

玄铁RISC-V处理器 入门与实战



- 带你了解开源、精简、高效的玄铁RISC-V处理器
- 手把手教你基于玄铁处理器的开发流程



平头哥芯片开放社区交流群
扫码关注获取更多信息



平头哥 RISC-V 系列课程培训
扫码登录在线学习



扫码注册平头哥 OCC 官网
观看各类视频及课程



阿里云开发者“藏经阁”
海量电子手册免费下载

目录

RISC-V 处理器架构	5
1.RISC-V 架构起源	5
2.RISC-V 架构发展	5
3.RISC-V 架构与 X86、ARM 在商业模式上的区别	6
4.RISC-V 架构现状和未来	7
5.RISC-V 处理器课程学习	9
平头哥玄铁 CPU IP	10
1.概述	10
2.面向低功耗领域 CPU	10
3.面向中高端服务器 CPU	16
4.面向高性能领域 CPU	23
5.玄铁 CPU 课程学习	26
无剑平台	27
1.无剑 100 开源 SoC 平台	27
2.无剑 600 SoC 平台	28
平头哥 RISC-V 工具链	34
1.RISC-V 工具链简介	34
2.剑池 CDK 开发工具	37
3.玄铁 CPU 调试系统	44
4.HHB	51

5.剑池 CDK 开发工具课程学习	54
平头哥玄铁 CPU 系统	55
1.YoC	55
2.Linux	56
3.Android	62
RISC-V 玄铁系列开发板实践	67
1.基于玄铁 C906 处理器的 D1 Dock Pro 开发实践	67
2.基于玄铁 E906 处理器的 RVB2601 开发实践	82
RISC-V 应用领域开发示例	100
1.基于 D1 Dock Pro 应用开发示例	100
2.基于 RVB2601 应用开发示例	106
RISC-V 未来探索	116
1.平头哥开源 RISC-V 系统处理器	116
2.平头哥对 RISC-V 基金会贡献	117
3.高校合作	117

RISC-V 处理器架构

1.RISC-V 架构起源

RISC-V 架构是一种开源的指令集架构。最早是由美国伯克利大学的 Krest 教授及其研究团队提出的，当时提出的初衷是为了计算机/电子类方向的学生做课程实践服务的。由于这是伯克利大学研究并流片的第五代 RISC 架构处理器，因此就命名为 RISC-V，其中“V”是罗马字母的数字 5 的意思。2015 年，RISC-V 架构开始以基金会的形式运营，Google、Nvidia、WesternDigital 等国际知名的互联网和半导体企业都是其创始会员。

2020 年，RISC-V 基金会考虑到整个组织和技术的中立性，将总部由美国迁移到了欧洲的瑞士（瑞士一直是国际上有名的中立国，在第一次世界大战和第二次世界大战期间都保持了中立国的角色），并将名字由 RISC-V Foundation 改名为 RISC-V International，RISC-V 国际希望坚持架构的中立、开放与普惠，并服务于全球技术生态。



2.RISC-V 架构发展

在处理器发展的几十年历史中，各种处理器架构曾经层出不穷，从开始的 IntelX86 架构，到后来的 IBMPOWER 架构、MIPS 架构、DECAlpha 架构、SunSparc 架构以及后来的 ARM 架构，此外还有 Motorola、TI、Hitachi、RENESAS 等各种小众处理器架构等，各种处理器架构曾百花齐放过。

但是经过近 30 年产业的大浪淘沙，其他的处理器架构逐步地留在了历史的博物馆中，目前只留下了 X86 和 ARM 两大强势的处理器架构。未来，从全球开发者的角度看，一定是希望基于有限的处理器架构做上层软件的开发，因此可以基本确定的是未来只有 2-3 家处理器架构可以存活下来，而 RISC-V 有很大概率成为继 X86 和 ARM 架构之外，处理器架构的第三级（甚至是第二级）。

3.RISC-V 架构与 X86、ARM 在商业模式上的区别

我们介绍了 X86 架构是一种完全封闭的指令架构，相比之下 ARM 架构是一种“开放”的指令架构，只要获得了 ARM 的 IP 授权就可以使用 ARM 架构并享受之上沉淀的软件生态。但是 ARM 也只是“开放”，在 ARM 之后，RISC-V 出现了。

RISC-V 不是一种技术的创新（因为在此之前已经有非常多的成熟优秀的 RISC 架构了），而是商业模式的重大变革，RISC-V 直接将指令架构给“开源了”（注意此处是开源），这是之前完全没有的一种新的指令架构运营模式（虽然 OpenRISC 等也是开源，但是后文会分析为什么 RISC-V 会做得比 OpenRISC 成功）。

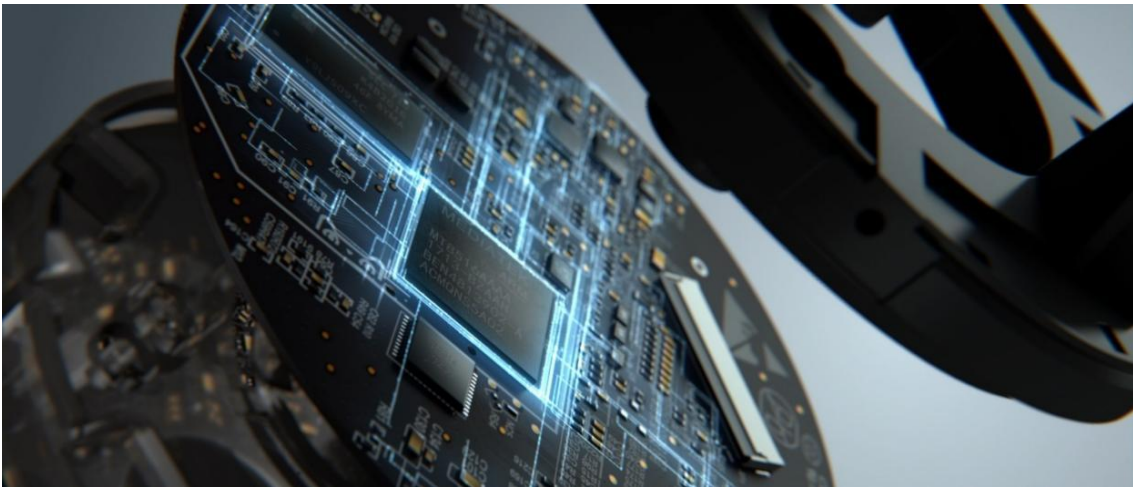
RISC-V 通过“指令架构开源”与“架构标准全球开放讨论”两大杀手锏吸引全球的处理器架构大拿、软件开发者与开源爱好者围绕着它做全球生态建设。读者看到这里是不是有种“众人拾柴火焰高”的感觉，这也是 RISC-V 架构的高明之处，说白了，RISC-V 是一种与 X86 和 ARM 完全不一样的运作方式，他将指令集架构推进到了一种最极致的方式：就是指令架构开源，全球无论是企业、研究所、个人都可以完全免费使用。

RISC-V 和 X86、ARM 的竞争完全是不同维度的竞争，他们三个分别是全球处理器技术演进在不同阶段的产物，而基本可以确定的是未来三种架构会长期共存，只是应用的侧重领域有所不同罢了。

看到这里，读者可能有一些疑问，在 RISC-V 之前，已经有 OpenRISC 等开源指令架构了，那为什么 OpenRISC 没起来，而是 RISC-V 起来了。

笔者认为这主要还是得益于伯克利团队的根正苗红以及恰到好处的运营方式，伯克利大学和斯坦福大学在历史上首次提出了精简指令集架构的处理器并完善了其基础理论，对处理器架构的发展做出了巨大贡献，所以当伯克利大学提出了 RISC-V 指令集并开源的时候迅速获得了工业界和开发者的高度关注，这之后 Krest 教授团队又采用第三方独立的基金会的方式进行运营，其“中立”和“开放”的属性吸引了大批的企业和开发者投身其中。

这之后，虽然 MIPS 也想模仿 RISC-V，但是由于其中立属性问题以及企业的日薄西山，MIPS 架构开源的事情草草收场。

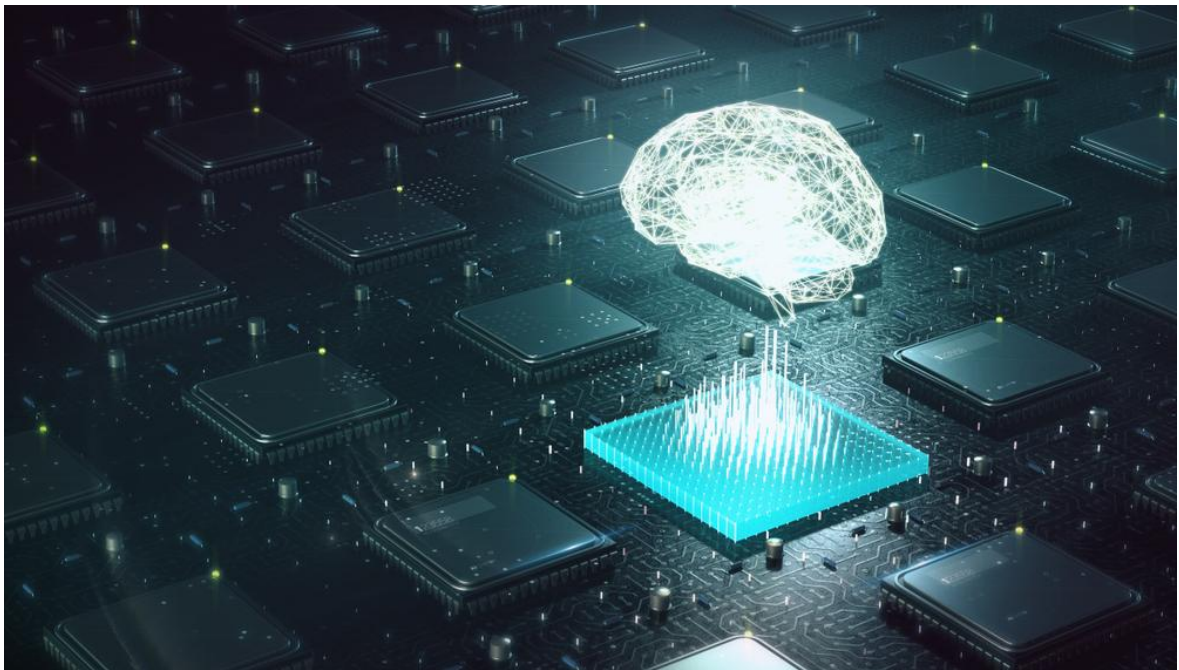


4.RISC-V 架构现状和未来

现在，虽然 RISC-V 架构还不完善，在诸如安全、虚拟化架构、IOMMU/SMMU、中断控制架构、RAS (Reliability, Availability and Serviceability) 等方面还刚起步，在代码密度 (code size)、虚拟内存管理、原子操作效率等方面也还存在一些缺陷，但这不会妨碍 RISC-V 架构的长远向好发展，因为其开源的本质不曾改变。

回顾 Linux 内核的发展历史，在 Linux 内核之前，IBM 的 Unix 收费操作系统无论在稳定性和用户体验上都是非常成功的，而相比之下刚出道的 Linux 内核无论在稳定性以及用户体验上都比较糟糕，但是由于 Linux 内核的开源属性迅速在 5 年时间内（1991 年到 1996 年）吸引了超过 350 万开发者的使用。

经过 30 年的发展，Linux 操作系统（基于 Linux 内核开发的各种操作系统）已经成为世界上最主流的操作系统之一（Windows 主要在桌面机，IOS/安卓主要在智能手机和平板电脑），无论在服务器、云计算以及嵌入式领域基本已是 Linux 内核的天下。



微软公司也在 2016 年加入了 Linux 基金会并成为 Linux 社区的主要贡献企业之一。同样的，RISC-V 虽然现在有不足和缺陷，但是基本可以确定的是，会有大量的企业、高校与个人爱好者会持续地围绕 RISC-V 开发并不停地推动 RISC-V 架构的成熟完善。

这里需要稍微引申一下的是，虽然 RISC-V 架构本身是开源的，但是基于 RISC-V 架构开发的 CPU IP 核可以是收费的，这个道理很像开源的 Linux 与收费的 Redhat Linux 一样，Redhat 可以提供相较开源 Ubuntu/Debian 更好的稳定性及对上层软件的兼容性，同理的，商业的 RISC-V IP 核可以提供比开源核更好的稳定性、发展连续性、售后服务等。

5.RISC-V 处理器课程学习

平头哥芯片开放社区官网上线了 RISC-V 系列课程培训，该课程和清华大学、大连理工大学等合作，包括由教育部评选为“国家级一流本科课程”的《嵌入式软件设计》，成为国内嵌入式教育的标杆课程。在“中国大学 MOOC”平台已开设 9 轮课程，累计选课人数达四万人次以上，广受学生欢迎。通过 XuanTie RISC-V Institute，可以学习该课程，通过云上实验室远程操作开发板进行实战应用。

欢迎广大开发者[登录 OCC](#)了解学习，学习后还可以参加在线 RISC-V 课程考试，考试通过后会颁发电子证书，进入账号【个人中心】可以查看到。

XuanTie RISC-V Institute

初识RISC-V处理器

课程目录 课程要点 课程问答

内容简介

RISC-V处理器架构是一种开源的指令集架构。最早是由美国伯克利大学的Krest教授及其研究团队提出的，当时提出的初衷是为了计算机/电子类方向的学生做课程实践服务的。由于这是伯克利大学研究并流片的第五代RISC架构处理器，因此命名为RISC-V，其中“V”是罗马字母的数字5的意思。2015年，RISC-V架构开始以基金会的形式运营，Google、Nvidia、Western Digital等国际知名的互联网和半导体企业都是其创始会员。

主讲教师：尹首一，集成电路学院，清华大学；赖晓晨，软件学院，大连理工大学

本章节主要介绍RISC-V指令集架构的基础知识，以及RISC-V生态发展的现状。

课程资料

1. 初识RISC-V处理器-清华大学教师：尹首一
2. RISC-V指令集概览-大连理工大学教师：赖晓晨

CERTIFICATE

合格证书

★★★★★

证书编号：OCCRVB23000001XX

XXXX

参加芯片开放社区“RISC-V课程考试”，通过考试，特此证明。

 XUAN TIE
RISC-V
INSTITUTE

芯片开放社区
Open Chip Community
2023.03.02

平头哥玄铁 CPU IP

1. 概述

平头哥玄铁 CPU 是智能、安全、端云一体芯片架构的基石，为数字化时代提供计算核心。玄铁自诞生以来，坚持核心技术自主研发道路，新系列产品积极拥抱开源 RISC-V 架构。玄铁 CPU 广泛应用于计算视觉、数据存储、工业互联、网络通信、智能家居、生物识别、信息安全等领域，截至目前，玄铁 CPU 累计授权芯片出货数量超过 30 亿颗。

2. 面向低功耗领域 CPU

1) 入门级微控制器：E902

a) 概述

玄铁 E902 采用 2 级极简流水线并对执行效率进行了增强，典型工作频率 > 150MHz，是首款支持硬件安全扩展技术的 RISC-V 处理器。可以应用在对功耗和成本极其敏感的 IoT、MCU 等场景。

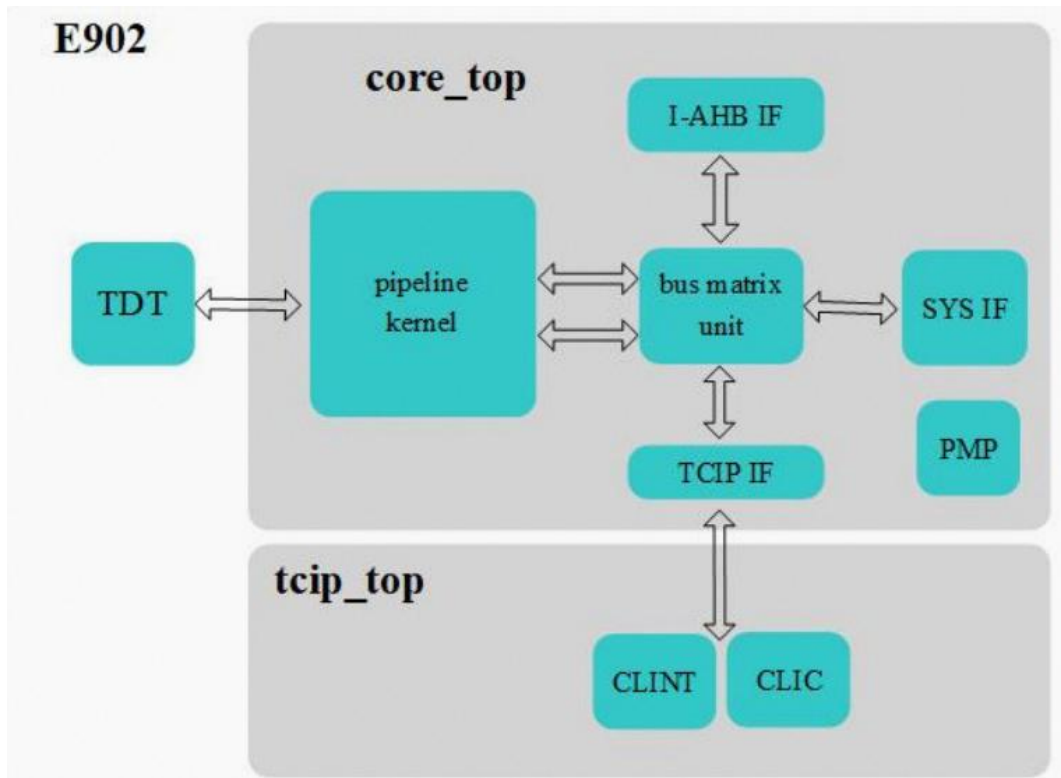
E902 处理器体系结构的主要特点如下：



特性	描述
架构	RV32E[M]C
流水线	2级
玄铁扩展	支持玄铁MCU增强扩展，包括中断加速和增强指令集
总线接口	AMBA3 AHB-Lite 32-bit 主接口
指令集cache	最大8KB(可选)
中断	最多64个中断源，以及1个不可屏蔽中断 (NMI)
睡眠模式	睡眠和深睡眠模式
调试	2线JTAG调试接口

b) 处理器简介

E902 的结构框图如下所示：



结构图 E902 处理器采用 2 级流水线结构：取指和执行。指令取指阶段主要负责从内存中获取指令；指令执行阶段主要负责指令译码、执行和回写。

可配置的物理内存保护单元（Physical Memory Protection, PMP）负责物理地址的权限检查实现内存的保护功能，权限可划分为：不可读写/只读/可读写，可执行/不可执行。

TDT 调试单元 (T-Head Debug Trace) 支持各种调试方式, 包括软件断点、内存断点、单步和多步的指令跟踪等多种方式, 可在线调试 CPU、通用寄存器 (GPR)、控制寄存器 (CSR) 和内存。

E902 设计有片上紧耦合的 IP 接口和多条 AHB-Lite 的总线接口。片上紧耦合的 IP 接口集成处理器中断控制器 (CLIC), 支持中断嵌套以及处理器核局部中断 (CLINT)。外部中断源数量最高可配置 240 个, 中断优先级支持 4/8/16/32 级可配置。

2) 高效微控制器: E906

a) 概述

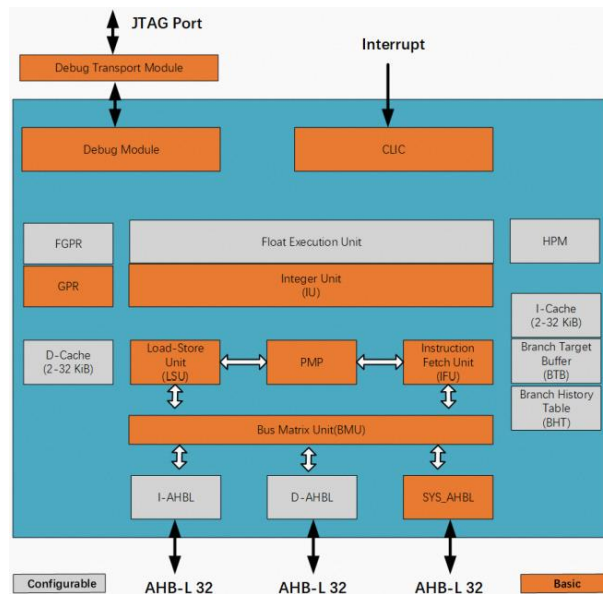
玄铁 E906 采用 5 级按序流水线, 典型工作频率 > 1GHz, 可选性能优异的单精度浮点单元以及标量 DSP 计算单元。可以应用在无线接入、音频、中高端 MCU、导航等场景。

E906 处理器体系结构的主要特点如下:

特性	描述
架构	RV32IMAFC
流水线	5级 (整型)
总线接口	AMBA3 AHB-Lite 32-bit 主接口
浮点单元	单精度浮点单元
DSP 增强	深度优化的DSP单元, 配合CSI-PSD库。兼容P扩展 v0.9.2
混合分支预测	BHT/BTB/RAS(可选)
指令 cache	16KB
数据 cache	16KB
中断	最多128个中断源, 以及1个不可屏蔽中断 (NMI)
硬件性能监测 (HPM)	RISC-V标准HPM (可选)
玄铁扩展	支持玄铁MCU增强扩展 包括中断加速和增强指令集
睡眠模式	睡眠和深睡眠模式
调试	RISC-V调试, 可配置不同的资源级别

b) 处理器简介

E906 的结构框图如下所示:



E906 处理器采用 5 级流水线结构：取指、译码、执行、内存访问、写回。

- 取指阶段，访问指令 Cache 或者总线，获取指令，同时访问 BTB，发起 0 延时跳转。
- 译码阶段，访问动态分支预测器和返回栈，发起分支的预测跳转，同时进行指令译码，读取寄存器堆，处理数据相关性和数据前馈。
- 执行阶段，完成单周期整型计算指令和多周期乘除法指令的执行、存储/加载指令地址计算和跳转指令处理。其中，整型计算包括普通的算术指令和逻辑指令。
- 内存访问阶段，利用执行阶段产生的存储/载入指令的目标地址访问数据 Cache 或者总线。
- 写回阶段，将指令执行结果写回寄存器堆。

可配置的物理内存保护单元（Physical Memory Protection, PMP）负责物理地址的权限检查，实现内存的保护功能。权限可划分为：不可读写/只读/可读写，可执行/不可执行。

调试单元支持各种调试方式，包括软件断点、内存断点、单步和多步的指令跟踪等，可在线调试 CPU、通用寄存器（GPR）、控制寄存器（CSR）和内存。

E906 设计有片上紧耦合的 IP 接口和多条 AHB-Lite 的总线接口。片上紧耦合的 IP 接口集成矢量中断控制器 (CLIC)，支持中断嵌套。外部中断源数量最高可配置 240 个，中断优先级支持 4/8/16/32 级可配置。

3) 计算增强型微控制器：E907

a) 概述

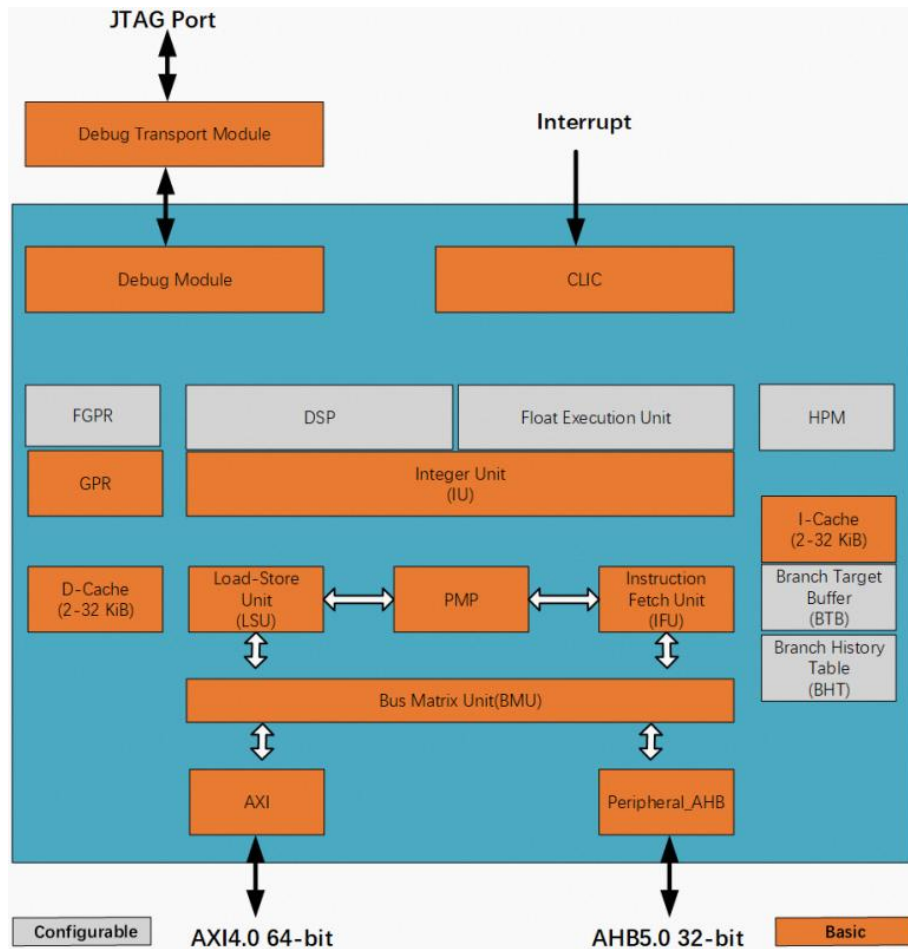
玄铁 E907 采用 5 级按序流水线，典型工作频率 > 1GHz，是玄铁 MCU 处理器中的性能最高的处理器核，可选配高性能浮点以及 DSP 计算单元，同时支持 TCM 扩展以及中断加速技术以进一步提升实时性。可以应用在语音入口 MCU、TWS、MPU、多模无线接入等场景。

E907 处理器体系结构的主要特点如下：

特性	描述
架构	RV32IMA[F][D]C[P]
流水线	5级 (整型)
主接口	AXI4.0 64-bit主接口
外设接口	AHB5 32-bit主接口
浮点单元	单精度浮点单元
DSP 增强	深度优化的DSP单元，配合CSI-DSP库 兼容P扩展v0.9.2
混合分支预测	BHT/BTB/RAS(可选)
指令cache	最大32KB(可选)
数据cache	最大32KB(可选)
中断	最多240个中断源，以及1个不可屏蔽中断(NMI)
硬件性能监测 (HPM)	RISC-V标准HPM(可选)
玄铁扩展	支持玄铁MCU增强扩展，包括中断加速和增强指令集
睡眠模式	睡眠和深度睡眠模式
调试	RISC-V调试，可配置不同的资源级别

b) 处理器简介

E907 的结构框图如下所示：



E907 处理器采用 5 级流水线结构：取指、译码、执行、内存访问、写回。

- 取指阶段，访问指令 Cache 或者外部总线，获取指令，同时访问 BTB，发起 0 延时跳转。
- 译码阶段，访问动态分支预测器和返回栈，发起分支的预测跳转，同时进行指令译码，读取寄存器堆，处理数据相关性和数据前馈。
- 执行阶段，完成单周期整型计算指令和多周期乘除法指令的执行、存储/加载指令地址计算和跳转指令处理。其中，整型计算包括普通的算术指令和逻辑指令。
- 内存访问阶段，利用执行阶段产生的存储/载入指令的目标地址访问数据 Cache 或者外部总线。
- 写回阶段，将指令执行结果写回寄存器堆。

可配置的物理内存保护单元（Physical Memory Protection, PMP）负责物理地址的权限检查实现内存的保护功能，权限可划分为：不可读写/只读/可读写，可执行/不可执行。

调试单元（Debug Module, 以下简称 DM）支持各种调试方式，包括软件断点、内存断点、单步和多步的指令跟踪等多种方式，可在线调试 CPU、通用寄存器（GPR）、控制寄存器（CSR）和内存。

E907 设计有片上紧耦合的 IP 接口和两条主设备总线接口。片上紧耦合的 IP 接口集成向量中断控制器（CLIC），支持中断嵌套。外部中断源数量最高可配置 240 个，中断优先级支持 4/8/16/32 级可配置。

3. 面向中高端服务器 CPU

1) 高效应用处理：C906

a) 概述

玄铁 C906 采用 5-8 级变长流水线，典型工作频率 > 1GHz，标配内存管理单元，可运行 Linux 等操作系统，并可选性能优异的单精度浮点和矢量运算单元。可以应用在消费类 IPC、多媒体、消费类电子等场景。

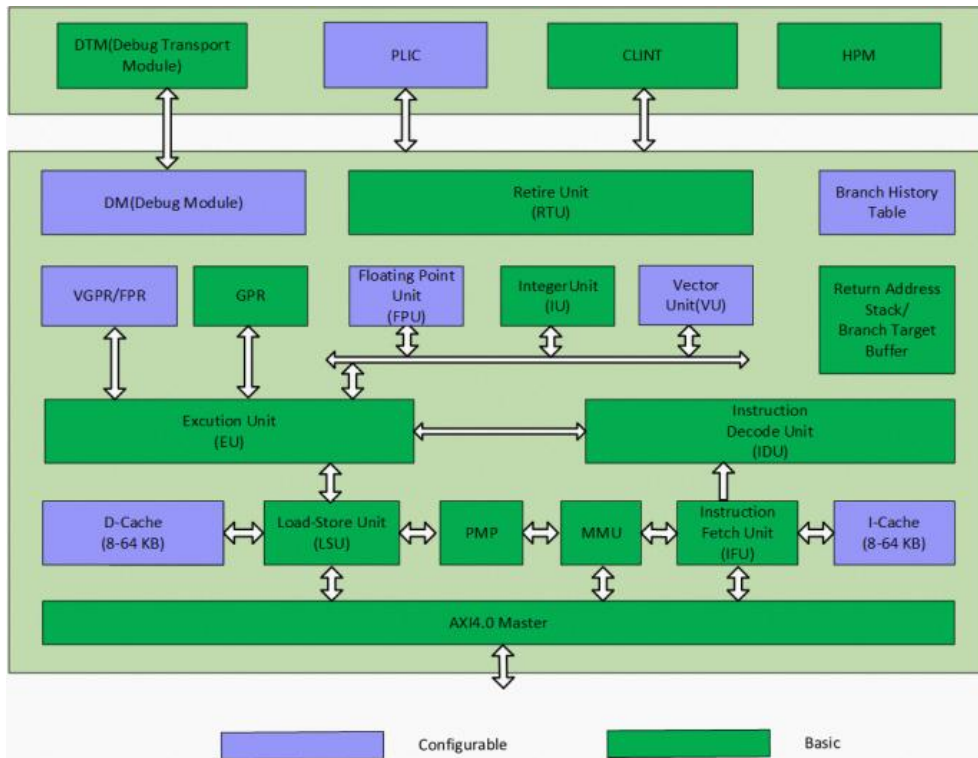
C906 处理器体系结构的主要特点如下：

特性	描述
架构	RV64GC
流水线	5级
玄铁扩展	玄铁指令扩展(XIE) 玄铁内存属性扩展(XMAF)
浮点单元	支持RISC-V Half、F、D指令扩展 支持IEEE 754-2008标准
矢量单元	支持RISC-V V扩展(可配置) 向量寄存器宽度128位 元素宽度8/16/32/64位 支持INT8/INT16/INT64/BFP16/FP16/FP32
总线接口	AXI4-128主接口
指令cache	32KB
数据cache	32KB
中断控制器	可配置的平台级中断控制器(PLIC)
MMU	Sv39地址转换
PMP	最多16个区域
调试	RISC-V调试



b) 处理器简介

C906 的结构框图如下所示：



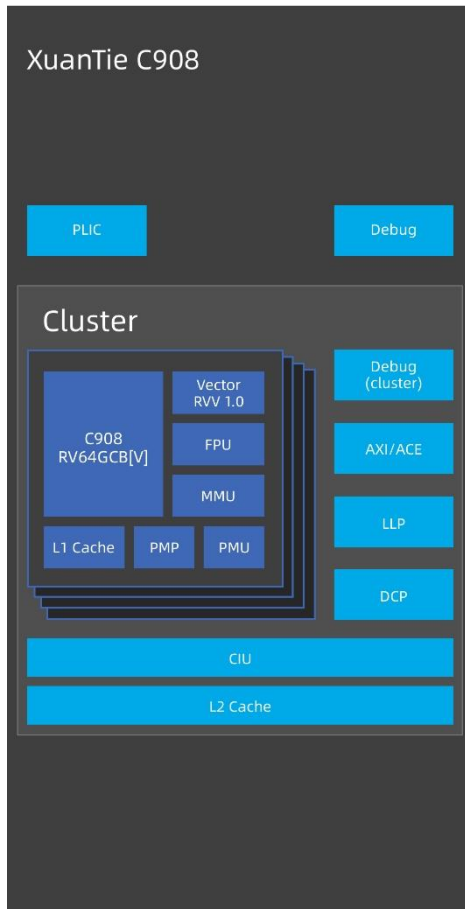
C906 核内子系统主要包含：指令提取单元（IFU）、指令译码单元（IDU）、整型执行单元（IU）、浮点单元（FPU）、可配的矢量执行单元（VPU）、存储载入单元（LSU）、指令退休单元（RTU）、虚拟内存管理单元（MMU）、物理内存保护单元（PMP）、主设备接口单元（AXI Master IF）等。

2) 兼容 64 位高效处理：C908

a) 概述

玄铁 C908 采用 9 级双发按序流水线，典型工作频率 > 2GHz，通过指令融合技术进一步提升流水线效率，实现了卓越的能效比。兼容 RVA22 标准，同时兼容 RISC-V 最新 Vector1.0 标准以进一步提升 AI 算力。可以应用在行业 IPC、智能交互、AR/VR、无线通讯等场景。

