armbian / **build**

Armbian Linux build tools

www.armbian.com

GPL-2.0 License

1.6k stars **925** forks

Star Watch

Code Issues 34 Pull requests 8 Actions Projects Security ...

master ...

igorpecovnik ... 2 days ago

View code

README.md

build tools

Build passing follow @armbian 3.9k

Table of contents

What this project does?

- builds custom Debian based Linux system optimized for supported single board computers,
- covers root filesystem generation, kernel image compilation and bootloader compilation,
- maintains low-level control software for a selection of hardware,
- provides a consistent user experience by keeping system standards across different SBC platforms.

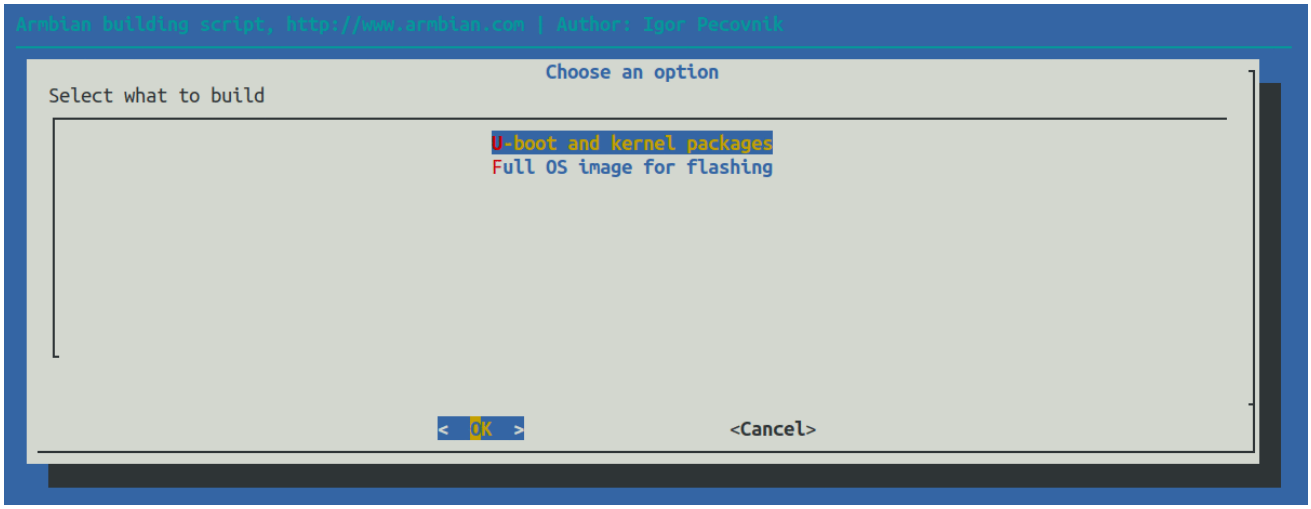
What do you need to get started?

- x64 machine with at least 2GB of memory and ~35GB of disk space for the VM, container or native OS,
- Ubuntu Focal 20.04 x64 for native building or any Docker capable x64 Linux for containerised,
- superuser rights (configured sudo or root access).

⇧

How to build an image or a kernel?

```
apt-get -y install git
git clone https://github.com/armbian/build
cd build
./compile.sh
```



- The script will take care about preparing the workspace like installing necessary dependencies and downloading sources and tools.
- It guides through the process until either a kernel package set or a readyo-to-image for a sdcard is created.

⇧

Build parameter examples

Show work in progress areas in interactive mode:

```
./compile.sh EXPERT="yes"
```

Run build tools inside Docker container:

```
./compile.sh docker
```

Build minimal CLI Debian buster based image for Odroid XU4. Use modern kernel and write image to the SD card:

```
./compile.sh BOARD="odroidxu4" BRANCH="current" RELEASE="buster"
CARD_DEVICE="/dev/sda" \
KERNEL_ONLY="no" KERNEL_CONFIGURE="no" BUILD_DESKTOP="no"
BUILD_MINIMAL="yes"
```

Build parameters, advanced build options, user defined configuration, build with Docker?

⇧

Compare with industry standards

Check similarity, advantages and disadvantages compared with leading industry standard build software.

