

# XMC13 BLDC Scalar Control Software based on 3 Hall sensor **Getting Started**

XMC™ Microcontrollers  
Oct 2016



# Agenda

1 Overview of BLDC Scalar Control SW

2 Software Overview

3 Hardware Overview

4 Tools Overview

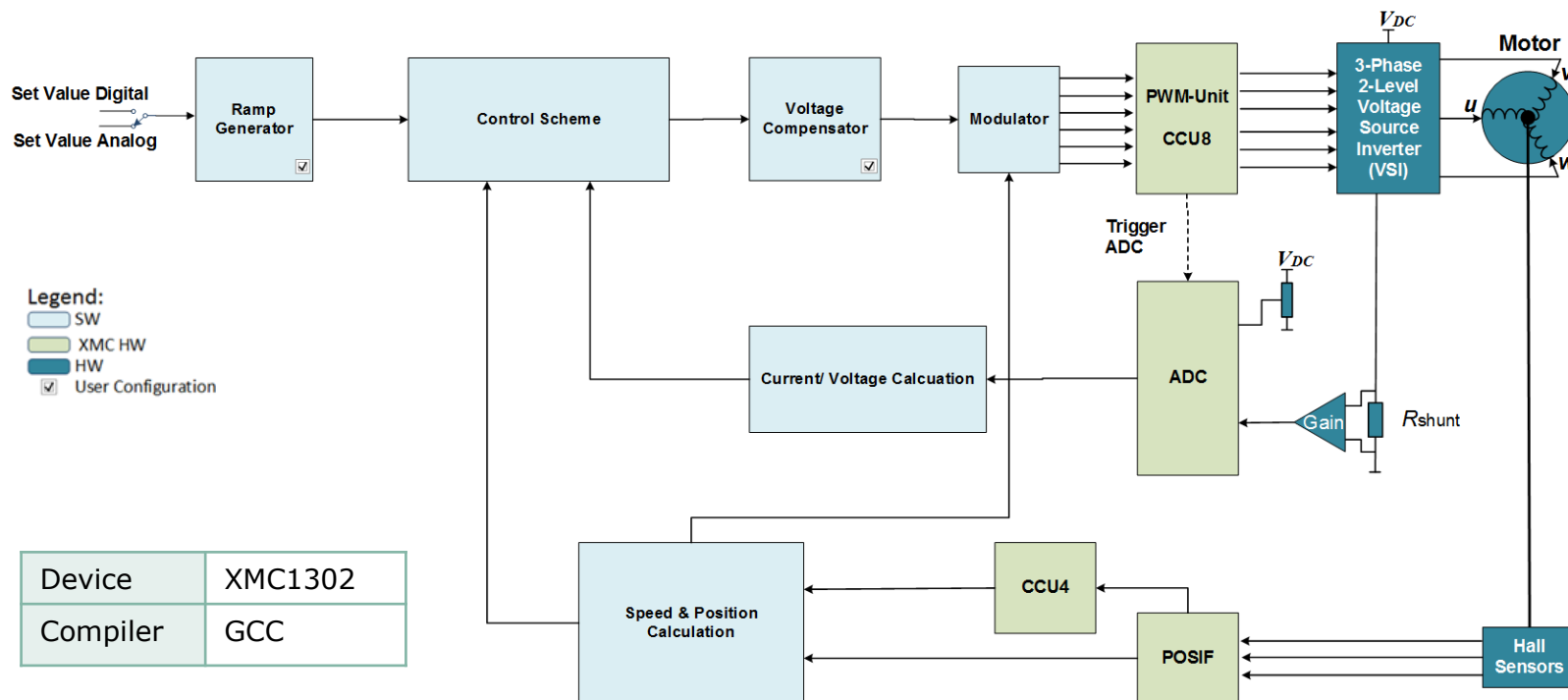
5 Getting Started

6 General Information

# Overview - BLDC Scalar Control SW

- › This document provides information about usage of BLDC scalar control example software on Infineon's XMC1300 series micro-controllers platform.
- › BLDC scalar control example software is offered as "simple main project in DAVE™ IDE".
- › BLDC scalar control example project consists of Hall based 3-Phase BLDC Motor control algorithm software, targeted end applications are fans, pumps, power tools 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

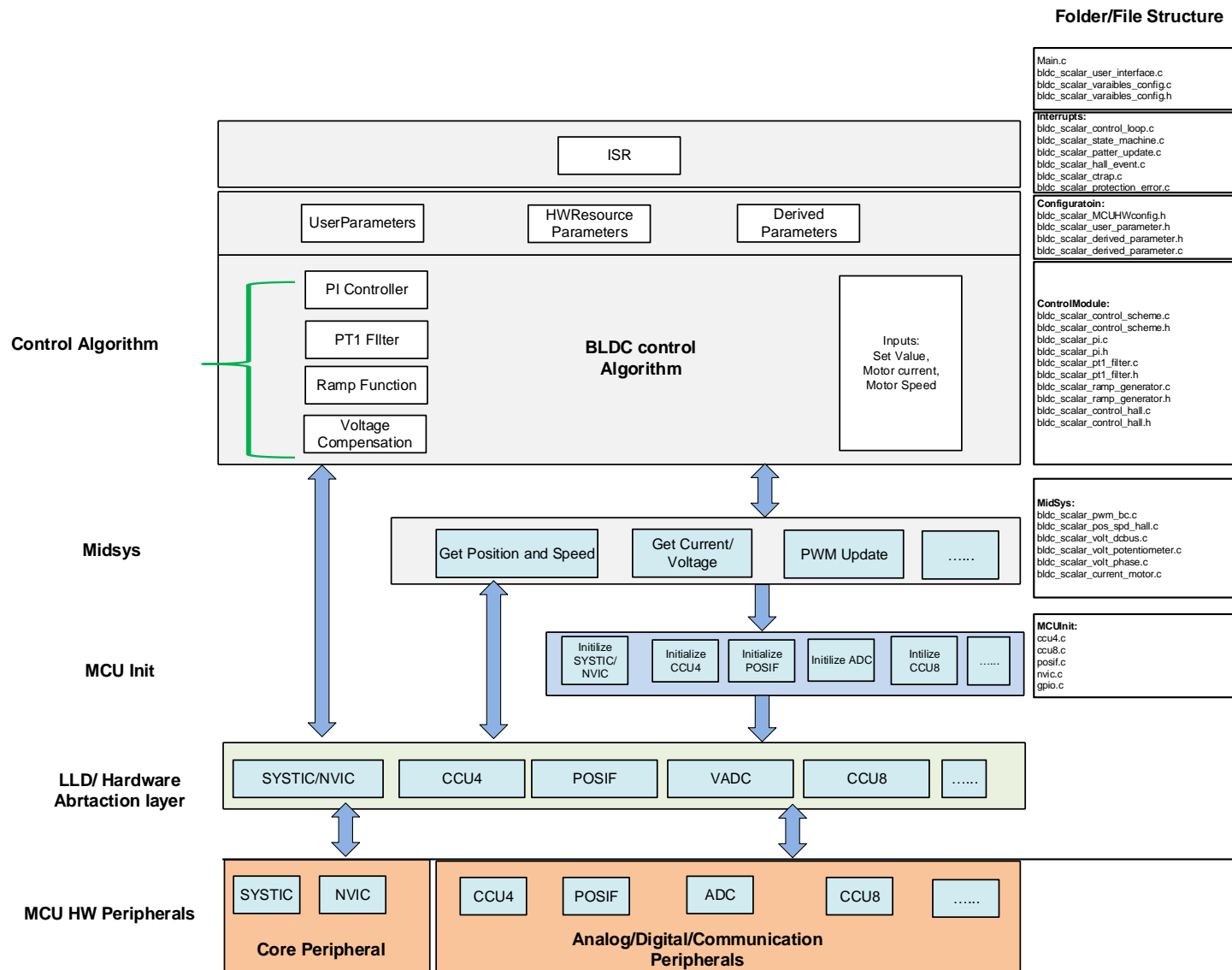


Software Blocks	Supported Options
<b>Control Scheme</b>	Open loop voltage control, speed control, current control and speed inner current control
<b>PWM Modulation (Modulator)</b>	High side modulation, low side modulation, high side with synchronous rectification
<b>Current/Voltage Measurement</b>	Direct DC link and average current measurement, DC link Voltage & Potentiometer (Analog Input)

# Software Overview – Key Features

Supported Features	Description
<b>Seamless bi-directional control</b>	Reverse the motor direction without stopping the motor
<b>On fly start-up</b>	Catch spinning motor at start-up without stop
<b>Adaptive Hall pattern learning</b>	Synchronise inverter commutation logic with Hall pattern
<b>Accurate measurement of speed (across wide range)</b>	Use floating pre-scalar
<b>Demagnetization blanking</b>	Remove spike in direct DC link current measurement
<b>DC bus voltage clamping</b>	Prevent over-voltage during fast braking
<b>Protection</b>	Stall Detection Over-current Short circuit Under/Over voltage C-trap with MCU hardware features

