Advanced BLDC Motor Drive and Control

Giovanni Tomasello – Applications Engineer

life.augmented

Technology Tour 2017

3-Phase BLDC Motor-Control Block Diagram

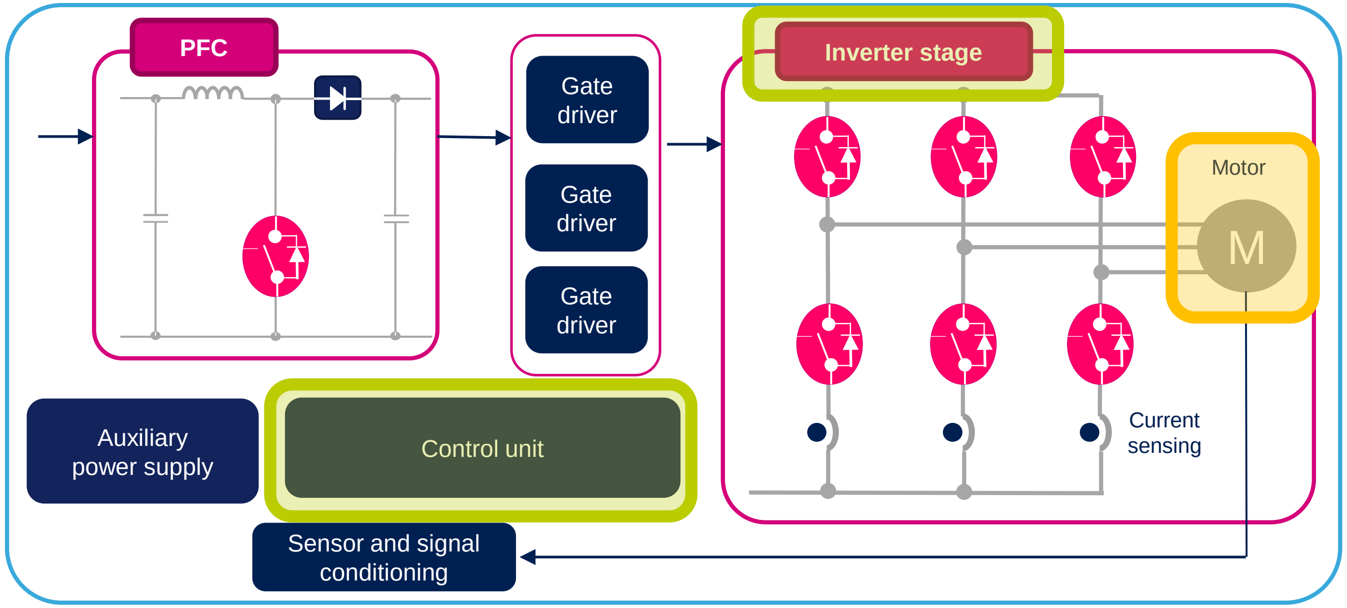
HV Only

PFC Controllers (L49xx)
Rectifiers (STTHxx, STPSxx)
Power MOSFETs (Mdmesh™ M2, M5 600V-650V**)**

HV and LV

IGBT (TFS 600V - 1200V)
IPM (SLLIMM™)
Power MOSFETs (HV and LV**)**
SYSTEM IN PACKAGE (SiP): STSPIN32 (up to 45V)

Tools (HW & SW)



Power Management
VIPERxx, LDO, DC-DC…

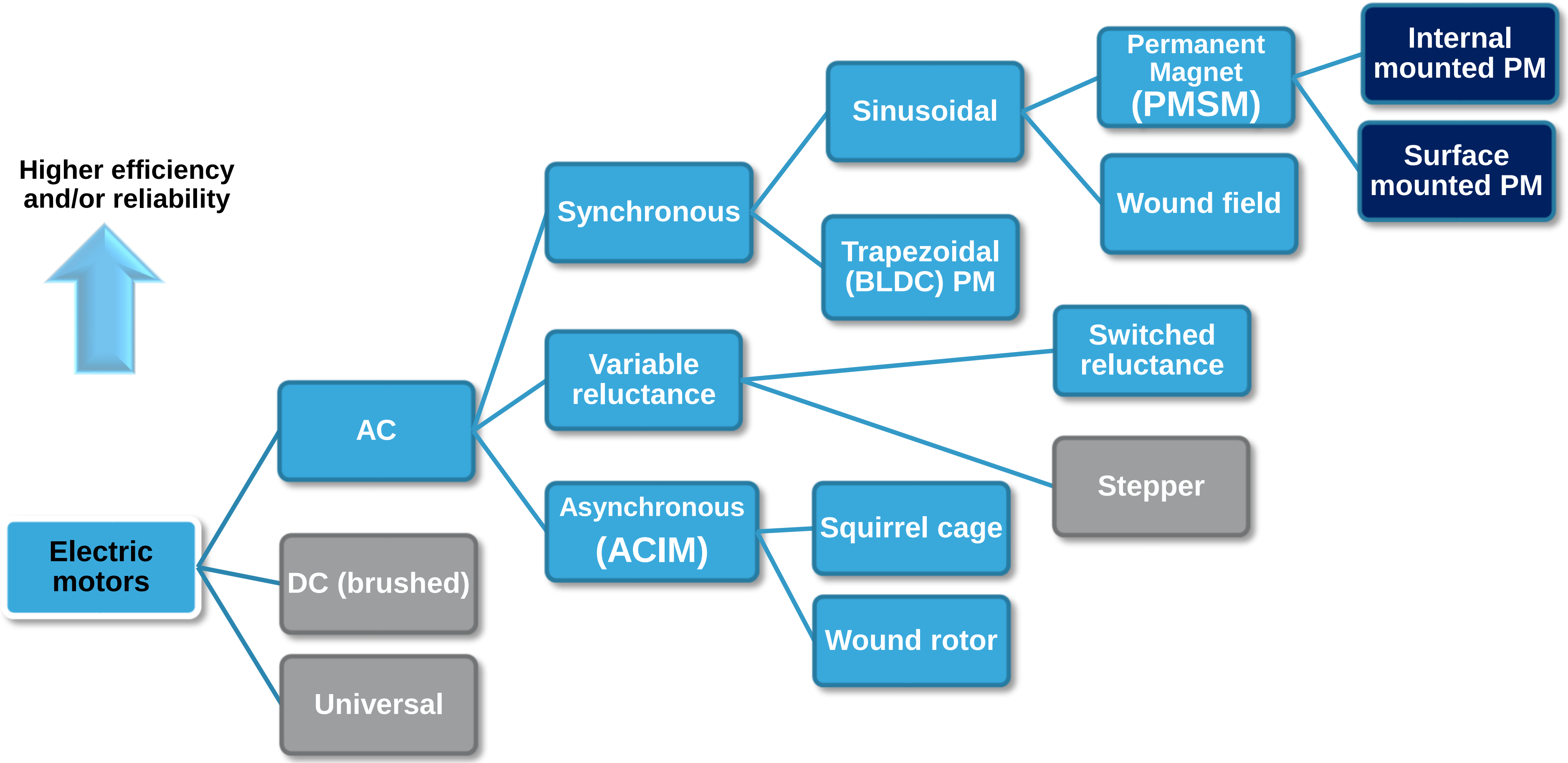
Op. Amp. and comparators

Microcontrollers 8-bit / 32-bit

Gate Drivers
L638x, L639x, L649x[1], STGAPxx

life.augmented

Electric Motor: Classification



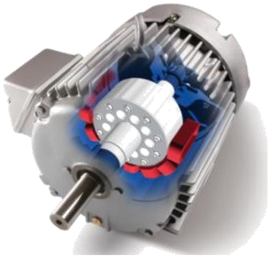
- **PMSM:** 3-phase permanent magnet synchronous motor
- **ACIM:** 3-phase induction motor

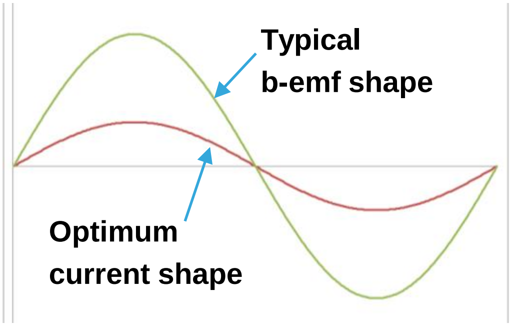
Limited computation needs
Driving method well-known, mastered by customer
Light ecosystem
Basic ADC/PWM requirement

Computation intensive
Complex driving, requires specific knowledge and/or support
Complete ecosystem necessary
Requires 3-phase timer + sync'd ADC

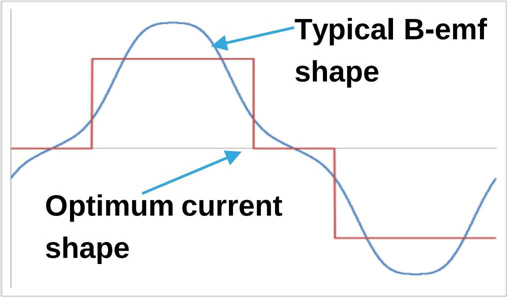
PMSM and BLDC Motors

- **Permanent Magnet Synchronous Motor (PMSM)**
  - Stator is the same as AC IM: three phase windings
  - Rotor houses permanent magnets
    - on the surface → Surface Mounted (SM) PMSM
    - Buried within the rotor →Internal (I) PMSM
  - Rotation induces sinusoidal Back Electro-Motive Force (BEMF) in motor phases
  - Gives best performances (torque steadiness) when driven by sinusoidal phase current





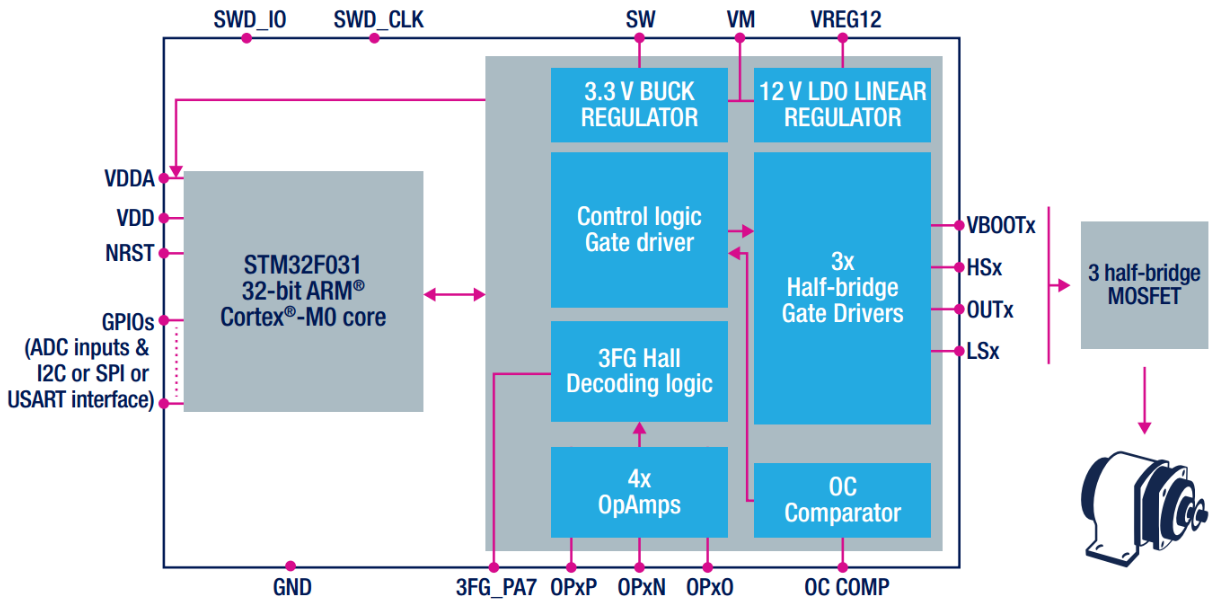
- **Permanent Magnet BrushLess DC motors (BLDC)**
  - Like PMSM - and despite of their name - require alternating stator current
  - Like in PMSM, rotor houses permanent magnets, usually glued on the surface
  - Like PMSM, stator excitation frequency matches rotor electrical speed
  - Unlike PMSM, rotor spinning induced trapezoidal shaped Back Electro-Motive Force (Bemf)
  - Gives best performances (torque steadiness) when driven by rectangular-shaped currents

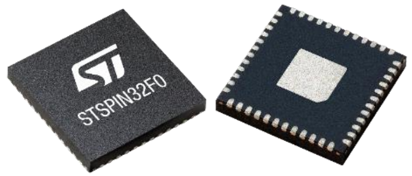


life.augmented

STSPIN32F0 Technical Details

- Operating voltage from 8V to 45V
- 3-phase gate driver for high performances
  - 600mA current capability
  - Real-time programmable over current
  - Integrated bootstrap diodes
  - Cross conduction, under-voltage and temperature protections
- 32-bit STM32F0 MCU with ARM ® Cortex® M0 Core
  - STM32F031x6x7 48MHz, 4-Kb SRAM and 32-Kb Flash
  - 12-bit ADC
  - 1 to 3 shunts FOC supported
  - Communication interfaces I2C, UART and SPI
  - Complete Development Ecosystem available
- 4x Operational Amplifiers and a Comparator
- On-chip generated supplies for MCU driver and external circuitry
  - 3.3V DC/DC buck regulator – Input voltage up to 45V
  - 12V LDO linear regulator
  - UVLO protection on all power supply voltages
- Embedded Over-Temperature Protection

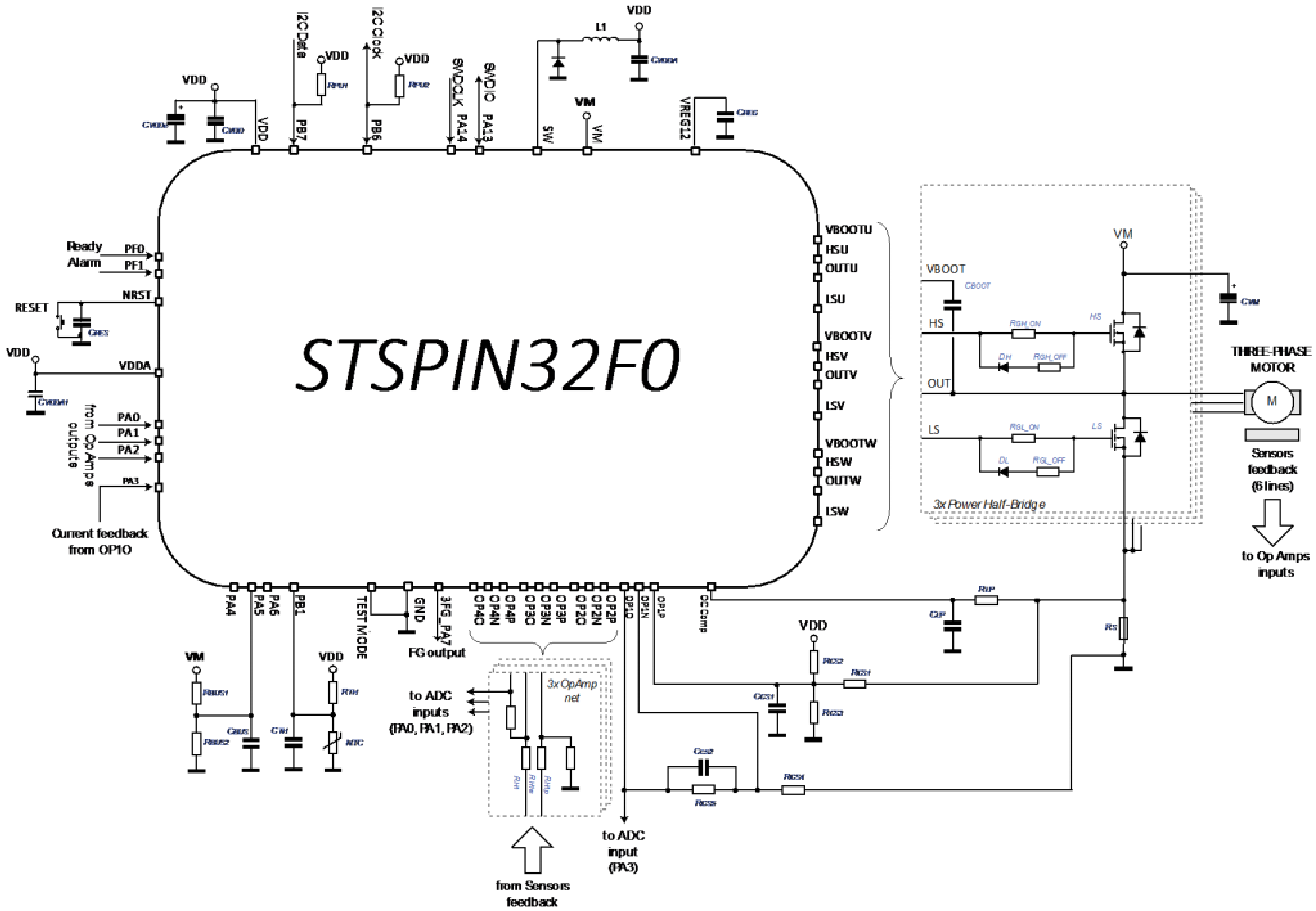




- Package : VFQFPN 7 x 7 x 1.0 - 48L

life.augmented

STSPIN32F0 Application Example



Making Your Designs Easier

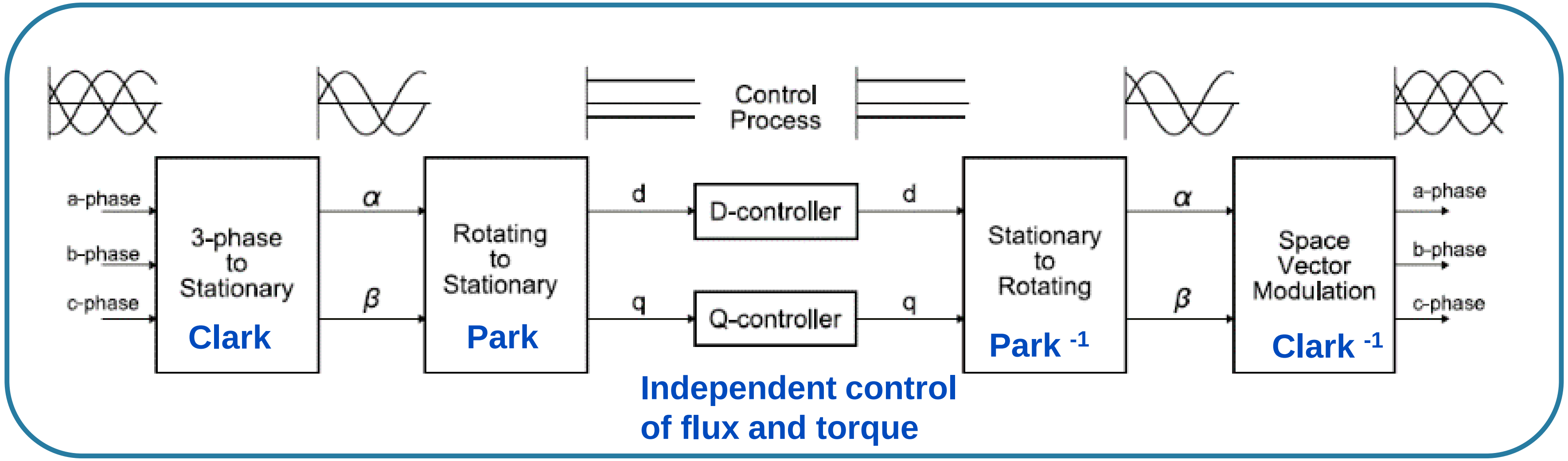
To support STSPIN32F0, a comprehensive set of design tools is available, including:

Reference Code	Description
STEVAL-SPIN3201	STSPIN32F0 evaluation board Three-phase brushless DC motor driver evaluation board • Input voltage from 8 to 45 V • Output current up to 15 Arms • Power stage based on STD140N6F7 MOSFETs • Sensored or sensorless field-oriented control algorithm with 3-shunt sensing
UM2154	User manual for STEVAL-SPIN3201: advanced BLDC controller with embedded STM32 MCU evaluation board
STSW-SPIN3201	Firmware example for field oriented motor control (FOC)
UM2152	User manual for Getting started with the STSPIN32F0 FOC firmware example STSW-SPIN3201
STSW-STM32100	Library: STM32 PMSM FOC Software Development Kit

life.augmented

Why FOC?

