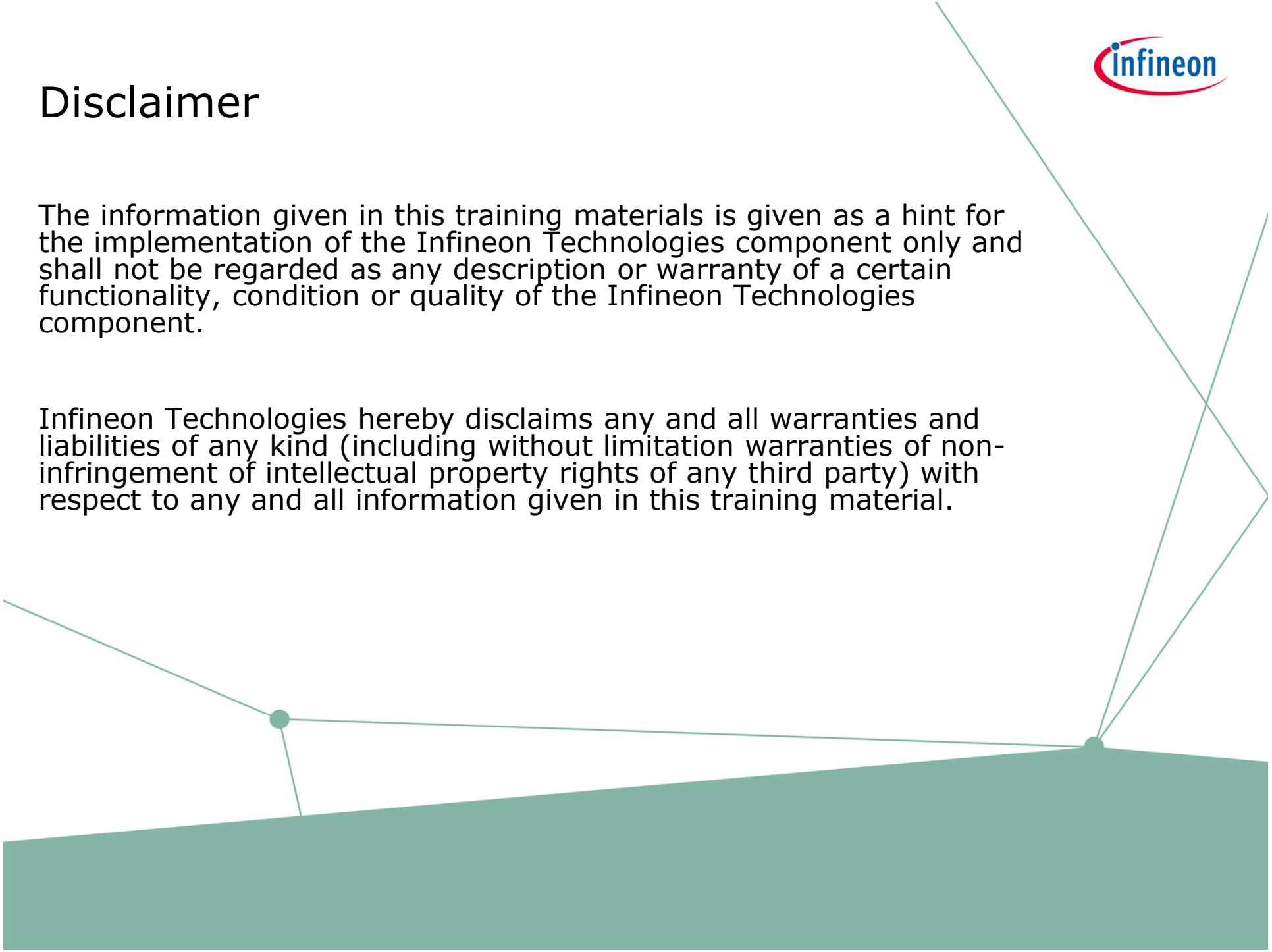
XMC PMSM FOC SENSORLESS SW
Getting Started

XMC™ Microcontrollers
September 2016

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.



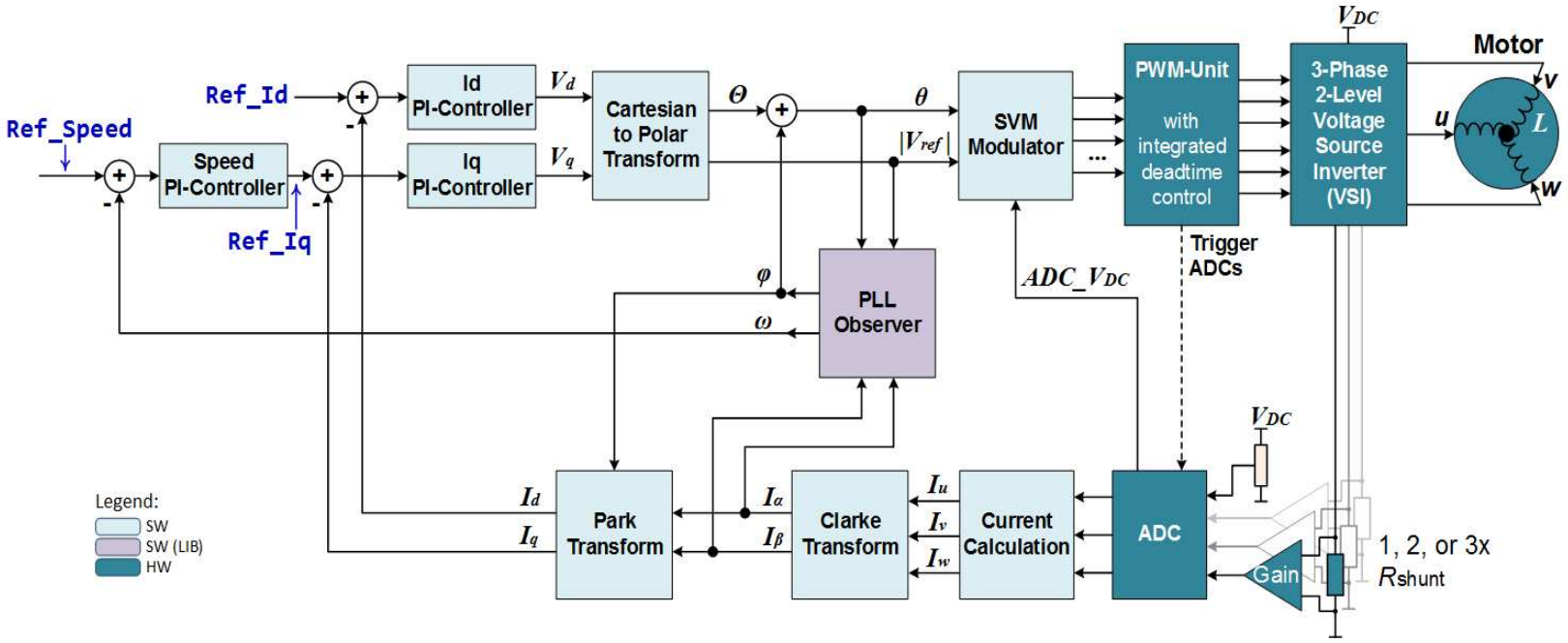
Agenda

1. Overview of FOC Sensorless SW
2. Software Overview
3. Hardware Overview
4. Infineon Tools Overview
5. Getting Started
6. General Information

Overview – PMSM FOC Sensorless SW

- This document provides information about usage of PMSM FOC Sensorless example software on Infineon's XMC1300 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.

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

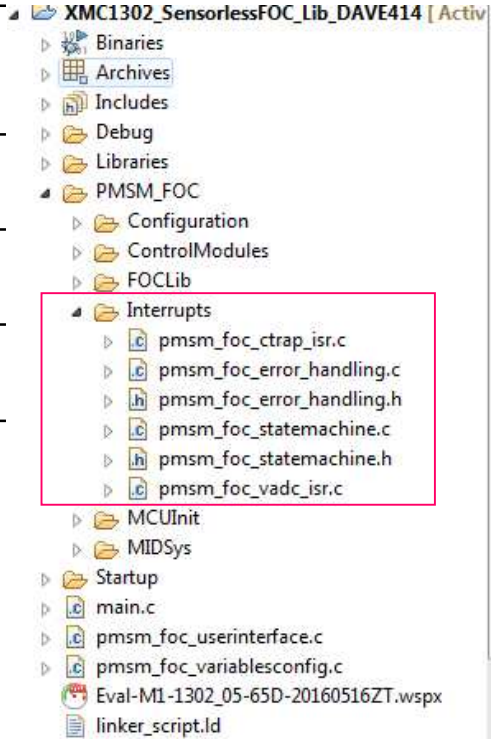
Software Overview – Key Features

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	Over-current Short circuit Under/Over voltage C-trap with MCU hardware features

