

IBM PC and PCjr

Subroutine Cookbook

BASIC
TRICKS

Five Shows Daily

★ Show Times ★

1-2, 3-4

5-6,

9-10

David D. Busch

Brady

IBM PC AND PCjr SUBROUTINE COOKBOOK

David D. Busch

BRADY COMMUNICATIONS COMPANY, INC.
BOWIE, MARYLAND 20715
A Prentice-Hall Publishing Company

Publishing Director: David Culverwell
Acquisitions Editor: Gisele M. Asher
Production Editor/Text Designer: Janis K. Oppelt
Art Director/ : Don Sellers
Assistant Art Director: Bernard Vervin
Cover Photograph: George D. Dodson
Manufacturing Director: John A. Komsa

Typesetter: Emerald Graphic Systems, Syracuse, NY
Printer: R. R. Donnelley & Sons Co., Harrisonburg, Virginia
Typefaces: Garamond

Acknowledgments to Dave Kisser's rabbit, Fluffy, for appearing on the front cover.

Copyright © 1985 by Brady Communications Company, Inc.
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission in writing from the publisher. For information, address Brady Communications Company, Inc., Bowie, Maryland 20715.

Library of Congress Cataloging in Publication Data

Busch, David D.

IBM PC and PCjr subroutine cookbook.

Includes index.

1. IBM Personal Computer—Programming. 2. IBM PCjr (Computer)—Programming. 3. BASIC (Computer program language) 4. Subroutines (Computer programs) I. Title.

II. Title: I.B.M. PC. and PC. jr subroutine cookbook.

QA76.8.I2594B87 1985 001.64ff2 84-11079

ISBN 0-89303-542-4

Prentice-Hall International, Inc., London
Prentice-Hall Canada, Inc., Scarborough, Ontario
Prentice-Hall of Australia, Pty., Ltd., Sydney
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Prentice-Hall of Southeast Asia Pte. Ltd., Singapore
Whitehall Books, Limited, Petone, New Zealand
Editora Prentice-Hall Do Brasil LTDA., Rio de Janeiro

Printed in the United States of America

85 86 87 88 89 90 91 92 93 94 95 1 2 3 4 5 6 7 8 9 10

Note to Authors

Do you have a manuscript or a software program related to personal computers? Do you have an idea for developing such a project? If so, we would like to hear from you. The Brady Co. produces a complete range of books and applications software for the personal computer market. We invite you to write to David Culverwell, Publishing Director, Brady Communications Company, Inc., Bowie, Maryland 20715.

PREFACE

How many times have you looked over a program listing in a magazine, and thought, "Gee, I could have saved a lot of time if I'd used this joystick subroutine in my own game program!"

Did you read an explanation of how to use your IBM Personal Computer's PLAY statement, only to wonder, "Well, I think I understand how it works—but how do I actually do it?"

Worse, do you find that examples are too complex to understand, or that tightly packed programs that you try to dissect are so interwoven and poorly commented that it's impossible to extract the purpose of each statement? Have you been reading a lot of useful tips and programming tricks, but lost track of them because they were scattered among a few dozen books and magazines?

This book may be the reference you need and may serve as your shortcut to programming proficiency. Herein are a variety of programming "recipes" in the form of BASIC subroutines that, for the most part, perform only a single task. Useful functions are laid out in subroutines that you can transplant directly to your own programs.

In most cases, the routines are presented in simply constructed lines with only one or two statements per line, and no extraneous material. That makes it easy for you to look at the routines, and discover on your own the function of each statement. But, to make sure that you grasp each concept, there is a line-by-line description and an explanation of the important variables used in each subroutine.

Some of the information in this book is available elsewhere, but you'd have to compile a huge stack of material to collect all of it in one place. Instead of searching through back issues of magazines, the reader can thumb through the Contents or Index, and find out how to simulate joysticks or paddles, generate specific sound effects, or perform various types of sorts.

Most subroutine books concentrate on "general" business or personal routines. Those are included here, too, but we've also emphasized IBM-specific tips aimed at your special needs. New capabilities have been added to IBM BASIC. Special features such as key trapping, use of asynchronous communications, generating sounds, programming the special function keys, and using the built in real-time clock are covered.

Whether you're already expert in BASIC programming, and looking for a handy reference guide, or a new user seeking access to sophisticated subroutine tricks, this book should satisfy your hunger.

