

RT-Thread助力全球开发者共建 RISC-V嵌入式操作系统生态





1 RISC-V内核的支持

RISC-V BSP的支持

3 开发环境支持

开源社区支持

RT-Thread & RISC-V的未来 展望



2

4

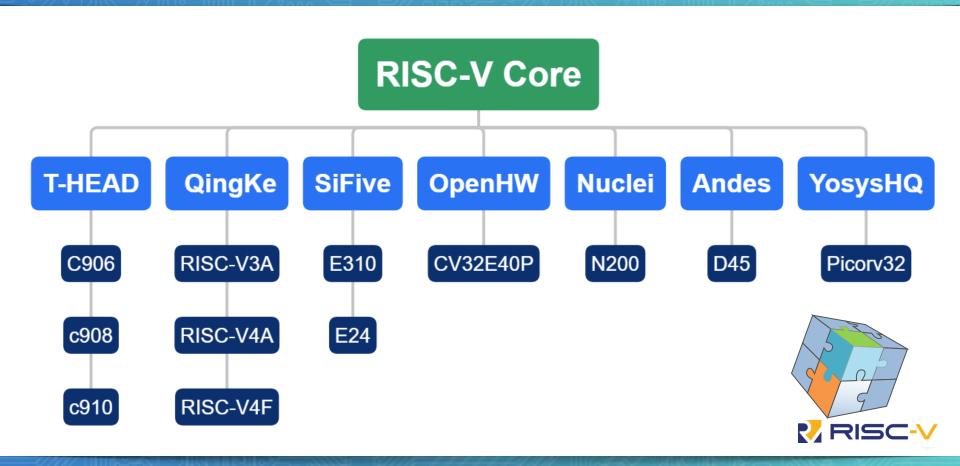


1. RISC-V内核的支持



RT-Thread支持的RISC-V内核

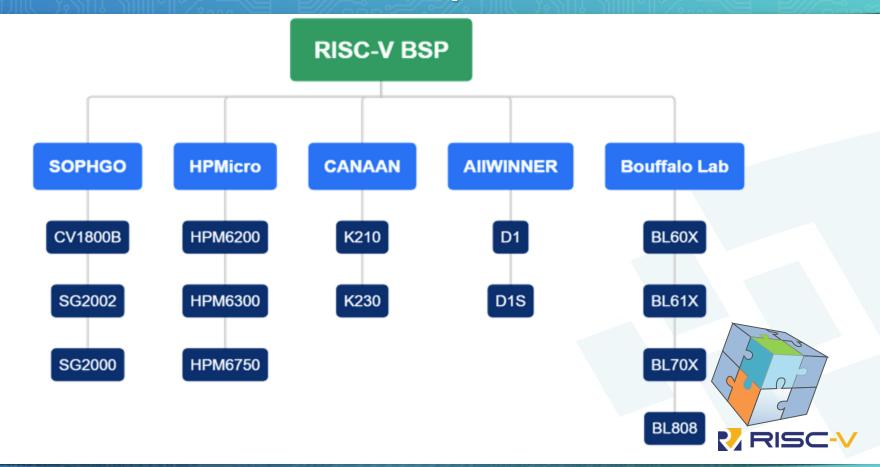






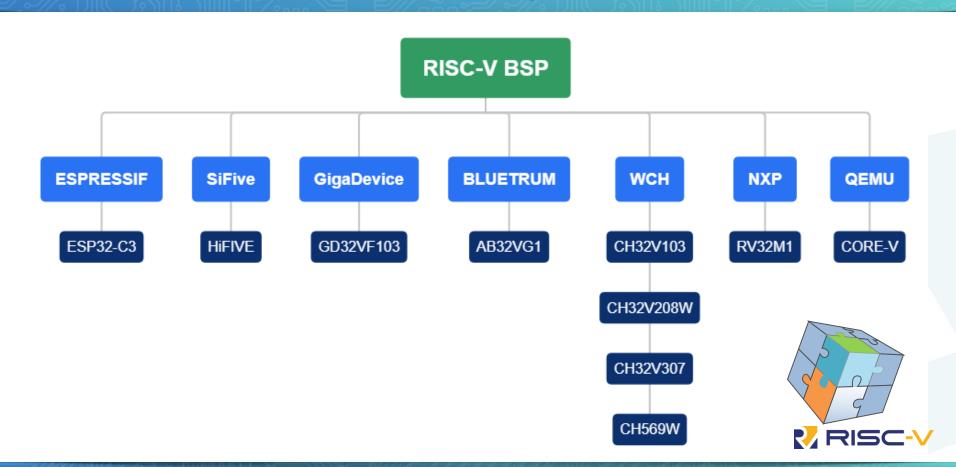
RT-Thread支持的RISC-V Chip





RT-Thread支持的RISC-V Chip





RT-Thread支持的RISC-V 开发板

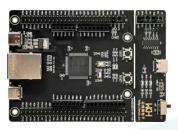




HPM6750EVKMINI



HPM5300EVK



CH32V307V-R1



AB32VG1-ab-prougen



HiFive1 Rev B01



全志D1S RDC2022纪念版



RT-Thread支持的RISC-V 开发板





Milk-V duo



Milk-V duo256M







GD32VF103V





K210

K230



3.开发环境支持





持



首页

动态

教程

贡献

RT-Thread 软件包

请输入搜索内容

Q)

RT-Thread 软件包是运行在 RT-Thread 物联网操作系统平台之上的通用代码库 这里是软件包开放平台,你能发现各类最新、最流行的软件包

4

简单易用,提高效率

查看软件包

文档丰富, 轻松上手

查看文档

(2)

