FTF FREESCALE TECHNOLOGY FORUM 2015

Hands-On Workshop: Motor Control–**Customize Your Motor Control** Solution Using Kinetis V Series MCUs and the Complete Suite of Tools and Software – Advance

FTF-IND-F1295

Eduardo Viramontes | Applications Engineer

J U N E . 2 0 1 5

freescale™

External Use

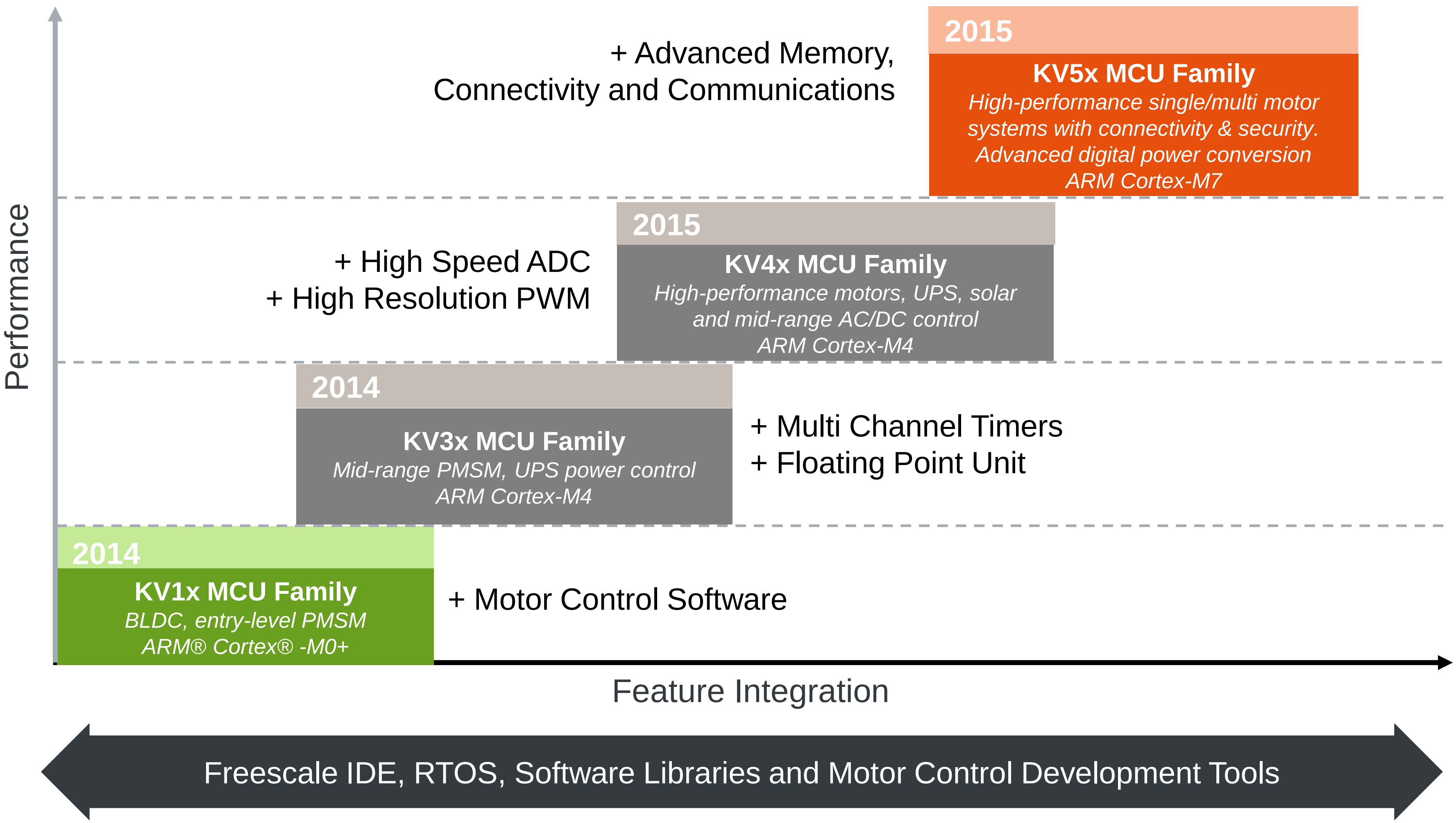
Agenda

- Maybe just no agenda?
- We'll learn some stuff together and later talk about robots or guitars or something cool.

