XMC13 Sensorless BLDC Scalar Control Software **Getting Started**

XMC™ Microcontrollers
Oct 2016

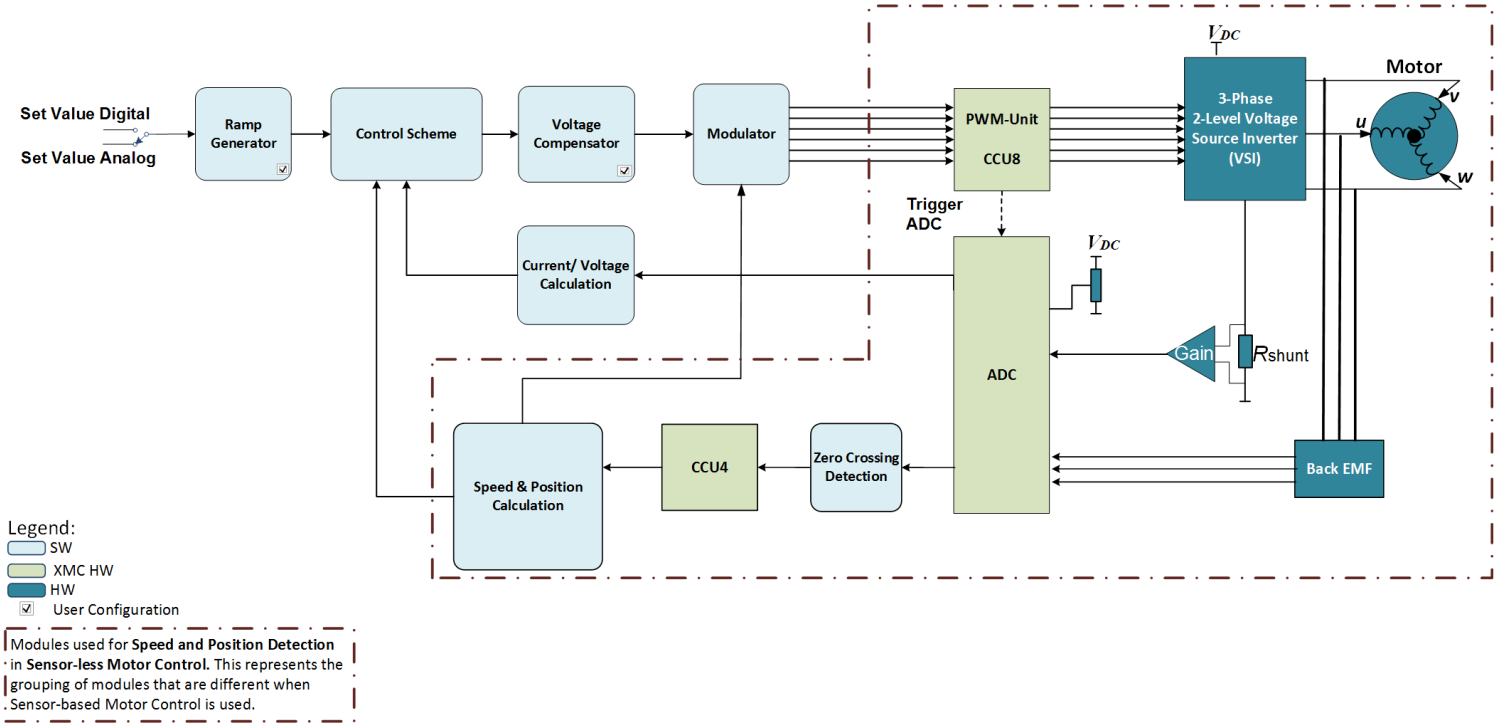
Agenda

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 Sensorless BLDC scalar control example software on Infineon's XMC1300 series micro-controllers platform
- Sensorless BLDC scalar control example software is offered as "simple main project in DAVE™ IDE"
- Sensorless BLDC scalar control example project consists of sensorless 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

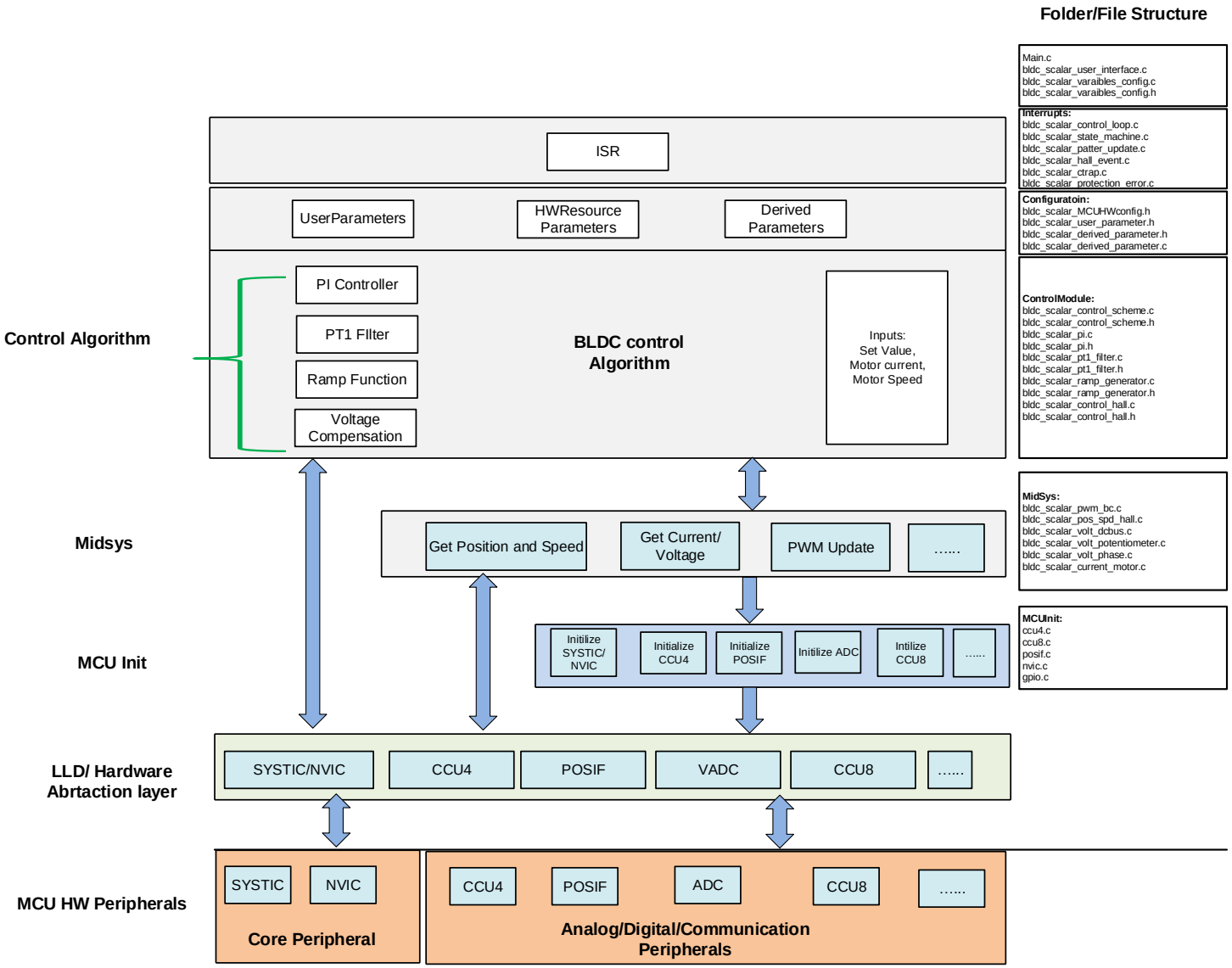


Software Blocks	Supported Options
Control Scheme	Open loop voltage control, speed control, current control and speed inner current control
PWM Modulation (Modulator)	High side modulation, low side modulation, high side with synchronous rectification
Current/Voltage Measurement	Direct DC link and average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

Software Overview – Key Features

Supported Features	Description
Bi-directional control	Reverse the motor direction when change direction variable is set
On fly start-up	Catch spinning motor at start-up without stop
Back-emf zero crossing detection	Back-emf zero crossing detection using ADC boundary checking
Motor Start-up from standstill	Support inductive position detection or alignment for smooth start
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