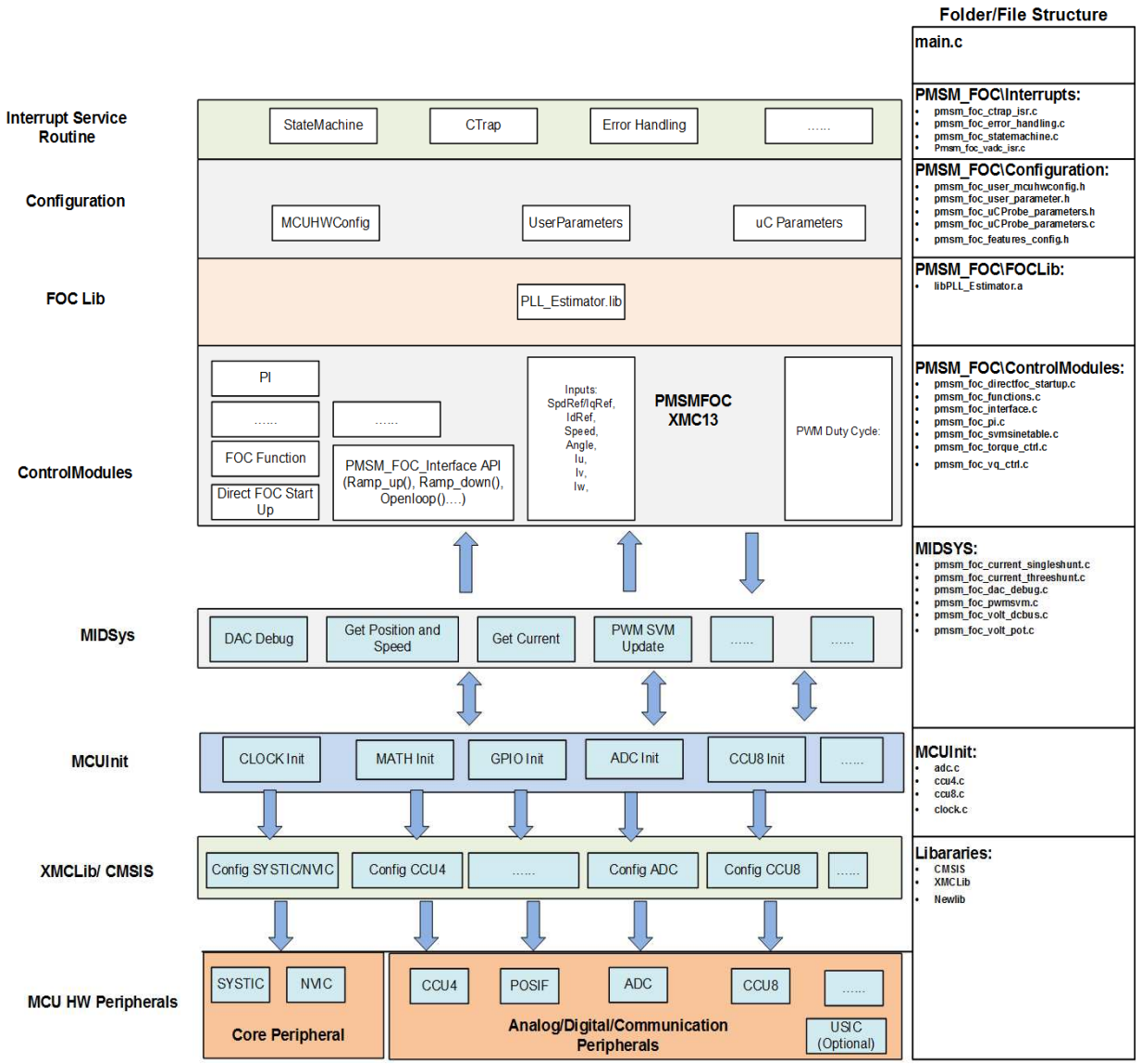
Software Overview - Interrupt Service Routines

Folder: Interrupts

Peripheral	Interrupt Subroutines (ISR)	NVIC node	Interval	Priority
VADC	VADC_Source_IRQHandler	18	Asynchronous	0
CCU8	CTRAP	26	Asynchronous	0
CCU8	One match event (Phase U)	25	1/ PWM frequency	2



Software Overview - Files Structure



Software Overview – Example Configuration

Example Name	PMSM_FOC_XMC13
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	VF_MET_FOC
PWM frequency (Hz)	20000
Speed (rpm)	4200
Ramp up/down rate	500
Protection	VDC under/over voltage protection, over current protection

Hardware Overview - XMC 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 5.1K/(5.1K + 47K)
10	POT	ADC for potentiometer	**P2.5** / G1/CH7	

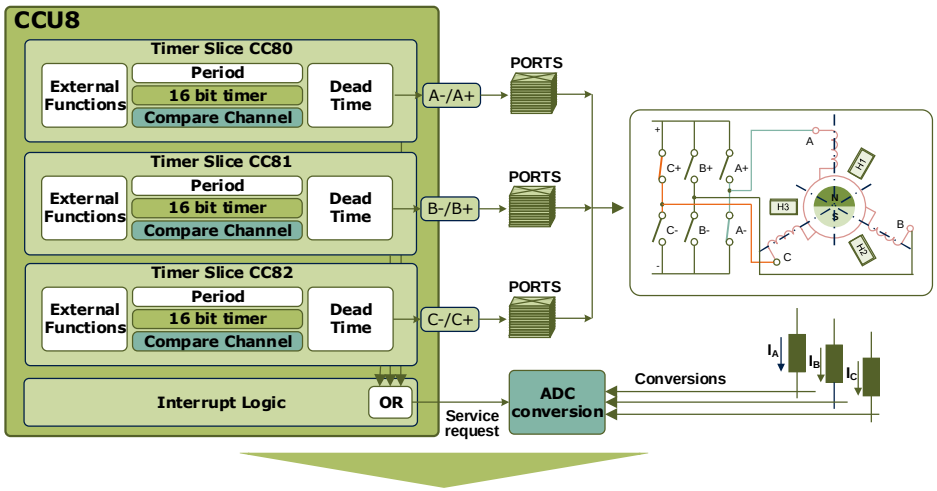
Hardware Overview - XMC Peripheral usage (2/2)

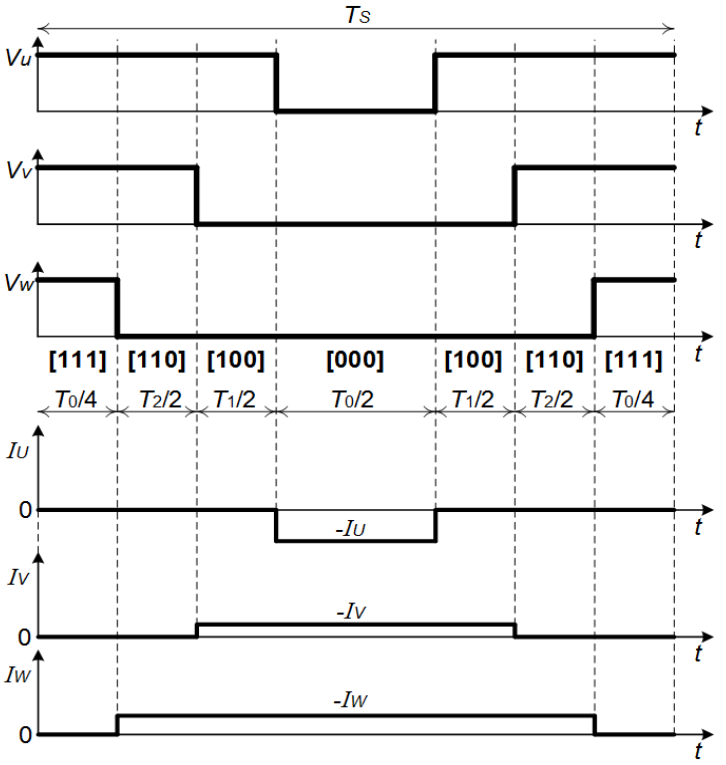
No	Category	Description	XMC1302 Pins	Remark
11	3-Shunt Phase Current	Amplifier output for Phase U shunt	**P2.9** / (G0.CH2/G1.CH4)	3-shunt 50 mΩ, with Op-Amp gain
12		Amplifier output for Phase V shunt	**P2.10** / (G0.CH3/G1.CH2)	
13		Amplifier output for Phase W shunt	**P2.11** / (G0.CH4/G1.CH3)	

