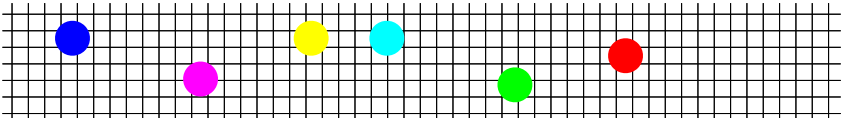


---

BBC MASTER GUIDES

**MASTERING  
ASSEMBLY  
CODE**

RICHARD VIALLS





---

# CONTENTS

<b>Preface</b>	<b>5</b>
<b>1 Assembly language programming</b>	<b>6</b>
Number systems	7
The memory	13
The CPU	15
Commands	17
Addressing modes	21
Conditional branches	24
The index registers	26
Logical commands	39
Indexed indirect addressing	42
<b>2 The operating system</b>	<b>45</b>
Useful OS routines	45
Memory usage	51
<b>3 Pure machine code</b>	<b>55</b>
Addressing modes	56
A machine code monitor	60
<b>4 Interrupts</b>	<b>84</b>
The system VIA	87
Events	97
BRK	100
<b>5 A few ways to protect your programs</b>	<b>103</b>
Locked tapes files	104
Unlistable programs	105
Disc tricks	108
<b>6 The keyboard</b>	<b>113</b>
A BASIC input routine	114
A machine code input routine	114
The BREAK key	120

---

<b>7</b>	<b>General graphics</b>	<b>123</b>
	The graphics registers	123
	The video ULA	123
	Screen splitting	128
	Screen swapping	135
	A BASIC swap	139
	Three-dimensional graphics	143
<b>8</b>	<b>Fill routines</b>	<b>149</b>
	A BASIC fill	150
	A machine code fill	159
	A faster fill	167
<b>9</b>	<b>Screen dumps</b>	<b>180</b>
	A simple BASIC dump	180
	A machine code equivalent	184
	A colour-as-tone dump	188
	A miniature dump	199
<b>10</b>	<b>Sprite graphics</b>	<b>216</b>
	A BASIC sprite routine	217
	A machine code sprite routine	222
	Moving sprites	230
	The flicker lick	239
	Using the mover	252
	Anyone for tennis?	255
	<b>Appendix A</b>	<b>276</b>
	Two's-complement table	
	<b>Appendix B</b>	<b>278</b>
	Assembler commands and op-codes	
	<b>Appendix C</b>	<b>279</b>
	Op-codes and assembler commands	

---

# PREFACE

This book is aimed at the programmer who has become proficient in BASIC and wants to explore the realms of machine code. The first section of the book sets out to give a detailed description of assembly language programming. However, it is impossible to teach someone to program creatively and professionally. The second section of the book discusses techniques and gives a series of examples of the uses of machine code. It is hoped that, by examining these programs in detail, you will begin to think in the ways that produce a good machine-code programmer.

Don't assume that the programs in this book are at a height of perfection. There are probably a number of improvements that can be made to them. Don't just use the programs in this book without thought. If, for example, you write an arcade game don't just use the sprite routine at the end of chapter ten - study that routine and then either write your own or adapt that routine and then either write your own or adapt it to suit your particular needs. This will not only produce better programs, but should also help to make you a better programmer.

The overall message of this book is that a professional programmer is a perfectionist and will do everything within his power to improve a program to its limits.

