

VxWorks 7 技术简介

VxWorks 7 是新一代的嵌入式实时操作系统，全新的架构设计，灵活化的组件配置，高可靠性与安全性，丰富的第三方软件包，极大的方便用户的开发和使用。本文对 VxWorks7 的功能做个简要的记录。

VxWorks 7 平台和配置

- VxWorks 7 内核平台提供 VxWorks 和 Workbench 组件
- 基于市场方向的平台配置

Wind River Workbench

Integrated Development Environment (IDE)

- Compilers
- Project facilities
- Static debugging and analysis
- Dynamic debugging and analysis
- Target simulation

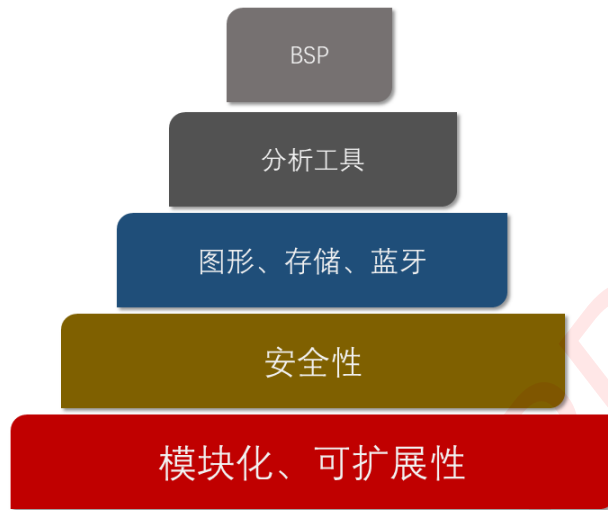
支持多样化的构建工具用于应用开发。

VxWorks 7 技术简介

VxWorks 7 技术简介

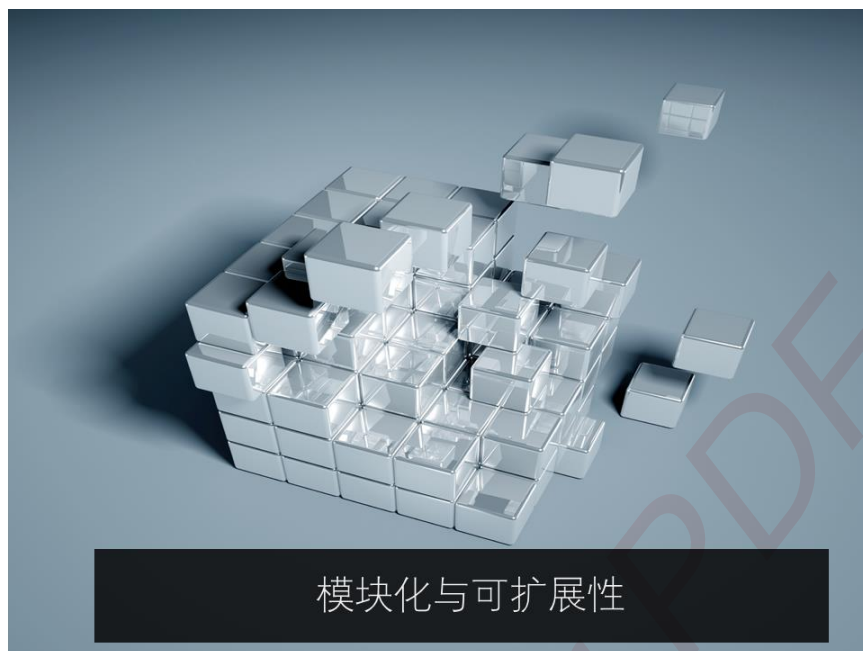
By 春秋

VxWorks 7 功能锥形图



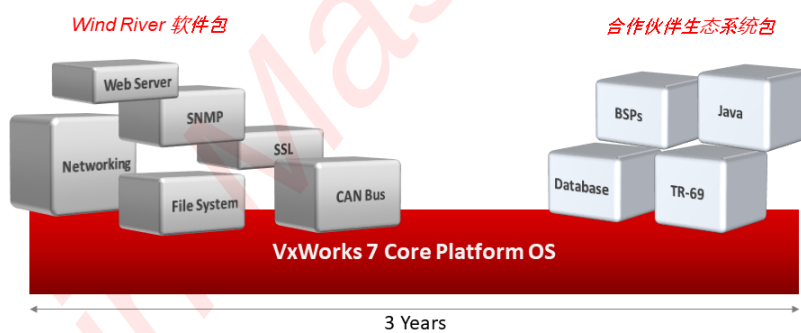
高质量的VxWorks 7 系统!