- Best energy ***efficiency*** even during ***transient operation.***
- ***Responsive speed control*** to load variations.
- ***Decoupled control*** of both electromagnetic torque and flux.
- Acoustical ***noise reduction*** due to sinusoidal waveforms.
- Active ***electrical brake*** and ***energy reversal.***

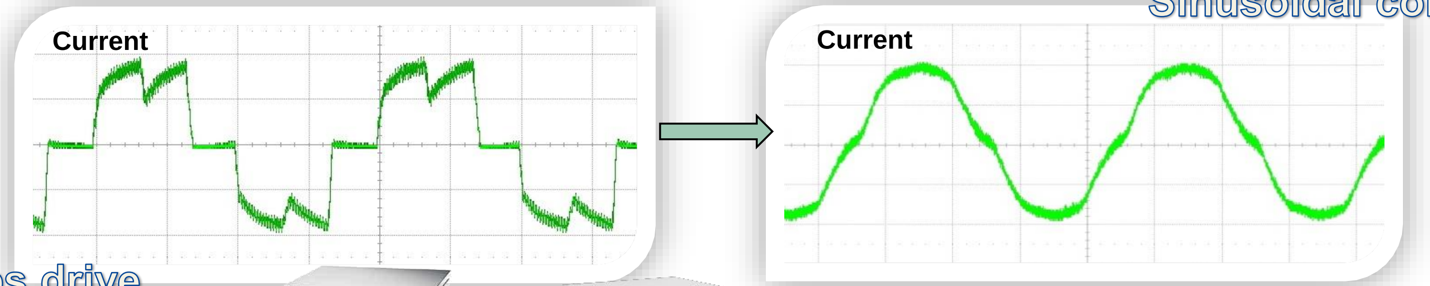


FOC Single Motor For Budgetary Applications

- **Target applications:**
  - All those applications where:
    - Dynamic performance requirements are moderate
    - Quietness of sinusoidal current control (vs six steps drive) is valuable
    - Extended speed range is required
  - Particularly suitable for **pumps, fans and compressors**

STM32F100x
STM32F0xx

Sinusoidal control



Six-steps drive

DW Spray & drain pumps

WM Drain pump

Fridge compressor

- More silent
- Lower torque ripple
- Extended speed range easier to be achieved

life.augmented

FOC Single Or Dual Motor For Higher Performance

- **Target applications:**
  - Wide range from home appliances to robotics, where:
    - Accurate and quick regulation of motor speed and/or torque is required (e.g. in torque load transient or target speed abrupt variations)
    - CPU load granted to motor control must be low, due to other duties

STM32F103 → STM32F3xx → STM32F2xx → STM32F4xx

Games


Home appliances


Power tools


Industrial motor drives

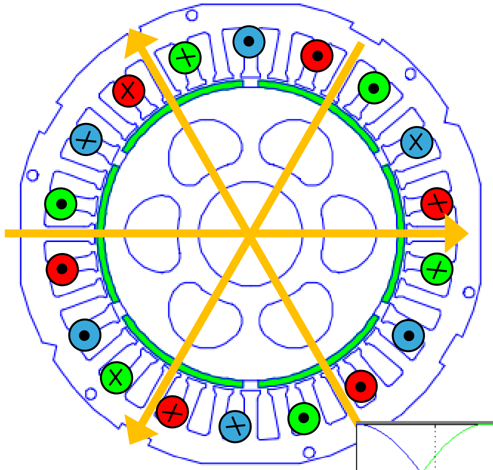
Fitness, wellness and healthcare

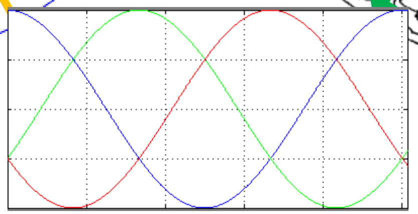

Escalators and elevators

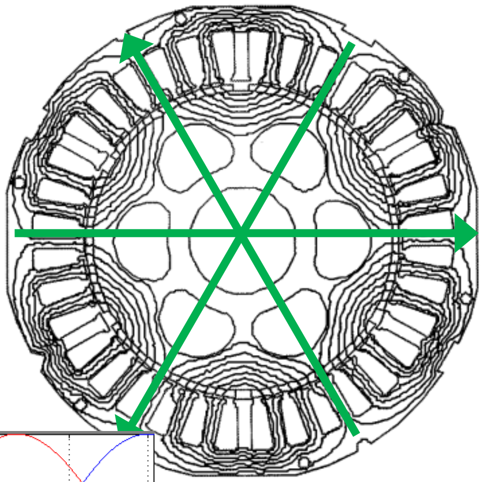
And much much more…

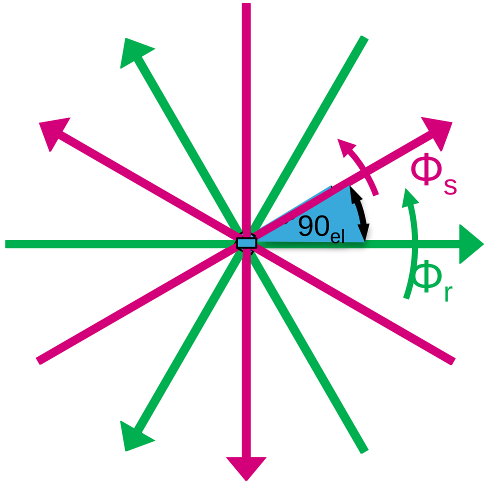
PMSM FOC Overview

- <u>Field Oriented Control: stator currents (Field) are controlled in amplitude and phase (Orientation) with respect to rotor flux</u>
  - ➢ current sensing is mandatory (3shunt/1shunt/ICS)
  - ➢ speed / position sensing is mandatory (encoder/Hall/sensorless alg)
  - ➢ current controllers needed (PI/D,FF)
    - ❖ not easy… high frequency sinusoidal references + stiff amplitude modulation..
    - ❖ reference frame transformation (Clarke / Park) allows to simplify the problem:





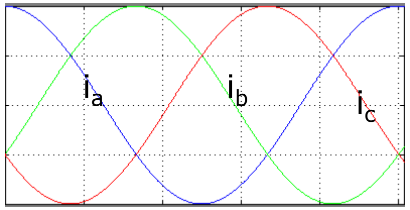




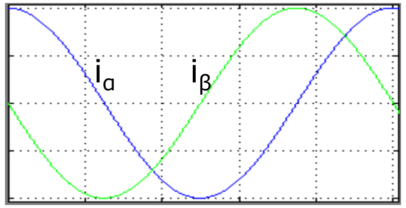
T_e maximized if…

PMSM FOC Overview: Reference Frame Transformations

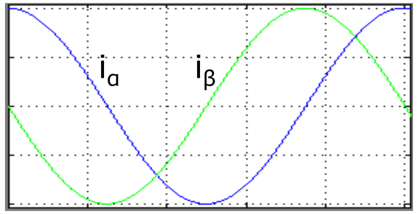
- Clarke: transforms i_a,i_b,i_c (120°) to i_α,i_β (90°); (consider that ia+ib+ic=0);



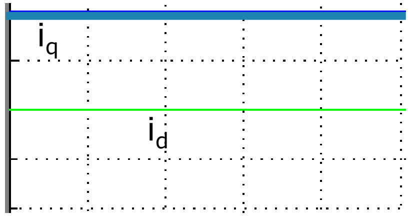
$$i_\alpha = i_{as}$$
$$i_\beta = -\frac{i_{as} + 2i_{bs}}{\sqrt{3}}$$



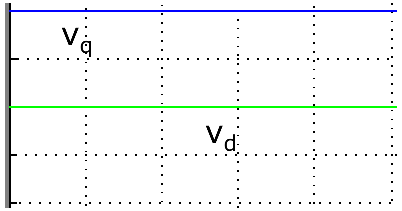
- Park: currents i_α,i_β , transformed on a reference frame rotating with their frequency, become DC currents i_q,i_d (90°)!



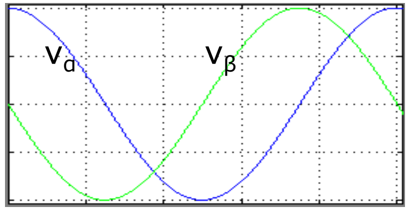
$$i_{qs} = i_\alpha \cos\theta_r - i_\beta \sin\theta_r$$
$$i_{ds} = i_a \sin\theta_r + i_\beta \cos\theta_r$$



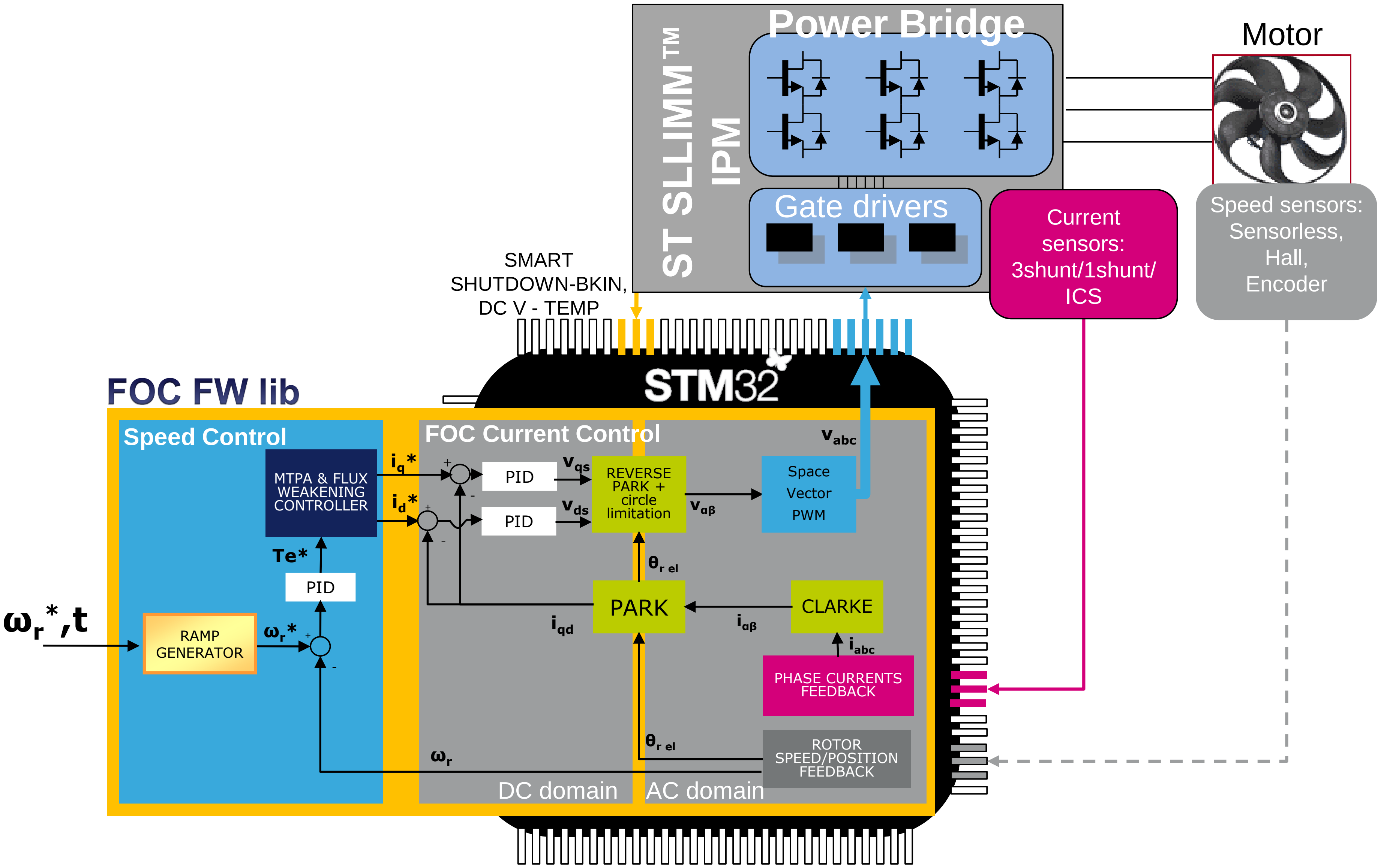
- PI regulators now work efficiently in a ‘DC’ domain; their DC outputs, voltage reference v_q,v_d are handled by the Reverse Park -> v_α,v_β AC domain



$$v_\alpha = v_{qs} \cos\theta_r + v_{ds} \sin\theta_r$$
$$v_\beta = -v_{qs} \sin\theta_r + v_{ds} \cos\theta_r$$

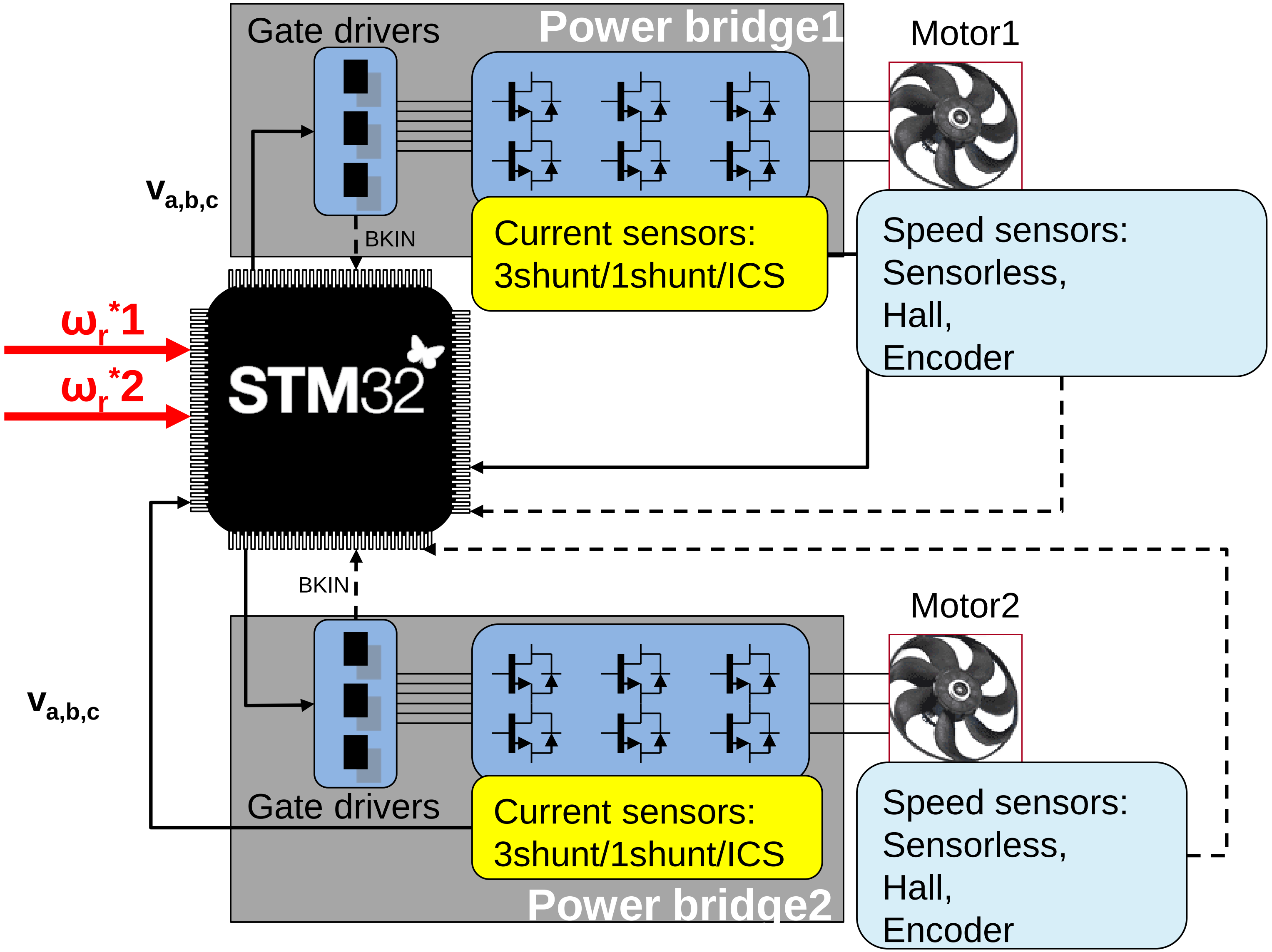


Single PMSM FOC – Block Diagram



life.augmented

Dual PMSM FOC – Block Diagram



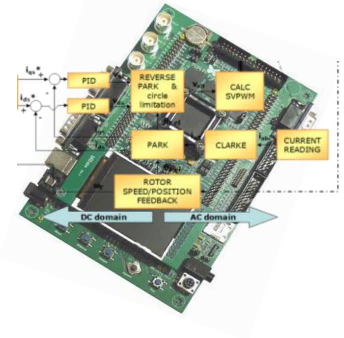
life.augmented

Motor Current Sensing – Why?

- The purpose of field oriented control is to regulate the motor phase currents.
- To do this the three motor phase current need to be measured.

Measured motor
phase current
Ia, Ib, Ic

FOC

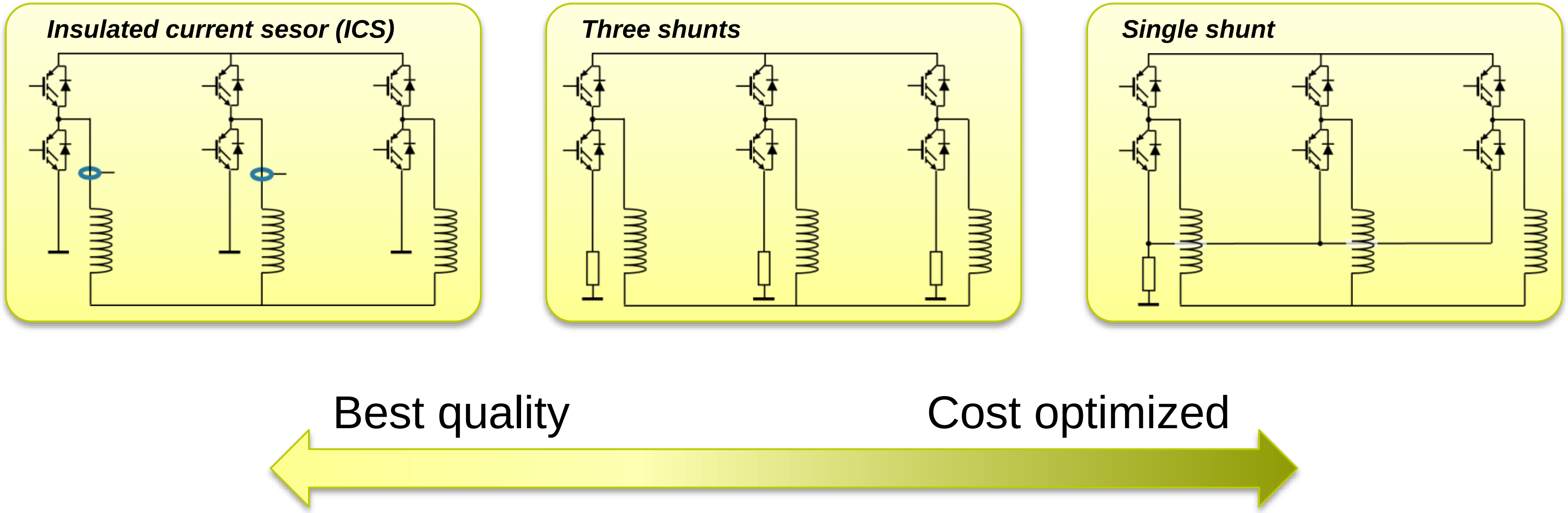


Current reference

Voltage command
PWM duty

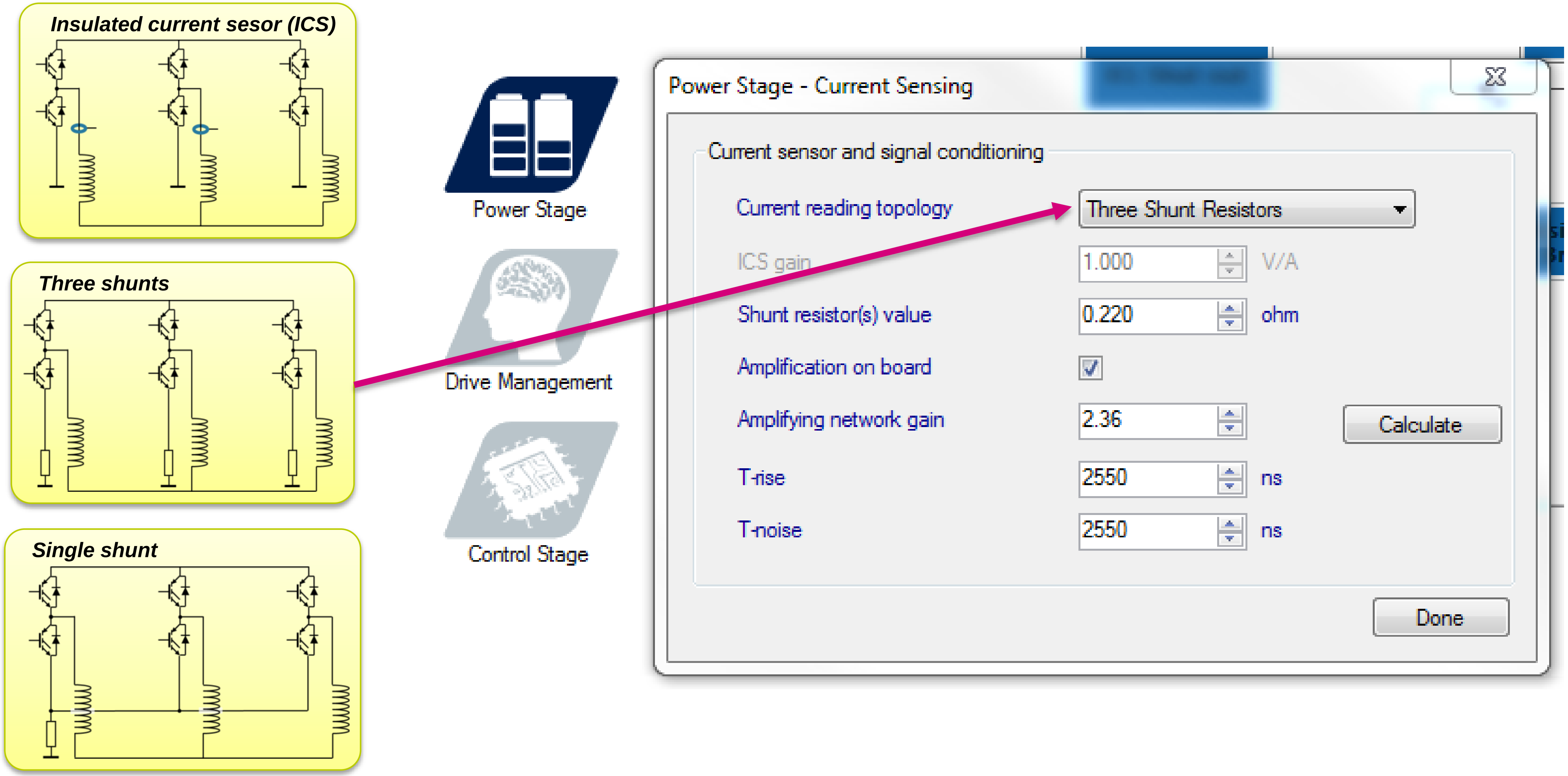
Current Sensing Topologies

- To measure the motor phase currents a conditioning network is required.
- The STM32 FOC SDK supports three current sensing network
  - Insulated current sensor (ICS)
  - Three shunts