Goals

- **Learn the basics of using the FSLESL**
  Freescale Embedded Software Libraries for motor control
- **Tune and spin a motor using MCAT and FreeMASTER**
  Use tools and explore their power in motor control applications
- **Learn about common motor control issues**
  Things you might run into when bringing up a motor control application and how to overcome them

New Levels of Performance, Reliability and Power Efficiency for Motor Control and Digital Power Conversion



Kinetis V Series Performance and Feature Scalability

Key IP for Motor and Power Control Applications

MCU Family	Core	Memory	Motor Control Timers	Enhanced Timers	ADC	DAC	CMP	Comms	Packages
KV5x	240MHz CM7 DSP + FPU	512kB-1MB Flash	2 x 8ch 1x 2ch FlexTimers	2 x 12ch eFlexPWM + Nano-Edge	4 x 12bit 5Msps 1 x 16bit	1x 12-bit	4x ACMP with 6-bit DAC	ENET 3 x CAN	144 pin 100 pin
KV4x	160MHz CM4 DSP + FPU	64-256kB Flash	2 x 8ch 1x 2ch FlexTimers	12ch eFlexPWM + Nano-Edge	2x 12bit 4.1Msps / 1.9Msps	2x 12-bit	4x ACMP with 6-bit DAC	2 x CAN	100 pin 64 pin 48 pin
KV3x	100/120MHz CM4 DSP + FPU	64-512kB Flash	2x 8ch 2x 2ch FlexTimers	-	2x 16-bit 1.2Msps	2x 12-bit	2x ACMP with 6-bit DAC	-	100 pin 64 pin 48 pin 32 pin
KV1x	75MHz CM0+ H/W DIV & SQRT	16-32kB Flash	1x 6ch 2x 2ch FlexTimers	-	2x 16-bit 1.2Msps	1x 12-bit	2x ACMP with 6-bit DAC	1 x CAN	64 pins 48 pin 32 pin

Scalable performance, timing and analog functionality based on application needs

Kinetis V Series Motor Control Performance

MCU Family	Motor control application level
KV5x	Single up to 4 motor control (depending on application complexity) Faster sampling/control rates, application integration and communications
KV4x	Single up to 4 motor control (depending on application complexity) Faster sampling/control rates for faster electrical constant motors
KV3x	Single motor + PFC, higher complexity, faster electrical coefficient Possible dual motor depending on application complexity
KV1x	Single motor + PFC or quadrature encoder, slower control (about up to 12 kHz in sensorless FOC)

