

嵌入式实时虚拟机ZVM关键核心技术进展介绍

汇报人：熊程来，openEuler SIG-Zephyr Maintainer

湖南大学

嵌入式与网络计算湖南省重点实验室谢国琪教授团队

2023-12-16

嵌入式实时虚拟机ZVM关键核心技术进展介绍

- ZVM总体系统架构介绍

- ZVM设计思路;
- ZVM架构设计;

- ZVM系统功能特性

- ZVM系统功能特性;
- ZVM系统新功能支持;
- 性能测试对比;

- 未来应用场景

嵌入式OS对富功能与硬实时的混合关键部署要求

双子星联动

富功能OS

- 智能座舱
- 数据可视化
- 场景可视化

实时控制OS

- 智能驾驶
- 底盘控制
- 动力控制

Linux宏内核

Linux: 功能强大
但实时性不足

实时微内核

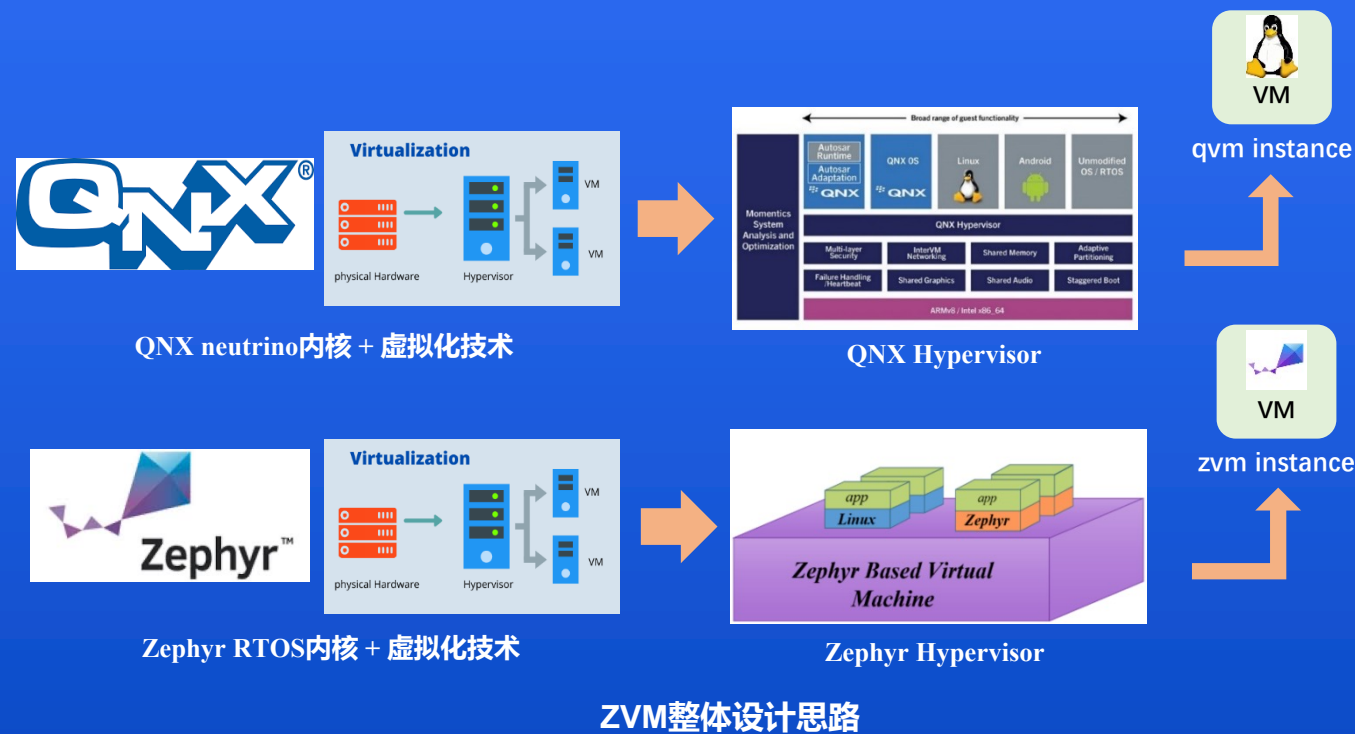
QNX: 以闭源的方式
来形成技术壁垒

满足关键场景对智能化、实时性及安全性的多重要求!

