

多机通信管理文档

1 概述

ROS 2用于通讯的默认中间件是DDS。在DDS中，不同逻辑网络共享物理网络的主要机制称为域 DomainID。同一域上的ROS 2节点可以自由地相互发现并发送消息，理论上来说，只要机器连接在同一网络下并且 DomianID 相同，cyberdog、PC 之间均能相互通信。然而若过多的设备连接到同一个 DDS 域中时，会导致数据可靠性和实时性的下降，cyberdog 默认关闭了多机通信的能力。

2 Cyberdog 启用多机通信步骤

2.1 cyberdog 网络接口名称查询

- cybreog 通过扫码手机二维码的方式连接互联网
- 通过 APP 查看 cyberdog 的 IP 地址

设置 --> 通用 --> WIFI --> IP 地址



如上图所示， cyberdog 的 IP 地址为 192.168.31.90

- 登入 cyberdog 查看连接网络的端口名称

```
1 ssh mi@192.168.31.90 # 密码 123  
2 ifconfig
```

终端将打印类下列信息，该机器连接的网络名称为 wlan0

```
1 can0: flags=193<UP,RUNNING,NOARP> mtu 16
2         unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10
3             RX packets 109 bytes 851 (851.0 B)
4             RX errors 0 dropped 0 overruns 0 frame 0
5             TX packets 13 bytes 20 (20.0 B)
6             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
7
8 can1: flags=193<UP,RUNNING,NOARP> mtu 16
9         unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10
10            RX packets 17 bytes 122 (122.0 B)
11            RX errors 0 dropped 0 overruns 0 frame 0
12            TX packets 6 bytes 5 (5.0 B)
13            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
14            device interrupt 64
15
16 docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
17         inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
18             ether 02:42:b1:c9:f6:03 txqueuelen 0 (Ethernet)
19             RX packets 0 bytes 0 (0.0 B)
20             RX errors 0 dropped 0 overruns 0 frame 0
21             TX packets 0 bytes 0 (0.0 B)
22             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
23
24 eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
25         inet 192.168.44.1 netmask 255.255.255.0 broadcast 192.168.44.255
26             ether 48:b0:2d:5f:bb:21 txqueuelen 1000 (Ethernet)
27             RX packets 1719 bytes 333701 (333.7 KB)
28             RX errors 0 dropped 0 overruns 0 frame 0
29             TX packets 101 bytes 9915 (9.9 KB)
30             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
31             device interrupt 37
32
33 l4tbr0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
34         inet 192.168.55.1 netmask 255.255.255.0 broadcast 192.168.55.255
35         inet6 fe80::d88a:74ff:fed2:aa95 prefixlen 64 scopeid 0x20<link>
36         inet6 fe80::1 prefixlen 128 scopeid 0x20<link>
37             ether da:8a:74:d2:aa:95 txqueuelen 1000 (Ethernet)
38             RX packets 84 bytes 12075 (12.0 KB)
39             RX errors 0 dropped 0 overruns 0 frame 0
40             TX packets 106 bytes 16676 (16.6 KB)
41             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
42
43 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
44         inet 127.0.0.1 netmask 255.0.0.0
```

```
45      inet6 ::1  prefixlen 128  scopeid 0x10<host>
46          loop txqueuelen 1  (Local Loopback)
47              RX packets 168616  bytes 45254904 (45.2 MB)
48              RX errors 0  dropped 0  overruns 0  frame 0
49              TX packets 168616  bytes 45254904 (45.2 MB)
50              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
51
52  rndis0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
53          inet6 fe80::d88a:74ff:fed2:aa95  prefixlen 64  scopeid 0x20<link>
54          ether da:8a:74:d2:aa:95  txqueuelen 1000  (Ethernet)
55              RX packets 19  bytes 3340 (3.3 KB)
56              RX errors 0  dropped 0  overruns 0  frame 0
57              TX packets 119  bytes 24603 (24.6 KB)
58              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
59
60  usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
61          inet6 fe80::d88a:74ff:fed2:aa97  prefixlen 64  scopeid 0x20<link>
62          ether da:8a:74:d2:aa:97  txqueuelen 1000  (Ethernet)
63              RX packets 93  bytes 12720 (12.7 KB)
64              RX errors 0  dropped 0  overruns 0  frame 0
65              TX packets 112  bytes 26776 (26.7 KB)
66              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
67
68  wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
69          inet 192.168.31.90  netmask 255.255.255.0  broadcast 192.168.31.255
70          inet6 fe80::dd91:a002:3001:2ed2  prefixlen 64  scopeid 0x20<link>
71          ether 70:f7:54:b8:8b:49  txqueuelen 1000  (Ethernet)
72              RX packets 45  bytes 4648 (4.6 KB)
73              RX errors 0  dropped 0  overruns 0  frame 0
74              TX packets 74  bytes 8070 (8.0 KB)
75              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
76
```

