Controller for BLDC Motors

Florin Ravigan[*], Laurentiu Alboteanu[*], Marius Dumitriu[**], Ionut Zglimbea[**]
[*]Department of Electromechanical, Environmental and Applied Informatics,
Faculty of Electrical Engineering, University of Craiova, Craiova, Romania
florin.ravigan@ie.ucv.ro, lalboteanu@em.ucv.ro
[**]Department of Research And Development, Nextrom Industries SRL, Ghercesti, Romania
marius.dumitriu@nextpack.ro, ionut.zglimbea@nextpack.ro

***Abstract*—In this paper it is presented a comprehensive design of a dedicated controller for BLDC motor with maximal electric power around 1000W. That controller is a new product in testing stage and it uses high-end components to provide high performance and more flexibility of the implemented algorithms at a low-level cost. The controller can drive a BLDC motor providing the possibility of rotating in both directions, it creates an electromagnetic brake and kinetic energy recovery system. The controller has equipped a 500W electric scooter, and a 1000W electric motor cart and it proved to be reliable, performance and especially flexible. All software routines make the code very fast portable by parameterizing all the implemented software functions.**

Cuvinte cheie: motor brushless, hardware, software, KERS, Infineon, microcontroller.

Keywords—*BLDC, hardware, software, KERS, Infineon, controller.*

I. INTRODUCTION

Brushless DC electric motor (BLDC motors) also known as electronically commutated motors (ECMs), or synchronous DC motors, are synchronous motors powered by DC electricity from an inverter which produces an AC electric current to drive each phase of the motor via a closed loop controller.

The controller needs the rotor's position (relative to the stator) to implement the traditional brushes' functionality. Some designs use Hall effect sensors or a rotary encoder to directly measure the rotor's position. Others measure the back-EMF in the undriven coils to infer the rotor position, and therefore are often called sensor less controllers [7]. The first, due to the development in the field of electronics, solution is cheaper and easier to implement.

The control loop also needs to measure the current in motor windings and have to provide the protection at overcurrent and also to prevent cross-conduction of inverter bridge controls. For the current measurement it is used a solution based on a shunt resistance and an operational amplifier whose output is evaluated by an ADC.

These requirements impose the use a powerful processor and response loops serviced by optimized software routines.

That kind of motor is used in a very large range of applications. One of those is the production of two-wheel electric vehicles (electric scooters).

II. HARDWARE DESCRIPTION

The development of the controller, it started by identifying the initial data and setting its operating parameters based on the range of powers to be used.

The motor used are BLDC type with three phase and 120-degree hall position sensors(fig.1). The powers imposed on the used motors are in the 300W and 1000W range, and power source was a LiPo battery with U_N=36V.

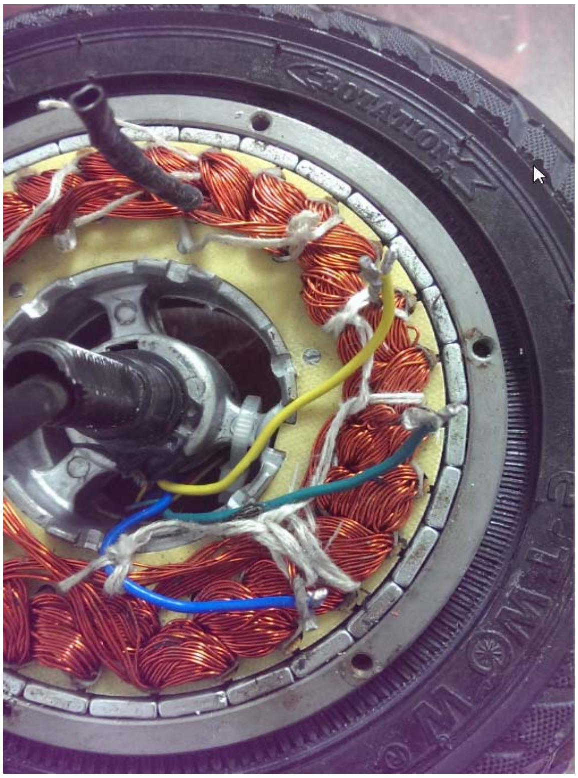

Fig.1 The BLDC motor used to test the controller

Thus, the following features resulted:

