Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

Hardware Simulation and Debugging For Microchip RISC Microcontrollers

Aurel Gontean, Mircea Băbăiţă, Roxana Jibleanu[1]

Abstract – This paper introduces the possibility of using simultaneously both the Microchip MPLAB IDE and the Lab Center Proteus simulation software. Differences between the simulated results and the real world behavior are emphasized in terms of timing and initialization. Source code debugging (stepping and free run modes), assembling and compiling within Proteus are discussed.

Keywords: simulation, microcontroller, debug

I. INTRODUCTION

Programming embedded systems usually rely on Integrated Development Environment software, such as Microchip's MPLAB IDE (Fig.1). The source code is edited, simulated and eventually programmed in the microcontroller using the same application - a step ahead compared with the command line DOS based versions of earlier assemblers.

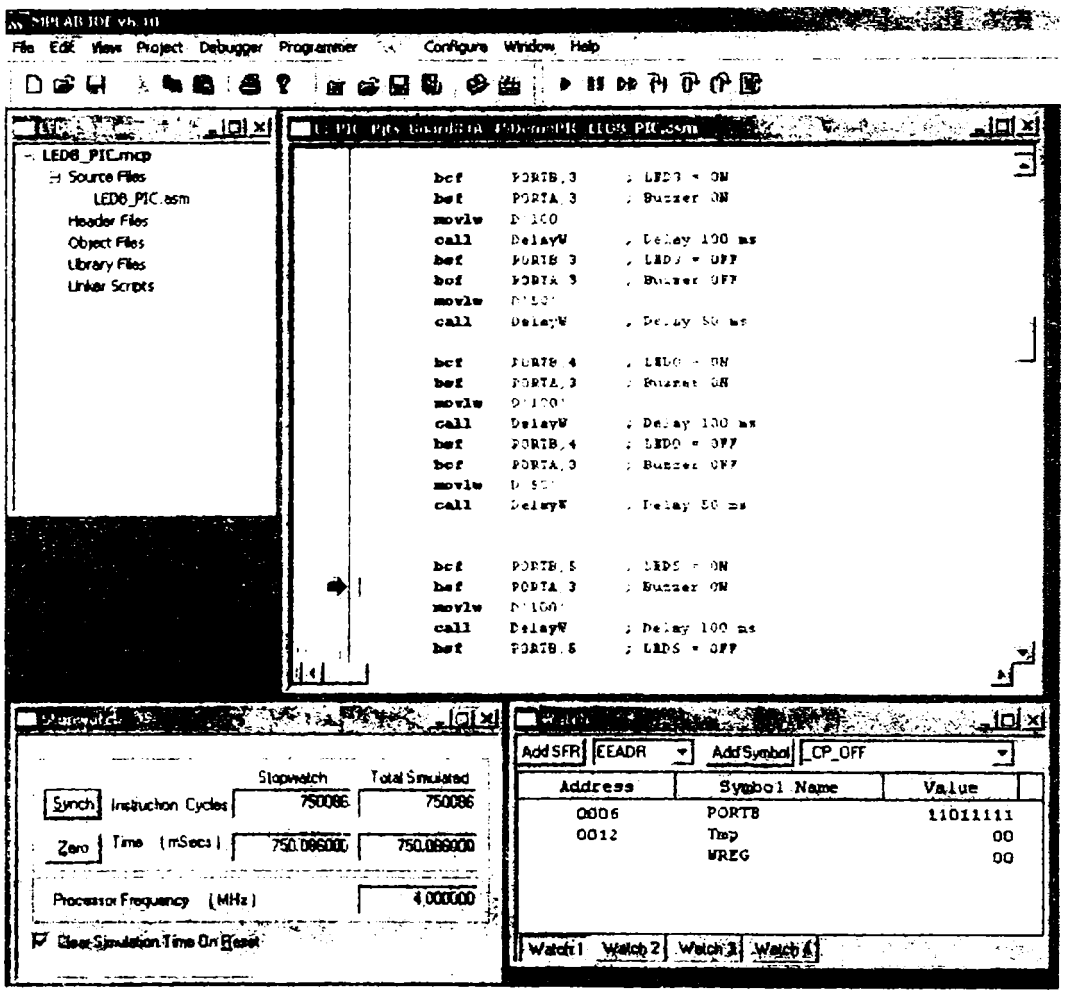

Fig. 1 MPLAB IDE - A Typical Integrated Development System.

