



A9 飞控用户手册

V2.2



广州博睿创新科技有限公司

www.acfly.cn

1 阅读提示

1.1 使用建议

建议用户在 ACFLY 飞行控制的 QQ 交流群（购买后由官方发送邀请）下载或者 B 站（*）观看教学视频和阅读《ACFLY A9 飞控代码》了解产品，使用《ACFLY A9 飞控用户手册》可快速了解使用过程。

*B 站：<https://space.bilibili.com/250212993>

产品对应资料有两种下载方式：

- (1) ACFLY 售后群-群文件
- (2) 百度网盘：https://pan.baidu.com/s/1jh9w8LYkVDMt_XuVKbfJyg 提取码：acac

1.2 安全提示

1. 严禁电机上电带桨进行程序烧录，以免发生意外；
2. 严禁电机上电带桨插拔电调信号线，以免发生意外；
3. 严禁飞行器上电后用表笔直接对板子进行测量，容易导致单片机烧坏；
4. 产品一切源码和资料仅限于学习交流，禁止用于商业用途！违者必究！

1.3 正版注册

无人机飞行控制系统主要由电路板、固件和其他软件组成，ACFLY 主要产品已经具备自主知识产权，为了防止被直接抄板和拷贝代码，对飞控板均采用正版注册的防抄袭方式，**出厂默认已经注册**。

将飞控连接到电脑，飞控正常情况下会在**几声滴滴滴后，绿色灯慢闪**，此种状态证明该飞控证明已经注册，可正常使用，并可在我的电脑—>右键—>管理—>设备管理器—>端口中查看飞控的 COM 口号。若飞控灯不亮或者亮其他颜色，请联系客服。

1.4 关注微信公众号

请使用移动设备的微信软件扫描以下二维码，关注【博睿开源飞控】微信公众号，实时获取产品更新信息以及商务合作信息。



博睿开源飞控

2 开发环境搭建

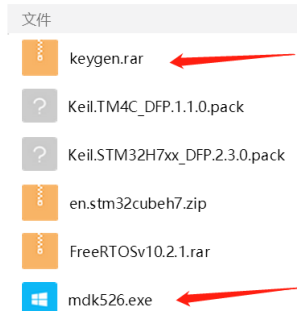
2.1 编译环境（MDK）安装

（1）MDK 开发软件安装

到售后群下载或者在 keil 官网 <http://www.keil.com/> 的 Download 页面自行下载双击安装 MDK5.26:

ACFly EDU售后群

群文件 > Keil MDK 开发环境资料



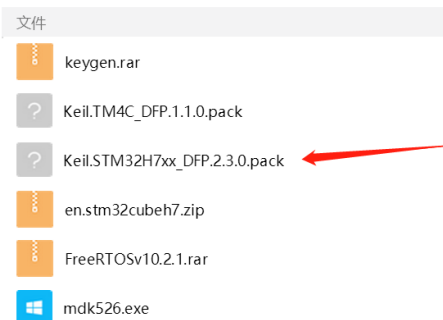
选自己喜欢的目录安装完成后，使用 keygen 注册机进行破解（File->License Manger 中输入序列号，具体怎么破解可以查百度）。

（2）芯片支持包安装

在售后群中下载支持包进行安装：

ACFly EDU售后群

群文件 > Keil MDK 开发环境资料

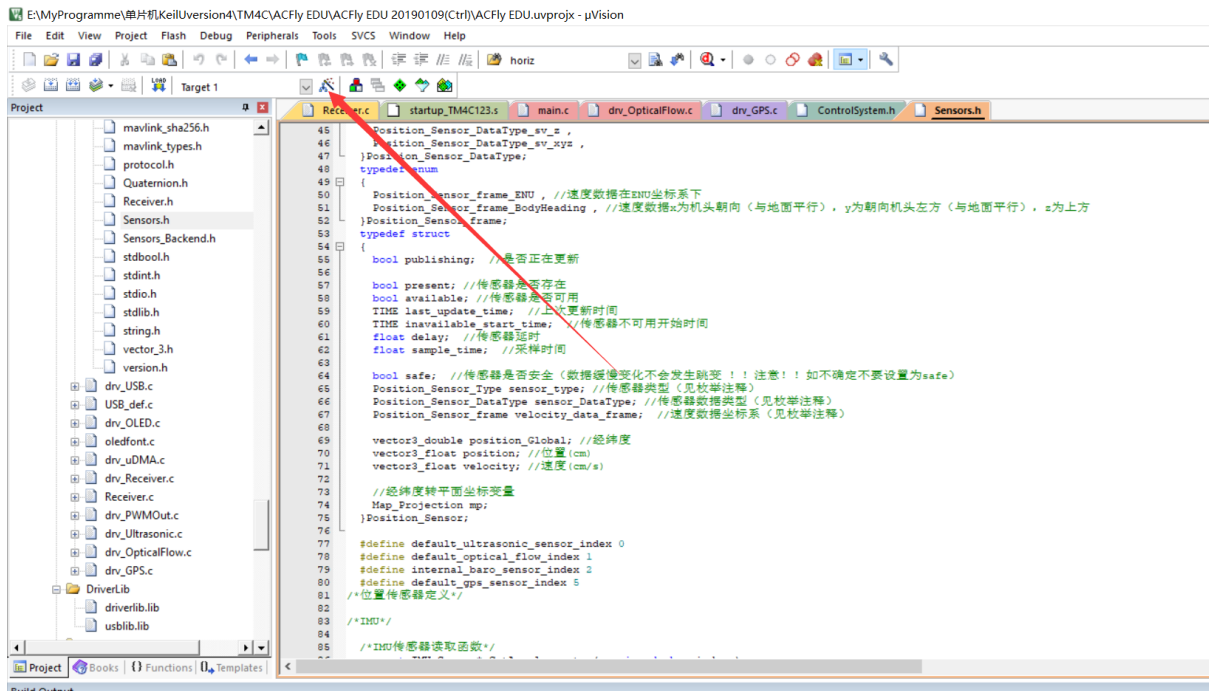


或在 keil 官网 <http://www.keil.com/pack> 下载下图中的最新版本软件支持包并安装

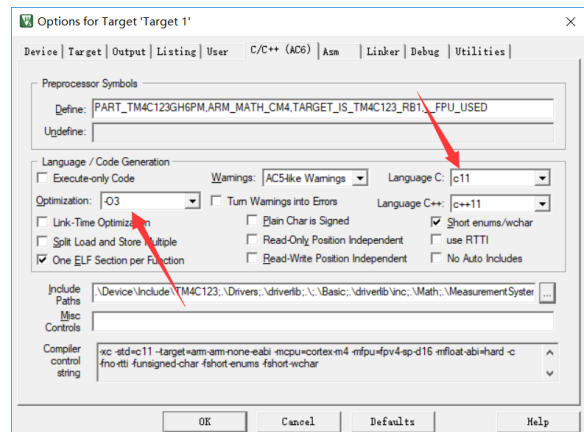
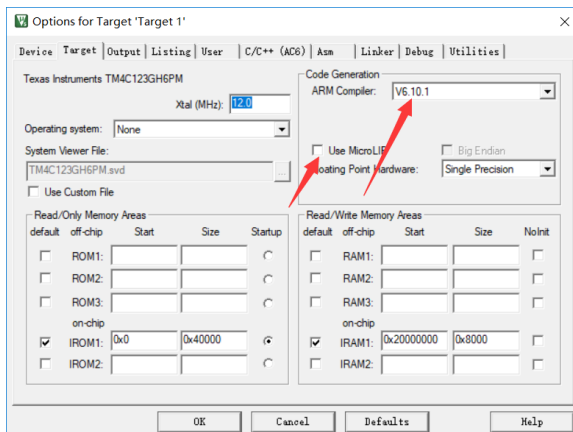


2.2 编译环境设置






在 MDK 界面下点击下列按钮进入工程配置选项：



在 Target 选项卡中选择 V6 开头的编译器（如果没有，可以安装 MDK 较新的版本或者前往 <https://developer.arm.com/products/software-development-tools/compilers/arm-compiler/downloads/version-6> 网页单独下载编译器并安装进 MDK），并取消 Use MicroLib 复选框；在 C/C++(AC6)选项卡中进行下图中的配置：



