

# PREFACE

The evolution of microcontrollers (i.e., single-chip microcomputers) has been driven by two strong trends. One trend is the demand for a low-cost means of putting intelligence into a product. When a product is produced in quantities of millions, a small savings in component cost will drive component selection decisions. Another trend is the demand for a programmable solution when an application has severe timing constraints, such as those in automobile engine control. This demand has led to ever faster, more sophisticated microcontroller architectures. Microchip Technology has evolved its product line of PIC microcontrollers from simple, low-cost devices. Subsequent generations of PIC microcontrollers have included enhancements to meet the needs of an increasing circle of applications. At the same time, Microchip has never lost sight of the value of introducing new technology at the lowest possible cost and thereby “buying” market share. As pointed out in Chapter 1 of this text, Microchip has followed this strategy to the point where it is now the number two producer of eight-bit microcontrollers in the world, with a market share that continues to rise each year. Because eight-bit microcontrollers span the needs of most applications, “eight-biters” dominate the world of microcontrollers.

This book is developed around Microchip’s latest family of parts, the PIC18FXXX family. It focuses on the PIC18F452, a new part brought to market in May 2002. It is a 40-pin microcontroller upgrade of their earlier PIC16C74 and PIC16F877 microcontrollers. This upgrade will be familiar to users of those parts, but with myriad enhancements. Virtually all of the quirks that characterized earlier PIC microcontroller parts have been eliminated. The flash program memory of the PIC18F452 microcontroller makes low-cost development possible. Not only is the part inexpensive (less than \$10 in single quantities), but it is supported by Microchip’s free assembler and by a free QwikBug monitor program. QwikBug, once programmed into the chip, supports the downloading of successive iterations of a user program and supports the ability to debug the user program with the help of running to a breakpoint, monitoring/changing watch variables, and single-stepping.

To provide a “learning by doing” environment, a QwikFlash development board is included inside the back cover of this first printing of the book. A photograph of the populated board is shown in Figure 4–1. Readers not interested in using this board may wish to know that the board is included, *gratis*, courtesy of a friend of the author. Parts for populating the board can be purchased under a special arrangement with the Digi-Key Corporation. Appendix A1 includes the Digi-Key parts list as well as construction hints, intended both for the novice who has never soldered before and for the experienced student, professional, and hobbyist. Alternatively, a populated board can be purchased from MICRODESIGNS, Inc. via their Web site, [www.microdesignsinc.com](http://www.microdesignsinc.com).

Throughout this book, the approach taken is to introduce a *template* of assembly language code that encompasses a set of features of the PIC18F452 plus its

interactions with some of the I/O devices resident on the QwikFlash board. In this way, it is intended that the reader will find a smooth path to the creative process of writing enhanced application code. There is no end to the variety of such enhancements, many of which are suggested in the end-of-chapter problems. Others are listed in Appendix A10 as a set of suggested lab projects that might accompany the first offering of a college or university course organized around this book. Still other project enhancements are suggested by the presence of the many features of the microcontroller and by the versatility of the QwikFlash board's I/O devices themselves.

Many years ago, Skip Addison impressed me with the clarity that his *structured* assembler could bring to the writing of code. Its implementation of nested IF . . . ELSE. . . ENDIF, WHILE . . . ENDWHILE, and REPEAT . . . UNTIL constructs gave assembly language the clean control flow normally associated with a compiled language such as C. At this time, Jessica Meremonte has prepared a free structured assembly preprocessor that, together with Microchip's free (and excellent) assembler, does this same thing for code writing with the PIC18F452 microcontroller. Introduced in Chapter 6, the use of Jessica's *sasm* utility acts like a one-step assembler. It enhances the code writing process, and code understandability, for the remaining 14 chapters of the book.

To utilize a microcontroller effectively, a designer should develop at least three capabilities, each of which is addressed in this book. First, he or she must understand available components. This begins with the microcontroller itself, with its CPU register structure, its instruction set, its addressing modes, and its on-chip resources. It extends to user I/O devices such as keypads and displays and sensing/control devices such as temperature transducers and stepper motor actuators. Using interrupt control, a microcontroller can juggle many real-time activities simultaneously. To achieve this without error, the designer must understand how the microcontroller handles interrupts and the timing issues related to them.

Second, the designer must thoroughly understand the algorithmic processes required by each aspect of the design and be able to translate them into the language of the microcontroller. For example, the design of an antilock brake system for an automobile involves an understanding of both the brake system dynamics and the implementation of a suitable control algorithm.

Third, the designer must understand how the extensive requirements of an instrument or device can be broken down into manageable parts. Almost any project can be likened to the process of jumping from boulder to boulder to cross a stream. Each boulder may represent the design ideas needed to understand and use a device such as a liquid crystal display. But in addition to studying boulders, the designer must pay attention to how streams are crossed. Through examples, both with the template programs and with complete designs available on the author's web site (e.g., the source code for the QwikFlash instrument described in Section 4.3), the reader can view the structure of several projects.

