Malaysian Technical Universities Conference on Engineering and Techology (MUCET 2010)
Malacca, Malaysia, June 28-30, 2010

Designing of a Multi-Microcontroller Hardware Platform for Academic Environment: The UMP-EVT

R.S.K. Selvakumar[1], K.H. Ghazali[2], N.M.K. Nik Yusoff[3]

Abstract **-** In traditional Embedded Control Technology courses, students learn to develop assembly language programs to control peripherals, handle interrupts, and perform I/O operations. However, students find the subject are diffcult as the subject is presented in a lecture format. Unfortunetly, this Embedded Controller Technology (ECT) course is a compulsory course in any electrical or electronic field of engineering. For this reason, a new laboratory evaluation tool (UMP-EVT) specifically will be designed to be as a learning tool for those who intend to learn microcontroller and for use in the academic environment. By using this UMP-EVT, users are exposed to practical experience of the microcontroller and provide an easy path to learn this intelligent electronic device in short time. In this respect, this UMP-EVT would be applicable for education and expose the electrical engineering students to the understanding fundamental of microcontroller in electronic design field.

Keywords**:** Educational, MCS51 microcontroller, HC11 microcontroller, PIC microcontroller

Raja Saravana Kumar A/L Selvakumar is with Faculty of Electrical & Electronic Eng., Universiti Malaysia Pahang, Kuantan, Malaysia. (Phone: 6012-9631397; email: inderjit_1791@yahoo.com).

Kamarul Hawari bin Ghazali is with Faculty of Electrical & Electronic Eng., Universiti Malaysia Pahang, Kuantan, Malaysia. (Phone: 6017-7712224; email: kamarul@ump.edu.my).

Nik Mohd Kamil bin Nik Yusoff is with Faculty of Electrical & Electronic Eng., Universiti Malaysia Pahang, Kuantan, Malaysia. (Phone: 6017-9707986; email: nik@ump.edu.my).

I. INTRODUCTION

Embedded systems are designed for dedicated applications running in control systems. The unique feature of such systems is the capability to perform timely and predictable operations in response to concurrent requests arriving from the external environment [1]. To create an effective embedded system, one must properly employ the appropriate system architecture, hardware/software interfaces, peripheral devices, and software components [9]. Currently, embedded systems companies are facing with a shortage of engineers having the appropriate skills to respond to market opportunities [8]. However, most electrical and electronics engineering programs teach programming and design skills that are appropriate for a general-purpose computer operating under control of a commercial operating system rather than for the more specialized embedded systems [5].

Therefore, a flexible and versatile prototype system will be developed. The system can be used extensively in experiment or project for diploma, undergraduate and short courses. This UMP-EVT will also be boosted with a simple application board that is suitable for the students to test their capabilities and to improve their knowledge in this course. In addition, a monitor program will be developed to integrate the basic software such as communication software, text editor, cross assembler, compiler, simulator and emulator.

In this paper, we are going to propose such a systematic approach for hardware implementation and discuss how it could be used successfully in embedded system education of this UMP-EVT in ECT class. This paper presents the importance of microcontrollers course in education syllbus and the design of UMP-EVT that is based on 8-bit microcontroller Atmel® AT89C51, Microchip® PIC 16F877 and Freescale® MC68HC11. It begins with the discussion of the hardware implementation of the system. It proceeds then with software implementation of the system to be interfaced with the PC creating a user-friendly environment, followed by a discussion. Finally, a conclusion is given in section 5.

(ISBN: 978-983-2948-77-3)

II. Implementation of Hardware Section

A) Design Consideration for Student usage

Development tool reliability tends to be a problematic issue with microcontroller evaluation boards, especially in the hands of inexperienced students [30]. Dysfunctional hardware complicates laboratory activities and detracts from educational opportunities. Keeping microcontroller evaluation boards (EVB) functional often becomes a significant task for the lab faculty. One key design consideration for this educational training system was to evaluate past hardware issues, identify the student usage profile, and incorporate a reasonable degree of protection. Electrical failure is the most common fault mode in a lab environment. Three major sources of electrical failure were identified:

- Providing incorrect power supply to the processor including reverse polarity or the wrong voltage. Many EVBs don't contain an on-board power supply, so reverse polarity or excessive supply voltage will destroy the processor very quickly.
- Shorting output lines to power or ground resulting in excessive current. Most consumer products are designed such that no sequence of commands will result in hardware damage. EVBs are typically not so forgiving. Students may intend to provide a grounded input, for example, but inadvertently configure the port data control register as an output with logic high. The resultant condition will damage the output or the entire processor.
- Electrostatic discharge is the silent killer of CMOS devices. Students routinely connect wires to microcontroller pins, without properly grounding themselves.

Mechanical and electro-mechanical failure modes are less prevalent, but do occur. For example, damage may occur due to excessive board flex while mating wide ribbon cable connectors, or due to objects dropped on the circuit board (including conductive objects) while the board is operating.

The UMP-EVT key constraints are to design a board that didn't excessively limit external device interfacing, is cost effective, and robust relative to the student usage profile. Several compromises are necessary. The goal was never a "Student-proof" board, but rather a board that would handle the majority of expected student usage profiles. A failure-mode effects analysis (FMEA) approach will be utilized where severity and probability of occurrence will be ranked against cost of mitigation.

A major distinction between the UMP-EVT and most other EVBs is circuit protection. All user I/O will be interfaced through carefully segmented and marked I/O pins. Each I/O pin will be contained over-current and ESD protection, as well as an LED state indicator to eliminate the need for logic probes. The UMP-EVT I/O circuitry will be designed such that a short circuit between any user pin and any power-source from –15V to +15V will not result in damage to the EVT, even if the pin is configured as an output. Theoretically, the protection circuitry limits the drive strength and bandwidth of the peripheral interface, but the board is capable of maintaining CMOS logic levels at about 1mA, and with a 5 kHz bandwidth. The circuit protection will preclude the use of bas-based interface experiments, but permits most sensor and actuator experiments with little risk to the hardware.

The UMP-EVT also will incorporate with 5-volt power supplies. The supply will be thermally protected, and the on-board Fan, will be used as an I/O experiment for PWM speed control defaults to the on state and is strategically located to provide power supply cooling for additional design margin. Power for user experiments will be provided via a separate tracking regulator. The regulator is capable of supplying 1.5 A with less than 10mV of error with respect to 5.000V [21], but is isolated such that excessive voltage inadvertently applied to the user circuitry does not back-feed the processor and cause damage. The design will be included provisions for a clear plastic cover to provide mechanical protection, and the circuit board will be fastened to a rigid 1/8" thick fibreglass panel to prevent flexing. The boards may be stacked without mechanical damage.

III. Implementation of Software Section

A) Assembly Language Programming

Our department has so far retained our assembly language programming requirement. However, it was recently revamped for two fundamental reasons. The first reason is hardware suitability and availability. One of the most common computing platforms used at many universities is a basic PC running a recent Windows operating system [20]. The complexity of these processors makes it an ideal target of study for a computer architecture class. However, this same complexity makes it unsuitable for a beginning course in assembly language programming. To complicate matters, modern operating systems work against the student who needs to see the direct relationship between assembly language commands and hardware response. Operating systems present a highly sophisticated interface to the hardware that is only with difficulty breached by assembly language practitioners [20]. Interrupt processing, once a major

(ISBN: 978-983-2948-77-3)

topic in assembly language classes, is not as easy to implement under the watchful and restrictive operating systems of today. To address these concerns, we felt we should explore the suitability of other computing platforms.

A perceived lack of relevance of assembly language skills was the second reason for the changes to our educational training systems. Although a very visible part of the computer science world is engaged in advancing the state of the art in high-level software design and user-friendly interfaces, another, less-visible part of the industry, is busy developing software for a subset of microprocessors targeting embedded applications [26]. Although C is a popular language for development in many of these microcontrollers, assembly language is still an important tool. Assembly language provides the most natural and direct access to the processor's capabilities. By introducing assembly language programming in such an environment, students can see the relevance of this language while gaining an appreciation for the details of program execution at the instruction set architecture level.

