

# TinyFrame\_Interface

Fall\_detection 跌倒检测

# 目录

3.1. 通信链路介绍 .....	1
3.2. 通信信息示例 .....	1
消息类型：查询固件状态 (TYPE:0xFFFF) .....	1
消息类型：返回固件状态 (TYPE:0xFFFF) .....	2
消息类型：上传目标信息 (TYPE:0x0A08) .....	3
4. 协议约定说明.....	4
4.1. 协议结构框图 .....	4
4.2. 协议结构简介 .....	4
4.3. TF 帧解析流程.....	5
4.4. TF 示例帧解析详解 .....	6
5. 通用消息类型 .....	10
5.1. 消息类型：查询固件状态 (TYPE:0xFFFF) .....	10
5.2. 消息类型：返回固件状态 (TYPE:0xFFFF) .....	10
5.3. 消息类型：进入 OTA 升级 (TYPE:0x3000) .....	11
5.4. 消息类型：OTA 升级 (TYPE:0x3000) .....	11
7. 跌倒检测项目 .....	12
消息类型：报告跌倒检测项目测试结果 0x0E02.....	12
消息类型：设置安装高度 0x0E04.....	12
消息类型：设置安装高度后回传结果 0x0E04.....	13
消息类型：获取雷达参数 0x0E06.....	13
消息类型：获取雷达参数后回传结果 0x0E06.....	14
消息类型：设置跌倒阈值 0x0E08.....	15
消息类型：设置跌倒阈值后回传结果 0x0E08.....	15
消息类型：设置跌倒灵敏度 0x0E0A.....	16
消息类型：设置跌倒灵敏度后回传结果 0x0E0A.....	16
消息类型：高度上传结果 0x0E0E.....	16
消息类型：打开或关闭 User log 信息 0x010E .....	17
消息类型：设置报警区域参数 0x0E0C.....	17

消息类型：设置报警区域参数回传结果 0x0E0C.....	18
消息类型：雷达初始化设置参数 0x2110.....	19
消息类型：报告 3D 点云检测项目测试结果 0x0A08.....	19
消息类型：报告有无人检测项目测试结果 0x0F09.....	20
<b>8. 编程接口 .....</b>	<b>22</b>
<b>8.1. 编码 TF 消息.....</b>	<b>22</b>
<b>8.2. 解码 TF 消息.....</b>	<b>22</b>
<b>8.3. 示例代码 .....</b>	<b>22</b>

## 1. 目的

本文档定义雷达模组通信协议。

## 2. 范围

本文档适用于项目开发人员、维护人员、测试人员和客户。

## 3. 数据通信链路约定

### 3.1. 通信链路介绍

服务端是指模组端，客户端是指上位机端

服务端的通信方式：主动上传和被动上传

客户端的通信方式：主动下发

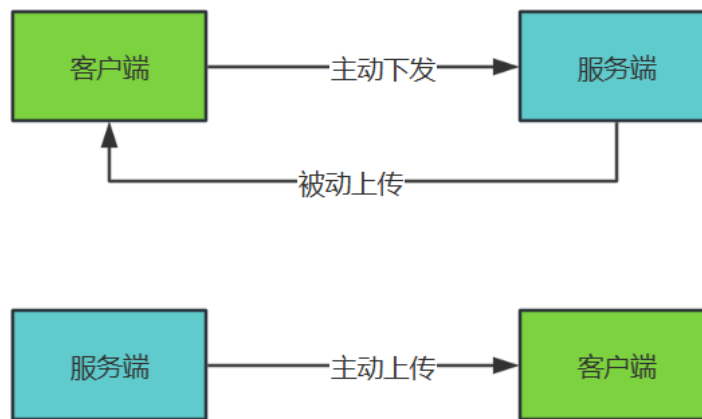


图 1: 通讯方式

### 3.2. 通信信息示例

消息类型：查询固件状态 (TYPE:0xFFFF)

通信方式:	主动下发		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和

示例 TF 帧: 01 00 01 00 00 FF FF FF

消息类型：返回固件状态 (TYPE:0xFFFF)

通信方式：		被动上传	
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和
DATA (见下文)			
DATA_CKSUM	1 byte	uint8	数据校验和

以下是 DATA 位对应的含义：

DATA			
字节数	基本类型	帧结构	帧含义
1 byte	uint8	[project_name]	表示雷达所运行的项目 0：表示存在感知项目 1：表示呼吸检测项目。 2：表示手势检测项目。 3：表示测距项目。 4：表示人员计数项目。 5：表示 3D 点云检测项目。
1 byte	uint8	[major_version]	表示雷达主版本号
1 byte	uint8	[sub_version]	表示雷达子版本号
1 byte	uint8	[modified_version]	表示雷达修订版本号

示例 TF 帧：01 04 AF 00 04 FF FF 51 06 03 0A 00 F0

消息类型：上传目标信息 (TYPE:0x0A08)

通信方式：	主动上传		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和
DATA (见下文)			
DATA_CKSUM	1 byte	uint8	数据校验和

以下是 DATA 位对应的含义：

DATA			
字节数	基本类型	帧结构	帧含义
4 byte	int32	[target_num]	目标个数
4 byte	float	[x]	x 坐标, 数据类型 float, 单位: 米(m)
4 byte	float	[y]	y 坐标, 数据类型 float, 单位: 米(m)
4 byte	float	[z]	z 坐标, 数据类型 float, 单位: 米(m)
4 byte	int32	[dop_idx]	数据类型 int32, 速度 dop_idx
4 byte	int32	[cluster_id]	数据类型 int32, 聚类目标 ID
...	...	...	