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Electronică Aplicată, Bd. V. Pârvan Nr. 2, 300223 Timişoara, e-mail {aurel.gontean, mircea.babaita, roxana.jibleanu}@etc.utt.ro

Although evolved, modern IDE software is not hardware oriented – it is impossible to simulate continuous voltage values, currents, and transistors not to mention LEDs, LCDs, buzzers or motors. There are areas not covered at all by the IDE software RS232, I²C, or SPI communication, or difficult to use PWM, soft delays. Therefore there is a high demand for an appropriate simulation tool in both design and teaching embedded systems.

II PROTEUS SIMULATOR

Proteus software offered by Labcenter Electronics is a solution allowing for mixed analog and digital simulation, along with models for Microchip midrange microcontrollers.

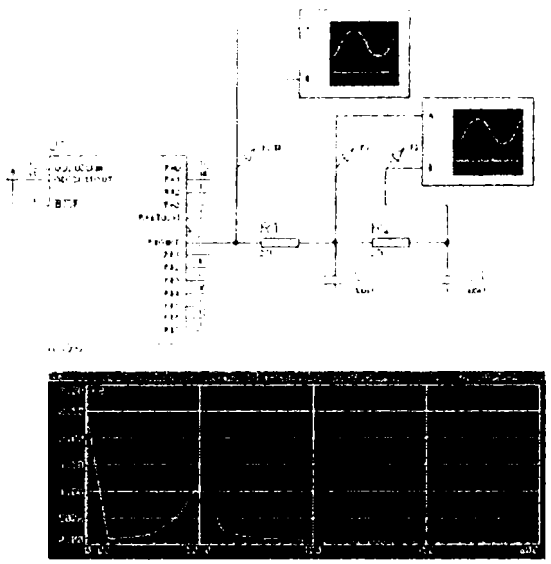

Fig. 2. Using Fourier analyses to evaluate the output filtered signal

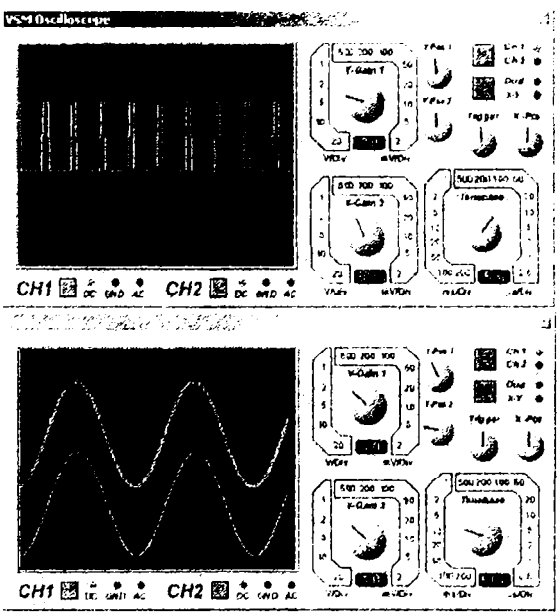

Fig. 3. PWM Simulation Results

A software PWM generator is almost impossible to be evaluated under MPLAB control. Once built, the hex file is loaded in the microcontroller (Fig. 2) and the dynamical behavior may be analyzed with virtual instruments (Fig. 3).

The combination of the two software packages allows the user to design most or even all of the schematics and to edit, debug and verify the software with little or even no need for prototyping, shortening the overall design time.

