EXPLORING C FOR MICROCONTROLLERS

Exploring C for Microcontrollers

A Hands on Approach

JIVAN S. PARAB
Goa University
India

VINOD G. SHELAKE
Shivaji University
Kolhapur, India

RAJANISH K. KAMAT
Shivaji University
Kolhapur, India

and

GOURISH M. NAIK
Goa University
India

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 978-1-4020-6066-3 (HB)
ISBN 978-1-4020-6067-0 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

Contents

Foreword ix
Preface xiii
Acknowledgments xvii

1

Microcontrollers: Yesterday, Today, and Tomorrow 1

1.1 Defining Microcontrollers 1
1.2 Eagle's View: Microcontrollers and Other Competing Devices 2
1.3 Vignettes: Microcontrollers 3
1.4 Microcontroller Applications 5
1.5 Growth Economics 7
1.6 The Major Players in the Microcontroller Chip Market 8
1.7 Architectural Trends 10
1.8 Jump Starting Your First Embedded System Project 14
1.9 Execution of Embedded System Project: Is it a Gordian's Knot? 16

2

Integrated Development Environment 19

2.1 Getting Familiar with the IDE 19
2.2 Working with Keil IDE 20
2.3 Development Flow for the Keil IDE 21

3

Art of C Programming for Microcontrollers 29

3.1 Familiarizing with Your Compiler Capabilities 29
3.2 Whether to Use Headers or Not? 30
3.3 Basic C Program Structure 34
3.4 Differences from ANSI C 34

4

Exploring the Capabilities of On-Chip Resources Programming for I/O Ports, Interrupts and Timer/Counter 37

4.1 Importance of Ports 37
4.2 Simple Ideas for Port Expansion 46
4.3 LED Interfacing 46
4.4 Relevance of LEDs in Today's Lightening Industry 48
4.5 Different Programs for LED Interfacing 48
4.6 More Projects on LED Interfacing to Microcontrollers 51
4.6.1 Running LEDs 51
4.6.2 Running Bicolor LED 51
4.6.3 Interfacing 6 LEDs Using 3 Microcontroller Pins by 51
4.6.4 The LED Dimmer Demoboard 51
4.6.5 Fading RGB LED 52
4.6.6 LED Moving Font 52
4.7 DIP Switch Interfacing: Getting Input to Your Embedded System 52
4.8 LCD Interfacing 54
4.9 Interrupts in Microcontrollers 57
4.9.1 Writing ISRs in C 58
4.9.2 A Word about Interrupt Latency 63

5

Hyperterminal-Based Control 69

5.1 Hyperterminal 69
5.2 Packet-based Control Scheme 69
5.3 Mechanism and Lots of Possibilities 70
5.4 Application 1: Packet-based Interface for LEDs 71
5.5 Application 2: Packet-based Interface for Stepper Motor Control 74
5.6 Application 3: Home Automation from PC HyperTerminal 78

6

Embedding Microcontroller in Routine Applications 85

6.1 Application 1: Podium Timer 85
6.2 Application 2: Front Desk Notifier 90
6.3 Application 3: Cafeteria Food Alert/Microcontroller-based Menu Card 94
6.4 Application 4: Chimney Sentinel 100
6.5 Application 5: Who's First Timer 104

6.6 Application 6: Counting Cars 110
6.7 Application 7: Anonymous Voting 114
6.8 Energy Efficient Lighting Using Microcontroller 119

7

Microcontroller-based Measurement and Control Applications 123

7.1 Application 1: Reading a PWM Waveform Using Microcontroller 123
7.2 Single Set-point On/Off Controller 127
7.3 Application 3: I^2C Interface with Serial EPROM 133

8

Securing Your Embedded System Application 139

8.1 Security Challenges in Embedded Systems 139
8.2 Application 1: Authentication for Your Embedded System Application 140
8.3 Application 2: Timeout Waiting for Input Data 144

References 151

Index 155

Foreword

If we accept the premise that an embedded engineer is made rather than born, then how does one go about making a good one? The authors of this book *Exploring C for Microcontrollers: A Hands-on Approach* are certainly “good ones”. Not only do they explore some of the influences that shaped themselves but they also try to shape “would-be” embedded engineers. Research and developmental activities in embedded systems has grown in a significant proportion in the recent past. Embedded software design is not new to the world, but with the changing time, it has gained considerable momentum in the recent past, and many young engineers are strongly inclined to pursue their future in this field. The book is mainly targeted to these engineers who would like to understand in great depth the synergetic combination of hardware and software.

