DESIGN OF CONTROL APPLICATION USING PROCESSOR EXPERT BLOCKSET

P. Stružka[1], L. Waszniowski[2], R. Bartosinski[3], T. Bystersk[2]

[1]UNIS, spol. s r.o.,

[2]Department of Control Engineering, FEE, Czech Technical University in Prague

[3]Institute of Information Theory and Automation, Czech Academy of Sciences

Abstract

This paper describes design of a controller of the brushless DC motor using Processor Expert blockset in Mathworks Simulink. The controller algorithm is designed in Simulink and the Processor Expert blockset provides access to the microcontroller hardware through its Hardware Abstraction Layer. The behavior of the controller (including the peripherals represented by Processor Expert blocks) is verified by simulation. The Real-Time Workshop is used for the C source code generation from the controller model.

The objective of this paper is to show on this non-trivial case-study, how the model based design methodology can be successfully used for effective rapid development and quality production code generation.

1 Introduction

The integration of The Mathworks® Simulink® and Processor Expert™ results in Processor Expert blockset, which provides new rewarding functionality for users. The main idea of the integration is utilization of the Hardware Abstraction Layer (HAL) [2], provided by Processor Expert Embedded Beans. This is achieved through the Processor Expert blockset used inside Mathworks Simulink. Blocks represent basic Processor Expert components corresponding to a microcontroller peripherals e.g. analog to digital converter (ADC), pulse with modulation output (PWM) etc. Blocks provide a simulation features and thanks to the connection of blocks to Processor Expert, it is possible to set up the parameters needed for selected peripheral configuration. The parameters for simulation in Simulink and for peripheral driver generated by Processor Expert are synchronized automatically.

When the designed control system is verified in simulation, an effective C code can be generated for the controller by Real-Time Workshop Embedded Coder® (RTW). Processor Expert blockset allows use of the generated driver‘s code in the resulting code generated by RTW. Composition of the resulting code ensures Processor Expert Embedded Real-Time Target (PEERT), which in addition to code generated for an algorithm, provides also connection of its inputs and outputs with proper function of MCU’s peripheral. Since HAL provides an interface between an embedded application and the target embedded board, the resulting application becomes a device-independent one and the platform-specific dependencies are hidden for it. Even this paper describes the case-study of model design on specific target board, the application can be easily transferred to whatever MCU supported by Processor Expert.

2 The BLDC Motor controller model

The paper shows and explains the BLDC motor controller main principle and how the model and its sub-systems are implemented. The closed-loop model used for the simulation is depicted in Figure 1. It consists of the following sub-systems:

Controller – this sub-system represents the BLDC motor controller. The inputs of this sub-system are quadrature signals (index and phases) and the outputs are PWM driven power signals for motor.

Encoder – simulates the quadrature encoder output used for sensing the rotor position. It simulates two phases A and B that represents the rotor position and the Index pulse – zero position.

BLDC – simulates the behavior of the BLDC motor. The inputs of this sub-system are 3 phases from controller (analog power signal voltage) and expected load of the shaft. The outputs are current, voltage, rotor angle, and hall sensor signal. Only the shaft angle is used for control. The other outputs are used for analysis of the motor behavior only.

Load – simulates the torque load of the motor shaft.

