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

XMC^{TM} Microcontrollers
Oct 2016

Agenda

1 Overview of BLDC Scalar Control SW

2 Software Overview

3 Hardware Overview

4 Tools Overview

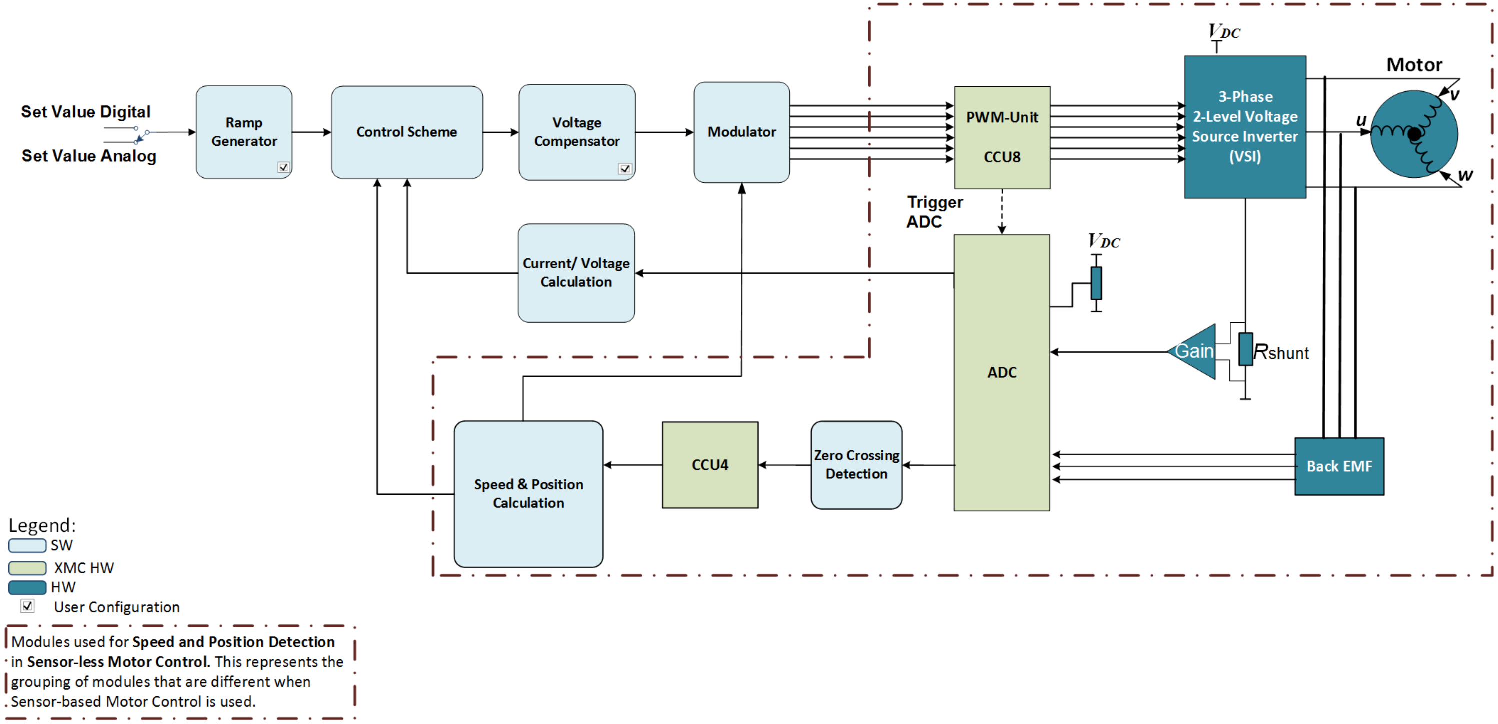
5 Getting Started

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^{TM} 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



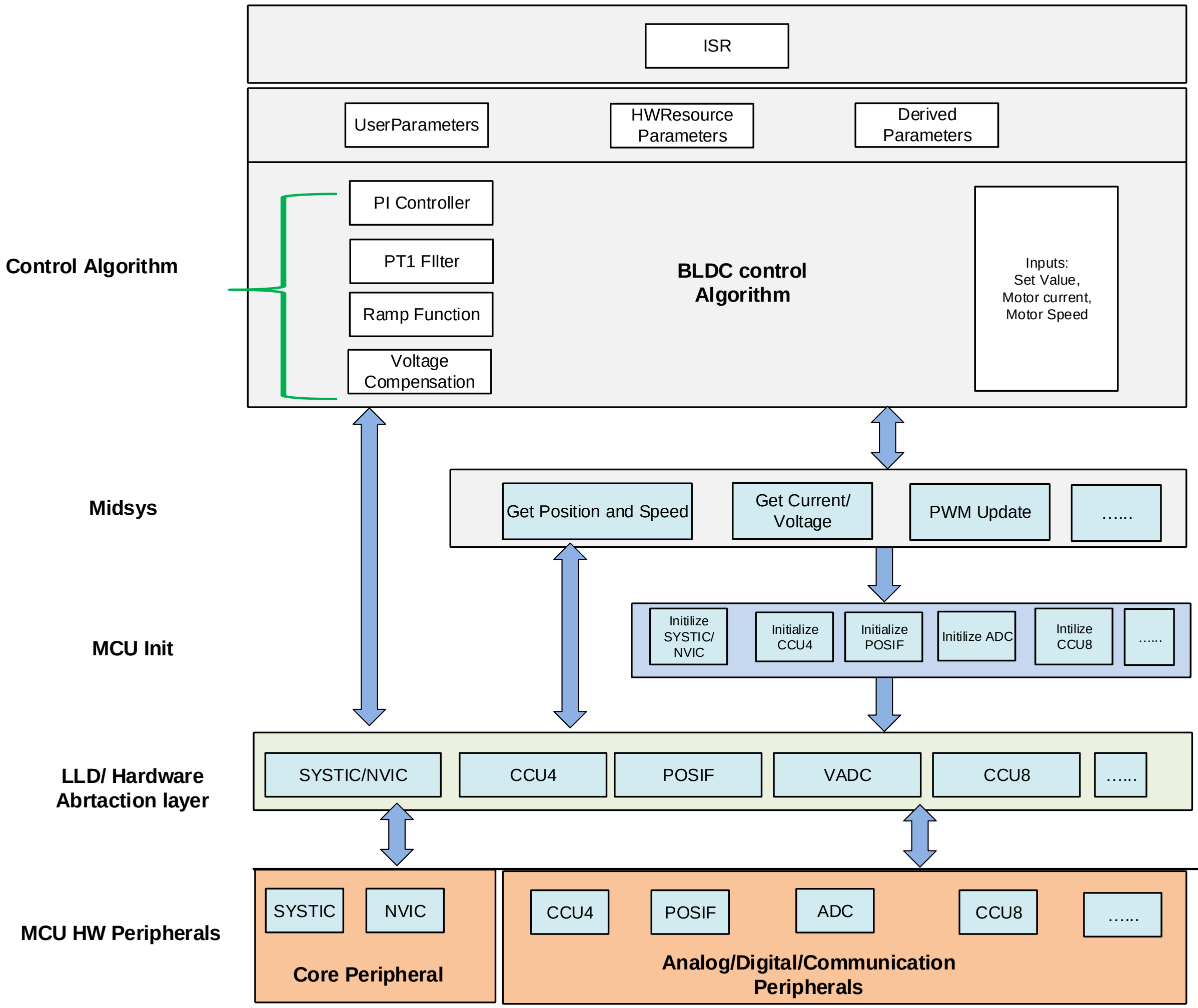
Modules used for **Speed and Position Detection** in **Sensor-less Motor Control.** This represents the grouping of modules that are different when Sensor-based Motor Control is used.

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



Folder/File Structure

Main.c
bldc_scalar_user_interface.c
bldc_scalar_varaibles_config.c
bldc_scalar_varaibles_config.h

Interrupts:
bldc_scalar_control_loop.c
bldc_scalar_state_machine.c
bldc_scalar_patter_update.c
bldc_scalar_hall_event.c
bldc_scalar_ctrap.c
bldc_scalar_protection_error.c

Configuratoin:
bldc_scalar_MCUHWconfig.h
bldc_scalar_user_parameter.h
bldc_scalar_derived_parameter.h
bldc_scalar_derived_parameter.c

ControlModule:
bldc_scalar_control_scheme.c
bldc_scalar_control_scheme.h
bldc_scalar_pi.c
bldc_scalar_pi.h
bldc_scalar_pt1_filter.c
bldc_scalar_pt1_filter.h
bldc_scalar_ramp_generator.c
bldc_scalar_ramp_generator.h
bldc_scalar_control_hall.c
bldc_scalar_control_hall.h

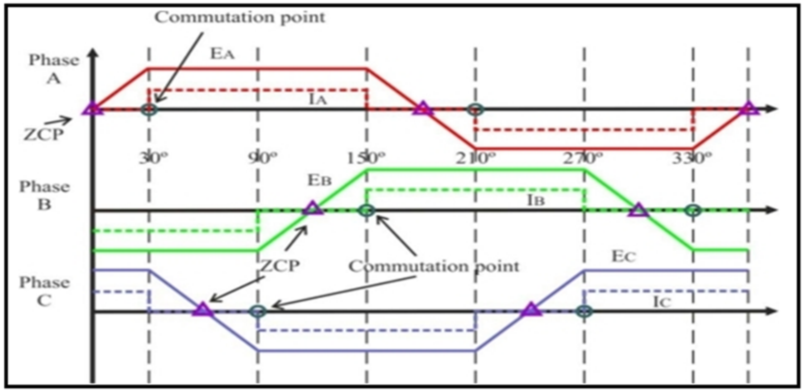
MidSys:
bldc_scalar_pwm_bc.c
bldc_scalar_pos_spd_hall.c
bldc_scalar_volt_dcbus.c
bldc_scalar_volt_potentiometer.c
bldc_scalar_volt_phase.c
bldc_scalar_current_motor.c

MCUInit:
ccu4.c
ccu8.c
posif.c
nvic.c
gpio.c

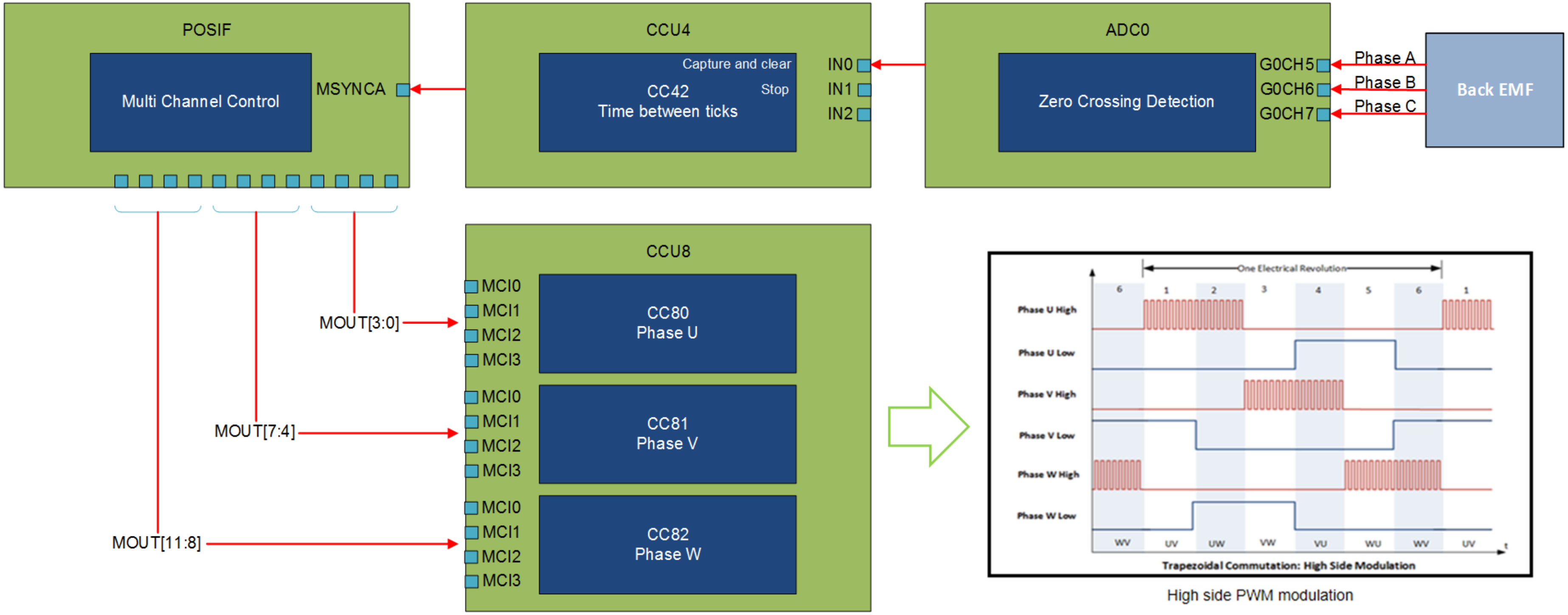
Software Overview - XMC Peripheral usage