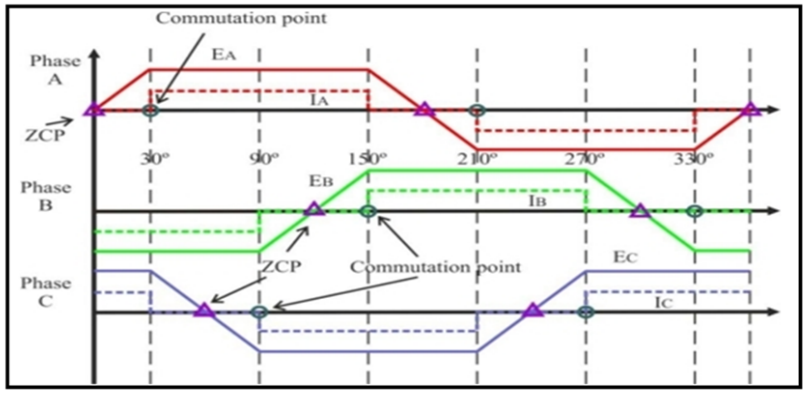
Software Overview - Files Structure



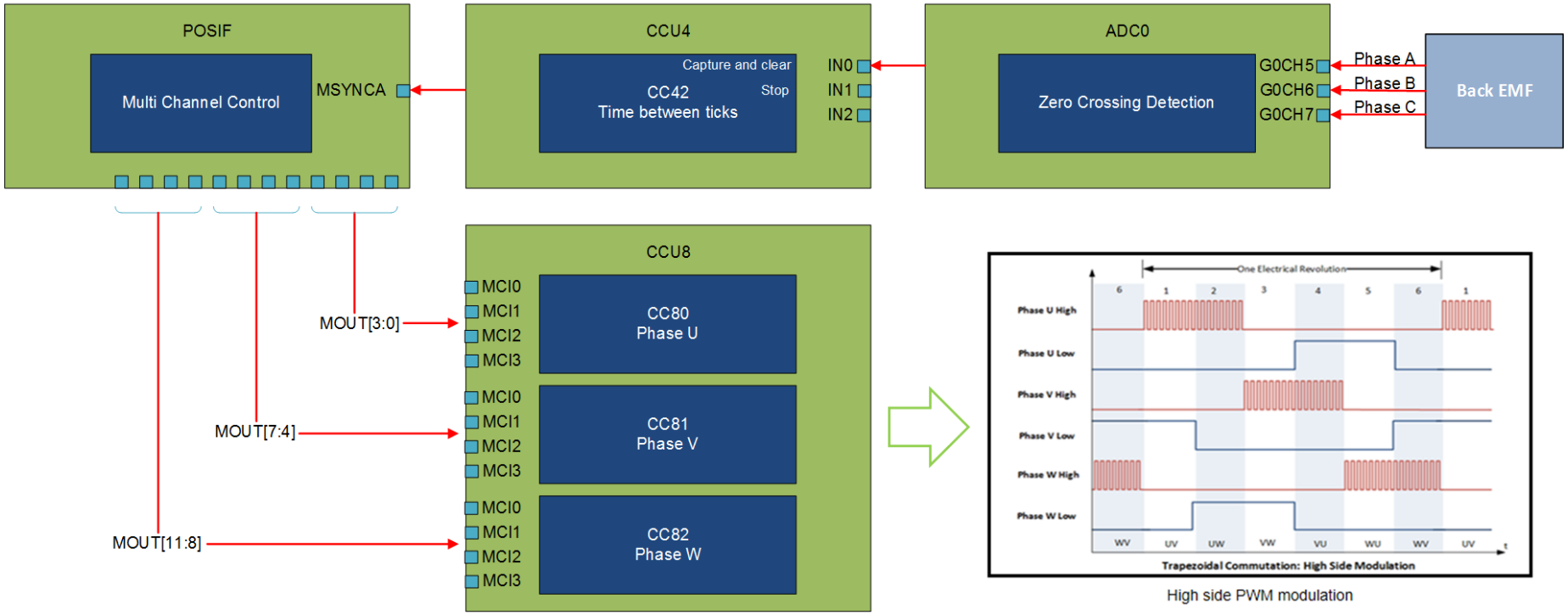
Software Overview - XMC Peripheral usage

No	Resource	Resource usage	Purpose
1	CCU40 _CC40	Always	Commutation timer (Pre-scalar value calculated in zero crossing event)
2	CCU40 _CC41	Fast Sync is disabled	Multi-channel Pattern synchronization
3	CCU40 _CC42	Always	Used for motor speed capture
4	POSIF0	Always	MCM configuration
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



BLDC Commutation in Sensorless mode



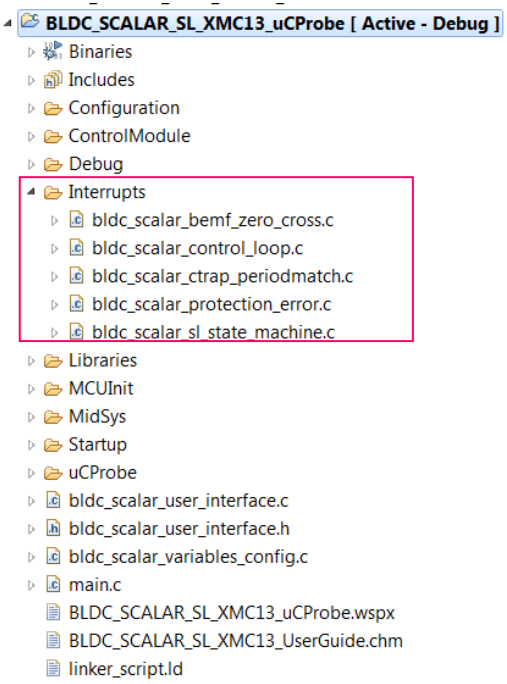
High side PWM modulation

Software Overview - Interrupt Service Routines

Folder: Interrupts

File name: bldc_scalar_state_machine.c

Peripheral	Interrupt Subroutines (ISR)	NVIC node	Interval	Priority
VADC	Channel event (current + voltage outside boundary)	15	Asynchronous	0
CCU8	One match event (phase U)	25	1/ PWM frequency	2
CCU8	CTRAP (phase U)	26	Asynchronous	0
SYSTIMER	Systick Scheduler	-1	1 mSec (configurable)	3
VADC	Zero-crossing	16	Variable (motor speed dependent)	1



Software Overview – Example Configuration

Example Name	BLDC_SCALAR_SL_XMC13_uCProbe
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	BLDC_SCALAR_SPEED_CTRL
PWM Modulation	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

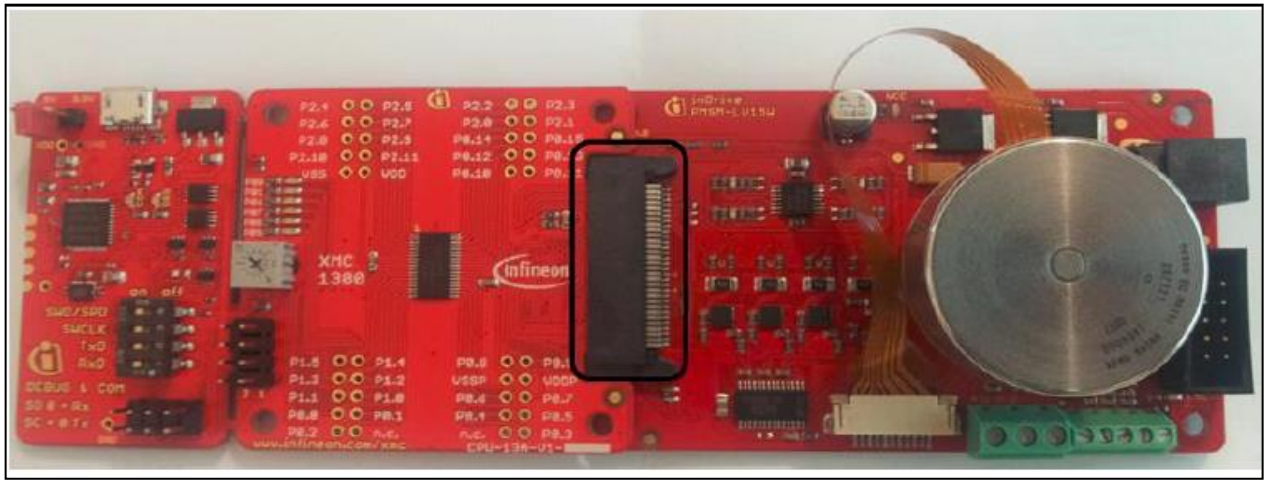
	Execution Time (us)	Code size (kbytes)
Control loop ISR	9.2	
Motor state machine	5.8	
BEMF zero cross ISR	14.2	13.976

Default configuration: Execution Time and Code Size

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)

