InstaSPIN-BLDC Lab

Introduction

For this lab we are using the DRV8312 "Low Voltage, Low Current" Power Stage (the DRV8301/2 Kit can also be used) with Piccolo F28035 controlCARD to run the sensorless InstaSPIN-BLDC technique. Here is a list of tasks that must be done to spin motors:

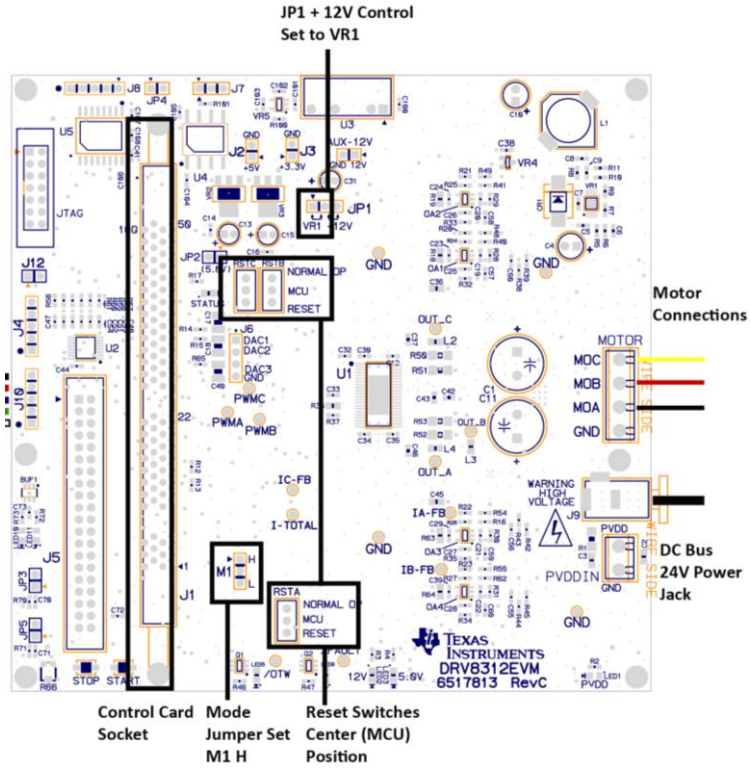
- Make sure all physical connections and jumper settings on the kit are correct.
- Flash the InstaSPIN-BLDC object file (.out) onto the controlCARD
- Launch the GUI
- Start spinning motors!

DRV8312 Setup

Jumpers and switches must be setup properly or the kit will not function correctly!

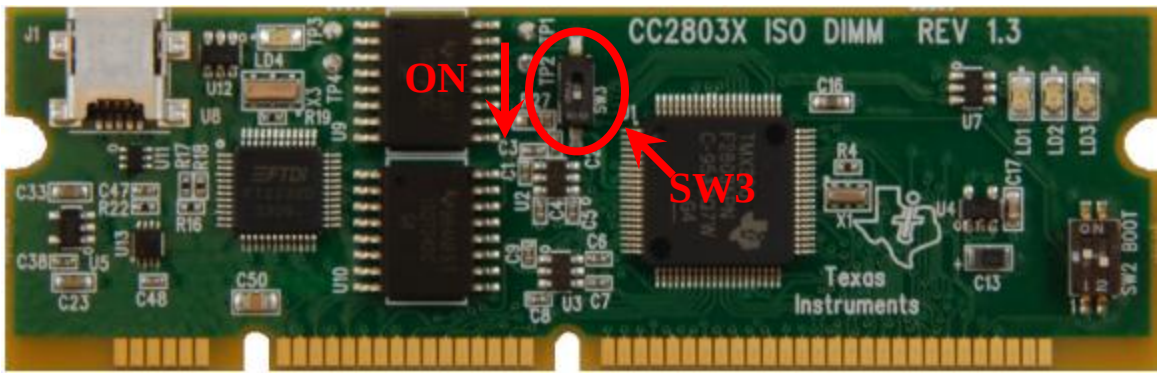
- Three position toggle switches **RSTA**, **RSTB** and **RSTC** must be in the **center/middle** position.
- Jumper **JP1** must be in the **VR1** position.
- Jumper **M1** must be in the **H** position.
- Switch **SW3** on the Control Card must be in the **On** position see **Error! Reference source not found.**.
- Fasten the three motor phase wires into **MOA**, **MOB**, and **MOC** of the kit.
    - **Question:** What do you think will happen if you connect the motor wires in a different order?
        - **A.** The brushes in the motor will overheat, and could cause a fire
        - **B.** The current in the wires will reverse the polarity of the anti-protons in the magnetic field, causing an antimatter explosion
        - **C.** Nothing will happen except the motor may spin in the opposite direction
        - **D.** You could potentially damage the 8312, resulting in your immediate termination from TI.
        - **E.** This is a trick question since the connector is keyed, prohibiting the motor wires from being connected in any other way than what is shown.

        The correct answer is given at the end of this document.



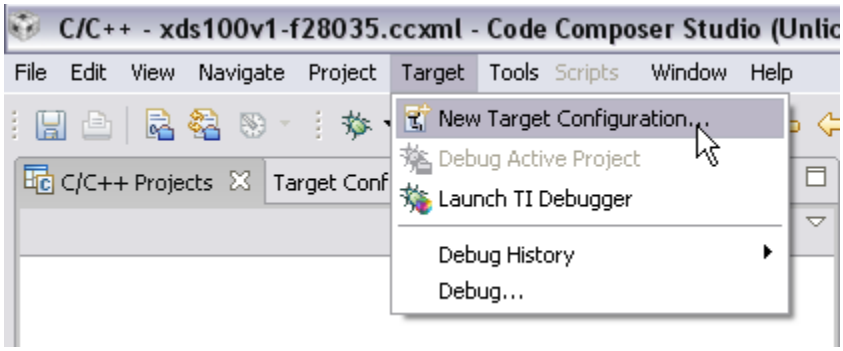
controlCARD Setup & Programming

The included controlCARD comes with the switch (SW3) set to hold the XDS100v1 emulator in JTAG reset (UP/OFF) for use with USB-UART GUI communication. To re-program the Flash using CCS we will need to allow JTAG access by **setting SW3 DOWN/ON**.