- Usable for BLDC motor, with power between 300W and 1200W with sensors positioned at 120 degrees electric;
- Maximum supply voltage: 60 V;
- Maximum allowable current: 30A, limited at 16 A;
- Energy efficiency> 90%;
- Power consumption without motor: under 1,5 W;
- Speed and torque control;

- Possibility of speed limitation;
- ARM Cortex architecture microprocessor control;
- Overcurrent protection for the motor;

Serial communication interface.

The block diagram of the controller is shown in figure 2.

The BLDC motor used in design and tests has following parameters:

- Nominal voltage: U_N= 24 ÷ 36 V;
- Nominal current: I_N=16 ÷ 20A;
- Position information: 3 Hall effect sensors at 120 degrees;
- Number of stator poles: 27 with 30 permanent magnets on rotor.

The controller is composed of a part of the power electronics and a part that ensures the control of its operation according to the imposed algorithm.

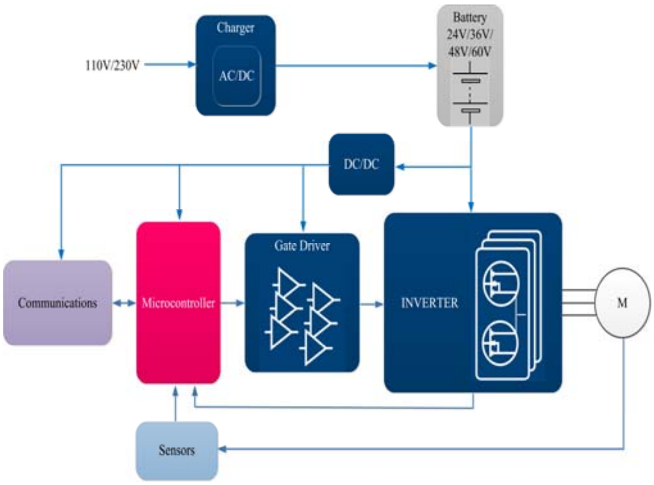

Fig.2 The block diagram of the controller

Power electronics is represented by the 3-phase inverter with 6 MOSFET transistors and it has the following architecture (fig.3). In some situations, the 6 transistors can be doubled to increase the current value.

By command the motor current can have a trapezoidal or sinusoidal evolution. The trapezoidal evolution was chosen because it offers better torque at higher speeds.

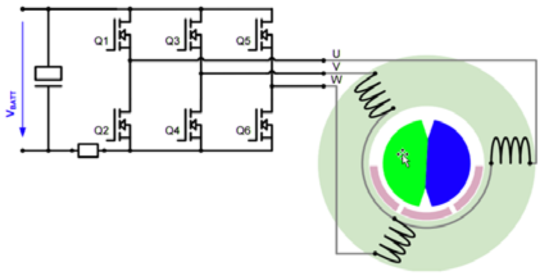

Fig.3 The 3-phase inverter

The control of the MOSFET transistors is made by the microcontroller via a dedicated driver from Infineon. Two solutions have been identified:

-driver for 6 single-chip MOSFETs in the 6EDL family;
-driver for 2 MOSFETs per 2ED family phase.

The first tested driver is 6EDL04N02PR and it has the following features [10]:

- Thin-film-SOI-technology;
- Maximum blocking voltage +600V;
- Separate control circuits for all six drivers;
- CMOS and LSTTL compatible input (negative logic);
- Signal interlocking of every phase to prevent cross-conduction;
- Detection of over current and under voltage supply;
- Externally programmable delay for fault clear after over current detection;

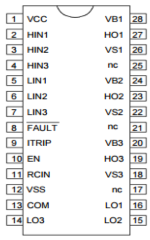

Fig.4 The package of the driver 6EDL04N02PR

The first solution which has been identified the advantage of on-chip hardware implementation of a current peak protection function called ITRIP.

From the dimensioning of a component group (C_{RCIN} in combination with R_{RCIN}), you can set the peak current value and driver intervention time [10].

