Micro Ros Handleiding

Agit Tunc

13 november 2024

Inhoudsopgave

1 Inleiding

Dit document bestaat uit twee onderdelen. Het introduceren van micro-ROS en hoe dit gebruikt wordt in het huidige project. In dit document gaan we er van uit dat de lezer weet wat ROS 2 is.

2 Wat is micro-ROS?

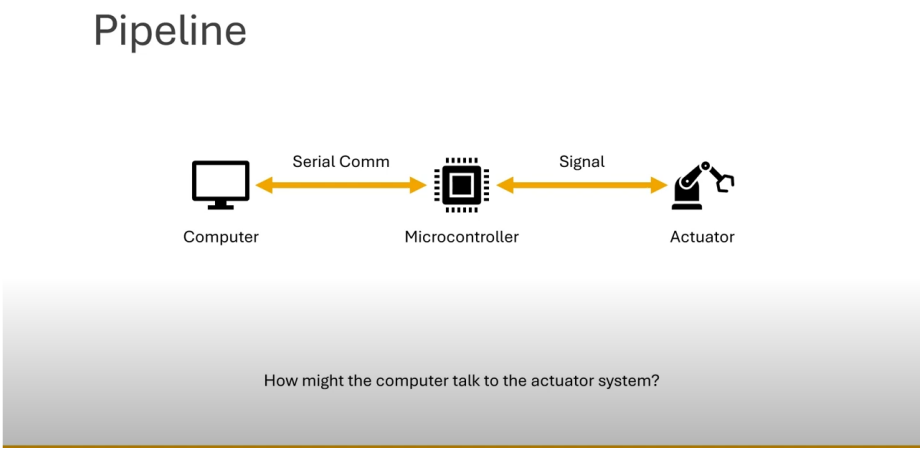
Micro-ROS is een versie van ROS2 (Robot Operating System 2) speciaal ontwikkeld voor microcontrollers. Het is bedoeld voor apparaten met beperkte rekenkracht, geheugen en energieverbruik. Micro-ROS kan dus zowel voor microcontrollers, singleboardcomputers als computers gebruikt worden.

Micro-ROS brengt de functionaliteiten van ROS2 naar embedded systemen, waardoor ze eenvoudig kunnen communiceren met grotere ROS-systemen. Het wordt veel gebruikt in IoT, robotica, en embedded systemen die moeten samenwerken met ROS-gebaseerde infrastructuren.

3 Waarom micro-ROS?

Om te begrijpen waarom micro-ROS zo nuttig is binnen een ROS-gebaseerd systeem, bekijken we een voorbeeld van een robotarmsysteem.

Stel dat we een robotarm willen ontwikkelen waarin alle "low level" aansturingen worden afgehandeld door een microcontroller, terwijl de "high level" besturing op een pc plaatsvindt. Een dergelijk systeem zou er in grote lijnen uitzien zoals weergegeven in Figuur 1.

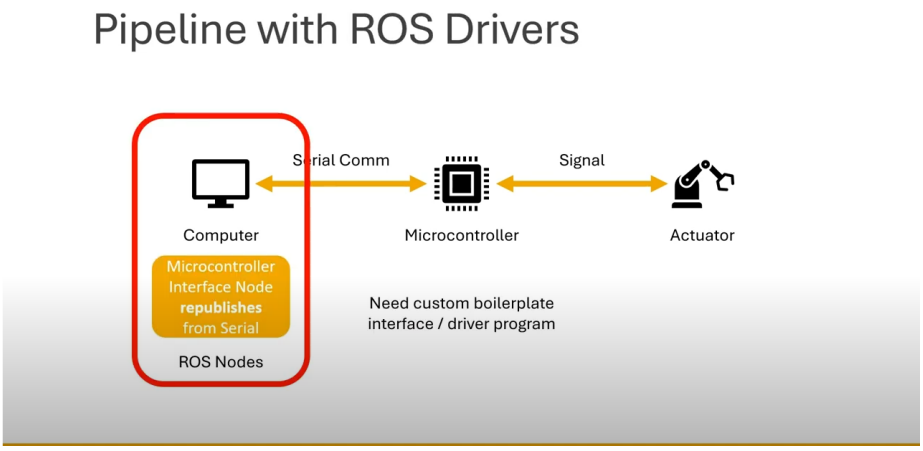


Figuur 1: Pipeline robotarm aansturing (GT Marine Robotics Group, 2023a).

3.1 Traditionele benadering zonder micro-ROS

De traditionele benadering om een robotarm vanuit ROS aan te sturen zou zijn om een "custom driver"te ontwikkelen in de vorm van een ROS-node. Deze driver maakt de communicatie mogelijk tussen de pc en de microcontroller (MCU). De ROS-node fungeert hierbij als een doorgeefluik: hij zet seriële data van de MCU om in ROS-berichten, zodat deze binnen het ROS-netwerk bruikbaar zijn, en vertaalt ROS-berichten weer terug naar seriële data voor de MCU.