Scalable performance, timing and analog functionality based on application needs

Kinetis V Series KV10 & KV11 : Block Diagram

Core/System
- 75MHz Cortex-M0+ with Hardware Divide & Square Root
- 4ch DMA

Memory
- **16/32/64/128KB Flash**
- **8/16KB SRAM**
- Option with FAC

Communications
- Multiple serial ports + **1 FlexCAN***

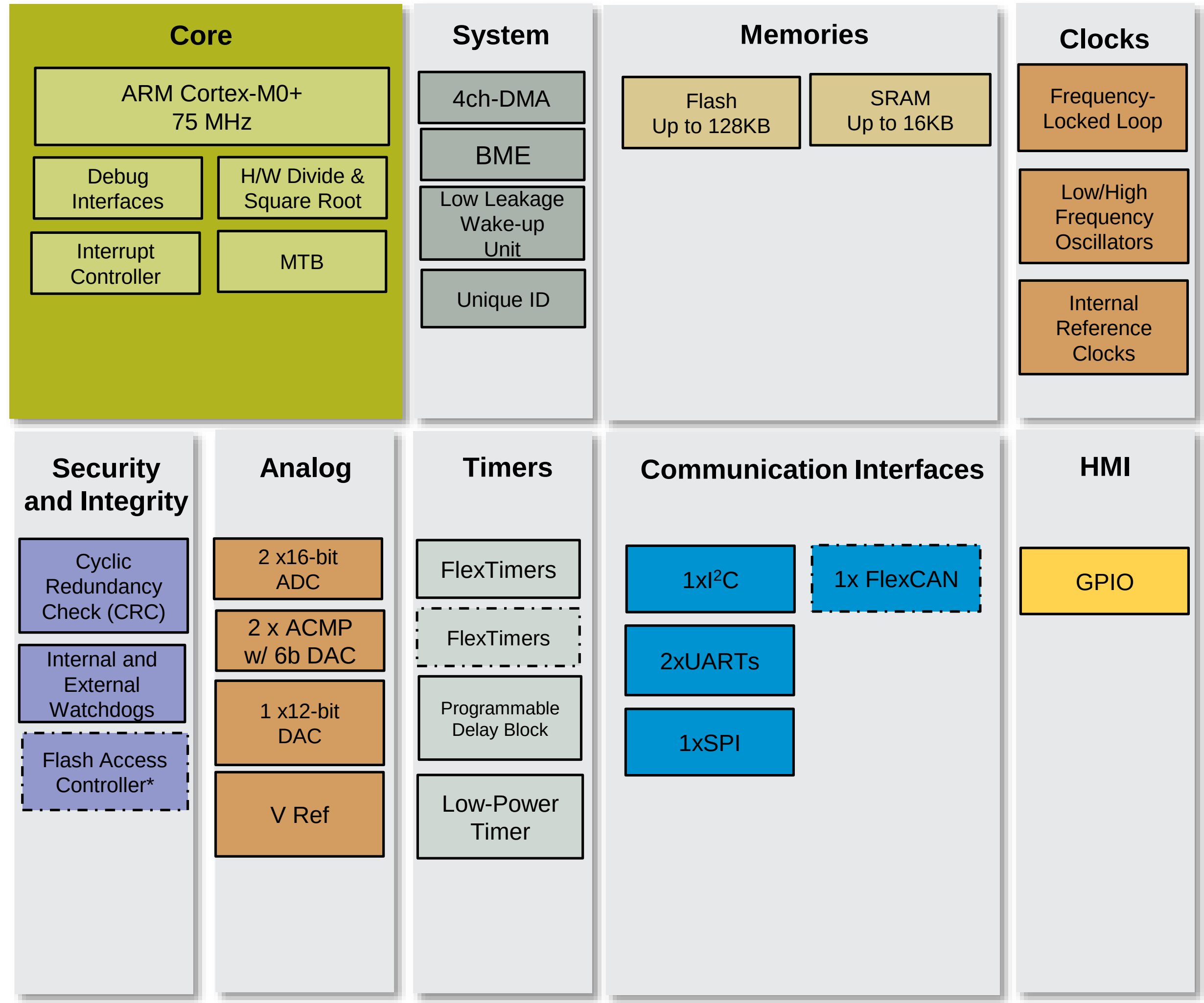
Analog
- 2 x 8ch 16-bit ADC
  - 1.2Msps in 12-bit mode (835ns)
- 1 x12-bit DAC
- 2 x ACMP with 6-bit DAC

Timers
- **Up to 2x6ch FlexTimer (PWM) ***
- Up to 4x2ch FlexTimer (PWM/Quad Dec.)
- Low Power Timer

Other
- 32-bit CRC
- Up to 40 I/Os
- 1.71V-3.6V; -40 to 105C

Packages
- 32QFN, *32LQFP, 48LQFP, **64LQFP**
- * Package Your Way

From $0.89 to $1.89 @ 10k units



*Optional

Availability: 16 & 32KB in Production Now

64 & 128KB PK samples Now, production July 2015

Starting with a Reference Design

- Download a motor control reference design
- Evaluate and understand application
- Experiment with FreeMASTER

