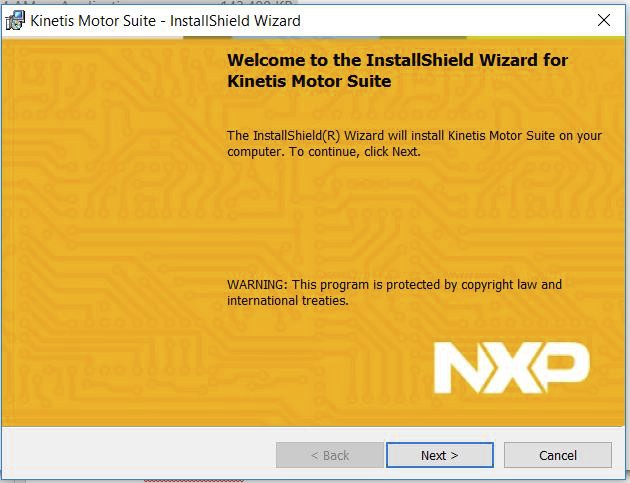
KMS and KV31

Part I: Setup the Environment

1. Software

1.1. Kinetis Motor Suite PC software

• Kinetis Motor Suite PC software refers to the PC executable that runs the KMS user interface, communicates with the selected microcontroller (MCU), and adapts   
embedded firmware reference projects for your system.



• Locate the “Kinetis-Motor-Suite-

1.1.0-Installer” at your course

website, download it and follow

the steps to install it.



1.2. Kinetis Software Development Kit (SDK)

• KMS builds on the Kinetis software development kit (KSDK) to configure hardware and peripherals for the supported microcontrollers.

• If you did not yet install “Kinetis SDK 1.3.0” on your computer then locate it on your course website, download it and install it.



1.3. Development tools

• In order to access the open-source embedded firmware provided as part of KMS and to compile KMS firmware, your PC must an integrated development environment (IDE) such as Kinetis Design Studio (KDS).

• It is a free IDE based on the Eclipse software framework and distributed by NXP.

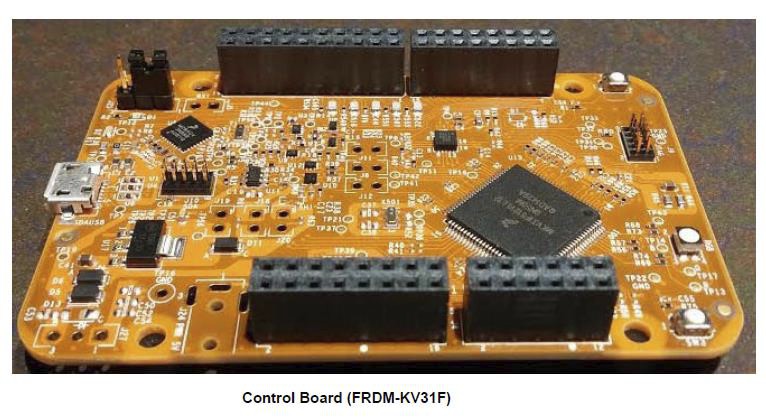
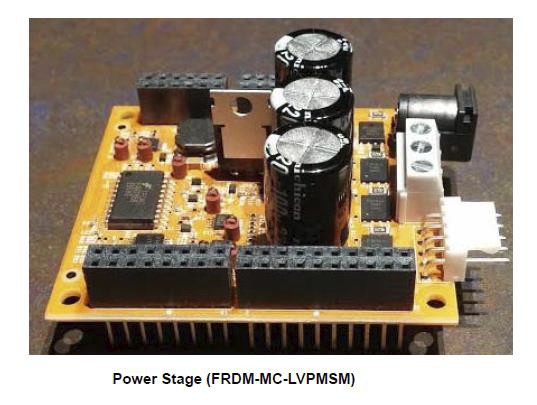
• If you did not yet install the “Kinetis-design-studio\_3.2.0” on your computer then locate it on your course website, download it to and install it.

1.1.4. Communication

• Kinetis development platforms are already preloaded with P&E

Microcomputer Systems (P&E Micro) openSDA software which

communicate serially via USB with your PC using P&E Micro Windows Hardware Interface Drivers.



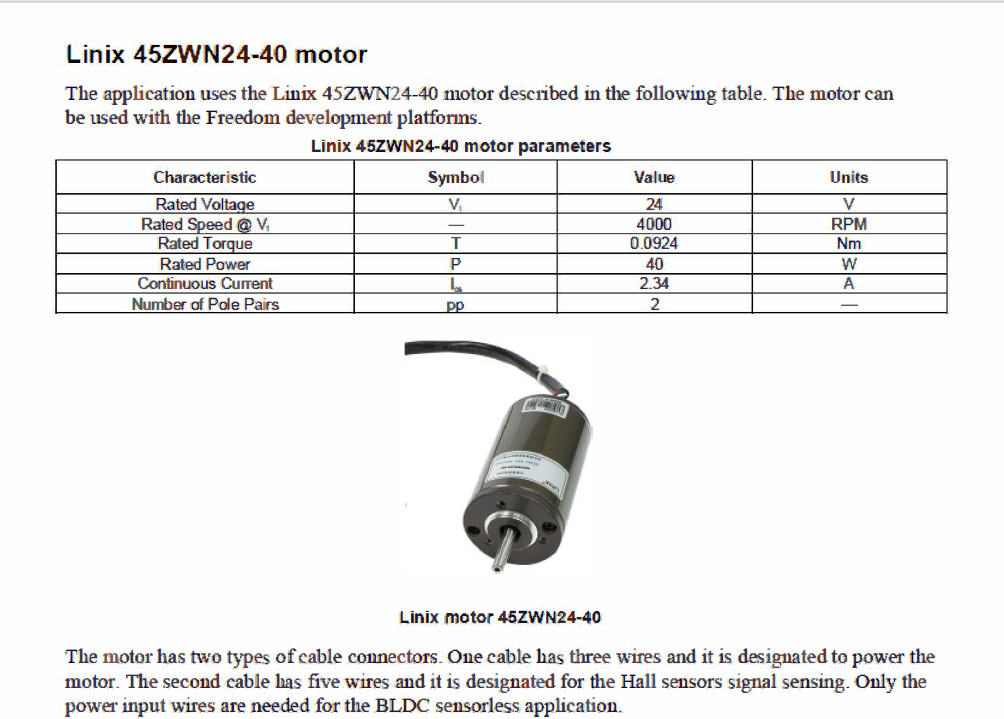
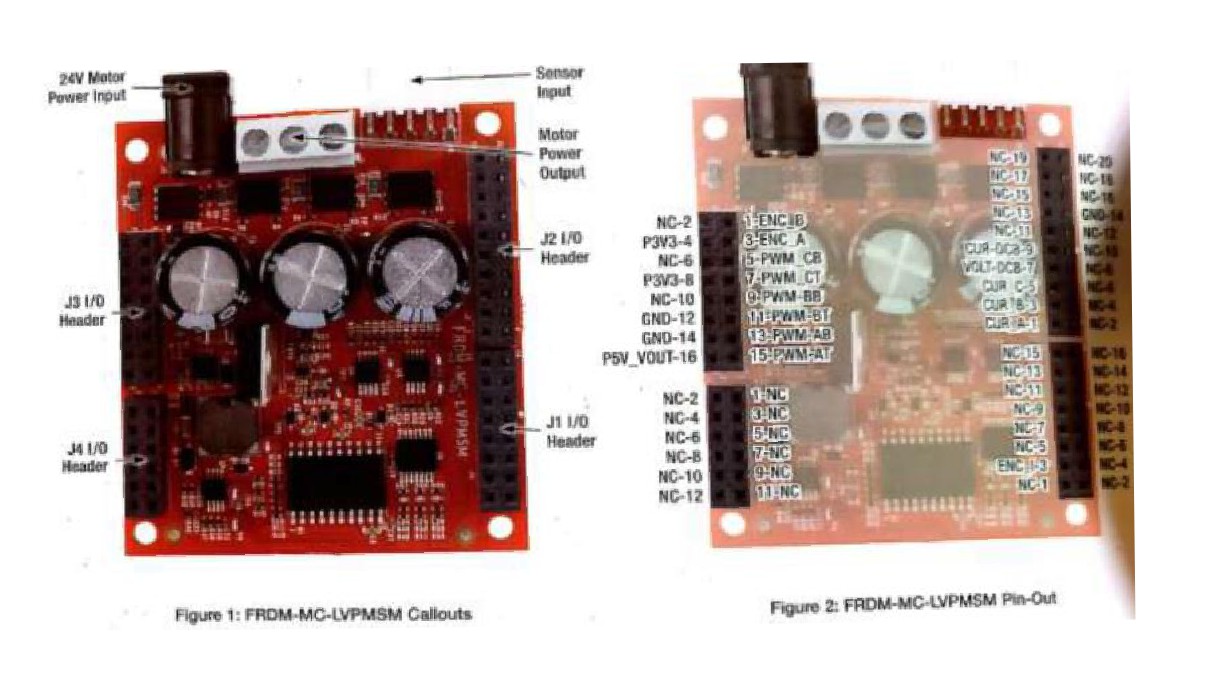
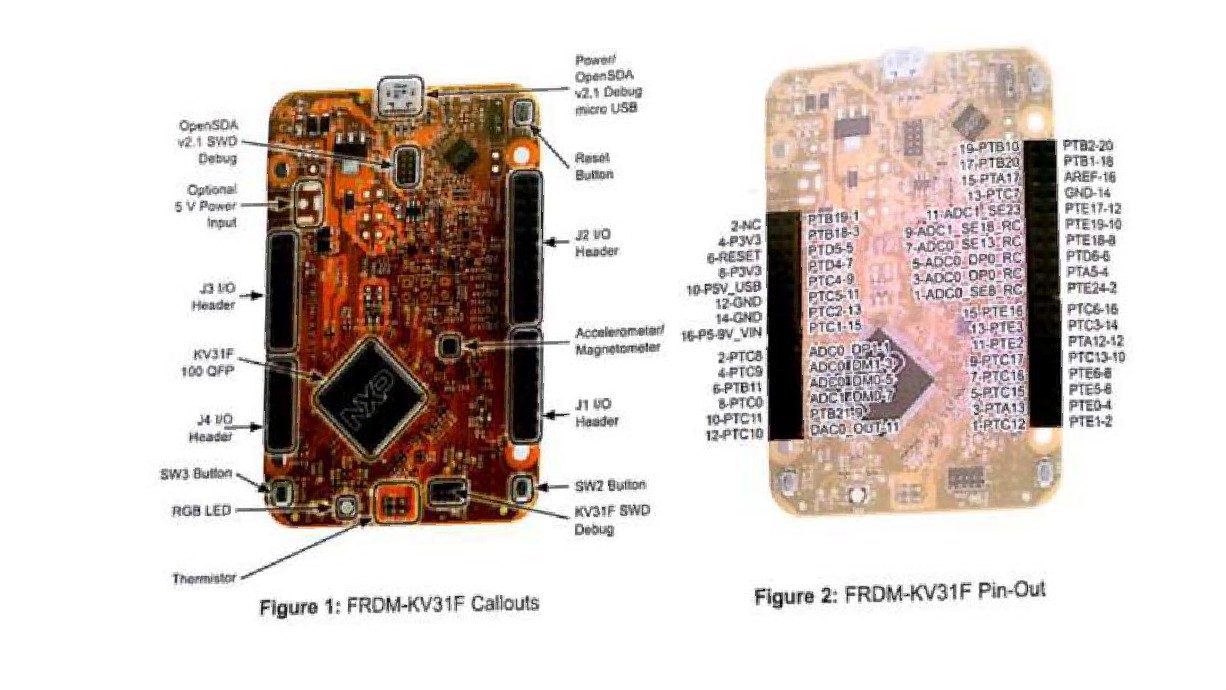
2. Locate your hardware:

