RuyiSDK - An Integrated and Customizable Toolkit for RISC-V Software Development

Weilin Cai[1], Yilin Chen, Yunxiang Luo
Programming Language and Compiler Technology Lab, Institute of Software, Chinese Academy of Sciences (ISCAS)
Email: {caiweilin,luoyunxiang}@iscas.ac.cn

Introduction and Motivation

- The modular design of the RISC-V instruction set has fostered a highly diverse ecosystem. However, the introduction of vendor-defined extensions has led to potential fragmentation. For SoCs based on specific CPU cores, achieving optimal performance often require customized toolchains, emulators, firmware, and operating systems.
- To save on maintenance costs, vendors typically only maintain specific versions of toolchains and emulators, rather than actively continuing maintenance or pushing the code upstream.
- RuyiSDK's goal is to provide a comprehensive, one-stop solution tailored for RISC-V developers. It aims to address these challenges by integrating foundational software, promoting adaptation of unsupported applications, and cultivating a vibrant developer community.
- RuyiSDK maintains GCC and LLVM forks, providing open-source vendor extension support and pushing patches upstream. It also maintains distributions such as RevyOS, based on Debian, to adapt and continuously support development boards using specific SoCs. The aim is to provide a stable and upstream-evolving software ecosystem for all development boards.
- The software provided by RuyiSDK, together with upstream and vendor-provided tools, forms a large and complex ecosystem. For developers and enthusiasts new to RISC-V, this ecosystem can be overwhelming. The Ruyi Package Manager and its Package Index will make everything more accessible and developer friendly.

Methodologies

- The design of the Ruyi Package Manager is illustrated in Figure 1. It clones the Ruyi Packages Index to the local system or updates it based on an existing cache. Users can then select and download the required packages online and perform offline development using the Ruyi Virtual Environment, or flash system images to development boards using the Device Provisioning Wizard.

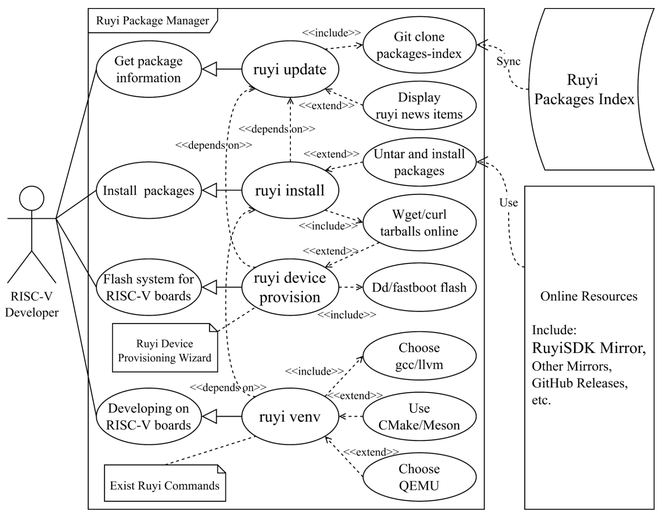


Figure 1: Design of Ruyi Package Manager

- The Ruyi Device Provisioning Wizard provides an interactive Q&A interface that helps users flash system images onto their devices and access relevant development resources. When users receive a new RISC-V development board, this feature assists them in obtaining the latest firmware and system updates. If a development board can be connected to the host machine in a supported manner and somehow accurately identified, the wizard is able to recognize the board and list the latest available firmware and system images in the world. Alternatively, the user can manually specify the board model to initiate the process.
- Once the development board has been updated and successfully booted, RISC-V software development can begin. Developers can use the Ruyi Package Manager to install the appropriate toolchain and create a Ruyi Virtual Environment. In a typical cross-development workflow, source packages are extracted into the workspace, built using GCC, and tested with emulators. For specific SoCs, extra arguments are required to enable or disable instruction set extensions. Ruyi Packages Index includes a set of entity lists that record information about development boards, SoCs, toolchains, and microarchitectures, allowing developers to create a Ruyi Virtual Environment and begin development simply by specifying the model of the board.
- This process demonstrates that the Ruyi Packages Index serves a far greater role than a typical package mirror. Its design is shown in Figure 2. Manifests show that Ruyi packages fall into six categories: toolchain, emulator, source tarball, board image, analyzer, and extra package. Entities are four sets of metadata of RISC-V boards, especially for create Ruyi Virtual Environment.

