

Install and start microros agent

Install and start microros agent

- 1、 1. Docker starts the microros agent
 - Docker starts WIFI proxy
 - Docker starts serial port agent
 - Agent startup failure
- 2、 Source code to start microros agent
 - Install tinyxml2 dependencies
 - Install python3-rosdep tool
 - Compile micro_ros_setup environment
 - Compile micro_ros_agent environment
 - Source code to start microros agent
 - ROS2 source code starts WIFI agent
 - ROS2 source code starts serial port agent

Note: Docker startup mode and source code startup mode can be installed on the same system at the same time, but only one of the startup modes can be selected at startup. Under normal circumstances, it is simple and convenient to use the docker startup method. It is recommended to use the Docker startup method.

1、 1. Docker starts the microros agent

This time, the microros agent is started through docker, so the system docker needs to be set up first. The relevant docker environment has been set up in the factory system and can be run directly.

Docker starts WIFI proxy

```
docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble udp4 --port 8090 -v4
```

```
root@micro:~$ docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8090 -v4
[1705977885.692932] info | UDPv4AgentLinux.cpp | init |
running... | port: 8090
[1705977885.693647] info | Root.cpp | set_verbose_level | 1
logger setup | verbose_level: 4
```

Among them, --port 8090 is the network port number, and -v4 is the LOG printing level. The higher the value, the more printing will be done. Modifications can be made according to actual conditions.

If you need to end the proxy, press Ctrl+C in the terminal to exit the proxy.

Note that you cannot close the terminal directly, otherwise docker will continue to run in the background.

If the microcontroller starts the reconnection agent multiple times, causing ROS2 to search for multiple identical nodes, it does not actually affect the use. Press Ctrl+C to end the agent, then reset the microcontroller and reconnect to the agent.

Docker starts serial port agent

```
docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble serial --dev /dev/ttyUSB0 -b 921600 -v4
```

```
~$ docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host microros/micro-ros-agent:humble serial --dev /dev/ttyUSB0 -b 921600 -v4
[1705977936.382362] info      | TermiosAgentLinux.cpp | init
| running...              | fd: 3
[1705977936.382954] info      | Root.cpp               | set_verbose_level
| logger setup            | verbose_level: 4
```

Among them, `--dev /dev/ttyUSB0` is the serial port device number, and `-b 921600` is the baud rate. Modifications can be made according to actual conditions.

If you need to end the proxy, press `Ctrl+C` in the terminal to exit the proxy.

Note that you cannot close the terminal directly, otherwise docker will continue to run in the background.

If the microcontroller starts the reconnection agent multiple times, causing ROS2 to search for multiple identical nodes, it does not actually affect the use. Press `Ctrl+C` to end the agent, then reset the microcontroller and reconnect to the agent.

Agent startup failure

```
~$ docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8090 -v4
[1704875081.820608] error    | UDPv4AgentLinux.cpp | init
bind error              | port: 8090, errno: 98
Error while starting IPvX agent!
[1704875081.821300] info     | UDPv4AgentLinux.cpp | fini
server stopped          | port: 8090
```

The microROS agent can only be opened in one terminal. If a terminal has already opened the microROS agent in the background, an error will be reported when opening the agent again. Please press `Ctrl+C` on the original agent terminal to exit the agent and then run the agent.

If the agent fails to be started the next time because the terminal is closed directly, you can restart the virtual machine/computer or manually end docker to solve the problem.

How to manually end docker:

Please first query the current docker process number and end the current agent docker process.

```
docker ps -a | grep microros/micro-ros-agent
docker stop xxxxxxxxxx
docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble udp4 --port 8090 -v4
```

```

[redacted]:~$ docker ps -a | grep microros/micro-ros-agent
6c8aac7b18e5    microros/micro-ros-agent:humble    "/bin/sh /micro-ros_..."    4 seconds ago    Up 3 seconds    jovial_feynman
[redacted]:~$ docker stop 6c8aac7b18e5
6c8aac7b18e5
[redacted]:~$ docker ps -a | grep microros/micro-ros-agent
[redacted]:~$ docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8090 -v4
[1704875781.918662] info    | UDPv4AgentLinux.cpp | init    |
running...    | port: 8090
[1704875781.919189] info    | Root.cpp    | set_verbose_level    | 1
logger setup    | verbose_level: 4