注：当目标存在 N 个的时候，x、y、z、dop\_idx、cluster\_id 也存在 N 个。

示例 TF 帧：01 00 06 00 04 0A 08 FE 00 00 00 00 FF

## 4. 协议约定说明

### 4.1. 协议结构框图

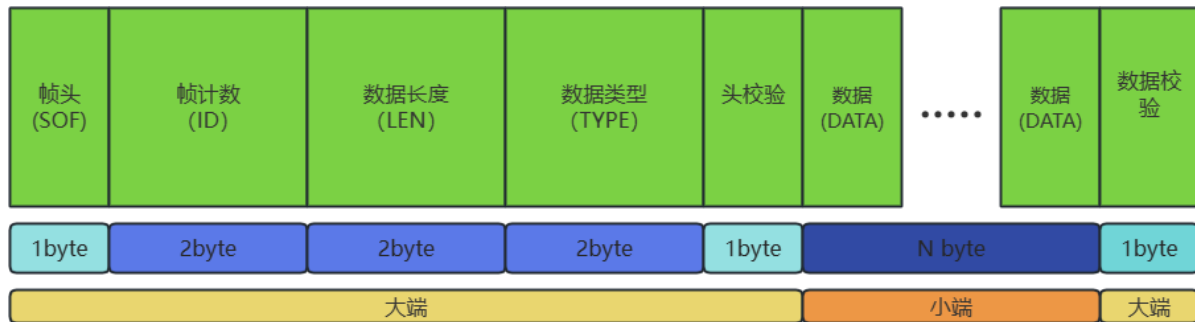


图 2：协议结构

### 4.2. 协议结构简介

格式	字节数	存储方式	格式含义
SOF	1	大端序	代表一帧 TF 数据的起始位，通常固定为 0x01。
ID	2	大端序	代表发送的包序，可用于检测客户端和服务端数据帧是否连续。
LEN	2	大端序	程序中默认设置总数据帧长度不超过 1024 个字节。
TYPE	2	大端序	代表整个数据帧的消息类型，不同的消息类型所代表的功能不同。
HEAD_CKSUM	1	大端序	代表头帧校验和，从 SOF 位到 TYPE 位先全部进行异或运算，再取反。
DATA	N	小端序	代表数据的有效载荷，DATA 位数据长度与 LEN 位相关。例如 LEN 位值为 2，则 DATA 位由两个字节数据组成。
DATA_CKSUM	1	大端序	代表尾帧校验和，所有 DATA 位先全部进行异或运算，再取反。

注：SOF 位~HEAD\_CKSUM 位与 DATA\_CKSUM 采用大端排序原因，方便查看指令的 TYPE 类型以及数据长度，DATA 位采用小端的排序，符合计算器存储顺序，方便解析 DATA 数据（可以强行转换数据类型）。

