# Drone
# Electronic Speed Controller(ESC) with XMC™, Gate Driver, OptiMOS™

October 2016

# Learning objectives

- To demonstrate the implementation of sensorless FOC as Electronic Speed Controller (ESC) in quadcopter application
- Key software functions, a step-by-step implementation, and linking up with µC/Probe™ XMC™
- To use of µC/Probe™ to visualise data and fine-tune ESC
- After the learning of this PPT, users will be able to fine-tune FOC example software which is easy scalability for quadcopter ESC applications

# Agenda (1/2)

# Agenda (2/2)

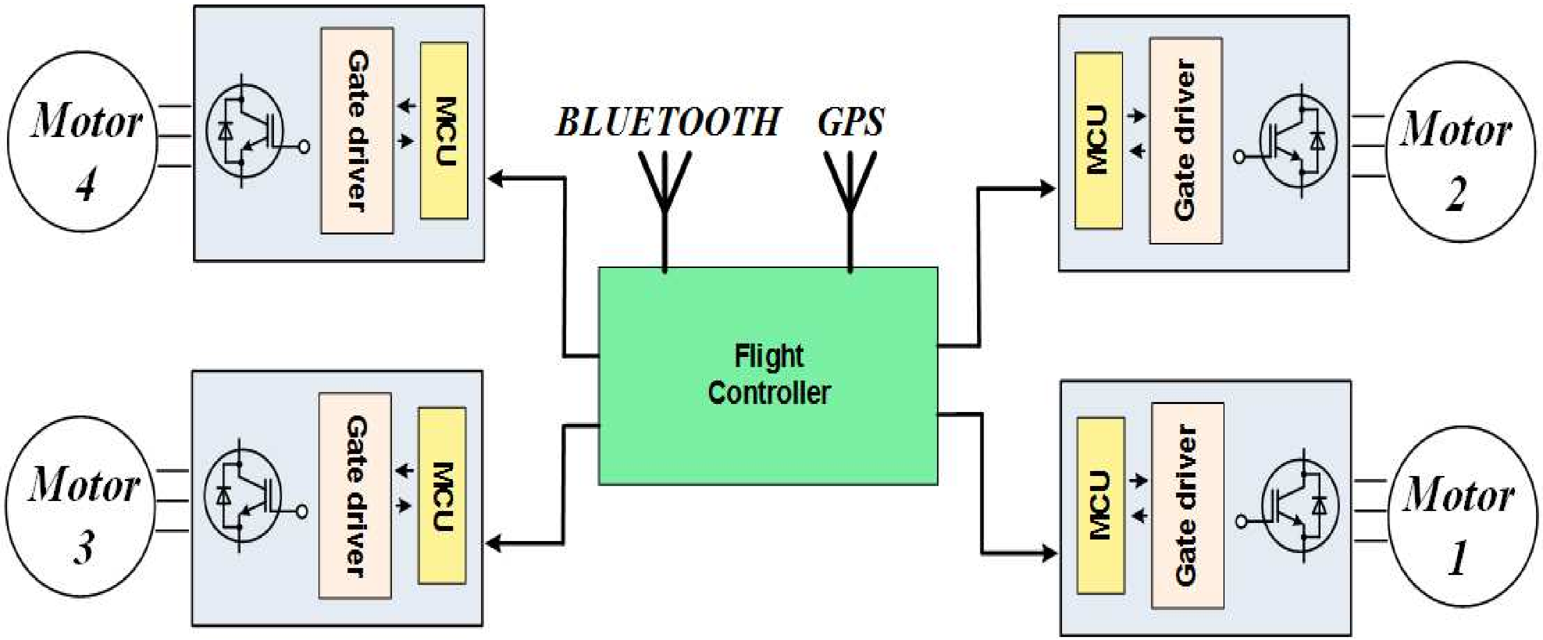
9 Resource listing

# Electronic Speed Controller Reference Design-Overview

- The purpose of the training slides is to elaborate a low-cost and high-performance quadcopter ESC solution, using
  - XMC™ PINUS Inverter
  - 12V 2300KV 3 phase brushless motor
  - DAVE™ 4 ESC example project - PMSM_FOC_SL_XMC13
- The HOT examples cover the key features and controls of the 3 phase brushless drone motor

# Quadcopter System Block

› Consists of a flight controller and four ESCs, one for each motor.




› Each ESC contains a three phase inverter driven by MCU with a specialized FOC motor control software

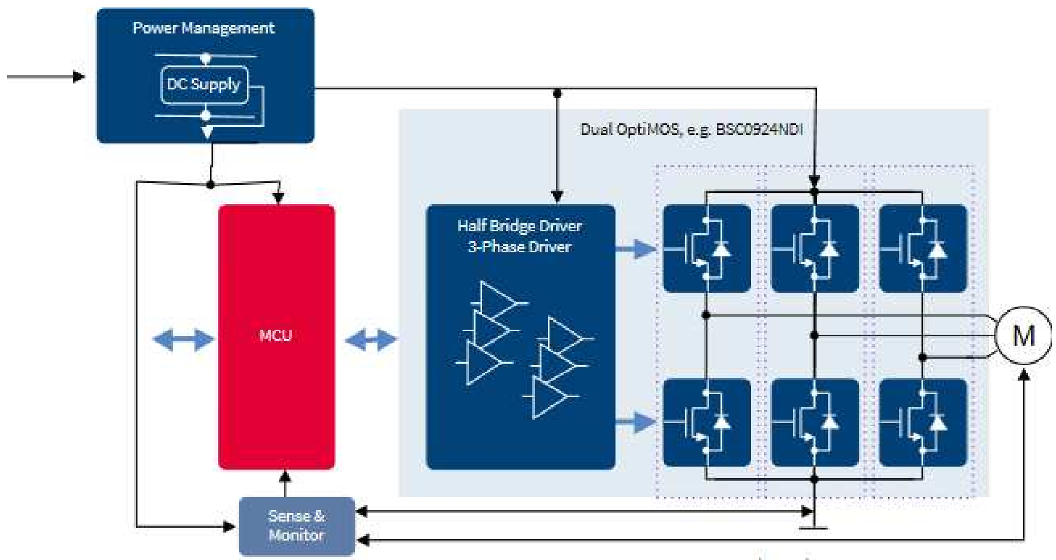
# Infineon can provide all the critical components for quadcopter ESC Reference Design

## Motor control

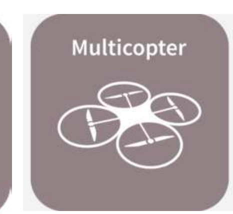
| Functionality | IFX components |
|---|---|
| › FOC controller (3-phase PWM generation, motor phase current sensing, bus voltage sensing, over-current & over-voltage protection) | › **XMC1302**: ARM® Cortex®-M0 32-bit processor @ 32 MHz, up to 200 kB flash,16 kB SRAM,MATH coprocessor, 12-bit ADC with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

## Power stage

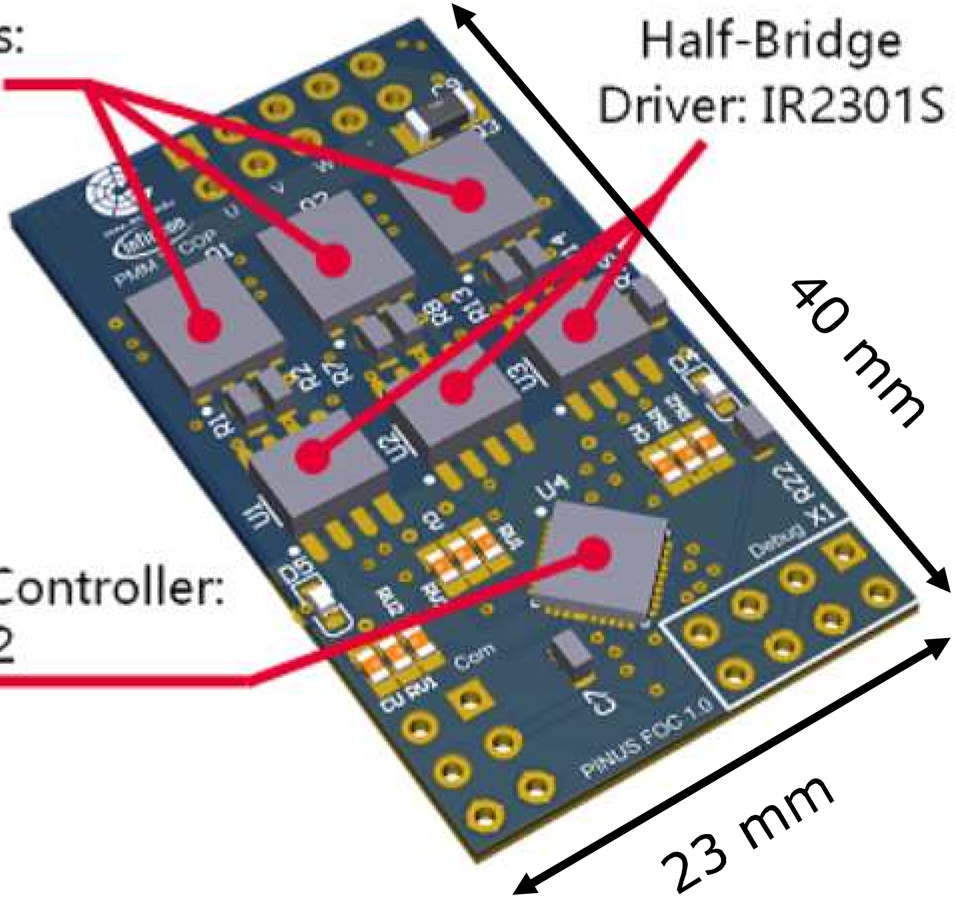
| Functionality | IFX components |
|---|---|
| › Power inverter | › **BSC0925ND**: OptiMos Power-MOSFET Dual N-channel<br>› **IR2301S:** High and Low Side gate driver |
| › Auxiliary power supply | › **IFX91041V50**: 5 V LDO (Coming from Flight Controller) |





# ESC Reference Design 3D PCB view



# Value arguments of ESC Reference Design

| | Benefits | Addressed customer needs |
|---|---|---|
| **Strong arguments** | Small code size and fast execution time | Allows ample CPU time for more tasks<br>Small code size < 16 kB, and super-fast code execution < 20 $\mu$s (for optimized code) |
| | Robust start-up | Direct-sensorless-FOC startup is robust, smooth and energy efficient at various load conditions of ESC |
| | 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 |
| **Medium arguments** | ARM® Cortex®-M0 with MATH coprocessor | Replace costly MCUs (e.g. ARM® Cortex®-M3) from competitors |
| | Embedded security solutions | Protect customer solution from being copied by competitors |
| **Soft arguments** | 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 |

# ESC Reference Design - Key features

## Target application

- Quadcopter application

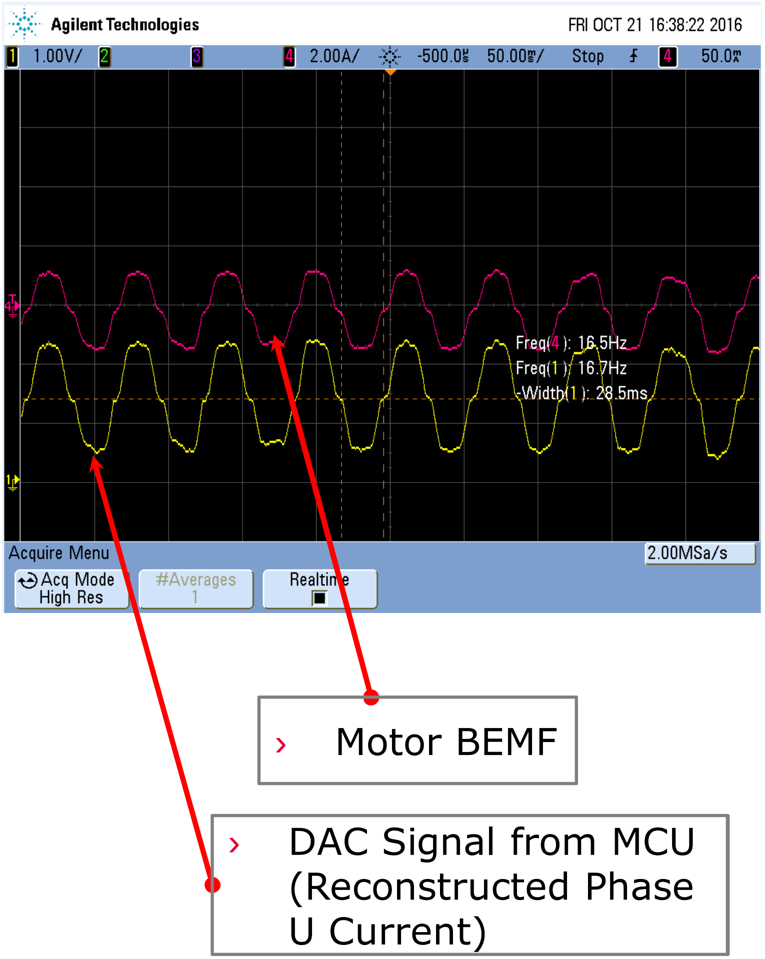
## Key features

- Sensorless FOC control even at ultra-low speed
- Robust direct-sensorless-FOC startup
- Fast execution FOC with XMC™ Cortex M0
- XMC™ on-chip ADC gain to reduce system BOM cost

