

Raspberry Pi OS
ASSEMBLY
LANGUAGE

Hands-on-Guide

Fourth Edition

www.brucesmith.info

Raspberry Pi OS Assembly Language Hands On Guide

© Bruce Smith

Editions 1 ,2 3, Previously published as *Raspberry Pi Assembly Language*
Raspbian ISBN 9781492135289

Fourth edition Revised and updated: March 2021 [0004]

iSBN: 978-0-6480987-3-7

Editor: Alan Ford Edits, Melanie Smith

Cover: Sumit Shringi, Graphic Designer (Book Cover-Design)

Typeset by BSB using Affinity Publisher

All Trademarks and Registered Trademarks are hereby acknowledged.
Within this *Hands On Guide* the term BBC refers to the British Broadcasting Corporation. Raspberry Pi and the Raspberry Pi logos are registered trademarks of the Raspberry Pi Foundation.

Raspberry Pi OS Assembly Language: Hands On Guide is not endorsed by the Raspberry Pi Foundation.

All rights reserved. No part of this book (except brief passages quoted for critical purposes) or any of the computer programs to which it relates may be reproduced or translated in any form, by any means mechanical electronic or otherwise without the prior written consent of the copyright holder.

Disclaimer: Whilst every effort has been made to ensure that the information in this publication (and any programs and software) is correct and accurate, the author and publisher can accept no liability for any consequential loss or damage, however caused, arising as a result of the information printed in this book and on any associated websites. Because neither BSB nor the author have any control over the way in which the contents of this book is used, no warranty is given or should be implied as to the suitability of the advice or programs for any given application. No liability can be accepted for any consequential loss or damage, however caused, arising as a result of using the programs or advice printed in this book.

Published by BSB. www.brucesmith.info.

Table of Contents

Table of Program Listings.....	10
1: Introduction	15
Imagination Unlimited	16
Start Experimenting.....	17
GNU C Compiler	17
Learn by Example	18
What Will You Learn?	18
Fourth Edition and Compatibility	19
Raspberry Pi OS.....	20
What About 64-bit?	20
Keyboard Computing.....	21
The Significance of ARM.....	21
Raspberry Pi Through the Ages	22
Compute Modules.....	22
Notation in Use.....	23
Table Terminology.....	24
Centre for Computing History	24
Companion Website and Free Books	26
Acknowledgements.....	26
2: Starting Out	27
Numbers with Meaning	27
ARM Instructions	28
The Transformation Process.....	29
Why Machine Code?	30
Language Levels	30
Into Orbit.....	31
RISC and Instruction Sets	32
Assembler Structure	32
Error Of Your Ways.....	33
Cross Compilers	33

The Raspberry Pi ARM Chip(s)	34
3: First Time Out	35
The Command Line	35
Creating A Source File	36
Come to the Execution	39
Assembler Errors	40
The Components	41
Lack of <code>_start</code>	43
Linking Files.....	44
Tidying Up	46
A Comment on Comments	46
Geany Programmer's Editor	47
4: Bits of a RISC Machine.....	49
Binary to Decimal	50
Decimal to Binary	51
Binary to Hex	52
Hex to Decimal and Back.....	53
Binary Addition	54
Subtraction	55
Twos Complement Numbers	56
When Twos Don't Add Up	57
Desktop Calculator	58
5: ARM Arrangements	59
Word Lengths	59
Byte and Word Accessed Memory	60
Registers.....	61
R15 - Program Counter	63
Current Program Status Register	63
Bits and Flags.....	63
Setting Flags	64
S Suffix.....	65
R14: The Link Register.....	66
R13: Stack Pointer.....	66
6: Data Processing.....	67
Addition Instructions	68
Subtraction	71
Multiplication.....	73
Divide Arrives.....	75
Move Instructions.....	76

