

农用（植保）无人驾驶航空器通信协议技术规范

V1.0.1

2025年1月1日

目录

1. 适用范围	1
2. 规范性引用文件	1
3. 术语和定义	1
4. 缩略语	1
5. Protobuf 简要说明	2
6. 协议组成	2
6.1 IDL 定义 (.proto)	2
6.2 DataType 消息类型	6
6.3 ProtocolVersion 协议版本号	7
6.4 LoginInfo 登录信息	7
6.5 LoginResponse 登录响应信息	7
6.6 FlightRecord 飞行记录	8
6.7 FlightRecordDetail 飞行记录明细	10
6.8 UavInfo 无人机信息	12
7. 协议基础	13
7.1 通信方式	13
7.2 通信链接	13
7.3 通信 IP 和端口号	13
7.4 无人机鉴权	13

1. 适用范围

本协议适用于数据发送端与数据采集中心服务器之间的数据传输。该协议定义了农用（植保）无人驾驶航空器（以下简称无人机）飞行记录、飞行片段、位置数据、飞行参数等信息的传输格式和规则，确保数据能够准确、可靠地从数据发送端传输到服务器。

2. 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注明日期的引用文件，仅该日期对应的版本适用于本文件；未注明日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

3. 术语和定义

数据接收平台（Data receiving platform）：用于接收、解析、存储和处理数据发送端上报数据的服务平台。

Protocol Buffers：一种数据描述语言，类似于 XML 能够将结构化数据序列化，可用于数据存储、通信协议等方面。

UUID：全局唯一标识符，用于唯一标识一次飞行记录。

Timestamp：时间戳，表示特定时刻的时间，单位为毫秒。

LoginInfo：登录信息，每次连接以后发送登录信息，验证通过后发送其他数据信息。

LoginResponse：服务器响应信息。

FlightRecord：一次起降的飞行记录。

Position：位置数据，使用 WGS84 坐标系的经纬度。

Velocity：速度数据，包括北东地 x、y、z 轴上的速度。

SprayInfo：喷洒信息，包括喷洒类型、流速和剩余药量。

SowInfo：播撒信息，包括播撒类型、播撒模式、转速等。

TimeRange：时间范围，包含起始时间和截止时间。

4. 缩略语

IP：互联网协议（Internet Protocol）

TCP：传输控制协议（Transmission Control Protocol）

5. Protobuf 简要说明

Protocol Buffers (protobuf) 是一种用于数据序列化的开源机制。它提供了一种语言中立、平台中立、可扩展的方法，用于对结构化数据进行序列化和反序列化。与传统的 XML 或 JSON 格式相比，protobuf 具有以下优点：

高效：protobuf 序列化后的数据比 XML 或 JSON 更小，解析速度更快。

可扩展：支持向现有的数据结构中添加新字段，而无需更改原有的解析代码。

多语言支持：protobuf 支持多种编程语言，如 C、C++、Java、Python、Go 等。

6. 协议组成

协议使用 Protocol Buffers (protobuf) 定义消息格式，主要包括以下七个部分：

- (1) **DataType**: 枚举类型，表示具体的消息类型。
- (2) **ProtocolVersion**: 枚举类型，表示具体的协议版本类型。
- (3) **LoginInfo**: 登录信息，包括用户名、密码。
- (4) **LoginResponse**: 服务器响应信息，包括状态码、响应消息。
- (5) **FlightRecord**: 飞行记录，包括基本信息、时间信息、位置信息、飞手信息、工作参数和飞机信息。
- (6) **FlightRecordDetail**: 飞行记录详情，用于描述飞行中产生的具体作业数据。
- (7) **UavInfo**: 无人机信息，包括无人机 SN、机身编码、无人机型号、北斗标识、厂家编号。

