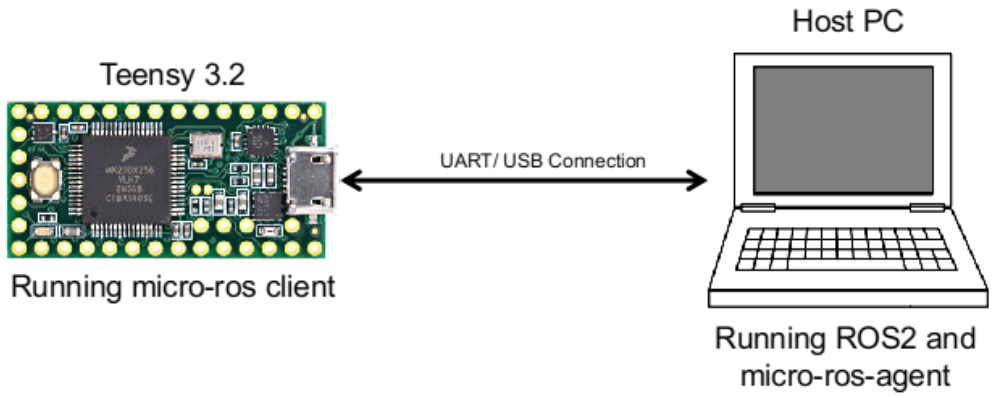
Teensy con Arduino

Written for Foxy

En este tutorial aprenderás a conectar Teensy con micro-ROS y ROS 2. También aprenderás a instalar el agente micro-ROS en sistemas Linux para comunicarse con la placa Arduino basada en Teensy mediante el IDE de Arduino. Este tutorial también abordará un tema sencillo de publicación, publicado desde Teensy y suscrito mediante la interfaz de ROS 2.

Plataforma de destino

Para empezar, necesitaremos un ordenador host con Ubuntu 20.04 nativo instalado en ROS 2 Foxy o una versión Docker de ROS 2 Foxy recién compilada (desde este enlace). Veamos también el diagrama de conexión para comprender mejor el panorama general.



Instalación de ROS 2 y micro-ROS en el ordenador host:

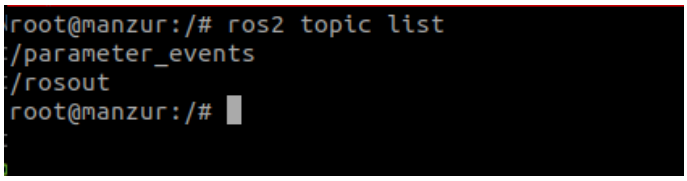
Nota: Estos primeros pasos son los mismos que en la página de instalación de micro-ROS como en este enlace

Para este tutorial, debes instalar ROS 2 Foxy Fitzroy en tu ordenador con Ubuntu 20.04 LTS. Puedes hacerlo desde los binarios, a través de los paquetes de Ubuntu, que se detallan *aquí* (https://docs.ros.org/en/foxy/Installation/Alternatives/Ubuntu-Install-Binary.html) .

Nota: De lo contrario, es posible utilizar la nueva compilación de Docker de la instalación de ROS 2 Foxy ejecutando estos comandos:

```
sudo apt install docker.io
sudo docker run -it --net=host -v /dev:/dev --privileged ros:foxy
```


Después de ejecutar el contenedor, siga el comando para verificar si ROS2 se está ejecutando y muestra la lista de temas:



Las compilaciones de Docker de la versión ROS 2 Foxy también se pueden utilizar donde no es posible instalar ROS 2 Foxy nativo desde binarios, por ejemplo, Jetson Nano con Jetpack 4.5 con Ubuntu 18.04.