VxWorks 7 软件包范例

- 设计全新基础结构用于安装、分发和部署应用程序包
- 支持增量添加或删除组件



软件包生命周期独立于内核

- 能够根据需要随时修补或更新单个软件包

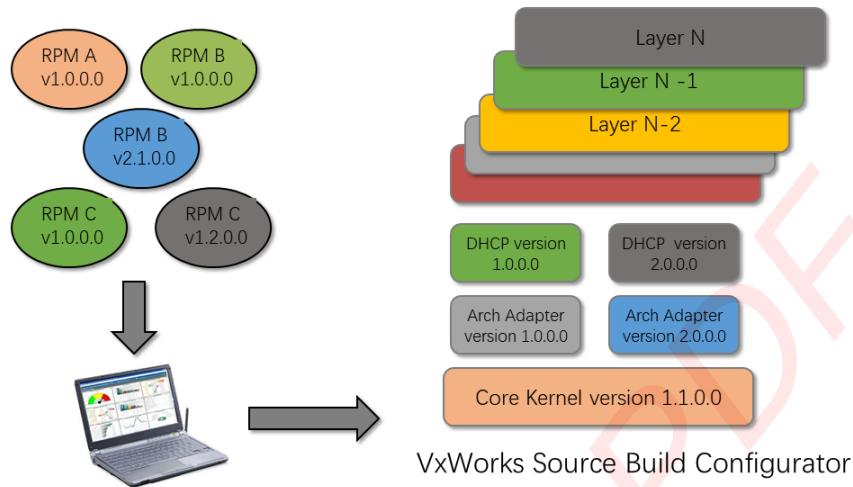


多个版本的软件包可以在开发树中共存

- 允许按照需求测试和回滚补丁或新版本包



RPM 和 Layer 编译系统



8

VxWorks Layers

- 任意功能模块支持多个版本同时存在
- 允许开发者选择功能模块的任一版本编译到 VxWorks 内核库
- 支持自包含(self-contained)、异步软件更新功能
 - 将所有 VxWorks 组件化
- 允许合作伙伴和最终用户增加软件代码的同时, 不需修改 Wind River 发行版的代码
- 仍旧保持 VxWorks 基础开发流程
 - Source Build → Kernel Image → DKM/RTP

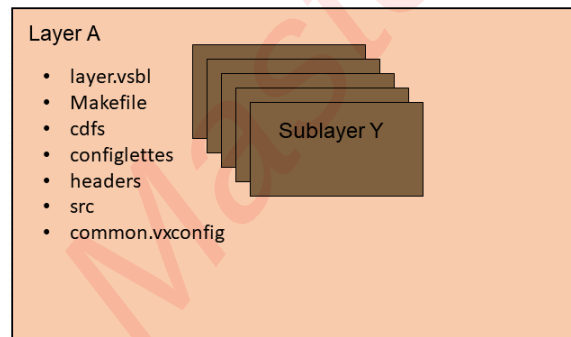
9

VxWorks Layers 特性

- Layers让开发者灵活地配VxWorks Source Builds
 - 增加 layer
 - 删除 layer
 - 编译单独 layer
 - 清除单独 layer
 - Layers 列表
- 提供全新的编译机制
 - Private, protected, and public header handling
 - RTPs 共享库独立于layer进行编译
 - 支持二进制layers (no source code)

10

Layer 模型 – Layer里面有什么?