• FRDM-MC-LVPMSM: Low-Voltage,

3 Phase Motor Control Module

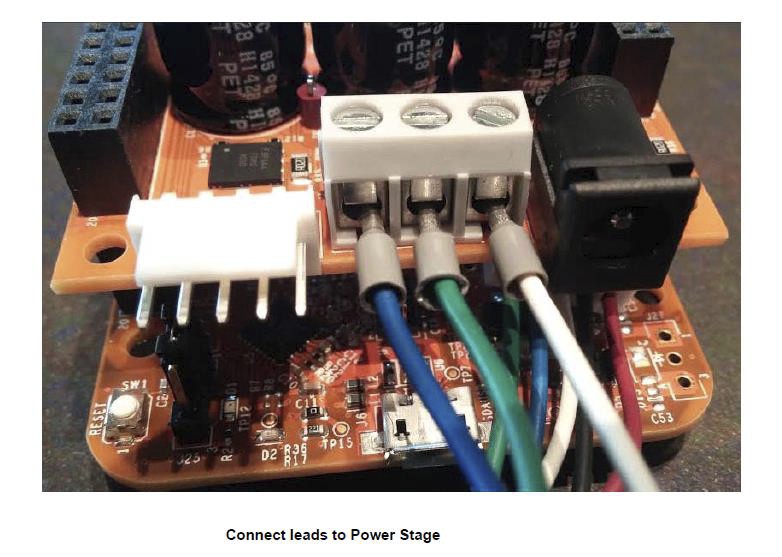
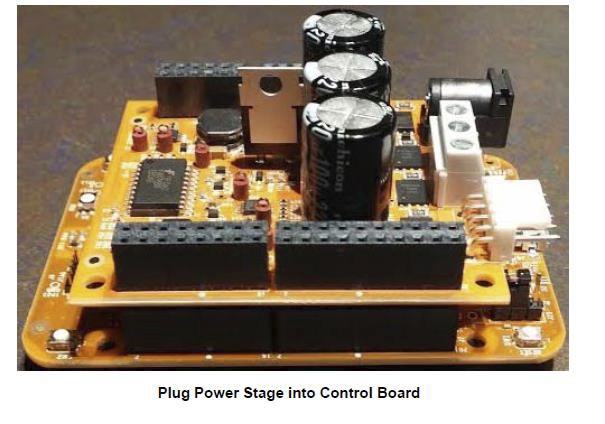
• FRDM-KV31F: Kinetis KV3x FRDM   
Development Module

• Power supply with 24V, 5A output and 5.5 x 2.1mm barrel connector



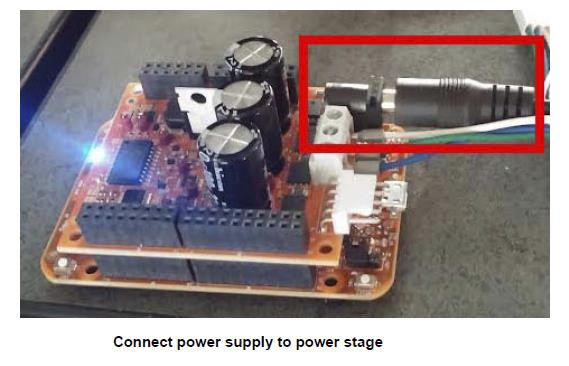
3. Connect your hardware.

• Connect the Power Stage and the Control Board by inserting the pins from the bottom of the Power Stage into the top of connectors on the Control Board. The default jumper configuration is appropriate.

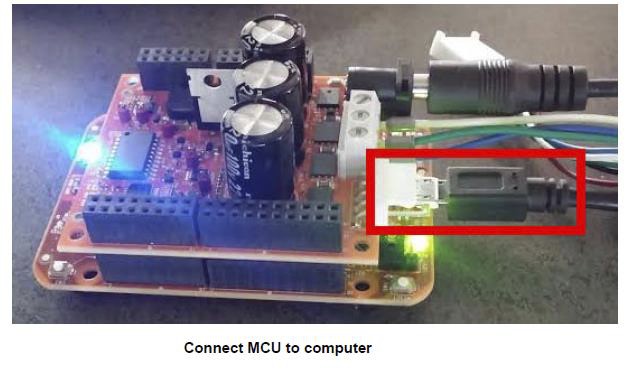


• Connect the power supply to the FRDM-MC-LVPMSM.