This book attempts to organize and unify the development of these three capabilities: to understand and use components, to exploit powerful algorithmic processes, and to break down the complexity of an instrument or device so as to meet its specifications.

## ABOUT THE BOOK

This book will typically be used in a one-semester course at the senior level. Alternatively, it might be used at the junior level if it is deemed worthwhile to trade the increased engineering experience of seniors for the opportunity to follow this course with other design-oriented courses and individual project activities. Although the context of the book is electrical, each component is sufficiently explained to permit the book to be used in a variety of curricula as an introduction to design using a microcontroller. The incentive to so use the book lies in the diverse applications made possible by the availability of a "controller on a chip."

There has been an attempt to make many parts of the book self-contained. A reader might scan Chapter 1 to gain the perspective of Steve Sanghi, CEO and President of Microchip Technology. Chapters 2 and 3 describe the PIC18F452's CPU structure, instructions, and addressing modes. The brief Chapter 4 describes the features of the QwikFlash board (i.e., the built-up version of the bare board located on the inside back cover of the book). This description includes one application of the board as a QwikFlash instrument to measure frequencies and time intervals with uncanny accuracy, using a free program available from the author's web site.

Chapters 5 and 6 present the first template of code and a discussion of development alternatives. This latter discussion is augmented in the appendices. Appendix sections A1.2 and A4.3 describe Microchip's ICD2 in-circuit debugger. Appendix A4 describes the free QwikBug on-chip monitor program and what it takes to program it into a chip. Appendix A3 describes how the free *sasm* structured assembler utility can be used with or without the support of Microchip's MPLAB facility. Appendix section A5.2 describes the free QwikAddress utility and the help it provides in conjunction with the *sasm* utility.

Chapter 7 develops the use of the liquid-crystal display built into the QwikFlash board. A second template of code is presented, serving as a stepping stone for subsequent project work. Chapters 8 and 12 augment the discussion of user I/O available on the board. Chapter 8 discusses a widely used input device, the rotary pulse generator (RPG). Chapter 12 develops what amounts to an allocation scheme for the display and the RPG, for user input.

Chapter 9 discusses the timing and the *critical region* issues associated with interrupts. It describes the high- and low-priority mechanisms built into the chip to field interrupts and explains how to allocate interrupt sources between the two so as to be most effective in meeting the timing requirements of an application.

The remaining chapters of the book can be "mixed and matched" to meet the needs of a course and its sequencing of lab projects. Chapters 13 and 16 explain in detail how to use the PIC18F452's superb timers for capturing the timing of input events and for controlling the timing for output events. Chapters 10 and 11 describe the nuances of using the chip's analog and digital input pins. Chapter 14 explains how to use any of Microchip's library of 36 integer subroutines for dealing with the multiplication and division of multiple-byte unsigned and signed numbers. It also explains how to use their eight floating-point subroutines and the role of these subroutines in retaining the precision of a computation. Chapters 15 and 17 delve into two approaches for expanding the number of I/O pins available as well as two convenient means for enhancing the peripheral capabilities of the chip beyond those peripherals already built into the chip. Chapter 18 discusses the use of the UART built into the chip while Chapter 19 looks at how a low-end PIC microcontroller can be used as a smart peripheral, connecting to the PIC18F452 through one of its UART pins. Chapter 20 discusses a variety of disparate but significant features of the chip, including the use of its non-volatile data EEPROM, its watchdog timer and brownout reset circuits, and how it can be operated to achieve an exceedingly low battery supply current.

The last two template programs, P3 and P4, are recast as C programs written by Mike Chow and included in Appendix A7. Each one is written for both Microchip Technology's C compiler and Hi-Tech Software's C compiler, two high-quality C compilers. A class that makes the switch from assembly coding can benefit from the use of the P3 template, cast in C. As a template for subsequent work, it encompasses many issues that will arise as further code is written. The P4 template illustrates how to handle interrupts.

This book is directed toward a specific goal of engineering studies—the development of creative design capability. Given a powerful, low-cost microcontroller chip, a low-cost development board, and the development tools needed to develop and debug program code, we are able to focus the design process on the microcontroller chip itself and a variety of I/O devices, thereby giving zest to the process

of learning. While some of these I/O devices are available on the QwikFlash board itself, the laboratory for a college or university course can make use of a second QwikProto board to expand the variety of I/O resources easily, as described in Appendix A2 and illustrated, by example, in the photograph of Figure A2-5. Other I/O possibilities are described there and in Appendix section A10.2. Also refer to the author's web site, [www.picbook.com](http://www.picbook.com), for URLs pointing to I/O device information and project ideas being employed at other universities. To help readers take advantage of the opportunities for developing microcontroller design capability, most chapters close with an assortment of problems, many having a design flavor. A suggested course calendar is included at the beginning of Appendix A10 along with 10 "starter" lab projects.