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

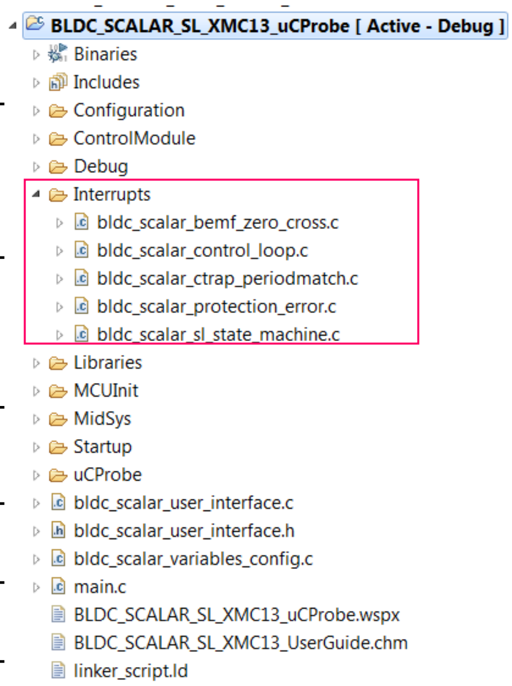


High side PWM modulation

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



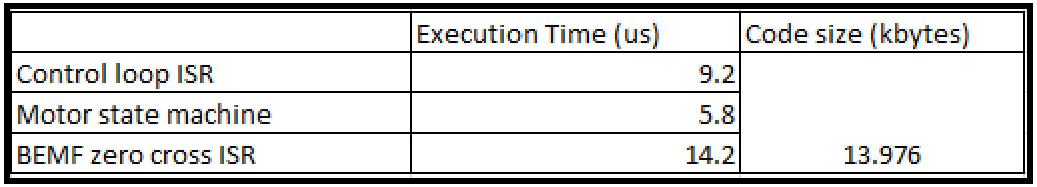
```
BLDC_SCALAR_SL_XMC13_uCProbe [ Active - Debug ]
  Binaries
  Includes
  Configuration
  ControlModule
  Debug
  Interrupts
    bldc_scalar_bemf_zero_cross.c
    bldc_scalar_control_loop.c
    bldc_scalar_ctrap_periodmatch.c
    bldc_scalar_protection_error.c
    bldc_scalar_sl_state_machine.c
  Libraries
  MCUInit
  MidSys
  Startup
  uCProbe
  bldc_scalar_user_interface.c
  bldc_scalar_user_interface.h
  bldc_scalar_variables_config.c
  main.c
  BLDC_SCALAR_SL_XMC13_uCProbe.wspx
  BLDC_SCALAR_SL_XMC13_UserGuide.chm
  linker_script.ld
```

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



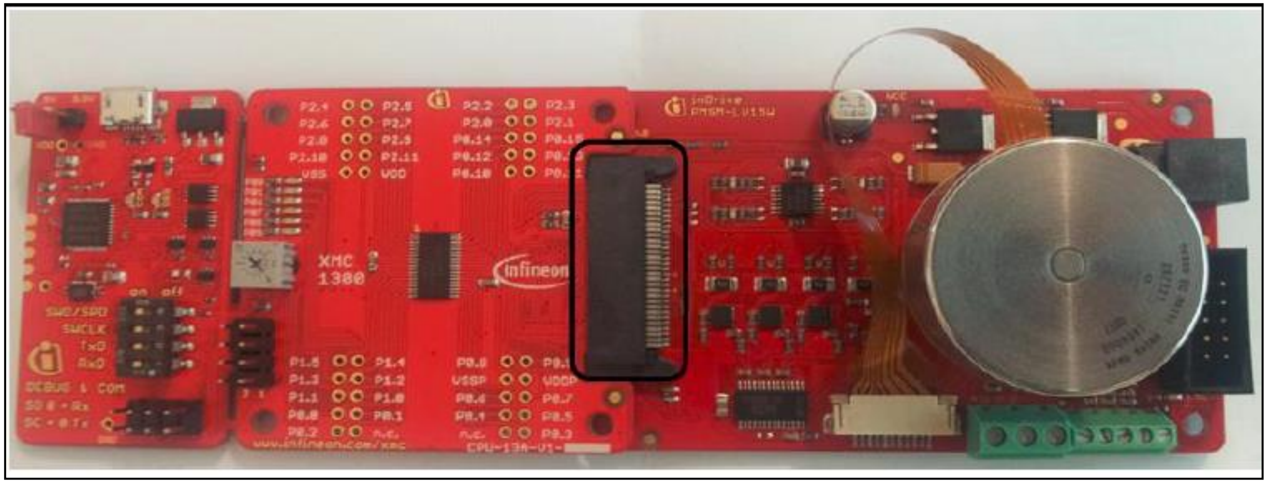
	Execution Time (us)	Code size (kbytes)
Control loop ISR	9.2	13.976
Motor state machine	5.8	
BEMF zero cross ISR	14.2	

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)