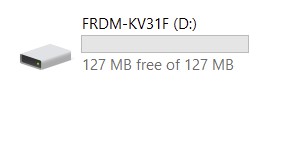
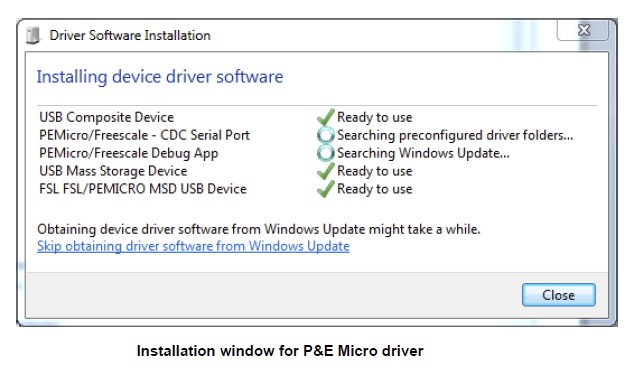
• Plug in Power Supply and connect it to the Power Stage.



Note: Be sure that you are using the correct 24 V power supply. Please check with your instructor.

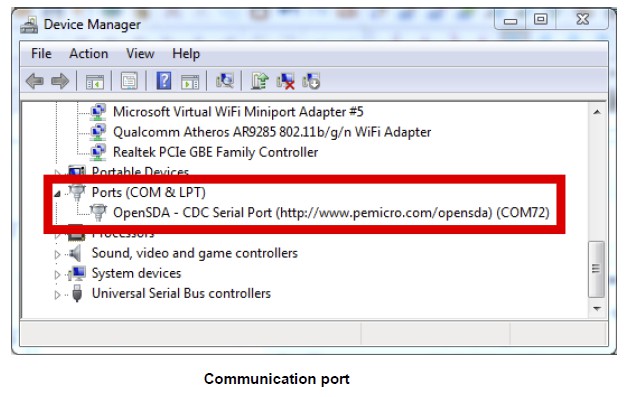


• Connect the Control Board to the computer using the USB cable.



Check that FRDM-KV31F appears as a drive.

• Confirm successful connection by verifying that a COM port has been assigned to a P&E Micro OpenSDA Serial Port



Part II: Run your project on MCU.

1. In your Kinetis Motor Suite, Click Project -> Run Project on MCU. This will Download the software required to run the Kinetis Motor Suite to the board
2. Wait for the download to finish. After download, the LED on the board should start glowing in various colors rather than a blinking LED.
3. KMS relies on the Kinetis Software Development Kit (KSDK) version 1.3.0.

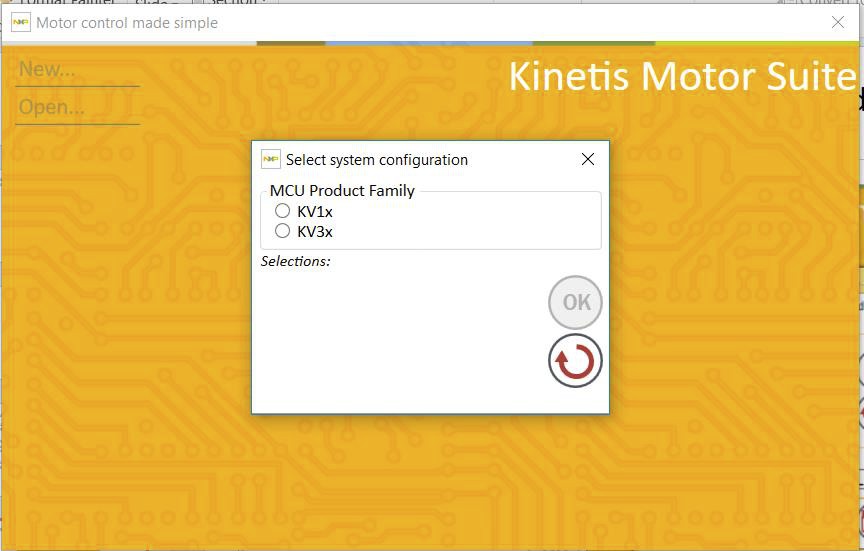
Part III: Run your Motor using Motor Tuner.

Velocity lab: Spin your motor using Motor Tuner.

• Just getting a motor to spin is typically challenging. However, Kinetis Motor Suite makes it easy.

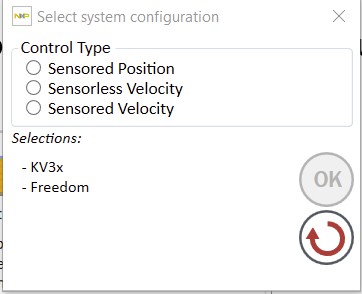
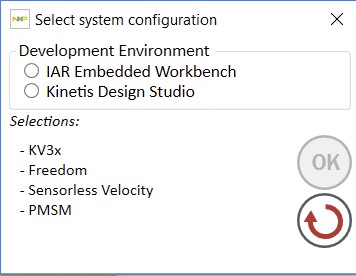
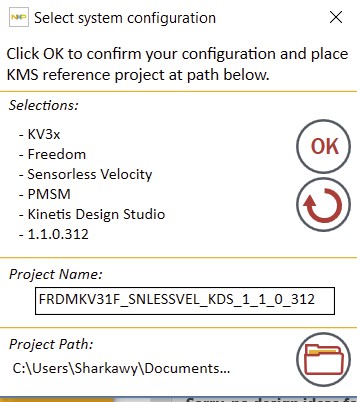
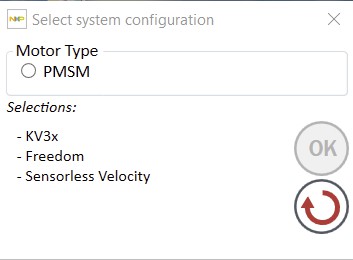
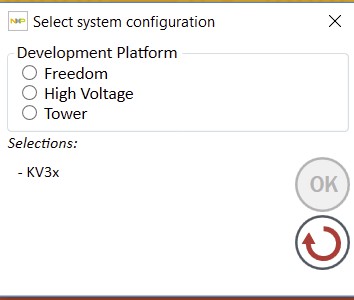
• We are using KMS Motor Tuner to characterize a motor and then operate it across the speed range - with minimal user configuration and without coding.

• Launch KMS by double-clicking the Desktop icon.

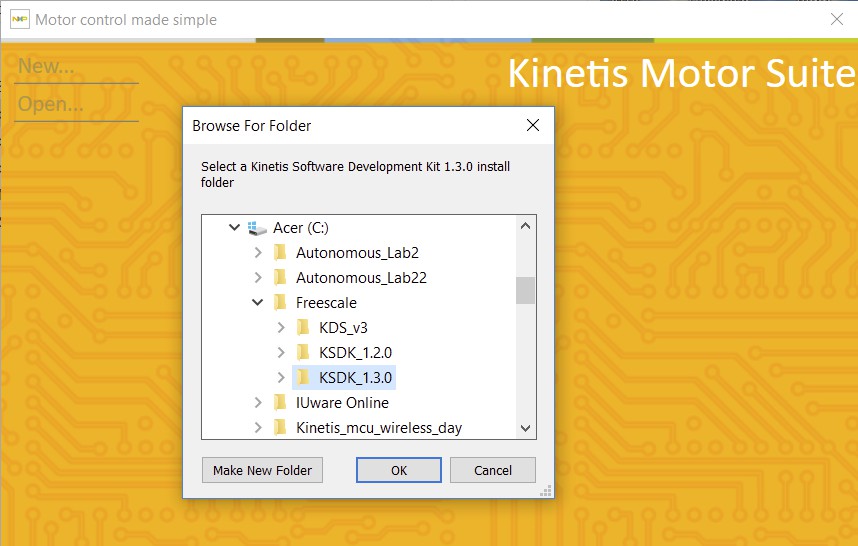


• Select New͙ and specify your desired system configuration

Select KV31F.



