Washing machine motor control solution with XMC™, IGBT, CoolSET™

January 2017

Learning objectives

- To demonstrate sensorless FOC for washing machine PMSM motor control
- Key software functions, a step-by-step implementation, and linking up with µC/Probe™ XMC™
- To use of µC/Probe™ XMC™ to visualise data and fine-tune FOC
- After the learning of this PPT, users will be able to fine-tune XMC1302 FOC example SW for own washing machine motor control

Agenda (1/2)

1. Overview
2. Key features
3. Specification
4. System block diagram
5. Hardware overview
6. Software overview
7. Highlight MCU features
8. Get started

Agenda (2/2)

9 Resource listing

Washing machine motor control - Overview

- The purpose of the training slides is to elaborate a low-cost and high-performance washing machine motor control solution, using
  - XMC™ 750 watt motor control application kit
  - XMC1300 drive card
  - DAVE™ 4 example project - PMSM_FOC_SL_XMC13
- The HOT examples cover the key features and controls of the washing machine motor control application

Washing machine motor control - the solution focuses on mainly 2 areas

› Infineon focus areas for washing machine motor control

Motor control

Power management

Focus of this value proposition

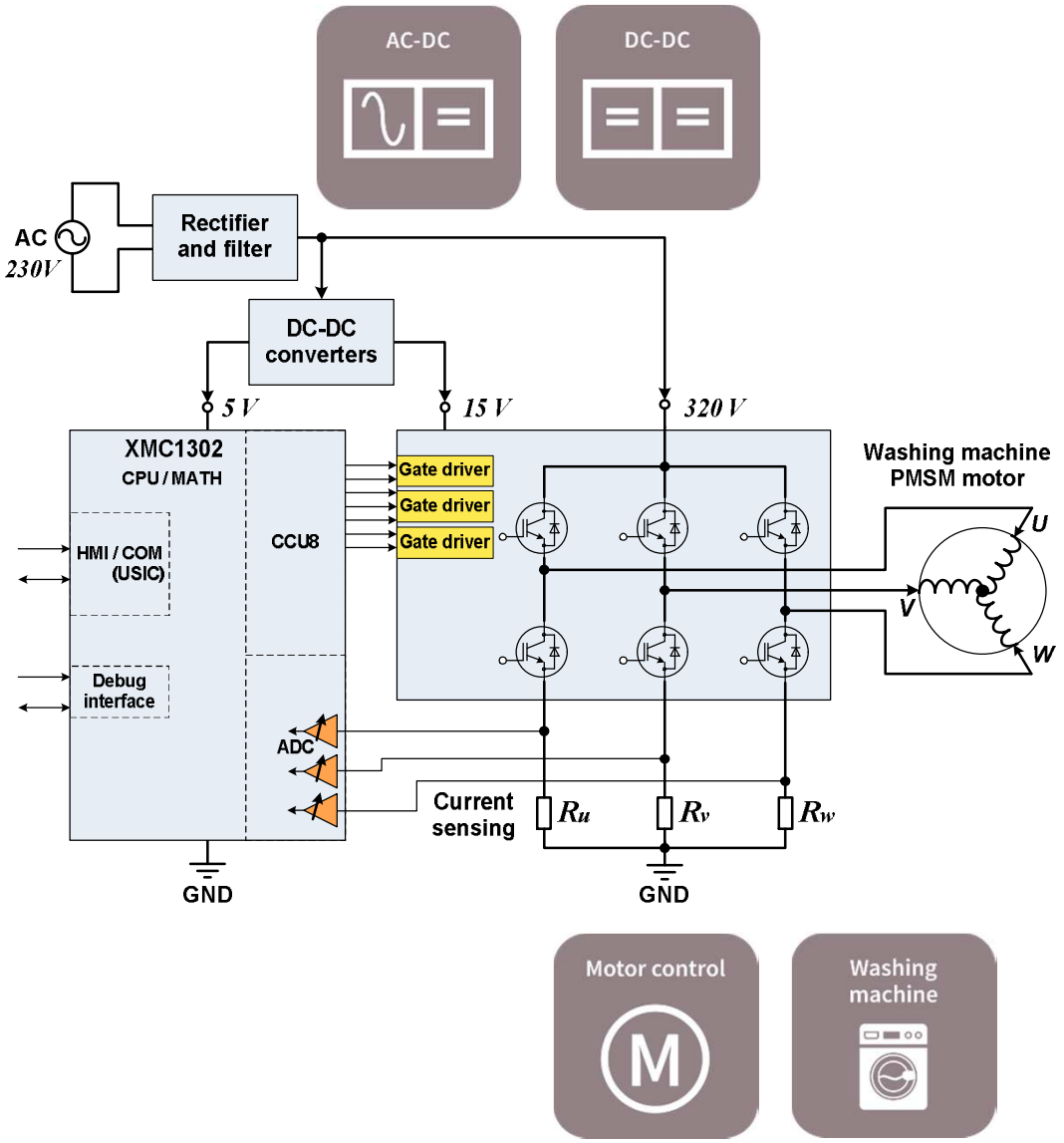
Infineon can provide all the critical components for washing machine motor control

Motor control

Functionality	IFX components
› FOC controller (3-phase PWM generation, motor phase current sensing, bus voltage sensing, over-current & over-voltage protection)	› **XMC1302**: ARM® Cortex®-M0 32-bit processor @ 32 MHz, up to 200 kB flash,16 kB SRAM,MATH coprocessor, 12-bit ADC with 2 sample & hold stages, motor control PWM timer (CCU8), general purpose timer (CCU4), serial communication (USIC)

Power management

Functionality	IFX components
› Power inverter	› **IKD10N60R**: discrete IGBT with anti-parallel diode, 600 V › **6EDL04I06NT**: EiceDRIVER™ gate driver IC
› Auxiliary power supply	› **3F/2QR CoolSET™ flyback controller family** e.g.: ICE3RBR4765JG › **IFX1763XEJV33**: wide input range low noise LDO



Why Infineon?
Washing machine motor control solution

Benefits	Addressed customer needs
Small code size and fast execution time	Allows ample CPU time for more tasks Small code size < 16 kB, and super-fast code execution < 20 μs (for optimized code)
Robust start-up	Direct-sensorless-FOC startup is robust, smooth and energy efficient at various load conditions of washing machine
Less dependencies more robust	Only need one motor parameter to estimate rotor angle and speed for sensorless FOC
Ultra-low speed sensorless control	Robust / quieter operation with sensorless FOC drive even at ultra-low speed (e.g.: 0.8% of max speed)
BOM savings	€ 0.49-0.9 reduction of system BOM using XMC™ on-chip ADC gain. Complete sensorless motor control eliminating Hall sensors / tachometer
ARM® Cortex®-M0 with MATH coprocessor	Replace costly MCUs (e.g. ARM® Cortex®-M3)
Embedded security solutions	Protect customer solution from being copied by customer's competitors
Product portfolio	Infineon a "one-stop-shop" for motor control applications with complete power semiconductor portfolio
Knowledge of vendor	Dedicate motor control expert team support with multiple connection of expertise for local support

Washing machine motor control - Key features

Target application

- Washing machine motor control

Key features

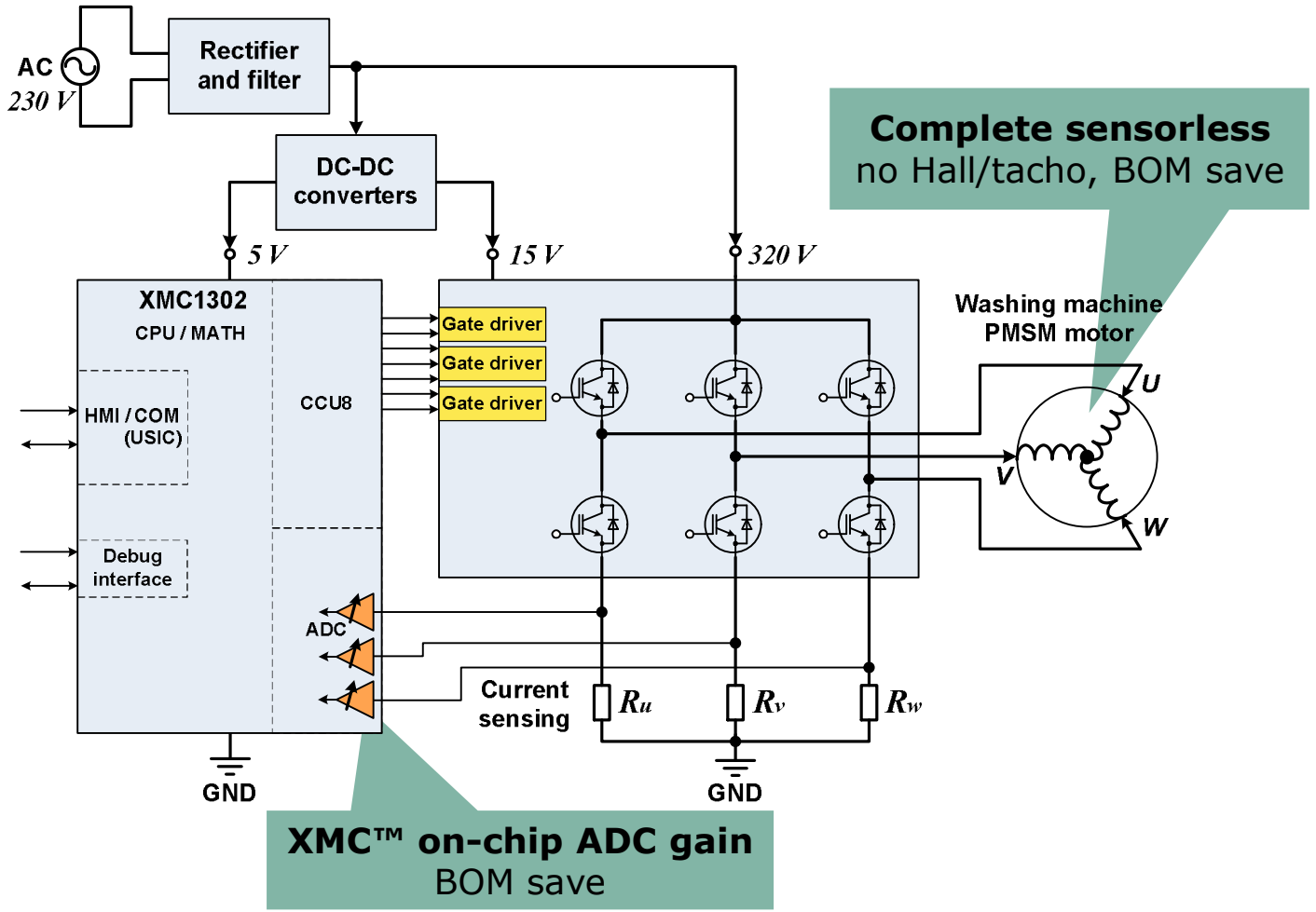
- Sensorless FOC control even at ultra-low speed
- Robust direct-sensorless-FOC startup
- Smooth bi-directional control at low-speed
- Flux Weakening (FW) at high speed for spin dry
- XMC™ on-chip ADC gain to reduce system BOM cost

Washing machine motor control - Specification

Specifications

- AC input voltage: 230 V_{AC}
- Motor mechanical speed
  - Minimum 100 rpm
  - Maximum speed 16,000 rpm at Flux Weakening (FW) for spin dry
  - Motor speed at washing: 600 rpm
- Drum speed: 10 rpm to 1,600 rpm
- Bi-directional control at low-speed (e.g.: 100 rpm)
- No failure at least 500 times for motor startup bi-directionally at various load conditions

Washing machine motor control - System block diagram



System block diagram: washing machine motor control

Washing machine motor control - Hardware overview

- Kit information

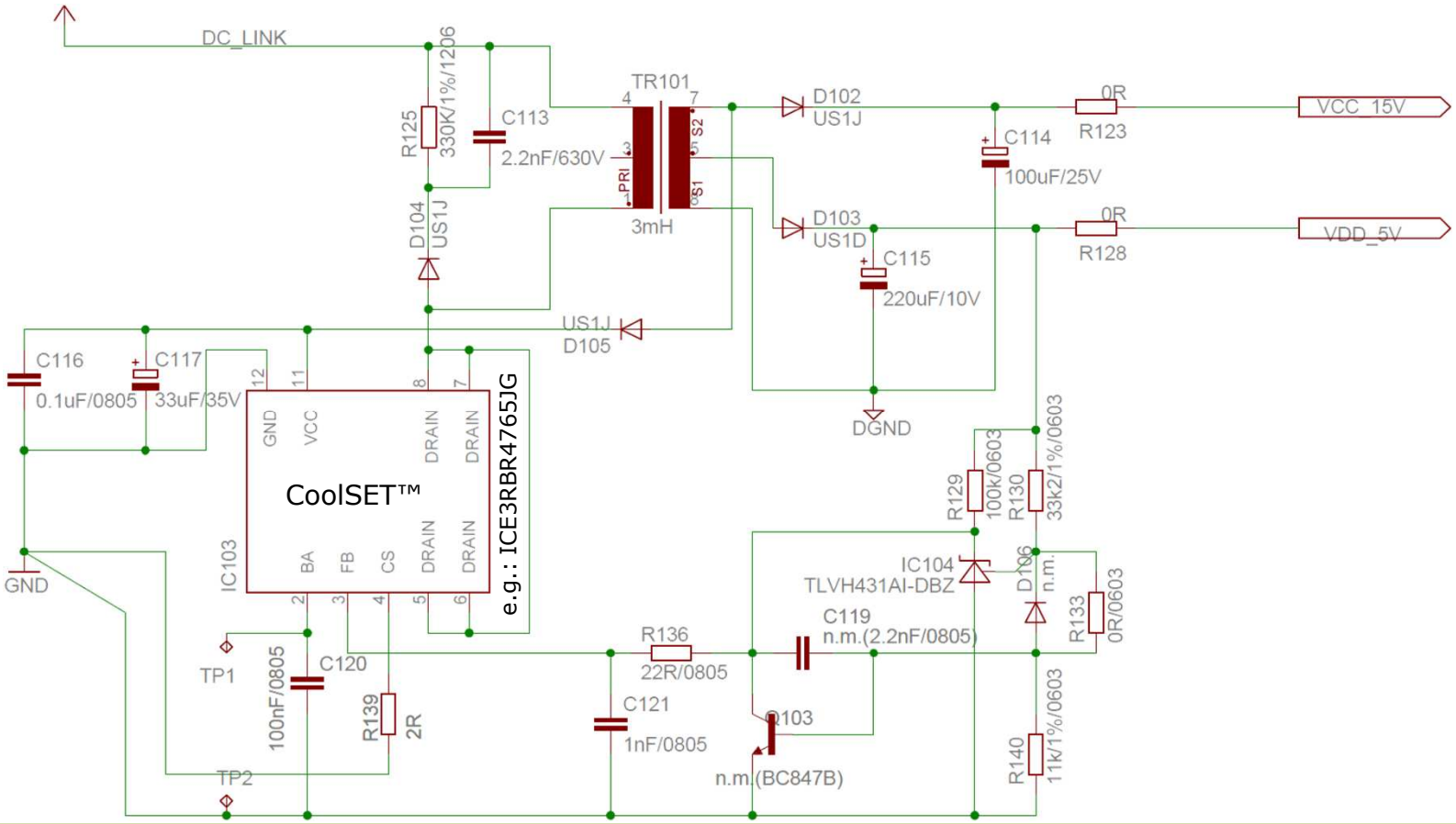
No.	Kit name	Kit description	Order number
1	KIT_XMC1300_DC_V1	XMC1300 drive card	KIT_XMC1300_DC_V1
2	KIT_XMC750WATT_MC_AK_V1	XMC™ 750 watt motor control application kit	KIT_XMC750WATT_MC_AK_V1

