HARDWARE AND SOFTWARE SOLUTION DEVELOPED IN ARM MBED ENVIRONMENT FOR DRIVING AND CONTROLLING DC BRUSHLESS MOTORS BASED ON ST X-NUCLEO DEVELOPMENT BOARDS

P. Primiceri [1] , P. Visconti [2], A. Melpignano [3], G. Colleoni [4#], A. Vilei [5#]

Department of Innovation Engineering, University of Salento, 73100, Lecce, Italy
[#] AST Advanced System Technology - STMicroelectronics, 73100, Lecce, Italy.
Emails: patrizio.primiceri@unisalento.it [1], paolo.visconti@unisalento.it [2], antonio.melpignano@gmail.com [3] , gianmarino.colleoni@st.com [4] , antonio.vilei@st.com [5].

Submitted: July 15, 2016 ***Accepted: Aug. 3, 2016*** ***Published: Sep. 12, 2016***

Abstract – Aim of this work is the design and realization of a driving system for monitoring and controlling of a BLDC motor with Hall sensors embedded. The realized system is composed by three principal blocks: the control electronic board, the power driving board and the BLDC motor. The first block is based on the STM32 Nucleo development board assembled with the second one, the ST-X-Nucleo-IHM07M1 motor driver expansion board which integrates an L6230 IC driver. The used BLDC motor is the DF45M024053-A2 model provided by Nanotec. The firmware, needed to properly control motor operation, was developed in ARM mbed environment, a development tool available on cloud which allows to send the .bin file (obtained after firmware compilation) directly to the STM32 development board, regarded from operating system, once connected via USB to PC, simply as an external memory. By PC connected via USB with STM32 board, the user can choose the motor rotation direction, set the desired rpm value and, by varying potentiometer value located on board, change the rotation speed. Furthermore, different controls are performed during motor operation such as on PWM duty-cycle value (if it is equal to 100% , then power supply is removed), on temperature value of L6230 IC driver and a control of motor rotation; in this latter case, if BLDC motor is stalled for a time period higher than 3 seconds, then the power supply is interrupted in order to safeguard the motor/system integrity.

Index terms: Electronic control systems, Brushless DC Motor, Driving board, Firmware programming.

I. INTRODUCTION

The economic constraints and new standards legislated by governments place increasingly stringent requirements on electrical systems. New generations of equipment must have higher performance parameters such as better efficiency and reduced electromagnetic interference. Furthermore, the usage of green and eco-friendly electronics are greatly developed to save the energy consumption of various devices. All these improvements must be achieved but, at the same time, also system cost and power consumption have to be reduced. Brushless motor technology makes it possible to achieve these specifications; such motors combine high reliability, efficiency, lower power consumption and lower maintenance costs in comparison with brushed motors [1-5].

Brush-Less Direct Current (BLDC) motors, also called *Permanent Magnet Brushless DC Motors* (PMBLDC), are used for consumer and industrial applications owing to their compact size, controllability and simple frame. In this type of motors, the mechanical switching, used in brushed DC motors, is replaced with electronic switches suitably driven by an electronic circuit; this feature allows to obtain an improving of reliability and efficiency of the motor itself and a long operating life (due to the absence of brushes erosion). Another advantage of BLDC motors is that, at the same power and thus for equal motor torque and supply voltage values, since there are no brushes, they can be manufactured with reduced dimensions, resulting therefore more suitable for applications that require smaller volumes and weights [4-8]. The main limitation to the wider deployment of BLDC motors is the need of a complex electronic control, including position sensors, for the motor management. The development time and related costs needed to support BLDC motor may discourage manufacturers to abandon ac motors, to which they are accustomed. However for a growing number of manufacturers, the higher complexity related to the adoption of a BLDC motor is more than rewarded by increasing demand for products able to save power consumption.

BLDC motors are mainly characterized by sinusoidal or trapezoidal Back Electromotive Force (BEMF) signals, being the BEMF the voltage that occurs when there is a relative motion between magnets of the rotor and magnetic field from the windings. Motors with trapezoidal BEMF and rectangular stator currents, typology used in this work, are widely used as they offer advantages such as constant mechanical torque (assuming motor with pure trapezoidal BEMF and stator phases commutation accurate), and very high mechanical power density [9-11]. Anyway, BLDC motor is provided by permanent magnets on moving part (i.e.

rotor) and windings on fixed part (i.e. stator); energized stator windings create electromagnetic poles and the rotor (equivalent to a bar magnet) is attracted by the energized stator phase. By using appropriate sequence to supply stator phases, a rotating field on stator is created and maintained. This action of rotor, chasing after the electromagnet poles on stator, is the fundamental action used in synchronous permanent magnet motors. The lead between rotor and rotating field must be controlled to produce torque and this synchronization implies accurate knowledge of rotor position. For this reason, three Hall sensors are used to detect rotor orientation and speed very precisely by measuring variations in the magnetic field [12-17]. The BLDC motor control can be done in sensor or sensor-less mode; the advantage of sensor-less BLDC motor control is that the sensing part can be omitted and thus overall costs can be reduced, on the contrary the disadvantages are higher requirements for control algorithms and more complicated electronics. In this work a BLDC motor control, provided by Hall sensors embedded, has been realized by using development electronic boards properly programmed, as shown in figure 1, which allow to get very low implementation costs.

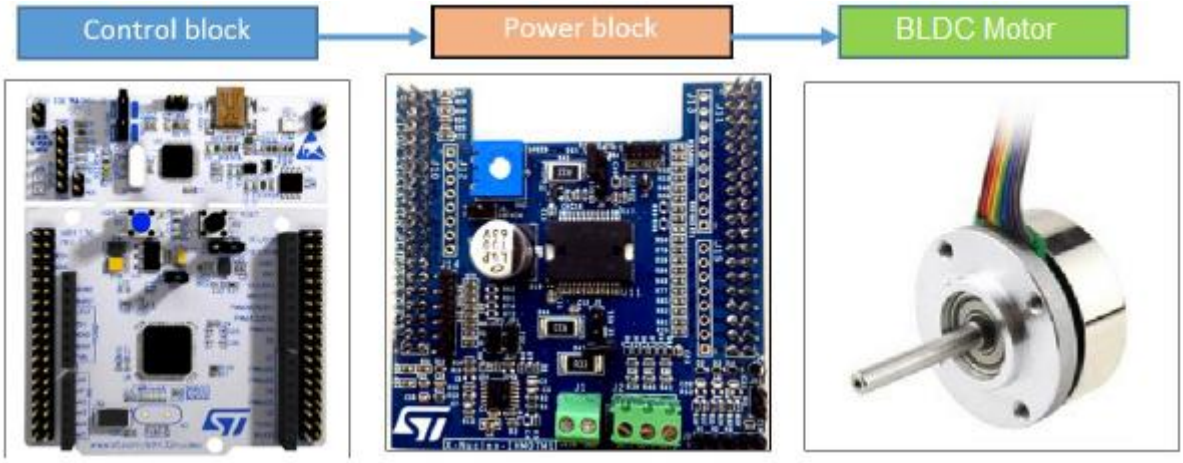

Figure 1. Block diagram of realized system: STM32 Nucleo board on the left, X-Nucleo-IHM07M1 motor driver expansion board at the center and BLDC motor on the right.

BLDC motors need a dedicated electronic control in order to function properly. The microcontroller, used for driving phases commutation, has to feed the different motor phases, correctly in sequence, by processing the output signals from Hall sensors placed on the stator. By means of the *ST X-Nucleo-IHM07M1* motor driver board, realized for this purpose, it is possible to program microcontroller using a dedicated firmware in order to control the BLDC motors, both those sensor-less and those equipped with Hall sensors. In this research work, a software in ARM mbed environment was developed, using object-oriented programming (C ++ language), realizing, in this way, the firmware for STM32 Nucleo development board needed to control and drive properly any BLDC motor equipped with Hall effect sensors.

II. BRUSHLESS MOTOR APPLICATIONS IN MANY INDUSTRIAL FIELDS.

BLDC motors are one of the motor types that rapidly are gaining popularity. They are used in devices such as computers, hard drives and CD/DVD players, in cordless power tools where the increased efficiency of the motor leads to longer periods of use before the battery needs charged; furthermore, small cooling fans in electronic equipment are powered exclusively by BLDC motors. Apart from these, they are used in many industrial applications such as appliances, automotive, aerospace, consumer, medical, industrial automation equipment and instrumentation. In particular, transport electric vehicles and hybrid vehicles (e.g. scooter, vectrix maxi-scooter, bicycles) employ on board many BLDC motors.

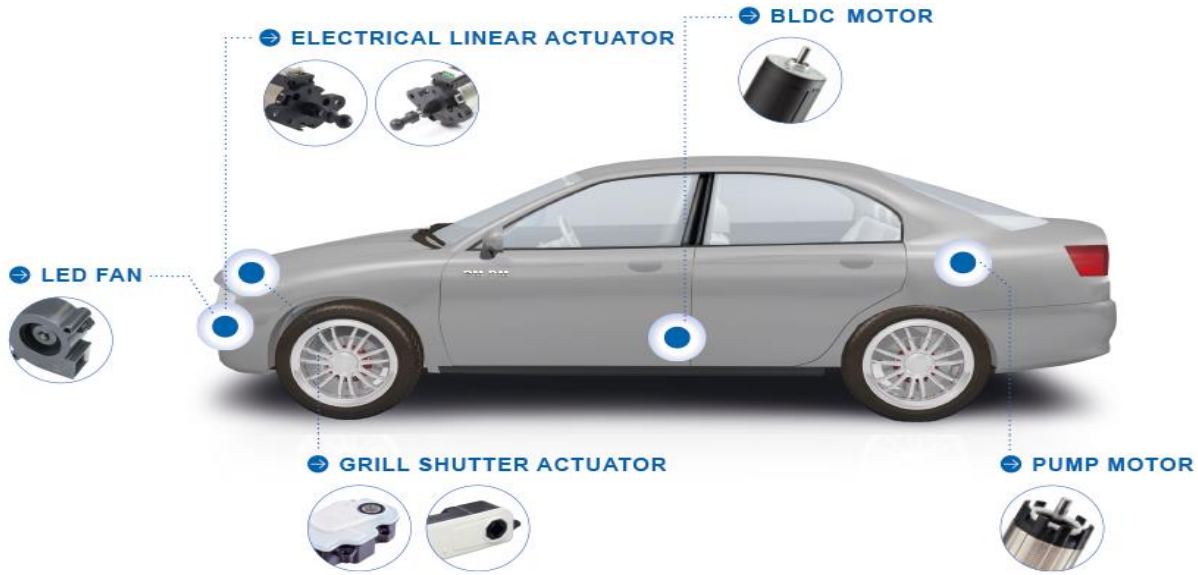

Figure 2. Automotive industry: all BLDC motors employed on vehicles.

In industrial engineering, BLDC motors are used in positioning or actuation systems, in motion control (as pump, fun or spindle drives), in adjustable or variable speed applications and in automated remote control systems. Also because of growing trend in the HVAC and refrigeration industries, BLDC motors are used due to reduced power consumption compared to AC motors. In summary, modern air conditioners, home appliances, tools, electric bikes, drones (due to the lower weight than other motor typologies) use BLDC motors (figure 3).

Another most important field of application is healthcare; the treatment of sleep apnea requires the use of Positive Airway Pressure (PAP) respirators [18, 19]. The patient uses a special breathing mask attached to the PAP respirator. A blower fan within respirator pressurizes the air in the mask to create positive airway pressure that helps the patient to breathe while asleep. The blower fan must raise or lower the patient’s airway pressure in response to his breathing pattern.

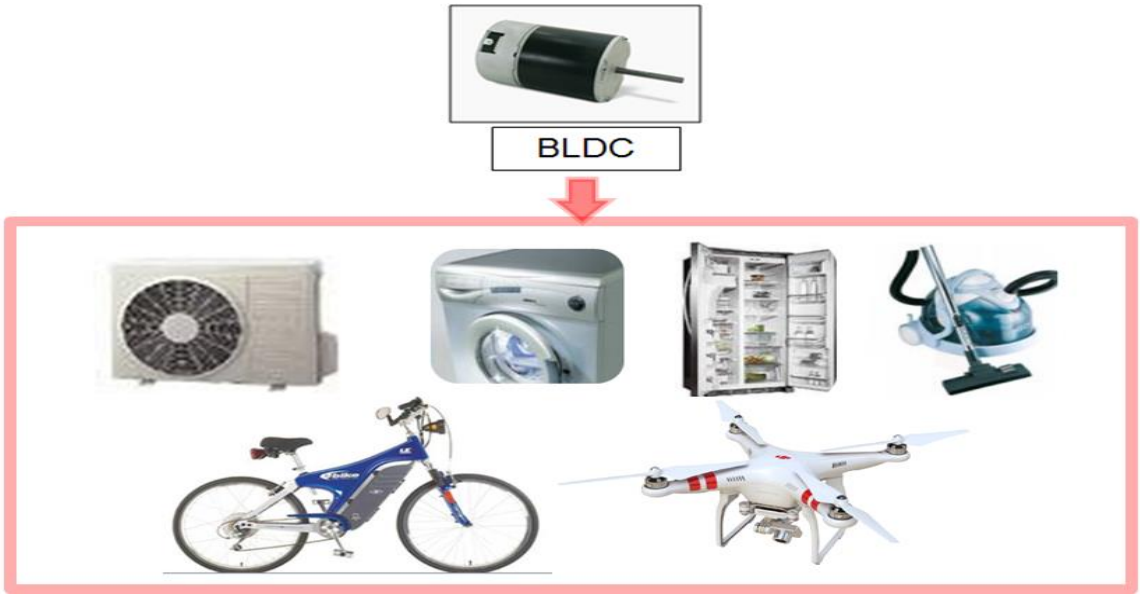

Figure 3. BLDC motors application in home appliances, tools, electric bikes, drones etc.

