THE TRANSITION FROM THE 8085 TO THE "MODERN" WORLD

James S. Finne, PE
Assistant Professor, Engineering Technology
Middlesex County College

John Carpinelli, PhD
Associate Professor, Electrical and Computer Engineering
Director, Center for Pre College Programs
New Jersey Institute of Technology

William Barnes, PE
Associate Professor, Electrical and Computer Engineering Technology
New Jersey Institute of Technology

Abstract

Teaching the **Introduction to Microprocessors** course at Middlesex County College using a pedagogically relevant but technically obsolete Intel 8085 is a dilemma. The modern 8 bit Micro Control Unit is far more complicated and more sophisticated than the 8085. The goal was to reduce a modern microprocessor to the level of the Intel 8085 for the purpose of introducing students with no prior background to programming and interfacing at the microprocessor, or micro control unit, level. The choice of processors and demonstration boards makes the process even more daunting. The field has gone from one choice (Intel 8085) to thousands (Intel, Atmel, Motorola, MicroChip, and others). Which is the right choice? This paper describes the tack taken by Middlesex County College, using a MicroChip PIC16®, 8 bit MCU, for the Introduction to Microprocessors, and the use of a plain, solderless breadboard instead of an available demonstration board.

An Introduction to Microprocessors course must include the topics of register set and instruction set architecture, machine level programming and interfaces to the physical world. Taught at the community college level, the topics need to be understandable and relevant. The topics also must be general enough to apply to what ever processor or system may be in vogue at the students' chosen upper division school. There is no single correct choice. There is also no wrong choice.

This paper discusses the selection of a processor and the rationale for using a solderless breadboard rather than a more sophisticated demonstration board. The paper also discusses why we chose an assembly language over a higher level programming language for the introductory course and the set of laboratory exercises that guide the student through the learning process in a third semester engineering technology course.

Goal and Situation

The goal is to teach an Introduction to Microprocessors course to third semester Engineering Technology students in a way that will catch their interest and provide a background on which subsequent courses can build. At Middlesex County College, Introduction to Microprocessors is

a 3 credit course taught in a 14 week semester. Classes meet for 2 lecture hours and 3 laboratory hours each week

MCC previously used a lab manual developed by Prof. Steve Foster around 1980, in conjunction with available texts and the SDK-85 single board computer with an Intel 8085 processor. The SDK-85 was pre wired with 16 LEDs on two 8 bit ports. Programs were entered from the keypad using hexadecimal numeric op codes that were displayed on a seven segment display. The keypad and display interface routines were preprogrammed in ROM. Some of the SDK-85s had been in service for 25 years.

The lab manual introduced the subject, including the 8085 assembly language with numeric op codes and register set, with a series of hands-on exercises that required the student to write code, interface components to the I/O ports, and to answer workbook questions. This manual was a work horse for the program and served well for many years. The equipment, however, was becoming troublesome and was no longer supported by the manufacturer.

The 8085 is a good teaching tool. The architecture and language are both simple enough that a student can understand and visualize the fundamental principles, yet complex enough to introduce important concepts, including programming structures, memory and port data transfers, interrupts, stacks and indirect addressing .

The questions that we set out to answer were:

- Is it appropriate to abandon this excellent teaching tool in favor of a more modern and more complex processor?

- Is it appropriate to teach assembly code (and machine op code) instead of using a compiled language such as Basic or C?

- How would we use a more complex processor to teach the fundamental principals without getting mired in the complexities of the language and the register set?

- What choices do we have for processors, single board computers and text books?

Is it appropriate to abandon the 8085?

Yes, with qualification. To adopt another processor, or family of processors, would require that we simplify it to the level of the 8085 and let the students discover the complexities as they become comfortable with the language and hardware. This makes the issue one of teaching, not hardware. The choice of hardware now becomes more subjective. The student might think that the best choice is the latest Intel®Core ™ 2 Duo Processor with a 4 MB cache memory.

The modern microcontroller is, for all intents and purposes, the single board computer with the Intel 8085 processor. The microcontroller has built in non-volatile memory, built in ports and other support hardware such as timers, analog/digital converters, and asynchronous communications. Microcontrollers will usually be found in “embedded” applications where they

will control a single piece of hardware and have limited user interface. Typical examples given to students are the microwave oven and anti-lock brakes. Microprocessors, on the other hand, are more generalized computers that facilitate user application programs. These are processors that facilitate math functions and strive to execute the maximum number of instructions in a given period of time. Microcontrollers typically use 8 bit data while microprocessors use 16, 32 or even 64 bits data words.