Start with a Freescale Reference Design

AN5049 Three-Phase PMSM Sensorless FOC Using the MKV10Z32 with Automated Motor Parameters Identification	Application Notes	Multiple	pdf	1378	0	12/19/2014	Download
AN4986 Automated PMSM Parameter Identification	Application Notes	Multiple	pdf	483	0	10/16/2014	Download
AN4935 PMSM Sensorless FOC for a Fan Using the Kinetis KV10 Show more languages	Application Notes	Multiple	pdf	2777	1	9/24/2014	Download
AN4649 Power supply options for Tower development platform Show more languages	Application Notes	Multiple	pdf	2559	0.1	9/23/2014	Download
AN4373 Cookbook for SAR ADC measurements	Application Notes	Multiple	pdf	823	1	4/17/2014	Download

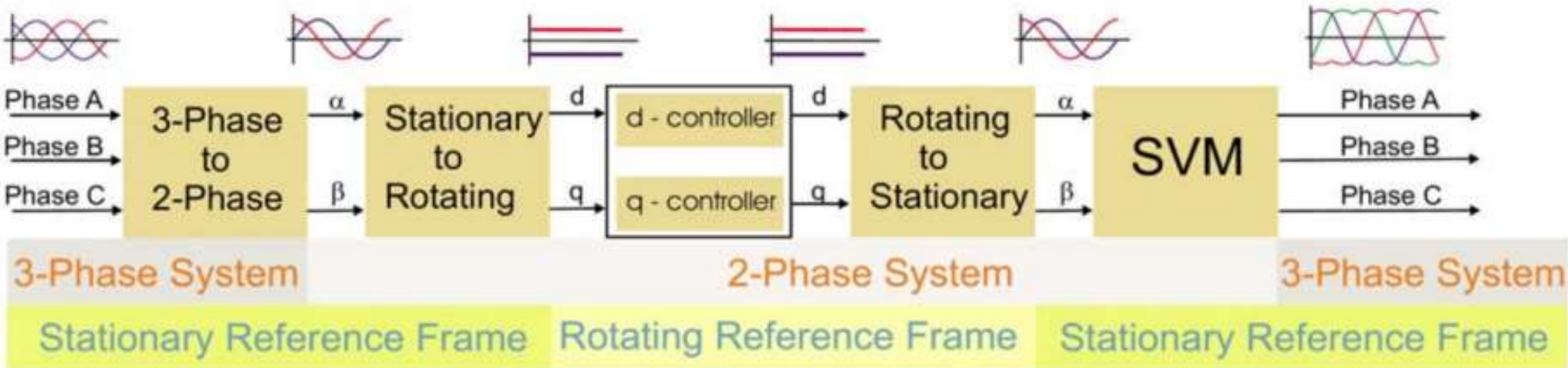
- Instead of writing all the code from scratch, start with a reference design or app note.
- Customize as needed.

Evaluate the Reference Design – Lab 1

- Boards have already been flashed with code to save some time.
- Open your lab guide and follow steps to lab 1 only.
- Open a few graphs in FreeMASTER figure out what they’re showing and how you can use that to evaluate and develop motor control applications.

What is this Application Doing?

