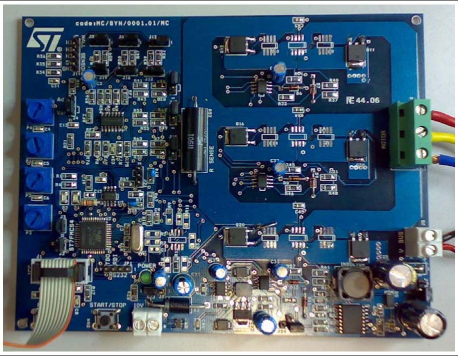
# UM0432
# User manual

## Low voltage motor control demonstration board based on ST7MC MCU

### Introduction

The low-voltage motor control demonstration board (also referred to by its order code STEVAL-IHM015V1) is a complete development platform for low-voltage motor control applications. Based on a cost-effective, flexible and open design, it allows easy demonstration of ST7MC capabilities to drive low-voltage synchronous motors. It includes an ST7MC 8-bit microcontroller with a 16-Kbyte internal Flash memory. The STEVAL-IHM015V1 comes with all the hardware necessary for developing motor control applications based on ST7MC peripherals including a motor control peripheral (MTC) and serial communication interface (SCI). The STEVAL-IHM015V1 uses an in-circuit communication (ICC) standard interface to connect to the host PC via in-circuit debuggers/programmers such as the in DART-STX board from Softec. The board's power stage is designed to support up to 25 A and up to 48 V. With the included power MOSFET device STS8DNH3LL in SO-8 package, the maximum input voltage is 30 V and the maximum current rating is 8 A. The power supply stage can be easily configured to accept a wide range of input voltages.

**Figure 1. STEVAL-IHM015V1**



### Features

- Voltage range from 5 to 48 V
- Maximum current up to 25 A
- Power MOSFET STS8DNH3LL (dual n-channel) 8 A/30 V included
- Compatible with power MOSFET in SO-8 and DPAK packages
- 10 V auxiliary power supply connector
- Serial communication interface connector
- Programming and debug support via a 10-pin ICC connector
- On-board 2 K-bit (256 bytes) serial memory
- Four potentiometers for runtime settings
- Start/stop button
- Reset button
- Debug pins available

# Contents

# List of tables

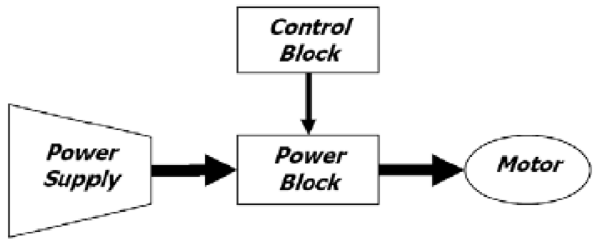
# List of figures

# 1 System architecture

A generic motor control system can be basically schematized as the arrangement of four main blocks (see *Figure 2*).

- A control block, whose main tasks are to accept user commands and motor drive configuration parameters and to provide digital signals to implement the correct motor driving strategy.
- A power block that makes a power conversion from the DC bus transferring it into the motor by means of a 3-phase inverter topology.
- The motor itself. The STEVAL-IHM015V1 board can drive the following kinds of motors.
  - PMDC (or BLAC/BLDC) permanent magnet motors, 6-step current, both in sensored and sensorless mode.
  - PMAC (or BLAC) permanent magnet motors, sinusoidal current, only in sensored mode.
- A power supply block that accepts a wide range of input voltages and provides the appropriate levels to supply both the control block and power block devices.

**Figure 2. Motor control system architecture**




The system proposed by the STEVAL-IHM015V1 includes all the above hardware blocks (apart from the motor) plus a software GUI to configure the motor drive. Additionally, open-source C code is available, derived from the ST7MC motor control libraries, allowing easy customization and extension of the control algorithms.

The control block's core consists of an ST7MC MCU that provides the driving signals to the power block according to a driving strategy that is closely related to the motor type and characteristics.

The driving signals consist of three complementary PWM signals (in the range of 0-5 V) for providing logic inputs for the high-/low-side gate drivers that belong to the power block. The proposed system has three legs (3-phase inverter).

The power block, based on the gate drivers L6387 and power MOSFET (STS8DNH3LL), converts the control signals from the ST7MC MCU to power signals for the 3-phase inverter in order to drive the motor.

The board can be supplied by a DC power supply from 5 up to 48 V with a maximum current rating up to 25 A. Refer to *Section 6* for more information on the system's architecture.

With the included power MOSFET device STS8DNH3LL, the maximum voltage rating is 30 V and the maximum current rating is 8 A.

# 2 Safety and operating instructions

## 2.1 General

**Warning: During assembly and operation, the STEVAL-IHM015V1 demonstration 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 and use, as well as 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 electronic power systems.

## 2.2 Intended use of the demonstration board

The STEVAL-IHM015V1 demonstration board is a component designed for demonstration purposes *only*, and must not be used for electrical installations or machinery. Technical data and information concerning the power supply conditions are detailed in the documentation and should be strictly observed.

## 2.3 Installing the demonstration board

The installation and cooling of the demonstration kit boards must be in accordance with the specifications and targeted application (see *Section 7*).

- The motor drive converters must be protected against excessive strain. In particular, components should not be bent or isolating distances altered during transportation or handling.
- No contact must be made with other electronic components and contacts.
- The boards contain electrostatically-sensitive components that are prone to damage if used incorrectly. Do not mechanically damage or destroy the electrical components (potential health risks).

## 2.4 Electronic connections

Applicable national accident prevention rules must be followed when working on the main power supply with a motor drive. The electrical installation must be completed in accordance with the appropriate requirements (for example, cross-sectional areas of conductors, fusing, PE connections, etc). For further information, see *Section 7*.

## 2.5 Operating the demonstration board

A system architecture that supplies power to the STEVAL-IHM015V1 demonstration board must be equipped with additional control and protective devices in accordance with the applicable safety requirements (for example, compliance with technical equipment and accident prevention rules).

**Warning:** **Do not touch the board immediately after it has been disconnected from the voltage supply as several parts and power terminals containing possibly-energized capacitors need time to discharge.**

# 3 ST7FMC2S4T6 microcontroller functions

## 3.1 Main features

- TQFP44 package.
- 16 Kbyte dual-voltage Flash program memory with read-out protection capability.
- 768 bytes of RAM (256 stacked bytes).
- Clock, reset and supply management with:
  – enhanced reset system,
  – enhanced low-voltage supervisor (LVD) for the main supply and auxiliary voltage detector (AVD) with interrupt capability,
  – clock sources: crystal/ceramic resonator oscillators and bypass for the external clock, clock security system,
  – four power-saving modes: *halt*, *active-halt*, *wait* and *slow*.
- Configurable window watchdog timer.
- Nested interrupt controller with 14 interrupt vectors.
- Two 16-bit timers.
- One 8-bit auto-reload timer.
- Serial peripheral interface (SPI).
- Serial communication interface (LINSCI™).
- Motor controller (MTC) peripheral with:
  – 6 high-sink pulse-width modulator (PWM) output channels,
  – asynchronous emergency stop,
  – analog inputs for rotor position detection,
  – permanent magnet motor coprocessor including multiplier, programmable filters, blanking windows and event counters,
  – op-amp and comparator for current limitations.
- 10-bit analog-to-digital converter (ADC) with 11 inputs.
- In-circuit communication interface (ICC, debug).

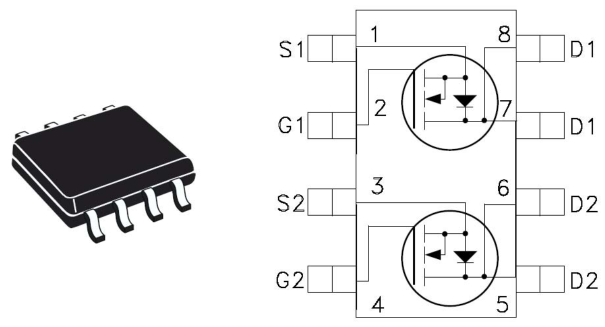
Table 1. ST7FMC2S4T6 functions

| Function | I/o name | Description (depends on embedded software) |
|---|---|---|
| MTC | MCO0 | PWM outputs high-side phase A |
| | MCO1 | PWM outputs low-side phase A |
| | MCO2 | PWM outputs high-side phase B |
| | MCO3 | PWM outputs low-side phase B |
| | MCO4 | PWM outputs high-side phase C |
| | MCO5 | PWM outputs low-side phase C |
| | MCIA, MCIB, MCIC | Analog or digital input for position sensor or B.E.M.F. detection |
| | $MCV_{REF}$ | B.E.M.F. detection comparator reference |
| | NMCES | Emergency stop |
| | OAP | Operational amplifier positive input |
| | OAN | Operational amplifier negative input |
| | OAZ | Operational amplifier output |
| | $MCC_{REF}$ | Current limitation reference |
| | MCPWMV | PWM Output V user for current reference |
| | MCZEM | Debug pin C/Z event |
| | MCDEM | Debug pin C/D event |
| SPI | MISO | Master in/slave out data |
| | MOSI | Master out/slave in data |
| | SCK | Serial clock |
| LINSCI™ | RDI | Received data input |
| | TDO | Transmit data output |
| 10-bit ADC | AIN0 | Temperature sensor input |
| | AIN1 | Bus voltage sensing input |
| | AIN13 | Trimmer P1 reading input |
| | AIN11 | Trimmer P2 reading input |
| | AIN7 | Trimmer P3 reading input |
| ICC | ICCCLK | Output serial clock |
| | ICCDATA | Input/output serial data |
| | ICCSEL/Vpp | Programming voltage input |
| Other I/O | PE0 | GPIO_A for GE method of BEMF detection |
| | PE1 | GPIO_B for GE method of BEMF detection |
| | PE2 | GPIO_C for GE method of BEMF detection |
| | PE3 | Start/stop push-button switch |
| | PB7 | LED management |

# 4 STS8DNH3LL characteristics

The STS8DNH3LL is a dual N-channel power MOSFET in SO-8 package (30 V, 0.018 Ω, 8 A), featuring a low gate charge and STripFET™ III.

**Figure 3. STS8DNH3LL**




- $V_{DSS}$ = 30 V
- $R_{DS(on)}$ = 0.018 Ω
- $I_D$ = 8 A

**Table 2. STS8DNH3LL absolute maximum ratings**

| Symbol | Parameter | Value | Unit |
|---|---|---|---|
| $V_{DS}$ | Drain source voltage ($V_{GS}$ = 0) | 30 | V |
| $V_{DGR}$ | Drain gate voltage ($R_{GS}$ = 20 kΩ) | 30 | V |
| $V_{GS}$ | Gate source voltage | ± 16 | V |
| $I_D$ | Drain current (continuous) at $T_C$ = 25 °C | 8 | A |
| $I_D$ | Drain current (continuous) at $T_C$ = 100 °C | 5 | A |
| $I_{DM}$(1) | Drain current (pulsed) | 32 | A |
| $P_{tot}$ | Total dissipation at $T_C$ = 25 °C | 2 | W |

1. Pulse width limited by safe operating area.

*Note: Stresses above the limits shown in Table 2 may cause permanent damage to the device.*

# 5 Electrical characteristics of the board

Stresses above the limits shown in *Table 3* may cause permanent damage to the devices present inside the board. These are stress ratings only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

A 10-volt bias current measurement may be useful to check the working status of the board. If the measured value is considerably greater than the typical value, some damage has occurred to the board. See the power supply configuration for a 10-V auxiliary supply.