社区活跃,沟通及时

前往社区

696 软件包数量

13,450,814

软件包下载量

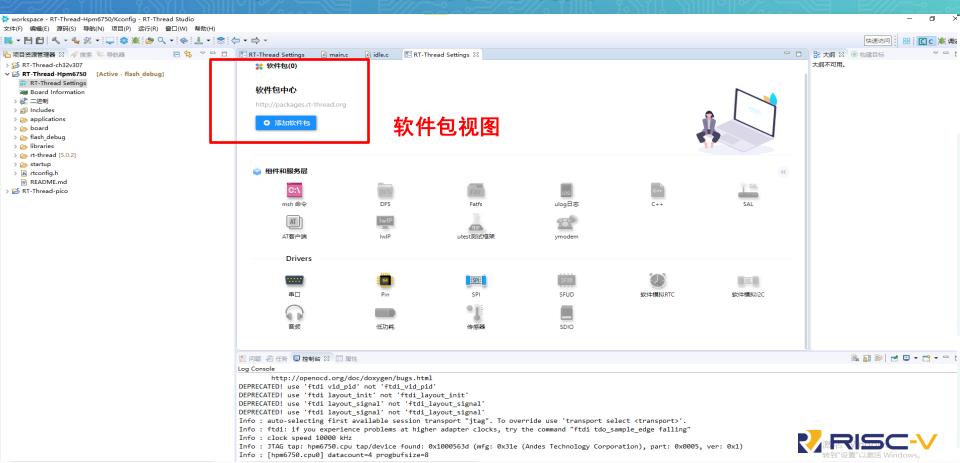
204

贡献者数量

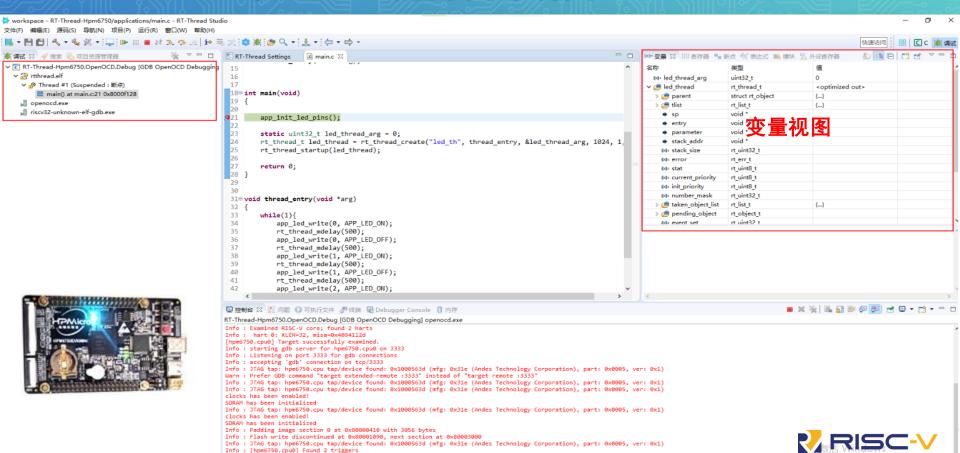
软件包分类



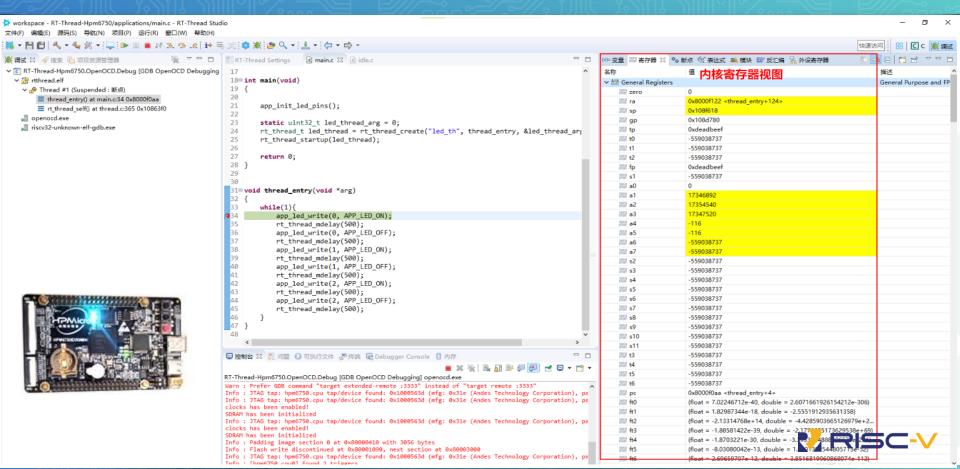








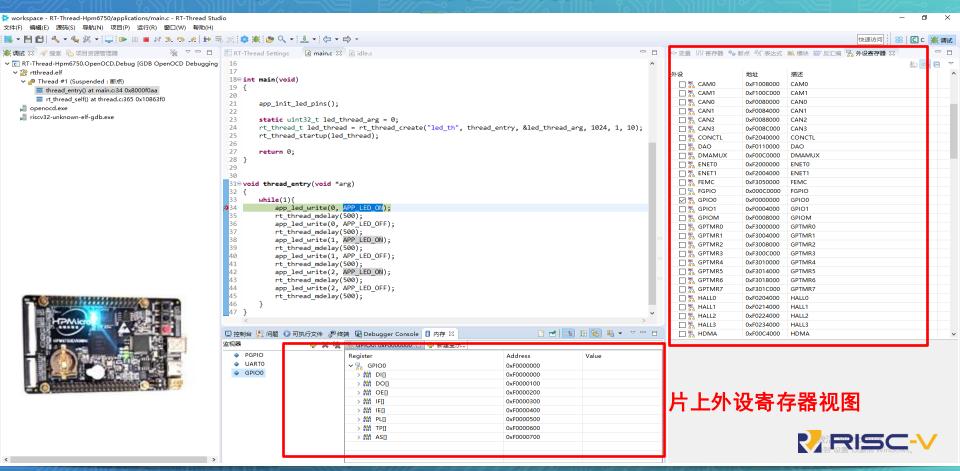




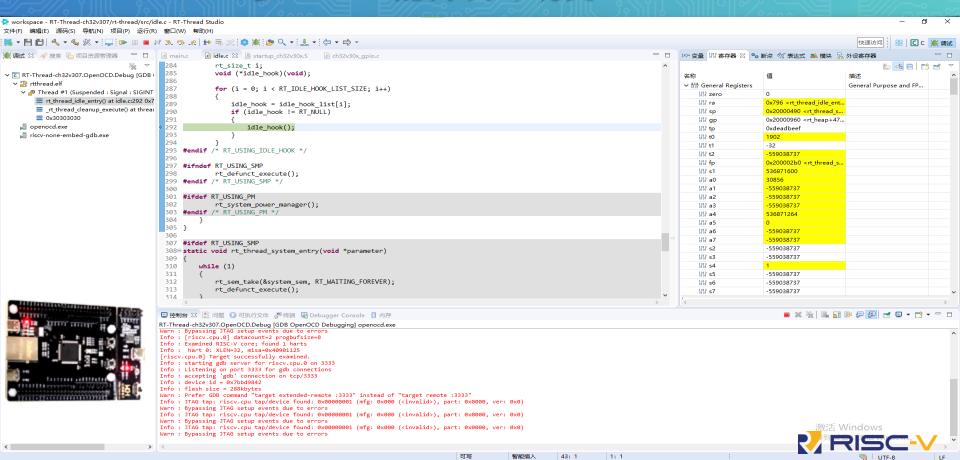


🦤 workspace - RT-Thread-Hpm6750/applications/main.c - RT-Thread Studio 文件(F) 编辑(E) 源码(S) 导航(N) 项目(P) 运行(R) 窗口(W) 帮助(H) 📕 + 📔 🔁 | 🔦 + 🧠 🕉 + 🛄 🕟 👊 🔳 🕬 🖎 🧟 🔆 | 📦 🔍 - + i 👤 + i 🗇 + 🖈 + 快速访问 : 器 C C 🕽 🕌 调试 🛭 🧳 搜索 🔓 项目资源管理器 变量 888 寄存器 ⁰。断点 ⋘ 表达式 🛋 模块 🎫 反汇编 🖂 晃 外设寄存器 ▼ RT-Thread-Hpm6750.OpenOCD.Debug IGDB OpenOCD Debugging √ ② ⑤ ⑤ ⑤ □ Enter location here 18@ int main(void) while (*str != '\0') ∨ № Thread #1 (Suspended: 断点) 19 { 8000f088: 0x8000f098 <rt hw console output+24> uart send byte(BOARD APP UART BASE, *str++); 20 thread entry() at main.c:34 0x8000f0aa 8000f08a: 21 app init led pins(); rt thread self() at thread.c:365 0x10863f0 8000f08c: a1,a1,255 22 a openocd.exe 8000f090: a0.0xf0074 23 static uint32 t led thread arg = 0: ra,0x8000df62 <uart 8000f094: riscv32-unknown-elf-qdb.exe 24 rt thread t led thread = rt thread create("led th", thread entry, &led thread arg while (*str != '\0') 8000f098: 1b a1,0(s0) 25 rt thread startup(led thread); 8000f09c: bnez al,0x8000f08a <rt hw console output+10: 26 27 return 0; 8000f09e: ra,12(sp) 28 8000f0a0: lw 29 8000f0a2: addi sp, sp, 16 30 8000f0a4: thread_entry: ■31@ void thread_entry(void *arg) 8000f0a6: addi sp,sp,-16 8000f0a8: ra,12(sp) while(1){ app led write(0, APP LED ON); app led write(0, APP LED ON): 8000f0ac: li rt thread mdelay(500); 8000f0ae: ra,0x8000f016 <app led write> app_led_write(0, APP_LED_OFF); rt thread mdelay(500); rt thread mdelay(500): 8000f0b2: app led write(1, APP LED ON); 8000f0b6: auinc ra.0x81078 rt thread mdelay(500); 8000f0ba: -1776(ra) # 0x10869c6 <rt thread mdelay> app led write(0, APP LED OFF); app led write(1, APP LED OFF); 8000f0be: rt thread mdelay(500); 8000f0c0: 1i app_led_write(2, APP_LED_ON); 8000f0c2: ra,0x8000f016 <app led write> rt thread mdelay(500); rt thread mdelay(500); app led write(2, APP LED OFF); 8000f0c6: a0.500 8000f0ca: ra,0x81078 rt thread mdelay(500): 8000f0ce: -1796(ra) # 0x10869c6 <rt thread mdelay> app led write(1, APP LED ON); 47 3 8000f0d2: a1.0 48 8000f0d4: li 8000f0d6: ra,0x8000f016 <app_led_write> rt_thread_mdelay(500); 8000f0da: 8000f0de: auipc ra,0x81078 8000f0e2: -1816(ra) # 0x10869c6 <rt thread mdelay> RT-Thread-Hpm6750.OpenOCD.Debug [GDB OpenOCD Debugging] openocd.exe app led write(1, APP LED OFF); Warn : Prefer GDB command "target extended-remote :3333" instead of "target remote :3333" 8000f0e6: 8000f0e8: li Info : JTAG tap: hpm6750.cpu tap/device found: 0x1000563d (mfg: 0x31e (Andes Technology Corporation), pa Info : JTAG tap: hpm6750.cpu tap/device found: 0x1000563d (mfg: 0x31e (Andes Technology Corporation), pa 8000f0ea: ra,0x8000f016 <app led write> rt_thread_mdelay(500); clocks has been enabled! 8000f0ee: a0,500 SDRAM has been initialized 8000f0f2: Info : JTAG tap: hpm6750.cpu tap/device found: 0x1000563d (mfg: 0x31e (Andes Technology Corporation), pa ra,0x81078 8000f0f6: -1836(ra) # 0x10869c6 <rt thread mdelay> clocks has been enabled! app led write(2, APP LED ON); SDRAM has been initialized Info : Padding image section 0 at 0x80000410 with 3056 bytes 8000f0fa: Info : Flash write discontinued at 0x80001090, next section at 0x80003000 8000f0fc: 8000f0fe: ra,0x8000f016 <app led write> Info : JTAG tap: hpm6750.cpu tap/device found: 0x1000563d (mfg: 0x31e (Andes Technology Corporation), pa rt thread mdelay(500); Info : [hpm6750.cpu0] Found 2 triggers 8000f102: Info : progbuf is writable at 0x30000080











