TI Spins Motors

Texas Instruments

C2000Ware MotorControl SDK InstaSPIN Universal GUI

QUICK START GUIDE

A series of lab projects are available to help you evaluate InstaSPIN™-FOC solutions. The C2000Ware MotorControl SDK InstaSPIN Universal GUI allows you to instrument and interact with these projects over a JTAG connection.

This InstaSPIN Universal GUI supports:

- LAUNCHXL-F280049C LaunchPad with on-board XDS110 debug probe
- TMDSCNCD280049C controlCARD with on-board XDS100v2 debug probe
- Any InstaSPIN-enabled F28004x device on custom hardware with a JTAG connection

Contents

1 Installation

The InstaSPIN Universal GUI is available in the TI Cloud Tools Gallery and can be run in your web browser without any installation of the application itself. As you follow the steps outlined in this document, dev.ti.com may prompt you to install tools as needed—such as the TI Cloud Agent to allow your browser to connect to your board device or the GUI Composer Runtime to allow you to run the GUI standalone on your desktop. Perform these installations as requested, following the instructions on the site.

2 Hardware setup

Please review the quick start or hardware guides for your particular controlCARD, LaunchPad, and/or motor drive kit (that is available in the MotorControl SDK) for details of hardware setup, including jumper and switch settings.

3 Using the GUI

The InstaSPIN Universal GUI is available in the TI Cloud Tools Gallery (Chrome is the preferred browser). Type “InstaSPIN” into the search box and hit enter. You should see the “C2000Ware MotorControl SDK InstaSPIN GUI” in the search results.

Clicking the GUI image will launch the application in your browser. **However**, by default, the GUI is configured to load and run a .out for the LAUNCHXL-F280049C with a BOOSTXL-DRV8320RS with default user.h parameters. As this is a very specific configuration, you should complete the following steps to run the application instead.

3.1 Create a binary

The Universal GUI allows you to instrument variables that exist in a .out file that the GUI will load and run. Variables that you will interact with primarily reside in the userParams and motorVars structures (declared in the main.c of most InstaSPIN labs). You will need to make sure that the variables with which you want to interact are compiled into your .out Also, be aware that even though a variable may exist in the these structures, changing the variable in the GUI may have no effect if it is not “get” or “set” in the application being run.

To start, it is best to use a .out from an InstaSPIN lab project from the MotorControl SDK. If you are new to InstaSPIN and CCS, start with the **MotorControl SDK InstaSPIN Lab Guide** found in the *\solutions\common\sensorless_foc\docs\labs* directory of the SDK. It will give you an introduction to the labs’ functionality, how to import them into CCS, how to customize the code for your motor, and how to build the .out. This document will use the InstaSPIN Motor ID lab (is05_motor_id) for its examples.

It is recommended that you use the Flash_Lib build configuration.

Once you have built your project, locate the .out in your workspace. You will need it for the following steps.

3.2 Configuring the GUI

From the aforementioned TI Cloud Tools Gallery search result page, click the “Import to GUI Composer” icon for the Universal GUI.

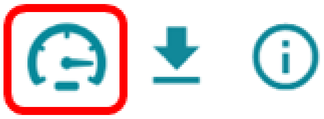
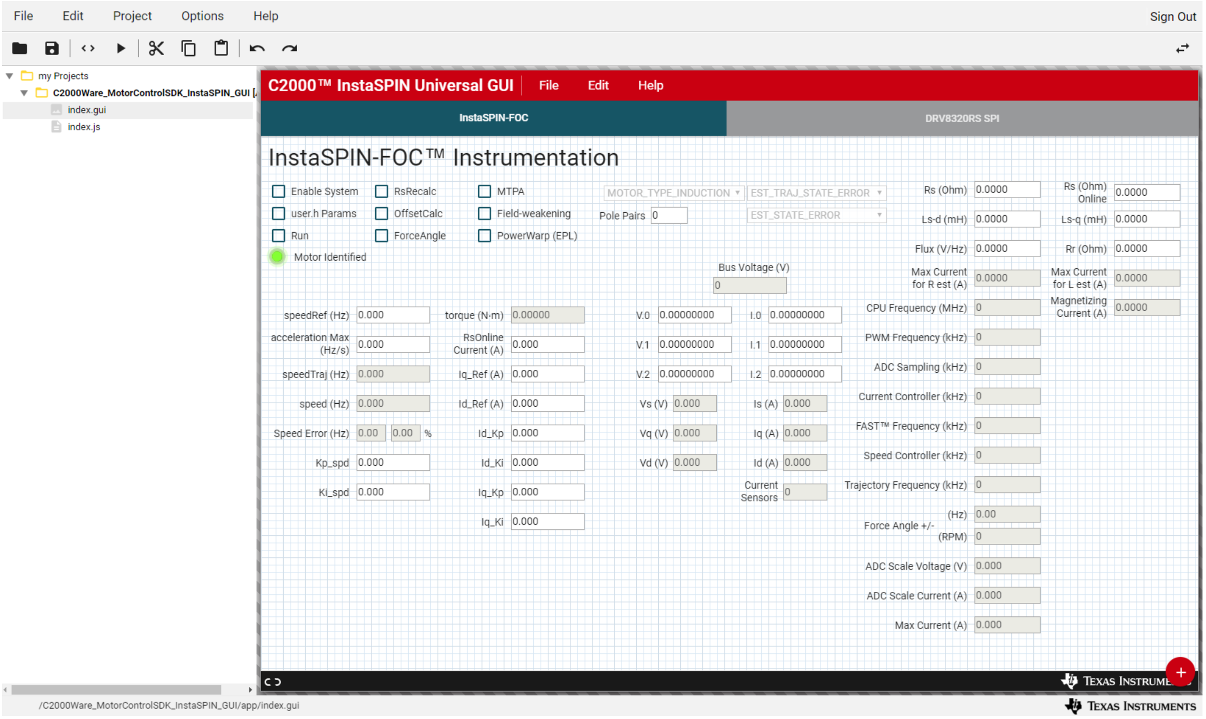