Function	Armbian	Yocto	Buildroot
Target	general purpose	embedded	embedded / IOT
U-boot and kernel	compiled from sources	compiled from sources	compiled from sources
Board support maintenance	complete	outside	outside
Root file system	Debian or Ubuntu based	custom	custom
Package manager	APT	any	none
Configurability	limited	large	large
Initramfs support	yes	yes	yes
Getting started	quick	very slow	slow
Cross compilation	yes	yes	yes

⇧

Where to download prebuilt images?

https://www.armbian.com/download/

Armbian releases quarterly at the end of February, May, August, November. You are welcome to propose changes to our default images build list.

⇧

Additional information

- Build parameters and advanced build options,
- Make use of user defined configurations,
- Docker and Vagrant building guides,
- Engage in Armbian build framework forums,
- Check Jira project management application status,
- Make use of central project search engine,
- Browse IRC channel logs or interact at #armbian on freenode.

⇧

Build tools overview

```
├── cache                          Work / cache directory
│   ├── rootfs                     Compressed vanilla Debian and
Ubuntu rootfilesystem cache
│   ├── sources                    Kernel, u-boot and various drivers
sources. Mainly C code
│   ├── toolchains                 External cross compilers from
Linaro™ or ARM™
├── config                         Packages repository configurations
│   ├── targets.conf               Board build target configuration
│   ├── boards                     Board configurations
│   ├── bootenv                    Initial boot loaders environments
per family
│   ├── bootscripts                Initial Boot loaders scripts per
family
│   ├── kernel                     Kernel build configurations per
family
│   ├── sources                    Kernel and u-boot sources
locations and scripts
│   ├── templates                  User configuration templates which
populate userpatches
│   └── torrents                   External compiler and rootfs cache
torrents
├── lib                            Main build tools libraries
├── output                         Build artifact
│   └── deb                        Deb packages
│   └── images                     Bootable images - RAW or
compressed
│   └── debug                      Patch and build logs
│   └── config                     Kernel configuration export
location
│   └── patch                      Created patches location
├── packages                       Support scripts, binary blobs,
packages
│   ├── blobs                      Wallpapers, various configs,
closed source bootloaders
│   ├── bsp                        Scripts and configs overlay for
rootfs
│   └── extras-buildpkgs           Optional compilation and packaging
engine
├── patch                          Collection of patches
│   ├── atf                        ARM trusted firmware
│   ├── kernel                     Linux kernel patches
│   │   └── family-branch          Per kernel family and branch
│   ├── misc                       Linux kernel packaging patches
│   └── u-boot                     Universal boot loader patches
│       ├── u-boot-board           For specific board
│       └── u-boot-family          For entire kernel family
└── userpatches                    User: configuration patching area
```

```
    ├── lib.config                  User: tools common config/override
file
    ├── config-default.conf         User: default user config file
    ├── customize-image.sh          User: script will execute just
before closing the image
    ├── atf                         User: ARM trusted firmware
    ├── kernel                      User: Linux kernel per kernel
family
    ├── misc                        User: various
    └── u-boot                      User: universal boot loader
patches
```

⇧

Support

- Have you found a bug in the **build tools**?

  Try to recreate it with a clean build tools clone. Then search for existing and closed issues. If you don't find it there, open a new issue.

- Do you have troubles **elsewhere**?

  Armbian is free software and provides **best effort help** through community forums. If you can't find answer there and/or with help of general project search engine and documentation, consider hiring an expert.

- Personalised support?

  It is limited to active project supporters and sponsors. The shortest way to become one and receive our attention is a four figure donation to our non-profit project.

⇧

Contribute

- Adding a new feature?

  You are welcome to suggest or contribute directly to the code with a pull request. In case your proposed changes are large, remember to discuss them prior to development.

- Join development?

  Join regulars on their active missions, start maintaining any part of the code: patches, drivers or scripted applications like armbian-config, address community wishes,

- Help elsewhere?

Maintain and develop documentation, CI, autotests, seed torrents, help on forum moderating, project administration, costs.

⇧

Social

- Participate in Armbian forums,
- Chat with fellow users on IRC #armbian on Freenode,
- Follow @armbian on Twitter or LinkedIN.

⇧

Credits

- Current and past contributors, our families and friends,
- Support staff that keeps forums usable,
- Individuals that help with their ideas, reports and donations.

⇧

Sponsors

Most of the project is sponsored with a work done by volunteer collaborators, while some part of the project costs are being covered by the industry. We would not be able to get this far without their help.

Do you want to see yourself below?

⇧

Sponsor this project

https://www.armbian.com/donate

Contributors 283

+ 272 contributors

Languages

Shell 76.4% **Roff** 9.5% **Batchfile** 5.1% **PHP** 3.0% **HTML** 2.5% **Makefile** 1.7%
Other 1.8%