Renode - a flexible simulator for CI in complex embedded systems

Linux Plumbers Conference, 2020-08-27
Michael Gielda, mgielda@antmicro.com
ANTMICRO

- Founded 2009, developing Renode since 2010
- We use, develop, advocate open source
- Member of Linux Foundation, Zephyr Project, Platinum Founding member of RISC-V International and OpenPOWER Foundation
- Turning ideas into software-driven products
- New design methodologies based on open source
- Embedded systems: AI/ML, autonomous vehicles, robotics, IoT, defense, aerospace, industrial automation
What is Renode?

Open source hardware-simulation framework for:

- Development of complex software for embedded and IoT systems
- Architectural exploration and research
- Pre-silicon prototyping and HW-SW co-development

Features in brief:

- Plug-and-play building blocks
- Simulates system on many levels - CPUs, SoCs, peripherals, sensors, wired/wireless connection
- Flexible, deterministic and software-agnostic
- Continuous Integration-oriented
- “Batteries included” - lots of demos and binaries
Simulate your system on many levels
Layer #1: System-on-Chip

- SoC diagram

- CAN
- Ethernet
- RAM
- I2C
- SPI
- UART
- USB
- CPU
  - Cortex-A7 Core
  - Cortex-M4 Core
- LCD
- Flash
Layer #2: The device
Layer #3: Complex system

- Sensor nodes
- Gateway
## Supported platforms/vendors

<table>
<thead>
<tr>
<th>LEON3</th>
<th>STMicroelectronics</th>
<th>QuickLogic</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORDIC Semiconductor</td>
<td>Microchip</td>
<td>NXP</td>
</tr>
<tr>
<td>Silicon Labs</td>
<td>Texas Instruments</td>
<td>ZYNQ</td>
</tr>
<tr>
<td>SiFive</td>
<td>RISC-V</td>
<td>OpenPOWER</td>
</tr>
</tbody>
</table>

By antmicro
Supported boards
Platform description format

- human readable
- modular
- extendible
- enable new boards / platforms w/o coding

uart: UART.MiV_CoreUART @ sysbus 0x70001000
clockFrequency: 66000000

cpu: CPU.RiscV @ sysbus
cpuType: "rv32g"

plic: Interrupts.PlatformLevelInterruptController @ sysbus 0x40000000
IRQ -> cpu@1
numberOfSources: 31 //based on release notes
Various integrations

<table>
<thead>
<tr>
<th>GitLab</th>
<th>Wireshark</th>
<th>Robot</th>
<th>Jenkins</th>
</tr>
</thead>
</table>

| GDB     | Verilator | USB/IP |
Various integrations

- Verilator for HW-SW co-development ([note](#) with Google on our blog)
- PlatformIO integration ([note](#) on Linux Foundation blog)
- TensorFlow Lite support ([note](#) with Google on TF Lite blog)
- Robot Framework, Jenkins, GitLab CI, GitHub Actions (e.g. [blog note](#) by Memfault)
- protocol / stack testing: OPC-UA, TSN, 6lowpan, Thread etc.
Renode & Linux

Renode enables Linux development through supported architectures:
Pretty neat that with @renode on my laptop, I can emulate Linux running on a RISC-V processor including a framebuffer to show that adorable Tux

github.com/renode/renode/...
Renode’s Linux-Capable Platforms and Demos

- Kendryte (Linux-nommu)
- LiteX/VexRiscv
- HiFive Unleashed (buildroot)
- PolarFire SoC Icicle Kit (Yocto/buildroot)
- Zedboard
- Vybrid
- Versatile
- Versatile Express
- Nvidia Tegra 3
VexRiscv

- VexRiscv - best performing soft (FPGA) RISC-V implementation
- super scalable, from Lattice iCE40 to big FPGAs
- winner of 2018 RISC-V soft CPU contest
- multi-core, Linux-capable platform (blog note), lots of I/O & features supported in Renode
PolarFire SoC

- 1st mass-produced Linux-capable RISC-V implementation
- Early adopters of the platform
- Pre-silicon access to it in Renode
- All peripherals supported
- Affordable and powerful
- Already generating lots of interest and expected to be very successful
Supported boards

Renode supports a wide array of hardware platforms, covering multiple architectures, CPU families and providing various I/O capabilities.

This chapter contains an (incomplete) list of selected supported hardware targets - all of these include sample software binaries that run both on real hardware and in Renode.

To run example software on any of the below boards, simply run Renode and use:

```
@scripts/PATH/TO/SCRIPT-NAME.resc
```

Tab completion is available also for filenames, so be sure to explore the available demos.

The ultimate goal of Renode is to run any binary-compatible software targeted for any of those hardware platforms without modification, although of course your specific use case may require extending the provided hardware description / models.