Hardware Overview - 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



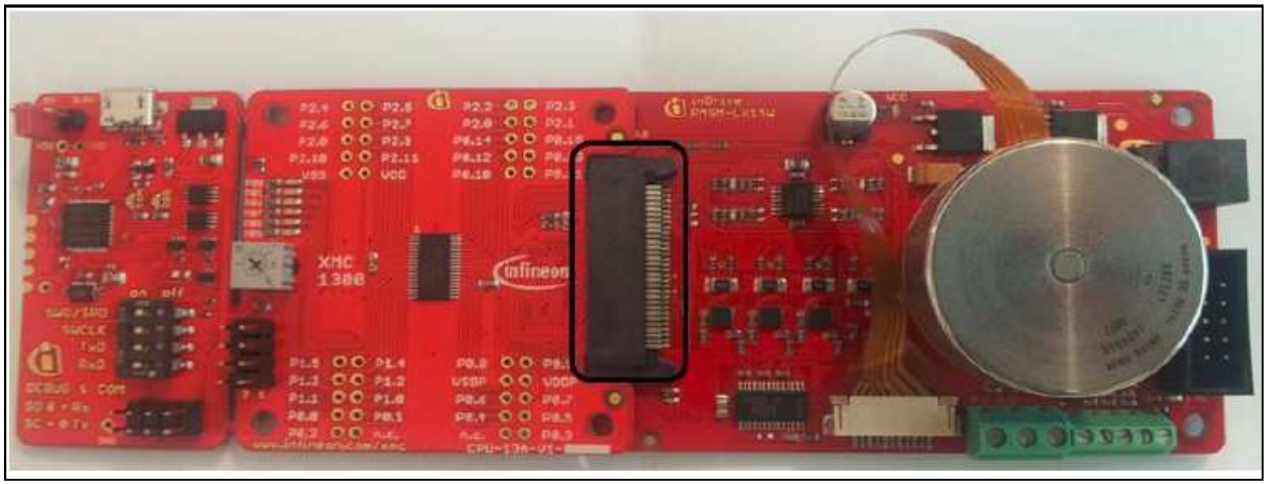


Generate **ADC conversion triggers** synchronized with PWM signal

Hardware Overview – Application Kit Package

- Infineon's XMC1000 Motor Control Application Kit

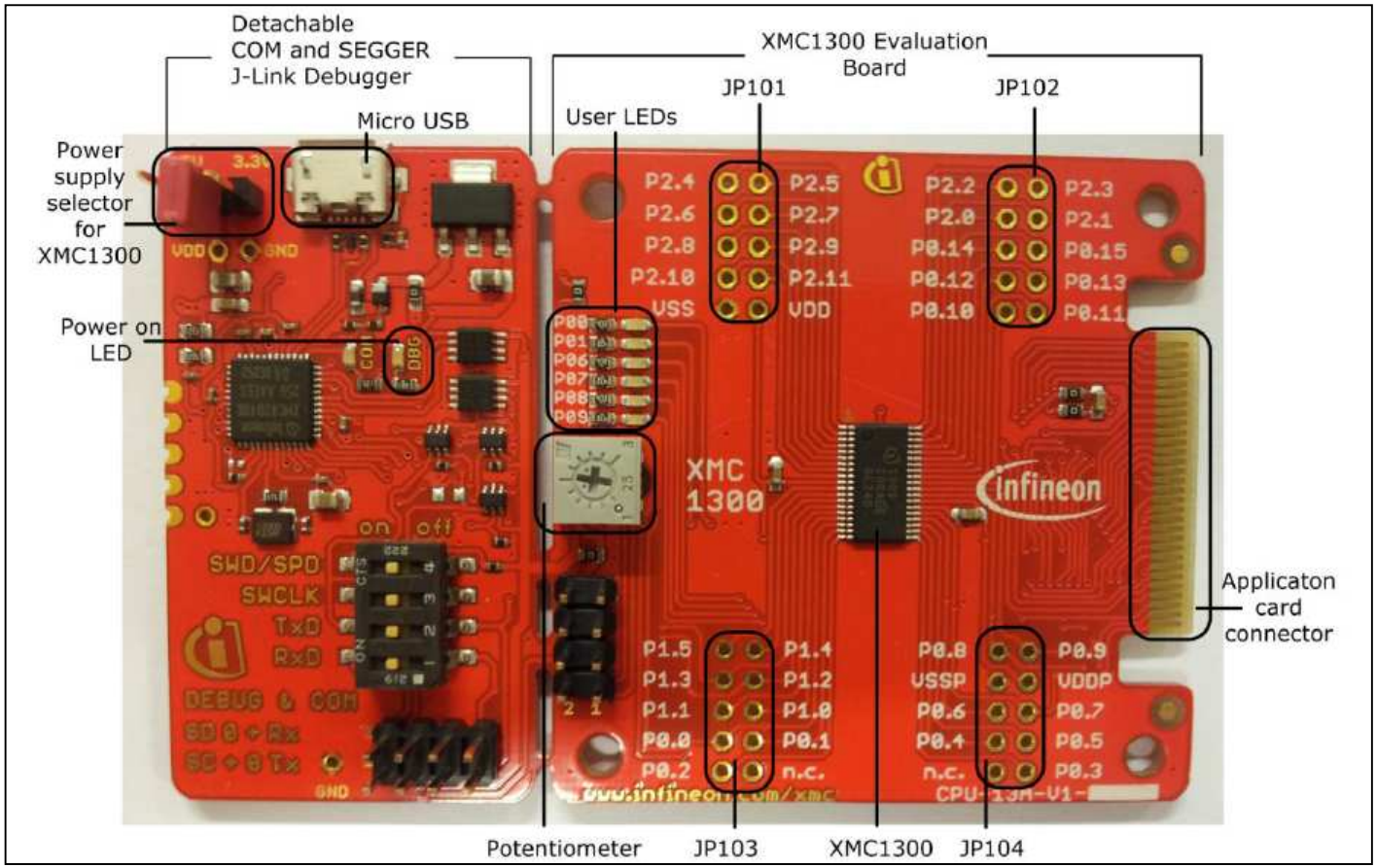
Item	Description
XMC1300 CPU Card	MCU board with XMC1300 and detachable SEGGER J-Link debug interface
PMSM Low Voltage 15W Motor Card	12 – 24V Up to 3A On board 3-phase motor (24V, 15W) with hall sensors
Accessories	Power Supply Adaptor (24V, 1A) Micro USB connector (1x)



