SOUTH-WEST UNIVERSITY "NEOFIT RILSKY"
BLAGOEVGRAD, BULGARIA

VOLUME 5
2007

SCIENTIFIC Research

ISSN 1312-7535

ELECTRONIC ISSUE

Combined application of Assembler and C languages in the Education of Microcontrollers

L. Taneva, V. Hristov*
* South- West University- Bulgaria, 66 Ivan Mihajlov Street, Blagoevgrad, Bulgaria, Phone (+359)738889132; E-mail: v_hristov@aix.swu.bg

Abstract. *The educational system has to support a way of learning, combining theoretical and practical approaches. The investigations prove that active learning, using development tools is an effective way of receiving knowledge. The hardware platform and the integrated software environment are good teaching tools and the students find the course motivating. The hardware platform as well as the integrated software environment and Combined application of Assembler and C languages offer knowledge that cannot be learned in classical lectures.*

The aim of the present paper is to describe the planning and conducting of an experiment within the students' groups educated in Microcontrollers and Microprocessors (MCU/MP) architecture and application. The purpose of this experiment is to analyze the integrated development environments (IDE) for software development and to study the balance in programming with Assembler and C. Evaluation criteria are defined for theoretical, practical skills and expected results. The obtained results were analyzed.

1. Introduction.

The choice of program language is based on the complex task and level of students' programming skills. The existence of Integrated Development Environments with embedded Assembler, C-compiler, In–Circuit Simulators and Debuggers makes possible the development of software with program languages on low and high level. The low-level language Assembler presents the binary code with mnemonic or symbolic code. The relation between user and program requires a good knowledge of the system structure - working with registers, buses and flags. Assembler is a language useful for an effective program realization and hardware optimization. It translates the mnemonic source code into a realizable binary code, defines the syntactic errors and shows them before the stage of execution. The Assembler advantages are: fast execution of the source code, small space of required program memory, possibilities and knowledge that increase the program technique of high level languages. A disadvantage of the Assembler is the required longer time for its learning and developing the program.

The high level C/C++/Embedded C gives an opportunity to use powerful instructions for the program algorithm and decreases the development time. Programming is usually ten times faster than programming in Assembler, especially when the code is long. The disadvantage by using C is that the compiler is not effective – it translates one instruction in a great number of binary instructions. The compiler does not optimize the use of registers and transfers to most data between them. The compiler generates two to five time more instructions than a well-written program in Assembler. It requires more memory and the execution of the program code is two to five times slower.

This paper describes the main results from a test carried out at school and university with two groups of students. The aim is to compare how different teaching approaches exert influence on the students' learning process and to find the balance in using pure C and C plus Assembler in learning MCU. The comparison between the two groups of different age (18 yr. and 21 yr.) is possible because the level of memorizing information is almost equivalent in both groups (fig. 5).

The development of a test on Microprocessors/Microcontrollers is not an easy task because of the fact that the theoretical knowledge has to be estimated on the one hand and the practical skills on the other hand. There are different strategies of education with microcontrollers in the different universities. The difference is not only in the lessons but also of the curricula in general. For example, in some schools learning MCU is named "Organization of Electronic Machines" and education concerns the hardware and the instruction set of Assembler. The students do not write a real source code in Assembler – this code is for the compiler. In other schools the object is named "Organization of PC and programming in Assembler" and hardware and software aspects of the electronic device control are combined. Others learn "Computer Architectures" and so on.

The analysis of different university curricula shows the existence of a great number of approaches in teaching MCU. Their evaluation is a difficult task and requires an estimation of the theoretical knowledge after the course. The test is a standard way to do this. Another task is the estimation of practical skills – they could be evaluated by developing programs for a concrete application and estimated according to the way and the time for solving the problem, the complexity of the problem, the use of program language, etc.

2. An experimental teaching on MCU.

During 2005/2006 the students of the 11th class of the Technological School "Electronic Systems" were taught on MCU with Assembler and C programming languages. At the same time the 3rd year students of the Southwest University "Neofit Rilski" – Blagoevgrad were taught on MCU with C language only. The purpose of the experiment is to evaluate the theoretical knowledge of the two groups by test after ending the course. The expected result is that the level of knowledge of the group learning MCU with Assembler and C will be better. Generally the experiment included 38 schoolboys and schoolgirls from the high school (age 18) and 34 students (age 21). The lecturers were the same people.