The third reason for the change is related to another curricular decision in our department, the replacement of assembly language programming by C programming in our introductory courses. Although assembly language programming was used as a medium to teach fundamental programming concepts, it also offered the opportunity (and need) to discuss low-level details such as number representation (which in assembly is hardware dependent) and storage allocation (pointers and pointer arithmetic, static, dynamic and automatic allocation, and details of array storage). C programming hides many of these details from the programmer so there is less motivation to discuss what is happening at the processor level when using this language [13]. In our curriculum revision, the assembly language programming, which is usually taken after our students have experienced the basic fundamental data structures (arrays, stacks, queues, and linked structures), was a natural place to introduce the instruction set architecture of a real processor and demonstrate how it could be programmed using assembly language [4].

B) Hardware-Software Graphical User Interface (GUI)

The software development of the system will be divided into two categories; the monitor program and the Integrated Development Environment (IDE) parts. The monitor program will be developed by using the assembly language to allow communication between the PC and the system module. From the PC, the user can communicate with the system board to issue commands to upload or download, execute and read or write to memory location.

The concept diagram for the monitor program is shown in the Figure 1. This concept diagram is the main structure of the monitor program. It consists of several subroutines which each performs different tasks. The basic function of the monitor software is to read the input based on user's selection. Based on the selection, it will determine which procedure will be executed. In this section, several subroutines are developed, such as read keyboard, read string, display character, display string and many more [21].

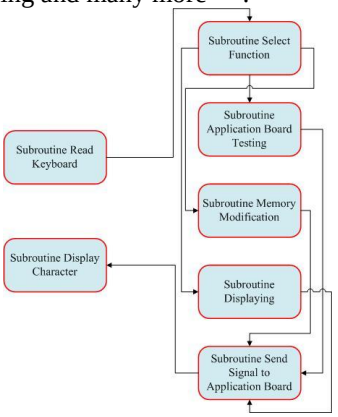

Fig 1: Concept Diagram of Monitor Software

The IDE, on the other hand, integrates various basic software such as cross assembler, communication software and text editor to create user-friendly environment. This new software tool will be developed to allow the user to perform all development activities without needing to exit any programs. This environment tool is developed by using an object-oriented programming Microsoft Visual Studio. The software will offers standard windows button such as command buttons, check boxes, option buttons, text boxes, and etc and produces a user friendly environment system.

IV. Discussion

A through free run test on the UMP-EVT system will be performed to ensure system reliability. Likewise, a monitor program will be developed, assembled, and downloaded into the EEPROM or NVRAM. Several tests of mini operating system will be tested several times and prove that the system is reliable and sufficiently stable. It will show that the monitor program is able to communicate with the computer and capable of performing several commands issued from the host.

A) Survey Result

A survey on student's understanding in embedded systems was conducted to detect the level of understanding in this course and the feedback from the students if there are develop in-

(ISBN: 978-983-2948-77-3)

house microcontroller educational development system based on the MCS51, HC11, and PIC 16 series microcontroller were developed. So far only 21 responses from the students in FKEE who replied the surveys and ppreliminary results are encouraging. The majority of the students found the UMP-EVT system will helpful in understanding the knowledge in embedded system. The UMP-EVT system will also helped them to better understand how the microcontrollers worked and how hardware and software worked together if the UMP-EVT are build. The instructors teaching this lab noticed that the need of this UMP-EVT system could make a significant improvement in students' performance and understanding of the lab. For more detail on this UMP-EVT system's survey, please log on into this webpage, "http://www.kwiksurveys.com/online-survey.php?surveyID=BIEIM_5d8a6dc9".

V. CONCLUSION

A prototype of UMP-EVT system will be designed and developed as teaching and learning tools and the system can be used extensively in experiment or project for diploma, undergraduate and short courses. The UMP-EVT system will provide a simple application board that is suitable for the students to test their capabilities and to improve their knowledge in this course. A mini operating UMP-EVT system based on IDE concept will be developed to integrate the basic software such as communication software, text editor, cross assembler and compiler.

