C for Electronics and Computer Engineering Technology

C for Electronics and Computer Engineering Technology

PETER J. HOLSBERG
Mercer County Community College

PRENTICE HALL, *Englewood Cliffs, New Jersey 07632*

Library of Congress Cataloging-in-Publication Data

Holsberg, Peter J.
C for electronics and computer engineering technology / Peter J.
Holsberg.
p. cm.
ISBN 0-13-109703-2
1. C (Computer program language) I. Title.
QA76.73.C15C164 1990
005.13'3—dc20

Editorial/production supervision and
interior design: Maria McColligan
Cover design: Wanda Lubelska
Manufacturing buyer: Gina Chirco-Brennan

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0-13-109703-2

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Preface

Since their initial appearance in the late 1970s, microprocessors have revolutionized the field of electronics. Because they are general-purpose logic elements that can be programmed to do different tasks in different control systems, they have become the design tool of almost every electronics designer. For this reason, today's electronics engineering technology and computer engineering technology student studies microprocessor-based systems in detail.

While the microprocessor integrated-circuit chip has the advantage of being able to replace many individual, less capable chips, it has caused engineers, technologists, and technicians to need to know something about programming. Programability is the feature of the microprocessor that permits it to perform differently in different systems.

For ten years or more, engineers, technologists, and technicians wrote control programs in "assembly language," a microprocessor-specific language that permits control of the microprocessor hardware. However, as systems design became more complex and the number of different kinds of microprocessors grew rapidly, practitioners realized that assembly language programming had become very expensive on large complex systems. Further, maintenance of the programs for that system became more difficult if the original programmer-engineers had changed jobs. So they began to look for so-called "higher-level languages" that would give them the control over the microprocessor that they had with assembly language, but would allow programming to be easier to do and to maintain than assembly language programs. They found C.

C is a general-purpose programming language that has modern data structures and control flow, and also allows programming at the "bit level" so necessary for microprocessor-based control systems. It was developed in the 1970s and has become available on computers of every conceivable size in the last five years. This book is intended to teach you enough to make you familiar with the so-called "personal computer," the art of programming to solve problems, and some of the fundamental features of the C programming language. When you finish, you will know enough to be able to solve the kinds of technical problems you face in the first two years of your professional education. If you would like to become a truly proficient C programmer, you will need to take a traditional "Programming in C" course.

I would like to acknowledge the help I received from reviewers, both professional and amateur alike. I was flattered by their kind remarks and pleased with the detailed suggestions they made. I would especially like to thank a young programmer—Lisa Holsberg, my daughter—and my two EE135 classes of the fall 1988 semester. And I would be remiss in not mentioning the encouragement I received from my sons, Alan Holsberg and Bill Vandegrift, and my wife, Cathy Ann Vandegrift. It is to her I dedicate this book. To all, I say a sincere "Thanks!".

Peter J. Holsberg

Contents

1 Computers and Programming

1-1 INTRODUCTION AND OBJECTIVES

The purpose of this book is to help you learn enough about computers and programming so that you will be able to write simple programs that help you solve problems as you study electronics engineering technology (EET) or computer engineering technology (CET). The skills you learn will be valuable to you in your career, because most engineers and technologists are involved with computers in their jobs. One of the things they may be involved in is programming. Many people who started out with a strong interest in computer hardware and electronics circuits have become interested in writing programs because of the kinds of challenges and rewards that are associated with software.

Programs fall generally into two broad categories: *systems* and *applications*. Systems programs include things you have probably not heard of because they are the programs that make a computer system easier for us to use. They include text editors, operating systems, input/output schedulers, compilers, interpreters, assemblers, and more. Applications programs include most of the programs that you have heard of or have even used: games, word processors, banking systems, store credit card systems, income tax programs, inventory programs, and so on, and what engineers and technologists call *control programs*. Control programs are what engineers and technologists write to control systems—from relatively common things such as microwave ovens, TV sets, and VCRs to exotic items such as automotive antiskid brakes and aircraft navigation systems.

