车规级 MEMS
6 自由度惯性传感器

IMU 270 PRO 产品手册

特性

车规级 MEMS 陀螺仪

- 4.5° /hr 零偏不稳定性（X，Y 轴）
- 0.8° /hr 零偏不稳定性（Z 轴）
- 0.3°/$\sqrt{hr}$ 角度随机游走（X，Y 轴）
- 0.065°/$\sqrt{hr}$ 角度随机游走（Z 轴）
- 0.2°/s 温漂(-40 ~ 85℃, <=1℃/min@1σ) X，Y 轴
- 0.02°/s 温漂(-40 ~ 85℃, <=1℃/min @1σ) Z 轴

车规级 MEMS 加速度计

- 30ug 零偏不稳定性
- 0.03m/s/$\sqrt{hr}$ 速度随机游走
- 3mg 温漂(-40 ~ 85℃, <=1℃/min @1σ)

大范围精细化温度补偿

- -40℃至 85℃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

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）
- 稳定工作温度：-40℃ ~ 105℃
- 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
- 支持串口、SPI 多种接口
- 22.4*23.75*8mm，重量仅 6.5g

产品概述

IMU 270 PRO 是HSASK倾力打造的6自由度MEMS车规级惯性传感器模块。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

- 自动驾驶：乘用车、工程车、商务车

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

Change History 变更历史

Version 版本	Modified Date 修改日期	Approved Date 审批日期	Author 作者	Approver 审核人	Changes 修改内容
V1.0	2024-09-10	/	郝廉效	/	初次拟定
V1.1	2024-12-18	/	黄映峰	杨磊	更新软件协议

目录

2. 外形结构

图 1 外观造型

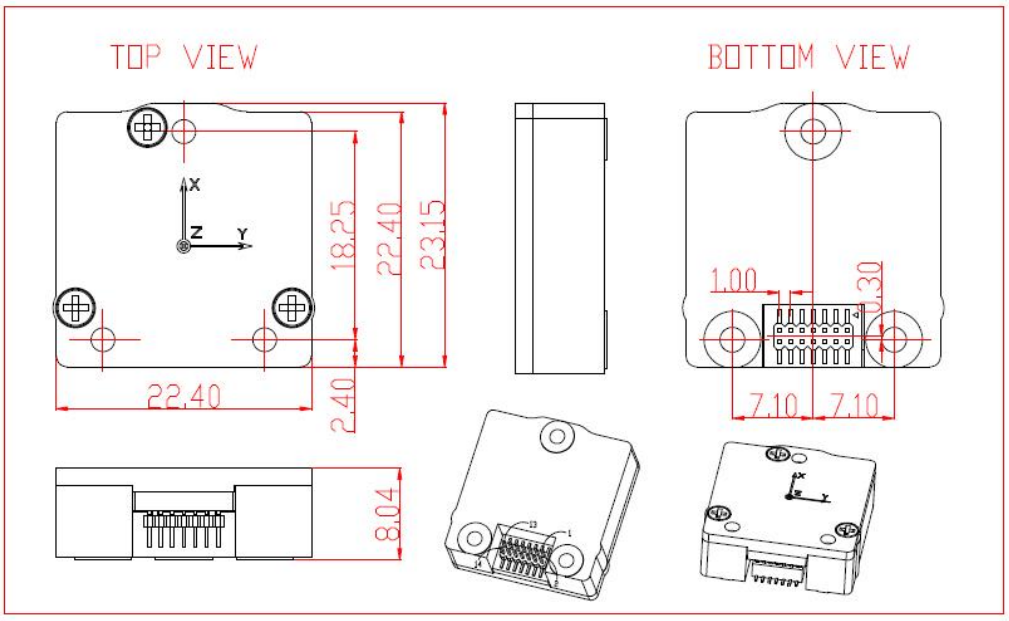


图 2 结构及安装尺寸

注意：

1、连接器的上排针对应螺丝孔的中心线

2、建议螺孔通孔直径为2.2mm。

3、建议使用 M2 螺钉

4、公差为：±0.1mm

3. 电气特性

3.1 最大耐受值

表 3 最大额定绝对值

参数	符号	范围	单位
供电电压	VCC	-0.3 to 4	V
电源地	GND	-	-
输入管脚电压	Vin	-0.3 to VCC+0.2	V
使用温度	Tot	-40 to 105	℃
存储温度	Tstg	-40 to 105	℃
ESD 等级	ESD	±2	KV