* After selecting all the above configuration select path for your project to be saved, also you can set the project name.
* Go to Project 🡪 Select Path 🡪 Kinetis SDK

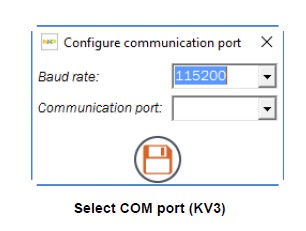


• Find and select the folder

named KSDK\_1.3.0. The default installation location is:

C:\Freescale\KSDK\_1.3.0.

Specify the location of KSDK 1.3.0

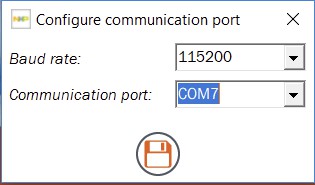


• Wait a minute or two for the

unzipping and placement of the   
KMS reference project.

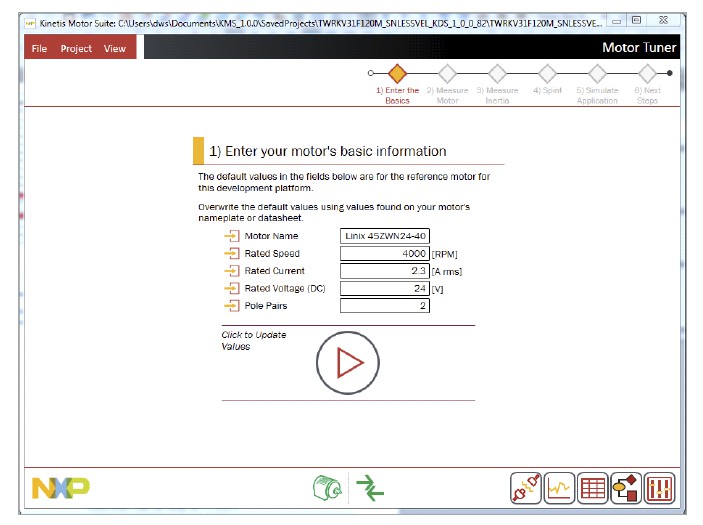
• Select the proper communication port for your board from the dropdown menu.

• The proper communication port can be determined by looking in your Windows Device Manager (accessible from Control Panel)   
and finding the OpenSDA communication port.



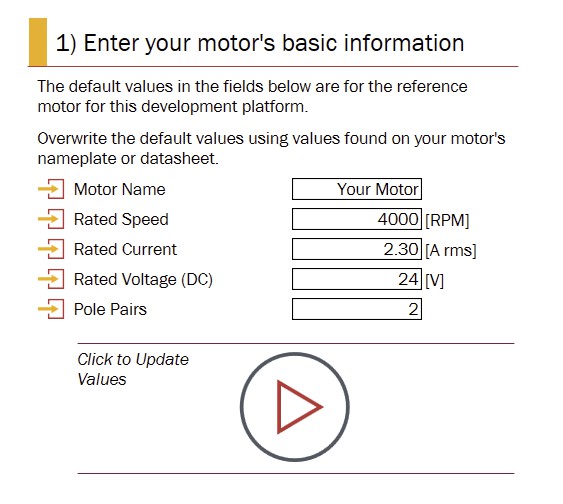
• KMS should open to its Main Window and indicate successful

communication by virtue of green arrows at the bottom of the screen.



Successful opening of KMS main window

1. Enter the basics.



• KMS opens to the first page of Motor Tuner, the wizard-style interface for identifying and running your motor that is the subject of this lab.

• The first step in commissioning your system is to enter your motor’s basic information.

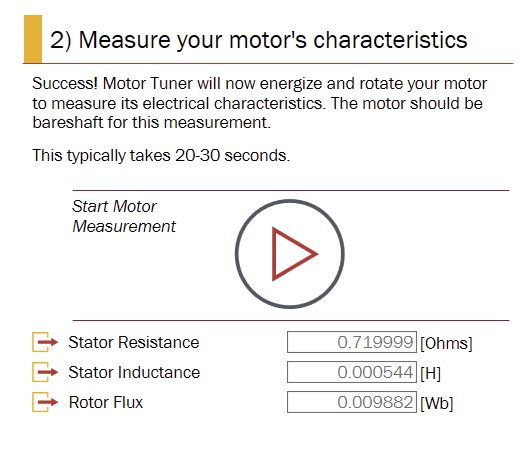
• Linix 45ZWN24-40 information is prepopulated in KMS for the Freedom development platform (sensor less velocity).



• Click to update the basic motor values.

2. Measure motor

• Motor Tuner applies voltage to measure the motor’s resistance and inductance. Motor Tuner then rotates the motor’s shaft at a low speed to measure the rotor flux. Click to measure your motor’s characteristics.



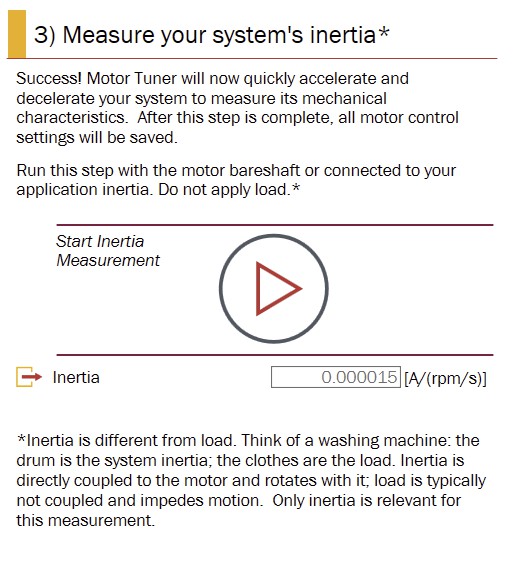
3. Measure inertia

• Inertia is important for controlling the motion of the application but is often neglected by traditional approaches.

• Motor Tuner uses inertia as a direct input to create an appropriate model of the system for advanced control.

• Your motor is running without anything connected to the motor shaft, then Motor Tuner measures the inertia represented by the motor shaft.

• Click to start inertia measurement. The motor spins briefly. If   
successful in the default configuration, KMS also updates the reference project that you created upon launching KMS with your   
motor and inertia characteristics.



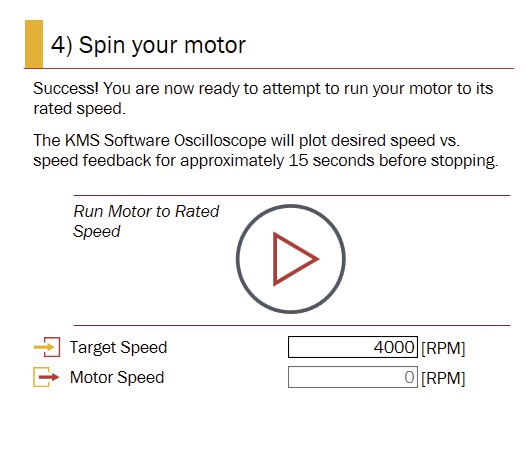
4. Spin

• Motor Tuner automatically advances to the next step when the system inertia measurement is successful.

• In this step, Motor Tuner runs your motor to its rated speed, which you entered or confirmed on step 1) Enter the Basics.

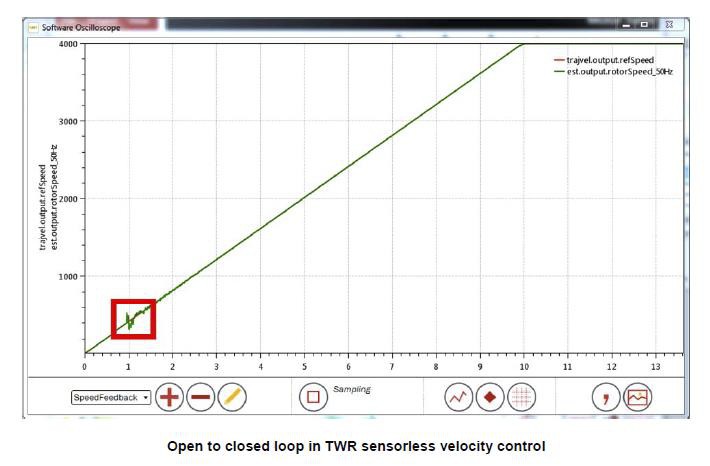
• Click to run your motor to the rated speed.

