

各位无人机发烧友们：

你们好，最近在研究 Mavlink，发现国内没有中文资料可查，所幸翻译了一些，供大家交流使用。由于时间匆忙翻译得很粗糙，大家明白大意就好啦！

有意交流者可通过 [thunderleiming@126.com](mailto:thunderleiming@126.com) 与我联系，期待你的出现！

## mavlink 微型飞行器的通信协议

*mavlink 是一个非常轻量级，对微型飞行器数据头信息进行引导的库。*

它可以高速地传递 c 语言并发送这些数据包到地面控制站。它已被广泛的在 PIXHAWK，SLUGS 和 ArduPilotMega 平台上测试，并作为信息传递地主干网对有 MCU / IMU 以及 Linux 的进程和地面链路通信进行信息的传递。

## 整合教程

这些教程显示如何在您的自动驾驶仪整合 mavlink。

- **板载集成教程，使用内建的自动驾驶仪软件 *mavlink***
- **ROS（机器人操作系统）集成教程微型空中机器人中使用 *mavlink***
- ***mavlink* 地面控制集成教程适应 *QGroundControl*。**

***QGroundControl* 用户：虽然 *mavlink* 本身不依赖于它，但 *QGroundControl* 调整的意见和基础设置都依赖 *mavlink* 的语言。*QGroundControl* 还使用这个消息跟踪如果一个系统不论它是正常的还是断开连接的。因此，请确保每 60，30，10 或 1 秒（发送消息 1 Hz 的建议，但不要求）**

## Mavlink 编码和驱动

有一个“common message set”包含广泛使用的消息。如果你想使用自己的自定义消息，您可以使用与这些 mavlink 启动器如（C/ C + +， Python）或 QGroundControl。（下载 QGC 请访问 <http://qgroundcontrol.org/downloads>）  
托管 mavlink 二进制版本：

<https://github.com/mavlink/mavlink/downloads>

mavlink 消息定义和驱动代码托管：

- [https://github.com/ mavlink / mavlink /](https://github.com/mavlink/mavlink/)
- 以 mavlink 为 桥 的 ROS :  
[https://github.com/ mavlink / mavlink -ros](https://github.com/mavlink/mavlink-ros)
- MAVCONN 轻 型 空 中 飞 行 器  
<http://pixhawk.ethz.ch/software/middleware/start>
- oooArk / MAVSim <http://www.youtube.com/watch?v=-wQVrM5SL2o&fe>
- mavlink 的 Python 绑定
- MAVProxy (允许连接多个 UDP /串行链路, 包括 FlightGear 的) :

`git://git.samba.org/tridge/UAV/MAVProxy.git`

使用 mavlink 的项目 (使用请点击以下网站)

- ArduPilotMega <http://code.google.com/p/ardupilot-mega/>
- MatrixPilot UAV DevBoard <http://code.google.com/p/gentlenav/>
- PIXHAWK <http://pixhawk.ethz.ch/>
- ETH Flying Machine Arena  
[http://www.idsc.ethz.ch/Research\\_DAndrea/FMA](http://www.idsc.ethz.ch/Research_DAndrea/FMA)
- ETH SenseSoarSolar airplane project  
<http://www.sensesoar.ethz.ch/doku.php?id=news>
- ETH Skye Blimp Project <http://www.projectskeye.ch/>
- ArduCAM OSD <http://code.google.com/p/arducam-osd/>
- UC Santa Cruz SLUGS <http://slugsuav.soe.ucsc.edu/index.html>

## 牌照

mavlink 是 LGPL 的许可, 因此可以被用来作为一个库在闭源和开源应用程序的免版税。

## 常见问题 (FAQ)

从 0.9.0 升级到 1.0.0

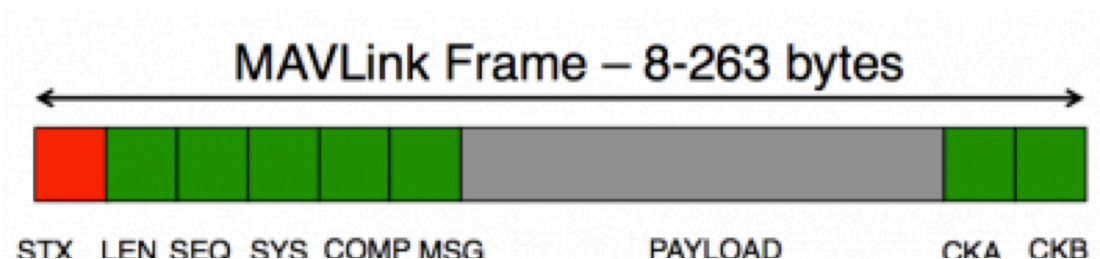
- **问：是升级容易做吗？** - 是的，v. 1.0.0 几乎是完全兼容的 API 与 v. 0.9.0。只有少数的消息已经改变，像消息协议解析(mavlink\_parse\_char)的所有功能完全兼容。
- **问：升级的主要好处是什么？** - 1 mavlink 现在使用 little-endian 的编码线，这使得很多在大多数平台上更有效。它仍然支持低端和高端平台。2. 现在检查每个消息包格式，所以如果两个通讯伙伴有不同的格式相同的消息，它会检测这种不匹配。3. 现在有较小的消息缓冲区的支持。4. 参数的协议，现在支持 32 个无符号和有符号整数和浮点值。5. mavlink 现在支持 IEEE 754 双精度浮点数。6. 所有邮件都自动生成的测试套件测试，使该协议甚至更安全。7. 一些消息已经清理，以更好地适应实际飞行使用。8. mavlink 现在默认 C / C++ 和 Python 的支持。
- **问：新版本将很快用于？** - 是的，QGroundControl 完全移植到和 ArduPilotMega 和 PIXHAWK 已经将他们的软件移植。不久，所有采用移植到新的版本，只是因为它是没有太多精力。

## 用户

- **问：有多少飞机确实 mavlink 支持** - 255 型飞机，从 1 到 255 不等。0 不是一个有效的飞机编号。
- **问：如何高效是 mavlink？** - 每包 8 个字节，包括开始的标志和丢包检测，mavlink 是一个非常有效的协议。
- **问：在哪里可以使用 mavlink？** - mavlink 已被证明是工作在多个微控制器和操作系统，包括 ARM7 的，ATMEGA，DSPIC，STM32 的和 Windows，LINUX 和 MacOS。
- **问：安全性如何？** mavlink - 两年多前已用于 mavlink 在 PIXHAWK MAV 的飞行项目，并依靠行之有效的 ITU X.25 的校验数据包损坏的检测。

## 包解剖

这是一个包的解剖。它的灵感来自于 CAN 和 SAE AS-4 标准。



字节的索引	内容	值	解释
0	包开始标志	V1.0: 0xFE 的 (V0.9: 将 0x55)	表示开始一个新的包。
1	有效载荷的长度	0 - 255	表示以下有效载荷的长度。
2	包序列	0 - 255	每个组件计数他的发送序列。允许检测丢包
3	系统 ID	1 - 255	发送系统的 ID。允许在同一网络上区分不同的飞行器。
4	组件的 ID	0 - 255	发送组件的 ID。允许在同一系统中区分不同的组件, 如 IMU 和自动驾驶仪。
5	消息 ID	0 - 255	消息 ID - ID 定义有效载荷 “是指” 应如何正确解码。
6 (N+6)	数据	(0 - 255) 字节	消息数据, 取决于消息的 ID。
(N +7) 至 (n +8)	校验和 (低字节, 高字节)	国际电联 X. 25/SAE AS-4 拼凑, 不含包开始的标志, 所以字节 1 .. (n+6) 注: 校验包括 MAVLINK_CRC_EXTRA (数计算从消息字段保护了不同版本的解码包相同的数据包, 但不同的变量)。	

- 校验和是相同的, 在 ITU X. 25 和 SAE AS-4 标准 (使用 CRC-16-CCITT 的), SAE 的 AS5669A 记录。请 `mavlink` 为一个文件的 C 执行它。源代码的链接为 **CHECKSUM**
- 确认无有效载荷包的最小数据包长度为 8 字节
- 最大包长度是充分有效载荷为 263 字节

#### 支持的数据类型

`mavlink` 支持固定大小的整数数据类型, IEEE 754 单精度浮点数, 这些数据类型的数组 (如 `char [10]`) 和特别的 `mavlink_version` 领域, 该协议将自动添加。这些类型可供选择:

- `char` - 字符/字符串

- `uint8_t` - 无符号 8 位
- `int8_t` - 有符号 8 位
- `uint16_t` - 无符号 16 位
- `int16_t` - 有符号 16 位
- `uint32_t` - 32 位无符号
- `int32_t` - 有符号 32 位
- `uint64_t` - 无符号 64 位
- `int64_t` - 有符号的 64 位
- `float` - IEEE 754 单精度浮点数
- `double` - IEEE 754 双精度浮点数
- `uint8_t_mavlink_version` - 无符号的 8 位字符自动补充到当前的 mavlink 版本-它不能被写入，只是读按照 8 位字符的方式读取。

## 性能

此协议由两大特性：传输速度快和安全性高。它可以检查邮件的内容，它也可以检测丢失的消息，但仍然只需要每个数据包 6 个字节的开销。

## 传输的例子

链接速度	硬件	更新率	有效载荷	float 值
115200 波特	2.4 千兆赫的 XBee	50 赫兹	224 字节	56
115200 波特	2.4 千兆赫的 XBee	100 赫兹	109 字节	27
57600	2.4 千兆赫的 XBee	100 赫兹	51 字节	12
9600 波特	XSC 900 的 XBee	50 赫兹	13 个字节	3
9600 波特	XSC 900 的 XBee	20 赫兹	42 字节	10

## 今后的工作/意见

- 变长数组
- 位域的支持（如包装成一个 `uint8_t` 8 个布尔值，但呼吁所有八个布尔提供的 C 函数。因此，用户不会转移/屏蔽自己摆弄。）
- 多变化的头函数，允许设置目标体系和目标组件（没有协议的变化，唯一的便利功能访问

## ROS（机器人操作系统）MAVLink 集成教程

安装 ROS 可登录 <http://ros.org>。在写作此教程时，当前版本是 ROS Fuerte（<http://ros.org/wiki/fuerte/Installation>）。

### MAVLink ROS 桥安装

ETH PIXHAWK 团队提供了一个 MAVLink ROS 桥为最常见的消息。桥之间通过 UART / 串行端口和 UDP 连接，传输到装有 MAVLink 和 ROS 实例的计算机上。

#### 快速简介

- `$ HOME` 就是你的主目录，如果你的用户名是 John，它保存在 Mac OS /Users/John 中。
- `/opt/ros/fuerte` 是 ROS 安装的地方。如果你的 ROS 安装到不同的位置，请自行调整。

#### 下载文件

ROS 工作区的改变：

```
roscd
```

或者如果您想有一个不同的目录（我们），将它添加到 `ROS_PACKAGE_PATH`。要做到这一点，编辑 `$ HOME / .bashrc` 中（在 Linux 中）或 `$ HOME / .bash_profile`（MAC）和采购 ROS 的安装脚本：

```
source /opt/ros/fuerte/setup.sh

export ROS_PACKAGE_PATH=$HOME/src:$ROS_PACKAGE_PATH
```

这一步后，关闭所有终端，并重新打开

现在去你想要安放 MAVLink 桥或者是 ROS 工作区的地方，目录：

```
roscd
```

或在不同的目录，如下：

```
CD ~/ src
```

下载的 MAVLink 桥：

```
git clone https://github.com/mavlink/mavlink_ros.git
```

更改到目录和编译

```
CD mavlink_ros
```

```
rosmake
```

## 航点协议

航点的协议说明了发送到航点是如何从 MAV 读取的。目标是确保发送方和接收方之间的一致状态。QGroundControl 有一个实施协议，每一个使用 MAVLINK 使用者可以与 QGroundControl 沟通和交流，并进行实时的更新。

## 通讯/主机

该协议包括不同的交易，每个交易或成功完成，或在以前的状态在由于接收端的航点列表方式不变而失败。只有当在通信双方之间没有其他的交易是活跃的时候，一个交易才可以启动一个特定的消息。这意味着启动一个交易双方通信双方必须是在空闲状态。如果一个事务开始的状态发生变化时，用这样一个简单的状态机的协议，可以轻松实现。

在每次发送的消息，发送组件启动一个计时器。如果没有收到任何答复，直到指定的时间已经过去，再次请求消息发送。重发时重试的次数，最后重试超时交易失败后，如果没有答案。重试机制意味着所有组件必须能够处理重复消息。

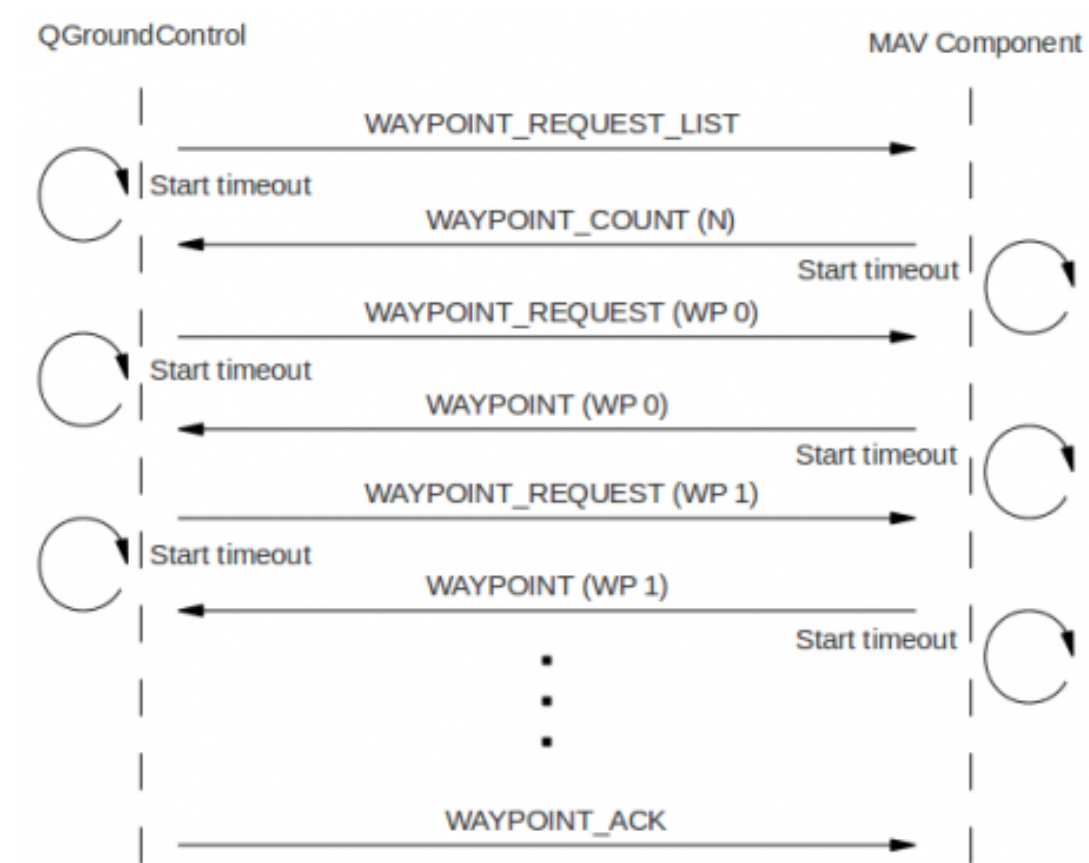
## 阅读 MAV 的航点列表

要检索的所有航点从组件列表 `WAYPOINT_REQUEST_LIST` 消息发送。有针对性的组件将响应，说明在其名单的航点数量 `WAYPOINT_COUNT` 消息。