An Electronic System

As an example of a system that uses a control program, consider the microwave oven (see Fig. 1-1). The microwave oven contains an electronic circuit that controls both the temperature of the microwave heating element and the time that the heating element is on. The heart of the circuit is a *microprocessor*, a programmable electronic device. The fact that it is programmable means that the manufacturer

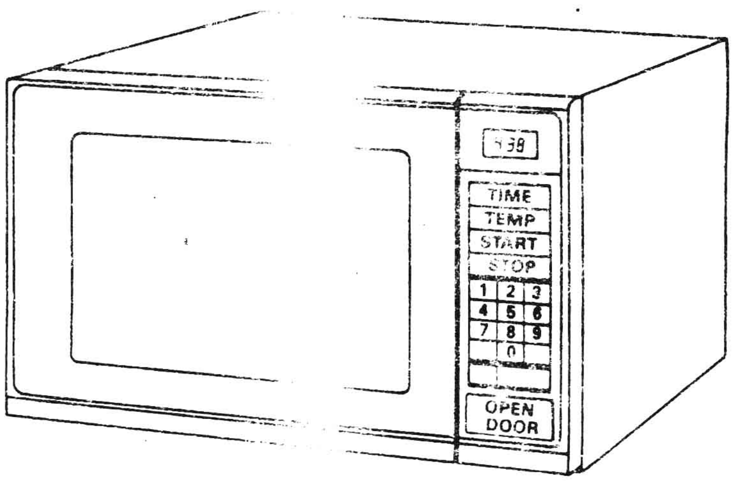

Figure [illegible] A microwave oven.

of the microwave oven can change how the oven works simply by changing the program that this microprocessor (MPU) runs. There is usually no need to develop an entirely new electronic circuit board for next year's model.

Although we do not know the details of the microwave oven program, we can determine what kinds of things it makes the MPU do. Since the oven has a keyboard, the program must contain instructions that tell the MPU how to get information from that keyboard. The oven also has a small display, so the program must contain instructions on what to display and how to display it. And the program must instruct the MPU how to keep track of the time that has elapsed since the user pushed the oven's "Start" button. In short, the program tells the MPU how to control the hardware--keyboard, display, timer, cooking element, and so on. Before you can write such a program, you will need to learn about MPU-based hardware and the instructions for a particular MPU.

Since so much of the electronics industry uses MPUs, you need to have some skills in writing programs to be successful after you are graduated. In this book, you will learn how to write general programs. In later courses, you will learn about MPU-based hardware devices, and then about writing programs to control them. In their last MPU course, the students at my college eventually design a traffic light system for a busy intersection; the system allows for every possible combination of "delayed green," turning lanes, day/night cycles, and so on. They use a microprocessor and MPU-based hardware connected to a variety of lights, switches, and timers, and write the program that controls the system.

We will begin by writing applications programs that will be useful in your "Circuits I" course. Later in your schooling, as you learn more and more about electronics, you will also learn more and more about control programming. This experience will not make you a professional programmer. If you wanted to be a professional programmer, you probably would have majored in computer science, not engineering technology. My goal is to provide you with the tools that let you write the programs that you will need to solve problems relating to CET/EET.

The C Language

I have chosen the programming language called C as the programming language for this book. My reasons were these:

1. Now that a standard has been published by the American National Standards Institute (ANSI), programs written in C can be used on any computer (i.e., they are portable.
2. C has many of the characteristics of a so-called higher-level language (HLL). These allow a programmer to write working programs without knowing a lot about computers.
3. C has many of the characteristics of a so-called lower-level language. These allow a programmer to use features of the computer hardware as they are needed. This feature will be important to us when we write control programs.

The particular C we use is published by Borland International and is called *Turbo C*, version 2.0. Although *any C that conforms to the 1989 ANSI standard for C may be used*, I chose Turbo C because I think that you will find the Turbo C "Integrated Development Environment" very easy to use. However, there is nothing in 99 percent of the programs in this book that depends on your using Turbo C. Turbo C runs on any computer that is compatible with the IBM PC.

Objectives

Upon successful completion of this chapter, you will be able to:

- State the name of the language we will use to solve problems on a PC in EET and CET
- Feel confident that you are not going to be forced to become a computer programmer
- State and explain that a PC has a CPU (also called an MPU), volatile main memory, nonvolatile secondary memory, a keyboard, and a screen
- Explain that information consists of programs or data values
- Explain that a program is a list of instructions for a computer
- State that Turbo C has an editor, a compiler, and a linker built in
- Use **#include** in a program
- Use comments (/*. . .*/) in a program
- Use **main()** in a program
- Use **puts()** in a program
- Know what a **string** is
- Know what a **newline** is

- Know what an **escape sequence** is
- Use braces in a program

Note: do not be concerned if some of the terms mentioned above are new to you. They are either part of the C language or technical terms, and will be explained in the text.

