# XMC PMSM FOC SENSORLESS SW **V1.5.8**

XMC™ Microcontrollers
May 2019

# PMSM FOC Agend

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

# Agenda

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

# Overview – PMSM FOC Sensorless SW

- This document provides information about usage of PMSM FOC Sensorless example software on Infineon's XMC13/XMC14 series micro-controllers platform.
- PMSM FOC Sensorless control example software is offered as "simple main project in DAVE™ IDE".
- PMSM FOC Sensorless control example project consists of Single Shunt/three shunt Field Oriented control algorithm software, targeted end applications are fans, pumps, and e-bike segment.
- This example project will provide high level of configurability and modularity to address different segments.
- This project can be easily configured as per requirements with the help of configuration files.

# PMSM FOC Sensorless SW - Overview

- Dedicate Application Note AP32370
- Easy partitioning of code structure, easy to understand algorithm and apply custom modification.
- Small code size, and fast execution (21usec all FOC algorithm)
- Flexible hw configuration
- Possibility to integrate with APPs
- Easy to update existing prjs with new Library version
- Compiler Support GCC, Keil
- Complete access to source code and parameters calculation (except for PLL library, available under SLA)

**PMSM FOC motor control software using XMC™**

**XMC1000**

**About this document**

**Scope and purpose**

This document describes the implementation of the PMSM FOC motor control software for 3-phase motor using the Infineon XMC1302, XMC1402, XMC1404 microcontroller.

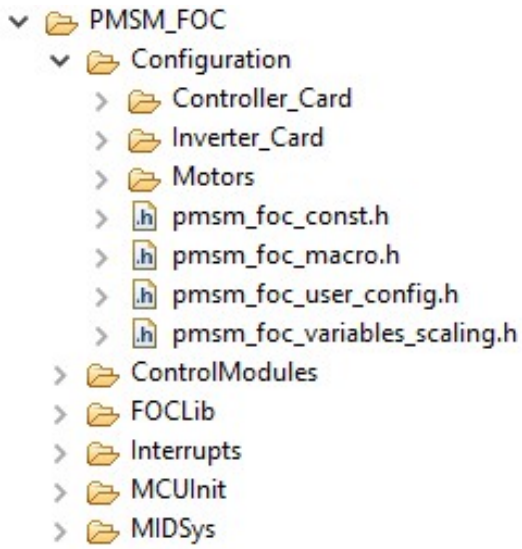
**Intended audience**

This document is intended for customers who would like a configurable system for FOC control with sensorless feedback on XMC™ series microcontroller.

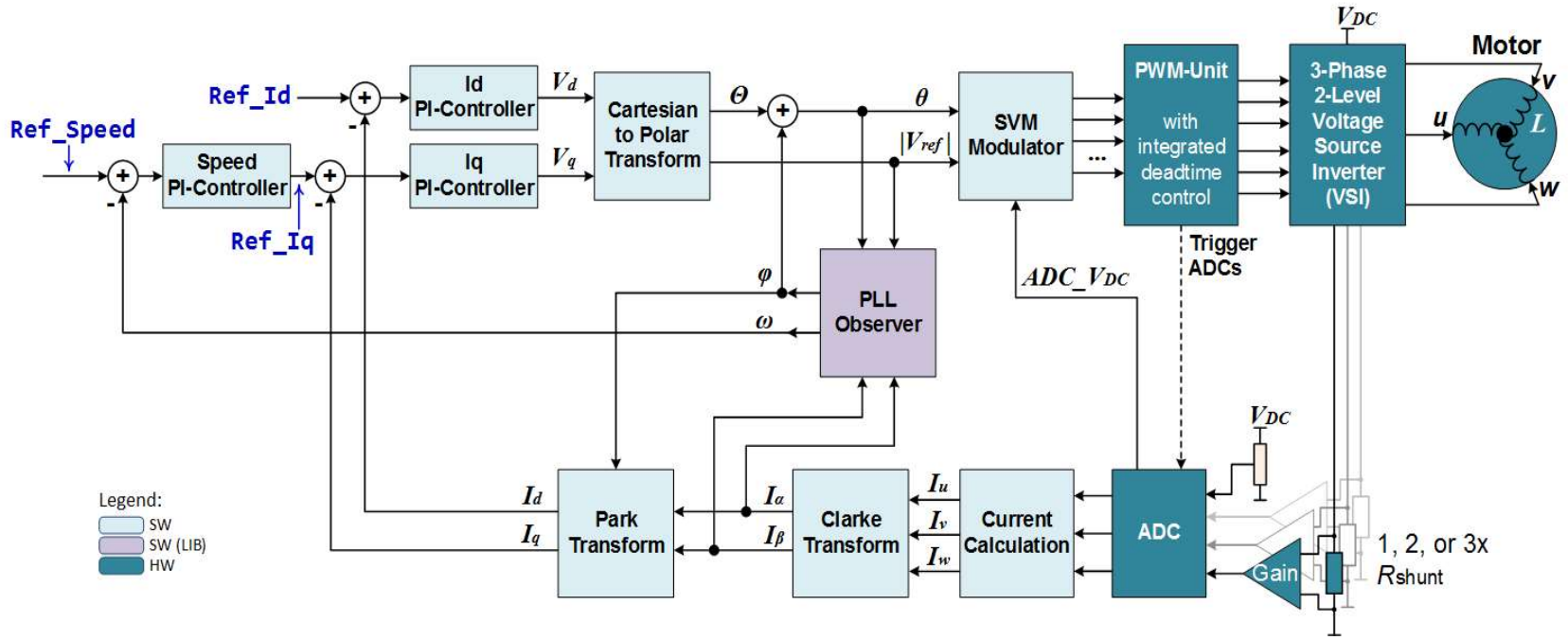
**Referenced documents**

[1] XMC1300 AB-Step Reference Manual, XMC1000 Family

[2] XMC1400 AA-Step Reference Manual, XMC1000 Family



# Software Overview – Software Blocks




| Software Blocks | Supported Options |
|---|---|
| **Control Scheme** | V/F control, V/F to closed loop control, direct constant speed control, direct constant torque control, direct constant Vq control |
| **PWM Modulation (Modulator)** | 7 Segment SVM, Over-modulation, |
| **Current/Voltage Measurement** | DC voltage compensation, DC bus voltage clamping during fast braking |

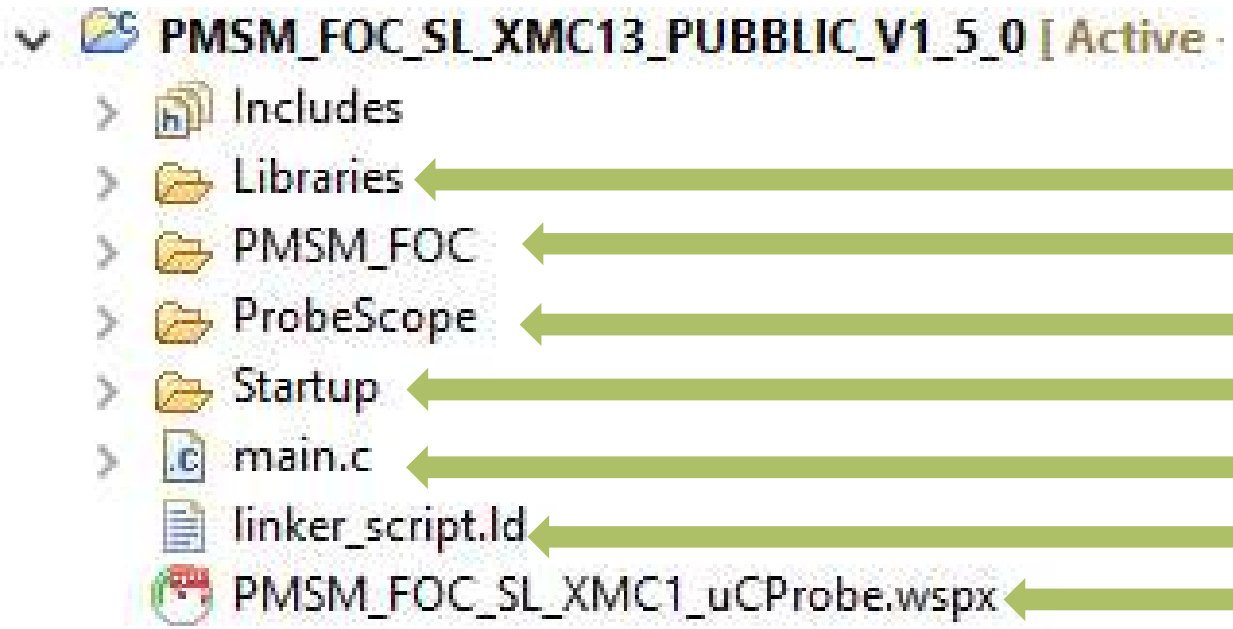
# Software Overview – Key Features

| Supported Features | Description |
|---|---|
| **DC bus voltage clamping** | Prevent over-voltage during fast braking |
| **Ramping** | S-curve Speed ramping, Linear Speed Ramping |
| **PI Controller** | Speed PI anti-windup (local and system), Torque PI controller, Flux PI controller |
| **Startup Algorithm** | Rotor alignment (Direct FOC), Open loop to MET |
| **Protection** | Under/Over voltage<br>C-trap with MCU hardware features |

