

## ROS2GO 准备工作

安装 rmtt\_ros 系列功能包：

从~/tt\_ws/src/（自行创建文件夹）打开一个 terminal，通过 git clone 下载功能包：

```
git clone https://github.com/cavayangtao/rmtt_ros.git
```

或者：

```
git clone https://gitee.com/cavayangtao/rmtt\_ros.git
```

在该工作空间（~/tt\_ws/）下打开终端：

编译工作空间：catkin\_make\_isolated；

运行功能包中的节点前，请在该终端及新终端中输入：source ./devel\_isolated/setup.bash；

或在 home 中打开终端，输入：gedit ~/.bashrc，将以上命令粘贴入文档并保存；

关闭所有终端，再次打开新终端可直接运行功能包中的节点。

## TT 无人机 EXT 模块安装

安装完成后的效果：



图 1：Robomaster TT 旋翼无人机

## TT 无人机电池充电

充电逻辑说明：

每节电池依次充电，并不是同时充电；

绿色闪烁为充电中，绿色常亮为充满，黄色常亮为等待充电。

注意事项：

随时保持电池处于充电状态，方便使用。

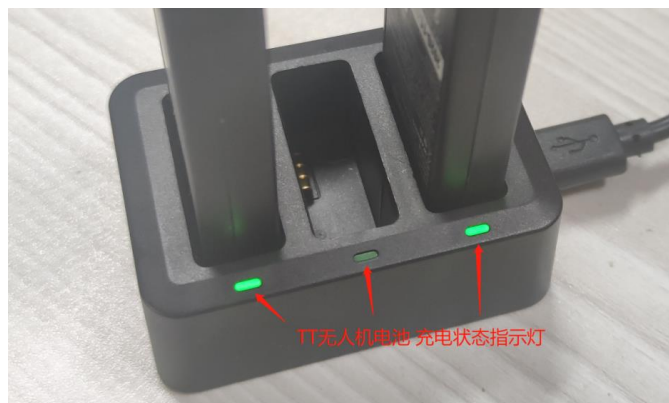


图 2：充电指示

## 激活 TT 无人机

通过 Tello 手机客户端对 TT 无人机进行激活：

开机和关机均通过短按 TT 无人机的电源键，稍等片刻；

将手机通过 wifi 连接至 TT 无人机，将 EXT 模块调至 AP 模式；

打开 Tello 手机客户端，**全新的 TT 无人机需要首先连接客户端进行激活，才能继续后续任务；**

看看 Tello 手机客户端都提供了哪些功能，飞一下吧；

每块电池满电的飞行时间在十分钟左右，记得时刻给不使用的电池充电；

之后我们将通过 ROS 对 TT 无人机进行控制和编程。

## TT 网络模式配置

TT 无人机具有 STA（路由）模式与 AP（热点）模式。STA 模式可以在通过一台电脑连接和控制多台无人机，本节课我们只使用 AP 模式。

TT 进入 AP 模式：

将 EXT 模块上的“模式开关”切换到“AP”

