XMC13 Sensorless BLDC Scalar Control Software
Getting Started
XMC™ Microcontrollers
Oct 2016
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

<table>
<thead>
<tr>
<th>Software Blocks</th>
<th>Supported Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control Scheme</strong></td>
<td>Open loop voltage control, speed control, current control and speed inner current control</td>
</tr>
<tr>
<td><strong>PWM Modulation (Modulator)</strong></td>
<td>High side modulation, low side modulation, high side with synchronous rectification</td>
</tr>
<tr>
<td><strong>Current/Voltage Measurement</strong></td>
<td>Direct DC link and average current measurement, DC link Voltage &amp; Potentiometer (Analog Input)</td>
</tr>
</tbody>
</table>
## Software Overview – Key Features

<table>
<thead>
<tr>
<th>Supported Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-directional control</td>
<td>Reverse the motor direction when change direction variable is set</td>
</tr>
<tr>
<td>On fly start-up</td>
<td>Catch spinning motor at start-up without stop</td>
</tr>
<tr>
<td>Back-emf zero crossing detection</td>
<td>Back-emf zero crossing detection using ADC boundary checking</td>
</tr>
<tr>
<td>Motor Start-up from standstill</td>
<td>Support inductive position detection or alignment for smooth start</td>
</tr>
<tr>
<td>Accurate measurement of speed</td>
<td>Use floating pre-scalar</td>
</tr>
<tr>
<td>(across wide range)</td>
<td></td>
</tr>
<tr>
<td>Demagnetization blanking</td>
<td>Remove spike in direct DC link current measurement</td>
</tr>
<tr>
<td>DC bus voltage clamping</td>
<td>Prevent over-voltage during fast braking</td>
</tr>
<tr>
<td>Protection</td>
<td>Stall Detection</td>
</tr>
<tr>
<td></td>
<td>Over-current</td>
</tr>
<tr>
<td></td>
<td>Short circuit</td>
</tr>
<tr>
<td></td>
<td>Under/Over voltage</td>
</tr>
<tr>
<td></td>
<td>C-trap with MCU hardware features</td>
</tr>
</tbody>
</table>
Software Overview - Files Structure

Folder/File Structure

Main:
- bldc_scalar_user_interface.c
- bldc_scalar_variables_config.c
- bldc_scalar_variables_config.h

Interrupts:
- bldc_scalar_control_loop.c
- bldc_scalar_state_machine.c
- bldc_scalar_filter_update.c
- bldc_scalar_hall_event.c
- bldc_scalar_trap.c
- bldc_scalar_protection.c

Configuration:
- bldc_scalar_MCUCURconfig.h
- bldc_scalar_user_parameter.h
- bldc_scalar_derived_parameter.c

Control Module:
- bldc_scalar_control_scheme.c
- bldc_scalar_control_scheme.h
- bldc_scalar_pll.c
- bldc_scalar_pll.h
- bldc_scalar_pll_filter.c
- bldc_scalar_pll_filter.h
- bldc_scalar_ramp_generator.c
- bldc_scalar_ramp_generator.h
- bldc_scalar_hall.c
- bldc_scalar_hall.h

MidSys:
- bldc_scalar_pwm_bc.c
- bldc_scalar_pos_spd_hall.c
- bldc_scalar_soft_dibus.c
- bldc_scalar_soft_potentiometer.c
- bldc_scalar_soft_phase.c
- bldc_scalar_current Motor.c

MCUInit:
- ccu4.c
- ccu8.c
- posif.c
- nvic.c
- gpio.c
## Software Overview - XMC Peripheral usage

<table>
<thead>
<tr>
<th>No</th>
<th>Resource</th>
<th>Resource usage</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CCU40_CC40</td>
<td>Always</td>
<td>Commutation timer (Pre-scalar value calculated in zero crossing event)</td>
</tr>
<tr>
<td>2</td>
<td>CCU40_CC41</td>
<td>Fast Sync is disabled</td>
<td>Multi-channel Pattern synchronization</td>
</tr>
<tr>
<td>3</td>
<td>CCU40_CC42</td>
<td>Always</td>
<td>Used for motor speed capture</td>
</tr>
<tr>
<td>4</td>
<td>POSIF0</td>
<td>Always</td>
<td>MCM configuration</td>
</tr>
<tr>
<td>5</td>
<td>CCU80_CC8x</td>
<td>Always</td>
<td>PWM Generation – Phase U</td>
</tr>
<tr>
<td>6</td>
<td>CCU80_CC8y</td>
<td>Always</td>
<td>PWM Generation – Phase V</td>
</tr>
<tr>
<td>7</td>
<td>CCU80_CC8z</td>
<td>Always</td>
<td>PWM Generation – Phase W</td>
</tr>
<tr>
<td>8</td>
<td>VADC Group A Queue A</td>
<td>Any ADC measurement is enabled</td>
<td>DC link direct/ Average current , DC link voltage, user defined and potentiometer measurement</td>
</tr>
<tr>
<td>9</td>
<td>NVIC</td>
<td>Always</td>
<td>Used for ISRs</td>
</tr>
<tr>
<td>10</td>
<td>SYSTICK</td>
<td>Always</td>
<td>Used for state machine</td>
</tr>
</tbody>
</table>

