УДК 004[896:94]

Hajduk Mikuláš, prof., Ing., PhD.
Sukop Marek, doc. Ing. PhD.
Varga Jozef, Ing. PhD.
Department of Robotics at Faculty of Mechanical Engineering, Technical University of Košice, Slovakia

CREATING A DC MOTOR CONTROL NECESSARY FOR TEACHING MOBILE ROBOTICS

The article is devoted to the description of the implementation of the basic PSD controller for DC motor. PSD controller is designed for the Arduino Mega ADK board. As a program in genvironment is used Flowcode 6, which is very suitable for teaching programmable modules for robotics. It is particularly suitable for students who still do not know any programming language.

Programming environment

Flowcode software allows you to quickly and easily develop complex electronic and electromechanical systems. The graphical programming tool allows even those with little experience to develop complex electronic systems in minutes. Flowcode is one of the world's most advanced environments for electronic and electromechanical system development. Engineers use Flowcode to develop systems for control and measurement based on microcontrollers, on rugged industrial interfaces or on Windows compatible personal computers. In our opinion Flowcode in conjunction with Arduino and its sensory and motion units is very suitable for the development of applications in mobile robotics. This connection Arduino with Flowcode is generally preferred for use in mechatronics.[1]

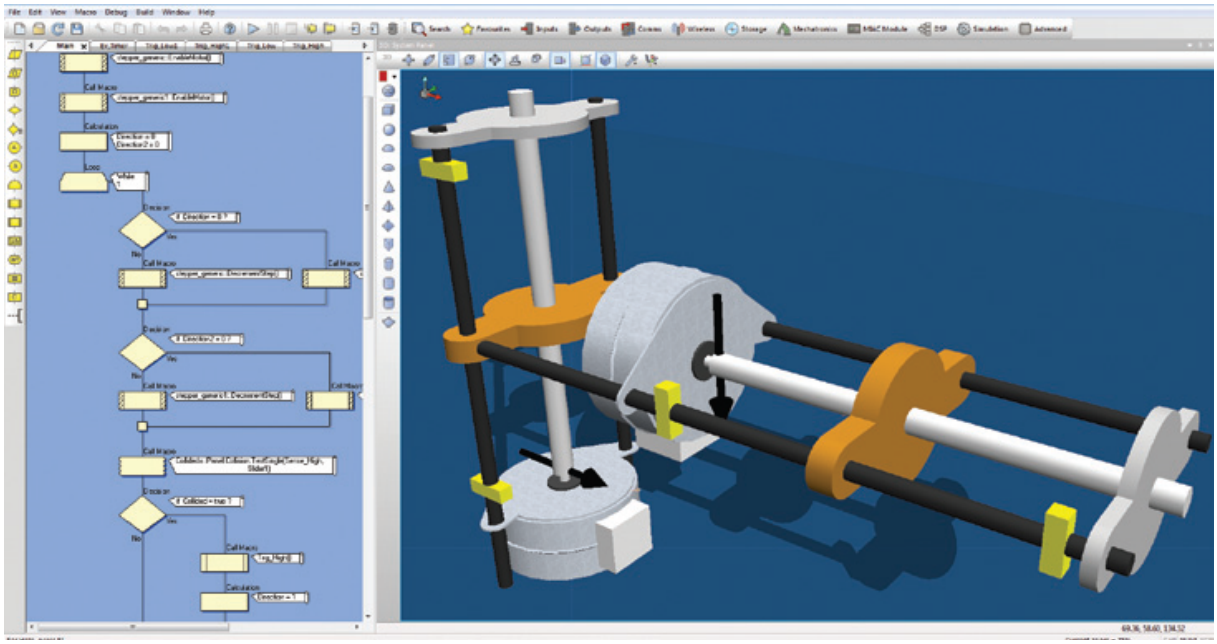

Fig. 1 Flowchart and electromechanical components example in Flowcode [1]

Components used

Just three modules we have used for DC motor driver example:

- Arduino MEGA2560 ADK (rev.3) board (figure 2),
- L298 dc motor driver module (figure 3),
- DC motor with incremental quadrature encoder (in our case it was Faulhaber 2224 IE2-512) on the figure 4.

Certainly, we have used external power supply, because of dc motor supply (maximum current for our dc motor is 3A for 6V).