1-2 PROGRAMMING

Programming is the art of getting a computer to do what you want it to do. As an EET/CET, you will want a computer to do many things for you.

For example, in your first electronics circuits course, you will be required to solve a number of circuits problems. These will involve writing one or more equations, supplying values for some of the unknowns, and solving for others. Analyzing these circuits is the engineering technology aspect; plugging in the numbers and cranking out the answer is something that a junior high school student with a calculator can do. If you can program the computer to accept your equations and values, it will do the junior high school work for you. The advantage of having the computer do the tedious work of multiple calculations is that it can do them much more accurately and quickly than you can.

There are applications for computers in the EET/CET world that are even simpler than doing circuit calculations. For example, one of the first things that you will be asked to learn in your circuits course is a table of engineering prefixes and corresponding powers of 10 (e.g., "micro" means "10^{-6}"). If you write a program that will print the entire table, you will have a handy reference when you need it.

Figure 1-2 shows a very simple C program. You will see a detailed explanation of this program later in this chapter, but it will be easier to talk about C programs if you see one before we got too far along. Feel free to guess at what each line does; you may be surprised at how correct you are!

```
/* my first C program!  */

#include <stdio.h>

main()
{
        puts("\n\nhello, world!\n");
}
```

Figure 1-2 A simple C program.

1-3 COMPUTERS

Our focus is on the microcomputer or personal computer (PC). You may already have some experience using PCs, but in writing this book, I have assumed that you have no experience. I will tell you everything that I believe you need to know.

Here is a partial list of some of computers that are considered to be PCs:

1. The IBM PC, PC-XT, PC-AT, models of the PS/2, and so on.
2. Computers that behave just like these IBM computers; these include computers manufactured by Zenith, Epson, Compaq, Tandy, Toshiba, NEC, AT&T, Hewlett-Packard, AST Research, and others, as well as many "off-brands" manufactured in the United States and in the Far East. These are frequently called *compatibles* or *clones*, but I will call these PCs, too.
3. The Apple IIe, Macintosh, and other Apple computers.
4. The Commodore 64 and 128 and the Amiga.
5. The Atari family.

A PC usually has a microprocessor as its CPU (central processing unit—the "brains" of a computer); memory; a typewriter-like keyboard; a video display screen; some form of "mass storage," also called "secondary memory"—for example, one or more disk drives; and perhaps a printer. The CPU, memory, and disk drives are usually inside the case that houses the computer. The keyboard and screen are usually external to that case. If keyboard and screen are in one housing, it is called a *video display terminal* (VDT).

The CPU

The CPU's job is to execute a program's instructions. That is, if an instruction tells the CPU to take the number found in a specified place in memory (called the "location") and add it to the number found in another specified memory location, the CPU will examine the first location, copy the number stored there onto an electronic scratchpad, examine the second location, copy its number, add it to the first number, and write the result on the electronic scratchpad. The CPU has a built-in arithmetic-logical unit (ALU) that handles calculations.

All computers have CPUs. In some machines, the CPU is a microprocessor, so for those machines, we can refer to the "brains" as either the CPU or the MPU. The microprocessor is a miracle of electronics packaging techniques. It contains a vast number of simple electronic circuits interconnected to form a complex device, all on a chip of silicon.

The Memory

Memory is the place in the PC where **information** is stored. It is like the notebook you might use in a laboratory; the notebook contains the instructions for doing the lab experiment and has places for you to write the measurements you took during the experiment. Computer information is divided into two categories: **data** and **program**. Data are the pieces of information that are processed by the CPU. For example, the prefix "micro" and the number 10^{-6} are two pieces of information that would be part of a table of information for relating engineering prefixes and powers of 10. Processing these particular pieces of information might require nothing more than simply sending them to the screen for display. Other data may need calculations performed. For example, the number of hours you worked last weekend and the number representing your hourly pay rate can be considered as data items. For them, processing would include multiplying them together to determine your gross pay. A program, as I have said, is a list of instructions for the CPU to execute. But it is more than that. A program is a **complete** list of instructions that tell the CPU **exactly** how to perform a given task. Like the MPU, a PC's main memory is electronic and implemented on chips, usually called **RAM**. (*Note:* RAM was at one time a proper acronym, but today is simply the name given to main memory.)

