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)

AEROSPACE AND DEFENSE SECTOR

CHEMICAL SECTOR

COMMERCIAL BUILDING SECTOR

COMMUNICATIONS SECTOR

CRITICAL MANUFACTURING SECTOR

DAMS SECTOR

EMERGENCY SERVICES SECTOR

ENERGY SECTOR

FINANCIAL SERVICES SECTOR

FOOD AND AGRICULTURE SECTOR

GOVERNMENT BUILDING SECTOR

IT SECTOR

MEDICAL SECTOR

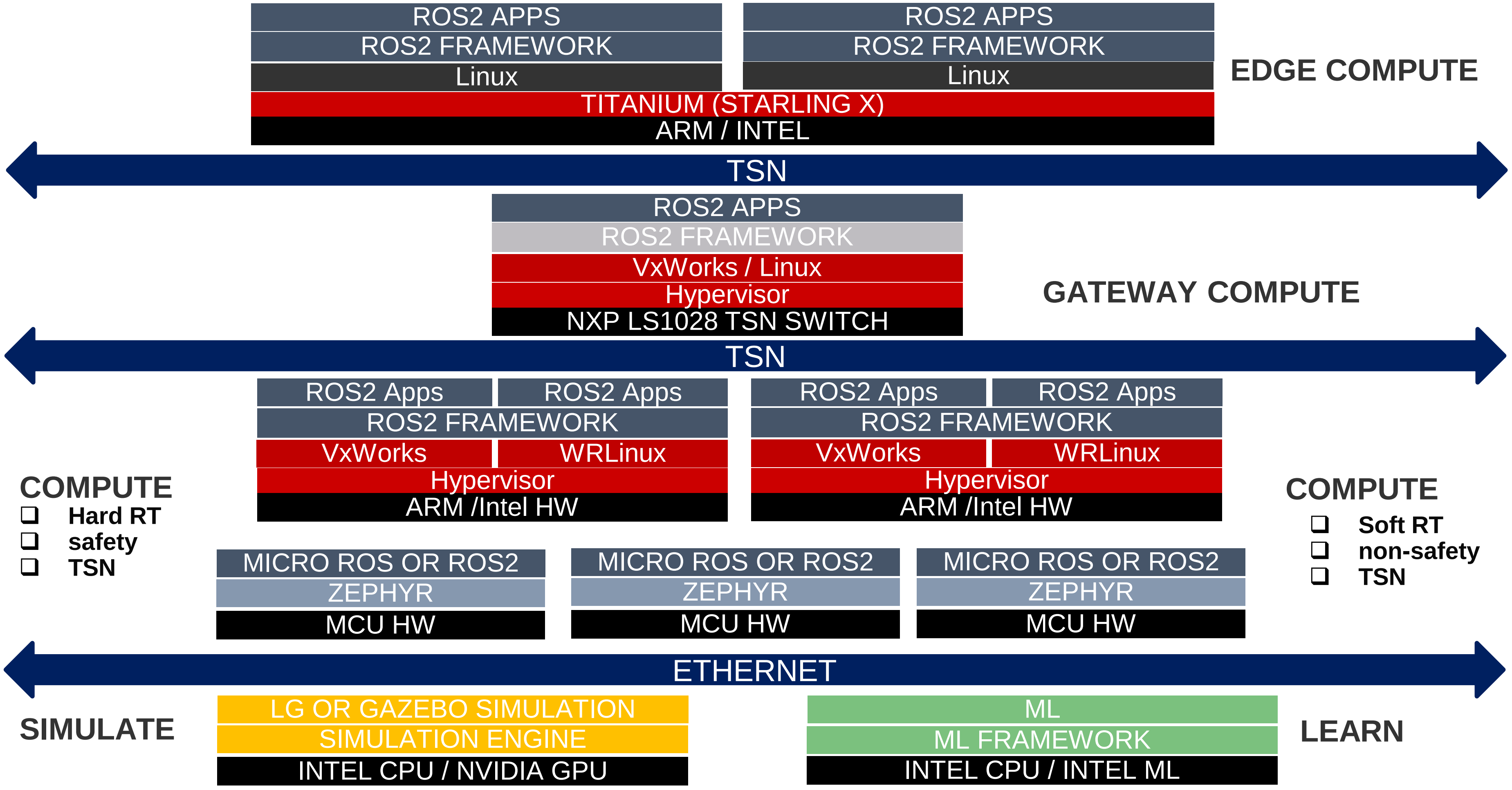
NUCLEAR SECTOR

TRANSPORTATION/AUTO SECTOR

WATER AND WASTEWATER SECTOR

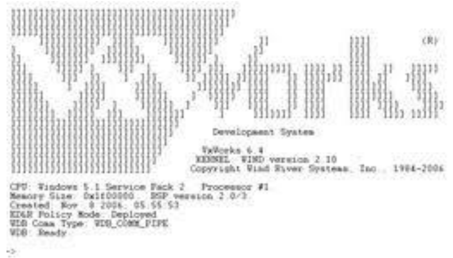
WIND

INDUSTRIAL PROFILE



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



```
                                   Development System

                                 VxWorks 6.4
                               KERNEL: WIND version 2.10
                              Copyright Wind River Systems, Inc., 1984-2006

CPU: Windows 5.1 Service Pack 2   Processor #1
Memory Size: 0x1f00000   BSP version 2.0/3
Created: Nov  8 2006, 05:55:53
ED&R Policy Mode: Deployed
WDB Comm Type: WDB_COMM_PIPE