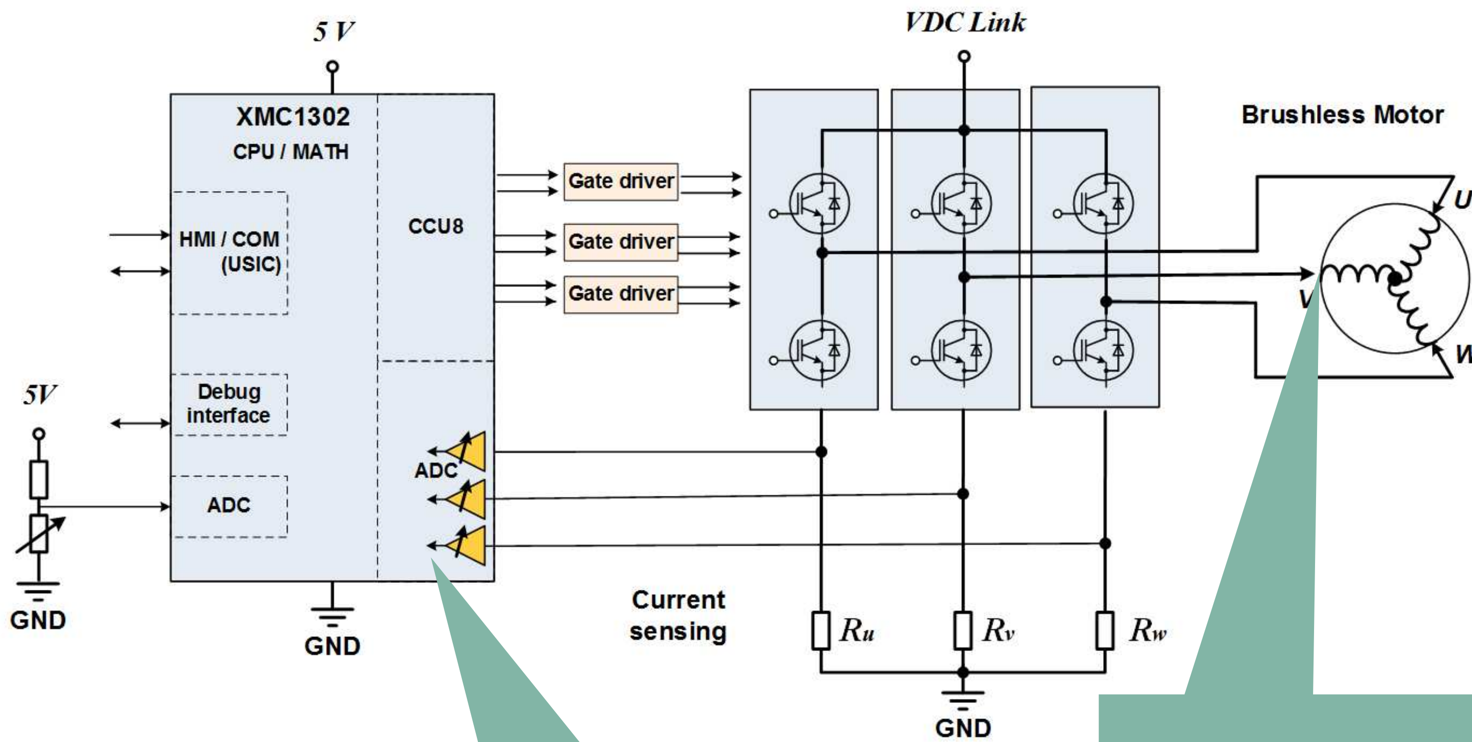
# ESC Reference Design - Specification

## Specifications

- DC input voltage: 12Vdc and 5Vdc (Coming from Flight Controller)
- Motor mechanical speed
  - Maximum speed 15,000 rpm
  - Motor Pole Pair: 6
  - KV = 2300
  - Max Continuous current (A) = 5.7A
  - Max Continuous Power (W) = 63.3W
- Speed adjustable via Potentiometer / uC Probe GUI / UART Communication



# ESC Reference Design - block diagram




**XMC™ on-chip ADC gain**
BOM save

**Complete sensorless**
no Hall/tacho, BOM save

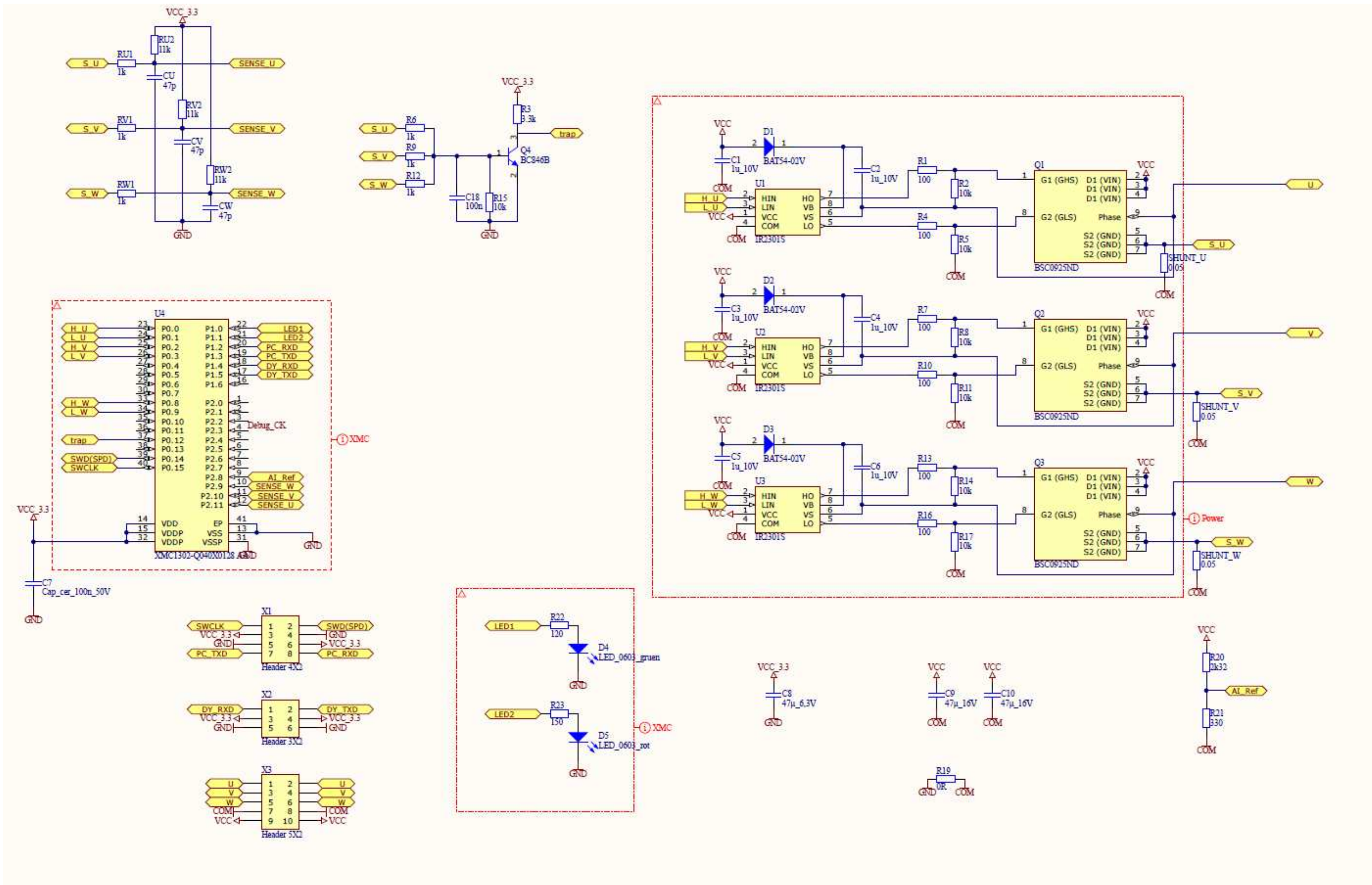
ESC block diagram: ESC motor control

# ESC motor control - Hardware overview

- Key Infineon components utilized on PINUS Inverter:

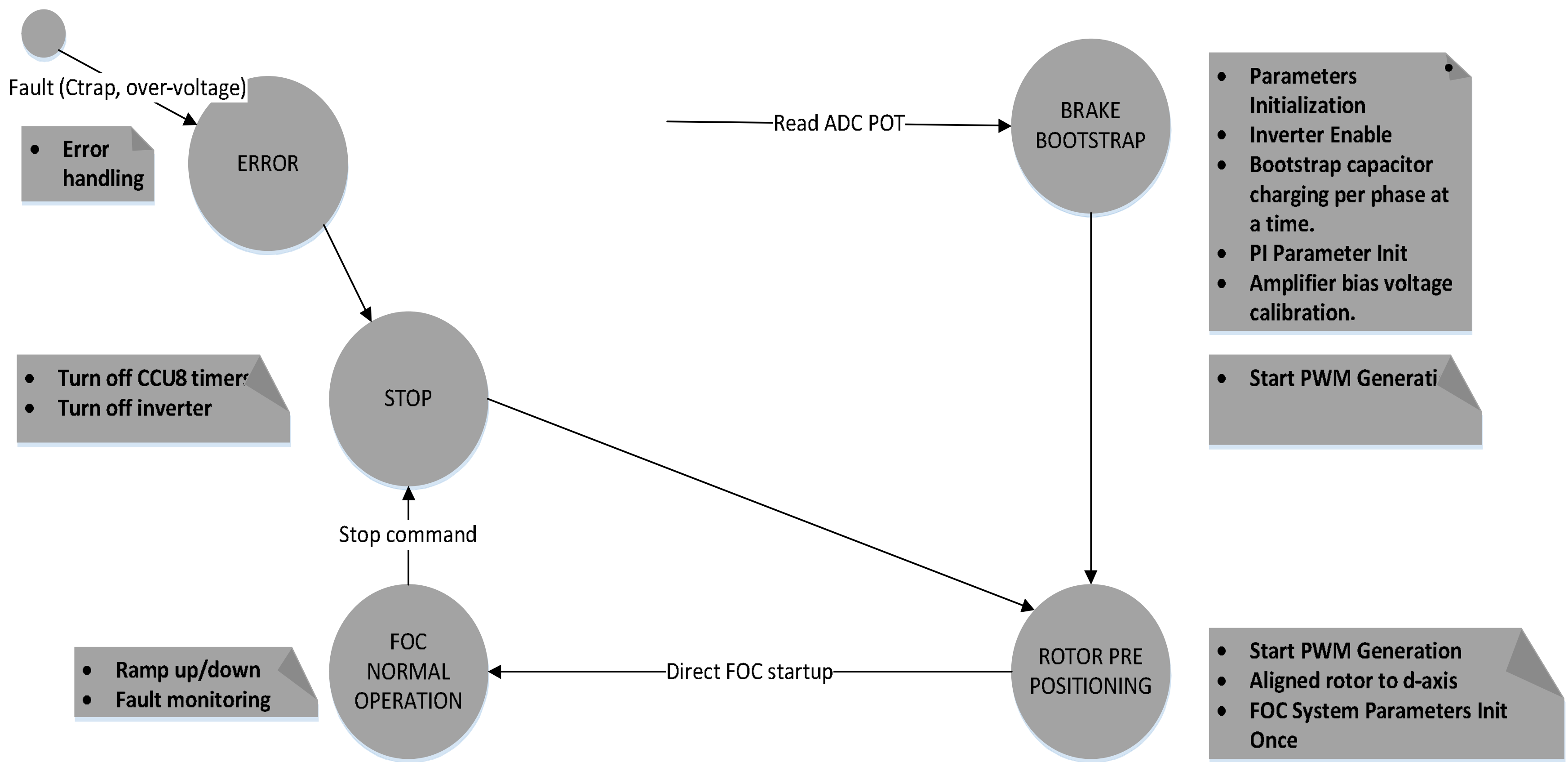
| No. | Infineon components | Order number |
| --- | --- | --- |
| 1 | XMC™ microcontroller | XMC1302-Q040X0128 |
| 2 | Gate Driver IC | IR2301S |
| 3 | OptiMOS™ Power MOSFET, 30V, Dual N Channel | BSC0925ND |

