Use of Modern Software Systems for Design and Realization of Prototype of Three-dimensional Model

Naqib Daneshjo [1], Erika Dudáš Pajerská [1], Zuzana Hajduova [1],
Cristian Dan Stratyinski [2], Mirwais Danishjoo [3]

[1]*University of Economics in Bratislava Faculty of Business Economics with seat in Kosice, Kosice, Slovak Republic*
[2]*Avast s.r.o, 040 11 Kosice, Slovak Republic*
[3]*Enterprice, Fredericksburg Rd Apt San Antonio, San Antonio, Texas, United States of America (USA)*

Abstract **– The work includes an overview of modern software systems suitable for prototype design of complex technical field products as well as serial production support. The introductory part of the thesis includes a research of computer-aided systems for engineering designs, control system designs, electronics as well as the chip itself with a control algorithm. Greater attention is paid to the more detailed analysis of the electronically switched engine, its mechanical design, its electronic engagement as well as its control. Brushless DC Electric Motor (BLDC) motors are most commonly used in small unmanned aircrafts, helicopters and n-clovers. In the practical part of the thesis is described the own design of equipment for testing and calibration of BLDC engines. Own methods of controlling and scanning BLDCs are described through Field-programmable Gate Arrays (FPGAs) and Multipoint Control Units (MCUs). The proposed system offers parallel control and testing of two BLDC engines.**

Keywords **– three-dimensional model, FPGA, BLDC, electric motor control.**

DOI: 10.18421/TEM82-05
https://dx.doi.org/10.18421/TEM82-05

Corresponding author: Naqib Daneshjo,
University of Economics in Bratislava Faculty of Business Economics with seat in Kosice, Kosice, Slovak Republic
Email: daneshjo47@gmail.com

Received: 14 January 2019.
Accepted: 09 March 2019.
Published: 27 May 2019.

1. Introduction

Today's technological growth is mainly due to the availability of professional software systems. These software systems are financially incomparable with the laboratory equipment that was necessary for the development, production and testing of new technologies. The aim of this work is to point out the possibilities of using the appropriate software needed for the design and preparation of the new design. The introductory chapter briefly describes the Computer-aided Designed (CAD) tools used in this work. Consequently, the separate issue of BLDC engine management is being approached, increasingly used in drone as well as in aerospace and aerospace as an actuator. The benefits of these engines are high speed, long service life and precise steering.

The practical part of the paper focuses mainly on the management of the above mentioned motors and the sensing of their operating parameters. A laboratory device for testing and debugging BLDC control was proposed. The work contains electronic circuit diagrams in which test methods are applied from personal practice. These schemes were then reprinted into printed circuits. The necessary documentation for production is part of this work. Also, the chapter is dedicated to the FPGA, where the internal architecture blocks and communication with the MCU are designed and delineated.

Separate laboratory equipment to test BLDC engine control methods can only serve as a BLDC engine test unit, so it will sense the stroke, speed, near-motor, current, and voltage. At lower speeds, it is also possible to read the rotor rotation angle. The device is designed for two BLDC motors that can be tested in parallel. One motor is controlled by an H-bridge directly from the FPGA and the other is controlled via the Electronic Speed Control (ESC) module based on the Pulse Width Modulation (PWM) signal generated by the same FPGA. The work presents a possible design of computer

applications for recording the measured values, their analysis and subsequent implementation into the control.

2. Architecture of drive control of bldc motor based on inverter

The main control unit is built on the Altera development board. The block architecture of the FPGA is shown in Figure 1. The interface of the control unit with the engine consists of the inverter bridge and the three effects of Hall sensor sensors. The design below shows the block architecture of a complex control system. The logic blocks are implemented in the VHSIC Hardware Description Language (VHDL) program using Altera hardware. The whole design is synchronous and runs at 20MHz hourly signal [4],[5].