Ahora, una vez que tenga una instalación de ROS 2 en la computadora o en el contenedor, siga estos pasos para instalar el sistema de compilación micro-ROS:

```
# Source the ROS 2 installation
source /opt/ros/foxy/setup.bash
# Create a workspace and download the micro-ROS tools
mkdir microros_ws
cd microros_ws
git clone -b $ROS_DISTRO https://github.com/micro-ROS/micro_ros_setup.git src/micro_ros_setup
# Update dependencies using rosdep
sudo apt update && rosdep update
rosdep install --from-paths src --ignore-src -y
# Install pip
sudo apt-get install python3-pip

# Build micro-ROS tools and source them
colcon build
source install/local_setup.bash
```

Una vez completada la instalación de micro-ROS, podemos proceder a instalar el agente de micro-ROS en el equipo host o en la versión de Docker. Dado que usaremos Teensy 3.2 y la biblioteca cliente de micro-ROS precompilada para nuestra demostración, no compilaremos el firmware y, por lo tanto, omitiremos los pasos de compilación del primer tutorial de la aplicación de micro-ROS en un RTOS (../first_application_rtos/) .

Para instalar el agente micro-ros siga los pasos a continuación:

```
# Download micro-ROS agent packages
ros2 run micro_ros_setup create_agent_ws.sh
```

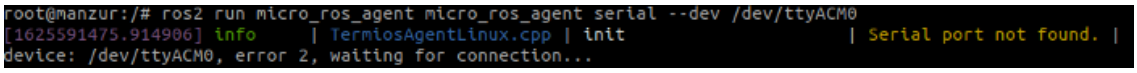
Ahora construiremos los paquetes del agente y, cuando esto esté hecho, generaremos la instalación:

```
# Build step
ros2 run micro_ros_setup build_agent.sh
source install/local_setup.bash
```

Ahora, hagamos un ensayo ejecutando el agente micro-ROS siguiendo el comando:

```
ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyACM0
```

El resultado debería mostrar algo como esto:



Esto significa que la instalación del agente se realizó correctamente. Ahora podemos continuar con el siguiente paso: instalar Arduino IDE y Teensyduino, y parchear la placa Teensy basada en Arduino para usar las bibliotecas precompiladas, como se describe *en el repositorio micro_ros_arduino* (https://github.com/micro-ROS/micro_ros_arduino#patch-teensyduino) .

Instalación de Arduino IDE, Teensyduino y configuración del parche para usar Teensy con micro-ROS y ROS2 foxy:

Siga el enlace para descargar la última versión de *Arduino 1.8.15* (https://github.com/arduino/Arduino/releases/download/1.8.15/arduino-1.8.15.tar.xz) e instálela siguiendo este *enlace para la versión de Linux* (https://www.arduino.cc/en/Guide/Linux) aquí.

Después de instalar el IDE de Arduino, descargue Teensyduino desde este *enlace* (https://www.pjrc.com/teensy/td_154/TeensyduinoInstall.linux64) y siga las instrucciones *de esta página* (https://www.pjrc.com/teensy/td_154/TeensyduinoInstall.linux64) . En resumen, las instrucciones son las siguientes:

```
1. Download the Linux udev rules and copy the file to /etc/udev/rules.d.
https://www.pjrc.com/teensy/00-teensy.rules

2. type the following command in a terminal
$ sudo cp 00-teensy.rules /etc/udev/rules.d/

3. Download and extract one of Arduino's Linux packages.
Note: Arduino from Linux distro packages is not supported.

4. Download the corresponding Teensyduino installer.

5. Run the installer in a termincal by adding execute permission and then execute it.
$ chmod 755 TeensyduinoInstall.linux64
$ ./TeensyduinoInstall.linux64
```

Ahora configuremos el parche para que el pequeño Arduino use las bibliotecas micro-ros-client precompiladas. Abra una ventana de terminal y siga los comandos a continuación: Para obtener más información, siga el enlace de GitHub desde *micro_ros_arduino* (https://github.com/micro-ROS/micro_ros_arduino/tree/foxy)

