D2.7

Report on the micro-ROS hardware bridges

Grant agreement no. 780785
Project acronym OFERA (micro-ROS)
Project full title Open Framework for Embedded Robot Applications
Deliverable number D2.7
Deliverable name Report on the micro-ROS hardware bridges
Date December 2019
Dissemination level public
Workpackage and task 2.3
Author Juan Flores Muñoz (eProsima)
Contributors Borja Outerelo Gamarra (eProsima)
Keywords Hardware-Bridge, Raspberry-Pi, requirements, ROS, ROS 2
Abstract This document provides a report on the micro-ROS to ROS 2 hardware bridge. Corresponding with an embedded Linux board with proper native ROS 2.0 support and with required communications interfaces, such as Ethernet, Serial or 6LoW-PAN.

Contents

1 Summary

The hardware bridge is a physical device that allows communication between micro-ROS and ROS 2 devices. This device must include all the required communications protocols and been able to run micro-ROS full stack with the required dependencies. On the next document, we will study the required characteristics for this hardware device, we will compare the different devices available, and we will finally propose one of the studied devices.

2 Acronyms and keywords

Acronym	Definition
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
CPU	Central Processing Unit
LoRaWAN	Low Power Wide Area Network
MCU	Microcontroller Unit
RAM	Random Access Memory
ROS	Robot Operating System
RPI	Raspberry Pi
SoC	System On Chip
YAML	Serialize data format for humans

3 Introduction

The Micro-ROS hardware bridge is a physical device that allows the interoperability between micro-ROS devices and ROS 2. This device runs micro-ROS Agent as the main piece of software, which works as a bridge between ROS 2 and micro-ROS applications. micro-ROS Agent enabled hardware opens the door to use microcontroller units (MCU) on the ROS 2 world, with the advantages of this kind of devices: fast boot, power consumption and price tag.

To select a device, we will perform an analysis of the available options and which can meet our requirements. At this moment, due to the high popularity of the embedded PCs, there is a wide variety of embedded boards that could accomplish our required points. Still, we will focus on the most popular because from one side; they have better support from the community and from the other side, it would be easier to find a supplier.

Once we've analyzed the possible devices, we will propose one board that will be the default one.

4 Requirements

We have four main blocks of requirements: - Ability to run micro-ROS Agent. - Support the required communications. - Low power consumption. - Affordable and easy to buy.

Due to each point has a sublist of requirements, we're going to study each point in detail, to finally get a better idea of what do we need.

4.1 Run micro-ROS Agent

The selected device must run a micro-ROS Agent as the main point. This piece of software has a set of requirements. The hardware bridge should accomplish:

- Able to execute in Linux based device.
- UDP communications support.
- Serial Communications support.
- ROS 2 support.
- RAM: 8.1 MiB
- Flash: 75 MiB (Memory data measured using ShapesDemo publish/subscribe sample program)

If we deep on the requirements of each point, we see that as a basic start, we need support for the next points, to finalize the building process of the software:

- Full C++ 14 support.
- Python3 support.
- YAML
- CMake & MAKE

Theses requirements should be accomplished by most of Linux distributions available for embedded systems, so we could avoid the last three points considerating that they are included on Linux.

4.2 Support required communications

The project **require** at least the next communication support: - Ethernet connectivity. - Serial communication. - 6LoWPAN.

And also is **desirable but not mandatory**, to have support for the next communications: - Bluetooth. - Wifi. - Long-range radio distance, such as LoRaWAN.

4.3 Low Power Consumption

The micro-ROS bridge should able to run on scenarios without power network access, so it must be a power constraint device to be able to run on batteries. On the other side, a regular embedded PC can consume high amounts of power in regular mode, so it must implement a power management module, to adapt to each circumstance and save as much energy as possible.

In conclusion, we need a device that must accomplish the list of requirements: - It can run Linux. - It can execute ROS 2. - Support for Ethernet connectivity. - Support for Serial connectivity. - Support for 6LOWPAN. - Almost 8.1 MiB free of RAM and 75 MiB free of Flash memory. - Power constraint to run with a battery and power management. - Affordable and easy to buy.