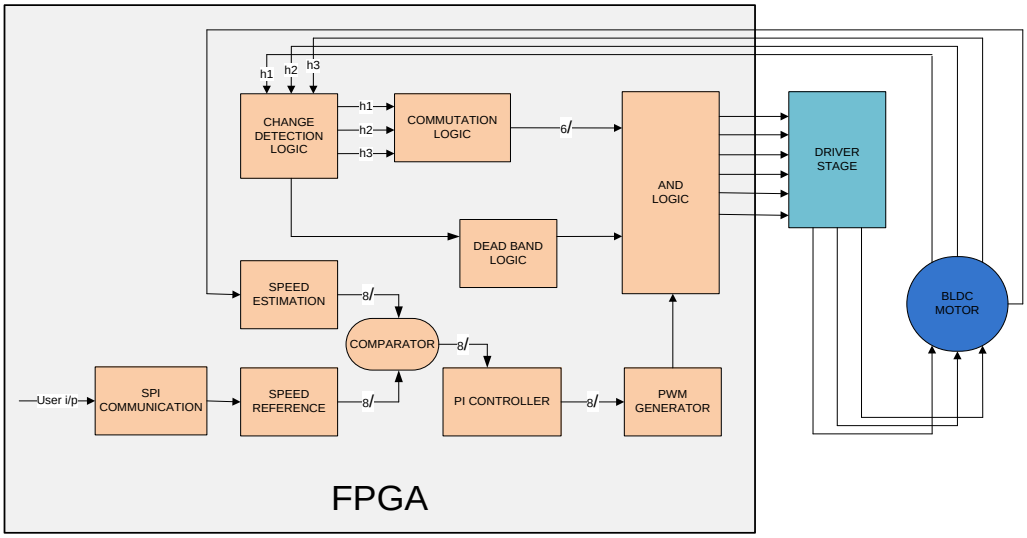

Figure 1. Architecture of drive control of a BLDC motor

Change detection logic – The current states of the BLDC Hall sensors are first stored in three D-type flip-flops that tilt the leading edge of the clock signal. Other flip-flops are used to write the previous values of the inputs obtained from Hall sensors, which are the following states of the Hall sensors. These states of the Hall sensors are then led to a commutation logic module [1],[2]. The new inputs are then compared to the previous inputs and therefore the input changes are defined as logical "1". When there is no change between the new and the previous inputs from the Hall sensors, the logic "0" is set. This one cycle pulse is used for the Dead-Band logic module for the start of the dead band computer.

Commutation logic – For correct rotation of the motor, the switching logic is used for sequential feeding of individual coils which is predefined according to the selected direction of rotation. Changing the sequence of the switching sequence means changing the direction of the rotation of the motor. By defining the switching sequence, the path of the current in the coils is determined and the magnetic field is generated on a single coil. By rotating the coil feed in one direction or the other, the permanent magnet rotor moves. Based on the commutation table – Table 1 is a proposed commutation logic shown in Figure 2. This commutation logic is controlled by Hall sensors (A, B, C).

Table 1. Commutation table based on Hall sensors

Hall sensor A	Hall sensor B	Hall sensor C	Phase A	Phase B	Phase C
1	0	0	$-V_{DCB}$	$+V_{DCB}$	NC
1	0	1	NC	$+V_{DCB}$	$-V_{DCB}$
0	0	1	$+V_{DCB}$	NC	$-V_{DCB}$
0	1	1	$+V_{DCB}$	$-V_{DCB}$	NC
0	1	0	NC	$-V_{DCB}$	$+V_{DCB}$
1	1	0	$-V_{DCB}$	NC	$+V_{DCB}$

The commutation logic controls six signals representing the status of one of the 6 transistors of the control unit. Decisive are inputs from Hall sensors that enter the rotor position [3],[7].

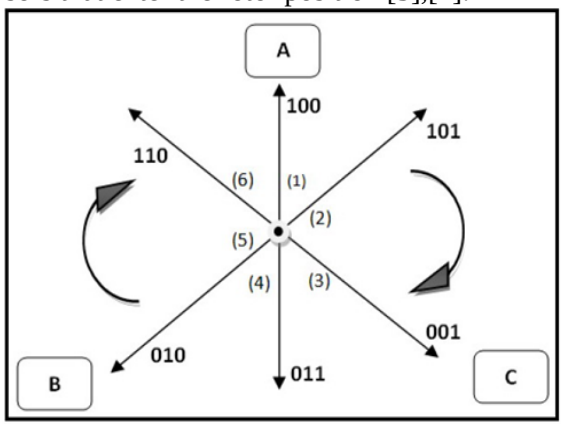

Figure 2. Commutation in the clockwise direction

2.1 Dead Band Generation Logic

As it was already mentioned, switching from one voltage state to another should be separated by a "dead band" appropriately to cover the transistor switch both on switching and off-time. This time interval must be set according to the type of transistors used, usually at least 0.5 μs. The dead band generator includes an 8-bit counter, a comparator and a D-flip-flop with synchronous reset. As soon as the input from Hall sensors changes, the counter and the D flip-flop are reset. If the logic module generated generates a logical "1", this is the input for the dead band generation module [4],[8]. Based on this event, the counter starts counting. The output of the counter is sold to the comparator which then sets the flip-flop. Thus, the output of the flapper is logical "0" for a short time interval equal to the time required to change the states in the logic change module and the time of the counter reading.

