Hands-On – Get your motor spinning with Kinetis Motor Suite on Kinetis KV1x,KV3x and KV4x

Philip Drake

NXP Systems Applications Engineer

May 2018 | AMF-IND-T3033

SECURE CONNECTIONS
FOR A SMARTER WORLD

Agenda

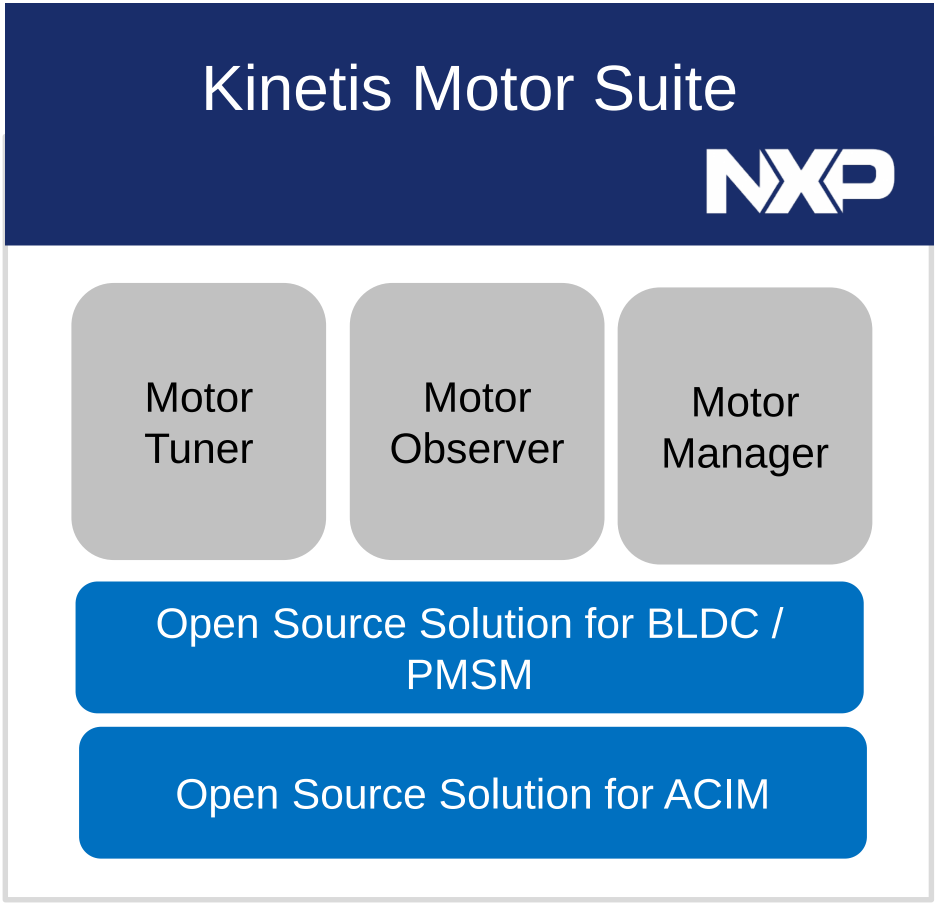
- 0.1 Updates – Overview for KMS 1.2
- 0.2 KMS Technology
- 0.3 MCU and KMS Roadmap
- 0.4 Labs 1-8
- 0.5 Enablement
- 0.6 Resources – Community Support

Questions and Answers as we go

0.1 KMS: Overview & Updates

Kinetis Motor Suite v1.2

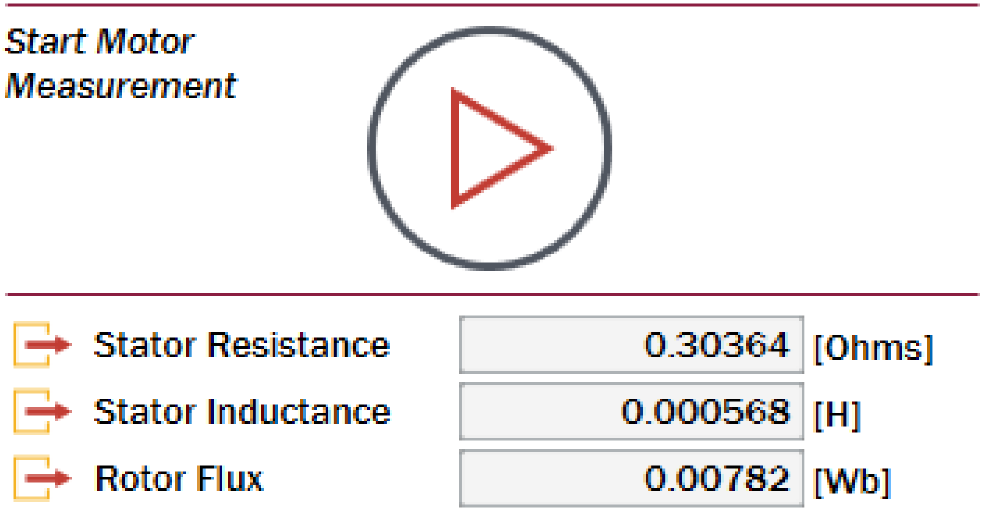
- Kinetis Motor Suite, or KMS, is a software solution that builds on the Kinetis V series MCU portfolio, providing all the low level and middleware software required to tune and control your BLDC, PMSM, and ACIM motors.
- KMS enables you to focus your resources on your end application, removing the complex and time consuming task associated with motor control solution development.
- Increased efficiency with
  - Active Disturbance Rejection
  - Single Parameter Tuning
  - Automatic Motor Parameter Identification
  - Automatic System Inertia Estimation
- New support for:
  - MCUXpresso SW and Tools (SDK, IDE, and Config Tools)
  - AC Induction Motor support on KV3x and KX4x
  - Hall effect sensor-enabled startup to sensorless velocity control
  - Improved mass erase protections for MCU flash


Kinetis Motor Suite
NXP
Motor Tuner
Motor Observer
Motor Manager
Open Source Solution for BLDC / PMSM
Open Source Solution for ACIM

0.2 Kinetis Motor Suite: The Technology

Motor Parameter Identification

- Automatically identifies motor electrical parameters
  - Resistance
  - Inductance
  - Flux
- Uses these to tune the motor control loops
- Designed to be done during development
- Should be done with motor disconnected from any load, if possible



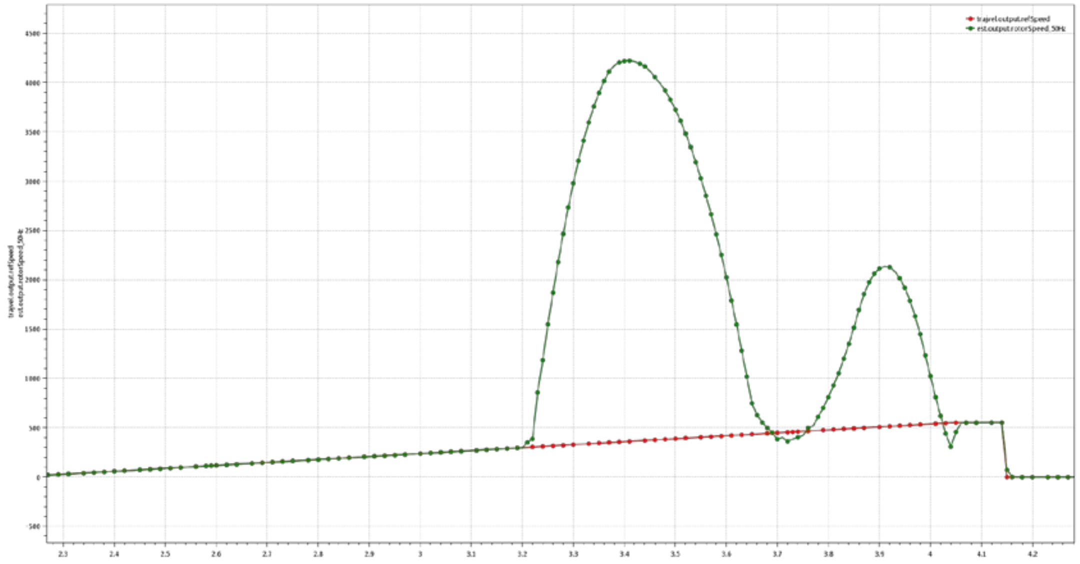
Stator Resistance 0.30364 [Ohms]

Stator Inductance 0.000568 [H]

Rotor Flux 0.00782 [Wb]

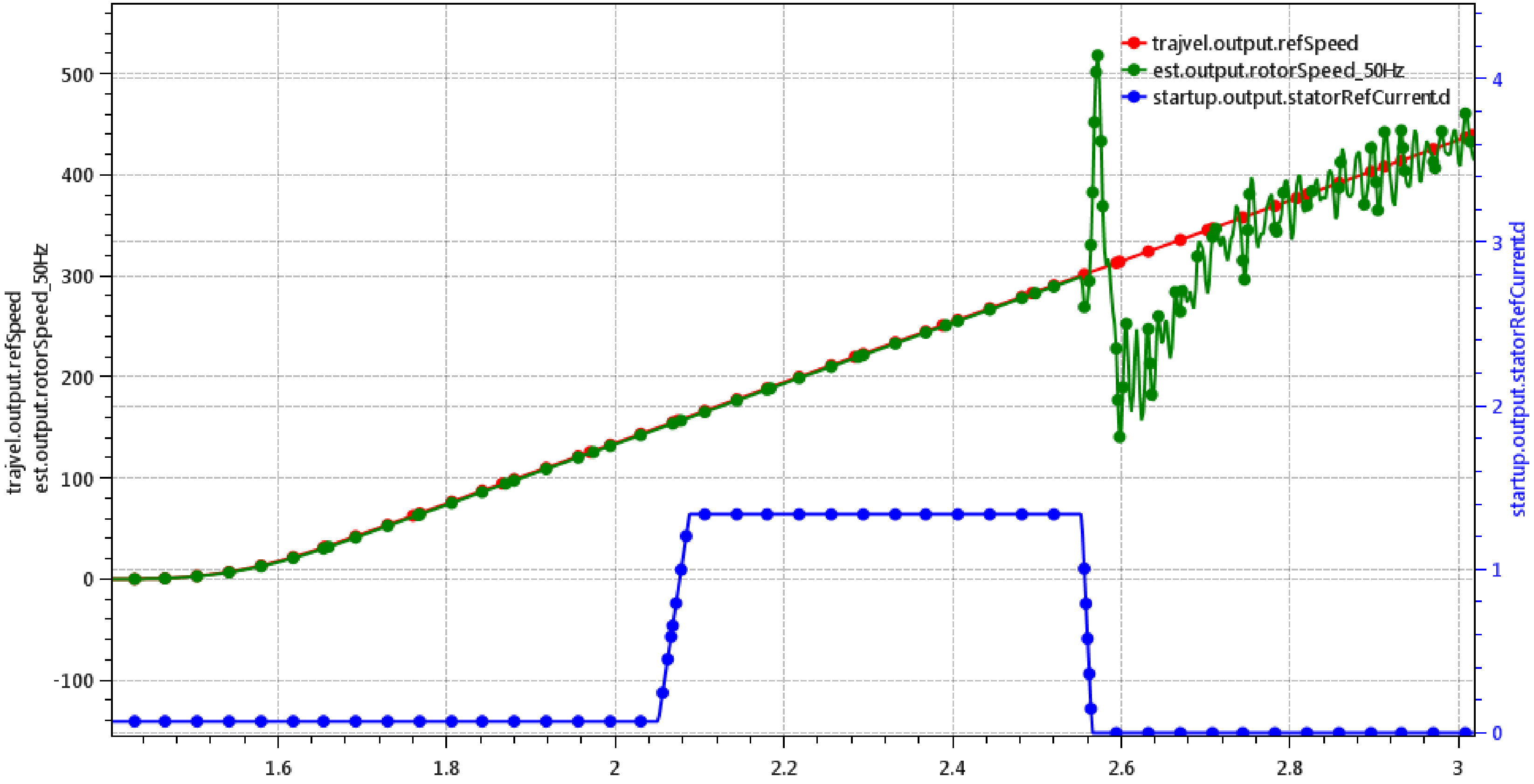
Inertia Estimation

- Value is critical for good motion control
- PI tuning indirectly takes this into account
  - Adjusting the Kp & Ki “tune” the controller for the motion system dynamics
- Estimating this value and directly using it in control yields better performance
- Designed to be done during development

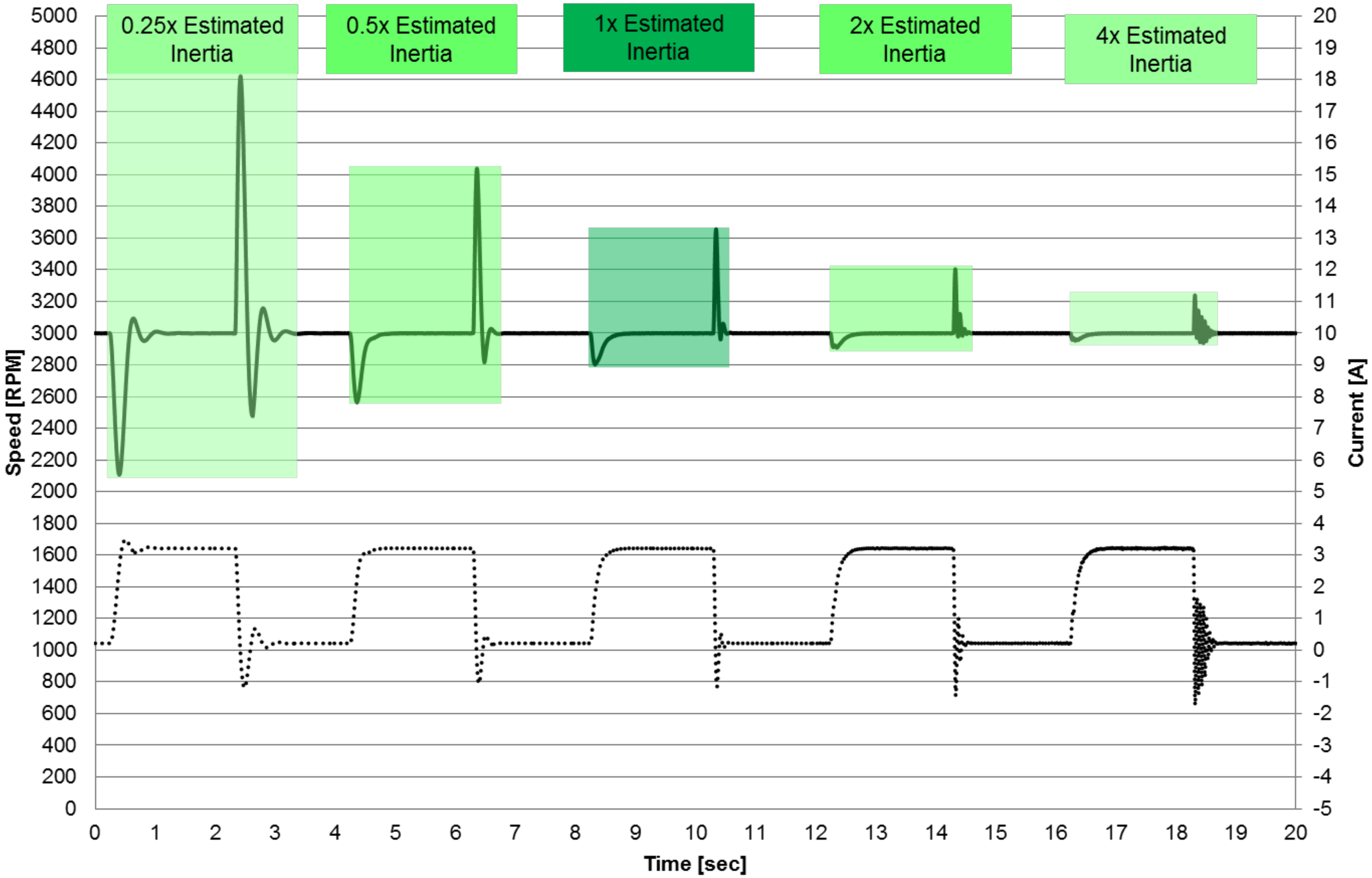


Soft Startup

- Automatically increases current to ensure successful startup
- Will retry startup if unsuccessful

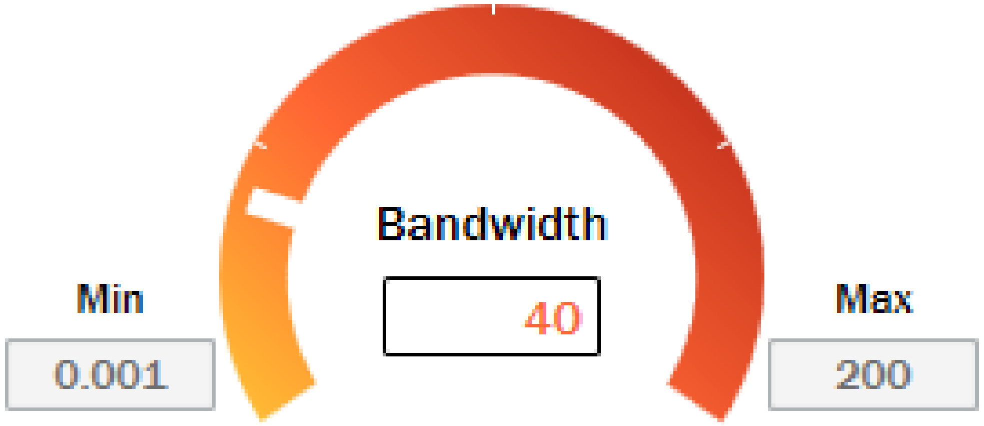


Inertia Range Tolerance



Speed Controller

- Disturbance-rejecting controller
  - Not purely an error-based controller
  - Based on ADRC algorithm
- Single variable to tune response
  - Typically effective across full variable speed and load range