Current Sensing Topologies

- According to the HW the current sensing topology can be selected in the power stage scetion of Workbench



STM32 PMSM FOC SDK v4.3

- STSW-STM32100 - **includes the PMSM FOC FW library, ST MC Workbench (GUI) and Motor Profiler (GUI),** allowing the user to evaluate ST products in applications driving single or dual Field Oriented Control of 3-phase Permanent Magnet motors **(PMSM)**, featuring STM32F3xx, STM32F4xx, STM32F0xx, STM32F1xx, STM32F2xx

Single / Dual
simultaneous FOC

STM32F3xx
STM32F4xx
STM32F0xx
STM32F1xx
STM32F2xx

Wide range of motor control algorithms implemented for specific applications

STM32 PMSM FOC SDK v4.3

Motor Profiler
Real time measurement and tuning of the motor

STM32F PMSM FOC FW library
MISRA C rules 2004 compliancy
Strict ANSI C compliancy

ST MC Workbench
Full customization and real time communication through PC software

life.augmented

Feature Set According To The Micro

Motor Control Firmware Library

STM32F4xx, STM32F3xx

STM32F103x HD/XL, STM32F2xx

STM32F103x LD/MD

STM32F100x, STM32F0xx

1shunt	Flux Weakening	IPMSM MTPA
Feed Forward	Sensor-less (STO + PLL)	Sensor-less (STO + Cordic)
Encoder	Hall sensors	Startup on-the-fly
ST MC Workbench support	USART based com protocol add-on	Max FOC(2) F100 ~11kHz F0xx ~12kHz

STM32F103x LD/MD:

- **3shunt** *F0 supported* New
- FreeRTOS
- ICS
- Max FOC(2) ~23kHz

STM32F103x HD/XL, STM32F2xx: New

- **Digital PFC(3)**
- Dual FOC
- Max FOC(2) F103 ~23kHz F2xx ~40kHz
- Max Dual FOC(2) F103 ~20kHz F2xx ~36kHz

STM32F4xx, STM32F3xx: New

- **Motor Profiler**
- HFI(1)
- Max FOC(2) F3xx ~ 30kHz F4xx ~50kHz
- Max Dual FOC(2) F3xx ~27kHz F4xx~45kHz

$$Foc\ rate = \frac{PWM\ freq}{Execution\ rate}$$

(1) High Frequency Injection
(2) Max FOC estimated in sensorless mode
(3) STM32F103xC/D/E/F/G and STM32F303xB/C

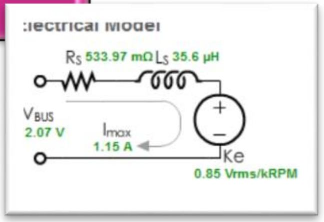
STM32 FOC SDK – Lab Session

Tools: configuration with PC SW ‘STMCWB’, ‘Motor Profiler’, IDEs

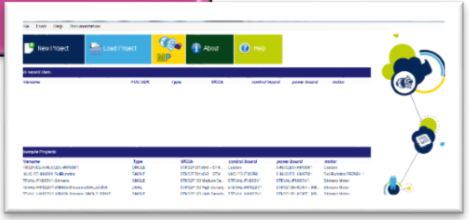
Motor Control – SDK – Workflow

Setup the HW

Use Motor specs or Identify the motor with Motor Profiler



Finalize the project with Workbench

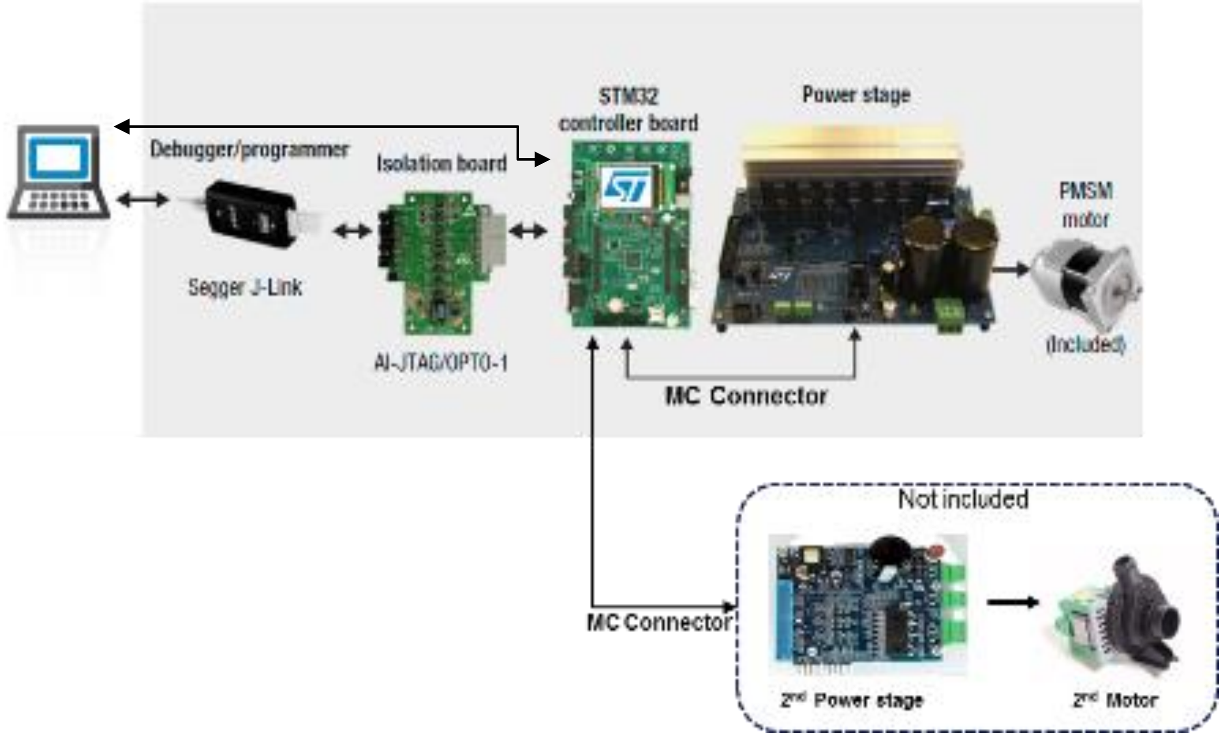


Debug and Real time monitoring

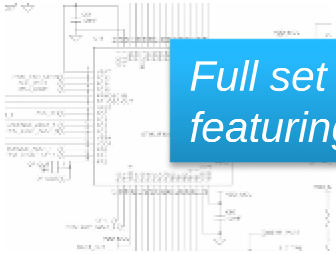
life.augmented

Motor Control – SDK – Workflow 1/4

- First step → ***Setup the Hardware,*** according the user's targets it is possible to choose the more suitable HW among the different ST “ready-to-start” evaluation boards.
- Setup them according the specification stated in each related user manual.
- Connect the board together (if required), power supply and plug your motor.



Flexible MC Platform



Full set of control board featuring all ST MCUs

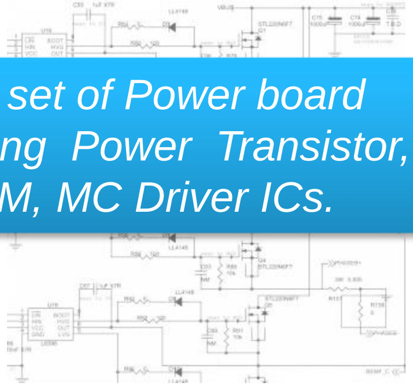
STM32XX-EVAL
Control board

NUCLEO-XX
Control board

+

X-NUCLEO-IHM09M1
Connector Adapter

MC Connector



Full set of Power board featuring Power Transistor, IPM, MC Driver ICs.

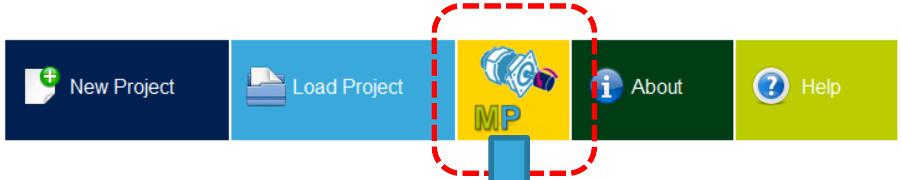
STEVAL-XX
Power board

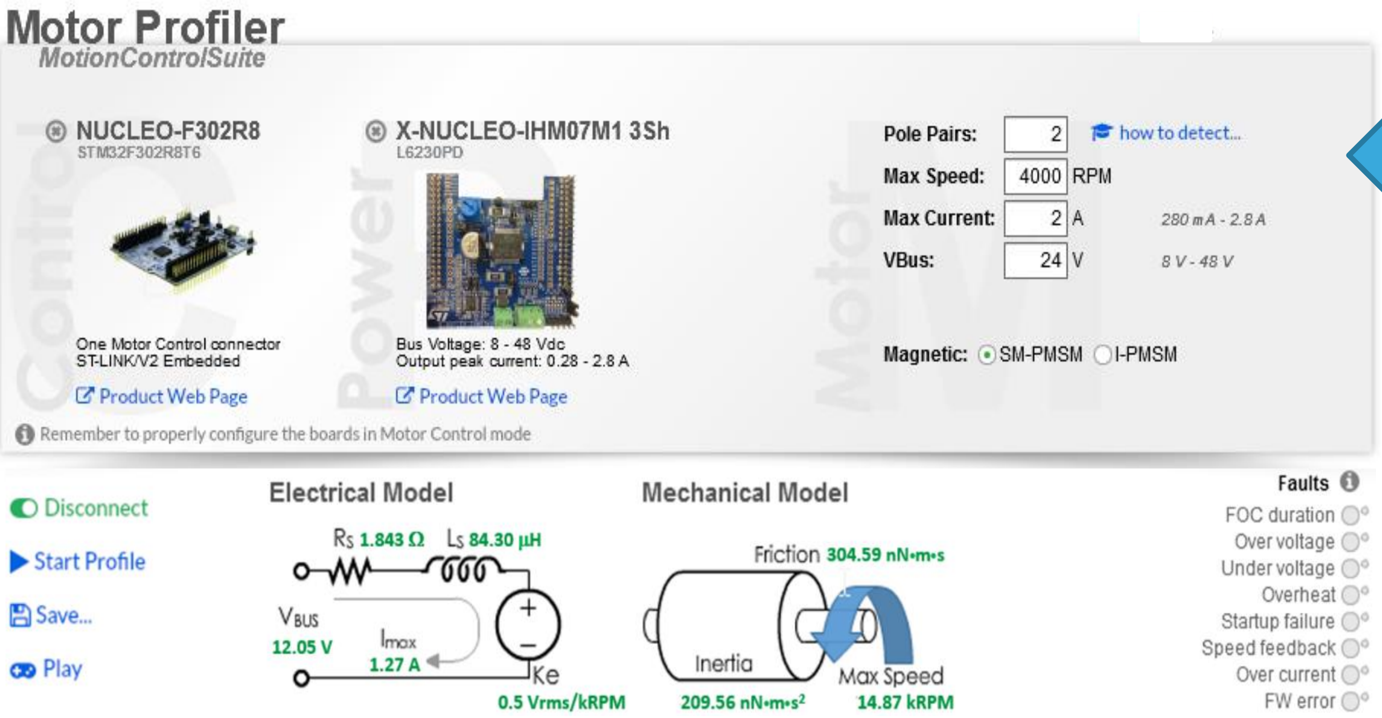
life.augmented

Motor Control – SDK – Workflow 2/4

- When the hardware is ready, if the user does not know the motor parameters, he can identify the motor.

- How? Using the ***Motor Profiler***!!





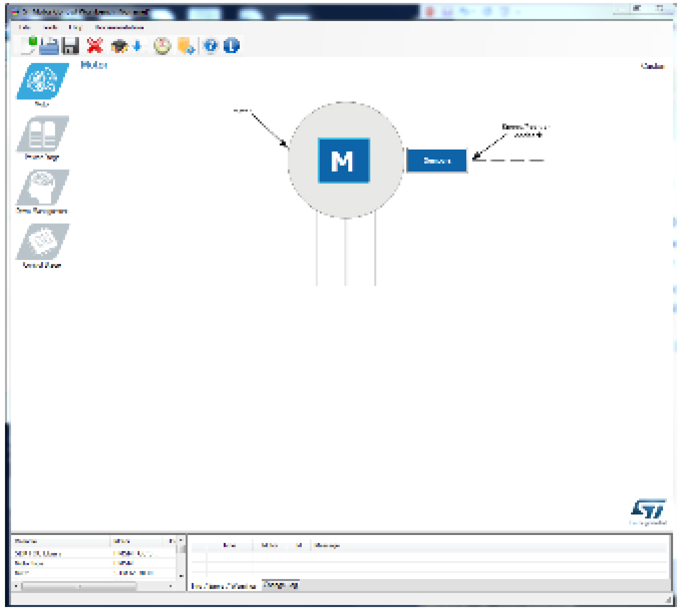
life.augmented

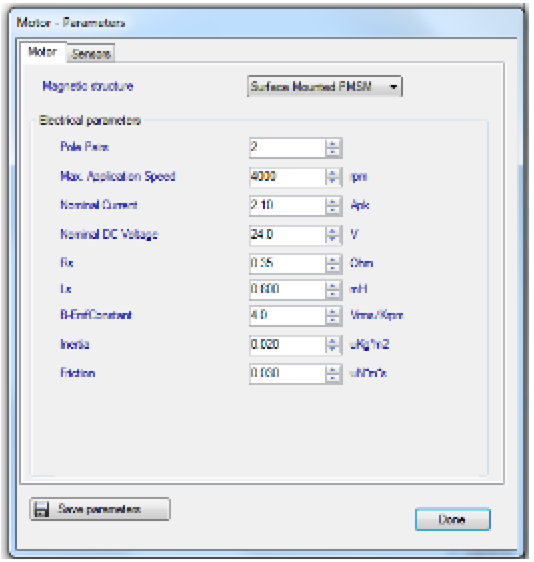
Motor Profiler

Set Up Motor Parameters

- ST MC Workbench – Motor section contains:
  - Motor parameters
  - Motor sensor parameters

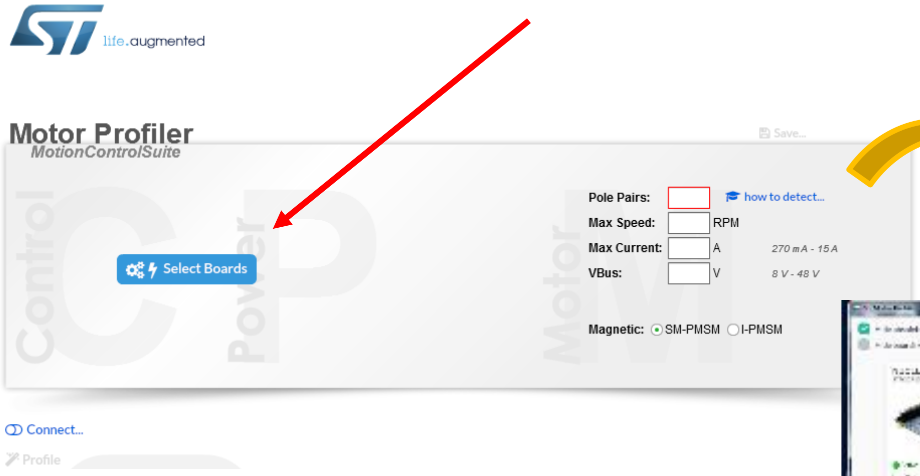
- For a custom project, the user can set all the parameters.





Setup Motor Profiler

- “Select Boards” button and a list of supported boards will be shown. The Motor Profiler feature can be used only in the systems listed there.

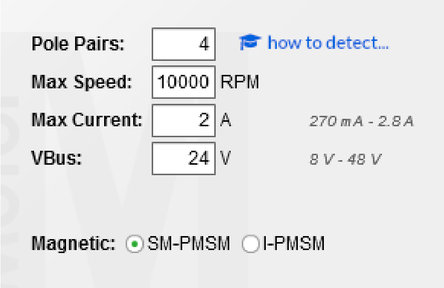


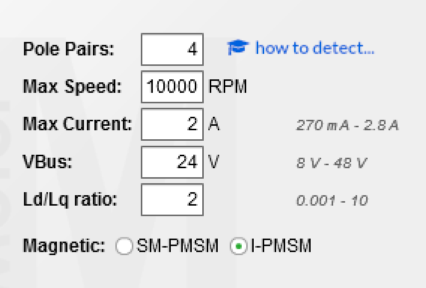
Setup Motor Profiler

Parameters set by the user:

- Motor pole pairs (Mandatory)
- Maximum application speed
  - Not mandatory, if not selected, the Motor Profiler will try to reach the maximum allowed speed.
- Maximum Peak Current
  - The maximum peak current delivered to the motor
- Expected bus voltage provided to the system.
- Selecting the kind of Motor
  - Surface Permanent Magnet motor SM-PMSM or Internal Permanent Magnet motor I-PMSM In this last case is necessary to provide also the Ld/Lq ratio as input.

SM-PMSM



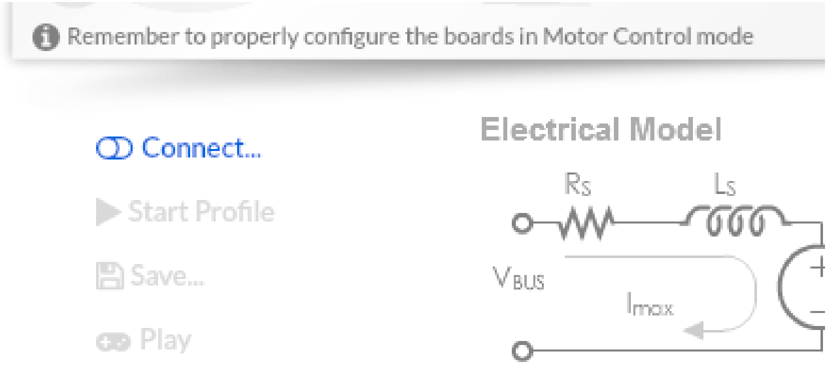


I-PMSM

life.augmented

Setup Motor Profiler

- Connect the HW chosen to the PC
- Click on the “Connect” button
  - If the communication has succeed
- Click on the “Profile” button

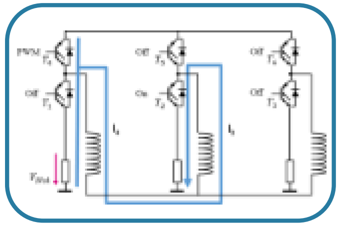


Run Motor Profiler

- Procedure will end in about 60 seconds.

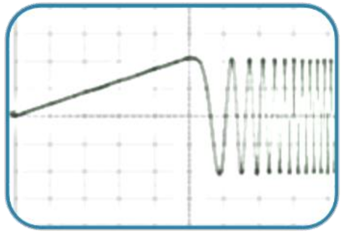
Motor stopped

- Rs measurement
- Ls measurement
- Current regulators set-up



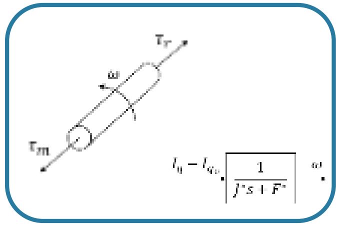
Open loop

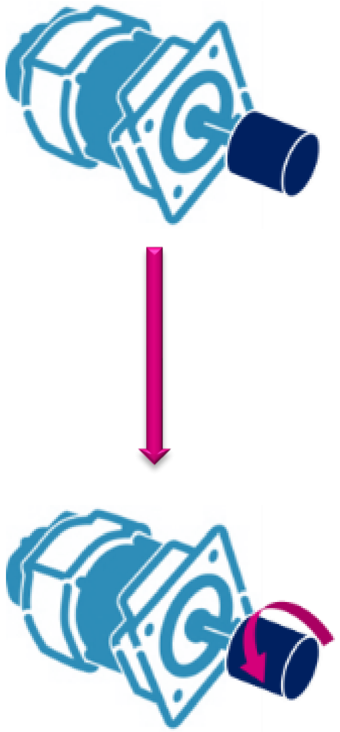
- Ke measurement
- Sensorless state observer set-up
- Switch over



Closed loop

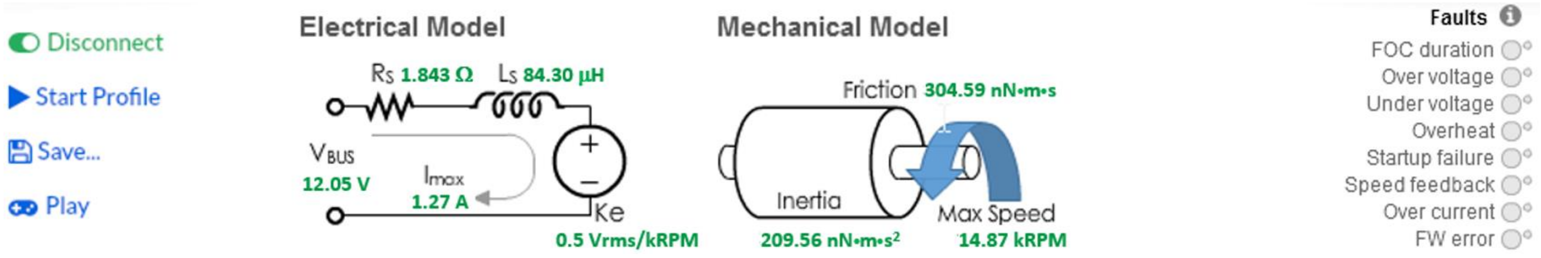
- Friction coefficient measurement
- Moment of inertia measurement
- Speed regulator set-up



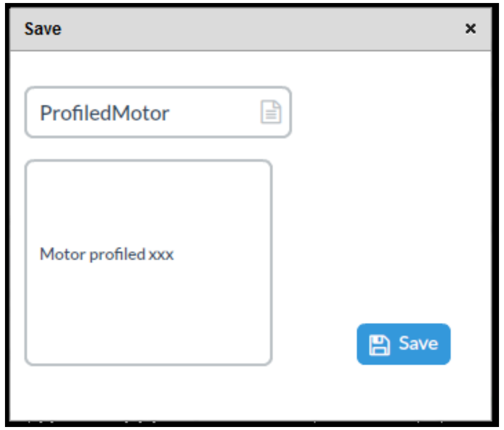


Motor Profiler Complete

- At the end of the procedure, the measured parameters will be shown on a dedicated window.



