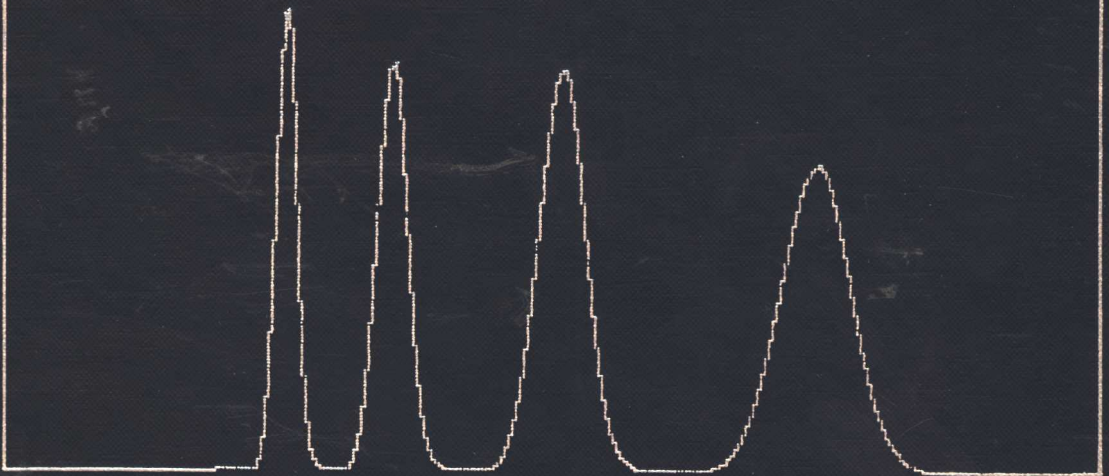


Raymond Annino
Richard Driver

**SCIENTIFIC
AND ENGINEERING
APPLICATIONS
WITH
PERSONAL COMPUTERS**



Scientific and Engineering Applications with Personal Computers

A SOFTWARE APPROACH WITH
EXAMPLES FOR THE APPLE-
IBM-PC-, AND CP/M-BASED
MICROCOMPUTER SYSTEMS

RAYMOND ANNINO
RICHARD D. DRIVER

A WILEY-INTERSCIENCE PUBLICATION
JOHN WILEY & SONS

New York • Chichester • Brisbane • Toronto • Singapore

9 8 7 6 5 4 3 2 1

Copyright © 1986 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

Annino, Raymond.

Scientific and engineering applications with personal computers.

"A Wiley-Interscience publication."

1. Science—Data Processing. 2. Engineering—Data processing.
3. Programming (Electronic computers) 4. Microcomputers.
- I. Driver, Richard D. II. Title.

Q183.9.A56 1985 502'.8'5416 85-20275
ISBN 0-471-79978-5

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

I fy mhlant,
Gareth,
Ellen
a Megan
R.D.D.

Marie,
the dawn
is breaking
R.A.

Preface

As scientists and experimentalists we view a laboratory computer system as an instrument to aid us in our work—to ease the burden of data acquisition and reduction, and to enhance the quality and value of experimental data. The availability of data acquisition and control hardware interfaced to personal computers (PCs), as well as the cost effectiveness of these systems, which can be a fraction of the total instrumentation cost in an experiment, has accelerated their laboratory use. In addition, the host microcomputer can be used for such tasks as modeling or word processing.

We have used computers for years in our own research and are aware of the problems faced by the laboratory user. We have also been fortunate to participate in the recent explosive growth of PCs in science and engineering. For us at least, using PCs for data acquisition and control has made laboratory work much easier. Although a number of books are available that discuss the interfacing of PCs to the real world, there is no single book that adequately discusses the important software aspects of these systems. We feel that the strength of these systems will only be realized by fully utilizing and extending the software tools currently available.