# ESC motor control -
# 3-phase PINUS inverter schematics




Schematic: ESC motor control - 3-phase inverter

# ESC motor control - Software overview (Torque Control Mode)




Flow chart: ESC motor control - software overview

# ESC 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 $\mu$s
  - 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

# ESC motor control - Get started - HW connections

› Connect BLDC motor U, V and W phases to PINUS Inverter

› Connect POT P2.3

› Connect to Jlink Debugger SC

› VDC Link (12V)

› Connect to Jlink Debugger SD

› GND

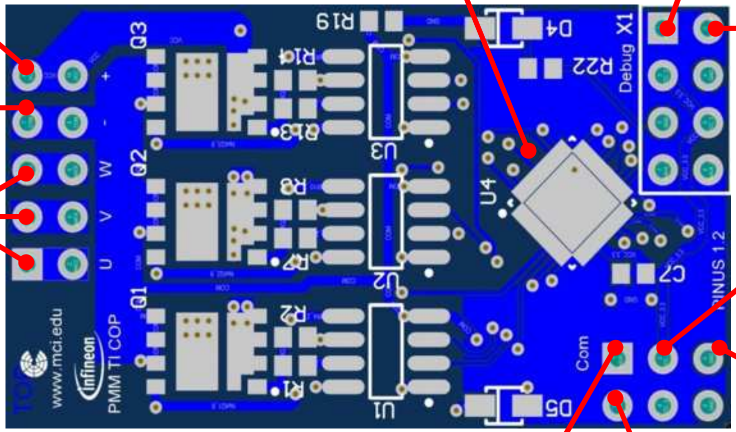
› Connect to Drone BLDC Motor U,V,W

› Supply +5V

› GND

› UART RX P1.4

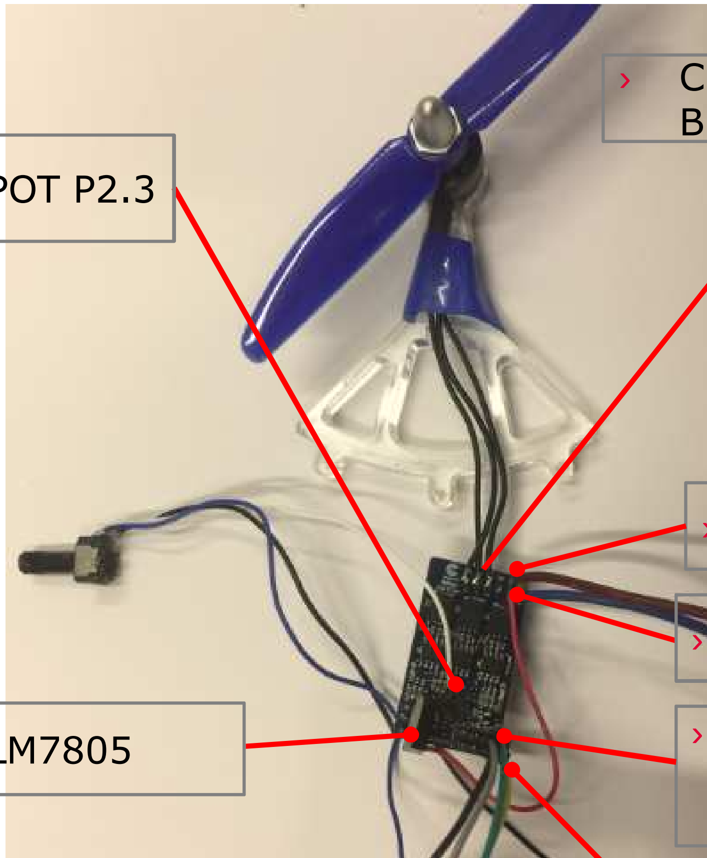
› UART TX P1.5

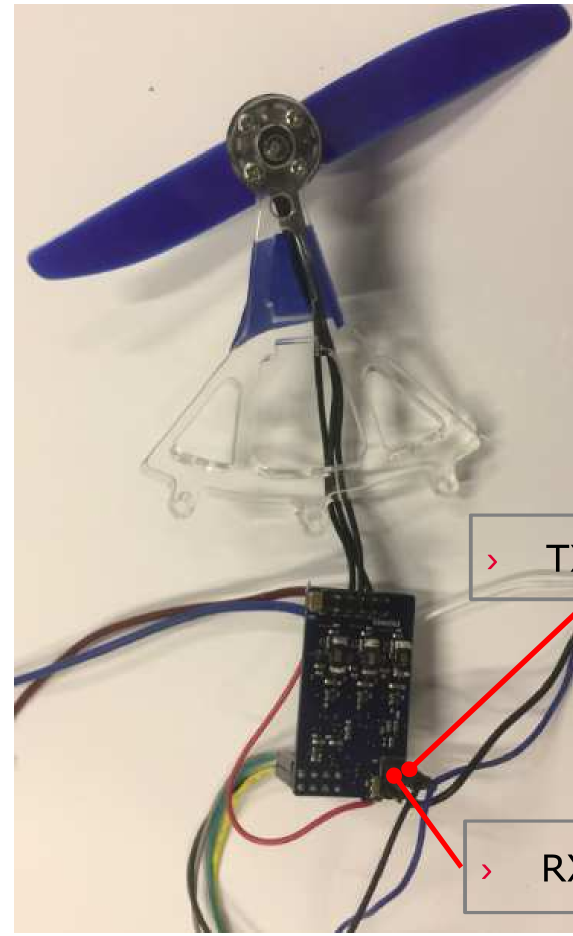


# ESC motor control -
# Get started – Real HW connections

› UART RX P1.4

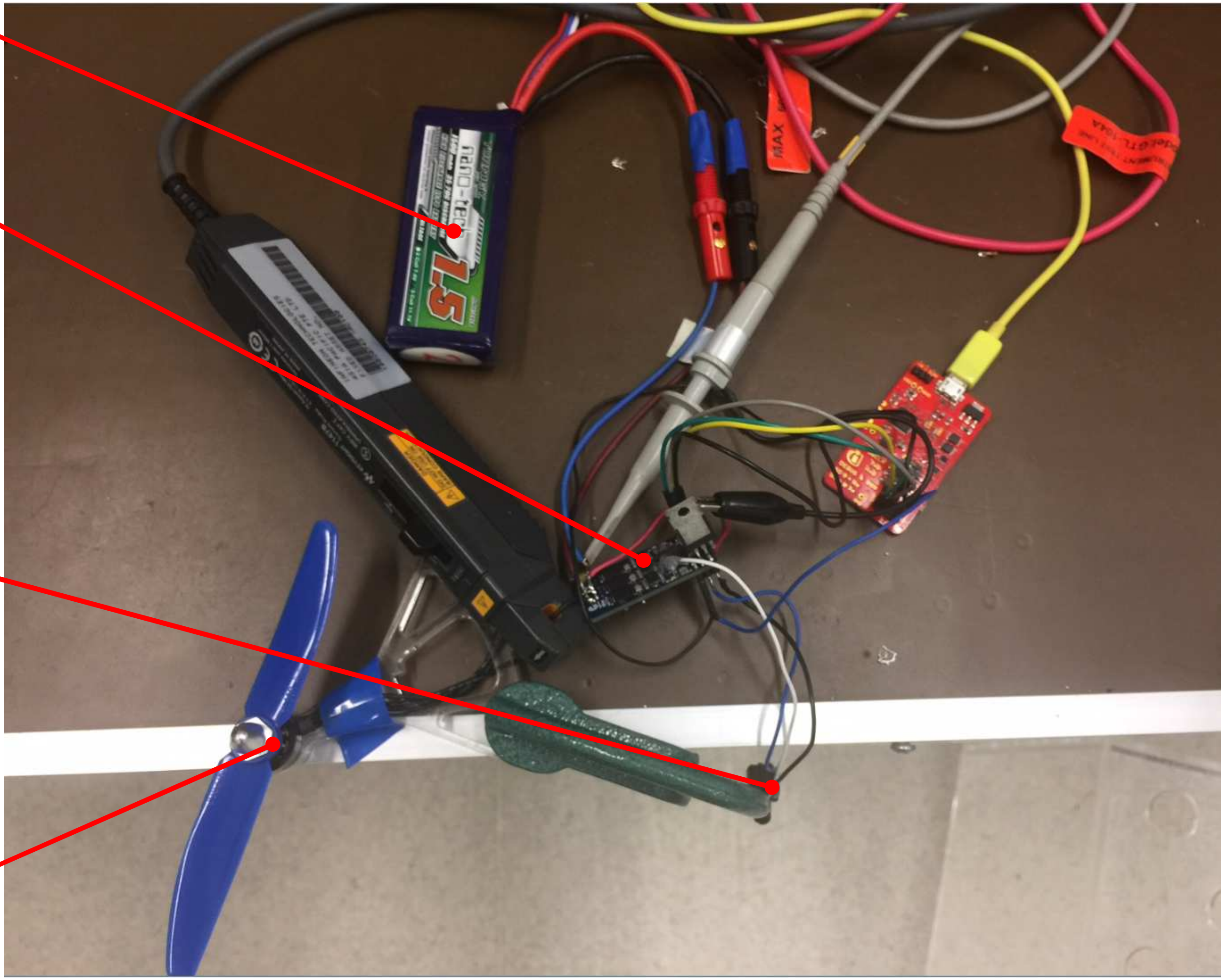
› Connect BLDC motor U, V and W phases to PINUS Inverter



# ESC motor control – Test Setup

- 3S LiPo Battery pack
- PINUS ESC
- Potentiometer
- BLDC 3 Phase Motors



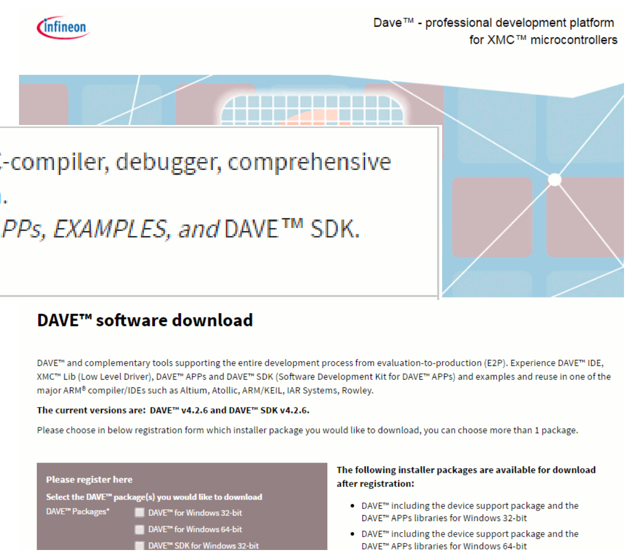
# ESC motor control - Get started - DAVE™ 4

