ROS2: The evolution of Robot Operative System

Jose Luis Rivero
Open Robotics

Open Robotics

We create open software and hardware platforms for robotics. We use those platforms to solve important problems and we help others to do the same.

open robotics

The origin of ROS 1.x

First release 2007 - Ongoing

ROS: open source SDK to build robot software

ROS principles

1 **Distribution**

2 **Abstraction**

3 **Introspection**

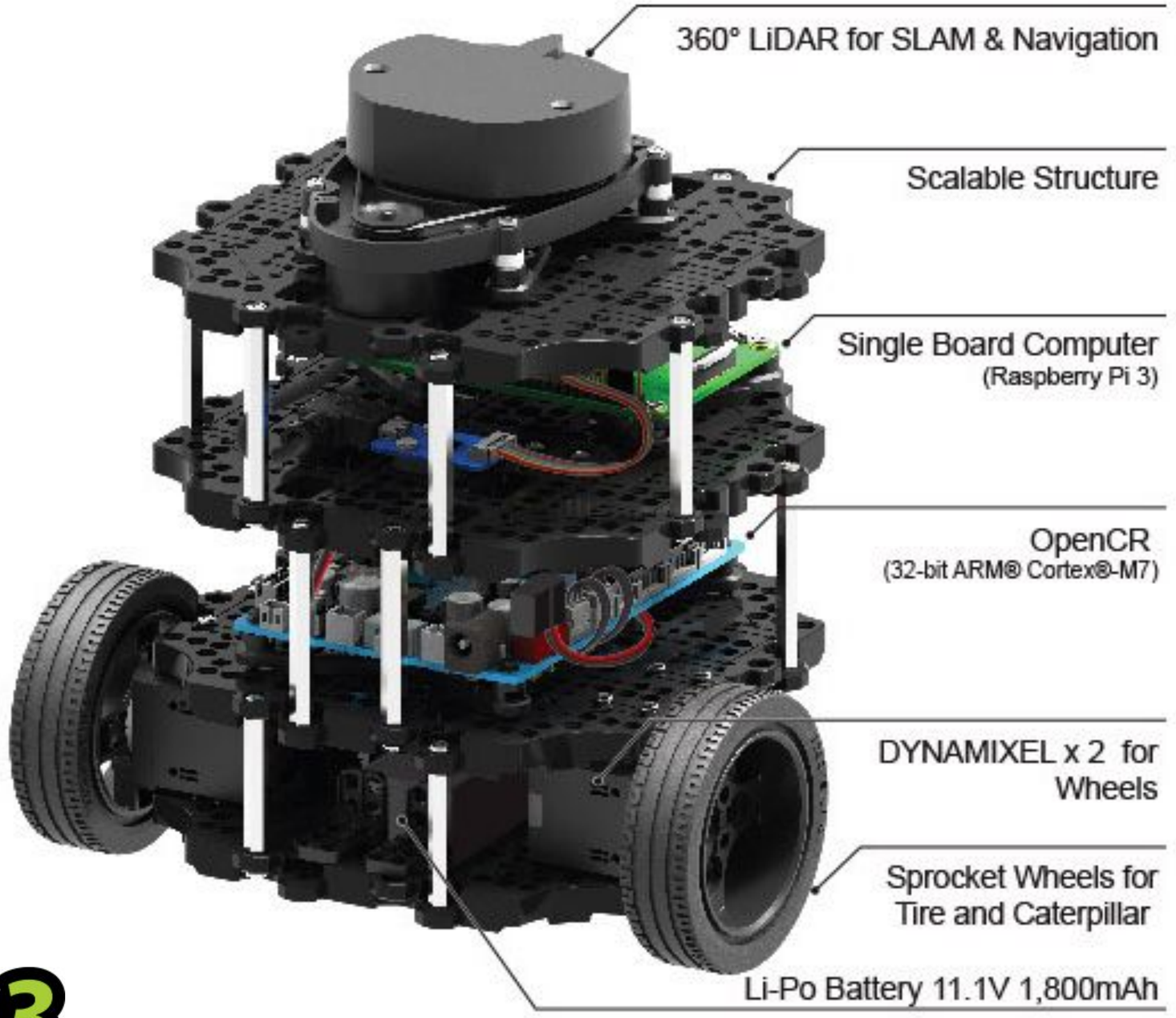
4 **Federation**

ROS

open robotics

19.2cm

Price: ~1100€
Weight: 1Kg

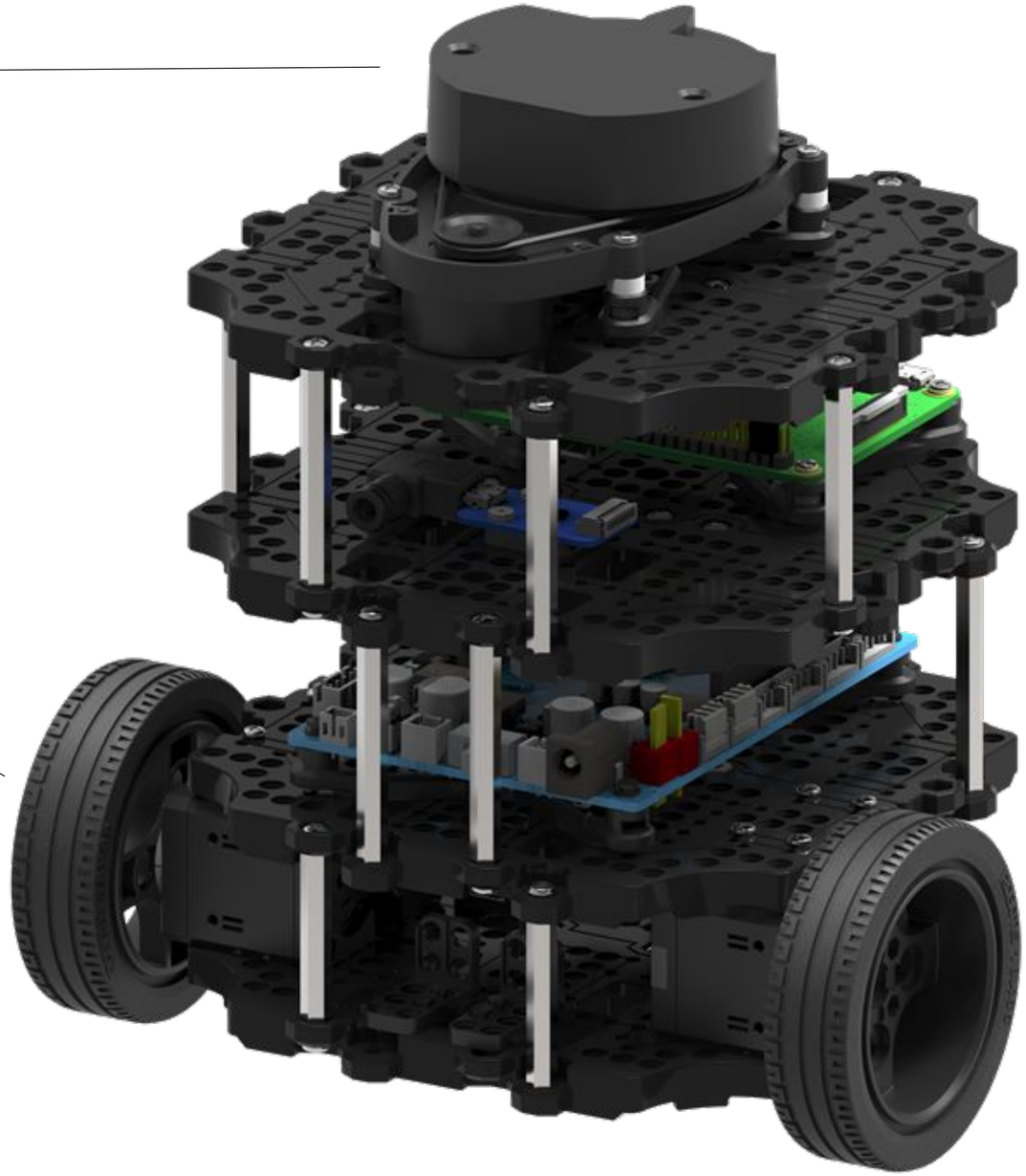


TURTLEBOT3

360° Laser
Distance Sensor
LDS-01

Wheels
(Dynamixel actuators)

Remote
controller



open robotics

Remote controller

ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Documentation Browse Software News

ps3joy

indigo kinetic lunar melodic *Show EOL distros:* ☐

Documentation Status

***joystick_drivers**: joy | ps3joy | spacenav_node | wiimote*

Package Summary

✔ Released ✔ Continuous Integration: 1 / 1 ✔ Documented

Playstation 3 SIXAXIS or DUAL SHOCK 3 joystick driver. Driver for the Sony PlayStation 3 SIXA UAL SHOCK 3 joysticks. In its current state, this driver is not compatible with r Bluetooth HID devices. The driver listens for a connection on the HID ports, streaming data, and passes the data to the Linux uinput device so that it rmal joystick.

Package Links

Code API
Tutorials
Troubleshooting
FAQ
Changelog
Change List
Reviews

Dependencies (5)
Used by (4)

open robotics

ROS

ROS MASTER

naming and registration

register node and /joy

register receive metadata

joy_node

publish

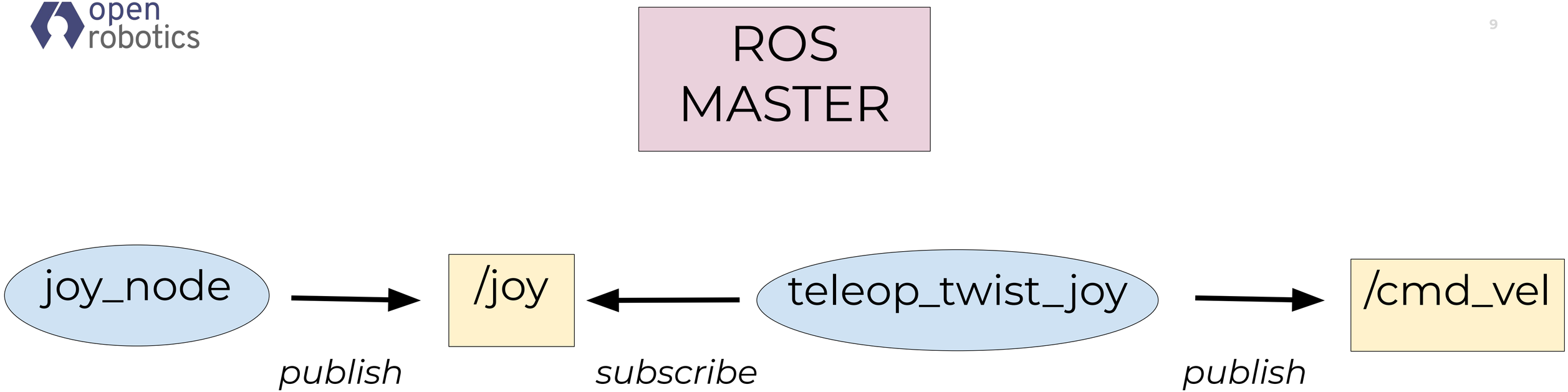
/joy

subscribe

teleop_twist_joy

publish

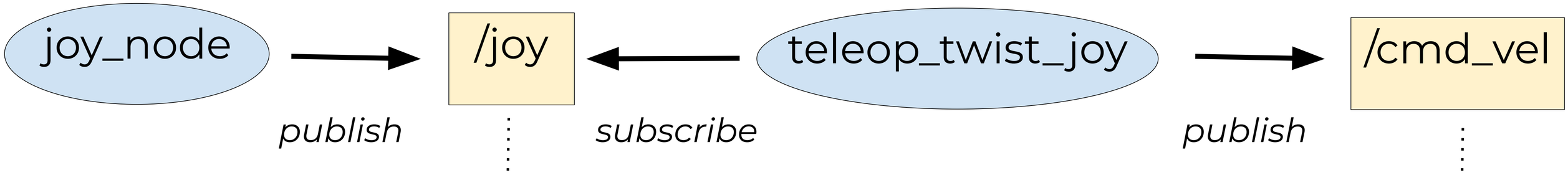
/cmd_vel



1 **Distribution**

ROS

- Publish / subscribe messaging
- Discovery, transport, serialization
- Isolate components from each other
- Allow independent development