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

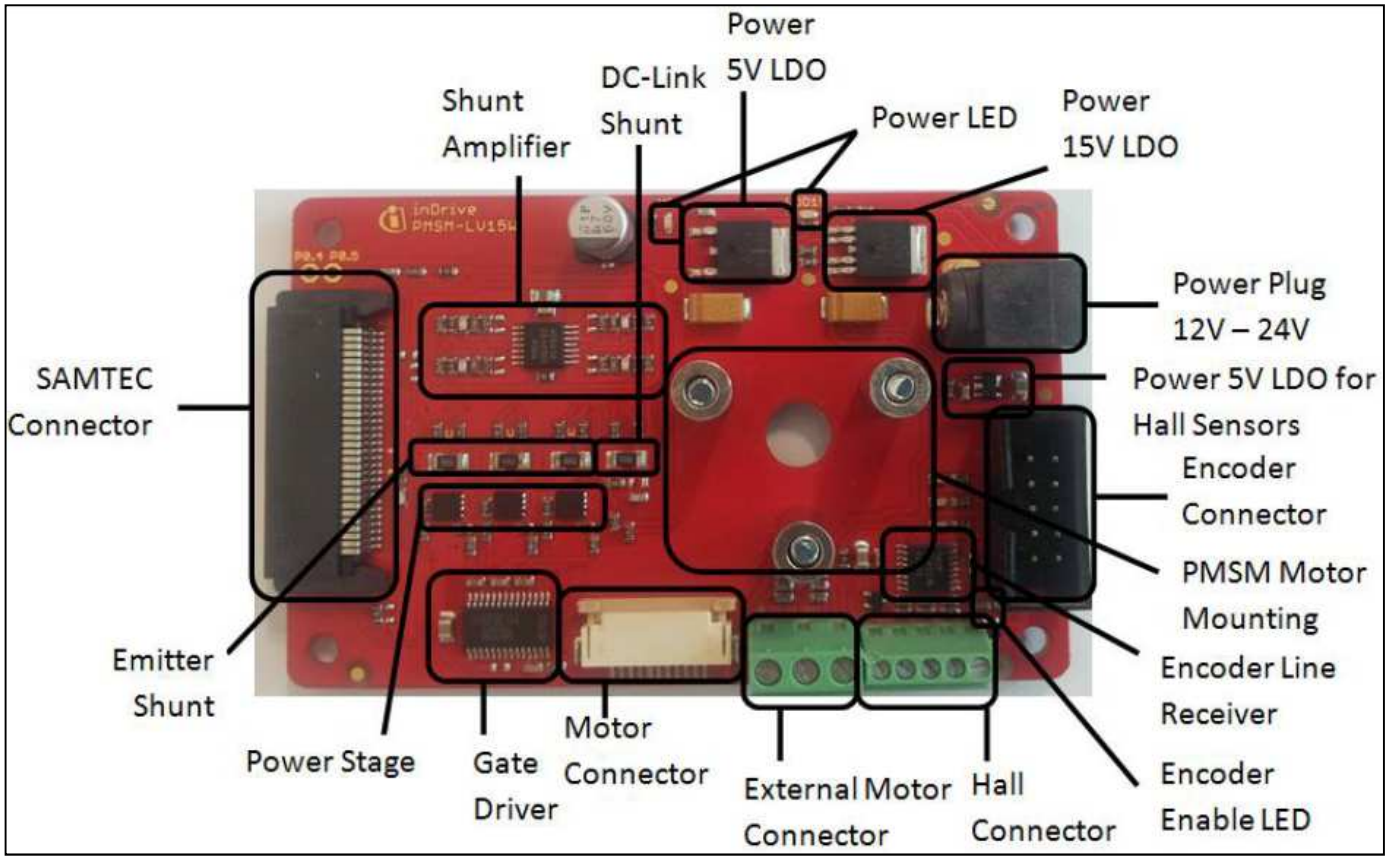
Hardware Overview – XMC1300 CPU Card

› XMC1300 CPU Card



Hardware Overview – Motor Card

› PMSM Low Voltage 15W Motor Card



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

Infineon Tools Overview

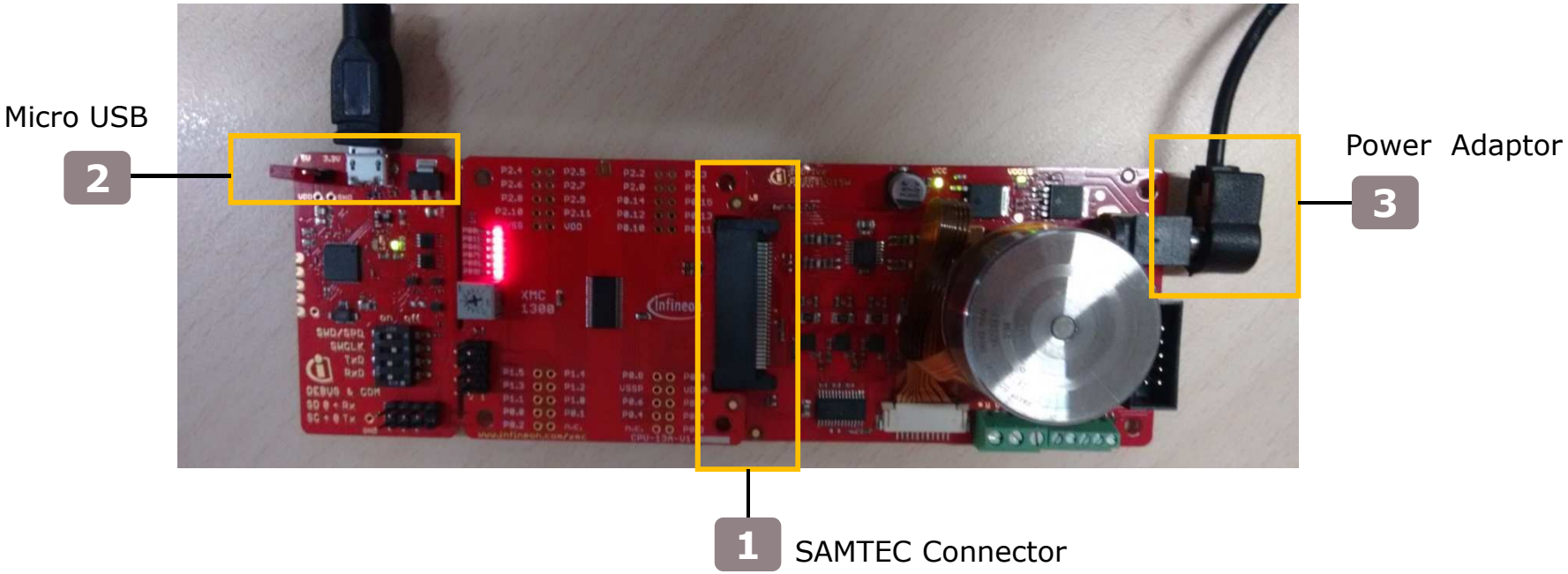
- DAVE™ (V4.2.6 onwards)
  - Download DAVE™ installer package from

http://www.infineon.com/dave

  - Download and unzip the installer package

Getting Started – Connecting the Board

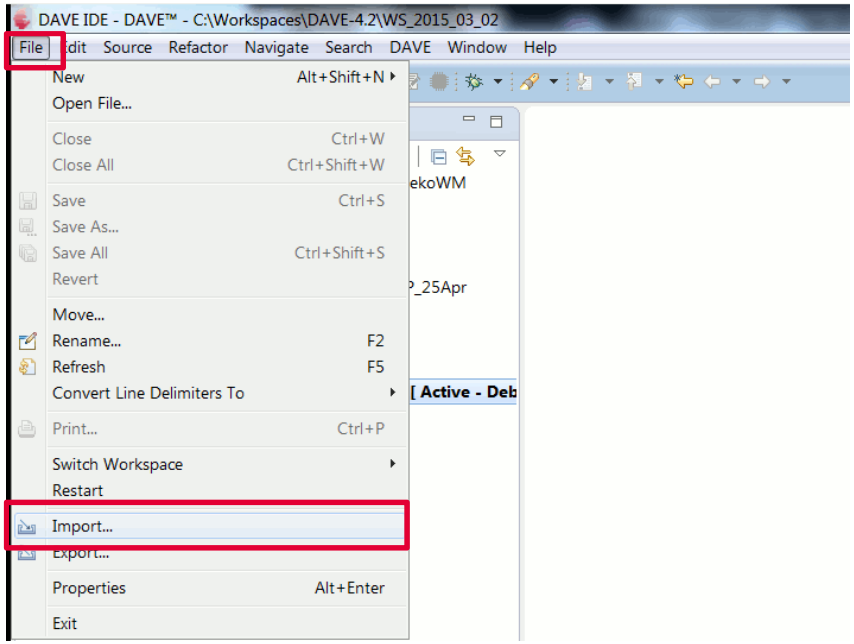
1. Connect XMC1300 CPU Card to PMSM Low Voltage 15W Motor Card using SAMTEC connector interface
2. Connect XMC1300 CPU Card to PC via Micro USB cable
3. Connect power adaptor to PMSM Low Voltage 15W Motor Card

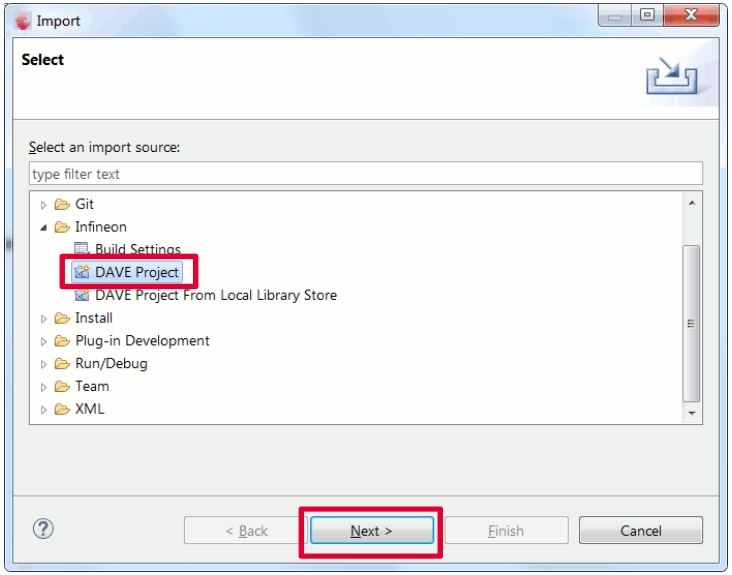


Getting Started – Download Project from DAVE [1/2]

1. Open DAVE™ 

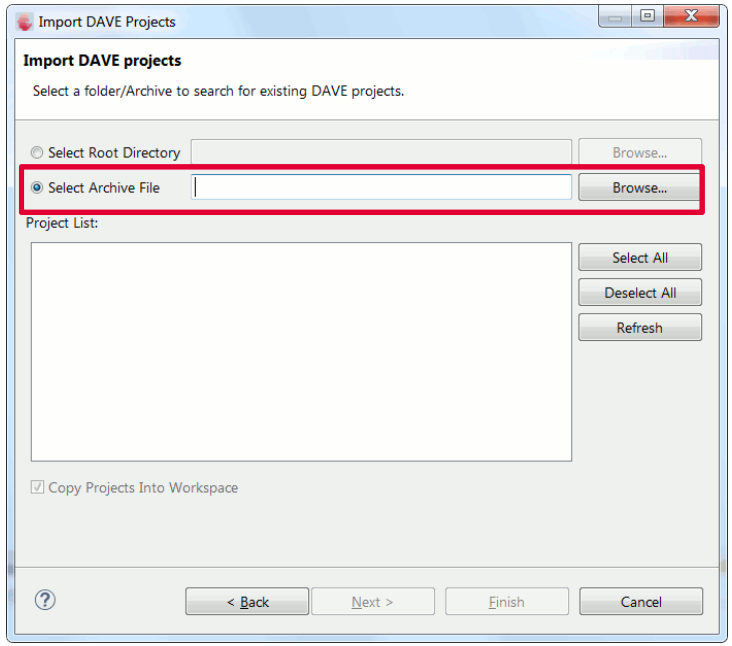
2. Click on **File > Import** to import sample code
3. Select **Infineon > DAVE project** and click "**Next**"

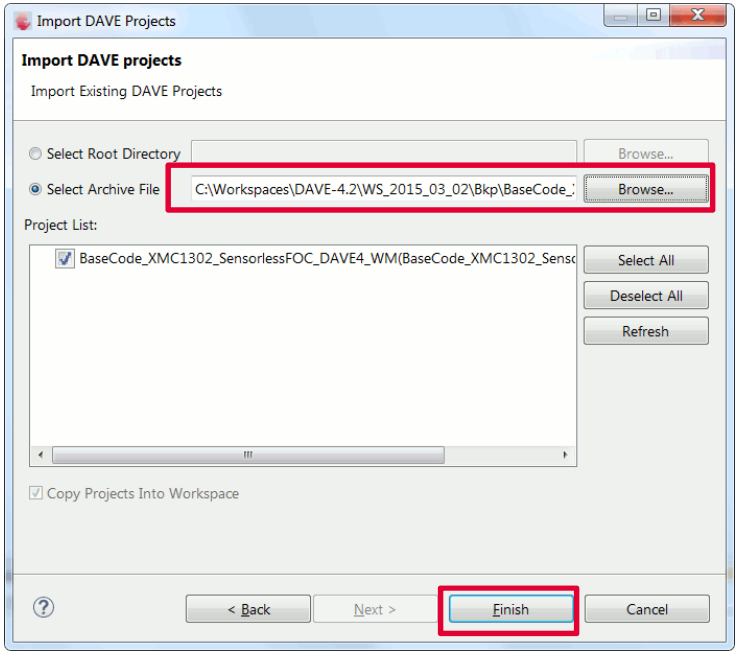




Import FOC example SW to DAVE™ 4 (2/2)

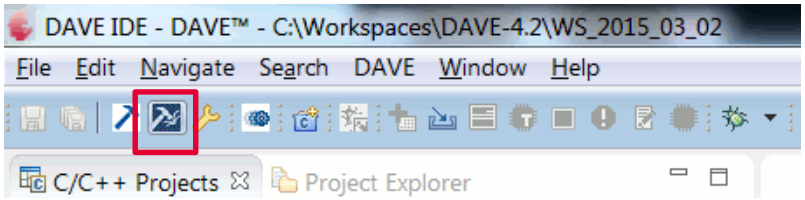
- Next click on **Select Archive File > Browse**.
- Select the folder containing the sample code and click "**OK**".
- Click on "**Finish**" to import the code into DAVE™ 4.