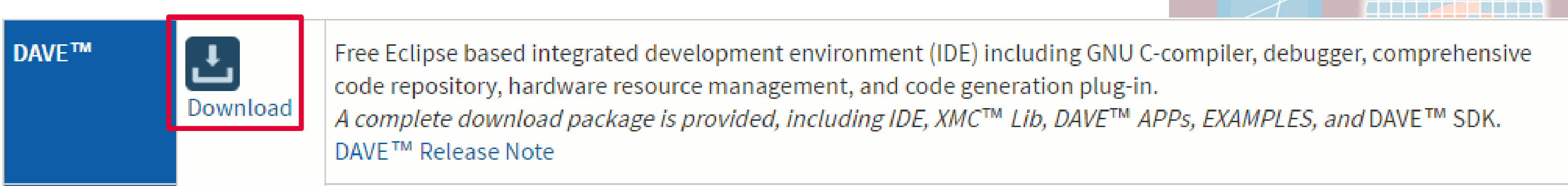
- Download the latest DAVE™ 4 installer package from

  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)



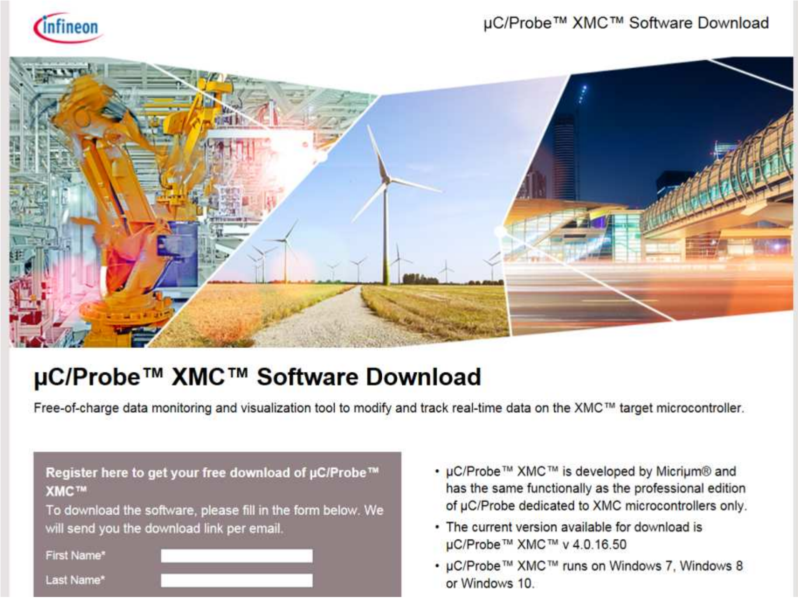
# ESC motor control -
# Get started - µC/Probe™ XMC™

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

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

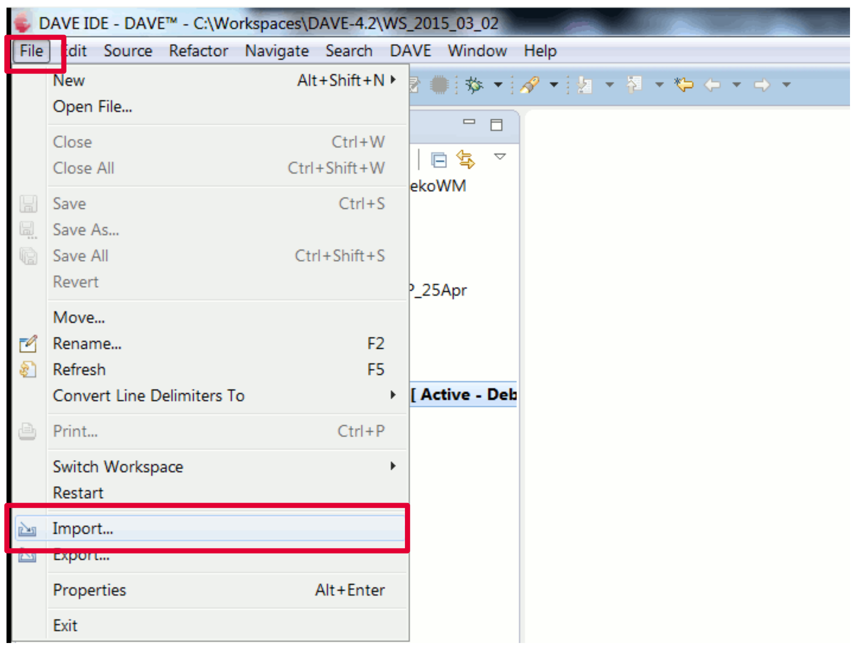
- Installation requirements

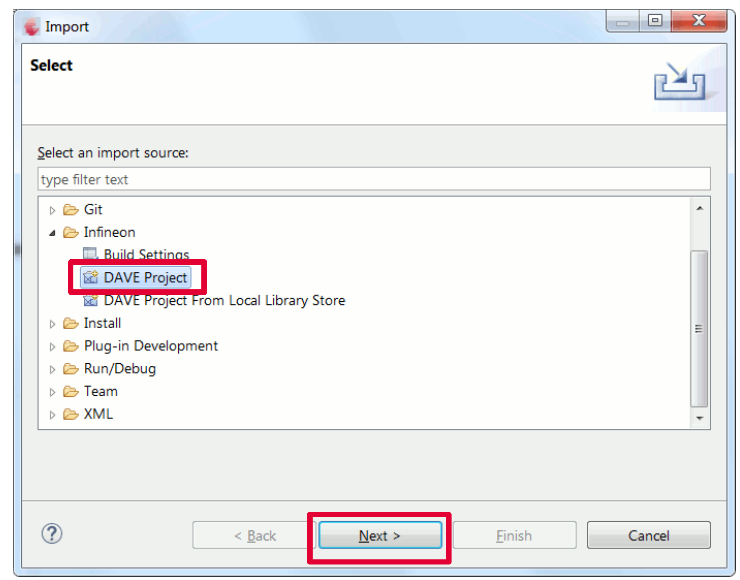
  1. PC with Windows Vista, Windows 7, Windows 8, Windows 10 - 32bit & 64bit
  2. RAM - 3 GB or more



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

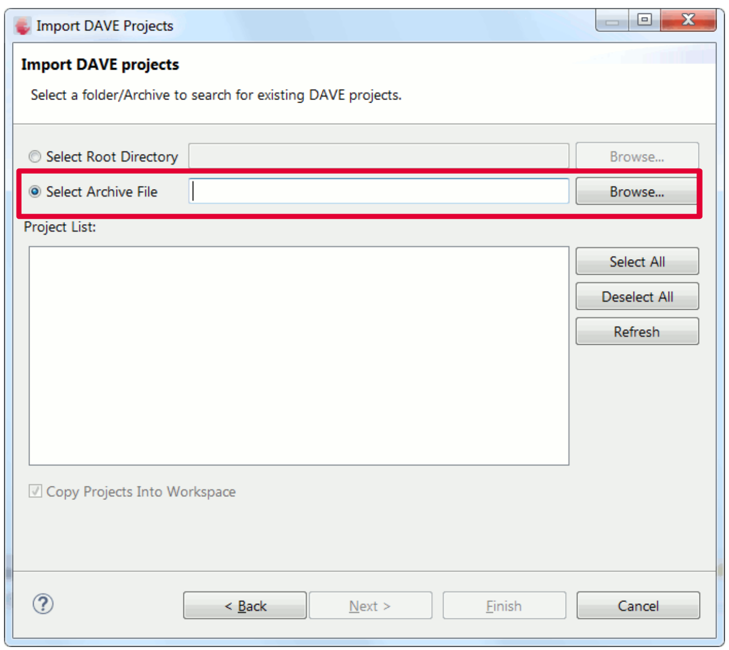
- Open DAVE™ 4
- Click on **File > Import** to import sample code
- Select **Infineon > DAVE project** and click "**Next**"

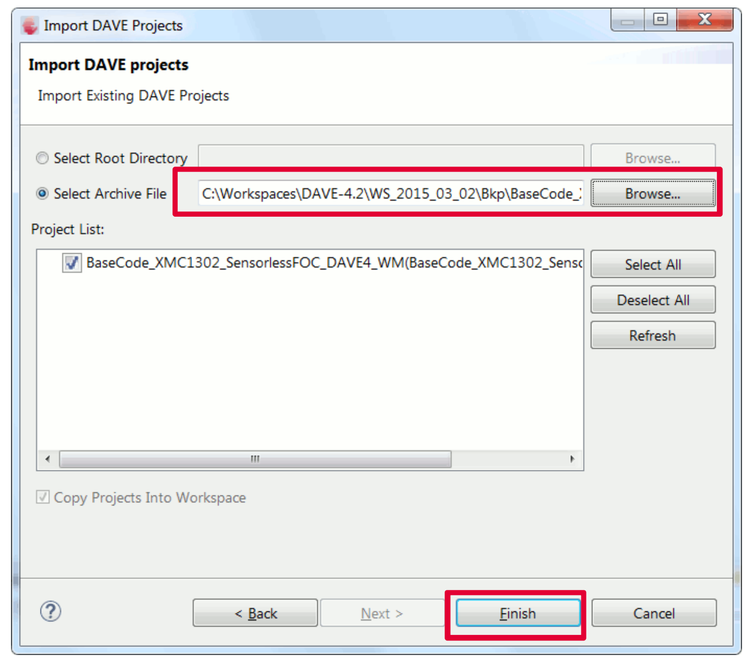


# ESC motor control - Get started - import SW to DAVE™ 4 (2/2)

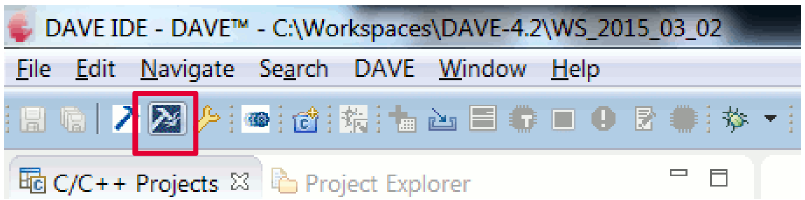
- Next click on **Select Archive File > Browse**
- Select the folder containing the sample code and click “**OK**”
- Click on “**Finish**” to import the code into DAVE™ 4



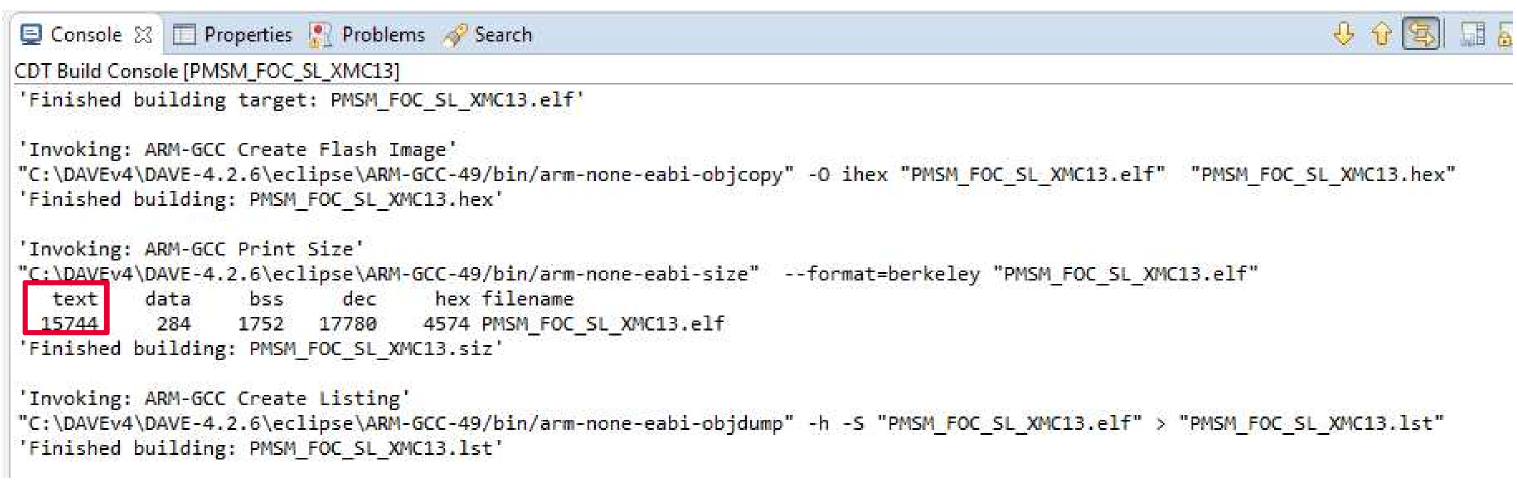



# ESC motor control -
# Get started - build SW in DAVE™ 4

› Click “**Rebuild Active Project**”