# Software Overview - Files Structure

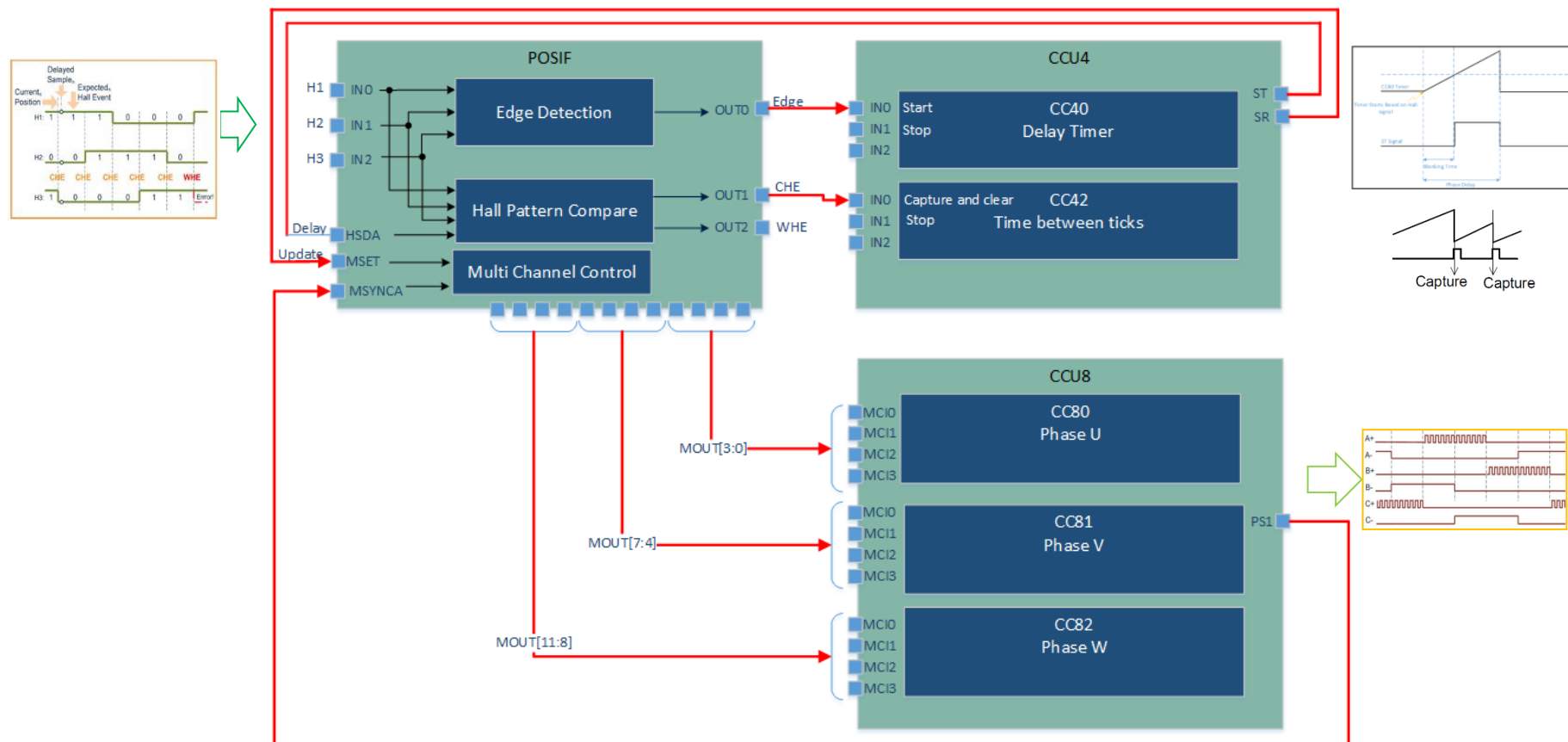


# Software Overview - XMC Peripheral usage

No	Resource	Resource usage	Purpose
1	CCU40 _CC40	Always	Phase delay and blanking for hall sampling
2	CCU40 _CC41	Fast Sync is enabled	Multi-channel Pattern synchronization
3	CCU40 _CC42	Always	Capture time interval between two hall
4	POSIF0	Always	Hall and MCM handling
5	CCU80_CC8x	Always	PWM Generation – Phase U
6	CCU80_CC8y	Always	PWM Generation – Phase V
7	CCU80_CC8z	Always	PWM Generation – Phase W
8	VADC Group A Queue A	Any ADC measurement is enabled	DC link direct/ Average current , DC link voltage, user defined and potentiometer measurement
9	NVIC	Always	Used for ISRs
10	SYSTICK	Always	Used for state machine

Note : x,y,z, A – Resource number based on configuration

# Software Overview - Peripheral Interconnection



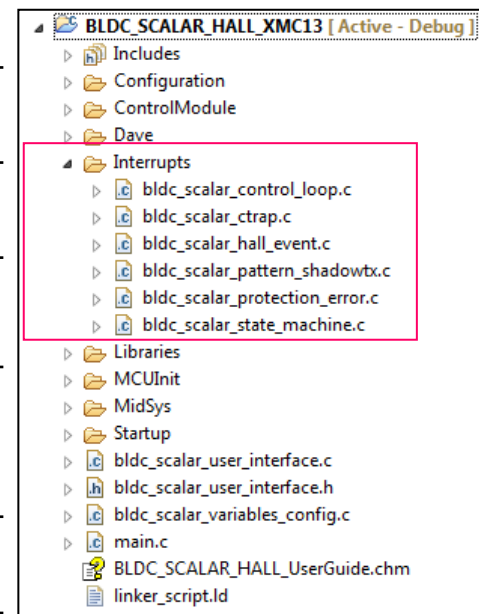


# Software Overview - Interrupt Service Routines

**Folder: Interrupts**

**File name:** bldc\_scalar\_state\_machine.c

Peripheral	Interrupt Subroutines (ISR)	NVIC node	Interval	Priority
VADC	Protection	19	Asynchronous	0
CCU8	CTRAP	26	Asynchronous	0
POSIF	HALL Event	27	Asynchronous (Hall edge event only for catch-free running)	1
POSIF	Pattern Update	28	Based upon speed: $1/(\text{electrical speed in Hz} * 6)$	1
CCU8	Control Loop	25	1/ PWM frequency	2
SYSTIMER	Scheduler	-1	1 mSec (configurable)	3



# Software Overview – Example Configuration

<b>Example Name</b>	BLDC_SCALAR_HALL_XMC13
<b>Kit Description</b>	Drive 3-phase Maxon's BLDC motor using XMC1000 motor control application kit
<b>Part Number</b>	KIT_XMC1X_AK_MOTOR_001
<b>Schemes</b>	<b>Default Configuration in Example Software</b>
<b>Control Scheme</b>	MOTOR0_BLDC_SCALAR_SPEED_CTRL
<b>PWM Modulation</b>	MOTOR0_BLDC_SCALAR_PWM_HIGHSIDE
<b>PWM frequency (Hz)</b>	20000
<b>Speed (rpm)</b>	2000
<b>Ramp up/down rate</b>	500
<b>Protection</b>	Over-current protection with direct DC link current measurement

## › Performance Matrix

Execution Time and Code Size		
	(us)	Code Size (bytes)
Control loop ISR	6.5	8296
Pattern update ISR	14.6	
Motor state machine - Normal state	5.92	

Execution Time and Code Size For default configuration

# Hardware Overview – Application Kit Package

## › Infineon's XMC1000 Motor Control Application Kit

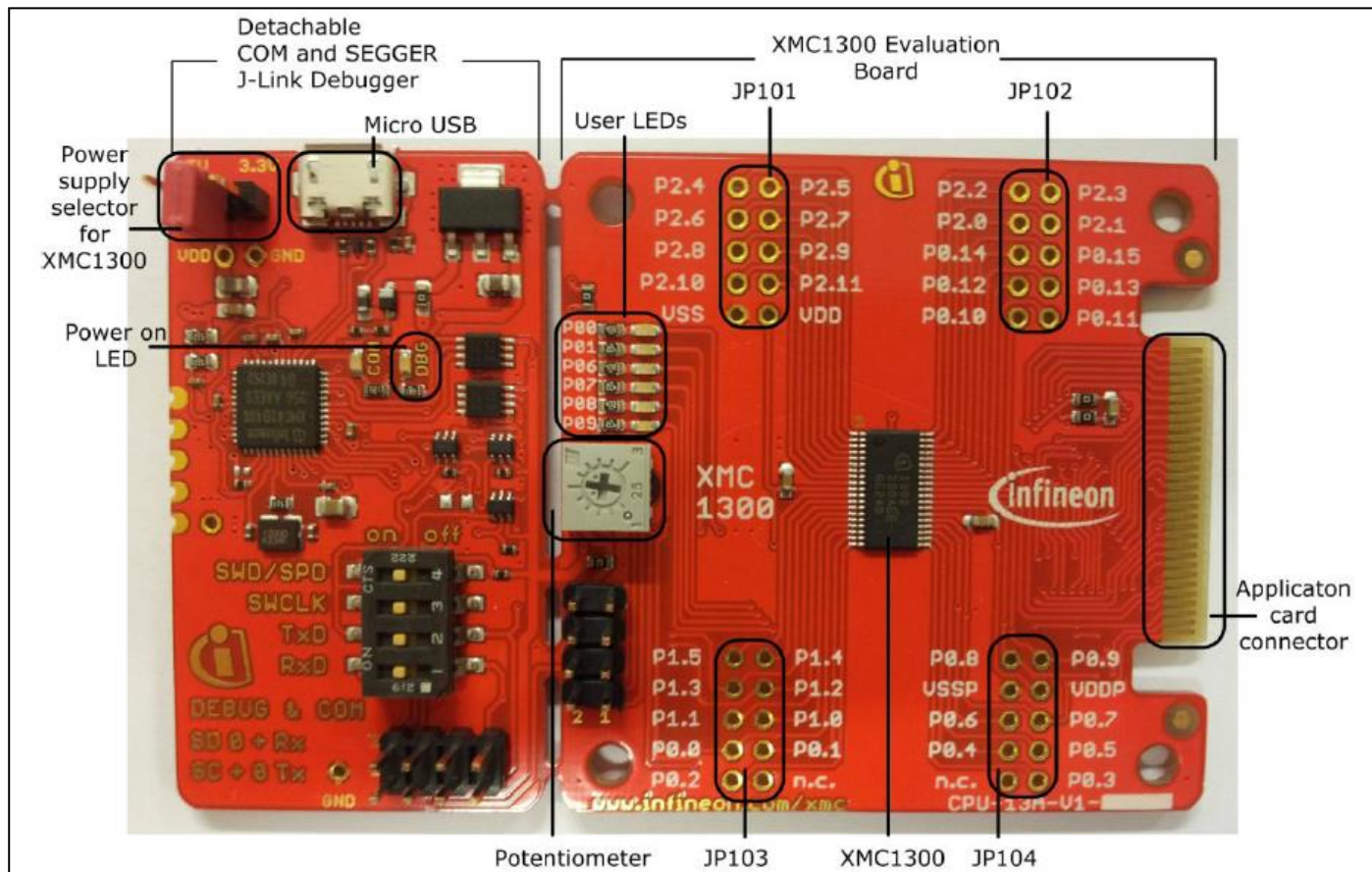
Item	Description
XMC1300 CPU Card	MCU board with XMC1300 and detachable SEGGER J-Link debug interface
PMSM Low Voltage 15W Motor Card	12 – 24V Up to 3A On board 3-phase motor (24V, 15W) with hall sensors
Accessories	Power Supply Adaptor (24V, 1A) Micro USB connector (1x)



**XMC1300 CPU Card**

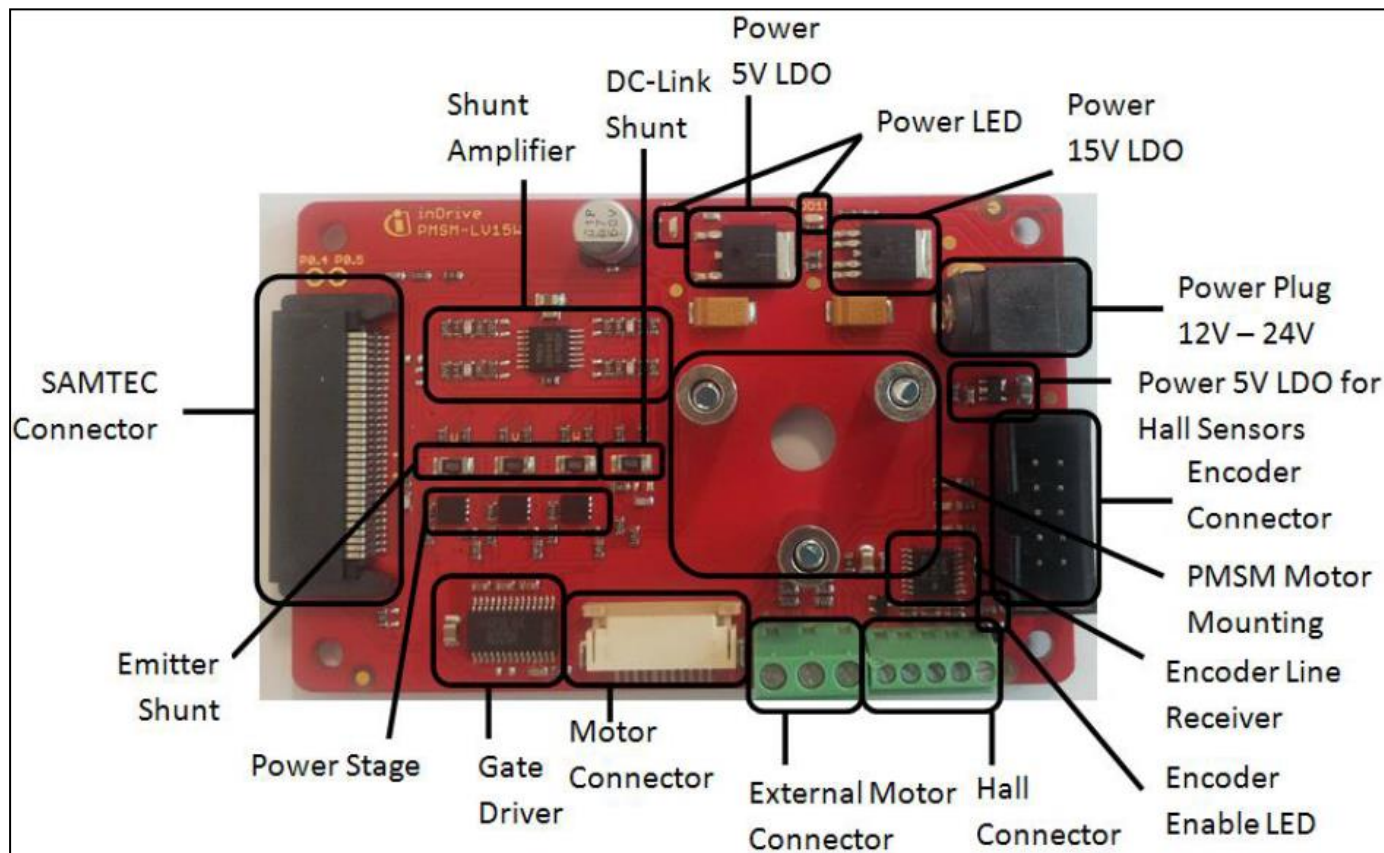
**PMSM Low Voltage 15W Motor Card**

## > 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

# Tools Overview

- › DAVE™ (V4.2.6 onwards)
  - Download DAVE™ installer package from <http://www.infineon.com/dave>
  - Download and unzip the installer package

DAVE™



Download

Free Eclipse based integrated development environment (IDE) including GNU C-compiler, debugger, comprehensive code repository, hardware resource management, and code generation plug-in.