From an instructional point of view, the microcontroller suits the classroom because it is fundamentally simpler and it accommodates simple electronic interfaces.

We limited the choice to 8 bit microcontrollers and decided on the MicroChip PIC16. Specifically we chose the PIC16F877 to represent the family of micro controllers. The PIC16 uses a simple assembly language with only 35 instructions, and its register set is not unwieldy. This choice is as valid as any other manufacturer or processor. Microchip is one of the dominant manufacturers for the microcontroller market.

Language

We can debate the value of introducing the subject using assembly code versus using a compiled language but, as an introduction, we believed that the fundamental principle is to understand the transfer and manipulation of data within the set of registers and the interface of the registers to the physical world. The linear one-for-one relationship of assembly code illustrates the fundamental workings of a microprocessor in a way that is not apparent in a higher level language. Assembly code, however, is cumbersome for complex tasks, but for an introductory course the tasks are simple and fundamental. At Middlesex County College the Electrical Engineering Technology students are required to take a course in C++ programming, but this course is not a pre- or co-requisite for the Introduction to Microprocessors course.

Assembly code is how the computer works and a higher level language is always translated into assembly code. Understanding how the processor handles assembly code helps the student to understand how the processor uses a higher level language. A processor always manipulates one register at a time, doing one task at a time. This task is defined by the assembly code instruction that is being executed. We know that this is not necessarily true in many or most modern microprocessors, but at the introductory level it is a valid statement.

Demoboards

MicroChip products have associated with them a wide choice of single board computers or demonstration boards that are prewired with devices that are typically interfaced to a microcontroller (LEDs, encoders, switches, potentiometers, LCD displays, and RS232 ports). Most leave a little physical room for the user to install other components.

Our difficulty with demo boards is that they are prewired and therefore do not fully allow for the designing and building of interface circuits. The common denominator is that all have a power supply circuit, oscillator and a means to interface with a development environment's programming device. These are necessary for every operation.

Our second difficulty with demo boards is that they have a significant cost which makes them a capital investment for the school or a very steep laboratory fee for the student.

The approach that we took was to provide the necessary hardware and then use a solderless breadboard to build a single board computer. We designed an adapter board that consists of a power jack, 5 Vdc regulator, RJ12 jack required to interface the Microchip ICD-2 programming device, and an oscillator. It connects to a solderless breadboard with six pins. With six wires and a processor the student has a working single board computer for under $30.00. With some care at the assembly stage this board should be serviceable for several semesters and is, in fact, used for the subsequent capstone project course. The only capital expenditure is for the programming device that connects the development software to the microcontrol unit. MCC purchased MicroChip ICD-2s to support the class laboratory exercises.

Textbook

This is controversial. The hard realities are that all of the available texts are expensive and, more often than not, the texts are underused by the students. It is common to find students who never even purchase them. Rather then requiring a textbook we chose to use reference materials that are available from MicroChip, use MicroChip's technical data sheets and manuals and provide lecture notes or PowerPoint slides. A textbook is optional.

This approach gets a mixed reception from students. Some want a textbook but most seem to have little difficulty using the data sheets. The learning process is predominately hands-on so reference material is more useful than lengthy descriptive material.

For the MicroChip products, most of the available texts are written around the PIC18 devices. Our choice was the PIC16 because the assembly language was somewhat simpler and some of the more sophisticated data handling features (indirect addressing and auto indexing) are not present. The PIC18 is more suitable to projects using a compiled language and projects that require more mathematical operations.

Laboratory Exercises

A new laboratory manual was written, using the old manual as a model. It is published on a CD and currently distributed to the students. All students are required to individually complete the first five laboratory exercises. The first exercise is to assemble the adapter board and to populate a solderless breadboard with the processor, an LED display and the adapter board. We distribute processors with programmed flash memory that demonstrates that the system is fully functional. This exercise is tedious because for many students it is their first experience at assembling a printed circuit board. It can be argued that such an assembly is outside of the scope for this course but such experience is valuable nonetheless. Pre assembled adaptors are a possibility for the future.

