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

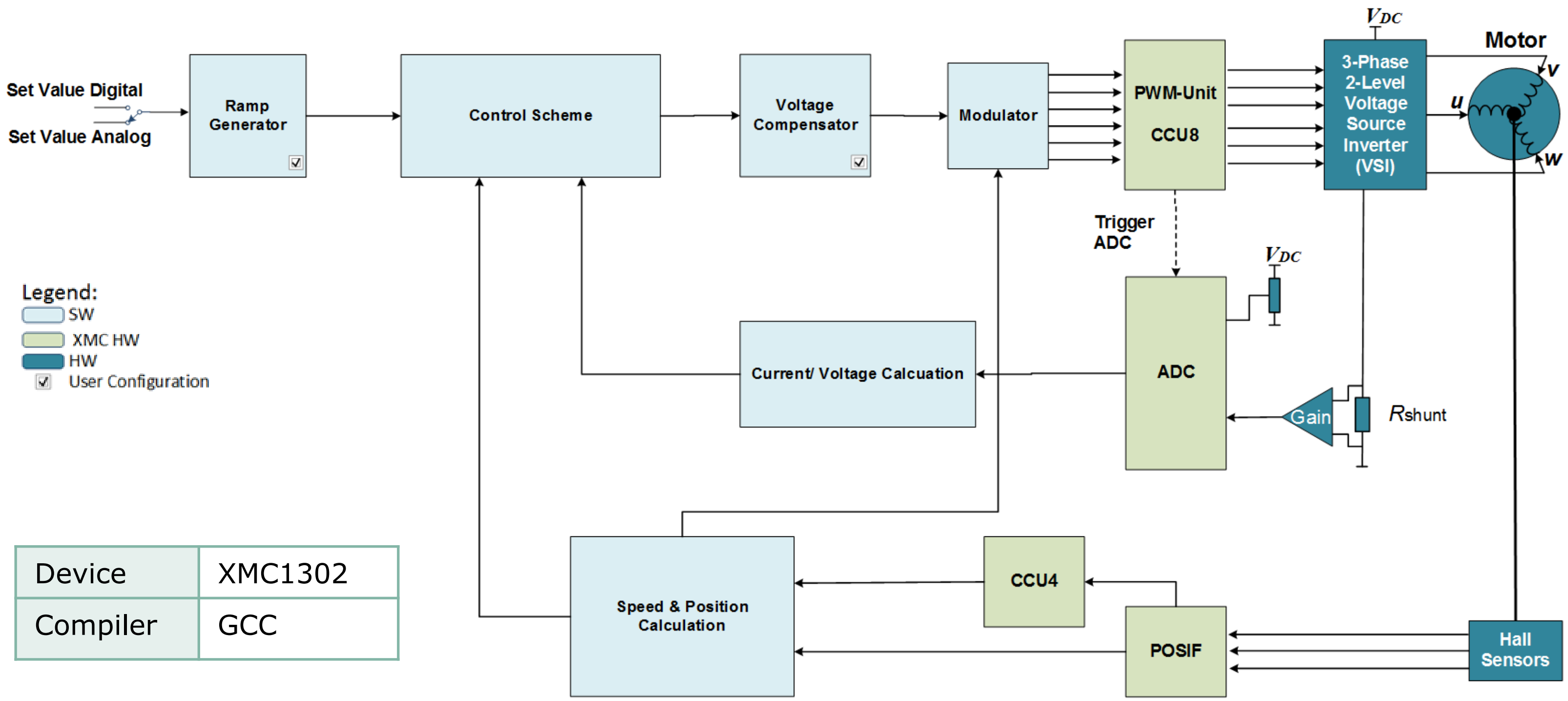
XMC^{TM} Microcontrollers
Oct 2016

Agenda

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^{TM} 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