请求组件会在此时要求每一个序列号 0 开始发送 `WAYPOINT_REQUEST` 消息，同时目标组件相应的回答每一个请求包含航点的数据信息 `waypoint` 航点。

当最后一个航点被成功接收请求组件发送 `WAYPOINT_ACK` 消息到目标的组成部分时，交易完成。请注意，有针对性的组件有听到 `WAYPOINT_REQUEST` 最后一个航点的消息，直到它得到的 `WAYPOINT_ACK` 或启动一个不同的交易或超时发生另一条消息。



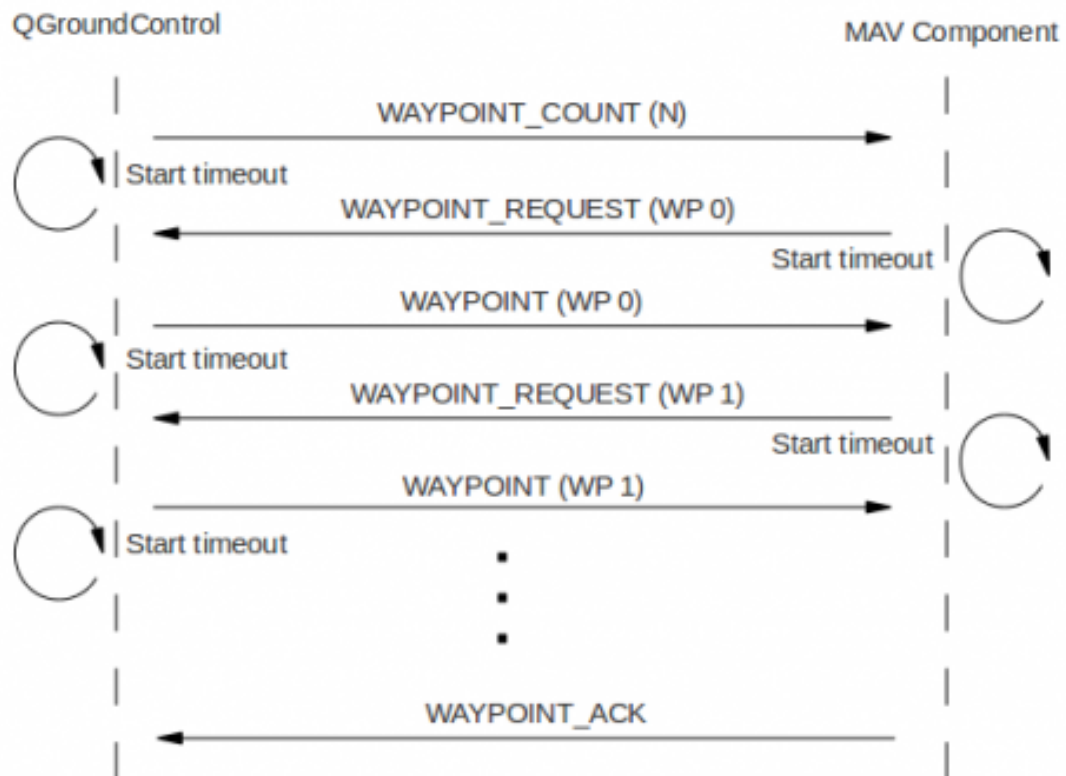


## 写 MAV 的航点列表

发送一个航点列表 `WAYPOINT_COUNT` 含有清单中的数量航点的消息被发送到目标的组成部分。组件会做好接待准备，并开始从 0 发送检索 `WAYPOINT_REQUEST` 所有航点的序列号。航点列表的组件发送应答与所有的请求 `WAYPOINT` 消息

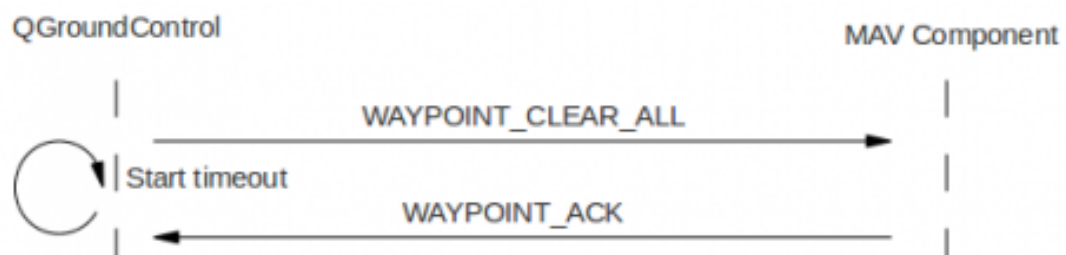
当最后一个航点已成功通过有针对性的组件收到，发送 `WAYPOINT_ACK` 邮件发送组件。这时交易完成。

如果一个航点规划者组件接收 `waypoint` 以外的交易的消息，它应以 `WAYPOINT_ACK` 消息的形式回应。



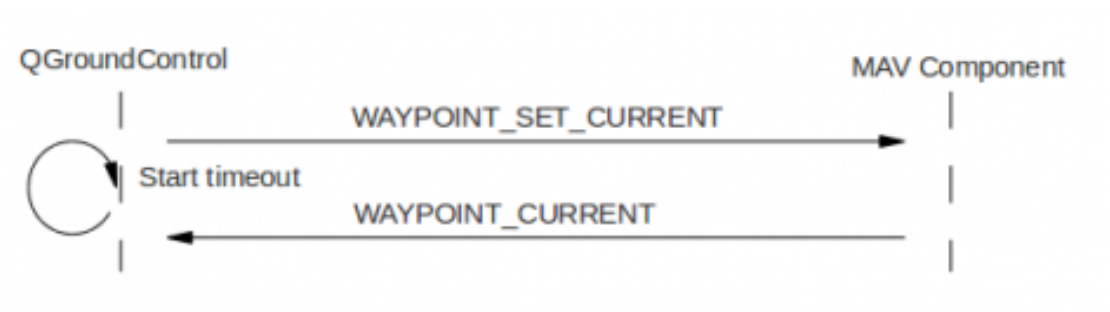
## 清除 MAV 的航点列表

在清除航点列表有针对性的组成部分答案 `WAYPOINT_ACK` 消息后。要清除一个航点列表组成部分 `WAYPOINT_CLEAR_ALL` 的消息发送。



## 设置新的当前 MAV 的航点

在用新的序列号 `WAYPOINT_CURRENT` 消息改变目前的航点后，对新的组件设置一个的活跃航点消息 `WAYPOINT_SET_CURRENT`。



### 航点达到 MAV 的状态消息

如果在 MAV 的航点规划者达到一个航点，它广播 `WAYPOINT_REACHED` 的消息。由于 `ACK` 消息没有通过广播发送，而且也没有接待保证可以被采取。

### 当前航点，改变了从微型飞行器的状态消息

如果在 MAV 的航点规划师，选择当前目标作为其新的航点，广播 `WAYPOINT_CURRENT` 的消息。由于 `ACK` 消息没有通过广播发送，而且也没有接待保证可以被采取。建议将这个消息发送一个小的延迟，以确保达到所有的接收器，高达两倍概率。

## 图像传输协议

本页面描述影像流功能如何实现，包括实施细则（在 MAV 和 QGroundControl）以及 MAV 和 QGroundControl 之间的通信。

图像传输协议包括两个模块：一个图像流和视频流组件：

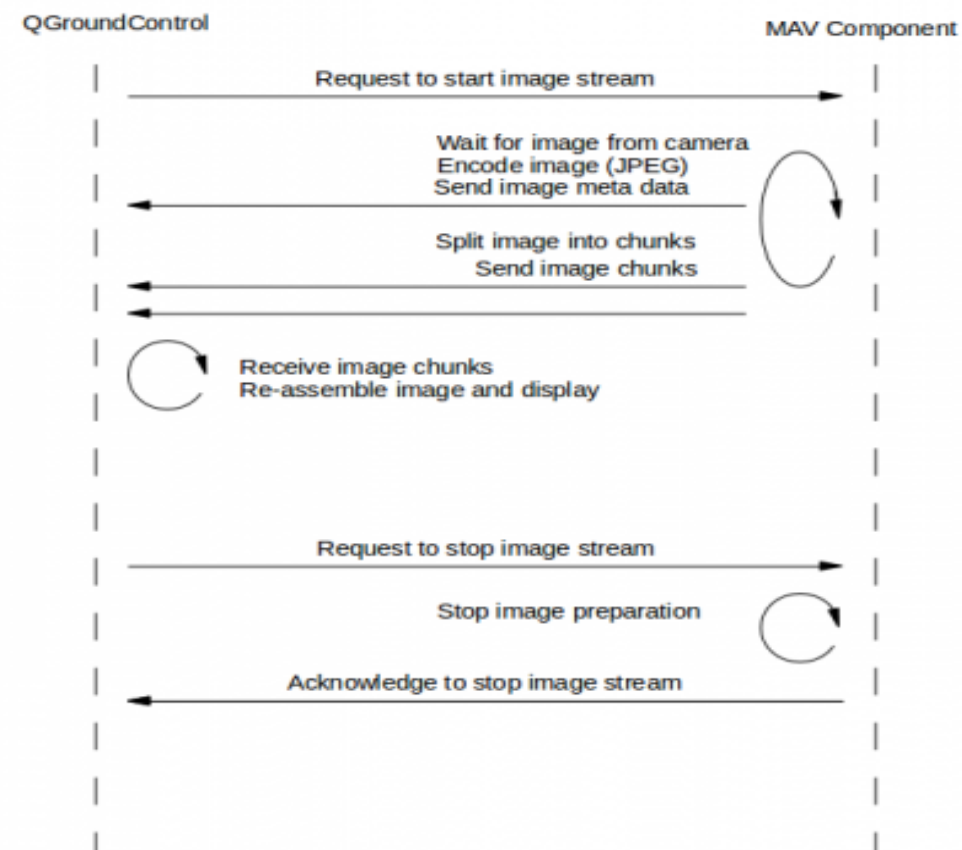
- **图像流** 组件使用 MAVLink 为沟通渠道，并可以用来运送任何从 MAV 到 QGroundControl 的一种图像。它基本上是一个活的摄像机图像，把它分割成小块再通过 MAVLink 发送。此模块主要用于图像直接传送到 QGroundControl（即 HUD 的组件如下所示）
- 组件的**视频流**为“真实的”MPEG2 格式的视频。它采用现场摄像机图像和 [FFmpeg](#) 的视频流进行编码。本模块的主要使用的情况下是观看上几乎任何移动设备（笔记本电脑，智能电话，...），此组件还可以通过移动媒体来发送大量的数据块图像而不局限于 QGroundControl 或 MAVlink 客户端。

视频流，图像流组件的主要优点是更好地融入到 QGroundControl。主要缺点是它需要 MAVLink 支持（因此不作为跨 platform 视频流组件）。

## 通讯

### 影像串流

图像流组件使用两个 mavlink 消息：一个手动消息，`DATA_TRANSMISSION_HANDSHAKE` 发起，控制和停止图像流;和数据容器的消息，`ENCAPSULATED_IMAGE`，传输图像数据（见下图）。



（1）通信由 QGroundControl 发起请求启动流。要做到这一点，必须设置以下字段在 MAVLink 消息：

- 目标：到有针对性的微型飞行器的 ID，
- 状态：通过 0 来发送请求请求，
- ID：ID 的图像流，注：形象流光的那一刻，只支持每一个图像类型的流，因此需要你设置 ID 相同的整数类型的字符。
- 类型：在枚举类型的 mavlink.h 库中的任何 MAVLINK\_DATA\_STREAM\_TYPES
- 频率：大于 0 为“每秒帧数”，低于 0 为“每帧秒数”

要求一个特定的图像质量，这是可能的。要做到这一点，你必须设置质量领域。在初始请求里，所有其他领域应该是零。

(2) 当目标 MAV 的接待 handshake 请求，它发回一个确认，并开始在请求帧率的图像流。要求由 QGroundStation (状态设置为 1，因为它是一个 ACK) handshake ACK 包通常包含相同的值，并增加了关于未来发送图像大小的数据：

- 地址 field 包含的 MAVLink 中的 ENCAPSULATED\_DATA 包，
- payload 字段指定每个数据包的有效载荷（一般为 252 个字节）的大小，
- size 字段指定以字节为单位的图像大小。

(3) 图像数据分割成块以适应分为正常 MAVLink 消息。然后将它们装进 ENCAPSULATED\_DATA 包和超过 MAVLink 发送。每个数据包包含一个序列号，以及它属于图像流的 ID。图像流光现在定期发送新的图像而没有进一步交互的需要。每一个新的形象是一个新的 DATA\_TRANSMISSION\_HANDSHAKE 与更新图像 ACK 数据包 size, packets 和 payload 领域。之后，新图像作为 ENCAPSULATED\_DATA 系列的包到达。注：对于每个流的新图像序列号都是从 0 开始的。

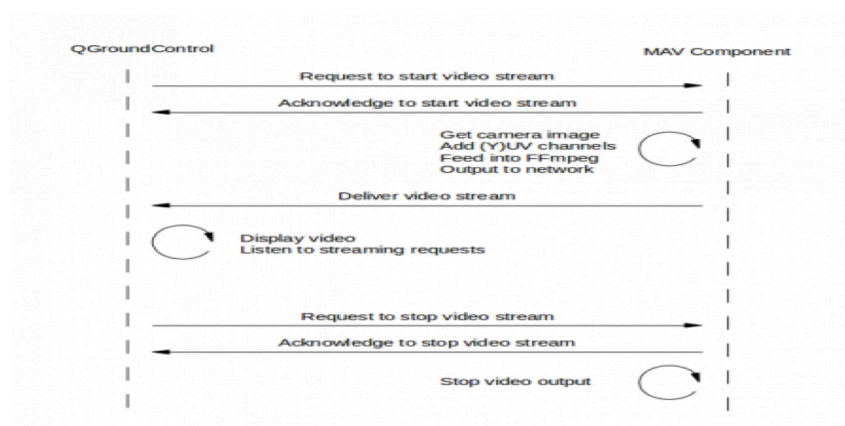
(4) 要停止图像数据流，你必须发送一个新的频率设置为 0 的 DATA\_TRANSMISSION\_HANDSHAKE 请求数据包。微型飞行器将承认由回送一个 ACK 包，包含在请求相同的数据。