Module of PWM Generator and AND logic – The Pulse Width Modulation (PWM) generator module contains an 8-bit computer and zero detector, D flip-flop, and SR flip-flop. The counter is controlled by a 10MHz clock cycle. The PWM signal is generated and controlled by the Cyclone-4 processor described in VHDL. Switching signal parameters is frequency switching and power ratio. Each output from the commutation logic module, the dead band generation module and the PWM module are sold to the AND logic module for the power calculation of the six transistor control units, also known as the transducer [4].

2.2 PI controller

Speed control is addressed by the Proportional Integral (PI) controller. The wrong difference between the current speed and the reference speed is calculated for each PWM cycle and sold as the input to the PI controller.

The output cycle from the PI controller is displayed in the time domain:

$$Dutycycle = KP * error + KI * \int error\ d_t \quad (1)$$

Where:
KP- Proportional Gain
KI- Integral Gain
Error-The difference between the reference speed and the current speed
Dutycycle- Working cycle controller

In the discrete time domain, the same PI controller is described by the following equations:

$$y_n(k+1) = y_n(k) + KI * e(k) \quad (2)$$

$$Y_n(k+1) = y_n(k+1) + KP * e(k) \quad (3)$$

Where:
KP- Proportional Gain
KI- Integral Gain
e(k)- The difference between the reference speed and the current speed
$Y_n(k+1)$- Current cycle calculation
$y_n(k+1)$- Current error time
$y_n(k)$- Previous error time

2.3 SPI communication

Between the MCU of the BLDC motors and the FPGA BLDC engine, a Serial Peripheral Interface (SPI) communication is designed, where the FPGA is the SPI Slave part. SPI is a four-wire synchronous serial conductor that connects peripherals – in our case, MCU and FPGA.

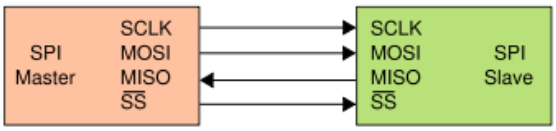

Figure 4. SPI interface

SCLK - Serial Clock; MOSI/SIMO - Master Output, Slave Input; MISO/SOMI - Master Input, Slave Output; SS - Slave Select (active low - the logic unit represents a lower