```
🔖 workspace - RT-Thread-pico/applications/main.c - RT-Thread Studio
文件(F) 编辑(E) 源码(S) 导航(N) 项目(P) 运行(R) 窗口(W) 帮助(H)
# 調试 ※ 🛷 搜索 № 项目资源管理器
                                                          2 * Copyright (c) 2006-2021, RT-Thread Development Team

▼ T-Thread-pico,OpenOCD,Debug [GDB OpenOCD Debugging]

                                                                                                                                                                                                                                     ‱ → □ □ □ □

→ 

→ rtthread-pico.elf

                                                           4 * SPDX-License-Identifier: Apache-2.0
    ∨ № Thread #1.1 (Name: rp2040.core0, state: breakpoint) (Suspen-
                                                                                                                                                                                       ∨ MM General Registers
                                                                                                                                                                                                                                   General Purpose and FPU...
        main() at main.c;24 0x10000334
                                                                                                                                                                                           1010 rO
                                                           6 * Change Logs:

✓ 

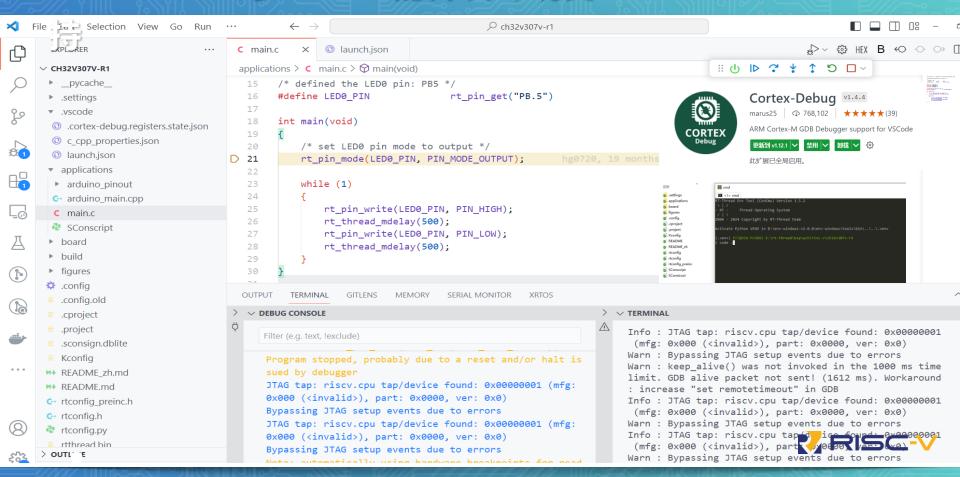
M Thread #2 2 (Name: rp2040.core1, state: debug-request) (Sus.)

                                                                                                                                                                                           1010 r1
                                                                                                                                                                                                             268435456
                                                              * Date
                                                                                 Author
                                                                                                Notes

≡ rt hw secondary cpu idle exec() at cpuport.c:87 0x10000f8²

                                                           8 * 2021-01-28
                                                                                                                                                                                           1010 r2
                                                                                 flybreak
                                                                                                 first version
        idle thread_entry() at idle.c:258 0x1000c9c0
                                                           9 */
                                                                                                                                                                                           1010 r3
        rt list remove() at rtservice.h:92 0x1000fc94
                                                           10
