Firmware plugin for STSW-ESC001V1 board with ST Motor Control FOC SDK

Introduction

The STSW-ESC001V1 firmware package for the STEVAL-ESC001V1 board includes the application code to support the electronic speed controller (ESC).

It embeds a sensorless FOC algorithm with speed regulation and an active braking function. It also performs a PWM detection routine capable of accepting command signals from external units for driving flight control boards (FCU) or equivalent.

In case of stalling, a dedicated function manages motor restart. For safety reasons, the ESC implements the arming/disarming feature.

Contents

List of figures

1 Getting started

1.1 Hardware overview

The STEVAL-ESC001V1 evaluation board is based on the STM32F303 microcontroller, L6398 driver and STL160NS3LLH7 power MOSFETs. It is an electronic speed controller (ESC) designed to drive a single three phase brushless motor (BLDC/PMSM), performing a sensorless FOC algorithm with speed regulation and active braking function. This unit can accept commands from an external unit for driving and monitoring a flight control board, for instance. For this purpose, several communication bus interfaces (UART, CAN, I²C) are available.

The board includes a BEC 5 V circuit and embeds an overcurrent/overvoltage and thermal protection circuit. Its form factor renders it suitable for small and light R/C vehicles and its motor current capability meets the power requirements of larger vehicles like professional drones.

The STM32F303 microcontroller powering the digital section is an ARM®Cortex®-M4 32-bit RISC core with FPU operating at a frequency of up to 72 MHz, and embedded floating point unit (FPU) and memory protection unit (MPU). It has high-speed embedded memories with 128 Kbytes of Flash and 32 Kbytes of SRAM and an extensive range of enhanced I/Os and peripherals connected to two APB buses. The devices offer up to four fast 12-bit ADCs (5 Msps), seven comparators, four operational amplifiers, up to two DAC channels, a low-power RTC, up to five general-purpose 16-bit timers, one general-purpose 32-bit timer and two timers for motor control. They also feature standard and advanced communication interfaces: up to two I2Cs, up to three SPIs, three USARTs, up to two UARTs, CAN and USB. A dedicated port is included for microcontroller programming.

The power section is based on STL160NS3LLH7 Power MOSFETs and the L6398 driver. The STL160NS3LLH7 exhibits low on-state resistance and capacitance for improved conduction and switching performance (very low on-resistance, very low Qg, high avalanche ruggedness and embedded Schottky diode)

The L6398 is a high voltage device manufactured with the BCD™ "offline" technology. It is a single-chip half bridge gate driver for N-channel power MOSFETs or IGBTs. The high-side (floating) section is designed to withstand a voltage rail up to 600 V. The logic inputs are CMOS/TTL compatible down to 3.3 V for easy microcontroller/DSP interfacing.
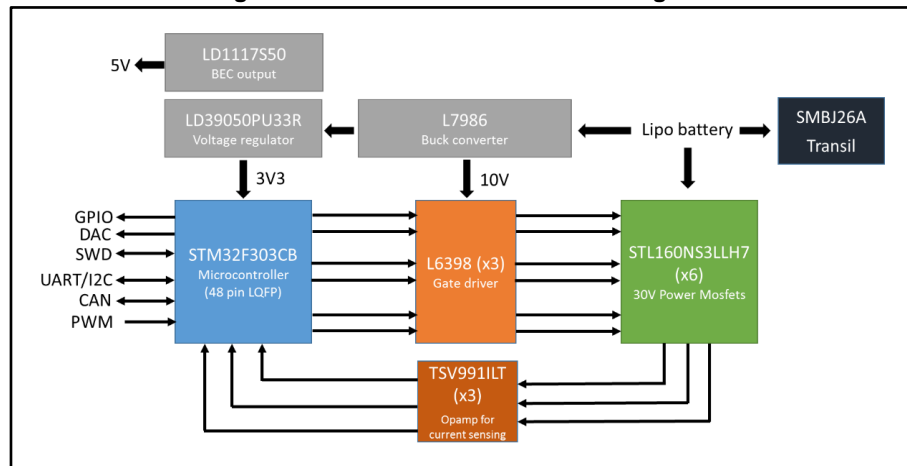
Figure 1: STEVAL-ESC001V1 evaluation board
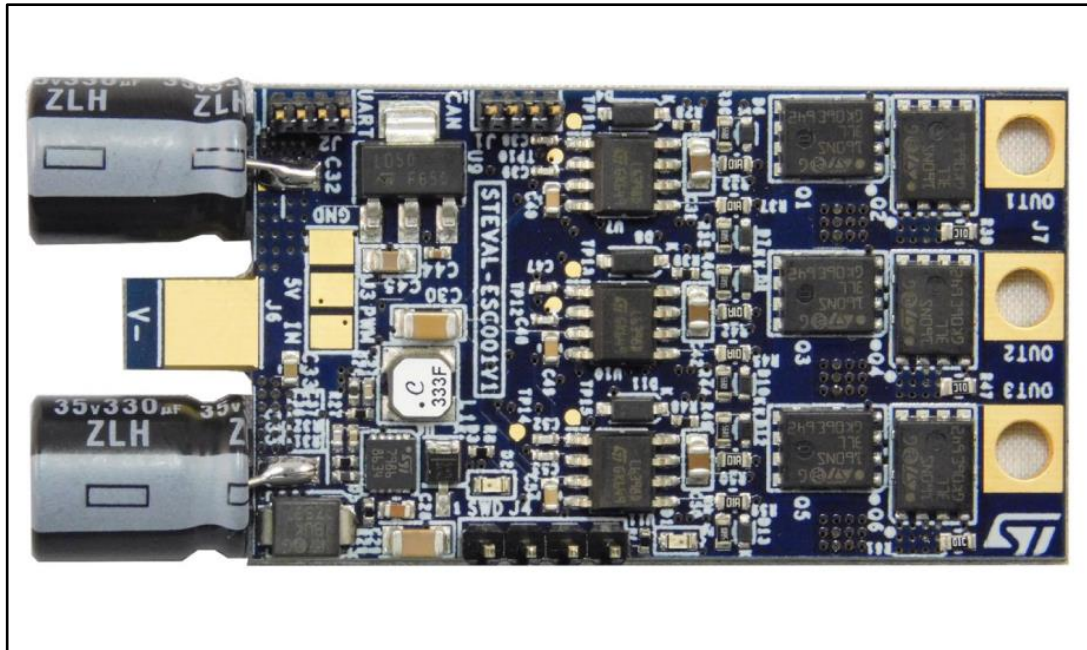
1.1.1 STEVAL-ESC001V1 architecture

The STEVAL-ESC001V1 power and control platform is based on the ST componentry illustrated below.