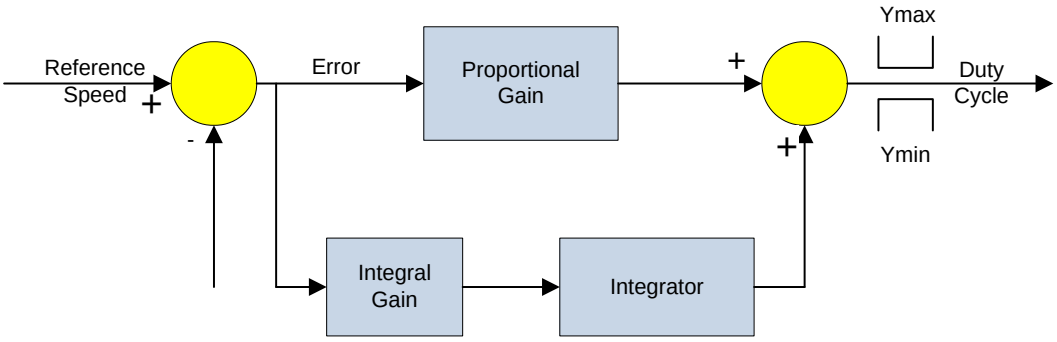

Figure 3. PI controller

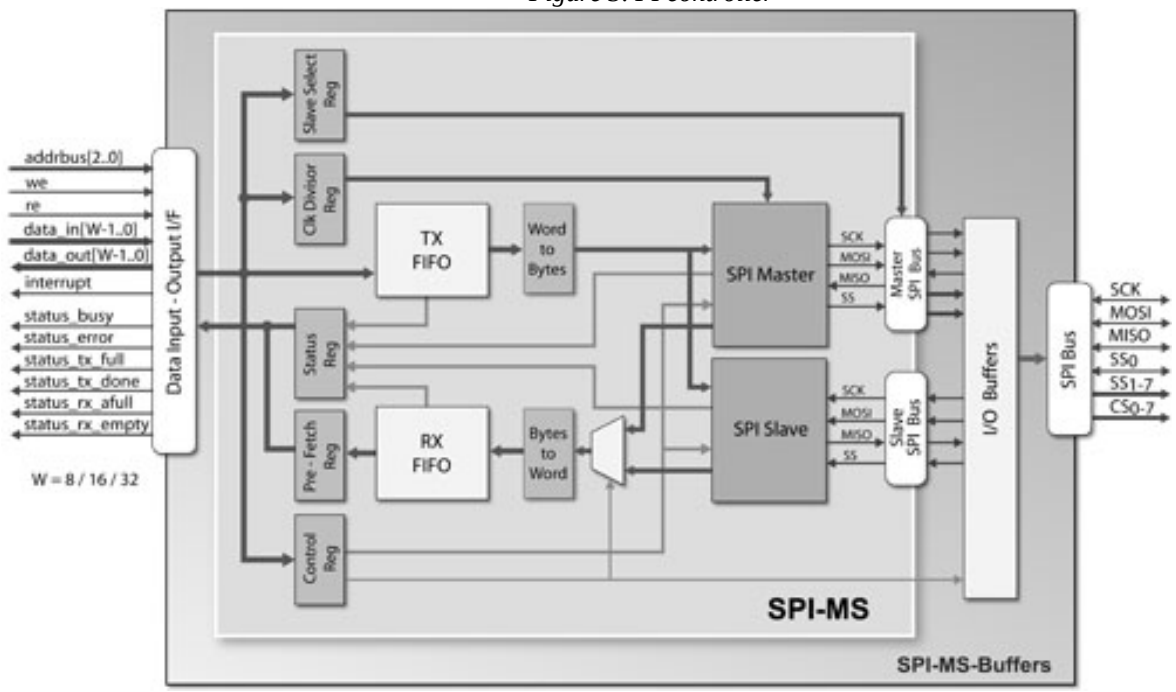

Figure 5. Architecture of FPGA for SPI – Master and Slave Interface

3. Control architecture for single ESC-BASED BLDC engine with PWM OR I2C interface

When controlling BLDC engines based on Entity Component System (ECS) with PWM or I2C, FPGA architecture is simplified by removing the Hall sensor sensing blocks that gave us the exact position of the rotor to the stator. As a consequence of removing the module for tracking the current position, we will not need even a modulation of the achieved logic, the output of which was the order of switching of individual coils. Since each individual coil switching in the stator is controlled by an ECS member, the dead band generation module as well as the logic module for switching power transistors is also omitted.

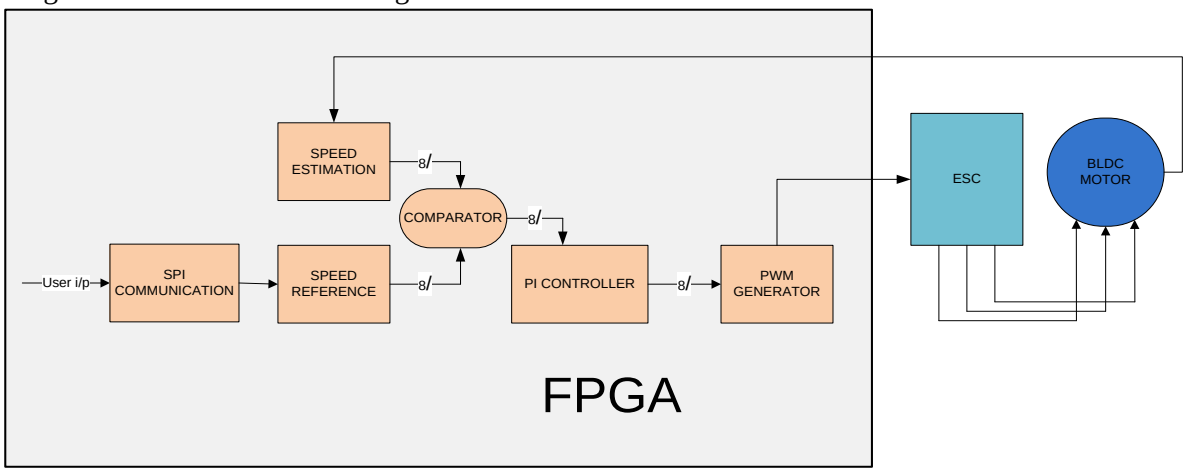

Figure 6. FPGA architecture for BLDC control of motor based on PWM

3.1 Pulse-width modulation

Pulse Width Modulation (PWM) is a method of encoding or modulation of a signal where the pulse width is a function of the amplitude of the input signal. The average value of the output modulated signal is the level of the input signal.