- Extremely short introduction to Field-Oriented Control*

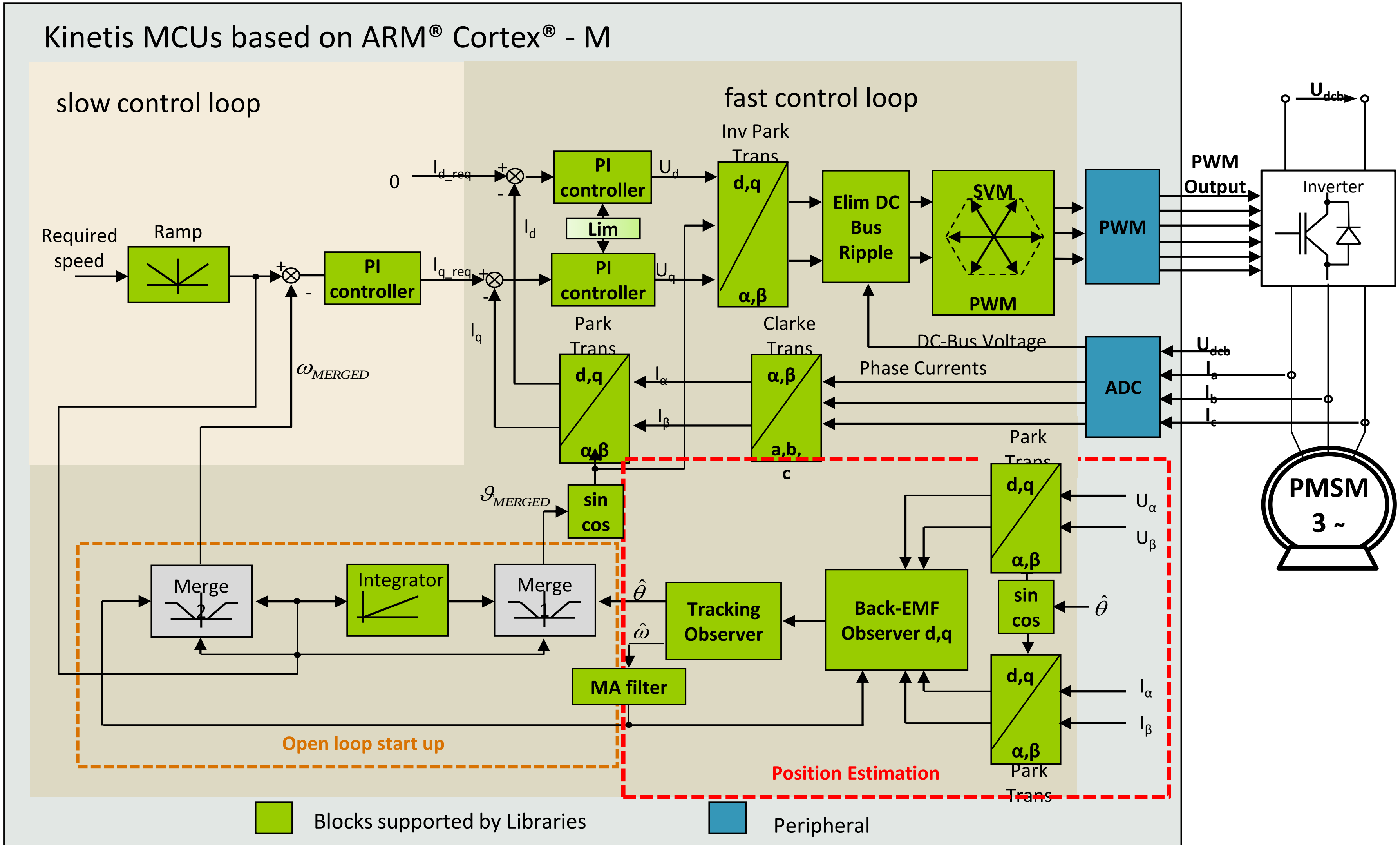


Clark transformation	Park transformation	PI Controllers	Inverse park transformation	Inverse Clark transformation + space vector

* Does not replace an actual class in FOC

What is this Application Doing?