Figure 2: STEVAL-ESC001V1 block diagram

1.2 Installing the software

Visit *www.st.com* to download and install:

1. **STSW-STM32100**: STM32 PMSM FOC Software Development Kit v4.3 (STSW-STM32100) with the MC FW library, the Motor Control Workbench GUI and the Motor Profiler Tool.
2. **STSW-ESC001V1**:
   – ESC_STM32_FOC_FW dedicated FW example based on MC FW library
   – ESC_STMCWB_prj project for the Motor Control Workbench
   – ESC_STMotorProfiler_addon addon to upload the STEVAL-ESC001V1 board on the current Motor Profiler version

The ESC board is not provided with the firmware preloaded.

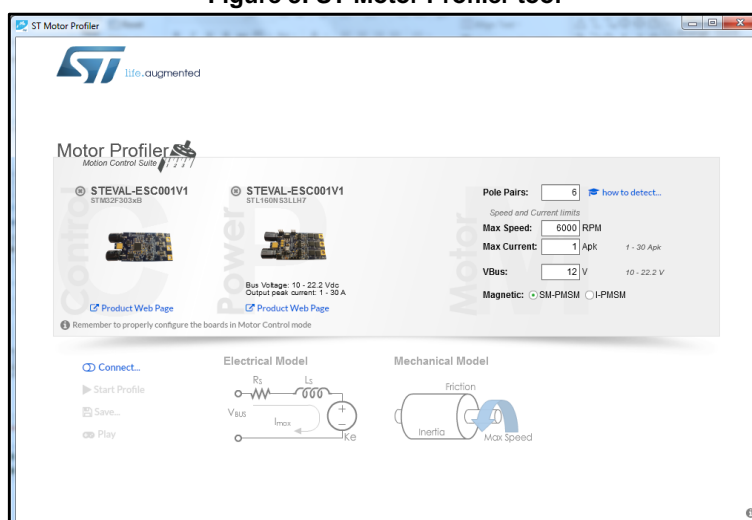### 1.2.1 Motor Profiler update with STEVAL-ESC001V1 support

To upgrade your Motor Profiler to **version 1.1.0**, paste the contents of the ESC_STMotorProfiler_addon\resources folder into the ST SDK FOC library installation folder:

- e.g., C:\Program Files (x86)\STMicroelectronics\FOC SDK\v4.3.0\STMotorProfiler\

Replace any duplicate files.

Figure 3: ST Motor Profiler tool

1.3 Firmware package overview

The STSW-ESC001V1 firmware enables field oriented control (FOC) and connection with the ST MC Workbench and ST Motor profiler tools.

It is based on ST PMSM FOC SDK (STSW-STM32100) and includes an FOC FW example for ESC management.

The changes to the user project and MC library project include:

- User project space
    a. Main.c - modified
    b. STEVAL_ESC001V1.c – added
    c. CrossCheck.h – modified
- MC Library project space
    a. PWMnCurrFdbkPrivate.h - modified
    b. PWMnCurrFdbkClass.h - modified
    c. R3_1_F30X_PWMnCurrFdbkClass.h - modified
    d. R3_1_F30X_PWMnCurrFdbkClass.c - modified
    e. NTC_TemperatureSensorClass.c - modified
    f. PWMnCurrFdbkClass.c - modified

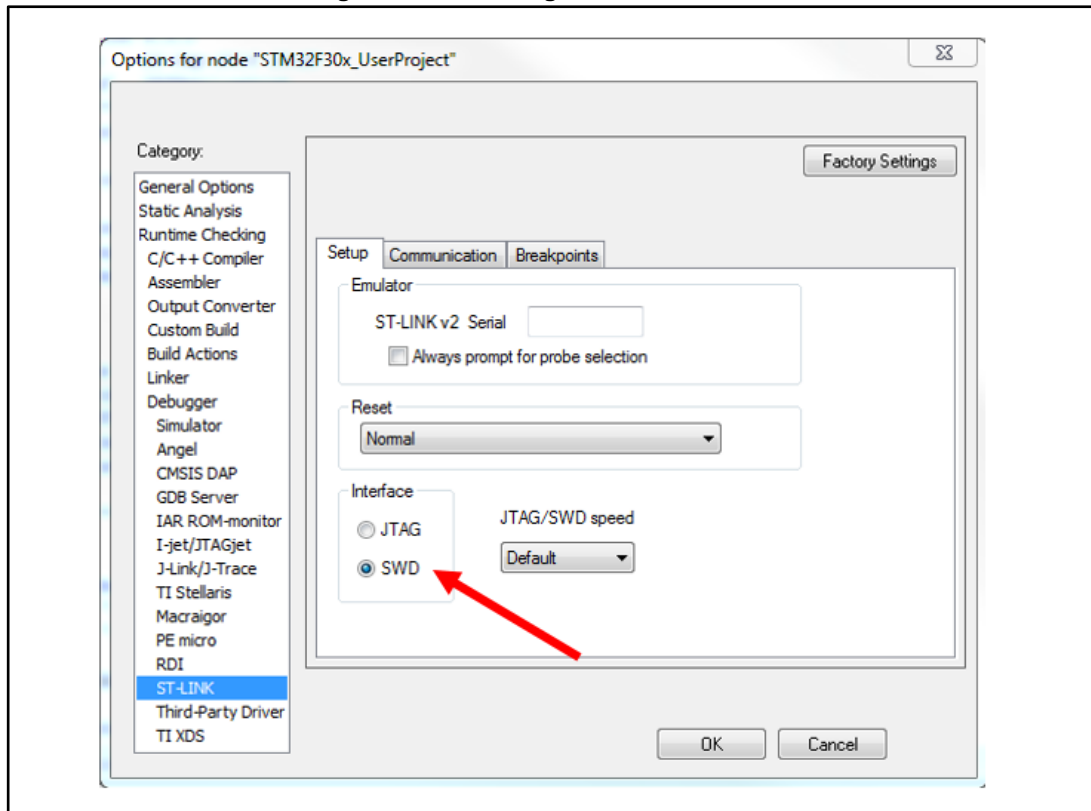   g.   Control stage parameters.h - modified

You can compile the source code by selecting "STM32303C-EVAL_SINGLEDRIVE" in the IAR Workbench project settings and upload the firmware to the microcontroller.