## 视频流

视频传输通信协议是比流的一个图像要简单得多：它只是一个 MAVLink 消息，由

VIDEO\_STREAM，  
这是用来启动  
和停止视频流  
(见左图)。

消息有两处需  
要设置：



- `target`: 目标微型飞行器
- `START_STOP`: 1, 开始, 0 停止。

微型飞行器方面 FFmpeg 所产生的视频流。小 MAVLink 包装抓起摄像机的图像，添加（Y）的紫外线渠道 YUV420 rawimage 格式和成分到 FFMPEG 形象。然后将输出发送到地面站（注：目前这需要一个固定的地面站的 IP 以及一个初始配置步骤时，设立了微型飞行器）。根据视频流 recieving, QGroundControl 打开了 VLC 窗口中重新分配视频流：这需要从微型飞行器流，此流的 RTP 流（组播地址）和提供的 HTTP 流（直接单播流）网络。这样做不转原流，以保持尽可能低性能的影响。

现在可以连接到其他移动设备上的多播地址 `239.255.12.45` 流，或 HTTP 的流上 `http://[QGC-HOST]/MAVLive.mpg`。组播流通过 SAP 宣布一下的名称“MAVLive”。

## 使用/配置

MAV 的使用两个模块，你必须执行以下步骤。

### 影像串流

1. 编译您 MAV 的中间件：`mavconn`。
2. 开始对微型飞行器至少这些组件：

```
px_mavlink_bridge_udp &

px_system_control --heartbeat &

px_camera -o lcm &
```

3. 编译和启动 QGroundControl

4. 启动图像流组件（为了看到一些更多的输出，您可以添加-v 标志）：

```
px_imagestreamer
```

5. 发起的图像流：打开 HUD 的部件，用鼠标右键单击到部件，并选择“启用实时图像流”。

你现在应该能够看到一个每秒的图像（默认情况下，目前是硬编码）视频直播。

## 视频流

1. 执行步骤图像流 1 至 3
2. 在你的 home 目录中创建一个符号链接：

```
cd ~  
  
ln -s mavconn/src/comm/video/px_videostreamer.sh px_videostreamer.sh
```

（注意：你也可以复制该文件，但不建议。）

3. 开始对 MAV 的视频流组件：

```
px_videostreamer
```

4. 启动视频流：打开 HUD 的部件，到部件右键单击并选择“启用视频直播饲料”。

一个的 VLC 窗口现在应该打开。不要关闭窗口，只要你想要的视频流给他人！如果你想观看当前流，只是在另一个 VLC 窗口打开流。



## 参数协议

内建的参数接口允许读取和写入到当前存储器(RAM)，也将永久存储(EEPROM 或硬盘)参数(如 PID 增益)实现，它可以是一个微控制器(如与 ARM7 pxIMU)标准的软件(例如在 Linux px\_multitracker 过程)。

## 支持的数据类型

MAVLink V1.0 支持这些数据类型：

- uint32\_t - 32 位无符号整数 (使用的枚举值 `MAVLINK_TYPE_UINT32_T`)
- int32\_t - 32 位有符号整数 (使用枚举值 `MAVLINK_TYPE_INT32_T`)
- 浮点- IEEE754 单精度浮点数 (使用枚举值 `MAVLINK_TYPE_FLOAT`)

## 多系统和多组件支持

MAVLink 支持并行在同一链路上的多个系统/飞机。除此之外，它也支持在同一飞机多个 MAVLink 设备。例如协议允许一个无线链路的自动驾驶仪和有效载荷单元进行通信。参数协议，也因为这个原因，组件之间的区别。从系统得到一个完整的参数列表，在 `target_component` 设置为 0 时 (枚举值: `MAV_COMP_ID_ALL`)，发送请求参数的消息。所有板上的元件响应参数请求消息，用自己的 ID 或 `IDMAV_COMP_ID_ALL` (0)。QGroundControl 默认查询所有系统组件 (只查询当前选定的系统，并不是所有的系统)，因此发送的 ID 0 / `MAV_COMP_ID_ALL`。

## 通讯/主机

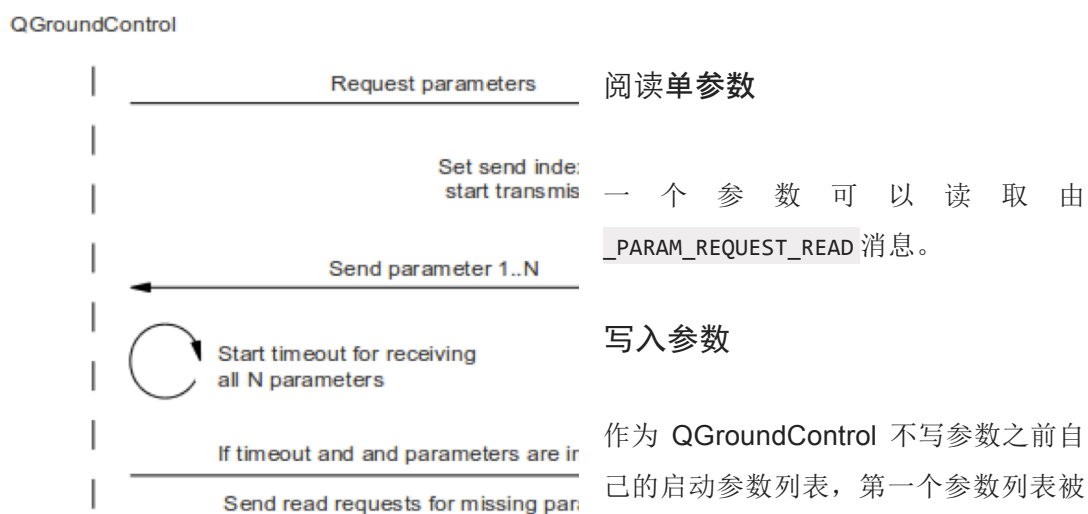
内建的参数，确定由 14 字符的字符串，并存储一个浮点值 (独立外部评价的 754 单精度，4 个字节)。此键→值对有许多重要的特性：

- 可读的名字，对于用户来说非常有用，但它仍然是足够小
- 地方选区没有提前知道板载参数存在

- 支持对未知的自动驾驶仪，只要他们执行该协议，保证存在
- 加入参数仅仅是一个内建代码的变化。

## 读取参数

读取参数列表被激活发送 `PARAM_REQUEST_LIST` 消息。板载组件应该开始收到此消息后，单独传送的参数。每个参数后，应推迟发送，以不占用完整的无线电带宽。



面发送键→值对组件。作为 QGroundControl 保持轨道改变的参数，点击 Write 按钮只会发送那些已经改变的值。MAV 需要承认写操作所散发出的新参数值与 `PARAM_VALUE` 值的消息。

## 板载永久存储

参数接口提供了两个按钮，载入参数从永久板载存储到 RAM 和当前参数保存到永久存储在 RAM。写一个新的值在上 QGroundControl 永久存储，首先值需要被传送，事后再写入 ROM 命令发出。

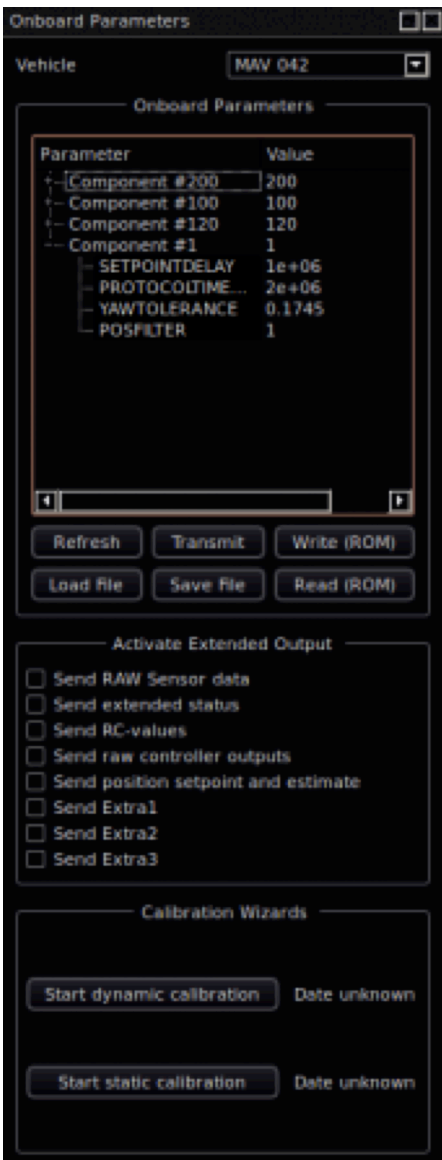
## QGroundControl 参数文件

QGroundControl 允许板载参数的当前值保存在一个文本文件。该文件可以被再次引入和传输的微型飞行器。这样就使许多飞行器装配有相同的省略值。

## 在 QGroundControl 图形用户界面

出于这个原因参数界面会区分系统的组成，这样不需要就翻译参数的中心载板的情况下能够明显地得到分散的成分参数。右边的图像中可以看到，每一个组件代表在参数树的高位节点。在顶层的下拉菜单，可以选择系统（微型飞行器）。**GUI** 保持轨道改变的参数将发送这些参数，从而改变相应的组件。

为了促进许多参数使用，树是在根据参数名称中的第一个下划线(“\_”)的顶层结构，。所以 `PID_POS_X_P` 和 `PID_POS_Y_P` 将低于 `PID` 节点进行分组。



字段及描述

字段名称	描述
AV_TYPE_GENERIC	通用的微型飞行器。
AV_TYPE_FIXED_WING	固定翼飞机。
AV_TYPE_QUADROTOR	旋翼
AV_TYPE_COAXIAL	共轴式直升机
AV_TYPE_HELICOPTER	普通带尾桨的直升机。
AV_TYPE_ANTENNA_TRACKER	地面安装装置
AV_TYPE_GCS	操作员控制单元/地面控制站
AV_TYPE_AIRSHIP	飞艇（受控制）
AV_TYPE_FREE_BALLOON	自由气球（不受控制）
AV_TYPE_ROCKET	火箭
AV_TYPE_GROUND_ROVER	地面流动站
AV_TYPE_SURFACE_BOAT	水面舰艇，艇，船
AV_TYPE_SUBMARINE	潜艇

MAV_TYPE_HEXAROTOR	六旋翼
MAV_TYPE_OCTOROTOR	八旋翼
MAV_TYPE_TRICOPTER	三旋翼
MAV_TYPE_FLAPPING_WING	扑翼
MAV_TYPE_KITE	扑翼

## MAV\_MODE\_FLAG

这些标志 MAV 的模式进行编码。

CMD 编号	字段名称	描述
128	MAV_MODE_FLAG_SAFETY_ARMED	0b10000000 MAV 的安全设置。  电机启动/运行/可以启动。准备起飞。
64	MAV_MODE_FLAG_MANUAL_INPUT_ENABLED	0b01000000 启用远程控制输入。
32	MAV_MODE_FLAG_HIL_ENABLED	0b00100000 硬件在线仿真。所有电机/伺服是可以完整的运行的。
16	MAV_MODE_FLAG_STABILIZE_ENABLED	0b00010000 系统稳定电子状态（可选位置）。但它需要进一步控制输入。
8	MAV_MODE_FLAG_GUIDED_ENABLED	0b00001000 制导模式启用，系统飞行任务

CMD 编号	字段名称	描述
		任务栏。
4	MAV_MODE_FLAG_AUTO_ENABLED	0b00000100 自驾模式启用，系统可找到E  位置。取决于实际情况，可以选择是否设  目标点。
2	MAV_MODE_FLAG_TEST_ENABLED	0b00000010 系统启用测试模式。此标志用  临时的系统测试，稳定的情况下不应使用
1	MAV_MODE_FLAG_CUSTOM_MODE_ENABLED	0b00000001 保留以供将来使用。

## MAV\_MODE\_FLAG\_DECODE\_POSITION

这些值编码解码位置位的位置。这些值可以用来读取 *base\_mode* 变量结合起来，与标志位置值的一个标志位的值。其结果将是 0 或 1，取决于如果设置与否的标志。

CMD 号	字段名称	描述
128	MAV_MODE_FLAG_DECODE_POSITION_SAFETY	首先一点：10000000
64	MAV_MODE_FLAG_DECODE_POSITION_MANUAL	第二位：01000000
32	MAV_MODE_FLAG_DECODE_POSITION_HIL	第三位：00100000
16	MAV_MODE_FLAG_DECODE_POSITION_STABILIZE	第四位：00010000
8	MAV_MODE_FLAG_DECODE_POSITION_GUIDED	第五位：00001000

CMD 号	字段名称	描述
4	MAV_MODE_FLAG_DECODE_POSITION_AUTO	第六位：00000100
2	MAV_MODE_FLAG_DECODE_POSITION_TEST	第七位：00000010
1	MAV_MODE_FLAG_DECODE_POSITION_CUSTOM_MODE	第八位：00000001

## MAV\_GOTO

覆盖命令，暂停执行当前的任务，并立即移动到一个位置

CMD 编号	字段名称	描述
0	MAV_GOTO_DO_HOLD	保持在当前位置。
1	MAV_GOTO_DO_CONTINUE	继续在执行任务的下一个项目。
2	MAV_GOTO_HOLD_AT_CURRENT_POSITION	保持在当前的系统位置
3	MAV_GOTO_HOLD_AT_SPECIFIED_POSITION	保持在 DO_HOLD 参数指定的位置。

## MAV\_MODE

这些定义预定义或组合模式标志。有没有必要使用此枚举的值，但它简化了使用的模式标志。  
注意：启用安全覆盖所有的模式，手动输入。

CMD 号	字段名称	描述
0	MAV_MODE_PREFLIGHT	在无标志设置的情况下，系统还没有准备好起飞，5，校准等。
80	MAV_MODE_STABILIZE_DISARMED	辅助的 RC 控制下，系统是活跃的。

CMD 号	字段名称	描述
208	MAV_MODE_STABILIZE_ARMED	辅助的 RC 控制下，系统是活跃的。
64	MAV_MODE_MANUAL_DISARMED	在人工 RC 控制下，系统是活跃的，但是不稳定。
192	MAV_MODE_MANUAL_ARMED	在人工 RC 控制下，系统是活跃的，但是不稳定。
88	MAV_MODE_GUIDED_DISARMED	手动设定自主控制下，系统是活跃的。
216	MAV_MODE_GUIDED_ARMED	手动设定自主控制下，系统是活跃的。
92	MAV_MODE_AUTO_DISARMED	在主控制和导航状态下，系统是活跃的（轨迹由板载和未预先设定的 MISSIONS 决定）。
220	MAV_MODE_AUTO_ARMED	在主控制和导航状态下，系统是活跃的（轨迹由板载和未预先设定的 MISSIONS 决定）。
66	MAV_MODE_TEST_DISARMED	未定义模式。这完全取决于上的自动驾驶仪（谨慎只适用于开发商。）
194	MAV_MODE_TEST_ARMED	未定义模式。这完全取决于上的自动驾驶仪（谨慎只适用于开发商。）