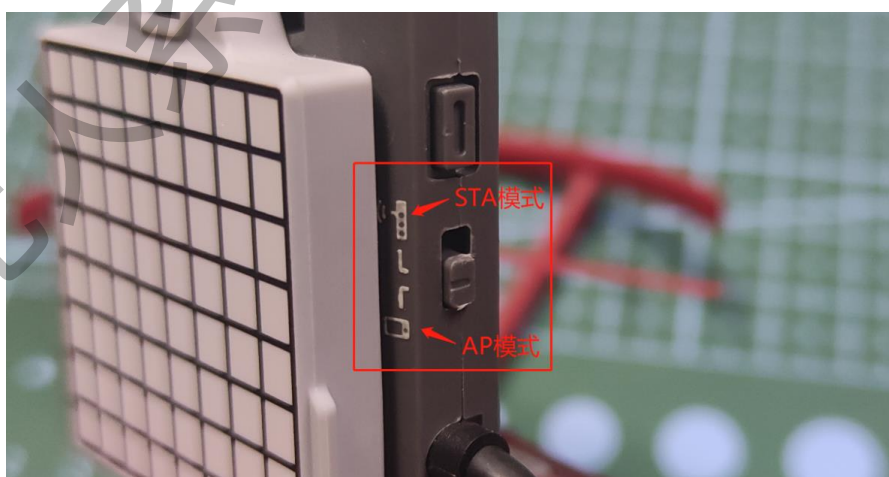


图 3：切换路由和热点模式

短按电源按钮，启动 TT 无人机，无人机的 wifi 名称于 LED 模块背面：RMTT-xxx。

稍等片刻(TT 启动后)，使用 PC 连接到 RMTT-xxx 网络。

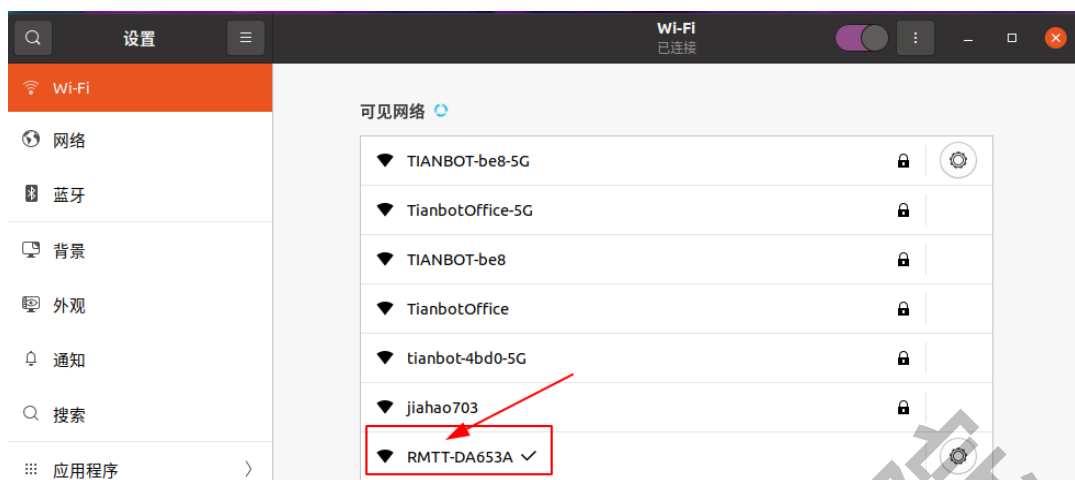


图 4：连接到 RMTT-xxx 网络

## 在 ROS 中驱动 TT 无人机

硬件驱动启动文件：rmtt\_bringup.launch。

参数说明：

drone\_ip: TT 无人机的 IP 地址；

local\_ip: PC 的 IP 地址；

drone\_name: 名字空间，用于区分多台 TT 无人机（在后面的应用中，对名字空间会有更感性的认识，来理解名字空间的作用）

local\_ip 通过在新的终端执行 ifconfig 来获取。

在新的终端执行以下命令：

```
roslaunch rmtt_driver rmtt_bringup.launch
```

或者：

```
roslaunch rmtt_driver rmtt_bringup.launch drone_name:=rmtt_01
```

注意：由于目前我们只连接一台无人机，无需设定 drone\_ip 和 local\_ip。

执行以上命令后，我们可以看到如下图所示的返回信息：

```
setting /run_id to eca810c8-8af6-11eb-910f-e5c311262486
process[rosout-1]: started with pid [10759]
started core service [/rosout]
process[rmtt_driver-2]: started with pid [10766]
[INFO] [1616407871.568722]: RoboMaster TelloTalent Initializing...
[INFO] [1616407871.901708]: drone sn: 
[INFO] [1616407871.912234]: start camera video stream!
[INFO] [1616407871.943901]: battery level: 100
```

在返回信息中，我们可以看到 TT 无人机的序列号、视频流以及电量信息。看到以上信息时，表示 TT 无人机已经被正确的启动，并在 ROS 中驱动成功。

## TT 无人机 ROS 驱动的话题简要说明

首先，根据上节的方法，在 ROS 中驱动 TT 无人机。

使用 rostopic list 命令查看 ROS 驱动发布的话题列表（看看有无 drone\_name:=rmtt\_01 话题名称的区别）：

```

tianbot@ros2go:~$ rostopic list
/altitude
/battery
/camera_info
/cmd_vel
/flip
/image_raw
/image_raw/compressed
/imu_data
/land
/led
/mission_pad_id
/mled
/pose
/rosout
/rosout_agg
/takeoff
/tof_btm
/tof_ext
tianbot@ros2go:~$ 