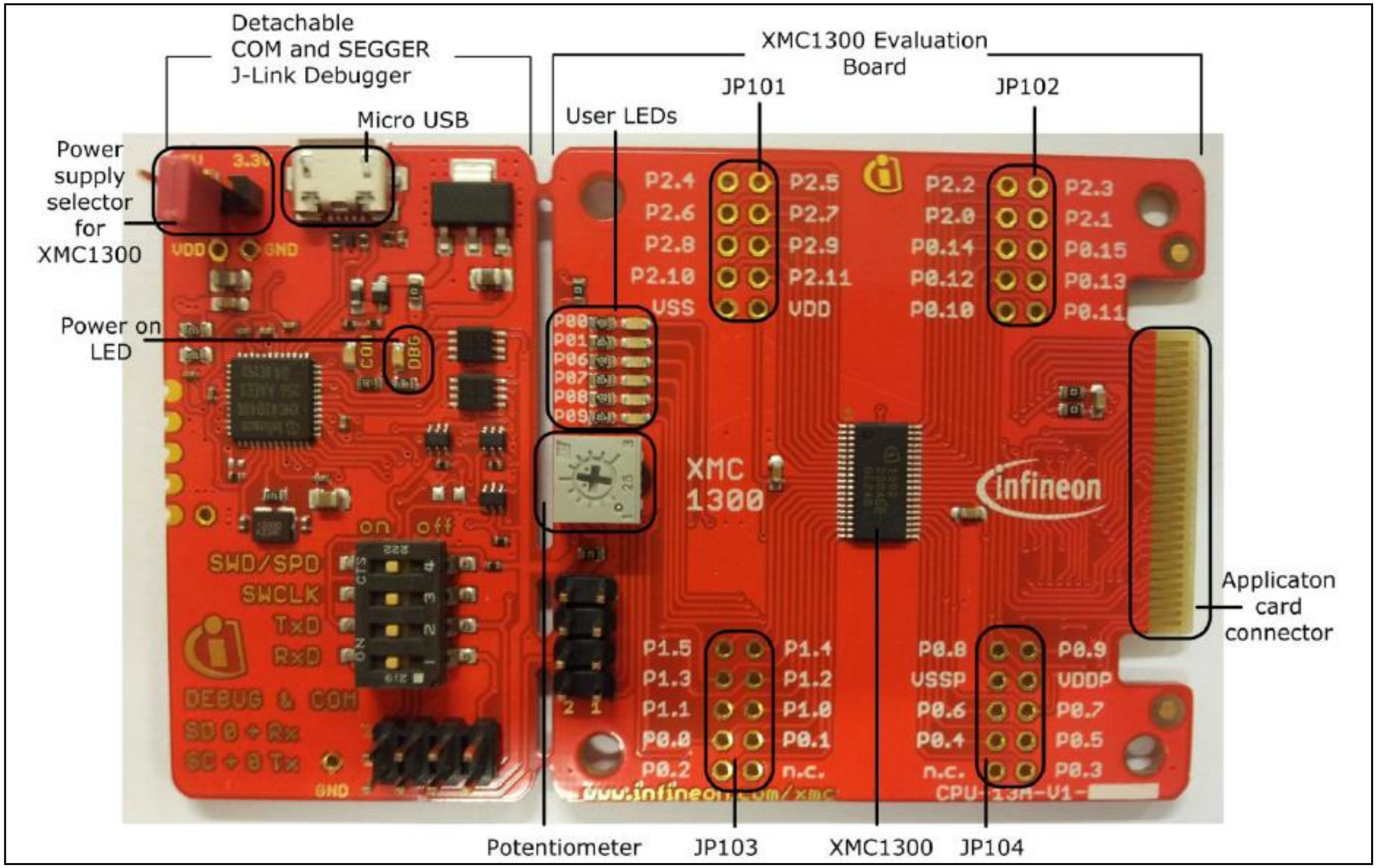


XMC1300 CPU Card

PMSM Low Voltage 15W Motor Card

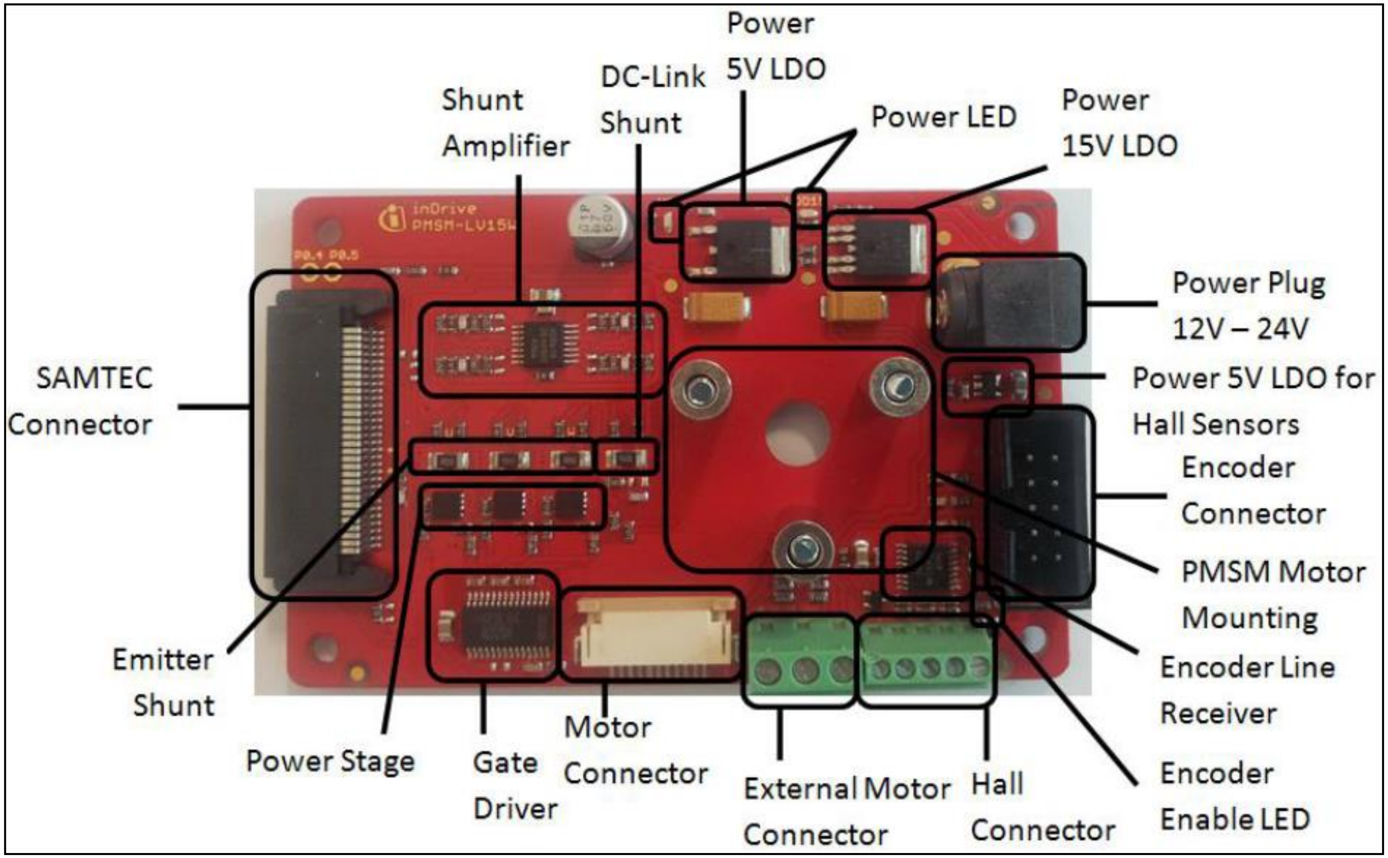
Hardware Overview – XMC1300 CPU Card

› XMC1300 CPU Card



Hardware Overview – Motor Card

› PMSM Low Voltage 15W Motor Card

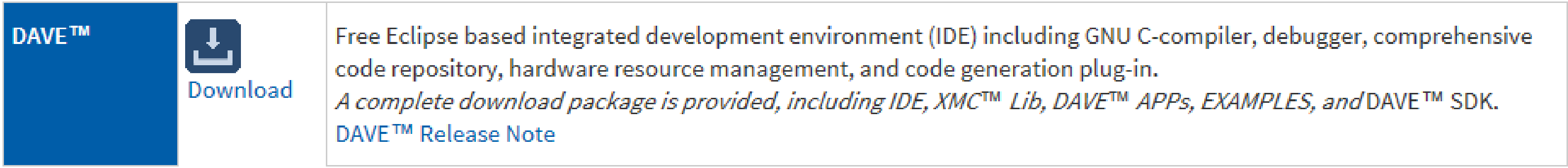


Hardware Overview – Kit Order information

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



DAVE™

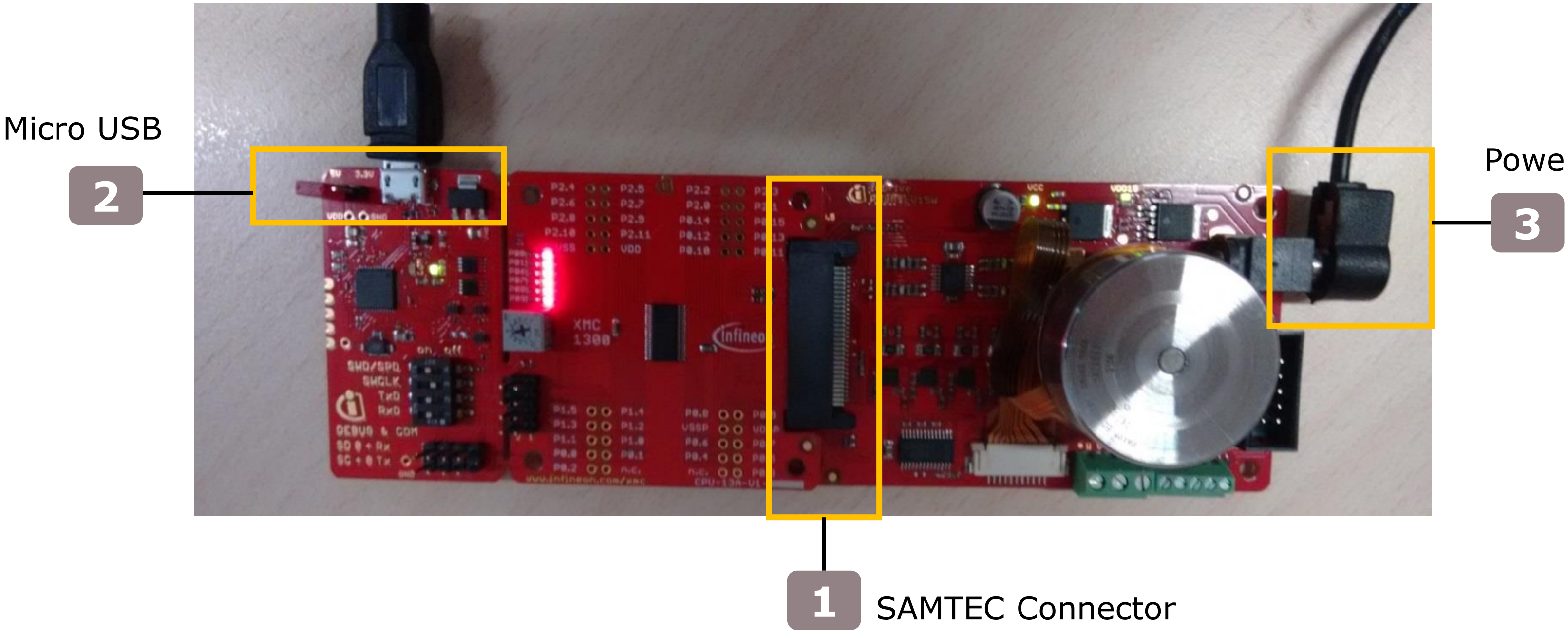
Download

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

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



Micro USB

2

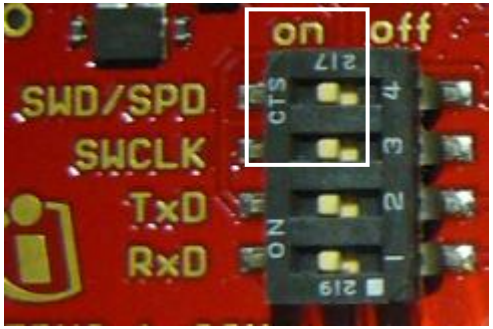
Power Adaptor

3

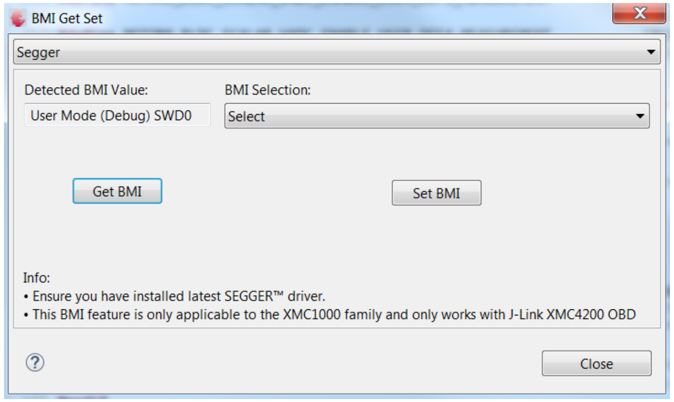
1 SAMTEC Connector

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™ DAVE-4.2.6, select *"BMI Set Get"* to check BMI is set to SWD0 mode



Detected BMI Value: User Mode (Debug) SWD0

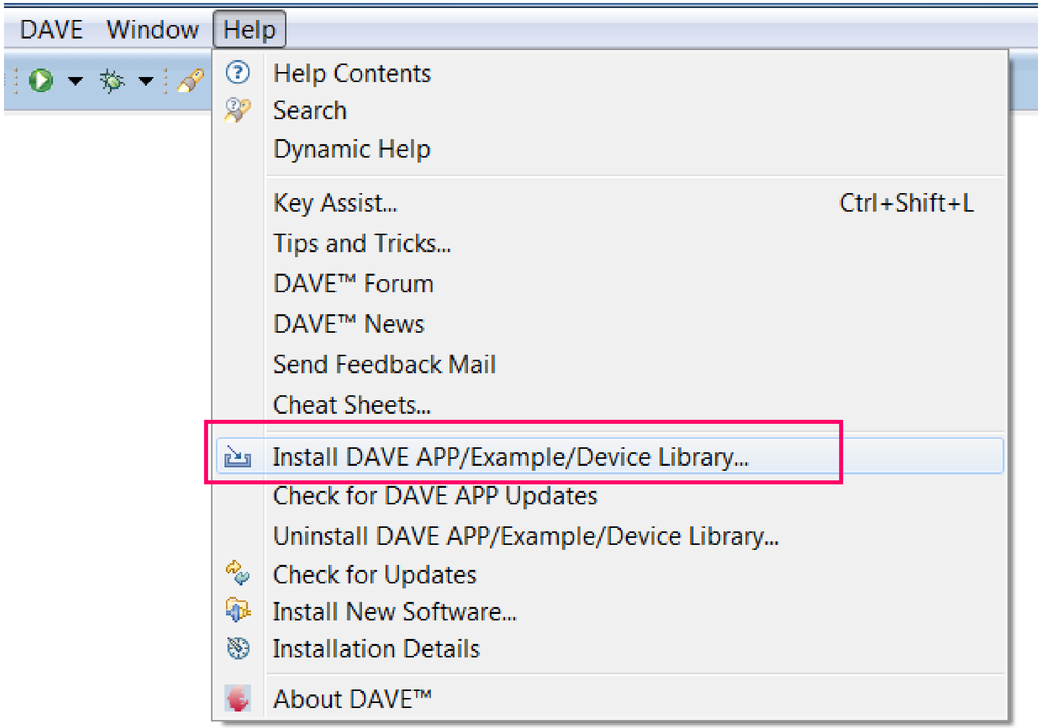
BMI Selection: Select

Info:
- Ensure you have installed latest SEGGER™ driver.
- This BMI feature is only applicable to the XMC1000 family and only works with J-Link XMC4200 OBD

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

1. Open DAVE™

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

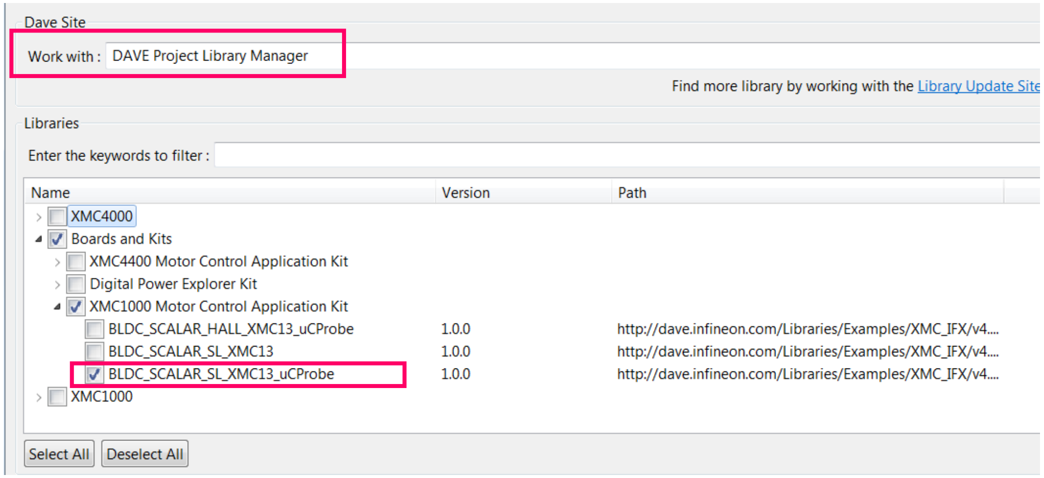


DAVE Window Help

Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
DAVE™ Forum
DAVE™ News
Send Feedback Mail
Cheat Sheets...
Install DAVE APP/Example/Device Library...
Check for DAVE APP Updates
Uninstall DAVE APP/Example/Device Library...
Check for Updates
Install New Software...
Installation Details
About DAVE™

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”



Dave Site

Work with : DAVE Project Library Manager

Find more library by working with the Library Update Site

Libraries

Enter the keywords to filter :

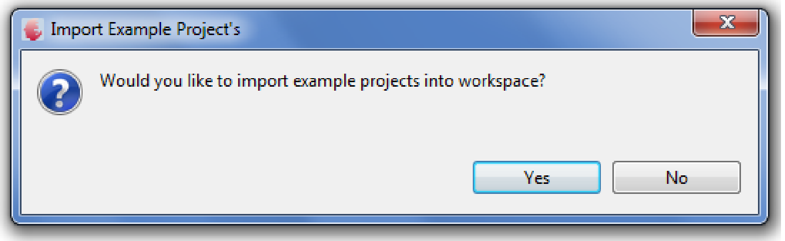
Name	Version	Path
› □ XMC4000		
◢ ☑ Boards and Kits		
› □ XMC4400 Motor Control Application Kit		
› □ Digital Power Explorer Kit		
◢ ☑ XMC1000 Motor Control Application Kit		
□ BLDC_SCALAR_HALL_XMC13_uCProbe	1.0.0	http://dave.infineon.com/Libraries/Examples/XMC_IFX/v4....
□ BLDC_SCALAR_SL_XMC13	1.0.0	http://dave.infineon.com/Libraries/Examples/XMC_IFX/v4....
☑ BLDC_SCALAR_SL_XMC13_uCProbe	1.0.0	http://dave.infineon.com/Libraries/Examples/XMC_IFX/v4....
› □ XMC1000		

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



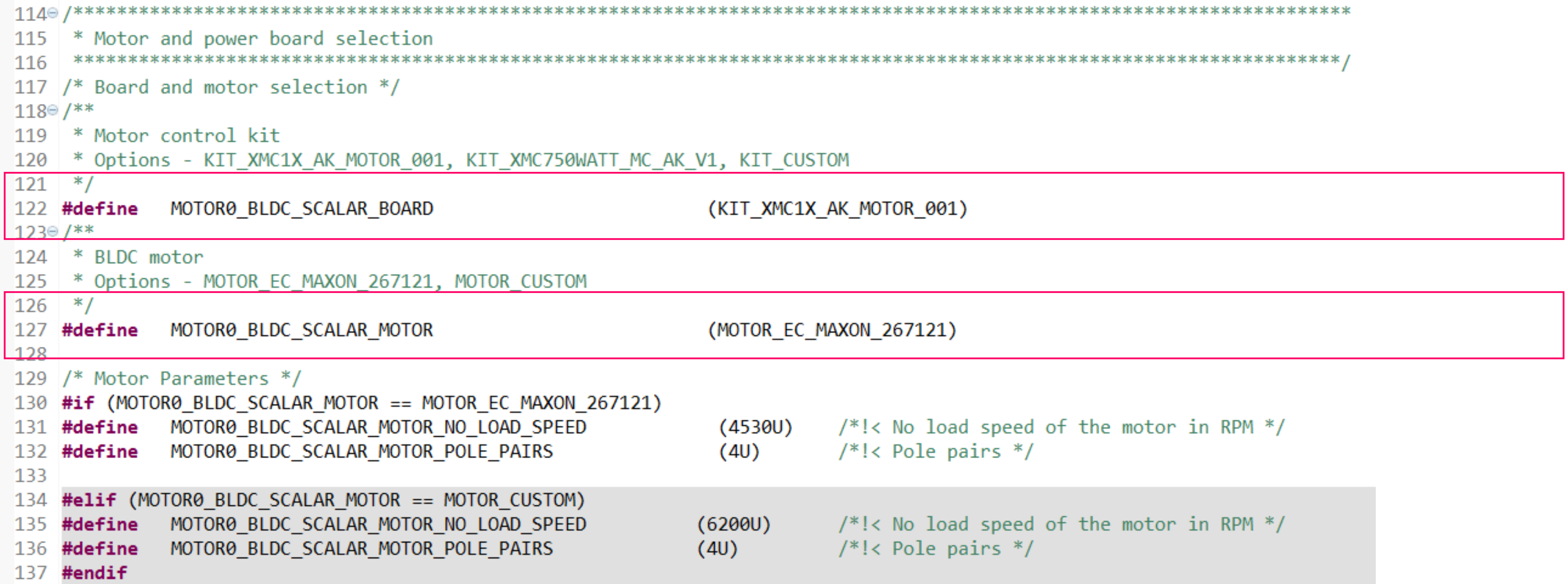
Import Example Project's

Would you like to import example projects into 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

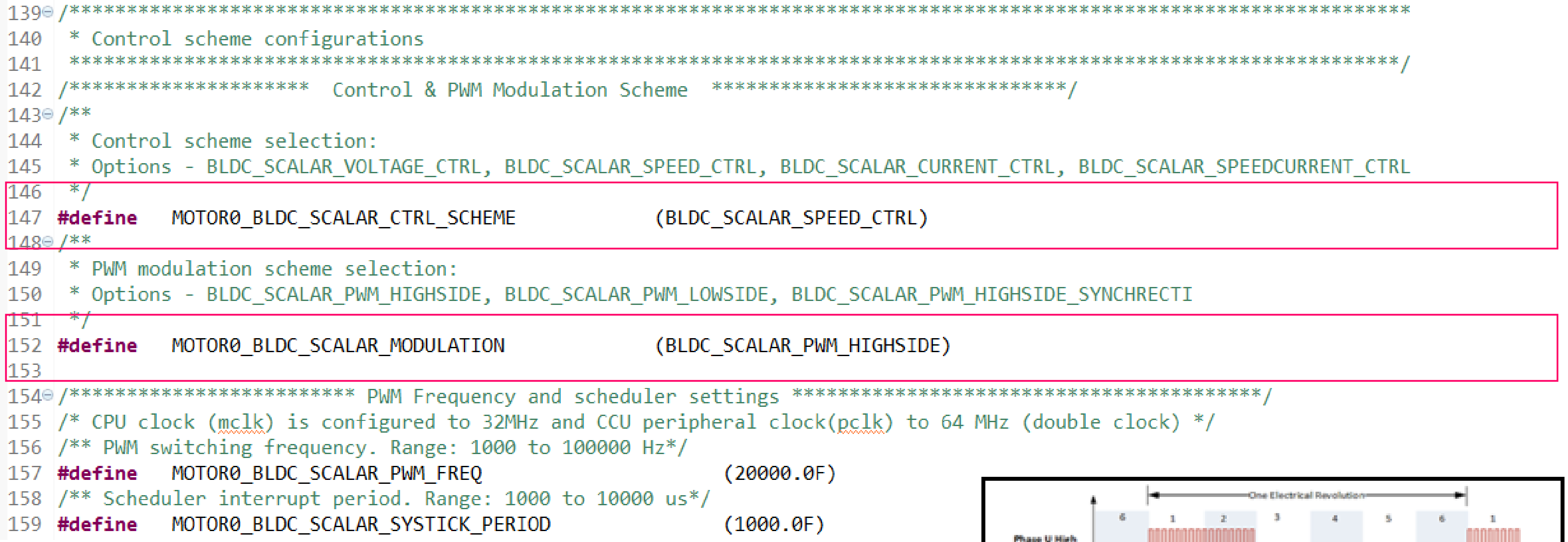


```
/*********************************************************************************************************************
 * Motor and power board selection
 ********************************************************************************************************************/
/* Board and motor selection */
/**
 * 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
```