### 4.3. TF 帧解析流程

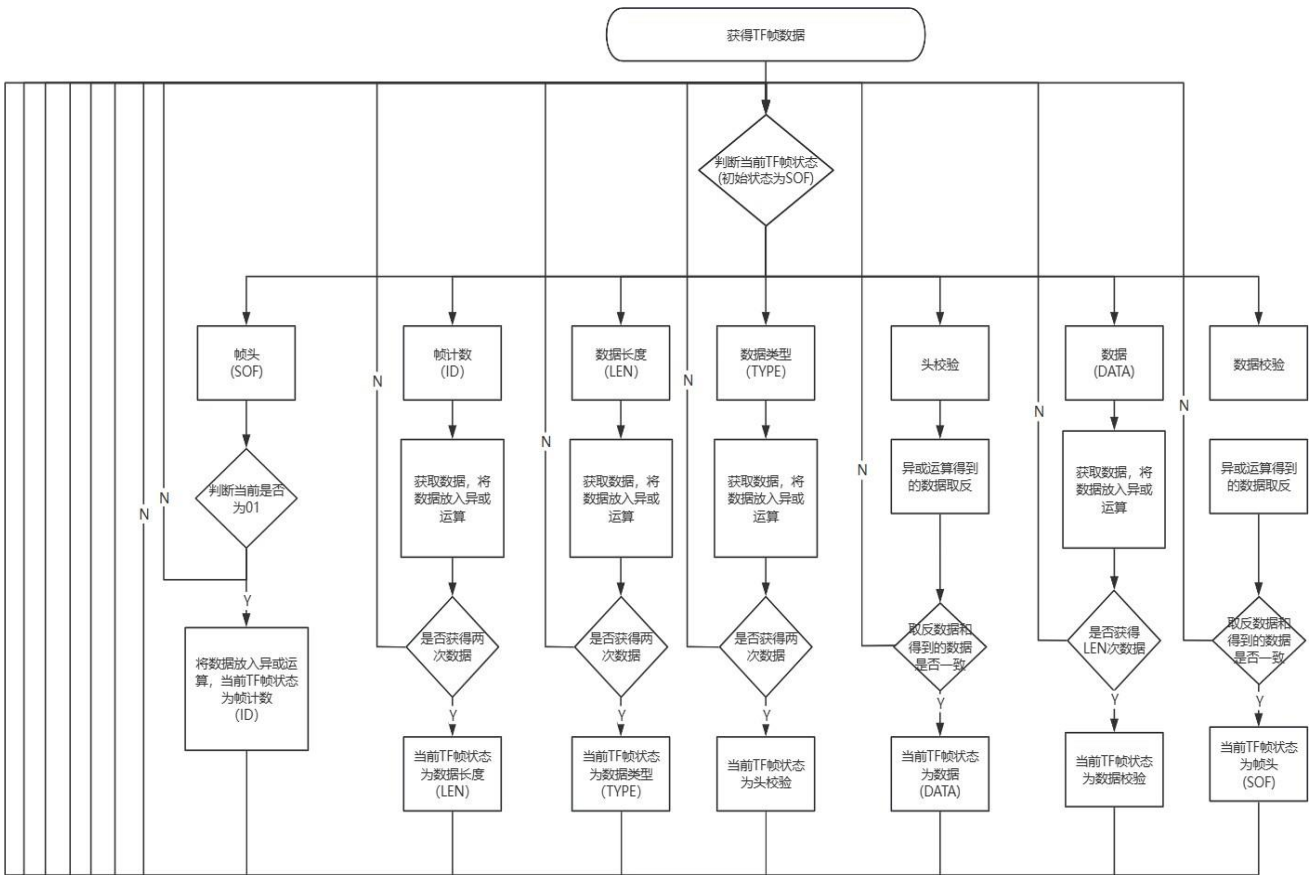
