ROS 2 ON VXWORKS
CHALLENGES IN PORTING A MODERN SOFTWARE FRAMEWORK TO AN RTOS

ANDREI KHOLODNYI, PRINCIPAL TECHNOLOGIST, TECHNOLOGY OFFICE
TOPICS OF MY TALK TODAY

- What Is VxWorks?
- Why Is ROS 2 on VxWorks?
- ROS 2 at Wind River (Why We Are Doing It)
- Technical Challenges
- Non-technical Challenges
- Conclusions and Next Steps

Andrei Kholodnyi
Principal Technologist
CTO Office

FOCUS

> ADAS/HAD, Connected Vehicle, IVI, OTA, Automotive Security
> ROS 2 Mobile Robotics
> Products, Solutions; Partnerships & University Programs

RELEVANT PROJECTS (selection)

> First GENIVI Linux based IVI project development for BMW
> AdvancedTCA (Networking) for Alcatel and Lucent
> Leadership and delivery of IVI, telematics, and connectivity projects
> Thematic pathfinding for the new technologies (autonomous robotics, dependability)
> Continuous innovation
> Improvement for existing and creation of new company products
> Participation in alliances (GENIVI, FASTR, AUTOSAR Adaptive, PICMG)
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
WHY ROS 2 AT WIND RIVER

- Show VxWorks running robotics platform based on ROS 2
- Engage with R&D customers already interested in leveraging ROS 2
- Upstream ROS 2 changes back to the community

- Identify challenges in porting a modern software framework (with Python, Boost, cmake, etc.) to VxWorks and address those challenges
- Provide some hints (common problem-solving patterns) for how to build Linux applications under VxWorks
- Identify gaps in a development workflow (VxWorks versus Linux) and address those gaps
ROS 2 IN KEYWORDS (2017)

VxWorks

ROS 2
ROS 2 IN KEYWORDS (2017)

VxWorks

::: ROS 2

!!!Pain!!!

alone
ROS 2 IN KEYWORDS (2019)

- Docker
- DDS
- eProsima
- ROS-Industrial
- ROS community
- Titanium
- VxWorks
- microROS
- turtlebot3
- Wind River Labs
- ROS 2 Lab
- USB2Serial
- Dashing
- Zephyr
- Helix Virtualization Platform
- TSN
- CES2020
- Developer Journey
- ROS2019
- LLVM
- IP Review
- ARM
- SDK
- Intel
- UML
- LG
- UP2
- Wind River Labs
- RPI4
- meta-ros
- Autoware
- Wind River Linux
- Simulation
- Apollo
- Digital Twin
- Webinar
- Training
- Blog
- Apollo
- Digital Twin
- Webinar
- Training
- Blog
THE ‘RIPPLE’ EFFECT OF ROS 2
ROS 2 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


based on the ROS 2 dashing release
approx. 200 ROS 2 packages

OSS_BUILD layer
UNIX_EXTRA layer
https://labs.windriver.com
(ROS 2 ON VXWORKS, WIND RIVER LINUX)
VXWORKS7-ROS2-BUILD (A HELPER REPO)

- [Link](https://github.com/Wind-River/vxworks7-ros2-build)
- Makefile build: `BUILD_TYPE=Debug BOARD=up2 make`
- Set of scripts to build:
  - bootloader, kernel, userspace, ROS2, and the rootfs (boot from the USB stick)
- Board support:
  - UP, UP2, RPI3/RPI4, VxWorks Simulator, QEMU, and others
- Docker build:
  - VxWorks product install
VXWORKS7-LAYER-FOR-ROS2

- [https://github.com/Wind-River/vxworks7-layer-for-ros2](https://github.com/Wind-River/vxworks7-layer-for-ros2)
- VxWorks layers (build infrastructure):
  - OSS_BUILD, UNIX_EXTRA
- VxWorks layers (ROS 2 dependencies):
  - ASIO, tinyxml2
- ROS 2 patches:
  - fastcdr, fastrtps, rcl, rclutils, etc.
ROS 2 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
  - `./wrenv.linux -p vxworks-7 && ./vxworks_env.sh`
DUMMY ROBOT
CHALLENGES IN PORTING A MODERN SOFTWARE FRAMEWORK (BASED ON THE LATEST VXWORKS RELEASE)

- IP compliance !!!
- Missing libraries (ROS 2 dependencies need to be ported)
- Missing UNIX functions (e.g., fnmatch, memccpy, some others)
- cmake support in VxWorks is good, but not good enough
- Missing autotools support under Windows
- Dinkum C++ library is not equal to stdlibc++
- Python support is missing in VxWorks
IP COMPLIANCE

- ROS 2 license is fine (BSD 3-Clause), but …
- There are many dependencies that have different licenses
  - ASIO (Boost Software License)
  - EIGEN (Mozilla Public License 2.0, GPL and other licenses)
  - PCL (BSD license)
  - POCO (Boost Software License)
  - TINYXML (zlib License)
  - TINYXML2 (zlib License)
- This is a real problem for customers who want to run their software on top of it
- Would be good to review a complete licenses list
ROS 2 DEPENDENCIES

- Some of them are not really necessary (e.g., difficult to certify, not needed for non-Windows systems)
  - POCO
  - ASIO (from FastRTPS)
  - tinyxml vs tinyxml2

- ROS 2 dependencies need to be ported
  - Not a straight (colcon) way to port them in compare to ROS 2 packages

- Missing UNIX/Linux functions
  - fnmatch, memccpy, some others

- Probably we need a version without some dependencies
  - POCO is a good example (used to run ROS 2 under Windows)
ROS 2 EMBEDDED VS. ROS 2 FULL

Embedded
- No HMI (meant graphics)
- No visualization tools (Qt, Rviz…)
- No simulation tools (Gazebo)
- No Python
- Remove some dependencies

To make it possible building a ROS2 embedded version
CONCLUSIONS AND NEXT STEPS

- ROS 2 runs on VxWorks (great experience)
- Current results are published on https://labs.windriver.com
- Docker-based host environment that can be used for reproducible cross-platform builds

- Make VxWorks officially supported by ROS 2 Dashing
- Provide non-commercial VxWorks SDK for RPI4
- Real-time performance tests: VxWorks, real-time ROS2 working group
- Python: VxWorks integration and test
- Windows development host support: test VxWorks build