ZVM总体系统架构介绍

1 基于Zephyr RTOS的虚拟机设计思路

- Zephyr RTOS现有功能支持
- 核心虚拟化模块实现
- 可裁剪Zephyr 系统模块
- 运行虚拟机实例



目标：满足开源、实时，灵活，智能特性的嵌入式虚拟机管理平台

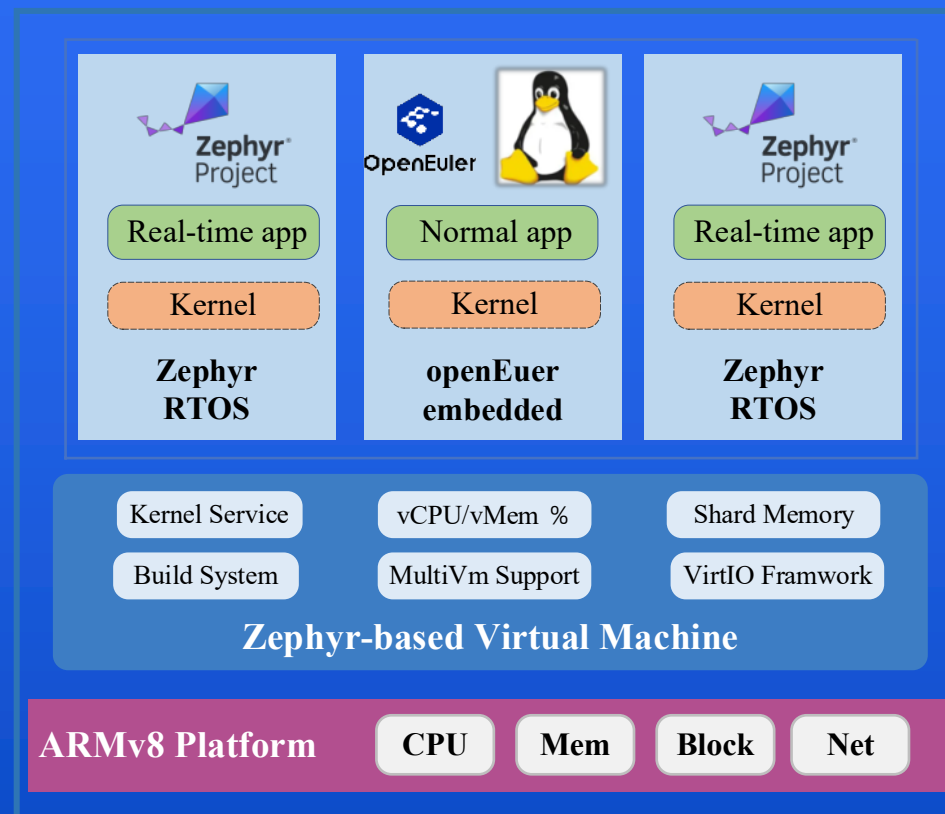
ZVM总体系统架构介绍

2

基于Zephyr RTOS的虚拟机架构设计

分层架构设计：包括硬件、hypervisor、guest OS到应用的多层级系统：

- 硬件层：ARMv8处理器平台，包括处理器、内存、中断控制器等核心设备的支持；
- 虚拟化层：核心虚拟化功能模块化设计，虚拟化拓展模块设计，复用Zephyr RTOS内核服务及设备模型；
- 内核层：运行Zephyr RTOS、Linux等ARMv8架构内核；
- 应用层：客户虚拟机中运行实时或非实时任务；



