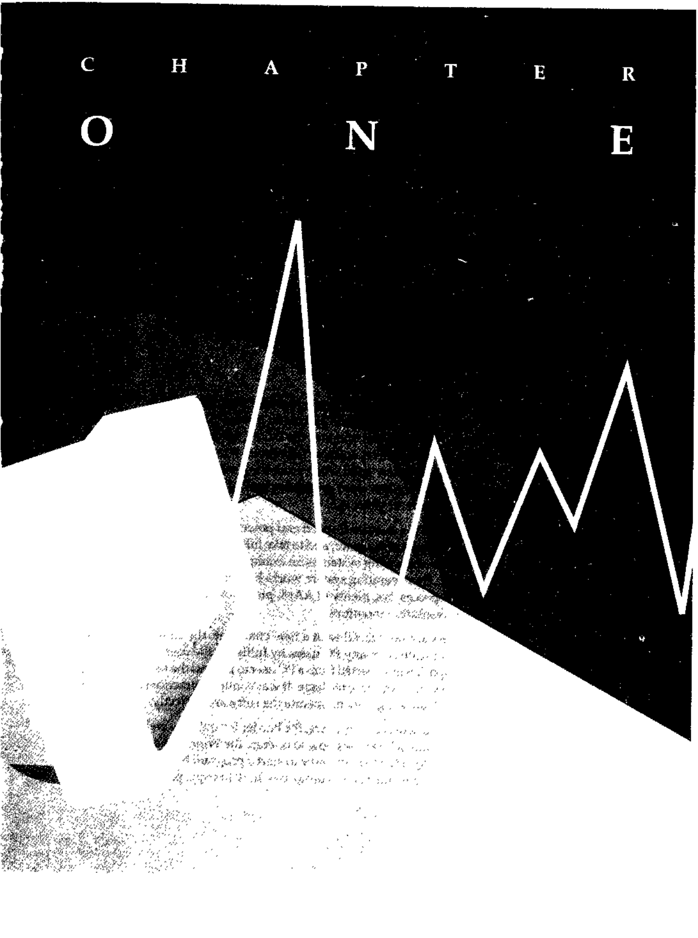
CHAPTER ONE



Sound Programming

Do you want your PC to speak and play music from within your C programs? If so, then this book is for you.

The Concept of the PC Speaking and Playing Music

PCs are becoming less expensive, more powerful, and more popular. They can be found in almost every business and many homes. An integral companion to the PC is the software that operates the PC. PC software has become more powerful, more competitive, and less expensive. For example, several years ago, there were only a few good word processing software packages. Today, you can find many excellent word processing packages at a fraction of the price that was offered several years ago.

Because of the PC's increased speed and power, users expect the software to perform more tasks, to be faster, and to take full advantage of the PC power. The MS Windows operating system is an example of this hardware and software evolution. This operating system works best on powerful PCs that have fast 80386/486 processors, plenty of RAM, plenty of hard drive, and a VGA or other high-resolution monitors.

Programmers are now faced with a new challenge: the challenge of writing programs that attract many PC users by fulfilling the users expectations. Several years ago, it was expected from a PC user to possess the technical ability to operate a powerful software package. It was common practice to blame the end user for not knowing how to operate the software correctly.

With the introduction of Windows, the border is now very well defined. A qualified user is one who knows how to operate the Windows operating system. The user is expected to know how to start a program by clicking the program icon, how to execute several programs, how to copy, paste, and cut text and graphic objects from one program to another, how to access the program menus, and how to perform other common Windows tasks.

As a programmer, you are faced with the challenge of writing your program so that a user who is equipped with the knowledge of performing these basic operations will be able to operate your program without any difficulties.

Programs that Appeal to the Human Senses

Besides being easy to operate, your programs should be attractive to the user and pleasant to use. To make a program attractive and pleasant to use, you should appeal to the human senses: to the visual sense and to the hearing sense.

