How to send data through NVIDIA Jetson TX2 USB port using ROS serial library.

Created by: David Rodriguez ([davidrm@dc.tohoku.ac.jp](mailto:davidrm@dc.tohoku.ac.jp))

Last update: Nov. 7, 2019

Updated by: David Rodriguez

# Preparation

Connect the device to the USB port or USB Hub port on the NVIDIA Jetson TX2. We need to know what the */dev* path to the usb is. *Note: usually it will be assigned to /dev/ttyUSB0*.

There are many ways to do this. A simple way is to create and run the following bash shell file.

#!/bin/bash

#Simple file to find the dev path of peripherals.

for sysdevpath in $(find /sys/bus/usb/devices/usb\*/ -name dev); do

(

syspath="${sysdevpath%/dev}"

devname="$(udevadm info -q name -p $syspath)"

[[ "$devname" == "bus/"\* ]] && continue

eval "$(udevadm info -q property --export -p $syspath)"

[[ -z "$ID\_SERIAL" ]] && continue

echo "/dev/$devname - $ID\_SERIAL"

)

done

Save this file as find\_dev.sh and run it.

$ bash find\_dev.sh

Save this file as find\_dev.sh and run it.

Next thing we need is to install a serial port library call *serial. Note: change* melodic *for whatever ROS distribution you are using. Documentation about the serial library can be found in* [*http://wjwwood.io/serial/doc/1.1.0/index.html*](http://wjwwood.io/serial/doc/1.1.0/index.html)

$ sudo apt install ros-melodic-serial

# Operation

The following is an example code to send a single byte through USB.

// send\_byte.cpp program

#include <ros/ros.h>

#include <serial/serial.h>

#include <stdio.h>

#include <string>

#include <sstream>

#include <time.h>

#include <stdlib.h>

std::string port = "/dev/ttyUSB0"; //Include here the dev path of your USB device!!

uint32\_t baud = 115200; //setup baud

serial::Serial serial(port, baud, serial::Timeout::simpleTimeout(200));

int main(int argc, char\*\* argv)

{

char cmd;

ros::init(argc, argv, "send\_byte"); // ノードの宣言

while(ros::ok())

{

ros::spinOnce();

std::cout << "cmd >> ";

std::cin >> cmd;

serial.write((uint8\_t\*)(&cmd), 1); // データを送信する

ROS\_INFO("%s", &cmd); // デバッグ用

}

return 0;

}

Make sure the variable *port* contains the right */dev* path. After declaring all dependencies and executable files and building the workspace, we need to assign to the USB port the right permissions to write and read. Run the following command. *Note: make sure you use your own* dev *path.*

$ sudo chmod a+rwx /dev/ttyUSB0

Run roscore and the send\_byte node. One byte will be sent through every time you input a character into the command line.

$ cmd >> A

Done.