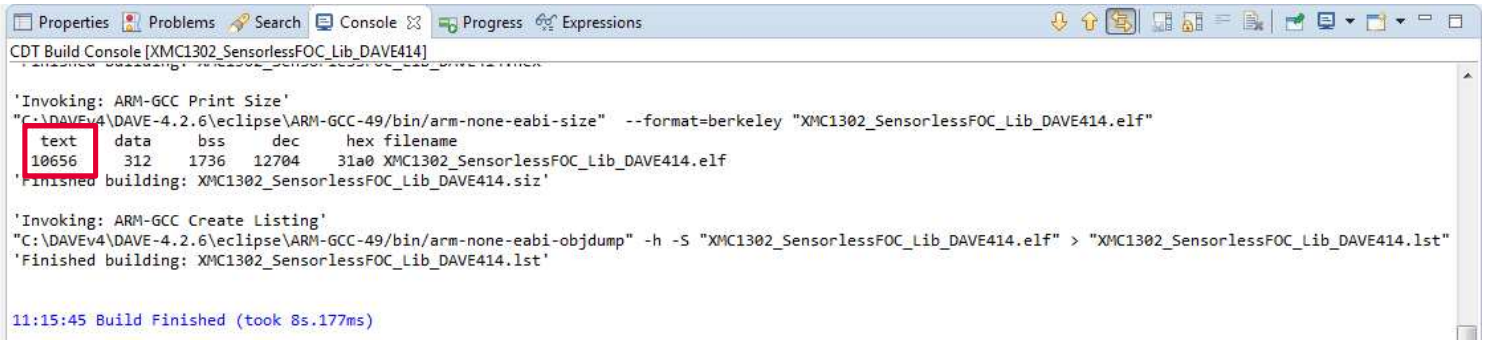




Build FOC example SW in DAVE™ 4

› Click "**Rebuild Active Project**"



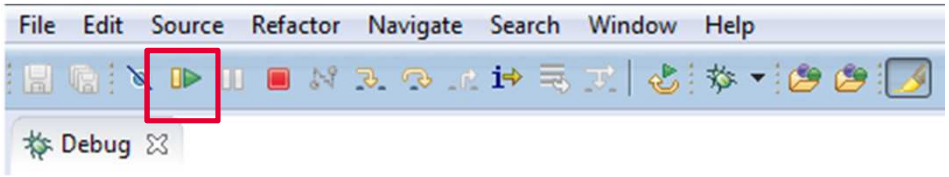


› "text" in red box indicates that code size, e.g.: about 11 Kbytes with UART feature disabled

Download FOC example SW in DAVE™ 4

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

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



Getting Started – Configure the Project [1/9]

Folder: Configuration
File name: pmsm_foc_user_parameter.h

1. Select the KIT_XMC1X_AK_MOTOR_001 and MAXON motor

```
#ifndef CONFIGURATION_PMSM_FOC_USER_PARAMETER_H_
#define CONFIGURATION_PMSM_FOC_USER_PARAMETER_H_

#include "pmsm_foc_features_config.h"

/*      ------------------------------------------------- Inverter Hardware Type Selection ------------------------------------------- */
#define  PMSM_FOC_HARDWARE_BOARD                KIT_XMC1X_AK_MOTOR_001                  /*1. KIT_XMC1X_AK_MOTOR_001
                                                                                        2. KIT_XMC750WATT_MC_AK_V1
                                                                                        3. IFX_XMC_LVPB_R2
                                                                                        4. IFI_EVAL_24V_250W
                                                                                        5. IFX_XMC_LVPB_R3
                                                                                        6. IFX_XMC_PINUS_V2*/

/*      ------------------------------------------------- Motor Type Selection ------------------------------------------------- */
#define  MOTOR_TYPE                             MAXON_MOTOR                     /*1. MCI_DRONE_MOTOR
                                                                                2. DJI_DRONE_MOTOR
                                                                                3. VORNADO_FAN_MOTOR
                                                                                4. NANOTEC_MOTOR
                                                                                5. MAXON_MOTOR
                                                                                6. BEKO_WM_MOTOR
                                                                                7.EBM_PAPST_VENTI_FAN_MOTOR
```

Getting Started – Configure the Project [2/9]

Folder: Configuration
File name: pmsm_foc_user_parameter.h

2. Select the Control Scheme and PWM Modulation Scheme

```
86  /*      ------------------------------------------------------ Current feedback Sensing Mechanism ------------------------------------------ */
87  #define  CURRENT_SENSING                          USER_THREE_SHUNT_SYNC_CONV                /*1. USER_SINGLE_SHUNT_CONV
88                                                                                              2. USER_THREE_SHUNT_ASSYNC_CONV
89                                                                                              3. USER_THREE_SHUNT_SYNC_CONV*/
90  /*      ------------------------------------------------------ FOC Control and Startup Scheme (Only Select 1 Scheme at one time) ----------
91  #define MY_FOC_CONTROL_SCHEME                    CONSTANT_SPEED_VF_MET_FOC                 /* 1. CONSTANT_SPEED_VF_ONLY,
92                                                                                              2. CONSTANT_SPEED_VF_MET_FOC
93                                                                                              3. CONSTANT_SPEED_DIRECT_FOC
94                                                                                              4. CONSTANT_TORQUE_DIRECT_FOC
95                                                                                              5. CONSTANT_VQ_DIRECT_FOC */
```

- By enabling different marco control scheme will enable different type of statemachine.

```
603
604 #if (MY_FOC_CONTROL_SCHEME == CONSTANT_SPEED_VF_ONLY)
605   #define VF_ONLY_CCU80_0_IRQHandler           CCU80_0_IRQHandler
606 #elif (MY_FOC_CONTROL_SCHEME == CONSTANT_SPEED_VF_MET_FOC)
607   #define VF_FOC_CCU80_0_IRQHandler            CCU80_0_IRQHandler
608 #elif ((MY_FOC_CONTROL_SCHEME == CONSTANT_SPEED_DIRECT_FOC) || (MY_FOC_CONTROL_SCHEME == CONSTANT_TORQUE_DIRECT_FOC) || \
609        (MY_FOC_CONTROL_SCHEME == CONSTANT_VQ_DIRECT_FOC) )
610   #define DirectFOCStartUp_CCU80_0_IRQHandler   CCU80_0_IRQHandler
611 #endif
612
613 #endif /* PMSM_FOC_CONFIGURATION_PMSM_FOC_USER_PARAMETER_H_ */
```

Getting Started – Configure the Project [3/9]

3. Configure Inverter Hardware

Folder: Configuration
File name: pmsm_foc_user_parameter.h

```
#elif(PMSM_FOC_HARDWARE_BOARD == KIT_XMC1X_AK_MOTOR_001)
#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_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 ratio
#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_MAX

#if(INTERNAL_OP_GAIN == ENABLED)
#define OP_GAIN_FACTOR                        (3U)                     /* Different HW Board has different OP Gain factor, XMC13 built-in Gai
#elif(INTERNAL_OP_GAIN == DISABLED)
#define OP_GAIN_FACTOR                        G_OPAMP_PER_PHASECURRENT
#endif
```

Getting Started – Configure the Project [4/9]

4. Configure MCU CCU8 resources

Folder: Configuration
File name: pmsm_foc_user_mcuhwconfig.h

- Predefined CCU8 resources with KIT_XMC1x_AK_Motor_001

```
/*******************************************************************************************************************
 * KIT_XMC1X_AK_MOTOR_001
 * GPIO Resources Configuration
 *******************************************************************************************************************/
#define TRAP_PIN               P0_12
#define INVERTER_EN_PIN        P0_11

#define PHASE_U_HS_PIN         P0_0
#define PHASE_U_HS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

#define PHASE_U_LS_PIN         P0_1
#define PHASE_U_LS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

#define PHASE_V_HS_PIN         P0_7
#define PHASE_V_HS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

#define PHASE_V_LS_PIN         P0_6
#define PHASE_V_LS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

#define PHASE_W_HS_PIN         P0_8
#define PHASE_W_HS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

#define PHASE_W_LS_PIN         P0_9
#define PHASE_W_LS_ALT_SELECT  XMC_GPIO_MODE_OUTPUT_PUSH_PULL_ALT5

/*******************************************************************************************************************
 * CCU8 Resources Configuration
 *******************************************************************************************************************/
#define CCU8_MODULE           CCU80
#define CCU8_MODULE_PHASE_U   CCU80_CC80
#define CCU8_MODULE_PHASE_V   CCU80_CC81
#define CCU8_MODULE_PHASE_W   CCU80_CC82
#define CCU8_MODULE_ADC_TR    CCU80_CC83

//#define CCU8_PASSIVE_LEVEL    XMC_CCU8_SLICE_OUTPUT_PASSIVE_LEVEL_HIGH
#define CCU8_PASSIVE_LEVEL    XMC_CCU8_SLICE_OUTPUT_PASSIVE_LEVEL_LOW

#define CCU8_INPUT_TRAP_LEVEL  XMC_CCU8_SLICE_EVENT_LEVEL_SENSITIVITY_ACTIVE_LOW
//#define CCU8_INPUT_TRAP_LEVEL  XMC_CCU8_SLICE_EVENT_LEVEL_SENSITIVITY_ACTIVE_HIGH

#define INVERTER_ENABLE_PIN    (1U)            /* 1 = Active High, 0 Active Low*/
```