Raw Message Definition

```
# Reports the state of a joysticks axes and buttons
Header header                # timestamp in the header i
float32[] axes               # the axes measurements fro
int32[] buttons              # the buttons measurements
```

Raw Message Definition

```
# This expresses velocity in free space
Vector3  linear
Vector3  angular
```

2 **Abstraction**

ROS

- Well-defined interfaces
- Syntax & semantics
- Tools target generic interfaces
- Support multiple languages

360° Laser Distance Sensor LDS-01

ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.r

Documentation Browse Software News

hls_lfcd_lds_driver

kinetic melodic

Documentation Status

Package Summary

✔ Released ✔ Continuous Integration ✔ Documented

ROS package for LDS(HLS-LFCD2). The LDS (Laser Distance Sensor) is a sensor sending the data to Host for the simultaneous localization and mapping (SLAM). Simultaneously the detecting obstacle data can also be sent to Host. HLDS(Hitachi-LG Data Storage) is developing the technology for the moving platform sensor such as Robot Vacuum Cleaners, Home Robot, Robotics Lawn Mower Sensor, etc.

Package
Code AP
FAQ
Changelo
Change Li
Reviews

Dependencies (4)
Used by (1)
Jenkins jobs (9)

ROS MASTER

laser_lds → /scan

publish

Raw Message Definition

```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header               # timestamp in the header is the acquisition time
                            # the first ray in the scan.
                            #
                            # in frame frame_id, angles are measured around
                            # the positive Z axis (counterclockwise, if Z is
                            # with zero angle being forward along the x axis

float32 angle_min           # start angle of the scan [rad]
float32 angle_max           # end angle of the scan [rad]
float32 angle_increment     # angular distance between measurements [rad]

float32 time_increment      # time between measurements [seconds] - if your s
                            # is moving, this will be used in interpolating p
                            # of 3d points
float32 scan_time           # time between scans [seconds]

float32 range_min           # minimum range value [m]
float32 range_max           # maximum range value [m]

float32[] ranges            # range data [m] (Note: values < range_min or > r
float32[] intensities       # intensity data [device-specific units].  If you
                            # device does not provide intensities, please lea
                            # the array empty.
```

open robotics

turtlebot3_fake.rviz* - RViz

Interact | Move Camera | Select | Focus Camera | Measure | 2D Pose Estimate | 2D Nav Goal | Publish Point

Displays

- Global Options
  - Fixed Frame: odom
  - Background Color: 255; 255; 255
  - Frame Rate: 30
- Global Status: Ok
- Grid
- Axes
- RobotModel
  - Status: Ok
  - Visual Enabled
  - Collision Enabled
  - Update Interval: 0
  - Alpha: 1
  - Robot Description: robot_description
  - TF Prefix
  - Links
- Odometry
  - Status: Ok
  - Topic: /odom
  - Color: 255; 25; 0
  - Position Tolerance: 0.1
  - Angle Tolerance: 0.1
  - Keep: 100
  - Length: 0.3
- TF

Remove | Rename

Views

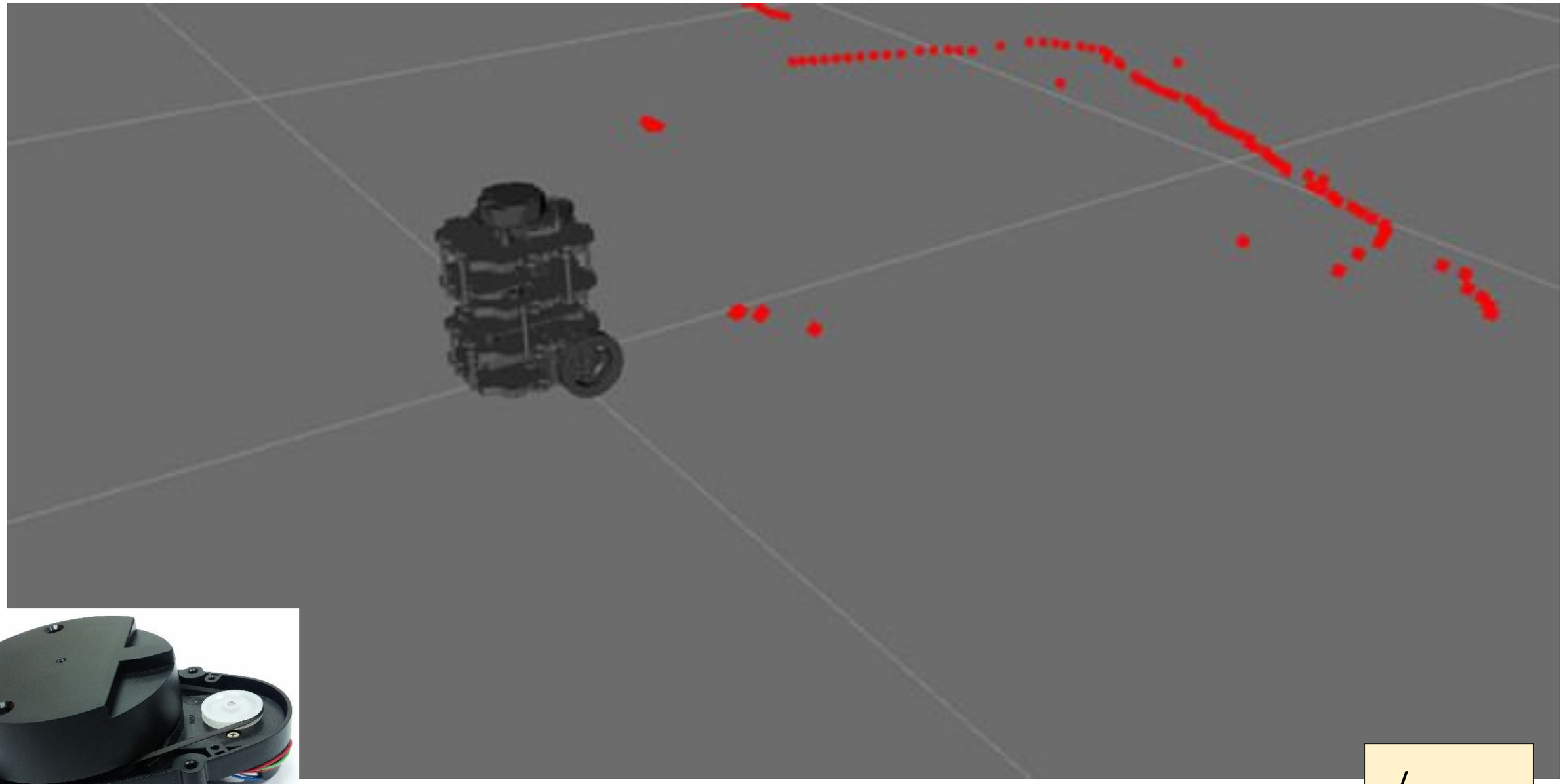
Type: XYOrbit (rviz) | Zero

Current View	XYOrbit (rviz)
Near Clip ...	0.01
Target Fr...	<Fixed Frame>
Distance	0.403429
Focal Sha...	0.05
Focal Sha...	
Yaw	0.785398
Pitch	0.785398
Focal Point	0.021407; 0.01...

Save | Remove | Rename

477.25 | ROS Elapsed: 15.55 | Wall Time: 1496356477.29 | Wall Elapsed: 15.49 | Experimental

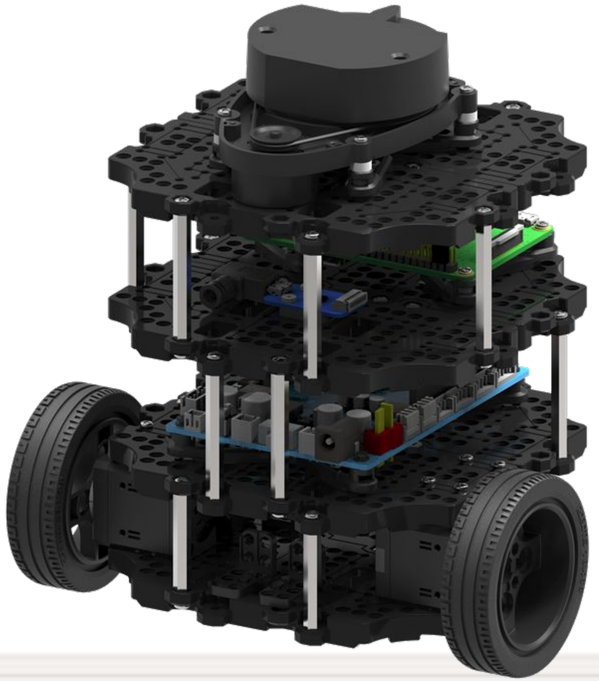
otate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift**: More options. | 31 fps



/scan

Topic Monitor

Topic	Type	Bandwidth	Hz	Value
/diagnostics	diagnostic_msgs/DiagnosticArray	276.41B/s	1.00	
header	std_msgs/Header			
status	diagnostic_msgs/DiagnosticStatus[]			
[0]	diagnostic_msgs/DiagnosticStatus			
hardware_id	string			'MPU9250'
level	byte			0
message	string			'Good Condition'
name	string			'IMU Sensor'
values	diagnostic_msgs/KeyValue[]			[]
[1]	diagnostic_msgs/DiagnosticStatus			
hardware_id	string			'DYNAMIXEL X'
level	byte			0
message	string			'Torque ON'
name	string			'Actuator'
values	diagnostic_msgs/KeyValue[]			[]
[2]	diagnostic_msgs/DiagnosticStatus			
hardware_id	string			'HLS-LFCD-LDS'
level	byte			0
message	string			'Good Condition'
name	string			'Lidar Sensor'
values	diagnostic_msgs/KeyValue[]			[]
[3]	diagnostic_msgs/DiagnosticStatus			
hardware_id	string			'Battery'
level	byte			0
message	string			'Good Condition'
name	string			'Power System'
values	diagnostic_msgs/KeyValue[]			[]
[4]	diagnostic_msgs/DiagnosticStatus			
hardware_id	string			'OpenCR Button'
level	byte			0
message	string			'Pushed Nothing'
name	string			'Analog Button'

