

本着想在PX4基础上加些什么东西的我又开始折腾了，先尝试用串口加传感器通过QGC查看，要是能在原固件上加内容就棒哉了。先立Flag

## 自定义uORB消息

ca\_trajectory.msg

```
uint64 time_start_usec
uint64 time_stop_usec
uint32 coefficients
uint16 seq_id

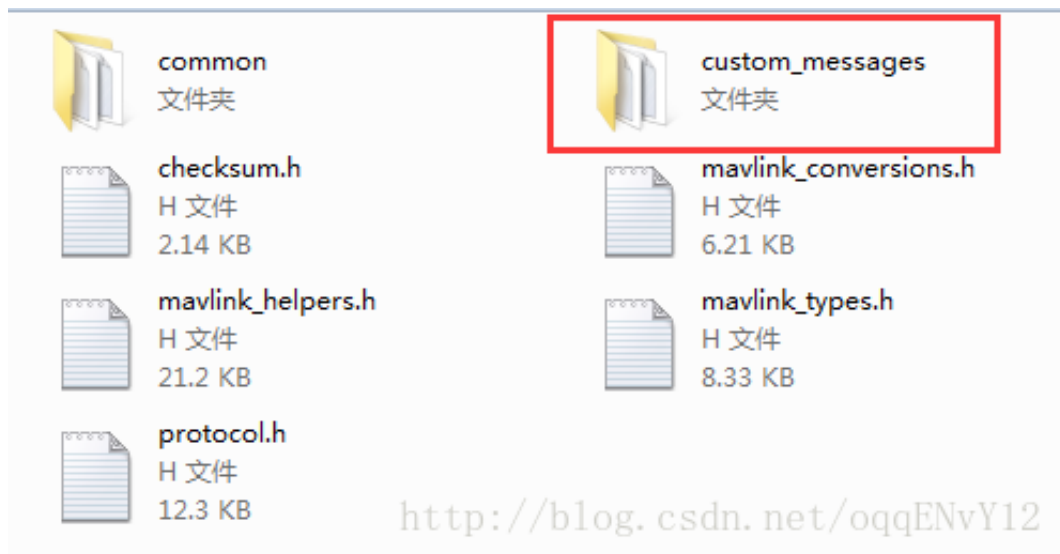
#TOPICS ca_trajectory
```

## 自定义MAVLink消息

custom\_messages.xml

```
<?xml version="1.0"?>
<mavlink>
  <include>common.xml</include>
  <!-- NOTE: If the included file already contains a version tag, remove the version
tag here, else uncomment to enable. -->
  <!--<version>3</version>-->
  <enums>
  </enums>
  <messages>
    <message id="166" name="CA_TRAJECTORY">
      <description>This message encodes all of the raw rudder sensor data from the
USV.</description>
      <field type="uint64_t" name="timestamp">Timestamp in milliseconds since
system boot</field>
      <field type="uint64_t" name="time_start_usec">start time, unit usec.</field>
      <field type="uint64_t" name="time_stop_usec">stop time, unit usec.</field>
      <field type="uint32_t" name="coefficients">as it says.</field>
      <field type="uint16_t" name="seq_id">can not cheat any more.</field>
    </message>
  </messages>
</mavlink>
```

使用 `python -m mavgenerate` 打开mavlink消息生成器导入上面的xml文件，生成如下文件：



将生成的custom\_messages文件夹拖到Firmware/mavlink/include/mavlink/v1.0目录下

## 发送自定义MAVLink消息

添加 `mavlink` 的头文件和uorb消息到[mavlink\\_messages.cpp](#)

```
#include <uORB/topics/ca_trajectory.h>
#include <v1.0/custom_messages/mavlink_msg_ca_trajectory.h>
```

在[mavlink\\_messages.cpp](#)中创建一个新的类

```
class MavlinkStreamCaTrajectory : public MavlinkStream
{
public:
    ...
private:
    ...
protected:
    ...
    void send(const hrt_abstime t)
    {
        ...
        mavlink_msg_ca_trajectory_send_struct(_mavlink->get_channel(), &msg);
    }
};
```

附加流类 `streams_list` 的到[mavlink\\_messages.cpp](#)底部

```
StreamListItem *streams_list[] = {  
    ...  
    new StreamListItem(&MavlinkStreamCaTrajectory::new_instance,  
    &MavlinkStreamCaTrajectory::get_name_static, &MavlinkStreamCaTrajectory::get_id_static),  
    nullptr  
};
```

最后在[mavlink main.cpp](#)加入自定义的消息以及期望的更新频率

```
configure_stream("CA_TRAJECTORY", 100.0f);
```

## 接收自定义MAVLink消息

在[mavlink receiver.h](#)中增加一个用来处理接收信息得函数

```
#include <uORB/topics/ca_trajectory.h>  
#include <v1.0/custom_messages/mavlink_msg_ca_trajectory.h>
```

在 [mavlink receiver.h](#)中增加一个处理类 `MavlinkReceiver` 中的输入mavlink消息的函数

```
void handle_message_ca_trajectory_msg(mavlink_message_t *msg);
```