A. *Hardware interaction*

Beyond pure software simulation, Proteus offers also hardware interaction. A microcontroller model with a digital sounder may work together in order to generate audible sounds. Fig. 4 demonstrates a typical case where Proteus VSM is running a simulation of a PIC program which generates audio tones in real time. The sounder model picks up the transitions on port A, RA3, and converts them into a 44 kHz data stream which is sent to the sound card. On a Pentium II or better PC, the simulated PIC will run fast enough to generate audio tones in real time.

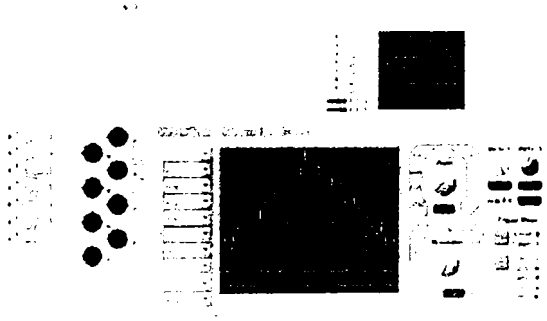

Fig. 4. Using the Logic Analyzer

B. *Using displays*

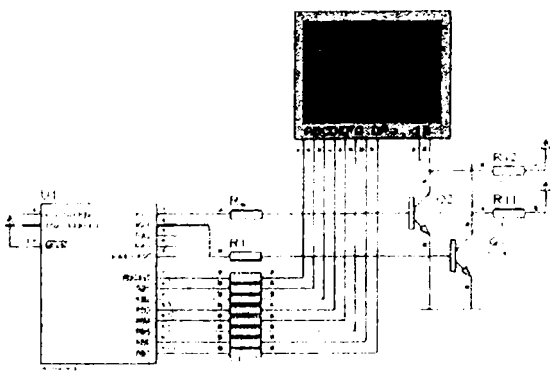

Fig. 5. Using Digit Multiplexing

A variety of displays is a key feature of the simulator. Fig. 5 introduces the 7 digit display multiplexing schematic, where the real behavior is fairly closed with the simulated one.

Another useful feature is the LCD simulation for the standard LM032 display (fig.6). The simulation supports both 4 bit and 8 bit modes, while busy flag can be read on D7, exactly like in the normal use.

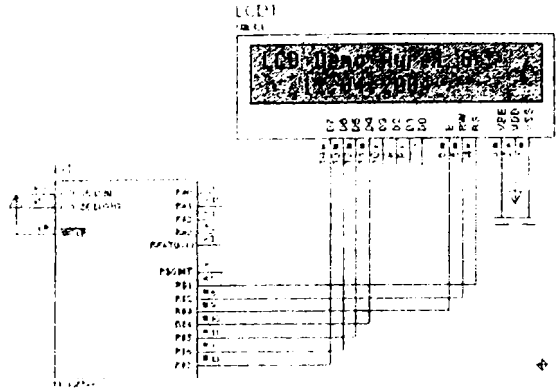

Fig.6 The LM032 LCD simulation

C. *Serial transmission*

Serial transmission may be accomplished either with the virtual terminal (Fig. 7), or using the hardware interaction implemented in the Compim device. The virtual terminal allows reading and transmitting data using standard TTL levels, while the Compim device reads and writes an existing Com port in the PC. XON/XOF and hardware control may be used and framing error is signalized through proper indicators.