**Table 3. Electrical characteristics of the control board**

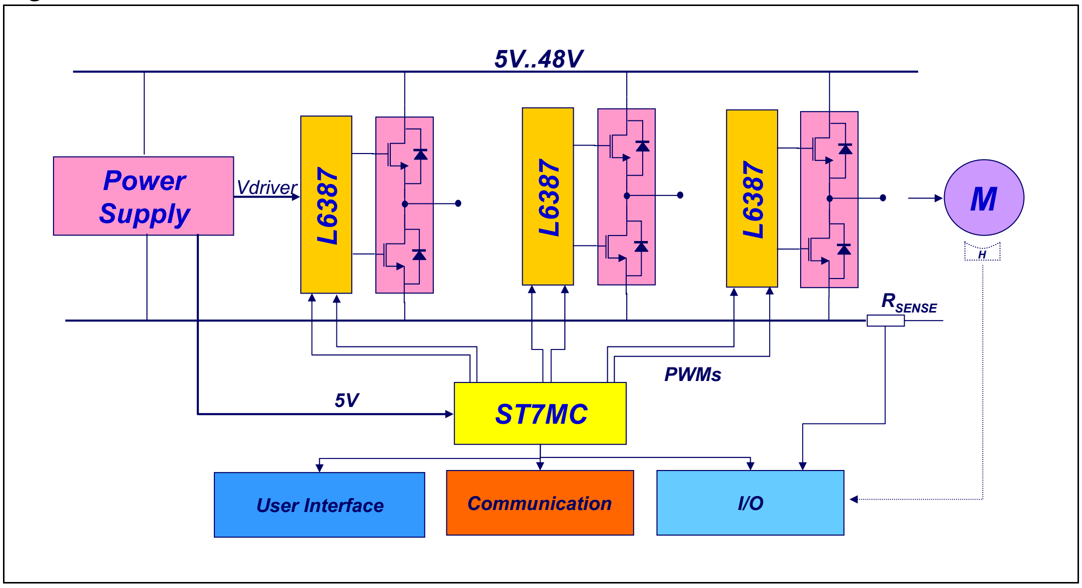
| Control board parameters | STEVAL-IHM015V1 | | Unit |
|---|---|---|---|
| | Min | Max | |
| 10 V auxiliary supply range – J6 | 5 | 15 | V |
| 10 V bias current (typical) | 30 | 70 | mA |
| $V_{BUS}$ – J5 (low voltage configuration)(1) | 5 | 6.5 | V |
| $V_{BUS}$ – J5 (medium voltage configuration)(1) | 6.5 | 9 | V |
| $V_{BUS}$ – J5 (high voltage configuration)(1) | 9 | 48 | V |

1. See *Section 6.1: Power supply on page 15.*

# 6 Board architecture

The STEVAL-IHM015V1 can be schematized as shown in *Figure 4*.

Figure 4. Board architecture




The control board's core is the ST7MC microcontroller with a dedicated peripheral included to drive the 3-phase brushless motor.

The user interface is made up of four potentiometers (P1, P2, P3, P4) used to set parameters related to the specific drive (see *Section 7*).

Two push-button switches are also implemented.

- A *Reset* button for hardware resets.
- A *Start/Stop* button used to start and stop the motor drive (see *Section 7*).

Two LEDs (green/red), each related to a specific drive, are used to obtain a status on the system (see *Section 7*).

In *normal* functioning mode, the board is supplied by the $V_{BUS}$ connector J5 but an auxiliary supply connector (J6) is included on the board to feed the drivers and the microcontroller.

The board is supplied with a 2-Kbit EEPROM (M95020) connected to the micro by the SPI bus. To enable the on-board EEPROM memory, the J2 jumper must be closed and the debug feature must be disabled inside the firmware. J4 can be set by connecting a jumper between pins 1-2 or 2-3. This setting is related to a specific drive (see *Section 7*).

Two communication systems can be established with the microcontroller.

- ICC: used for programming/debugging purposes.
- SCI: used for data exchanges through the SDI connector.

## 6.1 Power supply

The power supply has been designed to address a wide range of DC bus voltages from 5 to 48 V. Three power supply configurations have been implemented.

- Low voltage: 5–6.5 volts based on the L5970 boost converter.
- Medium voltage: a 6.5–9-volt direct connection to the bus.
- High voltage: 9–48 volts based on an L4976 buck converter (for $V_{BUS}$ > 28 V, the power MOSFET must be replaced with the STD20NF06LT4 or STS7NF60L).

**Figure 5. Power supply architecture**

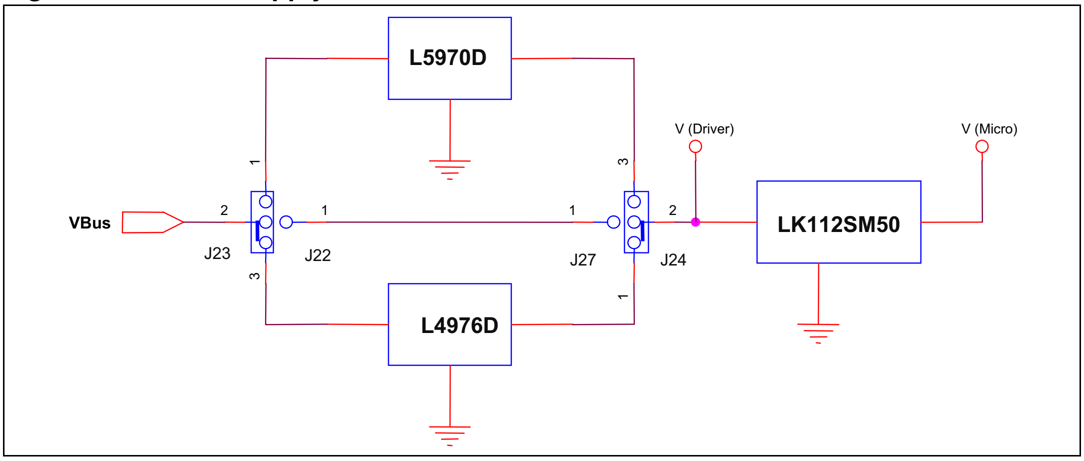



**Table 4. Power supply jumper settings**

| Configuration | $V_{BUS}$ range | V Driver | Jumper configuration |
|---|---|---|---|
| Low voltage | 5–6.5 V | 7 V | J23 between 1-2<br>J24 between 2-3 |
| Medium voltage | 6.5–9 V | $V_{BUS}$ | J23 pin2 – J22<br>J24 pin 2 – J27 |
| High voltage | 9–48 V | 8 V | J23 between 2-3<br>J24 between 1-2 |

To supply the 5 volts to the micro, you can use either the Vref signal of the L4976D regulator if it is used (high voltage setting in *Table 4*), or the LK112SM50TR. The first setting can be configured by putting J8 between pins 2-3 and the second configuration by putting J8 between pins 1-2.

With the L4976D's Vref signal, it is possible to provide the microcontroller with up to 20 mA. With the LK112SM50TR it is possible to provide up to 200 mA.

## 6.2 Power stage

The power stage is based on six power MOSFETs in full 3-phase bridge configuration. The board has three STS8DNH3LLs in SO-8 package, each containing two devices. The maximum driving current is 8 A and the suggested maximum driving voltage is 24 V.

Different power MOSFET devices can be used, replacing U15, U17 and U18 with other devices. Compatible devices are power MOSFETs in SO-8 package containing single or dual devices and power MOSFETs in DPAK package. The board is designed to work up to 25 A and 48 V.

**Figure 6. Footprint selection of power devices**

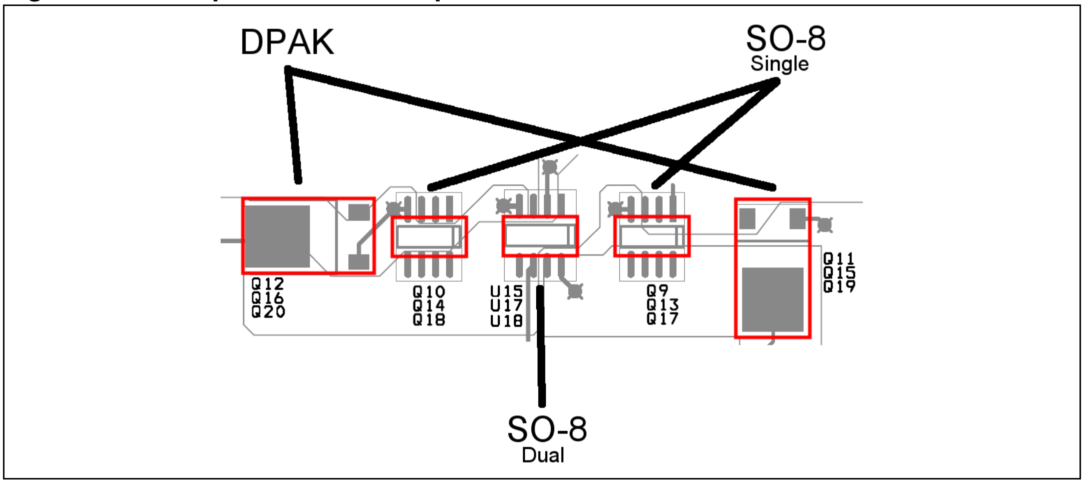


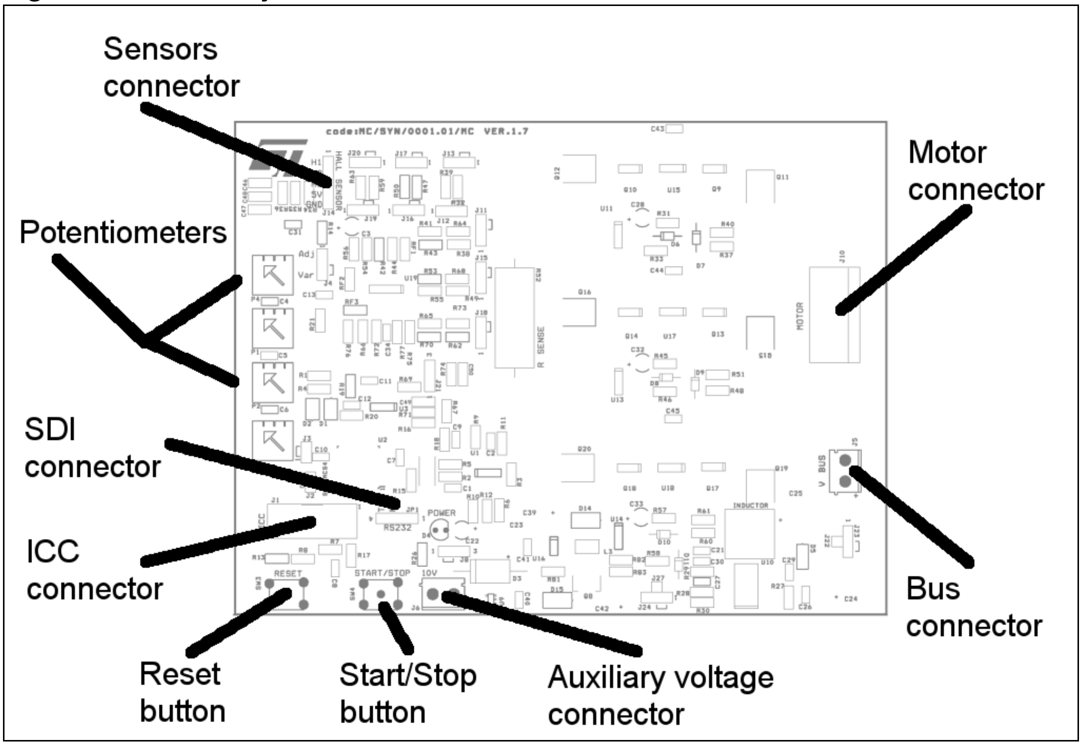

*Figure 6* shows one of the three legs of the bridge where it is possible to mount one of the three kinds of devices. *Table 5* shows a list of compatible devices.

**Table 5. Compatible devices**

| Place holder | Packages | Type | Compatible devices |
|---|---|---|---|
| U15, U17, U18 | SO-8 | Dual | STS8DNH3LL(1) |
| Q9, Q10, Q13, Q14, Q17, Q18 | SO-8 | Single | STS20NHS3LL(1)<br>STS7NF60L(1) |
| Q11, Q12, Q15, Q16, Q19, Q20 | DPAK | Single | STD20NF06LT4(1) |

1. The above are examples of compatible devices only. See the power MOSFET selection guide for a complete list of devices.

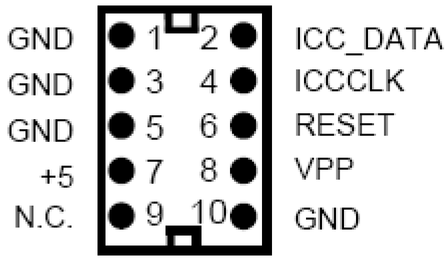
**Figure 7. Board layout**




## 6.3 ICC connector

The ICC connector (J1) is used to establish ICC communication for programming/debugging purposes. Its pinout is shown in *Figure 8*. This connector is compatible with Softec's inDART-STX board (not included in the package).

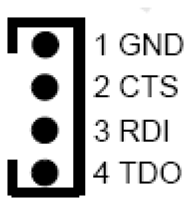
**Figure 8. ICC connector**



## 6.4 SDI interface (serial data interface)

The board is provided with an SDI interface (JP1) to establish SCI communication with external devices. We suggest using an isolation board between the SDI interface and the external devices. The pinout is shown in *Figure 9*.

