InstaSPIN-BLDC Lab using Stellaris LM3S ControlCARD

Introduction:

For this lab we are using the DK-LM3S-DRV8312 Digital Motor Control (DMC) development kit to exercise motor control using 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.
- Install drivers onto the host computer.
- Flash the InstaSPIN_GUI file onto the controlCARD using the LM Flash Programmer.
- Launch the GUI
- Start spinning motors!

DRV8312 Board Set-Up
Running the GUI
Overview of GUI Controls

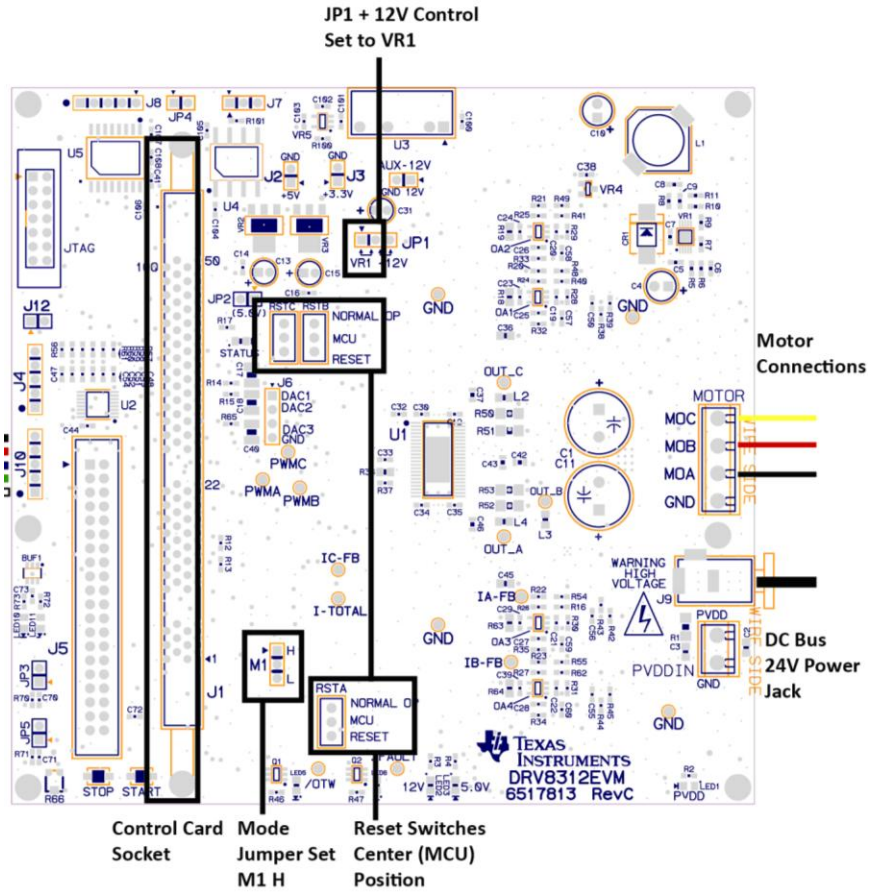
SET-UP
    Stellaris LM3S controlCARD Set-Up
    Stellaris USB-JTAG Driver
    Programming the controlCARD

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.

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



Powering Up the Board:

- Plug in USB cable to the controlCARD and LED4 and 5 on the controlCARD are green (indicating power).
- Connect the included 24V DC power supply to the power entry jack on the DRV board and LED1, 2, and 3 on the DRV8312EVM baseboard are green (indicating power)

GUI setup:

The Quickstart GUI (*Sandstorm_InstaSpin-BLDC-GUI_V102.exe*) is included in the DK-LM3S-DRV8312 self-installer (DK-LM3S-DRV8312-1280.exe) file included on the Development Kit CD.

- The installer places the source and executable files in the following location:

C:\StellarisWare\AppNotes\sw01289

- The Stellaris InstaSPIN-BLDC GUI program can be found here:

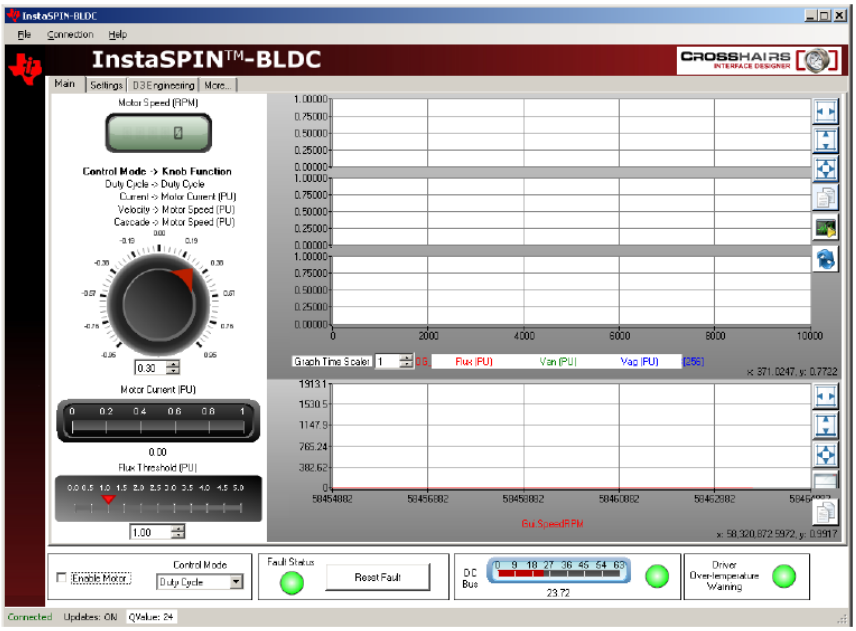
C:\StellarisWare\AppNotes\sw01289\GUI\Sandstorm_InstaSpin-BLDC-GUI_V102.exe

Note: The GUI requires Microsoft .NET framework 3.5 SP1 or higher to run. You must install this software prior to running this program.

The development kit ships with a Stellaris controlCARD which is pre-programmed with the InstaSPIN-BLDC binary.

Running the GUI:

Navigate to the ***Sandstorm_InstaSpin-BLDC-GUI_V102.exe*** file and double-click it to launch the GUI pop-up window (as shown in figure below).