To make a program pleasant and appealing to the visual sense of the user, the application should be displayed in a graphics mode, with colors. When applicable, the program should include animation (for example, the illusion of a moving, mechanical pushed button when the user clicks a push button in a Windows application).

To make a program pleasant and appealing to the hearing sense of the user, the application should talk and play real music.

Several years ago, almost all programs used the internal built-in speaker to produce beeps that alerted the user of certain conditions and events that occur during the program's execution. Nowadays, a well-written program should speak to the user, telling the user in human voice the reason for the audio alert rather then producing annoying beeps.

There are two ways to enable the PC to produce human voice: by using a sound card, and by using the internal built-in speaker that already exists in every PC.

The Concept of the PC Speaking With and Without Additional Hardware

The Personal Computer (PC) is a computing machine designed to accomplish almost all conceivable tasks. There is one task, however, that the PC is not designed to do: to speak in real human voice, and to play real music.

To add this capability to the PC, the sound card was invented. The sound card is a card that is plugged into one of the slots of the PC's motherboard. It includes jacks to connect an external microphone and external speakers to it. Your program may send commands to the sound card, telling it to play sound files.

The sound card is a peripheral that introduces sophistication to your programs. Sound cards need devices such as a microphone and external speakers. The reality, however, is that not all PCs have sound cards installed.

In this book you will learn how to write C programs that make the PC play sound through both a sound card as well as through the PC's internal built-in speaker.

This book teaches you how to write C programs that make the PC play sound for programs that are executed under the DOS operating systems and for programs that are executed under the Windows operating system. The speech files played by the PC are real human voice (not synthesized voice), and the music files played by the PC are real music.

You will also learn how your program can detect the presence or absence of a sound card in the PC, and how to direct the sound to either the sound card or to the internal built-in speaker. If your program discovers that the PC does not have a sound card installed, it will.play through the internal built-in PC speaker. This little speaker (that already exists on all PCs) can produce real human speech and real music. This concept is illustrated in Figure 1.1.

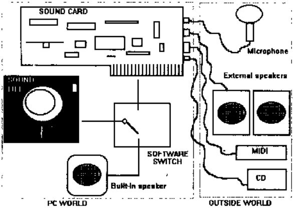


Figure 1.1. The concept of playing through the internal built-in PC speaker instead of through the sound card.

Once you learn how to incorporate sound into your programs, you'll never be able to write a program that does not include sound. As a programmer, you probably already utilize the PC speaker to produce some primitive beeps during program execution. No more! From now on, you'll replace those nasty annoying beeps with intelligent audio prompts.

Installing this Book's Disk

To write C programs that can record and play sound files, you need various software. The book's disk includes software libraries of C functions that your program can use. Thus, to use these C functions, your program has to be written in the C language, and then compiled and linked using either the Microsoft C compiler or the Borland C compiler.

The TSEngine

The libraries of the C functions included on the book's disk are *regular* C functions that can be called from your C programs for DOS, and from your C programs for Windows. They enable you to incorporate sound into your programs with great ease. This powerful collection of C functions is referred to in the book as the TSEngine.

This Book's Disk

This book's disk includes the C source code of this book's

Programs

Software utilities

Sound files

Libraries of C functions for the Microsoft C compiler for creating Windows applications

Libraries of C functions for the Borland compiler for creating Windows applications

Libraries of C functions for the Microsoft C compiler for creating DOS applications

Libraries of C functions fcr the Borland compiler for creating DOS applications

Other software utilities

This software is the short version of the TS Sound Plus library by TegoSoft Inc. Although it is the short version, the supplied software enables you to compile, link, and execute all the book's program examples by yourself. It also enables you to write similar programs by yourself.

Installing this Book's Disk

The software included with this book is stored in a compressed form. You cannot use the software without first installing it on your hard drive.

Follow these steps to install the files on your hard drive:

1. From a DOS prompt, set your default drive to the drive that contains the installation disk. For example, if the disk is in drive A:, type **A:** and press Enter.
2. Type **INSTALL** and press Enter.