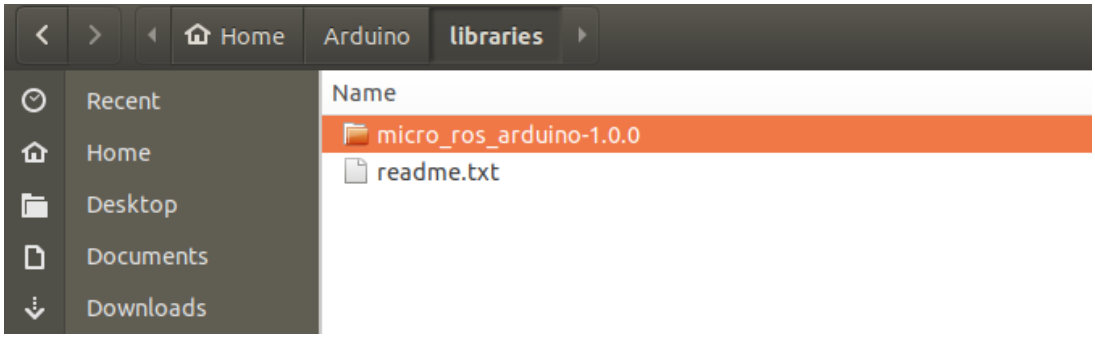
```
# for me it was $ export ARDUINO_PATH=/home/manzur/arduino-1.8.13/
export ARDUINO_PATH=[Your Arduino + Teensiduino path]

cd $ARDUINO_PATH/hardware/teensy/avr/

curl https://raw.githubusercontent.com/micro-ROS/micro_ros_arduino/foxy/extras/patching_boards/platform_teensy.txt > platform.txt
```

Una vez completada la instrucción anterior, ahora podremos utilizar el Teensy 3.2 y programarlo con las bibliotecas micro-ros-client precompiladas usando Arduino IDE.