That controller has to read the position of the motor and to compute the proper command sequence for the bridge inverters. In most cases, the position of the rotor is detected using Hall elect sensors.

These Hall sensors are located at 120 electrical degrees and they have a response time around 25μs.

How to use the driver is shown in the fig.5. It provides 6 inputs for command signals generated by microcontroller and 6 outputs for transistors' gates. Also, it has a current reaction input and a digital output (FAULT) active in case of malfunctioning.

At the base of the configuration for ITRIP function, was used the driver datasheet. This is represented by a document provided by Infineon and contains the necessary information for calculating the passive elements that determine the value of its intervention (maximal current and the time after that the protection unlock the commands) (fig.6).

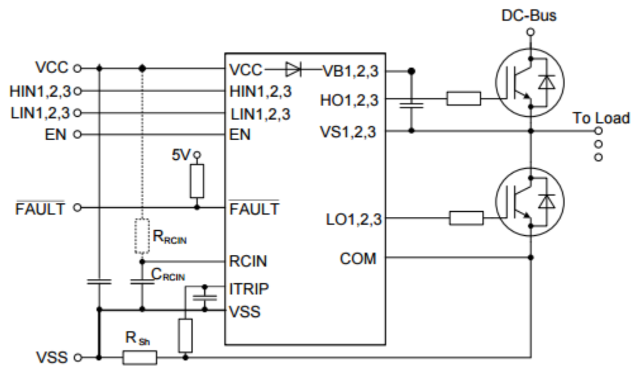

Fig.5 The driver usage

The driver's intervention at currents over a set value consists in interrupting the command to the MOSFET transistors. Thus, the motor current will be reduced by avoiding its dangerous values.

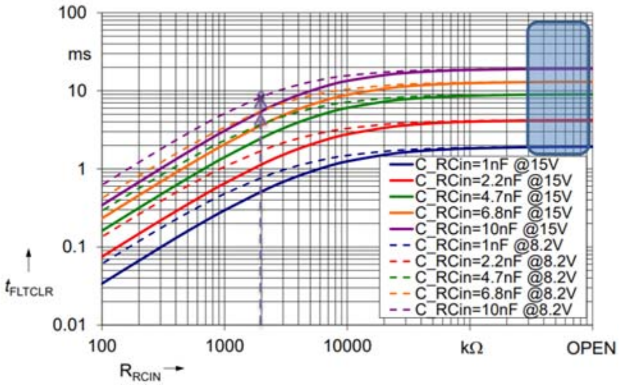

Fig.6 Fault clear time of C_{RCIN} in combination with R_{RCIN}.

The second part of controller, it's represented by the processor that provide the following main functions:

- ADC acquisition of 4 channels: the prescribed speed, the value of break input, and the value of current in the motor's windings, the battery voltage;
- Evaluation of the positions of motor reading 3 hall sensors;
- It computes the speed and send that information to another module (the display module) as a PWM signal;
- Generation of PWM signal for inverters' transistors.

Before PCB design, for this controller it was made a testing board to evaluate the microcontroller and MOSFET's driver.

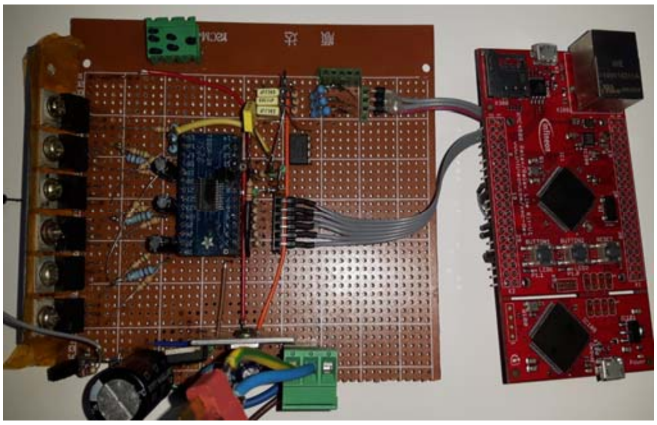

Fig.7 Testing board

In the next stage, it was released the PCB for the prototype shown in fig.8.