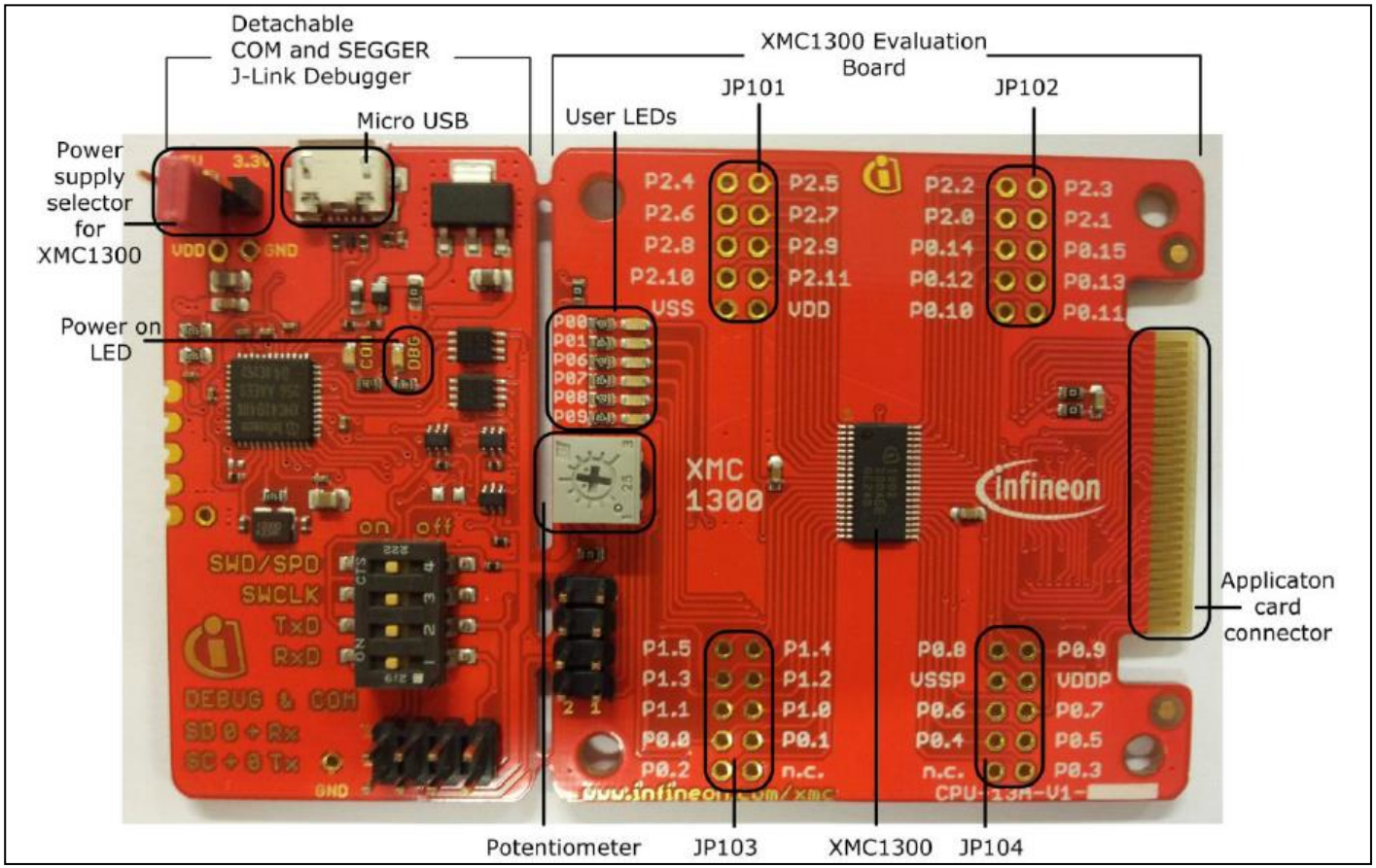


XMC1300 CPU Card

PMSM Low Voltage 15W Motor Card

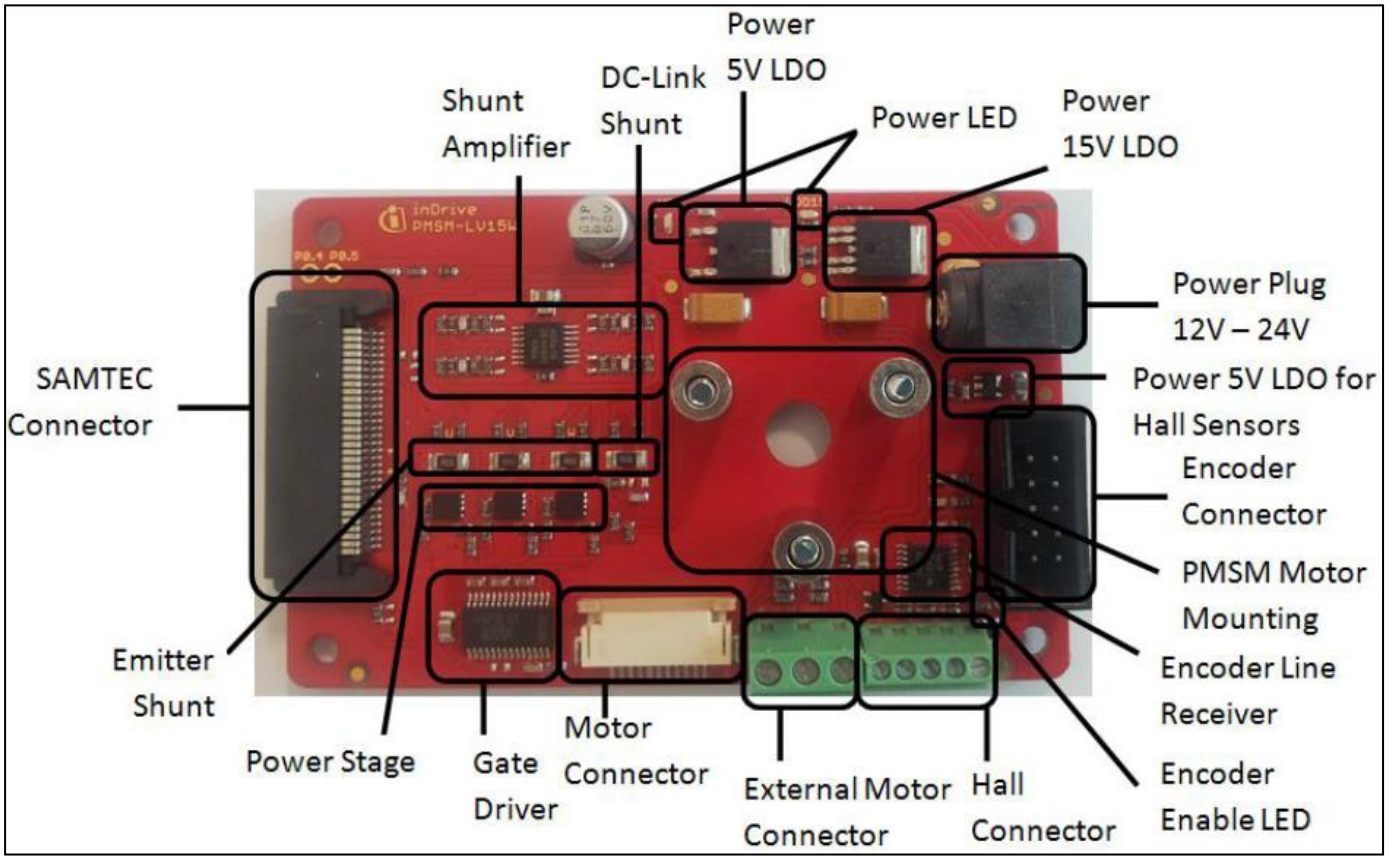
Hardware Overview – XMC1300 CPU Card

- 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

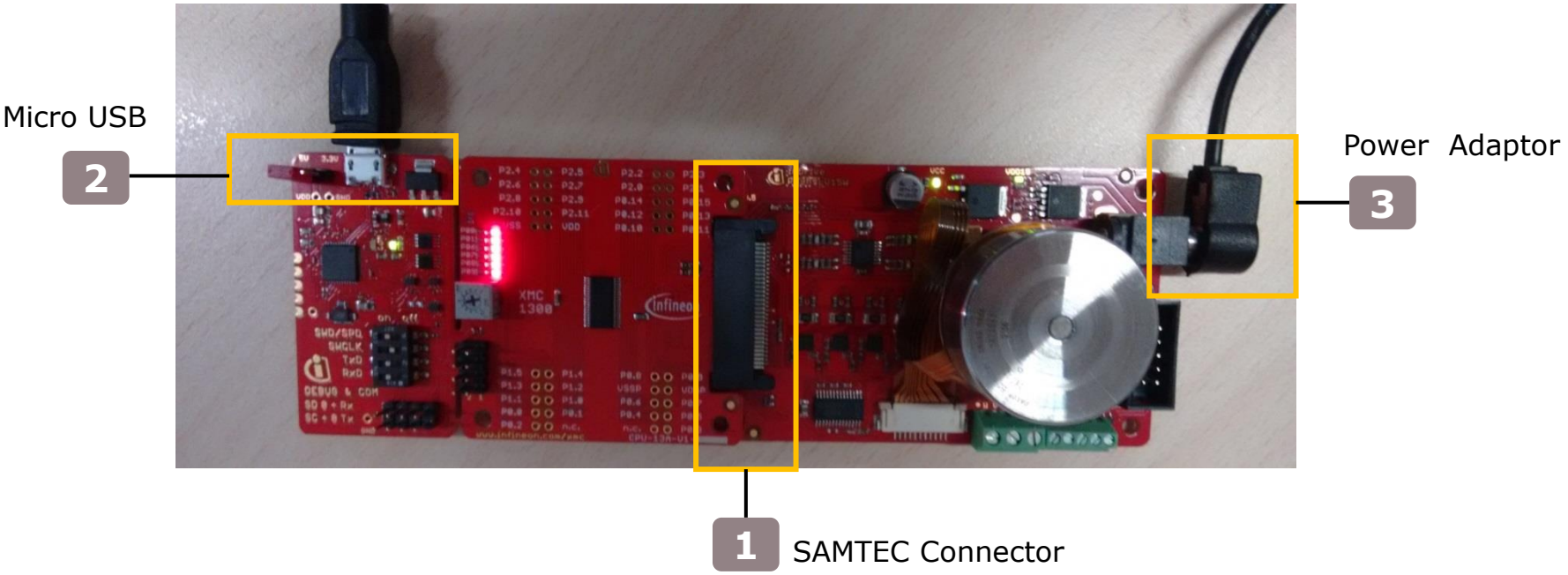
  - Download and unzip the installer package