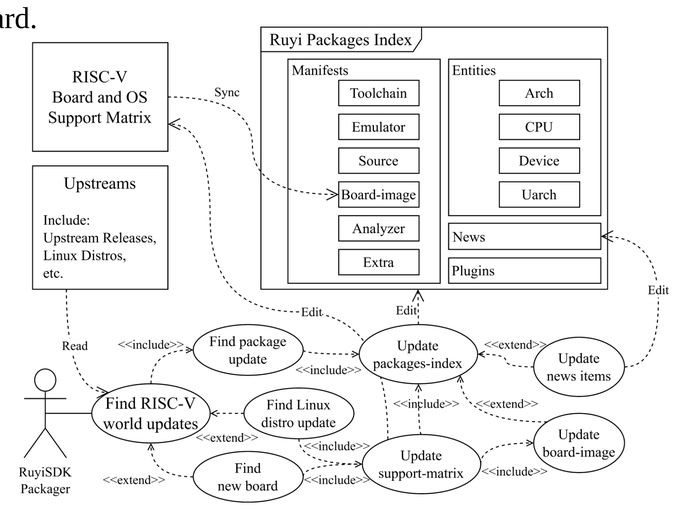


Figure 2: Design of Ruyi Packages Index

- For maintainability, the actual binary repositories and the package index are separated. The Ruyi Packages Index is just a set of TOML configuration files, which includes not only download links but also metadata for Ruyi Virtual Environment, the Ruyi Device Provisioning Wizard, plugins and more.
- Most of these packages are distributed as binaries released by RuyiSDK or other organizations. Toolchains and emulators include upstream versions, vendor-specific versions and RuyiSDK maintained versions, to ensure broad support for various development boards. Meanwhile, board image packages are synchronized with another RuyiSDK subproject called the RISC-V Board and OS Support Matrix, which ensures alignment with upstream releases while also providing test reports and compatibility assessments. Since the Ruyi Packages Index consists of plain text files, it can be easily managed using a Git repository, avoiding the need to redistribute vendor-released binaries.

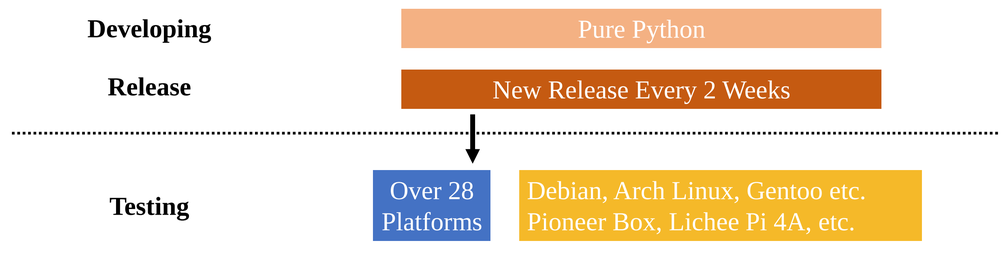
RuyiSDK Website

Ruyi Package Manager Download

RuyiSDK RISC-V Board and OS Support Matrix

Result

- The Ruyi Package Manager has already implemented its core functions and is still under rapid development and iteration. It releases a new version every two weeks and tests on over eight Linux distributions and three architectures (x86_64, riscv64, and aarch64), supports most Linux distributions via Nuitka binaries.



