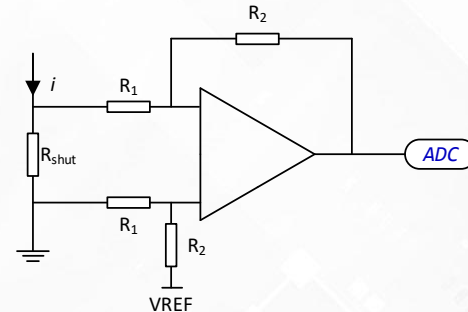
3. Setting the circuit parameters (if use DRV8323RS EVM, then skip this step)



```
126 /** @brief Computing voltage scale factor */
127 #define USER_DEFAULT_VOLTAGE_SF (HAL_ADC_DEFAULT_REF_VOLTAGE * \
128                                   (USER_DEFAULT_FOC_VOLT_RATIO))
129
130 /** @brief Computing current scale factor */
131 #define USER_DEFAULT_CURRENT_SF (HAL_ADC_DEFAULT_REF_VOLTAGE / \
132                                   (2 * USER_DEFAULT_DRV_RSHUNT * USER_DEFAULT_BASE_CSA_GAIN))
```



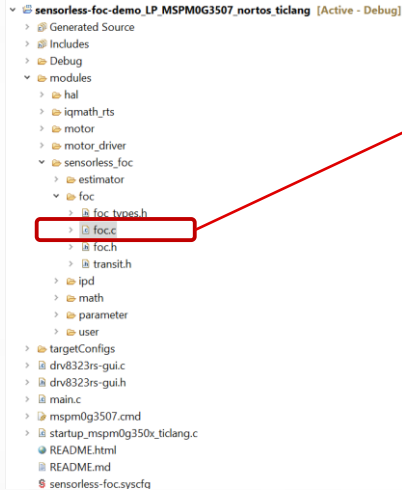
Voltage scale circuit



Current scale circuit

CCS Project setup

3. Setting the circuit parameters



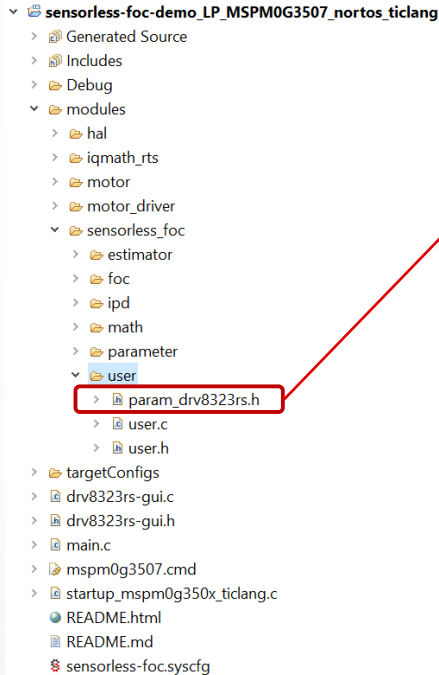
```
126 /** @brief Computing voltage scale factor */
127 #define USER_DEFAULT_VOLTAGE_SF (HAL_ADC_DEFAULT_REF_VOLTAGE * \
128                                   (USER_DEFAULT_FOC_VOLT_RATIO))
129
130 /** @brief Computing current scale factor */
131 #define USER_DEFAULT_CURRENT_SF (HAL_ADC_DEFAULT_REF_VOLTAGE/ \
132                                   (2 * USER_DEFAULT_DRV_RSHUNT * USER_DEFAULT_BASE_CSA_GAIN))
```

The reason that there needs “2” for current scale:

In the current calculation formula, it has divided by “2”, so here gives a compensation.