3 飞控接口定义

序号	图示	接口定义									
A		UART7: TX RX GND 5V					Ultrasonic: Echo Triger GND 5V				
		IIC2: 5V GND SCL SDA					UART5: 5V GND RX TX				
B		UART1: TX RX GND 5V					CAN: CANL CANH GND 5V				
		UART3: 5V GND RX TX					SWJ: 3.3V GND SWCLK SWDIO				
C		PPM	SBUS	PWM1	PWM2	PWM3	PWM4	PWM5	PWM6	PWM7	PWM8
		S	S	S	S	S	S	S	S	S	S
		5V	5V	5V	5V	5V	5V	5V	5V	5V	5V
		GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
D		UART8: TX RX GND 5V					PowerADC: BatADC CurrentADC GND 5V				
		IIC1: 5V GND SCL SDA					Power: 5V GND GND 5V				
E		GND	3.3V	SCL	SDA	REST	DC	BLK			

源固件默认接口，可根据串口配置功能参数更改

- 超声波：Ultrasonic 接口
- 光流串口：Uart5 接口
- GPS 串口：Uart8 接口
- 外置罗盘：IIC1 接口
- OpenMV 串口：Uart3 接口
- 数传串口：Uart1 接口
- 激光串口：Uart7 接口


4 飞控尺寸重量

- 长 55mm，宽 44mm，高 19mm
- 重 39 克



5 飞控供电

A9 飞控支持双冗余供电, 供电电压 5V, 分别是 powerADC 端口和 Power 端口。以下的电流计为例, 电流计共有 6 个引脚, 可以将其一路 5V 和一路 GND 接到飞控的 Power 端口, 而剩余的 5V、GND、CURRENT、VOLTAGE 引脚则对应接到飞控的 powerADC 接口。**可直接购买使用 ACFLY 电流计, 免焊线。**

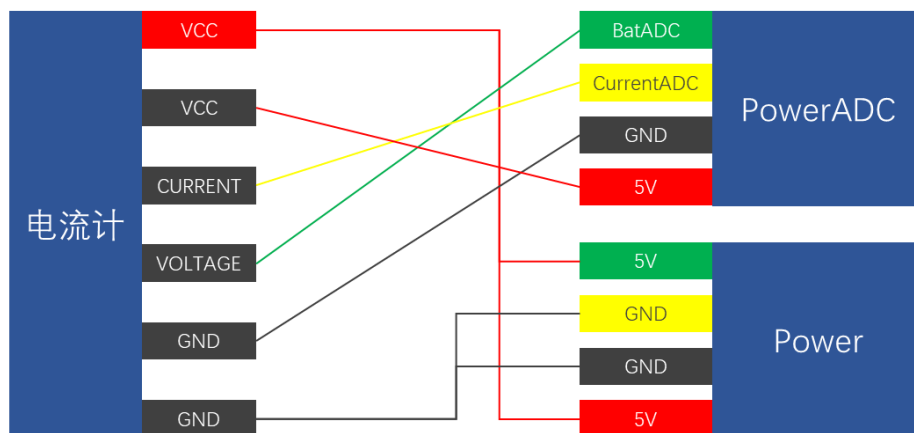
	UART8: TX RX GND 5V	PowerADC: BatADC CurrentADC GND 5V
	IIC1: 5V GND SCL SDA	Power: 5V GND GND 5V

飞控供电端口说明



Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	VCC	+5V
3 (blk)	CURRENT	+3.3V
4 (blk)	VOLTAGE	+3.3V
5 (blk)	GND	GND
6 (blk)	GND	GND

电流计引脚说明



电流计与飞控的接口 GH1.25 4pin 接线方式

6 屏幕显示说明



参数	说明						
M00:	未定义，一直不变，可以不管						
HthA:	姿态解算健康度，0%-100%，正常都是 100%						
Rol:	Roll 的简写，横滚角度，单位度						
Pit:	Pitch 的简写，俯仰角度，单位度						
Yaw:	偏航角度，单位度						
Vz:	垂直方向速度，单位 cm/s						
HthP:	水平位置解算健康度						
Vx:	X 轴方向速度，单位 cm/s						
VY:	Y 轴方向速度，单位 cm/s						
Mag:	0	1	2	Rc:	接收机		
	外置罗盘 1	外置罗盘 2	内置罗盘				
Pos:							
0	1	2	3	4	5	6	7
未定义	超声波	激光测距	外部气压计	内部气压计	未定义	RTK	GPS
光流	未定义	未定义	未定义	未定义	未定义	未定义	未定义

*符号说明:

✓ : 正常，已使用 (GPS 表示定位成功)

· : 已识别

× : 没连接/没识别/不正常

7 飞控安装

7.1 飞控固定

- 飞控支持免减震安装,需紧固在飞行器上,可使用 3M 胶固定飞控四个角即可,切勿只粘住飞控中间,飞控松动会影响飞行稳定性。

7.2 连接 USB

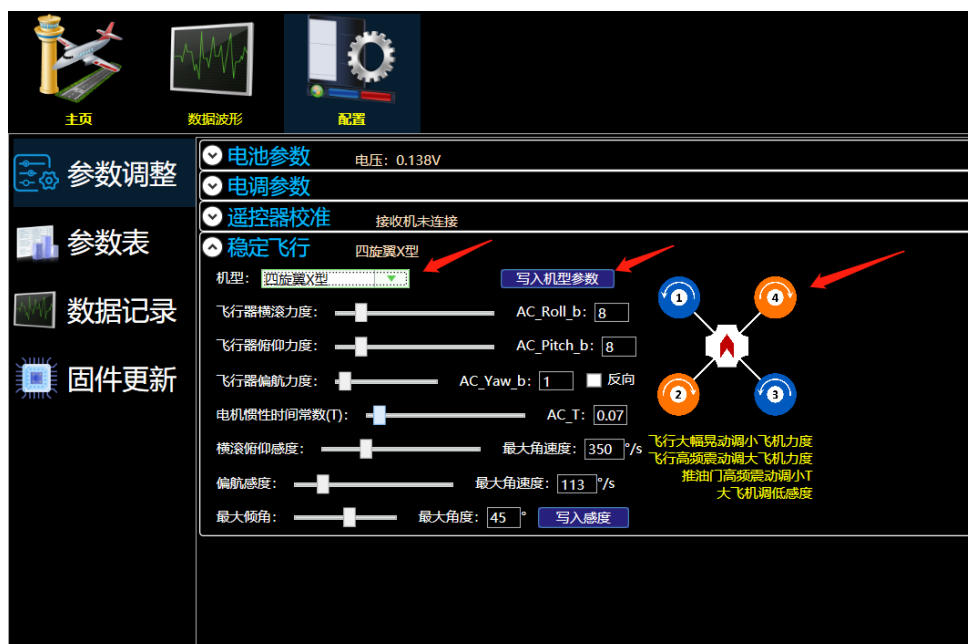
- 飞控通过 USB 和电脑连接后,电脑会出现一个虚拟串口和一个 U 盘(需插 SD 卡),串口用于连接地面站, U 盘是飞控的黑匣子,可用于导出 POS 记录、飞行日志(log)、ubx 文件和更新固件。
- 飞控检测到 USB 连接则停止 SD 卡记录,保证记录文件的一致性和可靠性。

7.3 上电初始化

- 飞控上电后会进行初始化,尽量保持静止状态,初始化过程会进行陀螺校准,此过程需要十几秒,等待初始化时状态灯蓝灯快闪烁,待状态灯绿色慢闪变化后代表自检完成。

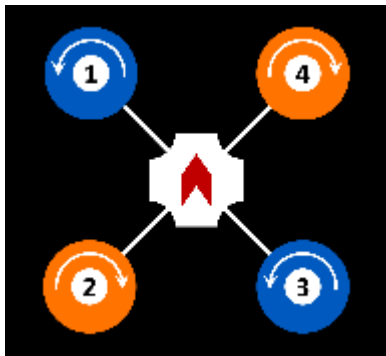
7.4 选择机型

- 将飞控用 USB 或数传连接到电脑,打开 ACFLy 调参地面站,在 COM 选择对应飞控端口号,点击连接。点击配置-参数调整-稳定飞行,下拉选择支持的机型。图片标明了电机序号(对应飞控 M1-M8 引脚)和电机转向和机头方向。



7.5 电机转向及顺序

- 以四旋翼为例，红色箭头为机头方向，飞控需按照下右图中的方向放置。飞控左上角电机为 1 号电机，六旋翼八旋翼类似，按逆时针顺序分别为 1、2、3、4。1 和 3 号电机逆时针旋转，2 和 4 号电机顺时针旋转。将飞控按上述方向紧固在飞行器上（需固定飞控四个角）。
- 分别将 1、2、3、4...号电机所连接的电调的信号线分别插在飞控板的 M1-M8 口。S 代表信号线（白色），5v 代表电源线（红色），G 代表地线（黑色）。