1-XMC1402F064X0200 microcontroller;

2-6EDL04N02PR driver;

3-motor's power wires;

4-Hall sensors wires;

5-12V source;

6-Invertor bridge MOSFET's.

The used processor is from XMC 32-bit family made by Infineon. The development team evaluated two processor subfamilies: XMC1000 and XMC4000, ultimately choosing the XMC1402F064X0200 processor. This is 32-bit Microcontrollers with ARM Cortex-M0 with focus on embedded control applications and has 64.0KB Flash, 16KB RAM, Core frequency: 48MHZ, Peripherals clock: 96 MHZ, RTC, WDT, Temperature range: -40 to 105°C [6].

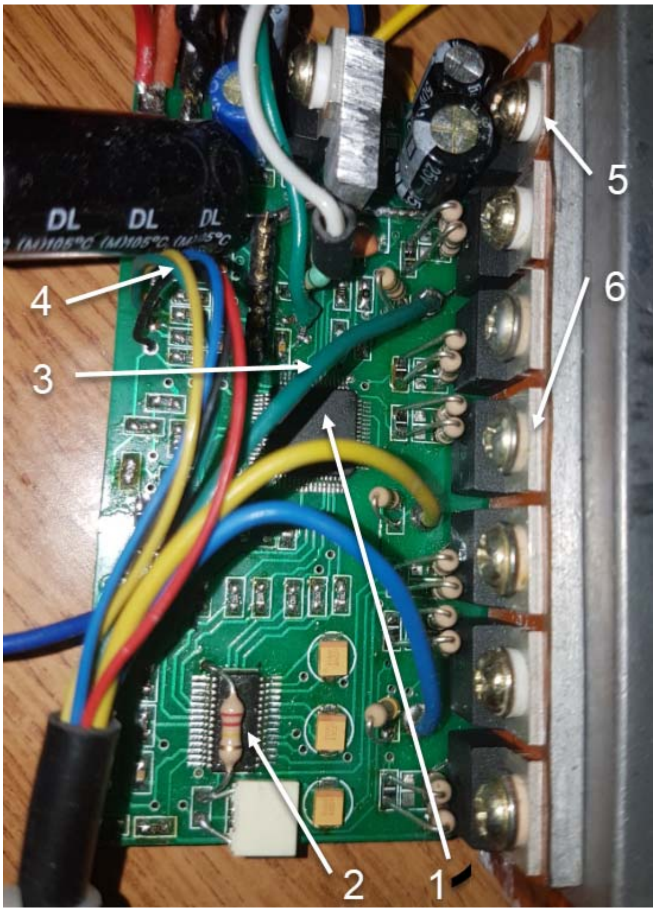

Fig.8 The controller's prototype

The usage of 6EDL04N02PR driver simplify a lot the design of controller and its performances can be viewed in fig.9. There are presented some samples of phase voltages and MOSFET gate's commands.

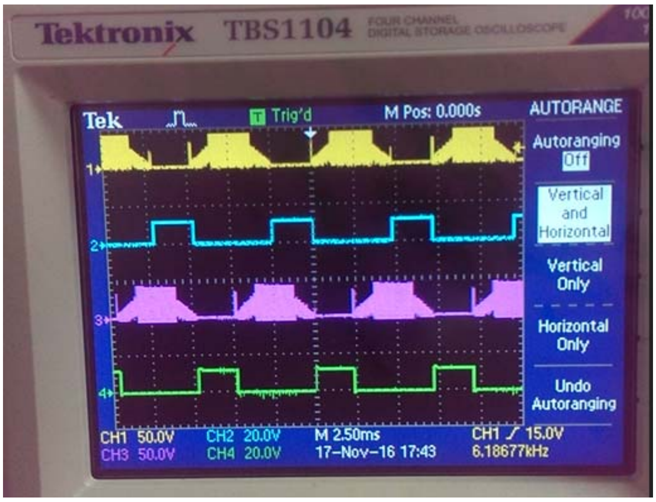

Fig.9 Sample of signals

Also, the 6EDL04N02PR driver has the possibility to insert a deadtime between the command for MOSFETS from same phase to avoid the short-circuits.

