Single-Axis Motor Control and PFC Kit

User's Guide

Texas Instruments

Literature Number: SPRUI27
July 2015

Contents

List of Figures

List of Tables

Single-Axis Motor Control and PFC Kit

1 Introduction

This guide presents the steps to run the 1AxisMtrPfc5x kit with the software supplied through controlSUITE™.

The software is at the following path:

controlSUITE\developement_kits\TIDM-1AXISMTR-PFC-5x_v1.0

The software for the interleaved PFC for this demonstration is derived from a project at the following path:

controlSUITE\development_kits\ILPFC_v1.0

For the motor control demonstration, the kit provides only HVPM_Enhanced_Sensorless and is derived from a project at the following path:

controlSUITE\developement_kits\HVMotorCtrl+PfcKit_v2.1

In this location, the following fixed-point motor control projects based on Piccolo (TMS320F2803x) are available for the HVMotorPfckit:

- HVACI_Scalar: Scalar Control of AC Induction Motor
- HVACI_Sensorless: Sensorless Field-Oriented Control of AC Induction Motor
- HVACI_Sensored: Sensored Field-Oriented Control of AC Induction Motor
- HVPM_Sensorless: Sensorless Field-Oriented Control of Permanent Magnet Motor
- HVPM_Sensored: Sensored Field-Oriented Control of Permanent Magnet Motor
- HVBLDC_Sensorless: Sensorless Trapezoidal Control of BLDC Motors
- HVBLDC_Sensored: Sensored Trapezoidal Control of BLDC Motors

You can port any of these projects to the 1AxisMtrPfc5x kit and evaluate the performance of the project.

The guide assumes you have read the *Single-Axis Motor Control and PFC Kit Hardware Reference Guide* (SPRUI28) and ensure you understand the safety measures.

You can find the guide at the following path:

controlSUITE\developement_kits\ TIDM-1AXISMTR-PFC-5x_v1.0\~Docs

controlSUITE, Code Composer Studio, C2000 are trademarks of Texas Instruments.

WARNING

This EVM is to be operated in a lab environment only. TI does not consider it to be a finished end-product fit for general consumer use.

This EVM must be used only by qualified engineers and technicians familiar with risks associated with handling high-voltage electrical and mechanical components, systems, and subsystems.

This equipment operates at voltages and currents that can cause electrical shock, fire hazard, and/or personal injury if not properly handled or applied. Equipment must be used with necessary caution and appropriate safeguards must be employed to avoid personal injury or property damage.

You must identify and understand the voltages before energizing the board and/or simulation. When energized, the EVM or components connected to the EVM must not be touched

Isolation transformers must be used when connecting grounded equipment to the EVM.

2 Hardware Configuration (Motor Control)

To experiment with the digital motor control in the kit, ensure you have the following hardware components:

- TIDM-1AXISMTR-PFC-5X kit
- Three-phase PMSM
- An incremental encoder or sprocket (optional)
- PC with Code Composer Studio™ (CCSv6.x) installed
- Additional instruments such as an oscilloscope, a digital multimeter, a current sensing probe, and function generator. (Read the *Single-Axis Motor Control and PFC Kit Hardware Reference Guide* (SPRUI28) to determine the isolation needs for the equipment.)
- A high-voltage DC power supply (isolating)
- An isolated, 15-V power supply

The following sections present the experimental setup and connection of the system. For a detailed configuration of each component and connection in the system, see the *Single-Axis Motor Control and PFC Kit Hardware Reference Guide* (SPRUI28) and the 1AXISMTRPFC5X kit schematic.

NOTE: Keep all the power supplies to zero unless advised otherwise in this guide or the application report.

The 1AXISMTRPFC5X kit has separate function-specific macro blocks.

To locate the parts on the board, each component has a macro number in brackets followed by a dash and the reference number.

The following list presents the macros and descriptions of what each macro does:

- [Main] – Jumpers, communications (isoCAN), instrumentation (DACs), QEP and CAP connection, and voltage translation
- [M1] – An AC power entry takes AC power from the wall/mains power supply and rectifies it. This macro can then act as an input of the PFC stage or directly generate the DC bus for the inverter.
- [M2] – An auxiliary power supply, 400 V to 5 V, and 15-V module that can generate 15 V, 5-V power for the board from rectified AC power
- [M3] – An isolated USB emulation provides isolated JTAG connection to the controller and can be an isolated SCI when JTAG is not required.
- [M4] – A 2-phase interleaved PFC stage
- [M5] – A 3-phase inverter to enable control of high voltage 3-phase motors
- [M6] – A DC power entry generates 15 V, 5 V, and 3.3 V for the board from DC power fed through the DC-jack using the power supply with the board.
- [M7] – An F2805x CPU
- [M8] – The PWMDACs