> For hardware connection details, refer to user manual *UM2197 Electronic speed controller for BLDC and PMSM three phase brushless motor on* www.st.com.

Before uploading the firmware, set the SWD interface in the IDE tool.

Figure 4: SWD configuration on IAR tool

2 Configuring the firmware example

2.1 ST Workbench tool with STEVAL-ESC001V1

1   Open the *SDK43x-STM32F303-STEVAL_ESC001V1.stmcx* workbench project from the ESC_STMCWB_prj folder.

Figure 5: ST Motor control workbench – motor section

2   You can enter the motor parameters manually if you know them.

If you do not know the motor parameters, you can use the Motor Profiler tool to calculate them.

All the other parameters, power stage, control stage and driver management are already prefilled for STEVAL-ESC001V1

Figure 6: ST Motor control workbench – motor parameters

3   Click on the generation button.

The header files are built in the FW FOC library package of at:
..\STSW-ESC001V1\ESC_STM32_FOC_FW\Web\SystemDriveParams

4   Compile the workspace for STM32303C-EVAL_SINGLEDRIVE with IAR workbench tool.

You should now be able to work with ST MC SDK.

5   Use the ST Workbench tool to monitor the main FOC algorithm variables, motor speed, applied bus voltage and board temperature.

The STEVAL-ESC001V1 board communicates with ST MC Workbench through a typical USB/RS232 converter connected to the UART port on the board.
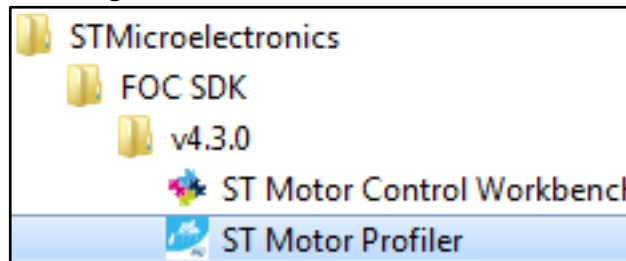
Figure 7: ST Workbench tool
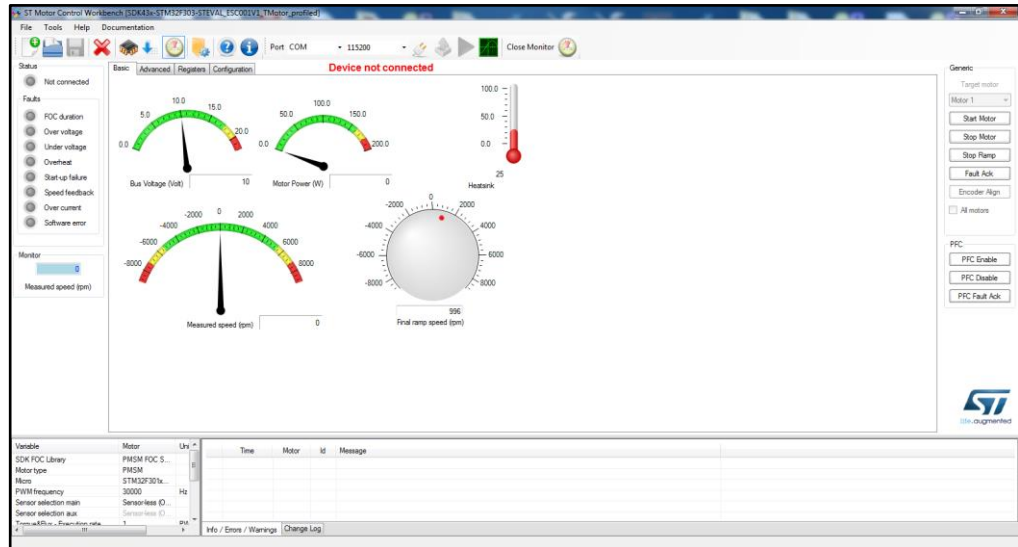
2.2   ST Motor profiler tool with STEVAL-ESC001V1

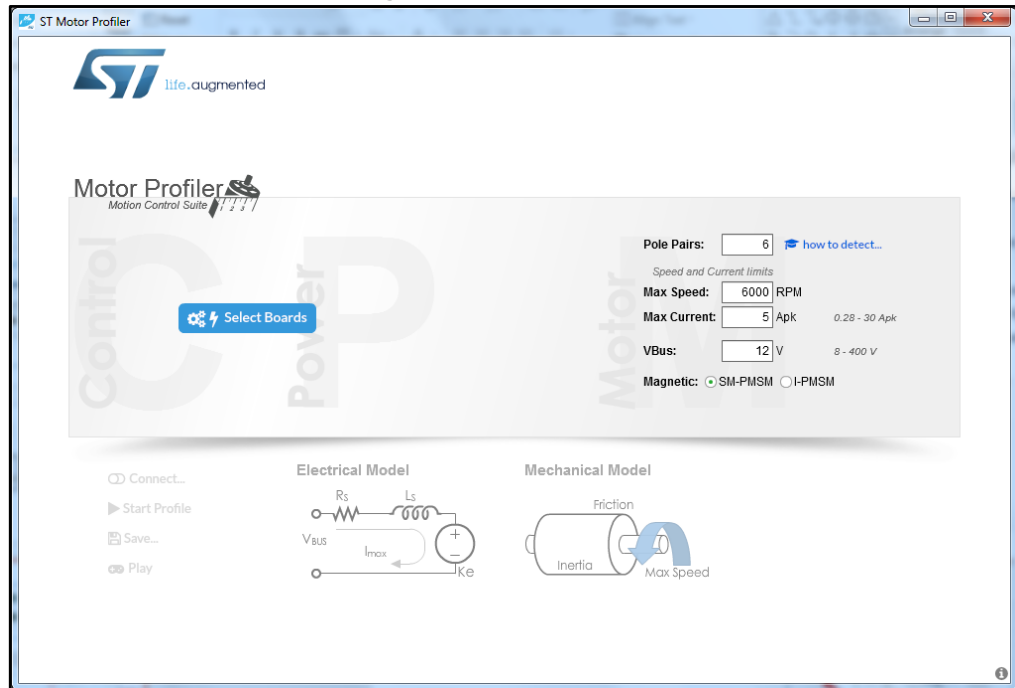The Motor Profiler tool can determine unknown motor parameters