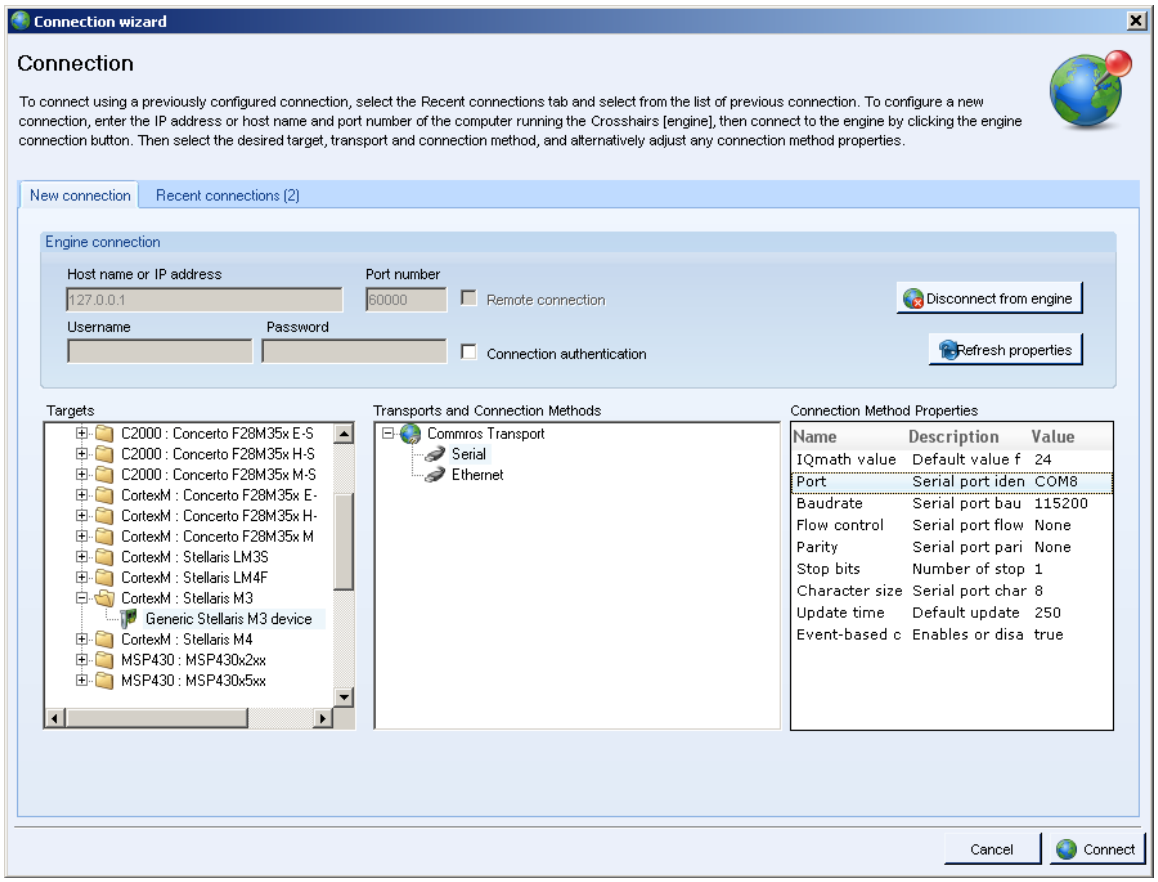


The GUI auto-detects and connects to the EK-LM3S-DRV8312. If auto-connect fails, you can set up the connection manually.

- Access the Connection Wizard through the Connection menu. Click “Connect to engine” to view a list of available targets. Set up the Connection Wizard Dialog to match Figure 4.

- Select *ARM Cortex-M3* from the target list and *Serial* for the connection method. Determine the correct COM port number for your system by using the Control Panel:
  Control Panel > System > Hardware tab > Device Manager > Ports (COM & LPT)

  - Look for the port which is described as the Stellaris Virtual Serial Port and make a note of the COM port number. Enter this value in the Port field of the Connection Method Properties box. Click Connect once when you are finished.



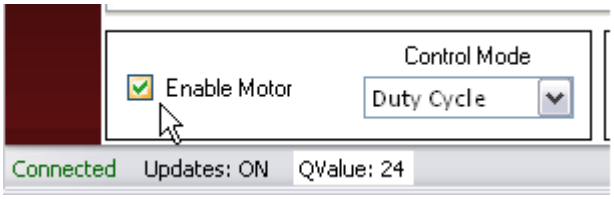
If no error message is reported, then the GUI has successfully established a connection with the target. The DC bus voltage and other indicators are now receiving real-time updates.

If an incorrect image is flashed on the controlCARD, the connection fails

If this happens, you should flash the controlCARD again with the correct image. Instructions for this step can be found here.

Spinning the motor:

- From the Main Tab Click “Enable Motor”

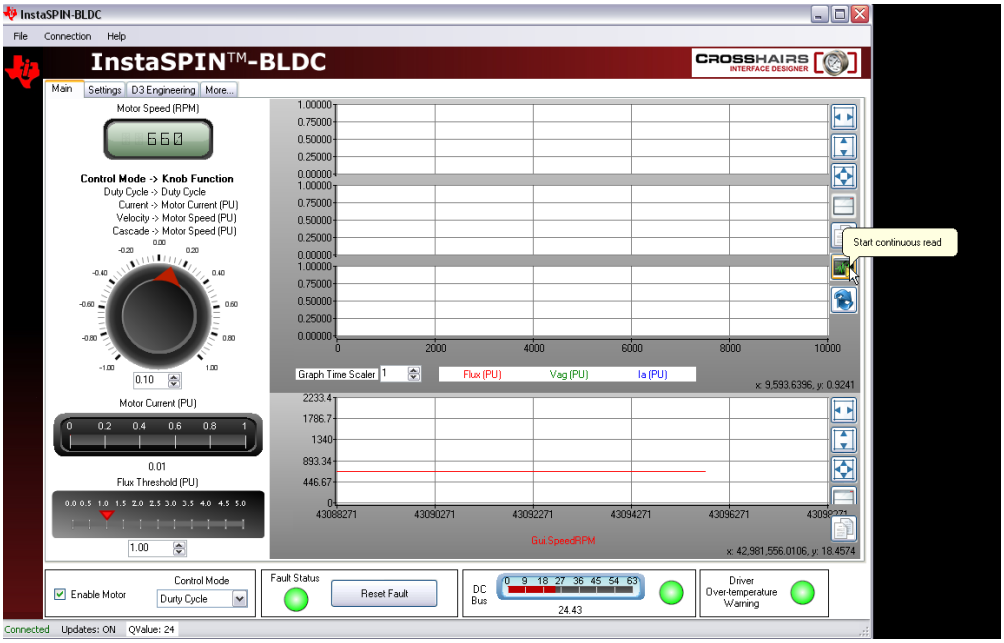


- The motor will start spinning under default settings
  - Duty Cycle = 0.3 PU = 2129 RPM
- **Congratulations! You just did InstaSPIN!!!!**