INTRODUCTION

Be forewarned. This book is unlike any other collection of subroutines that you might have seen before. Herein are nearly six dozen useful, ready-to-transplant subroutines and programming tips that you can use to make your own programs simulate joystick action or resound with music. These are IBM PC and PCjr-specific routines that take the mystery out of using function keys, the built-in clock, interrupt routines, and other special IBM PC and PCjr features.

Most “subroutine” books are top-heavy with exotic math functions and rarely used statistical programs. If you’ve ever picked up one of those volumes, you were probably dismayed to find that most of the subroutines were not very useful. Most of us don’t really use higher mathematics in our everyday work. That type of subroutine was fine back in the days when microcomputers were used primarily by scientists, computer nuts, and other high-tech types who doted on newer and better ways of doing things like Fast Fourier transforms.

However, the IBM PC and PCjr, while they are powerful, capable microcomputers, are being sold to a broad range of users. Some buy PCs or XTs, and only want to use their computers for business. Other users of both the PC and PCjr are more interested in learning programming and may have a limited technical background. Then, there are those of you who really do understand computers but would like to avoid reinventing the wheel.

The *IBM PC and PCjr Subroutine Cookbook* is meant for all of you. There are some useful, general routines included here. This book also bristles with modules designed specifically to perform some sorely needed task for the IBM PC and PCjr alone.

Interested in using the cursor pad keys as pseudo-joysticks to manipulate objects on the screen? Just transplant one of the joystick routines included in this book. We even show you how to move your missiles and enemy aliens around on the screen.

Using the IBM PC and PCjr’s real-time clock to measure elapsed time or to control outside events is also provided for. Generate musical notes within your own programs—or add sound effects. Ready-made subroutines are provided for your use.

Games players on both the PC or PCjr will find tips on routines that spice up their own arcade-quality games, while those interested in programming for business will revel in the user-friendly input routines, menus, and sort routines.

More advanced programmers can use several routines as utilities to make their work easier, while doing sophisticated “soft” POKing of individual bits within a multipurpose IBM PC and PCjr register.

We've gone light on the simpler subroutines, although plenty of the more important conversion and financial routines are provided. The emphasis here is on modules you can't find anywhere else but which will help you improve your programming immediately.

Sorting is another task that is typically very slow in BASIC. However, because of the great demand for this routine, two sorts are included here. For limited-size lists, one of them should be entirely acceptable.

HOW TO USE THIS BOOK

This is not a first programming book. There are dozens of books that can teach you BASIC. However, there are fewer volumes like this one that can help you go the next step—beyond basic BASIC to true proficiency. If you already know what a FOR-NEXT loop is and what happens when your PC encounters GOTO, you are ready for the lessons contained here. Ideally, you should have written several programs on your own, and be ready to tackle some more sophisticated programming. Those who need this book most will know who they are. You're the programmers who need some gentle guidance, a few inside tips, and the luxury of not having to reinvent the wheel.

While many of the subroutines in this book are ready-to-run programs in their own right, they will be most useful to you when you transplant them into your own programs. In doing so, it may be convenient to relocate them. Because they begin their search for a line number at the beginning of a program, all IBM PCs work fastest when accessing subroutines located there. So, you will probably want to deposit yours there. Using the BASIC RENUM facility will make this task easy.

To make things simpler, the routines are divided into sections. The basic routine itself is clearly labelled. This portion may be renumbered and placed wherever convenient. If renumbering manually, make sure the GOTO's and GOSUB's in the new modules are correct. You don't want a line that reads: "1000 A\$=INKEY\$:IF A\$="" GOTO 160".

Another section of each subroutine will usually be labelled "Initialization." These lines will contain values that must be set once during a program, before the routine is run, or the variables will be those that must be defined by your program before calling the subroutine. Frequently, these lines can be deleted or an equivalent line placed within your own program. The explanation with each subroutine tells the purpose of the important variables.

The purpose of all the variables that you need to define, as well as the variable returned by the subroutine for your program's use, is explained as well. Because many of the subroutines are rather complex, some of the variables used only internally, as well as various operations, may not be explained. This should be rare, as the line-by-line descriptions cover nearly all the functions of every program. However, if this book does not tell you what a variable does, it is information you do not need to know in order to use the subroutine.

In some cases there are several related routines. For example, there are several joystick routines. Some of the concepts are explained only once. You will be directed to look at previous subroutines for longer explanations at times. This allows you to access the routines in any order, without reading the entire book.