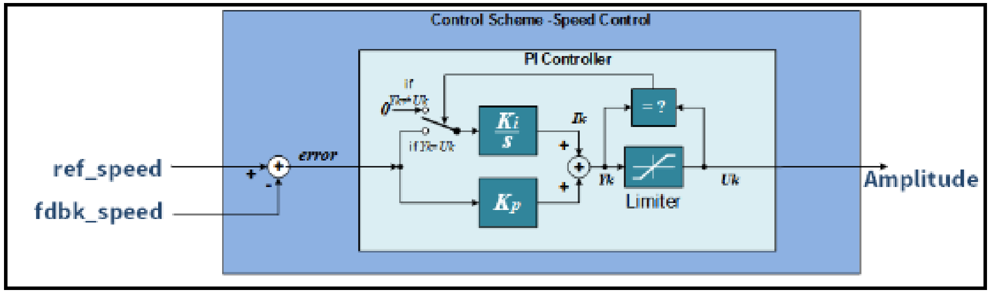
Getting Started – Configure the Project [2/6]

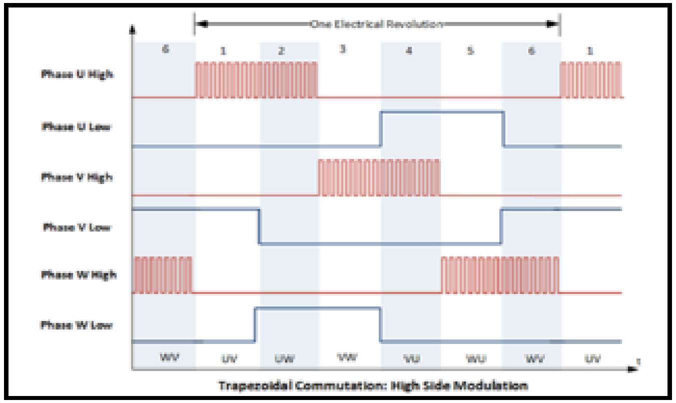
Folder: Configuration
File name: bldc_scalar_user_config.h

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)
```



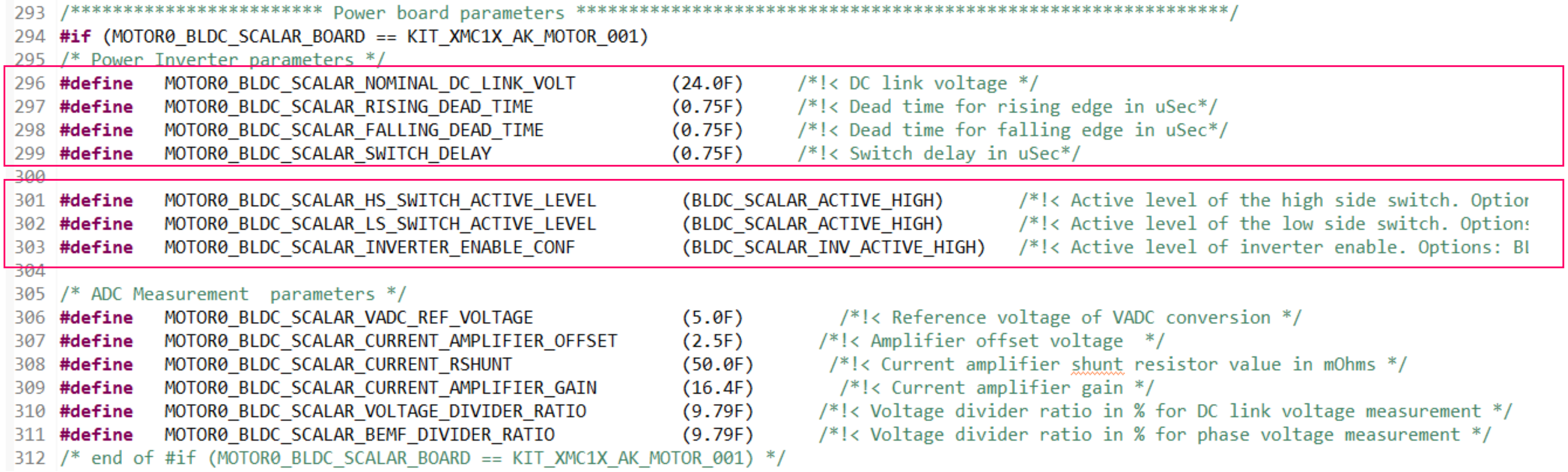


High side PWM modulation

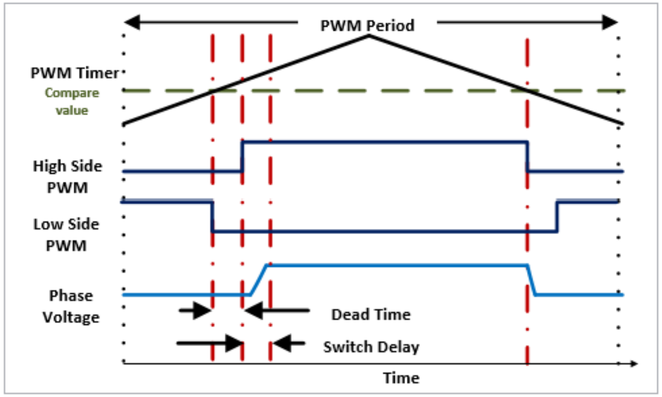
Getting Started – Configure the Project [3/6]

3. Configure the Power Board

Folder: Configuration
File name: bldc_scalar_user_config.h



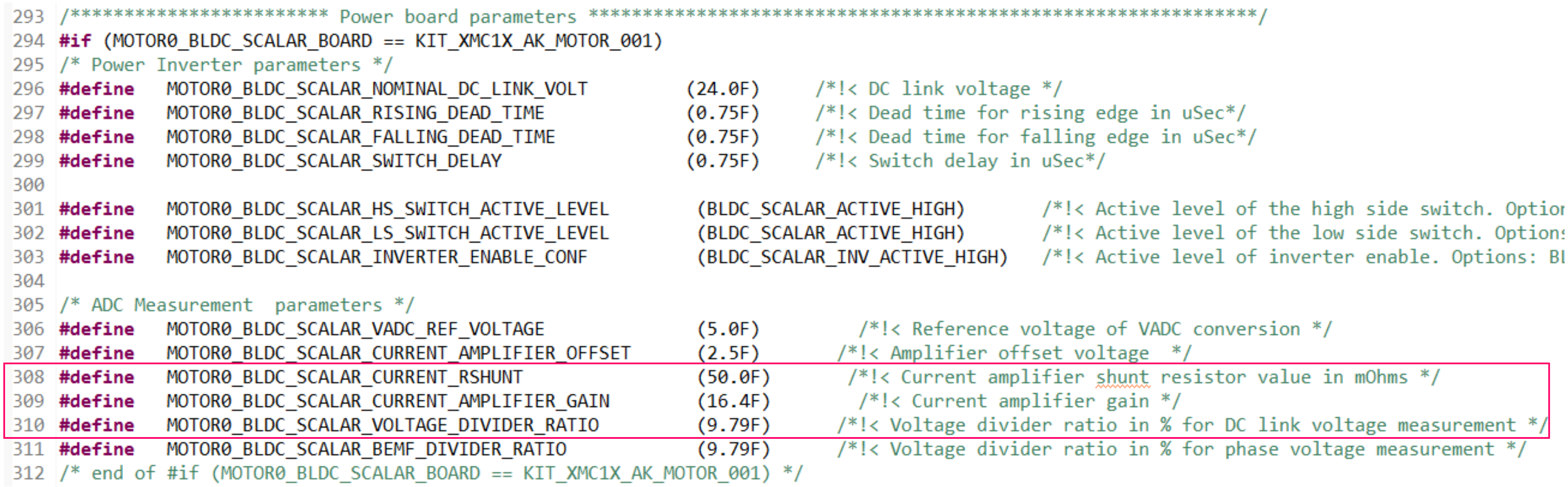
```
/************************ Power board parameters ***************************************************************/
#if (MOTOR0_BLDC_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
/* Power Inverter parameters */
#define    MOTOR0_BLDC_SCALAR_NOMINAL_DC_LINK_VOLT         (24.0F)     /*!< DC link voltage */
#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. Option
#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: Bl

/* 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) */
```



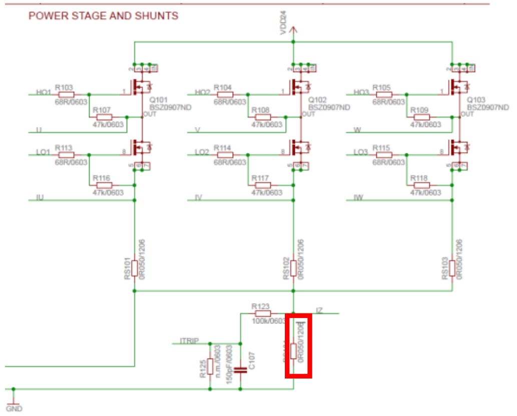
Getting Started – Configure the Project [4/6]

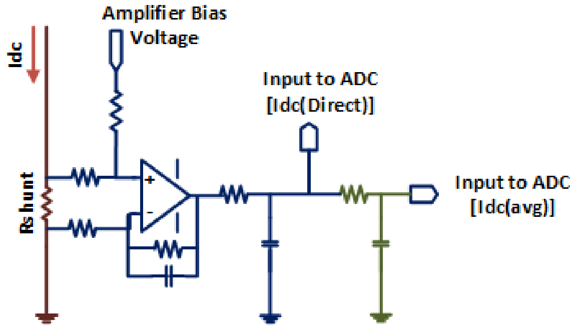
4. Configure the Power Board

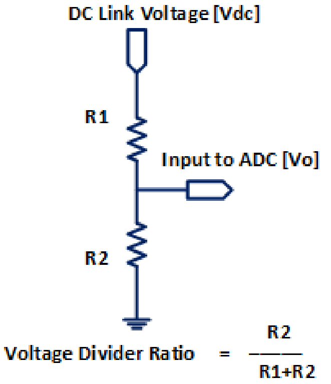
Folder: Configuration
File name: bldc_scalar_user_config.h



```
/************************ 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: Bl

/* 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) */
```





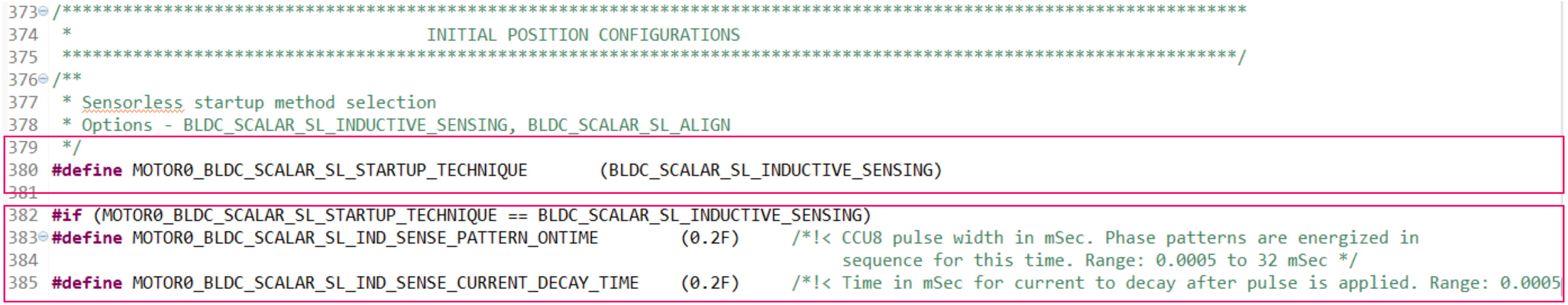


Voltage Divider Ratio $= \frac{R2}{R1+R2}$

Getting Started –
Configure the Project [5/6]

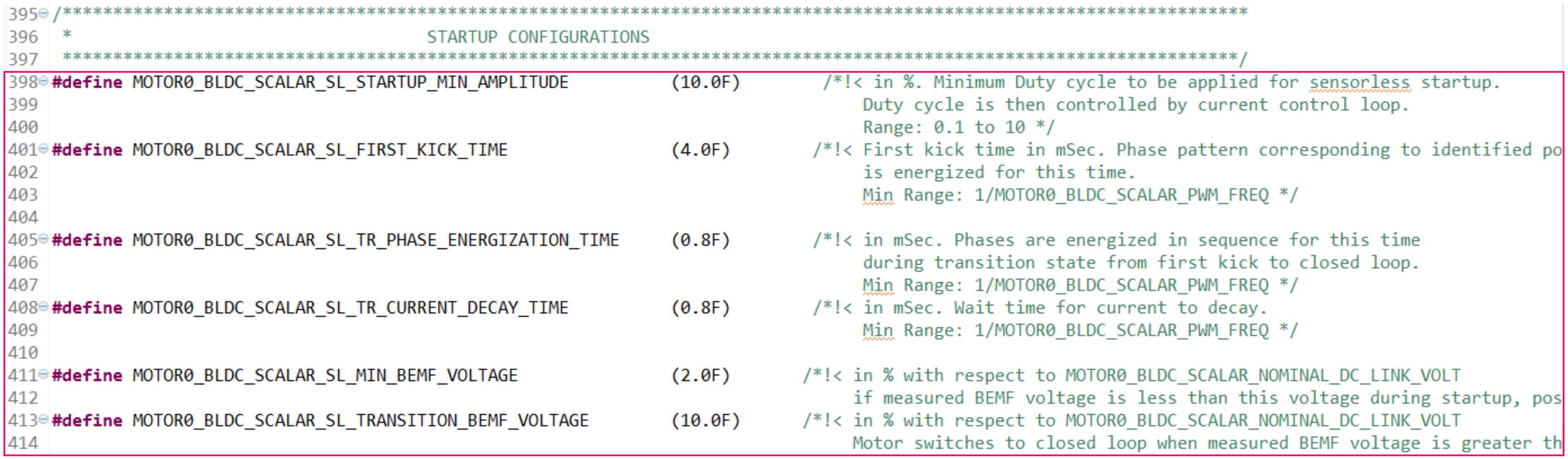
Folder: Configuration
File name: bldc_scalar_user_config.h

5. Configure the startup method