Figure 1 - Red box highlights the "Import to GUI Composer" icon in the Gallery.

You should now see **C2000Ware_MotorControlSDK_InstaSPIN_GUI** in your GUI Composer projects list, and the front tab of the Universal GUI in the main panel.



InstaSPIN-FOC™ Instrumentation

□ Enable System □ RsRecalc □ MTPA
□ user.h Params □ OffsetCalc □ Field-weakening
□ Run □ ForceAngle □ PowerWarp (EPL)
● Motor Identified

MOTOR_TYPE_INDUCTION
EST_TRAJ_STATE_ERROR
Pole Pairs: 0
EST_STATE_ERROR
Bus Voltage (V): 0

speedRef (Hz): 0.000
acceleration Max (Hz/s): 0.000
speedTraj (Hz): 0.000
speed (Hz): 0.000
Speed Error (Hz): 0.00 0.00 %
Kp_spd: 0.000
Ki_spd: 0.000

torque (N·m): 0.00000
RsOnline Current (A): 0.000
Iq_Ref (A): 0.000
Id_Ref (A): 0.000
Id_Kp: 0.000
Id_Ki: 0.000
Iq_Kp: 0.000
Iq_Ki: 0.000

V.0: 0.00000000
V.1: 0.00000000
V.2: 0.00000000
Vs (V): 0.000
Vq (V): 0.000
Vd (V): 0.000

I.0: 0.00000000
I.1: 0.00000000
I.2: 0.00000000
Is (A): 0.000
Iq (A): 0.000
Id (A): 0.000
Current Sensors: 0

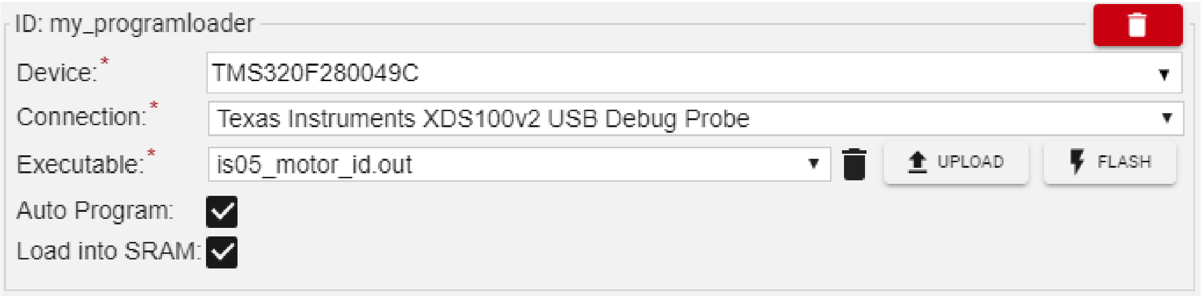
Rs (Ohm): 0.0000
Ls-d (mH): 0.0000
Flux (V/Hz): 0.0000
Max Current for R est (A): 0.0000
CPU Frequency (MHz): 0
PWM Frequency (kHz): 0
ADC Sampling (kHz): 0
Current Controller (kHz): 0
FAST™ Frequency (kHz): 0
Speed Controller (kHz): 0
Trajectory Frequency (kHz): 0
Force Angle +/- (Hz): 0.00
Force Angle +/- (RPM): 0
ADC Scale Voltage (V): 0.000
ADC Scale Current (A): 0.000
Max Current (A): 0.000

Rs (Ohm) Online: 0.0000
Ls-q (mH): 0.0000
Rr (Ohm): 0.0000
Max Current for L est (A): 0.0000
Magnetizing Current (A): 0.0000

Figure 2 - View of the GUI Composer project workspace with the Universal GUI imported.

To configure your connection settings and .out, go to “Project”->”Properties…” Leave the first page of properties as it is and hit “Next.” You should see the Target Programming page where you can select your Device, Connection, and Executable. Go ahead and make the appropriate selections for your kit, uploading the .out generated in the previous section for the Executable.

