



Demonstration firmware for the
STEVAL-IHM022V1 dual motor FOC drive board

Introduction

This user manual describes the demonstration firmware running on the STEVAL-IHM022V1 STM32™-based dual motor FOC drive demonstration board. The STEVAL-IHM022V1 is designed as a platform to evaluate the capabilities of the high-density STM32F103ZE microcontroller and its on-board peripherals to perform dual motor control operations in simultaneous mode. Up to two motors can be driven in field-oriented control (FOC), single-shunt resistor and in sensorless mode.

The dual motor control firmware uses the FOC routines implemented on the STM32 PMSM library ver. 2.0 firmware package and shares the same principles for configuring the motor drive with user parameters.

Moreover, the firmware architecture has been extended to treat each motor as an independent entity, rendering the control of each motor completely independent from the other.

An embedded user interface (UI) composed of an LCD TFT 320 x 240 display and 5-position joystick allows the user to set some motor control drive parameters in real time during motor operations.

Contents

- 1 Functional description 5**
 - 1.1 Power control 5
 - 1.2 Clocking 5
 - 1.3 Reset control 5
 - 1.4 Debug JTAG interface 5
 - 1.5 Serial wire debugger interface 5
 - 1.6 Display devices 5
 - 1.6.1 LCD 5
 - 1.6.2 LED 5
 - 1.7 Interfaces 6
 - 1.7.1 RS-232 6
 - 1.7.2 CAN 6
 - 1.7.3 USB 6
 - 1.8 Motor control 6
 - 1.9 Miscellaneous peripherals 6
 - 1.9.1 Joystick 6
 - 1.9.2 Push-buttons 6
 - 1.9.3 Storage memory 6
 - 1.9.4 STEVAL-IHM022V1 board jumper configuration 6

- 2 Running the demonstration 7**
 - 2.1 Demonstration firmware system setup 7
 - 2.1.1 Hardware requirements 7
 - 2.1.2 Setup of the power boards 7
 - 2.1.3 Control board setup 7
 - 2.1.4 Software requirements 8
 - 2.1.5 Configuring the firmware with user parameters 8
 - 2.1.6 Building and downloading the firmware code into the STM32 using EWARM v5.11 8
 - 2.1.7 Connecting the STEVAL-IHM022V1 to the MB459B power stage board 9
 - 2.1.8 Connecting the STEVAL-IHM022V1 to the STEVAL-IHM011V1 power stage board 9

| | | |
|----------|--|-----------|
| 2.2 | How to navigate the system menu | 9 |
| 2.3 | Getting started with the demonstration board | 10 |
| 2.3.1 | Starting or stopping the motor(s) | 10 |
| 2.3.2 | Alternate motor starting | 11 |
| 2.4 | Advanced operations | 11 |
| 2.4.1 | Setting the control strategy | 11 |
| 2.4.2 | Setting the flux weakening parameters | 14 |
| 2.4.3 | Viewing the status of the power stage | 15 |
| 2.4.4 | Setting the observer and PLL gains | 15 |
| 2.5 | Debugging mode | 15 |
| 3 | Control strategy | 17 |
| 3.1 | Block diagram | 17 |
| 3.2 | ADC & FOC execution time diagram | 18 |
| 3.2.1 | Workload versus PWM frequency | 19 |
| 3.3 | Block diagram of the peripherals | 20 |
| 3.4 | Firmware configuration | 21 |
| 4 | Revision history | 24 |

List of figures

| | | |
|------------|--|----|
| Figure 1. | Main menu | 10 |
| Figure 2. | Selections for simultaneous dual-motor start | 10 |
| Figure 3. | Motor selection for single motor start | 11 |
| Figure 4. | Speed control settings | 11 |
| Figure 5. | Selecting the target speed | 12 |
| Figure 6. | Editing the target speed | 12 |
| Figure 7. | Torque control settings | 12 |
| Figure 8. | Speed PID coefficient settings | 13 |
| Figure 9. | Torque PID coefficient settings | 14 |
| Figure 10. | Flux PID coefficient settings | 14 |
| Figure 11. | Flux weakening coefficient settings | 14 |
| Figure 12. | Power stage status variables | 15 |
| Figure 13. | Observer and PLL gains setting | 15 |
| Figure 14. | DAC settings menu | 16 |
| Figure 15. | DAC settings items | 16 |
| Figure 16. | DAC settings navigation menus | 16 |
| Figure 17. | Dual motor control strategy block diagram | 17 |
| Figure 18. | ADC & FOC execution time | 18 |
| Figure 19. | Workload diagram | 19 |
| Figure 20. | Free microcontroller time vs. PWM frequency | 20 |
| Figure 21. | Block diagram of the peripherals | 20 |
| Figure 22. | Motor options: example 1 | 21 |
| Figure 23. | Motor options: example 2 | 22 |
| Figure 24. | Motor parameters example | 23 |

1 Functional description

1.1 Power control

The demonstration board can be powered from an external 5 V supply or from the USB connector. All other required voltages are provided by on-board voltage regulators.

1.2 Clocking

Two clock sources are available on the STEVAL-IHM022V1 demonstration board:

- 32 kHz crystal for embedded RTC
- 8 MHz crystal for the STM32F103ZE

1.3 Reset control

The reset can be generated by means of hardware or software:

- Reset button - activates the RESET input when pressed
- JTAG reset

1.4 Debug JTAG interface

Software debugging is carried out via a standard ARM® JTAG connection, a 20-pin IDC (insulation displacement connector) for connection to the standard ARM host interface.

1.5 Serial wire debugger interface

The serial wire debug port (SWD-DP) provides a 2-pin (clock + data) interface to the AHPAP port.

1.6 Display devices

1.6.1 LCD

A color LCD module is mounted on the STEVAL-IHM022V1 demonstration board. It interfaces with the microcontroller via an SPI peripheral.

1.6.2 LED

Four general-purpose LEDs are available, which are used as a display.

1.7 Interfaces

1.7.1 RS-232

The STEVAL-IHM022V1 demonstration board provides two on-board RS-232 serial ports. Both RS-232 ports are accessible through DB9 connectors.

1.7.2 CAN

The STEVAL-IHM022V1 supports CAN 2.0 A/CAN 2.0 B-compliant CAN bus communication based on a 3.3 V CAN transceiver. Both high-speed mode and slope-control mode are available. They are selected by setting a dedicated jumper.

1.7.3 USB

The demonstration board supports full-speed USB 2.0 communication, thanks to its embedded USB peripheral.

1.8 Motor control

The STEVAL-IHM022V1 demonstration board features multiple motor controls via three 34-pin connectors, which provide all required control and feedback signals to and from the motor power-driving boards.

1.9 Miscellaneous peripherals

1.9.1 Joystick

The joystick is four-directional and includes a selection key.

1.9.2 Push-buttons

The following push-buttons are provided:

- Key: user push-button
- Tamper: user push-button
- Wakeup: push-button used to wake up the processor from low-power mode

1.9.3 Storage memory

The STEVAL-IHM022V1 has a 64 Mbit SPI Flash memory connected to the SPI1 peripheral.

1.9.4 STEVAL-IHM022V1 board jumper configuration

To run the STEVAL-IHM022V1 demonstration firmware, please, refer to [Section 2.1.3](#).

2 Running the demonstration

2.1 Demonstration firmware system setup

2.1.1 Hardware requirements

- Dual motor control demonstration board STEVAL-IHM022V1.
- 5 VDC / 2 A isolated power supply (the TR15RA050 power adapter included with MB525 and MB679 demonstration boards is recommended. They can be ordered through order codes STM3210B-EVAL/A and STM3210E-EVAL/A, respectively).
- Two power boards (3PH inverter stage) MB459B or STEVAL-IHM011V1. Alternatively, any power board equipped with an MC connector can be used. Please refer to [Section 2.1.2](#) for instructions on how these boards must be configured.
- Two 24VDC / 80 W Shinano motors. This motor, included with the STM3210B-MCKIT starter kit, is an inner rotor-type 4-pole brushless DC motor with Hall sensor and encoder. For electrical specifications and mechanical dimensions, refer to the Shinano datasheets included on the STM3210BMCKIT CD-ROM. For different low-voltage PMSM motors, please refer to [Section 2.1.5](#).
- Two 34-pin flat cables for MC connectors.
- One 24 VDC / (5+5) A isolated power supply (for supplying the two power stages).
- One JTAG programmer/debugger dongle (a J-Link from SEGGER or IAR Systems™ is recommended). It is unnecessary if no modifications to the firmware code have been performed (see [Section 2.1.4](#)).