## MAV\_STATE

CMD 编号	字段名称	描述
0	MAV_STATE_UNINIT	初始化系统，状态是未知的。



CMD 编号	字段名称	描述
	MAV_STATE_BOOT	系统启动。
	MAV_STATE_CALIBRATING	系统校准和飞行准备。
	MAV_STATE_STANDBY	系统接地，处于待命状态。它可以随时发射。
	MAV_STATE_ACTIVE	系统是活跃的，可能已经空降。机体正在准备。
	MAV_STATE_CRITICAL	系统是在非正常的飞行模式。不过，它可以仍然导航。
	MAV_STATE_EMERGENCY	系统是在非正常的飞行模式。它失去了部分或整个机身的控制。发出求救信号并进行降落。
	MAV_STATE_POWEROFF	系统只是初始化断电程序，将立即关闭。

## MAV\_COMPONENT

CMD 号	字段名称	描述
0	MAV_COMP_ID_ALL	
220	MAV_COMP_ID_GPS	
190	MAV_COMP_ID_MISSIONPLANNER	
195	MAV_COMP_ID_PATHPLANNER	
180	MAV_COMP_ID_MAPPER	

CMD 号	字段名称	描述
100	MAV_COMP_ID_CAMERA	
200	MAV_COMP_ID_IMU	
201	MAV_COMP_ID_IMU_2	
202	MAV_COMP_ID_IMU_3	
240	MAV_COMP_ID_UDP_BRIDGE	
241	MAV_COMP_ID_UART_BRIDGE	
250	MAV_COMP_ID_SYSTEM_CONTROL	
140	MAV_COMP_ID_SERVO1	
141	MAV_COMP_ID_SERVO2	
142	MAV_COMP_ID_SERVO3	
143	MAV_COMP_ID_SERVO4	
144	MAV_COMP_ID_SERVO5	
145	MAV_COMP_ID_SERVO6	
146	MAV_COMP_ID_SERVO7	
147	MAV_COMP_ID_SERVO8	

CMD 号	字段名称	描述
148	MAV_COMP_ID_SERVO9	
149	MAV_COMP_ID_SERVO10	
150	MAV_COMP_ID_SERVO11	
151	MAV_COMP_ID_SERVO12	
152	MAV_COMP_ID_SERVO13	
153	MAV_COMP_ID_SERVO14	

## MAV\_FRAME

CMD 编号	字段名称	描述
0	MAV_FRAME_GLOBAL	全球坐标系，WGS84 坐标系。第一个值/ X：纬度，第二个值/ Y：经度，第三个值/ Z：高出海平面的高度值。
1	MAV_FRAME_LOCAL_NED	局部坐标系，Z 轴正方向（X：北部，Y：东，Z：向下）
2	MAV_FRAME_MISSION	不是一个坐标系，表示任务的命令。
3	MAV_FRAME_GLOBAL_RELATIVE_ALT	全球坐标系，WGS84 坐标系统，对地面的相对高度与位置。第一个值/ X：纬度，第二个值/ Y：经度，第三个值/ Z：高出海平面的高度值。

CMD 编号	字段名称	描述
4	MAV_FRAME_LOCAL_ENU	局部坐标系, Z 轴负方向 (X: 东, Y: 北, Z: 下)

## MAVLINK\_DATA\_STREAM\_TYPE

CMD 号	字段名称	描述
	MAVLINK_DATA_STREAM_IMG_JPEG	
	MAVLINK_DATA_STREAM_IMG_BMP	
	MAVLINK_DATA_STREAM_IMG_RAW8U	
	MAVLINK_DATA_STREAM_IMG_RAW32U	
	MAVLINK_DATA_STREAM_IMG_PGM	
	MAVLINK_DATA_STREAM_IMG_PNG	

## MAV\_CMD

要执行的命令的微型飞行器。他们可以执行用户的请求，或者作为一个任务脚本的一部分。如果这个动作是在任务中使用，参数映射到航点/任务的消息如下：1 PARAM, 参数 2, 参数 3, 参数 4, X: PARAM 5 Y: PARAM 6, Z 轴: PARAM 7。此命令列表是类似的 ARINC 424 是为商用飞机的数据格式如何解释航点/任务数据。

CMD 编号	字段名称	描述
16	MAV_CMD_NAV_WAYPOINT	导航任务。
	Mission Param # 1	保持十进制的时间。（固定翼，旋转翼忽略）

CMD 编号	字段名称	描述
		略)
	Mission Param # 2	以米为半径的验收（如果这个半球被击中，任务才算达成)
	Mission Param # 3	0 通过的 WP，如果> 0 米通过 WP。正值  针轨道，负数逆时针轨道。允许轨迹控制。
	Mission Param # 4	MISSIONS 中的偏航角（旋转翼)
	Mission Param # 5	纬度
	Mission Param # 6	经度
	Mission Param # 7	海拔
	17 MAV_CMD_NAV_LOITER_UNLIM	无限延长执行任务的时间
	Mission Param # 1	空的
	Mission Param # 2	空的
	Mission Param # 3	以米为单位在 MISSIONS 周围。如果是， 时针，否则逆时针
	Mission Param # 4	期望偏航角。
	Mission Param # 5	纬度
	Mission Param # 6	经度

CMD 编号	字段名称	描述
18	Mission Param # 7	海拔
	<b>MAV_CMD_NAV_LOITER_TURNS</b>	执行 MISSIONS，X 秒
	Mission Param # 1	打开
	Mission Param # 2	空的
	Mission Param # 3	使命米，周围半径。如果正游荡顺时针方 ， 否则逆时针
	Mission Param # 4	期望偏航角。
	Mission Param # 5	纬度
	Mission Param # 6	经度
19	Mission Param # 7	海拔
	<b>MAV_CMD_NAV_LOITER_TIME</b>	解决这个任务游荡 X 秒
	Mission Param # 1	秒（十进制）
	Mission Param # 2	空的
	Mission Param # 3	以米为单位在 MISSIONS 周围。如果是， 时针， 否则逆时针
	Mission Param # 4	期望偏航角。
	Mission Param # 5	纬度

CMD 编号	字段名称	描述
20	Mission Param # 6	经度
	Mission Param # 7	海拔
	<b>MAV_CMD_NAV_RETURN_TO_LAUNCH</b>	返回发射地点
	Mission Param # 1	空的
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
21	<b>MAV_CMD_NAV_LAND</b>	在指定位置
	Mission Param # 1	空的
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	期望偏航角。
	Mission Param # 5	纬度
	Mission Param # 6	经度

CMD 编号	字段名称	描述
22	Mission Param # 7	海拔
	<b>MAV_CMD_NAV_TAKEOFF</b>	从地面/手中起飞
	Mission Param # 1	最小间距（实时空速传感器）， 无需传感器
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	偏航角（如果磁力出席）， 若无体则忽略
	Mission Param # 5	纬度
	Mission Param # 6	经度
80	Mission Param # 7	海拔
	<b>MAV_CMD_NAV_ROI</b>	设置传感器集的地区或机体本身的利反馈 ROI）。然后可以使用的飞机控制系统来控制机体的态度和各种传感器，如相机的态度
	Mission Param # 1	interest 模式的地区。（见 MAV_ROI 枚举）
	Mission Param # 2	任务指标/目标 ID。（见 MAV_ROI 枚举）
	Mission Param # 3	反馈指标（允许飞机管理多个 ROI）
	Mission Param # 4	空的
	Mission Param # 5	x 的位置固定的反馈值（见 MAV_FRAME 枚举）



CMD 编号	字段名称	描述
81	Mission Param # 6	$\dot{Y}$
	Mission Param # 7	$\dot{Z}$
	<b>MAV_CMD_NAV_PATHPLANNING</b>	微型飞行器自主路径规划的控制
	Mission Param # 1	0: 禁用局部避障/本地路径规划（无需重置地图），1：启用本地路径规划，2：启用和局部路径规划
	Mission Param # 2	0: 禁用完整的路径规划（无需重置地图） 1：启用 2：启用和重置地图/占用电网， 3：启用和重置计划的路线，但不占用电网
	Mission Param # 3	空的
	Mission Param # 4	在目标偏航角度，指南针度，[0 到 360]
	Mission Param # 5	纬度/目标 X
	Mission Param # 6	目标的经度/Y
	Mission Param # 7	目标的高度/Z
95	<b>MAV_CMD_NAV_LAST</b>	NOP - 此命令仅用于 NAV/ACTION 的测量
	Mission Param # 1	空的

CMD 编号	字段名称	描述
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
112	<b>MAV_CMD_CONDITION_DELAY</b>	延迟任务状态。
	Mission Param # 1	在几秒钟的延迟（十进制）
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
113	<b>MAV_CMD_CONDITION_CHANGE_ALT</b>	以一定速率上升/下降。拖延执行任务的机 直到到达所需的高度。
	Mission Param # 1	上升/下降的速率（米/秒）

CMD 编号	字段名称	描述
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	完成的高度
114	<b>MAV_CMD_CONDITION_DISTANCE</b>	推迟 NAV 点的直到打到下一状态
	Mission Param # 1	距离（米）
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
115	<b>MAV_CMD_CONDITION_YAW</b>	达到一定的目标角度。
	Mission Param # 1	目标角度：[0-360], 0 北

CMD 编号	字段名称	描述
	Mission Param # 2	在偏航变化速度：（每秒度）
	Mission Param # 3	负值：逆时针；正值：顺时针[-1,1]
	Mission Param # 4	相对偏移或绝对角度：[1,0]
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
159	<b>MAV_CMD_CONDITION_LAST</b>	NOP - 此命令只能用来标记在测量上限的
	Mission Param # 1	空的
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
176	<b>MAV_CMD_DO_SET_MODE</b>	设置系统模式。
	Mission Param # 1	模式，用 ENUM MAV_MODE 来定义

CMD 编号	字段名称	描述
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
177	<b>MAV_CMD_DO_JUMP</b>	跳转到任务列表中所需的命令。只有在指定次数下重复这个动作
	Mission Param # 1	序列号
	Mission Param # 2	重复计数
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
178	<b>MAV_CMD_DO_CHANGE_SPEED</b>	改变速度和/或节流阀的设定。

CMD 编号	字段名称	描述
	Mission Param # 1	速度类型（0 =空速, 1 =地面速度)
	Mission Param # 2	速度（米/秒, -1 表示没有变化)
	Mission Param # 3	节流阀（百分比, -1 表示没有变化)
	Mission Param # 4	空的
	Mission Param # 5	空的
	使命 PARAM # 6	空的
	使命 PARAM # 7	空的
179	<b>MAV_CMD_DO_SET_HOME</b>	以当前的位置或指定位置改变地面站的位置
	Mission Param # 1	目前使用（1 =使用当前的位置, 0 =使用指定的位置)
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	纬度
	Mission Param # 6	经度
	Mission Param # 7	海拔
180	<b>MAV_CMD_DO_SET_PARAMETER</b>	设置系统参数。注意, 使用此命令需要参

CMD 编号	字段名称	描述
		数字测算的知识。
	Mission Param # 1	参数号
	Mission Param # 2	参数值
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
181	MAV_CMD_DO_SET_RELAY	继电器设置条件。
	Mission Param # 1	继电器号码
	Mission Param # 2	设置（1 =开启；0 =关闭，其他的取决于系统自身硬件）
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的

CMD 编号	字段名称	描述
182	<b>MAV_CMD_DO_REPEAT_RELAY</b>	周期继电器和关闭所需的时间与所需的圈量。
	Mission Param # 1	继电器号码
	Mission Param # 2	循环计数
	Mission Param # 3	周期时间（秒，十进制）
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
183	<b>MAV_CMD_DO_SET_SERVO</b>	设置伺服所需的 PWM 值。
	Mission Param # 1	伺服器数量
	Mission Param # 2	PWM（微秒，典型的 1000 至 2000）
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的



CMD 编号	字段名称	描述
184	<b>MAV_CMD_DO_REPEAT_SERVO</b>	周期之间的设置和指定时间  和指定周期 PWM 所需数量
	Mission Param # 1	伺服器数量
	Mission Param # 2	PWM (微秒, 典型的 1000 至 2000)
	Mission Param # 3	循环计数
	Mission Param # 4	周期时间 (秒)
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
200	<b>MAV_CMD_DO_CONTROL_VIDEO</b>	控制板上的摄像系统。
	Mission Param # 1	摄像机 ID (所有的: -1)
	Mission Param 2	变速箱: 0: 禁用 1: 启用压缩, 2: 启用原
	Mission Param 3	传输模式: 0: 视频流, > 0: 每 n 秒的单 像 (十进制)
	Mission Param # 4	录音: "0: 禁用 1: 启用压缩; 2: 启用原
	Mission Param # 5	空的
	Mission Param # 6	空的

CMD 编号	字段名称	描述
240	Mission Param # 7	空的
	<b>MAV_CMD_DO_LAST</b>	NOP - 此命令只能用来标记的 DO 命令枚
		上限
	Mission Param # 1	空的
	Mission Param # 2	空的
	Mission Param # 3	空的
	Mission Param # 4	空的
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
241	<b>MAV_CMD_PREFLIGHT_CALIBRATION</b>	触发校准。此命令将只接受飞行前的模式。
	Mission Param # 1	陀螺仪校准：0：无， 1：是
	Mission Param # 2	磁力计校准：0：无， 1：是
	Mission Param # 3	接地压力：0：NO， 1：是
	Mission Param # 4	无线电校准：0：无， 1：是
	Mission Param # 5	空的
	Mission Param # 6	空的

CMD 编号	字段名称	描述
242	Mission Param # 7	空的
	<b>MAV_CMD_PREFLIGHT_SET_SENSOR_OFFSETS</b>	设置传感器补偿。此命令将只接受飞行前式。
	Mission Param # 1	传感器调整偏移：0：陀螺仪，加速度计，磁力仪，3：晴雨表，4：光流
	Mission Param # 2	X 轴偏移（或通用尺寸 1），在传感器的原始单位
	Mission Param # 3	Y 轴偏移（或通用尺寸 2），在传感器的原始单位
	Mission Param # 4	Z 轴偏移（或通用尺寸 3），在传感器的原始单位
	Mission Param # 5	通用尺寸 4，在传感器的原始单位
880	Mission Param # 6	通用尺寸 5，在传感器的原始单位
	Mission Param # 7	通用尺寸 6，在传感器的原始单位
	<b>MAV_CMD_PREFLIGHT_STORAGE</b>	不同的参数值和日志要求存储。
	Mission Param # 1	参数存储：0 阅读：闪存/EEPROM，1 写前的闪存/EEPROM
	Mission Param # 2	使命存储：0 阅读：闪存/EEPROM，1 写

