ROS 2 Embedded WG meeting #4 Agenda July, 29th 2019

29 July 2019 18:00 PM, Hangouts Meet.

ATTENDEES

eProsima: Jaime Martin Losa, Julian Bermudez Ortega, Borja Outerelo Gamarra.

Bosch: Ingo Lütkebohle, Ralph Lange

Nobleo: Ishu Goel, Martin Cornelis

ifm: Sean Kelly

Amazon: Nick Burek

eSOL: Akihiko Tsukuda, Shoji Morita

Renesas: Max Matsushima, Yuuki Okamiya

robodev: Andreas Bihlmaier

AGENDA

1. **Preliminary results of the poll**
2. **SIG Scope: Software, Hardware, etc…**
3. **Organisation of the SIG: Schedule of meetings, organizers, …**
4. **Performance analysis status update**
5. **Micro-ROS Update: Dashing Support, hw platform update**

Minutes

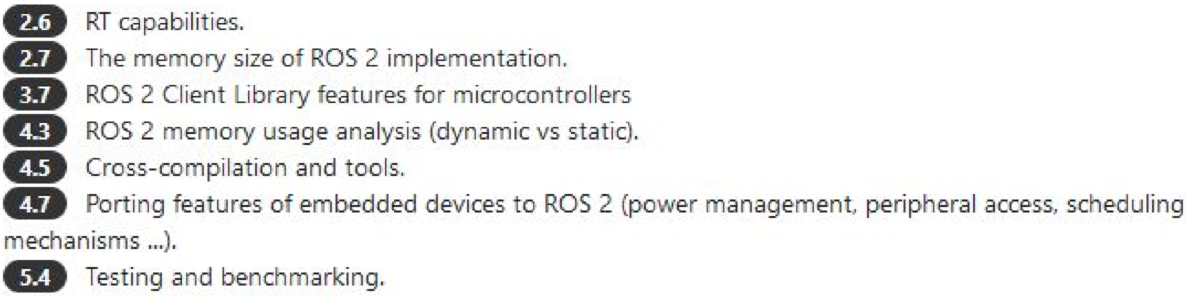
Preliminary results of the poll

We will analyze the poll in more detail soon. During the meeting, we only discussed some obvious distributional things. Also, the poll is still open, so the results may change.

- OSs:
    - Linux (Ubuntu, Yocto, Debian)
    - Linux with Xenomai/RTAI

    - RTOS (NuttX, ChibiOS, eMCOS, FreeRTOS, VxWorks),
- Hardware: Wide variety, ARM, Intel, MCUs. Several custom boards in there. Also GPUs (Tegra). No mention of FPGA?
- IOs: CAN, UART, I2C,

Interests (higher numbers mean more interest)



2.6 RT capabilities.
2.7 The memory size of ROS 2 implementation.
3.7 ROS 2 Client Library features for microcontrollers
4.3 ROS 2 memory usage analysis (dynamic vs static).
4.5 Cross-compilation and tools.
4.7 Porting features of embedded devices to ROS 2 (power management, peripheral access, scheduling mechanisms ...).
5.4 Testing and benchmarking.

SIG Scope: Software, Hardware, etc…

Polled by asking people to add "+" to the option they support. Just one +, please ;-)

1. In favor of having both Embedded Linux and MCUs in the same WG: +++++++++
2. Opposed to: +++

Nick: One worry is that expanding scope would be too much.
Ingo: Currently large overlap in work, may change in future?

Topics of Interest for voting

1. Resource use & benchmarking: +++++
2. Real-Time: +++++++
3. Security:+++
4. Tools (Cross-compilation, build systems, etc.):++++
5. Embedded-Specific Interfaces (Power, etc.):+++
6. Specific Executors: ++++++
7. Safety: +++

This seems to be a fairly even spread, with some particular interest resource use, real-time, tools and execution. The topic of real-time has some potential overlap with the Real-Time WG.

Organisation of the SIG: Schedule of meetings, organizers, …

Call ourselves a WG? Yes.

Akihiko can help organizing meetings if local meetings are needed in Asia/Pacific.

Meeting schedule

- every 2 weeks:
- every 4 weeks: +++++++++++

- every 6 weeks:
- every 8 weeks
-

Every 4th Monday of the month, alternating between 6pm European Time and 12pm.
We'll have a google calendar.

Google Calendar of the Manipulation WG is at https://calendar.google.com/calendar/embed?src=acutronicrobotics.com_8kj1403jcghr73vvt4635motdo%40group.calendar.google.com&ctz=Europe%2FMadrid

Performance analysis status update

Nobleo presented their work from https://discourse.ros.org/t/singlethreadedexecutor-creates-a-high-cpu-overhead-in-ros-2/10077 and https://answers.ros.org/question/327477/ros2-uses-6-times-more-cpu-than-fastrtps/
See also https://github.com/scgroot/ros2_performance

To summarize: The executor CPU usage is about ⅔ of the whole run time, with a large chunk in the collect_entities method. The management of the subscribers, publishers and timers with converting from weak to shared ptrs is causing significant overhead.

Ingo (Bosch) added that their intern Christoph Bedard is also currently working on related measurements and could replicate the results using a different measurement approach. He will post more concrete results after some cleanup and additional validation, and Bosch will also make available the measurement tools.

Discussion
There was some discussion on what needed to be done to address this. Generally, only very few people felt comfortable doing work on the executor, but were willing to help with testing and benchmarking. Bosch (who has people working on executors for micro-ros already) will look into whether they can lead a more general effort here.

Micro-ROS Update: Dashing Support, HW Platform Update

eProsima reported on their porting work regarding the middleware layer to dashing

- We have the core ported, but working in the examples and the repos.
- Preliminary information on the Kobuki demo: https://micro-ros.github.io/docs/tutorials/demos/kobuki_demo/

Bosch reported that their executor and mode work will become available a bit after eProsima finalizes their work.

A question came up about the future of the project because of the announcement by Acutronic Link Robotics (ALR) that they are liquidating their company effective July 31st 2019. ALR has been a partner in the EU-funded project OFERA that supported the development of Micro-ROS. Therein, they were primarily responsible for the Hardware and RTOS work (reference platforms and support for NuttX).

Nobody from ALR could be present, but other OFERA partners discussed how to continue the work. The most important foundational work has been done, and the board support for the reference platforms is available, so the immediate work is not affected. However, the project is at just half the planned time (18 of 36 months), and is obviously not yet finished. There are several options to be discussed internally and the result will be announced in due time.