**Step 1 — install pre-requisites**

|  |
| --- |
| sudo apt install build-essential cmake gcc-arm-none-eabi libnewlib-arm-none-eabi doxygen git python3 |

*Dòng lệnh này thường được sử dụng khi bạn muốn thiết lập môi trường phát triển cho các dự án nhúng trên nền tảng ARM và có nhu cầu sử dụng các công cụ như CMake, GCC, Git, và Python.*

+ Setup env variables or alternatively you can open ~/.bashrc file using editor of your choice and add two export lines at the end of the file, save and source the ~/.bashrc file.

|  |
| --- |
| echo "export PICO\_TOOLCHAIN\_PATH=..." >> ~/.bashrc  echo "export PICO\_SDK\_PATH=$HOME/micro\_ros\_ws/src/pico-sdk" >> ~/.bashrc  source ~/.bashrc |

**Step 2 — Download the pico sdk code and micro-ROS example code from github repos**

|  |
| --- |
| mkdir -p ~/micro\_ros\_ws/src  cd ~/micro\_ros\_ws/src  ### clone the pico-sdk  git clone --recurse-submodules<https://github.com/raspberrypi/pico-sdk.git>  ### clone the micro-ros example repo.  git clone https://github.com/micro-ROS/micro\_ros\_raspberrypi\_pico\_sdk.git |

+ The git clone steps will take a while, so be patient, its > 400mb.

+ Your directory structure should look like this:

~

--micro\_ros\_ws

----src

------pico-sdk

------micro\_ros\_raspberrypi\_pico\_sdk

**Step 3— Build the code that will generate a uf2 file that can be copied to Pico**

|  |
| --- |
| cd micro\_ros\_raspberrypi\_pico\_sdk  mkdir build  cd build  cmake ..  make |

1. Di chuyển đến thư mục build: cd build
2. Chạy cmake .. để CMake tạo ra các tệp Makefile hoặc dự án trong thư mục build dựa trên mã nguồn trong thư mục cha.
3. Sau đó, bạn có thể chạy make để biên dịch chương trình.

**Step 4— Plug-in Pico to show up as USB drive and copy the uf2 file generated for the example code**

+ At this point hold the BOOTSEL button down on Pico and plugin the USB cable into your computer. It should show up as USB drive.

+ Cope the .uf2 file generated from build step earlier to Pico like this

|  |
| --- |
| cp pico\_micro\_ros\_example.uf2 /media/$USER/RPI-RP2 |

**Step 5— Install and configure snap plugin for micro-ROS agent**

+ Install micro-ros agent snap, enable hotplug feature and restart snap service

|  |
| --- |
| sudo snap install micro-ros-agent  *Sau khi quá trình cài đặt hoàn tất, bạn có thể bắt đầu sử dụng Micro-ROS Agent để kết nối và giao tiếp giữa ROS 2 và các thiết bị nhúng. Điều này có thể hữu ích trong các ứng dụng robotica và nhúng khi bạn muốn tích hợp ROS với các thiết bị có tài nguyên hạn chế.*  sudo snap set core experimental.hotplug=true  sudo snap get core experimental.hotplug  *Hotplug: Hotplug là khả năng kết nối hoặc ngắt kết nối các thiết bị từ hệ thống mà không cần khởi động lại. Trong ngữ cảnh của Snap, experimental.hotplug là một tính năng thử nghiệm mà Snap đang phát triển để hỗ trợ khả năng kết nối động (hotplug) cho các ứng dụng.*  *Bằng cách chạy lệnh trên, bạn đang thông báo cho Snap rằng bạn muốn kích hoạt tính năng hotplug thử nghiệm. Tính năng này có thể hữu ích trong các tình huống khi bạn muốn cài đặt, cập nhật hoặc ngắt kết nối các snap mà không cần phải khởi động lại hệ thống. Tuy nhiên, hãy lưu ý rằng vì đây là một tính năng thử nghiệm, có thể có các rủi ro và hạn chế, và nên sử dụng cẩn thận trong môi trường sản xuất*  sudo systemctl restart snapd  *systemctl: Là một tiện ích quản lý dịch vụ trên hệ điều hành Linux, giúp bạn tương tác với hệ thống init (systemd) để quản lý các dịch vụ.*   1. *restart: Là một tùy chọn của systemctl, được sử dụng để khởi động lại dịch vụ. Khi bạn chạy systemctl restart, nó sẽ dừng và sau đó khởi động lại dịch vụ ngay lập tức.*   *Bằng cách chạy lệnh trên, bạn đang yêu cầu hệ thống khởi động lại dịch vụ snapd. Điều này có thể hữu ích trong một số tình huống, ví dụ như khi bạn đã thay đổi cấu hình của snapd và muốn áp dụng những thay đổi mà không cần phải khởi động lại toàn bộ hệ thống.*  export ROS\_DOMAIN\_ID=0 |

**Step 6— Connect to Pico using snap over serial port**

+ Make sure Pico is plugged in and run following to confirm serial is working properly

|  |
| --- |
| snap interface serial-port  Khi bạn chạy lệnh snap interface serial-port, nó sẽ hiển thị thông tin về các quyền truy cập được cấp phép cho các snap với interface "serial-port". Các thông tin này có thể bao gồm các quyền như đọc, ghi, và điều khiển các cổng nối tiếp.  Việc kiểm tra quyền truy cập này có thể quan trọng khi bạn muốn cài đặt hoặc chạy một ứng dụng Snap liên quan đến việc sử dụng cổng nối tiếp, để đảm bảo rằng ứng dụng có đủ quyền để thực hiện các tác vụ liên quan đến serial port. |

$ snap interface serial-port

name: serial-port

summary: allows accessing a specific serial port

plugs:

- micro-ros-agent

slots:

- snapd:pico (allows accessing a specific serial port)

+ Connect the plug and slot for serial

|  |
| --- |
| snap connect micro-ros-agent:serial-port snapd:pico |

**Step 7— Start the micro-ROS example application on Pico**

|  |
| --- |
| sudo micro-ros-agent serial --dev /dev/ttyACM0 baudrate=115200 |

+ If the LED light does not light up, unplug and plug the USB cable back. Wait for few seconds and it should show that the micro-ROS node is starting to publish

install [micro-ROS-Agent](https://github.com/micro-ROS/micro-ROS-Agent) in ROS foxy

sudo snap install micro-xrce-dds-agen

snap set micro-xrce-dds-agent daemon=true

sudo snap refresh core --edge

sudo snap set core experimental.hotplug=true

sudo systemctl restart snapd

snap interface serial-port

snap connect micro-xrce-dds-agent:serial-port snapd:pico

sudo micro-xrce-dds-agent serial --dev /dev/ttyACM0 baudrate=115200