The book is divided into eight chapters. Chapter 1 introduces a brief background about micro-controllers and explains how they are embedded into products commercially available in the market to emphasize the importance of these in the daily life of mankind. It also gives an insight into the architectural details and embedded system concepts for students’ projects to motivate them into this exciting field. The rest of the book concentrates on software development. The integrated development environment (IDE) is introduced in Chapter 2. Again the screen shots and step-by-step procedure will certainly make the students and engineers fully understand the development process. Chapter 3 differentiates the embedded C paradigm from the conventional ANSI C. Again the authors explain how to successfully overcome the memory and time constraints while developing an embedded C program. Chapter 4 gives an overview of program development for on-chip resources for MCS51 family of microcontrollers. Chapters 5–8 are devoted to live case studies.

The book has come out with an elegant presentation to aspiring students and engineers from the teaching experience and technical knowledge the authors have put over a long time in this field. I strongly recommend this book for intermediate programmers, electronics, electrical, instrumentation engineers or any individual who is strongly inclined to take up his or her career in embedded C programming. I am sure the reader will experience learning embedded programming by example and learning by doing. Last but not the least, this book will certainly be a value addition to the world of embedded programming.

Dr. A. Senthil Kumar
Head
Data Quality Evaluation
National Remote Sensing Agency
Department of Space
Government of India

Dr. Senthil Kumar is Head of DQE and PQCD sections of National Remote Sensing Agency (NRSA) an autonomous operational center under Department of Space (DOS), Government of India. This is the nodal agency in the country for receiving, processing, and distributing the satellite and aerial remote sensing data and products. NRSA is also responsible for providing end-to-end solutions for utilization of data for geospatial applications and information services. NRSA has a huge archive of remote sensing data acquired through Indian and foreign satellites and also has the capability to acquire data pertaining to any part of the globe on demand. It is one of the important centers for promotion of remote sensing and geographic information system technologies in India. NRSA has set up satellite data processing facilities starting from data reception to utilization at various centers within India and across the globe.

Preface

The past few decades have witnessed evolution of microcontrollers. They have revitalized a number of products or equipment in almost all fields including telecommunications, medical, industrial, and consumer products. These embedded microcontroller systems now resides at the heart of modern life with a variety of applications in fields like consumer electronics, automotive systems, domestic, and even in aerospace products. Embedding a microcontroller in an electronics instrument or product requires a specialized design skill which requires a synergy of hardware and software skills.

In our day-to-day life we come across a number of embedded products. When we switch on the washing machine or send an SMS on a cell phone one cannot prevent without thinking the mechanism and the co-working of hardware and software in the background. The market for such smart embedded products is occupying newer and newer applications seemingly impossible few years back. Last year the IDC, a premier global market intelligence firm, revealed that the embedded industry product development was expected to be as high as $75 billion. This entails the industry requirement of trained human resource with mixed skill set both in hardware and software. Unfortunately the synergetic demand of hardware and software or sometimes even referred to as firmware competency has led to a supply–demand gap of HR in this field. This gap expressed in numerical figures led to requirement of around 150,000 embedded engineers to serve the global embedded industry. This book is ideal for all those who would like to pursue their career in the exciting world of microcontroller-based embedded systems. The approach is pedagogical; first the hardware module is presented and then the associated software code in Keil C.

The hardware designed is useful for engineering graduates and practicing professionals with the required knowledge and practical hands on skills to design with embedded systems. However, the prerequisite for the book is background of theoretical aspects of architecture of microcontrollers especially the MCS-51 family. The book starts with initial experiments, which provide familiarization with the capabilities and the limitations of the basic 8051 microcontroller using a simulator. Once the reader is comfortable with these primitive programs which covers almost all the on-chip resources, he or she can switch to more advanced ones.

The Scope of the Book

We now review the topics covered in sequence, chapter by chapter.

Chapter 1 provides an overview of microcontrollers and their applications in different domains. The architectural trends and the growth economics emphasizes the importance of the subject. The photograph of the setup and the hints toward project execution will definitely boost the confidence of the novice to kick-start the project with minimal resources.

Chapter 2 is devoted to the IDE for the MCS-51 family. The simulation and single stepping as described in this chapter will solve all the project intricacies of the readers. Chapter 3 illustrates the basic difference in traditional C programming and embedded C. Chapter 4 deals with the programming of on-chip resources of MCS-51 family microcontrollers in C. The theoretical details of these on-chip resources such as ports, timers, etc., are completely eliminated. As the book aims at hands-on approach, the programs for the on-chip resources have been developed and their execution is illustrated in the Keil simulation environment.

