机载算力盒子使用说明 v3.0

一、 设备连接

如图所示，debug 口连接 U 转串口模块，其中，黑色线为地线，白色为发送，黄色为接收。

注：上电后，如果终端中没有看到数据，可以将串口收发交换一下。

二、 系统登录

1、 串口连接算力盒子，可以在串口终端中看到如图信息。

```
[    8.169306] rc.local[2718]: success !!!
[    8.859447] ^^========, wl_setup_wiphy(11266), wdev->wiphy->interface_modes = 000007
[    8.890604] ANDROID-MSG) wl_ext_iapsta_attach_netdev: ifidx=0, bssidx=0
[    9.004523] ANDROID-MSG) wl_ext_iapsta_attach_netdev: ifidx=0, bssidx=0

Ubuntu 18.04.6 LTS myir ttyS0

myir login: root
Password:

Login incorrect
myir login: root
Password:
Last login: Thu Mar  2 20:58:21 CST 2023 on ttyS0
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.9.170 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
-bash: /usr/local/share/dji_sdk/setup.bash: No such file or directory
root@myir:~#
```

2、 登录系统

用户名：root

密码：123456

2、连接 wifi

nmcli device wifi connect "wifi 名称" password "密码" ifname wlan0

三、 程序烧录

1、 下载镜像文件

链接：

https://pan.baidu.com/s/12JC7kXTeOmcK_diTNzh87A

提取码：k3l9

2、 烧录

A、 SD 卡插入算力盒子

B、 格式化：mkfs.ext4 /dev/ mmcblk1

C、 拔出 SD 卡，拷贝镜像到 SD 卡中

D、 SD 卡插入算力盒子

E、 重新启动

F、 挂载：mount /dev/mmcblk1 /mnt/

```
root@myir:~# mount /dev/mmcblk
mmcblk0       mmcblk0p1     mmcblk0p4     mmcblk0p7     mmcblk1
mmcblk0boot0  mmcblk0p2     mmcblk0p5     mmcblk0p8
mmcblk0boot1  mmcblk0p3     mmcblk0p6     mmcblk0rpmb
root@myir:~# mount /dev/mmcblk1 /mnt/
root@myir:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p4  5.9G  5.4G  423M  93% /
devtmpfs        986M     0  986M   0% /dev
tmpfs           996M     0  996M   0% /dev/shm
tmpfs           996M  868K  995M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           996M     0  996M   0% /sys/fs/cgroup
tmpfs           200M     0  200M   0% /run/user/0
/dev/mmcblk1     15G  7.4G  6.5G  54% /mnt
root@myir:~#
```

G、 烧录

指令：dd of=/dev/mmcblk0 if=/mnt/backup.img bs=1MiB status=progress

```
root@myir:~#
root@myir:~#
root@myir:~#
root@myir:~# dd of=/dev/mmcblk0 if=/mnt/backup.img bs=1MiB status=progress
1429209088 bytes (1.4 GB, 1.3 GiB) copied, 40 s, 35.7 MB/s
```

H、 烧录完成

```
root@myir:~#
root@myir:~#
root@myir:~# dd of=/dev/mmcblk0 if=/mnt/backup.img bs=1MiB status=progress
7812939776 bytes (7.8 GB, 7.3 GiB) copied, 248 s, 31.5 MB/s
7456+0 records in
7456+0 records out
7818182656 bytes (7.8 GB, 7.3 GiB) copied, 248.157 s, 31.5 MB/s
root@myir:~#
```