- Key Infineon components utilized on kit:

No.	Infineon components	Order number	Quantity per kit
1	XMC™ microcontroller	XMC1302-T038X0200	1
2	3F/2QR CoolSET™ flyback controller family	e.g.: ICE3RBR4765JG	1
3	EiceDRIVER™ gate driver IC	6EDL04I06NT	1
4	Discrete IGBT with anti-parallel diode, 600 V	IKD10N60R	6
5	LDO, wide input range, low noise	IFX1763XEJV33	1

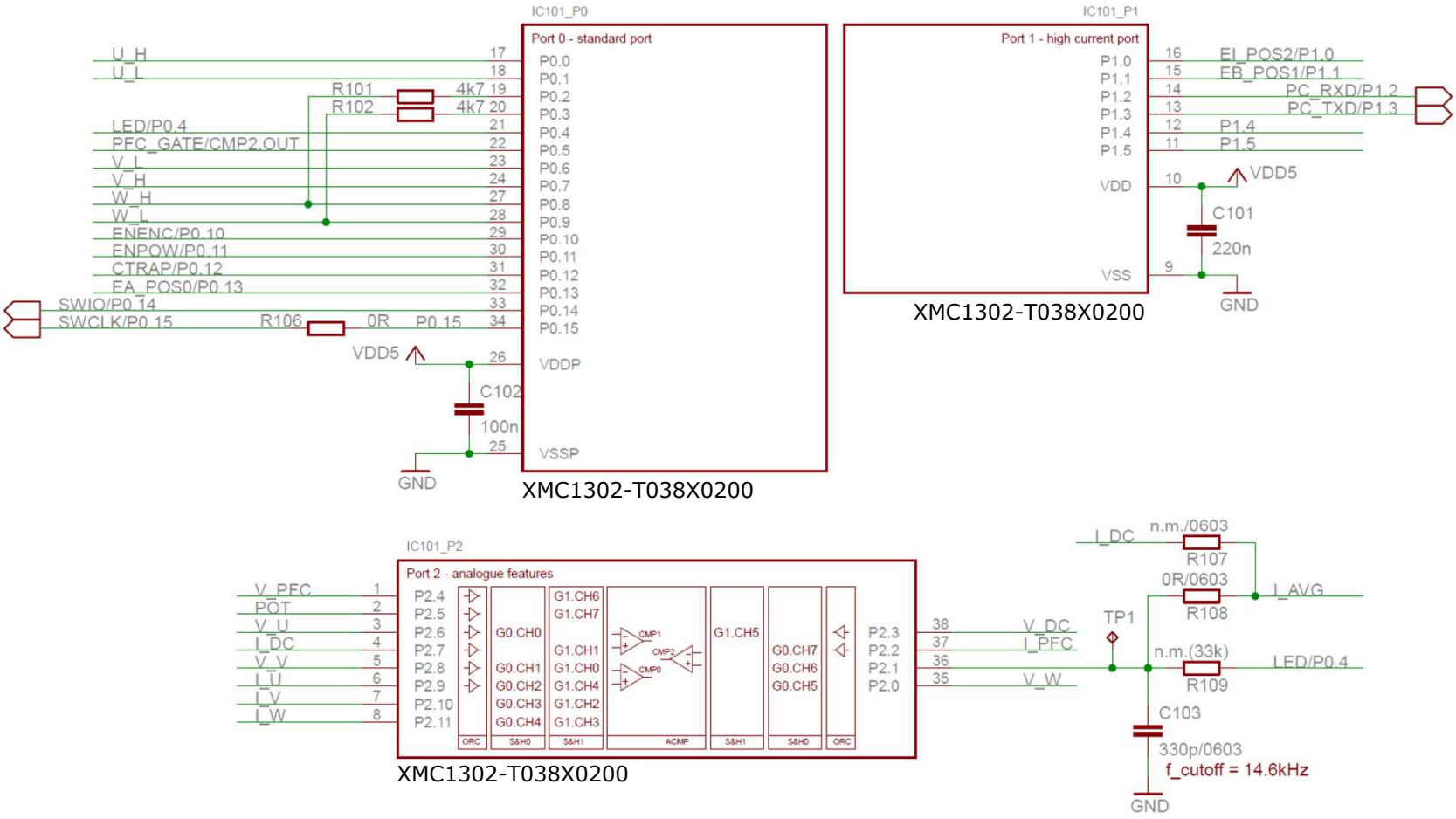
Washing machine motor control - 5 V and 15 V flyback converter schematics

- Infineon offers both Quasi Resonant (QR) CoolSET™ and Fixed Frequency CoolSET™ in various packages, below design is an example



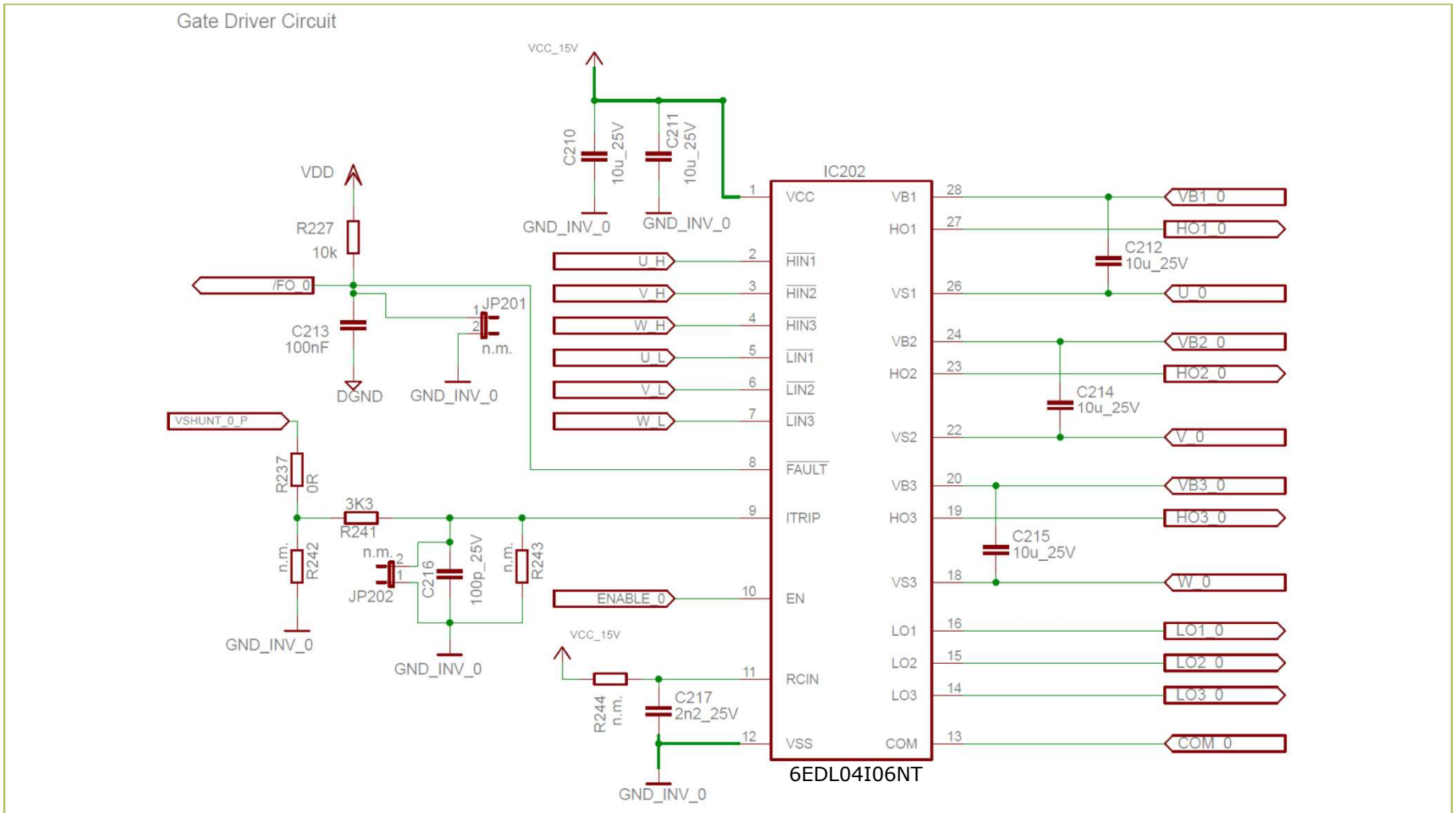
Schematic: washing machine motor control - 5 V and 15 V flyback converter

Washing machine motor control - Microcontroller schematics



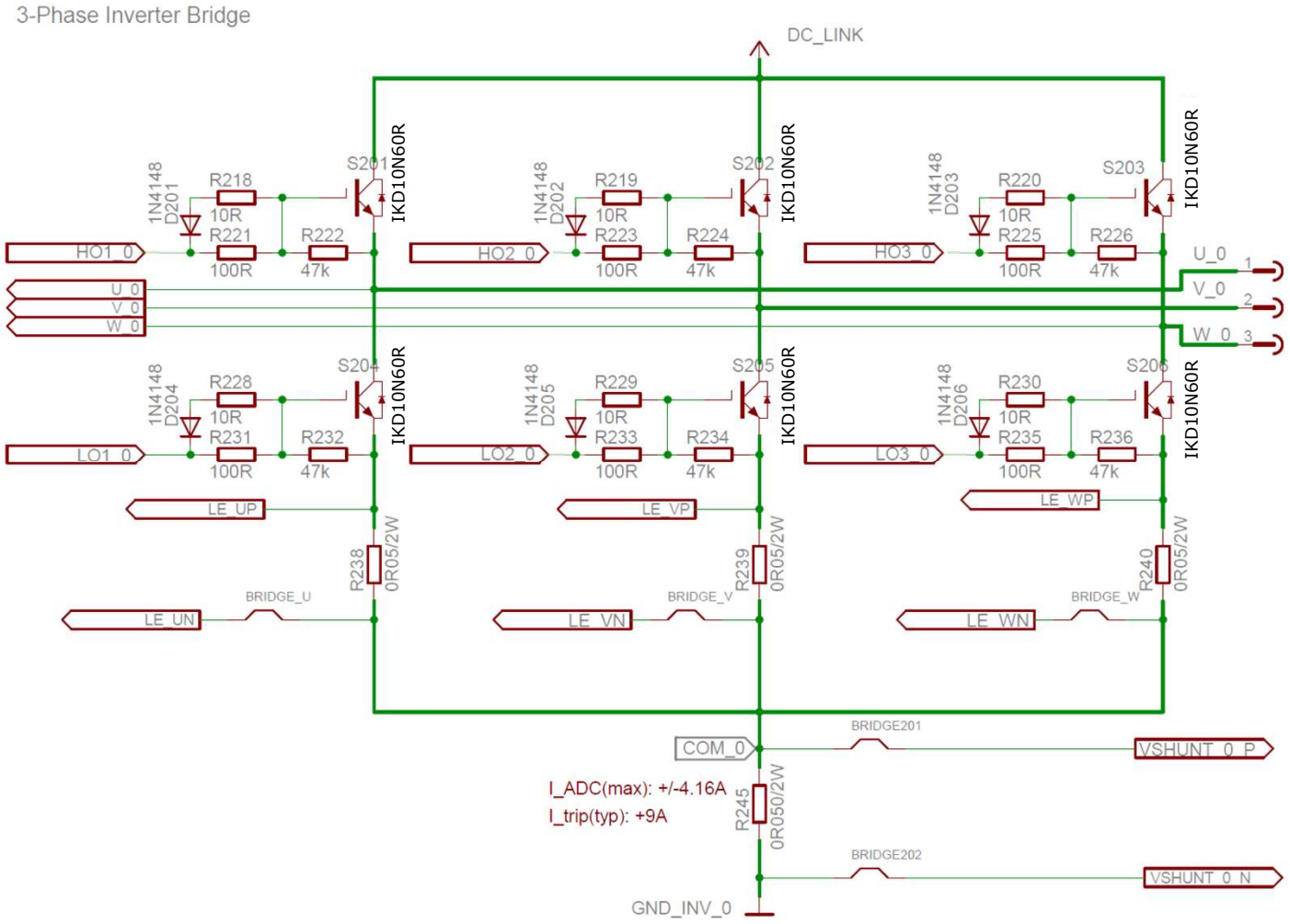
Schematic: washing machine motor control - microcontroller

