STM32 PMSM SDK 5.2 training

T.O.M.A.S. team

ST MC
Timebase & Interrupt structure

NVIC Configuration 1/2

STM32 NVIC (Nested Vectored Interrupt Controller) priority group configuration is NVIC_PriorityGroup_3

Priorities used in MC Library:

STM32 Others (Cortex M3, M4, M7)

IRQ	Pre-emption priority
TIM1 UPDATE	0
TIM8 UPDATE	0
DMA	0
RESERVED	1
ADC	2
USART (UI LIB)	3
TIMx (General, for speed sensor decoding, Hall, Encoder etc.)	3
TIM1 BRK	4
SYSTICK	4

STM32F0xxx (Cortex M0+)

IRQ	Pre-emption priority
TIM1 UPDATE	0
DMA	0
ADC	1
TIMx (General, for speed sensor decoding, Hall, Encoder etc.)	2
SYSTICK	2
USART (UI Lib)	3

life.augmented

NVIC Configuration 2/2

STM32 Others (Cortex M3, M4, M7)

COMPONENT	Pre-emption priority
MC LIBRARY	0,1,2,3,4
TIMEBASE (SYSTICK)	4
USER	5,6,7

STM32F0xxx (Cortex M0+)

COMPONENT	Pre-emption priority
MC LIBRARY	0,1,2,3
TIMEBASE (SYSTICK)	2
USER	3

Timebase / clocks

A Timebase is needed to clock the MC Application

The demo Timebase.c can be considered an example or used as it is

SysTick timer (main .c)

```c
void SysTick_Configuration(void)
{
  /* Setup SysTick Timer for 500 usec interrupts  */
  if (SysTick_Config((SystemCoreClock) / SYS_TICK_FREQUENCY))
  {
    /* Capture error */
    while (1);
  }

  NVIC_SetPriority(SysTick_IRQn, SYSTICK_PRIORITY);
  NVIC_SetPriority(PendSV_IRQn, PENDSV_PRIORITY);
}
```

SysTick Handler (stm32fxxx_it.c)

```c
  * @brief  This function handles SysTick Handler.
  * @param  None
  * @retval None
  */
void SysTick_Handler(void)
{
  TB_Scheduler();
}
```

STM32F0/F1/F2/F3/F4	Task name	Main role	Frequency (Period)
	Safety task	Over voltage management Switch off PWM if fault occurs User ADC conversions	2kHz (500µs)
	Medium frequency task	Execute the speed regulation loop State machine management	Speed Regulator Execution Rate (typical: 1 KHz) period must be a multiple of 500 µs

MC Application and user - Tasks

- Safety task, medium frequency task, UI task and User task use Systick as time base.

Scope	Name	Main role	Priority	Frequency (Period)
MC tasks	**High frequency task**	Execute the current regulation loop	IRQ	$\frac{PWM\ frequency}{FOC\ rate}$
	Safety task	Over voltage management Switch off PWM if fault occurs User ADC conversions	SysTick	2kHz (500µs)
	Medium frequency task	Execute the speed regulation loop State machine management	SysTick	Speed Regulator Execution Rate (typical: 1 KHz) period must be a multiple of 500 µs
User tasks	**User task code**	User code	Main	Deepens on User code

* Default value, can be changed in the WB in Drive Management – Drive Settings – Speed regulator – Execution rate

MC FW library features list

available in Motor Control SDK v5.x (X-CUBE-MCSDK)

STM32 series	F0	F1	F3	F4	F7 (v5.3)	L4 (V5.3)
• 1 Shunt	✓	✓	✓	✓	✓	✓
• 3 Shunt	✓	✓	✓	✓	✓	✓
• Hall sensors	✓	✓	✓	✓	✓	✓
• Encoder	✓	✓	✓	✓	✓	✓
• ICS	✗	✓	✓	✓	✓	✓
• Flux weakening	✓	✓	✓	✓	✓	✓
• MTPA	✓	✓	✓	✓	✓	✓
• Startup On the fly	✓	✓	✓	✓	✓	✓
• Sensorless (PLL / Cordic)	✓	✓	✓	✓	✓	✓
• Feed Forward	✓	✓	✓	✓	✓	✓
• Single FOC (max. FOC freq.*)	12 kHz	11/23 kHz	30 kHz	50 kHz	✓	✓
• Dual FOC (max. FOC freq.*)	✗	20 kHz / ✗	23 kHz	45 kHz	✓ / ✗	✗

* FOC frequency tested in v4.3

life.augmented

MC SDK v5.x – Perf. Measurements

PWM set at 20KHz / 10KHz FOC

			MCSDK4.3				MCSDK5.1.1					
MCU	Nb Motor	Config	CPU Workload (%)	Total Code size (Kb)	MC Lib (Kb)	STD Lib (Kb)	CPU Workload (%)	Grand Total Code Size (+ HAL) (Kb)	Grand Total Code Size (+ LL) (Kb)	MC Lib (Kb)	HAL (Kb)	LL (Kb)
F072RB	Single	1x Shunt	52.0	19.3	17.3	2.0	46.4	18.0	16.9	13.1	5.2	3.2
F072RB	Single	3x Shunt	49.0	19.6	17.7	2.0	42.6	17.1	16.3	12.5	4.6	3.2
F303RE	Single	1x Shunt	20.0	21.2	18.2	3.0	20.4	22.4	19.9	14.9	8.1	4.4
F303RE	Single	3x Shunt	18.5	23.0	20.6	2.4	17.8	23.4	19.3	16.1	7.7	2.6
F446RE	Single	1x Shunt	10.5	20.1	17.7	2.4	10.2	20.1	19.0	14.7	5.5	3.3
F446RE	Single	3x Shunt	8.9	17.8	15.8	2.0	8.2	18.2	15.7	13.2	4.8	2.0
F303VE	DUAL	3x Shunt	38.9	25.2	17.4	2.4	38.2	25.5	21.8	18.6	7.9	2.6
F415ZG	DUAL	3x Shunt	23.1	19.9	17.9	2.0	18.3	19.6	17.7	15.2	4.8	2.0

life.augmented

Braking & overvoltage handling

Brake strategies

- Unless you are using a power system with regenerative capabilities, your inverter bulk capacitors will be charged if:
  - The six inverter switches are opened and the motor is running at a speed higher than the nominal one
  - The control tries to brake

- Different ways can be utilized to dissipate the motor kinetic energy, in particular:
  - Shorting motor windings
  - Brake resisitor
  - HFI – active control at zero speed
  - DC vector

Shorting motor windings

- Not supported by the library, it could be implemented with STM32 (TIM1 configured so that low side switches are turned on when the MOE bit is reset)

- Shorting the motor winding, the motor current path does not go through the bulk capacitor and flow only inside the motor phases

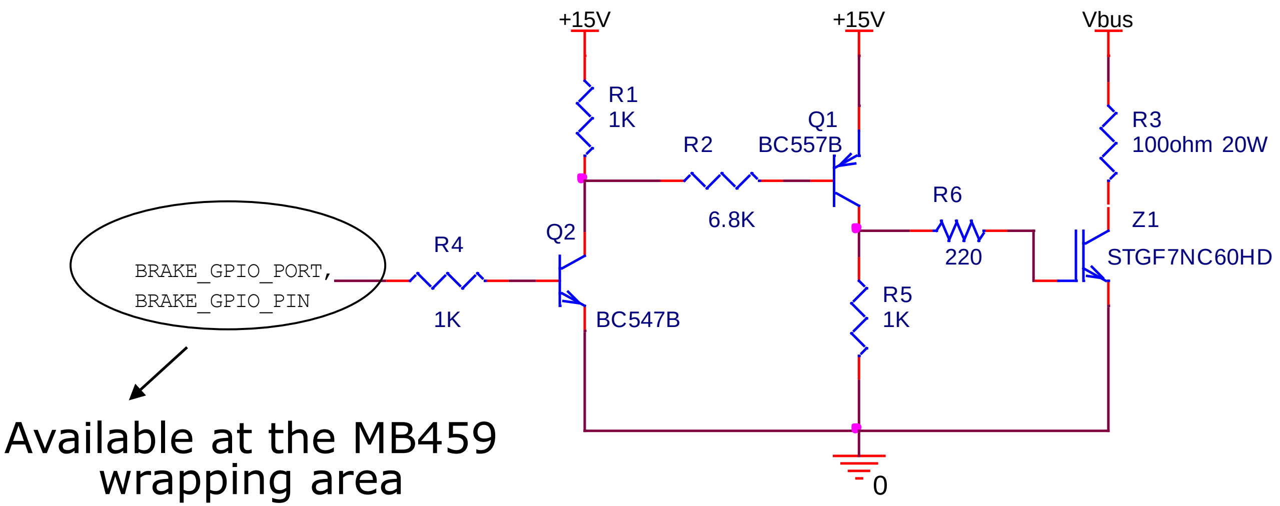
- Steady state ISC is limited by motor inductance:

$$I_{SC} = \left| \frac{K_e \cdot \omega_r}{R_s + j\omega_r L_S} \right| = \frac{K_e}{\sqrt{\frac{R_S^2}{\omega_r^2} + L_S^2}}$$