In most cases, the subroutines will work equally well in machines equipped with either a color/graphics adapter or the IBM monochrome adapter. Therefore, in most, but not all, of the subroutines, appropriate SCREEN statements have been left out. If you have both types of monitor adapters (like the author) you will need to set the MODE yourself. To make these subroutines easily adaptable for both 80-column and 40-column screen widths, they have been written somewhat generically. You can add screen formatting touches, including more refined LOCATE statements, to suit your particular configuration. This is a subroutine cookbook; the finishing touches of the meal are up to you.

Variable names have been chosen, when possible, to reflect their functions in the subroutines. In most cases, the variable names from one subroutine do not conflict with those of another. However, when writing a complex program using several of these modules, you should check to see that the same variable is not used twice for different purposes. Keep in mind that, unlike many other BASICs, up to 40 characters are significant for variable names in IBM PC and PCjr BASIC. So, PAYMENT, used in one subroutine, is actually a different variable name than PAID, which might be used in a second. But, if your subroutine and program both make independent use of the variable PAID, conflicts resulting in incorrect answers could result. You should take this precaution with any program you write, whether "foreign" subroutines are being transplanted or not. Variable names may contain reserved words but may not consist of reserved words alone. Therefore, "TOTAL" is fine as a variable, but "TO" is not.

Although most of the subroutines will operate properly with any of the available IBM BASIC's, they were written specifically for Advanced Basic. Those that require special features of BASIC 2.0 are noted.

If you are eager to get started, and have some experience in programming, you might want to skip ahead to any subroutine that looks tempting.

Each subroutine explanation includes a list of three sample applications for the subroutine to get you started. The descriptions usually include other hints on where you can use them in your programs. But, ideas should not be in short supply.

You should find this book a shortcut to programming proficiency. To paraphrase a common saying, if you use a subroutine correctly three times, it will be a permanent part of your vocabulary. Given a bit of practice, you can soon have all your friends drooling over your programs and asking you for your favorite subroutine recipes. Good luck.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The author and publisher of this book have used their best efforts in preparing this book and the programs contained in it. These efforts include the development, research, and testing of the programs to determine their effectiveness. The author and the publisher make no warranty of any kind, expressed or implied, with regard to these programs, the text, or the documentation contained in this book. The author and the publisher shall not be liable in any event for claims of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the text or the programs. The programs contained in this book and on any diskettes are intended for use of the original purchaser-user for backup purposes without requiring express written permission of the copyright holder.

TRADEMARKS OF MATERIAL MENTIONED IN THIS TEXT

- IBM MACHINES—International Business Machines Corporation
- APPLE MACHINES—Apple Computer, Inc.
- COMMODORE MACHINES—Commodore Business Machines, Inc.
- RADIO SHACK MACHINES—Tandy Corporation
- MASTERMIND—Invicta
- MICROSOFT—Microsoft Corporation
- BREAKOUT—Atari, Inc.
- PONG—Atari, Inc.
- PACMAN—Midway Mfg. Co., A Bally Company
- HOMEWORD—Sierra On-Line, Inc.
- ATARI MACHINES—Atari, Inc.
- TEXAS INSTRUMENTS—Texas Instruments, Inc.
- DUNGEONS & DRAGONS—TRS Hobbies, Inc.



The Brady IBM PC Library

Inside the IBM PC: Access to Advanced Features and Programming

Peter Norton

The most widely read author on the IBM PC explains the workings of the computer.

1983/320 pp/ISBN 0-89303-556-4

Book/diskette package ISBN 0-89303-561-0

IBM PC: An Introduction to the Operating System, BASIC Programming and Applications—Revised and Enlarged

Larry Joel Goldstein and Martin Goldstein

An updated and expanded version of what has become the classic self-study book for the IBM PC.

1983/392 pp/ISBN 0-89303-530-0

Book/diskette package ISBN 0-89303-527-0

IBM PC & XT Assembly Language: A Guide for Programmers

Leo J. Scanlon

An introduction to the principles of microprocessors (specifically the 8088), numbering systems, and assemblers.

1983/320 pp/ISBN 0-89303-241-7

Book/diskette package ISBN 0-89303-535-1

MS-DOS and PC-DOS: User's Guide

Peter Norton

The authority, Peter Norton, gives an introduction and explanation of the Microsoft Disk Operating System. Explains DOS 1.0 and 2.0 versions.

1984/266 pp/ISBN 0-89303-645-5

8087 Applications and Programming for the IBM PC and Other PCs

Richard Startz

Starts with a non-technical introduction and turns into a very detailed description of the myriad applications for the 8087 microprocessor.