Tuning the Motor Commutation:

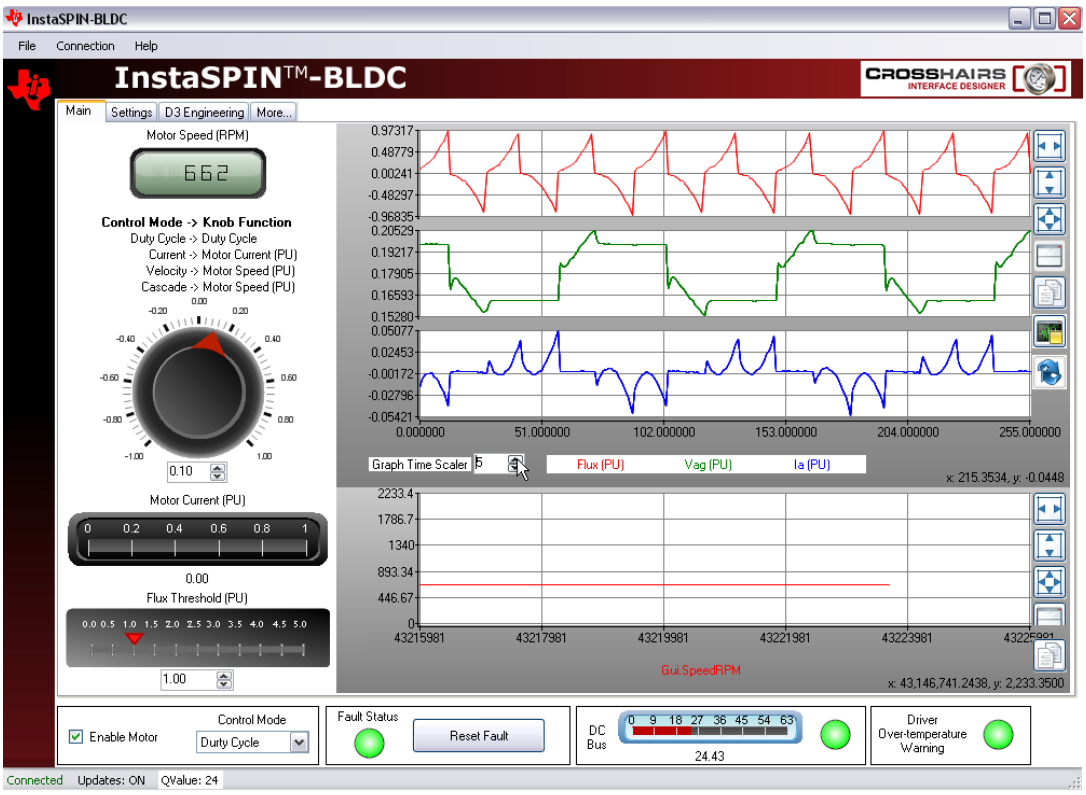
All motors tune differently, but InstaSPIN-BLDC and the GUI will allow easier tuning information to be gathered for the motor in question. The following steps are used to tune the motor that comes with the DRV8312 kit.

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

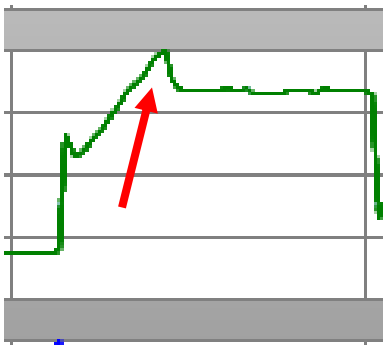


- Adjust Graph Time Scale to 5

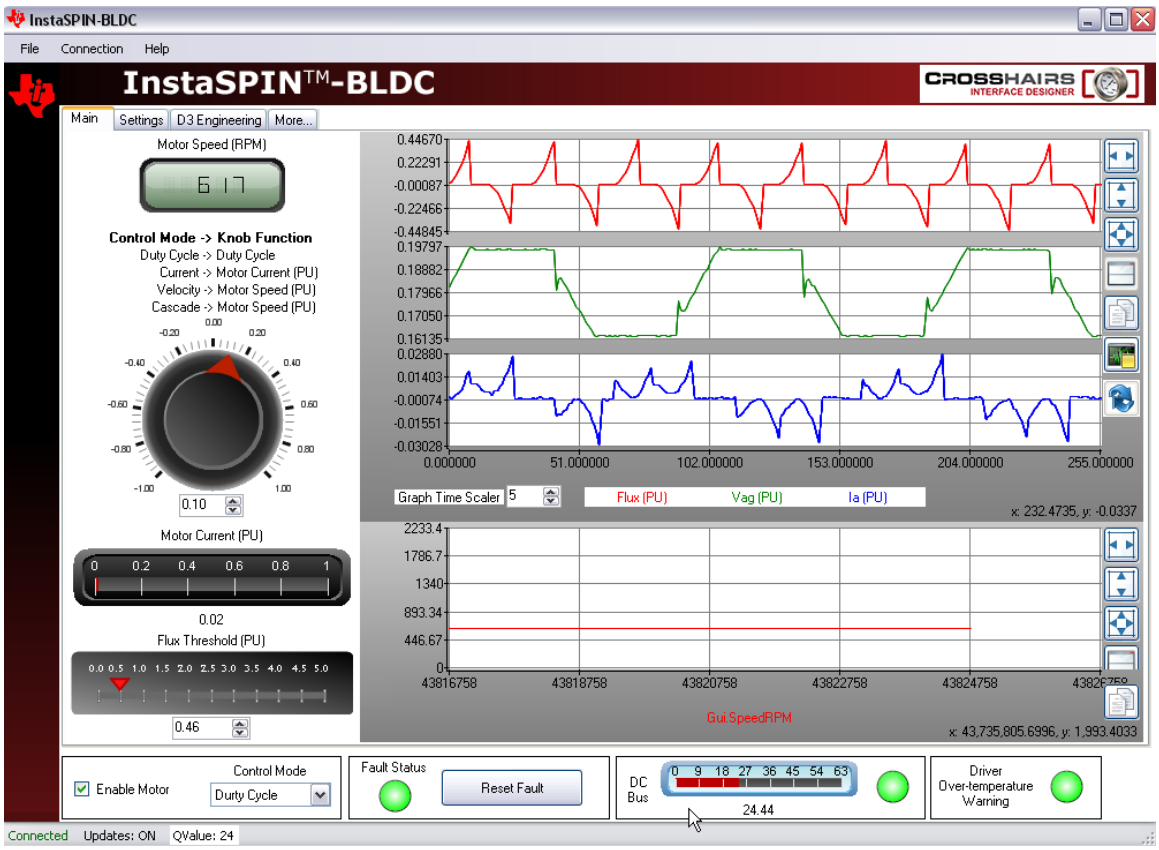
  Top Graph: Displays the per-unit (PU) integrated motor flux.
  Second Graph: Displays the per-unit (PU) Phase A BEMF waveform.
  Third Graph: Displays the per-unit (PU) Phase A current waveform.