Our purpose, therefore, in writing this book is to provide the experimentalist with a single source that introduces the important software aspects of laboratory PC usage. Although the material is primarily software oriented, some basic hardware considerations are discussed. But this is *not* an “interfacing” book. Rather, it continues where many of these interfacing books end. Our contention is that whereas it is one thing to connect hardware to perform a specific function, it is yet another to have the system do meaningful work. Resolving this latter problem is our concern here.

Finally, we recognize that, in this fast-moving computer world, specific

hardware and software items quickly become obsolete. But if we tried to generalize too much, the book would lose its practical value. Hence, we tried to compromise as much as possible. We use existing software and computer products in all our examples, yet at the same time endeavoring to explain the particular solutions in terms of the limitations of the system used.

RAYMOND ANNINO
RICHARD D. DRIVER

*North Smithfield, Rhode Island
Cambridge, Massachusetts
March 1986*

Acknowledgments

We gratefully acknowledge the help and encouragement of many people, without whom we would have surely floundered.

The systems houses were more than cooperative. Paul Warne of Interactive Microware not only furnished us with user manuals but also reviewed the text of the ADLAB software. Likewise, Rush Slivjanovski of MIA supplied documentation and a complementary demonstration disk and reviewed the EASY-LAB software description that we wrote. Cheryl Lekstrom of Data Translation sent us all the documentation she could lay her hands on. Mike Mathison, Marketing Director of Keithley DAS, supplied us with user manuals for their IBM-based systems, and Peg Burgess loaned us her copy of their Apple-based user's manual.

The people at Cyborg were super. Rick Kramer was a tremendous help in programming the ISAAC/Apple system in assembly language. Ron Sostek, Gary Gaudet, Jeff Hookaloo, Jim Ciocolo, and Joy Dardinski were always eager to demonstrate hardware, furnish us with whatever information or documentation we needed, and do whatever else they could to help us to finish the book. The people at Midwest Software provided us their graphics package, and Southwestern Data Systems kindly provided us their assembly language library package. Jeff Bernoff of Software Arts was particularly helpful in helping us develop the TKSolver engineering model and to demonstrate the power of that software. Dick Miller of the Foxboro Company guided us through the intricacies of fluid flow and came up with a modeling exercise from his handbook that is guaranteed to challenge any programmer.

Don Nadon of National Instruments and their representatives in the Boston area, Scheinffein Associates, were very willing to help and gave us much valuable information on the GPIB bus. Bill Kemper of Hewlett-Packard

provided invaluable help on the HP-IL interface. We would especially like to thank the Manx Software Company and Harry Suckow for providing us a C compiler on such short notice. Other individuals and organizations deserving our thanks are Jeff Burman of Ashton Tate, Judy Hanby of Quadram, Adam Green of Software Banc, the people at Eco-C, and Tandy Trower of Microsoft. Gerard Geggis of Computer Showcase, Norwood and Leo Wong of Computer Maintenance Service gave us invaluable help in keeping our trusty Kaypros in top form.

Without the help of an old friend and fantastic programmer, Aaron Sawyer, assembly programming would have been much more difficult—if not impossible. He also reviewed and edited the assembly language chapter. Individuals who provided invaluable hints and suggestions throughout the course of this project were Jan Pejchar, Dave Johnson, Duane Thompson, Greg Maurer, Bob Vernon, and Bob McNally.

Finally, I (RA) would like to acknowledge the patience of my wife, who, for a year and a half, watched me disappear every night into my study with my faithful Kaypro II. She not only put up with this nonsense, but she learned Perfect Writer and typed a large portion of this manuscript during the day while I was at work. I (RDD) would like to thank my family for the support and encouragement necessary to complete this project, and the Delaneys for allowing me to spread my papers across their dining room table during the most intense phase of this project.

R.A.
R.D.D.

