A MULTIMEDIA ENVIRONMENT FOR TEACHING AND LEARNING THE INTERFACING OF EXTERNAL DEVICES WITH A MICROCOMPUTER

José Celso Freire Junior[1], Samuel E. de Lucena[2], Ulisses Pimentel[3] and Wando N. Rocha[4]

***Abstract* ¾ *This paper presents a multimedia system, developed with Borland C++ Builder, to provide a dynamic environment for teaching and learning interfacing techniques between personal computer (PC) and external devices connected to its parallel port. Also shown in this work is a didactic circuit board with many usual electronic devices, which has been designed by the authors as a complementary tool. The multimedia teaching and learning system presents three integrated environments, called Theory, Compiler and Notes, which allow students to follow and/or put into practice some microcomputer interfacing techniques. The integrated teaching environment facilitates immensely the learning of the interfacing between microcomputers and real world devices (such as step motors, displays and data converters), enabling the students to absorb concepts and techniques present in the day-to-day of current industrial systems. While a long-term assessment for the developed system has been started, preliminary results with a limited number of students have already demonstrated its suitability and power.*

Index Terms ¾ *Computer-based learning, computer interfacing, data acquisition, PC parallel port programming.*

INTRODUCTION

Microcomputer-based industrial controllers are nowadays universally accepted and can be found everywhere. Time-to-market, ease of design, and development cost, however, have reduced the controller development to the integration around a personal computer (PC) of commercially available off-the-shelf hardware blocks, such as signal conditioning, data acquisition and control boards, wherein general-purpose or specialized control software does the rest of the job [1]. One can hardly forget Matlab™ and National Instruments™ names, to list just two of a myriad of microcomputer-based control software and hardware suppliers. On the other hand, from the viewpoint of electrical engineering education, board-level hardware design and ready-to-use control software should be postponed until students have mastered component level design and they are able to write their own computer programs to control the interactions of microcomputer and external world [2].

To master microcomputer-based control, electrical engineering students must know the basics of the following three distinct subjects: 1) electronics devices; 2) computer architecture; 3) and high level programming language [3], [4].

In the following sections a multimedia environment will be described which is composed of a knowledge base about computer architecture, a C language editor and compiler, and a window in which students can take notes about the experiences they are carrying out. Also included is a brief description of a circuit board with fundamental electronic devices to allow interactive experiences.

THE MULTIMEDIA LEARNING ENVIRONMENT – MLE

The Multimedia Learning Environment offers the students three integrated environments to facilitate their hands-on study of the interfacing and control of external devices connected to the PC's parallel port. These environments are named *Theory*, *Compiler* and *Notes*. The *Theory* environment, whose initial window is shown in Figure 1, is invoked by the *Theory* tab to present much detailed and theoretical information about the PC, such as the CPU types and features, motherboards, standard bus, serial and parallel ports, etc.

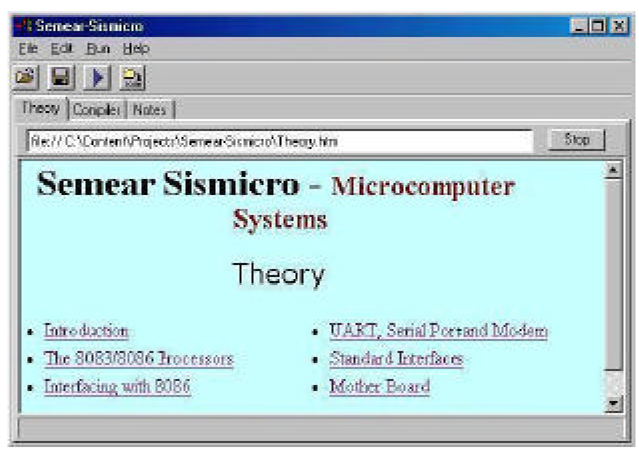


FIGURE 1
MLE – THEORY ENVIRONMENT.