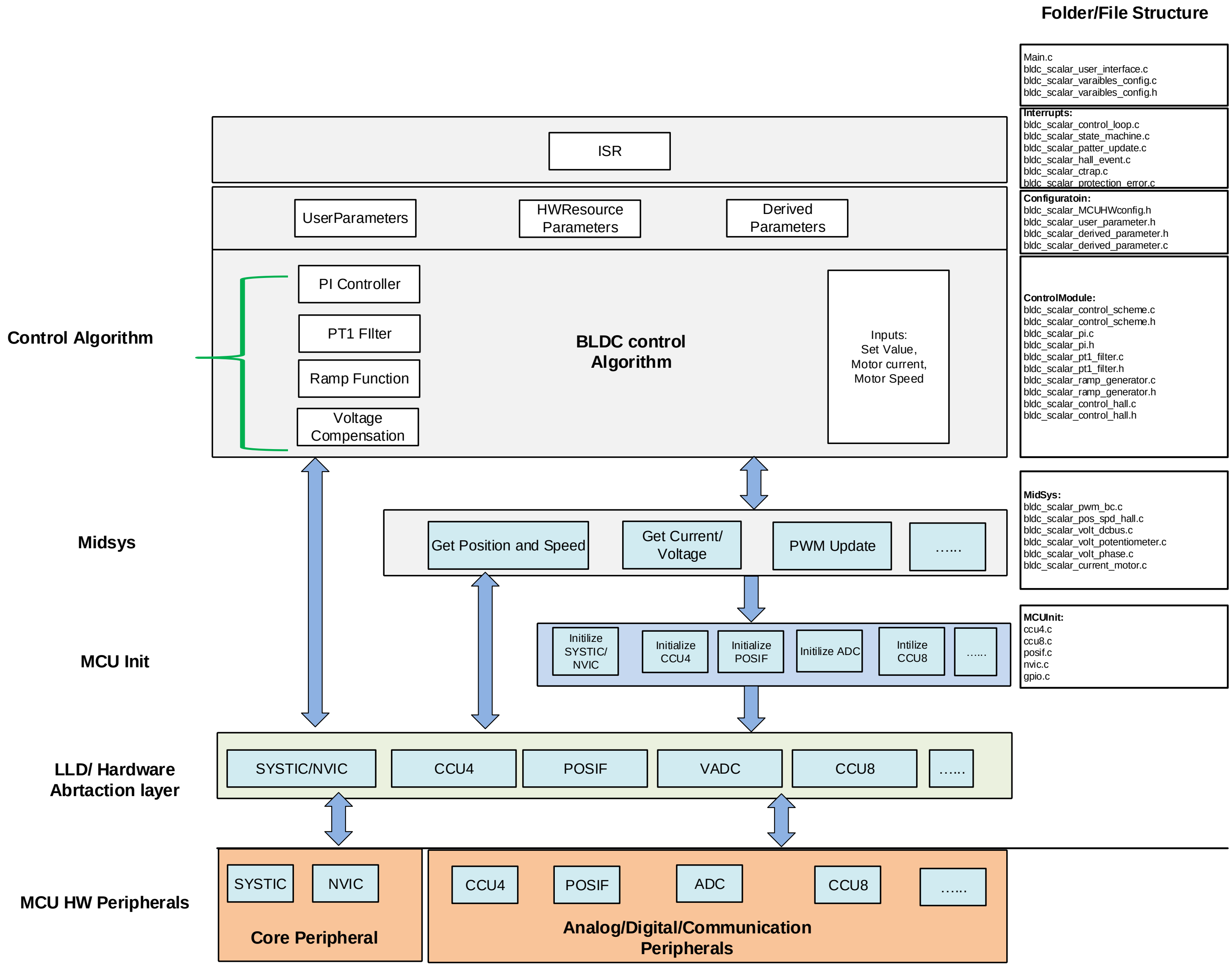
Device	XMC1302
Compiler	GCC

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

Software Overview – Key Features

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

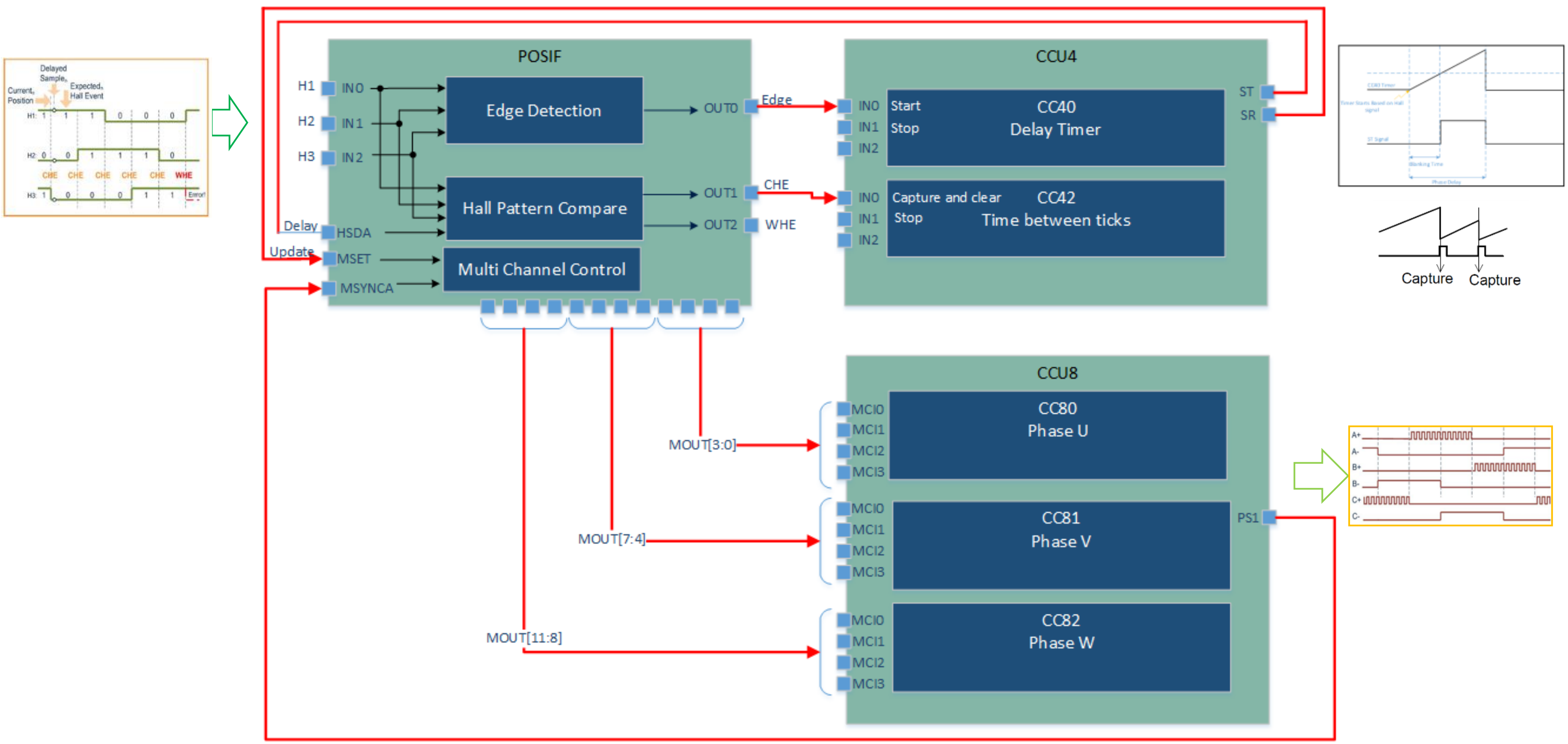
Software Overview - Files Structure



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

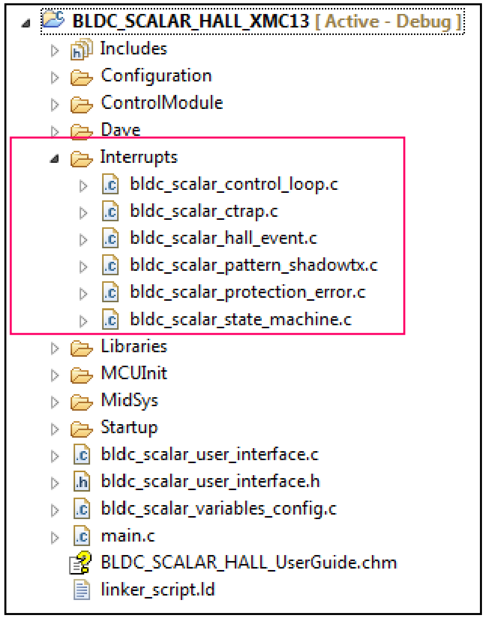


Software Overview - Interrupt Service Routines

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/(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

Example Name	BLDC_SCALAR_HALL_XMC13
Kit Description	Drive 3-phase Maxon's BLDC motor using XMC1000 motor control application kit
Part Number	KIT_XMC1X_AK_MOTOR_001

Schemes	Default Configuration in Example Software
Control Scheme	MOTOR0_BLDC_SCALAR_SPEED_CTRL
PWM Modulation	MOTOR0_BLDC_SCALAR_PWM_HIGHSIDE
PWM frequency (Hz)	20000
Speed (rpm)	2000
Ramp up/down rate	500
Protection	Over-current protection with direct DC link current measurement

› Performance Matrix

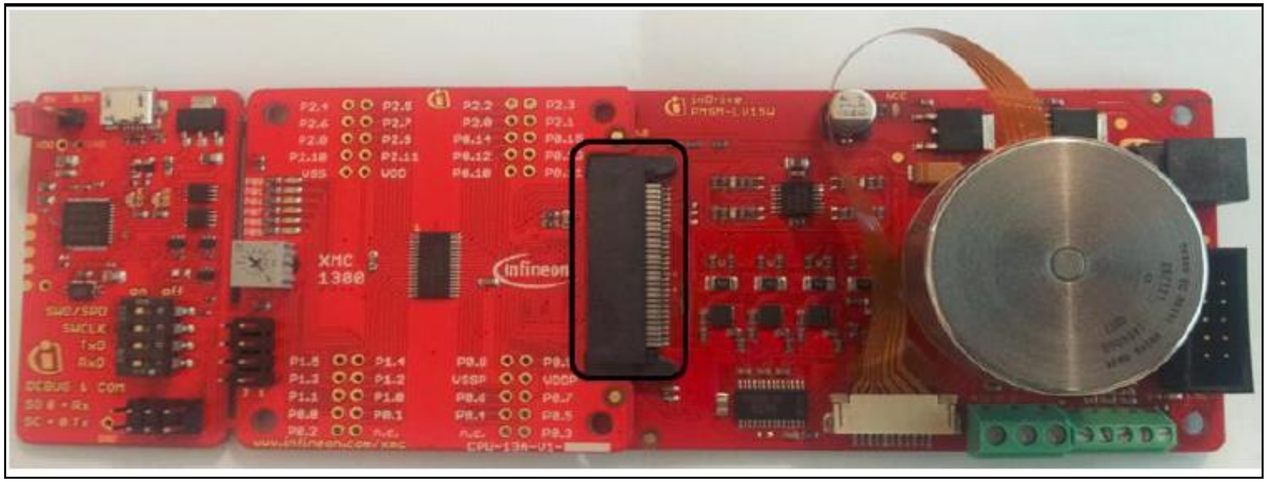
Execution Time and Code Size		
	(us)	Code Size (bytes)
Control loop ISR	6.5	8296
Pattern update ISR	14.6	
Motor state machine - Normal state	5.92	

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)