Contents

1	INTRODUCTION TO THE SYSTEM	1
1.1	Introduction to Microcomputers	1
1.2	Getting Started with the Apple-Based System	7
1.2.1	<i>Diskettes and Off-Line Storage</i>	8
1.2.2	<i>The Operating System</i>	9
1.3	Overview of the System	11
1.4	Hardware Extensions	14
1.4.1	<i>Analog/Digital (A/D) Conversion</i>	14
1.4.2	<i>Data Rate and Memory</i>	15
1.4.3	<i>Digital I/O</i>	16
1.4.4	<i>Parallel IEEE-488 Standard Interface</i>	16
1.4.5	<i>Serial ASCII Interface</i>	16
1.5	Languages and Language Extensions	17
1.6	Data Handling	19
2	CRITICAL AREAS OF BASIC PROGRAMMING FOR A PC	21
2.1	BASICA vs APPLESOFT	21
2.2	Advanced BASIC Commands	22
2.2.1	<i>DEF SEG</i>	24

- 2.2.2 *POKE and PEEK Commands* 24
- 2.2.3 *The CALL Command* 25
- 2.2.4 *The CLEAR Command* 26
- 2.3 Introduction to DOS 27
 - 2.3.1 *DOS Instructions* 27
 - 2.3.2 *Execution of DOS Commands From Within a Program* 28
- 2.4 Introduction to the Memory Map 31
 - 2.4.1 *Review of The Numbering System* 31
 - 2.4.2 *IBM-PC Memory Map* 33
 - 2.4.3 *Apple II Memory Map for the Apple II Plus* 34
 - 2.4.4 *RAM Organization* 38
 - 2.4.5 *HIMEM* 38
 - 2.4.6 *The Beginning and End of the BASIC Program* 38
 - 2.4.7 *Storage of Simple Variables* 39
 - 2.4.8 *Array Space and the End of Numeric Storage* 40
 - 2.4.9 *String Storage* 40
 - 2.4.10 *Screen Buffers* 40
 - 2.4.11 *Soft500 Memory Management* 42
- 2.5 The Use of Ampersand, &, in Applesoft 43

3 FILE AND PROGRAM MANAGEMENT

46

- 3.1 Mass Storage 46
- 3.2 Data Filing in Applesoft 49
 - 3.2.1 *Sequential Text Files* 51
 - 3.2.2 *Text File Subroutine Language Extensions* 54
 - 3.2.3 *Random Access Text Files* 56
 - 3.2.4 *Binary File Text Storage* 58
 - 3.2.5 *File Naming* 60
 - 3.2.6 *Data Compression* 61
- 3.3 Data Filing under BASICA on the IBM-PC 62
 - 3.3.1 *Sequential Data Files Under BASICA/MBASIC* 62
 - 3.3.2 *Random Data Files Under BASICA/MBASIC* 65
- 3.4 Program Management Under Applesoft 66
 - 3.4.1 *Chaining Programs Together* 67
 - 3.4.2 *The EXEC Command* 68
 - 3.4.3 *Dissecting and Concatenating Programs* 69
 - 3.4.4 *Command Files* 70

- 3.5 Program Management with BASICA 71
 - 3.5.1 Chaining Programs in BASICA 71
- 3.6 Data File Structures 72
 - 3.6.1 The VISILOT/VISITREND Data Structure 73
 - 3.6.2 The Data Interchange Format 76
 - 3.6.3 A File Structure for Experimental Data 81
- 3.7 Database Management with dBASEII 86
 - 3.7.1 Introduction 86
 - 3.7.2 Records and Fields 87
 - 3.7.3 Working with Records 88
 - 3.7.4 The dBASEII File System 90
 - 3.7.5 The dBASEII Command File 91