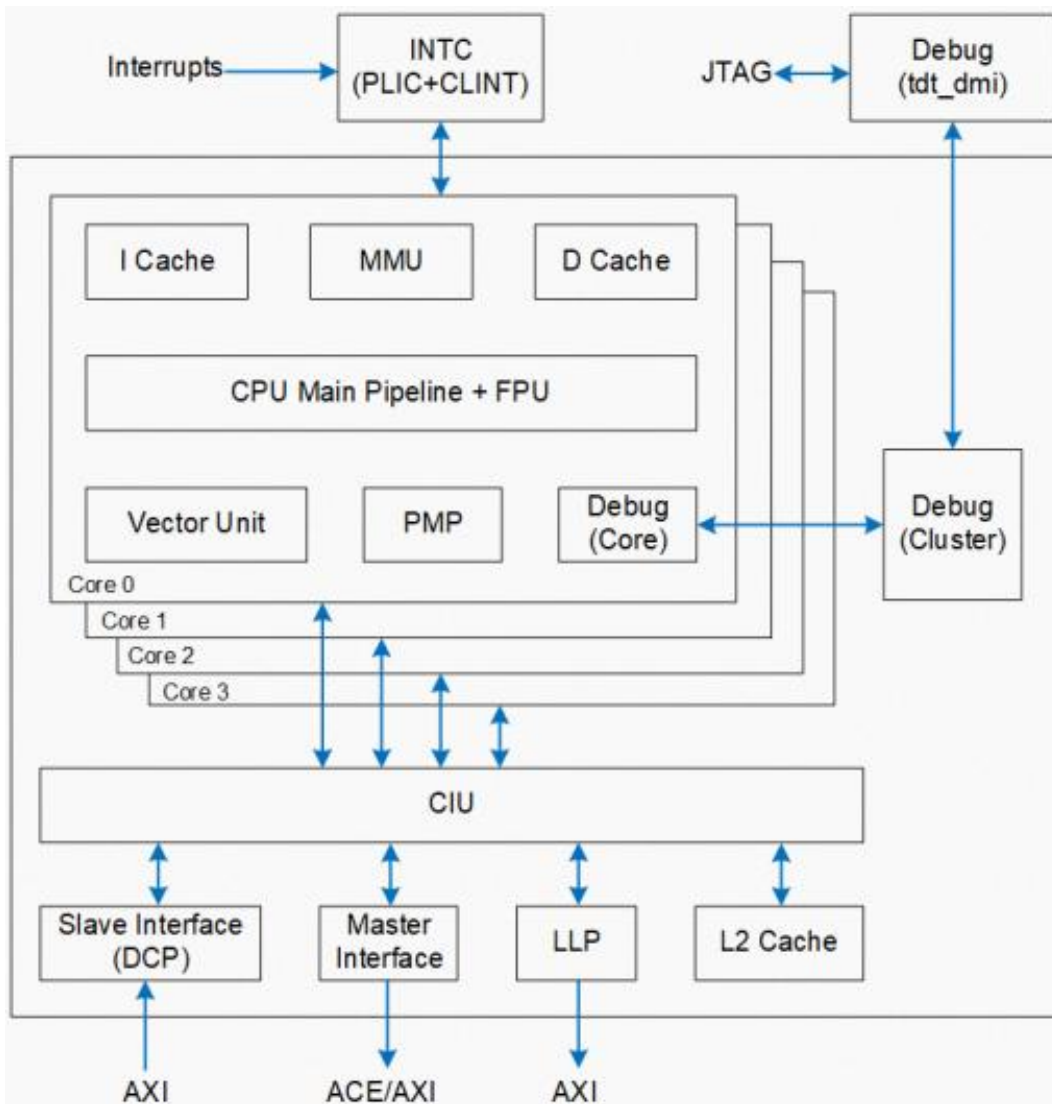
C908 核心的主要特点：



特性	描述
架构	RISC-V 64GCB[V]
SMP	同构多核，支持多Cluster，每个Cluster支持1~4个C908
寄存器	32个64位GPR，32个64位FGPR
流水线	9级双发按序流水线
玄铁扩展	玄铁指令扩展（XIE） 玄铁内存属性扩展（XMAE） 玄铁安全扩展（T-HEAD TEE）
浮点单元	支持BF16、FP16、FP32
主设备接口	支持AMBA 4.0 ACE 或 AXI 总线协议
低延时外设接口	支持AXI 4.0协议，128-bit
设备一致性接口	支持AXI 4.0协议，128-bit
指令Cache	16KB/32KB/64KB可选，支持奇偶校验
数据Cache	16KB/32KB/64KB可选，支持奇偶校验和ECC校验
L2 Cache	128KB/256KB/512KB/1MB/1.5MB/2MB/3MB/4MB，支持ECC校验
MMU	Sv39和Sv48内存管理
PMP	8/16/32/64表项可选，支持ePMP
中断控制器	支持私有中断控制器CLINT和共有中断控制器PLIC，支持多Cluster中断分发
矢量扩展	RISC-V Vector 1.0
矢量单元支持数据	浮点：FP16/BFP16/FP32 整型：INT8/INT16/INT32/INT64
矢量寄存器位宽	128/256可配置 VLEN
调试	RISC-V 调试框架，支持多核多 cluster 调试，支持劳特巴赫等第三方IDE/调试软件
性能监测单元	RISC-V标准性能监测单元
混合分支预测	分支跳转目标预测器、间接分支预测器、返回地址预测器、快速跳转目标预测器

b) 处理器简介

C908MP 结构框图如下图所示。



C908 核内子系统主要包含：指令提取单元（IFU）、指令执行单元（IEU）、矢量浮点执行单元（VFPU）、存储载入单元（LSU）、虚拟内存管理单元（MMU）和物理内存保护单元（PMP）。

C908 多核子系统包含：数据一致性接口单元（CIU）、二级高速缓存、主设备接口单元、可配置的 AXI4.0 设备一致性接口（DCP, Device Coherence Port）。

3) 高性能应用处理：C910

a) 概述

玄铁 C910 采用 12 级多发乱序流水线，典型工作频率 > 2.5GHz，是首款实现规模化量产的高性能乱序 RISC-V 处理器。采用 3 发射、8 执行的深度乱序执行架构，针对算术运算、内存访问以及多核同步等方面进行了增强。可以应用在对通用性能要求较高的高性能消费终端、边缘计算等场景。

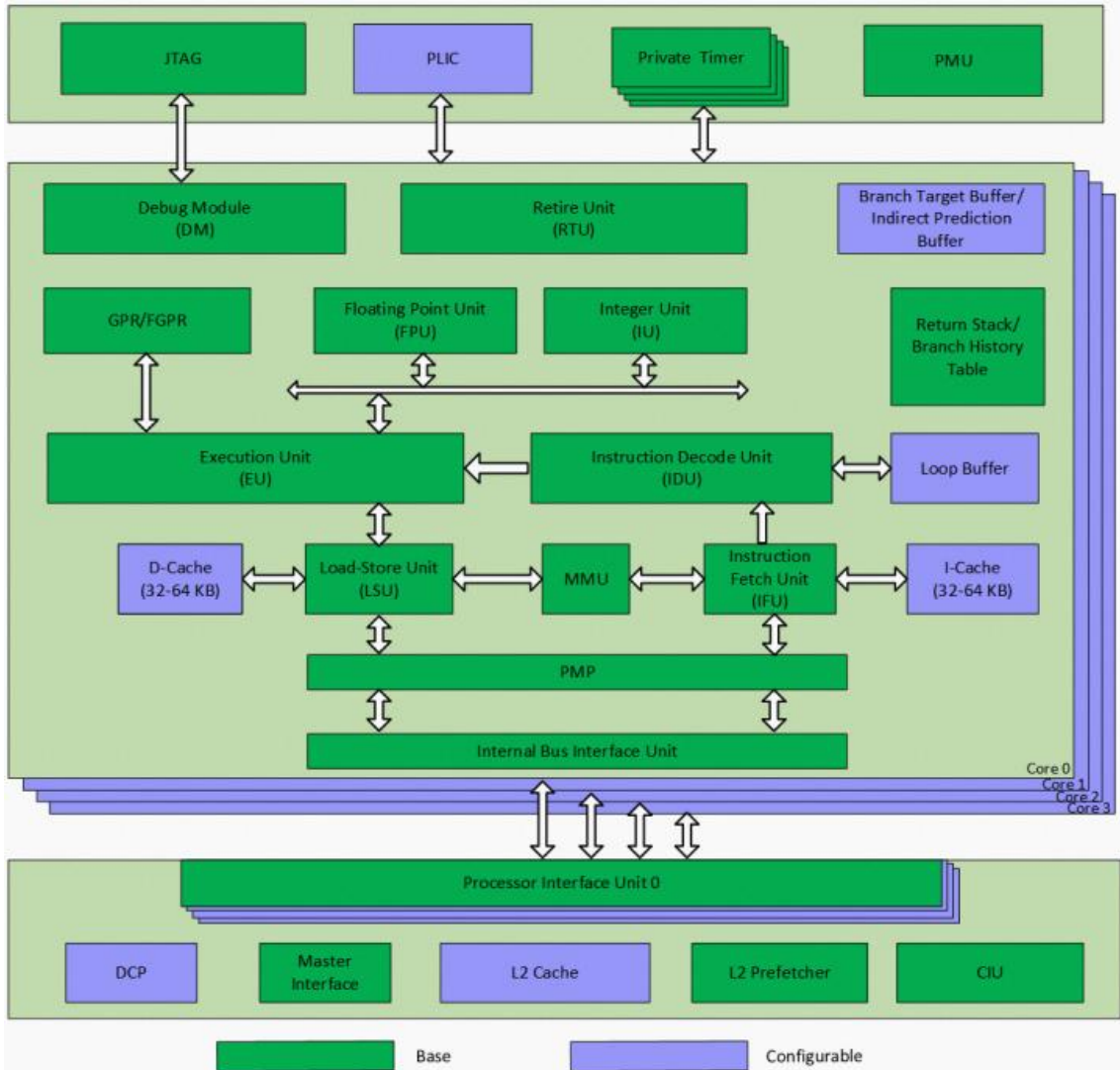
C910 核心的主要特点：



特性	描述
架构	RV64GC
SMP	同构多核，每个cluster最多4核
流水线	12 级
玄铁扩展	玄铁指令扩展 (XIE) 玄铁内存属性扩展 (XMAE)
浮点单元	支持单精度浮点运算 支持RISC-V F、D指令扩展 支持IEEE 754-2008标准
接口总数	AXI4-128 主接口
设备一致性接口	AXI4-128 从接口
指令cache	32KB/64KB可选，可选奇偶校验
数据 cache	32KB/64KB可选，可选ECC
L2 cache	256KB ~ 8MB可选，可选ECC
MMU	Sv39内存管理
PMP	0-16个保护区可选
中断控制器	可配置的平台级中断控制器 (PLIC)

b) 处理器简介

C910MP 结构框图如下图所示。



C910 核内子系统主要包含：指令提取单元（IFU）、指令译码单元（IDU）、整型执行单元（IU）、浮点单元（FPU）、存储载入单元（LSU）、指令退休单元（RTU）、虚拟内存管理单元（MMU）和物理内存保护单元（PMP）。

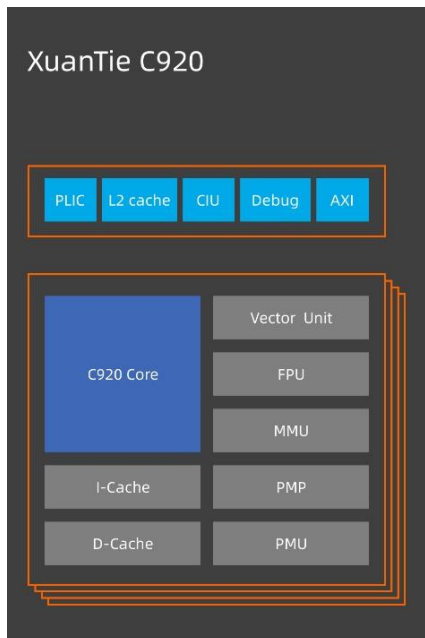
C910 多核子系统包含：数据一致性接口单元（CIU）、二级高速缓存、主设备接口单元、可配置的 AXI4.0 设备一致性接口（DCP，Device Coherence Port）、平台级中断控制器（PLIC）、计时器和自定义多核单端口调试框架。

4) AI 加速引擎：C920

a) 概述

玄铁 C920 采用 12 级多发乱序流水线，典型工作频率 > 2.5GHz，标配单精度浮点单元，并可进一步选配高性能乱序矢量运算单元。同时具备出色的访存能力，支持高性能数据预取技术。可以应用在有高并发算力要求的人工智能、自动驾驶等场景。

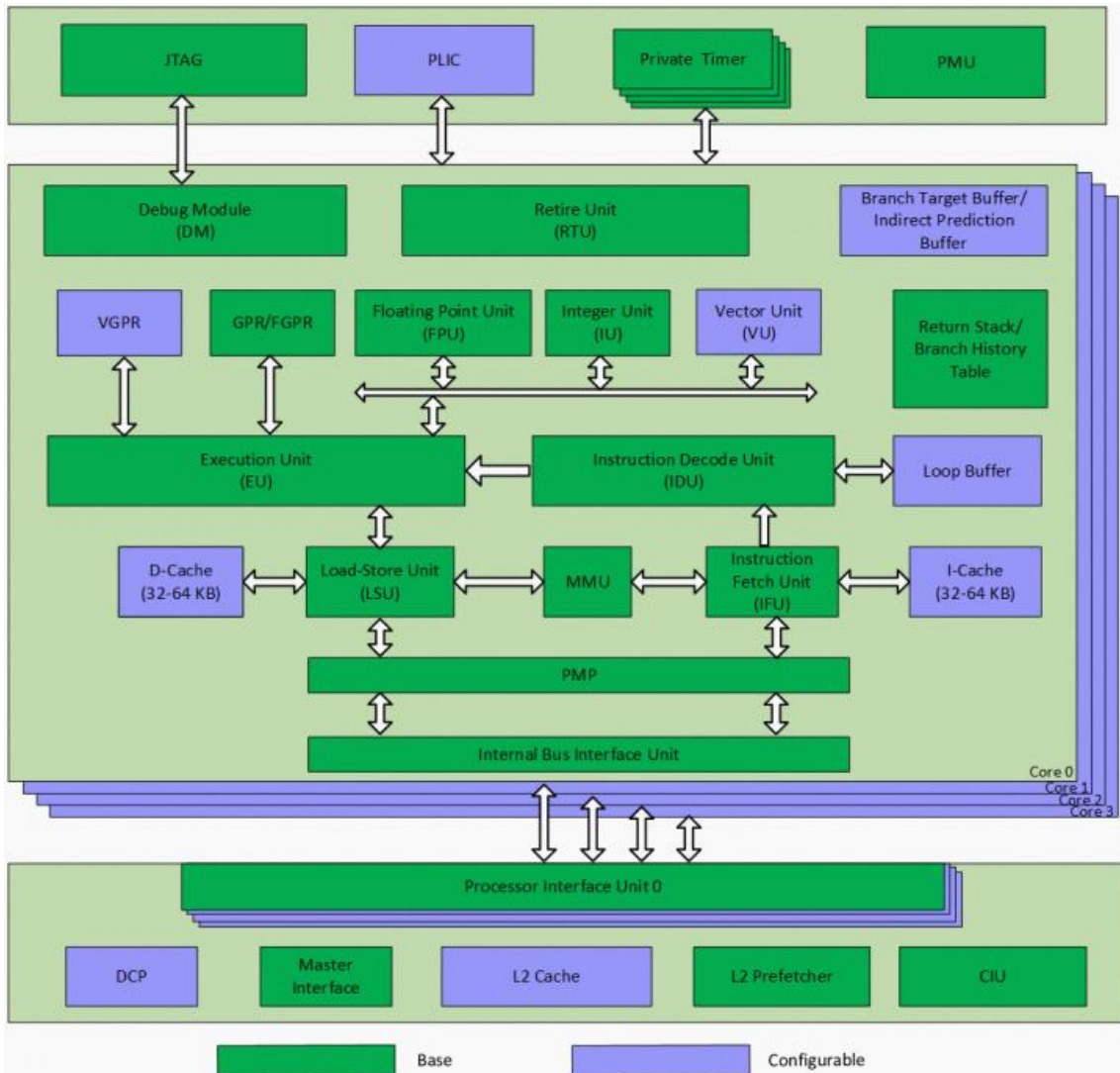
C920 核心的主要特点：



特性	描述
架构	RV64GCV
SMP	每个cluster最多4核
流水线	12级（整型）
玄铁扩展	玄铁指令扩展（XIE） 玄铁内存属性扩展（XMAE）
浮点单元	支持RISC-V F和D指令扩展 支持IEEE 754-2008标准
矢量单元	支持RISC-V V指令扩展 128位矢量寄存器 元素类型支持FP16/FP32/ INT8/INT16/INT32/INT64
总线接口	AXI4-128主接口
设备一致性接口	AXI4-128从接口
指令cache	最大64KB，可选奇偶校验
数据cache	最大64KB，可选ECC
L2 cache	最大8MB，可选ECC
MMU	5v39地址转换
PMP	最多16个区域
中断控制器	可配置的平台级中断控制器（PLIC）

b) 处理器简介

C920MP 结构框图如下图所示。



C920 核内子系统主要包含：指令提取单元（IFU）、指令译码单元（IDU）、整型执行单元（IU）、浮点单元（FPU）、矢量执行单元（VU）、存储载入单元（LSU）、指令退休单元（RTU）、虚拟内存管理单元（MMU）和物理内存保护单元（PMP）。

C920 多核子系统包含：数据一致性接口单元（CIU）、二级高速缓存、主设备接口单元、可配置的 AXI4.0 设备一致性接口（DCP，Device Coherence Port）、平台级中断控制器（PLIC）、计时器和自定义多核单端口调试框架。

4.面向高性能领域 CPU

1) 高性能应用处理：C910

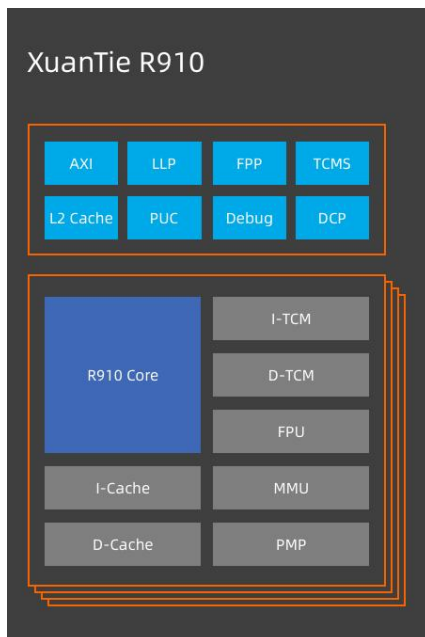
玄铁 C910 采用 12 级多发乱序流水线，典型工作频率 > 2.5GHz，是首款实现规模化量产的高性能乱序 RISC-V 处理器。采用 3 发射、8 执行的深度乱序执行架构，针对算术运算、内存访问以及多核同步等方面进行了增强。可以应用在对通用性能要求较高的高性能消费终端、边缘计算等场景。

2) 可靠实时增强：R910

a) 概述

玄铁 R910 用 12 级多发乱序流水线，典型工作频率 > 2.5GHz，同时支持 Cache 以及 TCM 存储架构，各级片上存储支持校验纠错以提升可靠性，可进一步选配快速外设接口以及一致性外设接口，从而大幅提升系统实时性。可以应用在对实时性及算力有高要求的企业级 SSD，网络通信等场景。

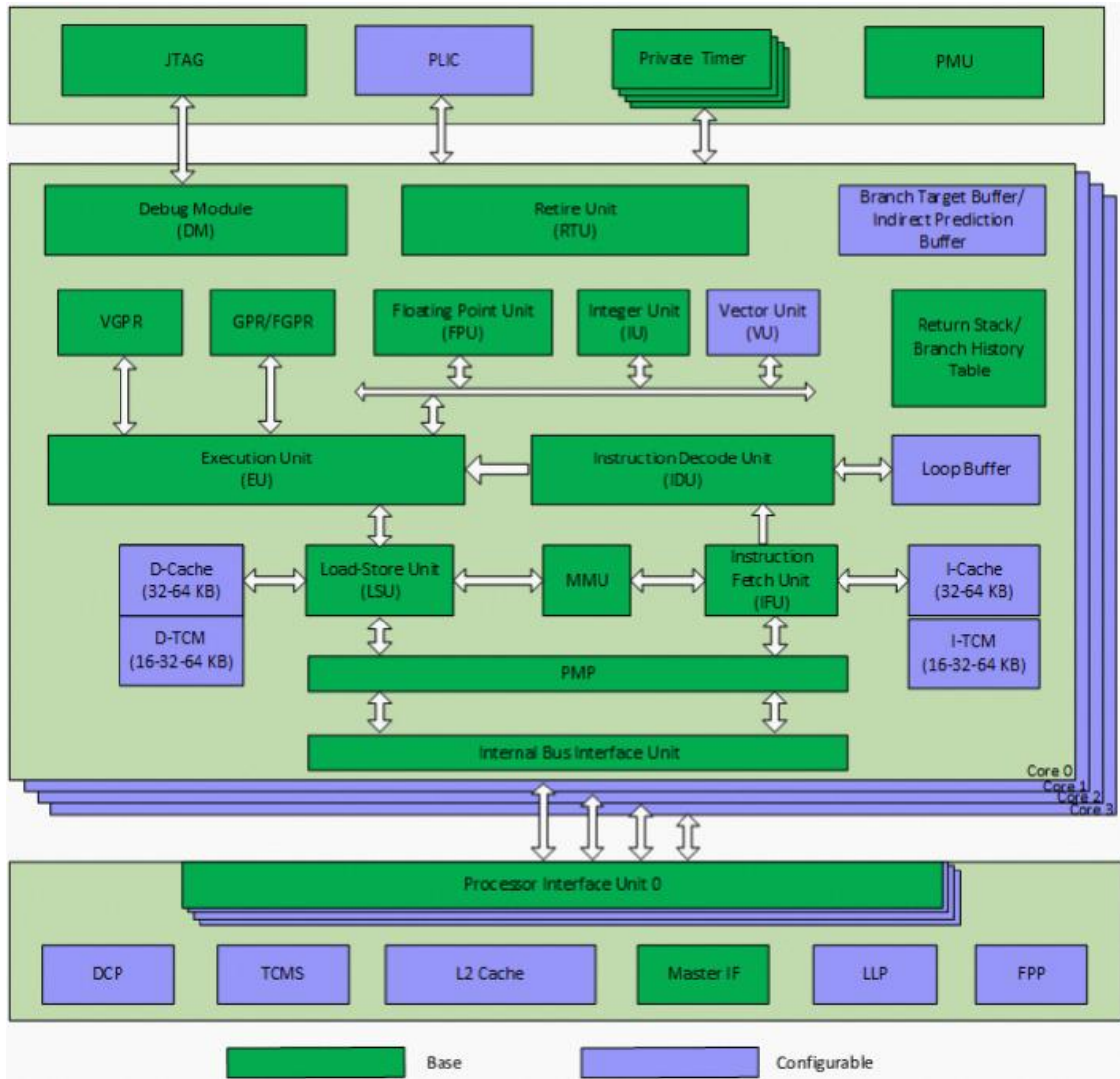
R910 核心主要特点：



特性	描述
架构	RV64GC
SMP	每个cluster最多4核
流水线	9~12级
玄铁扩展	玄铁指令扩展 (XIE) 玄铁内存属性扩展 (XMAE)
浮点单元	支持RISC-VHalf、F、D指令扩展 支持IEEE754-2008标准
TCM	16~64KB可配指令紧耦合内存 16~64KB可配数据紧耦合内存 可选ECC
Cache	最大64KB指令高速缓存 最大64KB数据高速缓存 最大8MB二级高速缓存 可选ECC
内存管理	Sv39MMU, MPU
总线接口	AXI4.0Main接口用于作为主Master发起访问; -AXI4.0LLP接口用于作为Master快速发起外设访问; -APB4.0FPP接口用于作为Master快速发起外设访问; -AXI4.0TCMS接口用于作为Slave接收外部对TCM的访问; AXI4.0Slave接口用于提供Device/Accelerator与Core的 datacoherence

b) 处理器简介

R910MP 结构框图如下所示。



R910 核内子系统主要包含：指令提取单元（IFU）、指令译码单元（IDU）、整型执行单元（IU）、浮点单元（FPU）、存储载入单元（LSU）、指令退休单元（RTU）、虚拟内存管理单元（MMU）和物理内存保护单元（PMP）。

R910 多核子系统包含：数据一致性接口单元（CIU）、二级高速缓存、主设备接口单元、可配置的快速外设访问接口（LLP）、可配置的 APB 主设备接口（FPP）、可配置的紧耦合内存访问接口（TCMSP）、可配置的 AXI4.0 设备一致性接口（DCP, Device Coherence Port）、平台级中断控制器（PLIC）、计时器和自定义多核单端口调试框架。

5.玄铁 CPU 课程学习

平头哥芯片开放社区官网上线了 RISC-V 系列课程培训，该课程和清华大学、大连理工大学等合作，包括由教育部评选为“国家级一流本科课程”的《嵌入式软件设计》，成为国内嵌入式教育的标杆课程。在“中国大学 MOOC”平台已开设 9 轮课程，累计选课人数达四万人次以上，广受学生欢迎。通过 XuanTie RISC-V Institute，可以学习该课程。

欢迎广大开发者[登录 OCC](#) 了解学习。有关玄铁 CPU 详细介绍内容，也可以登录[平头哥芯片开放社区 \(OCC\)](#) 查看。

XuanTie RISC-V Institute

玄铁RISC-V处理器

[课程目录](#) [课程要点](#) [课程问答](#)

内容简介

玄铁系列处理器IP，是智能、安全、端云一体芯片架构的基石，为数字化时代提供计算核心。玄铁CPU广泛应用于计算视觉、数据存储、工业互联、网络通信、智能家居、生物识别、信息安全等领域。在2021年云栖大会平头哥宣布开源玄铁RISC-V系列处理器（E902, E906, C906, C910），并开放系列工具及系统软件。这是玄铁系列处理器与基础软件的全球首次全栈开源，涵盖了从终端到云的各种场景，是边缘和云的智能安全集成芯片架构的基石。截至目前，平头哥已拥有从低功耗、低成本到高性能、高效率的丰富RISC-V处理器产品家族，广泛应用于边缘计算、无线通讯、工业控制、通用MCU等30多个领域及应用场景。玄铁处理器已成为国内RISC-V领域影响力和市场占有率最大的处理器产品系列。

主讲教师：赖晓晨，软件学院，大连理工大学

本章节主要从指令集架构，编程模型，存储系统等维度，介绍玄铁E906 CPU，帮助用户全面了解并学习玄铁E906 CPU。

课程资料

1. 玄铁E906处理器简介
2. E906的编程模型
3. E906的指令集
4. E906的异常和中断系统
5. E906的存储系统
6. E906的总线矩阵和总线接口
7. E906的中断系统
8. E906的流水线

无剑平台

1.无剑 100 开源 SoC 平台

1) 概述

[无剑 100 开源 SoC 平台](#)是围绕玄铁 RISC-V E902 打造的软硬一体的开源 MCU 芯片平台。可用于面向 IoT 设备接入，音视频播放及智慧家居、环境监测、健康医疗等多种应用场景芯片的开发。平台包含芯片开发，剑池 CDK 开发工具，支持 YoC/RTOS 操作系统，同时支持通过 EDA 工具进行前端仿真和制作 FPGA 测试，帮助用户聚焦芯片差异化定制，主打低功耗和安全特性。

2) CH2601 芯片

CH2601 是业内首款基于无剑 100 开源 SoC 设计的通用型 MCU。该芯片基于玄铁 E906 内核，最高主频 220MHz，支持 AliOS Things, YoC 及 RTOS 操作系统及平头哥剑池开发工具 (CDK)。同时含有丰富的存储资源支持多媒体播放，并提供 CSI 驱动接口，让开发更便捷。可广泛应用于 IoT 设备接入，智慧家居，工业控制及健康医疗等场景。

芯片框图如下：



2.无剑 600 SoC 平台

1) 无剑 600 SoC 平台概述

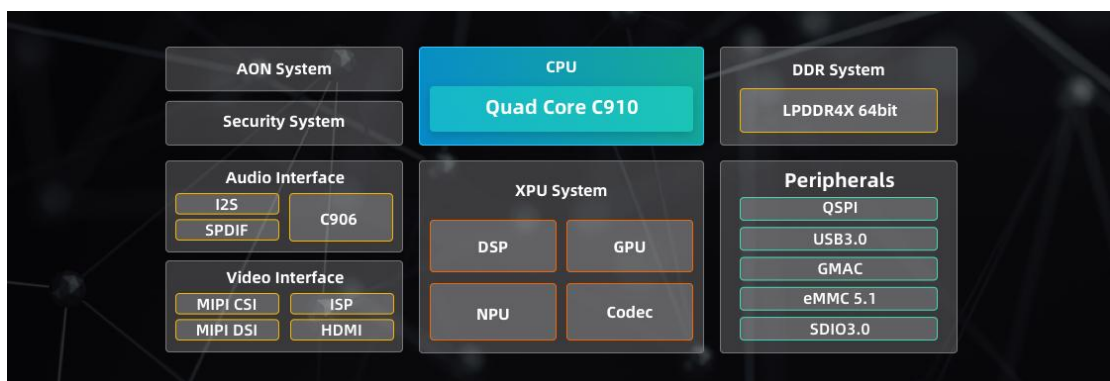
[无剑 600](#) RISC-V 芯片设计平台是平头哥推出的高性能异构芯片设计和软硬件全栈的平台。具有高性能、高内存带宽、异构计算和人工智能加持的特性。兼顾高安全、多模态感知和软硬一体的能力。客户可面向场景定制开发芯片，缩短产品研发周期，降低开发难度。同时，助力更多的芯片资源接入到 RISC-V 生态，推动下游应用。



2) 曳影 1520 概述

曳影 1520 是首颗基于无剑 600 平台研发的量产多模态 AI 处理器 SoC 原型，采用高性能玄铁 RISC-V CPU，支持全链路安全防护，具备全数据通路性能均衡的特点。

该原型芯片已通过平头哥 5000+项硬件测试（涵盖：电气特性/高低温/高低压/可靠性等），30000+项软件测试（涵盖：功能/性能/兼容性/功耗/压力测试等），从硬件到软件均已完成了量产应用场景的适配。开发者在等待定制化芯片的同时，可以先在曳影 1520 芯片上开发自己的系统，极大地缩短最终产品量产的时间。



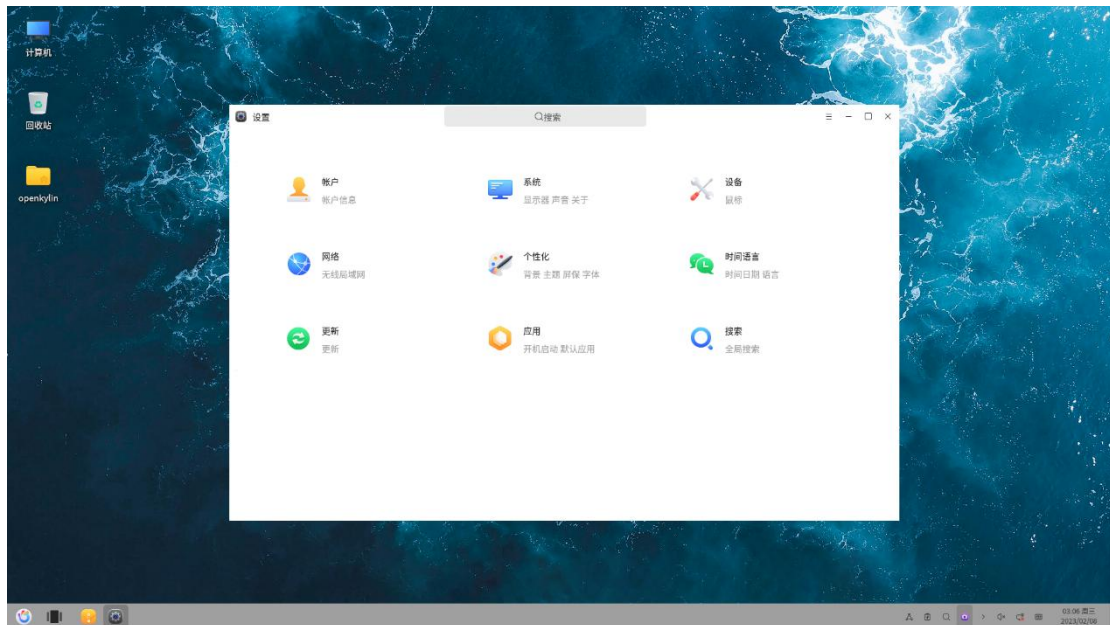
3) 电影 1520 适配不同 OS 平台

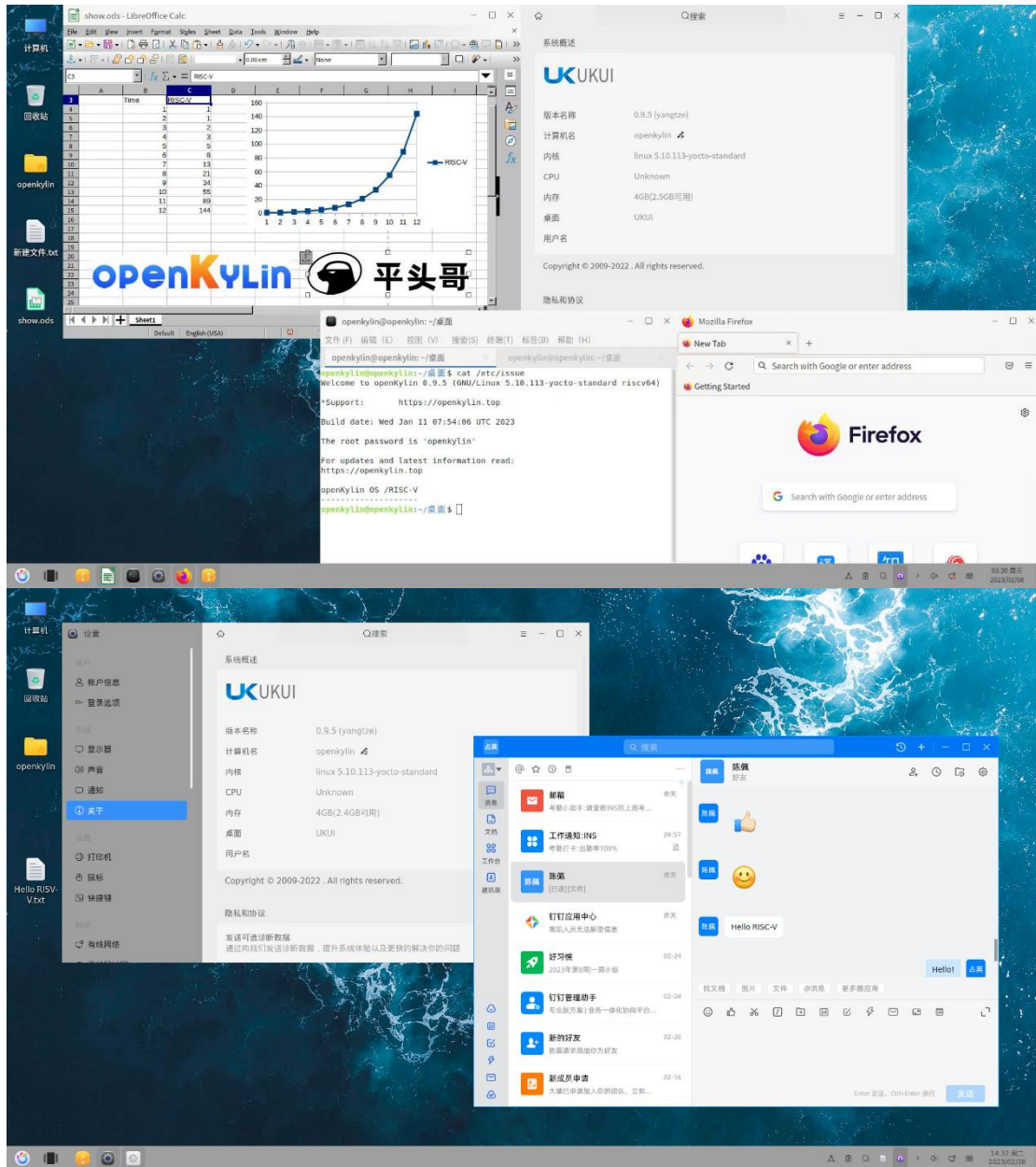
a) openKylin 社区

openKylin（开放麒麟）社区旨在以“共创”为核心，在开源、自愿、平等、协作的基础上，通过开源、开放的方式与企业构建合作伙伴生态体系，共同打造桌面操作系统顶级社区，推动 Linux 开源技术及其软硬件生态繁荣发展。

目前，平头哥已与 openKylin 社区合作，完成了电影 1520 与 openKylin 操作系统的适配工作，并基于社区软件源构建了镜像版本。

该版本包含近 1600 个软件包，集成了一系列稳定版本的基础库和图形开发库，能够顺畅运行 UKUI 桌面环境。此外，系统包含丰富的 openKylin 自研软件和第三方开源软件，如浏览器、视频、音频和文档编辑等，可以满足用户的基本使用需求。