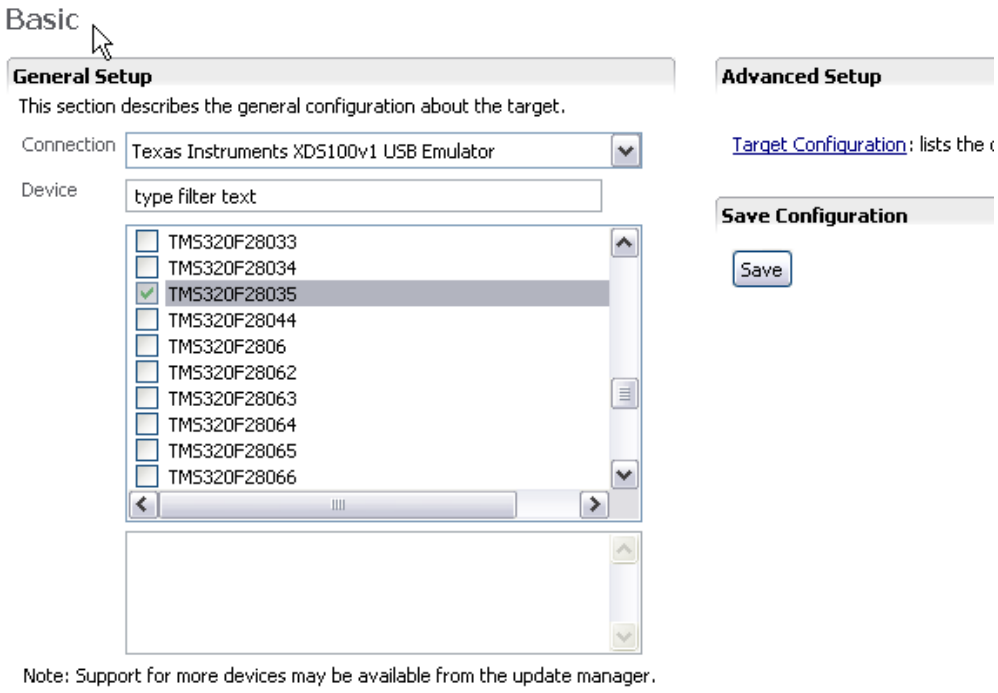


- Insert controlCARD into DRV8312 baseboard
- Plug the 24v power supply into J9 of the DRV8312 kit
- Connect the USB cable from your PC to the controlCARD
- Open Code Composer Studio version 4 (CCS4).

