UM1080
User manual

Quick start guide for STM32F PMSM single/dual FOC SDK v4.3

Introduction

The STM32 microcontroller offers the performance of the industry-standard Cortex®-M core at the service of vector or field oriented control (FOC) algorithms, widely used in high-performance drives.

The STM32 PMSM FOC SDK (STSW-STM32100), which includes the PMSM FOC FW library and ST MC Workbench, gives you the opportunity to evaluate the performance of the STM32 microcontroller in applications driving single or dual FOC of three-phase permanent magnet motors (PMSM, BLDC).

STM32 PMSM FOC SDK is part of ST's motor control ecosystem which offers a wide range of hardware and software solutions for motor control applications.

The ST MC Workbench software for the PC reduces design time and effort with respect to the STM32 PMSM FOC firmware library configuration. Through its graphical user interface (GUI), you can generate all the parameter header files to configure the library according to your application needs, as well as monitor and adjust certain algorithm variables in real time.

Contents

List of tables

List of figures

1 Motor control library features

- Motor profiler:
  - a new algorithm able to auto-measure electromechanical parameters of PMSM motors (STM32F30x and STM32F4xx only).
- One touch tuning:
  - a new algorithm that uses a single parameter to set up the speed controller according to the type of load. Together with the Motor profiler, it can be enabled to achieve the setup and run of an unknown motor from the scratch (only the STM32F30x and STM32F4xx).
- On-the-fly sensorless startup, a new algorithm able to detect if the motor is running before the startup and skip the acceleration phase if not necessary. The motor is run in FOC from the begin without need to stop it before the start. This feature is particular useful for fan application (any STM32F supported).
- Single or simultaneous Dual PMSM FOC
  - sensorless/sensored (Dual PMSM FOC only when running on STM32F103xx High-Density, STM32F103xx XL-Density, STM32F2xx, STM32F303xB/C or STM32F4xx)
- Speed feedbacks:
  - Sensorless (High Frequency Injection HFI plus B-EMF State Observer, PLL rotor speed/angle computation from B-EMF, only for STM32F30x or STM32F4xx);
  - Sensorless (B-EMF State Observer, PLL rotor speed/angle computation from BEMF);
  - Sensorless (B-EMF State Observer, CORDIC rotor angle computation from BEMF);
  - 60° or 120° displaced Hall sensors decoding, rising/falling edge responsiveness;
  - Quadrature incremental encoder;
  - For each motor, dual simultaneous speed feedback processing;
  - On-the-fly speed sensor switching capability;
- Current sampling methods:
  - Two ICS (only when running on STM32F103xx, STM32F2xx, or STM32F4xx);
  - Single, common DC-link shunt resistor (ST patented);
  - Three shunt resistors placed on the bottom of the three inverter legs (only when running on STM32F103xx, STM32F2xx, STM32F302xB/C, STM32F303xB/C or STM32F4xx);
- Embedded analog (STM32F30x only):
  - PGA (Programmable Gain Amplifiers) for current sensing: support for three-shunt and single shunt, internal and external gain;
  - Comparators for overcurrent protection: support for three-shunt and single shunt, internal and external threshold;
  - Comparators for overvoltage protection: support for motor phases short-circuiting mode and free-wheeling mode, internal and external threshold;
- FOC hardware acceleration (STM32F30x only);
  - ADC queue of context (ST patented architecture) support;
  - CCM (core coupled memory) RAM support;
  - Advanced Timer structures for single shunt (ST patented) support;
- Flux weakening algorithm to attain higher than rated motor speed (optional);
- Feed-Forward, high performance current regulation algorithm (optional);
- SVPWM generation:
  - Centered PWM pattern type;
  - Adjustable PWM frequency;
- Torque control mode, speed control mode; on-the-fly switching capability;

- Brake strategies (optional):
  - Dissipative DC link brake resistor handling;
  - Motor phases short-circuiting (with optional hardware over-current protection disabling);
  - motor phases free-wheeling;
- When running Dual FOC, any combination of the above-mentioned speed feedback, current sampling, control mode, optional algorithm;
- Optimized I-PMSM and SM-PMSM drive;
- Programmable speed ramps (parameters duration and final target);
- Programmable torque ramps (parameters duration and final target);
- Real-time fine tuning of:
  - PID regulators;
  - Sensorless algorithm;
  - Flux weakening algorithm;
  - Startup procedure (in case of sensorless);
- Fault management:
  - Overcurrent;
  - Overvoltage;
  - Overtemperature;
  - Speed feedback reliability error;
  - FOC algorithm execution overrun;
- Easy customization of options, pin-out assignments, CPU clock frequency through ST MC Workbench GUI;
- C language code:
  - Compliant with MISRA-C 2004 rules;
  - Conforms strictly with ISO/ANSI;
  - Object-oriented programming architecture;

1.1 User project and interface features

There are two available options:

- FreeRTOS-based user project (for STM32F103xx and STM32F2xx only);
- SysTick-timer-easy-scheduler-based user project;

Available User Interface options (and combinations of them):

- Full LCD plus joystick;
- Light LCD plus joystick;
- Serial communication protocol bidirectional (compatible with ST MC Workbench GUI);
- Serial communication protocol fast unidirectional;
- Drive system variables logging/displaying via:
  - SPI;
  - DAC (DAC peripheral is not present in the STM32F103xx low or medium density; in this case, RC-filtered PWM signal option is available);

2 Documentation architecture

2.1 Where to find the information you need

Technical information about the MC SDK is arranged thus:

- User manual UM1052: STM32F PMSM single/dual FOC SDK v4.3 (available on www.st.com):
  - features
  - architecture
  - workspace
  - customization processes
  - overview of algorithms implemented (FOC, current sensors, speed sensors, embedded analog topologies supported)
  - MC API
  - sample user project
  - sample LCD user interface
  - sample serial communication protocol
- User manual UM1053: Advanced developers guide for STM32F MCUs PMSM single/dual FOC library (available on www.st.com):
  - object oriented programming style used for developing the MC library
  - description of classes that belong to the MC library
  - interactions between classes
  - description of tasks of the MCA
- MC library source documentation (Doxygen compiled HTLM file).
  - describes the public interface of each class of the MC library (methods, parameters required for object creation)
- MC Application source documentation (Doxygen compiled HTML file).
  - describes the classes that make up the MC API
- User Interface source documentation (Doxygen compiled HTML file).
  - describes the classes that make up the UI library
- STM32F0xx, STM32F10xx, STM32F2xx, STM32F30x or STM32F4xx Standard Peripherals Library source documentation (Doxygen compiled HTML file).
- ST MC Workbench GUI documentation.
  - describes the steps and parameters required to customize the library, as shown in the GUI
- In-depth documentation of certain algorithms (sensorless position/speed detection, flux weakening, MTPA, feed-forward current regulation).

Visit www.st.com or contact your nearest ST sales office for further information.

2.2 Related documents

Available from www.arm.com

- Cortex®-M0 Technical Reference Manual, available from: http://infocenter.arm.com.
- Cortex®-M3 Technical Reference Manual, available from: http://infocenter.arm.com.
- Cortex®-M4 Technical Reference Manual, available from:http://infocenter.arm.com.

Available from www.st.com or your STMicroelectronics sales office

- STM32F030x datasheets
- STM32F051x datasheets

- STM32F100xx datasheets
- STM32F103xx datasheets
- STM32F20x and STM32F21x datasheets
- STM32F302x6/8 datasheets
- STM32F302xB/C datasheets
- STM32F303xB/C datasheets
- STM32F40x and STM32F41x datasheets
- STM32F051x user manual (RM0091)
- STM32F100xx user manual (RM0041)
- STM32F103xx user manual (RM0008)
- STM32F20x and STM32F21x user manual (RM0033)
- STM32F30x user manual (RM0316)
- STM32F40x and STM32F41x user manual (RM0090)
- STM32F103xx AC induction motor IFOC software library V2.0 (UM0483)
- STM32 and STM8 Flash Loader demonstrator (UM0462)

3 The new Motor Profiler procedure

1 Execute the “STMotorProfiler.exe” application from the Motor Profiler (MP) button shown in the ST Motor Control Workbench figured below.

Alternatively, you can run the file from the “STMotorProfiler” folder in the installation folder.

Figure 1: Link to ST Motor Profiler

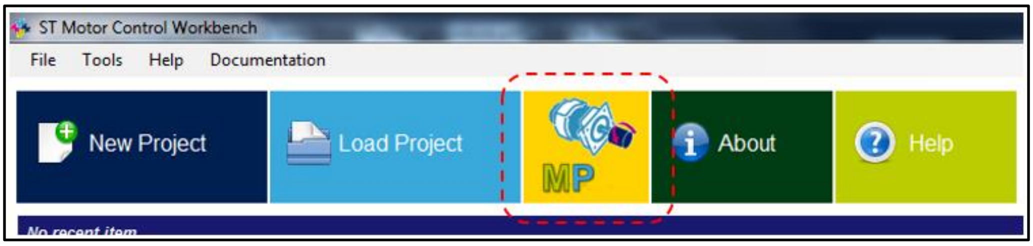
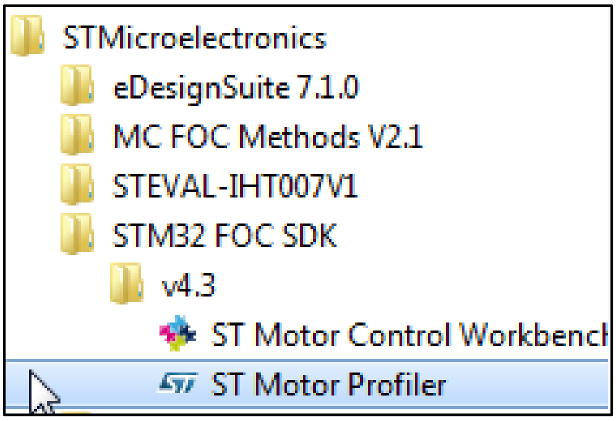
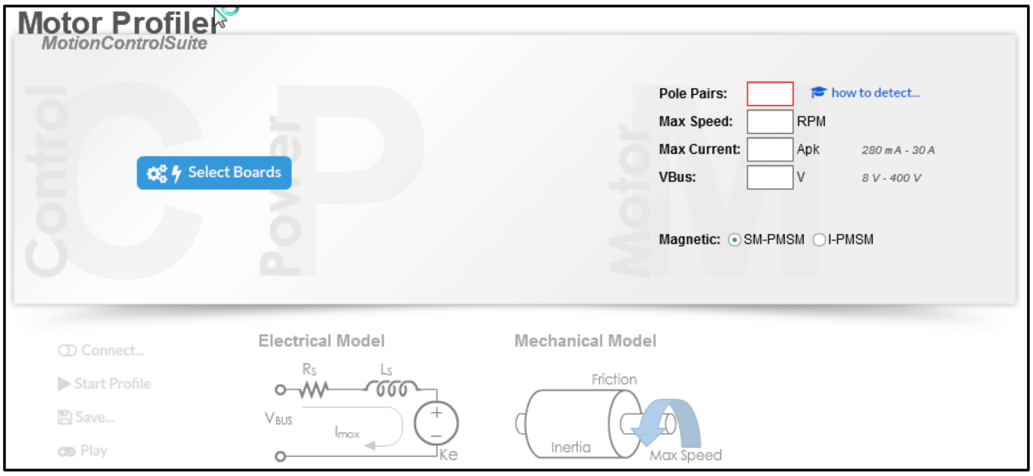


Figure 2: Installation folder tree



STMicroelectronics
eDesignSuite 7.1.0
MC FOC Methods V2.1
STEVAL-IHT007V1
STM32 FOC SDK
v4.3
ST Motor Control Workbench
ST Motor Profiler

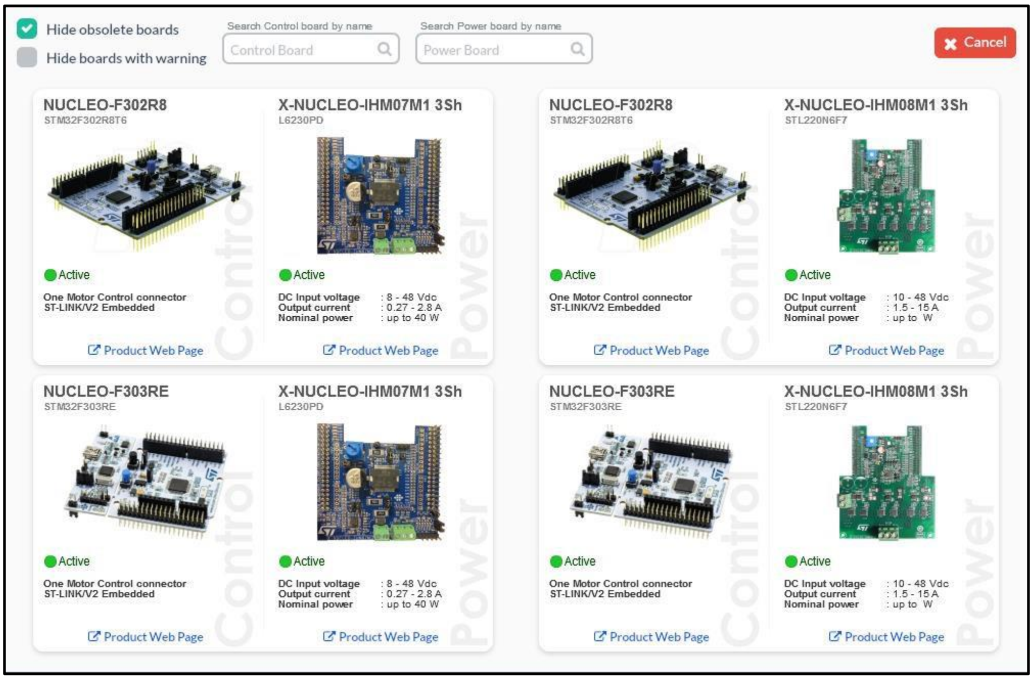
Figure 3: ST Motor Profiler



2 Click “Select Boards” button in the Motor Profiler opening screen to open a list of supported boards.

The Motor Profiler feature can be used only in the systems listed there.

Figure 4: List of compatible systems



☑ Hide obsolete boards

□ Hide boards with warning

Search Control board by name: Control Board

Search Power board by name: Power Board

NUCLEO-F302R8
STM32F302R8T6
● Active
One Motor Control connector
ST-LINK/V2 Embedded
Product Web Page

X-NUCLEO-IHM07M1 3Sh
L6230PD
● Active
DC Input voltage : 8 - 48 Vdc
Output current : 0.27 - 2.8 A
Nominal power : up to 40 W
Product Web Page

NUCLEO-F302R8
STM32F302R8T6
● Active
One Motor Control connector
ST-LINK/V2 Embedded
Product Web Page

X-NUCLEO-IHM08M1 3Sh
STL220N6F7
● Active
DC Input voltage : 10 - 48 Vdc
Output current : 1.5 - 15 A
Nominal power : up to W
Product Web Page

NUCLEO-F303RE
STM32F303RE
● Active
One Motor Control connector
ST-LINK/V2 Embedded
Product Web Page

X-NUCLEO-IHM07M1 3Sh
L6230PD
● Active
DC Input voltage : 8 - 48 Vdc
Output current : 0.27 - 2.8 A
Nominal power : up to 40 W
Product Web Page

NUCLEO-F303RE
STM32F303RE
● Active
One Motor Control connector
ST-LINK/V2 Embedded
Product Web Page

X-NUCLEO-IHM08M1 3Sh
STL220N6F7
● Active
DC Input voltage : 10 - 48 Vdc
Output current : 1.5 - 15 A
Nominal power : up to W
Product Web Page

3 Select the system that you are using.

For example, if you are using the P-NUCLEO-IHM001, select the system with NUCLEO-F302R8 plus X-NUCLEO-IHM07M1.

4 Insert the number of pole pairs on your motor.

This field is mandatory.

5 Insert the Max Speed field.

This field not mandatory; if not selected, the Motor Profiler attempts to reach the maximum allowed speed according to the motor and to the system you are using.

6 Insert the Max Current field.

This field not mandatory; it is expressed in amperes and represents the maximum peak current delivered to the motor.

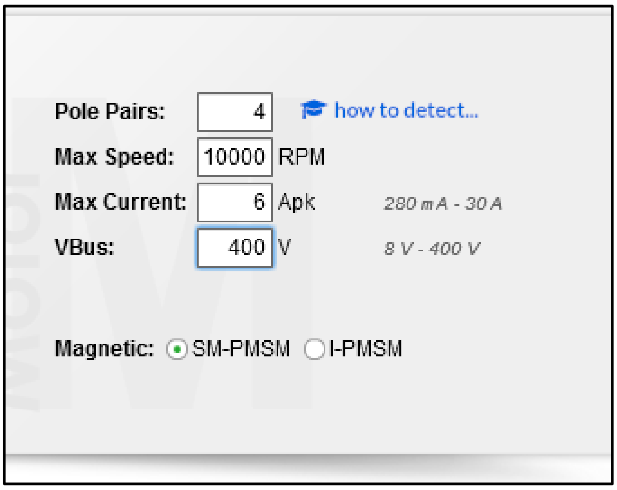
7 Insert the expected bus voltage provided to the system.