Compare Instructions.....	76
Ordering Numbers	77
7: ROS Ins and Outs	79
SWI and SVC Commands	79
Writing to the Screen.....	80
Reading from the Keyboard.....	82
eax and Others.....	84
Makefile	84
8: Logical Operations	87
Logical AND	87
Logical OR.....	88
Logical EOR.....	88
Logical Instructions	89
ORR to Convert Character Case	90
Bit Clear with BIC.....	91
Flag Tests.....	92
System Call Registers	95
9: Conditional Execution	97
Single Flag Condition Codes	99
EQ: Equal	99
NE: Not Equal	99
VS: Overflow Set.....	99
VC: Overflow Clear	100
MI: Minus Set	100
PL: Plus Clear	100
CS: Carry Set (HS: Higher or Same)	101
CC: Carry Clear (LO: Lower).....	101
AL: Always	101
NV: Never	102
Multiple Flag Condition Code	102
HI: Higher (Unsigned)	102
LS: Lower Than or Same (Unsigned)	102
GE: Greater or Equal (Signed)	103
LT: Less Than (Signed)	103
GT: Greater Than (Signed)	103
LE: Less Than or Equal To (Signed)	103
Mixing the S Suffix	104

10: Branch and Compare	105	Pre-Indexed Addressing.....	147
Branch Instructions	105	Accessing Memory Bytes	149
The Link Register.....	106	Address Write Back	150
Using Compare Instructions	106	Post-Indexed Addressing.....	150
Compare Forward Thinking.....	107	Byte Conditions.....	153
Using Conditionals Effectively	108	PC Relative Addressing.....	153
Branch Exchange	110		
11: Shifts and Rotates	111	16: Block Transfer	155
Logical Shifts	111	Write Back	157
Logical Shift Right	113	Block Copy Routine.....	158
Arithmetic Shift Right	113		
Rotations.....	114	17: Stacks	161
Extended Rotate.....	114	Push and Pull.....	161
Uses of Shifts and Rotates	115	Stack Growth.....	163
Immediate Constant Range.....	115	Stack Application	165
Top Move	118	Framed Work.....	165
		Frame Pointer	166
12: Smarter Numbers	119		
Long Multiplication	119	18: Directives and Macros	167
Long Accumulation	121	Data Storage Directives.....	167
Division and Remainder.....	122	Aligning Data.....	169
Smarter Simple Multiplication.....	123	Macros	170
Much More Inside.....	124	Including Macros	173
13: Program Counter R15	125	19: File Handling	177
Pipelining.....	126	File Permissions	182
Calculating Branches	127		
		20: Using libc	185
14: Debugging with GDB	129	Using C Functions in Assembler.....	185
Frozen Cases.....	129	Source File Structure	186
Assembling for GDB	131	Investigating the Executable	188
The Disassembler.....	133	Number Input with Scanf.....	190
Breakpoints	134	Getting This Information.....	193
Breakpoint Labels	137		
Memory Dump.....	138	21: Coding Functions	195
Shortcuts.....	139	Function Standards.....	195
GDB Make Options	140	Register Use	197
		More Than Three.....	197
15: Data Transfer	143	Preserving Links and Flags.....	199
ADR Directive.....	143	Robust Print Routines	200
Indirect Addressing.....	146	Bubble Sort.....	202
ADR and LDR	147		

22: Disassembling C	205		
GCC The Swiss Army Knife.....	205		
A Simple C Framework.....	206		
Sourcing the Assembler	208		
Building Blocks.....	209		
A printf Example.....	211		
Frame Pointer Variables	213		
Disassembling System Calls.....	213		
23: GPIO Functions	215		
Memory Mapping	215		
The GPIO Controller.....	217		
GPIO In and Outs.....	219		
Other GPIO Functions.....	230		
GPIO Pins Explained	231		
24: Floating-Point	233		
VFP Architecture	233		
The Register File	236		
Managing and Printing	237		
Assembling and Debugging VFP with GDB	238		
Load, Store and Move	240		
Precision Conversion.....	242		
Vector Arithmetic.....	243		
25: VFP Control Register	245		
Conditional Execution	245		
Scalar and Vector Operations.....	248		
Which Type of Operator?	250		
Len and Stride.....	251		
26: Neon	255		
Neon Assembler.....	257		
Neon Instructions and Data Types	259		
Addressing Modes.....	261		
VLD and VST in their Stride	261		
Load of Others	264		
Neon Intrinsic	264		
Neon Arrays.....	265		
Order Correctly	270		
Matrix Math	271		
Multi Matrix	274		
			Macro Matrix Example
			279
27: Thumb Code	281		
Differences.....	281		
Assembling Thumb	283		
Stack Operators	288		
Single and Multi-Register	288		
Functions in Thumb.....	289		
ARMv7 Thumb Instructions	289		
28: Unified Language	291		
Thumb Changes	293		
New A32 Instructions	293		
Compare by Zero.....	294		
Assembling UAL.....	295		
29: Exception Handling	297		
Modes of Operation.....	297		
Vectors	298		
Register Arrangements	300		
Exception Handling.....	302		
MRS and MSR.....	303		
Interrupts When?	304		
Your Interrupt Decisions.....	305		
Returning from Interrupts.....	306		
Writing Interrupt Routines	306		
30: System on a Chip	307		
The ARM Chip & Instruction Sets	308		
Co-processors	309		
Pipeline	309		
Memory & Caches	310		
The GPU.....	310		
ARMv8 Overview	311		
Raspberry Pi OS 64-Bit.....	312		
In Summary.....	312		
Archimedes Principle.....	312		
A: ASCII character Set	314		
B: ARM Instruction Set	316		
Compare and Test Instructions	317		
Branch Instructions	317		

