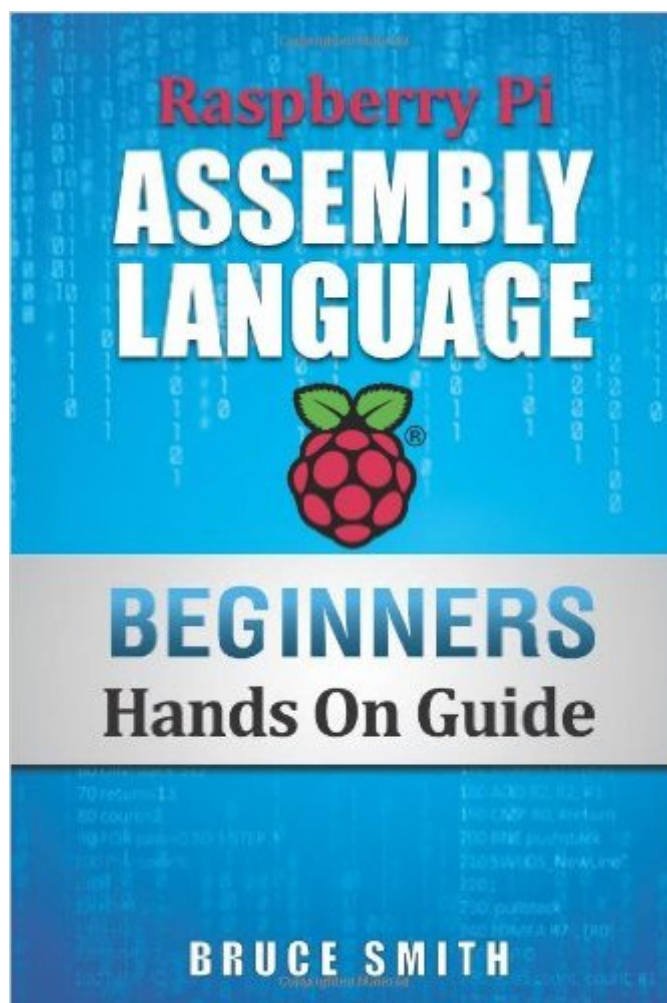


The book was found

# Raspberry Pi Assembly Language Beginners: Hands On Guide



## Synopsis

Raspberry Pi Assembly Language Beginners is no longer available. Please see [www.brucesmith.info](http://www.brucesmith.info) for more information.

## Book Information

Series: Hands On Guide

Paperback: 200 pages

Publisher: CreateSpace Independent Publishing Platform; 1 edition (December 17, 2012)

Language: English

ISBN-10: 148112790X

ISBN-13: 978-1481127905

Product Dimensions: 6 x 0.5 x 9 inches

Shipping Weight: 8.5 ounces

Average Customer Review: 4.1 out of 5 stars [See all reviews](#) (24 customer reviews)

Best Sellers Rank: #1,527,588 in Books (See Top 100 in Books) #90 in [Books > Computers & Technology > Programming > Languages & Tools > Assembly Language Programming](#) #483 in [Books > Computers & Technology > Hardware & DIY > Single Board Computers](#) #2563 in [Books > Computers & Technology > Programming > Introductory & Beginning](#)

## Customer Reviews

I thought this was an awful book to learn assembly language usage on a Pi. The author seems to have repackaged some of his other writing on assembly language. The book has nothing to do with a Raspberry Pi. There is not one example dealing specifically with a Pi. No mention of Pi's GPIO, timer, etc. At least I learned how to write an assembly language program to switch an infrared LED, triggering my Nikon D70 camera remotely. But it was not from this book but a "free" tutorial from [...]. I'm a novice looking for help and this book was a disappointment.