**Figure 9. SDI connector**



## 6.5 Board schematic

**Figure 10. Board schematic: control block**

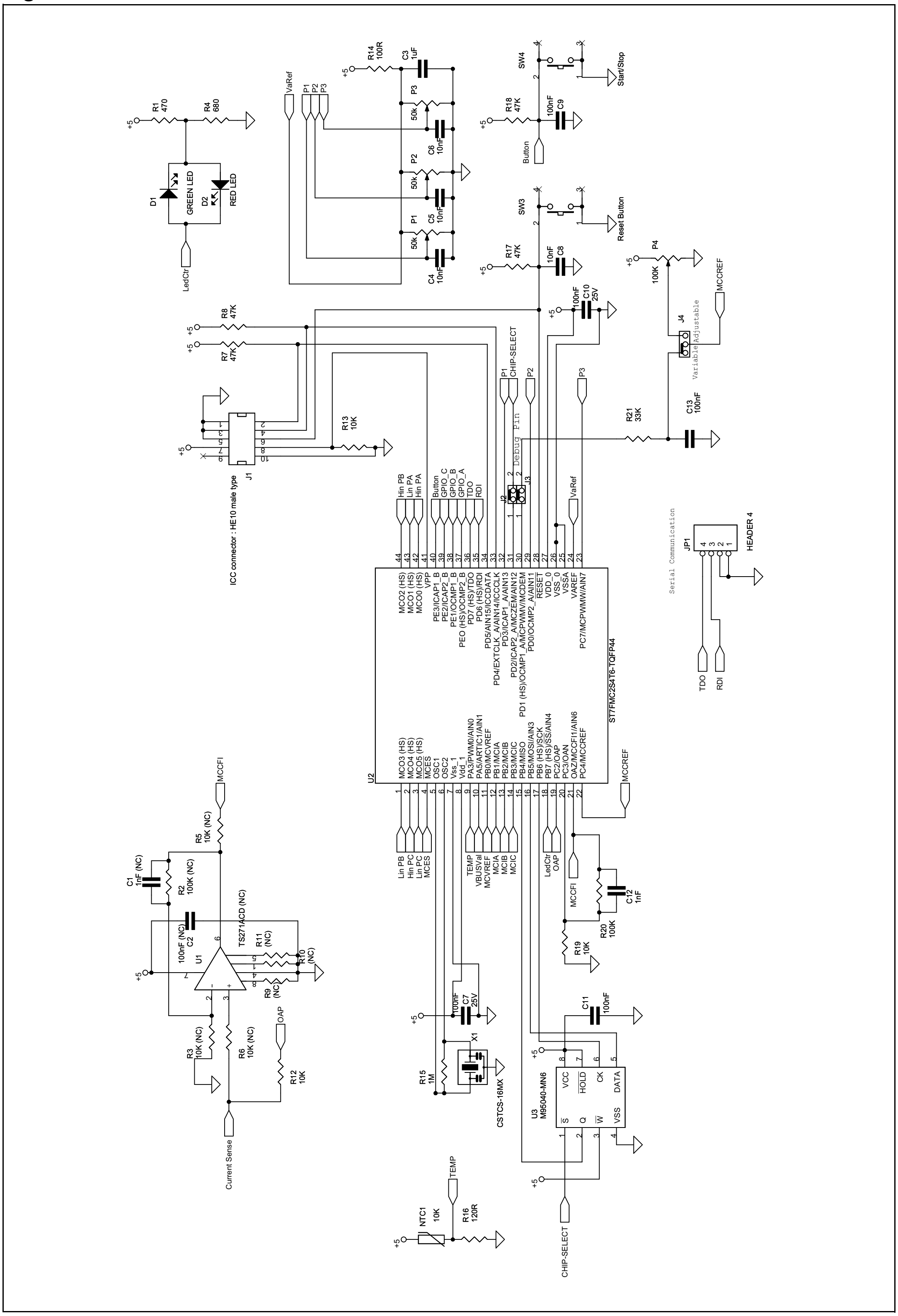
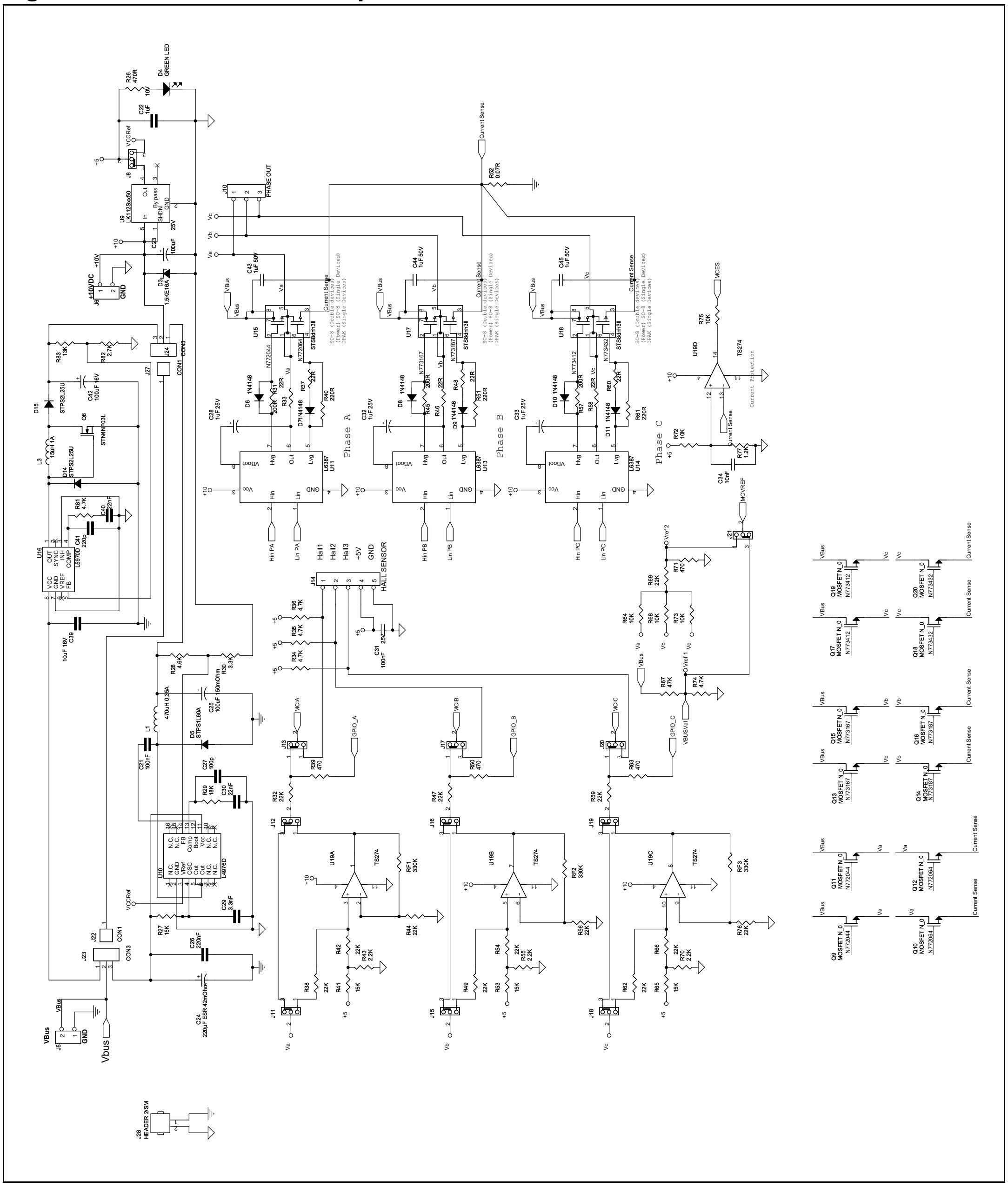


**Figure 11. Board schematic: power block**



# 7 Motor control operations

## 7.1 Environmental considerations

**Warning: The STEVAL-IHM015V1 demonstration board must only be used in a power laboratory. The voltage used in the drive system presents a shock hazard.**

The kit is not electrically-isolated from the DC input. This topology is very common in motor drives. The microprocessor is grounded by the integrated Ground of the DC bus. The microprocessor and associated circuitry are hot and MUST be isolated from user controls and serial interfaces.

**Warning: Any measurement equipment must be isolated from the main power supply before powering up the motor drive. To use an oscilloscope with the kit, it is safer to isolate the DC supply AND the oscilloscope. This prevents a shock occurring as a result of touching any SINGLE point in the circuit, but does NOT prevent shocks when touching two or more points in the circuit.**

An isolated DC power supply can be constructed using an isolation transformer and a variable transformer. A schematic of this DC power supply is shown in the application note, "AN438, TRIAC + Microcontroller: safety precautions for development tools" (although this application note was written for TRIAC, the isolation constraints still apply for switching semiconductor devices such as MOFSETs.)

*Note: Isolating the application rather than the oscilloscope is highly recommended in any case.*

## 7.2 Hardware requirements

To set-up the STEVAL-IHM015V1 demonstration board system, the following items are required.

- The board itself: STEVAL-IHM015V1
- An insulated DC power supply up to 24 V, 8 A
- Softec's inDART-STX (not included in the package)
- Softec's ICC isolation board (not included in the package)
- Two 10-pin flat cables (not included in the package)
- A brushless PM motor Ametek (not included in the package)
- An insulated oscilloscope (as needed)
- An insulated multimeter (as needed).

## 7.3 Software requirements

To customize, compile and download the motor control firmware, the following software must be installed.

- "LV ST7MC - GUI" (included in the CD-ROM).
- STVD7 for inDART-STX V.3.11 (also called "ST7 Toolset" downloadable from Softec's website: www.softecmicro.com).
- Cosmic compiler - ST7 C compiler 16 Kbyte free version - 4.5c (downloadable from Cosmic's website: www.cosmic-software.com).

### 7.3.1 Installing the software

#### LV ST7MC - GUI installation

Insert the CD-ROM provided with the kit and run *Setup.exe*.

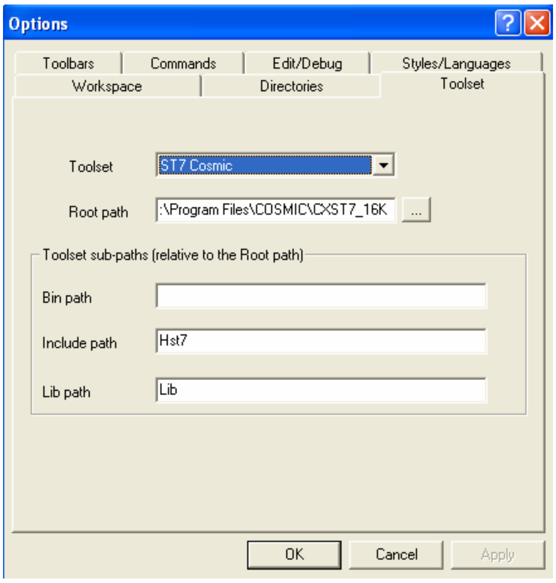
#### Third-party software installation

Follow the instructions for the related software to install and configure STVD7 for inDART-STX and cosmic compiler.

#### Installation steps

1. Install the Cosmic compiler first. Use the default installation folder: "*C:\Program Files\COSMIC\CXST7_16Kbite*". Registration is required before using the product. You can perform this procedure at any time by running the "*lmreg16k.exe*" file inside Cosmic's installation folder. Complete the form and click on the "*Register by email*" button. You will receive a license file "*license.lic*" that must be copied to the "*license*" folder in the installation folder.
2. Install STVD7 for inDART-STX. During the first run of the software after installation, a prompt should appear to configure the toolset. The toolset can be configured at any time by clicking on **Tools > Options** inside STVD7. In the *Options* window, select the *Toolset* tab, then *ST7Cosmic* from the pull-down menu and configure the toolset as shown in *Figure 12*.

Figure 12. STVD7 for InDART-STX toolset configuration




## 7.4 Board setup

### 7.4.1 Choosing the right firmware

The motor control firmware is arranged according to the kind of motor to be driven and to the driving strategy. See *Table 6* to choose which firmware should be used.

Together with the installation of "LV ST7MC - GUI" the firmware source code is installed on the PC inside the installation folder under the "LVK_Firm" folder.

Each firmware is stored inside the working folder under the same name as the firmware itself. The following files are present inside each working folder:

– a "*.stw*" file: STVD7 workspace file.
– a "*.stp*" file: STVD7 project file.
– a "*source*" folder containing all required *.c* and *.h* files.

Table 6. Firmware libraries arranged according to driving strategy

| Firmware name | Description |
|---|---|
| BLAC_3PH_SR | To drive sensored PM brushless motor sinusoidal driven |
| BLDC_3PH_SL | To drive sensorless PM brushless motor trapezoidal driven |
| BLDC_3PH_SR | To drive sensored PM brushless motor trapezoidal driven |

*Note: We suggest making a backup copy of the original working folder for each firmware. The following procedure modifies the original content of the workspace folder without leaving the possibility to return to a previous step.*

### 7.4.2 Configuring the firmware using the GUI

Before actually using the firmware, it must be configured. The term *configure* indicates the act of selecting a specific driving strategy, such as open or closed loop, voltage or current mode, and so on. Customized parameters such as current limitation, motor settings, driving related parameters and so on are also indicated.