```

使用 `rostopic info xxx` 查看 xxx 话题的基本信息:

```

/tof_ext
tianbot@ros2go:~$ rostopic info /cmd_vel
Type: geometry_msgs/Twist

Publishers: None

Subscribers:
* /rmtt_driver (http://ros2go:44015/)

```

图中以 `/cmd_vel` 话题为例, 我们可以看到发布者: None, 订阅者: `/rmtt_driver`。

如果在驱动 TT 无人机时使用了 `drone_name` 参数, 假设 `drone_name:=rmtt_01`, 那么我们将会看到如下的话题列表:

```

tianbot@ros2go:~$ rostopic list
/rmtt_01/altitude
/rmtt_01/battery
/rmtt_01/camera_info
/rmtt_01/cmd_vel
/rmtt_01/flip
/rmtt_01/image_raw
/rmtt_01/image_raw/compressed
/rmtt_01/imu_data
/rmtt_01/land
/rmtt_01/led
/rmtt_01/mission_pad_id
/rmtt_01/mled
/rmtt_01/pose
/rmtt_01/takeoff
/rmtt_01/tof_btm
/rmtt_01/tof_ext
/rosout
/rosout_agg
tianbot@ros2go:~$ 

```

接下来我们一一介绍上图中所示的话题的具体功能:

<code>/rmtt_01/altitude</code>	#TT 无人机内置气压计海拔数据
<code>/rmtt_01/battery</code>	#TT 无人机电池电量百分比
<code>/rmtt_01/camera_info</code>	#TT 无人机前置摄像头参数信息
<code>/rmtt_01/cmd_vel</code>	#控制 TT 无人机运动

/rmtt_01/flip	#控制 TT 无人机翻转
/rmtt_01/image_raw	#TT 无人机前置摄像头画面
/rmtt_01/image_raw/compressed	#TT 无人机前置摄像头画面(压缩)
/rmtt_01/imu_data	#TT 无人机内置 IMU 数据
/rmtt_01/land	#控制 TT 无人机降落
/rmtt_01/led	#EXT 模块顶端全彩 LED 的颜色
/rmtt_01/mission_pad_id	#DJI 定位毯坐标 id
/rmtt_01/mled	#控制 EXT 模块全彩 LED 点阵显示字符串
/rmtt_01/pose	#TT 无人机相对于 DJI 定位毯的姿态
/rmtt_01/takeoff	#控制 TT 无人机起飞
/rmtt_01/tof_btm	#TT 无人机本体底部的 tof 传感器距离值（范围：0.3-1.5）
/rmtt_01/tof_ext	#EXT 模块前端 tof 传感器距离值（范围：0.01-1.5）

在新的终端部利用 `rostopic echo xxx` 命令查看 xxx 话题的数据，和发布频率（回忆上节课内容）。用手移动无人机查看诸如/rmtt\_01/imu\_data、/rmtt\_01/tof\_btm、/rmtt\_01/tof\_ext等话题的数据及其意义。

**小贴士：**在使用 `rostopic list` 命令的时候加入 `-v` 指令，可以快速地区分哪些话题是被发布的与订阅的。

```
Published topics:
* /rosout_agg [rosgraph_msgs/Log] 1 publisher
* /rosout [rosgraph_msgs/Log] 1 publisher
* /rmtt_01/imu_data [sensor_msgs/Imu] 1 publisher
* /rmtt_01/pose [geometry_msgs/PoseStamped] 1 publisher
* /rmtt_01/mission_pad_id [std_msgs/UInt8] 1 publisher
* /rmtt_01/image_raw/compressed [sensor_msgs/CompressedImage] 1 publisher
* /rmtt_01/image_raw [sensor_msgs/Image] 1 publisher
* /rmtt_01/camera_info [sensor_msgs/CameraInfo] 1 publisher
* /rmtt_01/tof_btm [sensor_msgs/Range] 1 publisher
* /rmtt_01/tof_ext [sensor_msgs/Range] 1 publisher
* /rmtt_01/altitude [std_msgs/Float32] 1 publisher
* /rmtt_01/battery [std_msgs/Float32] 1 publisher

Subscribed topics:
* /rosout [rosgraph_msgs/Log] 1 subscriber
* /rmtt_01/cmd_vel [geometry_msgs/Twist] 1 subscriber
* /rmtt_01/takeoff [std_msgs/Empty] 1 subscriber
* /rmtt_01/land [std_msgs/Empty] 1 subscriber
* /rmtt_01/flip [std_msgs/Empty] 1 subscriber
* /rmtt_01/led [std_msgs/ColorRGBA] 1 subscriber
* /rmtt_01/mled [std_msgs/String] 1 subscriber
```