When the patient inhales, the blower fan must speed up to supply a larger volume of air to the lungs. When the patient exhales, the blower fan slows down to reduce the volume of air and to let the patient breathe out. BLDC motors and drives are an ideal power source for the blower fans: the motor never needs to operate below the minimum threshold speed of the drive and there is no risk of a sudden change in the load.

Another advantage of BLDC motors is that they are quiet; the motors utilized for hospital equipment or patient-care facilities must be quiet to comply with low-noise-level standards. Motors in sleep-apnea equipment operate at high speeds and yet must comply with even lower noise level standards because equipment is in the patient's bedroom at home. So the absence of a commutator and brushes in BLDC motors removes an additional source of motor noise.

Figure 4. BLDC motor used in sleep-apnea equipment is able to remove a motor noise source.

The BLDC motor will continue to emerge within medical applications as brushless DC drives continue to develop and their costs are reduced. Medical analyzers are multi-function

machines used to test human bodily fluids such as blood and urine. In a medical analyzer, fluid samples are transported from station to station to conduct various tests. Motor driver solutions include stepper motor drivers, DC motor drivers and brushless motor drivers.

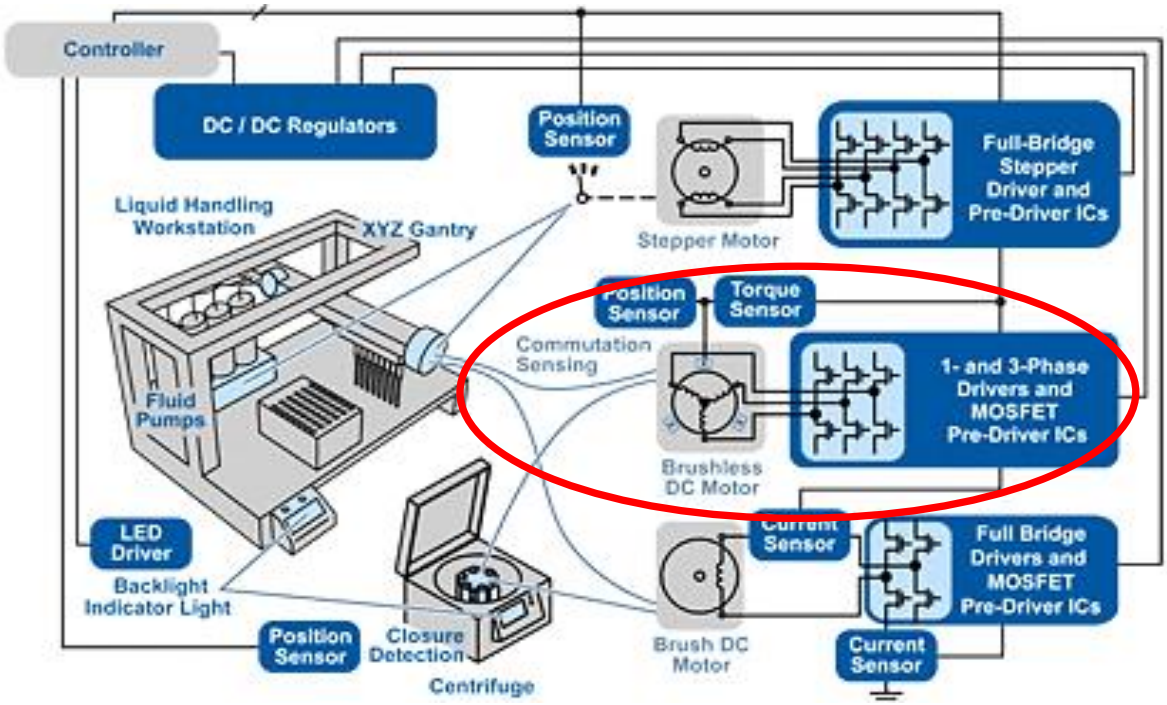

Figure 5. Medical analyzers are multi-function machines that use BLDC motors.

III. FUNCTIONAL BLOCKS OF THE REALIZED SYSTEM

The designed electronic system is composed by three principal functional blocks: the *STM32 Nucleo-64pins*, a low-cost and easy-to-use development platform to quickly evaluate and start a system design with an STM32 microcontroller in LQFP64 package, the *X-Nucleo-IHM07M1* which is a three-phase BLDC motor driver expansion board based on the *L6230 IC* driver and a BLDC motor model *Nanotec DF45M024053-A2* with Hall effect sensors embedded. Following the three system parts, just mentioned, are explained in detail.

✓ *STM32-Nucleo 64-pins development board*

The STM32 Nucleo board (shown in figure 6) provides an affordable and flexible tool for users to try out new idea and build prototypes by using the on board STM32 microcontroller, choosing from various combinations of performance, power consumption and features. STM32 Nucleo board offers extension resources both for STM Morpho extension pin headers for full access to STM32 I/O pins and for Arduino Uno Revision 3 connectivity. The development board provides on board the ST-LINK/V2, an in-circuit debugger and programmer with a JTAG/serial wire debugging (SWD) interfaces (highlighted in figure 6).

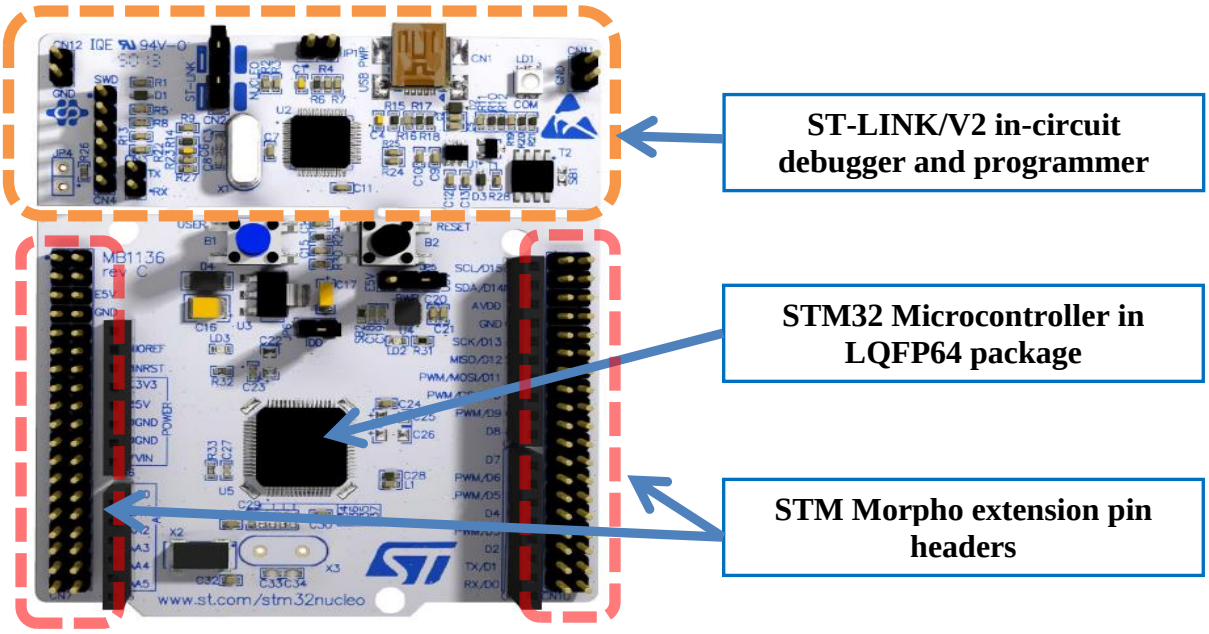

Figure 6. View of the STM32 Nucleo 64-pins board.

The board can be supplied in different way: by the host PC through the USB cable (5V), by external V_{IN} (7V < V_{IN} < 12V) or by 5V or 3.3V supply voltage (depending on jumper position) from Arduino or ST Morpho connectors; in case of 3.3V supply voltage, the ST-LINK is not powered, thus the programming and debug features are unavailable. External power supply can be used when the current consumption of Nucleo and extensions boards exceeds the allowed current on USB. In this condition, it is still possible to use the USB for communication, for programming or debugging only. In figure 7, the hardware block diagram of the STM32 Nucleo board and the microcontroller embedded on board are shown. As shown in figure 7, two push buttons B1 USER and B2 RESET, different leds (for USB communication, power led and user led) are present on board. In addition, the STM32 Nucleo board is divided in two circuital sections: the ST-LINK and target MCU sections. ST-LINK part of the PCB can be cut out to reduce the board size. In this case, remaining MCU part can only be powered by external V_{IN} by using STM Morpho connector or Arduino connector.

In this research work, the platform model Nucleo-F302R8 with a STM32F302R8T6 microcontroller was used. STM32F302R8T6 microcontroller belongs to STM32F302x6/8 family based on the high-performance ARM® Cortex®-M4 32-bit RISC core operating at a frequency of up to 72 MHz and embedding a floating point unit (FPU). ARM® Cortex®-M4 processor was developed to provide a low-cost platform that meets the needs of MCU implementation, with a reduced pin count and low-power consumption, while delivering outstanding computational performance and an advanced response to interrupts.

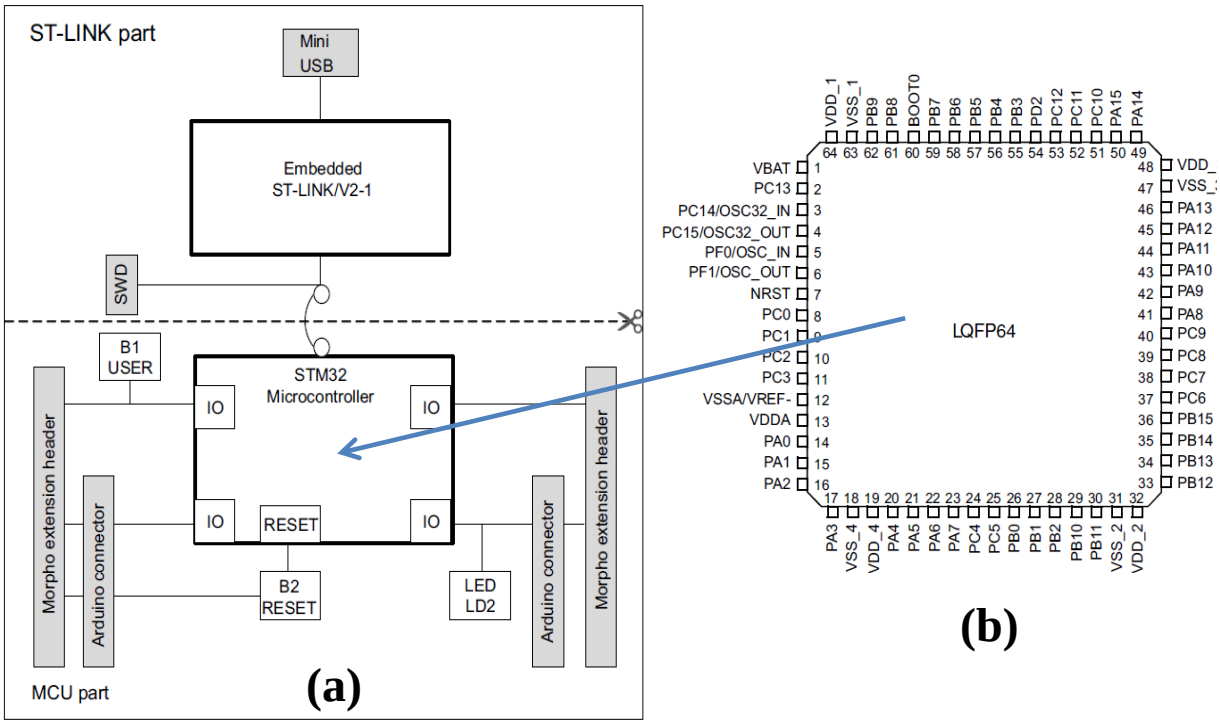

Figure 7. Block diagram of STM32 Nucleo board: connections between STM32 control unit and its peripherals (ST-LINK/V2-1, push-buttons, LEDs, Arduino connectors and STM Morpho connectors) (a), microcontroller STM32F302R8T6 embedded on board (b).