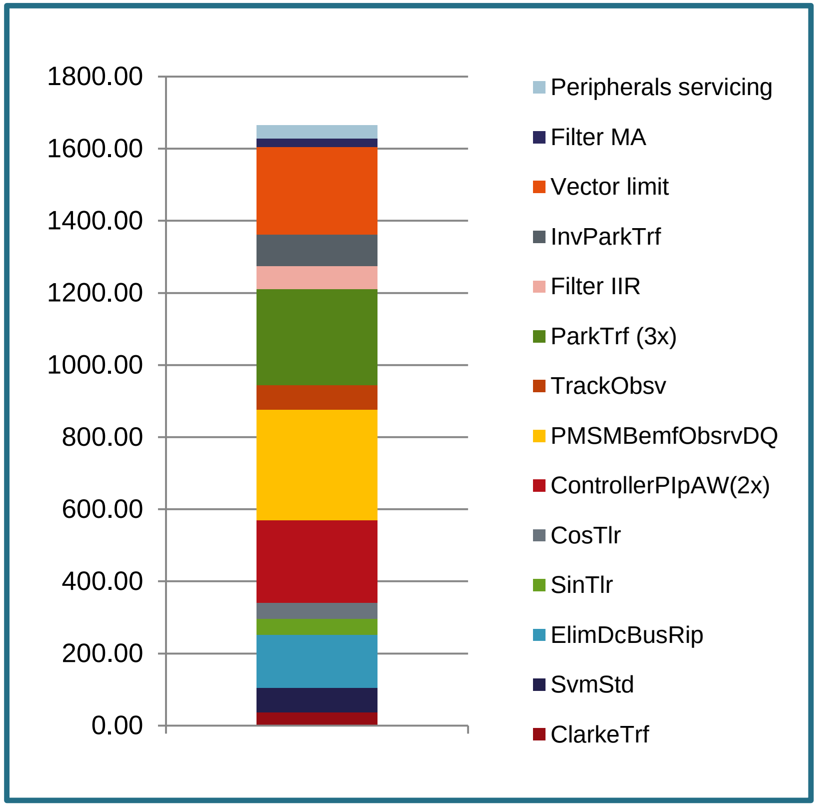
FOC Implementation Block Diagram

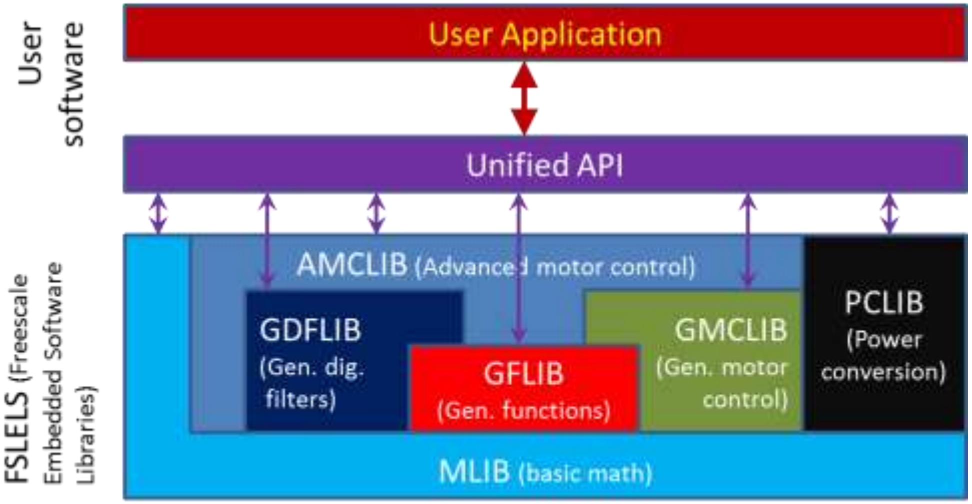


What is this Application Doing?

- Main FSLESL functions used in implementation

Sensorless Solution





Graph describes functions in the current control loop which is both the critical and the most CPU-consuming portion of the application

