7TH INTERNATIONAL CONFERENCE ON ELECTROMECHANICAL AND POWER SYSTEMS
October 8-9, 2009 - Iaşi, Romania

A MULTI-APPROACH DESIGN IN TRAINING MOTION CONTROL FOR THE STEPPING MOTORS

Dan S. MIHAI

University of Craiova, Romania, dmihai@em.ucv.ro

Abstract – **The paper deals with some modern design tools for preparing a motion control experimental platform for the stepper motors. Intended as a training tool for the terminal year students in electrical engineering, this platform involves both software and hardware knowledge, from simulation to the management of the peripherals for microcontrollers. Some classical but useful tools, like the algorithmic state machines, are implemented in software. A new very high level programming language is considered: Flowcode. It delivers both the source code, the object code and a powerful graphical user interface for a virtual test of the application. A next paper will present the physical support for the platform.**

Keywords: *Training, Stepper motor, Control design.*

1. INTRODUCTION

The training of the future electrical engineers in the digital control theory and practice is not only a more and more important part of the nowadays teaching curricula but implies many aspects and available tools. The preoccupations of universities in improving both the background content and the computed aided support are obvious – [3], [9], [10], [12]. In the information era there are many new training tools; sometimes an optimization strategy could be necessary, related to a well balanced answers to the arisen questions: the virtual training could be a valuable alternative to physical platform? does it matter if the students make real manipulations with in front equipment or if they use a remote control laboratory far away? the classical design tool for the standard digital circuit have still a justification in the microprocessors era? which modelling and simulation tools are closed enough to the practical implementation? The stepper motors are the best known applications for the digital control, beginning with the standard TTL circuits. Now, this area is full of performing and various solutions. However, the interest in studying and making new researches is still alive – [2], [5]. Not only because the use of stepper (robotics, machines tools, special laboratory equipment) justifies this interest but new improvement are still possible. For the training of the students in electrical engineering, the stepper control will be always a valuable support in the understanding, designing and experiencing both hardware and software knowledge and tools. In this meaning, the author presents a modular system with several components: the tasks identification, the preparing of a background knowledge by modelling and simulation (including animation), the design of the associated automata (with tests of the results with the available software, the design of the control program, the hardware design, implementation.

2. THE SPECIFICATIONS AND THE DESIGN STAGE

The system must be designed as a modular platform based on a RISC microcontroller (μC). A first step is to specify the functions of the equipment and the main on-line tasks for the stepper control and the associated principle solutions:

- the phase commutation (on/off), by sending specific control binary words to the output digital port of the μC;
- the time delay for a step in order to obtain the control frequency and, by that, the motor speed;
- the acquisition of:
  - digital inputs for:
    - Start / Stop;
    - direction for the motion: Direct / Reverse;
    - selection of the control sequence (full step single phase supplied, full step two phases supplied, half step);
    - erasing of the LCD.
  - analog input (potentiometer) for the speed reference;
- the step counting;
- the displaying on LEDs and LCD of several data: the state of the motor; the direction; the control sequence type; the motor speed; the number of steps made by the motor (with sign); the program version.

The design stage means:

- the investigation on the possibilities to benefit from available programs/models/tools inside some

universal programming platform, like Matlab / Simulink, Simplorer (rich in Toolboxes and applications) – [4], [11] and operation with models and simulations (found, created);
- the synthesis of the sequencial equations for the basic controls (by classical methods) and their test by mean of an appropriate simulation program on computer;
- the computation of the program data for the real-time control (timer programming, mainly);
- the choice of a programming support for the real-time control, the program design and its simulation using all similitudes with the target plaform;
- the hardware configuration, by pre-fabricate modules, including or not new own boards (drivers, motors, connection elements);
- the program transfer into the μC;
- the on-line experiments and the improvment of the program and data.

3. THE PREPARATORY DESIGN

A first model could be made in a general-purpose program, like Matlab/Simulink. Fig. 1 presents such a model for the stepper control. For a virtual instrumentation similar with the real equipment, another programs (like MultiSim – [16]) could give good solutions. An intuitif and relevant study for the control sequence is possible by using an animation program – [1], fig. 2.The results will be helpful in filling a control table for each control sequence.

For the Start / Stop controls, operated alternatively from the same puch-button, there several possibilities:

a. The design of a C sequence inserted in the control program:

```
vButtonReleased=1;
{           if (button pressed)
            {           if (vButtonReleased)
                        {           change way();
                                    vButtonReleased=0
                        }
            }
            else        { vButtonReleased=1;
                        }
}
```

b. The sinthesys of an algorithmic state machine (ASM) as for the classical wired circuits with logic gates;

c. A specific real-time sequence based on the interrupt system of the microcontroller (each new request will complement a state variable).