WDB: Ready

->
```

COMAU

KUKA

SIEMENS

MITSUBISHI ELECTRIC

ABB

NASA

YASKAWA

Motoman Robotics compact
NXC100 robot controller offers high performance, open communication, and integrated cell control
NXC100
ROBOT CONTROLLER

BOSCH

Schneider Electric

WIND

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

USB2Serial

Dashing

Zephyr

ROS 2 Lab

Helix Virtualization Platform

TSN

CES2020

ROS2019

ROS 2

ARM

SDK

Intel

UML

Developer Journey

LLVM

IP Review

GPL

Wind River Linux

LG

UP2

RPI4

Autoware

Wind River Labs

meta-ros

MUSL

EWC2020

AUTOSAR Adaptive

Simulation

Apollo

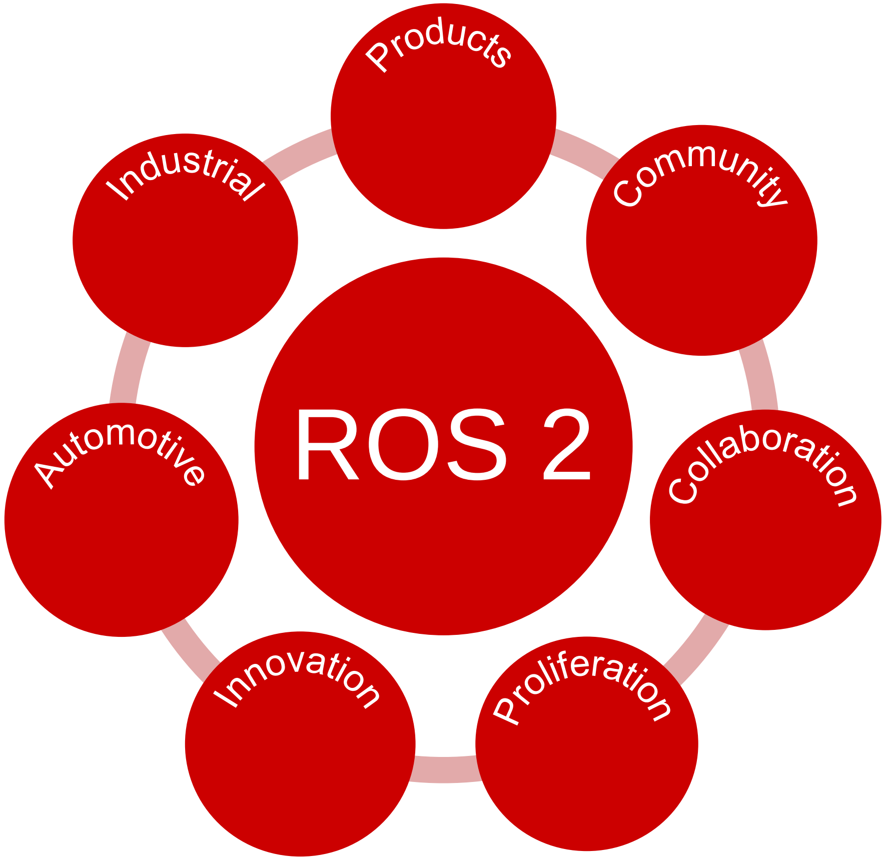
Digital Twin

Webinar

Training

Blog

THE 'RIPPLE' EFFECT OF ROS 2



ROS 2 DASHING RELEASE VXWORKS PORT

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

https://raw.githubusercontent.com/ros2/ros2/release-latest/ros2.repos

- 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

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)