CMD 编号	字段名称	描述
		前的 FLASH / EEPROM
	Mission Param # 3	版权所有
	Mission Param # 4	版权所有
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
246	<b>MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN</b>	重新启动或关闭系统组件。
	Mission Param # 1	0: 不对自驾仪产生影响，1：重新启动自驾仪，2：关闭自动驾驶仪。
	Mission Param # 2	0: 机载计算机，1：板载重新启动计算机，2：关闭机载计算机。
	Mission Param # 3	版权所有
	Mission Param # 4	版权所有
	Mission Param # 5	空的
	Mission Param # 6	空的
	Mission Param # 7	空的
252	<b>MAV_CMD_OVERRIDE_GOTO</b>	保留/继续目前的行动

CMD 编号	字段名称	描述
	Mission Param # 1	MAV_GOTO_DO_HOLD: 按住 MAV_GOTO_DO_CONTINUE: 继续下一个项目任务计划
	Mission Param # 2	在指定的位置: 在当前位置 MAV_GOTO_HOLD_AT_SPECIFIED_POSITION: 指定位置  MAV_GOTO_HOLD_AT_CURRENT_POSITION: 当前位置
	Mission Param # 3	MAV_FRAME 坐标框架保持点
	Mission Param # 4	所需偏航角的度数
	Mission Param # 5	纬度/ X 的位置
	Mission Param # 6	经度/ Y 的位置
	Mission Param # 7	高度/ Z 的位置
300	MAV_CMD_MISSION_START	开始执行任务
	Mission Param # 1	first_item: 第一个任务项目运行
	Mission Param # 2	last_item: 最后一个任务运行项目 (该项目运行后, 任务结束)
400	MAV_CMD_COMPONENT_ARM_DISARM	装备/卸载组件

CMD 编号	字段名称	描述
	Mission Param # 1	1 : 安装 ; 0 : 解除

## MAV\_DATA\_STREAM

数据流的 ID。一个数据流是没有一套固定的消息，而是一个自动驾驶仪软件的建议。个别的自动驾驶仪可能或可能不遵守建议的消息。

CMD 编号	字段名称	描述
0	MAV_DATA_STREAM_ALL	使所有的数据流
1	MAV_DATA_STREAM_RAW_SENSORS	启用 IMU_RAW, GPS_RAW, GPS_STATUS
2	MAV_DATA_STREAM_EXTENDED_STATUS	启用 GPS_STATUS, CONTROL_STATUS, AUX_STATUS
3	MAV_DATA_STREAM_RC_CHANNELS	启用 RC_CHANNELS_SCALED, RC_CHANNELS_RAW, SERVO_OUTPUT_RAW“
4	MAV_DATA_STREAM_RAW_CONTROLLER	启用 ATTITUDE_CONTROLLER_OUTPUT, POSITION_CONTROLLER_OUTPUT, NAV_CONTROLLER_OUTPUT。
6	MAV_DATA_STREAM_POSITION	启用的 LOCAL_POSITION, GLOBAL_POSITION / GLOBAL_POSITION_INT 消息。

CMD 编号	字段名称	描述
10	<b>MAV_DATA_STREAM_EXTRA1</b>	取决于自动驾驶仪
11	<b>MAV_DATA_STREAM_EXTRA2</b>	取决于自动驾驶仪
12	<b>MAV_DATA_STREAM_EXTRA3</b>	取决于自动驾驶仪

## MAV\_ROI

车辆的 *ROI*（感兴趣区域）。这可以被用来为相机车辆/车辆态度对齐（见 *MAV\_CMD\_NAV\_ROI*）。

CMD 编号	字段名称	描述
0	<b>MAV_ROI_NONE</b>	无关区域。
1	<b>MAV_ROI_WPNEXT</b>	指向下一个任务。
2	<b>MAV_ROI_WPINDEX</b>	指向赋予的使命。
3	<b>MAV_ROI_LOCATION</b>	指向固定位置。
4	<b>MAV_ROI_TARGET</b>	指向给定的 id。

## MAV\_CMD\_ACK

的 *ACK* / *NACK* / 错误作为一个 *MAV\_CMDs* 的结果和任务项目传输值。

CMD 编号	字段名称	描述
	<b>MAV_CMD_ACK_OK</b>	命令/任务项目是确定的

CMD 编号	字段名称	描述
	MAV_CMD_ACK_ERR_FAIL	通用的错误信息（如果没有其他原因失败及详细的错误报告）
	MAV_CMD_ACK_ERR_ACCESS_DENIED	该系统拒绝接受此命令/通讯伙伴。
	MAV_CMD_ACK_ERR_NOT_SUPPORTED	命令或任务的项目不支持其它命令将被接受。
	MAV_CMD_ACK_ERR_COORDINATE_FRAME_NOT_SUPPORTED	不支持此命令/任务项目坐标系。
	MAV_CMD_ACK_ERR_COORDINATES_OUT_OF_RANGE	这个命令的坐标系是正确，但他的坐标值超过本系统的安全限制。这是一个通用的错误，如果可能的话使用下面更具体的错误代码
	MAV_CMD_ACK_ERR_X_LAT_OUT_OF_RANGE	X 或纬度值超出范围。
	MAV_CMD_ACK_ERR_Y_LON_OUT_OF_RANGE	Y 或经度值超出范围。
	MAV_CMD_ACK_ERR_Z_ALT_OUT_OF_RANGE	Z 或高度值超出范围。



# MAV\_RESULT

导致从 *mavlink* 命令

CMD 编号	字段名称	描述
0	MAV_RESULT_ACCEPTED	被接受和执行的命令
1	MAV_RESULT_TEMPORARILY_REJECTED	命令暂时拒绝/拒绝
2	MAV_RESULT_DENIED	命令一直被拒绝
3	MAV_RESULT_UNSUPPORTED	命令的未知/不支持
4	MAV_RESULT_FAILED	执行命令，但失败了

# MAV\_MISSION\_RESULT

导致在 *mavlink* 使命 ACK

CMD 编号	字段名称	描述
0	MAV_MISSION_ACCEPTED	任务接受确定
1	MAV_MISSION_ERROR	一般性错误/不接受任务命令所有权利
2	MAV_MISSION_UNSUPPORTED_FRAME	不支持坐标系
3	MAV_MISSION_UNSUPPORTED	命令不支持
4	MAV_MISSION_NO_SPACE	任务项目超过存储空间
5	MAV_MISSION_INVALID	参数的值无效

CMD 编号	字段名称	描述
6	<b>MAV_MISSION_INVALID_PARAM1</b>	param1 的有一个无效的值
7	<b>MAV_MISSION_INVALID_PARAM2</b>	param2 的有一个无效的值
8	<b>MAV_MISSION_INVALID_PARAM3</b>	param3 有一个无效的值
9	<b>MAV_MISSION_INVALID_PARAM4</b>	param4 有一个无效的值
10	<b>MAV_MISSION_INVALID_PARAM5_X</b>	param5 有一个无效的值
11	<b>MAV_MISSION_INVALID_PARAM6_Y</b>	param6 有一个无效的值
12	<b>MAV_MISSION_INVALID_PARAM7</b>	param7 有一个无效的值
13	<b>MAV_MISSION_INVALID_SEQUENCE</b>	航线偏离
14	<b>MAV_MISSION_DENIED</b>	不从此接受任何任务的命令

## MAV\_SEVERITY

表示严重性级别，一般用于状态消息，以表明其相对的紧迫性。使用扩展的定义：  
<http://www.kiwisyslog.com/kb/info:-syslog-message-levels/>基于 RFC-5424。

CMD 编号	字段名称	描述
0	<b>MAV_SEVERITY_EMERGENCY</b>	系统无法使用。这是一种“恐慌”的状况。
1	<b>MAV_SEVERITY_ALERT</b>	应立即采取行动。表示在非关键系统的错误。
2	<b>MAV_SEVERITY_CRITICAL</b>	必须立即采取行动。表示在主系统发生故障。

CMD 编号	字段名称	描述
3	MAV_SEVERITY_ERROR	表示在次级/多余的系统错误。
4	MAV_SEVERITY_WARNING	关于未来可能出现的错误，表示如果这不是一个给定的时间内解决。比如一个低电量警告。
5	MAV_SEVERITY_NOTICE	一个不寻常的事件发生， 但不是一个错误。这应该追究其源。
6	MAV_SEVERITY_INFO	正常运作的消息。用于记录有用的。这些消息不需要任何作。
7	MAV_SEVERITY_DEBUG	有用的非经营性的信息， 可以协助调试。这些不应该发生正常运作。

# MAVLink 讯息

## 心跳 ( #0 )

心跳信息显示系统，是实时的。MAV 和自动驾驶仪的硬件类型，允许接收系统来处理，从这个系统得到适当（例如，通过铺设的用户界面上的自动驾驶仪）的进一步消息。

字段名称	类型	描述
type	uint8_t	微型飞行器的类型（旋翼，直升机等，高达 15 在 MAV_TYPE ENUM 的定义)
autopilot	uint8_t	自动驾驶仪的类型/类。

字段名称	类型	描述
		在 MAV_AUTOPILOT ENUM 的定义
base_mode	uint8_t	系统模式位域，枚举  MAV_MODE_FLAGS 在 mavlink /include/  mavlink_types.h
custom_mode	uint32_t	一个为自动档特定标志的使用位域。
system_status	uint8_t	系统状态标志，见 MAV_STATE 的 ENUM
mavlink_version	uint8_t_mavlink_version	MAVLink 版本，而不是由用户可写的，被添加  议，因为神奇的数据类型：  uint8_t_mavlink_version

## # (SYS\_STATUS )

一般的系统状态。如果系统是继 MAVLink 标准，是系统状态，主要是由三个正交的状态/模式：该系统的模式，这是锁定（发动机关闭和锁定），手动（下钢筋混凝土控制系统），导（系统定义自主定位控制，位置的设定手动控制）或自动（系统路径/航点规划师的指导下）。NAV\_MODE 定义当前的飞行状态：升空（通常是一个开环的机动），着陆，航点或向量。这代表内部导航状态机。系统状态显示天气的系统是目前活跃或不活跃，如果发生紧急情况。微型飞行器的关键和紧急状态期间仍然被认为是积极的，但应启动应急自主程序。发生故障后，首先应该在经过一定的盲区允许人工干预。

字段名称	类型	描述
onboard_control_sensors_present	uint32_t	位掩码显示板载控制器和传感器都存在。  0：没有目前的价值。  1：目前的价值。指数：0：3D 陀螺仪，1：3D A  ，2：三维磁，3：绝对压力，差压，5：全球定位

字段名称	类型	描述
		统，6：光流，7：计算机视觉位置，8：基于激光位置，9：外部地面真相（维科或 Leica）。控制器：10：3D 角速度管制 11：心态趋稳，12：13 偏航位置，Z /高度控制，14：X / Y 位置控制，15：电机输出/控制
onboard_control_sensors_enabled	uint32_t	启用板载控制器和传感器的位掩码表现：值为 0：禁用。值 1：启用。指数：0：3D 陀螺仪，1：3D ACC，2：三维磁，3：绝对压力，差压，5：全球定位系统，6：光流，7：计算机视觉位置，8：基于激光的位置，9：外部地面真相（维科或 Leica）。传感器：10：3D 角速度管制 11：心态趋稳，12：13 偏航位置，Z /高度控制，14：X / Y 位置控制，15：电机输出/控制
onboard_control_sensors_health	uint32_t	板载控制器和传感器的位掩码，放映业务或有一个错误：值为 0：不启用。值 1：启用。指数：0：3D 陀螺仪，1：3D ACC，2：三维磁，3：绝对压力，差压，5：全球定位系统，6：光流，7：计算机视觉位置，8：基于激光的位置，9：外部地面真相（维科或 Leica）。控制器：10：3D 角速度管制 11：心态趋稳，12：13 偏航位置，Z /高度控制，14：X / Y 位置控制，15：电机输出/控制

字段名称	类型	描述
加载	uint16_t	最大使用主循环时间的百分比（0%：0， 100%：0）应该始终低于 1000
voltage_battery	uint16_t	电池电压，毫伏（1 = 1 毫伏）
current_battery	int16_t	电池的电流，在 10 毫安（1 = 10 毫安）， -1：自动驾驶仪不测量电流
battery_remaining	int8_t	剩余电池能量：（0%：0， 100%：100）， -1：自动驾驶仪，估计剩余的电池
drop_rate_comm	uint16_t	%，（0， 100%：0%10000）通信（UART 的 I2下降的各个环节上的包（包上的微型飞行器被损坏待）
errors_comm	uint16_t	通信错误（的 UART， I2C， SPI 和 CAN），下降个环节上的数据包（接收损坏的微型飞行器的数据
errors_count1	uint16_t	自动驾驶仪，具体的错误
errors_count2	uint16_t	自动驾驶仪，具体的错误
errors_count3	uint16_t	自动驾驶仪，具体的错误
errors_count4	uint16_t	自动驾驶仪，具体的错误

## SYSTEM\_TIME ( [#2](#) )

系统时间是主时钟的时间，通常是主要的车载计算机的计算机时钟。

字段名称	类型	描述
------	----	----

字段名称	类型	描述
<b>time_unix_usec</b>	uint64_t	在 UNIX 时间点中以来的微秒为单位。
<b>time_boot_ms</b>	uint32_t	自启动时间以毫秒为单位。

## PING ( [#4](#) )

一个 ping 消息请求或 ping 响应。这使得测量系统的延迟，包括串口，无线调制解调器和 UDP 连接。

字段名称	类型	描述
<b>time_usec</b>	uint64_t	Unix 时间轴（微秒）
<b>SEQ</b>	uint32_t	ping 序列
<b>target_system</b>	uint8_t	0: 请求平从所有接收系统，如果大于 0：消息是 ping 响应和数量是请求系统的系统 ID
<b>target_component</b>	uint8_t	0: 平要求所有接收组件，如果大于 0：消息是 ping 响应和数量是请求系统的系统 ID

## CHANGE\_OPERATOR\_CONTROL ( [#5](#) )

要求控制这种微型飞行器

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 GCS 的控制要求为
<b>control_request</b>	uint8_t	0: 请求控制这种微型飞行器，1：释放这种微型飞行器控制

字段名称	类型	描述
version	uint8_t	0: 纯文本文档, 1-255 在以后不同的散列/加密变种的选区一般应使用最安全的模式, 然后逐渐向下移动加密级别, 如果它得到一个 NACK 消息, 表明加密不匹配。
密钥	CHAR [25]	密码/密钥, 根据版本或加密。25 个或更少的字符, NULL 终止。可能涉及的人物包括大小写英文字母, 数字或标点符号。