Bandwidth 40

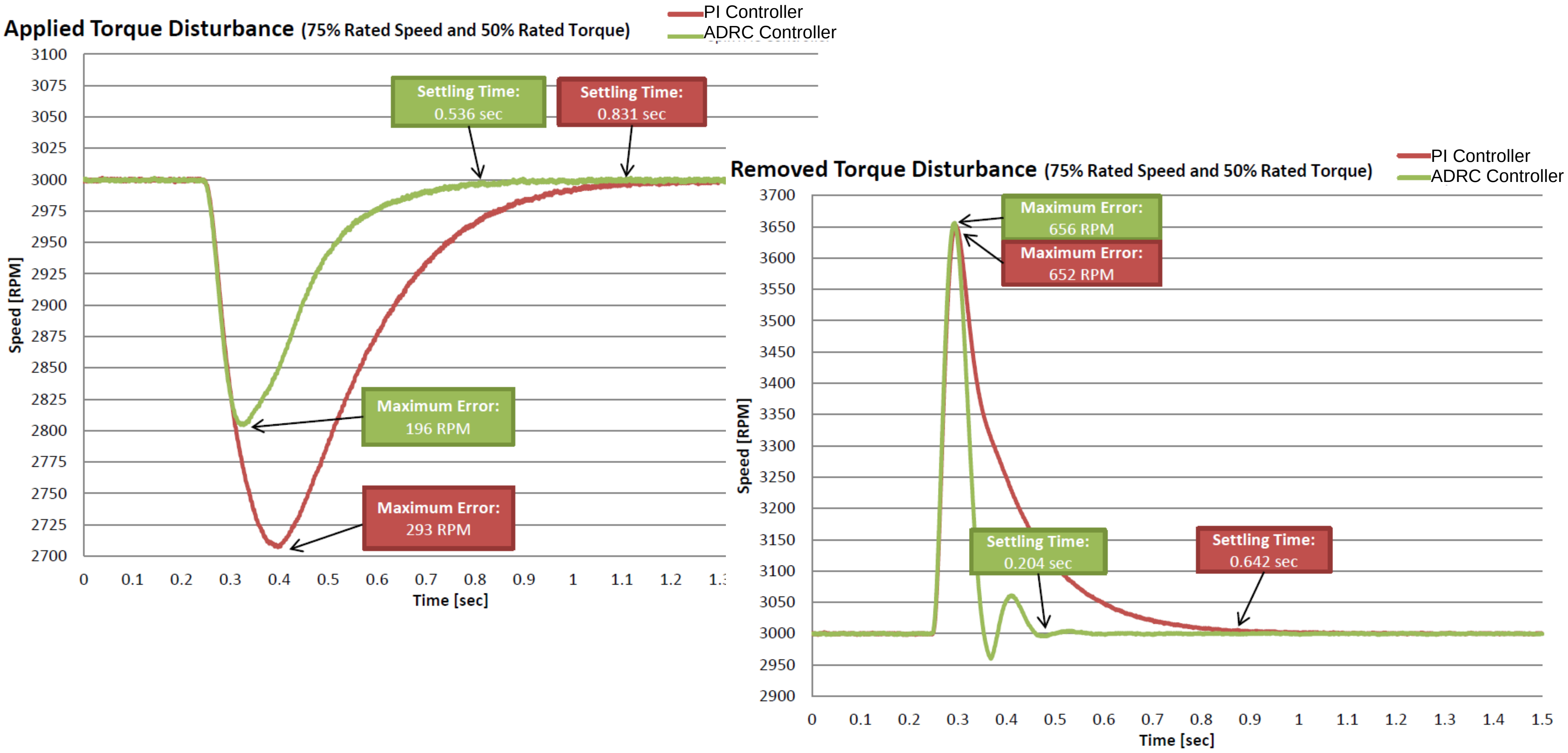
Min 0.001

Max 200

What is ADRC?

- **A**ctive **D**isturbance **R**ejection **C**ontrol
- Fundamentally different control technique that outperforms PID
- Based on a minimum information model
  - System Inertia -> Measurable
  - System Order -> Known
- Disturbance based controller
  - Any non-ideal behavior (i.e. load, wear, inertia change) is observed and compensated for by the controller
- Single parameter control tuning

Performance Comparison



Applied Torque Disturbance (75% Rated Speed and 50% Rated Torque)

Settling Time: 0.536 sec

Settling Time: 0.831 sec

Maximum Error: 196 RPM

Maximum Error: 293 RPM

Removed Torque Disturbance (75% Rated Speed and 50% Rated Torque)

Maximum Error: 656 RPM

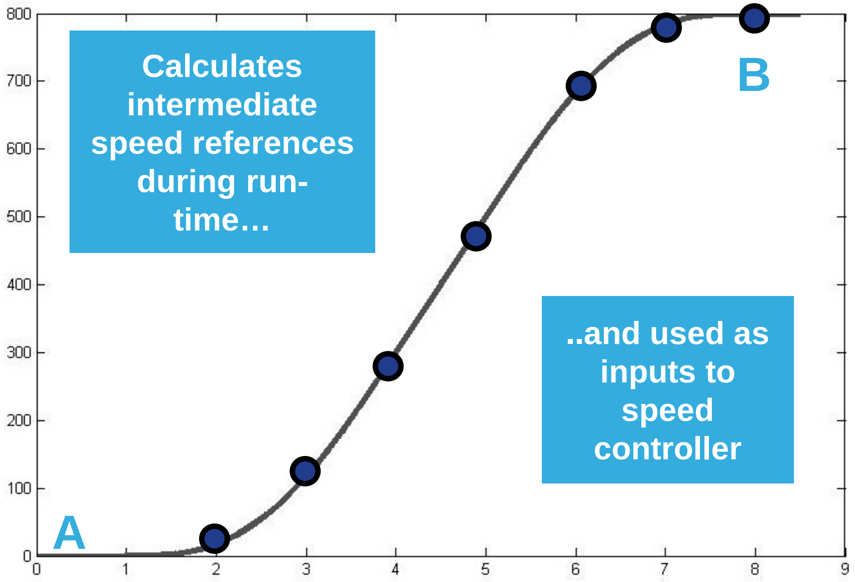
Maximum Error: 652 RPM

Settling Time: 0.204 sec

Settling Time: 0.642 sec

Advanced Profile Generation

- Constraint-based
  - User provides limits on motion
  - Profile will always respect those limits
- Time-optimal
  - Profile will be the fastest possible within limits
  - Profile will be sample time aligned
- Run-time Calculated
  - Profile is calculated on MCU
- Each profile can use a unique set of limits

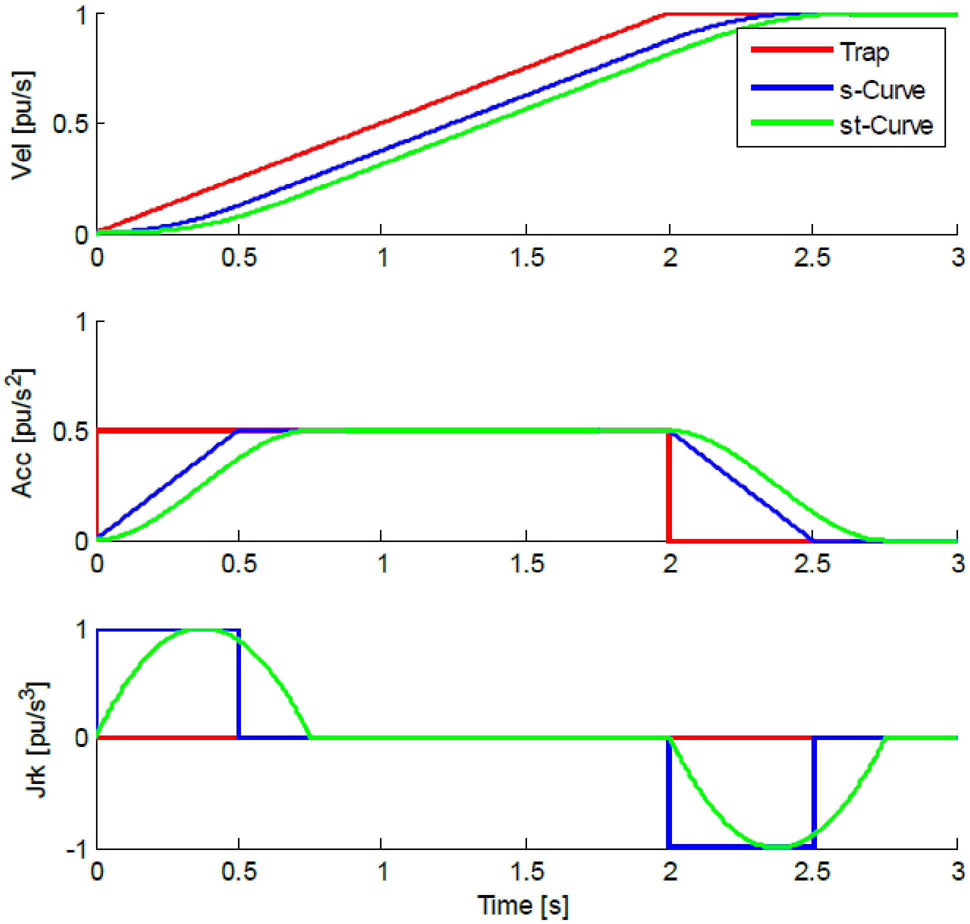


Calculates intermediate speed references during run-time…

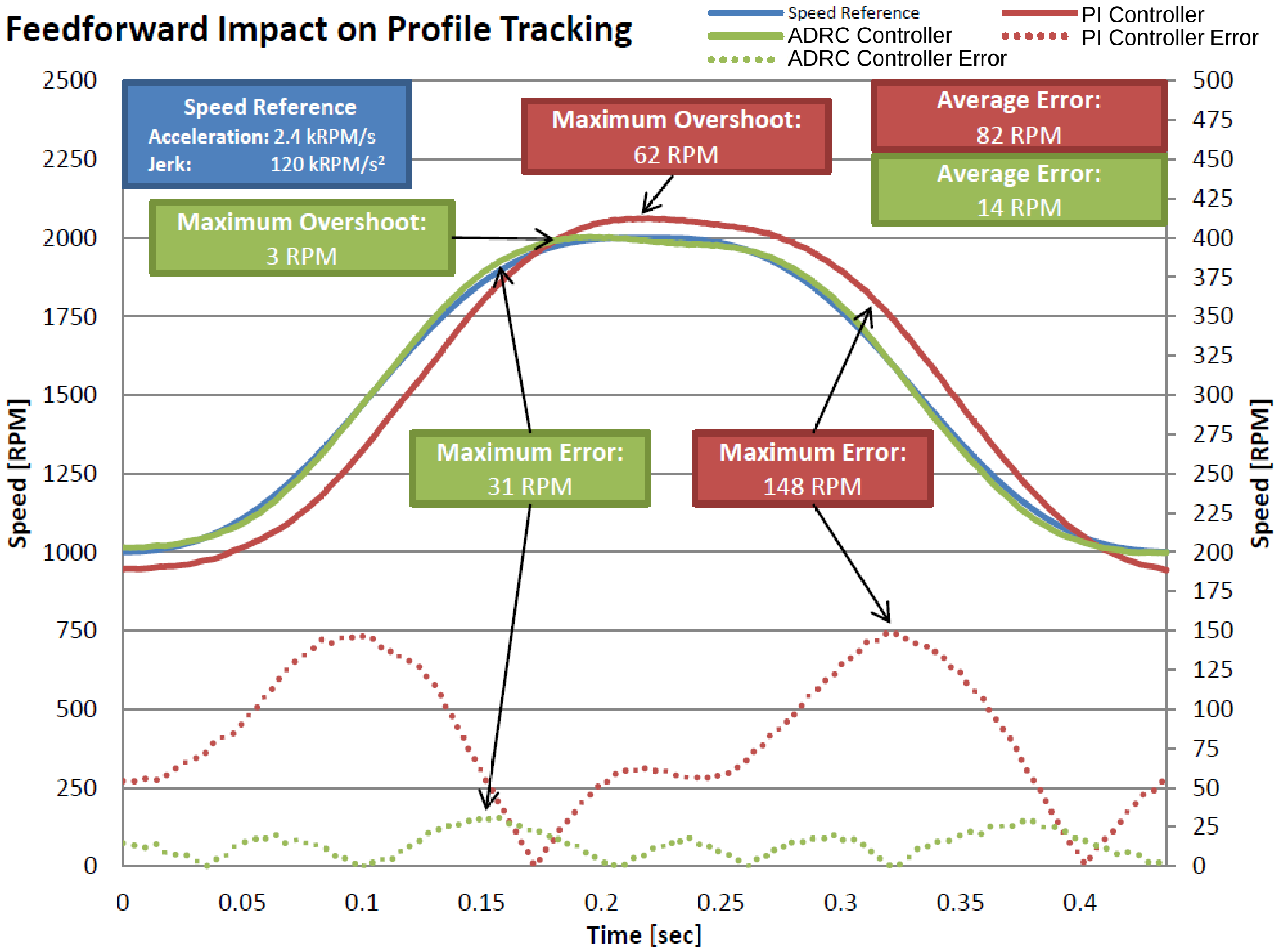
..and used as inputs to speed controller

Profile Generation Curves

- Trapezoid
  - Bounded Acceleration
  - Impulse Jerk
- s-Curve
  - Continuous Acceleration
  - Bounded Jerk
- st-Curve
  - Smooth Acceleration
  - Continuous Jerk

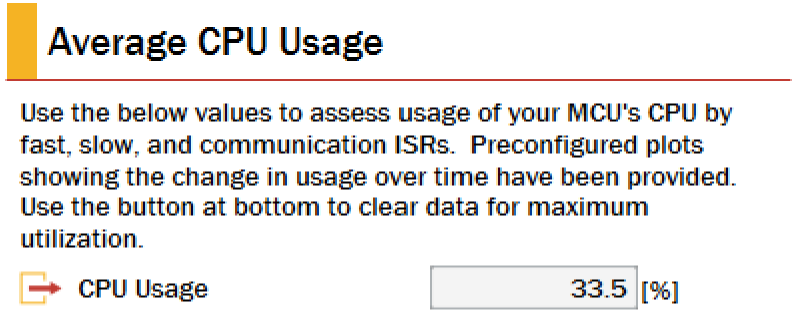


Feedforward



Low CPU Usage

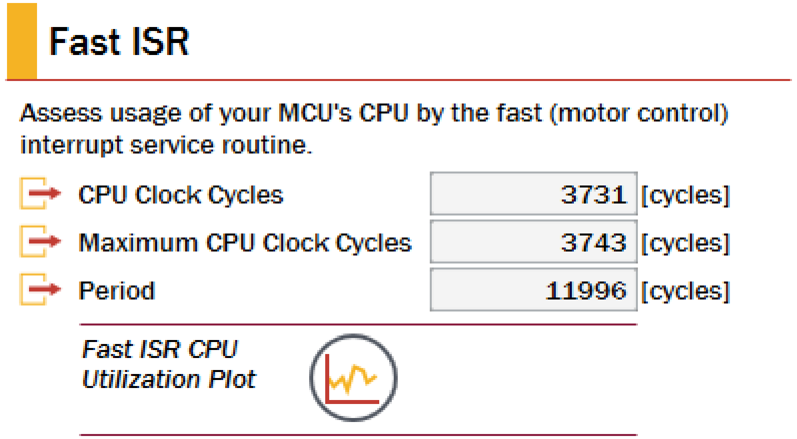
- Sensorless FOC takes about 33% of 120MHz KV3
- 4 ADC Channels
- 6 PWM Channels



Average CPU Usage

Use the below values to assess usage of your MCU's CPU by fast, slow, and communication ISRs. Preconfigured plots showing the change in usage over time have been provided. Use the button at bottom to clear data for maximum utilization.

CPU Usage	33.5	[%]



Fast ISR

Assess usage of your MCU's CPU by the fast (motor control) interrupt service routine.