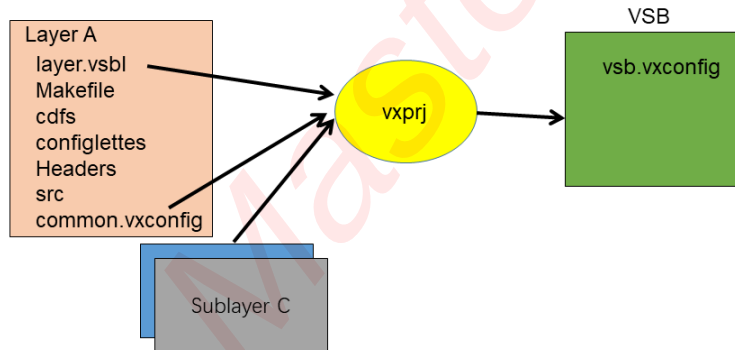
11

VSB Layers → layer.vsb 内容

```
Layer GPUDEV {  
  SYNOPSIS      GPU driver interface  
  VERSION       1.0.0.0  
  VSB_REQUIRES  SMP && CPU_ARMARCH7  
  DEPENDS  
  PARENT  
  OPTIONAL      YES  
  DEFAULT       YES  
}
```

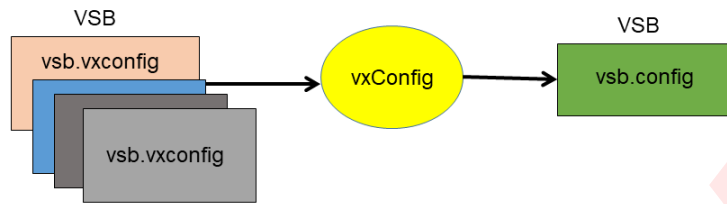
- VSB_REQUIRES – 设置过滤器用于控制layer的可见性
- DEPENDS – 设定当前layer的依赖layer列表

VSB Layers → vxprj → vsb.vxconfig



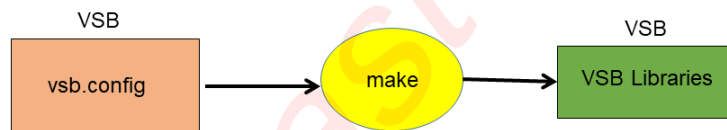
- VSB layer的元数据 (layer.vsb) 被vxprj转换成VSB的配置数据
- 此VSB配置数据连接到vsb.vxconfig文件，其中包含安装中找到的所有其他common.vxconfig数据

vsb.vxconfig->vxConfig->vsb.config



- vxConfig (kconfig) 使用 vsb.vxconfig 文件作为 VSB 配置的原始输入
- 通过与用户的交互 (Workbench 或命令行), vxConfig 生成输出文件 vsb.config

vsb.config-> make -> Libraries



- make 将源代码编译到 VSB 库, vsb.config 文件用于定义相关的内容和宏

VxWorks 软件包管理

- 用户可以使用Workbench GUI或命令行（CLI）的包管理工具创建RPM包
- 使用Workbench导出向导创建RPM包或RPM SPEC文件

16

并行编译

- 使用GNU并行编译特性
- 命令行下并行编译命令：`make -j <number of jobs>`
- 也可以使用Workbench使能并行编译并设置jobs数量
- 适用于VSB 和 VIP两种工程
- 在Workbench下，用户也可以使能DKM和RTP的并行编译

BSP	Windows不使用并行编译时间	Windows 并行编译时间	Linux不使用并行编译时间	Linux 并行编译时间
simpc	28 minutes	13 minutes	10 minutes	3 minutes

* 以上数据是在特定的硬件环境下测试结果

17

多编译器支持

- 支持多种/多版本的编译器，diab/gnu/icc
- 支持用户在某个工程中选择旧的编译器，另外的工程选择新版本的编译器
- 不需要强制升级

支持的编译器列表：

BSP	GNU GCC	Diab	ICC
VxWorks 6.9 Compatible	4.3.3.1	5.9.1.0	12.0.0.0
VxWorks 7	4.8.1.0	5.9.4.0	14.0.0.017

