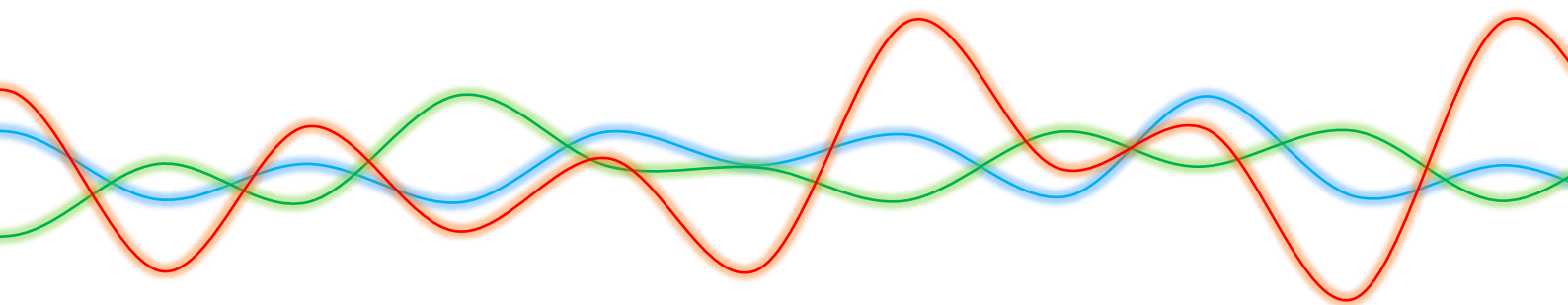



Yesense 通讯协议说明文档


Document Version 1.3, 15 May. 2019



Yesense Technologies Co., Ltd.

 027-87003846

 contact@yesense.com

 www.yesense.com

目 录

1	文档概述	2
2	协议介绍	2
2.1	串口输出协议	2
2.1.1	协议帧格式	2
2.1.2	有效数据定义规则	3
2.1.3	校验和计算方法	7
2.2	串口输入（设置）协议	7
2.2.1	协议帧格式	7
2.2.2	有效数据定义规则	8
2.2.3	校验和计算方法	11
2.3	SPI 输出协议	12
3	协议参考	15
3.1	协议默认配置	15
3.1.1	通信接口默认配置	15
3.1.2	输出协议默认设置	15
3.1.3	输入（设置）默认配置参数	15
3.2	协议示例	16
3.2.1	串口输出协议示例	16
3.2.2	串口输入协议示例	17
4	Revisions	19

1 文档概述

本文档主要是描述 Yesense 私有协议，用于集成和配置 Yesense 产品的重要资源。

Yesense 私有协议规范描述了 Yesense 产品使用的通信数据格式，包括两部分输出数据帧和输入（配置）数据帧。

2 协议介绍

Yesense 私有协议设计用于与主机之间的通信。该协议有以下特性：

- 1) 紧凑——使用 8Bit 二进制数据
- 2) 数据校验——使用低开销的校验算法
- 3) 统一——所有数据都是小端模式

2.1 串口输出协议

2.1.1 协议帧格式

Yesense 私有输出协议采用固定的格式，帧结构包含五部分：帧头、时间戳、数据长度、数据、校验码，如下图 1 所示。

YS Header	TID	LEN	MESSAGE	CK1	CK2
Dec 89 83 Hex 59 53	2 Bytes Message TID	Length of Message (1 Byte) excluding YS Header, TID	Message size depend on Len filed	two bytes checksum	

图 1 Yesense 私有输出协议帧格式

完整帧格式的详细描述如下表 1 所示。

Type	LEN (Bytes)	Description
YS Header	2	YS 数据包的起始帧，0x59,0x53
TID	2	时间戳标识，最大值 60000
LEN	1	MESSAGE 的长度，最大值 255
MESSAGE	0-255	数据包的有效数据
CK1	1	校验码，计算公式见 2.1.3
CK2	1	校验码，计算公式见 2.1.3

表 1 完整帧格式描述

- 每一帧数据以两个特定字节开始：0x59 0x53。
- 接下来两个字节的 TID。TID 用来标识数据帧的序号。

- 紧着是一个字节的数据长度域。该数据长度标识有效数据（即 Payload）的长度。
- 有效数据域是一个可变的域。
- CK_1 和 CK_2 是一个 16bit 的校验和，其计算方法定义见 2.1.3。