[1] José Celso Freire Junior, UNESP/FEG/DEE, Av. Ariberto Cunha, 333, 12516-410, Guaratinguetá, SP, Brazil, Jose-Celso.Freire@feg.unesp.br
[2] Samuel E. de Lucena, State University of São Paulo(UNESP), Av. Ariberto Cunha, 333, 12516-410, Guaratinguetá, SP, Brazil, lucena@feg.unesp.br
[3] Ulisses Pimentel, State University of São Paulo (UNESP), Av. Ariberto Cunha, 333, 12516-410, Guaratinguetá, SP, Brazil, ele97058@feg.unesp.br
[4] Wando N. Rocha, State Universtity of São Paulo (UNESP), Av. Ariberto Cunha, 333, 12516-410, Guaratinguetá, SP, Brazil, ele97064@feg.unesp.br

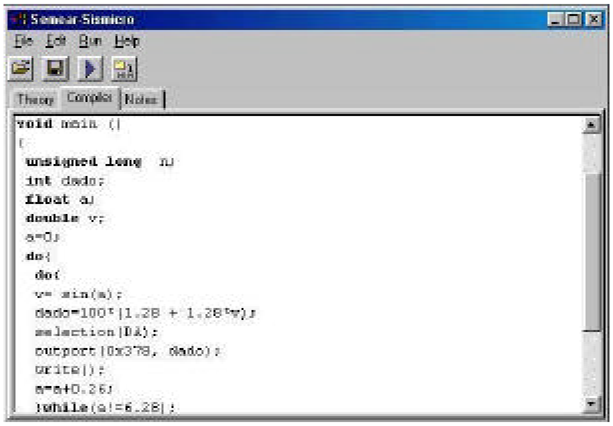


FIGURE 2
MLE – COMPILER ENVIRONMENT.

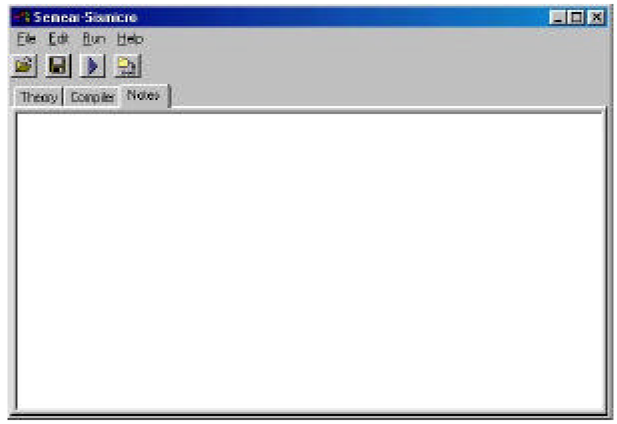


FIGURE 3
MLE – NOTES ENVIRONMENT.

The *Compiler* environment is selected with the *Compiler* tab. This option allows the students to edit, compile, and run C programs. The students can create and test their own programs or, at first, they can recall prewritten programs from a library. It has been employed a freeware standard C compiler [5]. Figure 2 shows the Compiler form containing some lines of a program to synthesize a sinusoidal signal.

On clicking at the *Notes* tab, students can easily and conveniently prepare and file their laboratory reports, as the laboratory classes go on. Figure 3 presents a void example of the *Notes* form. The three environments employ identical page control buttons.

The MLE has been developed using Borland C++ Builder because of its easy of programming, powerful features and widespread use [6]-[8].

To provide the students with a hands-on study about computer interfacing, but avoiding the use of cumbersome internal plug-in control cards, it has been designed and assembled a special board to be connected to the PC via its parallel (printer) port, as schematically shown in Figure 4. The following section is a brief description of it.

The Parallel Port Plug-in Circuit Board (LTH)

The LTH can be easily plugged in the computer parallel port, using a standard DB-25 connector, avoiding the annoying and time-consuming opening of the computer case to insert control cards [9]. Otherwise, cumbersome commercial multiple control cards had to be employed, to include all the devices present in the LTH. Figure 5 shows the LTH prototype, while a summary of its components can be seen in Table 1.

