



An OS to build, deploy and securely manage billions of devices

Latest News:

Apache Mynewt 1.10.0, Apache NimBLE 1.5.0 (/download) released (May 6, 2022)

[Docs \(/documentation/\)](/documentation/) / [OS User Guide \(../../os_user_guide.html\)](../../os_user_guide.html) / [Porting Mynewt OS \(port_os.html\)](port_os.html) / [Porting Mynewt to a new CPU Architecture](#)

[✎ Edit on GitHub \(https://github.com/apache/mynewt-core/edit/master/docs/os/core_os/porting/port_cpu.rst\)](https://github.com/apache/mynewt-core/edit/master/docs/os/core_os/porting/port_cpu.rst)

Version: latest



[Introduction \(../../index.html\)](../../index.html)

[Setup & Get Started \(../../get_started/index.html\)](../../get_started/index.html)

[Concepts \(../../concepts.html\)](../../concepts.html)

[Tutorials \(../../tutorials/tutorials.html\)](../../tutorials/tutorials.html)

[Third-party Resources \(../../external_links.html\)](../../external_links.html)

[OS User Guide \(../../os_user_guide.html\)](../../os_user_guide.html)

[Kernel \(../mynewt_os.html\)](../mynewt_os.html)

[System \(../../modules/system_modules.html\)](../../modules/system_modules.html)

[Hardware Abstraction \(../../modules/hal/hal.html\)](../../modules/hal/hal.html)

[Secure Bootloader \(../../modules/bootloader/bootloader.html\)](../../modules/bootloader/bootloader.html)

[Split Images \(../../modules/split/split.html\)](../../modules/split/split.html)

[Porting Guide \(port_os.html\)](#)

[BSP Porting \(port_bsp.html\)](#)

[Porting Mynewt to a new MCU \(port_mcu.html\)](#)

[Porting Mynewt to a new CPU Architecture](#)

[Baselibs \(../../modules/baselibs.html\)](#)

[Drivers \(../../modules/drivers/driver.html\)](#)

[Device Management with Newt Manager \(../../modules/devmgmt/newtmgr.html\)](#)

[Device Management with MCUMgr \(../../modules/mcumgr/mcumgr.html\)](#)

[Image Manager \(../../modules/imgmgr/imgmgr.html\)](#)

[Compile-Time Configuration \(../../modules/sysinitconfig/sysinitconfig.html\)](#)

[System Initialization and Shutdown \(../../modules/sysinitdown/sysinitdown.html\)](#)

[Build-Time Hooks \(../../modules/extcmd/extcmd.html\)](#)

[File System \(../../modules/fs/fs.html\)](#)

[Flash Circular Buffer \(../../modules/fcb/fcb.html\)](#)

[Sensor Framework \(../../modules/sensor_framework/sensor_framework.html\)](#)

[Test Utilities \(../../modules/testutil/testutil.html\)](#)

[JSON \(../../modules/json/json.html\)](#)

[Manufacturing support \(../../modules/mfg/mfg.html\)](#)

[Board support \(../../bsp/index.html\)](#)

[BLE User Guide \(../../network/index.html\)](#)

[Newt Tool Guide \(../../newt/index.html\)](#)

[Newt Manager Guide \(../../newtmgr/index.html\)](#)

[Mynewt FAQ \(../../mynewt_faq/index.html\)](#)

[Appendix \(../../misc/index.html\)](#)

Porting Mynewt to a new CPU Architecture

A new CPU architecture typically requires the following:

- A new compiler
- New architecture-specific code for the OS

- Helper libraries to help others porting to the same architecture

These are discussed below:

Create A New Compiler

NOTE: Newt does not automatically install the compilers require to build all platforms. Its up to the user using their local machines package manager to install the compilers. The step described here just registers the compiler with newt.

Create a new directory (named after the compiler you are adding). Copy the `pkg.yml` file from another compiler.

Edit the `pkg.yml` file and change the configuration attributes to match your compiler. Most are self-explanatory paths to different compiler and linker tools. There are a few configuration attributes worth noting.

Configuration Attributes	Description
<code>pkg.keywords</code>	Specific keywords to help others search for this using newt
<code>compiler.flags.default</code>	default compiler flags for this architecture
<code>compiler.flags.optimized</code>	additional flags when the newt tool builds an optimized image
<code>compiler.flags.debug</code>	additional flags when the newt tool builds a debug image

Implement Architecture-specific OS code

There are several architecture-specific code functions that are required when implementing a new architecture. You can find examples in the `sim` architecture within Mynewt.

When porting to a new CPU architecture, use the existing architectures as samples when writing your implementation.

Please contact the Mynewt development list for help and advice portingto new MCU.

⏪ Previous: Porting Mynewt to a new MCU ([port_mcu.html](#))

Next: Baselibc ⏩ ([../../modules/baselibc.html](#))

Apache Mynewt is available under Apache License, version 2.0.



Apache Mynewt, Mynewt, Apache, the Apache feather logo, and the Apache Mynewt project logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and other countries.