2.1.2 Setup of the power boards

Set up each MB459B or STEVAL-IHM011V1 power board to operate at low voltage and to be compliant with sensorless PMSM motor control driving with single shunt current reading settings. For instructions on how to set up the power boards, see user manuals UM0379 (for the MB459B) or UM0428 (for the STEVAL-IHM011V1).

2.1.3 Control board setup

Set up the STEVAL-IHM022V1 control board as follows (sensorless mode):

- Close jumper JP9
- Close jumper JP12
- Place a jumper between pin 1 and 2 of JP15
- Place a jumper between pin 2 and 3 of JP14
- Place two jumpers between 1-3 and 2-4 of J1
- Place two jumpers between 1-3 and 2-4 of J2

To use an encoder as a speed sensor for one or both of the motors, the following additional jumpers settings are required:

- JMP3 between 2 and 3 (enable encoder reading from channel A for the motor connected to MC main connector)
- JMP4 between 2 and 3 (enable encoder reading from channel B for the motor connected to MC main connector)

- JMP9 between 2 and 3 (enable encoder reading from channel B for the motor connected to MC sub connector)
- JMP10 between 2 and 3 (enable encoder reading from channel A for the motor connected to MC sub connector).

2.1.4 Software requirements

The source code of the demonstration firmware is provided by STMicroelectronics free of charge (after agreeing to the terms of the license agreement during the installation procedure). However, there are some portions of code that are protected by STMicroelectronics patent, and therefore are not accessible.

Thanks to the JTAG interface, any development tool can be used to customize the firmware and load it into the internal Flash memory of the STM32.

To download the dual motor control application demonstration without using the JTAG interface, the code is also provided in binary output format (.bin).

One method for loading firmware into the STM32 internal Flash memory is to use the Flash loader demonstrator application, which illustrates the system memory bootloader capabilities via UART1 serial port. Please refer to application note AN2606 and user manual UM0462 documents for details. Both the firmware application and documents are available on www.st.com at no charge.

2.1.5 Configuring the firmware with user parameters

The demonstration firmware provided can be directly loaded into the STM32 program memory (see [Section 2.1.4](#)) and is ready to be used at first power-up or immediately after a board reset event (after system hardware configuration has been completed, as described in [Section 2.1.7](#) and [2.1.8](#)).

If different motor and/or drive parameters must be modified before running the demonstration, please refer to [Section 3.4](#) for instructions. Once the modifications have been applied, the firmware must be re-built and loaded into the STM32 microcontroller using a development tool.

2.1.6 Building and downloading the firmware code into the STM32 using EWARM v5.11

The dual MC firmware demonstration provided has been built and loaded into STM32 Flash memory using IAR EWARM full version 5.11. For this tool the workspace files have been provided.

IAR Systems allows free downloading of its 32 K limited version (IAR KickStart Kit™) for product demonstration purposes. Although the dual MC demonstration firmware exceeds the maximum size limit, it is still possible to use the IAR EWARM KickStart Edition and dual MC firmware by choosing an operating configuration that allows the code size to not exceed 32 K. For example, if each of the two motors use the encoder sensor to sense the rotor speed (rather than using a sensorless algorithm) building the code is not an issue. Please refer to [Section 3.4](#) for how to perform this customization.

2.1.7 Connecting the STEVAL-IHM022V1 to the MB459B power stage board

- Connect each 24 V / 5 A DC power supply to the J3 connector (pins 2 and 3) of the relevant MB459 power board.
- Connect each Shinano motor to the J5 connector (pin 1, 2 and 3) of the relevant MB459 power board.
- Connect the MC_Main connector CON8 of the STEVAL-IHM022V1 board to the J7 connector of one MB459. Thereafter, this power board is referred to as the MAIN_MOTOR power board and the relative motor simply as the MAIN_MOTOR. Supply the voltage to the MAIN_MOTOR power board.
- Connect the CON9 MC_Sub1 connector of the STEVAL-IHM022V1 board to the J7 connector of the second MB459 board. This power board and the relative motor is called, respectively, the SUB_MOTOR power board and the SUB_MOTOR. Supply the voltage to the SUB_MOTOR power board.
- Supply the STEVAL-IHM022V1 board with its dedicated power supply.

2.1.8 Connecting the STEVAL-IHM022V1 to the STEVAL-IHM011V1 power stage board

- Connect each 24 V / 5 A DC power supply to the J6 connector.
- Connect each Shinano motor to the J4 connector.
- Connect the MC_Main connector CON8 of the STEVAL-IHM022V1 board to the J3 connector. Thereafter, this power board is referred to as the MAIN_MOTOR power board and the relative motor simply as the MAIN_MOTOR. Supply the voltage to the MAIN_MOTOR power board.
- Connect the MC_Sub1 connector CON9 of the STEVAL-IHM022V1 board to the J7 connector. This power board and the relative motor is called, respectively, the SUB_MOTOR power board and the SUB_MOTOR. Supply the voltage to the SUB_MOTOR power board.
- Supply the STEVAL-IHM022V1 board with its dedicated power supply (the same power adapter included with STM3210B(E)-EVAL/A demonstration boards can be used).

2.2 How to navigate the system menu

Three basic elements constitute the hardware for the system user interface: a 320 x 240 TFT LCD display, a 5-way (left, right, up, down, selection) micro-joystick and a push-button "Key".

The joystick is primarily used for navigating between the various menu screens, within a screen and between screen items. In addition, the joystick allows the selection and editing of item values.

The push-button "Key" is used to start/stop one or both motors.

During parameter-setting operations, the user can navigate quickly between different menu screens by moving the joystick key to the left or right.

At the first system power-up, the idle menu screen is displayed (*Figure 1*). Moving the joystick up, down, left or right allows the user to navigate between items.

To move between items within each menu screen, move the joystick up or down. The selected item appears in red text.

To change an item value, first select it and then press the joystick selection key (joy-sel).

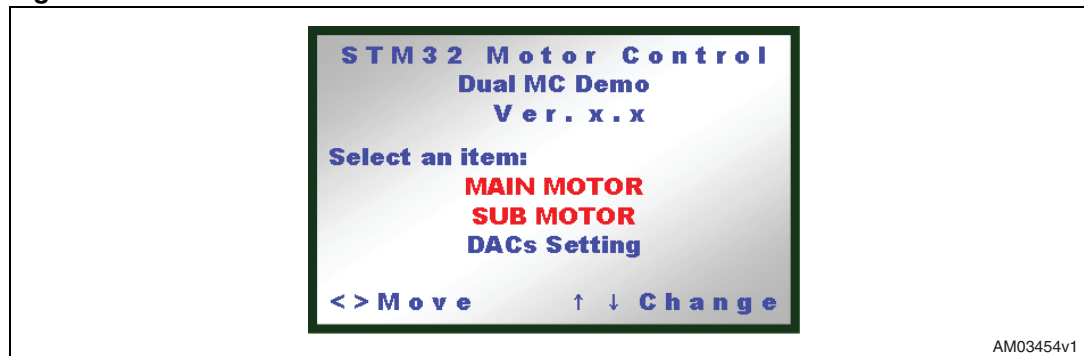
To start the motor (or both motors), press the “Key” button. To exit from the submenu, move the joystick upwards when the first item of the menu is selected.