## ACKNOWLEDGMENTS

I have had the good fortune to have the counsel and support of Roger Webb and Bill Sayle, Chair and Associate Chair, respectively, of the School of Electrical and Computer Engineering at the Georgia Institute of Technology. At the same time, I have been aided over many years by some outstanding students who have shared my interests and passions. Chris Twigg has been prominent during this past year in a dozen endeavors involved with this book and with the switching of my lab, and its supporting utilities, over to the PIC18F452. He has developed the code presented in this book for the I<sup>2</sup>C interface of Chapter 17, for the creative Bargraph subroutine of Figure 7-17, for the transformation of the math subroutines prepared for use by Microchip's C compiler into a library of subroutines easily accessible for assembly programming use, and for the QwikPH utility for updating the *Program Hierarchy* included in an application program. He has developed the [www.picbook.com](http://www.picbook.com) web site. He, Cory Hawkins, and Paul Nichols have developed the QwikFlash instrument described in Section 4.3. Jessica Meremonte's handiwork with her *sasm* structured assembler utility reaches throughout the book. David Flowers QwikAddress utility resolves several issues arising from the use of the *sasm* utility in an elegant fashion. The team of students listed at the end of Appendix A4, culminating with the work of Burt Sims and then Rawin Rojvanit, have developed the QwikBug resident debugger utility. This combination of capabilities provides complete, and free, support for developing applications with the QwikFlash development board.

Rick Farmer and I have had a cooperative relationship in the development of PC boards for my lab over the last several years. Rick has worked with PIC microcontrollers for years and has been a productive, insightful designer for even more years. His insights through three iterations of the QwikFlash board have resulted in a fine, solid design. His original *PICloader* "load and go" resident utility for the PIC16F877 microcontroller provided the impetus for QwikBug development.

Microchip, through its ever helpful personnel have been invaluable in the development of this book about a chip yet to be introduced commercially at the time of this writing. Scott Fink's early suggestion that I learn about the forthcoming PIC18F452 lay the foundation for this project and for our development activities. Al Lovrich's support with every request armed me with the tools and the information needed to carry out this work. Carol Popovich and Paul Landino have supported my every request for the tools and the pre-production chips needed for an early switch of my instructional lab over to the PIC18F452. Craig Miller and Greg Robinson have been free with their knowledge of the chip's background debug mode (used by its in-circuit debugger) to help us develop QwikBug. Josh Conner and (outside consultant) Frank Testa aided in our acquisition and understanding of the math routines. Brett Duane helped me understand some of the issues that arose in my early experimentation with the microcontroller. Steve Sanghi and Eric Sells provided valuable material and insights for Chapter 1.

Digi-Key has helped to expedite the ordering of all the parts for stuffing the QwikFlash board. They identified parts available from Digi-Key that I had overlooked, making possible one single-source order. They set up the single identifying kit number, 18F452-KIT, that acts as an alias for the one-by-one enumeration of each part listed in Figure A1-1. Simplicity and accuracy in placing the order as well as significantly reduced parts cost are the result.

I am grateful to Bill Kaduck and Dave Cornish of MICRODESIGNS, Inc. for making a low-cost, built-up QwikFlash board available to readers of this book via their Web site, [www.microdesignsinc.com](http://www.microdesignsinc.com). See Appendix section A1.1. Doug Armstrong of AppForge has been involved with cutting-edge design work for years and has generously helped with applications for my instructional laboratory. Leland Strange of Intelligent Systems Corp. has long been a source of insight and counsel. Skip Addison's photography gives the clarity of a visual perspective throughout this book.

At Prentice Hall I have been fortunate to have had two fine editors, first Eric Frank and presently Tom Robbins. When we first discussed how the QwikFlash board might be included with the book, Eric even pursued his creative idea of die-cutting the front cover to hold the board in a transparent pocket. I am grateful to Tom Robbins for including the board with the first printing of the book, on the basis that readers would not have to bear the cost of the board in the cost of the book. With that as a given, I am grateful to Jim Carreker for his support in paying for these first-printing PC boards. As a spellbindingly energetic undergraduate, he was responsible for a major redirection in my professional life. It is a delight to me that he has once again been able to impact my life in this way.

Also, at Prentice Hall I have been fortunate to work with Lynda Castillo and David George on the production of this book. Cindy Miller of Clarinda Publications has done a fine and careful job as production editor for the book. Laserwords of Chennai, India holds my gratitude for their accurate and expeditious rendering of all my pencil-drawn figures. This is my first experience with artwork corrections being expedited by the instantaneous transfer of pdf files, and I could not be more satisfied.

Finally, on the publication of this, my sixth textbook, I am again grateful to my wife, Marilyn, for a partnership that has been in place for each book. We pace each other, with Marilyn graciously clarifying my stilted sentences as she enters handwritten text into her word processor. Throughout my career I have been fortunate that we both enjoy this form of working together.

JOHN B. PEATMAN  
*Georgia Institute of Technology*