- 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.



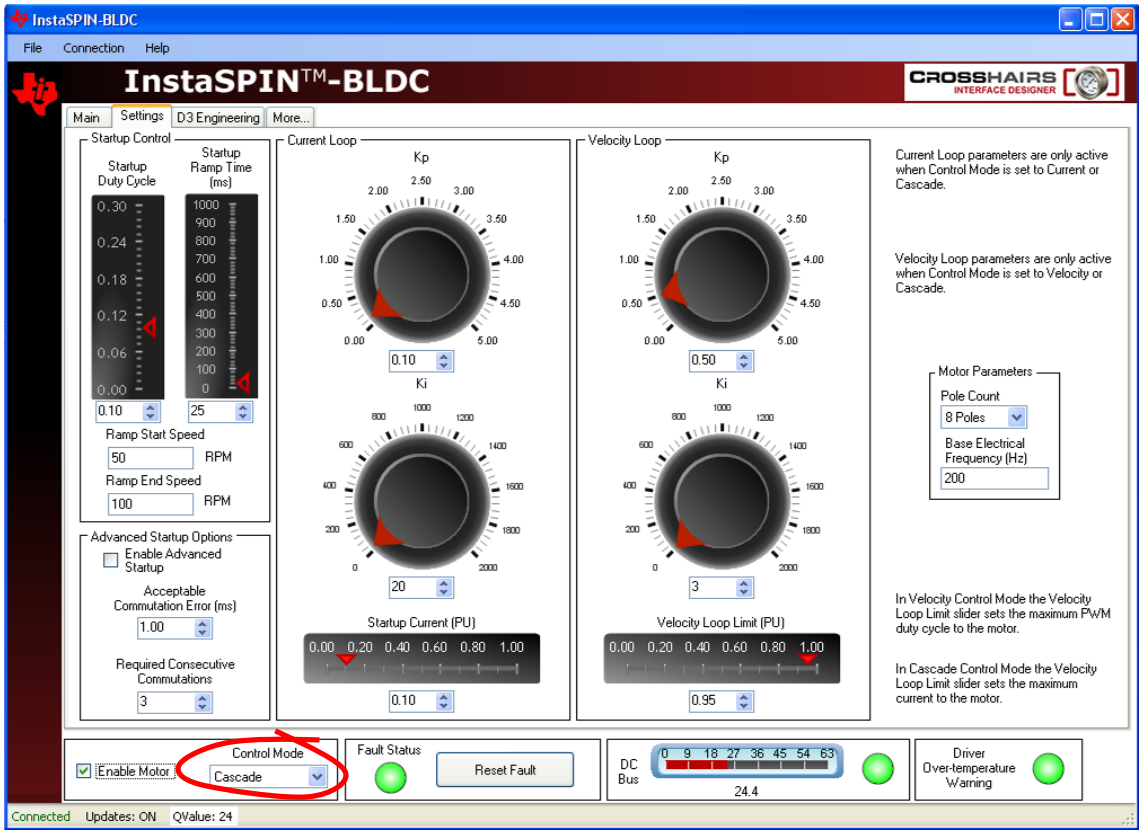
The motor is now tuned for InstaSPIN-BLDC!!!

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.



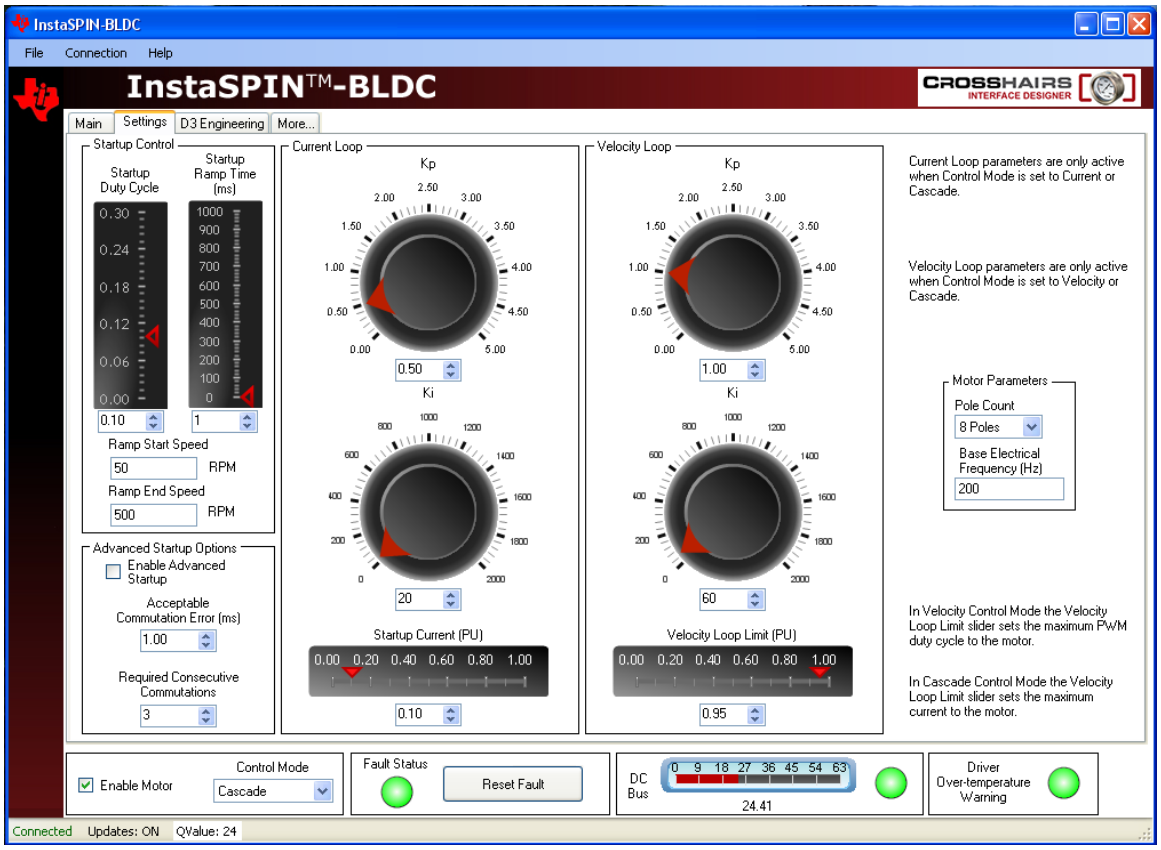
Switch to the “Main” tab again and verify that the Control knob is set to its default value of 0.3 PU. Enable the motor and verify that it starts up smoothly and ramps to a velocity value of about 900 RPM. This can be seen by observing the speed in the Motor Speed RPM window:

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.