- 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

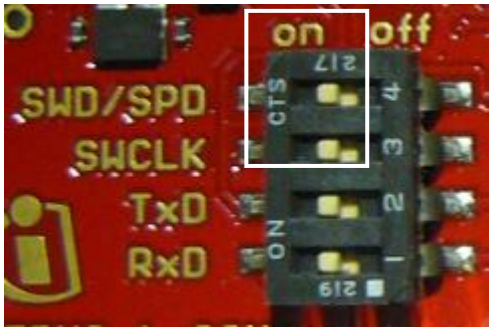
Getting Started – Connecting the Board

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



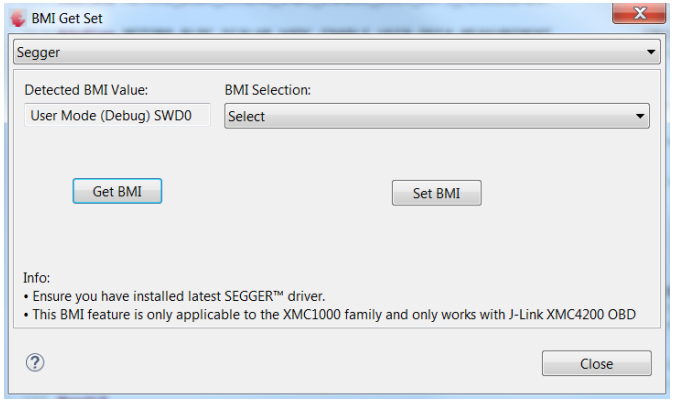
Getting Started – Setting up the Board

1. Check the SWD/SPD and SWCLK on the dip switch are set to **"ON"** position



2. Open DAVE™ 

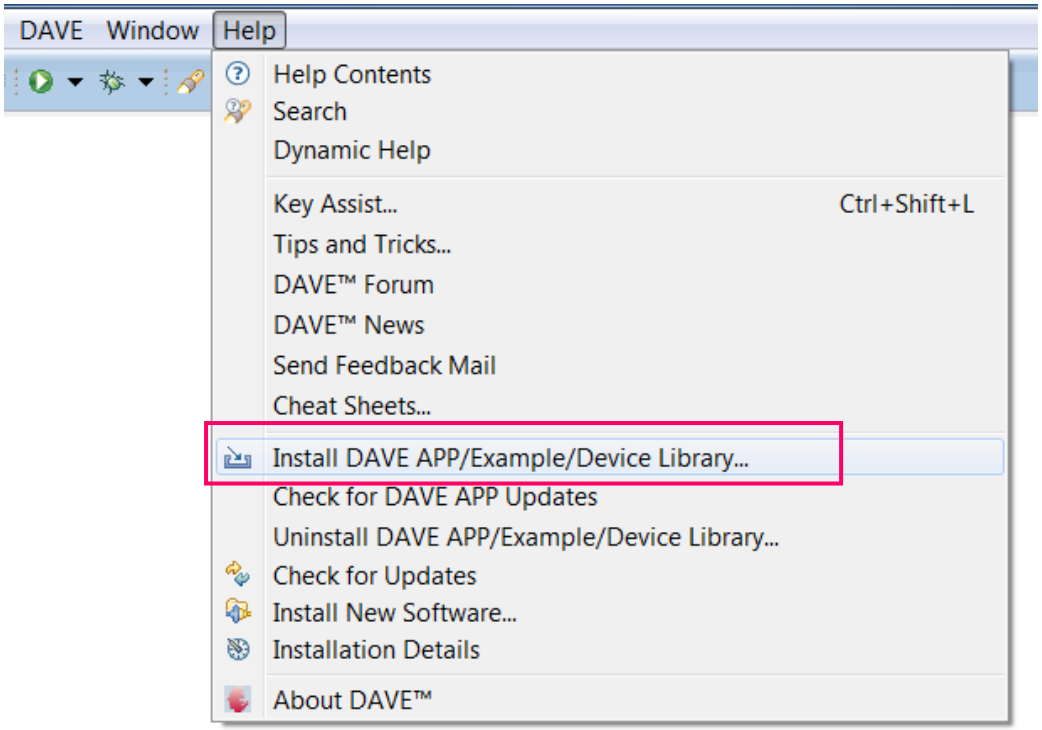
, select "*BMI Set Get*"  to check BMI is set to SWD0 mode



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

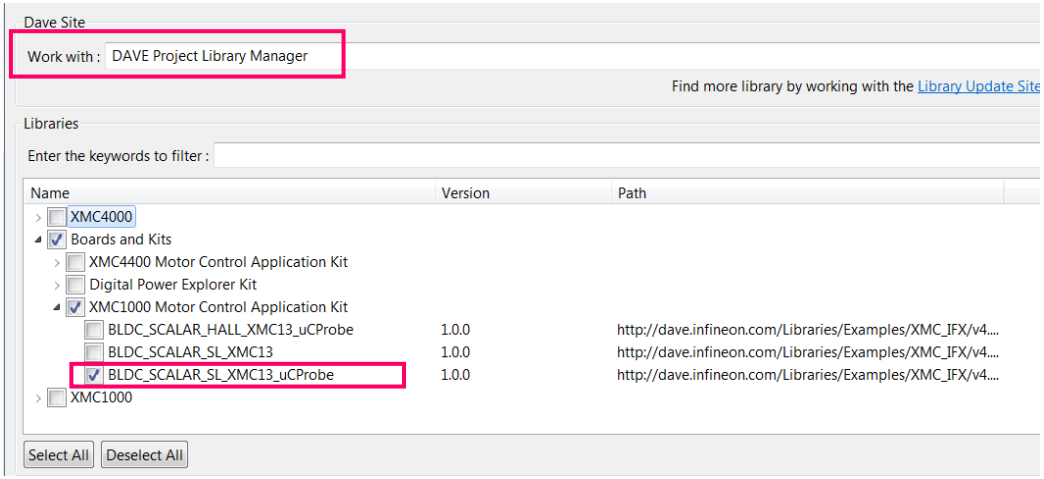
1. Open DAVE™ 

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_SL_XMC13_uCProbe"

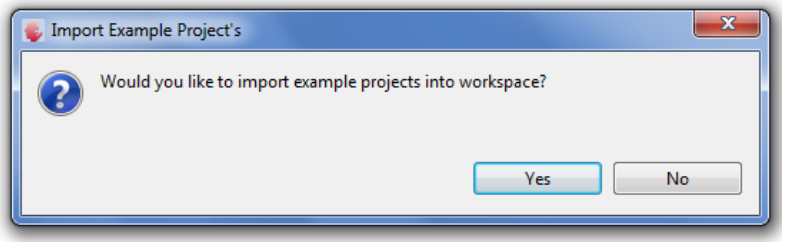


It is available at 2 locations.

[#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 [1/6]

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 [2/6]

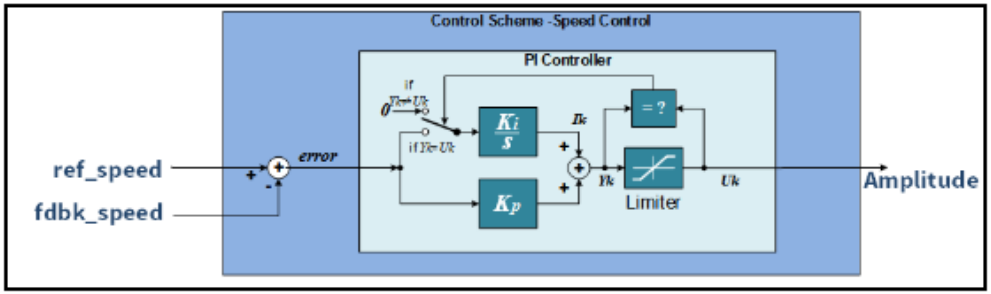
Folder: Configuration

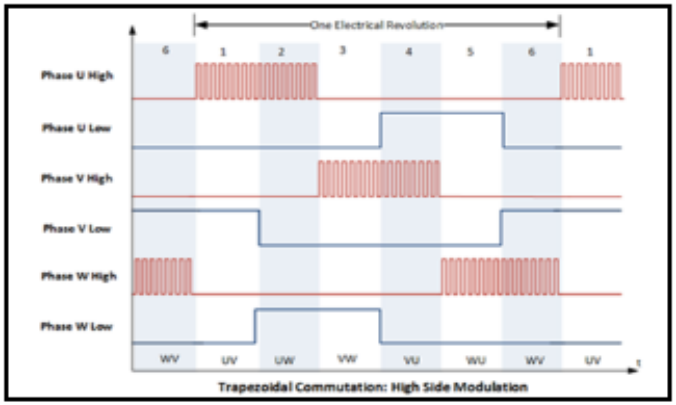
File name: bldc_scalar_user_config.h

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 [3/6]

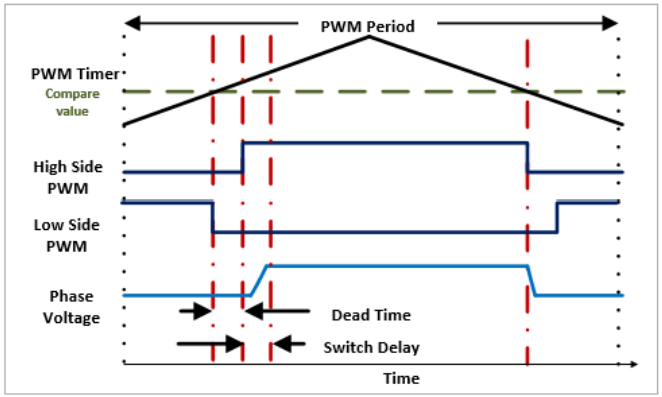
3. Configure the Power Board

Folder: Configuration
File name: bldc_scalar_user_config.h

```
/********************** 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 switch. Optior
#define   MOTOR0_BLDC_SCALAR_LS_SWITCH_ACTIVE_LEVEL       (BLDC_SCALAR_ACTIVE_HIGH)       /*!< Active level of the low side switch. Options
#define   MOTOR0_BLDC_SCALAR_INVERTER_ENABLE_CONF         (BLDC_SCALAR_INV_ACTIVE_HIGH)   /*!< Active level of inverter enable. Options: BI

/* 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 measurement */
#define   MOTOR0_BLDC_SCALAR_BEMF_DIVIDER_RATIO           (9.79F)      /*!< Voltage divider ratio in % for phase voltage measurement */
/* end of #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
```



Getting Started – Configure the Project [4/6]

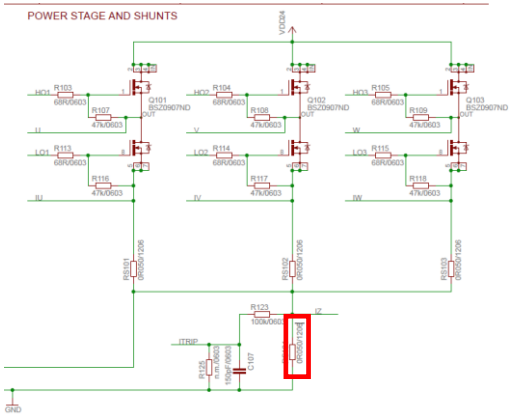
4. Configure the Power Board

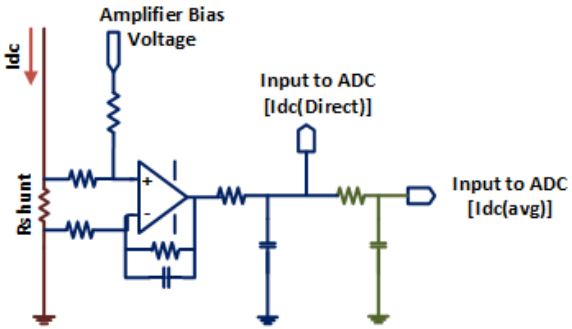
Folder: Configuration
File name: bldc_scalar_user_config.h

