FMT Mavros使用指南

FMT目前已经支持mavros的大部分消息功能，基于ROS Melodic进行测试通过。下面介绍FMT Mavros功能的基本使用方法。以下都以ICF5为例，其它硬件平台可类似配置。

机载通信接口配置

首先需要配置飞控上的sysconfig.toml文件来配置飞控跟机载电脑通信的接口。需配置mavproxy，如下所示。这里定义了三个设备，其中两个通道0（chan=0）的设备用于跟地面站连接，一个通道1（chan=1）的设备用于跟机载电脑连接。所以这里的配置是使用serial2即飞控上的TELEM2口来跟机载电脑通信。

```
# Mavproxy Device Configuration
[mavproxy]
    # ground control station channel devices
    [[mavproxy.devices]]
    chan = 0
    type = "serial"
    name = "serial1"
    baudrate = 57600

    [[mavproxy.devices]]
    chan = 0
    type = "usb"
    name = "usbd0"
    auto-switch = true

    # onboard computer channel devices
    [[mavproxy.devices]]
    chan = 1
    type = "serial"
    name = "serial2"
    baudrate = 115200
```

如果要使用USB来跟机载电脑通信，则可以如下配置

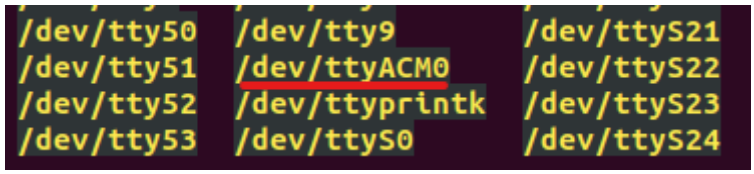
```
# Mavproxy Device Configuration
[mavproxy]
    # ground control station channel devices
    [[mavproxy.devices]]
    chan = 0
    type = "serial"
    name = "serial1"
    baudrate = 57600

    # onboard computer channel devices
    [[mavproxy.devices]]
    chan = 1
    type = "usb"
```

```
    name = "usbd0"
```

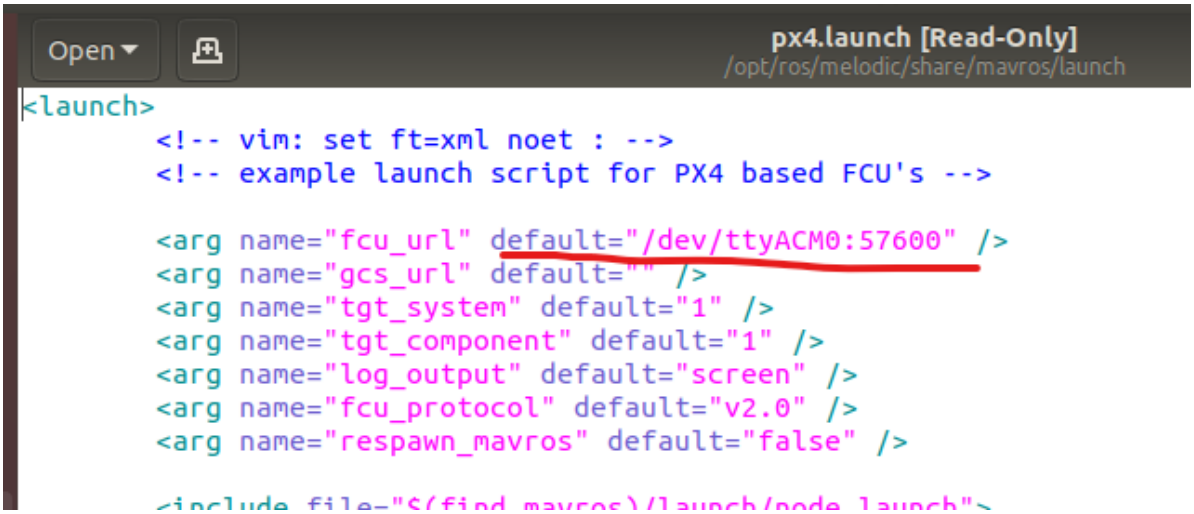
> 修改sysconfig.toml文件后，需将其上传到飞控的板载文件系统的/sys目录下并重启方可生效。