2.2 cyberdog 多机通信文件修改



多机通信相关文件

1. </etc/mi/cyclonedds.xml>
2. /etc/mi/ros2_env.conf

- 登入 cyberdog 查看并修改 cyclonedds.xml

```
1 ssh mi@192.168.31.90
2 sudo vim /etc/mi/cyclonedds.xml
```

```
1 // 修改前
2 <?xml version="1.0" encoding="UTF-8" ?>
3 <CycloneDDS xmlns="https://cdds.io/config" xmlns:xsi="http://www.w3.org/2001/X
4     <Domain id="42">
5         <General>
6             <NetworkInterfaceAddress>lo</NetworkInterfaceAddress>
7             <AllowMulticast>false</AllowMulticast>
8         </General>
9         <Discovery>
10            <ParticipantIndex>auto</ParticipantIndex>
11            <MaxAutoParticipantIndex>30</MaxAutoParticipantIndex>
12            <Peers>
13                <Peer address="localhost"/>
14            </Peers>
15        </Discovery>
16    </Domain>
17 </CycloneDDS>
18 -----
19 // 修改后
20
21 <?xml version="1.0" encoding="UTF-8" ?>
22 <CycloneDDS xmlns="https://cdds.io/config" xmlns:xsi="http://www.w3.org/2001/X
23     <Domain id="42">
24         <General>
25             <NetworkInterfaceAddress>wlan0</NetworkInterfaceAddress>
26             <AllowMulticast>true</AllowMulticast>
27         </General>
28         <Discovery>
29             <ParticipantIndex>auto</ParticipantIndex>
30             <MaxAutoParticipantIndex>30</MaxAutoParticipantIndex>
31             <Peers>
32                 <Peer address="localhost"/>
33             </Peers>
34         </Discovery>
35     </Domain>
36 </CycloneDDS>
```

- 查看并修改 ros2_env.conf

`ros2_env.conf` 默认不需要修改，只有当 `cyclonedds.xml` 中的 `DomianID` 修改时，`ros2_env.conf` 中的 DomainID 做相应改变即可

```
1 sudo vim /etc/mi/ros2_env.conf
```

终端显示：

```
1 export ROS_VERSION=2
2 export ROS_PYTHON_VERSION=3
3 export ROS_LOCALHOST_ONLY=0
4 export ROS_DISTRO=galactic
5 export ROS_DOMAIN_ID=42
6 export LD_LIBRARY_PATH=/usr/lib/aarch64-linux-gnu/gazebo-11/plugins:/opt/ros2/
7 export PYTHONPATH=/opt/ros2/cyberdog/lib/python3.6/site-packages:/opt/ros2/gal
8 export AMENT_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
9 export CMAKE_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
10 export COLCON_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
11 export PATH=/opt/ros2/galactic/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/u
12 export PKG_CONFIG_PATH=/opt/ros2/galactic/lib/aarch64-linux-gnu/pkgconfig:/opt
13 export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
14 export CYCLONEDDS_URI=file:///etc/mi/cyclonedds.xml
15 export XDG_RUNTIME_DIR=/run/user/1000
16 eval "$(register-python-argcomplete3 ros2)"
17 eval "$(register-python-argcomplete3 colcon)"
18
19 # export DISPLAY=:0
20 # export ROS_LOCALHOST_ONLY=1
21
```

3 修改名称空间

cyberdog 开机时会根据 eth 信息自动生成名称空间，每次开机名称空间都会变化，为了在多级通信的过程中通过名称空间的方式区分机器，现提供固定名称空间的方案。

需要修改两个路径下的同名 `manual.py` 文件，将其中函数 `get_namespace()` 的默认值修改为固定值。

```
1 /opt/ros2/cyberdog/share/cyberdog_bringup/bringup/manual.py
2 /opt/ros2/cyberdog/lib/python3.6/site-packages/mi/cyberdog_bringup/manual.py
```