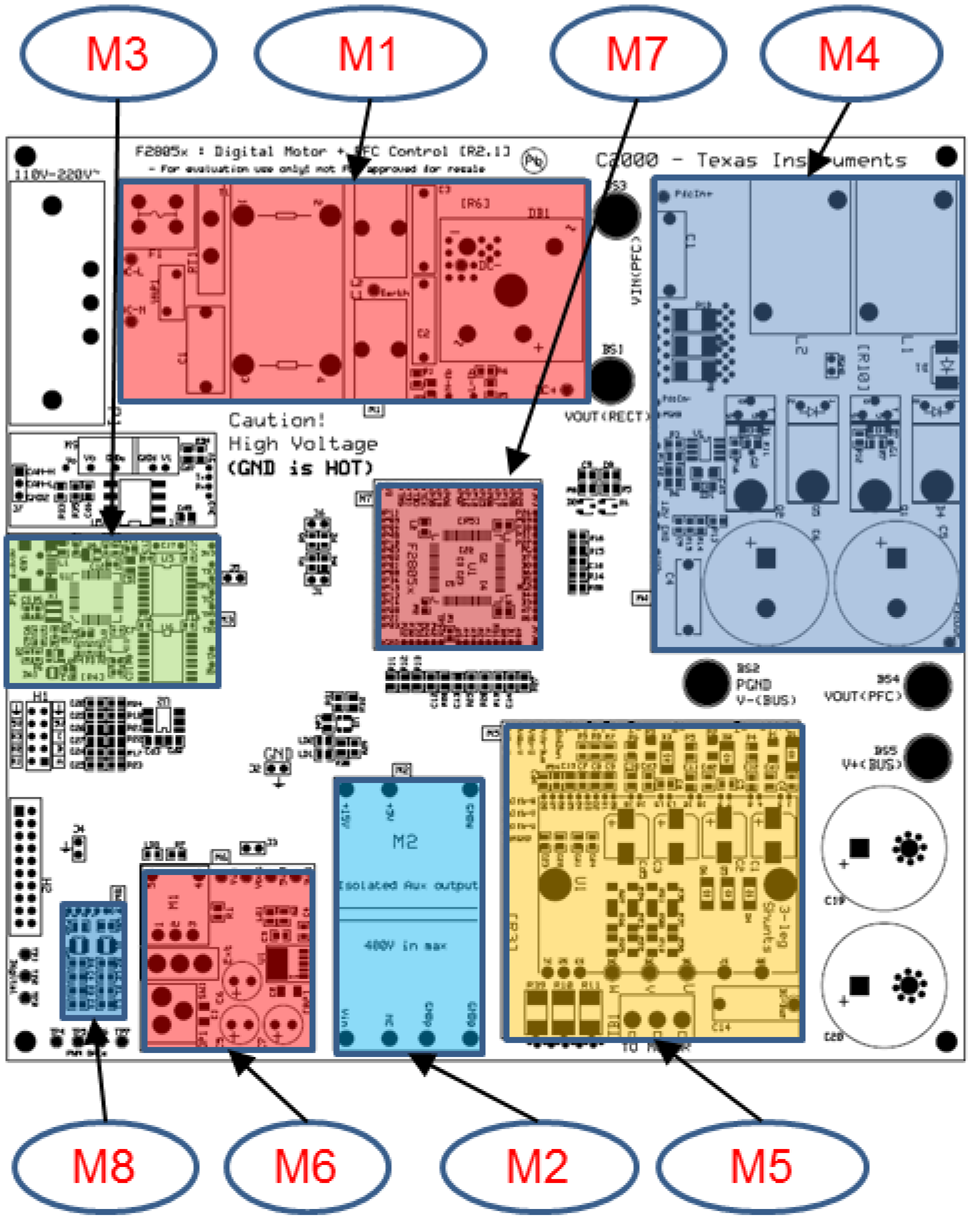


Figure 1. Layout of 1AXISMTRPFC5X Kit With Various Macros

Table 1. Macros in TIDM-1AXISMTR-PFC-5X Kit

Macro	Description
[Main]	Jumpers, macro interconnections, and PGA resistors
[M1]	An AC power entry
[M2]	An auxiliary power supply, 400 V to 5 V, and 15 V
[M3]	Isolated USB emulation
[M4]	A 2-phase, interleaved PFC stage
[M5]	A 3-phase inverter
[M6]	A DC-power entry
[M7]	CPU F2805x
[M8]	PWMDACs

The 1AXISMTRPFC5X platform has the following main power domains:

- The controller power domain that provides 15 V, 5 V, and 3.3 V to the microcontroller, the logic, and the sensing circuit on the board. The following sources can drive power for this domain:
  - The 15-V DC power supply connecting to the DC jack ([M6]-JP1) on the DC power entry macro

    NOTE: TI recommends this power source for all experiments.

  - The aux power supply module [M2] on the board that can generates 15-V and 5-V DC from the rectified AC.
- DC inverter bus power is a high voltage line that provides the voltage to the inverter to generate the 3-phase AC to control the motor. The following sources can provide power to the inverter:
  - An external isolated variable DC power supply (maximum 350 V)

    NOTE: TI recommends this source for all experiments.

  - An isolated variable AC power supply
  - AC Main ([Mains]-P1, 110-V or 220-V AC power supply)

NOTE: For your safety, TI recommends using a variac (variable AC transformer) and an isolator when starting to use this power source.

NOTE: Three-phase induction motors are typically rated at 220-V AC, so you must use 320-V DC-bus voltage. When using 110-V AC power source to generate the DC bus for the inverter, the motor runs properly only below half the rated speed without saturating the PID regulators in the control loop. You can run the PFC on the HV DMC drive platform as a boost converter to increase the DC bus voltage level or directly connect to a DC power supply.

3 Setting Up the Motor Control Experiment Hardware

For projects concerning only motor control, use the DC power entry macro to get the voltages for the controller. For the DC bus for the inverter, use either of the following options:

1. Install the jumpers [Main]-J5 for the JTAG reset line.
2. Connect a USB cable to connector [M3]-JP1 to enable isolated JTAG emulation to the C2000™ device.

NOTE: [M3]-LD1 turns on.

3. Ensure that [M6]-SW1 is off.
4. Connect the 15-V DC power supply to [M6]-JP1.
5. Turn on [M6]-SW1.

NOTE: [M6]-LD1 turns on.

6. Connect the motor to the [Main]-TB1 terminals after completing the first incremental build step.
7. Apply the DC bus power to the inverter when instructed. The following options provide DC bus power to the inverter:
   - To use an external isolated DC power supply, do the following:
     (a) Set the power supply output to zero.
     (b) Connect [Main]-BS5 and BS2 to DC power supply and ground, respectively.
     (c) Connect [Main]-BS5 and [Main]-BS4 as the CPU senses bus voltage information from the PFC output stage where [Main]-BS4 is. (See Figure 2.)
   - To use AC mains power without a PFC, do the following:
     (a) Connect [Main]-BS1 and [Main]-BS5 to each other using a banana plug cord.
     (b) Connect [Main]-BS5 and [Main]-BS4 as the CPU senses bus voltage information from the PFC output stage where [Main]-BS4 is. (See Figure 3.)
     (c) Connect one end of the AC power cord to [Main]-P1.
     (d) Connect the other end to the output of an isolated variable AC supply.

     NOTE: If variable AC supply is unavailable, connect the other end of the cord to a variac connected to AC mains through an isolation transformer. See Figure 4.

     (e) Ensure the variac output is set to zero before connecting to wall supply.
   - To use AC mains power with a PFC, do the following:
     (a) Connect [Main]-BS1 and BS3 to each other using banana plug cord.
     (b) Connect [Main]-BS5 and [Main]-BS4.
     (c) Connect one end of the AC power cord to [Main]-P1.
     (d) Connect the other end to the output of an isolated variable AC supply.

     NOTE: If variable AC supply is unavailable, connect the other end of the cord to a variac connected to the AC mains through an isolation transformer.

     (e) Ensure the variac output is set to zero.
     (f) Connect the variac to the wall supply through an isolator. (See Figure 4.)

CAUTION

An isolator is required if measurement equipment is connected to the board.

