

# 匿名科创

## 匿名通信协议

**V8.10**

**2024.09.22**

目录

- 一、匿名通信协议介绍..... 4
  - 1. 通信帧格式介绍 ..... 4
  - 2. 匿名安全通信协议..... 6
    - 1) ID: 0x00: 数据校验帧..... 6
  - 3. 灵活格式帧..... 7
    - 1) ID: 0xF1~0xFA: 灵活格式帧..... 7
- 二、基本信息类帧..... 8
  - 1. 飞控相关信息类..... 8
    - 1) ID: 0x01: 惯性传感器数据..... 8
    - 2) ID: 0x02: 罗盘、气压、温度传感器数据..... 8
    - 3) ID: 0x03: 飞控姿态: 欧拉角格式..... 8
    - 4) ID: 0x05: 高度数据..... 8
    - 5) ID: 0x06: 飞控运行模式..... 9
    - 6) ID: 0x07: 飞行速度数据..... 9
    - 7) ID: 0x08: 位置偏移数据..... 9
    - 8) ID: 0x09: 风速估计..... 9
    - 9) ID: 0x0A: 目标姿态数据..... 9
    - 10) ID: 0x0B: 目标速度数据..... 9
    - 11) ID: 0x0C: 回航信息..... 9
    - 12) ID: 0x0D: 电压电流数据..... 10
    - 13) ID: 0x0E: 外接模块工作状态..... 10
    - 14) ID: 0x0F: RGB 亮度信息输出..... 10
  - 2. 飞控控制量输出类..... 11
    - 1) ID: 0x20: PWM 控制量..... 11
    - 2) ID: 0x21: 姿态控制量..... 11
  - 3. 飞控接收信息类..... 12
    - 1) ID: 0x30: GPS 传感器信息 1..... 12
    - 2) ID: 0x31: 原始光流信息..... 12
    - 3) ID: 0x32: 通用位置型传感器数据 (非捷联载体测量型)..... 12
    - 4) ID: 0x33: 通用速度型传感器数据 (捷联载体测量型)..... 12
    - 5) ID: 0x34: 通用测距传感器数据 (捷联载体测量型)..... 12
    - 6) ID: 0x35: 通用图像特征点信息帧..... 13
  - 4. 飞控接收控制指令类..... 14
    - 1) ID: 0x40: 遥控器数据..... 14
    - 2) ID: 0x41: 实时控制帧..... 14
  - 5. 光流信息类..... 15
    - 1) ID: 0x51: 匿名光流数据..... 15
  - 6. GPS 航点读写帧..... 16
    - 1) ID: 0x60: 航点读取..... 16
    - 2) ID: 0x61: 航点写入、航点读取返回..... 16
- 三、功能触发类帧..... 17
  - 1. ID: 0xC0: CMD 命令帧..... 17
  - 2. ID: 0xC1: CMD 功能帧..... 17
  - 3. ID: 0xC2: CMD 命令信息帧..... 17
- 四、参数读写类帧..... 19
  - 1. ID: 0xE0: 参数命令..... 19
  - 2. ID: 0xE1: 参数值写入、参数值读取返回..... 20

3.	ID: 0xE2: 参数信息返回 .....	20
4.	ID: 0xE3: 设备信息返回 .....	20
5.	通用校准参数.....	21
五、	固件升级.....	22
六、	其他帧.....	24
1)	ID: 0xA0: LOG 信息输出--字符串.....	24
2)	ID: 0xA1: LOG 信息输出--字符串+数字 .....	24
3)	ID: 0xB0: 图像数据.....	24
4)	ID: 0xB1: 基于 IP 组网的数据（格式 1） .....	24
5)	ID: 0xB2: 基于 IP 组网的数据（格式 2） .....	25
七、	数据定义.....	26
1.	硬件地址定义.....	26

# 一、匿名通信协议介绍

## 1. 通信帧格式介绍

为了适应多种数据类型的传输，保证高效的通信效率，所有数据的通信，均需要遵守本通信帧格式。本格式在确保通信高效、源码简单、可移植性高的基础上，实现数据正确性判断，有效避免数据传输过程中出现的错误数据导致的错误解析。

具体帧格式如下：

帧头 HEAD	源地址 S_ADDR	目标地址 D_ADDR	功能码 ID	数据长度 LEN		数据内容 DATA	和校验 SUM CHECK	附加校验 ADD CHECK
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	n Byte	1 Byte	1 Byte

字段定义如下：

字段	长度	内容
帧头：HEAD	1 字节	固定为十六进制的 0xAB
源地址：S_ADDR	1 字节	表示发送数据的设备 ID，定义附后
目标地址：D_ADDR	1 字节	表示接收数据的设备 ID，定义附后
功能码：ID	1 字节	表示本帧的功能识别码，定义附后
数据长度：LEN	2 字节	表示数据区（DATA 区）长度
数据内容：DATA	N 字节	本帧数据携带的数据
和校验：SUM CHECK	1 字节	和校验字节，计算方法附后
附加校验：ADD CHECK	1 字节	附加校验字节，计算方法附后

匿名协议采用小端模式，低字节在前，高字节在后（协议内所有 1 字节以上的数据类型，比如数据长度 LEN 以及 DATA 数据内容中的数据等，均是低字节在前，比如 DATA 长度等于 5，那么此时数据长度 LEN=5，其对应的小端十六进制为 0x0500）。

### ■ 和校验 SUM CHECK 计算方法：

从帧头 0xAB 字节开始，一直到 DATA 区结束，对每一字节进行累加操作，只取低 8 位

### ■ 附加校验 ADD CHECK 计算方法：

计算和校验时，每进行一字节的加法运算，同时进行一次 SUM CHECK 的累加操作，只取低 8 位。

### ■ 校验计算示例：

假设数据帧缓存为 data\_buf 数组，0xAB 存放于数组起始位置，那么 data\_buf[4]、data\_buf[5]存放的是数据长度，校验程序如下：

#### ◆ 计算校验字节：

```
uint8_t sumcheck = 0;
uint8_t addcheck = 0;
uint16_t flen = data_buf[4] + data_buf[5] * 256;
For(uint16_t i=0; i < (flen+6); i++)
{
    sumcheck += data_buf[i];           //从帧头开始，对每一字节进行求和，直到 DATA 区结束
    addcheck += sumcheck;             //每一字节的求和操作，进行一次 sumcheck 的累加
}
//将计算出来的校验数据写入数据帧
data_buf[flen+6] = sumcheck;
data_buf[flen+7] = addcheck;
```