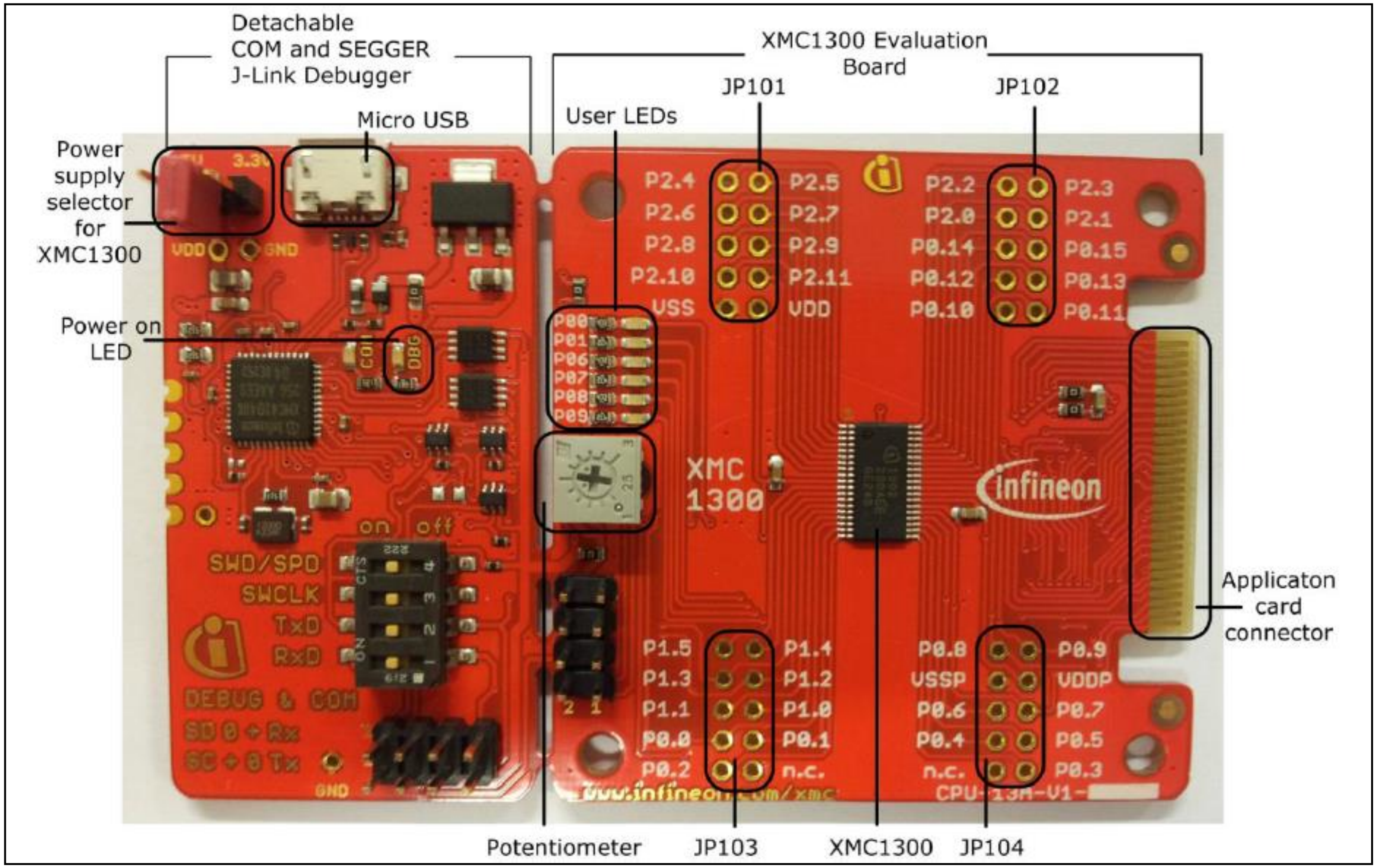


XMC1300 CPU Card

PMSM Low Voltage 15W Motor Card

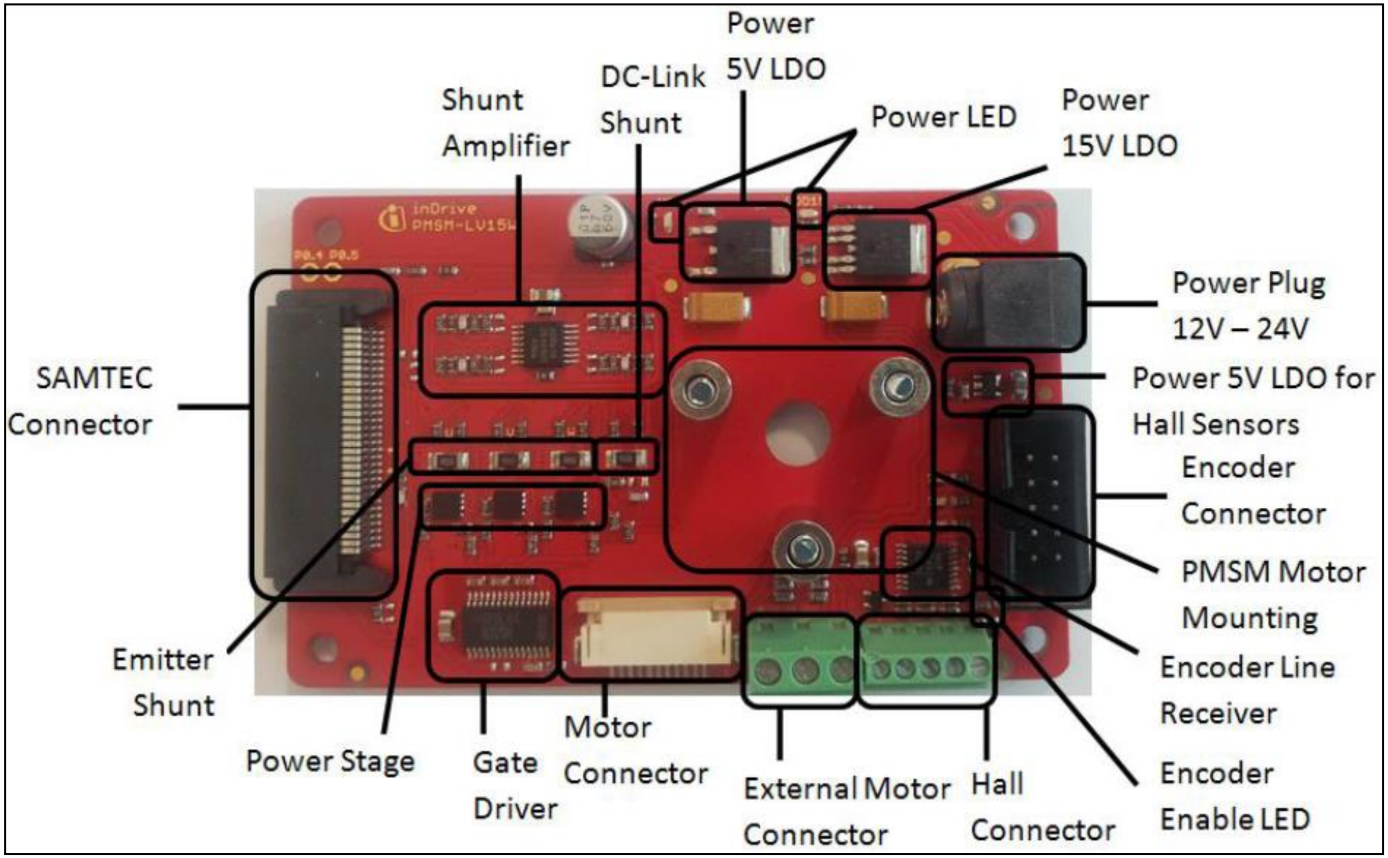
Hardware Overview – XMC1300 CPU Card

› XMC1300 CPU Card



Hardware Overview – Motor Card

› PMSM Low Voltage 15W Motor Card



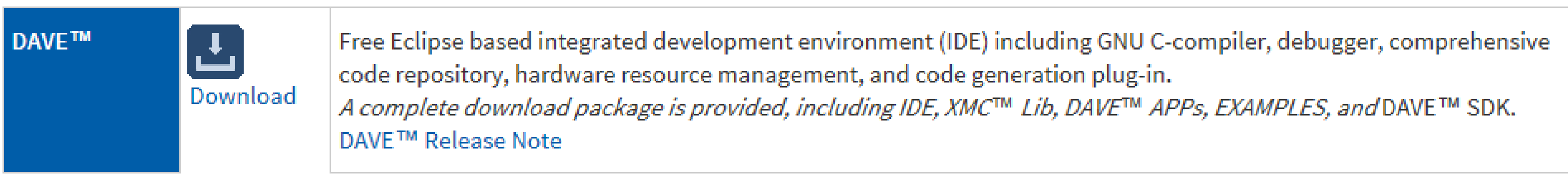
Hardware Overview – Kit Order information

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

Tools Overview

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

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

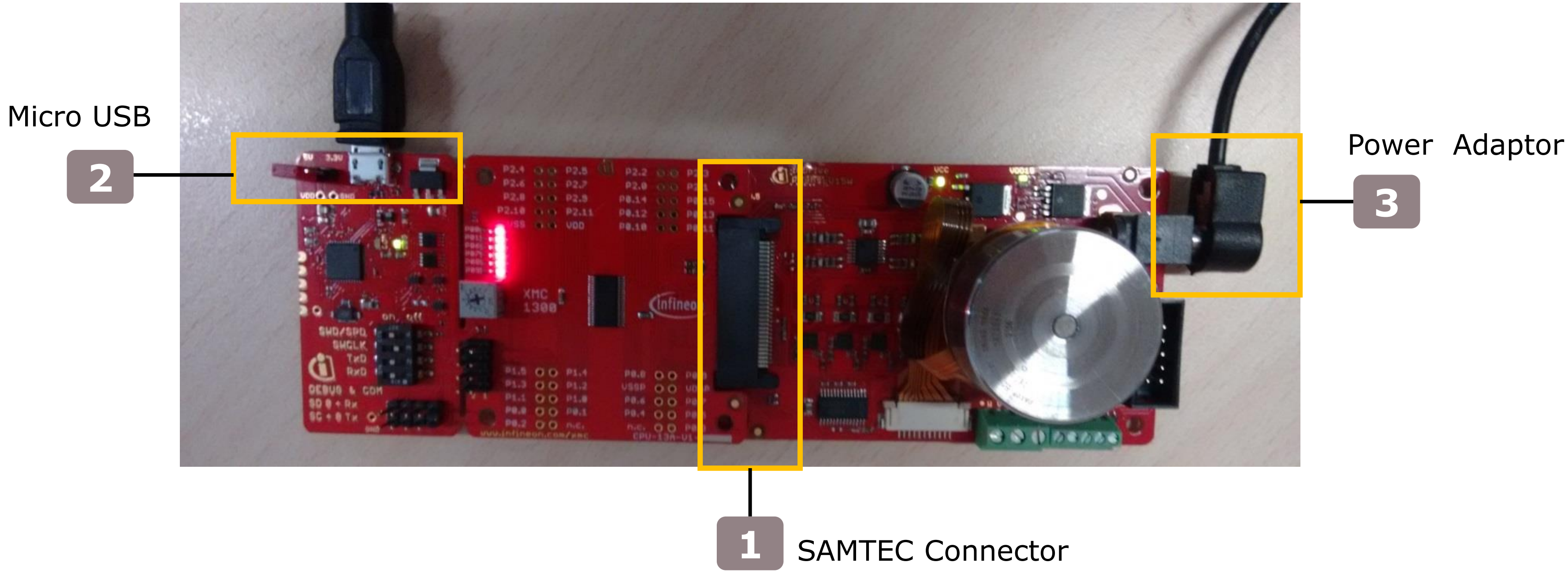


- µC/Probe™ XMC™ (v4.0.16.54 onwards) for Infineon industrial microcontrollers powered by Micrium®
  - Download from µC/Probe™ XMC™ from DAVE home page

    https://infineoncommunity.com/uC-Probe-XMC-software-download_ID712

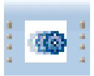
Getting Started – Connecting the Board

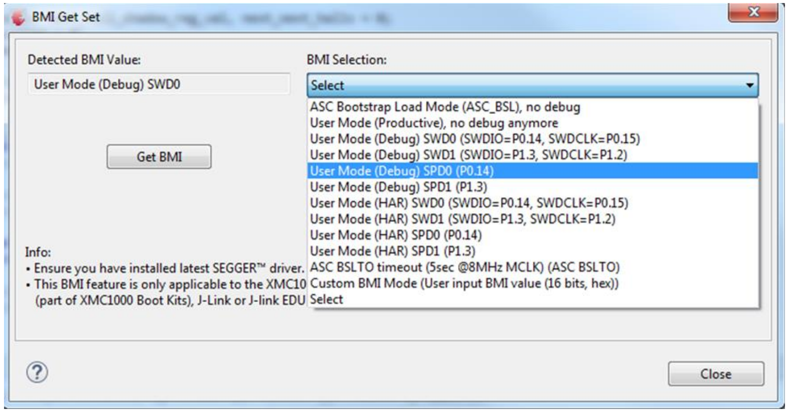
1. Connect XMC1300 CPU Card to PMSM Low Voltage 15W Motor Card using SAMTEC connector interface
2. Connect XMC1300 CPU Card to PC via Micro USB cable
3. Connect power adaptor to PMSM Low Voltage 15W Motor Card



Getting Started – Setting up the Board

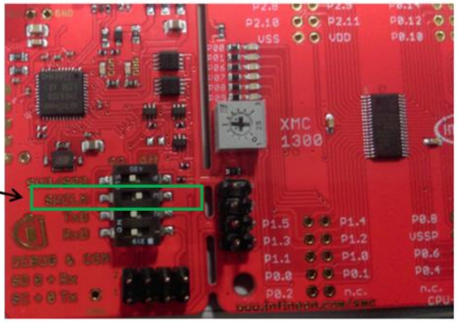
› **Note:** For this motor kit, one of the hall signal inputs are at P0.15. Therefore, to avoid conflict to the device, please ensure the following settings to the XMC1302 CPU Board.

