Teaching PIC microcontroller in EET Program

Asad Yousuf, Savannah State University
William Lehman, Advanced Systems and Materials
Muniyappa Venkatesha, Savannah Technical College
Chung-Ling Huang, Southern University

Abstract

In our Electronics Engineering Technology program at Savannah State University, we have been teaching the Motorola 68HC11 Microcontroller for the last decade. However, the trends in microcontroller technology have changed to provide faster and more robust systems. This demand has led to sophisticated microcontroller architecture. Microchip technology has designed several low cost PIC (Peripheral Interface Controller) microcontrollers.

To meet the demands of this new trend we have decided to teach the Microchip 8-bit PIC microcontrollers. However, before making this transition several factors such as textbook, hardware and software tools must be specified to implement the course in a real working environment.

In an effort to teach students the PIC microcontroller, the Electronics Engineering Technology Department has developed a course in which emphasis is directed towards the PIC microcontroller in addition to the traditional concepts of the MC68HC11. This paper will discuss the course outline, laboratory equipment, and embedded design example with the PIC18F452 microcontroller.

Introduction

Electronics Engineering Technology and Computer Science Technology curricula each requires two courses for the study of microprocessor and Microcontroller. Conventionally, the microprocessor course was designed to introduce the basic concepts in microprocessor system based design followed by a Microcontroller course, which covered the advanced topics and concepts of microcontroller interfacing. For the last decade we have been teaching the INTEL 80586 microprocessor in our Microprocessor course and the Motorola 8-bit 68HC11 microcontroller in our microcontroller course. However, over the past few years, we have seen numerous changes in the microprocessor and 8-microcontroller market. Motorola stopped the development of its popular 8-bit 68HC11 microcontroller for about ten years. With advancement in technology, modern system design requires the use of advanced microcontroller chip and tools.

Several new companies have emerged in the 8-bit microcontroller area to meet the complex design requirements. To fulfill the demands of the new technology more and more universities and colleges have shifted from teaching the traditional 68HC11 microcontroller to teach the Microchip PIC microcontrollers. To meet the design requirements of today's technology we have decided to shift our mode of teaching from the traditional INTEL based microprocessor course and the 68HC11 microcontroller based course to the Microchip PIC microcontroller. The course sequence offered to the students are Embedded Systems I and II. However, the traditional concepts of the MC68HC11 are still covered in the Embedded Systems I course.

The use of 8-bit microcontroller providing embedded control solutions have been incorporated into many products globally. Electronic and other products have been dominated by the traditional 8-bit Motorola microcontroller. However, the Microchip 8 bit micro controllers in recent years have become a significant factor in both industry and academia. The PIC 18XXXX family has been selected for the course as the best compromise to avoid using older technology such as the 1200-1700 series of Microchip micro controllers and the newer but higher cost DSPPIC series. Widespread support exists in Microchip support, books, and technical articles for the PIC 18XXXX. Flash versions of the PIC 18XXXX will be used to provide flexibility and ease of use in the lab [2].

Microchip designs and manufactures numerous families of 8-bit microcontrollers. Among them, the PIC16 family and the PIC18 family are considered as the two major families. There are many similarities in the two families. However, the PIC18 was introduced to remove the limitations of the PIC16 family. The PIC18 family has the following advantages over the PIC16 family[6]:

- PIC18 supports 2MB of program memory space, whereas the PIC16 family supports only 8KB of program memory
- PIC18 family supports external program memory. The PIC16 family does not.
- PIC18 family has much larger on-chip data memory to support the application.
- PIC18 family provides the access bank to minimize the data memory bank-switching overhead.
- PIC18 family provides more instructions, which makes assembly language programming a little easier.
- PIC18 family supports more peripheral functions than the PIC16 family does.
- PIC18 family can run at a higher clock rate and achieve better performance.

Furthermore, the PIC18 family provides a wider range of pin count from as few as 18 pins to as many as 80 pins. With these advanced features, the PIC18 microcontroller family is very applicable for those who want to learn about the modern microcontroller interfacing and programming [2].

This paper will focus on the new paradigm of teaching the PIC microcontroller. The course will be based on the modern development tools which are important in the learning of the PIC microcontroller. Following paragraphs will describe the course contents, laboratory equipment, and a hands-on exercise to show the implementation of the Microchip PIC microcontroller.

Course Content

Electronics and Computer Science Technology students are introduced to Embedded Systems I and II sequence courses at the junior and senior level respectively. Each course is a three-credit hour course consisting of two-hour lecture and a two-hour lab per week. Embedded Systems I is designed to introduce the basic concepts in microcontroller programming and interfacing, followed by Embedded Systems II, which covers the advanced topics.

