

OpenAMP / open-amp Public

<> Code Issues 23 Pull requests 15 Discussions Actions Projects

16 Branches 22 Tags Go to file Go to file <> Code

arnopo rpmsg: remove deprecated rpmsg_virtio API

89dea1d · 4 days ago

folder	.github	CI: Fix compliance check fold...	4 months ago
folder	cmake	cmake: remove deprecated c...	7 months ago
folder	doc	doxygen: remove obsolete o...	2 months ago
folder	lib	rpmsg: remove deprecated r...	4 days ago
folder	scripts	CI: Fix checkpatch	3 years ago
file	.checkpatch.conf	checkpatch: Increase the def...	4 years ago
file	.gitignore	gitignore: Add cscope files	9 years ago
file	.gitlint	gitlint: add gitlint configuratio...	6 years ago
file	.readthedocs.yaml	docs: update the readthedoc...	last year
file	CMakeLists.txt	cmake: remove possibility to ...	last year
file	LICENSE.md	Documentation: fix license.m...	2 years ago
file	MAINTAINERS.md	MAINTAINERS: Update main...	5 years ago
file	README.md	README: Fix URL format	2 months ago
file	VERSION	release: open-amp 2025.10.0	6 months ago

README License

open-amp

This repository is the home for the Open Asymmetric Multi Processing (OpenAMP) framework project. The OpenAMP framework provides software components that enable development of software applications for Asymmetric Multiprocessing (AMP) systems. The framework provides the following key capabilities.

1. Provides Life Cycle Management, and Inter Processor Communication capabilities for management of remote compute resources and their associated software contexts.
2. Provides a stand alone library usable with RTOS and Baremetal software environments
3. Compatibility with upstream Linux remoteproc and rpmsg components
4. Following AMP configurations supported a. Linux host/Generic(Baremetal) remote b. Generic(Baremetal) host/Linux remote
5. Proxy infrastructure and supplied demos showcase ability of proxy on host to handle printf, scanf, open, close, read, write calls from Bare metal based remote contexts.

OpenAMP Source Structure

```
| - lib/
|   | - virtio/      # virtio implementation
|   | - rpmsg/      # rpmsg implementation
|   | - remoteproc/ # remoteproc implementation
|   | - proxy/      # implement one processor access device on the
|   |               # other processor with file operations
| - cmake          # CMake files
| - scripts        # helper scripts (such as checkpatch) for
contributors.
```



OpenAMP library libopen_amp is composed of the following directories in `lib/` :

- `virtio/`
- `rpmsg/`
- `remoteproc/`
- `proxy/`

OpenAMP system/machine support has been moved to libmetal.

libmetal APIs used in OpenAMP

Here are the libmetal APIs used by OpenAMP, if you want to port OpenAMP for your system, you will need to implement the following libmetal APIs in the libmetal's `lib/system/<SYS>` directory:

- `alloc`, for memory allocation and memory free

- cache, for flushing cache and invalidating cache
- io, for memory mapping. OpenAMP required memory mapping in order to access vring and carved out memory.
- irq, for IRQ handler registration, IRQ disable/enable and global IRQ handling.
- mutex
- shm (For RTOS, you can usually use the implementation from `lib/system/generic/`)
- sleep, at the moment, OpenAMP only requires microseconds sleep as when OpenAMP fails to get a buffer to send messages, it will call this function to sleep and then try again.
- time, for timestamp
- init, for libmetal initialization.
- atomic

Please refer to `lib/system/generic` when you port libmetal for your system.

If you a different compiler to GNU gcc, please refer to `lib/compiler/gcc/` to port libmetal for your compiler. At the moment, OpenAMP needs the atomic operations defined in `lib/compiler/gcc/atomic.h`.

OpenAMP Compilation

OpenAMP uses CMake for library and demonstration application compilation. OpenAMP requires libmetal library. For now, you will need to download and compile libmetal library separately before you compiling OpenAMP library. In future, we will try to make libmetal as a submodule to OpenAMP to make this flow easier.

Some Cmake options are available to allow user to customize to the OpenAMP library for it project:

- **WITH_PROXY** (default OFF): Include proxy support in the library.
- **WITH_VIRTIO_DRIVER** (default ON): Build with virtio driver enabled. This option can be set to OFF if the only the remote mode is implemented.
- **WITH_VIRTIO_DEVICE** (default ON): Build with virtio device enabled. This option can be set to OFF if the only the driver mode is implemented.
- **WITH_VQ_RX_EMPTY_NOTIFY** (default OFF): Choose notify mode. When set to ON, only notify when there are no more Message in the RX queue. When set to OFF, notify for each RX buffer released.
- **WITH_STATIC_LIB** (default ON): Build with a static library.
- **WITH_SHARED_LIB** (default ON): Build with a shared library.
- **WITH_ZEPHYR** (default OFF): Build open-amp as a zephyr library. This option is mandatory in a Zephyr environment.