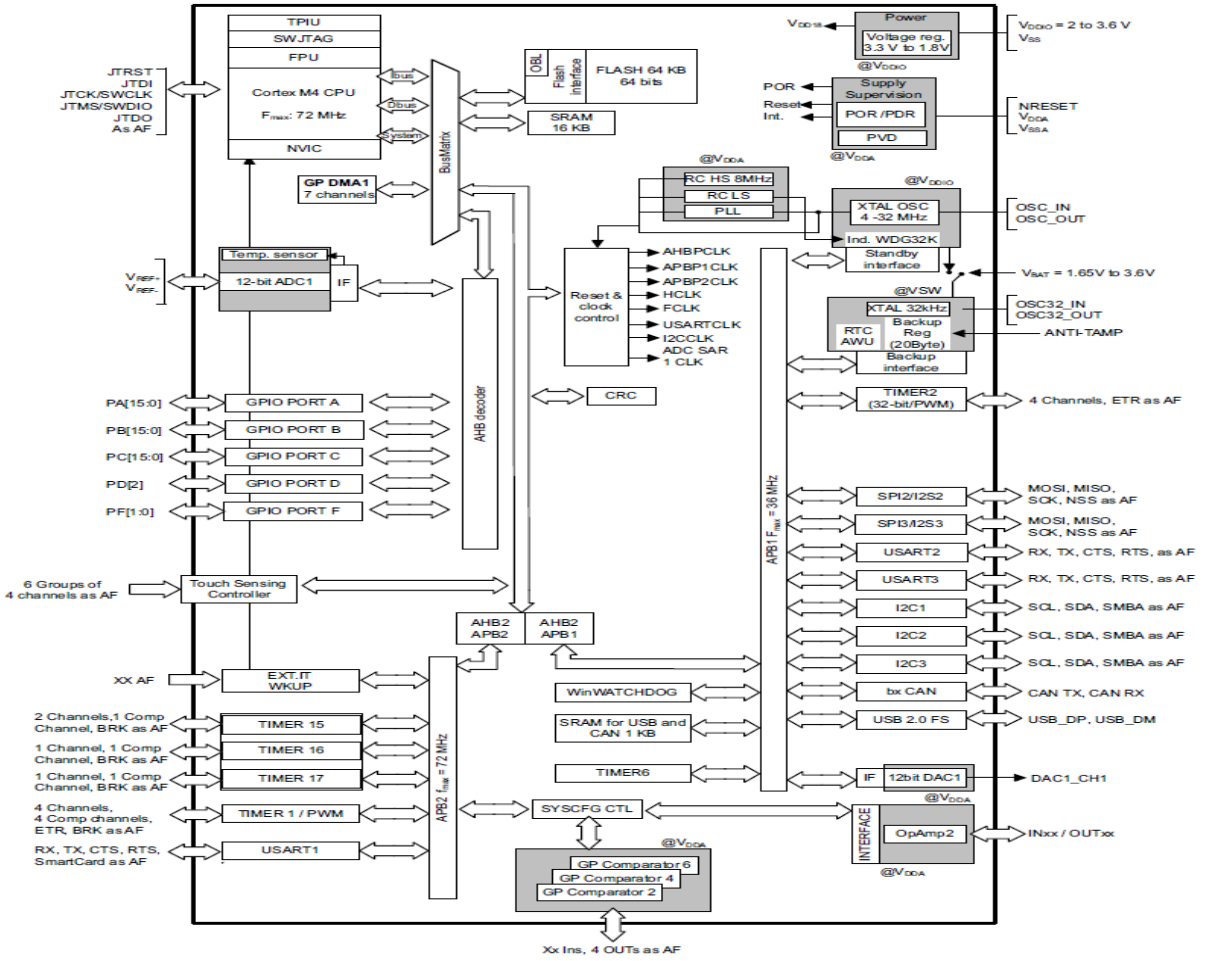

Figure 8. Block diagram of STM32F302R8T6 microcontroller embedded on STM32 Nucleo.

STM32F302R8T6 microcontroller incorporates high-speed embedded memories, an extensive range of enhanced I/O pins and peripherals connected to two Advanced Peripheral Buses (APB). The device offers a fast 12-bit ADC, three comparators, an operational amplifier, up to 18 capacitive sensing channels, one DAC channel, a low-power Real Time Clock (RTC), one general purpose 32-bit timer, one timer dedicated to motor control and one timer to drive the DAC. Microcontroller also provides standard and advanced communication interfaces: I^2C, USART, SPI with multiplexed full-duplex Inter-Integrated Sound (I2S) interface, a USB FS device, a CAN and an infrared transmitter. STM32F302R8T6 microcontroller operates in the –40°C to +105°C temperature range and with a power supply from 2V to 3.6V. A comprehensive set of power-saving modes allows the design of low-power applications.

In figure 9, electrical connections among the main blocks of STM32 Nucleo board are shown; the extension connectors, debugger/programmer section and microcontroller are highlighted.

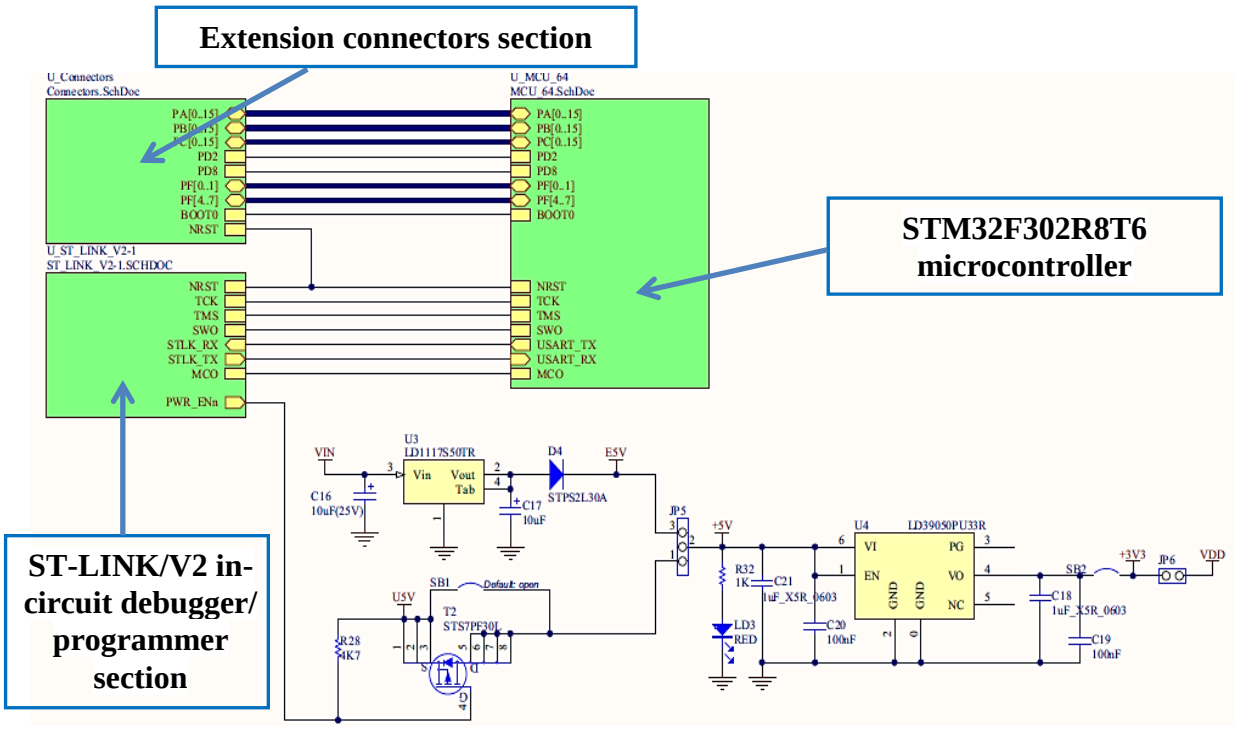

Figure 9. Electrical connections between the different sections of STM32 Nucleo board.

For connecting the expansion board in order to drive properly the BLDC motor, the STMicroelectronics Morpho connectors, highlighted in blue color in figure 10, were used. The STM Morpho connectors consist in male pin headers (CN_7 and CN_{10}) accessible on both sides of the board. They can be used to connect the STM32 Nucleo board to an extension board or a prototype/wrapping board placed on top or on bottom side of the STM32 Nucleo board. All signals and power pins of the MCU are available on STM Morpho connectors. This connectors can also be probed by an oscilloscope, logical analyzer or voltmeter.

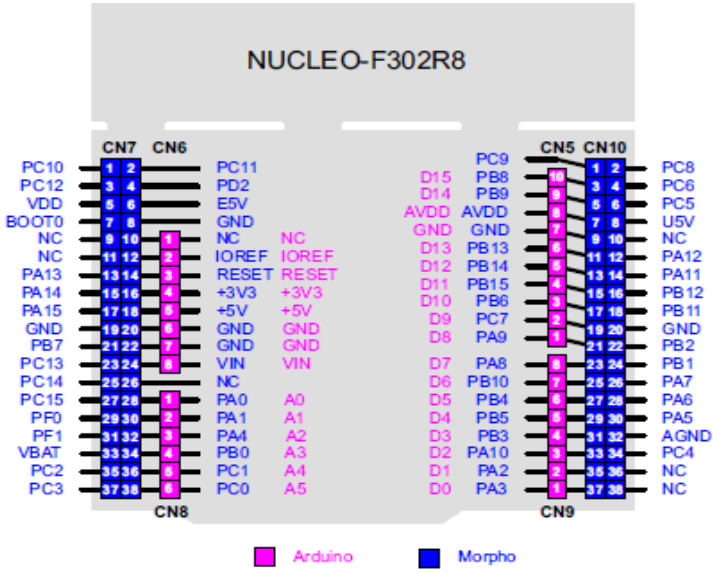

Figure 10. Extension connectors: with fuchsia color, the Arduino Uno Revision 3 connectors, instead with blue color the STMicroelectronics Morpho connectors.

✓ *X-NUCLEO-IHM07M1 ELECTRONIC BOARD FOR DRIVING BLDC MOTORS*

For driving the BLDC motor, we have chosen the X-Nucleo-IHM07M1 motor driver expansion board based on the L6230 IC driver. This last is a three-phase brushless DC motor driver and provides an affordable and easy-to-use solution for driving three-phase BLDC motors. The X-Nucleo-IHM07M1 board is compatible with the ST Morpho connectors and supports the addition of other boards which can be stacked onto the STM32 Nucleo board. The X-Nucleo-IHM07M1 is fully configurable and ready to support different closed-loop control modes for motors without or equipped with sensors; finally, it is compatible with three shunt or single shunt current measurement. In figure 11, the expansion board X-Nucleo-IHM07M1 (figure a) and its assembly on STM32 Nucleo 64-pins board (figure b) are shown.

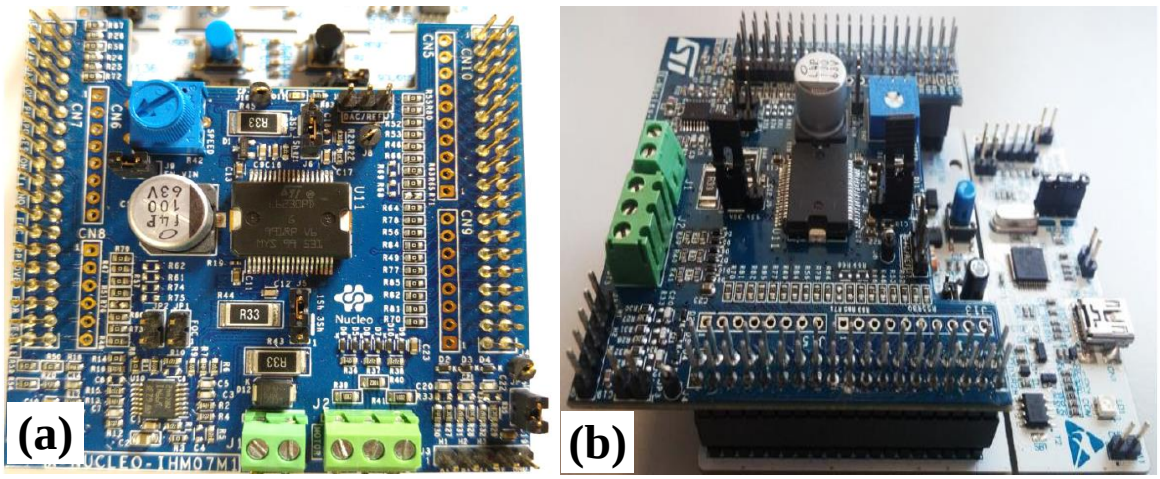

Figure 11. Nucleo-IHM07M1 motor driver expansion board (a) and its assembly on STM32 Nucleo 64-pins board through the Morpho connectors (b).

The interconnection between the STM32 Nucleo and X-Nucleo-IHM07M1 boards is fully compatible and no solder bridge modification is required. When stacked, after firmware programming, the system is ready to operate with the connection of a BLDC/PMSM motor.

The main section of the expansion driving board is based on L6230 IC, a DMOS fully-configurable driver for three-phase brushless BLDC/PMSM motors assembled in a Power-SO36 package (indicated in figure 12) with overcurrent and thermal protections. Supply voltage is provided through an external connector and it is possible to choose if digital section (STM32 Nucleo board) is supplied by USB or by the expansion board by J9 jumper setting.
The L6230 driver integrates a three-phase bridge which consists of six power MOSFETs. Using N-channel power MOSFETs for the upper transistors in the bridge, a gate drive voltage above the power supply voltage is needed. The boot-strapped supply voltage (V_{BOOT}) is obtained through an internal oscillator and a few external components to implement a charge pump circuit. The L6230 driver implements overcurrent protection with an internal detection circuit that does not require an external resistor. The current is compared with an embedded current reference and the output generates a fault condition to the DIAG pin (shown in figure 14 in the red circle) that goes to ground. This pin, connected to the STM32 Nucleo board, detects this condition and immediately disables the driving signals.

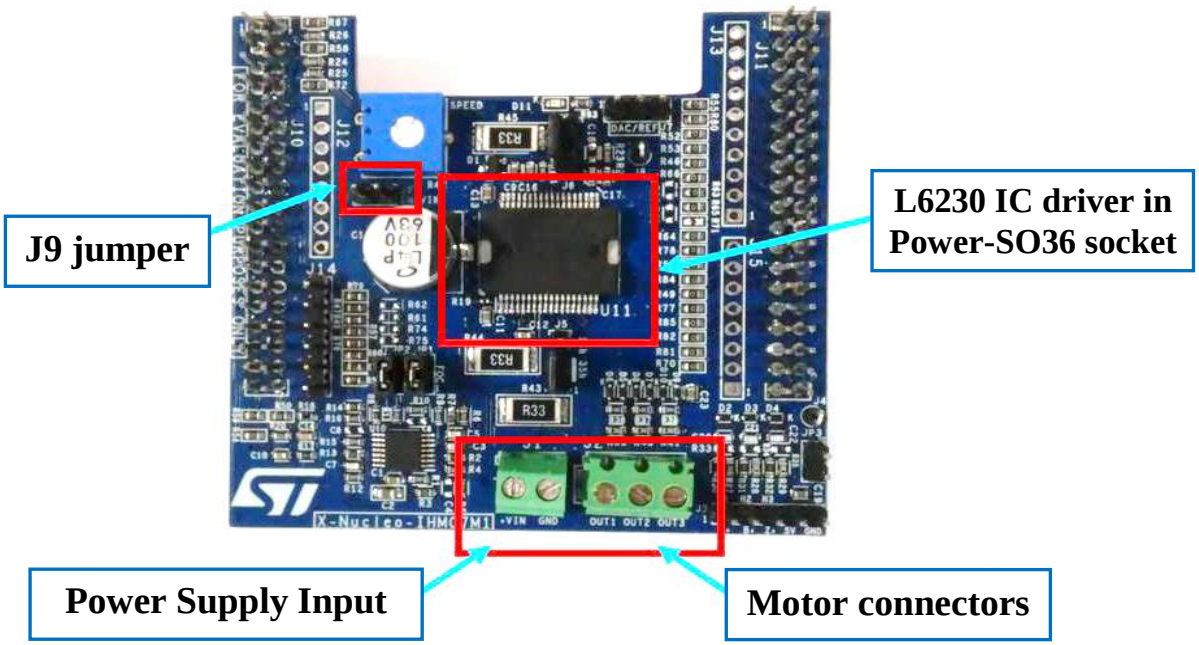

Figure 12. Nucleo-IHM07M1 expansion board with indication of the main blocks.