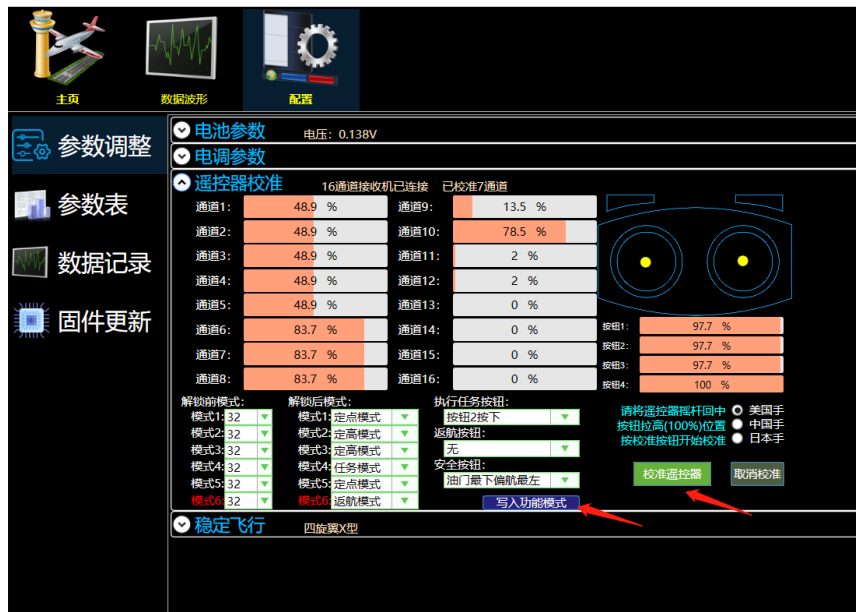


机头方向

7.6 连接遥控接收机

- 如使用 SBUS 接收机，将接收机的 SBUS 信号线（3 线）连接至飞控的 SBUS 排针口；
- 如使用 PPM 接收机，将接收机的 PPM 信号线（3 线）连接至飞控的 PPM 排针口。

8 遥控器校准



- 此飞控要求遥控器至少具有 6 个通道，包含 4 个摇杆和 2 个按钮，最多支持校准 8 个通道。校准的第一个按钮用来切换飞行模式。校准的第二、第三、第四个按钮用来设置执行任务、返航和安全按钮。
- 首先等待 5.1 中的飞控自检完成，飞控状态灯切换至绿灯慢闪。
- 将飞控用 USB 或者数传连接到电脑，打开 ACFLY 地面站，点击连接。
- 打开遥控器，连接接收机，部分遥控器需要提前设置好遥控器的 PPM/SBUS 模式，根据需要设置 2-3 个按钮，如果飞控识别到接收机，则地面站会显示遥控各通道的原始数值，识别不到则会显示 100%。
- 将四个摇杆通道回中，所要设置的 2-3 个按钮拨到通道值最大的位置。在 ACFLY 地面站中选择美国手、中国手或者日本手后，点击“校准遥控器”按钮。飞控进入遥控器校准模式。地面站有动画和文字提示，按照提示做即可。
- 等待蓝灯闪烁两次（蜂鸣器响两声），表示已记录所有摇杆初始位置
- 按如下顺序操作遥控器：（每次摇杆操作后摇杆需回中）
油门最下（哔）——>偏航最左（哔）——>俯仰最下（哔）——>横滚最左（哔）——>
油门最上（哔）——>偏航最右（哔）——>俯仰最上（哔）——>横滚最右（哔）——>
按钮 1 拨向 1 挡位置（哔）——>按钮 1 拨回原位——>按钮 2 拨向 1 挡位置（哔）——>按钮 2 拨回原位——>
按钮 3 拨向 1 挡位置（哔）（或把油门拉低完成较准）——>按钮 3 拨回原位——>按钮 4 拨向 1 挡位置（哔）
（或把油门拉低完成较准）——>按钮 4 拨回原位——>完成（绿灯闪哔哔）
- 错误情况：
红灯闪哔——长叫并退出模式：遥控器或接收机断开。
完成校准时红灯闪。

9 飞行模式设置

解锁前模式:		解锁后模式:		执行任务按钮:	
模式1:	32 ▼	模式1:	定点模式 ▼	按钮2变化 ▼	
模式2:	32 ▼	模式2:	无(不变) ▼	返航按钮:	
模式3:	32 ▼	模式3:	任务模式 ▼	按钮3变化 ▼	
模式4:	32 ▼	模式4:	无(不变) ▼	安全按钮:	
模式5:	32 ▼	模式5:	无(不变) ▼	油门最下偏航最左 ▼	
模式6:	32 ▼	模式6:	返航模式 ▼		

写入功能模式

- 遥控校准的第一个按钮用来切换飞行模式，定高、定点、任务和返航模式。校准的第二、第三、第四个按钮用来设置执行任务、返航和安全按钮(电机急停按钮)。
- 执行任务按钮是第二种进入任务模式的方式，在定位成功后按下执行任务按钮即可执行任务模式(需设置起飞点)。返航按钮则是执行一键返航功能。安全按钮为电机急停按钮。
- 第二、三、四个按钮有按下和变化两种设置(默认按下即可)，选择按下则对应通道值最大时为按下，选择变化则无论按钮在哪档，只要有变化(三档按钮变化一档即可)则执行对应功能。
- 若无需二次开发，解锁前模式都设置为 32 即可。

10 电调校准



为了安全起见，请先卸桨!!!

为了安全起见，请先卸桨!!!

(1) 电调油门行程校准

打开 ACFLY 调参地面站，将飞控通过 USB 或者数传与地面站连接。更改参数 Init_CalibESC_T 值为电调说明书对应的油门行程校准时间(好盈电调设置为 1.5 秒)，在配置-参数调整-电调参数栏目中，点击校准电调。飞控断电后，用电池给飞控和电机同时上电，然后根据电调说明书上的校准反应判断是否校准成功(好盈电调会发出滴滴-滴的声音)。

(2) 电调开/关 DEO(电调刹车，乐天 Pro 电调支持):

请先确认使用的电调是否支持 DEO 功能!!! 若不支持请忽略此小节。

将参数 Init_CalibESC_T 值改为电调开或关 DEO 所需的校准时间(单位：秒，不同电调所需的校准时间可能不一样，具体看自己使用的电调的说明书，自锁桨请关闭 DEO)。飞控断电后，用电池给飞控和电机同时上电，然后根据电调说明书上的校准反应判断是否校准成功。

(3) 起转油门和非线性系数设置:

天行者、铂金等非多旋翼电调必须进行起转油门设置，一般天行者起转油门设为 15。其他多旋翼电调不用设置起转油门。修改完需点击写入电调参数。

非线性系数和电调刹车设置有关，普通电调(未开刹车)非线性系数一般为 0.45，开刹车一般为 0.1，DJI 电调一般为 0.75。修改完需点击写入电调参数。

11 加速度计校准

- 出厂已进行过加速度计校准，一般情况无需用户校准。
- 遥控器油门最小，偏航最左，俯仰最下，横滚最左，两秒左右将进入加速度校准模式。
- 将无人机分别摆放六个面（不用按顺序，差不多水平即可）静止。每个面需静止放置 5 秒左右，静止放置时飞控指示灯显示蓝色，会由暗变亮指示当前进度，表示正在采集数据校准，完成一个面校准后指示灯会闪烁然后变红并且滴一声，然后更换下一个面校准。全部六面校准完毕后会退出校准模式，绿色慢闪。灯红色表示当前面已经校准或者飞控在移动无法校准。
- 需重启飞控校准才会生效。

12 磁罗盘校准

- 首先等待飞控初始化完成，飞控状态灯切换至绿色慢闪。遥控器油门最小，偏航最左，俯仰最上，横滚最左。两秒左右将进入磁罗盘校准模式。
- 旋转无人机尽量多的三维角度，旋转时飞控指示灯显示蓝色，会由暗变亮指示当前进度。指示灯红色表示飞控没有在旋转无法校准，继续旋转无人机即可。
- 完成后飞控指示灯闪烁并滴一声表示校准完成，绿色慢闪。
- 需重启飞控校准才会生效。

13 陀螺校准

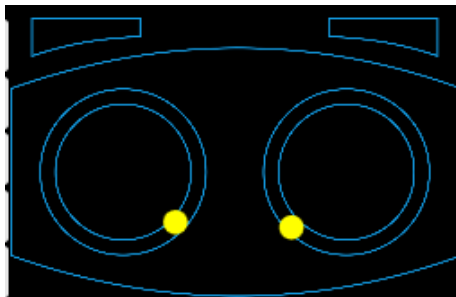
- 首先等待 5.1 中的飞控初始化完成，飞控状态灯切换至绿色慢闪。
- 遥控器油门最小，偏航最右，俯仰居中，横滚最右，两秒左右，滴一声进入陀螺校准模式，蓝绿快闪。
- 飞控需放置静止不动等待 5 秒左右，滴滴后校准完成将退出校准模式，绿色慢闪。
- 如果校准过程中飞控不静止，将会校准失败，出现红灯并滴一声退出校准，重新打杆校准即可。
- 需重启飞控校准才会生效。