CPU Clock Cycles	3731	[cycles]
Maximum CPU Clock Cycles	3743	[cycles]
Period	11996	[cycles]

Fast ISR CPU Utilization Plot

Advantages and Disadvantages of FOC

- Advantages
  - Performance
  - Efficiency
  - Low Speed Operation
  - Field Weakening

- Disadvantages
  - ~~PU intensive~~
  - ~~Costly sensor~~
  - ~~Expertise required~~

0.3 Kinetis V Series MCUs for Motor Control

Kinetis V Series MCUs

For Motor Control & Digital Power Conversion

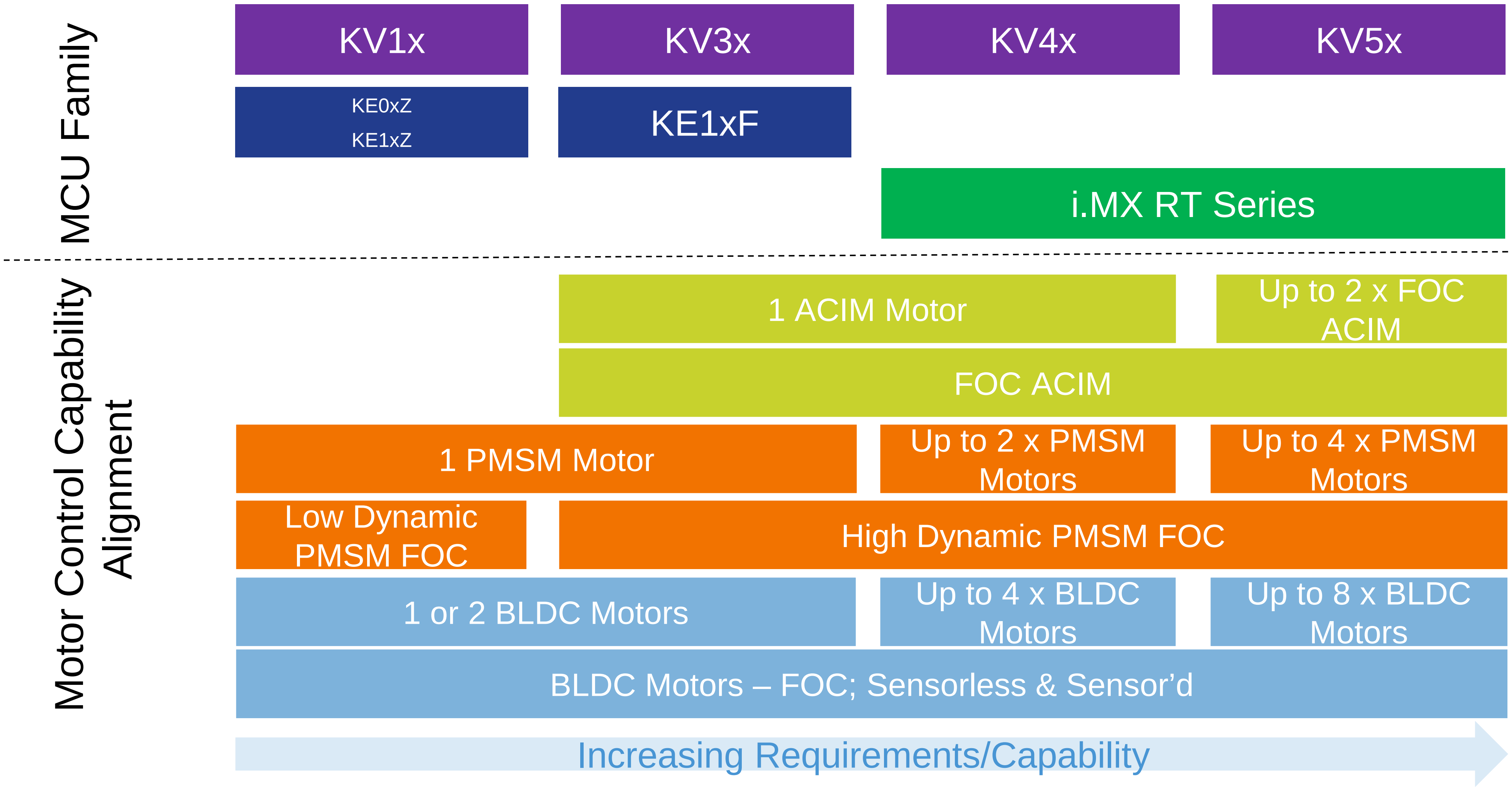
- NXP’s extensive motor and power control expertise and the latest ARM Cortex-M0+, M4 and M7 cores bring secure, connected, high efficiency motor control and power conversion to the mass market

- Efficient, next generation BLDC, PMSM and ACIM designs are enabled by optimized MCU performance and high speed/resolution analog and timing peripherals. High resolution eFlexPWMs support digital power conversion

- Performance and feature scalable MCU families from entry-level 75MHz MCUs, to advanced 240MHz MCUs, maximize hardware & software reuse and end product flexibility

- Enablement including NXP High Voltage and Freedom development boards, Embedded Software Libraries and Kinetis Motor Suite reduce motor control learning curve and speed time to market

Microcontroller and Motor Control Alignment



MCU Family

KV1x

KV3x

KV4x

KV5x

KE0xZ
KE1xZ

KE1xF

i.MX RT Series

Motor Control Capability Alignment

1 ACIM Motor

Up to 2 x FOC ACIM

FOC ACIM

1 PMSM Motor

Up to 2 x PMSM Motors

Up to 4 x PMSM Motors

Low Dynamic PMSM FOC

High Dynamic PMSM FOC

1 or 2 BLDC Motors

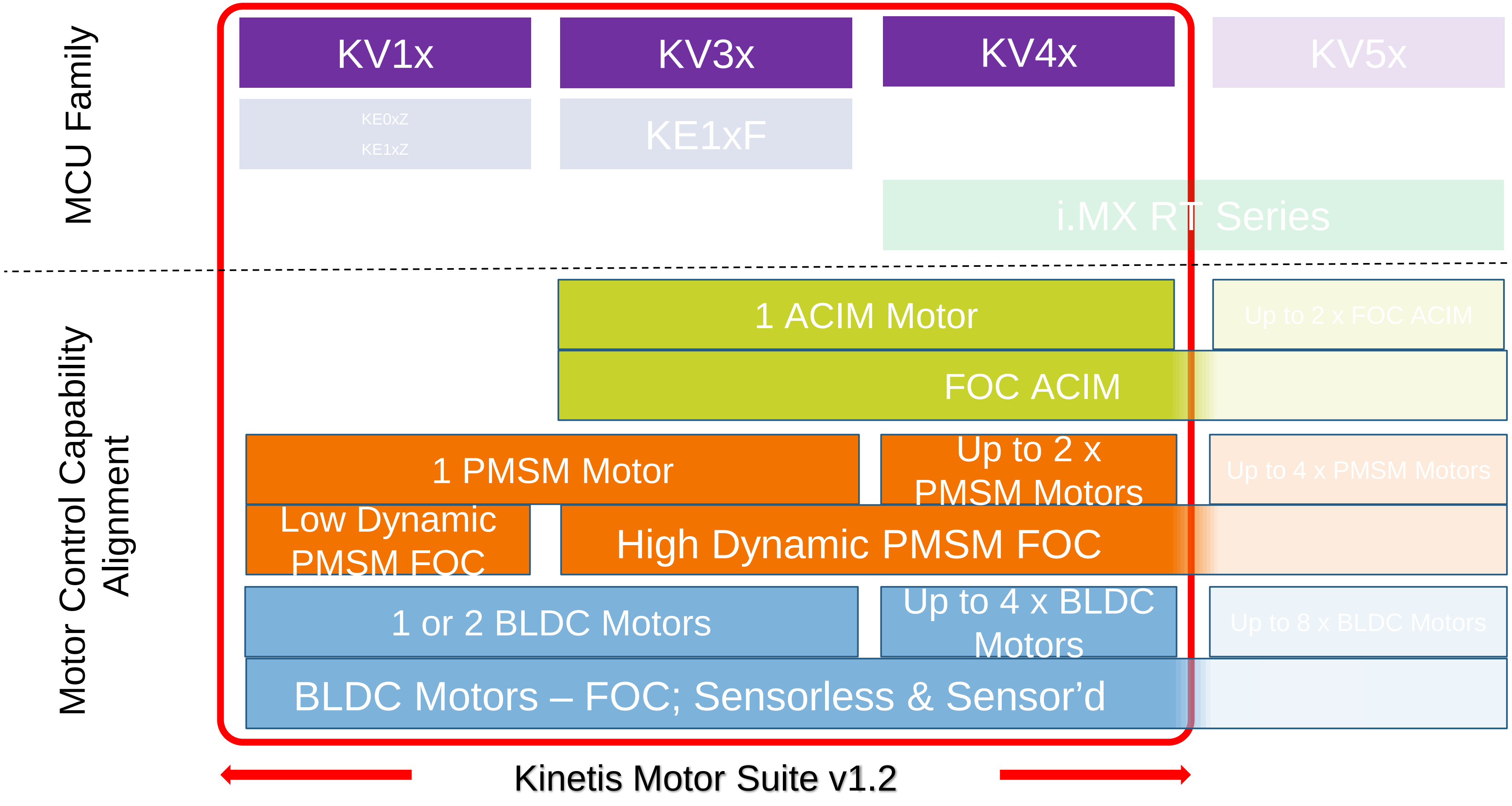
Up to 4 x BLDC Motors

Up to 8 x BLDC Motors

BLDC Motors – FOC; Sensorless & Sensor’d

Increasing Requirements/Capability

Microcontroller and Motor Control Alignment with KMS v1.2 Support



MCU Family

KV1x

KV3x

KV4x

KV5x

KE0xZ
KE1xZ

KE1xF

i.MX RT Series

Motor Control Capability Alignment

1 ACIM Motor

Up to 2 x FOC ACIM

FOC ACIM

1 PMSM Motor

Up to 2 x PMSM Motors

Up to 4 x PMSM Motors

Low Dynamic PMSM FOC

High Dynamic PMSM FOC

1 or 2 BLDC Motors

Up to 4 x BLDC Motors

Up to 8 x BLDC Motors

BLDC Motors – FOC; Sensorless & Sensor'd

Kinetis Motor Suite v1.2

0.4 Lab 1: Spin Your Motor & Use Motor Tuner

Objective:

Quickly use KMS to characterize your motor then operate it across the speed range

LAB 1 – Connect Your Hardware in This Order

1

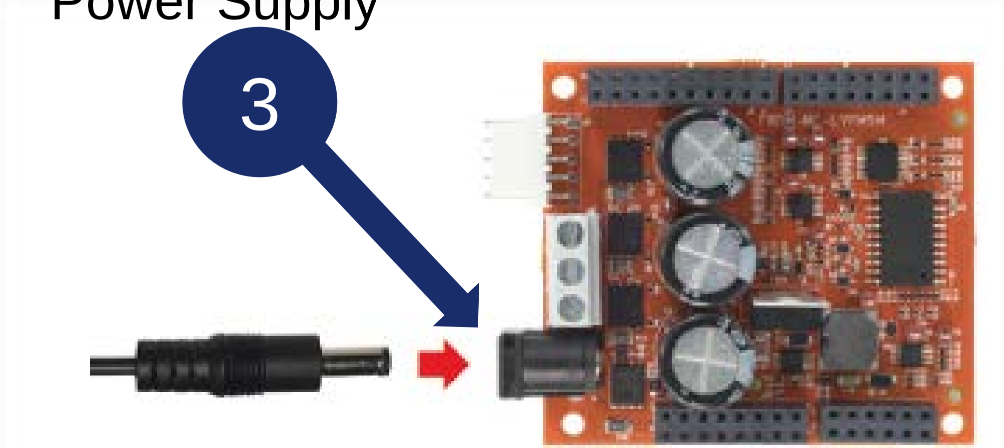
Connect Boards

Connect 3 Phases of motor to board and hall sensor ***(connectors upside down)***

2

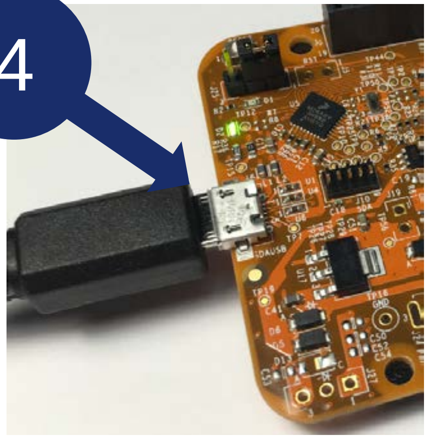
Power Supply

3



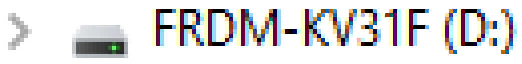
4

OpenSDA Debug



5

Check →Enumerates Virtual MSD



FRDM-KV31F (D:)

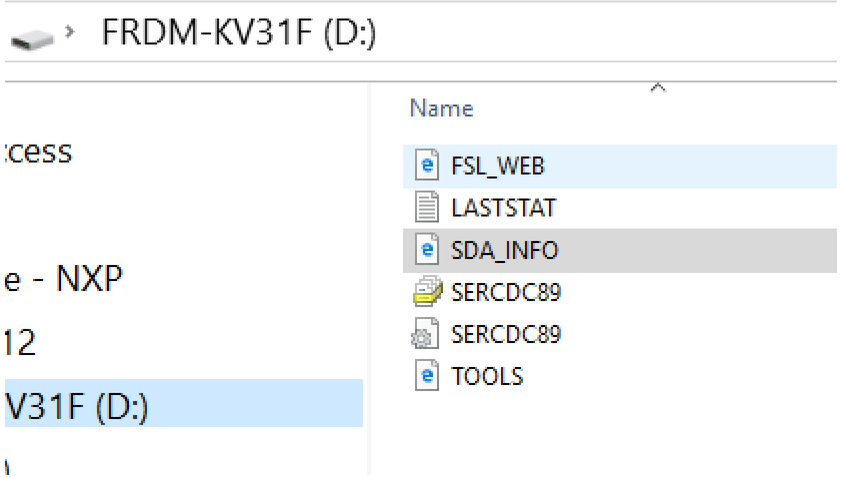
Verify Debug Firmware – Do Not Mass Erase MCU Flash

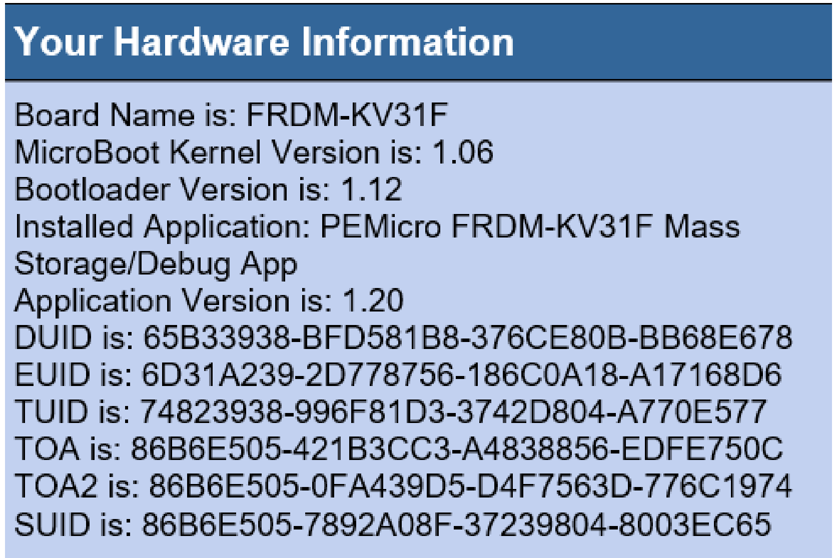
Open the MSD FRDM-KVxx you have and verify it’s the latest firmware. Update as needed.
www.pemicro.com/opensda

FRDM-KV31F - MSD-DEBUG-TWR-KV31F12_Pemicro_v120.SDA

FRDM-KV11Z - MSD-DEBUG-FRDM-KV11Z_Pemicro_v120.SDA

HVP-KV31F - MSD-DEBUG-HVP-KV31F120M_Pemicro_v121.SDA

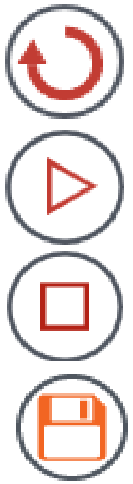




Your Hardware Information

Board Name is: FRDM-KV31F
MicroBoot Kernel Version is: 1.06
Bootloader Version is: 1.12
Installed Application: PEMicro FRDM-KV31F Mass Storage/Debug App
Application Version is: 1.20
DUID is: 65B33938-BFD581B8-376CE80B-BB68E678
EUID is: 6D31A239-2D778756-186C0A18-A17168D6
TUID is: 74823938-996F81D3-3742D804-A770E577
TOA is: 86B6E505-421B3CC3-A4838856-EDFE750C
TOA2 is: 86B6E505-0FA439D5-D4F7563D-776C1974
SUID is: 86B6E505-7892A08F-37239804-8003EC65