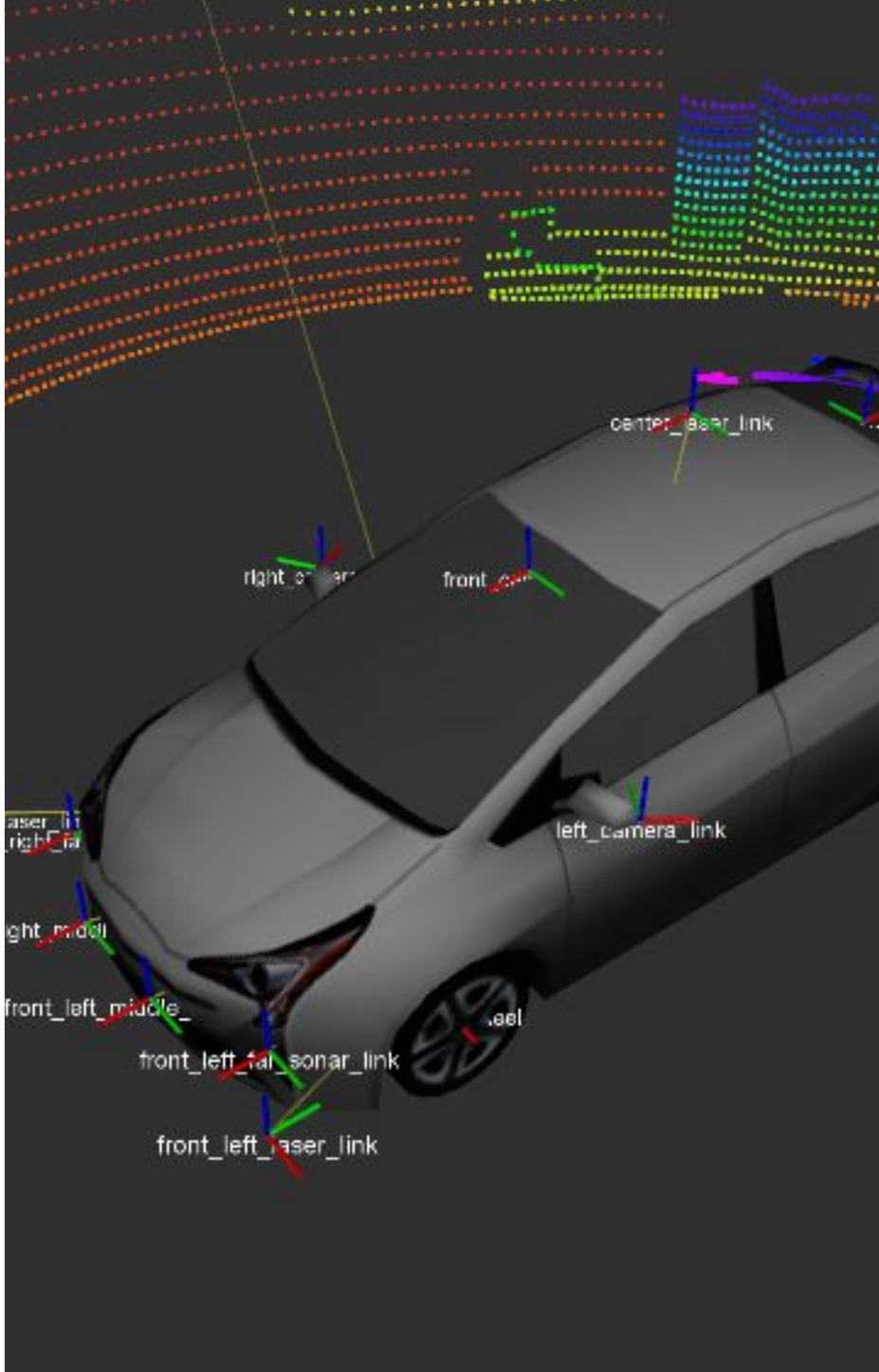


open robotics

3 Introspection

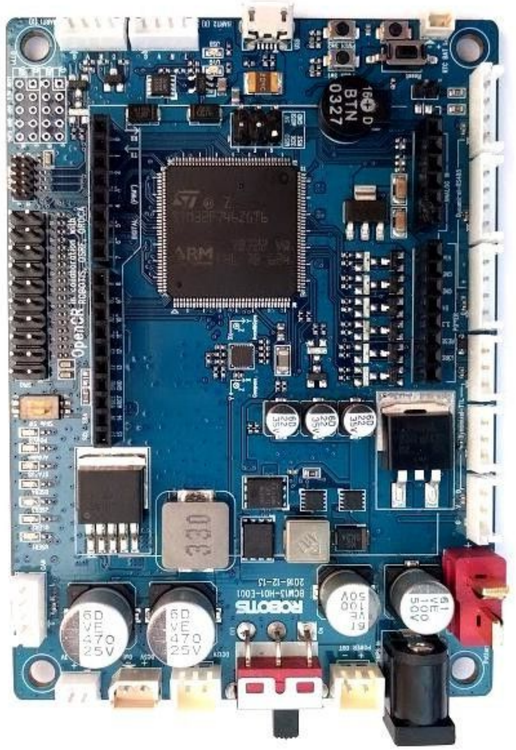
ROS

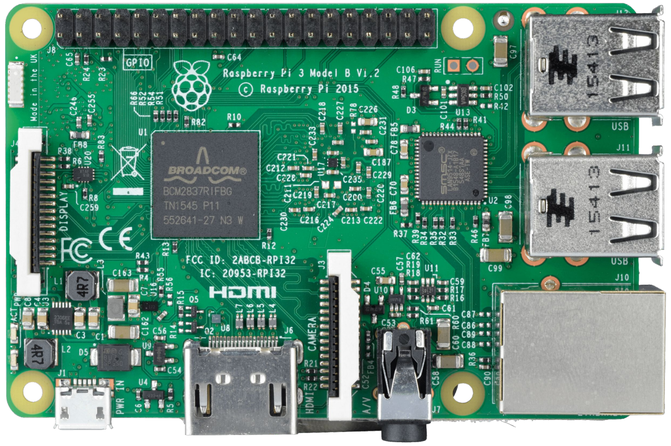
- All important data on the message bus
- Support incremental system exploration
- GUIs are always external tools
- Could build apps from CLI tools

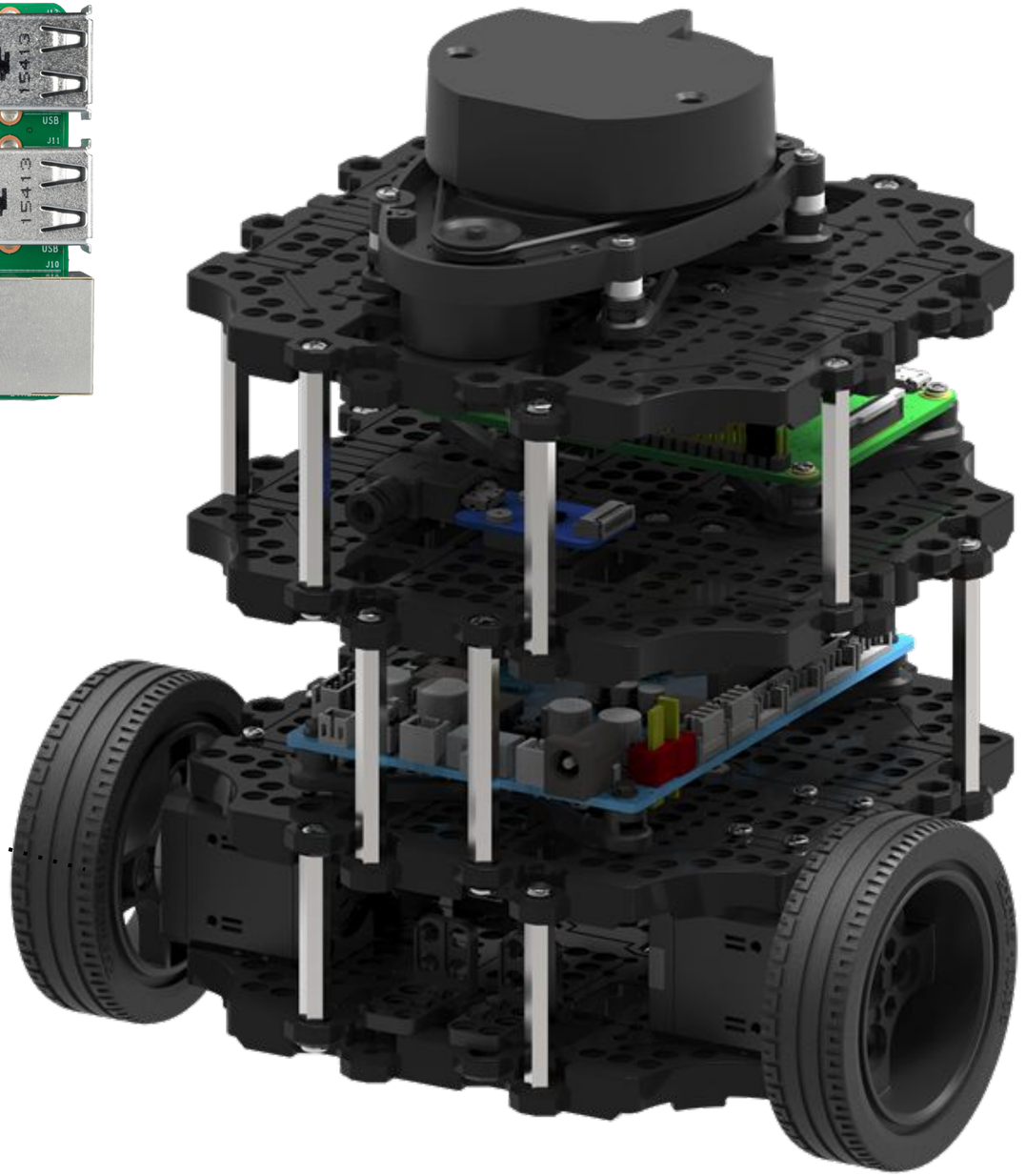


RaspBerry Pi

32 bits
Microcontroller







Wheels (Dynamixel actuators)

ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Documentation Browse Software News