# Agenda

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

# Software Project Overview - Files Structure




XMC & ARM library

PMSM FOC Root Folder

ucProbe streaming files

XMC startup files

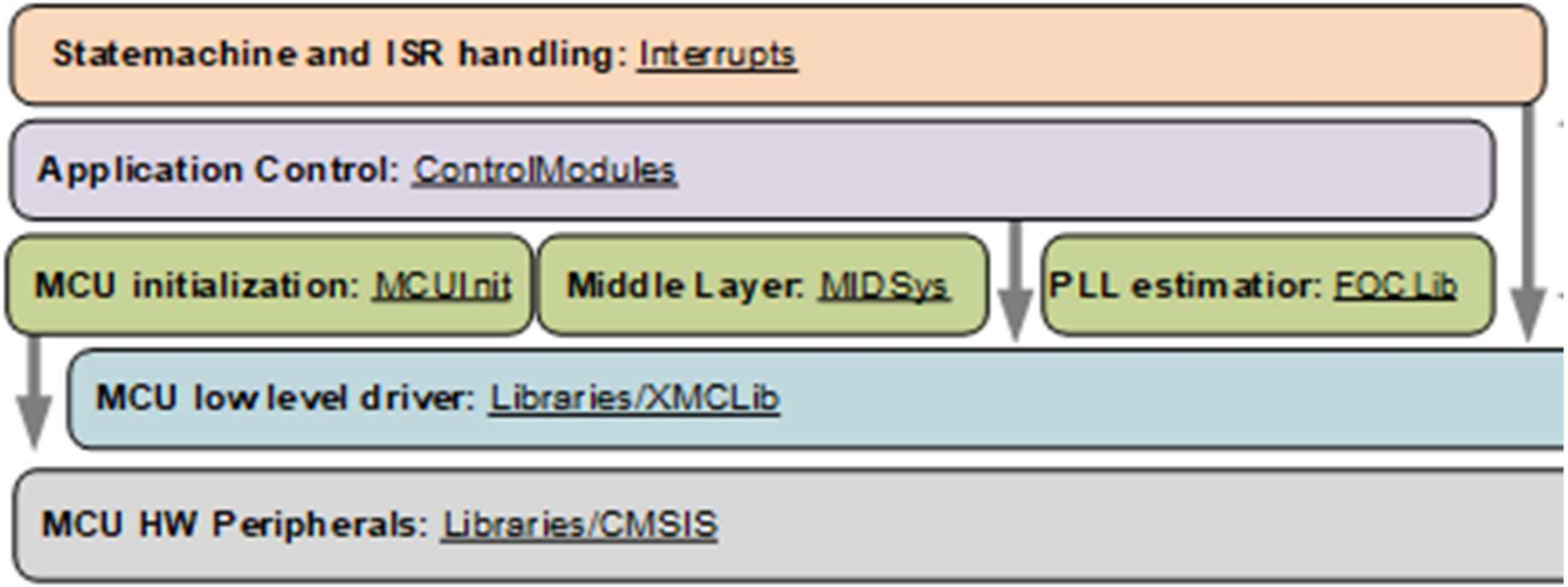
Main

Linker Desc. file

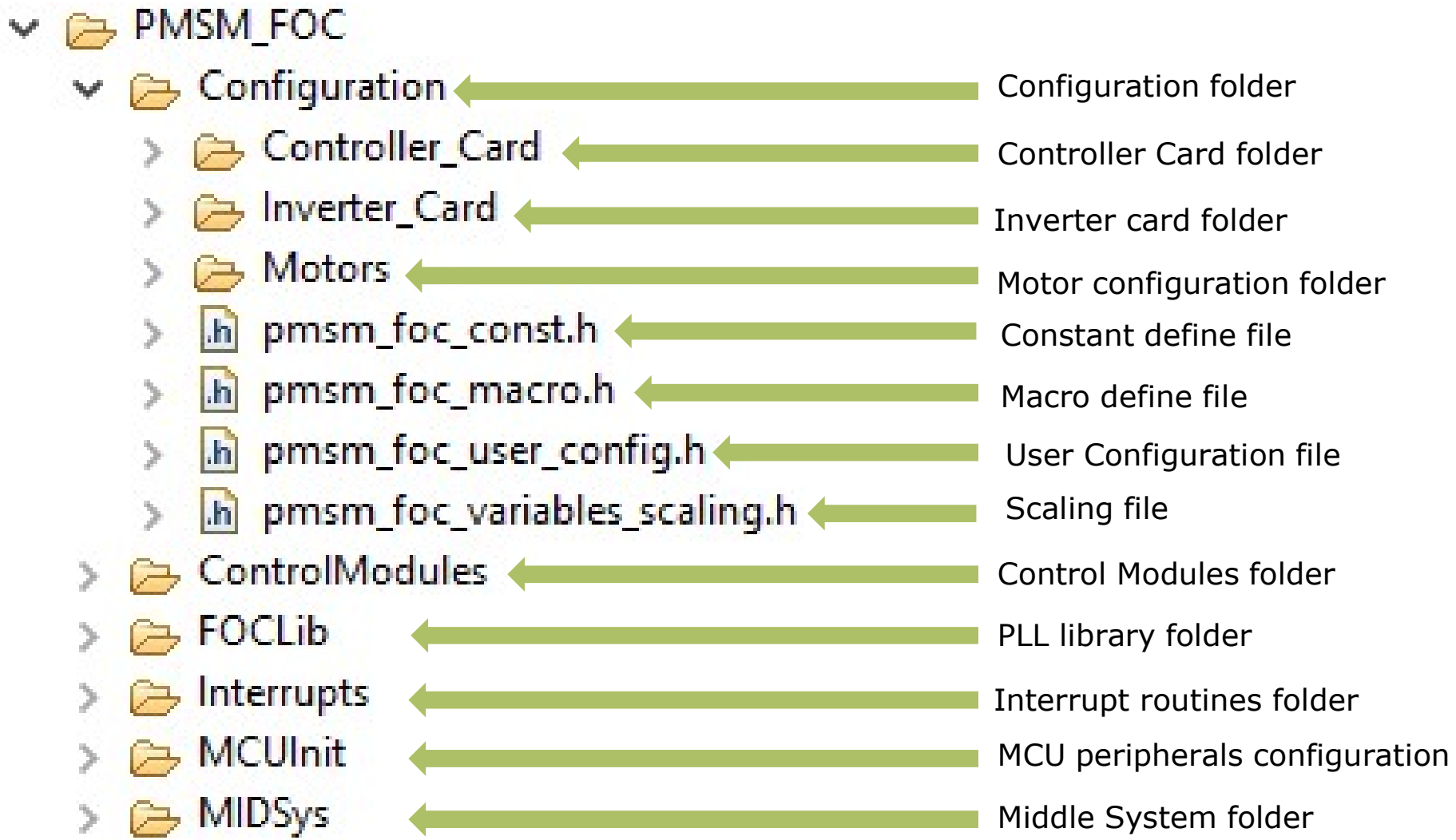
ucProbe GUI interface

# PMSM FOC Software Overview

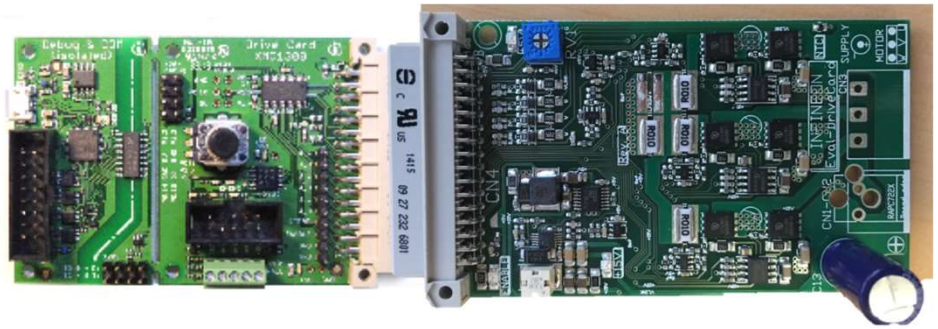
- PMSM FOC motor control software is developed based on well-defined layered approach. The layered architecture is designed in such a way as to separate the modules into groups. This allows different modules in a given layer to be easily replaced without affecting the performance in other modules and the structure of the complete system

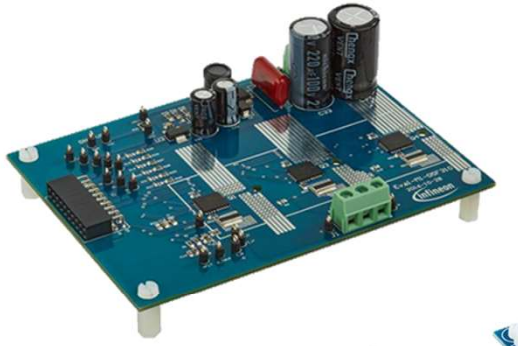


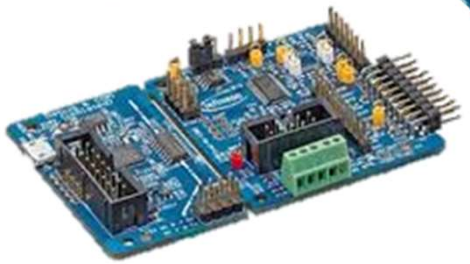
# PMSM FOC Software Overview- Files Structure

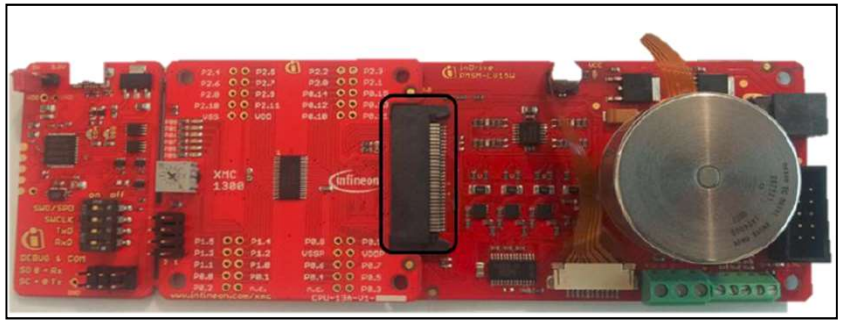


# PMSM FOC Software - Configuration







# Hardware supported 1/2

- KIT_XMC1X_AK_MOTOR_001
  - Infineon XMC1000 Motor Control Application Kit
- KIT_XMC750WATT_MC_AK_V1
  - XMC 750Watt Motor Control Application Kit
- KIT_XMC14_BOOT_001
  - XMC1404 CPU card for IT_XMC1X_AK_MOTOR_001
- KIT_MOTOR_DC_250W_24V
  - XMC13/XMC14 DC with 250W Low voltage Board
- IFX_MADK_EVAL_M1_05F310
  - MADK Low voltage board 250-300W
- IFX_MADK_EVAL_M1_05_65D_V1
  - MADK High voltage board 100-150W
- IFX_MADK_EVAL_M1_CM610N3
  - MADK High voltage board 750W
- CUSTOM_KIT
  - User defined motor control system

# Hardware supported 2/2

## Controller Card

- EVAL_M1_1302
- KIT_XMC13_BOOT_001
- KIT_XMC1300_DC_V1
- BOOTKIT_XMC1400_V1
- KIT_XMC1400_DC_V1
- CUSTOM_MCU

## Inverter Card

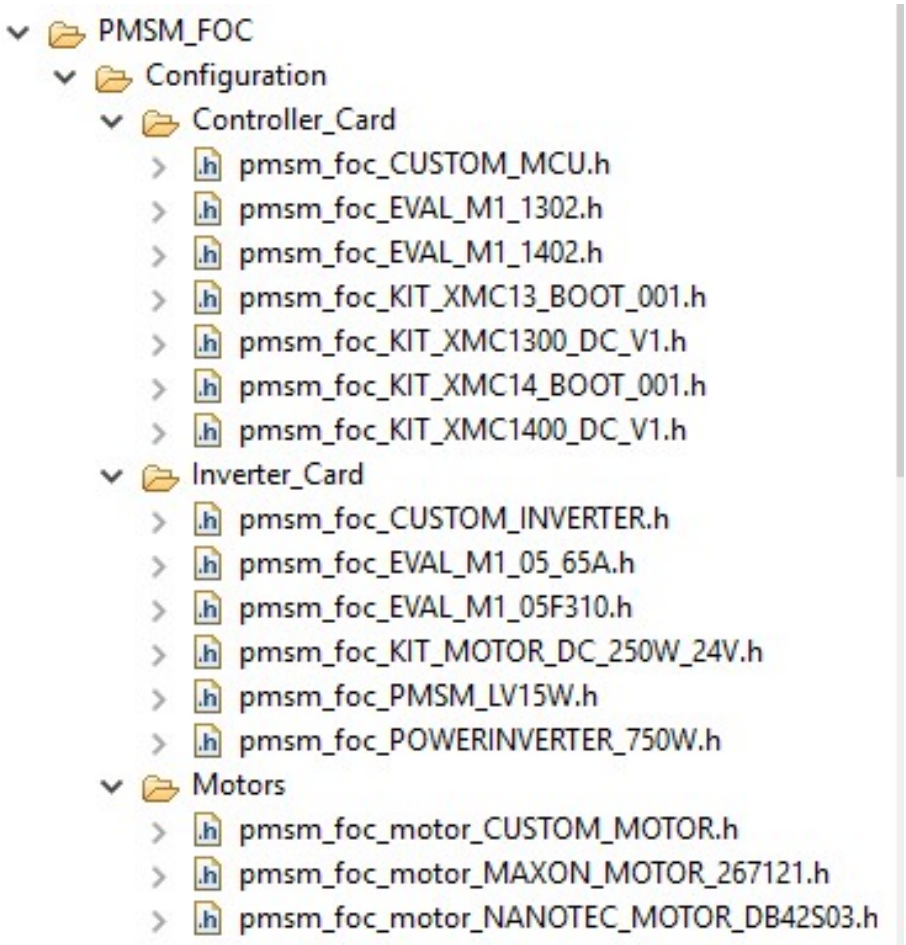
- EVAL_M1_05_65A
- EVAL_M1_05F310
- EVAL_M1_CM610N3
- KIT_MOTOR_DC_250W_24V
- PMSM_LV15W
- POWERINVERTER_750W
- CUSTOM_INVERTER

## Motor Supported

- MAXON_MOTOR_267121
- NANOTEC_MOTOR_DB42S03
- CUSTOM_MOTOR

# Configuration Files of Supported HW

- Customers can reuse predefine configuration and modify them or create a new custom configuration



# Software Overview – Example Configuration

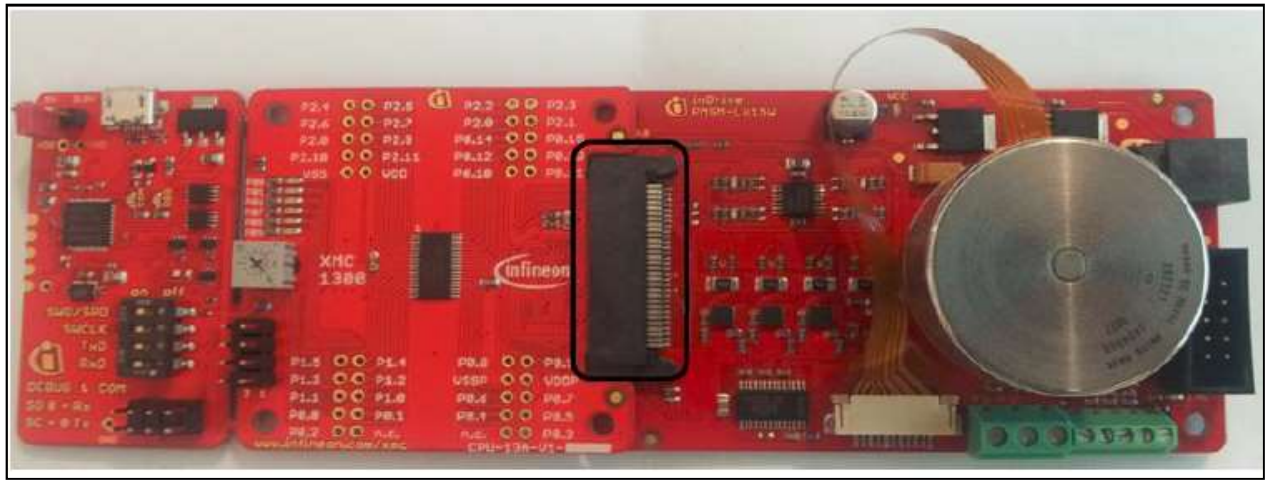
| Example Name | PMSM_FOC_SL_XMC13_PUBBLIC_V1_5_8_LIB |
|---|---|
| Kit Description | Drive 3-phase Maxon's motor using XMC1000 motor control application kit |
| Part Number | KIT_XMC1X_AK_MOTOR_001 |

| Schemes | Default Configuration in Example Software |
|---|---|
| **Control Scheme** | SPEED_CONTROLLED_DIRECT_FOC |
| **PWM frequency (Hz)** | 20000 |
| **Speed (rpm)** | 4200 |
| **Ramp up/down rate** | 500 |
| **Protection** | VDC under/over voltage protection, over current protection |

# Hardware Overview – Application Kit Package

› Infineon's XMC1000 Motor Control Application Kit

| Item | Description |
|---|---|
| XMC1300 CPU Card | MCU board with XMC1300 and detachable SEGGER J-Link debug interface |
| PMSM Low Voltage 15W Motor Card | 12 – 24V<br>Up to 3A<br>On board 3-phase motor (24V, 15W) with hall sensors |
| Accessories | Power Supply Adaptor (24V, 1A)<br>Micro USB connector (1x) |