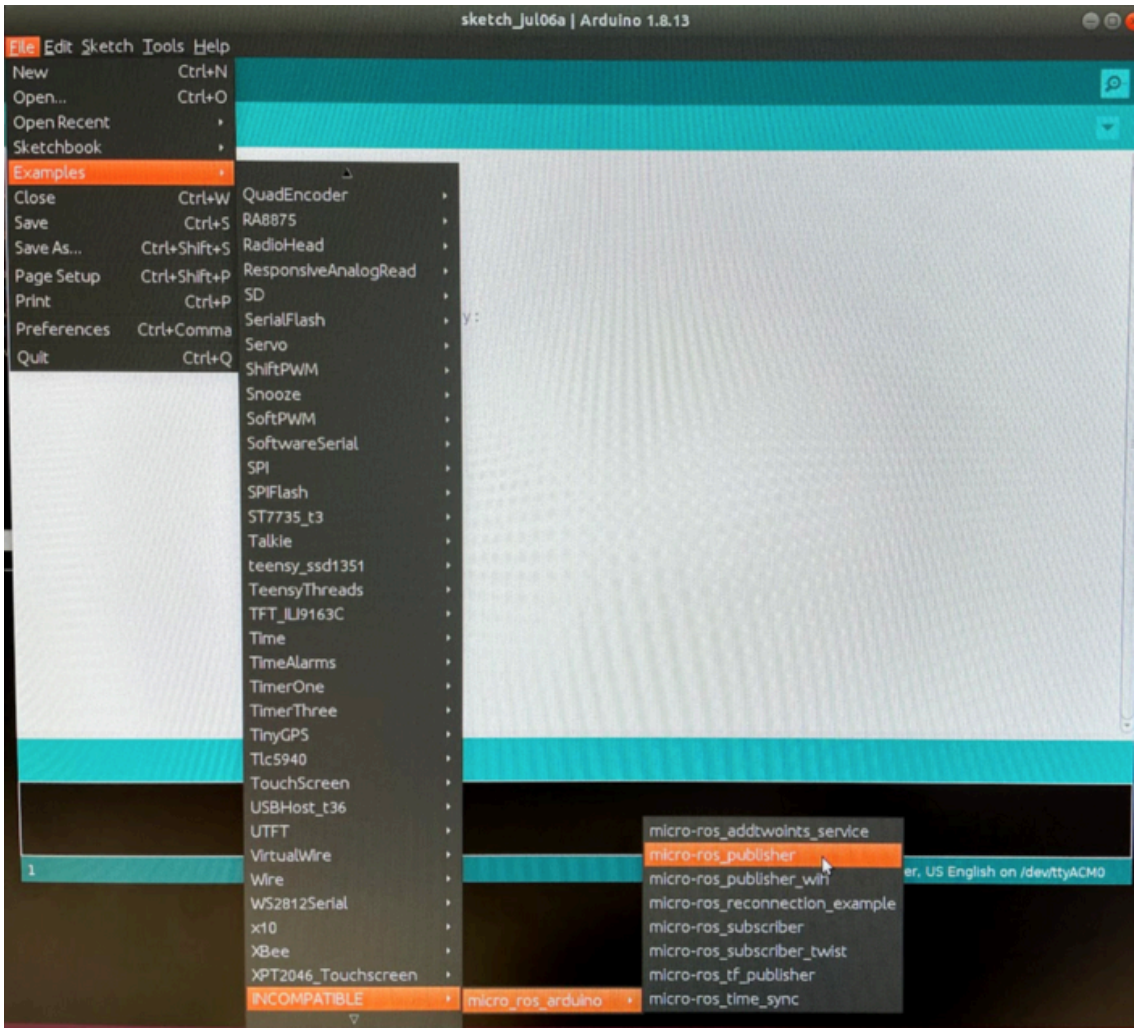
Programa el Teensy

Ahora que hemos parcheado el pequeño IDE de Arduino, podremos usar la biblioteca precompilada siguiendo estas instrucciones:

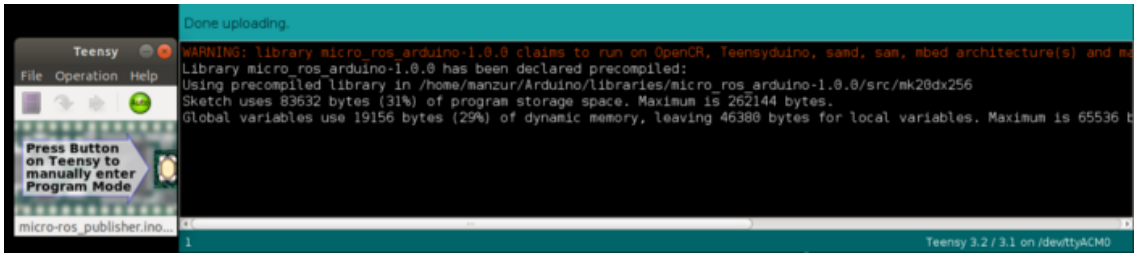
1. Vaya al *enlace de la sección de versiones* (https://github.com/micro-ROS/micro_ros_arduino/releases) y descargue la última versión de la biblioteca micro-ROS para Arduino. Coloque el archivo `/home/$USERNAME/Arduino/libraries/` como se muestra a continuación.



Una vez completado este proceso, veamos ahora la carpeta de ejemplo a continuación:



Para este tutorial y prueba, usaremos el ejemplo de mico-ros-publisher, como se muestra arriba, ya que este programa solo publicará datos enteros que aumentarán en cada ciclo. Una vez seleccionado el programa de ejemplo, cargaremos el código en el Teensy 3.2 conectado a nuestro ordenador host, que debería mostrar el siguiente resultado.