#### ◆ 校验数据帧：

```
uint8_t sumcheck = 0;
uint8_t addcheck = 0;
uint16_t flen = data_buf[4] + data_buf[5] * 256;
For(uint16_t i=0; i < (flen+6); i++)
{
    sumcheck += data_buf[i];           //从帧头开始，对每一字节进行求和，直到 DATA 区结束
    addcheck += sumcheck;             //每一字节的求和操作，进行一次 sumcheck 的累加
}
//如果计算出的 sumcheck 和 addcheck 和接收到的 check 数据相等，代表校验通过，反之数据有误
if(sumcheck == data_buf[flen+6] && addcheck == data_buf[flen+7])
    return true;                     //校验通过
else
    return false;                    //校验失败
```

## 2. 匿名安全通信协议

大家在调试时，经常会使用串口数传一类的无线通信模块，这类模块会极大提高调试和通信的便捷程度，可以无线实时监视设备状态、调整参数等。但无线数传相比有线通信，其稳定性大大降低，数据发送出去，并不能保证对侧能百分百接收到，又或者接收到了，但是数据有可能被干扰而接收到错误的数​​据。这在传输显示数据时没有问题，因为显示数据缺少部分数据并不会影响设备正常运行。但是如果关键敏感的数据丢失或者接收错误，比如控制命令、参数信息等，就会影响设备的正常运行，故必须定义一种安全通信协议。

匿名规定，参数写入类、命令控制类等非显示类帧，均需返回验证，其过程如下：

如发送一个参数 ID 为 10 的参数值给设备，当上位机发送参数后，会等待帧 ID 为 0 的校验帧，校验帧格式如下。只有当上位机收到校验帧，并且校验帧的 ID\_GET、SC\_GET、AC\_GET 与发送帧相同时，代表本次通信完成，设备已经正确收到了该参数。

若上位机在规定时间内没有收到帧 ID 为 0 的校验帧，或者校验帧数据和发送帧的不同，则上位机认为本次通信出错，会重新尝试发送该参数。

### 1) ID：0x00：数据校验帧

帧头 HEAD	源地址 S_ADDR	目标地址 D_ADDR	功能码 ID	数据长度 LEN		数据内容 DATA	和校验 SC	附加校验 AC
0xAB	0xXX	0xXX	0x00	3	0	格式如下	程序计算	程序计算

DATA 区域内容：

数据类型	U8	U8	U8
数据内容	ID_GET	SC_GET	AC_GET

ID\_GET：需要校验的帧的帧 ID 码。

SC\_GET、AC\_GET：需要校验的帧的和校验 SC 和附加校验 AC。

### 3. 灵活格式帧

灵活格式帧，我们也可以叫做用户自定义帧，也就是用户可以自己定义数据内容格式的数据帧。可能从名字无法很好的理解灵活格式帧有什么用，那么我们举一个简单的例子。

假如我在调试一个自己写的滤波算法，传感器原始数据 A，为 int16 格式，使用滤波算法对 A 进行滤波后，得到滤波后数据 B，B 也是 int16 格式。滤波后数据经过控制算法，输出一个控制量 C，C 是 int32 格式。那我需要对滤波算法和控制算法进行调试，肯定是需要得到 ABC 三个数据的波形，根据波形进行数据分析。

那么如何将数据 ABC 发送至上位机进行显示呢，就要用到灵活格式帧了。灵活格式帧共 10 帧，帧 ID 从 0xF1 到 0xFA，每一帧可以携带 N 个数据，每一个数据可以分别设置为 U8、S16、U16、S32、Float 等格式。

那么我们可以用 0xF1 帧，添加 3 个数据，第一个数据为 int16，第二个数据为 int16，第三个数据为 int32。如下图：

帧ID:	0xF1	数据个数:	3	数据长度:	8	帧长度:	16	接收频率:	0
数据ID:	0	数据名称:F1#0:	A	数据类型:	Int16	数据值:	0		✖
数据ID:	1	数据名称:F1#1:	B	数据类型:	Int16	数据值:	0		✖
数据ID:	2	数据名称:F1#2:	C	数据类型:	Int32	数据值:	0		✖
添加数据									

数据名称后面的输入框，还可以输入自定义数据的名称，这里以 A、B、C 为例。

到此，上位机的配置完成，只需要单片机按照如下协议格式将数据发送至上位机，即可观察到对应的数据值开始刷新，并可绘制对应数据波形。

#### 1) ID：0xF1~0xFA：灵活格式帧

帧头 HEAD	源地址 S_ADDR	目标地址 D_ADDR	功能码 ID	数据长度 LEN	数据内容 DATA	和校验 SC	附加校验 AC
0xAB	0xFF	0xFF	0xF1~0xFA	数据长度	格式如下	程序计算	程序计算

DATA 区域内容：

举例说明 DATA 区域格式，例如上文，需要发送 ABC 三个数据，AB 为 int16 型，C 为 int32 型，那么 ABC 三个数据共 2+2+4=8 字节，那么 LEN 字节为 8，帧 ID 为 0xF1，DATA 区域依次放入 ABC 三个数据，然后计算 SC、AC，完成后将本帧发送至上位机即可。

注意：

数据区使用小端模式，低字节在前。

## 二、 基本信息类帧

根据帧 ID 区分不同的信息，基本的帧格式如下：

帧头 HEAD	源地址 S_ADDR	目标地址 D_ADDR	功能码 ID	数据长度 LEN	数据内容 DATA	和校验 SC	附加校验 AC
0xAB	0XX	0XX	0XX	数据长度	格式如下	程序计算	程序计算

比如发送 ID：0x01：惯性传感器数据，那么功能码 ID 就等于 0x01，发送 ID：0x02：罗盘、气压、温度传感器数据，那么功能码 ID 就等于 0x02。

### 1. 飞控相关信息类

#### 1) ID：0x01：惯性传感器数据

数据长度：13 字节

DATA 区域内容：

数据类型	int16	int16	int16	int16	int16	int16	uint8
数据内容	ACC_X	ACC_Y	ACC_Z	GYR_X	GYR_Y	GYR_Z	SHOCK_STA

ACC：加速度传感器数据，单位 cm/ss。

GYR：陀螺仪传感器数据，32768 对应 2000deg/s。

SHOCK\_STA：震动状态。

#### 2) ID：0x02：罗盘、温度传感器数据

DATA 区域内容：

数据类型	int16	int16	int16	int16	uint8
数据内容	MAG_X	MAG_Y	MAG_Z	TMP	MAG_STA

MAG：磁罗盘传感器数据。

TMP：传感器温度，放大 10 倍传输，0.1 摄氏度。

MAG\_STA：罗盘状态

#### 3) ID：0x03：飞控姿态：欧拉角格式

DATA 区域内容：

数据类型	int16	int16	int16	uint8
数据内容	ROL*100	PIT*100	YAW*100	FUSION_STA

ROL、PIT、YAW：姿态角，依次为横滚、俯仰、航向，精确到 0.01。

FUSION\_STA：融合状态。

#### 4) ID：0x04：飞控姿态：四元数格式

DATA 区域内容：

数据类型	int16	int16	int16	int16	uint8
数据内容	V0*10000	V1*10000	V2*10000	V3*10000	FUSION_STA

