Embedded System Development with RISC-V Processors

A Brief Introduction and Market Overview

By Allan He (Embedded System Association, China)

This article introduces the origin and development of the RISC-V ISA. It briefly describes the technical characteristics and selection guidelines of various RISC-V CPU cores and SoC design platforms. We will also have a look into the features of the RISC-V instruction extensions and the open-source and commercial software development tools. The article finally looks forward to the trend of RISC-V in education and industry development.

In the past two decades, ARM has achieved milestones in mobility and embedded system areas. Today, we are observing ARM establishing its market leadership in IoT, while witnessing the demise of commercial processor architectures like MIPS. After ARM entered the PC and server markets that were previously dominated by x86, it has placed a fair amount of pressure on Intel. Right at this moment, the rise of RISC-V has attracted the attention of the industry. IT giants have been searching for an alternative to ARM; the openness of RISC-V ISA means they have found a viable option. In the academia, RISC-V is now replacing MIPS and x86 as the architecture of choice in textbooks at a break-neck pace. With governments and enterprises adopting RISC-V as the standard, we are seeing a blowout of new CPU and SoC designs, a growing ecosystem, and an ever more engaging development community.

What Is RISC-V?

An Instruction Set Architecture (ISA) refers to the instructions and their bytes encoding supported by a CPU; it is the bridge between computer software and hardware. CPU families, like x86, PowerPC, and ARM, have their own individual ISAs; RISC-V is currently the only open-source ISA. RISC-V is an open-source ISA, which means it is not an actual CPU chip, but a collection of specifications and standards on processor instructions; in fact, it is not even a complete instruction set. RISC-V originated from the University of California, Berkeley. In the summer of 2010, Professor Krste Asanovic launched a three-month research project with his students Andrew Waterman and Yunsup Lee. The goal was to develop an open ISA in order to work around the technical complexity and IP complications of x86 and ARM.

The RISC-V Foundation was established as a non-profit organization in 2015. The foundation's board consists of representatives from Bluespec, Google, Microsemi, NVIDIA, NXP, UC Berkeley, and Western Digital, and chaired by Professor Krste Asanovic. The foundation establishes standards and contributes to the building of the ecosystem of RISC-V cores; the standards are public and open for download. There are more than 300 paid members of the foundation, including Qualcomm, NXP, Alibaba and Huawei. Members are authorized to use RISC-V trademarks. RISC-V is licensed under the open-source BSD license, any company, institution, and individual can design a CPU with RISC-V architecture manuals. Last year, the RISC-V Foundation was renamed to RISC-V International Association, and moved its headquarters from the USA to Switzerland; the association completed its registration in Switzerland in March 2020 [1].

After a decade of development, RISC-V has logged significant achievements in CPU IP core, platform, SoC, and application. To name a few example applications of RISC-V cores: Western Digital's SSD and HDD controllers that sport SweRV cores, Shenzhen Bluetrum's TWS Bluetooth headset, China Canaan Creative's K210 AIoT SoC chips, and Bluetooth/generic 32bit/high-speed MCUs from Nanjing Qinheng Microelectronics.

RISC-V's Instruction Set

The instruction set of RISC-V uses a modular design to organize instructions, with one English letter representing a module. The most basic and required instruction set in RISC-V is the integer set, signified by the letter "I". Even with just one set, one can implement a complete software compiler. Other instruction sets are electable modules, a few notable examples are M/A/F/D/C. If a RISC-V cure is RV32IMAC, it means it implements I/M/A/C instruction sets. The 32I and 64I sets in RISC-V are already frozen, so are MAFDQC extensions; 32E, 128I, JBJTPV, and ZAM atomic access extensions are still under develop-

ment. Instruction set extension is a unique technical feature of RISC-V; by working with member institutions and the industry, RISC-V is able to maintain stable growth [2].