**XMC1300 CPU Card** **PMSM Low Voltage 15W Motor Card**

# pmsm_foc_user_config.h

- In this file there are a lot of defines to set-up HW/SW configuration:
- First Define is dedicated to Hardware configuration:
  in this case KIT_XMC1X_AK_MOTOR_001

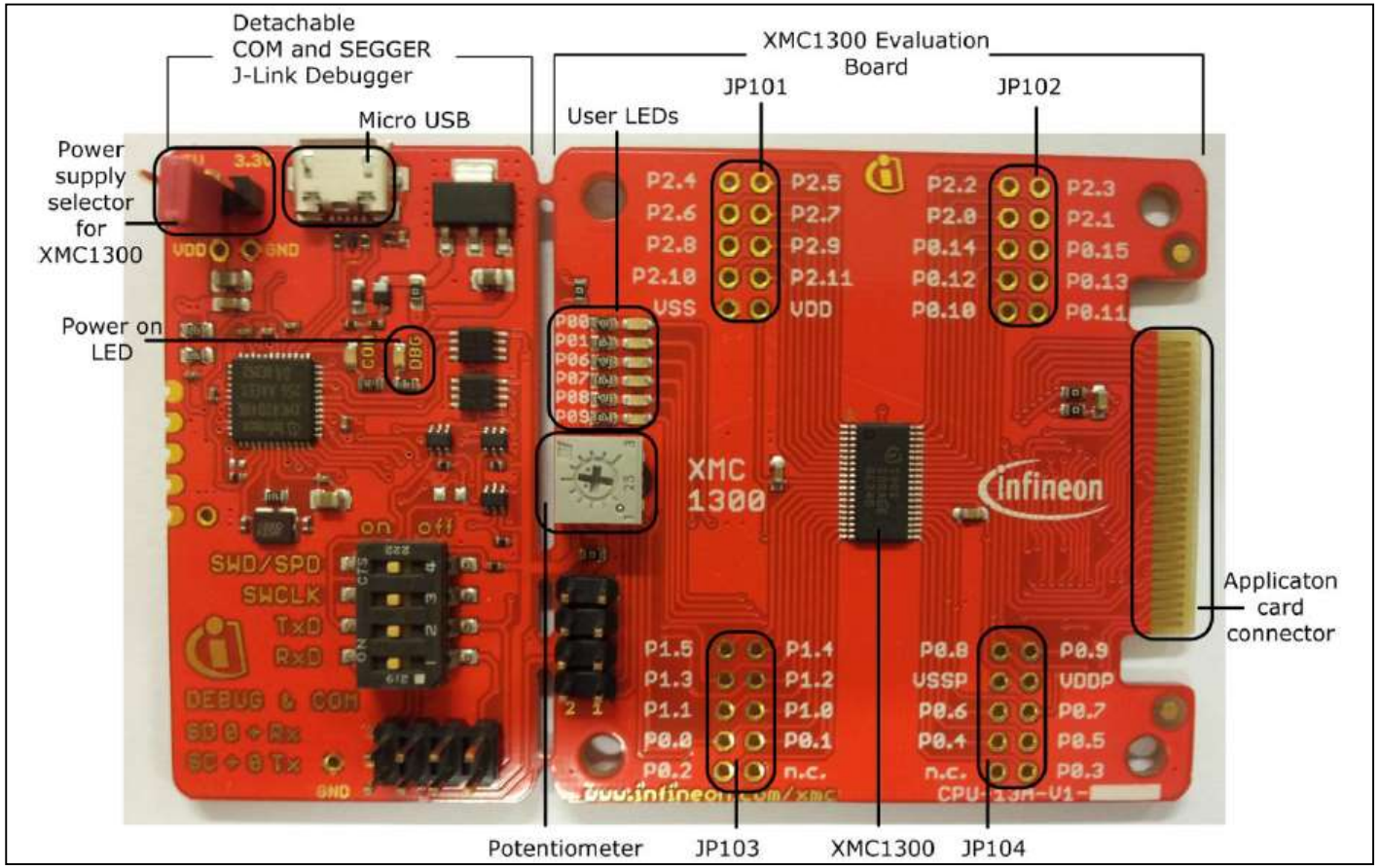
```
78 /*****************************************************************************************************************
79  * MACROS
80  *****************************************************************************************************************/
81 #define PMSM_FOC_HARDWARE_BOARD                    KIT_XMC1X_AK_MOTOR_001              /*1. KIT_XMC1X_AK_MOTOR_001
82                                                                                          2. KIT_XMC750WATT_MC_AK_V1
83                                                                                          3. CUSTOM_KIT*/
```

- In the same file at the end KIT_XMC1X_AK_MOTOR_001 is defined as collection of MCU Board+ Inverter Card + Motor

```
187 #if(PMSM_FOC_HARDWARE_KIT == KIT_XMC1X_AK_MOTOR_001)
188 #define MCUCARD_TYPE                               KIT_XMC13_BOOT_001
189 #define MCUCARD_TYPE_PATH                          "Controller_Card\pmsm_foc_KIT_XMC13_BOOT_001.h"
190 #define INVERTERCARD_TYPE                          PMSM_LV15W
191 #define INVERTERCARD_TYPE_PATH                     "Inverter_Card\pmsm_foc_PMSM_LV15W.h"
192 #define MOTOR_TYPE                                 MAXON_MOTOR_267121
193 #define MOTOR_TYPE_PATH                            "Motors\pmsm_foc_motor_MAXON_MOTOR_267121.h"
194 #elif(PMSM_FOC_HARDWARE_KIT == KIT_XMC750WATT_MC_AK_V1)
195 #define MCUCARD_TYPE                               KIT_XMC1300_DC_V1
196 #define MCUCARD_TYPE_PATH                          "Controller_Card\pmsm_foc_KIT_XMC1300_DC_V1.h"
197 #define INVERTERCARD_TYPE                          POWERINVERTER_750W
198 #define INVERTERCARD_TYPE_PATH                     "Inverter_Card\pmsm_foc_POWERINVERTER_750W.h"
199 #define MOTOR_TYPE                                 CUSTOM_MOTOR
200 #define MOTOR_TYPE_PATH                            "Motors\pmsm_foc_motor_CUSTOM_MOTOR.h"
201 #elif(PMSM_FOC_HARDWARE_KIT == KIT_XMC_IFI_24V_250W)
202 #define MCUCARD_TYPE                               KIT_XMC1300_DC_V1
203 #define MCUCARD_TYPE_PATH                          "Controller_Card\pmsm_foc_KIT_XMC1300_DC_V1.h"
204 #define INVERTERCARD_TYPE                          KIT_MOTOR_DC_250W_24V
```

# Hardware Overview – XMC1300 CPU Card

› XMC1300 CPU Card



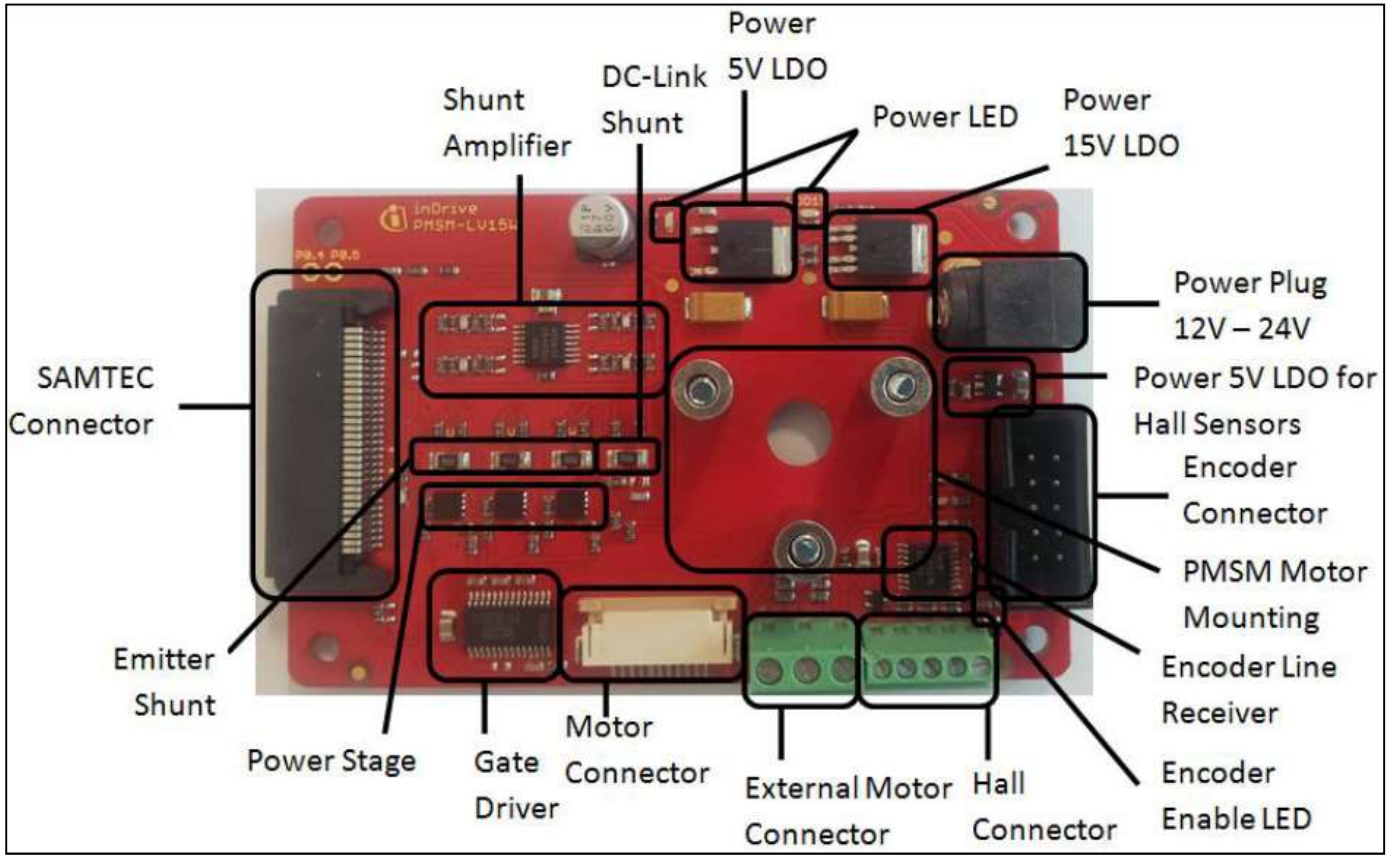
# pmsm_foc_KIT_XMC13_BOOT_001.h

- This files contains defines related to MCU card
  - PWM output pins
  - ADC input
  - Internal hw resources

```
522 /*****************************************************************************************************************
523  * KIT_XMC1X_AK_MOTOR_001
524  * GPIO Resources Configuration
525  *****************************************************************************************************************/
526 #define TRAP_PIN                P0_12
527 #define INVERTER_EN_PIN         P0_11
528
529 #define PHASE_U_HS_PIN          P0_0
530 #define PHASE_U_HS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
531
532 #define PHASE_U_LS_PIN          P0_1
533 #define PHASE_U_LS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
534
535 #define PHASE_V_HS_PIN          P0_7
536 #define PHASE_V_HS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
537
538 #define PHASE_V_LS_PIN          P0_6
539 #define PHASE_V_LS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
540
541 #define PHASE_W_HS_PIN          P0_8
542 #define PHASE_W_HS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
543
544 #define PHASE_W_LS_PIN          P0_9
545 #define PHASE_W_LS_ALT_SELECT   XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5
546
547 #define TEST_PIN           P0_4
548
549 /*****************************************************************************************************************
```