## CHANGE\_OPERATOR\_CONTROL\_ACK ( #6 )

接受/拒绝这个微型飞行器的控制

字段名称	类型	描述
gcs_system_id	uint8_t	此消息在地方选区的 ID
control_request	uint8_t	0: 请求控制这种微型飞行器, 1: 释放这种微型飞行器控制
ACK	uint8_t	0 确认, 1: NACK 的: 错误的密码, 2: NACK 的: 不持密钥加密方法, 3: NACK 的: 已受控制

## AUTH\_KEY ( #7 )

发出签名/加密密钥, 确定该系统。请注意: 此协议一直保持简单, 所以需要一个真正安全的加密通道传输的关键。



字段名称	类型	描述
关键	CHAR [32]	关键

## SET\_MODE ( [# 11](#) )

设置系统模式，由枚举 `MAV_MODE` 定义。模式定义为整体的飞机，不仅是一个组件，有没有目标组件的 `ID`。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统设置模式
<b>base_mode</b>	uint8_t	新基地的模式
<b>custom_mode</b>	uint32_t	新的自动驾驶仪的具体模式。此字段由自动驾驶仪可以略不计。

## PARAM\_REQUEST\_READ ( [# 20](#) )

要求读板载的 `param_id` 字符串 `ID` 参数。板载存储的关键参数 `key[const char*] -> value[float]`。这允许发送没有任何其他组件（如地方选区）可能的参数名称前面的知识需要一个参数。因此，同一个 `GCS` 可存储不同参数，不同的自动驾驶仪。还看到一个完整的文档 `QGroundControl` 和 `IMU` 代码  
详情见 [http://qgroundcontrol.org/parameter\\_interface](http://qgroundcontrol.org/parameter_interface)。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>param_id</b>	CHAR [16]	板载 id 参数，以 NUL 终止，如果长度少于 16 人可读的符和无空终止（以 NUL）字节的长度是 16 个字符 - 应用程序必须提供 16 +1 个字节存储，如果 ID 是作为字符串

字段名称	类型	描述
		储
<b>param_index</b>	int16_t	参数指标。发送-1 使用参数 ID 字段标识符（否则将被忽略的参数 ID)

## PARAM\_REQUEST\_LIST ( [#21](#) )

要求该组件的所有参数。他的请求后，所有参数都排放。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID

## PARAM\_VALUE ( [#22](#) )

放出一个内建的参数值。消息 *param\_count* 和 *param\_index* 列入允许收件人保持跟踪接收的参数，并让他重新请求后失踪的损失或超时参数。

字段名称	类型	描述
<b>param_id</b>	CHAR [16]	板载 id 参数，以 NUL 终止，如果长度少于 16 人可读的  和无空终止（以 NUL）字节的长度是 16 个字符 - 应用程序必须提供 16 +1 个字节存储，如果 ID 是作为字符串存储
<b>param_value</b>	浮动	板载参数值
<b>param_type</b>	uint8_t	板载参数类型：看到 MAVLINK_TYPE 枚举在 mavlink  mavlink_types.h

字段名称	类型	描述
<b>param_count</b>	uint16_t	板载参数总数
<b>param_index</b>	uint16_t	本板载参数指数

## PARAM\_SET ( [# 23](#) )

暂时设置一个参数值到 *RAM*。这将重置为默认系统重新启动。发送指令 *MAV\_ACTION\_STORAGE\_WRITE* 永久性 *RAM* 的内容写入到 *EEPROM*。重要事项：接收组件应该承认所有的通讯伙伴发送 *param\_value* 消息，作为新的参数值。这也将确保多个 *GCS* 的所有参数的最新名单。如果发送 *GCS* 的超时时间内没有收到 1 *PARAM\_VALUE* 消息，它应该重新发送的 *PARAM\_SET* 的消息。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>param_id</b>	CHAR [16]	板载 id 参数，以 NUL 终止，如果长度少于 16 人可读的字符和无空终止（以 NUL）字节的长度是 16 个字符 - 应用程序必须提供 16 +1 个字节存储，如果 ID 是作为字符串存储
<b>param_value</b>	浮动	板载参数值
<b>param_type</b>	uint8_t	板载参数类型：看到 MAVLINK_TYPE 枚举在 mavlink 的 mavlink_types.h

## GPS\_RAW\_INT ( [# 24](#) )

全球定位，全球定位系统（*GPS*）返回。这并不是全球的系统正位置估计，而是一个原始传感器值。全球定位估计消息 *GLOBAL\_POSITION*。坐标系是右手系，*Z* 轴的（*GPS* 帧）。

字段名称	类型	描述
<b>time_usec</b>	uint64_t	UNIX 时间轴（微秒）
<b>fix_type</b>	uint8_t	0-1: 没有修复, 2: 2D 修复, 3: 3D 定位。一些应用程序将不使用此字段的值, 除非它至少有两个, 所以总是修补程序的正确填写。
<b>Lat</b>	int32_t	在 1E7 度的纬度
<b>Lon</b>	int32_t	在 1E7 度经度
<b>Alt</b>	int32_t	高于平均海平面高度在 1E3 米 (毫米)
<b>Eph</b>	uint16_t	GPS 的 HDOP 水平的稀释厘米 (M * 100) 的立场。如不知道, 设置为: 65535
<b>EPV</b>	uint16_t	GPS 下降的高度。如果不知道, 设置为: 65535
<b>VEL</b>	uint16_t	GPS 地面速度 (米/秒* 100)。如果不知道, 设置为: 65535
<b>COG</b>	uint16_t	比地面 (没有标题, 但运动方向) * 100 度, 0.0 .. 359.99 度。如果不知道, 设置为: 65535
<b>satellites_visible</b>	uint8_t	可见的卫星数量。如果不知道, 设置为 255

## GPS\_STATUS ( [# 25](#) )

, 通过 GPS 定位状态。此消息是为了显示每个卫星接收机可见的状态信息。全球定位估计消息 GLOBAL\_POSITION。此消息可以包含多达 20 颗卫星的信息。

字段名称	类型	描述
<b>satellites_visible</b>	uint8_t	可见的卫星数量
<b>satellite_prn</b>	uint8_t [20]	全球卫星 ID
<b>satellite_used</b>	uint8_t [20]	0: 不使用卫星, 1 : 用于本地化
<b>satellite_elevation</b>	uint8_t [20]	卫星高度 (0 : 右顶部的接收器, 90 : 在地平线上)
<b>satellite_azimuth</b>	uint8_t [20]	卫星, 0 : 0 度, 255 : 360 度的方向。
<b>satellite_snr</b>	uint8_t [20]	卫星的信号噪声比

**SCALED\_IMU ( [# 26](#) )**

通常 9DOF 传感器设置的 RAW IMU 的读数。此消息应包含描述单位的刻度值

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间戳 (系统启动以来的毫秒)
<b>xacc</b>	int16_t	X 加速 (毫克)
<b>YACC</b>	int16_t	Y 加速 (毫克)
<b>zacc</b>	int16_t	Z 加速 (毫克)
<b>xgyro</b>	int16_t	绕 X 轴的角速度 (millirad /秒)
<b>ygyro</b>	int16_t	绕 Y 轴的角速度 (millirad /秒)
<b>zgyro</b>	int16_t	绕 Z 轴角速度 (millirad /秒)
<b>xmag</b>	int16_t	X 磁场 (毫特斯拉)

字段名称	类型	描述
<b>ymag</b>	int16_t	Y 磁场（毫特斯拉）
<b>zmag</b>	int16_t	Z 磁场（毫特斯拉）

## RAW\_IMU ( [# 27](#) )

通常 9DOF 传感器设置的 RAW IMU 的读数。此消息应始终包含真正的原始值无任何缩放，允许数据采集和系统调试。

字段名称	类型	描述
<b>time_usec</b>	uint64_t	时间戳（UNIX 纪元以来的微秒或系统启动以来的微秒）
<b>xacc</b>	int16_t	X 加速
<b>yacc</b>	int16_t	Y 加速
<b>zacc</b>	int16_t	Z 加速
<b>xgyro</b>	int16_t	角速度绕 X 轴
<b>ygyro</b>	int16_t	角速度绕 Y 轴
<b>zgyro</b>	int16_t	角速度绕 Z 轴
<b>xmag</b>	int16_t	X 磁场
<b>ymag</b>	int16_t	Y 磁场
<b>zmag</b>	int16_t	Z 磁场

## RAW\_PRESSURE ( [# 28](#) )

RAW 的典型设置一个绝对压力和差压传感器的压力读数。传感器的值应该是原始值，非标度的 ADC 值。

字段名称	类型	描述
<b>time_usec</b>	uint64_t	时间戳（UNIX 时间点的微秒或系统启动以来的微秒）
<b>press_abs</b>	int16_t	绝对压力
<b>press_diff1</b>	int16_t	差压 1
<b>press_diff2</b>	int16_t	差压 2
<b>temperature</b>	int16_t	原料温度测量

SCALED\_PRESSURE ( [# 29](#) )

典型设置一个绝对和压差传感器的压力读数。在每个字段指定单位。

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间戳（系统启动以来的毫秒）
<b>press_abs</b>	Float	绝对压力（百帕）
<b>press_diff</b>	Float	差压 1（百帕）
<b>temperature</b>	int16_t	温度测量（0.01 摄氏度）

态度 ( [# 30](#) )

在航空帧的态度（右手，Z 轴，X-前，Y 右）。

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间戳（系统启动以来的毫秒）
<b>滚动</b>	Float	侧倾角（RAD）

字段名称	类型	描述
音高	Float	俯仰角（RAD）
偏航	Float	偏航角（RAD）
rollspeed	Float	滚动的角速度（弧度/秒）
pitchspeed	Float	俯仰角速度（弧度/秒）
yawspeed	浮动	偏航角速度（弧度/秒）

## ATTITUDE\_QUATERNION ( [#31](#) )

航空帧的态度（右手，Z 轴，X-前，Y 右），表示为四元。

字段名称	类型	描述
time_boot_ms	uint32_t	时间轴（系统启动以来的毫秒）
Q1	Float	四元组件 1
Q2	Float	四元组件 2
Q3	Float	四元组件 3
Q4	Float	四元组件 4
rollspeed	Float	轧辊的角速度（弧度/秒）
pitchspeed	Float	俯仰角速度（弧度/秒）
yawspeed	Float	偏航角速度（弧度/秒）

## LOCAL\_POSITION\_NED ( [#32](#) )



过滤后的本地位置（例如，融合计算机视觉和加速度计）。坐标系是右手，Z 轴下降（航空帧，非执行董事/东北向下公约）

字段名称	类型	描述
time_boot_ms	uint32_t	时间戳（系统启动以来的毫秒）
x	Float	X 位置
y	Float	Y 位置
z	Float	Z 位置
vx	Float	X 速度
vy	Float	Y 速度
vz	Float	Z 速度

## GLOBAL\_POSITION\_INT ( [#33](#) )

过滤后的全球定位（如融合 GPS 和加速度计）。位置是在 GPS 帧（右手，Z）。它被设计为规模整数消息以来，浮法决议是不够的。

字段名称	类型	描述
time_boot_ms	uint32_t	时间戳（系统启动以来的毫秒）
Lat	int32_t	纬度，表示为* 1E7
LON	int32_t	经度，表示为* 1E7
ALT	int32_t	用* 1000（毫米）表示，在万米高空，高于海平面
relative_alt	int32_t	米，地面以上的高度，表示为* 1000（毫米）

字段名称	类型	描述
<b>VX</b>	int16_t	地面 X 速度（纬度）， 表示为米/秒* 100
<b>VY</b>	int16_t	地面 Y 高速（经度）， 表示为米/秒* 100
<b>VZ</b>	int16_t	地面， Z 高速（海拔高度）， 表示为米/秒* 100
<b>HDG</b>	uint16_t	罗盘航向度数* 100， 0.0 .. 359.99 度。如果不知道， 设为： 65535

RC\_CHANNELS\_SCALED ( [# 34](#) )

收到 RC 通道的规模值。 -10000 (-100%) ， 0 (0%) ， (100%) 10000

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间轴（系统启动以来的毫秒）
<b>Port</b>	uint8_t	伺服输出端口（8 个输出= 1 端口设置）。最飞行器将5 用一个， 但允许超过 8 伺服编码。
<b>chan1_scaled</b>	int16_t	RC 通道值为 1， 缩放， （-100%） -10000（0%） 0 （100%） 10000
<b>chan2_scaled</b>	int16_t	RC 通道缩放 2 值， （-100%） -10000（0%） 0 （100%） 10000
<b>chan3_scaled</b>	int16_t	RC 通道值 3 缩放， （-100%） -10000（0%） 0（ 100%） 10000
<b>chan4_scaled</b>	int16_t	RC 通道值 4 缩放， （-100%） -10000（0%） 0（100%

字段名称	类型	描述
chan5_scaled	int16_t	RC 通道价值尺度（-100%） -10000（0%） 0（100%） 10000
chan6_scaled	int16_t	RC 通道值 6 缩放，（-100%） -10000（0%） 0（100%） 10000
chan7_scaled	int16_t	RC 通道值 7 缩放，（-100%） -10000（0%） 0（100%） 10000
chan8_scaled	int16_t	RC 通道值 8 缩放，（-100%） -10000（0%） 0（100%） 10000
RSSI	uint8_t	接收信号强度指示器， 0：0%， 255：100%

## RC\_CHANNELS\_RAW ( #35 )

收到 RC 通道的原始值。标准 PPM 调制如下：1000 微秒：0%，2000 微秒：100%。个别接收器/发射器可能违反本规范。

字段名称	类型	描述
time_boot_ms	uint32_t	时间戳（系统启动以来的毫秒）
Port	uint8_t	伺服输出端口（8 个输出= 1 端口设置）。最飞行器将只使用一个，但允许超过 8 伺服编码。
chan1_raw	uint16_t	RC 通道值为 1，微秒
chan2_raw	uint16_t	RC 通道 2 的值，微秒
chan3_raw	uint16_t	RC 通道值 3，微秒

字段名称	类型	描述
chan4_raw	uint16_t	RC 通道值为 4，微秒
chan5_raw	uint16_t	RC 通道值为 5，微秒
chan6_raw	uint16_t	RC 通道值 6，微秒
chan7_raw	uint16_t	RC 通道值为 7，微秒
chan8_raw	uint16_t	RC 通道值 8，微秒
RSSI	uint8_t	接收信号强度指示器，0：0%，255：100%

## SERVO\_OUTPUT\_RAW ( [# 36](#) )

伺服输出的 RAW 值（从远程遥控输入，使用 RC\_CHANNELS“消息）。标准 PPM 调制如下：1000 微秒：0%，2000 微秒：100%。

字段名称	类型	描述
time_usec	uint32_t	时间轴（系统启动以来的微秒）
端口	uint8_t	伺服输出端口（8 个输出= 1 端口设置）。最飞行器将只使用一个，但允许超过 8 伺服编码。
servo1_raw	uint16_t	1 伺服输出值，微秒
servo2_raw	uint16_t	伺服电机的输出值，微秒
servo3_raw	uint16_t	3 伺服输出值，微秒
servo4_raw	uint16_t	4 伺服输出值，微秒
servo5_raw	uint16_t	5 伺服输出值，微秒