rosserial

indigo kinetic lunar melodic *Show EOL distros:*

Documentation Status

***rosserial**: rosserial_client | rosserial_msgs | rosserial_python*

Package Summary

✔ Released ✔ Continuous Integration: 12 / 12 ✔ Documented

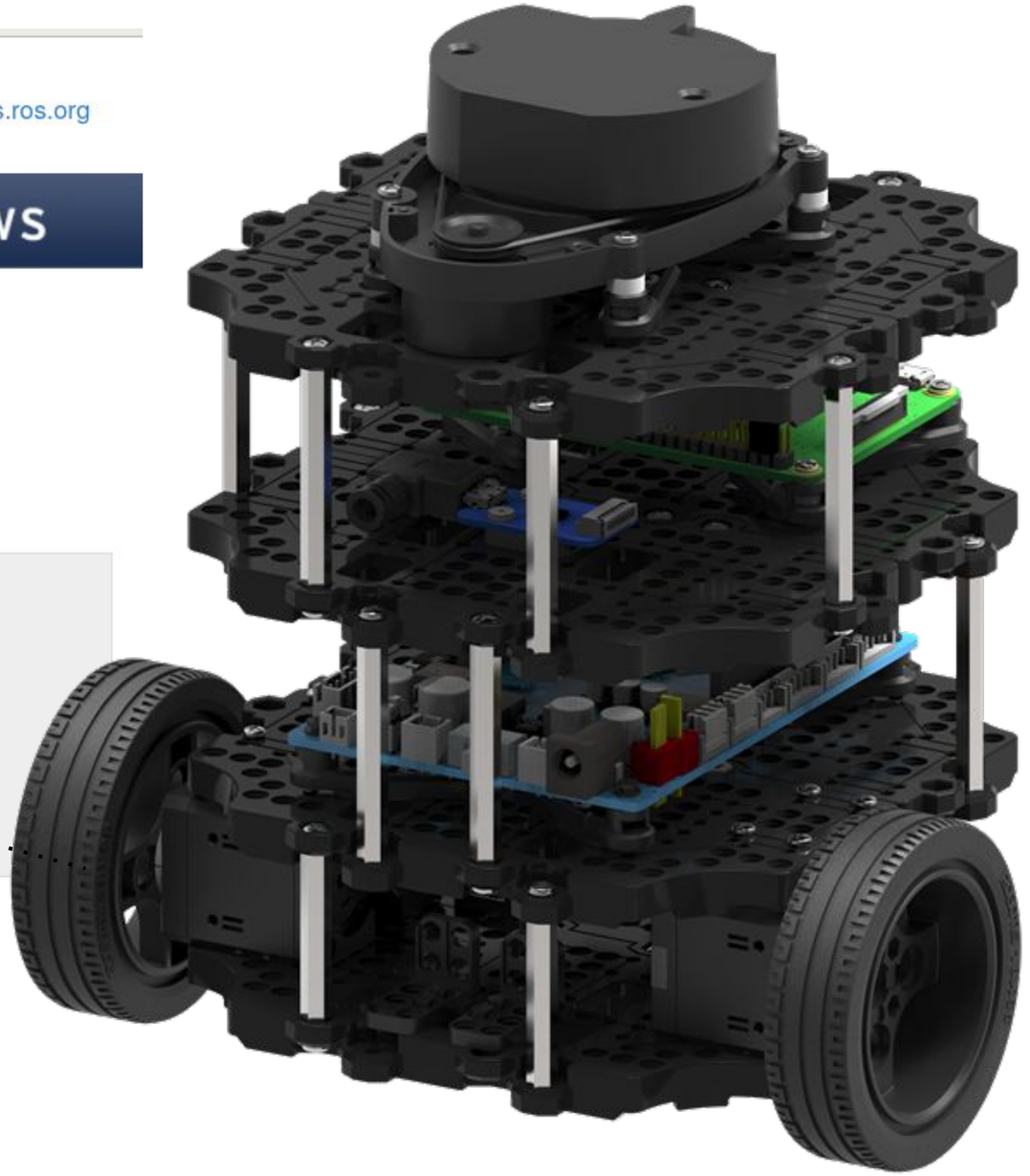
Metapackage for core of rosserial.

- Maintainer status: maintained
- Maintainer: Paul Bouchier <paul.bouchier AT gmail DOT com>, [illegible]s AT clearpathrobotics DOT com>
- Author: Michael Ferguson
- License: BSD
- Source: git https://github.com/ros-drivers/rosserial.git (branch: melodic-devel)

Package Links

Tutorials
FAQ
Changelog
Change List
Reviews

Dependencies (4)
Jenkins jobs (9)



Wheels (Dynamixel actuators)

open robotics

ROS Index BETA

ABOUT IN

ROS.org About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Documentation Browse Software News

rosserial

ROS.org About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Documentation Browse Software News

ps3joy

ROS.org About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Documentation Browse Software News

hls_lfcd_lds_driver

kinetic melodic

Documentation Status

Package Summary

Released Continuous Integration Documented

ROS package for LDS(HLS-LFCD2). The LDS (Laser Distance Sensor) is a sensor sending the data to Host for the simultaneous localization and mapping (SLAM). Simultaneously the detecting obstacle data can also be sent to Host. HLDS(Hitachi-LG Data Storage) is developing the technology for the moving platform sensor such as Robot Vacuum Cleaners, Home Robot, Robotics Lawn Mower Sensor, etc.

Package Links
Code API
FAQ
Changelog
Change List
Reviews

Dependencies (4)
Used by (1)
Jenkins jobs (9)

Playstation 3 SIX
SIXAXIS or DUA
the use of other E
starts the joystick
shows up as a nc

Home > Repos

DASHING CRYSTAL BOUNCY

Repositories

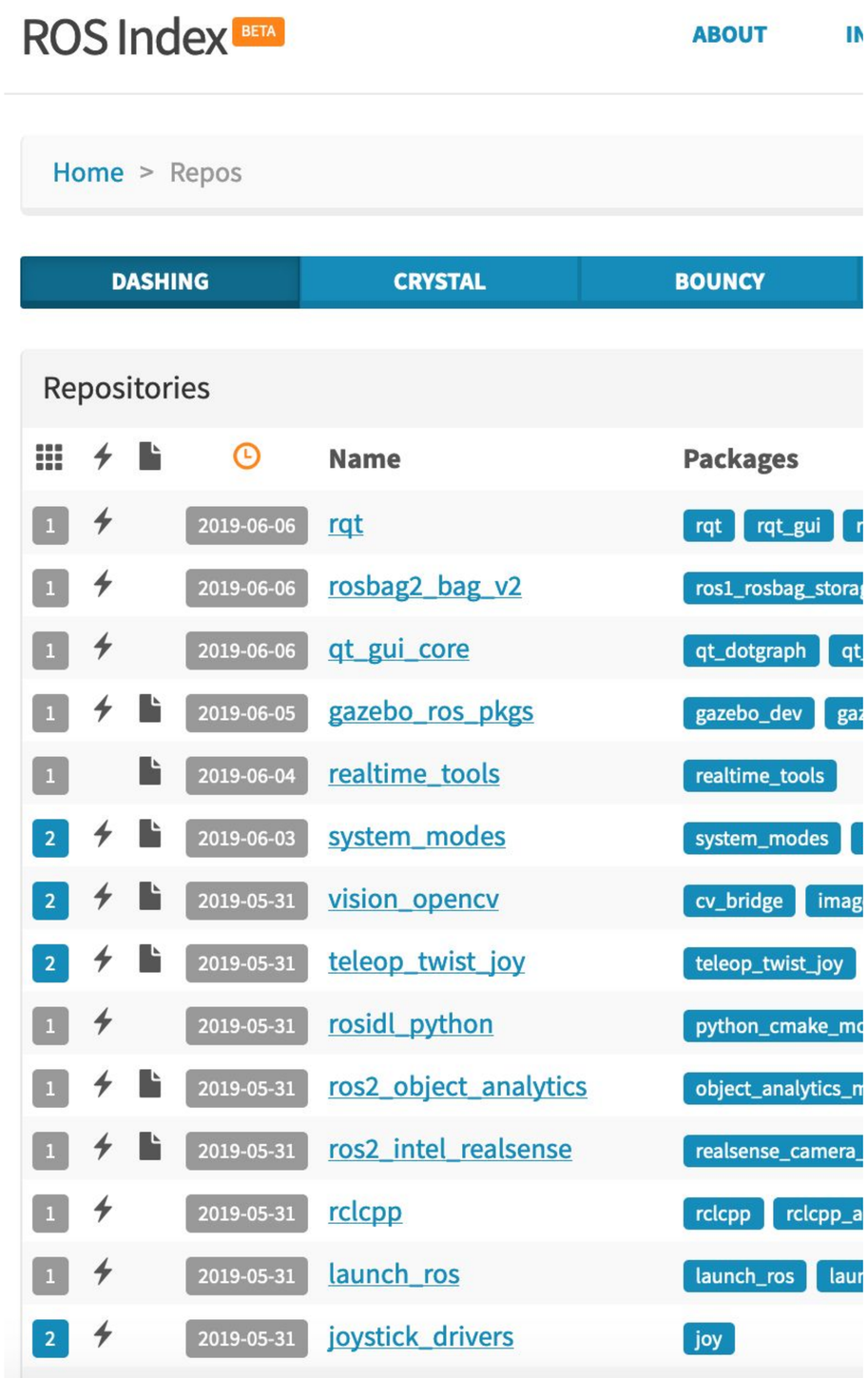
			Name	Packages
1	2019-06-06	rqt	rqt rqt_gui
1	2019-06-06	rosbag2_bag_v2	ros1_rosbag_stora
1	2019-06-06	qt_gui_core	qt_dotgraph qt
1	2019-06-05	gazebo_ros_pkgs	gazebo_dev gaz
1	2019-06-04	realtime_tools	realtime_tools
2	2019-06-03	system_modes	system_modes
2	2019-05-31	vision_opencv	cv_bridge imag
2	2019-05-31	teleop_twist_joy	teleop_twist_joy
1	2019-05-31	rosidl_python	python_cmake_mo
1	2019-05-31	ros2_object_analytics	object_analytics_m
1	2019-05-31	ros2_intel_realsense	realsense_camera_
1	2019-05-31	rclcpp	rclcpp rclcpp_a
1	2019-05-31	launch_ros	launch_ros laun
2	2019-05-31	joystick_drivers	joy

open robotics

4 Federation

ROS

- Let the code live where authors prefer
- Allow for independent releases
- Lower barrier for contributors
- Authors have control, branding, credit
- Requires tools for devs and CI



The next generation

First release 2018 - Ongoing

ROS 2: Goals

1 **Quality of design & implementation**

2 **System reliability**

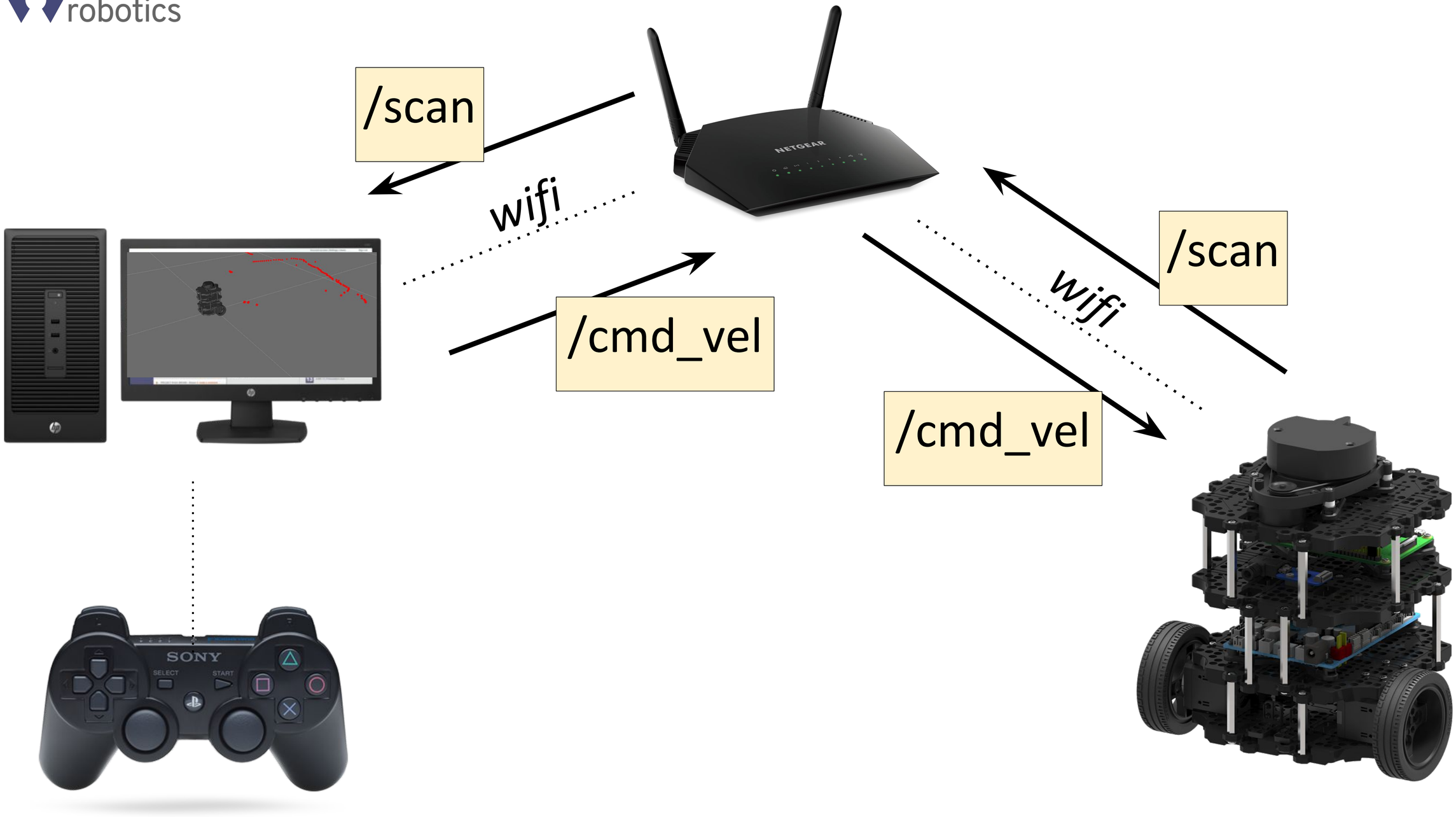
3 **Real-time control & deterministic execution**