The variant b was tested in simulation and the real-time final solution was c. For the motion directon (Direct / Reverse), an ASM was designed (Huffman synthesis) and, then, the equations were implemented in Flowcode instructions:

$$VDir = Y_1 = B1 \cdot \overline{y_2} + \overline{B1} \cdot y_1;\ Y_2 = \overline{B1} \cdot y_1 + B1 \cdot y_2 \quad (1)$$

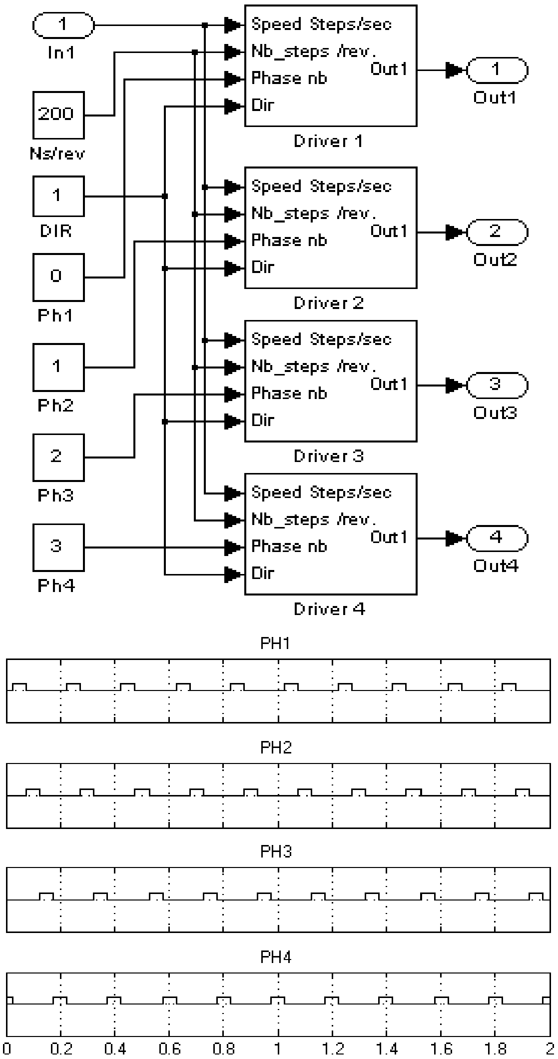

Figure 1:A Matlab/Simulink model for the controller and the phases pulses.

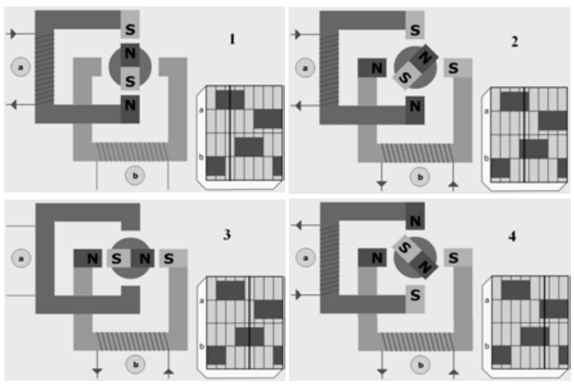

Figure 2: Animation for extracting the control binary words.

These equations were verified using a computer program for the digital circuits, the freeware Digital Woks - [14]. B1 is the push-button Direction connected to the bit B1 of the port B. Y_i and y_i are the secondary functions and the secondary variables, respectively.

B2 is the push-button for the sequence type; the first press is for the full step (single phase) control, a second one is for the full step (double phase) control and a last one is for the half step sequence. The ASM equations obtained by the same synthesis method (Table 1), are:

$$S_S = B_2 \cdot y_1 \cdot \overline{y_0} + \overline{B_2} \cdot \overline{y_1} \cdot \overline{y_0} + y_2 \cdot \overline{y_1} \qquad (2)$$

$$S_D = B_2 \cdot \overline{y_2} \cdot \overline{y_1} + \overline{B_2} \cdot \overline{y_2} \cdot \overline{y_0} \qquad (3)$$

$$S_H = B_2 \cdot \overline{y_2} \cdot y_1 + \overline{B_2} \cdot y_2 \cdot y_1 + y_2 \cdot y_0 \qquad (4)$$

$$Y_0 = B_2 \cdot y_2 \cdot y_0 + B_2 \cdot \overline{y_2} + \overline{y_2} \cdot y_0 \qquad (5)$$

$$Y_1 = \overline{B_2} \cdot \overline{y_2} \cdot y_0 + \overline{B_2} \cdot y_1 + y_2 \cdot y_0 + \overline{y_2} \cdot y_0 \qquad (6)$$

$$Y_2 = B_2 \cdot y_1 + B_2 \cdot y_2 + y_2 \cdot y_1 \qquad (7)$$

The simulation of the circuit (under the same Digital Woks program) activates both local display elements as the recording of the signal diagram – fig. 3. For the final instructions/blocks (Flowcode), some optimizations were made by using supplementary variables joining the most common blocks from (5), like:

$V1 = y_2.y_0$; … $V6 = B_2.\ y_1$.

