QUESTIONS

- Ali AlSaibie - Kuwait University
  The idea of the executor will replace the idea of multithreading inside a MCU? I'm interested in running a lot of threads inside a MCU and then using the ros messages I get from a pc and relate them to the other threads?

  Ralph [BOSCH]:
  Currently the executor is single-threaded. In rclcpp you have multithreaded executors.
  You should define multiple executors and bind them to specific threads, that in most RTOS would be called tasks
  So that on the one hand you have the scheduling from the RTOS between those tasks and thus executurors which allows to define different periods or rates or priorities and within one executors .. serve different computables which are in ros: the callbacks.
  we have on the plan that one executor can serve multiple tasks or threads (depends on RTOS). everything is highly dependent on rtos and MCU, and because of this we decided to keep this minimal and generic and not specific concepts underlying the platforms.
  it's then up to the developer to decide how to distribute the functionalities between threads/tasks.

- [from chat] Ariel Podlubne: are you aware if anyone has tried to port it to Zynq boards? (ARM Cortex-A53)? It already supports FreeRTOS

  Pablo [eProsima]:
  We have not tested it, but if you're able to port freertos + posix extensions of frrertos which are available in github, it shouldn't be difficult to integrate micro-ros environment in the board.
  open an issue and we can speak of this port.
  we're willing to create a tutorial to ease the port of micro-ros to new boards.

- [from comments to the agenda] Peter Polidoro interested in porting micro-ROS to Teensy 4.0 board https://www.pjrc.com/store/teensy40.html, which uses an ARM Cortex-M7 NXP iMXRT1062 microcontroller.

  Pablo [eProsima]:
  You should decide which RTOS because right now we don't support bare metal. we think that this board is interesting and highly capable.

- [from comments to the agenda] Peter Polidoro interested in using various Arduino libraries with it written in C++ and Miro Samek's real-time framework.

  Pablo [eProsima]:

Regarding integrating micro-ROS into arduino we have been asked about it from many users. In my opinion porting micro-ros to arduino is ambiguous because there are many boards supported, and not all are able to run micro-ros because of their resources. we can talk and help someone to implement an extension to our build systems to produce arduino libraries or to integrate arduino in this build system, but for the moment is not something on the table.

Ralph [BOSCH]:
Arduino comes with its own build tooling, which is highly convenient. We thought on how to ease the development for the typical robotics developer coming from ros2 and that led to this build tools which hides many aspects of the underlying RTOS from the developer but also comes with limitations in terms of how deep you can configure your system.

- [from chat] Ali AlSaibie: I would like to help on FreeRTOS porting guide. Also, there should be a tutorial on creating a new ros msg in the context of Micro-ROS project structure. related issue:
  https://github.com/micro-ROS/micro-ROS.github.io/issues/177

  Pablo [eProsima]:
  I'd like to work with you on a freertos tutorial because you've done this work. We'll contact you to provide a preliminary version.
  ros msg: I can explain this in a tutorial on the web page.

- [from chat] Rp Cme: in Slack, I asked the background on needing support on MPC57xx. related issue:
  https://github.com/micro-ROS/micro-ROS.github.io/issues/186

- Ali AlSaibie: Concepts of intraprocess communication: I've worked on PX4 that has the uORB structure for interprocess communication, and I know micro-ros is working to translate these uORB messages into ros messages, but that works within the context of nuttx, but would there be something more generic that works can be applied the structure of micro-ros, for someone that wants to create multiple blocks of freertos/zephyr threads? What is the method that would be suggested to do that? For taking a thread running into a ros node, receiving and sending the messages to a pc, and then taking the message and relating internally that to another thread?

  Ralph [BOSCH]:
  This is an open issue that should be solved in micro-ros. we want to make this transparent: that we want to have multiple nodes and want them to communicate and if they are on the same MCU the communication shouldn't go via the agent (which currently happens). It highly depends on Micro XRCE-DDS. Or should that be implemented in the higher layers rather than in the actual middleware?

  Pablo [eProsima]:

right now Micro XRCE-DDS is not ready to do interprocess communication. In an RTOS you have tasks running a micro-ros node and communicating with other tasks using rtos features such as cues and so on.
example: in crazyflie node, they do a lot of interprocess communication among different tasks but they have their own way for communicating e. g. the inertial unit sensor with the micro-ros task, so the drone can publish the sensor data that is being collected in another process.
however, for the moment you can't do that in micro-ros alone.
I'm sure that it is possible to run the micro-ros middleware in a multi-thread environment, but it's not trivial.

- Kashish Dhal: which board + RTOS will be covered in Foxy Tutorials?

  Pablo [eProsima]:
  I reccomend to use zephyr port for micro-ros and the ST discovery board, is the easiest way to have a micro-ros node running in a board.

Suggestions from users

- Create a porting guide for FreeRTOS boards (talk with Ali AlSaibie @alsaibie). Related issue:
  - https://github.com/micro-ROS/micro-ROS.github.io/issues/177
- Create a tutorial of how to create a custom ROS 2 msg in micro-ROS environment.
- Communication over real-time bus, CAN or CAN FD?