Mass Storage

Mass storage is needed because main memory—the memory that is in the same box as the CPU—is **volatile**. That is, when we turn off the power to the computer, everything that was contained in RAM is lost. The next time we turn on the power, random values—called *garbage*—will be written into every memory location, so any programs or data we had in memory before are lost forever. Because of the volatility of main memory, we need to have some nonvolatile secondary memory (i.e., mass storage) to retain the results of our hard work. This almost always is some kind of magnetic disk. As you know from your experience with audiocassette recorders, magnetism is relatively permanent, so when a recorder magnetizes spots on a disk, they will stay magnetized until they are erased or remagnetized. Computer disks are relatively inexpensive and hold a lot of information, so we can safely store large amounts of data and/or many programs on a disk. For example, the computer I am using to write this book has drives for two "floppy" disks and two "hard" disks. A disk drive and a disk are related like a cassette deck and a cassette. The disk is the medium on which information is stored, while the drive is the device that reads and writes the information on the disk. Just as an LP record revolves past a nearly stationary pickup arm, a disk revolves past the drive's read/write heads. On PCs, disk drives are lettered starting at "A". Conventionally, "A" and "B" are floppy disk drives, while "C" is a hard disk. Your school's computers may differ.

One of my floppy disk drives can handle a disk that is 5.25 inches in diameter; such a disk can hold about 360,000 (360K—"K" means "1024") "characters" of information; a character is a letter or digit or punctuation mark. Figure 1-3 shows a 5.25-inch floppy disk. Just like an audiocassette, the floppy disk uses a very flexible plastic material (hence the name "floppy disk") embedded with magnetizable material. This material is the disk itself (sometimes called "the medium") and is permanently housed in a protective jacket, usually heavy black paper. NEVER attempt to remove the plastic disk from this jacket! The jacket has openings that give the read/write heads of the drive access to the disk as it rotates inside the jacket. NEVER touch the disk through any of these openings! Every disk you buy comes in a paper sleeve. Keep the disk in there when it is not in a drive. The sleeve will protect the medium from fingers, pencils, and so on.

The other disk drive is for a 3.5-inch disk; one of these can hold twice as much information as a 360K 5.25-inch disk. Figure 1-4 shows a 3.5-inch floppy disk. Its floppy medium is housed in a hard plastic jacket and has a sliding metal cover for read/write head access. The disk has no need for a separate sleeve.

In about the same physical space volume as a floppy disk drive, each of my

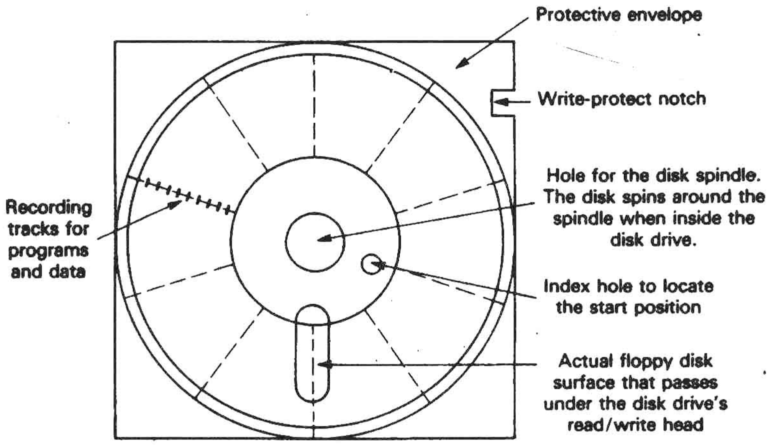
Protective envelope
Write-protect notch
Hole for the disk spindle. The disk spins around the spindle when inside the disk drive.
Recording tracks for programs and data
Index hole to locate the start position
Actual floppy disk surface that passes under the disk drive's read/write head

Figure 1-3 A 5.25-inch floppy disk. Dologite/Mockler, USING COMPUTERS, 2/e, © 1989, p. 59. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, N.J.

hard disks can store about 24,000,000 characters! Hard disks and their drives are usually integral; that is, the media cannot be removed from the drive. This permits the hard disk drive designer to design a system that not only holds much more information than a similar-sized floppy disk drive, but it can also be many, many times faster in reading and writing. However, this makes the hard drive system

