

ROS串列埠例程

原始碼(瀏覽器下載並解壓): [anrot_demo_ros.zip](#)

本文件介紹如何在ROS下來讀取ANROTIMU的數據，並提供了c++語言例程式碼，通過執行ROS命令，執行相應的節點，就可以看到列印到終端上的資訊。

- 測試環境：Ubuntu16.04
- ROS版本：ROS Kinetic Kame
- 測試裝置：Hi221 Hi226/229 CH100 CH110 CH104 CH108(USB)

1. 安裝USB-UART驅動

Ubuntu 系統自帶CP210x的驅動，預設不需要安裝串列埠驅動。將除錯版連線到電腦上時，會自動識別裝置。識別成功后，會在dev目錄下出現一個對應的裝置：ttyUSBx

檢查USB-UART裝置是否被Ubuntu識別：

1. 打開終端，輸入 `ls /dev`，先檢視已經存在的串列埠裝置。
2. 檢視是否已經存在 `ttyUSBx` 這個裝置檔案，便於確認對應的埠號。x表示USB裝置號，由於Ubuntu USB裝置號為從零開始依次累加，所以多個裝置每次開機後設備號是不固定的，需要確定裝置的裝置號。
3. 接下來插入USB線，連線除錯板，然後再次執行 `ls /dev`。dev目錄下多了一個裝置，如圖：

```
linux@ubuntu:/dev$ ls
agpgart      loop3      shm        tty32      tty63      ttyS7
autofs       loop4      snapshot   tty33      tty7        ttyS8
block        loop5      snd        tty34      tty8        ttyS9
bsg          loop6      sr0        tty35      tty9        ttyUSB0
btrfs-control loop7      stderr     tty36      ttyprintk   uinput
bus          loop-control stdin       tty37      ttyS0
```

`ttyUSB0` 檔案就是除錯版在ubuntu系統中產生的裝置(後面的數字是不固定的，有可能是 `ttyUSB1` 或 `ttyUSB2`)

5. 打開USB裝置的可執行許可權：

```
1 | $ sudo chmod 777 /dev/ttyUSB0
```

2. 安裝ROS serial軟體包

本例程依賴ROS提供的serial包實現串列埠通訊。

1. 首先執行如下命令，下載安裝serial軟體包：

```
1 | $ sudo apt-get install ros-kinetic-serial
```

2. 然後輸入 `roscd serial` 命令，進入serial下載位置，如果安裝成功，就會出現如下資訊：

```
1 | $:/opt/ros/kinetic/share/serial
```

3. 如果出現裝不上，可以到本文件的最後檢視本人遇到的問題和解決方式。

3. 編譯serial_imu_ws工作空間

1. 打開終端進入/examples/ROS/serial_imu_ws 目錄
2. 執行 `catkin_make` 命令，編譯成功后出現完成度100%的資訊。

4. 修改串列埠波特率和裝置號

1. 在Ubuntu環境中，支援的波特率為115200，460800，921600。本例程使用的預設波特率是115200，預設打開的串列埠名稱是/dev/ttyUSB0。
2. 如果您需要更高的輸出頻率，請編輯serial_imu.cpp檔案，修改serial_imu.cpp檔案中的#define區段，改為其他波特率。

```
1 | #define IMU_SERIAL ("/dev/ttyUSB0")
2 | #define BAUD      (115200)
```

注意修改後需要回到serial_imu_ws目錄下，重新執行 `catkin_make` 命令

5. 顯示數據

本例程提供了三種檢視數據方式：

1. 顯示所有的數據資訊，便於檢視數據。
2. 列印ROS標準imu_msg 數據
3. rviz工具實現視覺化
4. 3D顯示

5.1：輸出IMU原始數據

1. 打開另一個終端，執行：

```
1 | $ roslaunch imu_launch imu_msg.launch imu_package:=0x91
```

2. 如果執行失敗，提示找不到相應的launch檔案，則需要配置環境，在目前終端執行：

```
1 | $source <serial_imu_ws_dir>/devel/setup.bash
```

3. 執行成功后，就可以看到所有的資訊：

```
1 |      Devie ID:      0
2 |      Run times: 0 days 3:26:10:468
3 |      Frame Rate:   100Hz
4 |      Acc(G):      0.933    0.317    0.248
5 |      Gyr(deg/s):  -0.02     0.30    -0.00
6 |      Mag(uT):      0.00     0.00     0.00
7 |      EuL(R P Y):   52.01    -66.63   -60.77
8 |      Quat(W X Y Z): 0.770    0.066   -0.611   -0.172
```