This field not mandatory; it is the DC bus voltage supplied to the power stage for low voltage applications or $\sqrt{2}V_{ACrms}$ for applications supplied with a high voltage AC power supply.

8 Select SM-PMSM in the magnetic field if the motor has a surface permanent magnet.

It is the default setting.

Figure 5: Example of settings for surface permanent magnet motors



Pole Pairs: 4

Max Speed: 10000 RPM

Max Current: 6 Apk 280 mA - 30 A

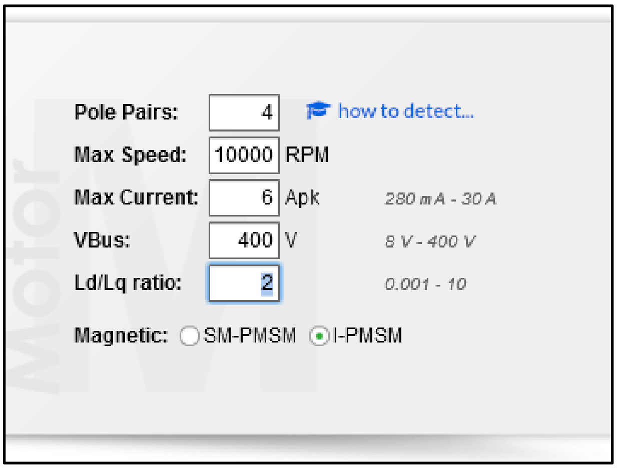
VBus: 400 V 8 V - 400 V

Magnetic: ● SM-PMSM ○ I-PMSM

9 If the motor has an internal permanent magnet, select I-PMSM.

You must also input the Ld/Lq ratio.

Figure 6: Example of settings for internal permanent magnet motors



Pole Pairs: 4

Max Speed: 10000 RPM

Max Current: 6 Apk 280 mA - 30 A

VBus: 400 V 8 V - 400 V

Ld/Lq ratio: 2 0.001 - 10

Magnetic: ○ SM-PMSM ● I-PMSM

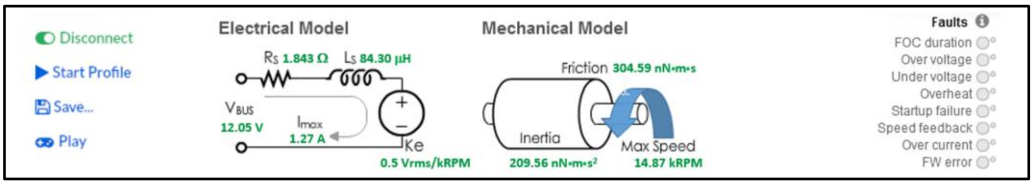
10 Check the boards are correctly configured for motor control, supplied with the expected input voltage and connected with the serial communication cable and JTAG/SWD programming cable. Check the user manual of each evaluation board for the correct configuration, voltage range, serial communication capabilities and programming/debugging interface.

11 Click on the “Connect” button.

- If the board is new or has been erased, the correct FW is automatically loaded into the microcontroller.
- If the programming procedure can’t be executed, check the JTAG/SWD programming cable.
- If the programming procedure is executed but Motor Profiler still can’t communicate with the board, check the serial communication connections.

12 Once connection is established, press the profile button to start the procedure.

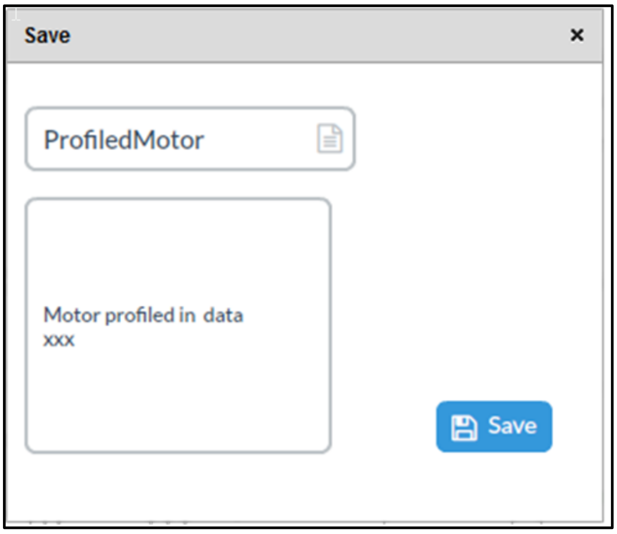
Figure 7: Motor Profiler results



Rs 1.843 Ω
Ls 84.30 µH
VBUS 12.05 V
Imax 1.27 A
Ke 0.5 Vrms/kRPM
Friction 304.59 nN•m•s
Inertia 209.56 nN•m•s²
Max Speed 14.87 kRPM

13 Click on the “Save” to store the motor profiled motor for later use with ST MC Workbench.

Figure 8: Save motor

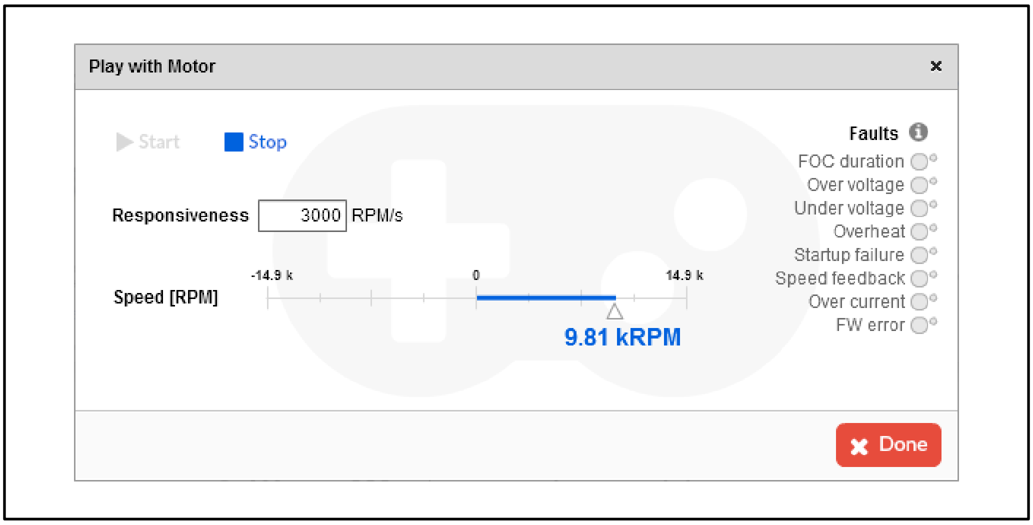


ProfiledMotor

Motor profiled in data
xxx

14 At the end of the procedure, the motor is recognized and can be controlled via the Play button without having to repeat the Motor Profiler procedure. The user can start and stop the motor, adjust the speed and accelerate in the clockwise or anti-clockwise directions.

Figure 9: Play mode



Responsiveness 3000 RPM/s

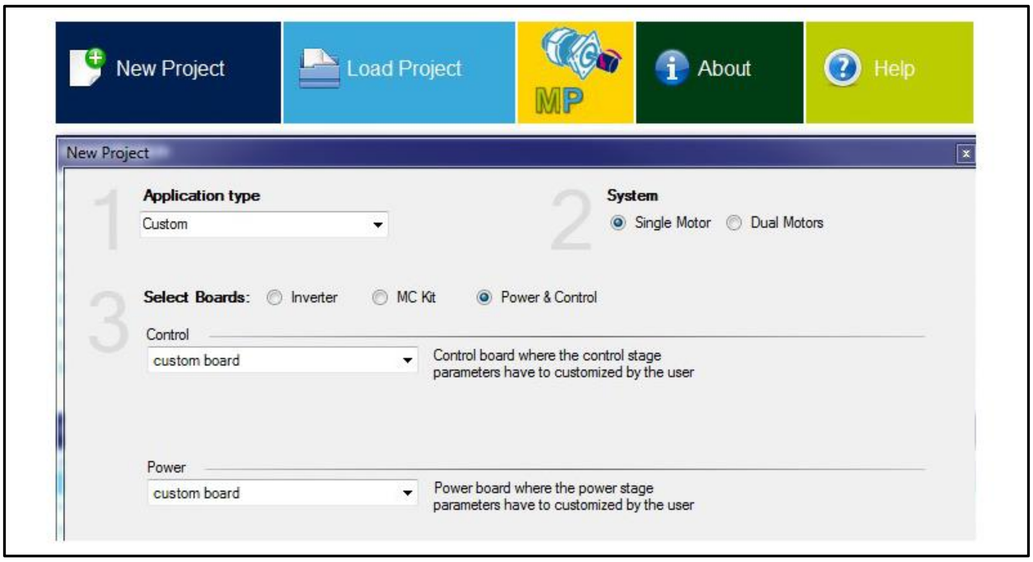
Speed [RPM] 9.81 kRPM

15 If the Motor Profiler procedure is required, click the Profile button to force a new procedure.

16 When the motor is successfully profiled, open ST MC Workbench and select “New Project”.

- Select the “Application Type”, Single Motor or Dual Motor, and the board to be used.
- If the board is a complete inverter (single board with power and control), select “Inverter” combo box and then select the inverter in the list box.
- If an MC Kit is used (like the P-NUCLEO-IHM001), select “MC Kit” combo box and then select the kit in the list box.
- If a system composed by a control evaluation board plus power evaluation board is used, select “Power & Control” combo box and select both Control board and Power board.

Figure 10: Workbench new project creation



Application type: Custom

System: ● Single Motor ○ Dual Motors

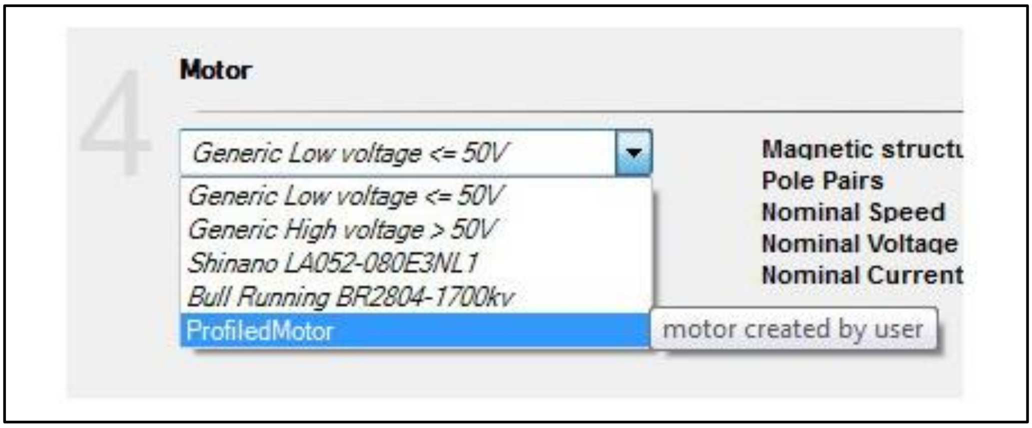
Select Boards: ○ Inverter ○ MC Kit ● Power & Control

Control: custom board — Control board where the control stage parameters have to customized by the user

Power: custom board — Power board where the power stage parameters have to customized by the user

17 Then select the motor from the “Motor” list box with all the saved motor profiles.

Figure 11: ST MC Workbench "Motor" list box



Motor: Generic Low voltage <= 50V

- Generic Low voltage <= 50V
- Generic High voltage > 50V
- Shinano LA052-080E3NL1
- Bull Running BR2804-1700kv
- ProfiledMotor

Magnetic structu
Pole Pairs
Nominal Speed
Nominal Voltage
Nominal Current

The created project imports the HW settings according to the selected boards and the motor settings according the Motor profiler results; it also imports other settings like PWM frequency and startup acceleration used during the Motor Profiler procedure.

18 Proceed with the usual generation of the .h file, compilation and flashing as described in Chapter 9 or in the “Hands-on” (STM32 PMSM FOC SDK Hands-on workshop with hardware tools).

3.1 Restrictions and disclaimer

The Motor Profiler algorithm is intended for rapid evaluation of the ST motor control solution. You can use it to drive any three-phase motor with an internal permanent magnet without any specific instruments or special skills.

The measurements performed, however, cannot be as precise as proper instrumentation.

The Motor Profiler measurement errors can be particularly significant for motors with parameters those indicated in the following table.

Moreover, it is important to choose the appropriate HW according to the characteristics of the motor. For instance, the maximum current of the motor should match the maximum current of the board as closely as possible.

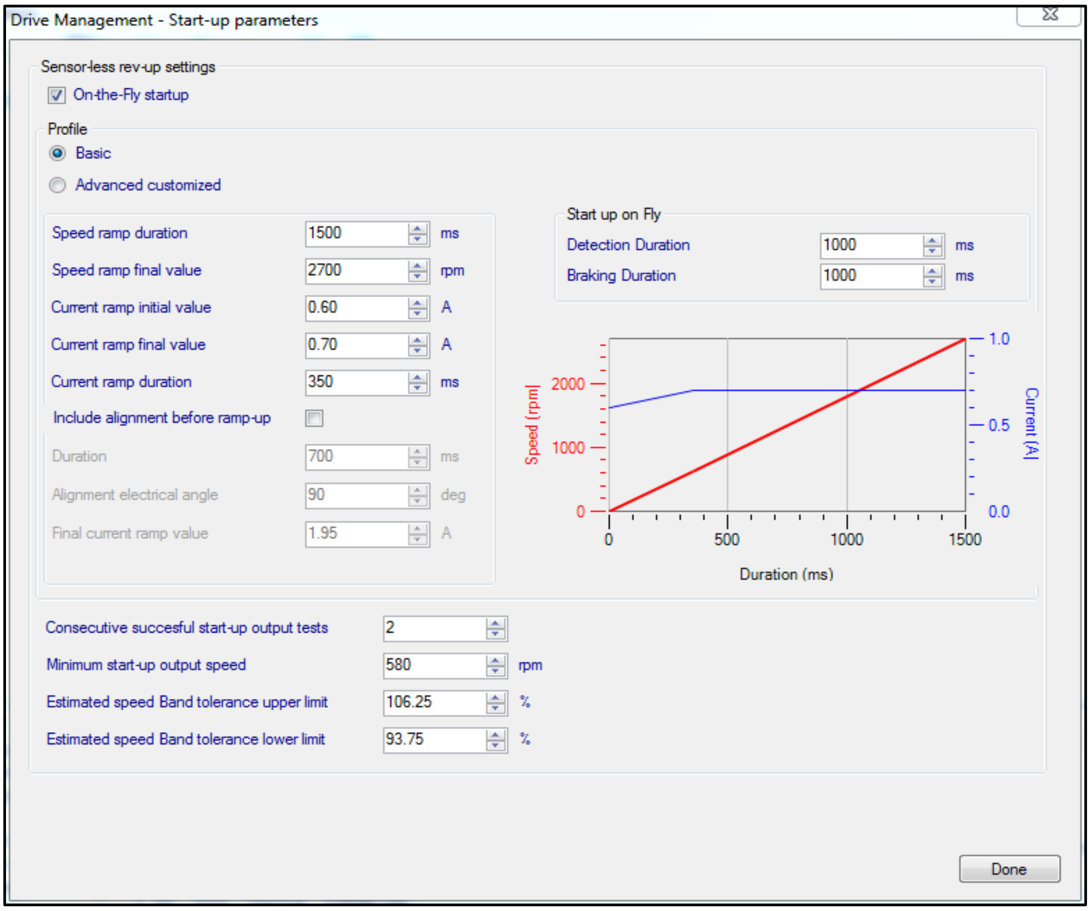
Motor Profiler can be used only using compatible ST evaluation boards. Please use the STMotorProfiler tool to view the list of the supported systems.

4 On-the-fly sensorless startup

The on-the-fly sensorless startup algorithm can detect if the motor is running before startup and skip the acceleration phase if it is not required. If the motor runs at a speed that is above the allowed threshold, the firmware applies the FOC from the beginning, without having to stop and restart it.

This feature is particularly useful for fan applications and can be enabled when Sensor-less (Observer+PLL) or Sensor-less (Observer+Cordic) is selected in the Drive Management – Speed Position Feedback Management dialog, by checking the On-the-Fly startup check box.