Following topics are covered in Embedded Systems I and II sequence courses:

1. Introduction to the PIC18 Microcontroller
2. PIC18 Assembly language Programming
3. PIC18 Development Tools
4. Advanced Assembly Programming
5. C language programming and PIC
6. Interrupts, Resets, and Configuration
7. Parallel Ports
8. Timers and CCP Modules
9. Addressable Universal Synchronous Asynchronous Receiver Transceiver
10. Serial Peripheral Interface
11. PIC18 I2C mode
12. Analog-to-Digital Converter
13. Controller Area Network
14. Internal and external memory programming
15. System configuration and Protection

The first three topics covers the PIC18 Architecture, instruction set, and PIC18 software and hardware development tools. In C language programming section the features of the Microchip C compiler is explored. Also, a tutorial is presented on using the PICC18 C compiler. Topics 1 through 7 are covered in Embedded Systems I and topics 8 through 15 are covered in Embedded Systems II.

The prerequisite for the course is digital logic design. It is highly recommended that the student should have a basic knowledge of high-level programming language. The knowledge in high-level programming language will provide the student not only an edge in implementing a program construct in C but will also provide the background to implement the appropriate assembly instruction sequence.

Since, the PIC microcontroller is a new technology for the individual teaching at the University and College level, it was a difficult task to select a textbook which is appropriate for the Electronics Engineering Technology students. Several, textbooks were considered and finally it was decided to adopt the "PIC Microcontroller: An introduction to Software and Hardware Interfacing" by Dr. Han-Way. This book was selected because it provides the students in electronics and computer engineering technology program a broad and systematic approach to the learning of the PIC microcontroller.

Laboratory Equipment

Typically, development tools needed for microcontroller can be divided into two different groups: software and hardware. Software tools include assemblers, compilers, program editors, debuggers, simulators, communication programs, and systems integration environment to implement solutions.

Hardware can include both hardware to download code from a PC via a serial parallel, USB or Internet connection. The most basic setup requires at least a serial cable as is used in the BASIC STAMP or BASIC STAMP II from Parallax Inc. There are a number of Third Party Tools provided by such companies as micro Engineering Labs, Inc that sell download hardware and software. Micro Engineering Labs, Inc also sells a basic compiler for the PIC[4].

Microchip Technologies Inc. provides an integrated development environment that also supports several C compilers.

Hardware tools include PIC microcontroller boards for interfacing and testing. Several other tools that support troubleshooting are logic analyzers, logic probe, emulators, and oscilloscopes.

Software Tools

As mentioned above the software tools include text editors, cross assemblers, cross compilers, simulator, source level debuggers, and integrated development environment. Following paragraphs will provide a brief information about each of them.

Text Editors

The text editor is program that allows the user to enter and edit programs and text files. A simple editor can be as simple as note pad or the word pad. However, for the microcontroller programmers, neither Notepad nor WordPad is appropriate because of their limited functions.

The Microchip MPLAB Integrated Development Environment (IDE) is a free, integrated tool set for the development of embedded applications. The MPLAB IDE includes a text editor for the user to enter programs.

Cross Assemblers and Cross Compilers

The executable code placed in ROM, EPROM, EEPROM, or flash memory of a PIC18 – based controller is generated by the cross assemblers and cross compilers. The most popular companies that provide the cross assemblers and cross compilers for PIC based microcontroller are Microchip, CCS, IAR, and Hi-Tech. At Savannah State University we use the compiler provided by Custom Computer Services (CCS).

Simulator

A simulator is a program that runs on the PC and pretends to be a microcontroller chip. A simulator offers all the normal debug capability such as single stepping and looking at variables. However, there is no interaction with real hardware. The MPLAB IDE development software from Microchip includes simulators for all PIC16/PIC17/PIC18 devices.

Source-Level Debugger

A source level debugger is a program that allows the user to find problems in the code at the high level or assembly level language. A source level debugger may have the option to run your program on the demo board or simulator. In our laboratory at Savannah State University we are using In Circuit Debugging (ICD), which is a complete debugging solution for Microchip's PIC16Fxx and PIC18Fxx MCUs.

Integrated Development Environment

An integrated development environment (IDE) includes the assembler, compiler, and debugger. It includes all the aforementioned software tools. A tool is invoked by clicking on the icon of the corresponding tool. The MPLAB is an IDE designed for all Microchip microcontrollers. In our laboratory we use the PCW windows IDE developed by the CCS. The compiler with Windows IDE have full IDE with many utilities to aid in program design and editing.