1   Launch ST Motor Profiler from the start menu.
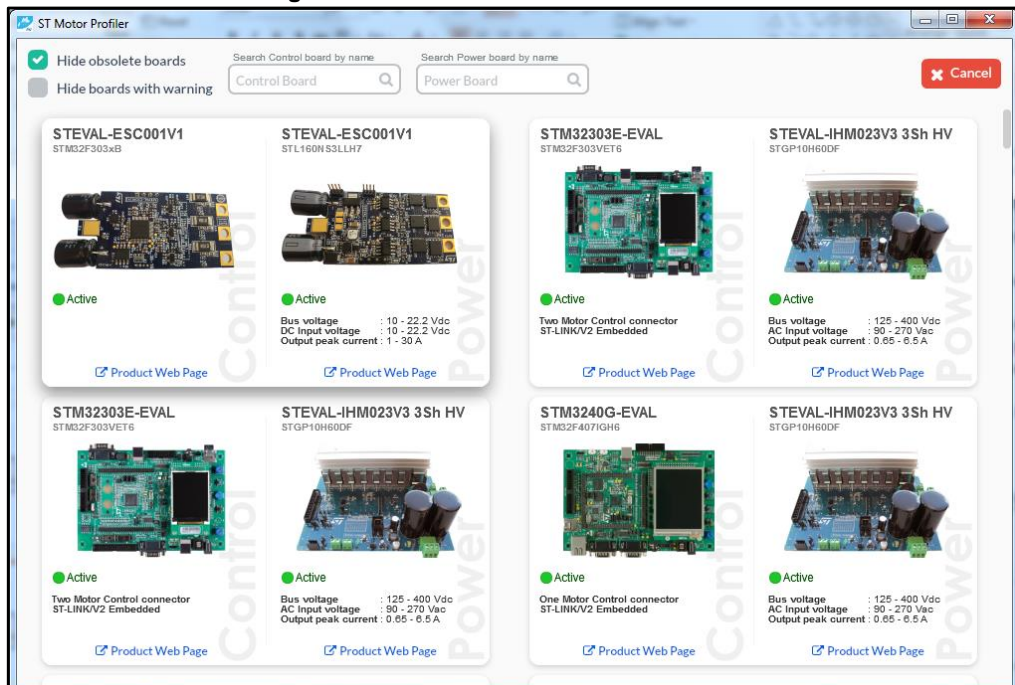
Figure 8: ST Motor Profiler in Start menu

2   The start screen of the Motor Profiler tool appears.

Figure 9: ST Motor Profiler

3   Click the Select Boards button in the Motor Profiler to open a list of supported boards. The Motor Profiler feature can only be used on the systems listed there.

4   Select the STEVAL-ESC001V1 board

Figure 10: ST Motor Profiler – list of boards

5   Fill in the required fields

i.e., Pole Pairs, Max Speed, Max Current, Vbus and Magnetic

Figure 11: ST Motor Profiler – parameters
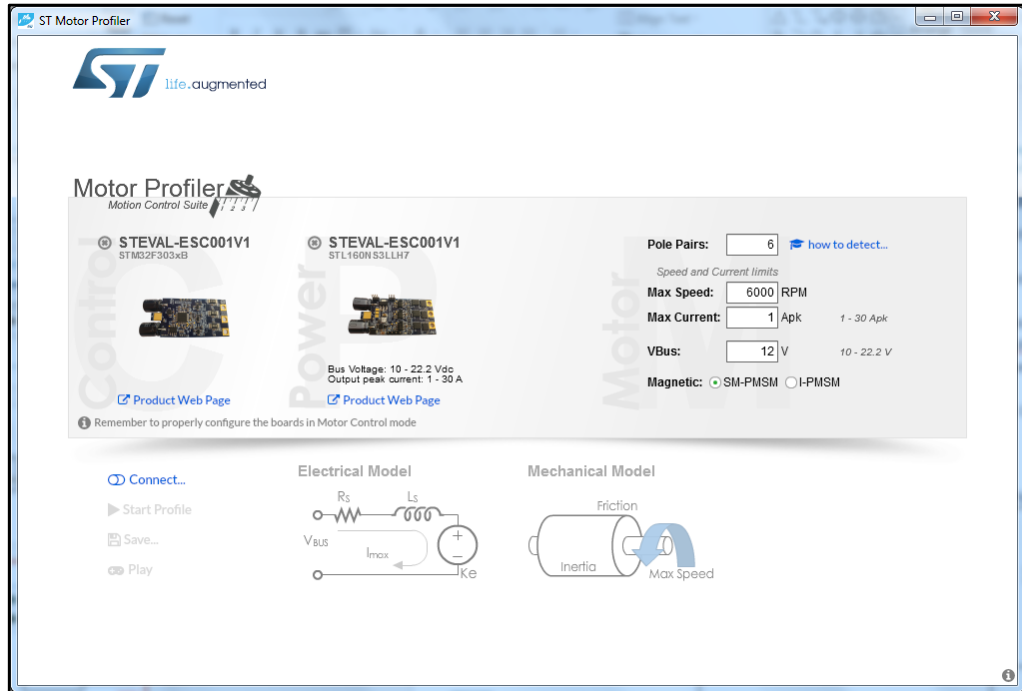
6   Check the boards are correctly configured for motor control, supplied with an appropriate input voltage and connected with the serial communication cable and JTAG/SWD programming cable.

7   Click on the Connect button.

    a.   If the programming procedure can't be executed, check the JTAG/SWD programming cable.