Getting Started – Configure the Project [4/9]

Folder: Configuration

File name: pmsm_foc_user_mcuhwconfig.h

```
#define INVERTER_ENABLE_PIN    (1U)            /* 1 = Active High, 0 Active Low*/

#if(INVERTER_ENABLE_PIN == 0U)
  #define ENABLE_LEVEL XMC_GPIO_OUTPUT_LEVEL_LOW
  #define DISABLE_LEVEL XMC_GPIO_OUTPUT_LEVEL_HIGH
#elif(INVERTER_ENABLE_PIN == 1U)
  #define ENABLE_LEVEL XMC_GPIO_OUTPUT_LEVEL_HIGH
  #define DISABLE_LEVEL XMC_GPIO_OUTPUT_LEVEL_LOW
#endif
```

Getting Started – Configure the Project [5/9]

5. Configure VADC resources

Folder: Configuration
File name: pmsm_foc_user_mcuhwconfig.h

- Predefined VADC resources with KIT_XMC1x_AK_Motor_001

```
/*******************************************************************************************************************
 * VADC Resources Configuration
 ******************************************************************************************************************/

#define VADC_I1_GROUP          VADC_G1
#define VADC_I1_CHANNEL        (0U)
#define VADC_I1_RESULT_REG     (0U)

#define VADC_I3_GROUP          VADC_G1
#define VADC_I3_CHANNEL        (1U)
#define VADC_I3_RESULT_REG     (1U)

#define VADC_I2_GROUP          VADC_G0
#define VADC_I2_CHANNEL        (0U)
#define VADC_I2_RESULT_REG     (0U)

#define VADC_I4_GROUP          VADC_G0
#define VADC_I4_CHANNEL        (1U)
#define VADC_I4_RESULT_REG     (1U)

/* Motor Phase U VADC define */
#define VADC_IU_G1_CHANNEL     (3U)        /* P2.11, VADC group1 channel 3 */
#define VADC_IU_G0_CHANNEL     (4U)        /* P2.11, VADC group0 channel 4 */

#define VADC_IU_GROUP          VADC_G1
#define VADC_IU_GROUP_NO       (1U)
#define VADC_IU_CHANNEL        (3U)        /* P2.11, VADC group1 channel 3 */
#define VADC_IU_RESULT_REG     (3U)

/* Motor Phase V VADC define */
#define VADC_IV_G1_CHANNEL     (2U)       /* P2.10, VADC group1 channel 2 */
#define VADC_IV_G0_CHANNEL     (3U)       /* P2.10, VADC group0 channel 3 */

#define VADC_IV_GROUP          VADC_G1
#define VADC_IV_GROUP_NO       (1U)
#define VADC_IV_CHANNEL        (2U)       /* P2.10, VADC group1 channel 2 */
#define VADC_IV_RESULT_REG     (2U)
```

```
/* DC link voltage VADC define */
#define VADC_VDC_GROUP         VADC_G1
#define VADC_VDC_GROUP_NO      (1U)
#define VADC_VDC_CHANNEL       (5U)      /* P2.3 VADC group1 channel 5 */
#define VADC_VDC_RESULT_REG    (5U)

/* DC link current VADC define */
#define VADC_IDC_GROUP         VADC_G1
#define VADC_IDC_GROUP_NO      (1U)
#define VADC_IDC_CHANNEL       (6U)       /* P2.4 VADC group1 channel 6 */
#define VADC_IDC_RESULT_REG    (6U)

/* Potentiometer VADC define*/
#define VADC_POT_GROUP         VADC_G1
#define VADC_POT_GROUP_NO      (1U)
#define VADC_POT_CHANNEL       (7U)      /* P2.5 VADC group1 channel 7 */
#define VADC_POT_RESULT_REG    (7U)

/* VADC Group 0 Alias channel 0 and channel 1 */
#define VADC_G0_CHANNEL_ALIAS0  VADC_IV_G0_CHANNEL
#define VADC_G0_CHANNEL_ALIAS1  VADC_IDC_CHANNEL

/* VADC Group 1 Alias channel 0 and channel 1 */
#define VADC_G1_CHANNEL_ALIAS0  VADC_IW_G1_CHANNEL
#define VADC_G1_CHANNEL_ALIAS1  VADC_IU_G1_CHANNEL
```

Getting Started – Configure the Project [6/9]

6. Enable/Disable UART

Folder: Configuration
File name: pmsm_foc_user_parameter.h

- Predefined UART resources with KIT_XMC1x_AK_Motor_001

```
#define UART_ENABLE            USIC0_CH1_P1_2_P1_3          /* 1. USIC_DISABLED_ALL
                                                               2. USIC0_CH0_P1_4_P1_5
                                                               3. USIC0_CH1_P1_2_P1_3 */
```

- Adjusting speed with POT available if user disable USIC feature.

Getting Started – Configure the Project [7/9]

7. Configure the Motor Profile

Folder: Configuration
File name: pmsm_foc_user_parameter.h

```
#elif (MOTOR_TYPE == MAXON_MOTOR)
/*      ------------------------------------------------- Motor Parameters ------------------------------------------ */
#define  USER_MOTOR_R_PER_PHASE_OHM                (6.8f)        /* Motor Resistance per phase in Ohm*/
#define  USER_MOTOR_L_PER_PHASE_uH                 (3865.0f)         /* Motor Inductance per phase in uH */
#define  USER_MOTOR_POLE_PAIR                      (4.0f)          /* Motor Pole Pairs */
/*      ------------------------------------------------- Constant Speed Control Mode (Used when Constant Speed Control is enabled) ---------
/*      ------------------------------------------------- POT ADC, or PWM to Adjust Speed ---------------------------------------- */
#define USER_SPEED_HIGH_LIMIT_RPM                 (4530.0f)
#define USER_SPEED_LOW_LIMIT_RPM                  (uint32_t) (USER_SPEED_HIGH_LIMIT_RPM / 30U)
#define USER_SPEED_RAMPUP_RPM_PER_S               (500U)
#define USER_SPEED_RAMPDOWN_RPM_PER_S             (500U)
/*      ------------------------------------------------- V/F Start Up Parameters ------------------------------------------ */
#define USER_STARTUP_SPEED_RPM                    (0U)
#define USER_STARTUP_SPEED_THRESHOLD_RPM          (200U)            /* threshold Speed to transit from Open loop to closed loop */
#define USER_STARTUP_VF_OFFSET_V                  (1.0f)            /* V/F startup offset in V */
#define USER_STARTUP_VF_SLEWRATE_V_PER_HZ         (0.1f)           /* V/F start up slew rate in V/Hz */
```

Getting Started – Configure the Project [8/9]