V0、V1、V2、V3：四元数，传输时扩大 10000 倍。

FUSION\_STA：融合状态。

#### 5) ID：0x05：高度数据

DATA 区域内容：

数据类型	Int32	Int32	Int32	UInt8
------	-------	-------	-------	-------



数据内容	ALT_BAR	ALT_ADD	ALT_FU	ALT_STA
------	---------	---------	--------	---------

ALT\_FU：融合后对地高度，单位厘米。

ALT\_BAR：气压计高度，单位 cm。

ALT\_ADD：附加高度传感高度数据，如超声波、激光测距，单位厘米。

ALT\_STA：测距状态。

### 6) ID：0x06：飞控运行模式

DATA 区域内容：

数据类型	U8	U8	U8	U8	U8
数据内容	MODE	SFLAG	CID	CMD0	CMD1

MODE：飞控模式。

SFLAG：功能标志， 0 锁定，1 解锁，2 已起飞。

CID、CMD0、CMD1：当前飞控执行的指令功能（指示最近的一次，完成后复位为“悬停功能”），对应后边指令表。

### 7) ID：0x07：飞行速度数据

DATA 区域内容：

数据类型	int16	int16	int16
数据内容	SPEED_X	SPEED_Y	SPEED_Z

SPEED\_XYZ：依次为 XYZ 方向上的速度，单位 cm/s。

### 8) ID：0x08：位置偏移数据

DATA 区域内容：

数据类型	S32	S32	S32
数据内容	POS_X	POS_Y	POS_Z

POS\_XYZ：融合后位置数据，相比起飞点的位置偏移量，单位 cm。

### 9) ID：0x09：风速估计

DATA 区域内容：

数据类型	int16	int16
数据内容	WIND_X	WIND_Y

WIND\_XY：风速估计，单位 cm/s。

### 10) ID：0x0A：目标姿态数据

DATA 区域内容：

数据类型	int16	int16	int16
数据内容	TAR_ROL	TAR_PIT	TAR_YAW

TAR\_ROL、TAR\_PIT、TAR\_YAW：依次为横滚、俯仰、航向的目标角度，精确到 0.01。

### 11) ID：0x0B：目标速度数据

DATA 区域内容：

数据类型	int16	int16	int16
数据内容	TAR_SPEED_X	TAR_SPEED_Y	TAR_SPEED_Z

TAR\_SPEED\_XYZ：依次为 3 轴目标速度，单位 cm/s。

### 12) ID：0x0C：回航信息

DATA 区域内容：

数据类型	Int16	UInt16
数据内容	R_A*10	R_D

R\_A: 回航角度, 正负 180 度, 传输时扩大 10 倍变成整数传输。  
R\_D: 回航距离, 单位为米。

### 13) ID: 0x0D: 电压电流数据

DATA 区域内容:

数据类型	U16	U16
数据内容	VOTAGE*100	CURRENT*100

VOTAGE、CURRENT: 依次为电压、电流, 传输时扩大 100 倍。

### 14) ID: 0x0E: 外接模块工作状态

DATA 区域内容:

数据类型	Uint8	Uint8	Uint8	Uint8
数据内容	STA_G_VEL	STA_G_POS	STA_GPS	STA_ALT_ADD

STA\_G\_VEL: 通用速度传感器状态

STA\_G\_POS: 通用位置传感器状态

STA\_GPS: GPS 传感器状态

STA\_ALT\_ADD: 附加测高传感器状态

传感器的工作状态, 0 为无数据, 1 为有数据但不可用, 2 为正常, 3 为良好 (3 为 GPS 专用)。

### 15) ID: 0x0F: RGB 亮度信息输出

DATA 区域内容:

数据类型	Uint8	Uint8	Uint8	Uint8
数据内容	BRI_R	BRI_G	BRI_B	BRI_A

BRI\_R、BRI\_G、BRI\_B、BRI\_A: 分别为 RGB 指示灯的红、绿、蓝三色的亮度, BRI\_A 为单独 LED 亮度, 有效范围 0-20, 表示从暗到亮共 21 级亮度, 0 最暗, 20 最亮。

## 2. 飞控控制量输出类

### 1) ID: 0x20: PWM 控制量

DATA 区域内容:

数据类型	U16	U16	U16	U16	U16	U16	U16	U16
数据内容	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7	PWM8

PWM: PWM 输出信号, 范围 0-10000, 默认 4 轴, 单位 0.01%油门。  
4 轴模式只输出前 4 通道, 6 轴模式输出 6 通道, 8 轴模式输出 8 通道。

### 2) ID: 0x21: 姿态控制量

DATA 区域内容:

数据类型	int16	int16	int16	int16
数据内容	CTRL_ROL	CTRL_PIT	CTRL_THR	CTRL_YAW

CTRL\_ROL、CTRL\_PIT、CTRL\_YAW: 依次为横滚、俯仰、油门、航向控制量, 范围统一为+/-5000。  
CTRL\_THR 为油门控制量, 范围为 0~10000;  
用户可以利用以上数据自行增加电机映射矩阵, 从而控制 6 轴/8 轴等更高轴数或者异结构飞行器。

3. 飞控接收信息类

1) ID：0x30： GPS 传感器信息 1

DATA 区域内容：

数据类型	U8	U8	S32	S32	S32	S16	S16	S16	U8	U8	U8
数据内容	FIX_STA	S_NUM	LNG*1e7	LAT*1e7	ALT_GPS	N_SPE	E_SPE	D_SPE	PDOP*0.01	SACC*0.01	VACC*0.01