建立Mavros和FMT的连接

首先将飞控的串口或者USB跟机载电脑连接起来。假设机载电脑安装的是Ubuntu，那么在Ubuntu的/dev目录下应该会出现一个对应的设备，如/dev/ttyACM0。

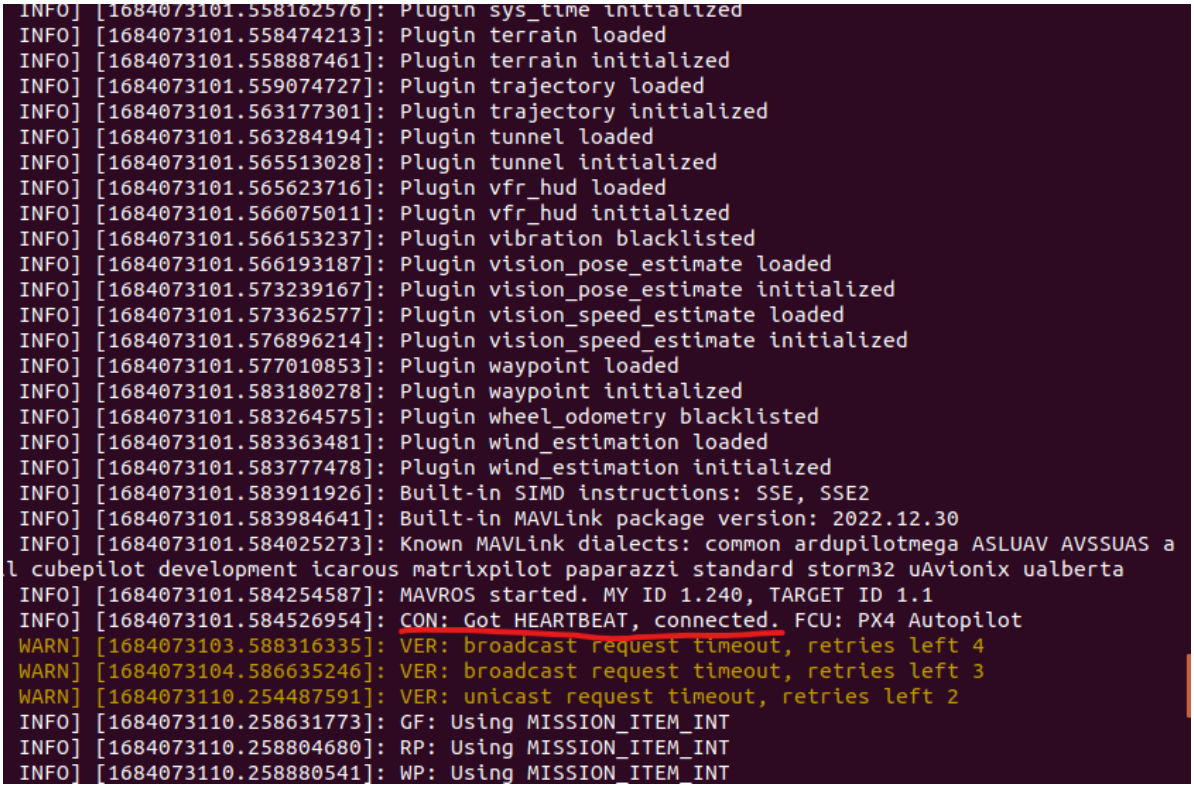


修改px4.launch文件，将设备端口和波特率改成跟FMT的一致。



运行mavros:

```
roslaunch mavros px4.launch
```



机载电脑控制无人机

目前FMT已经支持了如下的机载电脑的mavlink消息。如果需要其它消息支持，可以告知我们进行评估，或者也可以自行修改mavobc.c的代码添加对应mavlink消息的处理即可。