```
/*****************************************************************************************
 *                                     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)



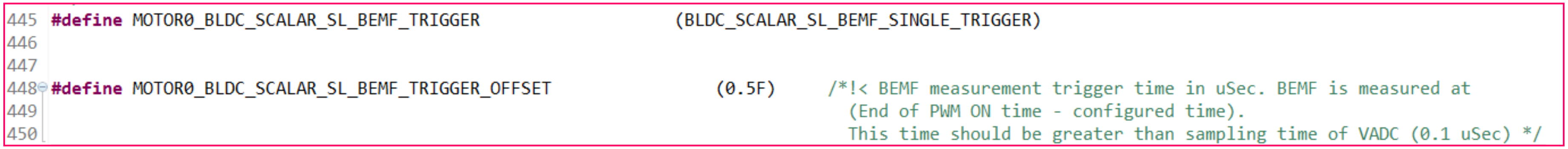
```
/*****************************************************************************************
 *                                     STARTUP CONFIGURATIONS
 *****************************************************************************************/
#define MOTOR0_BLDC_SCALAR_SL_STARTUP_MIN_AMPLITUDE          (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]

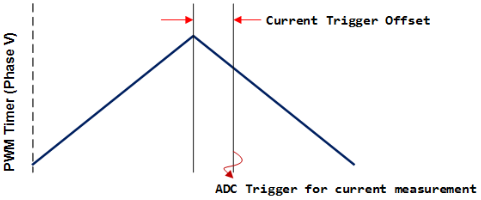
7. Configure the ADC Trigger

Folder: Configuration

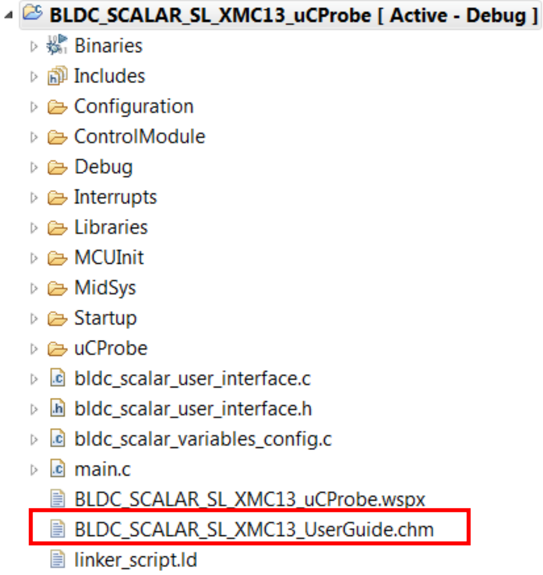
File name: bldc_scalar_user_config.h



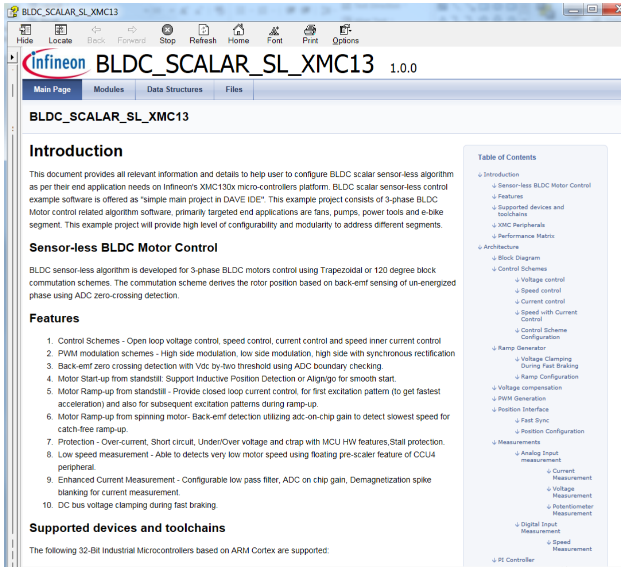
```
#define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER                      (BLDC_SCALAR_SL_BEMF_SINGLE_TRIGGER)


#define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER_OFFSET                   (0.5F)      /*!< BEMF measurement trigger time in uSec. BEMF is measured at
                                                                                  (End of PWM ON time - configured time).
                                                                                  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



- BLDC_SCALAR_SL_XMC13_uCProbe [Active - Debug]
  - Binaries
  - Includes
  - Configuration
  - ControlModule
  - Debug
  - Interrupts
  - Libraries
  - MCUInit
  - MidSys
  - Startup
  - uCProbe
  - bldc_scalar_user_interface.c
  - bldc_scalar_user_interface.h
  - bldc_scalar_variables_config.c
  - main.c
  - BLDC_SCALAR_SL_XMC13_uCProbe.wspx
  - BLDC_SCALAR_SL_XMC13_UserGuide.chm
  - linker_script.ld



BLDC_SCALAR_SL_XMC13

Introduction

This document provides all relevant information and details to help user to configure BLDC scalar sensor-less algorithm as per their end application needs on Infineon's XMC130x micro-controllers platform. BLDC scalar sensor-less control example software is offered as "simple main project in DAVE IDE". This example project consists of 3-phase BLDC Motor control related algorithm software, primarily 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.

Sensor-less BLDC Motor Control

BLDC sensor-less algorithm is developed for 3-phase BLDC motors control using Trapezoidal or 120 degree block commutation schemes. The commutation scheme derives the rotor position based on back-emf sensing of un-energized phase using ADC zero-crossing detection.

Features

1. Control Schemes - Open loop voltage control, speed control, current control and speed inner current control
2. PWM modulation schemes - High side modulation, low side modulation, high side with synchronous rectification
3. Back-emf zero crossing detection with Vdc by-two threshold using ADC boundary checking.
4. Motor Start-up from standstill: Support Inductive Position Detection or Align/go for smooth start.
5. Motor Ramp-up from standstill - Provide closed loop current control, for first excitation pattern (to get fastest acceleration) and also for subsequent excitation patterns during ramp-up.
6. Motor Ramp-up from spinning motor- Back-emf detection utilizing adc-on-chip gain to detect slowest speed for catch-free ramp-up.
7. Protection - Over-current, Short circuit, Under/Over voltage and ctrap with MCU HW features,Stall protection.
8. Low speed measurement - Able to detects very low motor speed using floating pre-scaler feature of CCU4 peripheral.
9. Enhanced Current Measurement - Configurable low pass filter, ADC on chip gain, Demagnetization spike blanking for current measurement.
10. DC bus voltage clamping during fast braking.

Supported devices and toolchains

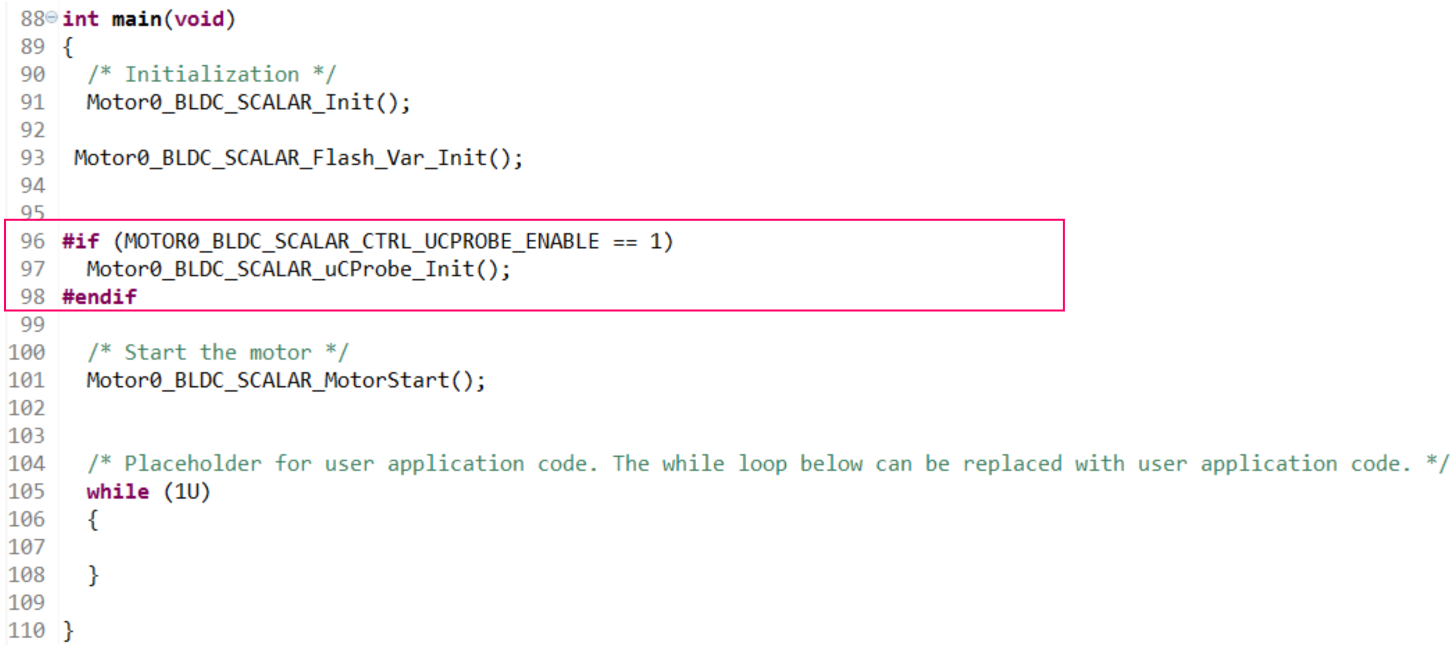
The following 32-Bit Industrial Microcontrollers based on ARM Cortex are supported:

Table of Contents

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

Folder: -
File name: main.c

1. Initialize the uCProbe before starting motor



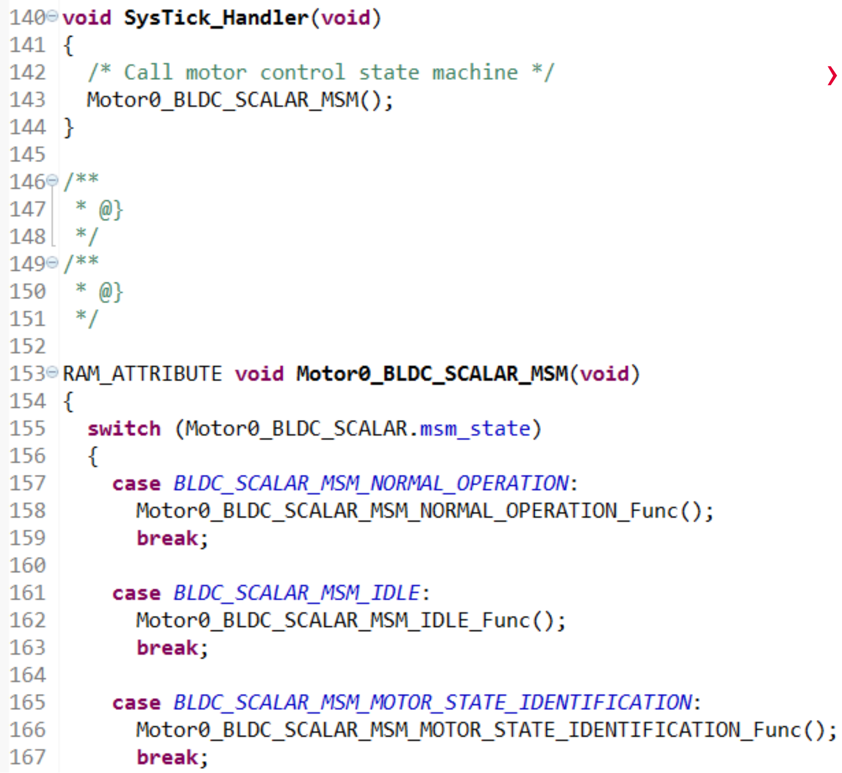
```
int main(void)
{
   /* Initialization */
   Motor0_BLDC_SCALAR_Init();

  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 replaced with user application code. */
   while (1U)
   {

   }

}
```

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

Folder: Interrupts

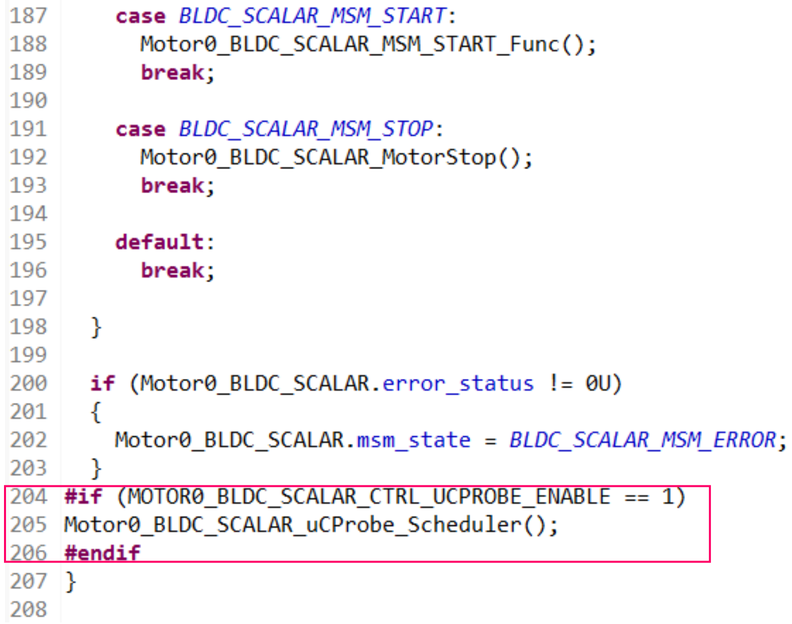
File name:
bldc_scalar_sl_state_machine.c