    a.   If the programming procedure is executed but Motor Profiler still can't communicate with the board, check the serial communication connections.
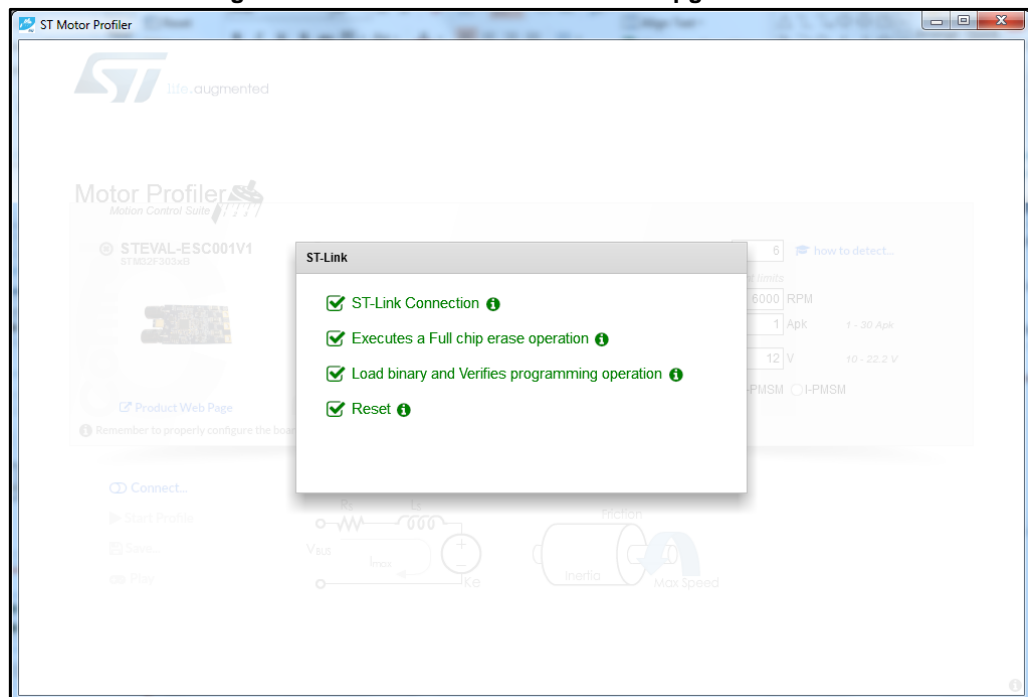
8   If the board is new or the flash on the microcontroller has been erased, click Upgrade Firmware to automatically load the correct firmware onto the microcontroller.

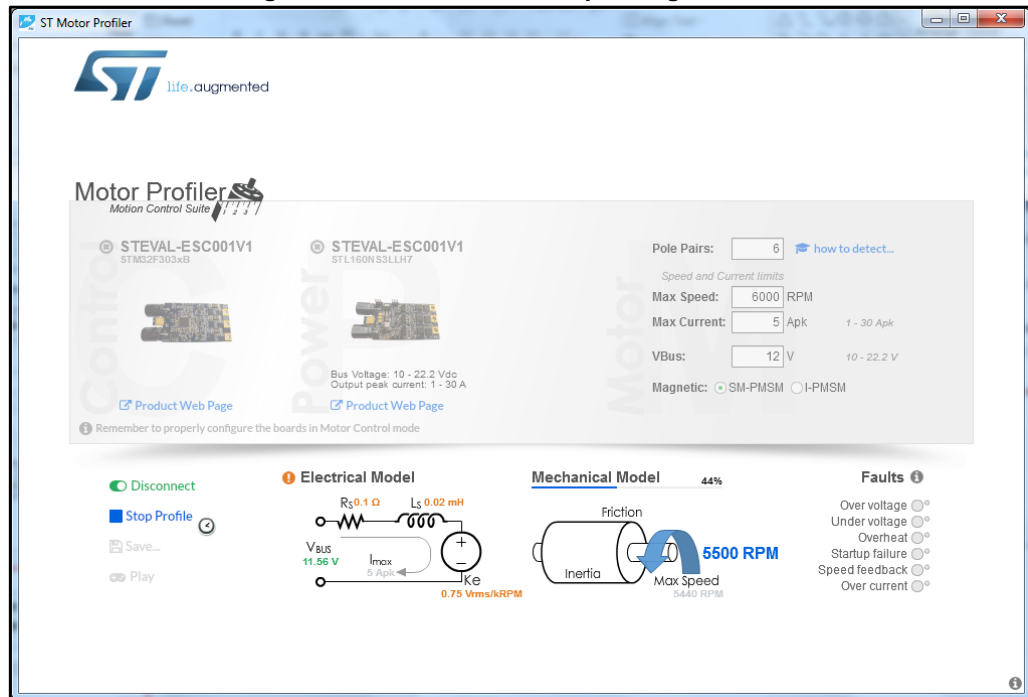**Figure 12: ST Motor Profiler – firmware upgrade**
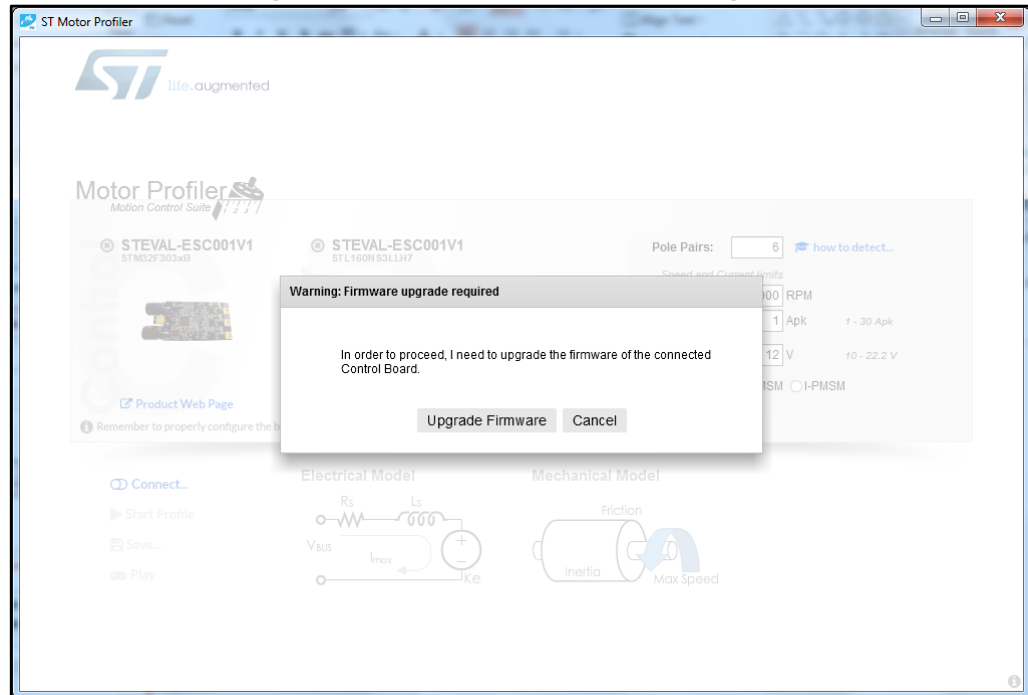
9   You can monitor the progress of the firmware update.

**Figure 13: ST Motor Profiler – firmware upgrade status**

10   Once connection is established, press Start Profile

At the end of the procedure, all measured parameters will appear.