8. Tuning of Kp, Ki value

Folder: ControlModules
File name: pmsm_foc_pi.h

```
278 #elif (MOTOR_TYPE == MAXON_MOTOR)
279           /* For Low Voltage 15W Board with MAXON Motor */
280 /****************** For Speed PI controller ********************/
281 #define PI_SPEED_KP           ((uint16_t)1U<<15U)         /* (1<<15). Proportional gain Kp, uint16_t. */
282 #define PI_SPEED_KI           ((uint16_t)3)               /* (1<<3). Integral gain Ki, uint16_t. */
283 #define PI_SPEED_SCALE_KPKI   (10 + RES_INC)              /* RES_INC: Angle/speed resolution increase from 16 bit.*/
284
285 /* Note: (IK_LIMIT_MIN << SCALE_KPKI) and (IK_LIMIT_MAX << SCALE_KPKI) are maximum int32_t. Same as below. */
286 #define PI_SPEED_IK_LIMIT_MIN  (-(((1<<15) * 3) >> 2))    /* (-(1<<15)). I[k] output limit LOW. */
287 #define PI_SPEED_IK_LIMIT_MAX  (((1<<15) * 3) >> 2)       /* (1<<15). I[k] output limit HIGH. */
288
289 #define PI_SPEED_UK_LIMIT_MIN  (16)          /* (-32767), 16. U[k] output limit LOW. */
290 #define PI_SPEED_UK_LIMIT_MAX  (32767)        /* MAX_I_REF. U[k] output limit HIGH. Normally no need change. */
291
292 /****************** For Torque / Iq PI controller ********************/
293 /* Kp and Ki (from excel file) calculated from motor parameter L and R. Normally no need change. */
294 #define PI_TORQUE_KP        (USER_DEFAULT_IQID_KP)          /* (1<<13). Proportional gain Kp, uint16_t. */
295 #define PI_TORQUE_KI        (USER_DEFAULT_IQID_KI >> 0)     /* (1<<6). Integral gain Ki, Ki/Kp = RxTs/L. uint16_t. */
296 #define PI_TORQUE_SCALE_KPKI    (SCALING_CURRENT_KPKI + 0)
297
298 #define PI_TORQUE_IK_LIMIT_MIN    (-32768)      /* (-(1<<15)). I[k] output limit LOW. Normally no need change. */
299 #define PI_TORQUE_IK_LIMIT_MAX    (32767)       /* (1<<15). I[k] output limit HIGH. Normally no need change. */
300
301 #define PI_TORQUE_UK_LIMIT_MIN    (-32768)      /* U[k] output limit LOW. Normally no need change. */
302 #define PI_TORQUE_UK_LIMIT_MAX    32767         /* U[k] output limit HIGH. Normally no need change. */
303
304 /****************** For Flux / Id PI controller ********************/
305 /* Kp and Ki (from excel file) calculated from motor parameter L and R. Normally no need change. */
306 #define PI_FLUX_KP          (USER_DEFAULT_IQID_KP)          /* (1<<13). Proportional gain Kp, uint16_t. */
307 #define PI_FLUX_KI          (USER_DEFAULT_IQID_KI >> 0)     /* (1<<6). Integral gain Ki, Ki/Kp = RxTs/L. uint16_t. */
308 #define PI_FLUX_SCALE_KPKI  (SCALING_CURRENT_KPKI + 0)
309
310 #define PI_FLUX_IK_LIMIT_MIN    (-32768)      /* (-(1<<15)). I[k] output limit LOW. Normally no need change. */
311 #define PI_FLUX_IK_LIMIT_MAX    (32767)       /* (1<<15). I[k] output limit HIGH. Normally no need change. */
312
313 #define PI_FLUX_UK_LIMIT_MIN    (-32768)      /* U[k] output limit LOW. Normally no need change. */
314 #define PI_FLUX_UK_LIMIT_MAX    32767         /* U[k] output limit HIGH. Normally no need change. */
315
316 /****************** For PLL rotor speed PI controller ********************/
317 #define PI_PLL_KP         ((uint16_t)(1<<8))            /* Proportional gain Kp, uint16_t. */
318 #define PI_PLL_KI         ((uint16_t)(1<<6))            /* (1<<4). Integral gain Ki, uint16_t. */
319 #define PI_PLL_SCALE_KPKI     (19 - RES_INC)
320
321 /* I[k] output limit LOW. */
322 #define PI_PLL_IK_LIMIT_MIN     (-(int32_t)((uint32_t)1 << (uint32_t)(30U-(uint32_t)PI_PLL_SCALE_KPKI)))
323 #define PI_PLL_IK_LIMIT_MAX     ((uint32_t)1 << (30U-(uint32_t)PI_PLL_SCALE_KPKI))     /* I[k] output limit HIGH. */
324
325 #define PI_PLL_UK_LIMIT_MIN     ((uint32_t)SPEED_LOW_LIMIT >> 4)              /* U[k] output limit LOW. */
326 #define PI_PLL_UK_LIMIT_MAX     (SPEED_HIGH_LIMIT + SPEED_LOW_LIMIT)          /* U[k] output limit HIGH.*/
```

DEFAULT_IQ_KP/KI

DEFAULT_ID_KP/KI

- **USER_DEFAULT_IQID_KP** and **USER_DEFAULT_IQID_KI** for Torque and Flux PI Controllers are calculated from the physical motor and system parameters, and typically **don't need to be tuned** in the first iteration
- The Speed PI and PLL PI controller parameters should start to be modified if the motor **cannot transit** from V/F open-loop to FOC closed-loop smoothly

Getting Started – Configure the Project [9/9]

9. Tuning Kp, Ki Speed & PLL Controller

Folder: ControlModules
File name: pmsm_foc_pi.h

```
#elif (MOTOR_TYPE == MAXON_MOTOR)
        /* For Low Voltage 15W Board with MAXON Motor */
/*################### For Speed PI controller ######################*/
#define PI_SPEED_KP           ((uint16_t)1U<<15U)          /* (1<<15). Proportional gain Kp, uint16_t. */
#define PI_SPEED_KI           ((uint16_t)3)                /* (1<<3). Integral gain Ki, uint16_t. */
#define PI_SPEED_SCALE_KPKI   (10 + RES_INC)               /* RES_INC: Angle/speed resolution increase from 16 bit */

/* Note: (IK_LIMIT_MIN << SCALE_KPKI) and (IK_LIMIT_MAX << SCALE_KPKI) are maximum int32_t. Same as below. */
#define PI_SPEED_IK_LIMIT_MIN  (-(((1<<15) * 3) >> 2))     /* (-(1<<15)). I[k] output limit LOW. */
#define PI_SPEED_IK_LIMIT_MAX  (((1<<15) * 3) >> 2)        /* (1<<15). I[k] output limit HIGH. */

#define PI_SPEED_UK_LIMIT_MIN  (16)           /* (-32767), 16. U[k] output limit LOW. */
#define PI_SPEED_UK_LIMIT_MAX  (32767)         /* MAX_I_REF. U[k] output limit HIGH. Normally no need change. */

/*################### For Torque / Iq PI controller ######################*/
/* Kp and Ki (from excel file) calculated from motor parameter L and R. Normally no need change. */
#define PI_TORQUE_KP        (USER_DEFAULT_IQID_KP)          /* (1<<13). Proportional gain Kp, uint16_t. */
#define PI_TORQUE_KI        (USER_DEFAULT_IQID_KI >> 0)     /* (1<<6). Integral gain Ki, Ki/Kp = RxTs/L. uint16_t. */
#define PI_TORQUE_SCALE_KPKI    (SCALING_CURRENT_KPKI + 0)

#define PI_TORQUE_IK_LIMIT_MIN    (-32768)      /* (-(1<<15)). I[k] output limit LOW. Normally no need change. */
#define PI_TORQUE_IK_LIMIT_MAX    (32767)       /* (1<<15). I[k] output limit HIGH. Normally no need change. */

#define PI_TORQUE_UK_LIMIT_MIN    (-32768)      /* U[k] output limit LOW. Normally no need change. */
#define PI_TORQUE_UK_LIMIT_MAX    32767         /* U[k] output limit HIGH. Normally no need change. */

/*################### For Flux / Id PI controller ######################*/
/* Kp and Ki (from excel file) calculated from motor parameter L and R. Normally no need change. */
#define PI_FLUX_KP          (USER_DEFAULT_IQID_KP)          /* (1<<13). Proportional gain Kp, uint16_t. */
#define PI_FLUX_KI          (USER_DEFAULT_IQID_KI >> 0)     /* (1<<6). Integral gain Ki, Ki/Kp = RxTs/L. uint16_t. */
#define PI_FLUX_SCALE_KPKI  (SCALING_CURRENT_KPKI + 0)

#define PI_FLUX_IK_LIMIT_MIN    (-32768)      /* (-(1<<15)). I[k] output limit LOW. Normally no need change. */
#define PI_FLUX_IK_LIMIT_MAX    (32767)       /* (1<<15). I[k] output limit HIGH. Normally no need change. */

#define PI_FLUX_UK_LIMIT_MIN    (-32768)      /* U[k] output limit LOW. Normally no need change. */
#define PI_FLUX_UK_LIMIT_MAX    32767         /* U[k] output limit HIGH. Normally no need change. */

/*################### For PLL rotor speed PI controller ######################*/
#define PI_PLL_KP         ((uint16_t)(1<<8))            /* Proportional gain Kp, uint16_t. */
#define PI_PLL_KI         ((uint16_t)(1<<6))            /* (1<<4). Integral gain Ki, uint16_t. */
#define PI_PLL_SCALE_KPKI     (19 - RES_INC)

/* I[k] output limit LOW. */
#define PI_PLL_IK_LIMIT_MIN     (-(int32_t)((uint32_t)1 << (uint32_t)(30U-(uint32_t)PI_PLL_SCALE_KPKI)))
#define PI_PLL_IK_LIMIT_MAX     ((uint32_t)1 << (30U-(uint32_t)PI_PLL_SCALE_KPKI))     /* I[k] output limit HIGH. */

#define PI_PLL_UK_LIMIT_MIN     ((uint32_t)SPEED_LOW_LIMIT >> 4)              /* U[k] output limit LOW. */
#define PI_PLL_UK_LIMIT_MAX     (SPEED_HIGH_LIMIT + SPEED_LOW_LIMIT)          /* U[k] output limit HIGH.*/
```

↑ this value by 1 will ↓ gain of Speed controller by half

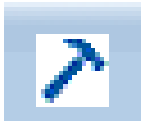
- ↑ the SCALEKPKI of PLL Control and check the motor behaviour.
- If motor start to move slowly, ↑ the SCALEKPKI further. Else, ↓ the SCALEKPKI
- Apply similar tactic for the tuning of Speed Control

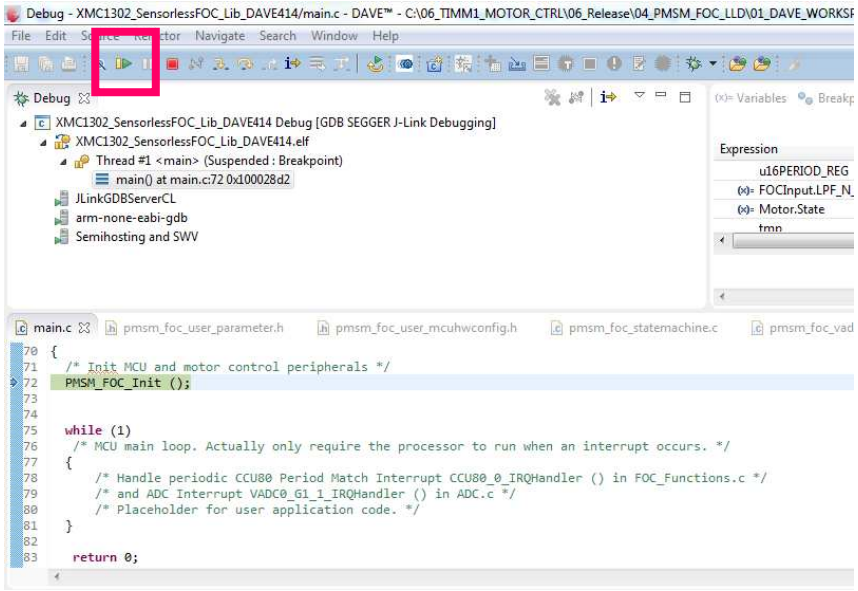
↑ this value by 1 will ↓ gain of PLL Estimator Controller by half

PI gains: $K_p = \frac{P\ setting}{2^{SCALEKPKI}}$

$K_i = \frac{I\ setting}{2^{SCALEKPKI}}$

Getting Started – Compile and Verify the project

1. Click “Build Active Project” 
2. Click “Debug Configuration” to download the code 
3. Click “Resume” to start the application

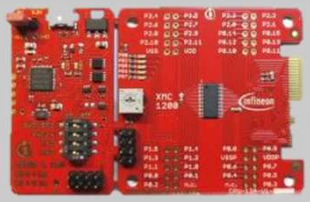


Observation:

› Motor should ramp to 4200RPM with ramp rate of 500RPM/s.