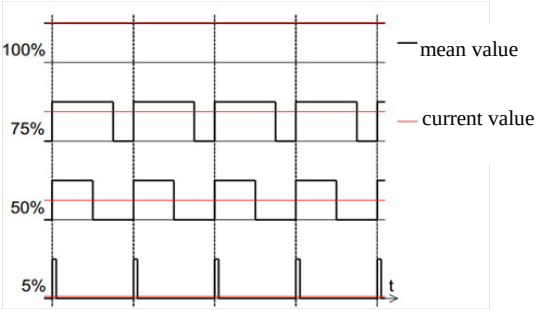

Figure 7. The pulse width modulated signal process

Figure 7 shows a pulse-width modulated constant signal with a different mean value. The mean value is given by the ratio of the modulation period and the class (active duty cycle) of the signal. The modulated output signal is usually bi-value and acquires values log. 0 and log. 1 respectively acquire the voltage levels associated with these values. Pulse width modulation can also be done bipolar. In this case, the three-state and third-state output signals are generally assigned a negative voltage level [6],[9].

Figure 8 illustrates a method of generating a pulse-width modulated signal. The input signal that is modulated is compared with the reference saw signal. If the input signal level is higher than the reference signal level, the output is in the log. 1, otherwise the output is in log. 0. The frequency of the output signal equals the frequency of the reference firing signal.