- Warning: during transient a higher current can flow. L6386 overcurrent must be properly sized to avoid forcing the low side switches turn-off from gate drivers

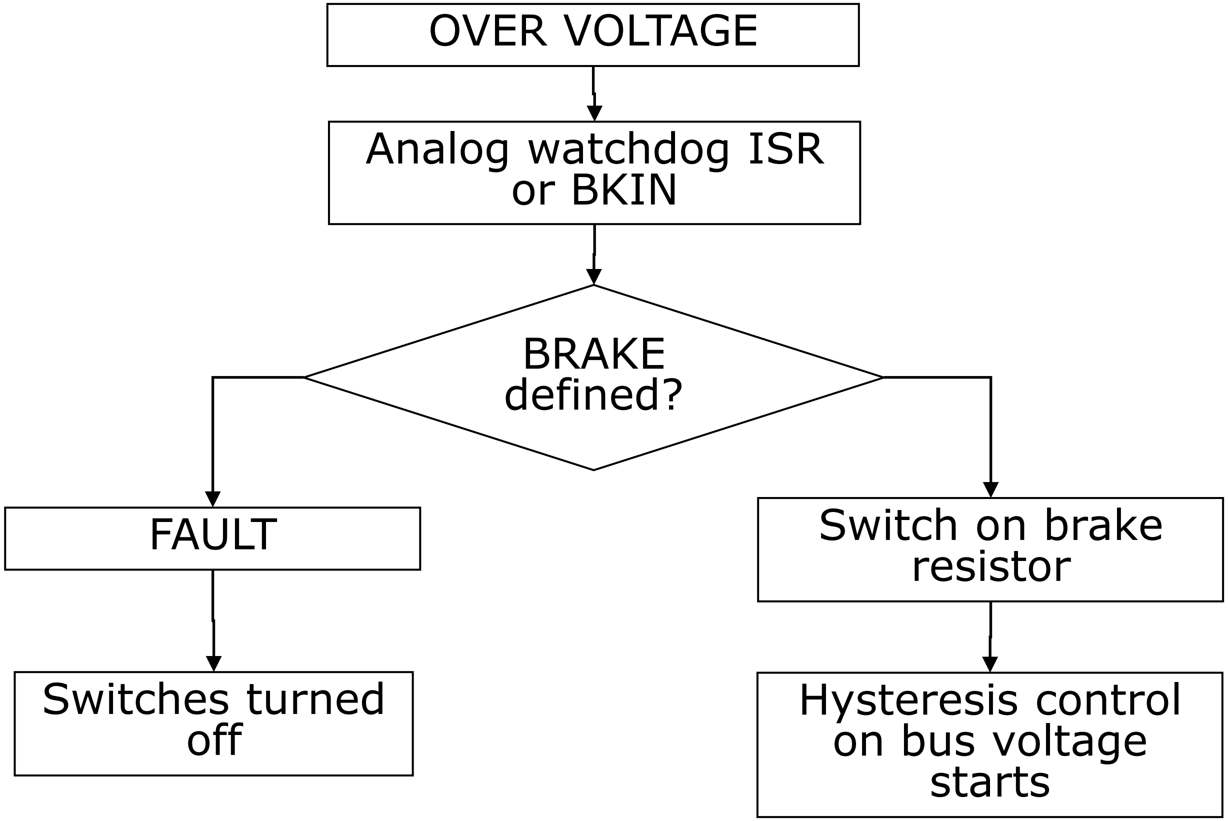
Brake resistor

- Acts by sinking current from DC bus
- Additional hardware to be soldered when using MB459



MANDATORY TO BE USED IN FIELD WEAKENING OPERATIONS AT HIGH VOLTAGE

Over voltage management



No over voltage FAULT is generated when BRAKE is defined

FAULT definitions

- **Over-voltage**, acts above OV_VOLTAGE_THRESHOLD_V (drive_parameters.h) only if BRAKE is not defined
- **Under-voltage**, acts below UD_VOLTAGE_THRESHOLD_V (drive_parameters.h)
- **Over-temperature**, acts below OV_TEMPERATURE_THRESHOLD_C, hysteresis is specified by OV_TEMPERATURE_HYSTERESIS_C (drive_parameters.h)
- **Start-up failure**, used only in sensorless and HFI, acts when speed ramp is finished and observer did not pass the start-up validity tests
- **Speed feedback**, occurs when too many errors have been observed in speed measurement (variance threshold, b-emf size, Hall time-out or invalid input) – can be caused by noise in current measurement and b-emf reconstruction, speed PID oscillations, too high cut-off frequency
- **Over-current**, detected by BKIN, BKIN2 input

Advanced function

Open Loop mode

- Activated by defining the macro OPEN_LOOP_FOC in “drive_params.h” together with other parameters
- In Open Loop mode the library generates rotating field, useful i.e. for ACIM motor testing
- Beware of initial voltage, the current at the beginning may be very high: Vol/(2 * Rs)

Part of file “Drive parameters.h”

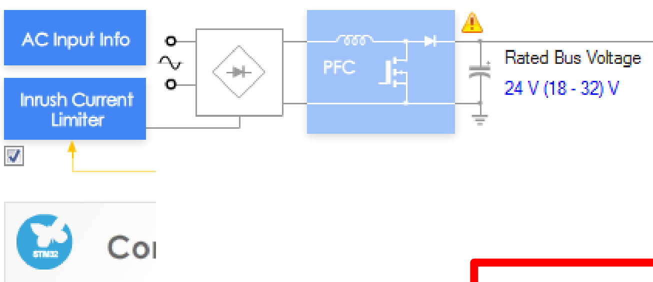
```
/******************************   ADDITIONAL FEATURES   *********************/
#define  BUS_VOLTAGE_READING                     ENABLE

#define  TEMPERATURE_READING                     DISABLE

#define  OPEN_LOOP_FOC                           ENABLE      /*!< ENABLE for open loop */
#define  OPEN_LOOP_VOLTAGE_d                     8000        / *!< Three Phase voltage amplitude in s16 format */
#define  OPEN_LOOP_SPEED_RPM                     100         /*!< Final forced speed in rpm */
#define  OPEN_LOOP_SPEED_RAMP_DURATION_MS        1000        /*!< 0-to-Final speed ramp duration  */
#define  OPEN_LOOP_VF                            ENABLE      / *!< TRUE to enable V/F mode */
#define  OPEN_LOOP_K                             44          /*! Slope of V/F curve expressed in s16 Voltage
                                                               for each 0.1Hz of ecchanical frequency increment. */

#define  OPEN_LOOP_OFF                           4400        /*! Offset of V/F curve expressed in s16 Voltage
                                                               applied when frequency is zero. */
```

life.augmented

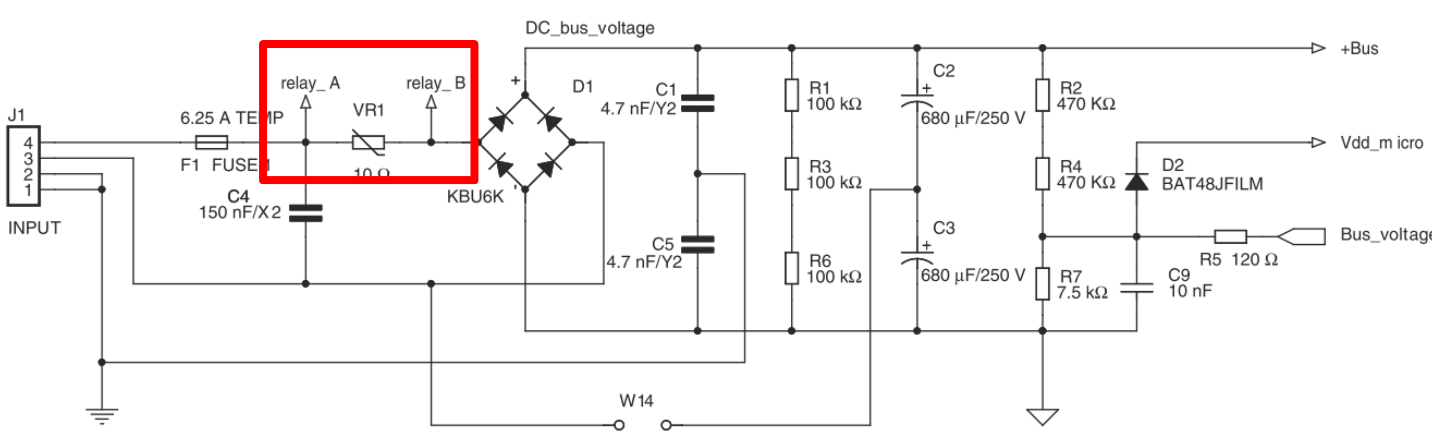
Inrush current limiter

- Blocks power stage overcurrent when charging bulk capacitors
- Defined in MCWB, Power stage
- Activated at start-up, deactivated and re-activated on voltage thresholds from Vbus measurement

Power Board: *STEVAL-IHM023V3*



Input part with bridge



Motor Startup “on-the-fly”

When? & How?