CCS Project setup

3. Setting the circuit parameters

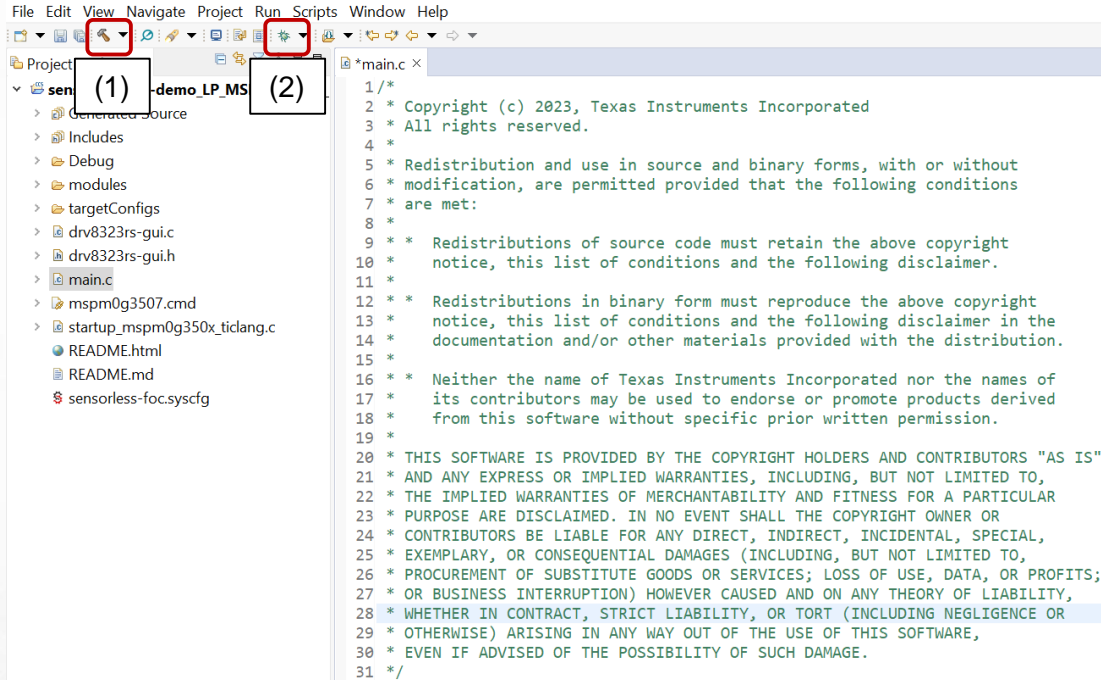


Basically, the foc control useful parameters is included in `param_drv8323rs.h`, include the voltage/current scale circuit macro definition params; setting params in each motor control state, ...etc.

Please refers to “***MSPM0 Sensorless FOC Software User’s Guide***” for details instruction, or refers to the note in `param_drv8323rs.h`.

CCS Project run

1. Enter debug mode to run the demo



Connect the hardware and turn on the power supply.

Click on build button.
- Project should build with no errors.

Click on debug button.

1. Enter debug mode to run the demo



2. Add the expression for observing

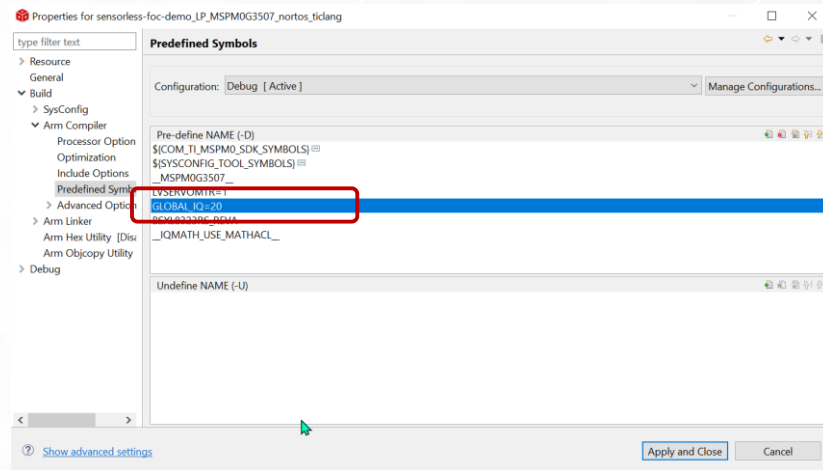
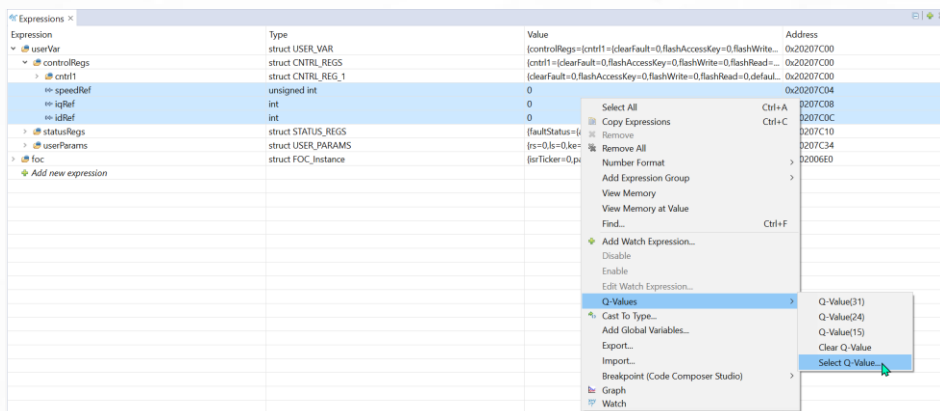
```
Add expressions:  
userVar;  
Foc;
```