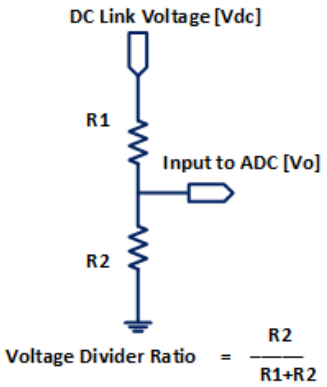
```
/********************** Power board parameters ************************************************************/
#if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
/* Power Inverter parameters */
#define    MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT         (24.0F)     /*!< DC link voltage */
#define    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 switch. Optior
#define    MOTOR0_BLDC_SCALAR_LS_SWITCH_ACTIVE_LEVEL        (BLDC_SCALAR_ACTIVE_HIGH)       /*!< Active level of the low side switch. Options
#define    MOTOR0_BLDC_SCALAR_INVERTER_ENABLE_CONF          (BLDC_SCALAR_INV_ACTIVE_HIGH)   /*!< Active level of inverter enable. Options: BI

/* 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 measurement */
#define    MOTOR0_BLDC_SCALAR_BEMF_DIVIDER_RATIO           (9.79F)      /*!< Voltage divider ratio in % for phase voltage measurement */
/* end of #if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
```







Getting Started – Configure the Project [5/6]

5. Configure the startup method

Folder: Configuration

File name: bldc_scalar_user_config.h

```
/*******************************************************************************************************************
 *                                    INITIAL POSITION CONFIGURATIONS
 *******************************************************************************************************************/
/**
 * Sensorless startup method selection
 * Options - BLDC_SCALAR_SL_INDUCTIVE_SENSING, BLDC_SCALAR_SL_ALIGN
 */
#define MOTOR0_BLDC_SCALAR_SL_STARTUP_TECHNIQUE       (BLDC_SCALAR_SL_INDUCTIVE_SENSING)

#if (MOTOR0_BLDC_SCALAR_SL_STARTUP_TECHNIQUE == BLDC_SCALAR_SL_INDUCTIVE_SENSING)
#define MOTOR0_BLDC_SCALAR_SL_IND_SENSE_PATTERN_ONTIME        (0.2F)     /*!< CCU8 pulse width in mSec. Phase patterns are energized in
                                                                             sequence for this time. Range: 0.0005 to 32 mSec */
#define MOTOR0_BLDC_SCALAR_SL_IND_SENSE_CURRENT_DECAY_TIME    (0.2F)     /*!< Time in mSec for current to decay after pulse is applied. Range: 0.0005
```

6. Configure the Inductive Sensing configuration (refer to DAVE help file -> BLDC_SCALAR_SL_XMC13.chm)

```
/*******************************************************************************************************************
 *                                    STARTUP CONFIGURATIONS
 *******************************************************************************************************************/
#define MOTOR0_BLDC_SCALAR_SL_STARTUP_MIN_AMPLITUDE          (10.0F)        /*!< in %. Minimum Duty cycle to be applied for sensorless startup.
                                                                                 Duty cycle is then controlled by current control loop.
                                                                                 Range: 0.1 to 10 */
#define MOTOR0_BLDC_SCALAR_SL_FIRST_KICK_TIME                (4.0F)        /*!< First kick time in mSec. Phase pattern corresponding to identified po
                                                                                 is energized for this time.
                                                                                 Min Range: 1/MOTOR0_BLDC_SCALAR_PWM_FREQ */

#define MOTOR0_BLDC_SCALAR_SL_TR_PHASE_ENERGIZATION_TIME     (0.8F)        /*!< in mSec. Phases are energized in sequence for this time
                                                                                 during transition state from first kick to closed loop.
                                                                                 Min Range: 1/MOTOR0_BLDC_SCALAR_PWM_FREQ */
#define MOTOR0_BLDC_SCALAR_SL_TR_CURRENT_DECAY_TIME          (0.8F)        /*!< in mSec. Wait time for current to decay.
                                                                                 Min Range: 1/MOTOR0_BLDC_SCALAR_PWM_FREQ */

#define MOTOR0_BLDC_SCALAR_SL_MIN_BEMF_VOLTAGE               (2.0F)       /*!< in % with respect to MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT
                                                                             if measured BEMF voltage is less than this voltage during startup, pos
#define MOTOR0_BLDC_SCALAR_SL_TRANSITION_BEMF_VOLTAGE        (10.0F)      /*!< in % with respect to MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT
                                                                             Motor switches to closed loop when measured BEMF voltage is greater th
```

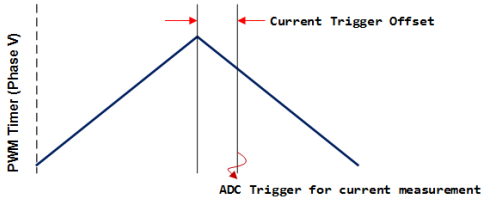
Getting Started – Configure the Project [6/6]

7. Configure the ADC Trigger

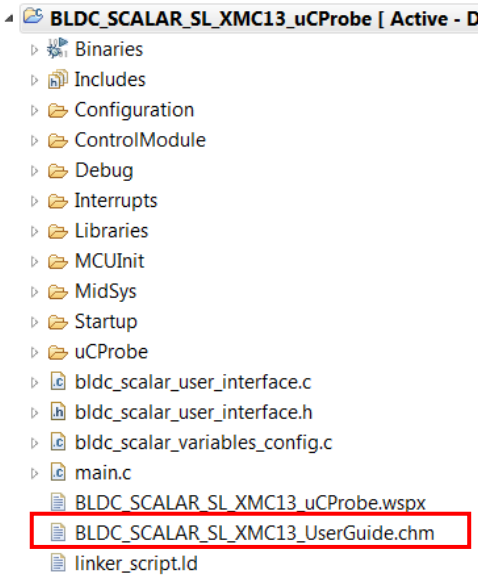
Folder: Configuration

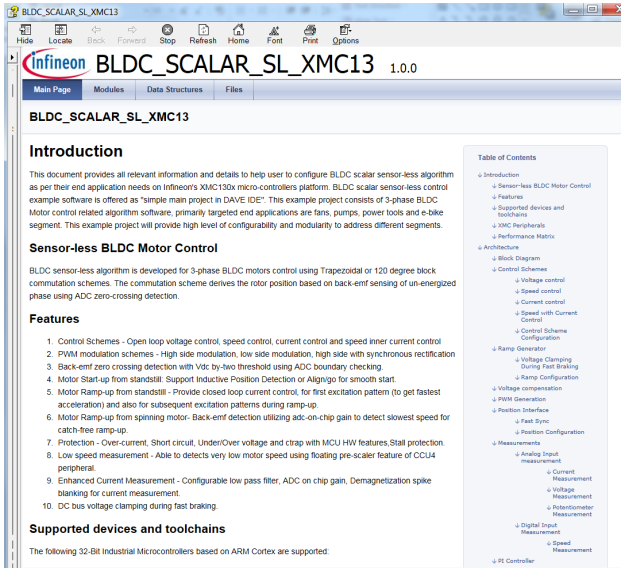
File name: bldc_scalar_user_config.h

```
445 #define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER                (BLDC_SCALAR_SL_BEMF_SINGLE_TRIGGER)
446
447
448 #define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER_OFFSET          (0.5F)     /*!< BEMF measurement trigger time in uSec. BEMF is measured at
449                                                                          (End of PWM ON time - configured time).
450                                                                          This time should be greater than sampling time of VADC (0.1 uSec) */
```



8. Get more detail information from the BLDC_SCALAR_SL_XMC13_UserGuide.chm





Getting Started – Code added to support uCProbe [1/3]

Folder: -
File name: main.c

1. Initialize the uCProbe before starting motor

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

Getting Started – Code added to support uCProbe [2/3]

Folder: Interrupts

File name:
bldc_scalar_sl_state_machine.c

2. Added uCProbe scheduler in motor state machine

```
140 void SysTick_Handler(void)
141 {
142   /* Call motor control state machine */
143   Motor0_BLDC_SCALAR_MSM();
144 }
145
146 /**
147  * @}
148  */
149 /**
150  * @}
151  */
152
153 RAM_ATTRIBUTE void Motor0_BLDC_SCALAR_MSM(void)
154 {
155   switch (Motor0_BLDC_SCALAR.msm_state)
156   {
157     case BLDC_SCALAR_MSM_NORMAL_OPERATION:
158       Motor0_BLDC_SCALAR_MSM_NORMAL_OPERATION_Func();
159       break;
160
161     case BLDC_SCALAR_MSM_IDLE:
162       Motor0_BLDC_SCALAR_MSM_IDLE_Func();
163       break;
164
165     case BLDC_SCALAR_MSM_MOTOR_STATE_IDENTIFICATION:
166       Motor0_BLDC_SCALAR_MSM_MOTOR_STATE_IDENTIFICATION_Func();
167       break;
```