Washing machine motor control - High-voltage gate driver schematics



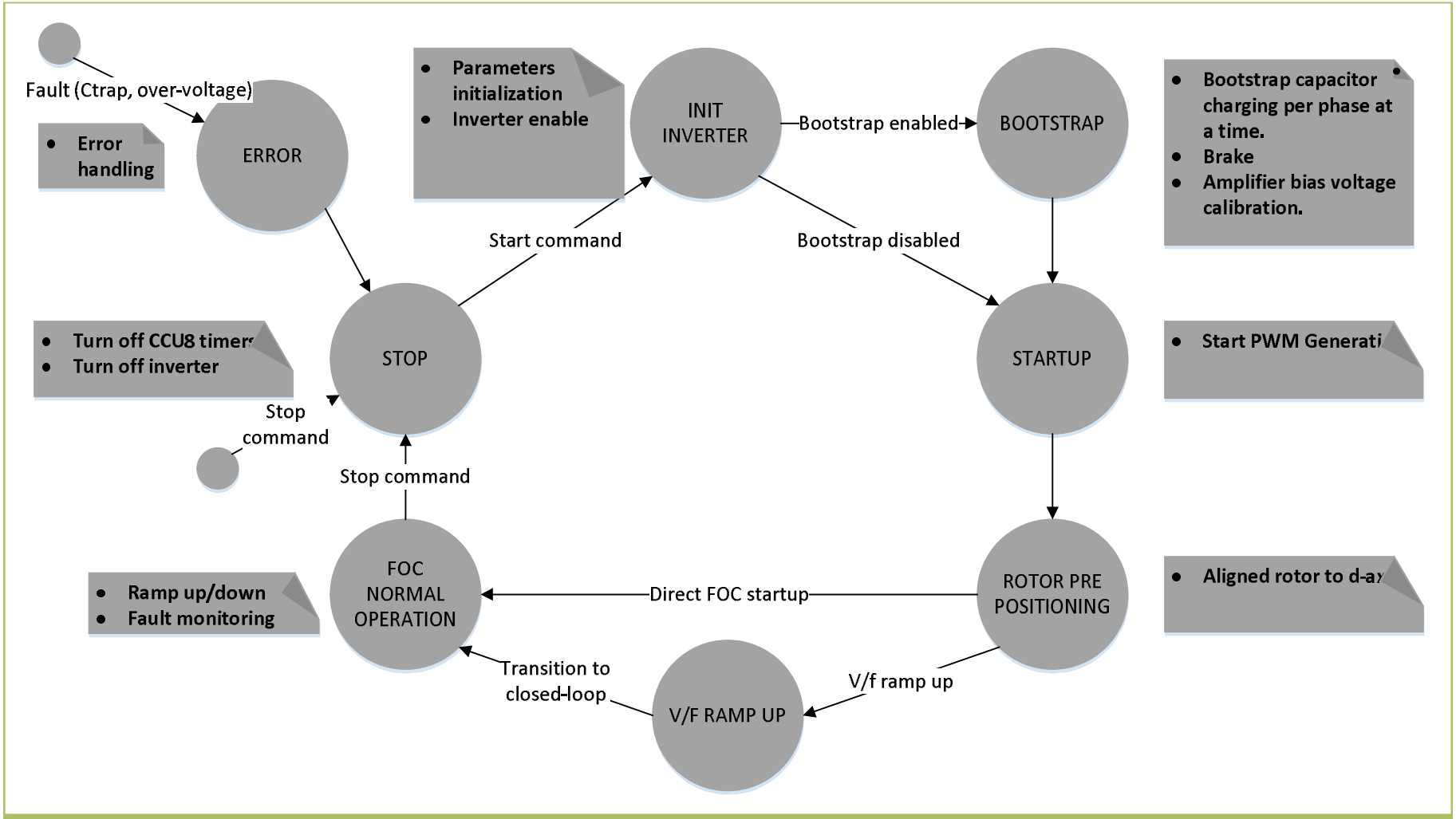
Schematic: washing machine motor control - high-voltage gate driver

Washing machine motor control - 3-phase power inverter schematics



Schematic: washing machine motor control - 3-phase power inverter

Washing machine motor control - Software overview



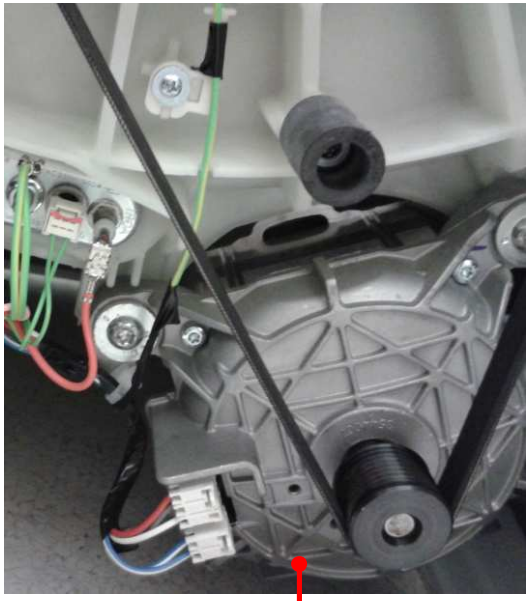
Flow chart: washing machine motor control - software overview

Washing machine motor control - Highlight MCU features

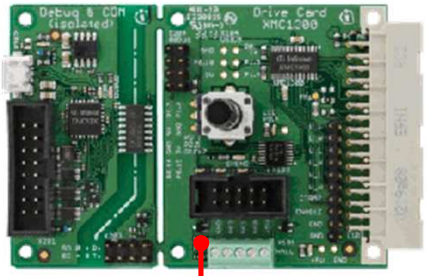
- MATH coprocessor
  - 38x faster sine, cosine and arctangent calculations
  - High-resolution Park/Inverse Park Transforms at 24-bit in less than 1 μs
  - 7x faster division compared to other ARM® Cortex®-M0 devices
- CCU8 PWM
  - Generate PWM patterns for all kind of motors
  - Interact with ADC for ADC triggering at sensorless control of motors
  - Operate always in a safe state - even in an error condition
  - Dead time control to minimum hardware effort
  - 16-bit resolution for high precision space vector PWM generation
- ADC
  - On-chip ADC gain (x1, x3, x6, or x12) to eliminate external Op-Amp
  - Simultaneously sample of multiple analog channels
  - Fast ADC reduces torque ripple due to minimized blind angle in sensorless FOC
  - Used to sense motor three phase current as feedback to the system

Washing machine motor control - Get started - HW connections

› Connect washing machine PMSM motor U, V and W phases to Infineon XMC™ 750 watt motor control application kit



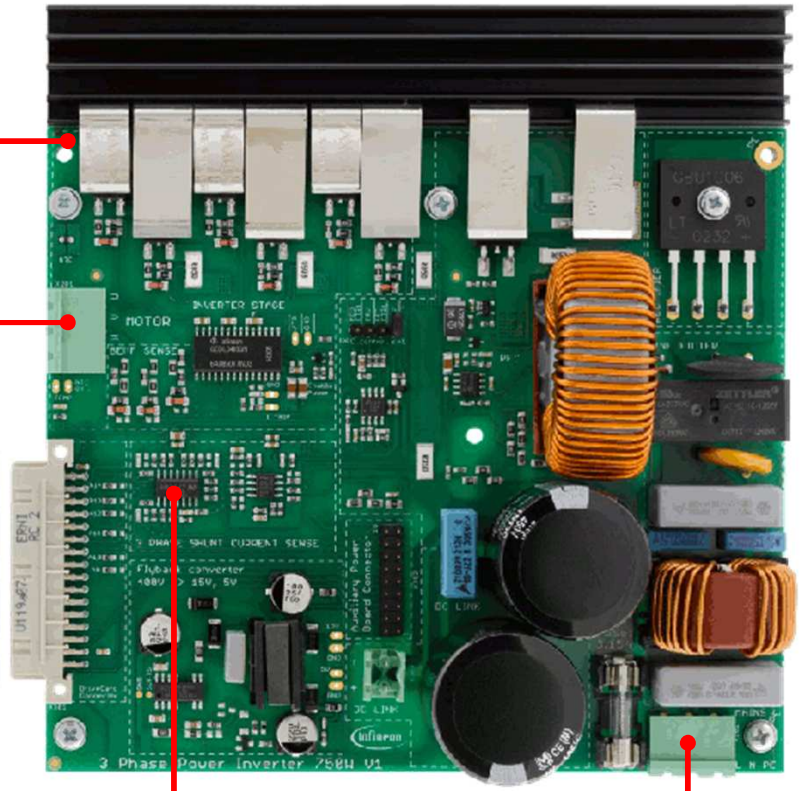
› Washing machine PMSM motor



› XMC1300 drive card

› Power board 110 V -230 V / 750 W

› To motor phases U, V and W



› 230 V_{AC} / 8 A

› Op-Amp gain changed to x6 (R205, R206, R207, R212, R214, R216 reworked to 2 kΩ)

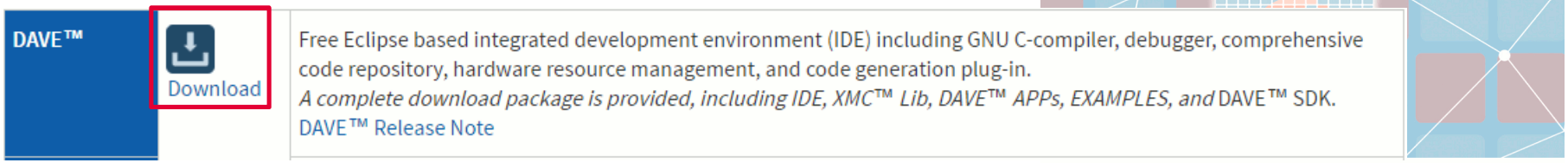
Washing machine motor control - Get started - DAVE™ 4

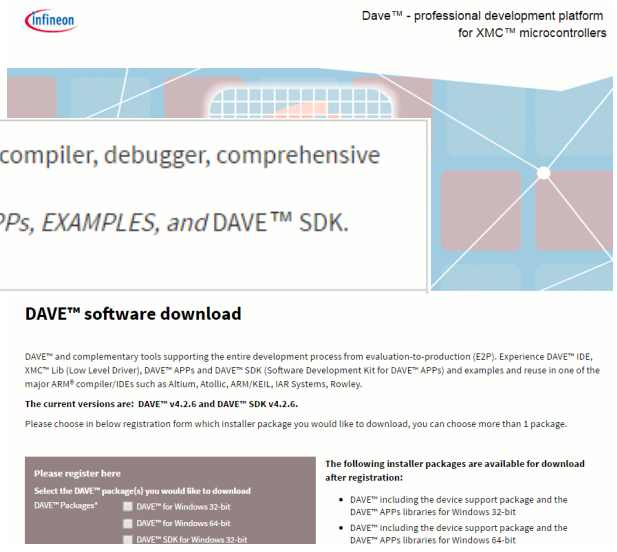
- Download the latest DAVE™ 4 installer package from

  DAVE™ (Version 4) - Development Platform for XMC™ Microcontrollers

- Installation requirements

  1. PC with Windows 7, Windows 8.1, Windows 10, Windows Vista - 32bit & 64bit
  2. RAM - 4 GB or more
  3. Remember to install SEGGER J-Link when installing DAVE™ 4 (if not done so)





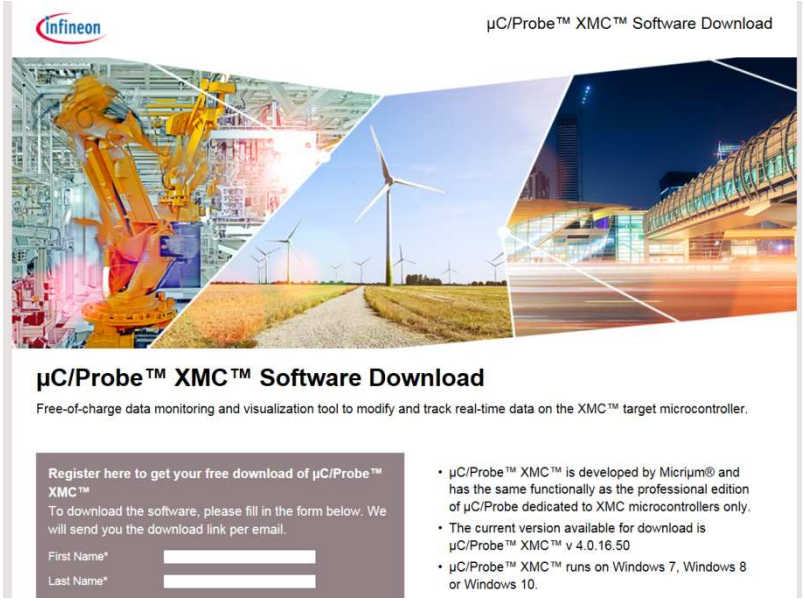
Washing machine motor control - Get started - µC/Probe™ XMC™

