

# RISC-V

## RISC-V处理器生态建设的实践

Practice of Ecosystems of RISC-V Processor

主讲人：何小庆/Allan He

1

什么是RISC-V ?

2

RISC-V 处理器家族

3

RISC-V 嵌入式生态建设

4

如何学习RISC-V ?

# 什么是RISC-V ?

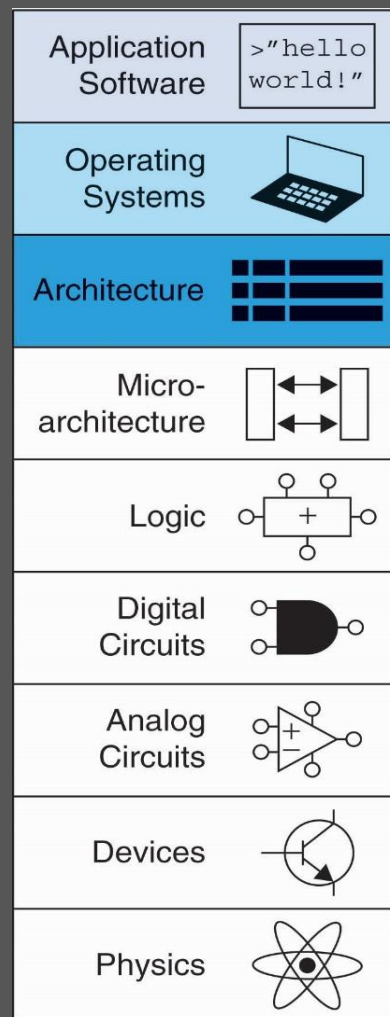
## RISC-V

- RISC-V 是一种开源的指令集架构，它不是一款CPU芯片。
- RISC-V 采用开源BSD 授权 ,任何企业、高校和个人都可以遵循**RISC-V架构指南**设计自己的CPU。
- RISC-V 起源于加州大学伯克利分校，RISC-V 最初是Krste Asanovic 教授和他的学生 Andrew Waterman 以及 Yunsup Lee 创建项目。
- 2015年，这个指令集在学术界开始出名了，为了更好的推动这个指令集在技术和商业上的发展，3位大佬做了下面两件事情。技术方向:成立**RISC-V基金会** --维护指令集架构的完整性和防止碎片化。商业方向：2015年成立**SiFive公司**，推动RISC-V的商业化。



# 指令、架构以及处理器家族

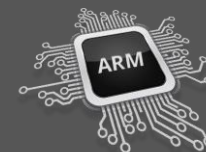
- 一个CPU支持的指令和指令级编码就是这个CPU的指令集 (ISA)
  - **指令**: 计算机语言的命令
  - **汇编语言**: 人类可读的指令格式
  - **机器语言**: 计算机可读格式 (1 和 0)
- **指令集**在软件和硬件设计者之间提供了一个接口。不同的CPU家族: X86、PowerPC和ARM, 都有不同的ISA, RISC-V 是目前其中知名开源ISA。的
- **架构**: 是从程序员的视角看计算机, 由指令和操作数来定义。
- **微架构**: 如何在硬件 (数字逻辑) 中实现架构。
- 全球已有两种成功计算架构, 并建立了自己的CPU家族生态
  - Intel的X86架构和ARM公司的ARM架构。
  - RISC-V 将与X86, ARM 一起支撑未来计算架构的舞台。
  - 其他的架构 PPC, MIPS, ARC, SPARC,..... 市场分额很少。



## RISC-V

SKY

MIPS  
by imagination



ARC  
Synopsys

ANDES  
TECHNOLOGY

SPARC



PowerPC™



# RISC-V International



- RISC-V International (简称: 基金会) 是一家总部位于瑞士的全球性非营利组织。RISC-V 成立于 2015 年, 汇集了来自 70 多个国家的跨行业和技术学科的会员, 现在有超过 2000 成员。
- 基金会制订开放的 RISC-V 指令集架构, 研究并建立指令集的扩展、工具和生态, 为未来 50 年的计算创新铺平道路。基金会还通过学术界、商业化和战略领导力将社区和行业联系起来。

The RISC-V Instruction Set Manual  
Volume I: Unprivileged ISA  
Document Version 20191213