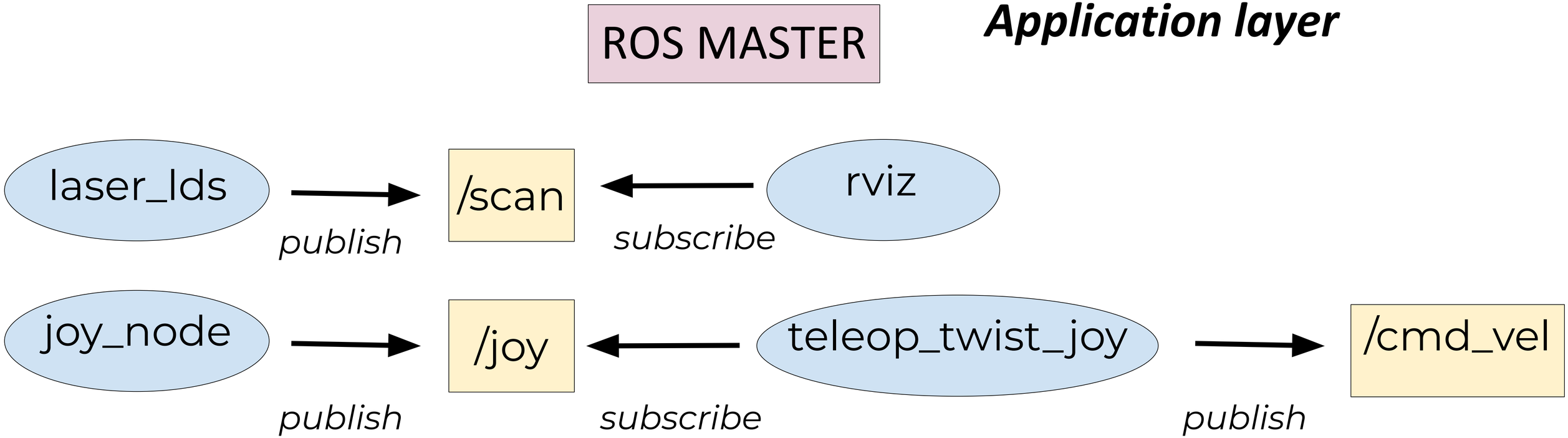
4 **Validation, verification, and certification**