For example, if you are using the TMDSCNCD280049C controlCARD with on-board XDS100v2 debug probe, you would change the Connection box to “Texas Instruments XDS100v2 USB Debug Probe” and upload the .out for an appropriate lab project.



ID: my_programloader

Device:* TMS320F280049C

Connection:* Texas Instruments XDS100v2 USB Debug Probe

Executable:* is05_motor_id.out

Auto Program: ☑

Load into SRAM: ☑

Figure 3 - Project properties window showing target connection settings.

Note that the “Auto Program” and “Load into SRAM” boxes are checked. Click “OK” when your selections are complete. You may want to save your project at this point.

3.3 Running the GUI

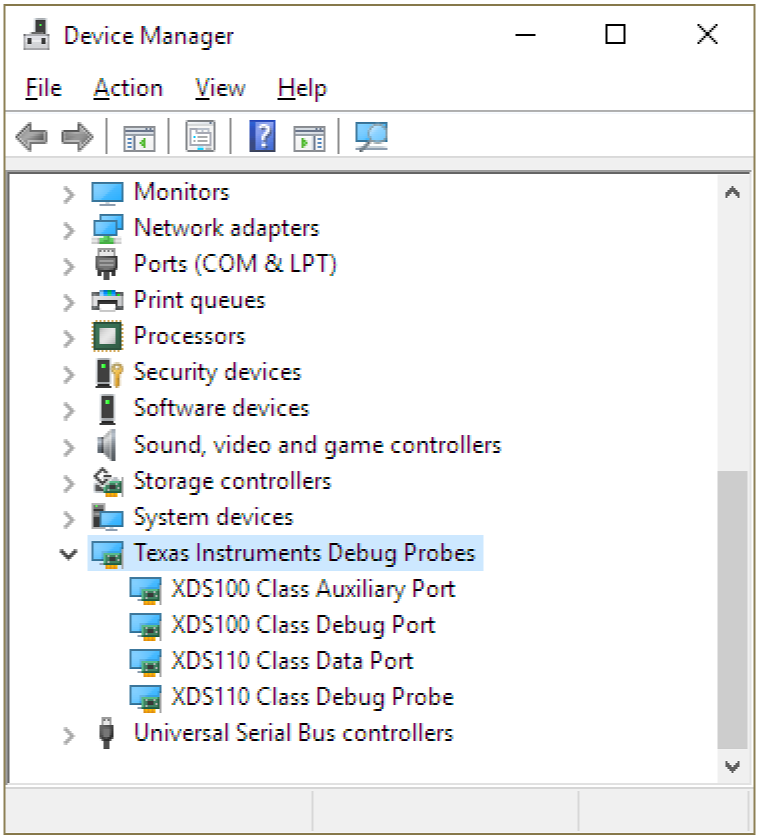
This section will outline how to run the GUI including how to set up hardware, launch the GUI, and interact with the application variables.

3.3.1 GUI connection

With the DC bus powered, connect a USB cable from the PC to the controlCARD or LaunchPad.

If using the high voltage kit, proceed with connecting to the target before energizing the high voltage AC input (110-220Vac) or high voltage DC Bus (50-350Vdc).

You can verify the connection to your emulator my checking the Windows Device Manager.



- Monitors
- Network adapters
- Ports (COM & LPT)
- Print queues
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Texas Instruments Debug Probes
  - XDS100 Class Auxiliary Port
  - XDS100 Class Debug Port
  - XDS110 Class Data Port
  - XDS110 Class Debug Probe
- Universal Serial Bus controllers

Figure 4 - Windows Device Manager showing both an XDS100 and an XDS110 connected.