3.2 工作条件

表 4 工作条件

参数	符号	最小值	典型值	最大值	单位
供电电压	VCC	3.0	3.3	3.6	V
VCC 最大纹波	Vrpp	0	±40		mV
功耗	P		0.145		W
标定工作温度	Tot	-40		85	℃
扩展工作温度	Tot	-40		105	℃
存储和非工作温度	Tstg	-40		105	℃

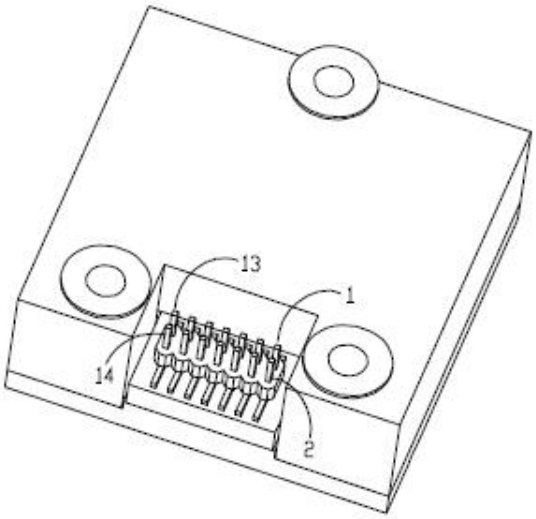
3.3 IO 阈值特性

表 5 IO 阈值特性

参数	符号	最小值	典型值	最大值	单位
输入管脚低电平	Vin_low	0		VCC*0.2	V
输入管脚高电平	Vin_high	VCC*0.7		VCC+0.2	V
输出管脚低电平	Vout_low	0		0.45	V
输出管脚高电平	Vout_high	VCC-0.45		VCC	V

4. 引脚定义

图 3 引脚示意图



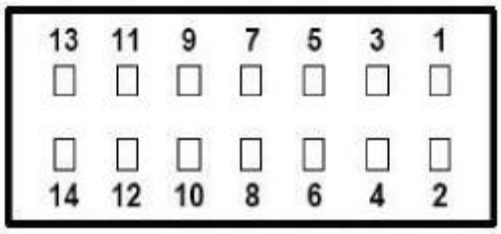


表 1 引脚定义

引脚序号	引脚名称	引脚描述
1	UART_TX 注[1]	串口发送； IMU 数据通信接口(LVTTL)
2	UART_RX	串口接收； IMU 数据通信接口(LVTTL)
3	IMU_MOSI 注[2]	SPI 串行数据输入
4	IMU_MISO	SPI 串行数据输出
5	IMU_CLK	SPI 串行时钟
6	IMU_CS	SPI 片选,低电平有效
7	NC	无连接
8	NC	无连接
9	NC	无连接
10	DR	DATA REDDY 外部同步采样触发信号
11	VCC	电源输入，+3.3V 输入，40mA，纹波不大于±40mV
12	FCCU_0注[3]	状态引脚指示 0
13	GND	电源地
14	FCCU_1	状态指示引脚 1

注 1： UART 通讯接口，默认波特率 115200bps。

注 2： SPI 作为从机，最高波特率 10MHz。

注 3：FCCU[0:1]为 IMU 状态指示引脚

模式	FCCU 0	FCCU 1	状态指示

1	0	0	IMU 复位状态
2	1	0	IMU 运行状态
3	0	1	IMU 错误状态
4	1	1	IMU 配置状态

5. 通信协议

5.1 串口通信协议

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。

5.1.1 串口接口参数

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

5.1.2 数据包格式

IMU 输出和用户输入的数据包结构组成如下：

用户输入数据结构

偏移量	数据类型	名称	描述
0	uint8	帧头 1	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		CRC_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，查表计算法见文档末尾

IMU 输出数据结构

偏移量	数据类型	名称	描述
0	uint8	帧头 1	IMU 输出帧头：0xAA，0x55
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	result（1 个字节）	数据返回值： 0 代表失败 1 代表成功 （具体错误码查看 payload 中的 errcode 字段）
7	uint8	payload （n 个字节）	数据负载
8+n	uint32	CRC_CEHCK （32 位数据低字节）	CRC 校验
9+n		CRC_CEHCK （32 位数据中低字节）	
10+n		CRC_CEHCK （32 位数据中高字节）	
11+n		CRC_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，查表计算法见文档末尾