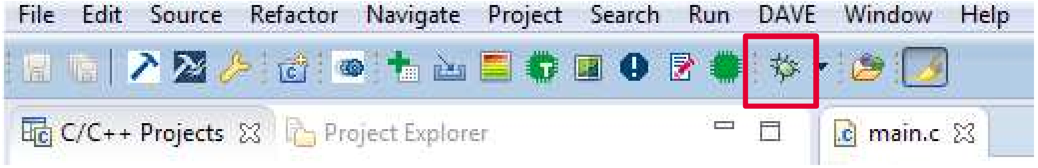



› “**text**” in red box indicates the code size, e.g.: about 15.7 kB

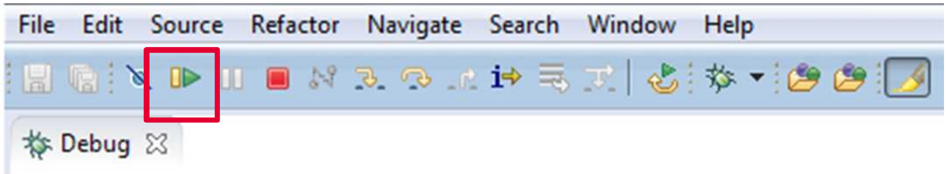


# ESC motor control -
# Get started - download SW in DAVE™ 4

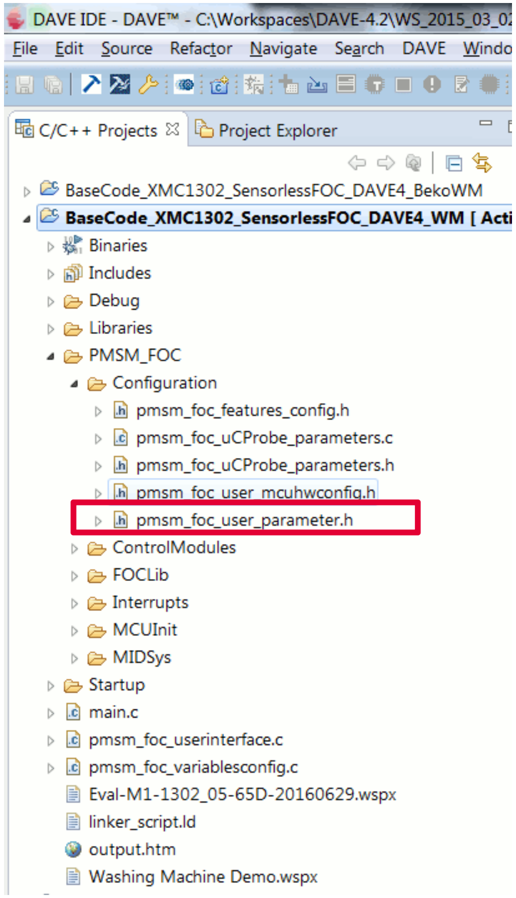
- Click "**Debug Configuration**" to download the code




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



# ESC 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

# ESC motor control - Get started - SW configuration (2/3)

```
/*******************************************************************************************************
 * MACROS
 *******************************************************************************************************/
#define  PMSM_FOC_HARDWARE_BOARD              IFX_XMC_PINUS_V2         /*1. KIT_XMC1X_AK_MOTOR_001
                                                                         2. KIT_XMC750WATT_MC_AK_V1
                                                                         3. IFX_XMC_LVPB_R2
                                                                         4. IFI_EVAL_24V_250W
                                                                         5. IFX_XMC_LVPB_R3
                                                                         [illegible]*/
/*      ---------------------------------------------------- Motor Type Selection ---------------------------------------- */
#define  MOTOR_TYPE                           MCI_DRONE_MOTOR            /*1. MCI_DRONE_MOTOR
                                                                         2. DJI_DRONE_MOTOR
                                                                         [illegible]
                                                                         4. NANOTEC_MOTOR
                                                                         5. MAXON_MOTOR
                                                                         6. BEKO_WM_MOTOR
                                                                         7.EBM_PAPST_VENTI_FAN_MOTOR                     */
/*      ---------------------------------------------------- Current feedback Sensing Mechanism -------------------------------------- */
#define  CURRENT_SENSING                      USER_THREE_SHUNT_SYNC_CONV   /*1. USER_SINGLE_SHUNT_CONV
                                                                          2. USER_THREE_SHUNT_ASSYNC_CONV
                                                                          3. USER_THREE_SHUNT_SYNC_CONV*/
/*      ---------------------------------------------------- FOC Control and Startup Scheme (Only Select 1 Scheme at one time) ---------------------------- */
#define MY_FOC_CONTROL_SCHEME                 CONSTANT_TORQUE_DIRECT_FOC  /*1. CONSTANT_SPEED_VF_ONLY,
                                                                         [illegible]
                                                                         3. CONSTANT_SPEED_DIRECT_FOC
                                                                         4. CONSTANT_TORQUE_DIRECT_FOC
                                                                         5. CONSTANT_VQ_DIRECT_FOC */
/*      ---------------------------------------------------- FOC Control Safety Protection ---------------------------------------- */
#define VDC_UNDER_OVERVOLTAGE_PROTECTION      ENABLED                  /*1. ENABLED       2. DISABLED*/
#define OVERCURRENT_PROTECTION                ENABLED                  /*1. ENABLED       2. DISABLED*/

#if(MOTOR_TYPE == MCI_DRONE_MOTOR)
/*      ---------------------------------------------------- Motor Parameters ------------------------------------------ */
#define  USER_MOTOR_R_PER_PHASE_OHM           (0.1f)          /* Motor Resistance per phase in Ohm*/
#define  USER_MOTOR_L_PER_PHASE_uH            (60.0f)         /* Motor Inductance per phase in uH */
#define  USER_MOTOR_POLE_PAIR                 (6.0f)          /* Motor Pole Pairs (change to integer) */
/*      ---------------------------------------------------- Constant Speed Control Mode (Used when Constant Speed Control is enabled) ---------------------------------------- */
/*      ---------------------------------------------------- POT ADC, or PWM to Adjust Speed ------------------------------------- */
#define USER_SPEED_HIGH_LIMIT_RPM             (16000U)          /* Max Speed of User Motor*/
#define USER_SPEED_LOW_LIMIT_RPM              (uint32_t) (USER_SPEED_HIGH_LIMIT_RPM / 30U)
#define USER_SPEED_RAMPUP_RPM_PER_S           (3000U)
#define USER_SPEED_RAMPDOWN_RPM_PER_S         (3000U)
/*      ---------------------------------------------------- V/F Start Up Parameters ------------------------------------------- */
#define USER_STARTUP_SPEED_RPM                (0U)
#define USER_STARTUP_SPEED_THRESHOLD_RPM      (100U)           /* threshold Speed to transit from Open loop to closed loop
                                                                * threshold Speed to transit from Open loop to
                                                                * closed loop
```

- #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

# ESC motor control - Get started - SW configuration (3/3)

```
#endif

/*      --------------------------------------------------- Hardware Inverter Parameters ---------------------------------------- */
#if(PMSM_FOC_HARDWARE_BOARD == IFX_XMC_PINUS_V2)
#define INTERNAL_OP_GAIN                      ENABLED              /*1. ENABLED       2. DISABLED (Please configure OP-Gain manually) */
#define USER_VDC_LINK_V                       (12.0f)               /* Hardware Inverter VDC link voltage in V  */
#define USER_DEAD_TIME_US                     (0.75f)               /* deadtime, rise(left) and fall values in us  */
#define USER_CCU8_PWM_FREQ_HZ                 (30000U)             /* CCU8 PWM Switching Frequency in Hz*/
#define USER_BOOTSTRAP_PRECHARGE_TIME_MS      (20U)                /* Initial Bootstrap precharging time in ms */
#define USER_DC_LINK_DIVIDER_RATIO            (float)(0.33f/(0.33f+2.32f))          /* R1/(R2+R1) ratio for DC link MCU ADC */
#define USER_VBEMF_RATIO                      (float)(0.33f/(0.33f+2.32f))          /* R1/(R2+R1) ratio for BEMF Voltage sensing circuit ratio */
#define USER_CURRENT_TRIP_THRESHOLD_A         (1.2f)               /* threshold current for trip detection in Ampere*/
#define USER_TRIP_THRESHOLD_TIME_MS           (100U)               /* threshold time for trip detection in ms */
#define USER_MAX_RETRY_MOTORSTARTUP_TRIP      (3U)                /* Max retry of motor startup if trip  */
/*      --------------------------------------------------- Motor Phase Current Measurement ---------------------------------------- */
#define USER_R_SHUNT_OHM                      (0.05f)              /* Phase shunt resistor in ohm */
#define USER_DC_SHUNT_OHM                     (0.030f)              /* DC link shunt current resistor in ohm */
#define USER_RIN_PHASECURRENT_KOHM            (5.1f)                /* R_IN (of equivalent amplifier) kohm */
#define USER_R_PHASECURRENT_FEEDBACK_KOHM     (39.0f)                /* R_FEEDBACK (of equivalent amplifier) kohm */
#define USER_RIN_DCCURRENT_KOHM               (10.0f)                /* Rf for dc current sensing */
#define USER_R_DCCURRENT_FEEDBACK_KOHM        (75.0f)                 /* Rin for dc current sensing */
#define USER_MAX_ADC_VDD_V                    (5.0f)                 /* VDD5, maximum voltage at ADC */
#define G_OPAMP_PER_PHASECURRENT              (USER_R_PHASECURRENT_FEEDBACK_KOHM / USER_RIN_PHASECURRENT_KOHM)
#define I_MAX_A                               ((VAREF_V/(USER_R_SHUNT_OHM * OP_GAIN_FACTOR)) / 2U)          /* For IFX_XMC_LVPB_R3, I_MAX_A = 13.16A */

#if(INTERNAL_OP_GAIN == ENABLED)
#define OP_GAIN_FACTOR                       (6U)                 /* Different HW Board has different OP Gain factor, XMC13 built-in Gain Factor available 1, 3, 6 and 12 only*/
#elif(INTERNAL_OP_GAIN == DISABLED)
#define OP_GAIN_FACTOR                        G_OPAMP_PER_PHASECURRENT
#endif
```

**#4** **#5** **#6**

- #4: PINUS inverter is designed to make use of XMC1302 internal OP-Amp Gain for phase current ADC sensing
- #6: CCU8 PWM Peripherals configuration
- #5: Macros to configure based on Inverter HW specs

# ESC 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 based on Inverter hardware specs

# ESC motor control - Get started - pmsm_foc_user_mcuhwconfig.h

- MCU hardware resource management (VADC, CCU8)

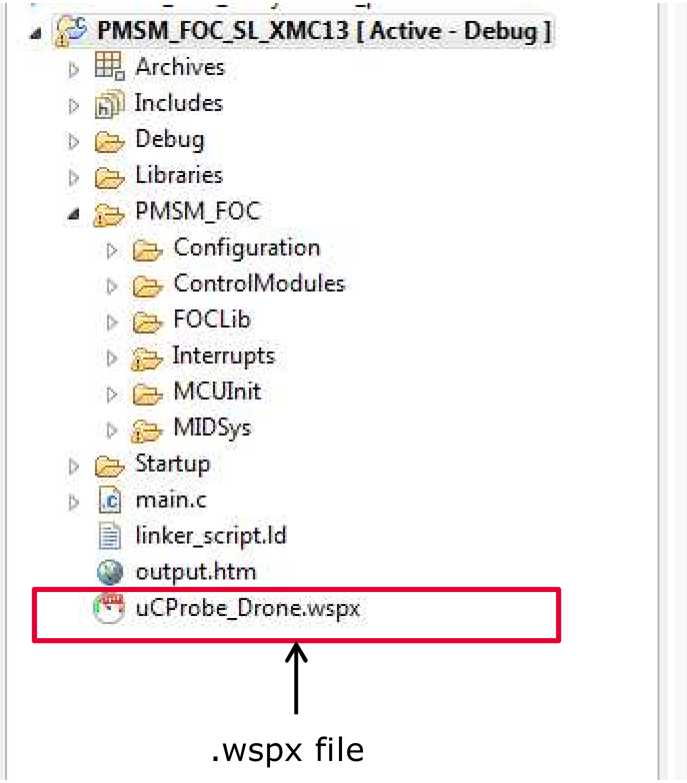
- NVIC interrupts service routine resource management