Kinetis Motor Suite Basic GUI Elements

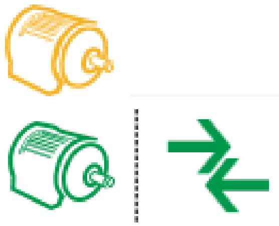


← Start Over

← Play

← Stop

← Save



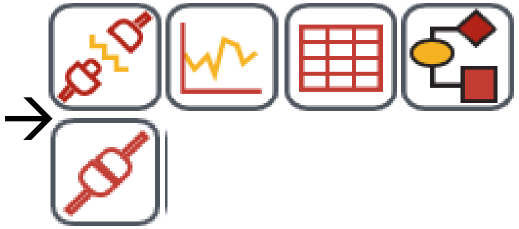
← No Status

← Connected

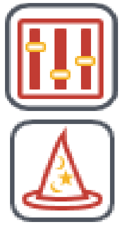
Communication

Yes →

No →



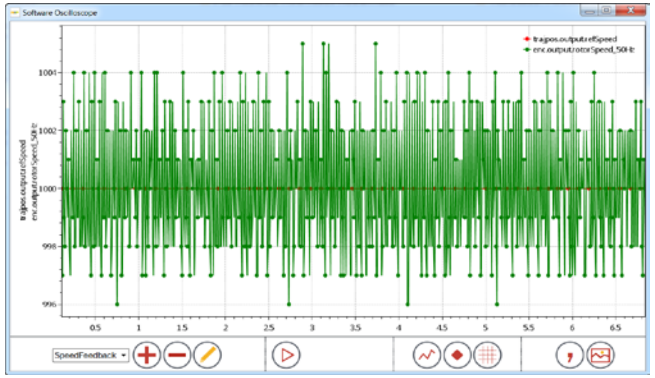
Tuner vs Manager

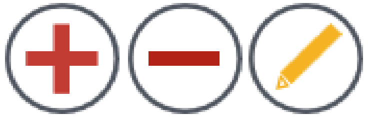


← Motor Tuner

← Motor Manager

Software Oscilloscope

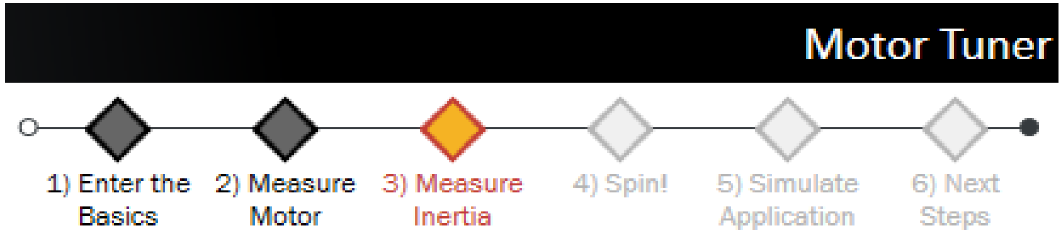


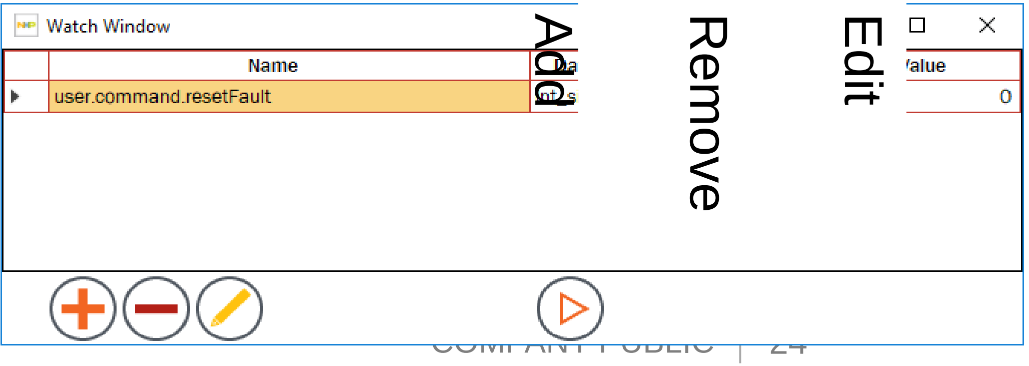


↑ Add

↑ Remove

↑ Edit





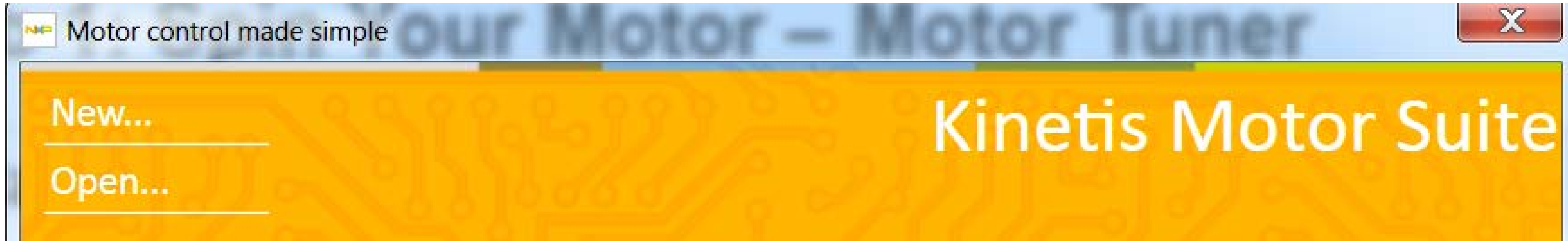
Name	Va	Value
user.command.resetFault	int si	0

Lab 1: Spin Your Motor – Motor Tuner (1)

Start KMS from the desktop icon or start icon. Type Kinetis and start it up.

Select New

1



Motor control made simple

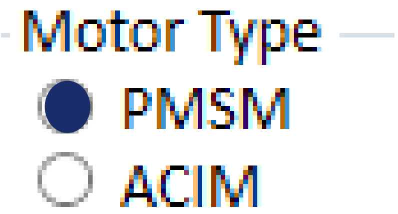
New...

Open...

Kinetis Motor Suite

Select the button next to the Motor Type

2



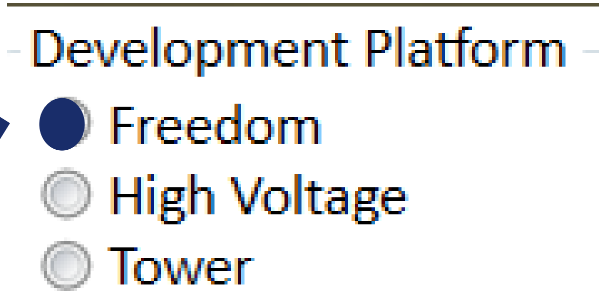
Motor Type

● PMSM

○ ACIM

and evaluation platform

4



Development Platform

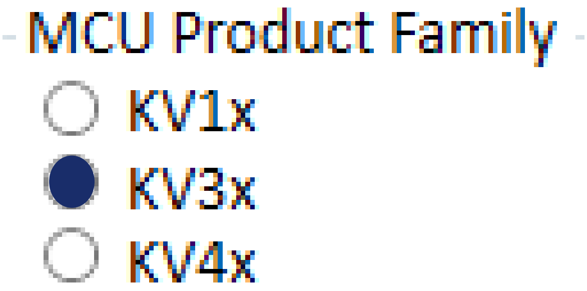
● Freedom

○ High Voltage

○ Tower

Select NXP MCU

3



MCU Product Family

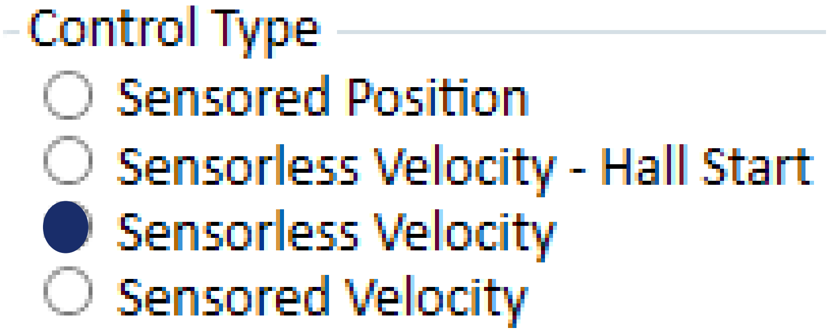
○ KV1x

● KV3x

○ KV4x

and control type

5



Control Type

○ Sensored Position

○ Sensorless Velocity - Hall Start

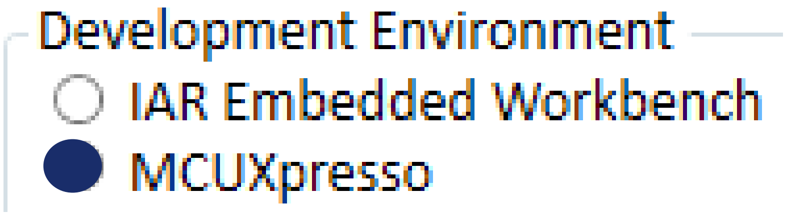
● Sensorless Velocity

○ Sensored Velocity

Lab 1: Spin Your Motor – Motor Tuner (2)

Select the IDE

6



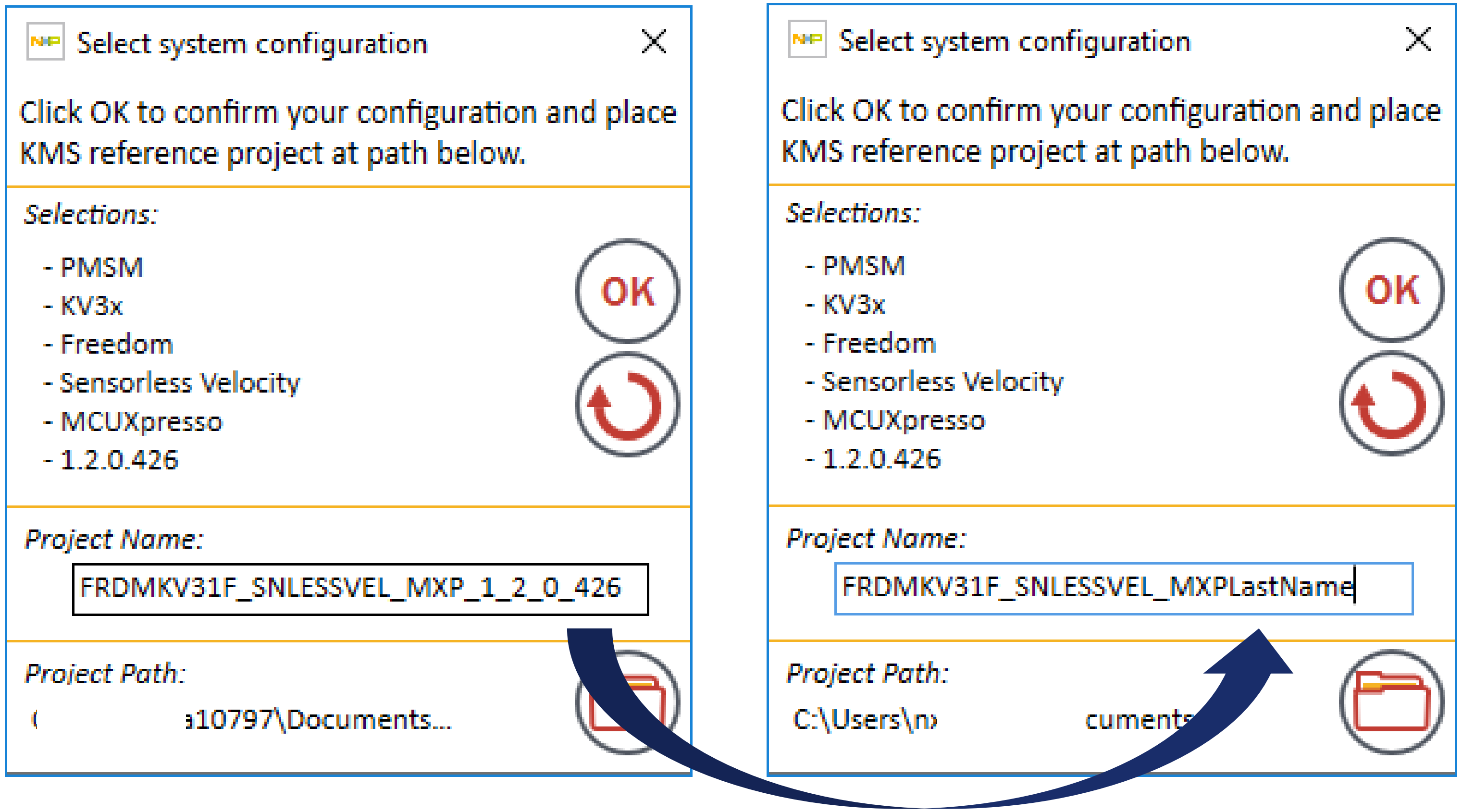
Development Environment

○ IAR Embedded Workbench

● MCUXpresso

Confirm
Change
Project name
(put in your
Last names)

Click OK



Select system configuration

Click OK to confirm your configuration and place KMS reference project at path below.

Selections:

- PMSM
- KV3x
- Freedom
- Sensorless Velocity
- MCUXpresso
- 1.2.0.426

Project Name: FRDMKV31F_SNLESSVEL_MXP_1_2_0_426

Project Path: (a10797\Documents...

Select system configuration

Click OK to confirm your configuration and place KMS reference project at path below.

Selections:

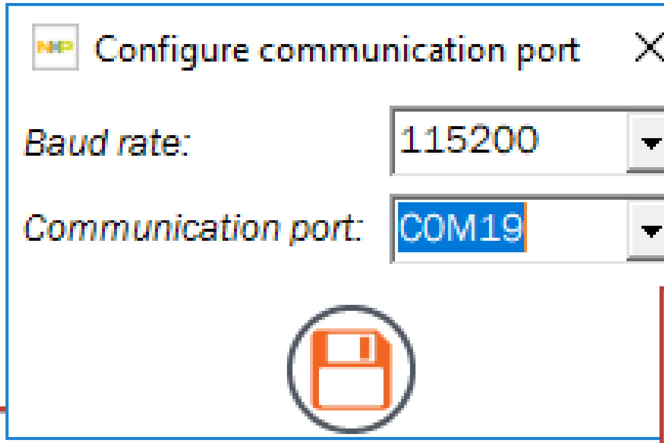
- PMSM
- KV3x
- Freedom
- Sensorless Velocity
- MCUXpresso
- 1.2.0.426

Project Name: FRDMKV31F_SNLESSVEL_MXPLastName

Project Path: C:\Users\n› cuments

Programming the MCU with the Project

Choose the Comm Port

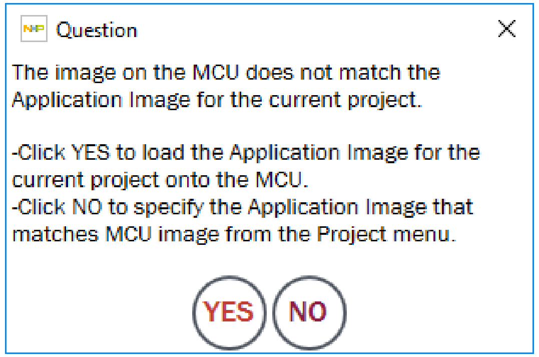


Configure communication port

Baud rate: 115200

Communication port: COM19

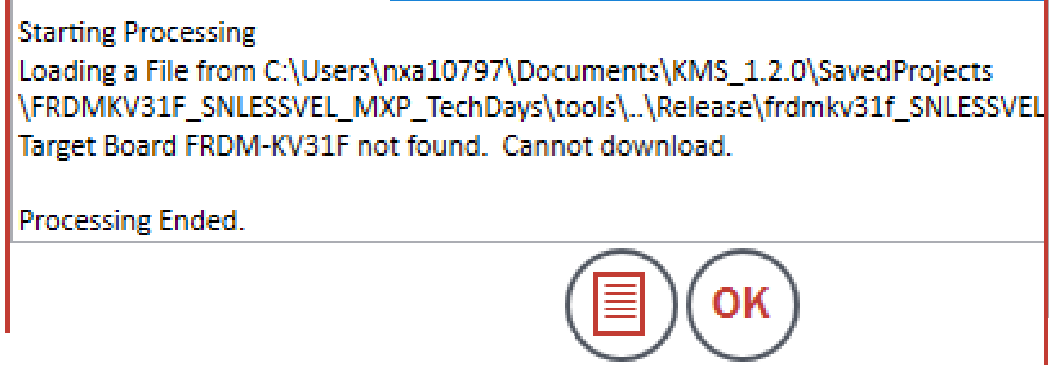
Click →Save



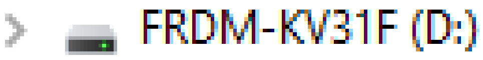
Question

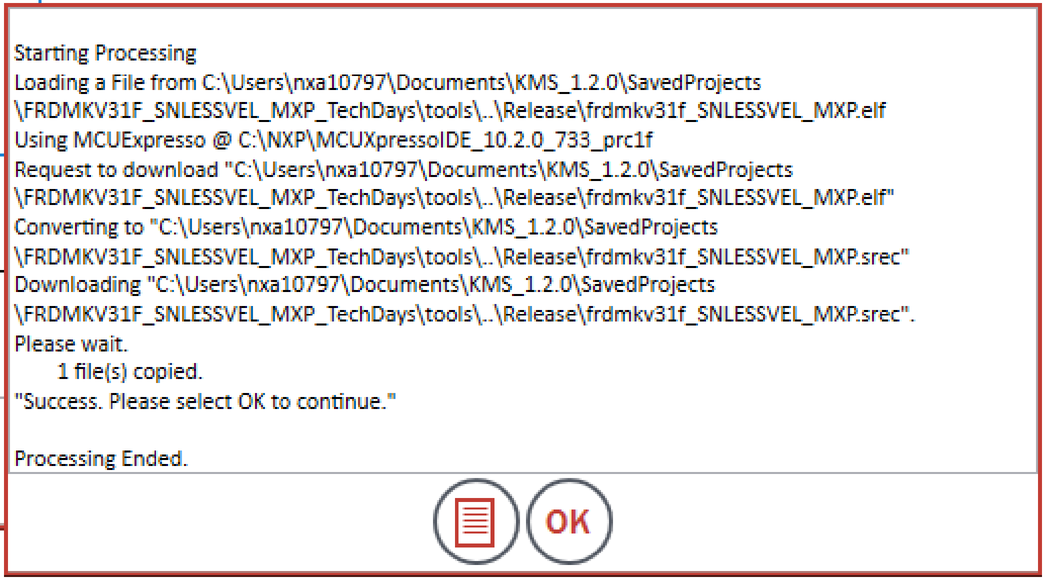
The image on the MCU does not match the Application Image for the current project.

-Click YES to load the Application Image for the current project onto the MCU.
-Click NO to specify the Application Image that matches MCU image from the Project menu.