5.1.3 数据流帧——AHRS 数据

串口 AHRS 数据格式

	帧头	帧头	ID	Result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	A1	uint32
编码	0xAA	0x55	0x0200	0x01	0x002C		crc32

注 1： 最大输出更新率不大于 500Hz@115200bps

表 10 串口 A1 负载数据格式

offset	名称	数据类型	单位	描述
0	timer	uint32	μs	时间标
4	pitch	float	°	俯仰角
8	roll	float	°	横滚角
12	yaw	float	°	航向角
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	°/s	X 轴角速度
32	gy	float	°/s	Y 轴角速度
36	gz	float	°/s	Z 轴角速度
40	temp	float	℃	IMU 芯片温度

例：获取到 AHRS 数据流：

AA 55 00 02 01 2C 00 AE 43 52 04 C3 F5 08 40 E1 7A 94 BE EC 11 B3 43 00 00 D8 3B 00 80 17 BD 00 28 78 BF 00 A0 0C 3D 00 3C A5 3E 00 F8 84 3E 00 00 18 42 3F 56 3E B1

解析如下：

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	0002	0200	Y 轴加速度	008017BD	-0.0369g
Result	01	1	Z 轴加速度	002878BF	-0.969g
长度	2C00	44	X 轴角速度	00A00C3D	0.0343°/s
时间标	AE435204	72500142	Y 轴角速度	003CA53E	0.3227°/s
俯仰角	C3F50840	2.14°	Z 轴角速度	00F8843E	0.2597°/s
横滚角	E17A94BE	-0.29°	IMU 芯片温度	00001842	38℃
航向角	EC11B343	358.14°	crc32 校验	3F563EB1	2973652543
X 轴加速度	0000D83B	0.00659g			

5.1.4 设置工作模式

设置工作模式数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	uint32	uint32
编码	0x55	0xAA	0x0001	0x0004	mode	crc32

mode 取值：

1 代表串口 AHRS 数据输出模式，串口会固定频率输出 IMU 的数据

100 代表 SPI 数据输出模式（上电默认 SPI 数据输出模式）

设置工作模式返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0001	res	0x0004	errcode	crc32

res 取值：

0 代表设置失败

1 代表设置成功

（具体错误码查看 payload 中的 errcode 字段）

errcode 取值：

参考 errcode 错误码表

例：设置工作模式

输入数据：55 AA 01 00 04 00 01 00 00 00 AD 0B F2 2E

响应数据：AA 55 01 00 01 04 00 00 00 00 00 E1 2D FC A7

根据响应数据，解析得到 res 为 1，设置串口 AHRS 数据输出模式成功。

5.1.5 设置串口波特率

设置串口波特率数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	uint32	uint32
编码	0x55	0xAA	0x0002	0x0004	baudrate	crc32

baudrate 取值：

115200/230400/460800/921600/1500000

设置串口波特率返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0002	res	0x0004	errcode	crc32

res 取值：

0 代表设置失败

1 代表设置成功

（具体错误码查看 payload 中的 errcode 字段）

errcode 取值：

参考 errcode 错误码表

例：设置串口波特率为 115200

输入数据：55 AA 02 00 04 00 00 C2 01 00 44 18 F1 93

响应数据：AA 55 02 00 00 04 00 0F 00 00 00 C6 0A 65 60

根据响应数据，可以看到 res 返回 0，error code 返回 0x0F，查表可知此时波特率与频率不匹配，可以根据频率与波特率对应表重新设置波特率或者频率。

而当此时设置的波特率与频率匹配的时候
响应数据：AA 55 02 00 01 04 00 00 00 00 00 24 11 71 9E
根据响应数据，可以看到 res 返回 1，设置成功，无错误码

设置波特率参数后，需要执行保存参数指令，重启（包含软件重启与断电重启）才能生效，默认采用 115200bps。

5.1.6 设置数据输出频率

设置数据输出频率数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	uint32	uint32
编码	0x55	0xAA	0x0003	0x0004	speed	crc32

speed 取值：

50/100/250/500 HZ

设置数据输出频率返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0003	res	0x0004	errcode	crc32

res 取值：

0 代表设置失败
1 代表设置成功
（具体错误码查看 payload 中的 errcode 字段）

errcode 取值：

参考 errcode 错误码表

例：设置数据输出频率
输入数据：55 AA 03 00 04 00 64 00 00 00 E1 0B 2F 60
响应数据：AA 55 03 00 01 04 00 00 00 00 00 67 05 0A 89
根据响应数据，解析得到 res 为 1，设置成功，无错误码