*A complete download package is provided, including IDE, XMC™ Lib, DAVE™ APPs, EXAMPLES, and DAVE™ SDK.*

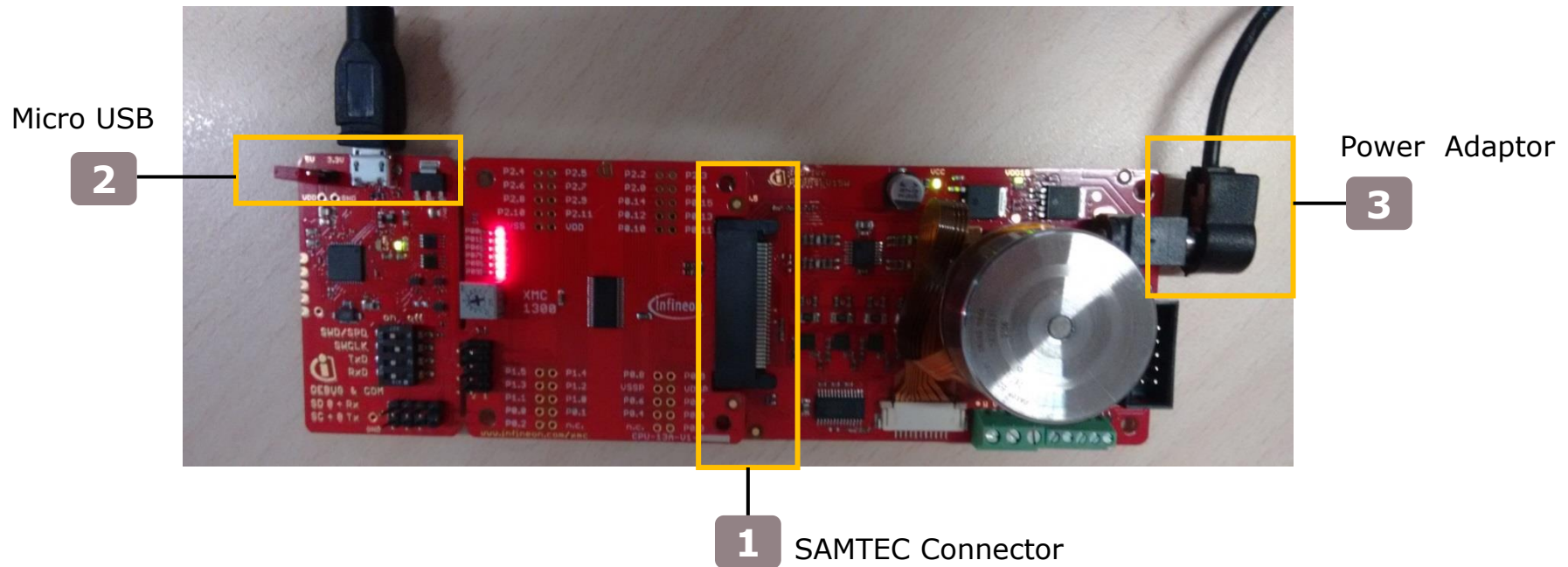
[DAVE™ Release Note](#)

- › µC/Probe™ XMC™ (v4.0.16.54 onwards) for Infineon industrial microcontrollers powered by Micrium®
  - › Download from µC/Probe™ XMC™ from DAVE home page [https://infineoncommunity.com/uC-Probe-XMC-software-download\\_ID712](https://infineoncommunity.com/uC-Probe-XMC-software-download_ID712)



# 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

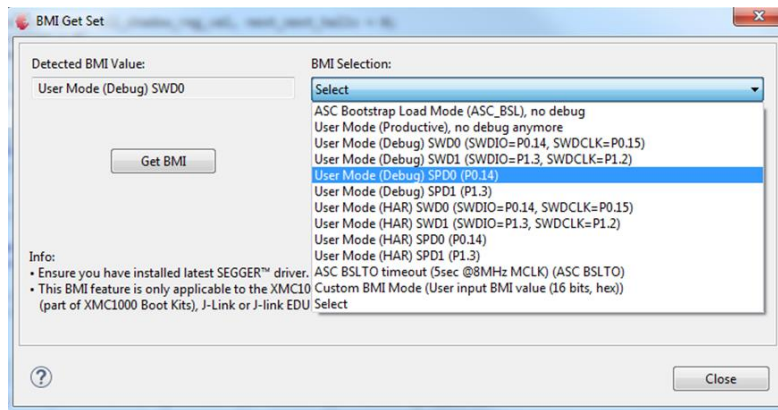




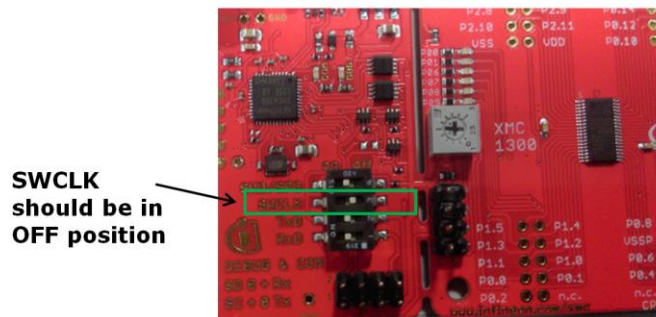
# Getting Started – Setting up the Board

- › **Note:** For this motor kit, one of the hall signal inputs are at P0.15. Therefore, to avoid conflict to the device, please ensure the following settings to the XMC1302 CPU Board.


1. In DAVE, select “*BMI Set Get*”  to update BMI to SPD0 mode

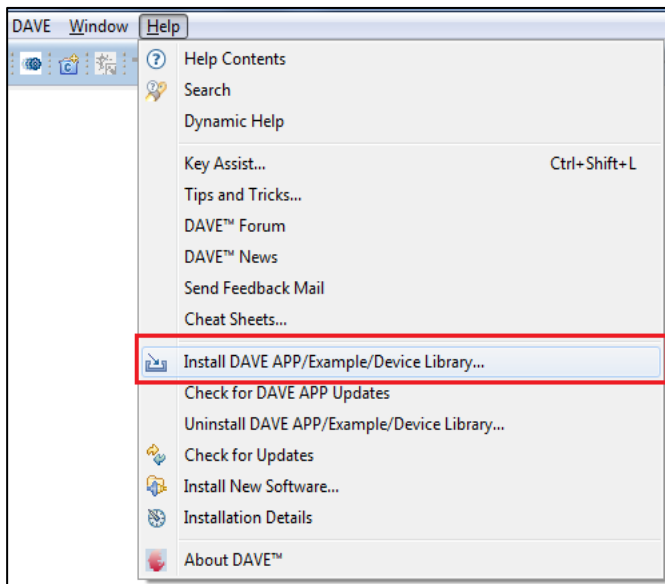


2. Set SWCLK on the dip switch to “**OFF**” position.



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

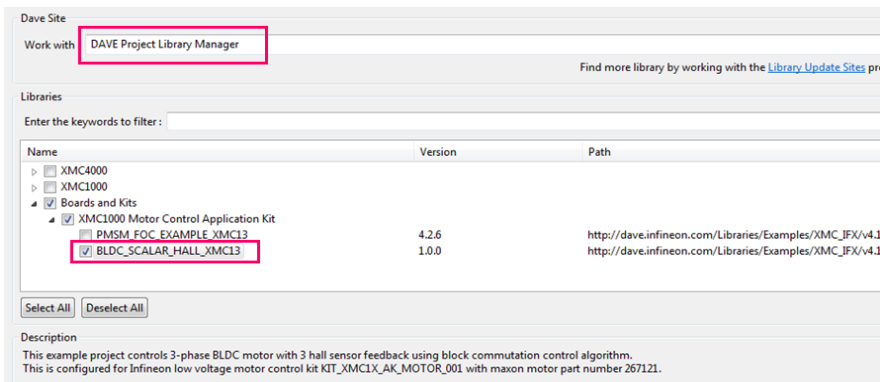
1. Open DAVE™ 
2. Install example project from DAVE:
  - Help → Install DAVE APP/Example/Device Library...



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

## 3. In the opened dialog “Dave Site”:

- In Option “Work With:”, select “*DAVE Project Library Manager*”
- In “Libraries”, select the project “*BLDC\_SCALAR\_HALL\_XMC13*”.



It is available at 2 locations.

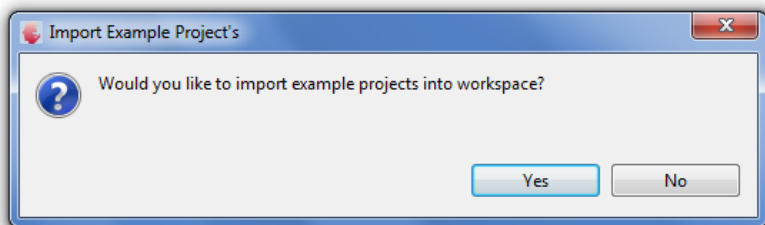
[#1] Boards and Kits →

*XMC1000 Motor Control Application Kit*

[#2] XMC1000 → XMC1300 Series →

*DAVE v4 Example Projects with XMC Lib → Motor Control*

## 4. Select “Yes” to import the example project in workspace



# Getting Started – Configure the Project [1/8]

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_config.h

## 1. Select the Motor Control Kit and BLDC motor

```
113- /******  
114  * Motor and power board selection  
115  *****/  
116 /* Board and motor selection */  
117- /**  
118  * Motor control kit  
119  * Options - KIT_XMC1X_AK_MOTOR_001, KIT_XMC750WATT_MC_AK_V1, KIT_CUSTOM  
120  */  
121 #define MOTOR0_BLDC_SCALAR_BOARD (KIT_XMC1X_AK_MOTOR_001)  
122- /**  
123  * BLDC motor  
124  * Options - MOTOR_EC_MAXON_267121, MOTOR_CUSTOM  
125  */  
126 #define MOTOR0_BLDC_SCALAR_MOTOR (MOTOR_EC_MAXON_267121)  
127-  
128 /* Motor Parameters */  
129 #if (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_EC_MAXON_267121)  
130 #define MOTOR0_BLDC_SCALAR_MOTOR_NO_LOAD_SPEED (4530U) /*!< No load speed of the motor in RPM */  
131 #define MOTOR0_BLDC_SCALAR_MOTOR_POLE_PAIRS (4U) /*!< Pole pairs */  
132-  
133 #elif (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_CUSTOM)  
134 #define MOTOR0_BLDC_SCALAR_MOTOR_NO_LOAD_SPEED (6200U) /*!< No load speed of the motor in RPM */  
135 #define MOTOR0_BLDC_SCALAR_MOTOR_POLE_PAIRS (4U) /*!< Pole pairs */  
136 #endif  
---
```