ZVM整体架构

ZVM系统功能特性

1

系统可配置性

- 支持配置ZVM的灵活配置，可在type-1到type-2型虚拟机的按需切换；

2

动态资源分配

- 支持运行时动态内存分配、启动时动态设备分配，自适应任务调度；

3

设备虚拟化

- 支持设备直通、完全虚拟化、半虚拟化virtIOv框架；

4

多类型Guest OS

- 支持社区Linux、openEuler Embedded、Zephyr RTOS等；

5

Guest OS间通信

- 支持基于共享内存的实时通信；

6

强实时性

- 复用Zephyr RTOS的强实时能力，支持抢占式、全局/分区调度算法；

7

端侧实时推理

- 支持Paddle Lite 端侧AI推理框架，支持ResNet50等多种AI模型；

8

ARMv8硬件架构

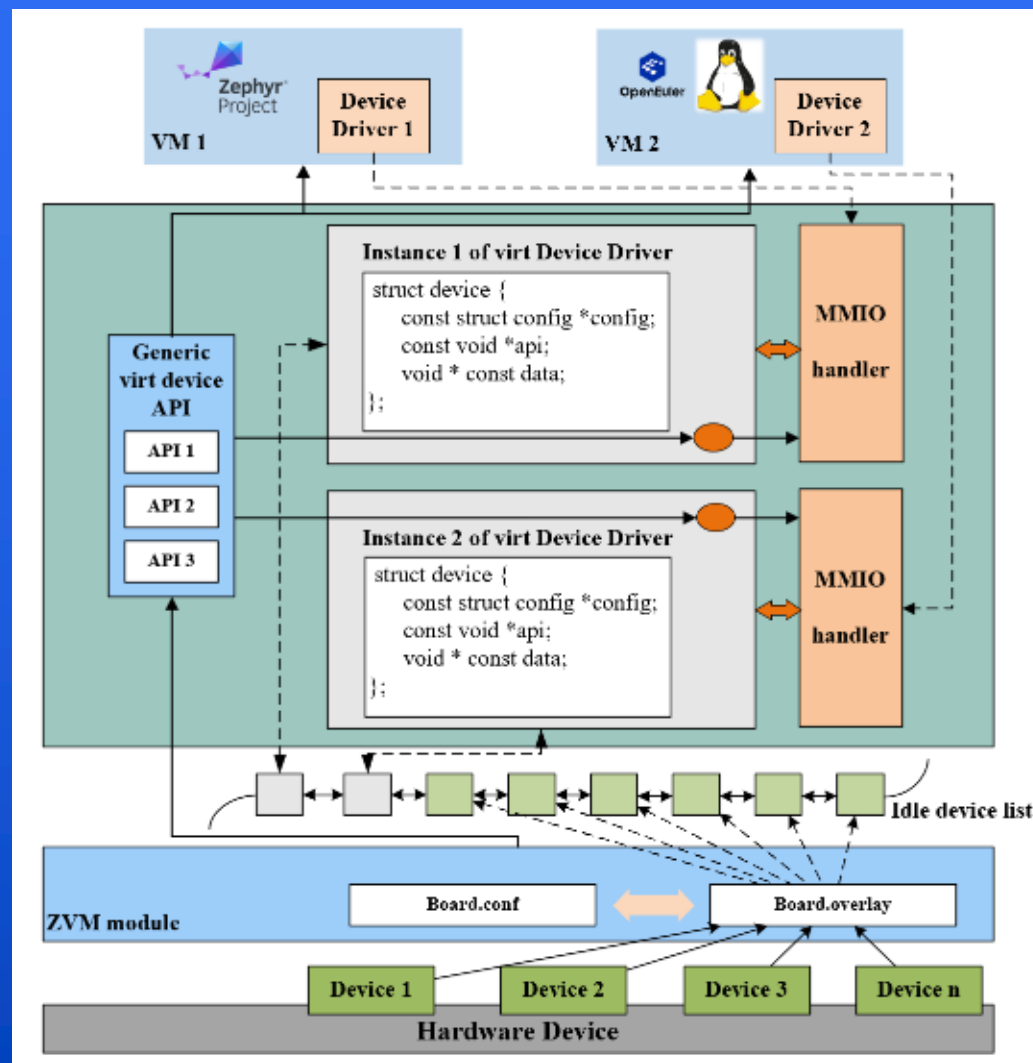
- 支持RK3568等国产嵌入式SoC板卡，利用VHE特性减少性能损耗；

ZVM系统功能新特性

1

ZVM系统设备模型

- 支持核心设备完全虚拟化;
- 支持半虚拟化方案;
- 支持ZVM虚拟机的直通设备模型;
 - 根据板卡overlay初始化板卡可用设备;
 - ZVM初始化idle device list;
 - VM初始化分配空闲设备;
 - 使用virt device driver实例虚拟化设备;
 - 建立虚拟机设备的MMIO映射;
 - 通过API -> handle函数处理设备中断;