Note: If it is not desirable to run your motor to rated speed (e.g., if your application is attached and the rated speed of the motor is too high for your application), overwrite the Target Speed value to a more suitable value before clicking to Run.



• Motor Tuner automatically activates the Software Oscilloscope so that you can watch the performance of your motor as it transitions to the rated speed. Commanded speed is plotted in red and estimated actual speed is plotted in green.

• When operating under sensor less velocity control, KMS transitions from open to closed loop speed control upon reaching 10% of the motor’s rated speed.



• This results in a noticeable bump in the previous plot as the system attempts to lock onto a signal that is a function of speed and thus small when running at low speed.

• This transition from open to closed loop control is a key challenge of sensor less velocity control.

• By contrast, when running sensor-based velocity control, the motor is always in closed loop control due to the external sensor.

• This results in smoother operation at low speed, which is a chief benefit of   
sensor-based control. Nevertheless, the encoder must catch up when starting   
from zero, so shows a small divergence from commanded speed near zero.

• The motor spins to the Target Speed and remains at that speed for 5 seconds. After that, Motor Tuner closes the Software Oscilloscope and proceeds to the next step.

5. Simulate application.

• Motion Sequence Builder is one of the tools available in KMS. It allows you to easily build complex motion sequences and automatically generate application code.

• One of the provided motion sequences simulates simple washing machine behavior. The motor:

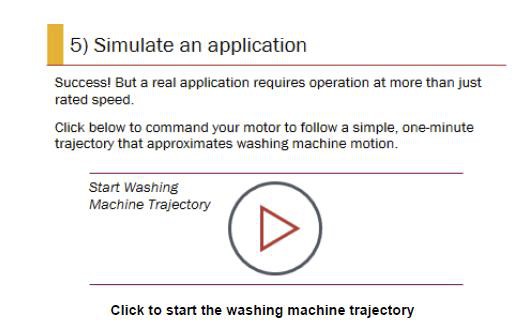
ͻ remains at 0 rpm speed during the "water fill" stage.

ͻ “agitates” by ramping to a certain speed, then reversing direction to reach

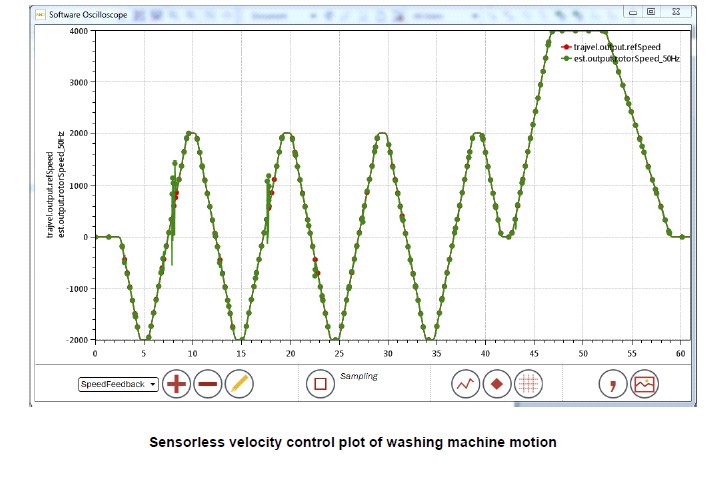
the same speed in the opposite direction (repeating this behavior several

times).   
ͻ ramps to twice the agitation speed during the “spin cycle”.

ͻ comes to a halt and concludes the motion sequence.

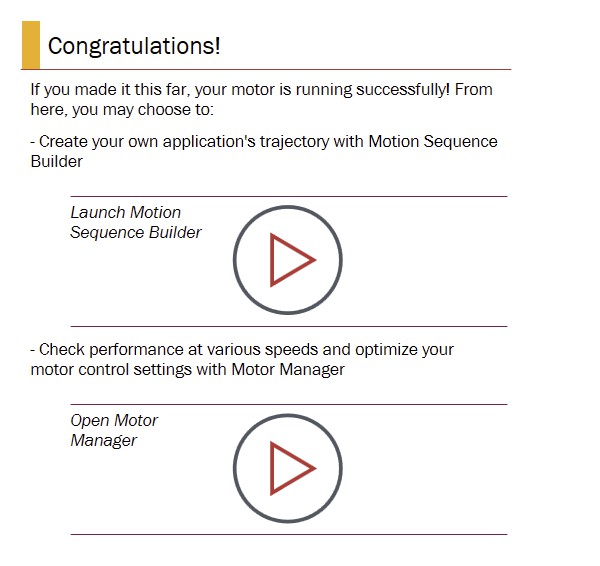


• Click to start the washing machine trajectory.



• Motor Tuner automatically launches the Software Oscilloscope so that

you can observe your motor’s performance as it executes the simulation.



Motor Manager will be used in Part IV of this lab

Summary

• In this lab, you performed the following steps to spin your motor:

• Entered basic information about the motor

• Characterized the electrical and mechanical system

• Ran the motor and validated its ability to achieve and hold rated speed

Part IV: Other methods for   
running your motor in KMS

Objective

• KMS is intended to allow flexibility for both beginner and expert users.

• In this part of the lab, spin your motor using Motor Manager, which is a superset of the functionality available in Motor Tuner, and the Watch Window, which allows runtime access to MCU variables.

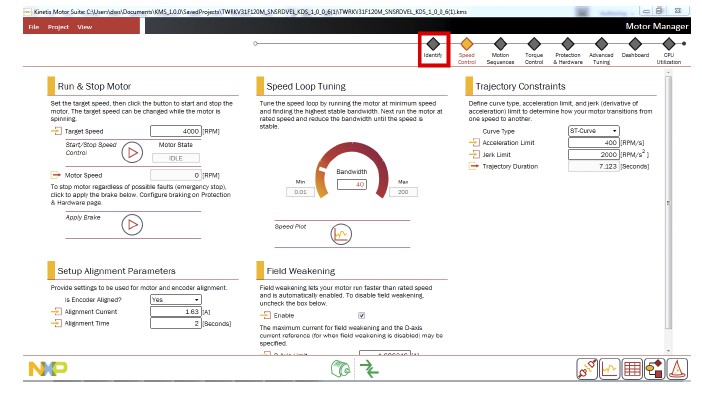
1. Start Motor Manager



• Click on Open Motor Manager.

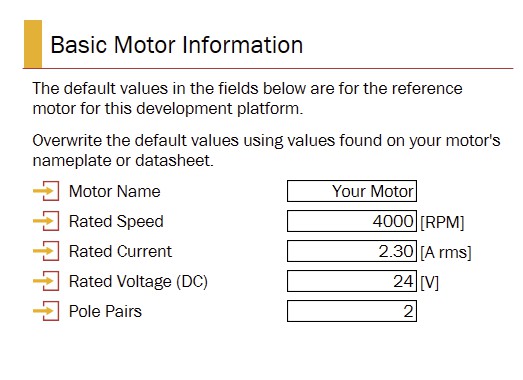


• Navigate to the Identify page by clicking on the appropriate selection in the navigation bar at top right.

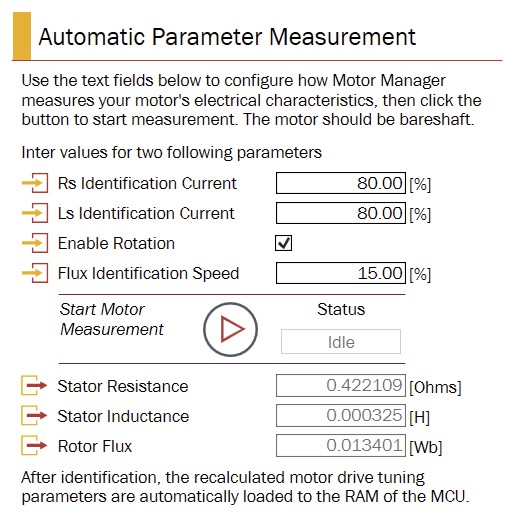


1. Basic motor information

• When using Motor Tuner to get your motor spinning, the first step was to enter the motor’s basic information. The same is true for Motor Manager. Enter the basic information for your motor in Basic Motor Parameters.