- The Ruyi Package Index currently provides over 60 packages for 30 different development boards. RuyiSDK has already achieved comprehensive support for RISC-V development board from MilkV and Sipeed, offering not only dedicated toolchains, emulators, and system images, but also source code demos. In adittion, some toolchains maintained by RuyiSDK provide support not yet fully available in upstream projects, for example, gnu-plct-rv64ilp32-elf supports the rv64ilp32 ABI, and llvm-plct includes support for the xtheadvector extension.

```
[hachi@youmu-sz6 ~]$ ruyi venv -t gnu-plct milkv-duo myon-venv
info: Creating a Ruyi virtual environment at myon-venv...
info: The virtual environment is now created.

You may activate it by sourcing the appropriate activation script in the
bin directory, and deactivate by invoking `ruyi-deactivate`.

A fresh sysroot/prefix is also provisioned in the virtual environment.
It is available at the following path:

    /home/hachi/myon-venv/sysroot

The virtual environment also comes with ready-made CMake toolchain file
and Meson cross file. Check the virtual environment root for those;
comments in the files contain usage instructions.

[hachi@youmu-sz6 ~]$ . myon-venv/bin/ruyi-activate
«Ruyi myon-venv» [hachi@youmu-sz6 ~]$
```

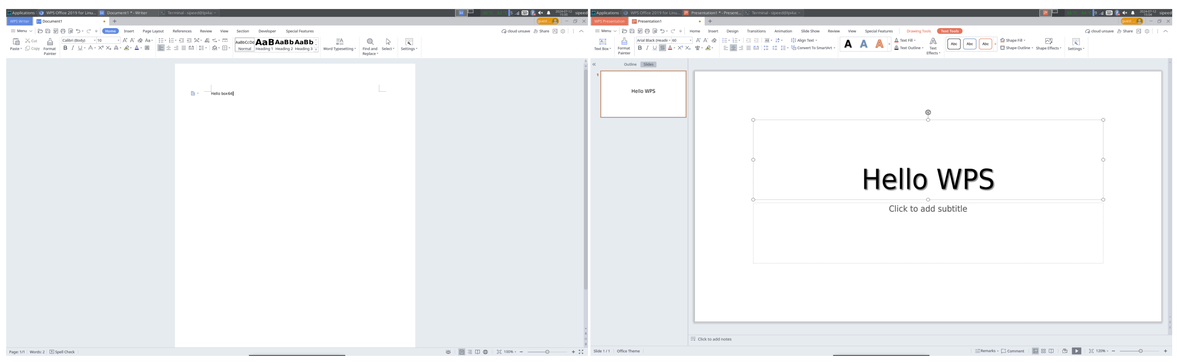
```
[hachi@youmu-sz6 ~]$ . myon-venv/bin/ruyi-activate
«Ruyi myon-venv» [hachi@youmu-sz6 ~]$ riscv64-plct-linux-gnu-gcc hello-ruyi.c -O2 -o
hello-gnu-plct
«Ruyi myon-venv» [hachi@youmu-sz6 ~]$ file hello-gnu-plct
hello-gnu-plct: ELF 64-bit LSB executable, UCB RISC-V, RVC, double-float ABI, version
 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-riscv64-lp64d.so.1, BuildID[
sha1]=644450dd59e8c7ef2f9ad8abf42afa80693f95f5, for GNU/Linux 4.15.0, with debug_info
, not stripped
«Ruyi myon-venv» [hachi@youmu-sz6 ~]$ ruyi-deactivate
[hachi@youmu-sz6 ~]$
[hachi@youmu-sz6 ~]$ . duo-venv/bin/ruyi-activate
«Ruyi duo-venv» [hachi@youmu-sz6 ~]$ riscv64-unknown-linux-musl-gcc hello-ruyi.c -O2
-o hello-milkv-musl
«Ruyi duo-venv» [hachi@youmu-sz6 ~]$ file hello-milkv-musl
hello-milkv-musl: ELF 64-bit LSB executable, UCB RISC-V, RVC, double-float ABI, versi
on 1 (SYSV), dynamically linked, interpreter /lib/ld-musl-riscv64.so.1, with debug_in
fo, not stripped
```