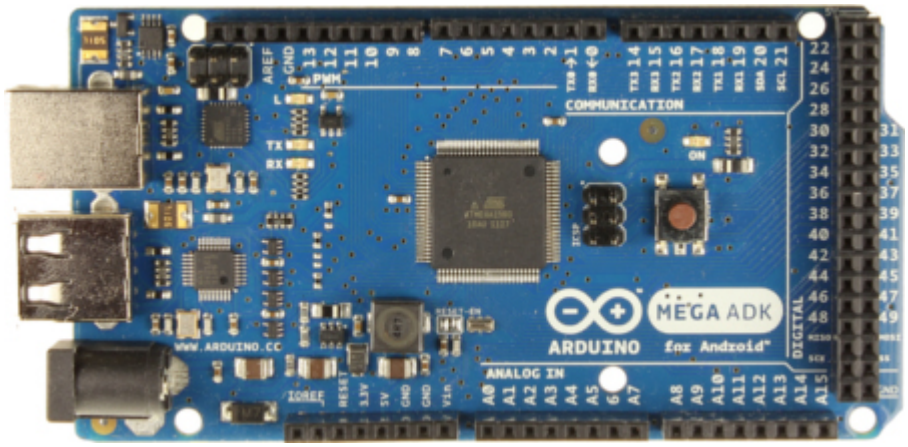

Fig. 2Used Aduino MEGA board

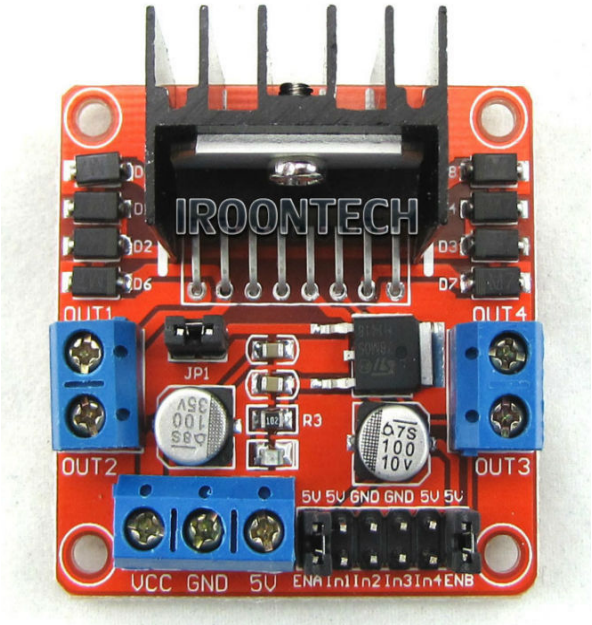

Fig. 3L298 dc motor driver module

Fig. 4Microcontroler Atmel Xmega

Circuit schematic is too simple. Motor encoder channels A and B are connected to AtMega interrupt pins: INT0/PIND.0 and INT1/PIND.1. Digital out 11 (PWM) is connected to dc motor driver board enable/ENA. Digital outs 6 and 7 are connected to In1 and In2 on motor driver board.

DC motor control example

We had decided to use simple PSD regulator:

$$Y_N = K_P.e_n + (S_{n-1} + K_S.e_n) + K_D.(e_n - e_{n-1})$$

Resulting algorithm is consist of three parts:

- Main part
- Encoder interrupt part
- PID control interrupt

Main part

At this part are necessary interrupt enables and PWM output enable. INT0 is set as interrupt from raising edge signal on PIND.0. It is signal A from encoder. TMR0 interrupt is set as sampling rate for PSD controller (approximately 1kHz). PWM_OUT macro sets pulse width modulation signal on Digital output 11. Its frequency is about 8kHz. Algorithm continues with never ending loop. All algorithm is on the figure 5.

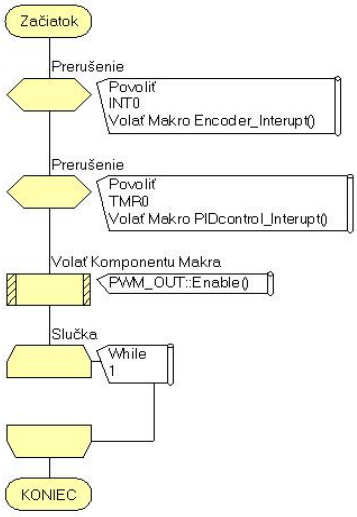

Fig. 5Main algorithm flowchart

Global variables are “pulses” and “requiredValue”. Both are as integer. Variable “pulses” is variable for increment or decrement number of encoder pulses. Next one is variable for setting number of pulses per 1ms.