- Download the latest µC/Probe™ XMC™ installer package from

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

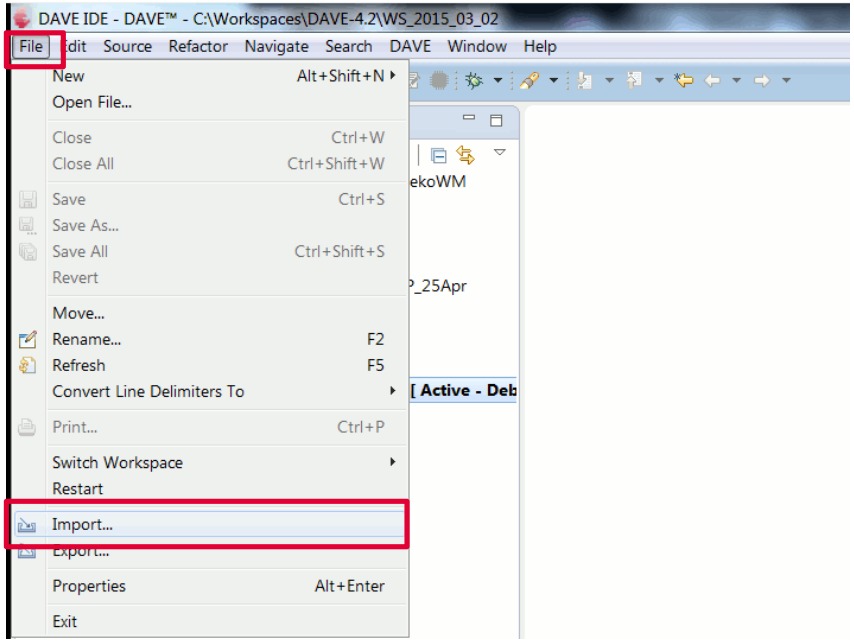
- Installation requirements

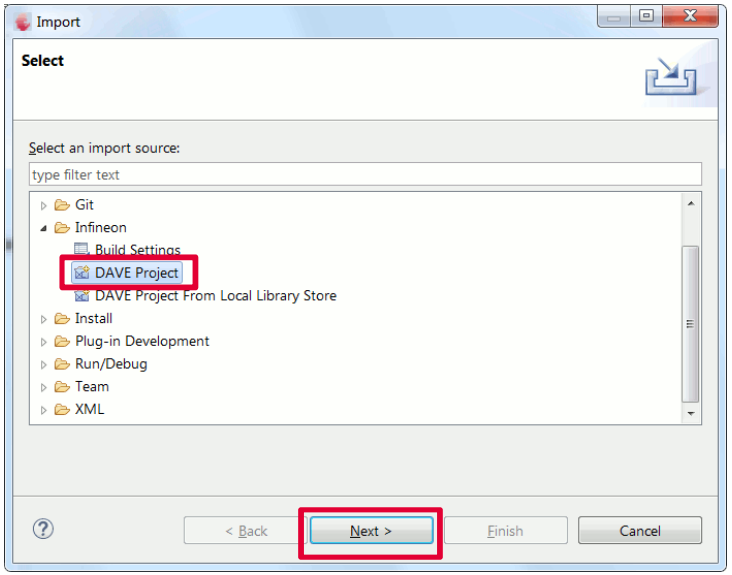
  1. PC with Windows Vista, Windows 7, Windows 8, Windows 10 - 32bit & 64bit
  2. RAM - 3 GB or more



Washing machine motor control - Get started - import SW to DAVE™ 4 (1/2)

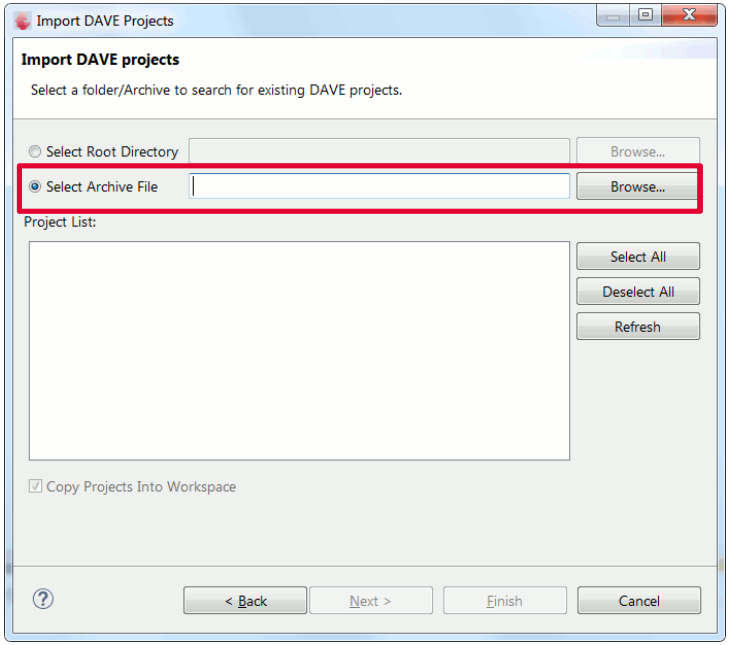
- Open DAVE™ 4
- Click on **File > Import** to import sample code
- Select **Infineon > DAVE project** and click "**Next**"

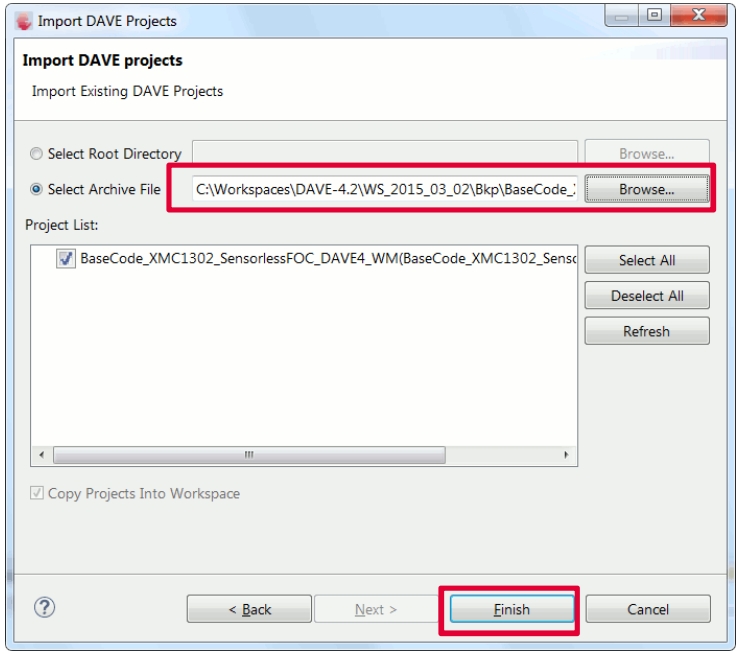




Washing machine motor control - Get started - import SW to DAVE™ 4 (2/2)

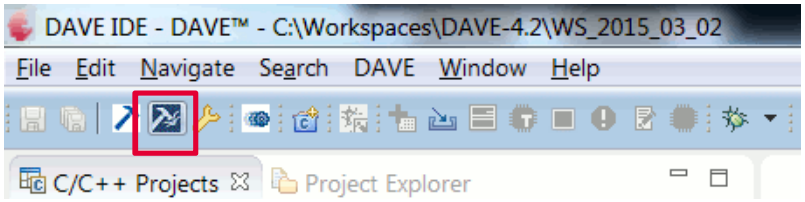
- Next click on **Select Archive File > Browse**
- Select the folder containing the sample code and click "**OK**"
- Click on "**Finish**" to import the code into DAVE™ 4



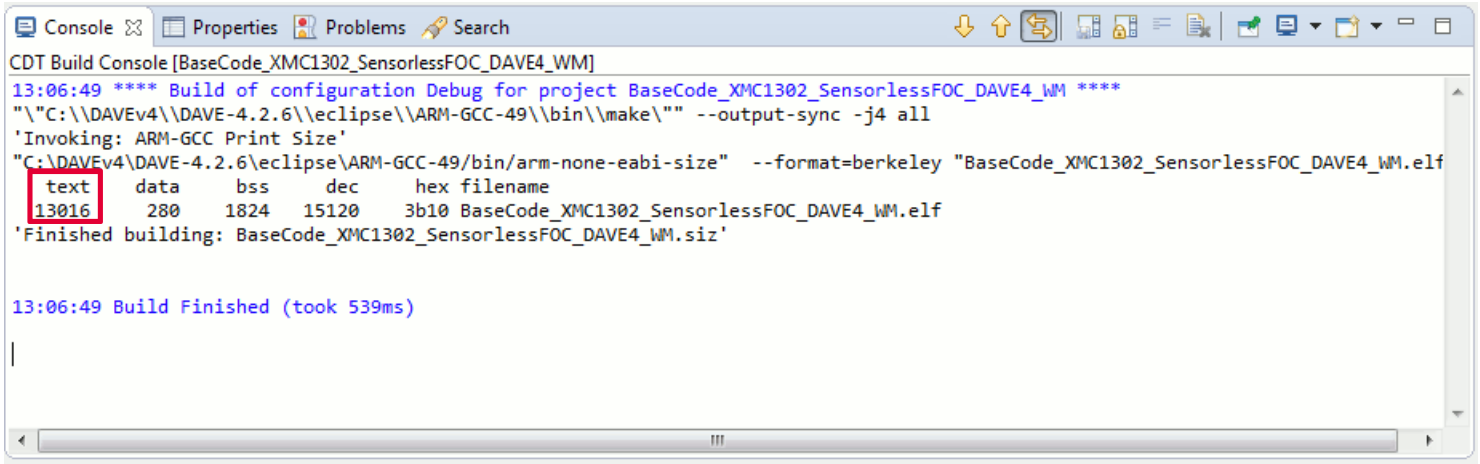


Washing machine motor control - Get started - build SW in DAVE™ 4

- Click "**Rebuild Active Project**"

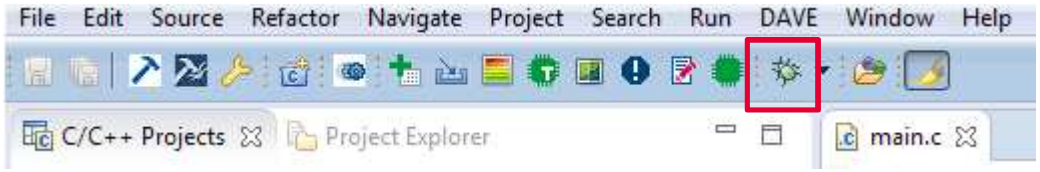


- "**text**" in red box indicates the code size, e.g.: about 13 kB

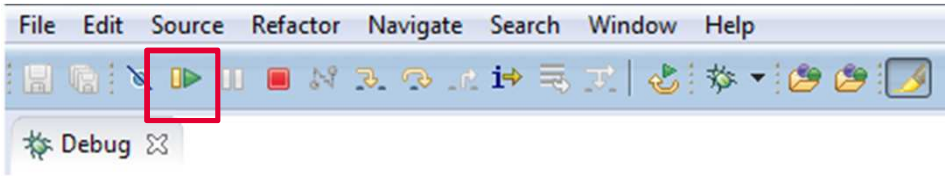


Washing machine motor control - Get started - download SW in DAVE™ 4

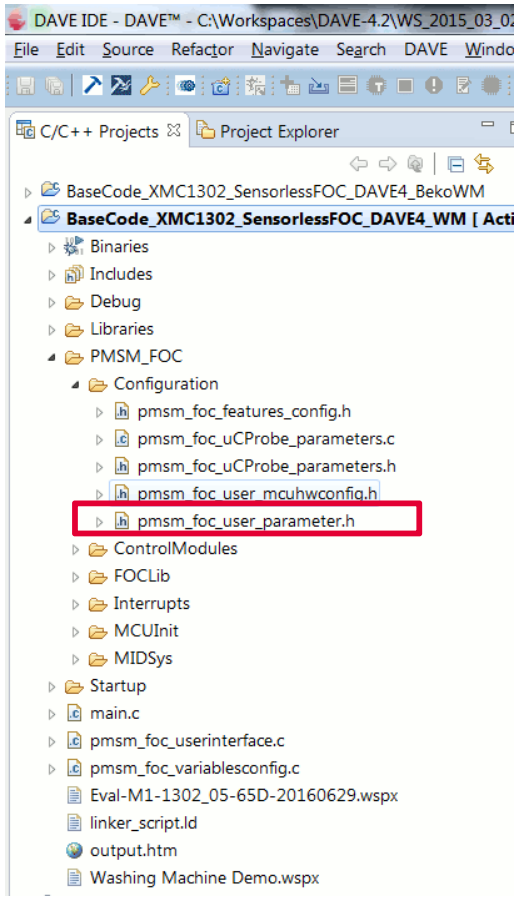
- Click "**Debug Configuration**" to download the code



- Click "**Resume**" to start the motor control application SW



Washing machine motor control - Get started - SW configuration (1/3)



- The FOC example SW enables the user to change certain parameters in order to fine tune motors
- To access the code within DAVE™ 4:
  - Under "**C/C++ Projects**" section, you will find your project (with Active - Debug on it)
  - Select **"Project Title" > PMSM_FOC > Configuration > pmsm_foc_user_parameter.h**
  - Double click to open the file
- **Note:** Files with **"user"** in it indicates that there are **parameters** that can be changed as per hardware and user requirements

Washing machine motor control - Get started - SW configuration (2/3)

pmsm_foc_user_parameter.h

```
52 #include "pmsm_foc_features_config.h"
53
54
55 #define  PMSM_FOC_HARDWARE_BOARD                IFX_XMC_LVPB_R3                /*1. KIT_XMC1X_AK_MOTOR_001
56                                                                                 2. KIT_XMC750WATT_MC_AK_V1
57                                                                                 4. IFX_XMC_LVPB_R3*/
58 #define  MOTOR_TYPE                             DJI_DRONE_MOTOR                /*1. MCI_DRONE_MOTOR
59                                                                                 2. DJI_DRONE_MOTOR*/
60
61 #define  CURRENT_SENSING                        USER_THREE_SHUNT_SENSE         /*1. USER_SINGLE_SHUNT_SENSE
62                                                                                 2. USER_THREE_SHUNT_SENSE*/
63 #define  ADC_CONVERSION                         USER_ADC_SYNC_CONV              /*1. USER_ADC_ASSYNC_CONV
64                                                                                 2. USER_ADC_SYNC_CONV*/
65
66 /*      ------------------------------------------------- FOC Control and Startup Scheme (Only Select 1 Scheme at one time) ---------------------------------------- */
67 #define MY_FOC_CONTROL_SCHEME                   CONSTANT_SPEED_DIRECT_FOC      /* 1. CONSTANT_SPEED_VF_ONLY,
68                                                                                 2. CONSTANT_SPEED_VF_MET_FOC
69                                                                                 3. CONSTANT_SPEED_DIRECT_FOC
70                                                                                 4. CONSTANT_TORQUE_DIRECT_FOC
71                                                                                 5. CONSTANT_VQ_DIRECT_FOC */
72
73 /*      ------------------------------------------------- Motor Parameters ----------------------------------------- */
74 #define  USER_MOTOR_R_PER_PHASE_OHM             (0.1f)          /* Motor Resistance per phase in Ohm*/
75 #define  USER_MOTOR_L_PER_PHASE_uH              (20.0f)           /* Motor Inductance per phase in uH */
76 #define  USER_MOTOR_POLE_PAIR                   (7.0f)            /* Motor Pole Pairs */
77 /*      ------------------------------------------------- Constant Speed Control Mode (Used when Constant Speed Control is enabled) ---------------------------------------- */
78 /*      ------------------------------------------------- POT ADC, or PWM to Adjust Speed ---------------------------------------- */
79 #define USER_SPEED_HIGH_LIMIT_RPM               (1000.0f)
80 #define USER_SPEED_LOW_LIMIT_RPM                (uint32_t) (USER_SPEED_HIGH_LIMIT_RPM / 30.0f)
81 #define USER_SPEED_RAMPUP_RPM_PER_S             (500U)
82 #define USER_SPEED_RAMPDOWN_RPM_PER_S           (500U)
83
```