Note: x,y,z, A – Resource number based on configuration
Software Overview - Peripheral Interconnection

![Diagram of BLDC Commutation in Sensorless mode]

- **POSIF**: Multi Channel Control
  - MSYNCA
  - MOUT[3:0]
  - MOUT[7:4]
  - MOUT[11:8]

- **CCU4**: Capture and clear
  - CC42 Time between ticks
  - IN0
  - IN1
  - IN2

- **ADC0**: Zero Crossing Detection
  - G0CH5
  - G0CH6
  - G0CH7

- **CCU8**: Phase U, Phase V, Phase W
  - MCIO
  - MC1
  - MCIO
  - MC1
  - MCIO
  - MC1

- **Back EMF**: Phase A, Phase B, Phase C

- **High side PWM modulation**

---

Copyright © Infineon Technologies AG 2016. All rights reserved.
<table>
<thead>
<tr>
<th>Peripheral</th>
<th>Interrupt Subroutines (ISR)</th>
<th>NVIC node</th>
<th>Interval</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>VADC</td>
<td>Channel event (current + voltage outside boundary)</td>
<td>15</td>
<td>Asynchronous</td>
<td>0</td>
</tr>
<tr>
<td>CCU8</td>
<td>One match event (phase U)</td>
<td>25</td>
<td>1/ PWM frequency</td>
<td>2</td>
</tr>
<tr>
<td>CCU8</td>
<td>CTRAP (phase U)</td>
<td>26</td>
<td>Asynchronous</td>
<td>0</td>
</tr>
<tr>
<td>SYSTIMER</td>
<td>Systick Scheduler</td>
<td>-1</td>
<td>1 mSec (configurable)</td>
<td>3</td>
</tr>
<tr>
<td>VADC</td>
<td>Zero-crossing</td>
<td>16</td>
<td>Variable (motor speed dependent)</td>
<td>1</td>
</tr>
</tbody>
</table>
# Software Overview – Example Configuration

<table>
<thead>
<tr>
<th>Example Name</th>
<th>BLDC_SCALAR_SL_XMC13_uCProbe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kit Description</td>
<td>Drive 3-phase Maxon's BLDC motor using XMC1000 motor control application kit</td>
</tr>
<tr>
<td>Part Number</td>
<td>KIT_XMC1X_AK_MOTOR_001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Default Configuration in Example Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Scheme</td>
<td>BLDC_SCALAR_SPEED_CTRL</td>
</tr>
<tr>
<td>PWM Modulation</td>
<td>BLDC_SCALAR_PWM_HIGHSIDE</td>
</tr>
<tr>
<td>PWM frequency (Hz)</td>
<td>20000</td>
</tr>
<tr>
<td>Speed (rpm)</td>
<td>2000</td>
</tr>
<tr>
<td>Ramp up/down rate</td>
<td>500</td>
</tr>
<tr>
<td>Protection</td>
<td>Over-current protection with direct DC link current measurement</td>
</tr>
</tbody>
</table>

» Performance Matrix

<table>
<thead>
<tr>
<th></th>
<th>Execution Time (us)</th>
<th>Code size (kbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control loop ISR</td>
<td>9.2</td>
<td></td>
</tr>
<tr>
<td>Motor state machine</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>BEMF zero cross ISR</td>
<td>14.2</td>
<td>13.976</td>
</tr>
</tbody>
</table>

Default configuration: Execution Time and Code Size
## Infineon’s XMC1000 Motor Control Application Kit

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMC1300 CPU Card</td>
<td>MCU board with XMC1300 and detachable SEGGER J-Link debug interface</td>
</tr>
</tbody>
</table>
| 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) |
Hardware Overview – XMC1300 CPU Card

XMC1300 CPU Card

- Detachable COM and SEGGER J-Link Debugger
- Power supply selector for XMC1300
- Power on LED
- Micro USB
- User LEDs
- XMC1300 Evaluation Board
- JP101
- JP102
- Application card connector
- Potentiometer
- JP103
- XMC1300
- JP104
Hardware Overview – Motor Card

› PMSM Low Voltage 15W Motor Card
Hardware Overview – Kit Order information

<table>
<thead>
<tr>
<th>No.</th>
<th>Kit Name</th>
<th>Kit Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KIT_XMC1x_AK_Motor_001</td>
<td>XMC1000 Motor Control Application Kit</td>
<td>KIT_XMC1x_AK_Motor_001</td>
</tr>
</tbody>
</table>
Tools Overview