2. Added uCProbe scheduler in motor state machine



```
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;

    case BLDC_SCALAR_MSM_IDLE:
      Motor0_BLDC_SCALAR_MSM_IDLE_Func();
      break;

    case BLDC_SCALAR_MSM_MOTOR_STATE_IDENTIFICATION:
      Motor0_BLDC_SCALAR_MSM_MOTOR_STATE_IDENTIFICATION_Func();
      break;
```

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

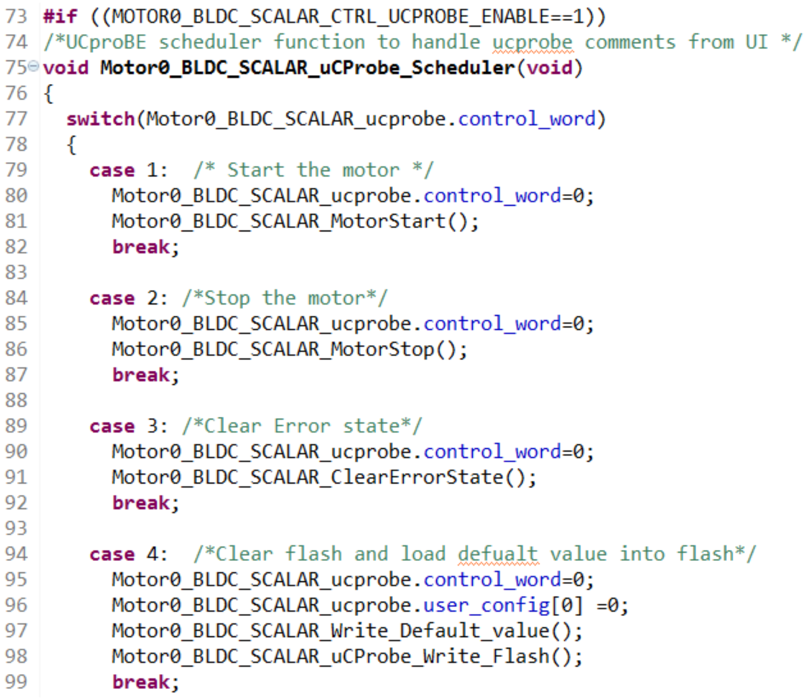


```
    case BLDC_SCALAR_MSM_START:
      Motor0_BLDC_SCALAR_MSM_START_Func();
      break;

    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 – Code added to support uCProbe [3/3]

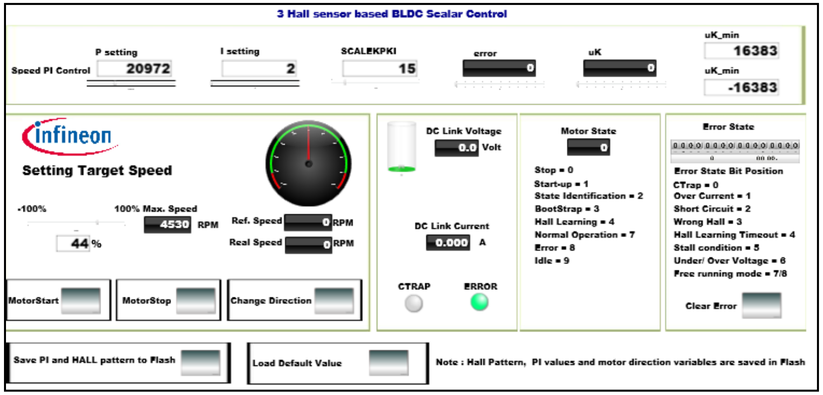
Folder: uCProbe
File name: ucProbe.c

3. Added 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



3 Hall sensor based BLDC Scalar Control

Speed PI Control
P setting: 20972
I setting: 2
SCALEKPKI: 15
error: 0
uK: 0
uK_min: 16383
uK_min: -16383

Setting Target Speed
-100%
100% Max. Speed: 4530 RPM
44 %
Ref. Speed: 0 RPM
Real Speed: 0 RPM

DC Link Voltage: 0.0 Volt
DC Link Current: 0.000 A

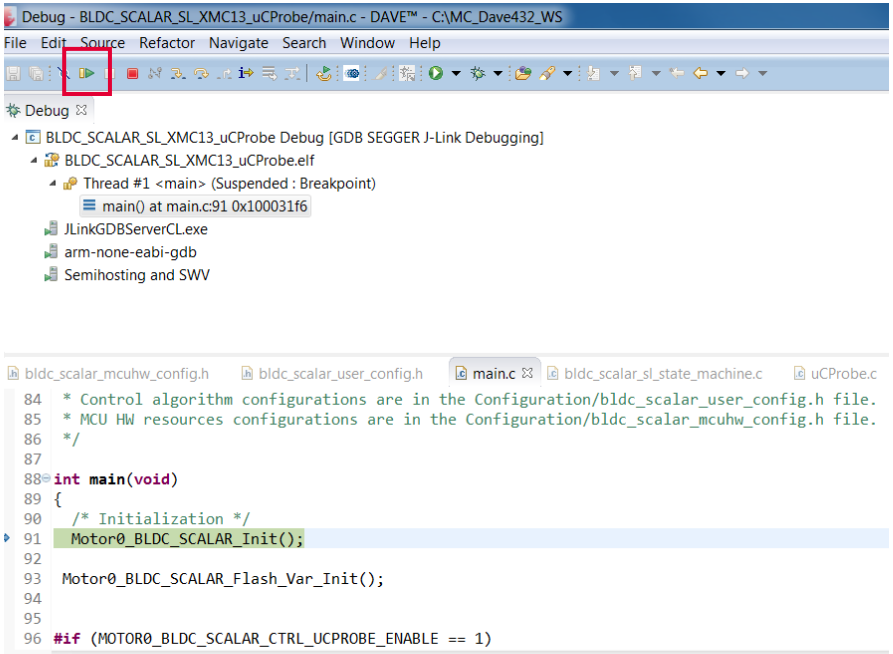
Motor State: 0
Stop = 0
Start-up = 1
State Identification = 2
BootStrap = 3
Hall Learning = 4
Normal Operation = 7
Error = 8
Idle = 9

Error State
Error State Bit Position
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

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

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


Debug - BLDC_SCALAR_SL_XMC13_uCProbe/main.c - DAVE™ - C:\MC_Dave432_WS
File Edit Source Refactor Navigate Search Window Help
Debug

- BLDC_SCALAR_SL_XMC13_uCProbe Debug [GDB SEGGER J-Link Debugging]
  - BLDC_SCALAR_SL_XMC13_uCProbe.elf
    - Thread #1 <main> (Suspended : Breakpoint)
      - main() at main.c:91 0x100031f6
  - JLinkGDBServerCL.exe
  - arm-none-eabi-gdb
  - Semihosting and SWV

bldc_scalar_mcuhw_config.h
bldc_scalar_user_config.h
main.c
bldc_scalar_sl_state_machine.c
uCProbe.c