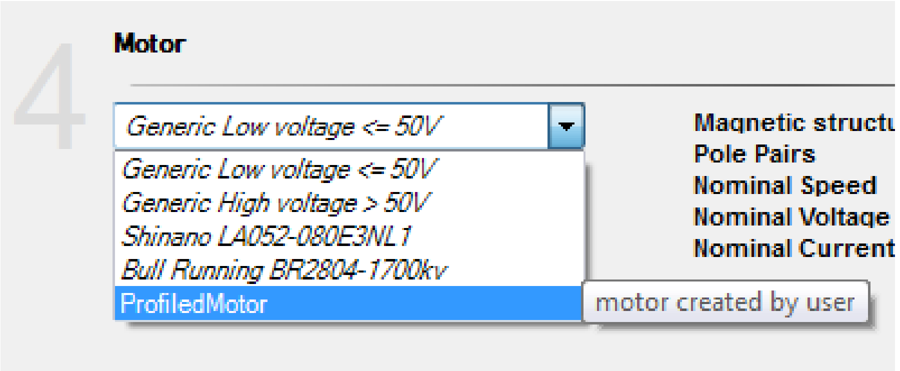
- It is possible to import them on the workbench project and save them for later use.



Motor Identified

- Motor Identified: user can start and stop the motor thorough “Start” and “Stop” button.

- it is possible to create ST MC Workbench new project with the profiled motor ,clicking “New Project”, in the Motor section the user can find



life.augmented

Motor Profiler

- The Motor Profiler algorithm is intended to be used for a fast evaluation of the ST three phase motor control solution (PMSM)
- Motor Profiler can be used only using compatible ST evaluation boards. Choosing the best ST HW according to the motor characteristics.
- The measurement precision can not be like when an instrumentation is used.
- Motor Profiler measurement cannot become significant for some motors, please see the limits reported in the software tool.

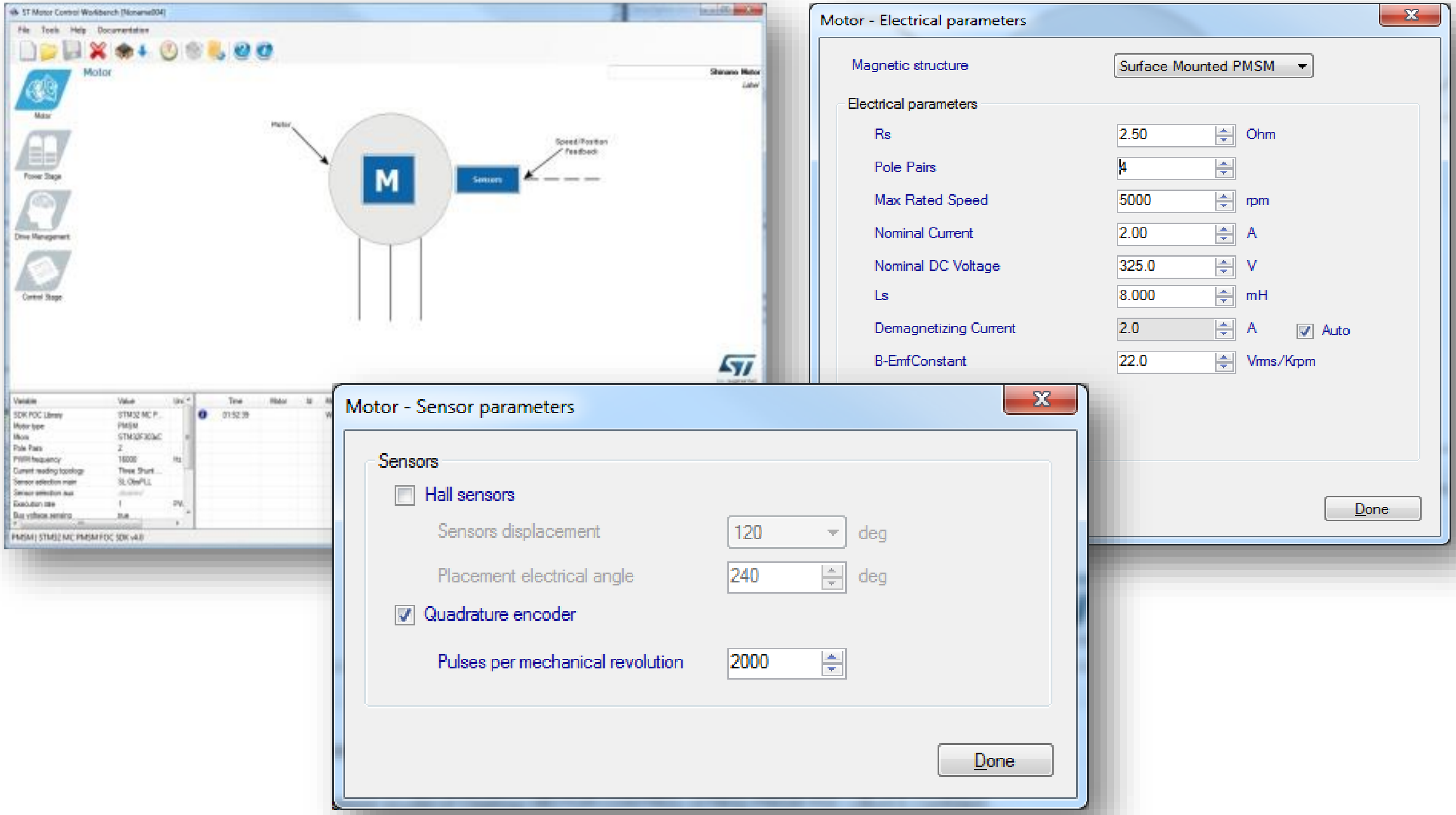
How To Manually Measure Motor Parameters

life.augmented

PMSM - Motor Parameters

STMCWB – Motor section contains:

- Electrical motor parameters
- Motor sensor parameters



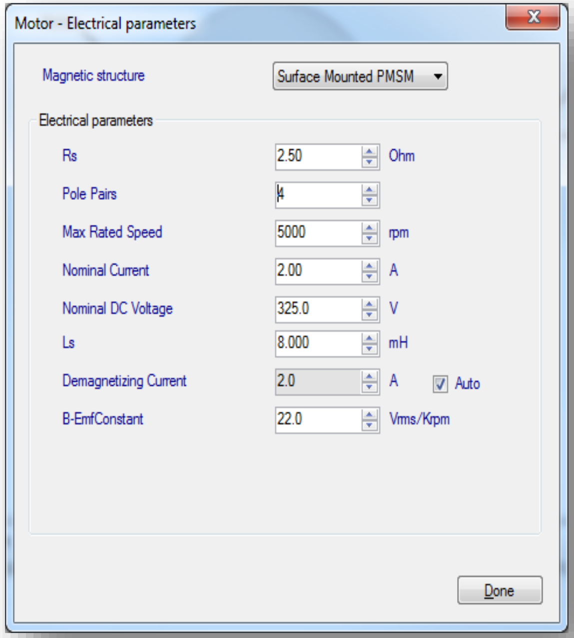
PMSM - Electrical Motor Parameters

- **Select either Internal PMSM or Surface Mounted PMSM according to the magnetic structure of your motor**

- **If you don't have this information you need to measure both Ld and Lq inductance for verifying it**

- **IF** **2*(Lq-Ld)/(Ld+Lq) <15%** → **SM-PMSM**

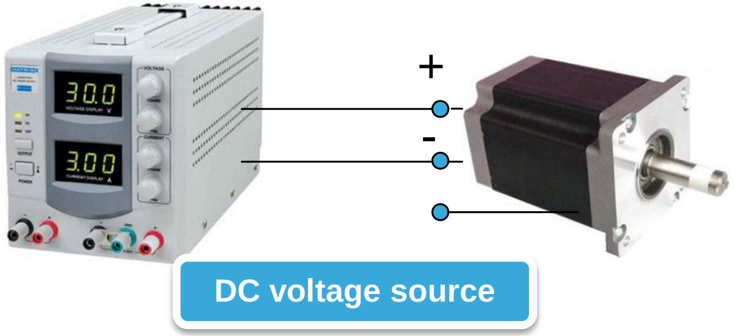
- **See next slides for learning how to measure motor inductances**



PMSM – Pole Pairs Number

- **Usually, it's provided by motor supplier**

- **In case it's not or if you'd like to double check it**
  - Connect a DC power supply between two (of the three) motor phases and provide up to 5% of the expected nominal DC bus voltage (you may also set current protection to nominal motor current)
  - Rotate the motor with hands (you should notice some resistance)

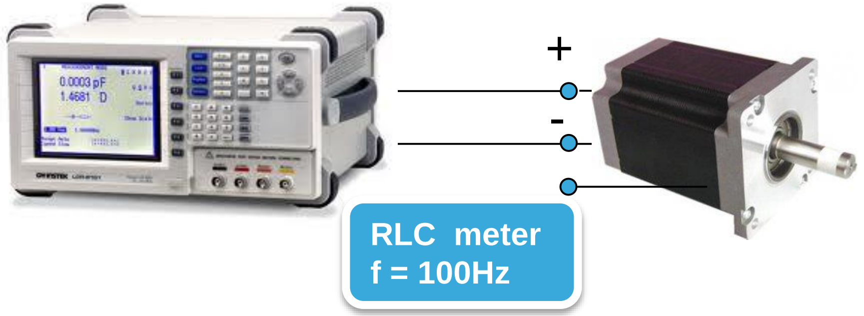


  - The number of rotor stable positions in one mechanical turn represents the number of pole pairs

How To Measure Motor Inductance 1/3

- **In case of SM-PMSM, the phase inductance does not depends on rotor position. In this case Ls notation is also utilized**

- **If you have a RLC meter**
  - Connect it phase-to-phase and measure series R and L at 100Hz (make sure rotor doesn't move)
  - Repeat 4*number of pole pairs times:
    - Turn the rotor by 360/(4*number of pole pairs) mechanical degrees,
    - Wait for new measurements to get stable
    - Read new measurement



- IF **2*(Lq-Ld)/(Ld+Lq) <15%** → **SM - PMSM**

  In this case, in the Workbench you can use for Rs and Ls half of the values read on the instrument STMCWB requires phase to neutral value

life.augmented

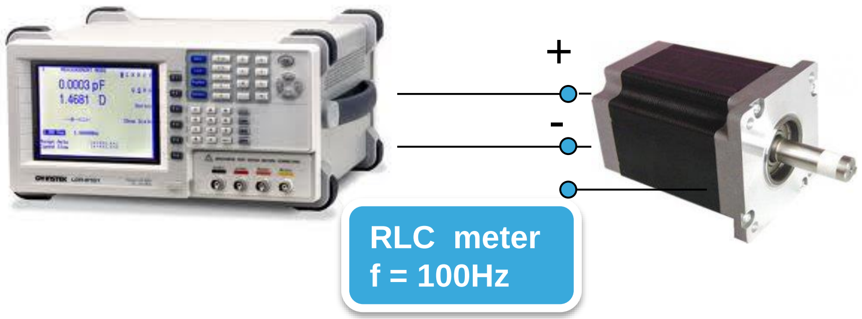
with RLC meter

How To Measure Motor Inductance 1/3

- IF **2*(max(L)-min(L))/(max(L)+min(L)) > 15%** → **I - PMSM**

In the Workbench you can set Ld equal to minimum measured value divided by 2 (STMCWB requires phase to neutral value), set Lq equal to maximum measured value divided by 2

Set Rs equal to average measured resistance divided by two

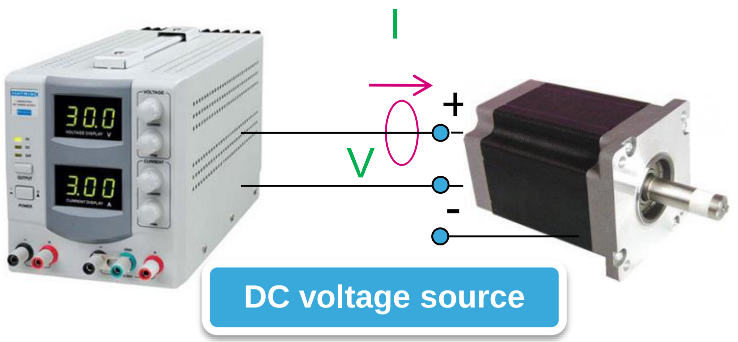


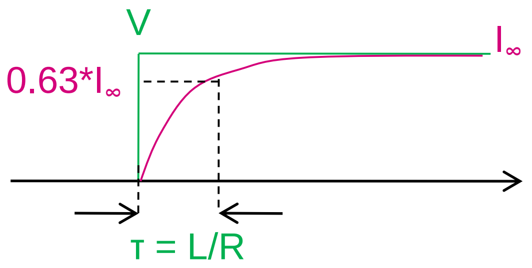
with RLC meter

How To Measure Motor Inductance 2/3

- **If you don't have a RLC meter**
  - For Rs, measure the DC stator resistance phase-to-phase and divide it by two
  - Once measured Rs, it's necessary to measure L/R time constant between two motor phases.

    - Connect DC voltage between two motor phases
    - Connect oscilloscope
    - Increase the voltage up to the value where the current equals the nominal one, rotor with align
    - Don't move rotor any more
    - Disable current protection of DC voltage source
    - Unplug one terminal of the voltage source cable without switching it off
    - Plug the voltage source rapidly and monitor on the scope the voltage and current waveform
    - The measurement is good if the voltage is a nice step and the current increase like I∞ * (1-e- t *L/R)
    - Measure the time required to current waveform to rise up to 63%
    - This time is Ld/Rs constant. Multiply it by Rs and you'll get Ld value





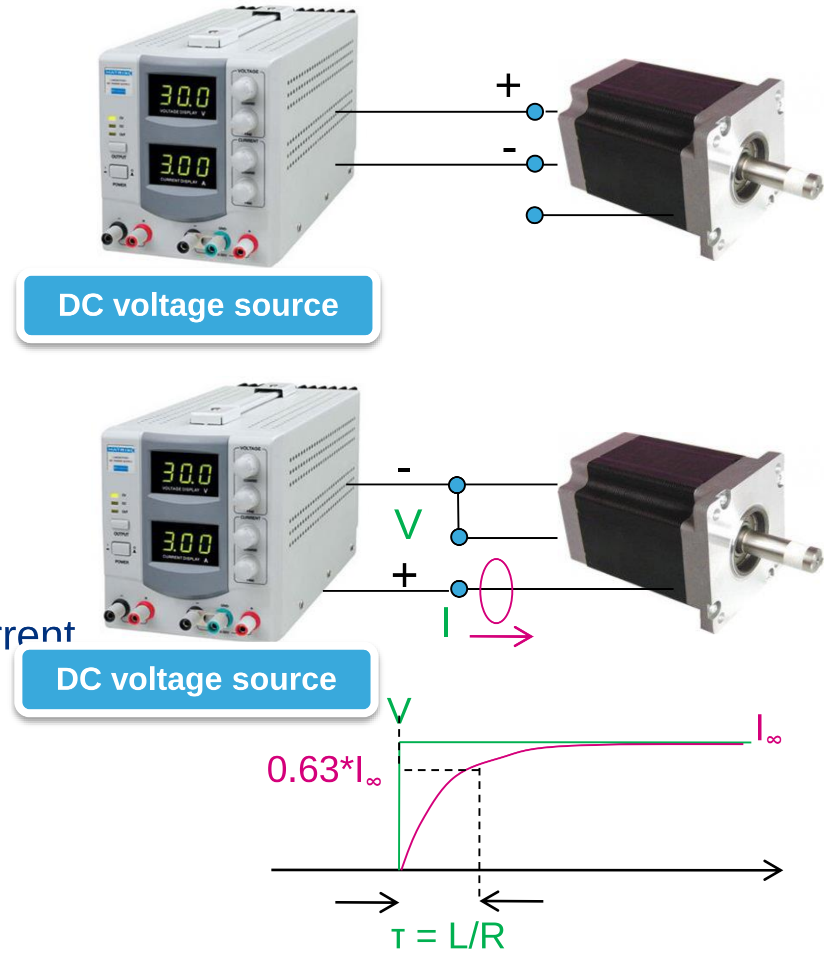
without RLC meter

life.augmented

How To Measure Motor Inductance 3/3

- **Once measured Rs, it's necessary to measure Lq/Rs time constant between two motor phases**.

  - Connect DC voltage between two motor phases
  - Connect oscilloscope
  - Increase the voltage up to the value where the current equals the nominal one, rotor with align
  - Lock the rotor in this position (so that it can not move anymore)
  - Change DC voltage source connections as shown in the second figure
  - Unplug one terminal of the voltage source cable without switching it off
  - Plug the voltage source rapidly and monitor on the scope the voltage and current
  - The measurement is good if the voltage is a step and the current increase like $I_\infty * (1\text{-}e^{-t *L/R})$
  - Measure the time required to current waveform to rise up to 63%
  - This time is Lq/Rs constant. Multiply it by 2Rs/3 and you'll get Lq value



without RLC meter

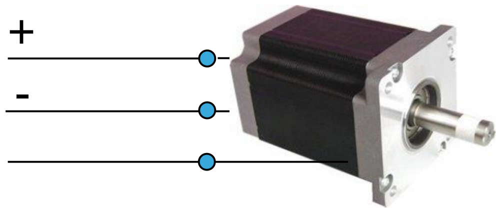
life.augmented

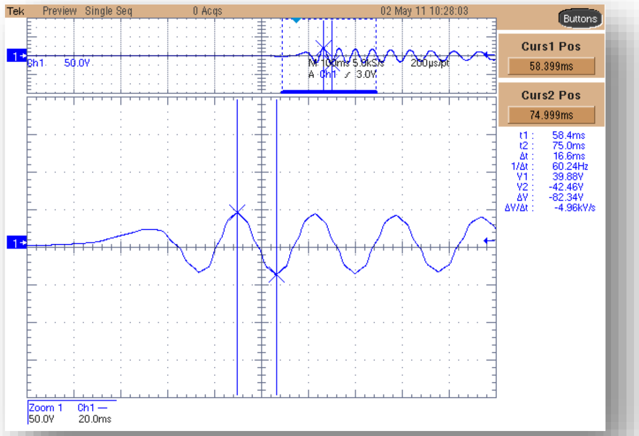
How To Measure Bemf 1/2

- The B-emf constant represents the proportionality constant between the mechanical motor speed and the amplitude of the B-emf induced into motor phases:

$$V_{Bemf} = K_e \cdot \omega_{mec}$$

- To measure K_e , usually is sufficient to turn the motor with your hands (or using drill or another motor mechanically coupled) and look with an oscilloscope to phase-to-phase induced voltage (V_{Bemf})





- If you have no access to the rotor (e.g. in compressor applications) see next slide

- Measure VBemf frequency (FBemf) and peak-to-peak amplitude (VBemf –A)