RISC-V Processor Core, SoC Platform, and ChipRISC-V Processor Core

Before we dive into embedded system development with RISC-V processors, let's first converge on a few concepts: RISC-V processor core (referred to as the core), SoC platform, and SoC chip. Since the inception of RISC-V architecture, there have been dozens of RISC-V cores and SoC chips; some of them are open-source, some are internal projects, while others are enterprise-grade cores and platforms. Western Digital's SweRV (RV32IMC) is a 32-bit sequential execution instruction architecture; it has a two-way superscalar design, a nine-stage pipeline, and is manufactured with a 28-nm process. With a top runtime frequency of 1.8 GHz, SweRV processor logs a performance benchmark of 4.9 CoreMark/MHz, which is slightly higher than ARM Cortex A15. SweRV is an open-source project and has already been deployed on SSD/HDD controllers produced by Western Digital.

There are quite a few examples of open-source RISC-V cores. Rocket Core from UC Berkeley is a classic RV64 design, while BOOM Core from the same university aims for higher performance. ETH Zurich's Zero-riscy is a typical RV32 design, while their R15CY Core is intended for ultra-low-power/small silicon and can be configured as RV32E. RISC-V Core PicoRV32, developed by Clifford Wolf, focuses on smaller footprint and optimization of CPU frequency.

Although open-source cores are suitable for research and teaching, there are more works to do if they want to design a commercial SoC. Yunsup Lee, a co-developer of RISC-V, started SiFive. In 2017, the company announced its first RISC-V core and SoC family, together with software and development boards. These chips include 28-nm process 64-bit multi-core CPU U500 that supports Linux, and 180-nm process low-cost IoT processor core E300. Many more manufacturers are in the business of developing RISC-V processor cores, including Codasip, Syntacore, T-Head, Andes, as well as startup companies, such as Nuclei System Technology.

RISC-V SoC Platform

ETH Zurich's PULPino, open-source project LowRISC, and UC Berkeley's Rocket Chip open-source SoC generator (based on Chisel) are some of the most well-known RISC-V processor SoC platforms. In China, Zhenbo Hu, founder of Nuclei, is the initiator of the now popular Hummingbird E200 open-source softcore SoC platform [3].

RISC-V CPU/MCU Chip

In recent years, RISC-V processor SoC chips have been leaping forward. One example of well-known SoC chips is the GD32VF103 MCU chip made by GigaDevice; it is based on Nuclei's Bumblebee core (RV32IMAC). The GD32VF103 series has a clock speed of 108 MHz, 16..128 KB of on-chip flash , 6..32 KB of SRAM, 4x 16-bit general-purpose timers 2x 16-bit basic timers, and 2 multi-channel DMA controllers. GD32VF103 MCU offers a newly designed Enhanced Core Local Interrupt Controller (ECLIC); it supports as many as 68 peripheral interrupts and 16 levels of interrupt priority, making the MCU capable of high-performance real-time computing.

There are several GD32VF103 MCU development boards: GD32VF103V-EVAL full-featured evaluation board and GD32VF103-START entry-level education board. Additionally, there are Nuclei's RV-Start and Sipeed Longan Nano development board [4], as well as IAR's latest RISC-V GD32 EVAL evaluation kit [5], as **Figure 1** below shows.

About the Author

Allan He is an Embedded Systems Specialist. He served as deputy director of the editorial board of *Microcontrollers and Embedded Systems* Journal. He is the founder of an embedded software company. He frequently speaks in conferences and teaches about embedded operating systems and IoT at universities and corporations.

China Canaan Creative's K210 is an AIoT SoC sporting a RISC-V CPU; the processor consists of two RV64GC cores with MAFD instruction extensions. K210 includes a KPU general-purpose convolution neural network processor to detect human faces and objects in real-time. The Fast Fourier Transform (FFT) Accelerator on K210 is implemented on hardware [6].

NXP RV32M1 integrates 4 cores: RISC-V RI5CY, RISC-V Zero-riscy, ARM Cortex M4, and ARM Cortex M0+ (**Figure 2**). From a professional's perspective, RV32M1 is more like an engineering prototype for developers to evaluate. NXP established the Open ISA community (https://open-isa.org/) and has been contributing to toolchain maintenance and construction of the software ecosystem.