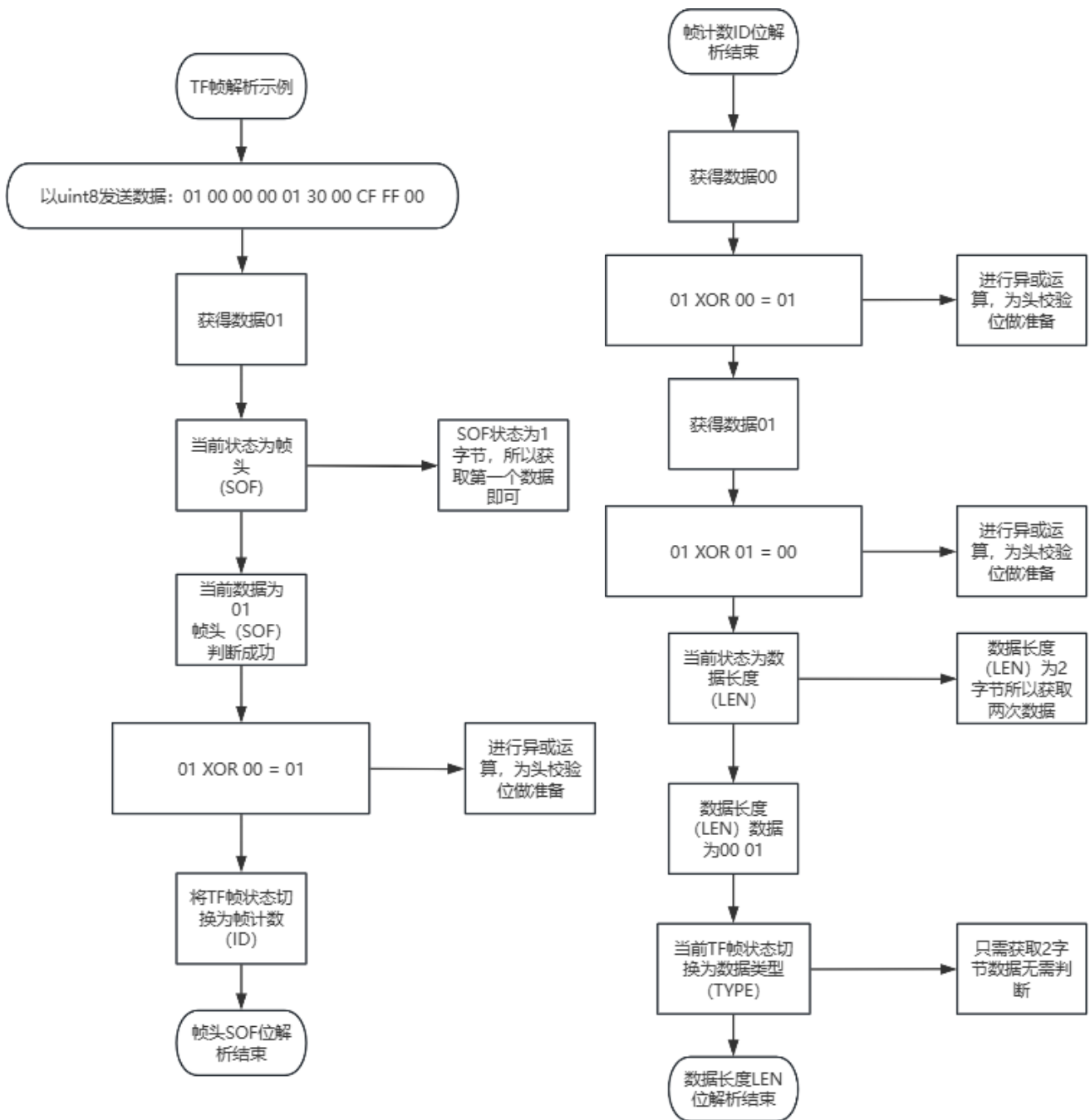
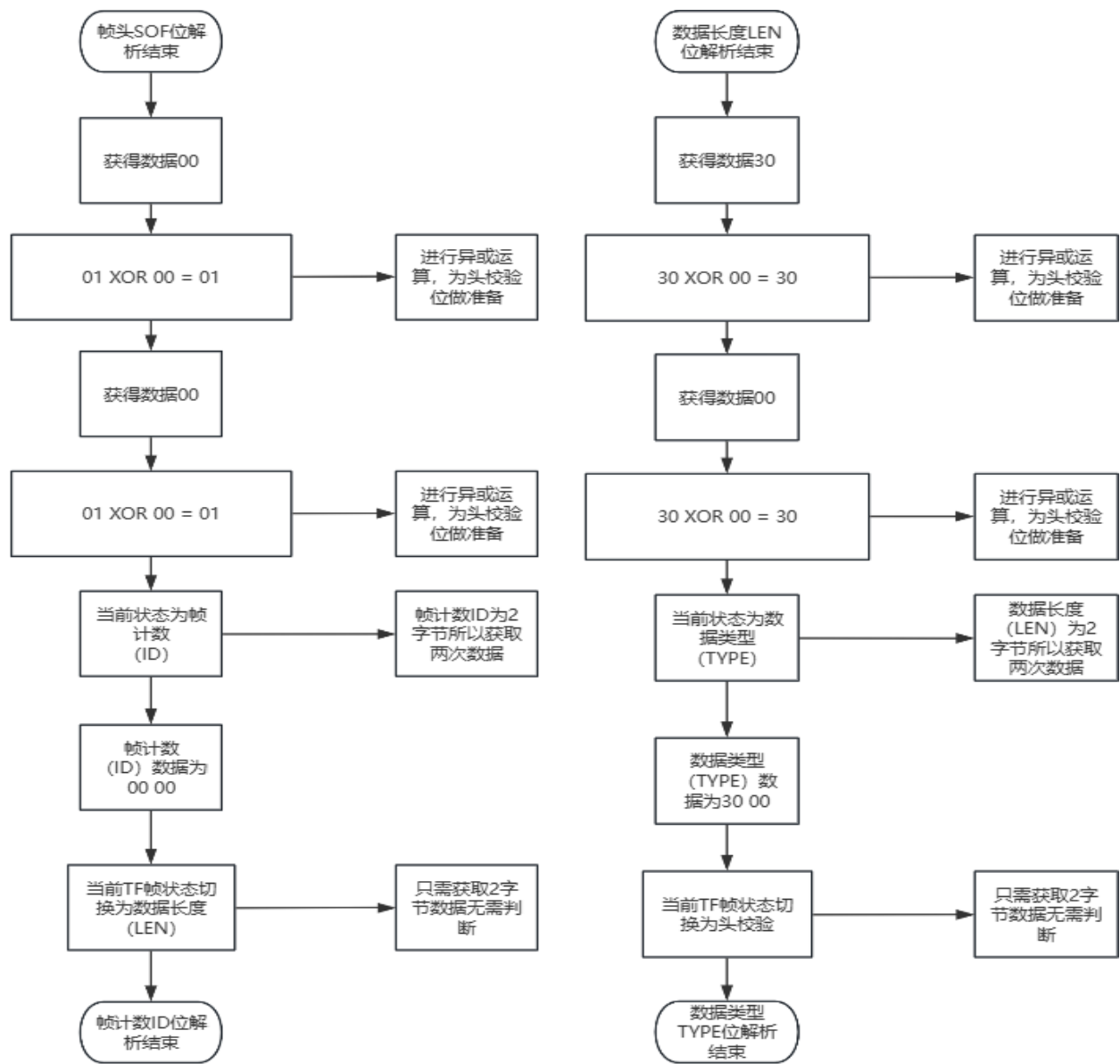