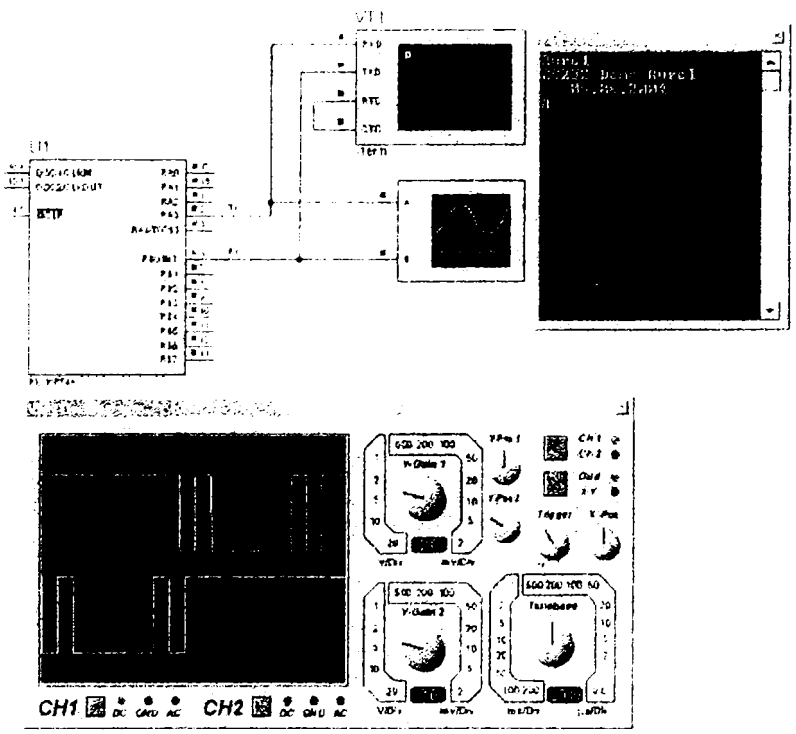

Fig.7 RS-232 Transmission

III. LIMITATIONS AND ERRORS

MPLAB IDE is a powerful tool which is continuously improved by Microchip. Some peripherals are not supported at all (such as the USART), and others are only partial simulated (the ADC conversion does not change the ADRESH and ADRESL registers). There are differences in timing – Timer1 has a constant 4 µs error compared with the real behavior.

Proteus simulator has also some errors, for example the 74LS148 model is wrong – the $\overline{I_0}$ input does not trigger the EO nor the $\overline{GS}$ lines, as it should (Fig. 8). Another drawback is that only single file code can be source code debugged – there is no possibility to link files within Proteus. However difficult items such as banking are accurate simulated – there are programmable warnings for stack under -or overflow, and incorrect jumping computation.

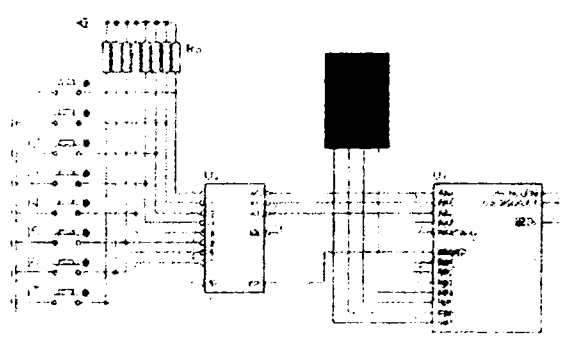

Fig. 8 There is no reaction for the I_0 input line stimulus

IV. COMBINED ACTION

For educational and design purposes, the best result is accomplished working using both packages. Changes in MPLAB source code are reflected in the .hex file and loaded in background in the Proteus model each time a new simulation is started (there is no need to restart the simulator). Working with an ICSP programmer is a low cost debug solution, suitable for students; while an ICD2 debugger is a professional approach at a moderate price, with the advantage of stepping through the code, breakpoints, real timing, the Proteus simulation may replace it for most cases.

V. CONCLUSIONS

The combination of the two software packages allows the user to design most or even all of the schematics and to edit, debug and verify the software with little or even no need for prototyping, shortening the overall design time.

For teaching purposes, the overall performance is excellent, the learning curve being dramatically reduced.

REFERENCES

[1] http://www.microchip.com

[2] http://www.labcenter.com

[3] • • • "*MPLAB® IDE Simulator, Editor User's guide*", Microchip, 2001

[4] • • • "*MPLAB® ICD2Debugger. User's guide*", Microchip, 2002.

[5] A. Gontean, *The PIC16F84A Microcontroller*, Editura Orizonturi Universitare, 2004 (in Romanian).