在 [mavlink receiver.h](#)中加入一个类 `MavlinkReceiver` 中的uORB消息发布者

```
orb_advert_t _ca_traj_msg_pub;
```

在[mavlink receiver.cpp](#)中实现 `handle_message_ca_trajectory_msg` 功能

```

void
MavlinkReceiver::handle_message_ca_trajectory_msg(mavlink_message_t *msg)
{
    mavlink_ca_trajectory_t traj;
    mavlink_msg_ca_trajectory_decode(msg, &traj);

    struct ca_trajectory_s f;
    memset(&f, 0, sizeof(f));

    f.timestamp = hrt_absolute_time();
    f.seq_id = traj.seq_id;
    f.time_start_usec = traj.time_start_usec;
    f.time_stop_usec = traj.time_stop_usec;
    f.coefficients = traj.coefficients;

    if (_ca_traj_msg_pub == nullptr) {
        _ca_traj_msg_pub = orb_advertise(ORB_ID(ca_trajectory), &f);
    } else {
        orb_publish(ORB_ID(ca_trajectory), _ca_traj_msg_pub, &f);
    }
}

```

最后确定函数在[MavlinkReceiver::handle\\_message\(\)](#)中被调用

```

# 在mavlink_receiver.cpp文件中添加
MavlinkReceiver::handle_message(mavlink_message_t *msg)
{
    switch (msg->msgid) {
        ...
    case MAVLINK_MSG_ID_CA_TRAJECTORY:
        handle_message_ca_trajectory_msg(msg);
        break;
        ...
    }
}

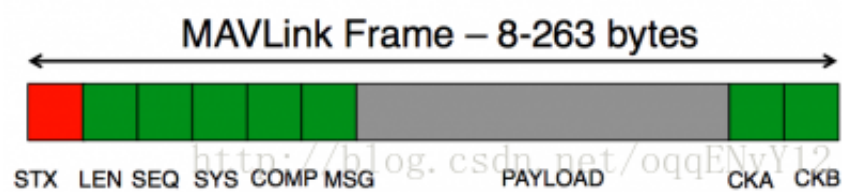
```

## 消息查看

将SD卡插入飞控连接电脑，然后会发现串口调试助手不停地吐MAVLink消息，如下所示（选择Hex方式接收，否则看到的都是乱码）：

FE	3E	F2	01	01	69	E8	81	0B	0B	00	00	00	00	62	D9	68	3E	37	38	EE	BE	99	F1	20	C1	7F		
4A	13	BC	EB	CA	03	3C	1E	E7	29	BA	B3	66	F9	3C	B2	C5	67	3E	AD	9B	19	3E	00	00	00	00		
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
8B	20	3D	17	ED	02	8D	4A	C6	B3	BF	12	0F	89	3B	D8	91	44	BA	D8	8F	E0	3A	B0	B4	FE	15		
F4	01	11	24	B2	60	0B	0B	84	03	84	03	84	03	84	03	00	00	00	00	00	00	00	00	01	FF	E0		
FE	20	F3	01	01	8D	FD	A7	0B	0B	00	00	00	00	08	70	E3	C1	00	00	C0	7F	90	0A	AD	3F	00		
00	C0	7F	00	00	C0	7F	00	00	C0	7F	96	92	FE	14	F6	01	01	4A	00	00	00	00	00	00	00	00		
08	70	E3	C1	00	00	00	80	17	01	00	00	87	F4	FE	28	F7	01	01	E7	2E	A3	0B	0B	00	00	00		
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	C0	7F	00	00		
00	00	00	00	00	00	00	7E	AA	FE	1C	F8	01	01	1E	E5	D3	02	00	B3	83	20	3D	1D	E9	02	3D	5D	
C6	B3	BF	8E	D6	27	3B	F0	E1	B2	39	62	57	9A	3A	0D	F0	FE	1C	F9	01	01	20	E5	D3	02	00		
00	00	00	00	00	00	00	00	00	00	73	D0	AC	BF	A5	5A	11	BC	49	B9	7B	3C	86	5B	36	BE	39	CD	FE
29	FA	01	01	8C	CA	B8	0B	0B	00	00	00	00	00	4C	B7	17	BD	0F	46	09	BD	2C	09	86	3F	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
08	D0	0B	0B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
3B	8A	90	D4	3A	A6	4C	FF	3C	19	5C	6C	3E	5C	A4	17	1E	00	30	00	00	00	00	00	00	00	00		
E3	C1	29	5C	01	42	FF	01	D7	FD	FE	1C	FC	01	01	1E	EF	D3	02	00	7C	84	20	3D	C7	E8	02		
3D	79	C6	B3	BF	F8	90	67	BA	D8	B8	40	3A	34	B5	04	BA	6F	7C	FE	1C	FD	01	01	1E	F7	D3		
02	00	BD	86	20	3D	3B	EA	02	3D	AB	C6	B3	BF	70	DC	7E	BA	78	58	78	3A	15	E8	CE	BA	B9		
10	FE	3E	FE	01	01	69	C8	2D	0C	0B	00	00	00	00	B6	28	27	3E	8F	56	A0	BE	A2	F5	20	C1		
A6	E2	CB	BB	0E	80	E8	3B	3E	CD	5C	3A	3E	B6	FA	3C	E6	10	6A	3E	19	83	15	3E	00	00	00		
00	00	00	00	00	00	00	1C	E2	C1	29	5C	01	42	FF	1B	30	F3	FE	1C	FF	01	01	1E	03	D4	02	00	
7A	7B	20	3D	35	E0	02	3D	8D	C6	B3	BF	C2	F8	32	BB	38	C5	2B	BA	16	31	E1	39	18	CE	FE		
1C	00	01	01	20	03	D4	02	00	00	00	00	00	00	00	00	00	00	CB	AB	AC	BF	9A	DF	10	BC	06	E3	
79	3C	47	8F	35	BE	71	3D	FE	29	01	01	01	8C	3E	30	0C	0B	00	00	00	00	00	96	41	27	BD	F9	
B7	08	BD	13	0C	86	3F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00	8E	9D	FE	1C	02	01	01	1E	0F	D4	02	00	4C	86	20	3D	8B	DF	02	3D	C4	C6	B3	BF	2D	4E		
7E	3B	EC	6A	B4	3A	FF	85	DC	B9	DD	DD	FE	15	03	01	01	24	83	18	0C	0B	84	03	84	03	84		
03	84	03	00	00	00	00	00	00	00	00	00	01	F4	54	FE	14	04	01	01	4A	00	00	00	00	00	00		
00	00	1C	E2	C1	00	00	00	80	17	01	00	00	61	04	FE	3E	05	01	01	69	E8	7B	0C	0B	00	00		
00	00	5D	16	39	3E	AC	73	0D	BF	EC	F9	20	C1	7A	7F	2D	3B	6E	73	00	3C	7A	E5	5F	39	A6		
4C	FF	3C	02	DF	66	3E	8F	11	17	3E	00	00	00	00	00	00	00	00	00	00	1C	E2	C1	29	5C	01	42	
FF	01	F7	D6	FE	1C	06	01	01	1E	18	D4	02	00	B7	85	20	3D	A6	DD	02	3D	0E	C7	B3	BF	FE		
E6	C8	3B	40	3A	DE	38	28	CA	4F	B9	B3	06	FE	2A	07	01	01	E6	29	9D	0C	0B	00	00	00	00		
00	00	00	00	4C	A8	48	39	06	B1	38	3C	47	1F	05	3B	00	00	00	00	00	00	00	00	00	00	00		
00	00	00	00	00	A5	02	87	C0	FE	20	08	01	01	F1	00	00	00	00	00	00	00	00	E8	2F	AC	30		
63	BE	1F	38	35	8A	9F	3A	00	00	00	00	00	00	00	00	00	00	00	00	99	B9	FE	1C	09	01	01		