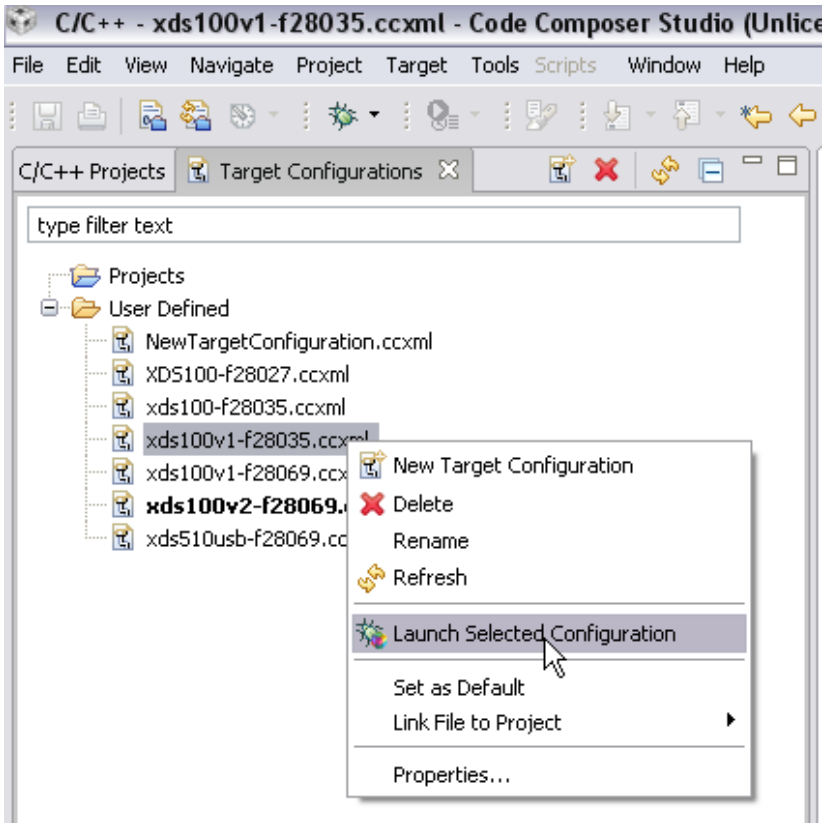
- Target → New Target Configuration



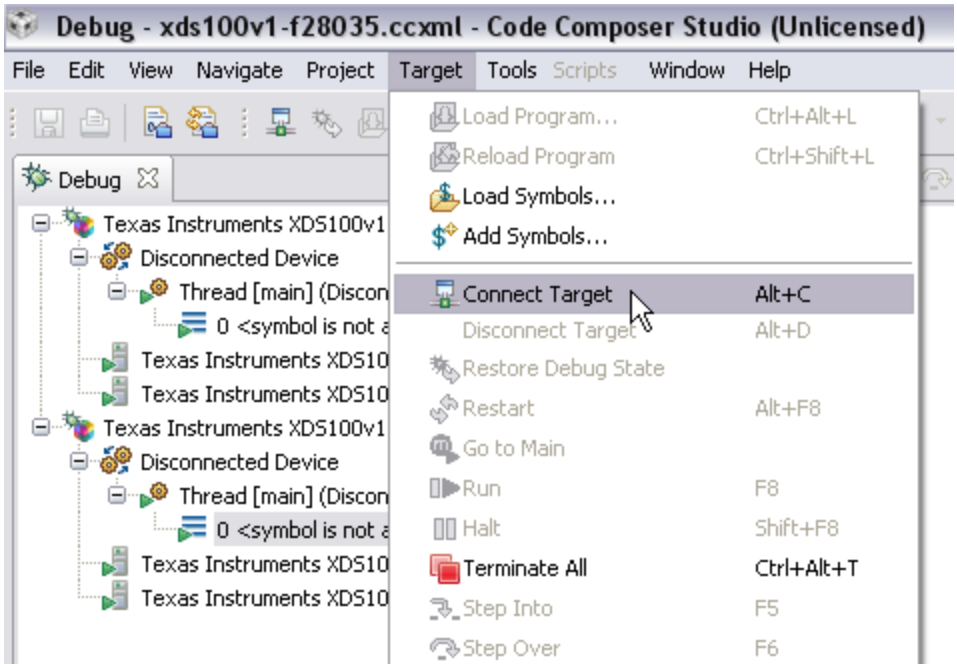
- Name what you like (ex: xds100v1-f28035.ccxml)
- Connection: Texas Instruments XDS100v1 USB Emulator
- Device: TMS320F28035 and click Save



- On the Target Configurations Tab, right click on your new Target and select "Launch Selected Configuration"

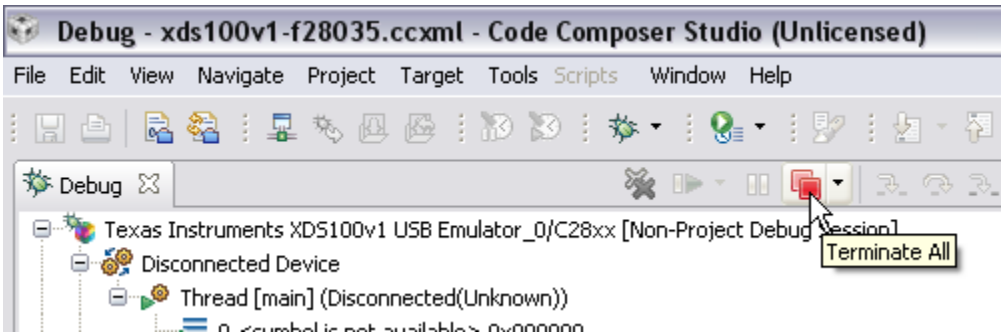


- Select "Target -> Connect Target"



- Choose the object file to flash onto the device by selecting the menu item "Target -> Load Program…"
- Choose the "Browse" button and find the file "InstaSPIN-BLDC_GUI_DRV8312_v100.out" from USB Stick or C:\ti\controlSUITE\development_kits\DRV8312-C2-KIT_v1XX\~GUI

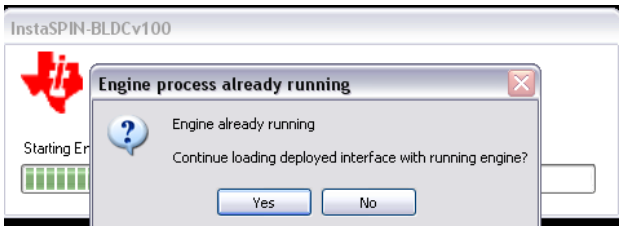
- "Open" then "OK": The flash will be re-programmed
- Select "Terminate All Icon" and close CCStudio



Cycle Power & Run

- Remove 24V power from the DRV8312 kit
- Remove the USB plug from the controlCARD
- Plug the 24v power supply into the DRV8312 kit
- Connect the USB cable between the computer and controlCARD
- Run InstaSPIN-BLDC_GUI_DRV83xx_v100.exe from USB Stick or C:\ti\controlSUITE\development_kits\DRV8312-C2-KIT_v1XX\~GUI

- If you see this message, select "Yes"



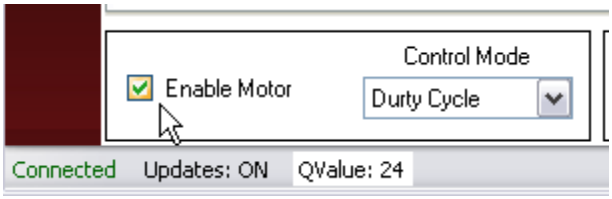
- The GUI should automatically connect. Verify by viewing “Connected” in the lower left hand corner. Also verify the Fault Status, DC Bus, and Over-temperature Warning lights are green.