If user want to observe any other variables, just add the expression here.

Enable continuous refresh

CCS Project run

2. Add the expression for observing




Some of the variables are in IQ20 format (Global default IQ format).

In order to be displayed and updated properly during debug, you should switch them to IQ(20) format in the CCS expressions window by right clicking the expression, select Q-Values, then click "Select Q-Value".

CCS Project run

3. Running the project

- Start the code 

You should turn on the power firstly and then start, or the current offset detection might be wrong.

Below is an example of the watch window:

Expression	Type	Value	Address
> userVar	struct USER_VAR	{controlRegs={cntrl1={clearFault=0,flashAc...	0x20207C00
> foc	struct FOC_Instance	{isrTicker=1894372,parameter=[398458,1...	0x202006E0
> isrTicker	unsigned int	1894690	0x202006E0
> parameter	unsigned int[34]	[398458,177,72351,400,0...]	0x202006E4
> state	enum MOTOR	MOTOR_STANDBY	0x202006E4
> faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvl=0,ovlo=0,o...	0x20200770
> cmd	struct COMMAND	{motorState=MOTOR_STATE_STOP,lqRef=...	0x20200774
> control	struct CONTROL	{align={cnt=0,alignTime=500,alignCurr=2...	0x20200790
> vdcAdc	unsigned int	843	0x202007C8
> iaAdc	unsigned int	2069	0x202007CC
> ibAdc	unsigned int	2050	0x202007D0
> icAdc	unsigned int	2070	0x202007D4
> offsetCalib	unsigned char	1 '\x01'	0x202007D8
> offsetA	unsigned int	2067	0x202007DC
> offsetB	unsigned int	2054	0x202007E0
> offsetC	unsigned int	2062	0x202007E4
> vdc	int	11.79782772 (Q-Value(20))	0x202007E8
> ia	int	0.1381130219 (Q-Value(20))	0x202007EC
> ib	int	-0.09207630157 (Q-Value(20))	0x202007F0
> overVoltageLimit	int	40.0 (Q-Value(20))	0x202007F4
> underVoltageLimit	int	7.0 (Q-Value(20))	0x202007F8
> overCurrentLimit	int	12.0 (Q-Value(20))	0x202007FC
> PISpdExecDivider	enum FOC_SPD_DIV	FOC_SPD_DIV_10	0x20200800
> PISpdExecCount	unsigned int	0	0x20200804
> enablePWM	unsigned char	0 '\x00'	0x20200808
> enablePWMStatus	unsigned char	0 '\x00'	0x20200809
> csaDiv	unsigned char	0 '\x00'	0x2020080A
> drv_handle	void *	0x2020080C	0x2020080C
> hal	struct FOC_HAL	{pwmAHal=HAL_PWM_CHANNEL_1_pwm...	0x20200810

Motor status and Fault status

Actual current value converted by adc

Current offset for bipolar current sensing

Detected voltage / current value for motor control

Protection parameters

Note: FOC algorithm only use two phase currents.