FIX\_STA：定位状态，UBX 协议的 FIX\_STA。

S\_NUM：卫星数量

LNG、LAT：依次为经度、纬度，传输时扩大 10000000 倍变成整数传输，使用时除以 10000000 即可。

ALT\_GPS：GPS 模块解算出的高度。

N\_SPE、E\_SPE、D\_SPE：NED 速度（cm/s）。

PDOP：定位精度，，0-20000，20000 表示 GPS 信息不可靠，传输时缩小 100 倍（0-200）

SACC、VACC：依次为速度精度、高度精度，最大值 20000（mm），传输时除以 100。

2) ID：0x31： 原始光流信息

DATA 区域内容：

数据类型	U8	S16	S16	U8
数据内容	TYPE	DX	DY	QUA

TYPE：光流模块型号。

DX、DY：光流移动距离（以像素为单位，一般为光流模块原始数据）。

QUA：光流信息质量，0 为无效，越大代表质量越好。

3) ID：0x32：通用位置型传感器数据（非捷联载体测量型）

DATA 区域内容：

数据类型	S32	S32	S32
数据内容	POS_X	POS_Y	POS_Z

POS\_XYZ：依次 XYZ 坐标轴的位置信息，单位 mm，0x80000000 表示数据无效。

注意：位置传感器坐标系与飞行器载体匿名坐标系应在飞控解锁时对准，即飞控解锁时，机头朝向位置传感器 X 轴正方向，飞机左侧指向定位传感器 Y 轴正方向。

4) ID：0x33：通用速度型传感器数据（捷联载体测量型）

DATA 区域内容：

数据类型	S16	S16	S16
数据内容	SPEED_X	SPEED_Y	SPEED_Z

SPEED\_XYZ：依次 XYZ 坐标轴的速度信息，单位 cm/s，0x8000 表示数据无效。

注意：速度型传感器安装时，应保证速度传感器坐标系与飞行器载体匿名坐标系对准，即飞控机头朝向速度传感器 X 轴正方向，飞行器左侧指向速度传感器 Y 轴正方向。

5) ID：0x34：通用测距传感器数据（捷联载体测量型）

DATA 区域内容：

数据类型	U8	BYTE*n
数据内容	TYPE	DATA

TYPE：测距类型定义，见下表。

TYPE	类型	数据长度
1	向前测距	1+4=5
2	向后测距	1+4=5
3	向左测距	1+4=5
4	向右测距	1+4=5
5	向上测距	1+4=5
6、0	向下测距	1+4=5
100	圆周扫描雷达测距	1+(2+2)*n

■ DATA 区域内容(TYPE=0,1,2,3,4,5,6):

数据类型	U8	S32
数据内容	TYPE=0,1,2,3,4,5,6	DIST

TYPE: TYPE=0,1,2,3,4,5,6 时，表示单向测距，方向见上表 TYPE 定义，数据为 Int32 格式，单位 mm。

■ DATA 区域内容(TYPE=100):

数据类型	U8	U16+U16	N*4
数据内容	TYPE=100	ANGLE+DIST	ANGLE+DIST...

TYPE: TYPE=100 时，表示圆周扫描雷达测距。

ANGLE: 以机头为 0 度，顺时针增加，范围为 0-360 度，精确到 0.01，比如 ANGLE=12345，表示 123.45 度。

DIST: 距离值，单位为厘米 cm，比如 DIST=12345，表示 123.45 米。

注意：当一帧数据只有 1 个测距值时，本帧数据长度 5 字节，一帧数据可发送多个测距值，此时，一个测距值包含 2 字节的角度和 2 字节的距离，故此时数据长度为测距点数量 N\*4+1 字节。

## 6) ID: 0x35: 通用图像特征点信息帧

DATA 区域内容:

数据类型	U8	S16	S16	U16
数据内容	ID	X	Y	ANGLE

ID: 目标编号，0~250 有效范围，255 表示未搜索到任何特征点，当 ID=255 时，后续 XY 以及角度信息无意义。

X、Y: 目标点对于图像中点的偏移信息，以图像长宽中点为基准点，左侧为 x 负半轴，下侧为 y 负半轴，量程为±1000。如果分辨率不是正方形，以长边为基准，长边两端分别对应±100，按分辨率比例，计算短边两个边缘的比例，以 320\*240 分辨率为例，横向分辨率 > 纵向分辨率，以图像中心为 0 点，图像最左侧对应 x=-100，图像最右侧对应 x=100，240/320=0.75，则图像最上侧对应 y=75，图像最底侧对应 y=-75。

例:

分辨率: 320\*240，目标点像素坐标 (100, 80)

X=-100/(320/2)\*1000=-625

Y=80/(320/2)\*1000=500

则对应本帧的 X、Y 分别为 -625、500。

ANGLE: 角度信息，单位度，以视野正上方为基准，顺时针为正

## 4. 飞控接收控制指令类

### 1) ID：0x40：遥控器数据

DATA 区域内容：

数据类型	int16	int16	int16	int16	int16	int16	int16	int16	int16	int16
数据内容	ROL	PIT	THR	YAW	AUX1	AUX2	AUX3	AUX4	AUX5	AUX6

THR、YAW、ROL、PIT、AUX：依次为油门、航向、横滚、俯仰、辅助通道值，数据范围 1000-2000。  
数据为 0 代表没有通信或者失控（与遥控设置有关）。

### 2) ID：0x41：实时控制帧

DATA 区域内容：

数据类型	int16	int16	int16	int16	int16	int16	int16
数据内容	CTRL_ROL	CTRL_PIT	CTRL_THR	CTRL_YAWDPS	CTRL_SPD_X	CTRL_SPD_Y	CTRL_SPD_Z

CTRL\_ROL、CTRL\_PIT：姿态角实时控制量，角度 $\pm 90$ ，放大 100 倍传输，精确到 0.01。  
CTRL\_THR：油门控制量，0-1000(0.1%)。  
CTRL\_YAW\_DPS：自旋角速度控制量， $\pm 200$  度每秒，单位度每秒。  
CTRL\_SPD\_X、Y、Z：分别为 XYZ 轴的期望速度值，单位厘米每秒，最大值由飞控参数设定部分定义。

数据/模式	0x41 数据有效/无效真值表(1 有效，0 无效)				
	姿态自稳	自稳+定高	定点飞行		定点+程控
			摇杆在中位	摇杆非中位	
CTRL_ROL	1	1	0	0	0
CTRL_PIT	1	1	0	0	0
CTRL_THR	1	0	0	0	0
CTRL_YAWDPS	1	1	1	0	0
CTRL_SPD_X	0	0	1	0	0
CTRL_SPD_Y	0	0	1	0	0
CTRL_SPD_Z	0	1	1	0	0

注意：

- 程控模式下，飞行控制由指令值设定，飞控全自动控制。
- 定点模式下，摇杆前 4 通道任一通道不在中位，将优先响应摇杆通道对应的输入。
- 定点模式下，摇杆的俯仰横滚中位死区为 $\pm 40$ ，油门航向中位死区为 $\pm 80$ 。

## 5. 光流信息类

### 1) ID：0x51：匿名光流数据

DATA 区域内容(MODE 0):

数据类型	U8	U8	S8	S8	U8
数据内容	MODE=0	STATE	DX_0	DY_0	QUALITY

MODE：0 表示本帧为原始的光流数据

STATE：状态标记位， 0：无效， 1：有效。

DX\_0、DY\_0：X、Y 轴的光流信息，对应移动的速度（像素移动速率，单位：像素/20 毫秒）

QUALITY：光流数据质量，数值越大，表示光流数据质量越好（0-255），仅供参考。

DATA 区域内容(MODE 1):

数据类型	U8	U8	S16	S16	U8
数据内容	MODE=1	STATE	DX_1	DY_1	QUALITY

MODE：1 表示本帧为融合的光流数据

STATE：状态标记位， 0：无效， 1：有效。

DX\_1、DY\_1：X、Y 轴的光流信息，对应移动的速度（地面速度,单位：厘米/秒）

QUALITY：光流数据质量，数值越大，表示光流数据质量越好（0-255），仅供参考。

注意：本组数据仅在高度传感器数据有效/陀螺仪无异常时有效。

DATA 区域内容(MODE 2):