Editors: Andrew Waterman<sup>1</sup>, Krste Asanović<sup>1,2</sup>

<sup>1</sup>SiFive Inc.,

<sup>2</sup>CS Division, EECS Department, University of California, Berkeley

andrew@sifive.com, krste@berkeley.edu

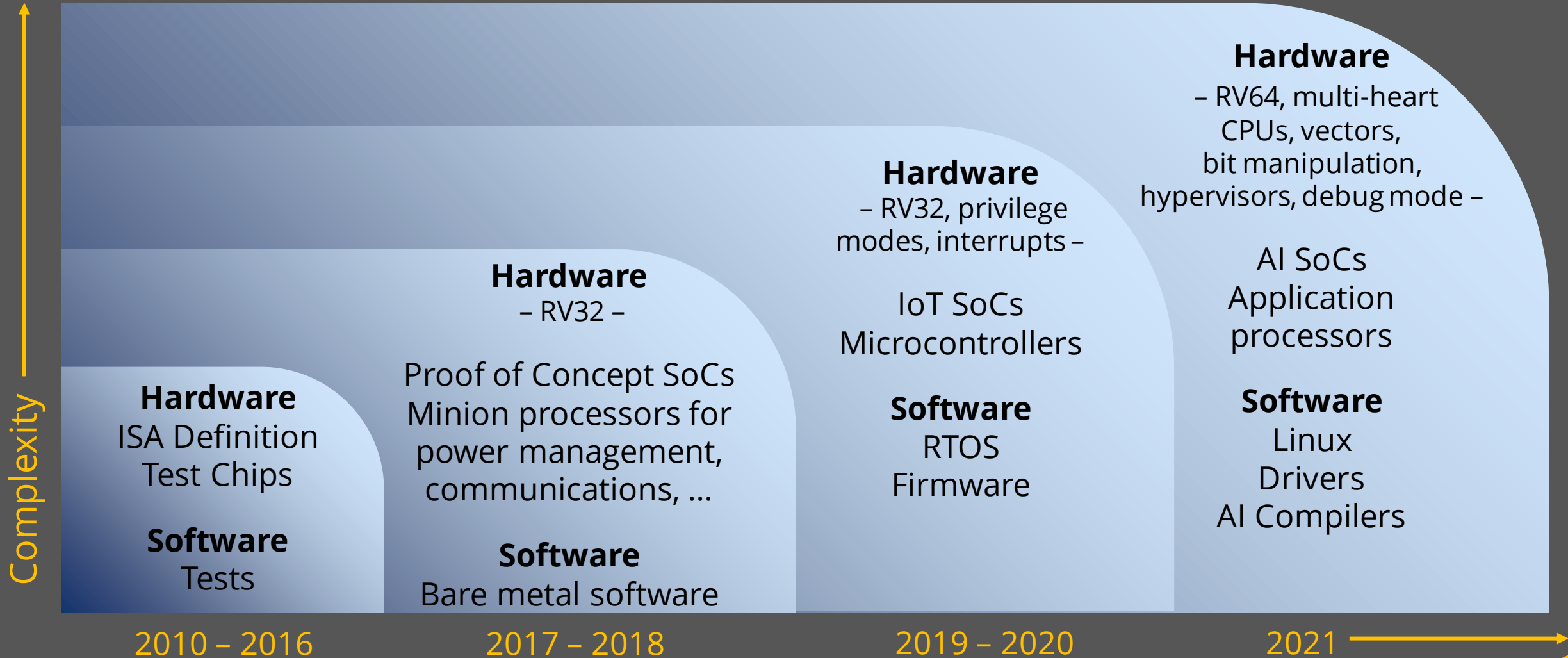
December 13, 2019



<https://riscv.org/specifications/isa-spec-pdf/>

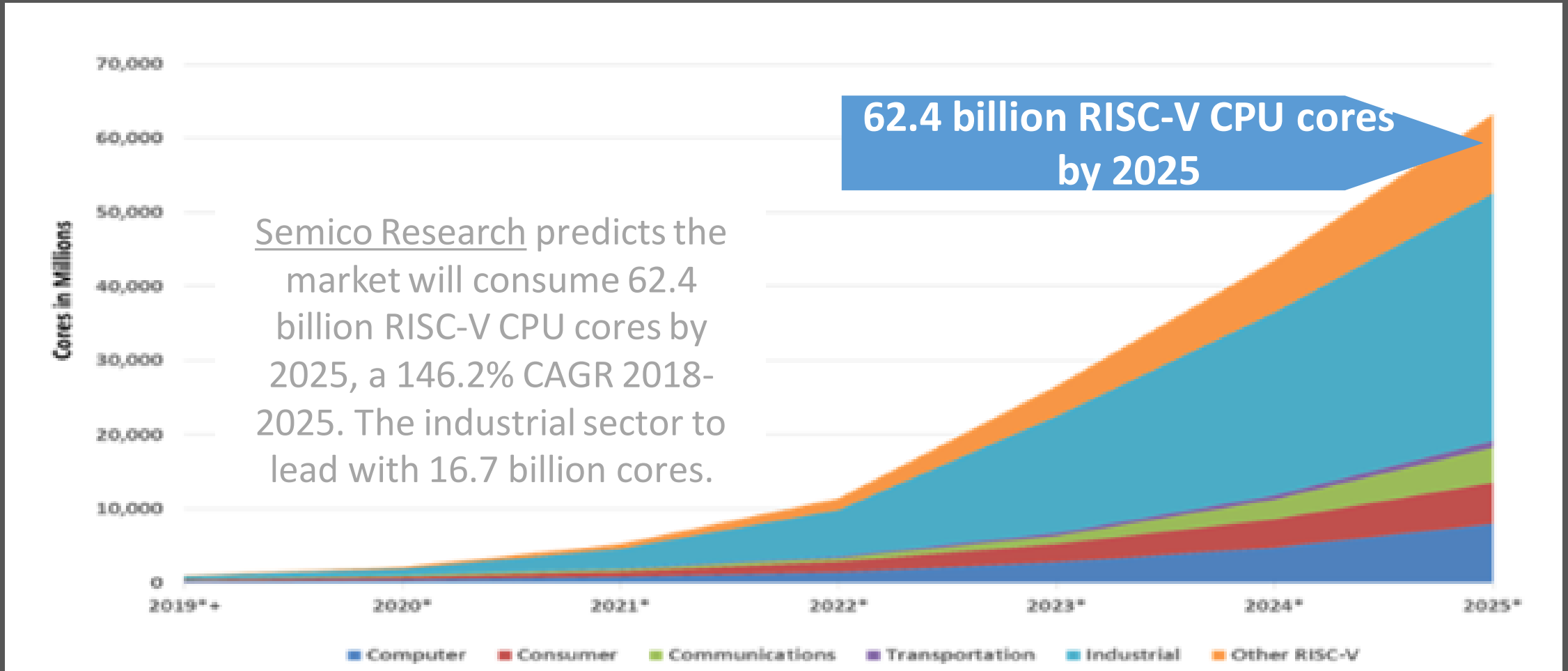
# RISC-V 的创新之路

# RISC-V



# 产业推动RISC-V 快速成长

## RISC-V



德勤预测，RISC-V处理内核的市场将在2022年比2021年翻一番，并且在2023年将再次翻番，因为RISC-V处理内核的可服务目标市场继续扩大

Source: Semico Research Corp

# 2021年 RISC-V 最新成就

## RISC-V

- 欧洲处理器计划 (European Processor Initiative) , 首个CPU原型EPAC1.0来了, 使用的是RISC-V架构, 22nm工艺, 有4个核心, 集成了众多专用的加速器, 测试芯片主频是1GHz。
- 阿里巴巴平头哥 Xuantie 910 处理器推出, 基于 RV64GCV 内核, 软件上支持Android AOSP 。最近平头哥宣布包括 Xuantie 系列处理器和软件开源。
- 香山处理器基于Chisel语言开发, 支持多核, 采用乱序执行、11级流水、6发射。香山处理器是计算所牵头, 多家企业联合开发的开源处理器核, 源代码和所有设计文档都开源。
- Ventana提供数据中心级高性能RISC-V CPU, 这种CPU具有以多核小芯片形式, 提供可扩展指令集功能, 该还提供定制化系统级(SoC)小芯片。
- Micro Magic 发布了一款速度很快的 64 位 RISC-V 内核, 可在 1.1V 电压下实现 5GHz 和 13,000 CoreMarks。
- NSITEXE 基于 RISC-V 的带有矢量扩展的并行处理器 IPDR1000C 已授权给瑞萨电子用于 RH850/U2B , 这是一款车用的MCU。



# RISC-V 社区

# RISC-V



## 目录

1	什么是RISC-V ?
2	<b>RISC-V 处理器家族</b>
3	RISC-V 嵌入式生态建设
4	如何学习RISC-V ?

# RISC-V处理器大家族

## RISC-V

- RISC -V 诞生以来，已有100多个版本的 RISC -V 处理器核，10多个SoC芯片以及30多个SoC平台。
  - **有开源免费内核、平台和SoC，有商业公司开发的处理器IP，平台和SoC 芯片。**
- RISC -V 具有短小精悍和模块化的ISA架构，指令简洁。基本的RISC-V (RV32I/RV64I) 指令数仅有47/59条，其他的模块化扩展指令多是几-几十条指令，RISC-V ISA 支持用户指令集扩展。
  - 'M' : Math extension. Multiply and divide
  - 'F' , 'D' : Floating point extensions, single and double precision
  - 'A' : Atomic operations
  - 'B' : Bit manipulation
  - 'T' : Transactional memory
  - 'P' : Packed SIMD (Single-Instruction Multiple Data)
  - 'V' : Vector operators
- RISC-V架构定义了特权模式 (Privileged) 和非特权模式 (unprivileged) 。
  - **特权模式包括机器模式 (M)，管理员模式 (S) 和超级管理员模式 (H) 。**
  - **非特权模式指用户模式 (U)**