4 MODULAR AND STRUCTURED PROGRAMMING

94

- 4.1 Introduction 94
 - 4.1.1 What Is Structured Programming? 94
 - 4.1.2 Computer Languages in This Book 95
- 4.2 Modular Program Development in the Laboratory 96
 - 4.2.1 Layout and Definition 96
 - 4.2.2 Modular Program Design 100
- 4.3 Modular Programming in Applesoft 102
 - 4.3.1 Dividing up the Program 102
 - 4.3.2 Program Flags 104
 - 4.3.3 The Module Structure 106
 - 4.3.4 Bookkeeping 109
 - 4.3.5 Modular Programming Using Program Files 110
- 4.4. Modular Programming in BASICA 110
- 4.5. Structured Programming 111
 - 4.5.1 Creating Structured Algorithms 111
 - 4.5.2 Structured Flow and Control Primitives 112
 - 4.5.3 Structured Data Types 114
 - 4.5.4 Structured Languages 118
- 4.6 Programming in C 120
 - 4.6.1 The C Language Syntax 120
 - 4.6.2 Creating a Running C Program 122
 - 4.6.3 Creating a C Library 125
 - 4.6.4 Pointers and Arrays 125
 - 4.6.5 Function Recursion 129

- 4.6.6 *Advanced Programming Example in C* 130
- 4.7 FORTH 131
 - 4.7.1 *FORTH Overview* 132

5 DATA ACQUISITION CONTROL HARDWARE 134

- 5.1 Introduction 134
- 5.2 Review of Hardware Requirements 135
 - 5.2.1 *Signal Conditioning* 135
 - 5.2.2 *Transmission Cabling* 135
 - 5.2.3 *Filtering* 137
 - 5.2.4 *Serial-Parallel Transmission* 142
 - 5.2.5 *Sampling Rate* 144
 - 5.2.6 *Analog-to-Digital Converters* 145
 - 5.2.7 *Important Specs* 147
- 5.3 Kiethley DAS Series 500 Data Acquisition System 148
- 5.4 ISAAC Integrated System for Automated Acquisition and Control 152
 - 5.4.1 *ISAAC—IBM-PC Systems* 156
 - 5.4.2 *ISAAC 2000* 156
- 5.5 Intelligent Interfaces for Laboratory Data Handling 159
- 5.6 PCDAX 161
 - 5.6.1 *Analog and Digital I/O for IBM-PC and APPLE II* 161
- 5.7 ADLAB Data Acquisition System 163
- 5.8 EASYLAB 164
- 5.9 Appligation Hardware 165

6 LABORATORY LANGUAGE SOFTWARE 166

- 6.1 Introduction 166
- 6.2 Labsoft I 168
 - 6.2.1 *Analog I/O* 169
 - 6.2.2 *Binary I/O* 172
 - 6.2.3 *The Counter and Timer Commands* 173

- 6.2.4 *Triggering* 174
- 6.2.5 *The Graphics Capability of Labsoft* 175
- 6.2.6 *Labsoft I for the IBM-PC* 176
- 6.3 *Soft500* 180
 - 6.3.1 *Soft500 for the IBM-PC* 180
 - 6.3.2 *Programming with Soft500* 182
 - 6.3.3 *Soft500 for the Apple II Plus and IIe* 184
- 6.4 *QUICK I/O* 186
 - 6.4.1 *QUICK I/O Commands* 186
- 6.5 *EASYLAB* 189
 - 6.5.1 *Programming with EASYLAB* 190
- 6.6 *PCLAB* 192
- 6.7 *Labsoft II on the IBM-PC* 195
 - 6.7.1 *Introduction* 195
 - 6.7.2 *Overview of the Labsoft II Package* 196
 - 6.7.3 *ISAACOMM Communications Software* 197
 - 6.7.4 *The ISAAC Command Language* 198

7 CONFIGURED LABORATORY SYSTEMS

201

- 7.1 *Designing a Computer-Controlled Chromatographic Analysis System* 201
 - 7.1.1 *Hardware Requirements* 202
 - 7.1.2 *Software Requirements* 204
 - 7.1.3 *Program Development Using Labsoft I* 206
- 7.2 *An Apple-Controlled Spectrometer System* 211
 - 7.2.1 *Background Material* 212
 - 7.2.2 *Hardware Overview of a Photon-Counting System* 214
 - 7.2.3 *The Software Modular Structure* 217
 - 7.2.4 *Macro-Driven Software* 218
 - 7.2.5 *Labsoft Implementation of Software* 220
- 7.3 *Commercially Available Data Handling Software* 223
 - 7.3.1 *Chromatochart and Other IMI Software* 224
 - 7.3.2 *Appligraton* 226
 - 7.3.3 *Model2600 Chromatography Software for IBM-PC* 228
 - 7.3.4 *ASYST* 229
 - 7.3.5 *Labtech Notebook* 230