数据类型	U8	U8	S16	S16	S16	S16	S16	S16	U8
数据内容	MODE=2	STATE	DX_2	DY_2	DX_FIX	DY_FIX	INTEG_X	INTEG_Y	QUALITY

MODE：2 表示本帧为惯导融合的光流数据

STATE：状态标记位， 0：无效， 1：有效。

DX\_2、DY\_2：X、Y 轴的光流信息，对应融合的移动的速度（地面速度,单位：厘米/秒）；

DX\_FIX、DY\_FIX：修正后的 X、Y 轴的移动速度，适用于积分计算（地面速度,单位：厘米/秒）；

INTEG\_X、INTEG\_Y：X、Y 轴的速度积分值（单纯积分，仅供参考，单位厘米，-32768~+32767 循环）。

QUALITY：光流数据质量，数值越大，表示光流数据质量越好（0-255），仅供参考。

注意：本组数据仅在光流融合有效、高度传感器数据有效/陀螺仪无异常时有效。

## 6. GPS 航点读写帧

### 1) ID: 0x60: 航点读取

DATA 区域内容(MODE 0):

数据类型	U8
数据内容	NUM

NUM=0xFF: 读取航点数量。

NUM=n (0≤n≤0xFE): 读取第 n 个航点。

注意:

- ◆ 当读取航点数量时, 飞控以本帧格式也就是帧 ID: 0x60 格式返回航点数量, NUM=航点数量。
- ◆ 当读取航点数据时, 飞控返回帧 ID: 0x61 的航点内容。

### 2) ID: 0x61: 航点写入、航点读取返回

DATA 区域内容(MODE 0):

数据类型	U8	S32	S32	S32	U16	U16	U8	U8	U8	U8	U8
数据内容	NUM	LNG	LAT	ALT	SPD	YAW	FUN	CMD1	CMD2	CMD3	CMD4

NUM: 当前读写的航点编号, 从 0 开始, 0 号航点表示 HOME 起飞点。

LNG/LAT: 航点经度、纬度信息, 以 int32 格式通信, 传输时乘以 10000000, 保留小数点后 7 位。

ALT: 航点高度, 单位 cm。

SPD: 飞行速度, 单位 cm/s。

YAW: 机头朝向, 单位度, 有以下几种用法:

- ◆ 0-359: 和地磁北的夹角;
- ◆ 400: 机头朝向目标点;

FUN: 航点功能。

CMD1-4: 功能参数。

注意: 飞控收到本帧后, 需返回校验信息, 即返回帧 ID=0x00 的校验帧。

航点通信过程:

读取飞控:

- 1、上位机发送: 0xAB+ADDR+0x60+0x01+0xFF+SC+AC, 查询飞控内有多少个航点信息;
- 2、飞控返回: 0xAB+ADDR+0x60+0x01+NUM+SC+AC, NUM 为飞控内航点数量;
- 3、上位机依次发送: 0xAB+ADDR+0x60+0x01+N+SC+AC, N=0, 读取第一个航点;
- 4、飞控收到读取命令, 返回第 N 个航点的内容, 利用帧 ID 为 61 的航点内容帧;
- 5、上位机收到帧 ID 为 61 的航点内容后, 重复执行第 3 步, 开始读取下一个航点信息, 直到读取完毕。

写入飞控:

- 1、上位机发送: 0xAB+ADDR+0x61+22+DATA+SC+AC, DATA 为航点数据, 其中 NUM=0, 即从第一个航点开始发送;
- 2、飞控收到帧 ID=61 的航点数据帧后, 需返回帧 ID 为 0 的校验帧 (帧格式见本手册开头部分);
- 3、上位机收到帧 ID 为 0 的校验帧后, 进行校验, 如果单位时间内未收到校验帧或者校验出错, 上位机会尝试重新发送本航点信息; 若校验通过, 则重复执行第一步, 开始发送 NUM=1 的第二个航点信息, 直到发送完毕。
- 4、



### 三、 功能触发类帧

为了提升上位机的通用性,上位机不再定义各类功能命令的格式,而是定义好功能帧的格式,具体命令的内容,由下位机来定义,这样就实现了一个上位机可以对应各类不同的硬件及软件的调试工作。

功能命令的使用流程:

1. 上位机发送读取命令,读取下位机的命令数量;
2. 上位机根据命令数量,依次读取下位机各条命令的内容以及相关信息;
3. 上位机读取所有命令后,在命令界面显示出所有命令内容,用户点击最后的触发按钮,即可触发对应的命令,同时命令还可以携带参数,参数的定义也在下位机实现。

#### 1. ID: 0xC0: CMD 命令帧

DATA 区域内容:

数据类型	U8	U8	U8								
数据内容	CID0	CID1	CID2	VAL0	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7

CID0-2: 本 CMD 功能种类,与命令信息帧的 CID 对应

VAL0-7: CMD 功能帧参数,最少 0 个数据,最多 8 个数据,可选项,与命令信息定义的数据个数和格式对应,比如命令信息定义本帧命令有 3 个 int16 型参数,那么 VAL0、1、2 为 int16 格式,后面 VAL3-7 删除不发送;若本帧命令为简单的触发命令,比如触发参数恢复默认、校准传感器等,根据命令信息定义,可只使用 3 字节 CID,后面不跟任何 VAL 参数。故本数据帧长度是根据 VAL 的个数及格式可变的。

注意: 飞控收到本帧后,需返回校验信息,即返回帧 ID=0x00 的校验帧。

#### 2. ID: 0xC1: CMD 功能帧

DATA 区域内容:

数据类型	Unt8	Unt16
数据内容	CMD	VAL

目标地址 D\_ADDR: 读取哪个设备的命令列表,目标地址就应配置为该硬件的地址信息,比如读取飞控命令列表时本字节就应为 0x05。

CMD=0: 读取命令列表数量,此时 VAL 无意义,等于 0 即可,下位机收到后返回 0xC1 帧。返回时 CMD=0, VAL 代表参数个数。

CMD=1: 读取命令列表内容,此时 VAL 为功能序号 ID,下位机收到读取命令后返回 0xC2 帧,

#### 3. ID: 0xC2: CMD 命令信息帧

DATA 区域内容:

数据类型	Uint16	Uint8*11	Char*20	Char*N
数据内容	CMD_ID	CMD	CMD_NAME	CMD_INFO

CMD\_ID: 第几条命令

CMD: 依次为 CID0-2、VAL0-7,共 11 字节,其中 VAL0-7 字节,定义本帧命令对应的参数个数和参数数据类型,定义见下表。比如本帧数据不需任何参数,那么 VAL0-7 这 8 字节均为 0;如果本帧数据需要 1 个 uint8 型参数,那么 VAL0=1, VAL1-7 为 0。