- With the Ruyi Virtual Environment, users do not need to manually specify -mcpu, -march, or -mabi parameters for a target RISC-V board. In recent versions, manual selection is still required for toolchain and emulator used in a virtual environment, but future versions will utilize the entity list to assist with this process. The Ruyi Packages Index already includes entity lists for 33 devices and 19 microarchitectures, ranging from the generic rv64gc, Sifive U74, Xuantie C910, to the currently uncommon Xiangshan Nanhu architecture.

```
RuyiSDK Device Provisioning Wizard

This is a wizard intended to help you install a system on your device for your
development pleasure, all with ease.

You will be asked some questions that help RuyiSDK understand your device and
your intended configuration, then packages will be downloaded and flashed onto
the device's storage, that you should somehow make available on this host
system beforehand.

Note that, as Ruyi does not run as root, but raw disk access is most likely
required to flash images, you should arrange to allow your user account sudo
access to necessary commands such as dd. Flashing will fail if the sudo
configuration does not allow so.

Continue? (y/N) y

The following devices are currently supported by the wizard. Please pick your device:

  1. Allwinner Nezha D1
  2. Canaan Kendryte K230
  3. Canaan Kendryte K230D
  4. Canaan Kendryte K510
  5. Milk-V Duo
  6. Milk-V Duo S
```

```
  6. Milk-V Duo S
  7. Milk-V Mars
  8. Milk-V Mars CM
  9. Milk-V Meles
 10. Milk-V Pioneer Box
 11. Milk-V Vega
 12. Pine64 Star64
 13. SiFive HiFive Unmatched
 14. Sipeed Lichee Cluster 4A
 15. Sipeed Lichee Console 4A
 16. Sipeed LicheePi 4A
 17. Sipeed Lichee RV
 18. Sipeed LicheeRV Nano
 19. Sipeed Maix-I
 20. Sipeed Tang Mega 138K Pro
 21. StarFive VisionFive
 22. StarFive VisionFive2
 23. WCH CH32V103 EVB
 24. WCH CH32V203 EVB
 25. WCH CH32V208 EVB
 26. WCH CH32V303 EVB
 27. WCH CH32V305 EVB
 28. WCH CH32V307 EVB
 29. WCH CH582F EVB
 30. WCH CH592X EVB
```

- The Ruyi Device Provisioning Wizard now allows users to flash (via dd or fastboot) the latest images to RISC-V boards and access embedded board development resources, with the assurance of image quality verified through test reports in the RISC-V Board and OS Support Matrix. Users can simply specify the model and variant of the development board to get a list of supported operating systems. Note that dd images to MicroSD card usually requires root privileges. The Ruyi Device Provisioning Wizard will only invoke sudo after obtaining the user's consent.



- Another challenge in promoting RISC-V boards is that closed-source software is often released only for x86_64 architecture. The Ruyi Package Index offers Box64 package which can run with systemd-binfmt, enabling users to easily configure environments for running Linux applications across architectures or using Wine to run Windows applications on RISC-V systems.

Conclusion

- Ruyi Package Manager has demonstrated wide compatibility across platforms and distributions, supporting both native and cross-development for RISC-V. Its powerful features, such as virtual environments, device provisioning, plugin extensibility, and user customizability, make it a practical and scalable toolkit for the growing RISC-V software ecosystem. It enables developers to focus on innovation and helps expand the influence of RISC-V, contributing to its development as a mainstream architecture.

Future work

- Although the Ruyi Package Manager is highly usable, several improvements are needed. It lacks traditional upgrade and uninstall mechanisms and has no package dependency tracking. File-level management may be required to handle conflicts when multiple toolchains share the same virtual environment. The Ruyi Packages Index also lacks automatic upstream version detection and packaging automation. Additionally, the command-line interface can present usability challenges for beginners. Ongoing work on the RuyiSDK IDE aims to reduce the learning curve and improve overall convenience for developers, ultimately contributing to a more accessible and complete development experience.