I purchased this book and returned it. Most people who buy a Raspberry Pi computer will probably start by installing Raspbian OS, a Raspberry Pi specific version of the Debian Linux distribution. To that end I expected this book to explain how to set up a toolchain to assemble, link, execute and debug programs. I also expected an introduction to the ARM architecture, including an overview of the instruction set with examples of calling system functions, examples showing how to accomplish a variety of tasks such as file and terminal i/o. What this book does provide is instructions to install a single user, cooperative multi-tasking operating system from the late 1980's, RISC OS. (Think

Windows 3.1 or Mac OS 7.) Furthermore rather than using a modern toolchain, or IDE, the user is instructed to use an assembler function of the BASIC interpreter. Now there is nothing wrong with RISC OS, especially if you've used one of the Acorn microcomputers, or BBC Basic. It was very popular in the UK, but relatively unknown in the US. Even if the user does gain a knowledge of assembly language as it relates to the ARM V6 core in the Raspberry Pi, the user doesn't end up with a knowledge of how to use that information in the Linux environment that most people will use. I gave this book a three star rating, because it does do a good job of covering Assembly Language for BBC Basic on RISC OS, it is just not what most people would expect it to be, an Assembly Language introduction for Raspbian OS (Linux).

This book is an excellent introduction to coding in ARM assembler on the Raspberry Pi with RiscOS. The focus is more on the assembler than on the Pi or on RiscOS, but I believe that if you want to get started using ARM assembler there is probably no easier platform to do that than the Pi with RiscOS. The author assumes no specific low-level knowledge and does a great job explaining the basics. Fortunately he does so in a way that an experienced old hack like myself can skip over the introductions and get straight to the meat. I was able to read the book in one flight from Zurich to San Francisco, although the problem with that is that my hands were itching all the time to get some coding done with my newly acquired knowledge :-). Do *not* buy this book if you want to know how to code in Python under Linux on the Pi. *Do* buy this book if you want a great introduction to the ARM architecture, instruction set and assembler. The author promised a follow-up book with more advanced topics which I will surely buy.

There are all kinds of things written about the wonders of learning assembly, as I did decades ago: you're up on the latest ARM technology, which also goes all the way back to the 80's; you finally really know what makes processors tick; you can create much more efficient programs than with Linux or Python... the list goes on. On top of it is the fact that we've pretty much lost a couple generations of machine level programmers due to object oriented, C and other chip requirements. But now for the ugly underbelly. Assembly is not "hard" to learn, but it is a PAIN. It is filled with all kinds of details like stacks, memory allocation, binary, hex, and hoardes of 1's and 0's. If no one else is willing to tell you, I will-- it can be BORING if you don't have a certain personality. So, odds are, some buyers will get this and it will gather dust on their shelves. This is especially true if you have ADHD, like a lot of us programmers ironically do! Any kind of programming skill requires the old "wax on - wax off" continual practice to get it-- many hours of it. With assembly, that means a

lot of tolerance for even more repetitive detail than classes and methods in C#! This book isn't as complete as many other 500 pagers on assembly detail, nor is it as project oriented as, for example, Maxfield's classic Boolean Boogie book--[Bebop to the Boolean Boogie, Third Edition: An Unconventional Guide to Electronics](#). Sure, Raspberry has an ARM core, but you can also get an ARM emulator, a TI SDK, and study the same stuff on your PC. So, The Raspberry, to be honest, is a "hip" way to get folks interested in this title. Not that it isn't well suited for learning-- it is, but not in any special way other than cost. It comes down to your time and focus. It MIGHT be better spent applying Python and Linux, ala [Programming the Raspberry Pi: Getting Started with Python](#) than assembly-- no one has unlimited time. Add the many possibilities of working at the circuit level with Pi/Propeller/Arduino combinations, and there are a LOT of opportunities with this platform, but we can't do them all! There are a lot of typos in this book, but not many in the code thank heavens. The author does create an excellent pace for beginners, and is quite thorough for what is covered. Please see this as a beginning, though, because even though ARM Cortex and assembly are still very relevant, covering them with a single processor doesn't begin to explore the real world of multi processor challenges like race conditions, memory, etc. The flip side is that, if you're just getting into programming, you'll stand out by learning assembly-- as I said above, most folks just don't have the time! As long as it's not at the expense of Java or Python, go for it! Just be aware that the time commitment is not light, and the subject takes big doses of patience and practice. Yes, an easy read, no, not an easy mastery. Caveat: In case you came here for SPECIFIC Raspberry info, this is NOT the book for you. The two (free/downloadable) operating systems for Raspberry (Raspbian and RaspMC, for the GPU-media center and to drive Linux, respectively) are not covered, get the free online documentation and/or a specific Rasp book like: [Raspberry Pi User Guide](#) or [Getting Started with Raspberry Pi \(Make: Projects\)](#). If you're REALLY serious about Assembly and ARM/Cortex regardless of Pi or platform, these are the "bibles" in increasing order of difficulty and completeness:-- Hyde's [The Art of Assembly Language](#)-- Linux/Dandamudi's [Guide to Assembly Language Programming in Linux](#)-- Blum's [Professional Assembly Language](#)-- Streib's [Guide to Assembly Language: A Concise Introduction](#) Since we mentioned Maxfield above, whether you use assembly for math, embedded systems, graphics, etc. matters a lot. In addition to his wonderful Boolean Boogie book, Maxfield also wrote a classic for High School students and hobbyists on how to build a virtual computer/calculator as a "project." When you're done, you know more assembly, without having even realized it due to the fun, than most "DRY" texts! The book is: [The Definitive Guide to How Computers Do Math : Featuring the Virtual DIY Calculator](#). I probably don't have to mention it, but ARM is the old UK "Acorn" system, so it was a natural, as the first RISC based