› **DAVE™ (V4.2.6 onwards)**
  - Download DAVE™ installer package from [http://www.infineon.com/dave](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](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”

   ![Screenshot of Dave Site dialog]

   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

   ![Screenshot of Import Example Project dialog]
1. Select the Motor Control Kit and BLDC motor

```c
/* Motor and power board selection
116 ***************************************************************************/
117 /* Board and motor selection */
118 */**
119 /* Motor control kit */
120 /* Options - KIT_XMC1X_AK_MOTOR_001, KIT_XMC750WATT_MC_AK_V1, KIT_CUSTOM */
121 */
122 #define MOTOR0_BLD_C_SCALAR_BOARD (KIT_XMC1X_AK_MOTOR_001)
123 */**
124 * BLDC motor
125 * Options - MOTOR_EC_MAXON_267121, MOTOR_CUSTOM */
126 */
127 #define MOTOR0_BLD_C_SCALAR_MOTOR (MOTOR_EC_MAXON_267121)
128 */
129 /* Motor Parameters */
130 #if (MOTOR0_BLD_C_SCALAR_MOTOR == MOTOR_EC_MAXON_267121)
131 #define MOTOR0_BLD_C_SCALAR_MOTOR_NO_LOAD_SPEED (4530U) /**< No load speed of the motor in RPM */
132 #define MOTOR0_BLD_C_SCALAR_MOTOR_POLE_PAIRS (4U) /**< Pole pairs */
133 #elif (MOTOR0_BLD_C_SCALAR_MOTOR == MOTOR_CUSTOM)
134 #define MOTOR0_BLD_C_SCALAR_MOTOR_NO_LOAD_SPEED (6200U) /**< No load speed of the motor in RPM */
135 #define MOTOR0_BLD_C_SCALAR_MOTOR_POLE_PAIRS (4U) /**< Pole pairs */
136 #endif
```
2. Select the Control Scheme and PWM Modulation Scheme

```c
/* Control scheme configurations */
/**
 * 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_SYNCHRECT
 */
#define MOTOR0_BLDC_SCALAR_MODULATION (BLDC_SCALAR_PWM_HIGHSIDE)
/**
 * PWM Frequency and scheduler settings
 */
#define MOTOR0_BLDC_SCALAR_PWM_FREQ (20000.0F)
/** Scheduler interrupt period. Range: 1000 to 10000 us*/
#define MOTOR0_BLDC_SCALAR_SYSTICK_PERIOD (1000.0F)
```

![Diagram showing control scheme and speed control](image)

Copyright © Infineon Technologies AG 2016. All rights reserved.
Getting Started – Configure the Project [3/6]

3. Configure the Power Board

Folder: Configuration
File name: bldc_scalar_user_config.h

```c
/* Power board parameters */

#define MOTORO_BLDSCALAR NOMINAL_DC_LINK_VOLT (24.0F)  /* DC link voltage */
#define MOTORO_BLDSCALAR RISING DEAD_TIME (0.75F)  /* Dead time for rising edge in uSec */
#define MOTORO_BLDSCALAR FALLING DEAD_TIME (0.75F)  /* Dead time for falling edge in uSec */
#define MOTORO_BLDSCALAR SWITCH_DELAY (0.75F)  /* Switch delay in uSec */

#define MOTORO_BLDSCALAR_HS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR ACTIVE HIGH)  /* Active level of the high side switch. Option */
#define MOTORO_BLDSCALAR_LS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR ACTIVE HIGH)  /* Active level of the low side switch. Option: */
#define MOTORO_BLDSCALAR INVERTER_ENABLE_CONF (BLDC_SCALAR_INV_ACTIVE_HIGH)  /* Active level of inverter enable. Options: BI */

/* ADC Measurement parameters */

#define MOTORO_BLDSCALAR VADC REF VOLTAGE (5.0F)  /* Reference voltage of VADC conversion */
#define MOTORO_BLDSCALAR CURRENT AMPLIFIER OFFSET (2.5F)  /* Amplifier offset voltage */
#define MOTORO_BLDSCALAR CURRENT RSHUNT (50.0F)  /* Current amplifier shunt resistor value in mOhms */
#define MOTORO_BLDSCALAR CURRENT AMPLIFIER GAIN (16.4F)  /* Current amplifier gain */
#define MOTORO_BLDSCALAR VOLTAGE DIVIDER RATIO (9.79F)  /* Voltage divider ratio in % for DC link voltage measurement */
#define MOTORO_BLDSCALAR BEMF DIVIDER RATIO (9.79F)  /* Voltage divider ratio in % for phase voltage measurement */

/* end of if (MOTORO_BLDSCALAR BOARD == KIT_XMC1X_AK_MOTOR_001) */
```

