XMC13 BLDC Scalar Control Software based on 3 Hall sensor

**Getting Started**

XMC™ Microcontrollers

Oct 2016
Agenda

1. Overview of BLDC Scalar Control SW
2. Software Overview
3. Hardware Overview
4. Tools Overview
5. Getting Started
6. General Information
Overview - BLDC Scalar Control SW

› This document provides information about usage of BLDC scalar control example software on Infineon's XMC1300 series micro-controllers platform.

› BLDC scalar control example software is offered as "simple main project in DAVE™ IDE".

› BLDC scalar control example project consists of Hall based 3-Phase BLDC Motor control algorithm software, targeted end applications are fans, pumps, power tools and e-bike segment.

› This example project will provide high level of configurability and modularity to address different segments.

› This project can be easily configured as per requirements with the help of configuration files.
Software Overview – Software Blocks

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)

Device | XMC1302
--- | ---
Compiler | GCC
## Software Overview – Key Features

<table>
<thead>
<tr>
<th>Supported Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Seamless bi-directional control</strong></td>
<td>Reverse the motor direction without stopping the motor</td>
</tr>
<tr>
<td><strong>On fly start-up</strong></td>
<td>Catch spinning motor at start-up without stop</td>
</tr>
<tr>
<td><strong>Adaptive Hall pattern learning</strong></td>
<td>Synchronise inverter commutation logic with Hall pattern</td>
</tr>
<tr>
<td><strong>Accurate measurement of speed (across wide range)</strong></td>
<td>Use floating pre-scalar</td>
</tr>
<tr>
<td><strong>Demagnetization blanking</strong></td>
<td>Remove spike in direct DC link current measurement</td>
</tr>
<tr>
<td><strong>DC bus voltage clamping</strong></td>
<td>Prevent over-voltage during fast braking</td>
</tr>
<tr>
<td><strong>Protection</strong></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.c:
- bldc_scalar_user_interface.c
- bldc_scalar_variables_config.h
- bldc_scalar_variables_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_init.c
- bldc_scalar_protection_evt.c

Configuration:
- bldc_scalar_MCUIInitconfig.h
- bldc_scalar_user_parameters.h
- bldc_scalar_derived_parameters.c

Control Module:
- bldc_scalar_control_scheme.c
- bldc_scalar_control_scheme.h
- bldc_scalar_pl.c
- bldc_scalar_pl.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.h
- bldc_scalar_control_hall.h

Midsys:
- bldc_scalar_pwm_bc.c
- bldc scalar_pos_fwdhall.c
- bldc scalar_volt_dibus.c
- bldc scalar_voltpotentiometer.c
- bldc scalar_volt_phase.c
- bldc scalar_current_motor.c

MCU Init:
- ccc44.c
- ccu4.c
- posif.c
- nvic.c
- gpio.c

LLD/ Hardware Abstraction layer

MCU HW Peripherals

Core Peripheral

Analoga/Digital/Communication Peripherals

Get Position and Speed
Get Current/Voltage
PWM Update

ISR

BLDC control Algorithm

Inputs:
Set Value, Motor current, Motor Speed

User Parameters
HW Resource Parameters
Derived Parameters

Control Algorithm

PI Controller
PT1 Filter
Ramp Function
Voltage Compensation

Midsys

MCU Init
# 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>Phase delay and blanking for hall sampling</td>
</tr>
<tr>
<td>2</td>
<td>CCU40 _CC41</td>
<td>Fast Sync is enabled</td>
<td>Multi-channel Pattern synchronization</td>
</tr>
<tr>
<td>3</td>
<td>CCU40 _CC42</td>
<td>Always</td>
<td>Capture time interval between two hall</td>
</tr>
<tr>
<td>4</td>
<td>POSIF0</td>
<td>Always</td>
<td>Hall and MCM handling</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 - Interrupt Service Routines