CCS Project run

4. Modify the parameters

User can change the params during the debug mode, select *userVar.userParams*

Expression	Type	Value	Address
userVar	struct USER_VAR	{controlRegs={cntrl1=(clearFault=0,flashAc...	0x20207C00
controlRegs	struct CNTRL_REGS	{cntrl1=(clearFault=0,flashAccessKey=0,fla...	0x20207C00
statusRegs	struct STATUS_REGS	{faultStatus={all=0,bits={extFaultIn=0,uvlo...	0x20207C10
userParams	struct USER_PARAMS	{rs=387973,ls=230,ke=10346,maxFreq=4...	0x20207C34
rs	unsigned int	0.3699998856 (Q-Value(20))	0x20207C34
ls	unsigned int	0.0002193450928 (Q-Value(20))	0x20207C38
ke	unsigned int	0.009866714478 (Q-Value(20))	0x20207C3C
maxFreq	unsigned int	400	0x20207C40
PWMFreq	enum FOC_PWM_FREQ	FOC_PWM_FREQ_10000	0x20207C44
deadband	unsigned int	400	0x20207C48
CSAGain	enum FOC_CSA_GAIN	FOC_CSA_GAIN_0	0x20207C4C
outerLoop	enum FOC_OUTER_LOOP	FOC_OUTER_LOOP_SPEED	0x20207C50
directionReversal	enum FOC_DIRECTION	FOC_DIRECTION_CW	0x20207C54
startupMethod	enum FOC_STARTUP	FOC_STARTUP_ALIGN	0x20207C58
slowCycFreq	unsigned int	5242880	0x20207C5C
alignCur	unsigned int	2097152	0x20207C60
alignTime	unsigned int	524288	0x20207C64
IPDCurrThresh	enum FOC_IPD_THRES_COUNT	FOC_IPD_THRES_COUNT_16	0x20207C68
IPDFreq	enum FOC_IPD_FREQ	FOC_IPD_FREQ_4000	0x20207C6C
rampupCur	unsigned int	2097152	0x20207C70
rampupRate	unsigned int	41943040	0x20207C74
rampupTarget	unsigned int	83886080	0x20207C78
speedRefRampRate	unsigned int	83886080	0x20207C7C
piSpdKp	unsigned int	104857	0x20207C80
piSpdKi	unsigned int	943	0x20207C84
piSpdDiv	enum FOC_SPD_DIV	FOC_SPD_DIV_10	0x20207C88
piIqKp	unsigned int	55574	0x20207C8C
piIqKi	unsigned int	12582	0x20207C90
piIdKp	unsigned int	55574	0x20207C94
piIdKi	unsigned int	12582	0x20207C98
overVoltageLimit	unsigned int	41943040	0x20207CA4
underVoltageLimit	unsigned int	7340032	0x20207CA8
overCurrentLimit	unsigned int	12582912	0x20207CAC
writeFlag	unsigned int	2023406814	0x20207CA8

Motor params

PWM setting

Motor control setting

Start up params. (Align & IPD)

Ramp up params

PI regulator params

Protection params

What GUI has

CCS Project run

User can change the params during the debug mode, select *userVar.userParams*

4. Modify the parameters

All the data listed in *userVar.userParams*, user can set it default value in the corresponding header files and macro definitions.

CCS Project setup describe some parameters definition.

If user can't find the definition you want to modify, you can use *userVar.userParams* to change during debug mode. It doesn't require reloading the program or restart the code, modifications will automatically take effect.

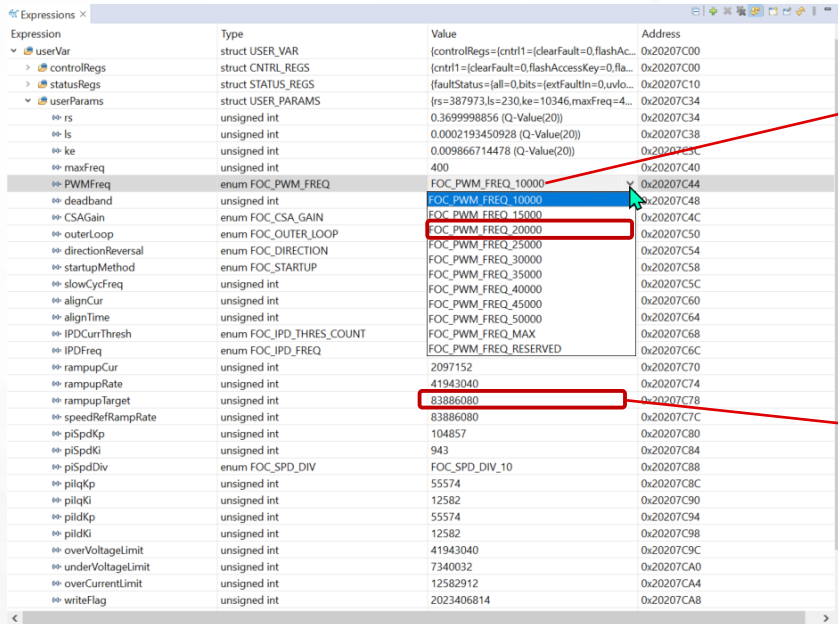
If user spin the motor through the GUI firstly, then he can use the same parameters in the GUI here.

