**MicroROS PingPong Zephyr Guideline**

**Created by Marlar Chan**

1. **MicroROS Setup**

# Source the ROS 2 installation

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

#Create a workspace and download the micro-ROS build

$mkdir microros\_ws

$cd microros\_ws

Download src folder

$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-path src --ignore-src -y

#Install Pipe

sudo apt-get install python3-pip

/\* Need to install once \*/

#Colcon Installation

$sudo apt install python3-colcon-common-extensions

#Build micro-ROS tools and source them

#It creates build, install, log folders

colcon build

source install/local\_setup.bash

1. **Create a micro-ROS Agent**

ros2 run micro\_ros\_setup create\_agent\_ws.sh

ros2 run micro\_ros\_setup build\_agent.sh

1. **Create firmware for build-in Board**

ros2 run micro\_ros\_setup create\_firmware\_ws.sh zephyr stm32f429i\_disc1

1. **Modify CMakeList.txt Line 5 to below**

*firmware/zephyr\_apps/app/ping\_pong/CMakeLists.txt*

set(COMPATIBLE\_BOARDS stm32f429i\_disc1 disco\_l475\_iot1 olimex\_stm32\_e407 native\_posix nucleo\_h743zi nucleo\_f746zg)

1. **Firmware Configuration**

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

$source install/local\_setup.bash

$ros2 run micro\_ros\_setup configure\_firmware.sh ping\_pong --transport serial --dev 1

*[--dev is for UART1]*

1. **Build firmware**

ros2 run micro\_ros\_setup build\_firmware.sh

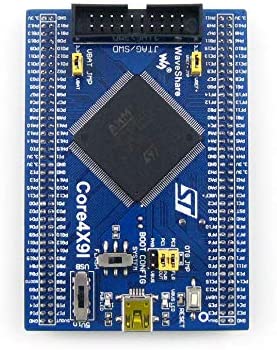
#Hex File can be found in the following location

*/uros\_ws/firmware/build/zephyr*

1. **Flashing Firmware to Board**

* USART1 use for running microROS agent.  
  (PA9 and PA10)
* **No need NSH for Zephyr**

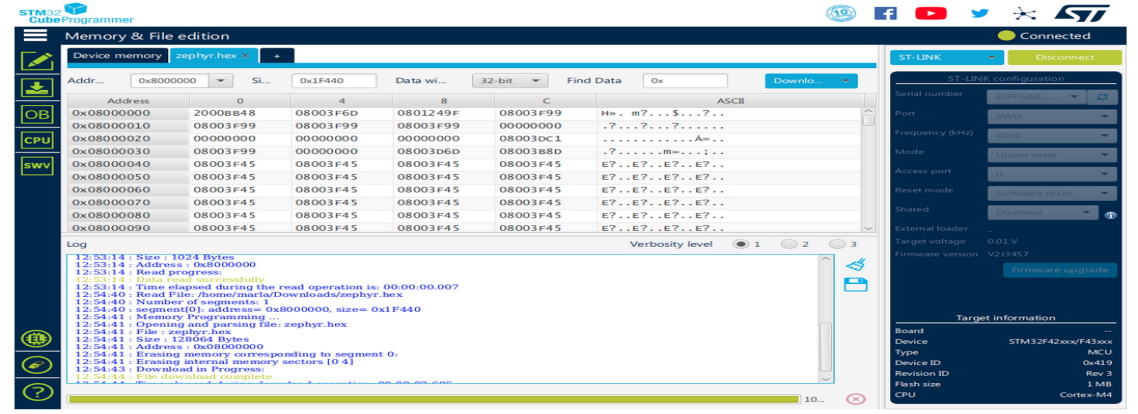
ST-Link Debug



Power Supply

USART1

USB\_NSH



**Create and Build Agent**

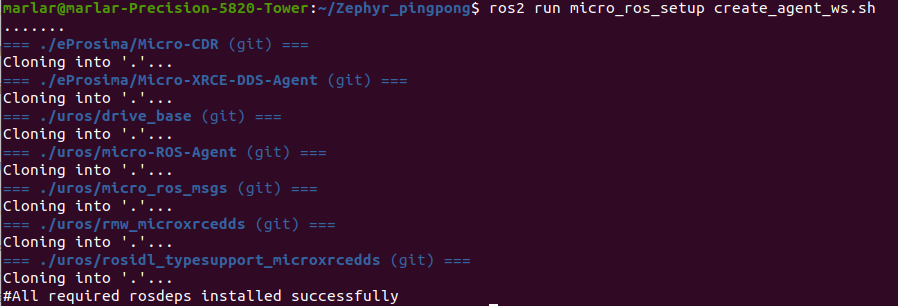
New Terminal

$cd microros\_waveshare/

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

$source install/local\_setup.bash

$ros2 run micro\_ros\_setup create\_agent\_ws.sh



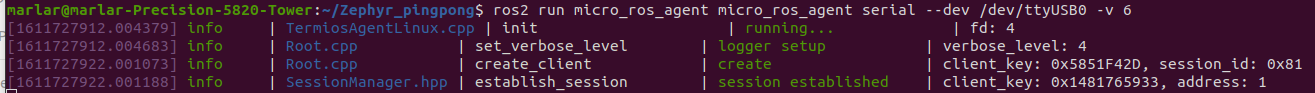
$ros2 run micro\_ros\_setup build\_agent.sh