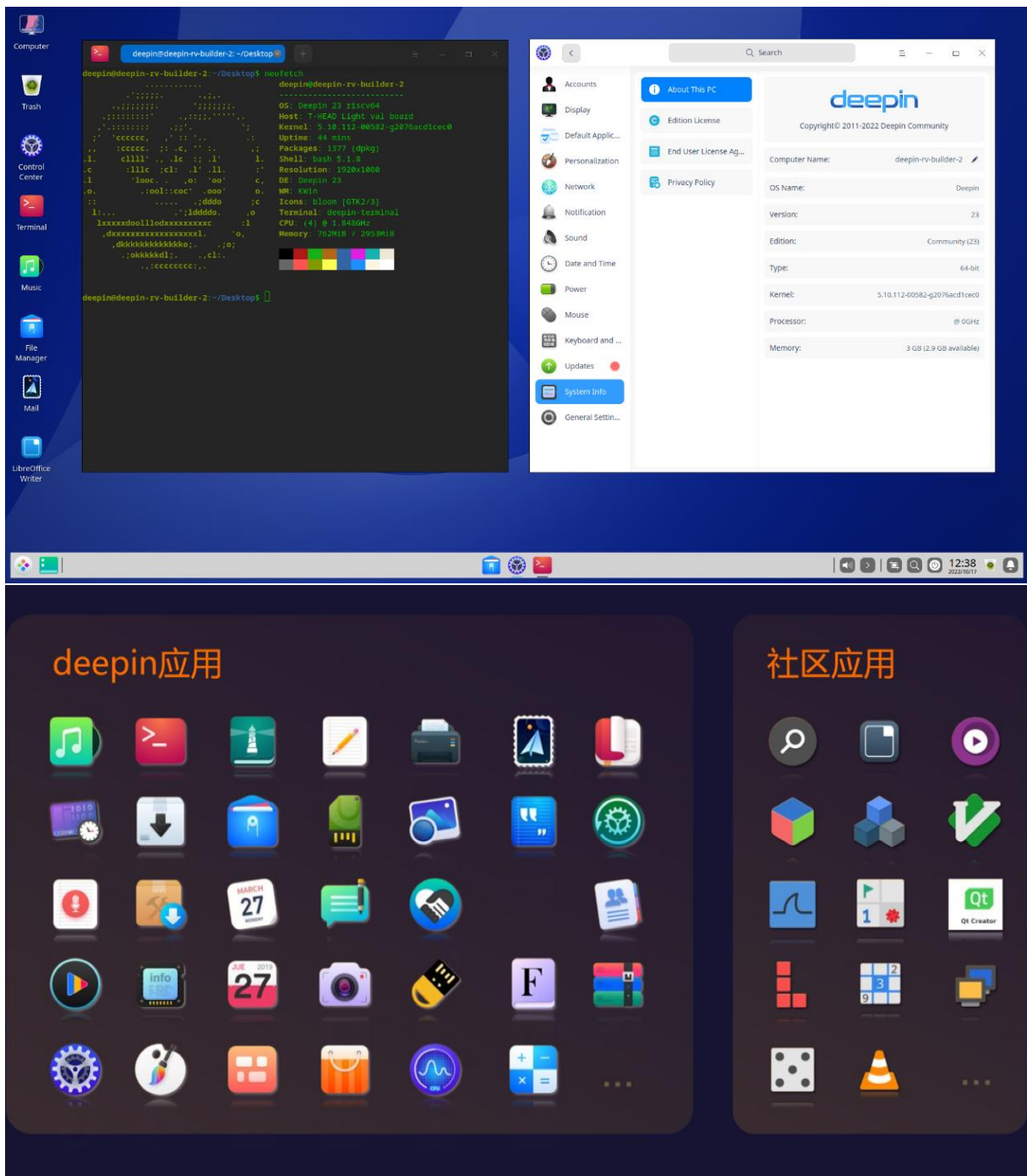
在 openKylin 操作系统的支持下，平头哥玄铁处理器完成与钉钉的适配。该合作完成了对钉钉 2.7 万个文件、17 个第三方库的编译，同时突破了 Qt（应用程序开发框架）、CEF（ChromiumEmbeddedFramework，Chromium 嵌入式框架）等关键软件框架的兼容性问题，使得钉钉能够顺利在 RISC-V 平稳运行。同时，钉钉成为首个 RISC-V 商用 IM 产品。

b) 统信

统信软件是国产操作系统的代表厂商，也是龙蜥社区副理事长单位。旗下拥有桌面、服务器、企业级应用、智能终端等四大产品线，统信桌面操作系统面向不同人群提供社区

版 (deepin)、家庭版、专业版等细分版本, 统信软件是龙蜥社区的主要贡献者, 在 2021 年 5 月全国首家发行了基于龙蜥社区技术路线的商业服务器操作系统版本。目前统信操作系统国内装机量累积超过 300 万台。

2022 年, 统信软件与龙蜥社区、平头哥展开合作, 成功在曳影 1520 启动了 deepin V23 操作系统。流畅运行桌面、办公、影音及开发等关键大型软件, 探索 RISC-V 在桌面端的潜力。

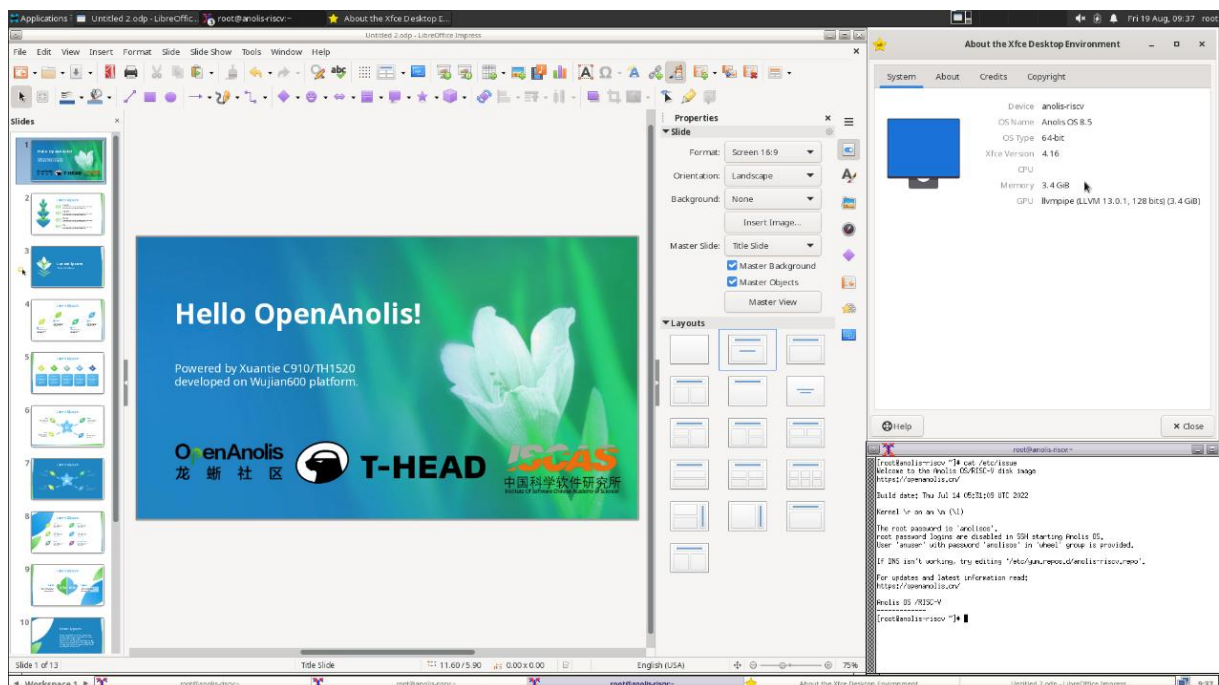


a) 龙蜥社区

龙蜥社区成立于 2020 年 9 月，是一个面向国际的操作系统开源社区及创新平台，致力于通过开放的社区合作，构建 Linux 开源发行版及开源创新技术，推动软、硬件及应用生态繁荣发展，共创数字化发展开源新基建。

社区理事会成员是由阿里云、统信软件、Arm、Intel 等 21 家国内外头部企业共同组成，有近 300 家合作伙伴参与生态共建，实现了主流芯片协同研发机制全覆盖、主流中间件/数据库全覆盖、主流整机 OEM 厂商全覆盖；100 余家产品与龙蜥操作系统 (AnolisOS) 完成适配。

在无剑 600 平台上，平头哥与龙蜥社区、中科院软件所 PLCT 实验室进行了软硬件全栈的联合优化，完成了 RISC-V 与龙蜥操作系统的 3000 多个基础包适配，并在曳影 1520 上首次运行 Firefox 浏览器、LibreOffice 等大型桌面级软件，以及 Hexo 和 OpenRocket 等基于 NodeJS 和 JAVA 的应用，极大拓展了 RISC-V 的想象力。



Time	RISC-V
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55
11	89
12	144

ISCAS 中国科学院软件研究所
T-HEAD 龙蜥社区
OpenAnolis

System About Credits Copyright

Device: andrisrv
OS Name: Anolis OS 8.5
OS Type: 64-bit
Xfce Version: 4.16
CPU:
Memory: 3.4 GB
GPU: llvmpipe LLVM 13.0.1, 128 bits (3.4 GB)

```
root@andrisrv:~# cat /etc/issue
Welcome to the World of RISC-V disk image
https://openanolis.org/

Build date: Thu Jul 14 05:31:08 UTC 2022

Kernel: v on an rv (1)

The root password is 'rootroot'.
root password login are disabled in SSH starting RHEL 85.
User 'anous' with password 'anolis' in 'wheel' group is provided.
If you aren't working, try exiting 'telnet/aurous@openanolis.org'.
For updates and latest information read:
https://openanolis.org/

RHEL: 85 /RISC-V
root@andrisrv:~#
```

Hello OpenAnolis!

Powered by Xuantie910/TH1520
developed on Wujian600 platform

T-HEAD OpenAnolis ISCAS
龙蜥社区 中国科学院软件研究所

System About Credits Copyright

Device: andrisrv
OS Name: Anolis OS 8.5
OS Type: 64-bit
Xfce Version: 4.16
CPU:
Memory: 3.4 GB
GPU: llvmpipe LLVM 13.0.1, 128 bits (3.4 GB)

```
root@andrisrv:~# cat /etc/issue
Welcome to the World of RISC-V disk image
https://openanolis.org/

Build date: Thu Jul 14 05:31:08 UTC 2022

Kernel: v on an rv (1)

The root password is 'rootroot'.
root password login are disabled in SSH starting RHEL 85.
User 'anous' with password 'anolis' in 'wheel' group is provided.
If you aren't working, try exiting 'telnet/aurous@openanolis.org'.
For updates and latest information read:
https://openanolis.org/

RHEL: 85 /RISC-V
root@andrisrv:~#
```

平头哥 RISC-V 工具链

1. RISC-V 工具链简介

1) 概述

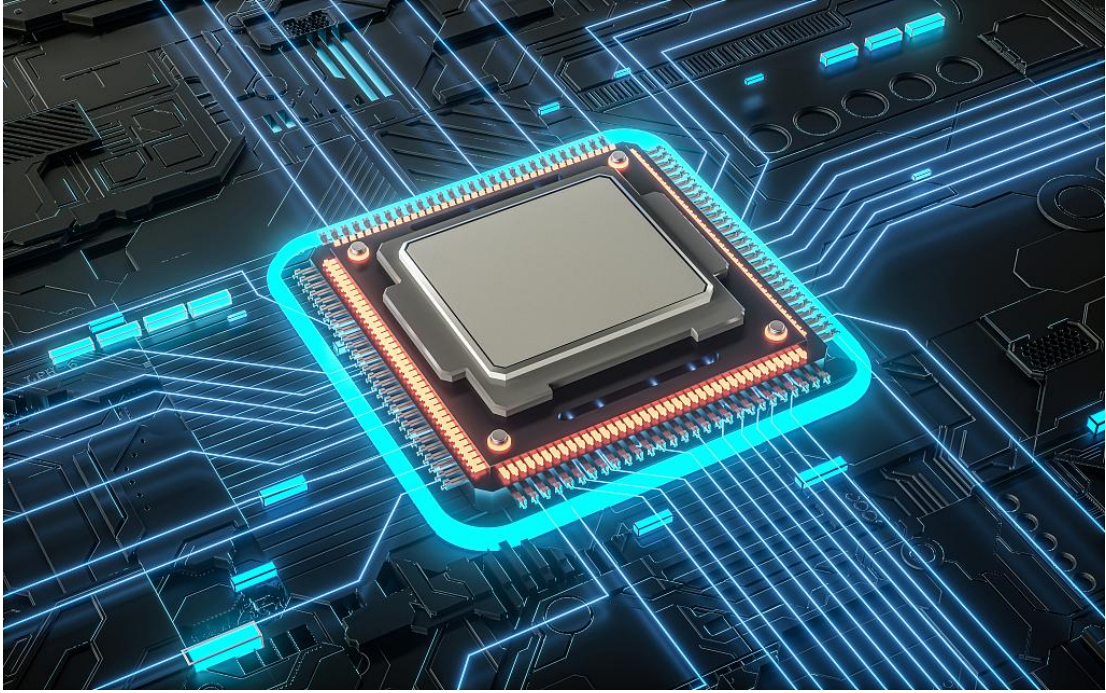
“工具链”是英文“toolchain”的翻译，它指辅助开发者完成程序开发、调试、性能分析调优等开发优化行为的一系列工具集合。它一般包含编译器、汇编器、链接器、调试器、模拟器等基础工具，集成开发环境作为一种图形化的综合开发工具也可以被纳入其中。由于指令集和用户编程模型不同，因此每个架构都需要一套完整的、独有的工具链。

2) 工具链的重要性

在计算机发展的早期，几乎没有工具链，科学家通过在程序卡片上打孔来表示不同指令的编码，计算机读取程序卡片的打孔信息执行指令流。随着制造工艺和计算机的发展，现代计算机的指令码存储于 ROM、RAM、Flash 等介质中，把指令码下载到指定的存储空间需要经过复杂的控制流程，因此必须需要借助于程序下载器才能完成下载。

为了提升程序开发效率，涌现了众多的高级开发语言。目前 99% 以上的开发者都使用高级语言完成业务的开发，然后使用编译器把高级语言翻译成目标机器指令下载到机器中执行。因此编译器成了工具链中一个必不可少的组件。

对于现代计算机系统，一套稳定、完整的开发工具链是一个架构的基础，所有软件的运行都依赖工具链；没有工具链，CPU 和硬件无法使用，开发者寸步难行。



工具链的关键指标可分为高效性和易用性两个方面。高效性又包括编译高效性、调试高效性、性能分析高效性等，其中编译的高效性尤为重要。

通过上文介绍，编译器是用于把高级语言翻译成目标机器指令的软件；一段高级语言会被翻译成若干条机器指令，编译器在保证功能正确的前提下，翻译出的机器指令数量和质量也非常关键，直接影响程序运行所需花费的时间和程序占用空间，即影响了处理器的性能和代码密度表现。

因此，处理器性能是硬件+工具链的综合能力体现，单比较硬件频率指标没有实际意义。另外调试高效性直接影响了开发者分析解决问题效率，也是决定了处理器能否开发者接受、是否可以被广泛应用。

3) RISC-V 工具链的现状

既然工具链这么重要，那么 RISC-V 的工具链现状如何，能否满足当前的应用需求？由于 RISC-V 的开放特性，加上基金会对软件生态的持续运作，吸引了大量的开发者投入到工具链的补充和完善工作上。目前从编译工具（GCC+LLVM）、调试工具到集成开发环境，在基本功能上都已经支持，但在高效性和易用性上仍需要继续加强和提升。如编译得到的程序代码大小相对 ARM 仍然存在不少差距；缺少性能调优工具；缺少图形化

实时操作系统调试工具；开源的集成开发环境只支持简单的调试插件等，这些方面都需要工具链开发者不断地进行完善，并通过应用场景的验证和打磨。

4) 剑池系列工具链

剑池系列工具链以开源工具链为基础，根据应用场景和玄铁处理器微架构高度优化的开发工具集合（见下图）。配合玄铁 Turbo 指令集，在 XT906 上相对开源版本平均有 13% 的性能提升，在 XT910 上平均有 20% 的性能提升。开发了基于指令 Trace 的性能分析工具，配合集成开发环境的图形化界面，为开发者提供快速性能 profiling 和优化的能力。

另外，开发了两套集成开发环境，分别面向 MCU 应用开发和通用应用开发，降低现有工程迁移到 RISC-V 架构芯片上的难度。关于剑池系列工具链的每个工具，在后续的文章中会做详细介绍。



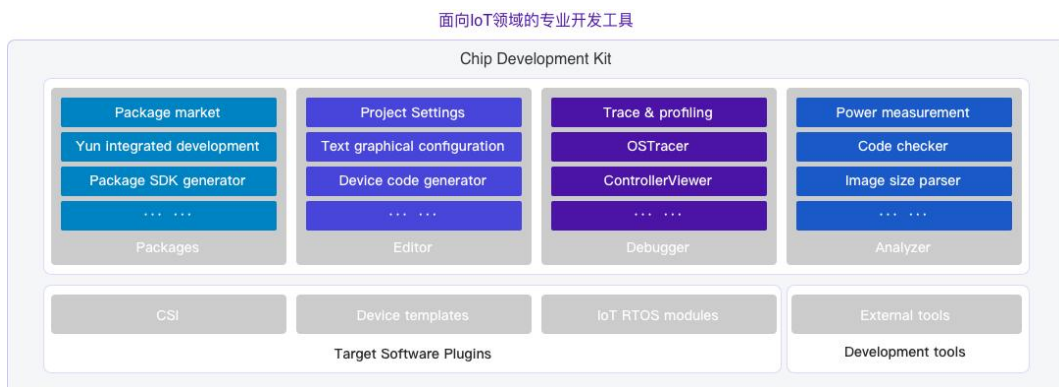
工具链是一个体系架构的重要且不可缺失的组成部分，它是基础软件的基础。RISC-V 工具链经过几年的发展，在功能上已经日渐完善；相信接下来的几年里，经过开源社区开发者和商业公司的共同努力，在编译调试高效性和易用性方面会有明显的提升，RISC-V 的芯片也会被更多的开发者接受，被应用到各种场景。

2. 剑池 CDK 开发工具

1) 概述

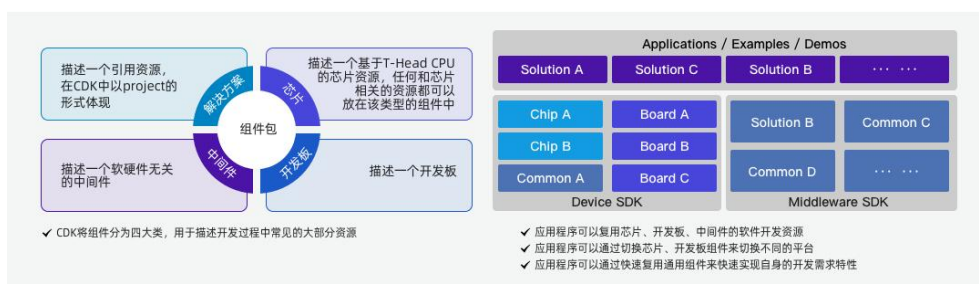
[剑池 CDK](#) 是平头哥面向 MCU、工控、IoT 等领域应用提供的专业集成开发环境。它以极简开发为理念，在不改变传统 MCU 应用开发习惯的基础上，全面接入云端开发资源，并结合图形化的 OSTRacer、Profiling 等调试分析工具，加速用户产品开发。

软件框图如下所示：

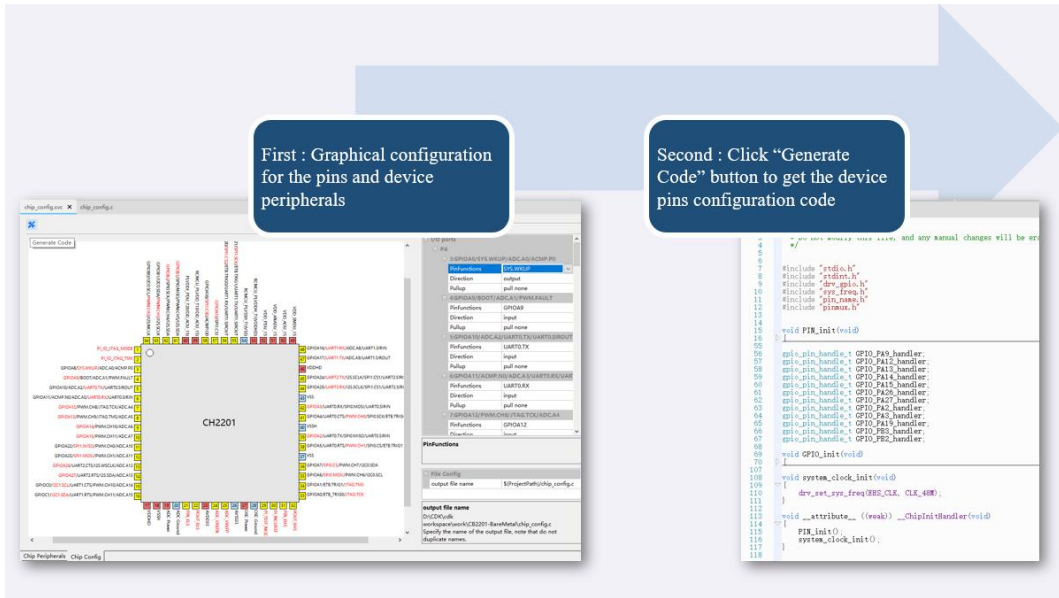


从功能划分角度分析，剑池 CDK 工具分为以下四部分：

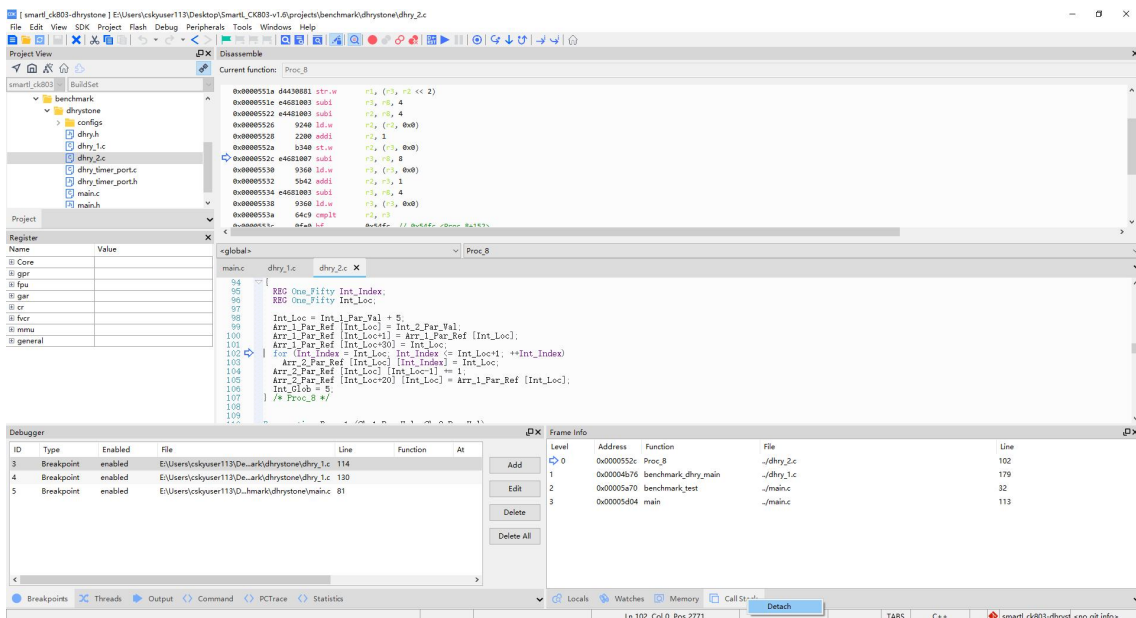
- Packages：组件化模块。用于支撑剑池 CDK 的组件化开发，为开发者提供一个制作松耦合的软件 SDK 的工具。



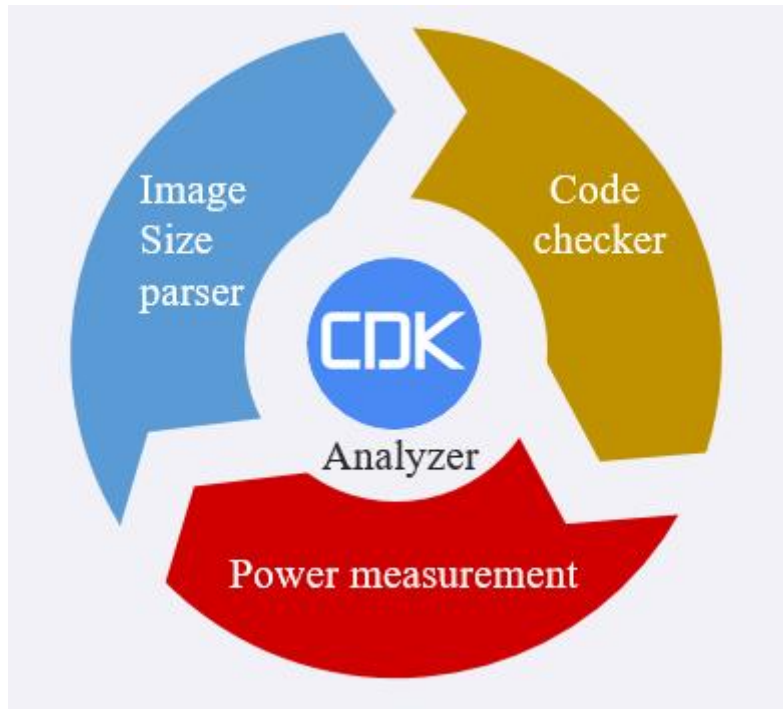
- Editor: 编辑器模块。图形化的方式解决嵌入式开发中晦涩、难懂的文本文档和代码编辑。



- Debugger: 调试器模块。图形化的方式提供芯片调试的查看和控制界面。



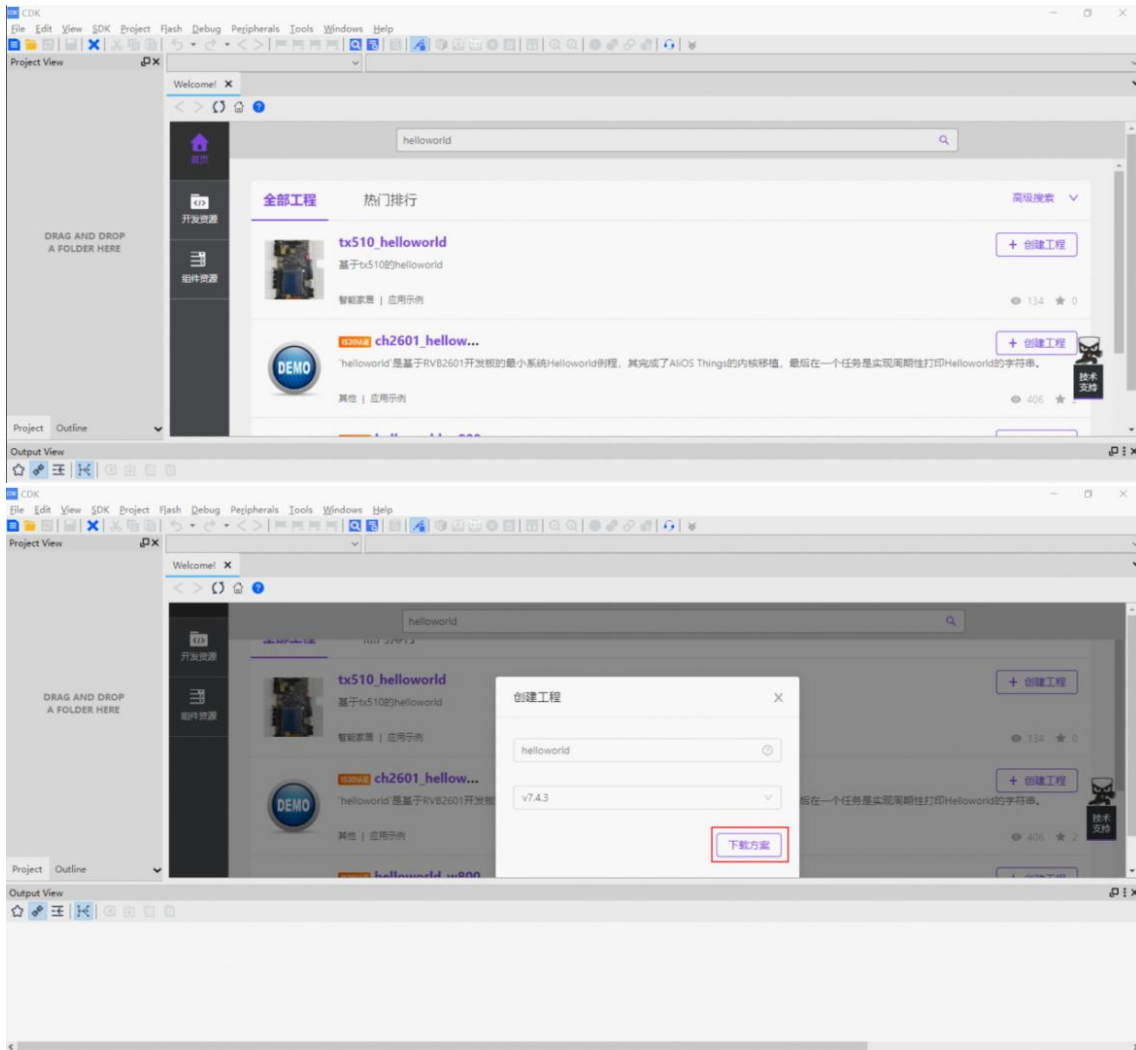
- Analyzer: 分析器模块。为开发者开发出更高效的嵌入式程序而提供的工具。



2) 工程创建

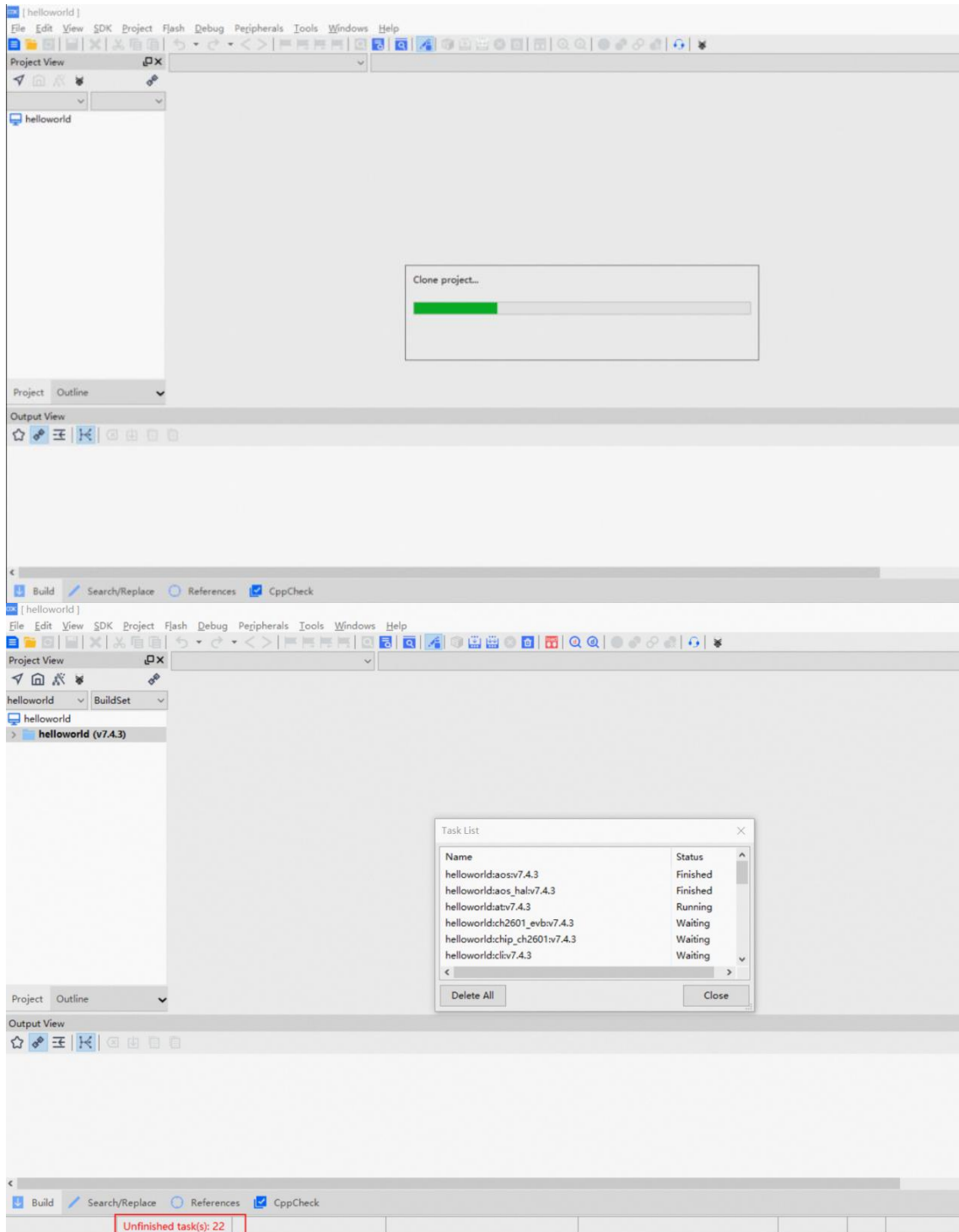
点击下图平头哥图标进入创建工程主页。



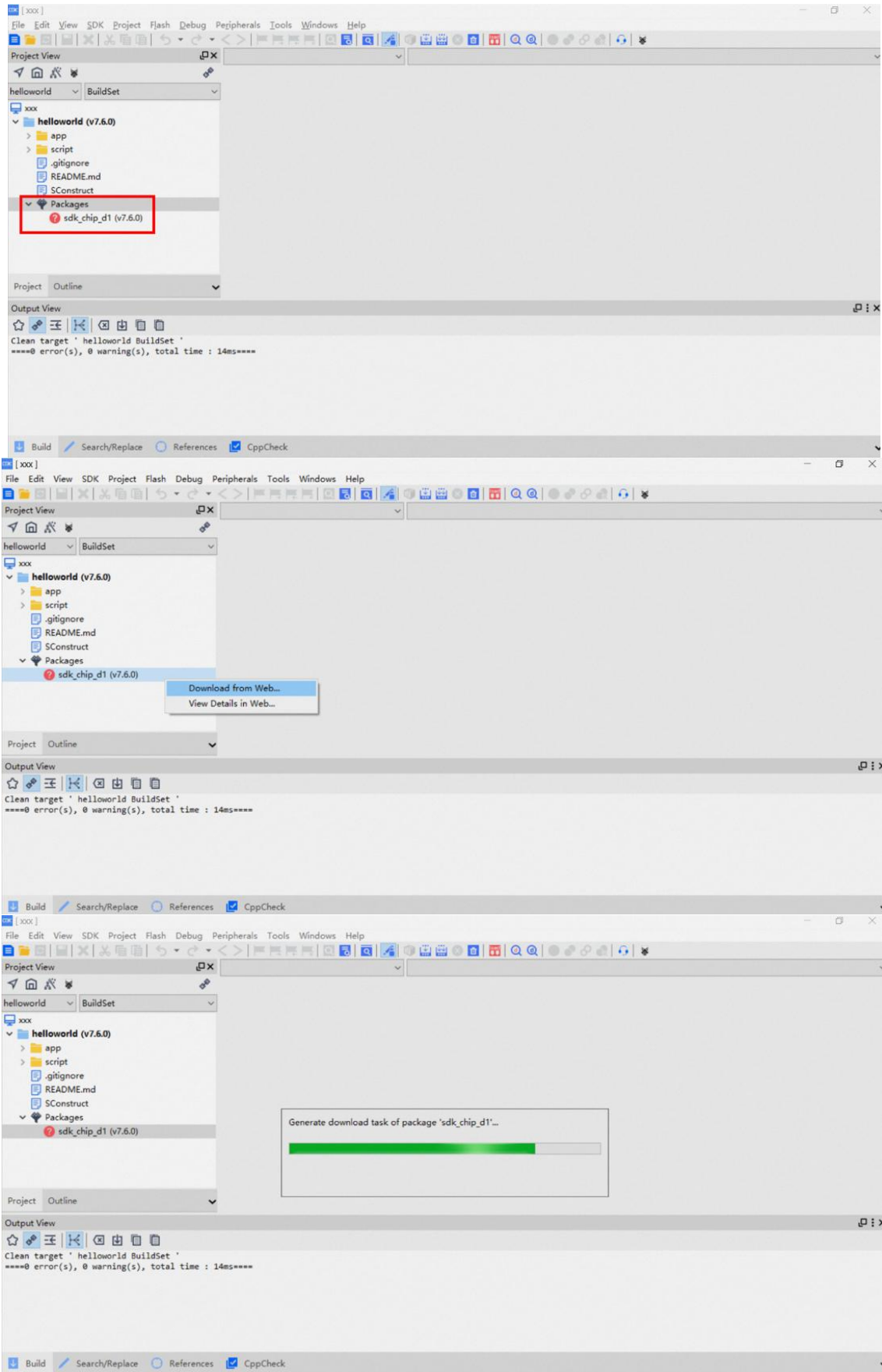


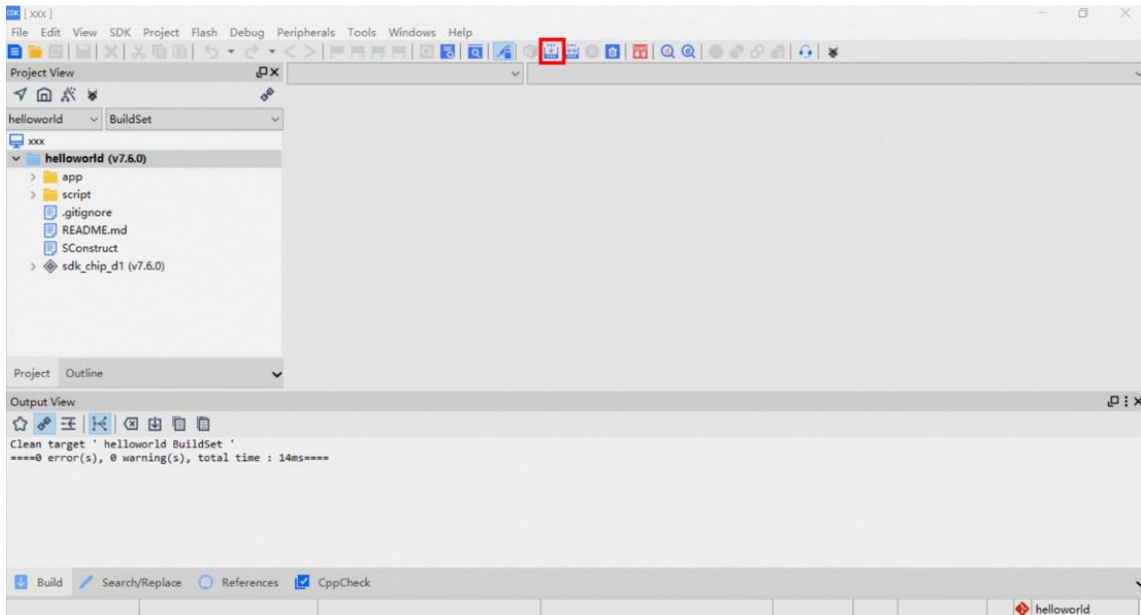
3) 工程编译

第一步：点击下载方案，下载过程需要花个几分钟，视网络情况而定。待下载完成之后，就可以开始编译烧录了。



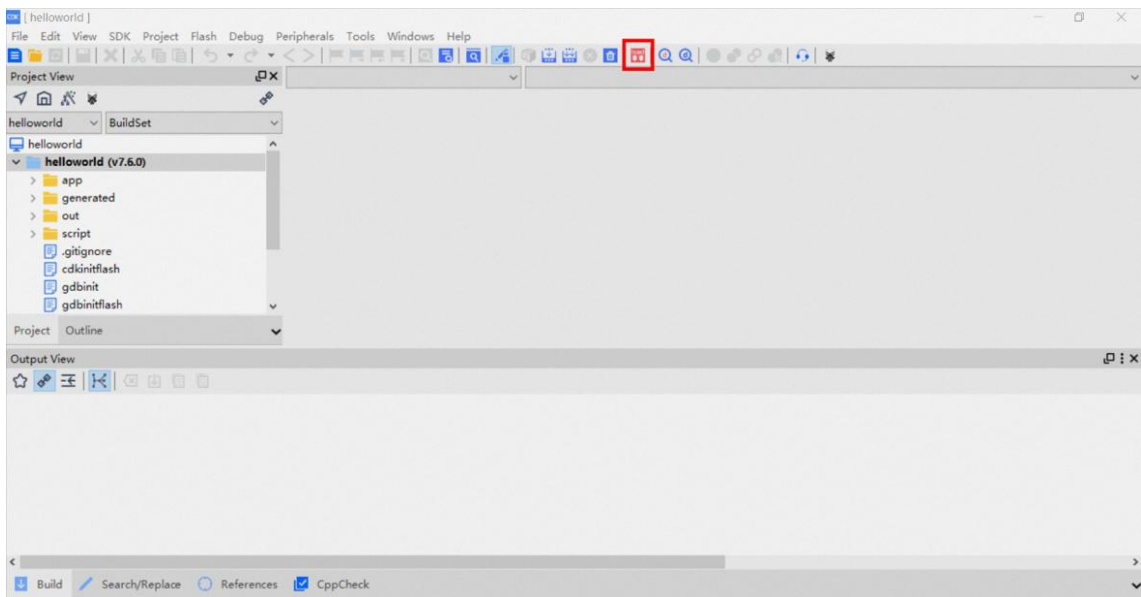
第二步：如果这个示例支持多个开发板的话，那么可以通过右键点击红色问号部分进行 SDK 的切换。待下载完成之后，就可以开始编译烧录了。