Two ways are used in the study of MCU: direct experimentation with laboratory module SPS430 based on microcontroller MSP430F149 (TI) and simulation of the work of MCU and the peripheral modules embedded in MCU using IDE Embedded Workbench (IAR Systems). The module SPS430 is developed for educational purposes and there are keyboard LEDs, buzzer, dynamic indication, potentiometer, etc. The structure of methods used in the education on MCU is shown in fig.1.

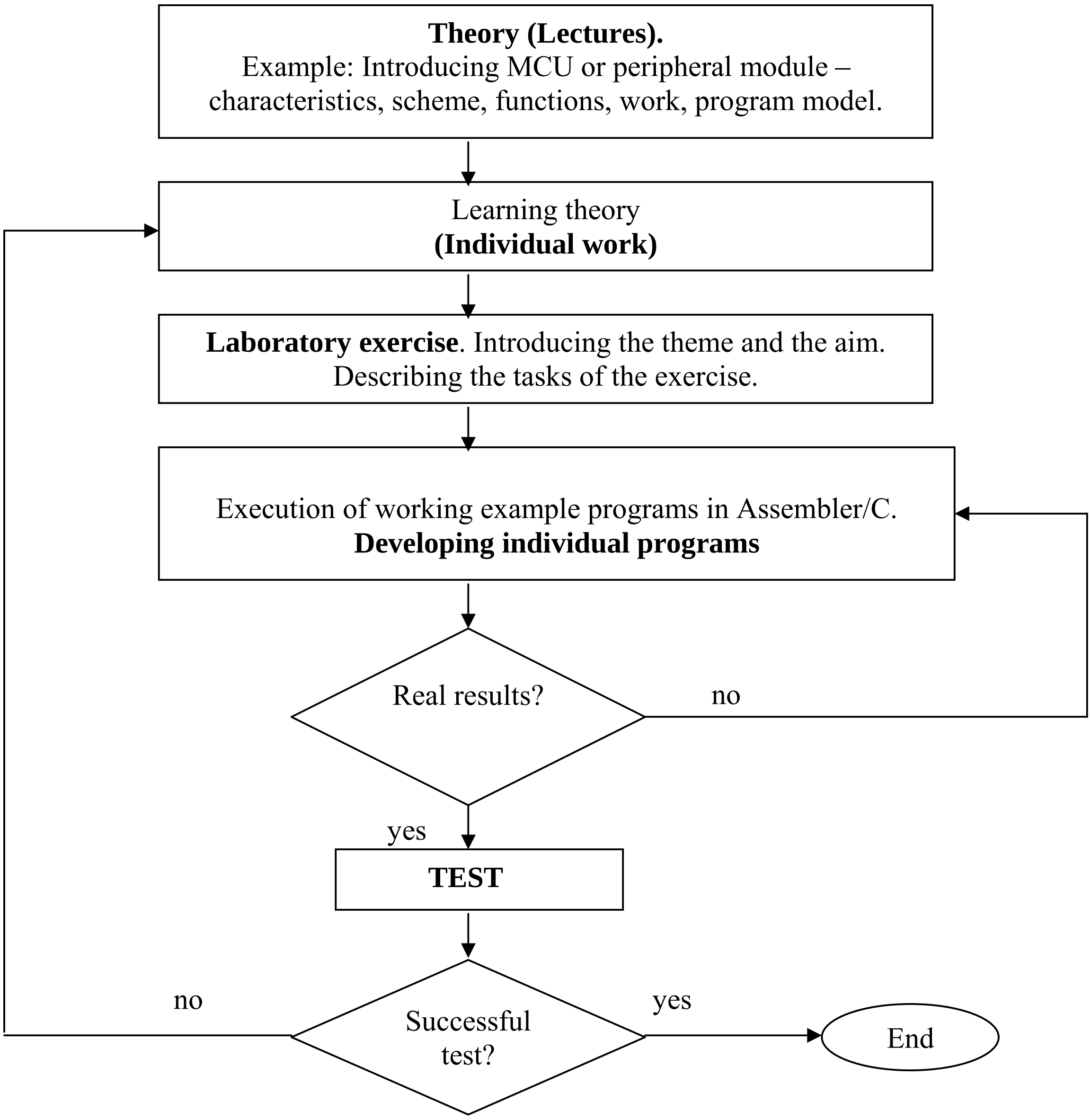

Fig.1. Methods of education on MCU.

