

Teaching and Practice Innovation of Embedded System Design Course Based on Proteus and Keil-electronic Stopwatch as an Example

Li Dan¹, He Erhu¹, Zhang Yanrong² and Zhang Yu¹

¹*Information and Computer Engineering College, Northeast Forestry University, Harbin, China*

²*College of Computer and Information Engineering, Harbin University of Commerce, Harbin, China;*

ld725725@126.com, Jacob_heh@126.com, zhangyanrong_5@163.com, wenyuan1206@qq.com

Abstract

This study aims at the status quo that theory is divorced from practice in embedded systems teaching process. A new theory teaching method is proposed that using Proteus as hardware simulation environment and Keil as software design environment. With this new method, in the teaching process, the hardware circuit can be simulated by the software environment. The joint use of Proteus and Keil in Embedded System Design Course teaching is clarified by an example of the design of electronic stopwatch which is using GPIO and timer in ARM7. This instance illustrates the flow and application methods about the joint application of Proteus and Keil in embedded system course teaching and practical application detailedly. By this method, it can stimulated students' interest in learning, and enhance the students' understanding of the principles and application of Embedded System Design. Thereby using this method can also improve the embedded system experiment teaching effect.

Keywords: *ARM, Practice teaching, Proteus, Keil, Electronic simulation*

1. Introduction

Embedded Systems Design is the main course of computer, electronic information, automation and other majors. Through learning this course, students can grasp the concept, architecture, system components and design methods of the embedded system. Besides, students can understand the architecture and instruction system of embedded processors. Furthermore, they can know some other knowledge as well, such as the overall structure of embedded processors, memory organization, system control module, the I/O peripheral control module, microprocessor architecture and instruction set, and the analytic and designed methods of the embedded system. The traditional teaching process of the embedded system use the teaching model which is "the theory explanation firstly, and then hands-on experiments." This teaching model uses a large number of principles to guide the students to get started. By this way, students feel boring and have trouble in understanding. Meanwhile, the overemphasis of the processor core principles and the disjuncture of the application and theory result in the scarcity of students' practical ability. Besides, the course is introduced by separating the introduction of hardware and software, and the system design methods which combine software with hardware are not stressed enough. So students are lack of the systematic way of thinking when building real applications. In practical aspects of traditional education process, students mainly use the embedded chamber or the embedded development board. And in the practical process, students do mechanical connection merely, so that they cannot master the detailed circuit

design and schematic design. Moreover, because there is a separation between practice and theory, the course cannot obtain the good teaching effect.

EDA stands for Electronic Design Automation. The emergence of EDA technology greatly improves the efficiency of the circuit design and reduces the labor intensity of designers. This study proposes assembling electronic components and conducting circuit simulation based on EDA technology in the teaching process. For that matter, it can strengthen students' understanding of the principles of embedded system design [1, 2].

2. The Teaching and Practice Innovation

Throughout the design process of teaching and practice, top-down design method is adopted. According to the different application, the design process is divided into five steps, such as requirements analysis, specifications, hardware and software building design, system integration, and system testing. This process starts with the most abstract description of the system, and then refines the details.

1. Requirements analysis: Distinguishing and refining hardware and software according to the experience of engineers. Determining the work which puts forward complete, accurate, clear and specific demands to the target system

2. Specifications: Refining the needs. Getting the system specifications as formal design guidance and standard of acceptance;

3. Hardware and software building design: First divided hardware and software according to the function, and then implemented software function by Keil IDE. Selecting electronic components involved in use by Proteus. Drawing hardware circuit diagram as the virtual hardware circuit simulated by embedded system.

4. System Integration: Loading the software design process into simulation circuit for system simulation.

5. System test: Carrying out the system functional test and the procedure stability test by the register, simulation oscilloscopes and other components in analog circuit. The whole innovation process is shown in Figure 1.