8 COMPUTER GRAPHICS 232

- 8.1 Introduction to Graphics 232
 - 8.1.1 *Display Technology* 233
 - 8.1.2 *Memory Requirements* 235
 - 8.1.3 *Advanced Graphics Systems* 236
- 8.2 Graphics on the Apple 238
 - 8.2.1 *Memory Requirements* 238
 - 8.2.2 *Graphics Software on the Apple II* 241
 - 8.2.3 *Applesoft Graphics Language Extensions* 244
 - 8.2.4 *Creating a Hard Copy* 244
 - 8.2.5 *Storing Graphical Images on Disk* 246
- 8.3 Graphics on the IBM-PC 248
 - 8.3.1 *Hardware Memory Requirements on the IBM* 248
 - 8.3.2 *Graphics Software on the IBM* 250
 - 8.3.3 *VISILOT on the IBM* 251
- 8.4 Three-Dimensional Graphics 252
 - 8.4.1 *A Recipe* 253
 - 8.4.2 *Transforming Coordinates* 255
 - 8.4.3 *Complex Transformations without Perspective* 256
 - 8.4.4 *Projections with Perspective* 259
 - 8.4.5 *Clipping* 261
 - 8.4.6 *An IBM Programming Example* 262
- 8.5 Hard Copy Graphics 263
 - 8.5.1 *Dot-Matrix Printer Graphics* 264
 - 8.5.2 *Graphics on an X-Y Recorder* 268

9 NUMERICAL ANALYSIS AND MODELING 272

- 9.1 Introduction 272
 - 9.1.1 *The Numerical Analysis Program Library* 272
 - 9.1.2 *Creating a Numerical Function Library in BASIC* 274
 - 9.1.3 *The Limitations of Numerical Analysis on Microcomputers* 276
 - 9.1.4 *Commercially Available Program Libraries* 278
- 9.2 Generating Mathematical Functions 279
- 9.3 Numerical Differentiation and Integration 285
 - 9.3.1 *Differentiation* 285

- 9.3.2 *Integration* 288
- 9.3.3 *The Gauss–Legendre Method* 293
- 9.4 *Spline Interpolation* 294
- 9.5 *Numerical Roots of Equations* 299
- 9.6 *Matrix Analysis* 302
- 9.7 *Solving Differential Equations Numerically* 307
- 9.8 *Simulating Data Noise* 314
- 9.9 *Solving Models with TKSolver* 317
 - 9.9.1 *Overview* 317
 - 9.9.2 *A Modeling Test for TKSolver* 320

10 ANALYSIS OF EXPERIMENTAL DATA

326

- 10.1 *Linear Regression Analysis* 326
- 10.2 *Polynomial Data Fitting* 331
- 10.3 *Nonlinear Least Squares Fitting* 335
- 10.4 *Filtering and Smoothing* 341
 - 10.4.1 *Boxcar Averaging* 341
 - 10.4.2 *The Method of Savitsky and Golay* 343
 - 10.4.3 *The First Derivative* 346
 - 10.4.4 *A Filtering Example* 346
- 10.5 *Fourier Methods* 350
 - 10.5.1 *The Fast Fourier Transform* 352
 - 10.5.2 *Curve Distortions* 359
 - 10.5.3 *Convolution and Data Smoothing* 361

11 ASSEMBLY LANGUAGE PROGRAMMING

365

- 11.1 *Introduction* 365
- 11.2 *Elements of Assembly Language Source* 366
 - 11.2.1 *Label* 366
 - 11.2.2 *Mnemonic* 366
 - 11.2.3 *Operand* 367
 - 11.2.4 *Comments* 367
- 11.3 *Assemblers* 367
- 11.4 *Introduction to the 6502 Microprocessor* 367