5.1.7 保存设置参数

保存设置参数数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0005	0	/	crc32

保存设置参数返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0005	res	0x0004	errcode	crc32

res 取值：

0 代表保存失败

1 代表保存成功

（具体错误码查看 payload 中的 errcode 字段）

errcode 取值：

参考 errcode 错误码表

例：保存设置参数

输入数据：55 AA 05 00 00 00 E4 3D 56 9A

响应数据：AA 55 05 00 01 04 00 00 00 00 00 ED 7C 10 FA

根据响应数据，解析得到 res 为 1，保存成功，无错误码

5.1.8 查询当前工作模式

查询当前工作模式数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0100	0	/	crc32

查询当前工作模式返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0100	res	0x0004	mode	crc32

res 取值：

0 代表获取失败

1 代表获取成功

（具体错误码查看 payload 中的 errcode 字段）

mode 取值：

1　代表串口 AHRS 数据输出模式，串口会固定频率输出 IMU 的数据

100　代表 SPI 数据输出模式（上电默认 SPI 数据输出模式）

例：获取当前工作模式

输入数据：55 AA 00 01 00 00 E1 A7 4A AC

响应数据：AA 55 00 01 01 04 00 64 00 00 00 68 59 69 C8

根据响应数据，解析得到 res 为 1，获取成功，payload 为 0x00000064，即 100，可知此时工作模式为 SPI 数据输出模式。（输出数据为小端模式数据）

5.1.9 查询当前串口波特率

查询当前串口波特率数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0101	0	/	crc32

查询当前串口波特率返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0101	res	0x0004	baudrate	crc32

res 取值：

0 代表获取失败

1 代表获取成功

（具体错误码查看 payload 中的 errcode 字段）

baudrate 取值：

115200/230400/460800/921600/1500000

例：查询当前串口波特率

输入数据：55 AA 01 01 00 00 84 C0 F6 14

响应数据：AA 55 01 01 01 04 00 00 C2 01 00 10 5E 66 E0

根据响应数据，解析得到 res 为 1，获取波特率成功，此时 payload 为 0x0001C200，即 115200 的波特率。（输出数据为小端模式数据）

5.1.10 查询当前数据输出频率

查询当前数据输出频率数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0102	0	/	crc32

查询当前数据输出频率返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0102	res	0x0004	speed	crc32

res 取值：

0 代表查询失败

1 代表查询成功

（具体错误码查看 payload 中的 errcode 字段）

speed 取值：

50/100/250/500 HZ

例：查询当前数据输出频率

输入数据：55 AA 02 01 00 00 6A 6F 43 06

响应数据：AA 55 02 01 01 04 00 32 00 00 00 90 21 F9 08

根据响应数据，解析得到 res 为 1，查询成功，payload 为 0x00000032，即 50，可知此时数据输出频率为 50hz。（输出数据为小端模式数据）

5.1.11 查询系统信息

查询系统信息数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0105	0	/	crc32

查询系统信息返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	S1	uint32
编码	0xAA	0x55	0x0105	res	0x0039		crc32

res 取值：

0 代表查询失败

1 代表查询成功

（具体错误码查看 payload 中的 errcode 字段）

串口 S1 数据格式

offset	名称	数据类型	长度
0	软件版本号	uint8 * 3	3
3	硬件版本号	uint8 * 3	3
6	板卡版本号	uint8 * 3	3
9	SN0	uint32 * 4	16
25	SN1	uint32 * 4	16
41	SN2	uint32 * 4	16

例：查询系统信息

输入数据：55 AA 05 01 00 00 D3 57 94 9B

响应数据：AA 55 05 01 01 39 00 01 00 04 01 00 00 18 09 0D 24 00 45 00 0A 50 4E 52 38 36 39 20 00 00 00 00 F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D7 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9B 59 6D 2D