Figure 12: Enabling on-the-fly startup



Drive Management - Start-up parameters

Sensor-less rev-up settings

☑ On-the-Fly startup

Profile

● Basic

○ Advanced customized

Speed ramp duration	1500	ms
Speed ramp final value	2700	rpm
Current ramp initial value	0.60	A
Current ramp final value	0.70	A
Current ramp duration	350	ms
Include alignment before ramp-up	□	
Duration	700	ms
Alignment electrical angle	90	deg
Final current ramp value	1.95	A

Start up on Fly

Detection Duration	1000	ms
Braking Duration	1000	ms

Consecutive succesful start-up output tests	2	
Minimum start-up output speed	580	rpm
Estimated speed Band tolerance upper limit	106.25	%
Estimated speed Band tolerance lower limit	93.75	%

Minimum startup-output speed represents the minimum speed for which the sensorless observer provides reliable measurements and is used to determine if it is possible to skip the acceleration phase. It can be set with reference to the nominal speed of the motor.

After enabling on-the-fly startup, the following parameters become available:

- Detection Duration – the OTF startup detection phase duration (in ms) of in which the reliability of the sensorless measurements are tested in order to validate the speed and run directly in FOC.
- Braking Duration – the braking phase duration (in ms) applied if the sensorless measurements are not reliable during the detection phase; the motor is brought to a stop before the new acceleration is applied.

Both basic and advanced startup profiles can be used if the OTF startup is enabled. The startup profile can be set by the user to define the acceleration strategy if the speed of the motor is below the reliability threshold during the detection phase.

5 Development environment

A suitable motor control SDK environment includes:

- a PC
- a third-party integrated development environment (IDE)
- a third-party C-compiler
- a JTAG/SWD interface for debugging and programming
- an application board with an STM32F0xx, STM32F100xx, STM32F103xx, STM32F2xx, STM32F30x or STM32F4xx properly designed to drive its power stage (PWM outputs to gate driver, ADC channels to read currents, DC bus voltage). Many evaluation boards are available from ST, some of them have an ST-LINK programmer on-board.
- a three-phase PMSM motor

5.1 Motor control workspace

Figure 17: "Customization process" shows the following projects in the motor control workspace:

1. **MC Library project**: with all the classes developed to implement the features; it is built as a compiled library, not an executable file.
2. **User project**: with the MC application layer and the demonstration program that uses the layer through its MC API and provides the required clocking and access to Interrupt Handlers. User API parameters and configurations are used here to create proper objects in what is called the run-time system 'boot'. The motor control API is the set of commands granted to the upper layer. The program can run some useful functions (depending on user options), such as serial communication, LCD/keys interface and system variable display through DAC.

There are two equivalent project alternatives: one implements its own simple time base; the other exploits an Operating System, FreeRTOS, to do it.

The available project workspaces differ by supported STM32 family, IDE supported, how they generate the clocks: a simple time base itself or an Operating System (FreeRTOS).

The fist 8 are for IAR™ EWARM IDE and are stored in the folder Project\\EWARM:

- STM32F0xx_Workspace for STM32F0xx devices and simple time base;
- STM32F10x_Workspace for both STM32F100xx and STM32F103xx devices and simple time base;
- STM32F2xx_Workspace for STM32F2xx devices and simple time base;
- STM32F30x_Workspace for STM32F302/303 devices and simple time base;
- STM32F4xx_Workspace for STM32F4xx devices and simple time base;
- STM32F10x_RTOS_Workspace for both STM32F100xx and STM32F103xx devices and FreeRTOS;
- STM32F2xx_RTOS_Workspace for STM32F2xx devices and FreeRTOS;
- STM32F10x_Example for both STM32F100xx and STM32F103xx devices with simple time base and ready-to-use examples.

The remaining 5 are for Keil uVision and are stored in the folder Project\\MDK-ARM:

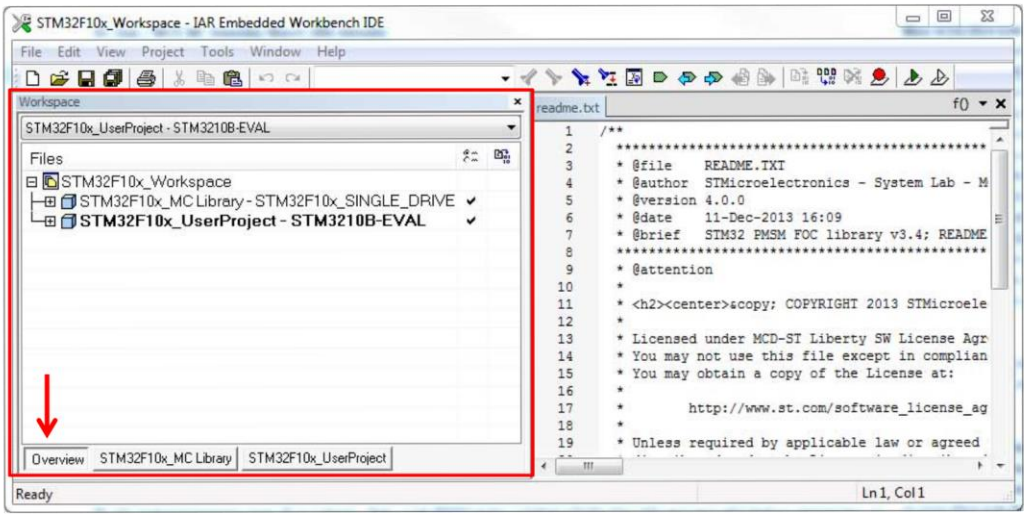
- STM32F0xx_Workspace for STM32F0xx devices and simple time base;
- STM32F10x_Workspace for both STM32F100xx and STM32F103xx devices and simple time base;
- STM32F2xx_Workspace for STM32F2xx devices and simple time base;
- STM32F4xx_Workspace for STM32F4xx devices and simple time base;

- STM32F30x_Workspace for STM32F302/303 devices and simple time base.

Previously, the built ".lib" files are linked with the user project in order to generate the file that can be downloaded into microcontroller memory for execution.

Figure 13: "IAR EWARM IDE Workspace overview" provides an overview of the IAR EWARM IDE workspace (located in Installation folder `\\Project\\EWARM\\STM32F10x_Workspace.eww`). The following sections provide details on this. The equivalent workspace based on FreeRTOS is located in Installation folder `\\FreeRTOSProject\\EWARM\\ STM32F10x_RTOS_Workspace.eww`.

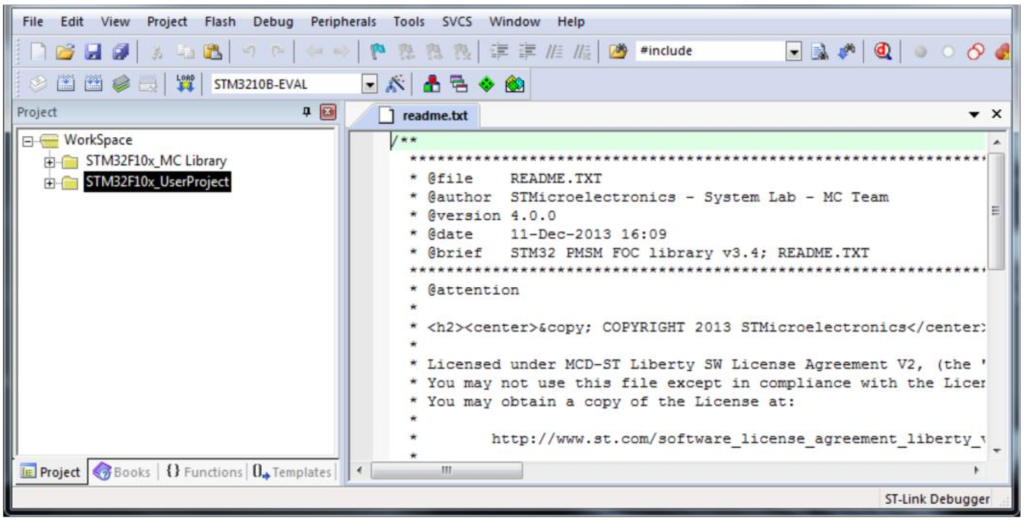
Figure 13: IAR EWARM IDE Workspace overview


STM32F10x_Workspace - IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
STM32F10x_UserProject - STM3210B-EVAL
Files
STM32F10x_Workspace
STM32F10x_MC Library - STM32F10x_SINGLE_DRIVE ✓
STM32F10x_UserProject - STM3210B-EVAL ✓
Overview
STM32F10x_MC Library
STM32F10x_UserProject
readme.txt
f()
Ready
Ln 1, Col 1

```
/**
 ************************************************
 * @file    README.TXT
 * @author  STMicroelectronics - System Lab - M
 * @version 4.0.0
 * @date    11-Dec-2013 16:09
 * @brief   STM32 PMSM FOC library v3.4; README
 ************************************************
 * @attention
 *
 * <h2><center>&copy; COPYRIGHT 2013 STMicroele
 *
 * Licensed under MCD-ST Liberty SW License Agr
 * You may not use this file except in complian
 * You may obtain a copy of the License at:
 *
 *        http://www.st.com/software_license_ag
 *
 * Unless required by applicable law or agreed
```

Figure 14: "Keil uVision workspace overview" provides an overview of the Keil uVision workspace (located in the Installation folder \\Project\\MDK-ARM\\STM32F10x_Workspace.uvmpw).

Figure 14: Keil uVision workspace overview



```
/**
 *****************************************************************
 * @file    README.TXT
 * @author  STMicroelectronics - System Lab - MC Team
 * @version 4.0.0
 * @date    11-Dec-2013 16:09
 * @brief   STM32 PMSM FOC library v3.4; README.TXT
 *****************************************************************
 * @attention
 *
 * <h2><center>&copy; COPYRIGHT 2013 STMicroelectronics</center>
 *
 * Licensed under MCD-ST Liberty SW License Agreement V2, (the '
 * You may not use this file except in compliance with the Licer
 * You may obtain a copy of the License at:
 *
 *        http://www.st.com/software_license_agreement_liberty_v
 *
```

5.2 MC SDK customization process

This section explains how to customize the Motor Control SDK using IAR EWARM IDE or Keil uVision, so that it corresponds with the user's current system.

1 Using the ST MC Workbench GUI configure the firmware according to the HW, motor and specific drive setting of the system. This part of the process ends by generating the .h parameters in the correct directory(`Installation` folder\\SystemDriveParams).

2 If the system is configured to enable the LCD User Interface, download the specific firmware. See *Section 6: "How to download the full LCD user interface"* (this step is to be done only once).

- Using IAR EWARM IDE follow the point: 3, 4, 5, 6.
- Using Keil uVision follow the point 7, 8, 9, 10.

3 Open the MC workspace of choice:

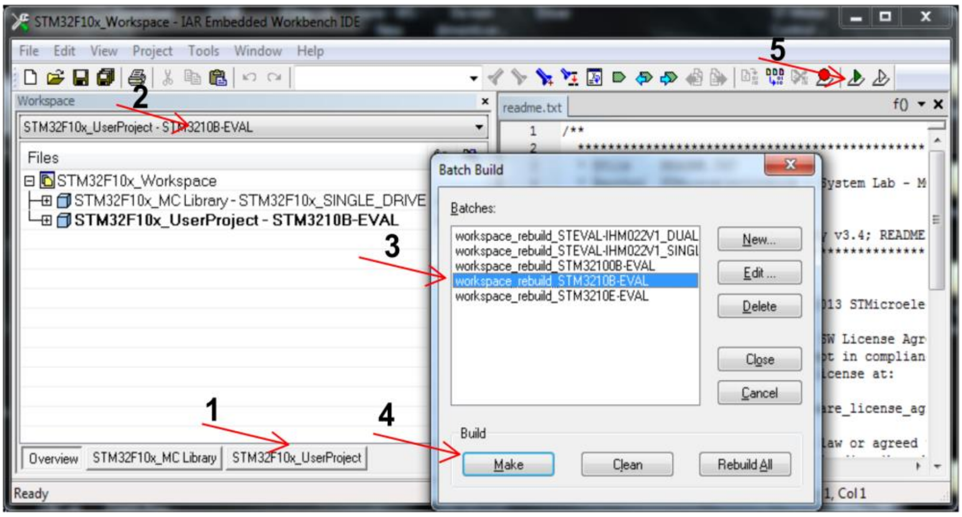
- `Installation folder\\FreeRTOS Project\\EWARM\\STM32F10x_RTOS_Workspace.eww`
- `Installation folder\\FreeRTOS Project\\EWARM\\STM32F2xx_RTOS_Workspace.eww`
- `Installation folder\\Project\\EWARM\\STM32F0xx_Workspace.eww`
- `Installation folder\\Project\\EWARM\\STM32F10x_Workspace.eww`
- `Installation folder\\Project\\EWARM\\STM32F2xx_Workspace.eww`
- `Installation folder\\Project\\EWARM\\STM32F30x_Workspace.eww`
- `Installation folder\\Project\\EWARM\\STM32F4xx_Workspace.eww`

4 Enable the user project (call-out 1 in *Figure 15: "Workspace batch build for IAR EWARM IDE"*) and select the appropriate option from the combo-box (call-out 2 in *Figure 15: "Workspace batch build for IAR EWARM IDE"*). If none of the boards displayed is in use, read *Table 1: "Project configurations"* to perform correct configuration.

5 Press F8 to batch-build the entire workspace. The dialog box shown in *Figure 15: "Workspace batch build for IAR EWARM IDE"* appears.

6 Select a batch command (call-out 3, *Figure 15: "Workspace batch build for IAR EWARM IDE"*) as in step 4, then click the Make button to make the build (call-out 4, *Figure 15: "Workspace batch build for IAR EWARM IDE"*). If no error or relevant warning appears, download the firmware (call-out 5, *Figure 15: "Workspace batch build for IAR EWARM IDE"*) and perform a test run.

Figure 15: Workspace batch build for IAR EWARM IDE



STM32F10x_Workspace - IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
STM32F10x_UserProject - STM3210B-EVAL
Files
STM32F10x_Workspace
STM32F10x_MC Library - STM32F10x_SINGLE_DRIVE
STM32F10x_UserProject - STM3210B-EVAL
Overview
STM32F10x_MC Library
STM32F10x_UserProject
Ready
readme.txt
Batch Build
Batches:
workspace_rebuild_STEVAL-IHM022V1_DUAL
workspace_rebuild_STEVAL-IHM022V1_SINGL
workspace_rebuild_STM32100B-EVAL
workspace_rebuild_STM3210B-EVAL
workspace_rebuild_STM3210E-EVAL
New...
Edit ...
Delete
Close
Cancel
Build
Make
Clean
Rebuild All
1
2
3
4
5

7 Open one of the MC workspaces:

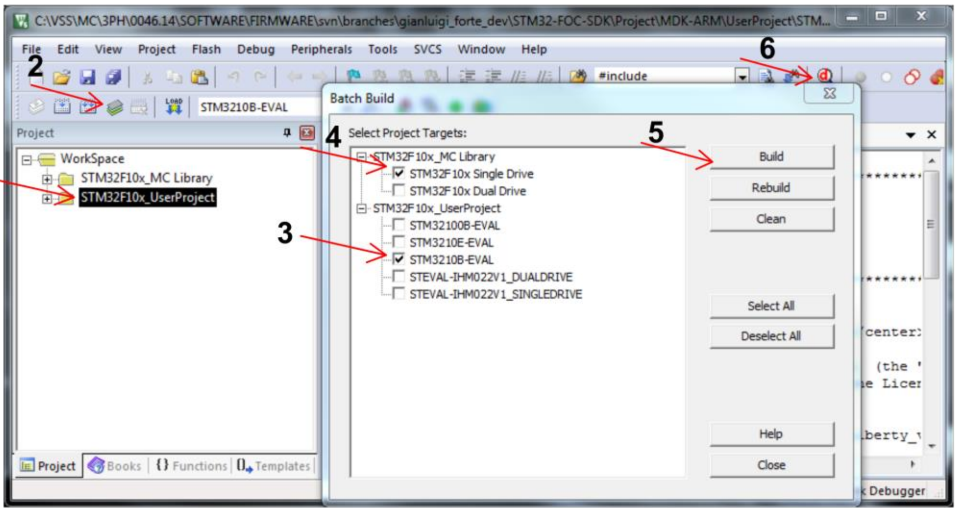
- `Installation folder\\Project\\MDK-ARM\\STM32F0xx_Workspace.uvmpw`
- `Installation folder\\Project\\MDK-ARM\\STM32F10x_Workspace.uvmpw`
- `Installation folder\\Project\\MDK-ARM\\STM32F2xx_Workspace.uvmpw`
- `Installation folder\\Project\\MDK-ARM\\STM32F3xx_Workspace.uvmpw`
- `Installation folder\\Project\\MDK-ARM\\STM32F4xx_Workspace.uvmpw`

8 Enable the UserProject (callout 1 in *Figure 16: "Workspace batch build for Keil uVision"*) right click on it and select "Set as Active Project" according the evaluation board used. If none of the boards displayed is in use, read *Table 1: "Project configurations"*: Project configurations to perform a correct configuration.

