

Programming 8-bit PIC Microcontrollers in C

with Interactive Hardware Simulation

***Programming 8-bit
PIC Microcontrollers in C***
with Interactive Hardware Simulation

Martin P. Bates



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Newnes is an imprint of Elsevier



Newnes

Newnes is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Copyright © 2008, Elsevier Ltd. All rights reserved.

Exception: Certain materials contained herein are reprinted with the permission of Microchip Technology Incorporated. No further reprints or reproductions may be made of said materials without Microchip Technology Inc.'s prior written consent.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: permissions@elsevier.com. You may also complete your request online via the Elsevier homepage (<http://www.elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."



Recognizing the importance of preserving what has been written, Elsevier prints its books on acid-free paper whenever possible.

Library of Congress Cataloging-in-Publication Data

Application submitted.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ISBN: 978-0-7506-8960-1

For information on all Newnes publications
visit our Web site at: www.books.elsevier.com

08 09 10 11 12 13 10 9 8 7 6 5 4 3 2 1

Typeset by Charon Tec Ltd., A Macmillan Company. (www.macmillansolutions.com)

Printed in the United States of America.

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER

BOOK AID
International

Sabre Foundation

Contents

Foreword	xi
Preface	xiii
Introduction	xvii

Part 1 PIC Microcontroller Systems	1
1.1 PIC16 Microcontrollers	1
MCU Features.....	3
Program Execution	3
RAM File Registers	7
Other PIC Chips.....	7
1.2 PIC16 MCU Configuration.....	8
Clock Options	9
Configuration Options	9
Configuration in C	11
1.3 PIC16 MCU Peripherals	12
Timers	13
A/D Converter.....	15
Comparator	15
Parallel Slave Port.....	16
Interrupts.....	16
1.4 PIC16 Serial Interfaces	18
USART	19
SPI Bus	21
I ² C Bus.....	21
1.5 PIC16 MPLAB Projects.....	22
MPLAB C Project	23
Project Files	25
1.6 PIC16 Program and Debug	26
Programming	27
Debugging.....	29
Design Package.....	30

Assessment 1.....	31
Assignments 1.....	32
Part 2 C Programming Essentials.....	35
2.1 PIC16 C Getting Started	35
Simple Program	35
Program Creation.....	36
Program Testing.....	37
Program Analysis.....	38
2.2 PIC16 C Program Basics.....	40
Variables	40
Looping.....	41
Decision Making.....	42
Loop Control.....	42
FOR Loop	45
SIREN Program.....	45
Blank Program.....	47
2.3 PIC16 C Data Operations	47
Variable Types.....	47
Assignment Operations.....	52
Conditional Operations.....	54
2.4 PIC16 C Sequence Control	56
While Loops	56
Break, Continue, and Goto	57
If..Else and Switch..Case	59
2.5 PIC16 C Functions and Structure	60
Basic Functions.....	62
Global and Local Variables.....	63
2.6 PIC16 C Input and Output	64
Serial LCD.....	65
Keypad and Calculator.....	68
2.7 PIC16 C More Data Types	73
Arrays	74
Indirect Addressing Operators	75
Enumeration.....	77
2.8 PIC16 C Compiler Directives.....	77
Program Directives	78
Header File	78
2.9 PIC16 C Assembler Routines.....	86
Program Compilation	87
Assembler Block.....	88
PIC Assembly Language	92