ZVM虚拟机直通设备模型

ZVM系统功能新特性

2

设备半虚拟化框架virtIO

- 符合OASIS virtIO V1.1规范，支持virtio_mmio传输方式，实现了virtIO块设备的后端支持

```
h@h-B450-AORUS-ELITE: ~/hxy/zvm_workplace/zvm
[*****]
zvm_host:~#Disk reports 256 sectors
Disk reports sector size 512
virtio_blk read: start_sector = 0, num = 8
virtio_blk read: start_sector = 0, num = 8
virtio_blk read: start_sector = 32, num = 8
virtio_blk read: start_sector = 64, num = 8
virtio_blk read: start_sector = 128, num = 8
virtio_blk read: start_sector = 8, num = 8
virtio_blk read: start_sector = 24, num = 8
virtio_blk read: start_sector = 56, num = 8
virtio_blk read: start_sector = 120, num = 8
virtio_blk read: start_sector = 16, num = 8
virtio_blk read: start_sector = 0, num = 8
virtio_blk read: start_sector = 24, num = 8
virtio_blk read: start_sector = 0, num = 8
virtio_blk read: start_sector = 0, num = 8
virtio_blk write: start_sector = 128, num = 128
virtio_blk write: start_sector = 10, num = 8
virtio_blk read: start_sector = 8, num = 32
virtio_blk write: start_sector = 44, num = 2
virtio_blk read: start_sector = 40, num = 8
virtio_blk write: start_sector = 0, num = 48
virtio_blk write: start_sector = 0, num = 8
[ 11.980102] random: dd: uninitialized urandom read (512 bytes read)
[ 18.503227] ifconfig (618) used greatest stack depth: 12576 bytes left
starting pid 624, tty '/dev/ttyAMA0': '/sbin/getty 115200 ttyAMA0'

openEuler Embedded(openEuler Embedded Reference Distro) 23.09 qemu-aarch64 /dev/
ttyAMA0

Authorized uses only. All activity may be monitored and reported.

qemu-aarch64 login: root
You are required to change your password immediately (administrator enforced).
New password:
Retype new password:
[ 30.813528] random: login: uninitialized urandom read (16 bytes read)
qemu-aarch64 ~ # ls
qemu-aarch64 ~ # mkfs.ext2 /dev/vda
mkfs 1.47.0 (5-Feb-2023)
Creating filesystem with 128 1k blocks and 16 inodes
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

qemu-aarch64 ~ #
```

3

openEuler Embedded虚拟机支持

- 增添对openEuler Embedded虚拟机的支持，并支持启动个性化定制内核

```
h@h-B450-AORUS-ELITE: ~/hxy/zvm_workplace/zvm_vl
zvm_host:~#zvm new -t linux
Ready to create a new vm..Disk reports 256 sectors
Disk reports sector size 512
[ 7.493994] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 7.532641] cacheinfo: Unable to detect cache hierarchy for CPU 0
[ 7.669501] brd: module loaded
[ 7.715966] loop: module loaded
[ 7.730046] virtio_blk virtio0: [vda] 256 512-byte logical blocks (13
KiB)
[ 7.730655] vda: detected capacity change from 0 to 131072
[ 7.780914] 9pnet: Installing 9P2000 support
[ 7.789057] registered taskstats version 1
[ 7.804515] Warning: unable to open an initial console.
[ 7.916024] Freeing unused kernel memory: 79616K
[ 7.918364] Run /init as init process
[ 8.745509] rcS (312) used greatest stack depth: 13040 bytes left
[ 15.705479] random: dd: uninitialized urandom read (512 bytes read)
[ 24.618541] ifconfig (618) used greatest stack depth: 12576 bytes left
starting pid 624, tty '/dev/ttyAMA0': '/sbin/getty 115200 ttyAMA0'

***** Create vm successful! *****
***** VM INFO *****
***** VM-NAME: linux_os-2 *****
***** VM-ID: 2 *****
***** VCPU NUM: 1 *****
***** VMEM SIZE: 1024(M) *****

zvm_host:~#zvm run -n 2
***** Start vm successful! *****
***** VM INFO *****
***** VM-NAME: linux_os-2 *****
***** VM-ID: 2 *****
***** VCPU NUM: 1 *****

openEuler Embedded(openEuler Embedded Reference Distro) 23.09 qemu-aarch
ttyAMA0

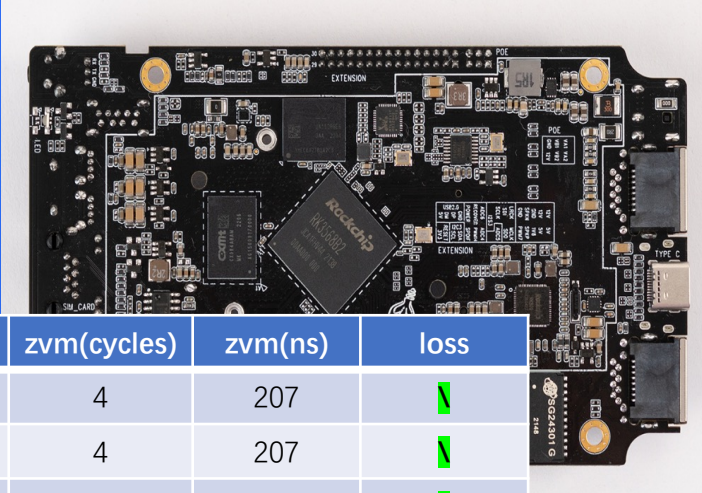
Authorized uses only. All activity may be monitored and reported.

zvm_host:~#virtio_blk read: start_sector = 0, num
qemu-aarch64 login:
```