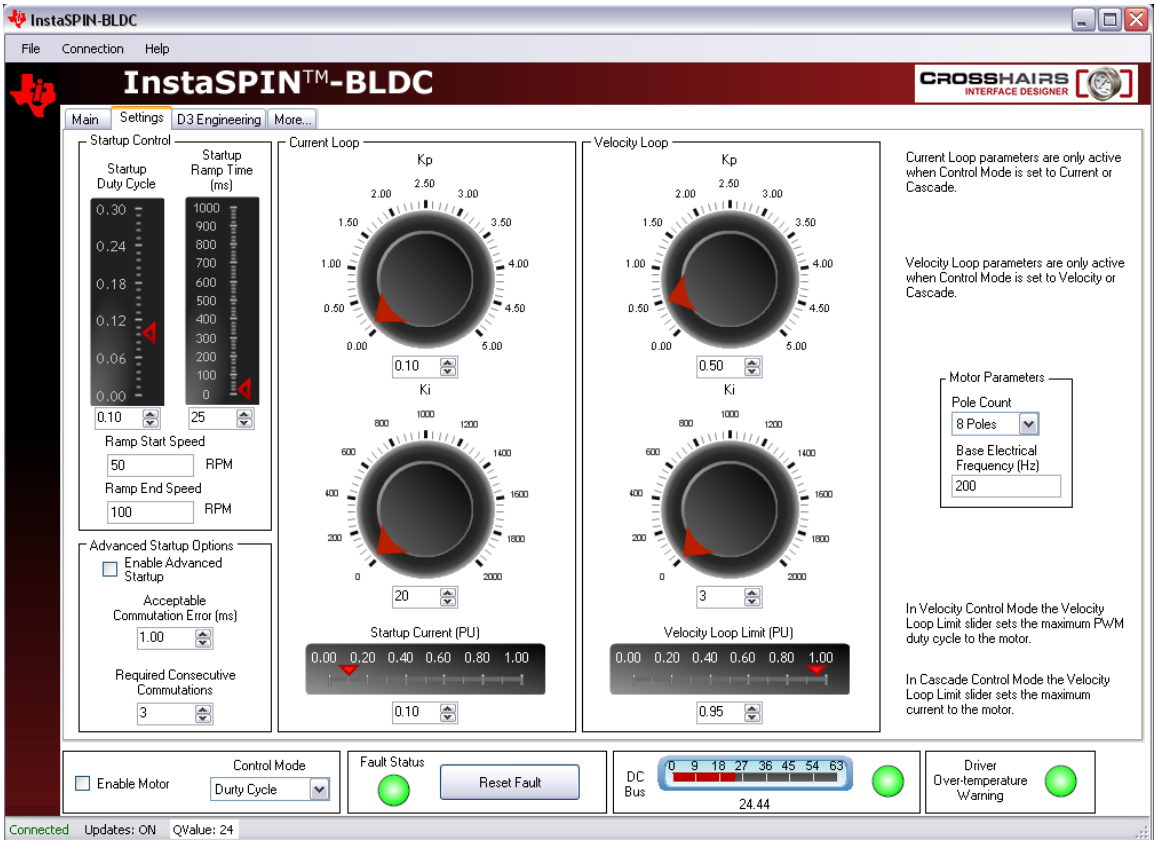
Graphs

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

END OF LAB

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

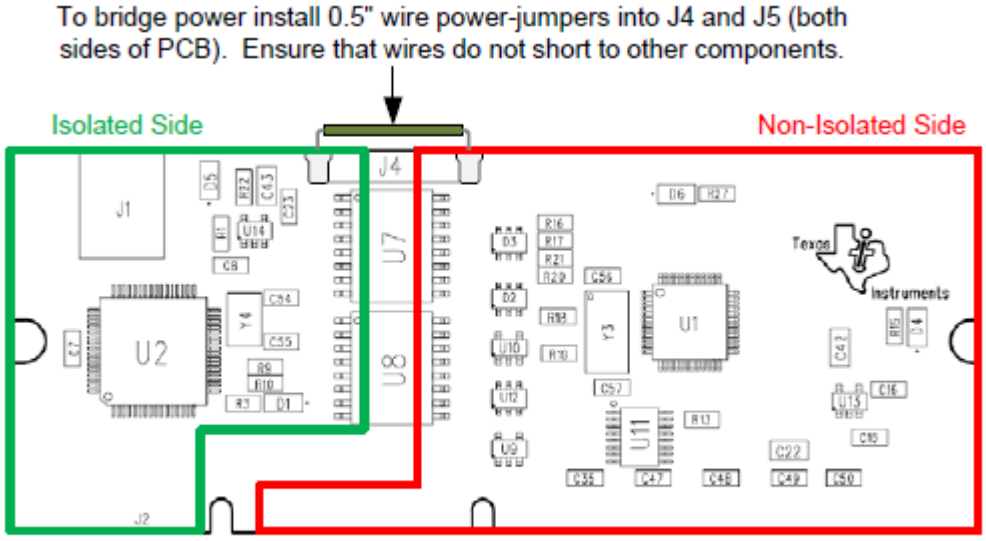
Set-Up

Stellaris ControlCARD setup:

Note: Do not apply power to board before you have verified these settings!

The kit ships with the controlCARD inserted and the jumper and switch settings pre-selected for connecting with the GUI. However, you must ensure that the following settings are valid on the board:

- Check to make sure that nothing is connected to the board, and that no power is being supplied to the board.
- Insert the controlCARD into the J1 controlCARD connector if not already populated, making sure any jumpers to bridge the isolation barrier for standalone operation have been removed as shown (figure shows only the top-side of the PCB):



WARNING: Do not install controlCARD in a base-board when wire power-jumpers are installed!