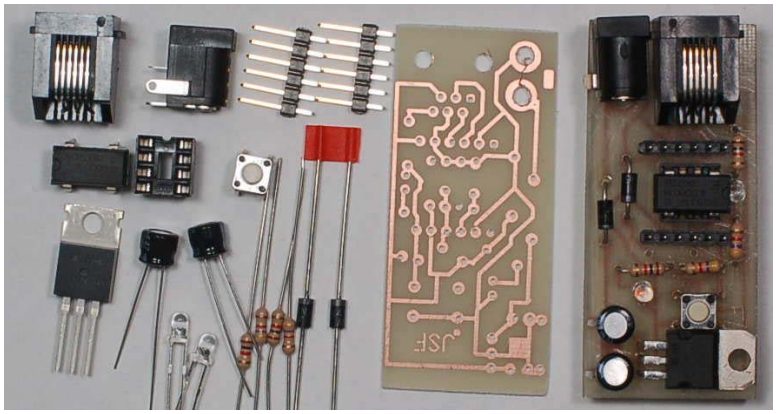


Figure 1: Kit for Adaptor Assembly showing all components and finished assembly.

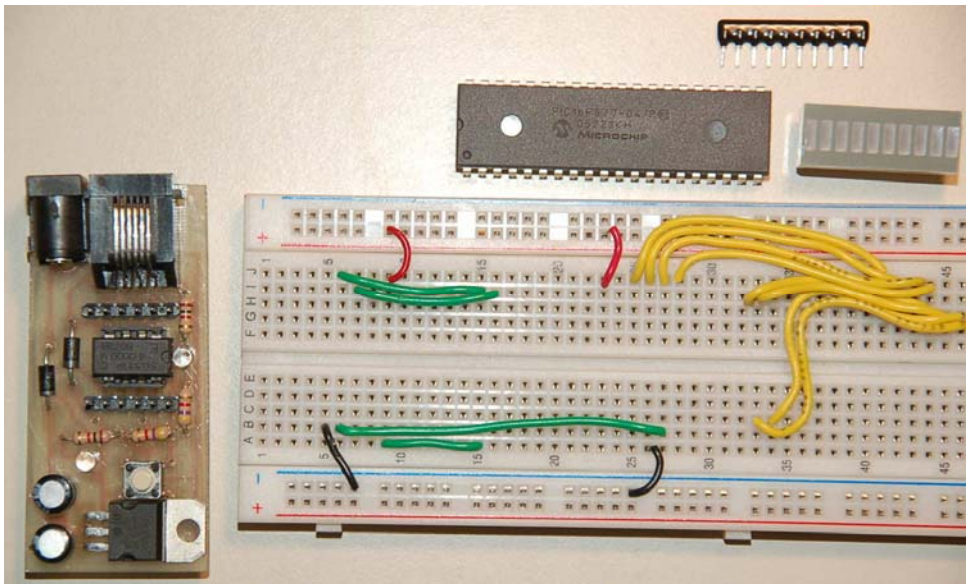


Figure 2: Solderless breadboard with wires cut short and trimmed. Vdd are red, Vss are black, green connect MCLR, OSC, RGC and RGD, and yellow connect the LED bar graph to PORTC of MCU. There are a total of 17 wires for this exercise.

The second exercise is a rote entry of a “hello world” program that takes the student through the step-by-step process of typing an assembly language program, connecting the development software to the processor, building the program into a machine code hex file, loading the program, and observing the resultant activity of the board. This exercise is done “by the numbers” to give the student a step-by-step procedure to follow for all future work. We point out to the students that the development software and the hardware are all standard industrial components that can be found in the workplace and the adapter hardware is what would normally be used in any design.