Hardware Tools

Several hardware development tools are available for the PIC18 microcontrollers. Microchip provides the following hardware development tools to support the hardware development of PIC18 based products:

- In circuit emulators
- Device programmers
- Demo boards
- In-circuit debugger (ICD)

- Software debugging technique can be used to identify errors in a program. However, there are errors that cannot be identified by just examining the program. Another way of identifying errors is to use the hardware assisted approach. In this approach the designers use an in-circuit emulator (ICE) to identify program errors. An ICE allows the user to debug software before the final hardware product is available. An ICE includes a target processor module to emulate the final hardware.

Device Programmer

The software must be downloaded into the on-chip ROM of the microcontroller before it can be tested. Microchip provides the following two device programmers:

- PICSTART PLUS
- PRO MATE II

PICSTART PLUS is a low cost programmer for the PIC18 microcontroller. PICSTART PLUS is connected to a PC through the serial port and is driven by the MPLAB IDE software.

PRO MATE II is another device programmer provided by Microchip. This programmer is also connected to a PC through the serial port and controlled by the MPLAB IDE software.

In-Circuit-Debugger II

In-Circuit-Debugger II (ICD2) is designed debugging devices that support In-Circuit Serial Programming (ICSP) protocol. The ICD2 contains the logic for debugging, programming, and control. The ICD-2 can be connected to either a PC's serial port through a nine-pin serial cable or a USB cable and to the PIC18 board or target application using a six-wire modular cable.

Microcontroller Laboratory at SSU

PC-based microcontroller laboratory at Savannah State University is centered PIC18F452 microcontroller series. There are several number of prototyping boards available from a number of sources. Some have an ICD interface and others simply have a socket for a chip that is externally programmed. Some boards have some advanced functionality on the board to help design complex software. For example CCS has a prototyping board with a full 56K modem on board and a TCP/IP stack chip ready to run internet applications such as an email sending program or a mini web server. After a great deal of thoughtful consideration it was decided to acquire the Embedded C Language Development Kit developed by Custom Computer Services (CCS). The flexibility offered by this unit for interfacing provides the students with the opportunity to concentrate on designing/implementing their project with tremendous amount of ease and speed. This unit is accompanied by the following:

- CD with software including the C compiler
- Serial PC to prototype board cable
- ICD unit that allows programming and debugging
- Prototyping board with the PIC18F452 processor chip
- Serial cable, 9 pin male to female (ICD to PC)

- Breadboard for prototyping circuits

Other parts for interfacing projects includes the following:

- 93LC serial EEPROM chip
- DS1631 digital thermometer chip
- NJU6355 real time clock chip with attached 32 kHz crystal
- Two digit 7 segment LED module
- Two 1K resistors

The typical laboratory is setup consists of the hardware and software. The hardware is connected as shown in figure 1.0. The PC is connected to the ICD using a 9-pin male to female serial cable. The prototyping board with a PIC18F452 is connected to the ICD using the modular cable. ICD-U can also be used to connect to the USB port of the PC. The software is based on the Integrated Development Environment (IDE) which typically consist of the editor, viewer, and the compiler.

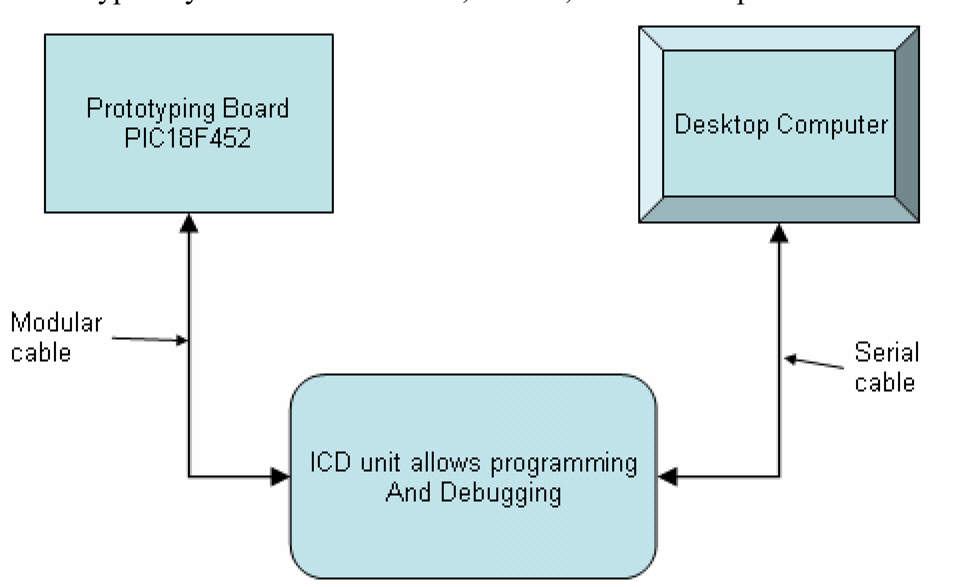

Figure 1.0 Hardware Configuration

Design Example

The following paragraphs will discuss an example, which will introduce the students to the CCS hardware and software tools.