6.1 IDL 定义 (.proto)

```
syntax = "proto3";  
  
package com.xx;  
  
// 主消息定义  
message Message {  
  
// 枚举类型, 消息类型  
  enum DataType {
```

```
LOGINRESPONSETYPE = 0;
LOGININFOTYPE = 1;
FLIGHTRECORDTYPE = 2;
FLIGHTRECORDDETAILTYPE = 3;
UAVINFOTYPE = 4;
}
```

```
enum ProtocolVersion {
    V1_0_0 = 0;
    V1_0_1 = 1;
}
```

```
DataType dataType = 1;
```

```
oneof dataBody {
    LoginInfo loginInfo = 2;
    FlightRecord flightRecord = 3;
    FlightRecordDetail flightRecordDetail = 4;
    LoginResponse loginResponse = 5;
    UavInfo uavInfo = 6;
}
```

```
ProtocolVersion version=7;
```

```
}
```

```
// 登录信息
```

```
message LoginInfo {
    string user_name = 1;
    string password = 2;
}
```

```
// 登录响应信息
```

```
message LoginResponse {
    int32 code = 1;
```

```
    string msg = 2;  
}
```

```
// 一次起降飞行记录
```

```
message FlightRecord {  
    string uuid = 1;  
  
    repeated SimplifyPoint simplify_points = 2;  
    int64 collect_time = 3;  
    int32 work_time = 4;  
    TimeRange start_end = 5;  
    int64 created_at = 6;  
    string location = 7;  
    Position position = 8;  
  
    float work_area = 9;  
    int32 usage_type = 10;  
    int32 spray_usage = 11;  
    int32 sow_usage = 12;  
    string drone_type = 13;  
    string hardware_id = 14;  
    string body_code = 15;  
    string battery_sn = 16;  
}
```

```
// 飞行记录详情
```

```
message FlightRecordDetail {  
    int32 count = 1;  
    string uuid = 2;  
    string hardware_id = 3;  
    string body_code = 4;  
    TimeRange start_end = 5;  
    repeated FlightData data = 6;
```

```

}

// 时间范围
message TimeRange {
    int64 start = 1;
    int64 end = 2;
}

// 位置信息
message Position {
    double longitude = 1;
    double latitude = 2;
}

// 抽稀后航点
message SimplifyPoint {
    Position position = 1;
    bool is_spray = 2;
    bool is_sow = 3;
    int32 work_mode = 4;

}

// 航点数据
message FlightData {
    int64 collect_time = 1;
    Position position = 2;
    Velocity velocity = 3;
    float height = 4;
    int32 work_mode = 5;
    float spray_width = 6;
    SprayInfo spray_info = 7;
    SowInfo sow_info = 8;
}

// 速度
message Velocity {
    float x = 1;
    float y = 2;
    float z = 3;
}

// 喷洒信息

```

```

message SprayInfo {
    int32 spray_type = 1;
    int32 flow_speed = 2;
    uint32 residual_pesticide = 3;
}
// 播撒信息
message SowInfo {
    int32 sow_type = 1;
    int32 sow_mode = 2;
    uint32 turn_speed = 3;
    uint32 hatch_size = 4;
    uint32 sow_speed = 5;
}
// 无人机信息
message UavInfo {

    string hardware_id = 1;
    string body_code = 2;
    string uav_mode = 3;
    string bd_identification = 4;
    string company_code = 5;

}

```

注：主消息定义中增加了“协议版本号”；另外增加了“无人机信息”消息。上述两项信息务必严格按照要求上报。

6.2 DataType 消息类型

DataType 枚举类型用于表示 Message 中包含的数据类型。每个数据类型都有唯一的枚举值，用于指示 dataBody 字段中包含的具体数据。

序号	字段名	字段类型	单位	备注
1	LOGINRESPONSETYPE	int32	-	表示登录响应信息
2	LOGININFOTYPE	int32	-	表示登录信息
3	FLIGHTRECORDTYPE	int32	-	表示飞行记录信息
4	FLIGHTRECORDDETAILTYPE	int32	-	表示飞行记录详情信息
5	UAVINFOTYPE	int32	-	表示无人机信息