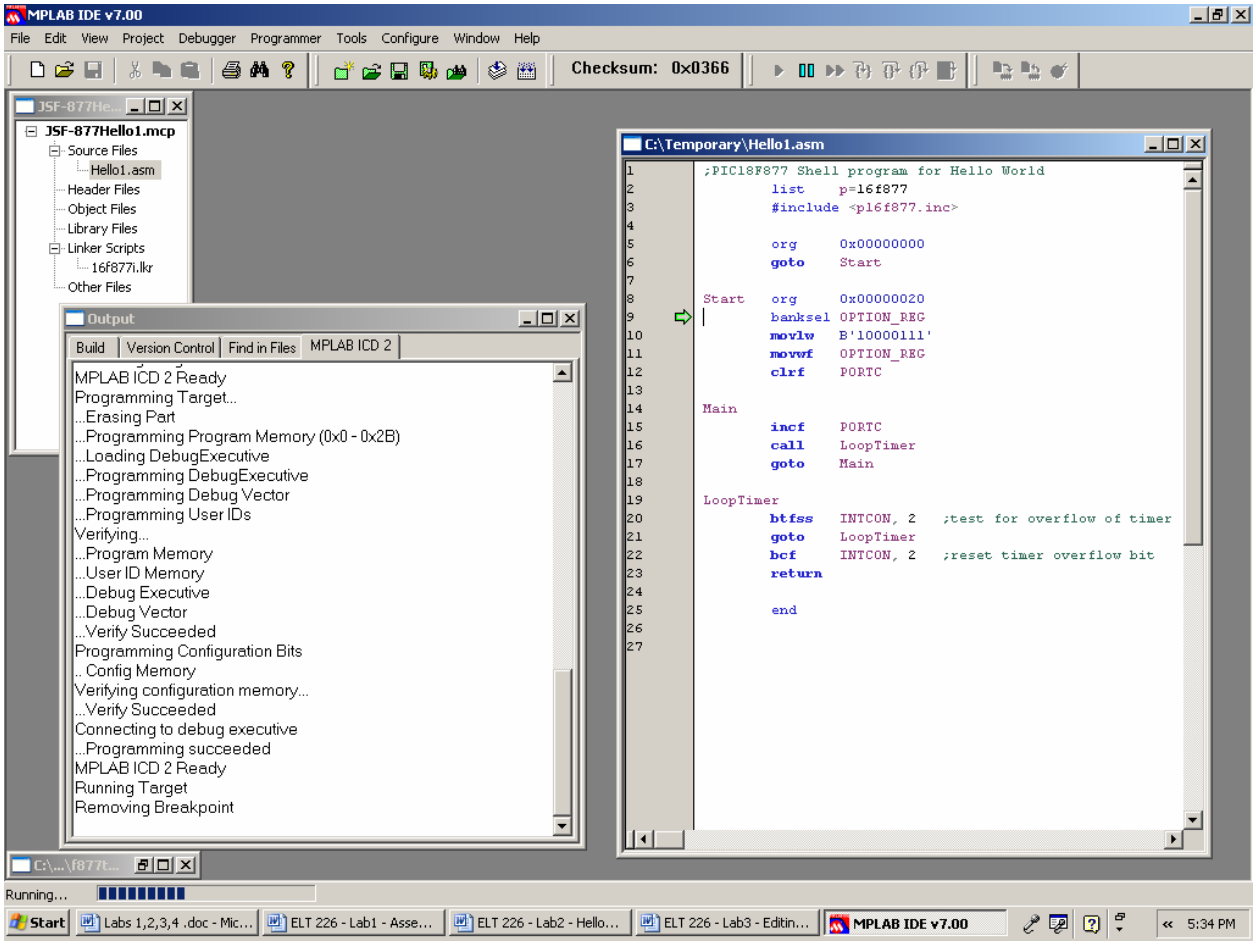


Figure 3: Screen capture of MPLAB IDE ver 7.0 showing the file structure, assembly language program and the communications between the IDE and the target MCU.

The next three exercises ask the student to modify their “hello world” program, to add inputs to one port of the processor and to modify the timing loop of the “hello world” program. At this point the student should be comfortable with simple electronic interfaces and writing simple assembly language code. We improvise variations on these exercises.

After the first five exercises, the students work in groups to develop a variety of interface circuits and to write code for the circuits. The exercises explore some of the hardware features of the PIC16, such as analog/digital conversion, interrupts, USART, SPI, I^2C, and pwm. The exercises explore other types of hardware interfaces and driver circuits. The intention with this phase of the course is to give the students the ability to pick what interests them and to give them control over their work.

From a teaching point of view, this means every group is on a different page and having a different set of problems. This does present logistical challenges, but it also allows students at different levels to progress at their own pace within the same course, thus maximizing individual student achievement. It also fosters student interactions with stronger students helping the weaker students with both reaping benefits.

The successes

The real success of the curriculum change is seen in the students' fourth semester capstone projects. We now require the capstone project to be microcontroller based and treat the design course as a follow-up course to Introduction to Microprocessors. This year's collection of projects include an autonomous robot, a basketball shot clock timer with wireless remote display, an ultrasonic backup alarm, and an architectural measuring tool that inputs directly to AutoCAD®. The level of sophistication exceeds anything attempted in the project class in many years and the work is almost totally original. Although it can only partially be attributed to the new microprocessor curriculum, three groups have, on their own, made the leap to programming in C, using a PIC18F or PIC30F.

It remains to be seen how these students will perform in their upper division microprocessor courses that are taught using different hardware and using a compiled language.