A full Quick Start Guide for the InstaSPIN-BLDC GUI can be found at:

C:\ti\controlSUITE\development_kits\DRV8312-C2-KIT_v1XX\~GUI

Spinning the motor

- From the Main Tab Click “Enable Motor”

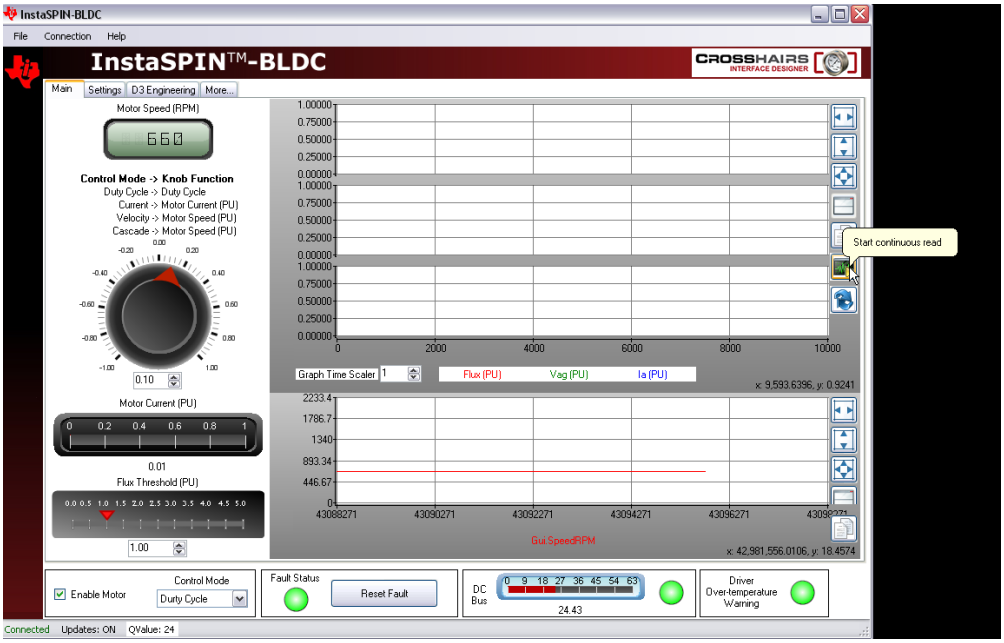


- The motor will start spinning under default settings
  - Duty Cycle = 0.3 PU = 2129 RPM
- **Congratulations! You just did InstaSPIN!!!!**

Tuning the Motor Commutation

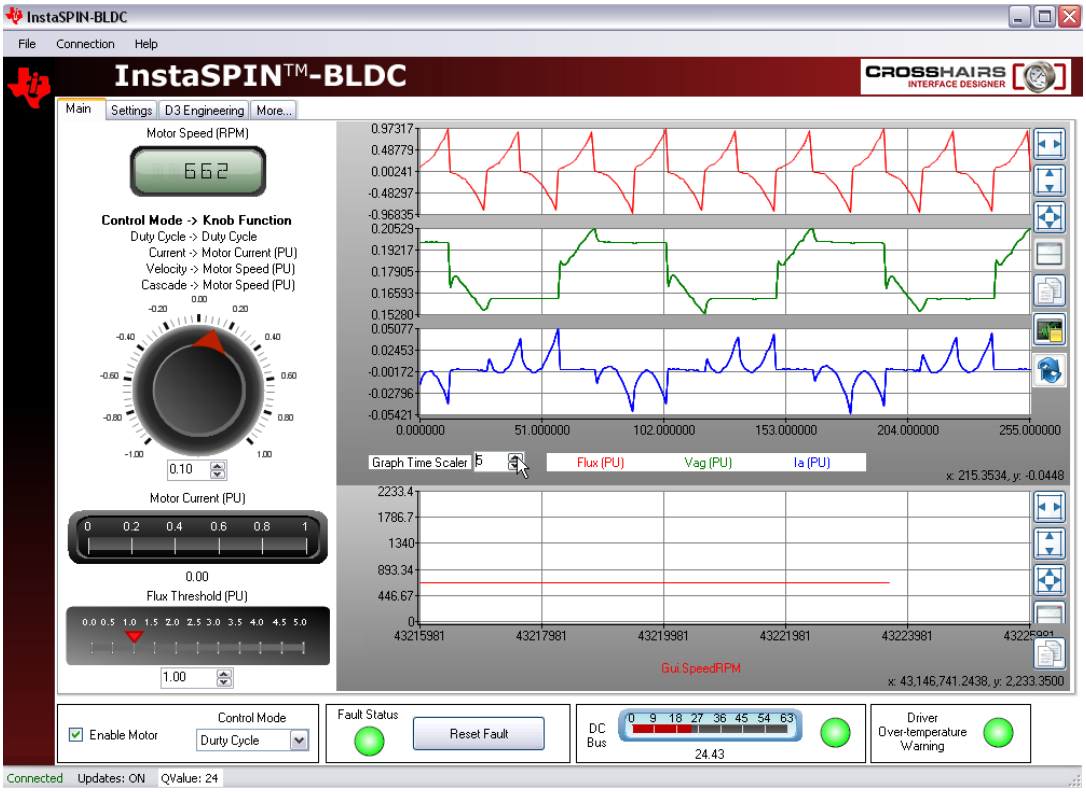
All motors tune differently, but InstaSPIN-BLDC and the GUI will allow easier tuning information to be gathered for the motor in question. The following steps are used to tune the motor that comes with the DRV8312 kit.