5.2：輸出ROS標準 Imu.msg

1. 在windows系統下進行配置模組，使能四元數輸出。
2. 預設皆使用0x91協議，如果有客戶自行修改過，再做此步驟修改回0x91：
使用Windows的ANROTIMU-UI上位機進行配置：先把模組連接到PC主機上，開啟ANROTIMU-UI，連接對應的com口，點選 **工具** ---> **配置模組**，在彈出的新視窗中，點選**ATCMD**，然後在輸入框中輸入AT指令：**AT+SETPTL=0x91**，點選發送，接收區最後顯示 **ok**，說明配置成功，斷電重啟模組。執行 `roslaunch imu_launch imu_msg.launch` 命令。執行成功后，就可以看到ROS定義的IMU話題訊息：

```

1 header:
2   seq: 595
3   stamp:
4     secs: 1595829903
5     nsecs: 680423746
6   frame_id: "base_link"
7 orientation:
8   x: 0.0663746222854
9   y: -0.611194491386
10  z: -0.17232863605
11  w: 0.769635260105
12 orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
13 angular_velocity:
14   x: 0.0851199477911
15   y: 0.0470183677971
16   z: 0.00235567195341
17 angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
18 linear_acceleration:
19   x: 0.93323135376
20   y: 0.317857563496
21   z: 0.247811317444
22 linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

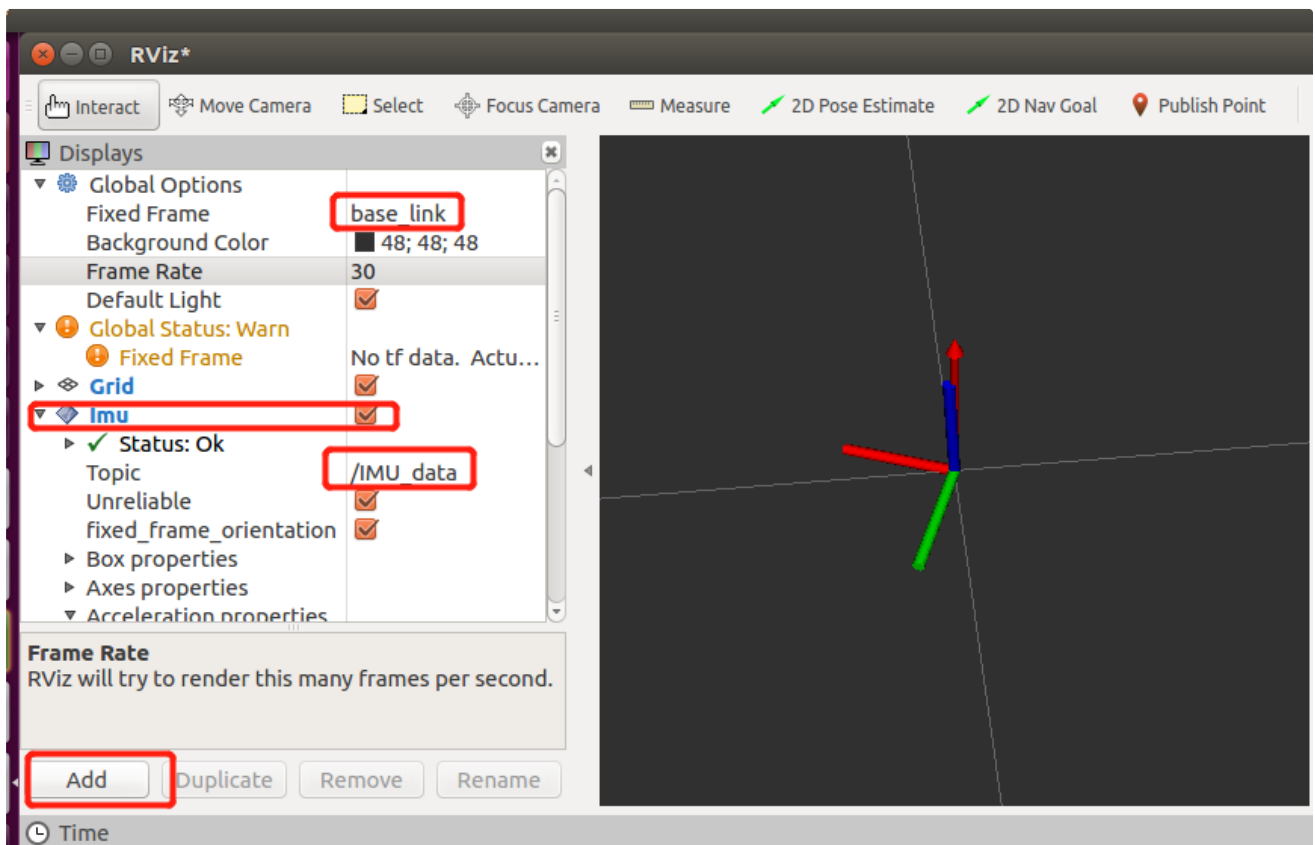
```