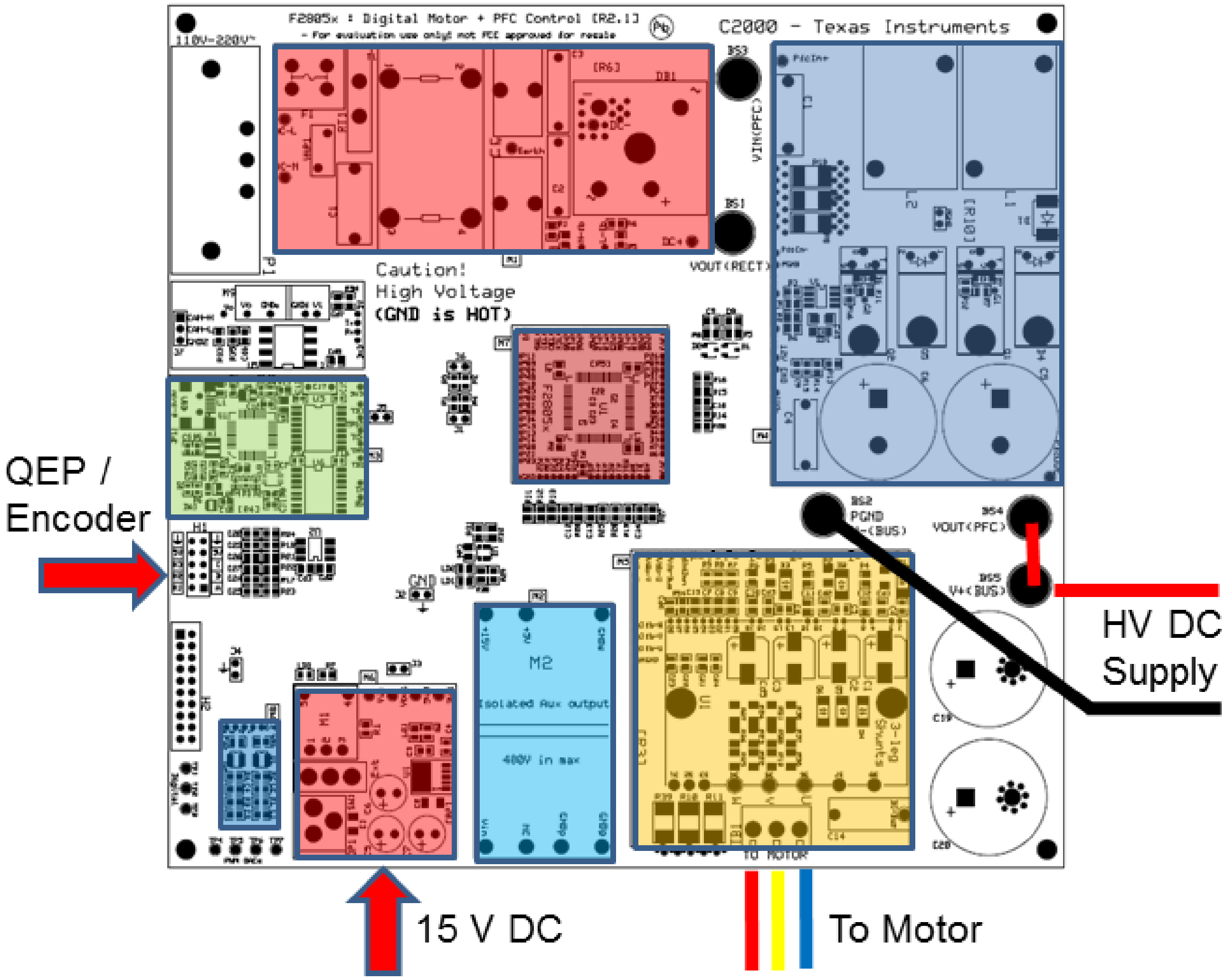


Figure 2. Using an External Power Supply to Provide the DC Bus Voltage for the Inverter

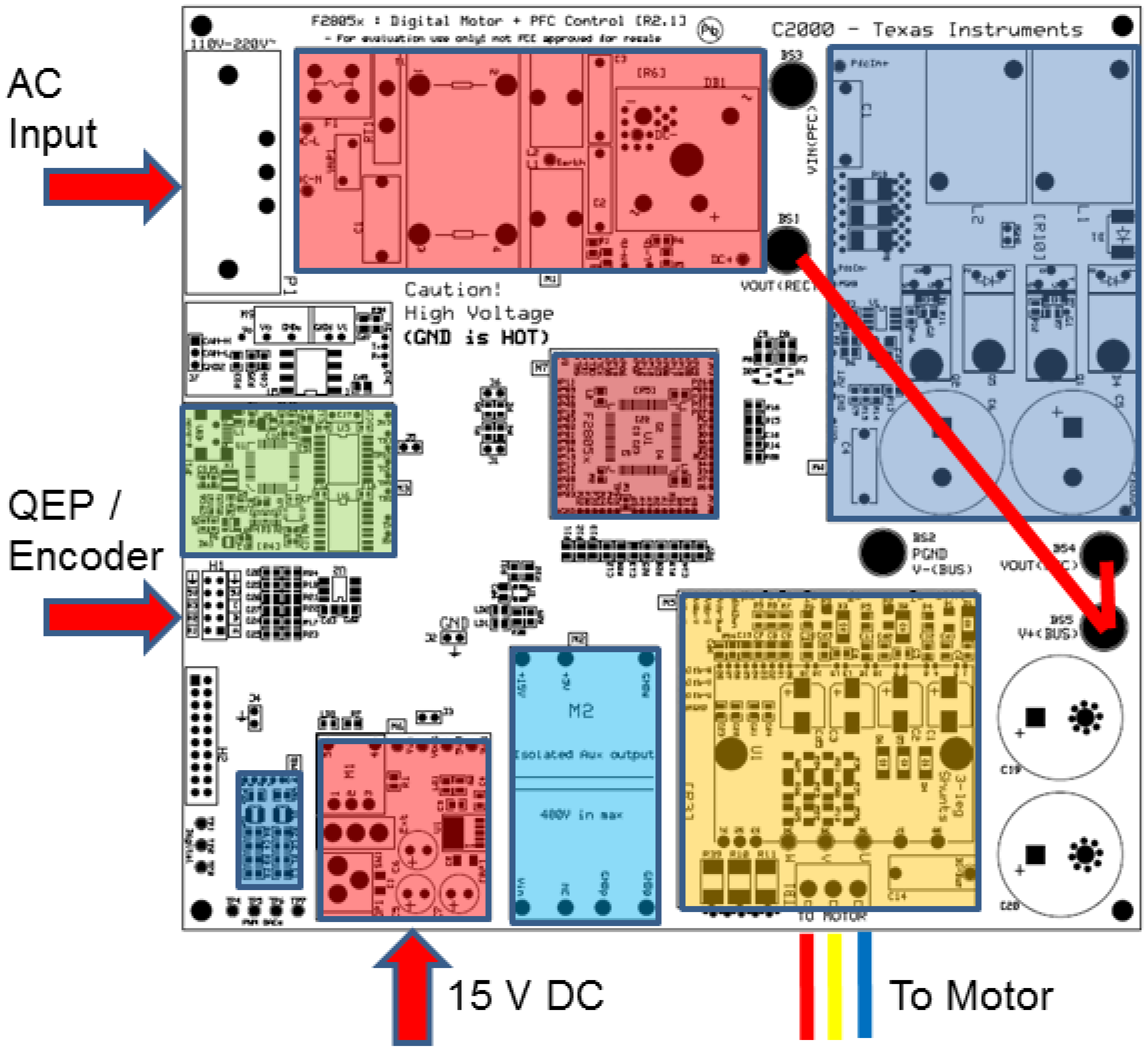


Figure 3. Using an AC Input to Generate DC Bus Voltage for the Inverter, Bypassing PFC