In order to interface so many different modules via a microcomputer parallel port, its 12 standard lines have been used in the creation of a common data bus, as well as a control bus containing general-purpose and device-specific signals. Accordingly, by emitting instructions in a suitable

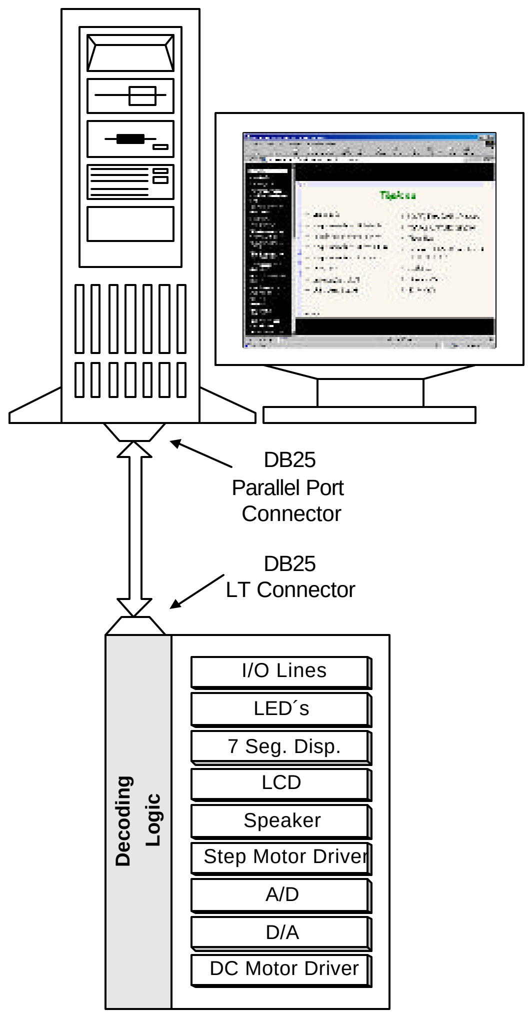


FIGURE 4
GENERAL VIEW OF THE MULTIMEDIA LEARNING ENVIRONMENT (MLE).

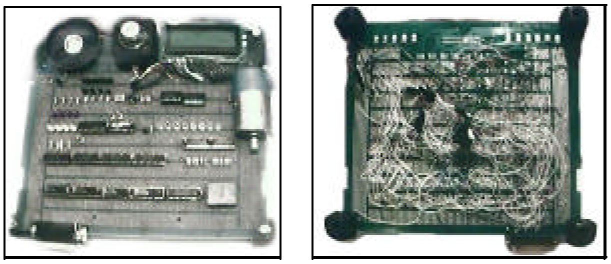


FIGURE 5
TOP (LEFT) AND BOTTOM (RIGHT) SIDES OF THE LTH PROTOTYPE.

sequence, as can be seen in Figure 8, the microcomputer can communicate with all devices present in the module, one at a time, to carry out a desired control task. A detailed description of the LHT has been published elsewhere [10].

APPLICATION EXAMPLES

Those skilled in the art will be well aware of the various application possibilities that can be made with the present multimedia system, together with the LTH module. Nevertheless, an illustrative list of some fundamental experiments has been included in Table II, to illustrate the many different experiments students can carry out.

The analog channels of the LTH, that means the A/D and D/A converters, virtually enable students to carry out many different experiments, well beyond those which can be done directly using only the LTH. For the sake of illustration, students could readily implement a thermometer, a temperature or velocity controller, an arbitrary signal generator, and many other different applications, according to the sensors and actuators they could put together with the LTH via breadboard or wire-wrap techniques.

Referring to the D/A only, students could write programs to generate several waveforms and show them in an oscilloscope. Some basic waveforms are sinusoidal, triangle, saw-tooth, square and pulse train signals. Arbitrary waveforms can be easily synthesized as well. Figures 6 and 7 are examples of four waveforms generated by the MLE and LTH and captured with a digital storage oscilloscope.

As of the A/D channel, students may readily put into practice the basic principles of data acquisition, such as A/D timing and minimum sampling frequency (Shannon's sampling theorem). To this end, samples of a sinusoidal function generator signal may be collected by the A/D and sent back to the analog world by the D/A. Meanwhile, a two-channel oscilloscope shall be displaying the A/D input and the D/A output signals. Students from university *campus* to whom this kind of experiment have been presented have reported enthusiastically on their learning.