14 标准电压和电量电压设置(必须设置)

参数名	定义	单位	说明
Bat_VoltMKp	电压比例系数	V/V	默认 11
Bat_STVoltage	电池标准电压	V	<p>默认为 11.6，如果检测到的电池电压大于或小于标准电压参数的 30% 无法解锁飞控，不接电压检测则无此限制。</p> <p>飞行器使用电池的标准电压，如 6s 电池为 22.8v，4S 电池为 15.2V，3S 电池为 11.6V。此值用于动态调整姿态控制器 b 参数。</p>
Bat_CurrentMKp	电流比例系数	A/V	标准 50
Bat_Capacity	电池容量	mAh	暂未使用
Bat_VoltP0 ~ Bat_VoltP10	十级电量百分比对应的电压	V	<p>相对 Bat_STVoltage 的电压偏移值。Bat_VoltP0 为电量为 0% 时的电压偏移值，Bat_VoltP1 为电量为 10% 时的电压偏移值。以此类推至 Bat_VoltP10 为电量为 100% 时的电压偏移值。</p>

15 解锁上锁

- 此为美国手和日本手的解锁手势，中国手的上锁手势。



- 此为美国手和日本手的上锁手势，中国手的解锁手势。



- 解锁后油门保持在中立位，电机依次从 1、2、3...开始旋转到怠速（电调需已校准），如怠速不稳定，加大起转油门百分比 AC_STThr。
- 降落后拉低油门自动上锁。
- 如遇到翻机或倒扣等炸鸡情况，可使用上锁手势强制上锁，因此飞行过程中严禁上锁操作！

16 固件更新

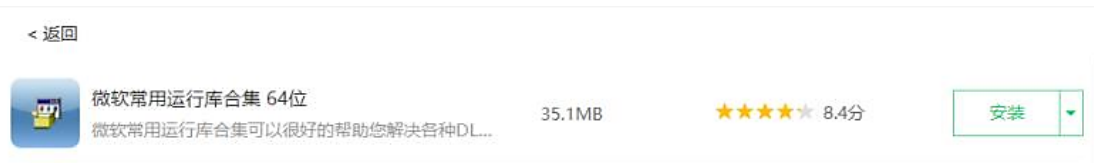
16.1 SD 卡更新飞控固件(推荐)

飞控用 USB 连接电脑，电脑会弹出一个 U 盘(飞控 SD 卡)，将固件文件(.hex)放进 U 盘里的 ACFLy 文件夹中，然后重启飞控，等待固件升级完成，升级过程中飞控蓝灯快闪，此过程需 12 秒左右。固件更新完会自动删除 ACFLy 文件夹里的固件，以免下次上电重复更新。

如果 SD 卡无法更新固件请使用 11.2 节方法更新固件。

16.2 USB 更新飞控固件

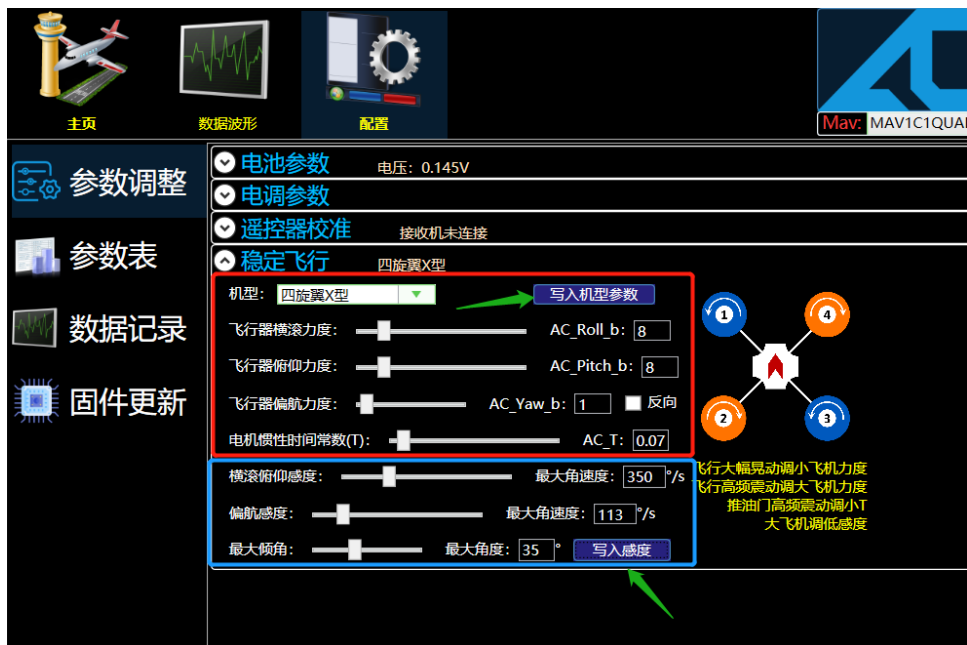
- 网站下载 ACFLY 地面站压缩包，其中包括名为 DFU 驱动的压缩包，解压后安装对应的版本。Win10 系统安装 win8.1 版本。
- 打开 ACFLY 地面站，点击配置-固件更新-浏览 双击选择固件(.hex)
- 插上飞控 USB，等待烧录完成。
- 注意：需要飞控先断电，地面站选择固件后 10 秒内将飞控通过 USB 连接至电脑。
- 部分用户电脑无法烧录固件，需按照如下微软常用运行库合集。



16.3 BootLoader 固件更新

- 飞控使用 USSB 连接上电脑后会弹出一个 U 盘，打开 ACFLY 文件夹，把 BootLoader 固件放进 BootLoaderUpdate 文件夹里，然后给飞控重新上电。

17 高级调参



- ACFLy 调参地面站支持图形化调参，用 USB 或者数传连接飞控后，点击连接-配置-参数调整-稳定飞行。地面站会自动加载飞控参数并显示。
- 一般情况下，所有机架都可使用默认参数起飞，后根据飞行情况调试以下参数。
- 无人机高频振荡发抖需将 b 参数调大，一般只需调 AC_Roll_b、AC_Pitch_b。
- 无人机打杆软绵绵没力需将 b 参数调小，一般只需调 AC_Roll_b、AC_Pitch_b。
- T 参数（参数名：AC_T）：飞行器电机的惯性时间常数。飞机桨加速至期望值的时间越长，此参数越大。此参数过小飞行器会高频振荡发抖。针对特定机型，此参数需微调。此参数越小（电机加速快），抗扰性能越好，此参数太小会导致 b 怎么调都会有震荡现象。此参数大不会震荡但是抗扰性能会打折扣（适中就行，没必要追求太强抗扰）。
- 感度调节：感度越大，无人机打杆反应速度越快。感度按照需求调节。感度设置好后需点击写入感度按钮等待参数写入成功。
- 最大倾角建议 35 度。

飞机配置	AC_T	AC_Roll_b/AC_Pitch_b	AC_Yaw_b
F330+朗宇x2212 1400kv电机+8038桨+3s	0.1	7.5	1.0
F450+U2216 800kv电机+1147桨+4s	0.1	5.5	1.0
F450+DJI 2312 940kv电机+9450桨+4s	0.1	7.5	1.0
精灵3+DJI 2312 940kv电机+9450自锁桨+3s	0.1	8.5	2.0
DH600+4114 400kv电机+飞越1555折叠桨+6s	0.05	5.5	1.0
Tarot X6+朗宇4108 320kv电机+1855碳桨+6s	0.15	1.2	0.8
F550 六轴+DJI 2312电机+9450自锁桨+4s	0.1	5.5	1.0