General Information (1/2)

› Where to buy kits:

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

General Information (2/2)

- For latest updates, please refer to:

  http://www.infineon.com/xmc1000

- DAVE™ development platform:

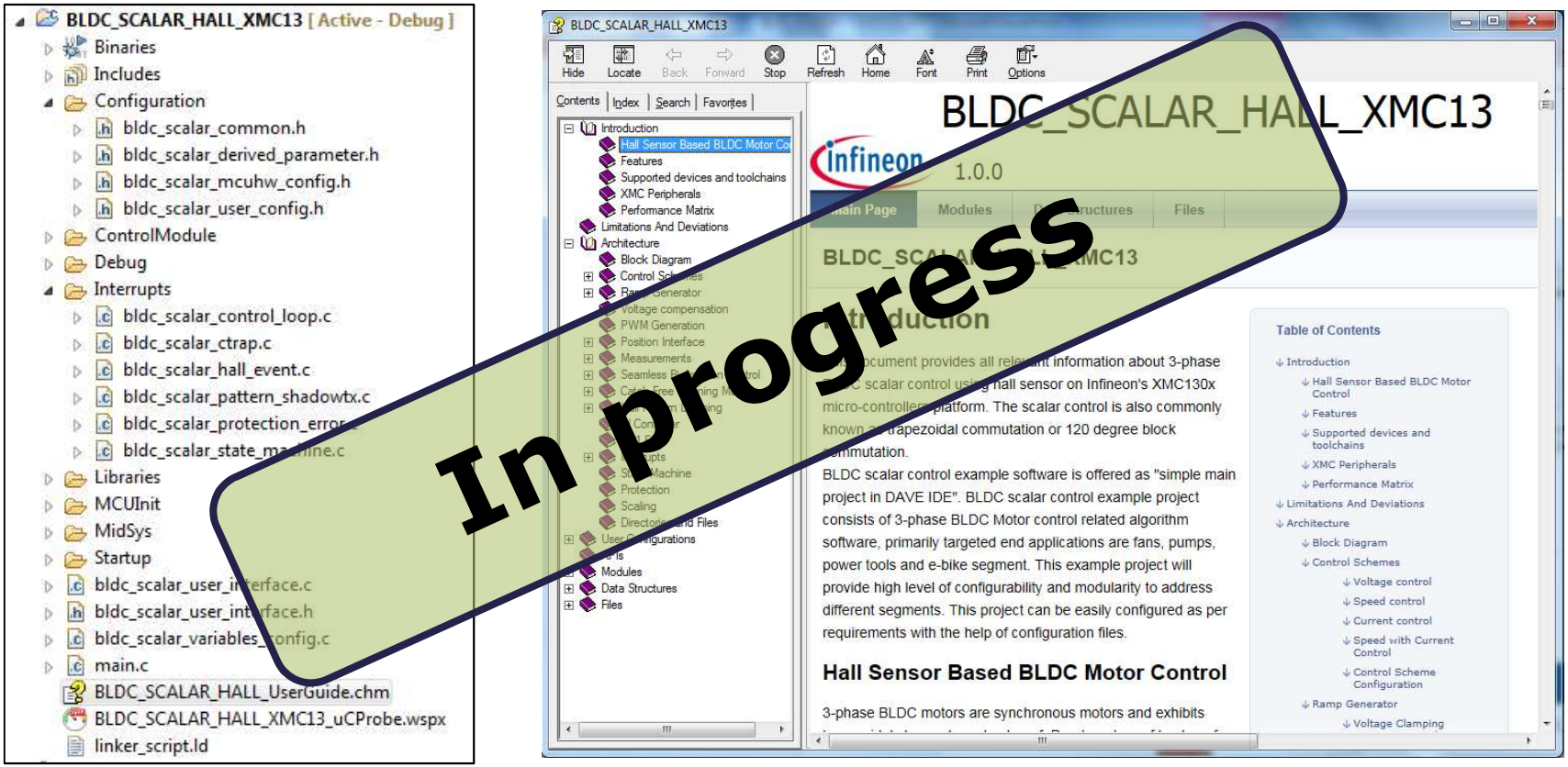
  http://www.infineon.com/DAVE

- For support:

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

References : Help Content

- Example SW user guide as chm format is part of this example SW



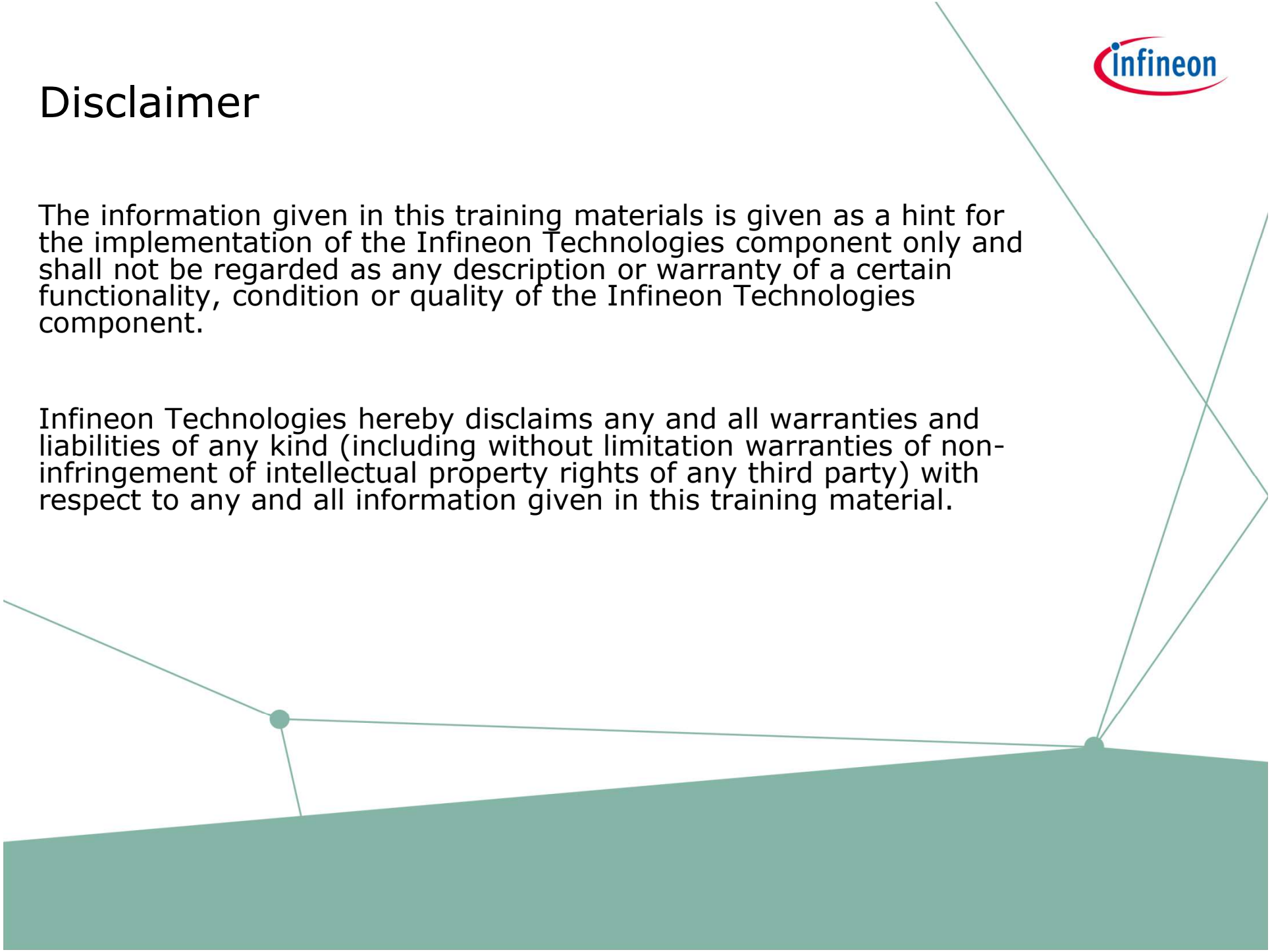
Glossary Abbreviations

- 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.

infineon