Ejecución del agente micro-ROS en ROS 2 Foxy

Ahora, desconectemos por ahora el Teensy del ordenador host. A continuación, abriremos una terminal o, en Docker, ejecutaremos el programa del agente de nuevo, como se muestra al final del paso 2. Asegúrese de obtener la ruta de ROS como se indica a continuación:

```
source /opt/ros/foxy/setup.bash
```

y luego ejecute el programa del agente:

```
ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyACM0
```

Una vez que el programa se esté ejecutando, mostrará este mensaje:

```
root@manzur:/# ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyACM0
[1625599638.035671] info     | TermiosAgentLinux.cpp | init                 | Serial port not found. |
device: /dev/ttyACM0, error 2, waiting for connection...
[1625599639.040674] info     | TermiosAgentLinux.cpp | init                 | Serial port not found. |
device: /dev/ttyACM0, error 2, waiting for connection...
[1625599640.047700] info     | TermiosAgentLinux.cpp | init                 | Serial port not found. |
device: /dev/ttyACM0, error 2, waiting for connection...
```

Luego volveremos a conectar el Teensy con el ordenador host y luego veremos que la conexión está completa y se muestra así:

```
[1625599717.423562] info     | TermiosAgentLinux.cpp | init                  | Serial port not found. |
device: /dev/ttyACM0, error 2, waiting for connection...
[1625599718.081994] info     | TermiosAgentLinux.cpp | init                  | running...             |
fd: 3
[1625599718.082523] info     | Root.cpp              | set_verbose_level     | logger setup           | ve
rbose_level: 4
[1625599719.985315] info     | Root.cpp              | create_client         | create                 | cl
ient_key: 0x517096EE, session_id: 0x81
[1625599719.985464] info     | SessionManager.hpp    | establish_session     | session established    | cl
ient_key: 0x1366333166, address: 0
[1625599720.986452] info     | ProxyClient.cpp       | create_participant    | participant created    | cl
ient_key: 0x517096EE, participant_id: 0x000(1)
[1625599720.991943] info     | ProxyClient.cpp       | create_topic          | topic created          | cl
ient_key: 0x517096EE, topic_id: 0x000(2), participant_id: 0x000(1)
[1625599721.005824] info     | ProxyClient.cpp       | create_publisher      | publisher created      | cl
ient_key: 0x517096EE, publisher_id: 0x000(3), participant_id: 0x000(1)
[1625599721.021179] info     | ProxyClient.cpp       | create_datawriter     | datawriter created     | cl
ient_key: 0x517096EE, datawriter_id: 0x000(5), publisher_id: 0x000(3)
```

Esto significa que la conexión con teensy, que contiene micro-ros-client y micro-ros-agent, está completa en el equipo host. Ahora, el momento clave es probar el tema de ROS publicado desde teensy. Esta vez, abriremos otra terminal o ventana de Docker y escribiremos lo siguiente:

```
ros2 topic list
```

Que debería listarse como se muestra a continuación:

```
root@manzur:/microros_ws# ros2 topic list
/micro_ros_arduino_node_publisher
/parameter_events
/rosout
root@manzur:/microros_ws#
```

Mira, ahora tenemos `/micro_ros_arduino_node_publisher` la publicación de temas en el host. Si escuchamos el tema, veremos algo como esto:

```
^Croot@manzur:/microros_ws# ros2 topic echo /micro_ros_arduino_node_publisher
data: 1
---
data: 2
---
data: 3
---
data: 4
---
data: 5
---
data: 6
---
data: 7
---
```

Los datos del mensaje entero aumentan en cada ciclo.

Nota: Este tutorial fue publicado por primera vez por el autor Manzur Murshid (https://github.com/shazib2t) en https://manzurmurshid.medium.com/how-to-connect-teensy-3-2-with-micro-ros-and-ros2-foxy-6c8f99c9b66a (https://manzurmurshid.medium.com/how-to-connect-teensy-3-2-with-micro-ros-and-ros2-foxy-6c8f99c9b66a) .