- Set duty cycle setting to 0.1
- Turn on “Start Continuous Read” on the upper Graphs

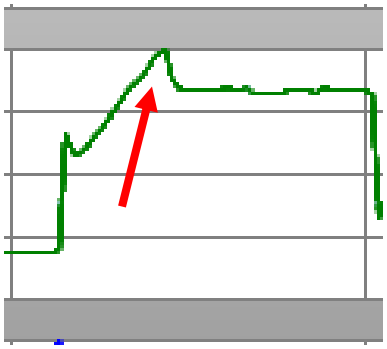


- Adjust Graph Time Scale to 5

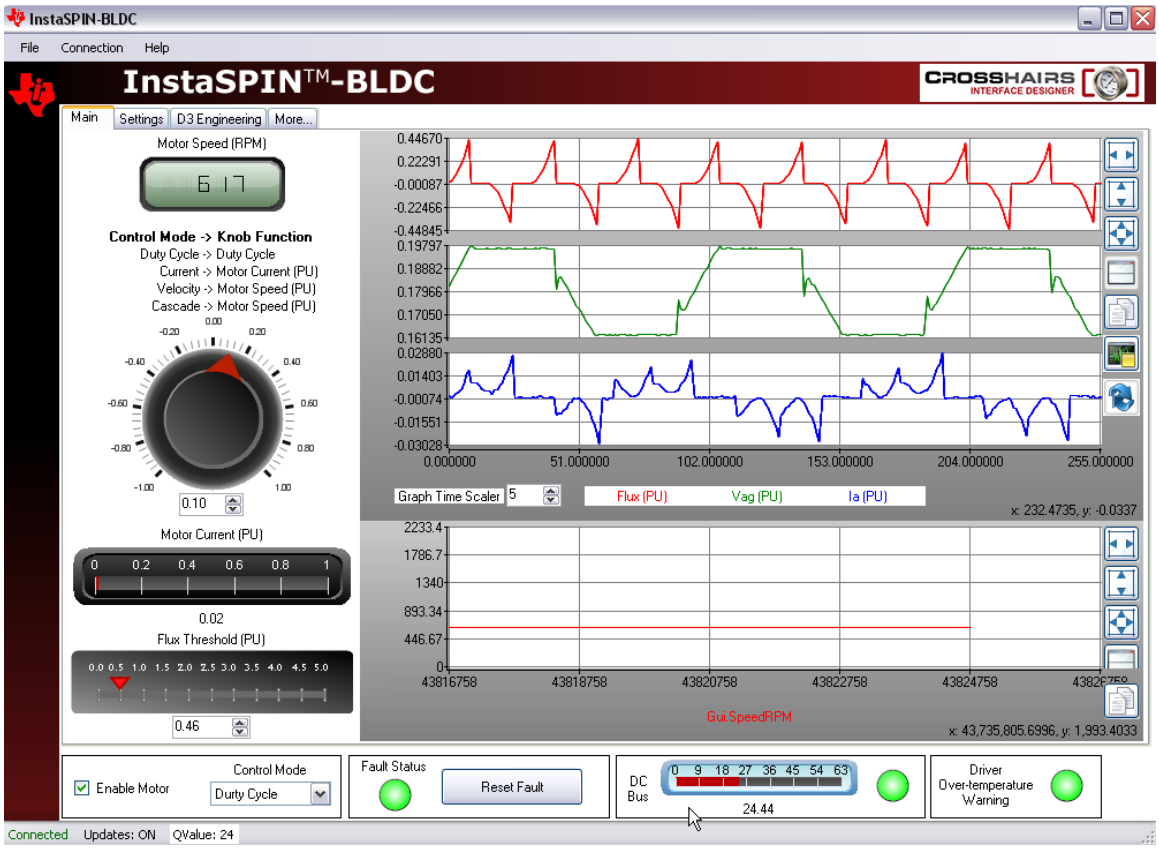
Top Graph: Displays the per-unit (PU) integrated motor flux.
Second Graph: Displays the per-unit (PU) Phase A BEMF waveform.
Third Graph: Displays the per-unit (PU) Phase A current waveform.



- Looking at the green voltage signal Vag, the commutation is happening too late as can be seen by the overshoot where the red arrow is pointing in the figure.



- Adjust the "Flux Threshold" to a smaller value (around 0.4) until the commutation happens right at the inflection point of the voltage waveform.



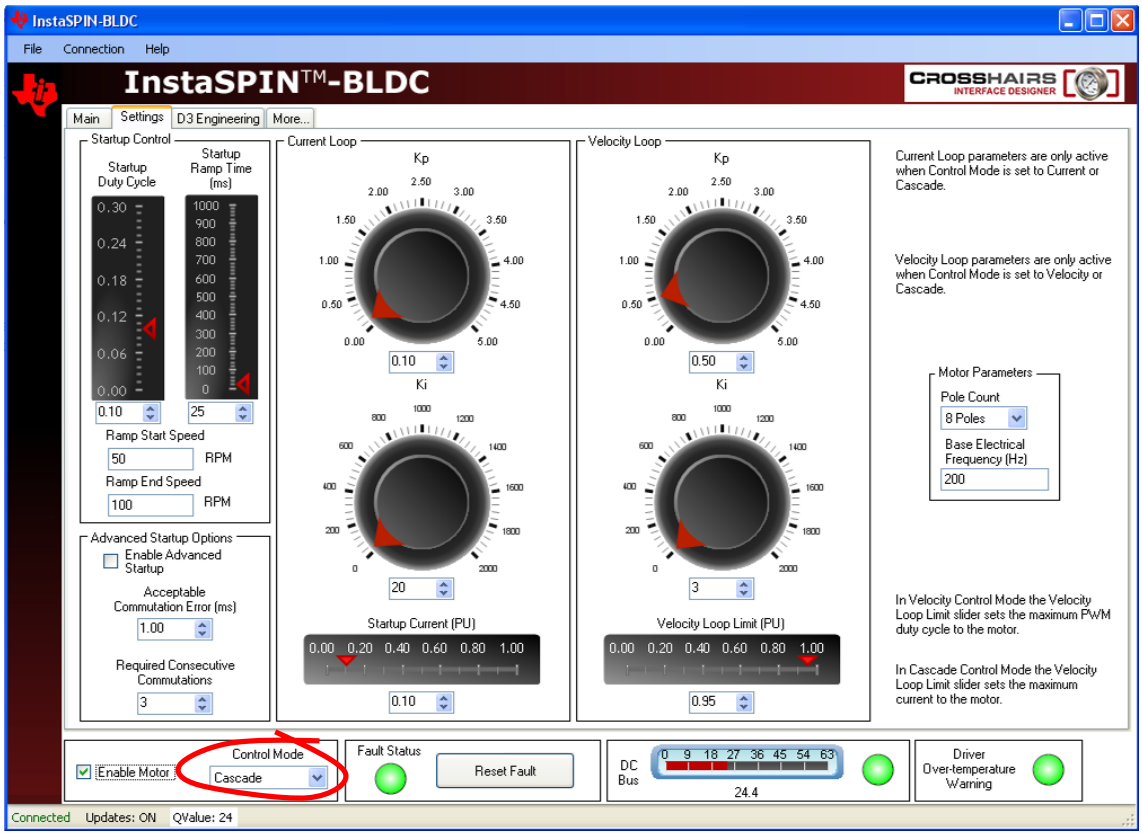
The motor is now tuned for InstaSPIN-BLDC!!!