#1

#2

2. CONSTANT SPEED VF_MET_FOC still under development, NOT READY!

#3

- #1: The motor type can be changed according to the motor being used
- #2: The control scheme can also be modified according to user requirements
- #3: The speed of the motor and its ramping can be modified as per the user's requirements

Washing machine motor control - Get started - SW configuration (3/3)

pmsm_foc_user_parameter.h

```
211
212 #define USER_ROTOR_PREPOSITION_TIME_MS              (100U)           /* Rotor startup pre alignment time in miliseconds */
213 #define USER_REFERENCE_SPEED_RPM                    (500U)
214 #define USER_SPEED_THRESHOLD_FW_RPM_PER_S           (7200U)        /* Threshold speed to use Flux Weakening */
215 /*      ------------------------------------------------- MET Fine-tuning ---------------------------------------- */
216 #define USER_MET_THRESHOLD_HIGH                     (64U)
217 #define USER_MET_THRESHOLD_LOW                      (32U)
218 #define USER_MET_LPF                                (2U)
219 /*      ------------------------------------------------- pmsm_foc_parameter.h ----------------------------------- */
220 /* ********************************************* Scaling SVM Modulator ************************************************************/
221 #define PWM_PERIOD_TS_US                              (1.0f/(USER_CCU8_PWM_FREQ_HZ)*1000000)
222 #define SVM_LAMDA                                     (1.0f/USER_INVERSE_SVM_LAMDA)
223 #define SVM_TZ_PZV                                    (uint32_t)((CCU8_PERIOD_REG * SVM_LAMDA) + 0.5f)
224 #define uTZ_LAMDA_TS_US                               (SVM_LAMDA * PWM_PERIOD_TS_US)
225 /* ********************************************* Scaling SVM Modulator ************************************************************/
226 #define KS_SCALE_SVM                                  (CCU8_PERIOD_REG / MAX_VREF_AMPLITUDE)
227 /* ********************************************* ADC Range  ***********************************************************************/
228 #define VAREF_V                                       (5.0f)
229 #define G_OPAMP                                       (USER_R_FEEDBACK_KOHM / USER_RIN_KOHM)
230 #define I_MAX_A                                       ((VAREF_V/ (USER_R_SHUNT_OHM *G_OPAMP)) / 2.0)
231 #define VADC_DCLINK                                   (uint32_t)(((USER_VDC_LINK_V * USER_VBEMF_RATIO)/VAREF_V) * (1<<12))
232 #define VDC_MAX_LIMIT                                 ((uint16_t)((VADC_DCLINK * 19U)>>4U))                    /* Vdc_ideal + 18.7%, DC link voltage Vdc maximum limit */
233 /* ********************************************* Normalization (u,v,w) represented by +2^15 ***************************************/
234 #define N_I_UVW_A                                     I_MAX_A
235 #define N_VREF_SVM_V                                  VREF_MAX_V
236 /* ********************************************* Normalization (alpha,beta) represented by +2^15 **********************************/
237 #define N_I_ALPHABETA_A                               I_MAX_A
238 #define N_VREF_ALPHABETA_V                            VREF_MAX_V
239 /* ********************************************* Normalization (d,q) represented by +2^15 ******************************************/
240 #define N_I_DQ_A                                      I_MAX_A
241 #define N_V_DQ_V                                      VREF_MAX_V
242 /* ********************************************* Motor Control Timing   ***********************************************************/
243 #define BOOTSTRAP_BRAKE_TIME                          ((USER_BOOTSTRAP_PRECHARGE_TIME_MS * 1000) / PWM_PERIOD_TS_US)
244 #define PRE_ALIGNMENT_TIME                            (uint32_t)(((USER_ROTOR_PREPOSITION_TIME_MS * 1000) / PWM_PERIOD_TS_US))
245 /* ********************************************* Amplitude Limit ******************************************************************/
246 #define MAX_VREF_AMPLITUDE                            (32768 - 0.5f)
247 #define VREF_MAX_V                                    (USER_VDC_LINK_V / USER_SQRT_3_CONSTANT)
248 /* ********************************************* V/F Startup Parameter  ***********************************************************/
249 #define STARTUP_CURRENT_A                             (uint32_t)((USER_STARTUP_VF_OFFSET_V / USER_MOTOR_R_PER_PHASE_OHM))
250 #define STARTUP_SPEED                                 (uint32_t)(((USER_STARTUP_SPEED_RPM * USER_MOTOR_POLE_PAIR)/(USER_CCU8_PWM_FREQ_HZ*60))*65536 * (1<< USER_RES_INC))
251 #define STARTUP_SPEED_THRESHOLD                       (uint32_t)(((USER_STARTUP_SPEED_THRESHOLD_RPM * USER_MOTOR_POLE_PAIR)/(USER_CCU8_PWM_FREQ_HZ*60))*65536 * (1<< USER_RES_INC))
252 #define STARTUP_VF_OFFSET                             (uint32_t)((USER_STARTUP_VF_OFFSET_V * 32768) / (N_VREF_SVM_V))
253 #define STARTUP_VF_SLEWRATE                           (uint32_t)(((USER_STARTUP_VF_SLEWRATE_V_PER_HZ * 32768) / (N_VREF_SVM_V) * (USER_MOTOR_POLE_PAIR))/(USER_CCU8_PWM_FREQ_HZ*60)*65536)
254 /* ********************************************* Constant Speed Control Mode (Used when Constant Speed Control is enabled)  ***********************************************************/
255 /* ********************************************* POT ADC, or PWM to Adjust Speed  *************************************************/
256 #define SPEED_LOW_LIMIT_RPM                           (uint32_t)(((USER_SPEED_LOW_LIMIT_RPM * USER_MOTOR_POLE_PAIR) /(USER_CCU8_PWM_FREQ_HZ*60))*65536 * (1<< USER_RES_INC))
257 #define REFERENCE_SPEED_USER                          (uint32_t)(((USER_REFERENCE_SPEED_RPM * USER_MOTOR_POLE_PAIR) /(USER_CCU8_PWM_FREQ_HZ*60))*65536 * (1<< USER_RES_INC))
```

Washing machine motor control - Get started - pmsm_foc_user_parameters.h

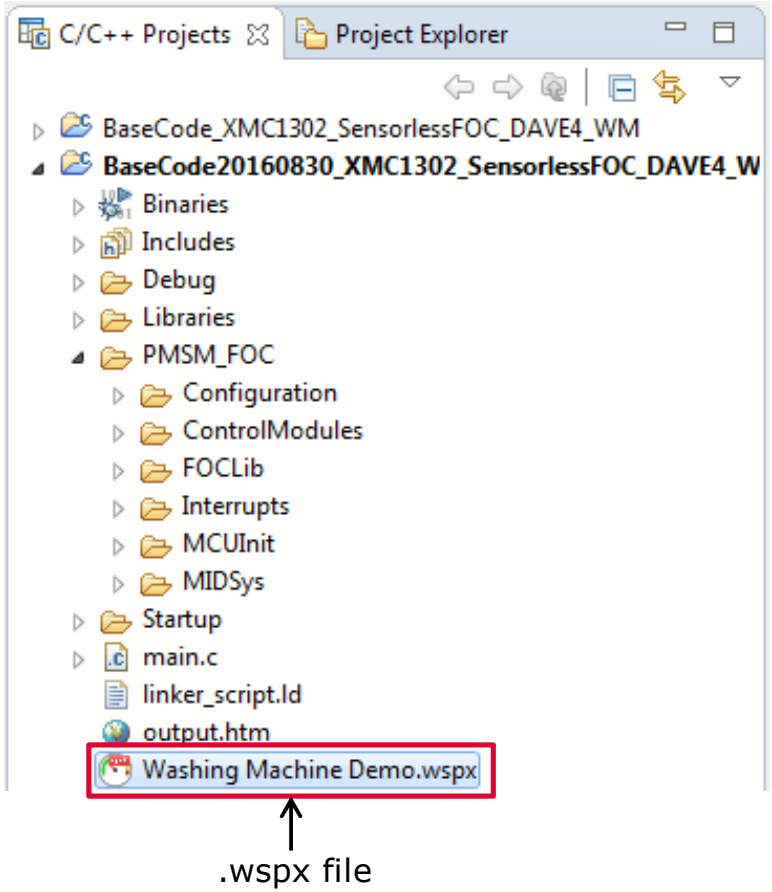
- XMC™ can use fixed points numbers / integers to represent floating-point quantities of the physical value (e.g. : in SI unit)

- User can defined different level of configurations (beginner level, intermediate level, advanced level)

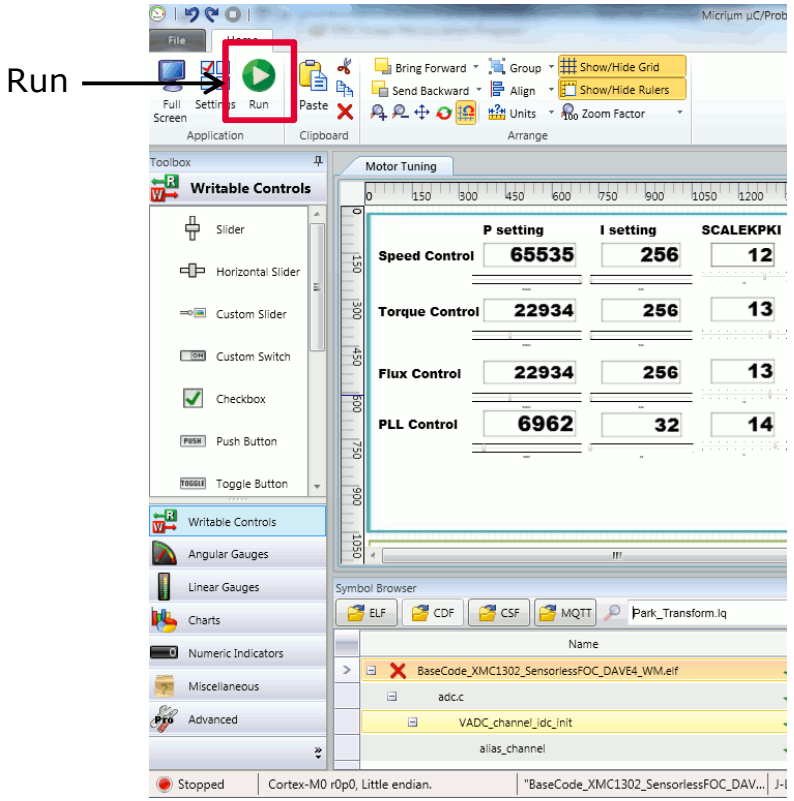
Washing machine motor control - Get started - pmsm_foc_user_mcuhwconfig.h

- MCU hardware resource management (VADC, CCU8)
- NVIC interrupts service routine resource management
- Debugging IO (DAC functionality)

Washing machine motor control - Get started - starting µC/Probe™ XMC™

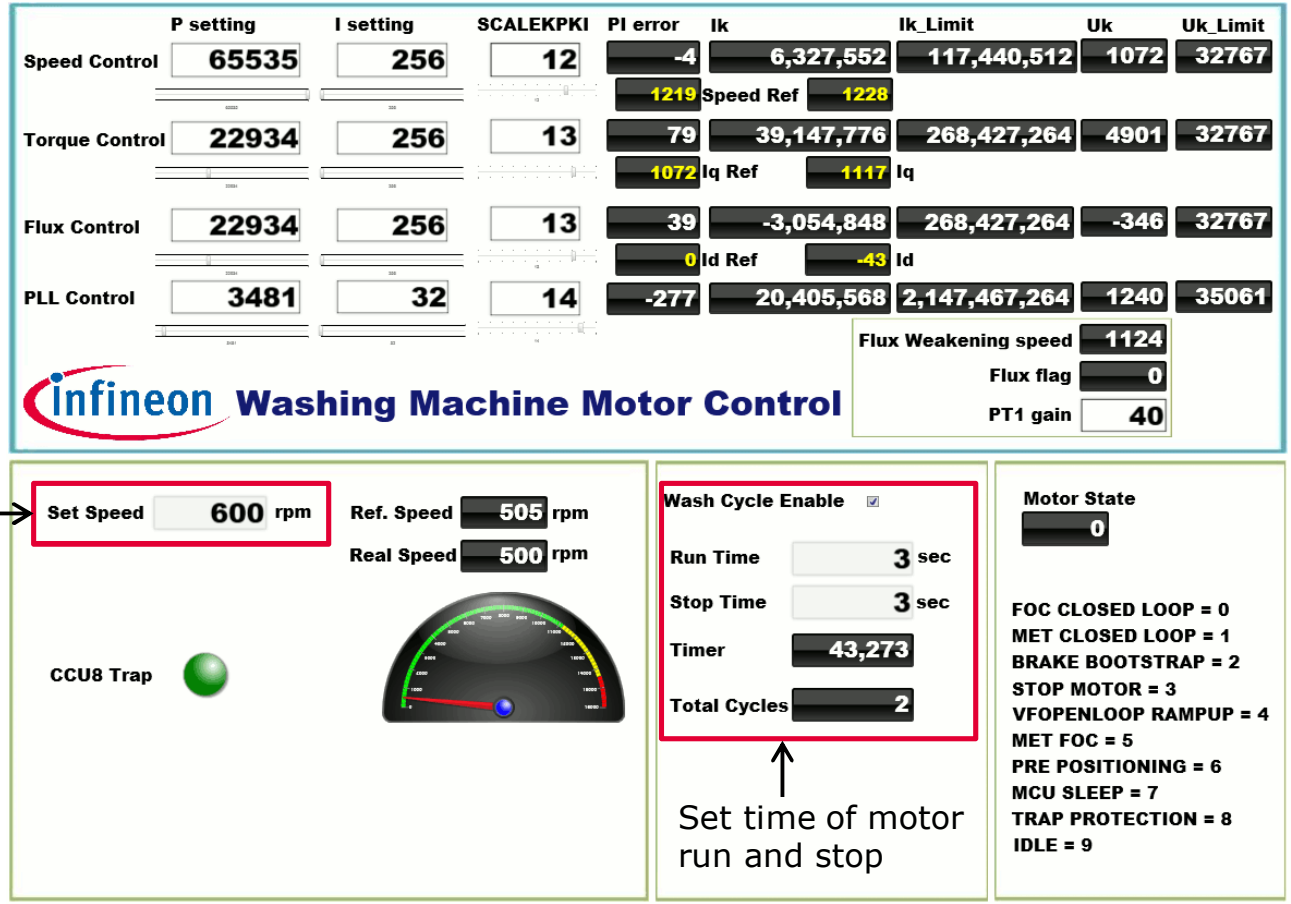


- Double-click "***.wspx**" file in the DAVE™ 4 IDE to start µC/Probe™
- Click "**Run**" to control the speed of the motor using µC/Probe™