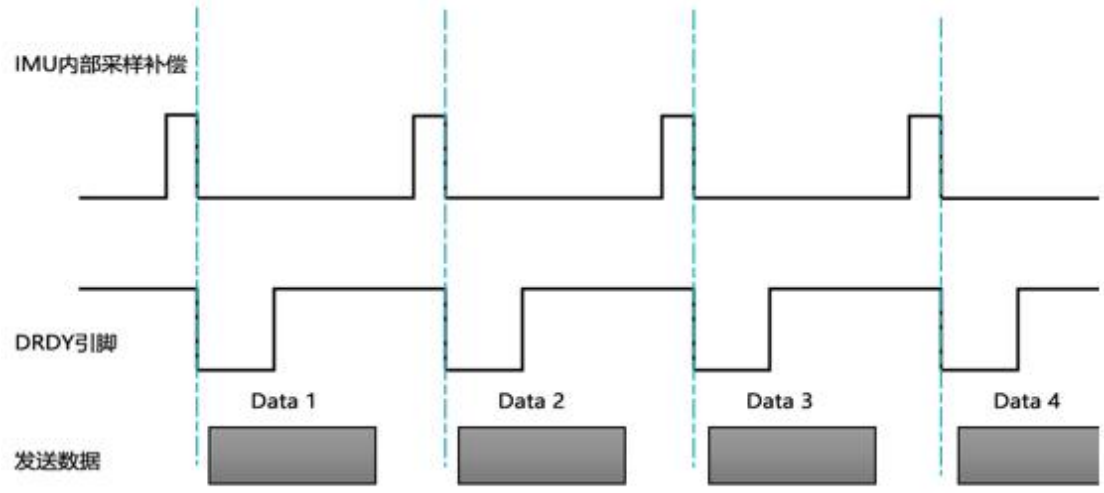
根据响应数据，解析得到 res 为 1，查询成功，软件版本号 1.0.4(01 00 04)，硬件版本号 1.0.0(01 00 00)。

5.1.12 DRDY

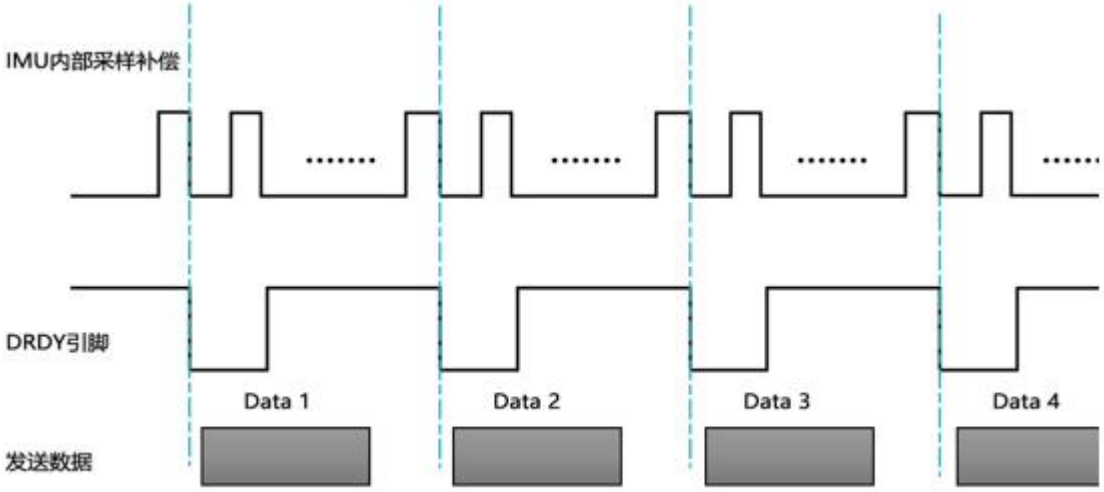
DRDY 引脚输出有两个目的：

1. 提供来自 IMU 内部的时钟同步信号；

2. 提供信号表示开始传送数据帧。



当 IMU 内部采样频率（最大 ODR）与串口输出频率（当前 ODR）一致时，每当 imu 数据采样补偿完成后，DRDY 引脚将被立即拉低，此时数据帧将从串口发送，在下一周期 DRDY 引脚将被重新拉高。

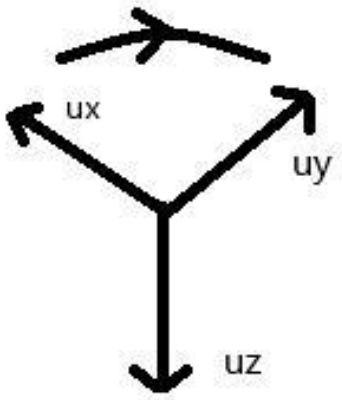


当串口输出频率小于 IMU 内部采样频率时，每当 imu 数据采样补偿完成后，根据分频计数值（最大 ODR/当前 ODR）决定 DRDY 引脚是否被立即拉低。DRDY 拉低后数据帧将从串口发送，在下一 IMU 采样周期 DRDY 引脚将被重新拉高。

