



Kostenloses eBook

LERNEN

arm

Free unaffiliated eBook created from
Stack Overflow contributors.

#arm

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit dem Arm	2
Bemerkungen.....	2
Examples.....	2
Varianten.....	2
ARM-Assembly erstellen und ausführen.....	2
Credits	4



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [arm](#)

It is an unofficial and free arm ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official arm.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit dem Arm

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was der Arm ist und warum ein Entwickler ihn verwenden möchte.

Es sollte auch alle großen Themen im Arm erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für arm neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Varianten

Es gibt viele verschiedene Varianten der ARM-Architektur und der Implementierungen, die sich im Laufe der Zeit entwickelt haben. Die Notation kann verwirrend sein. Zum Beispiel *ARM7* und *v Arm 7* sind völlig verschieden. Die erste ist eine CPU-Implementierung. Die zweite ist eine CPU-Architektur. Die *Architektur*, auch *Familie genannt*, ist ein Satz von Maschinenbefehlen (oder *ISA* für die *Befehlssatzarchitektur*), die im Allgemeinen kompatibel sind. Weitere [Informationen finden Sie in der Wikipedia-Liste der ARM-Mikroarchitekturen](#).

Verwandte Tags:

- [thumb](#) - Die erste Version bestand aus einem reduzierten Satz von 16-Bit-Anweisungen. **thumb2**, eingeführt mit armv6, enthält eine Mischung aus 16- und 32-Bit-Befehlen, die in armv7 so erweitert werden, dass sie die meisten Funktionen der normalen ARM-ISA ausführen kann.
- [neon](#) - eine SIMD-Erweiterung für ARM-CPU's
- [cortex-m](#) - eine eingebettete ISA des armv7, die nur **thumb2** unterstützt.
- [cortex-a](#) - die Anwendungsversion von armv7 ISA.
- [arm64](#) - die achte Armarchitektur (armv8) enthält 64-Bit-Register.
- [Vertrauenszone](#) - eine Sicherheitsfunktion in einigen Armv6-, Armv8- und Armv7-CPU's.
- [Amba](#)- Bus- oder Interconnect-Spezifikation für CPU's und Peripheriegeräte.

Andere CPU-spezifische Tags sind vorhanden, beispielsweise [Cortex-M3](#). Häufig beziehen sich diese Posts auf andere *Cortex-m*-CPU's, und / oder der Unterschied zwischen den Versionen ist wichtig zu verstehen. Auch neuere zukünftige CPU's können eine bestimmte CPU erweitern und Fragen in diesem Tag können relevant sein.

Wenn Sie nur im Tag- [Arm](#) posten, geben Sie einige Details zu dem von Ihnen verwendeten System an.

ARM-Assembly erstellen und ausführen

Um ARM-Assembly-Code auszuführen, benötigen Sie eine Maschine mit einem ARM-Prozessor.

Wenn Sie mit Linux arbeiten, können Sie die folgenden Befehle verwenden, um Ihr Programm zu kompilieren:

```
as -o prog_object.o my_prog_source.s
```

Link zum Abrufen der ausführbaren Datei:

```
ld -o run_prog prog_object.o
```

Ausführen mit:

```
./run_prog
```

Wenn Sie `echo $?` ausführen `echo $?` Es wird der in R0 gespeicherte Wert zurückgegeben

Erste Schritte mit dem Arm online lesen: <https://riptutorial.com/de/arm/topic/3630/erste-schritte-mit-dem-arm>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit dem Arm	artless noise , Community , DRUGMONSTER , Sean Houlihane