Assessment 2.....	94
Assignments 2.....	96
Part 3 C Peripheral Interfaces	99
3.1 PIC16 C Analog Input.....	99
Analog Setup	99
Voltage Measurement	100
3.2 PIC16 C Interrupts	104
C Interrupts	104
Interrupt Example	106
Interrupt Statements.....	107
3.3 PIC16 C Hardware Timers	108
Counter/Timer Operation.....	108
PWM Mode	109
Compare Mode	111
Capture Mode	111
3.4 PIC16 C UART Serial Link	113
3.5 PIC16 C SPI Serial Bus	116
3.6 PIC16 C I ² C Serial Bus.....	118
3.7 PIC16 C Parallel and Serial Interfaces.....	121
Parallel Slave Port.....	121
Comparison of Communication Links	123
3.8 PIC16 C EEPROM Interface	126
3.9 PIC16 C Analog Output	127
Assessment 3.....	132
Assignments 3.....	134
Part 4 C Mechatronics Applications	137
4.1 PICDEM Mechatronics Board Overview	137
PICDEM Hardware	137
Motor Drives.....	138
Test Program.....	143
Debugging.....	145
4.2 PICDEM Liquid Crystal Display.....	146
LCD Connections	146
LCD Test Program	149
BCD Count Program	152
4.3 PICDEM DC Motor Test Programs	152
Basic Control	152
Rev Counter	154
4.4 PICDEM Stepper Motor Control	156
Construction.....	157

	Stepper Motor Test.....	158
	Direction Control.....	158
4.5	PICDEM Analog Sensors	162
	Light Sensor.....	162
	Temperature Measurement	163
4.6	PICDEM Temperature Controller	166
	Specification	166
	I/O Allocation	168
	Implementation.....	168
4.7	PICDEM Board Simulation	171
	Circuit Description	173
	Demo Applications	175
	Assessment 4.....	175
	Assignments 4.....	177
	Part 5 PIC16 C Applications and Systems	179
5.1	PIC16 C Application Design.....	179
	Hardware Design	179
	Software Design.....	180
	Application Debugging and Testing	180
5.2	PIC16 C Temperature Controller	182
	System Operation	182
	Software Design and Implementation	183
5.3	PIC16 C Data Logger System.....	184
	BASE Board	186
	Program Outline	189
5.4	PIC16 C Operating Systems	189
	Polled I/O.....	189
	Interrupts.....	190
	PC Operating System.....	191
	Real-Time Operating System.....	192
5.5	PIC16 C System Design.....	195
	Hardware Selection.....	195
	Microcontrollers	196
	Hardware Design	197
	Software Design.....	199
	Assessment 5.....	200
	Assignments 5.....	201
	Appendix A Hardware Design Using ISIS Schematic Capture	203
	Design Specification	203
	Schematic Circuit.....	204
	Schematic Edit	205

Appendix B	Software Design Using CCS C.....	209
	BAR1 Source Code.....	211
	PIC Registers	212
	BAR1 List File.....	215
Appendix C	System Testing Using Proteus VSM.....	221
	Attaching the Program	221
	Program Debugging	221
	Typical Errors.....	224
Appendix D	C Compiler Comparison.....	227
	Microchip C18	228
	Hi-Tech PIC C.....	230
	Mikro C.....	234
	Matrix C.....	234
	Summary of C Compilers	236
Appendix E	CCS C Programming Syntax Summary.....	239
	Compiler Directives	239
	Program Blocks.....	239
	Punctuation	239
	Basic I/O Functions.....	240
Appendix F	CCS C Programming Function Reference.....	243
Answers.....		257
Index		271

Foreword

Embedded microcontrollers are everywhere today. In the average household you will find them far beyond the obvious places like cell phones, calculators, and MP3 players. Hardly any new appliance arrives in the home without at least one controller and, most likely, there will be several—one microcontroller for the user interface (buttons and display), another to control the motor, and perhaps even an overall system manager. This applies whether the appliance in question is a washing machine, garage door opener, curling iron, or toothbrush. If the product uses a rechargeable battery, modern high density battery chemistries require intelligent chargers.

A decade ago, there were significant barriers to learning how to use microcontrollers. The cheapest programmer was about a hundred dollars and application development required both erasable windowed parts—which cost about ten times the price of the one time programmable (OTP) version—and a UV Eraser to erase the windowed part. Debugging tools were the realm of professionals alone. Now most microcontrollers use Flash-based program memory that is electrically erasable. This means the device can be reprogrammed in the circuit—no UV eraser required and no special packages needed for development. The total cost to get started today is about twenty-five dollars which buys a PICkit™ 2 Starter Kit, providing programming and debugging for many Microchip Technology Inc. MCUs. Microchip Technology has always offered a free Integrated Development Environment (IDE) including an assembler and a simulator. It has never been less expensive to get started with embedded microcontrollers than it is today.