                                                                                                                                                                                           1010 r4
                                                                                                                                                                                                             -559038737
        ■ 0x0
                                                          11 #include <rtthread.h>
                                                                                                                                                                                           1010 r5
                                                                                                                                                                                                             -559038737
                                                          12 #include <rtdevice.h>
    openocd.exe
                                                                                                                                                                                           1010 r6
                                                                                                                                                                                                             -559038737
    arm-none-eabi-adb.exe
                                                                                                                                                                                           1010 r7
                                                                                                                                                                                                             536879624
                                                          14 #define LED PIN 25
                                                                                                                                                                                           1010 r8
                                                                                                                                                                                                             -559038737
                                                          15
                                                                                                                                                                                           1010 r9
                                                                                                                                                                                                             -559038737
                                                          160 int main(void)
                                                                                                                                                                                           1010 r10
                                                                                                                                                                                                             -559038737
                                                          17 {
                                                                  rt kprintf("Hello, RT-Thread!\n");
                                                                                                                                                                                           1919 r11
                                                                                                                                                                                                             -559038737
                                                                                                                                                                                           1919 r12
                                                                                                                                                                                                             536879552
                                                          20
                                                                  rt pin mode(LED PIN, PIN MODE OUTPUT);
                                                                                                                                                                                           1010 sp
                                                                                                                                                                                                             0x20002208
                                                          21
                                                                                                                                                                                           1919 Ir
                                                                                                                                                                                                             268485793
                                                          22
                                                                  while (1)
                                                                                                                                                                                           1919 pc
                                                                                                                                                                                                             0x10000334 <main+20>
                                                          23
                                                                                                                                                                                           IIII xPSR
                                                                                                                                                                                                             1627389952
                                                         24
                                                                       rt pin write(LED PIN, 1);
                                                                                                                                                                                           1010 msp
                                                                                                                                                                                                             0x20041fe0
