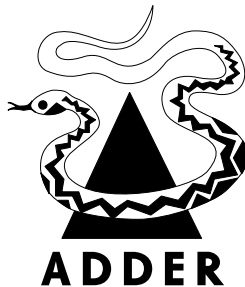


The BASIC ROM User Guide

**for the BBC microcomputer
and Acorn Electron**

Mark D. Plumbley BA,
Churchill College,
Cambridge University



Published by Adder Publishing, Cambridge

Published in the United Kingdom by:
Adder Publishing,
PO Box 148,
Cambridge CB 1 2EQ

ISBN 0 947929 04 5

Copyright © 1984 Adder Publishing
First published August 1984

The Author would like to thank Adrian Dickens, Nigel Dickens, Tim Gleeson, Ken Vail, Leicester Whewell, Albert Williams and everyone else who helped in the production of this book.

All rights reserved. This book is copyright. No part of this book may be copied or stored by any means whatsoever whether mechanical, photographic or electronic, except for private or study use as defined in the Copyright Act. All enquiries should be addressed to the publishers. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of information contained herein.

The Author would like to thank Rockwell International for their permission to reproduce diagrams from their R650X data sheet. These diagrams remain copyright of Rockwell International.

Please note that within this text the terms Tube, Econet, Atom and Electron are registered tradenames of Acorn Computers Limited. All references in this book to the BBC Microcomputer refer to the computer produced for the British Broadcasting Corporation by Acorn Computers Limited. This book was prepared using the Acornsoft VIEW wordprocessor on the BBC Microcomputer and then computer typeset by Parker Typesetting Service, Leicester. Book production by Adder Publishing.

Printed in Great Britain by The Burlington Press Ltd., Foxton, Cambridge.

Contents

Introduction	5
The Operation of BASIC	
1 The 6502 Microprocessor	7
1.1 The 6502 registers	7
1.2 Machine code arithmetic	10
1.3 The instruction set	14
1.4 Addressing modes	19
1.5 Addressing mode groups	23
1.6 The BASIC assembler	26
2 The BASIC System	28
2.1 An overview of BASIC	28
2.2 The BASIC 'CPU'	30
2.3 Tokenising	37
2.4 Program storage	44
2.5 Executing statements	45
3 Memory Use	47
3.1 Variables and the HEAP	47
3.2 The BASIC STACK	59
3.3 Workspace	60
4 Expression Evaluation	63
4.1 Operator precedence	63
4.2 Top-down analysis	63
5 Program Control Mechanisms	68
5.1 GOTO	68
5.2 GOSUB...RETURN	69
5.3 PROCs and FNs	70
5.4 IF...THEN...ELSE	75
5.5 REPEAT ... UNTIL	76
5.6 FOR...NEXT	77
5.7 ON...GOTO/GOSUB	80
5.8 ON ERROR	81
6 Assembling and Disassembling	83
6.1 The assembler	83
6.2 The disassembler	86

Enhancing BASIC

7 Adding New Commands	98
7.1 Trapping BRK	98
7.2 The 'Mistake' error	100
7.3 A single character statement	103
7.4 Recognising keywords	107
7.5 A renumber utility	115
8 Overlaying Procedures	128
8.1 The 'No such FN/PROC' error	128
8.2 Static overlaying	130
8.3 Dynamic overlaying	134
9 Trapping other Errors	143
9.1 Bad MODE recover	143
9.2 Bad program salvage	147
9.3 Error listing	155
10 ROM Routines	162
10.1 Restarting BASIC	169
10.2 Program handling	172
10.3 Statement handling	182
10.4 Expression evaluation	187
10.5 Variable/FN/PROC management	193
10.6 STACK management	207
10.7 Input/output	216
10.8 Type conversion	222
10.9 Integer routines	228
10.10 Floating point routines	237
10.11 Function entry points	262
11 Errors and Error Recovery	264
11.1 The BASIC BRK handler	264
11.2 Numbered errors	267
11.3 Fatal errors	311
Appendices	
A Syntax definition	319
B BASIC ROM summary	327
C 6502 instruction set summary	334
D Keyword summary	336
E Operating system calls and vectors	338
F OSBYTE/*FX call summary	339
G Variable locations	344
Bibliography	345
Glossary	346
Index	351

Introduction

Many books have been written explaining how to program in BBC BASIC, or how to program in 6502 machine code. Most people therefore know BASIC or machine code without really understanding what BASIC itself is up to. This book fills in that gap by providing a complete description of BASIC as a *system*.

Although BASIC is a very large machine code program, it is essentially very simple, as it is very *structured*: once you can see the overall structure of the system, it is very easy to delve deeper and deeper into its workings, to find out exactly what is happening. This book explains that overall structure: program storage, variable storage, expression evaluation, etc., right down to the mechanisms used by a FOR..NEXT loop or a procedure call. Armed with this knowledge, and the disassembler in chapter 6, you can probe right down to the machine code level of BASIC.

Understanding the operation of a large machine code program such as BBC BASIC has many advantages: not only does it point the way for writing large machine code programs yourself, but it also allows you to write your BASIC programs much more efficiently. Once you know what BASIC has to do to interpret a program, it is possible to write faster programs if you need to, by using resident integer variables wherever possible, using PROCs and FNs rather than GOSUBs, and so on.

The second part of this book describes how to add routines on to BASIC to expand the capabilities of your machine, mainly by trapping the errors that it generates. Adding new commands, overlaying procedures, etc., are all covered, together with how to get back into BASIC to continue afterwards. The examples also show you how to use some of the ROM routines to save space and time in you own machine code programs.

The example programs are complete in that you can type them in and run them, and many of them are useful utilities. However, they also indicate the possibilities available to the adventurous programmer – don't be afraid to chop them about, and use them as a basis to put your own ideas into practice. Chapter 10 provides a comprehensive listing of the BASIC ROM entry points (for both BASIC1 and BASIC2), so that you can experiment with other ideas for new utilities.

Of course, using ROM routines directly will mean that your programs might not work on the Tube, Econet, or with a different BASIC; in fact, the BASIC ROM may not even be 'paged in' when you try to use it. For experimenting with your own machine, however, this doesn't really matter. Commercial programs should never use any of these ROM routines; the program might find itself running in a situation you did not allow for. For such programs, or any others which are not restricted to a particular system configuration, only the officially documented facilities should be used.

Note that all Electrons, and the later BBC microcomputers, have BASIC2: the earlier BBC microcomputers have BASIC1. If you are not sure which version of BASIC is in your machine, typing **REPORT** after BASIC has just started up (after a **BREAK** or ***BASIC**), will print the copyright message. If the date is 1981, BASIC1 is fitted; if it is 1982, you have BASIC2. American machines, or those with a second processor, may have US BASIC or HIBASIC: the ROM routines will not be in the same place for these ROMs.

Armed with this book, and plenty of coffee, you should have many happy nights programming. Have fun!