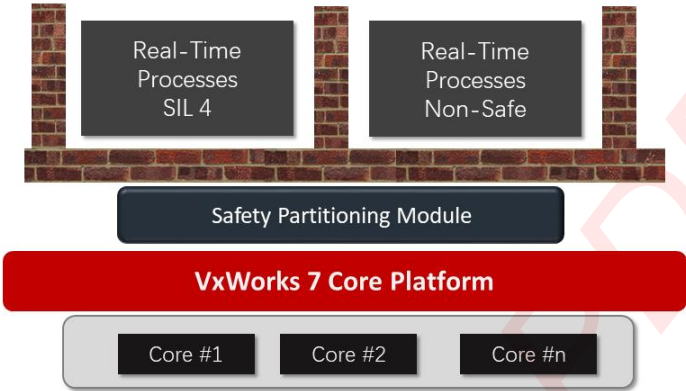
18



安全性

VxWorks 7 – 支持IEC 61508认证

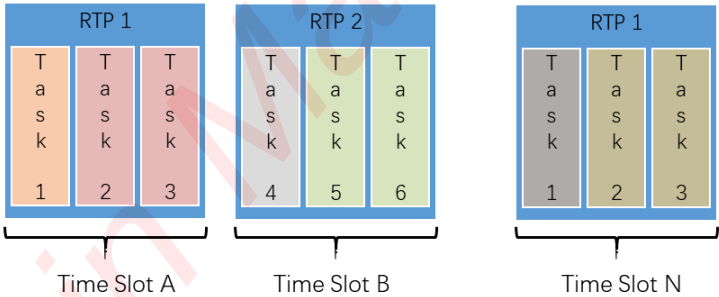
新版本的分区组件提供强健的时间、空间和资源的隔离



20

RTP 时间分区

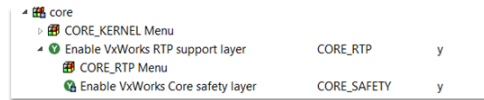
- 可选配的
- 仅支持单核或SMP模式
- 每个RTP进程分配固定的时间片



21

RTP 时间分区

- VSB必须包含**CORE_SAFETY**



- VIP必须包含**INCLUDE_TIME_PART_SCHEDULER**
- 分区在**usrTimePartScheduler.c**文件的全局结构体中被定义
- 允许给内核任务分配一个时间槽(slot)，或者允许内核从当前运行的RTP“偷取”运行时间

22

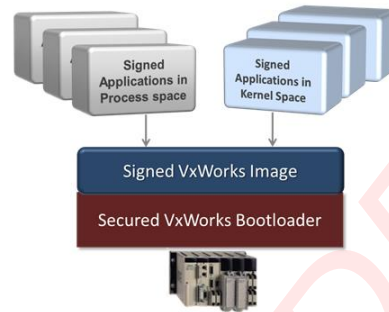
VxWorks提供的四大安全支柱



23

示例：高可靠性

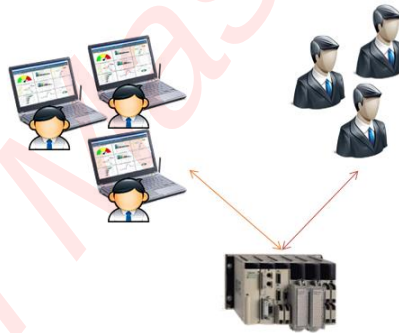
- 未经授权的映像不被加载
- 仅运行可信的模块
- 没有恶意代码注入
- 解密
- 数字签名认证(X.509)
- 支持
 - DKM
 - RTPs
 - Shared Libraries
- Workbench 插件被集成到编译系统
- 安全启动器
 - UEFI 检查VxWorks映像数字签名(Intel only)