4) 工程烧录

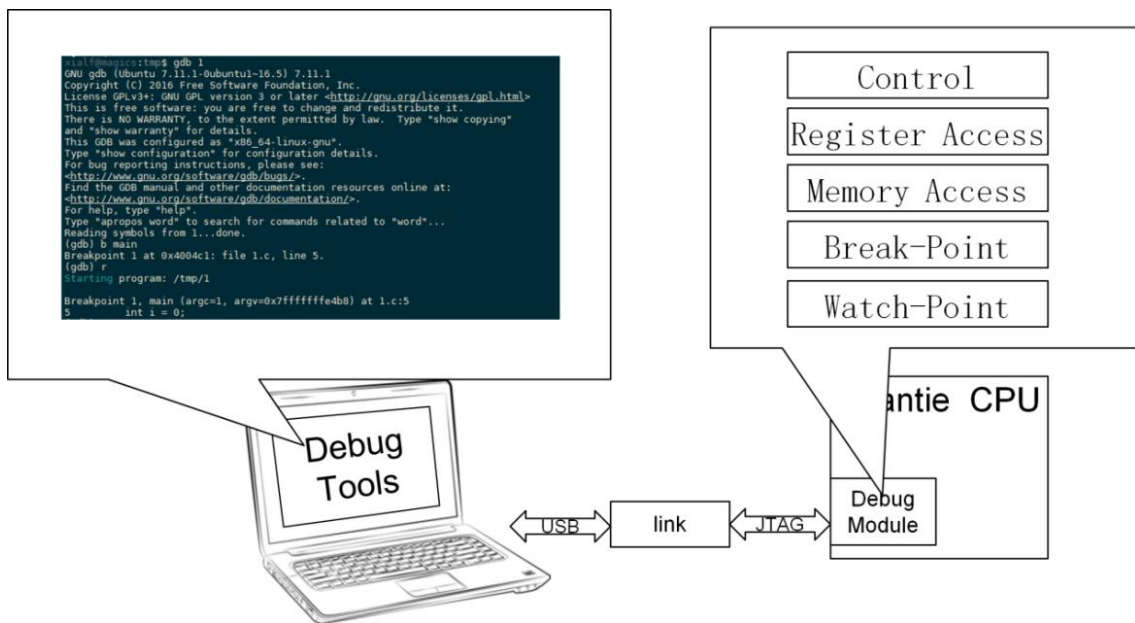
编译完成之后就可以烧录到对应的开发板了。烧录之前请确保开发板硬件连接正确。



3. 玄铁 CPU 调试系统

1) 调试系统基本介绍

调试系统是一套基于硬件调试模块，结合软件调试工具，为开发者提供解决软硬件 bug 的技术系统。软件调试部分包括调试工具、link 及其固件。硬件调试部分包括调试模块 (Debug Module)。



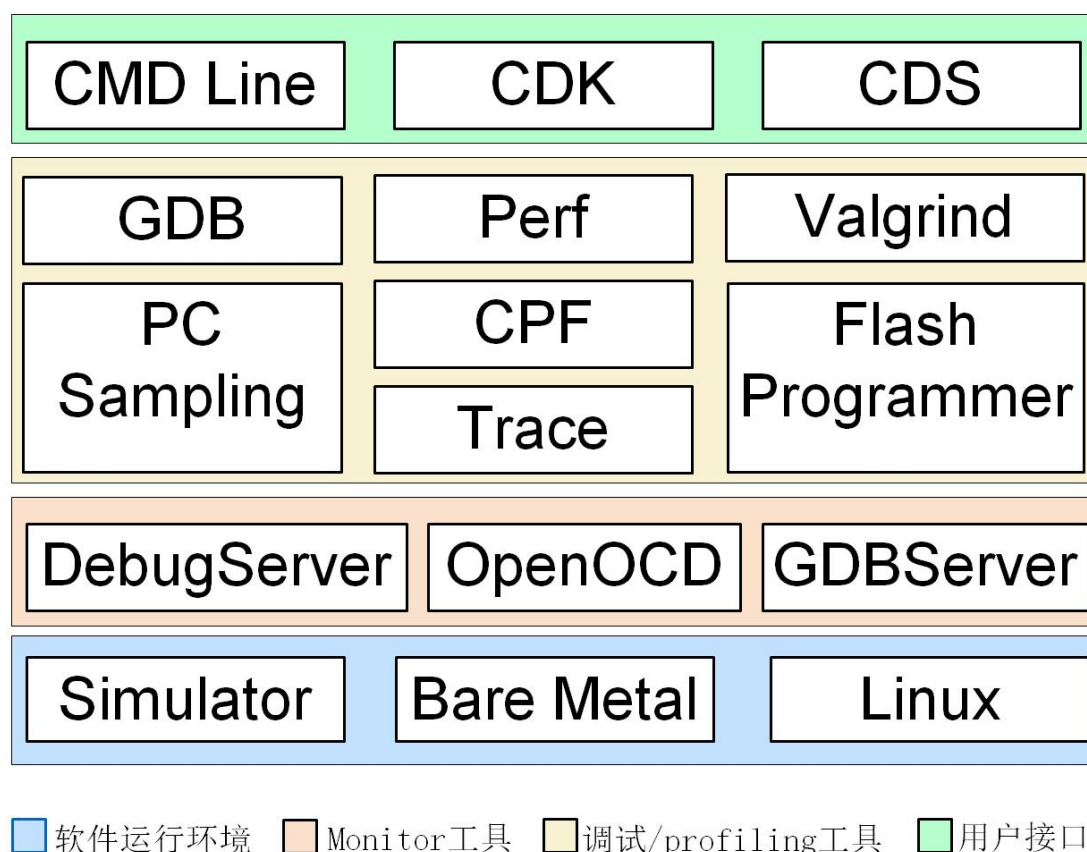
例如上图所示，在典型的 MCU 调试系统中：

调试工具 (Debug Tools) 是面向开发者调试的软件工具，一般运行在开发人员的主机、服务器等场景下。常见的调试工具有 GDB、IDE 等，如 T-Head 的 RISC-V 工具链中的 gdb, CDK, DebugServer 等工具。开发者可使用这类工具进行软件程序的调试、分析工作。

由于调试主机无法直接与 CPU 上的调试单元连接，因此需要 link 作为中间转换工具，使得调试工具的操作能够真实地作用在 CPU 上，比如平头哥的 CkLink, Segger 的 JLink 等等。

CPU 上的调试模块 (Debug Module) 是支持大部分调试手段的基础, 实现对 CPU 的控制和读写寄存器、读写内存等功能。

2) 玄铁 CPU 调试系统总览



如上图所示, 玄铁 CPU 调试系统主要由以下几部分组成:

- 软件运行环境, 指软件运行的实际环境, 包括模拟器、普通硬件环境 (Bare Metal)、基于 Linux 内核的环境;
- Debug Monitor, 主要指配合调试工具, 对被调试程序进行调试控制的工具, 包括 DebugServer/OpenOCD/GDBServer 等。对于实际开发板来说, 往往还需要在线仿真调试工具 (cable/link), 如 CkLink, JLink 等;
- 调试/Profiling 工具, 是开发者经常使用的用于调试程序的工具, 主要包括 GDB、perf、CPF、Trace、Flash Programmer 等;

- 用户接口，是用户直接接触到的操作界面，主要是命令行工具（CMD Line）、剑池 CDK、剑池 CDS 等。其中大部分调试/profiling 工具大部分都支持命令行方式呈现。剑池 CDK、剑池 CDS 相比命令行工具，有友好的界面，具有更好的开发体验。

玄铁 CPU 配套有诸多调试工具，本章节将分别介绍几个典型调试工具，如下表所示：

调试工具	描述
DebugServer	是一种 Debug Monitor 软件,用于和玄铁 CPU 的调试模块进行交互,完成调试操作。
GDB	开源 GDB 软件,支持玄铁 CPU 体系结构,包括玄铁 8 系列、9 系列 CPU
剑池 CDK	平头哥推出的玄铁 CPU 集成开发环境
Perf	Perf 是 Linux 上强大的性能分析工具集合。结合 PMU 事件可对程序热点采样, cycle、指令数统计等。
CPF	T-Head 研发的性能分析工具,能真实反映 CPU 的运行轨迹(目前仅在 T-Head qemu 上支持)

DebugServer

DebugServer 是平头哥自行设计实现的调试工具。其目的是配合 GDB 完成对 CPU 的软件调试,其中包含一套 GDB Remote Serial Protocol 的实现,并解析该协议,通过 USB 操作 link 完成协议的执行。同时 DebugServer 也实现了一套 CLI,完成一些简单的调试功能,如复位、指令单步,读写寄存器、读写内存等。

```
+---+
| C-Sky Debugger Server (Build: Feb 26 2021) |
| User Layer Version : 5.7.01 |
| Target Layer version : 2.0 |
| Copyright (C) 2021 Hangzhou C-SKY Microsystems co.,ltd |
+---+
T-HEAD: CKLink_Lite_V2, App_ver 2.23, Bit_ver null, Clock 2526.316KHz,
5-wire, Without DDC, Cache Flush On.
+-- CPU 0 --+
RISCV CPU Info:
  WORD[0]: 0x0814050d
  WORD[1]: 0x11001000
  WORD[2]: 0x202bfb7b
  MISA : 0x40909125
Target Chip Info:
  CPU Type is E907FP, Endian=Little, Version is R1S0P1.
  DCache size is 32K.
  ICache size is 32K.
  MGU zone num is 256.
  MGU zone size is 1KB.
  HWBKPT number is 3, HWWP number is 3.
  MISA: (RV32IMAFDXP, Imp M-mode, U-mode)

GDB connect command for CPU 0:
  target remote 127.0.0.1:1025
  target remote 172.16.150.243:1025

***** DebuggerServer Commands List *****

setclk
  Set the JTAG Clock(100KHz~24MHz), default unit is MHz, you can use KHz.
singlestep/si
  Execute single-step in the target.
sreset
  Soft reset the target and halt immediately, example: sreset -c 0xabcd1234.
nreset
  NReset command support:
  nreset, do NReset to reset the target.
  nreset halt, halt the target after reset.
reset
  Reset command support:
  reset, reset the target(if not RVDM debug, reset == nreset).
  reset halt, halt the target after reset.
pctrace
  Show the PCFIFO(8 <= length <= 4096, default 8).
print/p
  Print command support:
  print /x[d/f/o] *memory, eg: p /x *0x20000000.
  print /x[d/f/o] $registers, eg: p /x $hsr, eg: p $r1.
  print target, print target info.
  print cpu, print current cpu number.
set
  Set command support:
  set *memory=value, eg:set *0x80000=0x1234.
  set $reg=value, eg:set $hr=0x1234, eg:set $r1=0x1234.
  set resume-bkpt-exception on/off (should be only used by gdb monitor command).
  set mem-access progbuf/abscmd/sysbus (just for riscv dm debug).
  set cpu=value, select current cpu to value.
quit/q
  Quit Debugger Server.
help/h
  Show help informations.
CTRL+B ENTER
  Switch input channel.
*****
```

GDB

GDB 是一个软件调试工具，是 GNU 组织维护的开源软件：

<https://www.gnu.org/software/gdb/>。

GDB 支持不同的后端体系架构，目前 GDB 的发布的源码版本中已经支持玄铁 8 系列、9 系列 CPU。能够满足玄铁 CPU 的日常开发需求。

通过一段简单示例看一下 GDB 使用。

```
GNU gdb (T-HEAD RISC-V Tools V1.10.5 B20210228) 8.2.50.20190202-git
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=riscv64-unknown-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from out/smrtl_e907_evb.elf...
(gdb) target rem 127.0.0.1:1025
Remote debugging using 127.0.0.1:1025
benchmark_dhry_main () at /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/dhry_1.c:111
111     Next_Ptr_Glob = (Rec_Pointer) malloc (sizeof (Rec_Type));
(gdb) b benchmark_dhry_main
Breakpoint 1 at 0x542e: file /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/dhry_1.c, line 111.
(gdb) monitor reset
(gdb) c
Continuing.

Breakpoint 1, benchmark_dhry_main () at /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/dhry_1.c:111
111     Next_Ptr_Glob = (Rec_Pointer) malloc (sizeof (Rec_Type));
(gdb) b __exit
Breakpoint 2 at 0x542e: file /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/../../../../csi_driver/smrtl_rv32/startup.S$
line 129.
(gdb) c
Continuing.

Breakpoint 2, __exit () at /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/../../../../csi_driver/smrtl_r
v32/startup.S:129
129     }
    __exit
(gdb) q
A debugging session is active.

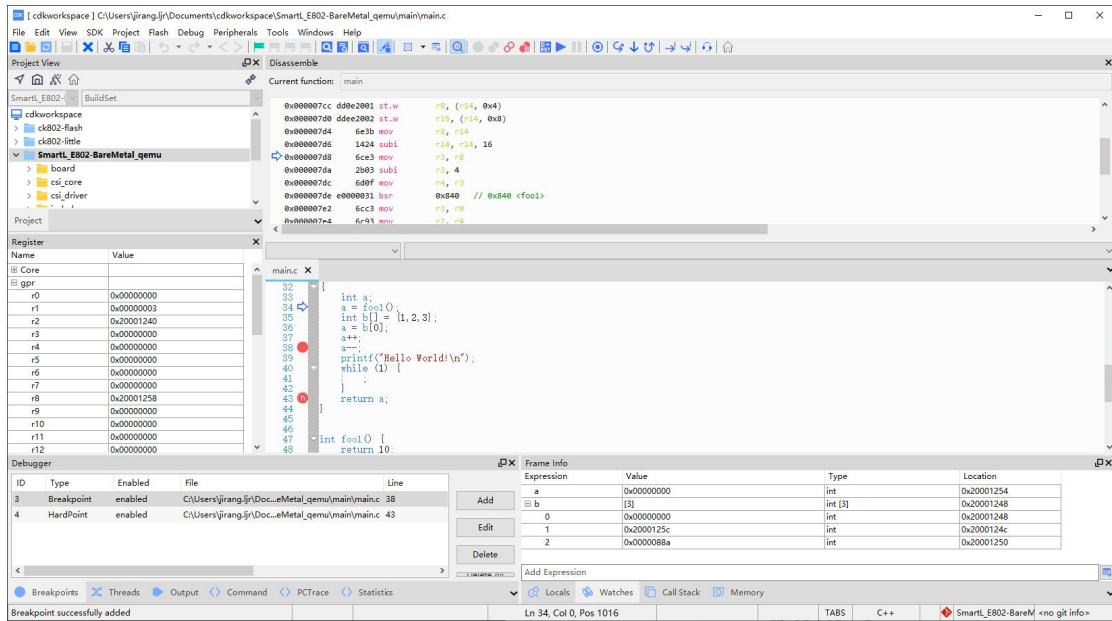
    Inferior 1 [Remote target] will be detached.

Quit anyway? (y or n) y
Detaching from program: /home/xiaof/workspace/Smrtl_E907FP/projects/benchmark/dhrystone/out/smrtl_e907_evb.elf, Remote target
Ending remote debugging.
[Inferior 1 (Remote target) detached]
```

GDB 详细手册参考：<https://www.gnu.org/software/gdb/documentation/>

剑池 CDK

剑池 CDK 是玄铁 CPU 的开发环境，集成了编辑、编译、调试等功能，便于开发者使用。本章节简要介绍以下 CDK 的调试功能，如下图所示：



CPF

CPF (C-SKY Profiling analysis utils) 是一套针对玄铁 CPU 应用程序的性能分析工具，包含了一系列记录和分析的工具。CPF 基于 T-Head trace（目前仅 T-Head qemu 支持 T-Head trace 功能）进行分析，通过 T-Head trace 硬件能够非侵入式地获取 CPU 运行时的信息，真实反映 CPU 的运行轨迹。借助于 CPF，开发者可以提高开发效率，优化系统设计。

```
# Tot Cycles: 2044 Tot Insts: 1714
# Read Bytes: 1543 Write Bytes: 781
# Overhead      Cycles      Insts      Call times      Symbol
# .....
# .....
# 42.80%      875      784      13      CK_Uart_PutChar
# 19.17%      392      344      13      fputc
# 7.24%      148      113      1      __GI_fputs
# 5.28%      108      81      1      CK_Uart_ChangeBaudrate
# 3.71%      76      56      1      CK_Uart_SetRXMode
# 3.57%      73      56      1      CK_Uart_SetTXMode
# 3.37%      69      53      1      CK_Uart_SetWordSize
# 3.32%      68      53      1      CK_Uart_SetParity
# 3.27%      67      53      1      CK_Uart_SetStopBit
# 2.49%      51      40      1      CK_Uart_Open
# 2.29%      47      38      1      CK_Uart_Init
# 0.92%      19      8      1      __GI_puts
# 0.88%      18      12      1      __start
# 0.83%      17      13      1      main
# 0.44%      9      5      1      __exit
# 0.14%      3      2      1      __GI_os_critical_enter
# 0.09%      2      1      1      __goto_c
# 0.09%      2      2      1      __GI_os_critical_exit
```

这是一个 HelloWorld 的打印输出示例程序，通过 CPF 工具，可以直观地看出该程序各函数在执行工程中的耗时占比，除此之外，还能看到程序执行的 cycle 数、指令数，被调用的次数等。

PERF

Perf 是一系列强大的性能分析工具集合。在 Linux 2.6.31 版本引入，至今 tool/perf 目录拥有 1 万多个提交，是内核开发中最活跃的几个领域之一，通过 perf 可以使用一到两行命令就完成像程序热点采样，接口调用分析，阻塞分析。

玄铁 CPU 的 PMU 单元支持了大量硬件事件计数器，包含了指令数，周期数，cache 访问，分支预测等等，下图以 memcpy 为例演示 perf 通过 PMU 或者程序执行的指令数、周期数。

tst-mem2 比 tst-mem 具有更大的循环拷贝次数。

```
# perf stat -e instructions,cycles ./tst-mem
Performance counter stats for './tst-mem':
      1784503      instructions      #    0.72  insn per cycle
      2471128      cycles
    0.101622134 seconds time elapsed
    0.058743000 seconds user
    0.054826000 seconds sys

# perf stat -e instructions,cycles ./tst-mem2
Performance counter stats for './tst-mem2':
     328971329      instructions      #    1.06  insn per cycle
     309475863      cycles
    10.344352812 seconds time elapsed
     9.293407000 seconds user
     1.051254000 seconds sys
```

关于平头哥玄铁 CPU 的调试系统介绍就到这里。其他关于玄铁 CPU 调试系统的文档请参考：

- RISC-V debug spec:
<https://github.com/RISC-V/RISC-V-debug-spec>
- GDB 用户手册:
<https://www.gnu.org/software/gdb/documentation/>
- 玄铁 CPU 调试技巧:
<https://occ.T-Head.cn/document?spm=a2cl5.27297580.0.0.4ee6F7oaF7oaEa&temp=thead-cpu-debug&slug=thead-cpu-debug>
- 剑池 CDK 用户手册:
<https://occ.T-Head.cn/document?temp=preface&slug=cdk-quick-userguide>

4.HHB 神经网络模型部署工具

1) 概述

HHB (Heterogeneous Honey Badger) 是针对玄铁全系列处理器、无剑 SoC 平台的神经网络模型部署工具集，包括了编译优化，性能分析，过程调试，结果模拟等一系列部署时所需的工具。该工具集通过向开发者提供从模型优化到部署调试全过程的辅助工具，简化 AI 模型应用落地到端侧设备的开发过程，从而缩短 AI 产品的开发周期。

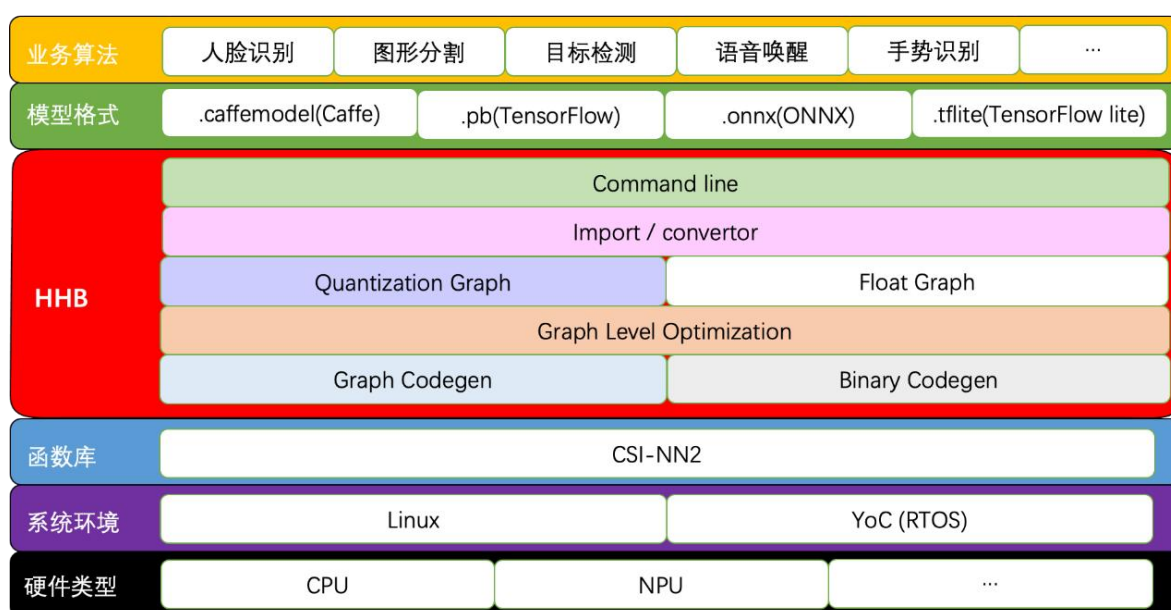
HHB 的功能与特性：

- 支持 caffe, tensorflow, onnx 和 tflite 对应格式的模型
- 支持 8/16 位定点和 16 位浮点等数据类型
- 支持对称和非对称定点量化，支持通道量化
- 可在部署前优化模型的网络结构
- 编译生成可在玄铁芯片平台上执行的二进制
- 支持在主机上做行为模拟
- 多组件形式的工具集方便二次开发
- 符合传统习惯的 Unix 命令行模式

2) HHB 层次结构

HHB 已支持语音和视觉的多种不同业务算法,可导入 Caffe 和 TensorFlow 等不同训练框架的模型。

HHB 以开源项目 [TVM](#)为基础架构, 添加了丰富的命令行选项提供命令模式; 预置了多类量化算法适配不同平台可支持的数据类型; 根据不同平台特点, 输出调用 CSI-NN2 的 C 代码, 或者直接输出可执行的二进制。

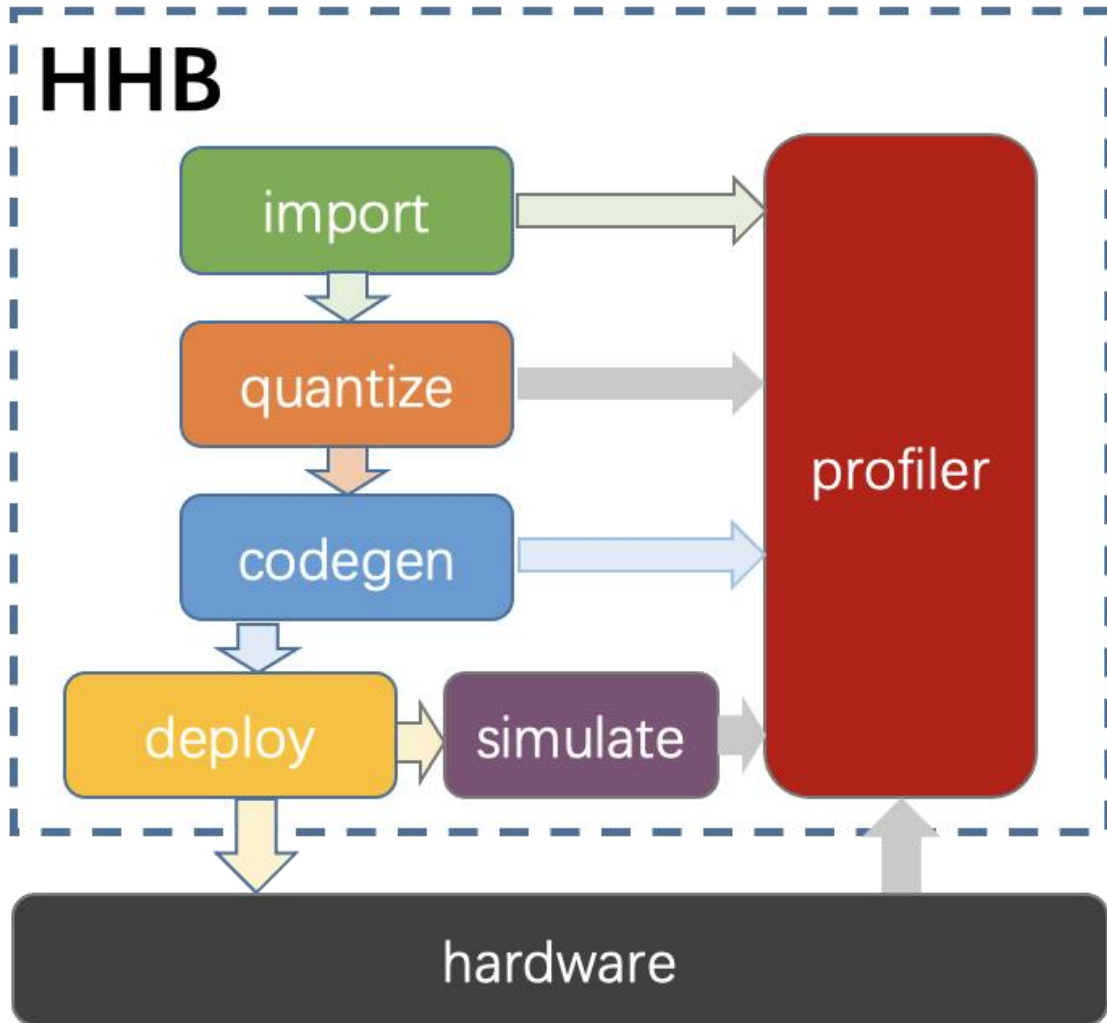


3) 命令行工具集

HHB 工具集包括了如下一些独立命令行工具:

- import: 模型导入工具, 负责将现有的模型转换为 HHB 私有的格式。
- quantize: 量化工具, 负责将 import 出来的文件量化, 生成量化后的中间文件。
- codegen: 代码生成工具, 调整中间文件为生成二进制做准备。
- deploy: 特定平台的部署工具, 负责生成二进制及配套的头文件和源代码。
- simulate: 行为模拟工具, 负责在主机上模拟模型执行。
- profiler: 性能分析工具, 负责分析网络中的热点子图, 以及提供在特定平台上的优化建议。

相互关系大致如下图所示，其中 deploy 和标灰的是暂时还不支持的部分：



4) 获取安装

HHB 版本发布时，提供独立的可执行程序包，也提供包含示例的完整 docker 环境。推荐使用 docker，docker 安装参考附录《[Docker 安装和导入镜像](#)》。

从 OCC 站点下载：[AI 部署工具](#)

环境要求

推荐使用已测试过的执行环境：

- 可执行程序包：ubuntu 18.04
- docker：Docker version 19.03.4

HHB 默认以 docker 镜像的方式发布，最简单的获取方式是 docker 官方的镜像仓库：[dockerhub](#)。

获取 HHB 2.0 可以使用命令：

```
docker push hhb4tools/hhb:lastest
```

如果未安装 docker，可以参考附录《[Docker 安装和导入镜像](#)》先安装 docker。更多关于 HHB 神经网络模型部署工具相关内容可以参考《[HHB 用户手册](#)》。

5. 剑池 CDK 开发工具课程学习

平头哥芯片开放社区官网上线了 RISC-V 系列课程培训，该课程和清华大学、大连理工大学等合作，包括由教育部评选为“国家级一流本科课程”的《嵌入式软件设计》，成为国内嵌入式教育的标杆课程。在“中国大学 MOOC”平台已开设 9 轮课程，累计选课人数达四万人次以上，广受学生欢迎。通过 XuanTie RISC-V Institute，可以学习以 RVB2601 开发板应用实例来介绍剑池 CDK 开发工具的特性以及使用教程。欢迎广大开发者[登录 OCC](#)了解学习。

XuanTie RISC-V Institute

基于玄铁E906处理器的RVB2601开发实践

[课程目录](#) [课程要点](#) [课程问答](#)

- RVB2601开发板
 - ☑ RVB2601开发板介绍与DEMO展示
 - ☑ RVB2601操作系统及开发工具介绍
 - ☑ RVB2601开发实战演示
- 云上实验室全面解读
 - ☑ 云上实验室功能全面解读
 - ☑ 云上实验室DEMO演示
- 实战展示
 - ☑ 平衡小车

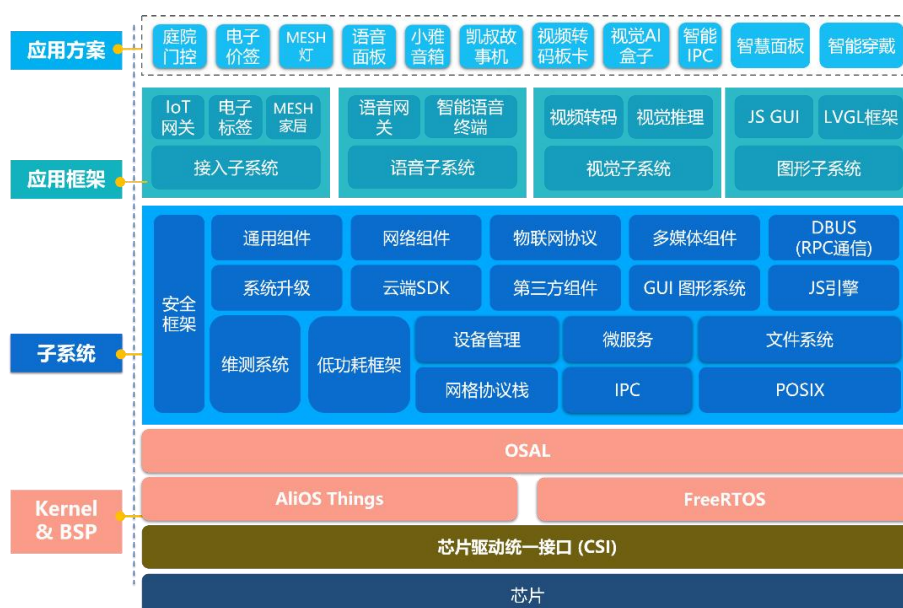
平头哥玄铁 CPU 系统

1.YoC

1) 平台架构

YoC (Yun on Chip) 以极简开发为理念，以 CPU 架构、芯片平台、操作系统、云服务和开发套件为基础，助力开发者从芯片到云的全链路高效设计，是面向 IoT 领域的全栈技术平台。

- **极简开发**：CDK 的 IDE 集成开发工具，让开发、调试更方便
- **芯片生态**：支持将快速 YoC 平台移植到各类芯片，打造丰富的芯片生态
- **深度优化**：基于平头哥处理器，对软件进行深度优化，具有更高的代码密度与最高的性能
- **组件丰富**：组件覆盖了实时操作系统、协议栈、音视频处理、文件系统、设备管理、低功耗管理等
- **面向业务**：面向行业领域的应用框架，进一步缩短产品开发周期，实现云端一体的编程体验
- **高级安全**：高级安全机制，TEE 框架为物联网设备提供安全保障
- **商业合作**：与行业标杆企业进行技术合作，提供成熟的技术解决方案