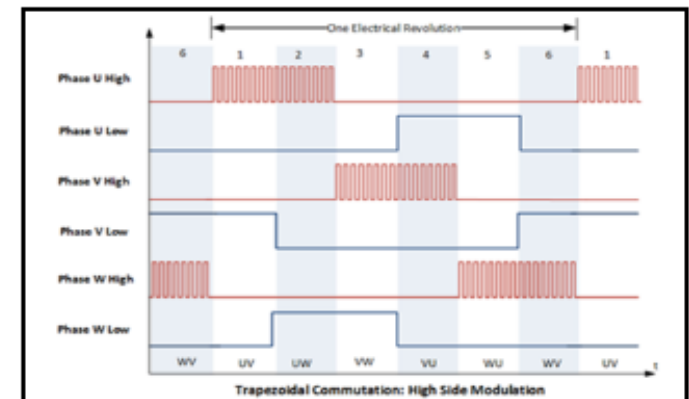
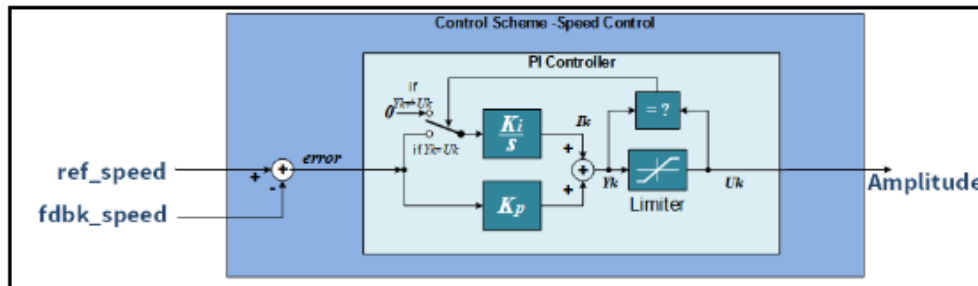
# Getting Started – Configure the Project [2/8]

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_config.h

## 2. Select the Control Scheme and PWM Modulation Scheme

```
138- /*****
139-  * Control scheme configurations
140-  *****/
141- /***** Control & PWM Modulation Scheme *****/
142- /**
143-  * Control scheme selection:
144-  * Options - BLDC_SCALAR_VOLTAGE_CTRL, BLDC_SCALAR_SPEED_CTRL, BLDC_SCALAR_CURRENT_CTRL, BLDC_SCALAR_SPEEDCURRENT_CTRL
145-  */
146- #define MOTOR0_BLDC_SCALAR_CTRL_SCHEME          (BLDC_SCALAR_SPEED_CTRL)
147- /**
148-  * PWM modulation scheme selection:
149-  * Options - BLDC_SCALAR_PWM_HIGHSIDE, BLDC_SCALAR_PWM_LOWSIDE, BLDC_SCALAR_PWM_HIGHSIDE_SYNCHRECTI
150-  */
151- #define MOTOR0_BLDC_SCALAR_MODULATION          (BLDC_SCALAR_PWM_HIGHSIDE)
152-
153-
154- /***** PWM Frequency and scheduler settings *****/
155- /* CPU clock (mclk) is configured to 32MHz and CCU peripheral clock(pclk) to 64 MHz [(double clock)] */
156- /** PWM switching frequency. Range: 1000 to 100000 Hz*/
157- #define MOTOR0_BLDC_SCALAR_PWM_FREQ          (20000.0F)
158- /** Scheduler interrupt period. Range: 1000 to 10000 us*/
159- #define MOTOR0_BLDC_SCALAR_SYSTICK_PERIOD    (1000.0F)
160-
```



High side PWM modulation

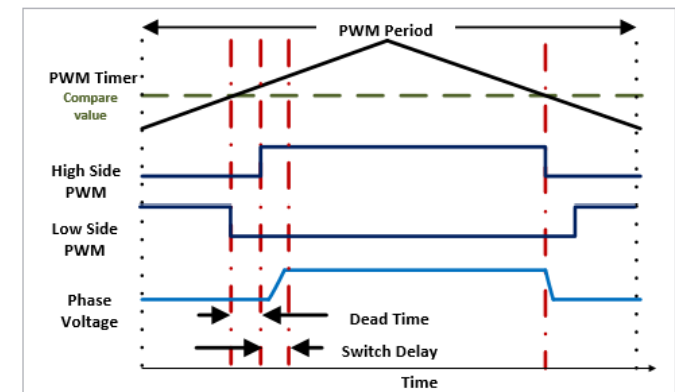
# Getting Started – Configure the Project [3/8]

## 3. Configure the Power Board

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_confiq.h

```
292 /*****
293  * Power board configurations
294  *****/
295 /***** Power board parameters *****/
296 #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
297 /* Power Inverter parameters */
298 #define MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT      (24.0F)      /*!< DC link voltage */
299 #define MOTOR0_BLDC_SCALAR_RISING_DEAD_TIME           (0.75F)      /*!< Dead time for rising edge in uSec*/
300 #define MOTOR0_BLDC_SCALAR_FALLING_DEAD_TIME          (0.75F)      /*!< Dead time for falling edge in uSec*/
301 #define MOTOR0_BLDC_SCALAR_SWITCH_DELAY               (0.75F)      /*!< Switch delay in uSec*/
302
303 #define MOTOR0_BLDC_SCALAR_HS_SWITCH_ACTIVE_LEVEL     (BLDC_SCALAR_ACTIVE_HIGH) /*!< Active level of the high side switch */
304 #define MOTOR0_BLDC_SCALAR_LS_SWITCH_ACTIVE_LEVEL     (BLDC_SCALAR_ACTIVE_HIGH) /*!< Active level of the low side switch */
305 #define MOTOR0_BLDC_SCALAR_INVERTER_ENABLE_CONF       (BLDC_SCALAR_INV_ACTIVE_HIGH) /*!< Active level of inverter enable. 0: disable, 1: enable */
306
307 /* ADC Measurement parameters */
308 #define MOTOR0_BLDC_SCALAR_VADC_REF_VOLTAGE           (5.0F)        /*!< Reference voltage of VADC conversion */
309 #define MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_OFFSET   (2.5F)        /*!< Amplifier offset voltage */
310 #define MOTOR0_BLDC_SCALAR_CURRENT_RSHUNT             (50.0F)       /*!< Current amplifier shunt resistor value in mOhms */
311 #define MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_GAIN     (16.4F)       /*!< Current amplifier gain */
312 #define MOTOR0_BLDC_SCALAR_VOLTAGE_DIVIDER_RATIO      (9.79F)       /*!< Voltage divider ratio in % for DC link voltage measurement */
313 /* end of #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
314
```



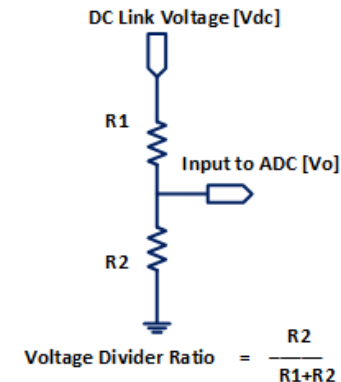
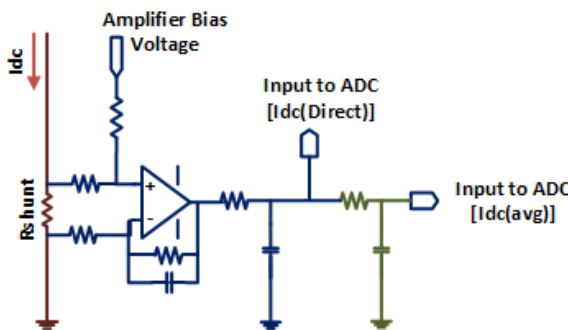
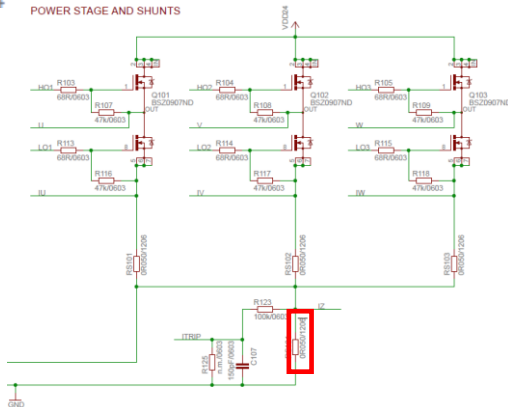
# Getting Started – Configure the Project [4/8]

## 4. Configure the Power Board

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_confiq.h

```
292 /*****
293  * Power board configurations
294  *****/
295 /***** Power board parameters *****/
296 #if (MOTOR0_BLDSCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
297 /* Power Inverter parameters */
298 #define MOTOR0_BLDSCALAR_NOMINAL_DC_LINK_VOLT      (24.0F)    /*!< DC link voltage */
299 #define MOTOR0_BLDSCALAR_RISING_DEAD_TIME          (0.75F)    /*!< Dead time for rising edge in uSec*/
300 #define MOTOR0_BLDSCALAR_FALLING_DEAD_TIME         (0.75F)    /*!< Dead time for falling edge in uSec*/
301 #define MOTOR0_BLDSCALAR_SWITCH_DELAY              (0.75F)    /*!< Switch delay in uSec*/
302
303 #define MOTOR0_BLDSCALAR_HS_SWITCH_ACTIVE_LEVEL     (BLDC_SCALAR_ACTIVE_HIGH) /*!< Active level of the high side swit
304 #define MOTOR0_BLDSCALAR_LS_SWITCH_ACTIVE_LEVEL     (BLDC_SCALAR_ACTIVE_HIGH) /*!< Active level of the low side switc
305 #define MOTOR0_BLDSCALAR_INVERTER_ENABLE_CONF       (BLDC_SCALAR_INV_ACTIVE_HIGH) /*!< Active level of inverter enable. (
306
307 /* ADC Measurement parameters */
308 #define MOTOR0_BLDSCALAR_VADC_REF_VOLTAGE           (5.0F)      /*!< Reference voltage of VADC conversion */
309 #define MOTOR0_BLDSCALAR_CURRENT_AMPLIFIER_OFFSET   (2.5F)      /*!< Amplifier offset voltage */
310 #define MOTOR0_BLDSCALAR_CURRENT_RSHUNT             (50.0F)     /*!< Current amplifier shunt resistor value in mOhms */
311 #define MOTOR0_BLDSCALAR_CURRENT_AMPLIFIER_GAIN     (16.4F)     /*!< Current amplifier gain */
312 #define MOTOR0_BLDSCALAR_VOLTAGE_DIVIDER_RATIO      (9.79F)     /*!< Voltage divider ratio in % for DC link voltage measur
313 /* end of #if (MOTOR0_BLDSCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
314
```