The last four chapters, i.e., 5–8 deal with various project case studies. Several case studies in various application domains such as lighting, measurement and control, security, and domestic applications are developed from scratch. The hardware and software developed in the form of case studies also caters to a set of mini projects, which are discussed in detail from the design phase to the actual implementation on a target system. There are 17 case studies given in this book on various systems that you may encounter in day-to-day life. Overall the hardware and software developed in this book can be reused for any embedded system project and is expected to act as a rapid prototyping unit for the embedded systems industry.

Reasons for Proposing this Book

The market is flooded with a number of good books on embedded systems designed especially with the most popular MCS-51 family. These books are traditional in nature, i.e., they start with the routine architectural features of 8051, description of registers, ports, interrupts, etc. Most of these are already covered in the device data sheet and application notes. In the present book all such routine features are skipped. The focus is on programming microcontrollers to be specific MCS-51 family in 'C' using Keil IDE. The book presents 20 live case studies apart from the many basic programs organized around every on-chip resource like port, time/counter, interrupt, serial I/O, etc. Rather than introducing the underpinning theory or reproducing lengthy data sheets, our approach is "learning-through-doing" and one that appeals to busy electronics designers. The 'C' codes given are well supported by easy-to-understand comments wherever required. Mastering the basic modules and hands-on working with the projects will enable the reader to grasp the basic building blocks for most 8051 programs. Whether you are a student using the MCS-51 family of microcontrollers for your project work or an embedded systems programmer, this book will give you a kick start in using and understanding the most popular microcontroller.

Authors through their interaction with the undergraduate and post-graduate students as well as industry professionals have found that such a book is the need of the microcontroller community interested in C programming. The book will bridge the gap between the microcontroller hardware experts and the C programmers.

Major Features

The objective of this book is to introduce the readers to the design and implementation of an embedded system. It covers the unique requirements and limitations of operating in an embedded environment. It also covers microcontrollers as the most widespread example of embedded systems. In particular, it focuses on the MCS-51 family of microcontrollers, their programming in C language, and interfacing techniques.

Special emphasis is to provide hands-on experience for the readers using a hardware and interfacing modules described in this book. The aim is to empower the reader to actually solve his or her problem with a practical hands-on pedagogy through the hardware and software presented in this book. The principle of "Design Reuse" is explained effectively.

Further, the readers will also learn how to follow the sequence of data flow through the microcontroller when a program is executed. Additionally, the readers will learn the operation of the microcontroller's I/O functions and the external devices driven by the microcontroller. Hardware and software design issues are discussed for specific systems implemented using MCS-51 as the embedded microcontroller.

Acknowledgments

We would like to take the opportunity to thank all those who have contributed or helped in some way in the preparation of this text. Particular thanks must go to our heads of the institutions – Professor M.M. Salunkhe, Vice Chancellor, Shivaji University, Kolhapur, India, and Professor P.S. Zacharias, Vice Chancellor, Goa University, Goa, India – for the encouragement and support. We would also like to thank Mr. P. Venugopal, Director, Software Technology Parks of India, Maharashtra Region, for his support. Dr. Kamat and Dr. Naik would like to thank their respective wives for their understanding and patience shown when the preparation of the book took time which could have been spent with the family. Our thanks then to Kamat's wife Rucha and Naik's wife Deepa.

Additionally, Mr. Shelake and Mr. Parab would like to express gratitude to their parents for their encouragement and support over the years. Kamat would like to dedicate his contribution to this book to the memory of the late Professor G.G. Tengshe and the late Dr. V. Rao Indolkar, ACD Machine Control Tools Ltd., Mumbai.

Dr. Kamat would also like to thank to his teacher in this field Mr. S. Ramgopal, Indian Institute of Science, Bangalore, and Dr. Senthil Kumar, Dr. Raghurama, Deputy Director (Academic) of BITS Pilani as well as Mr. K.S. Deorukhkar for help in reviewing and critical suggestions. The past batches of M.Sc. Electronics students of both Shivaji University and Goa University especially Mr. Roy, Mr. Rupesh from Satyam Computers must be thanked for generation of problems for programs developed in this book.

- Jivan S. Parab
- Vinod G. Shelake
- Dr. Rajanish K. Kamat
- Dr. Gourish M. Naik

Chapter 1

Microcontrollers: Yesterday, Today, and Tomorrow

1.1 Defining Microcontrollers

It is said that from the definition everything true about the concept follows. Therefore, at the outset let us take a brief review of how the all-pervasive microcontroller has been defined by various technical sources.