- Allows start of FOC in sensor-less while motor already turns (e.g. air-con fans, pumps)

- Implies zero amplitude on Iq, Id and waits for observer lock.

- If such start-up fails,
  then normal alignment and start-up are required

Motor Startup "on-the-fly"

Why is it needed in some applications?

- When freewheeling, the outdoor unit fan is exposed to wind, thus it can rotate – even very fastly – in both directions.

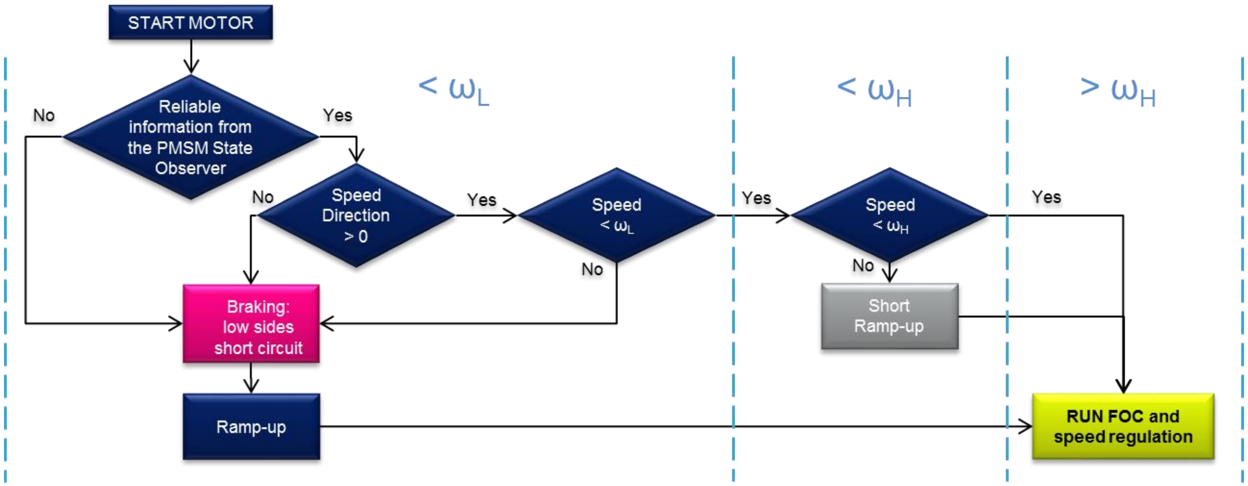
- Should the direction be positive (the one that is applied to maximize heat exchange) or negative, in any case the motor drive
  - – in sensorless mode (open-loop startup)
  - – can't be started efficiently:

- In fact, if the FOC starts with the wrong angle and torque reference, the situations that could occur are:
  - FOC is not able to control currents: Overcurrent. Stress on power devices
  - stator field not synchronous with rotor magnets field: Lengthy strong vibrations. Acoustic noise, stress on mechanics

Motor Startup “on-the-fly”

The New Algorithm

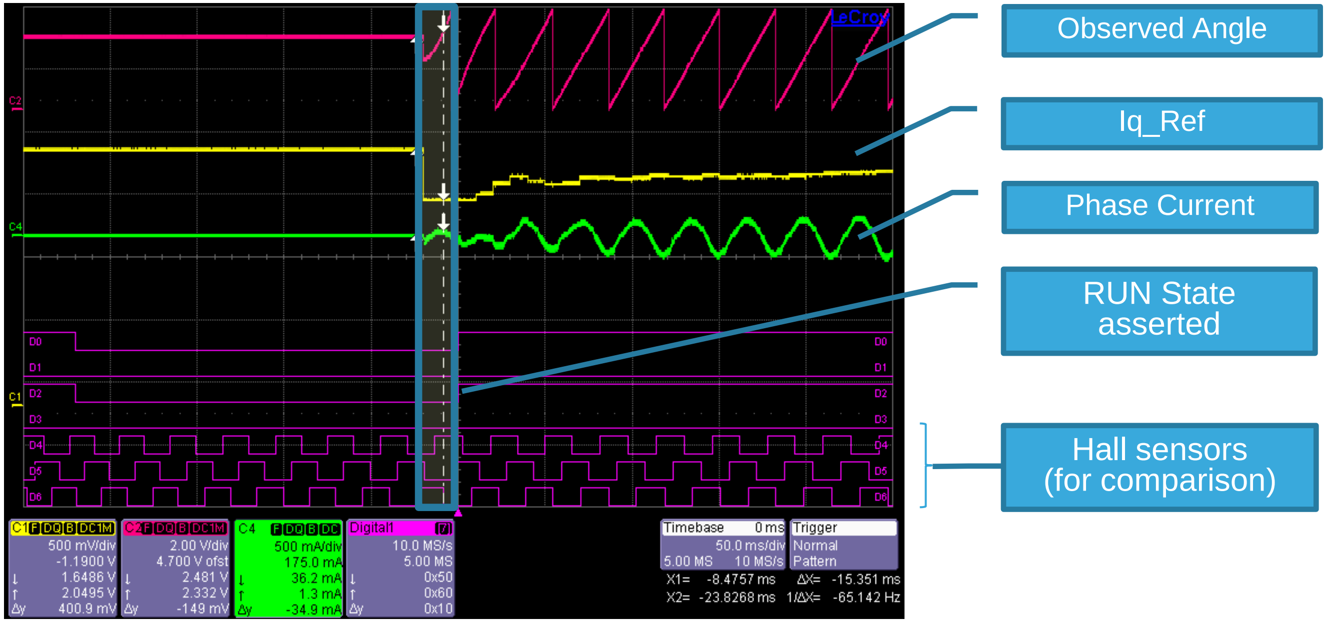
- ST’s “on-the-fly” startup algorithm allows a smooth drive insertion in direct and reverse speed.
- A null 3phase current is forced: according to rotor speed, 3 different behaviors are implemented:
  - Negative or very low ($<\omega_L$) positive speed: → braking with low sides
  - $\omega_{L,} <$ positive speed $< \omega_H$: → profiled acceleration before going in RUN mode
  - positive speed $> \omega_H$: → straight in RUN mode



Motor Startup "on-the-fly"