For detecting BEMF intensity, fundamental step for the proper operation of the BLDC motor, the following circuital section (figure 13) is used by the X-Nucleo-IHM07M1 board.

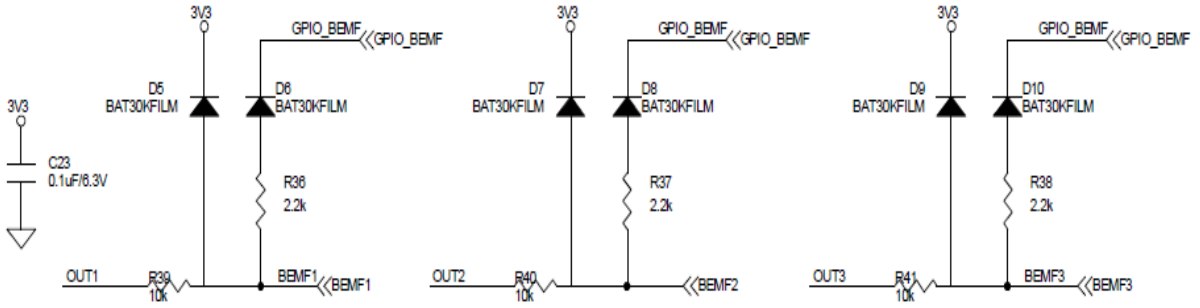

Figure 13. BEMF six-step detection circuit embedded on X-Nucleo-IHM07M1 board

There are three channels for the three phases of BLDC motor; consequently the three BEMF detected voltages are distinct and phase-shifted by 120 degrees.

The circuital section used to switch the motor phases and related to the motor control by the microcontroller is reported in figure 14. Utilizing the *six-step* algorithm explained following, through three input pins and three enable bits (highlighted in figure 14), it is possible to drive properly, in the correct sequence, the different motor phases. When each enable pin is at high logical value (set from firmware), the related logic signal with variable duty-cycle is enabled.

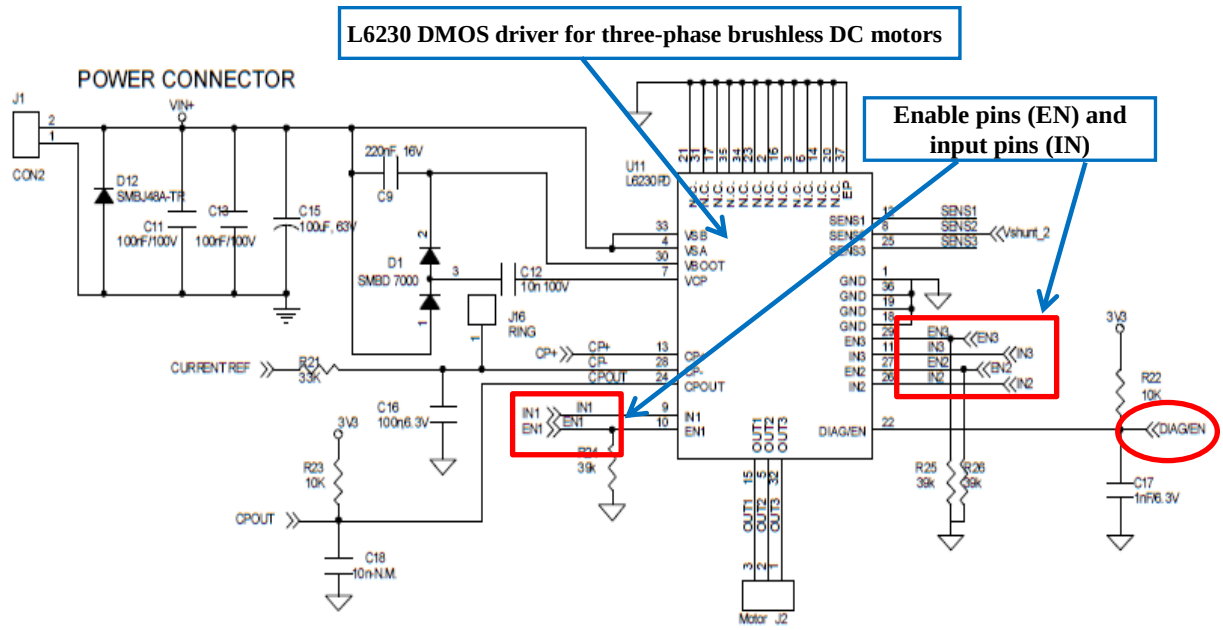

Figure 14. Circuital scheme, embedded on X-Nucleo IHM07M1 driving board, for switching motor phases based on L6230 DMOS driver.

For motor speed and spin direction measurements, the detecting circuit of Hall sensor signals, shown in figure 15, is used. The motor sensors outputs are connected to X-Nucleo-IHM07M1 board, through the J3 connector (highlighted in figure 15 in red circle), in order to detect motor movement features; a 5V voltage and GND are provided for power supply Hall sensors.

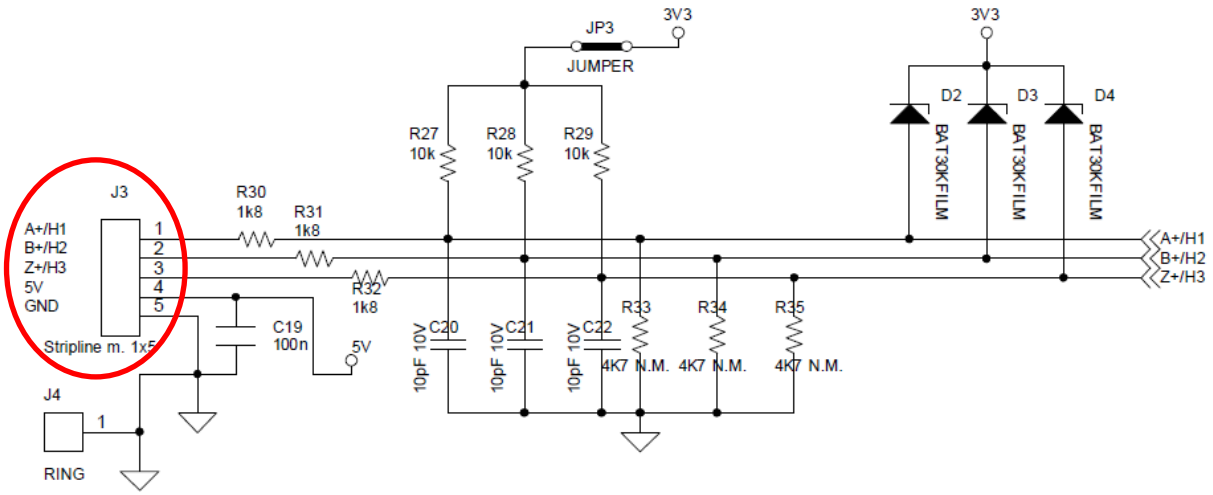

Figure 15. Detecting circuit of motor movement features embedded on X-Nucleo-IHM07M1 expansion board.

In addition, the X-IHM07M1 motor driver expansion board provides another hardware solution for motor position measuring based on sensor-less detection.

✓ *NANOTEC BRUSHLESS THREE PHASES DC -MOTOR*

A BLDC motor is highly reliable since it has no brush to wear out and replace. This motors typology is particularly well suited for applications that need smooth running and long service life. The high-energy permanent magnets (located on rotor) allow high acceleration and speeds up to 14.000rpm with exceptional efficiency. The rotor position is detected electronically using three Hall sensors, located to 120° between them. In figure 16, two images of used BLDC motor, provided by *Nanotec*, are shown.

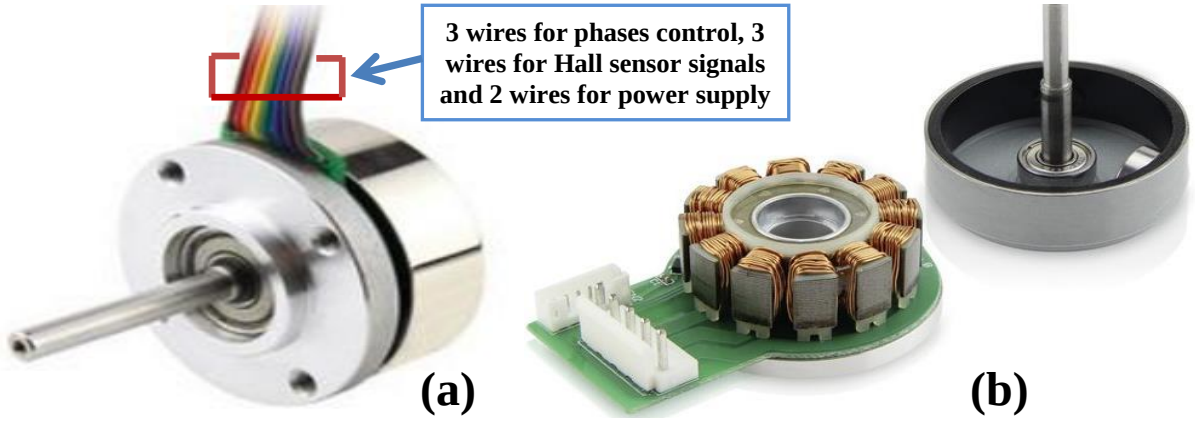

Figure 16. View of the used Brushless DC motor assembled (a) and disassembled (b).

Before analyzing the BLDC motor driving system implemented in this work, different typologies and related operating principle of BLDC motors are described in detail. Different types of BLDC motors can be used: "*internal rotor motors*" (figure 17a) in which the rotor is

rotated by permanent magnets on a shaft in an immobile stator with coils, or "*external rotor motors*" (used in this work and shown in figures 16 and 17b) in which stator is located inside and rotor consists of an externally rotating bell-shaped housing in which magnets are mounted.

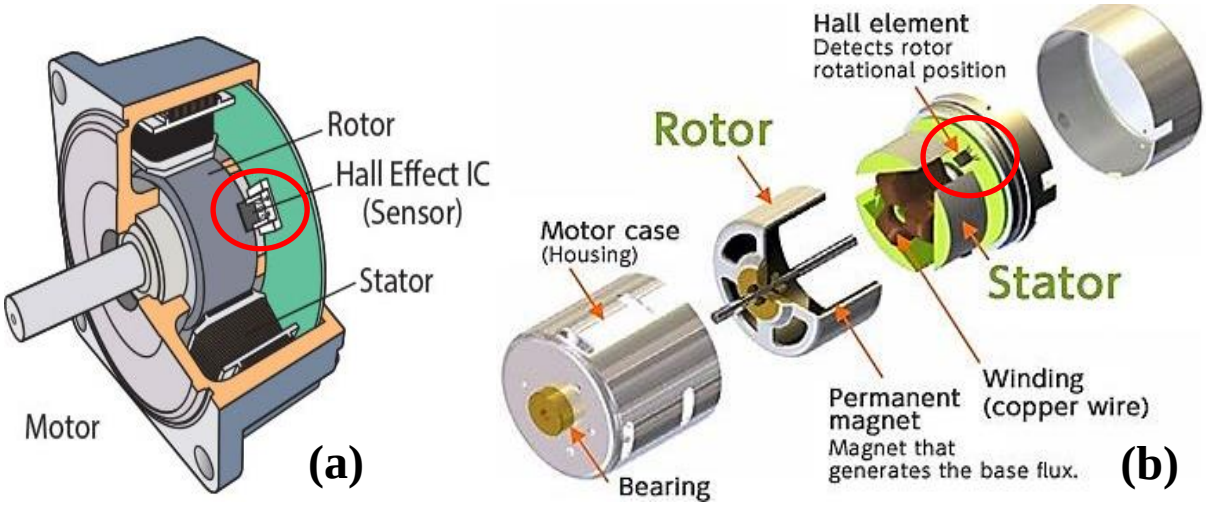

Figure 17. Different typologies of BLDC motors: *internal rotor motor* (a) and *external rotor motor* (b), both with Hall sensors to detect the position and speed of the rotating magnets.

BLDC motors are realized in single-phase, 2-phases and 3-phases configurations. The 3-phases motors are the most popular and widely used. The simplified cross sections of a single-phase and a three-phases BLDC motors are shown in figure 18.

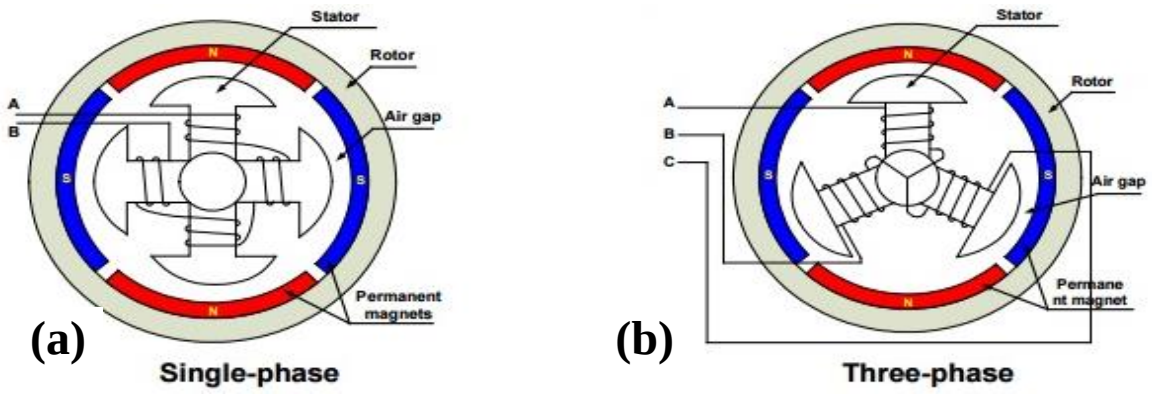

Figure 18. Simplified cross sections of a single phase and a three phases BLDC motors.