- Compute Ke in Vrms / Krpm:

$$K_e = \frac{V_{Bemf-A}\ [V\ peak-to-peak] \cdot pole\ pairs\ number \cdot 1000}{2 \cdot \sqrt{2} \cdot F_{Bemf}\ [Hz] \cdot 60}$$

life.augmented

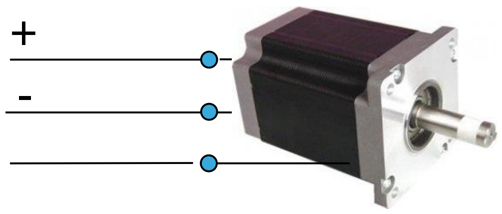
Access to rotor

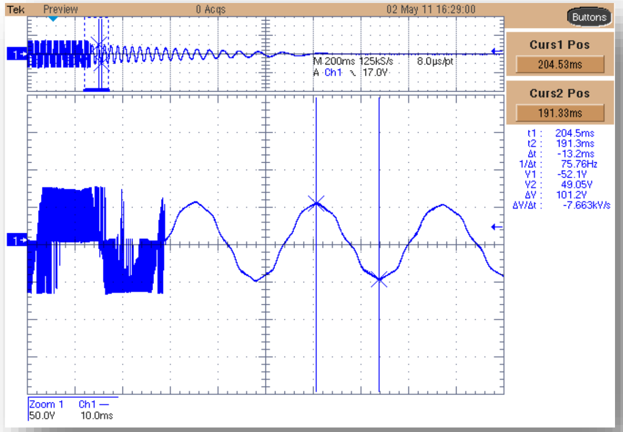
How To Measure Bemf 2/2

- If you have no access to rotor (e.g. in compressors) follow this procedure:
  - Configure power stage (see later)
  - Configure drive parameters for sensor-less as described in slides 16-19 but
    - Set current ramp initial and final values equal to motor nominal current value
    - Set the speed ramp duration to 5000ms and speed ramp final value to around 50% of maximum application speed
    - Set minimum start-up output speed higher than speed ramp final value
  - Configure control stage
  - Start the motor ramp-up and look with the oscilloscope to the

  voltage between two motor phases
  - When the driving signal are switched off, rotor was probably

  moving then look to the B-emf





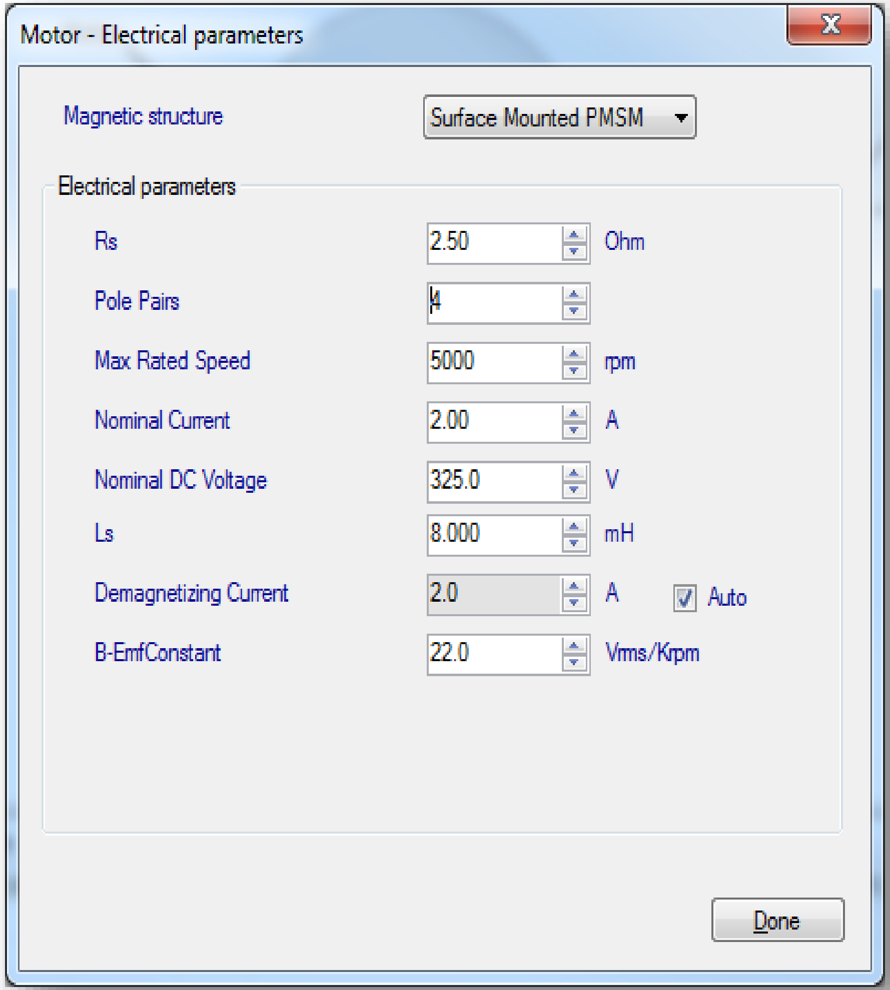
$$K_e = \frac{V_{Bemf-A}\ [V\ peak-to-peak] \cdot pole\ pairs\ number \cdot 1000}{2 \cdot \sqrt{2} \cdot F_{Bemf}\ [Hz] \cdot 60}$$

life.augmented

No Access to rotor

Other Electrical Parameters

- **Max rated speed (rpm)**
  - Should be provided by motor producer (if not, set it to max application speed)
  - Maximum motor rated speed above which motor can get damaged
  - Maximum application speed must be lower than this value
- **Nominal current (in A, 0-to-peak)**
  - Motor rated current, must be provided by motor producer
  - It will be used to limit the imposed motor phase current during normal operation
- **Nominal DC voltage**
  - Nominal DC bus voltage from which the motor should run, must be provided by motor producer
- **Demagnetizing current**
  - Rotor demagnetizing current, may be provided by motor producer (if not use default value, i.e. motor nominal current)
  - Used to limit the amount of target negative Id during flux weakening



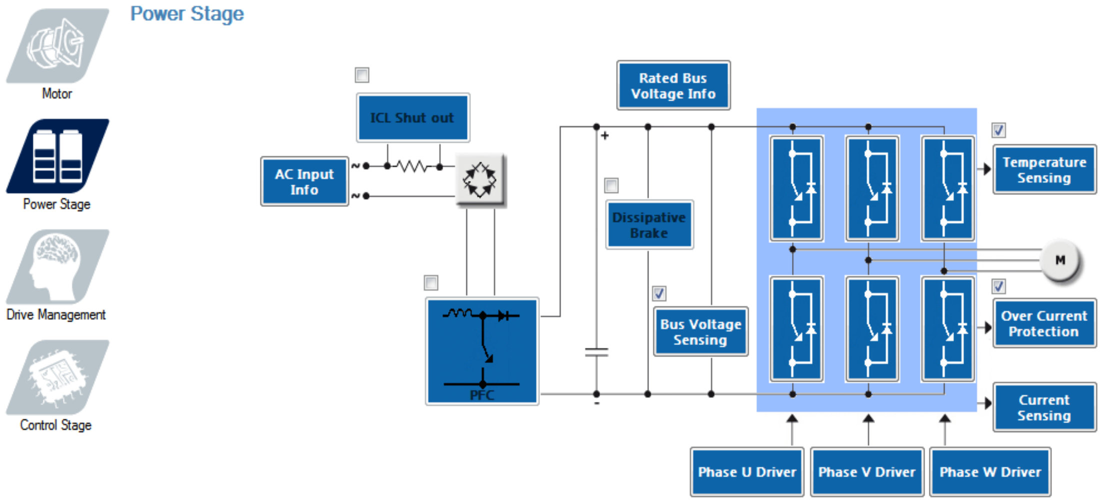
life.augmented

Motor Control – SDK – Workflow 3/4

- With Motor Profiler the motor is running but the user can develop his own code!

- **Finalize the MC project** using Workbench and use your favorite IDE to develop your code.

MC Workbench



IAR SYSTEMS

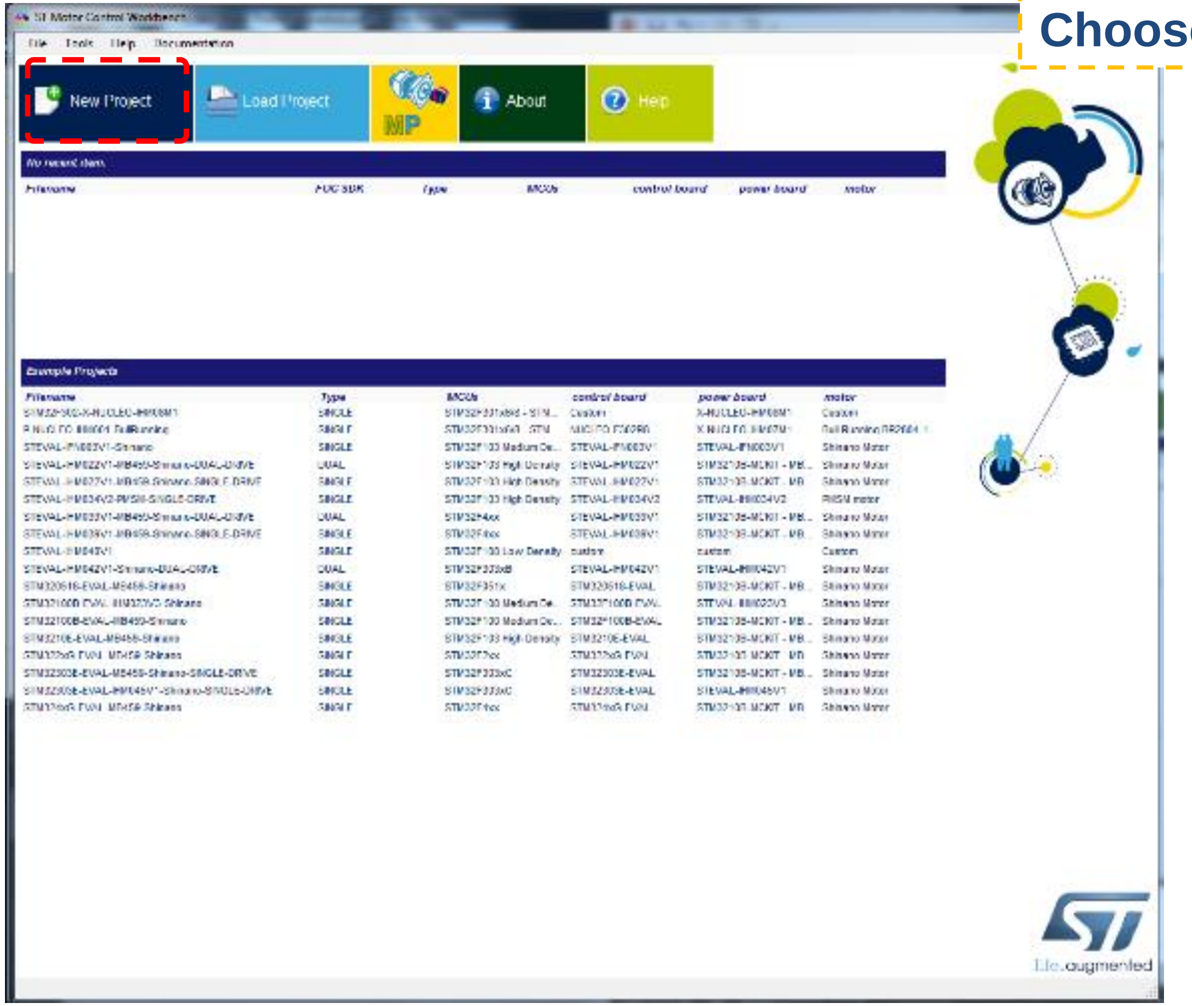
KEIL Tools by ARM

ac6 Assistance Conseil Systemes

life.augmented

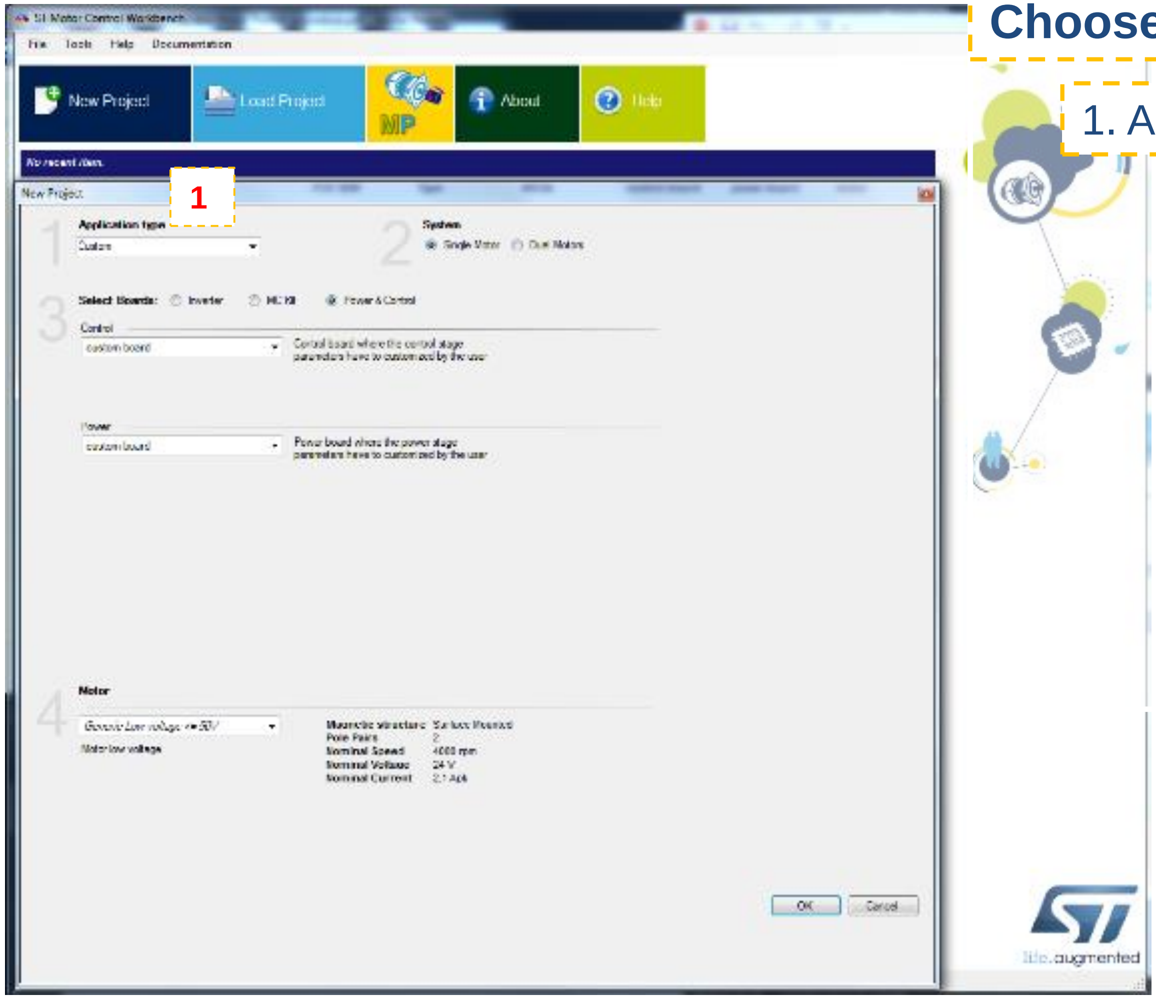
Create A New WB Project Based On The ST Evaluation Board

Choose: New Project



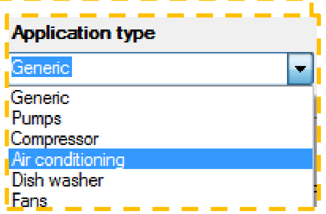
life.augmented

Create A New WB Project Based On The ST Evaluation Board

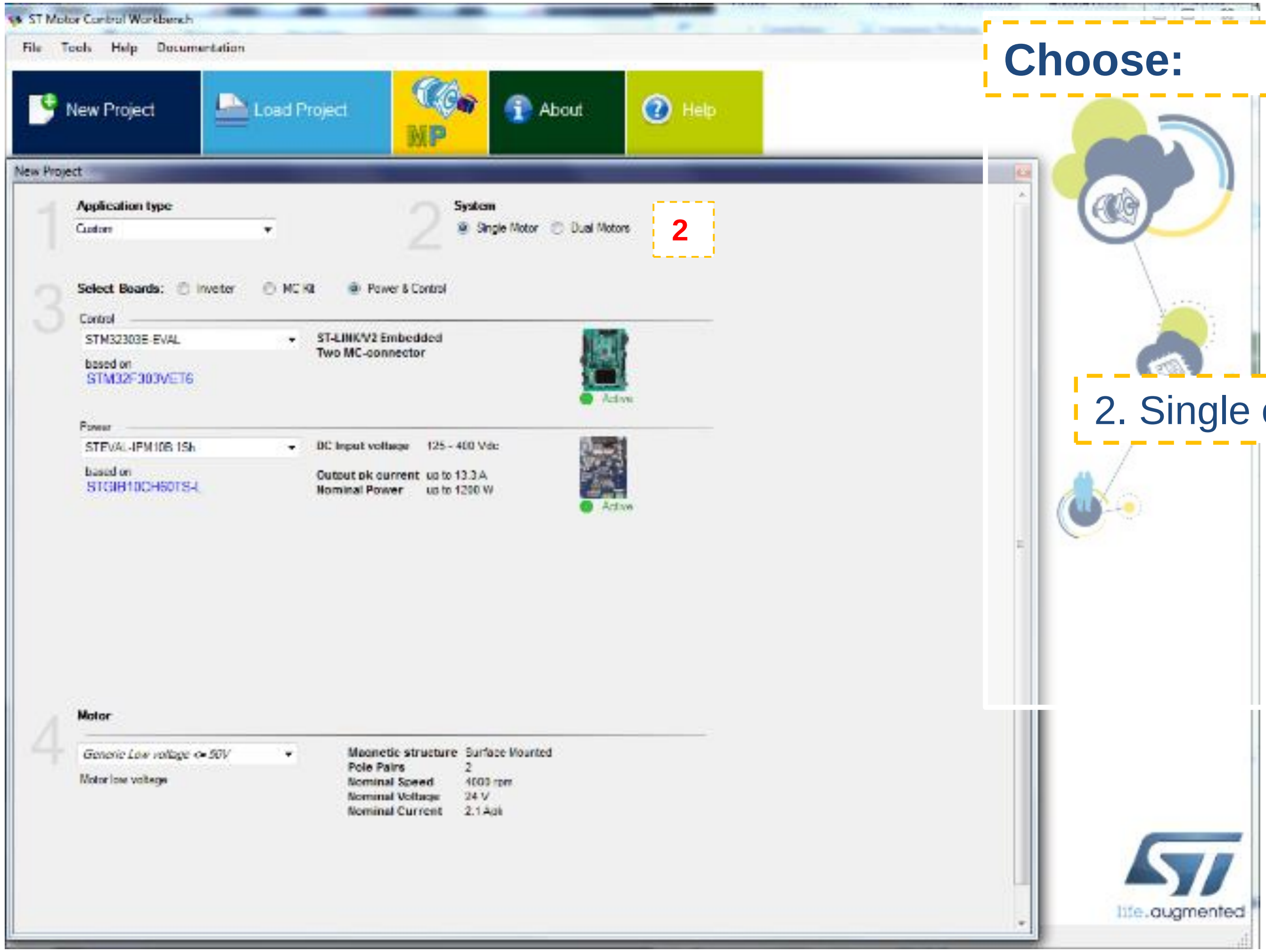


Choose:

1. Applications



Create A New WB Project
Based On The ST Evaluation Board

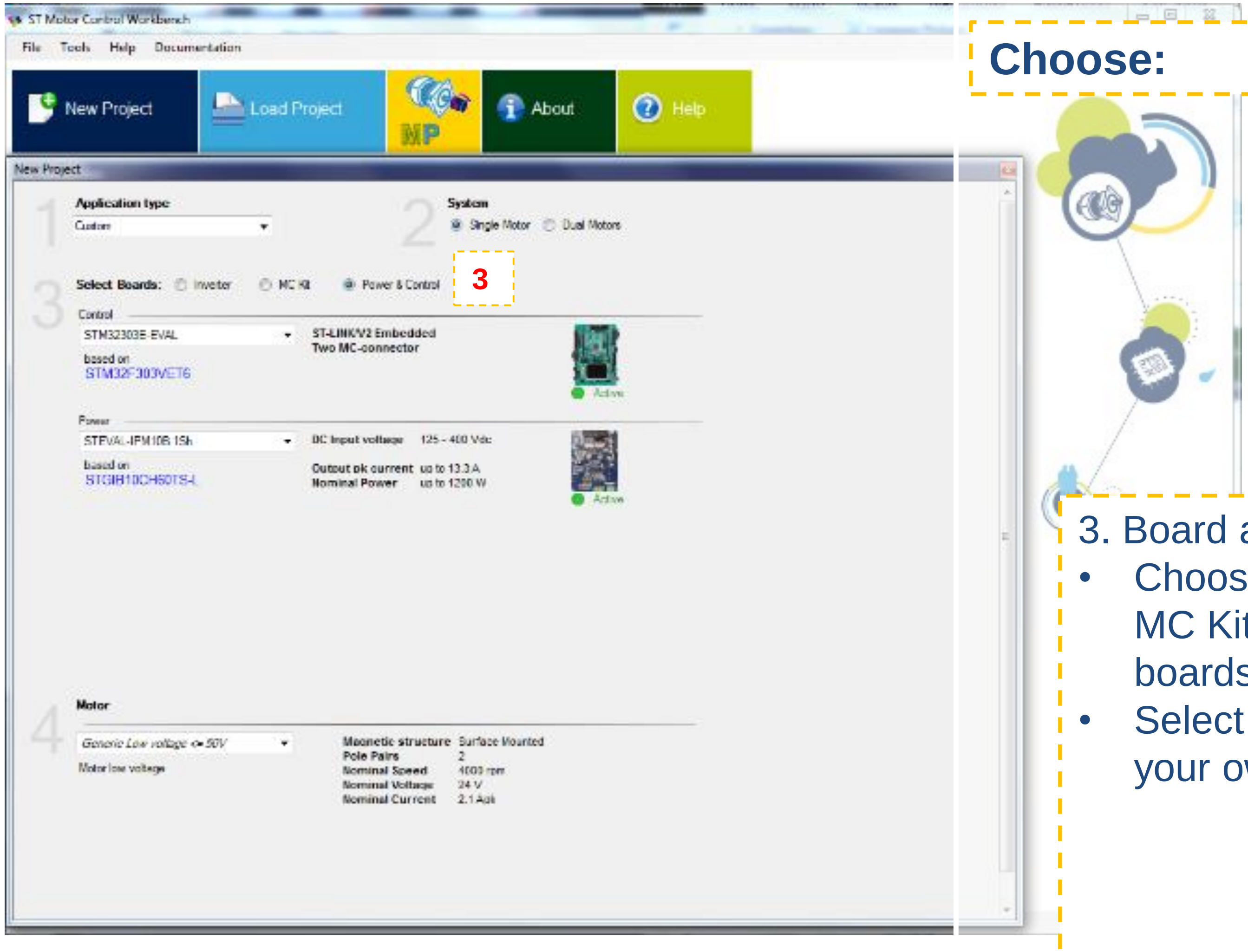


Choose:

2. Single or dual motor

life.augmented

Create A New WB Project
Based On The ST Evaluation Board

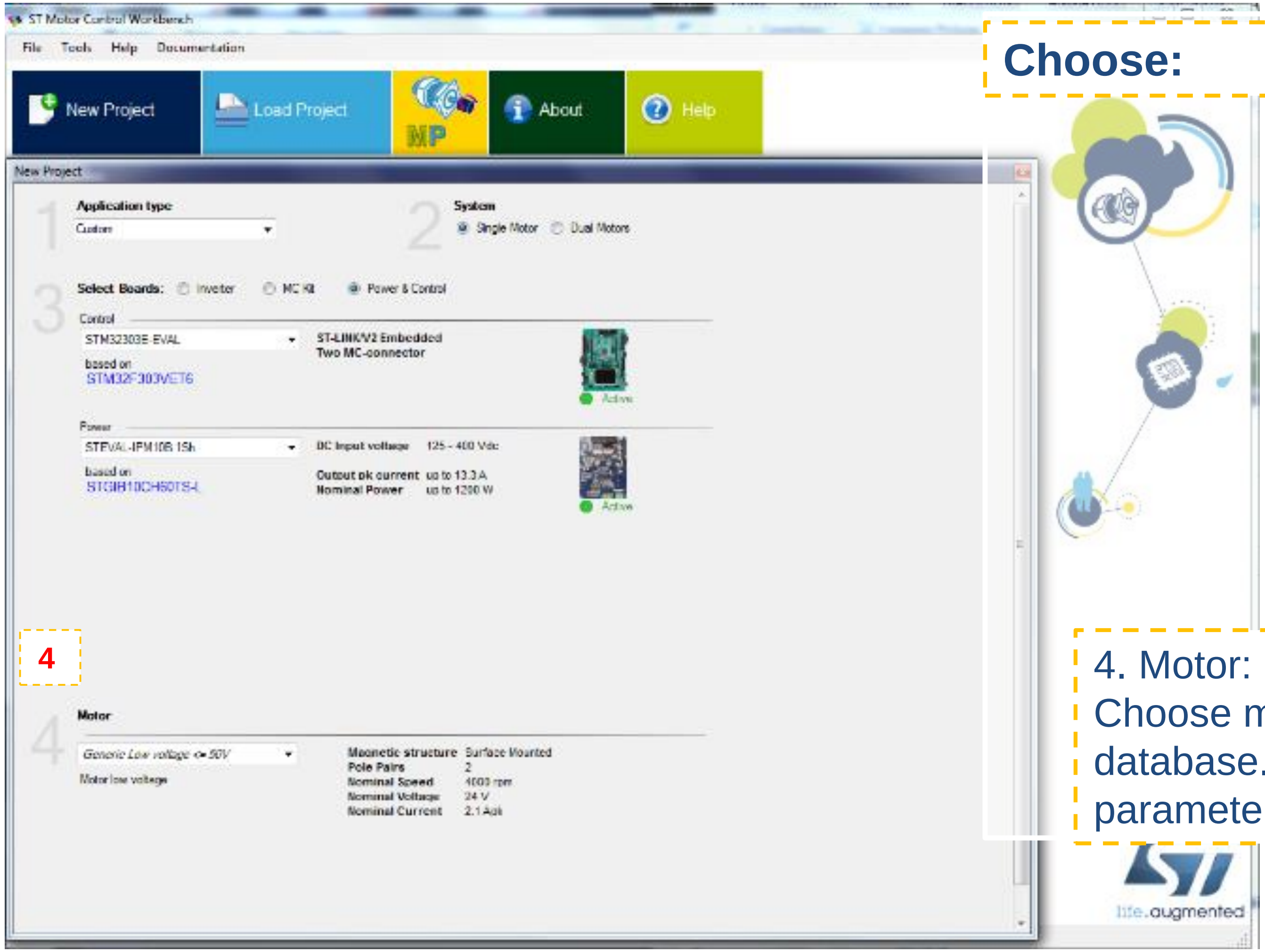


Choose:

3. Board approach:

- Choose if you are using Inverter, MC Kit or Power plus Control boards.
- Select the board used or create your own custom board.

Create A New WB Project Based On The ST Evaluation Board



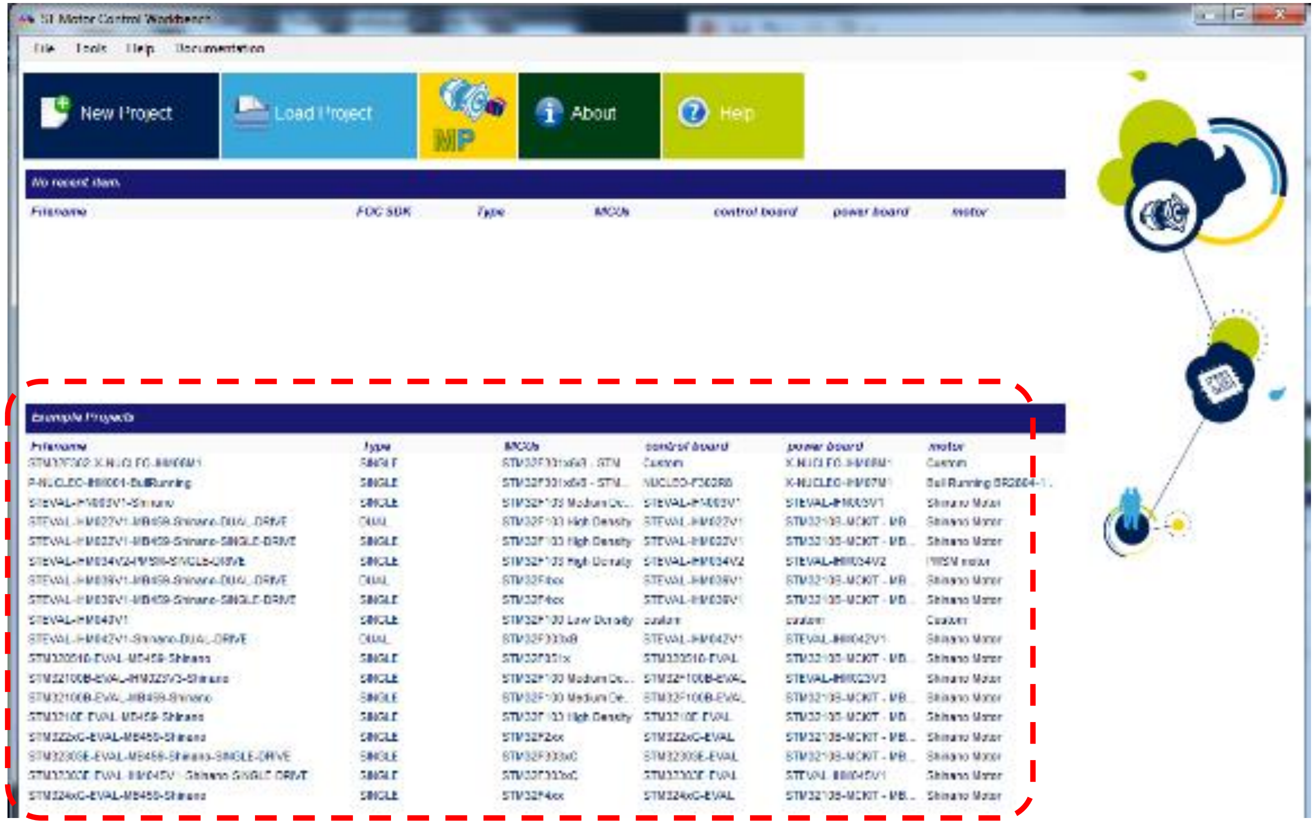
Choose:

4. Motor:
Choose motor from a motor database. (You can save your motor parameters from your project.)

life.augmented

Create A New WB Project Based On An Example

- Choose the WB example project that best fits your needs.
  - Choose the one with the same name of the ST evaluation board you are using, or
  - choose the one with the same microcontroller you are using.



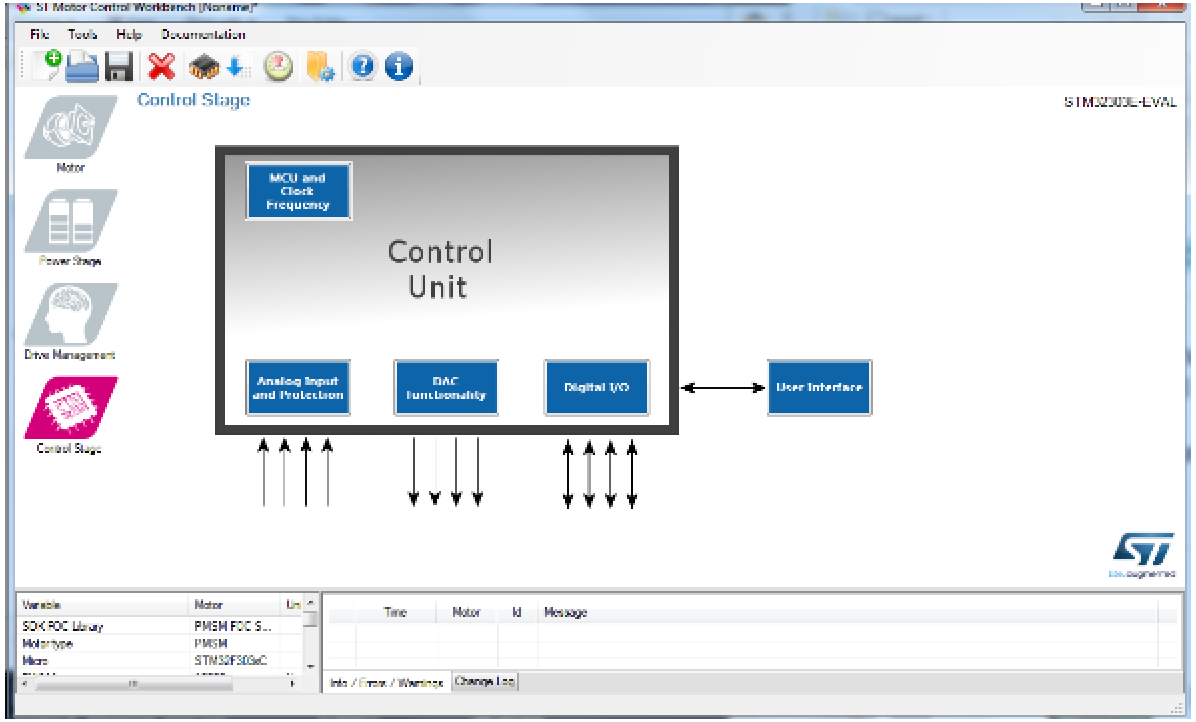
life.augmented

Create A New WB Project

- Starting from the board selection or example project, the control stage parameters will be populated with the correct values.

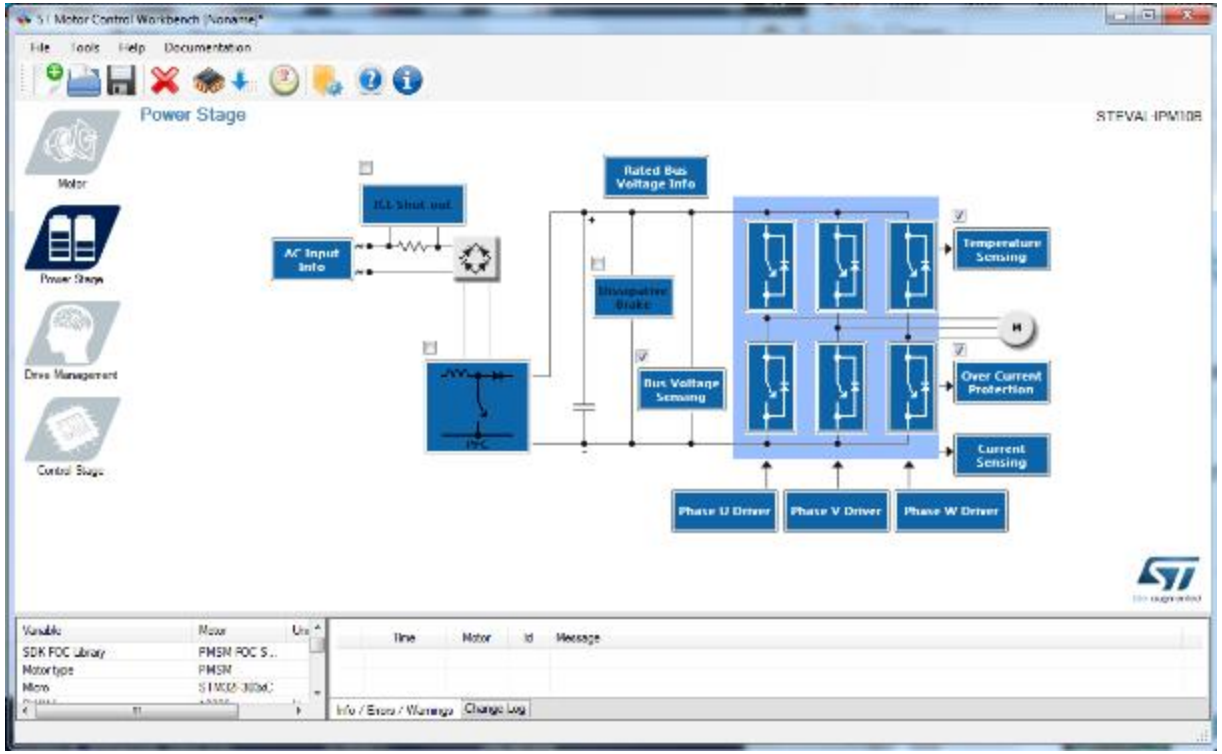
- For a custom project, the user can set all the parameters.

STM32303E-EVAL



Set Up Power Stage

- Starting from the board selection or example project, the power stage parameters will be populated with the correct values.
- For a custom project, the user can set all the parameters.

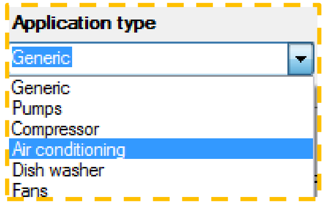


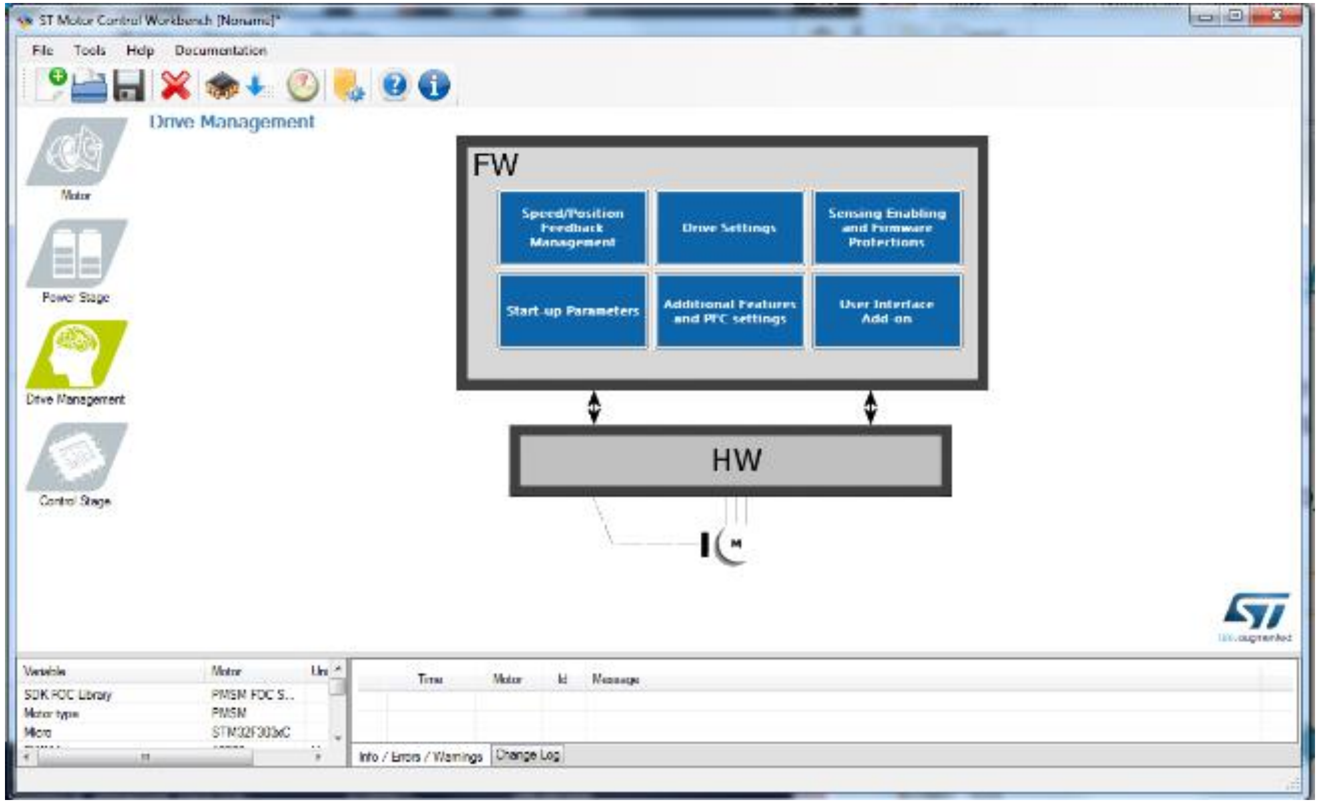
Set Up Drive Parameters

- Starting from the board selection according to the chosen application, drive parameters will be populated with the correct values.

- For a custom project, the user can set all the parameters.

Applications

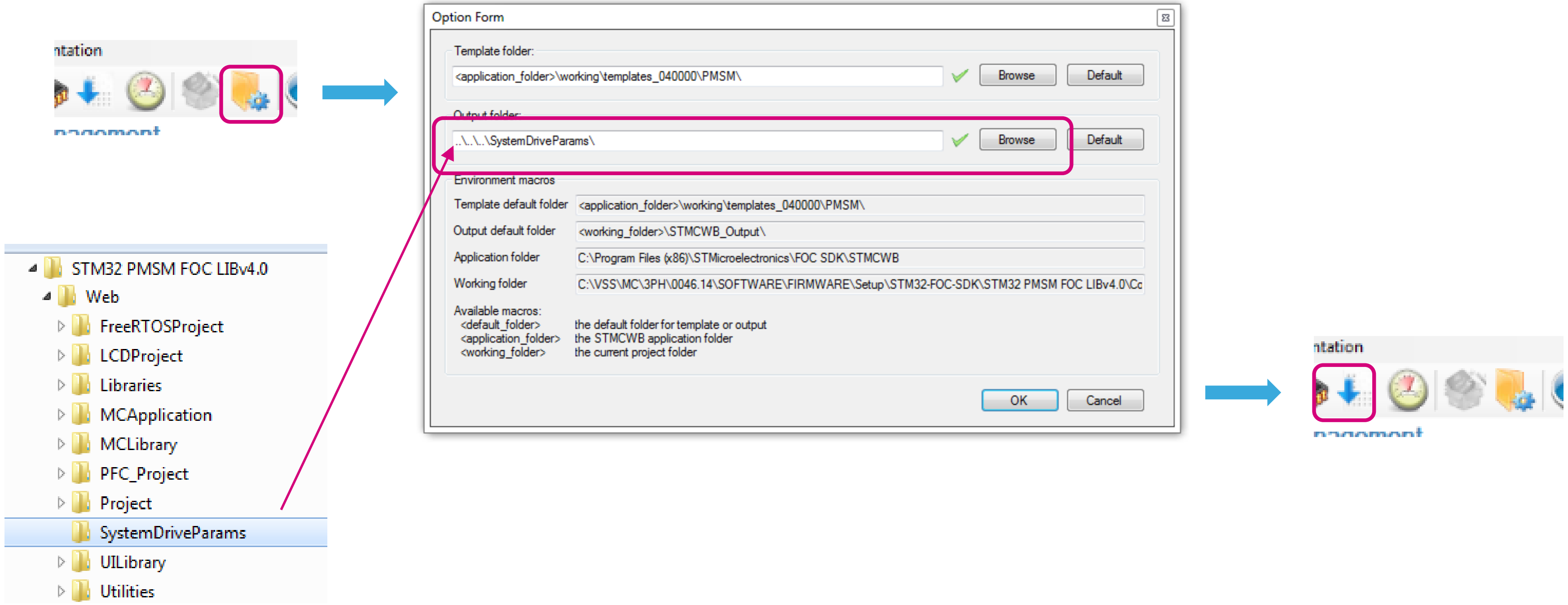




life.augmented

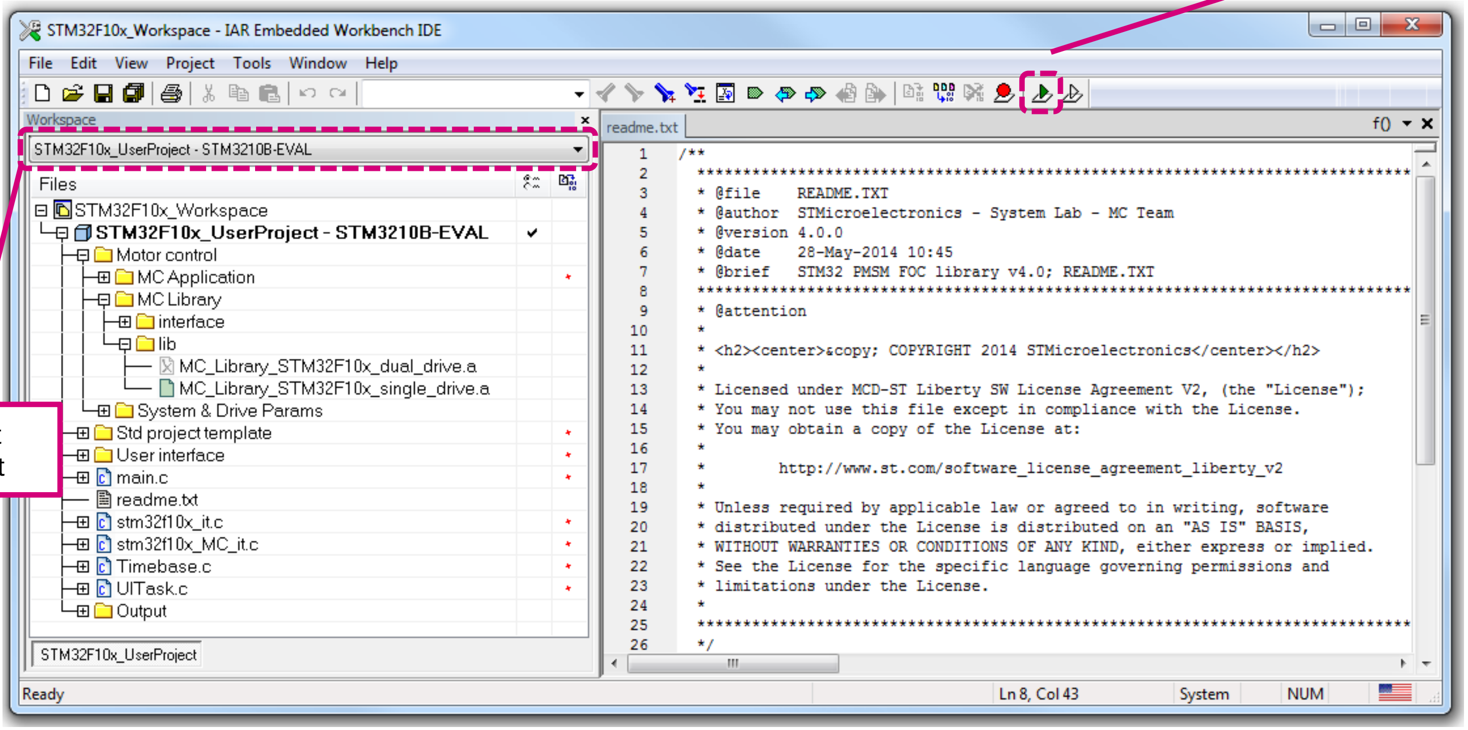
Parameter Generation

- Once all the parameters have been entered in the ST MC Workbench, select the output path in the option form and choose ‘**SystemDriveParams**’ present in the FW working folder.
- Click on the ‘**Generation**’ button to configure the project.