processor, for the Pi project to adopt it. Even though I mention TI because their SDK is so clean, there are obviously many licensees of the ARM IP, and in fact the Pi uses Broadcom's BCM2835 as its primary ARM peripherals platform-- there are over a dozen 2835s in the Raspberry B, so it would be helpful if you download the free spec sheets on them from Broadcom's website. Unlike TI or Intel, and like Qualcomm for example, Broadcom is a "fabless" manufacturer, meaning they are engineers who license IP like the ARM (especially for handheld devices), but do all "their" manufacturing in China. If you DO plan a career or hobby in embedded (chips/circuits in specific devices, cars, jets, etc.), today's top text using TI's SDK is Valvano's: *Embedded Systems: Introduction to Arm<sup>®</sup> Cortex(TM)-M Microcontrollers (Volume 1)*. Have fun!!!

[Download to continue reading...](#)

Raspberry Pi 3: 2016 Raspberry Pi 3 User Guide (Raspberry Pi, Raspberry Pi 2, Raspberry Pi Programming, Raspberry Pi Projects) (Volume 1) Raspberry Pi Assembly Language RASPBIAN Beginners: Hands On Guide Raspberry Pi Assembly Language RISC OS Beginners (Hands On Guide) Raspberry Pi Assembly Language Beginners: Hands On Guide Gun Digest Book of Firearms Assembly/Disassembly, Part 2: Revolvers (Gun Digest Book of Firearms Assembly/Disassembly: Part 1 Automatic Pistols) (Pt. 2) Kaizen Assembly: Designing, Constructing, and Managing a Lean Assembly Line Arduino: Getting Started With Arduino: The Ultimate Beginner's Guide (Arduino 101, Arduino sketches, Complete beginners guide, Programming, Raspberry Pi 2, xml, c++, Ruby, html, php, Robots) Learn CSS in One Day and Learn It Well (Includes HTML5): CSS for Beginners with Hands-on Project. The only book you need to start coding in CSS ... Coding Fast with Hands-On Project) (Volume 2) Learn C# in One Day and Learn It Well: C# for Beginners with Hands-on Project (Learn Coding Fast with Hands-On Project) (Volume 3) Hello Raspberry Pi!: Python programming for kids and other beginners Digitaltalk PARTS Workbench for Win32 - 32-Bit Parts Assembly and Reuse Tool Set - Script Language Guide - Win32 Series Version 3.0 ARM Assembly Language Programming & Architecture (ARM books) (Volume 1) Z80 Assembly Language Programming Assembly Language and Systems Programming for the M68000 Family Assembly Language Step-by-step: Programming with DOS and Linux (with CD-ROM) Introduction to 64 Bit Intel Assembly Language Programming for Linux: Second Edition Basic IBM Mainframe Assembly Language Programming 80386/80486 Assembly Language Programming Assembly Language for Intel-Based Computers (3rd Edition) Assembly Language for Intel-Based Computers (4th Edition)

[Dmca](#)