**RISC-V ISA + ISA 扩展 + 处理器模式 + 微架构 = RISC-V 大家族**

# 开源的RISC-V架构处理器核

- Rocket Core
  - USB BAR 开发，经典的RV64 设计。
- Zero-riscy
  - 苏黎联邦世理工学院 (ETH Zurich) 开发，经典的RV32 设计。
- R15CY
  - ETH Zurich开发，可配置成RV32E，面向超低功耗、超小面积的场景。
- PicoRV32
  - Clifford Wolf开发，重点在于追求面积和频率的优化。
- BOOM
  - UBC BAR 开发，与Rocket Core不同的是，BOOM Core面向更高的性能。
- XuanTie C910
  - 阿里平头哥 (T-head ) 开发，支持 Andorid AOSP 的64位 高性能CPU。
- Hummingbird E200
  - 芯来科技胡振波开发的32位 MCU 处理器核，国内知名的蜂鸟E200。

## RISC-V



SweRV Core™

**ETH** zürich



# 商业RISC-V 处理器核

# RISC-V





# 商业RISC-V 嵌入式处理器 (SoC)

## RISC-V

- **GD32VF103**

- 兆易创新开发基于芯来Bumblebee 内核 GD32VF103 MCU芯片。

- **CH32V103**

- 南京沁恒CH32V103/CH573/CH569 /CH32V307 RISC-V MCU。

- **Kendryte K210**

- K210 包含 RISC-V 64 位双核 CPU (Rocket core) , 每个核心内置独立 FPU , 是一个典型 AIOT SoC 。

- **NXP RV32M1**

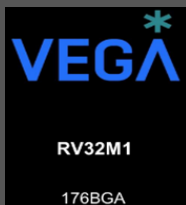
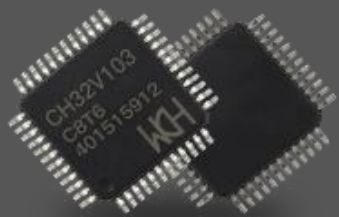
- 集成了4个cores: RISC-V RI5CY core, RISC-V ZERO-RISCY core, ARM CortexM4 core 和 ARM Cortex-M0+ core。

- **Microchip PolarFire SoC**

- 低成本, 多核RISC-V SoC FPGA, 4个 64-bit RV64GC RISC-V cores , 可运行Linux, 一个单核 RV64IMAC 可运行实时任务。

- **全志D1**

- 集成了阿里平头哥64位C906核心, 支持RV64GCV, 1GHz主频, 可支持Linux、RTOS等系统。



中科蓝讯蓝牙芯片AB32VG1, 乐鑫 IOT 芯片 ESP32-C3 , 华为海思Hi3861  
泰凌微蓝牙芯片LSR9xxx , 博流智能 IOT 芯片 BL60x/BL70x, 先缉半导体HPM600 MCU

# 如何选择RISC-V 家族成员?

## RISC-V

### 芯片设计者可选择RISC-V Core 和SoC Platform 构建自己的芯片

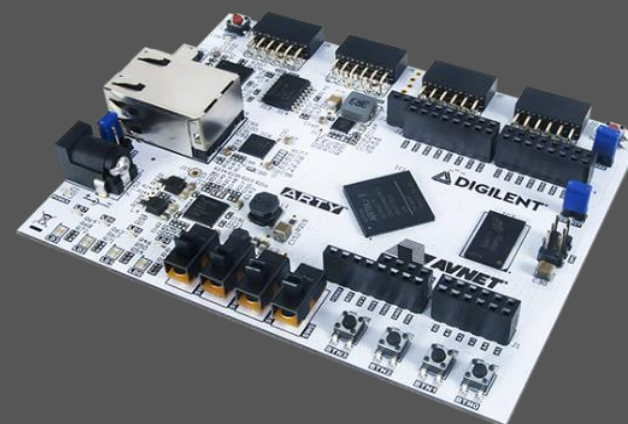
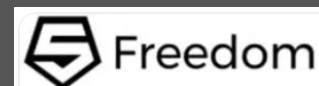
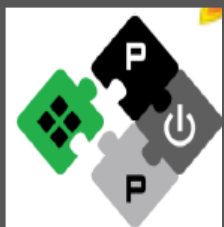
- 比如, 使用 PULPino 平台开发 SoC 芯片, 内核使用 RI5CY, (CV32E40P) Zero-risky(Lbex) , 国内企业和高校许多研究项目在用。

### 嵌入式和物联网开发者可以使用 RISC-V SoC 芯片

- 比如, 选择GD32VF103 系列 芯片做嵌入式项目开发, 功耗优于同类ARM MCU。

### 高校和研究机构可以选择开源RISC-V Core 和SoC Platform 在FPGA 平台上进行计算机体系架构、OS 和编译技术教学和研究工作

- 比如, 可在Arty FPGA 开发板上实现一个 SiFive 开源Freedom E310 微控制器并有相应软件工具链支持。



1

什么是RISC-V？

2

RISC-V 处理器家族

3

**RISC-V 嵌入式生态建设**

4

如何学习RISC-V？

# RISC-V为嵌入式开发带来什么？

## RISC-V

### RISC-V最大的优势是开源

- 开源是新的经济方式，是成功的商业之道、是学习的最好途径。
- ISA开源意味着开发者可以针对特定应用场景，创造自己的芯片架构。
- 免费可以降低入门的门槛，让草根开发者进入芯片设计。

### RISC-V 第二个优势是简单和灵活

- 基础的指令集有47条，模块化的4个基本指令集让设计者开发出很简化的RISC-V CPU，功耗可以很小，代码密度低，覆盖8051-ARM A系列 处理器。

### RISC-V 第三个优势是高效和安全

- 通过预留编码空间和用户指令，支持扩展的指令集。
- 通过指令集扩展实现运算加速，通过RISC-V 架构提供安全保护机制。

# RISC-V 开源工具链

GNU 工具链支持 RISC-V (SiFive 维护) 包含：

- riscv gcc GCC 编译器
- riscv binutils：二进制工具 链接器 汇编器等
- riscv gdb GDB 调试工具等
- riscv glibc GNU C 标准库实现
- riscv newlib：开源 C 标准库 主要用于嵌入式系统上
- riscv openocd：基于 OpenOCD 的 RISC -V 调试器软件