2.3 Getting started with the demonstration board

2.3.1 Starting or stopping the motor(s)

After a board reset, if the firmware is correctly loaded into the Flash memory and the power boards are correctly supplied, the main screen is displayed as shown in [Figure 1](#).

Figure 1. Main menu



Note: After a reset, if the screen is blank it means that the firmware is not correctly loaded into memory or that the microcontroller has been kept in halt state by the dongle.

First check if the microcontroller has been kept in halt state by the JTAG dongle. If so, remove the JTAG dongle from the CON6 connector and press the reset button.

If the screen is still blank, it is necessary to repeat the firmware configuration ([Section 2.1.4](#)), and the compiling and loading of the firmware ([Section 2.1.5](#)).

Note: If a fault condition is displayed which indicates a bus undervoltage, it means that no bus voltage is detected by the microcontroller. In this case it is necessary to check the power board settings, the connections and the power supplies.

In this condition the user can choose to start the two motors simultaneously, or each one singularly, by selecting two motors ([Figure 2](#)) or one motor ([Figure 3](#)) and then pressing the “Key” button.

Figure 2. Selections for simultaneous dual-motor start

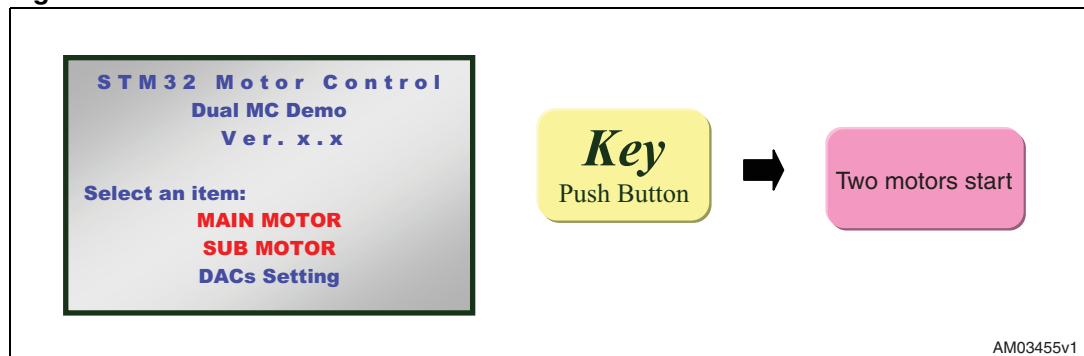
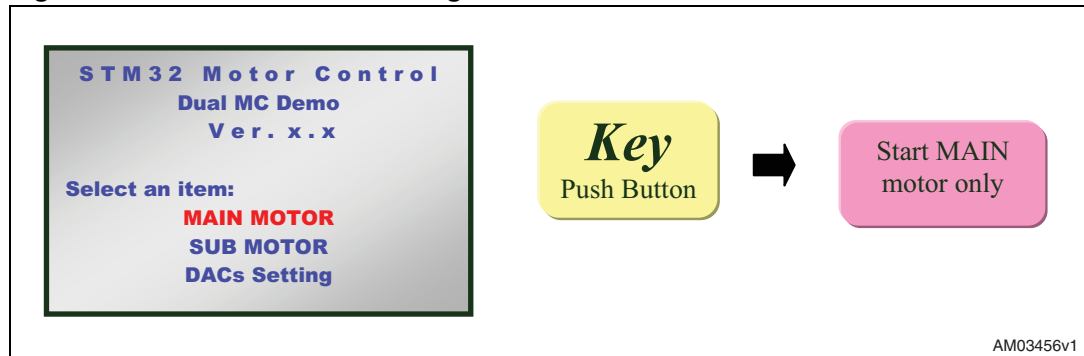


Figure 3. Motor selection for single motor start



To stop the motor(s), press the “Key” push-button again.

2.3.2 Alternate motor starting

It is possible to run the demonstration alternating between the two motors. To do this, first run one motor alone using the steps described above, then select both of the motors and push the “Key” button. This stops the motor that was running, and starts the other motor. So motor starting alternates between the two motors each time the “Key” button is pressed.

2.4 Advanced operations

2.4.1 Setting the control strategy

From the main menu, selecting a single motor and pressing the joystick button provides access a set of menu screens dedicated to the chosen motor’s parameter settings.

The first menu screen displayed allows setting of the control strategy. That is, speed control mode or torque control mode and the related controlled-motor variables, respectively, as well as motor speed and torque (and the flux).

Speed control mode

The menu screen appears as shown in *Figure 4* (for the MAIN motor).

Figure 4. Speed control settings



Moving the joystick downwards selects the target speed (shown in red), and pushing the joystick enables an edit mode where the value can be changed (shown in green). While in

edit mode, moving the joystick up increases the value, and moving the joystick down decreases it. [Figure 5](#) and [6](#) below show these steps.

Figure 5. Selecting the target speed

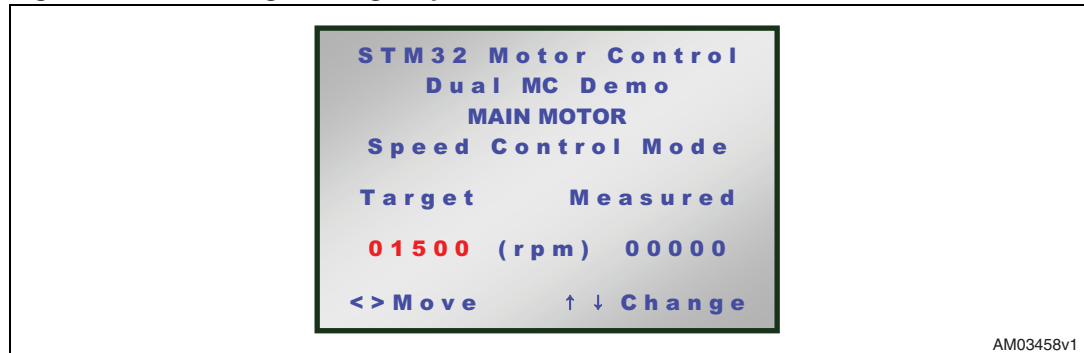
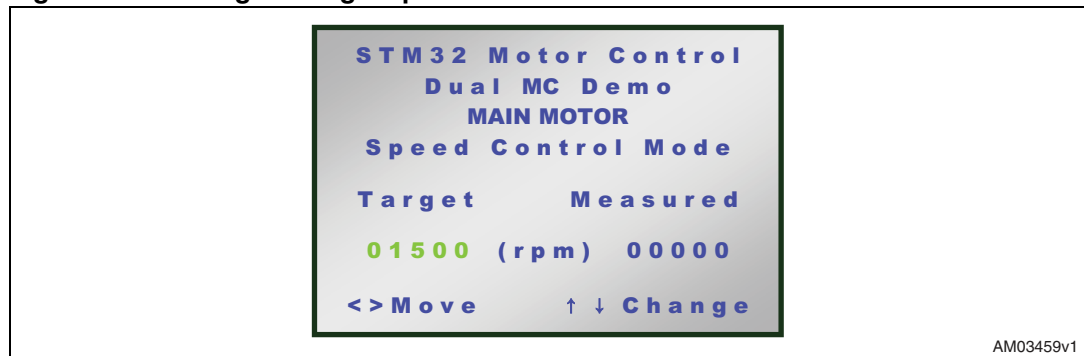


Figure 6. Editing the target speed



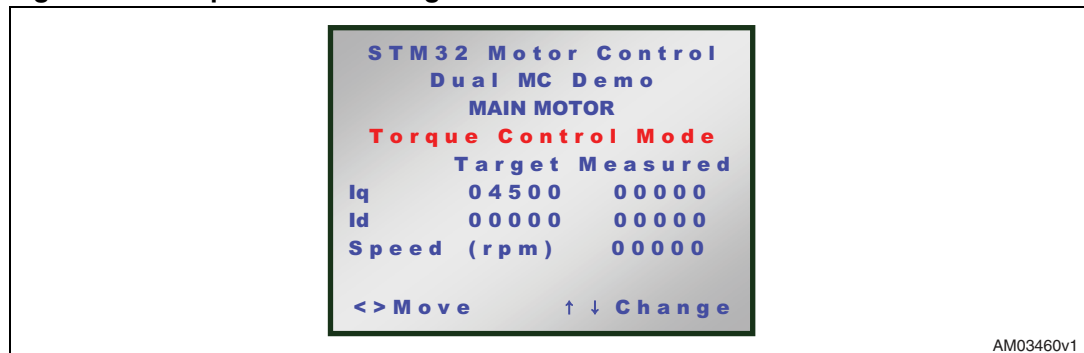
To exit the edit mode, push the joy-sel key again.

