



4 通道振弦式采集仪

用
户
手
册

(V1.01)

深圳安锐科技有限公司

2024 年 11 月

免责声明与警告

使用本产品之前，请仔细阅读本文档。本声明对安全使用本产品以及您的合法权益有着重要影响。

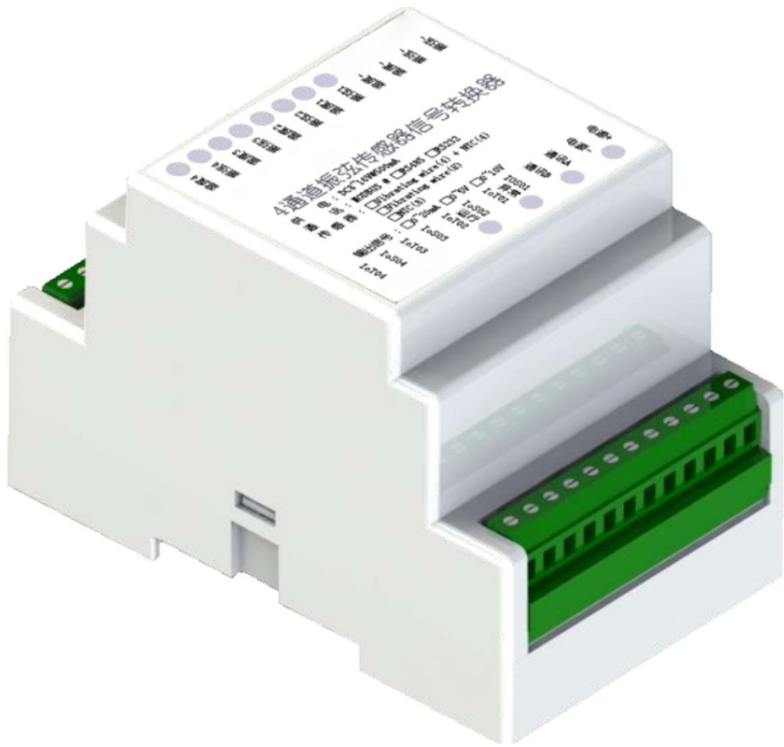
作为生产企业，我们生产的监测仪质量安全可靠，但是希望购买者要充分了解传感、电子、自动化测控、计算机等技术的相关知识。

我们不承诺它能兼容所有传感器以及满足您的所有使用要求和期望。

本手册不直接或间接构成任何承诺，在购买者自愿购买的基础上，如仪器在使用过程中对人体造成的伤害或经济损失，作为生产厂家我们不承担任何经济责任及法律责任。

概述

安锐测控振弦采集仪是轨道安装式振弦传感器信号转换器，可将振弦、温度传感器信号转换为RS485 数字信号和模拟信号输出，方便的接入已有监测系统。传感器状态专用指示灯方便现场安装调试。振弦采集仪 内部参数丰富、激励方法多样，可兼容国内外绝大多数振弦传感器。标准 DIN 导轨设计，非常适用于安装到标准仪表箱内。



4 通道振弦传感器信号转换器

主要特性

- 尺寸： 87.5x49.0x58.7mm
- 安装方式： 标准 TS35 导轨
- 端子类型： 3.81 螺丝固定式
- 供电： DC10~16V@500mA
- 功耗： 空载 40mA@DC12V
- 传感器接口
 - 1 频率+1 温度 4 频率+4 温度 ➢ 8 频率 ➢ 8 温度
- 数字接口
 - UART_RS485 ➢ UART_TTL
- 通讯协议： MODBUS
- 模拟信号输出
 - 0~20mA ➢ 0~5V ➢ 0~10V
- 模拟信号分辨率： 1/4095
- 工作温度： -25℃~85℃

主要性能指标

参数	条件	最小值	典型值	最大值	单位
电源					
VIN		10	12（推荐）	16	V
Iwork					mA
Isleep			40		mA
数字接口 RS232/RS485					
通讯速率		9600	9600	460800	bps
振弦（频率）传感器					
测评范围		100		6000	Hz
激励电压	低压扫频	定制	9.3	定制	V
	高压脉冲	----	150	200	V
频率分辨率			0.1		Hz
频率误差		0.01		0.05	Hz
温度（热敏电阻）传感器					
电阻范围		1	3	10	K Ω
测温范围	@NTC3K	-25		200	℃
温度分辨率			0.1		℃
温度误差			0.3		℃
其它					
RTC 精度			2	3.5	ppm
工作温度		-40		+85	℃

产品选型

VT I/V A B C

- VT：振弦传感器信号转换器
- I/V：模拟信号电流/电压
- A：内嵌核心设备数量（1 个）
- B：两位数字表示的振弦通道数量（04、08）
- C：固定为 DIN，表示安装方式为 DIN 导轨

型号	振弦	温度	模拟量 电压	激励方法		备注说明
				高压脉冲	低压扫频	
01	1	1		●	●	
04	4	4		●	●	
08	8				●	
04			4			

目录

目录

概述.....