The firmware is configured by compiling a set of "*.h*" files inside the source folder and writing a series of values as fields of *#define* statements. To perform this configuration, solid knowledge of the hardware and architecture of the firmware is required. Otherwise, the configuration tool provided inside the CD-ROM called "LV ST7MC - GUI" can be used. This tool allows you to choose and set all the required parameters visually and enables the software to automatically generate the required "*.h*" files (*Table 7* shows the "*.h*" configurations files related to each firmware).

**Table 7. ".h" configuration files**

| Firmware name | Configuration files |
|---|---|
| BLAC_3PH_SR | PMACparam.h |
| | config.h |
| | Mainparam.h |
| | MTCparam.h |
| BLDC_3PH_SL | MTC_Settings_Sensorless.h |
| | spec_settings.h |
| | version.h |
| BLDC_3PH_SR | MTC_Settings_Sensor.h |
| | spec_settings.h |
| | version.h |

For a detailed description of the configuration files and how to manually customize the related parameters refer to AN1905 and AN1947.

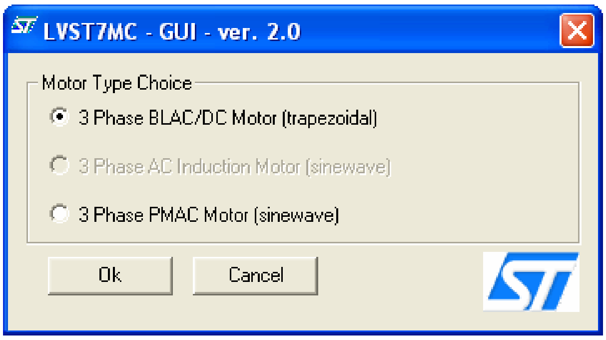
### 7.4.3 Motor type selection

Once the "LV ST7MC - GUI" is started, the *Motor Type Choice* dialog box appears (see *Figure 13*). In this window, you can select the kind of motor (PM brushless) and the driving strategy (sinusoidal or trapezoidal).

The two options are:

- "3-phase BLAC/DC (trapezoidal)" to select a PM brushless motor trapezoidal-driven, or
- "3-phase PMAC Motor (sinewave)" to select a PM brushless motor sinusoidal-driven.

**Figure 13. Motor Type Choice window**




You must select the desired value and press **OK**.

### 7.4.4 "3-phase BLAC/DC (trapezoidal)" settings

**Figure 14. "3-phase BLAC/DC (trapezoidal)" basic parameters window**

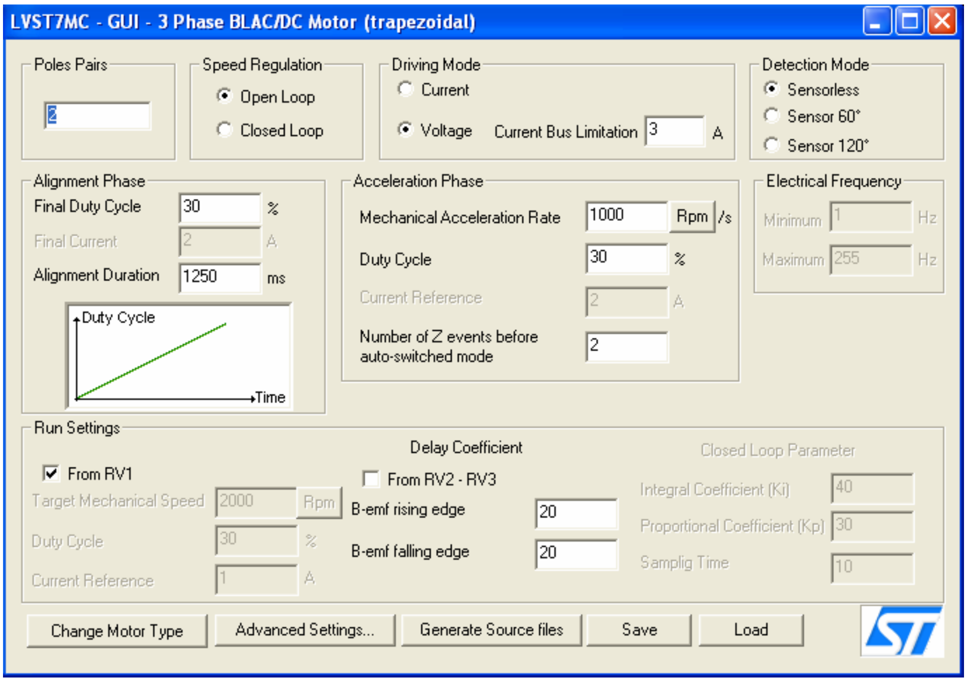


Table 8. "3-phase BLAC/DC (trapezoidal)" basic parameters

| Parameter name | Description |
|---|---|
| Poles pairs | Number of pole (north/south) pairs in the motor. |
| Speed regulation | Manner in which to run the motor, either open loop (without speed regulation) or closed loop (with speed regulation). |
| Driving mode | Motor driving mode, current mode or voltage mode. |
| Current bus limitation | Software current limitation value (only in voltage mode): if the current flowing inside one of the three phases of the motor reaches this value, overcurrent is not generated but the pwm is managed to limit the current at this level. |
| Detection mode | Back EMF (BEMF) detection mode (rotor position), either sensorless, hall (effect) sensor 60°, or hall (effect) sensor 120°. |
| Alignment phase (only for sensorless mode) | |
| Final duty cycle | Percentage of final duty cycle applied at the end of the alignment phase (only in voltage mode). |
| Final current | Value of current flowing inside the motor at the end of the alignment phase (only in current mode). |
| Alignment duration | Duration of the alignment phase in milliseconds (ms). |
| **Acceleration phase (only for sensorless mode)** | |
| Mechanical acceleration rate | Mechanical acceleration rate of the rotor during the ramp UP in RPMs (or Hz) per second (alternate between RPM and Hz settings by clicking on the RPM button). |
| Duty cycle | Duty cycle percentage during the ramp UP (only in voltage mode). |
| Current reference | Value of current flowing inside one of three phases of the motor at the end of the acceleration phase (only in current mode). |
| Number of Z events before auto-switched mode | Number of consecutive Z events that occur before the microcontroller runs the motor in auto-switched mode. |
| **Electrical frequency** | |
| Minimum | Minimum target rotor frequency in closed loop, expressed in Hz. |
| Maximum | Maximum target rotor frequency in closed loop, expressed in Hz. |
| **Run settings** | |
| From RV1 | When the "From RV1" checkbox is selected:<br>– duty cycle value is defined by the RV1 potentiometer (only for voltage mode), or<br>– current reference is defined by the RV1 potentiometer (only for current mode), or<br>– target speed is defined by the RV1 potentiometer (only for closed loop).<br>If this box is unchecked, the above parameters are set by the user. |
| Duty cycle | Duty cycle percentage when the motor is run in "open loop voltage mode". |
| Current reference | Value of current flowing inside one of three phases of the motor at run time in "open loop current mode". |
| Target speed | Target mechanical (rotor) speed in RPMs (or Hz) if speed regulation is set to "closed loop" (alternate between RPM and Hz settings by clicking on the RPM button). |
| Delay coefficient from RV2-RV3 | When the "*From RV2 - RV3*" checkbox is selected, the value of the rising delay is defined by the RV2 potentiometer and the value of the falling delay is defined by the RV3 potentiometer. If this box is unchecked, the above parameters are set by the user. |

Table 8. "3-phase BLAC/DC (trapezoidal)" basic parameters (continued)

| Parameter name | Description |
|---|---|
| B-emf rising edge | B-EMF rising edge delay coefficient value (from 0 to 255). |
| B-emf falling edge | B-EMF falling edge delay coefficient value (from 0 to 255). |
| **Closed-loop parameter (only in closed loop)** | |
| Integral coefficient (Ki) | Value of the integral coefficient (Ki) of the proportional integrative (PI) regulator. |
| Proportional coefficient (KP) | Value of the proportional coefficient (Kp) of the PI regulator. |
| Sampling time | Regulation sampling time (in milliseconds). |
| | |
| Change motor type | The **Change Motor Type** button enables the user to change the motor type (see *Figure 13*). |
| Advanced settings | The **Advanced Settings** button enables the user to set the advanced parameters (see *Section 7.4.5*). |
| Generate source files | The **Generate Source Files** button enables the user to generate the configuration ".h" files shown in *Table 7: ".h" configuration files*. A "Save" dialog window appears, where the user can select the folder in which to create the file. The user must choose the correct "Source" directory in the firmware working folder (see *Section 7.4.1*). |

### 7.4.5 "3-phase BLAC/DC (trapezoidal)" advanced settings

When you click on the **Advanced Settings** button (see *Figure 14*) the *Advanced Settings* dialog box appears (see *Figure 15*). This is where the advanced "3-phase BLAC/DC (trapezoidal)" motor type parameters are set.

Figure 15. "3-phase BLAC/DC (trapezoidal)" advanced parameters window

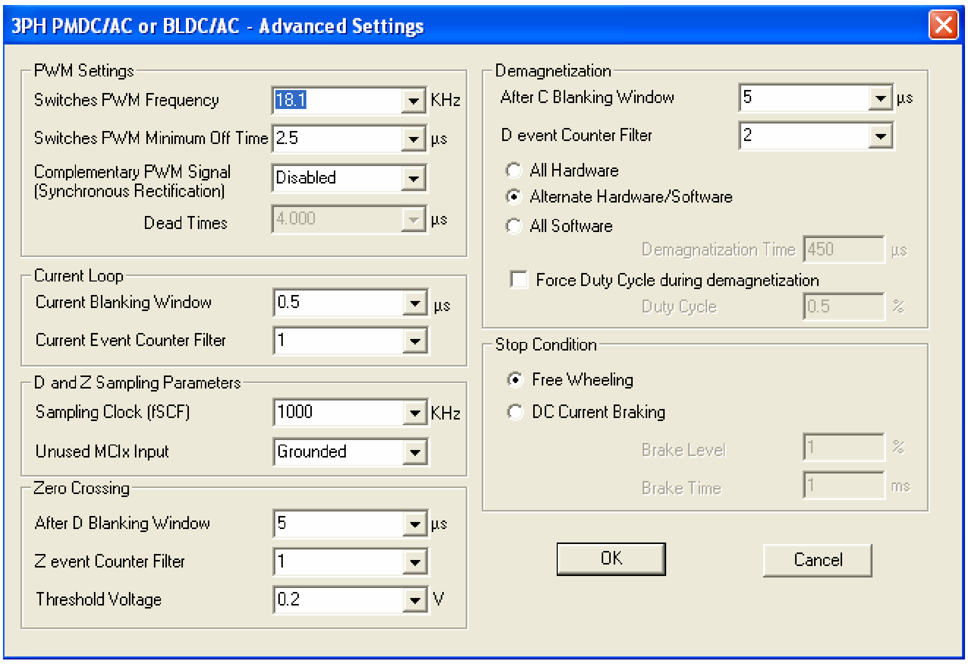


**Table 9. "3-phase BLAC/DC (trapezoidal)" advanced parameters**

| Parameter name | Description |
|---|---|
| Switches PWM frequency | Pulse width modulation (PWM) frequency in kHz. |
| Switches PWM minimum off time | PWM minimum off time in microseconds (µs) to detect the BEMF. |
| Complementary PWM signal | Enables or disables synchronous rectification. |
| Dead time | Value of dead time in µs (only if complementary PWM enabled). |
| **Current loop** | |
| Current blanking window | Time window filter in milliseconds to prevent erroneous sampling of the current after the PWM is turned ON. |
| Current event counter filter | Defines the number of counter events required to validate a current limitation event. |
| **D and Z sampling parameters** | |
| Sampling clock | Sets the frequency of the sampling clock for D and Z events in kHz. |
| Unused MCIx input | Defines in which state the unused MCI input is fixed, either "Grounded" or "Hi-Z". |
| **Zero crossing** | |
| After D blanking window | Sets the blanking window after a D event in microseconds (µs). |
| Z event counter filter | Defines the number of counter events required to validate a Z event. |
| Threshold voltage | Voltage set (in volts) for Z detection. |
| **Demagnetization** | |
| After C blanking window | Sets the blanking window after a C event in microseconds (µs). |
| D event counter filter | Defines the number of counter events required to validate a D event. |
| Demagnetization method | Three methods are available: "all hardware", "alternate hardware/software" or "all software". |
| Demagnetization time | fixed demagnetization time in microseconds (µs) (only with demagnetization methods “all software”) |
| Force duty cycle during demagnetization | Allows using a different value of the duty cycle rather than the one in the run time setting. |
| Duty cycle | Value of duty cycle percentage forced during demagnetization. |
| **Stop condition** | |
| Free wheeling | After stopping, the motor continues to spin freely. |
| DC current braking | Enables or disables DC current braking. |
| Brake level | Value of duty cycle percentage of PWM brake signal. |
| Brake time | Duration in milliseconds of the active brake. |