Other observations

The single largest difficulty is the assembly of the adapter PC board because it is the first experience with this kind of work for most of the students. Currently we use prototype boards manufactured in our own laboratory. These are single sided, through hole boards without solder mask. We expect to have commercially produced boards for Fall 2007. The commercial boards will have silkscreen component outlines and soldermask. This will solve most of the difficulties and it opens up the option of having pre-assembled boards made commercially. Assembly difficulties manifest themselves as short circuits (solder bridges) or open circuits (cold solder joints). We have developed a trouble shooting procedure; however, problems at this stage produce a level of frustration that, in some cases, is difficult to overcome. Troubleshooting defects is a challenge for many students at this level.

The Microchip Integrated Development Environment software is not "educational" software. This software is intended for the expert user and is often not user friendly. This sometimes produces a level of frustration for student and instructor alike. We remind the students that they are working with real software, not an educational version and not simulation.

The course was designed for ten laboratory exercises (five individual and five group). The history demonstrates that eight exercises is realistic. In every class, one or two students will run through ten exercises in half a semester and then move on to advanced material on their own.

A book (or CD) of laboratory exercises is good because flexibility in allowing variations is important. Publishing the lab manual as a CD facilitates revisions and additions. The CD also includes reference materials such as data sheets for interfaced devices.

Students are the best course developers and it is important to be able to pick up on and support their curiosities. For example,

- Modified, the timer subroutine becomes a 1 second/1 minute/1hour clock that then becomes the basketball shot clock timer. The shot clock timer will further morph into a classroom timing tool that, among other things, will facilitate the use of PowerPoint® Jeopardy played as a classroom exercise.
- Binary counter LED display (exercise 2) is modified to become "Knight Rider" or some other strobing display.
- USART communications becomes a multi group exercise with communications between student boards (Analog input for board 1 is displayed on board 2) and ultimately a wireless radio communications between boards.

Conclusion

Replacing the Intel 8085 single board computer with an 8 bit microcontrol unit, the PIC16, implemented on a solderless breadboard has been a very cost effective and pedagogically effective development tool for our Introduction to Microprocessors course. Using the solderless breadboard instead of a manufactured demonstration board has also made the course more relevant and realistic for the students.

References

Steven Foster "Microcomputers – ELT 226 Lab Manual", Middlesex County College, Electrical Engineering Technology Department

Roy Goody "Intel Microcomputers: Hardware, Software, and Applications" Glencoe

Wm. Kleitz "Digital and Microprocessor Fundamentals, Theory and Application" Prentice Hall

Han-Way Haung "PIC Microcontrollers: An Introduction to Software and Hardware Interfacing" Thomson

John B. Peatman "Embedded Design with the PIC18F452 Microcontrollers" Prentice Hall

John D. Carpinelli "Computer Systems: Organization & Architecture" Addison Wesley

"PICmicro™ Mid-Range MCU Family Reference Manual, Dec 1997"
Available at www.microchip.com

"PIC16F87X DataSheet – 28/40 Pin 8 Bit CMOS FLASH Microcontroller, 2001"
Available at www.microchip.com

Microchip MPLAB® IDE – Integrated Development Environment
Available at www.microchip.com or with purchase of In-Circuit-Debugger 2.

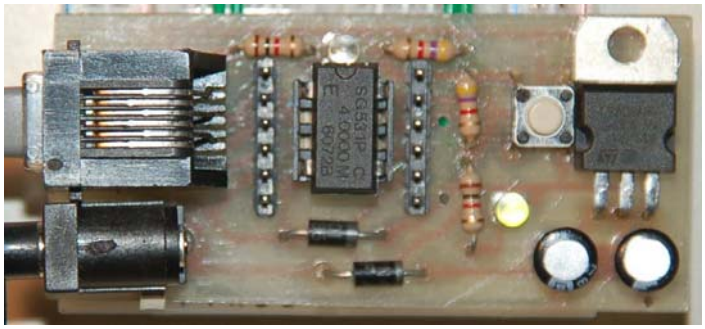
Resources

Any educational institution interested in adopting the approach of using our adapter PCB can contact the author at jfinne@middlesexcc.edu. We may be able to provide the bare circuit boards at a nominal cost and will also share the laboratory exercises with the understanding that this material will not be duplicated without permission.

Costs:

Adapter PCB (estimated)	$ 5.00	(one per student per semester)
Components	10.00	ditto
Solderless Breadboard	8.00	ditto
PIC16F877	5.00	ditto
Microchip ICD-2	150.00	(less 25% educational discount)

MCC purchased ICD-2s to support one laboratory as permanent equipment, however, many students did purchase their own equipment.



Laboratory Manual
Table of Contents