5 **Flexibility in communication**

6 **Support for small embedded systems**

Use case:
unstable networks or high latency scenarios


/scan
wifi
/cmd_vel
/scan
wifi
/cmd_vel
NETGEAR
SONY
SELECT
START



Transport layer

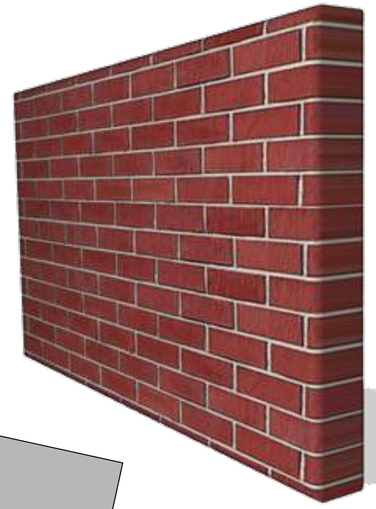
TCPROS / UDPROS

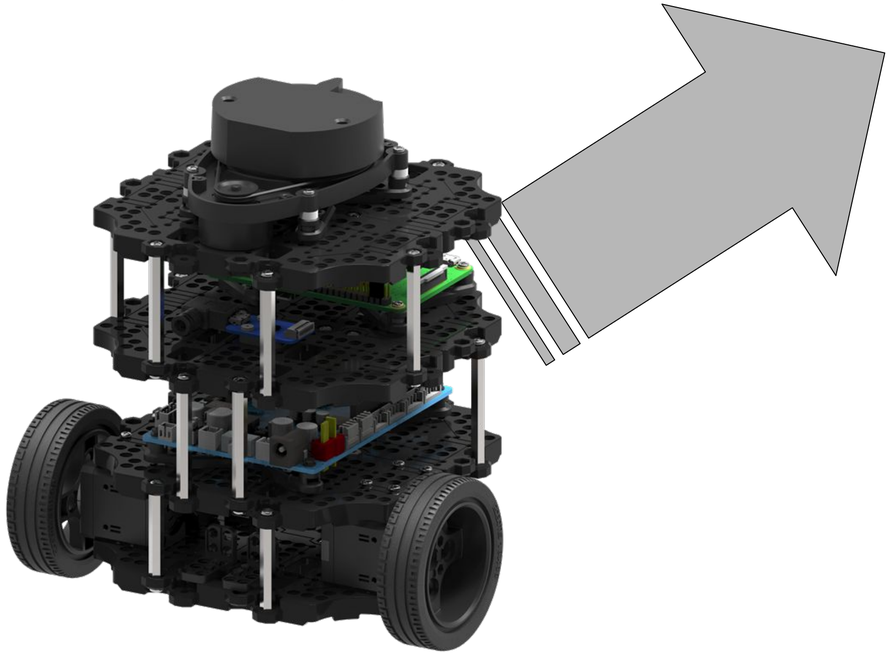
wifi

/cmd_vel

wifi

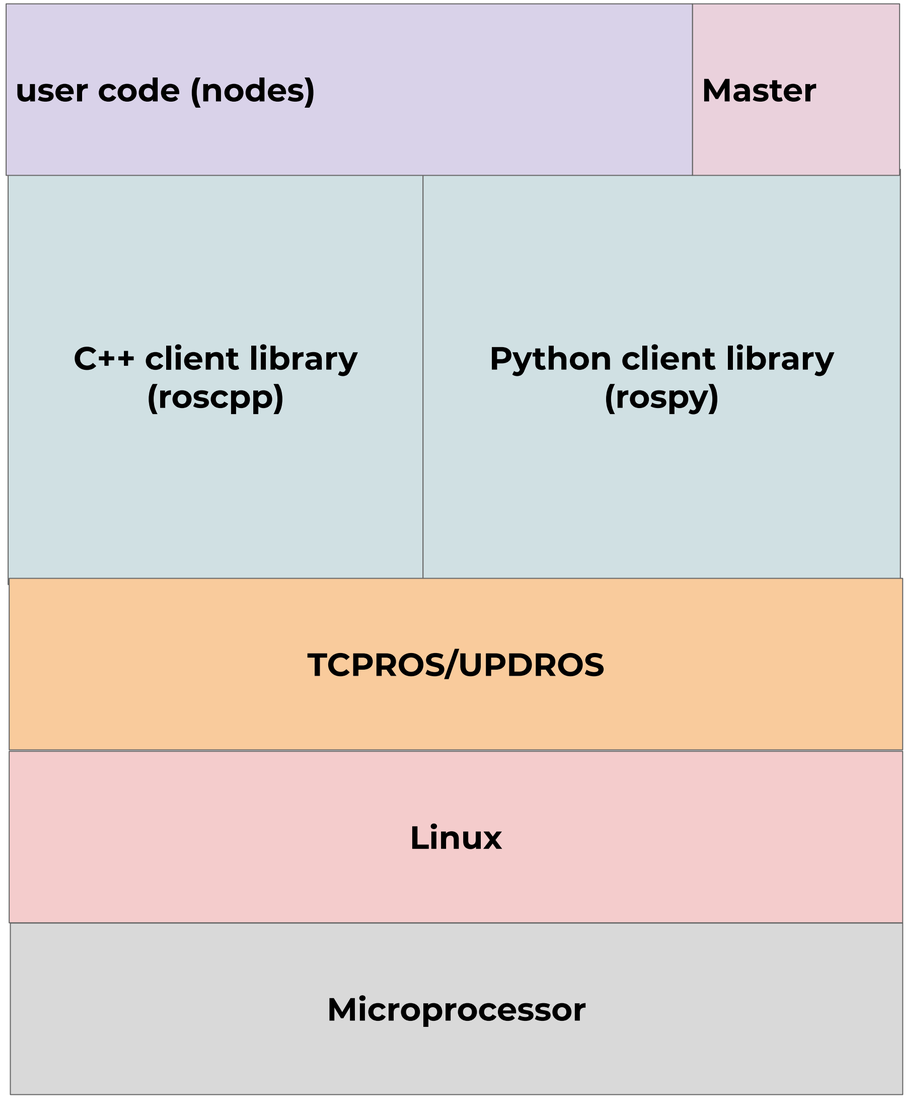
/scan



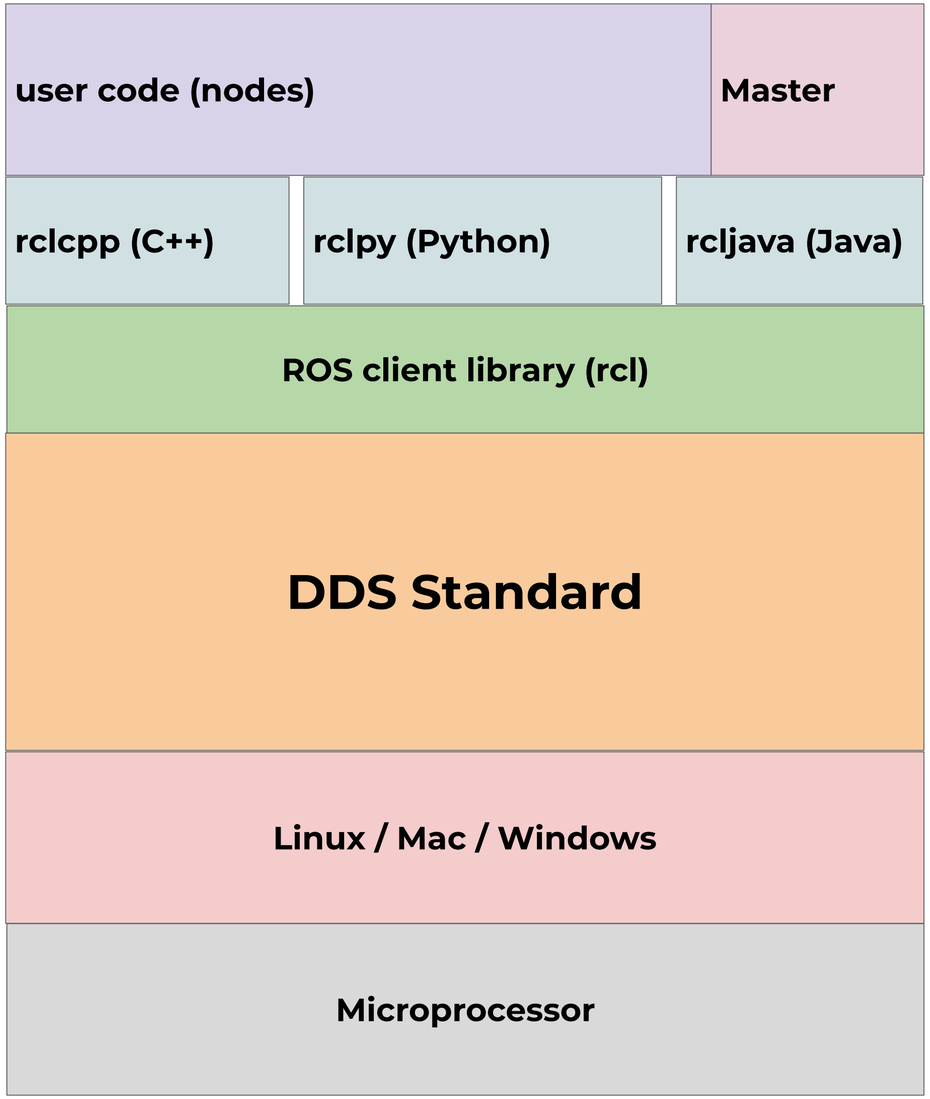


Architectural changes

ROS 1



Towards ROS2



open robotics

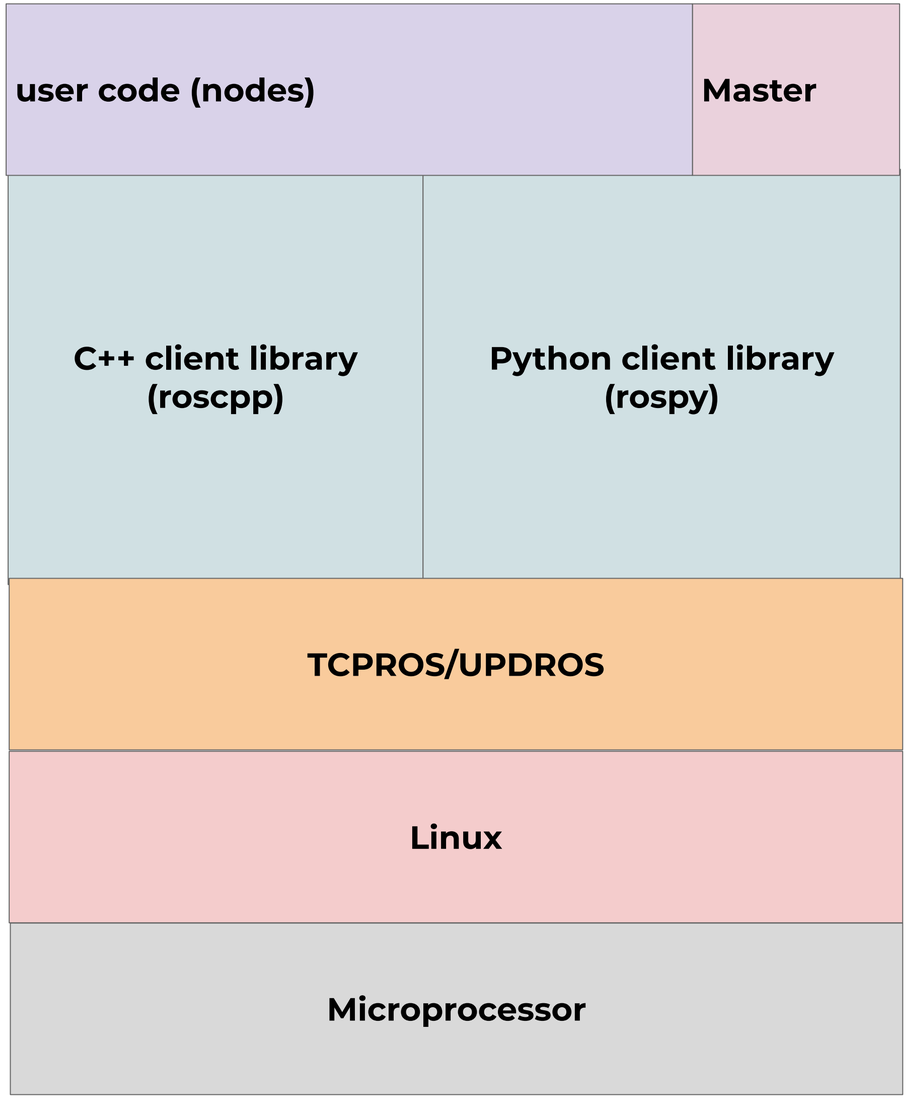
Data-Distribution Service Standard

"The OMG Data-Distribution Service for Real-Time Systems (DDS) is the first open international middleware standard directly addressing **publish-subscribe communications for real-time and embedded systems.**"

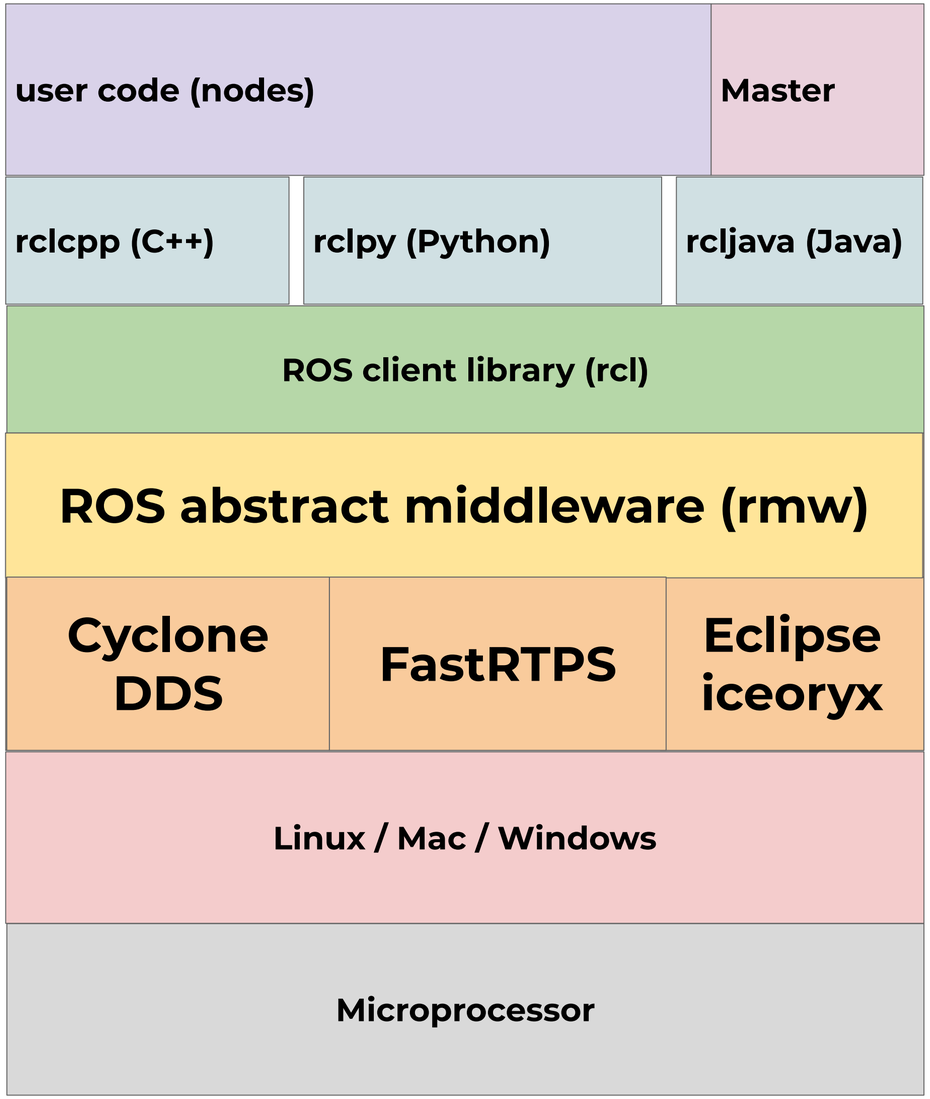
"... DDS features fine and **extensive control of QoS parameters, including reliability, bandwidth, delivery deadlines, and resource limits.** ..."

Architectural changes

ROS 1



Towards ROS2



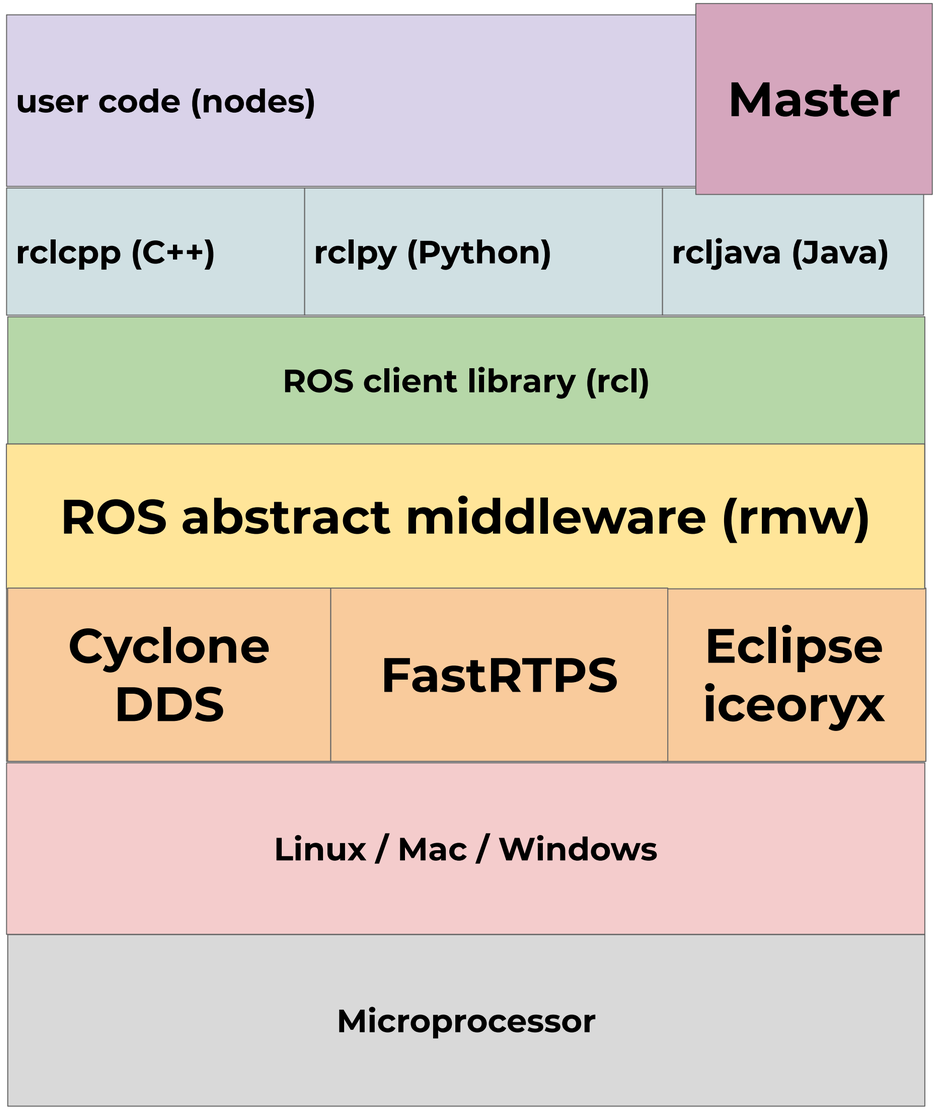
Use case:
manage groups of robots

ROS MASTER

Where? ....