A microcontroller (or MCU) is a computer-on-a-chip used to control electronic devices. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (the kind used in a PC). A typical microcontroller contains all the memory and interfaces needed for a simple application, whereas a general purpose microprocessor requires additional chips to provide these functions.…(Wikipedia [1])

A highly integrated chip that contains all the components comprising a controller. Typically this includes a CPU, RAM, some form of ROM, I/O ports, and timers. Unlike a general-purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task – to control a particular system. As a result, the parts can be simplified and reduced, which cuts down on production costs. …(Webopedia [2])

A {microprocessor} on a single {integrated circuit} intended to operate as an {embedded} system. As well as a {CPU}, a microcontroller typically includes small amounts of {RAM} and {PROM} and timers and I/O ports.…(Define That [3])

A single chip that contains the processor (the CPU), non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. …(PC Magazine [4])

A microprocessor on a single integrated circuit intended to operate as an embedded system. As well as a CPU, a microcontroller typically includes small amounts of RAM and PROM and timers and I/O ports. …(FOLDOC [5])

J.S. Parab et al. (eds.), Exploring C for Microcontrollers, 1–18.

The definitions given by various sources describe microcontroller as an integrated circuit (IC) with processor as well as peripherals on chip. But the crux of the matter is the widespread uses of microcontrollers in electronic systems. They are hidden inside a surprising number of products such as microwave oven, TV, VCR, digital camera, cell phone, Camcorders, cars, printers, keyboards, to name a few.

The last three decades and especially the past few years have witnessed the tremendous growth in the top-of-the-line processors used for personal computing and digital signal processor (DSP) applications. Today, the use of microcontrollers in embedded systems outnumbers the use of processors in both the PC and the workstation market. It is estimated that over 50% of the vast majority of the computer chips sold are microcontrollers. The main reasons behind their huge success are powerful yet cleverly chosen customizable electronics, ease in programming, and low cost. These days microcontrollers find increasing application even in areas where they could not be used previously. With the decreasing costs and footprints, it is now possible to have small-sized and cost-effective microcontroller units (MCUs) for new applications. The microcontrollers today are small enough to penetrate into the traditional market for 4-bit applications like TV remote controls, toys, and games. For the simplest of all control applications they can offer high value smart switch functionality for applications that previously used electromechanical devices. Also the customers now can add intelligence to their end products for low cost as per the microcontroller market report by Frost & Sullivan research service [6].

1.2 Eagle's View: Microcontrollers and Other Competing Devices

Generally the technical fraternity try to compare the various devices like microprocessors, PCs, microcontrollers, DSPs, and reconfigurable devices like FPGAs and CPLDs. An interesting point to note is that embedded systems are made using all the above-mentioned devices except PC owing to its general purpose architecture. As programmability is the common feature of all these devices, they have their firm footing in different application domains. On one side of the spectrum, microcontroller-based embedded system design emphasizes on task-specific dedicated applications at low power, low cost, high throughput, and highest reliability. On the other extreme of the spectrum, FPGA-based embedded systems dominate their custom computing architectures. Unlike microcontrollers, these systems can be reconfigured on the fly to suit the application with higher computational density and throughput. With the proliferation of density, FPGA-based embedded systems offer higher performance with the only challenging issue of memory required to store their configurations.

The technical community also tends to associate various characteristics of embedded systems with microprocessors and microcontrollers. The microprocessors are typically found to dominate the desktop arena focusing on more and more bit processing capability, with other features such as fairly good amount of cache with memory management and protection schemes supported by the operating system. Although microcontrollers share flexibility aspect of microprocessors through programming, 8-bit versions are more in use (although 16- and 32-bit exist) with RAM and ROM (instead of cache) added with many on chip peripherals like timer/counter, decoder, communication interface, etc. In the literature many embedded systems products have been reported as microprocessors. On the other side of the processor spectrum, a DSP possesses special architecture that provides ultra-fast instruction sequences, such as shift and add, multiply and add, which are commonly used in math-intensive signal processing applications. The common attributes associated with the DSPs are multiply-accumulate (MAC) operations, deep pipelining, DMA processing, saturation arithmetic, separate program and data memories, and floating point capabilities required most of the time. However, the line of differentiation between all these devices is getting blurred at a rapid pace. With the introduction of fuzzy logic, artificial intelligence and networked communication capabilities in the consumer products like refrigerators, mobile phones, and cars, convergence of the architectures of most of the above-mentioned programmable devices is witnessed by the industry. Today's ideal microcontroller is expected to offer plenty of MIPS, run the DSP programs with the same speed of the DSP processor, integrate all its peripherals and support flash, communicate with the world with I2C or CAN protocols, withstand extremes of environment in a car engine, and cost but a few cents.