3

主要特性.....

3

主要性能指标.....

4

产品选型.....

4

目录.....

5

端子定义说明.....

6

外形尺寸.....

6

指示灯与按键.....

6

注意事项.....

7

一、通讯协议.....

8

寄存器机制.....

8

数据模式.....

8

通讯协议.....

8

1.1MODBUS 通讯协议.....

8

1.2AABB 通讯协议.....

9

1.3\$字符串通讯协议.....

10

1.4 数据帧校验算法.....

11

1.5 寄存器（参数）汇总表.....

12

1.6 模块地址操作（ UART）.....

14

1.7 通讯速率和软件握手（ UART）.....

14

1.7.1 通讯速率.....

14

1.7.2 软件握手.....

15

1.7.4 读取未知设备地址的通讯速率.....

15

1.7.5 修改未知设备地址的通讯速率.....

15

1.8 测量模式.....

15

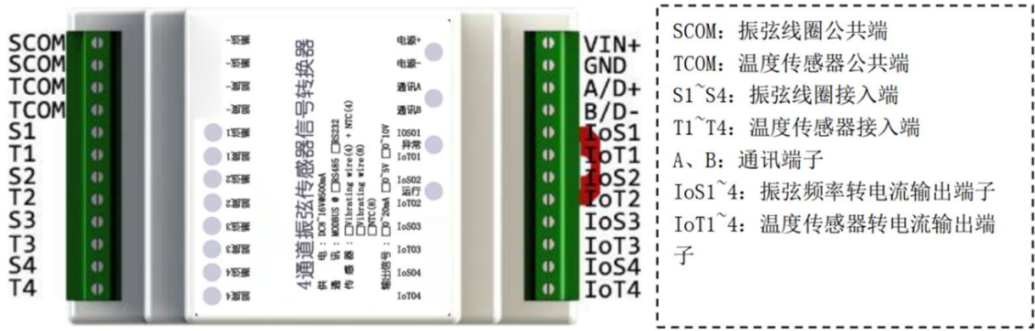
1.8.1 连续测量模式.....

16

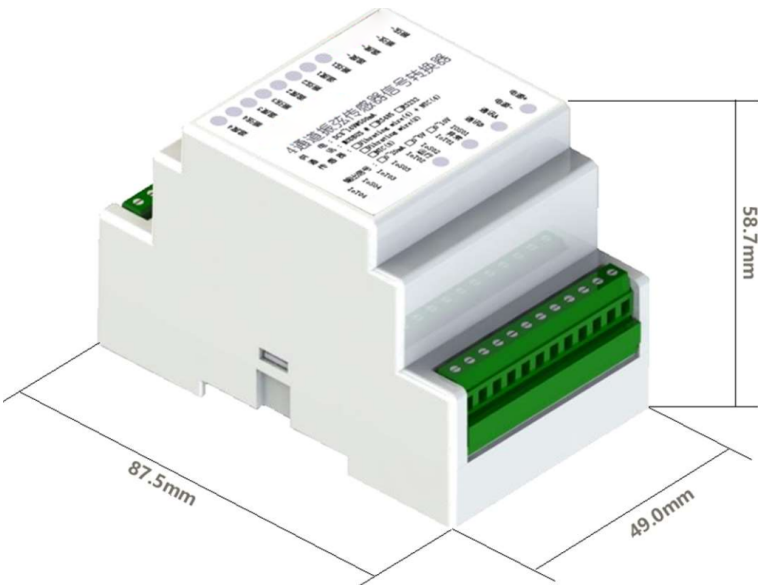
1.8.2 单次测量模式.....