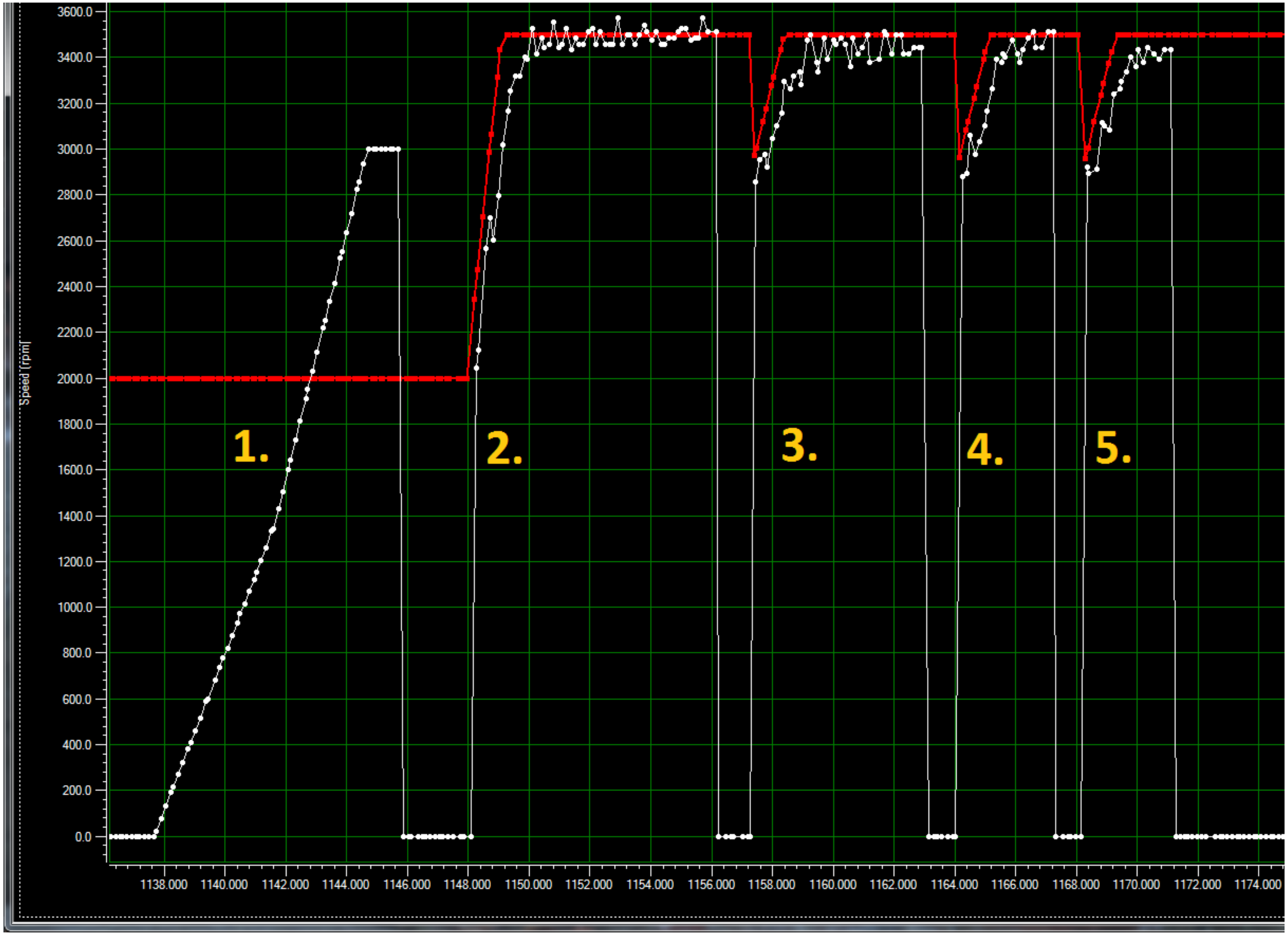
Live Capture: rotor speed > ωH

- **drive locks and runs in about 15 ms**
- **no perturbation on rotor speed**
- **Minimal perturbation on phase currents**



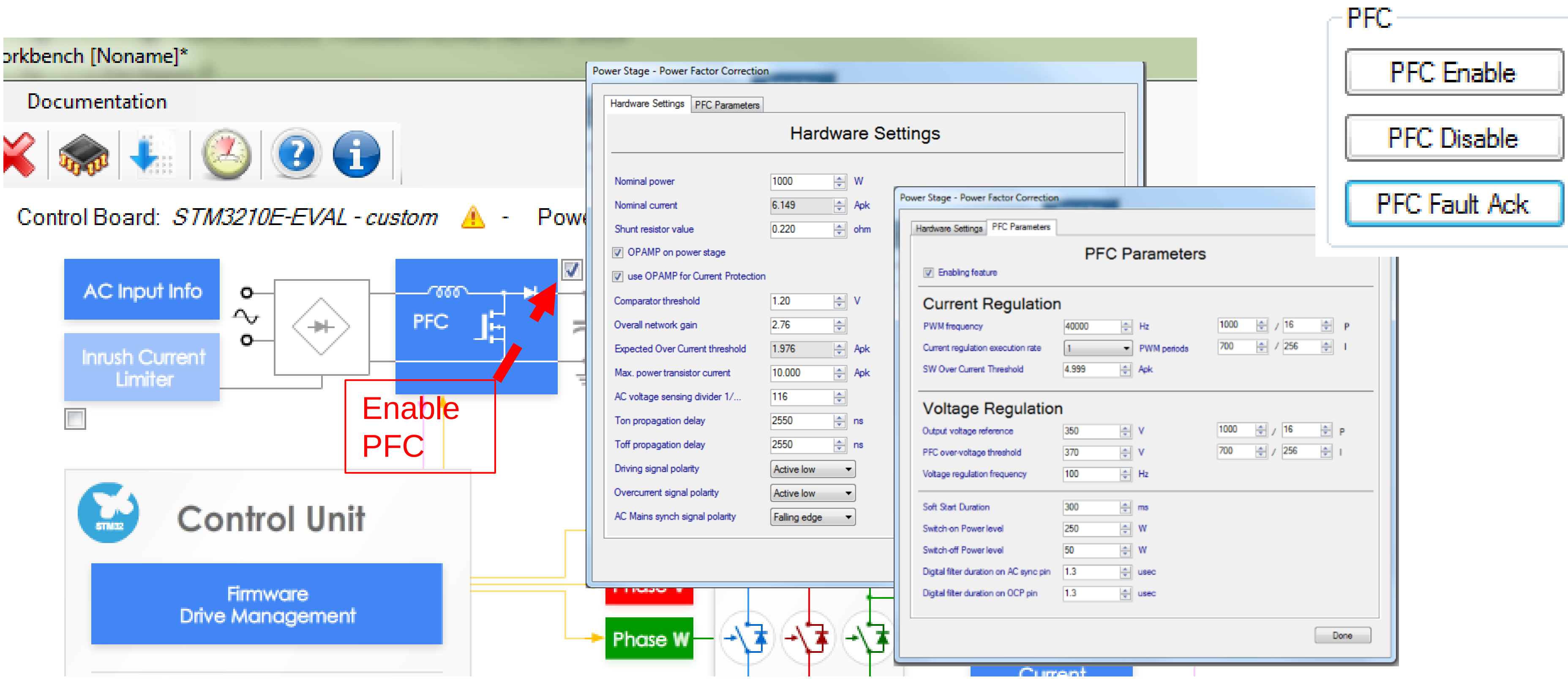
life.augmented

Motor Startup "on-the-fly" motor with big inertia



Digital PFC

- Digital PFC is embedded from the version v4.3
- Digital PFC FW is available (v5.2) for the
  - STM32F103 line
    (STM32F103xC, STM32F103xD, STM32F103xE)



STM32 FOC SDK 5.2 - user experience

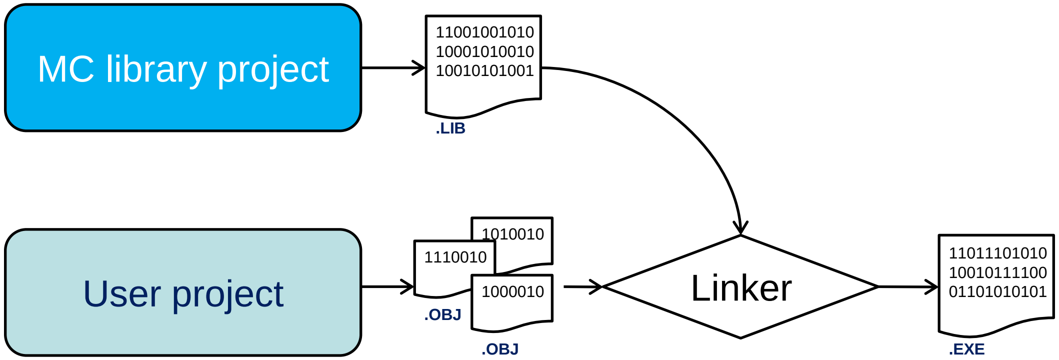
life.augmented

STM32 FOC SDK 5.2 - user experience

- HALL/LL Based & STM32CubeMX compatible
- Ready for most common IDEs (IAR, Keil, TrueSTUDIO)
- Software examples
- MC Application (State machine,Tasks)
- Fast Unidirectional/Bidirectional communication
- Workbench features (Startup sensorless, examples, doc link, PFC)
- Motor profiler / Self tuning algorithm
- HFI sensor-less algorithm

Workspaces and projects

- Each motor control workspace is composed of two projects:
  - the MC Library project and
  - the User project
- Both are required to build the executable
- In the WEB distribution the MC Library is provided already compiled for each configuration and for each supported IDE



Supported families

STM32F0xxFamily

NUCLEO-F030R8
NUCLEO-F072RB
STM32072B-EVAL

STM32F1xx Family :

NUCLEO-F103RB
STM3210E-Eval

STM32F3xx Family:

NUCLEO-F302R8
NUCLEO-F303RE
STM32303E-EVAL

STM32F4xx Family:

NUCLEO-F446RE
NUCLEO-F401RE
STM3240G-EVAL
STM3241G-EVAL
STM32446E-EVAL
STEVAL-IHM039V1

STM32F7xx Family:

NUCLEO-F746ZG
STM32F769I-EVAL

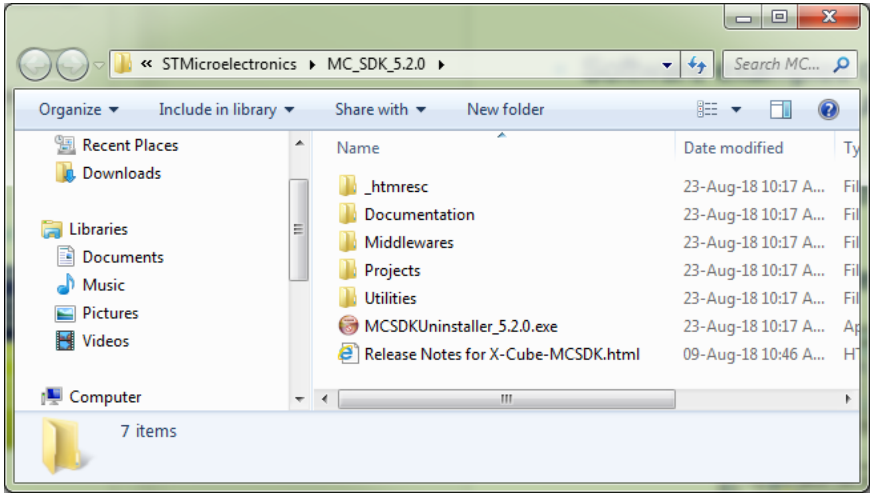
STM32L4xx Family:

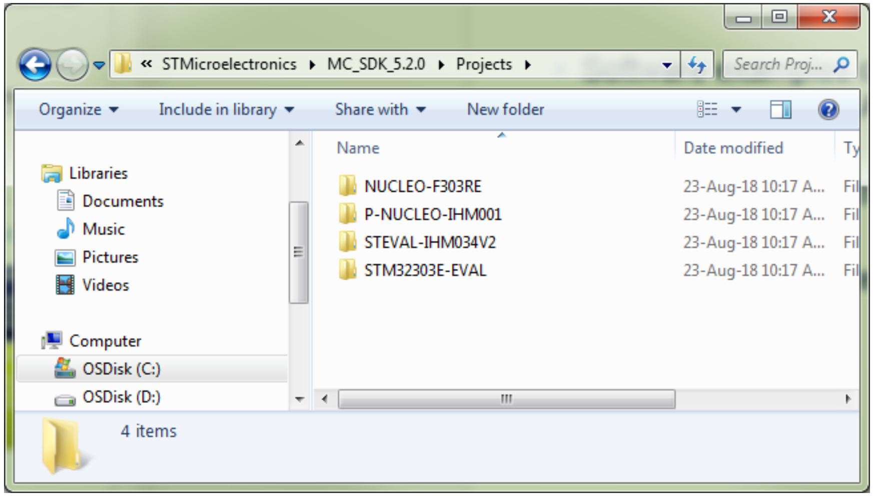
NUCLEO-L452RE
NUCLEO-L476RG
STM32L476G-EVAL

Software examples

- Software examples can be used as staring point for new design or guideline to understand the MC API.

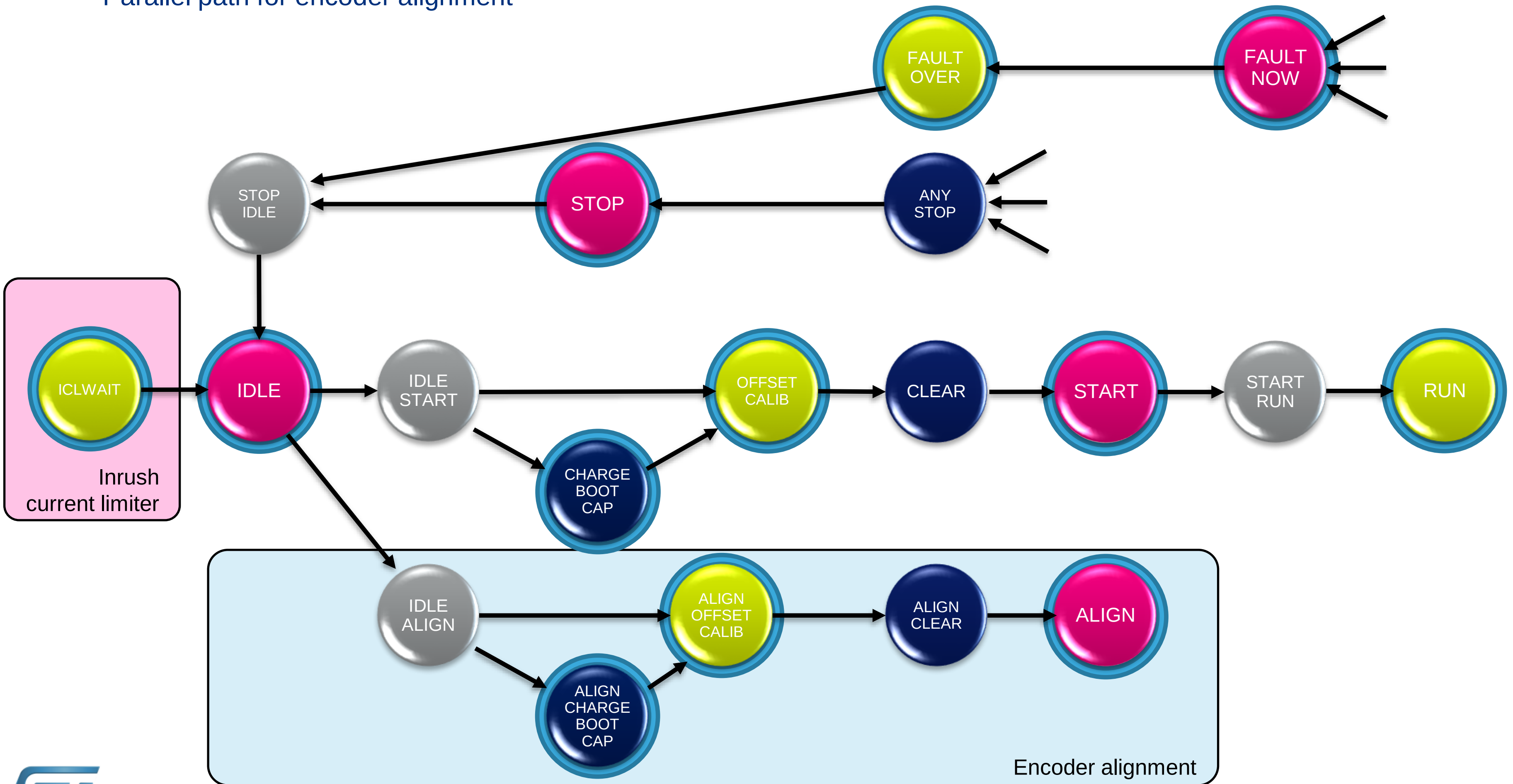
- You can found in installed path MC_SDK_5.x.x/Projects





MC Application – State machine

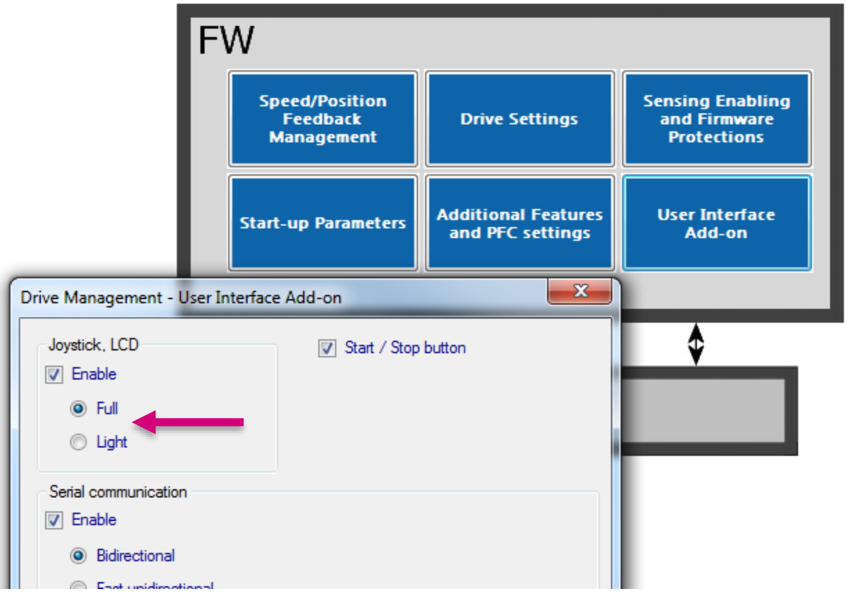
- State machine has been enriched with new states
  - Charge boot cap , Offset calibration, Clear, ICL Wait
  - Parallel path for encoder alignment



