

1. Record Nr.	UNINA9910349523003321
Autore	Smith Stephen
Titolo	Raspberry Pi Assembly Language Programming : ARM Processor Coding // by Stephen Smith
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2019
ISBN	1-4842-5287-X
Edizione	[1st ed. 2019.]
Descrizione fisica	1 online resource (380 pages) : illustrations
Collana	Technology in action
Disciplina	794.81526
Soggetti	Computer input-output equipment Hardware and Maker
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Chapter 1: Getting Started -- Chapter 2: Loading and Adding -- Chapter 3: Tooling Up -- Chapter 4: Controlling Program Flow -- Chapter 5: Thanks for the Memories -- Chapter 6: Functions and the Stack -- Chapter 7: Linux Operating System Services -- Chapter 8: Programming GPIO Pins -- Chapter 9: Interacting with C and Python -- Chapter 10: Multiply, Divide and Accumulate -- Chapter 11: Floating Point Operations -- Chapter 12: Neon Coprocessor -- Chapter 13: Conditional Instructions and Optimizing Code -- Chapter 14: Reading and Understanding Code -- Chapter 15: Thumb Code -- Chapter 16: 64-Bits -- Appendix A: The ARM Instruction Set -- Appendix B: Linux System Calls -- Appendix C: Binary Formats -- Appendix D: Assembler Directives -- Appendix E: ASCII Character Set.
Sommario/riassunto	Gain all the skills required to dive into the fundamentals of the Raspberry Pi hardware architecture and how data is stored in the Pi's memory. This book provides you with working starting points for your own projects while you develop a working knowledge of Assembly language programming on the Raspberry Pi. You'll learn how to interface to the Pi's hardware including accessing the GPIO ports. The book will cover the basics of code optimization as well as how to inter-operate with C and Python code, so you'll develop enough background to use the official ARM reference documentation for further projects. With Raspberry Pi Assembly Language Programming as your guide you'll study how to read and reverse engineer machine code and then

then apply those new skills to study code examples and take control of your Pi's hardware and software both.