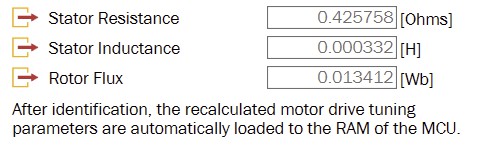


2. Automatic parameter measurement



• Motor Tuner applied a voltage to measure the motor’s resistance and inductance, and then rotated the motor’s shaft at a low speed to   
measure the rotor flux. Motor Manager does the same thing.

• Click to start the motor measurement.

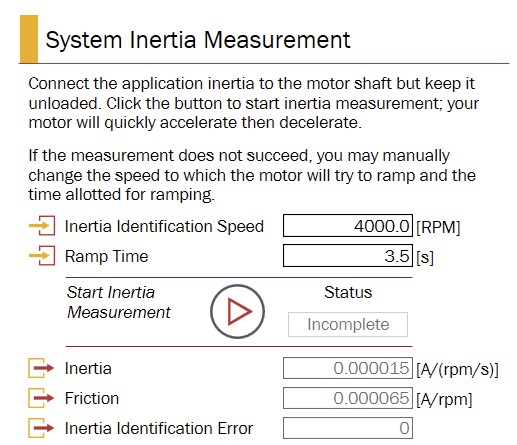


• After Motor Manager measures the motor parameters, they are updated on the screen

3. System inertia measurement

• In lab 1, part II: “spin your motor using Motor Tuner", we learned that Motor Tuner uses inertia as a direct input to create an appropriate model of the system for advanced control. Motor Manager does the same.

Note: When developing your application (not at this lab), connect the motor to the application inertia (anything that spins directly with the motor during operation) before running System Inertia Measurement.

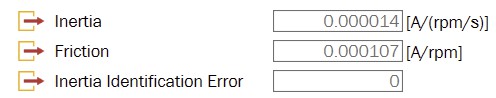


• Scroll down to System Inertia Measurement then click to start the inertia measurement.

• During system inertia measurement,   
the motor accelerates and decelerates in a manner governed by the configuration parameters inertia identification speed and ramp time.

• Unlike Motor Tuner, Motor manager allows you to make manual adjustments to system inertia measurement configuration.

• After Motor Manager identifies the system inertia, the inertia and friction values are displayed on the screen

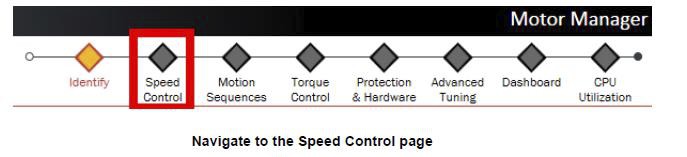
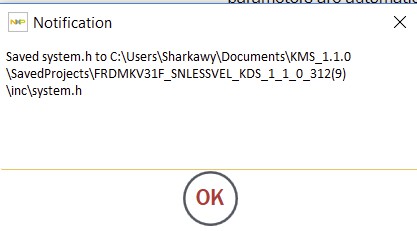
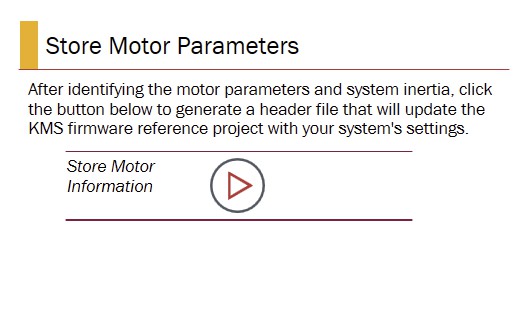


• KMS updates configuration values based on the measured motor parameters and inertia. These values can be saved to the Motor Observer reference project (created upon launching KMS) which can then be edited from your IDE. This happens automatically in Motor Tuner.

• In Motor Manager, you must click the Store Motor Information button to push your system

settings into the Motor Observer reference project. You receive a notification that the header file that contains this information has been saved in a particular location within the directory you defined after launching KMS.

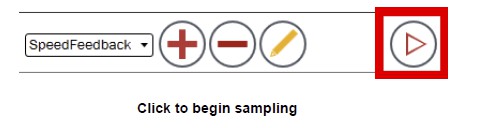
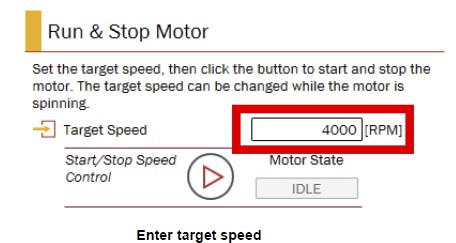
• Click OK after receiving this notification.



4. Start Motor.

• Click on the Speed Control page icon at top

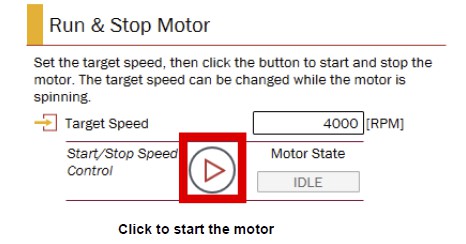
• Enter the motor’s rated speed as its Target Speed



• Click to activate the Software Oscilloscope’s Speed Plot.

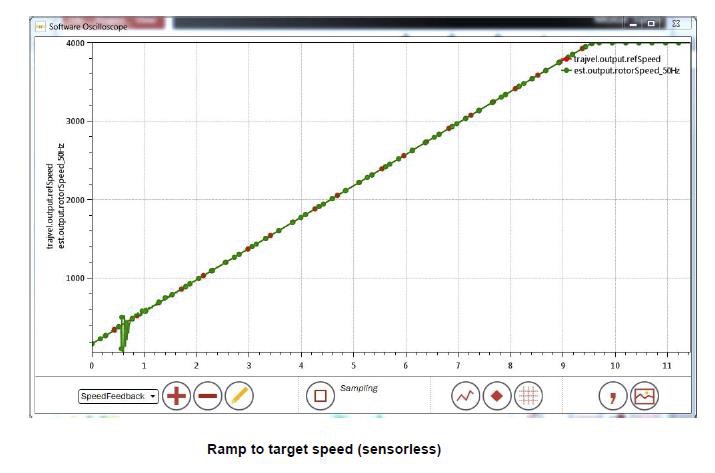
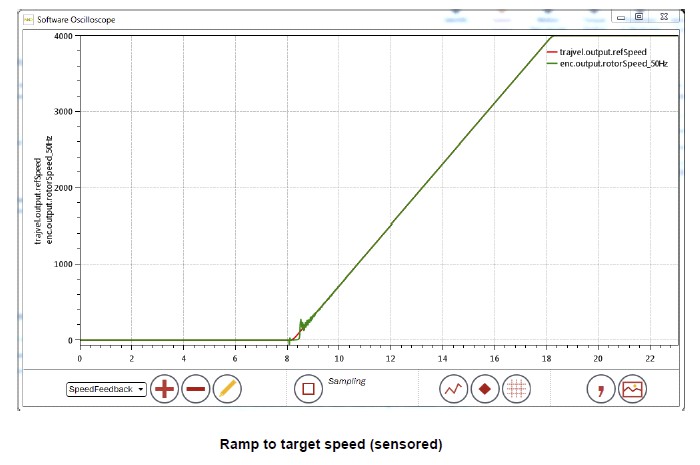
• Use the Software Oscilloscope to view your motor’s performance.

Click the Run button at the bottom of the oscilloscope display to start   
sampling.

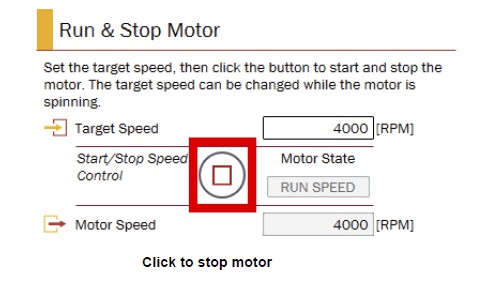


• Back in Motor Manager, click to Start Speed Control.