llvm

- llvm for riscv 编译器 目前主要由 lowRISC 维护

riscv 仿真器和 bootloader (SiFive、Berkeley 和WD 维护) 包括：

- riscv qemu QEMU 的 RISC- V 分支 主线已经合并了RISC -V支持。
- U boot Bootloader 目前在主干线维护。

## RISC-V





# RISC-V 嵌入式软件生态

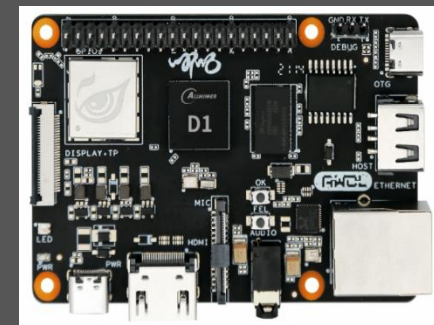
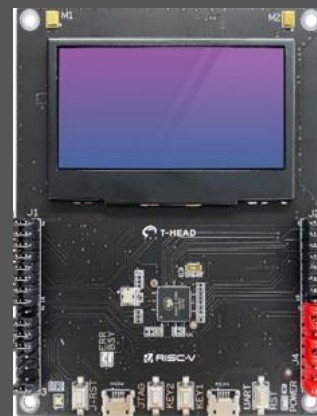
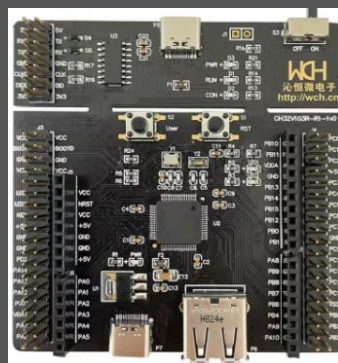
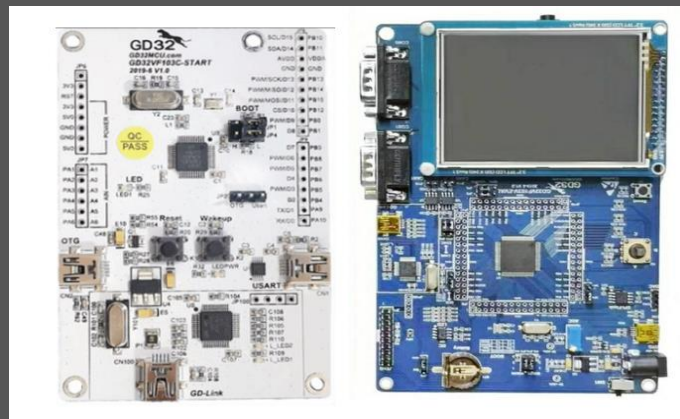
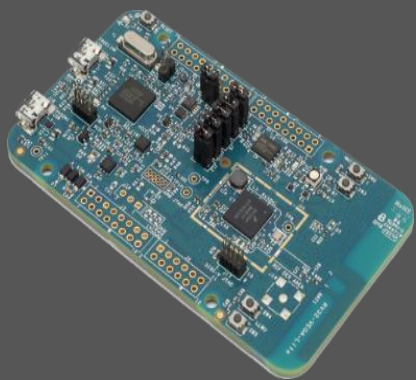
# RISC-V

- SiFive Freedom Studio
  - Eclipse CDT, GNU MCU Eclipse, pre-built GCC
  - OpenOCD – Built on Open Source technology
- Nuclei Studio , AndesSight 和 MounRiver
  - Eclipse based IDE for RISC-V
- RT-Thread Studio 和 卡姆派乐IDE
  - **China RISC-V IDE**
- SEGGER - JLINK Probe 和 Embedded Studio RISC-V IDE
- IAR - IAR Embedded Workbench for RISC-V
- Embedded RTOS
  - FreeRTOS、 Zephyr OS、 Thread X、  $\mu$ C/OS、 RIOT、 NuttX, RT-Thread、 Huawei LiteOS、 Sylix OS, OneOS 和TencentOS Tiny。



# RISC-V 生态-开发板

# RISC-V



# RT-Thread 支持RISC-V 的情况

RISC-V

## RV32

- HiFive1
- RV32M1\_VEGA
- GD32V103
- AB32VG1
- CH32V103
- CH32V307
- ...

## RV64

- K210
- Allwinner D1
- QEMU/RISCV64 VIRT
- ...

## 软核

- Nuclei hbird\_eval
- SMART-EVB for T-Head CPU E9xx Series
- SMART-EVB for E804/E804F/E804D
- PicoRV32
- ...

RT-Thread Studio all-in-one IDE also provides support for RISC-V chips.



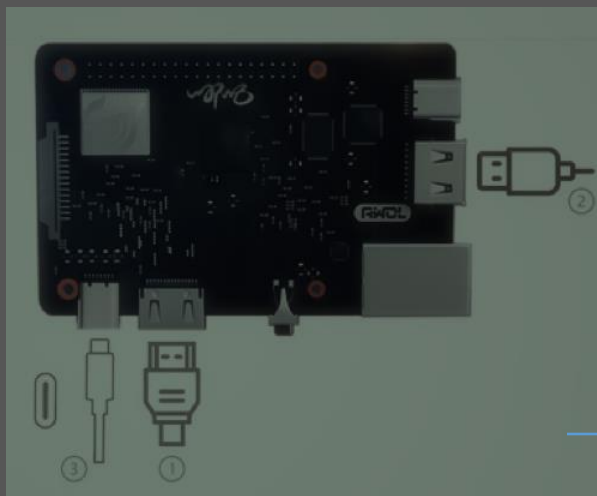
Strategic Members

# RISC-V 与 Linux

# RISC-V

## Fedora支持

- Fedora 是知名的Linux操作系统，是由全球社区爱好者构建的面向日常应用的快速、稳定、强大的操作系统。
- 目前在D1哪吒开发板上已经支持了Fedora系统。
- 软件包由合作伙伴 红帽-fuweai提供。



## Tina Linux

- D1 哪吒开发板默认自带Tina Linux系统。
- Tina Linux是全志科技基于Linux内核开发的针对智能硬件类产品的嵌入式软件系统。Tina Linux基于 openwrt-14.07 版本的软件开发包，包含了 Linux 系统开发用到的内核源码、驱动、工具、系统中间件与应用程序包。

## Debian 支持

- 本软件包由 PerfXLab 提供，主要包括：
- 基于D1哪吒开发板移植 Debian11 系统，并预装LXDE桌面环境。
- 适配多款传感器，包括：倾斜模块sw-520d、循迹避障模块TCRT5000、光照强度传感器BH1750FVI、大气压强传感器BMP280等。

## 哪吒开发板

主控：全志D1 C906 RISC-V 1GHz  
DDR3 1GB/2GB 256MB spi-nand  
USB外接U盘及SD卡拓展存储  
千兆以太网2.4G WiFi及蓝牙  
HDMI输出 MIPI-DSI+TP屏幕接口

## 目录

1	什么是RISC-V ?
2	RISC-V 大家族
3	RISC-V 嵌入式生态建设
4	如何学习RISC-V?