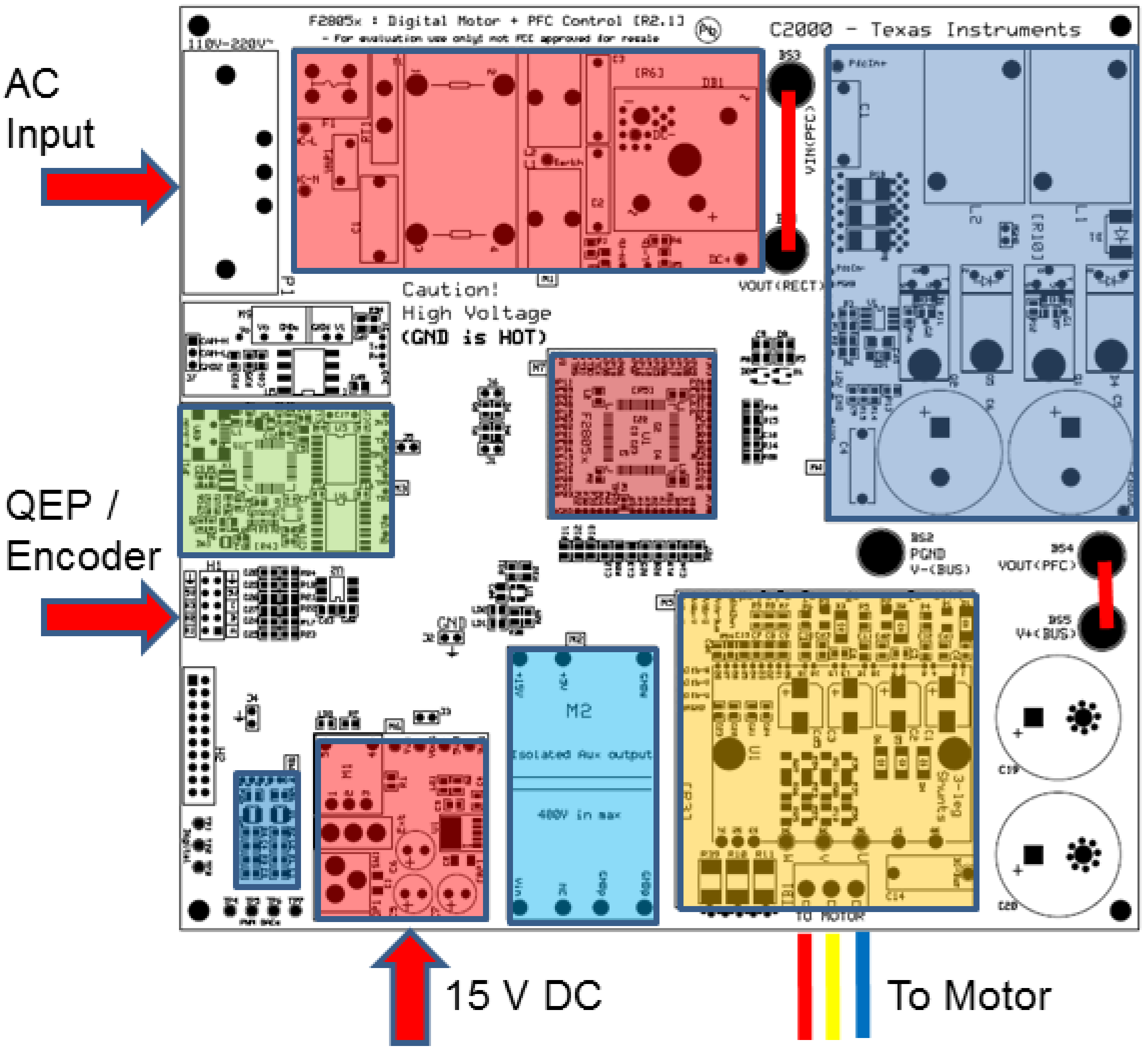


Figure 4. Using AC Input and PFC to Generate DC Bus Voltage for Inverter

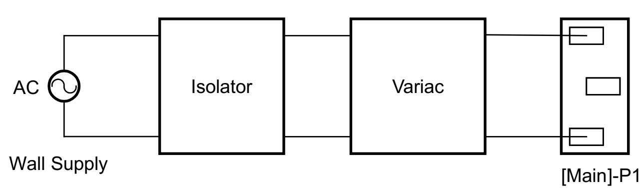


Figure 5. Wiring Diagram of the Kit With an Isolator and a Variac

WARNING

To avoid injury, use caution with DC bus capacitors as they remain charged after disconnecting the mains supply.

4 Setting Up the Software for 1AxisMtrPfc5x Kit Projects

4.1 Installing Code Composer and controlSUITE

If not already installed, install Code Composer v6.x or later.

To install code composer, do the following:

1. Go to http://www.ti.com/controlsuite.
2. Run the controlSUITE installer. (Let the installer to download and update any automatically checked software for C2000.)
3. Program the FTDI chip for xds100 emulator on board.
4. Follow the instructions in this TI E2E forum post. (If this link breaks, search https://e2e.ti.com.)

4.2 Setting Up Code Composer Studio to Work with the 1AxisMtrPfc5X kit (R2.1)

1. Open Code Composer Studio v6.

NOTE: When Code Composer Studio opens, the workspace launcher may appear and ask to select a workspace location. Workspace is a location on the hard drive where the user settings for the IDE (that is, which projects are open, what configuration is selected, and so forth are saved.) This location can be anywhere on the disk; the location in Step 3 is for reference. If this is not your first-time running Code Composer, this dialog may not appear.

2. Click Browse…
3. Create the following path: *C:\c2000 projects\CCSv6_workspaces\1AxisMtrPfc5x_workspace.*
4. Uncheck the box labeled *Use this as the default and do not ask again*.
5. Click OK.

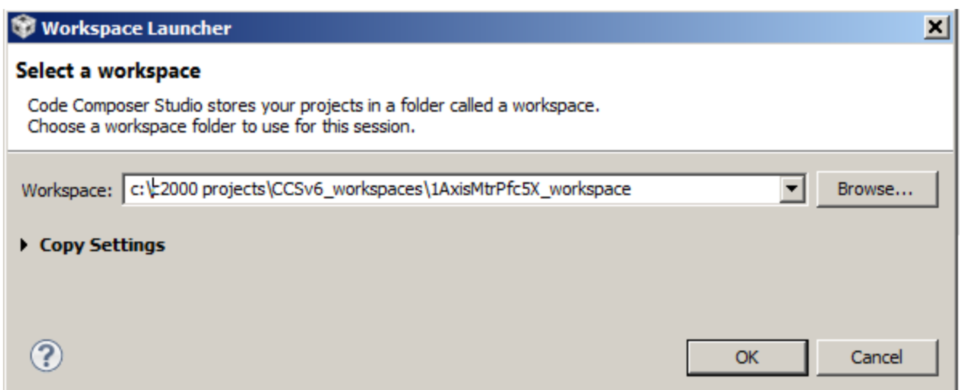


Figure 6. Workspace Launcher

NOTE: A Getting Started tab opens with links to create a new project, import an existing project, watch a tutorial on CCS, and so forth. Click Import Project to skip to Step 10 or close the Getting Started tab to configure Code Composer for the MCU to which it connects. To configure Code Composer, you must set up the Target Configuration. The project includes the configurations in xds100v1_F2805x.ccxml. For general information about setting up this configuration file, proceed to the next step.

6. Clicking ViewSet to set a new configuration file.
7. Click Target Configuration to open the Target Configuration window.
8. Click .
9. Enter a name for the new configuration file based on the target device. (See Figure 7.)

NOTE: If you click the box labeled *Use shared location*, you can store this configuration file in a common location by CCS to use with other projects.