- 11.5 The 6502 Instruction Set 371
- 11.6 Assembly Language Routines 372
 - 11.6.1 Addition 373
 - 11.6.2 Subtraction 374
 - 11.6.3 Division 374
 - 11.6.4 Multiplication 379
- 11.7 Assembly Language Communication to Peripheral Devices 380
- 11.8 Boxcar Filter 383
- 11.9 Interrupts 390
 - 11.9.1 Interrupt-Driven Boxcar Routine 398
 - 11.9.2 Subroutine Interrupts in BASICA 399
- 11.10 Modifying the Applesoft CALL Instruction to Pass Arguments 399
- 11.11 Assembly Language Programming the ADLAB Hardware 404

12 STANDARD COMMUNICATIONS INTERFACES

408

- 12.1 Bus Interfacing 409
 - 12.1.1 Bus Structure For Memory-Mapped I/O 409
 - 12.1.2 I/O Bus Structure 410
 - 12.1.3 The Interface 410
 - 12.1.4 Device Drivers 411
 - 12.1.5 The BASIC I/O System (BIOS) for the IBM-PC 411
 - 12.1.6 Bus-Based Microcomputers 411
- 12.2 The RS-232C Standard and Serial Communications 412
 - 12.2.1 Overview of the RS-232 Link 412
 - 12.2.2 Serial Data Format 414
 - 12.2.3 Serial Data Transmission Rate 417
 - 12.2.4 The RS-232 Signal and Control Lines 418
 - 12.2.5 Direct RS-232 Connections 421
 - 12.2.6 Protocol in Serial Communications 423
 - 12.2.7 The Modem 424
 - 12.2.8 The Apple Super Serial Card 425
 - 12.2.9 An SSC-Controlled Instrument 429
 - 12.2.10 The Serial Interface on the IBM-PC 432
- 12.3 The HP-IL Interface 435
 - 12.3.1 Loop Hierarchy 436

- 12.3.2 *Loop Structure* 437
- 12.3.3 *Interface Loop Commands* 439
- 12.3.4 *HP-IL Implementation on Personal Computers* 440
- 12.4 *Parallel Interfaces* 441
 - 12.4.1 *The ISAAC Parallel Interface* 442
 - 12.4.2 *The GPIO Parallel Interface* 444
- 12.5 *The BCD Interface* 445
 - 12.5.1 *The BCD Representation* 446
 - 12.5.2 *The BCD Connection* 447
 - 12.5.3 *BCD Computer Interfaces* 449
- 12.6 *The Centronics Interface* 451

13 THE GPIB INTERFACE

453

- 13.1 *Simple Overview of the GPIB Interface* 454
 - 13.1.1 *Bus Hardware* 454
 - 13.1.2 *Device Addressing* 455
 - 13.1.3 *Controllers, Talkers, and Listeners* 458
 - 13.1.4 *Bus Communications* 459
 - 13.1.5 *GPIB Functions* 462
- 13.2 *A Procedural Approach to GPIB Software* 464
 - 13.2.1 *Procedure Definitions* 465
- 13.3 *The GPIB Bus* 469
 - 13.3.1 *GPIB Bus Commands* 472
 - 13.3.2 *Synthesis of GPIB Command Macro Procedures from Primitive Bus Commands* 475
- 13.4 *GPIB on Hewlett-Packard Computers* 476
- 13.5 *GPIB Interfacing on the Apple* 478
 - 13.5.1 *The Apple GPIB Card* 479
 - 13.5.2 *The CCS-7490 {Apple} GPIB Card* 483
- 13.6 *GPIB Interfacing on the IBM-PC and IBM-XT* 486
 - 13.6.1 *Hardware Overview of the System* 487
 - 13.6.2 *GPIB Functions on the GPIB-PC* 488
 - 13.6.3 *The GPIB-PC Software Handler* 489
 - 13.6.4 *Using the GPIB-PC from BASICA* 490
 - 13.6.5 *Using the GPIB-PC from C* 493