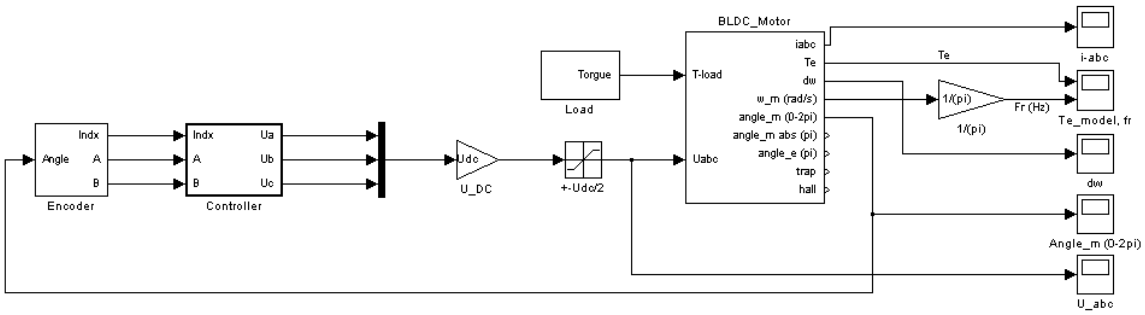

Figure 1 BLDC Motor controller model - simulation

The scopes are used for showing the simulated behavior. They show, apart of other things, currents and voltages of motor windings, the shaft angle and the rotation speed.

Notice that only the block Controller represents the designed application that will be later embedded into the microcontroller. The other blocks represent model of the controlled plant and they are necessary for closed loop simulation.

Following chapters describe the sub-models of the BLDC motor controller model shown on the picture above.

2.1 Controller sub-system

The controller sub-system, depicted in Figure 2, implements the designed control application that will be embedded in microcontroller. In addition to the controller algorithm, it contains Processor Expert (PE) blocks that allow access to the target microcontroller hardware. Also Processor Expert main block is included. It provides configuration of the hardware, required later for the RTW and PE code generation.

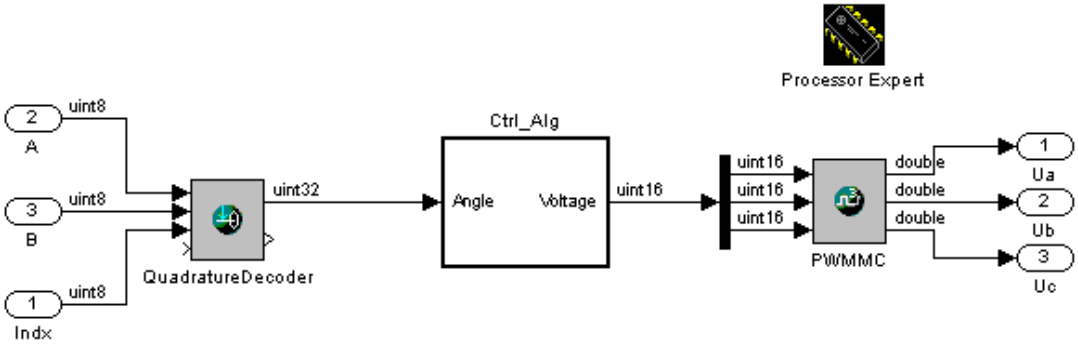

Figure 2 BLDC Motor controller sub-model

The input to the controller algorithm (block Ctrl_Alg) is the rotor angle. It is decoded from quadrature decoded signal (phases A and B and index) by block QuadratureDecoder. QuadratureDecoder represents the microcontroller hardware peripheral decoding electric signals. The electric signals are represented by inputs A, B and Indx for simulation purposes.

The output of the controller algorithm sub-block is required voltage for the motor. This value is input for the PWMMC block. The block PWMMC represents the microcontroller hardware peripheral generating the six-channel complementary pulse-width modulated signals controlling transistors switching the voltage applied on the motor windings. For simulation, the output is mean voltage on each phase.

The controller algorithm implemented in subsystem Ctrl_Alg is depicted in Figure 3.

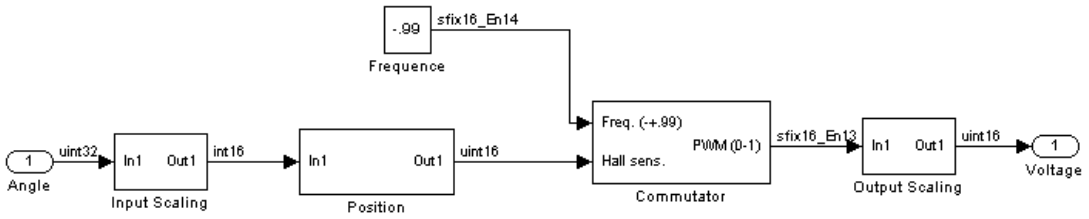

Figure 3 Controller algorithm sub-model

2.2 Encoder sub-system

The encoder sub-system, depicted in Figure 4, is used to simulate the rotor position encoder.