9 Press batch build button (callout 2 in *Figure 16: "Workspace batch build for Keil uVision"*) The dialog box shown in *Figure 16: "Workspace batch build for Keil uVision"*: Batch Build appears.

10 Select the configuration to be build according the step 7 (callout 3 in *Figure 16: "Workspace batch build for Keil uVision"*) and selecting the proper conflagration of the MC Library (Single or Dual drive) (callout 4 in *Figure 16: "Workspace batch build for Keil uVision"*). Then click Build button to make the build (callout 5 in *Figure 16: "Workspace batch build for Keil uVision"*). If no error or relevant warning appears, download the firmware (callout 6, *Figure 16: "Workspace batch build for Keil uVision"*) and perform a test run.

Figure 16: Workspace batch build for Keil uVision



C:\VSS\MC\3PH\0046.14\SOFTWARE\FIRMWARE\svn\branches\gianluigi_forte_dev\STM32-FOC-SDK\Project\MDK-ARM\UserProject\STM...

Batch Build

Select Project Targets:

- STM32F10x_MC Library
  - ☑ STM32F10x Single Drive
  - □ STM32F10x Dual Drive
- STM32F10x_UserProject
  - □ STM32100B-EVAL
  - □ STM3210E-EVAL
  - ☑ STM3210B-EVAL
  - □ STEVAL-IHM022V1_DUALDRIVE
  - □ STEVAL-IHM022V1_SINGLEDRIVE

When the system configuration or parameters are modified, just the User project requires to be recompiled.The batch build procedure is requested just if the MC Library is provided as source code and only for the first compilation for both single and dual drive configuration. See *Figure 17: "Customization process"*.

Figure 17: Customization process

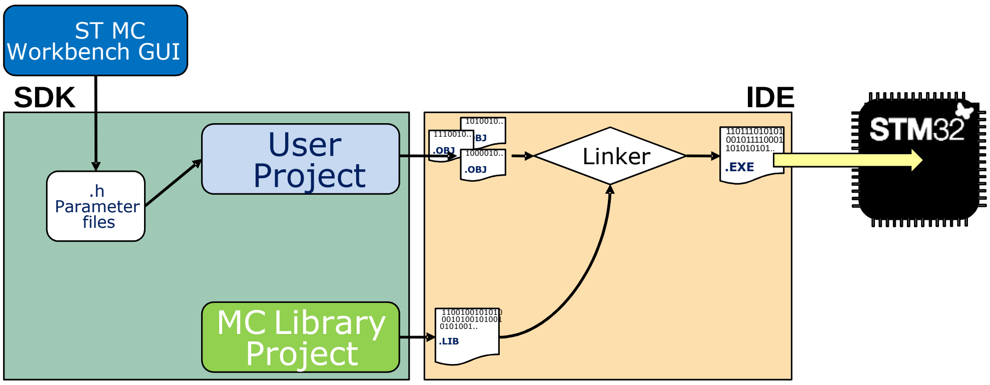


Table 1: Project configurations

STM32 device part, single/dual drive selection	Viable configuration among existing
STM32F0xx, Single motor drive	STM320518-EVAL
STM32F100 low / medium / high density	STM32100B-EVAL
STM32F103 low density/medium density	STM3210B-EVAL
STM32F103 high density/XL density, Single motor drive	STM3210E-EVAL or STEVAL-IHM022V1_SINGLEDRIVE

STM32F103 high density/XL density, Dual motor drive	STEVAL-IHM022V1_DUALDRIVE
STM32F2xx, Single motor drive	STM322xG-EVAL
STM32F2xx, Dual motor drive	STM32F2xx_dual
STM32F302xB/C, Single motor drive	STM32302C_SINGLEDRIVE
STM32F302x6/8, Single motor drive	P-NUCLEO-IHM001_SINGLEDRIVE
STM32F303xB/C, Single motor drive	STM32303C-EVAL_SINGLEDRIVE
STM32F303xB/C, Dual motor drive	STM32303C-EVAL_DUALDRIVE
STM32F4xx, Single motor drive	STM324xG-EVAL STEVAL-IHM039V1_SINGLEDRIVE
STM32F4xx, Dual motor drive	STEVAL-IHM039V1_DUALDRIVE

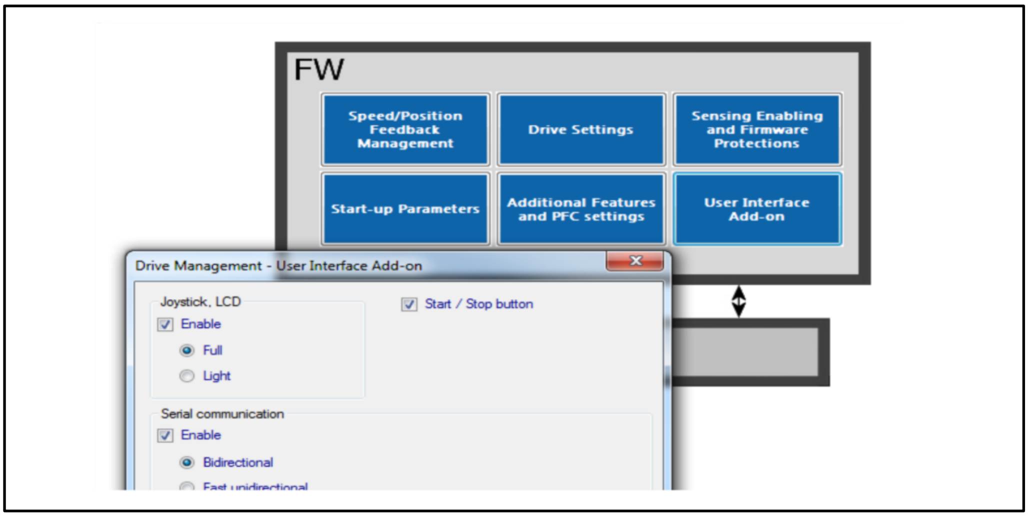
6 How to download the full LCD user interface

When an STM32 evaluation board equipped with LCD (such as STM3210B-EVAL, STM3210E-EVAL, STM32100B-EVAL, STEVAL-IHM022V1, STM322xG-EVAL, STM324xGEVAL, STEVAL-IHM039V1 or STM32303C-EVAL) is in use, you can enable the LCD plus Joystick User Interface; a useful feature of the demonstration user project that can be used as run-time command launcher, fine-tuning or monitoring tool (screens and functions are described in). This option can be selected via a setting in the ST MC Workbench GUI. See *Figure 18: "Enabling the full LCD UI in the ST MC Workbench"*.

In this case, download the LCD UI software (single or dual drive configuration) following the procedure explained below, in a reserved area in the microcontroller, located at the end of the addressable Flash memory. Unless you erase it or change the configuration from single-drive to dual-drive or vice-versa, there is no need to download it again. Even disabling the option with the GUI does not mean you need to flash it again when you re-enable the option.

The latest STM3210B-MCKIT Motor Control starter kits come with the Motor Control Library and LCD UI software (single-drive) pre-flashed. If your Motor Control kit has a previous version of Motor Control Library, you do not have the Motor Control kit but you are using one of the mentioned evaluation boards, or you are changing configuration (single-dual), you should follow one of the three procedures explained below to download the LCD UI.

Figure 18: Enabling the full LCD UI in the ST MC Workbench


FW
Speed/Position Feedback Management
Drive Settings
Sensing Enabling and Firmware Protections
Start-up Parameters
Additional Features and PFC settings
User Interface Add-on

Drive Management - User Interface Add-on

Joystick, LCD

☑ Enable

● Full

○ Light

☑ Start / Stop button

Serial communication

☑ Enable

● Bidirectional

○ Fast unidirectional

1 **Option 1** is the preferred option for its simplicity.

a. Use the STM32 ST-LINK Utility tool to download the LCD pre-compiled file opening the proper MC workspaces, as explained in *Section 5.1: "Motor control workspace"*.

b. File->Open file...

c. Select the appropriate pre-compiled file (STM3210B-EVAL.hex, STM32100B-EVAL.hex, STM3210E-EVAL.hex, STM322xG-EVAL.hex, STM324xG-EVAL.hex, STM32303C-EVAL_SINGLEDRIVE.hex, STM32303C-EVAL_DUALDRIVE.hex, STEVAL-IHM022V1_SINGLEDRIVE.hex, STEVAL-IHM022V1_DUALDRIVE.hex, STEVAL-IHM039V1_SINGLEDRIVE.hex, STEVAL-IHM039V1_DUALDRIVE.hex).

2 **Option 2**

a. Use the STM32 and STM8 Flash loader demonstrator PC software package. This is available from the ST web site (www.st.com and in the \\Installation folder\\Utilities\\Flash loader\\.). The User Manual, UM0462 (included in the package), fully explains how to operate it. For communication purposes, you need to verify that you have an available COM port (RS232) on your PC.

b. After the program is installed, run the Flash loader demonstrator application from the Programs menu, making sure that the device is connected to your PC and that the boot configuration pins are set correctly to boot from the system memory (check the evaluation board user manual).

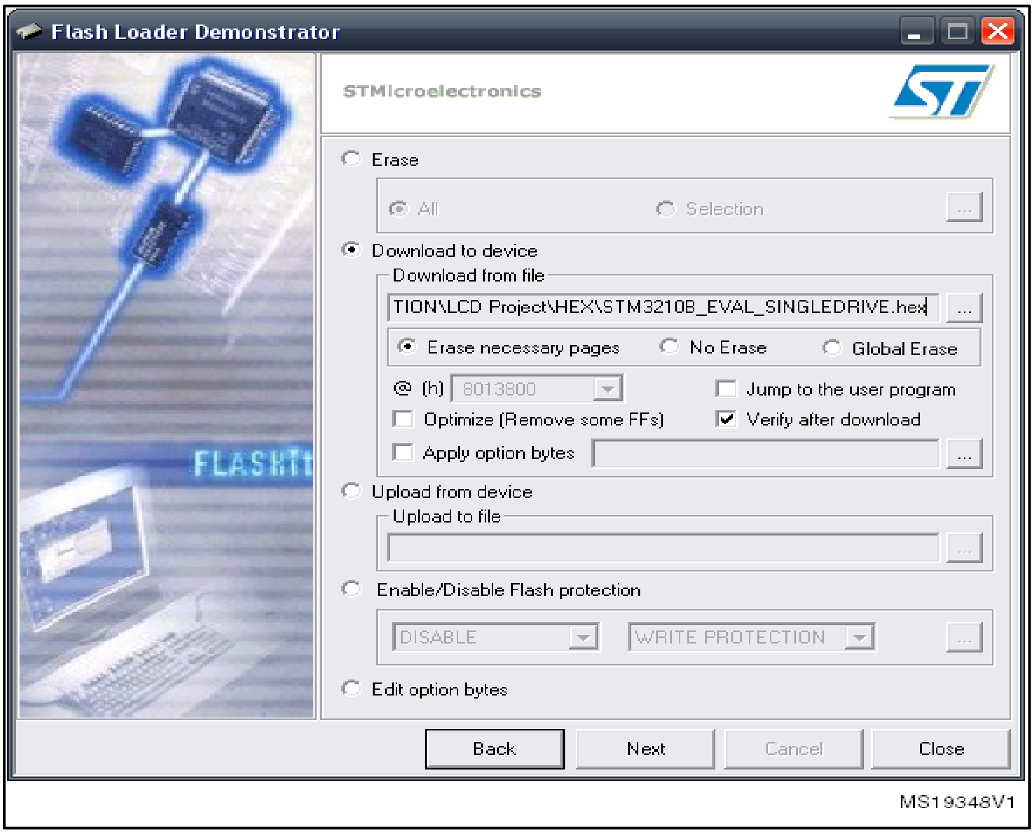
c. Reset the microcontroller to restart the system memory boot loader code.

d. When the connection is established, the wizard displays the available device information such as the target ID, the firmware version, the supported device, the memory map and the memory protection status. Select the target name in the target combo-box.

e. Click the Download to device radio button (see *Figure 19: "Flash loader wizard screen"*) and browse to select the appropriate hexadecimal file (STM3210B_EVAL.hex, STM32100B_EVAL.hex, STM3210E_EVAL.hex, STEVAL_IHM022V1_SINGLEDRIVE.hex, STEVAL_IHM022V1_DUALDRIVE.hex, STM322xG-EVAL.hex, STM324xG-EVAL.hex, STEVAL_IHM039V1_SINGLEDRIVE.hex or STEVAL_IHM039V1_DUALDRIVE.hex) from Installation folder\\LCD Project\\Hex\\.

f. Program the downloading to Flash memory. After the code is successfully flashed, setup the board to reboot from the user Flash memory and reset the microcontroller.

Figure 19: Flash loader wizard screen



○ Erase
● All ○ Selection

● Download to device

Download from file: TION\LCD Project\HEX\STM3210B_EVAL_SINGLEDRIVE.hex

● Erase necessary pages ○ No Erase ○ Global Erase

@ (h) 8013800

□ Jump to the user program

□ Optimize (Remove some FFs)

☑ Verify after download

□ Apply option bytes

○ Upload from device

Upload to file

○ Enable/Disable Flash protection

DISABLE WRITE PROTECTION

○ Edit option bytes

Back Next Cancel Close

3 **Option 3** is for users who want to modify the LCD UI code.

- Use an IDE to rebuild and download the LCD UI.
- After parameter files are generated by the GUI (to set the single/dual drive configuration) using KEIL uVision4 IDE, open the workspace located in:

```
Installation folder\\LCDProject\\MDK-ARM\\STM32F0xx_LCD Project.uvopt
Installation folder\\LCDProject\\MDK-ARM\\STM32F10x_LCD Project.uvopt
Installation folder\\LCDProject\\MDK-ARM\\STM32F2xx_LCD Project.uvopt
Installation folder\\LCDProject\\MDK-ARM\\STM32F3xx_LCD Project.uvopt
Installation folder\\LCDProject\\MDK-ARM\\STM32F4xx_LCD Project.uvopt
```

Figure 20: LCD UI project

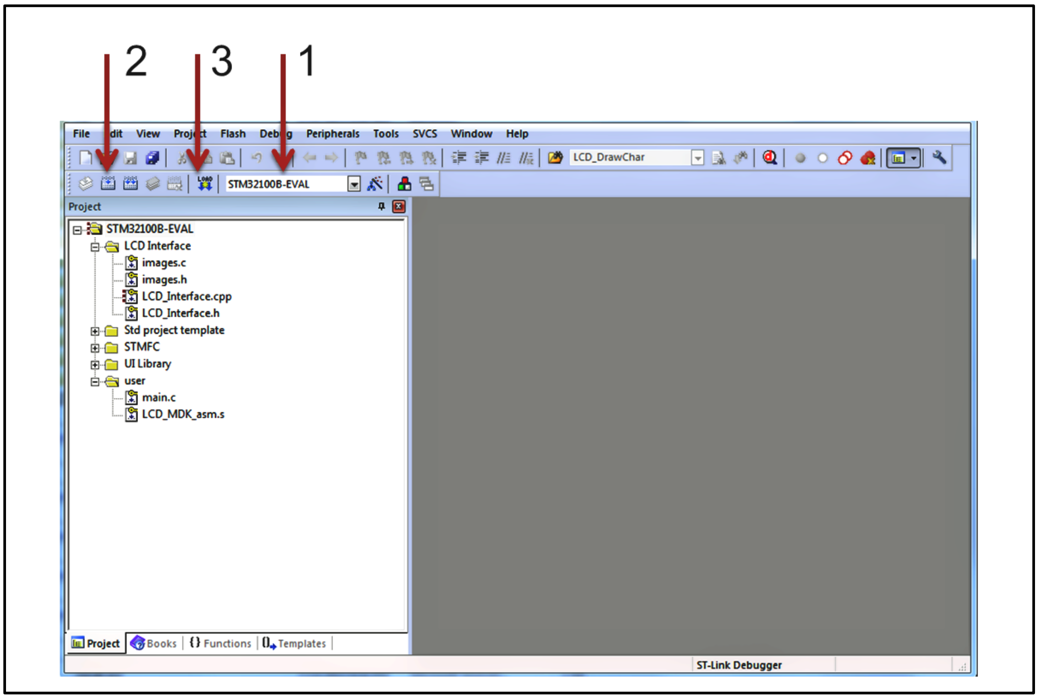


Figure 20: "LCD UI project" displays the logical arrangement of files (left-hand side) and actions that may be needed for set up and download. Five project configurations are provided (call-out 1, *Figure 20: "LCD UI project"*), one for each STM32 evaluation board that has been tested with the MC SDK:

- STM32F10B-EVAL

- STM32F10E-EVAL

- STM32F100B-EVAL

- STEVAL-IHM022V1_SINGLEDRIVE

- STEVAL-IHM022V1_DUALDRIVE

One project configuration is provided for the STM32F2xx_Workspace:p

- STM322xG-EVAL

Three project configurations are provided for the STM32F4xx_Workspace:

- STM324xG-EVAL

- STEVAL-IHM039V1_SINGLEDRIVE

- STEVAL-IHM039V1_DUALDRIVE

One project configuration is provided for the STM32F0xx_Workspace:p

- STM320518-EVAL

This configuration affects the LCD driver and linker file selection. Two project configurations are provided for the STM32F3xx_Workspace:

- STM32303C-EVAL_SINGLEDRIVE

- STM32303C-EVAL_SINGLEDRIVE

– Build the project (call-out 2, *Figure 20: "LCD UI project"*), and download it to the microcontroller memory (call-out 3, *Figure 20: "LCD UI project"*).