### 7.4.6 ”3-phase PMAC motor (sinewave)" settings

**Figure 16. "3-phase PMAC motor (sinewave)" basic parameters window**

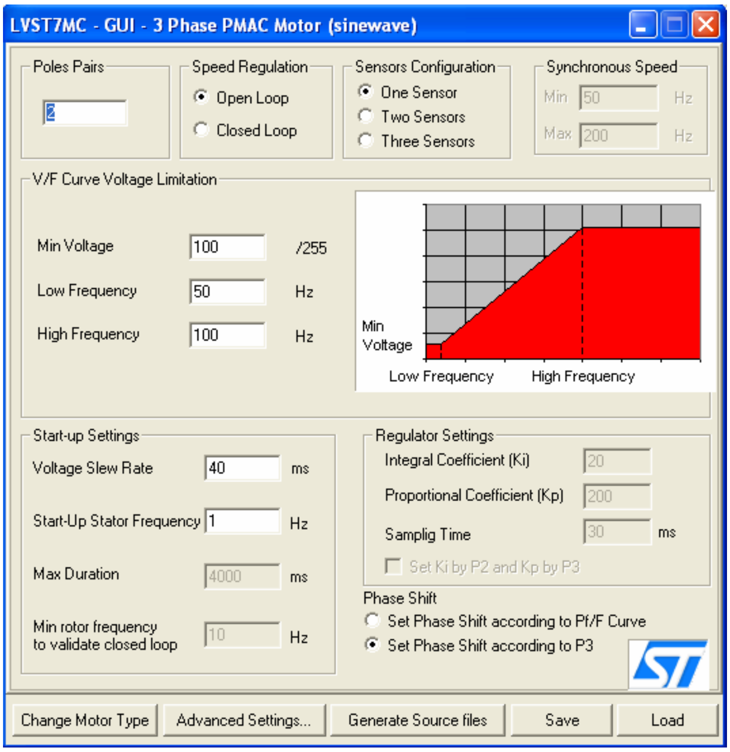



**Table 10. "3-phase PMAC motor (sinewave)" basic parameters**

| Parameter name | Description |
|---|---|
| Poles pairs | Number of pole (north/south) pairs in the motor. |
| Speed regulation | Manner in which to run the motor, either open loop (without speed regulation) or closed loop (with speed regulation). |
| Sensor configuration | Number of hall sensors on the motor: one, two or three sensors. |
| **Synchronous speed** | |
| Min | Sets the target minimum stator frequency in Hz for closed-loop mode. |
| Max | Sets the target maximum stator frequency in Hz for closed-loop mode. |
| **V/F (voltage vs. frequency) curve limitation** | |
| Min voltage | Sets the voltage level (expressed as a part of the 255$^{th}$ of the bus voltage) in the first corner of the V/F curve. |
| Low frequency | Sets the frequency of the first corner of the V/F curve in Hz. |
| High frequency | Sets the frequency of the second corner of the V/F curve in Hz. |
| **Start-up settings** | |
| Voltage slew rate | Affects the slew rate of the voltage during the motor start-up phase before reaching the potentiometer set value (only in open loop). |

Table 10. "3-phase PMAC motor (sinewave)" basic parameters (continued)

| Parameter name | Description |
|---|---|
| Start-up stator frequency | Sets the stator frequency during the start-up sequence (only in closed loop). |
| Max duration | Sets the maximum duration of the start-up sequence in milliseconds (ms) (only in closed loop). |
| Min rotor frequency to validate closed loop | Sets the rotor speed or frequency to validate the closed-loop mode (only in closed loop). |
| **Regulator settings** | |
| Integral coefficient (Ki) | Sets the value of the integral coefficient (Ki) of the proportional integrative (PI) regulator. |
| Proportional coefficient (Kp) | Sets the value of the proportional coefficient (Kp) of the PI regulator. |
| Sampling time | Sets the regulator sampling frequency in milliseconds (ms). |
| **Phase shift** | |
| Set phase shift according to Ph/F Curve | Software sets (in run time) the actual phase shift from the Ph/F curve defined in the advanced settings (based on the rotor speed). |
| Set phase shift according by P3 | Manual setting of the phase shift using potentiometer P3; the maximum CCW position is 0° of phase shift and the maximum CW position is 360° of phase shift. |
| Change motor type | The **Change Motor Type** button enables the user to change the motor type (see *Figure 13*). |
| Advanced settings | The **Advanced Settings** button enables the user to set the advanced parameters (see *Section 7.4.7*). |
| Generate source files | The **Generate Source Files** button enables the user to generate the configuration ".h" files shown in *Table 7: ".h" configuration files*. A "Save" dialog window appears, where the user can select the folder in which to create the file. The user must choose the correct "source" directory in the firmware working folder (see *Section 7.4.1*). |

### 7.4.7 "3-phase PMAC motor (sinewave)" advanced settings

When you click on the **Advanced Settings** button (see *Figure 16*), the *Advanced Settings* dialog box opens (see *Figure 17*). This is where the advanced "3-phase PMAC motor (sinewave)" motor type parameters are set.

**Figure 17. "3-phase PMAC motor (sinewave)" advanced parameters window**

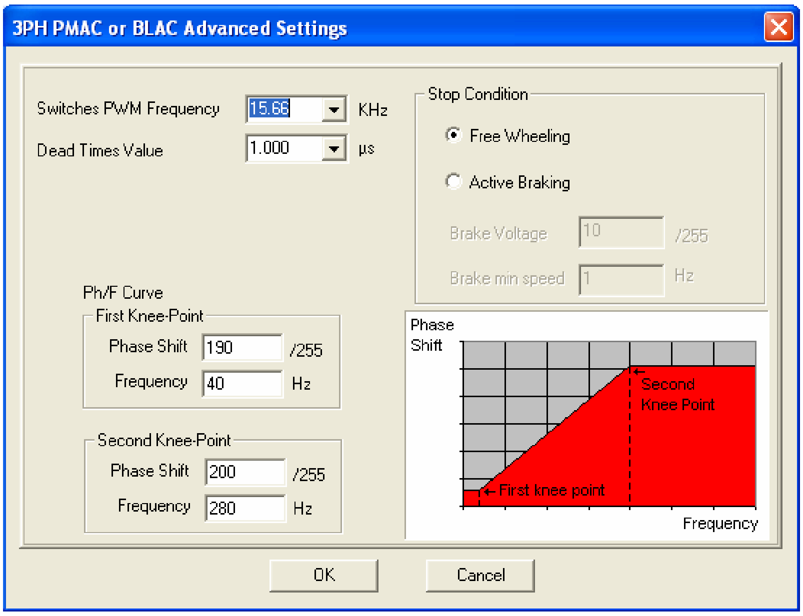



**Table 11. "3-phase PMAC motor (sinewave)" advanced parameters**

| Parameters name | Description |
|---|---|
| Switches PWM frequency | Pulse width modulation (PWM) frequency in kHz. |
| Dead times value | Selects from the available preset the dead time duration values in microseconds (µs). |
| Ph/F curve | The software uses this curve to set the value of the phase shift based on the actual value of the rotor frequency when the "set phase shift according to Ph/F curve" option is set in the main window. The curve is a linear interpolation between two knee-points. |
| **First knee-point** | |
| Phase shift | Sets the phase shift value of the first knee-point of the curve. |
| Frequency | Sets the frequency value of the first knee-point of the curve. |
| **Second knee-point** | |
| Phase shift | Sets the phase shift value of the second knee-point of the curve. |
| Frequency | Sets the frequency value of the second knee-point of the curve. |
| **Stop condition** | |
| Free wheeling | After stopping, the motor continues to spin freely. |
| Active braking | Motor is braked, generating a stator field 90° in advance with respect to the rotor field. |

Table 11. "3-phase PMAC motor (sinewave)" advanced parameters (continued)

| Parameters name | Description |
|---|---|
| Brake voltage | Voltage level (expressed as a part of the $255^{th}$ of the bus voltage) of the active braking stator field. |
| Brake min speed | Brake stays active until the motor is brought below this rotor frequency. |

### 7.4.8 Changing the maximum current allowed by GUI

The maximum current allowed by the GUI has been set to 8 A. This value may be changed by modifying the "*gui.ini*" file inside the folder where the "LVST7MC - GUI" file is installed.

Open the "*gui.ini*" file using the notepad and change the value of the following line:

MAX_CURRENT = 8

Replace the value 8 with the desired current limitation value expressed in amperes. Note that the hardware current limitation value must also be changed accordingly (see *Section 7.4.16* for instructions on how to modify this limitation).

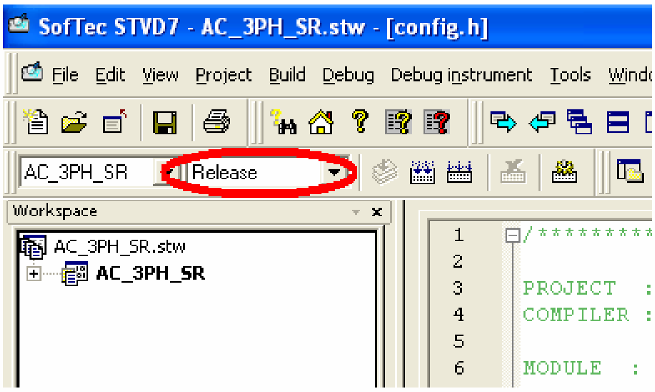
### 7.4.9 Compiling the firmware

Once the configuration files have been produced (manually or using the GUI), the binary executable file (*.s19*) must be compiled and produced.

To do this, the STVD7 for inDART-STX is used with the Cosmic compiler (see *Section 7.3*).

1. Run the STDV7 for inDART-STX and click on **File > Open Workspace**.
2. Select the workspace file in the firmware working folder depending on the motor type (see *Section 7.4.1*).
3. The default project in use is opened by the environment and is shown on the left side of the window below the opened "*.stw*" file.
4. Make sure that "*Release*" is set as the active project configuration (see *Figure 18*).

Figure 18. ST7VD active project configuration




5. Use the *Build* menu to display and select the "*rebuild all*" command. The project is compiled and built, and an executable file "*<firmware name>.s19*" is generated inside the "*release*" folder in the workspace.

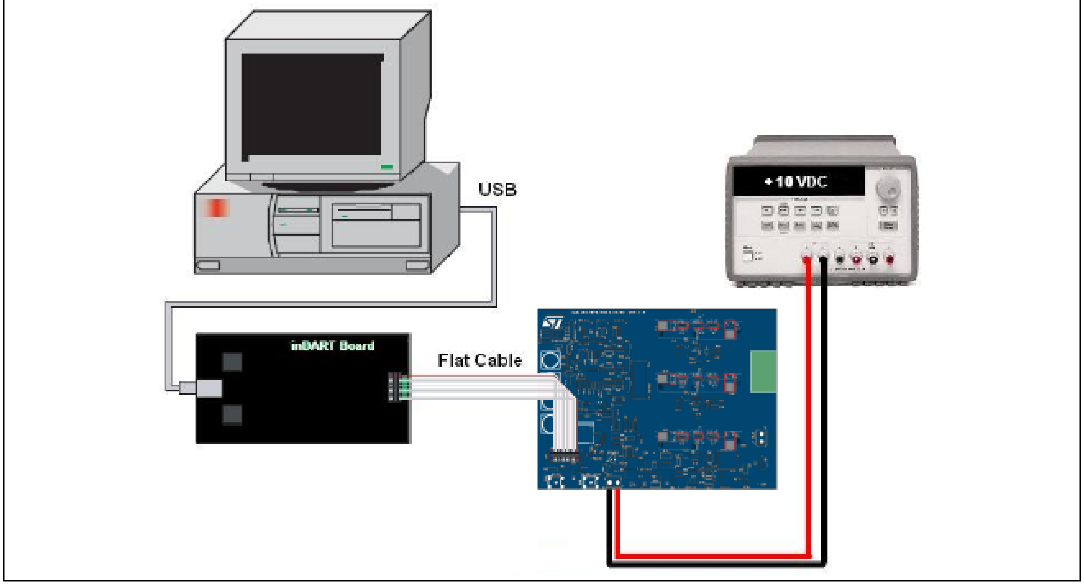
*Note:* 1 *Make sure that the following string: "<firmware name>.elf - 0 error(s), 0 warning(s)" is displayed inside the output pane after the building of the executable.*

2 *After the building of the executable, ensure that the file "<firmware name>.s19" generated inside the "release" folder in the workspace has been created. To do this, show the properties and check the creation date.*

### 7.4.10 Programming the firmware

Before programming the firmware, the board must be supplied and connected to the PC using the inDART board. We suggest setting up the system as depicted in *Figure 19*.