Washing machine motor control - Get started - start motor using µC/Probe™

- The motor can be started by keying in a number in rpm (e.g.: 600) in the "**Set Speed**" box

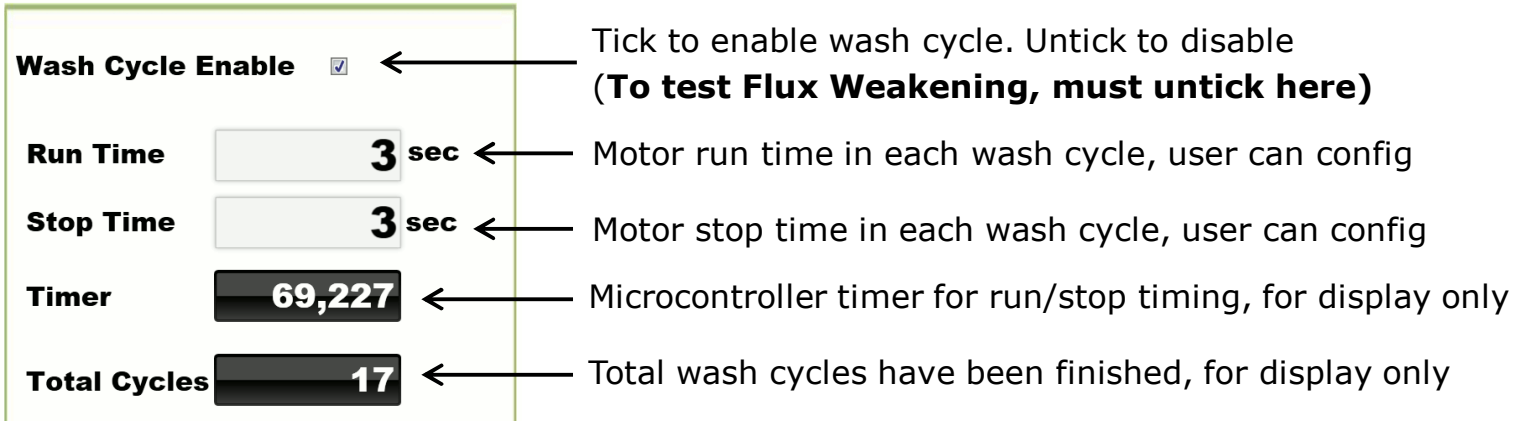


Washing machine motor control - Get started - wash cycle set using µC/Probe™

- Motor run time / motor stop time in each wash cycle can be modified on-the-fly at µC/Probe™
- Default timing configured at **pmsm_foc_user_parameter.h**

```
#define  USER_MOTOR_RUN_TIME_S      (5U)    /* Motor run time in second*/
#define  USER_MOTOR_STOP_TIME_S     (3U)    /* Stop time in second*/
#define  USER_ENABLE_WASHING_CYCLE  (1U)    /* 1U: Enable washing (wash/stop/wash), 0U: disable*/
```

- Total wash cycles undergone successfully is counted by XMC™, and displayed by µC/Probe™



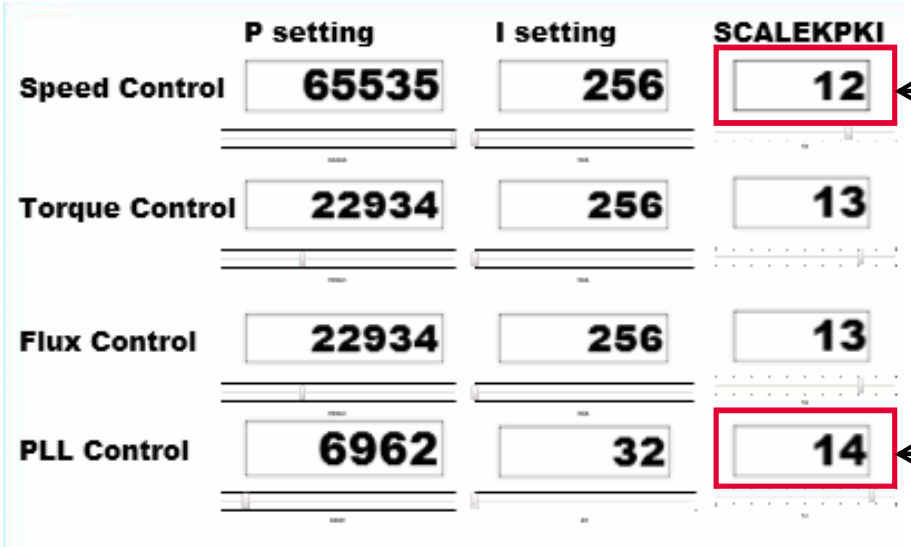
Washing machine motor control - Get started - fine-tune Kp/Ki using µC/Probe™

- If the motor does not spin in FOC close loop, ↑ the SCALEKPKI of PLL Control and check the motor behavior. If motor start to move slowly, ↑ the SCALEKPKI further, else, ↓ the SCALEKPKI
- Apply similar tactic for the tuning of Speed Control

PI gains: $K_p = \frac{P\ setting}{2^{SCALEKPKI}}$, $K_i = \frac{I\ setting}{2^{SCALEKPKI}}$

2

1

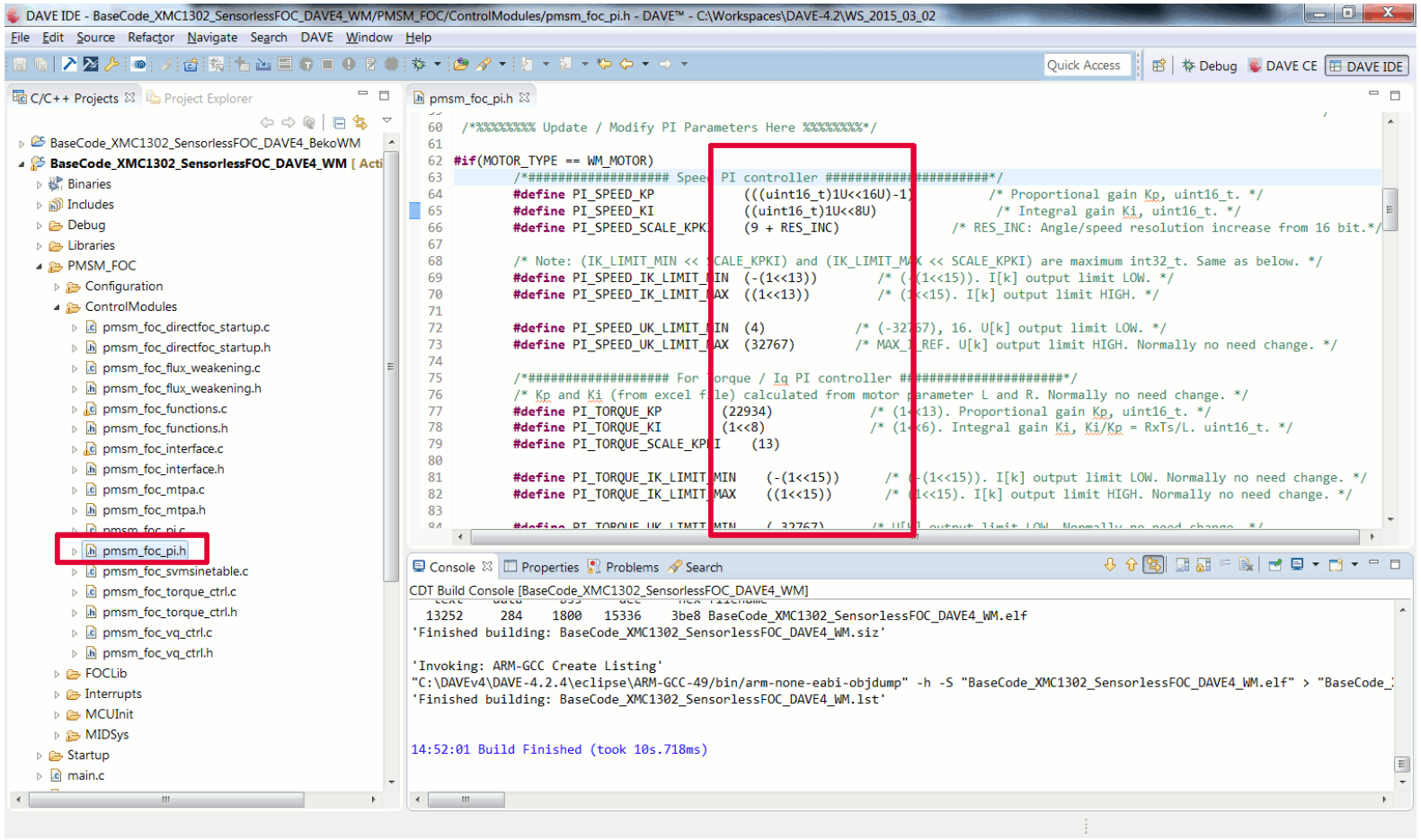


↑ this value by 1 will ↓ gain of Speed controller by half

↑ this value by 1 will ↓ gain of PLL estimator controller by half

Washing machine motor control - Get started - pmsm_foc_pi.h

- In DAVE™ 4, user needs to input /save the final optimal PI parameters to **pmsm_foc_pi.h**

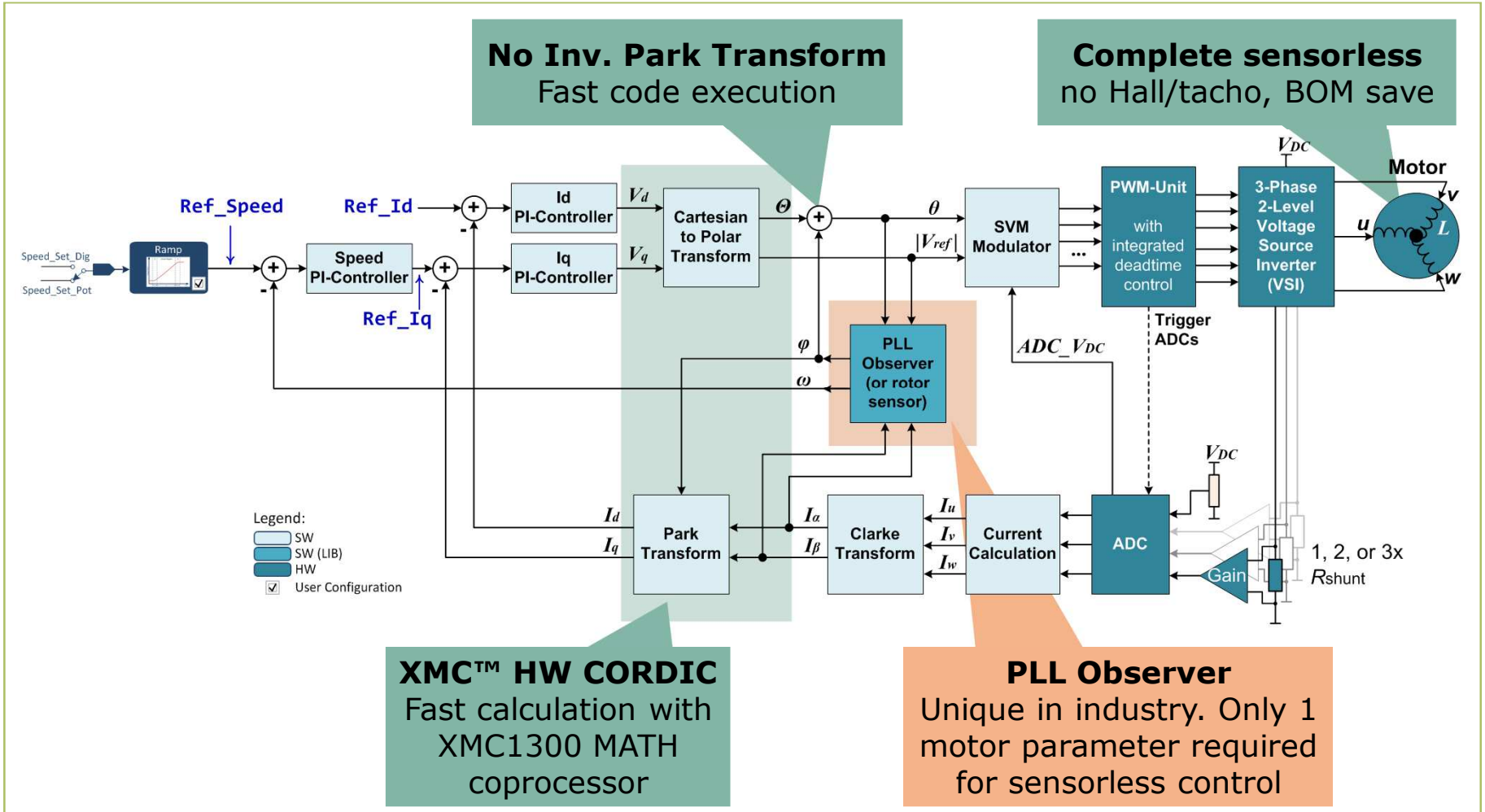


Washing machine motor control - Key features

1. Key feature - sensorless FOC control even at ultra-low speed
2. Key feature - robust direct-sensorless-FOC startup
3. Key feature - smooth bi-directional control at low-speed
4. Key feature - Flux Weakening (FW) at high speed for spin dry
5. Key feature - XMC™ on-chip ADC gain to reduce system BOM cost