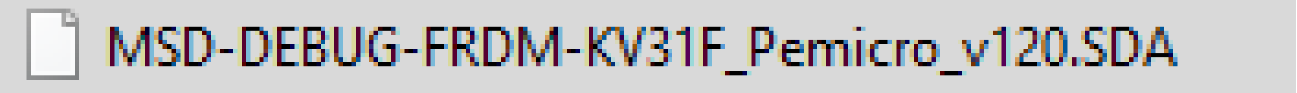


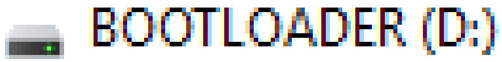
```
Starting Processing
Loading a File from C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects
\FRDMKV31F_SNLESSVEL_MXP_TechDays\tools\..\Release\frdmkv31f_SNLESSVEL
Target Board FRDM-KV31F not found.  Cannot download.

Processing Ended.
```

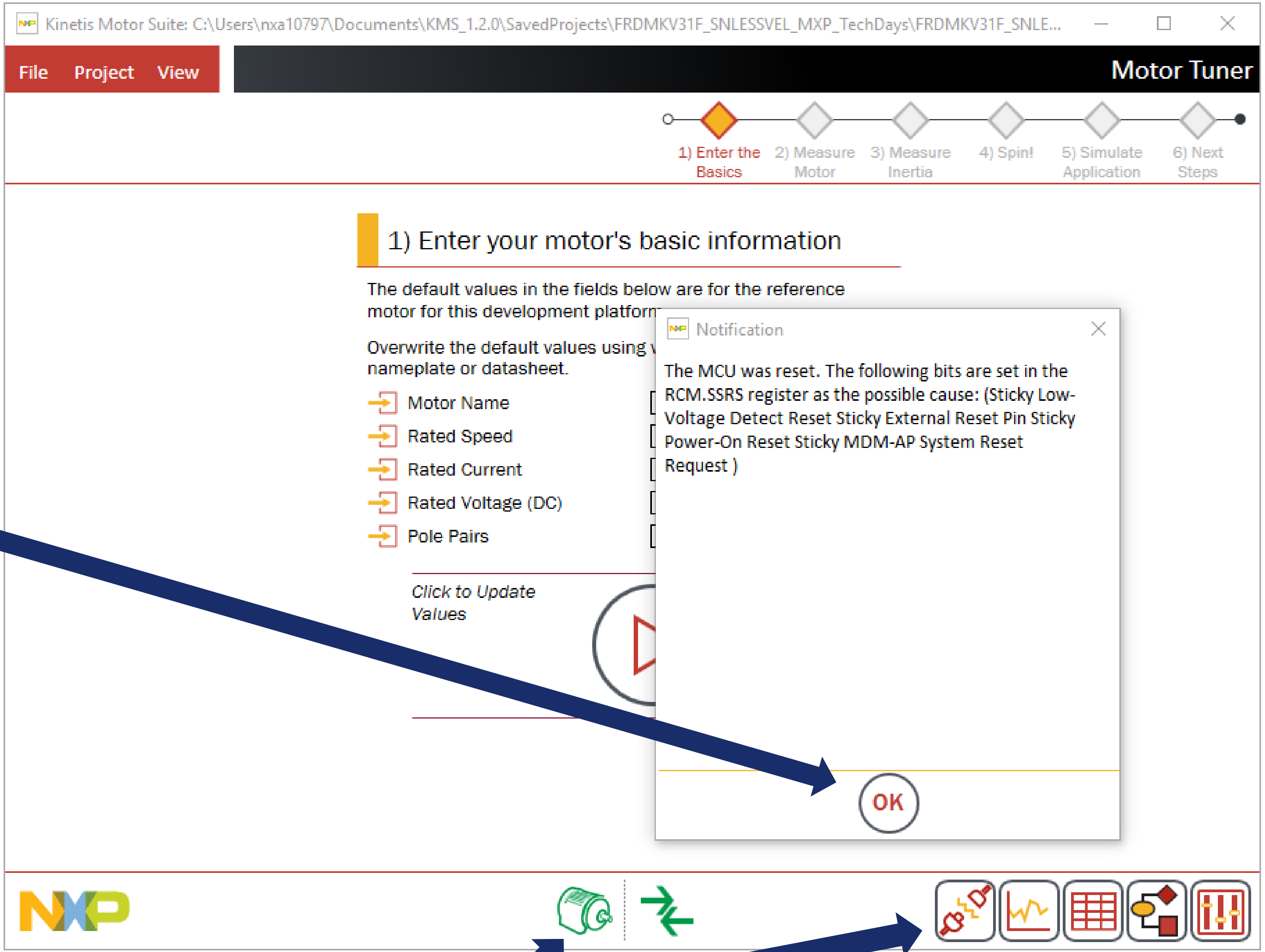




```
Starting Processing
Loading a File from C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects
\FRDMKV31F_SNLESSVEL_MXP_TechDays\tools\..\Release\frdmkv31f_SNLESSVEL_MXP.elf
Using MCUExpresso @ C:\NXP\MCUXpressoIDE_10.2.0_733_prc1f
Request to download "C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects
\FRDMKV31F_SNLESSVEL_MXP_TechDays\tools\..\Release\frdmkv31f_SNLESSVEL_MXP.elf"
Converting to "C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects
\FRDMKV31F_SNLESSVEL_MXP_TechDays\tools\..\Release\frdmkv31f_SNLESSVEL_MXP.srec"
Downloading "C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects
\FRDMKV31F_SNLESSVEL_MXP_TechDays\tools\..\Release\frdmkv31f_SNLESSVEL_MXP.srec".
Please wait.
        1 file(s) copied.
"Success. Please select OK to continue."

Processing Ended.
```







Kinetis Motor Suite: C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects\FRDMKV31F_SNLESSVEL_MXP_TechDays\FRDMKV31F_SNLE...
File
Project
View
Motor Tuner
1) Enter the Basics
2) Measure Motor
3) Measure Inertia
4) Spin!
5) Simulate Application
6) Next Steps

1) Enter your motor's basic information

The default values in the fields below are for the reference motor for this development platform

Overwrite the default values using v nameplate or datasheet.

The MCU was reset. The following bits are set in the RCM.SSRS register as the possible cause: (Sticky Low-Voltage Detect Reset Sticky External Reset Pin Sticky Power-On Reset Sticky MDM-AP System Reset Request)

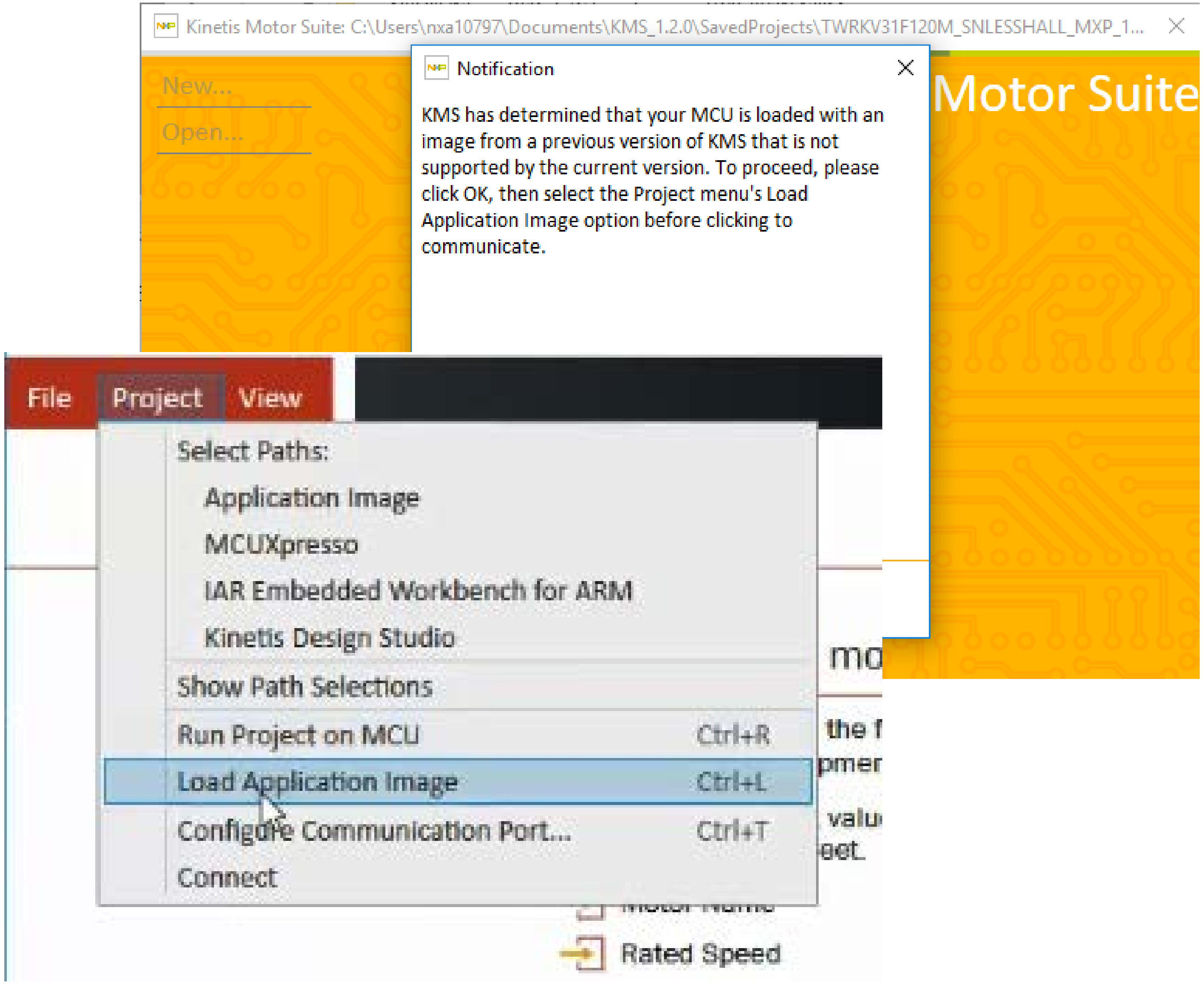
Press OK

Notice Green Motor and Active Communications Status

What to do When…

Follow the instructions given in the prompt

You must load the application image from the project menu!



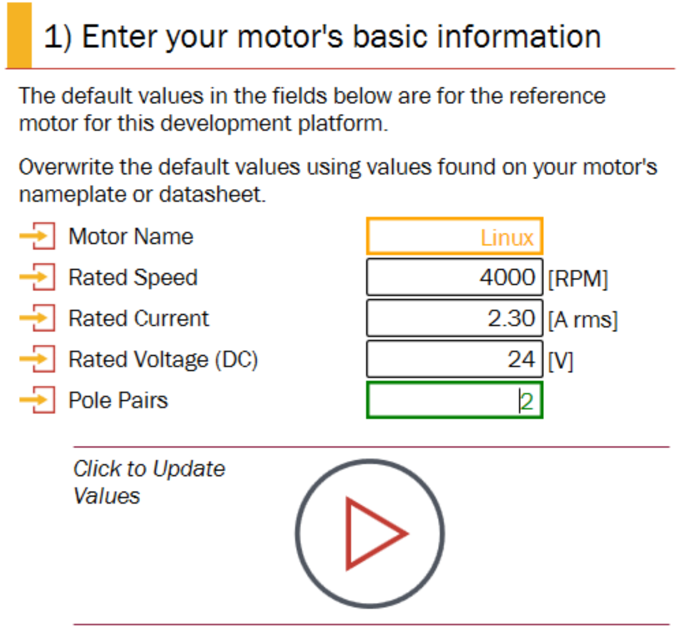
Kinetis Motor Suite: C:\Users\nxa10797\Documents\KMS_1.2.0\SavedProjects\TWRKV31F120M_SNLESSHALL_MXP_1...

Notification

KMS has determined that your MCU is loaded with an image from a previous version of KMS that is not supported by the current version. To proceed, please click OK, then select the Project menu's Load Application Image option before clicking to communicate.

One, Two, Three of Spinning Your Motor

Enter the motor name – rated speed – rated current – Voltage and the number of pole pairs.



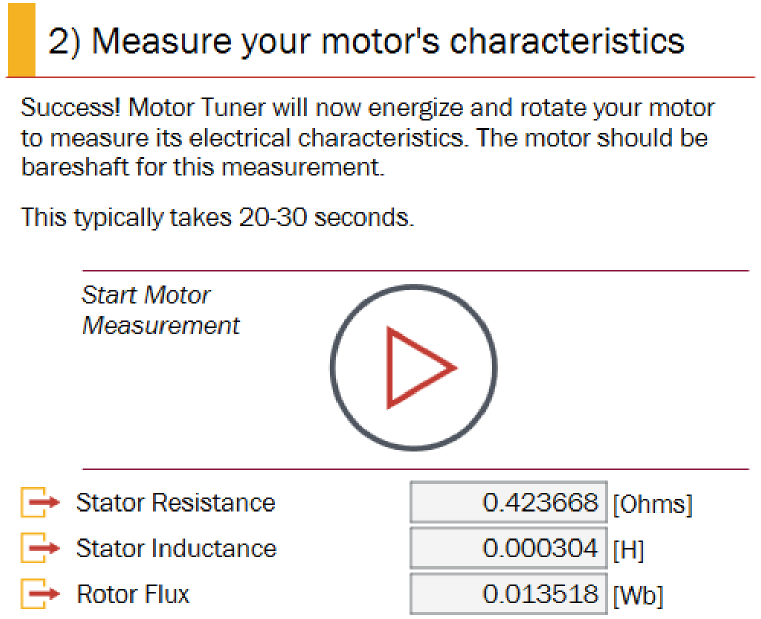
1) Enter your motor's basic information

The default values in the fields below are for the reference motor for this development platform.

Overwrite the default values using values found on your motor's nameplate or datasheet.

Motor Name	Linux	
Rated Speed	4000	[RPM]
Rated Current	2.30	[A rms]
Rated Voltage (DC)	24	[V]
Pole Pairs	2	

Click to Update Values



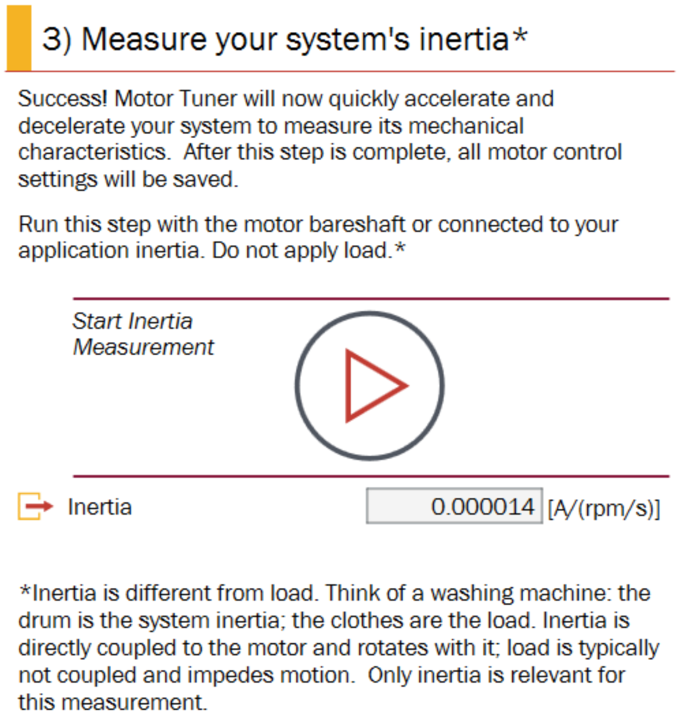
2) Measure your motor's characteristics

Success! Motor Tuner will now energize and rotate your motor to measure its electrical characteristics. The motor should be bareshaft for this measurement.

This typically takes 20-30 seconds.

Start Motor Measurement

Stator Resistance	0.423668	[Ohms]
Stator Inductance	0.000304	[H]
Rotor Flux	0.013518	[Wb]



3) Measure your system's inertia*

Success! Motor Tuner will now quickly accelerate and decelerate your system to measure its mechanical characteristics. After this step is complete, all motor control settings will be saved.

Run this step with the motor bareshaft or connected to your application inertia. Do not apply load.*

Start Inertia Measurement

Inertia	0.000014	[A/(rpm/s)]

*Inertia is different from load. Think of a washing machine: the drum is the system inertia; the clothes are the load. Inertia is directly coupled to the motor and rotates with it; load is typically not coupled and impedes motion. Only inertia is relevant for this measurement.

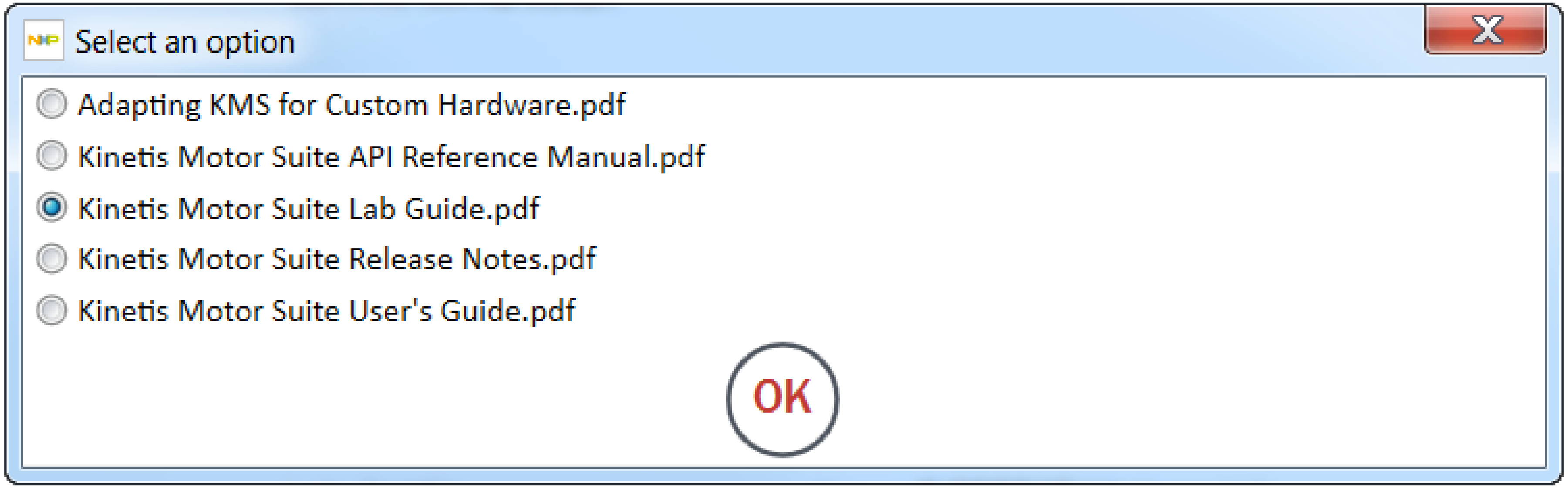
Select Play Button

Select Play Button

Select Play Button

Adapting KMS for Your Hardware

- To open the document ‘Kinetis Motor Suite Lab Guide’
- Within KMS: Select View → Documents



Select an option

○ Adapting KMS for Custom Hardware.pdf
○ Kinetis Motor Suite API Reference Manual.pdf
● Kinetis Motor Suite Lab Guide.pdf
○ Kinetis Motor Suite Release Notes.pdf
○ Kinetis Motor Suite User's Guide.pdf