7

ZVM性能测试与对比

处理器频率1.42Ghz, 计数器时钟频率24Mhz

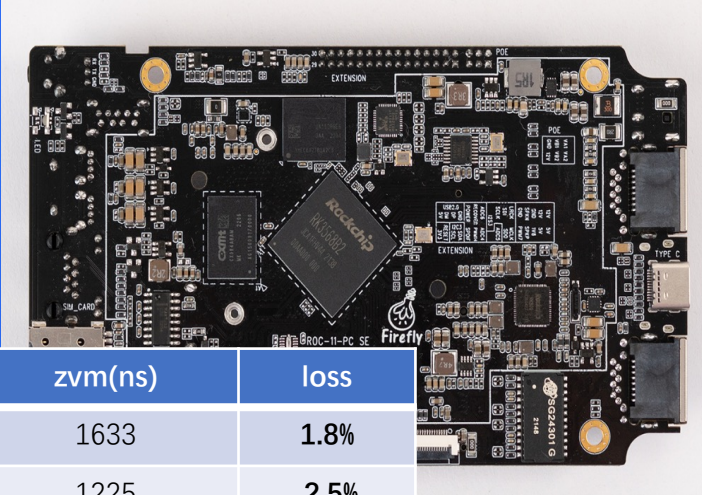


1 实时性对比, 基于Zephyr Latency_benchmark

items	board(cycles)	board(ns)	kvm(cycles)	kvm(ns)	loss	zvm(cycles)	zvm(ns)	loss
Preemptive threads ctx switch via k_yield (K -> K)	5	209	3	164	█	4	207	█
Cooperative threads ctx switch via k_yield (K -> K)	5	209	3	155	█	4	207	█
Switch from ISR back to interrupted thread	1	59	4	203	244.1%	1	59	█
Switch from ISR to another thread (kernel)	1	82	2	88	7.3%	1	82	█
Create kernel thread from kernel thread	33	1399	35	1459	4.3%	33	1396	█
Start kernel thread from kernel thread	6	268	7	321	19.8%	6	267	█
Suspend kernel thread from kernel thread	6	250	5	249	█	6	251	0.4%
Resume kernel thread from kernel thread	5	222	5	226	1.8%	5	221	█
Abort kernel thread from kernel thread	2	89	2	106	19.1%	2	88	█
Give a semaphore (no waiters) from kernel thread	0	3	0	2	█	0	3	█
Take a semaphore (no blocking) from kernel thread	429496	17895679	429496	17895672	█	429496	17895679	█
Take a semaphore (context switch K -> K)	5	241	5	227	█	5	244	1.2%
Give a semaphore (context switch K -> K)	7	296	5	237	█	7	296	█
Lock a mutex from kernel thread	429496	17895685	429496	17895673	█	429496	17895685	█
Unlock a mutex from kernel thread	429496	17895669	429496	17895660	█	429496	17895669	█
Average time for heap malloc	4	189	3	163	█	4	192	1.6%
Average time for heap free	3	163	3	148	█	3	157	█