CCS Project run

4. Modify the parameters

Examples:

User can change the params during the debug mode, select *userVar.userParams*



Expression	Type	Value	Address
userVar	struct USER_VAR	{controlRegs=(ctrl1=1,clearFault=0,flashAc...	0x20207C00
controlRegs	struct CNTRL_REGS	{ctrl1=1,clearFault=0,flashAccessKey=0,fla...	0x20207C00
statusRegs	struct STATUS_REGS	{faultStatus=(all=0,bits=1,extFaultIn=0,uvlo...	0x20207C10
userParams	struct USER_PARAMS	{rs=387973,ls=230,ke=10346,maxFreq=4...	0x20207C34
rs	unsigned int	0.3699998856 (Q-Value(20))	0x20207C34
ls	unsigned int	0.0002193450928 (Q-Value(20))	0x20207C38
ke	unsigned int	0.009866714478 (Q-Value(20))	0x20207C3C
maxFreq	unsigned int	400	0x20207C40
PwmFreq	enum FOC_PWM_FREQ	FOC_PWM_FREQ_10000	0x20207C44
deadband	unsigned int	FOC_PWM_FREQ_10000	0x20207C48
CSAGain	enum FOC_CSA_GAIN	FOC_PWM_FREQ_15000	0x20207C4C
outerLoop	enum FOC_OUTER_LOOP	FOC_PWM_FREQ_20000	0x20207C50
directionReversal	enum FOC_DIRECTION	FOC_PWM_FREQ_25000	0x20207C54
startupMethod	enum FOC_STARTUP	FOC_PWM_FREQ_30000	0x20207C58
slowCycFreq	unsigned int	FOC_PWM_FREQ_35000	0x20207C5C
alignCur	unsigned int	FOC_PWM_FREQ_40000	0x20207C60
alignTime	unsigned int	FOC_PWM_FREQ_45000	0x20207C64
IPDCurThresh	enum FOC_IPD_THRES_COUNT	FOC_PWM_FREQ_50000	0x20207C68
IPDFreq	enum FOC_IPD_FREQ	FOC_PWM_FREQ_MAX	0x20207C6C
rampupCur	unsigned int	FOC_PWM_FREQ_RESERVED	0x20207C6C
rampupRate	unsigned int	2097152	0x20207C70
rampupTarget	unsigned int	41943040	0x20207C74
speedRefRampRate	unsigned int	83886080	0x20207C78
piSpdKp	unsigned int	104857	0x20207C80
piSpdKi	unsigned int	943	0x20207C84
piSpdDiv	enum FOC_SPD_DIV	FOC_SPD_DIV_10	0x20207C88
piIqKp	unsigned int	55574	0x20207C8C
piIqKi	unsigned int	12582	0x20207C90
piIdKp	unsigned int	55574	0x20207C94
piIdKi	unsigned int	12582	0x20207C98
overVoltageLimit	unsigned int	41943040	0x20207C9C
underVoltageLimit	unsigned int	7340032	0x20207CA0
overCurrentLimit	unsigned int	12582912	0x20207CA4
writeFlag	unsigned int	2023406814	0x20207CA8

An example to change the PWM frequency.