16

端子定义说明



外形尺寸



指示灯与按键

指示灯名称	指示灯状态	指示灯功能描述
电源指示灯 (红色)	长亮	设备供电正常
	长熄	设备未供电或者供电异常
通讯指示灯 (红色)	快速闪烁	通讯接口有数据正在传输
	熄灭	通讯线上无数据
运行指示灯 (绿色)	闪烁 (1Hz)	运行正常
	常亮或者常熄	运行异常
异常指示	熄灭	
	常亮	工作异常
振弦指示灯 V1~V4	熄灭	未检测到有效的传感器线圈接入
	快速闪烁	已检测到振弦传感器线圈, 正在测量频率值 (频率值尚不稳定)
	常亮	振弦传感器频率测量已经完成, 信号稳定
温度指示灯 T1~T4	熄灭	未检测到有效温度传感器接入
	常亮	已检测到温度传感器接入, 温度测量正常

注意事项

✓ 严格接地

设备电源负极应与大地可靠连接，否则可能导致信号采集噪声过大或触电危险。

✓ 防止静电

静电可能使仪器组件和配件严重受损甚至无法使用。

✓ 不要超压使用

仅使用产品制造商提供的电源线和电源适配器。

✓ 不要超出驱动能力

严格按照本手册说明的电源输出和模拟量输出指标外接其它设备，不要超负荷使用。

✓ 轻拿轻放

使用和运输过程中应防止强震动，不要带电安装和转移设备。

✓ 避免挤压

设备外壳无承重能力，挤压变形会导致金属外壳与内部电子元件短路。

✓ 严禁水淋

本产品不具备防水能力，严禁以任何形式与水接触，不要安装于露天环境。

✓ 干燥环境存放

长时间放置于潮湿环境中会引起外壳和内部电子部件锈蚀，连接电源前应保证设备外部及内部干燥。

✓ 注意量程

不要试图连接超过测量量程范围的信号线到输入端子。

✓ 不要带电操作

严禁设备工作期间插拔、操作连接端子。

✓ 及时更换钮扣电池

内部时钟电池电量低会导致设备无法正常工作，建议每一到两年更换一次。

一、通讯协议

寄存器机制

振弦采集仪内部维护有若干寄存器，在寄存器参数值的控制下完成工作。寄存器的值总是以整数形式存在，基本操作单位为“字”（2字节整数，大端模式）。通过数字接口可完成对寄存器的读取和写入（修改）操作。寄存器写（修改）操作后可保存于内部存储器，掉电不遗失。

数据模式

寄存器数据值采用大端模式，数据的高字节保存在内存的低地址中，而数据的低字节保存在内存的高地址中，数据帧传输时先传输低地址字节后传输高字节。每个寄存器对应两个字节，则单个寄存器的值=低字节值*256+高字节值。若无特殊说明，本手册后续章节所述寄存器（或“参数”）均是指两字节构成的16bit数据。

通讯协议

本设备支持标准的工业 MODBUS 通讯协议（03、04、06 指令码）和自定义的简单 AABB 协议以及字符串指令集三种协议。MODBUS 和 AABB 通讯协议支持基于设备地址和总线连接的一主多从应用结构，在总线中振弦采集仪始终作为从机使用。

1.1 MODBUS 通讯协议

在 MODBUS 协议下，所有寄存器被定义为“保持寄存器”（详见 MODBUS 通讯协议标准说明），设备支持基于 MODBUS 协议的多个连续寄存器读取、单个寄存器写入两种指令码，对应指令码分别为 0x03、0x06。

（1）03（0x03）、04（0x04）指令码：读取多个连续的寄存器数据，指令格式如下

指令数据帧结构

地址码	功能码 0x03	开始地址	寄存器数量	CRC 校验
1 字节	1 字节	2 字节	2 字节	2 字节

返回数据帧结构

地址码	功能码 0x03	数据长度	数据	CRC 校验
1 字节	1 字节	2 字节	n 字节	2 字节

例：读取地址为 0x01 的设备寄存器值，寄存器开始地址为 0，连续读取 10 个寄存器

主机发送指令：0x01 0x03 0x00 0x00 0x00 0x0A **0xC5 0xCD**

从机返回应答：0x01 0x03 0x14 0x00 0x01 0x00 0x60 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x01 0x01 0xF4 0x00 0x00 0x00 0x64 0x00 0xC8 **0x5F 0x8F**（下划线为读取到的 10 个寄存器值）