此参数表仅做参考，根据实际效果细调

18 常用参数表

参数名	定义	单位	说明
AC_UAVType	飞行器类型		10: 四旋翼 X 型; 11: 六旋翼 X 型; 12: 八旋翼 X 型; 15: 四旋翼十字型; 20: 四旋翼 Double 十字型; 32: 六旋翼异构
AC_STThr	怠速油门	%	默认 10%
Init_CalibESC_T	电调校准延时	秒	1.5: 电调校准 5: 电调开 DEO 9: 电调关 DEO
Bat_VoltMKp	电压比例系数	V/V	标准 18.4
Bat_STVoltage	电池标准电压	V	飞行器使用电池的标准电压, 如 6s 电池为 22.8v, 4S 电池为 15.2V, 3S 电池为 11.6V。此值用于动态调整姿态控制器 b 参数。
Bat_CurrentMKp	电流比例系数	A/V	标准 50
Bat_Capacity	电池容量	mAh	暂未使用
Bat_VoltP0 ~ Bat_VoltP10	十级电量百分比对应的电压	V	相对 Bat_STVoltage 的电压偏移值。Bat_VoltP0 为电量为 0% 时的电压偏移值, Bat_VoltP1 为电量为 10% 时的电压偏移值。以此类推至 Bat_VoltP10 为电量为 100% 时的电压偏移值。
Sf_FcRTLVolt	返航电压	V	低于此电压时自动返航
Sf_FcLandVolt	降落电压	V	低于此电压时自动降落
Aux_12Func	通道 12 功能		33: 相机拍照
Aux_CamOffPwm	拍照关	PWM	默认 1100; 使用电平拍照时为 60000
Aux_CamOnPwm	拍照开	PWM	默认 2000; 使用电平拍照时为 0
Aux_CamShTime	拍照时间	秒	建议 0.1
PC_AutoVXY	航线水平速度	cm/s	默认 1000,10m/s
PC_AutoVXYZ	航线三维限制速度	cm/s	默认 1000,10m/s, 建议与 PC_AutoVXY 一致
PC_AutoVZUp	航线上升速度	cm/s	默认 450,4.5m/s
PC_AutoVZDn	航线下降速度	cm/s	默认 350,3.5m/s
PC_LandVel	末端降落速度	cm/s	距离地面 10 米内的末端降落速度, 建议 30-50cm/s
Sf_RtSpeed	返航速度	cm/s	默认 1000,10m/s
Sf_GbRtHRange	返航高度的作用范围 (距离家)	cm	范围内按当前飞行高度直接返航, 忽略返航高度参数。范围外自动判断, 如果当前高度低于返航高度, 则先升高再返航, 如果当前高度高于返航高度, 则按当前高度返航。 默认 3000, 即 30m
Sf_GbRtHeight	返航高度	cm	默认 5000,50m
PC_maxVelXY	手动控制时的最大水平速度	cm/s	默认 1500,15m/s

PC_maxVelUp	手动控制时的最大上升速度	cm/s	默认 500,5m/s
PC_maxVelDn	手动控制时的最大下降速度	cm/s	默认 400,4m/s
AC_maxLean	最大倾斜角度	°	默认 35°
AC_Pitch_b	俯仰控制 b 参数		默认 5.5, 高频振荡发抖需将 b 参数调大, 打杆软绵绵没力需将 b 参数调小
AC_Roll_b	横滚控制 b 参数		默认 5.5, 高频振荡发抖需将 b 参数调大, 打杆软绵绵没力需将 b 参数调小
GPS0_delay	GPS 延时	s	默认 0.1 M8N: 0.1
GPS0_GNSS	GPS 卫星配置		默认 0, 不改变 GPS+GLONASS: 65 GPS+GLONASS+GALILEO: 69 GPS+BEIDOU: 9 GPS+BEIDOU+GALILEO: 13(M8N 推荐) GPS+GLONASS+BEIDOU+GALILEO: 77(仅 F9P/RTK 适用)
Lk_Uart1Func	串口 1 功能		0: 无功能 1: 数传 12: gps 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMini 定高 100:用作二次开发接口 SDI 默认: 1; 范围: 0 - 255。
Lk_Uart1Param	串口 1 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 115200; 范围: 0 - 255。
Lk_Uart3Func	串口 3 功能		0: 无功能 1: 数传 12: gps 32: LC302 光流 33: LC306 光流 34: JL32xx 光流

			35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 100 范围: 0 - 255。
Lk_Uart3Param	串口 3 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart5Func	串口 5 功能		0: 无功能 1: 数传 12: gps 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 32; 范围: 0 - 255。
Lk_Uart5Param	串口 5 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart7Func	串口 7 功能		0: 无功能 1: 数传 12: gps 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高

			100:用作二次开发接口 SDI 默认: 52; 范围: 0 - 255。
Lk_Uart7Param	串口 7 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart8Func	串口 8 功能		0: 无功能 1: 数传 12: gps 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 12; 范围: 0 - 255。
Lk_Uart8Param	串口 8 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Aux_1Func	对应飞控 C1 通道		PWM 输出, 对应飞控 C1 通道, 单位: 无; 1-16: 映射遥控器对应通道: 1: 在 PWM 通道 1 映射遥控器通道 1 信号值 2: 在 PWM 通道 1 映射遥控器通道 2 信号值 ... 25-40: 用遥控器对应通道进行相机快门触发: 25: 用遥控器通道 1 在 PWM 通道 1 控制相机快门 26: 用遥控器通道 2 在 PWM 通道 1 控制相机快门 ... 49-64: 用遥控器对应通道进行云台控制: 49: 用遥控器通道 1 在 PWM 通道 1 进行云台控制。

		50: 用遥控器通道 2 在 PWM 通道 1 进行云台控制。 ...
Aux_2Func	对应飞控 C2 通道	PWM 输出, 对应飞控 C2 通道, 单位: 无; 1-16: 映射遥控器对应通道: 1: 在 PWM 通道 2 映射遥控器通道 1 信号值 2: 在 PWM 通道 2 映射遥控器通道 2 信号值 ... 25-40: 用遥控器对应通道进行相机快门触发: 25: 用遥控器通道 1 在 PWM 通道 2 控制相机快门 26: 用遥控器通道 2 在 PWM 通道 2 控制相机快门 ... 49-64: 用遥控器对应通道进行云台控制: 49: 用遥控通道 1 在 PWM 通道 2 进行云台控制。 50: 用遥控通道 2 在 PWM 通道 2 进行云台控制。 ...
Aux_3Func	对应飞控 C3 通道	PWM 输出, 对应飞控 C3 通道, 单位: 无; 1-16: 映射遥控器对应通道: 1: 在 PWM 通道 3 映射遥控器通道 1 信号值 2: 在 PWM 通道 3 映射遥控器通道 2 信号值 ... 25-40: 用遥控器对应通道进行相机快门触发: 25: 用遥控器通道 1 在 PWM 通道 3 控制相机快门 26: 用遥控器通道 2 在 PWM 通道 3 控制相机快门 ... 49-64: 用遥控器对应通道进行云台控制: 49: 用遥控通道 1 在 PWM 通道 3 进行云台控制。 50: 用遥控通道 2 在 PWM 通道 3 进行云台控制。 ...
Aux_4Func	对应飞控 C4 通道	PWM 输出, 对应飞控 C4 通道, 单位: 无; 1-16: 映射遥控器对应通道: 1: 在 PWM 通道 4 映射遥控器通道 1 信号值 2: 在 PWM 通道 4 映射遥控器通道 2 信号值 ... 25-40: 用遥控器对应通道进行相机快门触发: 25: 用遥控器通道 1 在 PWM 通道 4 控制相机快门 26: 用遥控器通道 2 在 PWM 通道 4 控制相机快门 ... 49-64: 用遥控器对应通道进行云台控制: 49: 用遥控通道 1 在 PWM 通道 4 进行云台控制。 50: 用遥控通道 2 在 PWM 通道 4 进行云台控制。 ...
Aux_5Func	对应飞控 C5 通道	PWM 输出, 对应飞控 C5 通道, 单位: 无; 1-16: 映射遥控器对应通道: 1: 在 PWM 通道 5 映射遥控器通道 1 信号值 2: 在 PWM 通道 5 映射遥控器通道 2 信号值 ...