5.3 : rviz視覺化

1、同上節，使能模組四元數輸出

2、打開終端，執行 `roslaunch imu_launch imu_rviz.launch` 命令，執行成功后，rviz工具被打開。

3、先點選左下角的Add標籤，然後在彈出視窗中，選擇 By display type標籤，查詢rviz_imu_plugin；找到之後，選擇它下面的imu標籤，點選OK，這時，我們可以看到rviz的左側的展示視窗中已經成功新增上了Imu的標籤。在FixedFrame中填入 `base_link`，topic中新增 `/IMU_data`。這時，可以看到座標系隨感測器改變而改變。



5.4 : 3D顯示

1、使能模組的四元數輸出

2、打開終端，執行 `roslaunch imu_launch imu_display_3D.launch` 命令，執行成功后，會出現3D圖形顯示

6. FAQ

1.如果是第一次裝ROS serial包，有很大的可能會失敗，因為本人在裝的時候，遇到了這個問題，這裡把解決方法提供出來，節約大家的時間。

當在終端執行 `sudo apt-get install ros-kinetic-serial` 這條命令的時候，有可能會提示你

```
linux@ubuntu:~$ sudo apt-get install ros-kinetic-serial
[sudo] password for linux:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package ros-kinetic-serial
linux@ubuntu:~$
```

爲了提供素材，serial故意輸錯的。

本人的解決辦法是：

```
1 | $cd /etc/apt/sources.list.d
2 | $sudo vi ros-latest.list
```

打開這個檔案之後，一般這個檔案中只有一個可用的源，就是指沒有被註釋的，現在把它註釋掉，在它的開頭輸入#即可註釋。

然後另起一行輸入：`deb https://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu/ xenial main`

然後儲存，關閉檔案。打開終端，執行 `sudo apt-get update`，然後執行 `sudo apt-get install ros-kinetic-serial`。

2、有時候主板上需要插好多的usb裝置，爲了方便開發，通常會編寫一個usb埠約束檔案。如果是不同型號的usb裝置，可以通過裝置的id號來區分。如果是同型號的裝置，他們的id號都是一樣的，這個時候就需要更多的細分資訊來區分不同的usb裝置。接下來就操作一下如何區分同型號的usb裝置。

```
1 | linux@ubuntu:~$ lsusb
2 | Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
3 | Bus 002 Device 012: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge /
4 | myAVR mySmartUSB light
5 | Bus 002 Device 011: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge /
6 | myAVR mySmartUSB light
7 | Bus 002 Device 010: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge /
8 | myAVR mySmartUSB light
9 | Bus 002 Device 008: ID 0e0f:0008 VMware, Inc.
10 | Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
11 | Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
12 | Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
13 | linux@ubuntu:~$
```

觀察上面的內容，發現有三個usb裝置的id號完全一樣，使用簡單的id號區分行不通了，需要更多的裝置資訊。

```

1 linux@ubuntu:~$ ls /dev
2 agpgart          loop3            shm             tty32          tty63          ttyS7
3 autofs           loop4            snapshot        tty33          tty7           ttyS8
4 block           loop5            snd             tty34          tty8           ttyS9
5 bsg             loop6            sr0            tty35          tty9           ttyUSB0
6 btrfs-control    loop7            stderr          tty36          ttyprintk      ttyUSB1
7 bus             loop-control     stdin          tty37          ttyS0          ttyUSB2
8 .....(未全部放出)

```

到這一步，`dev`檔案中產生三個usb裝置檔案，分別是：`ttyUSB0`，`ttyUSB1`，`ttyUSB2`。

現在先看`ttyUSB0`的詳細資訊：

```

1 linux@ubuntu:~$ udevadm info --attribute-walk --name=/dev/ttyUSB0
2 #通過這個命令可以檢視指定埠的詳細資訊
3 .....
4   ATTRS{devpath}=="2.2"
5   ATTRS{idProduct}=="ea60"
6   ATTRS{idVendor}=="10c4"
7   ATTRS{ltm_capable}=="no"
8   ATTRS{manufacturer}=="Silicon Labs"
9   ATTRS{maxchild}=="0"
10  ATTRS{product}=="CP2104 USB to UART Bridge Controller"
11  ATTRS{quirks}=="0x0"
12  ATTRS{removable}=="unknown"
13  ATTRS{serial}=="01E34546"
14  .....(資訊太多了，就不全部放出來了，大家可以自己去看看詳細的資訊，這裡只放出本次需要關心的資訊)