查看/rmtt\_01/image\_raw话题的视频画面：

打开终端，输入以下命令：`rqt`

（首次我们打开 `rqt` 工具，会看到一个空白的窗口，不要慌，这是正常的）





图 5: rqt 工具界面

点击 Plugins - Visualization - Image View, 添加视频画面查看组件

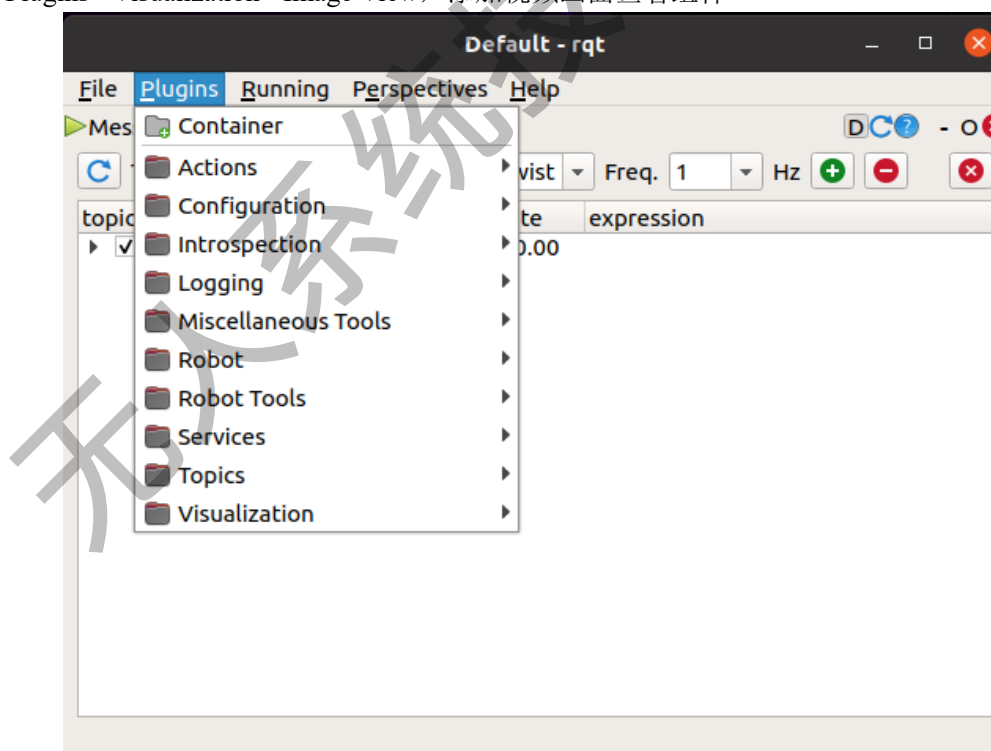


图 6: 添加组件菜单栏

在打开的 Image View 插件中, 我们选择/rmtt\_01/image\_raw 话题, 就可以看到画面了。  
还记不记得在 rviz 里怎么看到画面? 尝试一下。



图 7：在 rqt 中显示 TT 无人机摄像头画面

## LED 字符显示

默认状态下，LED 显示屏的最下方一行指示前方障碍物的距离，看看测距传感器在哪？

在一个新的终端运行：

```
rostopic pub /rmtt_01/mled std_msgs/String 'nwpu'.
```

也可以利用 rqt 工具选择对应话题，Plugins - Topics - Message Publisher，选择/rmtt\_01/mled。按下拉箭头，在 expression 处输入要显示的字符（比如“nwpu”）然后打勾发布：