3.3.2 Launching the GUI

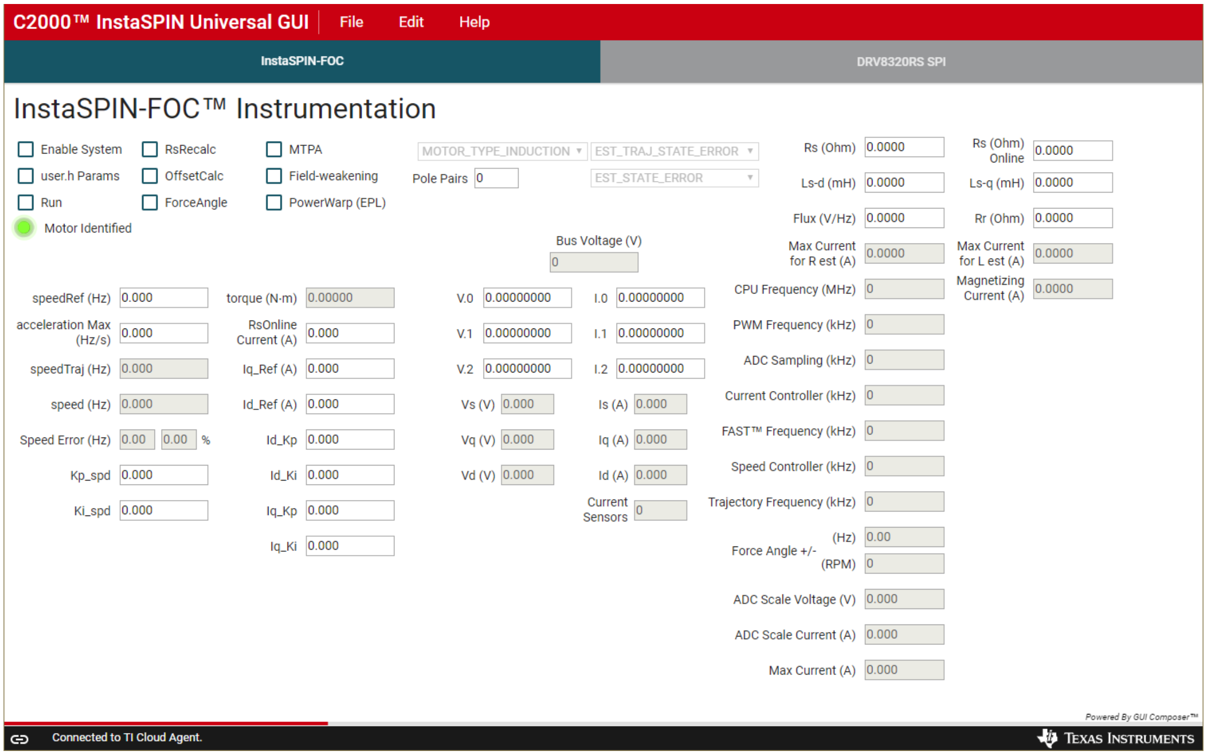
To launch the GUI, hit the Run button (the triangular icon) in GUI Composer. A new tab should open, showing the Universal GUI and a README window. Close the README once you have read it. If you need to see it again, you can access it under the “Help” menu.

The GUI will automatically connect to your device and load the application. You can track its progress by reading the status bar at the bottom. When it is connected and running, you should see “Connected to target. Hardware Connected.” in the status bar, and signs that the application is executing (e.g. a blinking LED on the board, a valid number of pole pairs displayed in the GUI).

If you need to disconnect/reconnect to the GUI, there is a button to toggle connection status in the lower left corner. If you need to reload the program or if it didn't auto-load upon connecting, there is a "Program Device…" selection under the "File" menu.

3.3.3 GUI overview

The first tab displayed in the GUI is the InstaSPIN-FOC Instrumentation tab. It provides instrumentation for the InstaSPIN-FOC labs, focused on the motorVars, and userParams structures. Use the InstaSPIN-FOC lab guide for more information on the items displayed and how to interact with them for each of the labs.



InstaSPIN-FOC™ Instrumentation

□ Enable System
□ user.h Params
□ Run
● Motor Identified
□ RsRecalc
□ OffsetCalc
□ ForceAngle
□ MTPA
□ Field-weakening
□ PowerWarp (EPL)

MOTOR_TYPE_INDUCTION ▾
EST_TRAJ_STATE_ERROR ▾
EST_STATE_ERROR ▾
Pole Pairs 0
Bus Voltage (V) 0

speedRef (Hz) 0.000
acceleration Max (Hz/s) 0.000
speedTraj (Hz) 0.000
speed (Hz) 0.000
Speed Error (Hz) 0.00 0.00 %
Kp_spd 0.000
Ki_spd 0.000

torque (N-m) 0.00000
RsOnline Current (A) 0.000
Iq_Ref (A) 0.000
Id_Ref (A) 0.000
Id_Kp 0.000
Id_Ki 0.000
Iq_Kp 0.000
Iq_Ki 0.000

V.0 0.00000000
V.1 0.00000000
V.2 0.00000000
Vs (V) 0.000
Vq (V) 0.000
Vd (V) 0.000

I.0 0.00000000
I.1 0.00000000
I.2 0.00000000
Is (A) 0.000
Iq (A) 0.000
Id (A) 0.000
Current Sensors 0