CMD\_NAME: 命令名称

CMD\_INFO: 命令信息介绍

参数数据类型定义如下：

命令信息帧 VAL 值	对应的数据类型	占用字节
0	无需参数	0
1	Uint8	1
2	Int8	1
3	Uint16	2
4	Int16	2
5	Uint32	4
6	Int32	4
7	Float	4

举例 1：

触发校准命令，无参数，那么可以定义如下命令信息：

{0x01,0xA0,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, "Cali\_Trig1", "Cali\_Acc", AnoPTv8CmdFun\_Cali\_Trig1}

其中，{0x01,0xA0,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}为 11 字节的命令 CMD 和 VAL 结构体，CMD0-2 分别为 0x01、0xA0、0x01，这 3 字节可随便定义，但不可与其他命令相同。后面 8 字节的 VAL 均为 0，表示本帧命令不需要参数。

后面的"Cali\_Trig1", "Cali\_Acc", AnoPTv8CmdFun\_Cali\_Trig1，分别为命令的名称，命令的简介，命令对应的函数指针。

举例 2：

一个用户自定义的带有 3 个参数的命令，第一个参数 uint8，第二个参数 int16，第三个参数 float，那么可以定义如下命令信息：

{0x01,0xB0,0x01,0x01,0x04,0x07,0x00,0x00,0x00,0x00,0x00,0x00}, "USERCMD1", "UserDefCmd1", AnoPTv8CmdFun\_UserCmd1}

其中，{0x01,0xB0,0x01,0x01,0x04,0x07,0x00,0x00,0x00,0x00,0x00,0x00}为 11 字节的命令 CMD 和 VAL 结构体，CMD0-2 分别为 0x01、0xB0、0x01，这 3 字节可随便定义，但不可与其他命令相同。后面 8 字节的 VAL 信息，第一字节为 0x01，表示第一个参数为 uint8 格式，第二字节为 0x04，表示第二个参数为 int16 格式，第三字节为 0x07，表示第三个参数为 float 格式。

## 四、 参数读写类帧

为了提升上位机的通用性，上位机不再定义参数的名称以及范围等各类参数信息，而是定义好参数的格式，各类设备按照定义好的格式来传输参数，而具体参数的范围、名称等信息，在上位机进行定义，这样就实现了一个上位机可以对应各类不同的硬件及软件的调试工作。

本协议支持多种格式的参数，包括 Uint8、Int8、Uint16、Int16、Uint32、Int32、Uint64、Int64、Float、Double、String（字符串）。

上位机触发读取命令时，会首先判断是否读取了当前设备的设备信息，因为如果设备信息为空，上位机不知道是要读取哪个设备的参数。若设备信息为空，上位机首先会尝试读取设备信息，读取到设备信息后，上位机开始读取该设备的参数。若未成功读取到设备信息，则上位机以广播地址（0xFF）为目标进行读取。

■ 参数读取的具体实现流程：

1. 上位机发送读取命令，读取下位机的参数数量；
2. 上位机根据参数数量，依次读取所有参数的信息，包含参数名称、参数格式、简介（参数序号从 0 开始，中间不能间断）；
3. 上位机根据参数数量，依次读取所有参数的数值信息；

■ 参数写入的具体实现流程：（智能写入：只发送修改过的参数）

1. 上位机判断哪些参数进行过修改，从第一个修改过的参数开始，依次发送；
2. 上位机发送某个修改过的参数值至下位机，并等待校验帧；
3. 下位机收到这个参数后，返回校验帧；
4. 上位机收到校验帧并校验通过后，开始发送下一个修改过的参数，直到发送完成；
5. 若上位机接收校验帧超时或校验出错，则重新发送本参数。

■ 参数写入的具体实现流程：（全部写入：发送所有参数）

6. 上位机从第一个参数开始，直到最后一个参数，依次发送；
7. 上位机发送某个参数值至下位机，并等待校验帧；
8. 下位机收到这个参数后，返回校验帧；
9. 上位机收到校验帧并校验通过后，开始发送下一个参数，直到发送完成；
10. 若上位机接收校验帧超时或校验出错，则重新发送本参数。

### 1. ID：0xE0：参数命令

DATA 区域内容：

数据类型	Unt8	Unt16
数据内容	CMD	VAL

目标地址 D\_ADDR：读取哪个设备的参数，目标地址就应配置为该硬件的地址信息，比如读取飞控参数时本字节就应为 0x05。

参数命令：

命令 CMD	内容 VAL	定义	返回数据
0x00	NA	读取设备信息，此时 VAL 无意义，等于 0 即可	下位机收到后返回 0xE3 帧
0x01	NA	读取参数个数，此时 VAL 无意义，等于 0 即可	下位机收到后返回 0xE0 帧，返回时 CMD=1，VAL 代表参数个数
0x02	ID	读取参数值，此时 VAL 为参数 ID	下位机收到后返回 0xE1 帧
0x03	ID	读取参数信息，此时 VAL 为参数 ID	下位机收到后返回 0xE2 帧

0x10	0xAA	所有参数恢复默认值	下位机返回 0x00 校验帧
	0xAB	保存参数命令	下位机返回 0x00 校验帧
	0x20	清除加速度校准数据，校准开始时会自动发送本命令，确保上传的数据为传感器原始数据	下位机返回 0x00 校验帧
	0x21	发送完加速度校准数据后发送本命令，用于触发参数保存及启用相关功能	下位机返回 0x00 校验帧
	0x30	清除罗盘校准数据，校准开始时会自动发送本命令，确保上传的数据为传感器原始数据	下位机返回 0x00 校验帧
	0x31	发送完罗盘校准数据后发送本命令，用于触发参数保存及启用相关功能	下位机返回 0x00 校验帧

2. ID：0xE1：参数值写入、参数值读取返回

DATA 区域内容：

数据类型	Unt16	Byte*n
数据内容	PAR_ID	PAR_VAL

目标地址 D\_ADDR：向哪个设备写入参数，目标地址就应配置为该硬件的地址信息，比如写入飞控参数时本字节就应为 0x05。如果本帧为下位机对上位机读取操作的返回，本字节为上位机地址 0xFE。

PAR\_ID：参数 ID。

PAR\_VAL：参数值，可变长度，长度由参数类型决定。

注意：飞控收到参数写入帧后，需返回校验信息，即返回帧 ID=0x00 的校验帧。

3. ID：0xE2：参数信息返回

DATA 区域内容：

数据类型	Unt16	Uint8	Char*20	Char*N
数据内容	PAR_ID	PAR_TYPE	PAR_NAME	PAR_INFO

PAR\_ID：本参数信息帧对应的参数 ID。

PAR\_TYPE：参数格式，格式定义如下：

PAR\_NAME：参数名称，字符格式，固定长度为 20 字符，不足部分补 0x00。

PAR\_INFO：参数介绍，字符格式，长度可变。

注意：上位机收到本帧数据后，并不需返回校验。

PAR_TYPE	参数格式	参数长度
0	Uint8	1
1	Int8	1
2	Uint16	2
3	Int16	2
4	Uint32	4
5	Int32	4
6	Uint64	8
7	Int64	8
8	Float	4
9	Double	8
10	String	n