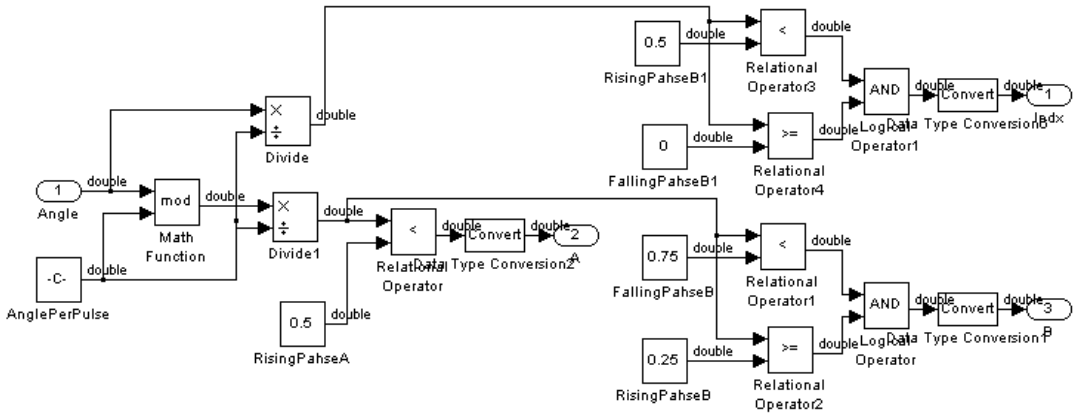

Figure 4 Encoder sub-model

The sub-model implements the algorithm of the phases A and B computation from the shaft angle and the Index pulse generation when in the zero position. The input of the sub-system is given angle of the rotor. The outputs are index, phase A and B.

2.3 BLDC motor sub-system

This sub-system represents mathematical model of the BLDC motor. Inputs of this sub-system are voltage on each of the phases and torque load.

Mathematical model of BLDC motor consist of electrical and mechanical part.

Electrical part of the BLDC motor is described by the following equations.

The stator phase voltages U_a, U_b, U_c cause currents i_a, i_b, i_c in the stator windings.

$$\frac{d}{dt}i_a = \frac{1}{3L_S}\left[2v_{ab} + v_{bc} - 3R_s i_a + \lambda p \omega_r \left(-2\Phi'_a + \Phi'_b + \Phi'_c\right)\right]$$

$$\frac{d}{dt}i_b = \frac{1}{3L_S}\left[-v_{ab} + v_{bc} - 3R_s i_b + \lambda p \omega_r \left(\Phi'_a - 2\Phi'_b + \Phi'_c\right)\right]$$

$$\frac{d}{dt}i_c = -\left(\frac{d}{dt}i_a + \frac{d}{dt}i_b\right)$$

where :

i_a, i_b, i_c	a, b and c phase currents
ω_r	angular velocity of the rotor
R_s	resistance of the stator winding
v_{ab}, v_{bc}	*ab* and *bc* phase to phase voltages
L_S	inductance of the stator windings
$\Phi'_a, \Phi'_b, \Phi'_c$	a, b and c phase electromotive forces (see Figure 5)
λ	amplitude of the flux induced by the permanent magnets of the rotor in the stator phases
p	number of pole pairs