I、 烧录完成，重启

四、 Psdk 程序运行

1、cd /root/tta_ros

2、执行命令：./dji_sdk_demo_linux_cxx

执行结果如图，证明 psdk 启动成功

```
root@myir:~/tta_ros# sudo ./dji_sdk_demo_linux_cxx
[0.006][core]-[Info]-[DjiCore_Init:96) Payload SDK Version : V3.4.0-beta.0-build.1743
[0.139][adapter]-[Info]-[DjiAccessAdapter_Init:180) Identify aircraft series is Mavic 3 Series
[0.139][adapter]-[Info]-[DjiAccessAdapter_Init:198) Identify mount position type is Extension Port Type
[2.157][adapter]-[Info]-[DjiPayloadNegotiate_Init:185) Waiting payload negotiate finish.
[3.157][adapter]-[Info]-[DjiPayloadNegotiate_Init:189) No need wait negotiate finished
[5.208][adapter]-[Info]-[DjiPayloadNegotiate_Init:185) Waiting payload negotiate finish.
[6.208][adapter]-[Info]-[DjiPayloadNegotiate_Init:189) No need wait negotiate finished
[6.209][core]-[Info]-[DjiIdentityVerify_UpdatePolicy:445) Updating dji sdk policy file...
[7.209][core]-[Info]-[DjiIdentityVerify_UpdatePolicy:448) Update dji sdk policy file successfully
[7.239][core]-[Info]-[DjiCore_Init:164) Identify AircraftType = Mavic 3 Enterprise, MountPosition = Extension Port, SdkAdapterType = None
[7.244][core]-[Info]-[DjiCore_ApplicationStart:231) Start dji sdk application
[7.244][user]-[Info]-[DjiUser_ApplicationStart:260) Application start.
[10.276][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:626) receive quaternion data.
[10.276][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:627) timestamp: millisecond 10273 microsecond 10273000.
[10.277][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:629) quaternion: 0.993765 0.000452 0.037764 -0.104901.
[10.277][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:632) euler angles: pitch = 4.31 roll = -0.40 yaw = -12.07.

[10.734][user]-[Info]-[Dji_FcSubscriptionStartService:305) dji system module init success!!
[10.734][flight]-[Info]-[DjiFlightController_RegisterLinkerObj_M3:111) Init mavic3 enterprise series flight controller linker successfully.
[11.279][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:626) receive quaternion data.
[11.279][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:627) timestamp: millisecond 11279 microsecond 11279000.
[11.279][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:629) quaternion: 0.993791 0.000412 0.037755 -0.104659.
[11.279][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:632) euler angles: pitch = 4.31 roll = -0.41 yaw = -12.04.

[12.285][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:626) receive quaternion data.
[12.285][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:627) timestamp: millisecond 12285 microsecond 12285000.
[12.285][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:629) quaternion: 0.993790 0.000587 0.037832 -0.104641.
[12.285][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:632) euler angles: pitch = 4.32 roll = -0.39 yaw = -12.04.

[13.296][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:626) receive quaternion data.
[13.296][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:627) timestamp: millisecond 13296 microsecond 13296000.
[13.296][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:629) quaternion: 0.993779 0.000621 0.037768 -0.104765.
[13.296][user]-[Info]-[Dji_FcSubscriptionReceiveQuaternionCallback:632) euler angles: pitch = 4.31 roll = -0.38 yaw = -12.05.

^C
```

五、 ROS 节点程序运行

1、 ros 节点程序目录:

/root/catkin_ws/

2、 source ./devel/setup.bash

3、 启动 ROS 环境

```
root@myir:~/catkin_ws#
root@myir:~/catkin_ws#
root@myir:~/catkin_ws# roscore
... logging to /root/.ros/log/7962da78-b8fb-11ed-ac26-babeb8e62f06/roslaunch-myir-7595.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://myir:38607/
ros_comm version 1.14.13


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.13

NODES

auto-starting new master
process[master]: started with pid [8077]
ROS_MASTER_URI=http://myir:11311/

setting /run_id to 7962da78-b8fb-11ed-ac26-babeb8e62f06
process[rosout-1]: started with pid [8242]
started core service [/rosout]
```

4、 换另一个中断，执行 ROS 节点程序

飞机数据节点:

rosrun ttauav_node uavdata

```
root@myir:~/catkin_ws# rosrun ttauav_node uavdata
new ------------------------------------DataPortKeeperThread
get msgid 8202 form FC
get msgid 8202 form FC
get msgid 8202 form FC
get msgid 8194 form FC
get msgid 8201 form FC
get msgid 8202 form FC
get msgid 8202 form FC
get msgid 8202 form FC
get msgid 8201 form FC
get msgid 8202 form FC
get msgid 8202 form FC
get msgid 8202 form FC
```

如图，表示正确接收到 psdk 数据。

无人机上报的数据如下图:

```
# positon in WGS84
float64 latit
float64 longi
float32 altit

float32 velN
float32 velE
float32 velD

float32 atti_pitch
float32 atti_roll
float32 atti_yaw

float32 gyro_pitch
float32 gyro_roll
float32 gyro_yaw

float32 accN
float32 accE
float32 accD

float32[] quat
```

六、 ROS 测试节点

监听节点:

rosrun ttauav_node listener

```
root@myir:~/catkin_ws/src/raspicam_node/src# rosrun ttauav_node listener
[ INFO] [1677762201.711624581]: I heard:[40.195992]
[ INFO] [1677762201.722732833]: I heard:[116.168511]
[ INFO] [1677762201.723000333]: I heard:[0.995497]
[ INFO] [1677762201.723063541]: I heard:[-0.000491]
[ INFO] [1677762201.723117916]: I heard:[0.037417]
[ INFO] [1677762201.723168874]: I heard:[-0.087099]
[ INFO] [1677762201.810789675]: I heard:[40.195992]
[ INFO] [1677762201.810960216]: I heard:[116.168511]
[ INFO] [1677762201.811054966]: I heard:[0.995495]
[ INFO] [1677762201.811141258]: I heard:[-0.000473]
```