The three-phases motor, used in this research work, offers a good compromise between precise control and number of power electronic devices required to control the stator currents. For the rotor, a greater number of poles usually determines a greater torque for the same level of current. On the other hand, by adding more magnets, a point will be reached where, due to the space needed between magnets, the torque will not increase more. The manufacturing cost increases with the number of poles; therefore, the number of poles is a compromise between cost, torque and volume. The used motor, in this work, is provided by 8 pole pairs.

As reported in the introduction, there are two different types of stator windings that give rise to trapezoidal and sinusoidal signals. This differentiation is made on the basis of the interconnection of coils in the stator windings to give the different types of BEMF signals, as shown in figures 19a and 19b.

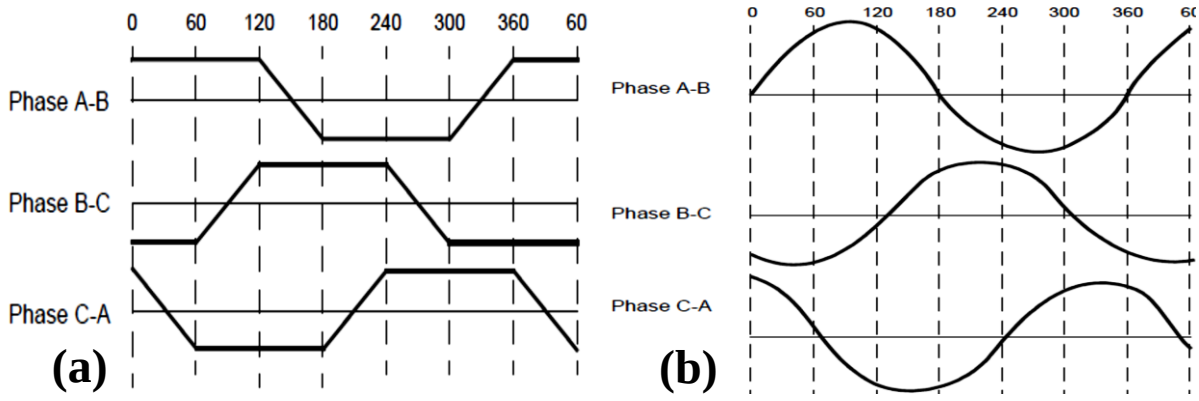

Figure 19. Trapezoidal BEMF (a) and sinusoidal BEMF (b) signals.

In addition to the BEMF, the phase current also has trapezoidal or sinusoidal shape in the respective motor types; this makes the torque provided by a sinusoidal motor smoother than that of a trapezoidal motor.

BLDC motor rotation is due to the presence of permanent magnets on the rotor which attempt to align themselves in the direction of the magnetic field generated by the stator windings during the current flow. The torque reaches its maximum when the magnetic fields (from permanent magnets and current flowing in stator windings) are perpendicular among them. During this phase, it is necessary to determine the rotor position because each current in the winding has to be correctly timed in order to keep the stator magnetic field always perpendicular to the rotor magnetic field; in this way, the motor will continue to rotate in the desired direction with the set speed.

Rotor position can be easily determined through Hall sensors embedded into the motor; these sensors can then be switched by a suitably aligned magnet on the rotor at the exact time the winding must be switched. The three windings then correspond to three Hall sensors; their states define how the windings must be connected. If the three windings are digitally switched, i.e. either there is no current or full current into the windings, this is referred to as block commutation. This combination of Hall sensors and block commutation is technically the easiest method for actuating and controlling a BLDC motor. Figure 20 shows a BLDC motor supplied from a three-phase inverter; the switching actions can be simply triggered by logic signals obtained by the Hall sensors mounted at appropriate points around the stator.

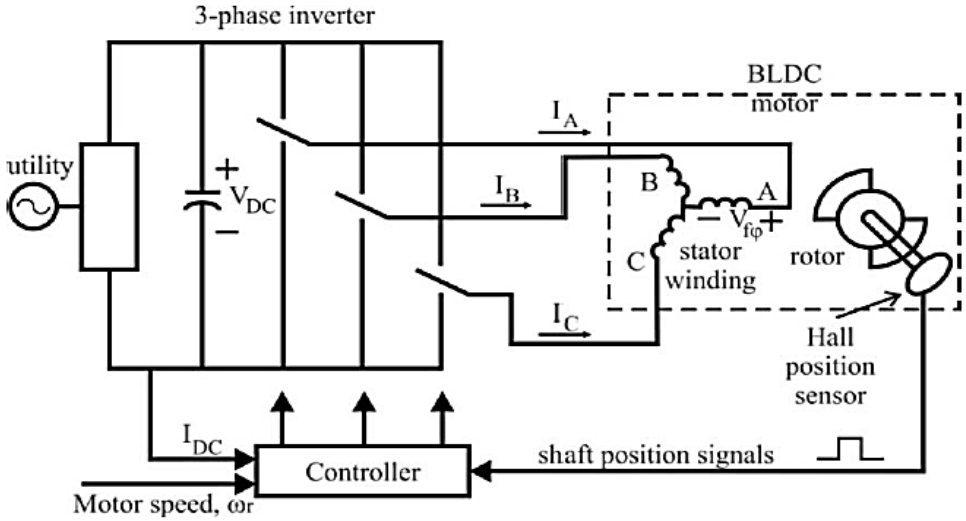

Figure 20. Electronically commutated driving system of a BLDC motor.

When mounted and aligned properly with stator phase windings, the Hall sensors deliver digital pulses that can be decoded into the desired three-phase switching sequence. This operation mode divides a rotation into six phases (3-bit code). The process of switching the current to flow through only two phases for every 60 electrical degrees rotation of the rotor is called *"electronic commutation"*.

Figure 21 shows an example of Hall sensors signals with respect to the BEMF values and phase currents. Each Hall sensor changes its state every 60 electrical degrees of rotation, therefore six steps are needed to complete an electrical cycle. However, an electrical cycle may not correspond to a complete mechanical revolution of the rotor. The number of electrical cycles to be repeated to complete a mechanical rotation is determined by the rotor pole pairs. For each rotor pole pair, one electrical cycle is completed; thus, for a complete mechanical revolution, the number of electrical cycles/rotations equals the rotor pole pairs (in the example of figure 21, one mechanical revolution requires two electrical cycles, therefore there are two rotor pole pairs). This sequence of conducting winding pairs is essential to the production of a constant output torque; each sequence has two powered windings, one positive, the other negative and the third winding opened (no current flows in this winding).

The output code of Hall sensors, at the initial instant, is 011; the phase A is zero, phase B positive and phase C negative. Subsequently, the output of sensors changes in 001 with phase A positive, B zero and phase C negative; in the following step, output code is 100 with phase A still positive, B negative and C is zero, and so on; these steps are repeated six times for each electrical cycle. The BLDC motor used in this work has 8 rotor pole pairs thus requiring, for a complete mechanical revolution, 48 electrical steps (i.e. 6 electrical cycles).

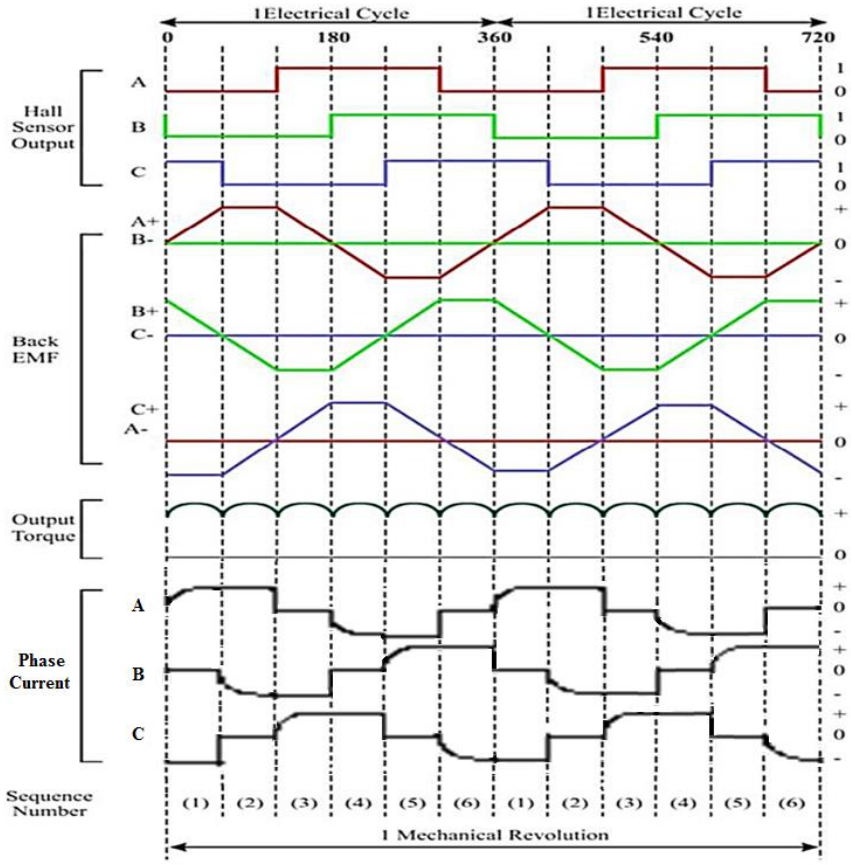

Figure 21. Hall sensor output signals, Back EMF, output torque signal and phase current signals.

IV. HARDWARE STRUCTURE OF DESIGNED SYSTEM: CONNECTION BETWEEN PC, STM32 NUCLEO, X-NUCLEO-IHM07M1 BOARD AND BLDC MOTOR

Aims of this research work are the driving and control of a BLCD motor with Hall sensors embedded in order to energize properly the motor for determining its rotation in a clockwise or anti-clockwise direction, to estimate and control the motor speed (by processing digital signals provided from Hall sensors) and finally to implement methods for the safeguard of motor itself and of realized driving electronic system. The development kit includes STM32 Nucleo board model *F302R8*, suitable for monitoring and controlling of the entire realized system by PC interfacing, the X-Nucleo *IHM07M1* motor driver expansion board required for proper feeding/driving the motor phases and for signals acquisition provided both from sensors installed on board itself and from Hall sensors embedded on BLDC motor. Obviously, the developed system is composed of a BLDC motor, *Nanotec DF45M024053-A2*, with Hall sensors embedded. The block scheme and a photo of the realized system are shown in figure 22 where the connections between the different blocks are highlighted. STM32 Nucleo board is connected to PC through a USB cable and supplied by 5V provided from USB connection.

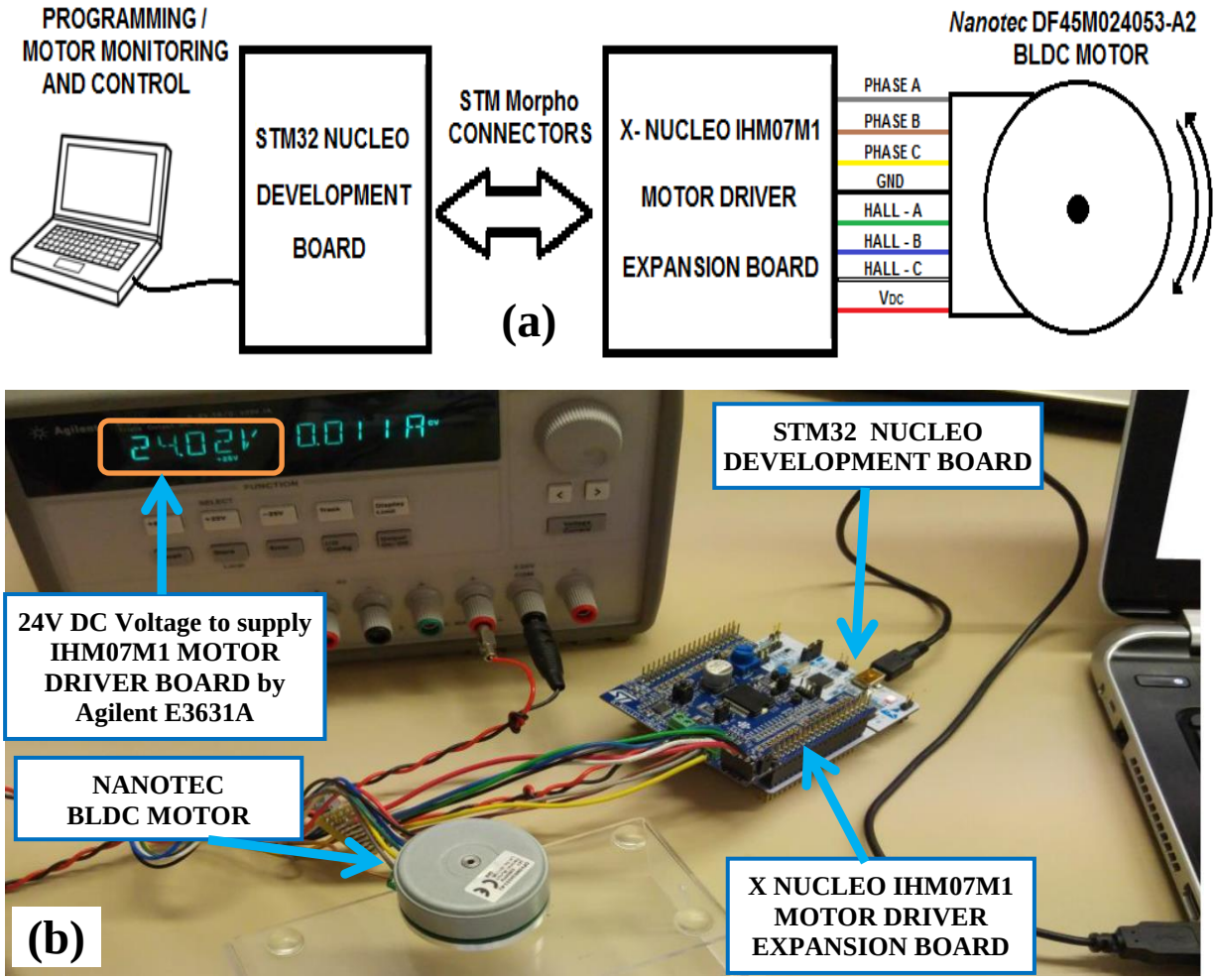

Figure 22. Block scheme (a) and photo (b) of the realized control system; BLDC motor driven by PC through X Nucleo expansion board and STM32 Nucleo development board.