This will create a directory named C:\SPSDK and install all the files to that directory. You will need at least 4.6M of free space on your hard drive to install the files.

[illegible] installation on your C: hard drive. This drive [illegible] some of the programs. If you wish to change [illegible] easy matter to change the drive references in the code.

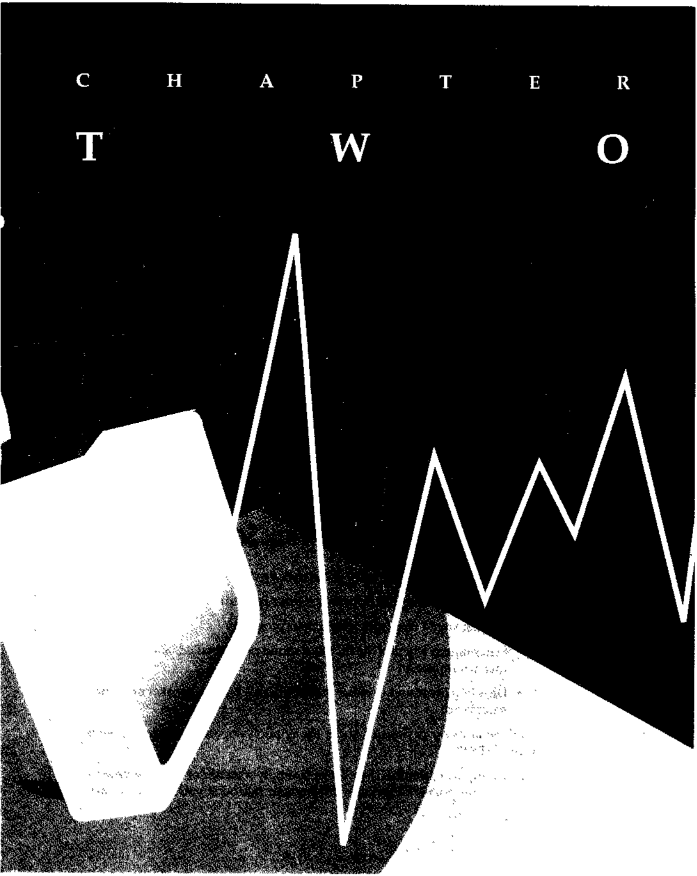
Once the files are installed in your hard drive, several subdirectories are created. Make sure you read the software license agreement that resides in the subdirectory, LICENSE.

The text of this book corresponds to the content of the book's disk. However, you may find some minor differences between some sections of code that appear in the book to the code that appears on the book's disk. These differences may be due to last minute changes, and they are minor. (For example, the book's disk may include different program icons.) In any event, all the book's programs were compiled, linked, and tested. So always refer to the code in the book's disk as the most upgraded code.

As stated, you'll be able to compile and link all the book's program examples by yourself. Nevertheless, the book's disk also includes the already compiled and linked Windows programs. This enables you to immediately execute the book's programs and *hear* the programs. This way, you'll gain a better understanding of what you are expected to learn from this book. The next chapter shows you how to execute these programs.

The program examples of this book are explained in an easy-to-learn, step-by-step manner. So relax, and prepare yourself for a very pleasant journey.

CHAPTER TWO



Sound Programming Techniques and Sound Libraries

There are two separate sound-programming topics to learn:

1. How to make the PC play sound (either through the PC's internal built-in speaker, or through an installed sound card).
2. How to best utilize the PC's ability to speak and play music for the creation of powerful and impressive programs.

How To Make the PC Play Sound

Throughout this book, you learn about making the PC play sound by:

- Recording sound files from within your programs
- Playing sound files from within your programs
- Determining whether the PC has a sound card installed, and based on this determination
- Directing the sound to either the PC internal built-in speaker or to the sound card

All these tasks are performed under the Windows operating system as well as under DOS.

Learning to accomplish these tasks is simply a matter of learning how to use the appropriate C functions. Throughout this book you will encounter many program examples that show you step-by-step instructions on how to do it.