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

```
187     case BLDC_SCALAR_MSM_START:
188       Motor0_BLDC_SCALAR_MSM_START_Func();
189       break;
190
191     case BLDC_SCALAR_MSM_STOP:
192       Motor0_BLDC_SCALAR_MotorStop();
193       break;
194
195     default:
196       break;
197
198   }
199
200   if (Motor0_BLDC_SCALAR.error_status != 0U)
201   {
202     Motor0_BLDC_SCALAR.msm_state = BLDC_SCALAR_MSM_ERROR;
203   }
204 #if (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
205 Motor0_BLDC_SCALAR_uCProbe_Scheduler();
206 #endif
207 }
208
```

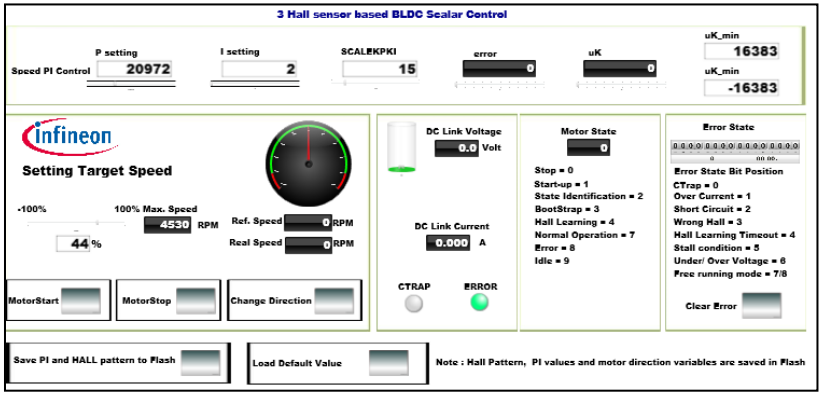
Getting Started – Code added to support uCProbe [3/3]