The two Nucleo boards, STM32 *F302R8* and IHM07M1 motor driver expansion board, are assembled together using the STM Morpho connectors; the expansion board is powered by 24V provided from *Agilent E3631A* power supply. BLDC motor is connected to the expansion board using the connectors, present on board itself, for driving the three motor phases and for detecting Hall sensors signals; also Hall sensors are supplied by means of 5V DC voltage (red wire) and GND (black wire), as shown in figures 22a and 23. Motor phases must to be connected to board's connectors, taking care to precise order; in this way, the motor phases commutations will be out of phase by 120 degrees and the motor will function properly. If connections position is not correct, the motor would emit an annoying sound, due to current flowing in the windings, without rotating with the risk to burn it. BLDC motor is fixed onto a plexiglass base by means of screws and rubber bearings as shown in figure 22b.
In figure 23, the I/O pins connections for driving motor phases, for detecting Hall sensor signals, to allow data communication with PC and to power supply all sections are shown.

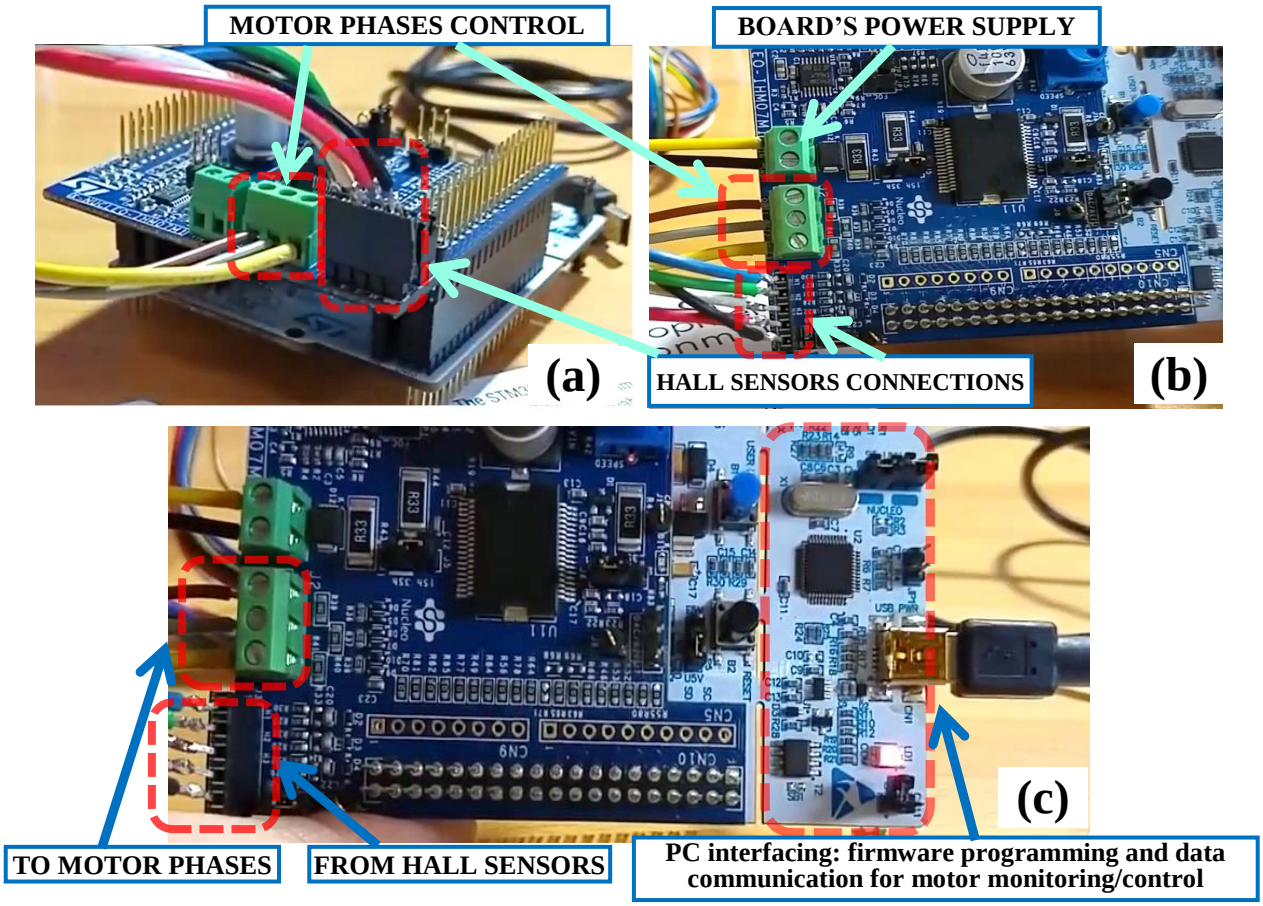

Figure 23. View of the hardware connections between X Nucleo motor driver expansion board and BLDC motor: lateral view (a) top view (b) and image with indication of PC - STM32 Nucleo board connection with on top mounted the X Nucleo expansion board (c).

Morpho connectors used for phases commutations (PA8, PA9, PA10) and for enabling power supply for each phase (PC10, PC11 e PC12), are highlighted in figure 24 in red and orange color respectively. Hall sensors signals are connected to input pins PA15, PB3 e PB10 (green color). The PC2 input pin (yellow) is used for signal acquisition from temperature sensor and the PB2 output digital pin (azure) for piloting a red LED on X-IHM07M1 board. Finally, pins PC3, PB0, PA7 (rose) are used for reading and viewing, via oscilloscope, the BEMF signals.

Figure 24. Pins connections between STM32 Nucleo and expansion IHM07M1 boards.

V. FIRMWARE IMPLEMENTATION IN ARM MBED ENVIRONMENT FOR DRIVING AND CONTROL OF THE BLDC MOTORS

Realized system allows to the user to choice motor rotation, in clockwise or anti-clockwise direction, and to set the desired rotation speed by PC terminal; in addition, the user can vary the motor speed by means of a potentiometer located on X-IHM07M1 board. Therefore, once the user determines the desired direction (by pressing "1" on keyboard for clockwise or "2" for anti-clockwise) and rotation speed (rpm value) by using PC terminal, then by pressing blue button located on STM32 Nucleo board, the motor begins rotation with a duty cycle that, starting from zero, increases (with steps of 5%) until motor reaches preset speed. A feedback control is used to detect rotation speed; if it is lower than set speed, then duty cycle increases, on the contrary if it is higher, duty cycle decreases. In figure 25, a flow-chart of developed firmware for driving and controlling of BLDC motors, by PC application, is reported.

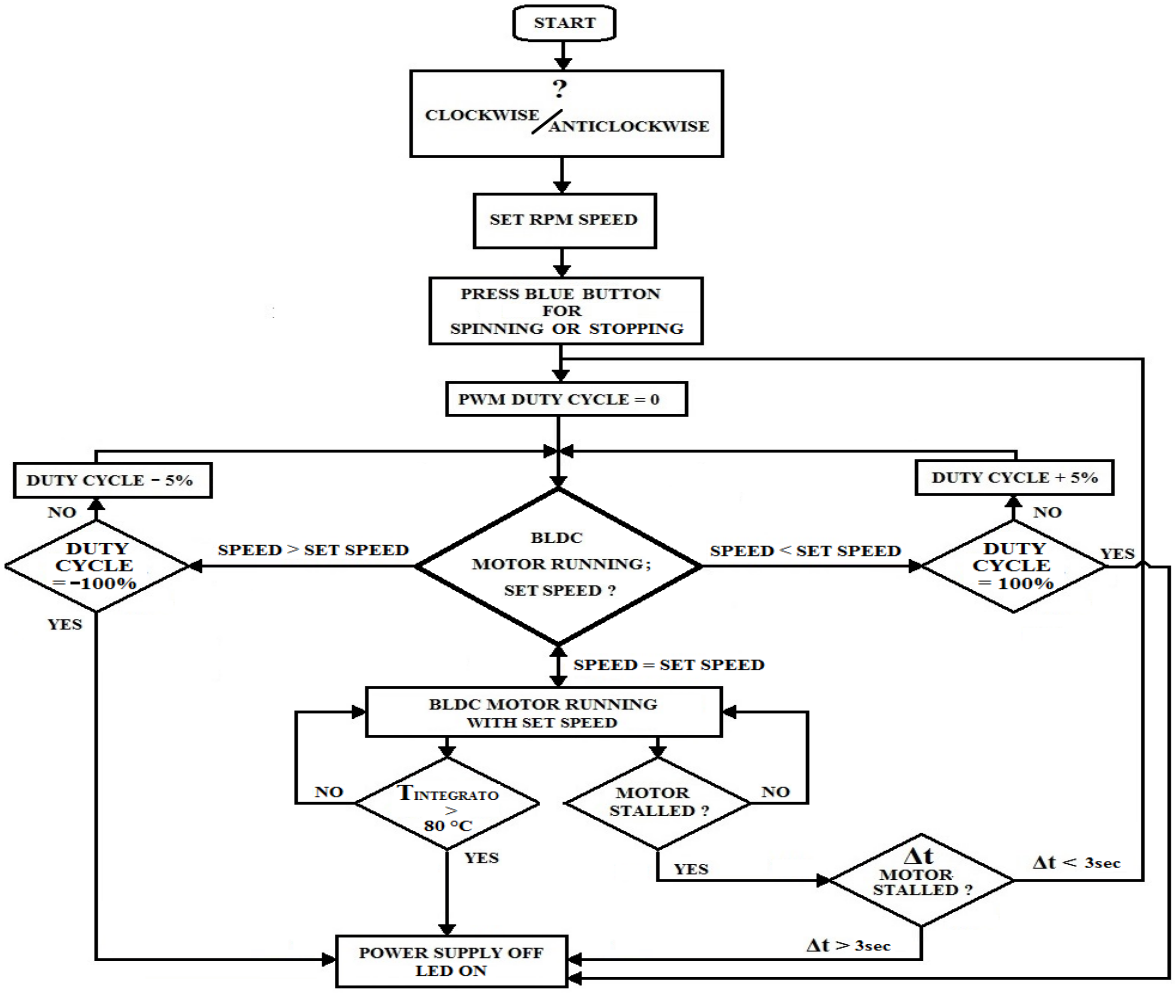

Figure 25. Flow-chart of the developed firmware for monitoring/driving BLDC motors.

Furthermore, as reported in the flow chart, realized firmware implements some controls listed following: if duty cycle is equal to 100% and motor does not reach the preset rpm value (set

speed in flow chart), to prevent motor damage, then the algorithm stops it by removing power supply. Another control is on the temperature reached by L6230 IC driver (reported in figure 12) and detected by temperature sensor embedded on X- IHM07M1 expansion board; if IC driver reaches temperature values equal or higher than 80°C, then power supply is interrupted, alerting the user with a red LED, thus safeguarding the driving system. In addition, power supply is removed if mechanical rotation is stopped for a time interval greater than 3 seconds.
Following, different sections of realized firmware are explained in detail and shown in figure 26. PWM duty-cycle and the activation of corresponding enable pin are set to the proper value as function of signals combination provided by Hall sensors. For each combination of Hall sensors output signals, a single logical combination of *Enable* and *Input* (PWM signal with variable duty cycle) pins corresponds. Six combinations of Hall sensors signals, as required by six-step algorithm, are possible; these are identified in the code as *"current sector"*, i.e. sector related to the instant in which Hall sensors provide a prefixed sequence/combination of output signals. The commutations of *Enable* and *Input* pins are performed during transition from low (0V) to high (5V) logic level of Hall sensors signals by using *"mbed"* methods (rise and fall modes) and a ticker having a pointer to the class method used for commutations. The following code, written in *ARM mbed* environment, is related to one of the three Hall sensors and to commutations of *Enable* and *Input* pins.

```
H1.rise(this, &BLDCmotorDriver::commutation);
H1.fall(this, &BLDCmotorDriver::commutation);
```