# Hardware Overview – Motor Card

› PMSM Low Voltage 15W Motor Card



# pmsm_foc_PMSM_LV15W.h

- This file contains all parameters related to inverter board:
  - Vdc Link
  - Dead time
  - Driver delay
  - Op-amp gain
  - Shunt resistor values
  - Polarity of Driver, Trap
  - …

```
/* ************************************** PMSM_LV15W ********************************************************************************
#if(INVERTERCARD_TYPE == PMSM_LV15W)
#define INTERNAL_OP_GAIN                        DISABLED              /*1. ENABLED       2. DISABLED (Please configure OP-Gain manually) */
#define USER_VDC_LINK_V                         (24.0f)                /* Hardware Inverter VDC link voltage in V  */
#define USER_DEAD_TIME_US                       (0.75f)                /* deadtime, rise(left) and fall values in us  */
#define USER_CCU8_PWM_FREQ_HZ                   (20000U)              /* CCU8 PWM Switching Frequency in Hz*/
#define USER_DRIVERIC_DELAY_US                  (1.0f)                 /*  Driver IC delay. It affects ADC trigger points during high speed as
#define USER_BOOTSTRAP_PRECHARGE_TIME_MS        (20U)                 /* Initial Bootstrap precharging time in ms */
#define USER_DC_LINK_DIVIDER_RATIO              (float)(5.1f/(5.1f+47.0f))          /* R1/(R2+R1) ratio for DC link MCU ADC */
#define USER_VBEMF_RATIO                        (float)(5.2f/(5.2f+47.0f))          /* R1/(R2+R1) ratio for BEMF Voltage sensing circuit rati
#define USER_CURRENT_TRIP_THRESHOLD_A           (3.0f)               /* threshold current for trip detection in Ampere*/
#define USER_TRIP_THRESHOLD_TIME_MS             (100U)               /* threshold time for trip detection in ms */
#define USER_MAX_RETRY_MOTORSTARTUP_TRIP        (3U)                /* Max retry of motor startup if trip  */
/*      ------------------------------------------------- Motor Phase Current Measurement ---------------------------------------- */
#define USER_R_SHUNT_OHM                        (0.05f)                /* Phase shunt resistor in ohm */
#define USER_DC_SHUNT_OHM                       (0.05f)                /* DC link shunt current resistor in ohm */
#define USER_RIN_PHASECURRENT_KOHM              (1.0f)                 /* R_IN (of equivalent amplifier) kohm */
#define USER_R_PHASECURRENT_FEEDBACK_KOHM       (16.4f)                  /* R_FEEDBACK (of equivalent amplifier) kohm */
#define USER_RIN_DCCURRENT_KOHM                 (10.0f)                  /* Rf for dc current sensing */
#define USER_R_DCCURRENT_FEEDBACK_KOHM          (75.0f)                   /* Rin for dc current sensing */
#define USER_MAX_ADC_VDD_V                      (5.0f)                   /* VDD5, maximum voltage at ADC */
#define G_OPAMP_PER_PHASECURRENT                (USER_R_PHASECURRENT_FEEDBACK_KOHM / USER_RIN_PHASECURRENT_KOHM)
#define I_MAX_A                                 ((VAREF_V/(USER_R_SHUNT_OHM * OP_GAIN_FACTOR)) / 2U)              /* For IFX_XMC_LVPB_R3, I_M
```

# Hardware Overview – MAXON MOTOR

› MAXON MOTOR 267121

| | Motor Data | | 267121<br>226006 |
|---|---|---|---|
| | **Values at nominal voltage** | | |
| 1 | Nominal voltage | V | 24 |
| 2 | No load speed | rpm | 4530 |
| 3 | No load current | mA | 36.9 |
| 4 | Nominal speed | rpm | 2760 |
| 5 | Nominal torque (max. continuous torque) | mNm | 25.5 |
| 6 | Nominal current (max. continuous current) | A | 0.5 |
| 7 | Stall torque | mNm | 85.8 |
| 8 | Starting current | A | 1.75 |
| 9 | Max. efficiency | % | 74 |
| | **Characteristics** | | |
| 10 | Terminal resistance phase to phase | Ω | 13.7 |
| 11 | Terminal inductance phase to phase | mH | 7.73 |
| 12 | Torque constant | mNm/A | 49 |
| 13 | Speed constant | rpm/V | 195 |
| 14 | Speed/torque gradient | rpm/mNm | 54.5 |
| 15 | Mechanical time constant | ms | 20 |
| 16 | Rotor inertia | gcm² | 35 |

# pmsm_foc_motor_MAXON_MOTOR_267121.h

- In this file are stored default motor parameters for our evaluation board ad customer can define own motors.
- L & R parameters are used to adjust some PI parameters automatically
- PI paramters, like Ki, Kp, scaling e aniwindup values are stored here.

```
394 #elif (MOTOR_TYPE == MAXON_MOTOR)
395 /*      ------------------------------------------------- Motor Parameters ---------------------------------------- */
396 #define  USER_MOTOR_R_PER_PHASE_OHM               (6.8f)        /* Motor Resistance per phase in Ohm*/
397 #define  USER_MOTOR_L_PER_PHASE_uH                (3865.0f)         /* Motor Inductance per phase in uH */
398 #define  USER_MOTOR_POLE_PAIR                     (4.0f)          /* Motor Pole Pairs */
399 /*      ------------------------------------------------- Constant Speed Control Mode (Used when Constant Speed Control is enabled) --------------
400 /*      ------------------------------------------------- POT ADC, or PWM to Adjust Speed ------------------------------------- */
401 #define USER_SPEED_HIGH_LIMIT_RPM                (4530.0f)
402 #define USER_SPEED_LOW_LIMIT_RPM                 (uint32_t) (USER_SPEED_HIGH_LIMIT_RPM / 30U)
403 #define USER_SPEED_RAMPUP_RPM_PER_S              (500U)
404 #define USER_SPEED_RAMPDOWN_RPM_PER_S            (500U)
405 #define USER_RATIO_S                             (1U)
406 /*      ------------------------------------------------- V/F Start Up Parameters --------------------------------------- */
407 #define USER_STARTUP_SPEED_RPM                   (0U)
408 #define USER_STARTUP_SPEED_THRESHOLD_RPM         (200U)            /* threshold Speed to transit from Open loop to closed loop */
409 //#define USER_STARTUP_VF_OFFSET_V                (float) (USER_VDC_LINK_V * 0.05f)                                  /* V/F startup offset in V *
410 //#define USER_STARTUP_VF_SLEWRATE_V_PER_HZ       (float) (USER_VDC_LINK_V /ELECTRICAL_SPEED_FREQ_HZ)
411 #define USER_STARTUP_VF_OFFSET_V                 (1.0f)            /* V/F startup offset in V */
412 #define USER_STARTUP_VF_SLEWRATE_V_PER_HZ        (0.1f)           /* V/F start up slew rate in V/Hz */
413
```

# Agenda

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

# SW Overview – XMC1302 Peripheral usage (1/2)

| No | Category | Description | XMC1302 Pins | Remark |
|---|---|---|---|---|
| 1 | Motor Phase U | High side driver Phase U MOSFET | **P0.0** / CCU80.OUT00 | Active level - **LOW** |
| 2 | | Low side driver Phase U MOSFET | **P0.1** / CCU80.OUT01 | |
| 3 | Motor Phase V | High side driver Phase V MOSFET | **P0.7** / CCU80.OUT10 | |
| 4 | | Low side driver Phase V MOSFET | **P0.6** / CCU80.OUT11 | |
| 5 | Motor Phase W | High side driver Phase W MOSFET | **P0.8** / CCU80.OUT20 | |
| 6 | | Low side driver Phase W MOSFET | **P0.9** / CCU80.OUT21 | |
| 7 | Inverter Enable | Enable gate driver I/O functionality | **P0.11** | Active **LOW** |
| 8 | DC Link Current | Amplifier output for DC link single shunt | **P2.7** / G1.CH1 | |
| 9 | DC Link Voltage | Voltage of DC link (with voltage divider) | **P2.3** / G1.CH5 | Divider resistors 990K/10K |
| 10 | POT | ADC for potentiometer | **P2.5** / G1/CH7 | |

# SW Overview – XMC1302 Peripheral usage (2/2)

| No | Category | Description | XMC1302 Pins | Remark |
|---|---|---|---|---|
| 11 | 3-Shunt Phase Current | Amplifier output for Phase U shunt | **P2.9** / (G0.CH2/G1.CH4) | 3-shunt 50 mΩ, with Op-Amp gain |
| 12 | | Amplifier output for Phase V shunt | **P2.10** / (G0.CH3/G1.CH2) | |
| 13 | | Amplifier output for Phase W shunt | **P2.11** / (G0.CH4/G1.CH3) | |

# FOC control Scheme

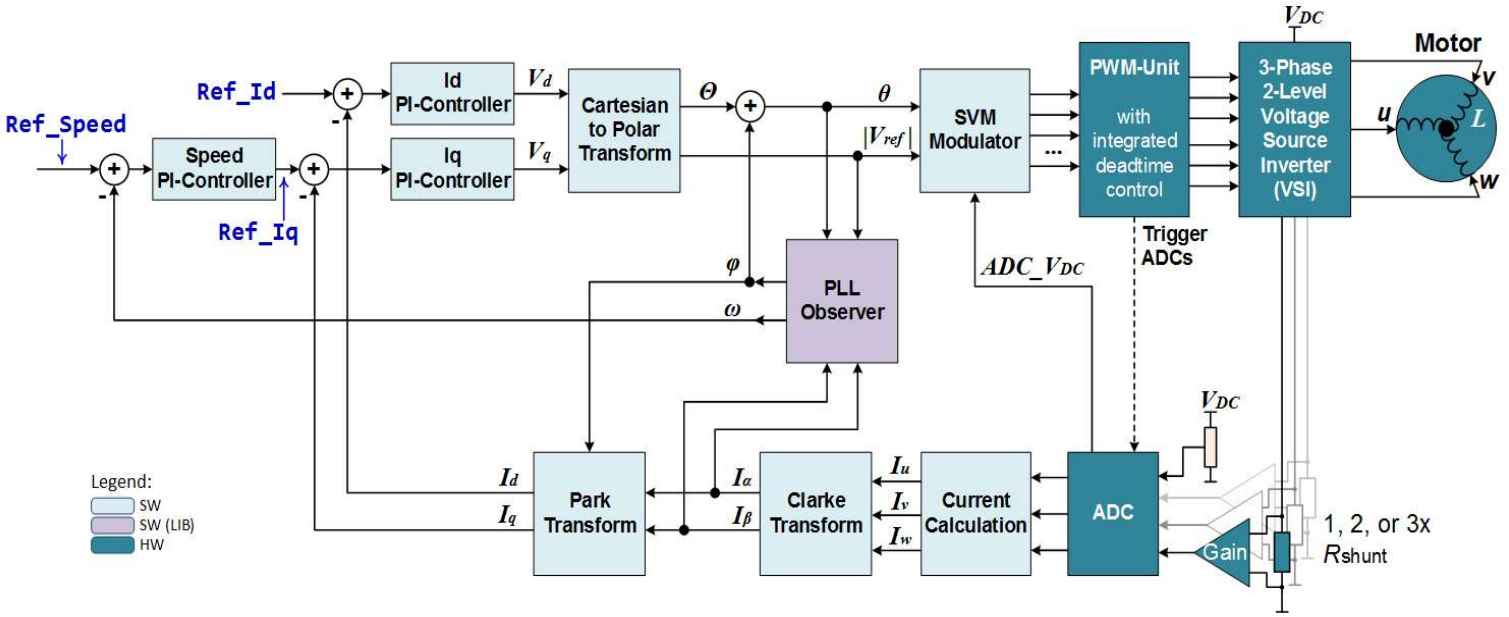


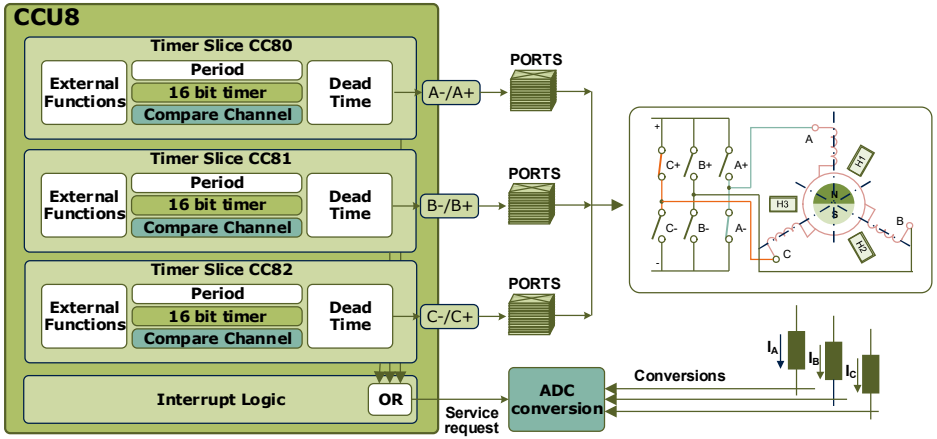

**Table 5 CPU utilization and memory usage for three shunt current sensing with XMC1300 and XMC1400**

| | | |
|---|---|---|
| **PWM frequency** | 20 kHz – Interrupt Service Routine runs every 50 µsec | |
| **DAVE™ 4 GCC compiler optimization level** | Optimized most (-O3) | |
| **MCU** | **XMC1300** | **XMC1400** |
| **CPU utilization** | 31 µsec (62%) | 21 µsec (42%) |
| **Flash code size (bytes)** | 10792 | 11148 |
| **SRAM code size (bytes)** | 348 | 352 |
| **SRAM data size (bytes)** | 1720 | 1716 |

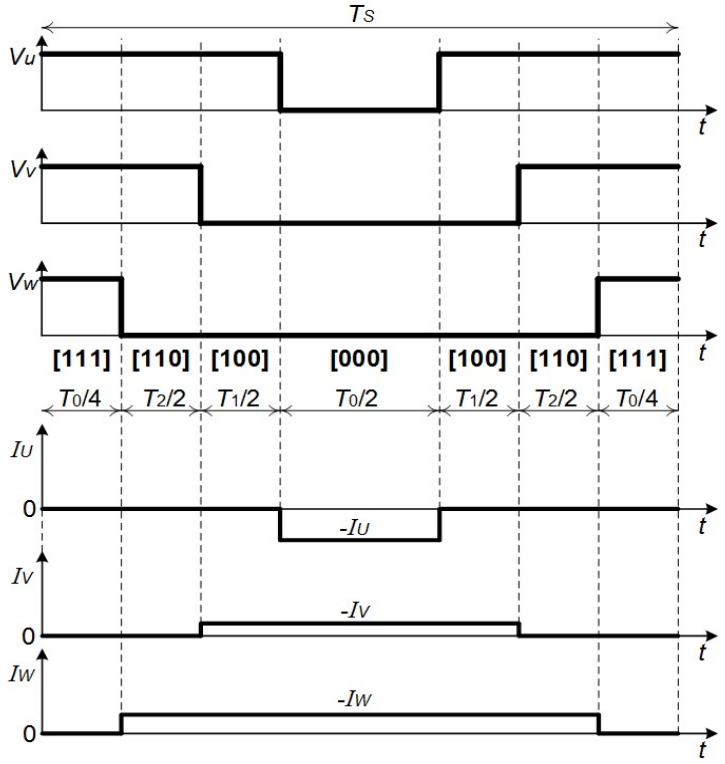
# Hardware Interconnection

## Interconnection between CCU8 (SVM PWM generation) with VADC

- To measure shunt currents in each PWM cycle
- CCU8 interrupt logic provides flexibility to group several triggers to one SR line