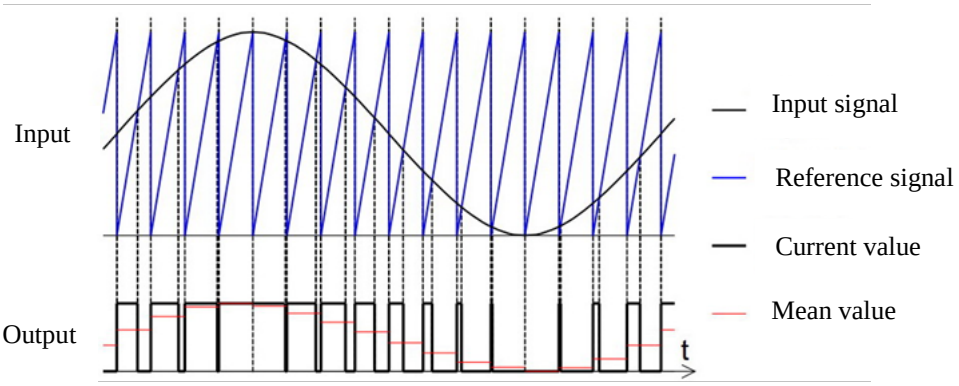

Figure 8. The principle of pulse-width modulated signal generation

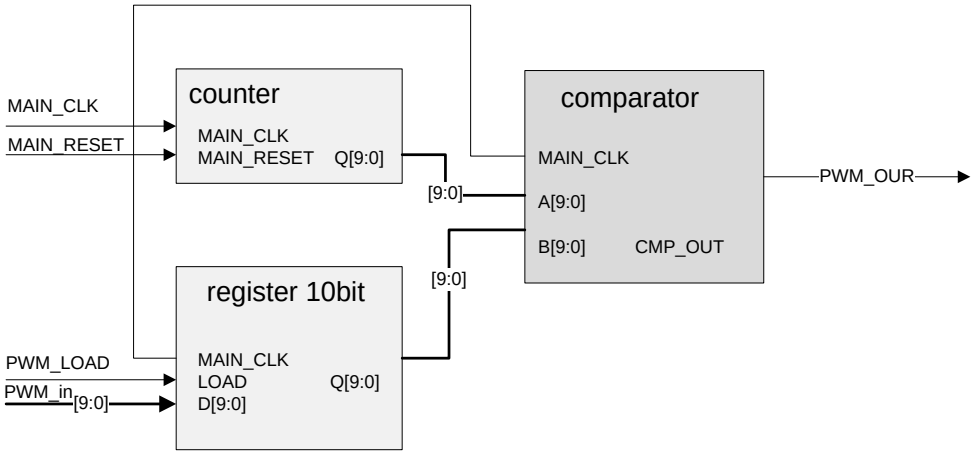

Figure 9. FPGA generation of PWM with 10bit register

```
--pwm_gen.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity pwm_gen is
Port (PWM_OUT:out;
      MAIN_CLK: in STD_LOGIC;
      MAIN_RESET: in STD_LOGIC;
      PWM_in:    in std_logic_vector(9 down to 0);
PWM_load:   in std_logic;);
      end pwm_gen;

architecture rtl of pwm_gen is
    signal counter: natural range 0 to 48000;
    signal result: STD_LOGIC_VECTOR (9 down to 0);
    signal pwmout: STD_LOGIC;
    signal clk_div: std_logic;

begin
PWM_in <= result;
PWM_load <= '1';
MAIN_RESET <= not MAIN_RESET;
 PWM_out <= pwmout;

process(MAIN_CLK,MAIN_RESET,counter)
begin
if (MAIN_RESET='0') then
counter <= 0;
clk_div <= '0';
elsif (MAIN_CLK'event and MAIN_CLK='1')    then
```

```
if (counter = 47999) then
clk_div <= not clk_div;
counter <= 0;
else
counter <= counter + 1;

end if;
end if;
end process;

process(clk_div, result)
begin
if (clk_div'event and clk_div='1')    then
result <= result + 1;
end if;

end process;

PWM_out <= (others => pwmout);
end rtl;
```

4. Design aspect of test and calibration equipment

The 3D model of the test device shows a kit for testing two BLDC engines in parallel. Every single engine is tested and controlled independently. (BLOCK A, BLOCK B).

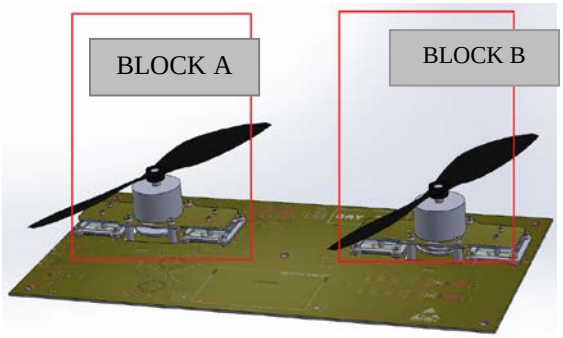

Figure 10: 3D model of the test device

The BLDC engine of block A, also known as the M1 engine, is controlled directly by the FPGA and the H-bridge of 6 transistors. BLDC block B engine, also called M2, is controlled by the RAY R-20B AC regulator (this can also be chosen differently). The PWM signal to control the AC regulator is generated by the FPGA [4],[10],[11].

The properties of BLDC motors for block A and block B are monitored by the following sensors:

- Stroke sensor.
- Speed sensor.
- Angle of rotation of the BLDC motor axis.
- Current sensor.
- Voltage sensor.
- Temperature sensing.

Testing the BLDC engine thrust is done by four strain gauges, each sensor block having two strain gauges. The figure below shows the grip and position of strain gauges.

4.1 Block diagram of a two-BLDC engine testing and control device based on the FPGA

The device includes two independent control blocks that are controlled in parallel by one FPGA device. Block A as well as block B has an independent microprocessor-coupled sensor which processes the readings, and, on request, the FPGAs expose them to the SPI communication conductor [4], [12].

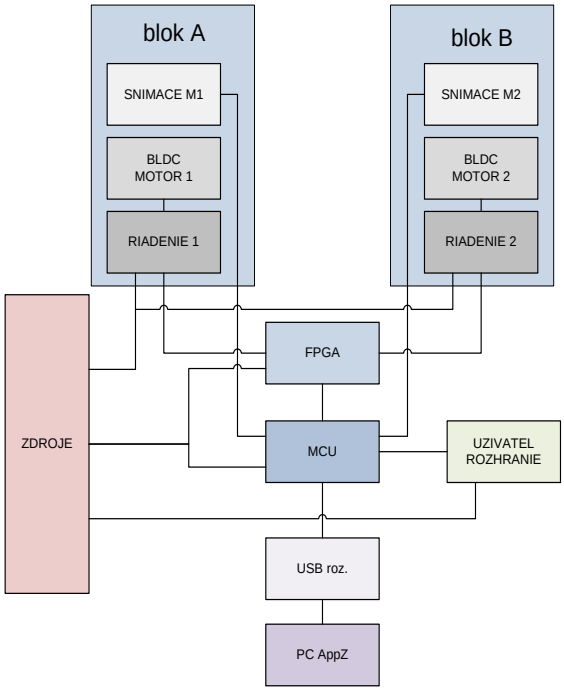

Figure 11. Testing device architecture

The printed circuit boards have been designed based on the electronic schemes of the above chapter. Their design was made by CAD software EAGLE 6.5.0 Professional. The "Zoidberg" test PCBs were designed as two-layer.