The best way to get a feeling for tuning is to take different types of motors and try to get them to spin properly across the whole duty cycle range by adjusting the Flux Threshold. At lower speeds, most motors will run quieter at lower flux threshold values, but will deliver more torque at higher flux thresholds. At higher speeds, torque response also improves by lowering the flux threshold (can you guess why?) …It turns out that a BLDC motor works very much like a combustion engine. At higher speeds, you want to advance the spark timing because it takes a finite amount of time to ignite the fuel vapor in the cylinder. If the engine is running at high RPM, by the time the fuel has ignited, and the force of the explosion is felt on the piston, the piston has already moved past the optimal point where that force can do the most good. The same is true for a BLDC motor. It takes a finite amount of time to commutate a BLDC motor (get the current extinguished in one coil and built up in another. If the motor is running at high RPM, by the time the commutation takes place, the rotor has already moved past the optimal point where the magnetic force can do the most good. Advancing the commutation timing is just like advancing the spark timing in a combustion engine.

Extra Credit: Cascade Speed Control with InstaSPIN-BLDC

With the Control Mode still set to “Duty Cycle”, enable the motor and load the motor down with your fingers. What happens to the speed of the motor when you do this? Since there is no control loop regulating the speed, the motor slows down more and more as you increase the load on the shaft. To obtain better speed regulation, you must enable either “Velocity” mode, or “Cascade” mode.

Disable the motor, and on the “Settings” tab, change the Control Mode to “Cascade”, but leave everything else set to its default value. The panel should then look like the following diagram.



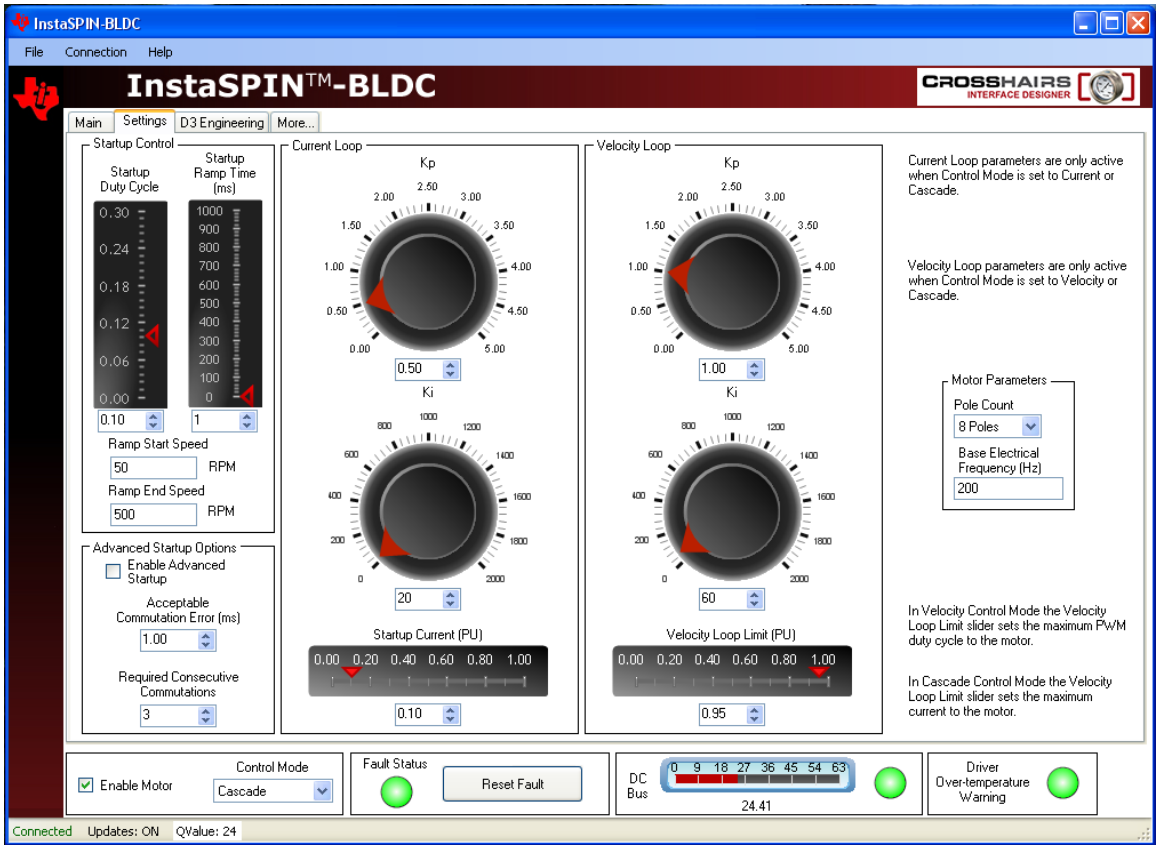
Switch to the “Main” tab again and verify that the Control knob is set to its default value of 0.3 PU. Enable the motor and verify that it starts up smoothly and ramps to a velocity value of about 900 RPM. This can be seen by observing the speed in the Motor Speed RPM window:

Now load the motor shaft with your fingers and observe the speed regulation performance of the system. What are your initial impressions? Notice that for constant steady torque loading, the motor will eventually return to its commanded value of about 900 RPM. This is because of the integrators which have been enabled in the control loop. An integrator in a negative feedback control system cannot tolerate any steady state error, and will drive it to zero (i.e., the commanded value and the controlled value will be equal). However, the response of the system is rather spongy, isn’t it? In other words, this is not a “stiff” control system. In order to stiffen up the response, we must increase the gains of the system.