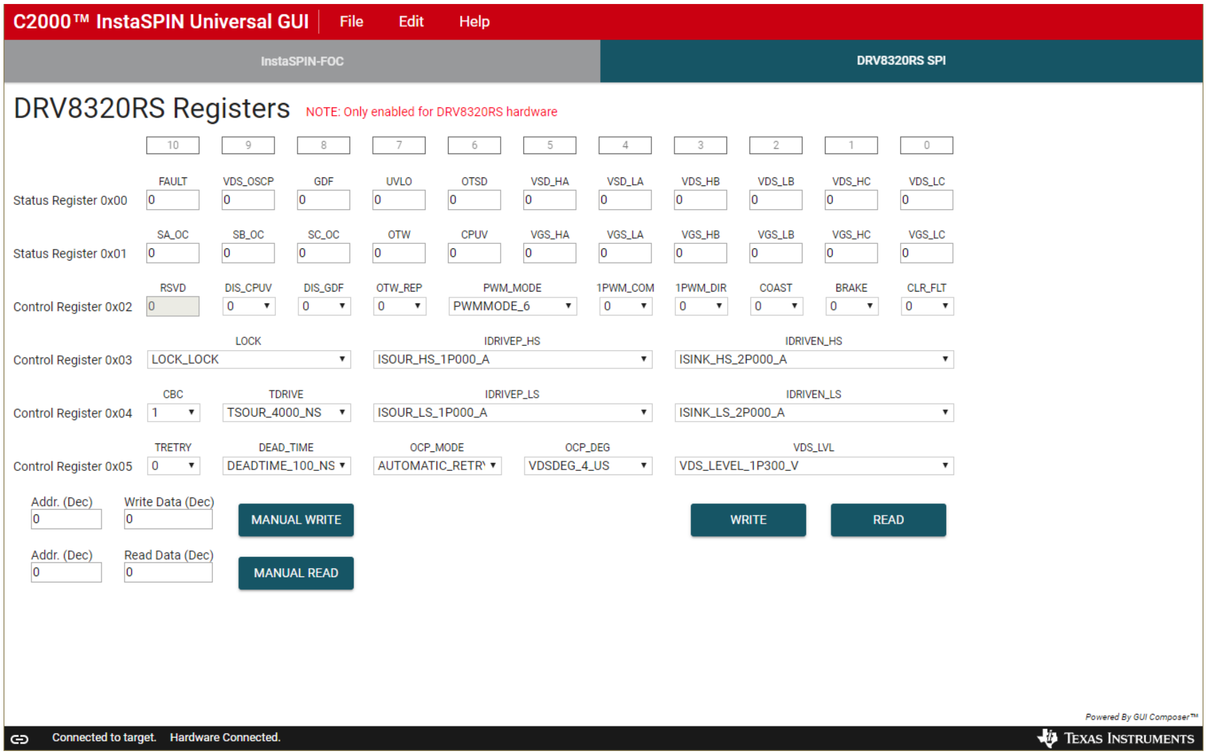
Rs (Ohm) 0.0000
Ls-d (mH) 0.0000
Flux (V/Hz) 0.0000
Max Current for R est (A) 0.0000
CPU Frequency (MHz) 0
PWM Frequency (kHz) 0
ADC Sampling (kHz) 0
Current Controller (kHz) 0
FAST™ Frequency (kHz) 0
Speed Controller (kHz) 0
Trajectory Frequency (kHz) 0
Force Angle +/- (Hz) 0.00
(RPM) 0
ADC Scale Voltage (V) 0.000
ADC Scale Current (A) 0.000
Max Current (A) 0.000

Rs (Ohm) Online 0.0000
Ls-q (mH) 0.0000
Rr (Ohm) 0.0000
Max Current for L est (A) 0.0000
Magnetizing Current (A) 0.0000

Powered By GUI Composer™
Connected to TI Cloud Agent.
Texas Instruments

Figure 5 – InstaSPIN-FOC instrumentation tab.

The next tab is provided to view and modify the configurations of the DRV8320RS gate driver if necessary. This tab only applies to the BOOSTXL-DRV8320RS and can be ignored if not in use.



DRV8320RS Registers NOTE: Only enabled for DRV8320RS hardware

	10	9	8	7	6	5	4	3	2	1	0
	FAULT	VDS_OSCP	GDF	UVLO	OTSD	VSD_HA	VSD_LA	VDS_HB	VDS_LB	VDS_HC	VDS_LC
Status Register 0x00	0	0	0	0	0	0	0	0	0	0	0
	SA_OC	SB_OC	SC_OC	OTW	CPUV	VGS_HA	VGS_LA	VGS_HB	VGS_LB	VGS_HC	VGS_LC
Status Register 0x01	0	0	0	0	0	0	0	0	0	0	0
	RSVD	DIS_CPUV	DIS_GDF	OTW_REP	PWM_MODE		1PWM_COM	1PWM_DIR	COAST	BRAKE	CLR_FLT
Control Register 0x02	0	0	0	0	PWMMODE_6		0	0	0	0	0
	LOCK			IDRIVEP_HS				IDRIVEN_HS			
Control Register 0x03	LOCK_LOCK			ISOUR_HS_1P000_A				ISINK_HS_2P000_A			
	CBC	TDRIVE		IDRIVEP_LS				IDRIVEN_LS			
Control Register 0x04	1	TSOUR_4000_NS		ISOUR_LS_1P000_A				ISINK_LS_2P000_A			
	TRETRY	DEAD_TIME		OCP_MODE		OCP_DEG		VDS_LVL			
Control Register 0x05	0	DEADTIME_100_NS		AUTOMATIC_RETR\`		VDSDEG_4_US		VDS_LEVEL_1P300_V			