**Figure 19. System setup for programming phase**




1. Use the USB cable to connect the inDART-STX board to the PC. The green LED on the inDART-STX board lights up. The Windows® operating system automatically detects the new hardware and loads the appropriate USB and inDART-STX drivers.

*Note:* *Windows 2000® and Windows XP® may issue a warning the first time the inDART-STX power board is connected to the PC. The USB driver used by inDART-STX is not digitally signed by Microsoft, however, the user may safely ignore the warning since every kind of compatibility and security test has been carried out by Softec Microsystems.*

2. Connect the inDART board with the J1 connector using the 10-pin flat cable.
3. Before supplying the board, remove J23 and J24. Check that J8 is connected between pins 1 and 2.
4. Supply the control board using a 10-V power supply connected to J6 observing the polarity.

Once the ST7VD for inDART has been installed, the "datablaze programmer" utility that can be used to program the firmware using the inDART-STX is automatically installed.

5. Run the Softec datablaze programmer utility.
6. Click on **Select Device** on the toolbar.
7. In the *Select Device* window, select "*inDART-STX*" in the *Programmer Hardware* box, and "*ST7FMC2S4*" as the device code, and press **OK**.

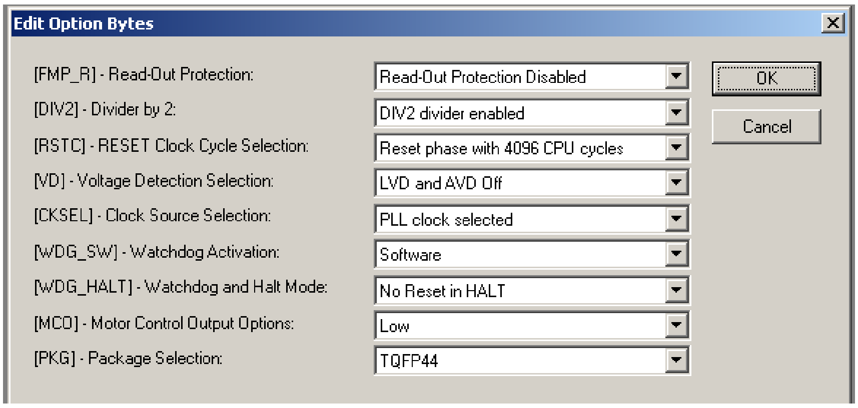
*Note: If an error occurs, make sure that the inDART-STX board is connected to the PC. A green LED lights up if the board is connected.*

8. Click on **File** > **Load** > **Code Buffer**.
9. In the *Load File to Code Buffer* dialog box format menu, select "*Motorola S-Rec*" settings.
10. Click the button near the *Name* box and select the binary code (*.S19*) to download into the microcontroller, then press **OK** (to know which binary code to select, see *Section 7.4.9*).

### 7.4.11 Setting the option bytes

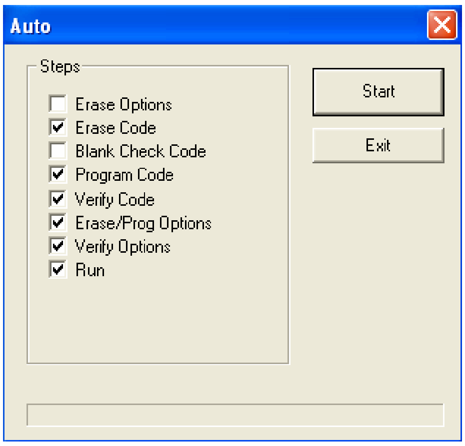
1. Press the **Option Byte** button in the toolbar and select the value as shown in the *Edit Option Bytes* configuration window (*Figure 20*), and press **OK**.

**Figure 20. Option byte settings**




2. Press the **Auto** button in the toolbar and select the programming options as shown in *Figure 21*.

**Figure 21. Programming option Auto window**




3. Press **Start** to program the device.

*Note: If an error message appears, make sure that the inDART-STX board is connected to the ControlBDST7MC2 control board and that the control board is well supplied.*

4. After programming, check the action of the LED to verify that the firmware has been correctly downloaded (see *Section 7.5.3*, *Section 7.6.3*, and *Section 7.7.3* ).

### 7.4.12 Configuring the DC input range

Before supplying the board with the bus voltage, it must be configured by setting the J22, J23, J24, J27 and J8 jumpers to the correct position. Refer to *Table 4* to configure the jumper based on the DC input range and keep the J8 jumper between 1-2.

### 7.4.13 Jumper settings table

**Table 12. Jumper settings table**

| Name | Selection | Description |
|---|---|---|
| J2 | Open | Disable the auxiliary Flash memory. The debug feature can be enabled. |
| | Closed(*) | Enable the auxiliary Flash memory. The debug feature cannot be enabled. |
| J3 | Open | Disable the current reference generated by PWMV. The debug feature can be enabled (this setting is not allowed when BLDC (trapezoidal) driving has been selected). |
| | Closed(*) | Enable the current reference generated by PWMV. The debug features cannot be enabled. |
| J4 | Between 1-2 | Adjustable – the current reference value (MCCREF) is set by potentiometer P4. Only for BLDC (trapezoidal) driving. |
| | Between 2-3(*) | Variable – the current reference value (MCCREF) is driven by the microcontroller PWM-V. Only for BLDC (trapezoidal) driving. |
| | Open | No current reference is required. For PMAC motor (sinewave) driving. |
| J21 | Open(*) | The threshold voltage used for BEMF detection is set internally by the microcontroller firmware. This option should be selected for the ST BEMF detecting strategy. |
| | Between 1-2 | The threshold voltage used for BEMF detection is derived from the phase voltage value. This option should be selected for the GE BEMF detecting strategy. |
| | Between 2-3 | The threshold voltage used for BEMF detection is derived from the bus voltage. This option should be selected for the GE BEMF detecting strategy. |

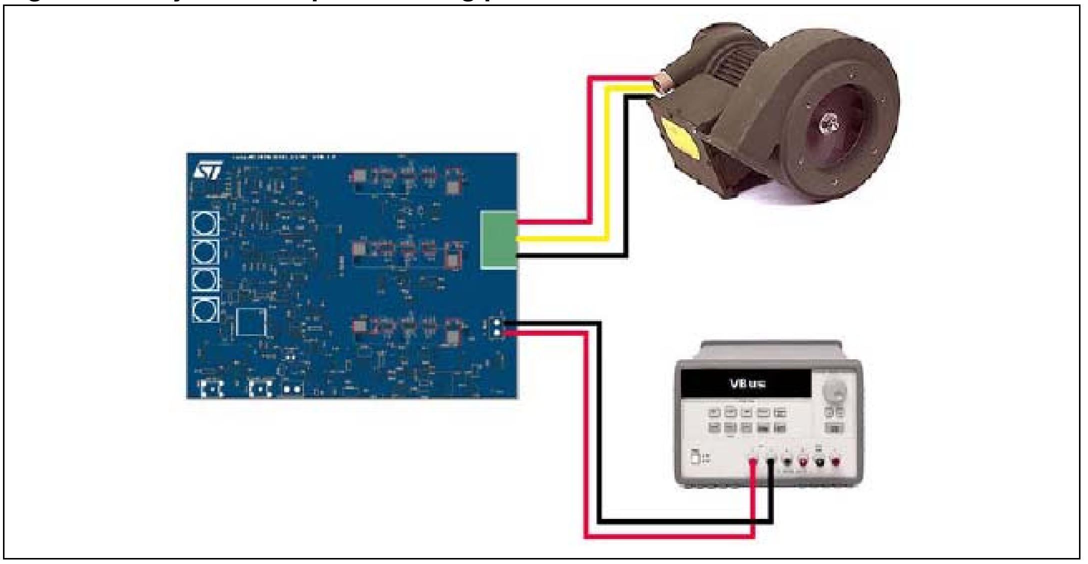
*Note: The default configuration is indicated with (*) in Table 12.*

### 7.4.14 Board connection

After the board has been programmed, the system can be configured as shown in *Figure 22*. This configuration is called a *running configuration*. Remove the ICC flat cable from the board if present.

1. Connect the insulated DC power supply to the J5 connector of STEVAL-IHM015V1, following the serigraphy for polarity.
2. Connect the phases of the motor to the J10 connector of the board and if required, connect the sensor signal to the hall sensor connector J14.

**Figure 22. System setup for running phase**




### 7.4.15 Changing the maximum allowed bus voltage level

The board includes an over-voltage protection mechanism that protects the system, preventing the motor from starting if the bus voltage is greater than a given voltage. If this occurs, the red LED starts blinking.

This threshold value is coded inside the firmware and its default value is set to 29 V. To change it, one parameter inside the source code must be modified and the executable must be rebuilt. To do this, find the following definition inside the *adc.c* source file.

```
#define HVBUS_THRESHOLD  539      // bus voltage reference (29 V)
```

Refer to *Table 13* to obtain the value relative to the desired bus voltage threshold and replace the number 539 in the statement above with this value.

**Table 13. Bus voltage threshold parameter**

| Bus threshold | HVBUS_threshold value |
|---|---|
| 6 V | 112 |
| 12 V | 223 |
| 18 V | 335 |
| 24 V | 446 |
| 30 V | 558 |
| 36 V | 670 |
| 42 V | 781 |
| 48 V | 893 |

### 7.4.16 Changing the maximum allowed current level

An over-current protection mechanism is included inside the board to protect the system, disabling all the power switches if the current that flows inside the motor is greater than a given threshold. If this occurs, the red LED starts blinking. This mechanism is called *hardware current protection*.

This threshold value is fixed by the hardware to 7.65 A. To change this threshold to $I_{MAX}$, the R77 resistor must be modified as per *Equation 1* (changing the hardware current protection threshold).

**Equation 1**

$$R_{77} = \frac{10K}{\left[\frac{5}{0.07 \bullet I_{MAX}} - 1\right]}$$

When the BLDC (trapezoidal) driving strategy is used, there is another current protection mechanism called *software current limitation*, which regulates the current that flows inside the motor. In *current* mode, the current is regulated to a certain value, and in *voltage* mode the current is limited to a maximum value.

The maximum current allowed by this mechanism is related to the hardware and is fixed at 6.5 A. To change this threshold, the R20 resistor must be modified as described below.

First, the amplification factor called AMP must be calculated using *Equation 2* (amplification factor).

**Equation 2**

$$AMP = \frac{5}{0.07 \bullet I_{MAX}}$$

Then, the value of $R_{20}$ can be calculated using *Equation 3* (changing the maximum allowed current).

**Equation 3**

$$R_{20} = (AMP - 1) \bullet 10K$$

After this modification, the "*gui.ini*" file must be modified. This file is inside the same folder as the "LVST7MC - GUI" file.

Open the "*gui.ini*" file using the notepad tool and change the values of the following lines:

```
MAX_CURRENT = 8

…

AMP_SENS = 11
```

Replace the value 8 with $I_{MAX}$ expressed in amps and 11 with the AMP value calculated by *Equation 2*.

Remember to close and re-open the "LVST7MC - GUI" file to activate this modification and follow the instructions in *Section 7.4.2*.

## 7.5 Driving the BLDC motor (trapezoidal - sensorless)

This section describes how to drive the sensorless brushless permanent magnet motor. You should first check that the board has been set-up for sensorless driving (see *Section 7.4.4*).

### 7.5.1 Specific connection (sensor)

To also drive the motor in closed-loop mode, it is not necessary that the motor include any position or speed sensor. For this demonstration we suggest using one Ametek BLDC blower motor (voltage max 30 Vdc).

### 7.5.2 Specific jumper settings

Set-up the board following the instructions in *Section 7.4.12* (bus voltage between 9 and 28 V) and *Section 7.4.13*.

**Table 14. BLDC SL jumpers setting**

| Driving mode | Jumper setting |
|---|---|
| BLDC_3PH_SL | J11 between 2-3, J12 between 2-3, J13 between 1-2 |
| | J15 between 2-3, J16 between 2-3, J17 between 1-2 |
| | J18 between 2-3, J19 between 2-3, J20 between 1-2 |
| | J3 Closed, J4 variable (2-3) |

Set-up the board as per *Table 14*.