<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>Protection</td>
<td>19</td>
<td>Asynchronous</td>
<td>0</td>
</tr>
<tr>
<td>CCU8</td>
<td>CTRAP</td>
<td>26</td>
<td>Asynchronous</td>
<td>0</td>
</tr>
<tr>
<td>POSIF</td>
<td>HALL Event</td>
<td>27</td>
<td>Asynchronous (Hall edge event only for catch-free running)</td>
<td>1</td>
</tr>
<tr>
<td>POSIF</td>
<td>Pattern Update</td>
<td>28</td>
<td>Based upon speed: 1/(electrical speed in Hz * 6)</td>
<td>1</td>
</tr>
<tr>
<td>CCU8</td>
<td>Control Loop</td>
<td>25</td>
<td>1/ PWM frequency</td>
<td>2</td>
</tr>
<tr>
<td>SYSTIMER</td>
<td>Scheduler</td>
<td>-1</td>
<td>1 mSec (configurable)</td>
<td>3</td>
</tr>
</tbody>
</table>

**Folder: Interrupts**

**File name:** bldc_scalar_state_machine.c

![Folder Structure](image)
Software Overview – Example Configuration

<table>
<thead>
<tr>
<th>Example Name</th>
<th>BLDC_SCALAR_HALL_XMC13</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>MOTOR0_BLDSCALAR_SPEED_CTRL</td>
</tr>
<tr>
<td>PWM Modulation</td>
<td>MOTOR0_BLDSCALAR_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>Execution Time and Code Size</th>
<th>Code Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control loop ISR</td>
<td>6.5</td>
</tr>
<tr>
<td>Pattern update ISR</td>
<td>14.6</td>
</tr>
<tr>
<td>Motor state machine - Normal state</td>
<td>5.92</td>
</tr>
</tbody>
</table>

Execution Time and Code Size For default configuration
### 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)                                                 |

[XMC1300 CPU Card](#)  
[PMSM Low Voltage 15W Motor Card](#)
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
- JP101
- JP102
- XMC1300 Evaluation Board
- 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
  - 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
Note: For this motor kit, one of the hall signal inputs are at P0.15. Therefore, to avoid conflict to the device, please ensure the following settings to the XMC1302 CPU Board.

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

2. Set SWCLK on the dip switch to “OFF” position.
Getting Started – Download Project from DAVE [1/2]

1. Open DAVE™

2. Install example project from DAVE:
   - Help → Install DAVE APP/Example/Device Library...
3. In the opened dialog “Dave Site”:
   - In Option “Work With:”, select “DAVE Project Library Manager”
   - In “Libraries”, select the project “BLDC_SCALAR_HALL_XMC13”.

It is available at 2 locations.

[#1] Boards and Kits →
XMC1000 Motor Control Application Kit

[#2] XMC1000 → XMC1300 Series →
DAVE v4 Example Projects with XMC Lib → Motor Control

4. Select “Yes” to import the example project in workspace
1. Select the Motor Control Kit and BLDC motor

```c
#define MOTOR0_BLDC_SCALAR_BOARD (KIT_XMC1X_AK_MOTOR_001)

#define MOTOR0_BLDC_SCALAR_MOTOR (MOTOR_EC_MAXON_267121)

#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 */
#endif

#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 */
#else
#endif
```
2. Select the Control Scheme and PWM Modulation Scheme

```c
#define MOTOR0_BLDC_SCALAR_CTRL_SCHEME (BLDC_SCALAR_SPEED_CTRL)
#define MOTOR0_BLDC_SCALAR_MODULATION (BLDC_SCALAR_PWM_HIGHSIDE)
```

---

---

- `ref_speed` (reference speed)
- `fdbk_speed` (feedback speed)

Control Scheme: Speed Control

PI Controller

```c
#define MOTOR0_BLDC_SCALAR_PWM_FREQ (20000.0F)
#define MOTOR0_BLDC_SCALAR_SYSTICK_PERIOD (1000.0F)
```
### 3. Configure the Power Board