- **WITH_DCACHE_VRINGS** (default OFF): Build with data cache operations enabled on vrings.
- **WITH_DCACHE_BUFFERS** (default OFF): Build with data cache operations enabled on buffers.
- **WITH_DCACHE_RSC_TABLE** (default OFF): Build with data cache operations enabled on resource table.
- **WITH_DCACHE** (default OFF): Build with all cache operations enabled. When set to ON, cache operations for vrings, buffers and resource table are enabled.
- **RMSG_BUFFER_SIZE** (default 512): adjust the size of the RMsg buffers. The default value of the RMsg size is compatible with the Linux Kernel hard coded value. If you AMP configuration is Linux kernel host/ OpenAMP remote, this option must not be used.

Example to compile OpenAMP for Zephyr

The [Zephyr open-amp repo](#) implements the open-amp library for the Zephyr project. It is mainly a fork of this repository, with some add-ons for integration in the Zephyr project. The standard way to compile OpenAMP for a Zephyr project is to use Zephyr build environment. Please refer to [Zephyr OpenAMP samples](#) for examples and [Zephyr documentation](#) for the build process.

Example to compile OpenAMP for communication between Linux processes:

- Install libsysfs devel and libhugetlbfs devel packages on your Linux host.
- build libmetal library on your host as follows:

```
$ mkdir -p build-libmetal
$ cd build-libmetal
$ cmake <libmetal_source>
$ make VERBOSE=1 DESTDIR=<libmetal_install> install
```



- build OpenAMP library on your host as follows:

```
$ mkdir -p build-openamp
$ cd build-openamp
$ cmake <openamp_source> -DCMAKE_INCLUDE_PATH=
<libmetal_built_include_dir> \
    -DCMAKE_LIBRARY_PATH=<libmetal_built_lib_dir>
$ make VERBOSE=1 DESTDIR=$(pwd) install
```



The OpenAMP library will be generated to `build/usr/local/lib` directory, and the headers will be generated to `build/usr/local/include` directory.

Example apps and tests

- The [openamp-system-reference](#) is a new repository for the OpenAMP demos.

Documentation

OpenAMP project documentation is available at: <https://openamp.readthedocs.io/en/latest/>

How to contribute:

As an open-source project, we welcome and encourage the community to submit patches directly to the project. As a contributor you should be familiar with common developer tools such as Git and CMake, and platforms such as GitHub. Then following points should be respected to facilitate the review process.

Licencing

Code is contributed to the Linux kernel under a number of licenses, but all code must be compatible with version the [BSD License](#), which is the license covering the OpenAMP distribution as a whole. In practice, use the following tag instead of the full license text in the individual files:

```
SPDX-License-Identifier: BSD-3-Clause  
SPDX-License-Identifier: BSD-2-Clause
```



Signed-off-by

Commit message must contain Signed-off-by: line and your email must match the change authorship information. Make sure your `.gitconfig` is set up correctly:

```
git config --global user.name "first-name Last-Namer"  
git config --global user.email "yourmail@company.com"
```



gitlint

Before you submit a pull request to the project, verify your commit messages meet the requirements. The check can be performed locally using the `gitlint` command.

Run gitlint locally in your tree and branch where your patches have been committed:

```
gitlint
```



Note, gitlint only checks HEAD (the most recent commit), so you should run it after each commit, or use the --commits option to specify a commit range covering all the development patches to be submitted.

Code style

In general, follow the Linux kernel coding style, with the following exceptions:

- Use `/** */` for doxygen comments that need to appear in the documentation.

The Linux kernel GPL-licensed tool checkpatch is used to check coding style conformity. Checkpatch is available in the scripts directory.

To check your `<n>` commits in your git branch:

```
./scripts/checkpatch.pl --strict -g HEAD-<n>
```



Send a pull request

We use standard github mechanism for pull request. Please refer to github documentation for help.

Communication and Collaboration

Releases 21

 **OpenAMP Release 2025.10.0 - v1.9.0** Latest
on Nov 3, 2025

[+ 20 releases](#)

Packages

No packages published

Contributors 54



[+ 40 contributors](#)

Languages