YoC 平台定义了芯片的统一接口，提供应用最基础的核心服务，提供了大量独立的应用组件，软件架构清晰、系统模块化并且可裁剪性非常好。针对芯片移植需求，只需要根据芯片驱动接口层（CSI）的定义，即可以将 YoC 移植到该芯片上。

针对资源受限的微控制器（MCU）系统，也可以裁剪出只需要几 KB 的 FLASH，几 KB 的内存消耗的系统。对于资源丰富的物联网设备，YoC 提供可定制的核心服务，支持 AOS API/POSIX API，以及丰富的独立组件，以及更加面向领域的业务框架。

2) YoC 特性

YoC 平台具备以下特点：

- **极简开发**
 - 提供 CDK IDE 开发工具
 - 提供 Shell 交互，支持内存踩踏、泄露、最大栈深度等各类侦测
 - 提供包括存储（掉电保护、负载均衡）在内的各类产品级别的组件

- **高度优化的性能**
 - 内核支持 Idle Task 成本，Ram<1K，Rom<2k，提供硬实时能力
 - 支持 CPU 加速指令，深度优化内核性能

- **全面的安全保护**
 - 提供系统和芯片级别安全保护
 - 支持可信运行环境（TEE）
 - 支持预置 ID2 根身份证和非对称密钥以及基于 ID2 的可信连接和服务
 - 优化 mbedtls，对 footprint 进行高度优化
 - 专业安全公司渗透测试

- **IoT 专属组件**

- 空中固件升级 (FOTA)
 - 集成 AT 模组指令
 - 快速 IoT 云端接入
 - 支持多种物联网协议: Alink、MQTT、COAP、LWM2M
 - 支持多种网络协议栈:
 - ①TCP/IP 协议栈 (LwIP)
 - ②套接字适配层 (SAL)
 - ③支持 WiFi、有线、NB-IoT、GPRS、Bluetooth 等通信硬件
 - 支持众多厂家的传感器接入
-
- **面向领域的软件框架**
 - 面向 IoT 数据上云软件框架
 - 面向智能语音应用的软件框架
 - 面向视觉 AI 应用的软件框架

2.Linux

1) 概述

[玄铁 Linux 操作系统](#)旨在为玄铁 RISC-V 处理器生态芯片提供丰富的 Linux 软件组件。平台提供图形、多媒体、网络等常用 1000+基础组件的同时,重点打造了与玄铁处理器紧密结合的核心组件,以及 Yocto 系统配置、构建和发布体系。

一个嵌入式 Linux 系统一般由 4 个部分组成:

- bootloader: 常使用 u-boot。芯片首先加载 u-boot,然后由 u-boot 加载并启动 Linux kernel。
- kernel: Linux 操作系统的核心,它负责硬件的驱动,以及提供系统所需的核心功能,如进程调度、内存管理、文件管理等。
- rootfs: 根文件系统包含系统启动时所必须的目录和关键性的文件,以及使其他文件系统得以挂载所必要的文件,如 init 进程、shell、基础配置文件等。

- Application: Linux 有着极其强大的开源生态环境，其应用程序小到一个实用的用具，大到像音频系统、图形系统、web 浏览器、数据库编程语言等都能找到对应的开源软件，用好这软件可以极大地加快产品化进程，同时开发者也应积极回馈开源社区，以期建立良性生态的回环。

T-Head 支持以下两种形式来快速构建 Linux 发行版：

- Buildroot
- Yocto

这两种构建方式各有特点，前者比较轻量化，后者可以构建更复杂的系统，并支持安装包管理。

2) 特色核心服务

T-Head Linux 发行版着重打造有特色的多个核心服务，加快应用开发与产品落地。



特色核心服务 1: JS GUI

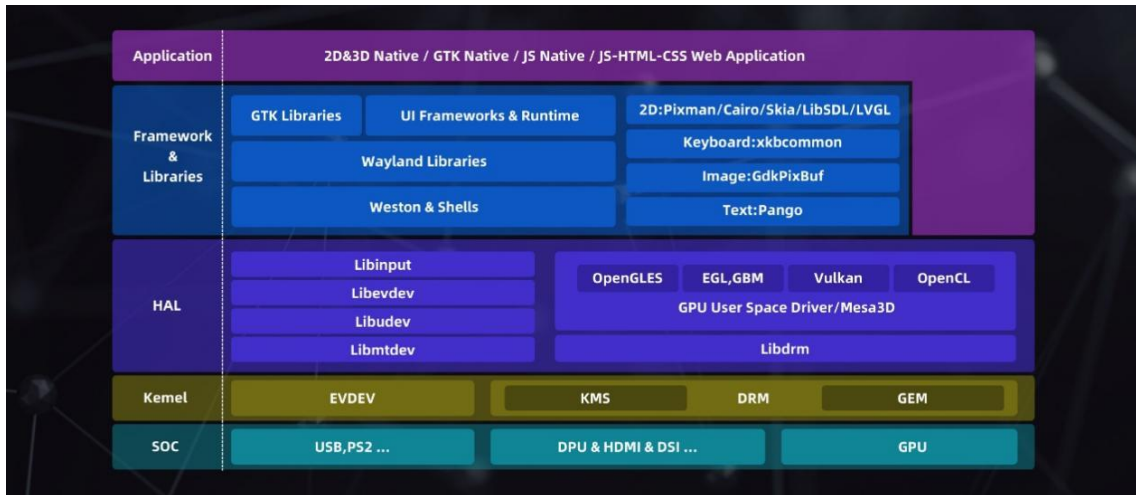
- 与阿里云 IoT 共建，JS GUI 平台在 RISC-V 上优化
- 提供丰富的应用生态，包括 WiFi 服务、音频服务、语音 AI 服务等
- 基于 Yocto 的精简的 BSP 开发包，操作系统镜像以及轻量级系统资源占用，启动加速，硬件加速

- JS GUI，阿里定义的 IoT 小程序规范，同时支持阿里生态应用框架；支持自动部署和快速简洁的开发应用；基于 vue.js，拥有简单的开发调试环境，以及丰富的 UI 组件



特色核心服务 2：图形系统

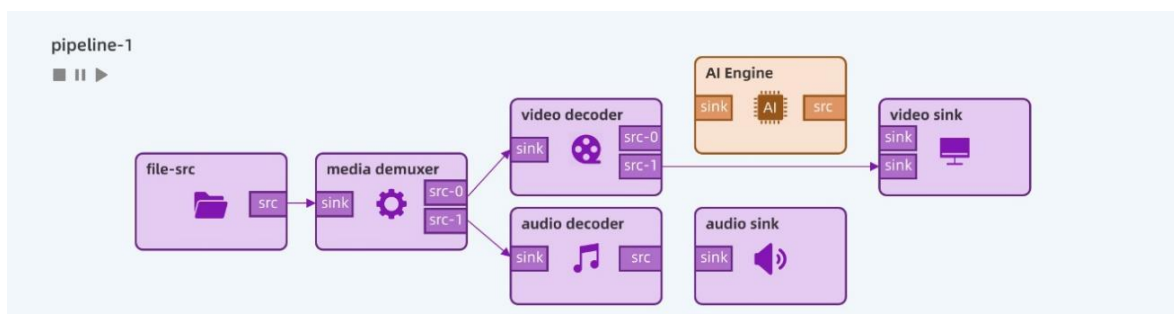
- 成熟完整，生态丰富，兼容性强
 - 追踪 Wayland-Linux 的最新标准
 - 实现 GTK-Linux 生态中完整且成熟方案 (+QT)
 - 运行于支持 Display&GPU 最新标准和传统图形硬件架构的 RISC-V SoC Linux 图形栈
 - 性能强大的 3D GPU，支持 OpenGL ES 3.x、Vulkan、OpenCL 等主流规范
- 扩展性强，易定制化
 - 采用 Weston，即 Wayland 的官方参考实现，模块化软件架构，易于二次开发和定制化
 - 社区对 Weston 已有的重量级应用生态有良好支持，上层 UI 框架可以透过 Wayland/Weston 作为显示窗口平台，而不用框架本身考虑硬件平台移植性和适用性



特色核心服务 3：视频视觉

采用业界流行的开源 GStreamer 方案，将 SoC 中的硬件加速模块做成 gst 插件，可以快速完成应用 pipeline 搭建。

- 自研插件包括：thead_camera（视频输入）、thead_infer（AI 推理）、thead_fce（特征值搜索引擎）
- 对接标准插件：gst-omx（视频编解码）、kmssink/waylandsink（视频图像输出）
- 海量官方插件：gst base/good/bad/ugly 共计 1600+个插件 gstreamer 插件串接示意图：



特色核心服务 4：安全子系统

1. RISC-V 业内首个可量产的安全系统
2. 金融支付级安全定位
3. 平头哥原生安全处理器的推广和落地
4. 共享ARM架构的OPTEE软件生态，根据不同的应用场景灵活扩展



Alibaba Cloud Computing Ltd.
 Building 8, NO.16
 Zhuangtang, Sci-tech Economic
 Block
 Xihu District, Hangzhou
 Zhejiang Province
 China

Contact:
 Lufei Wei
 Lufei.Wlf@alibaba-inc.com



全球首个基于RISC-V架构的GP TEE Secure Certificate

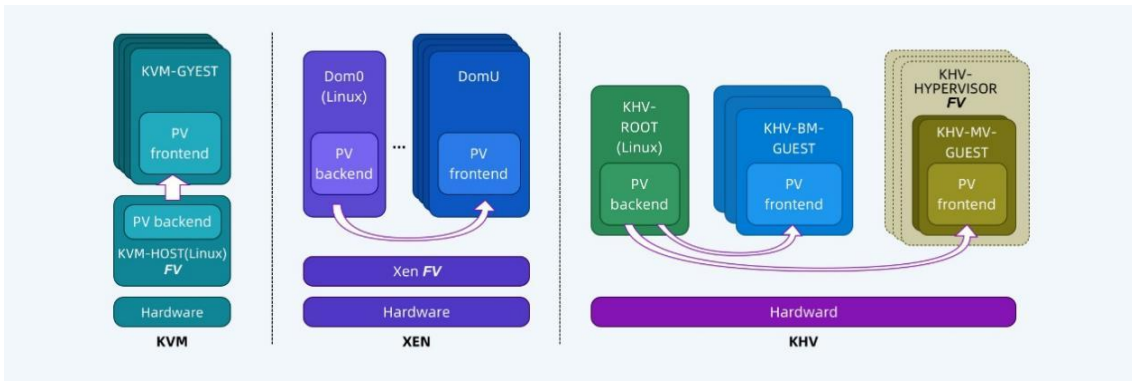


The diagram illustrates the architecture of the security subsystem, divided into two main parts: **Secure Provision Platform** and **Secure System Platform**. The **Secure Provision Platform** includes **Offline Key Management Service** and **Online Key Management Cloud Service**, which interact with **KP Management Service** and **KP Client** to manage **ATE** (Attestation Target Environment). The **Secure System Platform** is built on **Trusted Firmware (ATF v1.1, OpenSBI v0.9)** and is divided into **TEE** (Trusted Execution Environment) and **REE** (Rich Execution Environment). The TEE contains **TA-1, TA-2, TA-3** and uses **Global Platform TEE Internal APIs**, **Trusted H/w Driven**, **OPTEE-OS Kernel**, and **Primitive PTA**. The REE contains **TA-1, TA-2, TA-3** and uses **Global Platform TEE Client APIs**, **TEE Driver Library**, **Linux Kernel**, and **Secure Network**. The platform also supports **安全RV核** (Security RV Core) and **通用RV核** (General RV Core) on **平头哥** (Plattop) and **RISC-V** architectures, along with **Trusted H/W Resource**, **SE** (Secure Element), and **安全芯片平台** (Security Chip Platform). The platform functions include **安全启动** (Secure Boot), **安全升级** (Secure Upgrade), **安全存储** (Secure Storage), **安全隔离** (Secure Isolation), **安全计算** (Secure Computation), **安全调试** (Secure Debugging), and **安全TAA管理** (Secure TAA Management). The entire system is supported by **安全设置** (Security Settings). A purple arrow at the bottom indicates **End to End Security Solutions**.

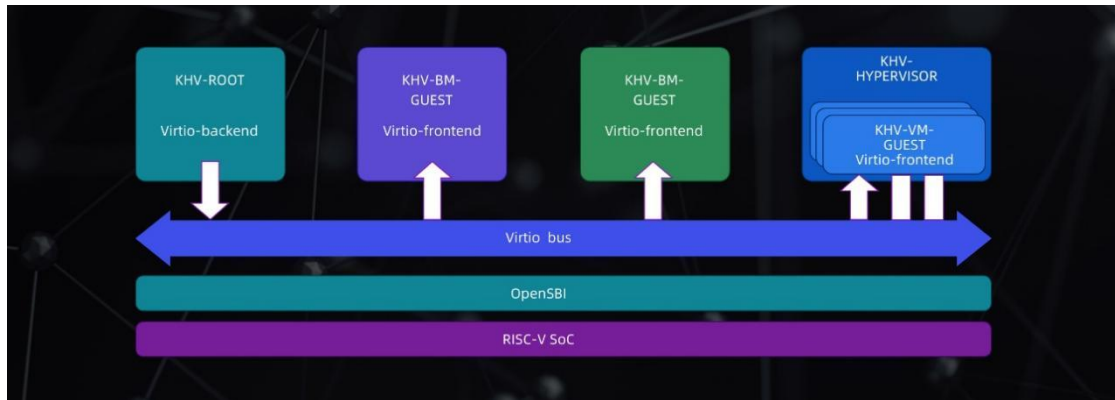
特色核心服务 5：异构虚拟化（HKV）

KVM 受制于 Linux 内核的复杂性，以及安全认证的难度和成本；因此，在嵌入式高可靠场景中并未普及。KHV-based Heterogeneous Virtualization（基于 KVM 的异构虚拟化），独立于硬件虚拟化特性（KVM 依赖 CPU 硬件虚拟化），并遵循原则静态分区，本地化 hypervisor。

此外，KHV 继承 KVM 软件生态，同时满足高可靠嵌入式系统对静态分区和灵活共享的要求。



KHV 相比 KVM 和 Type-1 hypervisors (e.g. XEN) 更适合要求静态分区的高可靠嵌入式场景。与 KVM 不同，即使 CPU 没有硬件虚拟化特性，KHV 也能运行，并复用 KVM 软件生态。



3.Android

1) 概述

[安卓 \(Android\)](#) 是全球主流移动设备操作系统，拥有丰富成熟的应用生态，其系统核心代码部分即 AOSP (Android Open Source Project)。AOSP 开源仓库对 RISC-V 移植补丁的合入，标志着 Android 应用生态与 RISC-V 国际生态的融合，也为 RISC-V 在高性能应用场景的商业落地打下基础。

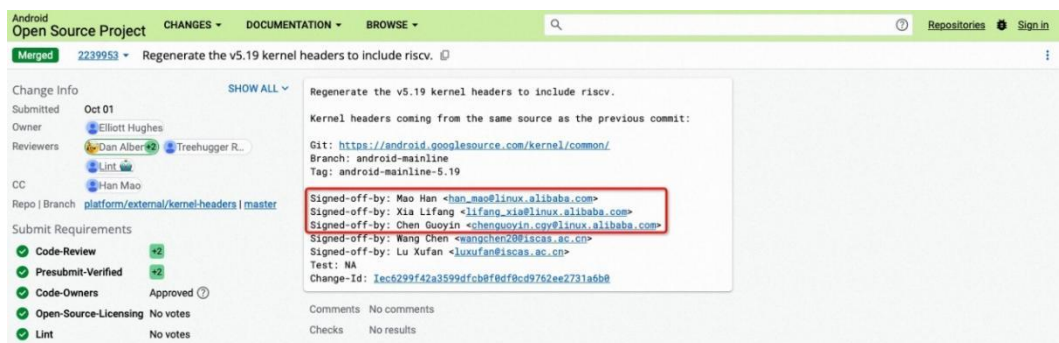
平头哥在 Android 基础组件和框架、Chrome 浏览器、安全、多媒体、视觉 AI 和诊断体系等方面做出了重要的贡献。

2) RISC-V ISA 的安卓软件栈支持

安卓软件栈主要包括系统内核、硬件抽象、运行时、框架层、应用五个层次的近千个软件包。其中作为基础的 Linux 内核、GCC 工具链、Clang/LLVM 工具链已经支持了 RISC-V 架构；因此对的主要支持工作集中于 RISC-V ISA 的编译框架支持、Bionic C 库的 ISA 支持、ART JAVA 运行时的 ISA 支持、RVB-ICE 开发板板级的驱动对接、OPENGL 的对接、Chromium webview 浏览器几个部分；其他依赖包括 NDK、VNDK、emulator、unwind 解析库、编解码库等等。

3) Android 接收 RISC-V 补丁

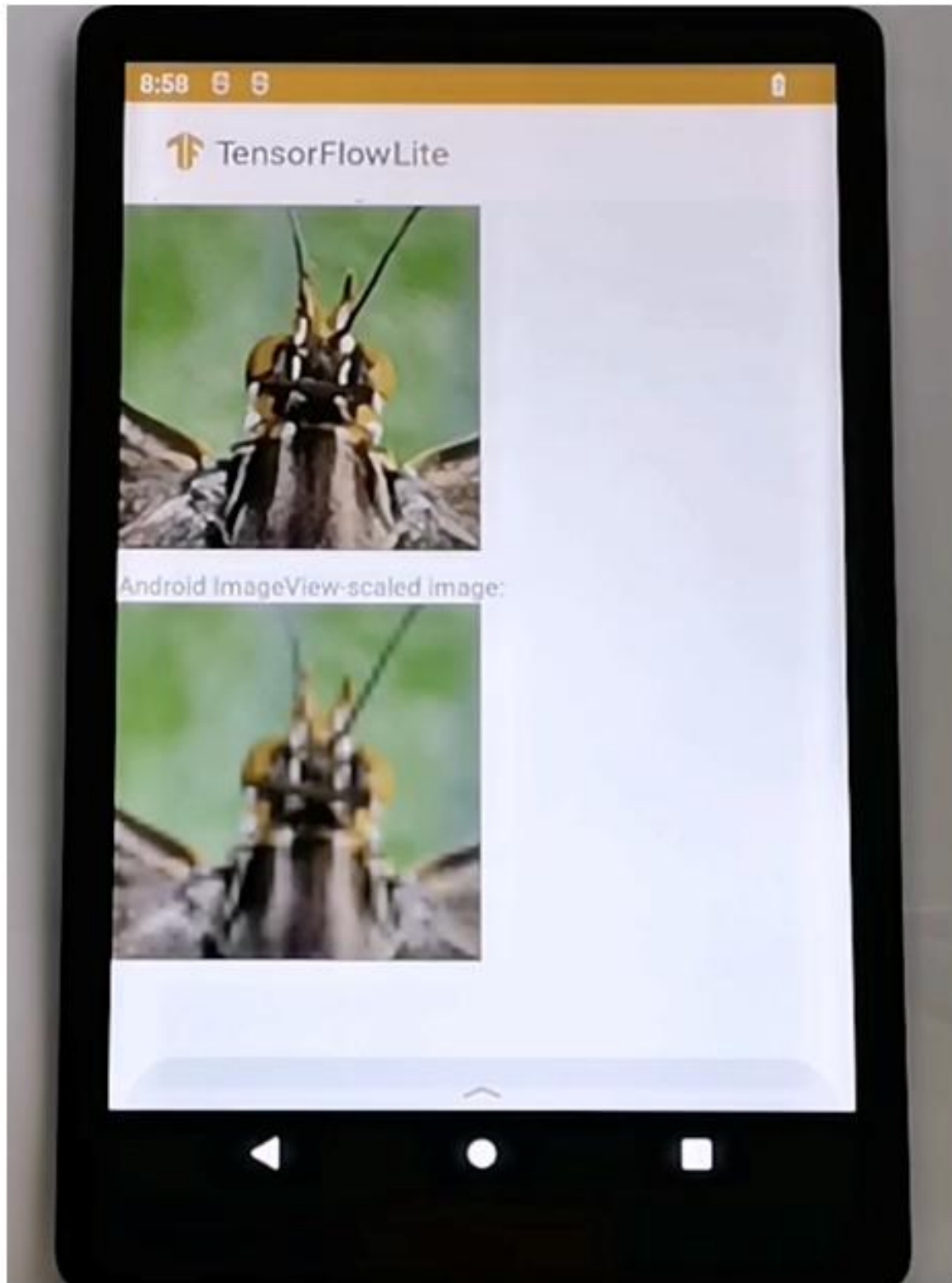
由阿里巴巴平头哥贡献的 RISC-V 移植安卓的代码补丁集合，被安卓 AOSP 社区收录进系统源代码，成为全球首批 RISC-V 兼容安卓的正式补丁。这意味着谷歌安卓开启了对 RISC-V 架构的官方原生支持，RISC-V 与安卓两大体系的融合驶入快车道。



作为近年备受瞩目的新架构，RISC-V 与安卓系统的适配，被业界普遍视为软硬件融合发展的新方向。但因 RISC-V 架构仍在演进，而安卓系统已成熟，打通两大体系的技术挑战十分艰巨，仅编译系统、LLVM 编译工具链、C 函数库等基础技术就需要大量的优化适配工作。

2021 年 10 月，平头哥最早完成了安卓系统的整体移植，在玄铁 C910 处理器上首次兼容安卓 10.0 系统，并运行 Chrome 浏览器等应用；2022 年 4 月，玄铁 C910 在安卓 12.0 系统上成功运行 TensorFlow Lite，集成多项第三方关键组件，证明了 RISC-V 可兼容不断更新的安卓系统。

过去两年间，平头哥在 RISC-V 领域的一系列技术突破及产业化落地，极大提振业界对 RISC-V 的信心，促成谷歌安卓对 RISC-V 架构提供官方支持。



2022年6月,谷歌与平头哥达成公司层面的贡献者许可协议(CLA, Contributor License Agreement), 双方就安卓系统支持 RISC-V 架构等工作进行技术合作和分享。

9月30日,谷歌安卓 AOSP 社区开始接收 RISC-V 补丁。平头哥率先提交关于 Bionic C 库、模拟器、三方上游模块等 76 项基础代码补丁, 其中, 与 AOSP 社区合作直接并入官方补丁 18 项, 拆分合入 C 库关键补丁 12 项, 外部项目合入补丁 4 项, 成为全球首批安卓系统吸纳的 RISC-V 补丁。

4) RISC-V 成功运行 Android 12

在 2022 RISC-V 国际峰会上, 阿里平头哥展示了 RISC-V 架构与安卓体系融合的最新进展: 基于 SoC 原型曳影 1520, RISC-V 在安卓 12 (AOSP) 上成功运行多媒体、3D 渲染、AI 识物等场景及功能。这意味着安卓系统在 RISC-V 硬件上得到进一步验证, 两大体系融合开始进入原生支持的应用新阶段。

在大部分基础功能成功实现后, RISC-V 与安卓的融合进入应用验证领域, 面临更多模块缺失、接口不一致等技术和系统挑战。比如在车载场景中, 硬件层需重新设计总线以支持多路输入, 系统层要兼容外设硬件、满足更高安全需求, 应用层还需打通和协调多个车载子系统。

“为更好补齐两大系统融合的生态短板, 平头哥着重在测试、性能优化及开源协作等方面推进根本问题的解决。” RISC-V 国际基金会安卓技术组 (Android SIG) 主席、平头哥技术专家毛晗说。据他介绍, 平头哥修复及澄清了 28% 的 CTS 和 57% 的 VTS 测试用例集, 用例通过率大幅提升至更稳定的 89% 及 78%, 使得测试完成从单位模块到整体系统的飞跃; 针对安卓系统里的 Bionic、Art 和图片编解码等关键原生功能, 平头哥通过软硬一体优化, 最高提升 60% 的性能。

基于曳影 1520, 平头哥实现了从底层硬件到安卓系统再到上层应用的完全打通, 相关成果 demo 在 2022 RISC-V 国际峰会上首次演示: 一个 demo 视频演示多路编解码能力, 编码时长从每帧 100ms 大幅压缩到 5ms; 另一个 demo 演示高性能 3D 渲染, 提升了两大体系融合的性能天花板。

“在完成 RISC-V 与安卓 10 系统的兼容、安卓 12 系统的深度验证后，平头哥推动两大体系融合挺进第三阶段：让 RISC-V 得到安卓原生支持，真正进入安卓系统上游代码，实现更多预编译组件和系统镜像的稳定编译和输出，推进两大体系的协同演化。”毛哈介绍说。据了解，目前，平头哥已提交 80 个 RISC-V 技术补丁，其中 50 个被吸纳入安卓 AOSP 原生系统中去，是两大体系融合的关键推动者。

RISC-V 玄铁系列开发板实践

1. 基于玄铁 C906 处理器的 D1 Dock Pro 开发实践

1) 概述

D1 Dock Pro 开发板是由深圳矽速科技有限公司研发，搭载了全志 D1 多媒体处理器，基于平头哥 RISC-V 64 位 C906 核心，支持 RVV，1GHz 主频，可运行 Linux、RTOS 等系统。为开源开发者设计的一款功能集成度高、体积小且价格亲民的 RISC-V 开发套件。为了让开发者方便调试各种显示功能，它不仅支持 HDMI 输出，还支持 MIPI 和 RGB 显示接口，板载丰富的外设（2.4G Wi-Fi+BT 模块、USB Host Type-A 连接器、数字麦克风和音频功放）。

除此之外，开发者还可以通过开发板的 GPIO 拓展端口，开发一些大家意想不到的新奇有趣的功能。

2) 开发板介绍

a) 规格介绍

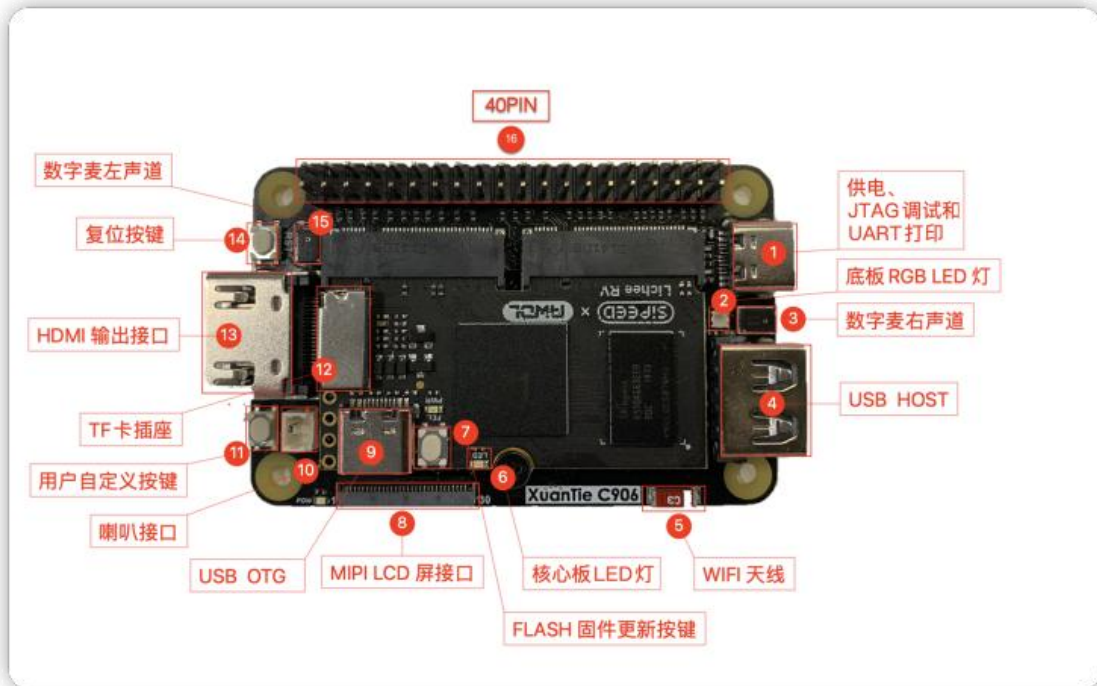
配置	说明
CPU	XuanTie C906 RISC-V CPU
内存	512MByte DDR3
显示输出	支持 RGB 接口输出，最大支持 1080P@60fps， 支持 MIPI 接口输出，最大支持 1080P@60fps， 支持 HDMI 接口输出，最大支持 4K@30fps

无线网络	支持 2.4G WiFi, 符合 IEEE 802.11b/g/n 支持 BT, 符合 BT4.2
USB 连接	板载一个 USB Host type-A 母座 (在底板) 板载一个 USB OTG type-C 母座 (在核心板)
音频	板载 3W 扬声器驱动电路 板载 2 路数字麦克风
存储	板载 TF 卡连接器 (在核心板) 板载 128Mbit SPI FLASH (在底板)
GPIO	通用 2x20PIN GPIO 口
其他功能	板载一个 WS2812 RGB LED, 一个复位按键和一个用户按键
外形尺寸	65.0mm x 42.5mm

b) 硬件接口说明

核心板资源介绍

D1 Dock Pro 开发板板载 JTAG 调试器, 在成功安装 T-Head DebugServer 调试工具后, 只需连接 USB 口 (序号 1) 即可开始烧录和调试工作。

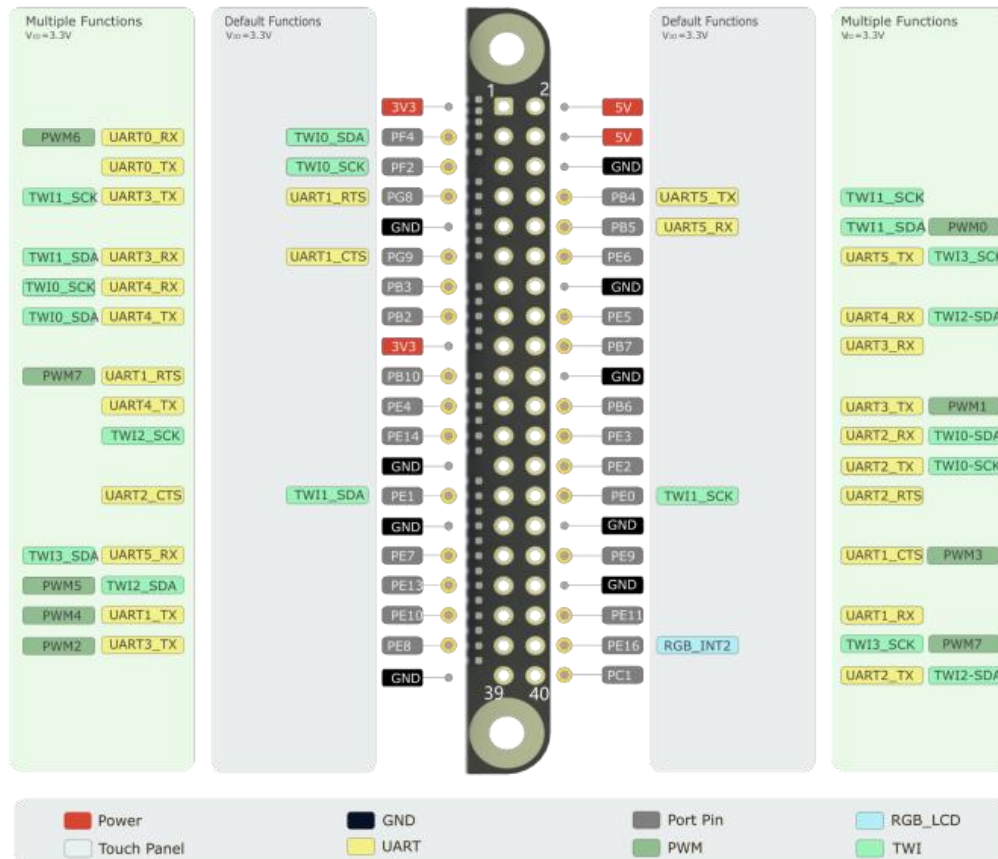


序号	接口	说明
1	TYPE-C 接口	支持 JTAG 调试 支持 UART 打印 可为板子提供 5V 供电
2	RGB LED	底板 LED，支持红、黄、蓝三色，默认常灭
3	MIC	板载数字麦克右声道
4	USB	USB2.0 HOST
5	WiFi 天线	支持 2.4G WiFi
6	LED	核心板用户自定义 LED 灯，默认常灭
7	FEL 按键	Flash 固件更新按键，配合全志烧写器使用。在断电模式下按住此按键不放，核心板 TYPE-C 连接电脑主机，然后松开此按键，进入固件升级状态，可进行固件烧录

8	MIPI DSI	MIPI LCD 屏接口
9	TYPE-C 接口	USB2.0 OTG 接口 可以用于 Linux 系统下的 adb 调试 可用于 flash 固件更新 可为板子提供 5V 供电
10	SPK	喇叭接口
11	KEY 按键	自定义按键
12	TF 卡插座	TF 卡插座
13	HDMI	HDMI2.0 输出接口
14	RET 按键	系统复位按键，按一次实现复位
15	MIC	板载数字麦克左声道
16	40PIN	40PIN 支持 UART、PWM、I2C、GPIO 等

40PIN 管脚说明

开发板提供了 UART、PWM、I2C、GPIO 等接口，详细配置需要参考[芯片数据手册](#)。



详细 D1 Dock Pro 开发板接口介绍可以参考《[Lichee D1 Dock 开发板用户指南](#)》。

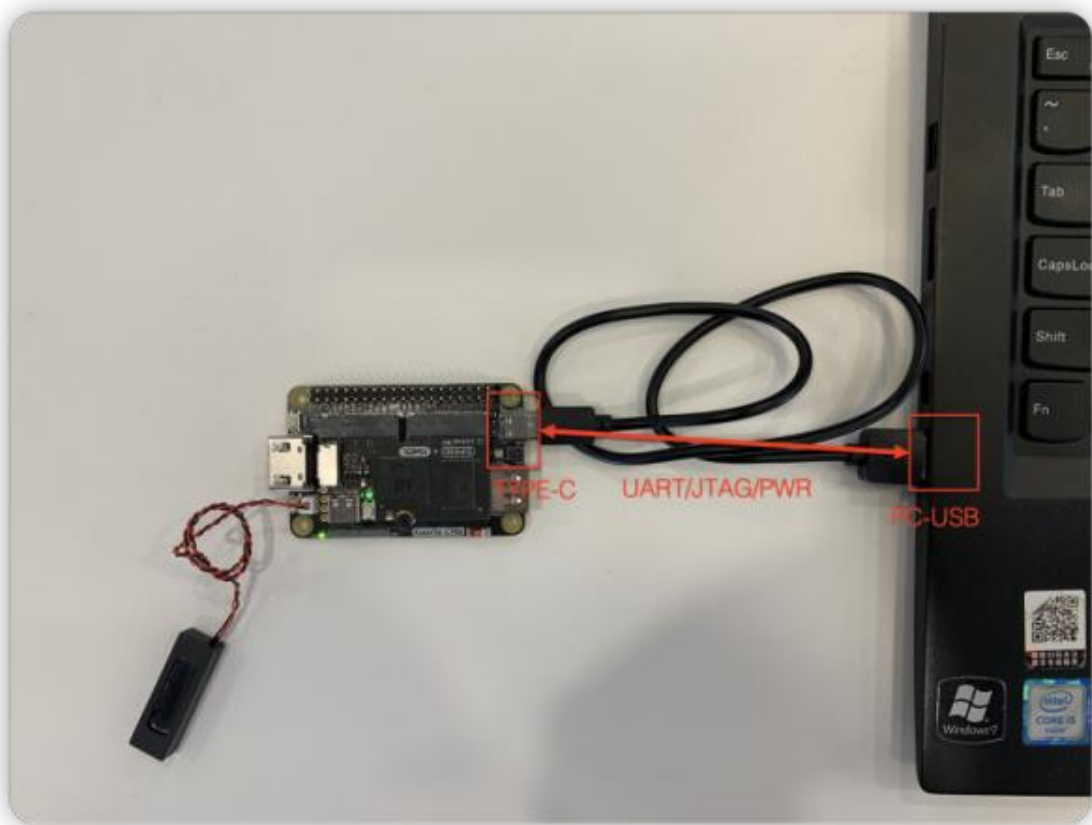
3) 开发环境搭建

a) 开发软件准备

栏目	获取方法	说明
SDK 包	源码获取	① 参考《源码获取》章节，下载源码并搭建好开发环境 ② yoctools 版本要求：>v2.0.32 ③ SDK 版本：>v2.1.0