1. Key feature - sensorless FOC control even at ultra-low speed

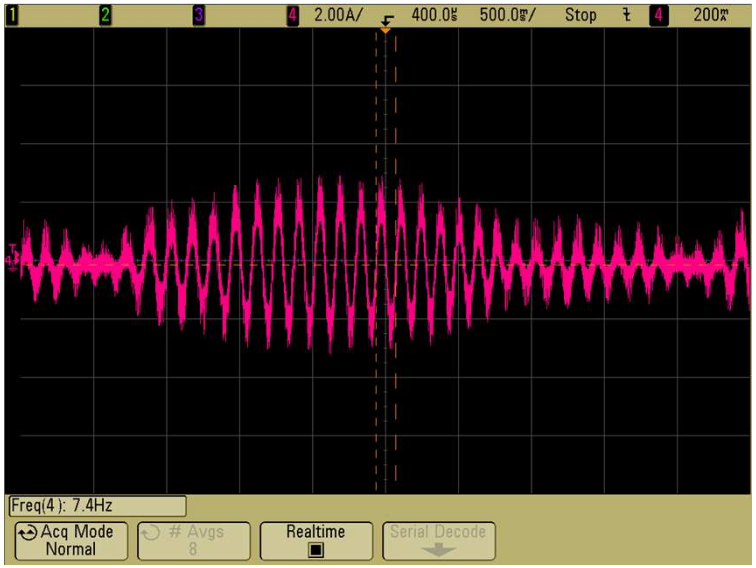
Washing machine - sensorless FOC - Block diagram



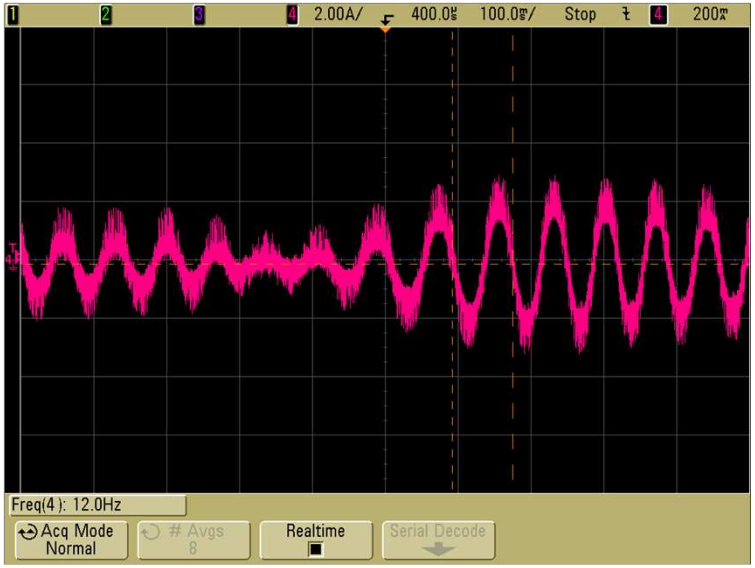
Block diagram: washing machine motor control - sensorless FOC

Sensorless FOC control at ultra-low speed - Waveforms

- Ultra-low speed: 100 rpm



200 rpm



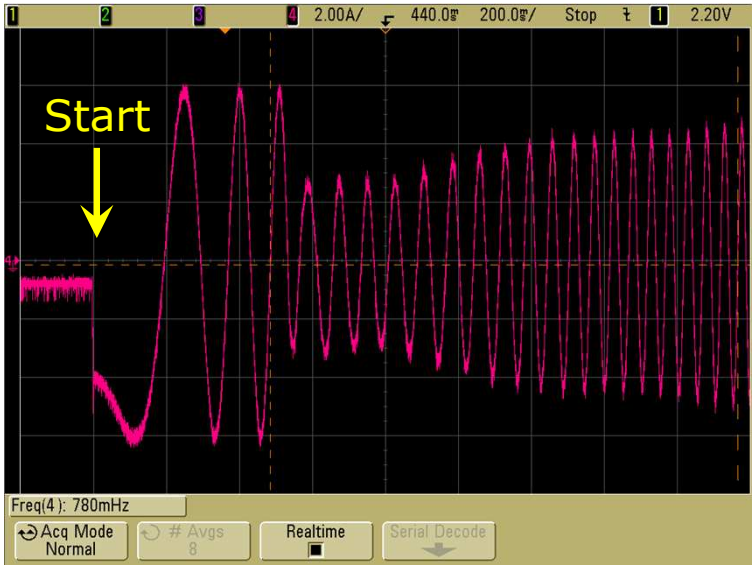
- CH4 (Pink) - phase current Iu, from current probe (2 A/div)

Motor parameter:
L (per phase): 14 *m*H
R (per phase): 2.5 *Ω*
Pole-pair No.: 4

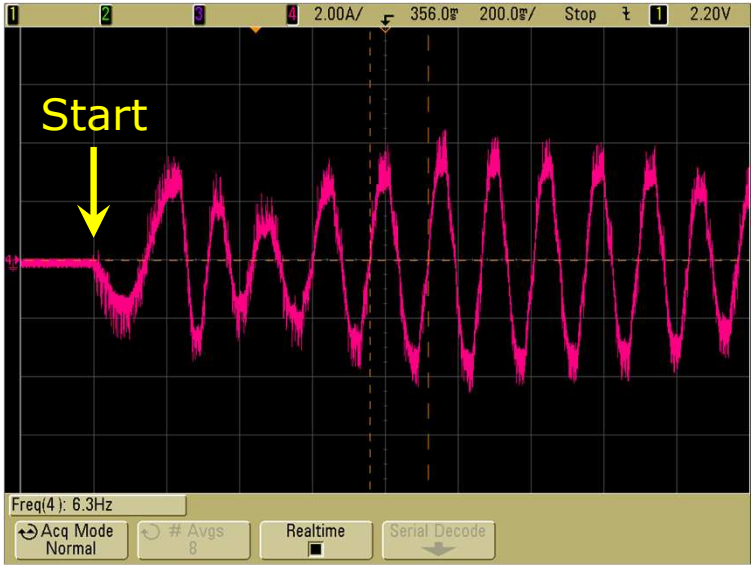
2. Key feature - robust direct-sensorless-FOC startup

Startup with load @ quilt + 37 kg water - Waveforms

› Competitor solution



XMC™ sensorless FOC

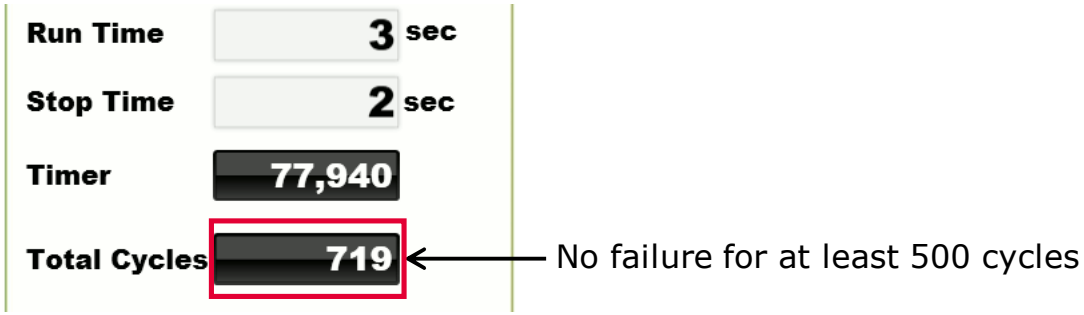


› CH4 (Pink) - phase current Iu, from current probe (2 A/div)

Motor parameter:
L (per phase): 14 *m*H
R (per phase): 2.5 *Ω*
Pole-pair No.: 4

Washing machine - sensorless FOC - Robustness of direct-sensorless-FOC startup

- No failure in more than 500 wash cycles for motor startup bi-directionally at various load conditions



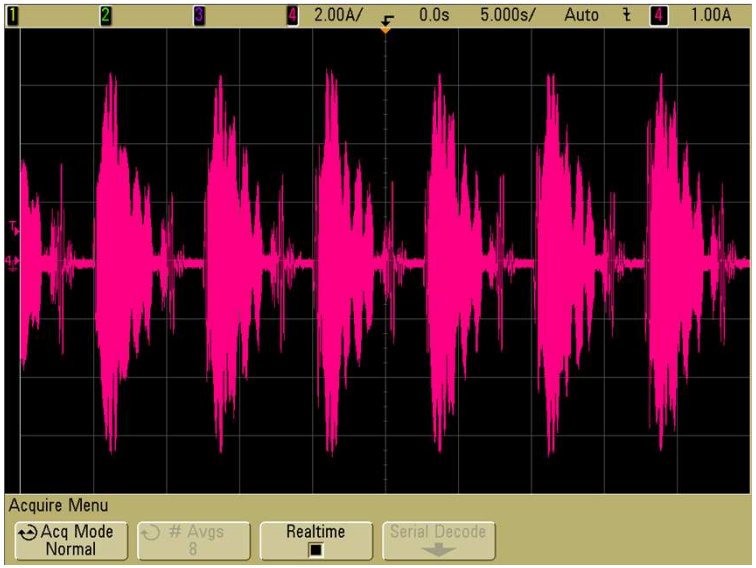
- During the test, one wash cycle is: … → motor start clockwise → washing → motor stop → motor start anticlockwise → washing → motor stop → …

3. Key feature - smooth bi-directional control

Wash cycle @ 600 rpm, quilt + 38 kg water - Waveforms

- Wash cycles

Direct-FOC-startup each time

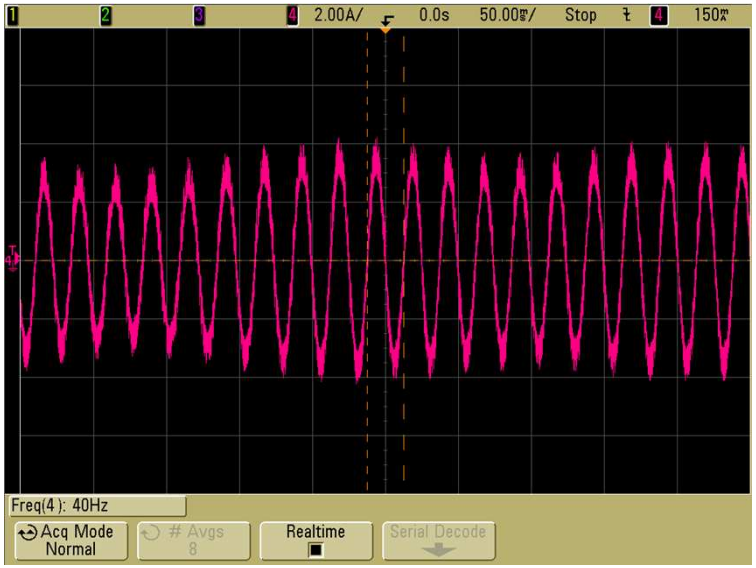


- CH4 (Pink) - phase current Iu, from current probe (2 A/div)

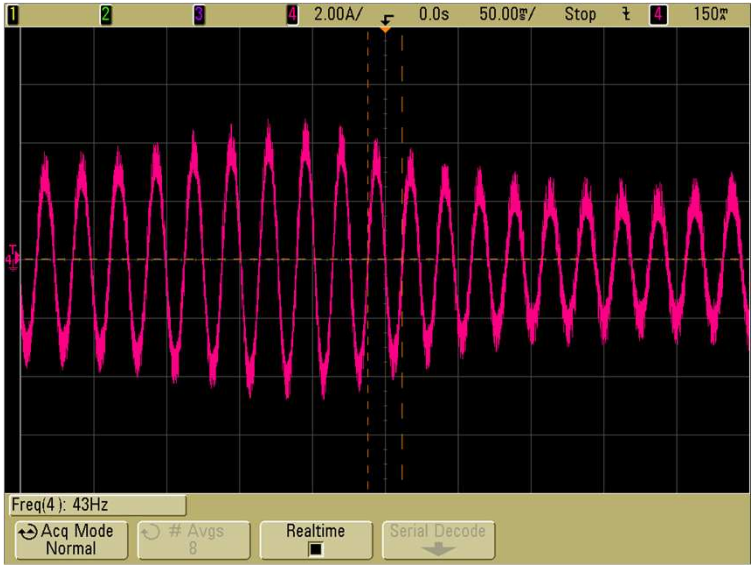
Motor parameter:
L (per phase): 14 *m*H
R (per phase): 2.5 *Ω*
Pole-pair No.: 4

Washing @ 600 rpm, quilt + 38 kg water - Waveforms

- 600 rpm (quilt + 38 kg water)



600 rpm (quilt + 50 kg water)



- CH4 (Pink) - phase current Iu, from current probe (2 A/div)

