Microcontrollers in the education – introduction and integration of different setups

Nikoletta Tolner and András Dávid

Campus of Alba Regia Technical Faculty of Óbuda University
H-8000 Hungary Székesfehérvár, Budai str. 45, Hungary
tolner.nikoletta@amk.uni-obuda.hu
david.andras@amk.uni-obuda.hu

Abstract— One of the strengths of Obuda University (Alba Regia Technical Faculty) is the practice-based information transfer. In accordance with the expectations of the industrial colleagues, fundamental knowledge on microcontrollers has been integrated in the curricula in the last fifteen years. Our group tries to integrate the most recent specific developments of the field into the seminars. The major aims of the present study are to describe integration strategies of up-to-date knowledge in the field of microcontrollers and to introduce educational examples for microcontroller DEMO cards. The usage of the above setups and experiences are discussed in the present manuscript.

I. Introduction

A. *A few words about microcontrollers*

Although there had been computers before, microcomputers have existed since 1971 or 1973. That kind of integrated circuits were produced then, which contained an Arithmetical Logical Unit and its Control Unit in one chip. That was the microprocessor, manufactured by Intel. In order to develop the microprocessor into an operating computer, several supplementary circuits were needed. The aim was put them together into the processor chip. As a result, microcontrollers were produced, also by Intel. It was type 8048, followed by type 8051 in 1980. This type is still the most common microcontroller type.

At the same time the Microchip Technology company in Arizona came out with the PIC microcontroller in 1985. There are many types of PIC microcontrollers available now. They have a reduced instruction set (RISC architecture), so it is easy to learn the instructions.

Microcontrollers are used nearly everywhere now. A lot of industrial and controlling tasks are solved by microcontrollers. They have become so widespread and are used in so many areas, that we found it necessary to include them in our curriculum.

You can find a book on microcontrollers type 8051 in reference [1]. PIC microcontrollers are discussed in books you can find in references [2] and [3]. All books contain theoretical and practical information about microcontrollers.

II. Aim

A. *The purpose of our article*

The teachers of the Alba Regia Technical Faculty of our university put a lot of energy in teaching hardware and microcontroller knowledge. We (the authors) are also engaged in this task. The aim of our article is to describe our work in this field.

B. *The structure of our article*

After a short introduction we will provide a list of the microcontroller related university subjects. You can also read about our curriculum and the tasks we designed for students to practice their knowledge of microcontrollers. At the end of the article we will sum up our experiences.

III. Methods

A. *Tools needed for education*

A standard x86 PC with 1 GB RAM minimum (2 GB recommended) and 500 MB+ of available disk space, running one of the following operating systems (both 32-bit and 64-bit systems are supported):

Microsoft® Windows XP (SP2 or greater)

Microsoft® Windows Vista

Microsoft® Windows 7

USB 2.0 port for downloading programs

DEMO cards

Silabs Software and hardware tools to support designers using 8-bit (8051) MCUs

MPLAB® X Integrated Development Environment for PIC microcontrollers

B. *Microcontrollers and DEMO cards needed for education*

We use two basic types of microcontrollers in our lessons. The schematic figures of both types can be found below, so you can see what units they contain.

When teaching the subject named Programmable Circuits we use a DEMO card containing 8051 based C8051F040, C8051F320 and C8051F340 microcontrollers, produced by Silicon Laboratories. Schematic figures of the listed microcontrollers can be seen on Figure 1.

When teaching a subject named Embedded Systems, we use a DEMO card containing a PIC16F887 type microcontroller, produced by Microchip Technology. Its schematic figure can be seen on Figure 2.

For more information about 8051 based microcontrollers that we use in lessons see the link in reference [4]. The link in reference [5] provides information about PIC microcontrollers and contains the datasheets as well.