Daarnaast zou je ook een eigen protocol moeten ontwerpen en implementeren om de seriële communicatie tussen de MCU-firmware en de ROS-node te beheren, zodat beide systemen goed op elkaar aansluiten.



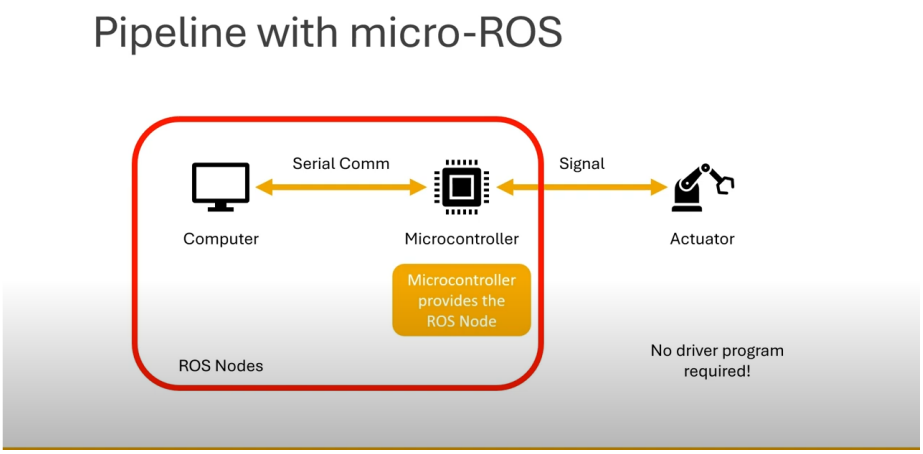
Figuur 2: Pipeline robotarm aansturing met custom driver (GT Marine Robotics Group, 2023b).

Het nadeel van deze aanpak is dat bij elke nieuwe functionaliteit die aan de robotarm wordt toegevoegd, ook het seriële communicatieprotocol en de ROS-node moeten worden aangepast. Dit leidt tot meer complexiteit en onderhoudswerk.

3.2 Voordeel van micro-ROS

Met micro-ROS wordt de communicatie tussen een MCU en pc aanzienlijk vereenvoudigd. Micro-ROS maakt het mogelijk om een ROS-node direct op een MCU te implementeren, zodat de MCU rechtstreeks deel kan uitmaken van het ROS-netwerk. Hierdoor hoef je geen eigen communicatieprotocol te ontwikkelen; micro-ROS maakt gebruik van ROS-berichten om data tussen de pc en de MCU uit te wisselen. Dit verlaagt de complexiteit aanzienlijk en maakt het

toevoegen van nieuwe functionaliteit eenvoudiger, omdat je binnen hetzelfde ROS-ecosysteem blijft werken.



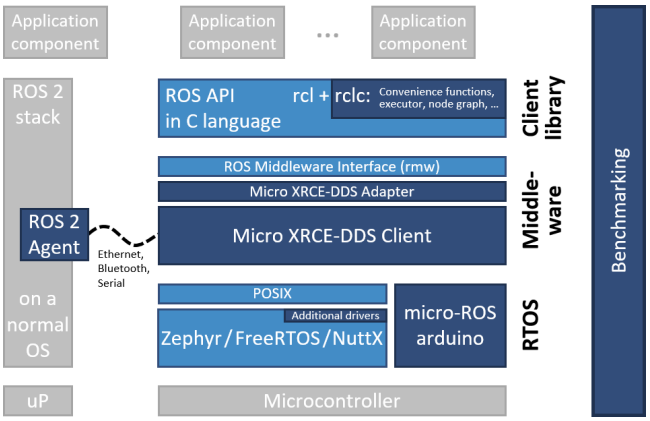
Figuur 3: Pipeline robotarm aansturing met micro-ROS (GT Marine Robotics Group, 2023c).

4 Mogelijke manieren om micro-ROS te gebruiken

Er zijn meerder wegen om micro-ROS in je project te gebruiken, de factoren die bepalen hoe je het specifiek voor je project gebruikt hangt af van de platform, RTOS en eventueel framework van je MCU. Om te achterhalen of micro-ROS geschikt is voor je MCU zou je een van de onderstaande opties kunnen uitvoeren:

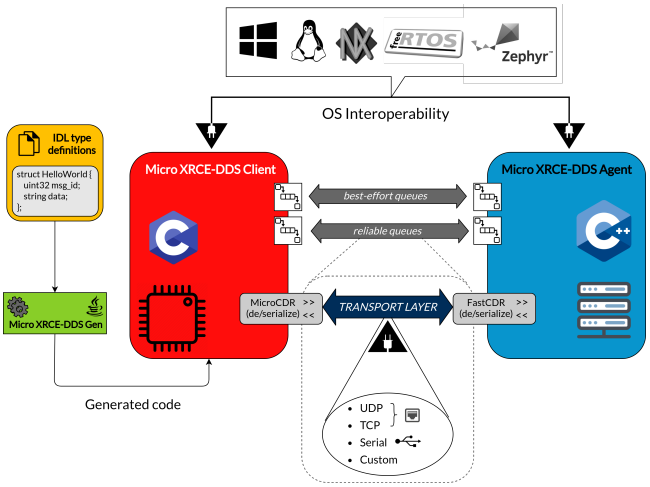
- Hardware opzoeken: Zoek je hardware op de micro-ROS website of github pagina en kom er achter welke RTOS hiermee ondersteunt wordt.
- RTOS of platform opzoeken: Zoek je gewenste RTOS of platform op de micro-ROS website of github pagina en kom er achter welke hardware hierbij ondersteund wordt.

In Figuur 4 en Figuur 5 wordt de architectuur van micro-ROS weergegeven. Hier kunnen we zien hoe de "stack" van de micro-ROS client bibliotheek eruit ziet. Hier is ook te zien dat er diverse keuzes van RTOS's en transportlagen beschikbaar zijn binnen micro-ROS. Ook is de arduino framework te gebruiken.



Figuur 4: Micro-ROS architecture (eProsima, z.d.).

Om een micro-ROS client te laten communiceren met het ROS2 netwerk, moet je host machine de micro-ROS agent draaien. De micro-ROS agent is infeite de brug tussen het ROS netwerk op je pc en de MCU.



Figuur 5: Micro-ROS client-agent (FranFin, 2021).

Het ontwikkelen en bouwen van een micro-ROS node kan op twee manieren, namelijk het gebruik van micro_ros_setup package of een micro-ROS component voor een specifiek platform.

4.1 micro_ros_setup package (micro-ROS build system)

De micro_ros_setup package biedt (RTOS) speficieke buildtools om micro-ROS te cross-compileren met ROS 2 voor de platformen te zien in Figuur 6.

```
hcl@healthbot-3-MAX-N:~/micro_ros_ws$ ros2 run micro_ros_setup create_firmware_ws.sh --help
Non valid RTOS/Platform: --help/generic
Available platforms:
. android
+-- generic
. freertos
+-- crazyflie21
+-- esp32
+-- nucleo_f446re
+-- nucleo_f446ze
+-- nucleo_f746zg
+-- nucleo_f767zi
+-- olimex-stm32-e407
. generate_lib
+-- generic
. host
+-- generic
. raspbian
+-- stretch_v8
+-- buster_v7
+-- buster_v8
. zephyr
+-- discovery_l475_iot1
+-- olimex-stm32-e407
+-- nucleo_f401re
+-- nucleo_h743zi
+-- nucleo_f746zg
```

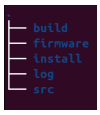
Figuur 6: Beschikbare platformen

Het icoon " - " staat voor RTOS/OS en "+–" staat voor platform. Zo zien we dat het mogelijk is om firmware voor de platform olimex-stm32-e407 te bouwen met de RTOS FreeRTOS of Zephry.

Voor een linux gebaseerd platform, zoals android, ubuntu en raspbian kunnen we de tutorial first application linux volgen voor een basis opzet.

Voor een RTOS gebaseerd toepassing kunnen we de tutorial first application rtos volgen voor een basis opzet.

Een micro-ROS workspace (i.e. firmware workspace) zal er zo uit zien:



Figuur 7: micro-ROS workspace map structuur nadat create_firmware.sh is aangeroepen.

In de `/firmware/{RTOS map}/apps/<your-app-name>` map ontwikkel je de node dat voor een MCU bestemd is. Voor FreeRTOS zou dit dan

`/firmware/freeRTOS/apps/<your-app-name>` zijn.

In de `/src/uros/micro-ROS-demos/rclc/<your-app-name>` ontwikkel je nodes voor een linux based platform.

Zoals eerder vermeld is ontwikkelen op deze manier bedoeld voor ontwikkelaars die beginners zijn met microcontrollers, je hoeft geen platform specifieke configuratie/instellingen uit te voeren. Echter is deze manier van ontwikkelen wel beperkt, omdat je gelimiteerd wordt met platform opties die de package aanbied.

4.2 micro-ROS component (External build system)

Micro-ROS kan je ook integreren met verschillende platform-buildtools. Het genereren van een micro-ROS component voor een specifieke platform kan met de commando `ros2 run micro_ros_setup component [COMPONENT_NAME]` uit de micro_ros_setup package. Een component kan een module of (pre)compiled biniries zijn, dit is afhankelijk van de gekozen platform. Een ander alternatief om een component te verkrijgen is om het binnen te halen via github.

