What is VxWorks RTOS?

- 32/64 bits on Arm/Intel/MIPS/PowerPC
- Proprietary real-time OS, POSIX PSE52
- Kernel/user space separation, user space optional
- C/C++11/14, possible to develop kernel C++ modules and user apps
- Safety certifiable: DO-178, ISO 26262, IEC 61508
- Toolchain LLVM 8, Dinkumware C/C++ libs
- Proprietary build system
- Kernel shell
- Eclipse-based IDE, Windows/Linux hosts
Industry Examples

- COMAU
- KUKA
- SIEMENS
- MITSUBISHI ELECTRIC
- ABB
- NASA
- YASKAWA
- BOSCH
- Schneider Electric
- GE
Embedded Development Landscape

- Heterogeneous HW: MCU, CPU, GPU, VPU, TCU, FPGA, SOC
- Embedded, Edge, Fog, Cloud
- Sensors: Camera, Lidar, Radar, IMU, Ultrasonic
- Real-time: hard, soft, best effort
- C/C++ programming language
- Not enough SW engineers with embedded development skills
- Many people can’t program (embedded)
Embedded Development Landscape

- Heterogeneous HW: MCU, CPU, GPU, VPU, TCU, FPGA, SOC
- Embedded, Edge, Fog, Cloud
- Sensors: Camera, Lidar, Radar, IMU, Ultrasonic
- Real-time: hard, soft, best effort
- C/C++ programming language
- Not enough SW engineers with embedded development skills
- Many people can't do not want to program (embedded)
- Stop scaring people with real-time, safety and security
Documented ROS Robots

https://robots.ros.org/

Amount of robots grows rapidly, but where are the new embedded developers? People don’t want to program bit and bytes anymore
ROS2 Architecture

C/C++ in an embedded API.
A Robotics domain specific API is needed instead.

* Intra-Process Comms and Type Masquerading could be implemented in the client library, but may not currently exist.
Technical Debt grows

An example of the technical debt grow in the kernel community

ROS2 Technical Debt

Reducing Technical Debt

- Extend testing and resolve bugs in the current code base
- Waitset inconsistency
- Multi-threading problems with components
- Fix flaky tests.
- Ability to run (all) unit tests with tools e.g. valgrind
- API review
- Synchronize / reconcile design docs with the implementation.
- Pre-release retrospective review (APIs, docs, etc.)
- Address / classify pending tickets
- Address TODOs in code / docs

It is probably the same for ROS2
Run-time optimization (performance, footprint, RAM, I/O etc.) is very difficult to handle

http://www.brendangregg.com/ebpf.html
Usability Problem

- Little kids playing with a smartphone, they don’t know how it works
- Increasing amount of robots versus decreasing amount of embedded software engineers
- Productivity crisis in the embedded software development?
- Look at ROS2 APIs only C++, C and Python – typical embedded
- TSN protocols
- Machine learning networks
- Security
- Safety
- Little kids playing with a robot?
ROS2 Developer Journey
Desktop

App 1  App N  App 1  App N

2 2

ubuntu®

Intel CPU  Intel CPU
ROS2 Developer Journey
embedded Linux development

A journey from Ubuntu desktop to the embedded Linux e.g. Yocto is the rocky one:
- Cross compilation
- Complicated Build system
- etc
ROS2 Developer Journey
embedded real-time

App 1   App N
2

Intel CPU

App 1   App N
2

ARM/Intel multi-core CPU

Real-Time App 1
Real-Time App N
2

RTOS

SoC
ROS2 Developer Journey
embedded real-time, safe and secure

App 1  App N

2

Intel CPU

App 1  App N

2

ARM/Intel multi-core CPU

Real-Time App 1  Real-Time App N

2

RTOS

SoC

Critical App 1  Critical App N

2

Safe and secure OS

Safe HW
ROS2 Developer Journey
where is the magic button to optimize a run-time behaviour?

Hypervisor

Intel CPU
ARM/Intel multi-core CPU
SoC
Safe HW

Safe and secure OS

Real-Time App 1
Real-Time App N

App 1
App N

Critical App 1
Critical App N

RTOS

RTOS

2

2

2

Yocto Project

Ubuntu®

ARM/Intel multi-core CPU
ROS2 Working Groups

In my view a usability WG is missing which will make ROS2 more user friendly
For the first time Wind River provide a downloadable SDK for the non-commercial usage
ROS2 is built on top of the VxWorks SDK

Developers can deploy and run VxWorks on ARM and Intel
ROS2 Dashing Release VxWorks Port

- Complete ROS 2 Dashing release has been ported to VxWorks
- Build using colcon, the same look and feel as a native ROS 2 build (command line)
- OpenCV integration
- Python (ported, not tested)
- Only graphical packages (like RViz) are not ported and stay on Ubuntu

ROS 2 Apps
- ROS 2 VxWorks SDK

ROS 2 dependencies: ASIO, tinyxml2, OpenCV
- Python 3.8
- POSIX
- Cmake / autotools build primitives
- LLVM C++11/C++14
- VxWorks SR620
- Intel 64-bit / Arm / QEMU

**VXWORKS7-ROS2-BUILD (A Helper Repo)**

- [https://github.com/Wind-River/vxworks7-ros2-build](https://github.com/Wind-River/vxworks7-ros2-build)
- Based on the VxWorks SDK
- Download the SDK
- Setup the development environment and do: make
- ROS2 middleware and dependencies will be built
- Board support:
  - RPI3, UP2 and QEMU
  - RPI4 and others to come
- Docker build:
  - VxWorks SDK
VXWORKS7-LAYER-FOR-ROS2 (VxWorks patches)

- [https://github.com/Wind-River/vxworks7-layer-for-ros2](https://github.com/Wind-River/vxworks7-layer-for-ros2)
- ROS 2 dependencies patches:
  - ASIO, tinyxml2
- ROS 2 patches:
  - fastcdr, fastrtps, rcl, rclutils, etc.
ROS2 Build Under VxWorks

- From the command line (ROS 2 native build)
  - colcon build --symlink-install --cmake-force-configure --cmake-args -DBUILD_TESTING=OFF

- The same look and feel as a ROS 2 native build
  - source $SDK_PATH/toolkit/wind_sdk_env.linux
TURTLEBOT3 Support will come soon
Wind River contribution

- VxWorks, WR Linux Yocto, StarlingX, Hypervisor, TSN, Zypher
- VxWorks SDK published on WR Labs
- ROS2 SDK on VxWorks and WRLinux
  - ARM (RPI4), Intel, QEMU
  - DevOps (containerized)
- VxWorks SDK preintegrated with Eclipse IDE and VSCode
- Participation in Real-time WG, Safety WG and Embedded WG

- Find us on github - https://github.com/Wind-River/vxworks7-ros2-build
Building from source 2019

We support building ROS 2 from source on the following platforms:

- Linux
- OS X
- Windows
Building from source

2020

We support building ROS 2 from source on the following platforms:

- Linux
- OS X
- Windows

VxWorks

We plan to add both to the list of officially supported OSes

WRLinux
Innovate with us on labs.windriver.com