24

示例：用户管理

- 经过授权的用户具有访问权限
- 定义和加强基于用户的策略
- 可定制化的检查和日志管理



25

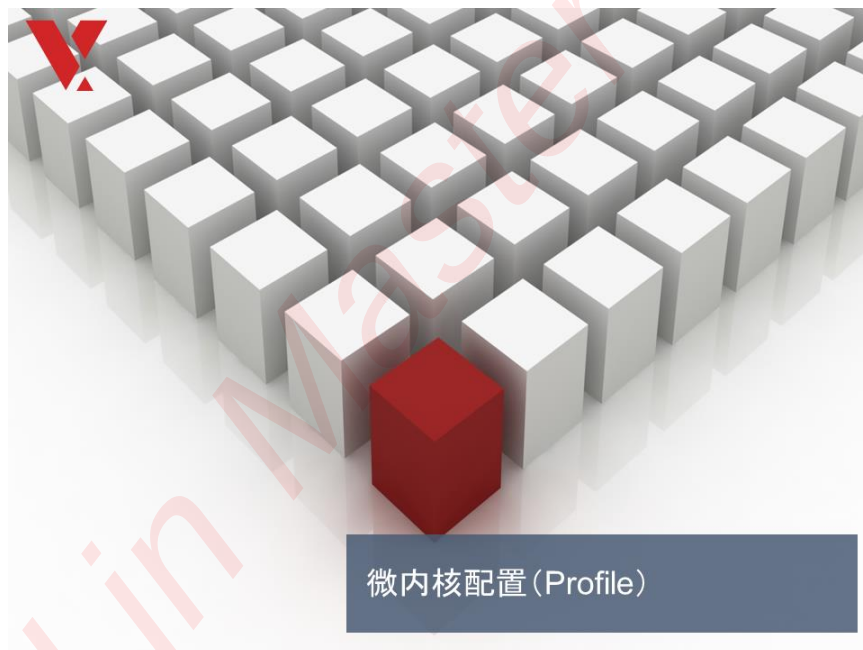
全新的用户数据库

- 必须在VSB中使能

4 security		
✓ Enable Hash algorithms	HASH	y
✓ Enable VxWorks login features	LOGIN	y
✓ Enable openssl library	OPENSSL	y
✓ Enable VxWorks user management features	USER_MANAGEMENT	y

- 在VIP工程中：
 - `INCLUDE_SECURITY`
 - `INCLUDE_USER_IDENTIFICATION`
 - `INCLUDE_USER_DATABASE`
- 没有默认用户
- 没有静态定义的用户
- 强制使用用户登录

26



什么是VxWorks微内核?

- 超级微型内核
 - 简约化
 - 有效性 (大约 25 μ s 启动时间)
 - 小型化 (单个任务最小占用空间(footprint) = 2,280 字节)
- 基于2002年发布的DSP RTOS 技术
- 主要特性
 - 硬实时性
 - 多线程
 - 支持多核
 - 电源管理



28

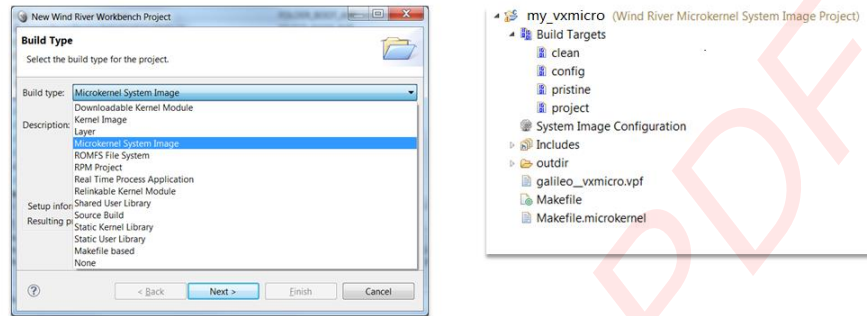
名副其实的实时操作系统(RTOS)

- 支持多线程
 - 基于优先级的可抢占机制
 - Run to completion
 - Round-robin / Time-slicing
 - 合作式多任务结构(轻量级线程执行)
- 没有动态内存分配 – 所有事物都是预先定义的
- 支持全部中断服务ISR
- 通信和同步
 - 信号量 (Binary, Counting and Mutex)
 - 消息队列, 通道(channel, 类似于管道), LIFOs 和 FIFOs