                                                          25
                                                                       rt thread mdelay(100);
                                                                                                                                                                                                             0x20002208
                                                                                                                                                                                           IIII psp
                                                          26
                                                                       rt pin write(LED PIN, 0);
                                                                                                                                                                                           IIII primask
                                                          27
                                                                       rt thread mdelay(100);
                                                          28
                                                                                                                                                                                           ₩ basepri
                                                                                                                                                                                                             0
                                                          29 }
                                                                                                                                                                                           1818 faultmask
                                                          30
                                                                                                                                                                                           WW control
                                                          31
                                                         □ 控制台 ※ 原 问题 ② 可执行文件 ₽ 终端 ■ Debugger Console □ 内存
                                                         RT-Thread-pico.OpenOCD.Debug [GDB OpenOCD Debugging] openocd.exe
                                                         xPSR: 0x01000000 pc: 0x00000184 msp: 0x20041f00
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0x01000000 pc: 0x00000184 msp: 0x20041f00
                                                         Warn : keep_alive() was not invoked in the 1000 ms timelimit. GDB alive packet not sent! (4951 ms). Workaround: increase "set remotetimeout" in GDB
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0xf1000000 pc: 0x0000000ea msp: 0x20041f00, semihosting
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0xf1000000 pc: 0x0000000ea msp: 0x20041f00
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0x21000000 pc: 0x10000f84 psp: 0x200010b0
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0x21000000 pc: 0x10000f84 psp: 0x200010b0
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0x21000000 pc: 0x10000f84 psp: 0x200010b0
                                                         target halted due to debug-request, current mode: Thread
                                                         xPSR: 0x21000000 pc: 0x10000f84 psp: 0x200010b0
```