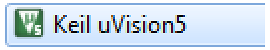
Compile And Program The MCU

- Run the IAR Embedded Workbench. IAR Embedded Workbench
- Open the IAR workspace (located in Project\EWARM) folder according to the microcontroller family (e.g. STM32F10x_Workspace.eww for STM32F1).
- Select the correct user project from the drop-down menu according to the control stage used (e.g. STM32F10x_UserProject - STM3210B-EVAL).
- Compile and download.

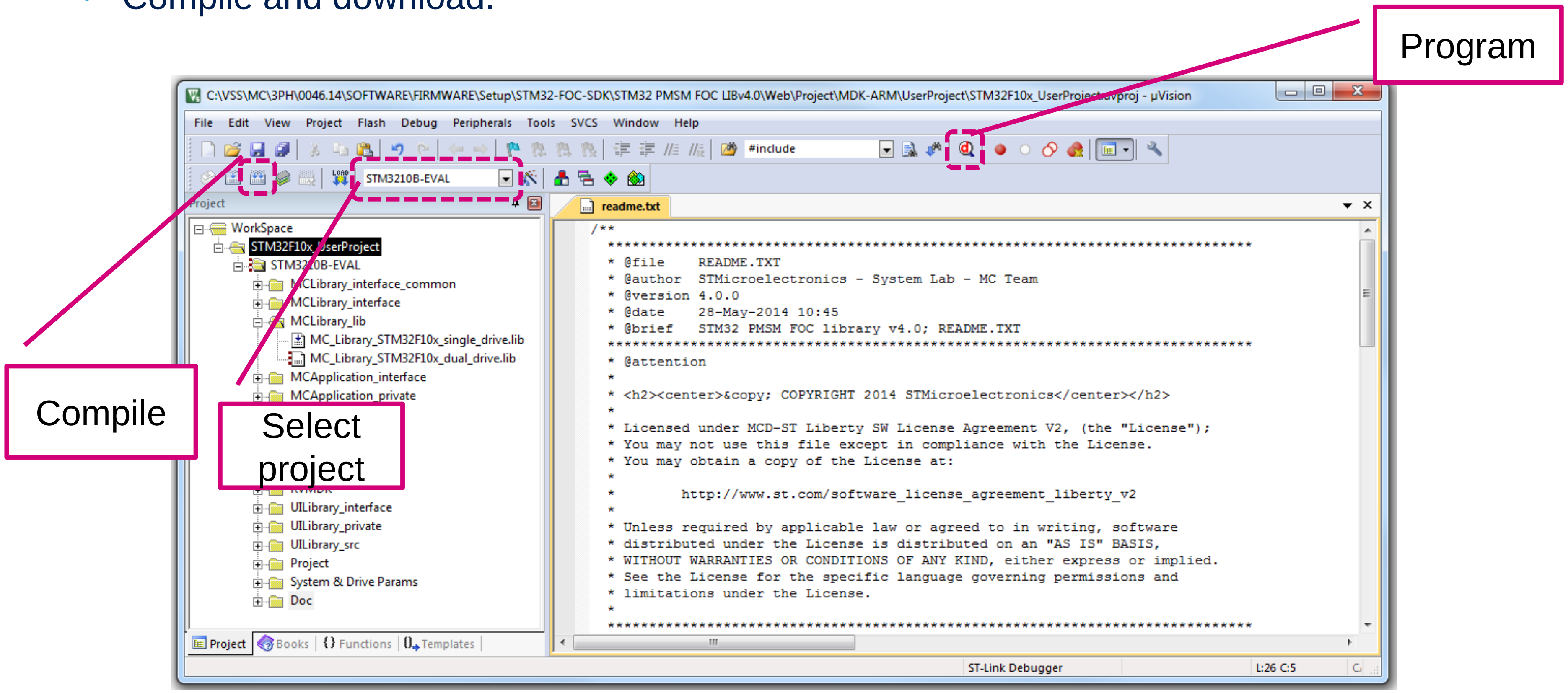


life.augmented

Compile And Program The MCU



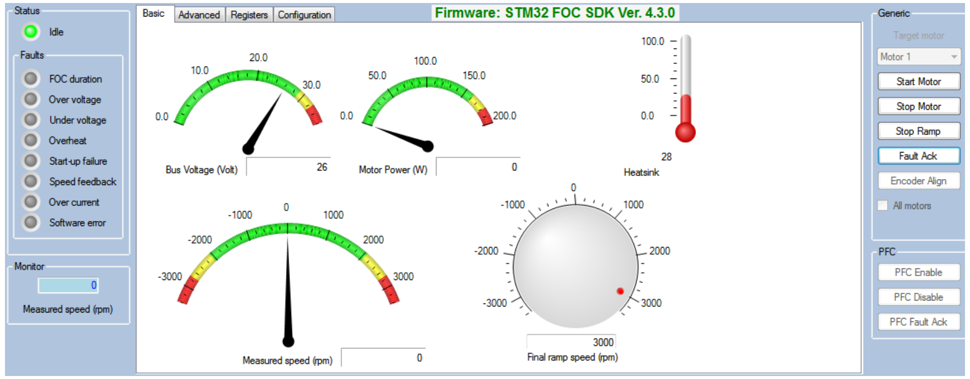
- Optionally, run Keil uVision.
- Open the Keil workspace (located in Project\MDK-ARM) folder according to the microcontroller family (e.g. STM32F10x_Workspace.uvmpw for STM32F1).
- Select the proper user project from the drop-down menu according to the control stage used (e.g. STM3210B-EVAL).
- Compile and download.



Motor Control – SDK – Workflow 4/4

- Finally the user can **send commands** (e.g. start, stop, execRamp, …) via serial communication.

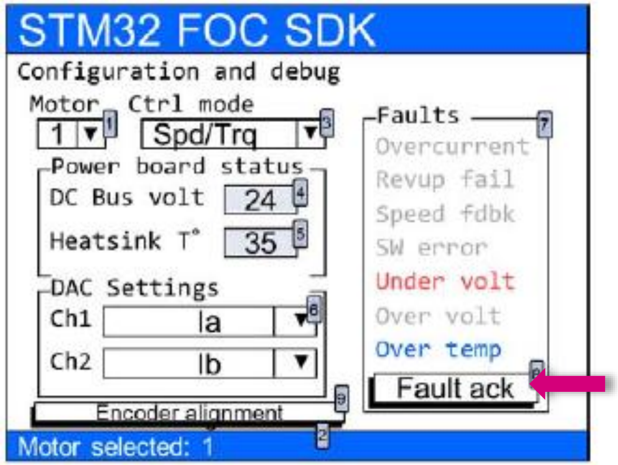
- Use the Workbench for debugging and real time communication.

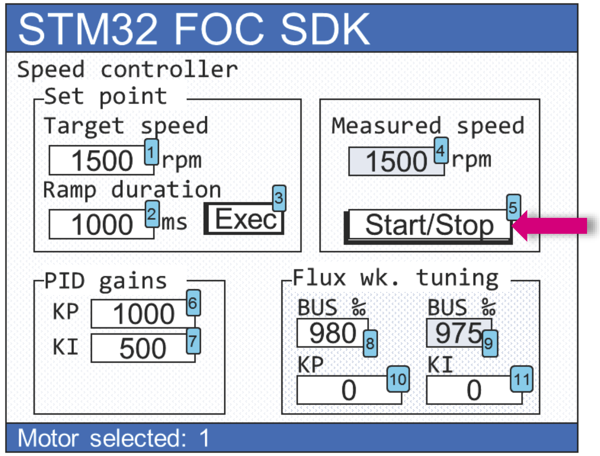


life.augmented

Run The Motor

- Arrange the system for running the motor:
  - Connect the control board with the power board using the MC cable.
  - Connect the motor to the power board.
  - Connect the power supply to the power board and turn on the bus.
- If the board is equipped with the LCD:
  - Press joystick center on *Fault Ack* button to reset the faults.
  - Press joystick right until the *Speed controller* page is reached.
  - The press joystick down to reach the *Start/Stop* button.
  - Press the center of the joystick to run the motor.

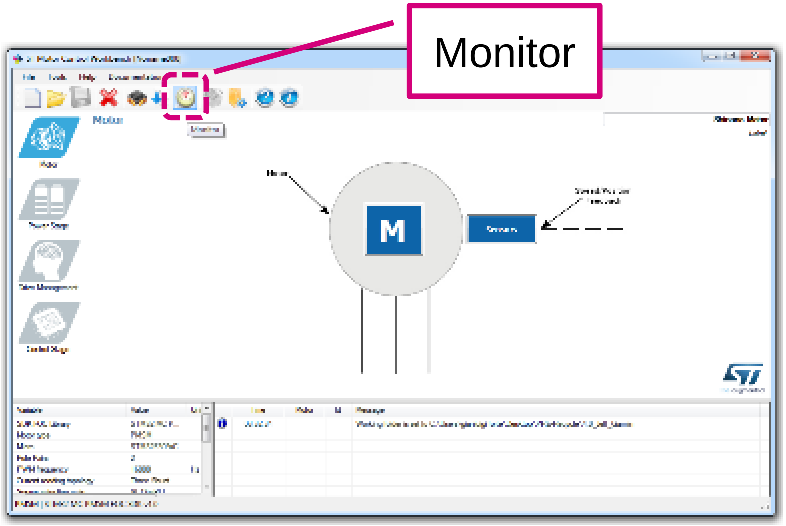


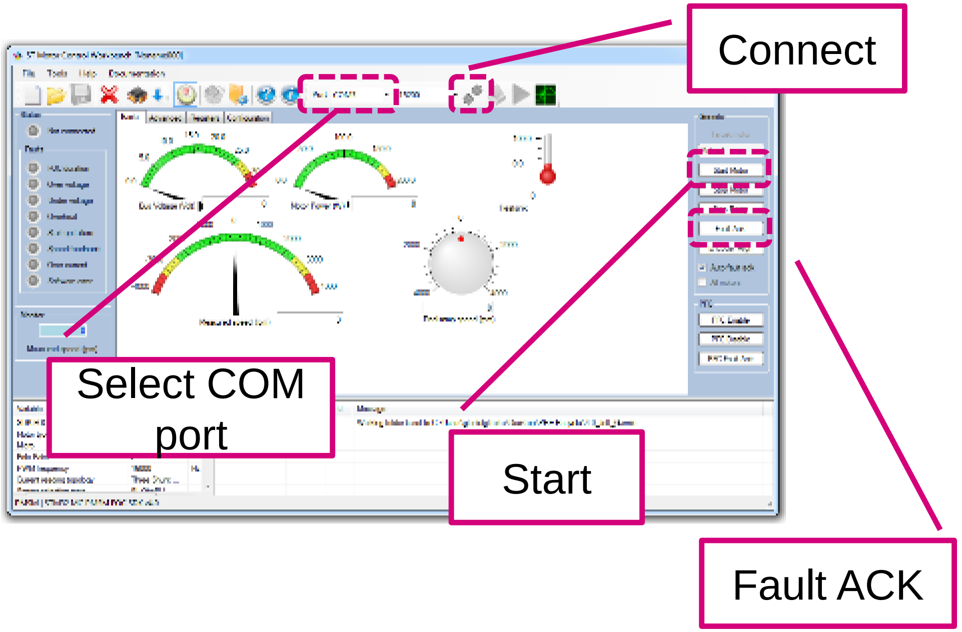


life.augmented

Run The Motor

- Optionally you can start the motor using the ST MC Workbench.
- Connect the PC to the control board with the USB to RS-232 dongle (and a null modem cable).
- Open the Workbench project used to configure the firmware and click on *Monitor* button.
- Select the *COM port* and click *Connect* button. This establish the communication with the firmware.
- To clear the fault, click *Fault Ack* and then *Start Motor* button to run the motor.



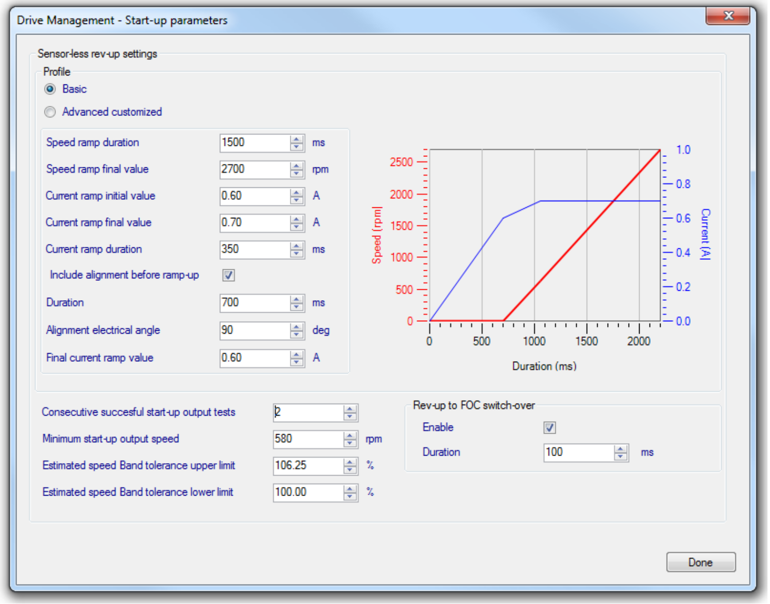


life.augmented

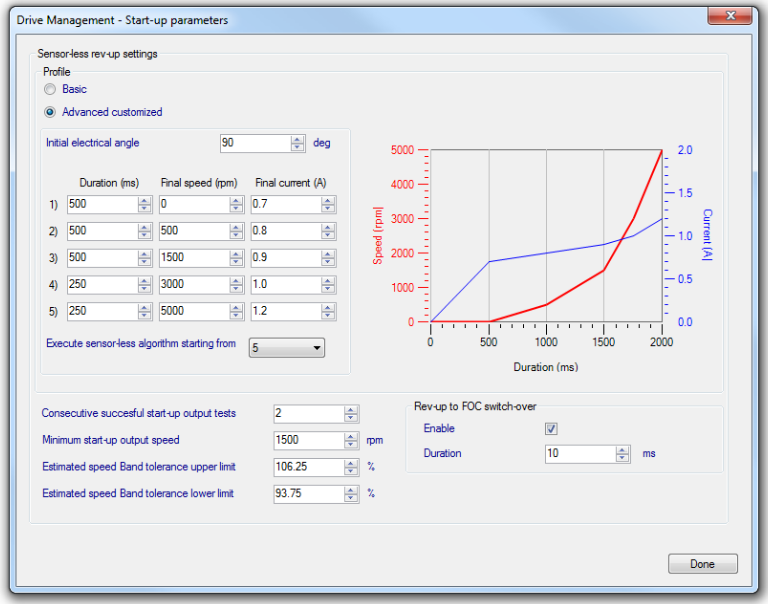
State Observer: Startup Procedure

- The sensorless algorithm is a bemf observer, so the motor should rotate to produce BEMF. That’s why a startup procedure is required. Startup needs to be tuned (depend on inertia, load..)
- Two options of settings: basic and advanced