Figure 14: ST Motor Profiler – profiling the motor

11   Note the parameters (Rs, Ls, B-Emf constant, friction and inertia) and use them in the MC Workbench motor parameters table

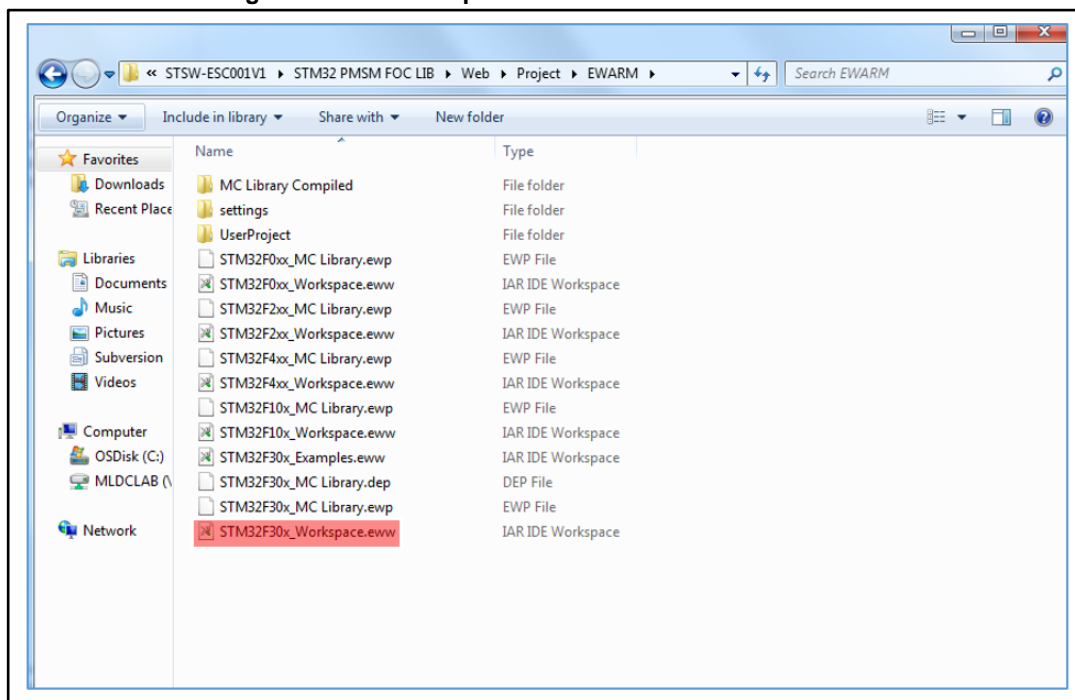**Figure 15: ST MC Workbench – motor parameters**
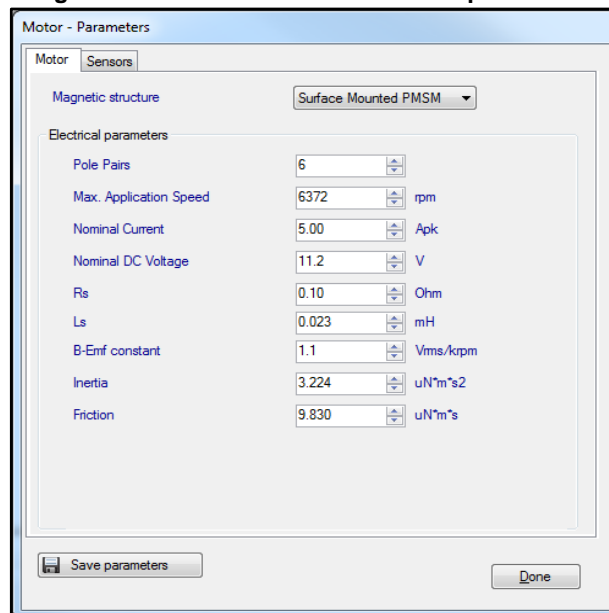
3 STSW-ESC001V1 firmware package

## 3.1 Application code for STEVAL-ESC001V1

The STSW-ESC001V1 firmware is configurable with the IAR tool. You can open STM32F30x_Workspace in the location shown below.