```

然後是`ttyUSB1`的詳細資訊：

```

1 linux@ubuntu:~$ udevadm info --attribute-walk --name=/dev/ttyUSB1
2 #通過這個命令可以檢視指定埠的詳細資訊
3 .....
4   ATTRS{devpath}=="2.3"
5   ATTRS{idProduct}=="ea60"
6   ATTRS{idVendor}=="10c4"
7   ATTRS{ltm_capable}=="no"
8   ATTRS{manufacturer}=="Silicon Labs"
9   ATTRS{maxchild}=="0"
10  ATTRS{product}=="CP2102N USB to UART Bridge Controller"
11  ATTRS{quirks}=="0x0"
12  ATTRS{removable}=="unknown"
13  ATTRS{serial}=="9c1d818b48aeeb119d082897637728c5"
14  .....(資訊太多了，就不全部放出來了，大家可以自己去看看詳細的資訊，這裡只放出本次需要關心的資訊)
15

```

最後是`ttyUSB2`的詳細資訊：

```

1 linux@ubuntu:~$ udevadm info --attribute-walk --name=/dev/ttyUSB2
2 #通過這個命令可以檢視指定埠的詳細資訊
3 .....
4   ATTRS{devnum}=="27"
5   ATTRS{devpath}=="2.4"
6   ATTRS{idProduct}=="ea60"
7   ATTRS{idVendor}=="10c4"
8   ATTRS{ltm_capable}=="no"
9   ATTRS{manufacturer}=="Silicon Labs"

```

```

10     ATTRS{maxchild}=="0"
11     ATTRS{product}=="CP2104 USB to UART Bridge Controller"
12     ATTRS{quirks}=="0x0"
13     ATTRS{removable}=="unknown"
14     ATTRS{serial}=="02228956"
15     .....(資訊太多了，就不全部放出來了，大家可以自己去看看詳細的資訊，這裡只放出本次需要關心的資訊)

```

通過上邊的三個串列埠裝置的資訊，發現ATTRS{serial}=="xxx"這一項，看起來特別隨意。實際上這個是硬體的id號，也是硬體的唯一id號，通過這個號，給它起一個別名，這樣一來，只要這個硬體id號被識別到，dev下就會出現自定義的埠名稱裝置檔案，實現永久繫結埠號。

```

1  linux@ubuntu:~$ cd /etc/udev/rule.d/
2  linux@ubuntu:/etc/udev/rules.d$ ls
3  70-snap.core.rules 70-ttyusb.rules 99-vmware-scsi-udev.rules
4  #這一步是看看都有哪些約束檔案，避免檔名重複
5  linux@ubuntu:~$ sudo vi defined_serial.rules
6  #這一步自定義一個串列埠約束檔名稱，後綴為'.rules'

```

然後在這個檔案中輸入如下內容：

```

1 KERNEL=="ttyUSB*",ATTRS{serial}=="9c1d818b48aeeb119d882897637728c5",ATTRS{idVendor}=="10c4",ATTRS{idProduct}=="ea60",MODE:="0777",SYMLINK+="Hi226"
2 KERNEL=="ttyUSB*",ATTRS{serial}=="01E34546",ATTRS{idVendor}=="10c4",ATTRS{idProduct}=="ea60",MODE:="0777",SYMLINK+="BLUETOOTH"
3 KERNEL=="ttyUSB*",ATTRS{serial}=="02228956",ATTRS{idVendor}=="10c4",ATTRS{idProduct}=="ea60",MODE:="0777",SYMLINK+="CH110"

```

格式如下：

```

1  KERNEL=="ttyUSB*", ATTRS{serial}=="xxx", ATTRS{idVendor}=="xxx", ATTRS{idProduct}=="xxx",
    MODE:="0777 (埠的許可權)",SYMLINK+="(自定義名稱)"

```

把對應的資訊填對，最後儲存並退出檔案，執行：

```

1  linux@ubuntu:~$ service udev reload
2  root privileges required
3  linux@ubuntu:~$ service udev restart
4  linux@ubuntu:~$ ls /dev
5  agpgart      loop1          sg1            tty32          tty7           ttyS9
6  autofs       loop2          shm            tty33          tty8           ttyUSB0
7  block        loop3          snapshot       tty34          tty9           ttyUSB1
8  BLUETOOTH    loop4          snd            tty35          ttyprintk      ttyUSB2
9  ....
10 CH110        mcelog         tty0           tty40          ttyS13         vcs1
11 ....
12 Hi221        rfkill         tty22          tty54          ttyS27         vfio
13 ....

```

現在可以看到，自定義的usb埠名稱已經出來了，在操作的時候，直接操作對應的裝置檔案就好了，不用去理會埠的編號是多了。