1.3 Vignettes: Microcontrollers

It is interesting to note that the development of microprocessors seems to be an accident out of the microcontroller synthesis. In 1969, Busicom, a Japanese company, approached Intel to convert their special purpose ROM and shift register–based calculator cores into a specialized application specific processor. The objective was the development of microcontrollers rather than a general purpose of CPU chips for keyboard scanning, display control, printer control, and other functions for a Busicom-manufactured calculator. However, the Intel engineers opted for a more flexible programmable microcomputer approach rather than the random logic chip-set originally envisioned by Busicom. The four designs [7] proposed by Federico Faggin, Ted Hoff, and Stan Mazor from Intel were a 2048-bit ROM with a 4-bit programmable input/output port (4001); a 4-registers $\times$ 20-locations $\times$ 4-bit RAM data memory with a 4-bit output port (4002); an input/output (I/O) expansion chip, consisting of a static shift register with serial input and serial/parallel output (4003); and the 4-bit CPU chip (4004). The 4001, 4002, and 4003 are very close to microcontroller kind of architecture rather than microprocessor. However, the Intel 4004, which was supposed to be the brains of a calculator, turned out to be a general-purpose microprocessor as powerful as ENIAC. The scientific papers and literature published around 1971 reveal that the MP944 digital processor used for the F-14 Tomcat aircraft of the US Navy qualifies as the "first microprocessor". Although interesting, it was not a single-chip processor, and was not general purpose – it was more like a set of parallel building blocks you could use to make a special purpose DSP form [8]. It indicates that today's industry theme of converging DSP-microcontroller architectures was started in 1971.

The other companies were also catching up at the same time. The first official claim of filing the patent goes to Texas Instruments under the trade name TMS1000 way back in 1974. This was the first microcontroller which included a 4-bit accumulator, 4-bit Y register and 2- or 3-bit X register, which combined to create a 6- or 7-bit index register for the 64 or 128 nibbles of on-chip RAM. A 1-bit status register was used for various purposes in different contexts. This microcontroller served as the brain of the Texas Instrument's educational toy named "Spark and Spell" shown in the movie *ET: The Extraterrestrial.* In 1976, Intel introduced the first 8-bit microcontroller family MCS-48 which was so popular that they could ship 251,000 units in that year. After four years of continuous research, the MCS-48 family was upgraded to 8051, an 8-bit microcontroller with on-board EPROM memory. Intel shipped 22 million pieces in 1980. The market requirement was so much that the total units sold in three years later were 91 million. The year 2005 is a special one for the microcontroller 8051. It has celebrated its 25th anniversary. But, also in 2005, Intel notified they would discontinue all automotive versions of their microcontrollers, including 8051. Car engine control units were once perhaps the most prominent application for 8051s. This means only one thing, Intel gives up the microcontrollers for good. This is confirmed by product change notification published in early 2006, announcing that Intel drops its whole microcontroller business [9].

1.4 Microcontroller Applications

The microcontroller applications are mainly categorized into the following types (see Figure 1.1):

- Audio
- Automotive
- Communication/wired
- Computers and peripherals
- Consumer
- Industrial
- Imaging and video
- Medical
- Military/aerospace
- Mobile/wireless
- Motor control
- Security
- General Purpose
- Miscellaneous

Automobile industry is the main driving force in propelling the growth of microcontrollers. It is estimated that the microcontrollers constitute 33% of the semiconductors used in a vehicle [10]. Requirements of the automobile sector has forced the microcontroller manufacturers to come out with the new bus protocols such as control area network (CAN) and local interconnect network (LIN). Microcontrollers of all bit cores are used in vehicles according to the Frost & Sullivan Industry report. The 8- and 16-bit microcontrollers are used for low-end applications and lower-cost vehicles while the 32-bit microcontrollers are used for high-end application and high-end vehicles. It is estimated that currently 30–40 microcontrollers are used in low-end vehicles and about 70 microcontrollers are used in high-end vehicles. These requirements are continuously increasing and it is highly likely that the count of microcontrollers in vehicles will further increase in the future, quotes World Microcontrollers Market Report by Frost & Sullivan [10].