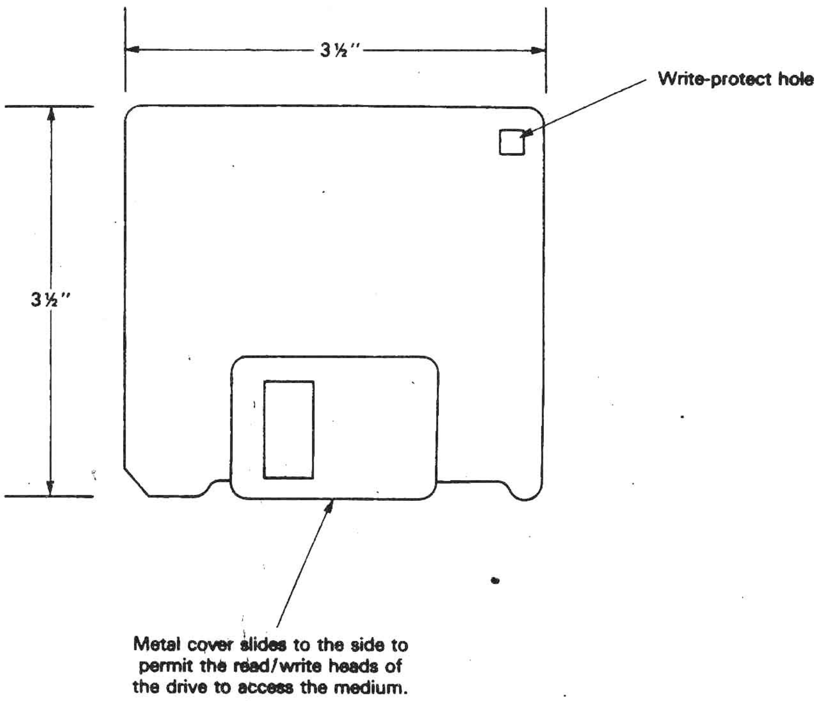
3½"
Write-protect hole
3½"
Metal cover slides to the side to permit the read/write heads of the drive to access the medium.

Figure 1-4 A 3.5-inch floppy disk.

more sensitive to vibration. Don't bang on a computer or drop your books on the table.

To use a 5.25-inch floppy disk (sometimes called a "diskette"), remove it from its sleeve, place it into the floppy disk drive, and close the door. (The 3.5-inch disk has no sleeve, and opening and closing the drive door is automatic.) Then you can either type commands that will transfer information from main memory to the disk, or vice versa, or you can use a program that will transfer information automatically. When the transfer is completed, remove the floppy from the drive, put it back into its protective sleeve, and put it somewhere safe. That means away from coffee spills, cigarette smoke, magnetic fields, anything that would bend the disk, and so on.

Before a floppy disk can be put into service for the first time, it must be initialized, or "formatted." Unlike an audio cassette or a VCR tape, you cannot just put a diskette into the drive and write information to it—you must format it first. Formatting a diskette divides the surface into a fixed number of concentric "tracks" (like the rings on a dart board) and then slices each track into a fixed number of sectors (like the slices of a pie). Next, the formatting program places location numbers (and some other information) on each sector; these permit the computer to find information written to the diskette. When the computer needs to retrieve some information from disk, it looks in a table to find out where (i.e., the track number and the sector number) the information is stored. Then it commands the drive to move the read/write heads to that track, and when the proper sector passes under the read/write heads, the drive reads the information from that sector. Formatting also removes certain information from the disk, so for all practical purposes, anything you had stored on the disk will be lost, so be careful before you format any floppy—instead of it being the new one you just bought, it may be the one that has your lab projects on it!

Originally, floppy disks were 8 inches in diameter. Advances in the production of magnetic media led to the 5.25-inch floppy diskette, and more recently, to the 3.5-inch floppy microdiskette. Amazingly, as the floppy has grown physically smaller, its capacity has risen. For example, the original 8-inch floppy disk could hold only 250,000 characters. Table 1-1 shows the evolution of floppy disks. "M" means "K^2" or 1,048,576.

Hard disk drives (also called "fixed disks") contain "platters" that can be as small as 3.5 inches or as large as 14 inches in diameter. However, platters are generally not removable from their drives, so you will probably never see one.

TABLE 1-1 EVOLUTION OF FLOPPY DISKS

Diameter (in.)	Number of characters	
8	250,000	(250K)
5.25	360,000	(360K)
5.25	1,200,000	(1.2M)
3.5	720,000	(720K)
3.5	1,440,000	(1.44M)