The phase currents i_a, i_b, i_c produce an electromagnetic torque T_e

$$T_e = p\lambda(\Phi'_a \cdot i_a + \Phi'_b \cdot i_b + \Phi'_c \cdot i_c)$$

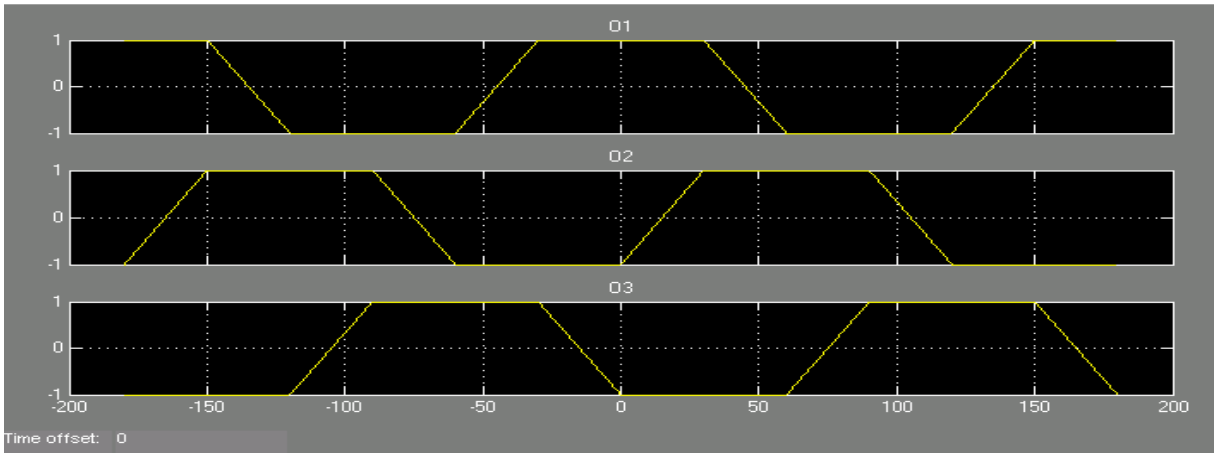

Figure 5 Distribution of the electromotive force $\Phi'_a, \Phi'_b, \Phi'_c$ by rotor angular position θ

Mechanical part of the BLDC motor is described by the following equation.

Electromagnetic torque T_e, shaft mechanical torque T_m and viscous friction of the rotor F determine the rotor angular velocity ω_r and rotor angular position θ.

$$\frac{d}{dt}\omega_r = \frac{1}{J}(T_e - F\omega_r - T_m)$$

$$\frac{d\theta}{dt} = \omega_r$$

where:

J	combined inertia of rotor and load
F	combined viscous friction of rotor and load
T_m	Load mechanical torque on shaft
θ	rotor angular position

The presented equations are implemented in the Simulink model depicted in Figure 6.