The execution of all laboratory experiments is in accordance with specially developed didactic materials for practical work in a laboratory environment (fig.2). The description of modules, the working rules and example programs are given. It is important to note that this experience should be understood as a new way to learn and not as a final proposal. The techniques used in this study to measure the learning results or, in other terms, the knowledge that the student acquires in a certain period of time are a test. This test consists of ten questions and three possible answers to choose from them. In each group of answers there is one correct answer.

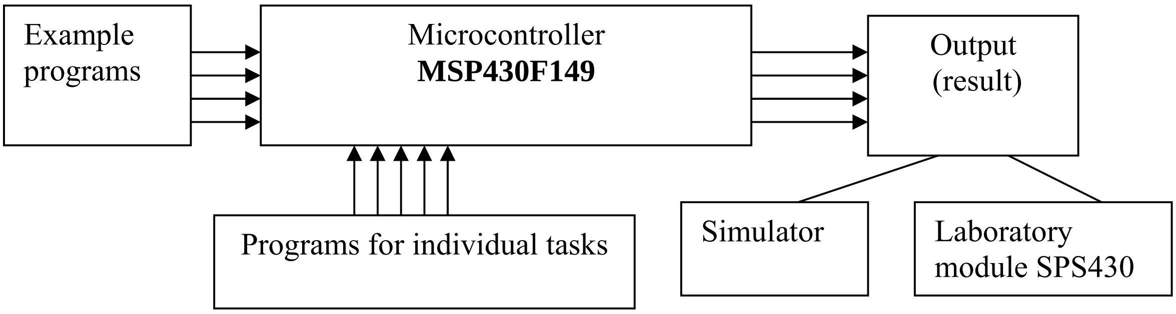

Fig.2. Education on microcontrollers.

2. Results from different teaching strategies.

The dispersion analysis made on the test results from the examination on MCU is given here. The two groups are estimated using an original ten-grade scale of 1 (minimum) to 10 (maximum) assessments.

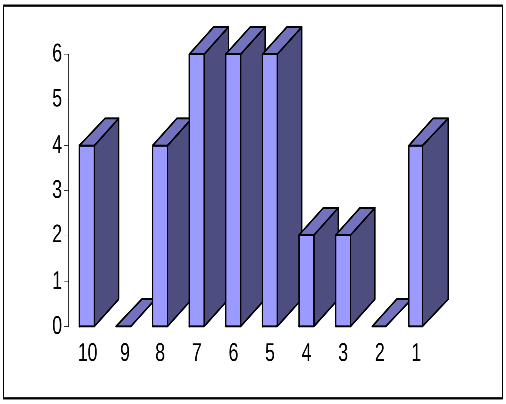

Fig. 3 Histogram of test results of the first group (students).

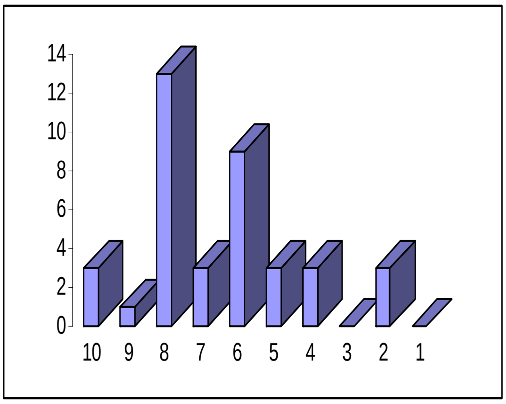

Fig. 4 Histogram of test results of the second group (schoolboys).

Two extracts are made for each group and the received results are given in histograms (fig.3, fig.4). The course of the dispersion analysis is shown in Table 1. The arithmetic mean of the different extracts (groups) and the general arithmetic mean are calculated by formula 3:

$$\bar{x} = \frac{\sum x}{n} \qquad (3)$$

The deviation calculation between the groups is given in Table 1. The general deviation: $\sum (x - \bar{x})^2 =$ 388.9861 is calculated by a consecutive subtraction of the general arithmetic mean mark $\bar{x} =$ 6.2361 from the examination mark of every student and the obtained results are squared.

Table1. The course of dispersion analysis.