2.1.2 有效数据定义规则

MESSAGE 的定义如下表 2 所示：

Packet 1			Packet N		
DATA ID	LEN	DATA (LEN Bytes)	DATA ID	LEN	DATA (LEN Bytes)

表 2 串口输出协议 Message 定义

MESSAGE 中的数据可由多个 Packet 组合组成。每一个不同的 Packet 包含特定的 DATA ID 以及与之对应的数据长度指示符 LEN 和数据长度为 LEN 的数据 DATA 组成。

DATA ID 与 LEN 和 DATA 的对应关系如下表 3 所示：

DATA NAME	DATA ID	LEN	DATA
IMU 温度	0x01	2	DATA1- DATA2
加速度	0x10	12	DATA1 – DATA12
角速度	0x20	12	DATA1 – DATA12
磁场归一化值	0x30	12	DATA1 – DATA12
磁场强度	0x31	12	DATA1 – DATA12
欧拉角	0x40	12	DATA1 – DATA12
四元数	0x41	16	DATA1 – DATA16
UTC 时间	0x50	11	DATA1 – DATA11
采样时间戳	0x51	4	DATA1 – DATA4
同步输出时间戳	0x52	4	DATA1 – DATA4
位置	0x60	12	DATA1 – DATA12
速度	0x70	12	DATA1 – DATA12

表 3 串口输出协议 DATA ID 与 LEN 及 DATA 的对应关系

DATA 的转换关系如下表 4 所示：

类型	数据	数据转换	单位
IMU 温度	DATA1 (DATA[7:0])	temp_imu = DATA × 0.01	°C
	DATA2 (DATA[15:8])		
加速度	DATA1 (DATA[7:0])	ax = DATA × 0.000001	m/s ²
	DATA2 (DATA[15:8])		

	DATA3 (DATA[23:16])	ay = DATA × 0.000001	
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])		
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	az = DATA × 0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
角速度	DATA1 (DATA[7:0])	wx = DATA × 0.000001	deg/s
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	wy = DATA × 0.000001	
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	wz = DATA × 0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
磁场归一化值	DATA1 (DATA[7:0])	mx = DATA × 0.000001	
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	my = DATA ×0.000001	
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	mz = DATA ×0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
磁场强度	DATA1 (DATA[7:0])	mx = DATA × 0.001	mGauss
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		

	DATA5 (DATA[7:0])	my = DATA × 0.001	
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	mz = DATA × 0.001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
欧拉角	DATA1 (DATA[7:0])	pitch = DATA × 0.000001	deg(°)
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	roll = DATA × 0.000001	
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	yaw = DATA × 0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
四元数	DATA1 (DATA[7:0])	q0 = DATA × 0.000001	
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	q1 = DATA × 0.000001	
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	q2 = DATA × 0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
	DATA13 (DATA[7:0])	q3 = DATA × 0.000001	
	DATA14 (DATA[15:8])		
	DATA15 (DATA[23:16])		
	DATA16 (DATA[31:24])		
UTC 时间	DATA1 (DATA[7:0])	iTOW = DATA	Ms
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		

	DATA4 (DATA[31:24])	Year = DATA	y
	DATA5 (DATA[7:0])		
	DATA6 (DATA[15:8])		
	DATA7 (DATA[7:0])	Month = DATA	month
	DATA8 (DATA[7:0])	Day = DATA	d
	DATA9 (DATA[7:0])	Hour = DATA	h
	DATA10 (DATA[7:0])	Min = DATA	min
	DATA11 (DATA[7:0])	Sec = DATA	s
采样时间戳	DATA1 (DATA[7:0])	Sample-timestamp = DATA	us
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
同步输出时间戳	DATA1 (DATA[7:0])	SyncOut-timestamp = DATA	us
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
位置	DATA1 (DATA[7:0])	Latitude = DATA * 0.0000001	deg
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	Longitude = DATA * 0.0000001	deg
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	Altitude = DATA * 0.001	m
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
速度	DATA1 (DATA[7:0])	Ve = DATA * 0.001	m/s
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
	DATA5 (DATA[7:0])	Vn = DATA * 0.001	m/s
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	Vu = DATA * 0.001	m/s
	DATA10 (DATA[15:8])		