1. Double click on the value
2. Select the frequency you want

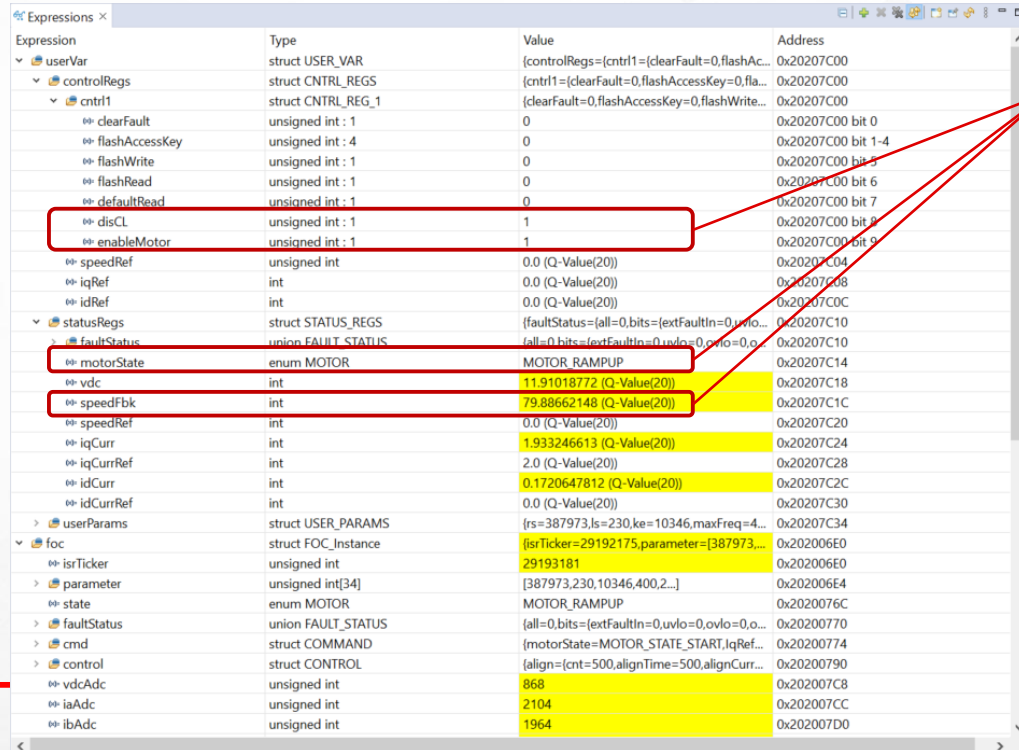
An example to change the rampup target speed

