

Internet-based multiuser in-circuit emulator design for 8051 microprocessors

Kubilay TAŞDELEN^{1,*}, Akif KUTLU²

¹Department of Electrical and Electronics Engineering, Süleyman Demirel University, Isparta, Turkey

²Department of Computer Engineering, Süleyman Demirel University, Isparta, Turkey

Received: 15.10.2014

Accepted/Published Online: 02.06.2015

Final Version: 20.06.2016

Abstract: A real-time, Internet-based 8051 microprocessor in-circuit emulator (MicroLabICE) was designed and run for microprocessor lessons for undergraduate and graduate programs of electrical, electronics, communication, and computer engineering. Users can run applications developed for the 8051 microprocessor online on a physical circuit not being limited to a specific time or place with the 8051 microprocessor in-circuit emulator (ICE). The programs prepared by users are transferred to the controller area network (CAN) through an online server and program the ICE experiment module. Codes loaded on the designed emulator experiment module are run step by step and values that 8051 microprocessor registers have taken could be monitored on the client interface real-timely. The client interface program, which enables users to control emulator experiment modules and server computer intercommunicating through CANBus, to connect to and to manage emulator experiment modules online, has been developed in C# programming language.

Key words: In circuit emulator, controller area network, 8051 Microprocessors, Internet-based control, embedded systems

1. Introduction

A microprocessor is a component that authorizes a series of components that check required transactions to interpret the commands in assembly language and runs its liable system according to the data collected at the end of this processing. A microcontroller is described as the component that is inserted in a chip in a way that makes it possible for input/output and memory units that operate with the microprocessor to be used together [1]. Nowadays processors, which are called microcontrollers, are commonly used in many areas such as industry, medicine, and education [2] and the high performance usage of microcontrollers in embedded systems has been developing rapidly [3]. Tasks that are carried out using external hardware components can be carried out with microcontroller control software. Thus, without changing the hardware, only software changes offer the opportunity of different operations on the same device [4]. Microcontrollers are the focus of embedded system designs [5]. An embedded system is defined as a combination of devices used to help monitor or control the operation of a device and mechanism [6], as well as hardware, software, and mechanical components that are designed to carry out a specific task [5,7]. Embedded systems are used in household items like washing machines and video recorders, consumer products like car electronics and telecommunication systems, and independent robots as in aircraft, railway, and industrial process control [8]. Today, the constant increase in design complexity has led to functional testing in running integrated circuits (ICs) with very special

*Correspondence: kubilaytasdelen@sdu.edu.tr

applications. Generally two test techniques are used: emulation and simulation [9]. In this study, a multiuser, web-based 8051 microprocessor emulator is designed for emulation, which is a widespread test technique.

The first aim of this study is to enable users to send the program of the microcontroller to the designed microprocessor emulator experiment module, to execute the assembler codes step by step, and to monitor online the register values of microprocessors that were attained after running the codes. Its second aim is that instead of accessing emulator experiment modules with data acquisition cards, the emulator installs the experiment modules on the local industrial network, and it increases the number of microprocessor emulator experiment modules, which enable synchronous connection. As the third aim, it provides an opportunity for users to monitor the software applications developed by another user synchronously. The fourth aim is to provide simultaneous Internet access to the microprocessor emulator experiment module. Moreover, it may give different users the chance to test software applications that belong to them on the emulator experiment modules without influencing or preventing one another synchronously.

The 8051 microprocessor emulator, which has been run, is composed of two main parts: software and hardware. In the software part, to develop server and client programs C# programming language, to store the information about 8051 microprocessor emulator SQL Server database program, and to develop the software that is activated on the 8051 microprocessor *mu* Vision program developed by the +Keil Company have been used. In the hardware part, PCICan-D card that is used to control the CANBus and on the emulator modules an 8051-based AT89C51CC01 integrated circuit developed by the Atmel Company exist.

2. Emulators

Recently microcontrollers and microprocessors have been getting more complex and the complexity is inclined to increase [10]. Since there are millions of transistors on each chip, complex chip systems have appeared. Functional verification of such systems has been problematic [11]. Nowadays, the constant increase in the complexity of designs has led to functional testing to run application specific integrated circuits [9]. Software and hardware devices such as simulators, ROM monitors, and emulators are used for functional testing. When these devices first appeared, their cost was extremely high [12].

In-circuit emulators are probably the most used monitors for embedded systems applications during their debug cycle [13]. For emulators used as a widespread method for debugging and functional verification, different descriptions have been used according to the tasks they carry out and their features from the day they first appeared onwards. Emulator is used as a general name for software and hardware that can be used to carry out the task of a system or device for it [9], a component of the development area for microprocessor or microcontroller based systems called target systems [14,15], and an important device to develop microprocessor based systems [16]. The introduction of emulators to embedded system designers is equal to the introduction of tractors to farmers: immediately life has become easier [17]. Emulators, like original microprocessors, keep their physical features and functionality while they provide more debugging and test mechanisms in the process of testing [13], debugging, and development of microprocessor based systems in terms of hardware and software to support designers. These mechanisms include step-by-step execution, breakpoint configuration and finding, tracing, monitoring, and arrangement of internal resources. Emulator performance, cost, power consumption, testing and debugging support, and hardware/software interface have become crucial matters in the microprocessor based platform design. Architecture platforms, hardware/software interfaces, and microprocessors whose operating methods are different change between emulators to a great extent. In addition, many emulators are registered trade products; extensive knowledge about their design is not accessible. Most of the accessible

information is in the form of application notes, which give only limited information on design and user manuals and so it is hard to compare the features of emulators [15].

The advantages of an emulator:

- It can work on its own without the target hardware.
- It allows reading and writing triggers.
- Real-time operation of a program can be monitored.
- It provides effective performance analysis.

The disadvantages of an emulator:

- High cost

3. Controller area network

CAN, which was developed by the Robert Bosch GmbH Company, is a serial data communications protocol [18–20]. CAN was invented at the beginning of the 1980s to meet the real-time transfer needs of the automobile industry and it was standardized as ISO 11898, ISO 11519 [20,21]. In 1985 after INTEL and Bosch GmbH became partner companies, communications protocol was regenerated, and the first silicon CAN chip that was applied on CAN protocol in 1989 could be used now [19]. CAN provides real-time performance thanks to its high reliability and low cost [22].

4. System design

This study consists of two parts including software and hardware. The hardware part consists of 8051, which will serve as an emulator, and CAN based circuits; the software part consists of client interface, database, server, and microprocessor. The block diagram study that was carried out in Figure 1 is shown.

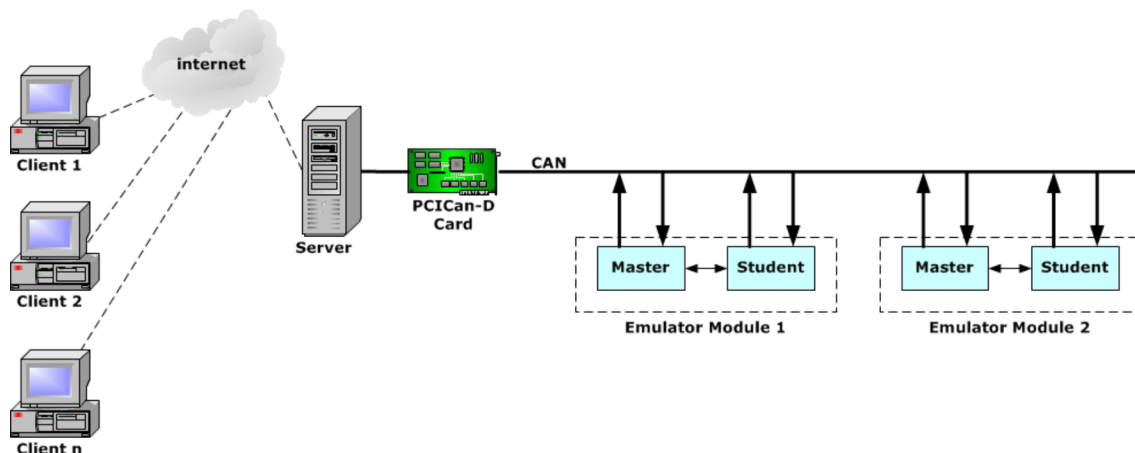


Figure 1. Block diagram of 8051 microprocessor emulator.

4.1. 8051 Microprocessor software

The most important component of the hardware part of the emulator experiment module, which consists of two emulator circuits, is microprocessor software, which is in the code memory of emulator circuits called master and client, providing meaning for electronic circuits. There are two kinds of microprocessor software that function differently in master and client circuits.

Assignments of master circuits:

- To migrate the client circuit to programming and executing modes.
- To send the “hex file” that was uploaded by the user to the client circuit.
- To send commands to the client circuit for the program to run step by step.
- To send the values that the registers of the 8051 microprocessor have taken as a result of step by step running to the server computer.

Assignments of client circuits:

- To execute the program installed in it with the step by step execution command of its master circuit.
- To send the values that the registers of its microprocessor have taken as a result of the running program to its master circuit.

There are two types of programs in the code memory of the integrated circuit AT89C51CC01 on the client circuit, the first of which is the application program that is sent by users through the client software to the code memory of the microprocessor. The second one is the emulator software that is kept in the code memory of the microprocessor in the circuit. The emulator software is responsible for sending the values that are constant on the code area to its master circuit and the registers of the microprocessor have taken as a result of step by step execution of the application program that is installed on the microprocessor.

The integrated circuit AT89C51CC01 has a code memory of 0000 and 7FFF address range 32 Kbytes. As one can see in Figure 2, for application program the 0000-6FFF address ranges of the code memory and for the emulator software 7000-7FFF address ranges have been separated.

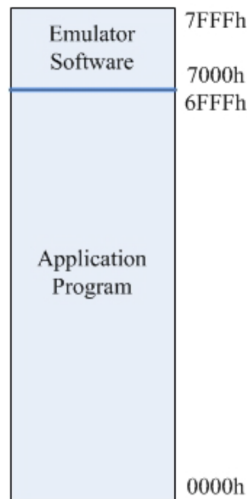


Figure 2. Memory organization.

4.2. Database software

Database software is generally responsible for storing experiment and user data of each emulator experiment module. The data stored in the database software include:

- The number of emulator experiment modules that will operate on the web-based 8051 microprocessor emulator system,
- User ID and password,
- The number of users that can be connected to modules,
- The time allotted for each emulator to the experiment administrator for completing the application,
- Opcodes and operands of commands that belong to the 8051 microprocessor, the names of commands, and the size in bites,
- The guest number of guests that are connected to the system according to their connection turns,
- Data on the user name of the guests that are connected to the system, the experiment number that they are connected to, the order in which they are connected to the module, whether they are administrators of the module.

4.3. Server software

It is the program that enables and controls connection between the hardware part of the emulator experiment module and the client interface. The server program operations:

- To connect to the experiment coming from the client program, to participate in an experiment, to carry out the necessary operations according to commands such as chat etc.,
- To delete or program the code memory of the microprocessor on the client circuit,
- To start the experiment coming from the client software, to reset the experiment, to read the registers of the microprocessors, to make operations such as leaving the experiment by evaluating the commands,
- To separate the hex code, which will be installed on the microprocessor, into its opcodes and operands, to determine the breakpoints.

In Figure 3, the flowchart of the server software is seen. As a result of the compilation of the code, which is numbered 1 by the 8051 microprocessor compiler, hex codes that have the number 2 supervene. These codes are retrieved by the server program, as seen in box 3, and are separated into their opcodes and operands. Thus, the server program determines where to put software breaks to enable step by step execution of codes. After breakpoints are embedded into the code, they are converted to their last form in box 4. The final version of the hex code is sent to the microprocessor.

4.4. Client software

The main task of the client software is to execute and monitor the emulator circuits through server software. While the client software carries out the mentioned tasks, it also performs the tasks required by the system in the background. These processes are necessary while the client software carries out its main task. The combination of program components that carry out these processes actually creates the client software. Processes that are carried out by the client software are listed. In Figure 4, the flowchart of the client software is seen.

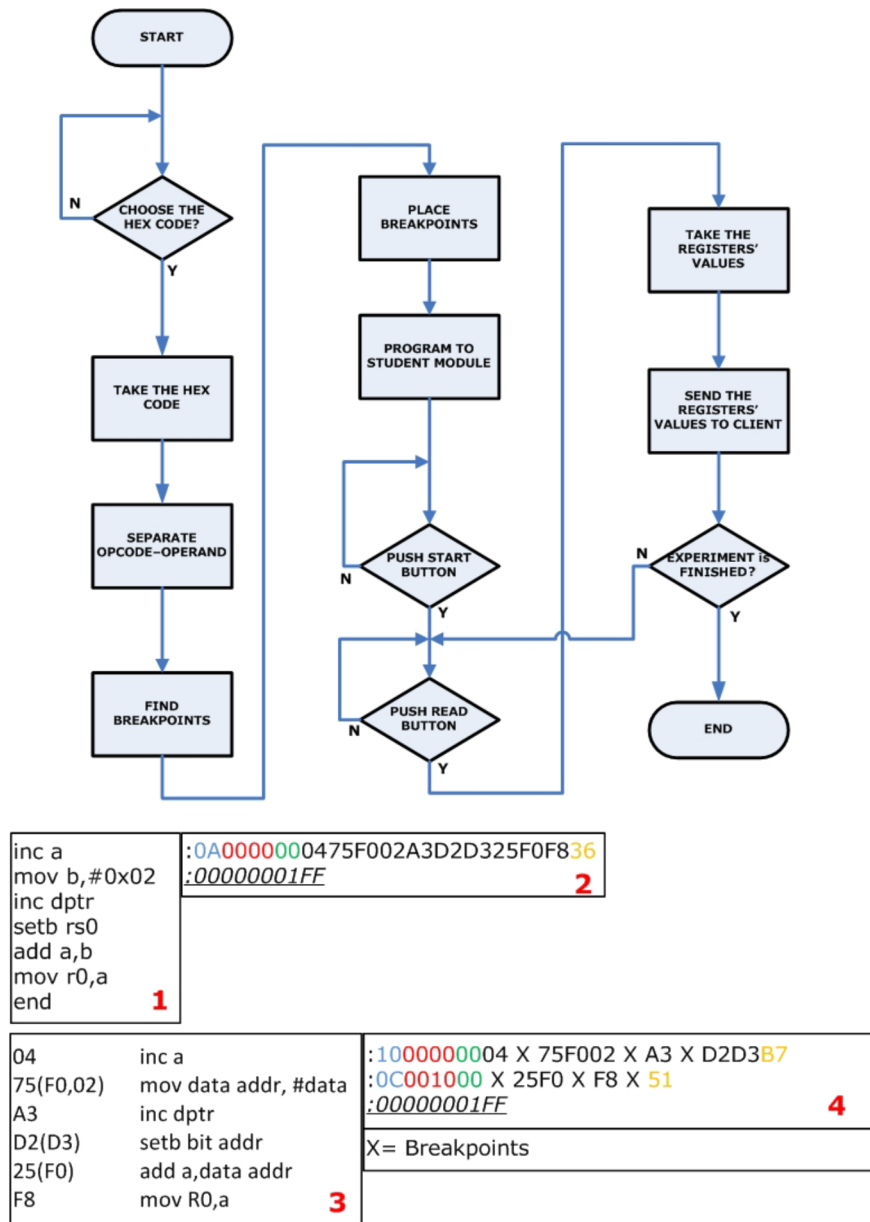


Figure 3. Flowchart of the server software.

- The connection of the client software to the server,
- The chat program, which enables the administrators and the audience to communicate with each other,
- Informing the user about the condition of the emulator experiment module,
- The experiment administrator sends the codes to program the microprocessor to start and finish the experiment, to reset the circuit, to start retrieving the registers, to pass the experiment administration, to leave the experiment,
- The display of the register values that belong to the microprocessor on the client interface, they are sent through the server from the emulator circuit.

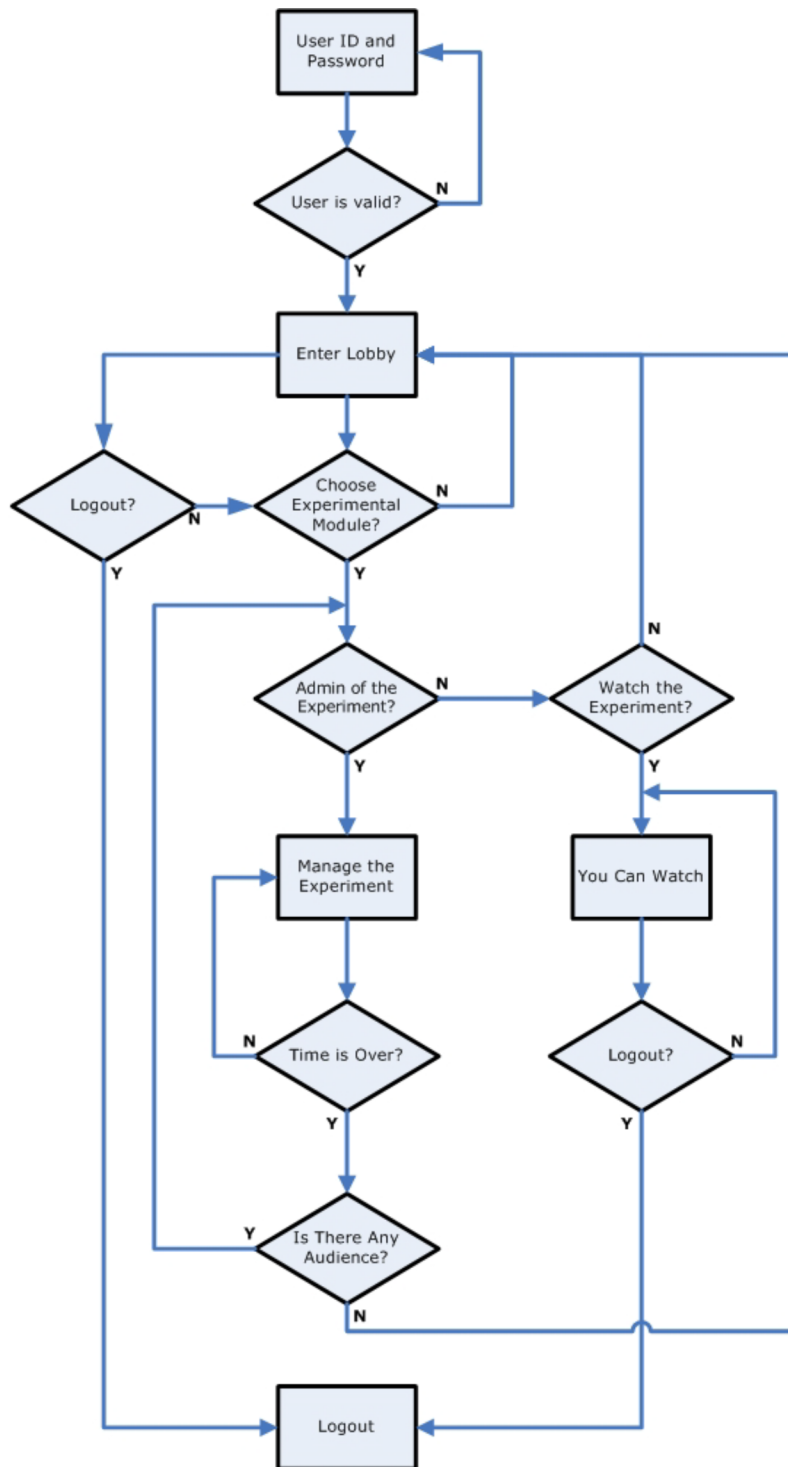


Figure 4. Flowchart of client software.

5. System operation

The system operates through the Internet; therefore, the user's computer must have an Internet connection. The users who have registered user names and passwords in the database can login to the system using the information. If they do not have usernames and passwords, they can login to the system as a guest without a user name or password, but the administrator must permit the guests to execute the emulators or to monitor the processes that a user has carried out on an emulator. The username information of the users who have logged in with their usernames and passwords will be displayed in the online users' part in the client software interface. On the other hand, the guests who take part in the experiment will be displayed by being numbered according to their login turns. The user or guest who has connected to the system from the login page is directed to the lobby page that can be seen in Figure 5 to select the emulator module to connect in the next step.

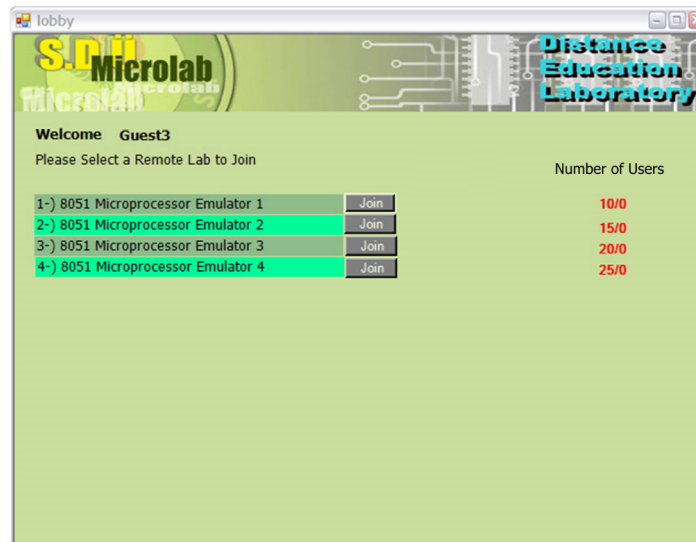


Figure 5. Lobby window.

Active emulator experiment modules and the number of users can be seen in the lobby window. The user can connect to the system by clicking the “Join” button that belongs to his preferred module. The client interface of the emulator experiment module that he has been connected to appears on the screen. In Figure 6, the client interface of an active user is seen. To explain the operation of the active client interface, it is divided into 6 groups as shown in Figure 6.

- The 1st group: In this group there are registers that belong to the 8051 microprocessor. These registers are ACC, B, PSW, SP, DPTR, R0 – R7; CY, AC, the general purpose F0, RS0 and RS1, OV, P bits of the 8 bits PSW register.
- The 2nd group: “Read” button allows the “hex” code that was uploaded on the microprocessor to operate step by step.
- The 3rd group: It shows the program that comes out as a result of the decoding of the “hex” code that was uploaded on the 8051 microprocessor and will be operated step by step.
- The 4th group: It shows the names of the users connected to the module.

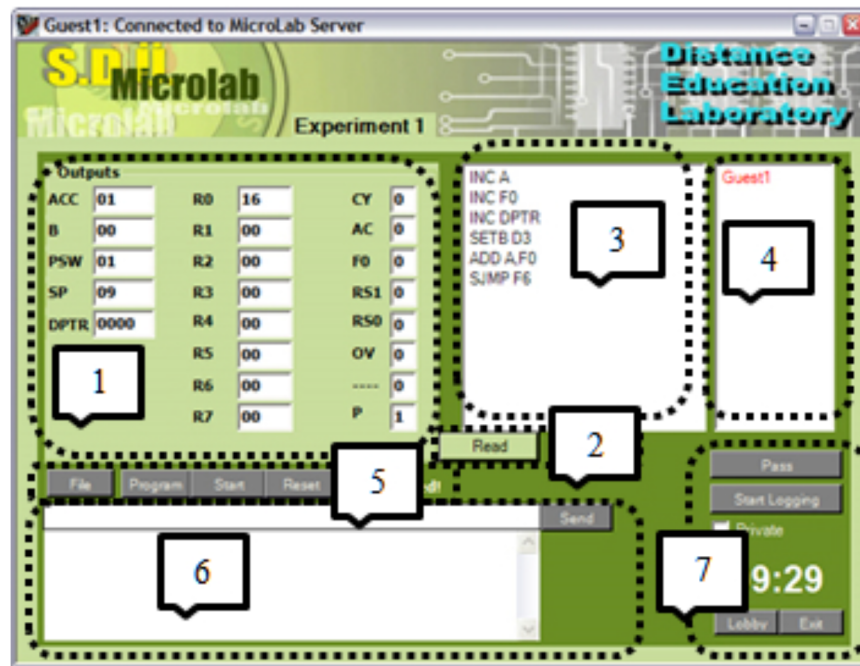


Figure 6. Client interface.

- The 5th group: There are necessary buttons to control the emulator module. “File” button, “Start” button to choose the “hex” extension file that will be uploaded on the microprocessor, and “Reset” button to display the values on the registers after the operation of the code are used to allow the code that was uploaded on the microprocessor to restart.
- The 6th group: Chat section of the module, which allows communication between connected users,
- The 7th group: “Pass” button, “Lobby” button is used to pass the administration to another user and to go to the lobby page on which emulator experiment modules are chosen, “Exit” button is used for the experiment exit.

The first user to connect to the emulator experiment module is always appointed as the administrator and there may be only one administrator in each module. The users connected to the module afterwards can only be audience. Since all controls of the client program interface are passive for audience, it is not possible to interfere in the module in any way. Only communication on the chat interface with other users who are connected on the same module is allowed. A certain amount of time is provided to the administrator of the emulator experiment module to enable his application. The remaining time to complete the application is seen on the client interface. The administrator must finish the application within the specified time period. If the application cannot be completed within this period, the administrator’s session will be closed. If there are other users on the module, the second user to be connected to the module will be assigned as the module admin. If the admin completes the application before the specified time period or leaves it unfinished, he can make another user the administrator. The new administrator who was made the module admin can either resume the application or begin a new application by installing his own code on the microprocessor. The administrator must install “hex” extension code of the application program that has been written for the 8051 microprocessor to be able to work on the emulator experiment module. Other file types are rejected by the client program.

6. Conclusion

In this study, an emulator experiment module, which could be executed online, was designed for 8051 microprocessors. This study displays not the ideal values like an 8051 simulator as a result of the execution of the codes installed on it, but real values since this study is an emulator design; this is the most important feature that distinguishes between emulator and simulators.

This system, which was designed as web-based, provides the opportunity for users to connect to the system from any computer with an Internet connection or anywhere 24/7. The most important disadvantages of emulators are their high cost and their sole existence in laboratories; these are major problems for researchers and students. The system has been put into the service of researchers and students who study with the 8051 microprocessor, and this has removed the restrictions like high cost and studying only in laboratories and laboratory hours.

CAN has been used to provide communication between the server and emulator modules in the Internet-based 8051 microprocessor. It is possible to connect more than one emulator experiment module to the system thanks to CAN, and this allows more than one user to work on different emulator experiment modules at the same time. In addition, the communication part was added to the system to enable other users, who monitor the application, to debate on the application while the user is working on the emulator experiment module.

The user, who runs an application on the emulator module, executes the application program that he has installed on the microprocessor step by step, and can monitor the values of the registers of the microprocessor as a result of the execution of each command. This feature allows the user to understand the program that he has developed, and to determine the points where the error has resulted as a result of faulty operation of the program.

The system designed was tested connecting two emulator modules on CAN. The operation of the system in which users were enabled to connect to the emulator experiment modules online as administrators or audience was controlled. All of the general properties of the Internet-based 8051 microprocessor emulator that was under scrutiny in this study can be found in Table 1.

Table 1. Properties of the Internet-based 8051 microprocessor emulator.

| | |
|----------------------------|---|
| Microprocessor | 8051 based AT89C51CC01 integrated circuit |
| Industrial network | Controller area network |
| Number of users | Multiuser |
| Number of emulator modules | 2 |
| Programming over Internet | Yes |
| CANbus speed | 500 Kbits/s |
| Code size for a user | 27 Kbyte |
| Real-time | Yes |
| Time | 7 days 24 hours |

Nowadays, many companies (Keil, Bascom, Phytion, Raisonance, Dunfield Development System etc.) have emulators produced for 8051 microprocessors. Commercial products do not have characteristics like enabling the working of more than one user and enabling online usage. Without pursuing a commercial goal, emulator types that came out as a result of scientific or patent studies are also available.

This study is also an example of an Internet-based laboratory. It has been thought that one of the most important shortcomings of distance education applications in electronic or computer environment is laboratory practice and it looms large in providing the opportunity to students to conduct real practice. In accordance

with the data, the Internet-based 8051 microprocessor emulator is an important device because it has main characteristics, for example, the system can be executed online, more than one emulator can be connected to the system using CAN, it can execute the codes step by step thus reducing the cost, it serves more than one user 24/7, and it helps one to understand the programming of the 8051 microprocessor.

When the studies in the literature are examined, features that an emulator must have are given in Figure 7. Since the junction point shows the common point of this study’s target features, it supplies explicit information about its original value. The Internet-based 8051 microprocessor emulator study is evaluated according to the features given in Figure 7 with commercial and scientific studies in the literature; the difference of the study is given in Table 2.

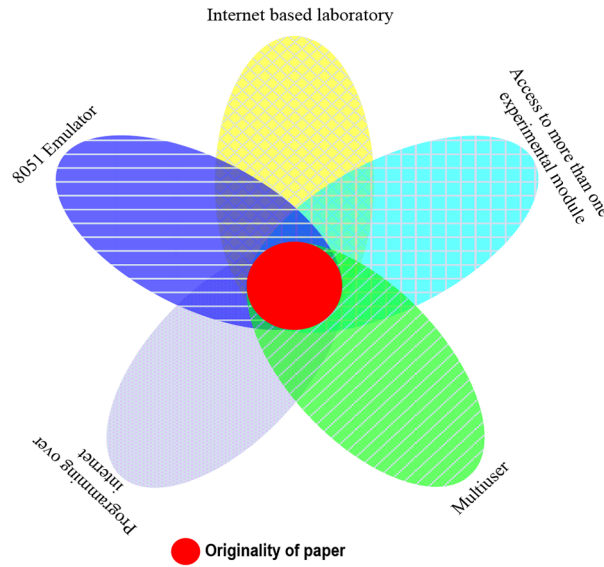


Figure 7. Requirements for the in-circuit emulator.

Table 2. Comparison of in-circuit emulators.

| References | 8051 Emulator | Internet-based laboratory | Access to more than one experimental module | Multiuser Multiuser | Programming over Internet |
|-------------|---------------|---------------------------|---|---------------------|---------------------------|
| [23] | | ✓ | ✓ | ✓ | ✓ |
| [24-27] | | ✓ | | | |
| [28,29] | | ✓ | ✓ | ✓ | |
| [30,31] | | ✓ | ✓ | ✓ | |
| [32] | | ✓ | | ✓ | ✓ |
| [33] | ✓ | | | | ✓ |
| MicroLABICE | ✓ | ✓ | ✓ | ✓ | ✓ |

All of the below issues were provided together in this study, and so this shows that the study has original value.

- Programming the 8051 microprocessor over the Internet,

- Access to simultaneous emulator experiment modules,
- Multiuser feature for each emulator experiment module,
- Executing the codes prepared for the emulator,
- Being able to execute the codes step by step,
- Displaying the microprocessor registers,
- Sharing the results with other persons online,
- Real-time data transfer through CAN serial industrial network communications protocol,
- Web interface design for the emulator system.

This study is thought to shed light as a reference on the following studies:

- Web-based emulator studies for PIC, ARM, AVR, Motorola 68000 based microprocessors,
- Examination of the effects of the emulator system on the sniff state using different industrial web protocols instead of CAN,
- Research on whether similar studies could be carried out not only for microprocessors, but also for other applied courses,
- Examination of its advantages and disadvantages compared to the traditional laboratories because of the fact that it is an Internet-based laboratory application,
- Performing a survey for the examination of either positive or negative skills about the training in the microprocessor lesson gained by students during the web-based 8051 microprocessor emulator study.

Acknowledgments

I would like to express my greatest gratitude to Süleyman Demirel University Scientific Research Projects Coordination Unit Board of Management, who supported this study with the project numbered 1632-D-08. I am also grateful to TÜBİTAK for their continuous support for this study with the project numbered 108E160.

References

- [1] Kaya A. Adapting microprocessor and microcontroller lesson to computer aided education. MSc, Sakarya University, Sakarya, Turkey, 2006.
- [2] Popa M, Popa AS, Cretu V, Micea M. Monitoring serial communications in microcontroller based embedded systems. *The International Conference on Computer Engineering and Systems*; 5–7 November 2006; Cairo, Egypt. pp. 56-61.
- [3] Macnamee C, Heffernan D. Emerging on-chip debugging techniques for real-time embedded systems. *Journal of Computing & Control Engineering* 2000; 11: 295-303.
- [4] Bay ÖF, Görgünoğlu, S. Design and implementation of an 8051 microcontroller educational set. *Journal of Polytechnic* 2002; 5: 195-207.
- [5] Okyay T. A control system design by using embedded systems. MSc, Ege University, İzmir, Turkey, 2006.
- [6] Kemp A. Embedded systems: the other problem. *Computers & Security* 1997; 16: 663-668.

- [7] Ovatman T, Brekling AW, Hansen MR. Cost analysis for embedded systems: experiments with priced timed automata. *Electronic Notes in Theoretical Computer Science* 2010; 238: 81-95.
- [8] Müllerburg M. Software intensive embedded systems. *Information and Software Technology* 1999; Vol. 41; pp. 979-984.
- [9] Ata Y. PC tabanlı sayısal entegre devre emülatörü tasarımı. MSc, Gebze Institute of Technology, Kocaeli, Turkey, 2000.
- [10] Melear C. Emulation techniques for microcontrollers. *Wescon/97 Conference Proceedings*; 4-6 November 1997; Santa Clara, California, USA: pp. 532-541.
- [11] Hassoun S, Kudluga M, Pryor D, Selvidge C. A transaction-based unified architecture for simulation and emulation. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems* 2005; 13: 278-287.
- [12] Martin S, Olive V. Embedded test environment. *7th IEEE International Workshop on Rapid System Prototyping*; 19-21 June 1996; Thessaloniki, Greece: pp. 50-54.
- [13] Jundi K, Moon D. Monitoring techniques for RISC embedded systems. *Proceedings of the IEEE 1993 National Aerospace and Electronics Conference*; 24-28 May 1993; New York, USA: pp. 542-550.
- [14] Huang IJ, Kao CF, Chen HM, Juan CN, Lu TA. A retargetable embedded in-circuit emulation module for microprocessors. *Design & Test of Computers* 2002; 19: 28-38.
- [15] Kao CF, Huang IJ, Chen HM. Hardware-software approaches to in-circuit emulation for embedded processors. *Design & Test of Computers* 2008; 25: 462-477.
- [16] Chen HM, Kao CF, Huang IJ. Analysis of hardware and software approaches to embedded in-circuit emulation of microprocessors. *Proceedings of the 7th Asia-Pacific Conference on Computer Systems Architecture*; January-February 2002; Melbourne, Australia: pp. 127-133.
- [17] Bannatyne R. Advances in debugging high performance embedded systems. *Wescon/97. Conference Proceedings*; 4-6 November 1997; Santa Clara, CA, USA: pp. 122-126.
- [18] Kutlu A. Wireless medium access control protocols for real-time industrial applications. PhD, The University of Sussex School of Engineering, Brighton, United Kingdom, 1997.
- [19] Gamiz J, Samitier J, Fuertes JM, Rubies O. Practical evaluation of messages latencies in CAN. *Proceedings of the 2003 IEEE Conference on Emerging Technologies and Factory Automation*; 16-19 September 2003; Lisbon, Portugal: pp. 185-192.
- [20] Ronnback S, Hyyppä K, Wernersson A. Remote CAN operations in MATLAB over the internet. *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*; 22-24 June 2004; Varna, Bulgaria: pp. 123-128.
- [21] Schickhuber G, McCarthy O. Distributed fieldbus and control network system. *Journal of Computing & Control Engineering* 1997; 8: 21-32.
- [22] Li F, Wang L, Liao C. CAN (Controller Area Network) bus communication system based on Matlab/Simulink. *4th International Conference on Wireless Communications, Networking and Mobile Computing*; 12-14 October 2008; Dalian, China: pp. 1-4.
- [23] Kutlu A. MicroLab: a web-based multi-user remote microcontroller laboratory for engineering education. *International Journal of Engineering Education* 2004; 20: 879-885.
- [24] Karadimas D, Efstathiou K. Design, implementation and evaluation of a remote laboratory system for electrical engineering courses. *Sixth International Conference on Advanced Learning Technologies*; 5-7 July 2006; Kerkrade, Netherlands: pp. 408-412.
- [25] Ko CC, Chen BM, Chen J, Zhuang Y, Chen TK. Development of a web-based laboratory for control experiments on a coupled tank apparatus. *IEEE Transaction on Education* 2001; 44: 76-86.
- [26] You S, Wang T, Eagleson R, Meng C, Zhang Q. A low-cost internet-based telerobotic system for access to remote laboratories. *Artificial Intelligence in Engineering* 2001; 15: 265-279.

- [27] Moros R, Luft F, Papp H. Virtual laboratory course in chemical engineering and unit operations (VIPRATECH) tutorials, simulations and remote process control. Proceedings of the International Conference on Computers in Education; 3–6 December 2002; Auckland, New Zealand: pp. 1447-1448.
- [28] Shen H, Shur M, Fjeldly T. Conducting remote instructor-controlled experimentation. Patent paper, No: US2003/0207243 A1, Int. Cl. G09B 7/00, 6 Nov 2003.
- [29] Bruker D. Remote laboratory. Patent paper, No: WO 01/09862 A1, Int Cl. G09B 3/00, 08 Feb 2001.
- [30] Alhalabi B, Hamza K, Remote laboratory experimentation. Patent paper, No: WO 02/086843 A1, Int Cl. G09B 7/00, 31 Oct 2002.
- [31] Chirico M, Scapolla AM, Bagnasco A. A new and open model to share laboratories on the internet. *IEEE Transaction on Instrumentation and Measurement* 2005; 54: 1111-1117.
- [32] Pastor J.S, Gonzalez I, Lopez J, Arribas F, Martinez J. A remote laboratory for debugging FPGA-based micro-processor prototypes. Proceedings of the IEEE International Conference on Advanced Learning Technologies; 30 August–1 September 2004; Joensuu, Finland: pp. 86-90.
- [33] Gomes L, Costa A. Remote laboratory support for an introductory microprocessor course. Proceedings of the 2005 IEEE International Conference on Microelectronic Systems Education; 12–13 June 2005; California, USA: pp. 21-22.