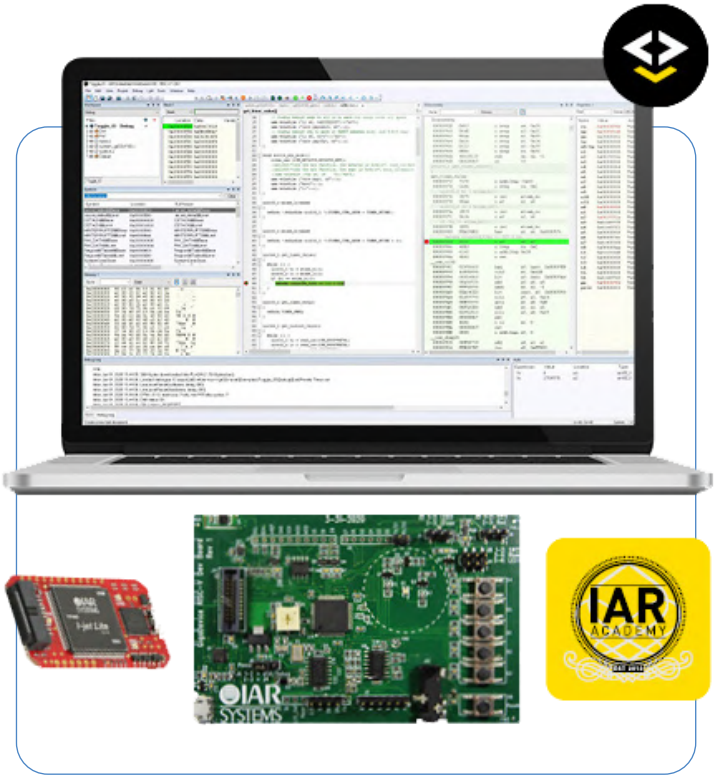


Figure 1: IAR RISC-V GD32V Evaluation Kit (Photo: iar.com).

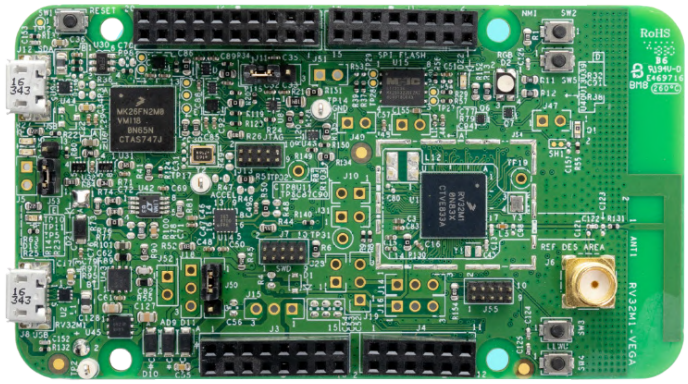


Figure 2: NXP RISC-V Vega Development Board (Photo: open-isa.org).

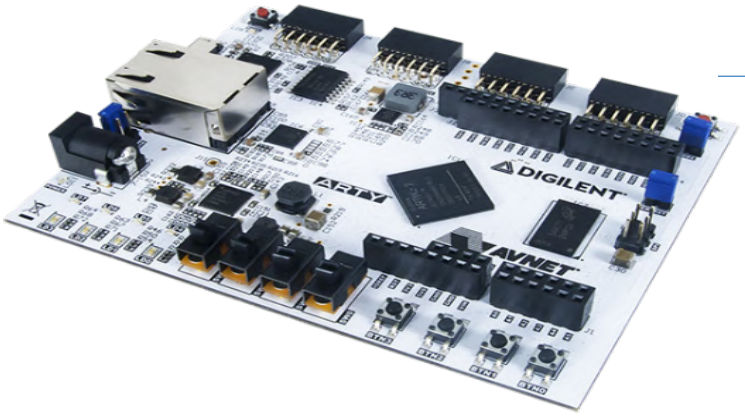


Figure 3: Arty FPGA Development Board Capable of Running RISC-V Softcore (Photo: digilentinc.com).