1. Select the correct IQ format for the data (IQ20)
2. Double click on the value
3. Input the value you want by keyboard

CCS Project run

5. Spin the motor – open loop

User can control motor status by *userVar.controlRegs*

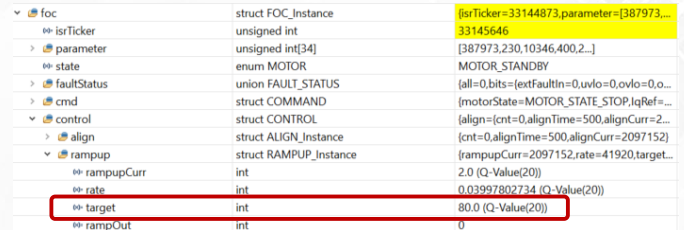


Expression	Type	Value	Address
userVar	struct USER_VAR	{controlRegs={cntrl1={clearFault=0,flashAc...	0x20207C00
controlRegs	struct CNTRL_REGS	{cntrl1={clearFault=0,flashAccessKey=0,fla...	0x20207C00
cntrl1	struct CNTRL_REG_1	{clearFault=0,flashAccessKey=0,flashWrite...	0x20207C00
clearFault	unsigned int : 1	0	0x20207C00 bit 0
flashAccessKey	unsigned int : 4	0	0x20207C00 bit 1-4
flashWrite	unsigned int : 1	0	0x20207C00 bit 5
flashRead	unsigned int : 1	0	0x20207C00 bit 6
defaultRead	unsigned int : 1	0	0x20207C00 bit 7
disCL	unsigned int : 1	1	0x20207C00 bit 8
enableMotor	unsigned int : 1	1	0x20207C00 bit 9
speedRef	unsigned int	0.0 (Q-Value(20))	0x20207C04
iqRef	int	0.0 (Q-Value(20))	0x20207C08
idRef	int	0.0 (Q-Value(20))	0x20207C0C
statusRegs	struct STATUS_REGS	{faultStatus={all=0,bits={extFaultIn=0,uvlo=...	0x20207C10
faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvlo=0,ovlo=0,o...	0x20207C10
motorState	enum MOTOR	MOTOR_RAMPUP	0x20207C14
vdc	int	11.91018772 (Q-Value(20))	0x20207C18
speedFbk	int	79.88662148 (Q-Value(20))	0x20207C1C
speedRef	int	0.0 (Q-Value(20))	0x20207C20
iqCurr	int	1.933246613 (Q-Value(20))	0x20207C24
iqCurrRef	int	2.0 (Q-Value(20))	0x20207C28
idCurr	int	0.1720647812 (Q-Value(20))	0x20207C2C
idCurrRef	int	0.0 (Q-Value(20))	0x20207C30
userParams	struct USER_PARAMS	{rs=387973,ls=230,ke=10346,maxFreq=4...	0x20207C34
foc	struct FOC_Instance	{isrTicking=29192175,parameter=[387973,...	0x202006E0
isrTicking	unsigned int	29192175	0x202006E0
parameter	unsigned int[34]	[387973,230,10346,400,2...]	0x202006E4
state	enum MOTOR	MOTOR_RAMPUP	0x202006E8
faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvlo=0,ovlo=0,o...	0x202006F0
cmd	struct COMMAND	{motorState=MOTOR_STATE_START,IqRef...	0x202006F4
control	struct CONTROL	{align={cnt=500,alignTime=500,alignCurr...	0x202006F8
vdcAdc	unsigned int	868	0x202006FC
iaAdc	unsigned int	2104	0x202006FF
ibAdc	unsigned int	1964	0x20200700

Set disCL = 1, and then Set enableMotor = 1. The motor will start to spin and end in open loop at the speed of the ramp up target you set.

User can control open loop speed by *userVar.userParams* ;

or control by *foc.control.rampup* ;