5.1.13 设置坐标系功能

设置固件坐标系，在上位机当中显示对应固件设计坐标系

图 6 固件原始坐标系



按照上图规则，当 x 和 y 轴确定之后，z 轴确定。Z 轴垂直于 X 轴到 Y 轴的面。
X/Y/Z 三轴的朝向总共有二十四种，可见坐标系朝向对应表

设置坐标系数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	uint32	uint32
编码	0x55	0xAA	0x0009	0x0004	Coordinates Flag	crc32

Coordinates Flag 取值：

101 ~ 124　具体坐标系朝向对应关系见坐标系朝向对应表

设置坐标系返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0009	res	0x0004	errcode	crc32

res 取值：

0 代表设置失败

1 代表设置成功

（具体错误码查看 payload 中的 errcode 字段）

errcode 取值：

参考 errcode 错误码表

例：设置坐标系为 102 朝向：

输入数据：

响应数据：

5.1.14 查询坐标系功能

查询当前坐标系数据格式

	帧头	帧头	ID	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint16	/	uint32
编码	0x55	0xAA	0x0108	0	/	crc32

查询当前坐标系返回数据格式

	帧头	帧头	ID	result	length	payload	帧尾
数据类型	uint8	uint8	uint16	uint8	uint16	uint32	uint32
编码	0xAA	0x55	0x0108	res	0x0004	Coordinates Flag	crc32

res 取值：

0 代表查询失败

1 代表查询成功

（具体错误码查看 payload 中的 errcode 字段）

Coordinates Flag 取值：

101 ~ 124　具体坐标系朝向对应关系见坐标系朝向对应表

例：获取当前坐标系朝向：

输入数据：

响应数据：

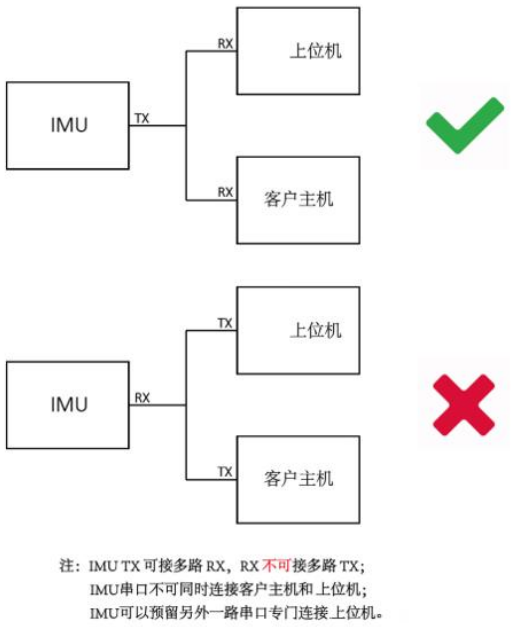
5.1.15 串口连接常见问题

1）IMU 的 RX 不能接 2 个主机 TX

串口的 RX 不能同时接 2 个 TX，所以如果需要连接我们上位机时，需要断开其与用户主机的串口通信，否则上位机只能接收到数据，不能发送命令给 IMU。

如下图所示：

图 7 串口连接方式示意图



注：IMU TX 可接多路 RX，RX 不可接多路 TX；
IMU串口不可同时连接客户主机和上位机；
IMU可以预留另外一路串口专门连接上位机。

2）获取不到版本号

检查串口线是否丢包，推荐使用 FT232 芯片的串口线，CH340、PL2303 数据线在高波特率时（>115200bps）会丢包

建议串口线直连，不建议串联，如 RS422 的接口接电脑，直接使用 RS422 转 USB 线，不要用 RS422 转 RS232+RS232Z 转 USB 线串联。

5.2 SPI 通信协议

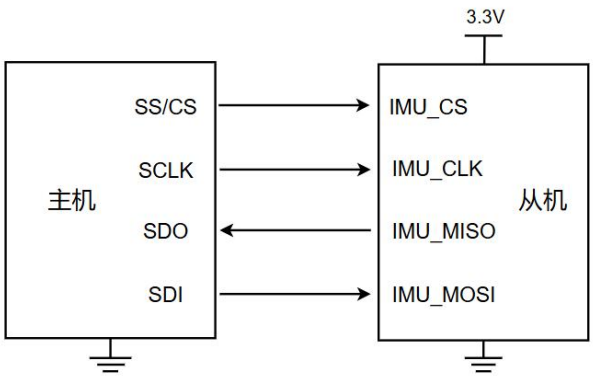
5.2.1 SPI 接口参数

表 29 SPI 接口参数

SPI 主机	本产品作为从机
SPI 速率	0.2~10MHz
SPI 字长	16bit
相位	上升沿触发（模式 3，CPHA=1）
极性	空闲为高电平（模式 3，CPOL=1）
位序	MSB 优先