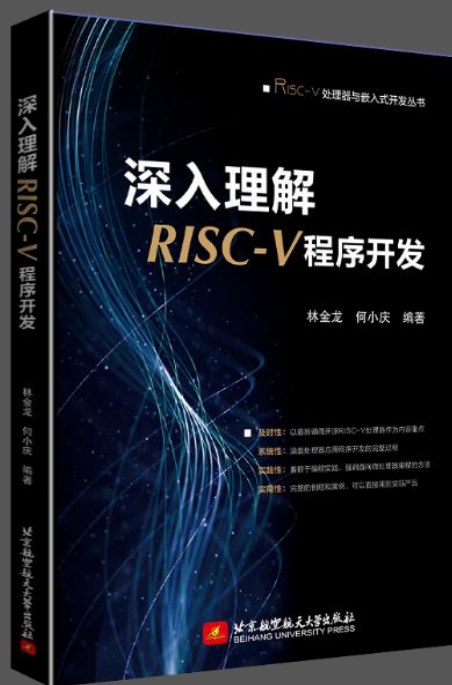


# 学习RISC-V 应用开发的路径

# RISC-V

## 图书目录

- 第1章 了解RISC-V
- 第2章 RISC-V 处理器芯片
- 第3章 RISC-V 软件开发工具
- 第4章 认识RISC-V内核
- 第5章 RISC-V软件开发
- 第6章 GD32VF103微控制器
- 第7章 GD32VF103中断系统及应用
- 第8章 深入RISC-V程序开发
- 第9章 嵌入式实时操作系统
- 第10章 物联网操作系统及其应用
- 第11章 基于RISC-V的电磁车设计
- 第12章 高性能RISC-V处理器



2021年7月出版

序号	名称	说明	章节
1	5_4_asm	汇编程序示例	5.4节
2	5_6_hello	打印“Hello risc-v”	5.6节
3	6_2_rcu_clock	系统时钟控制示例	6.2节
4	6_3_timer	定时器应用示例	6.3节
5	6_4_gpioled	GPIO应用示例	6.4节
6	6_5_uart_echo	串口示例	6.5节
7	6_6_i2c_eeprom	访问I2C接口存储器示例	6.6节
8	7_3_intkey	按键中断示例	7.3节
9	7_4_dma_adc	DMA中断示例	7.4节
10	7_5_exmc_tc	触屏中断示例	7.5节
11	chart9	2个FreeRTOS示例	第9章
12	chart10	rt-thread 和 TencentOS tiny示例	第10章
13	11_smart_car	智能电磁车示例	第11章

调研-内核-芯片-工具和软件-应用开发



# 如何在RISC-V 内核运行代码 (1)

# RISC-V

- 虽然RISC-V已经存在了一段时间，但可供使用的处理器芯片并没有很多。你想要在RISC-V的处理器上运行代码，最快方式是通过ESEO开发的**emulsiV 模拟器**。它使用名为“Virgule”的RISC-V内核实现，除了处理器，模拟器还提供文本I/O、位图输出和一些通用I/O (GPIO) 的支持。

The screenshot displays the emulsiV simulator interface. At the top, the URL is <https://guillaume-savaton-eseo.github.io/emulsiV/>. The interface includes a navigation bar with buttons for 'Reset', 'Run', 'Step', 'Fetch', 'Decode', 'ALU', 'Compare', and 'Mem/Reg'. The 'Fetch' button is currently selected.

**Memory:** A table showing memory addresses and instructions. The instruction at address 00000014 is highlighted in pink: `addi x1, x1, 1`.

Address	0	1	2	3	Instructions
00000000	93	00	00	02	<code>addi x1, x0, 32</code>
00000004	37	01	00	c0	<code>lui x2, 0xc0000000</code>
00000008	83	c1	00	00	<code>lbu x3, 0(x1)</code>
0000000c	63	88	01	00	<code>beq x3, x0, .+16</code>
00000010	23	00	31	00	<code>sb x3, 0(x2)</code>
00000014	93	80	10	00	<code>addi x1, x1, 1</code>
00000018	6f	f0	1f	ff	<code>jal x0, .+16</code>
0000001c	6f	00	00	00	<code>jal x0, .0</code>
00000020	48	65	6c	6c	-
00000024	6f	00	00	00	<code>jal x0, .0</code>
00000028	00	00	00	00	-

**Program counter:** pc: 00000014, mepc: 00000000, pc+4: 00000018.

**Bus:** addr: c0000000, data: 00000048, irq: false.

**Instruction reg.:** instr: 00310023, fn: sb, rs1: 2, rs2: 3, rd: -, imm: 00000000.

**ALU:** op: add, a: c0000000, b: 00000000, r: c0000000.

**General-purpose regs:** x9-x27 registers are shown, with x15-x27 containing 00000000.

**Comparator:** op: -, a: -, b: -, taken: false.

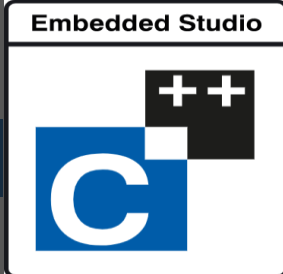
**Text I/O:** b0000000 (ctrl, data) contains 00 00; c0000000 (data) contains 48.

**General-purpose I/O:** d0000000 (dir) contains ff ff ff ff; d0000004 (ien) contains 00 00 00 00; d0000008 (rev) contains 00 00 00 00; d000000c (fev) contains 00 00 00 00; d0000010 (val) contains 00 00 00 00.

**Bitmap output (00000C00-00000FFF):** A black square representing the output.

**Footer:** Includes links for 'About emulsiV', 'Get the source code', and 'Report issues', along with the ESEO logo and the tagline 'INNOVATION MAKES SENSE'.





# 如何在RISC-V 内核运行代码 (2)

RISC-V

asm - SEGGER Embedded Studio for RISC-V V5.66 (64-bit) - Non-Commercial License (Stopped)

File Edit View Search Navigate Project Build Debug Target Tools Window Help

Disassembly

```
array + 0x8  
----- main.c ----- 24 -----  
do {  
i=1;  
080001C8 4785 li a5, 1  
080001CA C63E sw a5, 12(sp)  
080001CC BFF5 j 0x080001C8  
----- array.s ----- 2 -----  
.section .text, "ax" //  
080001CE 18000517 auipc a0, 0x18000  
080001D2 E3250513 addi a0, a0, -462  
080001D6 490C lw a1, 16(a0)  
080001D8 5118 lw a4, 32(a0)  
080001DA 95BA add a1, a1, a4  
080001DC D90C sw a1, 48(a0)  
080001DE 0001 nop  
080001E0 8082 jr ra  
----- compress.s ----- 3 -----  
.global ccode
```

main.c compress.s sum.s **arrays.s** macro.s flash\_placement\_riscv.xml riscv\_crt0.s

```
array0  
◆ array0  
.section .text, "ax" // 声明.text段及其属性  
.global array  
.type array, function  
array:  
la a0, array_data // 装载矩阵数组起始地址  
7 lw a1, 4*4(a0) // 读取A[1][0]  
lw a4, 8*4(a0) // 读取A[2][0]  
add a1, a1, a4 // 相加  
10 sw a1, 12*4(a0) // 写入A[3][0]  
nop // 空操作  
ret  
  
.section .data_run, "aw", %progbits // 声明可读写数据段  
.global array_data // 声明全局标签  
  
array_data: // 矩阵标签  
word 0x11, 0x12, 0x13, 0x14 // 分配内存, 并初始化数组  
word 0x21, 0x22, 0x23, 0x24 // 分配内存, 并初始化数组  
word 0x31, 0x32, 0x33, 0x34 // 分配内存, 并初始化数组
```

Registers 1

Name	Value
ABI, RV32I	
pc	0x080001d6
ra	0x080001c2
sp	0x20007fe0
gp	0x20000800
tp	0x20000000
a0	0x20000000
a1	0x00000000
a2	0x20000000
a3	0x00000000
a4	0x00000000
a5	0x00000000
a6	0x00000000
a7	0x00000000
t0	0x00000000
t1	0x080001bc
t2	0x00000000
t3	0x00000000
t4	0x00000000
t5	0x00000000
t6	0x00000000
s0	0x080002c8
s1	0x080002c8
s2	0x00000000
s3	0x00000000
s4	0x00000000
s5	0x00000000
s6	0x00000000
s7	0x00000000
s8	0x00000000
s9	0x00000000
s10	0x00000000
s11	0x00000000