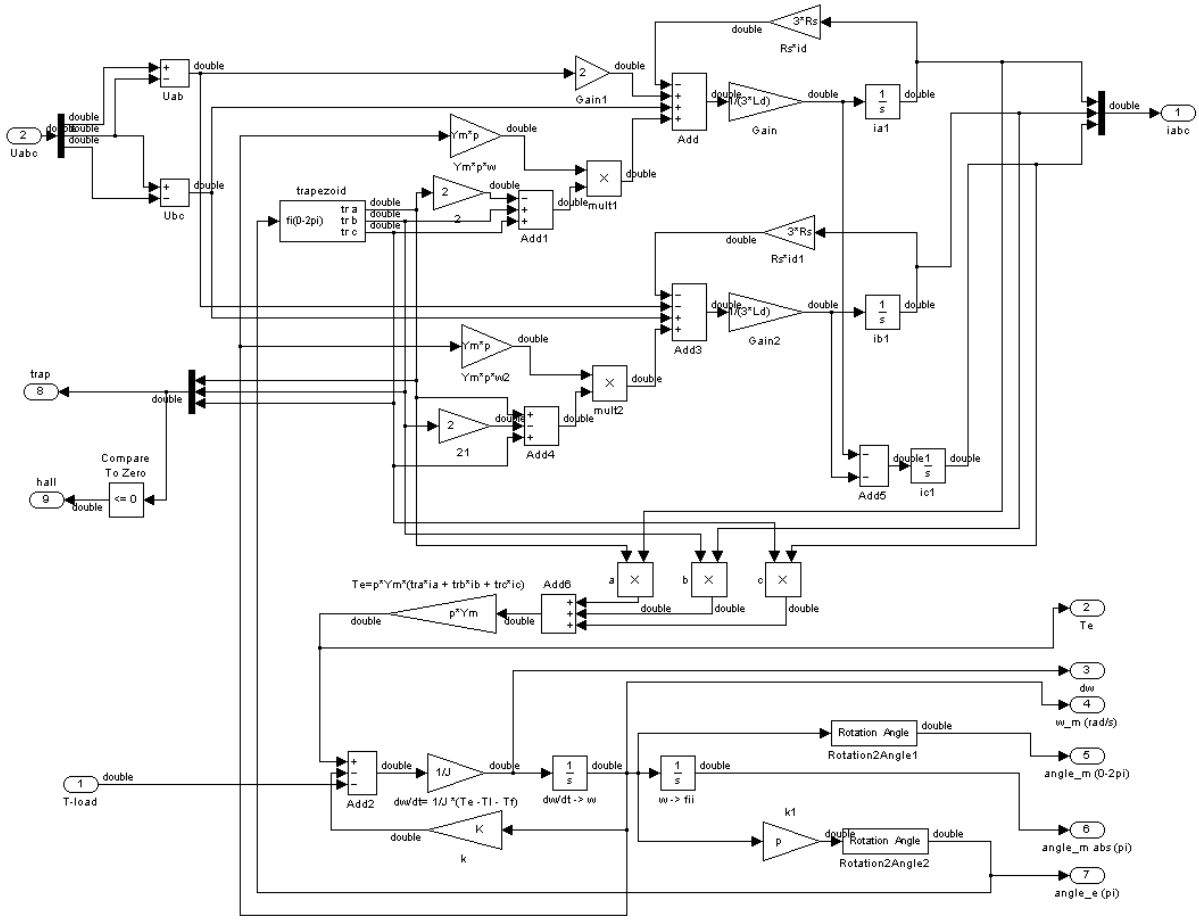

Figure 6 BLDC Motor sub-model

3 Simulation

The simulation is used for the designed controller algorithm verification. Since the simulation model contains also blocks representing the hardware peripherals (QuadratureDecoder and PWMMC) the simulation considers also hardware dependent aspect affecting precision of measurement and output signal generation. For example, the number of pulses of the encoder strongly affects the precision of the shaft angle and the rotation speed measurement.

During the simulation, the scopes show the most interesting signals (see the time diagrams in Figure 7, Figure 8 and Figure 9). The presented experiment shows reaction of the motor quantities on the impulse disturbance of the load torque. It starts at time 1 s and finishes at time 1.5s (see the purple line in Figure 7). The increased load causes decrease of the rotation speed of the shaft (see bottom diagram in Figure 7). As a consequence the currents increase (see Figure 8) since the lower angular velocity causes lower voltages induced by the permanent magnets of the rotor in the stator winding. Higher currents cause higher electromagnetic torque (see the yellow line in the top diagram in Figure 7), which together with the lower friction slightly compensates the decrease of the shaft angular velocity. As a consequence of the angular velocity change, the frequency of the phase voltage commutation is changed (see Figure 9), since it must be synchronous with the rotor rotation.