The PWM signals generated by microcontroller are presented in fig.10.

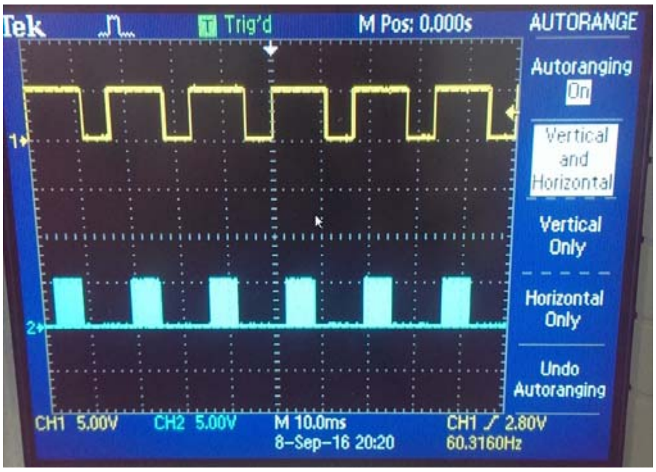

Fig.10 PWM signals generated by microcontroller

The frequency of PWM signal applied to MOSFETs gates was choose at 17.5kHz. This value was determined in experiments and it is the value at which the motor works best (it is more fluid at a large range of resistant torque).

In tests, the processor proved to be extremely stable, both at the acquisition of ADC and in generating command signals, the maximum CPU usage being below 20%.

III. The Software Package

For the development of the software package was used the integrated development environment provided by Infineon for the XMC microprocessor family.

The software development was made using Dave 4 IDE (Digital Application Virtual Engineer) from Infineon. This is a C/C++ language software development and code generation tool for Infineon microcontroller applications.

Dave 4 is a standalone system with automatic code generation modules and is suited to develop software drivers for Infineon microcontrollers and aids the developer with automatically created C-level templates and user desired functionalities.

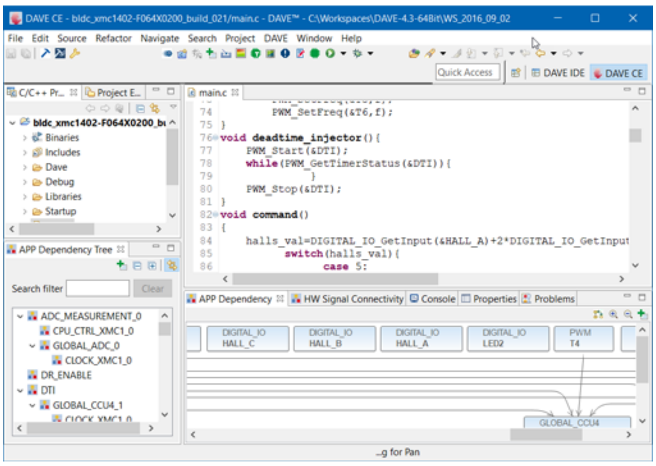

Fig.11 Dave 4 IDE Interface

This is a free-charge environment developed on the Eclipse platform and provides the toolchain needed to edit, compile, debug and burn the flash memory of the microprocessor. Also, Dave offers graphical interfaces to configure and generate some code segments, as well as multiple feature libraries optimized.

The basic function of the program is to generate the commands according to the operating protocol required for BLDC motor operation of a two-wheeled vehicle.

The afferent program flowchart is shown in fig.12.

At startup, the peripherals and hardware blocks used (PWM blocks, ADC conversion channels, input / output ports, timers, and interrupts) are initialized.