# Getting Started – Configure the Project [5/8]

## 5. Configure the Hall Pattern

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_confiq.h

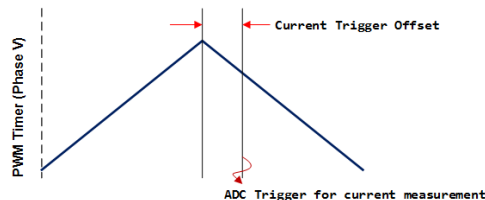
```
357- /******  
358  * Hall pattern and phase excitation pattern  
359  *****/  
360 #if (MOTOR0_BLDC_SCALAR_FEEDBACK == BLDC_SCALAR_3HALL)  
361  
362 #define MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ      (BLDC_SCALAR_HALL_SEQ_1) /*!< Select hall sequence for positive direction. (
```

```
363  
364 #if (MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ == BLDC_SCALAR_HALL_SEQ_1)  
365- /*
```

```
366  * Standard hall pattern for positive direction. Do NOT change this hall pattern values.  
367  * Update the phase excitation pattern corresponding to the hall pattern  
368  */  
369 #define MOTOR0_BLDC_SCALAR_HALL_PAT_A      (1U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
370 #define MOTOR0_BLDC_SCALAR_HALL_PAT_B      (3U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
371 #define MOTOR0_BLDC_SCALAR_HALL_PAT_C      (2U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
372 #define MOTOR0_BLDC_SCALAR_HALL_PAT_D      (6U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
373 #define MOTOR0_BLDC_SCALAR_HALL_PAT_E      (4U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
374 #define MOTOR0_BLDC_SCALAR_HALL_PAT_F      (5U)      /*!< (MSB)H3 H2 H1 (LSB)*/  
375  
376 #else  
378+ * Standard hall pattern for positive direction. Do NOT change this hall pattern values.  
388- #endif  
389  
390- /******  
391  /* Phase W, V, U excitation pattern */  
392  #if (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_EC_MAXON_267121)  
393  #define MOTOR0_BLDC_SCALAR_MC_PAT_A      (WH_VL_UOFF) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_A */  
394  #define MOTOR0_BLDC_SCALAR_MC_PAT_B      (WOFF_VL_UH) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_B */  
395  #define MOTOR0_BLDC_SCALAR_MC_PAT_C      (WL_VOFF_UH) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_C */  
396  #define MOTOR0_BLDC_SCALAR_MC_PAT_D      (WL_VH_UOFF) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_D */  
397  #define MOTOR0_BLDC_SCALAR_MC_PAT_E      (WOFF_VH_UL) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_E */  
398  #define MOTOR0_BLDC_SCALAR_MC_PAT_F      (WH_VOFF_UL) /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_F */  
399
```

## 6. Configure the ADC Trigger

```
446- #define MOTOR0_BLDC_SCALAR_CURRENT_TRIGGER_OFFSET      (MOTOR0_BLDC_SCALAR_SWITCH_DELAY) /*!< By default, current trigger
```





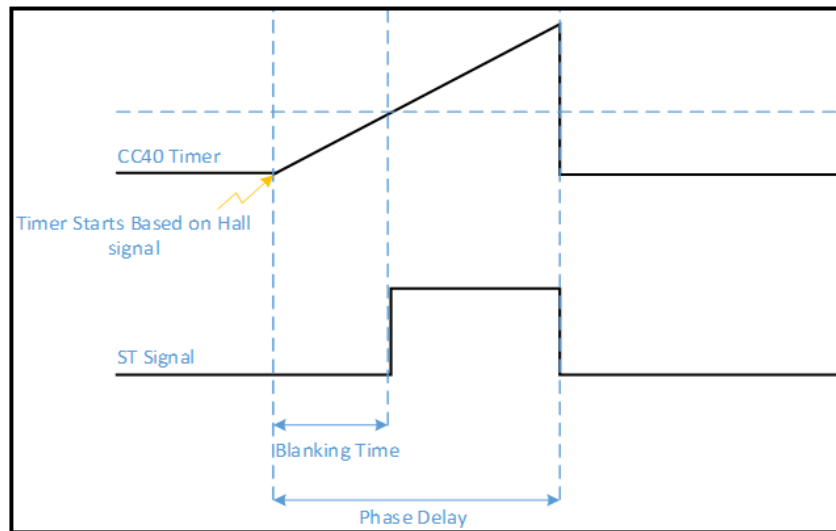
# Getting Started – Configure the Project [6/8]

**Folder: Configuration**

**File name:** bldc\_scalar\_user\_config.h

## 7. Configure the Hall sensor feedback

```
415 /*****
416  * Hall sensor feedback configurations
417  *****/
418 /* CCU4 clock is 2 MHz (clock time: 500 nS)*/
419 #if (MOTOR0_BLDC_SCALAR_FEEDBACK == BLDC_SCALAR_3HALL)
420 #define MOTOR0_BLDC_SCALAR_BLANKING_TIME (1.0F) /*!< Hall edge re-sampling time in uSec. Range:
421 #define MOTOR0_BLDC_SCALAR_PH_DELAY_TIME (1.5F) /*!< Time delay to update multi-channel pattern in
422                                             Phase delay time should be greater than blanking
423 #endif /* end of #if (MOTOR0_BLDC_SCALAR_FEEDBACK == BLDC_SCALAR_3HALL) */
424
```



CCU4 timer for hall edge detection

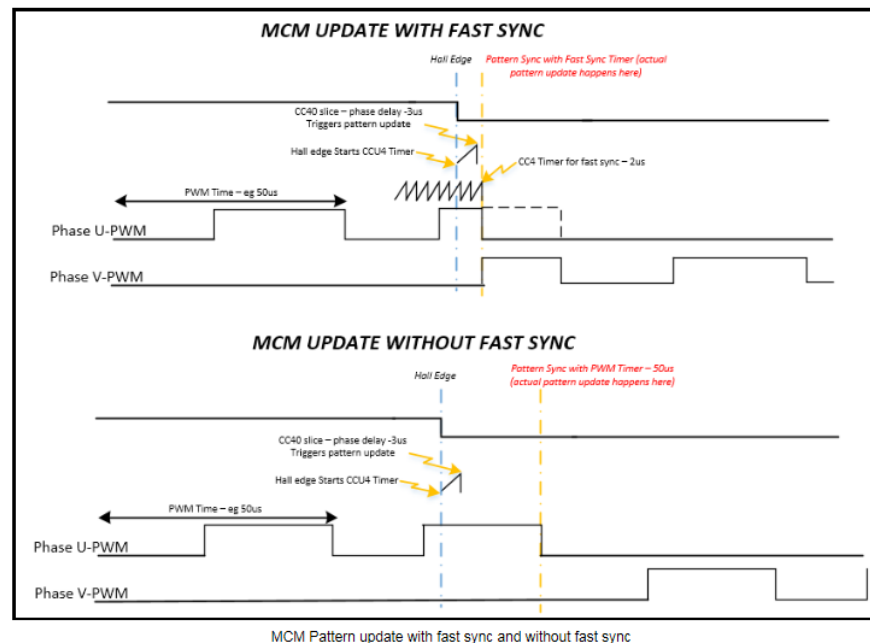
# Getting Started – Configure the Project [7/8]

## 8. Configure the MCM Transfer

**Folder:** Configuration

**File name:** bldc\_scalar\_user\_config.h

```
470 #define MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNC_CCU4 (1U) /*!< Enable (1)/disable (0)
471 #if (MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNC_CCU4 == 1U)
472 #define MOTOR0_BLDC_SCALAR_MCM_SYNCTRANSFER_TIME (2.0F) /*!< Multi-channel pattern
473 #endif /* end of #if (MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNC_CCU4 == 1U) */
474
```




# Getting Started – Configure the Project [8/8]

**Folder: Configuration**

**File name:** bldc\_scalar\_user\_config.h

## 9. Enable the Hall pattern learning

```
494 #define MOTOR0_BLDC_SCALAR_ENABLE_HALL_LEARNING (1U) /*!< Enable/disable hall pattern learning */
495 #if (MOTOR0_BLDC_SCALAR_ENABLE_HALL_LEARNING == 1U)
496 #define MOTOR0_BLDC_SCALAR_OPEN_LOOP_VOLTAGE (5.0F) /*!< Open loop voltage to be applied in % v
497 #define MOTOR0_BLDC_SCALAR_OPEN_LOOP_SPEED (2.0F) /*!< Speed to be applied in % with respect
498 #endif /* if(MOTOR0_BLDC_SCALAR_ENABLE_HALL_LEARNING == 1U) */
```

 closedloop_mc_pattern	uint16_t [7]	0x200007ca <BLDC_SCALAR_HallLearni...
(x)= closedloop_mc_pattern[0]	uint16_t	1 Sequence 1 or 2
(x)= closedloop_mc_pattern[1]	uint16_t	0x201 (Hex) WL_VOFF_UH
(x)= closedloop_mc_pattern[2]	uint16_t	0x210 (Hex) WL_VH_UOFF
(x)= closedloop_mc_pattern[3]	uint16_t	0x12 (Hex) WOFF_VH_UL
(x)= closedloop_mc_pattern[4]	uint16_t	0x102 (Hex) WH_VOFF_UL
(x)= closedloop_mc_pattern[5]	uint16_t	0x120 (Hex) WH_VL_UOFF
(x)= closedloop_mc_pattern[6]	uint16_t	0x21 (Hex) WOFF_VL_UH

Capture close loop multi-channel pattern sequence

Hall MACRO	Hall Pattern	MC MACRO	Configurable MC Pattern
BLDC_SCALAR_HALL_PAT_A	1	BLDC_SCALAR_MC_PAT_A	WL_VOFF_UH
BLDC_SCALAR_HALL_PAT_B	5	BLDC_SCALAR_MC_PAT_B	WL_VH_UOFF
BLDC_SCALAR_HALL_PAT_C	4	BLDC_SCALAR_MC_PAT_C	WOFF_VH_UL
BLDC_SCALAR_HALL_PAT_D	6	BLDC_SCALAR_MC_PAT_D	WH_VOFF_UL
BLDC_SCALAR_HALL_PAT_E	2	BLDC_SCALAR_MC_PAT_E	WH_VL_UOFF
BLDC_SCALAR_HALL_PAT_F	3	BLDC_SCALAR_MC_PAT_F	WOFF_VL_UH

hall\_pattern\_update

**Note:** For a new motor where the hall pattern is not known, this configuration can be enabled for hall pattern learning.