– To test that the LCD UI is flashed correctly: open, build and download the user project (see *Section 5.2: "MC SDK customization process"*). From the debug session, run the firmware (F5) and then, after a while, stop debugging (CTRL+Shift+D). The LCD UI is not flashed properly if the program is stalled in a trap in UITask.c, line 195.

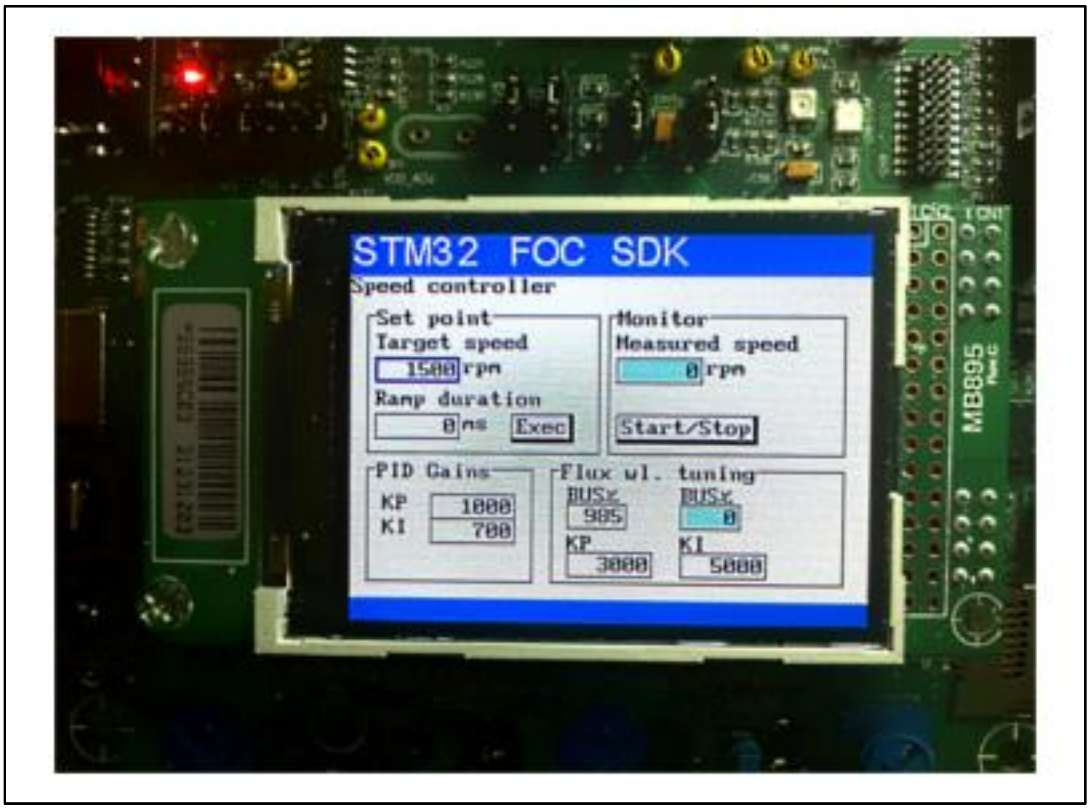
7 Full LCD user interface

7.1 Running the motor control firmware using the full LCD interface

The STM32 motor control library includes a demonstration program that enables you to display drive variables, customize the application by changing parameters, and enable and disable options in real time.

The user interface reference is the one present in the STM32 evaluation boards and is shown in *Figure 21: "User interface reference"*.

Figure 21: User interface reference



STM32 FOC SDK

Speed controller

Set point
Target speed 1500 rpm
Ramp duration 0 ms

Monitor
Measured speed 0 rpm

PID Gains
KP 1000
KI 700

Flux wl. tuning
BUSx 985
BUSx 0
KP 3000
KI 5000

The interface is composed of:

- A 320x240 pixel color LCD screen
- A joystick (see *Table 2: "Joystick actions and conventions"* for the list of joystick actions and conventions)
- A push button (KEY button)

Table 2: Joystick actions and conventions

Keyword	User action
UP	Joystick pressed up
DOWN	Joystick pressed down
LEFT	Joystick pressed to the left

Keyword	User action
RIGHT	Joystick pressed to the right
JOYSEL	Joystick pushed
KEY	Press the KEY push button

In the default firmware configuration, the LCD management is enabled. It can be disabled using the STM32 MC Workbench or disabling the feature and manually changing the line: `define #define LCD_JOYSTICK_BUTTON_FUNCTIONALITY DISABLE` (line 316) of the Drive parameters.h file.

7.2 LCD User interface structure

The demonstration program is based on circular navigation pages.

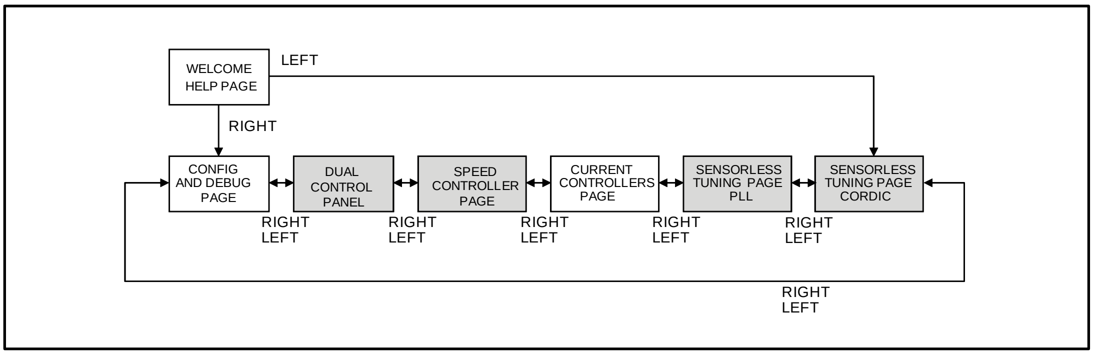
The visibility of certain pages shown in *Figure 22: "Page structure and navigation"* depends on the firmware configuration:

- Dual control panel is only present if the firmware is configured for dual motor drive.
- Speed controller page is only present when the firmware is configured in speed mode.
- Sensorless tuning page (PLL) is only present if the firmware is configured with state observer with PLL as primary or auxiliary speed sensor.
- Sensorless tuning page (CORDIC) is only present if the firmware is configured with state observer with CORDIC as primary or auxiliary speed sensor.

To navigate the help menus, use:

- RIGHT: navigate to the next page on the right
- LEFT: navigate to the next page on the left

Figure 22: Page structure and navigation

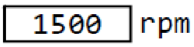
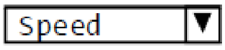
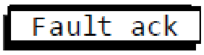


Each page is composed of a set of controls. *Table 3: "List of controls used in the LCD demonstration program"* presents the list of controls used in the LCD demonstration program. You can navigate between focusable controls in the page by pressing the joystick UP and DOWN. The focused control is highlighted with a blue rectangle. When focused, you can activate the control by pressing JOYSEL.

For some configurations such as STM32F100B-EVAL and STM320518-EVAL, a reduced set of LCD pages and/or controls is available.

Complete documentation about this LCD User Interface can be found in *User manual* STM32F PMSM single/dual FOC SDK (UM1052).

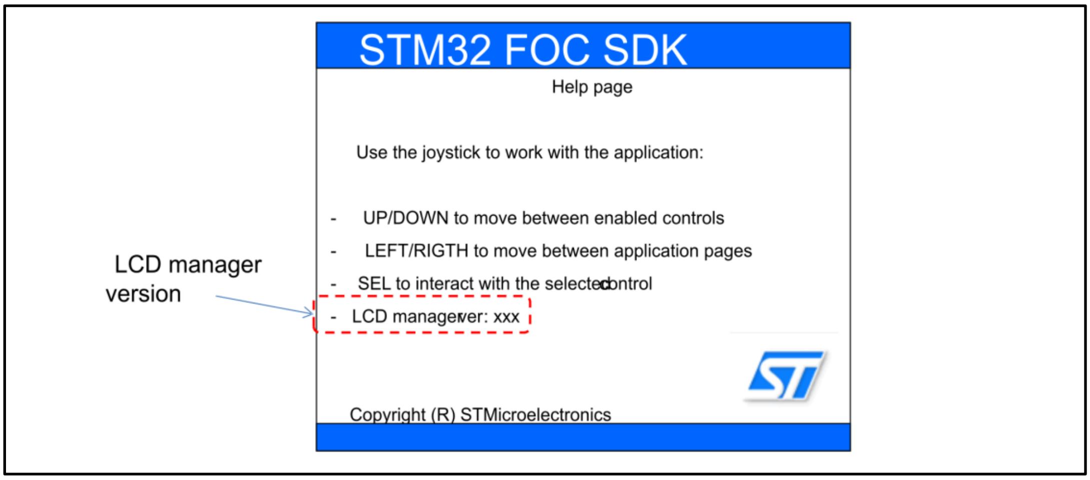
Table 3: List of controls used in the LCD demonstration program

Control name and examples	Description
Edit box  	Manages a numerical value. It can be "read only" or "read/write". A read only edit box has a gray background and cannot be focused. A read/write edit box has a white background and can be focused. When a read/write edit box is focused, it can be activated for modification by pressing JOYSEL. An activated read/write edit box has a green background and its value can be modified pressing and/or keeping joystick UP/DOWN pressed. The new value is set to the motor control-related object instantaneously when the value changes, unless otherwise mentioned in this manual.
Combo-box  	Manages a list of predefined values. For example, Speed or Torque control mode. When focused, it can be activated for modification by pressing JOYSEL. An activated combo-box has a green background and its value can be modified by pressing the joystick UP/DOWN. When the value changes, the new value is instantaneously set to the motor control-related object, unless otherwise mentioned in this manual.
Button  	Sends commands. For example, a start/stop button. A disabled button is drawn in light gray and cannot be focused. An enabled button is painted in black and can be focused. When focused, pressing JOYSEL corresponds to "pushing" the button and sending the related command.

7.2.1 Welcome message

After the STM32 evaluation board is powered on or reset, a welcome message appears on the LCD with loaded firmware code and version information.

Figure 23: STM32 Motor Control demonstration project welcome message



STM32 FOC SDK

Help page

Use the joystick to work with the application:

- UP/DOWN to move between enabled controls
- LEFT/RIGTH to move between application pages
- SEL to interact with the selected control
- LCD manager ver: xxx

Copyright (R) STMicroelectronics

7.2.2 Configuration and debug page

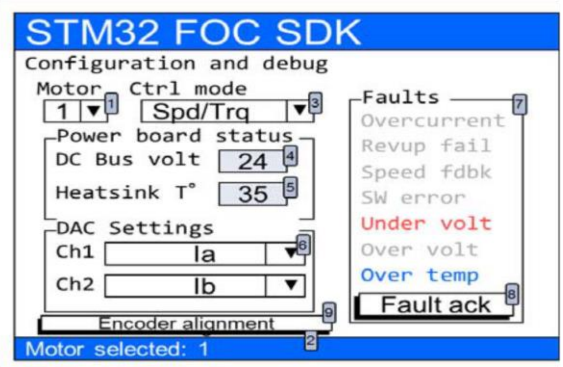
Press the joystick RIGHT from the welcome page to enter the Configuration and debug page.

To navigate between focusable controls on the page, press the joystick UP/DOWN.

Use the Configuration and debug page shown in *Figure 24: "Configuration and debug page"* to:

- Select the active motor drive (1). This control is present only for dual motor control applications. This combo-box enables you to select the active motor drive. Once the active motor is selected, it is shown in the status bar present at the bottom of the screen (2). Commands performed on, or feedback from a control are only relative to the active motor.
- Select the control mode (3). Two control modes are available: speed and torque. You can change the control mode from speed to torque and vice versa on-the-fly even if the motor is already running.

Figure 24: Configuration and debug page



STM32 FOC SDK

Configuration and debug

Motor: 1

Ctrl mode: Spd/Trq

Power board status

DC Bus volt: 24

Heatsink T°: 35

DAC Settings

Ch1: Ia

Ch2: Ib

Faults

Overcurrent

Revup fail

Speed fdbk

SW error

Under volt

Over volt

Over temp

Motor selected: 1

- Read the DC bus voltage value (4). This control is read-only.
- Read the heat sink temperature value (5). This control is read-only.
- Select the variables to be put in output through DAC channels (6). These controls are present only if the DAC option is enabled in the firmware. The list of variables also depends on firmware settings.
- It is possible to read the list of fault causes (7) if fault conditions have occurred, or if they are still present. The list of possible faults is summarized in *Table 4: "Fault conditions list"*. If a fault condition occurred and is over, the relative label is displayed in blue. If a fault condition is still present, the relative label is displayed in red. It is gray if there is no error.
- To acknowledge the fault condition, press the **Fault ack** button (8). If a fault condition occurs, the motor is stopped and it is no longer possible to navigate in the other pages. In this condition, it is not possible to restart the motor until the fault condition is over and the occurred faults have been acknowledged by the user, pushing the **Fault ack** button (8). If a fault condition is running, the **Fault ack** button is disabled.

Table 4: Fault conditions list

Fault	Description
Overcurrent	This fault occurs when the microcontroller break input signal is activated. It is usually used to indicate a hardware over current condition.
Revup fail	This fault occurs when the programmed rev-up sequence ends without validating the speed sensor information. The rev-up sequence is performed only when the state observer is configured as primary speed sensor.
Speed fdbk	This fault occurs only in RUN state when the sensor no longer meets the conditions of reliability.
SW error	This fault occurs when the software detects a general fault condition. In the present implementation, the software error is raised when the FOC frequency is too high to be sustainable by the microcontroller.
Under volt	This fault occurs when the DC bus voltage is below the configured threshold.
Over volt	This fault occurs when the DC bus voltage is above the configured threshold. If the dissipative brake resistor management is enabled, this fault is not raised.
Over temp	This fault occurs when the heat sink temperature is above the configured threshold.

- Execute encoder initialization. If the firmware is configured to use the encoder as the primary speed sensor or auxiliary speed sensor, the **Encoder alignment** button (9) is also present. In this case, the alignment of the encoder is required only once after each reset of the microcontroller.

7.2.3 Dual control panel page

This page is present only if the firmware is configured for dual motor drive.

To enter the Dual control panel page, press the joystick RIGHT from the Configuration and debug page.

It is possible to navigate between focusable controls present in the page by pressing the joystick UP/DOWN.

The Dual control panel page shown in *Figure 25: "Dual control panel page"* is used to send commands and get feedback from both motors. It is divided into three groups:

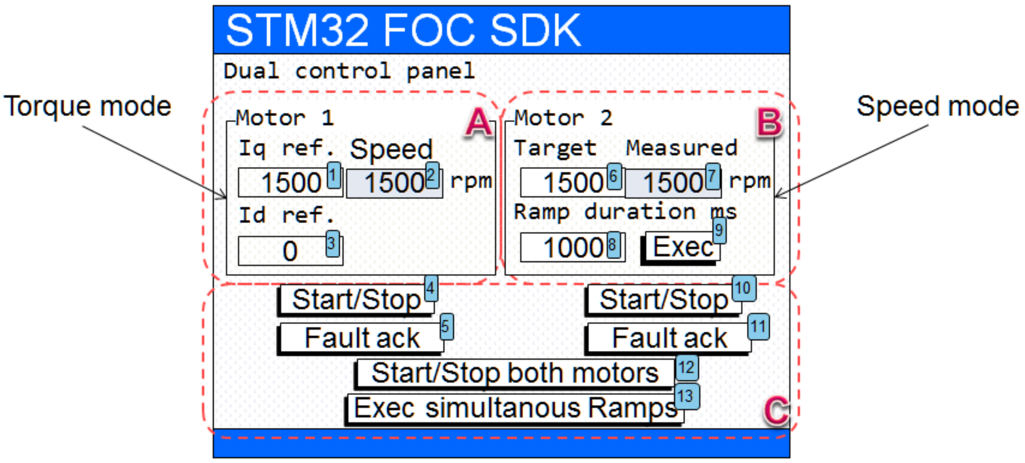
- Groups A and B depend on speed/torque settings. The group content is updated on-the-fly when the control mode (torque/speed) is changed in the Configuration and debug page. The control present in group A is related to the first motor. The control present in group B is related to the second motor.
- Group C does not depend on speed/torque settings. The control present in this group is related to both motors.