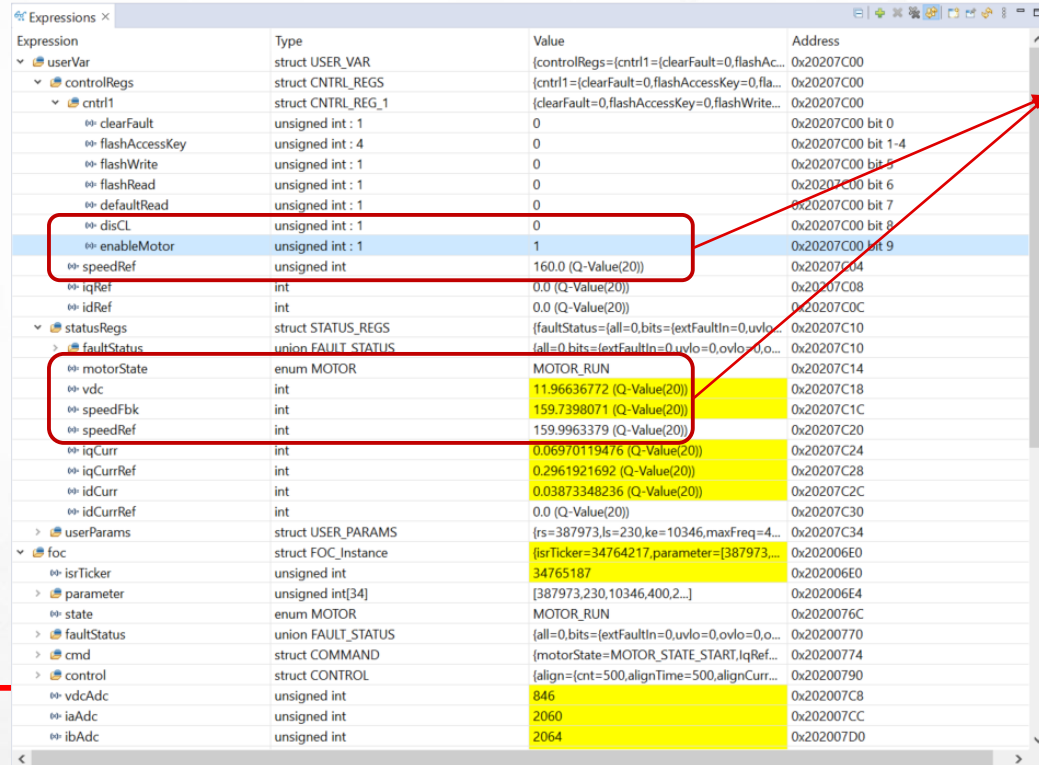


foc	struct FOC_Instance	{isrTicking=33144873,parameter=[387973,...
isrTicking	unsigned int	33144873
parameter	unsigned int[34]	[387973,230,10346,400,2...]
state	enum MOTOR	MOTOR_STANDBY
faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvlo=0,ovlo=0,o...
cmd	struct COMMAND	{motorState=MOTOR_STATE_STOP,IqRef=...
control	struct CONTROL	{align={cnt=0,alignTime=500,alignCurr=2...
align	struct ALIGN_Instance	{cnt=0,alignTime=500,alignCurr=2097152}
rampup	struct RAMPUP_Instance	{rampupCurr=2097152,rate=41920,target...
rampupCurr	int	2.0 (Q-Value(20))
rate	int	0.03997802734 (Q-Value(20))
target	int	80.0 (Q-Value(20))
rampOut	int	0

CCS Project run

5. Spin the motor – close loop

User can control motor status by `userVar.controlRegs`



Expression	Type	Value	Address
userVar	struct USER_VAR	{controlRegs={cntrl1={clearFault=0,flashAc...	0x20207C00
controlRegs	struct CNTRL_REGS	{cntrl1={clearFault=0,flashAccessKey=0,fla...	0x20207C00
cntrl1	struct CNTRL_REG_1	{clearFault=0,flashAccessKey=0,flashWrite...	0x20207C00
clearFault	unsigned int : 1	0	0x20207C00 bit 0
flashAccessKey	unsigned int : 4	0	0x20207C00 bit 1-4
flashWrite	unsigned int : 1	0	0x20207C00 bit 5
flashRead	unsigned int : 1	0	0x20207C00 bit 6
defaultRead	unsigned int : 1	0	0x20207C00 bit 7
disCL	unsigned int : 1	0	0x20207C00 bit 8
enableMotor	unsigned int : 1	1	0x20207C00 bit 9
speedRef	unsigned int	160.0 (Q-Value(20))	0x20207C04
iqRef	int	0.0 (Q-Value(20))	0x20207C08
idRef	int	0.0 (Q-Value(20))	0x20207C0C
statusRegs	struct STATUS_REGS	{faultStatus={all=0,bits={extFaultIn=0,uvlo...	0x20207C10
faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvlo=0,ovlo=0,o...	0x20207C10
motorState	enum MOTOR	MOTOR_RUN	0x20207C14
vdc	int	11.96636772 (Q-Value(20))	0x20207C18
speedFbk	int	159.7398071 (Q-Value(20))	0x20207C1C
speedRef	int	159.9963379 (Q-Value(20))	0x20207C20
iqCurr	int	0.06970119476 (Q-Value(20))	0x20207C24
iqCurrRef	int	0.2961921692 (Q-Value(20))	0x20207C28
idCurr	int	0.03873348236 (Q-Value(20))	0x20207C2C
idCurrRef	int	0.0 (Q-Value(20))	0x20207C30
userParams	struct USER_PARAMS	{rs=387973,ls=230,ke=10346,maxFreq=4...	0x20207C34
foc	struct FOC_Instance	{isrTicker=34764217,parameter=[387973,...	0x202006E0
isrTicker	unsigned int	34765187	0x202006E0
parameter	unsigned int[34]	[387973,230,10346,400,2...]	0x202006E4
state	enum MOTOR	MOTOR_RUN	0x2020076C
faultStatus	union FAULT_STATUS	{all=0,bits={extFaultIn=0,uvlo=0,ovlo=0,o...	0x20200770
cmd	struct COMMAND	{motorState=MOTOR_STATE_START,iqRef...	0x20200774
control	struct CONTROL	{align={cnt=500,alignTime=500,alignCurr...	0x20200790
vdcAdc	unsigned int	846	0x202007C8
iaAdc	unsigned int	2060	0x202007CC
ibAdc	unsigned int	2064	0x202007D0

Set `disCL = 0`, set `speedRef` as you want, and then Set `enableMotor = 1`. The motor will start to spin and end in close loop at the speed of the `speedRef` you set.

User can change close loop speed by changing the `speedRef`.