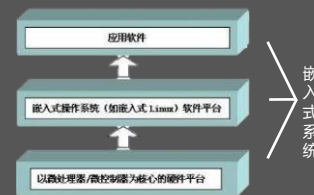
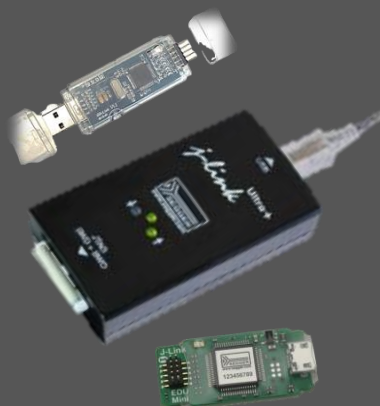
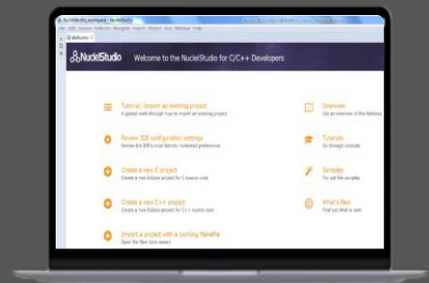
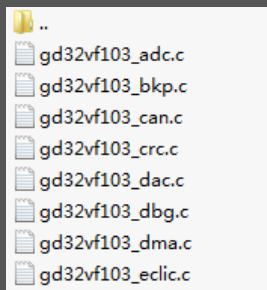
Code Outline

- #includes
  - ccode()
  - array0
  - sum()
  - macro()
  - main()

Memory 1

Address: 0x20000000	Size: Auto	Columns: Auto
20000000	00 00 00 00 00 00 04 00 00	13 00 00 00 14 00 00 00
20000010	21 00 00 00 00 22 00 00 00	23 00 00 00 24 00 00 00
20000020	31 00 00 00 00 32 00 00 00	33 00 00 00 34 00 00 00
20000030	44 33 22 11 55 44 33 22	66 55 44 33 77 66 55 44
20000040	88 77 66 55 99 88 77 66	00 99 88 77 11 00 99 88
20000050	22 11 00 99 11 00 99 88	33 22 11 99 11 00 99 88
20000060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
20000070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
20000080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
20000090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
200000A0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
200000B0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
200000C0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

# GD RISC-V MCU: 一站式快速产品开发 RISC-V



## Software Library

GD32V Library  
NMSIS

## IDE

Nuclei Studio  
RT-Thread Studio  
SEGGER Embedded Studio  
IAR Embedded  
Workbench for RISC-V  
MounRiver Studio  
卡姆派乐IDE  
Platform IO

## Program & Debug Tool

GD-Link  
SEGGER J-Link V10  
IAR I-Jet  
OPENOCD

## Embedded OS

μC/OS II  
FreeRTOS  
RT-Thread  
LiteOS  
TencentOS Tiny  
OneOS

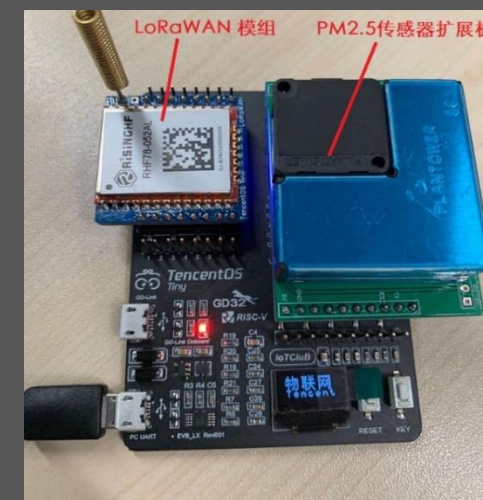
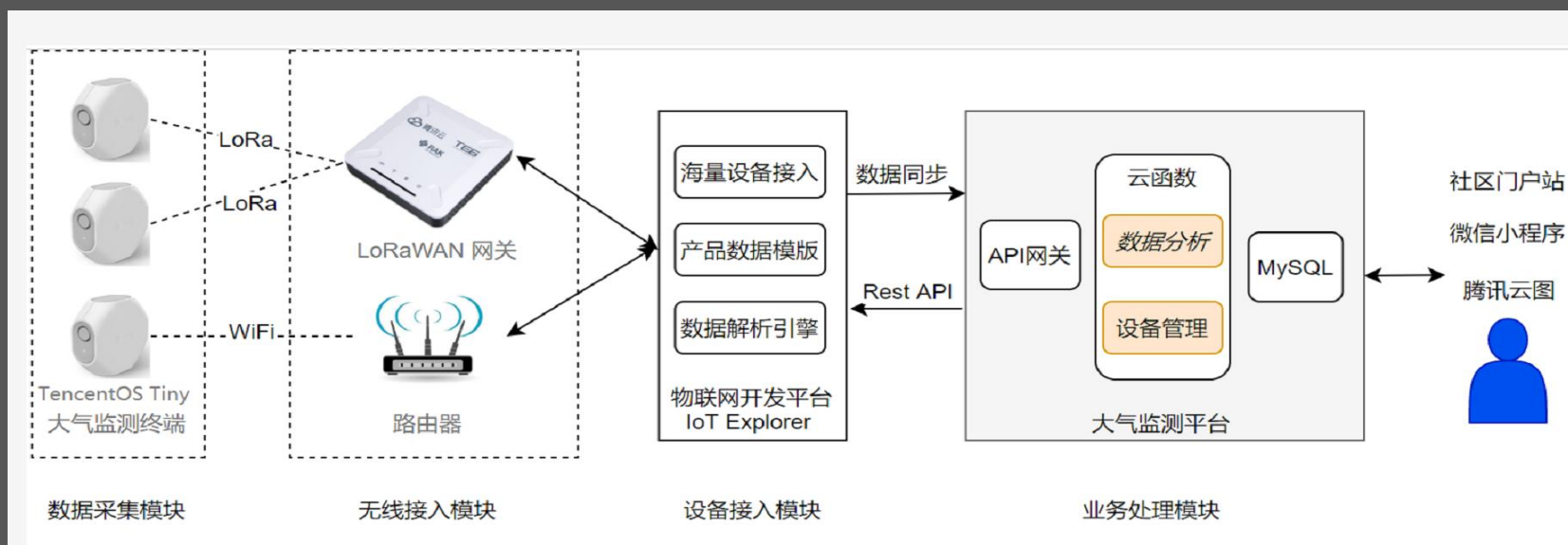
## Cloud Link

AWS  
Tencent Cloud

# TencentOS Tiny

# RISC-V

- 腾讯物联网终端操作系统TencentOS Tiny 由一个轻量级 RTOS 内核加多个物联网组件构成，具有低功耗，低源占用，模块化，安全可靠等技术特点，可有效提升物联网终端产品开发效率。
- 空气质量检测终端项目是一个以腾讯云物联网平台为基础，叠加多款产品后，实现的一个验证型项目。项目是腾讯与志愿者们共同建设一套用于监测大气环境的系统，监测终端分布在志愿者身边，提供更细粒度的数据监测能力。我们的图书分享了如何使用 TencentOS tiny，RISC-V定制开发套件打造一个PM2.5 监测终端过程。