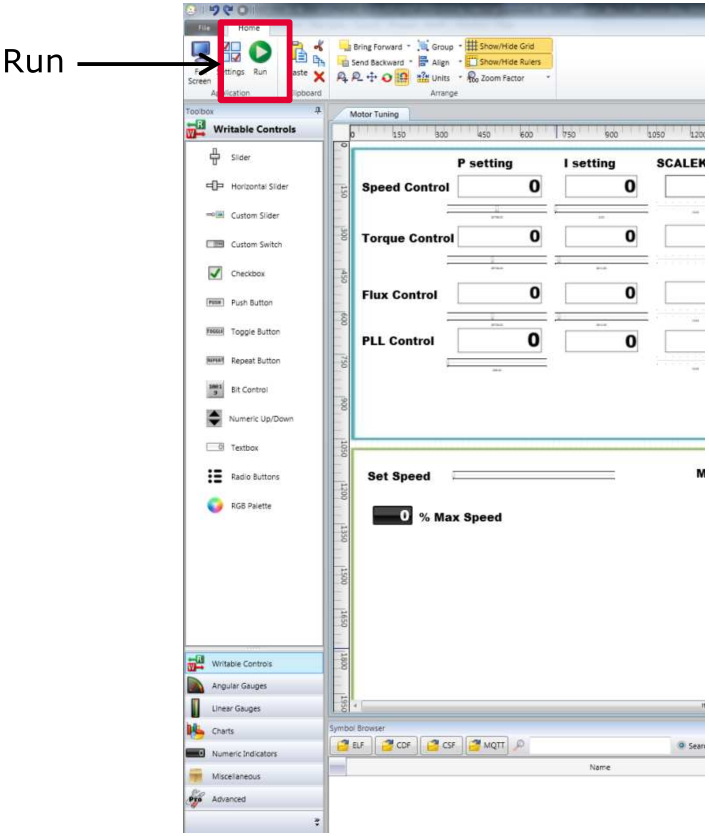
- Debugging IO (DAC functionality)

# ESC motor control - Get started - starting µC/Probe™ XMC™

› Double-click "***.wspx**" file in the DAVE™ 4 IDE to start µC/Probe™

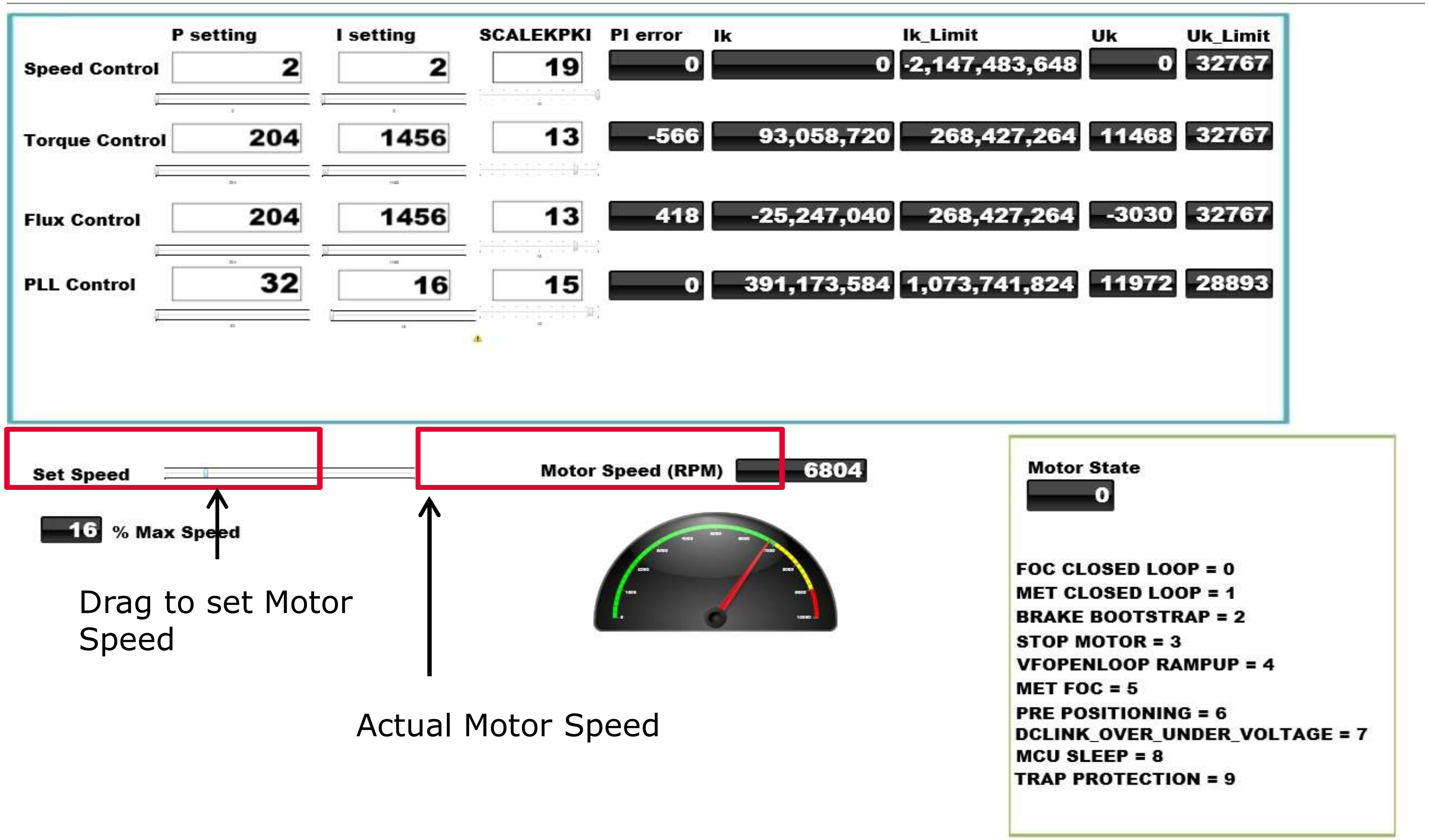
› Click "**Run**" to control the speed of the motor using µC/Probe™



# ESC motor control - Get started - start motor using µC/Probe™

› The motor can be started by adjusting the slider in the "**Set Speed**" box



# ESC motor control - Get started - fine-tune Kp/Ki using µC/Probe™

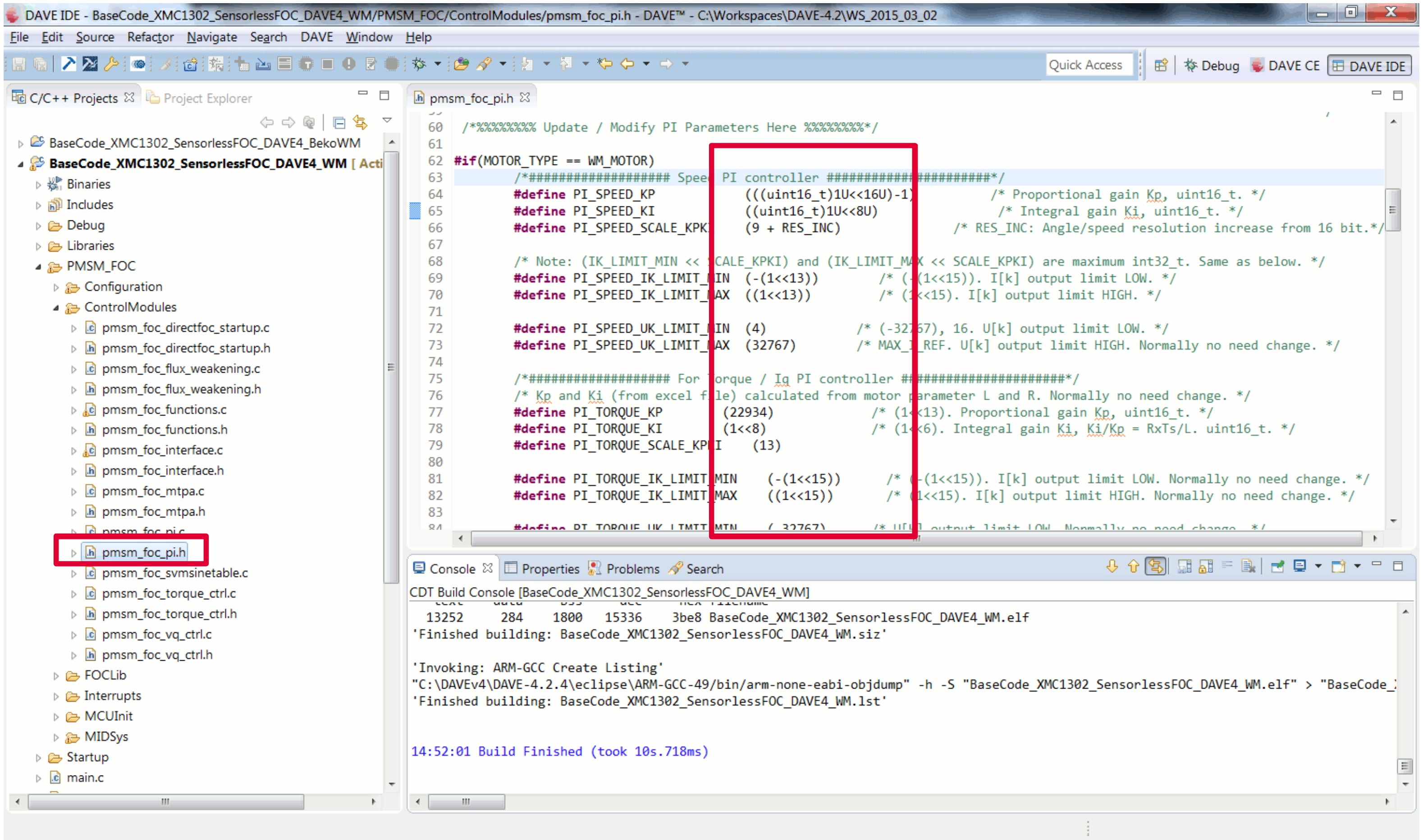
- If the motor does not spin in FOC close loop, ↑ the SCALEKPKI of PLL Control and check the motor behavior. If motor start to move slowly, ↑ the SCALEKPKI further, else, ↓ the SCALEKPKI
- Apply similar tactic for the tuning of Speed Control

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

| | | P setting | I setting | SCALEKPKI | |
|---|---|---|---|---|---|
| 2 | Speed Control | 65535 | 256 | 12 | ← ↑ this value by 1 will ↓ gain of Speed controller by half |
| | Torque Control | 22934 | 256 | 13 | |
| | Flux Control | 22934 | 256 | 13 | |
| 1 | PLL Control | 6962 | 32 | 14 | ← ↑ this value by 1 will ↓ gain of PLL estimator controller by half |