NXP played an important role in the early market cultivation of RISC-V by sending out development boards [7] and holding competitions/hackathons in China.

Microchip PolarFire SoC is a low-cost, multi-core SoC FPGA; it consists of four 64-bit RV64GC RISC-V application cores, and one RV64IMAC core for real-time and monitoring tasks. PolarFire SoC is capable to run Linux and is particularly suitable as a development platform for industrial control and IoT applications.

Choosing Between RISC-V Core, SoC Platform, and Chip

To the increasing number of practitioners that have started to participate in RISC-V research, development, and education, I have the below recommendations:

- Chip designers can opt to use RISC-V cores and SoC platforms to construct their own chips. For example, one can use PULPino platform to develop SoC, and adopt RI5CY and Zero-riscy cores; a number of companies and universities in China are using them in business/research projects.
- Embedded system and IoT developers will benefit from choosing RISC-V SoC chips. For instance, GD32VF103 series MCU has numerous development boards available and supports multiple toolchains; AIoT applications can go with K210, which is accompanied by a mature SDK and support of both FreeRTOS and bare metal. Recently, Linux 5.8 added support of K210 RISC-V to the mainline release. K210 has been successfully adopted in machine vision/hearing applications like face recognition and intelligent energy meter-reading.
- Universities and research institutions can elect open-source RISC-V core on FPGA to teach and research in areas like computer architecture, OS, compiler, and embedded system. As an example, Arty FPGA development board that implements SiFive open-source Freedom E310 MCU (as shown in **Figure 3**) is a mature platform with good software toolchain support [8].

RISC-V's Advantages in Embedded Systems

Apparently, embedded system, IoT, and AIoT are RISC-V's most active application markets. RISC-V brings quite a few advantages to the embedded system ; they can be summarized in three major points.