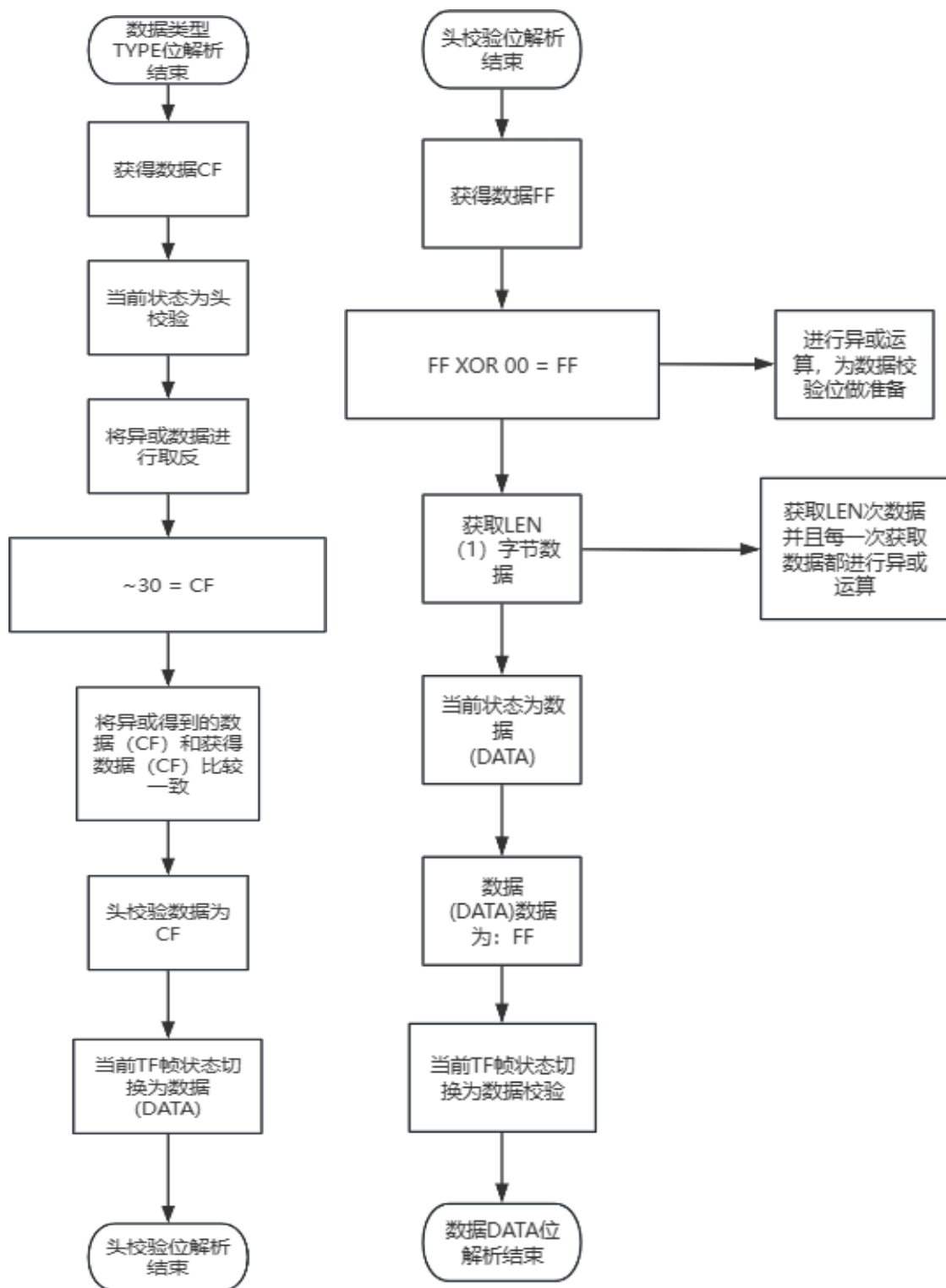
图 3: TF 帧解析流程



#### 4.4. TF 示例帧解析详解







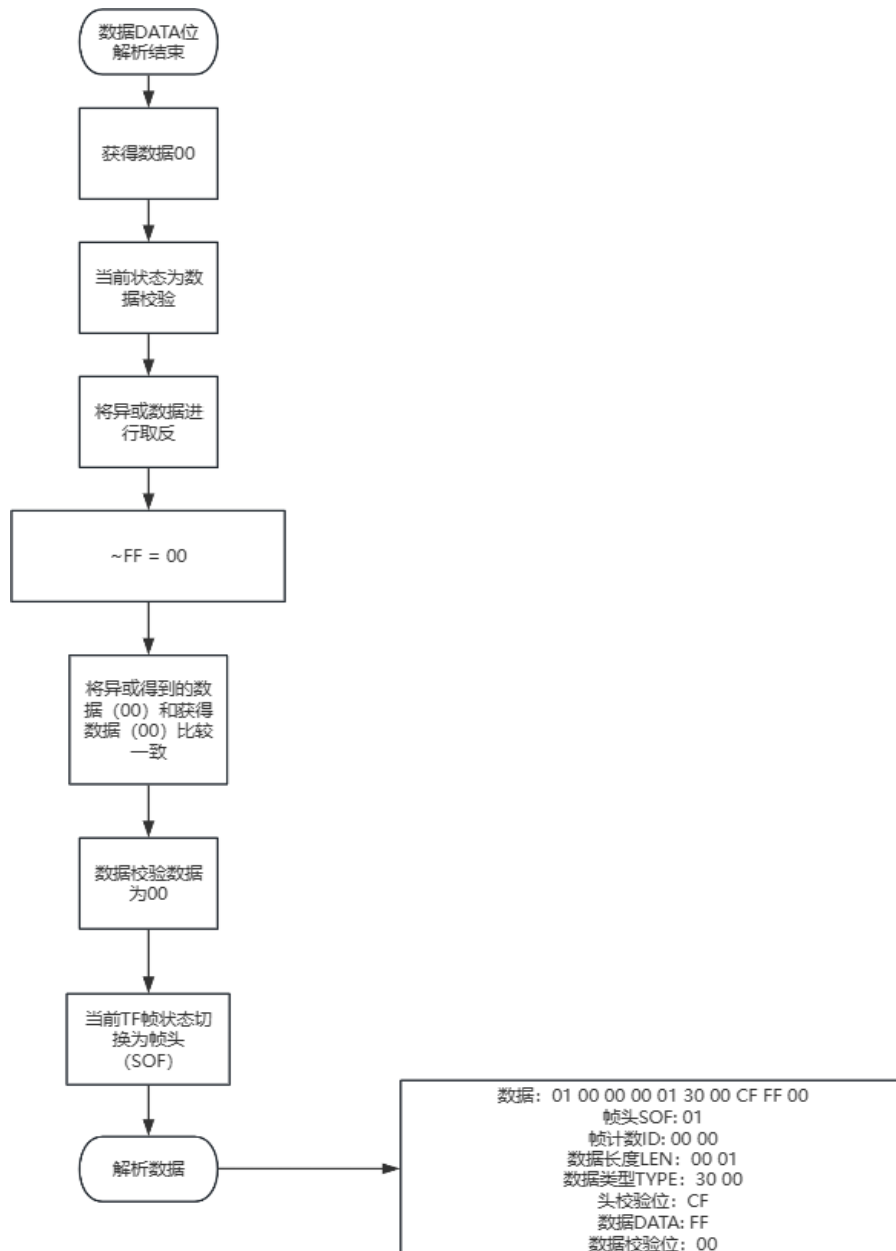


图 4: TF 示例帧解析详解

## 5. 通用消息类型

### 5.1. 消息类型：查询固件状态 (TYPE:0xFFFF)

通信方式:	主动下发		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和

示例 TF 帧: 01 00 01 00 00 FF FF FF

### 5.2. 消息类型：返回固件状态 (TYPE:0xFFFF)

通信方式:	被动上传		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和
DATA (见下文)			
DATA_CKSUM	1 byte	uint8	数据校验和