	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		

表 4 串口输出协议 Message 的 DATA 的转换关系

2.1.3 校验和计算方法

完整的数据帧是需要增加校验和的，校验的计算范围从 TID 开始到 Message 的最后一个字节，如下表图 2 所示，计算公式见下示例。

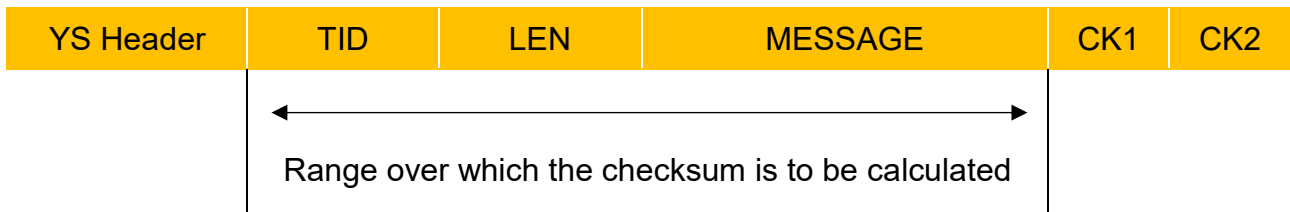


图 2 串口输出协议校验范围

假设校验范围内有 N 个字节 (buffer[N])，计算公式如下：

```

CK1 = 0; CK2 = 0;
For(i=0;i<N;i++)
{
    CK1 = CK1 + buffer[i];
    CK2 = CK2 + CK1;
}

```

2.2 串口输入（设置）协议

2.2.1 协议帧格式

Yesense 串口的输入协议的帧格式如图 3 所示。

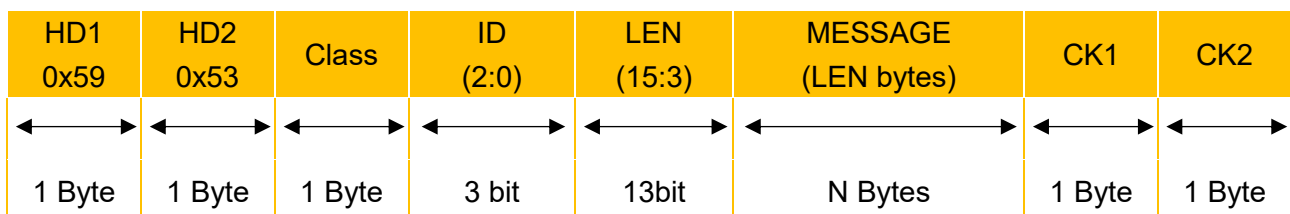


图 3 串口输入协议帧格式

- 每一帧数据以两个特定字节开始：0x59 0x53。
- 接下来为一个字节的 Class 域。Class 用来标识需要操作的对象类型。
- 紧着是两个字节的组合域。该域中的 ID 为 3Bit，表示对 Class 对象的操作；LEN 为

13Bit，表示 Message 的长度。

- CK1 和 CK2 是一个 16bit 的校验和，其计算方法定义见 2.2.3。

ID 定义如下表 5 所示：

ID	说明
0x00	查询状态
0x01	配置到 RAM，即掉电不保存
0x02	配置到 flash，即掉电保存

表 5 串口输入协议 ID 定义

Class 定义如下表 6 所示：

Class 类型	Class 数值
产品信息	0x00
UART 波特率	0x02
输出频率	0x03
输出内容	0x04
校准参数	0x05
模式设置	0x4D

表 6 串口输入协议 Class 定义

2.2.2 有效数据定义规则

各 CLASS 对应的 Message 定义如下表 7 所示：

Class 数值	Type 数值	说明
0x00	0x01	64 位产品 ID 信息
	0x02	软件版本信息
0x01	0x02	实时复位航向角，掉电不保存
0x02	0x01	设置波特率 9600 bps
	0x02	设置波特率 38400 bps
	0x03	设置波特率 115200 bps
	0x04	设置波特率 460800 bps
	0x05	设置波特率 921600 bps
0x03	0x01	设置输出频率 1Hz
	0x02	设置输出频率 2Hz
	0x03	设置输出频率 5Hz
	0x04	设置输出频率 10Hz