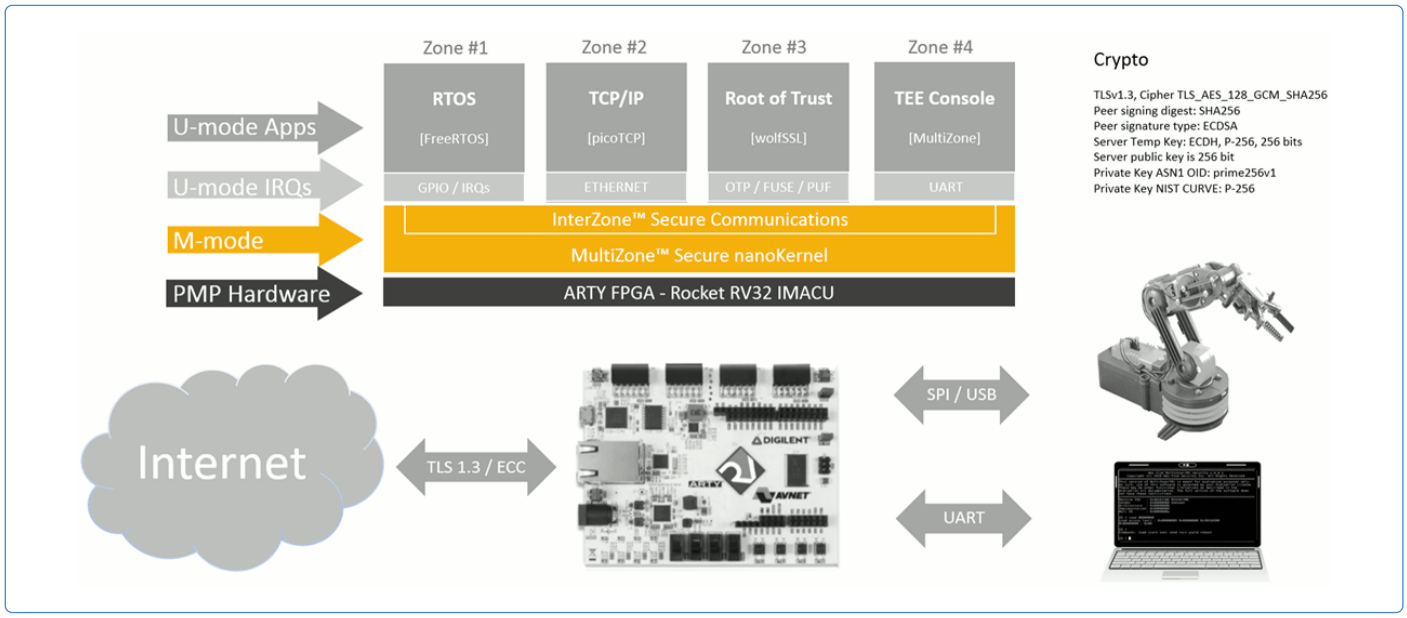


Figure 4: Reference Application of MultiZone Security IoT Stack (Photo: riscv.org).

- Open-source and free. Open-source is the new way to be economical and successful business-wise; it is also the best way for students and engineers to learn. Open-source ISA means developers can create their own chip architecture design targeting specific application scenarios. A free price tag lowers the bar to enter the chip design arena, allowing grassroots developers to participate.
- Simple and flexible. There are 50 fundamental RISC-V instructions in 4 basic instruction sets. The modular design allows a designer to create RISC-V CPU with a simplified instruction set, which by extension leads to lower code density and smaller energy footprint. The flexibility of RISC-V also means it can support a wide range of processors, from 8051 to the ARM A series.
- Highly efficient and secure. RISC-V can be easily extended with its reserved encoding space and user instructions; instruction extensions can accelerate computation and provide IoT security enhancements. A general approach in IoT security is to identify the area of Trusted Execution Environment (TEE) out from untrusted environment; hardware TEE is part of RISC-V's ISA standard specification and can be implemented on any RISC-V chip; the specifications include Physical Memory Protection (PMP).

Hex-Five has a demonstration of RV32 Core with IoT hardware-enforced security domain separation [9] (see **Figure 4**). Source code is available on GitHub [10]. From the design, we can see the X300 Bitstream SoC with Rocket RV32IMAC core and Digilent Arty A7 development board with Ethernet support. Software-wise MultiZone defines 4 zones of TEE:

- Zone 1: FreeRTOS, runs CLI, PWM LED, and robotic arm tasks.
- Zone 2: TCP/IP with PicoTCP, channel encrypted with TLS.
- Zone 3: WolfSSL TLS 1.3, trusted root, encryption keys/passwords, protected files.
- Zone 4: UART, TEE console.

RISC-V Embedded Software Ecosystem

RISC-V's software ecosystem is largely open-source with some commercial software. Several GNU open-source toolchains support RISC-V, like C compiler riscv-gcc, binary tools/linker/assembler riscv-binutils, debugger riscv-gdb, and open-source debugging software OpenOCD, which runs on PC and controls JTAG hardware like J-link. OpenOCD is built in with GDB server and support of GDB commands.

GNU Open-Source Toolchains

Common open-source IDEs include SiFive Freedom Studio, AndesSight, and Nuclei Studio IDE; they are Eclipse-based and are optimized for the vendors' own CPU cores. A developer can download JDK, Eclipse IDE for C/C++ developers, GNU MCU Eclipse build tools, and riscv32-unknown-elf-gcc toolchain to build his own RISC-V development environment. QEUM processor emulator already supports the RV32/RV64 instruction set, which is good news to the growing RISC-V family; developers can choose software and OS to run on the emulator, including FreeRTOS, Zephyr, and Linux [11].

Commercial embedded software companies have already started to support RISC-V RV32 and some processor core/chips (e.g. GD32VF103 and SiFive E310), Swedish company IAR's Embedded Workbench for RISC-V and German company SEGGER's Embedded Studio RISC-V are among the most popular products available in the market.