Generate **ADC conversion triggers** synchronized with PWM signal



# Current sensing Synchronous

- XMC1x has 2 S/H in parrallel, then to optimize current reading «alias»feature is used in order to sample only 2 current based on SVM sector

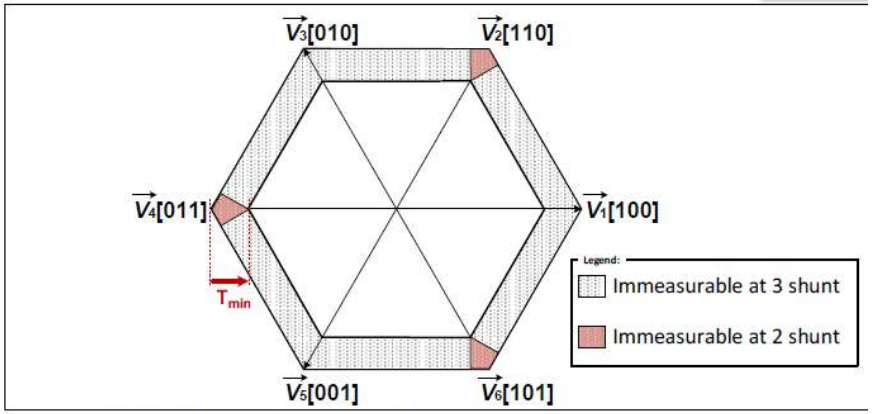
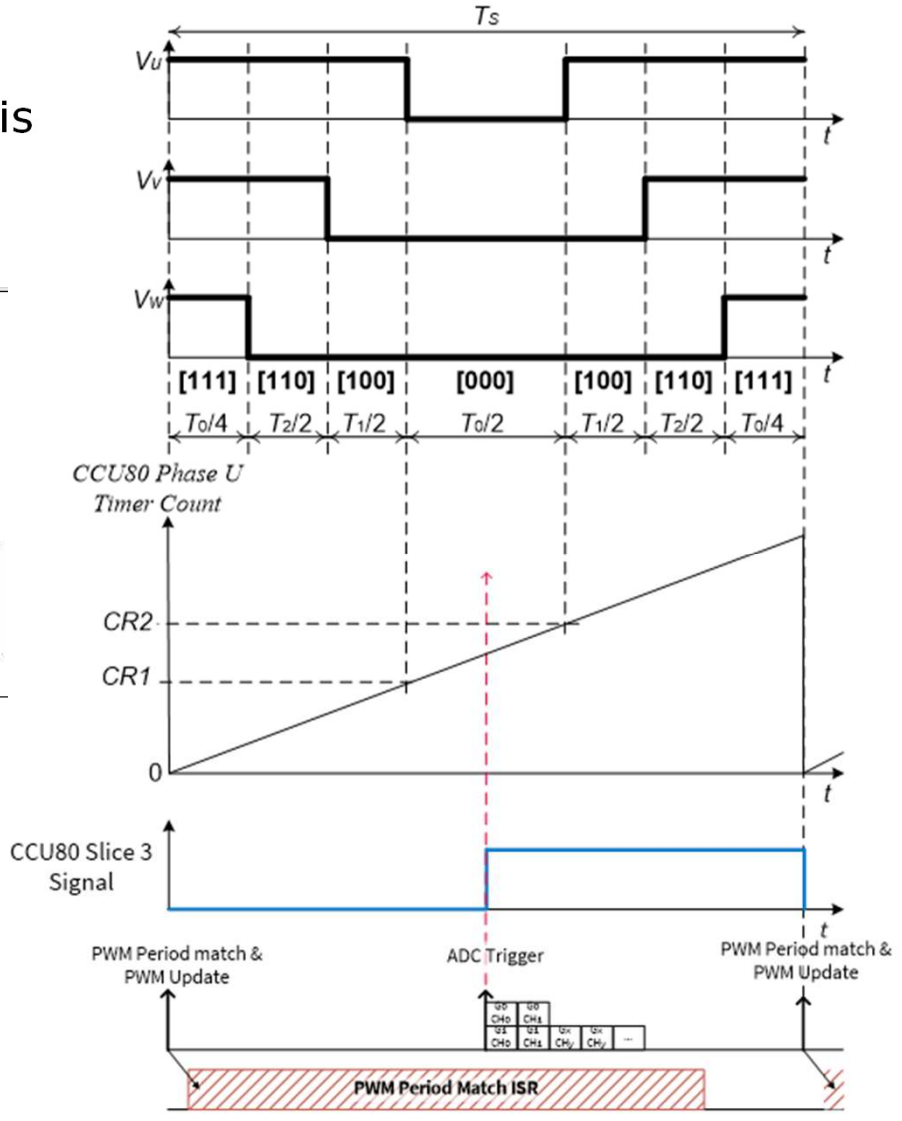



Figure 44 SVM 2/3 leg shunt immeasurable areas

**Table 14 Phase measurement per SVM sectors**

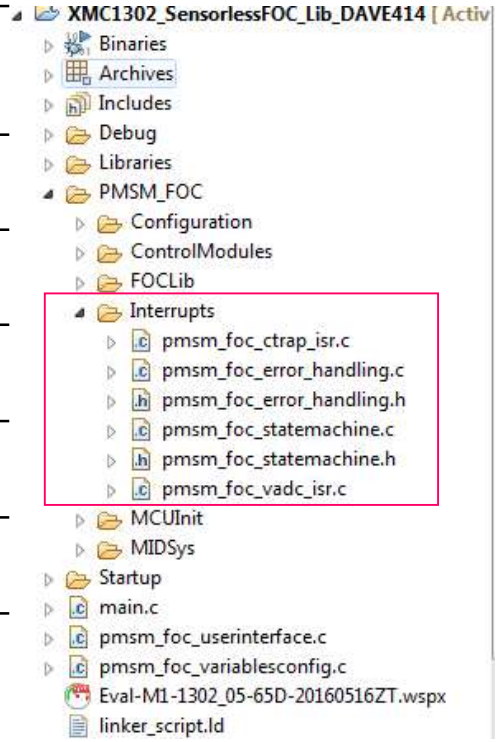
| SVM sectors | Shunt current measured |
|---|---|
| Sector A and F | Phase W |
| | Phase V |
| Sector B and C | Phase U |
| | Phase W |
| Sector D and E | Phase V |
| | Phase U |



# Software Overview - Interrupt Service Routines

**Folder: Interrupts**

| Peripheral | Interrupt Subroutines (ISR) | NVIC node | Interval | Priority |
|---|---|---|---|---|
| VADC | VADC_Source_IRQHandler | 18 | Asynchronous (only single shunt) | 1 |
| CCU8 | CTRAP | 26 | Asynchronous | 0 |
| CCU8 | One match event (Phase U) | 25 | 1/ PWM frequency | 2 |
| CCU4 | Period Match | 21 | Secondary Loop Freq., Timer starts sync to CCU8 | 3 |
| VADC | Vdc boudary | 19 | Vdc Link, over/under voltage | 1 |



# Secondary Loop - Callback function

- Callback function can be enable or disable by changing define in configuration files
  - Secondary function by default is 1Khz →1ms
- Customers can write inside their own code without impact on algorithm
- Start & Stop motor in the example are execute in callback function into the main.c together with Vdc_link reading.

```
void pmsm_foc_secondaryloop_callback(){

  if (motor_request_start & motor_off)
  {
    pmsm_foc_motor_start();
    motor_off = false;
  }
  else if (!motor_request_start & !motor_off )
  {
    pmsm_foc_motor_brake();
    motor_off = true;
  }

  Vdc_link = pmsm_foc_get_Vdc_link();


}
```

# Agenda

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

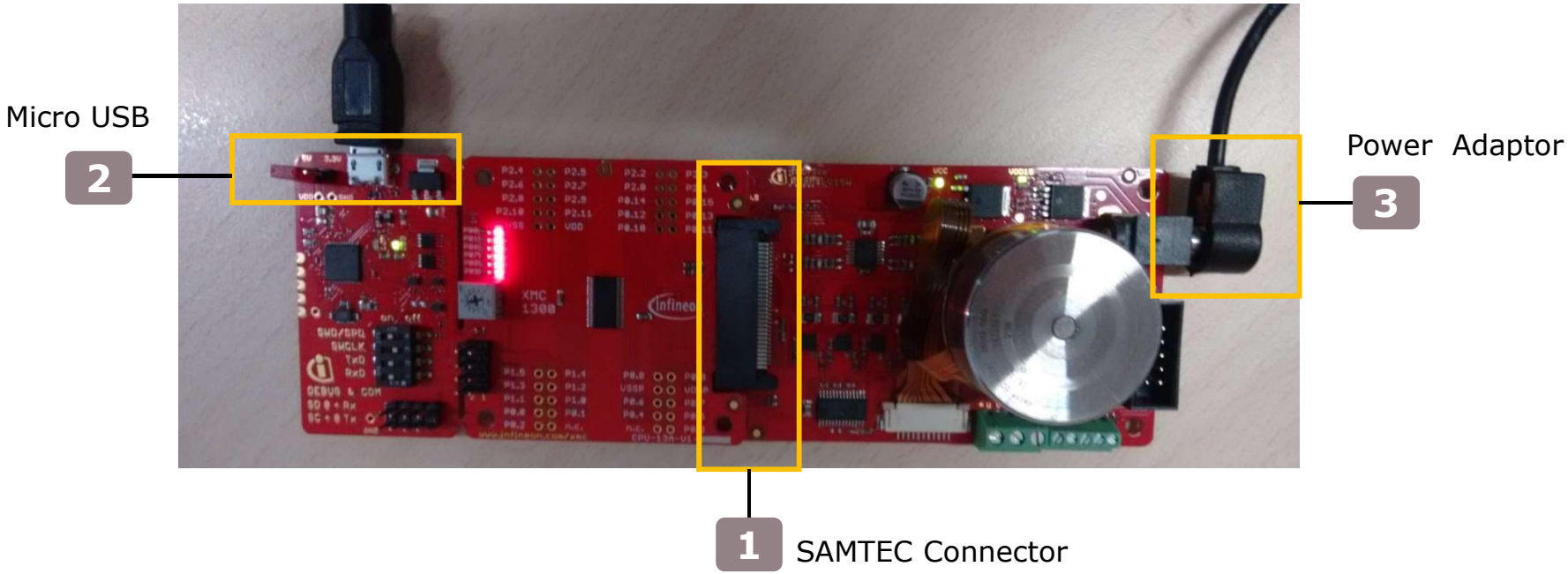
# Tools Overview

- DAVE™ (V4.4.2 onwards)
  - Download DAVE™ installer package from

    http://www.infineon.com/dave
  - Download and unzip the installer package

# Getting Started – Connecting the Board

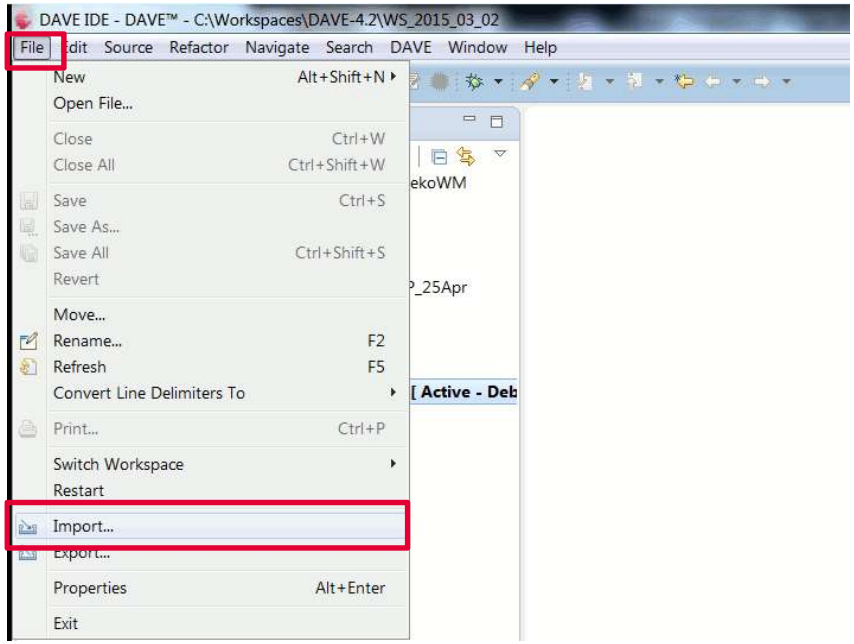
1. Connect XMC1300 CPU Card to PMSM Low Voltage 15W Motor Card using SAMTEC connector interface
2. Connect XMC1300 CPU Card to PC via Micro USB cable
3. Connect power adaptor to PMSM Low Voltage 15W Motor Card