Supported boards include:

- QuickFeather Development Kit
- OpenPOWER Microwatt on Digilent Nexys Video
- Microchip PolarFire SoC Icicle Kit
Renode and networking

- multi-node capability & determinism open up IoT testing use cases
- simple networking medium simulation (e.g. distance-based packet loss)
- enables testing protocols and stacks (OPC-UA, Zephyr)
- integrate with regular network testing tools
Continuous Integration

- CI is used extensively at Antmicro and our customers
- SW building, testing and deploying using isolated reproducible containerized environments, such as Docker and Singularity
- Automated build pipelines that can be executed by multiple parties

- Gives reproducible and traceable result
- Simplifies testing new changes
- Eliminates points of failure while working on multiple devices in different locations
- Can be used to build multiple product variants, entire product lines
Renode-based Continuous Integration workflow for IoT systems

- Company Environment
- Local PC
- Interactive test and debug in Renode
- Get help from colleagues
- Commit code
- Develop with favorite IDE/compiler
- Tests with various configurations
- Tests pass?
- Merge changes
- Field tests / deployment
- Push to server
- CI e.g. with Robot + Renode
Testing / CI

- Easy to create new test cases in Renode thanks to Robot Framework integration
- Tested software includes:
  - Linux
  - FreeRTOS
  - Zephyr
  - RIOT
  - TensorFlow Lite
  - Mbed
  - NuttX
- Tock
- micropython
- Android
- Bare Metal software
- Contiki
- Wolfboot
- eCos
- Redboot
- and more!
<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiteX_i2C_Zephyr.cs</td>
<td>Reduce verbosity of the demo</td>
<td>3 months ago</td>
</tr>
<tr>
<td>angle.data</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>circle.data</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>litevexriscv-tflite.repl</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>litevexriscv-tflite.resc</td>
<td>README: Use locally-built binaries when running Renode</td>
<td>8 days ago</td>
</tr>
<tr>
<td>litevexriscv-tflite.robot</td>
<td>Remove redundant ADXL345 model</td>
<td>3 months ago</td>
</tr>
<tr>
<td>File</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>LiteX_I2C_Zephyr.cs</td>
<td>Reduce verbosity of the demo</td>
<td>3 months ago</td>
</tr>
<tr>
<td>angle.data</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>circle.data</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>litevexriscv-tlite.repl</td>
<td>Add Renode section</td>
<td>9 months ago</td>
</tr>
<tr>
<td>litevexriscv-tlite.resc</td>
<td>README: Use locally-built binaries when running Renode</td>
<td>8 days ago</td>
</tr>
<tr>
<td>litevexriscv-tlite.robot</td>
<td>Remove redundant ADXL345 model</td>
<td>3 months ago</td>
</tr>
</tbody>
</table>
mateusz-holenko: README: Use locally-built binaries when running Renode

LiteX_I2C_Zephyr.cs: Reduce verbosity of the demo

angle.data: Add Renode section

circle.data: Add Renode section

litex-vexriscv-tf-lite.repl: Add Renode section

litex-vexriscv-tf-lite.resc: README: Use locally-built binaries when running Renode

litex-vexriscv-tf-lite.robot: Remove redundant ADXL345 model
mateusz-holenko

README: Use locally-built binaries when running Renode

- LiteX_I2C_Zephyr.cs
  - Reduce verbosity of the demo
- angle.data
  - Add Renode section
- circle.data
  - Add Renode section
- litevxriscv-tflite.repl
  - Add Renode section
- litevxriscv-tflite.resc
  - README: Use locally-built binaries when running Renode
- litevxriscv-tflite.robot
  - Remove redundant ADXL345 model
README: Use locally-built binaries when running Renode

- LiteX_i2C_Zephyr.cs
  Reduce verbosity of the demo
  3 months ago
- angle.data
  Add Renode section
  9 months ago
- circle.data
  Add Renode section
  9 months ago
- litex-vexriscv-tltite.repl
  Add Renode section
  9 months ago
- litex-vexriscv-tltite.resc
  README: Use locally-built binaries when running Renode
  8 days ago
- litex-vexriscv-tltite.robot
  Remove redundant ADXL345 model
  3 months ago
**README: Use locally-built binaries when running Renode**

- **LiteX_i2C_Zephyr.cs**
  - Reduce verbosity of the demo
  - 3 months ago

- **angle.data**
  - Add Renode section
  - 9 months ago

- **circle.data**
  - Add Renode section
  - 9 months ago

- **lite-x-vexriscv-tflite.repl**
  - Add Renode section
  - 9 months ago

- **lite-x-vexriscv-tflite.resc**
  - README: Use locally-built binaries when running Renode
  - 8 days ago

- **lite-x-vexriscv-tflite_robot**
  - Remove redundant ADXL345 model
  - 3 months ago
LINUX-LITEX-VERTRICPY-PIPELINE
A Linux build running on the Litex soft SoC

RIOT-CC2538DK-PIPELINE

RPL UDP IN CONTIKI-NG ON CC2538DK
Two CC2538 nodes running Contiki-NG connected with RPL-UDP

ZEPHYR TSN/GPTP ON SAM E70
Interesting?

Get Renode™ for:

- Debian
  Linux, (.deb)
  Download

- Fedora
  Linux, (.rpm)
  Download

- Arch
  Linux, (.pkg.tar.xz)
  Download

- macOS
  Download

- Windows
  Download
Renode 1.10 - latest news

- Renode 1.10 recently released
- Portable Renode installation option - no dependencies required
- New CPUs, SoCs, platform and boards, e.g:
  - NXP K64F
  - Nordic NRF52-DK (32-bit Arm Cortex-M4 CPU)
  - Microwatt
  - 4-core VexRiscv
  - PolarFire peripherals
  - Enhanced QuickLogic EOS S3 support
Commercial services by Antmicro

- Professional support, implementing new platforms
- Setting up CI and improving development workflows in your organization
- Building customized tools, user interfaces and integrations
- Embedded systems development services powered by the Renode methodology
THANK YOU FOR YOUR ATTENTION!