# ESC 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**

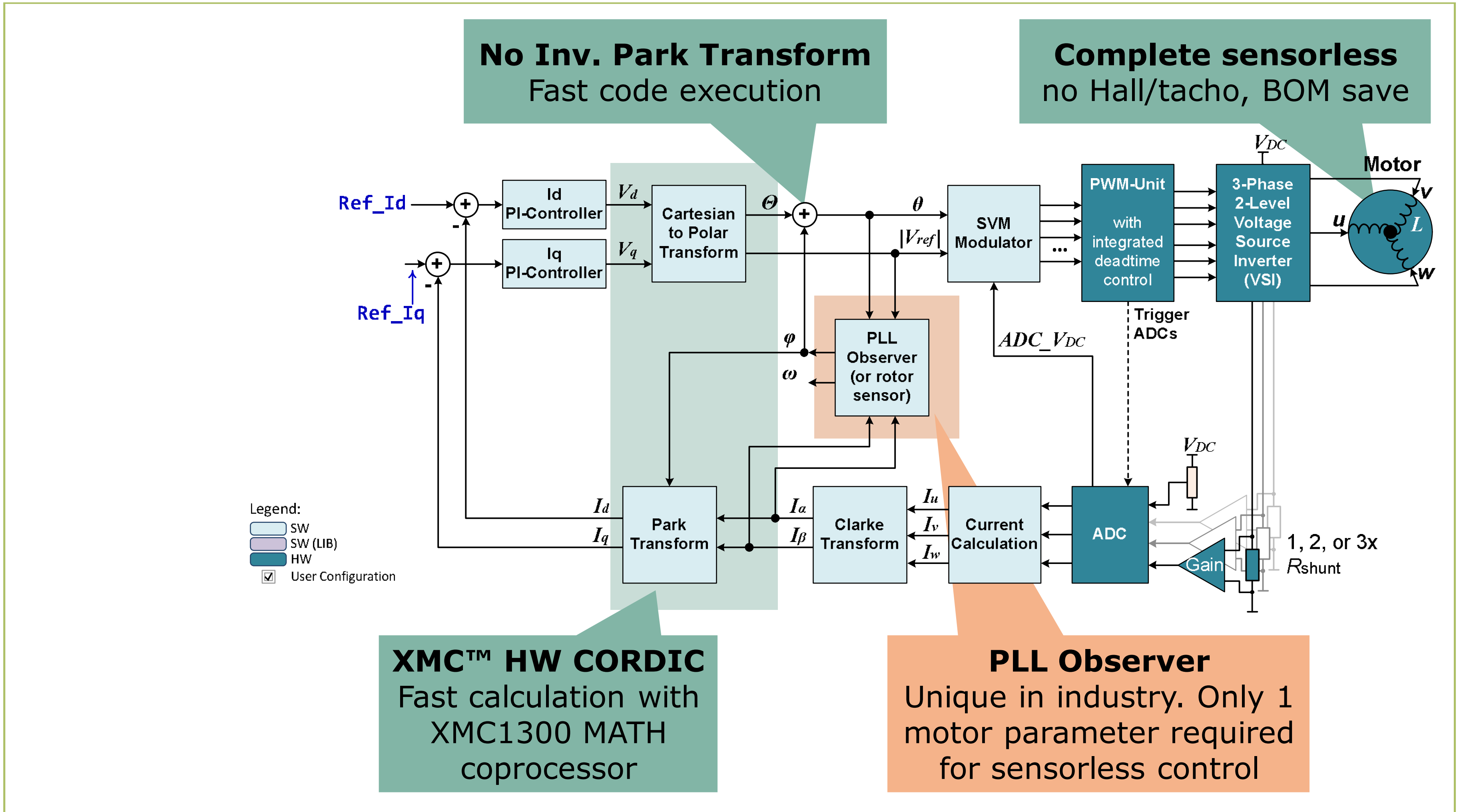


# ESC motor control - Key features

1. Sensorless FOC control even at ultra-low speed
2. Robust direct-sensorless-FOC startup
3. Fast execution FOC with XMC™ Cortex M0
4. XMC™ on-chip ADC gain to reduce system BOM cost

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

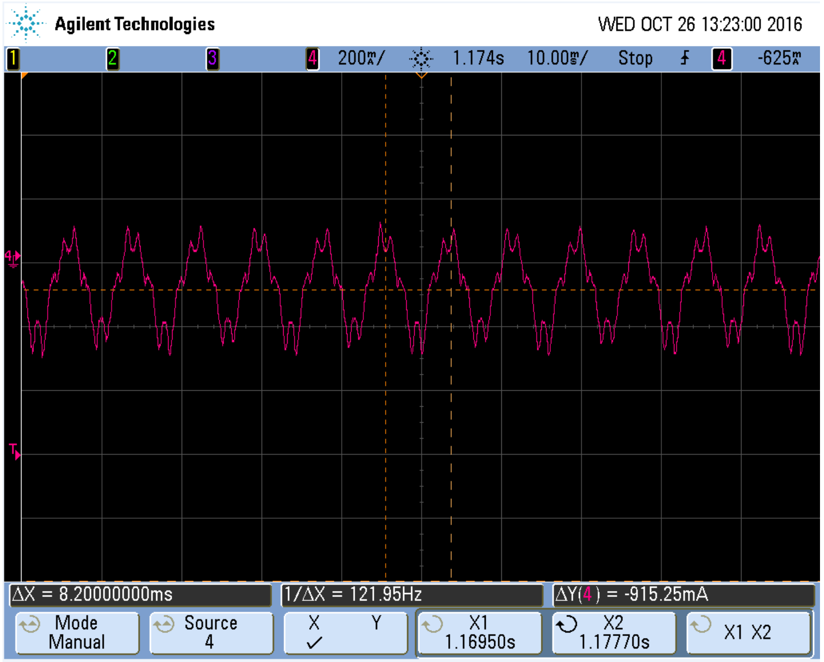
# ESC - sensorless FOC - Block diagram

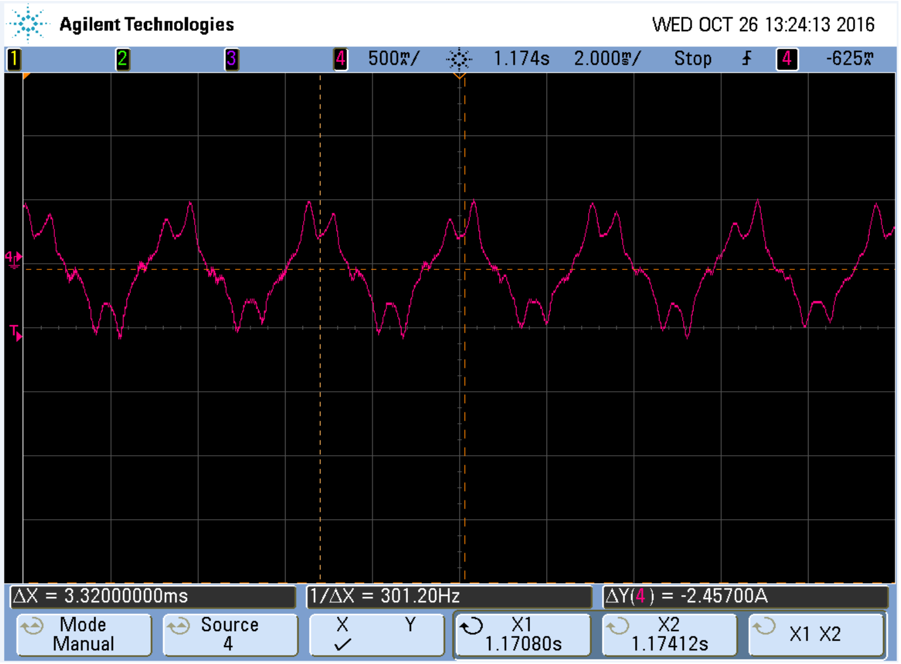



Block diagram: ESC motor control - sensorless FOC

# Sensorless FOC control at ultra-low speed - Waveforms

› Ultra-low speed: 1000 rpm 2000 rpm



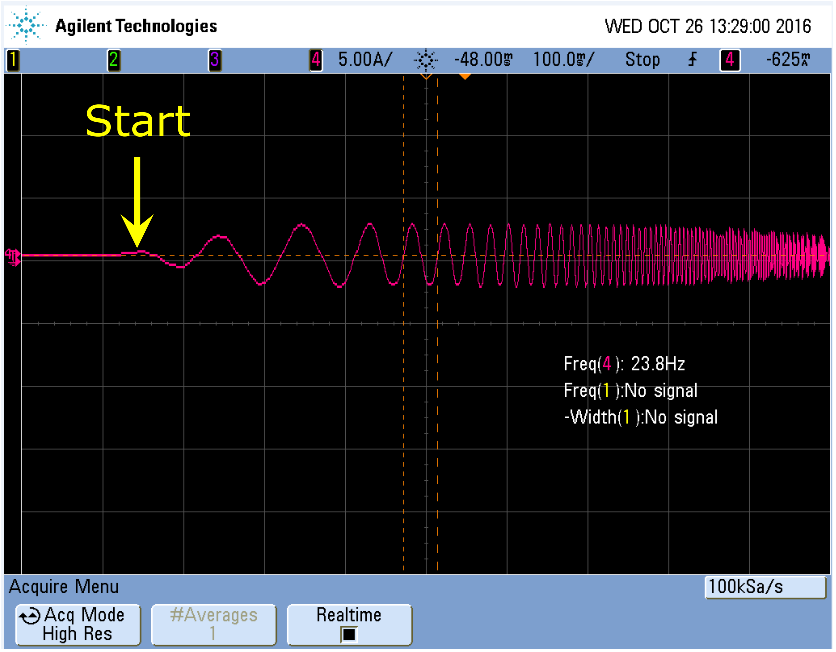

› CH4 (Pink) - phase current Iu, from current probe – with High Resolution Mode enabled

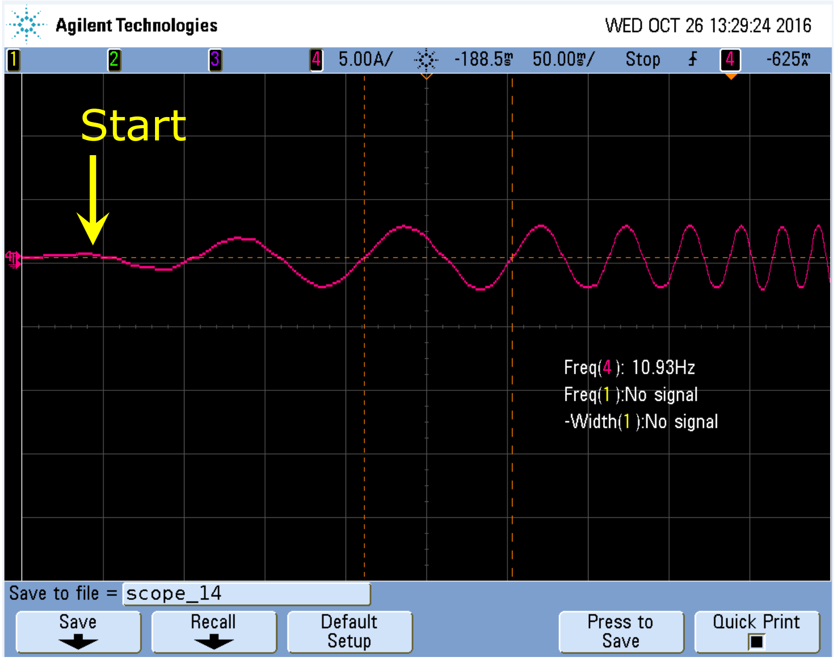
**Motor parameter:**
*L* (per phase): 60 *u*H
*R* (per phase): 0.1 *Ω*
Pole-pair No.: 6