	0x05		设置输出频率 20Hz
	0x06		设置输出频率 25Hz
	0x07		设置输出频率 50Hz
	0x08		设置输出频率 100Hz
0x04	Bit15 - 8		保留
	Bit7	1	加速度输出开启
		0	加速度输出关闭
	Bit6	1	角速度输出开启
		0	角速度输出关闭
	Bit5	1	磁场强度输出开启
		0	磁场强度输出关闭
	Bit4	1	欧拉角输出开启
		0	欧拉角输出关闭
	Bit3	1	四元数输出开启
		0	四元数输出关闭
	Bit2	1	保留
		0	
	Bit1	1	位置输出开启
		0	位置输出关闭
	Bit0	1	速度输出开启
		0	速度输出关闭
0x05	0x03		陀螺仪零偏
	0x07		磁力计校准参数
	0x11		姿态角（俯仰角、横滚角）重置信息
	0x12		航向角重置信息
0x4D	0x02		算法模式（AHRS/VRU/General Pos/Automotive）
	0x4F		SYNC OUT 模式
	0x50		陀螺零偏初始化

表 7 串口输入协议 Message 定义

各 Class 对应的 Message 说明如下表 8 所示。

Class	输入（主机→设备）	输出（设备→主机）
0x00	查询指令，message 长度为 1，定义见上一表格	即查询指令时，根据查询的类型返回不同的长度；
0x01	message 长度为 1	无
0x02	修改波特率时，message 长度为 1，定义见上一表格；	无
0x03	修改输出频率时，message 长度为 1，定义见上一表格；	无
0x04	修改输出格式时，message 长度为 2，定义见上一表格；	无
0x05	修改陀螺零偏/磁校准参数时，数据格式为 TLV（可以同时发多个），即 type(1B) + len(1B) + data(根据参数类型)，type 定义见上一表格；	无
0x4D	修改模式时，数据格式 type(1B) + data(1B) ，（可以同时发多个）type 定义见上一表格，data 定义见下一表格； 查询指令时，message 长度为 1，定义见上一表格；	即查询指令时，message 长度 根据查询指令的 type 而定 ，定义见下一表格；

表 8 串口输入协议 Message 说明

其中 Class 对应的 message 定义如下表 9 所示。

Class	Message 中的 type	Message 中 data 的定义	备注
0x00	0x01（产品 ID）	无	8 字节
	0x02（软件版本信息）	产品型号 (YISxxxxxx) + '-' + 分支版本 (-A/-E/-N) + '-' + 软件版本 (Vxx.yy.zzz) + ';' + 预留	30 字节
0x01	0x02（实时置零航向角）	无	
0x05	0x03（陀螺零偏）	Bias_g_x + Bias_g_y + Bias_g_z	每个数据为 4

	0x07 (磁力计校准参数)	$Sc_m_x1 + Sc_m_x2 + Sc_m_x3 + Sc_m_y1 + Sc_m_y2 + Sc_m_y3 + Sc_m_z1 + Sc_m_z2 + Sc_m_z3 + Bias_m_x + Bias_m_y + Bias_m_z$	字节 int 型, 实际大小为整数值 * 0.000001
	0x11 (姿态角 (俯仰角、横滚角) 重置信息, 掉电保存)	Pitch (2B) + Roll (2B)	实际值 = 数值 * 0.01
	0x12 (航向角重置信息, 掉电保存)	Yaw (2B)	实际值 = 数值 * 0.01
0x4D	0x02(算法模式)	0x01: AHRS 模式	1 字节
		0x02: VRU 模式	
		0x03: IMU 模式	
		0x04: GeneralPos 模式	
		0x05: Automotive 模式	
	0x4F(SYNC OUT 模式)	0x01: Data Ready	1 字节
		0x02: PPS	
	0x50 (陀螺零偏初始化)	0x01: 设备自采集	1 字节