B_2	0	1	S_S	S_D	S_F	B_2	0	1	S_S	S_D	S_F
A.	①	4	1	0	0	B.	①	2	1	0	0
	②	5	0	1	0		3	②	0	1	0
	③	6	0	0	1		③	4	0	1	0
	2	④	0	1	0		5	④	0	0	1
	3	⑤	0	0	1		⑤	6	0	0	1
	1	⑥	1	0	0		1	⑥	1	0	0

ⓧ denotes a stable state.

Table 1. The control sequence synthesis table - 2 variants.

4. THE CONTROL PROGRAM

A powerful very high level language program was preferred – [15]. The main loop is presented in fig. 4. Fig. 5 gives the image of some modules (the interrupt routine associated with the Start / Stop control, the time delay, step counting and its displaying, for the ASM implementation) and presents a part of the ghaphical user interface in standby mode, making visible the activated modules (from all 25 units) and the variables.

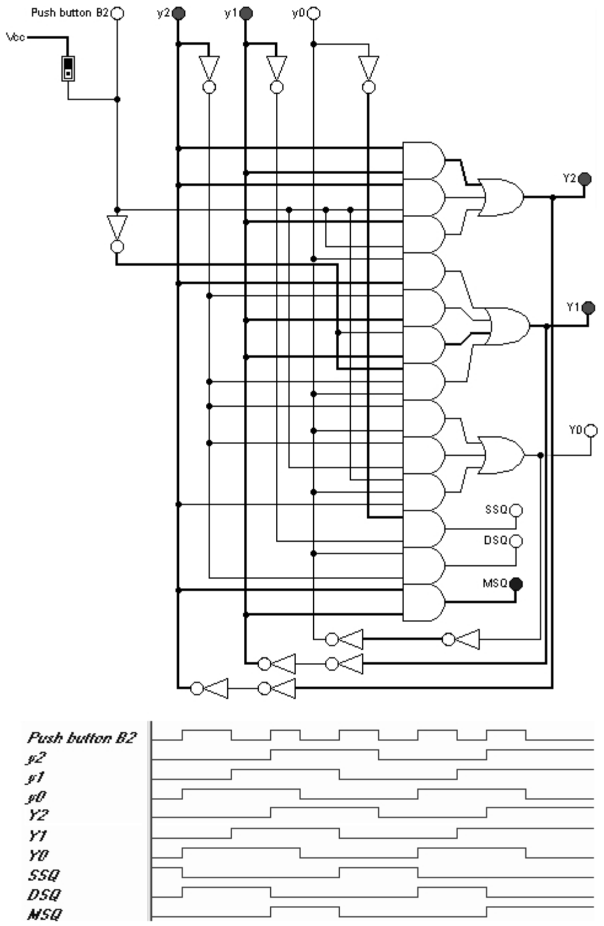

Figure 3: A wired variant of the ASM for the 3 control modes and the simulated signals.

5. CONCLUSIONS

The computer age brings always new tools both for the software and the hardware design associated with motion control applications. A more or less complex platform for the stepper motors remains a fruitful teaching application. The paper reveals the design chain using different classical methods and modern tools for modelling/simulation of the control structure. The author considers a new and powerful graphical language for the control program (Flowcode), with many facilities in exploiting the hardware and software abilities of the target microcontroller.

References

[1] Abati P., *Animation sous flash sur le moteur pas à pas* - http://pagesperso-orange.fr/xcotton/electron/coursedocs.htm.

[2] Dorf R.C., Bishop R.H., *Modern Control Systems*, 9th ed. Englewood Cliffs, NJ, Prentice Hall, 2000.

[3] Gettrick A. Mc, Theys M. D., Soldan D. L., Srimani, P. K., *Computer engineering curriculum in the new millennium*, IEEE Trans. Educ., vol.46, no. 4, pp. 456–462, Nov. 2003.

[4] Gonzalez S. Al., *Stepper Motor Toolbox For use with MATLAB,* Mar del Plata, ARGENTINA July, 1999.