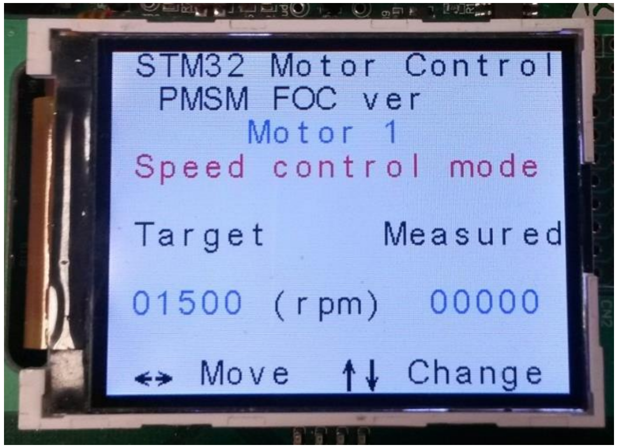
Is not supported by v5.2

Light/Full LCD

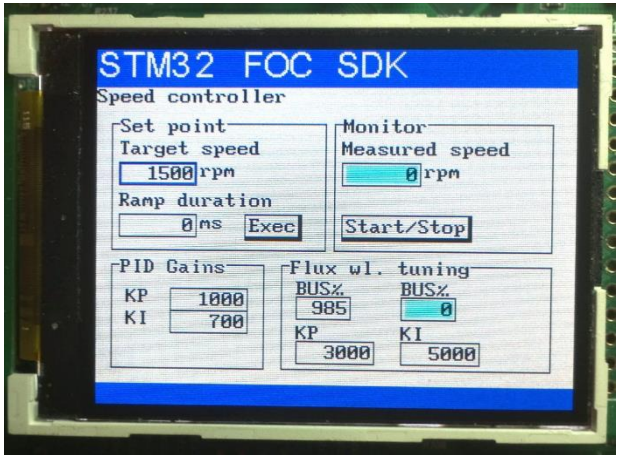
- Light LCD allows to customize the GUI itself adding extra interactions with the firmware



Light LCD



Full LCD

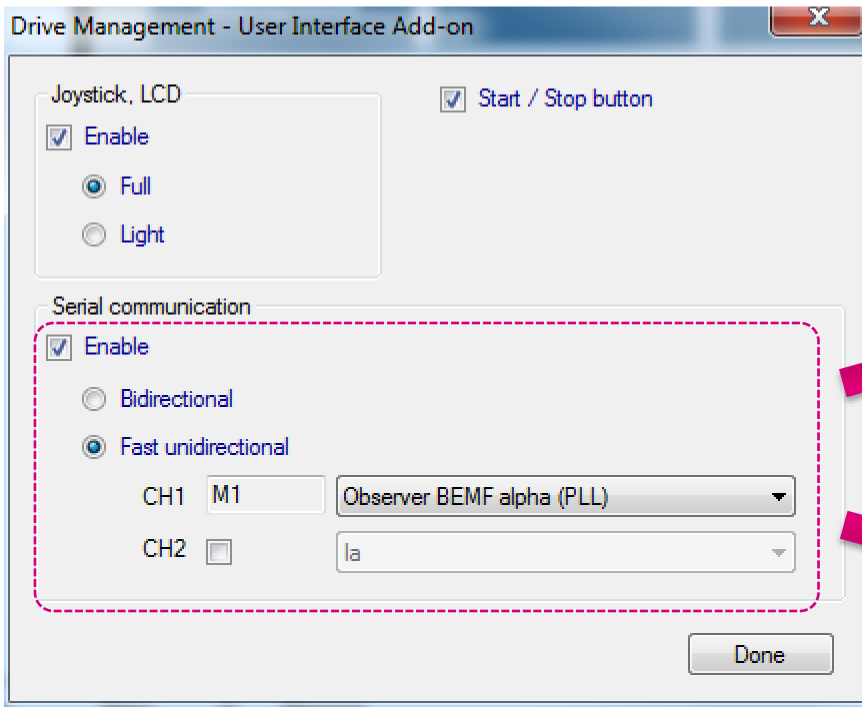


	Type	Customizable	Feature
Full LCD	External project to be flashed separately	Hard	Full
Light LCD	Inside MC project	Easy	Partial

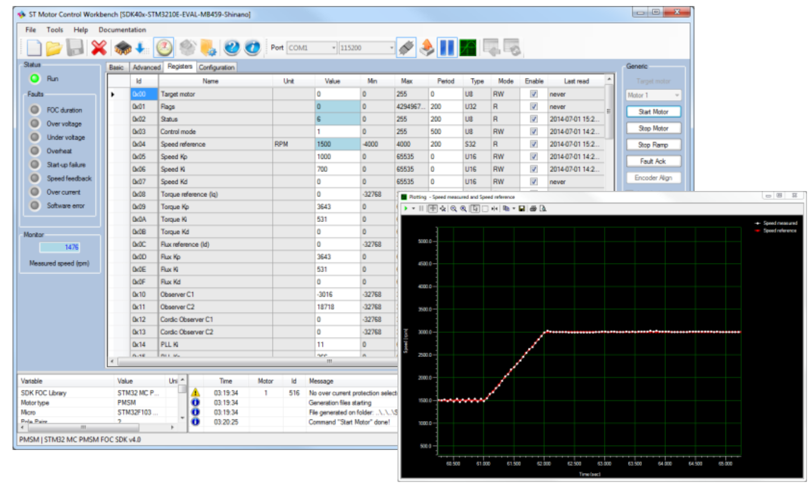
life.augmented

Fast unidirectional/Bidirectional communication

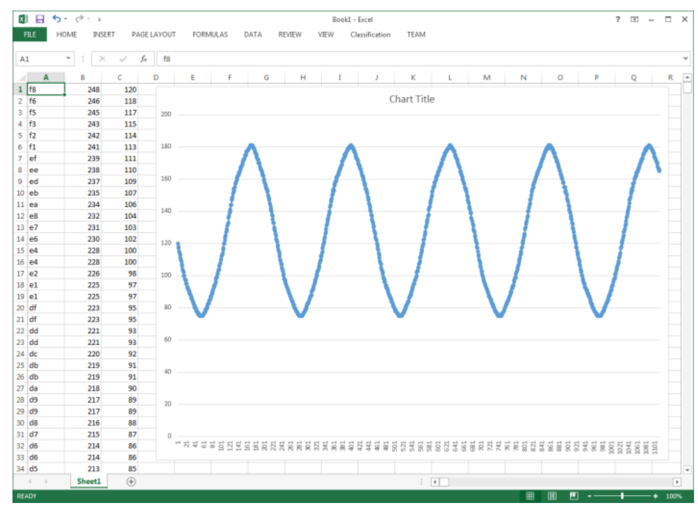
- With fast unidirectional communication the data is send from the firmware to the PC at the maximum speed rate without control bytes



Bidirectional + Workbench



Fast unidirectional + Excel

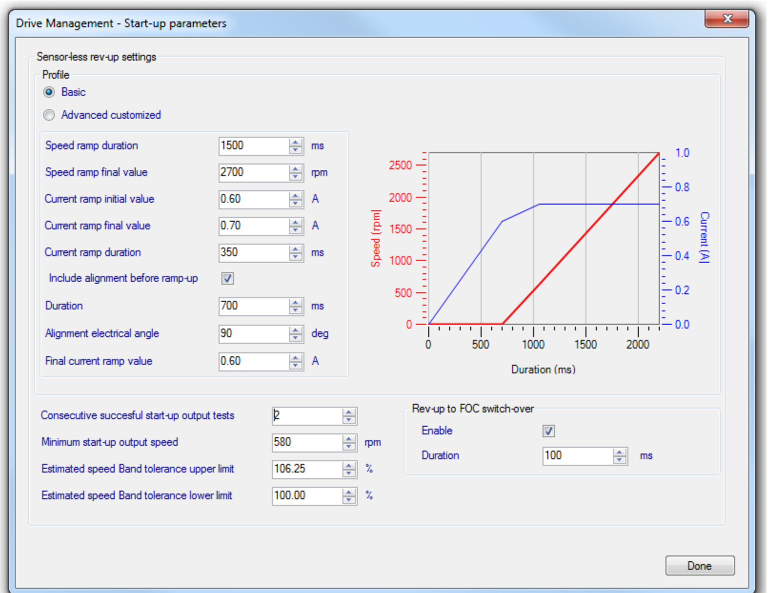


life.augmented

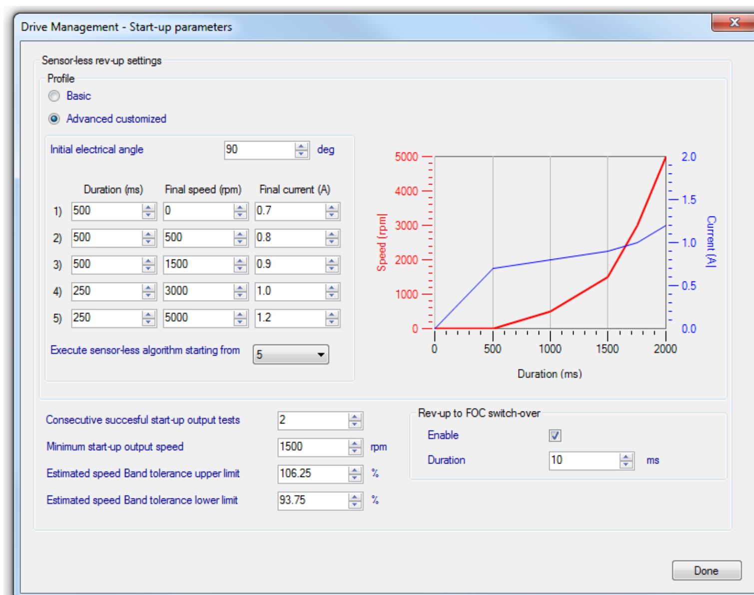
Workbench sensorless startup

- Graphical representation of the imposed current and acceleration.
- Will be fully presented in another session.