5.2.2 SPI 连接示意图

图 11 SPI 连线示意图



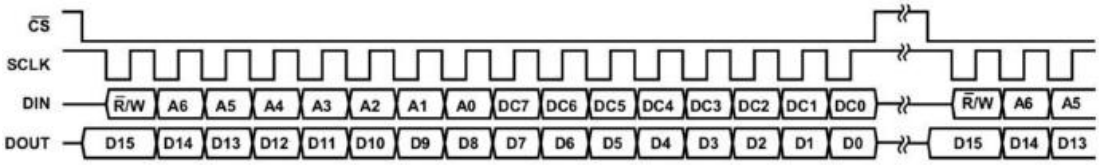
注 1：初始化读取前，需等待 1-3s，使得 IMU 进入正常工作状态。

注 2：不同 IMU 型号的 SPI 引脚参考对应手册

5.2.3 SPI 通信位序

SPI 接口支持全双工串行通信（同时执行发送和接收），采用下图所示的位序。

图 12 SPI 通信位序示意图



其中，DIN 最高位表示读/写操作，[A6:A0]表示寄存器地址，[DC7:DC0]表示写入的数据（写操作）或 DUMMY 数据（读操作）。

当$\overline{R}$/W =1 时，此SPI 周期的DOUT数据无意义。当$\overline{R}$/W =0 时，此SPI 周期的DOUT数据表示上一个周期的寄存器输出数据，具体见BURST读取示例。

5.2.4 SPI 寄存器

表 30 SPI 寄存器列表

名称	地址	读/写	默认值	描述
BURST	0x00	RW		连续读取
PROD_ID1	0x6C	R	0x494d	ID 号 1
PROD_ID2	0x6E	R	0x5532	ID 号 2
PROD_ID3	0x70	R	0x3730	ID 号 3
PROD_ID3	0x72	R	0x5052	ID 号 4(IMU270PR)

5.2.4.1 SPI BURST 寄存器

BURST 为连续读取寄存器，在一个数据流中读取所有数据，各16位段之间无停转。

表 31 SPI BURST 寄存器格式

地址	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	读/写
0x01									RW
地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读/写
0x00	BURST_CMD								RW

BURST 读取方法是：读取前发送 0x8000 表示设置 BURST 并开始读取，然后一直发送 0x0000 并接收数据，输出寄存器内容比读取指令发送偏移 1 个SPI 周期，读取期间一直持续片选低电平。

图 13 SPI BURST 连续读取示意图

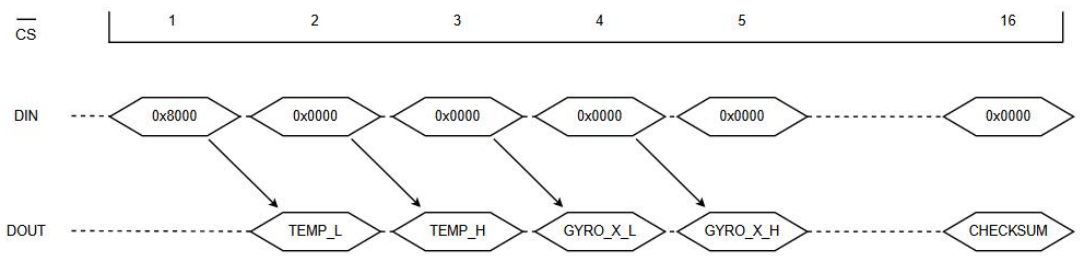


表 32 SPI BURST 连续读取基本格式

发送顺序	1	2	3	4	5	6
发送内容	TEMP_L	TEMP_H	GYRO_X_L	GYRO_X_H	GYRO_Y_L	GYRO_Y_H
发送顺序	7	8	9	10	11	12
发送内容	GYRO_Z_L	GYRO_Z_H	ACCL_X_L	ACCL_X_H	ACCL_Y_L	ACCL_Y_H
发送顺序	13	14	15			
发送内容	ACCL_Z_L	ACCL_Z_H	CHKSM			