```
84   * Control algorithm configurations are in the Configuration/bldc_scalar_user_config.h file.
85   * MCU HW resources configurations are in the Configuration/bldc_scalar_mcuhw_config.h file.
86   */
87
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)
```

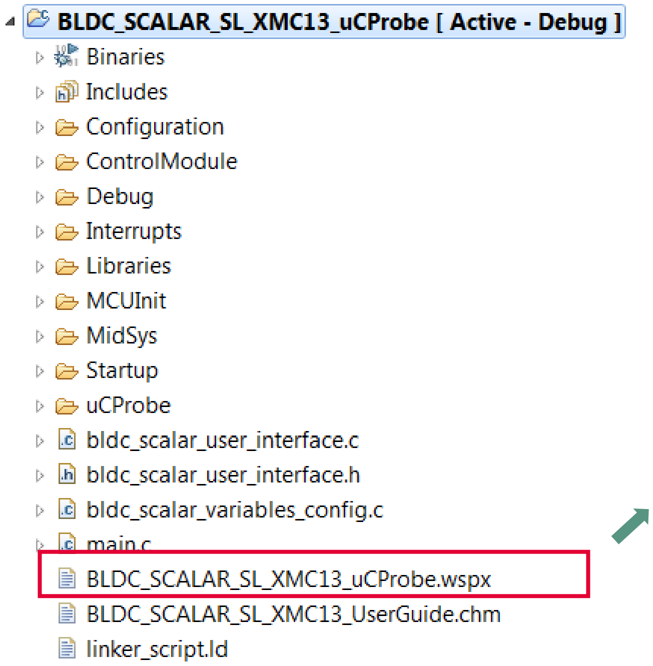
Observation:

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

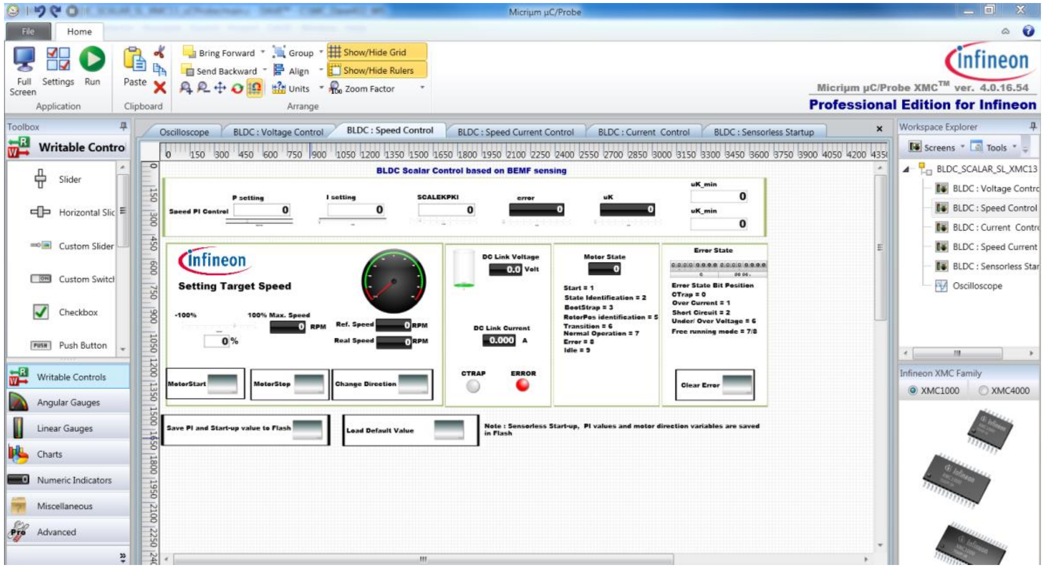
Getting Started – Interface with µC/Probe [1/6]

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

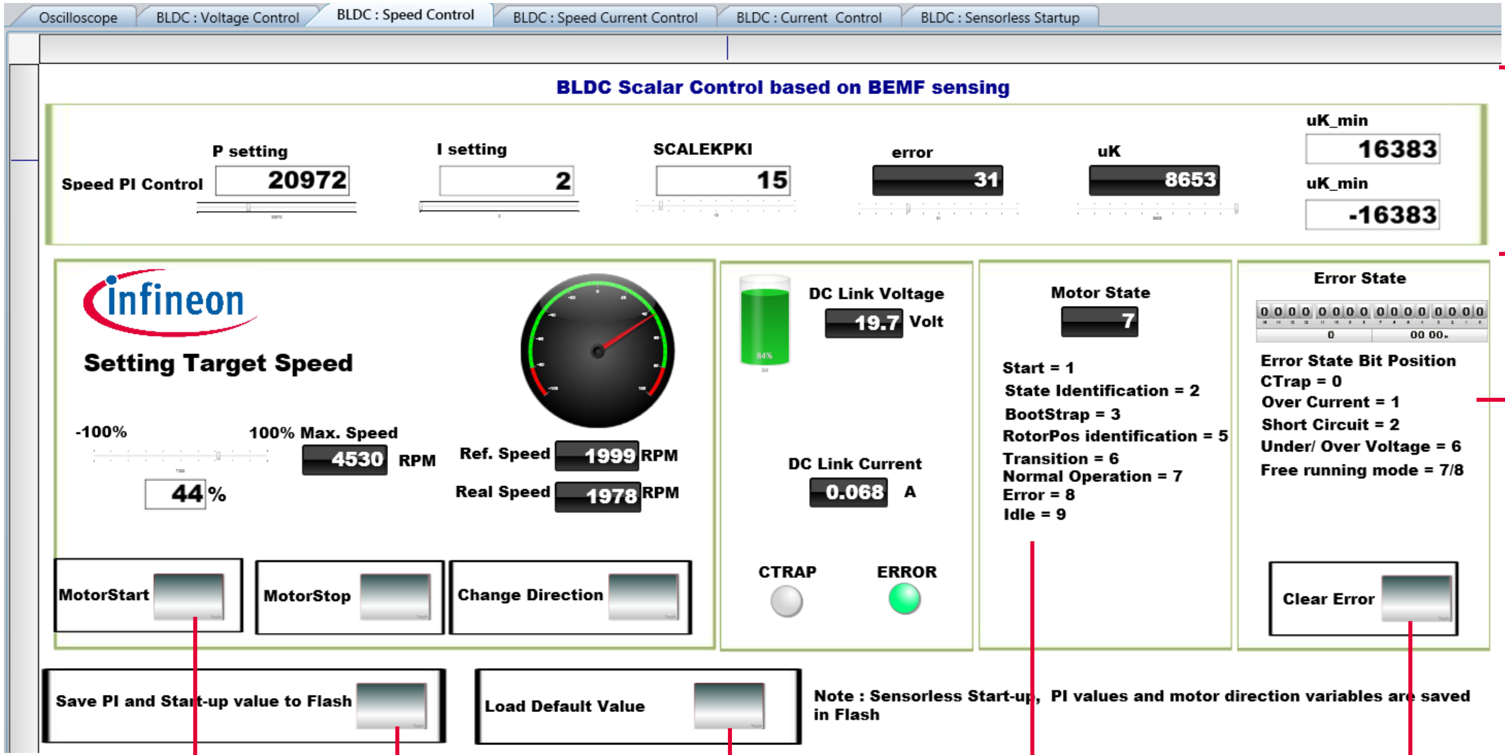
1. In “BLDC_SCALAR_SL_XMC13” example project , open µC/Probe™ XMC™ project file



- BLDC_SCALAR_SL_XMC13_uCProbe [Active - Debug]
  - Binaries
  - Includes
  - Configuration
  - ControlModule
  - Debug
  - Interrupts
  - Libraries
  - MCUInit
  - MidSys
  - Startup
  - uCProbe
  - bldc_scalar_user_interface.c
  - bldc_scalar_user_interface.h
  - bldc_scalar_variables_config.c
  - main.c
  - BLDC_SCALAR_SL_XMC13_uCProbe.wspx
  - BLDC_SCALAR_SL_XMC13_UserGuide.chm
  - linker_script.ld



Getting Started
Interface with µC/Probe [2/6]



Oscilloscope | BLDC : Voltage Control | BLDC : Speed Control | BLDC : Speed Current Control | BLDC : Current Control | BLDC : Sensorless Startup

BLDC Scalar Control based on BEMF sensing

Speed PI Control — P setting: 20972 — I setting: 2 — SCALEKPKI: 15 — error: 31 — uK: 8653 — uK_min: 16383 — uK_min: -16383

Setting Target Speed

-100% — 100% Max. Speed — 4530 RPM — 44 %

Ref. Speed 1999 RPM

Real Speed 1978 RPM

DC Link Voltage 19.7 Volt

DC Link Current 0.068 A

CTRAP — ERROR

Motor State: 7

Start = 1
State Identification = 2
BootStrap = 3
RotorPos identification = 5
Transition = 6
Normal Operation = 7
Error = 8
Idle = 9

Error State

Error State Bit Position
CTrap = 0
Over Current = 1
Short Circuit = 2
Under/ Over Voltage = 6
Free running mode = 7/8

Note : Sensorless Start-up, PI values and motor direction variables are saved in Flash

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

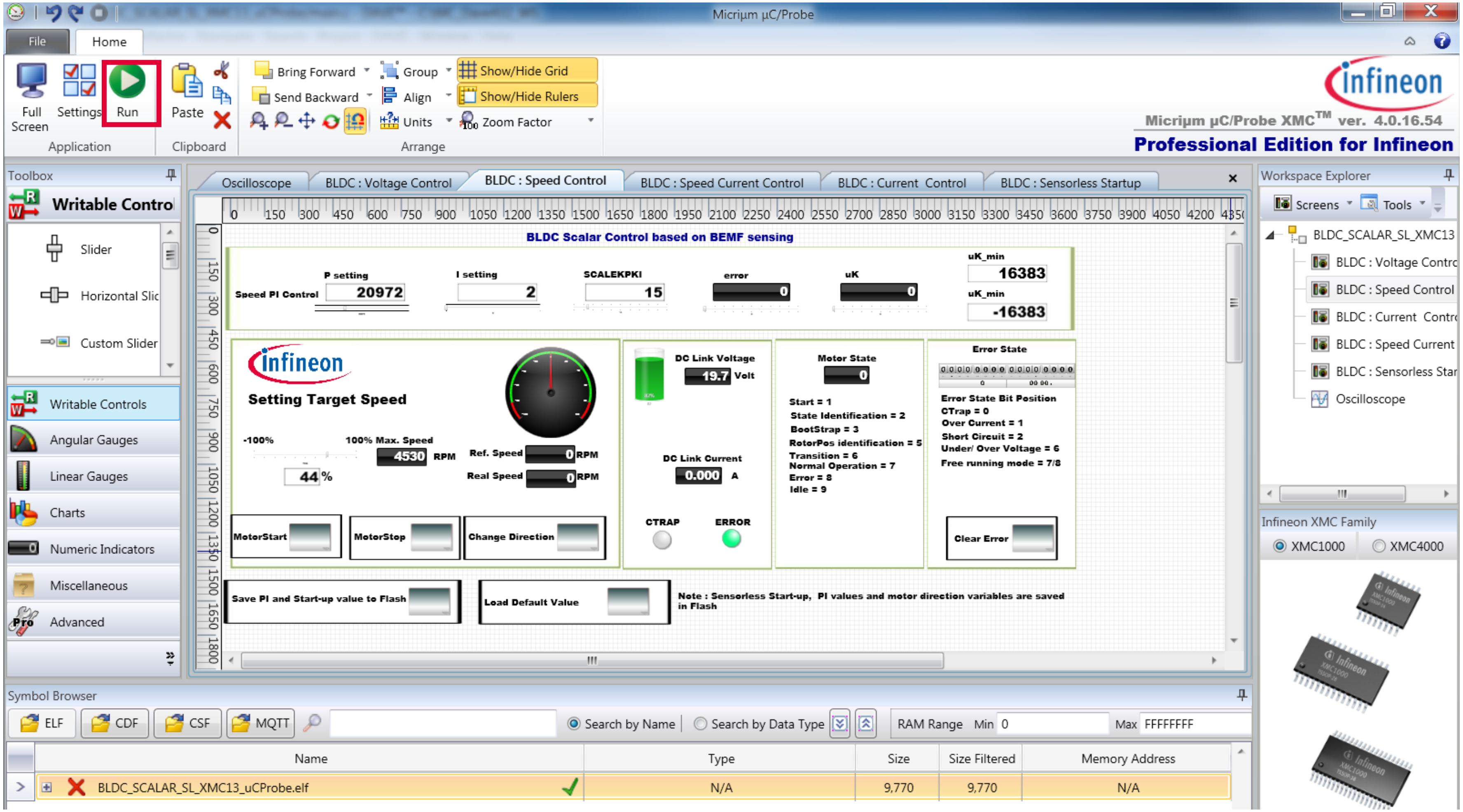
Load the default values into flash

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



P setting 20972 | I setting 2 | SCALEKPKI 15 | error 0 | uK 0 | uK_min 16383 | uK_min -16383

44 % | 4530 RPM | Ref. Speed 0 RPM | Real Speed 0 RPM

DC Link Voltage 19.7 Volt | DC Link Current 0.000 A | Motor State 0

Start = 1
State Identification = 2
BootStrap = 3
RotorPos identification = 5
Transition = 6
Normal Operation = 7
Error = 8
Idle = 9

Error State Bit Position
CTrap = 0
Over Current = 1
Short Circuit = 2
Under/ Over Voltage = 6
Free running mode = 7/8

Note : Sensorless Start-up, PI values and motor direction variables are saved in Flash

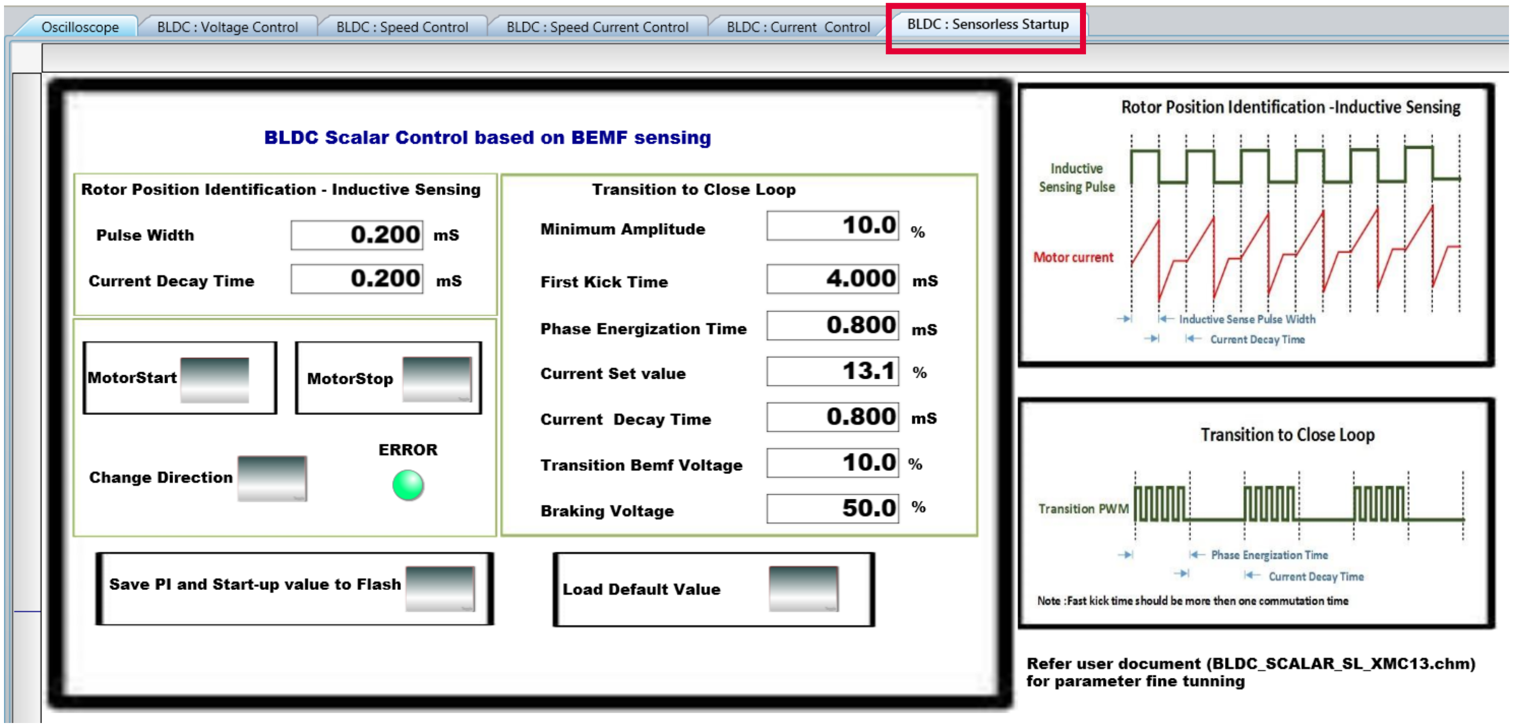
● XMC1000 ○ XMC4000

● Search by Name ○ Search by Data Type | RAM Range Min 0 Max FFFFFFFF

Name	Type	Size	Size Filtered	Memory Address
BLDC_SCALAR_SL_XMC13_uCProbe.elf ✓	N/A	9,770	9,770	N/A

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.



BLDC Scalar Control based on BEMF sensing

Rotor Position Identification - Inductive Sensing

Pulse Width	0.200	mS
Current Decay Time	0.200	mS

Transition to Close Loop

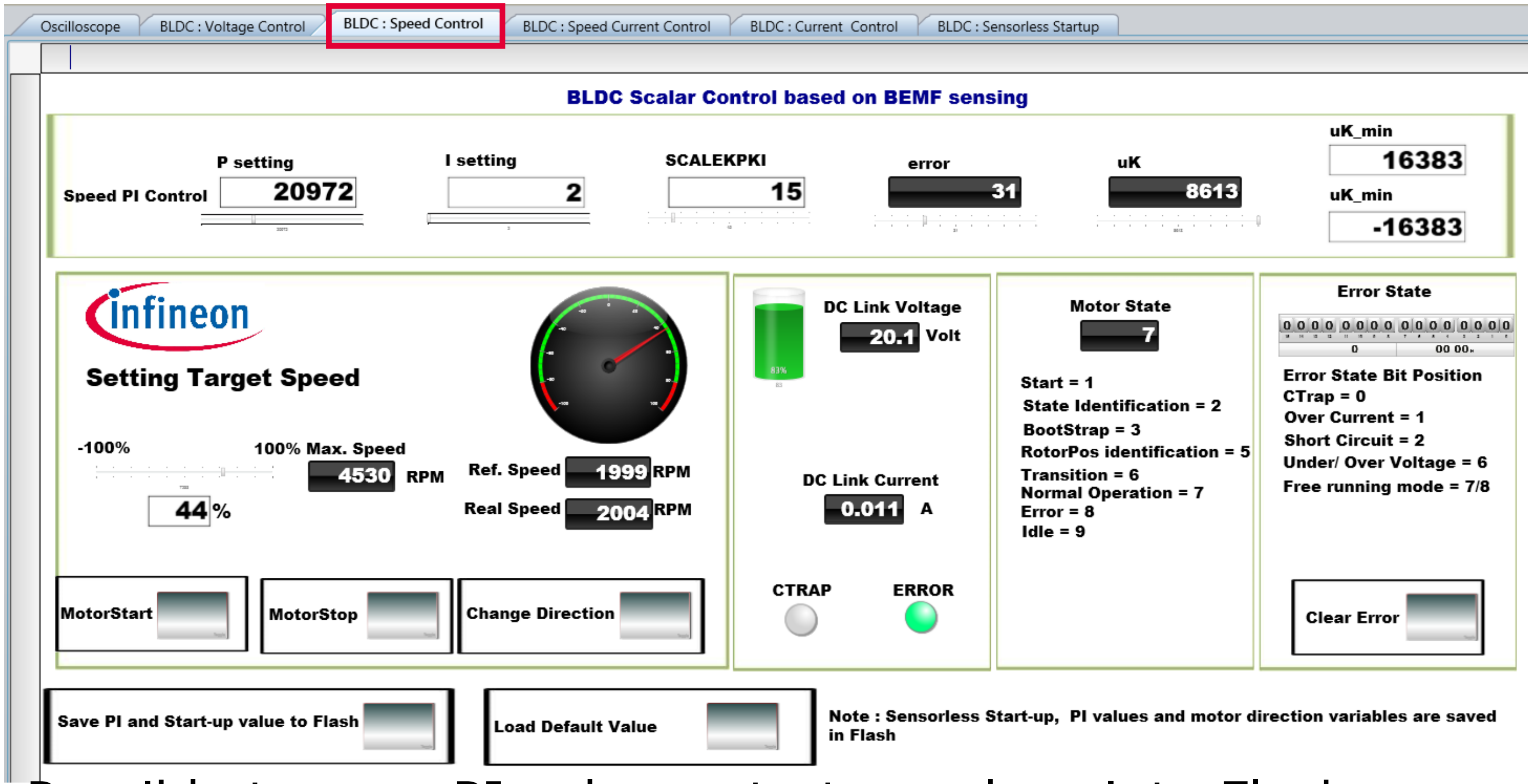
Minimum Amplitude	10.0	%
First Kick Time	4.000	mS
Phase Energization Time	0.800	mS
Current Set value	13.1	%
Current Decay Time	0.800	mS
Transition Bemf Voltage	10.0	%
Braking Voltage	50.0	%

Note :Fast kick time should be more then one commutation time

Refer user document (BLDC_SCALAR_SL_XMC13.chm) for parameter fine tunning

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



BLDC Scalar Control based on BEMF sensing

Speed PI Control
P setting 20972
I setting 2
SCALEKPKI 15
error 31
uK 8613
uK_min 16383
uK_min -16383

Setting Target Speed
-100% 100% Max. Speed
4530 RPM
44 %
Ref. Speed 1999 RPM
Real Speed 2004 RPM

DC Link Voltage 20.1 Volt
DC Link Current 0.011 A

Motor State 7
Start = 1
State Identification = 2
BootStrap = 3
RotorPos identification = 5
Transition = 6
Normal Operation = 7
Error = 8
Idle = 9

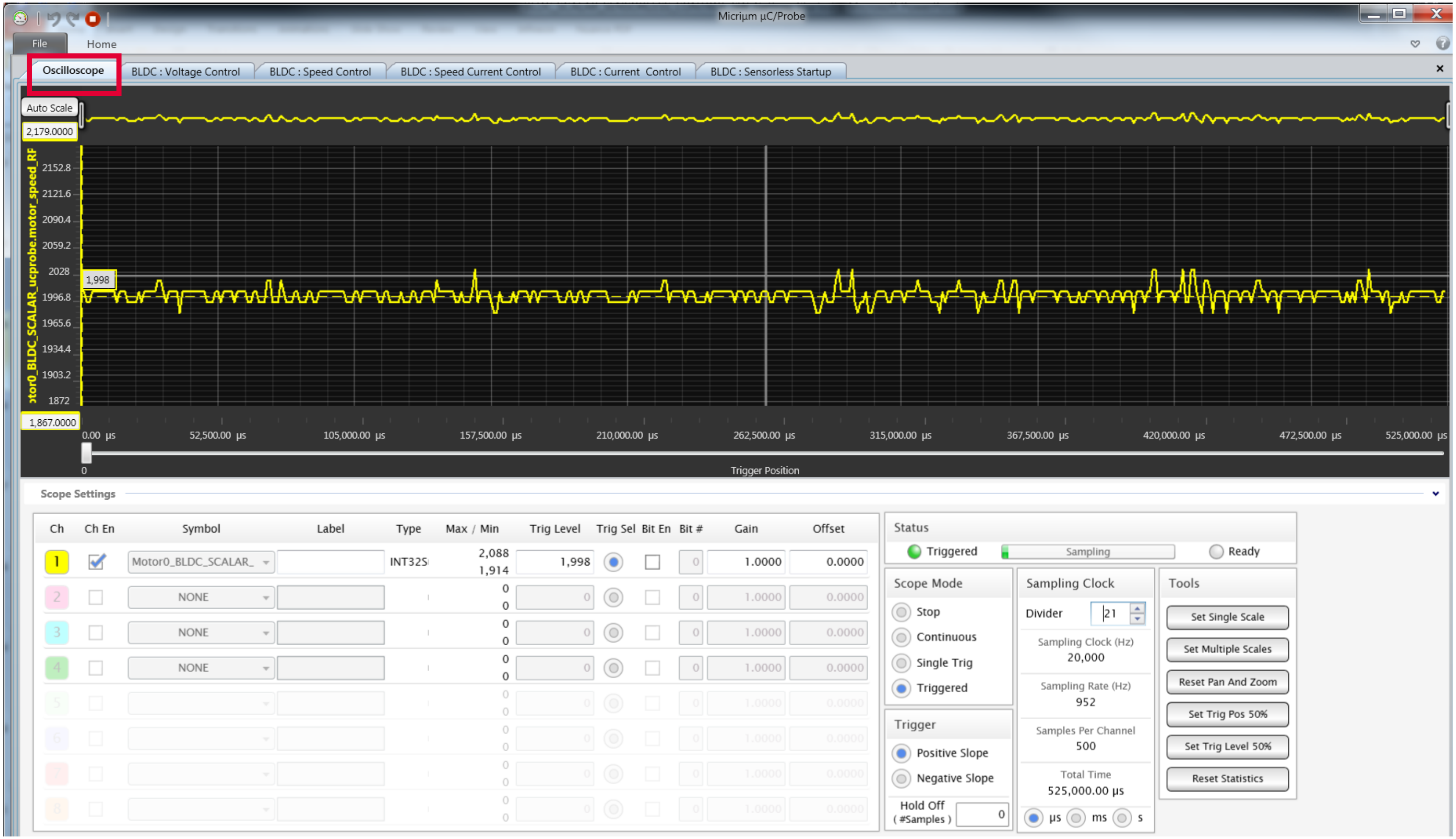
Error State
Error State Bit Position
CTrap = 0
Over Current = 1