In het onderstaande tabel wordt er een overzicht gegeven voor welke platformen je een micro-ROS component kan aanmaken. Daarnaast is er ook een github link voor het dat specifieke component, waar je met met behulp van git het component ook kan binnen halen, maar ook documentatie kan vinden hoe je het component moet gebruiken binnen de gekozen platform.

Platform	**Component name**	**GitHub link**
ESP-IDF	esp_idf	micro_ros_espidf_component
Zephyr RTOS	zephyr_rtos	micro_ros_zephyr_module
Mbed RTOS	mbed_rtos	micro_ros_mbed
NuttX RTOS	nuttx_rtos	micro_ros_nuttx_app
Microsoft Azure RTOS	azure_rtos	micro_ros_azure_rtos_app
RT-Thread RTOS[1]	rtthread_rtos	micro_ros_rtthread_component
TI Tiva™ C Series	tiva_c_series	micro_ros_tivac_launchpad_app
STM32CubeMX en STM32CubeIDE	stm32cube	micro_ros_stm32cubemx_utils
PlatformIO	platformio	micro_ros_platformio
Arduino IDE	arduino	micro_ros_arduino
Raspberry Pi Pico SDK	raspberry_pi_pico	micro_ros_raspberrypi_pico_sdk

Tabel 1: Overzicht van platforms, componentargumenten en bijbehorende GitHub-links

[1] Alleen beschikbaar voor de ROS-distributies foxy en humble

4.3 RTOS of Baremetal

TODO... micro-ROS wordt aangeboden altijd met RTOS. Enige optie om barematel te programmeren is arduino?

Ik denk dat je inprincipe een workspace kan maken met een gekozen RTOS en niet hier van gebruiken maakt? → Testen.

Als je een component gebruikt maak je per definitie geen gebruik van een RTOS ? → **Uitzoeken.**

Het is nu niet duidelijk of je barematel kan programmeren zonder arduino... Concreet voorbeeld: kan ik met esp32 in esp-idf programmeren zonder gebruik te maken van RTOS? Maakt micro-ROS onderliggend dan gebruik van een RTOS?

5 Gebruik van micro-ROS voor het HCL project

TODO...

Voor dit project gebruiken we een ESP32. De drivers van de sensoren en actuoren zijn beschikbaar in de vorm van arduino libraries. Dit laat de opties arduinio IDE en platformIO over. Voor de optie platformIO is er gekozen. RTOS gebruiken we niet, omdat er niet echt een reden voor is..

6 Informatie met nog geen hoofdstuk

TODO..

Let op! Micro-ROS is bedoeld om alleen te samen werken met het ROS2 infrastructuur. Mocht er de behoefte zijn om een microcontroller te samen werken met het ROS 1 infrastructuur dan zijn de volgende opties mogelijk:

6.1 ros1_bridge

Het gebruik van ros1_bridge.

Micro-ROS client op je MCU → Micro-ROS agent op je host machine (ROS 2 heb je nodig) → ros1_bridge → ROS 1 netwerk

6.2 rosserial

TODO..

Het gebruik van rosserial. Met rosserial kan je alleen via de serieele poort communiceren. Voor elke client moet er een server zijn.

rosserial-client librarie op je MCU → rosserial-server op je host (ROS 1)

6.3 custom driver

Uiteraard zou je ook je eigen driver kunnen schrijven om data binnen de ROS infrastructuur te krijgen.

Verklarende woordenlijst, afkortingen en acroniemen

MCU Microcontroller. 3

micro-ROS Een versie van ROS2, in C, speciaal ontwikkeld voor microcontrollers. 2

ROS Robot Operating System. 2

RTOS Real Time Operating System. 4

Referenties

eProsima. (z.d.). *Features and Architecture.* micro-ROS. Geraadpleegd op 7 november 2024, van https://micro.ros.org/docs/overview/features/.

FranFin. (2021, maart). *Micro XRCE-DDS Agent* [repository]. Github. Geraadpleegd op 7 november 2024, van https://github.com/eProsima/Micro-XRCE-DDS-Agent/blob/master/docs/General.png.

GT Marine Robotics Group. (2023a, februari). *Intro to micro-ROS* [Video]. Youtube. Geraadpleegd op 7 november 2024, van https://www.youtube.com/watch?v=aD3Lf-9cb0A&t=23s.

GT Marine Robotics Group. (2023b, februari). *Intro to micro-ROS* [Video]. Youtube. Geraadpleegd op 7 november 2024, van https://www.youtube.com/watch?v=aD3Lf-9cb0A&t=50s.

GT Marine Robotics Group. (2023c, februari). *Intro to micro-ROS* [Video]. Youtube. Geraadpleegd op 7 november 2024, van https://www.youtube.com/watch?v=aD3Lf-9cb0A&t=75s.