10. Click Finish.

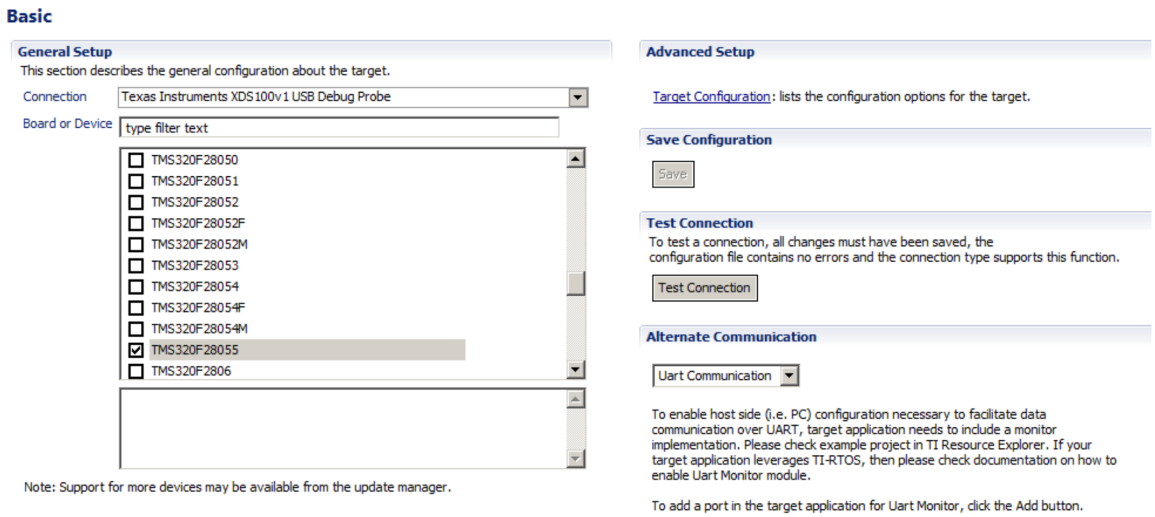


Figure 7. Creating a New Target

11. Select the options.
12. Click Save.
13. Close the window.

NOTE: The configuration is set as the default target configuration for Code Composer.

14. Click View to ensure the configuration is set as the default target configuration.
15. Click Target Configurations.
16. In the section labeled User Defined, right-click on the *xds100v1_F2805x.ccxml* file.
17. Click Set as Default.

NOTE: This tab lets you reuse existing target configurations and link them to specific projects.

18. Click ProjectAdd.
19. Click *Import CCS Projects* to add all the motor control projects to your workspace.
20. Select the root directory *C:\TI\controlSUITE\development_kits\TIDM-1AXISMTR-PFC-5X_v1.0\1AxisMtrPfc5x_PM_Sensorless*.

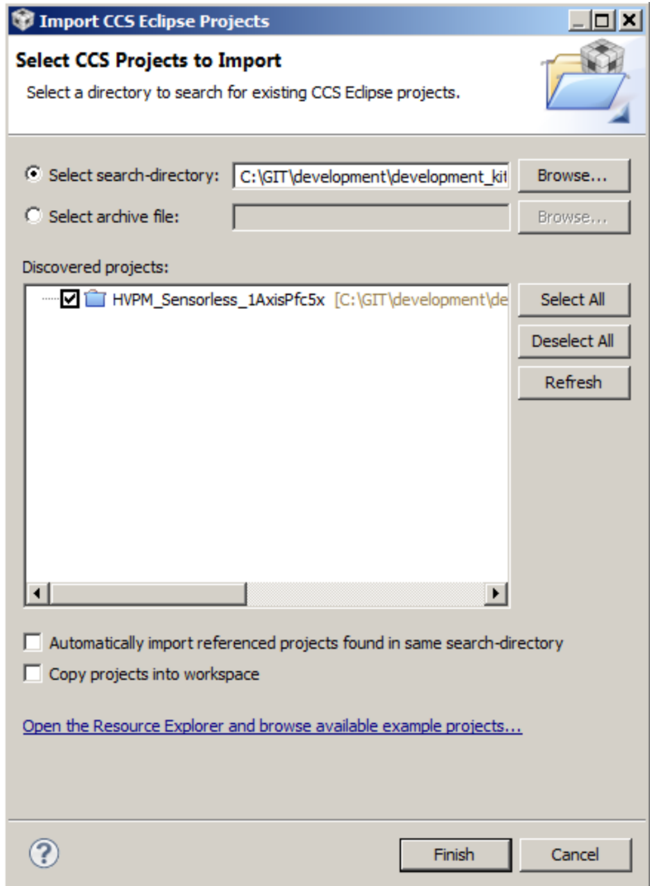


Figure 8. Importing the 1Axis Mtr PFC Project to Workspace

21. If multiple projects exist in this directory, choose the projects to import.
22. Click Finish.

NOTE: The selected projects copy into the workspace. In Figure 8, one project is selected.

4.3 Configuring a Project

To configure a project, do the following:

1. Navigate to the C/C++ Projects tab.
2. Click on the name of the project.
3. Set as an active project.

NOTE: If this is your first time using Code Composer, the xds100v1_F28035 should be the default target configuration.

To verify this and view the xds100_f2805x.ccxml file in the expanded project structure with a box beside it labeled *Active/Default*, do the following:

1. Click View.
2. Click Target Configurations.
3. Edit the existing target configurations.
4. Change the default or active configuration.

To link a target configuration to a project in the workspace, do the following:

1. Right-click on the name of the target configuration.
2. Select Link to Project.

---

4. Right-click on a project.

NOTE: You can configure each project to create code and run in either flash or RAM. For this particular device, use the flash configuration for both experimentation and production. See Figure 9.