- 模式设置
- 上锁/解锁（未测试）
- 起飞/降落/返航/Hold/Pause/Continue等指令支持
- MAVLINK_MSG_ID_SET_ATTITUDE_TARGET（支持坐标：FRAME_BODY_FRD）
- MAVLINK_MSG_ID_SET_POSITION_TARGET_LOCAL_NED（支持坐标：MAV_FRAME_LOCAL_NED，MAV_FRAME_LOCAL_FRD，MAV_FRAME_BODY_FRD）
- MAVLINK_MSG_ID_SET_POSITION_TARGET_GLOBAL_INT（支持坐标：MAV_FRAME_GLOBAL_INT）

以下是一些测试的ros指令，也可以使用c++或者python调用ros的接口，效果一样。

```
# 设置模式。注意，通过机载电脑设置模式需要遥控器关闭状态下才能够进行设置，因为遥控模式的优先级高于地面站和机载电脑。

# 设置Position
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'POSCTL'"

# 设置Mission
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'AUTO.MISSION'"

# 设置Offboard
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'OFFBOARD'"
```

```
# 发送指令

# 起飞
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'AUTO.TAKEOFF'"

# 降落
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'AUTO.LAND'"

# 返航
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'AUTO.RTL'"

# Hold
rosservice call /mavros/set_mode "base_mode: 0
custom_mode: 'AUTO.LOITER'"
```

```
# 发送位置控制指令，Offboard模式下有效。若进入Offboard模式后无控制信号指令发给飞控，即auto_cmd的消息的发布频率为0，飞机将进入Hold悬停状态。

rostopic pub /mavros/setpoint_raw/local mavros_msgs/PositionTarget "header:
```

```
  seq: 0
  stamp:
    secs: 0
    nsecs: 0
  frame_id: ''
coordinate_frame: 1
type_mask: 2560
position:
  x: 100.0
  y: 50.0
  z: 10.0
velocity:
  x: 0.0
  y: 0.0
  z: 0.0
acceleration_or_force:
  x: 0.0
  y: 0.0
  z: 0.0
yaw: 0
yaw_rate: 0.0" -r 10
```

> 注意ROS的坐标系跟飞控的坐标系不一样，mavros会进行转换
> 当发送这条消息后，飞控端会收到auto_cmd的消息，可以在飞控端输入mcn echo auto_cmd来打印输出：

```
msh />mcn list
Topic                #SUB    Freq(Hz)    Echo    Suspend
------------------ ------ ---------- ------ ---------
sensor_imu0_0          0        0.0     true    false
sensor_imu0            1        0.0     true    false
sensor_mag0_0          0        0.0     true    false
sensor_mag0            1        0.0     true    false
sensor_baro            1        0.0     true    false
sensor_gps             1        0.0     true    false
sensor_airspeed        1        0.0     true    false
mav_ext_state          0        0.0     false   false
ins_output             3      500.0     true    false
fms_output             4      250.0     true    false
control_output         2      500.0     true    false
pilot_cmd              3        0.0     true    false
rc_channels            0        0.0     true    false
rc_trim_channels       1        0.0     true    false
gcs_cmd                2        1.0     true    false
auto_cmd               1        7.6     true    false
mission_data           2        0.0     true    false
bat_status             0        2.0     true    false
msh />mcn echo auto_cmd
timestamp:1973057 frame:0
psi: 1.57
x: 50.00
y: 100.00
z: -10.00
u: 0.00
v: 0.00
```

```
w: -0.00
ax: 0.00
ay: 0.00
az: -0.00
----------------------------------------
timestamp:1973551 frame:0
psi: 1.57
x: 50.00
y: 100.00
z: -10.00
u: 0.00
v: 0.00
w: -0.00
ax: 0.00
ay: 0.00
az: -0.00
----------------------------------------
```

```
# 发送姿态控制指令，Offboard模式下有效。若进入Offboard模式后无控制信号指令发给飞控，即auto_cmd的消息的发布频率为0，飞机将进入Hold悬停状态。

rostopic pub /mavros/setpoint_raw/attitude mavros_msgs/AttitudeTarget "header:
  seq: 0
  stamp: {secs: 0, nsecs: 0}
  frame_id: ''
type_mask: 7
orientation:
  x: 0.047
  y: 0.113
  z: 0.373
  w: 0.920
body_rate:
  x: 0.0
  y: 0.0
  z: 0.0
thrust: 0.6
" -r 10
```