While MPLAB® includes the assembler for free, assembly code is more cumbersome to write, in the first place, and also more difficult to maintain. Developing code using C frees the programmer from the details of multi-byte math and paging and generally improves code readability and maintainability. CCS and Hi-Tech both offer free “student” versions of the compiler to get started and even the full versions are relatively inexpensive once the savings in development time has been taken into account.

While the C language eliminates the need to learn the PIC16 assembly language and frees the user from managing all the details, it is still necessary to understand the architecture. Clocking options, peripherals sets, and pin multiplexing issues still need to be solved. Martin's book guides readers, step-by-step, on the journey from "this is a micro-controller" to "here's how to complete an application." Exercises use the fully featured PIC16F877A, covering the architecture and device configuration. This is a good starting point because other PIC16s are similar in architecture but differ in terms of IO lines, memory, or peripheral sets. An application developed on the PIC16F877A can easily be transferred to a smaller and cheaper midrange PICmicro. The book also introduces the peripherals and shows how they can simplify the firmware by letting the hardware do the work.

MPLAB[®], Microchip's Integrated Development Environment, is also covered. MPLAB includes an editor and a simulator and interfaces with many compilers, including the CCS compiler used in this book. Finally, the book includes the Proteus[®] simulator which allows complete system simulation, saving time and money on prototype PCBs.

Dan Butler
Principal Applications Engineer
Microchip Technology Inc.

Preface

This book is the third in a series, including

- PIC Microcontrollers: An Introduction to Microelectronic Systems.
- Interfacing PIC Microcontrollers: Embedded Design by Interactive Simulation.
- Programming 8-bit PIC Microcontrollers in C: With Interactive Hardware Simulation.

It completes a set that introduces embedded application design using the Microchip PIC[®] range, from Microchip Technology Inc. of Arizona. This is the most popular microcontroller for education and training, which is also rapidly gaining ground in the industrial and commercial sectors. *Interfacing PIC Microcontrollers* and *Programming PIC Microcontrollers* present sample applications using the leading design and simulation software for microcontroller based circuits, Proteus VSM[®] from Labcenter Electronics. Demo application files can be downloaded from the author's support Web site (see later for details) and run on-screen so that the operation of each program can be studied in detail.

The purpose of this book is to

- Introduce C programming specifically for microcontrollers in easy steps.
- Demonstrate the use of the Microchip MPLAB IDE for C projects.
- Provide a beginners' guide to the CCS PCM C compiler for 16 series PICs.
- Explain how to use Proteus VSM to test C applications in simulated hardware.
- Describe applications for the Microchip PICDEM mechatronics board.
- Outline the principles of embedded system design and project development.

C is becoming the language of choice for embedded systems, as memory capacity increases in microcontrollers. Microchip supplies the 18 and 24 series chips specifically designed for C programming. However, C can be used in the less complex 16 series PIC, as long as the applications are relatively simple and therefore do not exceed the more limited memory capacity.

The PIC 16F877A microcontroller is used as the reference device in this book, as it contains a full range of peripherals and a reasonable memory capacity. It was also used in the previous work on interfacing, so there is continuity if the book series is taken as a complete course in PIC application development.

Microcontrollers are traditionally programmed in assembly language, each type having its own syntax, which translates directly into machine code. Some students, teachers, and hobbyists may wish to skip a detailed study of assembler coding and go straight to C, which is generally simpler and more powerful. It is therefore timely to produce a text that does not assume detailed knowledge of assembler and introduces C as gently as possible. Although several C programming books for microcontrollers are on the market, many are too advanced for the C beginner and distract the learner with undesirable detail in the early stages.