注 1：所有数据均为 16-bit 宽度

注 2：温度、陀螺、加速度计数据拼接后格式表示为 float

注 3：所 CHKSM 即 CHECKSUM，用于确认数据完整性。计算方法为将 CHECKSUM 之前的所有数据累加求和

在BURST连续读取过程中，32位的完整数据被拆分成高16位和低16位分别输出，输出时采用小端模式，即低字节先输出。用户需要将这两部分16位数据首尾拼接，还原出完整的32位数据。

图 14 SPI32 位数据还原示意图

按上图拼接得到完整的32位数据后，用户可将32位数据以float来解析，便可获得对应温度或者角速度和加速度信息。

5.2.4.2 SPI ID 寄存器

ID 寄存器为只读寄存器，数据内容为ASCII 编码形式的字符 “IMU”，读取方法类似BURST数据读取：读取时发送 0x6C00~0x7200，并接收数据。输出寄存器内容比读取指令发送偏移 1 个周期。发送指令之间延时需在10us以上。

将 4 个16位ID数据拼接后转为ASICII码，可获得产品的完整ID。拼接方法同BURST连续读取数据的拼接，PROD_ID1在高位，PROD_ID4在低位。

表 36 SPI ID 寄存器格式

地址	bit15 ~ bit0	编码	读/写
0x6C	PROD_ID1	0x494D	R
0x6E	PROD_ID2	0x5532	R
0x70	PROD_ID3	0x3730	R
0x72	PROD_ID4	0x5052	R

6. 注意事项

静电会导致间歇或永久的电路损伤，对电子产品危害很大，经分析多数为 ESD 损坏；

因此，模块的静电防护尤为重要，生产和运输过程需要严格按照静电防护进行作业，须遵循以下条件：

- 严禁裸手接触模块，尤其是引脚位置。
- SMT 贴片机、作业工作台、电烙铁等设备需接地。
- 作业人员佩戴具有良好接地线的人体防静电手环（不可使用无绳静电手环，建议戴防静电手套）。
- 包装和 PCB 必须是合格的防静电材料。

7. 包装

IMU270 PRO 模块采用转运托盘包装。

8. CRC 查表法计算

```
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, 0x6dde4eb, 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, 0xbfd06116, 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,
0x8bbeb8ea, 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,
0x616bffd3, 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,
};
/**
* crc32 计算函数
    uint32_t crc：计算时，初始值使用 1
*/
static 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;
}
```

9. errcode 错误码表

错误码	描述
1	保存参数至 flash 失败
2	读取 Flash 参数失败
3	IMU240 初始化失败
4	IMU4311 初始化失败
5	采样定时器启动失败
6	串口输出线程启动失败
7	数据命令解析错误
8	数据包 CRC 校验失败
9	设置未定义波特率
10	设置未定义输出频率
11	设置未定义模式
12	参数数据长度异常
13	设置 OTA 模式失败
14	未定义命令 ID
15	波特率与输出频率不匹配（波特率与频率参照参考频率与波特率对照表）
16	设置未定义坐标系朝向

10. 频率与波特率对照表

波特率	频率
115200	50

230400	50/100
460800	50/100/250
921600	50/100/250/500
1500000	50/100/250/500

11. 坐标系朝向对应表

朝向（value）	XAxis	YAXis	ZAxis	说明
101	+Ux	+Uy	+Uz	默认朝向
102	-Ux	-Uy	+Uz	
103	-Uy	+Ux	+Uz	
104	+Uy	-Ux	+Uz	
105	-Ux	+Uy	-Uz	
106	+Ux	-Uy	-Uz	
107	+Uy	+Ux	-Uz	
108	-Uy	-Ux	-Uz	
109	-Uz	+Uy	+Ux	
110	+Uz	-Uy	+Ux	
111	+Uy	+Uz	+Ux	
112	-Uy	-Uz	+Ux	
113	+Uz	+Uy	-Ux	
114	-Uz	-Uy	-Ux	
115	-Uy	+Uz	-Ux	
116	+Uy	-Uz	-Ux	
117	-Ux	+Uz	+Uy	
118	+Ux	-Uz	+Uy	
119	+Uz	+Ux	+Uy	
120	-Uz	-Ux	+Uy	
121	+Ux	+Uz	-Uy	

122	-Ux	-Uz	-Uy	
123	-Uz	+Ux	-Uy	
124	+Uz	-Ux	-Uy	