Basic



Advanced



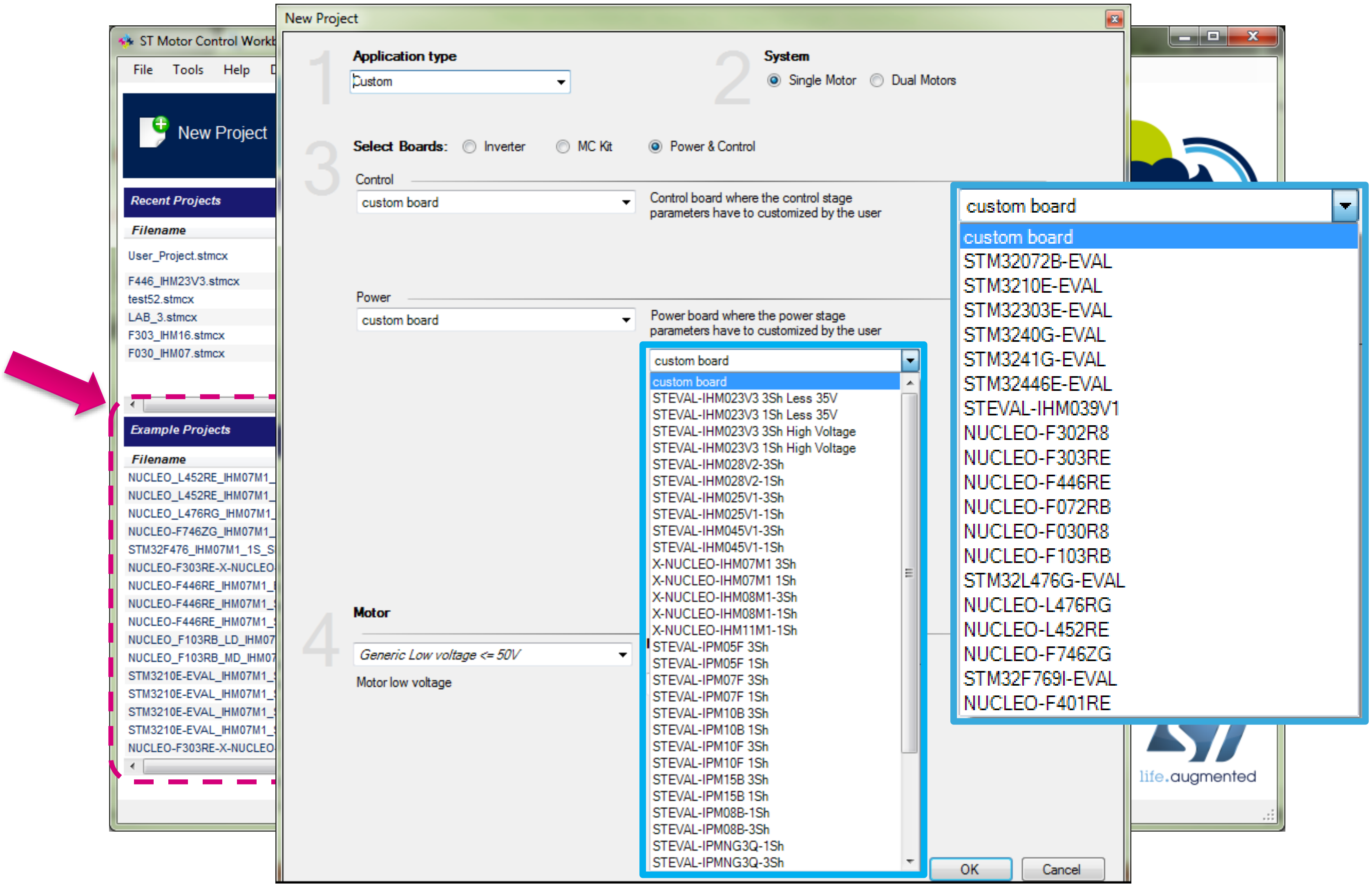
life.augmented

Workbench examples

- The number of example configuration is growing

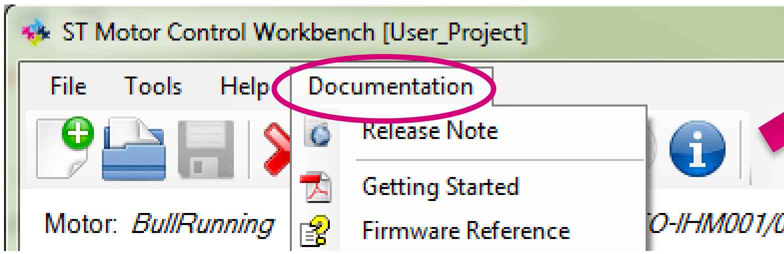
Workbench examples & new project

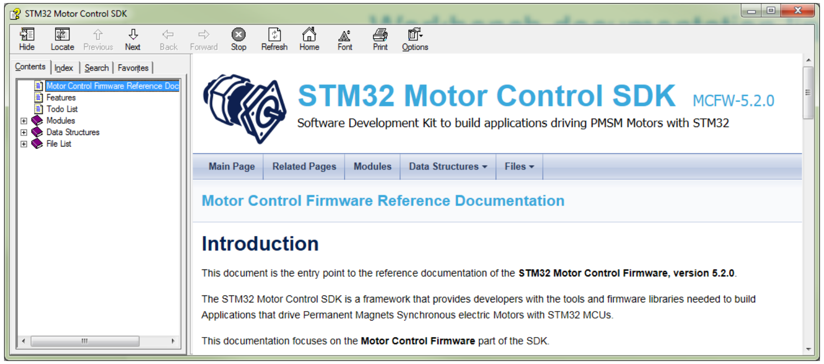
- New face in Workbench - STM32 FOC SDK 5.2.0