Once the target speed has been set, press the “Key” push-button to start/stop the motor (MAIN only). To select other drive parameters (described in the sections that follow), move the joystick right or left. Otherwise, to return to the main menu, move up repeatedly until it is displayed on the screen.

Torque control mode

From the menu screen shown in [Figure 4](#), press joy-sel to switch to Torque control mode.

Figure 7. Torque control settings



To modify the target values for Iq and Id current, select them by moving the joystick up and down, and press to enter edit mode and set their values.

In Torque control mode, the speed value is only displayed to provide feedback on motor speed.

Once the target currents have been set, push the “Key” button to start/stop the motor (the MAIN). Otherwise, to return to the main menu, move up repeatedly until it is displayed on the screen.

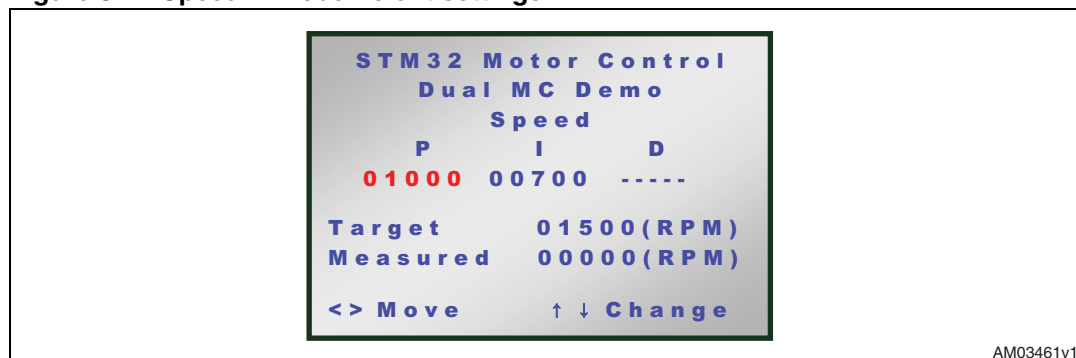
Setting the PID parameters

From the menu screen shown in [Figure 4](#), moving the joystick to the right displays the first PID menu screen. Up to three PID menus are possible when in Speed control mode, and only two when in Torque control mode.

PID menus allow the user to set the typical PID coefficient values, which are the proportional gain (P), the integral gain (I) and, optionally, the derivative gain (D).

When Speed control mode is chosen, the first PID is the one related to speed control (see [Figure 8](#)).

Figure 8. Speed PID coefficient settings



By moving the joystick up or down, it is possible to select the different items within this menu screen and by pressing the joy-sel key it is possible to change their values.

The items called “Target” and “Measured” in [Figure 8](#) (in this case, they refer to speed) are read-only and show how the controlled variable changes according to the variations made on the PID coefficients.

Starting from [Figure 9](#), two menu screens accessed by moving the joystick right, permit the setting of the coefficient values for the torque and flux PIDs. In these two PIDs, the controlled variables are, respectively, the Iq component and the Id component of the target motor current.

Figure 9. Torque PID coefficient settings

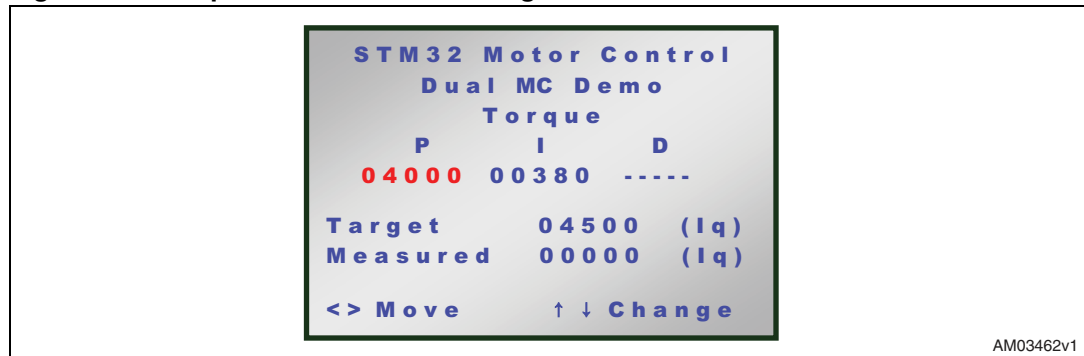
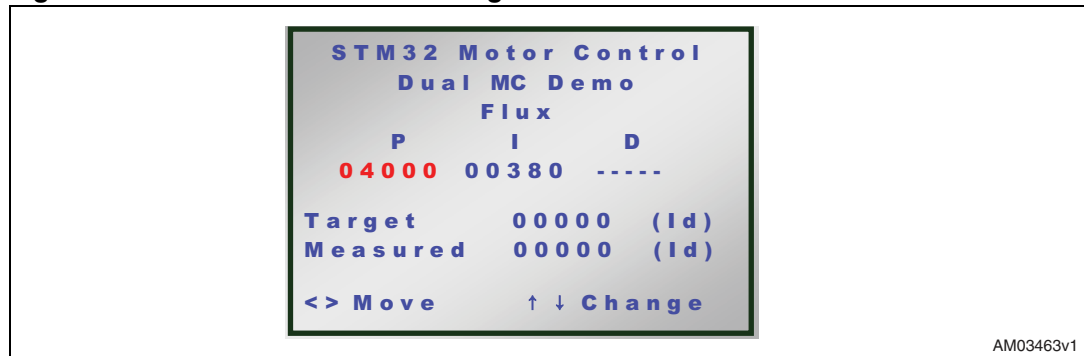


Figure 10. Flux PID coefficient settings

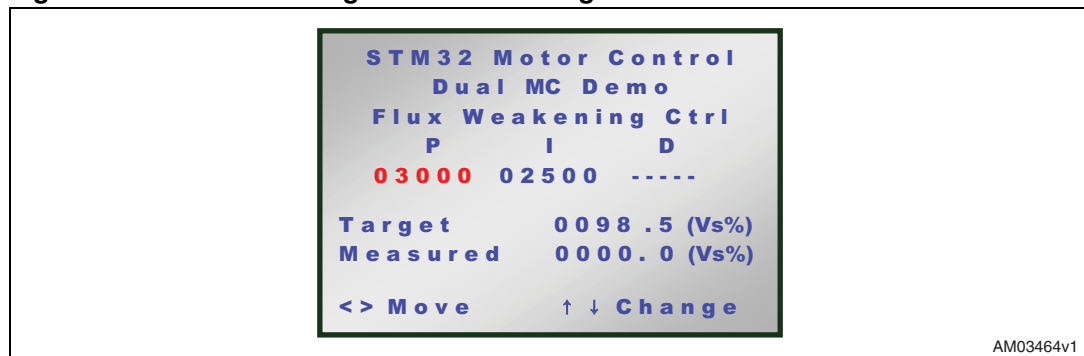


For all three PID screens above, the derivative coefficient (D) is optional depending on the firmware configuration. For more details, please refer to [Table 1](#) or to user manual UM0492 - STM32F103xx permanent-magnet synchronous motor FOC software library V2.0.