可以很容易的解析出每一帧MAVLink消息的内容，依次按照下图所示的顺序

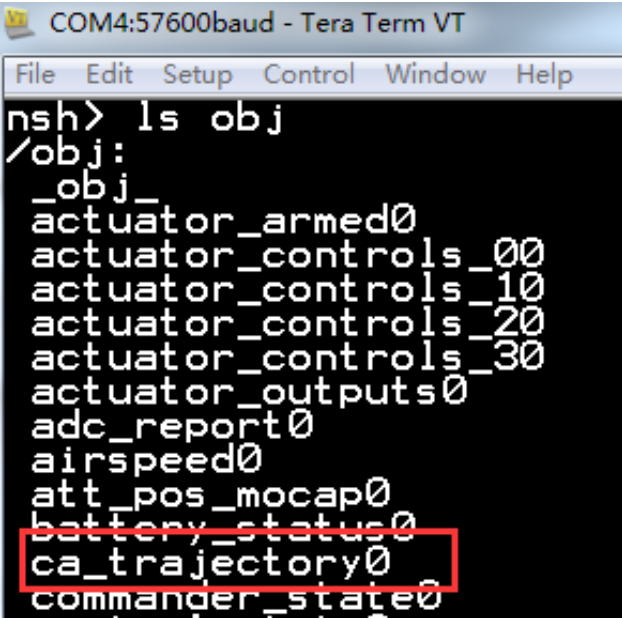


字节	内容	数值
0	起始标志位	0~255
1	载荷长度	0~255
2	包序号	0~255
3	SYSID	1~255
4	COMPID	0~255
5	MSGID	0~255
6~(n+6)	载荷	(0~255)字节
(n+7)~(n+8)	冗余校验	XXX

可以看到，并不是每一个MAVLink消息都是一直启动的，只有心跳包是时刻不断的发送消息的(1Hz)，其他的自启MAVLink消息规律暂时找不到，至少现在在串口读到的数据中还是找不到我定义的MSGID #166 = 0x6A。

要确认自定义的MAVLink真的存在，目前可以通过NSH查看

```
ls obj
```



与预想的直接可以在QGC地面站查看还有一定的差距。

大家有好办法的欢迎交流指导。

## 参考

1. [PX4中文维基之定义MAVlink消息](#)

2. [自定义uORB消息](#)
  3. [创建MAVLink消息](#)
  4. [生成MAVLink文件](#)
-