4. ID：0xE3：设备信息返回

DATA 区域内容：

数据类型	Unt8	Int16	Int16	Int16	Int16	Char*N
数据内容	DEV_ID	HW_VER	SW_VER	BL_VER	PT_VER	DEV_NAME

DEV\_ID：设备 ID，与设备地址相同。

HW\_VER、SW\_VER、BL\_VER、PT\_VER：分别为硬件版本、软件版本、Bootloader 版本、通信协议版本。

没有 Bootloader 时，BL\_VER=0 即可。

DEV\_NAME：设备名称，字符串，最长 20 字节，N 表示字符串长度。

**注意：**上位机收到本帧数据后，并不需返回校验。

## 5. 通用校准参数

匿名助手设计有加速度计 6 面校准和磁罗盘校准功能，上位机会根据传感器采样数据计算出偏移、模长、误差等校准参数，为实现以上参数的自动写入，本协议规定了以上相关参数的名称。

当需要读写校准参数时，上位机会在设备的参数列表中搜索校准参数的名称，从而确定参数的 ID 范围（上位机会搜索第一个校准参数和最后一个校准参数的名称，并计算之间的参数数量，必须名称和数量均与本协议定义相同才会进入读写逻辑）。

加速度六面校准相关参数：

Name	Info
CAL_AOFF_X	加速度 Offset：X 轴偏移
CAL_AOFF_Y	加速度 Offset：Y 轴偏移
CAL_AOFF_Z	加速度 Offset：Z 轴偏移
CAL_ASEN_X	加速度 Sensitivity：X 轴感度
CAL_ASEN_Y	加速度 Sensitivity：Y 轴感度
CAL_ASEN_Z	加速度 Sensitivity：Z 轴感度
CAL_AEM1_X	加速度误差矩阵 1：X 轴
CAL_AEM1_Y	加速度误差矩阵 1：Y 轴
CAL_AEM1_Z	加速度误差矩阵 1：Z 轴
CAL_AEM2_X	加速度误差矩阵 2：X 轴
CAL_AEM2_Y	加速度误差矩阵 2：Y 轴
CAL_AEM2_Z	加速度误差矩阵 2：Z 轴
CAL_AEM3_X	加速度误差矩阵 3：X 轴
CAL_AEM3_Y	加速度误差矩阵 3：Y 轴
CAL_AEM3_Z	加速度误差矩阵 3：Z 轴

磁罗盘校准相关参数：

Name	Info
CAL_MOFF_X	磁罗盘 Offset：X 轴偏移
CAL_MOFF_Y	磁罗盘 Offset：Y 轴偏移
CAL_MOFF_Z	磁罗盘 Offset：Z 轴偏移
CAL_MSEN_X	磁罗盘 Sensitivity：X 轴感度
CAL_MSEN_Y	磁罗盘 Sensitivity：Y 轴感度
CAL_MSEN_Z	磁罗盘 Sensitivity：Z 轴感度

注意：以上参数，第一个参数的 ID 不做要求，根据用户设备随意存放，但加速度 15 个校准参数或磁罗盘的 6 个校准参数，需要依次、连续的存储，中间不能间断或删除。

# 五、 固件升级

本协议提供通用固件升级功能，所有固件升级相关通信帧均使用帧 ID：0xF0。

帧 ID：0xF0

DATA 区域内容：

数据类型	Unt8	Unt8*n
数据内容	IAP_CMD	IAP_DATA

- 固件升级流程：
- 1、 读取设备信息（非必须），一般情况下，建议升级固件前，首先读取设备的信息，使用参数读写类帧里面的读取设备信息功能，首先发送帧 ID 为 0xE0，命令为读取设备信息的参数命令帧，设备返回帧 ID 为 0xE3 的设备信息帧，包含设备种类、硬件版本、软件版本等信息。
  - 2、 上位机发送固件信息，包含固件总字节长度，每帧固件字节长度，固件校验信息、固件版本、适用硬件版本等信息。
  - 3、 下位机收到固件信息后，判断固件是否满足要求，满足则回复固件合法命令；反之，回复固件非法命令。
  - 4、 上位机收到固件合法命令后，上位机发送擦除命令，擦除设备指定区域内的 flash。
  - 5、 下位机擦除完毕后，返回擦除成功命令；若擦除过程出错，返回擦除失败命令。
  - 6、 上位机收到擦除成功命令后，开始逐帧发送固件内容。
  - 7、 下位机收到每一帧固件后，根据是否校验通过或写入成功，返回成功或失败信息至上位机。
  - 8、 上位机收到当前发送固件帧的校验信息并校验通过后，开始发送下一帧；校验失败则重新发送本帧。
  - 9、 固件发送完毕后，上位机发送固件传输完成命令。
  - 10、 下位机收到传输完成命令后，开始对固件进行校验，并返回校验结果。
  - 11、 上位机根据下位机返回的校验结果，提示是否升级成功。

帧功能	IAP_CMD	IAP_DATA
固件命令	0x00	0x0101： 下位机判断固件信息非法 0x0102： 下位机判断固件信息合法 0x0110： 擦除 flash 命令 0x0111： 擦除 falsh 失败 0x0112： 擦除 falsh 成功 0x1001+BinFNum (uint16)： 第 BinFNum 帧写入失败 0x1002+BinFNum (uint16)： 第 BinFNum 帧写入成功 0x1010： 固件发送完成命令 0x1011： 下位机固件校验失败 0x1012： 下位机固件校验通过
固件信息	0x01	BinLen： uint32： 固件总长度 BinFLen： uint16： 每帧数据内固件长度 BinCrc： uint16： 固件 CRC 校验数据 BinVer： uint16： 固件软件版本（非必须，默认=0） HwTyp： uint8： 固件适用硬件种类（非必须，默认=0） HwVerMin： uint16： 固件适用最小硬件版本（非必须，默认=0） HwVerMax： uint16： 固件适用最大硬件版本（非必须，默认=0） Cmd： uint32： 附加命令（非必须，默认=0）
固件内容	0x10	BinFNum： uint16： 当前帧的序号，从 0 开始，依次递增 BinData： 长度为 BinFLen 字节，固件内容数据