表 9 串口输入协议特殊说明

注: 所有的设置命令的返回值都只有一个字节, 即结果表示码, 0x00 (成功) 或 0xff (失败)。设置串口波特率是没有返回结果的。

2.2.3 校验和计算方法

完整的数据帧是需要增加校验和的, 校验的计算范围从 Class 开始到 Message 的最后一个字节, 如下表图 4 所示, 计算公式见下示例。

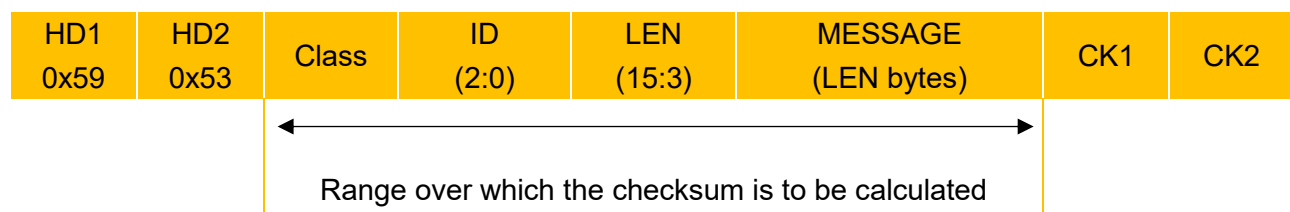


图 4 串口输入协议校验范围

输入（设置）协议的校验和计算公式与 2.1.3 一致。

2.3 SPI 输出协议

SPI 接口使用了比串口更精简的指令系统，使用方式如同常见的 IMU 读取方法，即采样寄存器地址的方式。

SPI 主机发送的第一个字节必须是包含寄存器地址的数据，该字节的第一个 bit 是读写位，1 表示读操作，0 表示写操作，格式如下表 10 所示。

MSB							LSB
R/W	A6	A5	A4	A3	A2	A1	A0

表 10 SPI/I2C 寄存器地址定义

SPI 读序列：

Master	AD + R			
Slave		DATA0	...	DATA _n

寄存器地址定义如下表 11 所示。

寄存器地址	说明	数据长度	备注
0x10	加速度	12 字节	从所有寄存器地址中读取的数据都是按照说明（即表格第二列）中描述的顺序发送。
0x20	角速度	12 字节	
0x40	欧拉角	12 字节	
0x05	加速度、角速度、磁力计	36 字节	
0x07	欧拉角、外部同步时间戳、Data Ready 时间戳	20 字节	
0x09	加速度、角速度、磁力计、欧拉角、外部同步时间戳、Data Ready 时间戳	56 字节	

表 11 寄存器地址定义

各个发送数据的值与真实值直接的对应关系如下表 12 所示。

类型	数据	数据转换	单位
加速度	DATA1 (DATA[7:0])	$ax = DATA \times 0.000001$	m/s ²

	DATA2 (DATA[15:8])			
	DATA3 (DATA[23:16])			
	DATA4 (DATA[31:24])			
	DATA5 (DATA[7:0])			
	DATA6 (DATA[15:8])	ay = DATA × 0.000001		
	DATA7 (DATA[23:16])			
	DATA8 (DATA[31:24])			
	DATA9 (DATA[7:0])			
	DATA10 (DATA[15:8])	az = DATA × 0.000001		
	DATA11 (DATA[23:16])			
	DATA12 (DATA[31:24])			
角速度	DATA1 (DATA[7:0])	wx = DATA × 0.000001		deg/s
	DATA2 (DATA[15:8])			
	DATA3 (DATA[23:16])			
	DATA4 (DATA[31:24])			
	DATA5 (DATA[7:0])	wy = DATA × 0.000001		
	DATA6 (DATA[15:8])			
	DATA7 (DATA[23:16])			
	DATA8 (DATA[31:24])			
	DATA9 (DATA[7:0])	wz = DATA × 0.000001		
	DATA10 (DATA[15:8])			
	DATA11 (DATA[23:16])			
	DATA12 (DATA[31:24])			
磁场强度	DATA1 (DATA[7:0])	mx = DATA × 0.001		mGauss
	DATA2 (DATA[15:8])			
	DATA3 (DATA[23:16])			
	DATA4 (DATA[31:24])			
	DATA5 (DATA[7:0])	my = DATA × 0.001		
	DATA6 (DATA[15:8])			
	DATA7 (DATA[23:16])			
	DATA8 (DATA[31:24])			
	DATA9 (DATA[7:0])	mz = DATA × 0.001		
	DATA10 (DATA[15:8])			
	DATA11 (DATA[23:16])			
	DATA12 (DATA[31:24])			
欧拉角	DATA1 (DATA[7:0])	pitch = DATA × 0.000001		deg(°)
	DATA2 (DATA[15:8])			
	DATA3 (DATA[23:16])			