6.3 ProtocolVersion 协议版本号

ProtocolVersion 枚举类型用于表示 Message 中包含的数据类型。每个数据类型都有唯一的枚举值，用于指示 version 字段中包含的具体数据。

序号	字段名	字段类型	单位	备注
1	V1_0_0	int32	-	表示协议版本为 v1.0.0
2	V1_0_1	int32	-	表示协议版本为 v1.0.1

6.4 LoginInfo 登录信息

序号	字段名	字段类型	单位	备注
1	user_name	string	-	登录用户名
2	password	string	-	登录密码

6.5 LoginResponse 登录响应信息

序号	字段名	字段类型	单位	备注
----	-----	------	----	----

序号	字段名	字段类型	单位	备注
1	code	int32	-	登录响应状态码 0: 成功 1: 失败
2	msg	string	-	登录响应消息

6.6 FlightRecord 飞行记录

FlightRecord 消息用于描述一次起降的飞行记录。它包含基本信息、时间信息、位置信息、飞手信息、工作参数和飞机信息。时间信息统一采用北京时间表示。

序号	字段名	字段类型	单位	备注
1	uuid	string	-	本次起降的 UUID
2	simplify_points	SimplifyPoint[]	-	抽稀航点，表示飞行轨迹上的抽稀点，用于表明实际作业时的飞行轨迹轮廓
3	collect_time	int64	ms	无人机本地创建第一个航点时间
4	work_time	int32	ms	工作时长
5	start_end	TimeRange	-	起始/截止时间戳
6	created_at	int64	ms	服务器创建时间戳
7	location	string	-	飞行地址
8	position	Position	-	飞行经纬度
9	work_area	float	亩	工作面积，精确到小数点后两位
10	usage_type	int32	-	任务类型 0: 喷洒 1: 播撒
11	spray_usage	int32	ml	喷洒用量
12	sow_usage	int32	g	播撒用量
13	drone_type	string	-	机型

序号	字段名	字段类型	单位	备注
14	hardware_id	string	-	无人机飞控序号
15	body_code	string	-	机身编码
16	battery_sn	string	-	创建该飞行记录时使用的电池SN

6.6.1 TimeRange 时间范围

TimeRange 消息用于时间范围，包括起始时间、截止时间。

序号	字段名称	字段类型	单位	备注
1	start	int64	时间戳	起始时间
2	end	int64	时间戳	截止时间

6.6.2 Position 位置信息

Position 消息用于描述 WGS84 坐标系下北斗定位坐标，包括经度、纬度。

序号	字段名称	字段类型	单位	备注
1	longitude	double	度	经度
2	latitude	double	度	纬度

6.6.3 SimplifyPoint 抽稀位置信息

SimplifyPoint 消息用于描述飞行片段，包括位置、是否有喷洒、是否有播撒。

序号	字段名称	字段类型	单位	备注
1	position	Position	-	北斗定位坐标
2	is_spray	bool	-	喷洒状态 0: 未工作 1: 工作中
3	is_sow	bool	-	播撒状态 0: 未工作 1: 工作中
4	work_mode	int32	-	飞行片段的工作模式 0: NULL 1: M (手动) 2: AB 3: M+ (手动增强) 4: WP (航线模式)

6.7 FlightRecordDetail 飞行记录明细

FlightRecordDetail 消息用于描述飞行中产生的具体作业数据。包含航点数量、本次起降的 UUID、无人机飞控序列号、机身编码和记录的实时航点数据。

序号	字段名	字段类型	单位	备注
1	count	int32	-	航点数量计数
2	uuid	string	-	本次起降的 UUID
3	hardware_id	string	-	无人机飞控序列号
4	body_code	string	-	机身编码
5	start_end	TimeRange	-	时间范围，表示片段的起始时间和结束时间
6	data	FlightData[]	-	实时航点

