yet another runtime environment onto embedded devices

Hideki Takase, Shintaro Hosoai, Yoichiro Hibara, Haruaki Tanaka (The University of Tokyo), Hidetoshi Yugen (Kyoto University), Shoji Morita (eSOL Co.,Ltd.)

A part of this work is supported by JST CREST JPMJCR21D2 and the commissioned research (04001) by National Institute of Information and Communications Technology (NICT), Japan.
Cutting-Edge Platform and Design Methodology for embedded/IoT Computing

Toward a world where anyone can easily create awesome products

Optimization
Methodology

Mission

Recent R&D
2 meets EMB for IoT!!

Platform for accelerating the development of robot systems

• The essence of ROS is **Plumbing**
  - Loosely coupled arch. of ROS nodes
  - Easy to register, delete, and restore them
  - Basis is pub/sub comm. via **Topic**

[Platform](http://www.ros.org/about-ros/)

Integration of Embedded Technology

- **comm. performance**
  - responsiveness (latency)
  - real-time performance (variation)
- **Power consumption**

ROS 2/DDS communication tech. is also expected to be deployed in the IoT field

---

Platform for accelerating the development of robot systems

• The essence of ROS is **Plumbing**
  - Loosely coupled arch. of ROS nodes
  - Easy to register, delete, and restore them
  - Basis is pub/sub comm. via **Topic**

[Platform](http://www.ros.org/about-ros/)

Integration of Embedded Technology

- **comm. performance**
  - responsiveness (latency)
  - real-time performance (variation)
- **Power consumption**

ROS 2/DDS communication tech. is also expected to be deployed in the IoT field
How to “embed”??

How to “embed”??

**native ROS 2 on SBC**

**mROS 2 on EMB board**

better perf. & mem. usage partially compatible with rclcpp only for Topic comm., and many unsupported features such as QoS, Service, , , ,

**micro-ROS on EMB board**

various kernels & boards serial, UDP, Bluetooth trans. fully compatible with rclc XRCE-DDS (with agent)

mROS-base/mros2
Dive into comm.  

• **RTPS**: comm. protocol of DDS  
  – SPDP/SEDP: autonomously searches/establishes communication partners/paths (w/o master)  
  – GPOSeS are essential!! (ROS 2 itself)  
• **micro-ROS**: de-fact of EMB platform  
  – employ Micro-XRCE-DDS (default)  
  > agent is necessary as the master

RTPS communication directly from EMB without an agent!!

※GPC: General Purpose Computer  
  EMB: Embedded Micro-controller Board
Software Stack

- mROS 2 application
- mROS 2 API
- mROS 2 comm. lib.
  - RTPS (embeddedRTPS)
  - UDP stack (lwIP)
  - CMSIS wrapper
- HAL library
- Embedded devices
- pub/sub messaging for Topic partially compliant with rclcpp
- autonomous communication in accordance with RTPS specification
- lightweight and efficient process by C/C++ for <~200MHz / <~1MB
- real-time kernels for EMB contributes perf. and mem. usage

mROS-base/mros2
Note: embeddedRTPS [A. Kampmann+ ITSC’2019]

- Portable RTPS implementation by C++
  - lwIP (Raw Mode) for UDP/IP
  - Micro-CDR for serialization
  - some code dependencies with FreeRTOS

- Main features and contributions
  - Discovery: SPDP & SEDP features
  - Interoperability: tested with FastDDS 2.3.1
  - QoS Policies: support best-effort & reliable
  - UDP Multicast: support multicast locators
  - Message size is limited to buffer size of lwIP

mROS-base/mros2

https://github.com/embedded-software-laboratory/embeddedRTPS
Getting started!!

- Steps for embedded board
  - $ git clone https://github.com/mROS-base/mros2-mbed
  - $ cd mros2-mbed
  - $ ./build.bash all NUCLEO_F767ZI echoback_string
  - $ picocom -b 115200 /dev/ttyACM0

- Step for the host
  - $ docker run --rm -it --net=host ros:humble /bin/bash
  - $ "mkdir -p ~/ros2/src & & cd ~/ros2/src
  - git clone https://github.com/mROS-base/mros2-host-examples & &
  - cd mros2-host-examples & &
  - colcon build --packages-select mros2_echoreply_string & &
  - cd .. / & & source install/setup.bash & &
  - ros2 run mros2_echoreply_string echoreply_node"
Currently Supported

**mROS-base/mros2-as-p3-f767zi**
reference implementation of mROS 2 for STM32
NUCLEO-F767ZI with TOPPERS/ASP3 kernel

- ⭐️ 14
- 🌟 2

**mROS-base/mros2-mbed**
reference implementation of mROS 2 for Mbed OS

- ⭐️ 29
- 🌟 4

**mROS-base/mros2-posix**
reference implementation of mROS 2 for POSIX layer

- ⭐️ 0
- 🌟 1
Evaluation mros2@v0.3.1 with Foxy

Round-Trip Time for UInt16, Twist, String by rclcpp::WallTimer.get_clock() on the host