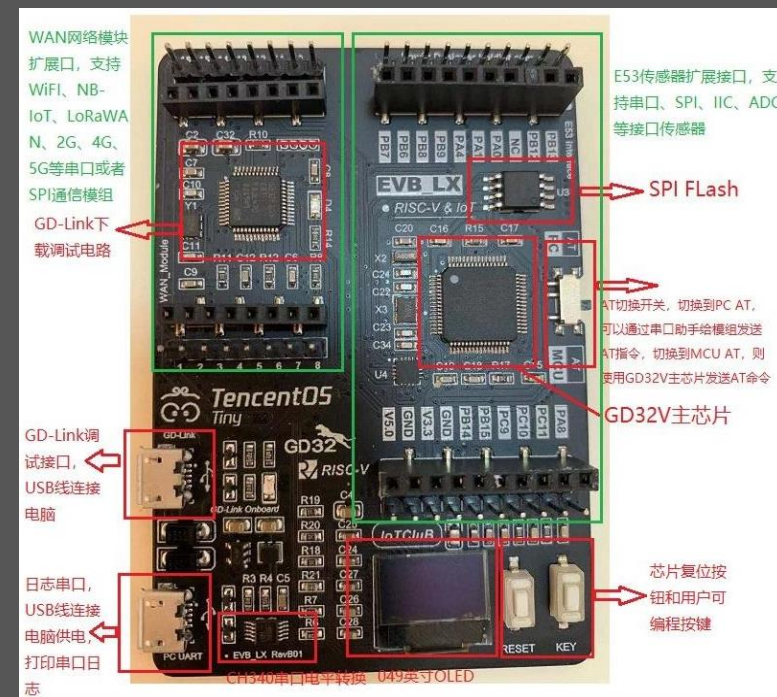
PM2.5监控终端

本示例代码已经在图书实验示例的网站页面 见P27 二维码

# TencentOS Tiny IoT 入门实例

# RISC-V

1. TencentOS Tiny内核在 GD32V RISC-V 芯片上的移植指南。  
TencentOS Tiny EVB\_LX RISC-V开发板软硬件**环境简介及搭建、内核移植流程介绍。**
2. 基于TencentOS Tiny 和RISC-V芯片 的光照传感器数据采集基于 TencentOS Tiny **采集环境光照强度**，并显示在RISC-V**开发板液晶屏和串口助手上。**
3. 基于TencentOS Tiny 和RISC-V芯片对接腾讯云物联网开发平台 TencentOS Tiny物联 网操作系统**AT框架简介**、如何基于RISC-V开发板和通信模组对接**腾讯云物联网开发平台IoT explorer**、并将环境光照强度上传，如何使用**腾讯连连小程序进行传感器数据展示。**



本示例将在“物联网操作系统公开课”介绍，并上传到图书实验示例的网站页面



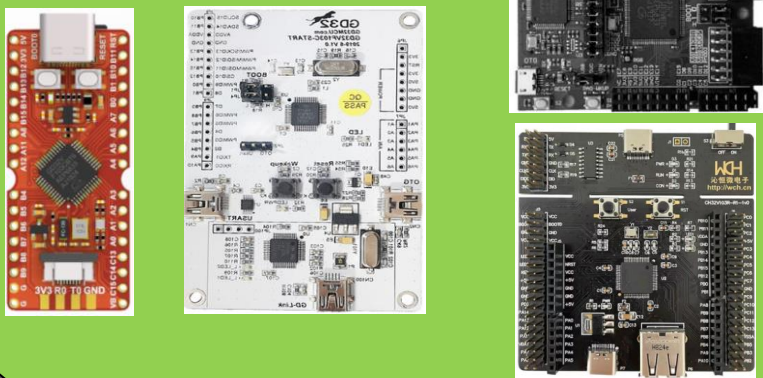
# RISC-V 教育

# RISC-V

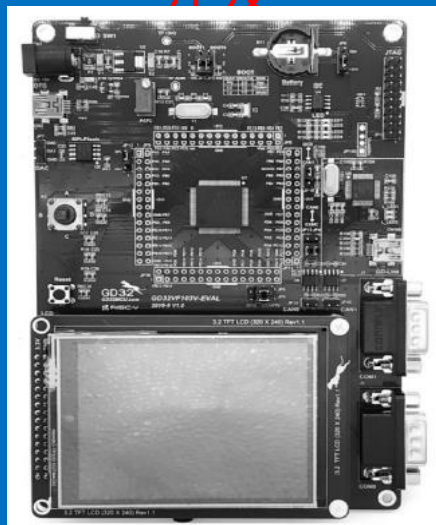
MIT, 清华、罗切斯特、剑桥、约克、纽约、康奈尔、弗吉尼亚大, 南京大学和清华大学等高校开设 RISC-V 内容的课程。毫无疑问 RISC-V 非常适合高校电子信息 and 计算机技术相关研究项目和教育课程!