Addr. (Dec) 0 Write Data (Dec) 0

Addr. (Dec) 0 Read Data (Dec) 0

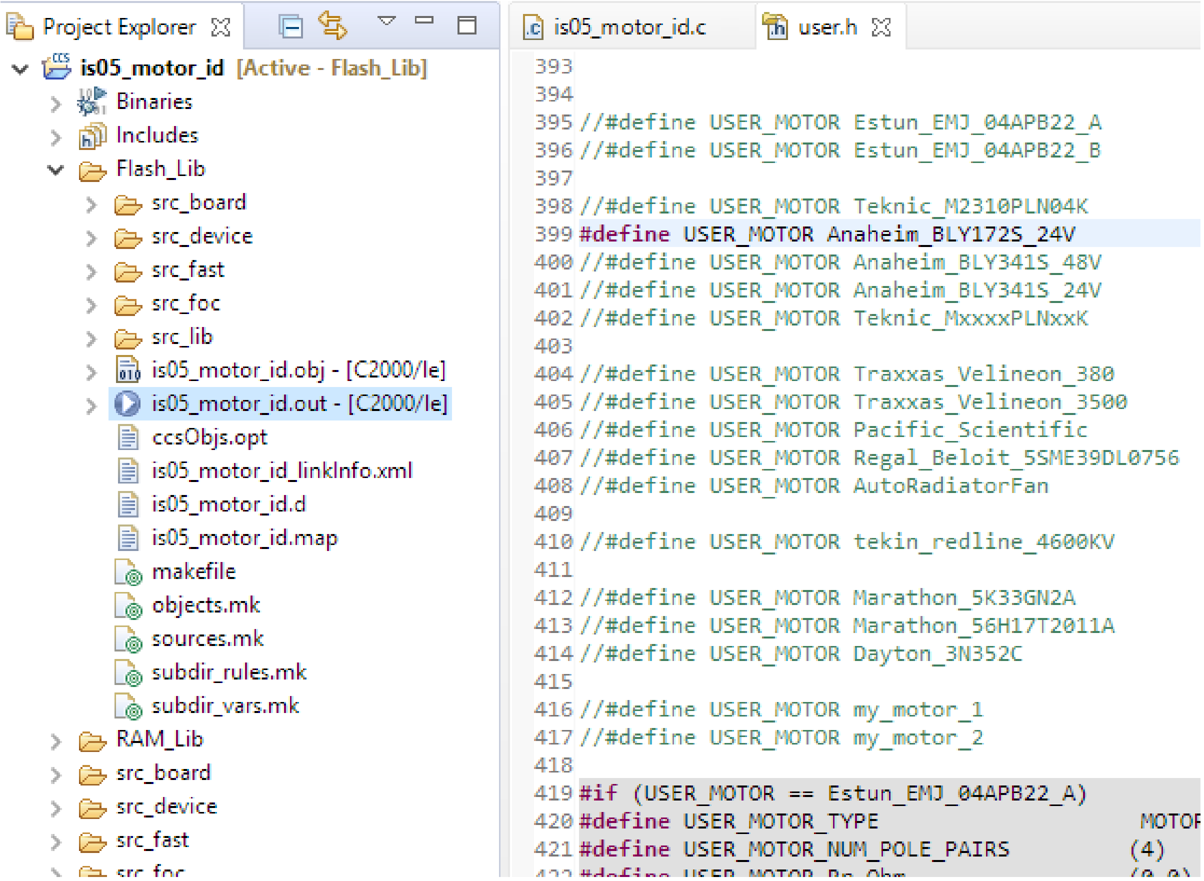
Connected to target. Hardware Connected.

Figure 6 – DRV8320RS Registers tab.

3.3.4 Motor ID lab example

This section will walk through running the InstaSPIN Motor ID lab (lab 05) on a LAUNCHXL-F280049C with a BOOSTXL-DRV8320RS. The motor used for the demo is an Anaheim Automation BLY172S-24V-4000. The hardware was set up as described in the readme and hardware guide documents.

Following the instructions in the InstaSPIN-FOC lab guide, the *is05_motor_id* lab was imported into CCSv8, the user.h was updated to reflect the motor, and the Flash_Lib configuration was built.