# Getting Started – Add code support for uCProbe [1/3]

## 1. Initialize the uCProbe before starting motor

**Folder:** -

**File name:** main.c

```
89
90 int main(void)
91 {
92     /* Initialization */
93     Motor0_BLDC_SCALAR_Init();
94
95     Motor0_BLDC_SCALAR_Flash_Var_Init();
96
97     #if (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
98         Motor0_BLDC_SCALAR_uCProbe_Init();
99     #endif
100
101
102     // /* Start the motor */
103     Motor0_BLDC_SCALAR_MotorStart();
104
105
106     /* Placeholder for user application code. The while loop below can be r
107     while (1U)
108     {
109
110     }
111 }
112
```

# Getting Started – Add code support for uCProbe [2/3]

**Folder: Interrupts**

**File name:**

bldc\_scalar\_state\_machine.c

## 2. Add uCProbe scheduler in motor state machine

- › Motor control state machine is called on each SysTick Interrupt
- › uCProbe Scheduler is called on each scheduler tick

```
137 void SysTick_Handler(void)
138 {
139     /* Call motor control state machine */
140     Motor0_BLDC_SCALAR_MSM();
141 }
142
143 /**
144  * @}
145  */
146 /**
147  * @}
148  */
149
150 RAM_ATTRIBUTE void Motor0_BLDC_SCALAR_MSM(void)
151 {
152     switch (Motor0_BLDC_SCALAR.msm_state)
153     {
154         case BLDC_SCALAR_MSM_NORMAL_OPERATION:
155             Motor0_BLDC_SCALAR_MSM_NORMAL_OPERATION_Func();
156             break;
157
158 #if (MOTOR0_BLDC_SCALAR_ENABLE_BIDIRECTIONAL_CTRL == 0U)
159
160 #if (MOTOR0_BLDC_SCALAR_ENABLE_CATCH_FREE == 1U)
161
162 #if (MOTOR0_BLDC_SCALAR_ENABLE_BOOTSTRAP == 1U)
163
164         case BLDC_SCALAR_MSM_ERROR:
165             Motor0_BLDC_SCALAR_MSM_ERROR_Func();
166             break;
167
168         case BLDC_SCALAR_MSM_START:
169             Motor0_BLDC_SCALAR_MSM_START_Func();
170             break;
171
172     }
173 }
```

```
183
184 #if (MOTOR0_BLDC_SCALAR_HALL_LEARNING == 1U)
185     case BLDC_SCALAR_MSM_HALL_LEARNING:
186         Motor0_BLDC_SCALAR_MSM_HALL_LEARNING_Func();
187         break;
188 #endif
189
190     case BLDC_SCALAR_MSM_STOP:
191         Motor0_BLDC_SCALAR_MotorStop();
192         break;
193
194     default:
195         break;
196 }
197
198 if (Motor0_BLDC_SCALAR.error_status != 0U)
199 {
200     Motor0_BLDC_SCALAR.msm_state = BLDC_SCALAR_MSM_ERROR;
201 }
202
203 #if (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
204     Motor0_BLDC_SCALAR_uCProbe_Scheduler();
205 #endif
206 }
207
208 }
```

# Getting Started – Add code support for uCProbe [3/3]

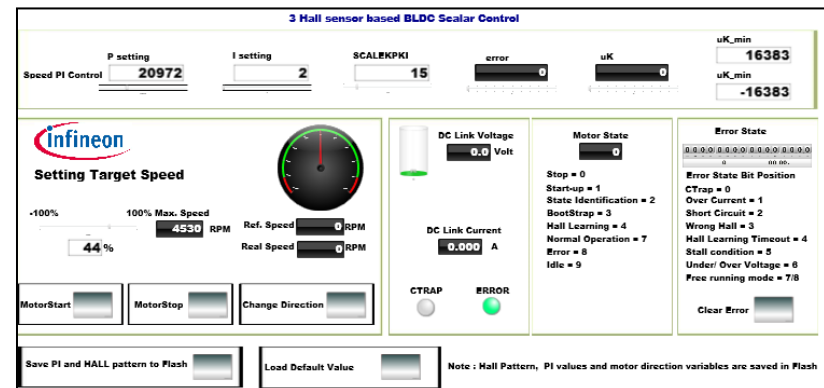
**Folder: uCProbe**

**File name: ucProbe.c**



## 3. Add uCProbe scheduler in motor state machine

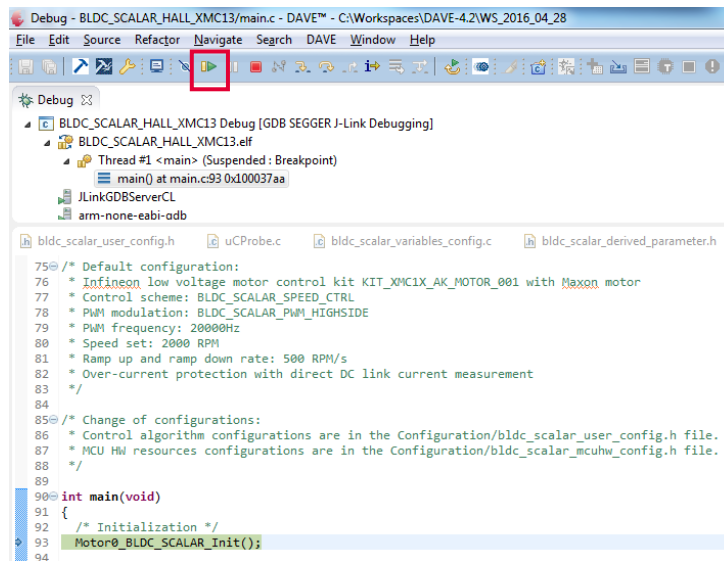
```
77 #if ((MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE==1))
78 /*UCprobe scheduler function to handle ucprobe comments from UI */
79 void Motor0_BLDC_SCALAR_uCProbe_Scheduler(void)
80 {
81     switch(Motor0_BLDC_SCALAR_ucprobe.control_word)
82     {
83     case 1: /* Start the motor */
84         Motor0_BLDC_SCALAR_ucprobe.control_word=0;
85         Motor0_BLDC_SCALAR_MotorStart();
86         break;
87
88     case 2: /*Stop the motor*/
89         Motor0_BLDC_SCALAR_ucprobe.control_word=0;
90         Motor0_BLDC_SCALAR_MotorStop();
91         break;
92
93     case 3: /*Clear Error state*/
94         Motor0_BLDC_SCALAR_ucprobe.control_word=0;
95         Motor0_BLDC_SCALAR_ClearErrorState();
96         break;
97
98     case 4: /*Clear flash and load default value into flash*/
99         Motor0_BLDC_SCALAR_ucprobe.control_word=0;
100         Motor0_BLDC_SCALAR_ucprobe.user_config[0] =0;
101         Motor0_BLDC_SCALAR_Write_Default_value();
102         Motor0_BLDC_SCALAR_uCProbe_Write_Flash();
103         break;
```

› uCProbe scheduler routine support control code to control the motor



# 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

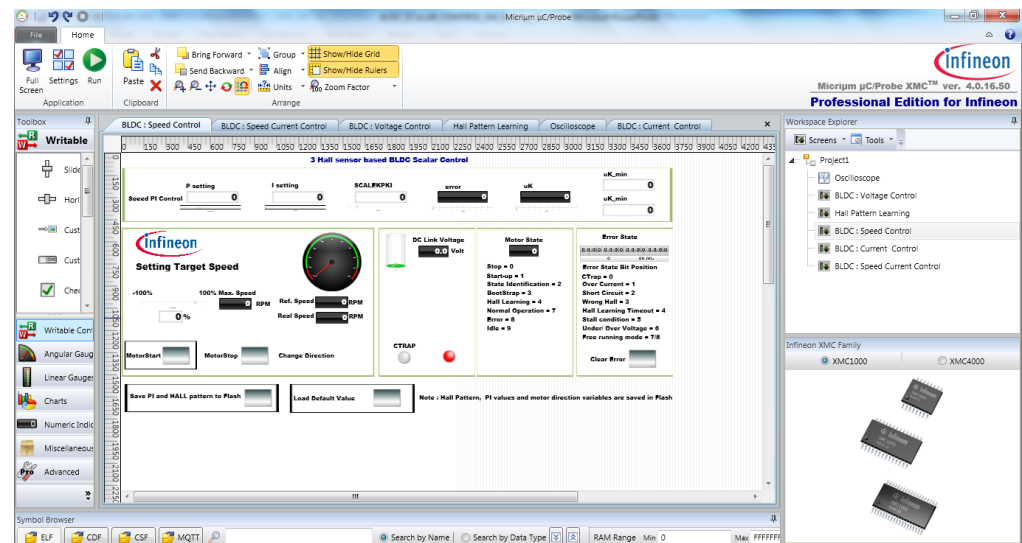
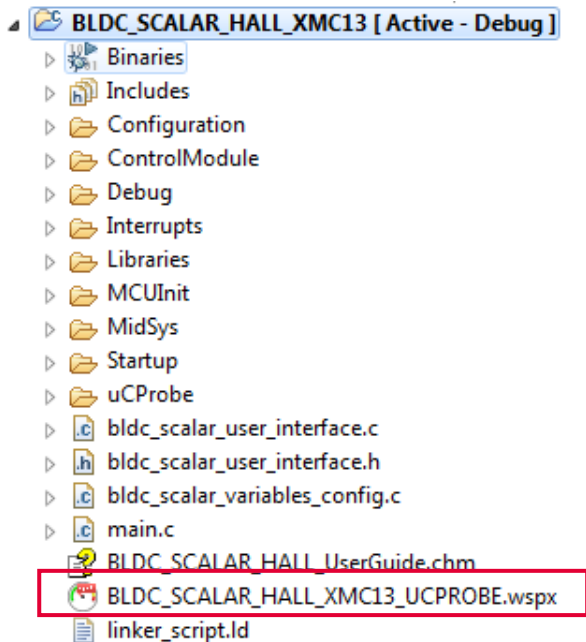


## Observation:

- › Motor should ramp to 2000RPM with ramp rate of 500RPM/s.

# Getting Started – Interface with $\mu$ C/Probe [1/6]

- Update of the motor and monitoring motor parameters can be executed using  $\mu$ C/Probe™ XMC™
1. In “BLDC\_SCALAR\_HALL\_XMC13” example project , open  $\mu$ C/Probe™ XMC™ project file.

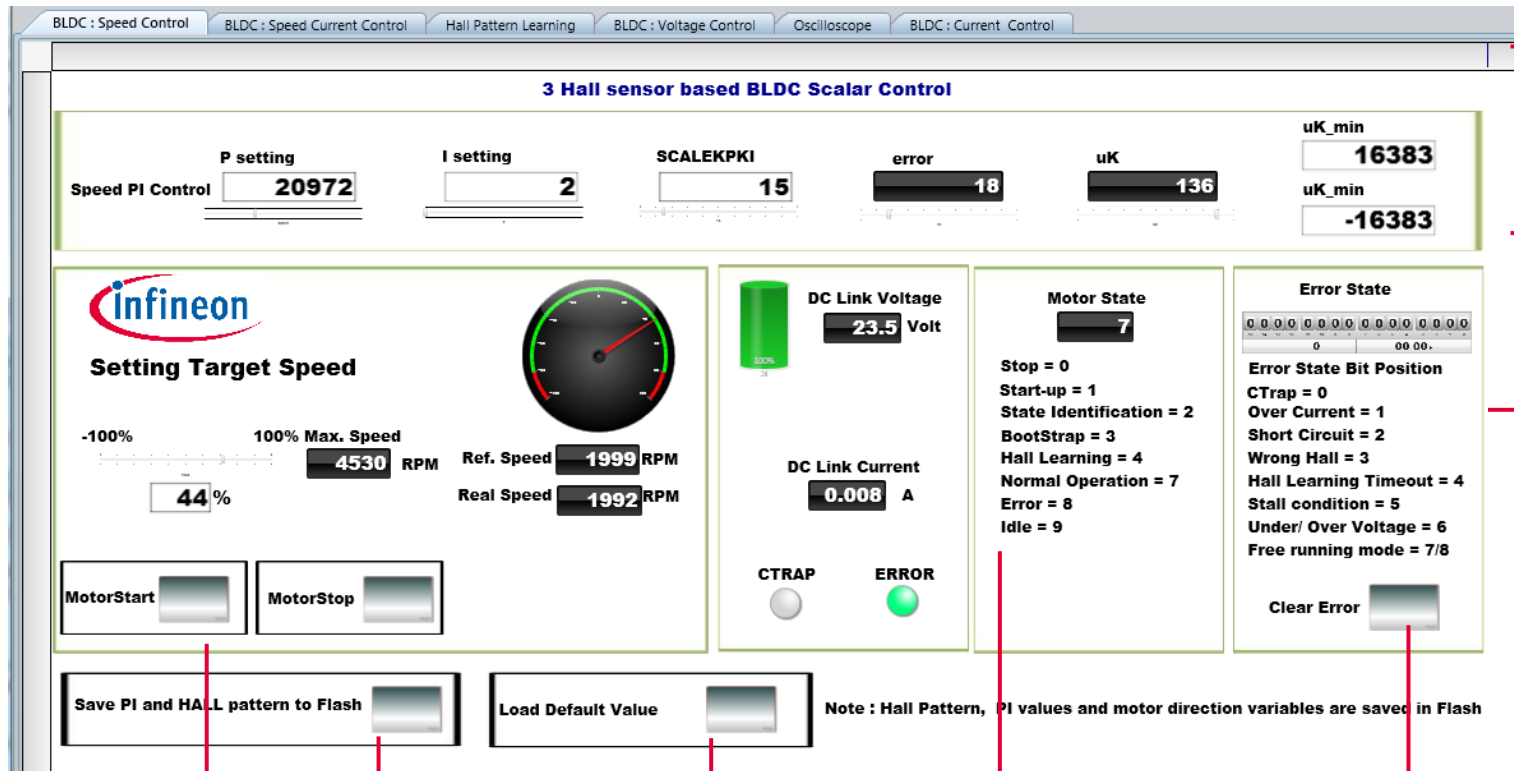


**Note:** As the BMI mode is set to SPD0 mode, the program needs to be started in Debug mode before connecting to the uCProbe project.