29

创建微内核系统映像

- 使用创建工程向导
 - Name
 - BSP
 - Options

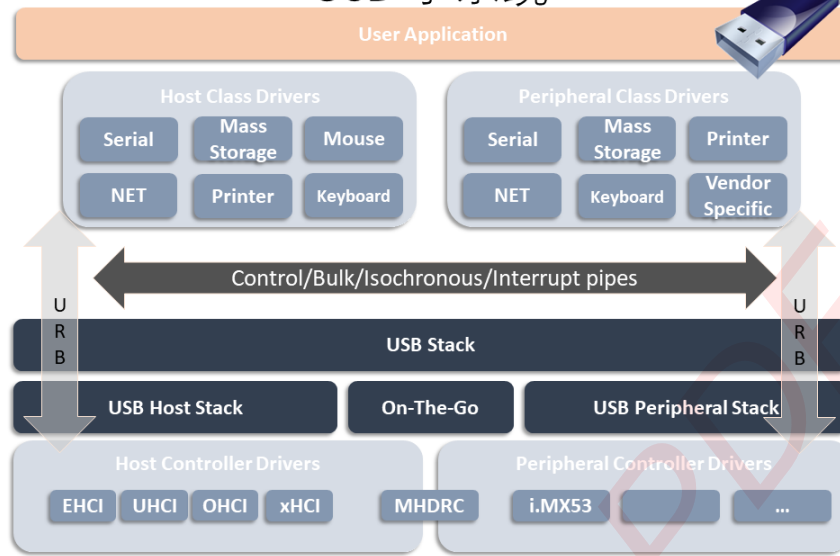


30



连通性，图形化，存储，中间件和应用

USB 子系统



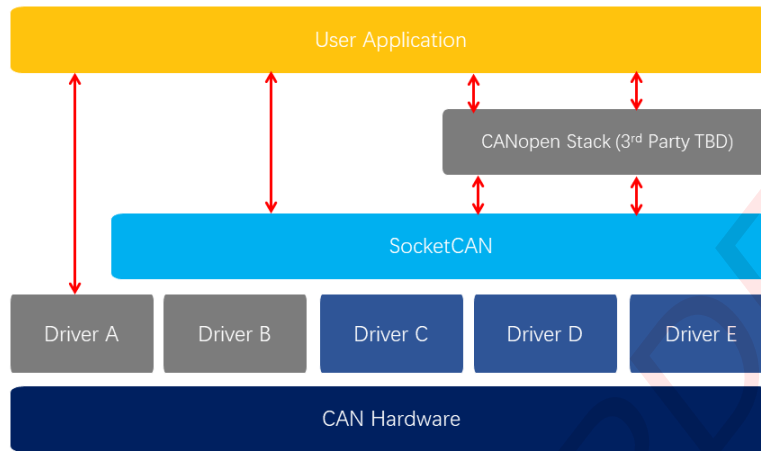
32

Workbench VIP中USB配置

USB Common	FOLDER_USB_COMMON
USB Common Stack	INCLUDE_USB
USB Common Stack Init	INCLUDE_USB_INIT
USB Debug	FOLDER_USB_DEBUG
USB Show	FOLDER_USB_SHOW
usbShow	INCLUDE_USB_SHOW
USB Track	FOLDER_USB_TRACK
USB Host	FOLDER_USB_HOST
USB Host Controller Init	FOLDER_USB_HOST_INIT
USB Host Devices	FOLDER_USB_HOST_DEVICES
USB GEN2 Class drivers	FOLDER_USB_GEN2_DEVICES
USB GEN2 Device Init	FOLDER_USB_GEN2_DEVICES_INIT
USB GEN2 Keyboard Initialization	FOLDER_USB_GEN2_KEYBOARD_INIT
USB GEN2 Serial Device Initialization	FOLDER_USB_GEN2_SERIAL_INIT
USB GEN2 END Device Init	INCLUDE_USB_GEN2_END_INIT
USB GEN2 Helper Init	INCLUDE_USB_GEN2_HELPER
USB GEN2 Mass Storage Init	INCLUDE_USB_GEN2_STORAGE_INIT
USB GEN2 Mouse Init	INCLUDE_USB_GEN2_MOUSE_INIT
USB GEN2 Printer Init	INCLUDE_USB_GEN2_PRINTER_INIT
USB GEN2 END Controllers	FOLDER_USB_GEN2_END_CONTROLLERS
USB GEN2 Serial Adapters Driver	FOLDER_USB_GEN2_SER_ADAPTERS
FTDI 232 Serial Adapter Driver	INCLUDE_USB_GEN2_SER_FTDI232
Prolific 2303 Serial Adapter Driver	INCLUDE_USB_GEN2_SER_PL2303
WRS USB CDC Serial Adapter Driver	INCLUDE_USB_GEN2_SER_WRS_CDC
USB GEN2 Mass Storage	INCLUDE_USB_GEN2_STORAGE
USB GEN2 Printer	INCLUDE_USB_GEN2_PRINTER
USB Host Class Driver Init	INCLUDE_USB_HOST_CLASS_INIT
EHCI	INCLUDE_EHCI
USBD Module	INCLUDE_USBD
USB PHY Drivers	FOLDER_USB_PHY_DRV
FDT Freescale Internal on-chip USB PHY Contr	INCLUDE_USB_PHY_FSL_ONCHIP
FDT USB Viewport ULPI PHY Controller driver	INCLUDE_USB_PHY_ULPI_VIEWPORT