工具链	yoc 指令 安装	① 版本信息： XuanTie-900-gcc-elf-newlib-x86_64-V2.4.0-20220428 ② 手动下载：链接
剑池 CDK	下载	Windows IDE 开发环境，版本要求：>v2.16.0
Debug Server 调试工具	下载	安装完成后，会提示安装驱动，请选择是。
D1 Dock Pro 开 发板	请联系商 务	D1 Dock Pro 核心板和扩展板

b) 硬件连接



参照上图进行开发板与 PC 的连接。

4) 不同开发环境下运行

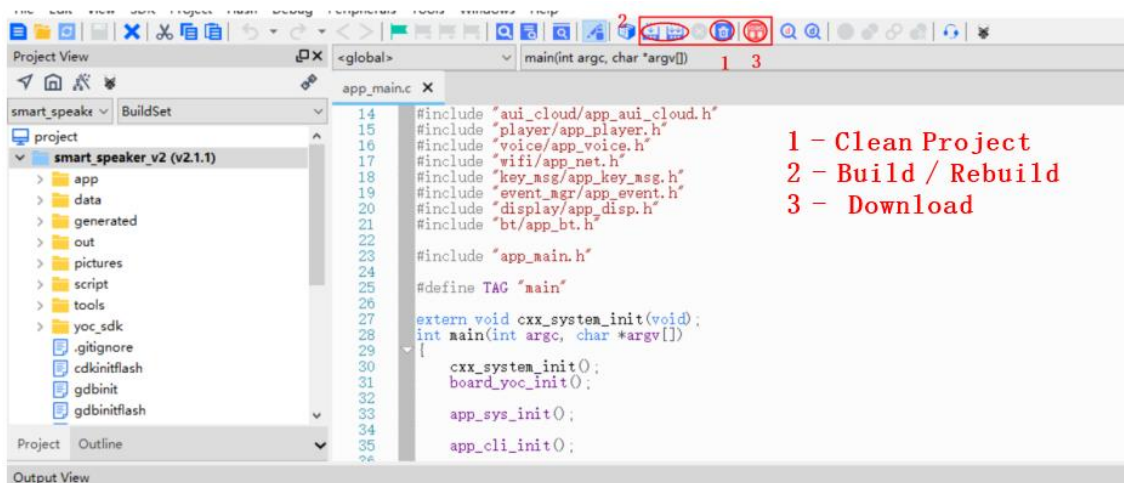
a) Windows 环境开发

本章节介绍 Windows 开发环境下的上手操作，操作步骤如下：

- 安装剑池 CDK。
- 参照《[源码获取](#)》下载相应版本的 SDK 源码。
- 重要步骤：进入 solutions/smart_speaker_v2 目录切换 yaml 文件。比如使用 package_d1_mind.yaml 替换 package.yaml。
- 双击 project.cdkproj 工程文件，先 Clean Project，再开始编译和调试。

编译

单击工具栏的【Clean Project】，再单击【Build】或者【Rebuild】按键，开发编译。

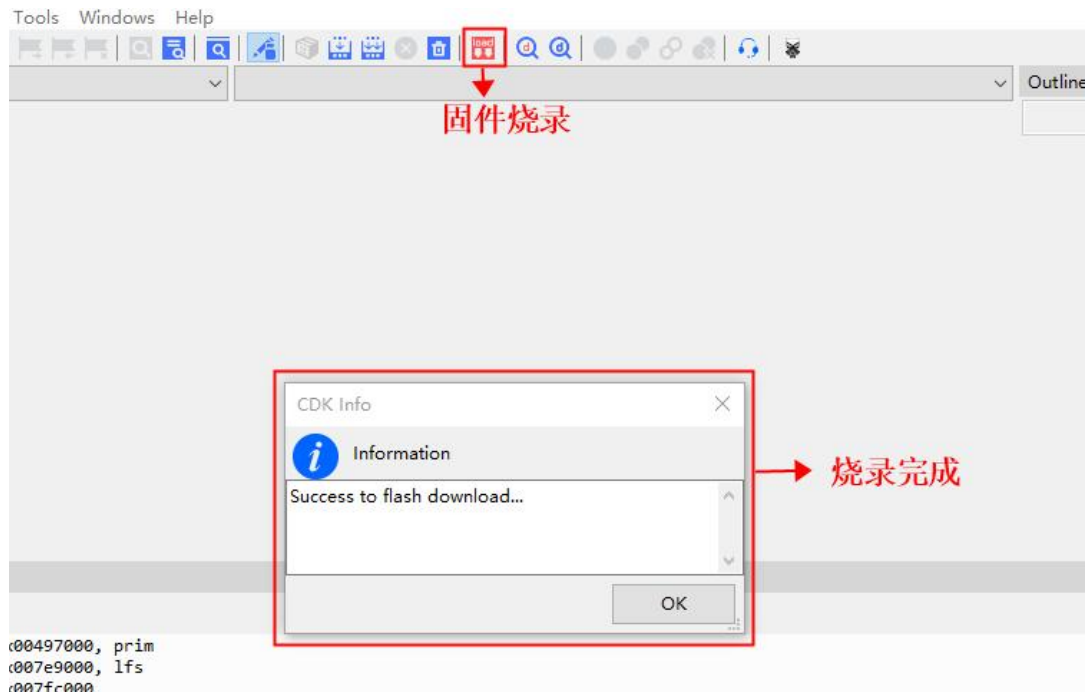


注意：

编译的速度依赖于 WINDOWS 下是否有安装杀毒软件或者监控软件，比如杀毒卫士 360，卡巴斯基，赛门铁克等。通过尝试删除或停止该类软件运行，可以提高编译速度。

烧录

① 先连接好硬件 JTAG。然后单击工具栏按钮开始烧写。



② 烧写完毕之后，按开发板复位键复位设备。

b) Linux 环境开发

本章节介绍 Linux 开发环境下的上手操作，推荐安装 Windows+WSL 环境。请参照《[源码获取](#)》下载相应版本的 SDK 源码和 yoctools 工具，同时搭建好开发环境。

编译

```
root@DESKTOP-xxx:/sdk_longyuan# cd solution/smart_speaker_v2

### 查看 yoctools 版本指令
root@DESKTOP-xxx:smart_speaker_v2# YoC--version
```

```
2.0.32
### 更新 yoctoools 版本指令
root@DESKTOP-xxx:smart_speaker_v2# sudo pip install yoctoools -U
...
Successfully installed yoctoools-x.x.xx

### 安装/更新工具链版本指令
root@DESKTOP-xxx:smart_speaker_v2# YoCtoolchain --RISC-V-f
Start to download toolchain: RISC-V 64-unknown-elf
 100.00% [#####] Speed:
5.428MB/S
Start install, wait half a minute please.
Congratulations!

### 查看工具链版本指令
root@DESKTOP-xxx:smart_speaker_v2# RISC-V 64-unknown-elf-gcc -v
...
gcc version 10.2.0 (XuanTie-900 elf newlib gcc Toolchain V2.4.0 B-20220427)

### 选择 D1 的 yaml 文件
root@DESKTOP-xxx:smart_speaker_v2# cp package_d1_mind.yaml
package.yaml

### WSL 环境下, 需要先执行下面的命令, 支持 32 位程序的生成
root@DESKTOP-xxx:smart_speaker_v2# sudo service binfmt-support start

### 编译
root@DESKTOP-xxx:smart_speaker_v2# make clean;make
...
Create yoc_rtos_8M.img in out directory Success!
scons: done building targets.
YoCSDK Done
```

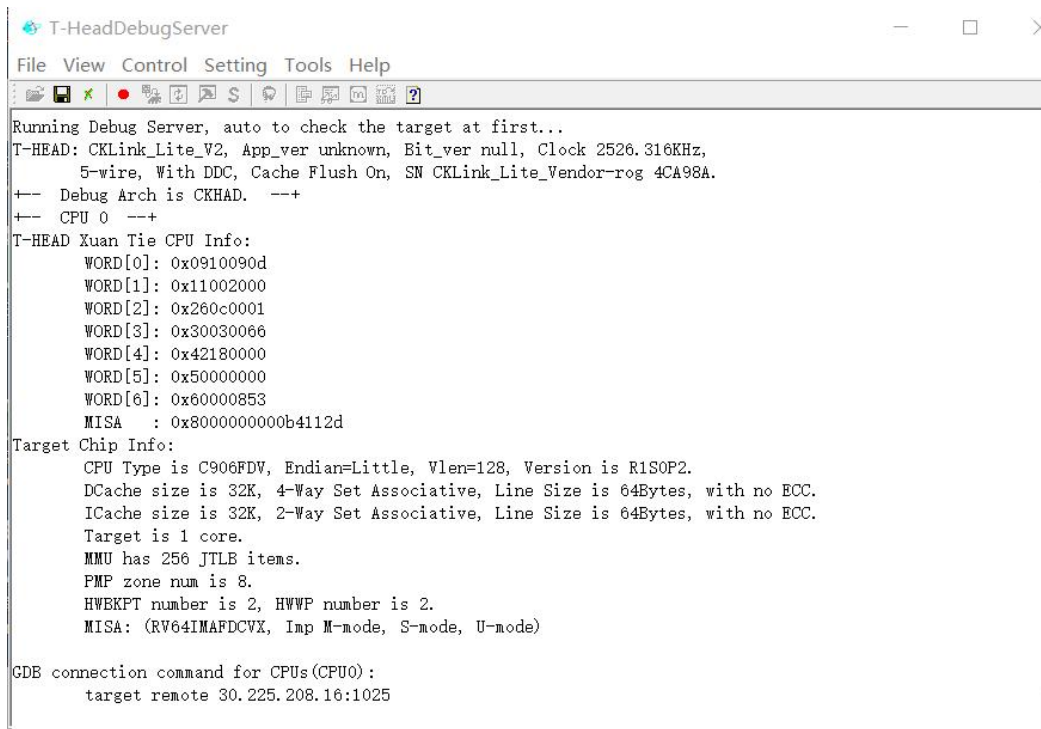
注意:

- 工具链版本需要与开发软件准备中的说明一致。
- yoctoools 版本需要与开发软件准备中的说明一致。

- 编译时需要选择 D1 的 yaml 文件。
- 编译前先执行下 “sudo service binfmt-support start” 命令。

烧写

① PC 机上启动 DebugServer，确认提示如下信息：



```
T-HeadDebugServer
File View Control Setting Tools Help
Running Debug Server, auto to check the target at first...
T-HEAD: CKLink_Lite_V2, App_ver unknown, Bit_ver null, Clock 2526.316KHz,
5-wire, With DDC, Cache Flush On, SN CKLink_Lite_Vendor-rog 4CA98A.
+- Debug Arch is CKHAD.  +-+
+- CPU 0  +-+
T-HEAD Xuan Tie CPU Info:
WORD[0]: 0x0910090d
WORD[1]: 0x11002000
WORD[2]: 0x260c0001
WORD[3]: 0x30030066
WORD[4]: 0x42180000
WORD[5]: 0x50000000
WORD[6]: 0x60000853
MISA : 0x8000000000b4112d
Target Chip Info:
CPU Type is C906FDV, Endian=Little, Vlen=128, Version is R1S0P2.
DCache size is 32K, 4-Way Set Associative, Line Size is 64Bytes, with no ECC.
ICache size is 32K, 2-Way Set Associative, Line Size is 64Bytes, with no ECC.
Target is 1 core.
MMU has 256 JTLB items.
PMP zone num is 8.
HWBKPT number is 2, HWWP number is 2.
MISA: (RV64IMAFDCVX, Imp M-mode, S-mode, U-mode)
GDB connection command for CPUs(CPU0):
target remote 30.225.208.16:1025
```

② 命令行下执行烧录命令：

```
### 烧录镜像指令
root@DESKTOP-xxx:smart_speaker_v2# make flashall
scons: Reading SConscript files ...
...
want to burn all
[2022-08-03 22:01:20] I am RISC-V
...
program 00x007e9000, 100%
burn all over!
root@DESKTOP-xxx:smart_speaker_v2#
```

注意：

- 烧写前会执行根目录下的 gdbinitflash 脚本进行 DebugServer 的连接，默认的连接是 target remote localhost:1025，如果无法连接请根据调试工具的提示修改 IP 地址和端口后重试。
- Windows+WSL 环境，请确认系统防火墙为关闭状态。

5) 例程运行

由于语音交互流程涉及云端处理，故本章节主要描述了如何使设备联网，最后描述了如何去人机交互。

a) 配置串口

① 打开串口调试工具，配置波特率为 115200，数据位 8，校验位 None，停止位 1，流控 None。

② 重新上电或按下 RST 键，系统启动，串口会有以下打印信息，表示系统运行成功。

```
[34]HELLO! BOOT0 is starting![Sep 18 2021, 11:27:51]
[39]BOOT0 commit : 3b45046
..... (省略) .....
[264]Jump to second Boot.
[266]jump to bootloader, [0x40000000]
Welcome boot2.0!
build: Feb 15 2022 15:41:15
cpu clock is 1008000000Hz
jump to [0x40040000]
j m
j 0x40040000
(cli-uart) # ###YoC###[Apr 11 2022, 11:36:32]
cpu clock is 1008000000Hz
..... (省略) .....
[ 0.320]<I>init app_init.c[80]: find 8 partitions
[ 0.330]<I>init app_init.c[51]: filesystem init ok.
```

```
[ 0.340]<D>smart_audio smart_audio.c[389]: Enter smtaudio_init
```

b) 配置 WiFi

使用下面命令配置 WiFi 名称和密码，并重启系统。

```
kv set WiFi_ssid0 {ssid}
kv set WiFi_psk0 {password}
reboot
```

注意：

WiFi 密码设置重启生效。

c) 语音交互

WiFi 连接成功后，使用“天猫精灵”语音唤醒，等提示音播放完成后说出命令词。

在线交互示例命令如下：

```
人：你好芯宝
机：咚
人：播放音乐
机：如实播放
人：你好芯宝
机：咚
人：上海天气
机：上海市今天多云转阴，2 到 8 摄氏度，东北风微风。PM2.5 指数 39。.....（省略）.....
人：音量最大
机：好的
```

d) 日志解析

```
### 上电启动
HELLO! BOOT0 is starting![Sep 18 2021, 11:27:51]
BOOT0 commit : 3b45046
```

```
..... (省略) .....
[264]Jump to second Boot.
[266]jump to bootloader, [0x40000000]
Welcome boot2.0!
build: Feb 15 2022 15:41:15
cpu clock is 1008000000Hz
jump to [0x40040000]
j m
j 0x40040000
(cli-uart) # ###YoC###[Apr 11 2022, 11:36:32]
cpu clock is 1008000000Hz
..... (省略) .....
[ 0.320]<I>init app_init.c[80]: find 8 partitions
[ 0.330]<I>init app_init.c[51]: filesystem init ok.
[ 0.340]<D>smart_audio smart_audio.c[389]: Enter smtaudio_init
..... (省略) .....

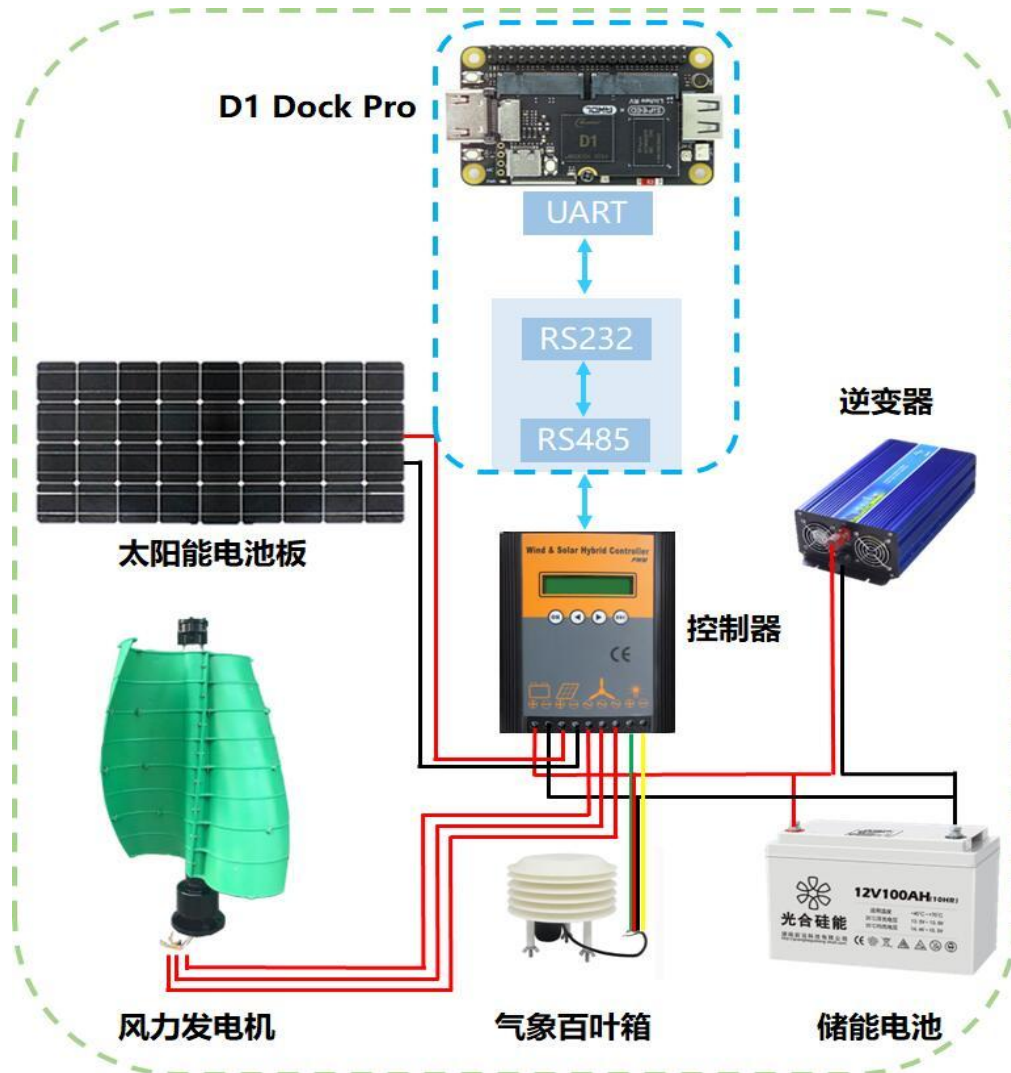
### 成功联网
<D>WiFil8723ds_devops.c[557]: @@@@ Connection Success
@@@@
<I>netmgr netmgr_service.c[187]: start dhcp
<I>netmgr netmgr_service.c[211]: IP: 192.168.1.2
<D>NTP ntp.c[200]: ntp1.aliyun.com
<D>NTP ntp.c[259]: NTP sec: 1649650295 usec: 746033
<D>NTP ntp.c[288]: sync success
```

6) 创意应用开发：分布式能源系统

能源和环保是关乎人类未来的重要课题。为实现碳中和目标，大力发展可再生清洁能源以替代传统化石能源，提高能源系统监控和消费的智能化水平，是可行的重要途径之一。

本项目以 RISC-V 架构的 D1 Dock Pro 和 D1 Nezha 开发板为硬件平台，应用物联网和区块链技术，设计开发一套分布式能源智慧管理小型示范系统，在该系统上实现能源生产和消费数据的实时监测。

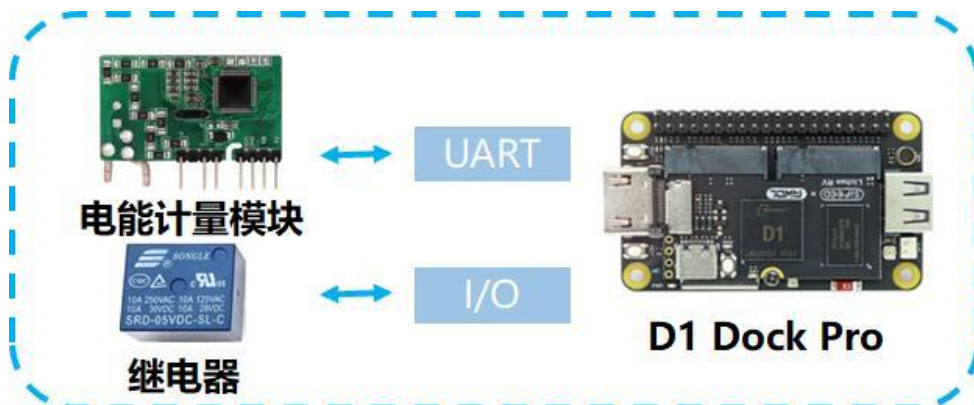
项目使用 D1DockPro 开发板设计开发一款专用网关，实时采集电池控制器、气象环境传感器等其它传感器的数据，并通过无线通信方式（WiFi）以 HTTP 协议或 MQTT 协议将传感器数据上传至物联网后台。如下所示为专用网关的示意图和实物图。

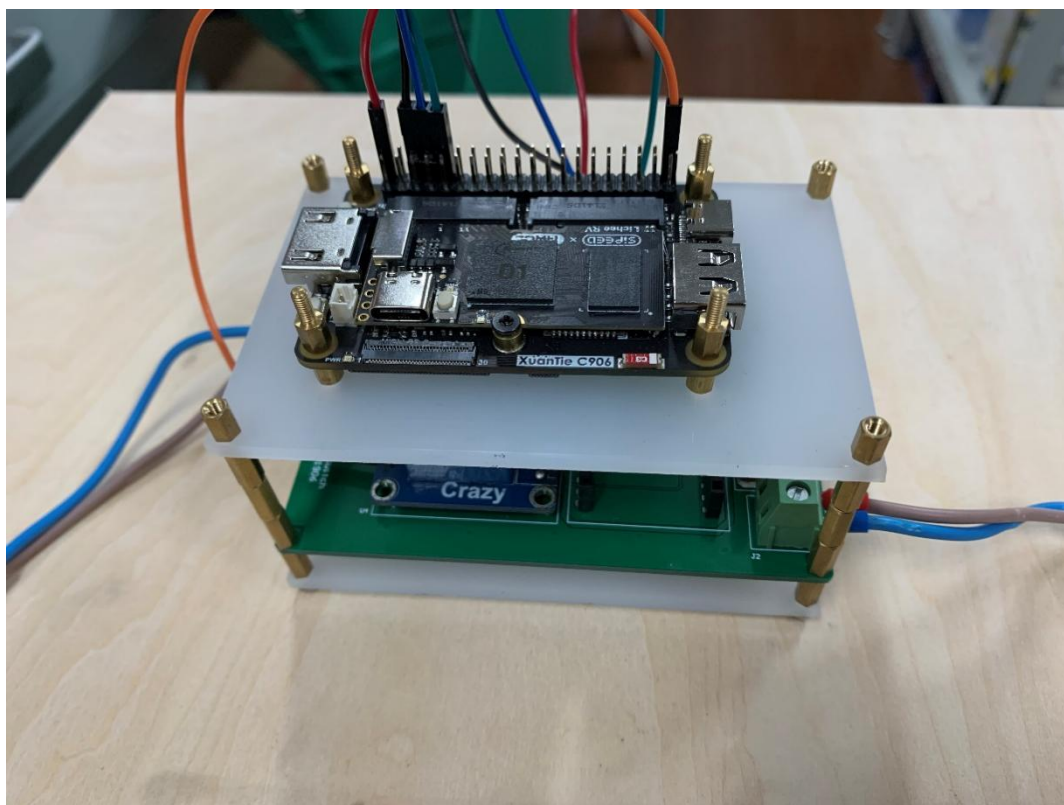




智能开关用于能源消费端，实现对能源消费者（电器负载）的供电控制、电能消费数据的采集和传输等功能。该智能开关基于 D1DockPro 开发板进行设计开发，通过开发板的 I/O 口控制继电器、UART 接收电能计量模块的数据。

设计一个扩展电路板与开发板配合使用，扩展电路板集成电能计量模块、继电器等。本文设计的智慧开关的功能主要是控制电器开关与计量电器用电参数以及环境参数并上传到云端服务器。如下所示为智能开关的示意图和实物图。





关于分布式能源系统详细的介绍信息可以进入平头哥[芯片开放社区 \(OCC\)](#) 查看。

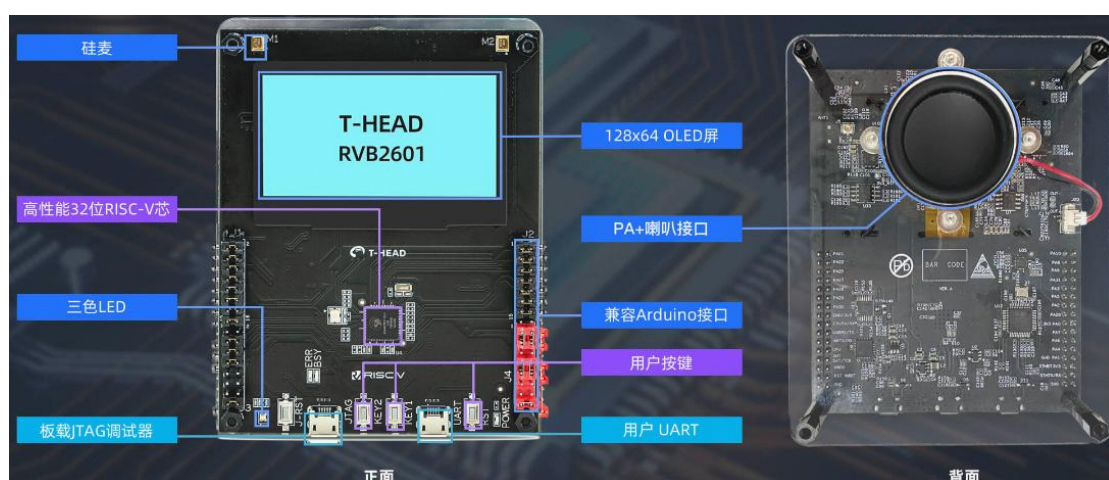
2. 基于玄铁 E906 处理器的 RVB2601 开发实践

1) 概述

CH2601 是基于玄铁 E906 的 RISC-V 生态芯片，最高主频 220MHz，支持 AliOS Things 物联网操作系统、平头哥 YoC 软件平台及平头哥剑池开发工具 (CDK)。

RVB2601 是基于平头哥生态芯片 CH2601 的开发板，板载 JTAG 调试器，WiFi&BLE 芯片 W800，音频 ADC-ES7210，音频 DAC-ES8156，128x64 OLED 屏幕，RGB 三色指示灯，用户按键，及兼容 Arduino 的扩展接口。

RVB2601 照片及资源如下图：



2) 开发板介绍

a) 规格介绍

RVB2601 包含以下功能模块：

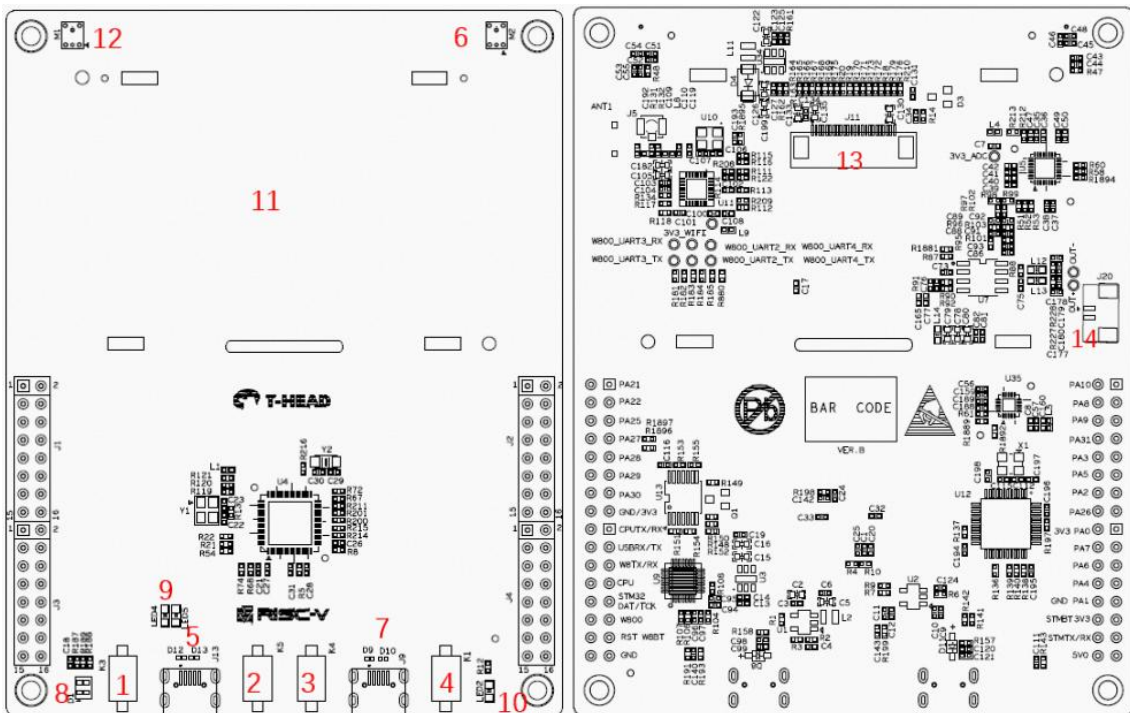
功能模块：

模块	描述
主控	CH2601，基于玄铁 E906，最高主频 220MHz
WiFi	W800，WiFi&BLE Combo 芯片，联盛德
JTAG 调试器	4 线 JTAG 调试器
USB 转串口芯片	CP2102 的 USB 转 UART 电路
音频 ADC	ES7210，顺芯
音频 DAC	ES8156，顺芯
天线	2.4G PCB 天线

OLED 屏幕	128*64 OLED 屏幕 (QG-2864ASWPG01), 智晶
喇叭	4 欧姆/2W 输出, 参考音回采
硅麦	MSM381A3729H9HP, 敏芯微
指示灯	3 路 PWM LED 灯、STM32 工作指示灯 x 2 电源指示灯 x1;
按键	一个 CH2601 的 CPU 复位按键、一个 JTAG 芯片复位按键、两个用户自定义按键;

b) 硬件结构和接口说明

RVB2601 尺寸 (L*W): 99.5 * 78.7 mm, 单板结构接口如下所示:



单板接口介绍：

序号	描述
1	JTAG 复位按键
2	用户自定义按键 1
3	用户自定义按键 2
4	CH2601 复位按键
5	USB 接口--JTAG（连接 CPU 或者 WiFi 芯片）
6	硅麦
7	USB 接口--UART（连接 CPU 或者 WiFi 芯片）
8	三色呼吸灯
9	STM32 工作指示灯
10	电源指示灯
11	OLED 屏幕
12	硅麦
13	OLED 屏幕连接器
14	喇叭连接器

为了方便开发者调试，RVB2601 提供了四个扩展接口，分别是 J1, J2, J3, J4，其信号如下：

J1		J2	
CH2601_PA10	ADC_MCLK (ES7210 的 MCLK)	W800_RST_N (W800 复位)	CH2601_PA21
CH2601_PA8	I2C_SCL (连接 ES7210 和 ES8156)	SPI_INT (W800 中断)	CH2601_PA22
CH2601_PA9	I2C_SDA (连接 ES_7210 和 ES8156)	WAKEUP (唤醒 W800)	CH2601_PA6
CH2601_PA31	PA_MUTE (喇叭静音)	SPI1_CS (连接屏幕)	CH2601_PA27
CH2601_PA3	DAC_LRCK (ES8156 的 LRCK)	SPI1_CK (连接屏幕)	CH2601_PA28
CH2601_PA5	DAC_SDIN (ES8156 的 SDIN)	SPI1_MOSI (连接屏幕)	CH2601_PA29
CH2601_PA2	DAC_SCLK (ES8156 的 SCLK)	SPI1_MISO (连接屏幕)	CH2601_PA30
CH2601_PA26	DAC_MCLK (ES8156 的 MCLK)	3V3	GND
J3		J4	
ADC_VREFP (ADC 的参考电平)	3V3	PA24_UART0_RXD (CH2601 UART)	PA23_UART0_TXD (CH2601 UART)
CH2601_PA7	LED_RED (三色灯-红灯)	CP2012_TXD	CP2012_RXD
CH2601_PA25	LED_GREEN (三色灯-绿灯)	W800_RXD (W800 UART)	W800_TXD (W800 UART)
CH2601_PA4	LED_BULE (三色灯-蓝灯)	JTAG_TCK (CH2601 JTAG)	JTAG_TMS (CH2601 JTAG)
ADC_VREFN (ADC 的参考电平)	GND	STM32_TCK	STM32_TMS
3V3	SMT32_BOOT0	W800_TCK	W800_SWO
MCU_RXD	MCU_TXD	W800_BOOT	RESET_BD
5V5	5V5	GND	GND

更多 RVB2601 开发板硬件接口及模块功能介绍,可以参考[《RVB2601 开发板用户手册》](#)。

3) 开发环境搭建

a) 开发软件准备

CDK 是平头哥面向 MCU、工控、IoT 等领域应用提供的专业集成开发环境。它以极简开发为理念,在不改变传统 MCU 应用开发习惯的基础上,全面接入云端开发资源,并结合图形化的 OStracer、Profiling 等调试分析工具,加速用户产品开发。

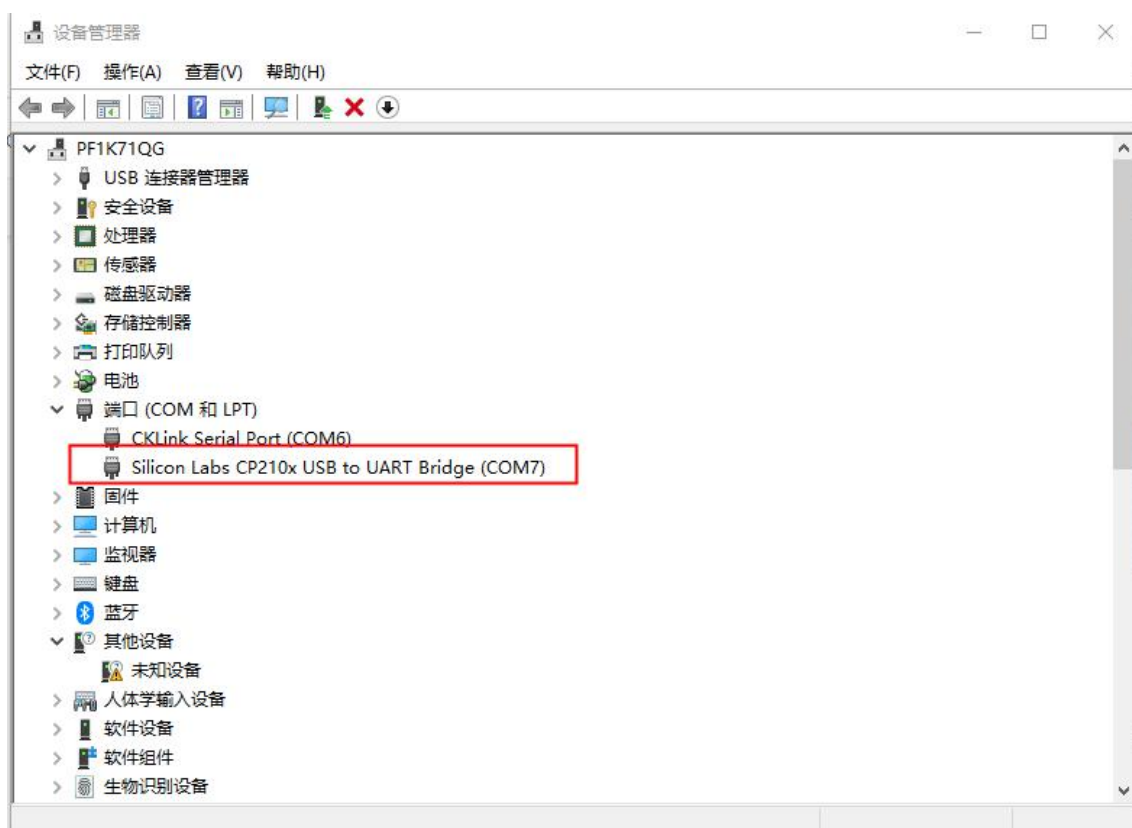
关于 CDK 的介绍，可以直接前往平头哥芯片开发社区进行查看，在资源下载处可以下载最新版剑池 CDK 软件包，解压缩直接双击 setup.exe 进行安装，按照安装提示逐步点击完整即可。

b) 串口调试环境准备

首先需要给开发板供电, RVB2601 开发板板载两个 Micro USB 口, USB 口输入电压 5V, 连接任意一个 USB 口, 即可对开发板供电。串口调试时, USB 线接用户 UART 口。