# Getting Started

## Interface with $\mu$ C/Probe [2/6]



Fine tune the PI value to get optimum motor behaviour

Error state Indication

Save the values into flash

Start, Stop and change direction of motor by clicking respective buttons


Load the default values into flash

Motor state Indication

In case of Error condition, can clear the error flag in the SW by click this button

# Getting Started – Interface with $\mu$ C/Probe [3/6]

## 2. Click the 'Run' button



The screenshot displays the Infineon  $\mu$ C/Probe Professional Edition software interface. The top toolbar contains a red box around the 'Run' button (a green play icon). The main workspace shows the 'BLDC : Speed Control' screen, which includes various control parameters and status indicators.

**Writables Controls:**

- Slider
- Horizontal Slider
- Custom Slider
- Custom Switch
- Checkbox
- Push Button
- Toggle Button

**BLDC : Speed Control**

**3 Hall sensor based BLDC Scalar Control**

**Speed PI Control**

- P setting: 0
- I setting: 0
- SCALEPKI: 0
- error: 0
- uK: 0
- uK\_min: 0

**Setting Target Speed**

- 100% to 100% Max. Speed slider
- 0 RPM
- Ref. Speed: 0 RPM
- Real Speed: 0 RPM

**MotorStart** **MotorStop**

**Save PI and HALL pattern to Flash** **Load Default Value**

**Note : Hall Pattern, PI values and motor direction variables are saved in Flash**

**DC Link Voltage**: 0.0 Volt

**DC Link Current**: 0.000 A

**CTR** **ERROR**

**Motor State**

- Stop = 0
- Start-up = 1
- State Identification = 2
- BootStrap = 3
- Hall Learning = 4
- Normal Operation = 7
- Error = 8
- Idle = 9

**Error State**

- Error State Bit Position: 0000000000000000
- CTrap = 0
- Over Current = 1
- Short Circuit = 2
- Wrong Hall = 3
- Hall Learning Timeout = 4
- Stall condition = 5
- Under/ Over Voltage = 6
- Free running mode = 7/8

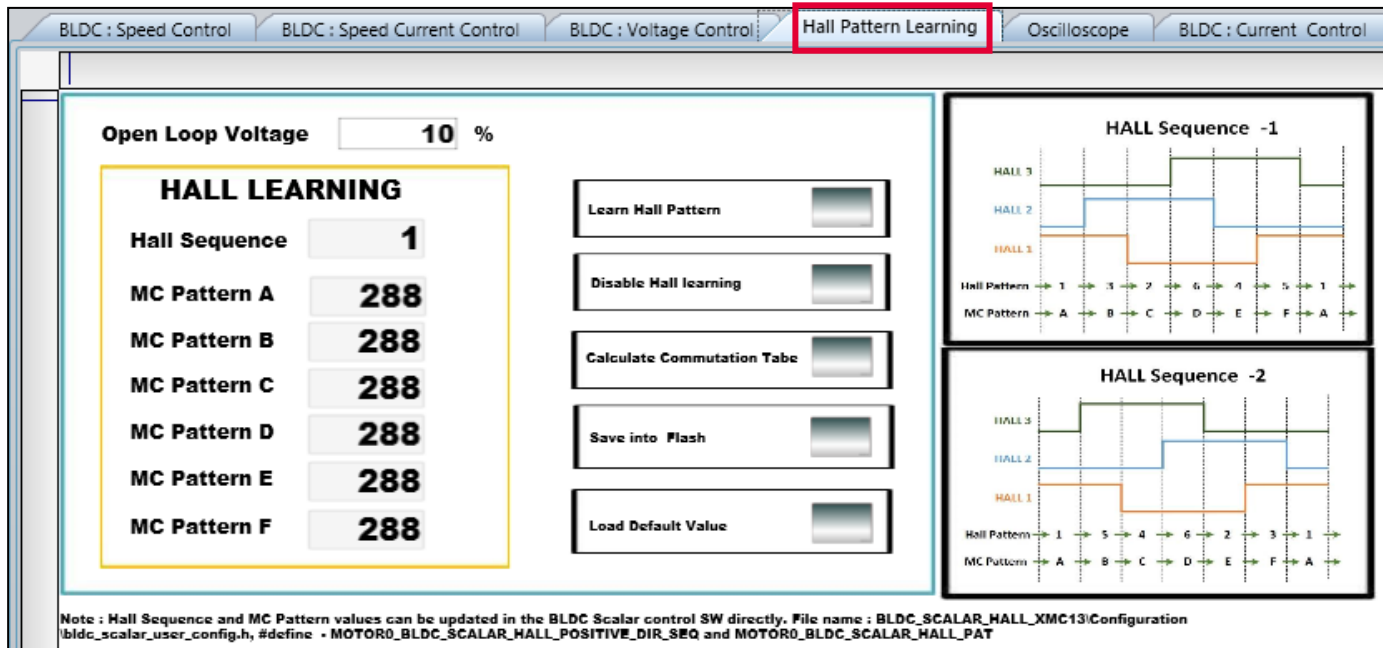
**Clear Error**

**Symbol Browser**

Name	Type	Size	Size Filtered	Memory Address
BLDC_SCALAR_HALL_XMC13.elf	N/A	9,508	9,508	N/A

# Getting Started – Interface with $\mu$ C/Probe [4/6]

3. Go to Tab: HALL Pattern Learning. This is used to find the relation between HALL and commutation pattern
4. Set the Open Loop Voltage to 10%
5. Select button "Learn Hall Pattern" to start the Hall Learning
6. Once the Hall learning is completed, the pattern is displayed.
7. If required, select "Save to Flash" to save the commutation table into the Flash.



The screenshot displays the 'Hall Pattern Learning' tab within the BLDC Scalar control software. The interface includes a top navigation bar with tabs for 'BLDC : Speed Control', 'BLDC : Speed Current Control', 'BLDC : Voltage Control', 'Hall Pattern Learning' (highlighted), 'Oscilloscope', and 'BLDC : Current Control'.

**Open Loop Voltage** is set to **10 %**.

**HALL LEARNING** section:

- Hall Sequence**: 1
- MC Pattern A**: 288
- MC Pattern B**: 288
- MC Pattern C**: 288
- MC Pattern D**: 288
- MC Pattern E**: 288
- MC Pattern F**: 288

Buttons on the right side of the Hall Learning section:

- Learn Hall Pattern
- Disable Hall learning
- Calculate Commutation Table
- Save into Flash
- Load Default Value

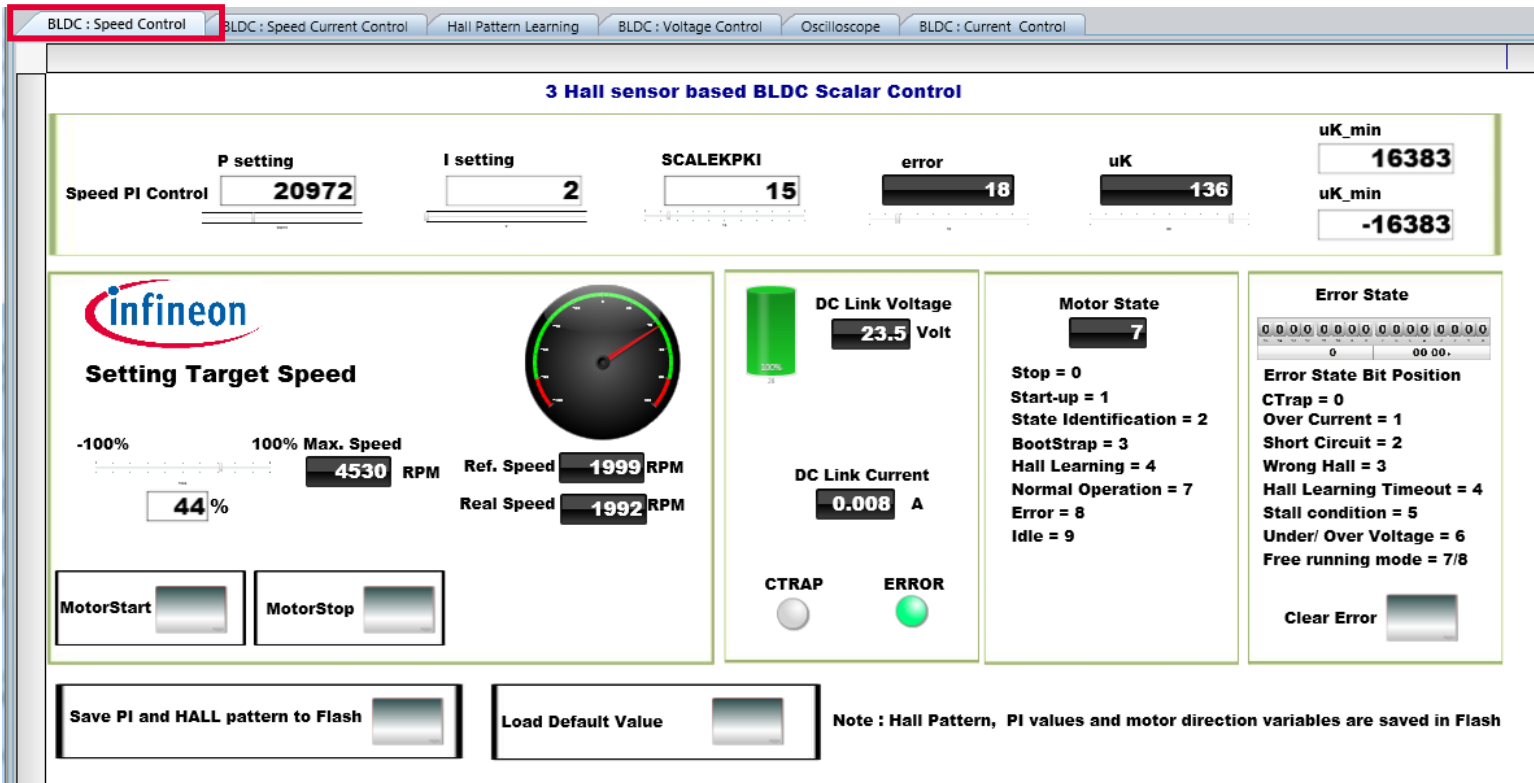
Two timing diagrams are shown on the right:

- HALL Sequence -1**: Shows Hall signals (HALL 3, HALL 2, HALL 1) and MC Pattern (A, B, C, D, E, F) over time. The Hall Pattern sequence is 1, 3, 2, 6, 4, 5, 1.
- HALL Sequence -2**: Shows Hall signals (HALL 3, HALL 2, HALL 1) and MC Pattern (A, B, C, D, E, F) over time. The Hall Pattern sequence is 1, 5, 4, 6, 2, 3, 1.