Figure 25: "Dual control panel page" shows an example in which the first motor is set in torque mode and the second motor is set in speed mode.

The controls present in this page are used as follows:

- To set the I_q reference (1). This is related to motor 1 and is only present if motor 1 is set in torque mode. I_q reference is expressed in s16A. In this page, the current references are always expressed as Cartesian coordinates (I_q,I_d).

Figure 25: Dual control panel page



- To set the I_d reference (3). This is related to motor 1. This control is only present if motor 1 is set in torque mode. Id reference is expressed in s16A. In this page, the current references are always expressed as Cartesian coordinates (I_q,I_d).

To convert current expressed in amps to current expressed in digits, use the following formula:

Current(s16A) = [Current(Amp) * 65536 * Rshunt * Aop] / Vdd micro.

- Set the final motor speed of a speed ramp (6). This is related to motor 2. This control is only present if motor 2 is set in speed mode. Motor speed is expressed in RPM. The value set in this control is not automatically sent to the motor control related object but it is used to perform a speed ramp execution. See the Exec button description (9).
- Set the duration of a speed ramp (8). This is related to motor 2. This control is only present if motor 2 is set in speed mode. The duration is expressed in milliseconds. The value set in this control is not automatically sent to the motor control related object, but it is used to perform a speed ramp execution. See the Exec button description (9). It is possible to set a duration value of 0 to program a ramp with an instantaneous change in the speed reference from the current speed to the final motor speed (6).
- Execute a speed ramp by pushing the “Exec” button (9). This is related to motor 2. This control is only present if motor 2 is set in speed mode. The Exec speed ramp command is sent to the motor control related object together with the final motor speed and duration currently selected (6). The Exec speed ramp command performs a speed ramp from the current speed to the final motor speed in a time defined by the duration. The command is buffered and takes effect only when the motor is in RUN state.
- To read the motor speed, respectively (2) and (7) for motor 1 and motor 2. The motor speed is expressed in RPM. This control is read-only.
- Send a start/stop command, (4) for motor 1, (10) for motor 2. This is performed by pushing the start/stop button. A start/stop command means: start the motor if it is stopped, or stop the motor if it is running. If the drive is configured in speed mode when the motor starts, a speed ramp with the latest values of the final motor speed

and duration is performed. If a fault condition occurs at any time, the motor is stopped (if running) and the start/stop button is disabled.

- When a fault condition is over, the **Fault ack** button, (5) for motor 1, (11) for motor 2, is enabled. Pushing this button acknowledges the fault conditions that have occurred. After the fault is acknowledged, the start/stop button becomes available again. When a fault occurs and before it is acknowledged, it is only possible to navigate in the Dual control panel page and the Configuration and debug page.
- To start or stop both motors simultaneously, push the **Start/Stop both motors** button (12). This button is enabled only when the motors are both in Idle state or both in RUN state. If any of the motors is configured in speed mode when it starts, a speed ramp with the last values of the final motor speed and duration is performed. It is possible to stop both motors at any time by pushing the KEY button.
- To execute simultaneous speed ramps on both motors, push the **Exec simultaneous** Ramps button (13). This button is disabled when at least one of the two motors is configured in torque mode. The Exec speed ramp command is sent to both motor control objects together with the related final motor speed and the duration currently selected. The Exec speed ramp command performs a speed ramp from the current speed to the final motor speed in a time defined by the duration for each motor. The commands are buffered and take effect only when the related motor is in RUN state.

7.2.4 Speed controller page

This page is only present if the control mode set in is the speed mode.

To enter the Speed controller page, press the joystick RIGHT from the Configuration and debug page (or from the Dual control panel page, if the firmware is configured in dual motor drive).

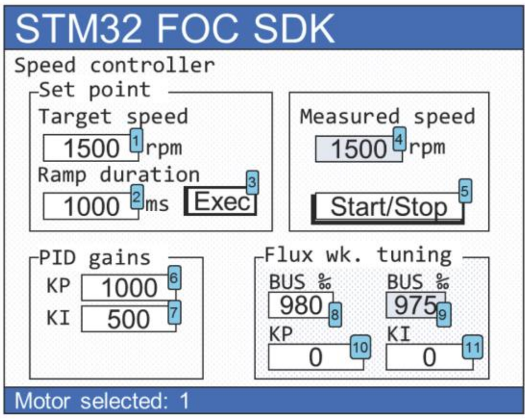
It is possible to navigate between focusable controls present in the page by pressing the joystick UP/DOWN.

The Speed controller page shown in *Figure 26: "Speed controller page"* is used to send commands and get feedback related to the speed controller from the active motor. There are four groups of controls in this page:

Table 5: Control groups

Control group	Description
Set point	Used to configure and execute a speed ramp
PID gains	Used to change the speed controller gains in real- time
Flux wk. tuning	Used to tune the flux weakening related variables
Measured speed with start/stop button	Composed of two controls that are also present in the Current controllers page and in the sensorless tuning page; it provides a fast access to the measured speed and to the motor start/stop function

Figure 26: Speed controller page



STM32 FOC SDK

Speed controller

Set point

Target speed 1500 rpm

Ramp duration 1000 ms

Measured speed 1500 rpm

PID gains

KP 1000

KI 500

Flux wk. tuning

BUS % 980

BUS % 975

KP 0

KI 0

Motor selected: 1

If the firmware is configured as dual motor drive, it is possible to know which motor is active by reading the label at the bottom of the page. To change the active motor, go to the Configuration and debug page and change (1) in *Figure 26: "Speed controller page"*.

Table 6: "Speed controller page controls " lists the actions that can be performed using this page.

Table 6: Speed controller page controls

Control	Description
Target speed (1 in *Figure 26: "Speed controller page"*)	Sets the final motor speed of a speed ramp for the active motor. The motor speed is expressed in RPM. The value set in this control is not automatically sent to the motor control related object, but it is used to perform a speed ramp execution. See the **Exec** button description (3)
Ramp duration (2)	Sets the duration of a speed ramp for the active motor. The duration is expressed in milliseconds. The value set in this control is not automatically sent to the motor control related object, but it is used to perform a speed ramp execution. See the **Exec** button description (3). It is possible to set a duration value of 0 to program a ramp with an instantaneous change in the speed reference from the current speed to the final motor speed (1).
Exec button (3)	Executes a speed ramp for the active motor. The execute speed ramp command is sent to the motor control related object together with the final motor speed and duration presently selected (1) and (2). The execute speed ramp command performs a speed ramp from the current speed to the final motor speed in a time defined by duration. The command is buffered and takes effect only when the motor becomes in RUN state.
Measured speed (4)	Reads the motor speed for the active motor. The motor speed is expressed in RPM and is a read-only control.
Start/Stop button (5)	Sends a start/stop command for the active motor. A start/stop command starts the motor if it is stopped, or stops a running motor. Used with a motor start, a speed ramp with the last values of the final motor speed and duration is performed. If a fault condition occurs at any time, the motor is stopped (if running) and the Configuration and debug page displays.
Speed PID gain KP (6)	Sets the proportional coefficient of the speed controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the speed controller.

Control	Description
Speed PID gain KI (7)	Sets the integral coefficient of the speed controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the speed controller.
Bus‰ (8)	The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of flux weakening controller. The value is expressed in per mil (‰) of DC bus voltage.
Bus‰ (9)	DC bus voltage percentage presently used for the active motor; it is a read-only control. This control is present only if the flux weakening feature is enabled in the firmware. The value is actually expressed in per mil (‰) of DC bus voltage.
Flux wk PI gain KP (10)	The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the flux weakening controller.
Flux wk PI gain KI (11)	The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the flux weakening controller.

7.2.5 Current controllers page

To enter the Current controllers page, press the joystick RIGHT from the Speed controller page (or from one of the pages described above if the Speed controller page is not visible).

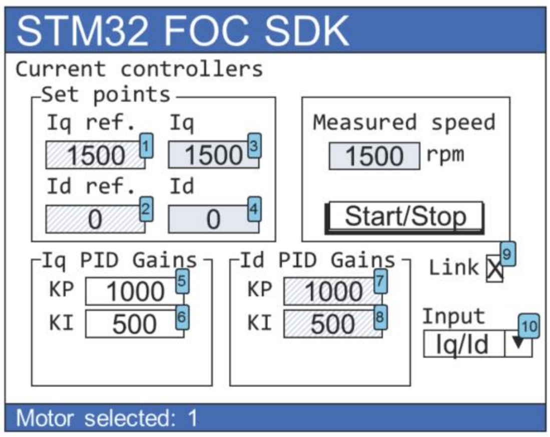
It is possible to navigate between focusable controls present in the page by pressing the joystick UP/DOWN.

The Current controllers page shown in *Figure 27: "Current controllers page"* is used to send commands and get feedback related to current controllers, from the active motor. There are five control groups in this page:

Table 7: Control groups

Control group	Description
Set point	Used to set the current references and read the measured currents
Iq PID gains	Used to change the speed controller gains in real time
Id PID gains	
Measured speed with start/stop button	Composed of two controls that are also present in the Current controllers page and in the Sensorless tuning page. It provides a fast access to the measured speed and to the motor start/stop function

Figure 27: Current controllers page



STM32 FOC SDK

Current controllers

Set points

Iq ref. 1500

Iq 1500

Id ref. 0

Id 0

Measured speed 1500 rpm

Iq PID Gains

KP 1000

KI 500

Id PID Gains

KP 1000

KI 500

Link ☒

Input Iq/Id

Motor selected: 1

If the firmware is configured as dual motor drive, it is possible to know which motor is active by reading the label at the bottom of the page. To change the active motor, go to the Configuration and debug page and change (1) in *Figure 27: "Current controllers page"*.

Table 8: "Current controllers page controls " lists the actions that can be performed using this page.

Table 8: Current controllers page controls

Control	Description
I_q reference (1 in *Figure 27: "Current controllers page"*)	Sets and reads the I_q reference for the active motor. This control is read-only if the active motor is set in speed mode, otherwise it can be modified. The I_q reference is expressed in s16A. To convert current expressed in Amps to current expressed in digits, use the formula: Current(s16A) = [Current(Amp) * 65536 * Rshunt * Aop] / Vdd micro
I_d reference (2)	Sets and reads the I_d reference for the active motor. This control is usually read-only if the active motor is set in speed mode, otherwise it can be modified. The I_d reference is expressed in s16A.
Measured I_q (3)	Reads the measured I_q for the active motor. Measured I_q is expressed in s16A and is a read-only control.
Iq PI(D) gain, KP (5)	Sets the proportional coefficient of the I_q current controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the current controller.
Iq PI(D) gain, KI (6)	Sets the integral coefficient of the Iq current controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the current controller.
Id PI(D) gain, KP (7)	Sets the proportional coefficient of the I_d current controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the current controller. This control is only read if the link check box is checked.
Id PI(D) gain, KI (8)	Sets the integral coefficient of the I_d current controller for the active motor. The value set in this control is automatically sent to the motor control related object, allowing the run-time tuning of the current controller. This control is only read if the link check box is checked.

8 Introduction to the PMSM FOC drive

This software library is designed to achieve the high dynamic performance in AC permanent-magnet synchronous motor (PMSM) control offered by the well-established field oriented control (FOC) strategy.

With this approach, it can be stated that, by controlling the two currents i_{qs} and i_{ds}, which are mathematical transformations of the stator currents, it is possible to offer electromagnetic torque (T_e) regulation and, to some extent, flux weakening capability.

This resembles the favorable condition of a DC motor, where those roles are held by the armature and field currents.

Figure 28: Basic FOC algorithm structure, torque control

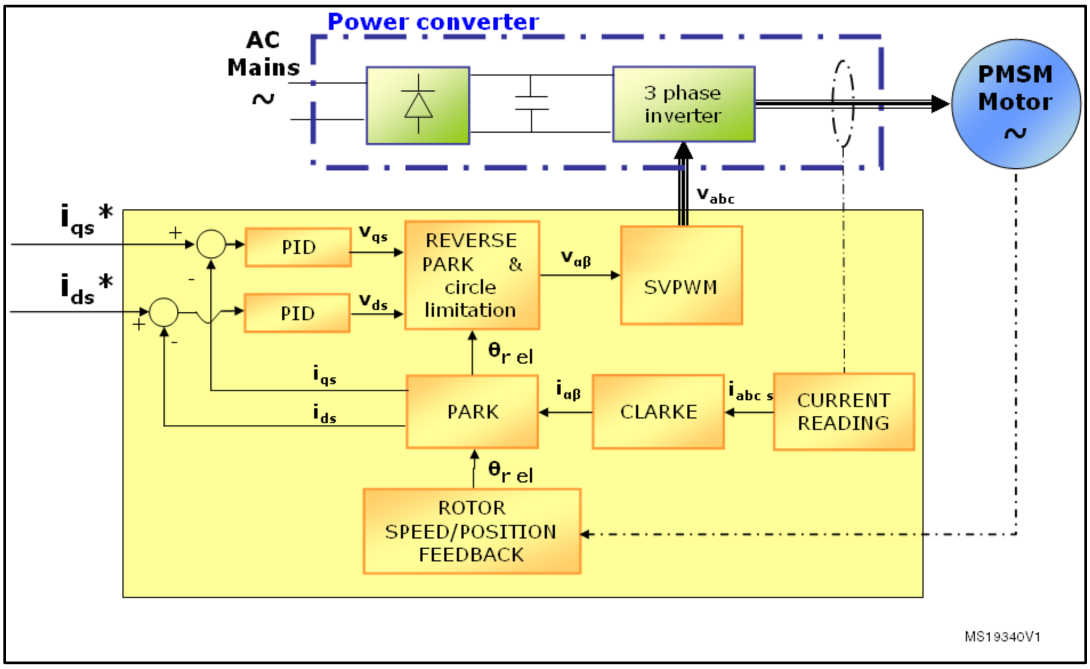
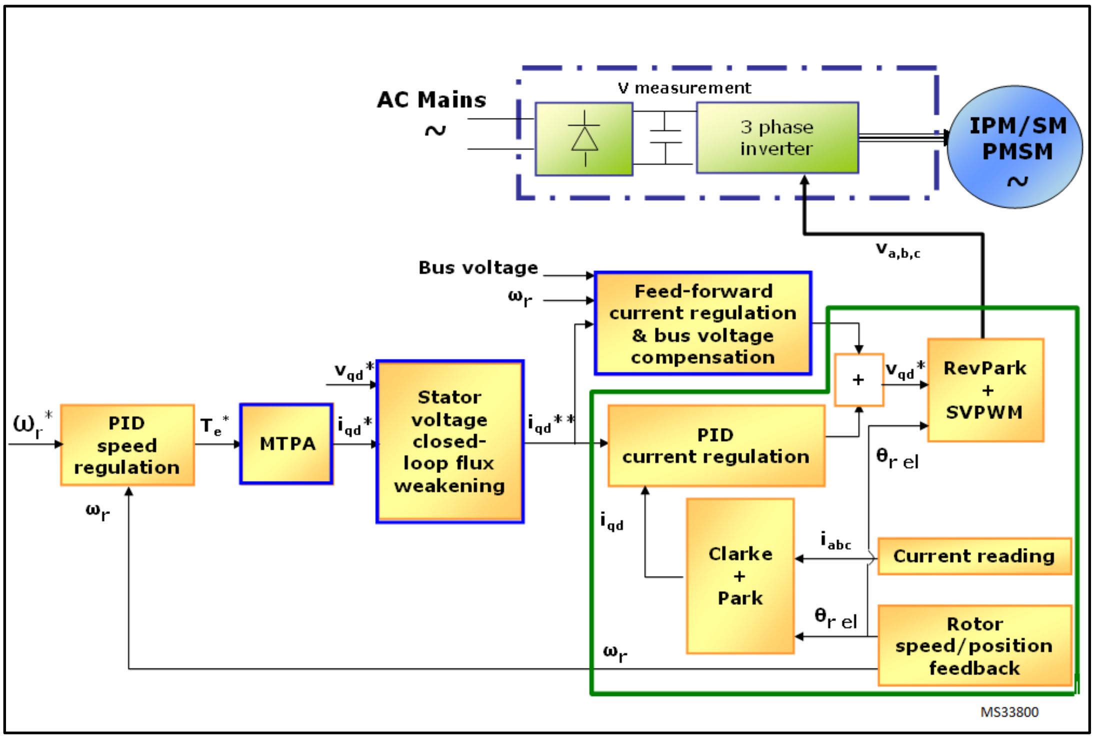