Objectives: Upon completion of this example, you should be able to:

- Enter a simple program using the PCW IDE [2]
- Debug and Test your example
- Download and run your design on the CCS prototyping board with a PIC18F452

Problem Statement: Write a program to light the green Light Emitting Diode (LED) for 5 seconds.

Compiling, Downloading, and Running a Program

Following paragraphs will list the steps required to compile, download, and run the program in a laboratory environment.

1. Open the PCW IDE. If any files are open, click File>Close All
2. Click File>New and enter the filename GreenLED.C
3. Type in the following program and compile.
4. Use the white box on the toolbar to select the compiler. CCS offers different compilers for each family of Microchip parts. Make sure Microchip PIC 18 is selected in the white box.

```
#include        <18f452>
#device         ICD=TRUE
#fuses          HS, NOLVP, NOWDT, PUT
#use delay      (clock = 20000000)
#define         GREEN_LED PIN_A5

main    ( )     {
        while (TRUE) {
                output_low (GREEN_LED);
                delay_ms (5000);
                output_high (GREEN_LED);
                delay_ms (5000);
        }
}
```

The following paragraphs will discuss the program and the interfacing of the program with the prototyping board.

The first four lines of this program define the basic hardware environment. The chip being used is the PIC18F452, running at 20MHz with the ICD debugger. The #define is used to enhance readability by referring to GREEN_LED in the program instead of PIN_A5. For further understanding of this concept the schematic is shown in figure 2. Notice that the Green LED is connected to pin A5. The “while (TRUE)” is a simple way to create a loop that never stops. Referring to the schematic notice that the “output_low” turns the LED on because the other end of the LED is connected to +5V. This is done because the chip can tolerate more current when a pin is low than when it is high. The “delay_ms(5000)” is a 5 second delay (5000 milliseconds). The aforementioned program can be downloaded to the prototype board for testing by the steps listed below:

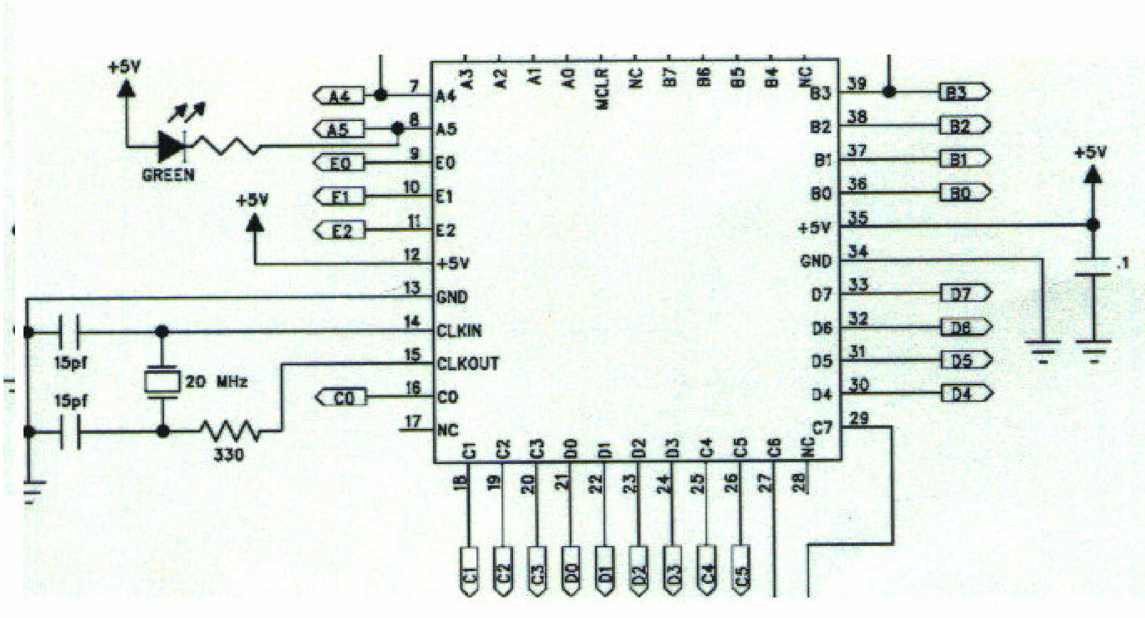

Figure 2.0 PIC Interface diagram

- Connect the ICD to the PC and the prototype board to the ICD. Power up the prototype board and verify the LED on the ICD is flashing.
- Click Debug>Enable and wait for the program to load
- Click the green go icon
- Expect the debugger window status block to urn yellow indicating the program is running
- The green LED on the prototype board should be flashing. Five second on and five second off.
- The program can be stopped by clicking on the stop icon.