Diagram showing PWM timing and dead time.
4. Configure the Power Board

```c
/*---------------------- Power board parameters ----------------------*/
#define MOTOR0_BLD_CSCALAR_BOARD == KIT_XMC1AK_MOTOR_001

/* Power Inverter parameters */
#define MOTOR0_BLD_CSCALAR_INPUT_VOLT (24.0F)  /*!< DC link voltage */
#define MOTOR0_BLD_CSCALAR_INPUT_RISING_DEAD_TIME (0.75F)  /*!< Dead time for rising edge in uSec */
#define MOTOR0_BLD_CSCALAR_INPUT_FALLING_DEAD_TIME (0.75F)  /*!< Dead time for falling edge in uSec */
#define MOTOR0_BLD_CSCALAR_INPUT_SWITCH_DELAY (0.75F)  /*!< Switch delay in uSec */

#define MOTOR0_BLD_CSCALAR_HS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR_ACTIVE_HIGH)  /*!< Active level of the high side switch. Option: */
#define MOTOR0_BLD_CSCALAR_LS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR_ACTIVE_HIGH)  /*!< Active level of the low side switch. Option: */
#define MOTOR0_BLD_CSCALAR_INVERNER_ENABLE_CONFIG (BLDC_SCALAR_INV_ACTIVE_HIGH)  /*!< Active level of inverter enable. Options: */

/* ADC Measurement parameters */
#define MOTOR0_BLD_CSCALAR_VADC_REF_VOLTAGE (5.0F)  /*!< Reference voltage of VADC conversion */
#define MOTOR0_BLD_CSCALAR_VOLTAGE_AMPLIFIER_OFFSET (2.5F)  /*!< Amplifier offset voltage */
#define MOTOR0_BLD_CSCALAR_CURRENT_RSHUNT (50.0F)  /*!< Current amplifier shunt resistor value in mOhms */
#define MOTOR0_BLD_CSCALAR_CURRENT_GAIN (16.4F)  /*!< Current amplifier gain */
#define MOTOR0_BLD_CSCALAR_VOLTAGE_DIVIDER_RATIO (9.79F)  /*!< Voltage divider ratio in % for DC link voltage measurement */
#define MOTOR0_BLD_CSCALAR_BEMF_DIVIDER_RATIO (9.79F)  /*!< Voltage divider ratio in % for phase voltage measurement */
```

Folder: Configuration
File name: bldc_scalar_user_config.h
5. Configure the startup method

Folder: Configuration
File name: bldc_scalar_user_config.h

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

```c
#define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER
#define MOTOR0_BLDC_SCALAR_SL_BEMF_TRIGGER_OFFSET

// 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
1. Initialize the uCProbe before starting motor

```c
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]

2. Added uCProbe scheduler in motor state machine

Folder: Interrupts
File name: bldc_scalar_sl_state_machine.c

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

```c
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;

        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;
}

#elif (MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE == 1)
Motor0_BLDC_SCALAR_ucProbeScheduler();
#endif
```
Getting Started –
Code added to support uCProbe [3/3]

3. Added uCProbe scheduler in motor state machine

```c
37 #if ((MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE==1))
38 /*UCproBE scheduler function to handle uprobe comments from UI */
39 void Motor0_BLDC_SCALAR_uCProbe_Scheduler(void)
40 {
41   switch(Motor0_BLDC_SCALAR_uprobe.control_word)
42   {
43     case 1: /* Start the motor */
44        Motor0_BLDC_SCALAR_uprobe.control_word=0;
45        Motor0_BLDC_SCALAR_MotorStart();
46        break;
47
48     case 2: /*Stop the motor*/
49        Motor0_BLDC_SCALAR_uprobe.control_word=0;
50        Motor0_BLDC_SCALAR_MotorStop();
51        break;
52
53     case 3: /*Clear Error state*/
54        Motor0_BLDC_SCALAR_uprobe.control_word=0;
55        Motor0_BLDC_SCALAR_ClearErrorState();
56        break;
57
58     case 4: /*Clear flash and load default value into flash*/
59        Motor0_BLDC_SCALAR_uprobe.control_word=0;
60        Motor0_BLDC_SCALAR_uprobe.user_config[] =0;
61        Motor0_BLDC_SCALAR_Write_Default_value();
62        Motor0_BLDC_SCALAR_uCProbe_Write_Flash();
63        break;
```

Folder: uCProbe
File name: ucProbe.c

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
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

Motor state Indication
2. Click the ‘Run’ button
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.
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
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:

<table>
<thead>
<tr>
<th>Development Boards</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMC1300 Boot Kit</td>
<td>KIT_XMC13_BOOT_001</td>
</tr>
<tr>
<td>PMSM Low Voltage 15W Card</td>
<td>KIT_XMC1x_AK_Motor_001</td>
</tr>
</tbody>
</table>
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
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
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.