- Connect a USB cable from J1 on the controlCARD to a PC and proceed with driver installation.

Install drivers on the host computer:

In order to debug and download the custom application in the microcontroller's Flash memory and use Virtual COM Port connectivity, you must first install the following drivers on the host computer:

- Stellaris Virtual Serial Port
- Stellaris ICDI JTAG/SWD
- Stellaris ICDI DFU

Note: This document describes the procedure to install drivers on the Windows XP operating system. There might be some variation for installing the drivers on other Windows operating systems, although the procedure should be similar.

To see which drivers are installed on the host computer, check the hardware properties using the Windows Device Manager. Do the following:

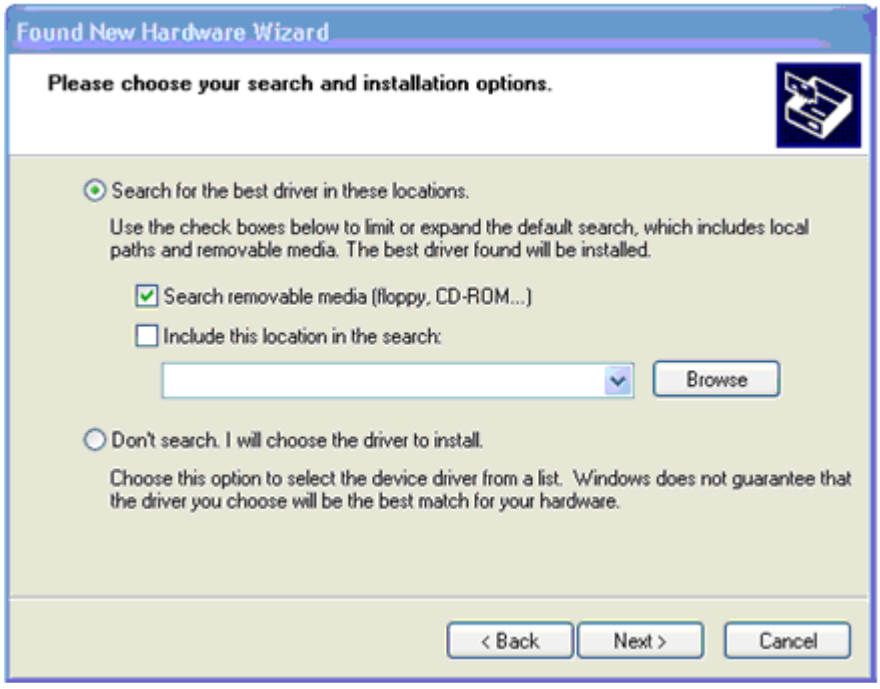
- Right-click My Computer from the Windows Start button and select Properties from the drop-down menu.
- In the System Properties window, click the Hardware tab.
- Click the Device Manager Button. The Device Manager window displays a list of hardware devices installed on your computer and allows you to set the properties for each device.

When the controlCARD is connected to the computer for the first time, the computer detects the on-board ICDI interface. Drivers that are not yet installed display a yellow exclamation mark in the Device Manager window.

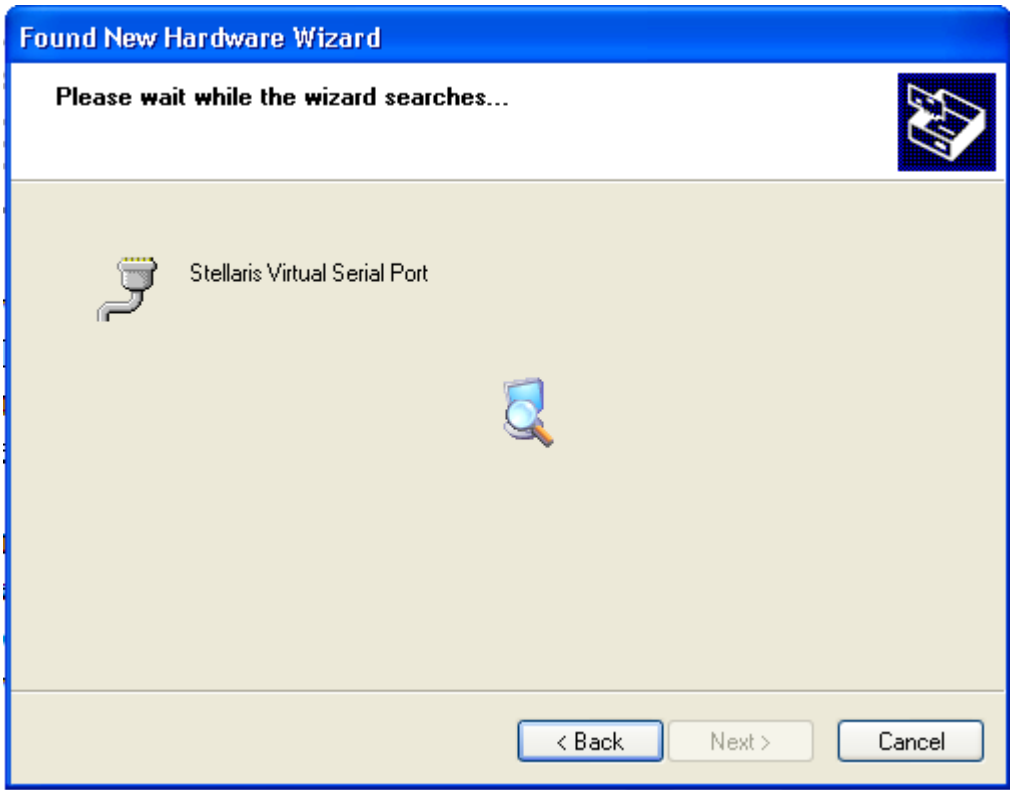
When you plug in the EVB for the first time, Windows starts the Found New Hardware Wizard and asks if you want to install the drivers for the Stellaris Virtual Serial Port.

- Select "Install from a list or specific location (Advanced)" and then click Next.