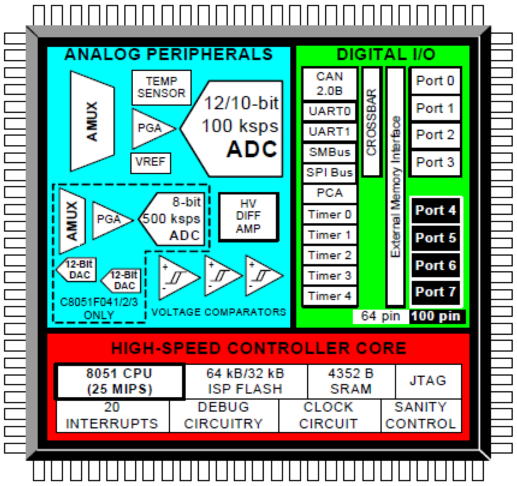

Figure 1a: The schematic figure of C8051F040 microcontroller

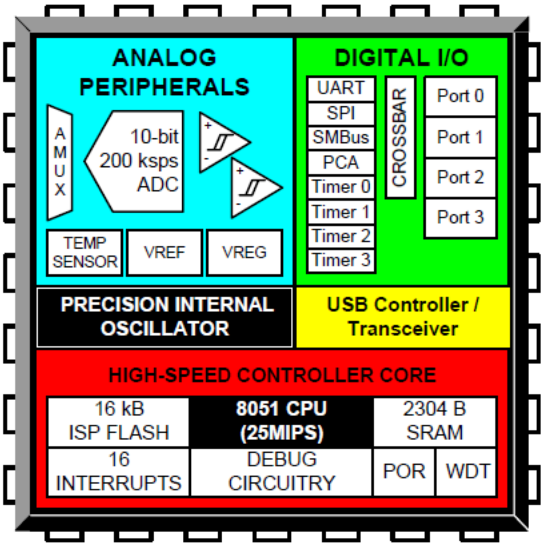

Figure 1b: The schematic figure of C8051F320 microcontroller

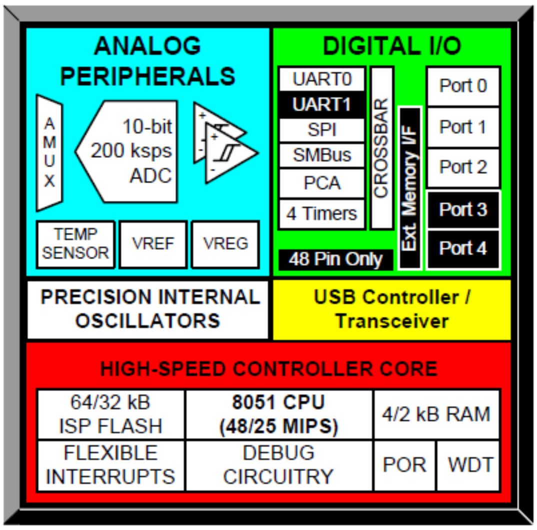

Figure 1c: The schematic figure of C8051F340 microcontroller

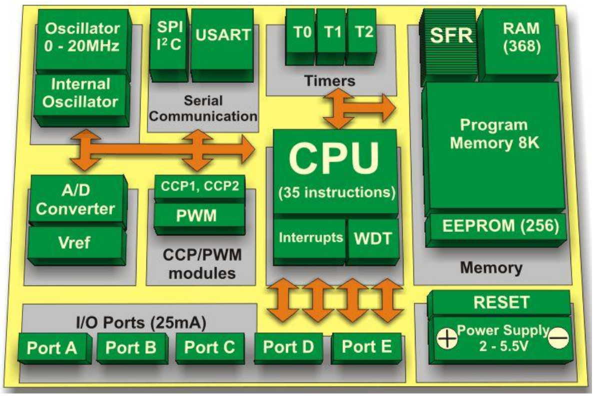

Figure 2: The schematic figure of PIC16F887 microcontroller

C. Which university courses contain microcontroller knowledge?

We teach microcontroller knowledge (based on type 8051) in Programmable Circuits and Digital Systems lessons. Programmable Circuits is a course for electrical engineer students, while Digital Systems is taught for informatics engineer students. Unfortunately we have time for only a few lectures in Digital Systems, so the present article deals with Programmable Circuits and Basics of Embedded System courses.