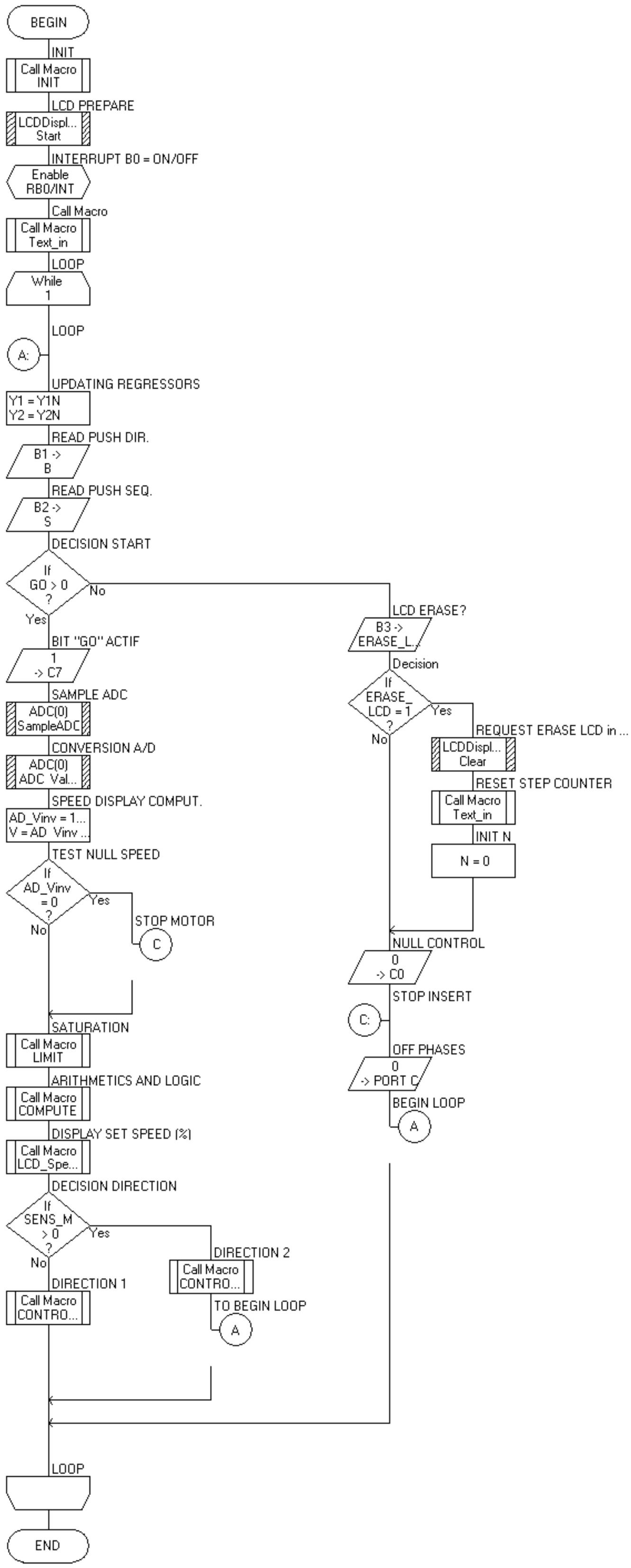

Figure 4: The main Flowcode program (loop) for the stepper control.