<table>
<thead>
<tr>
<th>API</th>
<th>uros-serial</th>
<th>uros-udp</th>
<th>uros-rtps</th>
<th>mros2-asp3</th>
<th>mros2-mbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTPS</td>
<td>Micro XRCE DDS</td>
<td>USART</td>
<td>UDP</td>
<td>RTPS on UDP</td>
<td>TOPPERS/ASP3</td>
</tr>
<tr>
<td>protocol</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTOS</td>
<td>FreeRTOS v2</td>
<td>8.3.1</td>
<td>9.3.1</td>
<td>7.3.1</td>
<td>10.3.1</td>
</tr>
<tr>
<td>compiler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://github.com/mROS-base/eval/releases/tag/v0.1.1
# Evaluation Results

## • UInt16 [ms]

<table>
<thead>
<tr>
<th></th>
<th>uros-serial</th>
<th>uros-udp</th>
<th>uros-rtps</th>
<th>mros2-asp3</th>
<th>mros2-mbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>11.710</td>
<td>2.109</td>
<td>5.182</td>
<td>0.570</td>
<td>0.646</td>
</tr>
<tr>
<td>max</td>
<td>17.370</td>
<td>4.240</td>
<td>11.190</td>
<td>0.810</td>
<td>0.940</td>
</tr>
<tr>
<td>min</td>
<td>7.590</td>
<td>1.900</td>
<td>1.940</td>
<td>0.490</td>
<td>0.560</td>
</tr>
<tr>
<td>std.p</td>
<td>3.094</td>
<td>0.244</td>
<td>2.684</td>
<td>0.067</td>
<td>0.081</td>
</tr>
</tbody>
</table>

## • Twist [ms]

<table>
<thead>
<tr>
<th></th>
<th>uros-serial</th>
<th>uros-udp</th>
<th>uros-rtps</th>
<th>mros2-asp3</th>
<th>mros2-mbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>19.530</td>
<td>2.304</td>
<td>5.508</td>
<td>0.593</td>
<td>0.703</td>
</tr>
<tr>
<td>max</td>
<td>25.510</td>
<td>11.250</td>
<td>9.860</td>
<td>0.850</td>
<td>0.880</td>
</tr>
<tr>
<td>min</td>
<td>15.590</td>
<td>2.050</td>
<td>2.610</td>
<td>0.520</td>
<td>0.640</td>
</tr>
<tr>
<td>std.p</td>
<td>3.666</td>
<td>0.904</td>
<td>1.551</td>
<td>0.065</td>
<td>0.042</td>
</tr>
</tbody>
</table>

## • String

### mem size (binary for Twist)

<table>
<thead>
<tr>
<th></th>
<th>uros-serial</th>
<th>uros-udp</th>
<th>uros-rtps</th>
<th>mros2-asp3</th>
<th>mros2-mbed</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>209,836</td>
<td>233,656</td>
<td>174,752</td>
<td>90,551</td>
<td>393,312</td>
</tr>
<tr>
<td>data</td>
<td>356</td>
<td>356</td>
<td>576</td>
<td>16,632</td>
<td>3,336</td>
</tr>
<tr>
<td>bss</td>
<td>110,280</td>
<td>108,160</td>
<td>282,016</td>
<td>111,800</td>
<td>70,688</td>
</tr>
<tr>
<td>total</td>
<td><strong>320,472</strong></td>
<td><strong>342,172</strong></td>
<td><strong>457,334</strong></td>
<td><strong>225,983</strong></td>
<td><strong>469,336</strong></td>
</tr>
</tbody>
</table>
論よりRUN!!

"ron yori run"
The RUN is mightier than the word

$ ros2 run mturtlesim turtlesim_node

$ picocom /dev/ttyACM0

publish Twist

Emb board + analog joystick
論よりRUN!! Part III powered by mROS

- robot body node
- wheeled inverted pendulum
- calc. PWM node
- IMU data
- to_control node
- to_pid node
Conclusion

• Agent-less and lightweight runtime environment for ROS 2
• Our Contribution
  – mROS 2 enables programs running on embedded devices to communicate autonomously with nodes on the native ROS 2
  – mROS 2 would contribute to the construction of distributed robot systems with excellent communication performance

If you wanna communicate **only with Topic**, please consider to try our mROS 2 as one of the candidates🤔🤔🤔

mROS-base/mros2
Check now!! & What’s next??

mROS-base/mros2
agent-less and lightweight communication library compatible with rclcpp for embedded d ...

⭐ 104  🐱 7

https://github.com/mROS-base/mros2

Please give us the Star! 😍 & your contribution!! 😊

- porting to other boards and kernels
- implement new targets with POSIX-compliant RTOS
- support QoS control, Service, Action, ...
- design a dedicated board for real robots??

WE WANT YOU IN OUR TEAM!

https://github.com/mROS-base/mros2