How To Best Utilize Sound from Within Your Programs

Once you know how to produce sound from within your programs and are able to grant the PC the power of speech, you'll have to think of how to best utilize this new dimension into your programs. This is where you will apply your technical artistic talent and ingenuity. Sound programming can be incorporated into all types of programs: serious business programs, word processors, software utilities, animation, demo programs, and so on.

When you study the program examples in this book, you will learn how the sound playing ability is used. You will be instructed to compile, link, and execute the program examples of this book. The book's disk includes all the source code of the programs.

We will now go over some of the programs found in the book. We recommend that you execute the programs before learning their codes, this way you'll *hear* and *see* what you are expected to gain and learn.

The Windows Operating System

The Windows operating system is becoming very popular. Currently, almost all PC vendors ship their PCs with a mouse device and with the Windows operating system already installed.

Creating New Icon Groups in the Program Manager

If your PC has the Windows operating system installed, you may follow the proceeding steps to create a new icon group in the Program Manager of Windows that contains the icons of the book's programs.

There are two methods for creating this icon group:

Method 1

To begin with, look at Figure 2.1 where you will see several icon groups in the Program Manager of Windows (for example, Main, Accessories, and so on). We will now add a new group of icons to the Program Manager—a group of icons that includes the programs of this book.

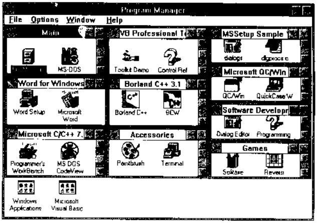


Figure 2.1. Several icon groups found in the Program Manager of Windows.

- Click the File menu of the Program Manager. The File menu appears, as shown in Figure 2.2

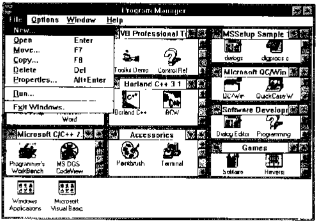
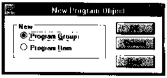


Figure 2.2. The File menu of the Program Manager.

- Select the New o ›tion from the File menu. The New Program Object dialog box appeais, as shown in Figure 2.3. This dialog box lets you create either a new program group or a new program item. Because we are now trying to create a new program group, click the Program Group radio button (see Figure 2.3).



New

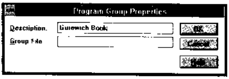
● Program Group

○ Program Item

Figure 2.3. The New Program Object dialog box.

- When you select the OK push button, the Program Group Properties dialog box appears. Type **Gurewich book** in the Description field of the Program Group Properties dialog box (see Figure 2.4). (You may leave the Group File field empty.)

  When you select the OK push button, a new icon group appears, as shown in Figure 2.5. As you can see, there are no icons in this group. We will now add some icons, each representing a different program from the book.



Description: Gurewich Book

Group File:

Figure 2.4. The Program Group Properties dialog box.

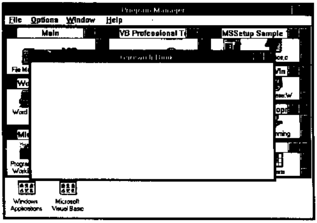
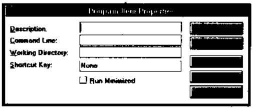


Figure 2.5. The Gurewich Book group.

We will now add icons to the Gurewich Book group.

- While the title of the Gurewich Book group is highlighted, click the File menu of the Program Manager. The File menu appears, as shown in Figure 2.2.
- Select the New option from the menu. The New Program Object dialog box appears, as shown in Figure 2.3.

  Because we are now adding a new program item (not a new program group), make sure that the Program Item radio button is pushed (not the Program Group radio button).
- Select the OK push button of the New Program Object dialog box. The Program Item Properties dialog box appears, as shown in Figure 2.6



Description:
Command Line:
Working Directory:
Shortcut Key: None
□ Run Minimized

Figure 2.6. The Program Item Properties dialog box.