---

5. Select Build Configurations.
6. Select Set Active.
7. Select *F2805x_FLASH*.

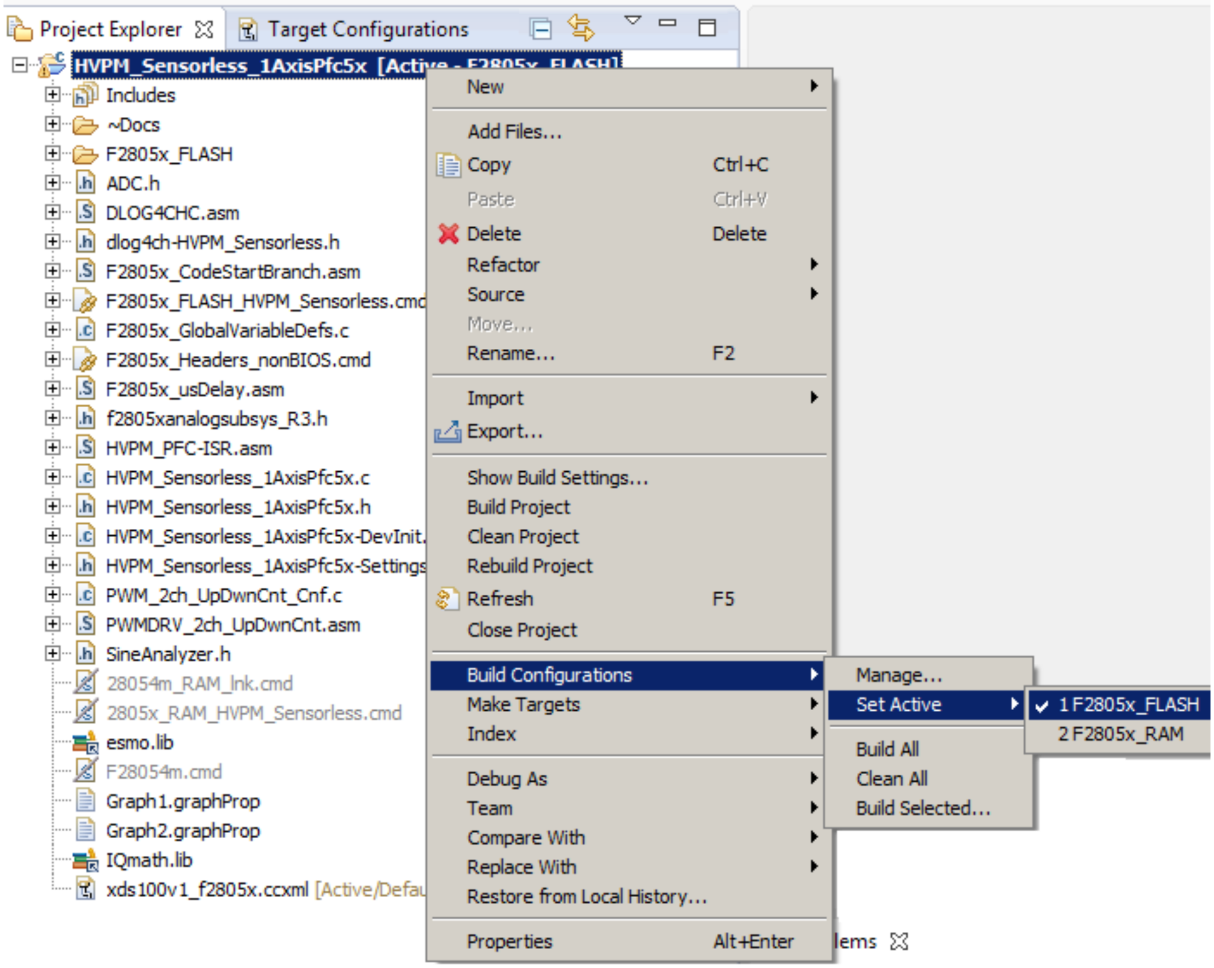


Figure 9. Selecting the F2805x_FLASH Configuration

4.4 Build and Load the Project

The TI motor control software has incremental builds where different components and macro blocks of the system are pieced together to form the system. This structure helps in step-by-step debugging and understanding the system.

To build and load the project, do the following:

1. From the C/C++ Project tab, open the file Pfc_PM_Sensorless-Settings.h.
2. Ensure MOTORBUILD is set to LEVEL1.
3. Save this file.

NOTE: After testing build 1, redefine this variable to move to build 2 and so on until completing the builds.

This project has the following sets of BUILD_LEVELs:

- MOTOR_BUILD for motor control software
- PFC_BUILD for PFC control software

4. Double-click DevInit_F2805x.c.
5. Confirm that GPIO00 to GPIO05 are configured as PWM outputs.
6. Open the [Project-Name].c file.
7. Navigate to the function MainISR().
8. Locate the following piece of code in incremental build 1. (The following code is an example where the datalog point to the space vector generator module.)

```
DlogCh1 = (int16)_IQtoIQ15(svgen_dq1.Ta);
                DlogCh2 = (int16)_IQtoIQ15(svgen_dq1.Tb);
                DlogCh3 = (int16)_IQtoIQ15(svgen_dq1.Tc);
                DlogCh4 = (int16)_IQtoIQ15(svgen_dq1.Ta-svgen_dq1.Tb);
```

9. Confirm the datalog buffers point to the correct variables.

NOTE: These datalog buffers are large arrays that contain value-triggered data that can display as a graph.

In other incremental builds, you can put different variables into this buffer to display as a graph.

10. Right-click the Project Name.
11. Click on Rebuild Project.
12. Watch the Console window for errors in the project.
13. Click  (the debug button) when the build completes successfully.

NOTE: The IDE connects automatically to the target, loads the output file into the device, and changes to the debug perspective.

14. Click Tools.
15. Click Debugger Options.
16. Click Program.
17. Click Memory Load Options.
18. Check Reset the target on program load or restart. (This action enables the debugger to reset the processor each time it reloads program.)
19. Click Remember My Settings. (This action makes this setting permanent.)
20. Click  (Enable silicon real-time mode).

NOTE: This action automatically selects (Enable polite real-time mode) and lets you edit and view variables in real-time.

Do not reset the CPU unless you disable these real-time options.

A message box may appear.

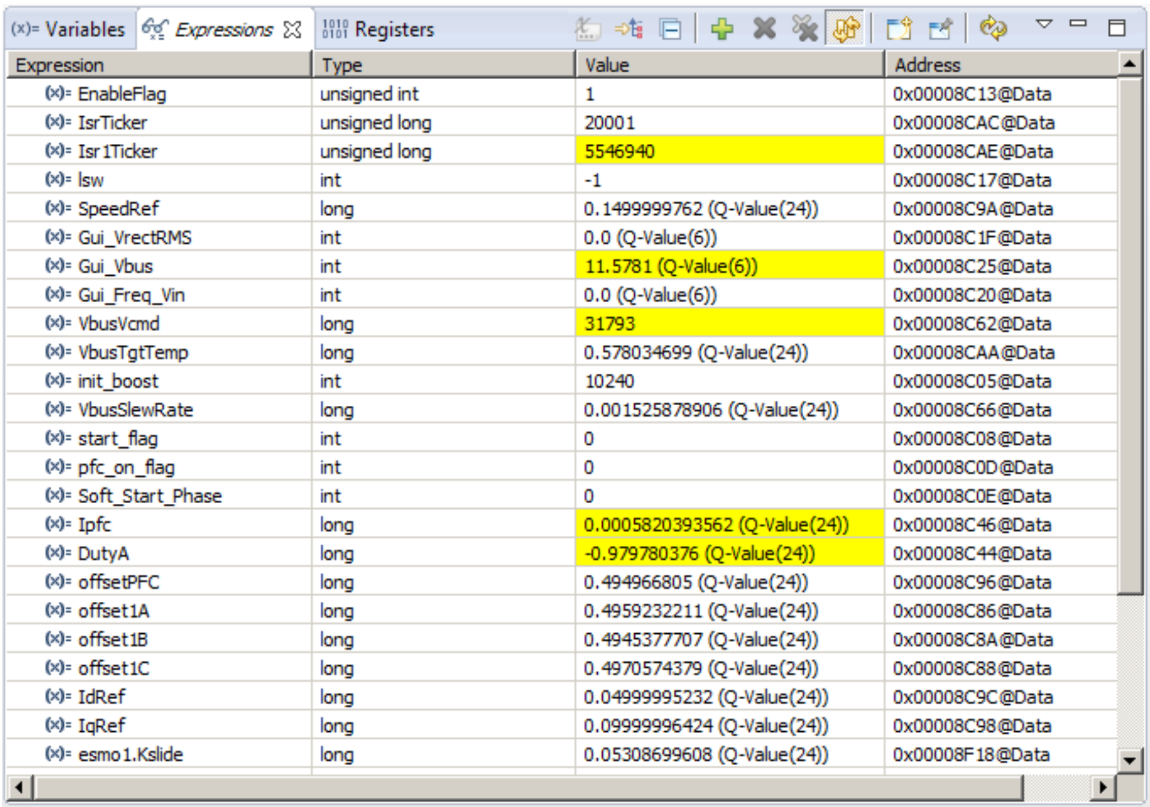
21. Select YES to enable debug event if a message box appears.