Figure 29: Speed control loop



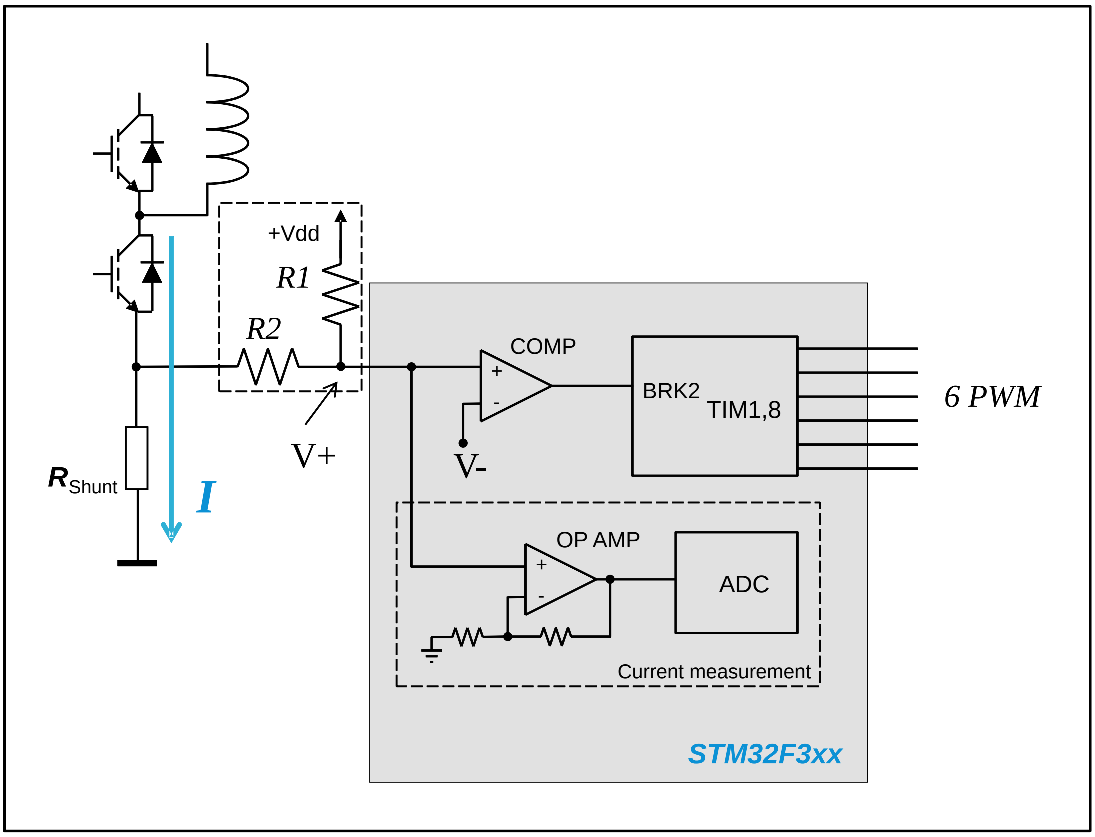
9 Current sensing and protection with embedded analog (STM32F3x)

9.1 Introduction

The STM32F302/303 microcontrollers feature an enhanced set of peripherals. They include comparators, PGAs, DACs and high-speed ADCs. This section describes how to use these peripherals according to what is made available by the MC library.

Figure 30: "Current sensing and over-current protection with STM32F302/303" shows a current sensing and over-current protection network that can be implemented using the internal resources of the STM32F302/303: due to the motor phase current, the voltage drop on the shunt resistor can be either positive or negative, an offset is set by R1 and R2. Then, the signal is linked to a microcontroller pin that has both functionality of amplifier and comparator non-inverting input.

Figure 30: Current sensing and over-current protection with STM32F302/303



This configuration shows the optimization that can be reached, using STM32F3, both in terms of reduced number of external component and microcontroller pins assigned for the MC application.

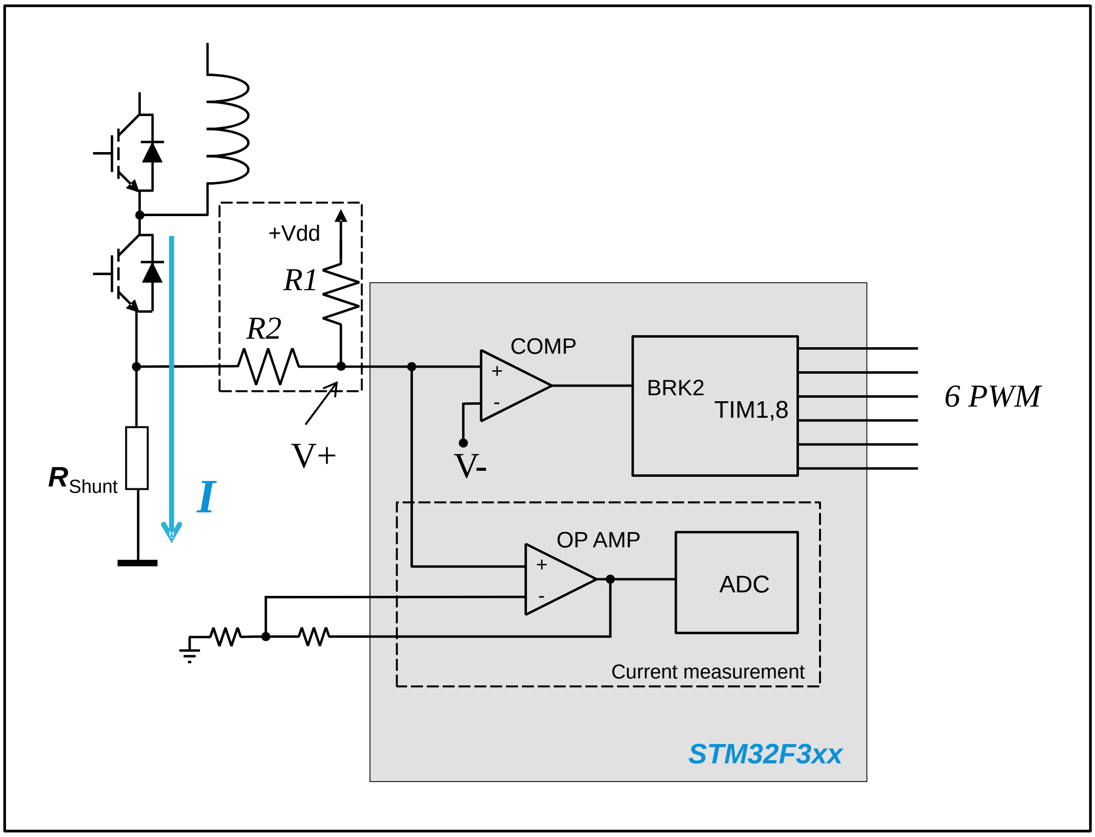
9.2 Current sensing

To maximize the resolution of the measurement, the embedded Programmable Gains Amplifier (PGA) can be used to adapt the level of voltage drop in the shunt resistor (Rshunt), caused by the motor current, up to the maximum range allowed by the analog to digital converter (ADC).

The PGA has a set of fixed internal gains (x2, x4, x8, x16). An alternative option in PGA mode allows you to route the central point of the resistive network on one of the I/Os connected to the non-inverting input: this feature can be used for instance to add a low-pass filter to PGA, as shown in *Figure 32: "Current sensing network using internal gains plus filtering capacitor"*

On the other hand, if a different value of amplification is required, it is possible to completely define the amplification network (for instance as it's shown in *Figure 31: "Current sensing network using external gains"*).

Figure 31: Current sensing network using external gains



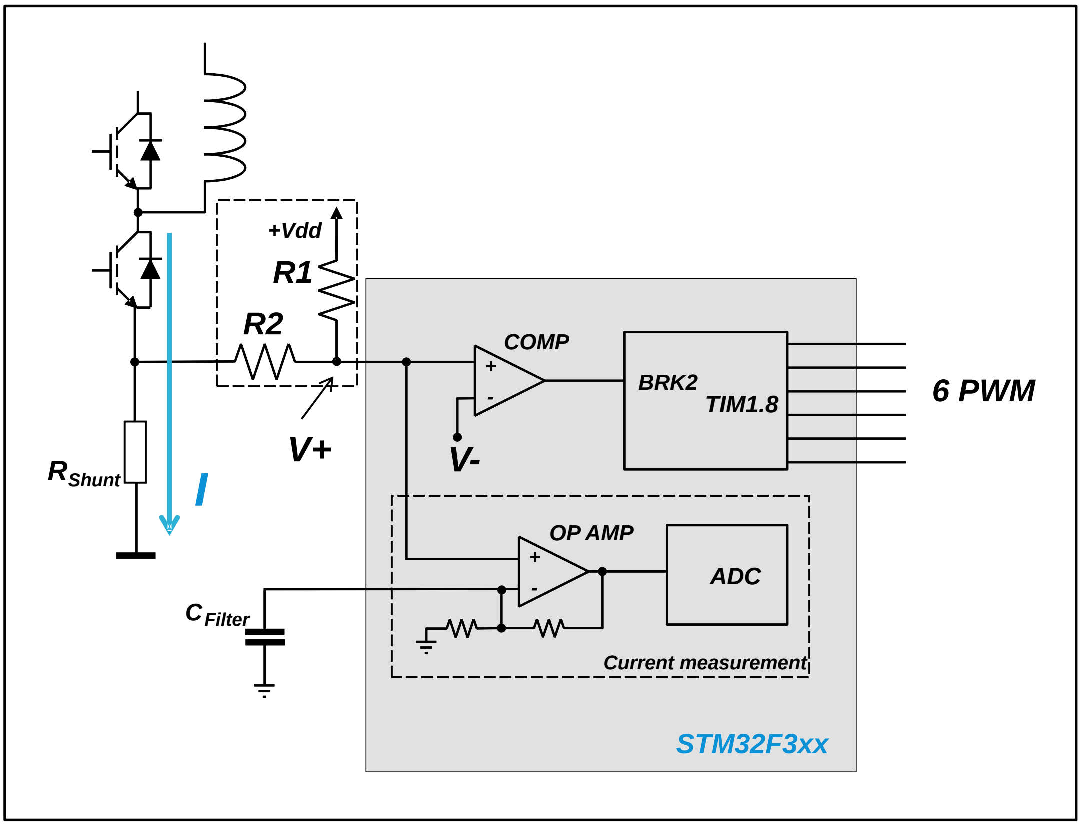
The MC library can be arranged to match all the configurations shown: by using the ST MC Workbench, creating a project based on STM32F302 or STM32F303, from the dialogue window located in Control Stage -> Analog Input -> Phase current feedback (*Figure 33: "STMCWB window related to PGA/COMP settings for motor currents"*) setting:

- “Embedded PGA” as current sensing topology;
- PGA internal gain (like in *Figure 30: "Current sensing and over-current protection with STM32F302/303"*): Settling “Internal” in the “OPAMP Gain” drop down list;

- PGA external gain (like in *Figure 31: "Current sensing network using external gains"*): Settling "External" in the "OPAMP Gain" drop down list;
- PGA internal gain with external filtering capacitor (like in *Figure 32: "Current sensing network using internal gains plus filtering capacitor"*): Settling "Internal" in the "OPAMP Gain" drop down list and checking the "Feedback net filtering" check box in the same group.

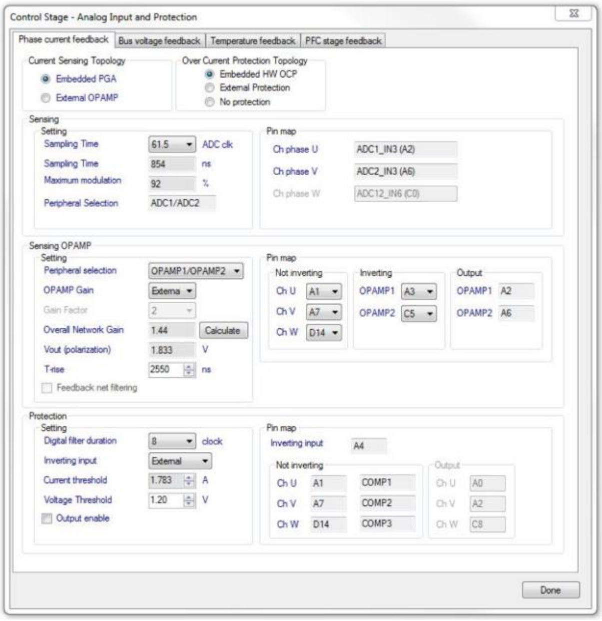
Only one of these settings is present in the workbench for each drives, since the configuration applies to each shunt resistor conditioning network.

Figure 32: Current sensing network using internal gains plus filtering capacitor



On the other hand, it is possible to setup the motor current measurement network to use external operational amplifiers. In this case the amplified signals are directly fed to the ADC channels. By using the ST MC Workbench, creating a project based on STM32F302 or STM32F303, from the dialogue window located in Control Stage -> Analog Input -> Phase current feedback, setting "External OPAMP" as current sensing topology.

Figure 33: STMCWB window related to PGA/COMP settings for motor currents



Current Sensing Topology
● Embedded PGA
○ External OPAMP

Over Current Protection Topology
● Embedded HW OCP
○ External Protection
○ No protection

Sensing

Setting
Sampling Time 61.5 ADC clk
Sampling Time 854 ns
Maximum modulation 92 %
Peripheral Selection ADC1/ADC2

Pin map
Ch phase U ADC1_IN3 (A2)
Ch phase V ADC2_IN3 (A6)
Ch phase W ADC12_IN6 (C0)

Sensing OPAMP

Setting
Peripheral selection OPAMP1/OPAMP2
OPAMP Gain Externa
Gain Factor 2
Overall Network Gain 1.44
Vout (polarization) 1.833 V
T-rise 2550 ns
□ Feedback net filtering

Pin map
Not inverting: Ch U A1; Ch V A7; Ch W D14
Inverting: OPAMP1 A3; OPAMP2 C5
Output: OPAMP1 A2; OPAMP2 A6

Protection

Setting
Digital filter duration 8 clock
Inverting input External
Current threshold 1.783 A
Voltage Threshold 1.20 V
□ Output enable

Pin map
Inverting input A4
Not inverting: Ch U A1 COMP1; Ch V A7 COMP2; Ch W D14 COMP3
Output: Ch U A0; Ch V A2; Ch W C8

9.3 Overcurrent protection

The basic principle of the hardware over-current protection mechanism can be summarized as follows:

- The phase current of the motor flows in the power transistor of the inverter bridge and passes through the shunt resistor (RShunt) producing a voltage drop (V+).
- This voltage drop is compared with a threshold (V-) defining the maximum admissible current.
- If the threshold is exceeded, a break signal stops the PWM generation putting the system in a safe state.

All of these actions can be performed using the internal resources of the STM32F302/303 and, in particular, the embedded comparators and the advanced timer break function (BRK2). As shown in *Figure 30: "Current sensing and over-current protection with STM32F302/303"*, *Figure 31: "Current sensing network using external gains"*, and *Figure 32: "Current sensing network using internal gains plus filtering capacitor"* the same signal is fed to both not inverting input of embedded comparators and PGA.

The overcurrent threshold (V-) can be defined in three different ways:

- using one of the available internal voltage reference (1.2V, 0.9V, 0.6V, 0.3V);
- providing it externally using the inverting input pin of the comparator;
- programming a DAC channel.

The MC library can be arranged to match all the configurations shown by using the ST MC Workbench, creating a project based on STM32F302 or STM32F303, from the dialogue window located in Control Stage -> Analog Input -> Phase current feedback (*Figure 33: "STMCWB window related to PGA/COMP settings for motor currents"*), setting:

- “Embedded HW OCP” radio button as overcurrent protection topology;

- HW OCP internal threshold: selecting “Internal” in the “Inverting input” drop down list and choosing the internal voltage reference (among available values) in “Voltage Threshold”.
- HW OCP external threshold: selecting “External” in the “Inverting input” drop down list and editing the external voltage reference in “Voltage Threshold”.
- HW OCP internal threshold using DAC: selecting “DAC” in the “Inverting input” drop down list and editing the DAC voltage reference to be generated in “Voltage Threshold”. A DAC channel must be assigned for this functionality (OCP) from the related dialogue window located in Control stage -> DAC functionality (*Figure 36: "STMCWB window related to DAC settings"*).

On the other hand, it is possible to setup the motor overcurrent protection network to use external components. In this case the overcurrent protection signal - coming from a comparator for instance - is directly fed to the advanced-timer's BKIN2 pin. By using the ST MC Workbench, creating a project based on STM32F302 or STM32F303, from the dialogue window located in Control Stage -> Analog Input -> Phase current feedback, setting “External protection” as OCP protection topology.