我们将进一步完善本书的课件和视频教程, 配合产业链企业支持高校老师开设基于 RISC-V 指令集架构的处理器技术、程序设计以及嵌入式与物联网课程。

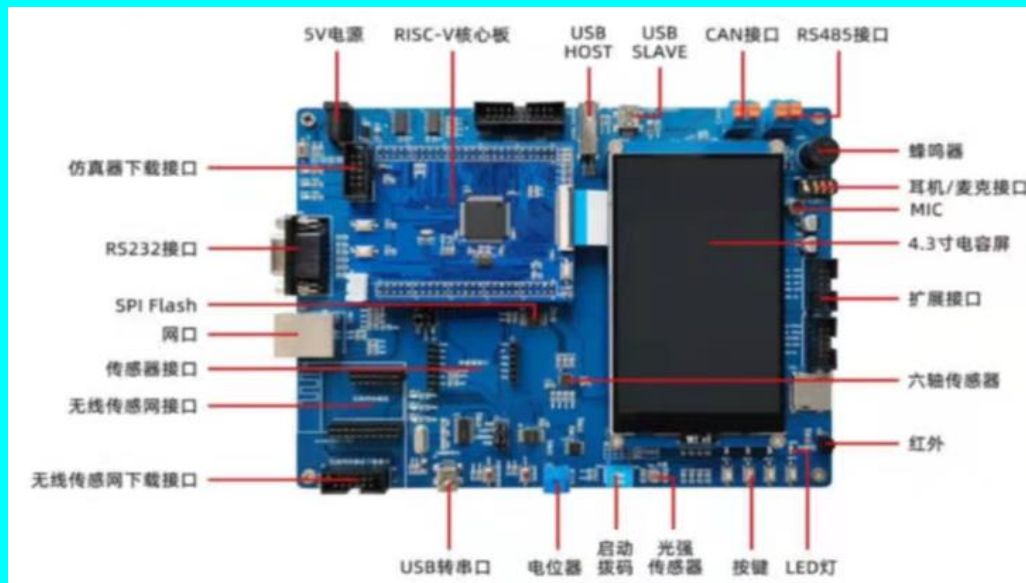
## 入门



## 开发



## 教学

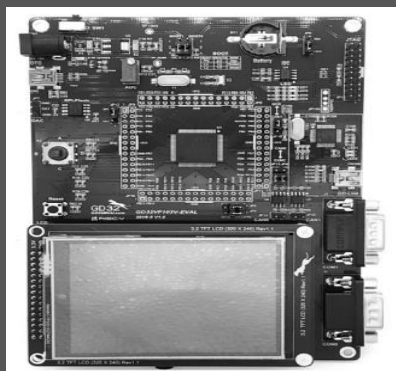




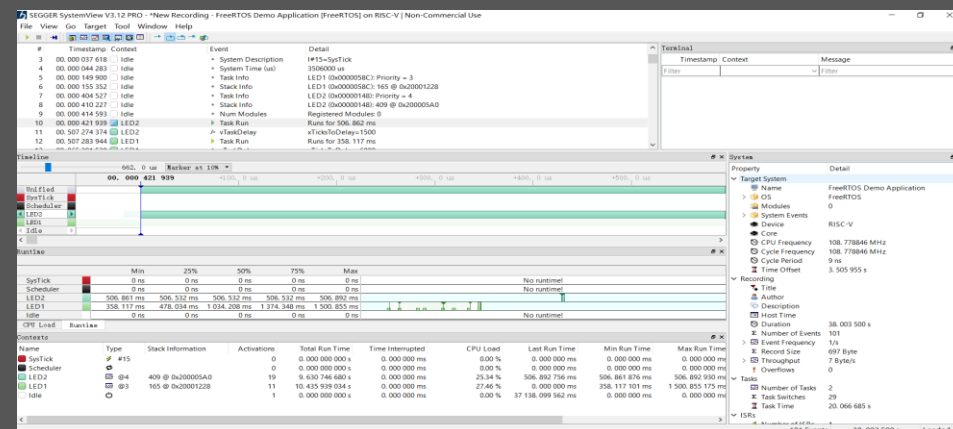
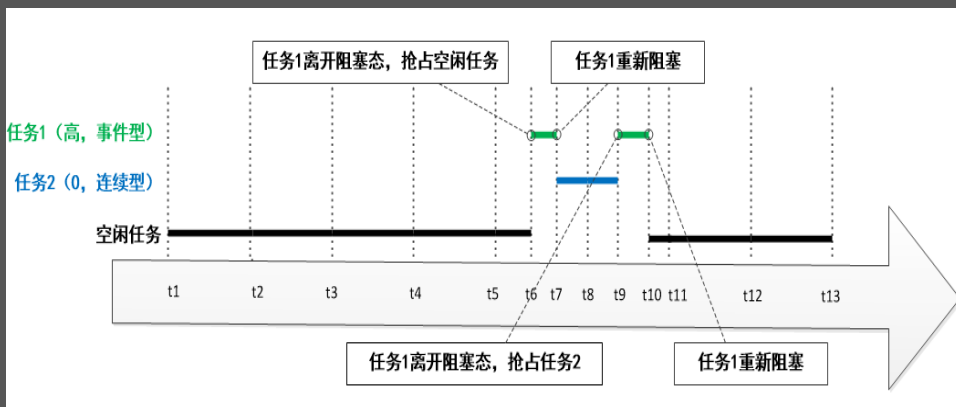
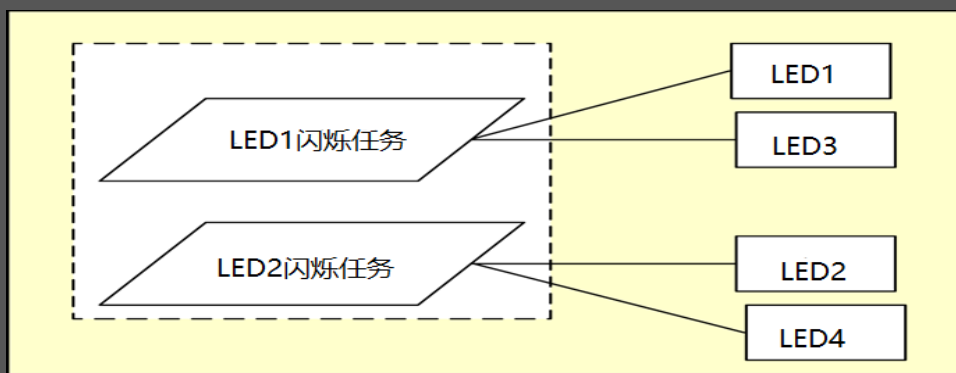


# 基于RISC-V RTOS 任务调度分析

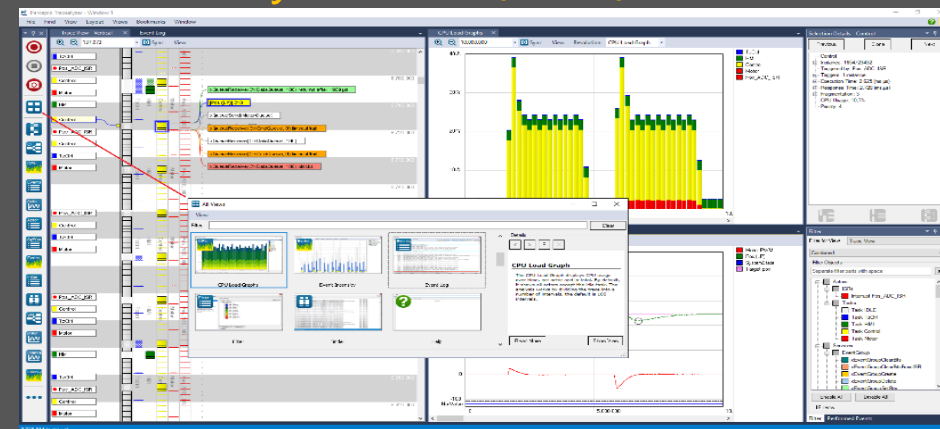
- 简单循环、定时循环和合作式调度
  - 协同调度- 由任务而不是调度器决定调度
  - 时间片轮询调度-定时循环调度
- 基于优先级调度策略
  - 抢占式调度



GD32VF103-EVAL



SystemView(v3.30)



Tracealyzer(v4.5.3)

- 计算架构将快速演进，异构多核计算时代已经来临，**RISC-V 将与ARM和X86 同行发展。**
- 具备**AI 属性的IoT应用(AIoT)**是智能系统发展新潮流，许多新 RISC-V 芯片是**AIoT 型SoC**。
- 嵌入式系统将**迎接RISC-V 时代**，基于RISC-V 的智能产品开发是未来热点。
- RISC-V 适合高校**研究项目和教育课程**，一种全新开源硬件模式，可将应用-处理器指令架构打通，构建**创新计算环境**！
- RISC-V 开发技术的学习：**调研-内核-芯片-工具和软件- 应用开发的路径展开。**

## 感谢您的聆听



欢迎扫描下面的二维码访问个人主页获取课件、  
实验代码、演讲稿和文章

