



**FREE eBook**

# LEARNING

---

## arm

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#arm**

# Table of Contents

About.....	1
Chapter 1: Getting started with arm.....	2
Remarks.....	2
Examples.....	2
Variants.....	2
Build and Run ARM Assembly.....	2
Credits.....	4

---

# About

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](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with arm

## Remarks

This section provides an overview of what arm is, and why a developer might want to use it.

It should also mention any large subjects within arm, and link out to the related topics. Since the Documentation for arm is new, you may need to create initial versions of those related topics.

## Examples

### Variants

There are many different variants of the ARM architecture and implementations that have evolved over time. The notation can be confusing. For instance, *arm7* and *armv7*, are completely different. The first is a CPU implementation; the second is a CPU architecture. The *architecture*, also called a *family*, is a set of machine instructions (or **ISA** for *instruction set architecture*) that are generally compatible. See: [Wikipedia's list of ARM microarchitectures](#) for more.

### Related tags:

- [thumb](#) - the first version consisted of a reduced set of 16bit instructions. **thumb2**, introduced with armv6, includes a mix of 16 and 32bit instructions, extended further in armv7 such that it can do most things the normal ARM ISA can.
- [neon](#) - a SIMD extension for ARM CPUs
- [cortex-m](#) - an embedded ISA of the armv7 that only supports **thumb2**.
- [cortex-a](#) - the application version of armv7 ISA.
- [arm64](#) - the eighth arm architecture (armv8) includes 64bit registers.
- [trust-zone](#) - a security feature in some armv6, armv8 and armv7 CPUs.
- [amba](#) - bus or interconnect specification used between CPUs and peripherals.

Other CPU specific tags exist, such as [cortex-m3](#). Often those posts apply to other *cortex-m* CPUs and/or the difference between the versions is important to understand. Also newer future CPUs may extend a specific CPU and questions in that tag maybe relevant.

If you are only posting to the tag [arm](#) try to give some specifics about the system you are using.

## Build and Run ARM Assembly

To run ARM assembly code you will need a machine with an ARM processor.

If you are on Linux you can use the following commands to compile your program:

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

Link to get the executable:

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

Run using:

```
./run_prog
```

If you run `echo $?` it will return the value stored in R0

Read **Getting started with arm** online: <https://riptutorial.com/arm/topic/3630/getting-started-with-arm>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with arm	<a href="#">artless noise</a> , <a href="#">Community</a> , <a href="#">DRUGMONSTER</a> , <a href="#">Sean Houlihane</a>