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



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

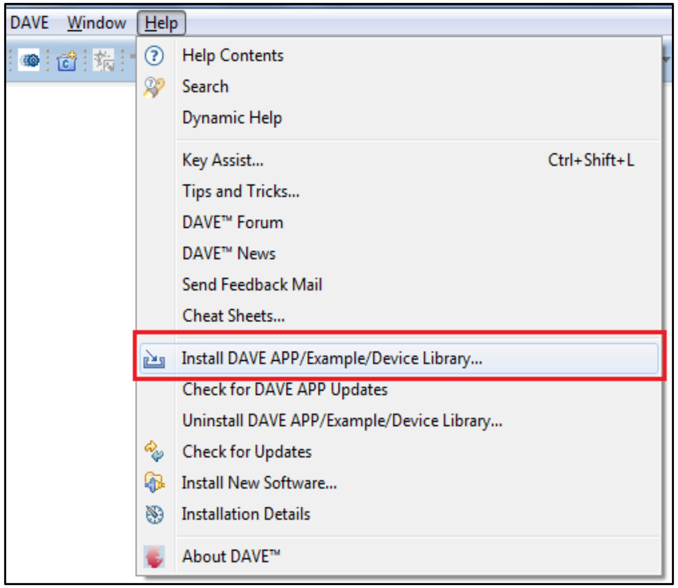
SWCLK should be in OFF position



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

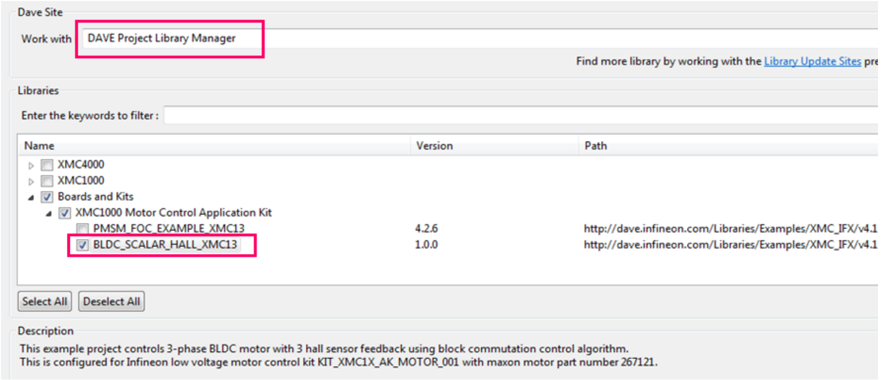
1. Open DAVE™ 

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”.

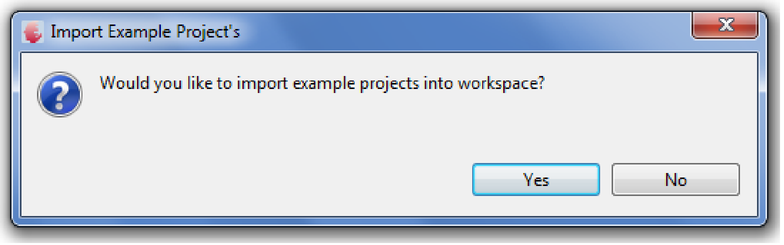


It is available at 2 locations.

[#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

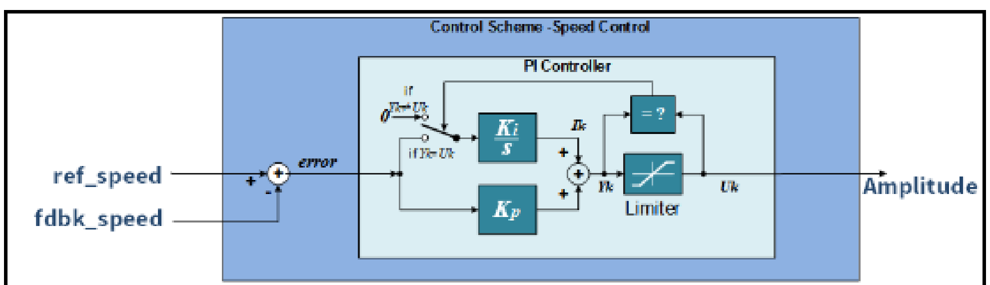
```
/*********************************************************************************************************************
 * Motor and power board selection
 ********************************************************************************************************************/
/* Board and motor selection */
/**
 * Motor control kit
 * Options - KIT_XMC1X_AK_MOTOR_001, KIT_XMC750WATT_MC_AK_V1, KIT_CUSTOM
 */
#define   MOTOR0_BLDC_SCALAR_BOARD                         (KIT_XMC1X_AK_MOTOR_001)
/**
 * BLDC motor
 * Options - MOTOR_EC_MAXON_267121, MOTOR_CUSTOM
 */
#define   MOTOR0_BLDC_SCALAR_MOTOR                         (MOTOR_EC_MAXON_267121)

/* Motor Parameters */
#if (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_EC_MAXON_267121)
#define   MOTOR0_BLDC_SCALAR_MOTOR_NO_LOAD_SPEED            (4530U)    /*!< No load speed of the motor in RPM */
#define   MOTOR0_BLDC_SCALAR_MOTOR_POLE_PAIRS               (4U)       /*!< Pole pairs */

#elif (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_CUSTOM)
#define   MOTOR0_BLDC_SCALAR_MOTOR_NO_LOAD_SPEED          (6200U)      /*!< No load speed of the motor in RPM */
#define   MOTOR0_BLDC_SCALAR_MOTOR_POLE_PAIRS             (4U)         /*!< Pole pairs */
#endif
```

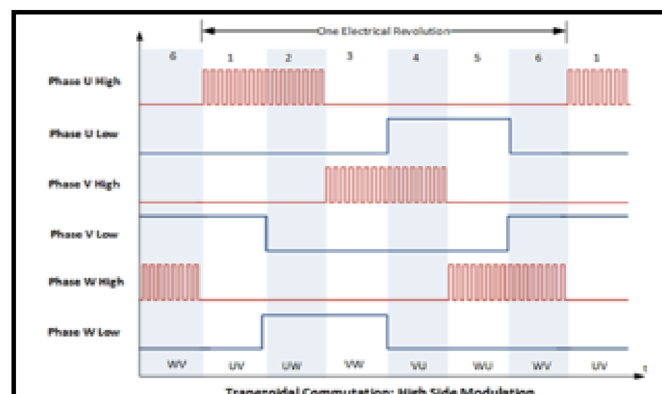
Getting Started – Configure the Project [2/8]

Folder: Configuration
File name: bldc_scalar_user_config.h

2. Select the Control Scheme and PWM Modulation Scheme

```
/*********************************************************************************************
 * Control scheme configurations
 *********************************************************************************************/
/********************  Control & PWM Modulation Scheme  ********************************/
/**
 * Control scheme selection:
 * Options - BLDC_SCALAR_VOLTAGE_CTRL, BLDC_SCALAR_SPEED_CTRL, BLDC_SCALAR_CURRENT_CTRL, BLDC_SCALAR_SPEEDCURRENT_CTRL
 */
#define   MOTOR0_BLDC_SCALAR_CTRL_SCHEME            (BLDC_SCALAR_SPEED_CTRL)
/**
 * PWM modulation scheme selection:
 * Options - BLDC_SCALAR_PWM_HIGHSIDE, BLDC_SCALAR_PWM_LOWSIDE, BLDC_SCALAR_PWM_HIGHSIDE_SYNCHRECTI
 */
#define   MOTOR0_BLDC_SCALAR_MODULATION             (BLDC_SCALAR_PWM_HIGHSIDE)


/*********************** PWM Frequency and scheduler settings ************************************/
/* CPU clock (mclk) is configured to 32MHz and CCU peripheral clock(pclk) to 64 MHz (double clock) */
/** PWM switching frequency. Range: 1000 to 100000 Hz*/
#define   MOTOR0_BLDC_SCALAR_PWM_FREQ                     (20000.0F)
/** Scheduler interrupt period. Range: 1000 to 10000 us*/
#define   MOTOR0_BLDC_SCALAR_SYSTICK_PERIOD               (1000.0F)

```





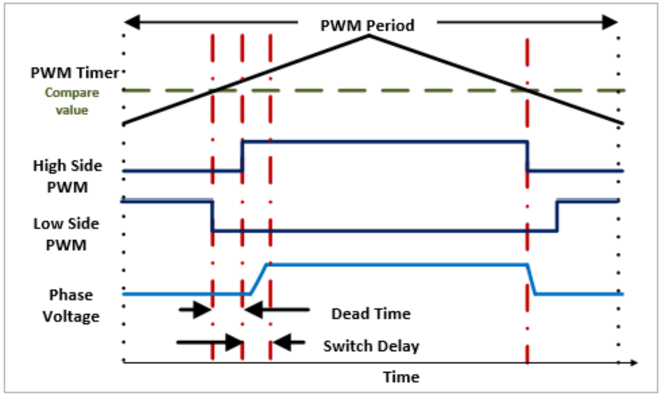
High side PWM modulation

Getting Started – Configure the Project [3/8]

3. Configure the Power Board

Folder: Configuration
File name: bldc_scalar_user_config.h

```
/***********************************************************************************************************************
 * Power board configurations
 **********************************************************************************************************************/
/************************ Power board parameters ************************************************************/
#if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
/* Power Inverter parameters */
#define    MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT          (24.0F)     /*!< DC link voltage */
#define    MOTOR0_BLDC_SCALAR_RISING_DEAD_TIME              (0.75F)     /*!< Dead time for rising edge in uSec*/
#define    MOTOR0_BLDC_SCALAR_FALLING_DEAD_TIME             (0.75F)     /*!< Dead time for falling edge in uSec*/
#define    MOTOR0_BLDC_SCALAR_SWITCH_DELAY                  (0.75F)     /*!< Switch delay in uSec*/

#define    MOTOR0_BLDC_SCALAR_HS_SWITCH_ACTIVE_LEVEL         (BLDC_SCALAR_ACTIVE_HIGH)      /*!< Active level of the high side swit
#define    MOTOR0_BLDC_SCALAR_LS_SWITCH_ACTIVE_LEVEL         (BLDC_SCALAR_ACTIVE_HIGH)      /*!< Active level of the low side switc
#define    MOTOR0_BLDC_SCALAR_INVERTER_ENABLE_CONF           (BLDC_SCALAR_INV_ACTIVE_HIGH)  /*!< Active level of inverter enable. (

/* ADC Measurement  parameters */
#define    MOTOR0_BLDC_SCALAR_VADC_REF_VOLTAGE               (5.0F)        /*!< Reference voltage of VADC conversion */
#define    MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_OFFSET       (2.5F)      /*!< Amplifier offset voltage  */
#define    MOTOR0_BLDC_SCALAR_CURRENT_RSHUNT                 (50.0F)      /*!< Current amplifier shunt resistor value in mOhms */
#define    MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_GAIN         (16.4F)       /*!< Current amplifier gain */
#define    MOTOR0_BLDC_SCALAR_VOLTAGE_DIVIDER_RATIO          (9.79F)     /*!< Voltage divider ratio in % for DC link voltage measur
/* end of #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
```