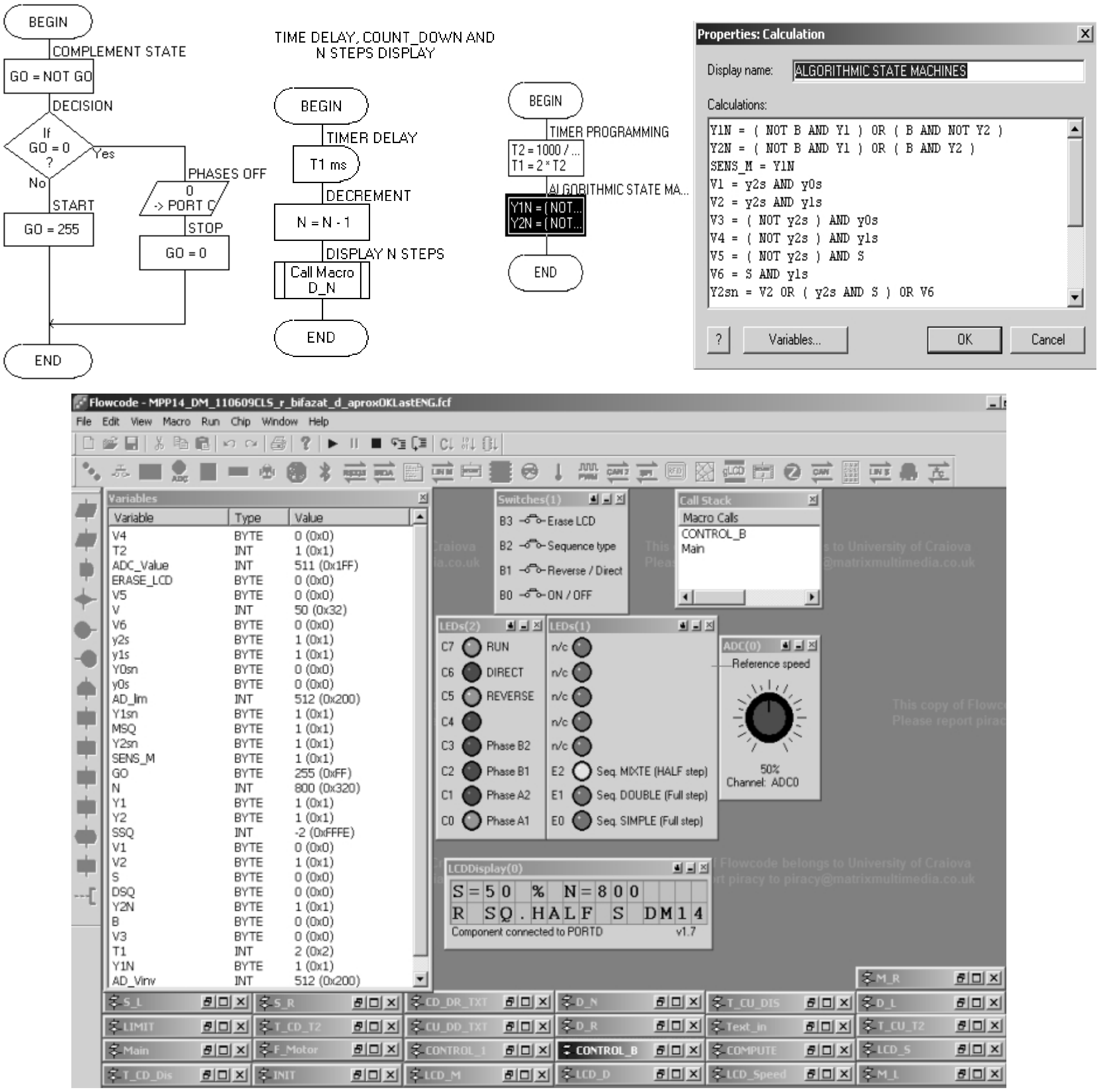

Figure 5: Samples from the Flowcode program for the stepper motors.

References

[5] Gumma S., A, *Radial Basic Function Neuro Controller for Permanent Magnet Stepper Motor*, Master Thesis,Science in Electrical Engineering degree, Cleveland State Umiversity, USA, 2004.

[6] Mihai D., *Systèmes d'entraînements électriques I. Problèmes fondamentaux*, Ed. Universitaria, Craiova, 2004.

[7] Mihai D., *High Quality Modelling and Simulations for On-Line Digital Control Pre-Analysis*, ACEMP '01, Kusadasi, 2001, Proceedings, pp. 373-378.

[8] Moallem M., *Design and Implementation of Computer Control Software, A low-cost laboratory setup*, IEEE Control Systems Magazine, February 2005, pp.26 -29.

[9] Murray R.M., Åström K.J., Boyd S.P., Brockett R.W., Stein G., *Future Directions in Control in an Information Rich World*, IEEE Contr.Syst. Mag., vol. 23, no. 2, pp. 20–33, 2003.

[10] Nooshabadi S., Garside J., *Modernization of Teaching in Embedded Systems Design - An International Collaborative Project*, IEEE Trans. Educ, vol. 49, no. 2, May 2006.

[11] Ong, C.M., *Dynamic Simulation of Electric Machinery using Matlab / Simulink*. Prentice Hall, Upper Saddle River, 1998.

[12] Shuhui Li, Challoo R., *Restructuring an Electric Machinery Course With an Integrative Approach and Computer-Assisted Teaching Methodology*, IEEE Trans. Educ, vol. 49, No. 1, Febr. 2006.

[13] *An Introduction to Microcontroller Programming*, CDRom ELFCSSI3, Matrix Multmedia, London, 2007.

[14] *Digital Woks 3.04*, www.electronics-lab.com.

[15] *Flowcode V3.4 for PIC*, CDRom TEFLC, Matrix Multimedia, London, 2008.

[16] *MultiSIM 9*, Electronics Workbench Corporation, Toronto, Canada, 2005.