### 7.5.3 LED action after power-on

Turn on the power supply. For this demonstration the power supply output voltage should be set to 20 Vdc and the current limitation of the power supply should be set to 4 A.

After power-on, the control board LEDs should blink red, signaling that the firmware has started to run. After a while a green LED stays on indicating an "idle" state.

### 7.5.4 Setting the potentiometers

Before running the motor, the three potentiometers P1, P2, P3 must be set (see *Table 15* for the correct configuration).

### 7.5.5 Running the motor (LED action)

Push the Start/Stop button. After pushing the button, the LEDs toggle from green to red to indicate a "run" state. The motor starts to run.

During any state: idle, start, run or brake, a blinking red LED together with the brake of the motor indicates a fault condition. A fault condition is due to one of the following:

- hardware overcurrent: current flowing inside the motor reaches a value greater than the maximum allowed current of 7.65 A – see *Section 7.4.16*.
- Over-voltage: the bus voltage reaches a value greater than 29 Vac.
- Over-temperature: the on-board temperature sensor measures a temperature greater than 110° C.
- Start-up failed: the start-up phase ends without getting a sufficient number of valid zero crossing events.
- Motor stalled: during the running of the motor no zero crossing events have been observed.

*Note: Blinking of the red LED during the running of the motor indicates that a software current limitation is in action.*

### 7.5.6 Changing the real-time parameters

The real-time parameters can be changed using the potentiometers of the control board. *Table 15* explains the potentiometer functionality based on the driving strategy.

**Table 15. Potentiometer functionality based on open/closed loop driving strategy**

| Voltage mode | | |
|---|---|---|
| | **Open loop** | **Closed loop** |
| P1 | Sets the duty cycle percentage from 0% to the maximum duty cycle allowed. | Sets the target rotor frequency value from the minimum value to the maximum value configured (see *Section 7.4.4*) |
| P2 | Sets the value of the rising delay coefficient from 0 to 255 | |
| P3 | Sets the value of the falling delay coefficient from 0 to 255 | |
| P4 | Not used | |
| **Current mode** | | |
| | **Open loop** | **Closed loop** |
| P1 | Sets the current reference value from 0 A to the maximum current allowed. | Sets the target rotor frequency value from the minimum value to the maximum value configured (see *Section 7.4.4*) |
| P2 | Sets the value of the rising delay coefficient from 0 to 255 | |
| P3 | Sets the value of the falling delay coefficient from 0 to 255 | |
| P4 | Not used | |

If during the GUI configuration phase the "from RV1" control was unchecked, the value of the duty cycle (or the value of the current reference) is not set by P1 but has a fixed value.

If during the GUI configuration phase the "from RV2 - RV3" control was unchecked, the value of the rising delay coefficient and the value of the falling delay coefficient are not set by P2 and P3 but have fixed values.

The maximum duty cycle allowed in voltage mode depends on the value of the PWM frequency and the value of the PWM minimum off time set by the GUI.

The maximum current allowed by the GUI has been set to 8 A (see *Section 7.4.8*).

### 7.5.7 Stopping the motor (LED action)

Push the Start/Stop button to stop the motor. The LEDs toggle from green to red to indicate an "idle" state.

### 7.5.8 Configuring the system for BEMF amplification

It is possible to configure the system to enable a BEMF amplification network. This configuration can be useful in any circumstance where the BEMF signal is very low, such as low speed for instance. This network can also compensate the free-wheeling diode voltage drop, which is not negligible in the case of a low bus voltage value. See AN1103 "Improved B-EMF detection for low-speed and low-voltage applications". Use the settings in *Table 16* to enable this feature.

**Table 16. BLDC SL with BEMF amplification jumper settings**

| Driving mode | Jumper setting |
|---|---|
| BLDC_3PH_SL | J11 between 2-3, J12 between 2-3, J13 between 1-2 |
| | J15 between 2-3, J16 between 2-3, J17 between 1-2 |
| | J18 between 2-3, J19 between 2-3, J20 between 1-2 |
| | J3 Closed, J4 variable (2-3) |

### 7.5.9 Detecting the BEMF during the PWM on time

The direct back-EMF sensing scheme used by default by the STEVAL-IHM015V1 demonstration board, synchronously samples the motor back-EMF during PWM "off" time without needing to sense or reconstruct the motor’s neutral point in a sensorless BLDC motor drive system. Since this direct back-EMF sensing scheme requires a minimum PWM "off" time to sample the back-EMF signal, the duty cycle cannot reach 100%. Furthermore, in some applications, that is, HVAC using high-inductance motors, the zero crossing detection is unsymmetrical at high speeds. It has been noted that the long settling time of a parasitic resonant between the motor inductance and the parasitic capacitance of power devices causes false zero crossing detections of the back-EMF. In such cases, the back-EMF detection during PWM "on" time can be used to solve the problem.

More information on this issue can be found in the application notes AN2030, AN1946 and AN1103. The STEVAL-IHM015V1 demonstration board can be configured to detect the BEMF during the PWM "on" time.

Sampling of the BEMF can be performed using three different references.

- Internal reference (default configuration)
- $V_{Bus/2}$ external reference
- Reconstructed neutral point external reference.

### 7.5.10 Detecting the BEMF using the internal reference

First of all, it is necessary to correctly size the value of the R39, R50 and R63 resistors using the following formula.

**Equation 4**

$$R_{39,\,50,\,63} = \frac{55000}{\left[\left(\frac{Vbus}{2}\right) - 2.5\right]}$$

Set the jumper as described in *Table 14*.

You must then configure the firmware. To do this, remove the comment in the following define in the *spec_settings.h* file.

```
#define BEMF_SAMPLING_DURING_TON
```

To set the motor to run at 100% of the duty cycle, change the following defines in the *spec_settings.h* file:

```
#define RV1_SEL 0
#define steady_duty  ((u8)255)
#define Max_Duty ((u16)(((u16)(mem_MCPOH)<<16)+mem_MCPOL)+1)
```

This is the default configuration and the BEMF sampling is performed using the 2.5-V internal reference.

### 7.5.11 Detecting the BEMF using the external reference $V_{Bus/2}$

The system can be set up to use $V_{Bus/2}$ as the external reference. In this case, in addition to the settings described in *Section 7.5.10*, you must replace R74 using the formula:

**Equation 5**

$$R_{74} = \frac{117500}{V_{Bus} - 2.5}$$

Also set J21 between pins 2 and 3 and change the following define in the *mtc.h* file.

```
#define mem_MCRC_GE ((u8)71)
```

You must then calculate $V_{Busth}$ using the formula:

**Equation 6**

$$V_{busth} = (V_{Bus} + 1) \cdot \frac{R_{74}}{R_{74} + 47000} \cdot \frac{1024}{5}$$

and modify the following define in the *adc.c* file replacing the calculated value:

```
#define HVBUS_THRESHOLD  VBusth
```

### 7.5.12 Detecting the BEMF using the external reference and reconstructed neutral point of the motor

The system can also be configured to use the reconstructed neutral point of the motor as reference.

To do this – in addition to the settings described in *Section 7.5.10* – you must mount the R71 resistor with the same value as R39, R50 and R63, and set J21 between pins 1 and 2.

Also change the following define in the *mtc.h* file:

```
#define mem_MCRC_GE ((u8)71)
```

## 7.6 Driving the BLDC motor (trapezoidal - sensored)

This section describes how to drive the brushless permanent magnet motor with three hall 60° sensors. First you should check that the board has been set-up for BLDC 60° sensor driving (see *Section 7.4.4*).

### 7.6.1 Specific sensor connections

In order to be driven correctly, the motor must have three position sensors, in this case three hall sensors. For this demonstration, we suggest using one Ametek BLDC blower motor (voltage max 30 Vdc).

Use the connections described in *Table 17* to connect the motor to the power board.

**Table 17. "BLDC sensored" motor connections**

| Motor | Power board |
|---|---|
| Phase A (red) | J10 pin 1 |
| Phase B (yellow) | J10 pin 2 |
| Phase C (black) | J10 pin 3 |
| Hall sensor 1 (white) | J14 pin 1 |
| Hall sensor 2 (green) | J14 pin 2 |
| Hall sensor 3 (blue) | J14 pin 3 |
| Hall sensor +5V (red) | J14 pin 4 |
| Hall ground (black) | J14 pin 5 |

### 7.6.2 Specific jumper settings

Set-up the board following the instructions in *Section 7.4.12* (bus voltage between 9 and 28 V) and *Section 7.4.13*.

**Table 18. BLDC SR jumper settings**

| Driving mode | Jumper settings |
|---|---|
| BLDC_3PH_SR | J11 open, J12 open, J13 between 2-3 |
| | J15 open, J16 open, J17 between 2-3 |
| | J18 open, J19 open, J20 between 2-3 |
| | J3 closed, J4 variable (2-3) |

Set-up the board as per *Table 18*.

### 7.6.3 LED action after power-on

Turn on the power supply. For this demonstration, the power supply output voltage should be set to 20 Vdc and the current limitation of the power supply should be set to 4 A.

After power-on, the control board's green LED should blink, signaling that the firmware has started to run. After a while a green LED stays on indicating an "idle" state.

### 7.6.4 Setting the potentiometers

Before running the motor, the three potentiometers P1, P2, P3 must be set (see *Table 19* for the correct configuration).

### 7.6.5 Running the motor (LED action)

Push the Start/Stop button. After pushing the button the LEDs toggle from green to red to indicate a "run" state. The motor starts running.

During any state: idle, start, run or brake, a still red LED together with the brake of the motor indicates a fault condition. A fault condition is due to one of the following:

- hardware overcurrent: current flowing inside the motor reaches a value greater than the maximum allowed current of 7.65 A (see *Section 7.4.16*).
- Over-voltage: the bus voltage reaches a value greater than 29 Vac.
- Over-temperature: the on-board temperature sensor measures a temperature greater than 110° C.
- Start-up failed: the start-up phase ends without getting a sufficient number of valid zero crossing events.
- Motor stalled: during the running of the motor no zero crossing events have been observed.

*Note: Blinking of the red LED during the running of the motor indicates that a software current limitation is in action.*

### 7.6.6 Changing the real-time parameters

The real-time parameters can be changed using the control board's potentiometers. *Table 19* explains the potentiometer functionality based on the driving strategy.

**Table 19. Potentiometer functionality based on open/closed loop driving strategy**

| | | |
|---|---|---|
| **Voltage mode** | | |
| | **Open Loop** | **Closed loop** |
| P1 | Sets the duty cycle percentage from 0% to 100%. | Sets the target rotor frequency value from the minimum value to the maximum value configured (see *Section 7.4.4*). |
| P2 | Sets the value of the rising delay coefficient from 0 to 255 | |
| P3 | Sets the value of the falling delay coefficient from 0 to 255 | |
| P4 | Not used | |
| **Current mode** | | |
| | **Open loop** | **Closed loop** |
| P1 | Sets the current reference value from 0 A to the maximum current allowed. | Sets the target rotor frequency value from the minimum value to the maximum value configured (see *Section 7.4.4*). |

Table 19. Potentiometer functionality based on open/closed loop driving strategy (continued)

| Voltage mode | |
|---|---|
| P2 | Sets the value of the rising delay coefficient from 0 to 255 |
| P3 | Sets the value of the falling delay coefficient from 0 to 255 |
| P4 | Not used |

If during the GUI configuration phase the "from RV1" control was unchecked, the value of the duty cycle (or the value of the current reference) is not set by P1 but has a fixed value.

If during the GUI configuration phase the "from RV2 - RV3" control was unchecked, the value of the rising delay coefficient and the value of the falling delay coefficient are not set by P2 and P3 but have fixed values.

The maximum current allowed by the GUI has been set to 8 A (see *Section 7.4.8*).

### 7.6.7 Stopping the motor (LED action)

Push the Start/Stop button to stop the motor. The LEDs toggle from green to red to indicate an "idle" state.

## 7.7 Driving the BLAC motor

This section describes how to drive the brushless permanent magnet motor. You should first check that the board has been set-up for BLAC driving (see *Section 7.4.6*).

### 7.7.1 Specific sensor connections

In order to be driven correctly, the motor must have three position sensors, in this case three hall sensors. For this demonstration we suggest using one Ametek BLDC blower motor (voltage max 30 Vdc). Refer to the connections in *Table 20* to connect the motor to the power board.

Table 20. "PMAC sensored" motor connections

| Motor | Power board |
|---|---|
| Phase A (red) | J10 pin 1 |
| Phase B (yellow) | J10 pin 2 |
| Phase C (black) | J10 pin 3 |
| Hall sensor 1 (white) | J14 pin 1 |
| Hall sensor 2 (green) | J14 pin 2 |
| Hall sensor 3 (blue) | J14 pin 3 |
| Hall sensor +5 V (red) | J14 pin 4 |
| Hall ground (black) | J14 pin 5 |

### 7.7.2 Specific jumper settings

Set-up the board as per the instructions in *Section 7.4.12* (bus voltage between 9 and 28 V) and *Section 7.4.13*.