In the *Theory* control page, there are many examples of debugged programs, written in C, to help students understand the PC-LTH interaction. These programs can be compiled and run in a step-by-step way, while students observe the effect of each individual instruction on the LTH.

TABLE I
LTH - ELECTRONIC DEVICES

- Analog-to-digital converter (1 channel, 8 bits)
- Digital-to-analog converter (1 channel, 8 bits)
- Step motor driver (Unipolar, 2 phases, 12 V @ 500 mA,) and 4 control lines
- DC motor driver (12 V @ 1 A)
- Alphanumeric liquid crystal display (2 rows, 16 columns)
- 7-segment LED display (2 digits)
- Speaker with driver (5 V @ 100 mA)
- 8 LED's with individual transistor driver
- 8 general purpose input/output lines

Figure 8 shows a flowchart and a C source code listing of a program to turn-on the 8 LED's of the LTH back and forth, one at a time. *Theory* environment of MLE contains several programs of this kind, one dedicated to each device of the LTH.

COURSE AND CLASS PLANNING

The MLE has been designed to improve the teaching and learning of microcomputer interfacing techniques. In the State University of São Paulo, at Guaratinguetá, undergraduate electrical engineering students have to attend a half-year long subject matter named Microcomputer System in their fifth term. This discipline consists of both theoretical and laboratory classes.

A general description of PC hardware operation is presented to the students once a week, during audiovisual 1-hour 40-minute long classes. Most PC hardware subsystems are discussed in great detail, e.g. parallel and serial ports, interrupt and DMA (direct memory access) controllers. On the other hand, in the laboratory classes, the professor gives additional explanation about lab specific topics (not deeply treated in theoretical classes). Then, he divides the students into a number of groups and gives them a predefined carefully chosen practical task (for instance, the control of a step motor shaft speed and position). The students spend a whole afternoon creating and testing the program. Now it is hope that the availability of the MLE will increase the course efficiency very much, as students feel encouraged by a dynamic, complete and practical learning environment.

TABLE II
EXAMPLES OF POSSIBLE EXPERIMENTS

Device	Suggested Experiment
• Speaker	• Tone generation with several pitches
• LED's	• Chasing, Bar graph indicator, etc.
• 7-Segment display	• Counting, state indicator.
• LCD display	• Messaging, digital clock.
• Step motor driver	• Shaft position and speed control.
• DC motor driver	• Shaft position and speed control.
• D/A converter	• Waveform generation, control signal.
• A/D converter	• Data acquisition for recording or control.
• I/O lines	• Key state detection, relay control signal.

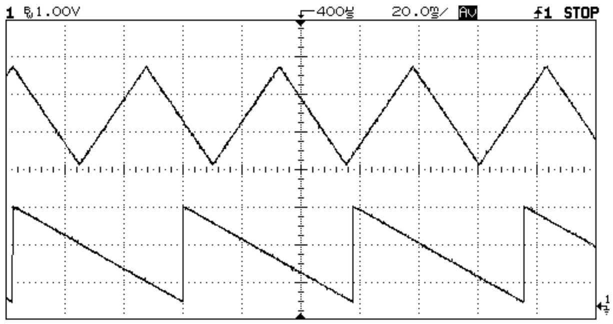


FIGURE 6
EXAMPLES OF WAVEFORM SYNTHESIS WITH THE MLE AND LTH.