Figure 1. The Process of Innovation Process

In the specific implementation process, the whole design process can be divided into the software platform design and the hardware platform simulation. The software design is built by Keil integration framework platform. Keil for ARM integrated development environment is μ Vision IDE. It is a window-based software development platform, and it integrates a powerful editor, a project manager, and makes tools. μ Vision IDE integrated tools includes the C compiler, the macro assembler, the linker / locator and the hex file generator. μ Vision has two kinds of work mode ,compiling and debugging. In the two modes, designers can view and modify the source file. They can also use single-step execution tools and software simulation tools to observe the operation of the program and the content change of registers, the on-chip RAM and chip RAM, ROM with the program running. This approach is very suitable for the software simulation experiments.

Hardware circuit simulation is carried out by Proteus, which is the world's leading EDA tool (simulation software). From the schematic layout, code debugging to the SCM and peripheral circuit co-simulation, a key switch to the PCB design, it truly achieves completely design from concept to product. Proteus is a design platform which combines circuit simulation software, PCB design software and virtual models simulation software. The whole simulation of the circuit design process in Proteus is as follows:

- (1)Open the Proteus, and select the button 'Component' in the toolkit.
- (2)Click the button 'P' in the Object Picker, and then the dialog 'PicDevices' will be popped up. Next, select some components needed for the project. This experiment mainly incorporates SCM, capacitance, resistance, crystal oscillator, button, Nixie tube, power source and so on.
- (3)Set the values of these components. First, select an object by right click. Then, open the properties dialog. Next, input the required value in the textbox 'Component Value'. Finally, finish the setting by click 'OK'.
- (4)Write the file about the instruction of the circuit. Click the button' Text script' and open the editor dialog. Afterwards, write the file at the apposite place of the circuit.
- (5) Arrange the components rational.
- (6) Connect the circuit and check it.

Through the steps above, the diagram of hardware design is achieved. Proteus has more than 6000 kinds of component models and peripheral devices commonly using in Embedded system design. Those peripheral devices ,including bus drivers, LED, LCD modules, programmable timer, parallel interface, matrix keyboard, real time clock, power modules, multiple D / A and A / D converter etc, can be called directly. It is particularly suitable for single-chip system simulation and use. The main components library in PROTEUS is shown in Table 1.

Table 1. The Main Components Library in PROTEUS

Library name	Component type or series
74std	8 Libraries like AS, F, HC, HCT, LS, S and ALS of 74 series
Analog	power supply circuit, 555, D/A , A/D and so on
Bipolar	Triode like 2N, BX, MU, TIP, 2Tx series
Cmos	Cmos circuits
Device	Conventional components, such as resistors, capacitors, inductors
Diode	Zener diode like IN, 3EZ, BAZ, BZX, MMBA, MZD series
Ecl	ECL integrated circuit
Fairchld	Triode like 2N, J, MP, PN, U, TIS series
Fet	FET tube like 2N, 2SJ, 2SK, BF, BUK, IRF, UN series
Lintec	Operational amplifier like LF, LT, LTC, OP series
Memory	EPROM , EEPROM , RAM
Micro	Processor like 51 series, 6800 series, PIC16 series, z80 series
Natdac	A/D and D/A like LF, LM, MF series
Natoa	Operational amplifier like F, LM, LPC series

Opamp	Operational amplifier like A D, CA, EL, MC, NE, OPA , TL series
Pld	PLD integrated circuit like AM16, AM20, AM22, AM29 series
Teccor	Silicon control like 2N, EC, L, Q, S, T, TCR series
Texoac	Operational amplifier like LF, LM, LP, TL, TLC, TLE, TLV series
Values	Electron tube
Zetex	Triode, diode, varactor
I2cmems	24 series , fm24 series, m24 series, nm24 series
Resistors	Resistor unit
Capacito	Resistor unit
rs	
Display	Display device: Digital tube like 7seg series; Liquid crystal like LM, MD, PG series
Active	Conventional components and instruments
Asimmdl	Basic digital logic gate circuits
s	

3. The Building of Simulation Platform

In embedded systems teaching process, to great extent, the new theory teaching method can encourage students to think critically instead of learning things by rote. Meanwhile, students will learn things through understanding on the ground of this new theory teaching method. For that matter, in order to efficiently demonstrate this new theory, the design of the electronic stopwatch is regarded as a necessity. Creating this electronic stopwatch depends on two working processes which are the design process and the implementation process.

In the design process, the paper takes ARM7 [3] micro-controller to devise a stopwatch as an example to set forth the whole design and simulation process. The first step is to create a new project application in KEIL integration framework. μ Vision 3 requires the designer to choose a corresponding processing from Proteus for the project. So that, Philips LPC2138 [4] controller is introduced to Proteus. Furthermore, in the basis of the controller, four Nixie tubes are imported to show the current time. Meanwhile, there are two buttons that capture the timer. As a result, by connecting them with each other, designers can realize the hardware environment of a stopwatch. Through it, μ Vision 3 can automatically set corresponding tool options for the project, which is based on the simulated hardware environment in Proteus. Finally, it can totally execute software design and eventually realize the design of a stopwatch.

In the implementation process, the drive and display of Nixie tubes can be achieved through GPIO. There are four registers mainly used in the design, which are IOPIN, IOSET, IOCLR and IODIR. Each register has the respective functions. The IOPIN register can show the current status of high and low level of pin. The IOSET register can control the pin to output high level. The LOCLR register can control the pin to output low level. The IODIR register can individually control the direction of each IO port. Consequently, the GPIO pins to drive LED are set to output. Then the intelligent drive of Nixie tubes can be achieved through the IOSET and IOCLR registers controlled by the flow of the program. It sets input and output status of each port by IODIR. The IOCLR clears the original state of the corresponding port. The IOSET sets corresponding port to output high. So the specific port of processor can be set by it. Specific operations: At first, reading the current stopwatch time n. Then corresponding each single bit of time n with each Nixie tube through the IOSET and the IOCLR. This design uses four Nixie tubes. When the time n is more than 9999, time will return back to zero. The main function of the program is an infinite loop. The loop makes the LPC2138 keep doing two things continuously----- displaying the current time with four Nixie tubes and monitoring whether there is a timer capture.

The designing process of electronic stopwatch in Keil can be described as:

- (1) Define an integer n that represents the current time, and when n is more than 9999, it will return back to zero.
- (2) Define an array `number[]` with initial values from zero to nine that represent each bit of n .
- (3) Define an array `record[]` that record the time when there is a timer capture. And there is an integer `count` that represents the number of the value of the array `record[]`.
- (4) Define an integer `flag` that show whether the time is running. When `flag=1`, the time is running. Otherwise, when `flag=0`, the time is stopped.
- (5) Devise a function `delay()`, when executed, this function represents that time passed by a second.
- (6) Devise a function `Time_Init()` to realize the initiation of the timer.
- (7) Devise a function `show()` to correspond each single bit of n with each Nixie tube.
- (8) In the main function, execute the function `Time_Init()` firstly, and define two integer `temp1` and `temp2` to capture the timer.
- (9) Next, the program enters an infinite loop. When `flag=1`, and then there is a timer capture `T0CR0`, the stopwatch will be stopped and `flag=0`, meanwhile, the array `record[]` will record the current time. Then, if there is a timer capture `T0CR0` again, the stopwatch will be running. Or, if there is a timer capture `T0CR1`, the value of the array `record[]` will be displayed with Nixie tubes.
- (10) During this loop, once a timer capture is happened, the integer `flag` changes into the opposite value, which means that `flag` is changed from 1 to 0 or `flag` is changed from 0 to 1.
- (11) When `flag=1`, the time is running, and the value of the integer n will be displayed with Nixie tubes. When `flag=0`, the time is stopped and the value of the array `record[]` will be displayed with Nixie tubes. Moreover, when n is much more than 9999, it will return back to zero.
- (12) The display of Nixie tubes that aims to show the current time and recorded time depended on the two registers which are `IOSET` and `IOCLR`.