以下是 DATA 位对应的含义:

DATA			
字节数	基本类型	帧结构	帧含义
1 byte	uint8	[project_name]	表示雷达所运行的项目 0: 表示存在感知项目 1: 表示呼吸检测项目。 2: 表示手势检测项目。 3: 表示测距项目。 4: 表示人员计数项目。 5: 表示 3D 点云检测项目。
1 byte	uint8	[major_version]	表示雷达主版本号
1 byte	uint8	[sub_version]	表示雷达子版本号
1 byte	uint8	[modified_version]	表示雷达修订版本号

示例 TF 帧: 01 04 AF 00 04 FF FF 51 06 03 0A 00 F0

### 5.3. 消息类型：进入 OTA 升级 (TYPE:0x3000)

通信方式：	主动下发		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和

示例 TF 帧：01 00 01 00 00 30 00 CF

### 5.4. 消息类型：OTA 升级 (TYPE:0x3000)

通信方式：	被动上传		
格式	字节数	基本类型	帧结构
SOF	1 byte	uint8	起始帧
ID	2 byte	uint16	帧 ID
LEN	2 byte	uint16	数据帧长度
TYPE	2 byte	uint16	帧类型
HEAD_CKSUM	1 byte	uint8	头校验和
DATA (见下文)			
DATA_CKSUM	1 byte	uint8	数据校验和

以下是 DATA 位对应的含义：

DATA			
字节数	基本类型	帧结构	帧含义
1 byte	uint8	[tmpCodeInfo]	FF:待升级

示例 TF 帧：01 00 03 00 01 30 00 CC FF 00

## 7. 跌倒检测项目

本项目为跌倒项目，数据通过 UART 接口传输，波特率为 115200，数据字长 8 位，停止位 1 位，无奇偶校验位，无硬件流控制。功能实现流程：检测人体高度信息离地面持续 30 帧 \* 灵敏度的总帧数（1s 20 帧）持续低于跌倒阈值高度信息，上报跌倒。

**消息类型：报告跌倒检测项目测试结果 0x0E02**

消息类型为 0x0E02，仅支持单向数据传输模式。

雷达发送数据给上位机：MSG_IND_FALL_STATUS					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	用于报告跌倒测试结果。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 02	
HEAD_CKSUM	1 byte	uint8	头校验和	F3	
DATA	1 byte	uint8	[is_fall]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

以下是每个 DATA 位对应的含义：

- [is\_fall]: 判断是否跌倒。
  - 值为 0: 正常。
  - 值为 1: 跌倒。

**消息类型：设置安装高度 0x0E04**

消息类型为 0x0E04，支持双向数据传输模式。

上位机发送数据到雷达：MSG_IND_FALL_SET_HIGH					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	用于设置雷达安装高度。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 04	
TYPE	2 byte	uint16	帧类型	0E 04	
HEAD_CKSUM	1 byte	uint8	头校验和	F0	
DATA	4 byte	float	[high]	00 00 20 40	
DATA_CKSUM	1 byte	uint8	数据校验和	9F	

以下是每个 DATA 位对应的含义：

- [high]：设置雷达安装高度，范围为 1~5m。

转换成 float：例如[x\_point]位为 0x66、0x66、0xA2、0x41，先拼成 uint32 位整形，由于 TF 帧 Data 位小端序，所以值为 0x41A26666，然后进行 float 类型强转，最终结果为：20.3。

```

1. int main(void)
2. {
3.     unsigned int param = 0x41A26666;
4.     float res = *(float *)&param;
5.
6.     printf("data: %f\n", res);
7.     return 0;
8. }
    
```

消息类型：设置安装高度后回传结果 0x0E04

雷达回传数据给上位机：MSG_IND_FALL_RES					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	用于设置雷达安装高度后，回传状态。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 04	
HEAD_CKSUM	1 byte	uint8	头校验和	F5	
DATA	1 byte	uint8	[result]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

当雷达收到 TYPE 为 0x0E04 时，雷达将会回传数据。[result]有两种结果：

- 值为 0：设置高度失败。
- 值为 1：设置高度成功。

消息类型：获取雷达参数 0x0E06

消息类型为 0x0E06，支持双向数据传输模式。



上位机发送数据到雷达：MSG_IND_FALL_GET_HIGH					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	获取下位机参数。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 00	
TYPE	2 byte	uint16	帧类型	0E 06	
HEAD_CKSU M	1 byte	uint8	头校验和	F6	

上位机下发命令给雷达，请求获取下位机的数据，雷达会回复类型为 0x0E06 的信号，详细介绍见下

消息类型：获取雷达参数后回传结果 0x0E06

雷达回传数据给上位机：MSG_IND_FALL_GET_HIGH					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	向上位机发送雷达的参数。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 1C	
TYPE	2 byte	uint16	帧类型	0E 06	
HEAD_CKSU M	1 byte	uint8	头校验和	EA	
DATA	4 byte	float	[high]	/	
DATA	4 byte	float	[threshold]	/	
DATA	4 byte	uint32	[sensitivity]	/	
DATA	4 byte	float	[rect_XL]	/	
DATA	4 byte	float	[rect_XR]	/	
DATA	4 byte	float	[rect_ZF]	/	
DATA	4 byte	float	[rect_ZB]	/	
DATA_CKSU M	1 byte	uint8	数据校验和	/	