Disable the motor again and return to the “Settings” tab. Insure that all of the values on the settings page are set to the values listed below. Values in red are non-default values.

Startup Duty Cycle:	0.10
Startup Ramp Time:	1 mS
Ramp Start Speed:	50 RPM
Ramp End Speed:	500 RPM
Kp (current):	0.50
Ki (current):	20
Startup Current (PU):	0.10
Kp (velocity):	1.00
Ki (velocity):	60
Velocity Loop Limit (PU):	0.95
Pole Count:	8 poles
Base Electrical Freq. (Hz):	200
Control Mode:	Cascade

If set correctly, the Settings page should look like the diagram below:



Assuming you haven’t changed the Control knob on the “Main” tab, it should still be set to a PU value of 0.3. If not, change it back to that value now. Once again, enable the motor, and verify that it starts up smoothly and achieves a steady state velocity of about 900 RPM.

Again, load the motor shaft with your fingers and observe the speed regulation performance of the system now. What are your initial impressions? Notice that the system is much stiffer in terms of maintaining the commanded speed, irrespective of motor loading.

So if the higher gain settings result in much better tracking performance, then why didn't we specify them as the default values to begin with? To answer this question, let's slowly reduce the speed and see what happens. Using the up-down arrows on the right of the control knob value field, slowly decrement the commanded speed one number at a time. When the value gets close to 19 or so, the system should go unstable.

QUESTION: So why would the system be stable at higher speeds, but unstable at lower speeds? (This is a tricky one). Think about it for a moment…

Hint: It has something to do with how the velocity feedback signal is synthesized.

ANSWER: In order to measure velocity in a sensorless system, the software must measure the duration of each commutation interval. When the motor is spinning at high speed, there are many commutation intervals per second, which means that the velocity feedback value is updated many times per second. However, when the motor is spinning at low speed, the commutation rate is low, which means the update rate for the velocity value is also low. Therefore, the velocity feedback value becomes "stale" when it is not updated for a long period of time. To be more precise, there is a phase lag between the true velocity and the measured velocity when the velocity value is updated so infrequently. This phase lag eats into the phase margin of your system, and when it is all gone, the system goes unstable.

In most sensorless BLDC applications (e.g., fans, blowers, and compressors), this is not a problem since it doesn't make sense to run these loads at ultra-low speeds anyway. But for other applications (e.g., washer drum agitation), low speed operation is a requirement, which suggests a different control scheme (such as InstaSPIN-FOC) is required.

Shutting Down

Un-check "Enable Motor", close the GUI, and remove power from the DRV8312 board.

Overview of the GUI Controls

Let's review each of the panels in the GUI interface and explore the functionality of each control.

Main Tab Overview

Enable Motor Check Box: The Enable Motor check box is used to start or stop the motor

Control Mode Drop Down Box

Allows the selection of four different control modes

1. Duty Cycle
    a. The motor is commutated using the sensorless algorithm but is driven in an open-loop duty cycle mode.
2. Current
    a. The motor is commutated using the sensorless algorithm while the current (torque) is regulated using a PI controller.
    b. (Note: An unloaded motor will rapidly accelerate to a very high speed in this mode.)
3. Velocity
    a. The motor is commutated using the sensorless algorithm while the motor speed is regulated using a PI controller. The output of the speed controller is a PWM duty cycle.
4. Cascade
    a. The motor is commutated using the sensorless algorithm while the motor speed is regulated using a PI controller. The output of the speed controller is a motor current command which is regulated by a lower level current PI controller.

The Flux Threshold slider is used to adjust the motor's commutation point

The Setpoint Knob takes on a separate function for each control mode:

- Duty Cycle: The knob adjusts the PWM duty cycle to the motor.
- Current: The knob adjusts the per-unit (PU) commanded current through the motor.
- Velocity: The knob adjusts the per-unit (PU) motor commanded speed.
- Cascade: The knob adjusts the per-unit (PU) motor commanded speed.