IAR Embedded Workbench

IAR's Embedded Workbench is a development tool that enjoys fame similar to Keil MDK in embedded system and MCU markets. It offers excellent optimization on code size and efficiency; developers can entrust it for compiling, analyzing, and debugging application code. The latest version of IAR Embedded Workbench for RISC-V is 1.30 with the features as:

- Further optimization of the compiler and the runtime library,
- Support DSP instructions in the P extension and the draft of Packed SIMD specification,
- Support debug based on the Nexus IEEE-ISTO 5001™ protocol and SiFive Insight solution and
- Added support for more than a dozen MCUs (e.g. GigaDevice GD32V).

SEGGER Embedded Studio

SEGGER is known for its hardware JLINK debugger in the embedded field. SEGGER's Embedded Studio (**Figure 5**) started to support RISC-V in 2017. In addition to JLINK, Embedded Studio is compatible with other debugging interfaces like OpenOCD and

SIGLENT®

The Best Value in Electronic Test & Measurement

Shorten Time to Market – thanks to innovative Measurement Technology

- Oscilloscopes: 50 MHz -> 1 GHz
- Signal Generators: 10 MHz -> 500 MHz
- DC-Power Supplies: -> 200 W
- Elect. DC-Loads: 200 -> 300 W
- Multimeter: 4 ½ -> 6 ½ digits

Siglent RF-Instruments: Helper in developing error-free and reliable Communication

Spectrum- & Vector Network Analyzer
- EMI-Pre-Compliance
- Modulation Analysis
- Distance-to-Fault Analysis

RF-Signal Sources
- with/without complex IQ-Modulation

Siglent Technologies Germany GmbH
www.siglenteu.com
Info-eu@siglent.com

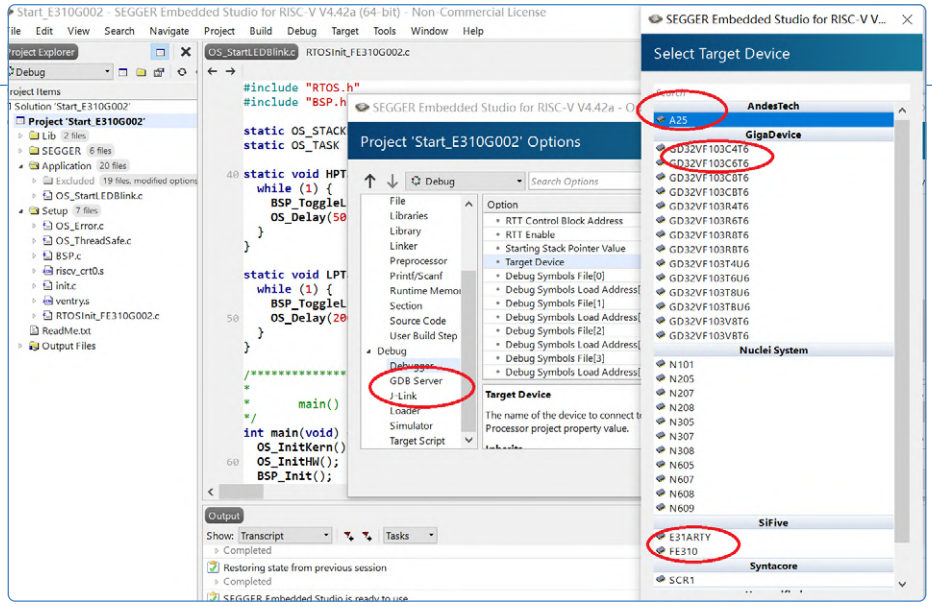


Figure 5: SEGGER Embedded Studio.

GD-Link. Embedded Studio has comprehensive support for the single-core RV32 instruction set, including RV32I, RV32IMA, RV32IMAC, RV32IMAF, RV32IMAFC, RV32G, and RV32GC; support for RV64 and multi-core are reportedly being tested. It also supports Nuclei Technology's RISC-V processor core, Real-Time Transfer (RTT) between host and debugger, as well as SEGGER SystemView software analytic tool.