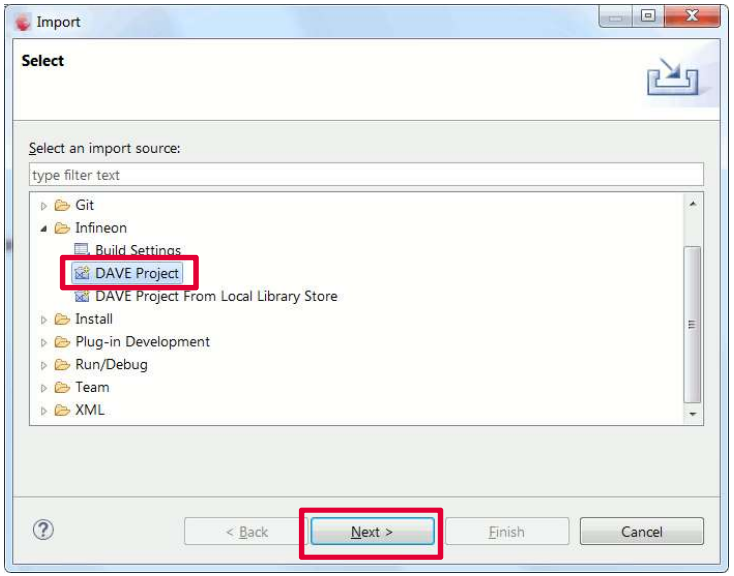


# Getting Started – Download Project from DAVE [1/2]

1. Open DAVE™ 

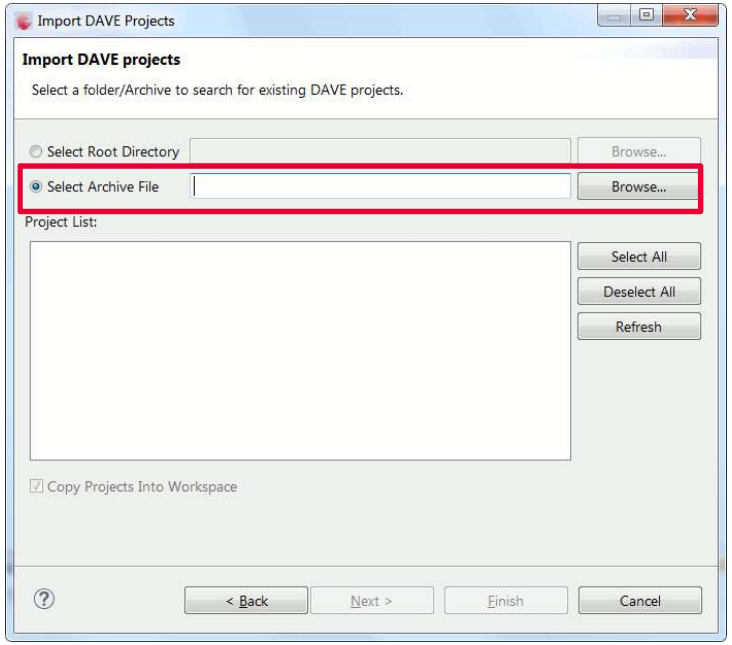
2. Click on **File > Import** to import sample code
3. Select **Infineon > DAVE project** and click "**Next**"

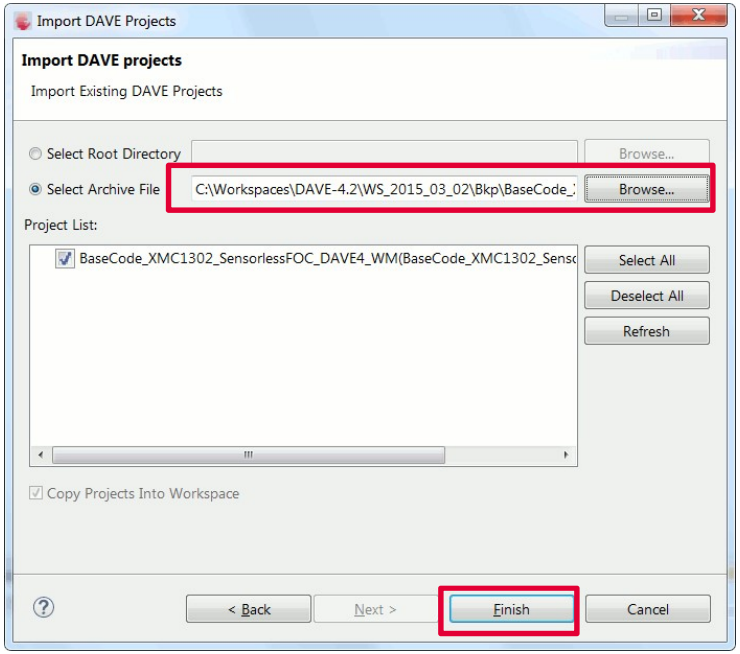


# Import FOC example SW to DAVE™ 4 (2/2)

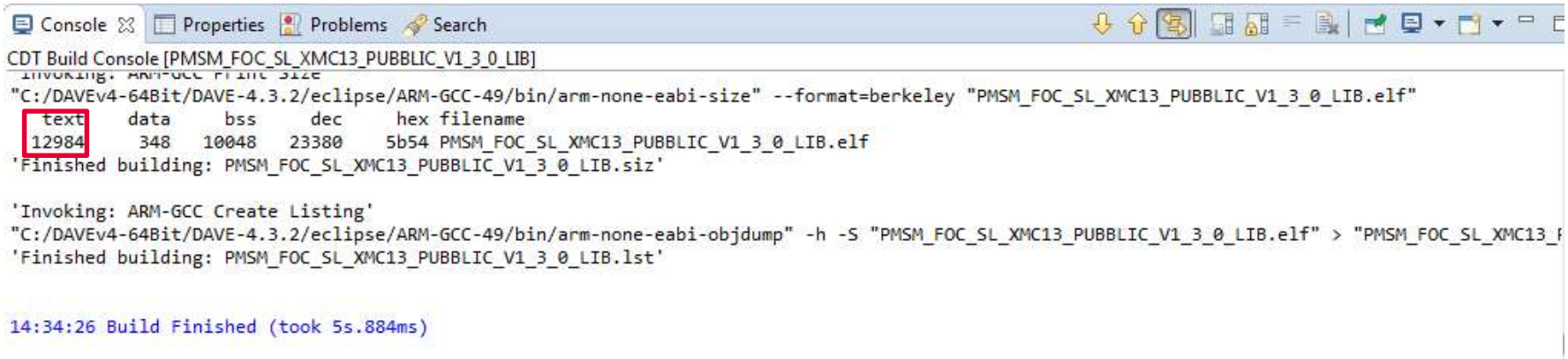
- Next click on **Select Archive File > Browse**.
- Select the folder containing the sample code and click "**OK**".
- Click on "**Finish**" to import the code into DAVE™ 4.



# Build FOC example SW in DAVE™ 4

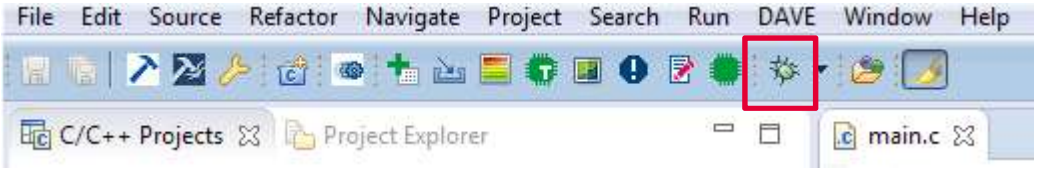
› Click "**Rebuild Active Project**"

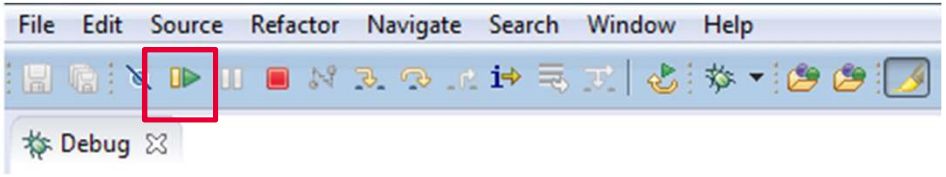

› "text" in red box indicates that code size

# Download FOC example SW in DAVE™ 4

- Click "**Debug Configuration**" to download the code




- Click "**Resume**" to start the motor control application



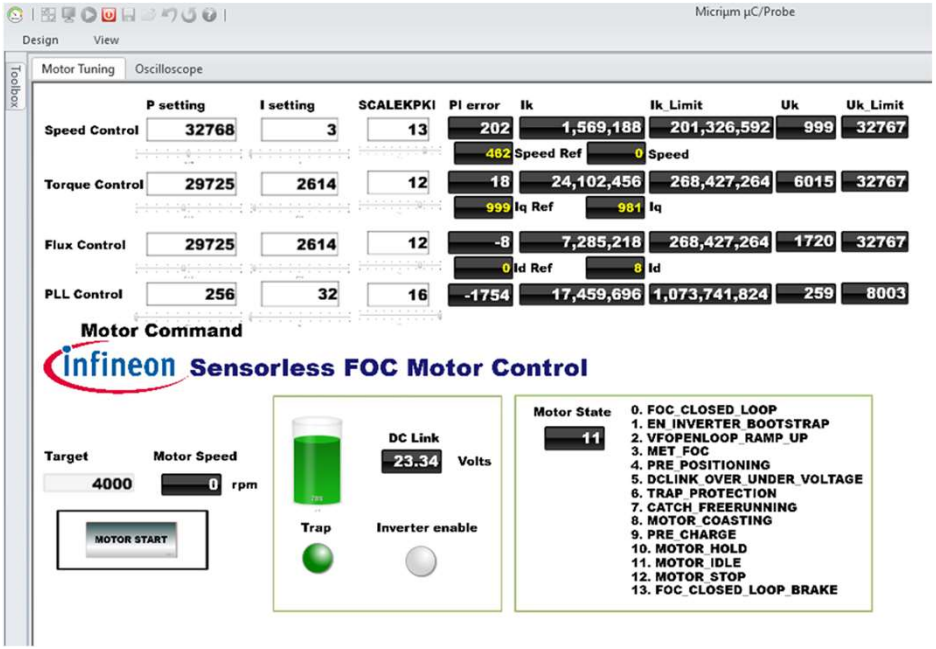

- Motor is not spinning

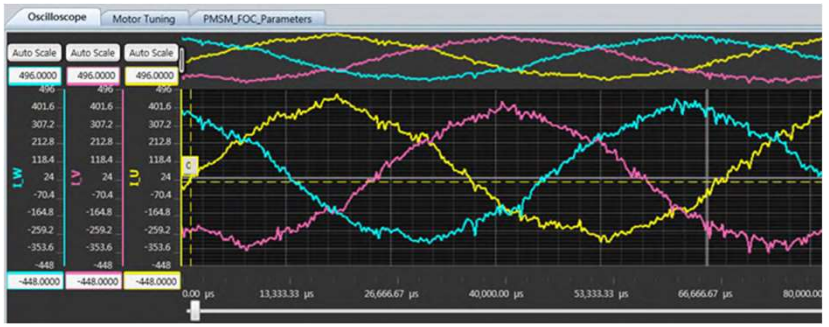
# Agenda

1. Overview
2. Example project details
3. XMC HW/SW implementation
4. Import DAVE prj and Download
5. Run motor with ucProbe GUI interface

# Uc/Probe

- The µC/Probe is a Windows application designed to read and write the memory of the embedded target processor during run-time. Memory locations are mapped to a set of virtual controls and indicators placed on a dashboard
- The µC/Probe is used to monitor the motor parameters. It can also be used to fine-tune the PI gains to get the optimum motor behavior. To start the motor, first click 'MOTOR START' button and then drag the 'Setting Target Speed' slider to the right.