Having in mind these requirements, we will study the next list of proposed devices that could fit our requirements.

5 Device comparation

We have elaborate a table with a list of proposed boards and the most remarkable characteristics to have in mind. Thi table will help to decide which board could fit most of our required points.

Table 2: hardware specifications comparison.

Name	CPU	RAM	Flash	Linux	Ethernet	UART	6LoWPAN	Price	Community support
RPI 3	1.2 GHz 64b quad-core ARMv8	1 GB	Ext	Yes	Yes	Yes	Yes (dongle)	36€	High
RPI 4	1.5 GHz 64b quad-core Cortex-A72	1.2/4 GB	Ext	Yes	Yes	Yes	Yes (dongle)	66€ (4GB model)	High
RPI Zero W	1 Ghz 32b ARM 11	512 MB	Ext	Yes	No	Yes	Yes (dongle)	7€	High
Orange Pi Plus	1.6 Ghz 64b Cortex A7	1 GB	8 GB	Yes	Yes	Yes	Yes (dongle)	20€	Low
Nvidia Jetson	1.45 Ghz 64b Cortex-A57	4 GB	16 GB	Yes	Yes	Yes	N/A	109€	High

(Note: The price was obtained from Amazon, and it can fluctuate)

Another of the critical points to have in mind with the selection of the board is the power consumption, so we have elaborate the following graphic. This graphic, show the consumption on load mode(Full usage of the board resources) and idle mode(We could considerer as a standby mode):