修改为：

```
1 # 修改前
2
3 def get_namespace():
4     hostname = socket.getfqdn(socket.gethostname())
5     mac = get_mac('eth0')
6     namespace = hostname + '_' + mac
7     namespace = re.sub('[^0-9a-zA-Z]+', '_', namespace)
8     return namespace
9
10 -----
11 # 修改后
12 def get_namespace():
13     hostname = socket.getfqdn(socket.gethostname())
14     mac = get_mac('eth0')
15     namespace = hostname + '_' + mac
16     namespace = re.sub('[^0-9a-zA-Z]+', '_', namespace)
17     return "custom_namespace"
```

4 PC配置（可选）

若需要使用 PC 监控不同机器中所有的 node，也可以将 PC 加入到同一 DomainID 域中。



PC机需具备的条件

- Ubuntu 20.04 系统
- 安装 ROS2 Galactic

4.1 shell 环境变量配置

```
1 ### ROS2 ####
2 source /opt/ros/galactic/setup.bash
3 export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
4 export CYCLONEDDS_URI=file:///home/user_name/.cyclonedds.xml
5 export ROS_DOMAIN_ID=42
```

其中 user_name 为Linux下的用户名，需根据电脑实际情况修改。

4.2 创建文件 .cyclonedds.xml

- 查看 PC 机的网络配置，确认其 WiFi 端口名称，本例中其名称为 wlan0。

```
1 docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
2         inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
3             ether 02:42:9e:de:c9:13 txqueuelen 0 (Ethernet)
4                 RX packets 0 bytes 0 (0.0 B)
5                 RX errors 0 dropped 0 overruns 0 frame 0
6                 TX packets 0 bytes 0 (0.0 B)
7                 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
8
9 enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
10        ether 90:20:3a:1a:b1:ff txqueuelen 1000 (Ethernet)
11            RX packets 0 bytes 0 (0.0 B)
12            RX errors 0 dropped 0 overruns 0 frame 0
13            TX packets 0 bytes 0 (0.0 B)
14            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
15
16 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
17         inet 127.0.0.1 netmask 255.0.0.0
18         inet6 ::1 prefixlen 128 scopeid 0x10<host>
19             loop txqueuelen 1000 (Local Loopback)
20             RX packets 53316 bytes 11898226 (11.8 MB)
21             RX errors 0 dropped 0 overruns 0 frame 0
22             TX packets 53316 bytes 11898226 (11.8 MB)
23             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
24
25 wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
26         inet 192.168.31.183 netmask 255.255.255.0 broadcast 192.168.31.255
27         inet6 fe80::83be:d38d:c71b:e869 prefixlen 64 scopeid 0x20<link>
28             ether 54:8d:5a:4e:80:6d txqueuelen 1000 (Ethernet)
29             RX packets 7460715 bytes 10967240230 (10.9 GB)
30             RX errors 0 dropped 0 overruns 0 frame 0
31             TX packets 615802 bytes 78246211 (78.2 MB)
32             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 创建文件 .cyclonedds.xml，其内容如下

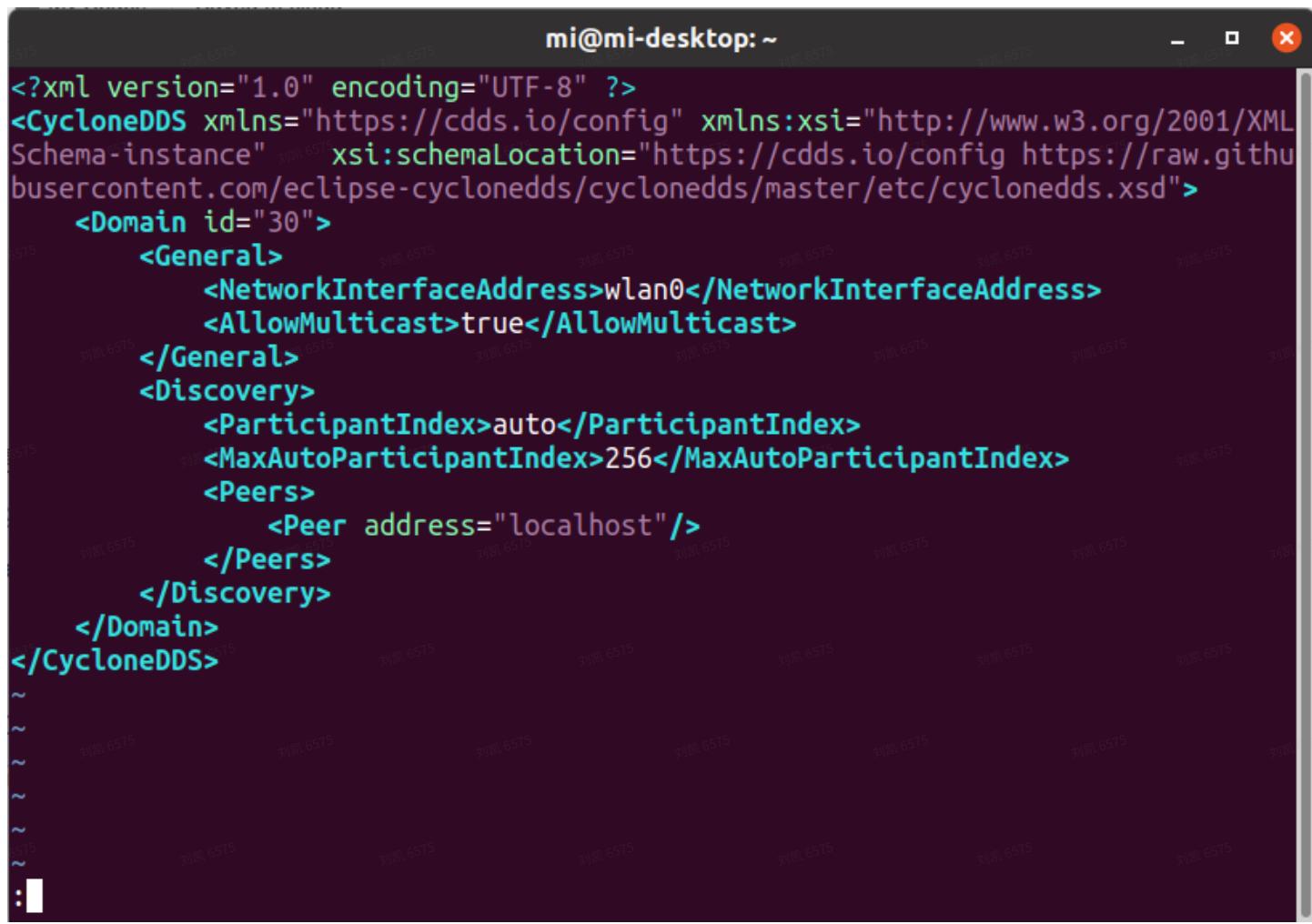
```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <CycloneDDS xmlns="https://cdds.io/config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://cdds.io/config CycloneDDS.xsd">
```