This text introduces embedded programming techniques using the simplest possible programs, with on-screen, fully interactive circuit simulation to demonstrate a range of basic techniques, which can then be applied to your own projects. The emphasis is on simple working programs for each topic, with hardware block diagrams to clarify system operation, full circuit schematics, simulation screenshots, and source code listings, as well as working downloads of all examples. Students in college courses and design engineers can document their projects to a high standard using these techniques. Each part concludes with a complete set of self-assessment questions and assignments designed to complete the learning package.

An additional feature of this book is the use of Proteus VSM (virtual system modeling). The schematic capture component, ISIS, allows a circuit diagram to be created using an extensive library of active components. The program is attached to the microcontroller, and the animated schematic allows the application to be comprehensively debugged before downloading to hardware. This not only saves time for the professional engineer but provides an excellent learning tool for the student or hobbyist.

Links, Resources, and Acknowledgments

Microchip Technology Inc. (www.microchip.com)

Microchip Technology Inc. is a manufacturer of PIC[®] microcontrollers and associated products. I gratefully acknowledge the support and assistance of Microchip Inc. in the development of this book and the use of the company trademarks and intellectual property. Special thanks are due to John Roberts of Microchip UK for his assistance and advice. The company Web site contains details of all Microchip hardware, software, and development systems. MPLAB IDE (integrated development system) must be downloaded and installed to develop new applications using the tools described in this book. The data sheet for the PIC 16F877A microcontroller should also be downloaded as a reference source.

PIC, PICmicro, MPLAB, MPASM, PICkit, dsPIC, and PICDEM are trademarks of Microchip Technology Inc.

Labcenter Electronics (www.labcenter.co.uk)

Labcenter Electronics is the developer of Proteus VSM (virtual system modeling), the most advanced cosimulation system for embedded applications. I gratefully acknowledge the assistance of the Labcenter team, especially John Jameson, in the development of this series of books. A student/evaluation version of the simulation software may be downloaded from www.proteuslite.com. A special offer for ISIS Lite, ProSPICE Lite, and the 16F877A simulator model can be found at www.proteuslite.com/register/ipmbundle.htm.

Proteus VSM, ISIS, and ARES are trademarks of Labcenter Electronics Ltd.

Custom Computer Services Inc. (www.ccsinfo.com)

Custom Computer Services Inc. specializes in compilers for PIC microcontrollers. The main range comprises PCB compiler for 12-bit PICs, PCM for 16-bit, and PCH for the 18 series chips. The support provided by James Merriman at CCS Inc. is gratefully acknowledged. The manual for the CCS compiler should be downloaded from the company Web site (Version 4 was used for this book). A 30-day trial version, which will compile code for the 16F877A, is available at the time of writing.

The Author's Web Site (www.picmicros.org.uk)

This book is supported by a dedicated Web site, www.picmicros.org.uk. All the application examples in the book may be downloaded free of charge and tested using an evaluation version of Proteus VSM. The design files are locked so that the hardware configuration cannot be changed without purchasing a suitable VSM license. Similarly, the attached program cannot be modified and recompiled without a suitable compiler license, available from the CCS Web site. Special manufacturer's offers are available via links at my site. This site is hosted by www.larrytech.com and special thanks are due to Gabe Hudson of Larrytech[®] Internet Services for friendly maintenance and support.

I can be contacted at the e-mail address martin@picmicros.org.uk with any queries or comments related to the PIC book series.

Finally, thanks to Julia for doing the boring domestic stuff so I can do the interesting technical stuff.

About the Author

Martin P. Bates is the author of *PIC Microcontrollers*, Second Edition. He is currently lecturing on electronics and electrical engineering at Hastings College, UK. His interests include microcontroller applications and embedded system design.