常 RT 計 → cad 文档中心

>简介

√快速 上手

Keil模拟器STM32F103

RT-Thread潘多拉STM32L475

野火霝道STM32F103

正点原子nanoSTM32F103

野火挑战者STM32F429

NXP FRDM-MCXN947

正点原子探索者STM32F407

正点原子阿波罗STM32F429

野火I.MX RT1052

正点原子号令者I.MX RT1052

瑞萨RA2L1-cpk

Nordic-nRE5x

英飞凌 CY8CKIT-062S2-43012

OPENHW Core-V-MCU

1 MCU 简介

2 编译说明

3 使用说明

其他开发板...

>内核

>设备和驱动

> Device IPC

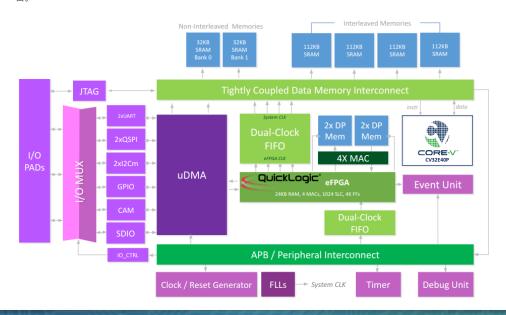
>组件

修改此文档
 ◎ 《PR提交规范》

core-v-mcu 上手指南

1 MCU 简介

core-v-mcu 的目的是展示 cv32e40p,这是 Open Hardware Group(OpenHW) 提供的经过充分验证的 RISC-V 内核。cv32e40p 核心连接到一组具有代表性的外围设备。









- 1. 支持的RISC-V (MCU/MPU) BSP数量多达30个
- 2. 在RISC-V MPU上支持 RT-Thread/RT-Smart
- 3. 简单易上手的RISC-V MCU移植框架
- 4. RISC-V64支持backtrace、mmu、atomic ...
- 5. RISC-V64支持RT-Thread SMP框架
- 6. QEMU-Virt64-RISCV BSP 支持 RV641LP32新32 位工具链



RT-Thread对RISC-V的移植支持



搜索

新手指导 标准版本 Nano版本 Smart版本 开发工具

♠ RT-Thread 文档中心

RT-Thread 标准版本

- >简介
- >快速上手
- >内核
- >设备和驱动
- > Device IPC
- / Device ir
- >组件
- >软件包 ~应用开发
- >开发环境搭建
- **~内核移植笔记**

Cortex-M 移植

RISC-V 移植

RISC-V 简介

内核移植简介

RISC-V 内核移植

参考文档

固件尺寸优化

在RT-Thread潘多拉开发板上实现电源管理

在RT-Thread小熊派开发板连接华为 云

网络协议栈驱动移植

在STM32F429上应用网络功能

✓ 修改此文档 ■《PR提交规范》

RISC-V 移植

RISC-V 简介

RISC-V (读作 "RISC-FIVE") 是 **基于精简指令集计算 (RISC)原理建立的开放 指令集架构** (ISA),V 表示为第五代 RISC (精简指令集计算机),表示此前已经有四代 RISC 处理器原型芯片。

内核移植简介

- 内核移植就是指将 RT-Thread 内核在不同的芯片架构、不同的板卡上运行起来,能够具备线程管理和调度,内存管理,线程间同步和通信、定时器管理等功能。移植可分为 CPU 架构移植和 BSP(Board support package,板级支持包)移植两部分。
- 在嵌入式领域有多种不同 CPU 架构,例如 Cortex-M、ARM920T、MIPS32、RISC-V 等等。为了使 RT-Thread 能够在不同 CPU 架构的芯片上运行,RT-Thread 提供了一个 libcpu 抽象层来适配不同的 CPU 架构。libcpu 层向上对内核提供统一的接口,包括全局中断的开关,线程栈的初始化,上下文切换等。
- RT-Thread 的 libcpu 抽象层向下提供了一套统一的 CPU 架构移植接口,这部分接口包含了全局中断开关函数、线程上下文切换函数、时钟节拍的配置和中断函数、Cache 等等内容。下表是 CPU 架构移植需要实现的接口和变量。

RTOS 内核移植

RTOS 内核移植一般需要实现如下功能,对接 RTOS

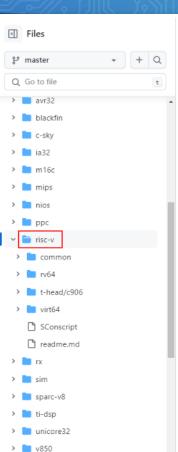
- 中断使能 / 失能
- 任务切换
- 中断处理
- 时钟节拍







↑ Тор



rt-thread / libcpu / risc-v / readme.md RV32移植指南

1.概述

为了简化32位RISC-V架构内核移植RT-Thread的流程,RT-Thread提供一分通用代码于common文件夹

文件名	文件内容
context_gcc.S	开关全局中断,线程上下文切换等
cpuport.c	线程栈初始化, 软件中断触发等
cpuport.h	数据加载写入指令
interrupt_gcc.S	线程间上下文切换
riscv-ops.h	控制状态寄存器读写
rt_hw_stack_frame.h	线程栈格式
trap_common.c	中断注册,中断查询分发