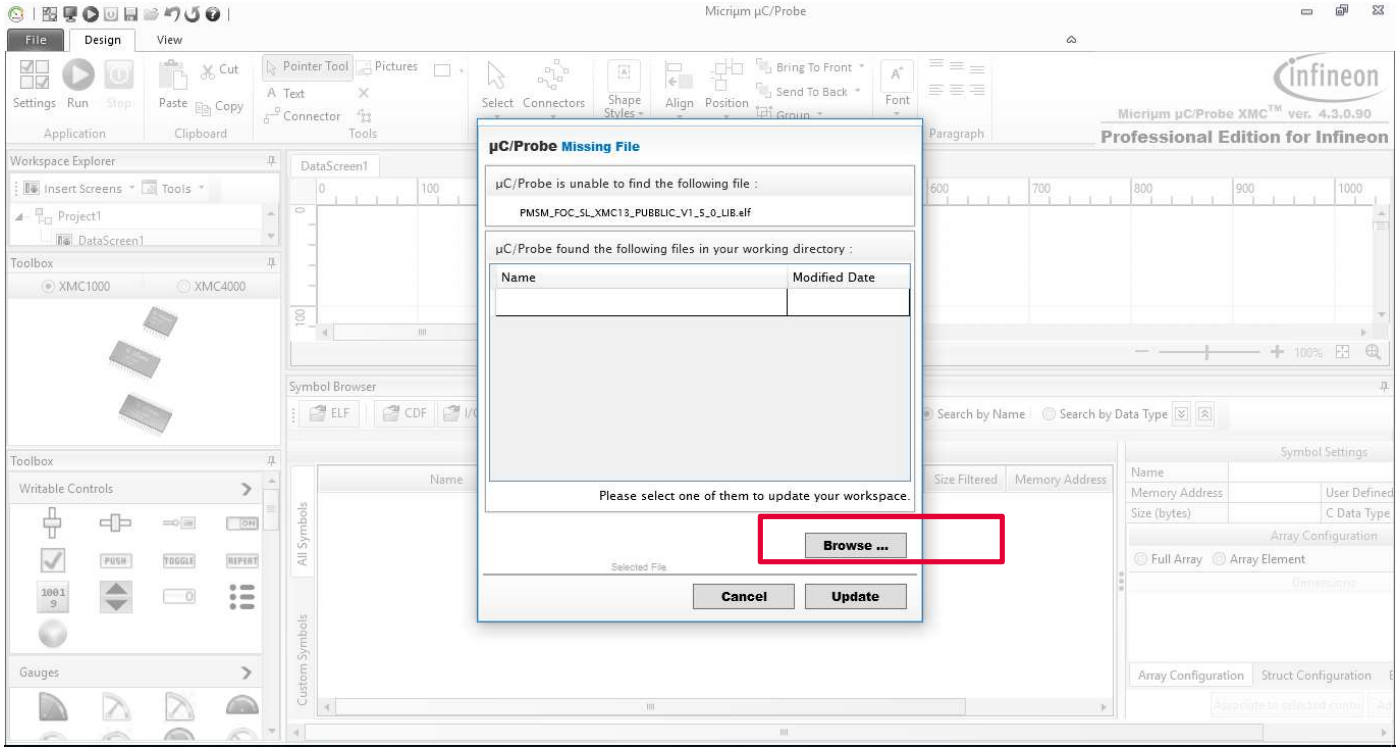
# Overview

µc/Probe is a real-time monitor and data visualization stand alone PC tool

- ☑ Easy out of the box non-intrusive variable monitoring
- ☑ Synchronous variable monitoring
- ☑ Connects to target using SWD, UART, USB or Ethernet
- ☑ Symbol browser: parse ELF file for variable selection
- ☑ Design rich graphical interfaces to control your application
- ☑ Oscilloscope
- ☑ Scripting
- ☑ Log captured data to a file
- ☑ Save/restore session

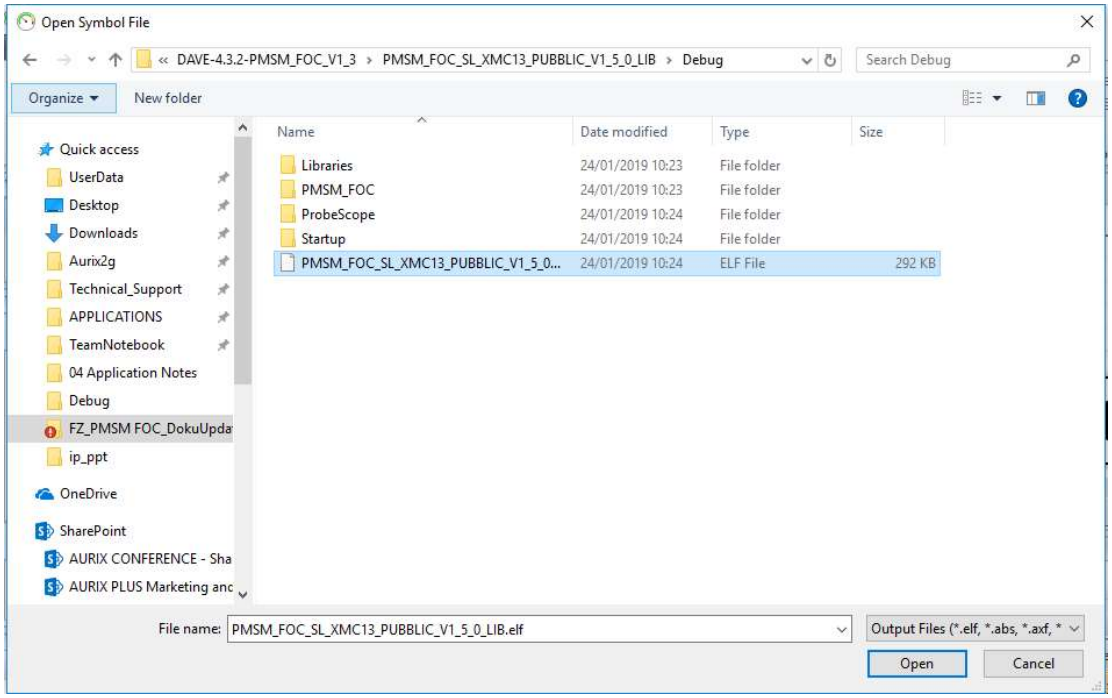
# Open .WSPX

- Open “PMSM_FOC_SL_XMC1_uCProbe.wspx” with uc/Probe
  - This file in root folder of PMSM_FOC_SL_XMC13_PUBBLIC_V1_5_8_LIB
- If ucprobe don’t find .elf file, pop-up will be open and we press “Browse” to find it



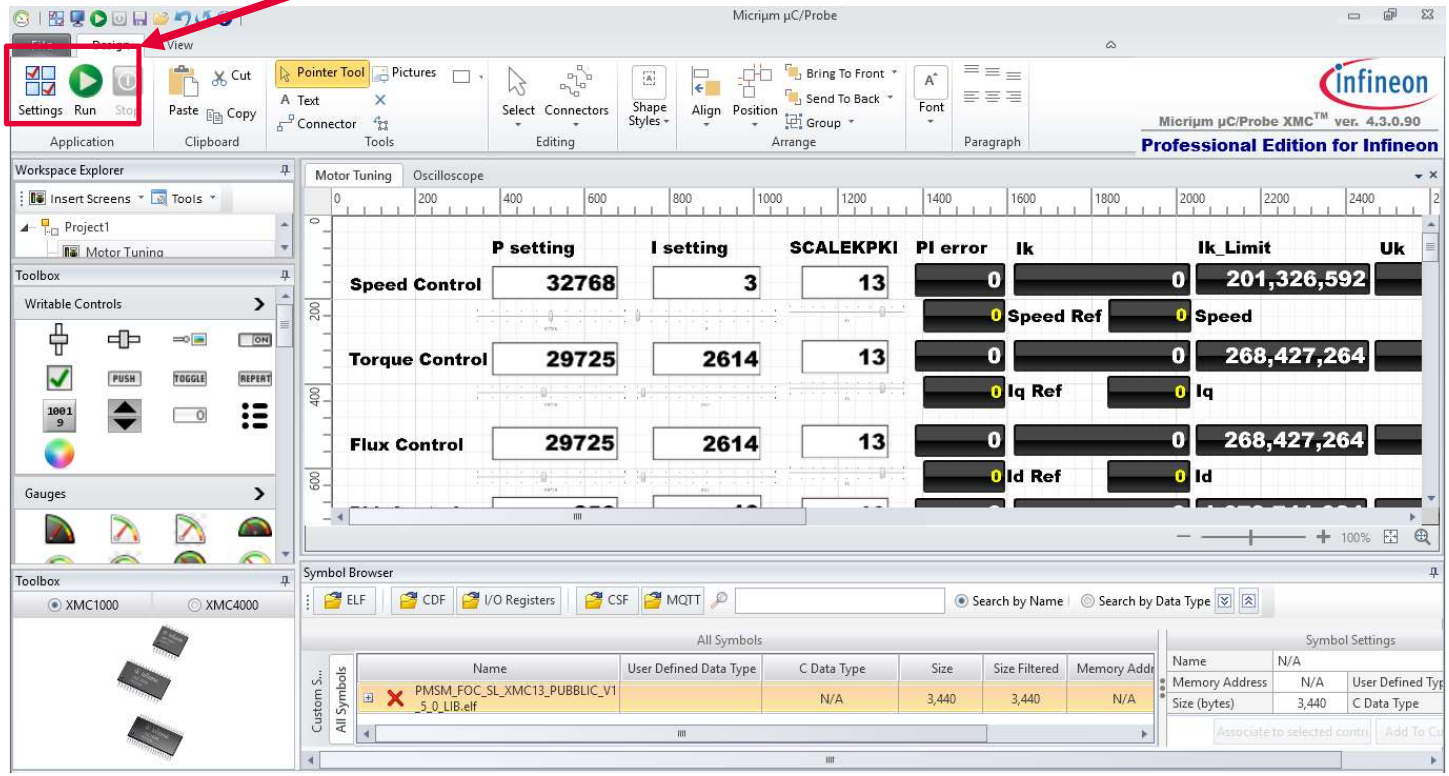
# Open .elf

- .elf file is in Debug folder of Dave Project
- ..\PMSM_FOC_SL_XMC13_PUBBLIC_V1_5_8_LIB\Debug\
  PMSM_FOC_SL_XMC13_PUBBLIC_V1_5_8_LIB.elf



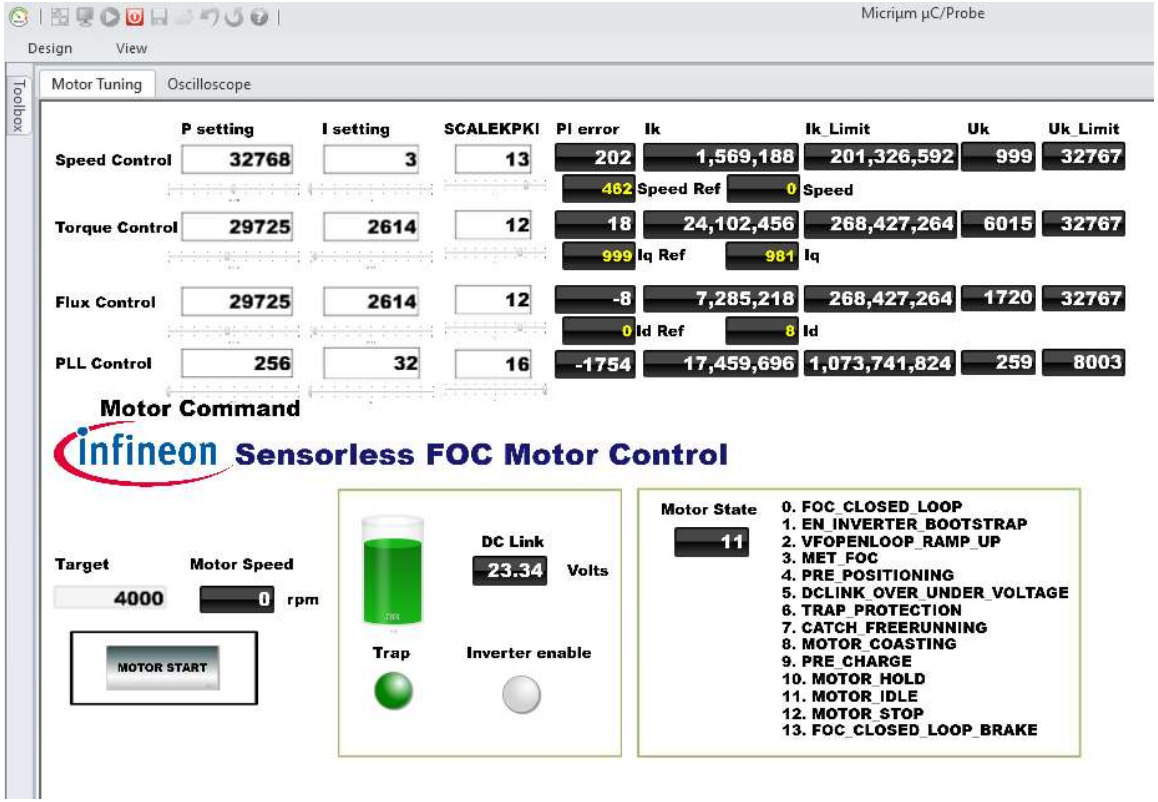
# ucProbe Design view

- This view allow customization of GUI interface
  - To start motor we need to press «RUN» button to switch to running mode