字段名称	类型	描述
<b>servo6_raw</b>	uint16_t	6 伺服输出值，微秒
<b>servo7_raw</b>	uint16_t	7 伺服输出值，微秒
<b>servo8_raw</b>	uint16_t	8 伺服输出值，微秒

## MISSION\_REQUEST\_PARTIAL\_LIST ( [#37](#) )

要求整体系统/组件的任务列表。[http://qgroundcontrol.org/mavlink/waypoint\\_protocol](http://qgroundcontrol.org/mavlink/waypoint_protocol)

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>start_index</b>	int16_t	起始索引，默认为 0
<b>end_index</b>	int16_t	结束索引，默认-1（-1：发送列表结束）。其他有效的索引列表

## MISSION\_WRITE\_PARTIAL\_LIST ( [#38](#) )

此消息被发送到了微型飞行器，写一个部分列表。如果起始索引==结束索引，只有一个项目将被发送/更新。如果起始索引是不为 0 及以上的当前列表的大小，这一要求应予以拒绝！

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID

字段名称	类型	描述
<b>start_index</b>	int16_t	启动默认情况下，小于/等于目前板载列表的最大指数的数，0 表示故障
<b>end_index</b>	int16_t	结束指数，等于或大于起始的指数。

## MISSION\_ITEM ( [# 39](#) )

为信息编码的使命项目。此消息发出后宣布使命项目的存在，并在系统上设置的使命项目。任务项目可以是在 X, Y, Z 米（类型：本地）或 x: 纬度, Y: LON, Z: 海拔高度。本地帧的 Z-下来，右手（NED），全球框架是 Z 轴，右手（ENU 的）。也见 [http://qgroundcontrol.org/mavlink/waypoint\\_protocol](http://qgroundcontrol.org/mavlink/waypoint_protocol)。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>SEQ</b>	uint16_t	序列
<b>Frame</b>	uint8_t	该 MISSION 的坐标系统。mavlink_types.h MAV_FRA
<b>Command</b>	uint16_t	为使命的行动计划。见 common.xml MAVLink 规格 MAV_CMD
<b>Current</b>	uint8_t	假：0，真实：1
<b>Autocontinue</b>	uint8_t	自动继续到下一个 WP
<b>PARAM1</b>	Float	PARAM1 /净值命令任务：MISSION 在被接受为达到在 米的半径,

字段名称	类型	描述
<b>param2 的</b>	Float	/ param2 的净值命令任务：前推进 MAV 的 应留 PARAM1 半径内的时间（以毫秒为单位），
<b>param 3</b>	Float	参数 3 /对于命令的任务：在半球周围圈的轨道。 如果积极的轨道应顺时针方向，如果负的轨道方向应该 逆时针。
<b>param4</b>	浮动	PARAM4 / NAV 和游荡命令任务：偏航度的方向， [0 到 360] 0 =向北
<b>x</b>	<b>Float</b>	PARAM5 /地方：X 位置，全球：纬度
<b>Y</b>	<b>Float</b>	PARAM6 / Y 位置：全球：经度
<b>Z</b>	Float	PARAM7 / z 位置：全球：海拔

## MISSION\_REQUEST ( [# 40](#) )

任务项要求的序列号 *SEQ* 的信息。系统响应此消息应该是 *MISSION\_ITEM* 的消息。  
[http://qgroundcontrol.org/mavlink/waypoint\\_protocol](http://qgroundcontrol.org/mavlink/waypoint_protocol)

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>SEQ</b>	uint16_t	序列

## MISSION\_SET\_CURRENT ( [# 41](#) )

设置作为当前项目的数量和序列 *SEQ* 任务项目。这意味着，微型飞行器将继续这一使命项目上的最短路径（不遵守之间的任务物品）。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>SEQ</b>	uint16_t	序列

## MISSION\_CURRENT ( [# 42](#) )

消息宣布的当前活动任务项目的序列号。微型飞行器将飞向这一使命的项目。

字段名称	类型	描述
<b>SEQ</b>	uint16_t	序列

## MISSION\_REQUEST\_LIST ( [# 43](#) )

要求从整体任务物品清单系统/组件。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID

## MISSION\_COUNT ( [# 44](#) )

此消息被发射作为响应，到 MAV 的 *MISSION\_REQUEST\_LIST*，开始写交易。各选区的，可以要求个别特派团的任务总数的知识基础上的项目。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID



字段名称	类型	描述
<b>target_component</b>	uint8_t	组件的 ID
<b>Seq</b>	uint16_t	序列中的任务的数量

## MISSION\_CLEAR\_ALL ( [#45](#) )

一次删除所有任务物品。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID

## MISSION\_ITEM\_REACHED ( [#46](#) )

某任务项目已经达到了。该系统将持有这一立场（或循环的轨道上）或（如果成立的 WP 自动继续）继续下一个任务。

字段名称	类型	描述
<b>SEQ</b>	uint16_t	序列

## MISSION\_ACK ( [#47](#) )

任务处理过程中的 ACK 消息。类型字段的状态，如果这个消息是一个积极的 ACK(类型=0)，或者如果发生错误（类型=非零）。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID

字段名称	类型	描述
Count	uint8_t	MAV_MISSION_RESULT 枚举

## SET\_GPS\_GLOBAL\_ORIGIN ( [# 48](#) )

由于本地特派团的存在，全球使命参考允许局部坐标系和全球（GPS）坐标系之间的转换。  
这可能是必要的，例如在户外设置连接和微型飞行器在户外移动时。

字段名称	类型	描述
target_system	uint8_t	系统 ID
latitude	int32_t	全球定位* 1E7
longitude	int32_t	全球定位* 1E7
altitude	int32_t	全球定位* 1000

## GPS\_GLOBAL\_ORIGIN ( [# 49](#) )

一旦 MAV 的设置一个新的 GPS 本地通信，此消息宣布的原点（0,0,0）位置

字段名称	类型	描述
latitude	int32_t	* 1E7 表示纬度（WGS84）。
longitude	int32_t	经度（WGS84 坐标），表示为* 1E7
altitude	int32_t	海拔高度。（WGS84），表示为* 1000

# SET\_LOCAL\_POSITION\_SETPOINT ( [# 50](#) )

设置为本地位置控制器的设定值。这是在局部坐标系的位置，*MAV* 的应飞。此消息发送路径/任务计划的板载位置控制器。正如一些小牛所需的偏航角偏航自由（例如所有直升机/*quadrotors* 的“）的程度，是消息的一部分。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>coordinate_frame</b>	uint8_t	坐标系 - 有效值只 MAV_FRAME_LOCAL_NED 的, 或 MAV_FRAME_LOCAL_ENU
<b>x</b>	Float	x 位置
<b>y</b>		y 位置
<b>z</b>	Float	z 位置
<b>Yaw</b>	Float	所需的偏航角

# LOCAL\_POSITION\_SETPOINT ( [# 51](#) )

其他飞行器（防撞）和 *GCS* 的控制器，传输目前的本地设定点。

字段名称	类型	描述
<b>coordinate_frame</b>	uint8_t	坐标系 - 有效值只 MAV_FRAME_LOCAL_NED 的, 或 MAV_FRAME_LOCAL_ENU
<b>x</b>	Float	x 位置
<b>y</b>	Float	y 位置

字段名称	类型	描述
<b>Z</b>	Float	z 位置
<b>Yaw</b>	Float	所需的偏航角

## GLOBAL\_POSITION\_SETPOINT\_INT ( [#52](#) )

其他飞行器（防撞）和 GCS 的控制器，传输目前的本地设定点。

字段名称	类型	描述
<b>coordinate_frame</b>	uint8_t	协调帧 - 有效值只有 MAV_FRAME_GLOBAL 或 MAV_FRAME_GLOBAL_RELATIVE_ALT 的该
<b>Latitude</b>	int32_t	WGS84 北纬度* 1E7
<b>Longitude</b>	int32_t	WGS84 经度位置度* 1E7
<b>Altitude</b>	int32_t	WGS84 的海拔高度在米* 1000（正达）
<b>偏航</b>	int16_t	* 100 度所需的偏航角

## SET\_GLOBAL\_POSITION\_SETPOINT\_INT ( [#53](#) )

设置当前的全球定位设定值。

字段名称	类型	描述
<b>coordinate_frame</b>	uint8_t	协调帧 - 有效值只有 MAV_FRAME_GLOBAL 或 MAV_FRAME_GLOBAL_RELATIVE_ALT
<b>Latitude</b>	int32_t	WGS84 北纬度* 1E7
<b>Longitude</b>	int32_t	WGS84 经度位置度* 1E7

字段名称	类型	描述
<b>Altitude</b>	int32_t	WGS84 的海拔高度在米* 1000（正达）
<b>Yaw</b>	int16_t	* 100 度所需的偏航角

## SAFETY\_SET\_ALLOWED\_AREA ( [#54](#) )

设置一个安全区（卷），这是由两个立方体的角落。可以使用这条消息告诉 MAV 的接受和拒绝设定值/任务。安全领域往往是由国家或竞争法规的执行。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>Frame</b>	uint8_t	坐标系，由枚举 MAV_FRAME 在 mavlink_types.h 定义 无论是全球，全球定位系统，Z 轴或地方，右手，Z 轴的右手。
<b>p1x</b>	Float	x 位置 1 /纬度 1
<b>P1Y</b>	Float	y 位置 1 /经度 1
<b>p1z</b>	Float	Z 位置 1 /海拔 1
<b>P2X</b>	Float	x 位置 2/2 纬度
<b>P2Y</b>	Float	y 位置 2/2 经度
<b>P2Z</b>	Float	z 位置 2 /海拔 2

## SAFETY\_ALLOWED\_AREA ( [#55](#) )

读出目前承担的微型飞行器的安全区。

字段名称	类型	描述
Frame	uint8_t	坐标系，由枚举 MAV_FRAME 在 mavlink_types.h 定义。 无论是全球，全球定位系统，Z 轴或地方，右手，Z 轴的右手。
p1x	浮动	x 位置 1 /纬度 1
P1Y	Float	y 位置 1 /经度 1
p1z	Float	Z 位置 1 /海拔 1
P2X	Float	x 位置 2/2 纬度
P2Y	Float	y 位置 2/2 经度
P2Z	Float	z 位置 2 /海拔 2

## SET\_ROLL\_PITCH\_YAW\_THRUST ( # 56 )

设置横滚，俯仰和偏航。

字段名称	类型	描述
target_system	uint8_t	系统 ID
target_component	uint8_t	组件的 ID
Roll	Float	所需的侧倾角（弧度）
Pitch	Float	所需的俯仰角（弧度）
Yaw	Float	以弧度所需的偏航角

字段名称	类型	描述
Thrust	Float	积累的推力

## SET\_ROLL\_PITCH\_YAW\_SPEED\_THRUST ( [# 57](#) )

设置横滚，俯仰和偏航。

字段名称	类型	描述
target_system	uint8_t	系统 ID
target_component	uint8_t	组件的 ID
roll_speed	Float	所需的轧辊 rad / s
pitch_speed	Float	所需的角速度 rad / s
yaw_speed	Float	所需的偏航角速度 rad / s
Thrust	Float	积累的力

## ROLL\_PITCH\_YAW\_THRUST\_SETPOINT ( [# 58](#) )

在目前活跃系统设定值横滚，俯仰，偏航。

字段名称	类型	描述
time_boot_ms	uint32_t	在系统启动以来的毫秒时间戳
Roll	Float	所需的侧倾（弧度）
Pitch	Float	所需的俯仰角（弧度）
Yaw	Float	以弧度所需的偏航角

字段名称	类型	描述
Thrust	Float	积累的力

## ROLL\_PITCH\_YAW\_SPEED\_THRUST\_SETPOINT ( [# 59](#) )

在目前活跃系统上设定在 *rollspeed*, *pitchspeed*, *yawspeed*。

字段名称	类型	描述
time_boot_ms	uint32_t	在系统启动以来的毫秒时间戳
roll_speed	Float	所需的轧辊 rad / s
pitch_speed	Float	所需的角速度 rad / s
yaw_speed	Float	所需的偏航角速度 rad / s
Thrust	Float	积累的力

## SET\_QUAD\_MOTORS\_SETPOINT ( [# 60](#) )

在四个电机转速设定

字段名称	类型	描述
target_system	uint8_t	系统 ID 系统应该设置这些电机命令
motor_front_nw	uint16_t	前方用 + 配置, 左前方用 x 配置
motor_right_ne	uint16_t	右方用 + 配置, 右前方用 x 配置
motor_back_se	uint16_t	后方用 + 配置, 右后方用 x 配置



字段名称	类型	描述
<b>motor_left_sw</b>	uint16_t	左方用 + 配置, 左后方用 x 配置

## SET\_QUAD\_SWARM\_ROLL\_PITCH\_YAW\_THRUST ( [#61](#) )

设定点多达 4 组四旋翼

字段名称	类型	描述
<b>Group</b>	uint8_t	旋翼组 ID ( 0 - 255 , 256 组支持)
<b>Mode</b>	uint8_t	飞行模式 (0 - 255, 最高支持 256 模式的 ID)
<b>Roll</b>	int16_t [4]	所需的侧倾角 (弧度) + -32767)
<b>Pitch</b>	int16_t [4]	所需的俯仰角 (弧度) + (-32767)
<b>Yaw</b>	int16_t [4]	所需的偏航弧度的角度, 缩放到 INT16 + (PI + -32767
<b>Thrust</b>	uint16_t [4]	集体推力, 缩放到 UINT16 (0 到 65535)

## NAV\_CONTROLLER\_OUTPUT ( [#62](#) )

APM 导航控制器的输出。此消息的主要用途是检查前实际飞行控制器的响应和迹象, 并协助调整控制器参数。

字段名称	类型	描述
<b>nav_roll</b>	Float	目前所需的滚动角度
<b>nav_pitch</b>	Float	目前所需的间距度数
<b>nav_bearing</b>	int16_t	目前所需的标定度数

字段名称	类型	描述
target_bearing	int16_t	轴承目前的任务
wp_dist	uint16_t	距离
alt_error	Float	当前高度误差（米）
aspd_error	Float	当前的空速误差（米/秒）
xtrack_error	Float	xy 平面上的错误（米）