NOTE: This action sets bit 1 (DGBM bit) of status register 1 (ST1) to 0. The DGBM bit is the debug enable mask bit.

When the DGBM bit is set to 0, memory and register values can be passed to the host processor to update the debugger windows.

4.5 Setting Up Expression Window and Graphs

1. Click View.
2. Click Expression on the menu bar to open an Expression window to view the variables in the project.
3. Add variables to the watch window. (See Figure 10.)
4. Right-click the variable to change the number format of the variable.
5. Select the appropriate Q format for the variable you want to watch.



Expression	Type	Value	Address
EnableFlag	unsigned int	1	0x00008C13@Data
IsrTicker	unsigned long	20001	0x00008CAC@Data
Isr1Ticker	unsigned long	5546940	0x00008CAE@Data
lsw	int	-1	0x00008C17@Data
SpeedRef	long	0.1499999762 (Q-Value(24))	0x00008C9A@Data
Gui_VrectRMS	int	0.0 (Q-Value(6))	0x00008C1F@Data
Gui_Vbus	int	11.5781 (Q-Value(6))	0x00008C25@Data
Gui_Freq_Vin	int	0.0 (Q-Value(6))	0x00008C20@Data
VbusVcmd	long	31793	0x00008C62@Data
VbusTgtTemp	long	0.578034699 (Q-Value(24))	0x00008CAA@Data
init_boost	int	10240	0x00008C05@Data
VbusSlewRate	long	0.001525878906 (Q-Value(24))	0x00008C66@Data
start_flag	int	0	0x00008C08@Data
pfc_on_flag	int	0	0x00008C0D@Data
Soft_Start_Phase	int	0	0x00008C0E@Data
Ipfc	long	0.0005820393562 (Q-Value(24))	0x00008C46@Data
DutyA	long	-0.979780376 (Q-Value(24))	0x00008C44@Data
offsetPFC	long	0.494966805 (Q-Value(24))	0x00008C96@Data
offset1A	long	0.4959232211 (Q-Value(24))	0x00008C86@Data
offset1B	long	0.4945377707 (Q-Value(24))	0x00008C8A@Data
offset1C	long	0.4970574379 (Q-Value(24))	0x00008C88@Data
IdRef	long	0.04999995232 (Q-Value(24))	0x00008C9C@Data
IqRef	long	0.09999996424 (Q-Value(24))	0x00008C98@Data
esmo1.Kslide	long	0.05308699608 (Q-Value(24))	0x00008F18@Data

Figure 10. Configuring the Watch Window for Fixed-Point Devices

Alternately, you can import a group of variables into the Expressions window as follows:

1. Click View.
2. Select Scripting Console.
3. Within Scripting Console, click Open.
4. Browse to the root directory of the project.
5. Click Variables_1AxisMtrPfc5x.js.
6. Click open to import the variables in Figure 10.
7. Click (Continuous Refresh) in the watch window.

NOTE: This action enables the window to run with real-time mode.

8. Click the down arrow in this watch window.
9. Select Customize Continuous Refresh Interval.

10. Edit the refresh rate of the watch window.
11. Open and set up time graph windows to plot the data log buffers as in Figure 11.

NOTE: The datalog buffers point to different system variables depending on the build level. The buffers let you inspect the variables and judge the performance of the system.

Alternatively, you can import graph configurations files in the project folder, but these files are unsupported by all CCS4 versions.

NOTE: If a second graph window is used, you could import Graph2.prop; the start Addresses must be DLOG_4CH_buff3 and DLOG_4CH_buff4.

The default dlog.prescaler is set to 5, that lets the dlog function to log only one out of every five samples.

To import graph configuration files in the project folder, do the following:

1. Click Tools.
2. Click Graph.
3. Click DualTime...
4. Select Import.
5. Navigate to *C:\TI\ControlSUITE\developement_kits\TIDM-1AXISMTR-PFC-5X_v1.0\<project. directory>* >.
6. Select Graph1.graphProp. (See Figure 11.)
7. Click OK.

NOTE: This action adds the graphs to your debug perspective.

---

8. Click .

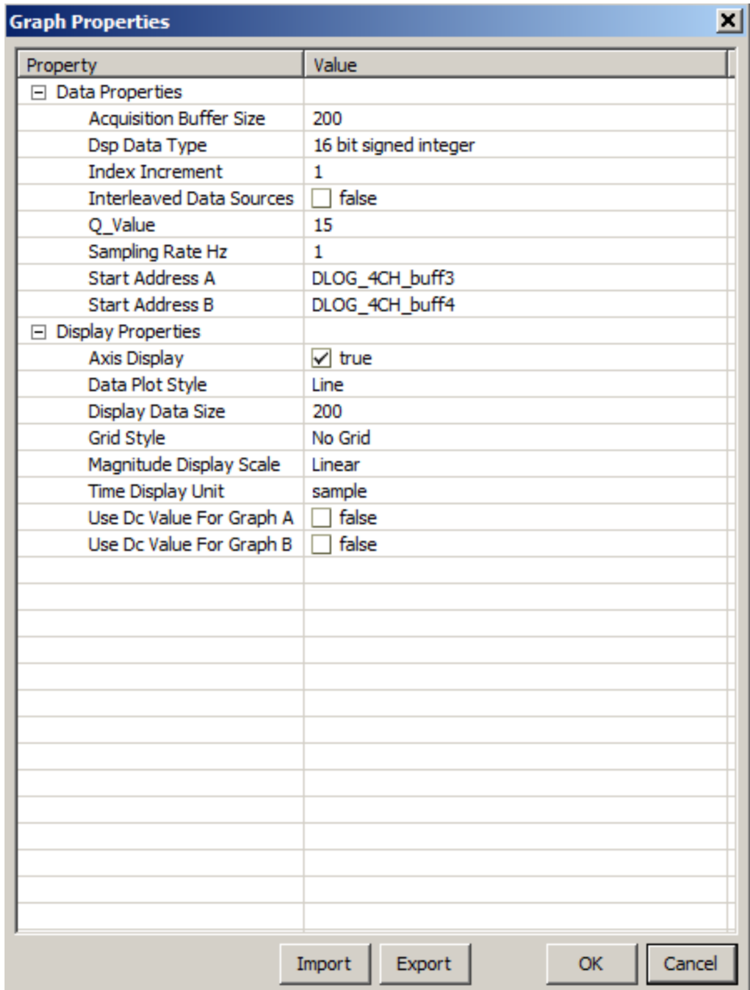


Figure 11. Graph Window Settings

4.6 Run the Code

To run the code, do the following:

1. Click in the Debug tab.

NOTE: The project runs and the values in the graphs and watch window update continuously. The following screen captures are typical of the CCS perspective. You may want to resize the windows.