Embedded Studio is an integrated development tool. According to SEGGER, it utilizes Clang/LLVM and GCC C/C++ compilers and also supports external toolchains. For instance, a Nuclei Embedded Studio project by default uses the GCC toolchain optimized by Nuclei. More information about Embedded Studio you can read Rolf Segger's blog *The SEGGER Compiler* [12].

Embedded Studio has a friendly licensing scheme. It is free to university students/lecturers and non-commercial usage; you can start using it right after agreeing to the license agreement.

Embedded Operating Systems

A RISC-V processor must accompany an embedded OS to work in embedded systems and IoT applications. There are already sample RISC-V ports in FreeRTOS, Zephyr OS, ThreadX and µC/OS. FreeRTOS version 10.3 already has ports on NXP Vega and SiFive Freedom HiFive1-revB boards; compiler-wise GCC and IAR are supported. SEGGER embOS has a demo with emWin on GD32VF103-EVAL board. SiFive released Amazon FreeRTOS port on Github; currently, SiFive Learn Inventor education board and Andes Corvette-F1 N25 platforms can take advantage of it. In China, RT-Thread has a demo on HiFive1-revB, Huawei LiteOS has demoed on GD32VF103-EVAL board, TencentOS Tiny has a demo PM2.5 monitoring application on GD32VF103 board.

As RISC-V ISA intends to enter the high-end computing market, it is critical for RISC-V to gain support by Linux and be admitted

LINKS AND LITERATURE

[1] **History of RISC-V**: https://riscv.org/about/history/

[2] **D. Patterson and A. Waterman, The RISC-V Reader: An Open Architecture Atlas, Strawberry Canyon LLC, 2018.**

[3] **Hu, Zhenbo. RISC-V Architecture and Introduction to Embeddded System Development. China Posts & Telecom Press, 2019. p34-39.**

[4] **RISC-V MCU Chinese Community**: https://www.riscv-mcu.com/

[5] **Evaluation kit for RISC-V**: https://www.iar.com/iar-embedded-workbench/tools-for-risc-v/evaluation-kit-for-risc-v/

[6] **Kendryte K210**: https://canaan-creative.com/developer

[7] **V32M1-VEGA Development Board User Guide, fetched from** : https://github.com/open-isa-org/open-isa.org/releases/download/1.0.0/Documentation.zip

[8] **Arty FPGA Evaluation Kit Hands-On + Verification of RISC-V Instruction on SiFve Chip (blog in Chinese)**: www.cnblogs.com/zjutlitao/p/9745365.html

[9] **The first TEE for RISC-V**: https://hex-five.com/multizone-security-sdk/

[10] **Sourcecode at GitHub**: https://github.com/hex-five/multizone-iot-sdk

[11] **RISC-V Foundation, „RISC-V - Getting Started Guide" May 2019, p5/14**

[12] **The Segger Compiler**: https://blog.segger.com/the-segger-compiler/

[13] **Linux on RISC-V**: https://kernelnewbies.org/Linux_5.8#RISCV

[14] **RISC-V Computer Architecture Course (includes teaching materials and hands-on exercises for students)**: https://bit.ly/3tXvr1h

[15] **rCore OS (Site in Chinese)**: https://rcore.gitbook.io

[16] **Embedded System and IoT Development Online Seminar (3)**: http://hexiaoqing.net/courses/

[17] **S. Greengard, "Will RISC-V Revolutionize Computing?" Communication of the ACM, Vol. 63 No. 5, May 2020.**

into the Linux tree. There have been some developments thanks to engineers of Andes, Western Digital, and SiFive. For instance, there are demos where Fedora and Debian run on SiFive HiFive Unleashed. RISC-V is a new processor architecture that open-source Linux developers clearly take interest in. However, as there weren't many low-cost RISC-V development boards, developers have found it difficult to work on RISC-V. Recently, Linux 5.8 added official support of K210 [13], which means K210 is no longer only supported by no-MMU Linux. The importance of this news is that developers can easily obtain a K210 (RV64GC) board to work on it.