```
3 <Domain id="42">
4   <General>
5     <NetworkInterfaceAddress>wlan0</NetworkInterfaceAddress>
6     <AllowMulticast>true</AllowMulticast>
7   </General>
8   <Discovery>
9     <ParticipantIndex>auto</ParticipantIndex>
10    <MaxAutoParticipantIndex>30</MaxAutoParticipantIndex>
11    <Peers>
12      <Peer address="localhost"/>
13    </Peers>
14  </Discovery>
15 </Domain>
16 </CycloneDDS>
17
```

5 测试示例

5.1 相关文件内容展示

- .cyclonedds.xml



The screenshot shows a terminal window titled "mi@mi-desktop: ~" displaying the contents of a ".cyclonedds.xml" configuration file. The file is an XML document with the following structure:

```
<?xml version="1.0" encoding="UTF-8" ?>
<CycloneDDS xmlns="https://cdds.io/config" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://cdds.io/config https://raw.githubusercontent.com/eclipse-cyclonedds/cyclonedds/master/etc/cyclonedds.xsd">
  <Domain id="30">
    <General>
      <NetworkInterfaceAddress>wlan0</NetworkInterfaceAddress>
      <AllowMulticast>true</AllowMulticast>
    </General>
    <Discovery>
      <ParticipantIndex>auto</ParticipantIndex>
      <MaxAutoParticipantIndex>256</MaxAutoParticipantIndex>
      <Peers>
        <Peer address="localhost"/>
      </Peers>
    </Discovery>
  </Domain>
</CycloneDDS>
```

- ros_env.config