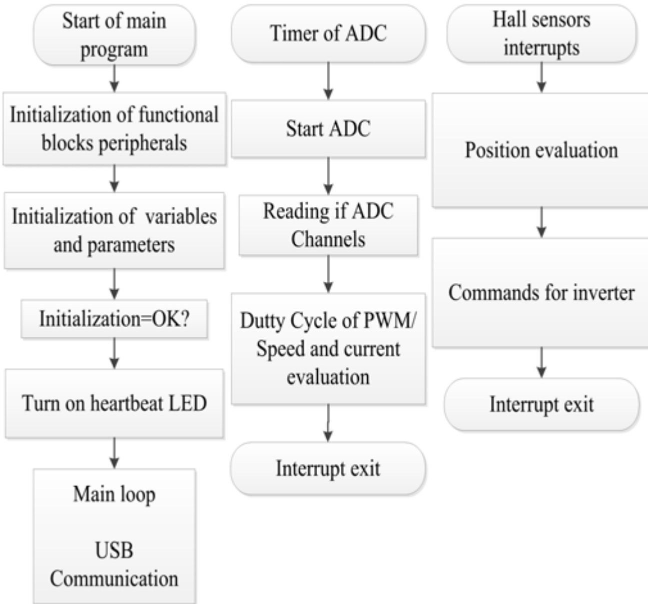

Fig.12 The program flowchart

Immediately after this, the variables are initialized. If everything went normal, the program goes into a continuous loop where it performs various tasks.

At present, these tasks are reduced to two: to signal the program running by lighting a led, and to assure the USB communication with the PC application. For the moment, this second task sends the calculated speed to USB and receives a prescribed fill factor for engine mode from a Win32 application.

In order to ensure USB communication, the drivers generated by the Dave4 environment must be installed on the PC.

The main loop is interrupted by multiple interrupt sources:

- 6 interruptions generated by increasing and decreasing the hall sensors used to detect the engine position. During these interruptions, the increment of a numerator used for calculating the speed is also achieved. They are treated by a single interruption routine;

- An interrupt generated by a timer set to 50ms which will trigger the acquisition of two ADCs: the current value and the value of the throttle lever position. Also, it is evaluated the speed based on time between the interrupts generated by hall sensors. Upon completion, the prescribed duty cycle for PWM signals is calculated.

The interrupt handling routine generated by the hall sensors performs the switching of the semiconductor elements in the power driver according to the fig.13.

For the command of BLDC was used a trapezoidal commutation because this is more efficient compared with sinusoidal drive.

HALL				MOTOR OUTPUTS													
				REAR							FRONT						
A U	B V	C W		T1 U+	T2 U-	T3 V+	T4 V-	T5 W+	T6 W-		T1 U+	T2 U-	T3 V+	T4 V-	T5 W+	T6 W-
1	1	0	3	0	0	1	0	0	1	HC0	0	0	0	1	1	0
0	1	0	2	0	1	1	0	0	0	HC1	1	0	0	1	0	0
0	1	1	6	0	1	0	0	1	0	HC2	1	0	0	0	0	1
0	0	1	4	0	0	0	1	1	0	HC3	0	0	1	0	0	1
1	0	1	5	1	0	0	1	0	0	HC4	0	1	1	0	0	0
1	0	0	1	1	0	0	0	0	1	HC5	0	1	0	0	1	0

Fig.13 The switching sequence

To generate the maximum torque, the voltage pulses must be applied to two windings so that the angle between the stator and rotor flux is kept as close as 90 °.

Also, a very high priority routine monitors the input of the prescribed speed to cut-off the command when it fails (broken wire).

A graphical user interface tool was created as a Windows application that can be used to exchange messages with the controller using its USB host support.

In this way the communication between a PC and a controller was tested to develop a graphical user interface to change certain parameters of the controller. Also, it can be used to extract messages in the form of a log (various states, errors) by means of which makes a faster diagnosis and an identification of malfunctions. A log with 100 entries of errors are stored in flash memory of MCU.

The application allows passing the controller into the test mode (by activating the Test Mode control) sending a value for the PWM signal for the upper transistors in the engine mode but also for the lower one in the recuperative braking mode.

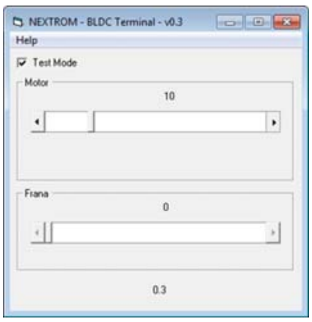

Fig.14 Win32 application

IV. TESTS

All tests were made on a dedicated testing stand. The controller has been tested both in traction mode and in the kinetic energy recovery mode.