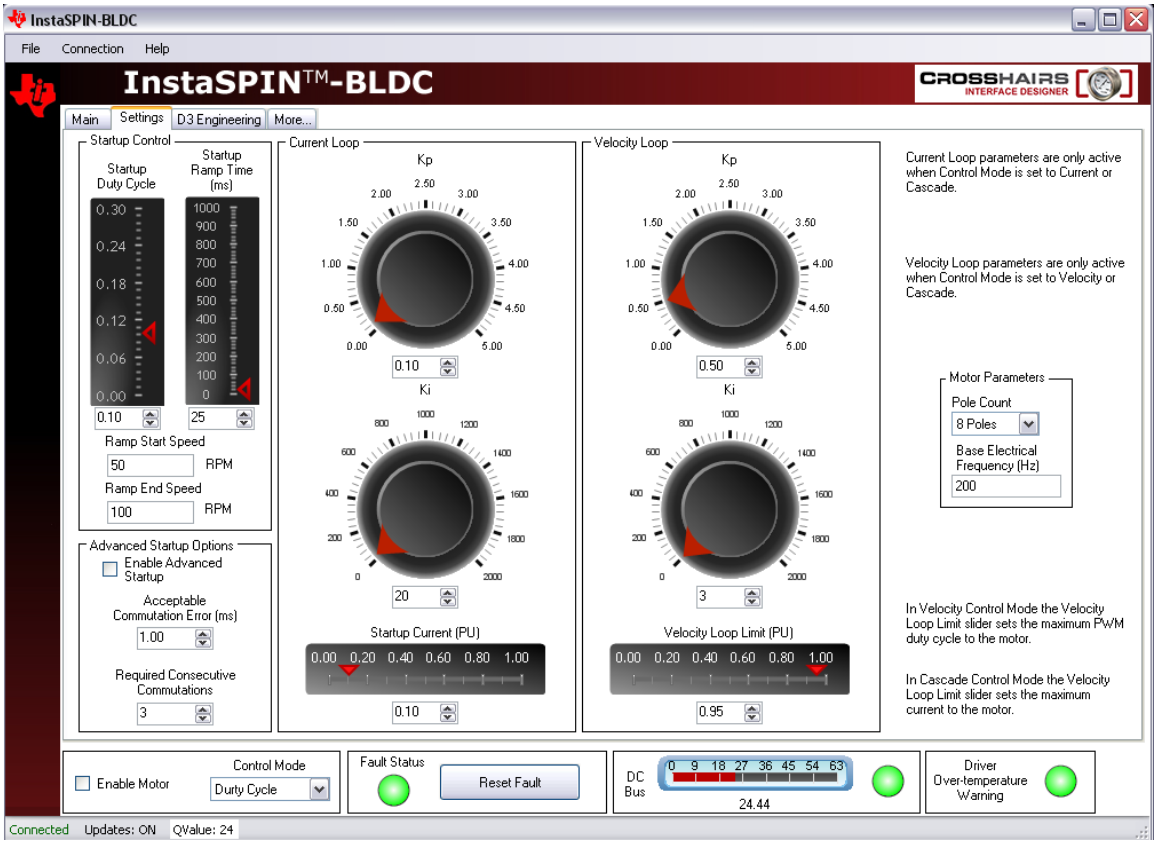
Graphs

1. Top Graph: Displays the per-unit (PU) integrated motor flux.
2. Second Graph: Displays the per-unit (PU) Phase A BEMF waveform.
3. Third Graph: Displays the per-unit (PU) Phase A current waveform.
4. Fourth Graph: Displays the motor speed in RPM.

The time scale of the top three graphs can be adjusted by incrementing/decrementing the Graph Time Scalar, the Motor Speed graph is not affected.

Settings Tab Overview

Contains parameters affecting motor startup and control loop tuning



Startup Control

These parameters control how the motor initially ramps up under forced commutation.
It is necessary to get the motor spinning and generating some BEMF in order for the sensorless algorithm to latch on and take over commutation.

Startup Duty Cycle

Sets the constant PWM duty cycle given to the motor during the forced commutation ramp up phase.

Startup Ramp Time

Sets the time taken to complete the forced commutation ramp up phase.

Ramp Start Speed

Sets the initial speed for the forced commutation ramp up phase.

Ramp End Speed

Sets the final speed for the forced commutation ramp up phase.

Advanced Startup Options

The controller will match InstaSPIN-BLDC detected commutation events to forced commutation events. Once the set number of commutation matches have been met commutation switches to InstaSPIN-BLDC

Enable Advanced Startup

Enable/Disable the Advanced Startup feature.

Acceptable Commutation Error

Allowable error, in ms, between forced commutation events and InstaSPINTM-BLDC detected commutation events. Commutation events within the error window will be considered "good."

Required Consecutive Commutations

Number of required consecutive "good" commutation matches before control switches from forced commutation to InstaSPIN-BLDC.

Motor Parameters

Pole Count (P)

Input the number of rotor magnet poles for the motor under test.

Base Electrical Frequency (N)

Input the maximum frequency (in cycles per second) of the waveforms to be applied to the motor.

The equation which governs motor speed based on the above parameters is shown below:

$$\frac{N\ cycles}{second} * \frac{60\ seconds}{minute} * \frac{2\ revolutions}{P\ cycles} * (PU) = \frac{N*60*2*(PU)}{P}\ \frac{revolutions}{minute}$$

where PU is the Per Unit knob value (0 to 1) for the knob located on the "Main" tab.

The default settings of 200 cycles/sec and 8 rotor poles will result in a motor speed of 3000 RPM when the knob is set to its maximum value of 1.

Current Loop

Contains parameters associated with the current control loop. These parameters are only active when Control Mode is set to Current or Cascade. Those are the only modes which make use of the current loop.

Kp

Sets the proportional gain for the current controller.

Ki

Sets the integral gain for the current controller.

Startup Current

When current control is active the motor is driven with constant current rather than a constant duty cycle during the forced commutation ramp up phase. This slider sets the current for this phase.

Velocity Loop

Contains parameters associated with the velocity control loop. These parameters are only active when Control Mode is set to Velocity or Cascade. Those are the only modes which make use of the velocity loop. **Note**: The velocity controller my need to be retuned when switching between Velocity and Cascade Control Modes.

Kp

Sets the proportional gain for the velocity controller.

Ki

Sets the integral gain for the velocity controller.

Velocity Loop Limit

i. In Velocity Control Mode the Velocity Loop Limit Slider sets the maximum PWM duty cycle to the motor

ii. In Cascade Control Mode the Velocity Loop Limit Slider sets the maximum current to the motor

The correct answer to the motor wire connection question is (C)