1983/276 pp/ISBN 0-89303-420-7

Diskette only ISBN 0-89303-421-5

Book/diskette package ISBN 0-89303-425-8

Communications and Networking for the IBM PC

Larry E. Jordan and Bruce W. Churchill

Brings together data communications applications and the IBM PC. Includes asynchronous and synchronous communications and a complete study of local area networking.

1983/237 pp/ISBN 0-89303-385-5

Handbook of BASIC for the IBM PC

David I. Schneider

Clearly translates the BASIC reference manual supplied with the IBM PC into understandable terms. Organized by BASIC programming statements.

1983/498 pp/ISBN 0-89303-506-8

Book/diskette package ISBN 0-89303-508-4

IBM PC and XT Owner's Manual: A Practical Guide to Operations

Barbara Lee Chertok, Dov Rosenfeld, and James H. Stone

Provides easy instructions on the operation of the IBM PC. Helps to unravel the jargon found in the supplied manual to allow for easy access to operation.

1984/200 pp/ISBN 0-89303-531-9

Advanced BASIC and Beyond for the IBM PC

Larry Joel Goldstein

A complete guide to the advanced skills of BASIC programming, files, graphics, event-trapping, machine language, and subroutines. A must for the IBM PC programmer.

1984/360 pp/ISBN 0-89303-324-3

Book/diskette package ISBN 0-89303-325-1

Business Problem Solving with the IBM PC & XT

Leon A. Wortman

Business professionals will use the dozens of programs specifically designed for decision-making and problem-solving. Source codes are included in BASIC and many are in Pascal.

1983/324 pp/ISBN 0-89303-282-1

Book/diskette package ISBN 0-89303-342-1

Business Applications for the IBM Personal Computer

Steven Zimmerman and Leo Conrad

Offers step-by-step instructions on the use and customization of existing business software programs.

1983/300 pp/ISBN 0-89303-243-3

Book/diskette package ISBN 0-89303-351-0

Games, Graphics, and Sound for the IBM PC

Dorothy Strickland, Dennis Rockwell, and Kevin Bowyer

Teaches how to program in BASIC, Pascal, and FORTRAN to create graphics and sound for the IBM PC. Illustrates how to integrate sound and graphics into animation.

1983/256 pp/ISBN 0-89303-469-3

Book/diskette package ISBN 0-89303-470-3

BASIC Engineering and Scientific Programs for the IBM PC

Philip M. Wolfe and C. Patrick Koelling

A source of BASIC programs for on-the-job use by engineers and scientists to provide important computer techniques for problem-solving and data manipulation.

1983/358 pp/ISBN 0-89303-330-8

Book/diskette package ISBN 0-89303-331-6

Pascal for the IBM PC: IBM DOS Pascal and UCSD p-System Pascal

Kevin W. Bowyer and Sherry J. Tomboulion

The first word written on combining the IBM PC with Pascal programming. An emphasis on sound and graphics applications is provided.

1983/416 pp/ISBN 0-89303-280-8

Book/diskette package ISBN 0-89303-761-3

WordStar for the IBM PC: A Self-Guided Tutorial

Micro Workshop of Cambridge

Now WordStar can be mastered with this step-by-step journey into it's many features.
1984/280 pp/ISBN 0-89303-956-X

The C Programmer's Handbook

Thom Hogan

The ultimate encyclopedia of information necessary to use C. Choose other books to learn C, choose this book to USE C.

1984/288pp/ISBN 0-89303-365-0

The C Programming Tutor

Leon A. Wortman and Thomas O. Sidebottom

Here is a tutorial that goes beyond the standard guides to C language. Contains a variety of C programming examples in a self-study format.

1984/274pp/ISBN 0-89303-364-2

Programming the IBM PC & XT: A Guide to Languages

Clarence B. Germain

A gold mine of information on programming the PC. An in-depth reference guide for programmers seeking skills necessary to operate and run programs on the IBM PC. Covers PC-DOS 1.1 through 2.0.

1984/338 pp/ISBN 0-89303-783-4

Available at your local bookstore or computer retailer. Or write to Brady Communications Company, Inc., Bowie, MD 20715. Phone (301) 262-6300.