Architectural changes

ROS2



DDS™

"Effective data communication between publishers and subscribers **requires dynamic and reliable discovery of publisher/subscriber endpoints in the system, which DDS currently supports** viaa standardized approach called the Simple Discovery Protocol (SDP)"[1]

[1] https://www.dre.vanderbilt.edu/~schmidt/PDF/debs2014_submission_163.pdf

Use case:
small systems as
first-class ROS systems




	8/16-bit MCU	32-bit MCU	ARM A-class smartphone without screen	SFF x86 laptop without screen
Example Chip	Atmel AVR	STM32	Samsung Exynos	Intel Core i5
Example System	Arduino Leonardo	Pixhawk PX4	ODROID	Intel NUC
MIPS	10's	100's	1000's	10000's
RAM	1-32 KB	4-256 KB	a few GB (off-chip)	2-16 GB (SODIMM)
Max power	10's of mW	100's of mW	1000's of mW	10000's of mW
Comms peripherals	UART, USB FS, ...	USB HS, Ethernet	Gigabit Ethernet	USB SS, PCIe

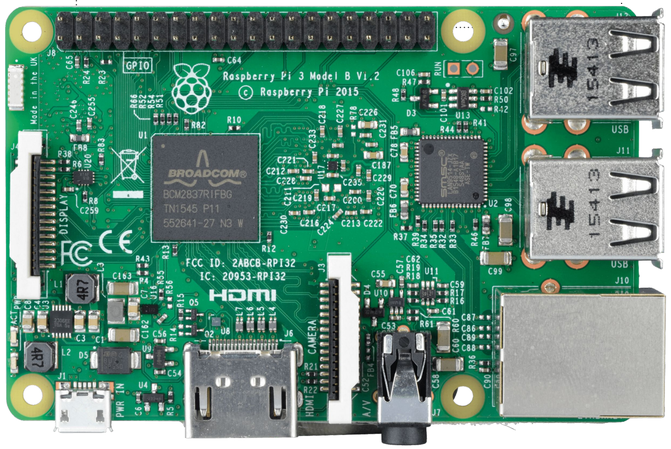
Future work

"normal" ROS2

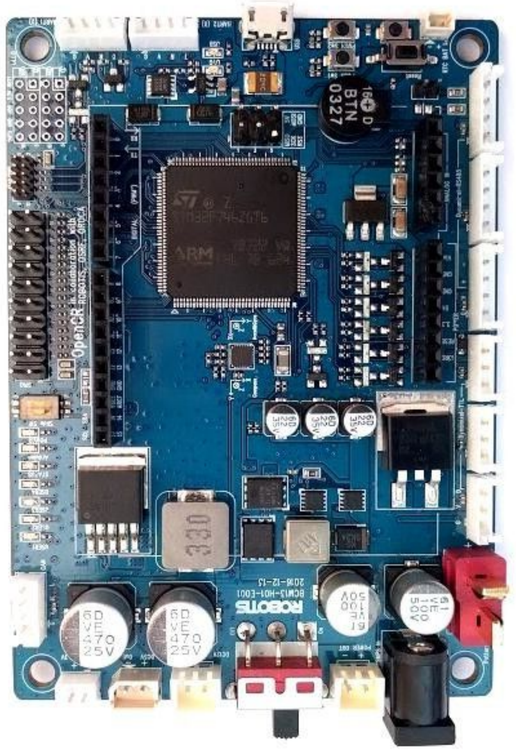
https://roscon.ros.org/2015/presentations/ros2_on_small_embedded_systems.pdf

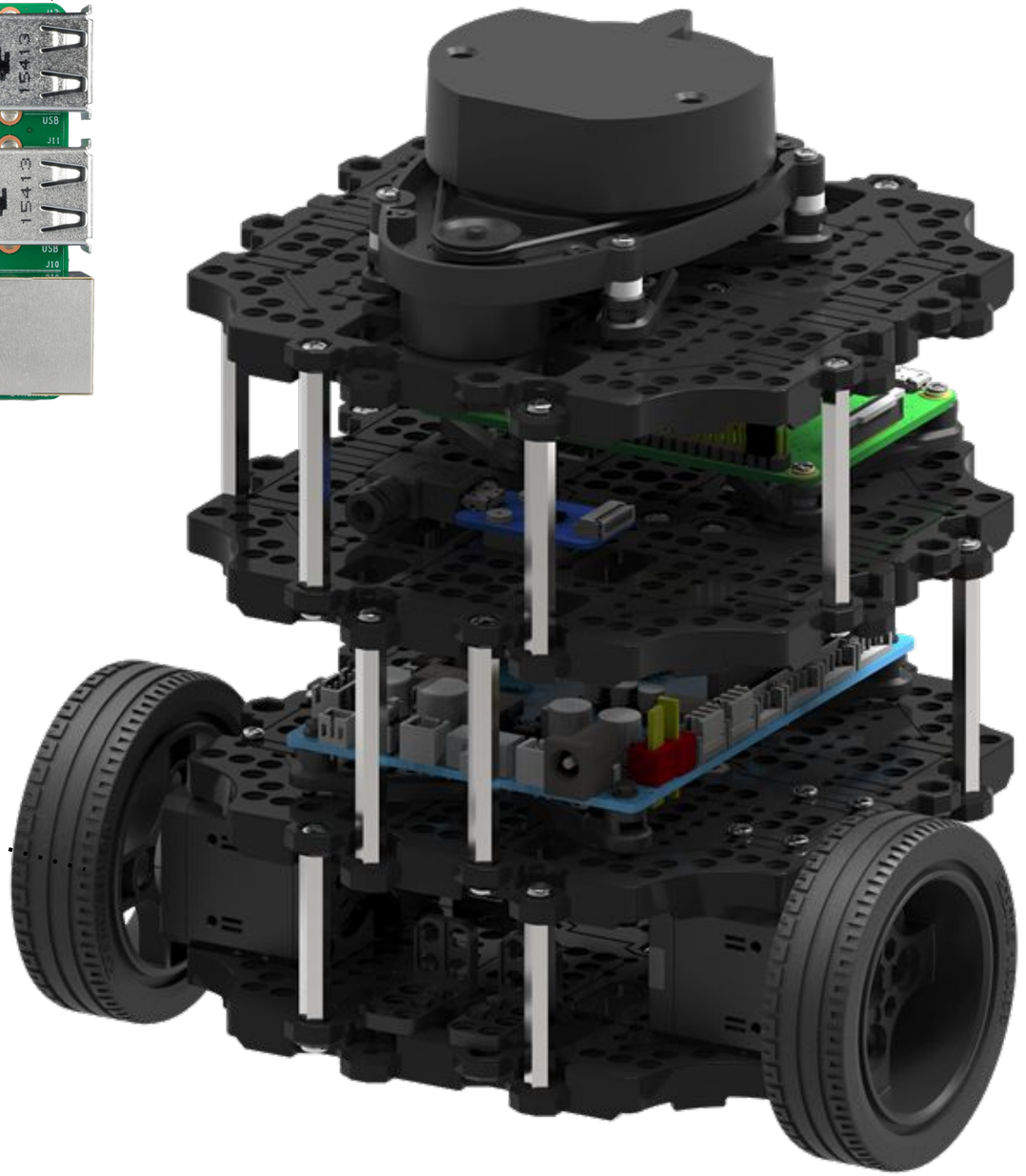
open robotics

RaspBerry Pi



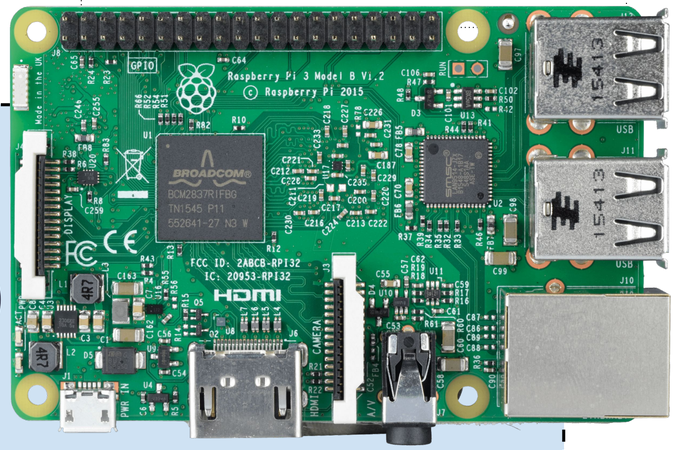
32 bits Microcontroller



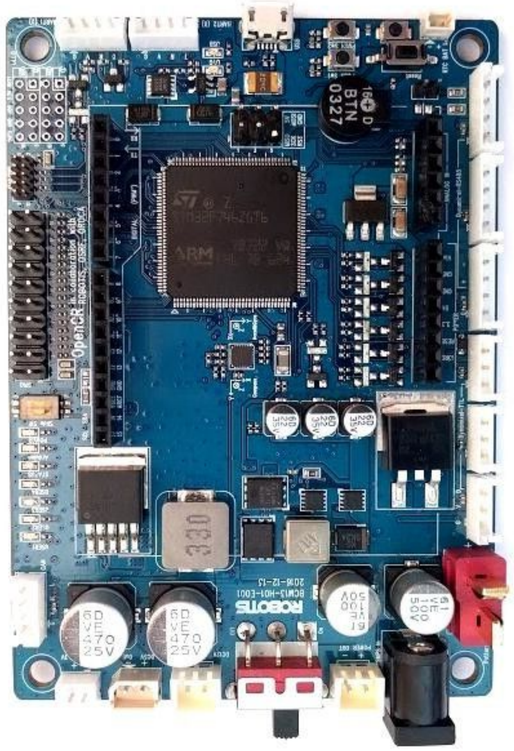


Wheels (Dynamixel actuators)