The encoding implementation of electronic stopwatch can be described as:

```
PINSEL0 := 0x00;           // set GPIO.
Function Time_Init ();     // initiate the timer.
While (1) do {            // infinite loop.
    Function delay();      // time passed by a second.
    If flag==1 then{
        n:=n+1;           // n represents the current time.
    }
    temp1:= T0CR0;        // there is a timer capture T0CR0.
    If temp1> temp2 then{  // this timer capture happened.
        If flag==1 then{
            flag:=0;      // time is stopped when there is a timer capture.
        }
        else{
            flag:=1;      // time is running when there is a timer capture.
        }
        temp2:=temp1;
        If flag==0 then{  // record the current time.
            record[count]:=n-1;
            count:=count+1;
        }
    }
}
If n>10000 then{         // when n>9999, n will return back to zero.
    IOCLR0:=0xFFFF<<7;
    IOSET0:= 0x0000<<7;
```

```
n:=0;
}
If flag==1 then{ // the stopwatch is running.
    Function show(n); // display the current time.
}
else{ // the stopwatch is stopped.
    temp1:=TOCR1; // there is a timer capture TOCR1.
    If temp1>temp2 then{ // this timer capture happened.
        Function show(record[]); // show each value in the array record[]
    }
}
}
```

Software design in the Keil as shown in Figure 2:

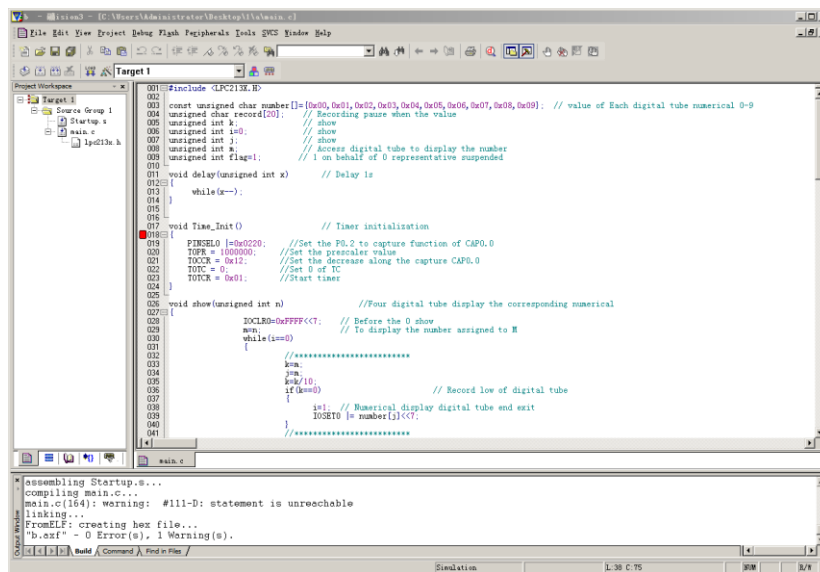


Figure 2. Keil IDE

Main steps of Proteus simulation as follow [5]:

At first, starting Proteus, creating a project and setting the paper size, project description;

Secondly, drawing components node, placing simulation components in the schematic editor window and connect the circuit. Through the above steps to complete the system schematic (Figure 2) drawing;

Then doing reasonable layout for components, configure POWER and GROUND module and connecting various components [6].

Finally, loading the compiled .hex object file into Proteus simulation circuit, in Proteus editing area, right-click the LPC2138 chip, selecting Edit Properties, find **. Hex file generated by KEIL software compiling, clicking the play or pause to put the system into operation status, conducting program simulation run. In hardware layout design, it selects P0.7 ~ P0.22 of LPC2138 [3] to connect four digital tube .The display of the display screen is controlled by their output signal. And P0.2 is connected with the first button. P0.4 is connected with the second button. These operations are done for the timer capture. Meanwhile, the first button is used to pause the stopwatch. The second button is used to display all paused value. As shown in Figure 3.