```

2、 Source code to start microros agent

Install tinysql2 dependencies

Enter the following command in the terminal to install tinysql2

```

cd ~/
git clone https://github.com/leethomason/tinysql2.git
cd tinysql2
mkdir build
cd build
sudo cmake ..
sudo make
sudo make install

```

Install python3-rosdep tool

Enter the following command in the terminal to install the rosdep tool. You can skip it if you have already installed it.

```

sudo apt install python3-rosdep

```

Compile micro_ros_setup environment

Activate the ROS2 environment variable. Here we take the humble version as an example. If it is already activated, you can skip the activation step.

```

source /opt/ros/humble/setup.bash

```

Create and enter the workspace uros_ws in the user directory

```

mkdir ~/uros_ws && cd ~/uros_ws
mkdir src

```

Download the micro_ros_setup file to the src folder

```
git clone -b $ROS_DISTRO https://github.com/micro-ROS/micro_ros_setup.git
src/micro_ros_setup
```

Initialize rosdep

```
sudo rosdep init
```

If there is a network problem, please add the -E parameter

```
sudo -E rosdep init
```

```
~/uros_ws$ sudo rosdep init
ERROR: cannot download default sources list from:
https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/sources.list.d/20-
default.list
Website may be down.
~/uros_ws$ sudo -E rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

rosdep update
```

If errors are reported in the above steps and rosdep still cannot be initialized, you can create a new 20-default.list file in the /etc/ros/rosdep/sources.list.d/ directory, add the following content, and then proceed to the next step.

```
# os-specific listings first
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-
homebrew.yaml osx

# generic
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
gbpdistro
https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
fuerte

# newer distributions (Groovy, Hydro, ...) must not be listed anymore, they are
being fetched from the rosdistro index.yaml instead
```

Update rosdep and install related driver packages

```
rosdep update && rosdep install --from-paths src --ignore-src -y
```

```
~/uros_ws$ rosdep update && rosdep install --from-paths src --igno
re-src -y
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
```

Compile workspace

```
colcon build
```

```
...:~/uros_ws$ colcon build
Starting >>> micro_ros_setup
Finished <<< micro_ros_setup [1.17s]

Summary: 1 package finished [2.23s]
...:~/uros_ws$ source install/local_setup.bash
```

Activate micro_ros_setup environment

```
source install/local_setup.bash
```

Compile micro_ros_agent environment

```
ros2 run micro_ros_setup create_agent_ws.sh
ros2 run micro_ros_setup build_agent.sh
```

```
...:~/uros_ws$ ros2 run micro_ros_setup create_agent_ws.sh
..
=== ./uros/micro-ROS-Agent (git) ===
Cloning into '.'...
=== ./uros/micro_ros_msgs (git) ===
Cloning into '.'...
#All required rosdeps installed successfully
...:~/uros_ws$ ros2 run micro_ros_setup build_agent.sh
Building micro-ROS Agent
Starting >>> micro_ros_msgs
Finished <<< micro_ros_msgs [2.71s]
Starting >>> micro_ros_agent
Finished <<< micro_ros_agent [6.14s]

Summary: 2 packages finished [10.0s]
```

If an error occurs when compiling build_agent.sh, please compile again.

Source code to start microros agent

Activate micro_ros_agent agent environment

```
source ~/uros_ws/install/local_setup.sh
```

ROS2 source code starts WIFI agent

```
ros2 run micro_ros_agent micro_ros_agent udp4 --port 8090 -v4
```

```

[1705977722.729787] info      | UDPv4AgentLinux.cpp | init
running...          | port: 8090
[1705977722.730290] info      | Root.cpp            | set_verbose_level    | 1
ogger setup         | verbose_level: 4

```

Among them, --port 8090 is the network port number, and -v4 is the LOG printing level. The higher the value, the more printing will be done. Modifications can be made according to actual conditions.

If you need to end the proxy, press Ctrl+C in the terminal to exit the proxy.

ROS2 source code starts serial port agent

```
ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyUSB0 -b 921600 -v4
```

```

[1705977805.553111] info      | TermiosAgentLinux.cpp | init
| running...          | fd: 3
[1705977805.553717] info      | Root.cpp              | set_verbose_level    | 1
ogger setup         | verbose_level: 4

```

Among them, --dev /dev/ttyUSB0 is the serial port device number, and -b 921600 is the baud rate. Modifications can be made according to actual conditions.

If you need to end the proxy, press Ctrl+C in the terminal to exit the proxy.