



# 超高精度 MEMS 6 自由度惯性传感器

## FSS-IMU1775 产品手册

### 特性

#### 战术级 MEMS 陀螺仪

- $0.07^{\circ} / \text{hr}$  零偏不稳定性
- $0.005^{\circ} / \sqrt{\text{hr}}$  角度随机游走
- $5^{\circ} / \text{hr}$  温漂 ( $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ ,  $\leq 1^{\circ}\text{C}/\text{min}$  @  $1\sigma$ )

#### 战术级 MEMS 加速度计

- $10 \mu\text{g}$  零偏不稳定性
- $0.009\text{m/s} / \sqrt{\text{hr}}$  速度随机游走
- $0.3\text{mg}$  温漂 ( $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ ,  $\leq 1^{\circ}\text{C}/\text{min}$  @  $1\sigma$ )

#### 大范围精细化温度补偿

- $-40^{\circ}\text{C}$  至  $85^{\circ}\text{C}$  温度补偿
- 精细化温度标定

#### 独立转台标定

- 独立标定每个模块：灵敏度、零偏、非正交误差
- 提供用户标定安装误差接口

#### 高强度工况耐受

- 超强冲击耐受：2000g (0.5ms, 半正弦, 3 轴)
- 超强振动耐受：10g (10~2KHz, 3 轴)
- 全温环境稳定工作： $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$
- 100%磁屏蔽

#### 实时而灵活的数字接口、体积小巧

- 高达 500Hz 的可配置输出采样率
- 支持 RS-422
- 尺寸  $\varnothing 88.9 \times 29.7\text{mm}$ , 重量约 242g

#### 产品概述

FSS-IMU1775 是原极科技倾力打造的 6 自由度超高精度 MEMS 惯性传感器模块。标配输出三轴陀螺仪与加速度信息。

高精度、高分辨率，可捕捉细微的震动与倾斜。大量程的输出，让大动态下的动作感知成为可能。所有模块出厂前都配置超宽温域的精细化温补与独立标定，让每个模块都能在各种极限工况下稳定发挥，同时保证所有产品性能高度一致。

#### 应用领域

- 航空测绘 等其他高精度惯导应用场景

在标准性能及输出参数的基础上，原极也为您的特殊需求提供定制化软件及 LOGO 定制服务，在产品上助您一臂之力！



## 目录

1. 性能参数	1
1.1 陀螺仪关键指标	1
1.2 加速度计关键指标	2
2. 外形结构	4
3. 规格	5
3.1 最大额定绝对值	5
3.2 EMC	5
3.3 ESD 等级	5
4. 电气特性	6
5. 引脚定义	7
6. 推荐连接方式	9
7. 串口通信协议	10
7.1 串口接口参数	10
7.2 数据包格式	11
7.3 数据流帧——AHRS 数据	12
7.4 命令模式 GET 输出——系统状态	13
7.5 命令模式 GET 输出——读取参数	14
7.6 命令模式 SET 指令	15
7.7 命令模式输出——用户命令响应	17
8. 常用 AT 指令	18
8.1.1 停止当前数据流输出	18
8.1.2 查询版本号	18
8.1.3 查询用户参数	18
8.1.4 设置和查询 ODR	18
8.1.5 设置和查询坐标系	19
8.1.6 设置和查询波特率	19
8.1.7 设置和查询滤波器	19
8.1.8 保存参数	19
9. CRC 查表法计算	20
10. 更新记录	23

## 1. 性能参数

### 1.1 陀螺仪关键指标

表 1 陀螺仪关键指标

参数	测试条件/备注	最小值	典型值	最大值	单位
测量范围			±300		° /s
零偏不稳定性 <sup>2</sup>	@25°C, ALLAN 方差, 1σ		x: 0.1 yz: 0.07		° /hr
零偏稳定性	国军标, 10s 平滑		0.3		° /hr
零偏重复性	国军标		x: 2 yz: 0.5		° /hr
分辨率			0.01		° /s
轴间非正交			0.01		deg
内部低通截止频率	软件可调整		30		Hz
采样率			500		Hz
测量延时			5.5		ms
全温范围零偏变化 <sup>1</sup>	-40°C ~ 85°C ≤1°C/min @1σ		x: 15 yz: 5		° /hr
随机游走 <sup>2</sup>	@25°C, ALLAN 方差, 1σ		xy: 0.005 z: 0.008		° /√hr
刻度系数误差			0.7		‰
刻度系数非线性			100		ppm