Basic

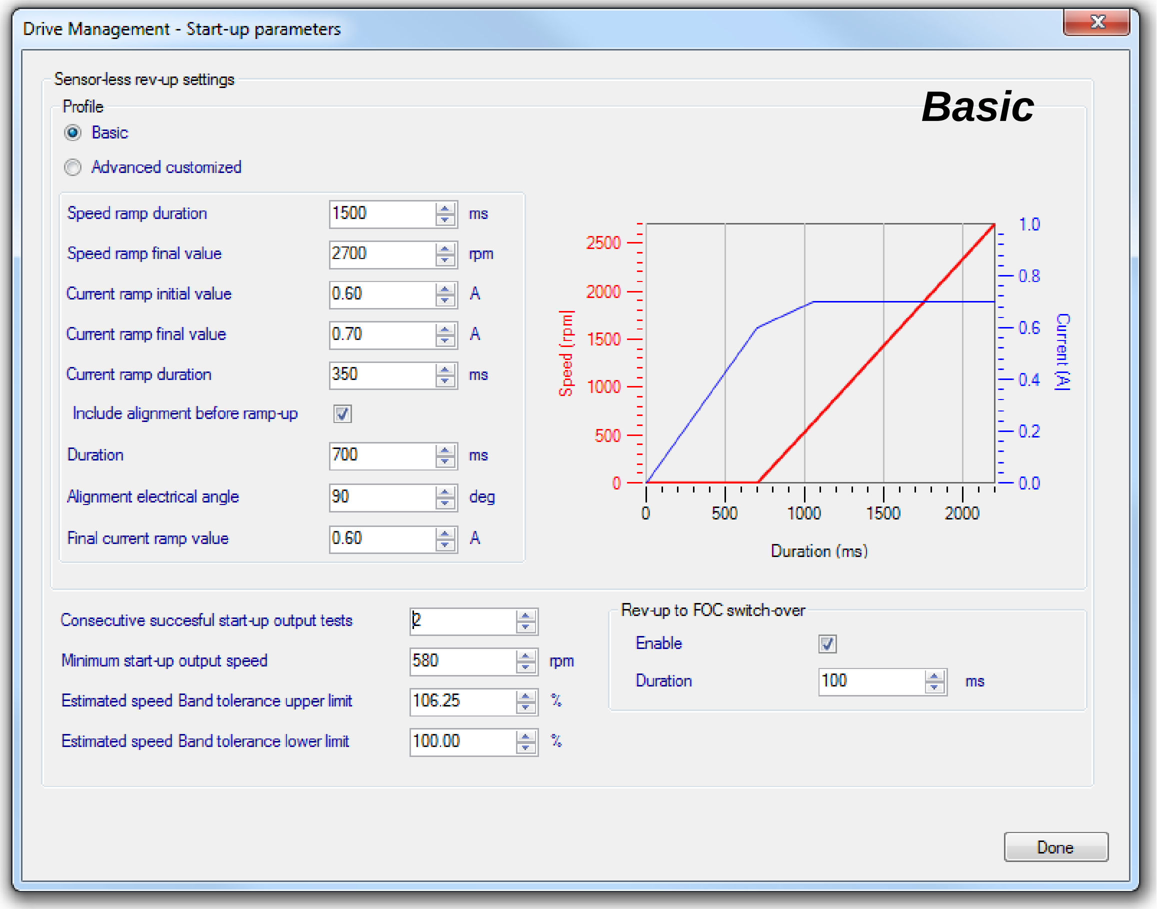


Advanced

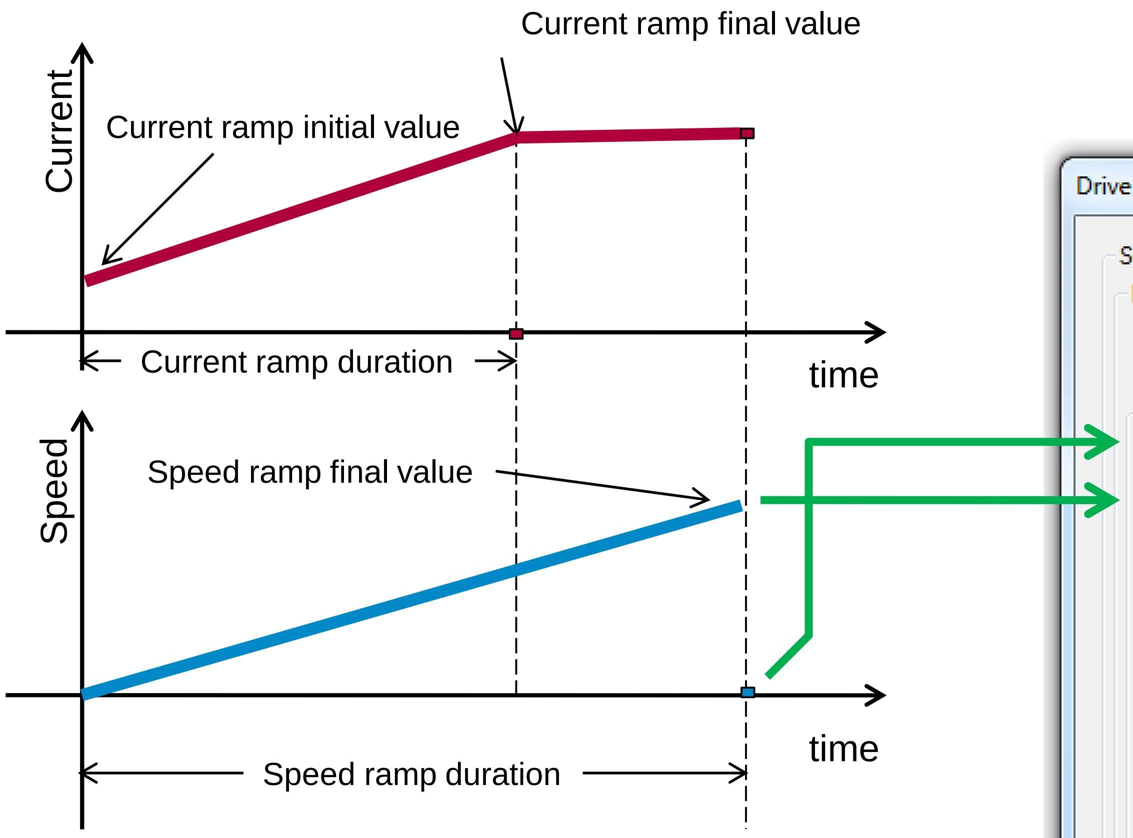


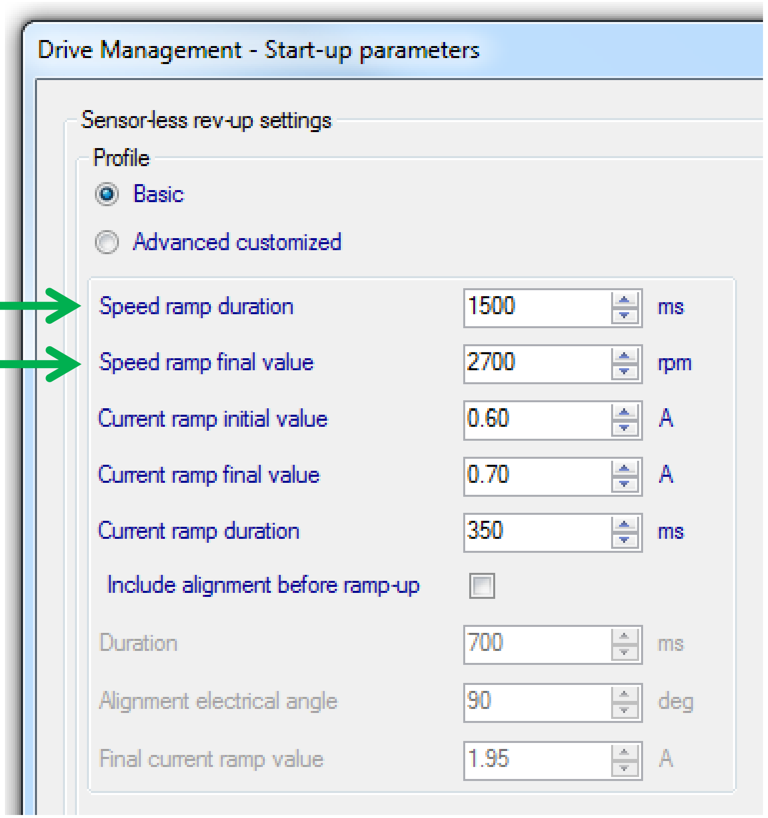
How To Customize The Sensor-Less Start-Up

- Set *current ramp initial and final values* equal to motor *nominal current* value / 2 (if load is low at low speed, otherwise it can be set up to 0.8-1.0 times *nominal current* value)
- Set *speed ramp final value* to around 30% of *maximum application speed*
- According to motor inertia it may be required to increase the *speed ramp duration*
- Set *minimum start-up output speed* to 15% of maximum application speed (if required, decreased it later)
- Set estimated speed band tolerance lower limit to 93.75%
- Enable the alignment at the beginning of your development (*duration* 2000ms, *final current ramp value* from 0.5 to 1 times motor nominal current according to load)



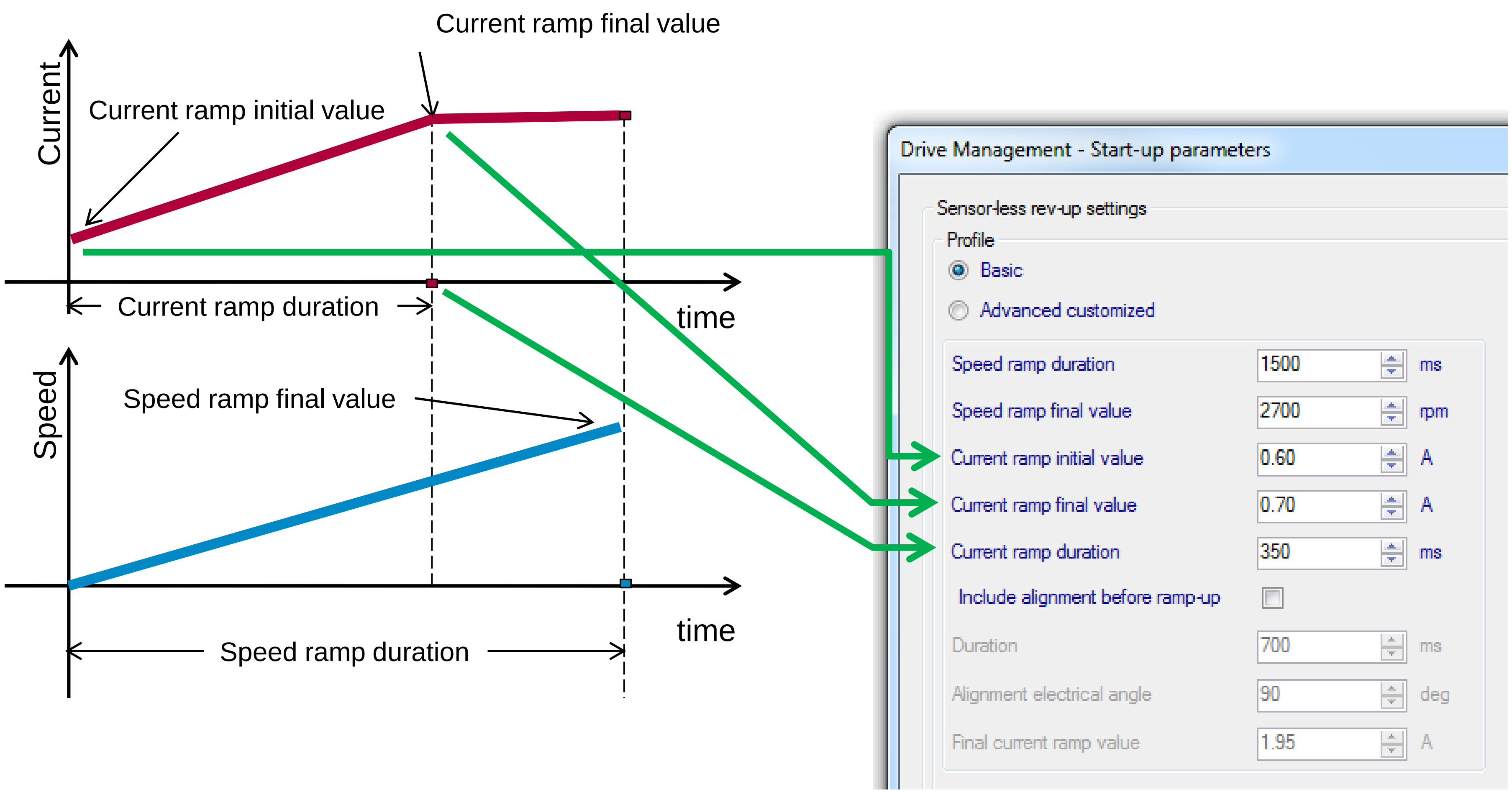
Startup Procedure: Basic, Acceleration





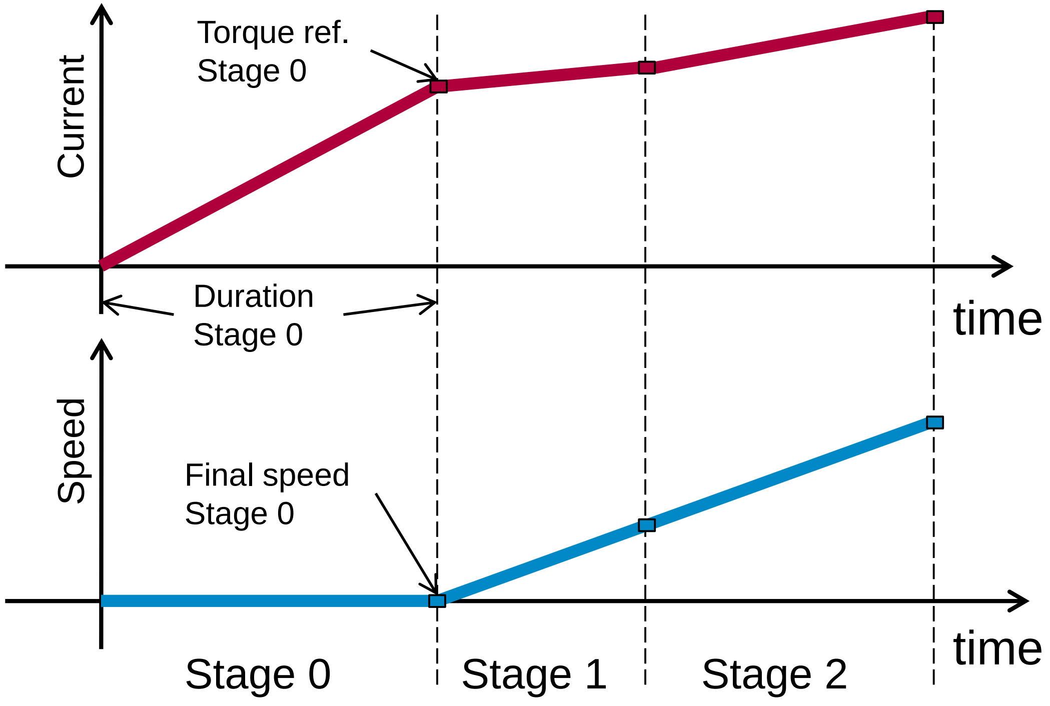
life.augmented

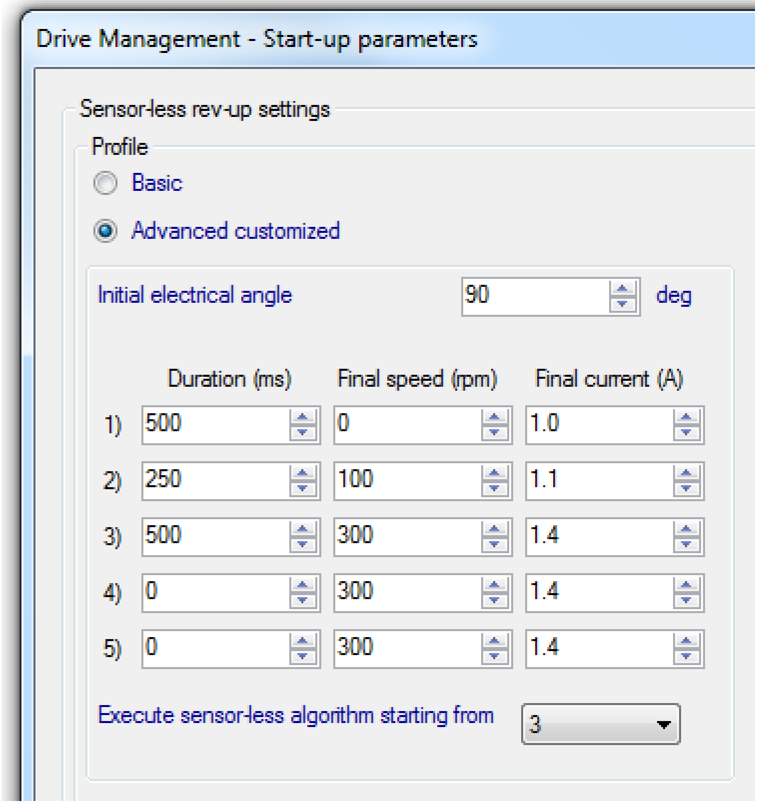
Startup Procedure: Basic, Current



Startup Procedure: Advanced

- The programmed rev-up sequence is composed by a number of stages; for each stage is possible to define the duration, the final torque reference and the final speed of the virtual sensor.
- It is possible to define the starting electrical angle.
- It is possible to set step variation in the current using duration zero.





life.augmented

Troubleshooting

- Problem: 'SW error' fault message appears and the motor do not even try to start
  - Source: the FOC execution rate is too high and computation can not be ended in time
  - Solution: In *Drive settings*, decrease ratio between *PWM frequency* and *Torque and flux regulator execution rate* (e.g. increasing Torque and flux regulator execution rate by one)
- Problem: 'Over-current' fault message appears and the motor do not even try to start
  - 1st possible source: wrong current sensing topology has been selected in *power stage → current sensing*
    - Solution: select right current sensing configuration
  - 2nd possible source: wrong current sensing parameters
    - Solution: check power stage parameters
  - 3rd possible source: current regulation loop bandwidth is too high for this HW
    - Solution: in *drive parameters → drive settings* decrease current regulation bandwidth (normally down to 2000 rad/sec for 3shunt topology and 1000 rad/s for single shunt topology)
    - Typical current regulation loop bandwidth max values are 4500 rad/sec for 1 shunt, 9000 rad/sec for 3-shunt

Troubleshooting

- Problem: Motor initially moves but then doesn't rev-up, then fault message 'Rev-up failure' appears
  - Source: typically this happens cause the current provided to the motor is not enough for making it accelerate so fast
    - 1st possible solution: decrease acceleration rate by increasing *Start-up parameters → speed ramp duration* (being *Start-up parameters → speed ramp final value* set to about 30% of maximum application speed)
    - 2nd possible solution: increase start-up current by increasing *current ramp initial* and *final* values up to *motor → nominal current*
    - Enabling 'Alignment phase' (at least at the beginning of the development) makes start-up more deterministic, use around 2000ms, half of nominal current as first settings

- Problem: The rotor moves and accelerate following the ramp-up profile but then it stops and the fault message 'Rev-up failure' appears (a mix of following problem sources can be occurring):
  - 1st possible source: Observer gain G2 is too high and this makes speed reconstruction a bit noisy (never recognized as reliable). A mix of following solutions could be required:
    - 1st possible solution: decrease observer gain G2 by successive steps: /2, /4, /6, /8
    - 2nd possible solution: Enlarge Drive parameters→ *Speed/position feedback management → variance threshold* so as to make rotor locked check less 'demanding'. (up to 80% for PLL and 400% for CORDIC)
  - 2nd possible source: the "window" where the reliability of the estimation is checked is too small
    - 1st possible solution: increase *speed ramp final* value to around 40% of maximum application speed
    - 2nd possible solution: decrease *minimum start-up output speed* to 10% of maximum application speed

life.augmented

Troubleshooting

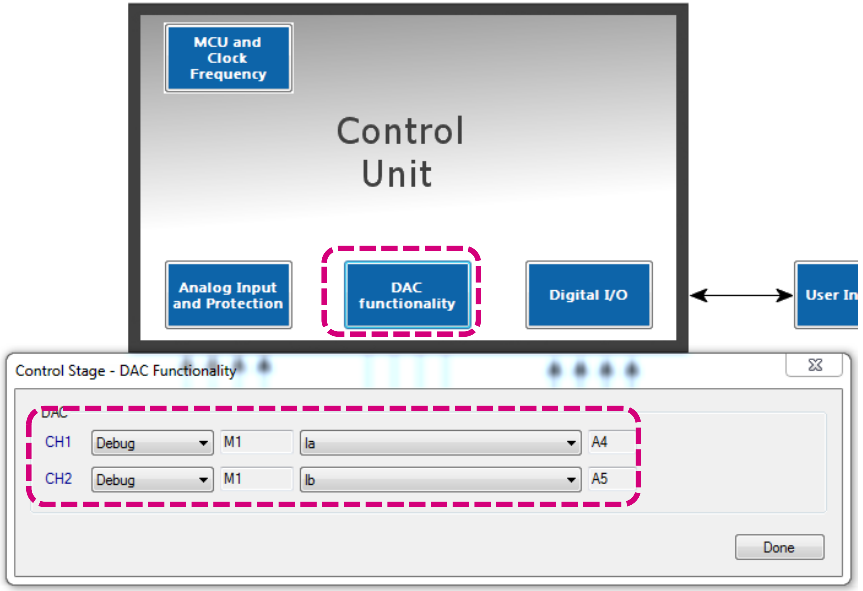
- Problem: The rotor moves and accelerate following the ramp-up profile but then it stops and the fault message ‘Speed feedback’ appears
  - Use speed ramps: having a target speed gently going from the start-up output speed to the final target will avoid abrupt variations of torque demand that could spoil B-emf estimation
  - A mix of following problem sources can be occurring:
  - 1st possible source: Observer gain G2 is too high and this makes speed reconstruction a bit noisy (for the selected speed PI gains). A mix of following solutions could be required:
    - 1st possible solution: decrease observer gain G2 by successive steps: /2, /4, /6, /8
    - 2nd possible solution: Run motor in torque mode, if trouble doesn’t exist in torque mode, it means speed regulator gains are not optimal try changing them
  - 2nd possible source: frequent situation when the start-up has been validated too early
    - Solution: Try increasing *Start-up parameters* → *consecutive successful start-up output test* (normally to not more than 4-5) being *minimum start-up output speed* set to 15% of maximum application speed (if required, decreased it later)

- Problem: motor runs but current are not sinusoidal at all
  - 1st possible source: speed PI gains are not good
    - Solution: decrease Kp gain (and act on Ki evaluating speed regulation over/under shooting during transients)

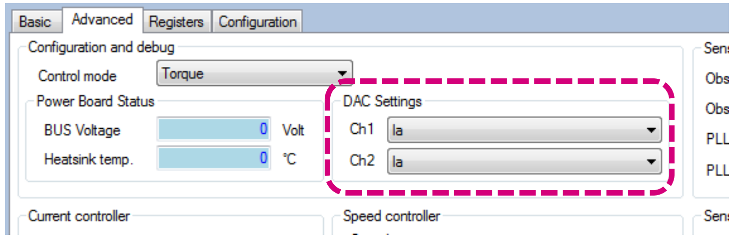
Use DAC Channels

- DAC functionality can help to debug and tune the application.

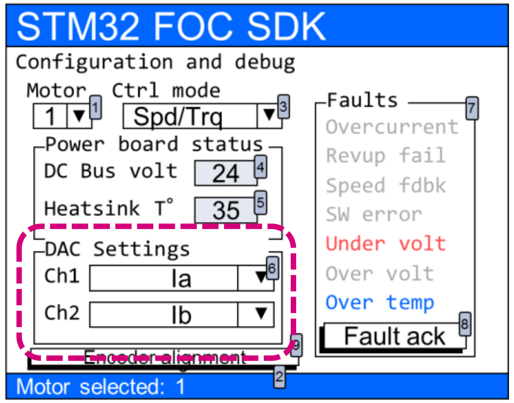
Enabling



Selection with WB



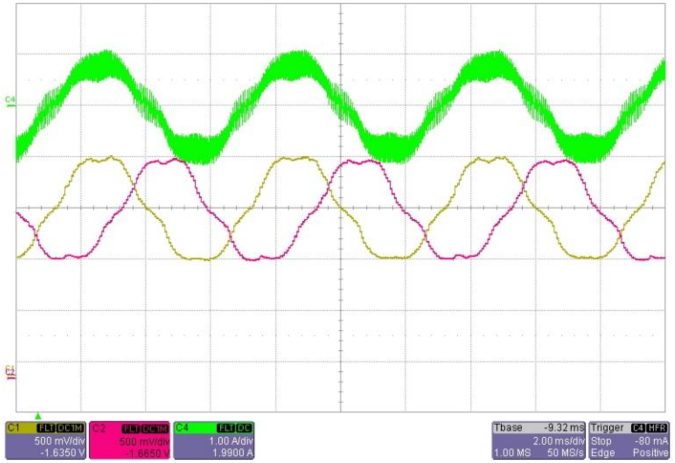
Selection with LCD



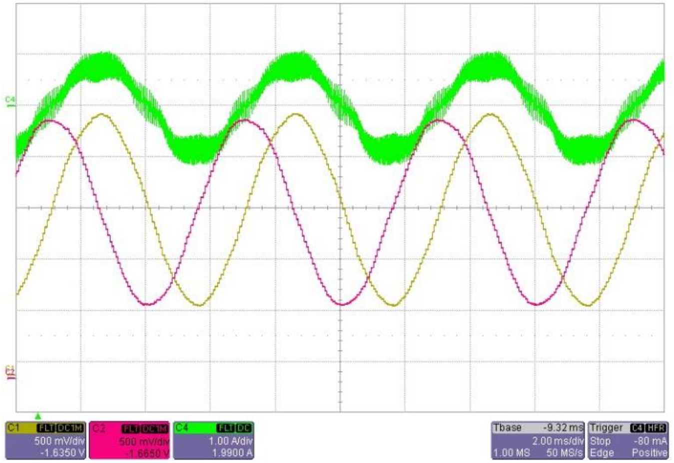
life.augmented

Use DAC Channels

- Typical DAC waveforms of tuned system



- Green: phase A motor current
- Yellow: DAC ch1 (Ia)
- Pink: DAC ch2 (Ib)



- Green: phase A motor current
- Yellow: DAC ch1 (Obs. BEMF Alpha)
- Pink: DAC ch2 (Obs. BEMF Beta)

life.augmented

Thank You!

life.augmented