驱动安装

进入我的电脑设备管理器查看, 可以看到该串口驱动已经正确安装。如果第一次使用, 会看到显示问号。请进入芯片开放社区里资源下载进行 CP210X 串口驱动下载, 或直接点击[这里](#)下载串口驱动。



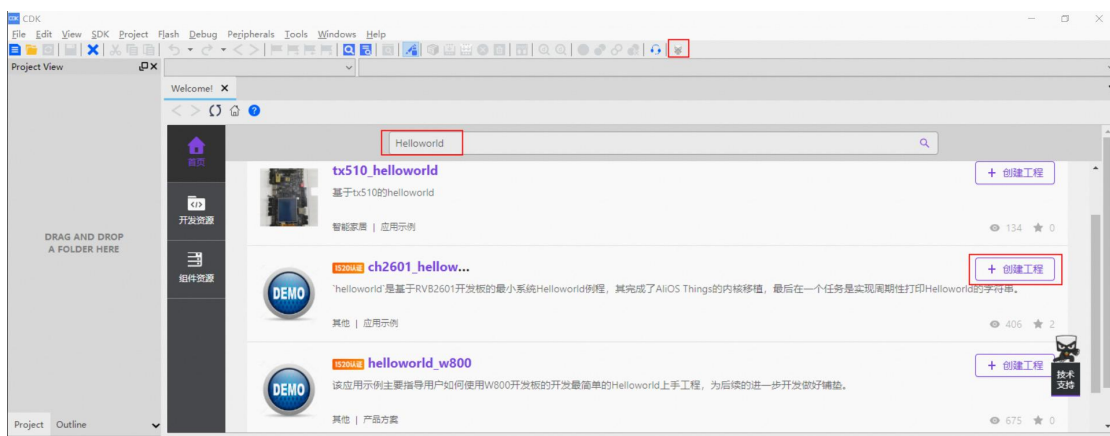
可以使用各种串口客户端软件, 选择波特率为 115200, 这里以 SSCOM 为例。



4) 例程运行

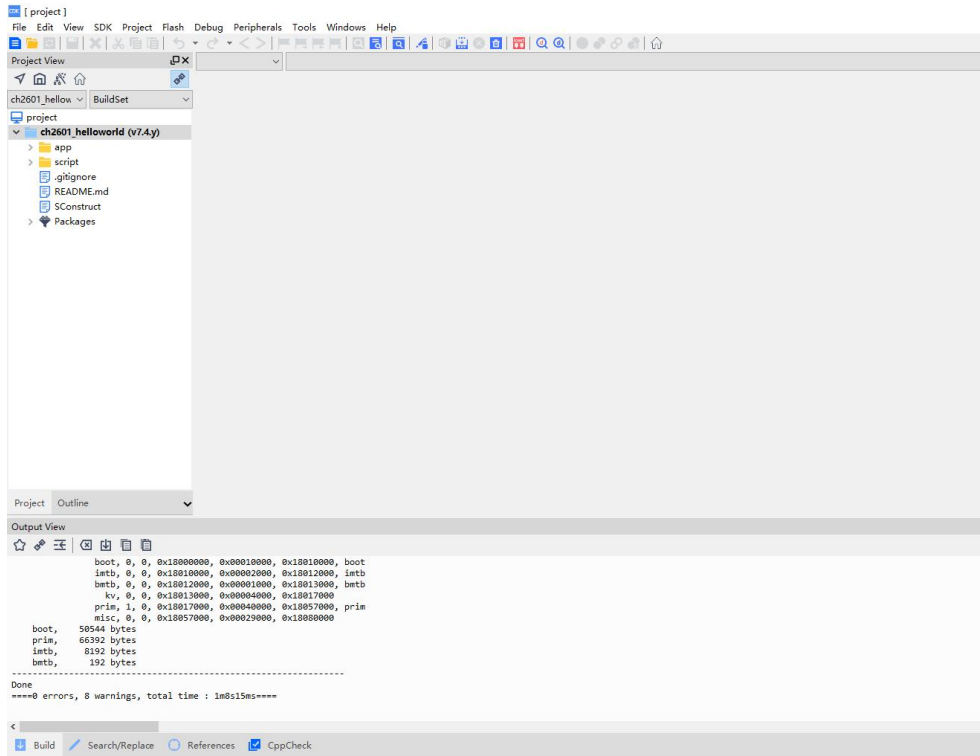
a) HelloWorld 程序获取

双击打开 CDK, 点击工具栏最右侧的平头哥图标, 在弹出来的搜索栏里输入 HelloWorld, 在结果里选择 ch2601_HelloWorld, 点击右侧的【创建工程】。



b) 编译

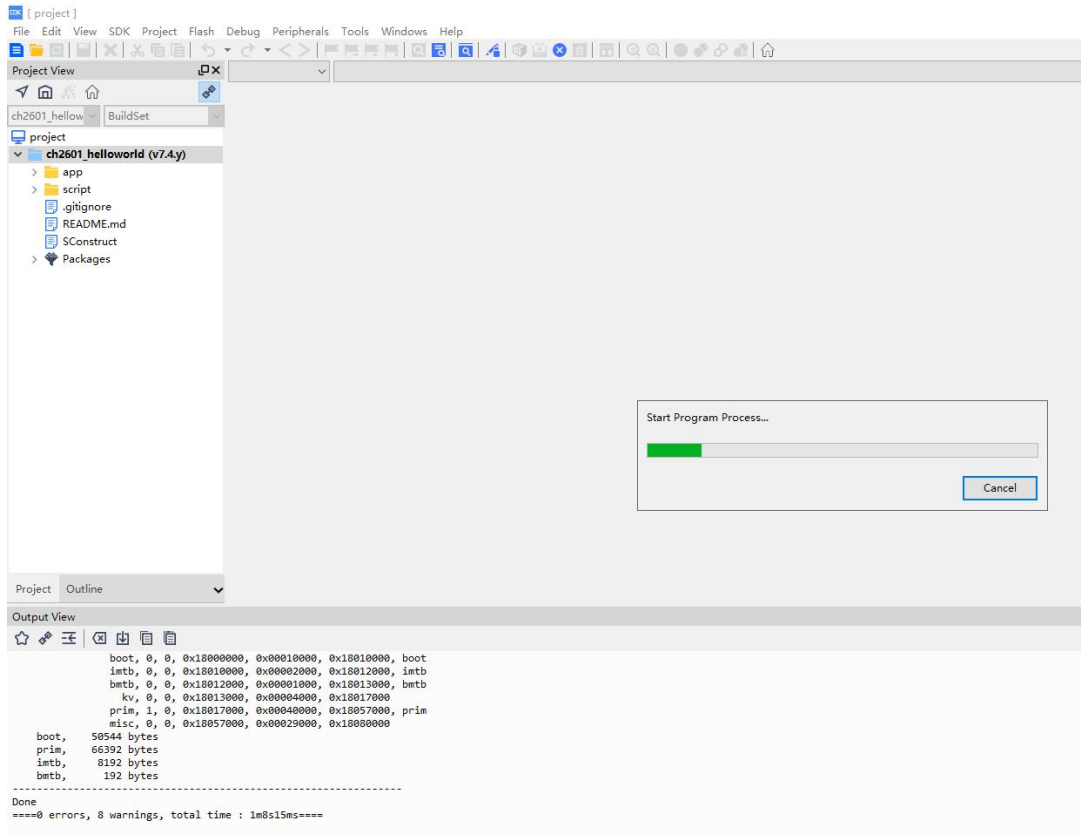
右键点击 Ch2601_HelloWorld 工程，选择 build，进行编译，直到结束。



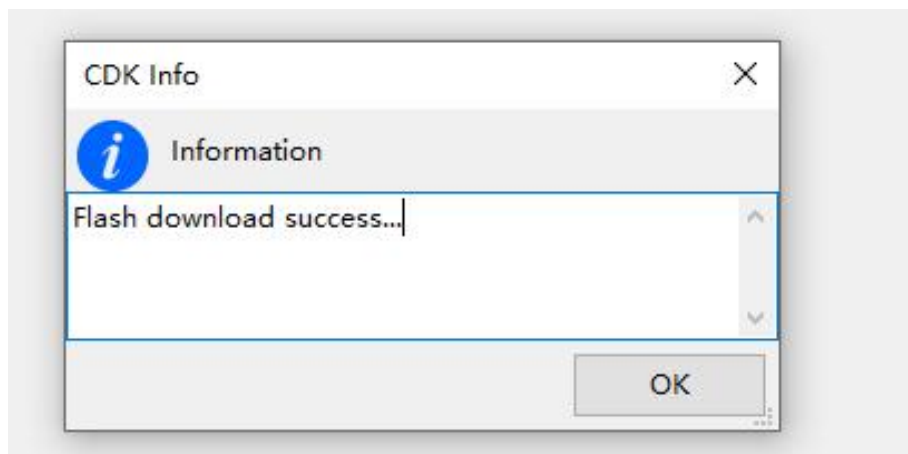
编译完成后，可以在工程目录下 obj 文件夹里找到 ch2601_HelloWorld.elf 可执行文件。

c) 下载

点击 flash 菜单，选择【Download】子菜单，进行镜像下载。



下载完成后可以看到以下对话框信息。



d) 运行

按照 RVB2601 板子上的 RST 复位键，程序会自动执行。

在串口客户端里可以看到以下打印信息，说明程序已经正确执行。



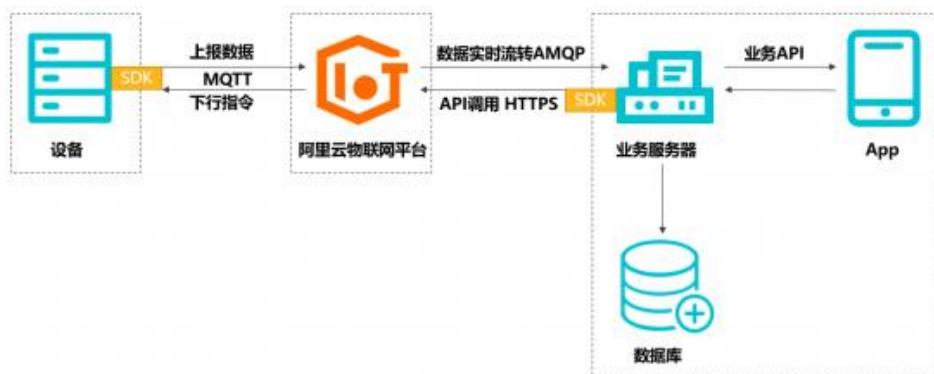
```
SSCOM 3.3
Welcome boot2.0!
build: Jan 20 2021 11:55:54
loading & jump to [prim]
load&jump 0x18017000,0x18017000,66392
xip...
j 0x18017044
[ 0.020]<I>INIT Build:Feb 24 2021,15:36:05
[ 0.020]<D>app □e5

[ 0.020]<D>app Hello world! YoC
[ 1.020]<D>app Hello world! YoC
[ 2.020]<D>app Hello world! YoC
[ 3.020]<D>app Hello world! YoC
[ 4.020]<D>app Hello world! YoC
[ 5.020]<D>app Hello world! YoC
[ 6.020]<D>app Hello world! YoC
[ 7.020]<D>app Hello world! YoC
[ 8.020]<D>app Hello world! YoC
[ 9.020]<D>app Hello world! YoC
```

5) 设备接入阿里云

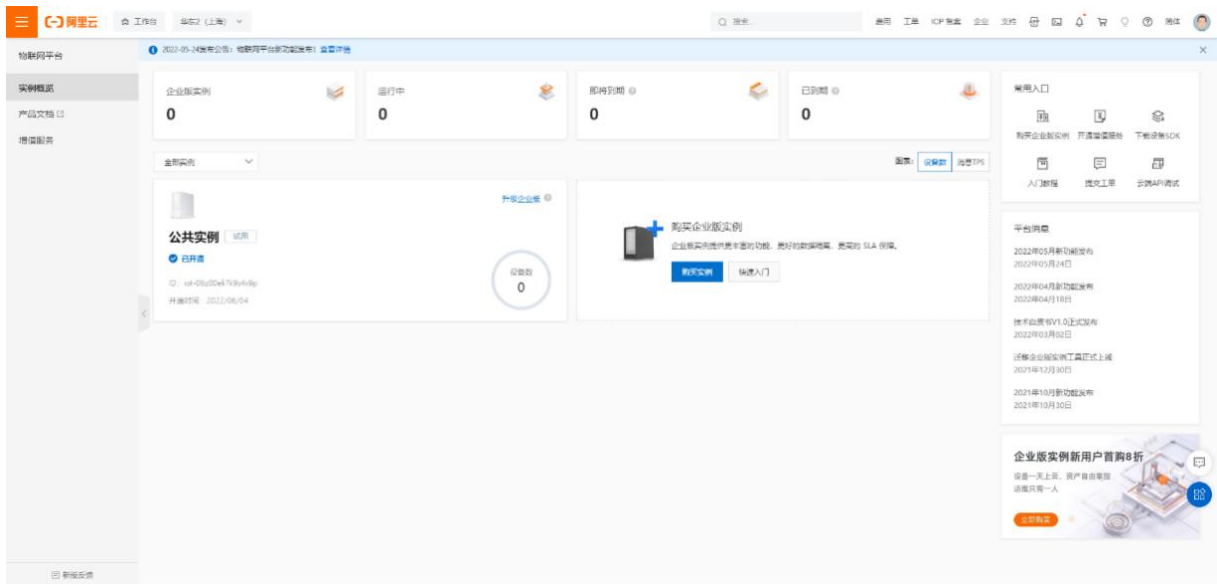
a) 物联网平台介绍

阿里云物联网平台是一个集成了设备管理、数据安全通信和消息订阅等能力的一体化平台。向下支持连接海量设备，采集设备数据上云；向上提供云端 API，服务端可通过调用云端 API 将指令下发至设备端，实现远程控制。物联网平台与设备、服务端、客户端的消息通信流程如下。

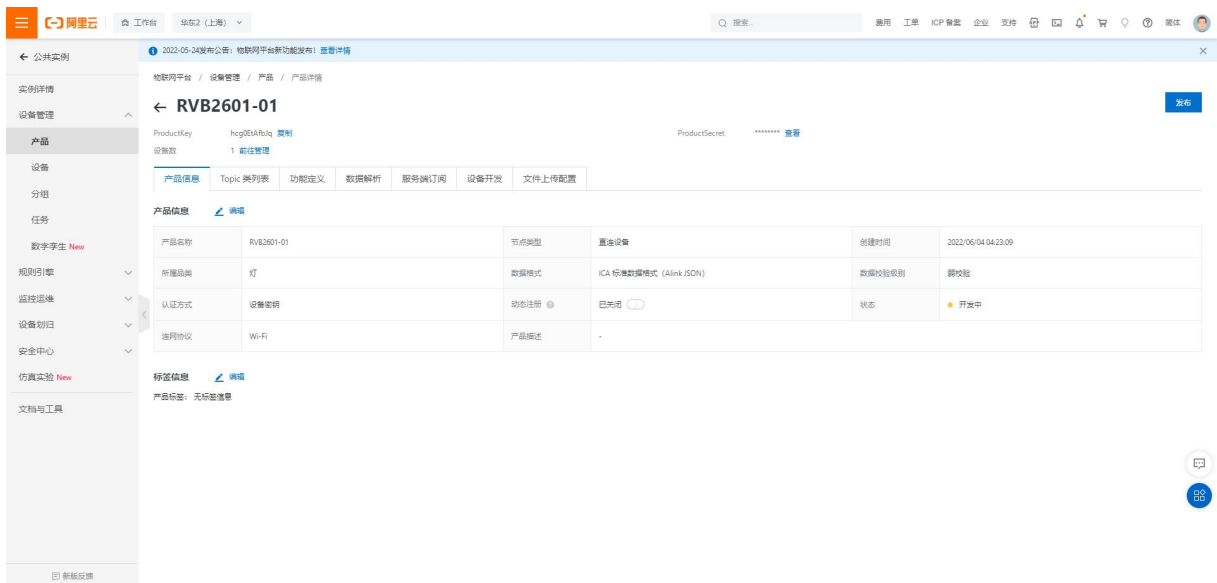


物联网平台链接：<https://iot.console.aliyun.com/lk/summary/new>

① 登录物联网平台控制台。然后在实例概览页面，单击【公共实例】。物联网平台界面如下：



② 在左侧导航栏，选择【设备管理】>【产品】，单击【创建产品】。根据自己的实际情况创建产品，例如创建一个简单的灯。



③ 创建完产品，可以在设备页面添加设备。

添加设备 ? ✕

i 特别说明: DeviceName 可以为空, 当为空时, 阿里云会颁发产品下的唯一标识符作为 DeviceName。

产品

RVB2601-01 ▼

DeviceName ?

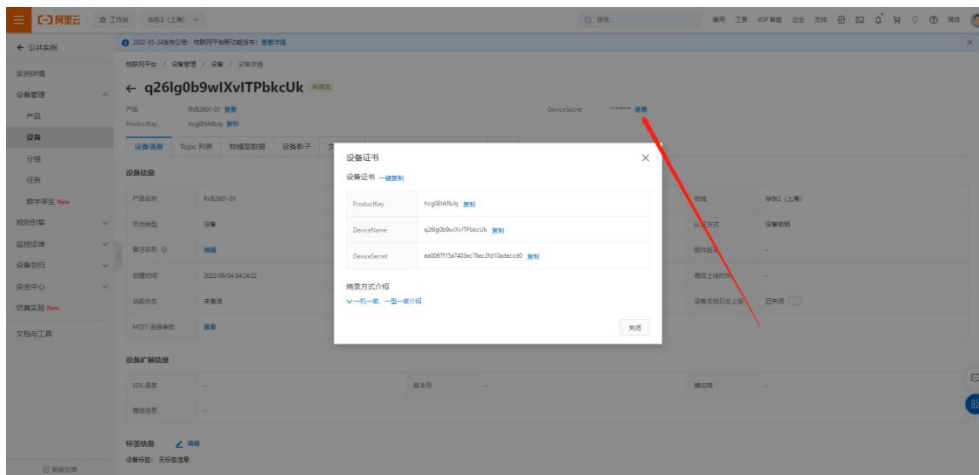
请输入 DeviceName

备注名称 ?

请输入备注名称

确认 **取消**

④ 在设备详情页面, 单击 DeviceSecret 右侧【查看】, 获取设备证书。



b) 设备端开发

跟其他功能开发一样, 下载 demo: ch2601_webplayer_demo。

下载到本地后编译，如有未下载的 SDK 库文件，右键下载。

无错误，无警告后，下载例程到开发板，按 RST 重启设备。连接串口助手，除了启动信息之外，隔一会就会打印数据 w800_apidishandle:2。

```
[1585.500]<E>w800_api dis handle: 2
[1635.510]<E>w800_api dis handle: 2
[1685.530]<E>w800_api dis handle: 2
[1735.540]<E>w800_api dis handle: 2
[1785.550]<E>w800_api dis handle: 2
[1835.570]<E>w800_api dis handle: 2
[1885.580]<E>w800_api dis handle: 2
[1935.600]<E>w800_api dis handle: 2
[1985.610]<E>w800_api dis handle: 2
[2035.630]<E>w800_api dis handle: 2
[2085.640]<E>w800_api dis handle: 2
[2135.660]<E>w800_api dis handle: 2
[2185.670]<E>w800_api dis handle: 2
[2235.690]<E>w800_api dis handle: 2
[2285.700]<E>w800_api dis handle: 2
[2335.720]<E>w800_api dis handle: 2
[2385.730]<E>w800_api dis handle: 2
[2435.750]<E>w800_api dis handle: 2
[2485.760]<E>w800_api dis handle: 2
[2535.780]<E>w800_api dis handle: 2
[2585.790]<E>w800_api dis handle: 2
[2635.810]<E>w800_api dis handle: 2
[2685.820]<E>w800_api dis handle: 2
[2735.830]<E>w800_api dis handle: 2
[2785.850]<E>w800_api dis handle: 2
```

① 首先需要给设备联网，再观察启动日志，发现这么一段代码。

```
[2.650]<I>netmgr_WiFissid{CSKY-T}, psk{test1234}
```

在工程中找到这段代码，修改相应的 WiFi 名和密码。如下，其中 123 和 12345678 分别对应 WiFi 名和密码。

```
#ifdef CONFIG_KV_SMART
    if ( aos_kv_get ( KV_WiFi_SSID , config->ssid_psk.ssid ,
&config->ssid_psk.ssid_length) < 0 ||
        aos_kv_get ( KV_WiFi_PSK , config->ssid_psk.psk ,
&config->ssid_psk.psk_length) < 0) {

        strcpy (config->ssid_psk.ssid, "123") ;
        config->ssid_psk.ssid_length = 6;

        strcpy (config->ssid_psk.psk, "12345678") ;
        config->ssid_psk.psk_length = 8;
    }
#endif
    slist_add_tail ((slist_t *) node, &netmgr->dev_list) ;

    hal_WiFi_init (node->dev) ;
}
```

② 再次编译下载、运行。在串口中输入 ifconfig 命令，发现设备已经联网成功。

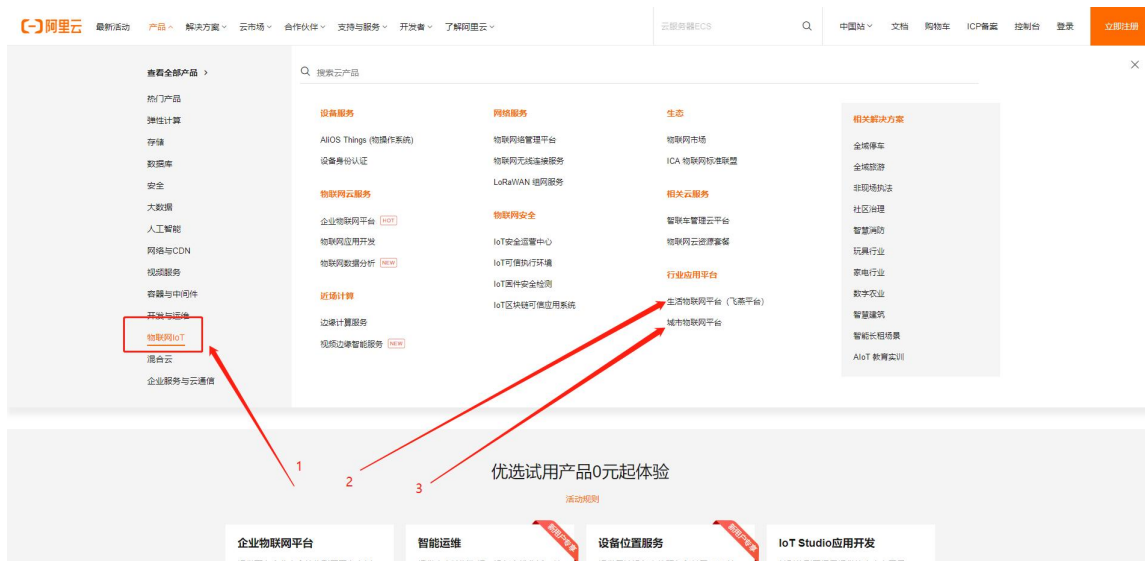
```
# ifconfig

wifi0    Link encap:WiFi  HWaddr 28:6d:cd:a0:46:a2
        inet addr:192.168.1.20
        Gwaddr:192.168.1.1
        Mask:255.255.255.0
        DNS SERVER NONE
mac: ec:f8:eb:fa:17:3f

WiFi Connected to ec:f8:eb:fa:17:3f (on wifi0)
SSID: 123
channel: 1
signal: -49 dBm

#
#
```

③ 完成联网后，下一步就是上云了，可以使用阿里云生活物联网平台



如上图，在产品中直接点物联网 IOT，右边还有两个物联网平台，这个 AT 指令集只是阿里云物联网平台的。

可以用阿里云物联网平台方式设置设备五元组，其实如果只是物联网平台，只用设置三元组。



5.1 设置设备五元组

AT+IDMAU="PRODUCT_KEY","DEVICE_NAME","DEVICE_SECRET","PRODUCT_SECRET"

AT+IDMPID=(TODO)

只需要设置一次，设置后重启。

5.2 配网AT指令

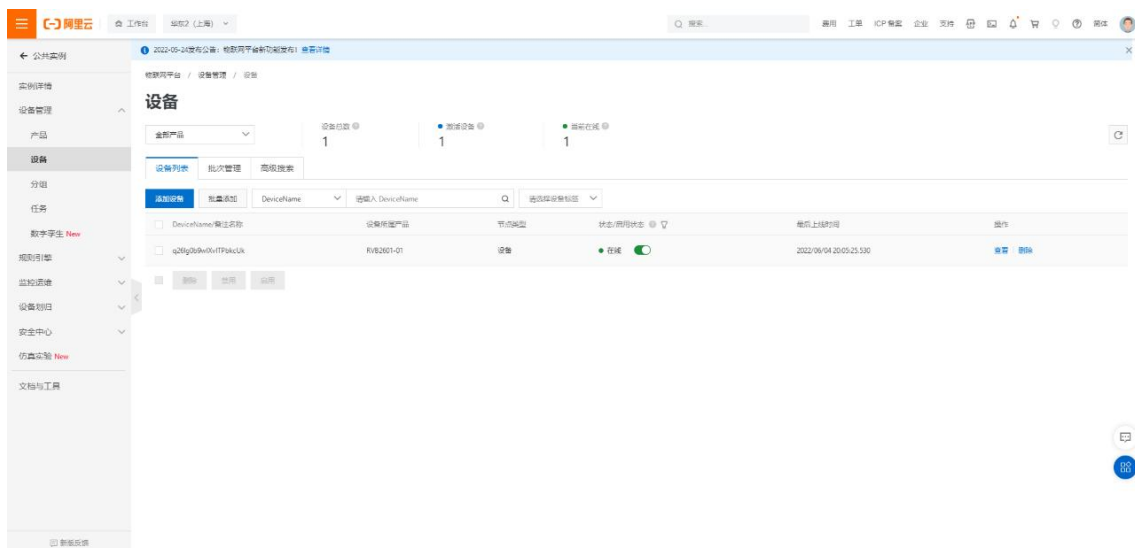
命令	描述
AT+IWSSTART	开始配网
AT+IWSSTOP	停止配网
AT+WJAP	手动配网
AT+WJAPD	删除配网信息

参考两条 AT 指令，设置设备五元组：

```
AT+IDMAU="PRODUCT_KEY", "DEVICE_NAME", "DEVICE_SECRET", "PRODUCT_SECRET"
连接+IDMCON
AT+IDMCON
```

④ 添加后编译下载，复位开发板。

进入【设备管理】->【设备】，在线了，就可以使用了。

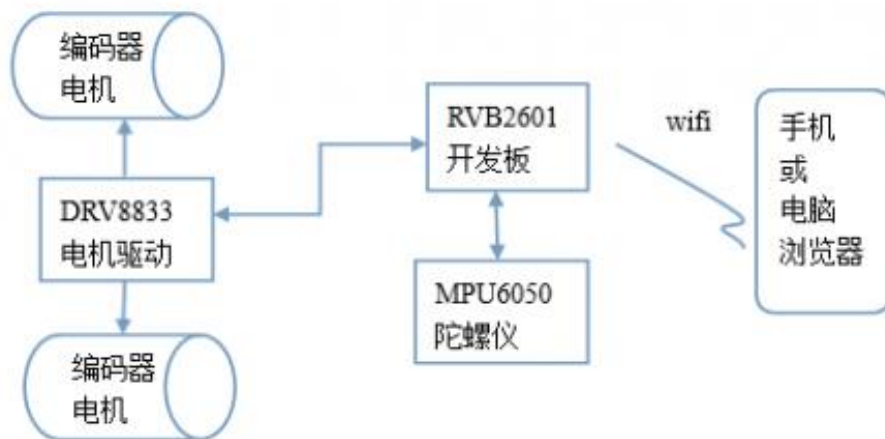


⑤ 流程已经通了，设备也能配网上云了，产品思路基本能实现，接下来就可以做些便捷开发，配网通过串口或者其他软件配置，数据能上云，云端能控制设备。

6) 创意应用开发：平衡小车

本平衡小车采用两节 18650 电池串联供电，一个 5V 的降压模块，给 RVB2601 供电，开发板上的 3.3V 用来给 MPU6050 陀螺仪模块供电，用的 IIC 与陀螺仪通信，另下还有一 DRV8833 电机驱动模块，驱动两个带编码器的电机，所以需要 4 路 PWM，每个电机还有 2 路编码器数据采集，所以一共需要 8 个 IO 口，因为开发自带了 WiFi，所以遥控部分就用 WiFi，内嵌一个小的 web 服务器，这样只要手机打开浏览器即可实现遥控平衡小车的运行。

系统框图如下：



关于平衡小车详细的介绍信息可以进入平头哥芯片开放社区（OCC）查看对应的[博文](#)和[视频](#)。

7) RVB2601 开发实践课程学习

平头哥芯片开放社区官网上线了 RISC-V 系列课程培训，该课程和清华大学、大连理工大学等合作，包括由教育部评选为“国家级一流本科课程”的《嵌入式软件设计》，成为国内嵌入式教育的标杆课程。在“中国大学 MOOC”平台已开设 9 轮课程，累计选课人数达四万人次以上，广受学生欢迎。通过 XuanTie RISC-V Institute，可以学习该课程，通过云上实验室远程操作开发板进行实战应用，了解 RVB2601 的基础功能和 DEMO 调试。

欢迎广大开发者[登录 OCC](#)了解学习。



XuanTie RISC-V Institute

基于玄铁E906处理器的RVB2601开发实践

课程目录 课程要点 课程问答

内容简介

RVB2601是基于平头哥生态芯片CH2601的开发板，板载JTAG调试器，WiFi&BLE芯片W800，音频ADCES7210，音频DACES8156，128x64 OLED屏幕，RGB三色指示灯，用户按键，及兼容Arduino的扩展接口。

云上实验室是平头哥面向广大生态开发者和企业厂商的一套云上生态测试服务。它提供软硬一体的测试环境和测试套件，帮助芯片企业检测并提高质量，同时为潜在用户提供高效快速云评估服务，帮助用户进行芯片和开发套件选型。

本章节主要向用户介绍RVB2601的基础功能和DEMO调试，以及如何使用云上实验室在线远程进行芯片选型和开发板评估。

课程资料

1. RVB2601
2. 云上实验室

RISC-V 应用领域开发示例

1. 基于 D1 Dock Pro 应用开发示例

1) 使用 LRADC 模块实现多种按键模式触发

a) 示例介绍

本示例主要使用了 chip_d1 组件驱动组件里的 LRADC。LRADC 模块属于输入设备，一般包括各种按键的实现。当用户按下按键的时候会触发中断，可以在中断子程序里面进行判断当前的按键属于哪种行为，可以支持多种按键模式的触发。涉及到的主要函数如下：

```
// 初始化 lradc
lradc_status_t csi_lradc_init (void) ;
// 反初始化 lradc
lradc_status_t csi_lradc_deinit (void) ;
// 注册回调函数
lradc_status_t csi_lradc_register_callback(lradc_callback_t callback);
```

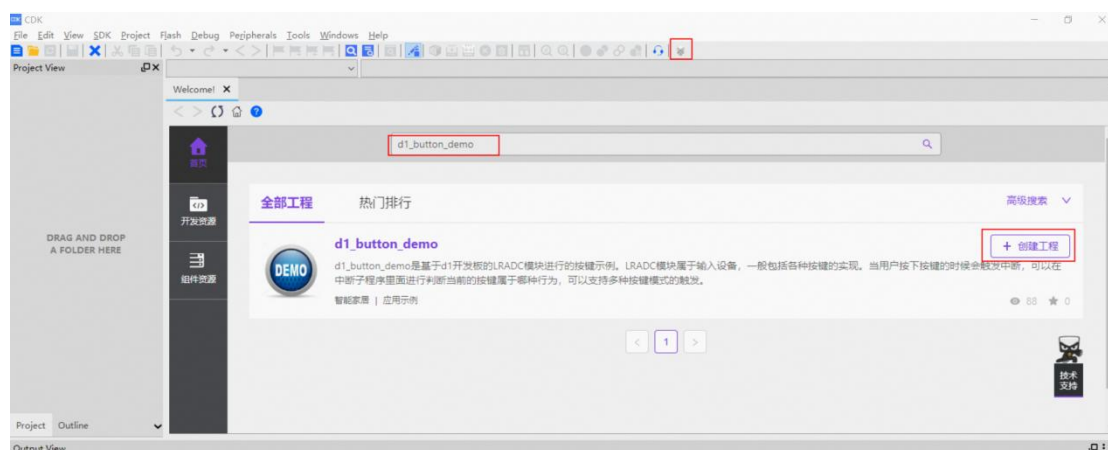
具体使用如下：

```
static void lradc_irq_callback (uint32_t irq_status, uint32_t data)
{
    // Here, we only capture key button's up IRQ to confirm user press
    key one times
    if (irq_status & LRADC_ADC0_UPPEND) {
        printf ("user key is pressed one times\n") ;
    }
}
static int test_lradc (void)
{
    int ret = -1;
    printf ("Run lradc test\n") ;
```

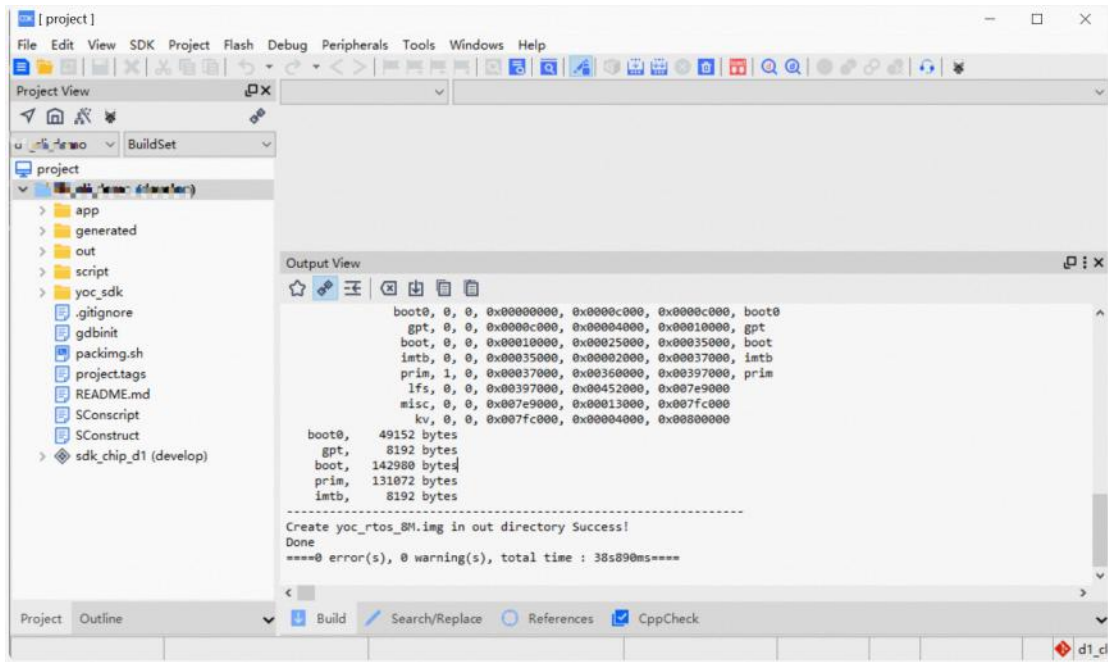
```
ret = csi_lradc_init ();  
if (ret) {  
    printf ("lradc init failed!\n");  
    return -1;  
}  
  
csi_lradc_register_callback (lradc_irq_callback) ;  
return 0;  
}
```

b) 示例演示

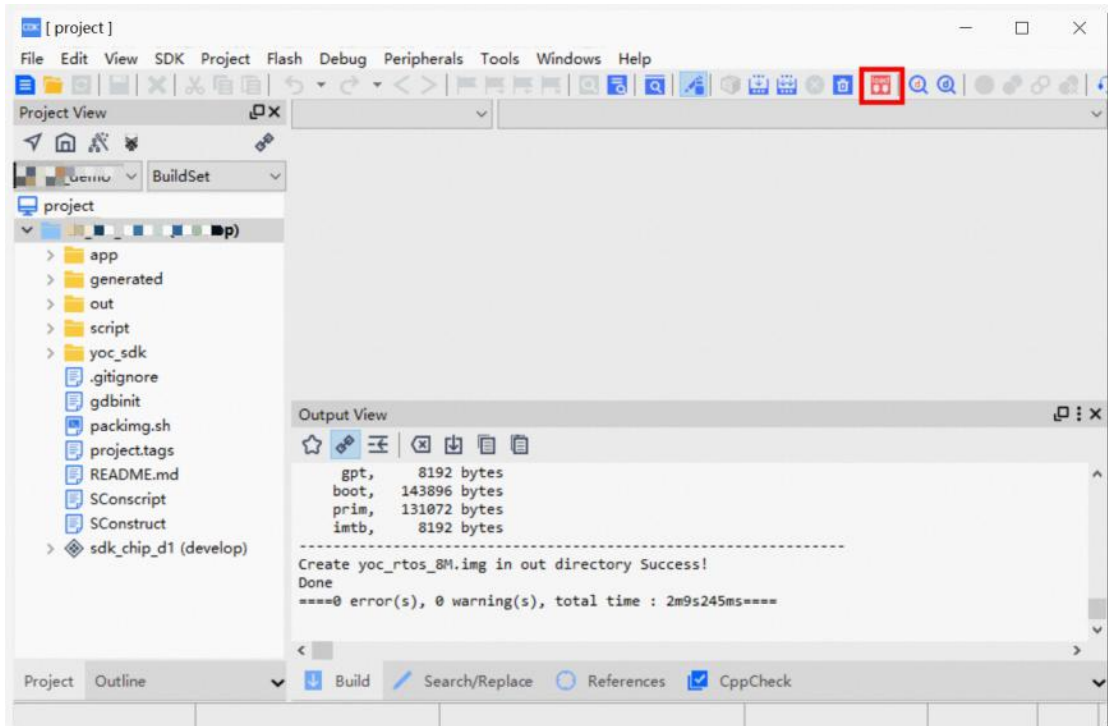
- ① 双击打开 CDK，点击工具栏最右侧的平头哥图标。
找到工程后点击【创建工程】按钮。