RaspBerry Pi



32 bits Microcontroller



rosserial protocol

battery state
IMU values

velocity and servo commands

ROS

rosserial

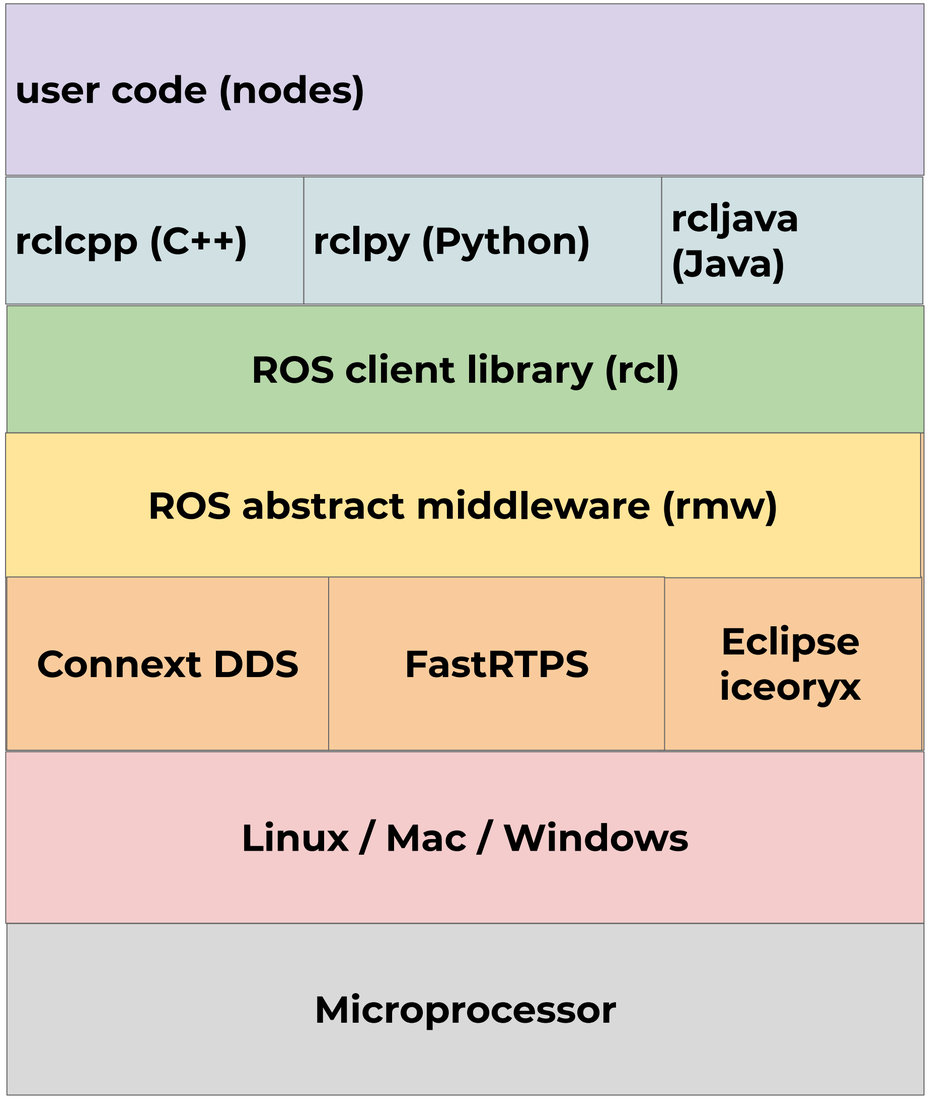
publish

/battery_state

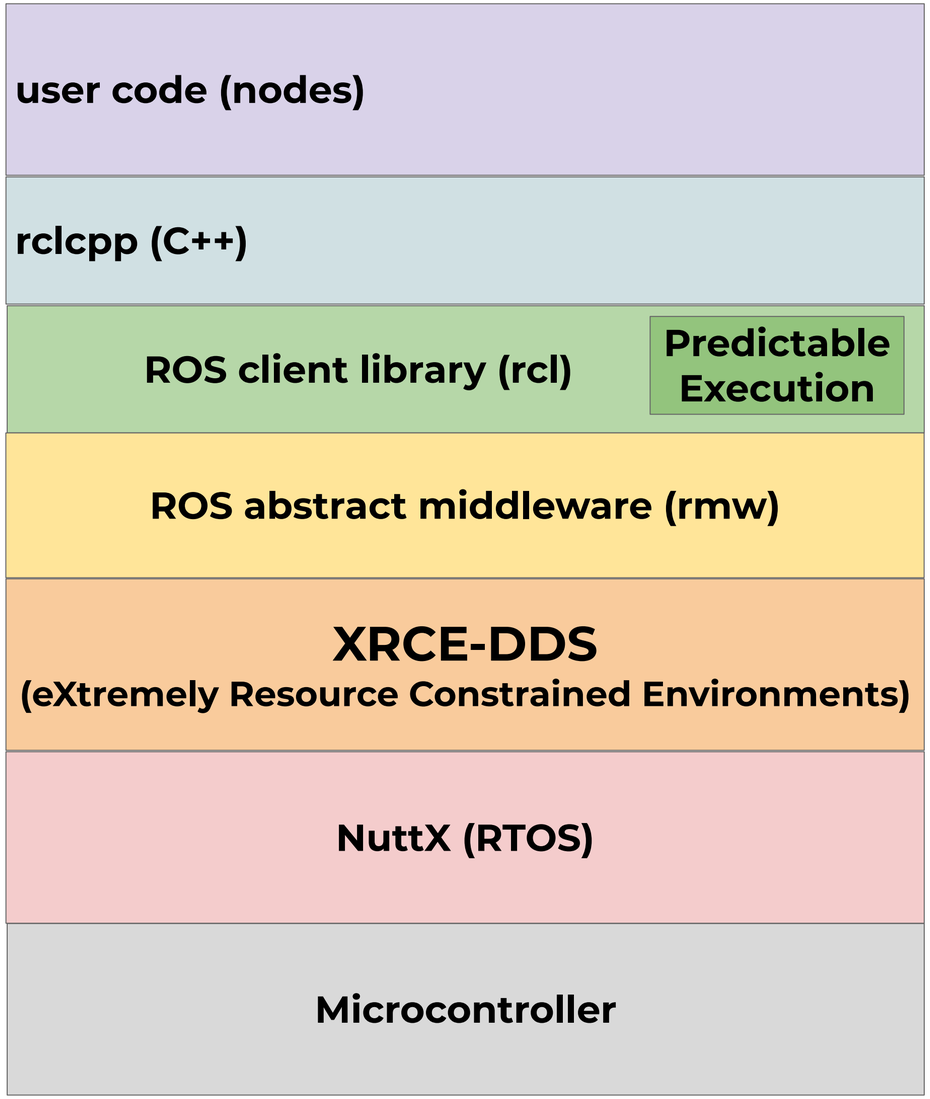
/imu

/cmd_vel

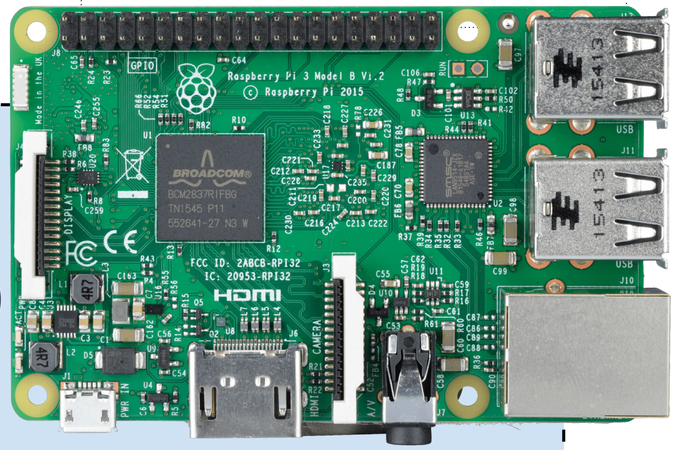
ROS2



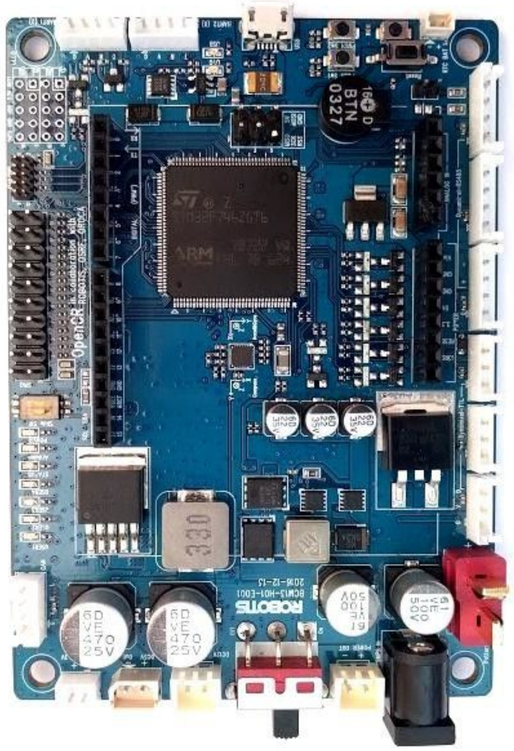
Micro-ROS



RaspBerry Pi



32 bits Microcontroller



rosserial protocol

battery state
IMU values

velocity and servo commands

ROS

rosserial

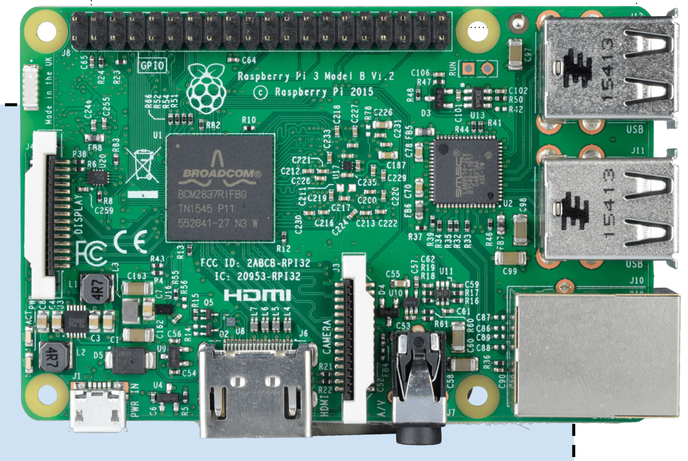
publish

/battery_state

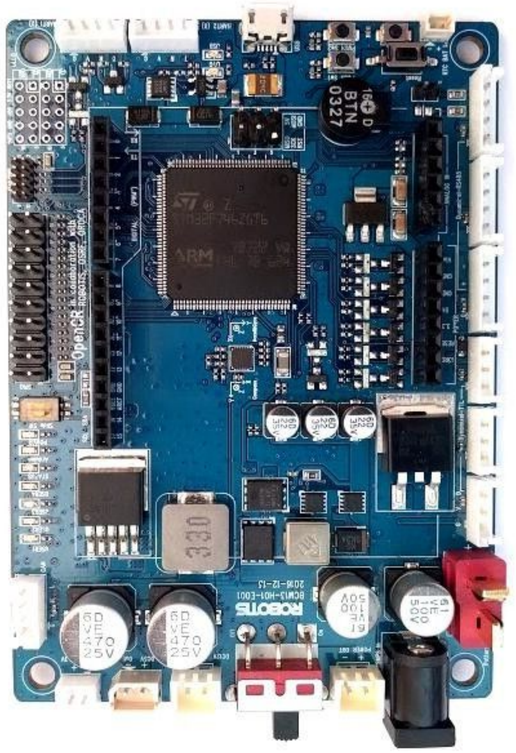
/imu

/cmd_vel

RaspBerry Pi



32 bits
Microcontroller



2

turtlebot3_core

publish

/battery_state

/imu

/cmd_vel

More and more features ...

ROS 2: other features

Real time capabilities

- ROS2 design facilitates to implement real-time compliant
- Not only Micro-ROS, also other approaches
- Not only NuttX but FreeRTOS, VxWorks or QNX

Security

- ROS1 has no security by design feature not a bug ;)
- ROS2 integrates security from DDS
- Features are: authentication, access control, cryptographic support, ect.

More: lifecycle for nodes, run multiples nodes in a single process, deterministic starting sequence, OpenEmbedded/Yocto support, etc.

Thanks! Questions?