Introduction

The book is organized in five parts. Part 1 includes an overview of the PIC microcontroller internal architecture, describing the features of the 16F877A specifically. This chip is often used as representative of the 16 series MCUs because it has a full range of peripheral interfaces. All 16 series chips have a common program execution core, with variation mainly in the size of program and data memory. During programming, certain operational features are configurable: type of clock circuit, watchdog timer enable, reset mechanisms, and so on. Internal features include the file register system, which contains the control registers and RAM block, and a nonvolatile EEPROM block. The parallel ports provide the default I/O for the MCU, but most pins have more than one function. Eight analog inputs and serial interfaces (UART, SPI, and I²C) are brought out to specific pins. The hardware features of all these are outlined, so that I/O programming can be more readily understood later on. The application development process is described, using only MPLAB IDE in this initial phase. A sample C program is edited, compiled, downloaded, and tested to demonstrate the basic process and the generated file set analyzed. The debugging features of MPLAB are also outlined: run, single step, breakpoints, watch windows, and so on. Disassembly of the object code allows the intermediate assembly language version of the C source program to be analyzed.

Part 2 introduces C programming, using the simplest possible programs. Input and output are dealt with immediately, since this is the key feature of embedded programs. Variables, conditional blocks (`IF`), looping (`WHILE`, `FOR`) are quickly introduced, with a complete example program. Variables and sequence control are considered in a little more detail and functions introduced. This leads on to library functions for operating timers and ports. The keypad and alphanumeric LCD are used in a simple calculator program. More data types (long integers, floating point numbers, arrays, etc.) follow as well as assembler directives and the purpose of the header file. Finally, insertion of assembler into C programs is outlined.

Part 3 focuses on programming input and output operations using the CCS C library functions. These simplify the programming process, with a small set of functions usually providing all the initialization and operating sequences required. Example programs for analog input and the use of interrupts and timers are developed and the serial port functions demonstrated in sample applications. The advantages of each type of serial bus are compared, and examples showing the connection of external serial EEPROM for data storage and a digital to analog converter output are provided. These applications can be tested in VSM, but this is not essential; use of VSM is optional throughout the book.

Part 4 focuses specifically on the PICDEM mechatronics board from Microchip. This has been selected as the main demonstration application, as it is relatively inexpensive and contains a range of features that allow the features of a typical mechatronics system to be examined: input sensors (temperature, light, and position) and output actuators (DC and stepper motor). These are tested individually then the requirements of a temperature controller outlined. Operation of the 3.5-digit seven-segment LCD is explained in detail, as this is not covered elsewhere. A simulation version of the board is provided to aid further application design and implementation.

Part 5 outlines some principles of software and hardware design and provides some further examples. A simple temperature controller provides an alternative design to that based on the mechatronics board, and a data logger design is based on another standard hardware system, which can be adapted to a range of applications—the BASE board. Again, a full-simulation version is provided for testing and further development work. This is followed by a section on operating systems, which compares three program design options: a polling loop, interrupt driven systems, and real-time operating systems. Consideration of criteria for the final selection of the MCU for a given application and some general design points follow.

Three appendices (A, B, and C) cover hardware design using ISIS schematic capture, software design using CCS C, and system testing using Proteus VSM. These topics are separated from the main body of the book as they are related more to specific products. Taken together, MPLAB, CCS C, and Proteus VSM constitute a complete learning/design package, but using them effectively requires careful study of product-specific tutorials. VSM, in particular, has comprehensive, well-designed help files; and it is therefore unnecessary to duplicate that material here. Furthermore, as with all good design tools, VSM evolves very quickly, so a detailed tutorial quickly becomes outdated.

Appendix D compares alternative compilers, and application development areas are identified that would suit each one. Appendix E provides a summary of CCS C syntax

requirements, and Appendix F contains a list of the CCS C library functions provided with the compiler, organized in functional groups for ease of reference. These are intended to provide a convenient reference source when developing CCS C programs, in addition to the full CCS compiler reference manual.

Each part of the book is designed to be as self-contained as possible, so that parts can be skipped or studied in detail, depending on the reader's previous knowledge and interests. On the other hand, the entire book should provide a coherent narrative leading to a solid grounding in C programming for embedded systems in general.