- Use the bookmarks to go to page 2

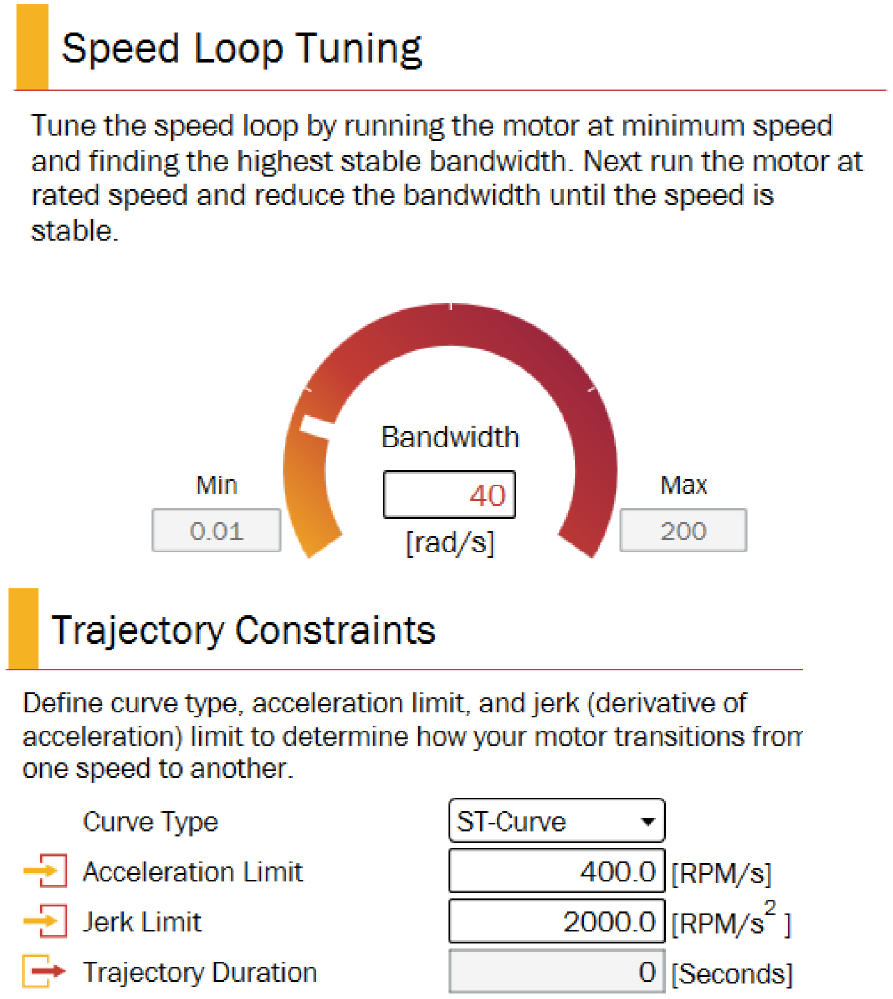
0.4 LAB 3 & 4: Use Motor Manager

Objective:

Quickly use KMS to characterize your motor then operate it across the speed range

Lab 1: Continued – Spin Your Motor with Motor Manager (1)

You will Learn how to use the Motor Manager to tune with the KMS tools



Speed Loop Tuning

Tune the speed loop by running the motor at minimum speed and finding the highest stable bandwidth. Next run the motor at rated speed and reduce the bandwidth until the speed is stable.

Bandwidth: 40 [rad/s]

Min: 0.01

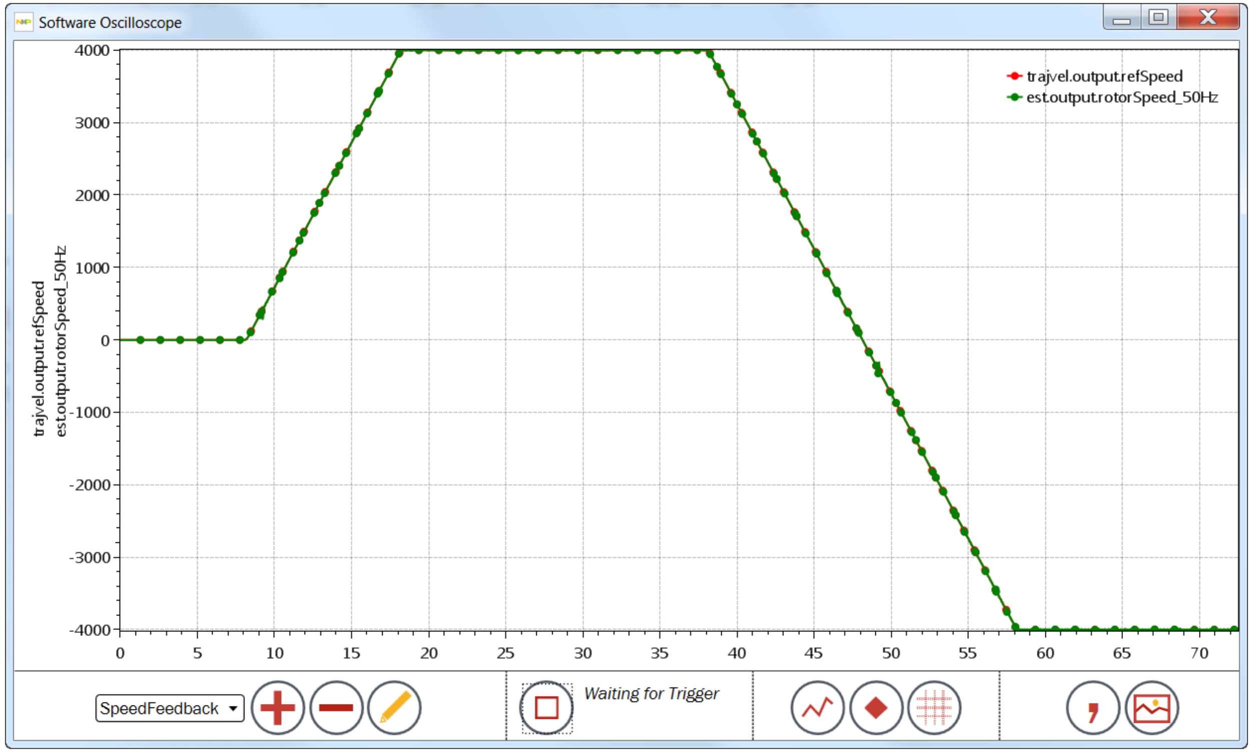
Max: 200

Trajectory Constraints

Define curve type, acceleration limit, and jerk (derivative of acceleration) limit to determine how your motor transitions from one speed to another.

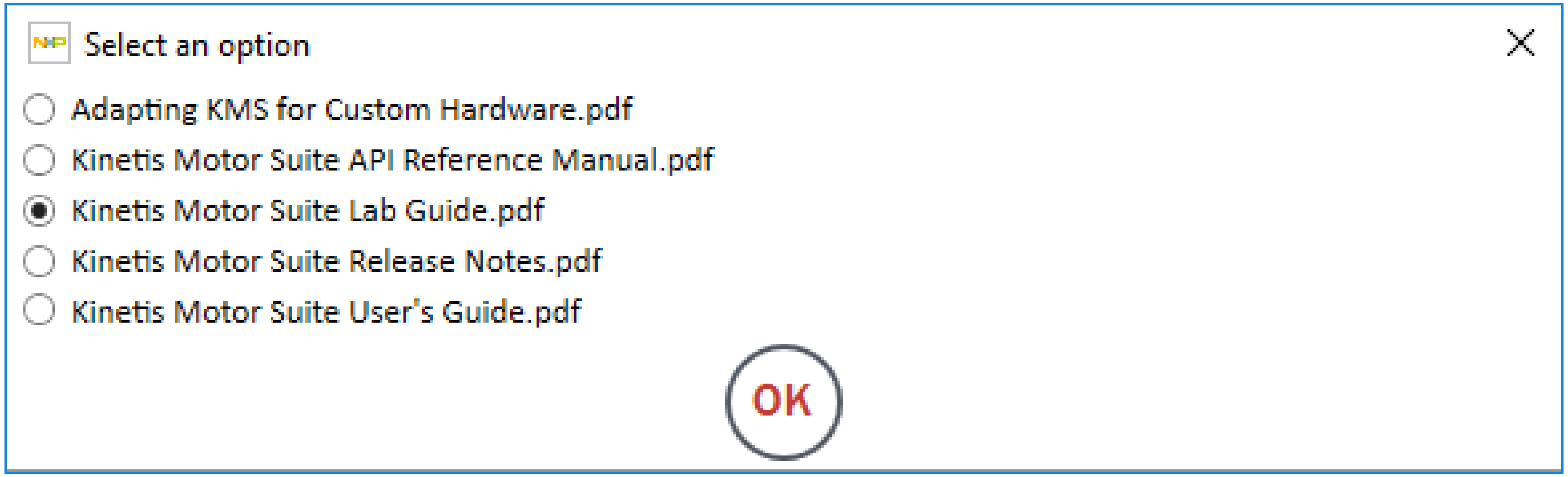
Curve Type	ST-Curve	
Acceleration Limit	400.0	[RPM/s]
Jerk Limit	2000.0	[RPM/s^2]
Trajectory Duration	0	[Seconds]

Software oscilloscope



Lab 1: Continued – Spin Your Motor with Motor Manager (2)

- Open the Document ‘Kinetis Motor Suite Lab Guide’
- From KMS: Select view→ Documents



Select an option

○ Adapting KMS for Custom Hardware.pdf

○ Kinetis Motor Suite API Reference Manual.pdf

● Kinetis Motor Suite Lab Guide.pdf

○ Kinetis Motor Suite Release Notes.pdf

○ Kinetis Motor Suite User's Guide.pdf

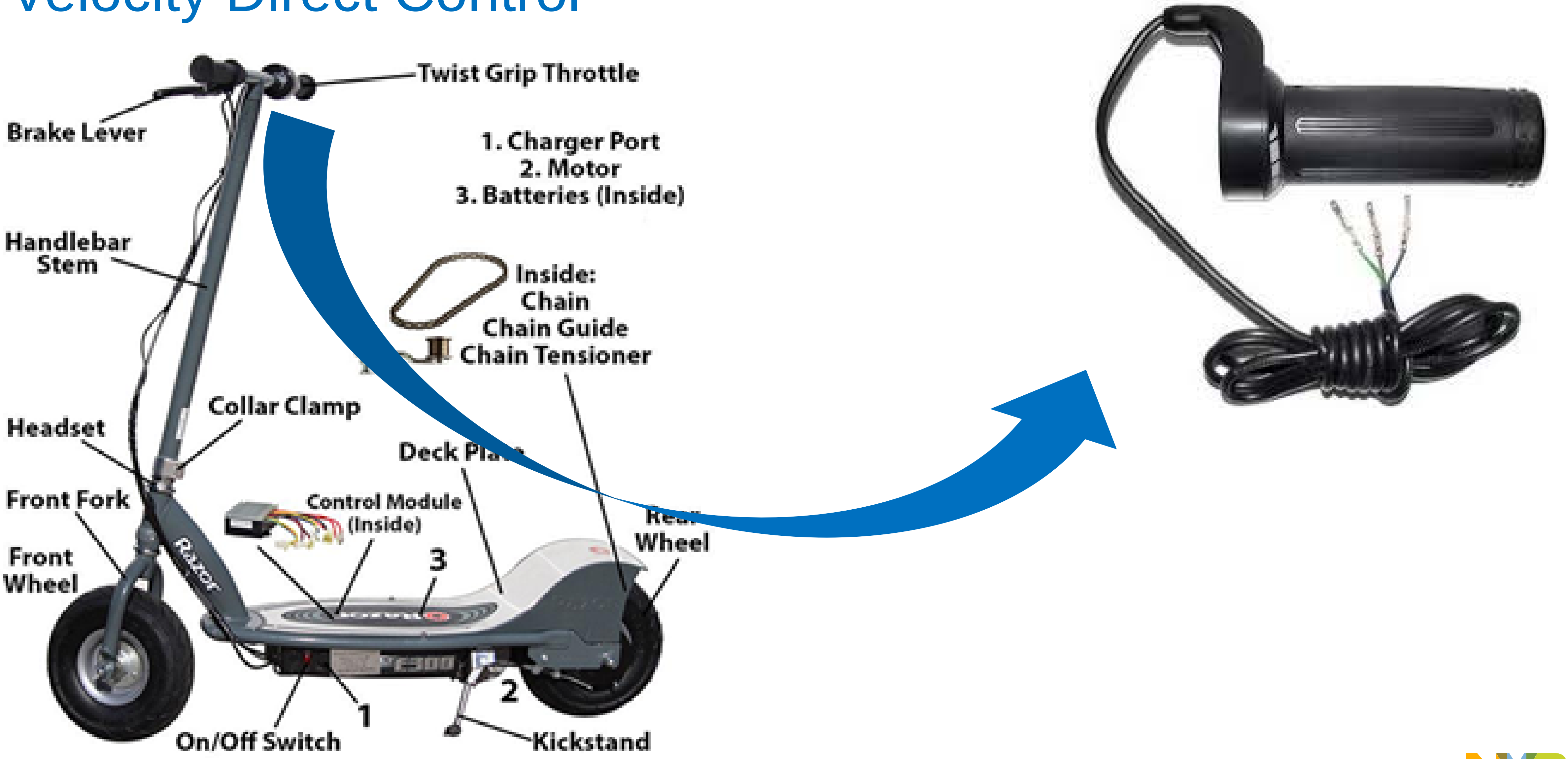
- Use the bookmarks to navigate to page 113: → Do Labs 3 and 4
  3.1.3 Velocity lab 3: tune your motor with Motor Manager
  3.1.4 Velocity lab 4: test trajectories

0.4 Lab 5: Adapting KMS for Your Hardware

Objective:

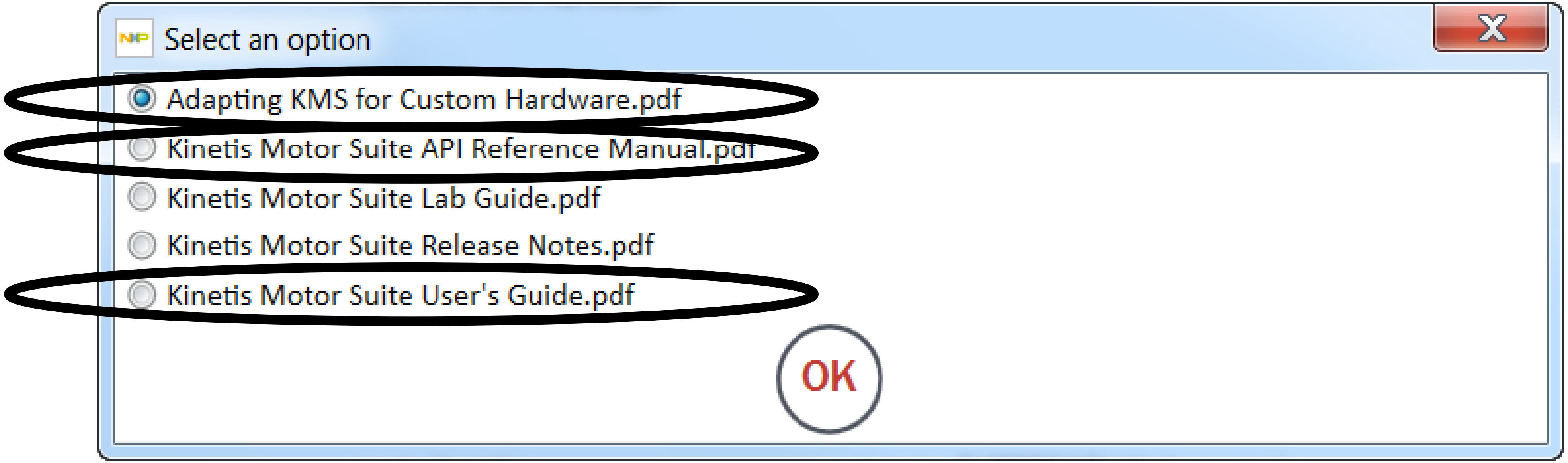
Explore uses of KMS with your Hardware

Velocity Direct Control



Adapting KMS for Your Hardware

- Open the Document ‘Adapting KMS for Custom Hardware’
- And the Document ‘Kinetis Motor Suite API Reference Manual’
- Within KMS: Select View → Documents



● Adapting KMS for Custom Hardware.pdf
○ Kinetis Motor Suite API Reference Manual.pdf
○ Kinetis Motor Suite Lab Guide.pdf
○ Kinetis Motor Suite Release Notes.pdf
○ Kinetis Motor Suite User's Guide.pdf

- Adapting HW - Use the bookmarks to go to page 2:
- In API RM – go to chapter 2 - User

Custom Hardware Resources Needed for KMS

- The KMS User’s Guide has a list of the resources used by the project and the tools. Open the Kinetis Motor Suite Users Guide and go to chapter 11 page 191.
- Execution cycles and clock speed
- Flash → KMS pre-programmed code in the top 8K of the Flash.
  + (50472 to 58804 using MCUXpresso) or + (43,224 to 48456 using IAR)
- RAM less than 8K
- Peripherals→ FAC, ADC0, ADC1, PDB0, FlexTimer0 & 1, UART0, GPIO for Hall Sensor. If the FET pre-driver is used (TWR-MC-LV3PH) then add SPI0, more GPIO
- KMS GUI interacts with UART0 and a RDA client in the reference project. If you don’t use UART0 you will need to reduce the com baud rate.
- IDE uses Debug interface for programming and debug.