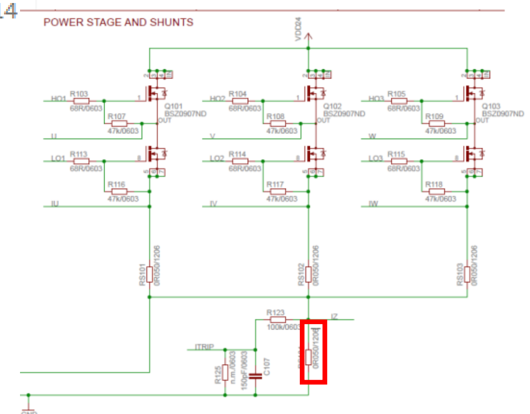
Getting Started – Configure the Project [4/8]

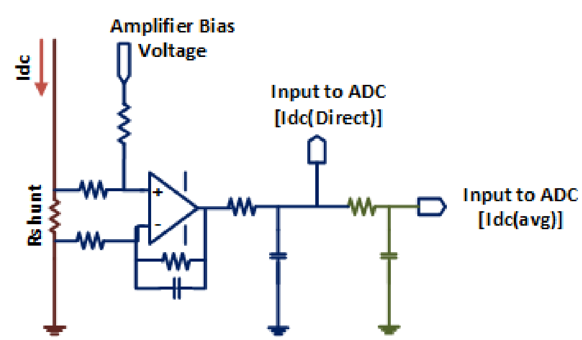
4. Configure the Power Board

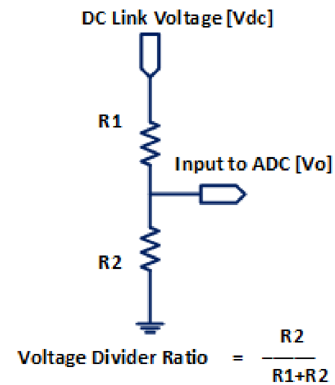
Folder: Configuration

File name: bldc_scalar_user_config.h

```c
/**********************************************************************************************
 * Power board configurations
 **********************************************************************************************/
/*********************** Power board parameters ***************************************************/
#if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
/* Power Inverter parameters */
#define   MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT        (24.0F)     /*!< DC link voltage */
#define   MOTOR0_BLDC_SCALAR_RISING_DEAD_TIME            (0.75F)     /*!< Dead time for rising edge in uSec*/
#define   MOTOR0_BLDC_SCALAR_FALLING_DEAD_TIME           (0.75F)     /*!< Dead time for falling edge in uSec*/
#define   MOTOR0_BLDC_SCALAR_SWITCH_DELAY                (0.75F)     /*!< Switch delay in uSec*/

#define   MOTOR0_BLDC_SCALAR_HS_SWITCH_ACTIVE_LEVEL       (BLDC_SCALAR_ACTIVE_HIGH)      /*!< Active level of the high side swit
#define   MOTOR0_BLDC_SCALAR_LS_SWITCH_ACTIVE_LEVEL       (BLDC_SCALAR_ACTIVE_HIGH)      /*!< Active level of the low side switc
#define   MOTOR0_BLDC_SCALAR_INVERTER_ENABLE_CONF         (BLDC_SCALAR_INV_ACTIVE_HIGH)  /*!< Active level of inverter enable. (

/* ADC Measurement  parameters */
#define   MOTOR0_BLDC_SCALAR_VADC_REF_VOLTAGE             (5.0F)        /*!< Reference voltage of VADC conversion */
#define   MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_OFFSET     (2.5F)      /*!< Amplifier offset voltage  */
#define   MOTOR0_BLDC_SCALAR_CURRENT_RSHUNT               (50.0F)      /*!< Current amplifier shunt resistor value in mOhms */
#define   MOTOR0_BLDC_SCALAR_CURRENT_AMPLIFIER_GAIN       (16.4F)       /*!< Current amplifier gain */
#define   MOTOR0_BLDC_SCALAR_VOLTAGE_DIVIDER_RATIO        (9.79F)     /*!< Voltage divider ratio in % for DC link voltage measur
/* end of #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
```







Getting Started – Configure the Project [5/8]

Folder: Configuration

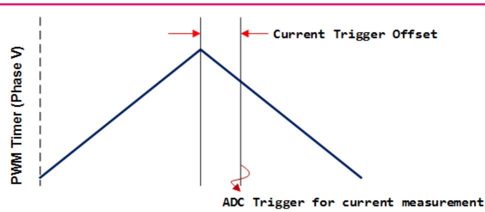
File name: bldc_scalar_user_config.h

5. Configure the Hall Pattern

```c
/*****************************************************************************************
 * Hall pattern and phase excitation pattern
 ****************************************************************************************/
#if (MOTOR0_BLDC_SCALAR_FEEDBACK == BLDC_SCALAR_3HALL)

#define MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ      (BLDC_SCALAR_HALL_SEQ_1)   /*!< Select hall sequence for positive direction. (

#if (MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ == BLDC_SCALAR_HALL_SEQ_1)
/*
 * Standard hall pattern for positive direction. Do NOT change this hall pattern values.
 * Update the phase excitation pattern corresponding to the hall pattern
 */
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_A            (1U)       /*!< (MSB)H3 H2 H1 (LSB)*/
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_B            (3U)       /*!< (MSB)H3 H2 H1 (LSB)*/
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_C            (2U)       /*!< (MSB)H3 H2 H1 (LSB)*/
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_D            (6U)       /*!< (MSB)H3 H2 H1 (LSB)*/
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_E            (4U)       /*!< (MSB)H3 H2 H1 (LSB)*/
#define    MOTOR0_BLDC_SCALAR_HALL_PAT_F            (5U)       /*!< (MSB)H3 H2 H1 (LSB)*/

#else
 * Standard hall pattern for positive direction. Do NOT change this hall pattern values.
#endif

/*************************************************************************/
/* Phase W, V, U excitation pattern */
#if (MOTOR0_BLDC_SCALAR_MOTOR == MOTOR_EC_MAXON_267121)
#define    MOTOR0_BLDC_SCALAR_MC_PAT_A              (WH_VL_UOFF)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_A */
#define    MOTOR0_BLDC_SCALAR_MC_PAT_B              (WOFF_VL_UH)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_B */
#define    MOTOR0_BLDC_SCALAR_MC_PAT_C              (WL_VOFF_UH)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_C */
#define    MOTOR0_BLDC_SCALAR_MC_PAT_D              (WL_VH_UOFF)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_D */
#define    MOTOR0_BLDC_SCALAR_MC_PAT_E              (WOFF_VH_UL)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_E */
#define    MOTOR0_BLDC_SCALAR_MC_PAT_F              (WH_VOFF_UL)  /*!< Phase pattern corresponding to MOTOR0_BLDC_SCALAR_HALL_PAT_F */
```

6. Configure the ADC Trigger

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



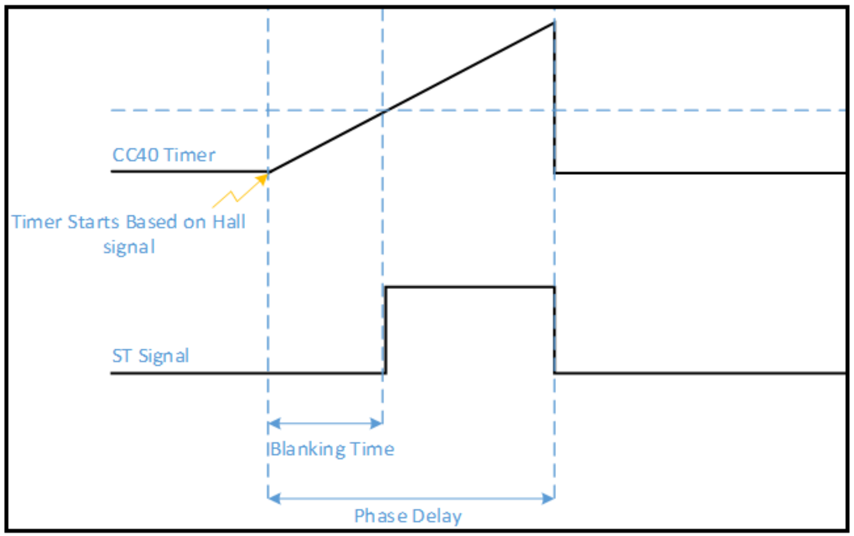
Getting Started – Configure the Project [6/8]

Folder: Configuration

File name: bldc_scalar_user_config.h

7. Configure the Hall sensor feedback

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



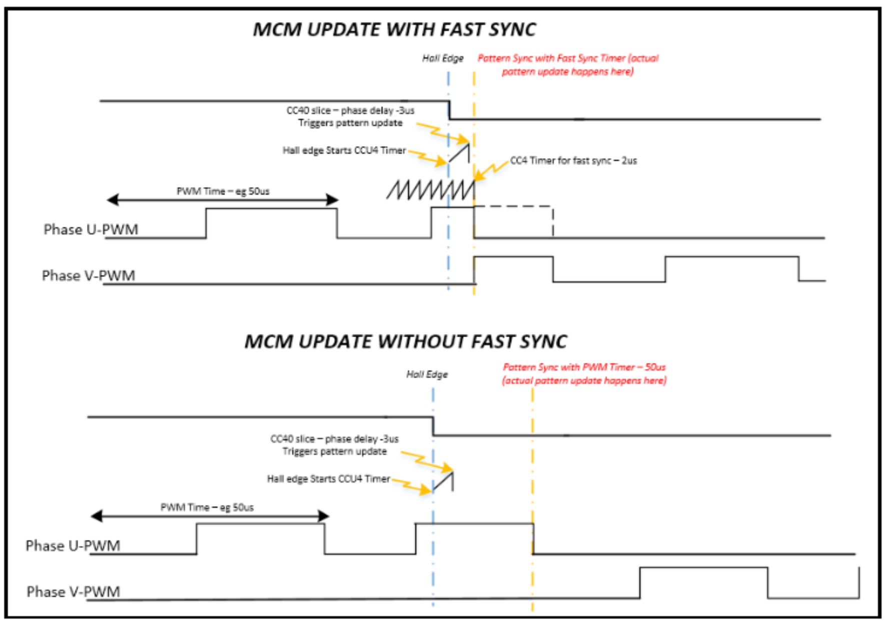
CCU4 timer for hall edge detection

Getting Started – Configure the Project [7/8]