本功能支持两种格式的固件，bin 和 bia 格式。

bin：通用固件，一般编译工具可直接生成 bin 格式的固件文件，上位机可直接加载 bin 文件。使用 bin 文件时，固件信息中的 BinVer、HwTyp、HwVerMin、HwVerMax、Cmd 均为默认值 0，不可修改。

bia：匿名定义的一种基于 bin 格式的固件，在 bin 固件的文件头，添加 100 字节的信息，100 字节的信息内容定义如下：

- Byte0：0x62，字母 b
- Byte1：0x69，字母 i
- Byte2：0x61，字母 a
- Byte3-Byte9：缺省值 0
- Byte10-Byte11：固件版本号，uint16 格式，小端模式
- Byte12-Byte13：固件 crc 校验
- Byte14-Byte19：缺省值 0
- Byte20：HwTyp，固件适用硬件种类
- Byte21-Byte29：缺省值 0
- Byte30-Byte31：HwVerMin，uint16：固件适用最小硬件版本
- Byte32-Byte33：HwVerMax，uint16：固件适用最大硬件版本
- Byte34-Byte49：缺省值 0
- Byte50-Byte53：Cmd，uint32：附加命令
- Byte54-Byte99：缺省值 0

## 六、其他帧

### 1) ID: 0xA0: LOG 信息输出--字符串

DATA 区域内容:

数据类型	Uint8	Uint8*(n-1)
数据内容	COLOR	STR

COLOR: 颜色, 0: 默认, 1: 红色, 2: 绿色。

STR: 需要显示的英文字符串, 比如需要显示字符串“ABCDE”, 则 STR 长度为 5 字节, 依次为 ABCDE 的 ASC 码。

### 2) ID: 0xA1: LOG 信息输出--字符串+数字

DATA 区域内容:

数据类型	Float	Uint8*(n-4)
数据内容	VAL	STR

VAL: 数值, int32 格式, 4 字节。

STR: 需要显示的英文字符串, 比如需要显示字符串“ABCDE”, 则 STR 长度为 5 字节, 依次为 ABCDE 的 ASC 码。

### 3) ID: 0xB0: 图像数据

DATA 区域内容:

数据类型	Uint16	UInt8	Uint16	Uint16	Uint8*10	Uint8*n
数据内容	FLAG	COLOR	WIDTH	HEIGHT	VAL*10	PIXDATA

FLAG: 标志位。

COLOR: 本帧图像的颜色, 具体颜色对应关系见下表。

WIDTH: 图像宽度, 比如 320\*240 的分辨率, 则 WIDTH=320。

HEIGHT: 图像高度, 比如 320\*240 的分辨率, 则 HEIGHT =240。

VAL: 调试参数, 用户自定义, 比如可以发送对应本帧图像的识别结果等, 共 10 字节。

PIXDATA: 图像数据, 一位代表一个像素, PIX\_DATA 长度应为本帧数据传输的所有图像行数\*WIDTH/8。

颜色定义:

COLOR	颜色
0x00	灰度, 0-255, 1 字节一个像素
0x01	默认颜色, 黑白二值化单色, 推荐
0x02	红色
0x03	绿色
0x04	蓝色
0x05	黄色
0x06	紫色

### 4) ID: 0xB1: 基于 IP 组网的数据 (格式 1)

DATA 区域内容:

数据类型	UInt8*4	Uint8*4	Uint16	Uint8*(n-10)
数据内容	数据源 IP	数据目标 IP	目标 Port	DATA

数据源 IP: 发送数据设备的 IP 地址, 比如 192.168.0.1, 用户发送数据时, 数据源 IP 的 4 字节均写 0, 数传发送时自动填充本机 IP 地址到数据源 IP。

数据目标 IP: 数据目标设备的 IP 地址, 比如 192.168.0.2, 发送数据时需用户指定目标设备 IP。

目标 Port: 数据目标 Socket 端口, 0 为数传默认数据端口

DATA: 数据内容, 若干帧普通数据, 数据内容为匿名 V8 协议帧, 从帧头 0xAB 开始直至附加校验 AC, 一次



可传输一帧或多帧数据。

5) ID：0xB2：基于 IP 组网的数据（格式 2）

DATA 区域内容：

数据类型	UInt8	UInt8	UInt16	UInt8*(n-4)
数据内容	数据源 IP 最低字节	数据目标 IP 最低字节	目标 Port	DATA

数据源 IP 最低字节：发送数据设备的 IP 地址的最低字节，比如 192.168.0.80，那么本字节为 80（十进制），用户发送数据时，本字节写 0，数传发送时自动填充本机 IP 最低字节。

数据目标 IP 最低字节：数据目标设备的 IP 地址的最低字节，，发送数据时需用户指定。

DATA：数据内容，若干帧普通数据，数据内容为匿名 V8 协议帧，从帧头 0xAB 开始直至附加校验 AC，一次可传输一帧或多帧数据。

注意：

本帧数据的 IP 地址，自动使用本机 IP 的网段，比如 WIFI 的 DHCP 设置的 192.168.0.X 网段，当数传模块连接到路由器后，假设分配的 IP 为 192.168.0.123，那么该数传使用本帧收发数据时，自动使用 192.168.0 前三字节，只需确定 IP 地址的最低字节即可。

6) ID：0xFB：特殊数据

DATA 区域内容：

数据类型	n
数据内容	DATA

本协议未定义的特殊数据，比如匿名数传的专用配置信息帧等。

该 ID 的数据帧用户无需关注。

# 七、 数据定义

## 1. 硬件地址（硬件 ID）定义

地址 ID 码	十六进制	定义	是否支持修改 ID
1-199	0x01-0xC7	用户自定义 ID	NA
1-20	0x01-0x20	推荐前 20 个 ID 配置给设备的不同端口	
200-254	0xC8-0xFE	匿名预定义设备 ID	
210	0xD2	匿名协议至 ROS2 系统通信节点（AnoRosBridge）	不支持 q
219	0xDB	匿名 MCU	支持
220	0xDC	匿名飞控(拓空者)	支持
221	0xDD	匿名 IMU	支持
222	0xDE	匿名光流	支持
223	0xDF	基于匿名协议的 OPENMV	支持
230	0xE6	匿名数传 V410 以上	不支持
253	0xFD	协议模拟器	支持
254	0xFE	上位机	不支持
255	0xFF	无特定目标，用于数据广播型输出	不支持

匿名设备会默认使用预先定义的 ID，用户自己的设备，可自行分配 1 到 199 共 199 个 ID，但应注意同一通信系统内，所有设备的 ID 应互不相同。

推荐 1 到 20 共 20 个 ID，配置给设备的不同端口，比如某设备为一个有 5 个串口的单片机，则可以为串口 1 到 5 分别配置为 ID1 到 5，单片机 ID 配置为 21，这样上位机如果数据发给单片机，则数据帧目标 ID 为 21，如果上位机需要向接在单片机串口 1 的设备发送信息，则数据帧目标 ID 为 1，单片机收到目标 ID 为 1 的数据后，通过串口 1 发送出去即可。

特殊情况：

比如一个飞控上面，通过用户二次开发，接入了两个匿名光流传感器，如果光流传感器都用默认 ID，则上位机或其他设备将无法区分两个光流模块。此时，可以修改两个匿名光流的硬件 ID 为不同的值（在 0-199 范围内），这样就可以实现识别两个光流的目的。