Code Review – USER_States for Speed Control

- Idle
- Fault
- Self-commissioning (SCM)
- Inertia Estimation (Inertia)
- PWM Duty Control
- Voltage Control
- Current Control (Current)
- Speed Control (Speed)
- Position Control (Position) [Sensored Position]
- Motion Sequence (Plan)
- Braking (Brake)
- Encoder Alignment (Align) [Sensored Velocity or Sensored Position]

Where to Put Your Control Code?

In the main loop, the code snippet below checks the state of the DC bus and the Fault indicator.

```
if (DRV_getIsFaultActive()) // Pulled Up Switch is Active Low
        {
            FAULT_LEDON;
        }
        else
        {
            FAULT_LEDOFF;
        }
```

/* insert control code here */

OR

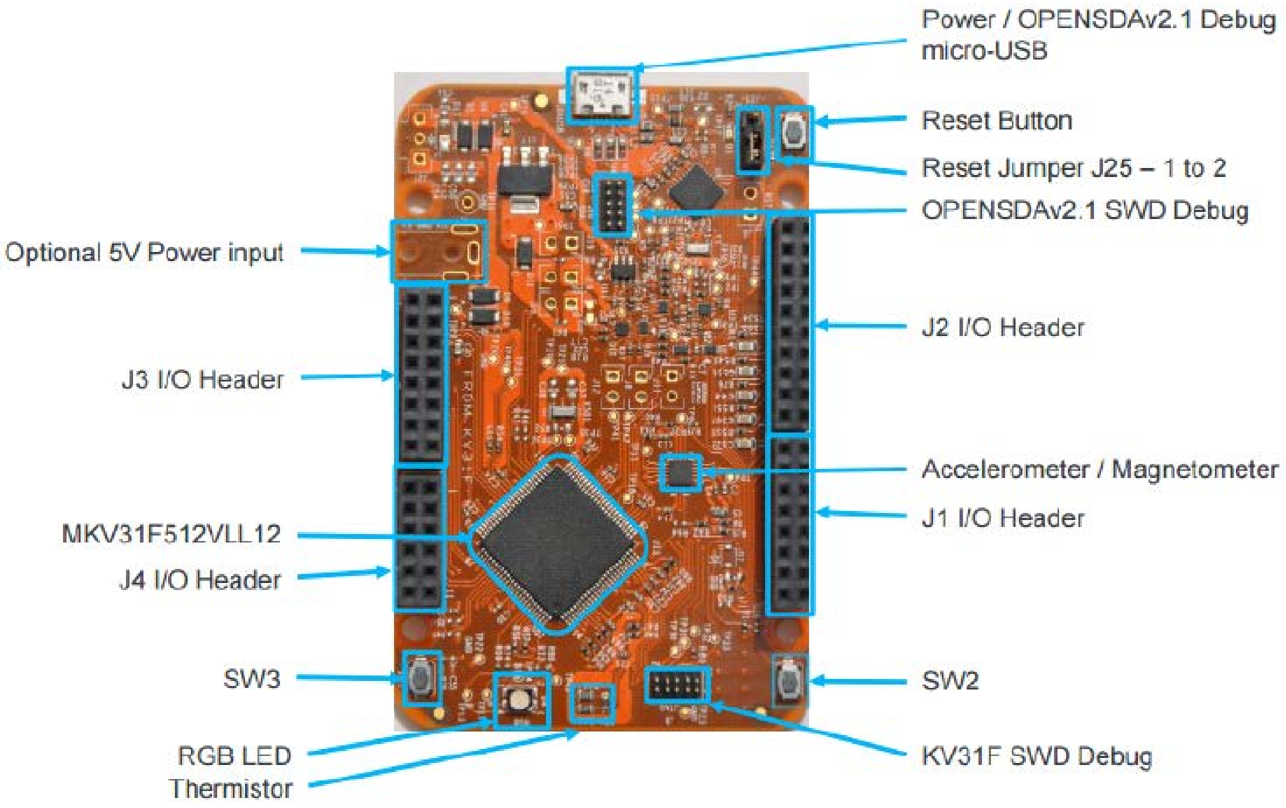
the interrupt service routine that executes at the fast or slow tick rate

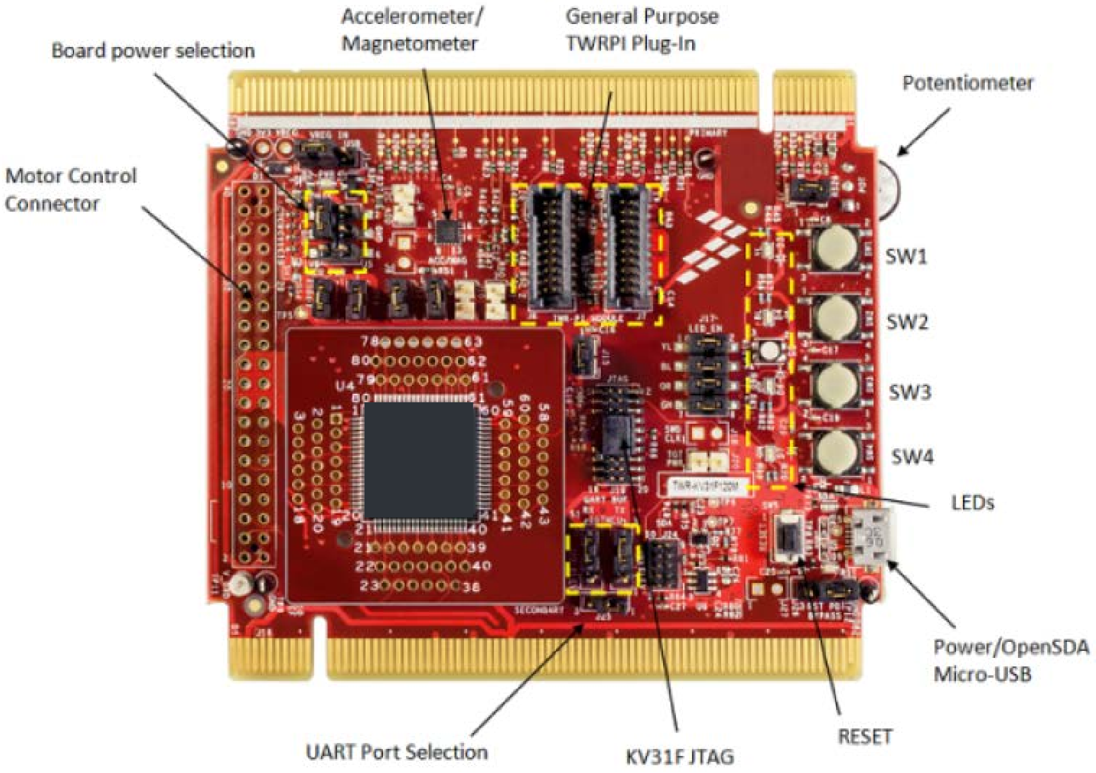
Steps to Adapt to Custom Hardware

Step	Description	KMS Component(s)
1	Spin your motor on evaluation hardware	GUI - Motor Tuner
2	Define ADC settings	GUI - Motor Manager
3	Define PWM settings	GUI - Motor Manager
4	Adjust additional settings	GUI - Motor Manager
5	Convert reference project	Motor Observer
6	Specify pins	Motor Observer
7	Compile and download firmware	Motor Observer
8	Verify DC bus and offsets	Motor Observer
9	Verify PWM operation	GUI - Motor Manager • GUI - Software Oscilloscope
10	Verify phase currents	GUI - Motor Manager • GUI - Software Oscilloscope
11	Validate that motor spins on custom hardware	GUI - Motor Tuner

Lab 3: Connect Your Motor to the Real World

- Using the same Document ‘Kinetis Motor Suite Lab Guide’
- Use the bookmarks to navigate o page 194:
  3.1.7 Velocity lab 7: connect your motor to the real world → Do Labs 7 & 8





Hardware Setup – Using MCUXpresso Pins Tools

- Set up three new pins on FDRM-KV31 J4, pins 2, 4 and 6
  - ADC input from a potentiometer
  - Drive potentiometer with VDD
  - Drive potentiometer with GND
- Open MCUXpresso IDE and import the KMS velocity speed project
- Open the pins Tool from MCUXpresso IDE
- Assign pin J4-2 (PTC8) as a GPIO output – Identifier → POTVDD
- Assign pin J4-6 (PTB11) as a GPIO output – Identifier → POTGND
- Assign pin J4-4 (PTC9) to be an ADC1 input 5b ADC1.SE5b Identifier → POT-IN
- Update Project Code
- Return to IDE project menu

Custom Electronic Motor Control LVPMSM Features

- MKV3xFxxx 48 pin LQFP MCU with FPU and 100-120MHz, 64-512 KB flash memory, 16-64 KB RAM
- DC Power Supply Input voltage DC: 24-48VDC
- Output current up to 30 amps RMS
- Power supply reverse polarity protection circuitry
- 3-phase bridge inverter (6-MOSFET's)
- 3-phase MOSFET gate driver with over current and under voltage protection
- Analog sensing (DC bus voltage, DC bus current, 3-phase back-EMF voltage)
- 5.5 VDC auxiliary power supply for battery elimination
  - **Optional Control Interface Features:**
- Asynchronous Serial Interface (UART) for command interface and firmware update
- PWM speed control input.
- Motor speed/position sensors interface (Encoder, Hall)
- CAN bus interface functionality

0.4 LAB 6: Connect Your Motor to the Real World & Standalone

Hardware Setup – Editor in MCUXpresso

Open file kms_hw.c and add code

```
/* Potentiometer config */
const gpio_pin_config_t pot_config =
{
    kGPIO_DigitalOutput,  /* Set current pin as digital output */
    (uint8_t)1U           /* Set default logic low */
};
    /* Enable port for potentiometer */
    GPIO_PinInit(BOARD_POTVDD_GPIO, BOARD_POTVDD_GPIO_PIN, &pot_config);
    GPIO_PinInit(BOARD_POTGND_GPIO, BOARD_POTGND_GPIO_PIN, &pot_config);
```

Open File kms_hw.h and add defines

```
/* Functions mapping to Potentiometer */
#define POT_VDD_0  GPIO_WritePinOutput(BOARD_POTVDD_GPIO, BOARD_POTVDD_GPIO_PIN, 0)
#define POT_VDD_1 GPIO_WritePinOutput(BOARD_POTVDD_GPIO, BOARD_POTVDD_GPIO_PIN, 1)

#define POT_GND_0  GPIO_WritePinOutput(BOARD_POTGND_GPIO, BOARD_POTGND_GPIO_PIN, 0)
#define POT_GND_1 GPIO_WritePinOutput(BOARD_POTGND_GPIO, BOARD_POTGND_GPIO_PIN, 1)
```

Sampling Additional ADC Channels

Adapting KMS for Custom Hardware AN5254

- The Motor Control ADC channels defined in the header file kms_hw.h
- If you want to sample two channels (23 and 24) on ADC 0 and one channel (12) on ADC 1 and sample channel 23 on ADC 0 twice as fast as channel 24.
- This can be easily configured using the code
  - change in main.c starting around line 194 of FRDMKV31F_SNLESSVEL_MXP/src/main.c

```
#define NUMBER_USER_ADC_CHANNELS (1)
static const bool bEnableRoundRobinAdc = true;
static const uint16_t adcUserChannelsChannel0[NUMBER_USER_ADC_CHANNELS] = {5};
static const uint16_t adcUserChannelsChannel1[NUMBER_USER_ADC_CHANNELS] = {5};
```

Sampling Additional ADC Channels – See Community Doc

In main.c add these variable declarations.

```
/* declare these as global variables for main.c */
_lq adcSpeed;
_lq adcSpeedAccum;
_lq adcSpeedAvg;
uint16_t speedUpdateCounter;
static uint8_t adcMux = 0;
extern USER_t user;
```

In main.c, after Board_InitGpio(); call add

```
/* Set potentiometer bias voltage so */
    POT_VDD_1;
    POT_GND_0;
```

Commanding a Motor Speed Toque or Position in a KMS reference project

https://community.nxp.com/docs/DOC-340364

Converting ADC Input to Speed Control Input – Add in Main

```
/* this code takes the ADC reading, downshifts it by 4 to remove noise
*/and uses that as a percentage of the maximum application speed (in this case 4krpm)*/

adcSpeed = _LQmpyLQX((adc1Results[0] >> 4), 8, _LQ(4000.0/FULL_SCALE_SPEED_RPM), 24);
adcSpeed = _LQsat(adcSpeed, _LQ(1.0), _LQ(0.0));
/* this code averages the adc reading over 125 samples before setting it to user.command.targetSpeed
 * it will also handle setting the control mode if the commanded speed is larger than 0 */
if(adcSpeed > _LQ(0.0)) {
                        user.state= USER_RUN_SPEED;
                        if(speedUpdateCounter >= 125)
                        {    speedUpdateCounter = 0;
                             adcSpeedAvg=(adcSpeedAccum/125)&0x00FF0000;
                             user.command.targetSpeed = adcSpeedAvg;
                             adcSpeedAccum = 0;
                        } else {
                             adcSpeedAccum = adcSpeed + adcSpeedAccum;
                             speedUpdateCounter++;
                        }
        } else {
                user.state= USER_IDLE;
                adcSpeedAccum = 0;
                adcSpeedAvg = 0;
                speedUpdateCounter = 0;
        }
```

0.5 Enablement

Low Cost Reference Platforms: Hardware + Software

BLDC & PMSM

Software Platform’s Supported

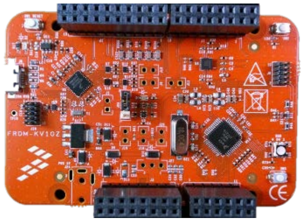
Kinetis Motor Suite

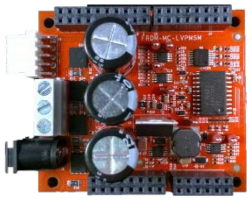
Kinetis Reference Design

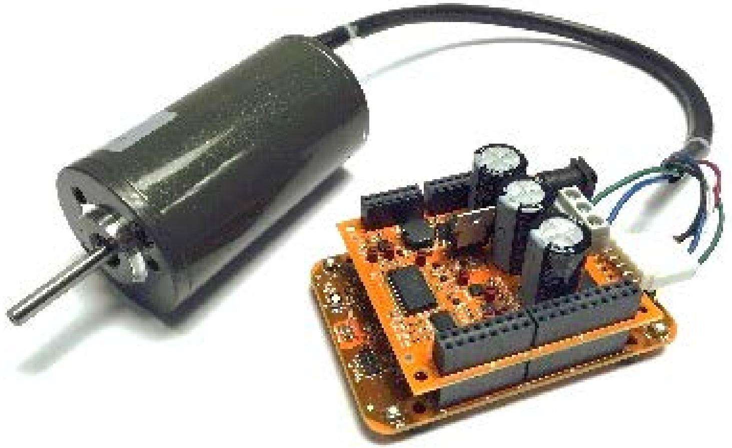
Motor Control Toolbox

Complete Reference Design

From $85







FRDM-KV31F: $20

FRDM-KV11Z: $20

FRDM-MC-LVPMSM: $50

- PMSM (Sinusoidal) control
- 24V, 5Amp, 120W

FRDM-MC-LVBLDC: $30

- BLDC (Trapezoidal) control
- 12V, 5Amp, 60W

FRDM-MC-LVMTR: $35

- 4000RPM, 90W

Or Your Motor

Custom Electronic Motor Control LVPMSM Features

FRDM-KV31F features

- MKV31F512 MCU with FPU and 120MHz, 512 KB flash memory, 64 KB RAM
- OpenSDA debug interface
- FXOS8700CQ Accelerometer
- RGB LED
- Thermistor
- Arduino header
- Freedom motor control header compatible with FRDM-LVBLDC and FRDM-LVPMSM shields
- Push buttons for manual control Please use correct number for sub-section title

FRDM-MC-LVPMSM features

- Power Supply Input voltage DC: 24-48VDC
- Output current up to 5 amps RMS
- Power supply reverse polarity protection circuitry
- 3-phase bridge inverter (6-MOSFET’s)
- 3-phase MOSFET gate driver with over current and under voltage protection
- Analog sensing (DC bus voltage, DC bus current, 3-phase back-EMF voltage)
- 5.5 VDC auxiliary power supply providing FRDM MCU board supplying
- Motor speed/position sensors interface (Encoder, Hall)

High Voltage Reference Platform: Hardware + Software

ACIM, BLDC & PMSM

Software Platform’s Supported

Kinetis Motor Suite

Kinetis Reference Design

Motor Control Toolbox

High Voltage Reference Design
From $600

=

+

+

HVP-MC3PH

includes HVP-KV46F150M card KMS-ACIM

85/230 volt,

1KW 3-ph motor control inverter

inc PFC circuitry

Controller Card

HVP-KV10Z32
HVP-KV11Z75M
HVP-KV31F120M
HVP-KV58F220M

Your motor

Motor Control Software Enablement

Easy & Fast Design

- **For customers who have**
  - limited experience of motor control or
  - a limited time to market.
  - budget constraints
- **Focuses on delivering**
  - ease of use
  - IP that can quickly improve efficiency
- Simple Graphical Interface for **ALL** configuration, tuning and initial application development.
- System manages as much of the development as possible
- Tight integration with toolchain simplifying development
- www.nxp.com/kinetismotorsuite

Most Efficient Design

- **For customer who have:**
  - Expert level of knowledge
  - Ability and time to tune motor efficiently
- **Focuses on delivering**
  - best in class motor control efficiency
  - highest rated motor performance
  - Integration with NXP's IEC60730 safety software
- Graphical Interface for motor parameter reading
- Leverages experience and IP developed working with OEMs to provide high efficiency
- System solution not just library functions of motor functions.
- www.nxp.com/kinetisdesigns

Model Based Design

- **For customer who have:**
  - Demand to use model based design
  - Limited time to market
- **Focuses on delivering**
  - Model to MCU code development efficiency.
  - Excellent rate motor perfromance
  - Integration with NXP's Embedded Software Librariy and FreeMASTER tool.
- Model based development environment that automatically generates MCU Targets software.
- Leverage the industry standard in Model Based design - MATLAB
- Integration with NXP's Motor Control toolchain;
- www.nxp.com/motorcontroltoolbox

Motor Control Software Enablement

Easy & Fast Design

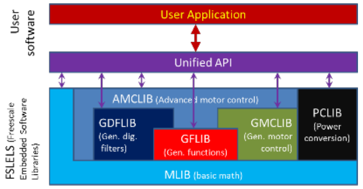
Kinetis Motor Suite

- Intuitive development solution that enables the design of sensored and sensorless BLDC & PMSM motor control applications quickly and efficiently via a simple graphical user interface.
- Automated motor and system inertia identification.
- Single Parameter to tune motor response.
- Disturbance rejection control algorithm ensures high performance even in highly dynamic operating conditions

- **Cost: Starts at $1.60 inc. MCU & SW**
- KV3x supported today, others in 2016
- www.nxp.com/kms

Most Efficient Design

Kinetis Reference Software


User software
User Application
Unified API
FSLELS (Freescale Embedded Software Libraries)
AMCLIB (Advanced motor control)
GDFLIB (Gen. dig. filters)
GFLIB (Gen. functions)
GMCLIB (Gen. motor control)
PCLIB (Power conversion)
MLIB (basic math)

- Sensorless FOC BLDC, PMSM & ACIM Supported today

- Complete Software Solutions built on NXP Embedded Software Libraries.