33

支持SocketCAN的CAN解决方案

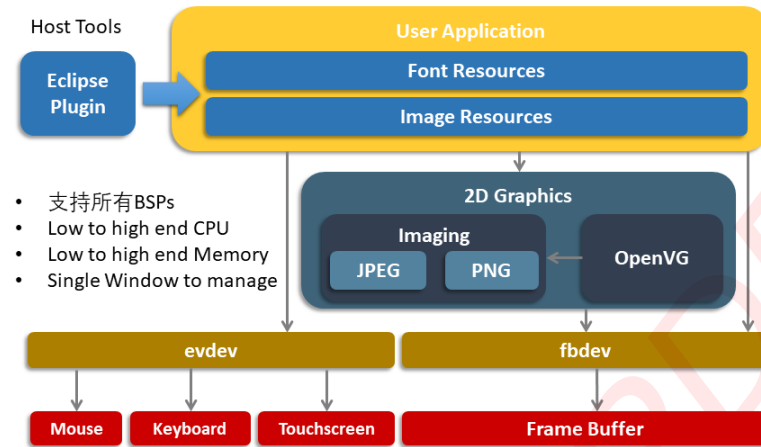


34

CAN组件

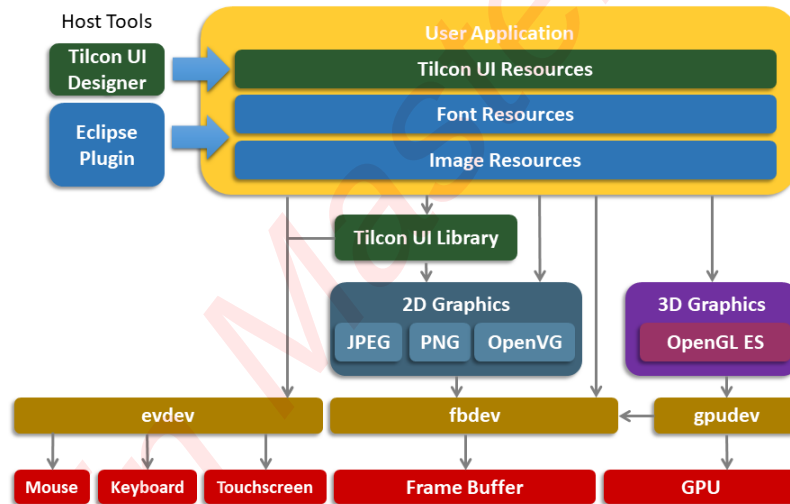
- SocketCAN stack replaces Proprietary CAN stack
- CAN 总线控制器支持列表
 - AM335x
 - Xilinx Zynq
 - i.MX6
 - SJA1000
 - TI Keystone II
 - Altera Cyclone V
 - Intel Tunnel Creek
 - Freescale QorIQ