**Table 21. PMAC SR jumper settings**

| Driving mode | Jumper settings |
|---|---|
| PMAC_3PH_SR | J11 open, J12 open, J13 between 2-3 |
| | J15 open, J16 open, J17 between 2-3 |
| | J18 open, J19 open, J20 between 2-3 |

Set-up the board as per *Table 21*.

### 7.7.3 LED action after power-on

Turn on the power supply. For this demonstration the power supply output voltage should be set to 30 Vdc and the current limitation of the power supply should be set to 4 A.

After power-on, the control board LEDs should blink green and red alternatingly, signaling that the firmware has started to run. After a while a green LED stays on indicating an "idle" state.

### 7.7.4 Setting the potentiometers

Before running the motor the two potentiometers P1 and P3 must be set (see *Table 22* for the correct configuration).

### 7.7.5 Running the motor (LED action)

Push the Start/Stop button. After pushing the button, the LEDs toggle from green to red to indicate the "run" state. Turn P1 in a clockwise direction. Keeping P1 fixed, turn P3 in a clockwise direction until the motor runs.

*Note: Turning the P3 potentiometer modifies the "phase shift" parameter. To optimize driving, the correct value of this parameter must be set. Finding the optimum value of the "phase shift" can be useful to monitor the DC current provided by the power supply. You should try to fine-tune the P3 potentiometer to minimize current absorption. When this parameter has been found, the potentiometer P3 can be left at this value for all future tests.*

During any state: idle, start, run or brake, a blinking red LED indicates a fault condition. A fault condition is due to one of the following:

- hardware overcurrent: current flowing inside the motor reaches a value greater than the maximum allowed current of 7.65 A (see *Section 7.4.16*).
- Over-voltage: the bus voltage reaches a value greater than 29 Vac.
- Over-temperature: the on-board temperature sensor measures a temperature greater than 110° C.
- Start-up failed: no signal is received from the sensors at the end of the start-up sequence.
- Motor stalled: during the running of the motor no sensor signal has been observed.

### 7.7.6 Changing the real-time parameters

The real-time parameters can be changed using the control board's potentiometers. *Table 22* explains the potentiometer functionality based on the driving strategy.

**Table 22. Potentiometer functionality based on an open-/closed-loop driving strategy**

| | Open loop | Closed loop |
|---|---|---|
| P1 | Sets the voltage modulation index from 0% to 100% of the bus voltage. | Sets the target rotor frequency value from the minimum value to the maximum value configured (see *Section 7.4.6*). |
| P2 | Not used | |
| P3 | Manual setting of the phase shift. The maximum CCW position is 0° of the phase shift and the maximum CW position is 360° of the phase shift. | |
| P4 | Not used | |

If during the GUI configuration phase the "set phase shift according to Pf/F curve" control was checked, the value of the "phase shift" is not set by P3 but is calculated at run-time based on the Pf/F curve (see *Section 7.4.7*).

The maximum current allowed by the GUI has been set to 8 A (see *Section 7.4.8*).

### 7.7.7 Stopping the motor (LED action)

Push the Start/Stop button to stop the motor. The LEDs toggle from green to red to indicate an "idle" state.

# 8 Bill of materials

Table 23. Bill of materials

| Item | Reference | Part | Footprint |
|---|---|---|---|
| 1 | C1 | 1 nF/25 V (NC) | smd 0805 |
| 2 | C2 | 100 nF/25 V (NC) | smd 0805 |
| 3 | C3 | 1 µF/25 V | Through hole |
| 4 | C4,C5,C6,C8,C34 | 10 nF/50 V | smd 0805 |
| 5 | C7,C9,C10,C11,C21 | 100 nF/50 V | smd 0805 |
| 6 | C12 | 1 nF/50 V (NC) | smd 0805 |
| 7 | C13,C31 | 100 nF/25 V | smd 0805 |
| 8 | C22 | 1 µF/10 V | Through hole |
| 9 | C23 | 100 µF/25 V | Through hole |
| 10 | C24 | 330 µF/50 V (42 mΩ) | Through hole |
| 11 | C25 | 100 µF/25 V (150 mΩ) | Through hole |
| 12 | C26 | 220 nF/50 V | smd 0805 |
| 13 | C43,C44,C45 | 220 nF/50 V | smd 0805 |
| 14 | C27 | 100 pF/50 V | smd 0805 |
| 15 | C28,C32 | 1 µF/25V | Through hole |
| 16 | C29 | 3.3 nF/50 V | smd 0805 |
| 17 | C30,C40 | 22 nF/50 V | smd 0805 |
| 18 | C33 | 1 µF 25 V | Through hole |
| 19 | C39 | 10 µF 16 V | Through hole |
| 20 | C41 | 220 pF/50 V | smd 0805 |
| 21 | C42 | 100 µF 16 V | Through hole |
| 22 | D1 | Green LED | SMD |
| 23 | D2 | Red LED | SMD |
| 24 | D4 | Green LED | Through hole |
| 25 | D3 | STMicroelectronics 1.5KE15A | Through hole |
| 26 | D5 | STMicroelectronics STPS1L60A | smd |
| 27 | D6,D7,D8,D9,D10,D11 | 1N4148 | Through hole |
| 28 | D14,D15 | STMicroelectronics STPS2L25U | smd |
| 29 | JP1 | 4-pin strip-line male | Through hole |
| 30 | J1 | ICC connector: HE10 male type | Through hole |
| 31 | J2,J3 | 2-pin strip-line male | Through hole |
| 32 | J4,J8,J11,J12,J13,J15, J16,J17,J18,J19,J20,J21 | 3-pin strip-line male | Through hole |

Table 23. Bill of materials (continued)

| Item | Reference | Part | Footprint |
|---|---|---|---|
| 33 | J5 | Vbus-2-WAY screw terminal | Through hole |
| 34 | J6 | +10 V | Through hole |
| 35 | J10 | Phase OUT-3-WAY screw terminal | Through hole |
| 36 | J14 | 5-pin strip-line male | Through hole |
| 37 | J22,J27 | 1-pin strip-line male | Through hole |
| 38 | J23,J24 | 3-pin strip-line male | Through hole |
| 39 | L1 | 470 µH 0.35 A | SMD |
| 40 | L3 | 15 µH 1 A | SMD 1210 |
| 41 | NTC1 | 10 kΩ | SMD 0603 |
| 42 | R12,R13,R19,R64,R68, R72,R73,R75 | 10 kΩ - 1/4 W | SMD 1206 |
| 43 | P1,P2,P3 | 50 kΩ - trimmer | Through hole |
| 44 | P4 | 100 kΩ - trimmer | Through hole |
| 45 | R20 | 100 kΩ - 1/4 W | SMD 1206 |
| 46 | Q8 | STMicroelectronics STN4NF03L | SMD |
| 47 | Q9,Q10,Q11,Q12,Q13,Q14, Q15,Q16,Q17,Q18,Q19,Q20 | (NC) | SMD |
| 48 | RF1,RF2,RF3 | 330 kΩ - 1/4 W | SMD 1206 |
| 49 | R1,R4, | 680 Ω - 1/4 W | SMD 1206 |
| 50 | R1,R50,R63,R71 | 470 - 1/4 W | SMD 1206 |
| 51 | R2 | 100 kΩ - 1/4 W (NC) | SMD 1206 |
| 52 | R3,R5,R6 | 10 kΩ - 1/4 W (NC) | SMD 1206 |
| 53 | R7,R8,R17,R18,R67 | 4 7 kΩ - 1/4 W | SMD 1206 |
| 54 | R9,R10,R11 | (NC) | SMD 1206 |
| 55 | R14 | 100 Ω - 1/4 W | SMD 1206 |
| 56 | R15 | 1 MΩ - 1/4 W | SMD 1206 |
| 57 | R16 | 1.2 kΩ - 1/4 W | SMD 1206 |
| 58 | R21 | 33 kΩ - 1/4 W | SMD 1206 |
| 59 | R26 | 470 Ω - 1/4 W | SMD 1206 |
| 60 | R27,R41,R53,R65 | 15 kΩ - 1/4 W | SMD 1206 |
| 61 | R28 | 4.6 kΩ - 1/4 W | SMD 1206 |
| 62 | R29 | 18 kΩ - 1/4 W | SMD 1206 |
| 63 | R30 | 3.3 kΩ - 1/4 W | SMD 1206 |
| 64 | R31,R45,R57 | 200 Ω - 1/4 W | SMD 1206 |
| 65 | R32,R47,R59,R69 | 22 kΩ - 1/4 W | SMD 1206 |

Table 23. Bill of materials (continued)

| Item | Reference | Part | Footprint |
|---|---|---|---|
| 66 | R38,R42,R44,R49,R54,R56, R62,R66,R76 | 22 kΩ - 1/4 W | SMD 1206 |
| 67 | R33,R37,R46,R48,R58,R60 | 22 Ω - 1/4 W | SMD 1206 |
| 68 | R34,R35,R36,R74,R81 | 4.7 kΩ - 1/4 W | SMD 1206 |
| 69 | R40,R51,R61 | 220 Ω - 1/4 W | SMD 1206 |
| 70 | R43,R55,R70 | 2.2 kΩ - 1/4 W | SMD 1206 |
| 71 | R52 | 0.07 Ω - 5 W | Through hole |
| 72 | R77 | 1.2 kΩ - 1/4 W | SMD 1206 |
| 73 | R82 | 2.7 kΩ - 1/4 W | SMD 1206 |
| 74 | R83 | 13 kΩ - 1/4 W | SMD 1206 |
| 75 | SW3 | Reset button | Through hole |
| 76 | SW4 | Start/stop button | Through hole |
| 77 | U1 | STMicroelectronics TS271ACD (NC) | SMD |
| 78 | U2 | STMicroelectronics ST7FMC2S4T6 | SMD |
| 79 | U3 | STMicroelectronics M95020-MN3TP/S | SMD |
| 80 | U9 | STMicroelectronics LK112SM50TR | SMD |
| 81 | U10 | STMicroelectronics L4976D | SMD |
| 82 | U11,U13,U14 | STMicroelectronics L6387 | SMD |
| 83 | U15,U17,U18 | STMicroelectronics STS8DNH3LL | SMD |
| 84 | U16 | STMicroelectronics L5970D | SMD |
| 85 | U19 | STMicroelectronics TS274 | SMD |
| 86 | X1 | CSTCS-16MX | SMD |
| 87 | J2,J3,J4,J8,J11,J12,J13,J15, J16,J17,J18,J19,J20,J23,J24 | Jumpers to be assembled on relative connector according to silk screen | N.A. |
| 88 | Spacer | Size 10 mm | N.A. |

# 9 References

This user manual provides information on how to use the STEVAL-IHM015V1 and its hardware features. For additional information on supporting software and tools, refer to the following documents.

1. ST7MC datasheet: complete information about microcontroller features and peripherals.
2. ST7MC motor control related application notes: complete information about motor control libraries developed for the ST7MC microcontroller.
3. STS8DNH3LL datasheet: complete information about the power MOSFET devices included.
4. Website *http://mcu.st.com/*, dedicated to the complete STMicroelectronics' microcontroller portfolio.
5. Motor control forum: http://mcu.st.com/mcu/modules.php?mop=modload&name=Splatt_Forums&file=viewforum&forum=13.

# 10 Known limitations

The following are the known limitations present on the STEVAL-IHM015V1 board.

- The filter capacitor for the microcontroller's power supply is physically situated some distance from the micro pins and near to the start/stop button. Pushing this button generates a glitch on the power supply, which in turn triggers the LVD thus causing an unwanted reset.

  Workaround: program the option byte with AVD/LVD off.

*Note: This limitation will be fixed in the next board release.*

- The BEMF amplifier gain network is not correctly sized to run the Ametek motor at 6 V.

  Workaround: replace R38, R42, R44, R49, R54, R56, R62, R66 and R76 with 2.2 kΩ.

*Note: This limitation will be fixed in the next board release.*

# 11 Revision history

Table 24. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 24-Jul-2007 | 1 | Initial release |
| 22-Oct-2008 | 2 | – Changed document title from "Low voltage motor control demo kit" to "Low voltage motor control demonstration board based on ST7MC MCU".<br>– Added *Section 7.5.9*, *Section 7.5.10*, *Section 7.5.11*, *Section 7.5.12* and *Section 10*.<br>– Modified *Figure 10*, *Figure 11* and *Figure 20*.<br>– Modified *Table 23*. |