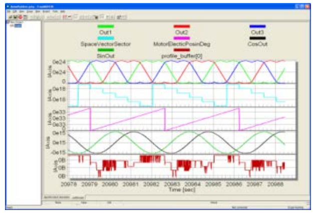
- Libraries of software algorithms for Math, Motor Control, Power Conversion, Filters and Advanced functions. ~200 algorithms available

- **Cost: free of charge**

- All Kinetis V MCU's are supported today
- www.nxp.com/kinetisdesigns

Model Based Design

Motor Control Toolbox



- MCU Targets for MATLAB™/Simulink™ modelling environment
- Motor control plug-in tool for automatic code generation.
- Supports multiple compilers. FreeMASTER compatible.
- Incorporates NXP ESL

- All Kinetis V MCU's are supported today

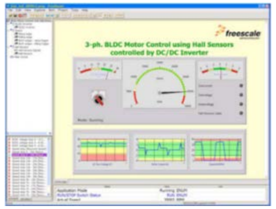
- www.nxp.com/motorcontroltoolbox

Software Enablement Guide

MCAT (Motor Control Application Tuner)

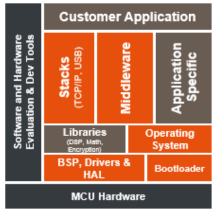
- GUI based FreeMASTER plug-in tool that provides real-time monitoring, tuning and updating of motor control system parameters
- Provided as a plug-in for the FreeMASTER tool. Designed to work with NXP Ref. Design s/w
- http://www.youtube.com/watch?v=ZsLQzSTnhgo
- **Cost: free of charge**
- Supported devices
  - KV1x, KV3x, KV4x, KV5x

FreeMASTER


freescale
3-ph. BLDC Motor Control using Hall Sensors
controlled by DC/DC Inverter

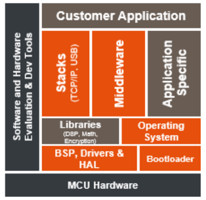
- Complimentary GUI based run-time debug monitor and data visualization tool
- Replaces debugger in situations when the core can not be simply stopped, ideal for motor control and power conversion application development
- http://www.youtube.com/watch?v=vKVlxu8ecdg
- **Cost: free of charge**
- Supported devices
  - KV1x, KV3x, KV4x, KV5x

MQX RTOS / FreeRTOS



- Commercial-grade MCU software platform at no cost with optional support packages
- RTOS Kernel, Real Time TCP/IP Communication Suite, File System, USB Host/Device Stack and Board Support Packages
- **Cost: free of charge**
- Supported devices
  - KV1x, KV3x , KV4x & KV5x

Kinetis SDK & Processor Expert Code Generator



- SDK – a complete software framework for developing applications across all Kinetis MCUs. h/w abstraction, peripheral drivers, stacks, RTOS's, utilities, and usage examples; delivered in C source
- Processor Expert – GUI Eclipse plug-in tool for creating and configuring software and peripheral drivers quickly & easily
- **Cost: free of charge**
- Supported devices
  - KV1x, KV3x & KV4, KV5x

KMS Product Enablement

- Unique part numbers help identify KMS enabled devices:
  - Part numbers are unique to the specific motor type enabled:
    - P = PMSM / BLDC – supports PMSM and BLDC motors with sinusoidal FOC control
    - Q = ACIM – supports ACIM motors with sinusoidal FOC control
    - Example: MKV30F128VLF10P

- License is specific to that MCU, enabling access to the KMS software components.
  - Only have to pay for the license's based on how many of your end products that you will build.
  - Enables access to NXP technical support for Kinetis Motor Suite and to the community.
    - Applications support available in each region.

- KMS is available on Kinetis KV3x MCU parts in production today and will be rolling out across the Kinetis V series in 2016
- Samples are available at www.nxp.com/kinetisv

- A small ASP premium is added for KMS enabled devices over the standard MCU pricing
  - For example the license is $0.30 for a KV3x with 128kb Flash @ 100MHz – total cost is $1.84 for 10k units
  - Comparing this to similar competitive software offerings where the premium for the license is more than $3 and the MCU is more than $7 @ 1k units

For a small premium adder you significantly reduce your development cycle time, improve system performance and increase end product longevity and in turn reduce your support costs.

How Do I Get Started With Kinetis Motor Suite Now?

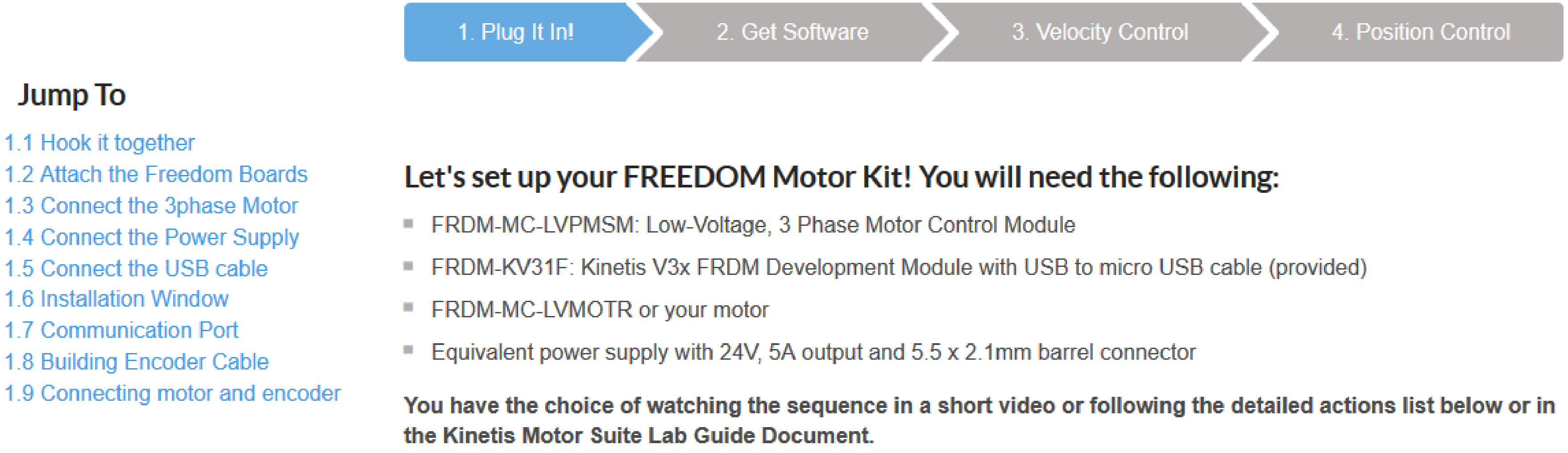
- Order your KMS enabled hardware, and walk through the OOBE video’s;
  - Low voltage: nxp.com/frdm-kv31f
  - High voltage: nxp.com/hvp
- Install Kinetis Standard Tools:
  - MCUXpresso IDE with integrated Pins and Clocks Tools – used for KMS PMSM and BLDC projects
  - MCUXpresso IDE with integrated Pins and Clocks Tools – used for KMS PMSM and BLDC projects
  - Kinetis Design Studio (KDS version 3.2.0) – used for KMS ACIM projects
  - MCUXpresso SDK 2.x
- Download KMS Installer and Documentation:
  - Kinetis Motor Suite Installer (version 1.2.0)
- Walk through the labs to get familiar with KMS
- Start your application development on the development boards
- Order your free MCU samples from nxp.com to build your production board prototypes

0.6 Resources

Getting Started with Kinetis Motor Suite

Go to http://www.nxp.com/kms

Click on the Getting Started Tab



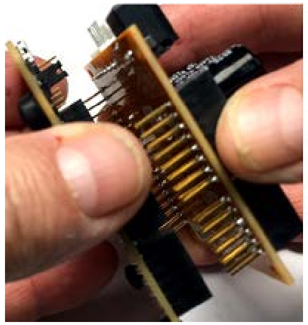
Jump To

- 1.1 Hook it together
- 1.2 Attach the Freedom Boards
- 1.3 Connect the 3phase Motor
- 1.4 Connect the Power Supply
- 1.5 Connect the USB cable
- 1.6 Installation Window
- 1.7 Communication Port
- 1.8 Building Encoder Cable
- 1.9 Connecting motor and encoder

Let's set up your FREEDOM Motor Kit! You will need the following:

- FRDM-MC-LVPMSM: Low-Voltage, 3 Phase Motor Control Module
- FRDM-KV31F: Kinetis V3x FRDM Development Module with USB to micro USB cable (provided)
- FRDM-MC-LVMOTR or your motor
- Equivalent power supply with 24V, 5A output and 5.5 x 2.1mm barrel connector

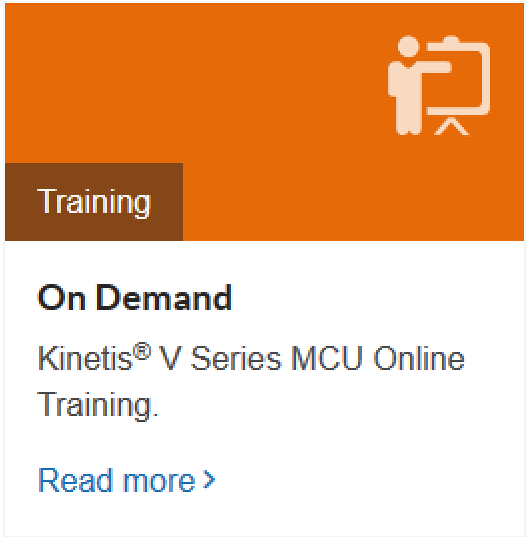
You have the choice of watching the sequence in a short video or following the detailed actions list below or in the Kinetis Motor Suite Lab Guide Document.



Kinetis Motor Suite Videos

Go to http://www.nxp.com/kms

- Scroll down to Training and Events
- Click *Kinetis V Series MCU Online Training*

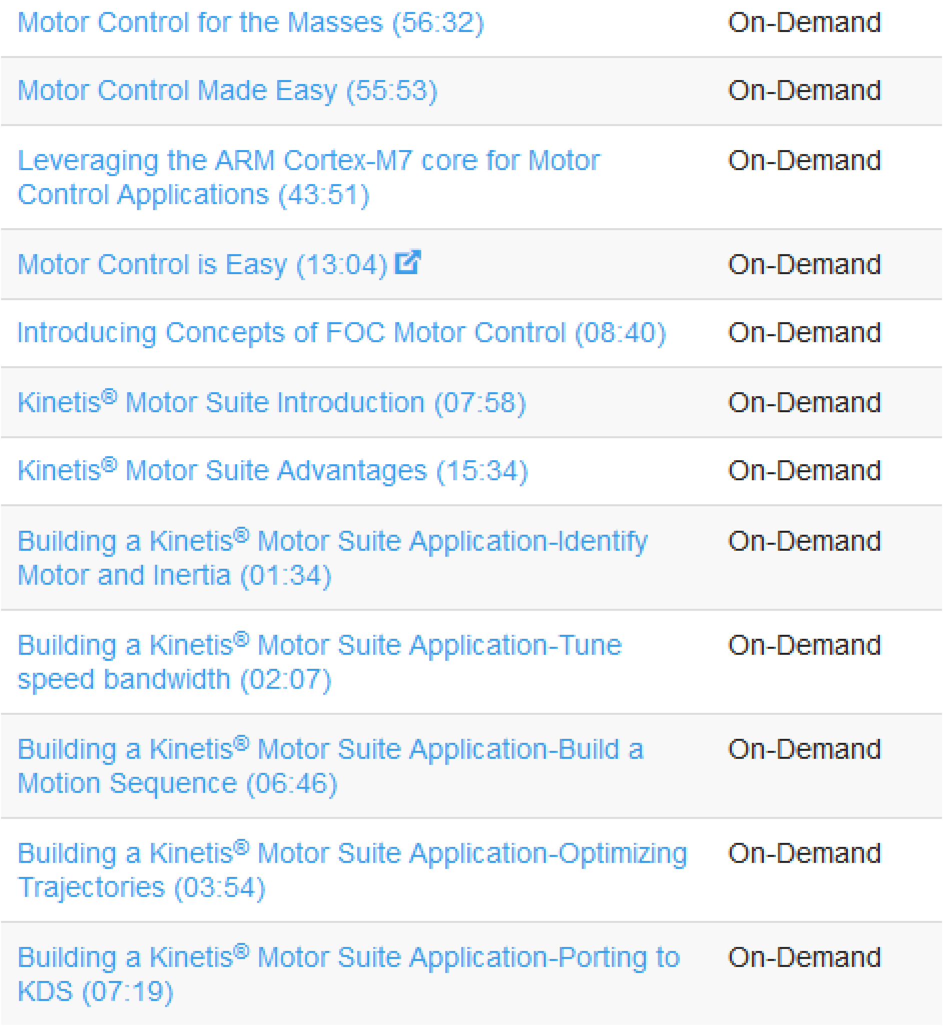


Training

On Demand

Kinetis® V Series MCU Online Training.

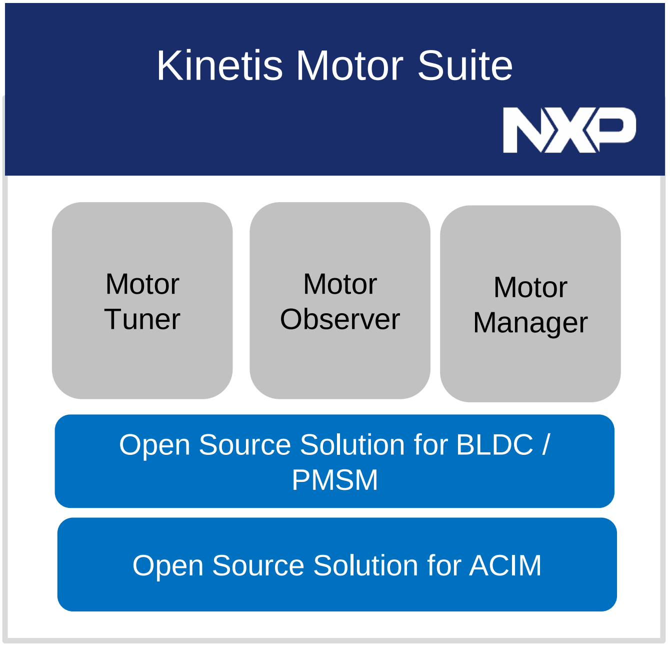
Read more ›



Motor Control for the Masses (56:32)	On-Demand
Motor Control Made Easy (55:53)	On-Demand
Leveraging the ARM Cortex-M7 core for Motor Control Applications (43:51)	On-Demand
Motor Control is Easy (13:04)	On-Demand
Introducing Concepts of FOC Motor Control (08:40)	On-Demand
Kinetis® Motor Suite Introduction (07:58)	On-Demand
Kinetis® Motor Suite Advantages (15:34)	On-Demand
Building a Kinetis® Motor Suite Application-Identify Motor and Inertia (01:34)	On-Demand
Building a Kinetis® Motor Suite Application-Tune speed bandwidth (02:07)	On-Demand
Building a Kinetis® Motor Suite Application-Build a Motion Sequence (06:46)	On-Demand
Building a Kinetis® Motor Suite Application-Optimizing Trajectories (03:54)	On-Demand
Building a Kinetis® Motor Suite Application-Porting to KDS (07:19)	On-Demand

Summary: Kinetis & KMS

- Kinetis V is having a lot of success in the market, with good growth.
- KMS1.2 expands NXP’s offering to cover more products in KV series, and more motors.
- Community support from technical experts



Kinetis Motor Suite

NXP

Motor Tuner

Motor Observer

Motor Manager

Open Source Solution for BLDC / PMSM

Open Source Solution for ACIM

Questions?

Resources and Support

- Kinetis V Training: www.nxp.com/kinetisvtraining
  - Kinetis Motor Suite 1Hr Webinar
  - Short Motor Control Introduction Videos
  - Short KMS Introduction Videos

- Kinetis V series MCUs: www.nxp.com/Kinetis

- Kinetis Motor Suite: www.nxp.com/KMS

- For Support log on to the Kinetis Community: http://community.nxp.com/community/kinetis/kms

- For questions, feel free to contact:
  - Fraser McHenry: @ Fraser.McHenry@nxp.com Global Product Manager
  - Richy Ye @ richy.ye@nxp.com – Technical Support Asia
  - Philip Drake @ philip.drake@nxp.com – KMS Technical Support

SECURE CONNECTIONS
FOR A SMARTER WORLD

www.nxp.com