ZVM性能测试与对比

处理器频率1.42Ghz, 计数器时钟频率24Mhz



2

系统性能，基于Zephyr sys_kernel benchmark

items	board(ns)	kvm(ns)	loss	zvm(ns)	loss
k_sem_init/ksem_take/ksem_give	1604	4112	156.4%	1633	1.8%
k_sem_init/ksem_take/ k_yield /ksem_give	1195	3937	229.5%	1225	2.5%
k_sem_init/ksem_take/ksem_give/ ksem_give /ksem_take	875	4316	393.3%	962	9.9%
k_lifo_init/k_lifo_get/k_lifo_put	1400	5920	322.9%	1370	█
k_lifo_init/k_lifo_get/ klifo_get /k_lifo_put/ k_yield	1283	5804	352.4%	1312	2.3%
k_lifo_init/k_lifo_get/k_lifo_put/ klifo_get /k_lifo_put	962	4608	379.0%	991	3.0%
k_fifo_init/k_fifo_get/k_lifo_put	1254	6008	379.1%	1283	2.3%
k_fifo_init/k_fifo_get/ k_fifo_get /k_lifo_put/ k_yield	1312	5116	289.9%	1283	█
k_fifo_init/k_fifo_get/k_fifo_put/ kfifo_get /k_fifo_put	1225	5891	380.9%	1225	█
k_stack_init/k_stack_pop/k_stack_push	1370	4533	230.9%	1370	█
k_stack_init/k_stack_pop/ k_stack_pop /k_stack_push/ k_yield	1195	5804	385.7%	1225	2.5%
k_stack_init/k_stack_pop/k_stack_push/ k_stack_pop /k_stack_push	933	4416	373.3%	933	█
k_mem_slab_alloc	145	300	106.9%	175	20.7%
k_mem_slab_free	116	262	125.9%	116	█

未来发展规划

1

ZVM未来发展规划（应用）

（1）增加对TSN的支持：

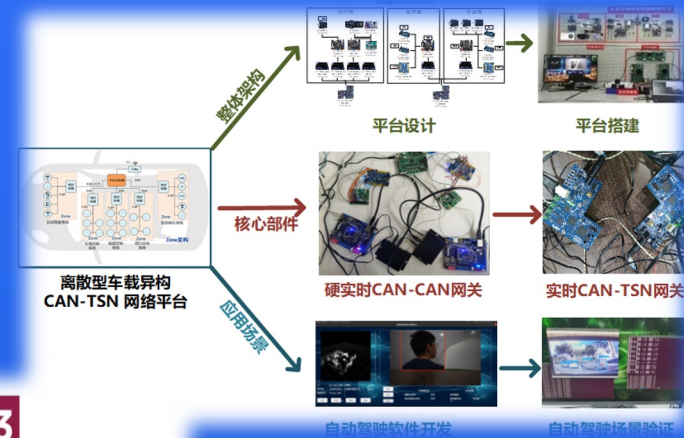
在虚拟机中**实现TSN通信协议**，保证车辆内多个域之间通信的**高可靠、低时延和确定性**，以满足严苛的实时要求和多个应用场景的需求。

（2）增加端侧AI推理的支持：

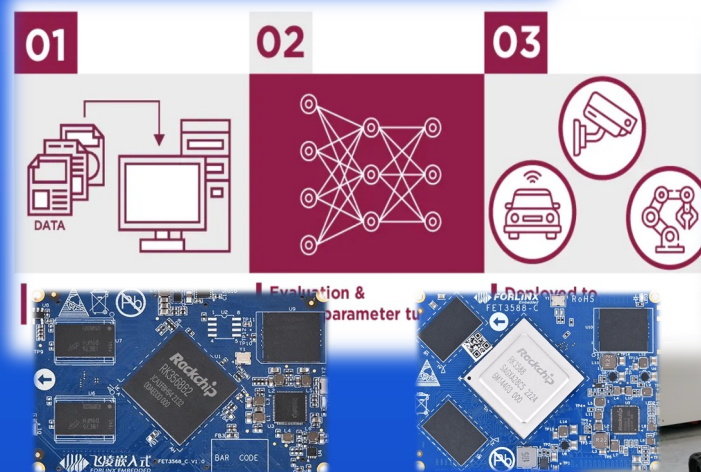
为ZVM适配更多国产高性能嵌入式板卡，为**AI模型的推理**提供充足的硬件资源。充分利用底层RTOS的实时特性，加速AI推理。

（3）增加对智能机器人控制系统的支持：

在智能机器人平台上搭建控制系统和人机交互系统，保证**实时控制**机器人的同时，实现较好的人机交互能力。



车载TSN网络



端侧AI推理支持



机器人控制系统

THANKS