2.4.2 Setting the flux weakening parameters

To set the flux weakening parameters, from the menu screen shown in [Figure 10](#) the joy-right key must be pressed. Please note that this menu screen is shown only if the FLUX_WEAKENING option is enabled in the firmware. Refer to [Table 1](#) or to the PMSM library user manual for additional details.

Figure 11. Flux weakening coefficient settings

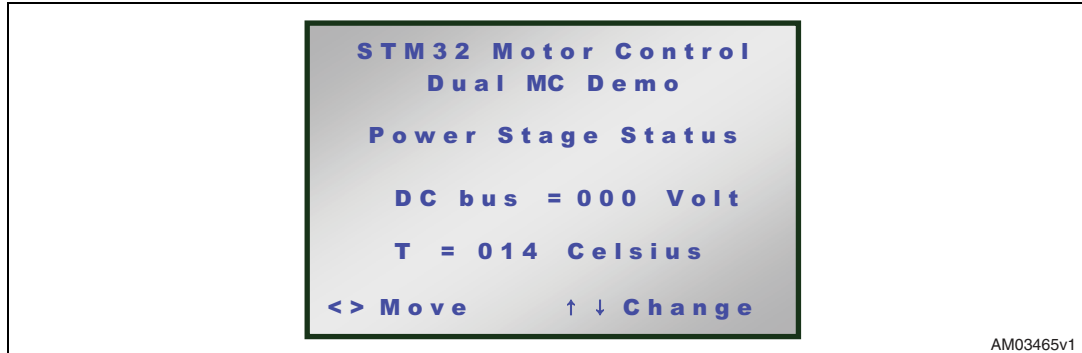


In [Figure 11](#), the item “Target” is also editable and is expressed as a percentage of stator voltage. Refer to [Section 3.4](#) of UM0492 for details.

2.4.3 Viewing the status of the power stage

If the joystick is moved rightwards starting from the screen shown in [Figure 11](#), the power stage status menu screen appears. This allows the monitoring of some critical variables of the power stage board, such as the inverter power switches (IGBT, MOSFET, etc.) temperature and the actual voltage of the DC bus.

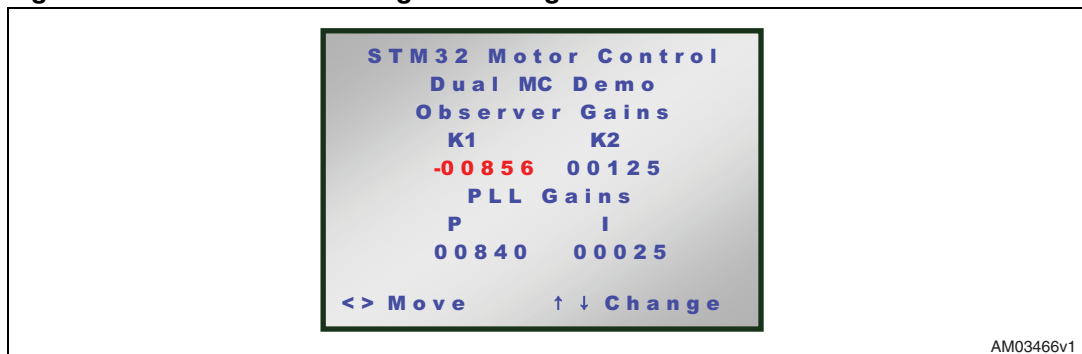
Figure 12. Power stage status variables



2.4.4 Setting the observer and PLL gains

The following menu screen is shown when the joystick is moved to the right from the screen in [Figure 12](#). It allows the handling of all items related to the state observer and those related to PLL. Refer to UM0492 for more information on the meanings of these two items.

Figure 13. Observer and PLL gains setting

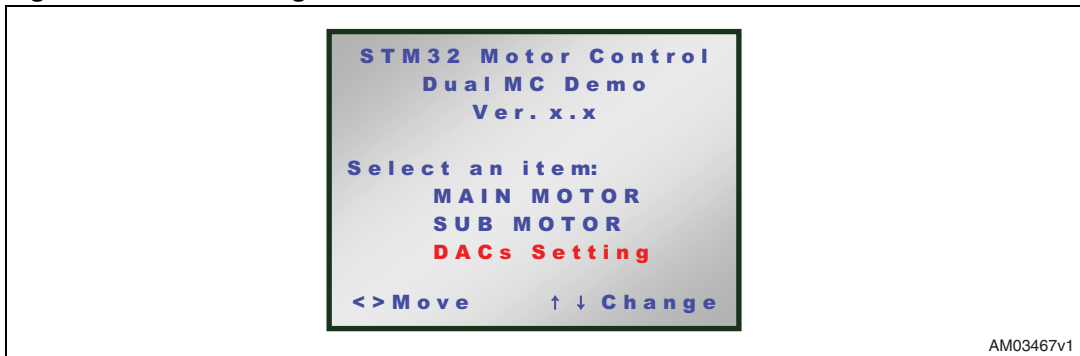


2.5 Debugging mode

For a predetermined set of motor control firmware variables, it is possible to obtain their digital values through two DAC output channels.

The choice of which motor variables to send via DAC is performed by means of the DAC settings menu. It can be accessed from the main menu screen ([Figure 3](#)), by selecting the DAC settings item and then pressing the joy-sel key.

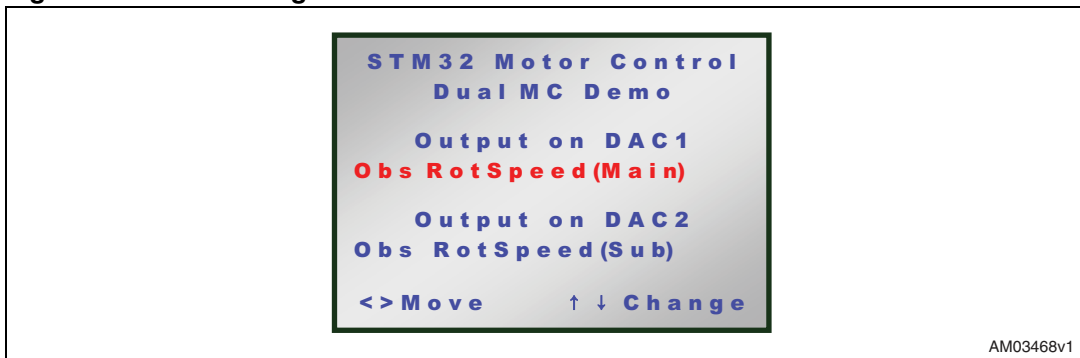
Figure 14. DAC settings menu



AM03467v1

In the following menu screen it is possible to select the variables to be sent to each DAC output.

Figure 15. DAC settings items

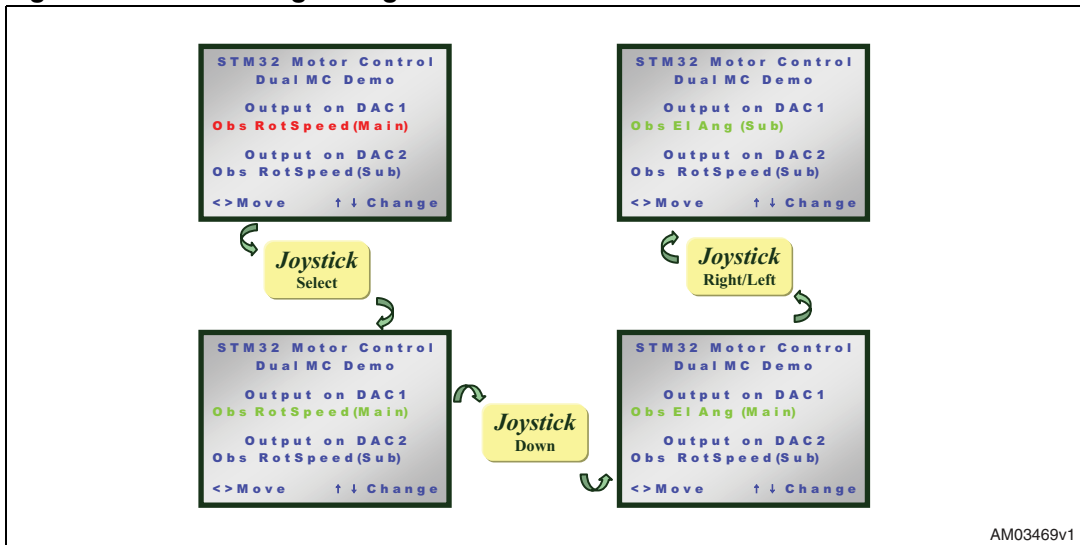


AM03468v1

Once an item has been chosen, the joy-sel key must be pressed to scroll through the list of variables. Move the joystick upwards or downwards to view all the possible variables.