PIC programming is taught in Programming II and Basic of Embedded System courses. Programming II is a course for electrical engineer students, Basic of Embedded Systems is for informatics engineer students. Both electrical and informatics engineer students are taught C based programming language before that. Programming II course for electrical engineer students contains I8086 microprocessor assembly, then also PIC assembly. Informatics engineer students do not learn assembly language before, so we teach them PIC programming in C language. In the present article we describe 8051 based microcontroller programming in assembly language, so we would like to describe PIC microcontroller programming in C language.

IV. RESULTS

A. Structure of Programmable Circuits course

This course contains 4 lectures per week. Because of the relatively small number of students, lectures are held in the Digital Technology Laboratory. The other reason for choosing the Laboratory was that we wanted the students to gain practical experience in this field. Lessons are structured in the following way: the first part of the lesson is a lecture, the second part is more like a seminar, so students can put their knowledge into practice immediately. For this purpose the University has bought DEMO cards together with the integrated development environment. Students use those tools also when writing their thesis. In the following part we describe the topics and structure of the Programmable Circuits course.

Lesson 1. History of microcontrollers

In this 4 hour lesson students learn about the development history of microcontrollers and study basic type 8051.

Lesson 2. Programming microcontroller type 8051

During this lesson the instruction set and instruction executive mode of microcontroller type 8051 is presented to students. We talk in detail about programming the units of the microcontroller and also about the setting possibilities. The units are the clock signal unit, the serial port, the UART and the timers. We explain how to handle the interrupt system. Assembly language is used for programming. Our concept was that hardware students should write programs not only in C language, but they also should gain practice in Assembly language. There are several firms that look for young engineers who can program microcontrollers in Assembly language. Although we get sample programs with DEMO cards, most of them are written in C language.

Lesson 3. Microcontroller type C8051F040/320/340

In these four hours students learn about the DEMO card and the microcontroller that can be found on it. We examine the integrated development environment when it operates. The integrated development environment contains a program that can be used for writing, compiling and downloading the program through USB.

The program also gives possibility for debugging and step by step execution. Moreover, it contains a utility program named config2 that provides what data you have to load and to which function register, to initialize the different units. At the end of the lesson students download onto the card a simple LED blinky program. We use it to explain how an assembly language program works and what parts it has. Students get to know the integrated development environment in use, too. The sample program can be found at the end of the chapter.

Lesson 4. Units of microcontrollers type C8051F040/ 320/340

During this lesson we study the settings of the clock signal unit and the operation of the UART in detail. In the second part of the lesson we also program these units. Then we discover what frequency clock speeds we can produce by using the inner or outer clock signal unit. When programming the serial line port, we also practice handling interrupts. By connecting two cards we can analyze the process of sending and receiving.

Lesson 5. Units of microcontrollers type C8051F040/ 320/340

In this 4 hour block students learn about the use of timers. The theoretical part is followed by a programming task. Now baud rate setting can be done by use of timers.

Lesson 6. Test writing and discussion of term homework

The test is designed to measure the students' theoretical knowledge. The term homework measures their programming skills. We do not appoint a concrete programming task but ask the students to come up with their own ideas. This way everyone chooses a topic they are interested in, so the term homework is not a boring compulsory task anymore.

Lesson 7. Units of microcontrollers type C8051F040/ 320/340

We teach about the theoretical basis of temperature measurement and the operation of TEMP sensor.

Lesson 8. Units of microcontrollers type C8051F040/ 320/340

Introducing SPI and its programming.

Lesson 9. Units of microcontrollers type C8051F040/ 320/340

Introducing SMBUS and its programming.

Lesson 10. Units of microcontrollers type C8051F040/ 320/340

In this 4 hour block we study the structure of the CAN bus. In the second part of the lesson we also program the units of the CAN bus.

Lesson 11. Units of microcontrollers type C8051F040/ 320/340

Introduction of USB and description of programming of this microcontroller.