Encoder interrupt

There is condition which makes choice if dc motor is rotating in clockwise or not. If “yes” then variable pulses is increment else decrement. Variable is set to zero after used by PID control interrupt (approximately every 1ms).

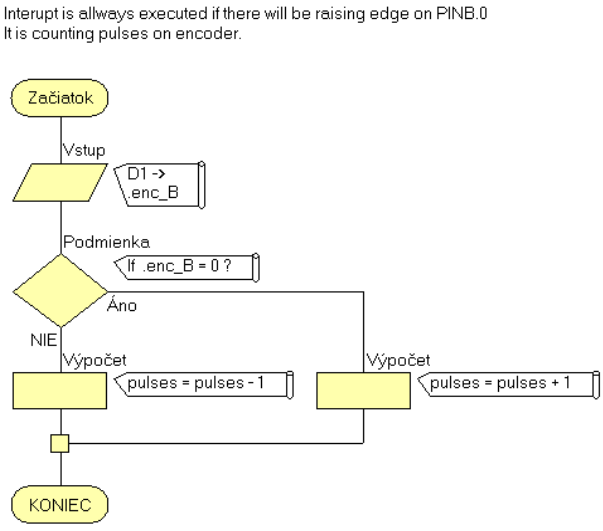

Fig. 6 Encoder interrupt algorithm flowchart

Local variable “enc_B” is temporary for save encoder B channel state.

PID (PSD) control interrupt

Interrupt is perform every 1ms (aprox.). At this interrupt is computing controller output by PSD algorithm (figure 7).

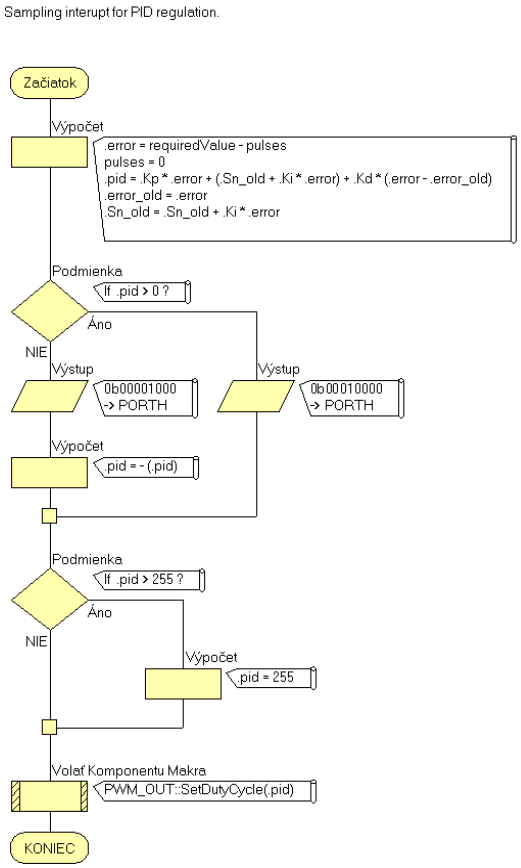

Fig.7 Power section for controlling DC motors

After counting controller output is set direction of rotation and removed sign if local variable "pid" is negative.After that is controlled maximum of "pid". It can be in area 0 to 255. PWM out is set in the and by macro.

Conclusions

This simple example is able to give a lot of knowledge to students who do not have basic programming. Students who are studying robotics in terms of designers are now able to get the basics of programming mobile robots.

In the future we want using Flowcode create complex applications for mobile robotics, which will show some signs of artificial intelligence.

List of sources used

[1] http://www.matrixtsl.com/flowcode/
[2] http://www.robosoccer.sk/

This contribution is the result of the project implementation: Research modules for intelligent robotic systems (ITMS: 26220220141), activity 2.2, supported by the Research & Development operational Program funded by the ERDF.