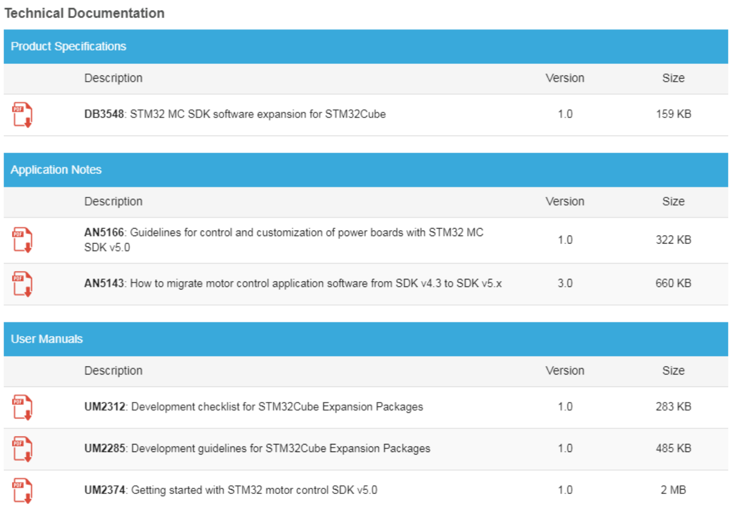
Workbench documentation link

- Direct link to documentation in ST Motor Control Workbench





- Or in ST website X-CUBE-MCSDK

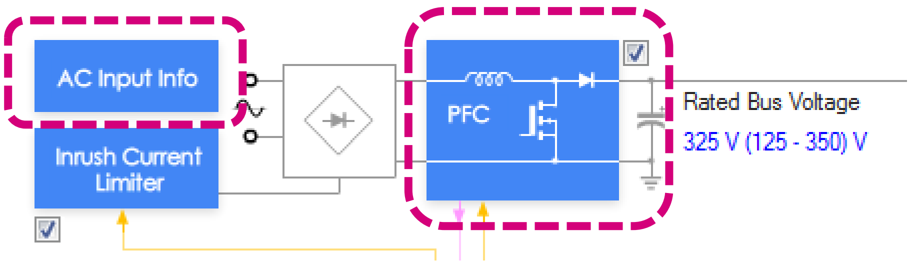


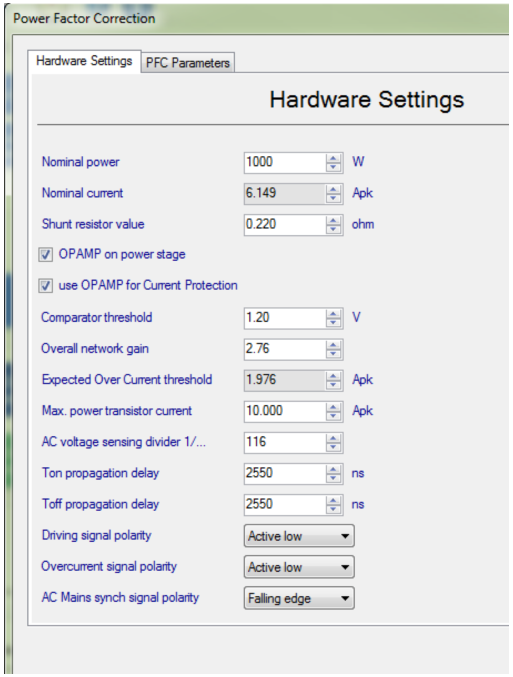
Technical Documentation

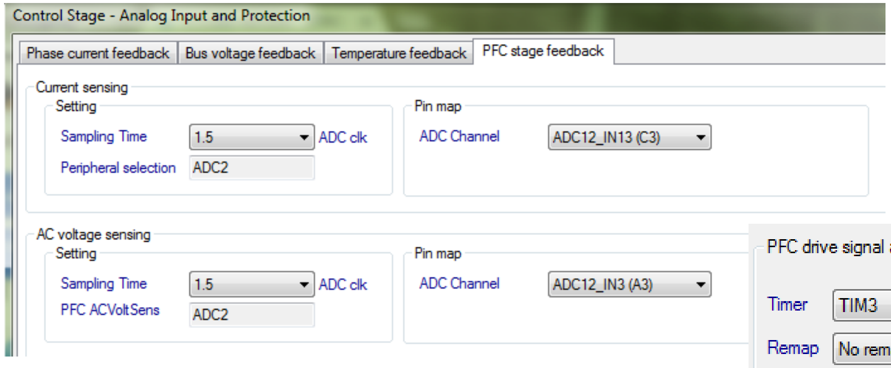
	Description	Version	Size
Product Specifications			
	DB3548: STM32 MC SDK software expansion for STM32Cube	1.0	159 KB
Application Notes			
	AN5166: Guidelines for control and customization of power boards with STM32 MC SDK v5.0	1.0	322 KB
	AN5143: How to migrate motor control application software from SDK v4.3 to SDK v5.x	3.0	660 KB
User Manuals			
	UM2312: Development checklist for STM32Cube Expansion Packages	1.0	283 KB
	UM2285: Development guidelines for STM32Cube Expansion Packages	1.0	485 KB
	UM2374: Getting started with STM32 motor control SDK v5.0	1.0	2 MB

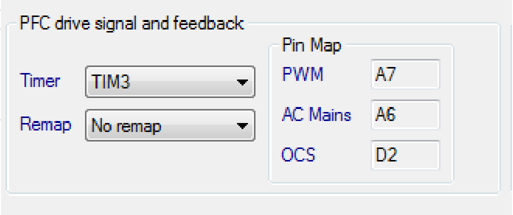
Workbench digital PFC support

- Complete support for the digital PFC plug-in*









life.augmented

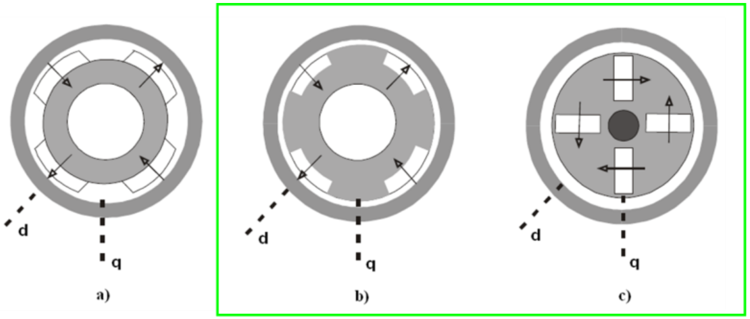
* Digital PFC plug-in is supported only for STM32F103 High density

Patented

HFI sensorless algorithm

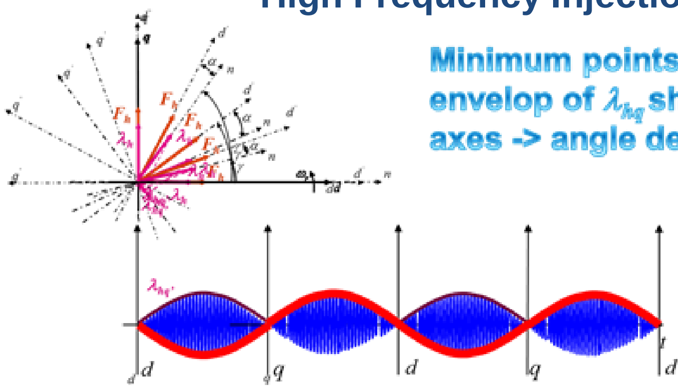
- HFI sensor-less algorithm for I-PMSM is able to detect rotor angular position at ***zero speed*** exploiting the peculiar anisotropy of their magnetic structure.

- Will be fully presented in another session.



a)SM-PMSM: Surface mounted permanent magnets -> $L_d=L_q$
b) & c)I-PMSM: Internal permanent magnets -> $L_d<L_q$
In particular: b) inset magnets; c) radial buried magnets.

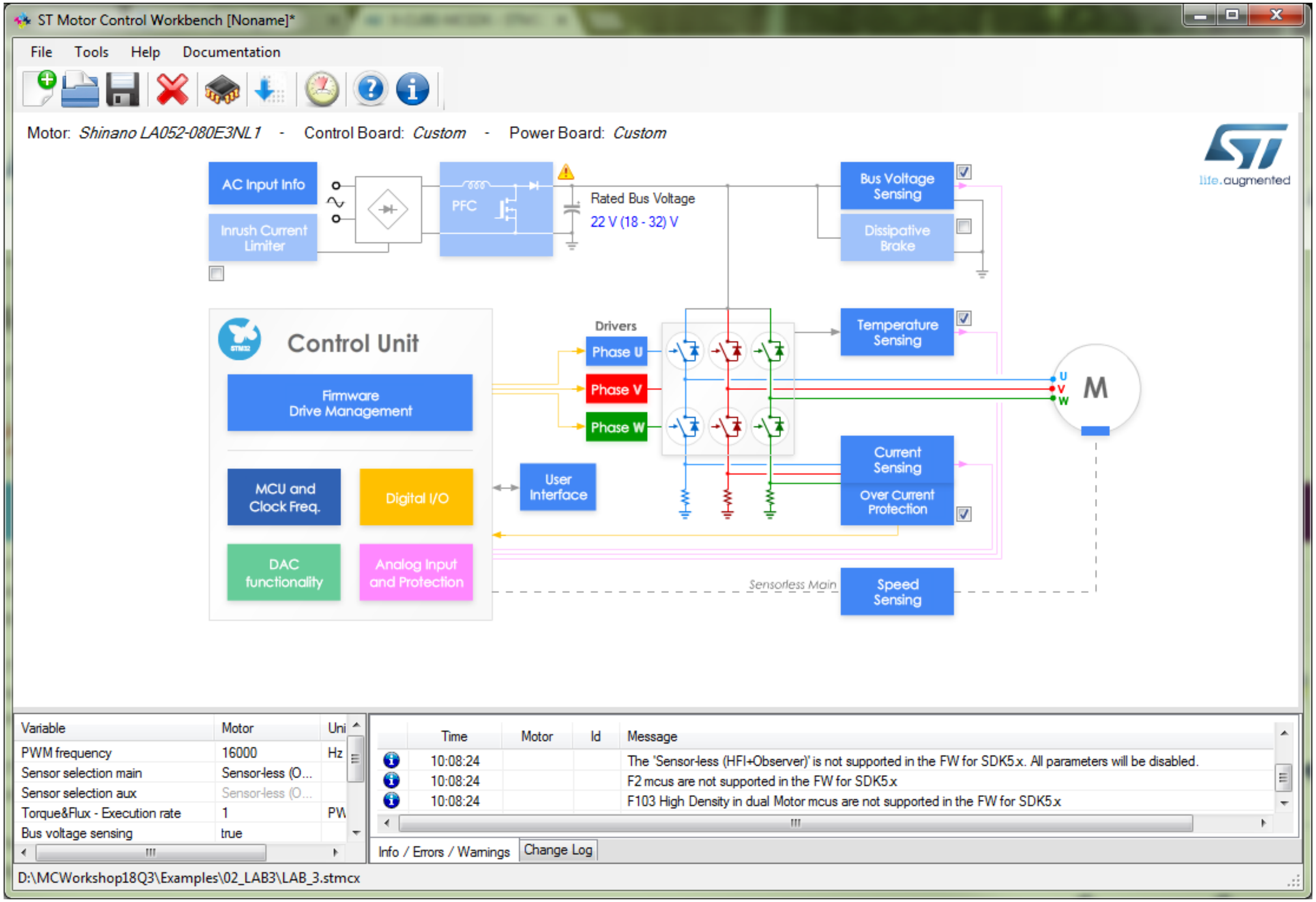
High Frequency Injection



Minimum points of the envelop of λ_{hq} show d/q axes -> angle detection !

Motor Control Workbench

- New face in Workbench - STM32 FOC SDK 5.2.0



life.augmented

Thank you