## SET\_QUAD\_SWARM\_LED\_ROLL\_PITCH\_YAW\_THRUST ( [# 63](#) )

设定点多达 4 组四旋翼

字段名称	类型	描述
Group	uint8_t	旋翼组 ID（0 - 255，256 组支持）
Mode	uint8_t	飞行模式（0 - 255，最高支持 256 模式的 ID）
led_red	uint8_t [4]	红色 RGB（0255）
led_blue	uint8_t [4]	绿色 RGB（0255）
led_green	uint8_t [4]	蓝色 RGB（0255）
Roll	int16_t [4]	所需的侧倾角（弧度）+（-32767）
Pitch	int16_t [4]	所需的俯仰角（弧度）+（-32767）
Yaw	int16_t [4]	所需的偏航弧度的角度，缩放到 INT16 +（PI + -32767）

字段名称	类型	描述
Thrust	uint16_t [4]	积累的力，缩放到 UINT16（0 到 65535）

## STATE\_CORRECTION ( [# 64](#) )

纠正系统的状态，加入误差修正项的位置和速度，并通过旋转校正角的态度。

字段名称	类型	描述
xErr	Float	x 位置错误
yErr	Float	y 位置错误
zErr	Float	z 位置误差
rollErr	Float	辊错误（弧度）
pitchErr	Float	螺距误差（弧度）
yawErr	Float	偏航误差（弧度）
vxErr	Float	x 速度
vyErr	Float	Y 速度
vzErr	Float	Z 速度

## REQUEST\_DATA\_STREAM ( [# 66](#) )

字段名称	类型	描述
target_system	uint8_t	目标要求发送的消息。
target_component	uint8_t	目标要求发送的消息。

字段名称	类型	描述
<b>req_stream_id</b>	uint8_t	所要求的数据流的 ID
<b>req_message_rate</b>	uint16_t	要求间隔之间的这种类型的两个消息
<b>START_STOP</b>	uint8_t	1 开始发送, 0 停止发送。

## DATA\_STREAM ( [#67](#) )

字段名称	类型	描述
<b>stream_id</b>	uint8_t	所要求的数据流的 ID
<b>message_rate</b>	uint16_t	要求间隔之间的这种类型的两个消息
<b>ON_OFF</b>	uint8_t	1 启用流, 0 停止。

## MANUAL\_CONTROL ( [#69](#) )

字段名称	类型	描述
<b>Target</b>	uint8_t	要控制系统
<b>Roll</b>	Float	滚动
<b>Pitch</b>	Float	倾斜
<b>Yaw</b>	Float	偏航
<b>Thrust</b>	Float	推力
<b>roll_manual</b>	uint8_t	滚动控制自动启用：0, 手动：1
<b>pitch_manual</b>	uint8_t	自动：0，手动：1

字段名称	类型	描述
<b>yaw_manual</b>	uint8_t	自动：0，手动：1
<b>thrust_manual</b>	uint8_t	自动：0，手动：1

## RC\_CHANNELS\_OVERRIDE ( [#70](#) )

RC 通道的原始值发送到 MAV 的覆盖从 RC 无线电接收的信息。-1 意味着该通道没有改变。应释放回值 0 手段控制该通道的 RC 电台。标准 PPM 调制如下：1000 微秒：0%，2000 微秒：100%。个别接收器/发射器可能违反本规范。

字段名称	类型	描述
<b>target_system</b>	uint8_t	系统 ID
<b>target_component</b>	uint8_t	组件的 ID
<b>chan1_raw</b>	uint16_t	RC 通道值为 1，微秒
<b>chan2_raw</b>	uint16_t	RC 通道 2 的值，微秒
<b>chan3_raw</b>	uint16_t	RC 通道值 3，微秒
<b>chan4_raw</b>	uint16_t	RC 通道值为 4，微秒
<b>chan5_raw</b>	uint16_t	RC 通道值为 5，微秒
<b>chan6_raw</b>	uint16_t	RC 通道值 6，微秒
<b>chan7_raw</b>	uint16_t	RC 通道值为 7，微秒
<b>chan8_raw</b>	uint16_t	RC 通道值 8，微秒

## VFR\_HUD ( [#74](#) )

指标通常显示在 HUD 固定翼飞机

字段名称	类型	描述
Airspeed	Float	米/秒， 在当前空速
Groundspeed	Float	目前地面速度 m / s
Heading	int16_t	当前的方向以度为，在指南针单位， （0 到 360, 0 =>
Throttle	uint16_t	当前油门设置（0%到 100%）
ALT	Float	当前高度（MSL）， 米
Climb	Float	在当前米/秒爬升率

## COMMAND\_LONG ( [#76](#) )

发送命令到四个参数的微型飞行器

字段名称	类型	描述
target_system	uint8_t	系统应执行的命令
target_component	uint8_t	组件， 它应该为所有组件执行的命令， 0
Command	uint16_t	命令 ID， MAV_CMD 枚举定义。
Confirmation	uint8_t	0: 这个命令的第一次传输。1-255：确认传输（例如 k 命令）
PARAM1	Float	1, 参数定义 MAV_CMD 枚举。
param2	Float	2, 参数定义枚举 M AV_CMD。
Param3	Float	3, 参数定义枚举 MAV_CMD。

字段名称	类型	描述
param4	Float	4, 参数定义枚举 MAV_CMD。
param5	Float	5, 参数定义枚举 MAV_CMD。
param6	Float	6, 参数定义枚举 MAV_CMD。
param7	Float	7, 参数定义枚举 MAV_CMD。

## COMMAND\_ACK ( [#77](#) )

报告状态的命令。包括反馈任务被执行死刑的命令。

字段名称	类型	描述
Command	uint16_t	命令 ID, MAV_CMD 枚举定义。
Result	uint8_t	MAV_RESULT 枚举

## LOCAL\_POSITION\_NED\_SYSTEM\_GLOBAL\_OFFSET ( [#89](#) )

偏移，在 X, Y, Z 和偏航 MAV 的 X 的 LOCAL\_POSITION\_NED 消息，并在 NED 的坐标的全球坐标系之间。坐标系是右手，Z 轴下降（航空帧，非执行董事/东北向下公约）

字段名称	类型	描述
time_boot_ms	uint32_t	时间戳（系统启动以来的毫秒）
x	Float	X 位置
y	Float	Y 位置
z	Float	Z 位置

字段名称	类型	描述
Roll	Float	滚动
Pitch	Float	音高
Yaw	Float	偏航

## HIL\_STATE ( [# 90](#) )

从模拟到自动驾驶仪发送。这个包是 useful 的高吞吐量的应用，如硬件在环仿真。

字段名称	类型	描述
time_usec	uint64_t	时间戳（UNIX 纪元以来的微秒或系统启动以来的微秒）
Roll	Float	侧倾角（RAD）
Piych	Float	俯仰角（RAD）
Yaw	Float	偏航角（RAD）
rollspeed	Float	轧辊的角速度（弧度/秒）
pitchspeed	Float	俯仰角速度（弧度/秒）
yawspeed	Float	偏航角速度（弧度/秒）
Lat	int32_t	纬度，表示* 1E7
LON	int32_t	经度表示为* 1E7
ALT	int32_t	海拔米，表示为* 1000（毫米）
VX	int16_t	地面 X 速度（纬度），表示为米/秒* 100



字段名称	类型	描述
VY	int16_t	地面 Y 高速（经度），表示为米/秒* 100
VZ	int16_t	地面, Z 高速（海拔高度），表示为米/秒* 100
xacc	int16_t	X 加速（毫克）
YACC	int16_t	Y 加速（毫克）
zacc	int16_t	Z 加速（毫克）

## HIL\_CONTROLS ( [#91](#) )

从自动驾驶仪发送到模拟。硬件在回路控制输出

字段名称	类型	描述
time_usec	uint64_t	时间戳（UNIX 纪元以来的微秒或系统启动以来的微秒）
roll_ailerons	Float	控制输出-1 .. 1
pitch_elevator	Float	控制输出-1 .. 1
yaw_rudder	Float	控制输出-1 .. 1
Throttle	Float	油门 0 ..... 1
AUX1	Float	AUX 1, -1 ..... 1
AUX2	Float	AUX 2, -1 ..... 1
AUX3	Float	辅助 3, -1 ..... 1
AUX4	Float	辅助 4, -1 ..... 1

字段名称	类型	描述
Mode	uint8_t	系统模式 (MAV_MODE)
nav_mode	uint8_t	导航模式 (MAV_NAV_MODE)

## HIL\_RC\_INPUTS\_RAW ( [#92](#) )

从模拟到自动驾驶仪发送。收到 RC 通道的原始值。标准 PPM 调制如下：1000 微秒：0%，2000 微秒：100%。个别接收器/发射器可能违反本规范。

字段名称	类型	描述
time_usec	uint64_t	时间轴 (UNIX 时间点 (微秒) 或系统启动以来 (微秒))
chan1_raw	uint16_t	RC 通道值为 1，微秒
chan2_raw	uint16_t	RC 通道 2 的值，微秒
chan3_raw	uint16_t	RC 通道值 3，微秒
chan4_raw	uint16_t	RC 通道值为 4，微秒
chan5_raw	uint16_t	RC 通道值为 5，微秒
chan6_raw	uint16_t	RC 通道值 6，微秒
chan7_raw	uint16_t	RC 通道值为 7，微秒
chan8_raw	uint16_t	RC 通道值 8，微秒
chan9_raw	uint16_t	RC 通道值 9，微秒
chan10_raw	uint16_t	RC 通道值为 10，微秒
chan11_raw	uint16_t	RC 通道值 11，微秒

字段名称	类型	描述
chan12_raw	uint16_t	RC 通道值为 12，微秒
RSSI	uint8_t	接收信号强度指示器， 0：0%， 255：100%

## OPTICAL\_FLOW ( [#100](#) )

流量传感器（如光学鼠标传感器的光流）

字段名称	类型	描述
time_usec	uint64_t	时间轴（UNIX）
sensor_id	uint8_t	传感器 ID
flow_x	int16_t	流在 X 负半轴传感器方向的像素
flow_y	int16_t	在 Y 负轴传感器方向的像素流程
flow_comp_m_x	Float	在米 X 负轴传感器的方向流动，角速补偿
flow_comp_m_y	Float	在 y（米）， 流量传感器方向，角速补偿
Quality	uint8_t	光流的质量。0：坏了， 255：最高质量
ground_distance	Float	地面的距离（米）。正值：已知的距离。负值：未知的距离

## GLOBAL\_VISION\_POSITION\_ESTIMATE ( [#101](#) )

字段名称	类型	描述
USEC	uint64_t	时间戳（毫秒）

字段名称	类型	描述
<b>x</b>	<b>Float</b>	全球的 X 位置
<b>Y</b>	<b>Float</b>	全球 Y 位置
<b>Z</b>	<b>Float</b>	全球 Z 位置
<b>Roll</b>	<b>Float</b>	在 RAD 中的侧倾角
<b>Pitch</b>	<b>Float</b>	在 RAD 的俯仰角
<b>Yaw</b>	<b>Float</b>	在 RAD 偏航角

**VISION\_POSITION\_ESTIMATE ( [# 102](#) )**

字段名称	类型	描述
<b>USEC</b>	uint64_t	时间戳（毫秒）
<b>x</b>	<b>Float</b>	全球的 X 位置
<b>Y</b>	<b>Float</b>	全球 Y 位置
<b>Z</b>	<b>Float</b>	全球 Z 位置
<b>Roll</b>	<b>Float</b>	在 RAD 中的侧倾角
<b>Pitch</b>	<b>Float</b>	在 RAD 的俯仰角
<b>Yaw</b>	<b>Float</b>	在 RAD 偏航角

**VISION\_SPEED\_ESTIMATE ( [# 103](#) )**

字段名称	类型	描述
------	----	----

字段名称	类型	描述
<b>USEC</b>	uint64_t	时间轴（毫秒）
<b>x</b>	<b>Float</b>	全球 X 速度
<b>Y</b>	<b>Float</b>	全局 y 速度
<b>Z</b>	<b>Float</b>	全球 z 速度

**VICON\_POSITION\_ESTIMATE ( [# 104](#) )**

字段名称	类型	描述
<b>USEC</b>	uint64_t	时间戳（毫秒）
<b>x</b>	<b>Float</b>	全球的 X 位置
<b>Y</b>	<b>Float</b>	全球 Y 位置
<b>Z</b>	<b>Float</b>	全球 Z 位置
<b>Roll</b>	<b>Float</b>	侧倾角（RAD）
<b>Pitch</b>	<b>Float</b>	俯仰角（RAD）
<b>Yaw</b>	<b>Float</b>	偏航角（RAD）

**MEMORY\_VECT ( [# 249](#) )**

发送原始控制器内存。此消息不鼓励使用正常的数据包，但相当有效的方式，为测试新的消息和实验调试输出。

字段名称	类型	描述
<b>Address</b>	uint16_t	调试变量的起始地址

字段名称	类型	描述
VER	uint8_t	版本代码的类型变量。0 =未知， 类型上被忽略， 并假设
		int16_t.1
Type	uint8_t	键入代码的内存变量。VER = 1 : 0 = 16×1 int16_t,
		uint16_t, = 16×2 = 16×Q15, 3 = 16×1Q14
Value	int8_t [32]	在指定地址的存储器中的内容

DEBUG\_VECT ( [# 250](#) )

字段名称	类型	描述
名称	CHAR [10]	名称
time_usec	uint64_t	时间戳
x	Float	x
Y	Float	Y
Z	Float	Z

NAMED\_VALUE\_FLOAT ( [# 251](#) )

作为浮动发送键。此消息不鼓励使用正常的数据包，但相当有效的方式，为测试新的消息和实验调试输出。

字段名称	类型	描述
time_boot_ms	uint32_t	时间轴（系统启动（毫秒））
Name	CHAR [10]	调试变量的名称

字段名称	类型	描述
<b>Value</b>	Float	浮点值

## NAMED\_VALUE\_INT ( [# 252](#) )

发送整数。此消息不鼓励使用正常的数据包，但相当有效的方式，为测试新的消息和实验调试输出。

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间轴（系统启动（毫秒））
<b>Name</b>	CHAR [10]	调试变量的名称
<b>Value</b>	int32_t	符号的整数值

## STATUSTEXT ( [# 253](#) )

状态的文本消息。这些信息印在黄色 *COMM* 中的 *QGroundControl* 的控制台。警告：他们消耗相当长的一段带宽，所以只用于重要的状态和错误消息。如果明智地使用，这些消息 *MCU* 上的缓冲和发送只在有限的速度（如 10 赫兹）。

字段名称	类型	描述
<b>Severity</b>	uint8_t	状态的严重性。依赖于定义在 RFC-5424。参见枚举 MAV_SEVERITY。
<b>Text</b>	CHAR [50]	状态的文本消息，无空终止字符

## 调试 ( [# 254](#) )

发送调试的数值。该指数是用来区分值。这些值显示作为调试 *N.* 中的 *QGroundControl* 的阴谋

字段名称	类型	描述
------	----	----

字段名称	类型	描述
<b>time_boot_ms</b>	uint32_t	时间轴（系统启动（毫秒））
<b>IND</b>	uint8_t	调试变量指数
<b>Value</b>	<b>Float</b>	调试值