

# Intuition Engine Programmer's Reference Guide

---

Copyright (c) 2026 Zayn Otley. All rights reserved.

## Preface

---

Intuition Engine is a modern 64-bit RISC machine: a re-imagining of Commodore/Atari/Sinclair/BBC/Amstrad/IBM 8/16/32-bit home-computer ideas from the 1980s and 1990s. It is built as an homage to that era of home computing, while remaining one Intuition Engine computer with a shared memory bus. Processors, video chips, sound engines, DMA hardware, file devices, input devices, and control registers all sit on the same backplane. When you move from VideoChip to VGA, or from SID to POKEY, or from IE64 to 6502, you are not changing computers. You are programming another card on the same bus.

This guide begins at the BASIC prompt because that is the quickest way to touch the machine. You will type short programmes, inspect memory with PEEK, change hardware with POKE, and then use IE Mon to enter machine-code bytes directly. The examples are written for that path. They do not require an assembler, a build command, or a second machine to understand what is happening.

The book is also a reference. Chapter 2 is deliberately a vocabulary chapter, so skim it on a first reading and return to it when a keyword needs checking. The real climb continues in Chapter 3, where the screen becomes visible, then through sound, memory, processors, I/O, and whole-machine workflows. Part VII is different: it is a guided demo-programming course that uses the same registers and commands to build frame loops, rotozoomers, music-synchronised sections, copper presentation, and complete intro structure. Part VIII is an advanced case study in a large game port. It shows how the same machine model scales when one programme uses an M68K game core, an IE coprocessor, Voodoo rendering, native audio, file storage, input, and profiling as one system.

Keep one rule in mind as you read: every chip and every CPU is part of the same Intuition Engine.

## Contents

---

### Part I - Intuition Engine BASIC

1. BASIC Programming Rules
2. BASIC Language Vocabulary

### Part II - Programming Graphics

3. Display Model Overview
4. VideoChip
5. VGA Text and Graphics Modes
6. TED Video
7. ANTIC and GTIA
8. ULA Display
9. Voodoo 3D Rasteriser
10. Tile and Sprite Layers from BASIC

## **Part III - Programming Sound and Music**

11. Audio Architecture Overview
12. SoundChip and SFX
13. PSG and AY-3-8910
14. SN76489
15. SID Family
16. TED Audio
17. POKEY
18. AHX Engine
19. MOD Playback
20. WAV Sample Player
21. MIDI/MUS, Live MIDI, and RawlandMini GM Synth
22. Paula DMA Engine
23. Music from BASIC and from each CPU

## **Part IV - BASIC to Machine Language**

24. Memory Model and MMIO Map
25. IE64
26. IE32
27. 6502
28. Z80
29. M68K MC68020-Class
30. x86
31. Processor Timing, Traps, and Exceptions
32. Coprocessor and Cross-CPU Calls
33. IE Mon - the Machine Monitor
34. IE Script

## **Part V - Input / Output Guide**

35. Disk and File I/O
36. The HOST Command
37. Keyboard, Mouse, Controller MMIO
38. Serial Devices

## **Part VI - Whole-Machine Project**

39. Whole-Machine Capstone
40. Interrupts, Raster Timing, and Polling
41. Building, Loading, and Laying Out Programmes
42. Coprocessor Positive Cookbook
43. Debugging and Profiling Cookbook
44. A Larger Whole-Machine Example

## **Part VII - Demo Programming**

45. Your First Frame Loop
46. The Rotozoomer In BASIC

47. Driving The Hardware From IE Script
48. From Floating Point To Tables
49. The Rotozoomer In IE64 And IE32
50. One Effect, Six CPUs
51. Wobble, Texture Building, And Logo Motion
52. Music-Synchronised Effects
53. Copper, Raster Bands, And Layered Presentation
54. Building A Complete Intro
55. When BASIC Is Not Enough

## **Part VIII - A Full Game Port Case Study**

56. Why This Port Is Different
57. Separating Game Code From Platform Code
58. The IE Runtime Layer
59. Fast3D To Voodoo
60. Hardware TnL With The Coprocessor System
61. Native IE Audio Instead Of RSP-Style Mixing
62. Assets, ROM Data, And Build Hygiene
63. Performance Work On A Real Port
64. Input, Save Data, And Player-Facing Polish
65. Lessons For Your Own IE Ports

## **Appendices**

- A. IE64 BASIC Keyword Abbreviations and Token Map
- B. Screen and Character Codes
- C. ASCII and CHR\$ Tables
- D. Per-Engine MMIO Maps
- E. Music Note and Frequency Tables
- F. Math and Derivative Helpers
- G. Per-CPU Opcode Quick Reference
- H. Per-CPU Symbol Index
- I. Error Message Index
- J. Full Memory Map
- K. Block Diagrams
- L. Index