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

# Startup with load Waveforms

› From 0 rpm to 10,000 rpm XMC™ sensorless FOC Direct Startup

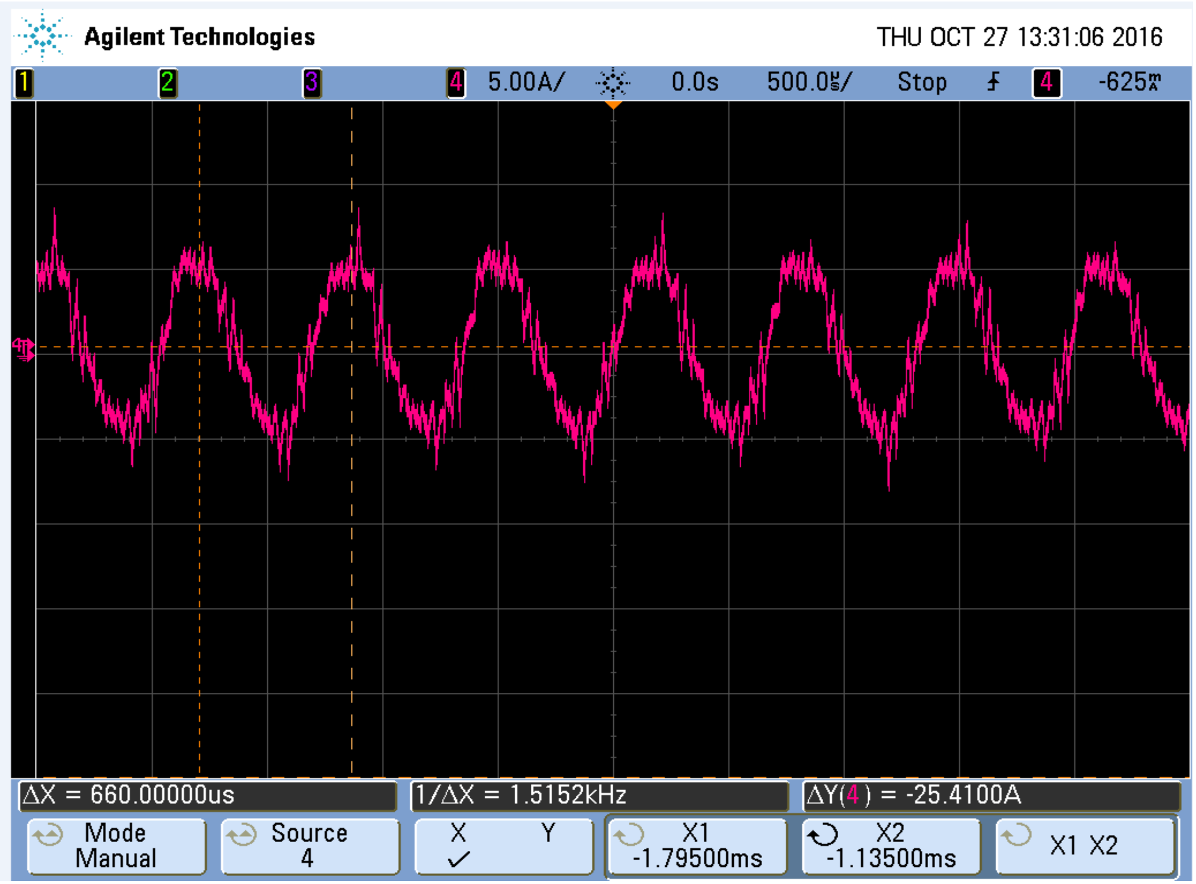



› CH4 (Pink) - phase current Iu, from current probe – with High Resolution Mode enabled

**Motor parameter:**
*L* (per phase): 60 *u*H
*R* (per phase): 0.1 *Ω*
Pole-pair No.: 6

# XMC™ sensorless FOC with load Waveform @ 15000 rpm VDC = 12V, 5A




- **Channel 4 (pink): Current of fan motor Phase U (measured by current probe, 0.1V/A)**

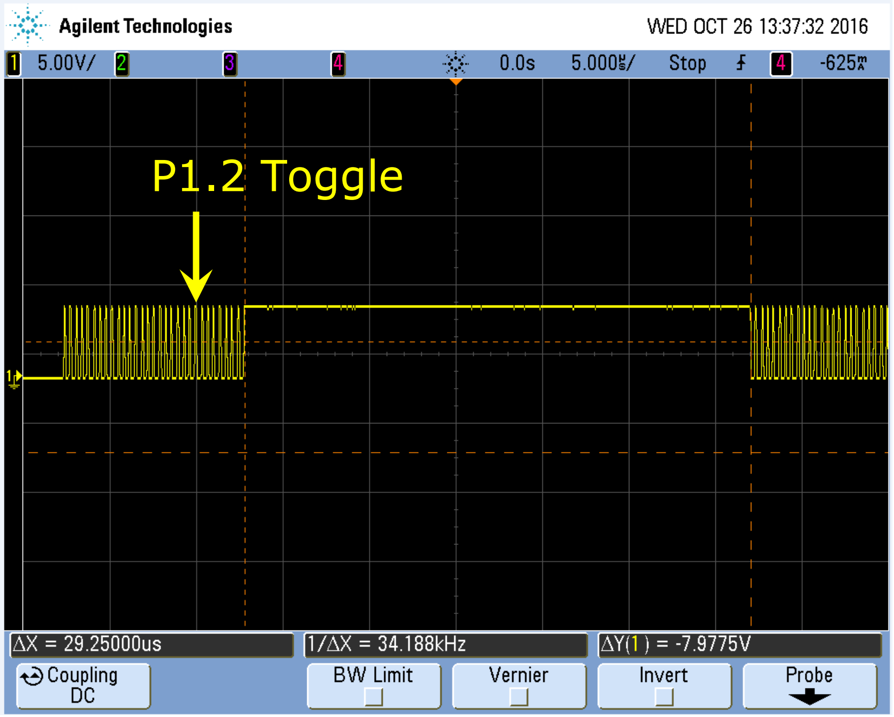
Waveforms: Drone

# 3. Key feature – Fast Execution FOC with XMC™ Cortex M0

# ESC FOC Execution with XMC™ 1302

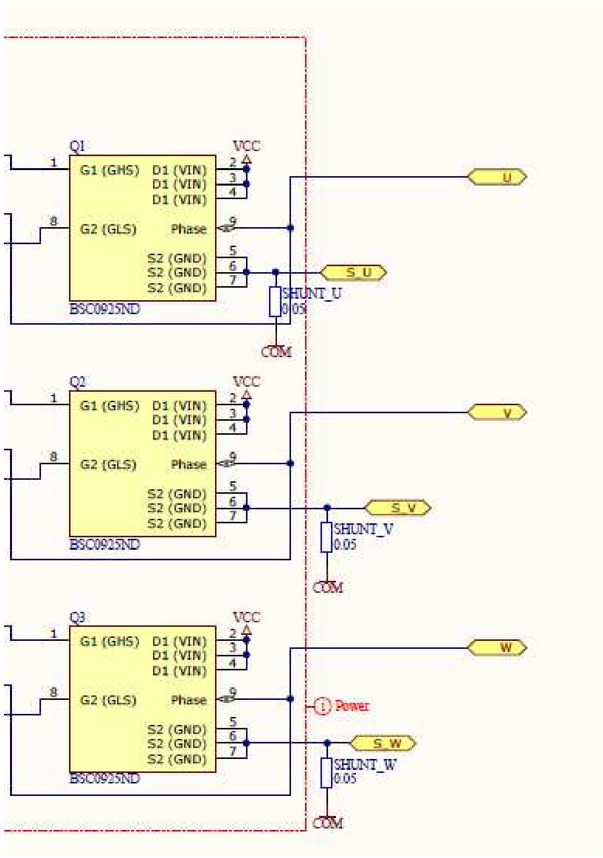
| S/N | FOC SW Description | Sensorless FOC Execution Time and CPU Load (CCU8 PWM 25kHz / 40$\mu s$) | | |
|---|---|---|---|---|
| | | Code Executed from RAM | | |
| | | Execution Time | Free Time per PWM | CPU Load |
| 1 | Latest PMSM_FOC Base Code | 29.25 $\mu s$ | 10.75 $\mu s$ | **75%** |

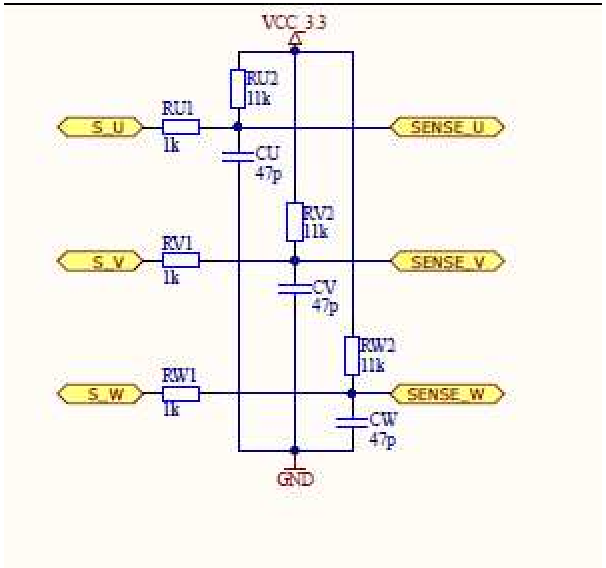
- **The execution time is measured by toggling P1.2 at while loop in main.c**



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

# ESC motor control - Current sensing schematics



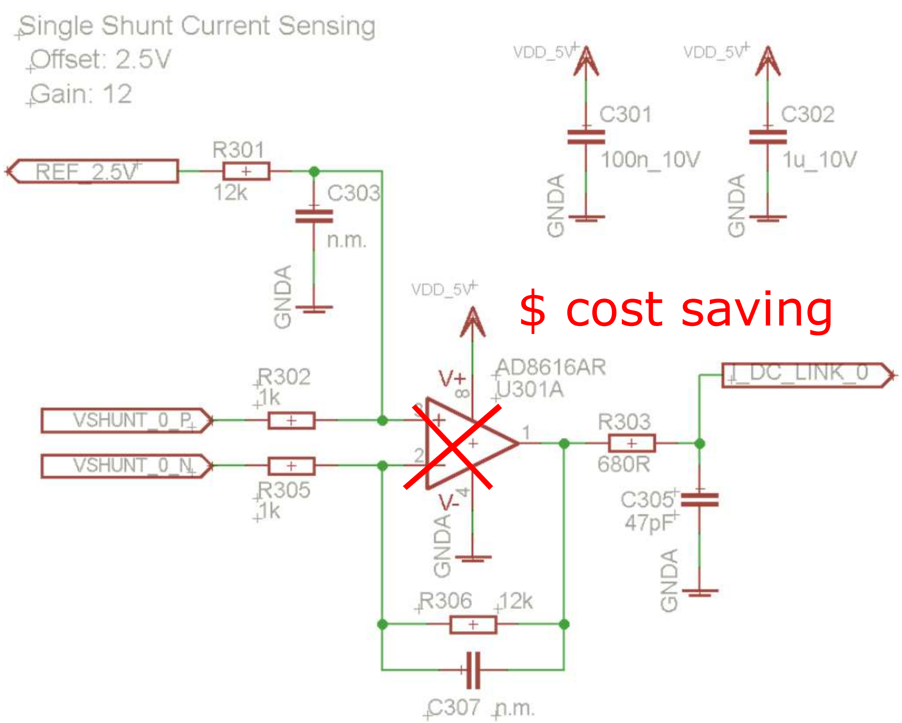



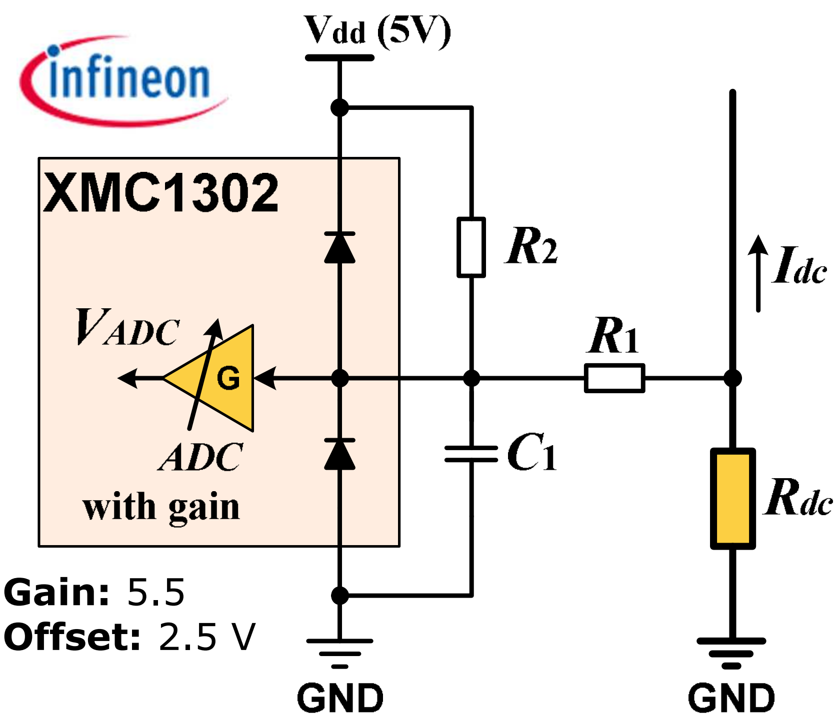

Schematic: ESC motor control - current sensing

# ESC 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

- ESC motor control
  - Documentation
  - Infineon multicopter solution landing page
  - DAVE™ project

# Support material:

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