Folder: Configuration  
File name: bldc_scalar_user_config.h

```c
/* Power board configurations */

// Power board parameters

#if (MOTOR0_BLD_C_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001)
#define MOTOR0_BLD_C_SCALAR_NOMINAL_DC_LINK_VOLT 24.0F /* DC link voltage */
#define MOTOR0_BLD_C_SCALAR_RISING_DEAD_TIME 0.75F /* Dead time for rising edge in uSec */
#define MOTOR0_BLD_C_SCALAR_FALLING_DEAD_TIME 0.75F /* Dead time for falling edge in uSec */
#define MOTOR0_BLD_C_SCALAR_SWITCH_DELAY 0.75F /* Switch delay in uSec */
#define MOTOR0_BLD_C_SCALAR_HS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR_ACTIVE_HIGH) /* Active level of the high side switch */
#define MOTOR0_BLD_C_SCALAR_LS_SWITCH_ACTIVE_LEVEL (BLDC_SCALAR_ACTIVE_HIGH) /* Active level of the low side switch */
#define MOTOR0_BLD_C_SCALAR_INVERTER_ENABLE_CONF (BLDC_SCALAR_INV_ACTIVE_HIGH) /* Active level of inverter enable. */

/* ADC Measurement parameters */
#define MOTOR0_BLD_C_SCALAR_VADC_REF_VOLTAGE 5.0F /* Reference voltage of VADC conversion */
#define MOTOR0_BLD_C_SCALAR_CURRENTAMPLIFIER_OFFSET 2.5F /* Amplifier offset voltage */
#define MOTOR0_BLD_C_SCALAR_CURRENTAMPLIFIER_SHUNT 50.0F /* Current amplifier shunt resistor value in mOhms */
#define MOTOR0_BLD_C_SCALAR_CURRENTAMPLIFIER_GAIN 16.4F /* Current amplifier gain */
#define MOTOR0_BLD_C_SCALAR_VOLTAGE_DIVIDER_RATIO 9.79F /* Voltage divider ratio in % for DC link voltage measurement */
#endif /* (MOTOR0_BLD_C_SCALAR_BOARD == KIT_XMC1X_AK_MOTOR_001) */
```

![Diagram showing PWM timer and phases](image-url)

---

October 2016  
Copyright © Infineon Technologies AG 2016. All rights reserved.
4. Configure the Power Board

Folder: Configuration
File name: bldc_scalar_user_config.h
5. Configure the Hall Pattern

```
570/********************************************************/
571 * Hall pattern and phase excitation pattern
572 *******************************************************/
573 #if (MOTOR0_BLDC_SCALAR_FEEDBACK == BLDC_SCALAR_3HALL)
574
define MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ (BLDC_SCALAR_HALL_SEQ_1) /*!< select hall sequence for positive direction */
575
define MOTOR0_BLDC_SCALAR_HALL_POSITIVE_DIR_SEQ -- BLDC_SCALAR_HALL_SEQ_1)
576 /* Standard hall pattern for positive direction. Do NOT change this hall pattern values. */
577 /* Update the phase excitation pattern corresponding to the hall pattern */
578 */
579 #define MOTOR0_BLDC_SCALAR_HALL_PAT_A (1U) /*!< (MSB)H3 H2 H1 (LSB)*/
580 #define MOTOR0_BLDC_SCALAR_HALL_PAT_B (3U) /*!< (MSB)H3 H2 H1 (LSB)*/
581 #define MOTOR0_BLDC_SCALAR_HALL_PAT_C (2U) /*!< (MSB)H3 H2 H1 (LSB)*/
582 #define MOTOR0_BLDC_SCALAR_HALL_PAT_D (4U) /*!< (MSB)H3 H2 H1 (LSB)*/
583 #define MOTOR0_BLDC_SCALAR_HALL_PAT_E (5U) /*!< (MSB)H3 H2 H1 (LSB)*/
584 */
585 */
586 /* Standard hall pattern for positive direction. Do NOT change this hall pattern values. */
587 */
```

6. Configure the ADC Trigger

```
3465 #define MOTOR0_BLDC_SCALAR_CURRENT_TRIGGER_OFFSET (MOTOR0_BLDC_SCALAR_SWITCH_DELAY) /*!< By default, current trigger
```

Copyright © Infineon Technologies AG 2016. All rights reserved.
7. Configure the Hall sensor feedback