	DATA4 (DATA[31:24])	roll = DATA × 0.000001	
	DATA5 (DATA[7:0])		
	DATA6 (DATA[15:8])		
	DATA7 (DATA[23:16])		
	DATA8 (DATA[31:24])		
	DATA9 (DATA[7:0])	yaw = DATA × 0.000001	
	DATA10 (DATA[15:8])		
	DATA11 (DATA[23:16])		
	DATA12 (DATA[31:24])		
外部同步时间戳	DATA1 (DATA[7:0])	Ext_sync_timestamp = DATA	us
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		
Data Ready 时间戳	DATA1 (DATA[7:0])	Data_ready_timestamp = DATA	us
	DATA2 (DATA[15:8])		
	DATA3 (DATA[23:16])		
	DATA4 (DATA[31:24])		

表 12 发送数据与真实数据的转换关系

3 协议参考

3.1 协议默认配置

3.1.1 通信接口默认配置

接口	典型值	单位
UART	波特率	460800
	数据位	8
	停止位	1
	奇偶校验	无
	硬件流控	无
SPI	模式	从机
	CPOA	CPOA_LOW
	CPHA	CPHA_1 ST _EDGE
	FIRST_BIT	MSB
	最大 CLOCK 值	6

3.1.2 输出协议默认设置

默认配置输出协议频率为 100Hz，输出内容为全部支持的数据。产品默认输出的内容为加计、陀螺、磁力计、欧拉角、四元数。

3.1.3 输入（设置）默认配置参数

- 输出波特率默认值为 0x04，即 460800bps。
- 输出频率默认值为 0x08，即 100Hz。
- 输出内容默认值为 0x00F8，即全部输出。

3.2 协议示例

3.2.1 串口输出协议示例

示例：

```
0x59 ,0x53 ,0x40 ,0x92 ,0x58 ,0x10 ,0x0c ,0x3b ,0x21 ,0xfe ,0xff ,0x89 ,0x2c ,0xfe ,0xff ,
0x59 ,0x9c ,0x6a ,0xff ,0x20 ,0x0c ,0xd1 ,0xa2 ,0x02 ,0x00 ,0xfa ,0xb1 ,0x05 ,0x00 ,0x38 ,
0xd0 ,0x00 ,0x00 ,0x30 ,0x0c ,0x20 ,0xff ,0xf1 ,0x07 ,0xd0 ,0xa0 ,0xea ,0xf4 ,0xc0 ,0x8e ,
0xc6 ,0xef ,0x31 ,0x0c ,0xb4 ,0x08 ,0x02 ,0x00 ,0xa2 ,0x29 ,0xfd ,0xff ,0xb8 ,0xd8 ,0xfb ,
0xff ,0x40 ,0x0c ,0xcb ,0xe4 ,0xf4 ,0xff ,0xb1 ,0xbe ,0x09 ,0x00 ,0x30 ,0x32 ,0xb6 ,0xf6 ,
0x41 ,0x10 ,0xd4 ,0x31 ,0x03 ,0x00 ,0x4d ,0xec ,0xff ,0xff ,0x86 ,0xe5 ,0xff ,0xff ,0xa9 ,
0x14 ,0xf1 ,0xff ,0x12 ,0xf3
```

以上输出数据各部分的解释如下：

序号		
1	0x59 ,0x53 ,	数据头
2	0x40 ,0x92 ,	TID
3	0x58 ,	有效数据长度
4	0x10 , ... , 0xff ,	有效数据
5	0x12 ,0xf3	校验和

有效数据部分的解析如下：

```
0x10 ,0x0c , 0x3b ,0x21 ,0xfe ,0xff ,0x89 ,0x2c ,0xfe ,0xff , 0x59 ,0x9c ,0x6a ,0xff
```

0x10: Data ID, 表示加速度;

0x0c: 表示该数据类型的长度;

剩余的为真实数据,每四个字节组成一个int型(有符号)数据,小端模式,然后乘以系数0.000001,如第一组四字节数据为0x3B 0x21 0xFE 0xFF,组成int型数据即为0xFFFE213B = -122565(dec),即真实的x轴加速度为 $122565 * 0.000001 = -0.122565 \text{ m/s}^2$ 。

```
0x20 ,0x0c , 0xd1 ,0xa2 ,0x02 ,0x00 ,0xfa ,0xb1 ,0x05 ,0x00 ,0x38 , 0xd0 ,0x00 ,0x00
```

0x20: Data ID, 表示角速度;

0x0c: 表示该数据类型的长度;

剩余的为真实数据,每四个字节组成一个int型(有符号)数据,小端模式,然后乘以系数0.000001,如第二组四字节数据为0xFA 0xB1 0x05 0x00,组成int型数据即为0x0005B1FA = 373242(dec),即真实的y轴角速度为 $373242 * 0.000001 = 0.373242 \text{ deg/s}$ 。

```
0x30 ,0x0c , 0x20 ,0xff ,0xf1 ,0x07 ,0xd0 ,0xa0 ,0xea ,0xf4 ,0xc0 ,0x8e , 0xc6 ,0xef
```

0x30: Data ID, 表示磁场归一化值;

0x0C: 表示该数据类型的长度;

剩余的为真实数据, 每四个字节组成一个 int 型(有符号)数据, 小端模式, 然后乘以系数 0.001, 如第三组四字节数据为 0xC0 0x8E 0xC6 0xEF, 组成 int 型数据即为 0xEFC68EC0 = -272200000 (dec), 即真实的 z 轴磁场归一化值为 $-272200000 * 0.000001 = -272.200000$ 。

0x31, 0x0C, 0xb4, 0x08, 0x02, 0x00, 0xa2, 0x29, 0xfd, 0xff, 0xb8, 0xd8, 0xfb, 0xff

0x31: Data ID, 表示磁场实际强度;

0x0C: 表示该数据类型的长度;

剩余的为真实数据, 每四个字节组成一个 int 型(有符号)数据, 小端模式, 然后乘以系数 0.001, 如第一组四字节数据为 0xB4 0x08 0x02 0x00, 组成 int 型数据即为 0x000208B4 = 133300 (dec), 即真实的 x 轴磁场归一化值为 $133300 * 0.001 = 133.30$ mGauss。

0x40, 0x0C, 0xcb, 0xe4, 0xf4, 0xff, 0xb1, 0xbe, 0x09, 0x00, 0x30, 0x32, 0xb6, 0xf6

0x40: Data ID, 表示欧拉;

0x0C: 表示该数据类型的长度;

剩余的为真实数据, 每四个字节组成一个 int 型(有符号)数据, 小端模式, 然后乘以系数 0.000001, 如第三组四字节数据为 0x30 0x32 0xB6 0xF6, 组成 int 型数据即为 0xF6B63230 = -155831760 (dec), 即真实的 yaw 为 $-155831760 * 0.000001 = -155.831760$ deg。

0x41, 0x10, 0xd4, 0x31, 0x03, 0x00, 0x4d, 0xec, 0xff, 0xff, 0x86, 0xe5, 0xff, 0xff, 0xa9, 0x14, 0xf1, 0xff

0x41: Data ID, 表示四元数;

0x10: 表示该数据类型的长度;

剩余的为真实数据, 每四个字节组成一个 int 型(有符号)数据, 小端模式, 然后乘以系数 0.000001, 如第四组四字节数据为 0xA9 0x14 0xF1 0xFF, 组成 int 型数据即为 0xFFFF114A9 = -977751 (dec), 即真实的四元数第四个数为 $-977751 * 0.000001 = -0.977751$ 。

3.2.2 串口输入协议示例

功能示例		指令/返回值
/*设置输出频率 20Hz 到 RAM*/	输入指令	59 53 03 09 00 05 11 2C
	成功返回值	59 53 03 09 00 00 0c 27
	失败返回值	59 53 03 09 00 ff 0b 26
/*设置输出频率 20Hz 到 ROM*/	输入指令	59 53 03 0A 00 05 12 2F
	成功返回值	59 53 03 0a 00 00 0d 2a
	失败返回值	59 53 03 0a 00 ff 0c 29

/*设置不输出任何数据, 掉电不保存*/	输入指令	59 53 04 11 00 00 00 15 58
	成功返回值	59 53 04 09 00 00 0d 2b
	失败返回值	59 53 04 09 00 ff 0c 2a
/*设置只输出加计, 掉电不保存*/	输入指令	59 53 04 11 00 80 00 95 58
	成功返回值	59 53 04 09 00 00 0d 2b
	失败返回值	59 53 04 09 00 ff 0c 2a
/*设置只输出陀螺, 掉电不保存*/	输入指令	59 53 04 11 00 40 00 55 D8
	成功返回值	59 53 04 09 00 00 0d 2b
	失败返回值	59 53 04 09 00 ff 0c 2a
/*设置不输出磁力计, 掉电不保存*/	输入指令	59 53 04 11 00 D8 00 ED 08
	成功返回值	59 53 04 09 00 00 0d 2b
	失败返回值	59 53 04 09 00 ff 0c 2a
/*设置输出全部数据, 掉电保存*/	输入指令	59 53 04 12 00 F8 00 0E 4C
	成功返回值	59 53 04 0a 00 00 0e 2e
	失败返回值	59 53 04 0a 00 ff 0d 2d
/*设置只输出欧拉角, 掉电保存*/	输入指令	59 53 04 12 00 10 00 26 7C
	成功返回值	59 53 04 0a 00 00 0e 2e
	失败返回值	59 53 04 0a 00 ff 0d 2d
/*串口波特率设置为 115200 到 ram*/	输入指令	59 53 02 09 00 03 0E 26
	成功返回值	无
	失败返回值	无
/*串口波特率设置为 460800 到 ram*/	输入指令	59 53 02 09 00 04 0F 27
	成功返回值	无
	失败返回值	无
/*串口波特率设置为 460800 到 rom*/	输入指令	59 53 02 0A 00 04 10 2A
	成功返回值	无
	失败返回值	无
/*航向实时置零*/	输入指令	59 53 01 09 00 02 0c 21
	成功返回值	59 53 01 09 00 00 1f 0a
	失败返回值	59 53 01 09 00 ff 1e 09
/*设置当前算法模式为 AHRS 到 ROM*/	输入指令	59 53 4d 12 00 02 01 62 ce
	成功返回值	59 53 4d 0a 00 00 52 57
	失败返回值	59 53 4d 0a 00 ff 51 56
/*设置 SYNC OUT 引脚模式为	输入指令	59 53 4d 11 00 4f 02 af 65

跟随 PPS 到 RAM*/	成功返回值	59 53 4d 0a 00 00 4f 56
	失败返回值	59 53 4d 0a 00 ff 4e 55
/*标定姿态角 (pitch/roll) 为 0 到 ROM*/	输入指令	59 53 05 32 00 11 04 00 00 00 00 4c 37
	成功返回值	59 53 4d 0a 00 00 52 57
	失败返回值	59 53 4d 0a 00 ff 51 56

注：

请谨慎配置波特率。波特率的设置需要考虑输出数据量以及输出频率，若设置的波特率过低会导致完成配置后数据发送不完整。一般地，波特率的最小值可按下述方式估算：

1. 估算串口数据传输时间。在当前频率下，数据输出周期时间减去 5ms 作为串口数据传输时间 $t(\text{ms})$ ；
2. 估算波特率。确定当前输出内容每帧数据量 n 字节（默认输出每帧 95 字节数据），则需要的波特率 $= n * 10 * 1000 / t$ 。

4 Revisions

Revision	Date	By	Changes
1.0	2018-10-30	Z. L.	• Initial release
1.1	2018-11-15	Z. L.	• Add SPI output protocol
1.2	2018-11-30	F. L.	• Update SPI output protocol
1.3	2019-05-15	F. L.	• Update output protocol