注 1: 1°C/分钟升温情况下全温零偏变化 1σ 值

注 2: IEEE 标准, 在静态 25°C 环境下 Allan 方差曲线给出

## 1.2 加速度计关键指标

表 2 加速度计关键指标

参数	测试条件/备注	最小值	典型值	最大值	单位
测量范围			±8		g
零偏不稳定性 <sup>2</sup>	@25°C, ALLAN 方差, 1 $\sigma$		10		$\mu\text{g}$
零偏稳定性	国军标, 10s 平滑		15		$\mu\text{g}$
零偏重复性	国军标		0.15		$\mu\text{g}$
分辨率			0.5102		mg
轴间非正交			0.01		deg
内部低通截止频率	软件可调整		30		Hz
采样率			500		Hz
测量延时			5.5		ms
全温范围零偏变化 <sup>1</sup>	-40°C ~ 85°C $\leq 1^\circ\text{C}/\text{min}$ @1 $\sigma$		0.3		mg
随机游走 <sup>2</sup>	@25°C, ALLAN 方差, 1 $\sigma$		0.009		m/s/ $\sqrt{\text{hr}}$
刻度系数误差			0.15		‰
刻度系数非线性			50		ppm

注 1: 1°C/分钟升温情况下全温零偏变化 1  $\sigma$  值

注 2: IEEE 标准, 在静态 25°C 环境下 Allan 方差曲线给出

图 1 陀螺仪 ALLAN 方差典型曲线

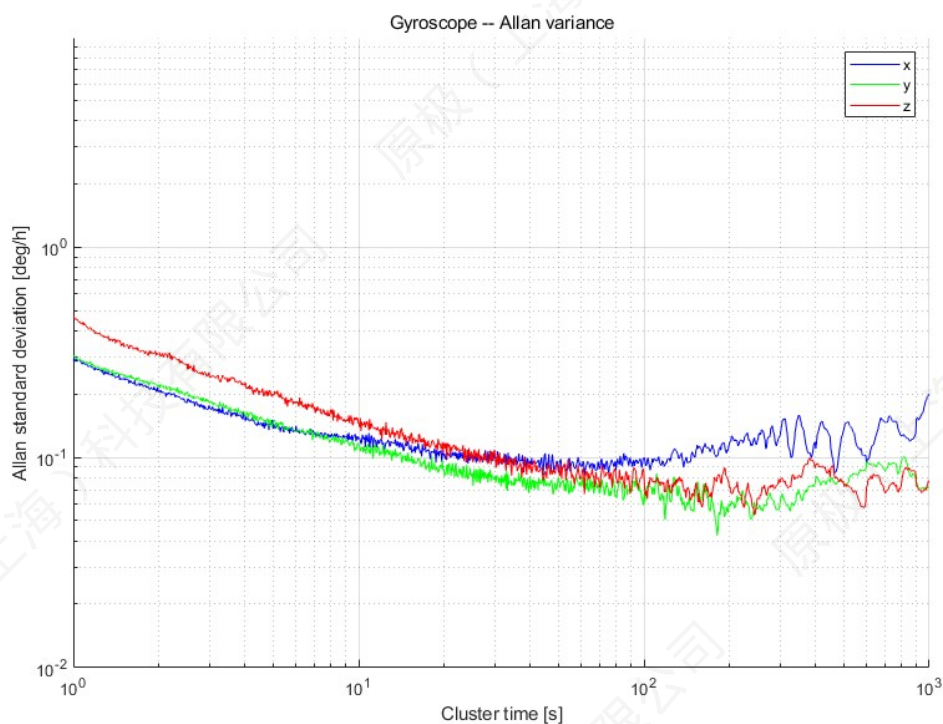
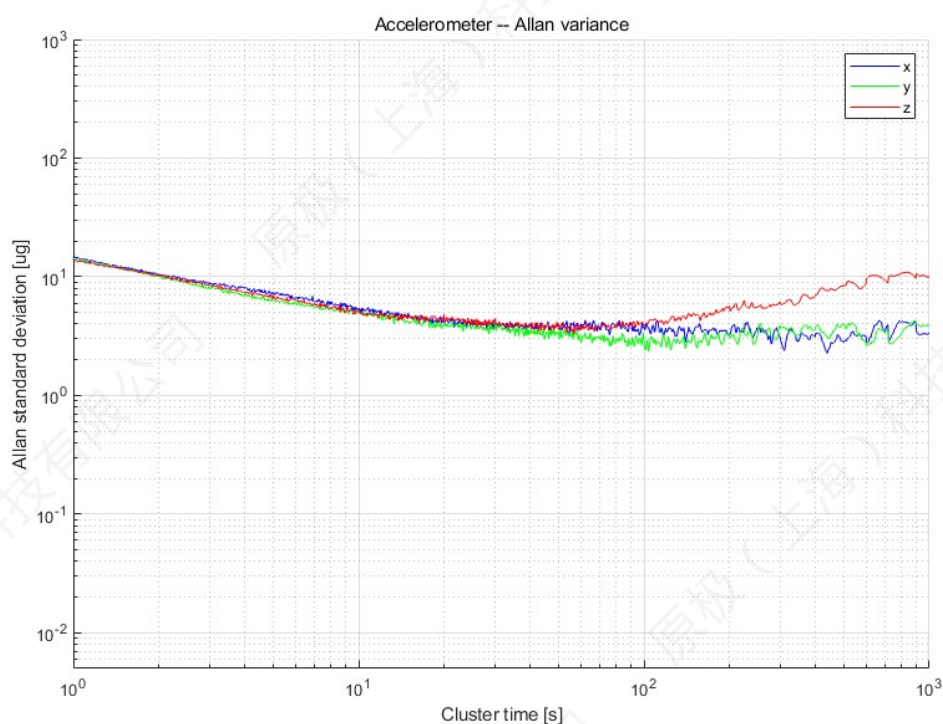


图 2 加速度计 ALLAN 方差典型曲线



## 2. 外形结构

图 3 外形结构及尺寸 (单位: mm)

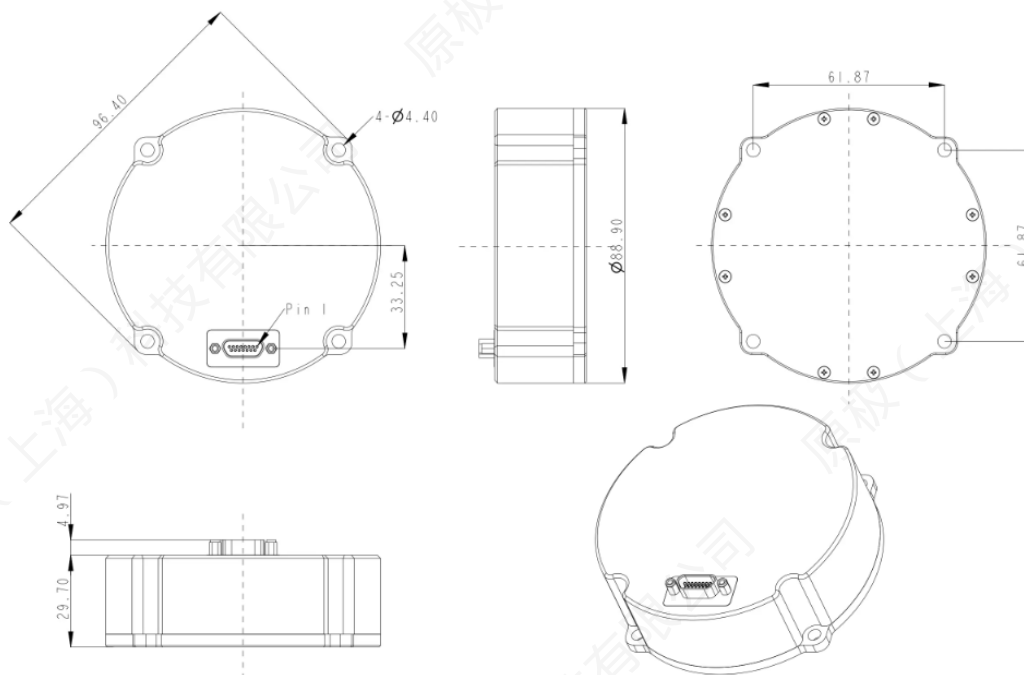
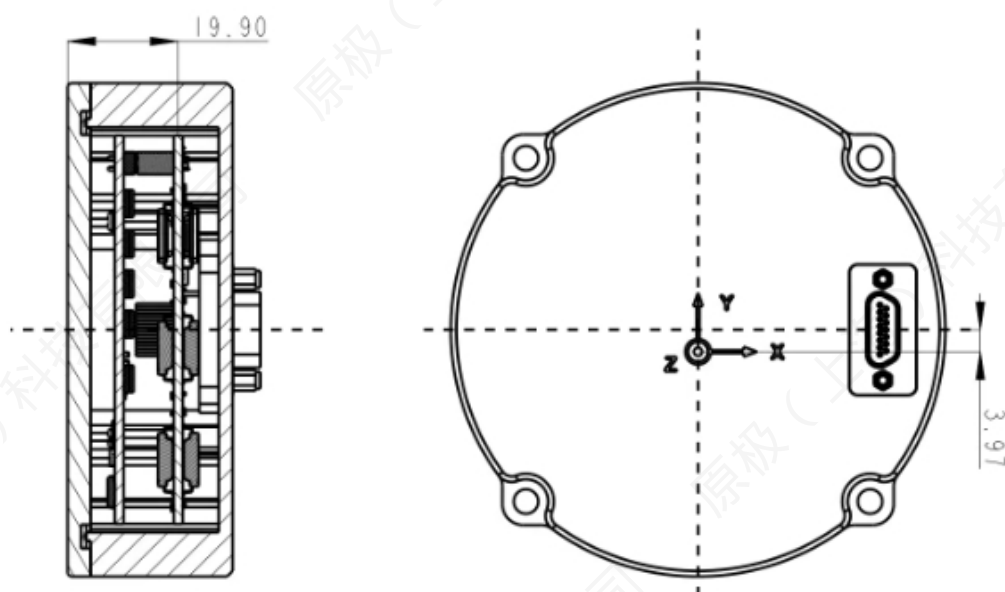


图 4 IMU 测量中心位置 (单位: mm)



### 3. 规格

#### 3.1 最大额定绝对值

表 3 绝对最大额定值

参数	级别	备注
储存温度	-55°C~90°C	
VSUP to GND	9~36V	
使用温度	-40~+85°C	
RXD+/RXD- to GND	-7.5V~12.5V	
TXD+/TXD- to GND	-7.5V~12.5V	
TOV-Out+/TOV-Out- to GND	-8~13V	
EXT-RST+/EXT-RST- to GND	-8~13V	
MSync+/MSync- to GND	-8~13V	
Config-RST-In+/Config-RST-In- to GND	-8~13V	

#### 3.2 EMC

表 4 EMC

测试项目	测试标准
CE	符合 EN55032 CISPR 16-2-1
CS	符合 EN55035 EN(IEC)61000-4-6
EFT	符合 EN55035 EN(IEC)61000-4-4
RE	符合 EN55032 CISPR 16-2-3
RS	符合 EN55035 EN(IEC)61000-4-3
ESD	符合 EN55035 EN(IEC)61000-4-2

#### 3.3 ESD 等级

表 5 ESD 等级

静电放电	测试模式	测试标准	测试等级
	接触放电	符合 EN(IEC)61000-4-2	±8 KV
	空气放电		±15 KV

## 4. 电气特性

表 6 电气特性

参数	条件	最小	典型	最大	单位
电源输入		9	12	32	V
功率			3.92		W
储存温度		-55		90	°C
使用温度		-40		85	°C
RS422 输入电阻			120		Ω



## 5. 引脚定义

图 5 引脚示意图

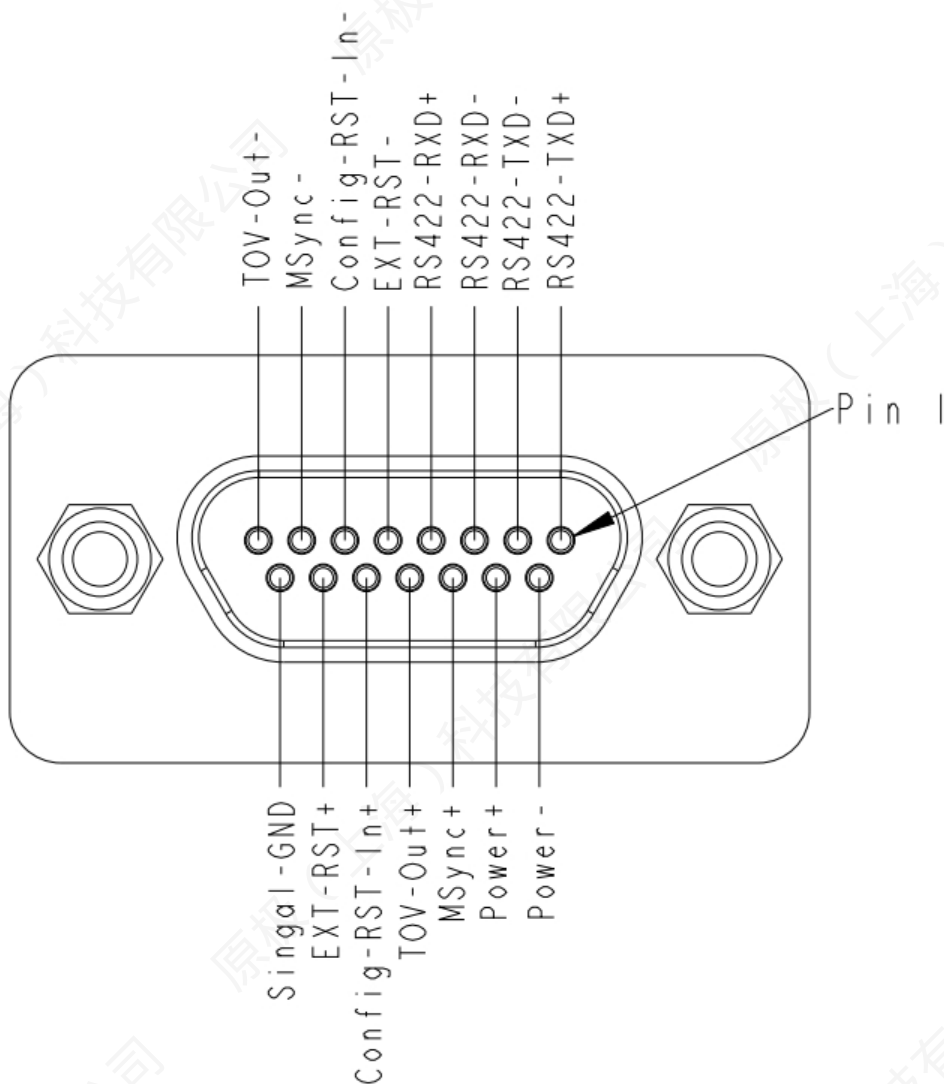


表 7 引脚定义

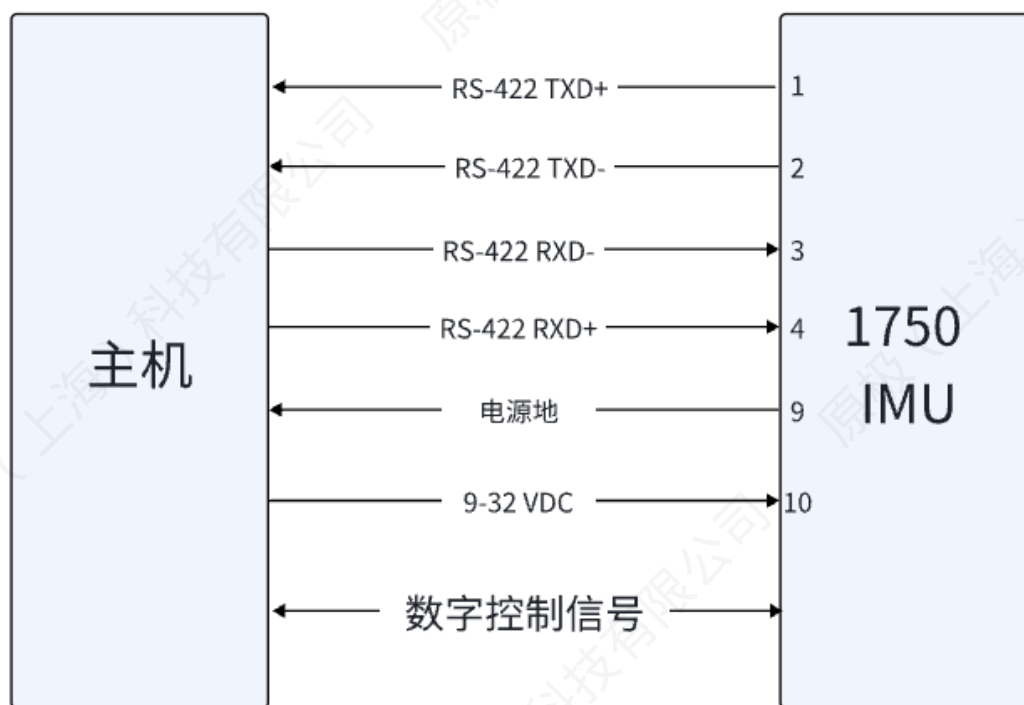
引脚序号	引脚名称	引脚描述	备注
1	RS-422, TXD+	IMU RS422 发送高电平	AHRS 数据
2	RS-422, TXD-	IMU RS422 发送低电平	
3	RS-422, RXD-	IMU RS422 接收低电平	
4	RS-422, RXD+	IMU RS422 接收高电平	



6	Config-RST-In-	IMU 配置重置低电平	配置重置
13	Config-RST-In+	IMU 配置重置高电平	
7	MSync-	主同步低电平	外部数据请求的主同步
11	MSync+	主同步高电平	
8	TOV-Out-	有效时间信号低电平	有效期
12	TOV-Out+	有效时间信号高电平	
9	Power-	电源地	电源
10	Power+	电源输入 9-32 VDC	
5	EXT-RST-	IMU 复位低电平	外部复位
14	EXT-RST+	IMU 复位高电平	
15	Singal-GND	信号地	信号地（可不连接）

## 6. 推荐连接方式

图 6 推荐连接框图



## 7. 串口通信协议

基于 QT、ROS 和 STM32 的串口协议示例：

<https://data.forsense-imu.com/page/download.html>

串口通信具有两种模式：数据流模式(Stream Mode)和命令模式(Command Mode)，IMU 在上电初始化完成后，根据参数配置的模式值进入对应模式。

数据流模式：以固定频率周期性输出 AHRS 数据；

命令模式：在此模式下，停止周期性输出，用户通过发送命令与 IMU 进行通信，可通过 GET 指令获取传感器数据、状态、参数等，也可配置 IMU 的参数。

### 7.1 串口接口参数

表 8 串口接口参数

传输速率范围	115200bps ~ 1.5Mbps
默认传输速率	115200bps
开始位	1 bit
数据位	8 bits
停止位	1 bit
奇偶校验	无

## 7.2 数据包格式

IMU 输出和用户输入的数据包结构组成如下：

表 9 IMU 输出和用户输入数据结构

偏移量	数据类型	名称	描述
0	uint8	帧头 1	IMU 输出帧头：0xAA, 0x55 用户输入帧头：0x55, 0xAA
1	uint8	帧头 2	
2	uint16	ID 低位	串口通信帧 ID 的低位字节
3		ID 高位	串口通信帧 ID 的高位字节
4	uint16	数据长度低位	串口通信帧长度的低位字节， length 为 payload 所占字节数，即为 n
5		数据长度高位	串口通信帧长度的高位字节， length 为 payload 所占字节数，即为 n
6	uint8	Payload (n 个字节)	数据负载
6+n	UInt32	CRC_CEHCK (32 位数据低字节)	CRC 校验
7+n		CRC_CEHCK (32 位数据中低字节)	
8+n		CRC_CEHCK (32 位数据中高字节)	
9+n		RC_CEHCK (32 位数据高字节)	

注 1：数据以小端格式传输，低字节在前，高字节在后

注 2：crc32 的初值为 1，CRC 计算不包括本身的本帧所有数据，查表算法见文档末尾

## 7.3 数据流帧——AHRS 数据

表 10 串口 AHRS 数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	A1	uint32
编码	0xAA	0x55	0x0002	0x002C		crc32

表 11 串口 A1 负载数据格式

offset	名称	数据类型	单位	描述
0	timer	uint32	$\mu s$	时间标
4	/	/	/	/
8	/	/	/	/
12	/	/	/	/
16	ax	float	g	X 轴加速度
20	ay	float	g	Y 轴加速度
24	az	float	g	Z 轴加速度
28	gx	float	$^{\circ}/s$	X 轴角速度
32	gy	float	$^{\circ}/s$	Y 轴角速度
36	gz	float	$^{\circ}/s$	Z 轴角速度
40	temp	float	$^{\circ}C$	IMU 芯片温度

例：获取到 AHRS 数据流：

```
AA 55 02 00 2C 00 6D 89 16 05 8F C2 65 40 14 AE 07 BF 5C 0F B2 43 25 06 81 3D
BC 74 13 3C 60 E5 80 BF EC 51 38 BD 0A D7 A3 BB CD CC CC BC D7 A3 EE 41 0C BF
84 80
```

解析如下：

表 12 串口 A1 获取到 AHRS 数据流

描述	原始值	解析值	描述	原始值	解析值
ID	0200	02	Y 轴加速度	BC74133C	0.009g
长度	2C00	44	Z 轴加速度	60E580BF	-1.007g
时间标	6D891605	85363053	X 轴角速度	EC5138BD	-0.045 $^{\circ}/s$
俯仰角	8FC26540	3.59 $^{\circ}$	Y 轴角速度	0AD7A3BB	-0.005 $^{\circ}/s$
横滚角	14AE07BF	-0.53 $^{\circ}$	Z 轴角速度	CDCCCCBC	-0.025 $^{\circ}/s$
航向角	5C0FB243	356.12 $^{\circ}$	imu 芯片温度	D7A3EE41	29.83 $^{\circ}C$
X 轴加速度	2506813D	0.063g	crc32 校验	0CBF8480	2156183308

## 7.4 命令模式 GET 输出——系统状态

表 13 串口系统状态数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	S1	uint32
编码	0xAA	0x55	0x00FF	N		crc32

注 1: 不同 IMU 型号, 此帧的长度会有差别, 都代表 S1 的长度, 需要根据 imu 型号确认。

表 14 串口 S1 负载数据格式

offset	名称	数据类型	描述
0	Software_ver	uint32	软件版本号
4	Hardware_ver	uint32	硬件版本号
8	rev	uint16	保留字节
10	sn0	uint32	第一 SN 号
14	sn1	uint32	第二 SN 号
18	sn2	uint32	第三 SN 号
22	Board_version	uint32	底板版本号
26	Rev[16]	Uin8	后续都是保留字节

注 1: 不同 IMU 型号, 后续保留字节也不同, 需要根据 imu 型号进行确认, IMU614E 为 16 字节。

例: 获取系统状态

输入数据: 55 AA 01 00 18 00 BD DB 31 34

响应数据: AA 55 FF 00 2A 00 1F 39 03 00 65 6F 01 00 50 83 30 33 35 55 34 50 15 FF 8F 5F FF FF 50 83 FF 1F 29 00 00 00 00 E0 00 07 10 17 08 50 D0 37 10 3B 7A C3 00 02

根据响应数据, 解析得到软件版本号 211231 (1F 39 03 00), 硬件版本号 94053 (65 6F 01 00)。

## 7.5 命令模式 GET 输出——读取参数

表 15 串口参数输入数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	P1	uint32
编码	0x55	0xAA	0x0006	0x0018		crc32

表 16 串口参数输出数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	P1	uint32
编码	0xAA	0x55	0x7530	0x0018		crc32

注 1：读取参数时，IMU 会将数据流关闭，设置完毕后需要重新开启数据流。

表 17 串口 P1 负载数据格式

offset	名称	数据类型	描述
0	Param1	float	获取的参数（输入数据可无视）
4	Param2	float	保留，默认为 0
8	Param3	uint32	设置的参数索引
12	Param4	uint32	保留，默认为 0
16	Param5	Int32	保留，默认为 0
20	Param6	Int32	保留，默认为 0

表 18 串口 P1 负载参数索引表

Param3	Param1	单位
3	串口输出波特率，支持以下波特率 115200、230400、460800、921600、1500000	bps
4	坐标系朝向（见表 25 坐标系朝向对应表）	
8	X 轴陀螺零偏标定结果，GYRO_X_OFF	° /s
9	Y 轴陀螺零偏标定结果，GYRO_Y_OFF	° /s
10	Z 轴陀螺零偏标定结果，GYRO_Z_OFF	° /s
21	AHRS 输出频率，默认 100Hz	Hz
31	内部滤波器配置，定义同 SPI 的 FILTER_CTRL 对照表	

例：获取 AHRS 输出频率

输入数据：55 AA 06 00 18 00 00 00 00 00 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 CB 46 AC

响应数据：AA 55 30 75 18 00 00 00 48 42 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 31 2F A2 0A

根据响应数据，解析得到输出频率为 50hz (00 00 48 42)。



## 7.6 命令模式 SET 指令

表 19 串口输入命令格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	R1	uint32
编码	0x55	0xAA	CMD	0x0018		crc32

注 1: CMD 与 R1 关系, 详见 R1 负载参数索引表

表 20 串口 R1 负载数据格式

offset	名称	数据类型	描述
0	Param1	float	设置的参数
4	Param2	float	保留, 默认为 0
8	Param3	uint32	设置的参数索引
12	Param4	uint32	保留, 默认为 0
16	Param5	Int32	保留, 默认为 0
20	Param6	Int32	保留, 默认为 0

表 21 串口 R1 负载参数索引表

CMD	Param1	Param3	描述
1	0	0	触发获取一次系统状态数据
2	0	0	触发获取一次 AHRS 数据
3	<mode>	0	设置输出模式: Mode=1, 数据流输出 AHRS Mode=100, 禁止数据流模式, 进入 COMMAD 模式
5	0	0	保存当前参数到 FLASH
6	0	<value>	读取参数, value 为要读取的参数索引, 即 P1.index, 详见串口应答性输出-参数读取 例如需读取 AHRS 输出频率 (ODR), 则设置 value=21 例如需读取串口波特率, 则设置 value=3 例如需读取内部滤波器, 则设置 value=31 例如需读取坐标系方向, 则设置 value=4
9	0	0	执行软件重启
14	<value>	3	设置串口输出波特率, 单位 bps value 的有效值为: 115200, 230400, 460800, 921600, 1500000 value 为其他值时, 默认采用 115200bps 设置波特率参数后, 需要重启才能生效。 不断电的设置流程: 设置波特率, 保存参数到 flash, 执行软件复位
14	<value>	21	设置周期性 AHRS 数据输出频率, 单位 Hz value 的常用值为:



CMD ID 填入 3，参数 1 填入 1，生成的十六进制数组可以填入串口助手或程序数组中发送给 IMU。

[illegible]

3

4

1

2

0

3

0

4

0

5

0

6

0

发送命令



### 表 22 设置参数串口响应数据格式

表 23 保留参数串口响应数据格式表 24 串口用户命令响应数据格式

例：设置串口输出波特率 115200

[illegible]

响应数据: AA 55 3D 75 04 00 34 75 03 00 A7 98 2A 54

设置周期性 AHRS 数据输出频率 100hz

输入数据: 55 AA 0E 00 18 00 00 00 C8 42 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 2B 2C 8D

响应数据: AA 55 3D 75 04 00 34 75 15 00 70 2D B2 48

## 保存当前参数到 FLASH

[illegible]

响应数据: AA 55 3D 75 04 00 05 00 01 00 5A CF B1 7C

### 设置输出模式为 AHRS 数据流

输入数据：55 AA 03 00 18 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 52 D8 8E E8

响应数据: AA 55 64 00 04 00 03 00 01 00 E7 87 E3 AD

## 8. 常用 AT 指令

### 8.1.1 停止当前数据流输出

指令：AT+SETNO\r\n

应答：OK\r\n

可以停掉当前数据流（不改数据流参数），应答 OK 后表示可以进行下一步操作。

如果未响应，可以继续发送 AT+SETNO\r\n 命令直到应答 OK。

### 8.1.2 查询版本号

指令：AT+VERSION\r\n

应答：SW_VERSION	固件版本
HW_VERSION	硬件版本
BOARD_VERSION	底板版本

### 8.1.3 查询用户参数

指令：AT+CONFIG\r\n

应答：BAUD_RATE	当前串口波特率
ORIENT	当前坐标系
IMU_ODR	当前 IMU 的输出频率
STREAM_MODE1	当前串口 1 的数据流模式
STREAM_MODE2	当前串口 2 的数据流模式
STREAM_MODE3	当前串口 3 的数据流模式
LP_CONFIG_REG	当前 IMU 的滤波

### 8.1.4 设置和查询 ODR

例：设置输出频率 ODR 为 50hz

指令：AT+SET\_ODR=50

应答：IMU\_ODR:50

查询 IMU 的 ODR

指令：AT+GET\_ODR

应答：IMU\_ODR:

### 8.1.5 设置和查询坐标系

例：设置 IMU 坐标系为右前上

指令：AT+SET\_ORIENT=101\r\n

应答：orientation:101

查询 IMU 当前坐标系

指令：AT+GET\_ORIENT\r\n

应答：orientation:

### 8.1.6 设置和查询波特率

例：设置 IMU 的波特率为 115200

指令：AT+SET\_BAUD=115200\r\n

应答：OK

查询 IMU 当前波特率

指令：AT+GET\_BAUD\r\n

应答：BAUD\_RATE:

### 8.1.7 设置和查询滤波器

例：设置 IMU 的滤波为 20hz

指令：AT+SET\_LPF=20\r\n

应答：LP\_CONFIG\_REG:20

查询 IMU 当前滤波

指令：AT+GET\_LPF\r\n

应答：LP\_CONFIG\_REG:

### 8.1.8 保存参数

指令：AT+SAVE\r\n

应答：OK

## 9. CRC 查表法计算

建议直接参考示例代码。

注 1：数据以小端格式传输，低字节在前，高字节在后

注 2：crc32 的初值为 1，CRC 计算不包括本身的本帧所有数据

C

```
static const uint32_t crc32_tab [ ] = {
0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706af48f,
0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91, 0x1db71064,
0x6ab020f2,
0xf3b97148, 0x84be41de, 0x1adad47d, 0x6ddde4eb, 0xf4d4b551,
0x83d385c7,
0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f,
0x63066cd9,
0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4,
0xa2677172,
0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b, 0x35b5a8fa,
0x42b2986c,
0xdbbbc9d6, 0xacbcf940, 0x32d86ce3, 0x45df5c75, 0xdcd60dcf,
0xabd13d59,
0x26d930ac, 0x51de003a, 0xc8d75180, 0xbf06116, 0x21b4f4b5,
0x56b3c423,
0xcfba9599, 0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2,
0xb10be924,
0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190,
0x01db7106,
0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f, 0x9fbfe4a5,
0xe8b8d433,
0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818, 0x7f6a0dbb,
0x086d3d2d,
0x91646c97, 0xe6635c01, 0x6b6b51f4, 0x1c6c6162, 0x856530d8,
0xf262004e,
0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6,
0x12b7e950,
0x8bbbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3,
0xfbd44c65,
0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2, 0x4adfa541,
0x3dd895d7,
```

0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a, 0x346ed9fc, 0xad678846,  
 0xda60b8d0,  
 0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c,  
 0x270241aa,  
 0xbe0b1010, 0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409,  
 0xce61e49f,  
 0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17,  
 0x2eb40d81,  
 0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6, 0x03b6e20c,  
 0x74b1d29a,  
 0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683, 0xe3630b12,  
 0x94643b84,  
 0x0d6d6a3e, 0x7a6a5aa8, 0xe40ecf0b, 0x9309ff9d, 0x0a00ae27,  
 0x7d079eb1,  
 0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d,  
 0x806567cb,  
 0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a,  
 0x67dd4acc,  
 0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5, 0xd6d6a3e8,  
 0xa1d1937e,  
 0x38d8c2c4, 0x4fdff252, 0xd1bb67f1, 0xa6bc5767, 0x3fb506dd,  
 0x48b2364b,  
 0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60, 0xdf60efc3,  
 0xa867df55,  
 0x316e8eef, 0x4669be79, 0xcb61b38c, 0xbc66831a, 0x256fd2a0,  
 0x5268e236,  
 0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f, 0xc5ba3bbe,  
 0xb2bd0b28,  
 0x2bb45a92, 0x5cb36a04, 0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b,  
 0x5bdeae1d,  
 0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a, 0x9c0906a9,  
 0xeb0e363f,  
 0x72076785, 0x05005713, 0x95bf4a82, 0xe2b87a14, 0x7bb12bae,  
 0x0cb61b38,  
 0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21, 0x86d3d2d4,  
 0xf1d4e242,  
 0x68ddb3f8, 0x1fda836e, 0x81be16cd, 0xf6b9265b, 0x6fb077e1,  
 0x18b74777,  
 0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c, 0x8f659eff,

```
0xf862ae69,  
0x616bfffd3, 0x166ccf45, 0xa00ae278, 0xd70dd2ee, 0x4e048354,  
0x3903b3c2,  
0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db, 0xaed16a4a,  
0xd9d65adc,  
0x40df0b66, 0x37d83bf0, 0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f,  
0x30b5ffe9,  
0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6, 0xbad03605,  
0xcdd70693,  
0x54de5729, 0x23d967bf, 0xb3667a2e, 0xc4614ab8, 0x5d681b02,  
0x2a6f2b94,  
0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d,  
}  
uint32_t crc_crc32 (uint32_t crc, const uint8_t *buf, uint32_t  
size) {  
    for (uint32_t i=0; i<size ; i++) {  
        crc = crc32_tab [ (crc ^ buf [i] ) & 0xff] ^ (crc >> 8 ) ;  
    }  
    return crc;  
}
```



## 10. 更新记录

版本	日期	状态/注释
版本 1.0	2024. 11. 13	初稿