Lesson 12. Test writing and presentation of term homework

After the test the students give a presentation of their term homework.

Lesson 13.

In this 4 hour block students rewrite the unsuccessful tests, and continue the presentation of term homework. Each student presents their homework orally, this way students can prepare for their thesis exam. We found that the majority of students find it difficult to express themselves orally, therefore we ask them to present their homework assignments orally as well. Of course, they also need to provide documentation of the task.

At this point the term ends. Students must sit an exam. Only those students can enter they exam who achieved at least 50% of points at both tests and handed in their term homework, and also produced an operating program.

B. Sample program in Programmable Circuits course

Sample program presented in Lesson 3.:

```
$NOMOD51
$include (c8051f040.inc)
; Include register definition file.
; EQUATES
GREEN_LED  equ   P1.6
; Port I/O pin connected to Green LED.
; RESET and INTERRUPT VECTORS
       cseg AT 0
       ljmp Main ; Locate a jump to the start of
                 ;code at the reset vector.
; CODE SEGMENT
Blink  segment  CODE
       rseg    Blink ; Switch to this code
                     ;segment.
       using   0     ; Specify register bank for
                     ; the following
 Main: ; Disable the WDT.
       ;(IRQs not enabled at this point.)
       ; If interrupts were enabled, we would
       ; need to explicitly disable
       ; them so that the 2nd move to
       ; WDTCN occurs no more than four clock
       ; cycles after the first move to WDTCN.
       mov   WDTCN, #0DEh
       mov   WDTCN, #0ADh
       ; Use SFRs on the Configuration Page
       mov   SFRPAGE, #CONFIG_PAGE

       ; Enable the Port I/O Crossbar
       mov   XBR2, #40h
       ; Set P1.6 (LED) as digital output
       ;in push-pull mode.
       orl   P1MDOUT,#40h
       ; Initialize LED to OFF
       clr   GREEN_LED
       ; Simple delay loop.
Loop2: mov   R7, #03h
Loop1: mov   R6, #00h
Loop0: mov   R5, #00h
       djnz  R5, $
       djnz  R6, Loop0
       djnz  R7, Loop1
       cpl   GREEN_LED         ; Toggle LED.
       jmp   Loop2
END
```

C. Structure of Embedded Systems course

This course contains a 2 hour laboratory practice per week. In the course students use a “44 pin demo board” DEMO card, that contains a PIC16F887 microcontroller. It has 8 LED diodes, 1 push button and 1 potentiometer. To make programs we use the integrated development environment named MpLabX.

Lesson 1. History of microcontrollers

In this 2 hour lesson students learn about the development history of microcontrollers and get to know the structure of PIC16 type microcontroller.

Lesson 2. The review of C programming

In this lesson we review the knowledge of C programming and introduce the special attributes needed for microcontroller programming.

Lesson 3. Introducing the block diagram of microcontroller PIC16F887

In this lesson students get information about the special attributes of this microcontroller. Informatics engineer students have not completed Computer Architectures course at this stage, so we must teach them some hardware knowledge as well. Among other things, we introduce Neumann Architecture, Harvard Architecture, structure of machine-code instructions, the concept and structure of the data and program memory, special function register (SFR) characteristics and role. We also describe they peripheries of the PIC16F887 microcontroller.

Lesson 4. Introducing the Integrated Development Environment (MpLabX) and the DEMO board

In this lesson the students get to know the integrated development environment (MpLabX). This environment contains a utility program that makes it easy to set the initial config settings. It generates a code that can be easily used in the program. We introduce the IDE and the DEMO card by our own sample program. By analyzing the sample program we demonstrate the operation of a microcontroller program step by step. We also talk about its parts: config, initializing of periphery, etc. The sample program can be found at the end of the chapter.

Lesson 5. Creating a „running light” on the LED row

During this lesson the functions of I/O ports are explained. Students make a program that creates a “running light” on the LED row by using software delay.

Lesson 6. Creating a „running light” on the LED row

In this lesson we review the knowledge, then students make a program which write a bit pattern the LED row values obtained with different methods of binary encoding (simple binary code, BCD code, gray code).