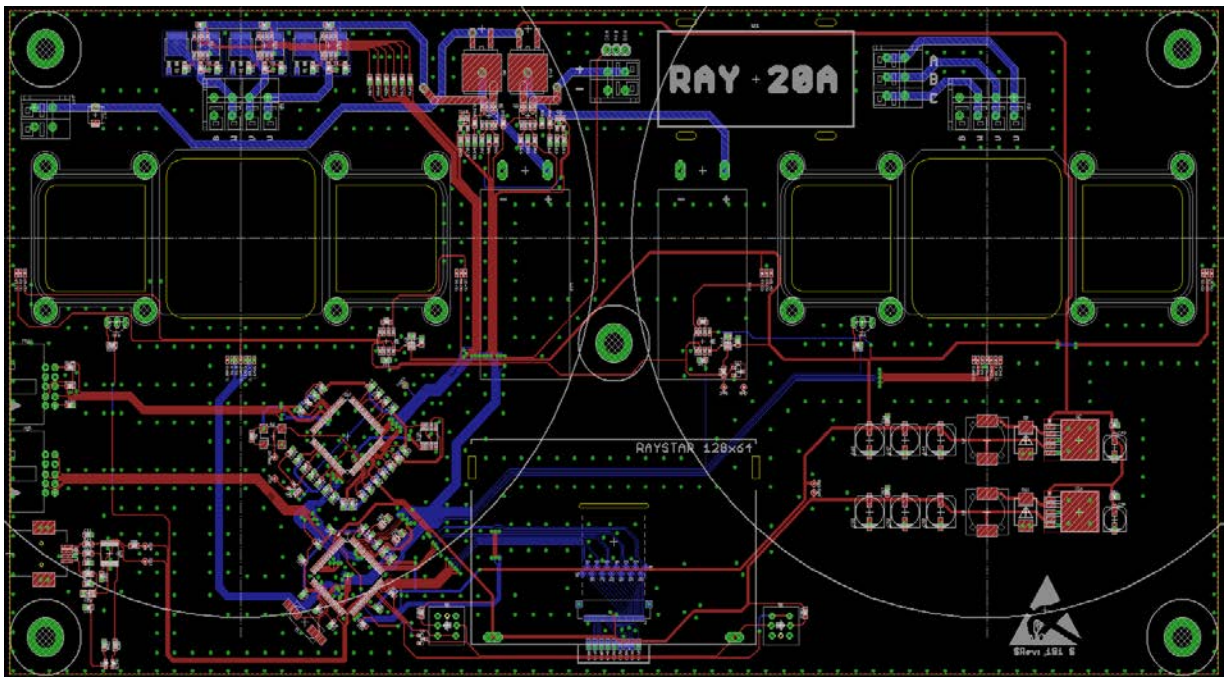

Figure 12. Preview of EAGLE source design PCB, main board

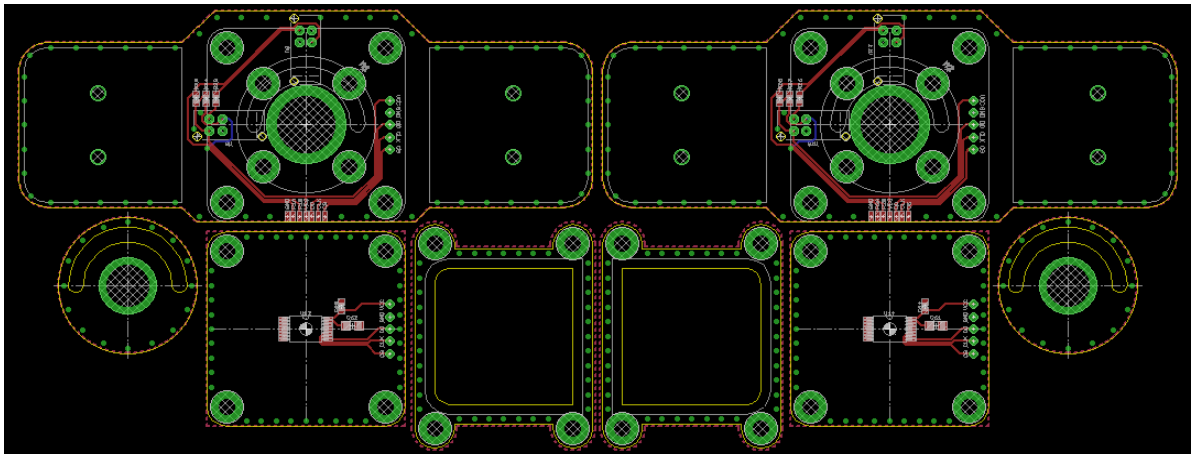

Figure 13. Preview of EAGLE source design PCB, sensor part

The implementation of computer technology and software in the product design process is supported by Computer Integrated Manufacturing. The current development of design and production of molded components in the consumer, engineering, automotive and aerospace industries is strongly supported by 3D CAD systems [4], [5], [13]. These systems make it possible to design 3D bulk and surface models that further serve as a basis for generating NC data (CAM) or analysis (CCA). The computer model becomes a necessity today in the process of designing, analyzing and producing components. Today's trend is to produce directly on the basis of generated NC data from a computer model. By creating logic descriptions, it is possible to create systems that can work independently, but are mostly used as an extension of the functionality of microcontrollers. Using programmable logic circuits, the performance of time-critical tasks can be optimized. Whether it's the calculations or the implementation of the required interface - which requires precise timing – and the processor may not be capable of. This may be due to interruptions or low computational power.