同样可以在飞控端输入mcn echo auto_cmd来打印收到的消息：

```
msh />mcn echo auto_cmd
timestamp:2051378 frame:2
phi: 0.17
theta: -0.17
psi: 0.79
throttle: 1600
----------------------------------------
timestamp:2051877 frame:2
phi: 0.17
theta: -0.17
psi: 0.79
throttle: 1600
----------------------------------------
timestamp:2052377 frame:2
```

```
phi: 0.17
theta: -0.17
psi: 0.79
throttle: 1600
----------------------------------------
timestamp:2052878 frame:2
phi: 0.17
theta: -0.17
psi: 0.79
throttle: 1600
----------------------------------------
```

机载电脑接收飞控发来的消息

FMT默认只会给机载电脑发送Heartbeat心跳包（1Hz）。机载电脑在收到飞控发来的心跳包后，需要发送MAVLINK_MSG_ID_REQUEST_DATA_STREAM消息给飞控来配置需要接收的消息。

目前FMT支持的机载电脑消息如下所示。如果需要其它消息支持，可以告知我们进行评估，或者也可以自行修改mavobc.c的代码添加对应mavlink消息的发送即可。

```
static msg_pack_cb_table mav_msg_cb_table[] = {
    { MAVLINK_MSG_ID_HEARTBEAT, mavlink_msg_heartbeat_pack_func },
    { MAVLINK_MSG_ID_HIGHRES_IMU, mavlink_msg_highres_imu_pack_func },
    { MAVLINK_MSG_ID_LOCAL_POSITION_NED, mavlink_msg_local_position_ned_pack_func },
    { MAVLINK_MSG_ID_ATTITUDE, mavlink_msg_attitude_pack_func },
    { MAVLINK_MSG_ID_ALTITUDE, mavlink_msg_altitude_pack_func },
    { MAVLINK_MSG_ID_DISTANCE_SENSOR, mavlink_msg_distance_sensor_pack_func },
    { MAVLINK_MSG_ID_EXTENDED_SYS_STATE, mavlink_msg_extended_sys_state_pack_func },
    { MAVLINK_MSG_ID_GPS_GLOBAL_ORIGIN, mavlink_msg_gps_global_origin_pack_func },
    { MAVLINK_MSG_ID_HOME_POSITION, mavlink_msg_home_position_pack_func },
    { MAVLINK_MSG_ID_RC_CHANNELS, mavlink_msg_rc_channels_pack_func },
    { MAVLINK_MSG_ID_SYSTEM_TIME, mavlink_msg_system_time_pack_func },
};
```

如下是一个设置FMT飞控发送消息的频率的ROS代码：

```
#include <ros/ros.h>
#include <mavros_msgs/StreamRate.h>

int main(int argc, char **argv) {
    ros::init(argc, argv, "set_stream_rate_example");
    ros::NodeHandle nh;

    // 创建一个 /mavros/set_stream_rate 服务的客户端
    ros::ServiceClient stream_rate_client =
nh.serviceClient<mavros_msgs::StreamRate>("/mavros/set_stream_rate");

    // 准备要更改频率的消息 ID 和频率
    int message_id = 105; // MAVLINK_MSG_ID_HIGHRES_IMU
    int message_rate = 100; // 将频率更改为 100Hz

    // 创建一个服务请求对象
    mavros_msgs::StreamRate srv;
    srv.request.stream_id = message_id;
    srv.request.message_rate = message_rate;
    srv.request.on_off = true; // 启用消息发送
```

```
    // 调用服务
    if (stream_rate_client.call(srv)) {
        ROS_INFO("Stream rate set successfully!");
    } else {
        ROS_ERROR("Failed to set stream rate.");
    }

    return 0;
}
```