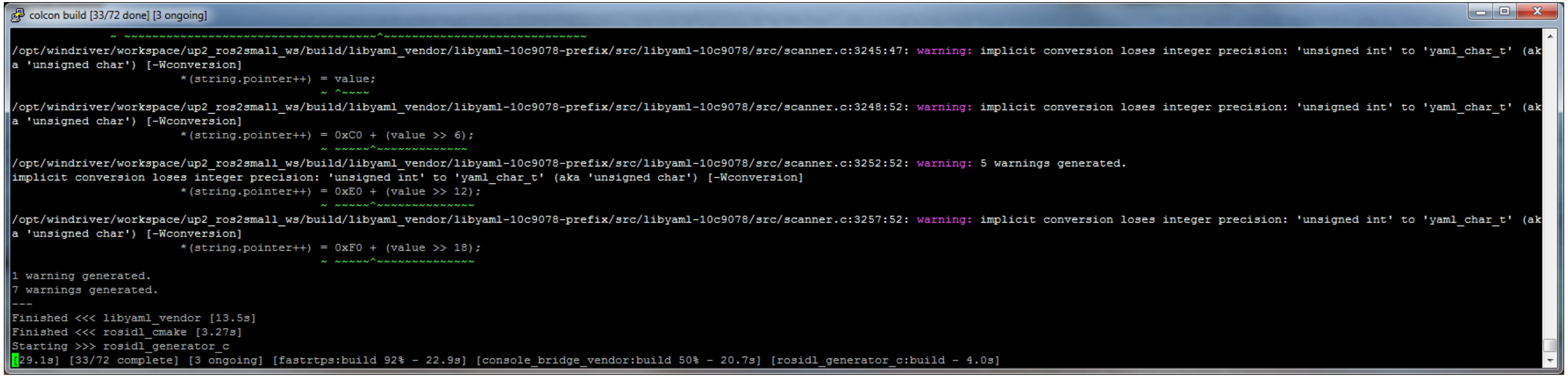
- https://github.com/Wind-River/vxworks7-ros2-build
- Makefile build: BUILD_TYPE=Debug BOARD=up2 make
- Set of scripts to build:
  - bootloader, kernel, userspace, ROS2, and the rootfs (boot from the USB stick)
- Board support:
  - UP, UP2, RPI3/RPI4, VxWorks Simulator, QEMU, and others
- Docker build: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

VXWORKS7-LAYER-FOR-ROS2

- https://github.com/Wind-River/vxworks7-layer-for-ros2
- 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
  - colcon build --symlink-install --cmake-force-configure --cmake-args -DCMAKE_TOOLCHAIN_FILE=$VSB_DIR/buildspecs/cmake/rtp.cmake -DCMAKE_PREFIX_PATH=$PRJ_WS/install;$VSB_DIR/usr/root -DBUILD_TESTING=OFF