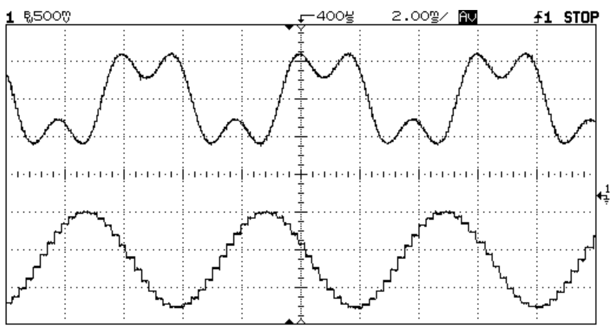
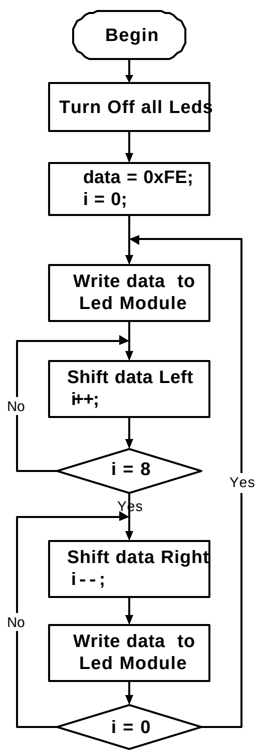


FIGURE 7
EXAMPLES OF WAVEFORM SYNTHESIS WITH THE MLE AND LTH.




```
#include <stdio.h>
#include <conio.h>

void write_led( int data)
{
  outportb(0x378, 0xFB);
  outportb(0x37A, 0x01);
  outportb(0x378, data);
  delay(1);
  outportb(0x37A, 0xFA);
  outportb(0x378, 0xF2);
  outportb(0x378, 0xFF);
  outportb(0x37A, 0x01);
  return;
}

void main(void)
{
  int i, data;
  write_led(0x00);
  data = 0xFE;

  do{
    for(i=0;i<8;i++){
      write_led(data);
      data = data << 1;
    }
    for(i=8;i>0;i--){
      write_led(data);
      data = data >> 1;
    }
  } while(!kbhit());
  return;
}
```

FIGURE 8
EXAMPLE OF FLOWCHART AND C PROGRAM.

CONCLUSION

This paper has described the development of a multimedia environment, designed in Borland C++ Builder, made of three integrated modules: a specialized electronic textbook (in HTML) on microcomputer interfacing, a standard C compiler, and a notepad window. The assessment of the entire system by students has shown that it makes it easier to teach and learn PC interfacing with real world devices, mainly as a result of the PC's friendly interface, the fast and interactive results and the hands-on training. The multimedia environment provides the students with an easy-to-use, hands-on learning tool, a trend in modern teaching technique [11]. Although excellent partial results have been obtained up to the present, a full assessment of the MLE, concerning its suitability and effectiveness as a learning tool and teaching aid is under way.

REFERENCES

[1] Schmalzel, J.L., "*Improving instrumentation and measurement education*", IEEE Instrumentation & Measurement Magazine, March 1999, pp. 10-13.

[2] Blyler, J.E., "*The softening of hardware*", IEEE Instrumentation & Measurement Magazine, March 2000, pp. 46-47.

[3] Horowitz, P., Hill, W., "*The art of electronics*", 2nd edition, Cambridge, Cambridge University Press, 1989.

[4] Buchanan, W., "*PC interfacing, communications and windows programming*", Essex, Addison–Wesley, 1999.

[5] Inprise Corporation, "*The Borland C++ Builder Compiler 5.5*", http://www.borland.com/bcppbuilder/freecompiler, 2000.

[6] Hollingworth, J., Butterfield, D., Swart, B., Allsop, J. (major authors), "*C++ Builder 5 Development Guide*", Indianapolis, SAMS Publishing, 2000.

[7] Miano, J., Cabanski, T., Howe, H., "*Borland C++ Builder How-To*", Corte Madera, Waite Group Press, 1997.

[8] Spanghero, A., "Aprendendo C++ Builder 3", São Paulo, Makron Books, 1999 (in Portuguese).

[9] Gadre, D. V., "*Programming the parallel port – interfacing the PC for data acquisition and process control*", Lawrence, R&D Books, 1998.

[10] Freire, Jr., et al., "*Development of a computer-based learning tool: an undergraduate research project experience*", International Conference on Engineering and Computer Education, São Paulo, 2000.

[11] Jeon, J.W., "*A microcomputer course: designing and implementing personal microcomputers*", IEEE Transactions on Education, Vol. 43, No. 4, November 2000, pp. 426-433.