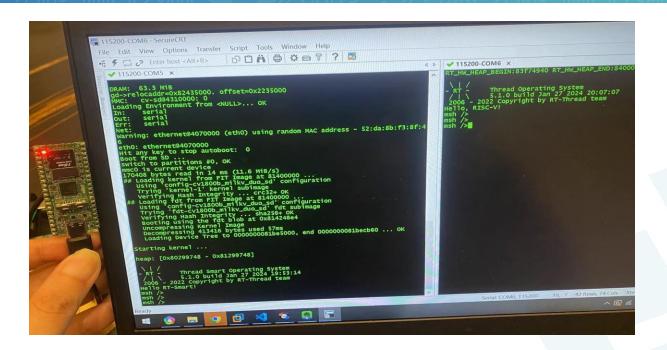
2.移植接口

(1) 软件中断触发函数,通常向量管理中断方式需实现该函数,非向量中断管理方式一般不需要

void rt_trigger_software_interrupt(void) (2) 保存上文后执行函数,该函数向量中断与非向量中断均需实现 RISC-V

void rt_hw_do_after_save_above(void)





RT-Thread社区开发者**燕十**三在RISC-V开发板milk-v duo上实现在大核(RISC-V C906):运行 RT-SMART,在小核(RISC-V C906 无MMU)运行RT-Thread。







首页

容

道

积分商城

专家

Z档中心

返回主站

Q 搜索问题、话题或文章

搜索







在 RT-Thread 嵌入式开源社区,交流问题,提升技能

我们一起学习技能,分享经验,发展职业前途,打造嵌入式开发者的乐园!







积分商城











搜索 RISC-V 结果



热门 未解决 已解决 未回答

104 kklam_涛 2024-03-04

移植rtt nano到Risc-v的时候, 出现rtthread.h:470: Error: unrecognized opcode

汇编 risc-v RISCV

LoveLonelyTime 2023-07-28

rt-thread-nano的risc-v下面的interrupt gcc.S文件存在问题

risc-v RISCV RT-Thread Nano

215 zhf 2023-05-30

导入wch\risc-v\ch32v307v-r1的BSP包,编译时候老是报错。发现编译器调用的编译器是 arm的,可是我riscv的我都装了啊 risc-v_RISCV 沁恒WCH

CORC_7424 - 2023-04-23

使用NUCLEI工具链编译RISC-V工程出现pthread定义冲突bug

pthread risc-v RISCV Nuclei 芯来科技

我爱茜茜公主 2023-03-27

rt studio 导入bsp\wch\risc-v\ch569w-evt 失败 RT-Thread Studio

推荐文章

- 1 RT-Thread应用项目汇总
- 2 玩转RT-Thread系列教程
- 3 机器人操作系统 (ROS2) 和 RT-Thread 通信
- 4 国产MCU移植系列教程汇总, 欢迎查看!
- 5 五分钟玩转RT-Thread新社区
- 6 【技术三千问】之《玩转ART-Pi》,看这篇就够 了! 干货汇总
- 7 关于STM32H7开发板上使用SDIO接口驱动SD卡 挂载文件系统的问题总结
- 8 STM32的"GPU"——DMA2D实例详解
- 9 RT-Thread隐藏的宝藏之completion
- 10 【ART-PI】RT-Thread 开启RTC 与 Alarm组件

最新文章











【RISC-V 移植】- 移植相关文件:

https://github.com/RT-Thread/rt-thread/tree/master/libcpu/risc-v/common

【RISC-V 移植】- 文档中心:

https://www.rt-thread.org/document/site/#/rt-thread-version/rt-thread-standard/application-note/porting/risc-v/port-risc-v

【RISC-V 移植 】- RISC-V MCU移植RT-Thread框架:

https://github.com/RT-Thread/rt-thread/tree/master/libcpu/risc-v

【问题检索与发布文章】- RT-Thread 论坛:

https://club.rt-thread.org/index.html





5. RT-Thread & RISC-V的未来展望



RT-Thread & RISC-V的未来展望

- 1. RT-Thread加入PLCT实验室面向RISC-V的甲辰计划,一起推动RISC-V开源软件
- 2. RT-Thread内核加入了玄铁RISC-V RTOS系统,共建AloT
- 3. RT-Thread正在推动RT-Smart微内核对RISC-V MPU的支持
- 4. RT-Thread成立RISC-V兴趣小组,推动RT-Thread对RISC-V组件与BSP的支持





谢谢!