```
mi@mi-desktop: ~
```

```
export ROS_VERSION=2
export ROS_PYTHON_VERSION=3
export ROS_LOCALHOST_ONLY=0
export ROS_DISTRO=galactic
export ROS_DOMAIN_ID=30
export LD_LIBRARY_PATH=/usr/lib/aarch64-linux-gnu/gazebo-11/plugins:/opt/ros2/galactic/opt/yaml_cpp_vendor/lib:/opt/ros2/galactic/opt/rviz_ogre_vendor/lib:/opt/ros2/galactic/lib:/opt/ros2/cyberdog/lib
export PYTHONPATH=/opt/ros2/cyberdog/lib/python3.6/site-packages:/opt/ros2/galactic/lib/python3.6/site-packages
export AMENT_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
export CMAKE_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
export COLCON_PREFIX_PATH=/opt/ros2/cyberdog:/opt/ros2/galactic
export PATH=/opt/ros2/galactic/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
export PKG_CONFIG_PATH=/opt/ros2/galactic/lib/aarch64-linux-gnu/pkgconfig:/opt/ros2/galactic/lib/pkgconfig
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
export CYCLONEDDS_URI=file:///etc/mi/cyclonedds.xml
export XDG_RUNTIME_DIR=/run/user/1000
eval "$(register-python-argcomplete3 ros2)"
eval "$(register-python-argcomplete3 colcon)"

# export DISPLAY=:0
:|
```

- manual.py

```
mi@mi-desktop: /opt/ros2/cyberdog/share/cyberdog Bringup /bringup - □ X

if len(mac) != 0:
    break
else:
    if net_list.count(target_network_card_name):
        mac = get_shell(
            'cat /sys/class/net/' + target_network_card_name + '/address'
        ).strip()
if len(mac) == 0:
    mac = str(uuid.UUID(int=uuid.getnode().hex[-12:]))
    mac = ':' . join([mac[e:e+2] for e in range(0, 11, 2)])
mac = re.sub('[:]+', '_', mac)
return mac

#
# 获取 namespace
#
def get_namespace():
    hostname = socket.getfqdn(socket.gethostname())
    mac = get_mac('eth0')
    namespace = hostname + '_' + mac
    namespace = re.sub('[^0-9a-zA-Z]+', '_', namespace)
    #return namespace
    return "cyberdog_2"

# 165,1 Bot
```

5.2 运行测试

- ROS2 node 间的发现能测试。

本次测试使用了两台 cyberdog，一台的名称空间为 cyberdog_1，一台的名称空间为 cyberdog_2。

```
1 # 查看 cyberdog_manager 相关的 node
2 ros2 node list | grep cyberdog_manager
```

mi@mi-desktop:~

```
mi@mi-desktop:~$ ros2 node list |grep cyberdog_manager
```

```
/cyberdog_1/cyberdog_manager
```

```
/cyberdog_1/cyberdog_manager_error
```

```
/cyberdog_2/cyberdog_manager
```

```
/cyberdog_2/cyberdog_manager_error
```

```
mi@mi-desktop:~$
```

刘凯 6575

6575

刘凯 6575

6575

刘凯 6575

6575

刘凯 6575

• 机器之间的通信能力测试

测试方案：一台机器发布 ROS2 topic，另一台订阅该Topic。

cyberdog A 发布数据

```
1 ros2 topic pub /test_topic std_msgs/msg/Bool
```

cyberdog B 订阅数据

```
1 ros2 topic list | grep test_topic
```

```
2 ros2 topic echo /test_topic
```

```
mi@mi-desktop:~$ ls
cyberdog_1 Desktop Downloads Music Public TomLog
debug Documents examples.desktop Pictures Templates Videos
mi@mi-desktop:~$ ros2 topic pub /test_topic std_msgs/msg/Bool
publisher: beginning loop
publishing #1: std_msgs.msg.Bool(data=False)
publishing #2: std_msgs.msg.Bool(data=False)
publishing #3: std_msgs.msg.Bool(data=False)
publishing #4: std_msgs.msg.Bool(data=False)
publishing #5: std_msgs.msg.Bool(data=False)
```

```
mi@mi-desktop:~$ ls
cyberdog_2  Documents  examples.desktop  Pictures  Templates  Videos
Desktop  Downloads  Music  Public  TomLog
mi@mi-desktop:~$ ros2 topic list | grep test_topic
/test_topic
mi@mi-desktop:~$ ros2 topic echo /test_topic
data: false
---
[ ]
[ ]
```