Regarding the feeding of the three motor phases, a *switch* instruction inside an *if* instruction was used; each *case* of *switch* corresponds to a specific rotor sector and to a precise sequence for providing on output pins the variable duty-cycle signal; the code relative to *rotor sector "0"* is reported in figure 26a. For rotation in anti-clockwise direction, the principle is the same but sectors have a negative sequence and commutations are made in an opposite mode, as shown by the code in figure 26b. The rotations counting is performed taking into account the commutations of logic signals provided by Hall sensors. Furthermore, for counting number of mechanical rotations, as function of time, a ticker was employed instead of a normal timer because, sometimes, the timer provided a random value which didn't correspond to real motor speed. The *timerBLDC class* (reported in the section *c* of figure 26) determines the right time interval needed to calculate the motor speed value (also turning on a LED for this time). In fact, from code implemented in the main file, by counting commutations of Hall sensors signals in the calculated time interval, and then dividing for the time interval itself, for the poles number and for six (number of electrical cycles), as reported in section *d*, the motor

speed value is obtained. This result is then multiplied by sixty to get the rpm speed value The poles number was set as a variable in order to proper modify it if a different motor is used.

```
if (dutyCycle > 0 ) {                                   (a)
        currentSector++;
if(currentSector > 5)
        currentSector = 0;
switch(currentSector) {
case 0:
  EN1 = 1; IN1 = dutyCycle;
  EN2 = 0; IN2 = 0;
  EN3 = 1; IN3 = 0;
break;
```

```
else if (dutyCycle < 0) {                               (b)
        currentSector--;
if(currentSector < 0)
        currentSector = 5;
switch(currentSector) {
case 1:
  EN1 = 0; IN1=0;
  EN2 = 1; IN2 = 0;
  EN3 = 1; IN3 = -dutyCycle;
break;
```

```
float BLDCmotorDriver::getCycles(){                     (d)
        int cycles_per_step = 6;
int revol = cycles/(cycles_per_step*poles);
return revolution; // number of visible cycles

float time = BLDCtimer(0,0,0).counter()/1000;
                    // sampling time in sec
int cycle  = M.getCycles();
float sampling = (60*cycle)/time;
```

```
float BLDCtimer::counter() {
tick_ticker.attach (this,&BLDCtimer::inc_tick, 0.001);
uint32_t  begin, end;
timer.start();
int t = timer.read();
while (1) {
        begin = ticks;
while(myled1==1);
        end = ticks;
float time = end - begin;                               (c)
return time;
```

```
if(sampling < a && b == 1 && dutyCycle <= 1.00f) {
M.setDutyCycle(dutyCycle);
dutyCycle += 0.05f;                          clockwise control
M.setDutyCycle(dutyCycle);
printf("dutyCycle  = %f \n", dutyCycle );
}else if(sampling>a && b==1 && dutyCycle<=1.00f) {
dutyCycle -= 0.05f;
M.setDutyCycle(dutyCycle);
printf("dutyCycle  = %f \n", dutyCycle );
```

```
} if(sampling <a && b ==-1 && dutyCycle >= -1.00f){
M.setDutyCycle(-dutyCycle);
dutyCycle -= 0.05f;                          anticlockwise control
M.setDutyCycle(dutyCycle);
printf("dutyCycle  = %f \n", dutyCycle ); }
else if(sampling>a &&b==-1&&dutyCycle>= -1.00f){
dutyCycle += 0.05f;
M.setDutyCycle(dutyCycle);
printf("dutyCycle  = %f \n", dutyCycle ); }          (e)
```

Figure 26. Firmware portions for controlling BLDC motor: code relative to rotor sector for clockwise (a) and anti-clockwise (b) direction, *timerBLDC class* for time calculation (c), code related to rpm speed value (d) and for controlling speed and PWM duty-cycle (e).

Section *e* of figure 26 is related to motor speed variation, in clockwise or anti-clockwise direction, by increasing/decreasing PWM duty-cycle; the direction is determined by *b* variable set as function of the key pressed by user. In the *if* instructions, controls are performed on *a* variable (that represents *rpm* current value), on *b* variable and on PWM duty-cycle that is increased (or decreased) until the preset *rpm* value is reached (despite the presence of loads). In addition, as previously reported, the firmware implements other control methods such as the 100% duty-cycle control, the temperature control of L6230 IC driver and a mode for removing power supply if mechanical rotation is blocked for a time period greater than 3 sec.

The firmware was developed in *ARM mbed* environment available on cloud; using internet network, after registration, the user can compile its own project. *Mbed* development environment, shown in figures 27, was initialized for STM32 Nucleo board (model *F302R8*, used in this work), thus allowing proper recognition when the firmware, through a normal drug and drop operation, will be uploaded on embedded PIC, regarded the STM32 Nucleo board by operating system as an external memory (figure 27b). The firmware output is

displayed on a suitable terminal, *Tera-Term* in this case, by means of USB serial communication between PC and STM32 Nucleo board. The programming language used in mbed environment is the object oriented language C ++.

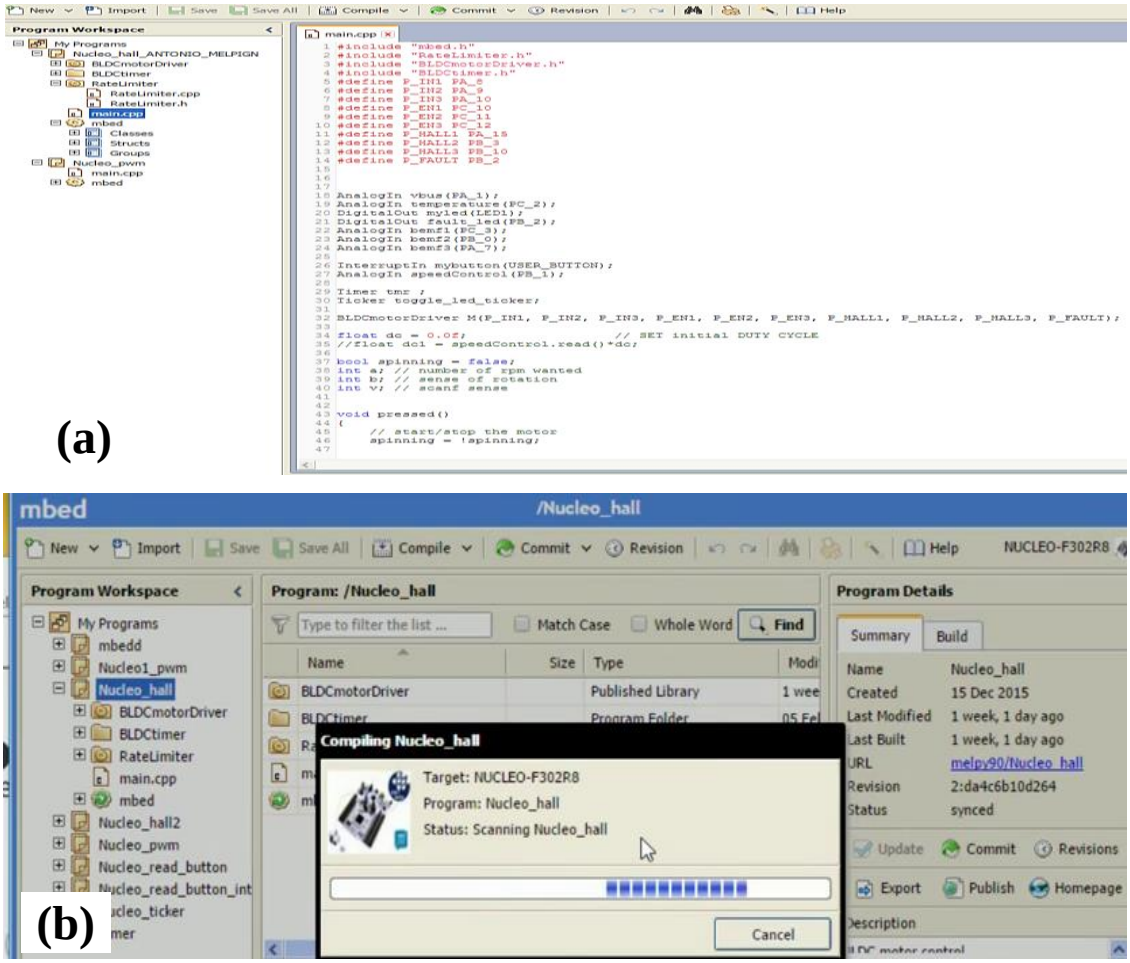

Figure 27. Screenshot of ARM mbed environment for writing code (a) and firmware compilation to generate and upload the .bin file to virtual disk created by STM32 Nucleo (b).

VI. HARDWARE AND FIRMWARE TESTING FOR BLDC MOTOR CONTROL AND SPEED DETECTION BY PC INTERFACING

In order to verify the proper operation of realized system, a *Picoscope 3000* digital oscilloscope was used; in figure 28, the most significant I/O pins are connected to analog/digital probes (yellow wires) for oscilloscope viewing. The digital probes are connected to *enable*, PWM variable duty-cycle pins and Hall sensors outputs, whereas analog probes are connected to BEMF signals pins (positions of cited pins were shown in figure 24). In this way, it was possible to observe signals trend during motor speed variation. In figure 29, temporal trend of D3-D4-D5 PWM signals related to input pins of IHM07M1 driving board, of D0-D1-D2 enabling pins and of D6-D10-D9 signals from Hall sensors are reported.

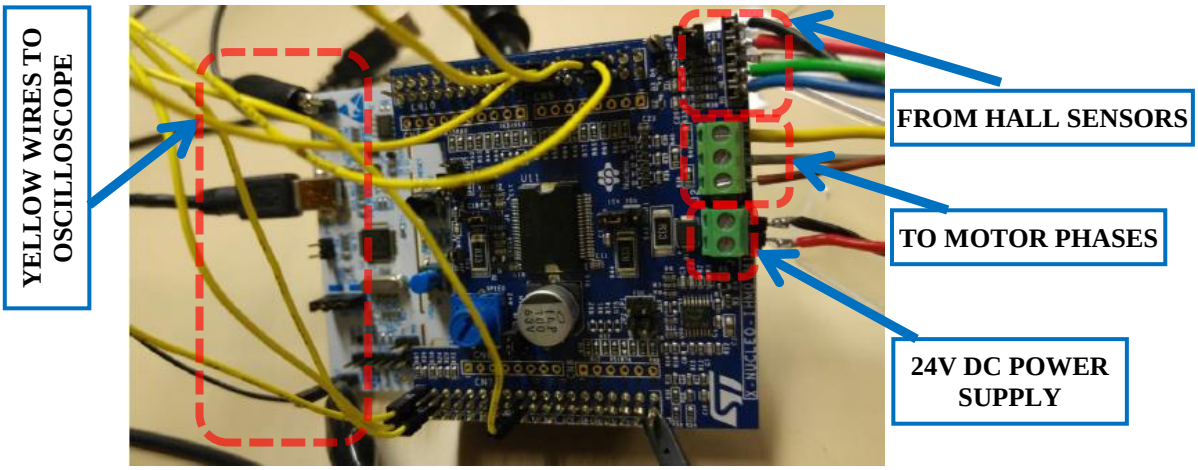

Figure 28. STM32 Nucleo board assembled with motor driver board connected to PC and BLDC motor during operation; yellow wires connected to oscilloscope for signals acquisition.

As shown in figure 29, soon after D6 signal (output signal form Hall sensor) is low, D3 PWM signal is disabled and D4 PWM input signal is enabled by D1 signal activation. In other words, D0-D1 and D2 signals enable PWM input signals as function of Hall sensor outputs commutations. By varying the potentiometer value, located on IHM07M1 driving board, therefore the PWM duty-cycle changes and also the motor rotation speed.

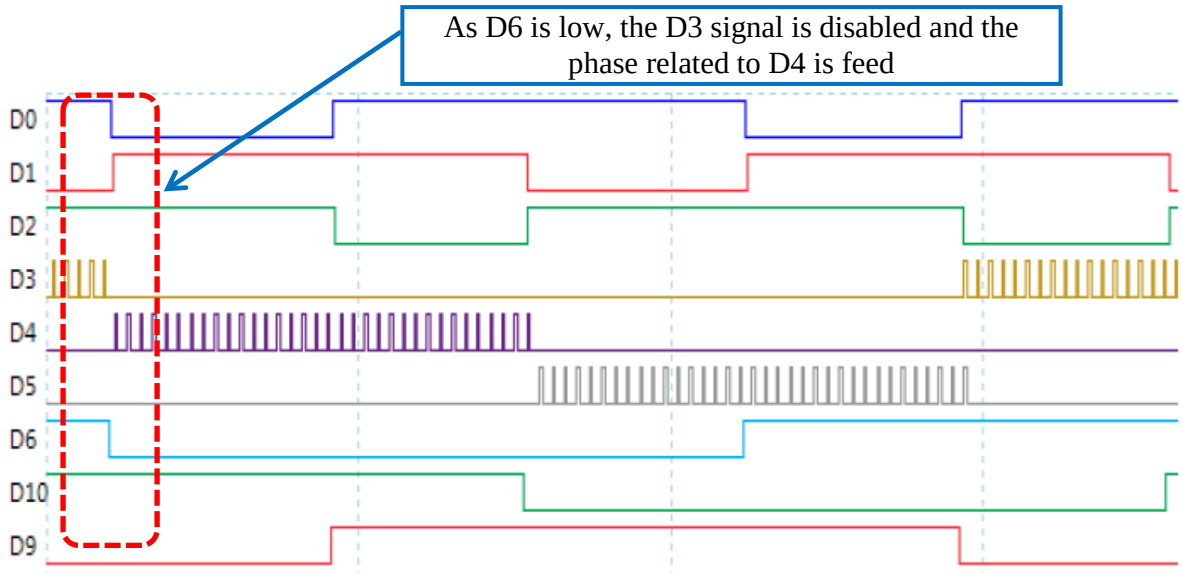

Figure 29. Signals taken by oscilloscope during motor operation: D0, D1 and D2 are enabling signals for D3, D4, D5 PWM signals respectively. D6, D10, D9 are Hall sensors signals

Figure 30 reports a screenshot of oscilloscope display taken during motor operation with duty-cycle equal to 60%. As shown in the figure, the BEMF signals are phase shifted by 120 electrical degrees, as can be revealed by considering the temporal trend of Hall sensors signals (D6-D10-D9), confirming that motor phases are correctly powered. In particular, since Hall sensors change state once every 60 electrical degrees, considering two commutations of D9

signal (delimited by vertical dashed lines), it is evident that BEMF analog signal repeats after 120 electrical degrees corresponding to two state commutations of D9 signal. The slight upward spike, highlighted with circles in the graph of BEMF signals, is due to the presence of *zero-crossing* phenomenon, occurring when BEMF signal passes through zero, during Hall sensor commutations.