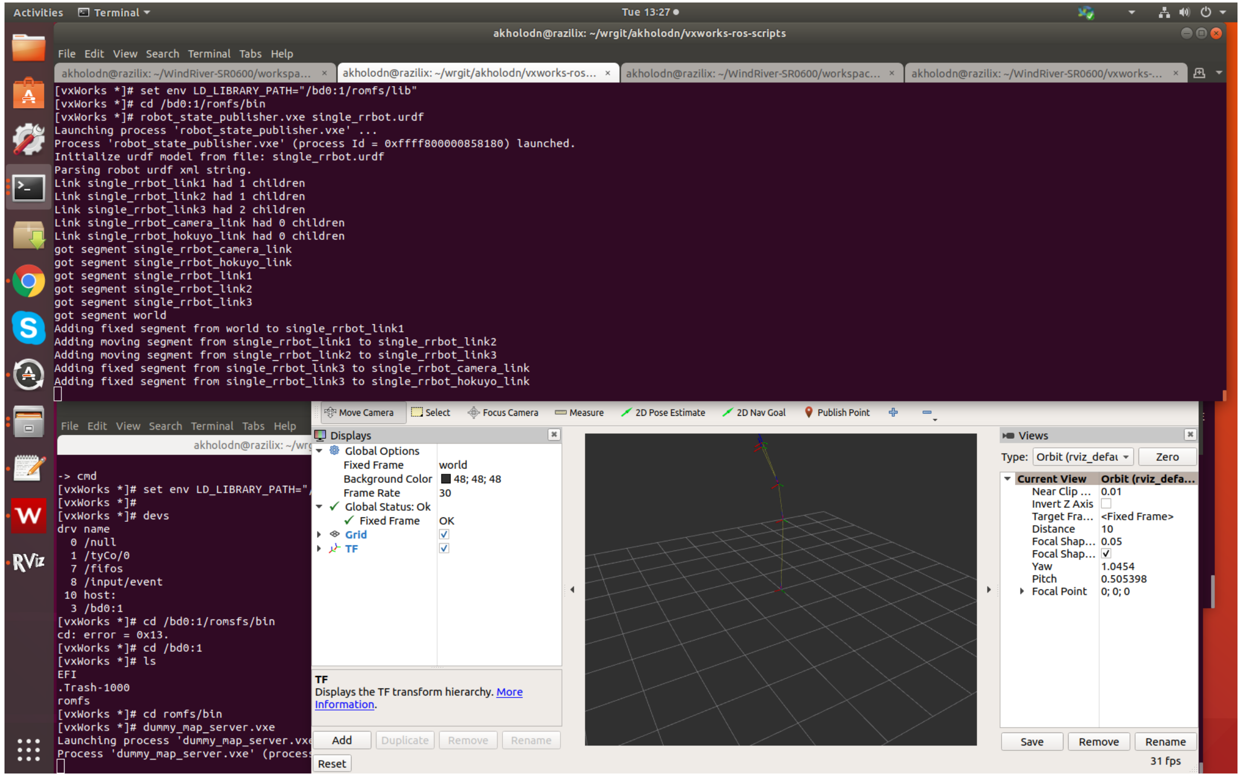


colcon build [33/72 done] [3 ongoing]

```
               ~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3245:47: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                        *(string.pointer++) = value;
                                            ~ ^~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3248:52: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                        *(string.pointer++) = 0xC0 + (value >> 6);
                                            ~ ~~~~~^~~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3252:52: warning: 5 warnings generated.
implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (aka 'unsigned char') [-Wconversion]
                        *(string.pointer++) = 0xE0 + (value >> 12);
                                            ~ ~~~~~^~~~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3257:52: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                        *(string.pointer++) = 0xF0 + (value >> 18);
                                            ~ ~~~~~^~~~~~~~~~~~~~~

1 warning generated.
7 warnings generated.
---
Finished <<< libyaml_vendor [13.5s]
Finished <<< rosidl_cmake [3.27s]
Starting >>> rosidl_generator_c
[29.1s] [33/72 complete] [3 ongoing] [fastrtps:build 92% - 22.9s] [console_bridge_vendor:build 50% - 20.7s] [rosidl_generator_c:build - 4.0s]
```

DUMMY ROBOT



Activities
Terminal
Tue 13:27
akholodn@razilix: ~/wrgit/akholodn/vxworks-ros-scripts
File Edit View Search Terminal Tabs Help
akholodn@razilix: ~/WindRiver-SR0600/workspa...
akholodn@razilix: ~/wrgit/akholodn/vxworks-ros...
akholodn@razilix: ~/WindRiver-SR0600/workspac...
akholodn@razilix: ~/WindRiver-SR0600/vxworks-...

```
[vxWorks *]# set env LD_LIBRARY_PATH="/bd0:1/romfs/lib"
[vxWorks *]# cd /bd0:1/romfs/bin
[vxWorks *]# robot_state_publisher.vxe single_rrbot.urdf
Launching process 'robot_state_publisher.vxe' ...
Process 'robot_state_publisher.vxe' (process Id = 0xffff800000858180) launched.
Initialize urdf model from file: single_rrbot.urdf
Parsing robot urdf xml string.
Link single_rrbot_link1 had 1 children
Link single_rrbot_link2 had 1 children
Link single_rrbot_link3 had 2 children
Link single_rrbot_camera_link had 0 children
Link single_rrbot_hokuyo_link had 0 children
got segment single_rrbot_camera_link
got segment single_rrbot_hokuyo_link
got segment single_rrbot_link1
got segment single_rrbot_link2
got segment single_rrbot_link3
got segment world
Adding fixed segment from world to single_rrbot_link1
Adding moving segment from single_rrbot_link1 to single_rrbot_link2
Adding moving segment from single_rrbot_link2 to single_rrbot_link3
Adding fixed segment from single_rrbot_link3 to single_rrbot_camera_link
Adding fixed segment from single_rrbot_link3 to single_rrbot_hokuyo_link
```

```
-> cmd
[vxWorks *]# set env LD_LIBRARY_PATH="/
[vxWorks *]#
[vxWorks *]# devs
drv name
  0 /null
  1 /tyCo/0
  7 /fifos
  8 /input/event
 10 host:
  3 /bd0:1
[vxWorks *]# cd /bd0:1/romsfs/bin
cd: error = 0x13.
[vxWorks *]# cd /bd0:1
[vxWorks *]# ls
EFI
.Trash-1000
romfs
[vxWorks *]# cd romfs/bin
[vxWorks *]# dummy_map_server.vxe
Launching process 'dummy_map_server.vxe
Process 'dummy_map_server.vxe' (process
```

Global Options	
Fixed Frame	world
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
TF	✓

TF
Displays the TF transform hierarchy. More Information.

Type: Orbit (rviz_defau

Current View	Orbit (rviz_defa...
Near Clip ...	0.01
Invert Z Axis	☐
Target Fra...	<Fixed Frame>
Distance	10
Focal Shap...	0.05
Focal Shap...	✓
Yaw	1.0454
Pitch	0.505398
Focal Point	0; 0; 0

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

WIND™

WHEN IT MATTERS, IT RUNS ON WIND RIVER.