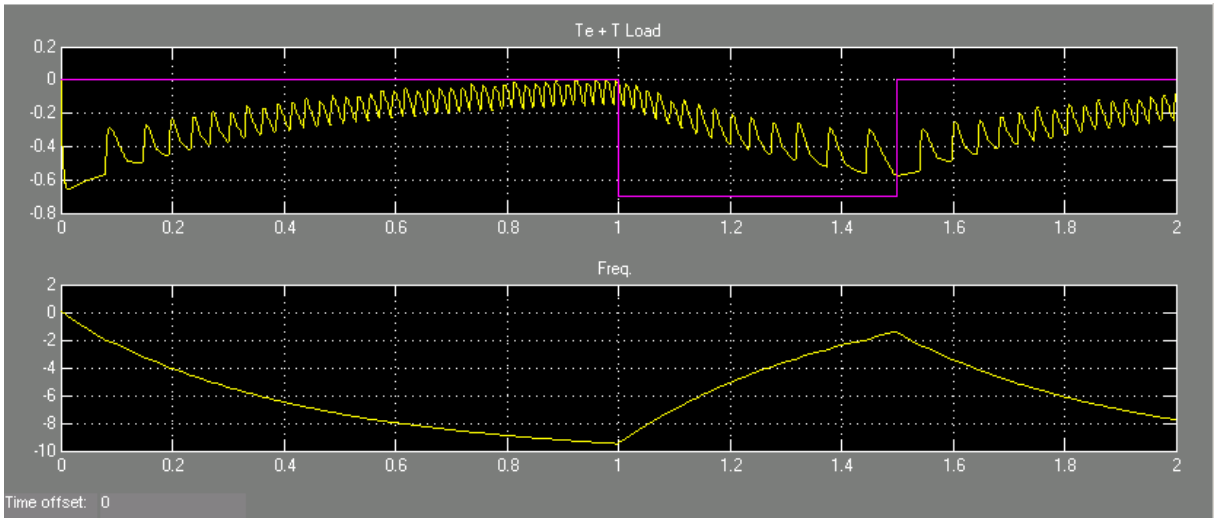

Figure 7 Time diagram of the load torque, the electromagnetic torque and the rotation speed of the shaft