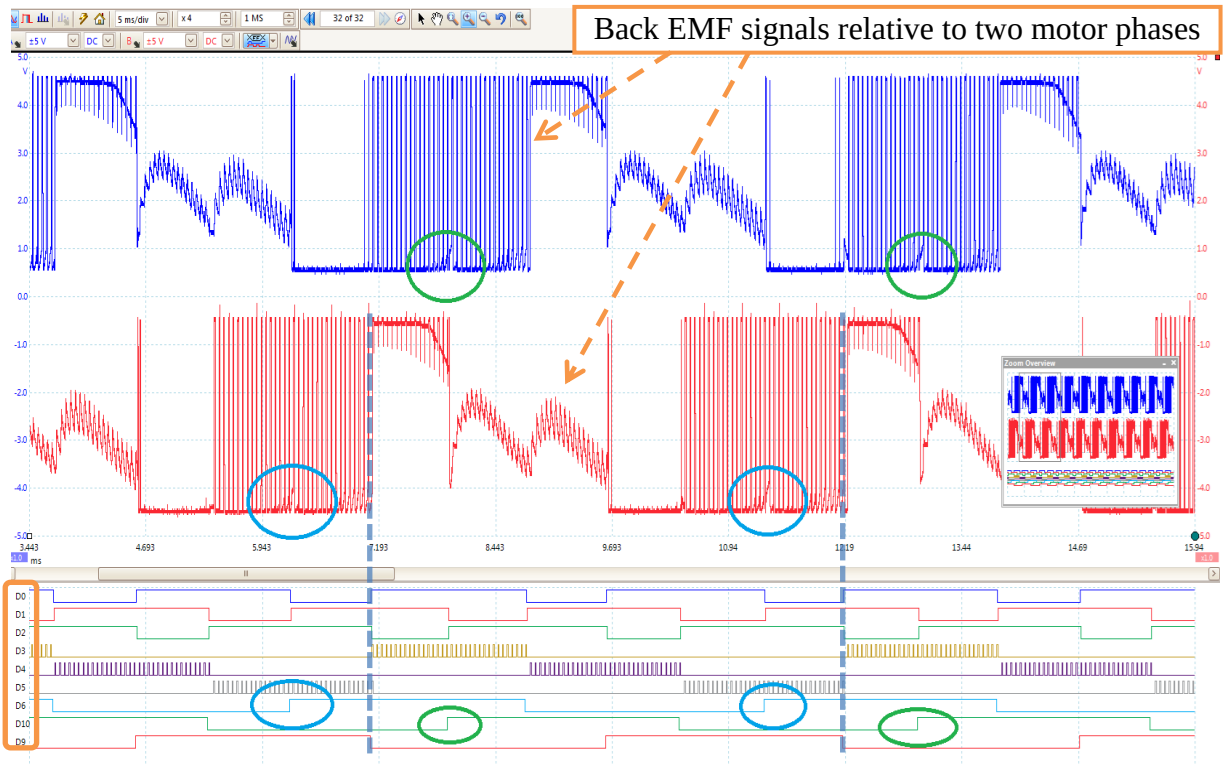

Figure 30. Oscilloscope screenshot during motor operation with shown temporal behavior of significant signals; the circles indicate the zero-crossing time instants.

Different tests were carried out to verify the proper function of BLDC motor; from a standing start, once inserted the parameters related to desired direction and rpm value, by pressing the on board button, the motor starts the rotation. In figure 31, a photo of BLDC motor connected to the driving system before starting the rotation and a screenshot of PC terminal with received data by USB connection with STM Nucleo board are reported. It is possible to see, on the right in figure, the selected clockwise direction, the set rpm value (480) and the L6230 IC driver temperature value (33.57°C) just before starting rotation. Also the duty-cycle value is reported, in this case, equal to 5%. Soon after starting rotation, data on PC terminal are updated (figure 32); it is possible to see that motor is reaching, in a short time, set rpm value being the PWM duty-cycle increased up to 25% whereas temperature value does not undergo significant variations. In the performed tests, it was verified that motor stops (by removing its power supply), when PWM duty-cycle value is 100% without reaching set speed value.

Figure 31. View of BLDC motor stopped before starting rotation (on the left) and screenshot of received data on PC by serial communication with STM32 Nucleo board (on the right).

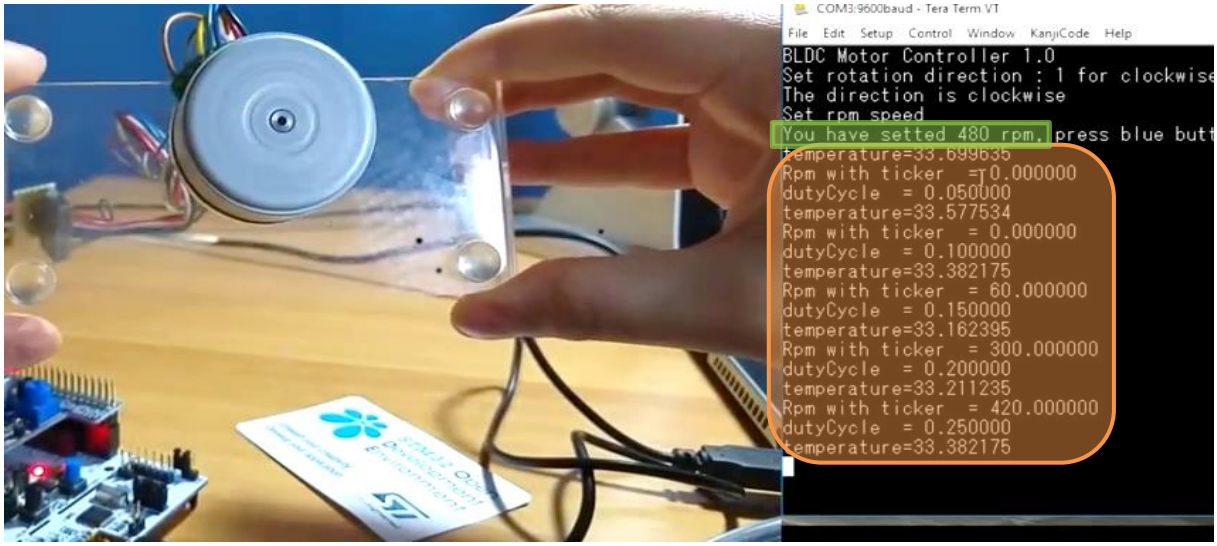

Figure 32. BLDC motor during rotation (on the left) and screenshot of PC terminal with data received from STM32 board (on the right) with increasing duty-cycle to reach set rpm value.

Figure 33 shows the motor operation changing rpm speed by varying the on board potentiometer value. As reported in the screenshot of PC terminal, the L6230 IC temperature, in this case, changes as function of potentiometer value variation and thus of the rpm speed; this is due to increased L6230 driver solicitation which determines its greater heating. In another test, it was verified that motor resumes the rotation after being blocked for a time interval lower than 3 sec; the rotation starts again reaching the initial rpm speed value. In this way, it was tested that realized driving system works properly, PC receives data from board relative to duty-cycle variation as function of set rpm or potentiometer value and relative to L6230 IC driver temperature so obtaining complete control and monitoring of motor rotation.

Figure 33. BLDC motor rotation with variable speed by changing on board potentiometer value; on the right, the variations of motor rpm value and temperature of L6230 IC driver.

VII. CONCLUSIONS

In this research work, an electronic system for driving and controlling a BLDC motor, with Hall sensors embedded, was realized. A BLDC motor, thanks to its properties of low maintenance costs, compact size, high reliability and efficiency, and low power consumption, is employed and its use is increasing, in many fields such as appliances, automotive, aerospace, medical, industrial automation equipment and instrumentation. The realized driving system is based on the STM32 Nucleo development board and on X-Nucleo IHM07M1 motor driver expansion board provided by ST Microelectronics. By PC connected via USB with STM32 board, it is possible to set motor rotation direction, rpm speed value and by potentiometer located on board to change rotation speed. A firmware to control motor operation, as function of signals provided by Hall sensors, was developed in *ARM mbed* environment. A feedback control was realized to detect rotation speed; if it is lower than set speed, then duty cycle is increased, on the contrary if it is higher, duty cycle is decreased. Different controls are performed during motor operation: if PWM duty-cycle value is equal to 100% without reaching set speed, then power supply is removed, if BLDC motor is stalled for a time greater than 3 sec, also power supply is removed to safeguard motor/system integrity. The L6230 IC driver temperature is detected to avoid IC damages. Utilizing realized system, by adding STM Bluetooth Low-Energy expansion board, it will be possible to monitor and control the BLDC motor wirelessly by smartphone or tablet. Finally, this experimental work can be extended to drive and control BLDC motors without Hall sensors embedded, by using the BEMF variation as input signals for determining motor phases commutations.

REFERENCES

[1] P. Visconti, A. Lay-Ekuakille, P. Primiceri, G. Cavalera: Wireless Energy Monitoring System Of Photovoltaic Plants With Smart Anti-Theft Solution Integrated With Control Unit Of Household Electrical Consumption. International Journal on Smart Sensing and Intelligent Systems, Vol. 9, No. 2, pp. 681 – 708, (2016).

[2] P. Visconti, R. Ria, G. Cavalera: Development Of Smart Pic-Based Electronic Equipment For Managing And Monitoring Energy Production Of Photovoltaic Plan With Wireless Transmission Unit. ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 20, pp. 9434 – 9441, (2015).

[3] R G. Balakrishna, P Y. Reddy: Speed Control of Brushless DC Motor Using Microcontroller. International Journal of Engineering Technology, Management and Applied Sciences, Vol. 3, Issue 6, pp 11-26 (2015).

[4] M. Rao : Energy efficient Ceiling fans using BLDC motors - A practical implementation. Proc. of the Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering. Editor In Chief Dr. R. K. Singh. Copyright © 2012 Universal Association of Computer and Electronics Engineers, pp 59-63, DOI:10.3850/978-981-07-1847-3 P0369, (2012).

[5] P. Visconti, P. Costantini, G. Cavalera: Design of electronic programmable board with user-friendly touch screen interface for management and control of thermosolar plant parameters. 15th International Conference on Environment and Electrical Engineering (EEEIC), IEEE Publisher, doi:10.1109/EEEIC.2015.7165553 (2015).

[6] M. Heidir M. Shah, M. Fua Rahmat, A.Kumeresan, Danapalasingam and Norhaliza Abdul Wahab: PLC Based Adaptive Fuzzy Pid Speed Control of DC Belt Conveyor System. Int. Journal On Smart Sensing And Intelligent Systems, Vol. 6, No. 3, pp. 1133 – 1152 (2013).

[7] A. Purna, C. Rao, Y.P.obulesh, C. Sai babu: Performance Improvement of BLDC Motor with Hysteresis Current Controller. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 12, pp. 5900 – 5907, (2013).

[8] M. Njah, M. Jallouli: Improving The Localization of Electric Wheelchair by using Particle Filter. Int. Journal on Smart Sensing and Intelligent Systems Vol. 7 (4) pp. 1923–1942 (2014).

[9] Arunkumar and Thangavel: A Review Paper on Torque Ripple Reduction and Power Quality Improvement in Brushless DC Motor Drives. International Electrical Engineering Journal (IEEJ), Vol. 5, No.10, pp. 1567-1575, (2014).

[10] R. K. Srivastava, S. K. Singh, A. Dwivedi, S. Gollapudi: Pm Enhanced Sensing Of Internal Emf Variation- A Tool To Study PMBLDC/AC Motors. International Journal On Smart Sensing And Intelligent Systems, Vol. 6, No. 4, pp. 1456 – 1478 (2013).

[11] S. A. Shahbaz: Ambulance Drone Support System (ADSS). International Journal of Engineering Science and Innovative Technology (IJESIT), Vol. 4, Issue 3, pp. 80 –89, (2015).

[12] M. Azadi and A. Darabi: Speed Control of an Eleven-Phase Brushless DC Motor. International Journal of Information and Electronics Engineering, Vol. 3, No. 4, pp. 374 – 378. DOI: 10.7763/IJIEE.2013.V3.338, (2013).

[13] M. A. Enany, H. M. Elshewy, F. E. Abdel-kader: Brushless DC Motor Performance Improvement through Switch-on and Switch-off Angles Control. Proceedings of the 14th International Middle East Power Systems Conference (MEPCON'10), Egypt, Web-site: http://sdaengineering.com/MEPCON10/Papers/285.pdf, (2010).

[14] W. Benrejeb, O. Boubaker: Fpga Modelling And Real-Time Embedded Control Design Via Labview Software: Application For Swinging-Up A Pendulum. International Journal On Smart Sensing And Intelligent Systems, Vol. 5, No. 3, pp. 576 – 591 (2012).

[15] X. Wu, P. Lou, K. Shen, G. Peng, D. Tang: Precise Transhippment Control Of An Automated Magnetic-Guided Vehicle Using Optics Positioning. International Journal On Smart Sensing And Intelligent Systems, Vol. 7, No. 1, pp. 48 – 71 (2014).

[16] O. O Laxman, G. Joshi: Bldc Motor Speed Control Using Co-Simulation Of Multisim and LABVIEW. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), Vol. 4, Issue 2, pp. 35 – 39, (2016).

[17] T.Hemanand, T. Rajesh: Speed Control of Brushless DC Motor Drive Employing Hard Chopping PWM Technique Using DSP. Proceedings of India International Conference on Power Electronics 2006, Chennai, DOI: 10.1109/IICPE.2006.4685404, (2006).

[18] I. Sloan: Brushless dc Motors in Medical Applications. NMB Technologies Corp., Chatsworth, Calif., Edited by Robert Repas, Machine Design. Web-site: http://machinedesign.com/motion-control/brushless-dc-motors-medical-applications, (2009).

[19] M. M. Meenu, S.Hariharan: Position Sensorless Control of BLDC Motor in Continuous Positive Airway Pressure Device. International Conference on Control, Communication & Computing India (ICCC), Trivandrum, DOI: 10.1109/ICCC.2015.7432897, (2015).