图形化



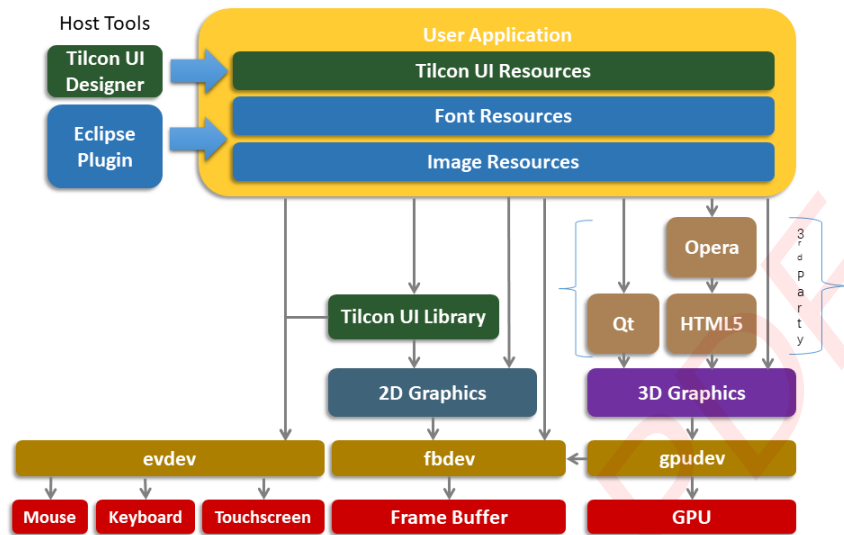
36

多媒体UI解决方案



37

高端的第三方UI解决方案

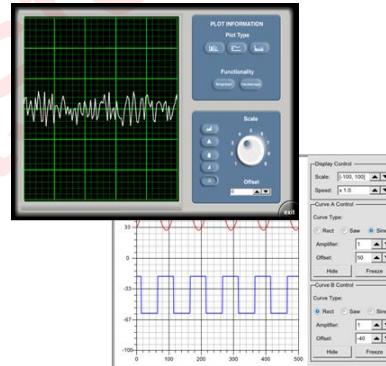


38

© 2013 Wind River. All Rights Reserved.

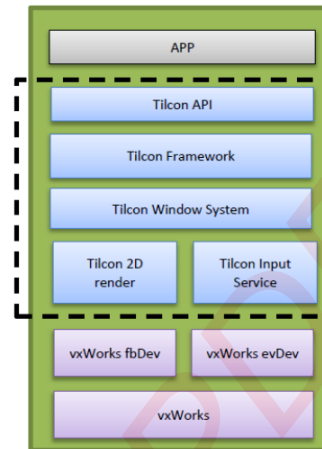
Tilcon 新特性

- Tilcon 新版本 (based on Tilcon 5.9) 7.0.0.0
- Wind River 轻量级部件开发工具
- 支持所有的基础原型设计
 - 支持Windows 和 Linux 开发环境
- 兼容多种硬件平台



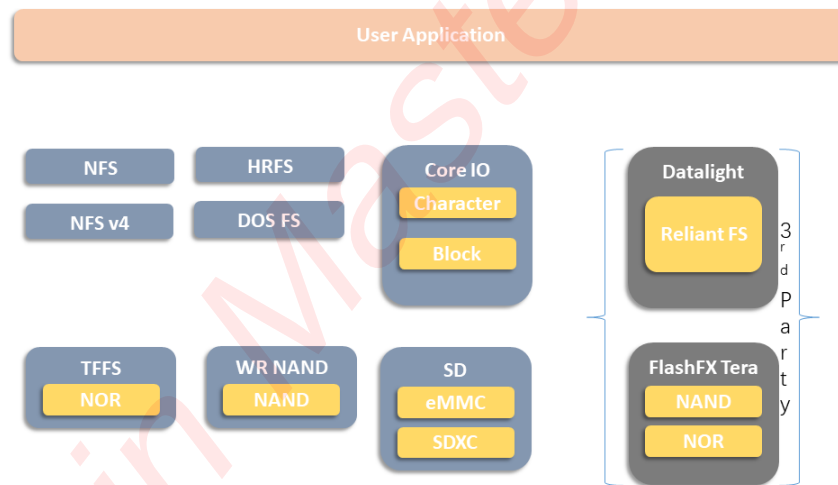
Tilcon 增强功能

- 采用C/S架构取代原有的中间件库
- 新一代的部件分发机制
 - 性能具有显著提高
- 集成小窗口系统
 - 最低要求: fbdev & evdev 驱动
- C++ 接口
 - 允许部件优化

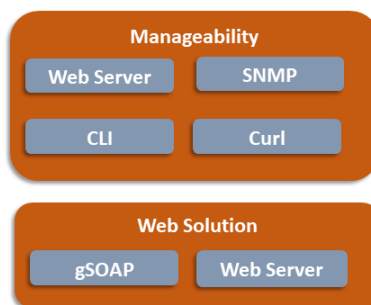


<https://www.vxworks.club> (VxWorks 俱乐部)

存储和文件系统



设备网络管理



Layers and RPMs实现组件化

42



BSP 模型：新一代简单化、有效化设计模型

VxWorks 6 BSP Model