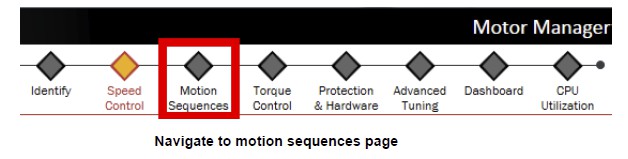
• You should see your motor ramp to the target speed.



• After the motor reaches and holds rated speed, as in Motor Tuner, click to Stop sampling.



• In Motor Manager, click to Stop the motor.

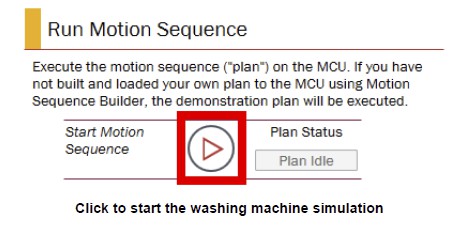


5. Run motion sequence

• Click on the Motion Sequences page icon at top.

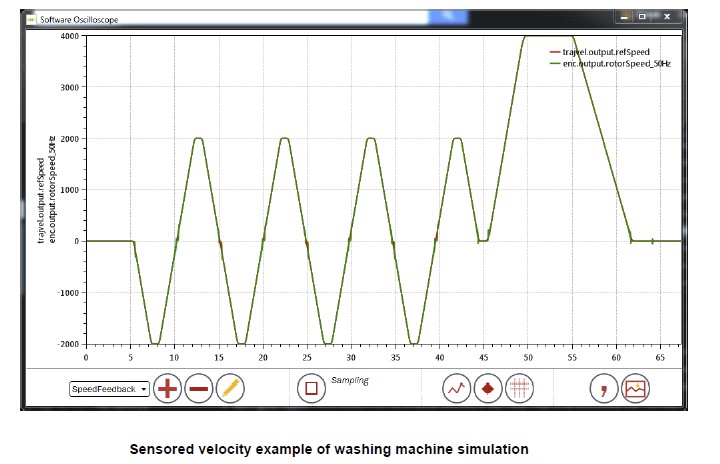
• In the Software Oscilloscope, click to restart sampling.

• Click to Start the washing machine simulation



• Return to the Software Oscilloscope so that you can view your motor’s performance.

• Watch your new motor run the same simple washing machine simulation as in "Velocity lab : spin your motor using Motor Tuner".



• Click to stop sampling.

6. Start Watch Window

• While Motor Manager offers more granular control of the core functions performed by Motor Tuner, the Watch Window goes even further.

• Via runtime access to MCU variables, the Watch Window can be used to set in motion the same functions as Motor Tuner and Motor Manager.

• However, the Watch Window exercises these functions as discrete elements, instead of as an integrated system.

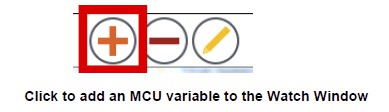
• For example, identifying motor parameters from the Watch Window will not automatically adapt the motor drive configuration based on updated values, as occurs in Motor Tuner and Motor Manager.

• Thus, to use the Watch Window properly, you should have some familiarity with the KMS firmware and with motor control principles.



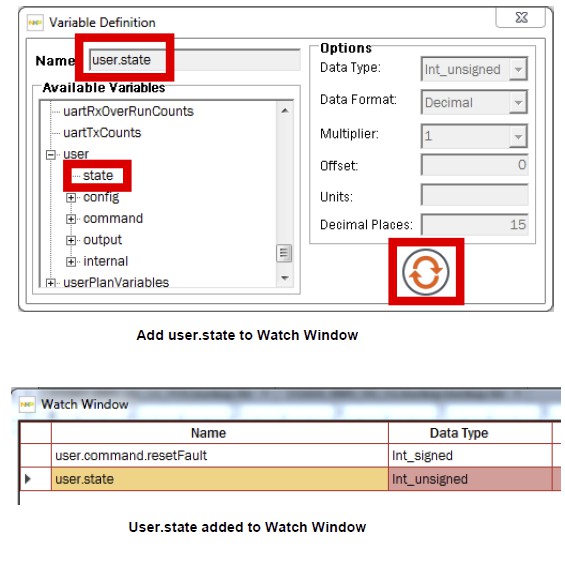
• The following steps, wherein only a subset of the functions described   
for Motor Tuner and Motor Manager are exercised, serve to reinforce   
the discrete nature of operation from the Watch Window.

• Click to activate the Watch Window using the button at bottom right



7. Automatic parameter measurement - flux only

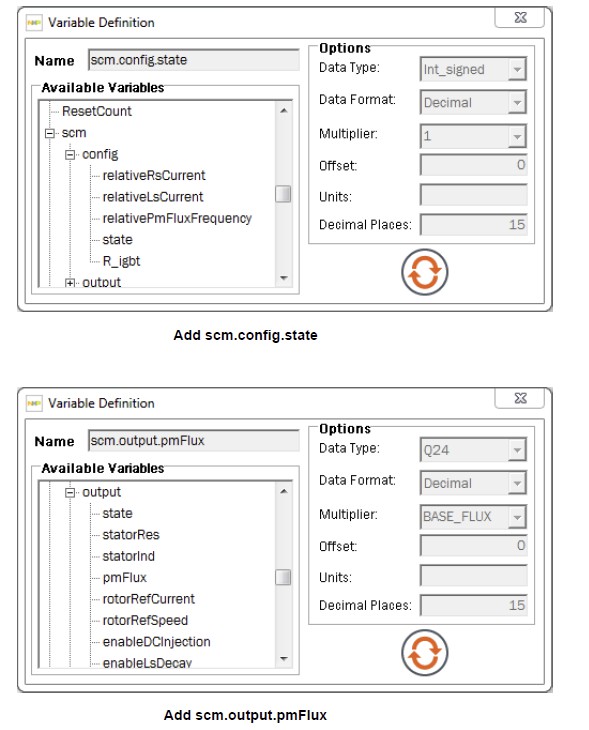
• Click the Add button at bottom left to show a list of the parsed   
variables available from the MCU.



• Scroll down to locate the “user”   
group, which aggregates variables   
found in the User module of   
Motor Observer firmware.

• Expand the user group, select   
“state,” and click to refresh.

• The variable user.state is added to the Watch Window.



• The variable user.state is intended   
to provide easy access to KMS’   
operating modes, including several

that are responsible for the Motor

Tuner and Motor Manager

functions you have already seen.

• To replicate a part of the motor   
measurement routine, click the   
Add button again and find the

variable scm.config.state.

• Do the same for

scm.output.pmFlux.

• The term “scm” refers to variables used in the Self Commissioning   
module of KMS firmware.

• Scm.config.state determines which value the system is trying to

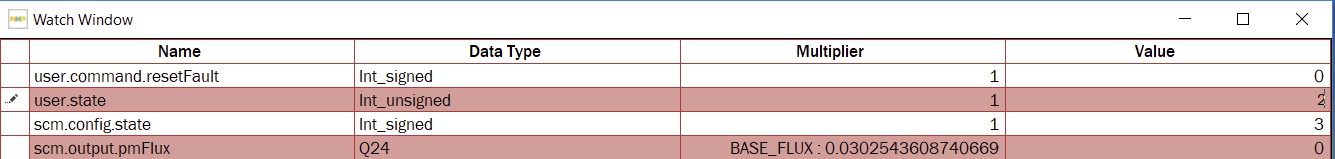
measure (resistance, inductance, or flux), and scm.output.pmFlux is   
the result of the flux measurement.

• To run the flux measurement from the Watch Window, first click the   
Run button to activate live updating of the Watch Window variables.