Folder: Configuration
File name: bldc_scalar_user_config.h
Folder: Configuration
File name: bldc_scalar_user_config.h

8. Configure the MCM Transfer

```c
#define MOTOR0_BLD_DC_SCALAR_ENABLE_FAST_SYNC_CCU4 (1U) /*!< Enable (1)/disable (0)
#if (MOTOR0_BLD_DC_SCALAR_ENABLE_FAST_SYNC_CCU4 == 1U)
#define MOTOR0_BLD_DC_SCALAR_MCM_SYNCTRANSFER_TIME (2.0F) /*!< Multi-channel pattern
#endif /* end of #if (MOTOR0_BLD_DC_SCALAR_ENABLE_FAST_SYNC_CCU4 == 1U) */
```
Getting Started – Configure the Project [8/8]

9. Enable the Hall pattern learning

Folder: Configuration
File name: bldc_scalar_user_config.h

```c
#define MOTOR0_BLDSCALAR_ENABLE_HALL_LEARNING (1U) /*! < Enable/disable hall pattern learning */
#if (MOTOR0_BLDSCALAR_ENABLE_HALL_LEARNING == 1U)
#define MOTOR0_BLDSCALAR_OPEN_LOOP_VOLTAGE (5.0F) /*! < Open loop voltage to be applied in % */
#define MOTOR0_BLDSCALAR_OPEN_LOOP_SPEED (2.0F) /*! < Speed to be applied in % with respect */
#endif /* if(MOTOR0_BLDSCALAR_ENABLE_HALL_LEARNING == 1U) */
```

Note: For a new motor where the hall pattern is not known, this configuration can be enabled for hall pattern learning.
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 run */
    while (1U)
    {
    }

    }

Folder: -
File name: main.c
2. Add uCProbe scheduler in motor state machine

› Motor control state machine is called on each Systick Interrupt

› uCProbe Scheduler is called on each scheduler tick
3. Add uCProbe scheduler in motor state machine

```c
#if ((__MOTOR0_BLDC_SCALAR_CTRL_UCPROBE_ENABLE==1))

/*UCproBE scheduler function to handle ucprobe comments from UI */

void Motor0_BLDC_SCALAR_uCProbe_Scheduler(void)
{
    switch(Motor0_BLDC_SCALAR_ucprobe.control_word)
    {
        case 1: /* Start the motor */
            Motor0_BLDC_SCALAR_ucprobe.control_word=0;
            Motor0_BLDC_SCALAR_MotorStart();
            break;

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

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

        case 4: /*Clear flash and load default 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;
    }
}
```

uC Probe scheduler routine support control code to control the motor.
Getting Started – Compile and Verify the project

1. Click “Build Active Project”
2. Click “Debug Configuration” to download the code
3. Click “Resume” to start the application

Observation:
- Motor should ramp to 2000RPM with ramp rate of 500RPM/s.
Getting Started – Interface with µC/Probe [1/6]

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

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

Note: As the BMI mode is set to SPD0 mode, the program needs to be started in Debug mode before connecting to the uCProbe project.
Getting Started
Interface with µC/Probe [2/6]

Fine tune the PI value to get optimum motor behaviour

Error state Indication

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

Save the values into flash

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

Save the values into flash

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
3. Go to Tab: HALL Pattern Learning. This is used to find the relation between HALL and commutation pattern.

4. Set the Open Loop Voltage to 10%.

5. Select button “Learn Hall Pattern” to start the Hall Learning.

6. Once the Hall learning is completed, the pattern is displayed.

7. If required, select “Save to Flash” to save the commutation table into the Flash.
8. In the tab “BLDC: Speed Control”, select the various widgets to control the motor.

- Start/Stop control
- PI tuning and monitoring

› Possible to save PI values, commutation table into Flash
9. Click on the “Oscilloscope” tab for monitoring motor control parameters
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
References: Help Content

- Example SW user guide as chm format is part of this example SW
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog Digital Converter</td>
</tr>
<tr>
<td>DAVE™</td>
<td>Digital Application Virtual Engineer (Free development IDE for XMC™)</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
</tbody>
</table>
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.