Folder: Configuration
File name: bldc_scalar_user_config.h

8. Configure the MCM Transfer

```
#define     MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNCH_CCU4         (1U)      /*!< Enable (1)/disable (0)
#if (MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNCH_CCU4 == 1U)
#define     MOTOR0_BLDC_SCALAR_MCM_SYNCTRANSFER_TIME         (2.0F)      /*!< Multi-channel pattern
#endif /* end of #if (MOTOR0_BLDC_SCALAR_ENABLE_FAST_SYNCH_CCU4 == 1U) */
```



MCM Pattern update with fast sync and without fast sync

Getting Started – Configure the Project [8/8]

Folder: Configuration
File name: bldc_scalar_user_config.h

9. Enable the Hall pattern learning

```
#define   MOTOR0_BLDC_SCALAR_ENABLE_HALL_LEARNING       (1U)     /*!< Enable/disable hall pattern learning */
#if (MOTOR0_BLDC_SCALAR_ENABLE_HALL_LEARNING == 1U)
#define   MOTOR0_BLDC_SCALAR_OPEN_LOOP_VOLTAGE          (5.0F)     /*!< Open loop voltage to be applied in % w
#define   MOTOR0_BLDC_SCALAR_OPEN_LOOP_SPEED            (2.0F)     /*!< Speed to be applied in % with respect
#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]

Folder: -
File name: main.c

1. Initialize the uCProbe before starting motor

```
int main(void)
{
  /* Initialization */
  Motor0_BLDC_SCALAR_Init();

  Motor0_BLDC_SCALAR_Flash_Var_Init();

#if (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
  Motor0_BLDC_SCALAR_uCProbe_Init();
#endif


//  /* Start the motor */
 Motor0_BLDC_SCALAR_MotorStart();


  /* Placeholder for user application code. The while loop below can be r
  while (1U)
  {

  }
}
```

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

```
void SysTick_Handler(void)
{
  /* Call motor control state machine */
  Motor0_BLDC_SCALAR_MSM();
}

/**
 * @}
 */
/**
 * @}
 */

RAM_ATTRIBUTE void Motor0_BLDC_SCALAR_MSM(void)
{
  switch (Motor0_BLDC_SCALAR.msm_state)
  {
    case BLDC_SCALAR_MSM_NORMAL_OPERATION:
      Motor0_BLDC_SCALAR_MSM_NORMAL_OPERATION_Func();
      break;

#if (MOTOR0_BLDC_SCALAR_ENABLE_BIDIRECTIONAL_CTRL == 0U)
#if (MOTOR0_BLDC_SCALAR_ENABLE_CATCH_FREE == 1U)
#if (MOTOR0_BLDC_SCALAR_ENABLE_BOOTSTRAP == 1U)

    case BLDC_SCALAR_MSM_ERROR:
      Motor0_BLDC_SCALAR_MSM_ERROR_Func();
      break;

    case BLDC_SCALAR_MSM_START:
      Motor0_BLDC_SCALAR_MSM_START_Func();
      break;
```

```
#if (MOTOR0_BLDC_SCALAR_HALL_LEARNING == 1U)
    case BLDC_SCALAR_MSM_HALL_LEARNING:
      Motor0_BLDC_SCALAR_MSM_HALL_LEARNING_Func();
      break;
#endif

    case BLDC_SCALAR_MSM_STOP:
      Motor0_BLDC_SCALAR_MotorStop();
      break;

    default:
      break;

  }

  if (Motor0_BLDC_SCALAR.error_status != 0U)
  {
    Motor0_BLDC_SCALAR.msm_state = BLDC_SCALAR_MSM_ERROR;
  }

  #if (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
  Motor0_BLDC_SCALAR_uCProbe_Scheduler();
  #endif
}
```

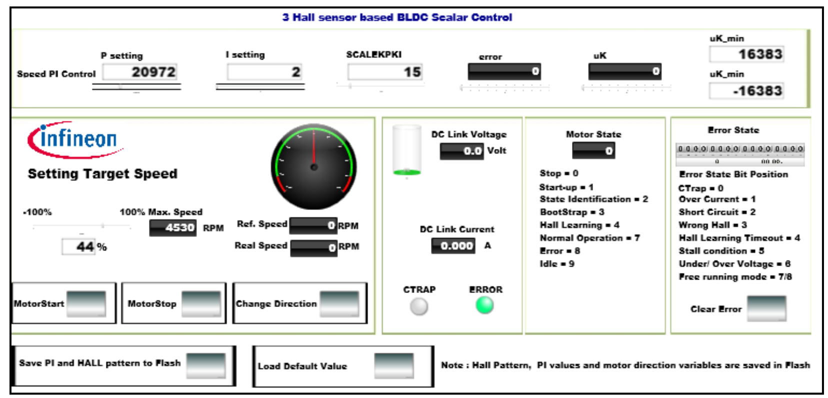
Getting Started – Add code support for uCProbe [3/3]

Folder: uCProbe
File name: ucProbe.c

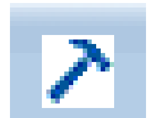
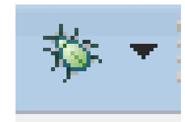
3. Add uCProbe scheduler in motor state machine

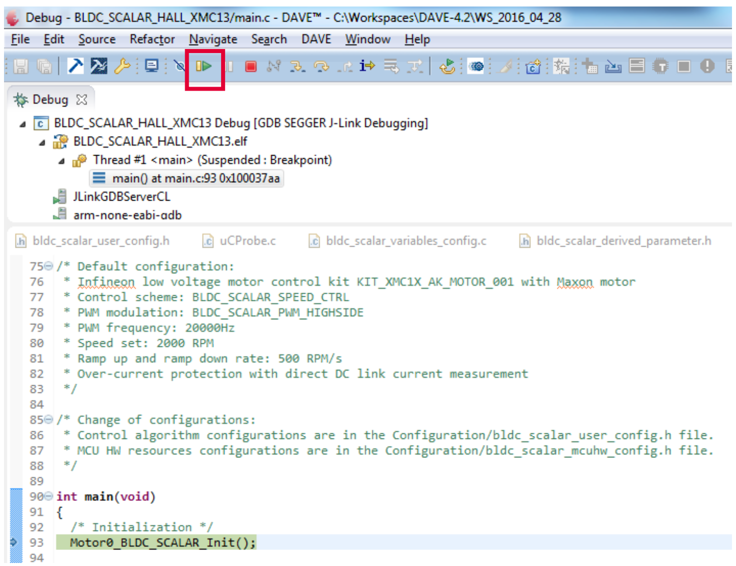
```c
#if ((MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE==1))
/*UCproBE scheduler function to handle ucprobe comments from UI */
void Motor0_BLDC_SCALAR_uCProbe_Scheduler(void)
{
  switch(Motor0_BLDC_SCALAR_ucprobe.control_word)
  {
    case 1:  /* Start the motor */
      Motor0_BLDC_SCALAR_ucprobe.control_word=0;
      Motor0_BLDC_SCALAR_MotorStart();
      break;

    case 2: /*Stop the motor*/
      Motor0_BLDC_SCALAR_ucprobe.control_word=0;
      Motor0_BLDC_SCALAR_MotorStop();
      break;

    case 3: /*Clear Error state*/
      Motor0_BLDC_SCALAR_ucprobe.control_word=0;
      Motor0_BLDC_SCALAR_ClearErrorState();
      break;

    case 4:  /*Clear flash and load defualt value into flash*/
      Motor0_BLDC_SCALAR_ucprobe.control_word=0;
      Motor0_BLDC_SCALAR_ucprobe.user_config[0] =0;
      Motor0_BLDC_SCALAR_Write_Default_value();
      Motor0_BLDC_SCALAR_uCProbe_Write_Flash();
      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



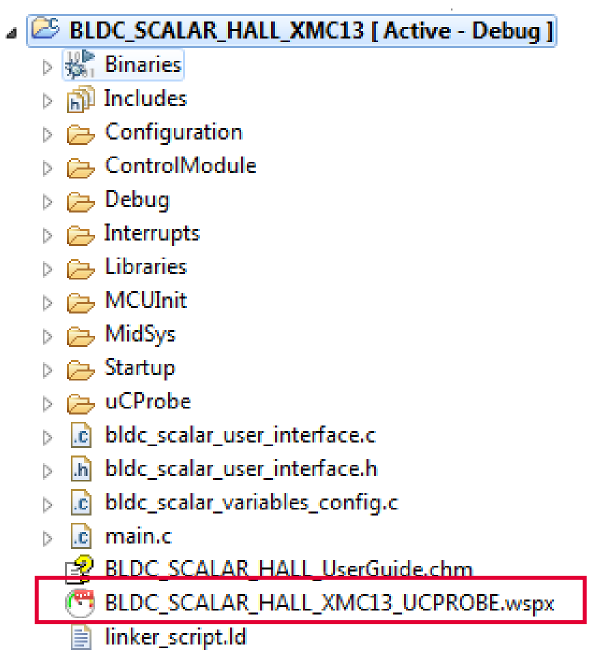
Observation:

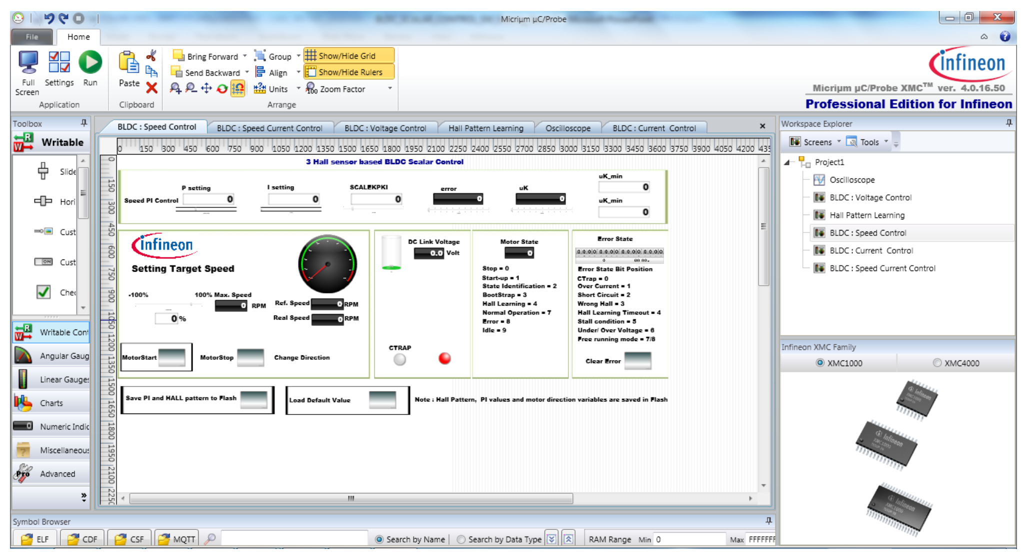
› Motor should ramp to 2000RPM with ramp rate of 500RPM/s.