How to use FSLESL?

```
/*************************************************************************//*!
*
* @brief   PMSM field oriented current control
*
* @param   MCS_PMSM_FOC_A1_T *psFocPMSM
*            - structure of PMSM FOC parameters
*            IN: -> sIABC - input ABC phases currents
*            IN: -> sAnglePosEl - angle where the currents were measured
*            IN: -> sAnglePosElReload - angle where the next PWM reload
*            IN: -> f16UDcBusFilt - DC bus voltage
*            IN: -> f16UDcBusFilt - actual DCBus voltage value
*            IN: -> f16DutyCycleLimit - determines the max. value of duty cycle
*            IN/OUT -> sIdPiParams - D current controller structure
*            IN/OUT -> sIqPiParams - Q current controller structure
*            IN/OUT -> i16IdPiSatFlag - D current controller saturation flag
*            IN/OUT -> i16IqPiSatFlag - Q current controller saturation flag
*            OUT -> sDutyABC - ABC duty cycles
*            OUT -> uw16SectorSVM - Next step SVM sector
*
* @return  N/A
*
******************************************************************************/
void MCS_PMSMFocCtrlCurrentA1(MCS_PMSM_FOC_A1_T *psFocPMSM)
{
    /* Position angle of the last PWM update */
    psFocPMSM -> sAnglePosEl = psFocPMSM -> sAnglePosElReload;

   /* sine and cosine of the angle */
    psFocPMSM -> sAnglePosElReload.f16Sin = GFLIB_Sin(psFocPMSM -> f16PosElReload);
    psFocPMSM -> sAnglePosElReload.f16Cos = GFLIB_Cos(psFocPMSM -> f16PosElReload);

    /* 3-phase to 2-phase transformation to stationary ref. frame */
    GMCLIB_Clark(&psFocPMSM -> sIAlBe, &psFocPMSM -> sIABC, F16);

    /* 2-phase to 2-phase transformation to rotary ref. frame */
    GMCLIB_Park(&psFocPMSM -> sIDQ, &psFocPMSM -> sAnglePosEl, &psFocPMSM -> sIAlBe, F16);

    /* D current error calculation */
    psFocPMSM -> sIDQError.f16D = MLIB_SubSat_F16(psFocPMSM -> sIDQReq.f16D, psFocPMSM -> sIDQ.f16D);
```

- Complex functions pass pointers to input and output data structures
- Use fractional arithmetic (words represent values from 0 to 1 or -1 to 1 based on max and min values)

Tuning a Motor

- Improve tuning and performance of the motor using MCAT
- Understand parameters that can be tuned and modified
- Understand what the FSLESL are doing

Tune Your Motor with MCAT – Lab 2

- We will now improve the performance of the motor tuning.
- Open the MCAT FreeMASTER project per Lab 2 guide.
- Follow the rest of the Lab 2 guide

Lab 2 – What We're Trying To Do

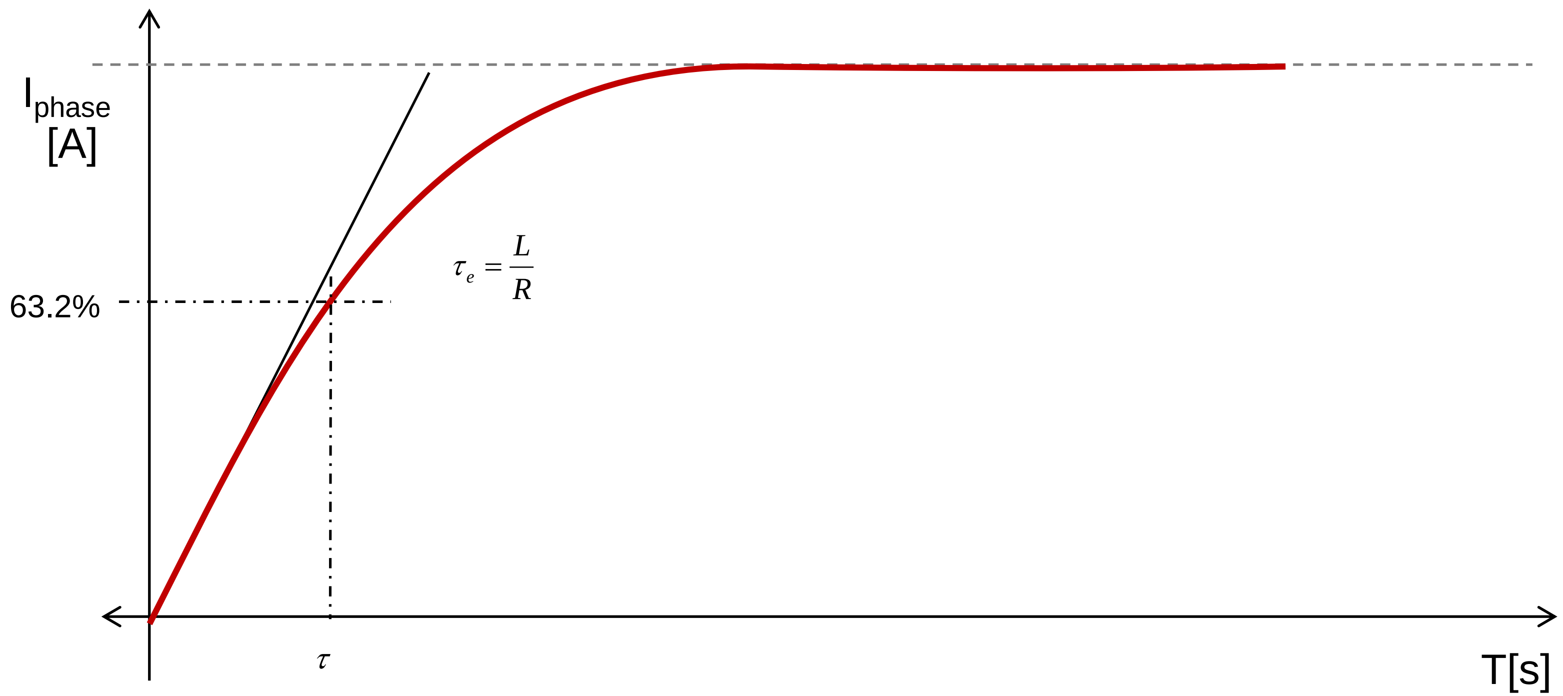
Or whichever performance is specified for your application