5. Conclusion

In the framework of the above mentioned paper is included a brief search of technologies and software suitable for the design of a complex system. We used these technologies in the practical part. One chapter was devoted to the management of BLDC engines, most commonly used in n-cavities. Practical part of the work was made by computer platforms CAD (mechanical and electronic) and by descriptive language VHDL (IP core FPGA). The work provides the necessary information and background for the actual

production of test and calibration equipment which can subsequently be used as a laboratory instrument for testing BLDC parameters. The proposed system is also suitable for verifying and debugging new BLDC engine management methods. The main advantage of the device is the sensing modules enabling the computer to be analyzed of the data being sampled. These modules can also be used independent of the control section, allowing testing of BLDC external drive units.

Programmable logic circuits are types of digital integrated circuits, the function of which is determined by the user through a program which represents the interconnection of individual blocks within the integrated circuit. For a better idea, it can be simply described as creation of combinational logic along with sequential logic.

Acknowledgement

This work has been supported by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic (VEGA 1/0251/17).

References

[1]. Baca, J. & Korecko, S. & Poruban, J. & Vaclavik, P. (2003). Didactic version of program system for synthesis and diagnostics of logic circuits. *Problemy programmirovanija*, *3*, 53-58.

[2]. Čonka, Z., Kolcun, M., & Morva, G. (2016). Utilizing of Phase Shift Transformer for increasing of Total Transfer Capacity. *Acta Polytechnica Hungarica*, *13*(5), 27-37. doi: 10.12700/APH.13.5.2016.5.2

[3]. Daneshjo N. (2012). Computers Modeling and Simulation, *Advanced Materials Research, (463-464)*,1102-1105. doi: 10.4028/www.scientific.net/AMR.463-464.1102.

[4]. Dan stratyinski, C. (2017). *The use of modern software systems in the design of propeller aircraft motors. dissertation work* (2017), 110 p.Technical University of Kosice.

[5]. Daneshjo, N., Korba, P., Vargová, R., & Tahzib, B. (2013). Application of 3D modeling and simulation using modular components. In *Applied Mechanics and Materials* (Vol. 389, pp. 957-962). Trans Tech Publications.

[6]. Korečko, Š., & Sobota, B. (2014). Petri nets to B-language transformation in software development. *Acta polytechnica hungarica*, *11*(6), 187-206.

[7] Mihályi, D., & Novitzká, V. (2013). What about linear logic in computer science. *Acta Polytechnica Hungarica*, *10*(4), 147-160. doi: 10.12700/APH.10.04.2013.4.9

[8]. Máslo, K., Kasembe, A., & Kolcun, M. (2016). Simplification and unification of IEEE standard models for excitation systems. *Electric Power Systems Research*, *140*, 132-138.

[9]. Puškár, M., Fabian, M., & Tomko, T. (2018). Application of multidimensional statistical model for evaluation of measured data obtained from testing of the HCCI engine prototype. *Diagnostyka*, *19*, 19-24.

[10]. Stanova, E., Fedorko, G., Fabian, M., & Kmet, S. (2011). Computer modelling of wire strands and ropes Part I: Theory and computer implementation. *Advances in Engineering Software*, *42*(6), 305-315.

[11]. Steingartner, W., Novitzká, V., Bačíková, M., & Korečko, Š. (2018). New approach to categorical semantics for procedural languages. *Computing and Informatics*, *36*(6), 1385-1414.

[12]. Perháč, J., Mihályi, D., & Novitzká, V. (2017). Modeling synchronization problems: from composed petri nets to provable linear sequents. *Acta Polytechnica Hungarica*, *14*(8). 165-182. doi: 10.12700/APH.14.8.2017.8.9.

[13]. Steingartner, W. & Polakova, A. & Praznák, P.; Novitzka, V. (2015). Linear logic in computer science. *Journal of Applied Mathematics and Computational Mechanics*, *14*(1), 91-100. doi: 10.17512/jamcm.2015.1.09.