2. Click Target.
3. Click Reset.
4. Click (Reset CPU) to reset the processor.
5. Click Target.
6. Click (Terminate All) to halt the program and disconnect Code Composer from the MCU.

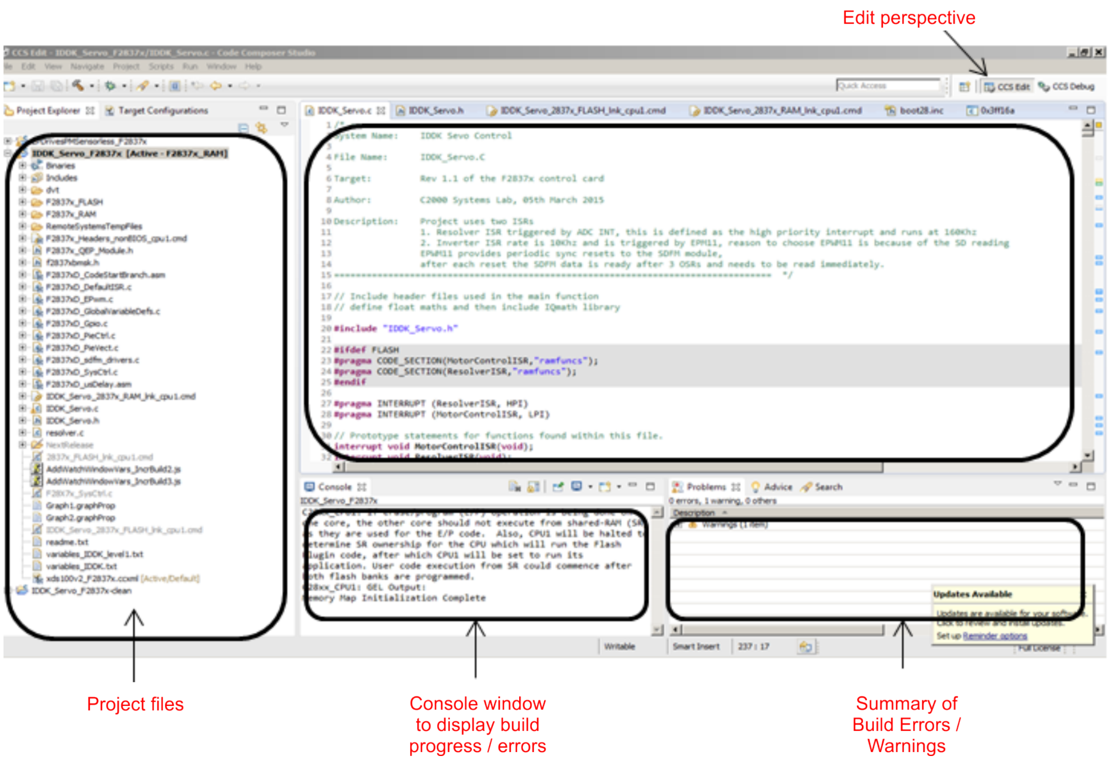


Figure 12. CCS IDE Showing Edit Perspective

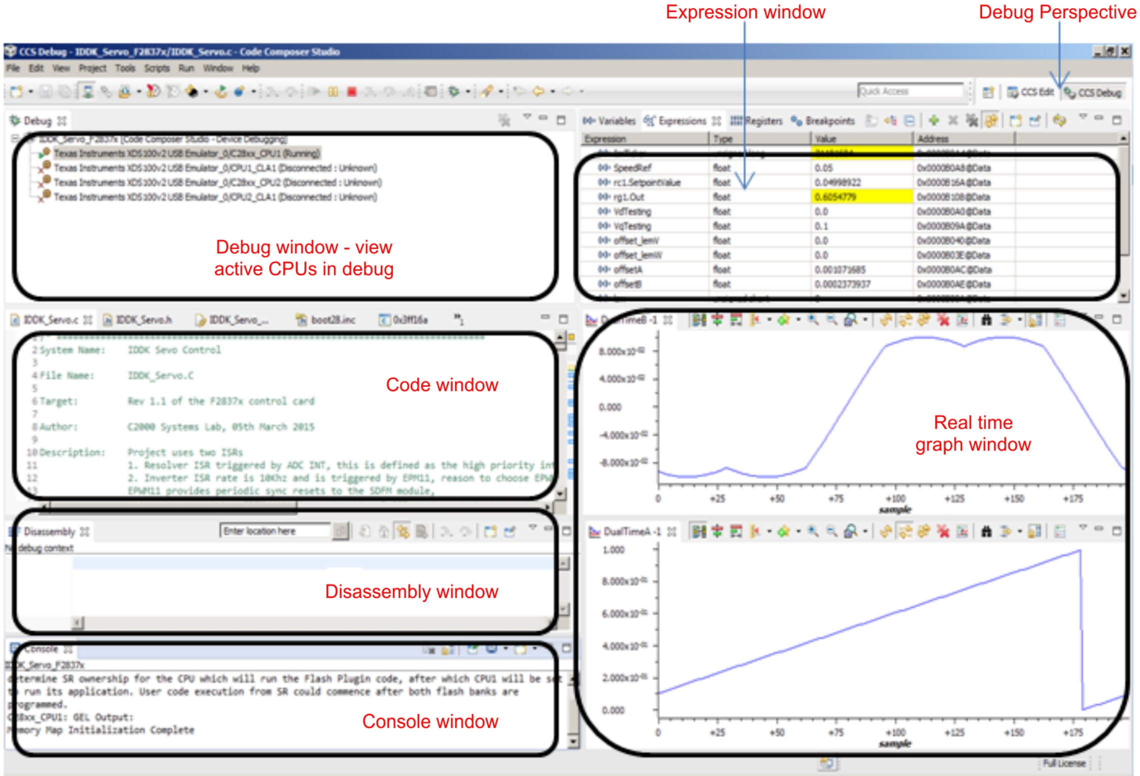


Figure 13. CCS IDE Showing Debug Perspective

4.7 Rebuilding the Project

Terminating the debug session each time you change or run the code is unnecessary.

After rebuilding the project,

1. Click Target.
2. Click Reset.
3. Click (Reset CPU).
4. Click Target.
5. Click Reset.
6. Click (Restart).
7. Enable real-time options
8. Disable real-time options when complete.
9. Reset the CPU.
10. Terminate the project if you change the target device or the configuration (RAM to flash or flash to RAM for example and before shutting down CCS).
11. Customize the project for your motor.
    (a) Change the motor parameters in [motorproject].h.
    (b) Change the PWM switching frequency (ISR frequency).
    (c) Change the base Q-value to balance accuracy.
    (d) Change the CPU bandwidth.
12. Open the lab manual in *C:\TI\controlSUITE\development_kits\TIDM-1AXISMTR-PFC-5X_v1.0\1AxisMtrPfc5x_PM_Sensorless\~Docs.*
13. Start experimenting.

IMPORTANT NOTICE

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265