		<p>25-40: 用遥控器对应通道进行相机快门触发:</p> <p>25: 用遥控器通道 1 在 PWM 通道 5 控制相机快门</p> <p>26: 用遥控器通道 2 在 PWM 通道 5 控制相机快门</p> <p>...</p> <p>49-64: 用遥控器对应通道进行云台控制:</p> <p>49: 用遥控通道 1 在 PWM 通道 5 进行云台控制。</p> <p>50: 用遥控通道 2 在 PWM 通道 5 进行云台控制。</p> <p>...</p>
Aux_6Func	对应飞控 C6 通道	<p>PWM 输出, 对应飞控 C6 通道, 单位: 无;</p> <p>1-16: 映射遥控器对应通道:</p> <p>1: 在 PWM 通道 6 映射遥控器通道 1 信号值</p> <p>2: 在 PWM 通道 6 映射遥控器通道 2 信号值</p> <p>...</p> <p>25-40: 用遥控器对应通道进行相机快门触发:</p> <p>25: 用遥控器通道 1 在 PWM 通道 6 控制相机快门</p> <p>26: 用遥控器通道 2 在 PWM 通道 6 控制相机快门</p> <p>...</p> <p>49-64: 用遥控器对应通道进行云台控制:</p> <p>49: 用遥控通道 1 在 PWM 通道 6 进行云台控制。</p> <p>50: 用遥控通道 2 在 PWM 通道 6 进行云台控制。</p> <p>...</p>
Aux_7Func	对应飞控 C7 通道	<p>PWM 输出, 对应飞控 C7 通道, 单位: 无;</p> <p>1-16: 映射遥控器对应通道:</p> <p>1: 在 PWM 通道 7 映射遥控器通道 1 信号值</p> <p>2: 在 PWM 通道 7 映射遥控器通道 2 信号值</p> <p>...</p> <p>25-40: 用遥控器对应通道进行相机快门触发:</p> <p>25: 用遥控器通道 1 在 PWM 通道 7 控制相机快门</p> <p>26: 用遥控器通道 2 在 PWM 通道 7 控制相机快门</p> <p>...</p> <p>49-64: 用遥控器对应通道进行云台控制:</p> <p>49: 用遥控通道 1 在 PWM 通道 7 进行云台控制。</p> <p>50: 用遥控通道 2 在 PWM 通道 7 进行云台控制。</p>
Aux_8Func	对应飞控 C8 通道	<p>PWM 输出, 对应飞控 C8 通道, 单位: 无;</p> <p>1-16: 映射遥控器对应通道:</p> <p>1: 在 PWM 通道 8 映射遥控器通道 1 信号值</p> <p>2: 在 PWM 通道 8 映射遥控器通道 2 信号值</p> <p>...</p> <p>25-40: 用遥控器对应通道进行相机快门触发:</p> <p>25: 用遥控器通道 1 在 PWM 通道 8 控制相机快门</p> <p>26: 用遥控器通道 2 在 PWM 通道 8 控制相机快门</p> <p>...</p> <p>49-64: 用遥控器对应通道进行云台控制:</p> <p>49: 用遥控通道 1 在 PWM 通道 8 进行云台控制。</p> <p>50: 用遥控通道 2 在 PWM 通道 8 进行云台控制。</p>

19 自动返航策略

一键返航、低电量自动返航和失控自动返航都遵循以下返航策略：

- 无人机在距 Home 点 Sf_GbRtHRange(默认 30 米)的水平距离范围内无视返航高度参数，直接返航。
- 范围外按照 Sf_GbRtHeight 高度参数进行判断。当前高度低于 Sf_GbRtHeight 时，无人机先升高至 Sf_GbRtHeight，再按照 Sf_RtSpeed 速度进行返航。当前高度高于 Sf_GbRtHeight 时，无人机直接以当前高度，按照 Sf_RtSpeed 速度进行返航。

19.1 一键自动返航

- 当设置了一键返航按钮并按下后，无人机开始自动返航，返航过程可以打任意摇杆中断，如果一键返航设置的是按下，中断后如果 3 秒内不打摇杆则会继续自动返航。如果一键返航设置的是变化，则打杆中断会取消返航，需要重新拨一下一键返航按钮继续返航。

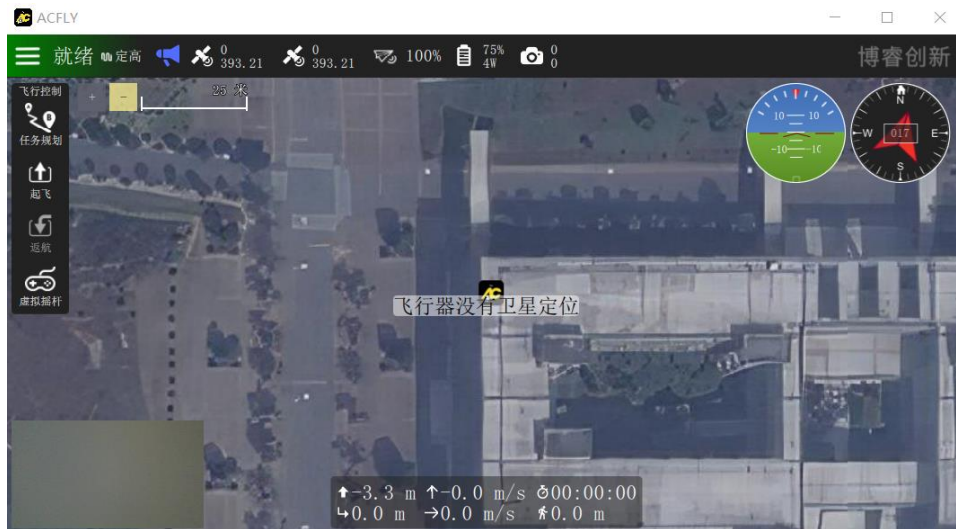
19.2 低电量自动返航或降落

- 设置安全模式参数 Sf_FcLandVolt 和 Sf_FcRTLVolt，单位：伏。
- 当无人机检测电池电压低于 Sf_FcRTLVolt 时，无人机将自动返航。
- 当无人机检测电池电压低于 Sf_FcLandVolt 时，无人机将自动降落。
- 如果上面两个参数设置的值一样，则会优先降落。

19.3 失控自动返航

- 如果无人机不是在任务模式，飞行过程中遥控器断联无人机会自动返航，使用 PPM 接收机则不支持此功能(因为 PPM 接收机在遥控断联后仍有输出)，返航过程中遥控器恢复连接会继续返航，可手动打杆夺取控制权。
- 执行任务过程中遥控器断联后无人机会继续执行任务(需有定位)，执行完任务后或者电量低于 Sf_FcRTLVolt 时无人机会自动返航。
- 如果无定位且遥控断联，无人机则会自动降落。

20 航线任务设置



- 飞控使用的是 mavlink 协议，支持主流开源地面站设置航点，包括 ACFLY GCS 地面站(PC 和安卓版本)、Mission Planner、QGC(PC 和安卓版本)等。最多设置 65536 个航点。
- 航点设置完成，无人机定位成功(蓝绿交替闪烁)后，模式按钮打到任务模式挡位，若设置的第一个任务是起飞任务，则解锁无人机即可自动起飞执行任务。若第一个任务不是起飞任务，则需手动起飞后无人机再切换到任务模式。

21 断电/断点续飞

(1) 断电续飞

如果参数 MFunc_RstWp0 为 0(默认)，则在执行任务过程中由于低电量自动返航或人为返航等原因中断任务执行后，飞机会存储当前航点信息，在下一次上电时将此航点设置为第一个任务。如果参数 MFunc_RstWp0 不为 0，则上电后从头开始执行航线任务。

(2) 断点续飞

如果在任务模式过程中打杆干预无人机，无人机会退出任务模式并自动记录断点处的经纬度，再次进入任务模式后飞机会从断点处开始继续执行任务。

22 相机触发拍照配置

参数名	定义	单位	说明
Aux_CamOffPwm	拍照关	PWM	默认 1100；使用电平拍照时为 60000
Aux_CamOnPwm	拍照开	PWM	默认 2000；使用电平拍照时为 0
Aux_CamShTime	拍照时间	秒	建议 0.1

- 地面测试拍照功能可使用 ACFLY GCS 地面站，数传连接地面站后点击下图的相机图案即可，拍照成功飞控会滴滴提示两声，绿灯闪烁，拍照失败则长鸣一声并显示红灯。拍照成功 POS 会记录到飞控内部 SD 卡中，可插 USB 导出 POS 记录查看。(插着飞控 USB 将停止 SD 卡数据记录，无法写入 POS)。A9 飞控无热靴功能。



23 POS 记录

- POS 记录将自动采用精度最高的 POS 信息进行记录。
- POS 格式兼容大疆精灵 4rtk 格式，使用 USB 的 U 盘功能可导出 pos 文件。

24 飞控及定位传感器安装偏移

飞控机体坐标系为前左上，前为 X 轴正方向(机头)，左为 Y 轴正方向，上为 Z 轴正方向。偏移坐标系和飞控坐标系相同。单位：cm。偏移区分正负，请按照坐标系区分。

参数名	定义	单位	说明
POfs_Fc_x	安装偏移	cm	飞控安装距离飞机中心的 x 轴偏移
POfs_Fc_y	安装偏移	cm	飞控安装距离飞机中心的 y 轴偏移
POfs_Fc_z	安装偏移	cm	飞控安装距离飞机中心的 z 轴偏移
POfs_S0_x - POfs_S15_x	安装偏移	cm	0-15 号位置传感器安装距离飞控的 x 轴偏移
POfs_S0_y - POfs_S15_y	安装偏移	cm	0-15 号位置传感器安装距离飞控的 y 轴偏移
POfs_S0_z - POfs_S15_z	安装偏移	cm	0-15 号位置传感器安装距离飞控的 z 轴偏移