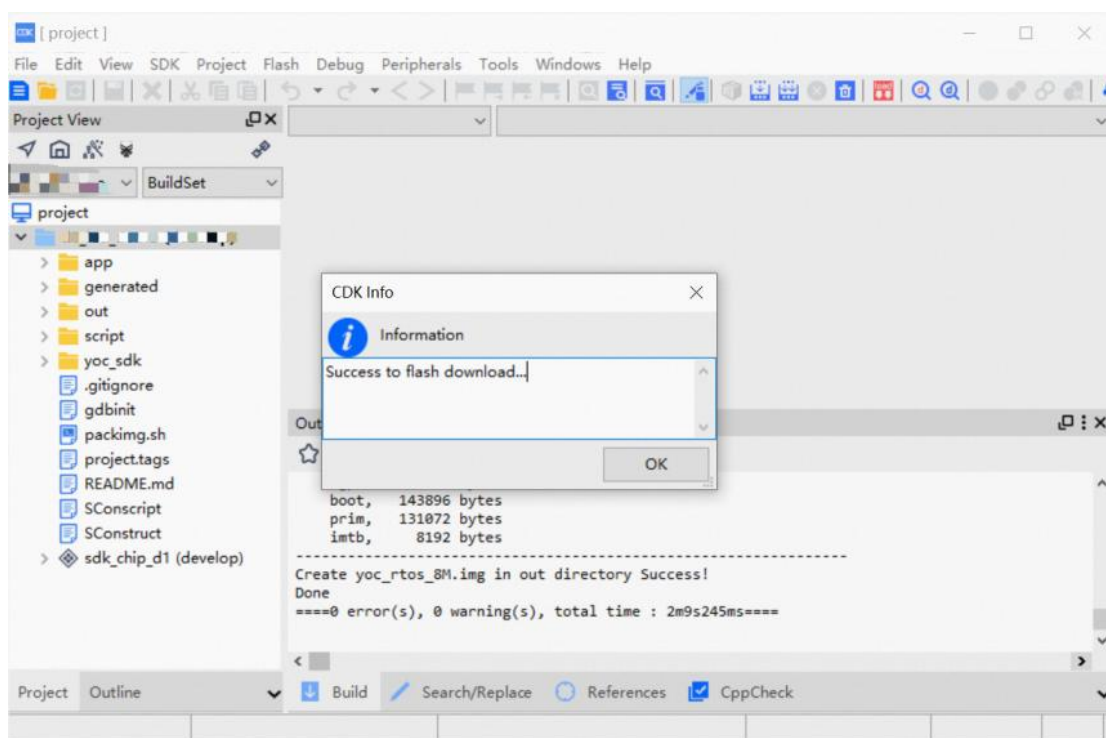


- ② 右键点击 d1_button_demo 工程，选择 build，进行编译，直到结束。



③ 先连接好硬件 JTAG。然后点击红色框处的按钮开始烧写。





烧写完毕之后，按复位键即可启动。

关于 D1 Dock Pro 开发板来实现多种按键模式的触发的详细介绍可以进入[芯片开放社区 \(OCC\)](#) 查看。

2) LCD 显示

a) 示例介绍

本示例主要使用了 chip_d1 组件驱动组件里的 mipi-dsi 驱动，用户可以参考驱动，对屏显示进行控制显示，包括屏的亮度显示等。

屏初始化

```
int csi_display_init (void)
```

屏显示控制

```
int csi_display_ioctl (disp_cmd_e cmd, void *arg)
```

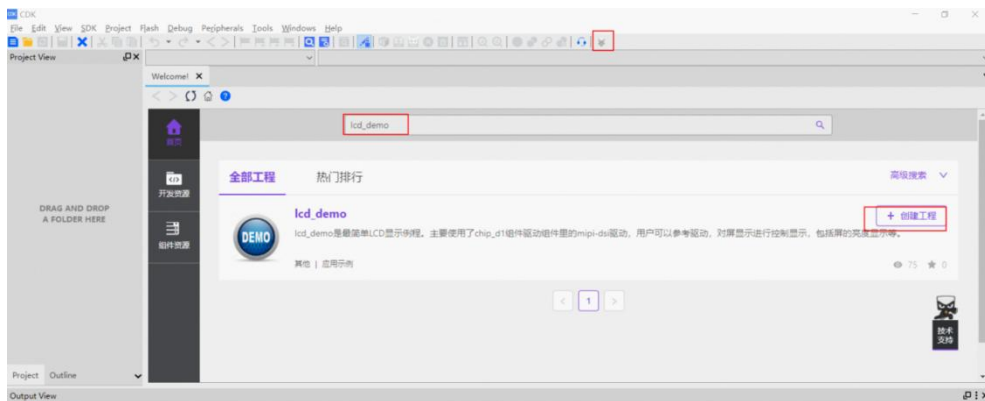
屏显示控制命令包括以下几类：

- Global interface
- Layer interface
- Capture interface
- LCD interface
- Smart backlight

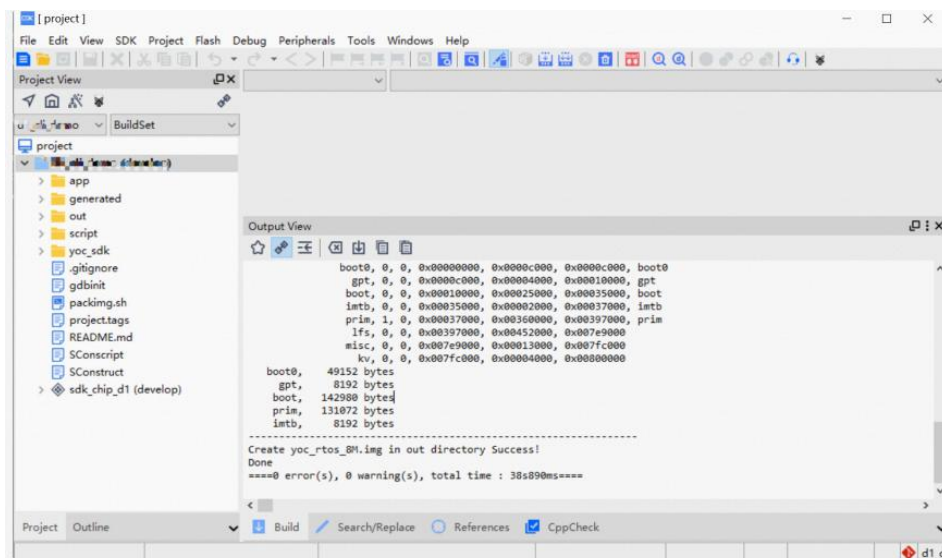
具体可以参考 sunxi_display2.h里的 tag_DISP_CMD 数据定义。

b) 示例演示

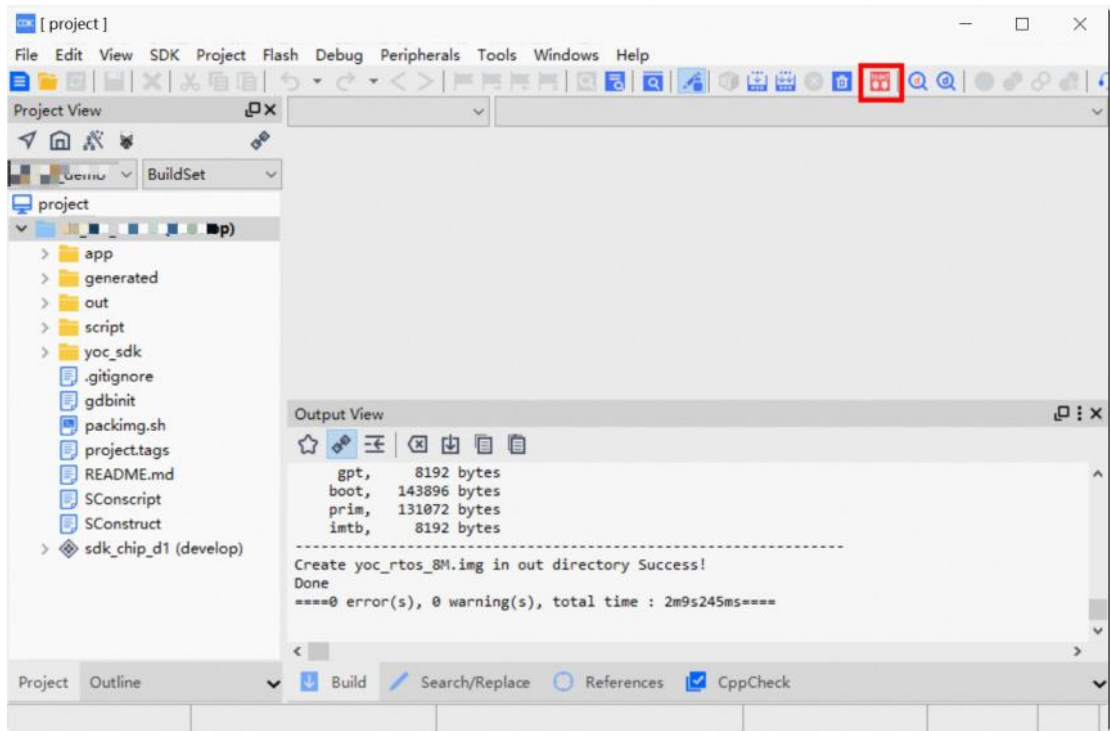
① 双击打开 CDK，点击工具栏最右侧的平头哥图标，在搜索栏里输入lcd_demo，在结果里选择 lcd_demo，点击右侧的【创建工程】，版本选择 v7.5.1。



② 右键点击 lcd_demo工程，选择 build，进行编译，直到结束。



③ 先连接好硬件 JTAG。然后点击红色框处的按钮开始烧写。



烧写完毕之后，按复位键即可启动。

④ 重新上电或按下 RST 键，系统启动，串口会有以下打印信息，表示系统运行成功。

```
[ 0.190]<I>[app]<app_task>app start.....  
[ 0.200]<I>[app]<app_task>Display screen background color testing
```

查看显示屏上会交错显示红、绿、蓝三色。



关于 D1 Dock Pro 开发板实现 LCD 屏显的详细说明可以进入[芯片开放社区 \(OCC\)](#) 查看。

2. 基于 RVB2601 应用开发示例

1) HelloWorld 最小系统快速上手

RVB2601 开发板是基于 CH2601 芯片设计的生态开发板，其具有丰富的外设功能和联网功能，可以开发设计出很多有趣的应用。为了开发者更好的了解如何在 CH2601 上开发应用，本文介绍了如何移植对接 CH2601 芯片到 YoC 最小系统，开发一个 HelloWorld 程序。

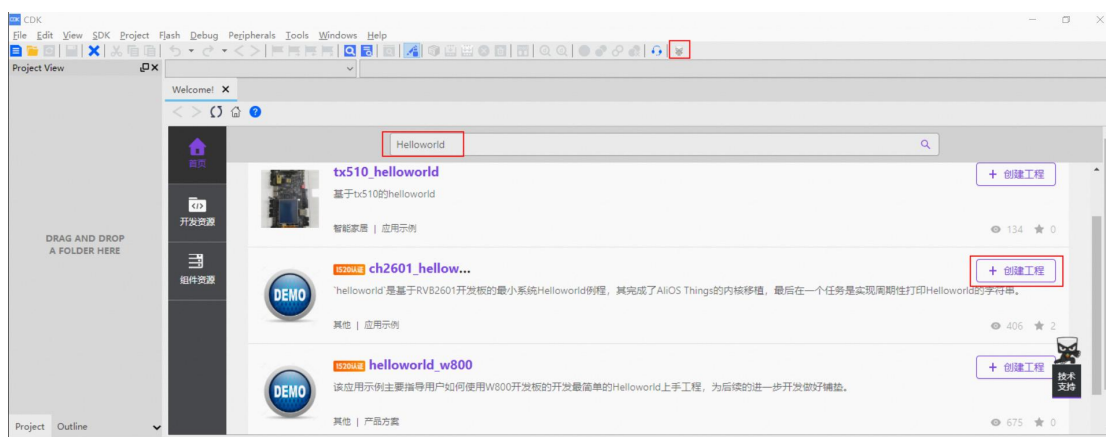
整个开发移植工作，我们都全部基于剑池 CDK 集成开发环境进行开发。剑池 CDK 以极简开发为理念，是专业为 IoT 应用开发打造的集成开发环境。它在不改变用户开发习惯的基础上，全面接入云端开发资源，结合图形化的 OStracer、Profiling 等调试分析工具，加速用户产品开发。

想要了解更多剑池 CDK 开发信息，请前往平头哥[芯片开发社区 \(OCC\)](#) 获取更多。

建议在在看本文之前，先详细看下《[RVB260 开发板快速上手教程](#)》。本例程名为 ch2601_HelloWorld_demo，可以通过 CDK 直接从 OCC 拉取。

a) 示例获取

① 打开 CDK 软件。点击工具栏最右侧的平头哥图标，搜索找到对应的工程后点击【创建工程】。



b) 开发 HelloWorld 程序

在 app/src/init/init.c里完成 board 初始化函数里完成串口的初始化。

```
void board_yoc_init ()
{
    board_init ();
    // uart_csky_register (CONSOLE_UART_IDX) ;
    console_init (CONSOLE_UART_IDX, 115200, 128) ;
    ulog_init () ;
    aos_set_log_level (AOS_LL_DEBUG) ;

    LOGI (TAG, "Build:%s, %s", __DATE__, __TIME__) ;
    board_cli_init () ;
}
```

最后在 main 函数里实现 HelloWorld 的循环打印。

```
int main (void)
{
    board_yoc_init () ;
    LOGD (TAG, "%s\n", aos_get_app_version ()) ;
    while (1) {
        LOGD (TAG, "Hello world! YoC") ;
        sample_test () ;
        aos_msleep (1000) ;
    }
```

```
}  
return 0;  
}
```

c) 编译运行

编译通过后,下载到 RVB2601 开发板后复位运行(具体下载运行操作可以参考 RVB2601 开发板快速上手教程),看到串口窗口出现一下打印,说明移植成功。



```
SSCOM 3.3  
Welcome boot2.0!  
build: Jan 20 2021 11:55:54  
load img & jump to [prim]  
load&jump 0x18017000,0x18017000,66392  
xip...  
; 0x18017044  
[ 0.020]<I>INIT Build:Feb 24 2021,15:36:05  
[ 0.020]<D>app □e5  
  
[ 0.020]<D>app Hello world! YoC  
[ 1.020]<D>app Hello world! YoC  
[ 2.020]<D>app Hello world! YoC  
[ 3.020]<D>app Hello world! YoC  
[ 4.020]<D>app Hello world! YoC  
[ 5.020]<D>app Hello world! YoC  
[ 6.020]<D>app Hello world! YoC  
[ 7.020]<D>app Hello world! YoC  
[ 8.020]<D>app Hello world! YoC  
[ 9.020]<D>app Hello world! YoC
```

关于 RVB2601 开发板实现周期性打印 HelloWorld 的字符串的详细说明可以进入[芯片开放社区 \(OCC\)](#) 查看。

2) 播放一首基于网络应用层协议的 MP3 音频

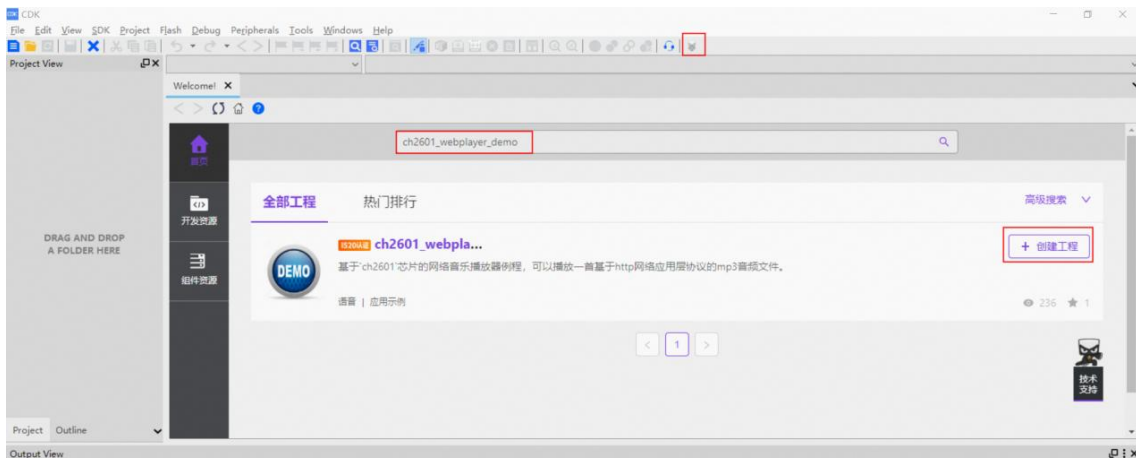
本例程基于 YoC 软件平台 av 组件采用http 协议播放一首网络 mp3 歌曲。

当开发板成功通过 sal (底层通过 at 指令连接内置的网卡芯片) 连接网络后,可输入相应串口命令行从 web 服务器上拉取 mp3 歌曲实现边拉取音频源数据边播放的功能。开

发者可基于该例程实现更为丰富的网络播放功能。本例程名为 ch2601_webplayer_demo，可以通过 CDK 直接从 OCC 拉取。

a) 下载代码并编译运行

① 打开 CDK 软件，点击工具栏最右侧的平头哥图标，搜索找到对应的工程后点击【创建工程】。



② 在 IDE 上编译通过后，点击下载进行烧录。烧录成功后，复位运行。成功运行后，串口会有如何打印输出：

```
Welcome boot2.0!
build: Mar  2 2021 19:16:27
load img & jump to [prim]
load&jump 0x18017000,0x18017000,302692
xip...
j 0x18017044
[ 0.020]<I>init find 5 partitions
[ 0.580]<D>w800_at #####spi speed:1024000

[ 0.600]<D>atempo regist affecter, name = atempo_sonic
[ 0.610]<I>netmgr start wifi
[ 0.610]<I>player player_new, 161 enter.
[ 0.620]<I>player player_new, 201 leave. player = 20009DD0
[ 0.620]<I>player player_play, 569 enter. player = 20009DD0
[ 0.630]<D>demux_mp3 mp3 probe, sync_cnt = 0, rc = 0, score = 100
[ 0.630]<I>player player_play, 589 leave. player = 20009DD0
[ 0.640]<I>demux find a demux, name = mp3, url = mem://addr=403018456&size=23220
[ 0.650]<D>avparser find a parser, name = mp3, id = 1
[ 0.650]<I>ad find a decode, name = pvmp3dec, id = 1
[ 0.660]<D>filter_vol open a avfilter, name = vol
[ 0.660]<D>ao_alsa ao open
[ 0.670]<D>ao ao ref: openref = 1, startref = 0, fun = __ao_open
[ 0.670]<D>ao ori sf ==> sf = 32768146, rate = 16000, ch = 1, bits = 16, siged = 1, float = 0, endian = 0
[ 0.680]<D>ao ao sf ==> sf = 32768146, rate = 16000, ch = 1, bits = 16, siged = 1, float = 0, endian = 0
[ 0.950]<D>ao ao ref: openref = 1, startref = 1, fun = __ao_start
[ 0.950]<D>player_demo =====player_event, 24, type = 2
[ 0.960]<D>player player_get_media_info, 809 enter. player = 20009DD0
[ 0.960]<D>player player_get_media_info, 821 leave. player = 20009DD0
[ 0.970]<D>player_demo =====rc = 0, duration = 2902ms, bps = 64011, size = 23220
[ 1.010]<I>player first frame output
[ 2.140]<D>w800 api WIFI SOFT RESET OK
```

b) 网络连接

通过 `ifconfig` 命令可配置需要连接的热点。具体命令为：

```
ifconfig ap WiFi_ssid WiFi_psk
```

热点配置成功后，会有下图如下打印：

```
#
# ifconfig ap yin 88888888
apconfig ssid:yin, psw:88888888
[ 468.0
5# 0]<I>netmgr start wifi
[ 468.570]<D>w800_api WIFI SOFT RESET OK

[ 468.610]<I>netmgr_wifi ssid{yin}, psk{88888888}

[ 470.870]<I>netmgr IP: 192.168.43.45
[ 470.870]<D>APP net got ip
```

c) 命令行播放控制

可通过在串口下输入如下命令来控制歌曲的播放

```
# player help
player play welcom/url[http://] #播放内置开机音频或网络歌曲
player pause #暂停播放
player resume #恢复播放
player stop #停止播放
player help #播放器帮助命令
```

播放 http 歌曲：

```
player play http://yocbook.oss-cn-hangzhou.aliyuncs.com/av_repo/alibaba.mp3,
```

示例如下：

```
player play
http://yocbook.oss-cn-hangzhou.aliyuncs.com/av_repo/alibaba.mp3
# [ 13.620]<E>w800_api domain to ip: 47.110.23.146
[ 13.630]<D>sals remote_port -- : 80
[ 13.710]<D>WEB http request:
GET /av_repo/alibaba.mp3 HTTP/1.0
Host: yocbook.oss-cn-hangzhou.aliyuncs.com
User-Agent: CSKY/YOC
[ 15.000]<D>stream upto cache threshold2, pos = 553, cache_pos =
809, diff = 256
[ 15.420]<D>avparser find a parser, name = mp3, id = 1
[ 15.440]<D>ad find a decode, name = pvmp3dec, id = 1
[ 15.450]<D>filter_swr open a avfilter, name = swr
[ 15.470]<D>filter_vol open a avfilter, name = vol
[ 15.470]<D>ao_alsa ao open
[ 15.490]<D>ao ao ref: openref = 1, startref = 0, fun = __ao_open
[ 15.510]<D>ao ori sf ==>sf = 90317074, rate = 44100, ch = 2, bits = 16
, signed = 1, float = 0, endian = 0
[ 15.540]<D>ao ao sf ==>sf = 90316946, rate = 44100, ch = 1, bits = 16
, signed = 1, float = 0, endian = 0
[ 15.810]<D>ao ao ref: openref = 1, startref = 1, fun = __ao_start
[ 15.820]<D>player_demo =====_player_event, 24, type = 2
[ 15.820]<D>player player_get_media_info, 809 enter. player = 20009E00
[ 15.830]<D>player player_get_media_info, 821 leave. player = 20009E00
[ 15.830]<D>player_demo =====rc = 0, duration = 415807ms, bps = 64000,
si
ze = 3326462
```

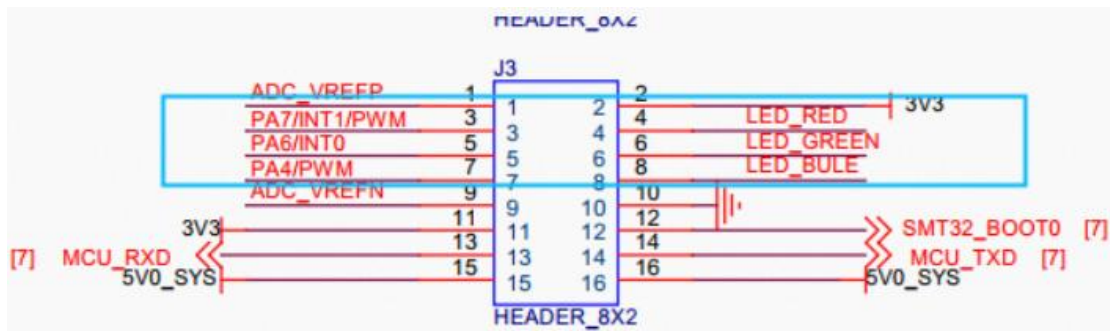
关于 RVB2601 开发板播放一首基于 http 网络应用层协议的 mp3 音频文件详细介绍，可以进入[芯片开放社区 \(OCC\)](#) 查看。

3) 控制一颗 RGB LED 灯珠实现跑马灯效果

基于 RVB2601 的跑马灯程序是利用通过控制一颗 RGB LED 灯珠实现三色跑马灯效果。本例程名为 ch2601_marquee_demo，可以通过 CDK 直接从 OCC 拉取。

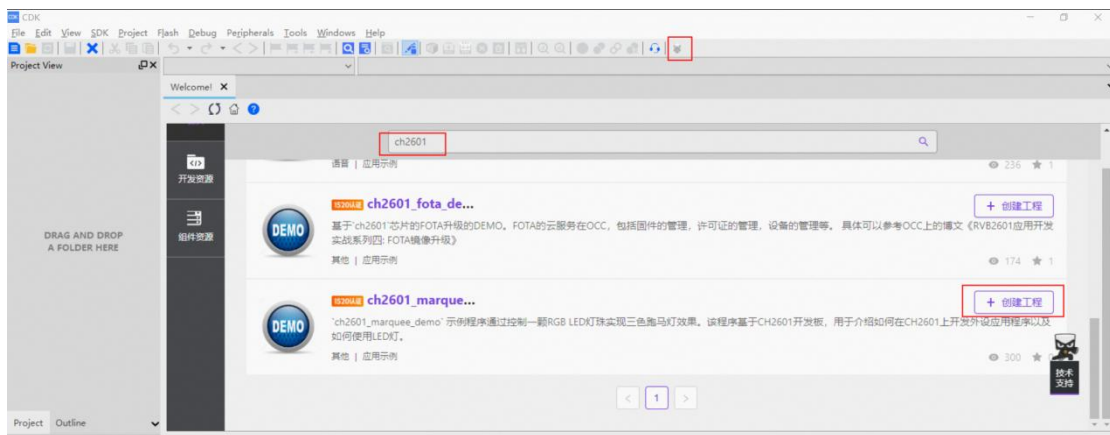
a) 硬件配置

三色LED 通过 PIN 引脚 PA7、PA25、PA4 与 CH2601 主芯片GPIO 控制器连通，主控可以通过对 PA7、PA25 和 PA4 的 GPIO PIN 的操作，可以实现三色LED 不同颜色的闪烁。LED 硬件原理图如下：



b) 跑马灯开发

① 打开 CDK 软件。点击工具栏最右侧的平头哥图标，搜索找到对应的工程后点击【创建工程】。



② LED 配置。

led_pinmux_init。该功能函数位于 app/src/led.c，用于初始化三色LED 灯的 IO 管脚。配置三色灯的 PA7、PA25 和 PA4 为 IO 模式，为每个 IO 管脚配置 PWM 通道以及占空比时间。

```
void led_pinmux_init ()
{
```



```
//7
csi_pin_set_mux (PA7, PA7_PWM_CH7) ;
csi_pin_set_mux (PA25, PA25_PWM_CH2) ;
csi_pin_set_mux (PA4, PA4_PWM_CH4) ;
ret = csi_pwm_init (&r, 0) ;
ret = csi_pwm_out_config (&r, 7 / 2, 300, 100, PWM_POLARITY_HIGH) ;
ret = csi_pwm_out_start (&r, 7 / 2) ;
//25
ret = csi_pwm_out_config (&r, 2 / 2, 300, 100, PWM_POLARITY_HIGH) ;
ret = csi_pwm_out_start (&r, 2 / 2) ;
//4
ret = csi_pwm_out_config (&r, 4 / 2, 300, 100, PWM_POLARITY_HIGH) ;
ret = csi_pwm_out_start (&r, 4 / 2) ;
}
```

led_refresh。该功能函数位于 app/src/led.c，用于依次闪烁 LED。利用 g_ctr 计数器来判断闪烁 LED 灯。通过操作三根 IO 管脚来确定显示指定颜色的灯。

```
void led_refresh ()
{
    g_ctr++;
    if (g_ctr == 3) {
        g_ctr = 0;
    }
    if (g_ctr == 0)
    {
        csi_pwm_out_start (&r, 7 / 2) ;
        csi_pwm_out_stop (&r, 2 / 2) ;
        csi_pwm_out_stop (&r, 4 / 2) ;
    }
    else if (g_ctr == 1)
    {
        csi_pwm_out_start (&r, 2 / 2) ;
        csi_pwm_out_stop (&r, 7 / 2) ;
        csi_pwm_out_stop (&r, 4 / 2) ;
    }
    else //2
```

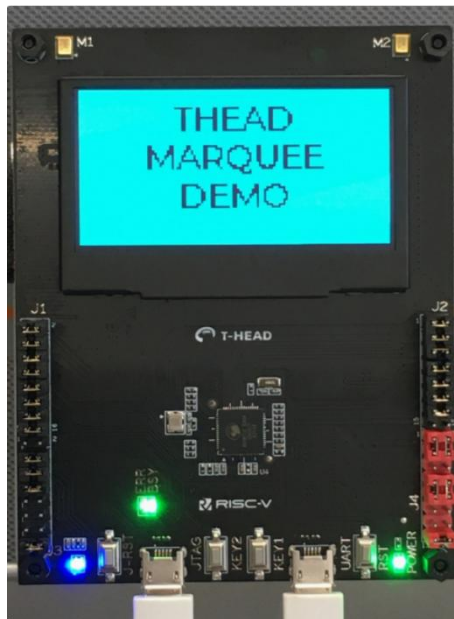
```
{
csi_pwm_out_start (&r, 4 / 2) ;
csi_pwm_out_stop (&r, 7 / 2) ;
csi_pwm_out_stop (&r, 2 / 2) ;
}
}
```

demo_task。该功能函数位于 app/src/main.c，用于周期性点亮 LED 灯，在主函数里调用。

```
static void demo_task (void *arg)
{
lv_init () ;
oled_init () ;
lable_test () ;
led_pinmux_init () ;
while (1)
{
lv_task_handler () ;
udelay (1000 * 1000) ;
lv_tick_inc (1) ;
led_refresh () ;
}
}
```

③ 编译运行

编译通过后，点击下载成功，复位运行。可看左下角的灯依次闪烁。



关于 RVB2601 开发板控制一颗 RGB LED 灯珠实现跑马灯效果的详细说明可以进入[芯片开放社区 \(OCC\)](#) 查看。

RISC-V 未来探索

1. 平头哥开源 RISC-V 系统处理器



在 2021 云栖大会上，阿里云智能总裁张建锋宣布，平头哥开源玄铁 RISC-V 系列处理器，并开放系列工具及系统软件。这是系列处理器与基础软件的首次全栈开源，将推动 RISC-V 架构走向成熟，帮助 RISC-V 软硬件技术加速融合发展，推动创新落地。

AIoT 时代，RISC-V 架构因其开放、灵活的特性，有望成为继 Intel X86、ARM 后的下一代广泛应用的 CPU 架构。但是，当前 RISC-V 架构面临应用碎片化、开发效率低、软硬件适配难等问题，软硬件生态尚未成熟。

玄铁 RISC-V 系列处理器采用自研技术，覆盖从低功耗到高性能的各类场景，支持 AliOS、FreeRTOS、RT-Thread、Linux、Android 等操作系统，并已成功应用于微控制器、工业控制、智能家电、智能电网、图像处理、人工智能、多媒体和汽车电子等领域。

不久前，玄铁 910 全球首次实现兼容安卓，极大拓展了 RISC-V 架构面向开放生态的想象力。

目前，玄铁系列处理器出货超 25 亿颗，拥有 150 余家客户、超 500 个授权数，已成为国内应用规模最大的国产 CPU。

此次开源的玄铁系列 RISC-V 处理器，包括玄铁 E902、E906、C906、C910 等 4 款量产处理器 IP，以及基于玄铁的多操作系统的全栈软件及工具。开发者可通过平头哥 Github 和芯片开放社区（Open Chip Community）下载玄铁源代码，在此基础上，实现开源 EDA 协同，创新硬件架构，丰富软件应用生态。

张建锋表示，“经过 3 年努力，我们今天又发布玄铁 RISC-V 系列开源版。未来，我们将开源更多 RISC-V 处理器，也希望有更多的合作伙伴，在玄铁基础上研发出更多有价值的 IP 核和基础软件，共同构建开放、透明和普惠的 RISC-V 生态。”

2. 平头哥对 RISC-V 基金会贡献

平头哥在 RISC-V 基金会 8 个技术委员会，29 个技术小组参与技术讨论、管理与方案制定。在 11 个技术小组担任主席或副主席职位，包含 Managed Runtime SIG, Android SIG, RVM-CSI TG 等，领导技术讨论，推动国际标准建立，贡献软硬件技术成果。在标准建设、指令集、软件、安全等多个技术方向，影响并共同推动 RVISC-V 发展。平头哥是国际基金会的董事会成员，同时是当前国内在基金会投入力量最大的组织。

3. 高校合作

2022 年平头哥芯片开放社区官网上线了 RISC-V 系列课程培训，该课程和清华大学、大连理工大学等合作，包括由教育部评选为“国家级一流本科课程”的《嵌入式软件设计》，成为国内嵌入式教育的标杆课程。在“中国大学 MOOC”平台已开设 9 轮课程，累计选课人数达四万人次以上，广受学生欢迎。通过 XuanTie RISC-V Institute，可以学习该课程，通过云上实验室远程操作开发板进行实战应用。

XuanTie RISC-V Institute 旨在打造专业开放的 RISC-V 学习交流平台。欢迎广大开发者[登录 OCC](#)了解学习。

XuanTie RISC-V Institute

为全球开发者提供RISC-V前沿技术、精品课程、最佳实践

课程

RISC-V课程考试

初识RISC-V处理器

4阶段 · 6课时 · 122人已学

加入学习

- RISC-V指令集
2课时
- RV32i基础指令集介绍
2课时
- RISC-V生态
1课时
- RISC-V软件生态发展
1课时

玄铁RISC-V处理器

5阶段 · 5课时 · 79人已学

加入学习

- E906的处理器简介
1课时
- E906的编程模型
1课时
- E906的指令集
1课时
- E906的异常和中断系统
1课时
- E906的存储和总线、流水线
1课时

感谢本电子书共同作者

(排名不分先后)

雲海 甲一 梦长 和球 傑枫 子丰 猛汉
恒伊 长卿 荣放 清欢 炎赫 少川 齐捷

素材整理

郑姜芬 琛琛

出品方

平头哥芯片开放社区 (OCC)

联合出品方

阿里云开发者社区