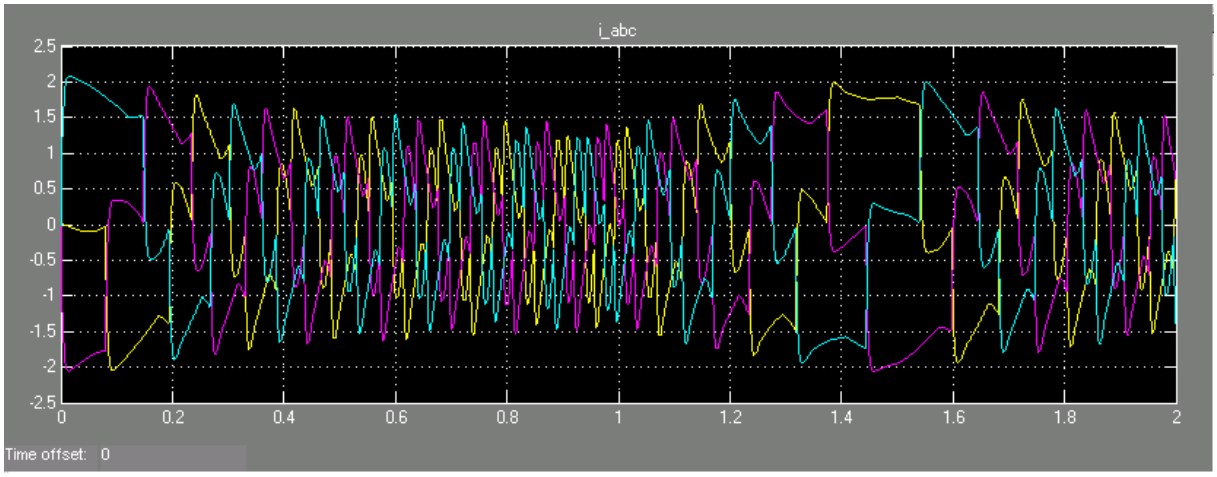

Figure 8 Time diagram of currents of motor winding

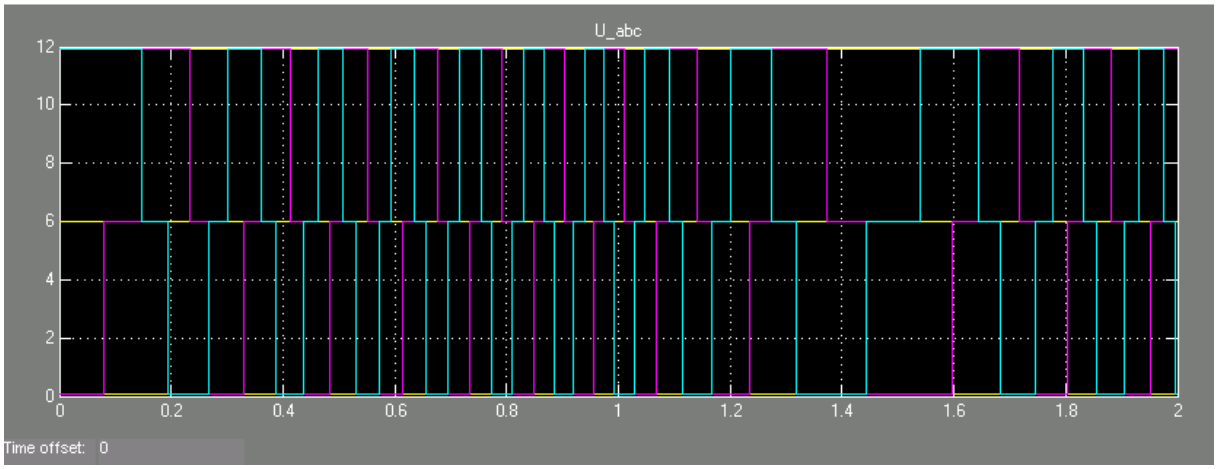

Figure 9 Time diagram of voltage applied on motor winding

4 Code Generation

For the code generation, the Controller sub-system is used only. It's possible to generate the source code of the controller algorithm using Real-Time Workshop. The Processor Expert Real-Time Embedded Target (PEERT) must be set and it is used for generating the source code and setting-up the Processor Expert project. The PEERT uses fixed step solver type with 1/5000 s sample time.

Processor Expert allows microcontroller hardware configuration with respect of the design setup. According to these settings, it generates the source code of the microcontroller initialization and all used peripheral drivers.

The resulting project then consists of the following parts (generated code):

- Processor Expert generated code
  - Code for the target device peripherals
  - CPU initialization
  - Beans' peripheral drivers
  - Make file and other needed files generated
- Real-time Workshop generated code
  - Code for model algorithm

All the generated code is included into the Processor Expert project. Processor Expert also allows invoking of the command line compiler tools.

5 Application testing

The application is tested on real hardware. The Freescale 56F8346EVM target board with DSC controller and EVM Motor Kit is used for driving the BLDC motor. The following picture shows the photo of the hardware configuration:

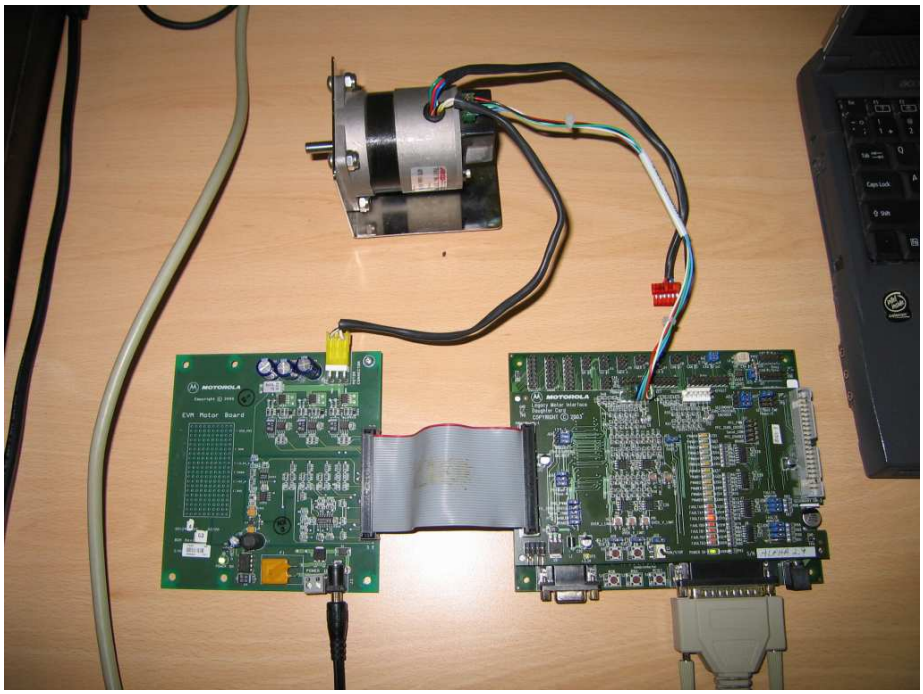

Figure 10 BLDC Motor controller configuration

The picture shows motor (on top), 56F8346EVM evaluation module (on right) and power daughter board (on left).

In this case study, the Freescale Digital signal controller has been selected as a target platform. CodeWarrior command line compiler used for the application building. Final binary code can downloaded the target board using CodeWarrior debugger.

6 Conclusion

Processor Expert™ hardware abstraction layers and the Embedded Beans™ have the pivotal value in rapid application development. Embedded Beans™ provide unique interface for the application creation, suitable for modeling.

The model simulation allows the application behavior verification on high level. This approach is usable especially for rapid application design and allows quickly design the controller application and test it on real hardware. In combination with Embedded Real-Time Workshop the time needed for application development can be shorten from months to weeks or even days.

This approach is designed to can find usable application not only in research environment but also in industrial applications, automotive industry, or general consumer market.

Acknowledgement

This project is supported by Academy of Sciences of the Czech Republic under project No. 1ET400750406.

References

[1] THE MATHWORKS: 2004. *Simulink – Simulation and Model-Based Design*. USA, The Mathworks, 2004.

[2] BARTOSINSKI, R., STRUŽKA, P., WASZNIOWSKI, L. *PEERT – Blockset for Processor Expert and MATLAB®/Simulink® integration.* Technical Computing Prague, Humusoft, 2005, ISBN 80-7080-57-3

[3] FREESCALE: *3-Phase BLDC Motor Control with Quadrature Encoder using 56F800/E*, Application note, AN1961, http://www.freescale.com/files/dsp/doc/app_note/AN1961.pdf, Freescale, Inc., 2005

[4] BARTOSINSKI, R., HANZÁLEK, Z., STRUŽKA, P., WASZNIOWSKI, L.: *Integrated Environment for Embedded Control Systems Design.* In IEEE International Parallel & Distributed Processing Symposium, WPDRTS07, March 26–30, USA, 2007, ISBN: 1-4244-0909-8

Petr Stružka
UNIS, spol. s r.o.
Jundrovská 33, 62400 Brno
e-mail: pstruzka@unis.cz

Libor Waszniowski
Department of Control Engineering, FEE, Czech Technical University in Prague
Karlovo nám. 13, 121 35, Praha 2
e-mail: xwasznio@fel.cvut.cz

Roman Bartosinski
Department of Signal Processing, UTIA, AS CR
Pod Vodaránskou věží 4, 182 08, Praha 8
e-mail: bartosr@utia.cas.cz

Tomáš Bysterský
Department of Control Engineering, FEE, Czech Technical University in Prague
Karlovo nám. 13, 121 35, Praha 2
e-mail: bystet1@fel.cvut.cz