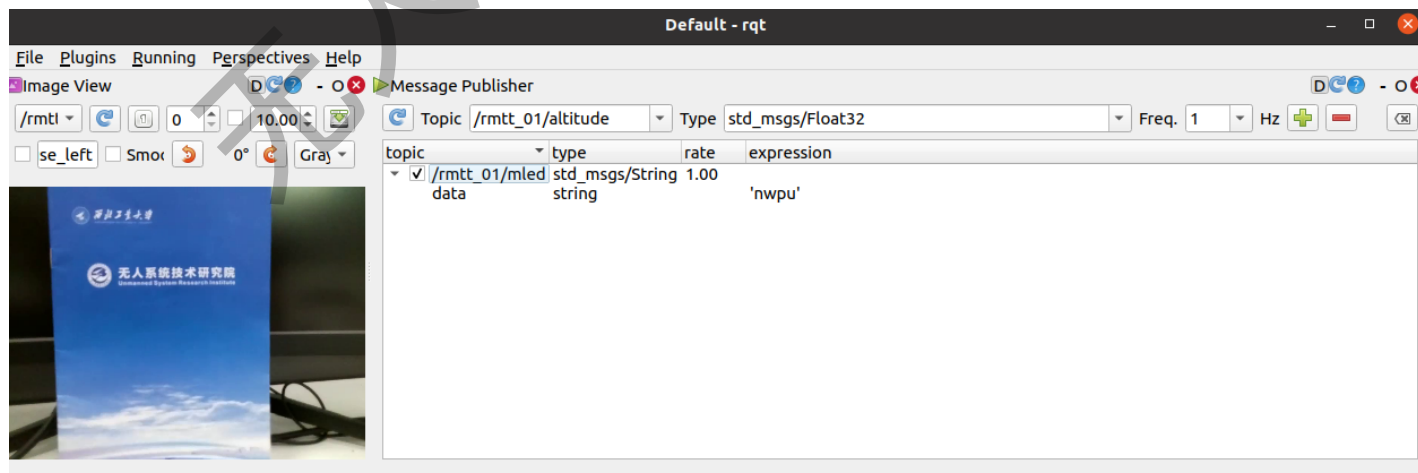


图 8：在 rqt 中发布话题消息

## 在 ROS 中使用键盘遥控 TT 无人机

根据之前的方法，在 ROS 中驱动 TT 无人机(drone\_name:=rmtt\_01)。

打开一个新的终端，输入如下指令：

```
roslaunch rmtt_teleop rmtt_teleop_key.launch drone_name:=rmtt_01
```

在本例中，我们依旧以 drone\_name:=rmtt\_01 为例，在启动 teleop 遥控程序时，也加入 drone\_name 参数，如上所示：

```
ROS_MASTER_URI=http://localhost:11311

process[rmtt_teleop_key-1]: started with pid [11419]
the rosdep view is empty: call 'sudo rosdep init' and 'rosdep update'

Reading from the keyboard and Publishing to Twist!
-----
Moving around:
mode two
LEFT HAND:
      w
    a      d
      x
RIGHT HAND:
  u  i  o
  j  k  l
  m  ,  .
- : to takeoff
= : to land
(need to hold SHIFT)
>/< : increase/decrease linear speed by 10%
X/Z : increase/decrease angular speed by 10%
anything else : stop
CTRL-C to quit

currently:      speed 0.2      turn 0.2

```

遥控程序启动后，我们在终端中可以看到如上返回信息。

保持此终端激活状态，就可以按照信息提示的方法遥控 TT 无人机：

起飞 键盘 -

降落 键盘 =

小贴士：什么是保持终端激活状态？当执行命令之后，不要将光标（鼠标）移动到其他窗口，此时光标所在的终端就是激活状态，在这个状态中的终端可以采集键盘的输入，用来控制机器人。

本例中的话题关系图：我们可以看出，在启动键盘遥控节点后，键盘遥控与 TT 无人机驱动通过/rmtt\_01/cmd\_vel 联系了起来，这也是键盘遥控可以控制 TT 无人机运动的机制。

```
tianbot@ros2go:~$ rostopic info /rmtt_01/cmd_vel
Type: geometry_msgs/Twist

Publishers:
* /rmtt_01/rmtt_teleop_key (http://ros2go:45851/)

Subscribers:
* /rmtt_01/rmtt_driver (http://ros2go:36255/)

tianbot@ros2go:~$
```