当雷达收到 TYPE 为 0x0E06 时，雷达将会回传数据。有两种结果：

- 值为 0：获取失败。
- 值为其它：获取成功。

消息类型：设置跌倒阈值 0x0E08

消息类型为 0x0E08，支持双向数据传输模式。

上位机发送数据到雷达：MSG_IND_FALL_SET_THRESHOLD					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	设置跌倒阈值，雷达默认跌倒阈值为 0.6m。地面处高度为 0m
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 04	
TYPE	2 byte	uint16	帧类型	0E 08	
HEAD_CKSUM	1 byte	uint8	头校验和	FC	
DATA	4 byte	float	[set_threshold]	9A 99 19 3F	
DATA_CKSUM	1 byte	uint8	数据校验和	DA	

转换成 float：例如[high]位为 0x9A、0x99、0x19、0x3F，先拼成 uint32 位整形，由于 TF 帧 Data 位小端序，所以值为 0x3F19999A，然后进行 float 类型强转，最终结果为：0.6。

```

1. int main(void)
2. {
3.     unsigned int param = 0x41A26666;
4.     float res = *(float *)&param;
5.
6.     printf("data: %f\n", res);
7.     return 0;
8. }
    
```

消息类型：设置跌倒阈值后回传结果 0x0E08

雷达回传数据给上位机：MSG_IND_FALL_SET_THRESHOLD					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	设置跌倒阈值后，回传状态。
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 08	
HEAD_CKSUM	1 byte	uint8	头校验和	F9	
DATA	1 byte	uint8	[value]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

当雷达收到 TYPE 为 0x0E08 时，雷达将会回传数据。回传的数据有两种结果：

- 值为 0x00：获取失败。
- 值为 0x01：获取成功。

**消息类型：设置跌倒灵敏度 0x0E0A**

消息类型为 0x0E0A，支持双向数据传输模式。

上位机发送数据到雷达：MSG_IND_FALL_SET_SENSITIVITY					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	设置跌倒灵敏度
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 0A	
HEAD_CKSUM	1 byte	uint8	头校验和	FB	
DATA	4 byte	uint32	[high]	03 00 00 00	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

设置跌倒灵敏度，初始值 10.(范围 3~30)

**消息类型：设置跌倒灵敏度后回传结果 0x0E0A**

雷达回传数据给上位机：MSG_IND_FALL_SET_SENSITIVITY					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	当雷达收到 TYPE 为 0x0E0A 时，雷达将会回传数据。回传的数据有两种结果： 值为 0：获取失败。 值为 1：获取成功。
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 0A	
HEAD_CKSUM	1 byte	uint8	头校验和	FB	
DATA	1 byte	uint8	[value]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

**消息类型：高度上传结果 0x0E0E**

雷达回传数据给上位机：MSG_IND_FALL_SET_SENSITIVITY					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	雷达自动上传当前目标点的高度信息（雷达位置高度为 0m）
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0E 0E	
HEAD_CKSUM	1 byte	uint8	头校验和	FF	
DATA	1 byte	Uint32	[value]	01 00 00 00	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

**消息类型：打开或关闭 User log 信息 0x010E**

消息类型为 0x010E，只支持单向数据传输模式

上位机发送数据到雷达					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	0: 关闭 User log 信息 1: 打开 User log 信息
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 04	
TYPE	2 byte	uint16	帧类型	01 0E	
HEAD_CKSUM	1 byte	uint8	头校验和	F5	
DATA	4 byte	uint32	[value]	01 00 00 00	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

- 值为 0x00：关闭 User log 信息。
- 值为 0x01：打开 User log 信息。

**消息类型：设置报警区域参数 0x0E0C**

消息类型为 0x0E0C，支持双向数据传输模式

上位机发送数据到雷达					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	设置跌倒报警区域范围，设置范围 0.3~1.5
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 10	
TYPE	2 byte	uint16	帧类型	0E 0C	
HEAD_CKSUM	1 byte	uint8	头校验和	EC	
DATA	4 byte	float	[rect_XL]	00 00 00 3F	
DATA	4 byte	float	[rect_XR]	00 00 00 3F	
DATA	4 byte	float	[rect_ZF]	00 00 00 3F	
DATA	4 byte	float	[rect_ZB]	00 00 00 3F	
DATA_CKSUM	1 byte	uint8	数据校验和	FF	

设置跌倒报警区域范围，设置范围 0.3~1.5

消息类型：设置报警区域参数回传结果 0x0E0C

雷达发送数据到上位机					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	0: 失败 1: 成功
ID	2 byte	uint16	帧 ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 04	
TYPE	2 byte	uint16	帧类型	0E 0C	
HEAD_CKSUM	1 byte	uint8	头校验和	F8	
DATA	1 byte	uint8	[value]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

- 值为 0x00：获取失败。
- 值为 0x01：获取成功。

### 消息类型：雷达初始化设置参数 0x2110

消息类型为 0x2110，支持单向数据传输模式

上位机发送数据给雷达					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	雷达初始化设置参数 High:2.4 Threshold:0.6 Sensitivity:10 Rect_XL:1.5 Rect_XR:1.5 Rect_ZF:1.5 Rect_ZB:1.5
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 00	
TYPE	2 byte	uint16	帧类型	2110	
HEAD_CKSUM	1 byte	uint8	头校验和	CF	