Arithmetic Instructions	317
Logical Instructions	318
Data Movement Instructions	318
C: ROS Syscalls	319
Index	323

Table of Program Listings

Program 3a. A simple source file.	39
Program 3b. Part 1 of the source file.	44
Program 3c. Part 2 of the source file.	45
Program 6a. Simple 32-bit addition.	69
Program 6b. 64-bit addition.	70
Program 6c. 32-bit multiplication.	73
Program 6d. Using MLA - Multiply with Accumulate.	74
Program 6e. Signed Division with SDIV.	75
Program 7a. Syscall 4 to write a string to the screen.	81
Program 7b. Syscall 3 to write a string to the screen.	82
Program 7c. Automate assembly and linking with Make.	85
Program 8a. Converting character case.	90
Program 8b. Printing a number as a binary string.	92
Program 10a. Conditional execution to improve program size.	109
Program 12a. Long multiplication ‘the hard way’	120
Program 12b. Division the long way.	122
Program 14a. Flexible makefile for debugging code.	139
Program 15a. Use of the ADR directive.	142
Program 15b. Use of pre-indexed indirect addressing.	147
Program 15c. Using post-indexed addressing.	150
Program 16a. Moving Blocks of Memory	157
Program 18a. Use of .byte and .equ directives	166
Program 18b. Implementing a simple macro.	169
Program 18c. Multi-calling a macro.	170

Program 18d. AddMult macro file	172
Program 18e. Macro Include Test	172
Program 19a. File Creation and Access.	176
Program 20a. GCC source file structure.	185
Program 20b. Passing parameters to printf.	187
Program 20c. Reading and converting a number with scanf.	189
Program 20d. Combining scanf and printf.	191
Program 21a. Passing function values via the stack.	196
Program 21b. Print vector of words	198
Program 21c. Test printw	199
Program 21d. Bubble Sort Routine	200
Program 21e. Bubble Sort Test	201
Program 22a. A simple C program to print an asterisk.	205
Program 22b. The final putchar listing.	208
Program 22c. C listing for the ‘hello world’ program.	209
Program 22d. A C listing for using the scanf function.	210
Program 23a. File access to GPIO memory.	219
Program 24a. Printing a floating-point value with printf.	235
Program 24b. Manipulating and printing two or more fp values.	239
Program 25a. Conditional VFP-based execution.	246
Program 25b. Using LEN and STRIDE to sum vectors.	251
Program 26a. A Simple Neon Test.	255
Program 26b. Rotate a 2D matrix by 90 degrees (clockwise).	264
Program 26c. Adding two 4x4 Matrices together.	270
Program 26d. Single-Precision Matrix Multiplication	274
Program 26e. Single-Precision Matrix Multi - Using Macro Version	277
Program 27a. How to invoke Thumb State and run Thumb code.	282
Program 27b. Using external functions by inter-working code.	286
Program 28a. Unified Assembly Language	294

**The source files from this book are available to
download at the author’s website:
www.brucesmith.info**