Customizing the Application

- Change PWM frequency
- Add a new task
- Re-flash the part, test the changes

Why Change PWM Frequency: Dynamic Performance

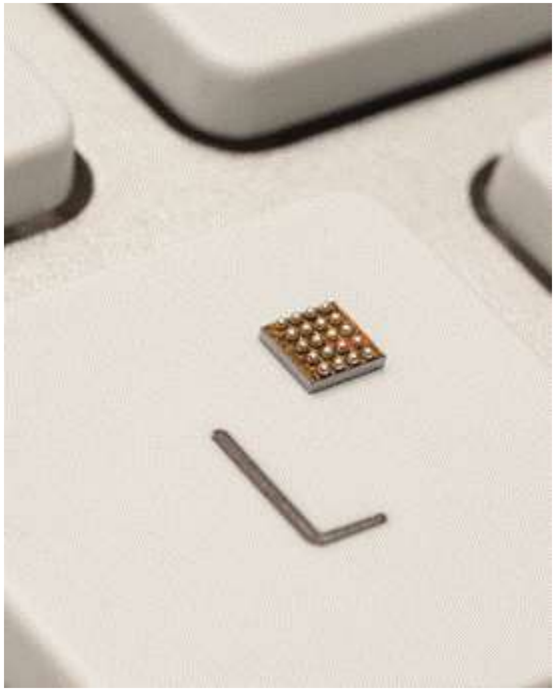
- High Dynamic Applications
  - Depending on motor design and application dynamics, it may be necessary to run the control loop faster than a certain threshold.
  - A good rule of thumb is that the application should be running 10 times faster than τ_e



Modify Stuff – Lab 3

- We will now modify the PWM frequency
- We will also change library parameters on the fly (ramp rate change)
- Open the lab 3 guide and follow instructions.
- Instructor will also be showing code modifications on screen.
- Retune application to new PWM rate.
- Test modifications.

What About the Hardware?



Things to consider when porting a motor application from evaluation to custom platform:

- Pinouts: make sure special functions are kept.
- Pre-driver: different pre-drivers may need different connections or communication methods (or not comms at all).
- Other protections: evaluation platforms will not necessarily include all protections that a final application might need.
- Scaling needs may be different, meaning changes to op-amp circuits and so on, requiring further changes to the software.

References

- AN5049
  Three-Phase PMSM Sensorless FOC Using the MKV10Z32 with Automated Motor Parameters Identification
- AN4912
  Tuning 3-Phase PMSM Sensorless control application using MCAT Tool
- DRM148
  Sensorless PMSM Control Design

www.Freescale.com