CONTENTS

PREFACE	vii
INTRODUCTION	ix
1 SUBROUTINE MAGIC	1
Sample Merging Run	5
2 BASIC TRICKS	9
Function Keys	10
Check Screen	13
Switch Displays	16
Test Adapter	18
Clear Keyboard Buffer	19
Center Screen	21
File Transfer	23
3 DATA INPUT, EDITING, AND OUTPUT	29
Number Input	30
Letter Input	33
Case Converter	35
String Sort	37
Number Sort	39
Array Loader	42
Insert String	45
CHR\$ Value	47
Sequential File—Write to Disk	50
Sequential File—Read from Disk	54
4 USING THE CLOCK AND INTERRUPTS	59
Elapsed Time	61
Timer	63
Second Timer	66
TIME\$ Interrupt	68
ON COM(n) Interrupt	71
Function Key Interrupt	73

5	BUSINESS AND FINANCIAL SUBROUTINES	77
	Loan Amount	78
	Payment Amount	81
	Number of Payments	83
	Years to Reach Desired Value	86
	Compound Interest	88
	Rate of Return	91
	Temperature	93
	Date Formatter	95
	Number of Days	98
	Day Converter	101
	Menu	103
	Time Adder	106
	MPG	108
6	BITS AND BYTES	113
	Peek Bit	115
	Bit Displayer	117
	Bit to One	119
	Bit to Zero	121
	Reverse Bit	123
	Binary to Decimal	125
	Rounder	127
7	JOYSTICKS AND PADDLES	131
	Horizontal Paddle Simulation	133
	Vertical Paddle Simulation	136
	Cursor Pad Joystick—N, S, E, W	139
	Cursor Pad Joystick—All Directions	142
	Drawing Subroutine	147
	Read Joysticks	150
	Joystick Button Interrupt	154
8	USING SOUND	157
	Music	159
	IBM Organ	162
	Siren	165
	Bomb	167
	Alarm	169
	Klaxon	171
	UFO	173
	Computer	175
	Laser	176
	Roulette Wheel	178

Heartbeat	180
Clock	182
Noise	184

9	GAME ROUTINES	187
	Deal Cards	188
	Random Range	192
	Coin Flip	194
	Dice	196
	Delay Loop	198

GLOSSARY	201
-----------------	------------

INDEX	205
--------------	------------

1

Subroutine Magic

One of the best things about subroutines is that they can be reused many times within an existing program and put to work in many different pieces of software as well. Once you have typed in, say, a joystick routine from this book, you will not need to retype it every time you write a new program requiring joystick handling. Because the subroutines in this book have been designed as stand-alone modules, with both the input and output clearly defined, they can be recycled quite easily. You will want to store your subroutine "library" on disk and call it into your programs as needed.

Incorporating existing code into a program is called "merging" and can be accomplished in many different ways. The very simplest can be used if only one stock subroutine will be used in your new program. In such cases, just load the subroutine you want into memory, and write all the other program lines around it.

But, what if you want to incorporate several subroutines into a program, or add them to one which has already been written? Doesn't loading a new subroutine or program destroy anything that is in memory? Not necessarily. The PC and PCjr have a powerful, simple, MERGE command that allows merging program lines and subroutines.

First, let's look at the two kinds of merging. In one case, your existing program and the subroutines to be merged have line numbers that do not conflict. Perhaps one or the other has low line numbers, while the code to be merged has high line numbers. That is, your program is numbered from 100 to 1000, while the subroutine(s) to be added all have line numbers higher than 1000. Computerists have a special name for this kind of merge: "appending." One program or module is added to, or appended to, the end of the other. This method is easiest to use, from the standpoint that there is no danger that wanted program lines will be written over with those of the merged program.

However, in the case of true merges your target program may have program lines that are inclusive of those in the subroutine to be merged. Your program numbered from 100 to 1000 can be merged with a subroutine that is numbered from 500 to 600. If any duplicate lines exist, those of the original program will be replaced by those of the merged program. With some planning, such a merging scheme can also be successful. You would need to make sure that there are no program line numbers in common by, say, purposely leaving a gap between lines 500 to 600 in your original program. Or, perhaps, those lines are occupied by a subroutine that you no longer want. When using this type of merge, be certain that there are no "leftover" lines from the original subroutine or program overlapping with those of your subroutine. For most, the append type of merge is the safest and easiest to implement.

To MERGE with the PC, go to BASIC and load the module that you wish to add to memory. The subroutine or program that you wish to MERGE must be stored in ASCII form, which is accomplished by appending ",A" to the filename when storing, e.g., SAVE "A:filename.BAS",A