Lesson 7. Using the push button on the DEMO card

In this lesson students learn how to handle the push button on the DEMO card. We handle the push button by polling and by interrupt, too. Student’s task is to stop the “running light” on the LED row by the push button, then to change its direction. Since students do not have a deep hardware and assembly programming knowledge, we talk about interrupts in detail.

In this lesson students are given a term homework that they have to hand in Lesson 13. and also have to give a presentation of it in Lesson 14.

Lesson 8. The presentation of the timers of the PIC16F887 microcontroller

In this lesson students learn how to handle the timers of PIC16F887 microcontroller. We transform the program written in the last lesson that we solve timing by timers.

Lesson 9. The presentation of the timers of the PIC16F887 microcontroller

In this lesson, the students get tasks based on programs written in the last lesson. They have to change the direction of “running light” on the LED by timers, using polling or interrupt. They also have to stop and restart the “running light” by using the push button.

Lesson 10. The presentation of the AD converter of the PIC16F887 microcontroller

In this lesson students learn with how to handle the AD converter in the microcontroller. The students have to set the running speed of the “running light” by using a potentiometer. We handle the AD converter by polling.

Lesson 11. The presentation of the AD converter of the PIC16F887 microcontroller

We demonstrate how to use AD converter by the use of interrupt. We transform the program written in the last lesson that it is handled by interrupt.

Lesson 12. The presentation of the Capture/Compare/PWM unit of the PIC16F887 microcontroller

In this lesson students learn with how to handle the Capture/Compare/PWM unit in the microcontroller. This is explained in the microcontroller, and we give some practical examples. The students make a program to generate a pulse-width modulated signal of varying frequency and duty cycle.

Lesson 13. Theoretical and practical test

In this lesson students write a theoretical and practical test, which means they have to make their own program. After the test students present their homework.

Lesson 14.

In this 2 hour block students rewrite unsuccessful tests, and continue the homework presentation. Everyone must present the task orally. Students get their term marks which are based on their achievement in the test and homework.

D. Sample program in Embedded Systems

Sample program presented in Lesson 4.:

```
#include <xc.h>
// CONFIG1
#pragma config FOSC = INTRC_NOCLKOUT
/* Oscillator Selection bits (INTOSCIO
   oscillator: I/O function on
   RA6/OSC2/CLKOUT pin, I/O function on
   RA7/OSC1/CLKIN) */
#pragma config WDTE = OFF
/* Watchdog Timer Enable bit (WDT disabled and
   can be enabled by SWDTEN bit of the WDTCON
   register) */
#pragma config PWRTE = ON
/* Power-up Timer Enable bit (PWRT enabled)
#pragma      config      MCLRE      =      ON
/* RE3/MCLR pin function select bit (RE3/MCLR
   pin function is MCLR) */
#pragma config CP = OFF
/* Code Protection bit (Program memory code
    protection is disabled) */
#pragma config CPD = OFF
/* Data Code Protection bit (Data memory code
   protection is disabled) */
#pragma config BOREN = ON
```

```
/*    Brown    Out    Reset    Selection    bits
    (BOR enabled) */
#pragma config IESO = OFF
/*    Internal    External    Switchover    bit
    (Internal/External   Switchover   mode   is
    disabled) */
#pragma config FCMEN = OFF
/*   Fail-Safe   Clock   Monitor   Enabled   bit
 (Fail-Safe Clock Monitor is disabled) */
#pragma config LVP = OFF
/* Low Voltage Programming Enable bit (RB3 pin
   has digital I/O, HV on MCLR must be used
   for programming) */
// CONFIG2
#pragma config BOR4V = BOR40V
/* Brown-out Reset Selection bit (Brown-out
   Reset set to 4.0V) */
#pragma      config      WRT       =       OFF
/* Flash Program Memory Self Write Enable bits
 (Write protection off) */
void kesleltetes()
{  for(int i=0; i<8000; i++); }

void main(void)
{ TRISD=0x00;
  TRISB=0x01;  //RB0 input
    ANSEL=0, ANSELH=0; //digital input
    int ledek=0x80, irany=0xFF;
    while(1)
    { if(irany==0xFF)
       { if(ledek==0) ledek=0x80;
             else ledek=ledek>>1; }
        else
        { if(ledek==0) ledek=0x01;
             else ledek=ledek<<1; }
        PORTD=ledek; kesleltetes();
        if(PORTBbits.RB0==0)
        { irany=irany ^0xFF;
          while(PORTBbits.RB0==0); } } }
```