Year 2005/2006	Number of students (f_i)	Sum of points from the test (x)	Arithmetic mean of the different groups ($\bar{x}_i$)	$\bar{x}_i - \bar{x}$	$(\bar{x}_i - \bar{x})^2$	$(\bar{x}_i - \bar{x})^2 f_i$
TUES	38	251	6.6053	0.3692	0.1363	5.1784
SWU	34	198	5.8235	-0.4126	0.1702	5.7876
	72		6.2361			10.966

The deviation between the groups or, the sum of squares from the differences between the examination marks of individual students in the groups and the group arithmetic means, is found by the general and between the groups deviations that are already calculated

(4) $\sum(\bar{x} - x_i)^2 = \sum(x - \bar{x})^2 - \sum(x_i - x)^2 f_i$ =388.986 - 10.966 =378.02

The variation degrees of freedom between the groups are k – 1 = 1 since the groups are two in general and inside the groups n – k = 70.

The two dispersion estimations are found on the basis of deviations between the groups and inside the groups:

$$(5) \quad \sigma_M^2 = \frac{\sum(\bar{x}_i - x)^2 f_i}{k-1} = 10.966$$

$$(6) \quad \sigma_B^2 = \frac{\sum(x - \bar{x}_i)^2}{n-k} = 378.02 / 70 = 5.4003$$

The relation between the two assessments of the dispersion is:

$$(7) \quad F = \frac{\sigma_M^2}{\sigma_B^2} = 2.031$$

To verify whether the combined application of languages C and Assembler in the respective laboratory exercises considerably exerts influence on the quality of education, a level of importance 0.01 is chosen. The Table for F distribution at level of importance 0.01shows that at degrees of freedom k – 1 = 1 и n – k = 70 the critical value of F is 7.011 (see the function in Microsoft Excel FINV (0.01; 1;70)=7.011).

Out of that the empiric characteristics (obtained on the basis of the two dispersion estimations), F = 2.031 is smaller than the critical value (F=7.011) at level of significance 0.01, it follows that data tests *do not give grounds to consider that the knowledge and use of Assembler language influences significantly on the acquired knowledge and skills of MPT.*

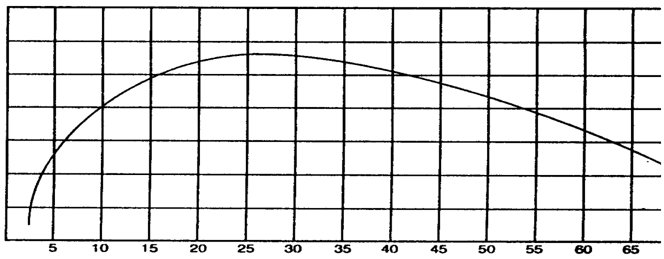

Fig. 5. Level of memorizing information in the different age groups.

CONCLUSIONS

The education with Assembler and C languages did not contribute significantly to the improvement in the students' learning process in comparison with C language education. On the other hand, the education with Assembler plus C language gave generally the best results: students seem to reach a better level of knowledge, ability and practice with microcontrollers. A number of conclusions can be drawn: the choice of program language has to be made after estimating the level of C-language knowledge and the program complexity.

- If the level of programming skills is not high and the program is too complex, using Assembler is irrelevant. In this case the program can be written very fast with high level language and debugging is very easy for beginners. The high level languages are good for writing programs by programmers without practice, especially in embedded systems. This fact is important for laboratory practice where the aim is the right using of program functions and not effectiveness.
- If the level of programming skills is high and the program is simple, using Assembler is appropriate. There is a possibility to work directly with the hardware and acquire knowledge about how the microcontroller acts on the lowest level. When the source code is short, it can be optimized to work with maximum effectiveness depending on the concrete hardware. In general, the bigger source code could decrease the effectiveness of the program work because of the too many details, which could be concerned. In this case, using Assembler is not appropriate.

REFERENCES

[1] Shtrakov S., Kalpachka G., Stoilov A., Computer Training Tests for Control and Evaluation of Student' Achievements, E+E, 2005, i. 3-4, p. 68-73.

[2] Dosev L., Pedagogical Psychology

[4] T. K. Hamrita and R. McClendon, A New Approach for Teaching Microcontroller Courses, *International Journal of Engineering Education*, Vol. 13, No. 4, pp269-274, 1997.

[5] Vandev D., Theory of Probability and Mathematical Statistics, Sofia, 2002, Science and Art.