Short Circuit = 2
Under/ Over Voltage = 6
Free running mode = 7/8

Note : Sensorless Start-up, PI values and motor direction variables are saved in Flash

› 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



Scope Settings

Ch	Ch En	Symbol	Label	Type	Max / Min	Trig Level	Trig Sel	Bit En	Bit #	Gain	Offset
1	☑	Motor0_BLDC_SCALAR_		INT32S	2,088 1,914	1,998	●	□	0	1.0000	0.0000
2	□	NONE			0 0	0	○	□	0	1.0000	0.0000
3	□	NONE			0 0	0	○	□	0	1.0000	0.0000
4	□	NONE			0 0	0	○	□	0	1.0000	0.0000
5	□				0 0	0	○	□	0	1.0000	0.0000
6	□				0 0	0	○	□	0	1.0000	0.0000
7	□				0 0	0	○	□	0	1.0000	0.0000
8	□				0 0	0	○	□	0	1.0000	0.0000

Status

● Triggered | Sampling | ○ Ready

Scope Mode

○ Stop
○ Continuous
○ Single Trig
● Triggered

Trigger

● Positive Slope
○ Negative Slope

Hold Off (#Samples) 0

Sampling Clock

Divider 21

Sampling Clock (Hz) 20,000

Sampling Rate (Hz) 952

Samples Per Channel 500

Total Time 525,000.00 μs

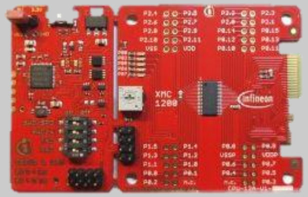
● μs ○ ms ○ s

Tools

Set Single Scale
Set Multiple Scales
Reset Pan And Zoom
Set Trig Pos 50%
Set Trig Level 50%
Reset Statistics

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^{TM} development platform:

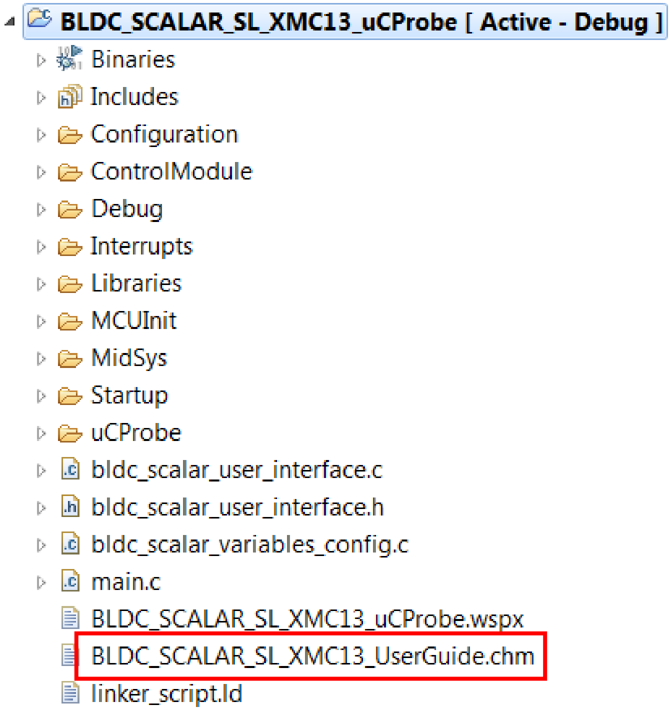
  http://www.infineon.com/DAVE

- For support:

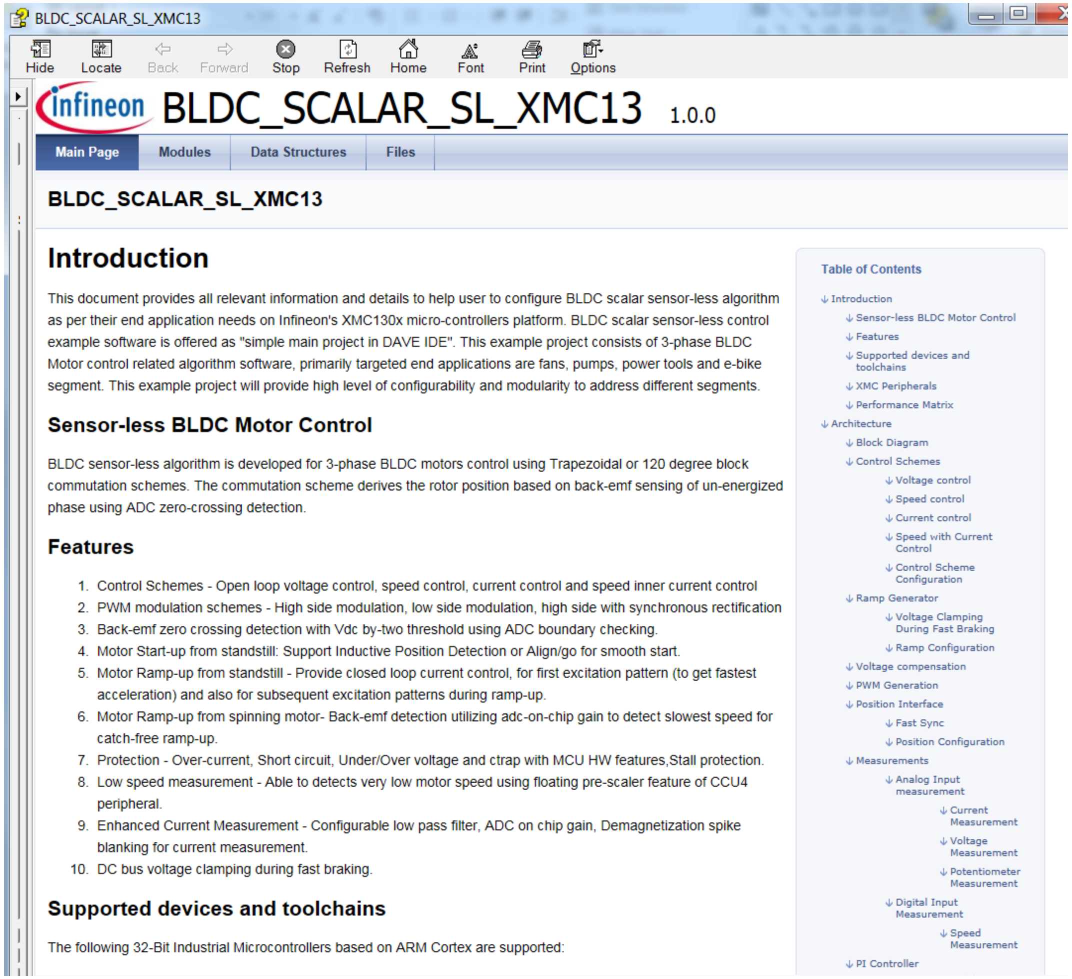
  http://www.infineonforums.com/forums/8-XMC-Forum

References : Help Content

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



BLDC_SCALAR_SL_XMC13_uCProbe [Active - Debug]
- Binaries
- Includes
- Configuration
- ControlModule
- Debug
- Interrupts
- Libraries
- MCUInit
- MidSys
- Startup
- uCProbe
- bldc_scalar_user_interface.c
- bldc_scalar_user_interface.h
- bldc_scalar_variables_config.c
- main.c
- BLDC_SCALAR_SL_XMC13_uCProbe.wspx
- BLDC_SCALAR_SL_XMC13_UserGuide.chm
- linker_script.ld



BLDC_SCALAR_SL_XMC13 1.0.0

BLDC_SCALAR_SL_XMC13

Introduction

This document provides all relevant information and details to help user to configure BLDC scalar sensor-less algorithm as per their end application needs on Infineon's XMC130x micro-controllers platform. BLDC scalar sensor-less control example software is offered as "simple main project in DAVE IDE". This example project consists of 3-phase BLDC Motor control related algorithm software, primarily 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.

Sensor-less BLDC Motor Control

BLDC sensor-less algorithm is developed for 3-phase BLDC motors control using Trapezoidal or 120 degree block commutation schemes. The commutation scheme derives the rotor position based on back-emf sensing of un-energized phase using ADC zero-crossing detection.

Features

1. Control Schemes - Open loop voltage control, speed control, current control and speed inner current control
2. PWM modulation schemes - High side modulation, low side modulation, high side with synchronous rectification
3. Back-emf zero crossing detection with Vdc by-two threshold using ADC boundary checking.
4. Motor Start-up from standstill: Support Inductive Position Detection or Align/go for smooth start.
5. Motor Ramp-up from standstill - Provide closed loop current control, for first excitation pattern (to get fastest acceleration) and also for subsequent excitation patterns during ramp-up.
6. Motor Ramp-up from spinning motor- Back-emf detection utilizing adc-on-chip gain to detect slowest speed for catch-free ramp-up.
7. Protection - Over-current, Short circuit, Under/Over voltage and ctrap with MCU HW features,Stall protection.
8. Low speed measurement - Able to detects very low motor speed using floating pre-scaler feature of CCU4 peripheral.
9. Enhanced Current Measurement - Configurable low pass filter, ADC on chip gain, Demagnetization spike blanking for current measurement.
10. DC bus voltage clamping during fast braking.

Supported devices and toolchains

The following 32-Bit Industrial Microcontrollers based on ARM Cortex are supported:

Table of Contents

Glossary Abbreviations

ADC	Analog Digital Converter
DAVE™	Digital Application Virtual Engineer (Free development IDE for XMC$^{\text{TM}}$)
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.