**Running Agent**

source /opt/ros/foxy/setup.bash

source install/local\_setup.bash

ros2 run micro\_ros\_agent micro\_ros\_agent serial --dev /dev/ttyUSB0 -v 6



**UROS PingPong Testing**

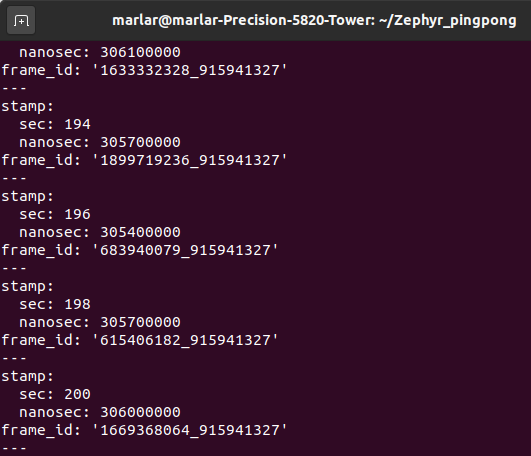
Add New terminal,

source /opt/ros/foxy/setup.bash

source install/local\_setup.bash

# Subscribe to micro-ROS ping topic

ros2 topic echo /microROS/ping



Add New terminal,

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

$source install/local\_setup.bash

# Subscribe to micro-ROS pong topic

$ros2 topic echo /microROS/pong

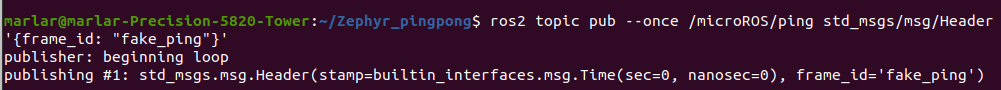
**Add New terminal,**

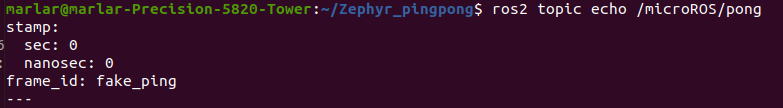
$source /opt/ros/foxy/setup.bash

$source install/local\_setup.bash

# Send a fake ping

ros2 topic pub --once /microROS/ping std\_msgs/msg/Header '{frame\_id: "fake\_ping"}'





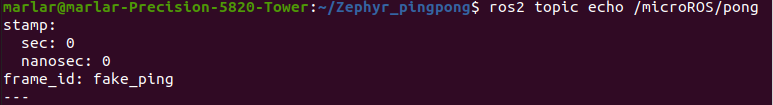
Add New terminal,

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

$source install/local\_setup.bash

micro-ROS pong publisher will answer with a pong. As a consequence, in the pong subscriber console, we should see the board’s answer to our fake\_ping

ros2 topic echo /microROS/pong



MicroROS Sensor Test

# Source the ROS 2 installation

source /opt/ros/foxy/setup.bash

# Create a workspace and download the micro-ROS tools

$mkdir zephyr\_microros

$cd zephyr\_microros

#Download src folder which include micro\_ros\_setup

$git clone -b $ROS\_DISTRO https://github.com/micro-ROS/micro\_ros\_setup.git src/micro\_ros\_setup

# Update dependencies using rosdepsudo apt update && rosdep update

rosdep install --from-path src --ignore-src -y

# Install pipsudo apt-get install python3-pip

# Build micro-ROS tools and source them

colcon build

source install/local\_setup.bash

#Create Firmware

$ros2 run micro\_ros\_setup create\_firmware\_ws.sh zephyr stm32f429i\_disc1

Create demo adxl372\_test

1. Copy Pingpong folder and rename as adxl372\_test.
2. Compare with adxl372 directory

Firmware/zephyrproject/zephyr/samples/sensor/ adxl372

Copy Src/main.c into adxl372\_test.

#To modify CMakeList.txt Line 5 to below

firmware/zephyr\_apps/app/adxl372\_test/CMakeLists.txt

set(COMPATIBLE\_BOARDS stm32f429i\_disc1 disco\_l475\_iot1 olimex\_stm32\_e407 native\_posix nucleo\_h743zi nucleo\_f746zg)

firmware/zephyr\_apps/app/adxl372\_test/boards/

Copy nrf52dk\_nrf52832.overlay and paste it in the same folder.

Rename it to stm32f429i\_disc1.overlay.

Edit as follow

&spi5 {

cs-gpios = <&gpioe 4 GPIO\_ACTIVE\_LOW>;

adxl372@0 {

compatible = "adi,adxl372";

reg = <0>;

spi-max-frequency = <8000000>;

label = "ADXL372";

int1-gpios = <&gpioe 6 GPIO\_ACTIVE\_HIGH>;

};

};

/\*\*\*\* SPI5, PE4 is CS, PE6 is INT1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Default Settings is ADXL372\_PEAK\_DETECT\_MODE.

If you want to change to ADXL372\_MEASUREMENT\_MODE,

Go to /zephyrproject/zephyr/drivers/sensor/adxl372/

Edit Kconfig: Line 30 to

choice

prompt "Operating mode"

default ADXL372\_MEASUREMENT\_MODE

#Compile Firmwrae

source /opt/ros/foxy/setup.bash

source install/local\_setup.bash

ros2 run micro\_ros\_setup configure\_firmware.sh adxl372\_test --transport serial --dev 1

#Build Firmware

ros2 run micro\_ros\_setup build\_firmware.sh