For our first icon, we'll add the Dog program.

- In the Description field of the Program Item Properties dialog box type **Dog**.
- Fill in the Command Line field of the Program Item Properties dialog box by typing

  c:\spSDK\Samp4Win\Dog.exe

  An alternative way to fill the Command Line field is to use Browse:

- Click the Browse push button and select `c:\spSDK\Samp4Win\`.
- Select the `DOG.EXE` file.

After you push the OK push button, the Command Line field should contain

```
C:\SPSDK\SAMP4WIN\DOG.EXE
```

You may leave the Working Directory field of the Program Item Properties dialog box empty.

When you push the OK push button of the Program Item Properties dialog box, a new icon is added to the Gurewich Book group of icons, as shown in Figure 2.7. You may now execute the Dog program by double clicking the Dog icon.

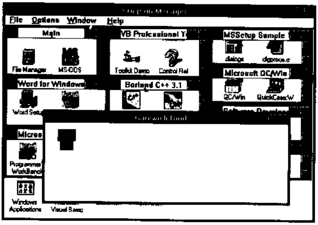


Figure 2.7. The Dog icon is added to the Gurewich Book group.

You can add more icons to the Gurewich Book group of icons by just repeating the process. Add an icon for each of the .EXE files that reside in c:\spSDK\Samp4Win\.

Once you finish creating icons to each of the programs of c:\spSDK\Samp4Win\, the group will contain the icons shown in Figure 2.8

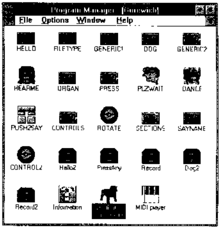


Figure 2.8. The Windows programs of the book.

Method 2

An alternative method for creating the new group of icons lets you create the new icon group faster:

- Log into c:\spSDK\Misc\.
- Copy the Gurewich.grp file to the Windows directory:

  ```
  Copy c:\spSDK\Misc\Gurewich.grp c:\Windows  {Enter}
  ```

- From the Program Manager, double click the File Manager icon (see Figure 2.9).

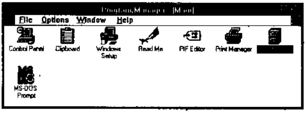
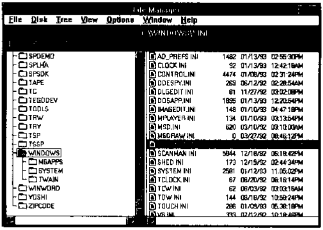


Figure 2.9. The File Manager icon.

The File Manager window appears, as shown in Figure 2.10.



AD_PREFS INI	1482	01/13/93	02:55:30PM
CLOCK INI	92	01/13/93	12:42:18AM
CONTROL.INI	4474	01/08/93	02:31:24PM
DDESPY.INI	269	06/12/92	02:28:54AM
DLGEDIT INI	61	11/27/92	03:02:08PM
DOSAPP.INI	1895	01/13/93	12:20:54PM
IMAGEDIT.INI	148	01/10/93	04:47:18PM
MPLAYER INI	134	01/10/93	03:13:54PM
MSD.INI	620	03/10/92	03:10:00AM
MSDRAW INI	0	03/27/92	08:46:12PM
SCANMAN INI	5844	12/18/92	06:18:42PM
SHED INI	173	12/15/92	02:44:34PM
SYSTEM INI	2581	01/12/93	11:05:02PM
TCLOCK.INI	67	08/20/92	06:16:14PM
TCW INI	62	06/03/92	03:03:16AM
TDW INI	144	08/18/92	10:58:24PM
TOUCH INI	286	01/09/93	05:38:18PM
VB.INI	333	07/12/92	10:18:46PM

Figure 2.10. The File Manager window.

- Double click the PROGMAN.INI file. (This file resides in c:\Windows\.)

  The Notepad program appears with the PROGMAN.INI file ready to be edited (see Figure 2.11).