3. Added uCProbe scheduler in motor state machine

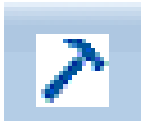
Folder: uCProbe
File name: ucProbe.c

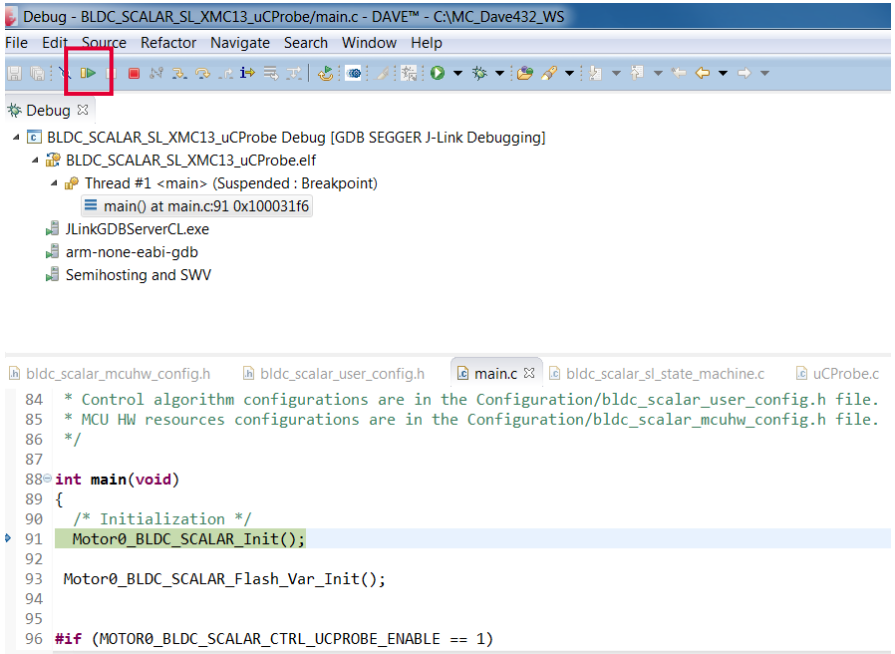
```
73 #if ((MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE==1))
74 /*UCproBE scheduler function to handle ucprobe comments from UI */
75 void Motor0_BLDC_SCALAR_uCProbe_Scheduler(void)
76 {
77   switch(Motor0_BLDC_SCALAR_ucprobe.control_word)
78   {
79     case 1:  /* Start the motor */
80       Motor0_BLDC_SCALAR_ucprobe.control_word=0;
81       Motor0_BLDC_SCALAR_MotorStart();
82       break;
83
84     case 2: /*Stop the motor*/
85       Motor0_BLDC_SCALAR_ucprobe.control_word=0;
86       Motor0_BLDC_SCALAR_MotorStop();
87       break;
88
89     case 3: /*Clear Error state*/
90       Motor0_BLDC_SCALAR_ucprobe.control_word=0;
91       Motor0_BLDC_SCALAR_ClearErrorState();
92       break;
93
94     case 4:  /*Clear flash and load defualt value into flash*/
95       Motor0_BLDC_SCALAR_ucprobe.control_word=0;
96       Motor0_BLDC_SCALAR_ucprobe.user_config[0] =0;
97       Motor0_BLDC_SCALAR_Write_Default_value();
98       Motor0_BLDC_SCALAR_uCProbe_Write_Flash();
99       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



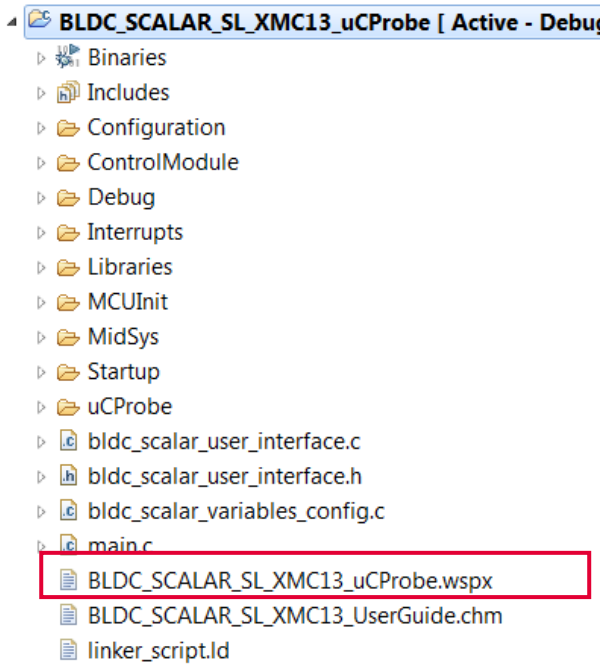
Observation:

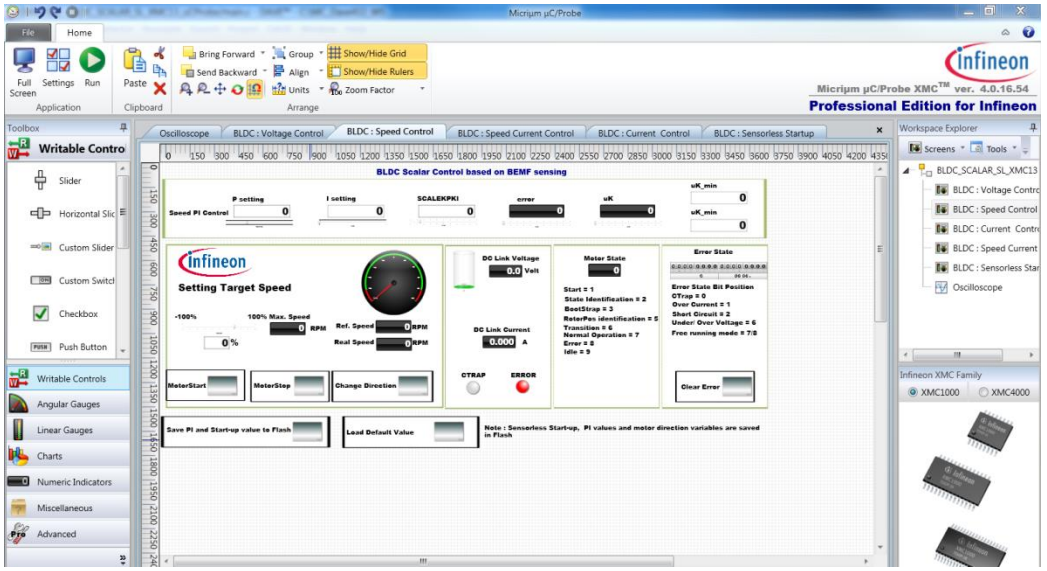
› Motor should ramp to 2000RPM with ramp rate of 500RPM/s

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

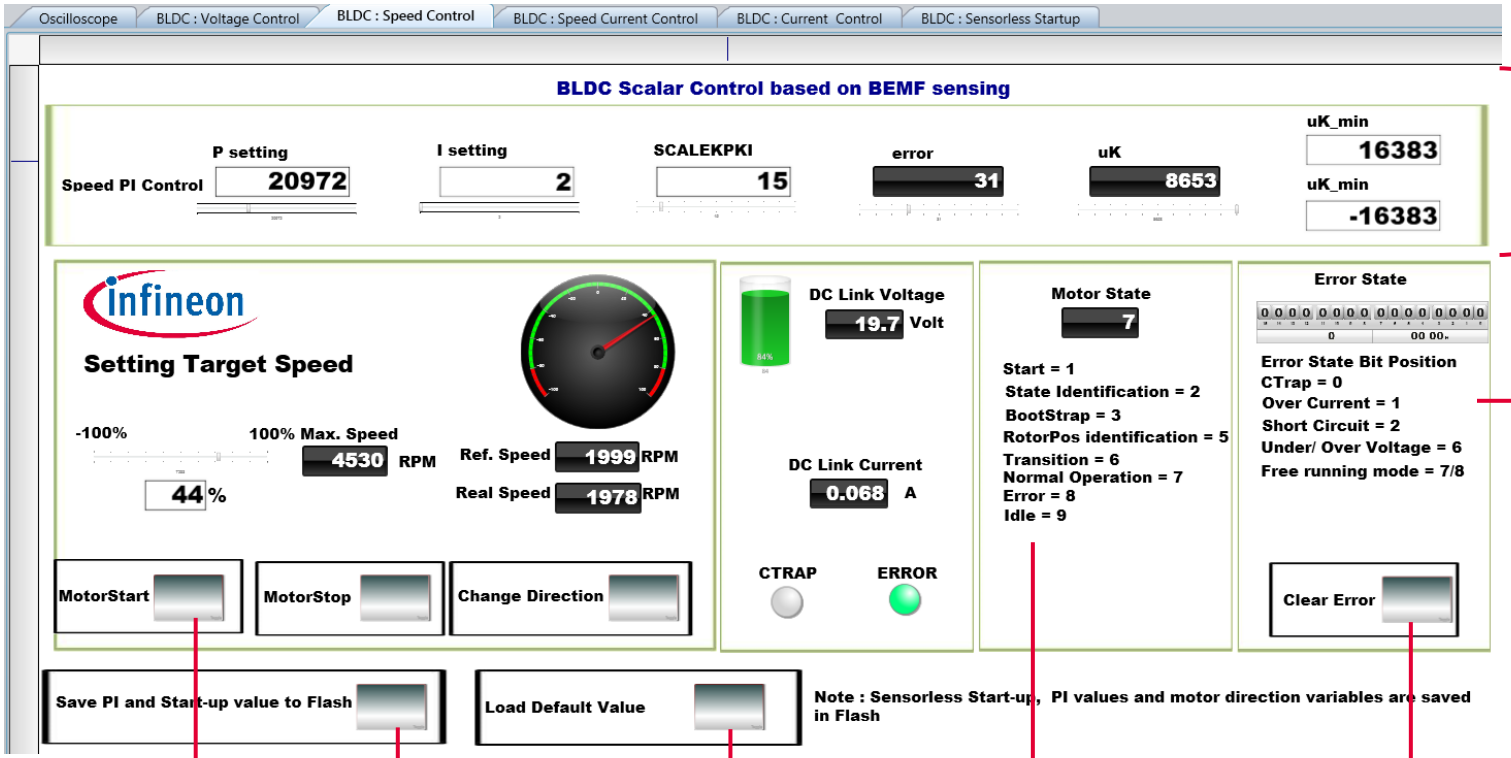