V. Discussion

The structure of two seminars and two adjacent sample programs were introduced in the present paper.

In these lessons we try to give practical knowledge to students. Students enjoy lessons and are interested in the topics. They are convinced they will be able to use this knowledge in the future.

A. *Educational experiences*

Teaching theoretical knowledge takes a long time, so it is not always possible to create a fully operating program by the end of each lesson. In such cases students get useful instructions for programming and finish the task at home.

There are some students who already have some knowledge of microcontrollers so we often ask them to help other students. They usually finish program writing sooner, then we ask them to help their mates and correct mistakes. This way good students do not get bored, weaker students get proper help even if the teacher does not have time for everyone.

Unfortunately most sample programs made by Silicon Laboratories are written in C language, so it would be a great help to have much more assembly language programs.

It is worth having a detailed lesson plan, so that students can work as fast as possible and have enough time for programming tasks.

It proved to be a good idea to ask the more experienced students to give a presentation during lessons, so they can share their programming experiences. By doing this, they get a deeper knowledge of the topic, too. The presentation is a requirement to get a good end-of-term mask.

Our experience is that 2 tests are necessary to make students study during term, so that they do not put off studying till the end of them.

It is very important that each student should work alone during lessons. It means that a proper number of computers and DEMO cards is needed in lessons. Our experience is that team work is not useful in this area, as usually only one person is active in each team.

Most microcontrollers are based on type 8051. If students get enough experience in using it, it will be easy for them to use a similar microcontroller produced by other companies. We tried to find a modern microcontroller that has many peripheries (CAN, SMBus, SPI, USB…). When choosing a manufacturer, the price was important, but we also wanted to find a company that can provide a complete system. This means the DEMO card, the integrated development environment, compiler and debugger. Finally we chose Silicon Laboratories that manufactures a wide range of microcontrollers and fulfils all our requirements.

We found it important that students get to know not only basic type 8051, but another type as well, so we chose PIC microcontrollers because of their reasonable price and popularity. The university bought DEMO cards manufactured by Microchip Technology and downloaded the integrated development environment from company website.

B. *Microcontroller education in other universities*

Many universities and colleges in Hungary provide microcontroller education. We have access to the curriculum of the Budapest University of Technology and Economics. It is very similar to our curriculum, although they can devote more time to these subjects. Though we have less time for teaching these subjects, we try to give practical knowledge to students.

VI. Acknowledgment

Special thanks to Dean Dr. György Györök and Imre Grosz for their support and help. When writing the present article we used articles written by Dean Dr. György Györök [reference 6] and Rita Ősz [reference 7]. We are also grateful to our students for their feedback.

VII. References

[1] Roland Dilsch: A 8051 Mikrokontroller-család, Műszaki Könyvkiadó, 1992

[2] Kónya László-Kopják József: PIC Mikrovezérlők alkalmazástechnikája, PIC programozás C nyelven (harmadik, bővített kiadás) 2003 ChipCad

[3] K. Lamar. A világ leggyorsabb mikrovezérlője. ChipCAD Kft., page 96, January 1999.

[4] http://www.silabs.com/products/mcu/pages/8051-microcontroller.aspx

[5] http://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers

[6] György Györök: Practical application of embedded microcontroller. HAWK Fachhochschule Hildesheim/ Holzminden/Göttingen, 2006.

[7] Rita Osz: Educational organization for new generation In: Szakál Anikó (szerk.). IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics, SAMI 2012 - Proceedings.