Note : Hall Sequence and MC Pattern values can be updated in the BLDC Scalar control SW directly. File name : BLDC\_SCALAR\_HALL\_XMC13\configuration\blcdc\_scalar\_user\_config.h, #define • MOTOR0\_BLDC\_SCALAR\_HALL\_POSITIVE\_DIR\_SEQ and MOTOR0\_BLDC\_SCALAR\_HALL\_PAT

# Getting Started – Interface with $\mu$ C/Probe [5/6]

8. In the tab "BLDC: Speed Control", select the various widgets to control the motor.
- Start/ Stop control
  - PI tuning and monitoring



The screenshot shows the "BLDC: Speed Control" tab in a software interface. The title bar includes tabs for "BLDC: Speed Control", "BLDC: Speed Current Control", "Hall Pattern Learning", "BLDC: Voltage Control", "Oscilloscope", and "BLDC: Current Control". The main content area is titled "3 Hall sensor based BLDC Scalar Control".

At the top, there are several control fields:

- Speed PI Control:** P setting (20972), I setting (2), SCALEKPKI (15), error (18), uK (136), uK\_min (16383), and uK\_max (-16383).

The main interface is divided into several sections:

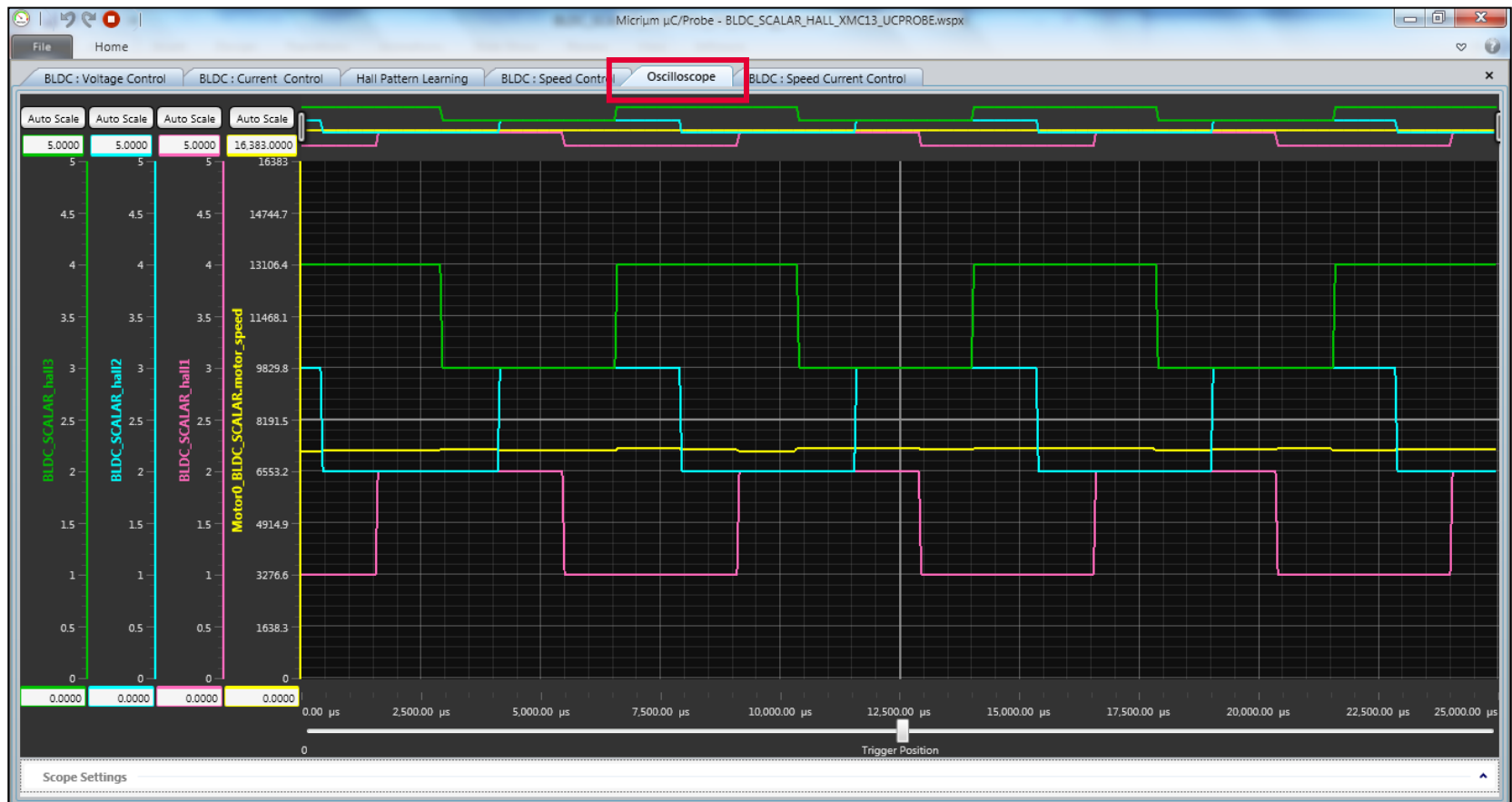
- Setting Target Speed:** Includes a speed slider (44%), a "100% Max. Speed" indicator (4530 RPM), a "Ref. Speed" (1999 RPM), and a "Real Speed" (1992 RPM). It also features "MotorStart" and "MotorStop" buttons.
- DC Link Voltage:** A green bar graph and a digital display showing 23.5 Volt.
- DC Link Current:** A digital display showing 0.008 A.
- Motor State:** A digital display showing 7, with a list of states: Stop = 0, Start-up = 1, State Identification = 2, BootStrap = 3, Hall Learning = 4, Normal Operation = 7, Error = 8, and Idle = 9.
- Error State:** A digital display showing 0, with a list of error states: Error State Bit Position, CTrap = 0, Over Current = 1, Short Circuit = 2, Wrong Hall = 3, Hall Learning Timeout = 4, Stall condition = 5, Under/ Over Voltage = 6, and Free running mode = 7/8. It also includes a "Clear Error" button.

At the bottom, there are two buttons: "Save PI and HALL pattern to Flash" and "Load Default Value". A note at the bottom right states: "Note : Hall Pattern, PI values and motor direction variables are saved in Flash".

- › Possible to save PI values, commutation table into Flash



# Getting Started – Interface with $\mu$ C/Probe [6/6]

9. Click on the "Oscilloscope" tab for monitoring motor control parameters



# General Information (1/2)

## › Where to buy kits:

Development Boards	Order Number
XMC1300 Boot Kit	 <a href="#">KIT XMC13 BOOT 001</a>
PMSM Low Voltage 15W Card	 <a href="#">KIT XMC1x AK Motor 001</a>

## General Information (2/2)

- › For latest updates, please refer to:

<http://www.infineon.com/xmc1000>

- › DAVE™ development platform:

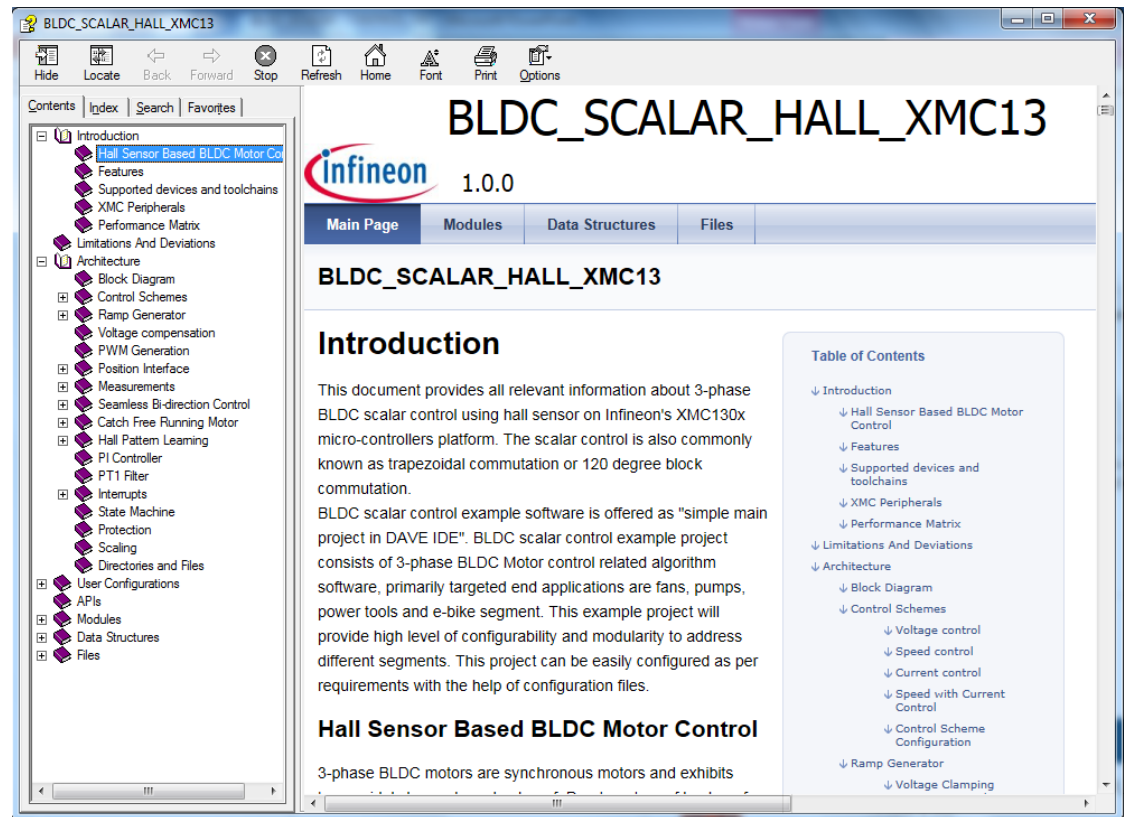
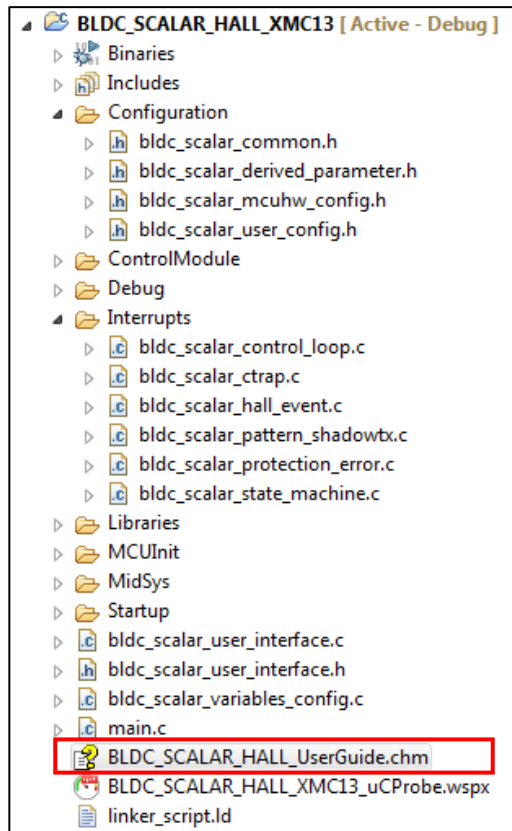
<http://www.infineon.com/DAVE>

- › 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 XMCTM )
- › 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.