Getting Started – Interface with µC/Probe [1/6]

› Update of the motor and monitoring motor parameters can be executed using µC/Probe™ XMC™

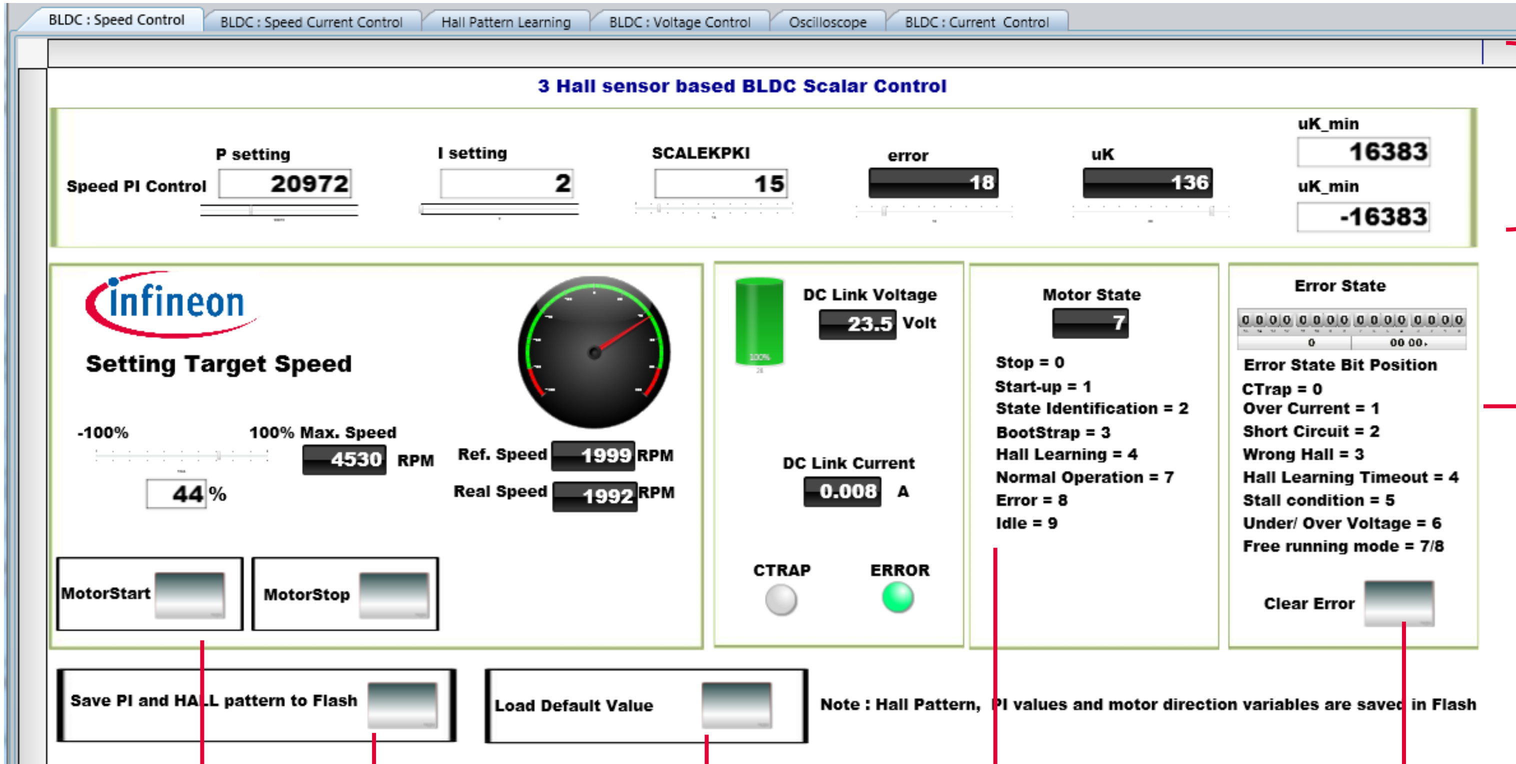
1. In "BLDC_SCALAR_HALL_XMC13" example project , open µC/Probe™ XMC™ project file.





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 µC/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

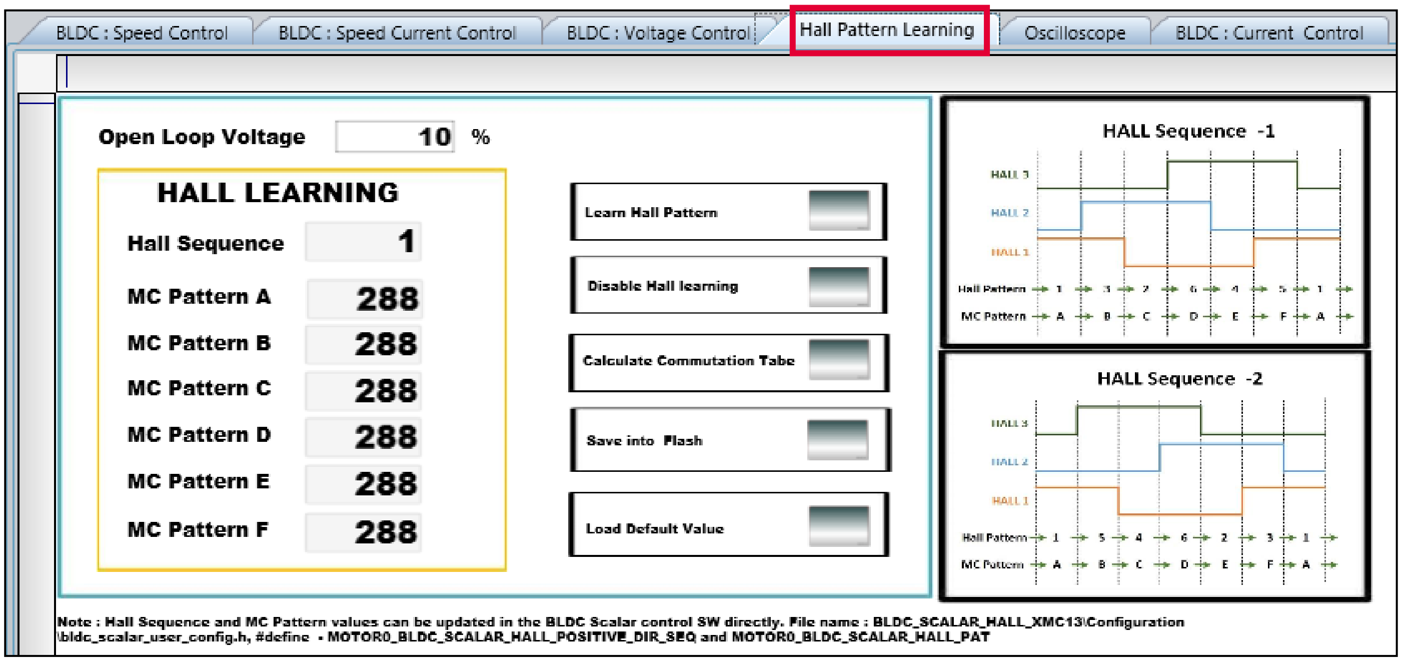
In case of Error condition, can clear the error flag in the SW by click this button

Getting Started – Interface with µC/Probe [3/6]

2. Click the 'Run' button

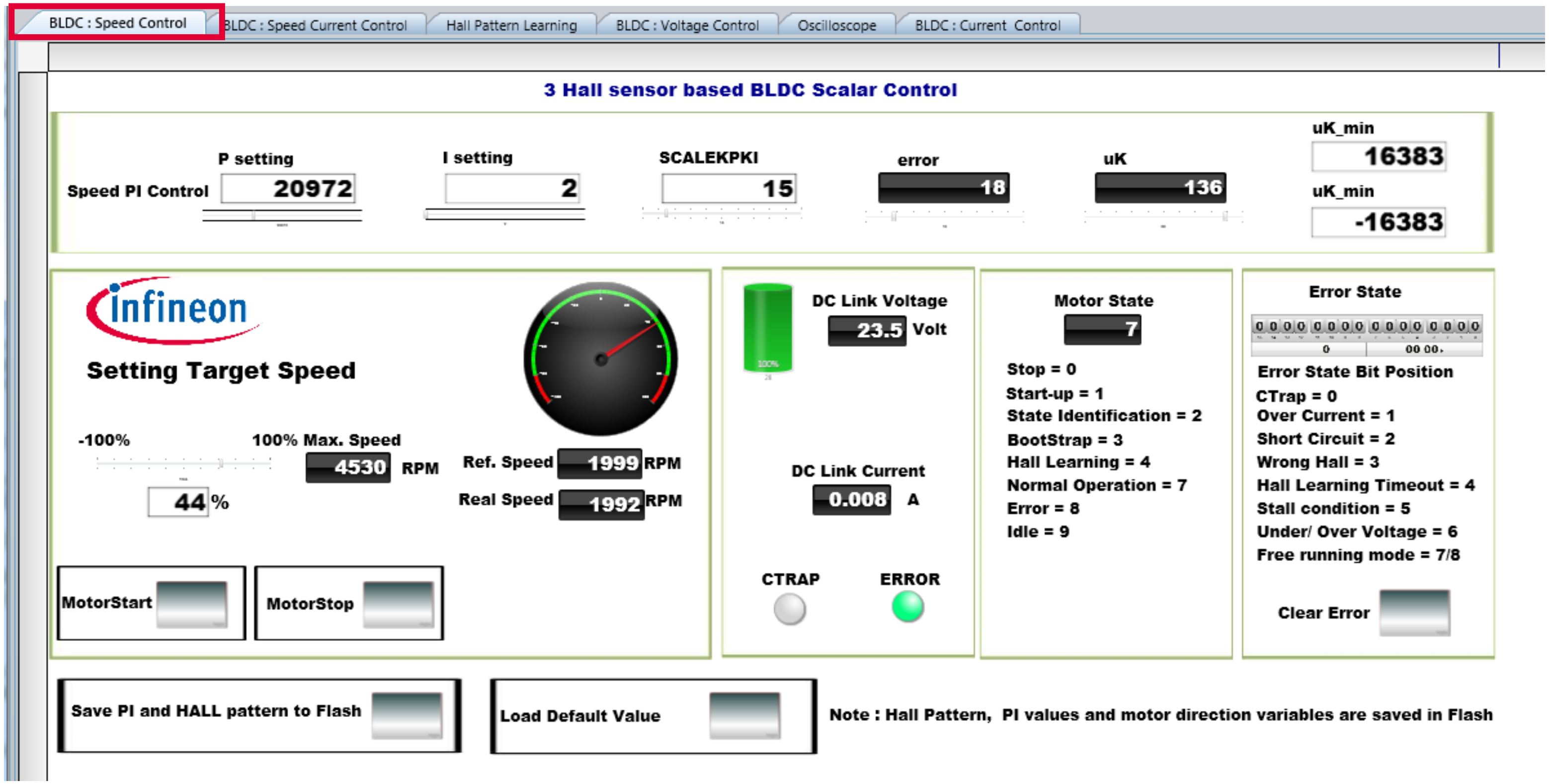
Getting Started – Interface with µC/Probe [4/6]

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.