- Update of the motor and monitoring motor parameters can be executed using µC/Probe™ XMC™

1. In "BLDC_SCALAR_SL_XMC13" example project , open µC/Probe™ XMC™ project file





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

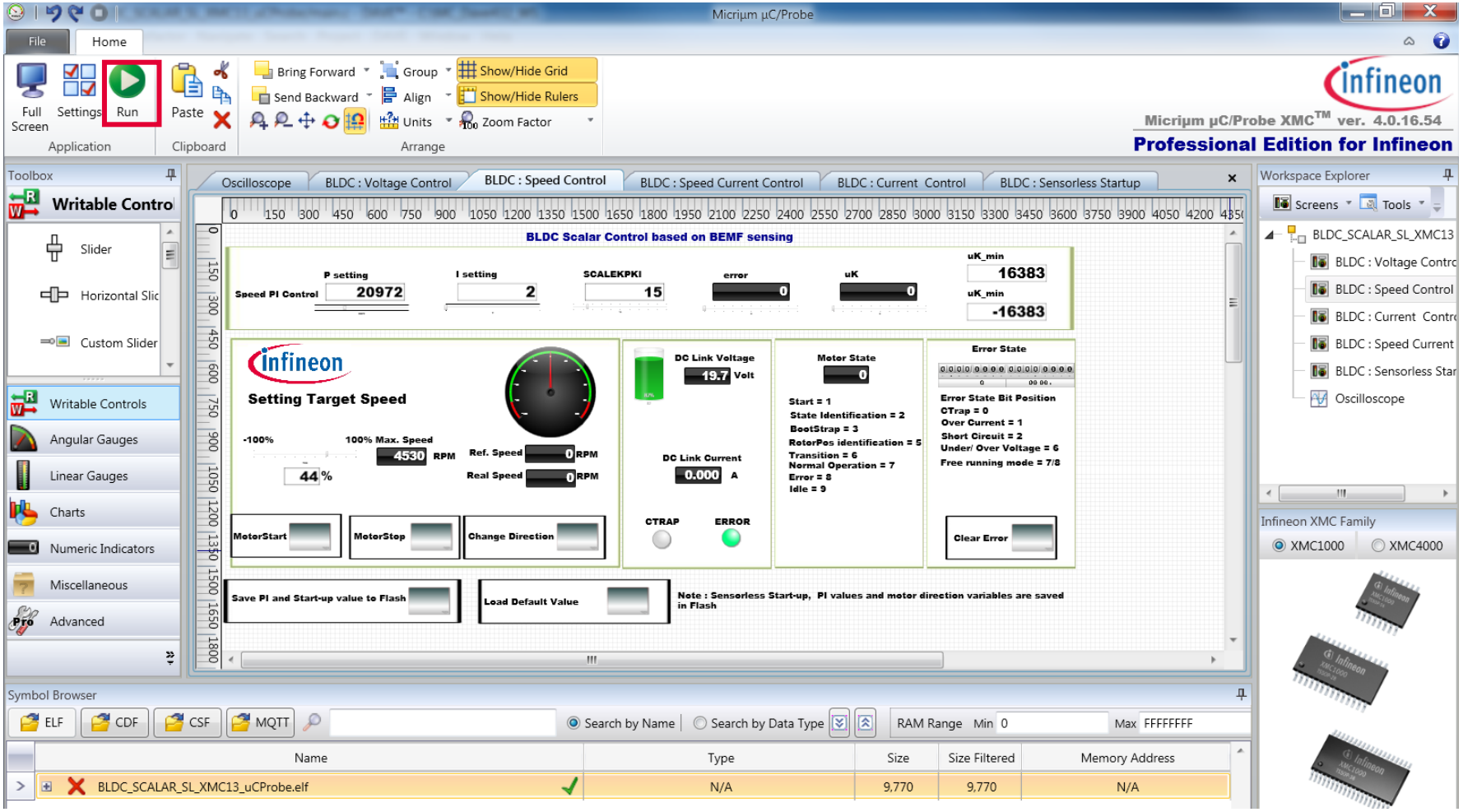
Load the default values into flash

Motor state Indication

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

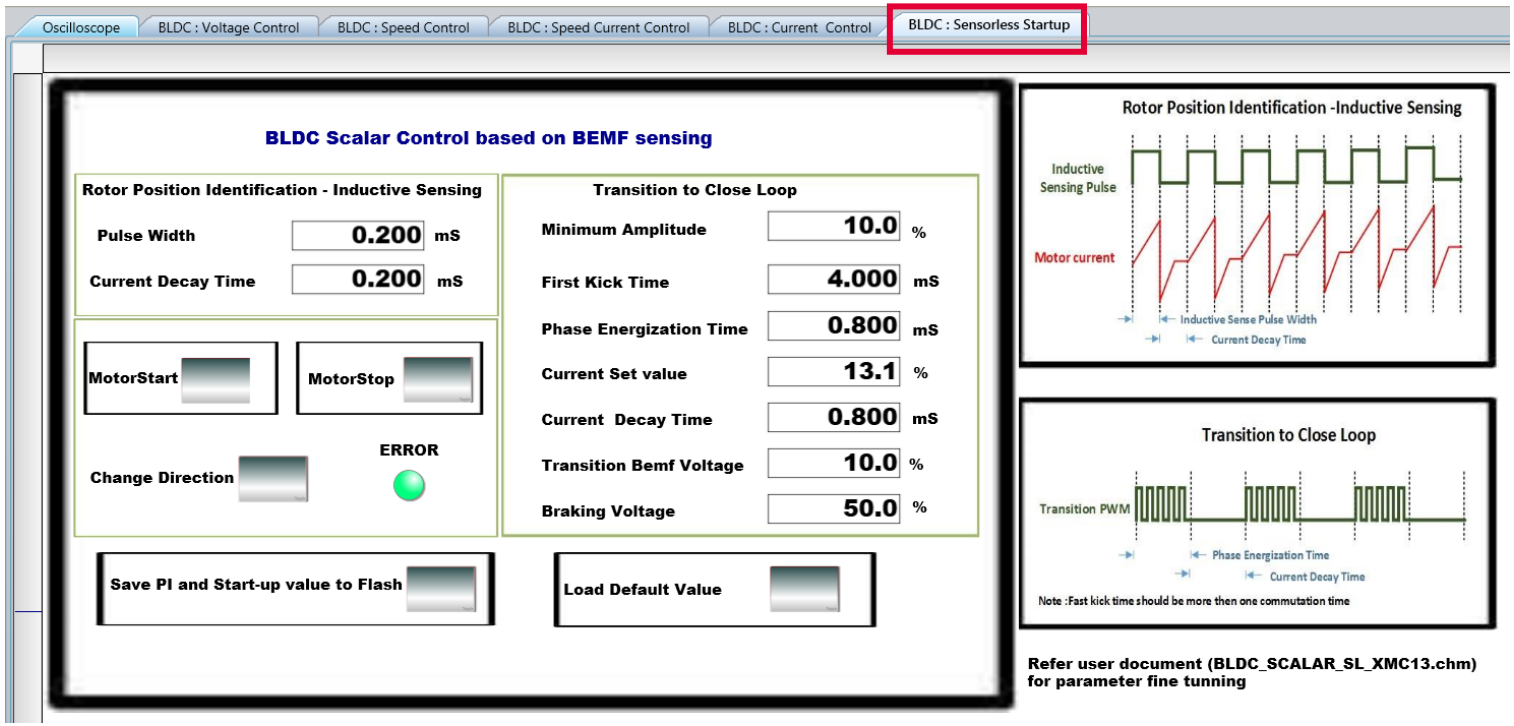
Getting Started – Interface with µC/Probe [3/6]

2. Click the 'Run' button



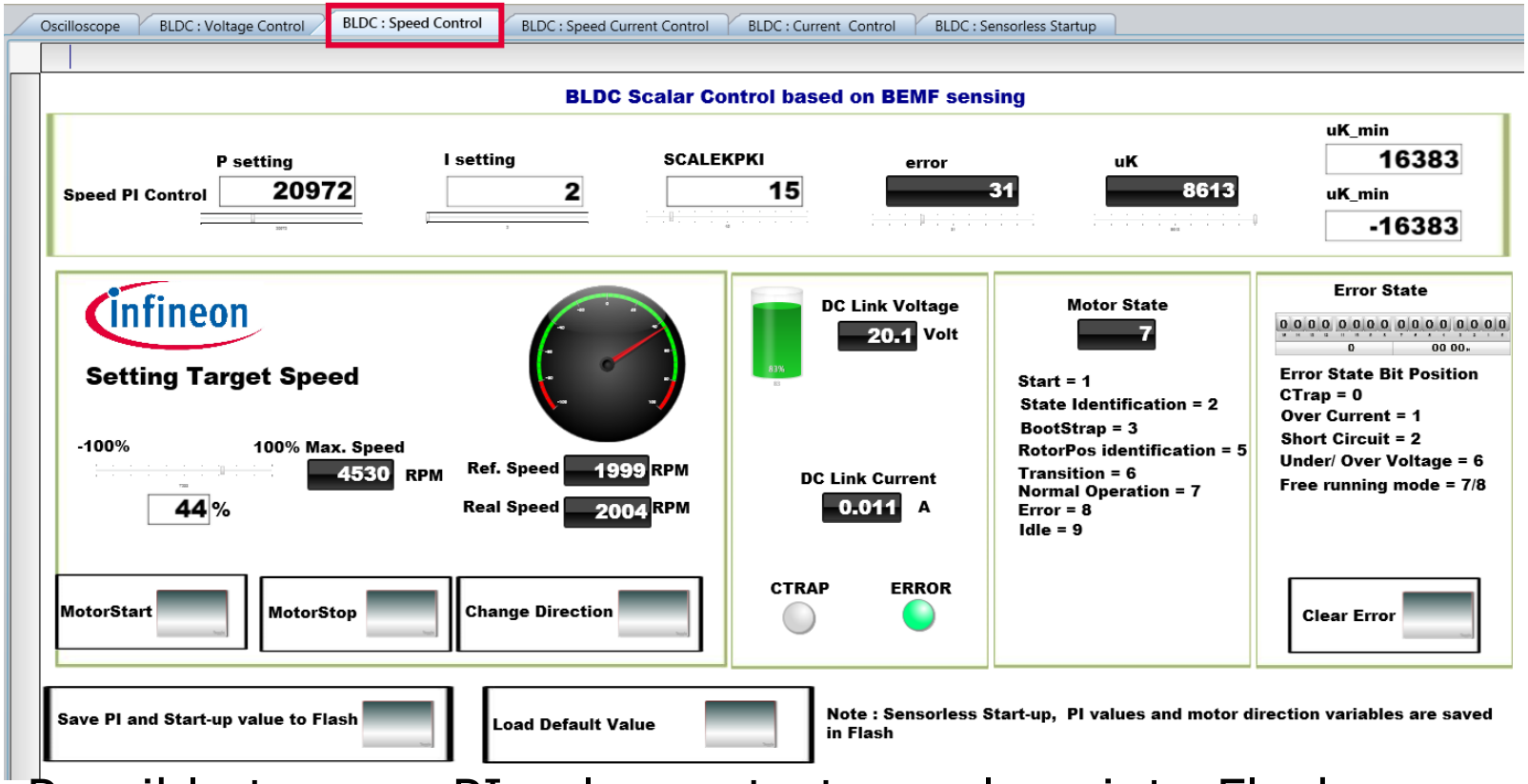
Getting Started – Interface with µC/Probe [4/6]

3. Go to Tab: Sensorless Startup. This is to find the rotor position when the rotor is in a standstill position
4. Set the pulse width and current decay time to 0.2ms
5. Once rotor position is determined, it will transit to closed loop control
6. Select "Save to Flash" to save the inductive sensing parameter into the Flash.



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

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



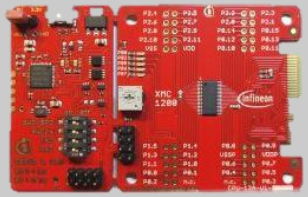
› Possible to save PI values, startup values into Flash

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

8. Click on the “Oscilloscope” tab for monitoring motor control parameters -> e.g. speed of rotation in rpm

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:

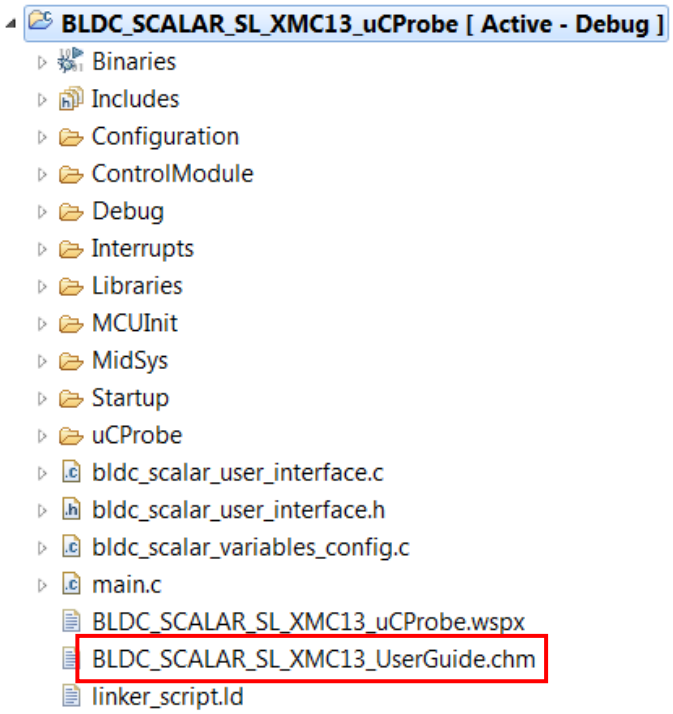
  http://www.infineon.com/DAVE

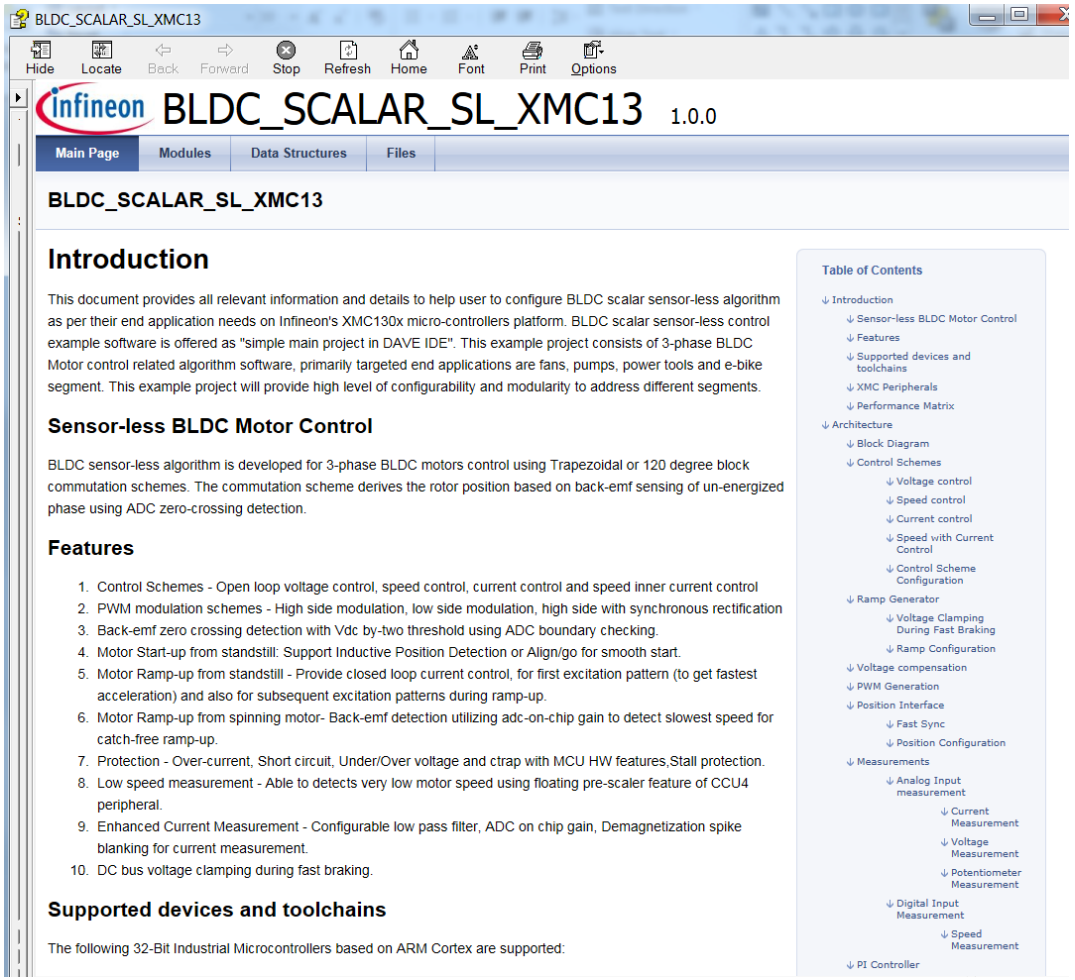
- For support:

  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.