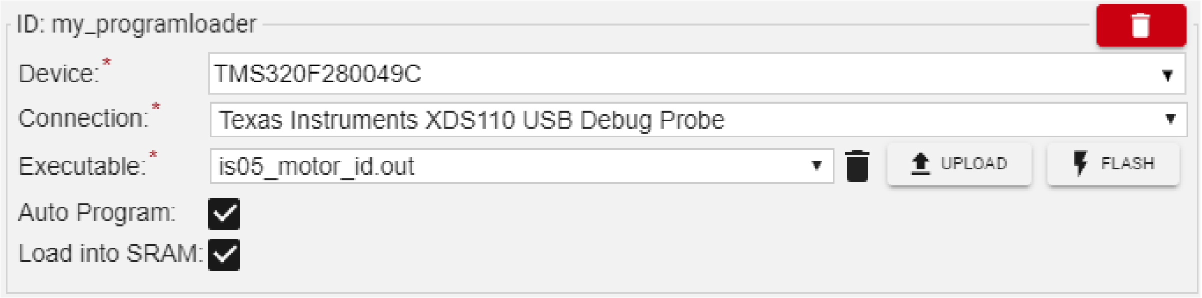


- is05_motor_id [Active - Flash_Lib]
  - Binaries
  - Includes
  - Flash_Lib
    - src_board
    - src_device
    - src_fast
    - src_foc
    - src_lib
    - is05_motor_id.obj - [C2000/le]
    - is05_motor_id.out - [C2000/le]
    - ccsObjs.opt
    - is05_motor_id_linkInfo.xml
    - is05_motor_id.d
    - is05_motor_id.map
    - makefile
    - objects.mk
    - sources.mk
    - subdir_rules.mk
    - subdir_vars.mk
  - RAM_Lib
  - src_board
  - src_device
  - src_fast
  - src_foc

```


//#define USER_MOTOR Estun_EMJ_04APB22_A
//#define USER_MOTOR Estun_EMJ_04APB22_B

//#define USER_MOTOR Teknic_M2310PLN04K
#define USER_MOTOR Anaheim_BLY172S_24V
//#define USER_MOTOR Anaheim_BLY341S_48V
//#define USER_MOTOR Anaheim_BLY341S_24V
//#define USER_MOTOR Teknic_MxxxxPLNxxK

//#define USER_MOTOR Traxxas_Velineon_380
//#define USER_MOTOR Traxxas_Velineon_3500
//#define USER_MOTOR Pacific_Scientific
//#define USER_MOTOR Regal_Beloit_5SME39DL0756
//#define USER_MOTOR AutoRadiatorFan

//#define USER_MOTOR tekin_redline_4600KV

//#define USER_MOTOR Marathon_5K33GN2A
//#define USER_MOTOR Marathon_56H17T2011A
//#define USER_MOTOR Dayton_3N352C

//#define USER_MOTOR my_motor_1
//#define USER_MOTOR my_motor_2

#if (USER_MOTOR == Estun_EMJ_04APB22_A)
#define USER_MOTOR_TYPE                        MOTOF
#define USER_MOTOR_NUM_POLE_PAIRS             (4)
#define USER_MOTOR_Rs_Ohm                     (0.0)
```

Figure 7 - CCS workspace showing motor ID lab project.

Opening the C2000Ware_MotorControlSDK_InstaSPIN_GUI project properties in GUI Composer as described earlier in this document, the new .out was added to the project and the Device and Connection were confirmed to be correct for the board being used.