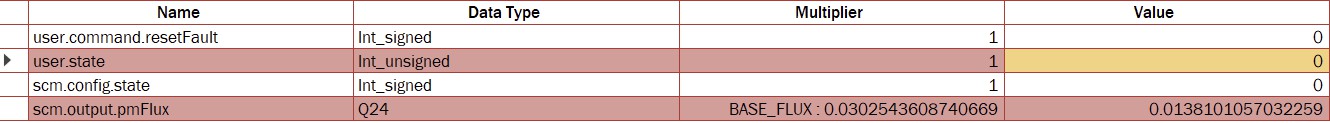


• Type 3 into the Value column for scm.config.state. This specifies flux   
as the desired measurement.

• Type 2 into the Value column for user.state. This tells KMS firmware to   
run in Self Commissioning mode - and thus measure flux given the   
value configured in the previous step.



(you may need to stop and run again)



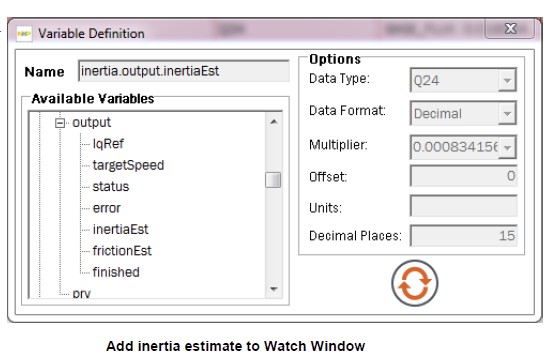
• The motor spins and on completion of the measurement, the value   
for scm.output.pmFlux will update in the Watch Window.

• User.state and scm.config.state return to the idle, 0 state.



8. System inertia measurement

• Click Stop button at bottom middle of Watch Window to halt live   
updating of MCU variables.



• Now add the variable inertia.output.inertiaEst.

• This represents the value resulting from the system inertia   
measurement

• Click to Run live updating in the Watch Window.



• Enter a value of 3 into the Value column for user.state.

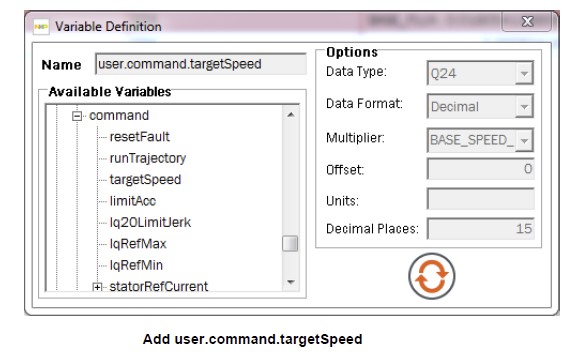
• This will place the firmware into inertia estimation mode.

• The motor spins and the value for inertia.est.inertiaEst updates.



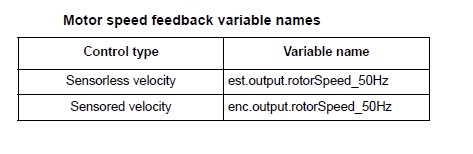
9. Start Motor

• Click to Stop updating the Watch Window again



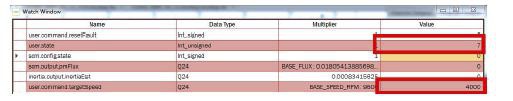
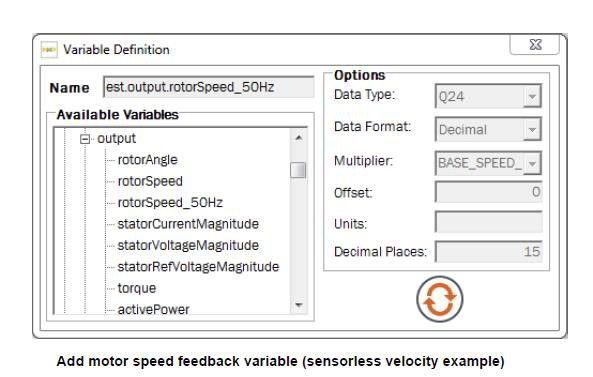
• Find and add the variable user.command.targetSpeed.

• This represents the speed command being sent to the motor.



• Find and add the variable showing the motor’s feedback speed. In

sensorless operation, this information comes from the Estimator module   
(EST).



• Click to Run live updating in the Watch Window.



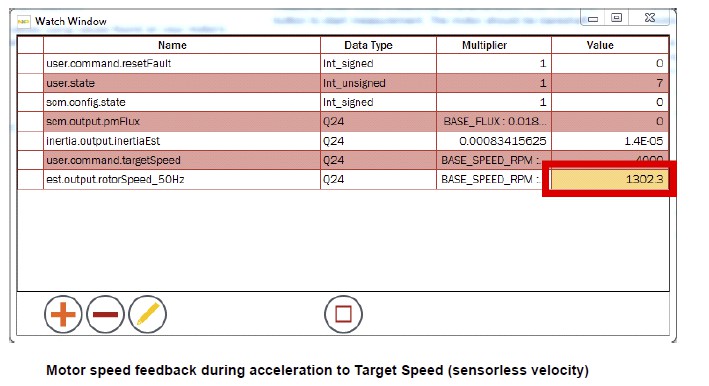
• Enter the Rated Speed for your motor into the Value column for

user.command.targetSpeed, then type a value of 7 into the Value column for user.state.

• This value for user.state places the motor into Speed Control mode.

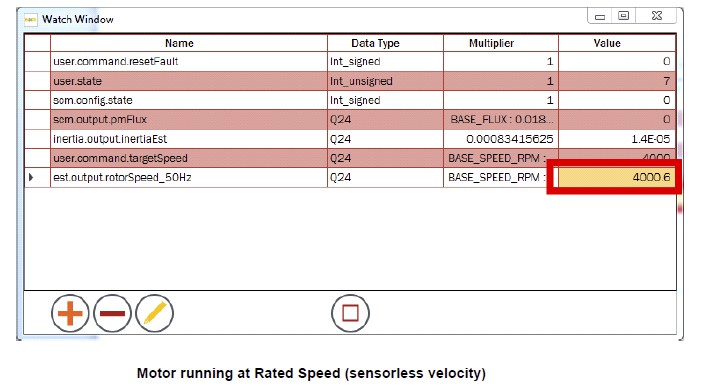
• The motor spins to Rated Speed just as it did in Motor Tuner and Motor Manager.

• Observe the motor speed feedback variable to see it gradually reach the   
Target Speed.



10. Run motion sequence

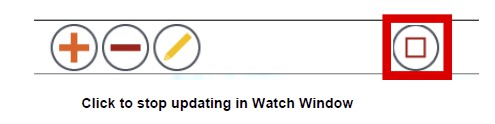
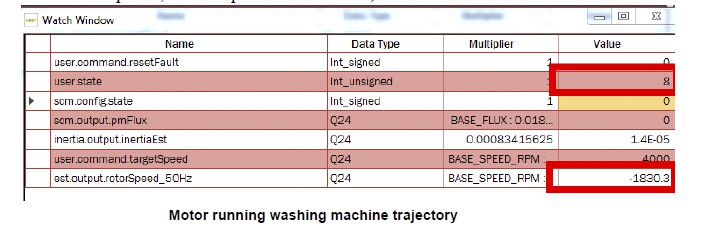
• Wait for the motor to reach Rated Speed according to your speed   
feedback value.



• With the Watch Window still live updating, type a value of 8 into the Value column for user.state.

• This starts the application motion sequence.

• First the motor slows to a stop, then proceeds through the washing machine sequence. Verify this by watching motor speed feedback (user.command.targetSpeed does not update because the speed reference   
is now coming from the motion sequence, not an explicit user command).



• Click to Stop updating the Watch Window after the motion sequence is complete.

Summary

• In this lab, you performed the following steps to spin your motor outside of the Motor Tuner wizard:

• measured motor and inertia from Motor Manager’s Identify page

• commanded the motor to run at a configurable speed from Motor Manager’s Speed Control page

• ran the example motion sequence from Motor Manager’s Motion Sequences page

• used the Watch Window to explore how MCU variables relate to PC GUI functions

• triggered motor & inertia measurements, speed control, and motion sequence operation from the Watch Window