REFERENCES

[1] Benjamin, M., Kaeli, D., And Platcow, R. 2006. Experiences with the Blackfin Architecture in an Embedded Systems Lab. Wcae '06

[2] Butler, J. And Brockman, J. Web-Based Learning Tools on Microprocessor Fundamentals for a First-Year Engineering Course

[3] Chandana Prasad et al., Computer System Organization and Architecture. Pearson Prentice Hall, Selangor, 2003.

[4] Celoxica. 2006. Dk Design Suite. Http://Www.Celoxica.Com/Products/Dk/Default.Asp

[5] Cottrell, S. And F. Vahid. A Logic Enabling Configuration by Non-Experts in Sensor Networks. Hfc. 2005.

[6] (Datasheet) 8-bit Microcontroller with 8K Bytes In-System Programmable Flash, Atmel Corporation, 2007, pp. 1-376, [Online]. Available: http://www.atmel.com

[7] Ecos. Http://Ecos.Sourceware.Org/

[8] F. Pardo, J.A. Boluda and E. de Ves, "Development Board for the Microcontroller Lab", Proc TELEC'2002, Santiago de Cuba, Cuba, July 2002, [online]. Available: http://tapec.uv.es/papers_en.html

[9] Halit Eren (2004)."Electronic Portable Instruments: Design and Applications." CRC Press United

[10] Halit Eren (2004)."Electronic Portable Instruments handbook." 3rd Ed.Mc. Graw Hill, US.32-37

[11] Hapsim. Http://Www.Helmix.At/Hapsim

[12] Hennessy, J. And Patterson, D. Computer Architecture – A Quantitative Approach. Morgan Kaufman Publishers. 3rd Edition. 1996

[13] Hodge, H. Hinton, H.S, and Lightner, M. Virtual Circuit Laboratory. Asee. American Society for Engineering Education. 2000

[14] I.S. Mackenzie, The 8051 Microcontroller, 3rd Ed., New Jersey, NY: Prentice Hall, 1999.

[15] Images Scientific Instruments. Http://Imagesco.Com

[16] J. W. Carter, Microprocessor Architecture and Microprogramming - A State Machine Approach. Prentice-Hall, Englewood Cliffs, NJ, 1996.

[17] Lcc. Http://Www.Cs.Princeton.Edu/Software/Lcc/

[18] Levis, P. And Culler, D. 2002. Maté: A Tiny Virtual Machine for Sensor Networks. Sigops Oper. Syst. Rev. 36, 5 (Dec. 2002), 85-95.

[19] M. Gunes, M.A. Thornton, F. Kocan, S.A. Szygenda, A survey and comparison of digital logic simulators, Circuits and Systems, 2005. 48th Midwest Symposium on, 2005, pp. 744-749 Vol. 741.

[20] N.M.K. Nik Yusoff, (2006 Jan), DEE3263 Microcontroller Technology – Student Performance Report, Universiti Malaysia Pahang, Malaysia.

[21] R.S.K Selvakumar, Intel Microcontroller 8051 Educational Development Board System: User Manual, UMP, 2008

[22] Ricks, K. G., Jackson, D. J., and Stapleton, W. A. 2005. An Evaluation of the Vme Architecture for Use in Embedded Systems Education. Sigbed Rev. 2, 4 (Oct. 2005), 63-69.

[23] Smith, J. And Nair, R. Virtual Machines: Versatile Platforms for Systems and Processes. Morgan-Kaufman Publishers. 2005.

[24] Stark, R., Schmid, J, and Borger, E. Java and the Virtual Machine- Definition, Verifications, and Validation. 2001.

[25] Virdes Development System. Http://Avoron.Com/Index.Php

[26] W. H. Gothmann, Digital Electronics - An Introduction to Theory and Practice. Prentice-Hall, Englewood Cliffs, NJ, USA, 1977.

[27] VMware. Http://Www.Vmware.Com/

[28] Windriver Systems. Http://Www.Windriver.Com/

[29] Y.Li, "Teaching Embedded System using a Modular-approach Microcontroller Training Kit", World Transaction on Engineering and Technology Education, Vol 6, No 1, 2007

(ISBN: 978-983-2948-77-3)