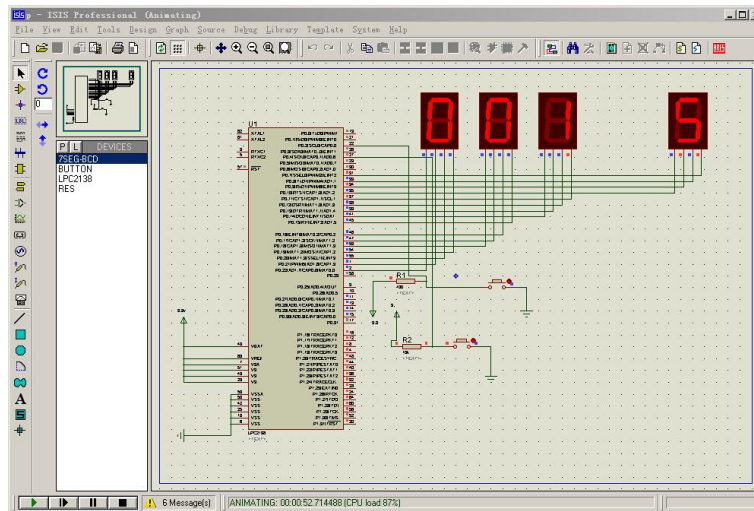


Figure 3. The Design of Hardware Circuit

4. Conclusion

This study has designed a kind of embedded system teaching platform based on Proteus and Keil. Through this platform, intelligent LED control simulation can be implemented. Besides, the simulation result is consistent with the experimental result produced by building realistic circuits. This method introduces EDA technology in teaching practice to implement the co-simulation test of embedded system program design. This method is not only a powerful complement to traditional physical circuit simulation, but also a new attempt. Practice has proved that this design method greatly reduces various work, such as circuit board production in process of the hardware circuit testing and the whole system debugging, the installation and welding of components and the adjustment and plug of components. Thereby development costs are reduced. However it should not and could not replace the traditional method. In order to train students to use basic theoretical knowledge and improve practical skills, we combine Proteus simulations with traditional experimental teaching organically. When students perform complex circuit design, they can use Proteus software to design. If designing scheme is validated viably, students build the physical circuit to debug.

ACKNOWLEDGEMENTS

This work has been supported by the Fundamental Research Funds for the Central Universities Nos. 2572014CB24 and China Undergraduate Scientific and Technological Innovation Nos. 201410225031.

References

- [1] F. Niu, W. G. Gao and C. X. Li, "Research on aviation managing simulation and performance estimation based on GNSS", GNSS world of china, vol. 32, no. 5, (2007), pp. 1-4.
- [2] I. T. Almkawi, M. G. Zapata, J. N. Al-Karaki and J. Morillo-Pozo, "Wireless multimedia sensor networks: current trends and future directions", Sensors, vol. 10, (2010), pp. 6662–6717.
- [3] "LPC2138 Datasheet", <http://www.NXP.com/>.
- [4] "FriendlyARM—ARM based Development Boards and Modules", (2012), <http://www.friendlyarm.net/>.
- [5] K. Gulati, J. F. Croix, S. P. Khatri and R. Shastry, "Fast circuit simulation on graphics processing units", in: Proceedings of the 2009 Asia and South Pacific Design Automation Conference, (2009), pp. 403–408.
- [6] M. R. Prasad, A. Biere and A. Gupta, "A Survey of recent advances in SAT-based formal verification", International Journal on Software Tools for Technology Transfer, vol. 7, no. 2, (2005), pp. 156–173.

Authors

Li Dan, Male, born in 1981, lecturer, working in Information and Computer Engineering College of Northeast Forestry University, mainly engaged in forestry informatization.