Students can further expand this program to light the green LED for 5 seconds, then yellow for 1 second and the red for 5 seconds. Based on this concept students are assigned to design/implement a traffic light controller in the laboratory environment using the prototype board and the interface to the LEDs.

Student Survey:

A total of 20 surveys were distributed to the Microcontroller class. The following questions were asked in the survey:

Q1. Do you prefer the PIC microcontroller design techniques or the traditional 68HC11?
According to the response received from the students 90% of the student preferred working with the PIC microcontroller because they can devote more time on design/development. However, the students general comments were that 68HC11 also provides the basic concept needed to understand the PIC based machine.

Q2. Should this course be expanded to two semester course?

90% of the students indicated that they would like to see this course offered as Embedded Systems I and Embedded Systems II.

Q3. Are you satisfied with the text?

95% of the students are satisfied with the text "PIC Microcontroller: An introduction to Software and Hardware Interfacing" by Han-Way Huang.

Q4. Do you prefer simulator or actual interfacing?

95% of the students prefer actual interfacing. Furthermore, with the availability of low cost tools such as the ICD there is less interest in simulators.

Summary:

This paper presented a brief discussion about the Microcontroller course offered at Savannah State University. The training introduces students to the techniques for interfacing and programming the PIC18 microcontroller. This course is intended for the students in Electrical and Computer Engineering Technology who are taking the course to learn about the PIC 18 microcontroller and use it in a design project. This course provides several complicated examples to explore the functions and applications of the PIC 18 microcontroller. On the software side the students are encouraged to write C language programs to implement their applications to gain the productivity advantage and flexibility provided by the C language. The microcontroller laboratory provides the necessary hands-on experience with the PIC 18 microcontroller, which is highly desirable in industry.

Bibliography

1. Han-Way Huang., PIC Microcontroller: An Introduction to Software and Hardware Interfacing (2005).
2. Myke Predko, "Programming and Customizing PICmicro® Microcontrollers", McGraw Hill 2000
3. "PicBasic Pro® Compiler", microEngineering Labs, Inc.
4. John Iovine, "PIC Microcontroller Project Book", McGraw Hill 2000
5. John Day, AN656 "In-Circuit Serial Programming™ (ICSP™) of Calibration Parameters Using a PICmicro® Microcontroller", Microchip Technology Inc.
6. "In-Cicuit Serial Programming for PIC16F62X FLASH MCUs", Microchip Technology Inc

ASAD YOUSUF

Asad Yousuf is a Professor of Electronics Engineering Technology at Savannah State University. He received his BS in Electrical Engineering from the NED Engineering University, Karachi, Pakistan in 1980 and MS in Electrical Engineering from the University of Cincinnati in 1982 and an EdD from the University of Georgia in 1999. Asad is a registered Professional Engineer in Georgia. He is also a Microsoft Certified Systems Engineer (MCSE).

WILLIAM LEHMAN

William Lehman received his BS in Electrical Engineering from the Catholic University of America, D.C. in 1979. He has worked through the years testing software and hardware systems in the aerospace and telecommunications industries. He is currently a consultant and may be reached at Advanced Systems and Materials, Lenardtown, Maryland (water@tqci.net)

MUNIYAPPA VENKATESHA

Muniyappa Venkatesha is a head of the department of Electronics and Computer Engineering Technology at Savannah Technical College. He received his BE degree in Electrical Engineering from BDT college of Engineering Davanagere, Mysore University, India in 1967 and MS in Electrical Engineering from the North Carolina A&T University of Greensboro in 1985 and Ed.S from the Georgia Southern University in 1993.

CHUN LING HUANG

Chun Ling Huang earned B.S., and M.S degrees in Mechanical Engineering from Chung Yuan Christian University (CYCU) in Taiwan, and a PhD degree in Mechanical Engineering from the University of Alabama(UA) at Tuscaloosa. Currently, Huang is the Professor of Mechanical Engineering at Southern University, Baton Rouge, Louisiana. He is a member of ASME and ASEE.