Getting Started – Interface with µC/Probe [5/6]

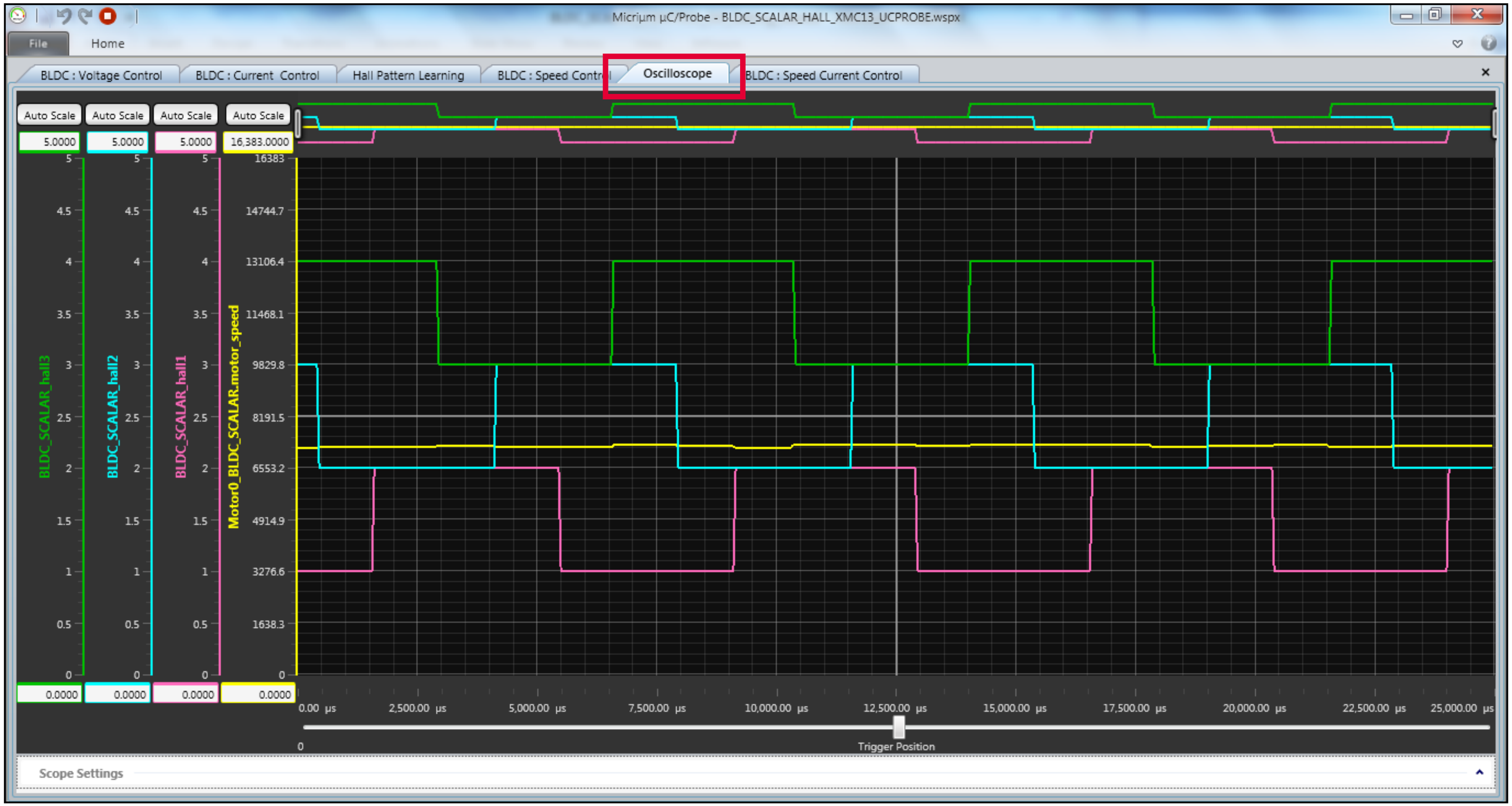
8. In the tab "BLDC: Speed Control", select the various widgets to control the motor.
   - Start/ Stop control
   - PI tuning and monitoring



› Possible to save PI values, commutation table into Flash

Getting Started – Interface with µC/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		KIT_XMC13_BOOT_001
PMSM Low Voltage 15W Card		KIT_XMC1x_AK_Motor_001

General Information (2/2)

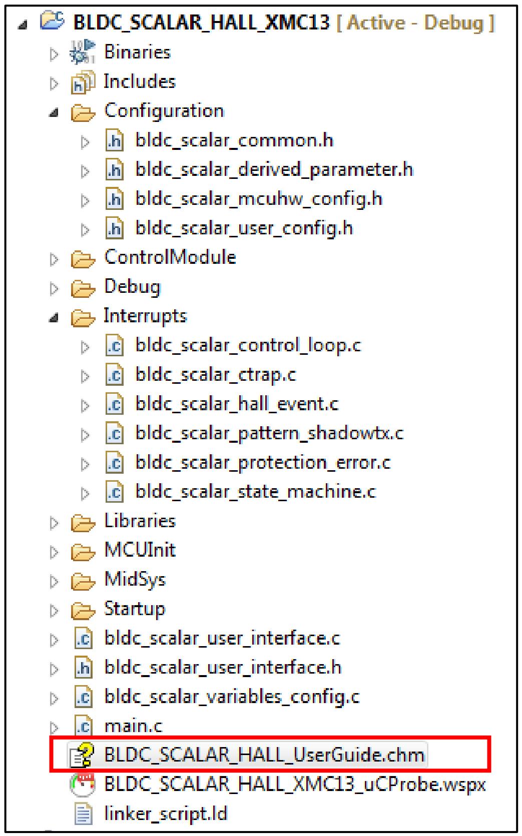
- For latest updates, please refer to:
  http://www.infineon.com/xmc1000

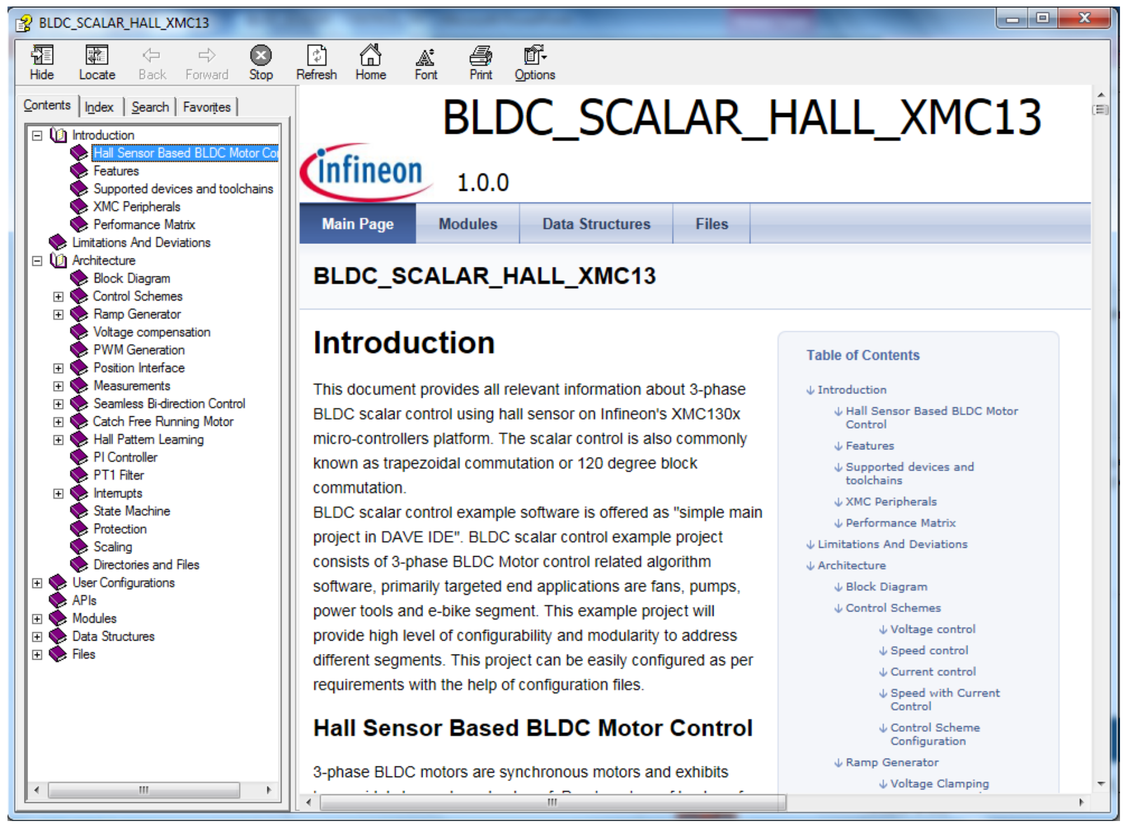
- DAVE™ development platform:
  http://www.infineon.com/DAVE

- 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.