6.7.1 FlightData 实时航点数据

FlightData 消息用于描述飞行中的具体数据点，包括时间戳、位置、速度、高度、工作模式、喷洒信息和播撒信息。

序号	字段名	字段类型	单位	备注
1	collect_time	int64	ms	时间戳
2	position	Position	-	位置
3	velocity	Velocity	-	速度
4	height	float	m	高度，精确到小数点后两位
5	work_mode	int32	-	工作模式 0: NULL 1: M (手动) 2: AB 3: M+ (手动增强) 4: WP (航线模式)
6	spray_width	float	m	喷洒宽度，精确到小数点后两位
7	spray_info	SprayInfo	-	喷洒信息
8	sow_info	SowInfo	-	播撒信息

6.7.2 Velocity 速度信息

Velocity 消息用于描述北东地坐标系下的速度。

序号	字段名称	字段类型	单位	备注
1	x	float	m/s	x 轴速度，精确到小数点后两位
2	y	float	m/s	y 轴速度，精确到小数点后两位
3	z	float	m/s	z 轴速度，精确到小数点后两位

6.7.3 SprayInfo 喷洒信息

SprayInfo 消息用于描述喷洒信息，包括喷洒类型、流速、剩余药量。

序号	字段名称	字段类型	单位	备注
1	spray_type	int32	-	喷洒状态 0: 未喷洒 1: 喷洒中
2	flow_speed	int32	mL/min	流速
3	residual_pesticide	uint32	ml	剩余药量

6.7.4 SowInfo 播撒信息

SowInfo 消息用于描述播撒信息，包括播撒类型、播撒模式、转速、仓口大小、流量。

序号	字段名称	字段类型	单位	备注
1	sow_type	int32	-	播撒状态 0: 未工作 1: 工作中
2	sow_mode	int32	-	播撒模式 0: 无播撒 1: 自动播撒 2: 手动播撒
3	turn_speed	uint32	-	转速
4	hatch_size	uint32	百分比	仓口大小
5	sow_speed	uint32	g/min	流量

6.8 UavInfo 无人机信息

UavInfo 消息用于描述无人机飞控 SN、机身编码、无人机型号、北斗标识、厂家编号。

序号	字段名称	字段类型	单位	备注
1	hardware_id	string	-	无人机飞控 SN

序号	字段名称	字段类型	单位	备注
2	body_code	string	-	机身编码
3	uav_mode	string	-	无人机型号
4	bd_identification	string	-	北斗标识 软件北斗: R 硬件北斗: Y
5	company_code	string	-	厂家编号

7. 协议基础

7.1 通信方式

1. 通信协议采用 TCP 协议传输，数据接收平台作为服务器端，数据发送端作为客户端。
2. 数据发送端通过数据接收平台的 IP 和端口号，建立连接并进行数据传输。
3. 数据传输前需要通过 key 和 secret 进行身份验证。
4. 验证通过后，客户端进行数据传输。

7.2 通信链接

数据通信链路正常时，客户端将信息按照本协议规定的格式上传至服务器。其中“飞行记录”、“飞行记录明细”不需要平台应答，“登录信息”需要平台应答。

7.3 通信 IP 和端口号

IP: 111.202.113.106

端口号: 29325

7.4 无人机鉴权

在开发者平台 (<http://dev-iot.dtwl360.com>) “账号设置”菜单“企业信息”中获取登录信息使用的 key 和 secret。在“无人机终端管理菜单”添加无人机飞控序列号。

1. 获取 key 和 secret

首页 / 帐号设置 / 帐号设置

帐号设置

The screenshot shows the '帐号设置' (Account Settings) page. On the left, there is a sidebar with '企业信息' (Company Information) and '修改密码' (Change Password). The main content area has a form with the following fields:

- * 厂家名称: [Input field]
- 厂家编号: 十进制: [Input field] 十六进制: [Input field]
- 北斗无人机数据传输: key: [Input field]
- secret: [Input field]

The 'key:' and 'secret:' input fields are highlighted with red boxes.

2. 添加无人机飞控序列号

在开发者平台“北斗无人机”、“无人机终端管理”添加无人机飞控序列号，添加飞控序列号之后进行数据对接。