如图所示，即为数据监听成功。

七、 飞行控制

启动飞行控制服务:

```
root@myir:~/catkin_ws# source ./devel/setup.bash
root@myir:~/catkin_ws# rosrun ttauav_node service
new ----------------------------------DataPortKeeperThread
[ INFO] [1677762398.825912762]: Start takeoffOrLanding server
[ INFO] [1677762398.849105473]: Start flightByOffset server
[ INFO] [1677762398.855937765]: Start flightByVel server
[ INFO] [1677762398.862264182]: Start gimbalControl server
```

飞行控制参数说明:

1、 起飞降落

```
# 1-takeoff  2-landing
int8 takeoffOrLanding

---
```

takeoffOrLanding 为 1 时，执行起飞，2 时，执行降落

2、 飞行控制，速度控

```
float32 vel_n
float32 vel_e
float32 vel_d

float32 targetYaw

float32 fly_time

---
```

Vel_n:前向速度

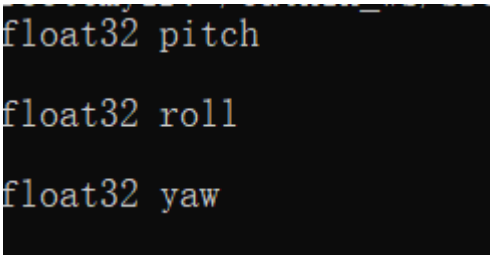
Vel_e:右向速度

Vel_d:向上速度

TargetYaw：飞行航向

Fly_time：以设置速度飞行的持续时间

3、 云台控制



Pitch：云台俯仰角度

Roll：云台横滚角度

Yaw:云台航向角度

注：所有角度值为相对角度。

4、 视频流地址

rtsp://本机 ip 地址:8554/live

如：rtsp://127.0.0.1:8554/live

使用 vlc 等播放器可以直接拉取播放。

八、 所有 ROS 程序源码，均在 /root/catkin_ws/src/raspicam_node 下。

```
tta@ubuntu:~/workspace/catkin_ws/src/raspicam_node$ tree
.
├── CHANGELOG.rst
├── CMakeLists.txt
├── include
├── launch
│   └── uavdata.launch
├── LICENSE
├── msg
│   └── uavdata.msg
├── package.xml
├── pipline.py
├── README.md
├── reconfigure_raspicam_node.png
├── src
│   ├── flightService.cpp
│   ├── listener.cpp
│   ├── proxy_src
│   │   ├── Buffer.cpp
│   │   ├── Buffer.h
│   │   ├── BufferItem.cpp
│   │   ├── BufferItem.h
│   │   ├── DataPort.cpp
│   │   ├── DataPort.h
│   │   ├── DataPortKeeperThread.cpp
│   │   ├── DataPortKeeperThread.h
│   │   ├── DataPortReadThread.cpp
│   │   ├── DataPortReadThread.h
│   │   ├── DataPortSendThread.cpp
│   │   ├── DataPortSendThread.h
│   │   ├── Lock.cpp
│   │   ├── Lock.h
│   │   ├── Log.cpp
│   │   ├── Log.h
│   │   ├── Port.cpp
│   │   ├── Port.h
│   │   ├── SerialDataPort.cpp
│   │   ├── SerialDataPort.h
│   │   ├── SerialPort.cpp
│   │   ├── SerialPort.h
│   │   ├── SocketUtils.cpp
│   │   ├── SocketUtils.h
│   │   ├── TcpDataPort.cpp
│   │   ├── TcpDataPort.h
│   │   ├── TcpPort.cpp
│   │   ├── TcpPort.h
│   │   ├── Thread.cpp
│   │   ├── Thread.h
│   │   ├── UdpDataPort.cpp
│   │   ├── UdpDataPort.h
│   │   ├── ttauav_node.cpp
│   │   ├── uavData.cpp
│   │   ├── uavData.h
│   │   ├── uavpos_node.cpp
│   │   └── utils
│   │       ├── cJSON.c
│   │       ├── cJSON.h
│   │       ├── CJsonObject.cpp
│   │       ├── CJsonObject.hpp
│   │       ├── CryptUtils.cpp
│   │       ├── CryptUtils.h
│   │       ├── diskInfo.cpp
│   │       ├── diskInfo.h
│   │       ├── FileUtils.cpp
│   │       ├── FileUtils.h
│   │       ├── misc.cpp
│   │       ├── misc.h
│   │       ├── public_utils.cpp
│   │       ├── public_utils.h
│   │       ├── ttalinkUtils.cpp
│   │       └── ttalinkUtils.h
├── srv
│   ├── flight.srv
│   └── takeoffOrLanding.srv
└── tools
```