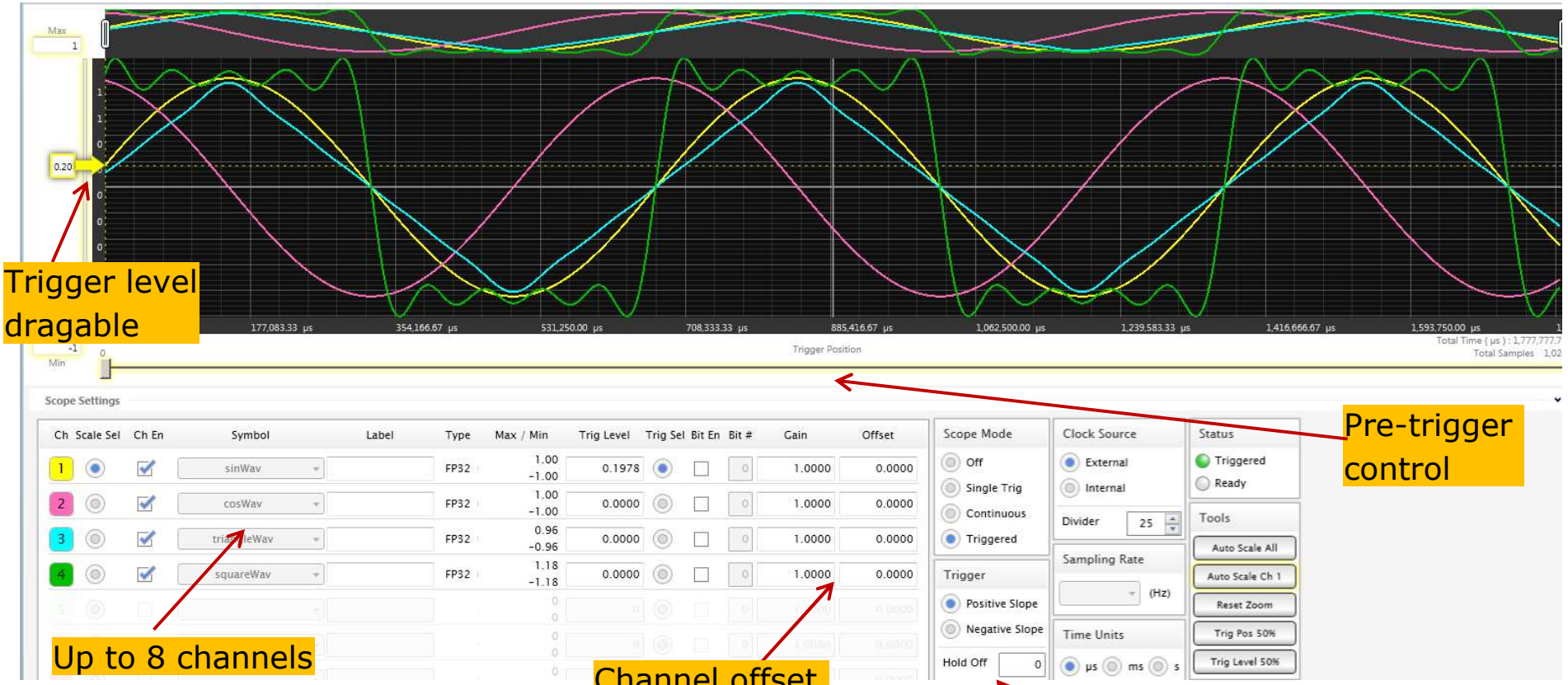
# ucProbe Run view

1. Press «Motor Start», then motor starts spinning
2. Change Tagert speed to increase morot speed
3. Play with PI value to fine tuning the motor beahviours



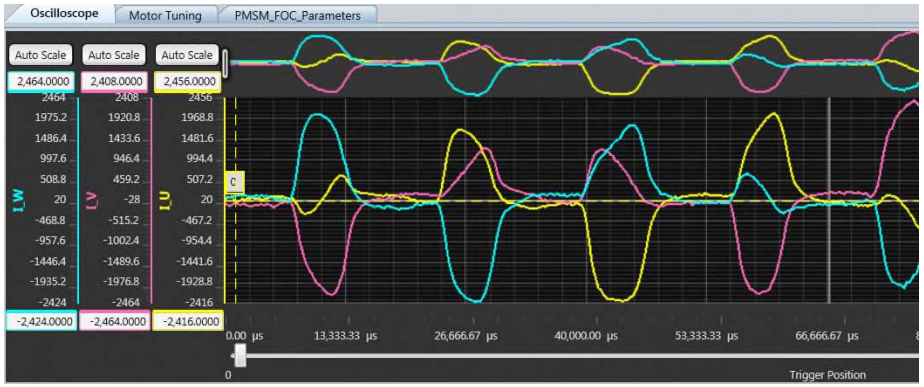
# Features
## Oscilloscope

- Requires application changes and it is minimal intrusive
- Continuous and triggered sampling mode
- Synchronous and asynchrnous control loop sampling mode

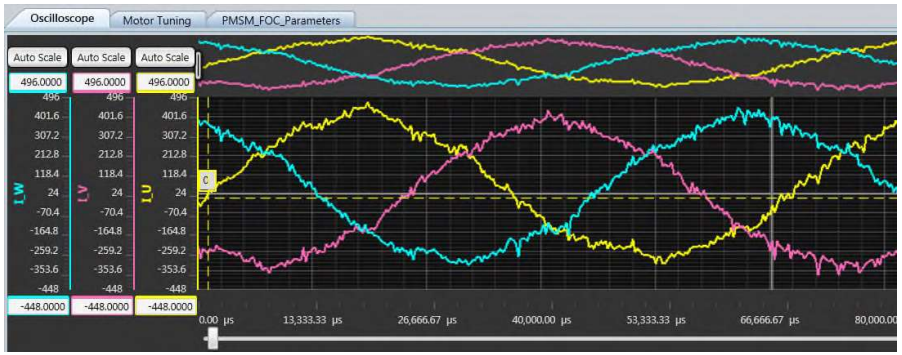


# Tuning of Kp, Ki value using the µC/Probe

- Adjust P and I setting of PLL Control/Speed Control for finer tuning of Motor behaviour. The final goal is to achieve **sinusoidal** current waveform in the Oscilloscope page of the µC/Probe GUI.




The P, I, SCALEKPKI values are not optimized or fine-tuned, so the 3 motor currents I_U, I_V, I_W are **not sinusoidal**

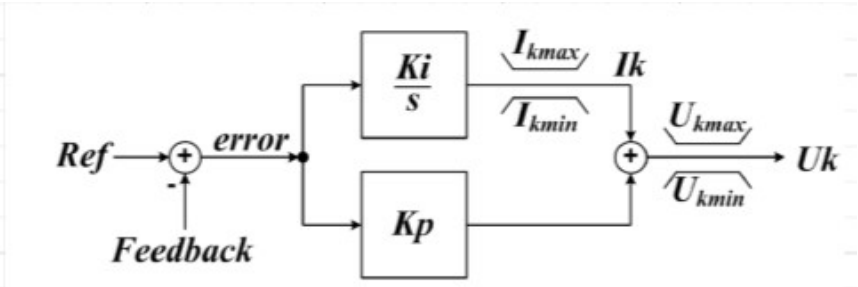



Target of tuning the P, I and SCALEKPKI values of the control loops is to achieve the target motor performance and get **sinusoidal** current shapes for the 3 motor currents I_U, I_V, I_W

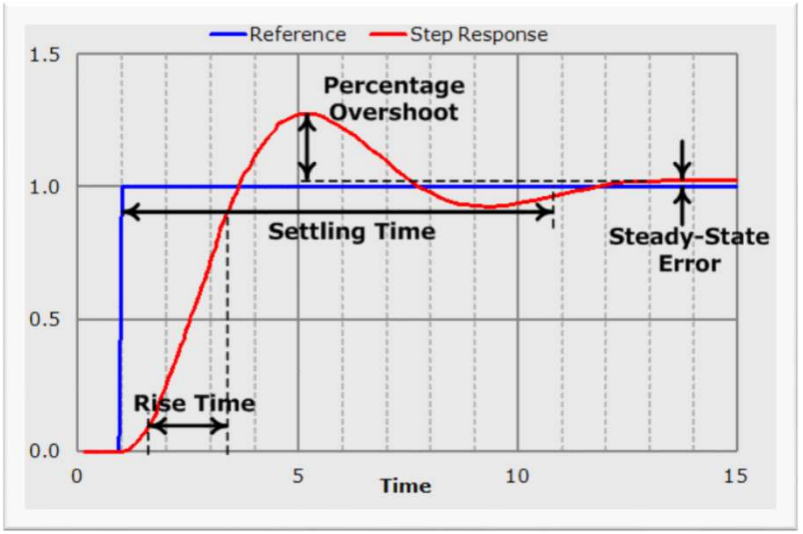
# Tuning of PI Controller Gains

› Effects of increasing proportional gain Kp or integral gain Ki of PI controller independently ↑ Increase ↓ Decrease

| | Gain Change | Effects on Step Response Characteristics | | | |
|---|---|---|---|---|---|
| | | Rise Time | Overshoot | Settling Time | Steady-State Error |
| 1 | $K_p$ ↑ $K_i$ unchanged | ↓ ☺ | ↑ ☹ | Minor Change 😐 | ↓ ☺ |
| 2 | $K_i$ ↑ $K_p$ unchanged | ↓ ☺ | ↑ ☹ | ↑ ☹ | Eliminate ☺ |




$$K = \frac{K\_integer}{2^{SCALE\_KIKP}}$$

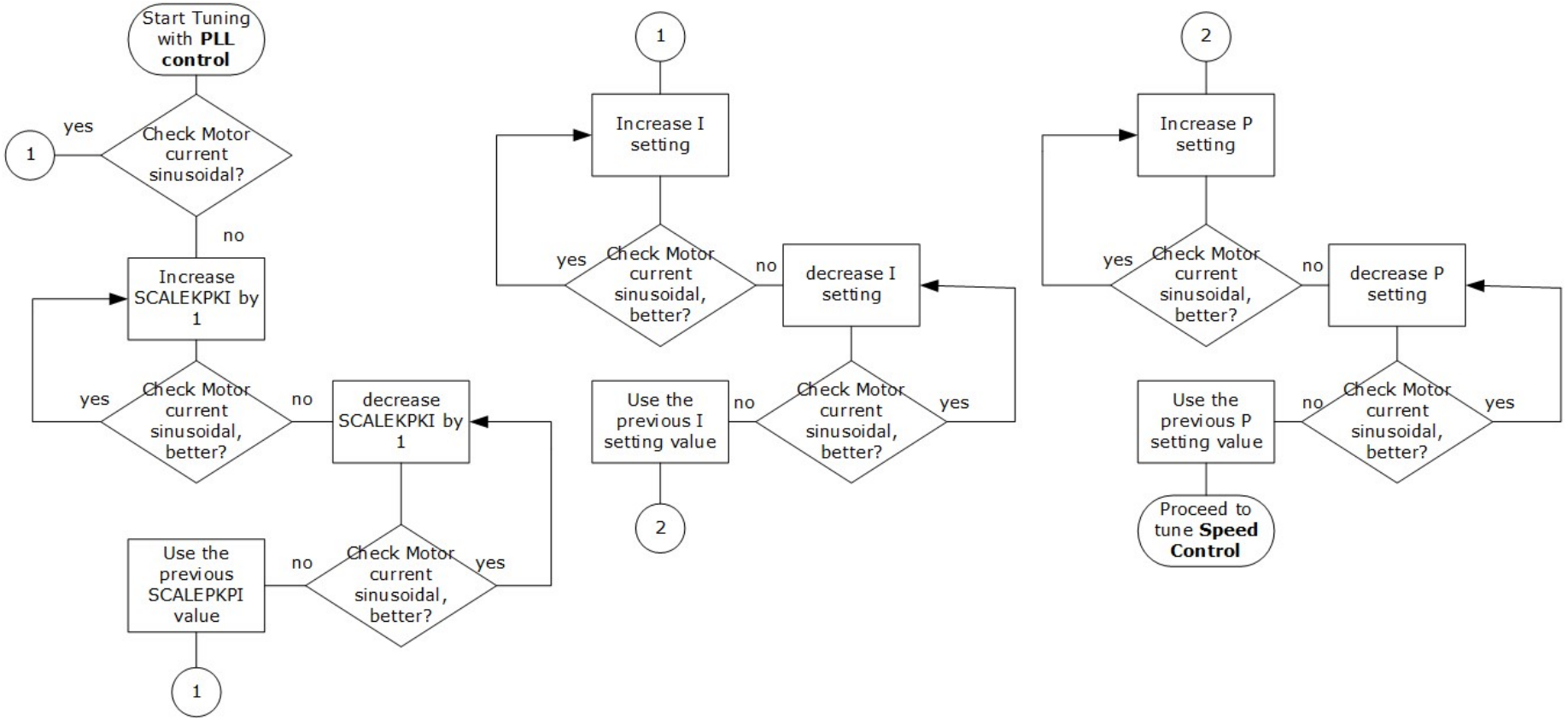


# Hint to fine tuning

1. Set Control Scheme VQ_CONTROLLED_DIRECT_FOC just to run motor and check HW/SW configurator
2. Change control scheme to TORQUE_CONTROLLED_DIRECT_FOC and adjust Ki Kp Parameters of PLL ,Torque and Flux PI
3. Then set control scheme to SPEED_CONTROLLED_DIRECT_FOC if speed control is need, and adjust Ki Kp of Speed PI

# PLL PI parameters Adjust

› Hints on tuning of SCALEKPKI, P, I value steps.



# General Information (2/2)

- For latest updates, please refer to:

  http://www.infineon.com/xmc1000

- DAVE™ development platform:

  http://www.infineon.com/DAVE

- For support:

  http://www.infineonforums.com/forums/8-XMC-Forum

# Hardware Overview – Kit Order information

| No. | Kit Name | Kit Description | Order Number |
|---|---|---|---|
| 1 | KIT_XMC1x_AK_Motor_001 | XMC1000 Motor Control Application Kit | KIT_XMC1x_AK_Motor_001 |

| Development Boards | | Order Number |
|---|---|---|
| XMC1300 Boot Kit | | KIT_XMC13_BOOT_001 |
| PMSM Low Voltage 15W Card | | KIT_XMC1x_AK_Motor_001 |

# Glossary Abbreviations

- ADC Analog Digital Converter
- DAVE™ Digital Application Virtual Engineer (Free development IDE for XMC™ )
- PWM Pulse Width Modulation
- SW Software

# Disclaimer

The information given in this training materials is given as a hint for the implementation of the Infineon Technologies component only and shall not be regarded as any description or warranty of a certain functionality, condition or quality of the Infineon Technologies component.

Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this training material.

Part of your life. Part of tomorrow.