To quickly change the motor with which a variable is associated, move the joystick right or left, and press the joy-sel key to choose the motor (see [Figure 16](#)).

Figure 16. DAC settings navigation menus



AM03469v1

3 Control strategy

3.1 Block diagram

Figure 17. Dual motor control strategy block diagram

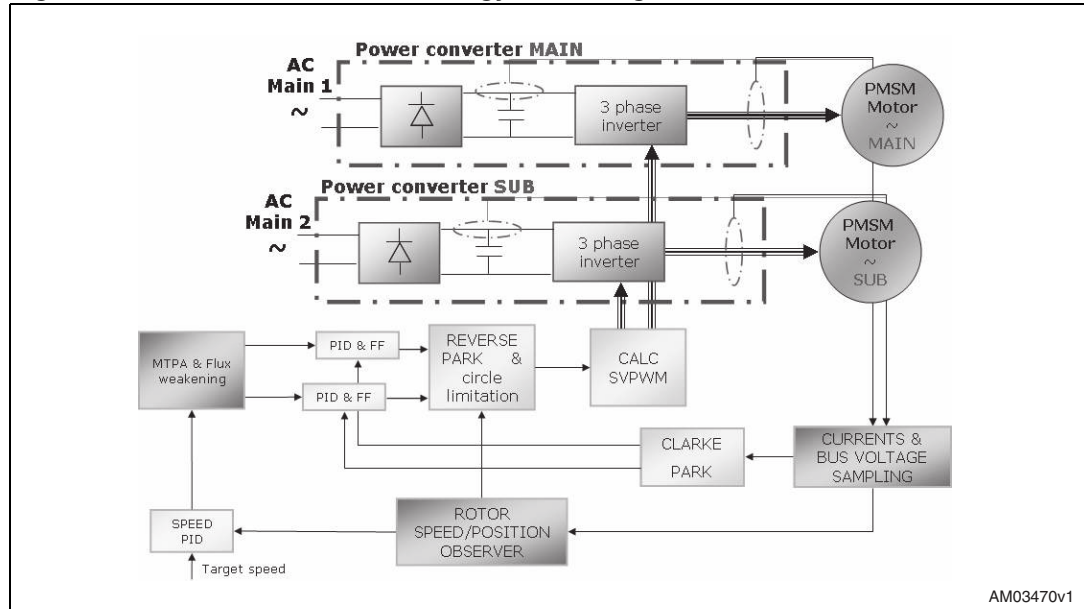


Figure 17 shows the block diagram of the field oriented dual motor control drive.

Phase currents of the Main and Sub motors are sampled each FOC cycle. The strategy adopted for current sampling and FOC algorithm execution is to dedicate one PWM period for each motor, halving the FOC execution rate compared to single motor driving.

Dual motor driving is possible thanks to the two "advanced PWM timers" (TIM1 and TIM8) in the high-density version of the STM32 microcontroller (STM32F103ZE).

These two timers are kept synchronous using the master/slave feature of each timer peripheral present in the microcontroller. This synchronization must be performed because the current sampling must occur in the proper time within each PWM period.

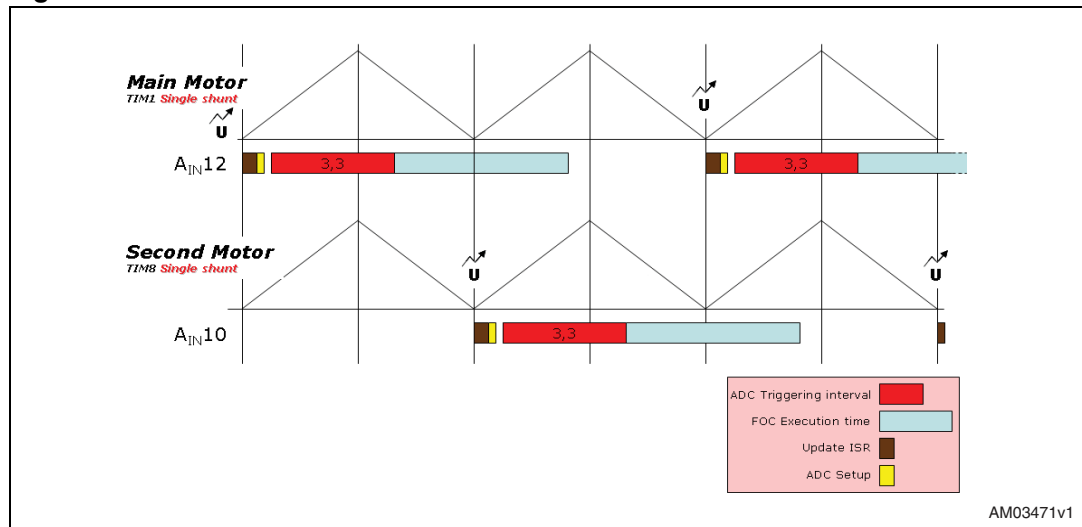
In particular, the single shunt solution implemented using ST's patented methodology waits for two ADC conversions for each motor to sample the phase currents. These two conversions are normally performed within the first half of the PWM period, or at most 3.5 μ s after the half (in the present solution).

For every FOC execution rate (two PWM periods), the values of the phase currents are sampled for one motor and then the FOC algorithm related to that motor is executed. So the currents are transformed with Clarke & Park transformations, torque and flux PID are executed and the voltage demand vector is computed using the reverse Park transformations and circle limitation.

From the voltage demand vector, using the space vector modulation, the values of the three duty cycles are computed to be applied to the inverter and the sampling points for the current conversion. Depending on the selected firmware options, the state observer can be used to estimate the rotor position and speed, as the MTPA (flux weakening and feed forward) can also be optionally executed.

3.2 ADC & FOC execution time diagram

Figure 18. ADC & FOC execution time



In [Figure 18](#) the ADC & FOC execution time diagram is shown. The two triangle-shaped signals are the representation of the two synchronized timer counters.

The two counters are up-counting and down-counting so the 12 PWM output signals are in a centered pattern. Three PWM signals plus the complemented ones go to one power stage to control the main motor, and the other three PWM signals plus the complemented ones go to the other power stage to control the sub motor.

In the diagram, the update instants are indicated with a U for each timer. The update instant is the moment in which the computed values of the duty cycle registers become active.

To allow dual motor control, the update instants of each timer are performed each two PWM periods ($REP\ RATE = 3$) and the updated instants of both timers does not occur at the same time but are shifted by one PWM period.

The trigger instant for starting ADC conversion can occur within the red bar in [Figure 18](#) (ADC triggering interval).

The ADC triggering interval for one timer does not overlap the other, so the samplings can be performed using the same ADC peripheral (ADC3).

Space for both FOC code instruction executions must be guaranteed and those routines must be completed before the corresponding successive update events.

In [Figure 18](#), the following time intervals are indicated:

- ADC triggering interval - The sampling point to convert the motor phase current using the single shunt current reading can range from a few microseconds immediately after the update event up to a few microseconds after half of the PWM period. It depends on several factors which, in turn, affect some parameters, such as the noise parameter, sampling time, and dead time. In the present solution, the last sampling can occur a maximum of $3.5\ \mu s$ after the mid-point of the PWM period.
- Update ISR - This is the fixed time ($5.3\ \mu s$) required to perform the update of the timer registers not performed by the peripheral hardware. These update are required by the

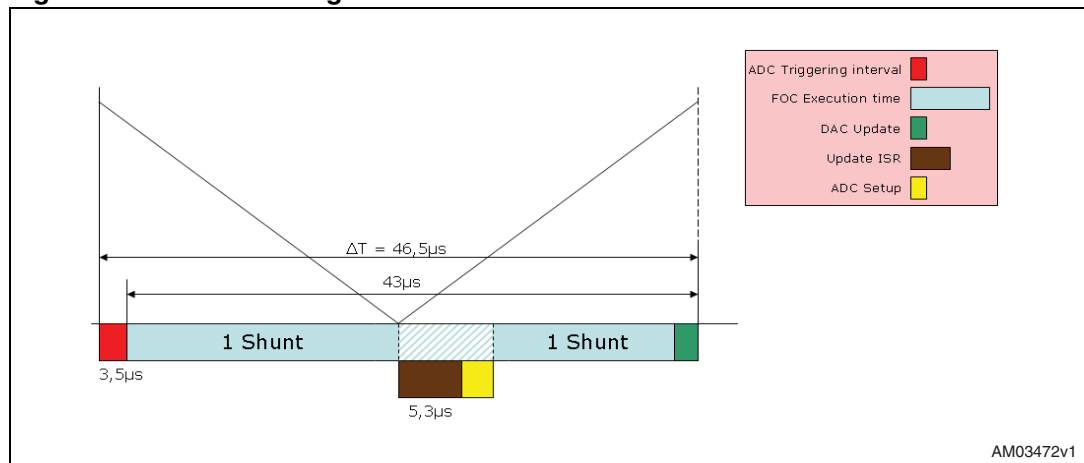
single shunt current reading. This interrupt must be executed with maximum priority so that the FOC execution algorithm can be interrupted by this ISR.

- ADC setup - This is the time required to set up the correct ADC channel for the next conversion.
- FOC execution time - This is the time required to perform the full FOC algorithm (current reading, optional state observer module when in sensorless mode, Clarke & Park, torque & flux PID, optional MTPA module, optional flux weakening, optional feed forward, reverse park & circle limitation, space vector modulation and sampling point definition, and DAC updates). Its duration depends on the motor control functionalities selected.

3.2.1 Workload versus PWM frequency

It is possible to compute the workload of the CPU for a PWM frequency.

Figure 19. Workload diagram



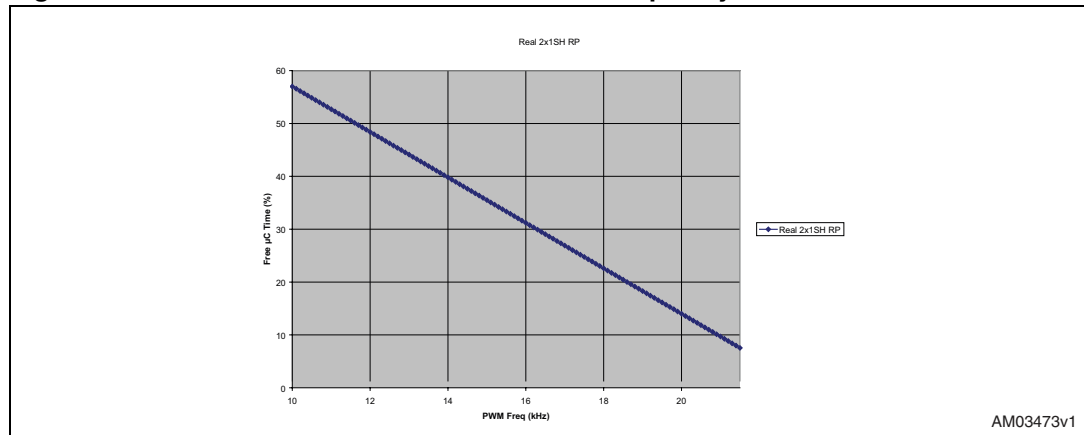
In the present solution the duration of the complete FOC algorithm is $\approx 43 \mu s^{(a)}$.

This algorithm must be executed each PWM period (one period for one motor and the next period for the other motor). So it is possible to compute the maximum PWM frequency applicable, which is 21.5 kHz.

The graph shown in [Figure 20](#) indicates the available CPU load time for other non motor control functions (display, keys, time base, communications) versus the applied PWM frequency.

a. This duration has been measured in an actual implementation under the worst-case conditions (sensorless control, DAC functionality, observer gain tuning and flux weakening enabled) and it includes the update ISR and ADC setup executions.

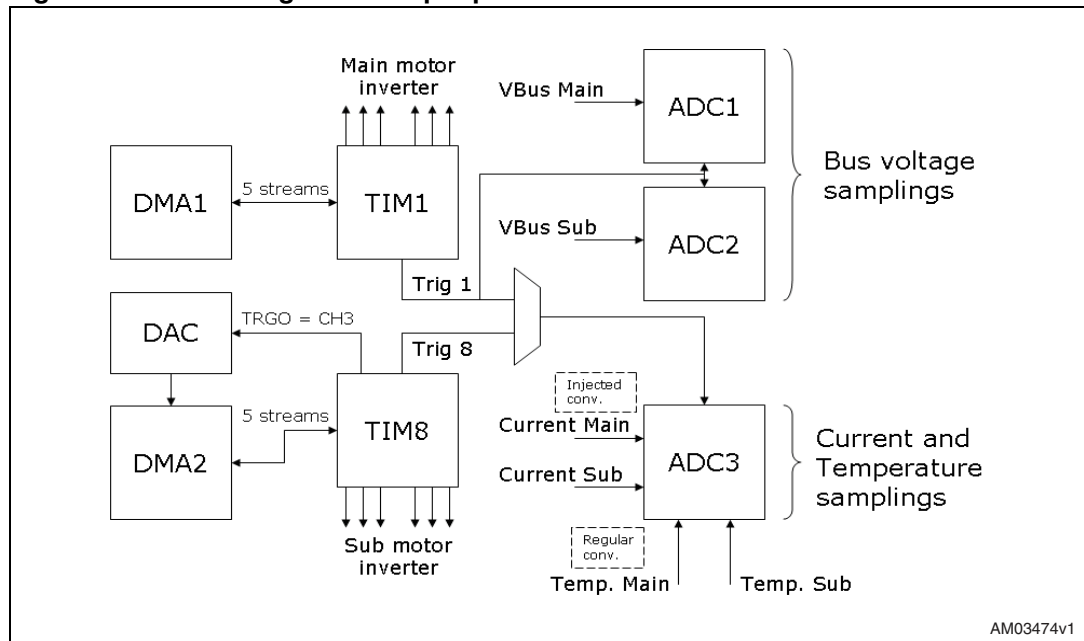
Figure 20. Free microcontroller time vs. PWM frequency



3.3 Block diagram of the peripherals

In Figure 21, the block diagram of the peripherals relative to the implemented dual motor control strategy is shown.

Figure 21. Block diagram of the peripherals



TIM1 and TIM8 are dedicated to the generation of the PWM signals for the power stages and to the generation of the triggering signals for the ADCs. In particular, ADC1 and ADC2 are always triggered by TIM1 TRGO, while ADC3 is triggered by TIM1 CH4 or TIM8 CH4.

ADC1 is dedicated to bus voltage sampling of the main motor using injected conversions, while ADC2 performs the bus voltage sampling of the sub motor.

ADC3 is dedicated to current sampling (injected conversion), while regular conversions are used for sampling the temperature sensor outputs coming from each power stage.

To perform single shunt current reading using the ST patented method, five DMA streams are required (TIMx_CH1, TIMx_CH2, TIMx_CH3, TIMx_CH4, TIMx_UP). The DMA1 controller in conjunction with the TIM1 is used for this purpose.

The DMA2 controller is used in conjunction with the TIM8 for the same purpose.

The DAC peripheral is always in use, even if the DAC_FUNCTIONALITY option is not selected in the firmware configuration. In fact, the DAC peripheral is used to perform the DMA synchronization for the third channel of TIM8.

3.4 Firmware configuration

The dual MC demonstration firmware can be customized by modifying the control drive parameters inside the following header files:

- stm32f10x_MCconf.h
- MC_Control_Param.h
- MC_PMSM_motor_param.h
- MC_State_Observer_param_Multi.h
- MC_PowerStage_Param.h
- MC_pwm_1shunt_prm.h
- MC_encoder_param.h.

The procedure for configuration is very similar to the one relative to the single motor drive firmware library v.2.0 (see UM0492 - STM32F103xx permanent-magnet synchronous motor FOC software library V2.0). The differences in the configuration procedure are described below.

stm32f10x_MCconf.h

In this header file it is possible to customize the options relative to each motor:

Figure 22. Motor options: example 1

```
#define OPTION_MAIN_MOTOR \
(\
/*ENCODER | */\
NO_SPEED_SENSOR | \
OBSERVER_GAIN_TUNING | \
NO_SPEED_SENSORS_ALIGNMENT | \
/*FLUX_TORQUE_PIDs_TUNING | */\
/*VIEW_ENCODER_FEEDBACK | */\
/*VIEW_HALL_FEEDBACK | */\
/*BRAKE_RESISTOR | */\
FLUX_WEAKENING | \
/*DIFFERENTIAL_TERM_ENABLED | */\
/*IPMSM_MTPA | */\
/*FEED_FORWARD_CURRENT_REGULATION | */\
DAC_FUNCTIONALITY \
)
```

AM03475v1

In example 1 (Figure 22), the main motor is configured in sensorless mode, with observer gain tuning, startup alignment, flux weakening and DAC functionality options enabled.

Note: *It is important to remember that each line of the macro define must end with the character "\", no interlines are allowed between #define and the last ")". It is not possible to comment the option with "/" but "/" and "*" must be used and it is not possible to nest the comment "/* /* */ */".*

Figure 23. Motor options: example 2

```
#define OPTION_MAIN_MOTOR \
(\
ENCODER | \
/* NO_SPEED_SENSOR | */\
/* OBSERVER_GAIN_TUNING | */\
/* NO_SPEED_SENSORS_ALIGNMENT | */\
/*FLUX_TORQUE_PIDs_TUNING | */\
/*VIEW_ENCODER_FEEDBACK | */\
/*VIEW_HALL_FEEDBACK | */\
/*BRAKE_RESISTOR | */\
FLUX_WEAKENING | \
/*DIFFERENTIAL_TERM_ENABLED | */\
/*IPMSM_MTPA | */\
/*FEED_FORWARD_CURRENT_REGULATION | */\
DAC_FUNCTIONALITY \
)
```

AM03476v1

In example 2 (Figure 23), the main motor is configured to use the encoder as a speed sensor.

Note: if both motors are configured as shown in example 2, it is possible to compile and build the application code by using EWARM KickStart Edition limited to 32 KB of code size.

The list of all the available options is described in Table 1 below:

Table 1. Available motor options

| Option | Description |
|----------------------------|---|
| ENCODER | Sensored mode. The encoder is used as primary speed sensor. The current control loop is performed using the angle measured by the encoder |
| HALL | Not available in dual MC demonstration SW ver. 1.0 |
| NO_SPEED_SENSOR | Sensorless mode. The state observer is used as primary speed sensor. The current control loop is performed using the angle estimated by the observer |
| OBSERVER_GAIN_TUNING | With this option it is possible to change the state observer and PLL parameters in run-time. If the encoder is selected as the primary speed sensor, this option enables the state observer as auxiliary speed sensor |
| NO_SPEED_SENSORS_ALIGNMENT | This option is used to perform the alignment before the startup when sensorless mode is selected. Is not possible to set it in sensed mode |
| FLUX_TORQUE_PIDs_TUNING | This option is used to enable the modality of PID tuning. See UM0492 for details. Is not possible to set it in sensorless mode |
| VIEW_ENCODER_FEEDBACK | If the state observer is selected as the primary speed sensor, this option enables the encoder as auxiliary speed sensor. Is not possible to set it in sensed mode |
| VIEW_HALL_FEEDBACK | Not yet available |
| BRAKE_RESISTOR | This option enables brake resistor management when overvoltage occurs |
| FLUX_WEAKENING | This option enables the flux weakening feature |
| DIFFERENTIAL_TERM_ENABLED | This option enables the differential term for all PID in the firmware |
| IPMSM_MTPA | This option enables the MTPA feature |

Table 1. Available motor options (continued)

| Option | Description |
|---------------------------------|--|
| FEED_FORWARD_CURRENT_REGULATION | This option enables the feed-forward current regulation feature |
| DAC_FUNCTIONALITY | This option enables DAC functionality. This means the selected two motor drive variables are available on DAC outputs every FOC control loop cycle |

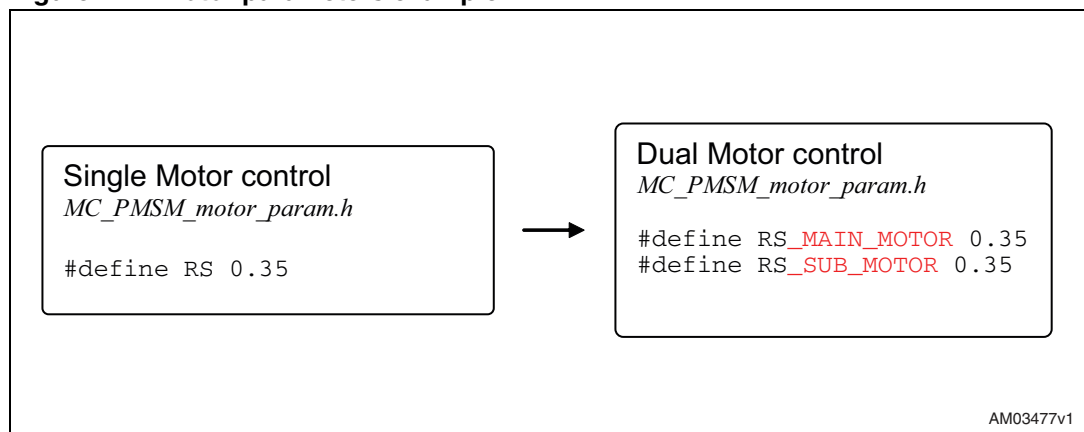
Configuring the other parameter (.h) files

As described in UM0492, it is possible to configure the field oriented control (FOC) parameters. Each file is related to a particular part of the system (see [Table 2](#)).

Table 2. Parameter (.h) files

| Parameter (.h) files | Description |
|---------------------------------|---|
| MC_Control_Param.h | PWM frequency, dead time, max modulation index, PID parameters and dividers |
| MC_PMSM_motor_param.h | Motor-related characteristics: Rs, Ls, Ke |
| MC_State_Observer_param_Multi.h | Parameter related to the state observer: K1, K2, dividers, startup values |
| MC_PowerStage_Param.h | Parameter related to the power stage used: ADC conversion ratio, bus overvoltage, undervoltage and hysteresis |
| MC_pwm_1shunt_prm.h | Noise parameters, ADC settings for current sampling, bus voltage and temperature sensing |
| MC_encoder_param.h | Encoder-related parameters: DPP, max speed, startup values |

Figure 24. Motor parameters example



The name of the motor (MAIN_MOTOR or SUB_MOTOR) is concatenated to the name of the parameter, as in the example in [Figure 24](#), for all parameters that must be differentiated between one motor and the other.

4 Revision history

Table 3. Document revision history

| Date | Revision | Changes |
|-------------|----------|------------------|
| 01-Sep-2009 | 1 | Initial release. |

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2009 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com