First test of this controller was made without load. In that case, the controller doesn't have to control the current because its value is very low as the fig.15.

In the fig.16 it is presented the current variation with a load represented by a person with the weight of 65 kg simulated on stand.

There can be observed the intervention of the control loop in the current limitation somewhere around of 19A.

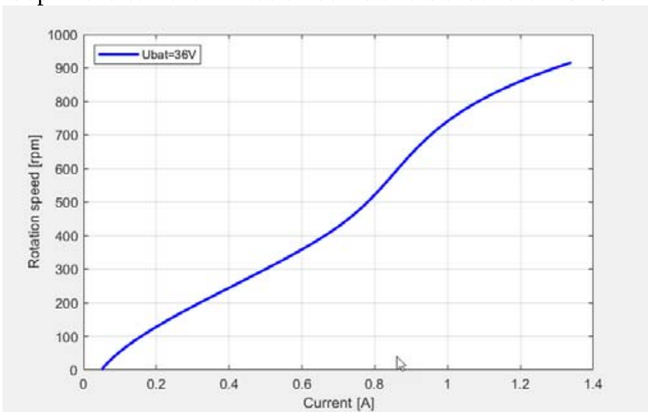

Fig.15 Current in motor without load

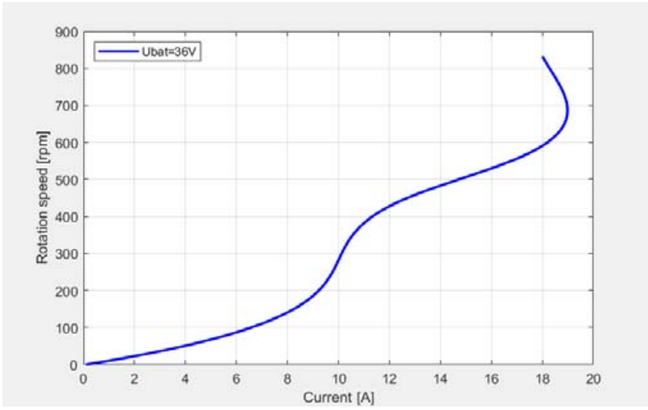

Fig.16 Current in motor with constant load represented by a person with the weight of 65 kg

The next test was done keeping the prescribed speed constant (at maximum value) and variable load. Also, here can be observed the intervention of current's control loop.

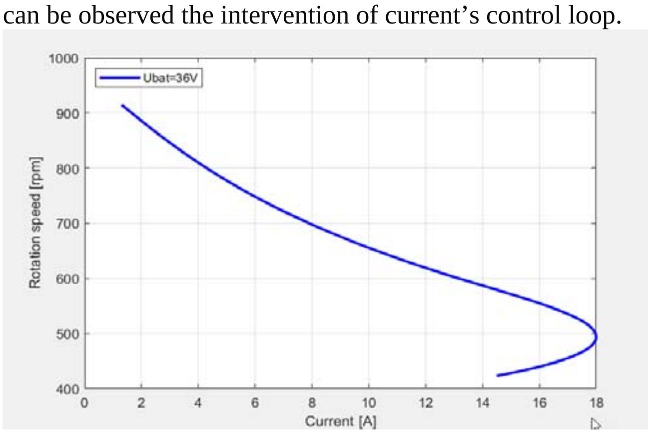

Fig.17 The current's evolution at maximum prescribed speed and variable load

The current can be also controlled by software algorithm. The current is read by a hardware loop and sampled by an analogic to digital channel.

The design of brake system was considered only the electric variant. The architecture of controller makes possible to recover the kinetic energy as electric power, charging the battery.

The brushless DC (BLDC) motor is an inside-out permanent magnet DC motor, in which the conventional multi-segment commutator, which acts as a mechanical rectifier, is replaced with an electronic circuit to do the commutation [6].

