Micro-ROS

Industrial ROS Conference
12/Dec/2017

Jaime Martin Losa
CTO eProsima
JaimeMartin@eProsima.com
+34 607 91 37 45

www.eProsima.com
eProsima in one shot

- Experts on middleware, focused on DDS.
  - DDS is the middleware powering **ROS2**.
  - eProsima Fast RTPS is one of the supported implementations
    - Currently the default implementation.

- OMG Members – Standard Contributors:
  - RPC over DDS, Web Enabled DDS,
  - DDS Security (Supporter), OPC-UA DDS (Supporter)
  - XRCE DDS (DDS for eXtremely Resource Constrained Environments)
Some Customers
Robotic Success Cases

- OSRF (ROS2)
- Dronecode (PX4)

- Many Robotics companies:
  - 3D Robotics
  - Intelligent Robots
  - Near Earth Autonomy
  - ...

Open Source Robotics Foundation

Dronecode

3DR
FIWARE Chapter Leader: Robotics and middleware

- **FIWARE:**
  - Big EU Project
  - Open Source component framework
  - More than 300M Eur of investment

- **Role of eProsima:**
  - Robotics and middleware chapter leader
  - Fast RTPS
  - Robotics Success Histories.
ePROSIMA
The Middleware Experts

Micro-ROS

Upcoming ROS2 Feature
Micro-ROS motivation

- Robots today are networks of microcontroller based sensors and actuators connected to general purpose computers.

- ROS2/DDS does not fit well in small embedded devices (microcontrollers):
  - Low computational power and memory
  - May be also:
    - Battery powered
    - Communicated through Constrained networks (LoWPANs)

- There are some “micro/tiny” DDS implementations but they cut the API and the protocol:
  - No standard yet, XRCE DDS is coming.
  - Lack of features.
Micro-ROS is a common Project from:
- eProsima (Coordinator)
- Bosch
- Acutronics (Erle, HROS)
- FIWARE Foundation
- PIAP
  - Industrial Research Institute for Automation and Measurements (Poland)

It will extend ROS2 to allow its use in microcontrollers.

Funded by EU,
- Letter of support: OSRF, ROS-I, FIWARE, Mobile Industrial Robots, Deepfield Robotics and Bosch powertools

Starting Jan, 2018
Micro-RTPS & Micro-ROS

- Micro-ROS will use micro-RTPS as the core middleware.
  - Micro-RTPS is an implementation of XRCE DDS (DDS for eXtreme Resource Constrained Environments), an upcoming standard of the OMG.
    - RTI, Twin Oaks and eProsima have already a joint submission.
    - It adapts RTPS/DDS to these scenarios.
  - Micro-RTPS first prototype already adopted by Dronecode
  - **Already an Alpha available! 😊**
Micro-RTPS links

- Github Repo: https://github.com/eProsima/micro-RTPS
- Readthedocs: http://micro-rtps.readthedocs.io
- Quick start video: https://youtu.be/XT-Y1CfOGJM
Micro-RTPS Architecture

- micro-RTPS Client
- micro-RTPS Agent
- DDS
- DDS/RTPS Participant

- RTPS
- Request/response Extension
- Agent Exposing DDS Entities
- Broker/Proxy Role
Micro-ROS Architecture
Hardware and Firmware

- Micro-ROS will select a reference platform to test and run the framework:
  - OS: NuttX
  - Hardware:
    - STM32L1: Low profile microcontroller
    - STM32F4: Medium performance microcontrollers
    - ARM Cortex-R: For certified apps.
  - Ethernet, Serial, and 6LowPan

- Creation of a reference Hardware Bridge between micro-ROS and ROS2
Comms and interoperability

- Enhanced micro-RTPS
  - Including Peer to Peer comms for selected cases.
  - Security integrated with DDS/ROS2 security.

- Interoperability
  - ROS1, ROS2, HROS, FIWARE Components
Enhanced API

- **Predictable scheduling and executing**
  - Based on the concept of executors introduced in ROS 2, we will introduce a core library for predictable scheduling and executing, allowing fine-grained control of the mapping of callbacks to tasks of the RTOS.

- **Lifecycle and system modes**
  - Starting from the node lifecycle concept proposed for ROS 2, we will design and implement advanced mechanisms for the runtime management of single application components, subsystems of multiple components and the overall system.

- **Embedded transform (TF) library**
  - A highly efficient transform library shall be developed, that integrates microROS components into the ROS TF tree in an efficient way.
Full Benchmarking and Test bed

- Multiple measures:
  - Latency, throughput, CPU use, memory...
- Reference platform benchmarking:
  - Hardware + OS
- Micro-ROS Framework benchmarking
- User benchmarking tools
Use Cases

1. - Drones (eProsimia):
   - PX4 Autopilot Example

2. - Micro-ROS sensors (Acutronics)
   - H-ROS

3. - Domestic outdoor robots (Bosch)
   - Indego

4. - Smart warehouse (PIAP)
   - PIAP SCOUT robot

FIWARE Integration of 1-4
   - Orion Context Broker
Key points:

- OMG standardization
- Truly Open Source
- ROS & ROS-I Community collaboration
- EU Projects Collaboration:
  - ROSIN, RobMoSys
  - FIWARE
Micro-ROS Building Blocks

- micro-ROS client library (ursi)
- predictable scheduling and execution (real-time executor)
- lifecycle and system modes
- embedded transform (TF)
- library

Application software

API and core libraries

Communication and interoperability

Hardware and firmware

- micro-ROS middleware interface (urmw)
- micro-RTPS

RTOS abstractions

RTOS (NutX) additional device drivers scheduler extensions

Microcontroller platform
Want to know more?

- [www.eProsima.com](http://www.eProsima.com)
- Youtube: [https://www.youtube.com/user/eProsima](https://www.youtube.com/user/eProsima)
- Mail: [JaimeMartin@eProsima.com](mailto:JaimeMartin@eProsima.com)
- Phone: +34 607913745
- Twitter: [@jaimemartinlosa](https://twitter.com/jaimemartinlosa)
- [http://es.slideshare.net/JaimeMartin-eProsima](http://es.slideshare.net/JaimeMartin-eProsima)
Thank you!

Jaime Martin Losa
CTO eProsima
JaimeMartin@eProsima.com
+34 607 91 37 45

www.eProsima.com