Figure 16: IAR workspace for STSW-ESC001V1 firmware

The user project appears as shown below; the application file with the source code for ESC functions is highlighted in red.
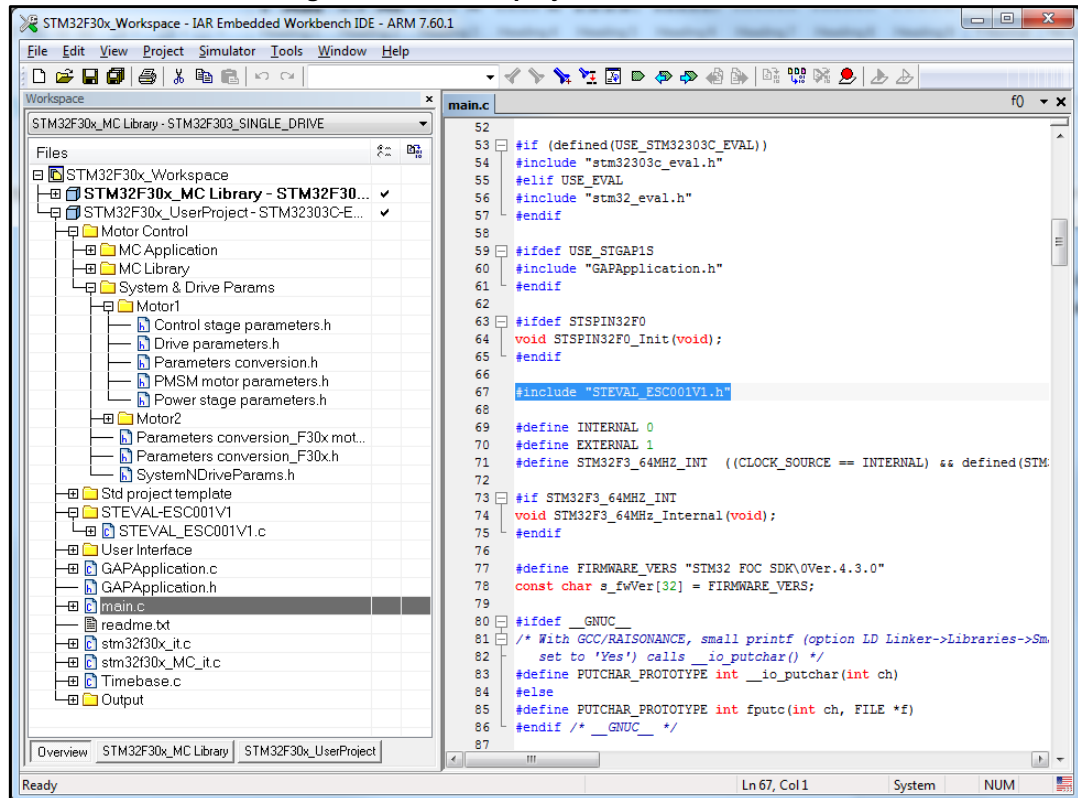
Figure 17: IAR User project for STEVAL-ESC001V1 – application file
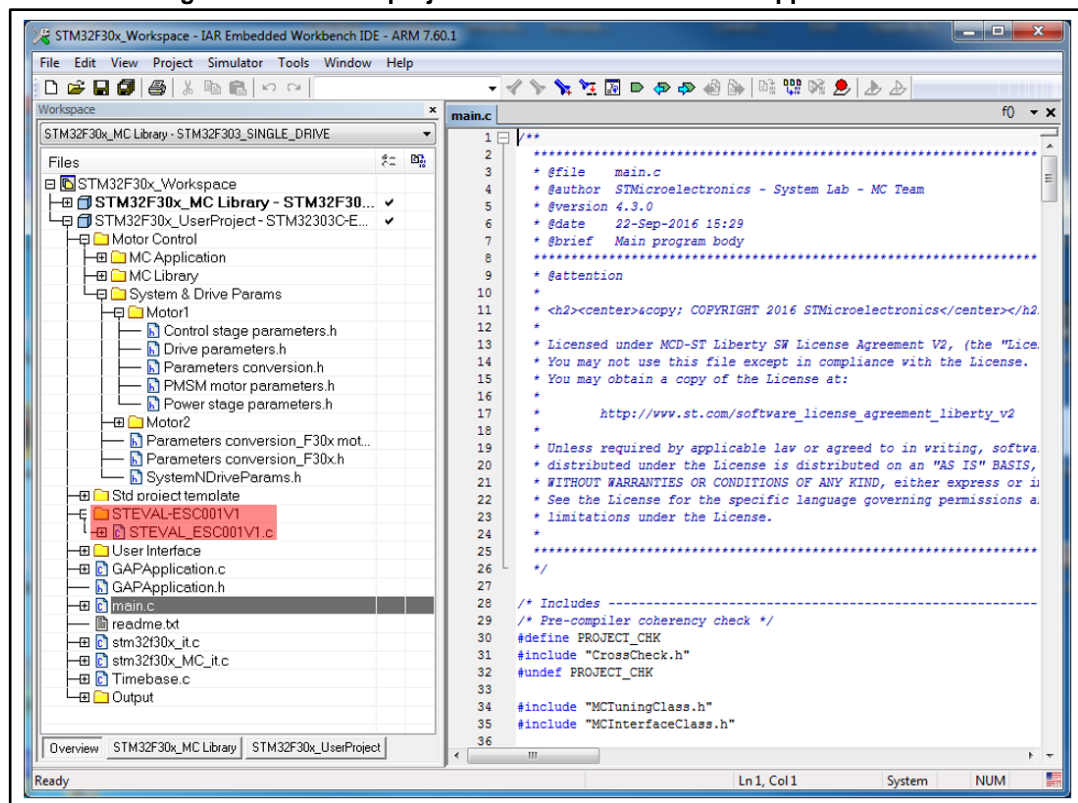
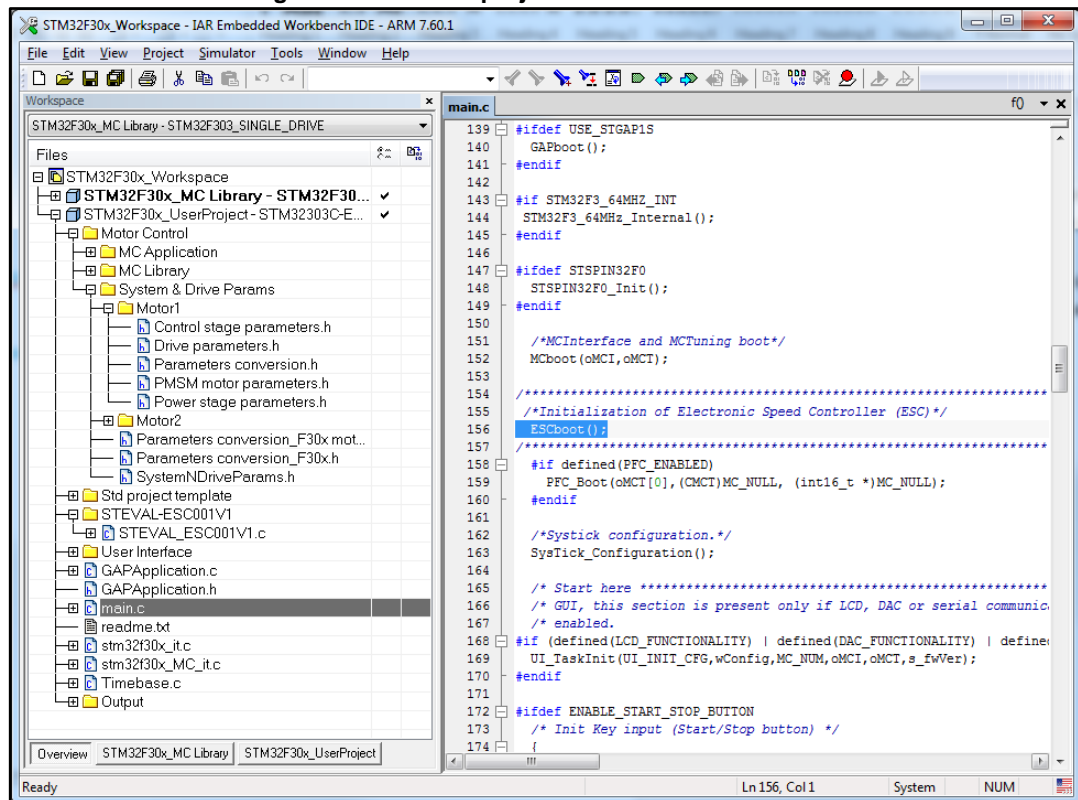The **main.c** file has the following extra lines of code:

- `#include "STEVAL_ESC001V1.h"`

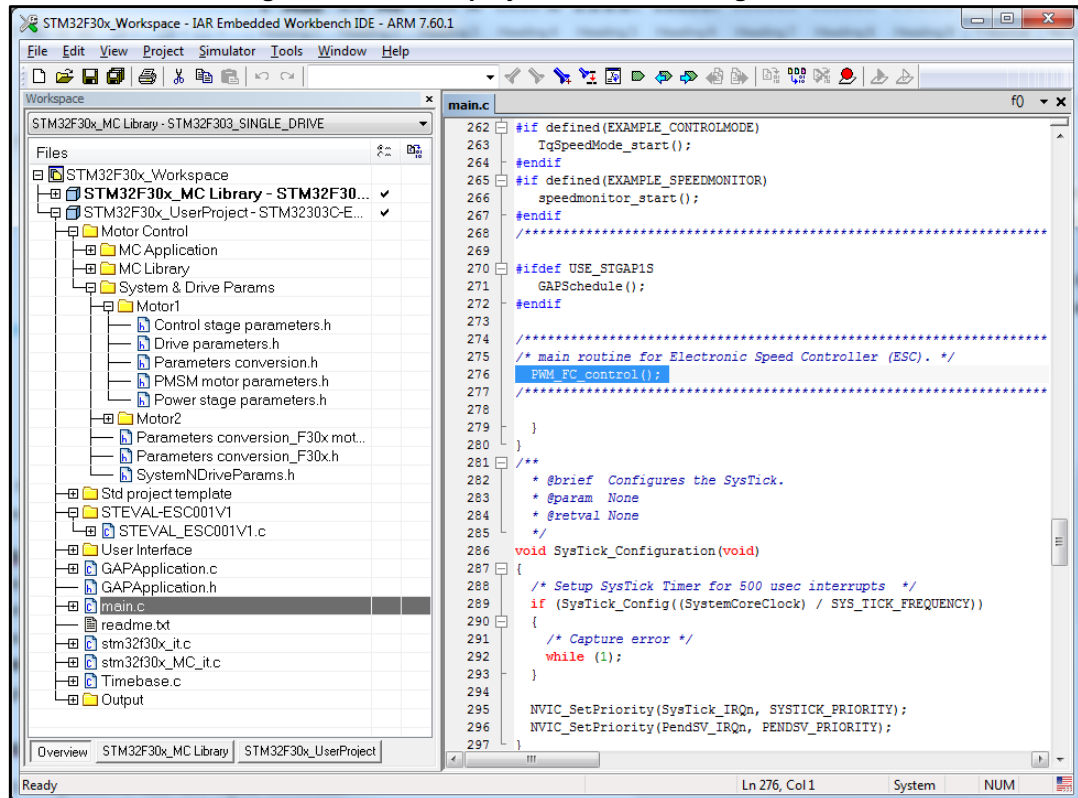**Figure 18: IAR User project – header included**

- `ESCboot()`

Figure 19: IAR User project – ESCboot function call

- `PWM_FC_control()`

Figure 20: IAR User project – PWM decoding function

These functions form the core of the ESC management.

3.1.1 ESCboot()

This function initializes the main peripheral to configure the input capture PWM.

It contains an `PWM_FC_Init()` that configures GPIO pin PA15 to receive the PWM signal and is connected internally with TIM2 peripheral (Input Capture mode). It also configures an additional ADC (ADC3) peripheral for regular conversion by two different ADCs.
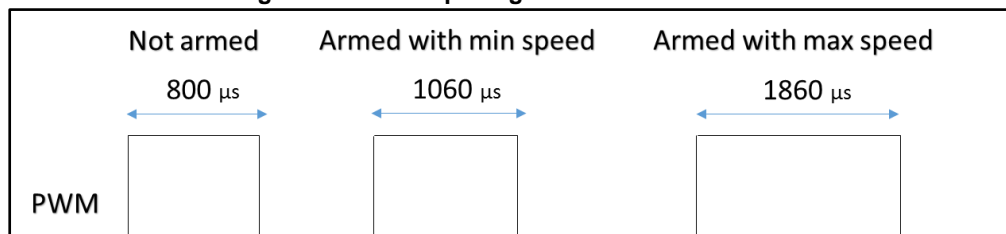
3.1.2 TIM2_IRQHandler(…)

This function implements the decoding function of the input PWM.

The `TIM2_IRQHandler` interrupt handler detects the rising and falling edge of the external PWM signal connected on INPUT channel (PA15) of the STEVAL-ESC001V1 board.

A digital filter returns the average value.

Figure 21: PWM input signal for the ESC control

3.1.3 PWM_FC_control()

This function implements the state machine to control the motor speed based on the PWM input signal.

It restarts the motor if a fault (e.g., propeller stopping suddenly) occurs. An internal time scheduler at fixed frequency regulates the change of the following states:

- **SM_ARMING**: controls the arming of the ESC (command motor start) according to the duty cycle value received. It also ensures that Ton (PWM) remains stable for a time (changeable in the FW) before changing step.
- **SM_ARMED**: the ESC calls the API function of the MC SDK to run the motor
- **SM_POSITIVE_RUN**:
  – If no fault occurs, this state regulates the motor speed according to the Ton value and checks whether this value is above the minimum threshold.Otherwise it requests a step change and stops the motor.
  – The minimum and maximum threshold are changeable by firmware (`Ton_min` and `Ton_max`).
  – It also manages the interruption of the PWM signal: a time delay (changeable in the firmware) is set before the motor is stopped due to ongoing failure of the PWM signal.
- **SM_STOP**: sends a stop command to the MC library and disarms the ESC.

The `PWM_FC_control()` function also implements PWM signal stability control so that a brief (less than 1500 ms) signal interruption does not trigger a motor stoppage event.

3.2 Configuration for ST Workbench tool usage

The STEVAL board can communicate with ST MC Workbench to monitor the variables and the state of the motor.

Start, stop and speed regulation are managed normally by the application code, but you can disable the application functions and activate manual mode.

To do so, comment the following line in STEVAL-ESC001V1.c, recompile and upload the firmware to the MCU:

```
// #define PWM_ESC   <- comment this line to enable the manual mode (ST WB mode)
```

After selecting the COM port and setting the correct UART speed (115200), the green LED on ST Workbench signals a correct connection.

4 Revision history

Table 1: Document revision history

Date	Version	Changes
10-Apr-2017	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY