AN3165
Application note

Digital PFC and dual FOC MC integration

Introduction

This application note explains how to integrate two sets of firmware to manage a digital PFC and a dual field-oriented control (FOC) motor control driver by means of a high-density STM32.

The purpose is to evaluate the potentiality of an STM32 to control a high-power PFC with performances comparable to a standard continuous mode PFC monolithic IC, while allotting sufficient micro resources (such as program memory and CPU computational capabilities) to make other complex operations such as the simultaneous driving of two 3-phase, field-oriented control motors in sensorless and single-shunt mode.

Section 2 and *Section 3* briefly describe the implementation of a digital PFC and dual motor control FOC with an STM32, while *Section 4* describes how to integrate these two parts in a single firmware with the main focus on the use of the STM32 resources and constraints.

Contents

List of tables

List of figures

1 Safety and operating instructions

1.1 General

During assembly and operation, the PFC power board poses several inherent hazards, including bare wires, moving or rotating parts and hot surfaces. Serious personal injury and damage to property may occur if the kit or its components are used or installed incorrectly.

All operations involving transportation, installation, use and maintenance should be carried out by skilled technical personnel (national accident prevention rules must be observed). "Skilled technical personnel" refers to suitably qualified persons who are familiar with the installation, use and maintenance of power electronic systems.

Warning: The board operates directly from the mains, is not galvanic insulated, and provides high voltage DC levels at the output that can cause serious electric shock, burns or death. Hot surfaces on the board can also cause burns.

This board must only be used in a power laboratory by engineers and technicians who are experienced in power electronics' technology and with adequate protection. STMicroelectronics shall not be considered responsible for damages to equipment or persons.

1.2 Intended use of the demonstration board

The system is designed for demonstration purposes only, and must not be used for electrical installations or machinery. Technical data and information concerning the supply conditions must be taken from the documentation provided and strictly observed.

1.3 Installation of the demonstration board

The system's installation and cooling must be in accordance with the specifications and targeted application.

- Excessive strain on the board must be avoided. In particular, no components are to be bent, or isolating distances altered, during the course of transportation or handling.
- No contact must be made with other electronic components and contacts.
- The board contains electro-statically sensitive components that are prone to damage through improper use. To avoid potential health risks, ensure that the electrical components are not damaged in any way.

1.4 Electronic connection

National accident prevention rules must be followed when working on the main power supply with another power supply or power board in general.

The electrical installation must be carried out in accordance with the appropriate requirements (cross-sectional areas of conductors, fusing, PE connections, etc).

1.5 Board operation

It is advised to use an AC insulated and protected against overloads and short-circuits during the evaluation test of the system (compliance with technical equipment and accident prevention rules).

A correct load able to dissipate, or in any case absorb and reuse, the power delivered by the system must be used. In the case of a resistive and dissipative dummy load, attention should be given to the temperature that the load could reach. Ensure the necessary equipment is provided to avoid hot surfaces and risk of fire during the tests (fan, water cooled load, etc).

Note: Do not touch the board or its components after disconnection from the voltage supply as several parts and power terminals which contain possibly energized capacitors need to be given time to discharge.

2 STM32 digital PFC

2.1 Introduction

A power factor correction (PFC)–also known as a power factor controller–is a feature that reduces the amount of reactive power generated by a non-linear load. Loads such as electrical motors distort the current drawn from the system and, in such cases, a power factor correction may be used to counteract the distortion and raise the power factor.

Reactive power operates at right angles to true power and energizes the magnetic field. Reactive power has no real value for an electronic device, but electric companies charge for both true and reactive power, resulting in unnecessary charges. PFC is a required feature for power supplies shipped to or within Europe.

In a PFC, the power factor is the ratio of the true power divided by the reactive power. The value of the power factor is between 0 and 1. If the power factor is above 0.8, the device is using power efficiently. A standard power supply has a power factor of 0.70 to 0.75, and a power supply with PFC has a power factor of 0.95 to 0.99.

PFC equipment is used to reduce the reactive power produced by fluorescent and high bay lighting, arc furnaces, induction welders and equipment that uses electrical motors.

2.2 System overview

This demonstration board implements a digital control for a high-power PFC controlled by an STM32. It has been designed to offer high performances in terms of PF, THD and DC output voltage ripple.

Contrarily to monolithic ICs, this digital approach facilitates the application of a sophisticated control algorithm and makes it easier to adjust system parameters to meet customer requirements.

The STM32 digital PFC hardware system is composed of two boards: a digital PFC board (STEVAL-ISF002V1) that implements the boost stage of the PFC, and a control board (STEVAL-IHM022V1) based on the STM32F103ZE microcontroller that implements the control stage of the PFC.

The digital PFC board can be connected through an MC connector to several evaluation kits available from STMicroelectronics, in particular those designed for motor control.

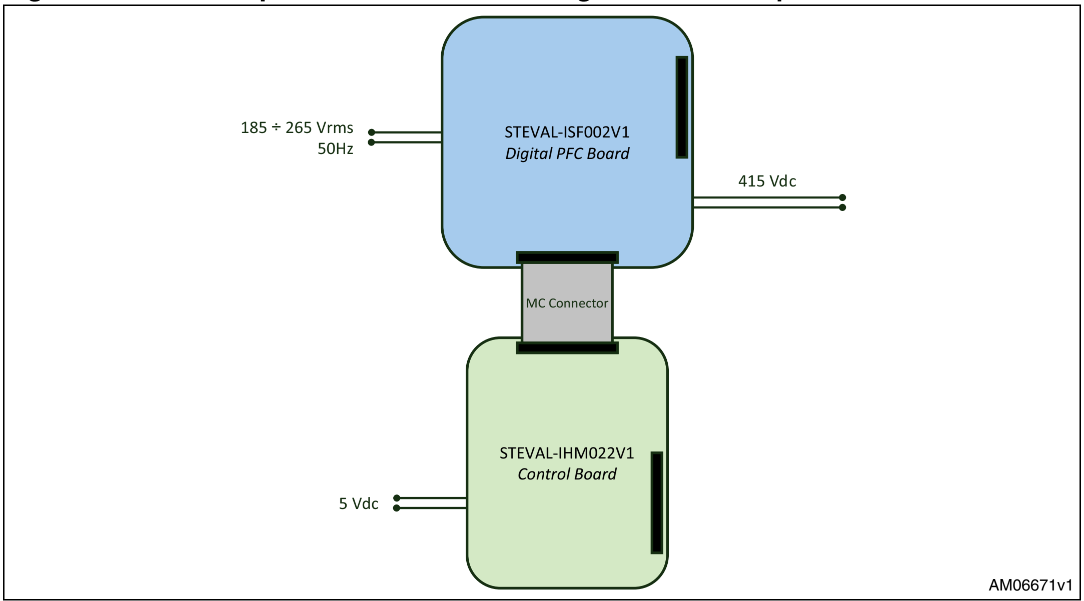
An on-board OFF-line switched mode power supply (SMPS) based on the VIPER12 is used to generate the 15 VDC voltages necessary to supply the drivers inside the power board. This board provides 5 volts to any control stage supplied via the MC connector.

Note: Refer to user manual UM0877 for a description of the STM32 digital PFC and an application example.

- Main system features
  - Maximum output power:1400 W
  - Input voltage range: 185 ÷ 230 Vrms / 50 Hz
  - Output voltage: 415 Vdc / 5% ripple
  - PF up to 0.998 (at nominal rated power)

- THD between 0.9% and 9% within entire operating range
- Boost topology for DC to DC converter
- Continuous conduction mode for PFC
- Switching frequency of 80 kHz
- Control loop frequency of 40 kHz
- Hardware overcurrent protection (14.3 A)
- Software current limitation (13 A)
- Software overvoltage protection (460 Vdc)
- Software voltage limitation (435 Vdc)
- Regulated DC output voltage with zero load
- Adjustable target value of output DC voltage (by firmware)
- Adjustable proportional and integral parameters for voltage and current (by firmware)

Figure 1. Block representation of STM32 digital PFC concept



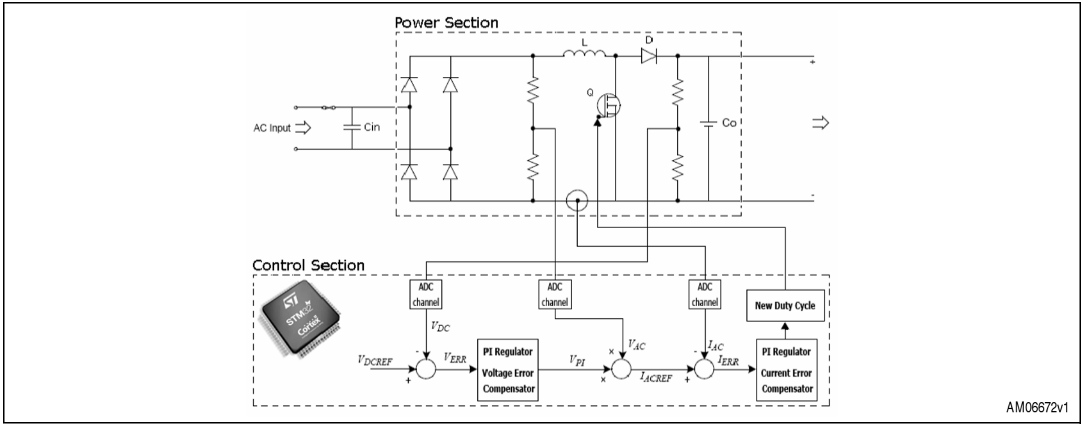
2.3 System architecture

To perform a digital power factor correction, the MCU of the control stage needs three input signals.

- Output DC voltage
- Input AC voltage
- Inductor current

From these inputs, the MCU control software modulates the duty cycle of the switching signal applied to the gate of the MOSFET transistor so that the AC input current is in phase with the input AC voltage. Moreover, the control strategy keeps the output DC voltage regulated at a stable value (target output reference voltage).

Figure 2. STM32 digital PFC: connections for the boost and control stages



The "voltage error compensator" regulates the output DC voltage at the target reference VDCREF. Its output is then used as a scaling factor for the input Vac. This product constitutes the current reference input IACREF for the "current error compensator", operating at 40 kHz. The output of this last PI is the actual duty cycle applied at the gate of the power MOSFET transistor Q.

The "voltage error compensator" uses a frequency of 100 Hz, in line with the Vdc ripple that has this frequency.

The "current error compensator" uses a frequency of 40 kHz, which is the frequency at which the system gets the new converted values of all the necessary signals.

2.4 STM32 peripheral utilization

The following peripherals are used to implement the digital PFC.

- TIM3: its frequency is fixed at 80 kHz. CH4 is used to drive the PFC MOSFET whereas CH3 is used as the start trigger for ADC1_2.
- ADC1: converts the output DC voltage (Vdc) and inductor current (Iac) alternatively.
- ADC2: converts a dummy value and input AC voltage (Vac) alternatively. ADC2 is set as the slave of ADC1 for simultaneous conversions. The dummy value is replaced with the sub-motor bus voltage when this firmware is merged with the MC firmware.
- DMA1: stores the converted values by means of its CH1. As soon as all values have been converted (with a frequency of 40 kHz), an IRQ is generated and the PFC routine is executed.
- EXTI_LINE1: retrieves overcurrent information from the power section. An IRQ is generated and the digital PFC is stopped if an overcurrent condition is detected.

Figure 3. Use of peripherals for STM32 digital PFC

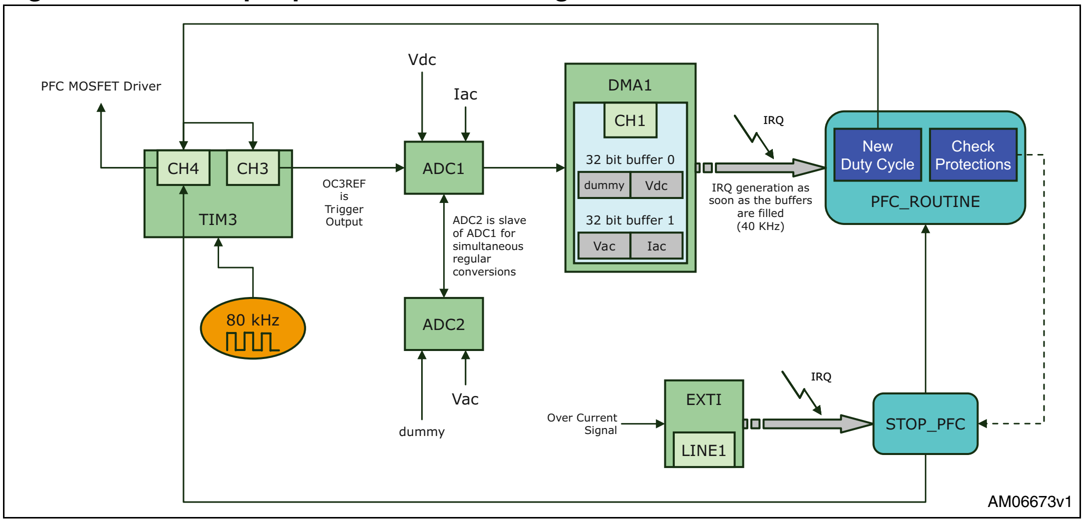


Table 1 describes the STM32 pins and their purpose.

Table 1. STM32F103ZE pin description

MCU pin	Description	MC + PFC connector
PA.03	Vdc – PFC output DC voltage	14
PA.04	Iac – PFC current	24
PA.05	Vac – input AC voltage	22
PC.09	PFC power MOSFET driver	29
PD.02	Vac zero-crossing detection	27
PD.10	Drives the relay to bypass the resistor when there is in-rush current at start up	21
PE.01	PFC hardware overcurrent detection	2

2.5 Timing

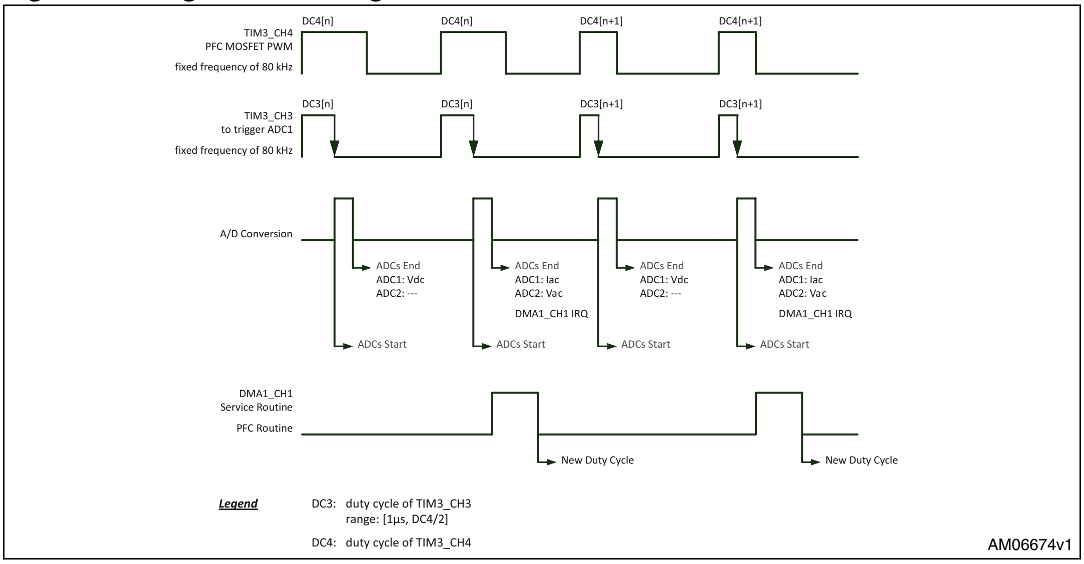
The signal output from TIM3_CH4 is applied to the power MOSFET gate and its frequency is fixed at 80 kHz; its duty cycle varies and is linked to the control strategy of the digital PFC.

TIM3_CH3 is used to trigger ADC1: a conversion is started at the end of each ON period.

The duty cycle of TIM3_CH3 is equal to half that of TIM3_CH4 but never lower than 1 µs to avoid invalid conversions due to noise generated by the switching of the power MOSFET.

Figure 4 shows the triggering mechanism of TIM3, ADC and DMA1.

Figure 4. Digital PFC timing



2.6 CPU load and memory size

Through experimental measurements, the CPU (operating at 72 MHz) takes 4.27 µs to execute the code of the "PFC Routine". By referring this time to a PFC control loop time of 25 µs (40 kHz), the CPU load can be computed as:

Equation 1

$$\text{CPU load} = \frac{4.27\ \mu s}{25\ \mu s} \cong 17\%$$

Table 2 reports the size of the PFC.o object module in terms of Flash and RAM memory sizes.

Table 2. STM32 digital PFC module summary

Read-only code (Flash)	Read-only data (Flash)	Read/write data (RAM)
2088	214	100

3 STM32 dual motor field-oriented control

3.1 Overview

The firmware running on the STEVAL-IHM022V1 demonstration board performs 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.

To begin with, the dual motor control firmware uses the FOC routines of the STM32 PMSM library version. 2.0 firmware package and, hence, shares the same principles as when the motor drive is configured with user parameters.

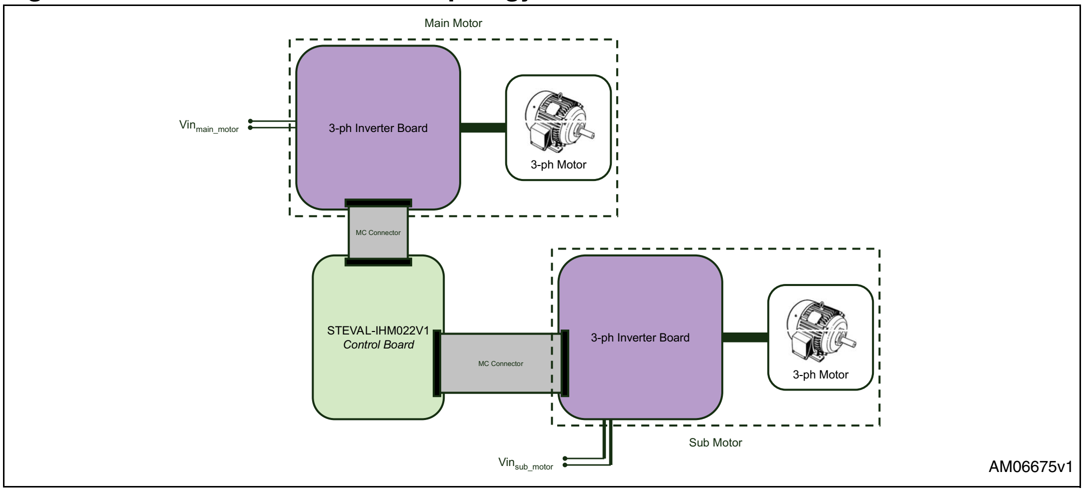
The software architecture has been extended to treat each motor as an independent instance, splitting the controls for each one in a completely independent manner.

To emphasize this concept, an embedded UI (LCD TFT 320 x 240 display and 5-position joystick) allows the user to adjust the motor control parameters in real time during the motors' operation.

The following figure shows a typical connection scheme between the STEVAL-IHM022V1 board and two inverter stage boards for performing simultaneous two-motor FOC control.

Note: For more information, refer to UM0683, UM0686 and UM0688 user manuals[a].

Figure 5. STM32 dual FOC MC topology



3.2 Dual field-oriented control motor driver strategy

Figure 6 shows the block diagram of the dual motor FOC mechanism.

The phase currents of the main and sub motor are sampled during each FOC cycle. The strategy adopted for the current sampling and execution of the FOC algorithm is to dedicate one PWM period to each motor, halving in this way the execution rate of the FOC with respect to single motor driving.

a. See *Chapter 5: References*.

Dual motor driving is possible by way of two advanced PWM timers (TIM1 and TIM8) inside the high density version of the STM32 microcontroller (STM32F103xC-D-E).

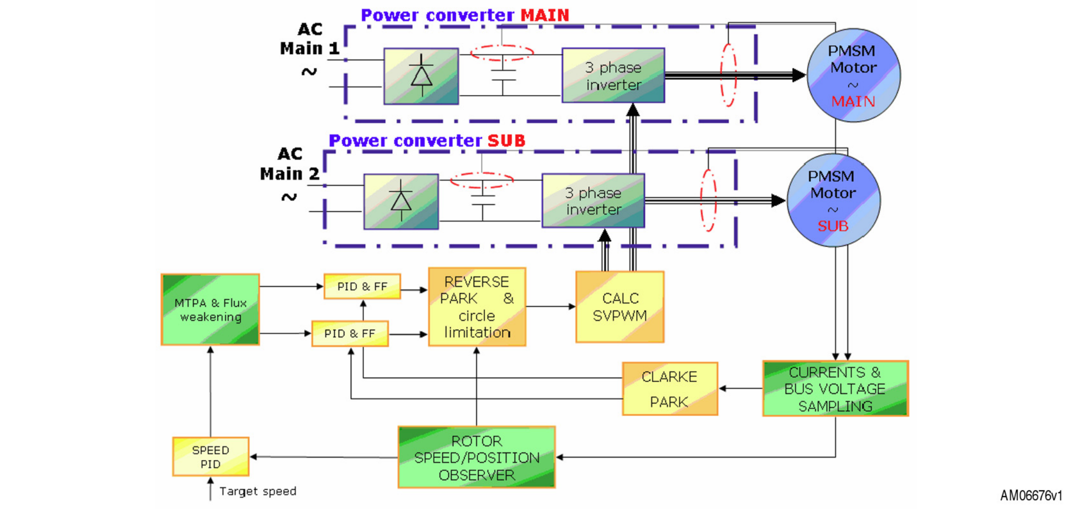
These two timers are kept synchronous using the master/slave feature of each timer peripheral present in the microcontroller. The timers must be synchronized to ensure that the current sampling occurs during the proper PWM period.

The single shunt solution implemented with the ST patented method expects 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 µs after that time.

During each FOC execution rate (two PWM periods), the values of the phase currents are sampled for one motor and the FOC algorithm related to that motor is executed. The currents are therefore transformed with Clark and Park transformations, torque and flux PID are executed and the voltage demand vector is computed using the reverse park transformations and circle limitation.

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

Figure 6. Control strategy block diagram for STM32 dual FOC MC



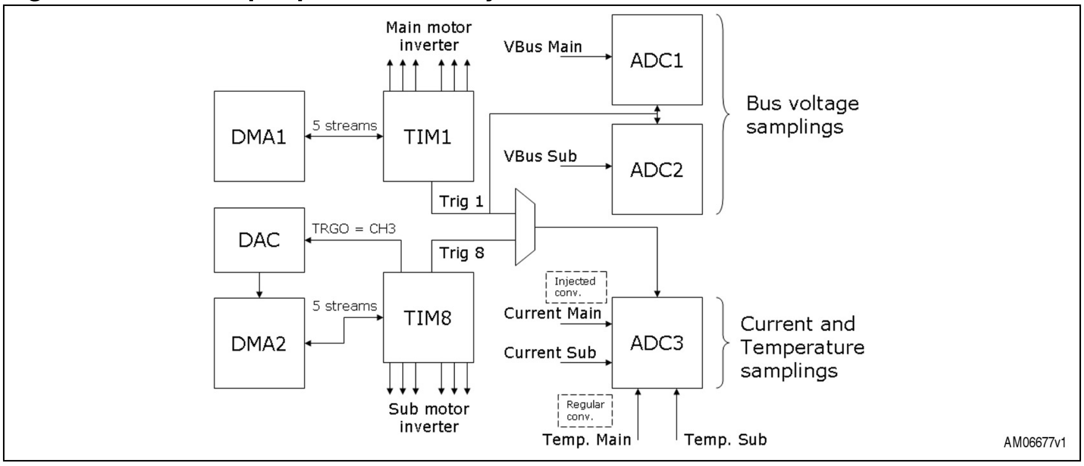
3.3 Peripherals

The following peripherals are used to implement the dual FOC MC.

- TIM1: generates the PWMs for controlling the main motor currents. Also triggers the start of ADC1, ADC2 and ADC3.
- TIM8: generates the PWMs for controlling the sub-motor currents. Also triggers the start of ADC3.
- ADC1: converts the bus voltage of the main motor.
- ADC2: converts the bus voltage of the sub motor.
- ADC3: converts the currents and temperature of the main or sub motor.
- DMA1: performs the single shunt for the main motor.
- DMA2: performs the single shunt for the sub motor.
- DAC: used for debugging.

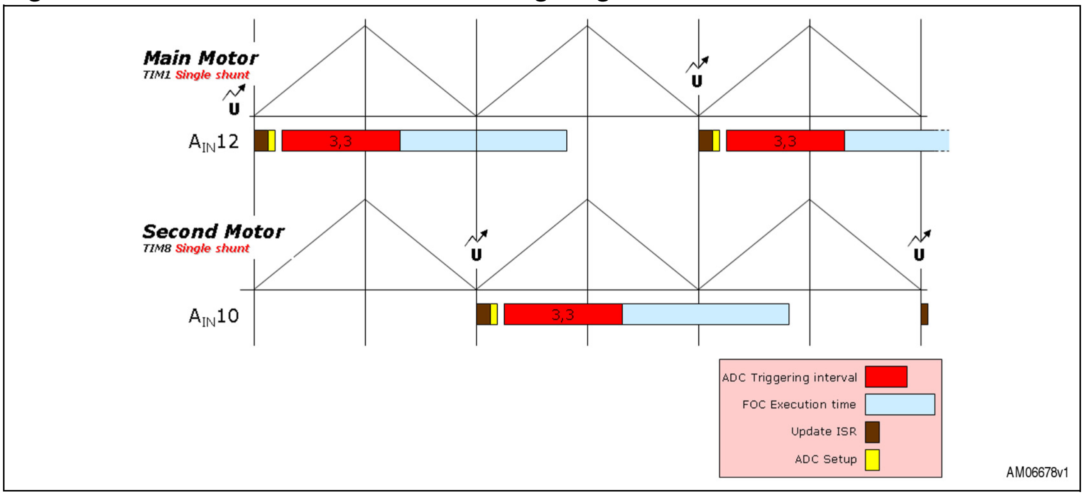
The following figure shows the STM32 peripherals used by the dual FOC MC firmware.

Figure 7. STM32 peripherals used by dual FOC MC



3.4 Timing

Figure 8. ADC and FOC execution timing diagram



The two triangular-shaped signals represent the two synchronized timer counters. The update points for each timer are indicated with a U. The update point is the moment at which the computed values of the duty cycle registers become active.

To allow dual motor control, each timer is updated every two PWM periods (REP RATE = 3), but not at the same time (each update is shifted by one PWM period).

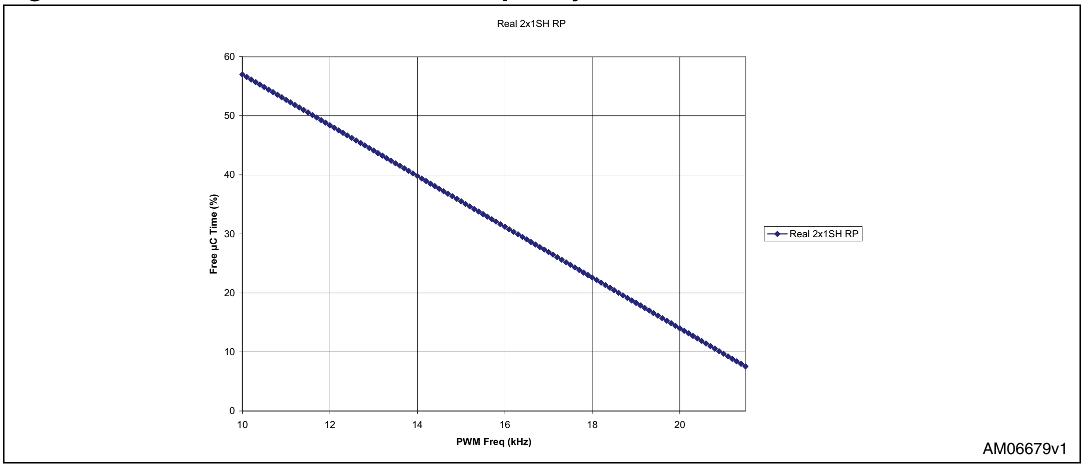
The trigger point for the ADC conversion occurs during the ACD triggering interval, depicted by a red bar in *Figure 8*.

The ADC's triggering interval related to a specific timer does not overlap the other, so the samplings can be performed using the same ADC peripheral (ADC3).

Space for both FOC instructions must be guaranteed and the routines completed before the next corresponding update event.

3.5 CPU load and memory size

Figure 9. Free MCU time vs PWM frequency in dual FOC MC



In the released code for the "STM32 dual FOC MC software demonstrator" the frequency of the PWM is set to 12 kHz. Therefore, according to *Figure 9*, the CPU load is approximately 52%.

Table 3. STM32 dual FOC MC module data summary

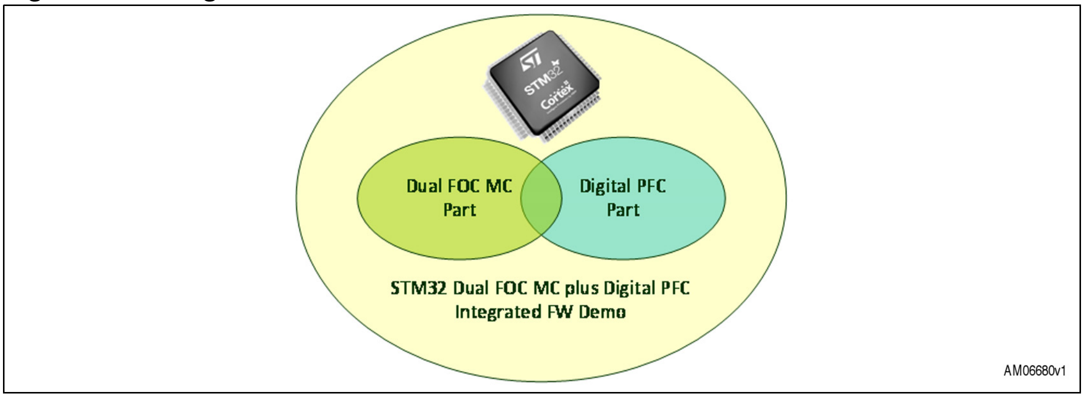
Read-only code	Read-only data	Read/write data
20932	1356	826

4 Integration principles and description

4.1 Aim

The goal of this integration firmware is to merge the two sets of firmware described previously into one single set that will manage both the dual FOC MC and the digital PFC through one single STM32 MCU.

Figure 10. Integrated firmware elements



4.2 Resource constraints

Several elements must be checked before the firmware can be integrated.

- Availability of CPU load
- Conflicts between peripherals
- Conflicts between ports
- IRQ priorities

4.2.1 CPU load

From the findings described in *Section 2.6* and *Section 3.5*, it has been demonstrated that there is sufficient CPU load available. The "STM32 Digital PFC" has a CPU load of 17% at 40 kHz, while it is of 58% at 12 kHz for the "STM32 dual FOC MC software demonstrator".

4.2.2 Conflicts between peripherals

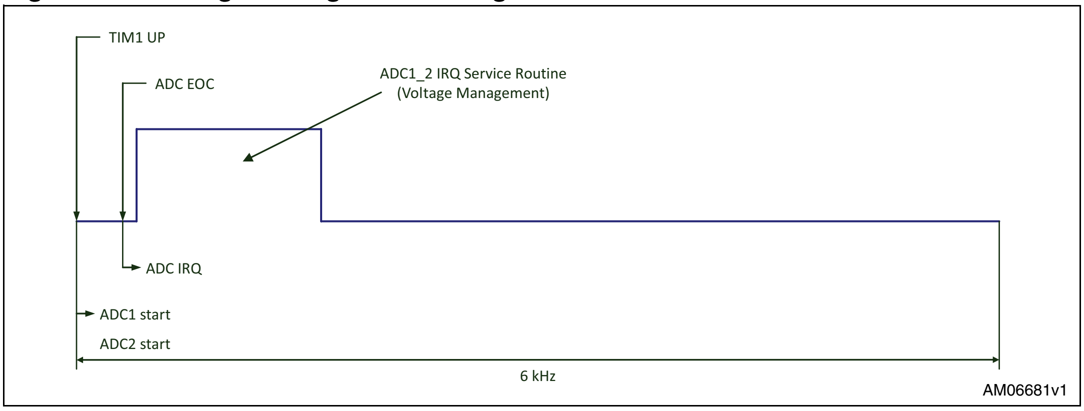
From *Section 2.4* and *Section 3.3*, it is deduced that both firmware sets use ADC1 and ADC2. *Table 4* explains their use.

Table 4. Use of A/D converters by both firmware sets

Control loop frequency	Digital PFC		Dual FOC MC
	40 kHz		6 kHz
	Regular conversions		Injected conversions
	1st conversion group	2nd conversion group	Every conversion
ADC1	VDC	IAC	Vin_{main_motor}
ADC2	Dummy	VAC	Vin_{sub_motor}

The data in *Table 4* suggests maintaining the PFC firmware sampling strategy and adding the sub-motor voltage bus conversion in the first group in place of the dummy value. It is necessary to specify when the firmware of the MC part has to use these values.

Figure 11. Voltage management timing in STM32 dual FOC MC

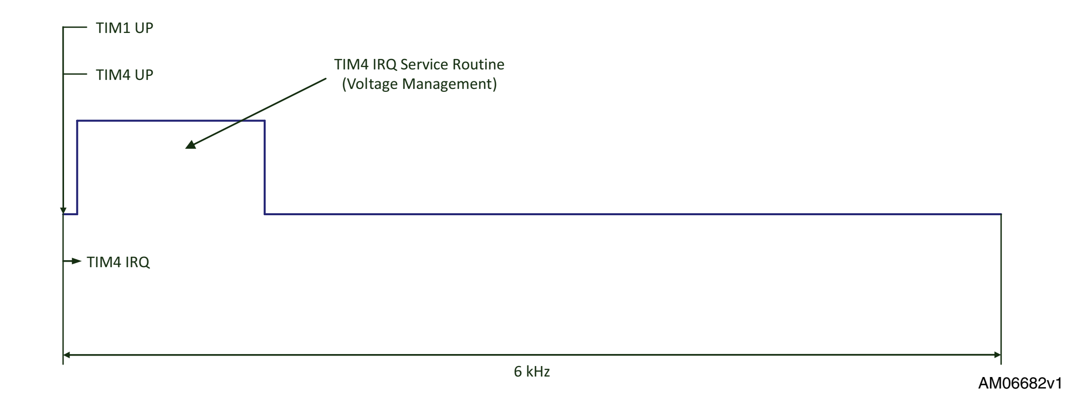


As shown in *Figure 11*, with the dual MC firmware the TIM1_UP event triggers the starts for ADC1 and ADC2. At the end of the conversion, the related ADC1_2_IRQ routine processes the acquired bus voltages of the main and sub motor stages.

In the integrated firmware, since the two bus voltages are already converted by the firmware of the PFC part, they have to be passed to the MC part at the correct moment, like the dual MC firmware has done, that is, immediately after the TIM1 update event.

To replicate this behavior, an auxiliary timer (TIM4) is used that processes the bus voltages after the TIM1_UP event. In practice, the code executed in the ADC1_2 IRQ routine in the dual FOC MC part is now executed inside the TIM4_IRQ routine.

Figure 12. Voltage management timing in integrated firmware



The dual FOC MC uses left-aligned data for the injected group while the digital PFC uses right-aligned data for the regular group. Therefore, the converted values coming from the code written for the digital PFC part have to be adjusted to fit the format of data used by the dual FOC MC part.

4.2.3 Conflicts between I/O pins

As described in *Section 3.3*, the dual FOC MC uses a DAC for debugging purposes.

Because the DAC takes control of PA4 (DAC_OUT_1) and PA5 (DAC_OUT_2), it is not possible to use it: the same pins are used by the digital PFC to read the PFC current and input AC voltage.

Another conflict is generated on the PC9 pin, which represents both the TIM8_CH4 and TIM3_CH4 signal outputs for the dual FOC MC and digital PFC respectively.

TIM3_CH4 is used to drive the power MOSFET gate, while TIM8_CH4 (or better, an edge commutation on it) is used internally as the trigger input for the ADC3 start conversion.

Although TIM8_CH4 is used as an internal trigger for ADC3, disabling its output on the PC9 pin also disables its triggering functionality. As such, it is necessary to use another start trigger source for the injected conversion of ADC3.

Figure 13. Triggers for STM32 ADC3

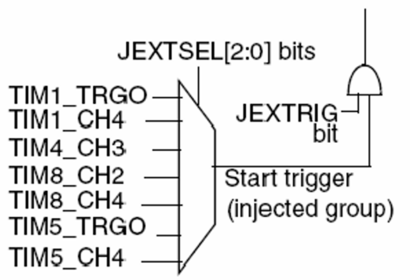


Figure 13 shows the trigger sources for the injected conversion of ADC3. Signals from TIM1 and TIM8 are already used.

TIM4_CH3 is mapped to pins used by the dual FOC MC.

TIM5_CH4 is mapped to PA3, which is used by the digital PFC (see *Table 1*).

Therefore, TIM5_TRGO has to be selected as the start trigger source for the injected conversion of ADC3. TIM5_CC3 is mapped to TIM5_TRGO for internal purposes only.

In the integrated firmware, TIM5_CC3 therefore replaces the function of TIM8_CH4 in the dual FOC MC part.

4.2.4 IRQ priorities

To set the IRQ priorities when the two parts are merged, it must be taken into account that the execution time of the FOC routine lasts longer than that of the PFC routine. Therefore, this last routine has to be able to interrupt the FOC routine.

As such, priority must be given to IRQs used for managing protection conditions or synchronizations.

Table 5 shows the interrupts and their priorities for the integrated firmware, with 0 being the highest priority.

Table 5. Used interrupts and their priorities for integrated firmware

Peripheral	IRQ use	Pre-emption priority	Sub priority
EXTI/Line1	Overcurrent protection	0	0
TIM1 BRK	Emergency condition for main motor	0	0
TIM8 BRK	Emergency condition for sub motor	0	0
TIM1 UP	Synchronization for managing main motor	0	0
TIM8 UP	Synchronization for managing sub motor	0	0
DMA1_CH1	PFC routine	1	0
TIM4	Manages bus voltages for MC	2	0
ADC1/2	Optionally manages brake resistor	2	0
ADC3	FOC implementation for both main and sub motor	2	0
SYSTICK	Timer for delays	4	0

4.3 Brief overview of firmware integration

In this application, the "STM32 dual FOC MC demonstration software" is intended as the host firmware. The modules "pfc.c" and "pfc.h" of the "STM32 digital PFC" must be integrated into the host firmware. Additionally, all the modifications described in the previous sections must be added to the new code. To initialize the PFC, the following function has to be added.

```
/* PFC Initialization */

PFC_INIT();
```

In particular, this function call has been added in the "main.c" file of the host firmware, and is called after each initialization of the dual FOC MC.

Likewise, all modifications to the host firmware have been inserted inside control structures characterized by the key word "PFC_ENABLE" defined in the module "pfc.h".

```
/* Global define ----*/
```

```
#define PFC_ENABLE
```

When this function is enabled, the built firmware will be the integration between the dual FOC MC and the digital PFC.

4.3.1 Firmware modifications to resolve conflicts between peripherals

The following is the code to resolve the conflicts outlined in *Section 4.2.2*, and demonstrates how the dual FOC MC part reads the bus voltages by means of the PFC part.

In the "stm32f10x_svpwm_1shunt.c" module:

```
void SVPWMGetBusSampling(void)
{
...
  pMotor = _GET_MOTOR_POINTER(MAIN_MOTOR);
#ifndef PFC_ENABLE
  pMotor->pPowerStage_Vars->h_ADCBusvolt =
ADC_GetInjectedConversionValue(pMotor-
>pBusVoltageADC,ADC_InjectedChannel_1);
#else
  pMotor->pPowerStage_Vars->h_ADCBusvolt = (Get_Vdc_main() << 3);
//to fit ADC injected configuration
#endif
...
  pMotor = _GET_MOTOR_POINTER(SUB_MOTOR);
#ifndef PFC_ENABLE
  pMotor->pPowerStage_Vars->h_ADCBusvolt =
ADC_GetInjectedConversionValue(pMotor-
>pBusVoltageADC,ADC_InjectedChannel_1);
  //pMotor->pPowerStage_Vars->h_ADCTemp    =
ADC_GetConversionValue(pMotor->pTemperatureADC)>>1;
#else
  pMotor->pPowerStage_Vars->h_ADCBusvolt = (Get_Vdc_sub() << 3);
//to fit ADC injected configuration
#endif
...
}
```

Note: The function "Get_Vdc_main()" exports the converted VDC value, which is also the value of the main motor bus voltage. The function "Get_Vdc_sub()" exports the converted value of the sub-motor bus voltage. Both are managed through the PFC part.

Below is the code for synchronizing TIM4 with TIM1.

In the "pfc.c" module:

```
void TIM4_Configuration(void)
{
  ...
  /* Selects TIM1 Output Trigger as input trigger for TIM4 */
  TIM_SelectInputTrigger(TIM4, TIM_TS_ITR0);

  /* Selects the Trigger Mode as Slave Mode for TIM4 */
  TIM_SelectSlaveMode(TIM4, TIM_SlaveMode_Trigger);
}
```

As such, the code executed in the ADC1_2 IRQ routine in the dual FOC MC part is now executed inside the TIM4_IRQ routine.

In the "stm32f10x_it.c" module:

```
 void ADC1_2_IRQHandler(void)
{
...
#ifndef PFC_ENABLE
  if((ADC1->SR & ADC_FLAG_JEOC) == ADC_FLAG_JEOC) // Test if ADC3 
JEOC is set
  {
    //It clear JEOC flag
    ADC1->SR = ~(u32)ADC_FLAG_JEOC;

    SVPWMGetBusSampling();
  }
  else
#endif
...
}
void TIM4_IRQHandler(void)
{
...
SVPWMGetBusSampling();
...
}
```

4.3.2 Firmware modifications to resolve conflicts between I/O pins

The following is the code to resolve the conflicts outlined in *Section 4.2.3*, and demonstrates how TIM5_CC3 replaces the function of TIM8_CH4 of the dual FOC MC part.

In the "pfc.c" module:

```
void TIM5_Configuration(void)
{
...
  /* Selects TIM5 Output Trigger as OC3REF */
  TIM_SelectOutputTrigger(TIM5, TIM_TRGOSource_OC3Ref);

  /* Selects TIM2 Output Trigger as input trigger fot TIM5 */
```

```
  TIM_SelectInputTrigger(TIM5, TIM_TS_ITR0);

  /* Selects the Trigger Mode as Slave Mode for TIM5 */
  TIM_SelectSlaveMode(TIM5, TIM_SlaveMode_Trigger);
}
```

Note: TIM2 is used by the dual FOC MC part to synchronize all timers.

In the "main.c" module:

```
int main(void)
{
...
#ifdef PFC_ENABLE
  TIM5_Configuration();
#endif

  SVPWM_1ShuntInit();
...
}
```

Note: `TIM5_Configuration` *must be called before any initializations performed by* `SVPWM_1ShuntInit()`*.*

In the "stm32f10x_svpwm_1shunt.c" module:

```
void SVPWMUpdateEvent_TIM8(void)
{
...
#ifdef PFC_ENABLE
    TIM5->CCMR2 &= 0xFF8F;
    TIM5->CCMR2 = (TIM8->CCMR2 >> 8);
#endif
...
#ifdef PFC_ENABLE
  TIM5->CCR3 = TIM8->CCR4;
#endif
...
#ifndef PFC_ENABLE
  ADC3->CR2 |= ADC_ExternalTrigInjecConv_T8_CC4;
#else
  ADC3->CR2 |= ADC_ExternalTrigInjecConv_T5_TRGO;
#endif
...
}
```

4.3.3 Firmware modifications to set IRQ priorities

The following is the code to resolve the conflicts outlined in *Section 4.2.4*.

In the private define of the "stm32f10x_svpwm_1shunt.c" module:

```
#ifndef PFC_ENABLE
#define ADC3_PRE_EMPTION_PRIORITY 1
#else
#define ADC3_PRE_EMPTION_PRIORITY 2
```

```
#endif
#define ADC3_SUB_PRIORITY 0

#ifndef PFC_ENABLE
#define ADC1_2_PRE_EMPTION_PRIORITY 1
#else
#define ADC1_2_PRE_EMPTION_PRIORITY 2
#endif
#define ADC1_2_SUB_PRIORITY 0
```

In the private define of the "stm32f10x_Timebase.c" module:

```
#ifndef PFC_ENABLE
#define SYSTICK_PRE_EMPTION_PRIORITY 3
#else
#define SYSTICK_PRE_EMPTION_PRIORITY 4
#endif
#define SYSTICK_SUB_PRIORITY 0
```

The other IRQ priorities related to the dual FOC MC are not changed.

4.4 Application example

This section lists the software and hardware requirements for the system to run correctly, and also describes the connection topology and performances obtained.

4.4.1 Software and hardware requirements

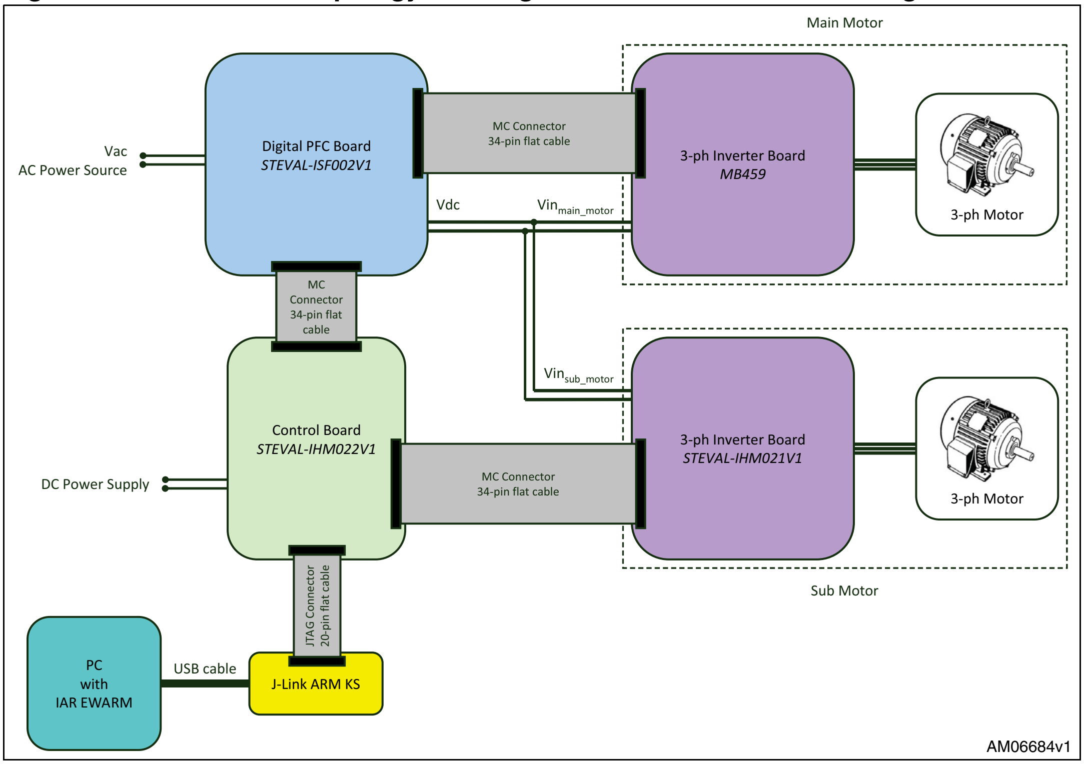
The integrated firmware has been tested with the following software and hardware elements.

- Software requirements
  - IAR embedded workbench IDE v.5.20
  - STM32 standard library "FWLib" v.2.0.1
  - STM32 Dual FOC MC + Digital PFC Demo v.1.0
- Hardware requirements
  - MB459 3-ph inverter board for main motor
  - STEVAL-IHM021V1 (modified for single shunt) as 3-ph inverter board for sub motor
  - Two 3-ph motors
  - STEVAL-ISF002V1 as PFC power board
  - Three 34-pin flat cables for MC connectors
  - AC power source able to provide 185÷230 Vrms at 50 Hz with 1000 VAC
  - Dual motor control demonstration board STEVAL-IHM022V1
  - DC power supply 5 V/2 A
  - J-Link ARM dongle
  - USB cable (type A/B plugs)
  - 20-pin flat cable for JTAG
  - PC
- For the main motor, a 3-ph motor with the following specifications:
  - Type: permanent magnet 3-ph motor
  - Number of polar couples: 2
  - Target speed: 4000 rpm
  - Target power: 600 W
  - R_S: 2.85 Ω
  - L_S: 18 mH
- For the sub motor, a 3-ph motor with the following specifications:
  - Type: permanent magnet 3-ph motor
  - Number of polar couples: 3
  - Target speed: 3200 rpm
  - Target power: 100 W
  - R_S: 110 Ω
  - L_S: 100 mH

4.4.2 Running of the system

This section describes how to connect together the various hardware elements (*Figure 14*), and shows the system performance.

Figure 14. Connection topology for integration of dual FOC MC and digital PFC

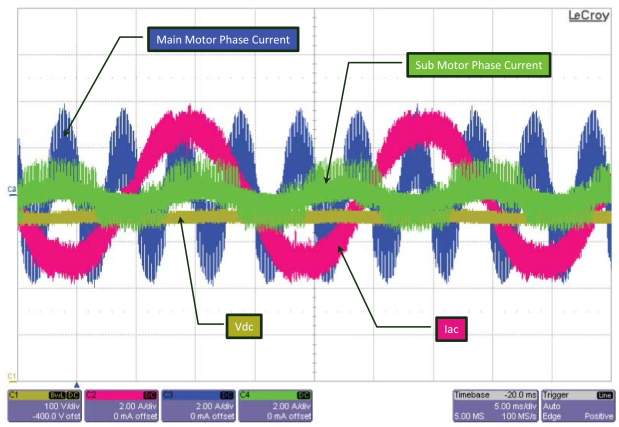


The regulated output DC voltage of the PFC is the input voltage for both 3-ph inverter boards. For this reason, Vdc has to be fixed to 350 V to be compliant with the input stage of the 3-ph inverter boards. This Vdc value forces Vac to be within the range [185÷230 Vrms] so as to obtain a good PF.

Figure 15 is a screenshot of the system when it is running. It shows:

- the bus voltage Vdc controlled by the PFC part.
- the input current Iac.
- the main motor's phase current.
- the sub motor's phase current.

Figure 15. System running



The following table shows the performances obtained.

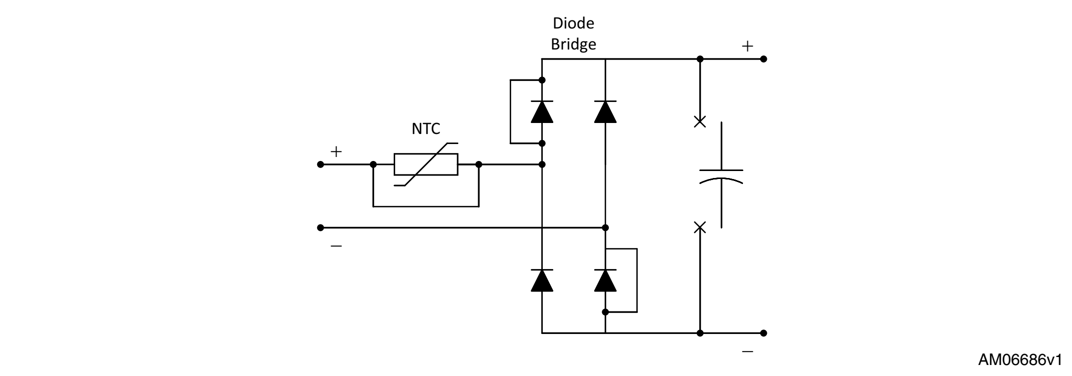
Table 6. System performances

Input voltage	Output voltage	Input power	P.F.	Current T.H.D.
185 Vrms/50 Hz	350 Vdc	850 W	0.996	2.7

4.4.3 3-ph inverter board input stage modification

When all boards are connected as shown in *Figure 14*, the 3-ph inverter boards are supplied by a regulated DC voltage and are intended as parts of a whole interconnected system. Therefore, their input stage must be modified as shown in *Figure 16*.

Figure 16. Modified input stage for 3-ph inverter boards



5 References

- UM0877
- UM0683
- UM0686
- UM0688

5.1 Useful links

- STEVAL-ISF002V1 - 1.4 kW Digital PFC power board based on STW23NM60N and TD352. Can be found at:
  http://www.st.com/stonline/products/literature/bd/17282/steval-isf002v1.htm
- STEVAL-IHM022V1 - High density dual motor control demonstration board based on the STM32F103ZE microcontroller. Can be found at:
  http://www.st.com/stonline/products/literature/bd/16072/steval-ihm022v1.htm

6 Revision history

Table 7. Document revision history

Date	Revision	Changes
27-Jul-2010	1	Initial release.