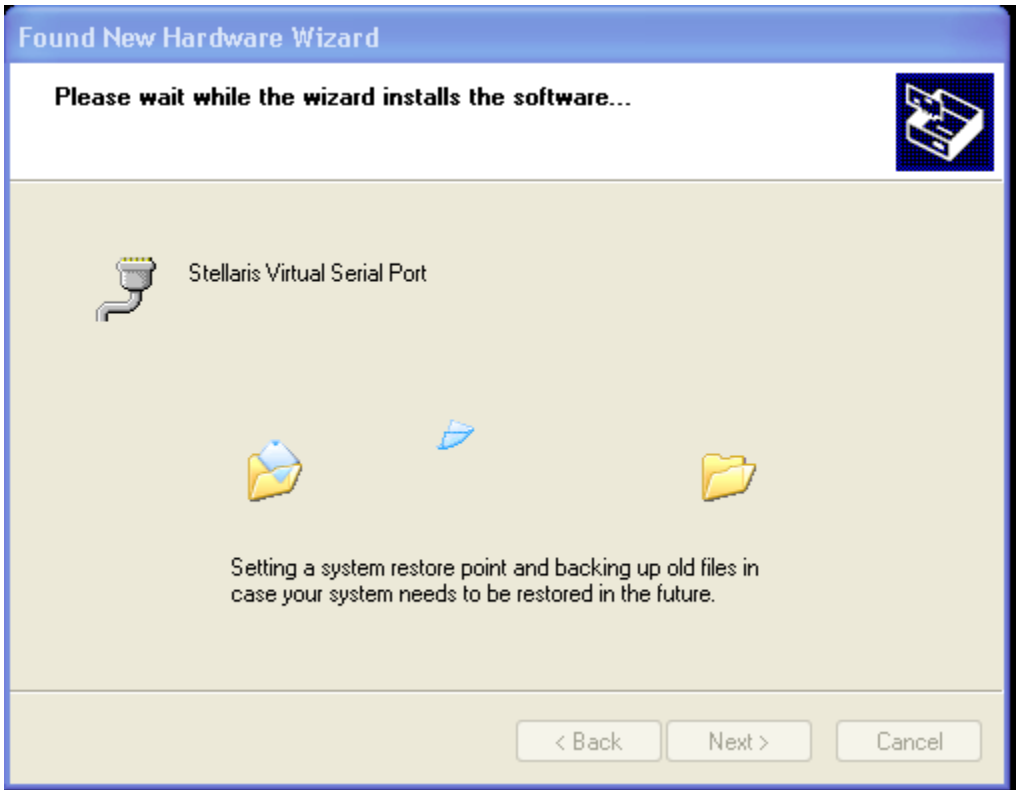
- Make sure the “Documentation and Software” CD that came with the development kit is in your CD-ROM drive. Select “Search for the best driver in these locations,” and check the “Search removable media (floppy, CD-ROM…)” option. Click Next.



- A warning may pop up during the installation process regarding the driver not being signed, click Continue Anyway to proceed. The wizard displays a “Please wait while the wizard searches…” status window. No user action is required.

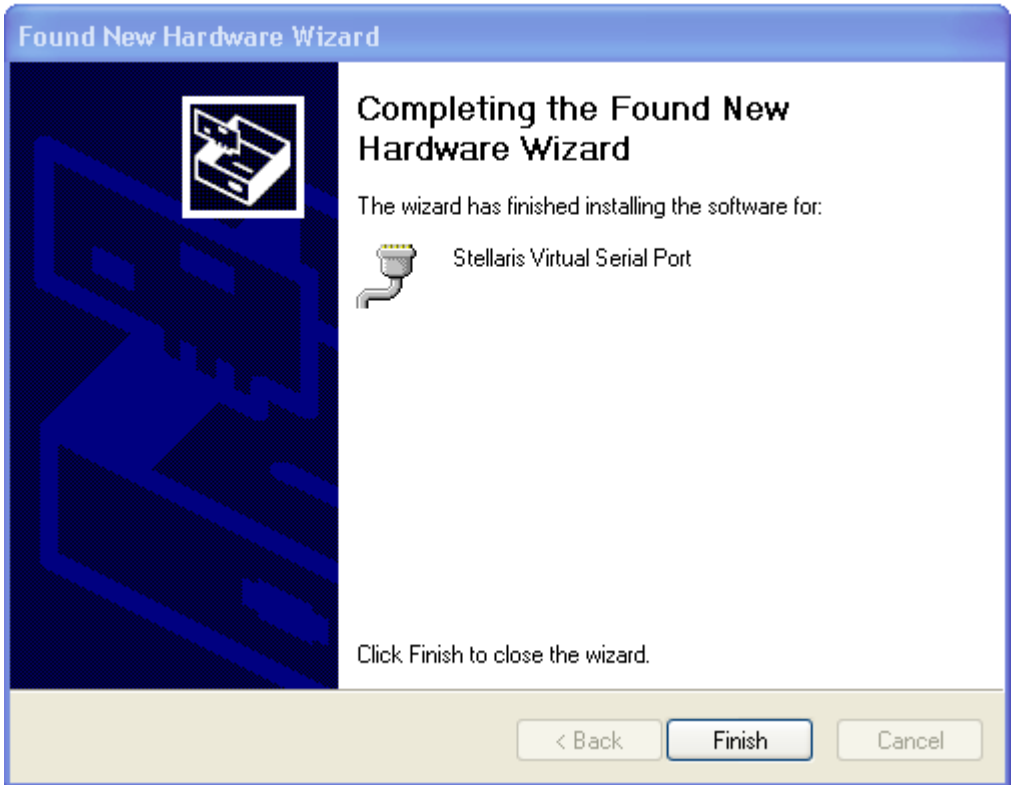


The wizard then displays a “Please wait while the wizard installs the software…” status window as the software is installed.



- During driver installation, the windows installer may ask for the path to.dll files. Browse to the path on the installation CD:\Software\ICDI\i386

- After the installation of the Stellaris Virtual Serial Port drivers, click Finish to close the dialog box.



You have just installed the drivers for the Stellaris Virtual Serial Port.

The Found New Hardware Wizard appears again for the Stellaris ICDI JTAG/SWD Interface and then one more time for the Stellaris ICDI DFU Device drivers. Follow the same instructions to install the drivers for these two devices.

You can confirm the three device driver installations by launching the Windows Device Manager and right-clicking to select “Scan for Hardware Changes.” This updates the Device Manager Properties list. The Stellaris Virtual Serial Port, Stellaris ICDI JTAG/SWD Interface, and Stellaris ICDI DFU Device now appear in the list. This indicates that the drivers have been successfully installed.