定位传感器名称	ID
超声波	1 号
光流	8 号
外置气压	3 号
GPS	7 号
测距(激光/毫米波)	2 号

25 串口功能配置

默认串口 1 为数传，波特率为 Lk_Uart1Param: 115200;

参数名	定义	单位	说明
Lk_Uart1Func	串口 1 功能		0: 无功能 1: 数传 12: gps 13: rtk 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 1; 范围: 0 - 255。
Lk_Uart1Param	串口 1 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 115200; 范围: 0 - 255。
Lk_Uart3Func	串口 3 功能		0: 无功能 1: 数传 12: gps 13: rtk 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 100 范围: 0 - 255。
Lk_Uart3Param	串口 3 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴;

			2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart5Func	串口 5 功能		0: 无功能 1: 数传 12: gps 13: rtk 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 32; 范围: 0 - 255。
Lk_Uart5Param	串口 5 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart7Func	串口 7 功能		0: 无功能 1: 数传 12: gps 13: rtk 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMMini 定高 100:用作二次开发接口 SDI 默认: 52; 范围: 0 - 255。
Lk_Uart7Param	串口 7 参数		如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴;

		3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。
Lk_Uart8Func	串口 8 功能	0: 无功能 1: 数传 12: gps 13: rtk 32: LC302 光流 33: LC306 光流 34: JL32xx 光流 35: GL9306 光流 52: TFMini 定高 100: 用作二次开发接口 SDI 默认: 12; 范围: 0 - 255。
Lk_Uart8Param	串口 8 参数	如果该串口为数传功能, 则此参数为波特率(1000 以上的波特率); 如果该串口为光流接口, 则此参数为光流方向: (飞控坐标系: 前左上(XYZ)) 0: 光流坐标系和飞控坐标系一样; 1: 飞控 X 轴 = 光流 Y 轴; 飞控 Y 轴 = -光流 X 轴; 2: 飞控 X 轴 = -光流 X 轴; 飞控 Y 轴 = -光流 Y 轴; 3: 飞控 X 轴 = -光流 Y 轴; 飞控 Y 轴 = 光流 X 轴; 默认: 0; 范围: 0 - 255。

26 AUX 参数功能

参数名	定义	说明
Aux_1Func- Aux_8Func	对应飞控 M1-M8 通道	1-16: 映射遥控器对应通。1-16: 对应遥控通道 1-16 25-40: 用遥控器对应通道进行相机快门触发, 25-40 对应遥控通道 1-16。 49-64: 用遥控器对应通道进行云台控制。49-64 对应遥控通道 1-16。

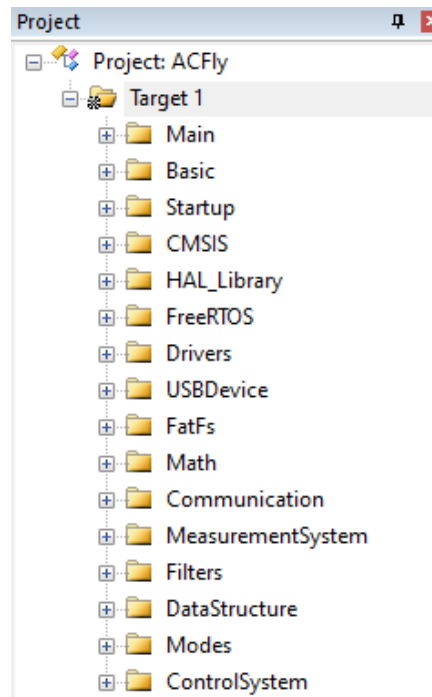
27 低电量报警功能

- 设置返航电压后(需大于 5V)则开启低电量报警功能。
- 检测到的电压(需大于 7V)低于返航电压则报警(红灯闪烁+蜂鸣器持续快响)。
- 飞控发送 “Low Power”警告提示到上位机。

28 代码框架

28.1 代码总体布局

本飞控代码已经分组在 14 个大类里，如图所示。建议看的部分：（3）驱动中的 Sensors 接口及 Receiver 接口；（5）Basic 中的时间实现；（7）解算系统中的解算系统接口；（11）模式中的飞行模式；（12）控制系统中的控制系统接口。



（1）CMSIS（板级支持包）

板级支持文件，最底层的库，不用看基本不用修改（可以修改 startup.s 里面的堆栈设置）。

（2）Main（主函数文件）

主函数包括：

- 初始化所有需要用到的外设；
- 开启任务调度器；
- 错误中断拉低所有输出。

（3）Driver（驱动）

驱动包含：

- 外设的初始化配置（drv_开头文件）
- 传感器接口，包括（建议细看，二次开发必备）：
 - Sensors.c: 传感器接口实现函数
 - Sensors.h: 传感器读取接口函数声明，建议细看
 - Sensors_Backend.h: 传感器注册、更新接口函数声明，建议细看
- 接收机接口，包括：
 - Receiver.c: 接收机接口实现函数

- Receiver.h: 接收机读取接口函数声明, **建议细看**

(4) HAL_Library

32HAL 库, 二次开发需要用到的但现工程没有的请自己加进工程。

(5) Basic (基本)

Basic.c 里面初始化 TIM5 定时器用于计时, 实现了 TIME 结构体用于时间计算, 其他部分程序所有时间相关操作都是基于 TIME, **建议细看**。

(6) Math (数学库)

包含四元数、三维向量运算, 以及一些简单的数学运算, 重力等常量的定义。

(7) MeasurementSystem (解算系统)

姿态解算及位置解算。

建议细看解算系统接口 MeasurementSystem.h, 包含解算结果的获取函数声明及使用说明。

(8) Filters (滤波器)

包含巴特沃斯低通滤波器、TD4 非线性滤波器实现。

(9) DataStructure (数据结构)

包含环形缓冲区的实现。

(10) Communic (通讯)

包含 Mavlink 库、调试通讯文件 Debug.c、通用端口交互文件 Commulink.c(驱动程序可通过 Commulink.h 里的函数注册端口成为通用端口用于 mavlink 等标准通讯)

(11) Modes (模式)

建议细看飞行模式! 二次开发必备

- 0-9 号为非飞行非校准的其他模式
- 10-19 号为校准模式
- 30-39 号为飞行模式
- M00 为初始化模式, 等待解算系统初始化完成

(12) ControlSystem (控制系统)

建议细看 ControlSystem.h! 二次开发必备

ControlSystem.h 包含控制系统的 API 接口。Ctrl_Attitude 和 Ctrl_Position 分别为姿态和位置控制器。

29 代码接口

- 本代码接口中所有位置、速度、加速度数据单位均为 **cm** 厘米
- 本代码接口中所有角度、角速度、角加速度数据单位均为 **rad** 弧度
- 本代码接口中所有磁场数据单位均为 **Gauss** 高斯
- 本飞控模块化编程，不同模块通过接口进行访问操作，此文档必看!!
- 此文档包括：传感器接口、接收机接口

29.1 传感器接口

- 位于 Drivers 目录下
- 传感器接口分为：传感器读取接口 和 传感器注册及更新接口。
- 函数声明分别位于：Sensors.h 和 Sensors_Backend.h 里
- 函数定义位于：Sensors.c 里

(1) IMU 传感器定义 (Sensors.h)

IMU 传感器包括加速度计、陀螺仪、磁力计，每种各支持 3 个。定义如下：

```

11  /*IMU传感器定义*/
12  typedef struct
13  {
14      bool present; //传感器是否存在
15      TIME last_update_time; //上次更新时间
16      float sample_time; //采样时间
17
18      float sensitivity; //灵敏度 (原始数据->实际单位 陀螺: rad/s 加速度: cm/s^2 磁场: gauss)
19
20      vector3_int data_raw; //原始数据
21      vector3_float data; //实际单位数据
22  }IMU_Sensor;
23  /*IMU传感器定义*/

```

(2) 位置传感器定义 (Sensors.h)