Motor parameter:
L (per phase): 14 *m*H
R (per phase): 2.5 *Ω*
Pole-pair No.: 4

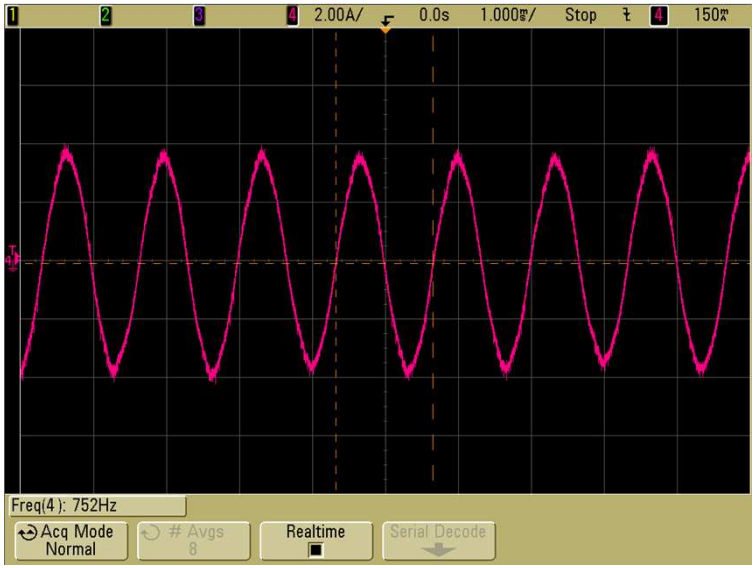
4. Key feature - Flux Weakening (FW)

Safety precautions

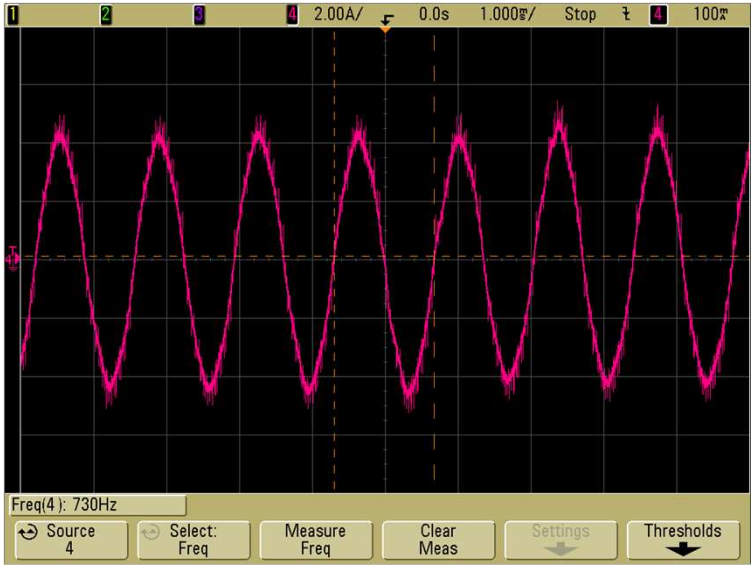
- **ATTENTION:** Washing machine motor phase-to-phase BEMF is kilovolts (kV) at high speed (e.g.: 1.6 kV_{p-p} @ 12,000 rpm motor mechanical speed)
- Any loss of control at Flux Weakening (FW) range can result in equipment / board / component damage
- Only personnel familiar with the advanced motor control / FW should implement and test FW

Washing machine - Flux Weakening (FW) - Waveforms

› Competitor solution 11,280 rpm



XMC™ FOC 11,000 rpm

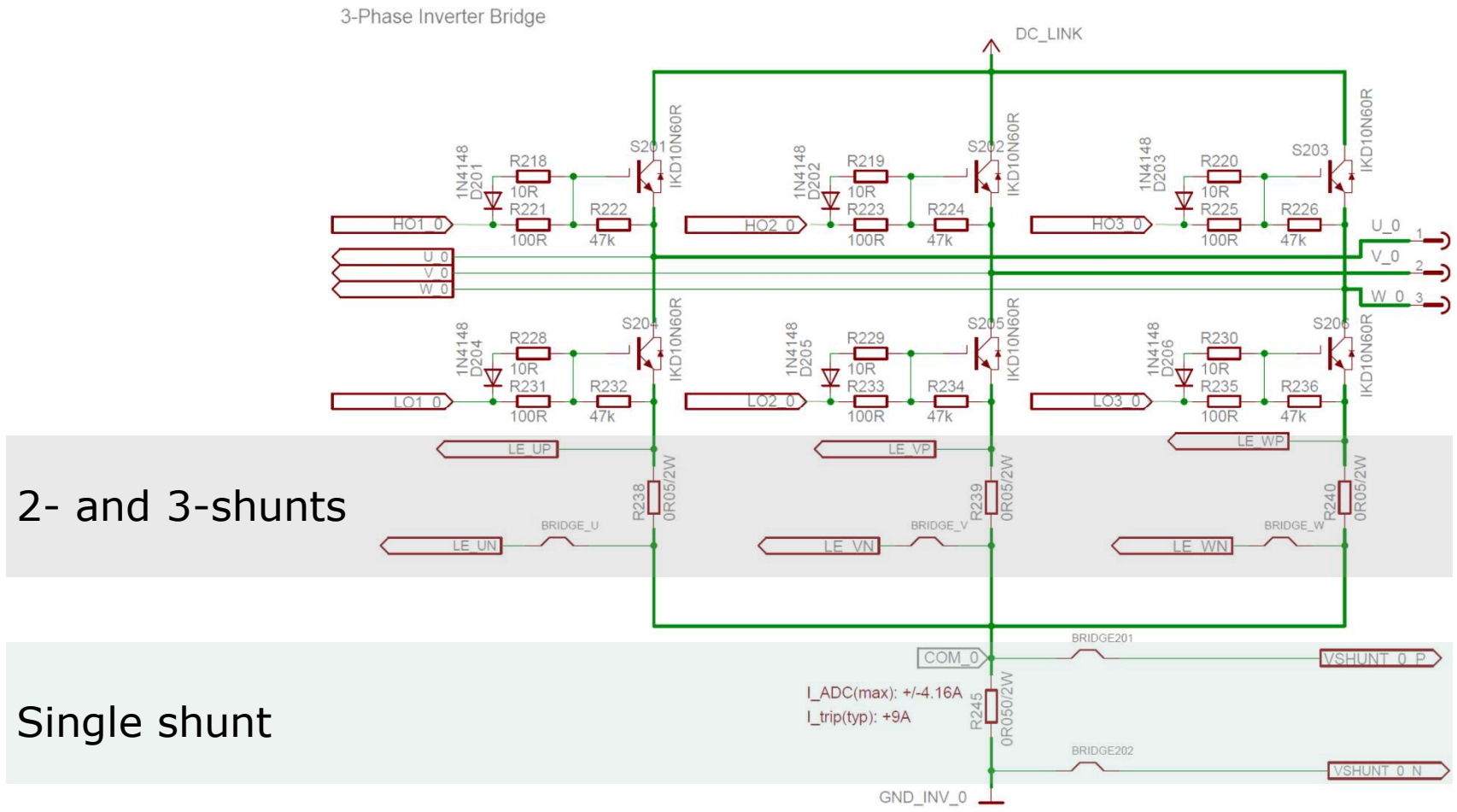


› CH4 (Pink) - phase current Iu, from current probe (2 A/div)

Motor parameter:
L (per phase): 14 *m*H
R (per phase): 2.5 *Ω*
Pole-pair No.: 4

5. Key feature - XMC™ on-chip ADC gain

Washing machine motor control - Current sensing schematics

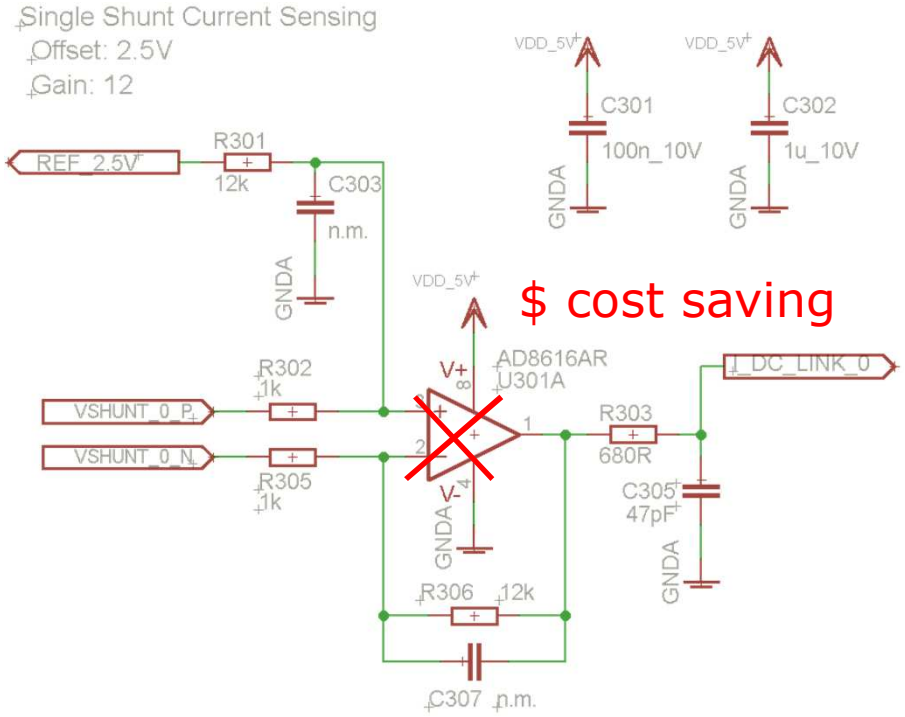


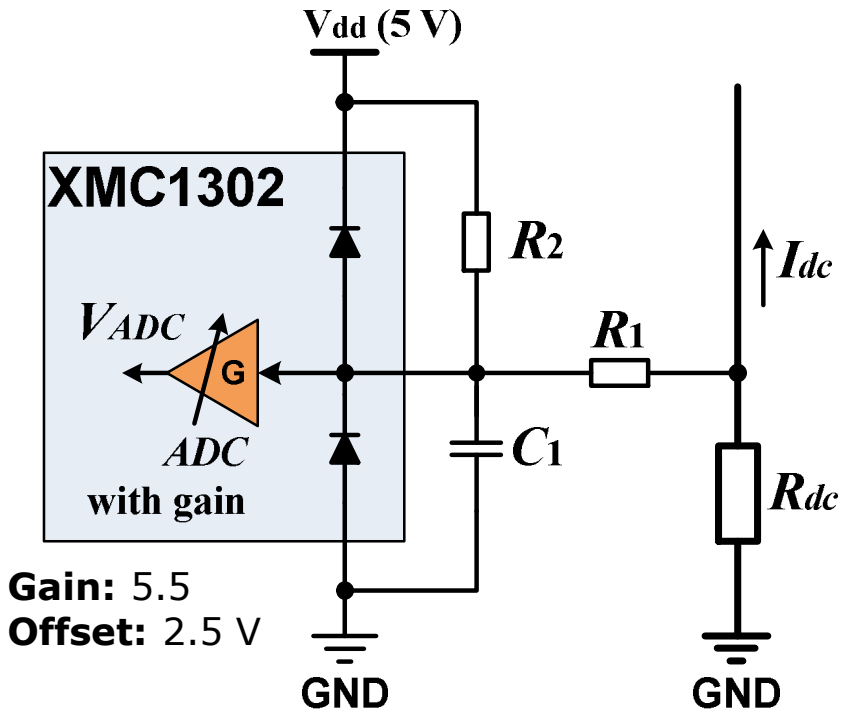
Schematic: washing machine motor control - current sensing

Washing machine motor control - XMC™ on-chip ADC gain for current sensing

- R_1 limits current flow in / out of XMC1302 ADC pin. R_2 offset ADC input
- e.g.: for an application G=6, R_{dc}=0.05 Ω, R_1=1 kΩ, R_2=11 kΩ, C_1=47 pF

$$V_{ADC} \approx \frac{GR_1}{R_1 + R_2} V_{dd} + \frac{R_2}{R_1 + R_2} G \cdot R_{dc} \cdot I_{dc}$$

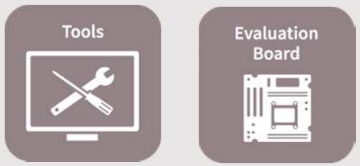




Resource listing

- Washing machine motor control
  - Documentation
  - XMC™ 750 watt motor control application kit
  - DAVE™ project

Support material:

Collaterals and Brochures 	– Product Briefs – Selection Guides – Application Brochures – Presentations – Press Releases, Ads	– www.infineon.com/XMC
Technical Material 	– Application Notes – Technical Articles – Simulation Models – Datasheets, MCDS Files – PCB Design Data	– www.infineon.com/XMC – Kits and Boards – DAVE™ – Software and Tool Ecosystem
Videos 	– Technical Videos – Product Information Videos	– Infineon Media Center – XMC Mediathek
Contact 	– Forums – Product Support	– Infineon Forums – Technical Assistance Center (TAC)

Glossary abbreviations (1/2)

- AC Alternating Current
- ADC Analog-to-Digital Converter
- BEMF Back ElectroMotive Force
- BOM Bill Of Material
- CPU Central Processing Unit
- DAC Digital-to-Analog Converter
- DAVE™ Digital Application Virtual Engineer
- DC Direct Current
- FOC Field-Oriented Control
- GUI Graphical User Interface
- HMI Human-Machine Interface
- HW Hardware

Glossary abbreviations (2/2)

- IDE — Integrated Development Environment
- IGBT — Insulated-Gate Bipolar Transistor
- MCU — MicroController Unit
- PLL — Phase-Locked Loop
- PMSM — Permanent Magnet Synchronous Motor
- PWM — Pulse Width Modulation
- RAM — Random-Access Memory
- SW — Software
- SWD — Serial Wire Debug
- UART — Universal Asynchronous Receiver / Transmitter
- USIC — Universal Serial Interface Channel
- XMC™ — Cross-Market Microcontrollers

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.

Infineon