ID: my_programloader

Device:* TMS320F280049C

Connection:* Texas Instruments XDS110 USB Debug Probe

Executable:* is05_motor_id.out

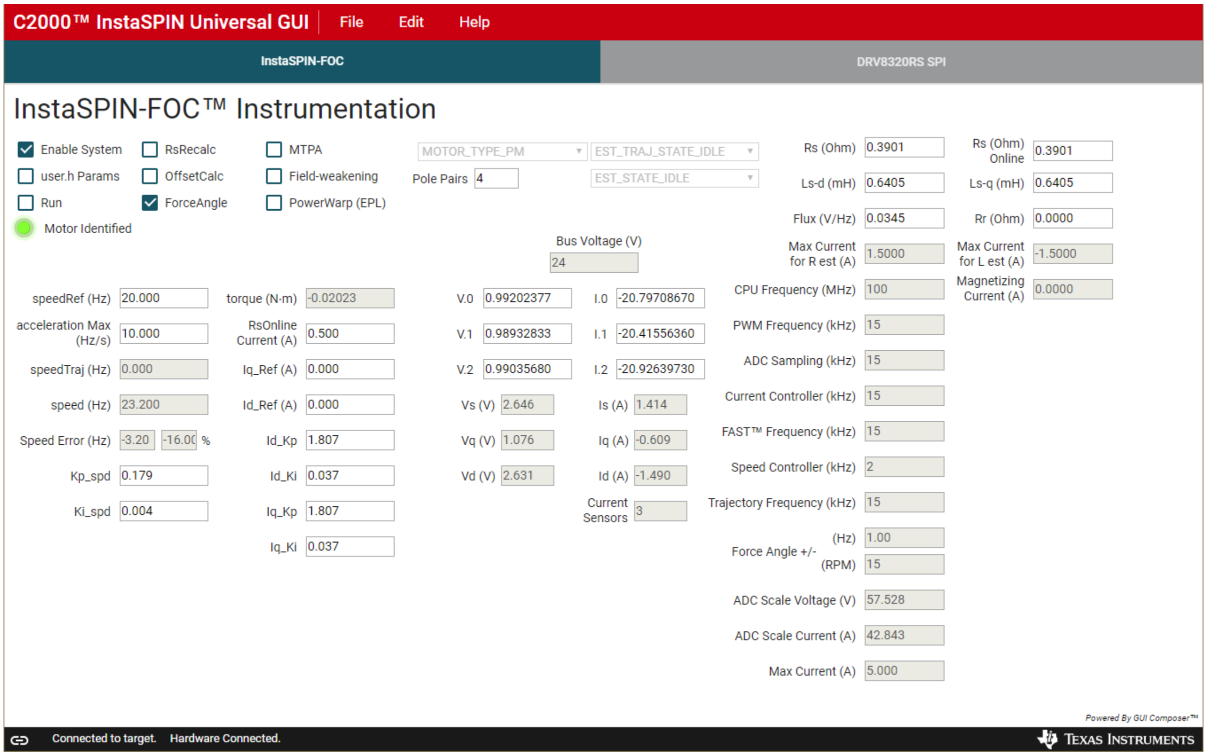
Auto Program: ☑

Load into SRAM: ☑

Figure 8 - GUI Composer project properties for motor ID lab on LAUNCHXL-F280049C with a BOOSTXL-DRV8320RS. Note this is also the default configuration.

After clicking “OK” and saving the application, the application was launched in the browser by hitting the run button. Once the application was loaded and running (LED on LaunchPad could be observed blinking), the motor identification instructions from the lab guide could be followed.

This started with checking the “Enable System” checkbox (setting motorVars.flagEnableSys) and then checking the “Run” checkbox (setting motorVars.flagRunIdentAndOnLine). Once motor identification ran and completed, the “Run” checkbox was automatically unchecked and the “Motor Identified” status indicator was set.



InstaSPIN-FOC™ Instrumentation

☑ Enable System
☐ user.h Params
☐ Run
● Motor Identified
☐ RsRecalc
☐ OffsetCalc
☑ ForceAngle
☐ MTPA
☐ Field-weakening
☐ PowerWarp (EPL)

MOTOR_TYPE_PM
Pole Pairs 4
EST_TRAJ_STATE_IDLE
EST_STATE_IDLE
Bus Voltage (V) 24

speedRef (Hz) 20.000
acceleration Max (Hz/s) 10.000
speedTraj (Hz) 0.000
speed (Hz) 23.200
Speed Error (Hz) -3.20 -16.0(%
Kp_spd 0.179
Ki_spd 0.004

torque (N·m) -0.02023
RsOnline Current (A) 0.500
Iq_Ref (A) 0.000
Id_Ref (A) 0.000
Id_Kp 1.807
Id_Ki 0.037
Iq_Kp 1.807
Iq_Ki 0.037

V.0 0.99202377
V.1 0.98932833
V.2 0.99035680
Vs (V) 2.646
Vq (V) 1.076
Vd (V) 2.631

I.0 -20.79708670
I.1 -20.41556360
I.2 -20.92639730
Is (A) 1.414
Iq (A) -0.609
Id (A) -1.490
Current Sensors 3

Rs (Ohm) 0.3901
Ls-d (mH) 0.6405
Flux (V/Hz) 0.0345
Max Current for R est (A) 1.5000
CPU Frequency (MHz) 100
PWM Frequency (kHz) 15
ADC Sampling (kHz) 15
Current Controller (kHz) 15
FAST™ Frequency (kHz) 15
Speed Controller (kHz) 2
Trajectory Frequency (kHz) 15
Force Angle +/- (Hz) 1.00
Force Angle +/- (RPM) 15
ADC Scale Voltage (V) 57.528
ADC Scale Current (A) 42.843
Max Current (A) 5.000

Rs (Ohm) Online 0.3901
Ls-q (mH) 0.6405
Rr (Ohm) 0.0000
Max Current for L est (A) -1.5000
Magnetizing Current (A) 0.0000

Connected to target. Hardware Connected.

Figure 9 – GUI showing the status of variables after running the motor identification lab.

The newly defined motor parameters could be noted in the GUI and recorded to update the user.h. To run the motor, the “Run” checkbox was checked again. At this point you can try changing the speedRef_Hz or the accelerationMax_Hzps values and observing their effects. When finished, the “Run” box was unchecked to disable the PWM output, and the GUI window could be closed.

4 Next steps

For next steps, you can continue to try the GUI with different InstaSPIN labs, following the lab guide instructions. Note that not all lab projects may be compatible with the GUI. For example, at this time the GUI does not support the dual motor lab (lab 11).

When the GUI is configured to your liking, you may want to consider exporting it as a standalone application so that it can be run offline by going to “File”->”Export”->”As Stand-Alone App” in GUI Composer and following the directions for your OS. Other customizations can be made to the GUI as well. Use the “Help” menu in GUI Composer to help you get started.

5 Support

For questions about InstaSPIN-FOC, the MotorControl SDK, and the InstaSPIN Universal GUI, visit the C2000 microcontrollers forum in the TI E2E Community. For general GUI Composer issues, visit the Code Composer Studio™ forum.