These drivers provide the debugger with access to the JTAG/SWD interface, and the host PC access to the Virtual COM Port. With these drivers installed, Windows automatically detects any new Stellaris boards (with a Stellaris-based ICDI) that you connect to your computer, and installs the required drivers for you

RESTART YOU R COMPUTER

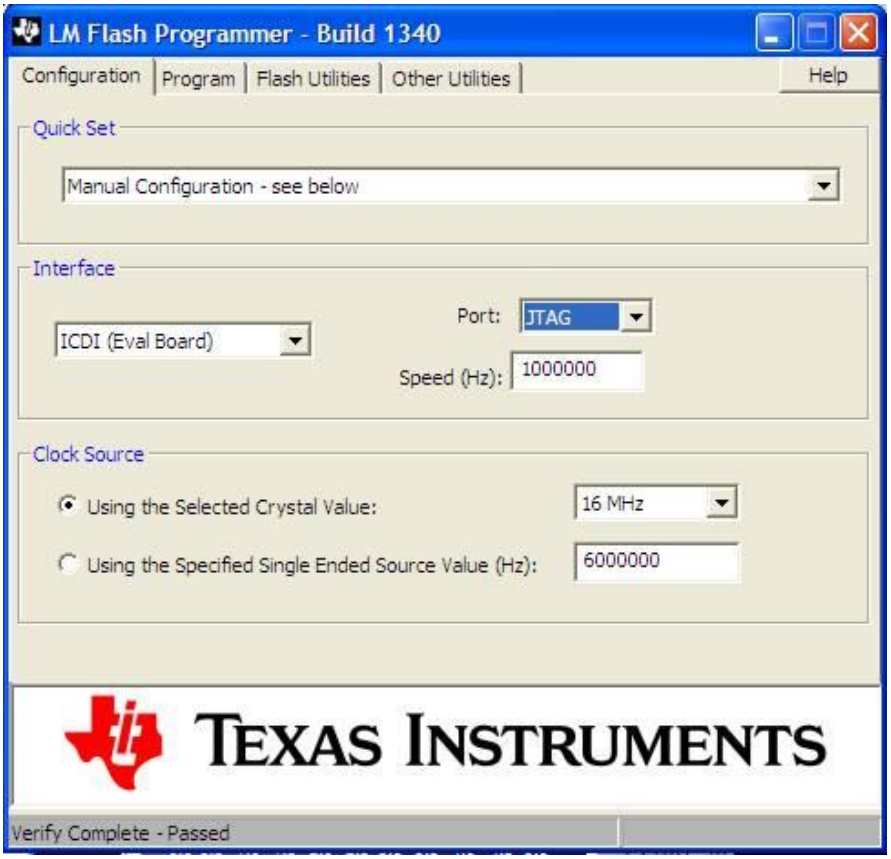
How to Flash the ControlCARD:

The DK-LM3S-DRV8312 kit includes a controlCARD that is pre-programmed with the InstaSPIN-BLDC binary. If the controlCARD was purchased separately, or if a software update is available, or for any reason if the controlCARD does not have the program preloaded then the binary on the controlCARD should be programmed using the LM Flash Programmer utility.

To program example applications into the MDL-LM3S818CNCD module using the Stellaris ICDI:

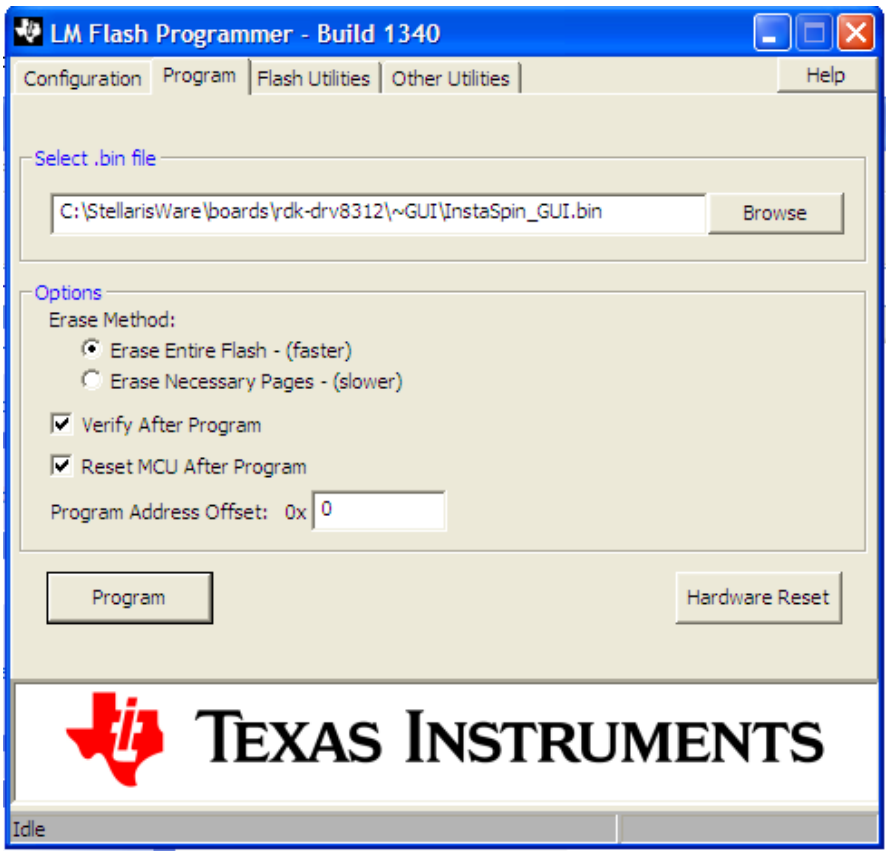
1. Install LM Flash Programmer on a Windows PC. LM Flash Programmer is provided on Stellaris development and evaluation kit CDs, or can be downloaded from www.ti.com/stellaris.
2. Connect the USB cable A-plug to an available port on the PC and the mini B-plug to the board.
3. Verify that both controlCARD power LEDs are lit. LED D4 indicates the status of the non-isolated microcontroller power and LED D5 indicates the status of the isolated USB power.
4. Run LM Flash Programmer.

5. Under **Configuration** tab (Figure 1)
   - Set the Quick set to **Manual Configuration**
   - Set the Interface to **ICDI (Eval Board)**, Port → **JTAG**, Speed (Hz) → **1000000**
   - Set the **Clock Source to Using the Selected Crystal Value → 16 Mhz**



Under **Program** tab (Figure 2 – next page)

- Select InstaSpin_GUI.bin in the **~GUI** folder under ***C:\StellarisWare\AppNotes\sw01289\InstaSpin_GUI\ccs\Debug***
- Under **Options** set erase method to **Erase Entire Flash.**
- Check **Verify After Program** and **Reset MCU After Program.**
- Set offset to 0.



Power Cycle the board before continuing.