Education and Development

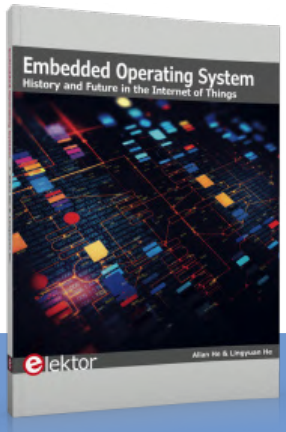
In September 2020, Imagination Technologies announced its undergraduate RISC-V-based Computer Architecture course titled "RVfpga: Understanding Computer Architecture" [14]. The course was developed by Imagination and Harvey Mudd College Associate Professor Sarah Harris, who is a co-author of the well-known *Digital Design & Computer Architecture* textbook. According to Sarah Harris: "RISC-V improves on previous processor generations in every conceivable way, from power consumption to performance and even increased security. As another huge step forward in computer architecture it is important for students to understand RISC-V at a fundamental level."

Associate Professor Yu Chen of Tsinghua University leads a team that has been working on implementing an OS in RUST on RISC-V. The objectives of their rCore OS, a port of uCore in RUST, are to improve OS development experience and quality using a modern programming language, as well as to explore future OS design and implementation [15]. I have participated in the 2nd and 3rd embedded system and IoT development online courses co-developed by BMR, IAR, SiFive, GigaDevice, and Nuclei. These courses aim to help developers, university lecturers, and students to learn about the latest in RISC-V, embedded system development, and development tools. Lab code projects and related materials of the courses have been made available for the community [16].

RISC-V faces many challenges in embedded system applications. First, embedded system development requires standardized general purpose development platforms; there are very few of them in the market. Second, commercial-grade open-source software is important; support of Linux has been newly added, but Android may not run on RISC-V in the immediate future. Third, RISC-V has an active ecosystem, but there hasn't been enough accumulation of achievements in the community.

Without a doubt, RISC-V is extremely suitable for education research and curriculum in electronics and information technology. A brand new open-source hardware model signifies the start of an environment that promotes innovation and cooperation. I will close with words from Michael Taylor, an Associate Professor in the School of Computer Science and Engineering at the University of Washington in Seattle: "There are no serious technical or practical issues with RISC-V. It will eventually supplant x86 and ARM as the primary instruction set for microprocessors. It will fundamentally change the computing world" [17].

200659-01



RELATED PRODUCT

› **A. He and L. He, Embedded Operating System: History and Future in the Internet of Things, Elektor 2020.**

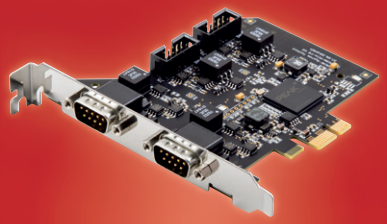
www.elektor.com/embedded-operating-system

You CAN get it...

Hardware & software for CAN bus applications...

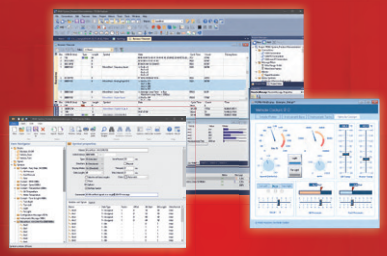
PCAN-Router Pro FD

Freely programmable 6-channel router for CAN and CAN FD with I/O and data logger. Delivery incl. development package with examples.



PCAN-PCI Express FD

CAN FD interface for PCI Express slots with data transfer rates up to 12 Mbit/s. Delivery incl. monitor software, APIs, and drivers for Windows and Linux.



PCAN-Explorer 6

Professional Windows software for observation, control, and simulation of CAN FD and CAN 2.0 busses.

www.peak-system.com

PEAK System

Otto-Roehm-Str. 69
64293 Darmstadt
Germany
Phone:+49 6151 8173-20
Fax: +49 6151 8173-29
info@peak-system.com