In any case, either using embedded comparators or external components, a digital filter, upstream the BKIN2 function, can be enabled and configured to reject noises.

9.4 Resource allocation - single drive

For project based on STM32F302/303 the current feedback network configurations supported by STM32 FOC SDK are single shunt and three shunt.

Single shunt topology: according to configuration (as explained in *Section 9.2: "Current sensing"* and *Section 9.3: "Overcurrent protection"*), one ADC, one OPAMP, one comparator, and one DAC channel must be assigned.

- If “Embedded PGA” is enabled, the selection of ADC peripheral (and input pin) is linked to that of PGA peripheral.
- If “Embedded HW OCP” and “Embedded PGA” are enabled, the selection of ADC and comparator (and their input and ‘+’ pins) is linked to that of PGA peripheral (and its '+' input).
- If “Embedded HW OCP” is enabled and “Embedded PGA” is disabled, the selection of comparator is free.
- If “Embedded HW OCP” and “Embedded PGA” are both disabled, the selection of comparator and ADC is free.
- If both PGA and comparator for OC protection are used they will share the same input pins for the motor current measurement signal.

Three shunt topology: according to configuration (as explained in *Section 9.2: "Current sensing"* and *Section 9.3: "Overcurrent protection"*), two ADCs, two OPAMPs, three comparators, and one DAC channel must be assigned.

- If “Embedded PGA” is enabled, the selection of ADC peripherals (and input pins) is linked to that of PGA peripherals.
- If “Embedded HW OCP” and “Embedded PGA” are enabled, the selection of ADCs and comparators (and their inputs and ‘+’ pins) is linked to that of PGA peripherals (and theirs '+' inputs).
- If “Embedded HW OCP” is enabled and “Embedded PGA” is disabled, the selection of comparators is free.
- If “Embedded HW OCP” and “Embedded PGA” are both disabled, the selection of comparators and ADCs is free.

The pair OPAMP1/OPAMP2 can be used in a project based on STM32F302 or STM32F303; the pair OPAMP3/OPAMP4 can be used additionally in a project based on STM32F303.

The pair ADC1/ADC2 can be used in a project based on STM32F302 or STM32F303; the pair ADC3/ADC4 can be used additionally in a project based on STM32F303.

If both PGA and comparator for OC protection are used they will share the same input pins for the motor current measurement signal.

9.5 Resource allocation - dual drive

If using a STM32F303 microcontroller is possible to create a dual drive project. The current feedback network configurations supported by STM32 FOC SDK are single shunt and three shunt.

Dual single shunt drive, dual three shunts drive and mixed “single plus three” shunts drives are allowed.

The sharing of peripherals between “single shunt drive” and “three shunt drive” is not allowed.

The sharing of peripherals between two “single shunt drive” is not allowed.

The sharing of peripherals between two “three shunt drive” is allowed, in the forms expressed below.

Single shunt topology: for each motor, according to configuration (as explained *Section 9.2: "Current sensing"* and *Section 9.3: "Overcurrent protection"*), one ADC, one OPAMP and one Comparator must be assigned.

- If “Embedded PGA” is enabled, the selection of ADC peripheral (and input pin) is linked to that of PGA peripheral.
- If “Embedded HW OCP” and “Embedded PGA” are enabled, the selection of ADC and comparator (and their input and ‘+’ pins) is linked to that of PGA peripheral (and its '+' input).
- If “Embedded HW OCP” is enabled and “Embedded PGA” is disabled, the selection of comparator is free.
- If “Embedded HW OCP” and “Embedded PGA” are both disabled, the selection of comparator and ADC is free.
- If both PGA and comparator for OC protection are used they will share the same input pins for the motor current measurement signal.

Three shunt topology mixed with Single shunt topology: according to configuration (as explained in *Section 9.2: "Current sensing"* and *Section 9.3: "Overcurrent protection"*), two ADCs, two OPAMPs, three Comparators, and one DAC channel must be assigned.

- If “Embedded PGA” is enabled, the selection of ADC peripherals (and input pins) is linked to that of PGA peripherals.
- If “Embedded HW OCP” and “Embedded PGA” are enabled, the selection of ADCs and comparators (and their inputs and ‘+’ pins) is linked to that of PGA peripherals (and theirs '+' inputs).
- If “Embedded HW OCP” is enabled and “Embedded PGA” is disabled, the selection of comparators is free.
- If “Embedded HW OCP” and “Embedded PGA” are both disabled, the selection of comparators and ADCs is free.
- The pair OPAMP1/OPAMP2 can be used in a project based on STM32F302 or STM32F303; the pair OPAMP3/OPAMP4 can be used additionally in a project based on STM32F303.

- The pair ADC1/ADC2 can be used in a project based on STM32F302 or STM32F303; the pair ADC3/ADC4 can be used additionally in a project based on STM32F303.
- If both PGA and comparator for OC protection are used they will share the same input pins for the motor current measurement signal.

Dual Three shunt topology, not shared resources: according to configuration (as explained in *Section 9.2: "Current sensing"* and *Section 9.3: "Overcurrent protection"*), four ADCs, four OPAMPs, six comparators, and two DAC channels must be assigned.

- If “Embedded PGA” is enabled, the selection of ADC peripherals (and input pins) is linked to that of PGA peripherals.
- If “Embedded HW OCP” and “Embedded PGA” are enabled, the selection of ADCs and comparators (and their inputs and ‘+’ pins) is linked to that of PGA peripherals (and theirs '+' inputs).
- If “Embedded HW OCP” is enabled and “Embedded PGA” is disabled, the selection of comparators is free.
- If “Embedded HW OCP” and “Embedded PGA” are both disabled, the selection of comparators and ADCs is free.

The pairs that can be used are:

- OPAMP1/OPAMP2 can be used in a project based on STM32F302 or STM32F303;
- OPAMP3/OPAMP4 can be used additionally in a project based on STM32F303.
- ADC1/ADC2 can be used in a project based on STM32F302 or STM32F303;
- ADC3/ADC4 can be used additionally in a project based on STM32F303.

If both PGA and comparator for OC protection are used they will share the same input pins for the motor current measurement signal.

Dual three shunt topology, shared resources: If both drives are three shunts it can be possible to share the ADC and/or the PGA to perform the motor current measurement. To do this, it is mandatory to have both use external operational amplifier or both use the embedded PGA for the motor current measurement signals amplification. It can be settled by the user in the ST MC Workbench clicking the “Shared resource” check box in the Control Stage -> Analog Input.

If shared resource is settled and external operational amplifier is used, it is possible to use the pairs ADC1/ADC2 or the pairs ADC3/ADC4 for both drivers. In this case ST MC Workbench proposes the allowed inputs pins for motor currents measurement.

If shared resource is settled and embedded PGA is used, the following configuration is used:

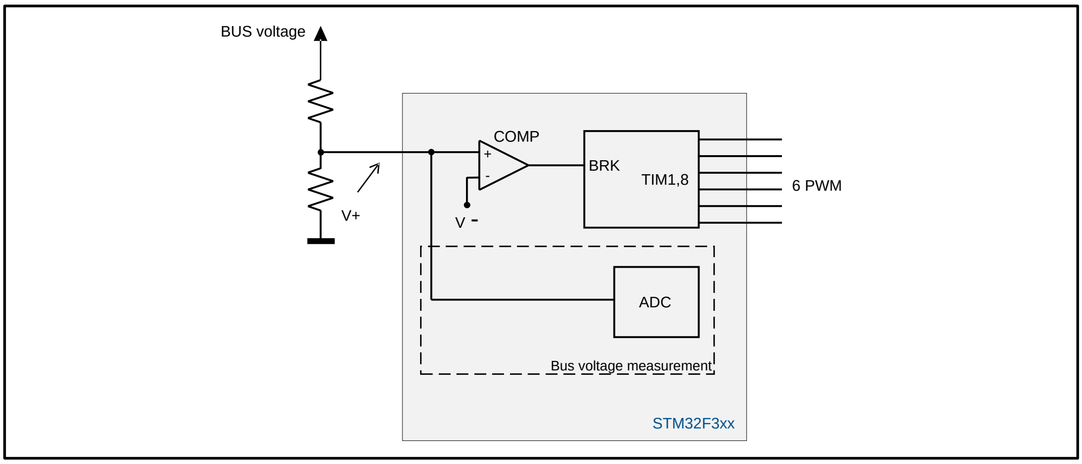
- The pair OPAMP1/OPAMP3 is used,
- OPAMP gains is only “Internal”,
- External capacitor filer is not allowed,
- Input pins are: PA5, PA7, and PB13 respectively U, V, W for motor 1 and PA1, PA3 and PB0 respectively U, V, W for motor 2.

In this case, if the hardware over current protection is managed by internal comparators, it is mandatory to connect externally the pins PA3 with one of the pins PB14 or PD14 and connect externally the pins PA5 with one of the pins PB11 or PD11. The pins selected can be settled in the workbench in Control Stage -> Analog Input -> Phase current feedback -> Protection.

10 Overvoltage protection with embedded analog (STM32F3x)

Figure 34: "Overvoltage protection network" shows a basic implementation of over-voltage protection network that can be implemented using the internal resources of the STM32F30x.

Figure 34: Overvoltage protection network



In this case, the principle is similar to the one described in :

- A resistive voltage divider provides a signal proportional to the bus voltage. It must be sized depending on the bus voltage range requested by the target application, so that is never exceeds the MCU's input maximum admissible voltage level.
- This reading is compared to an over-voltage threshold to generate a fault signal.
- If the threshold is exceeded, a break signal stops the PWM generation putting the system in a safe state.

As mentioned before, these actions can be performed automatically using the internal comparator of the STM32F30x. In this case, it is convenient to use the second break functionality (BRK) of the advanced timer in order to differentiate the action to perform on the PWM signals in case of an over-current: disable PMW generation or turn-on low side switches.

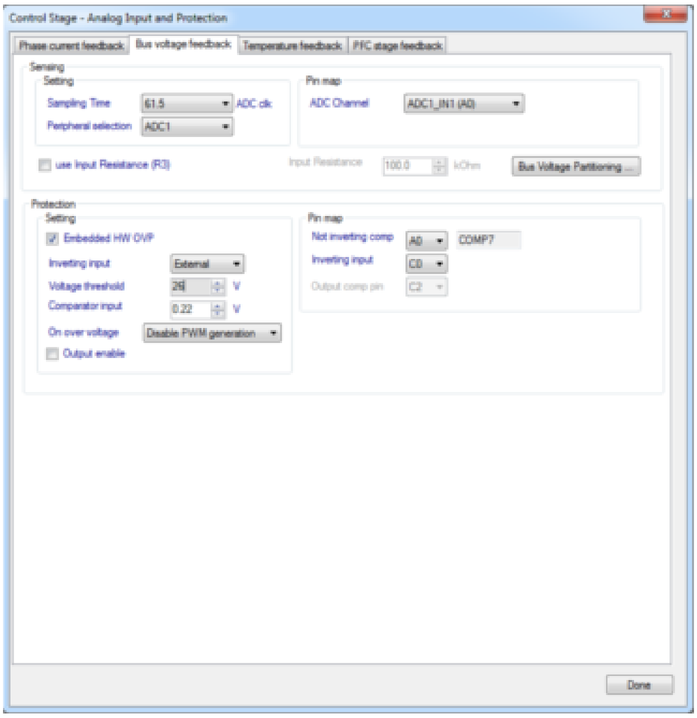
The MC library can be arranged to match these configurations by using the ST MC Workbench, to create a project based on STM32F302 or STM32F303. From the dialogue window located in Control Stage -> Analog Input -> Bus voltage feedback (*Figure 35: "STMCWB window related to ADC/COMP settings for DC bus voltage"*), setting:

- “Embedded HW OVP” checkbox;
- HW OVP internal threshold: selecting “Internal” in the “Inverting input” drop down list and choosing the internal voltage reference (among available values) in “Comparator Input”.
- HW OVP external threshold: selecting “External” in the “Inverting input” drop down list and editing the external voltage reference in “Comparator Input”.
- HW OVP internal threshold using DAC: selecting “DAC” in the “Inverting input” drop down list and editing the DAC voltage reference to be generated in “Comparator Input”. A DAC channel must be assigned for this functionality (OVP) from the related

dialogue window located in Control stage -> DAC functionality (*Figure 36: "STMCWB window related to DAC settings"*)

- The selection of 'not-inverting' input pin contextually picks the comparator to be used.
- The drive behavior when an overvoltage state is found: disable PWM generation, or turn-on low side switches;
- Enabling or disabling the comparator output has no effect on the overvoltage protection functionality itself.

Figure 35: STMCWB window related to ADC/COMP settings for DC bus voltage



Sampling Time 61.5 ADC clk

Peripheral selection ADC1

ADC Channel ADC1_IN1 (A0)

□ use Input Resistance (R3)

Input Resistance 100.0 kOhm

☑ Embedded HW OVP

Inverting input External

Voltage threshold 26 V

Comparator input 0.22 V

On over voltage Disable PWM generation

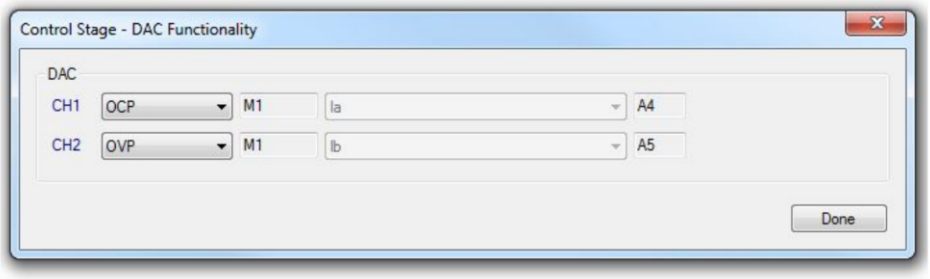
□ Output enable

Not inverting comp A0 COMP7

Inverting input C0

Output comp pin C2

Figure 36: STMCWB window related to DAC settings



CH1 OCP M1 Ia A4

CH2 OVP M1 Ib A5

11 Revision history

Table 9: Document revision history

Date	Version	Changes
03-May-2011	1	Initial release.
14-Nov-2012	2	Added references to STM32F05xx, STM32F2xx and STM32F4xx in the title and throughout the document. Updated SDK V3.0 to V3.4. Updated Motor Control Library project and User project in Section 5.1: Motor control workspace. Updated Section 5.2: MC SDK customization process, including Table 1: Project configurations. Added Option 1 to Section 6: How to download the full LCD user interface and updated Option 2 and Option 3. Added text to Section 7.2: LCD User interface structure.
16-Dec-2013	3	Changed the software library version (from V3.3 to v3.4). Added references to STM32F303xB/C in the title and throughout the document. Added new available ST documents in Section : Available from www.st.com or your STMicroelectronics sales office Updated option in Section 6: How to download the full LCD user interface. Updated Section 5.2: MC SDK customization process. Added Section 9: Current sensing and protection with embedded analog (STM32F3x). Added Section 10: Overvoltage protection with embedded analog (STM32F3x). Updated: Figure 15, Figure 17, Figure 18, Figure 19, Figure 20, Figure 21, Figure 27, Figure 29, Figure 30
22-May-2014	4	Changed software library version from v3.4 to v4.0. Removed Table1 Applicable product. Added text in Section 1: Motor control library features. Modified Section 1.1: User project and interface features. Modified Section 5.1: Motor control workspace. Updated Figure 7: IAR EWARM IDE Workspace overview Added Figure 8: Keil uVision workspace overview Updated Figure 9: Workspace batch build for IAR EWARM IDE Added Figure 10: Workspace batch build for Keil uVision Updated Figure 14: LCD UI project, Figure 15: User interface reference, Figure 17: STM32 Motor Control demonstration project welcome message and Figure 19: Dual control panel page

Date	Version	Changes
18-May-2015	5	Updated Introduction; Updated Section 1: Motor control library features Updated Section 2.2: Related documents Added Section 3: Motor profiler and One Touch Tuning Added Section 4: On-the-fly Sensorless startup Added figure from Figure 1: Motor profiler to Figure 6: Enabling “On-the- fly” start-up with Advanced profile Updated Figure 29: STMCWB window related to ADC/COMP settings for DC bus voltage
11-Sept-2015	6	Updated – Document title – Introduction – Table 1.: Project configurations – Section 2.2: Related documents
14-Sep-2016	7	Text and formatting changes throughout document Full rewrite of *Section "Introduction"* Full rewrite of *Section 3: "The new Motor Profiler procedure"* (was "Motor profiler and One Touch Tuning") and *Section 3.1: "Restrictions and disclaimer"* (was "Restrictions") In *Section 4: "On-the-fly sensorless startup"*, removed figure "Enabling “On-the-fly” start-up with Advanced profile"

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.