Regenerative braking can be achieved by the reversal of current in the motor-battery circuit during deceleration, taking advantage of the motor acting as a generator, redirecting the current flow into the supply battery. To create that regenerative brake, all top MOSFETs are off, and the bottom ones will each turn 120 degrees. PWM technique is used to control the level of regenerative braking by varying the duty cycle of the PWM. During regenerative braking, current in the winding can flow back through the freewheeling diodes and it is reversed and supplied into the battery.

When the battery is full, the battery management system will close the charging circuit. The controller will change the regenerative braking into dynamic braking. This braking can be implemented by disconnecting the power supply to windings and short circuiting them.

Another braking solution can be obtained by reversing the applied supply voltage, so that the input voltage assists the back emf in forcing armature current in reverse direction.

V. Conclusions

That paper, was presented a comprehensive design of a BLDC drive system using XMC Family processor and dedicated driver for MOSFET transistors. The hardware architecture is simpler and more reliable than others. The software tools made available by Infineon offers the conditions to create a very good controller for BLDC motors.

Fig.18 Testing setup

The results have been obtained for various load conditions. In all situations the designed system has behaved very well with great dynamic performance.

In future, it is desired the development of a controller based on this architecture by implementing software algorithms that lead to outstanding dynamic performance and an increased battery autonomy.

The controller was placed and tested on a two-wheeled vehicle platform. Following the tests, the engine-controller assembly behaved well, ensuring a maximum speed of over 40km/h and a very good torque at all speed range. In the near future, after the testing stage, the development team will bring this controller to the commercial stage and will equip the next generation of electric scooters.

Acknowledgment

The paper is the result of collaborative work between the research team from Faculty of Electrical Engineering at the University of Craiova and the research engineers from the Research & Development Department of E-Twow Electric Mobility S.A. who offered all support with equipment at them headquarter where, also, the tests were done.

Source of research funding in this article: Research & Development Department of E-Twow Electric Mobility S.A.

Contribution of authors:

Firs author – 50%

First coauthor – 20%

Second coauthor – 20%

Third coauthor – 10%

Received on Juin 17, 2018

Editorial Approval on December 5, 2018

References

[1] Abdul Rahim Abdullah, Mohd Arif Abu Hassan - Efficiency Comparison of Trapezoidal and Sinusoidal Method for Brushless DC Motor Drive. Applied Mechanics and Materials 785(2015):248-252 · April 2015

[2] Akhila M., Ratnan P. - Brushless DC Motor Drive with Regenerative Braking using Adaptive Neuro based Fuzzy Inference System. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016

[3] Emadi, A. - "Handbook of Automotive Power Electronics and Motor Drives", CRC Taylor & Francis. 2005

[4] Florin Ravigan, Niculae Boteanu, Marius Dumitriu, Ionut Zglimbea - BLDC Controller Based on Dedicated System-In-Package. International Journal of Circuits and Electronics, 2017

[5] Jarrad Cody, Özdemir Göl, Zorica Nedic, Andrew Nafalski, Aaron Mohtar - Regenerative Braking In An Electric Vehicle.

[6] M.Rakesh, P.V.R.L. Narasimham - Different Braking Techniques Employed to a Brushless DC Motor Drive used in Locomotives. International Electrical Engineering Journal (IEEJ) Vol. 3 (2012) No. 2, pp. 784-790 ISSN 2078-2365.

[7] Omar Mohammed - A study of control systems for brushless DC motors. The University of Toledo 2014

[8] Sergiu IVANOV, Virginia IVANOV, Daniel CISMARU, Florin RAVIGAN, Dan SELISTEANU, Dorin SENDRESCU, Comparison of several control strategies of the BLDC motors, 29th European Conference on Modelling and Simulation, 26-29.05.2015, Albena, Bulgaria, ISBN 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD), pp. 215-220.

[9] Xiaohong Nian, Fei Peng, and Hang Zhang, "Regenerative Braking System of Electric Vehicle Driven by Brushless DC Motor," Ieee Transactions On Industrial Electronics, Vol. 61, No. 10, October 2014

[10] *** - www.infineon.com.

[11] *** - en.wikipedia.org/wiki/Brushless_DC_electric_motor.