位置传感器包括气压、光流、超声波等，本飞控最多支持同时存在 8 个定位传感器。定义如下：

```

53  typedef struct
54  {
55      bool publishing; //是否正在更新
56
57      bool present; //传感器是否存在
58      bool available; //传感器是否可用
59      TIME last_update_time; //上次更新时间
60      TIME inavailable_start_time; //传感器不可用开始时间
61      float delay; //传感器延时
62      float sample_time; //采样时间
63
64      bool safe; //传感器是否安全 (数据缓慢变化不会发生跳变 !! 注意!! 如不确定不要设置为safe)
65      Position_Sensor_Type sensor_type; //传感器类型 (见枚举注释)
66      Position_Sensor_DataType sensor_DataType; //传感器数据类型 (见枚举注释)
67      Position_Sensor_frame velocity_data_frame; //速度数据坐标系 (见枚举注释)
68
69      vector3_double position_Global; //经纬度
70      vector3_float position; //位置 (cm)
71      vector3_float velocity; //速度 (cm/s)
72
73      //经纬度转平面坐标变量
74      Map_Projection mp;
75  }Position_Sensor;

```

其中：

- sensor_type 定义了传感器是经纬度定位、相对定位，还是测距定位传感器。
- sensor_DataType 定义了传感器的数据类型：例如 z 轴位置数据，xy 速度数据等。
- velocity_data_frame 针对速度传感器，定义了速度传感器所测速度所在的坐标系。

(3) IMU 传感器读取接口 (Sensors.h)

IMU 传感器读取接口会返回 const 的 IMU 传感器结构体：

```

84
85  /*IMU传感器读取函数*/
86  const IMU_Sensor* GetAccelerometer( unsigned char index );
87  const IMU_Sensor* GetGyroscope( unsigned char index );
88  const IMU_Sensor* GetMagnetometer( unsigned char index );
89  /*IMU传感器注册函数*/

```

(4) IMU 传感器注册、更新接口 (Sensors_Backend.h)

在 IMU 传感器更新前，首先调用注册函数进行注册，设置传感器的灵敏度。

注册完成后把 IMU 传感器编号及原始数据送入 update 接口即可完成更新。

```

11  /*IMU*/
12
13  /*IMU传感器注册函数*/
14  bool IMUAccelerometerRegister( unsigned char index , float sensitivity );
15  bool IMUGyroscopeRegister( unsigned char index , float sensitivity );
16  bool IMUMagnetometerRegister( unsigned char index , float sensitivity );
17  /*IMU传感器注册函数*/
18
19  /*IMU传感器更新函数*/
20  bool IMUAccelerometerUpdate( unsigned char index , vector3_int data );
21  bool IMUGyroscopeUpdate( unsigned char index , vector3_int data );
22  bool IMUMagnetometerUpdate( unsigned char index , vector3_int data );
23  /*IMU传感器更新函数*/

```

(5) 位置传感器读取接口 (Sensors.h)

位置传感器读取接口会返回 const 的位置传感器结构体：

```

95  /*位置传感器*/
96
97  /*位置传感器读取函数*/
98  const Position_Sensor* GetPositionSensor( unsigned char index );
99  /*位置传感器读取函数*/

```

(6) 位置传感器注册、更新接口 (Sensors_Backend.h)

在位置传感器更新前，首先调用注册函数进行注册，设置传感器的类型等参数：

```

29  /*位置传感器注册函数*/
30  //safe: 传感器是否安全（数据缓慢变化不会发生跳变！！注意！！如不确定不要设置为safe）
31  bool PositionSensorRegister(
32      unsigned char index ,\
33      Position_Sensor_Type sensor_type ,\
34      Position_Sensor_DataType sensor_data_type ,\
35      Position_Sensor_frame sensor_vel_frame ,\
36      float delay ,\
37      bool safe \
38  );
39  //注销传感器
40  bool PositionSensorUnRegister( unsigned char index );
41  /*位置传感器注册函数*/
42

```

如果位置传感器很久没有更新，MS_Main 解算任务中会自动把此传感器取消注册。

注册完成后把位置传感器编号及与传感器 sensor_data_type 对应的数据送入 update 接口即可完成更新：

```

49  /*位置传感器更新函数*/
50  //delay参数小于0则不会改变delay
51  bool PositionSensorUpdatePositionGlobal( unsigned char index , vector3_double p
52  bool PositionSensorUpdatePosition( unsigned char index , vector3_float position
53  bool PositionSensorUpdatePositionGlobalVel( unsigned char index , vector3_double
54  bool PositionSensorUpdatePositionVel( unsigned char index , vector3_float posit
55  bool PositionSensorUpdateVel( unsigned char index , vector3_float vel , bool av
56  /*IMU传感器更新函数*/
57
58

```


29.2 接收机接口

- 位于 Drivers 目录下
- 接收机接口分为：接收机读取接口 和 接收机更新接口。
- 函数声明分别位于：Receiver.h 和 Receiver_Backend.h 里
- 函数定义位于：Receiver.c 里

(1) 接收机定义 (Receiver.h)

接收机包含 SBUS、PPM 等协议的接收机（至少具有 6 个通道），定义如下：

```
6 //接收机定义
7 typedef struct
8 {
9     bool present; //是否存在
10    bool connected; //是否已连接
11    bool available; //是否可用
12    TIME last_update_time; //上次更新时间
13    float update_time; //更新时间间隔
14
15    float raw_data[16]; //原始数据
16    float data[8]; //校准后的数据
17 }Receiver;
```

(2) 接收机读取接口 (Receiver.h)

- get_Receiver 会返回指定接收机的 const 结构体
- get_current_Receiver 会返回当前接收机的 const 结构体（自动选择序号最低的可用接收机，无可用接收机是返回随机接收机）
- get_current_Receiver_Type 返回当前接收机的类型（SBUS 接收机、PPM 接收机等）

```
26 //获取指定的接收机
27 const Receiver* get_Receiver( RC_Type rc );
28 //获取当前使用的接收机
29 const Receiver* get_current_Receiver();
30 //获取当前使用的接收机
31 RC_Type get_current_Receiver_Type();
```

(3) 接收机更新接口 (Receiver_Backend.h)

把接收机类型、原始数据、是否已连接等信息发送给接口即可完成接收机数据更新。

```
8 //更新接收机数据
9 void Receiver_Update( RC_Type _rc , bool connected
```

30 代码二次开发教程

本飞控二次开发采用视频教程的方式，到售后群下载链接说明：

ACFLY售后群

群文件 > ACFLY 视频教程 (A9 & EDU)

文件



ACFLY 视频教程链接0916.pdf

31 版本更新日志

日期	新版本	旧版本	更新内容
20200427	V1.1	——	——
20200623	V1.3	V1.1	<ul style="list-style-type: none"> 更新第 7.2 节，参数分类增加了 PC---位置控制参数。 更新了 7.3 节，增加了调参教程内容。 更新了 7.4 节，增加了姿态参数及其说明。 更新了 7.5 节，增加了一些电池参数及其说明。 更新了 7.6 节，增加了位置控制参数及其说明。 更新了 7.7 节，增加了安全模式参数及其说明。(自动返航参数) 增加了第 8 大节：航点飞行和一键返航功能使用说明。 更新了 11 大节，飞控供电接口说明。
20200721	V1.4	V1.3	<ul style="list-style-type: none"> 更新第 11 节，优化供电接口说明 优化手册格式内容，条理更加清晰 优化手册图片内容，图片更加高清
20200910	V1.5	V1.4	<ul style="list-style-type: none"> 更新第五章校准流程 优化手册相关细节
20200925	V1.6	V1.5	<ul style="list-style-type: none"> 更新第五章 5.4 陀螺校准 优化配图说明 修改部分细节表述
20201119	V1.7	V1.6	<ul style="list-style-type: none"> 更新第 5 章、第 7 章和第八章 增加 4.5 节，屏幕显示内容说明 稳压连接挪到 4.4 节 优化配图说明 修改部分细节表述 ACFLY Prophet 飞控统一更名为 ACFLY A9 飞控
20201123	V1.8	V1.7	<ul style="list-style-type: none"> 增加 3.1 节、8.5 节和 8.7 节 修改 5.3 节任务模式描述 修改 7.10 节外置气压传感器 ID 号错误

20201202	V1.9	V1.8	<ul style="list-style-type: none">• 修改 7.8 节返航参数说明• 修改 8.3 节一键返航说明
20201207	V2.0	V1.9	<ul style="list-style-type: none">• 修改 5.4 节电调校准说明• 增加 7.11 节 Init 参数说明
20210320	V2.1	V2.0	<ul style="list-style-type: none">• 修改排版顺序• 增加第 20 大节串口功能配置参数说明
20210331	V2.1	V2.2	<ul style="list-style-type: none">• 增加第 2 大节开发环境搭建• 串口配置参数增加 RTK 配置• 修改全文 LOGO 与排版

ACFLY 开源飞控提供技术支持

内容如有更新，恕不另行通知

您可以在 ACFLY 开源飞控售后群或百度云链接查询最新版本《ACFLY A9 飞控用户手册》

Copyright © ACFLY 开源飞控 版权所有