以可视化的方式查看话题关系图：（在新的终端中输入：rqt\_graph）。



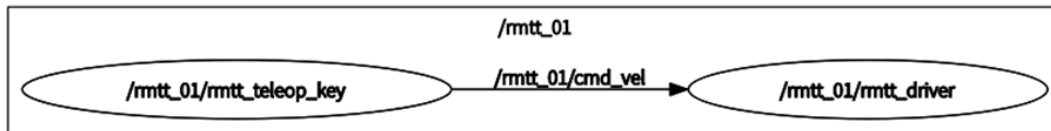


图 9：键盘遥控 TT 无人机节点话题关系图

本文所述的键盘遥控案例完整示例截图（TT 无人机 ROS 驱动、ROS 键盘遥控、rqt 关系图、话题列表）：

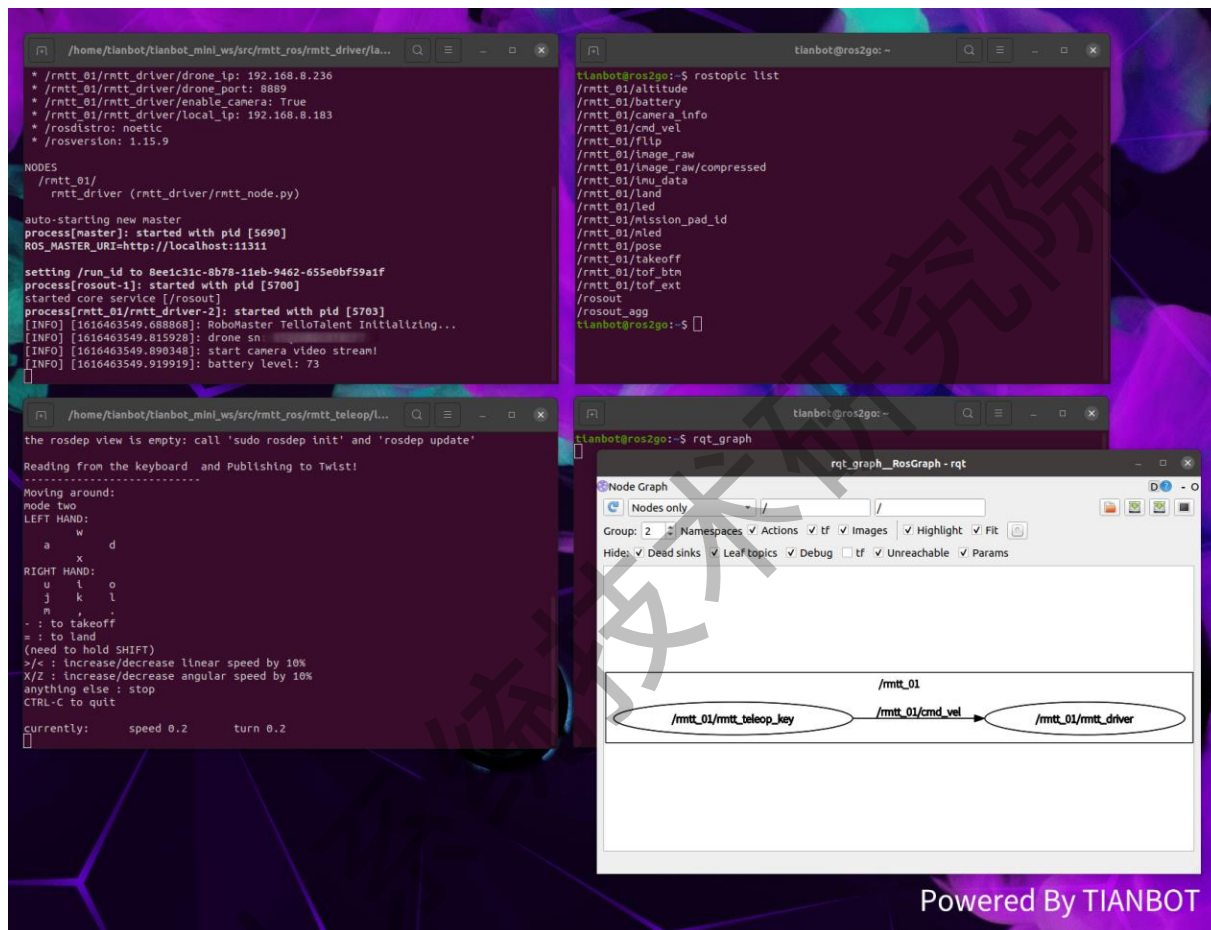


图 10：键盘遥控案例全部终端

异常处理（失控、无法降落）

关闭 `bringup.launch` 所在终端、30s 后 TT 无人机会自动降落。

。