



THE LINUX FOUNDATION



NORTH AMERICA

Easy Bring-up Your RISC-V SBC Using Yocto Project – RISC-V Architecture Layer

Khem Raj

18-May-2026



Agenda

- What is Yocto Project
- Yocto Project layer Architecture
- RISC-V support in Core
- What is meta-riscv
- Adding new machines/SBC support
- CI/CD
- Support meta-riscv

Yocto Project

“The Yocto Project is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.”

“The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IOT devices, or anywhere a customized Linux OS is needed.”

Yocto Project Layered Architecture

- Yocto project uses OpenEmbedded Build system
- Uses layered architecture
 - oe-core – Core packages, toolchains, abstractions, qemu machines
 - Meta-openembedded – Extended package recipes divided in layers e.g. meta-oe
 - BSP layers – Add machine support e.g. meta-raspberrypi, meta-qcom etc.
 - ISV layers – Middleware software infrastructures e.g. meta-qt6, etc.
 - Language support – Add support for language runtime/packages e.g. meta-perl, meta-python

RISC-V support in Core layer

- OE-Core supports multiple CPU architectures
 - X86_64, ARM, ARM64, RISCV32, RISCV64
- RISCV support is newer entrant
- Supports QEMU machine targets. qemuriscv64, qemuriscv32
- Runs Yocto CI and package tests (ptest)

What is meta-riscv

- meta-riscv is RISC-V architecture layer for yocto project
- Hosts machine support for Open SBCs
- Serves as sample template for writing BSP layers for RISC-V based SOCs
- Hosts support patches which are needed and are in-flight upstream
- Can be used to provide hardware testing
- Supports vendor kernel and mainline kernel (preferred)
- Tries to promote common software for BSPs e.g. upstream OpenSBI
- Provided Kas templates for project workspace setups
- WIC templates for monolithic image generation

meta-riscv architecture scope

- RV32 and RV64, bare-metal to desktop
- Some packages explicitly incompatible with riscv32 (klibc, lttng-tools, strace, android-tools) — tracked in layer.conf

| Target | Details |
|------------|---|
| riscv64 | 64-bit Linux, primary focus |
| riscv32 | 32-bit (bare-metal + Milk-V Duo) |
| Bare-metal | baremetal-riscv32, baremetal-riscv32nf, baremetal-riscv64 |
| QEMU | qemuriscv64 / qemuriscv32 – enhanced from OE-Core base |

Supported Hardware Machines

- A wide and growing board ecosystem

| SoC | Board(s) |
|-------------------|-----------------------------|
| StarFive JH7110 | VisionFive 2, PINE64 Star64 |
| SpacemiT K1 | BananaPi F3 |
| Allwinner D1/D1-H | Nezha, MangoPi MQ Pro |
| T-Head TH1520 | BeagleV-Ahead |
| ESWIN EIC7700X | Milk-V Megrez |
| Sophgo CV1800B | Milk-V Duo (RV32+RV64) |
| SiFive U540 | HiFive Unleashed (legacy) |
| Andes AX45MP | AE350 platform |
| OrangePi | RV2, R2S |

- BeagleV StarLight (JH7100) and VisionFive 1 — deprecated in Yocto 6.1

meta-riscv architecture

- What the layer provides

meta-riscv/

| | |
|-----------------------|---|
| ——— conf/machine/ | # 21 machine .conf files |
| ——— recipes-bsp/ | # OpenSBI, U-Boot, board firmware |
| ——— recipes-kernel/ | # Per-board Linux kernel recipes |
| ——— recipes-devtools/ | # RISC-V tooling (OpenOCD, riscv-tools) |
| ——— dynamic-layers/ | # Optional: GNOME, multimedia, Xvisor |
| ——— kas/ | # KAS build configs per board |

- Hard dependencies: openembedded-core, bitbake
- Optional: meta-oe, meta-virtualization, meta-gnome, meta-multimedia

Build system Integration

- KAS for reproducible builds

```
kas build kas/visionfive2.yml
kas build kas/bananapi-f3.yml
kas build kas/qemuriscv64.yml
```

- Every board has a kas/*.yml config
- All inherit base-riscv.yml: layer stack + poky-altcfg distro
- Default image: core-image-full-cmdline
- WIC partition layouts for SD card images

Adding new BSP support

- New machines added in meta-riscv
 - Should be readily available hardware
- Use proprietary layer modelled on meta-riscv
- Extend meta-riscv downstream

CI/CD

- Openembedded-core supports RISC-V for ptests
- meta-riscv could extend ptests to H/W platforms
- Use resulttool to report results back to Yocto CI runs

CI Quality

- Automated compatibility checks
 - GitHub Actions nightly CI — layer compatibility checking against OE-Core master
 - Runs on every pull request
 - Catches API/recipe breakage before it lands
- Deprecation tracking
 - DEPRECATED.md lists machines being phased out with target release dates
 - Gives downstream users advance notice

Future Roadmap

- Yocto 6.1 — JH7100 machines fully removed
- Continued mainline kernel migration (vendor → mainline variants for more boards)
- Expanding SpacemiT K1 support (BananaPi F3 mainline kernel)
- Nightly CI coverage expansion
- Growing the board portfolio as RISC-V silicon matures

Meta-riscv Support

- It's a community layer
- Submit patches as github PRs
- File Github issues
- Participate in discussions on github discussions

Thank you