当收到TYPE为0x2110时，雷达初始化设置参数。

High:2.4

Threshold:0.6

Sensitivity:10

Rect\_XL:1.5; Rect\_XR:1.5; Rect\_ZF:1.5; Rect\_ZB:1.5;

设置的内容和TYPE类型0x0E06的内容相关

### 消息类型：报告 3D 点云检测项目测试结果 0x0A08

消息类型为 0x0A08，仅支持单向数据传输模式(开启 User log 后自动上传)。

雷达发送数据给上位机：MSG_IND_3D_CLOUD_RES					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	用于报告 3D 点云检测结果。其中消息类型为 0x0A08 是点云信息
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	\	
TYPE	2 byte	uint16	帧类型	0A 08	
HEAD_CKSUM	1 byte	uint8	头校验和	\	
DATA	4 byte	int32	[target_num]	\	
DATA	4 byte	int32	[cluster_index]	\	
DATA	4 byte	float	[x_point]	\	
DATA	4 byte	float	[y_point]	\	
DATA	4 byte	float	[z_point]	\	
DATA	4 byte	float	[speed]	\	
DATA_CKSUM	1 byte	uint8	数据校验和	\	

以下是每个 DATA 位对应的含义：

- [target\_num]: 目标个数。
- [cluster\_index]: 聚类目标 ID。
- [x\_point]: x 坐标，单位：m。
- [y\_point]: y 坐标，单位：m。
- [z\_point]: z 坐标，单位：m。
- [z\_point]: z 坐标，单位：m。
- [speed]: 速度，单位：m/s。

当点云存在 N 个的时候，x、y、z、dop\_idx 也存在 N 个，cluster\_id 距离相近点云 ID 相同，示例：

```

01 41 A1 00 68 0A 08 74 05 00 00 00 00 00 00 00 62 F6 C2 BD 38 24 F7 3E E6 A8 41 3E 39 63 18 BC 00
起始帧 帧ID 数据帧长度 帧长度 头校验和 点云个数 点云 ID x0坐标 y0坐标 z0坐标 速度V0
00 00 00 D5 94 A5 BD 14 9E 18 3F 69 2C FA 3C 39 63 18 BC 00 00 00 00 73 5C 2F BE 3C 4F F8 3E EE EE
点云 ID x1坐标 y1坐标 z1坐标 速度V1 点云 ID x2坐标 y2坐标 z2坐标
E1 3E 39 63 18 BC 00 00 00 00 BF 56 CA BD F8 E4 EE 3E 39 CC 65 3E 39 63 18 3C 00 00 00 00 48 0B A3
速度V2 点云 ID x3坐标 y3坐标 z3坐标 速度V3 点云 ID x4坐标
BD 37 92 18 3F E0 2F 16 3D 39 63 18 3C 6F
y4坐标 z4坐标 速度 V4 数据校验和

```

消息类型：报告有无人检测项目测试结果 0x0F09

消息类型为 0x0F09，仅支持单向数据传输模式。

雷达发送数据给上位机：MSG_IND_FALL_STATUS					
格式	字节数	基本类型	帧结构	示例帧	帧含义
SOF	1 byte	uint8	起始帧	01	用于报告有无人测试结果。
ID	2 byte	uint16	帧ID	00 00	
LEN	2 byte	uint16	数据帧长度	00 01	
TYPE	2 byte	uint16	帧类型	0F 09	
HEAD_CKSUM	1 byte	uint8	头校验和	F9	
DATA	1 byte	uint8	[is_human]	01	
DATA_CKSUM	1 byte	uint8	数据校验和	FE	

以下是每个 DATA 位对应的含义：

- [is\_human]：判断是否有人。
  - 值为 0：无人。
  - 值为 1：有人。



## 8. 编程接口

### 8.1. 编码 TF 消息

```
void tinyFrameTx(TF_TYPE type, uint8 *data, TF_LEN len);
```

其中 type 为发送数据类型，uint16 类型，例如人员检测数据结果上报，数据类型为 0x0A10。见 4.2.1.6 详述。

uint8\* data 是发送数据的地址。

len 为发送数据的长度，uint16 类型。

### 8.2. 解码 TF 消息

```
TinyFrameRx tinyFrameRx(void);
```

成功接收消息后，接收的数据返回到一个 TinyFrameRx 类型的变量。

其中每个 CKSUM 的解析如下所示：

HEAD\_CKSUM：TF 帧头校验和 【从第一个字节开始到 HEAD\_CKSUM 位的上一个字节】

DATA\_CKSUM：TF 数据校验和 【DATA 的第一个到 DATA\_CKSUM 位的上一个字节】其

中计算 CKSUM 的方法 c 代码如下所示：

```
1. unsigned char getCksum(unsigned char *data, unsigned char len)
2. {
3.     unsigned char ret = 0;
4.
5.     for (int i = 0; i < len; i++)
6.         ret = ret ^ data[i];
7.
8.     ret = ~ret;
9.
10.    return ret;
11. }
12.
```

### 8.3. 示例代码

如果想要解析 TF 帧数据的 demo（包含 Linux 环境与 Keil  $\mu$ Vision5 环境下的 C 语言 demo、Python 语言 demo），可以直接与销售沟通获得。