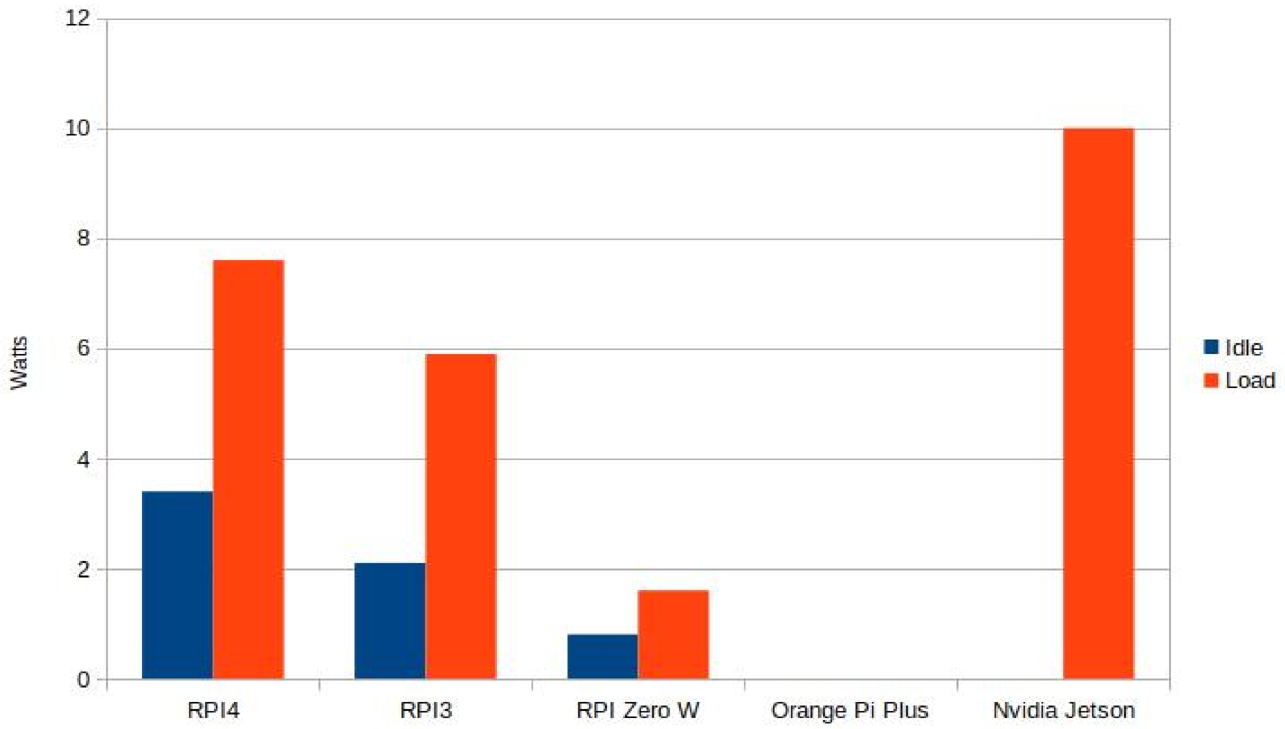

Figure 1: Power Consumption Comparation

Unfortunately, we didn't find all the required data, and we don't have all the proposed software to obtain by our self the measure, but we estimate base on the previous experience the following values:

- Orange Pi Plus Idle: ~4 Watts.
- Orange Pi Plus Load: ~7 Watts.
- Nvidia Jetson Idle: ~5 Watts.

With the previous data, we can conclude that the **Raspberry Pi3 B is the most convenient option**. The reasons to choose this board instead other is because it is the most balanced board of this study: - It consumes more power than the RPI Zero W, but it can be up to three times faster than it. RPI Zero W, based on its specifications, may not able to execute ROS 2. - The power consumption is considerably lower than the RPI4 and the NVidia Jetson. - It is the second much cheaper option. - Trendy and highly supported board, full of tutorials and community.

6 Proposed Device

Once we've chosen the **Raspberry Pi 3**, this is its full list of specifications:

- SoC: Broadcom BCM2837
- CPU: 4× ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV

- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: MicroSD
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial - - Interface (CSI), Display Serial Interface (DSI)

6.1 Supported Operating System

According to the official website, it can run the next O.S. :

Operating System	Support
NOOBS	Official
Raspbian	Official
Ubuntu Mate	Third Party
Ubuntu Core	Third Party
Ubuntu Server	Third Party
Windows 10 IOT Core	Third Party
OSTC	Third Party
LibreELEC	Third Party
Mozilla WebThings	Third Party
PiNet	Third Party
RISC OS	Third Party
Weather Station	Third Party

6.2 Test performed

We have performed the next list of tests to check if all requirements are accomplished.

- ROS 2 runs appropriately on this board. The native compilation is a lengthy process, but we supplied external cross-compilation tools and tutorials (All the information is available on the deliverable 2.5 and 2.6).
- Micro-ROS Agent works fine. As with ROS 2, the native compilation process is quite slow, so we use the same cross-compilation tools to solve this problem.
- All the communications work correctly.
- Achieve 6lowpan communications with other RPI boards and with NuttX.

In conclusion to the test, we obtain complete success on all of the performed tests achieving all the required functionalities.

7 Conclusion

After this analysis, we can conclude that the Raspberry Pi3 B is, from the set of analysed devices, the most suitable board for this task, thanks to the balance between, power, cost, and popularity. This board has massive popularity in the maker community, so this is a fascinating point because this ensures support for several years.

On the other hand, all the other boards analyzed are exciting, and probably it could run the micro-ROS Agent. Still, despite similarities between all of them, we think that it is better to focus on only one board, to avoid possible individual problems.

A final point and a point that reinforce the propose of the RPI 3 as default hardware bridge, is the conservative politics that follow the Raspberry Pi foundation on each iteration of the board. From the newest version(Raspberry Pi 4) to the first one, they didn't add any disruptive change on the board, only adding some add-ons such as Wi-FI or USB-C. So this, warranty us the compatibility of our software with a future release of this board, leveraging maintenance tasks.

8 Reference:

- ROS 2 Dependencies
- Raspberry Pi models specs
- Orange Pi Plus specs
- Nvidia Jetson Specs
- Raspberry Pi Power consumption study