读取多个连续寄存器时，单次读取不要超过 32 个寄存器，不要试图读取不存在的寄存器。

(2) 06 (0x06) 指令码：修改单个寄存器的值，指令格式如下

指令数据帧结构

地址码	功能码 0x06	寄存器地址	寄存器值	CRC 校验
1 字节	1 字节	2 字节	2 字节	2 字节

返回数据帧结构

地址码	功能码 0x06	寄存器地址	寄存器值	CRC 校验
1 字节	1 字节	2 字节	2 字节	2 字节

例：将地址为 0x01 的设备中的寄存器 8 的值修改为 100

主机发送指令：0x01 0x06 0x00 0x08 0x00 0x64 **0x09 0xE3**

从机返回应答：0x01 0x06 0x00 0x08 0x00 0x64 **0x09 0xE3**

1. 2AABB 通讯协议

AABB 通讯协议是一种非标准自定义协议，相较于 MODBUS 通讯协议，结构更简单，指令生成方法更容易，便于进行快速测试。AABB 通讯协议支持单寄存器读写两种指令。

(1) 读取单个寄存器

指令数据帧结构

指令头 0xAA 0xBB	地址码	寄存器地址	和校验
2 字节	1 字节	1 字节	1 字节

返回数据帧结构

指令头 0xAA 0xBB	地址码	寄存器地址	寄存器值	和校验
2 字节	1 字节	1 字节	2 字节	1 字节

指令头：固定为 16 进制 AABB

地址码：设备的地址（1~255，其中地址 255 为通用地址）

寄存器地址：要访问的寄存器地址，寄存器地址字节最高位是读写标志位，为 0 时表示读寄存器，为 1 时表示写寄存器。

和校验：之前所有数据之和，0xAA+0xBB+地址码+寄存器地址，校验和超过 255 时，仅使用低字节。如下例中，校验和=0xAA+0xBB+0x01+0x08=0x016E，则只使用 0x6E 作为最终和校验码。

例：读取地址为 0x01 的设备寄存器值，寄存器地址为 8

主机发送指令：0xAA 0xBB 0x01 0x08 0x6E

从机返回应答：0xAA 0xBB 0x01 0x08 0x00 0x60 0xCE

(2) 修改单个寄存器

指令数据帧结构

指令头 0xAA 0xBB	地址码	寄存器地址 0x80	寄存器值	和校验
2 字节	1 字节	1 字节	2 字节	1 字节

写寄存器指令中，寄存器地址字节的最高位应为 1，即地址值与 0x80 做“或”运算。

返回数据帧结构

指令头 0xAA 0xBB	地址码	寄存器地址	寄存器值	和校验
2 字节	1 字节	1 字节	2 字节	1 字节

例：修改地址为 0x01 的设备寄存器值，寄存器地址为 8，修改值为 100

主机发送指令：0xAA 0xBB 0x01 0x88 0x00 0x64 **0x52**

从机返回应答：0xAA 0xBB 0x01 0x08 0x00 0x64 **0xD2**

(3) 通用设备地址

AABB 通讯协议支持通用地址，无论设备的当前地址为何值，使用 0xFF 作为地址对其发送读写指令，均可得到正确响应。

例：使用通用地址，读取任一设备的寄存器 8

主机发送指令：0xAA 0xBB 0xFF 0x08 **0x6C**

从机返回应答：0xAA 0xBB 0x01 0x08 0x00 0xC8 **0x36**

注：当总线上连接有多个从机设备时（通常为 RS485 总线），使用通用地址时总线上所有设备均会响应指令，导致指令无法正常使用。

注：严禁在连接有多个从设备的总线中使用通用地址修改设备地址。

(4) 特殊设备地址

设备地址保存于寄存器 ADDR. [7:0]，取值范围为 1~255，这些地址中，255 在 AABB 协议中作为通用地址使用。

设备地址寄存器（0x00）

位	符号	值	描述	复位值
bit15:8			暂未定义	0
bit7:0		1~254	设备地址	1

1.3\$字符串通讯协议

字符串通讯协议是自定义的一种以英文字符‘\$’为固定帧头的字符串通讯协议，可对单个寄存器（参数）进行读写操作，方便进行快速、简便的测试。

(1) 读取单个寄存器

指令数据帧结构

帧头\$GETP	固定分隔符 '='	寄存器地址	帧尾\r\n (回车)
5 字节	1 字节	1~2 字节	2 字节

返回数据帧结构

帧头\$REG	寄存器地址 起始符[	寄存器地址	寄存器地址 终止符]	固定分隔符 '='	寄存器值	帧尾 \r\n
4 字节	1 字节	1~2 字节	1 字节	1 字节	1~5 字节	2 字节

例：读取地址为 0x01 的设备寄存器值，寄存器地址为 21

主机发送指令：\$GETP=21\r\n

从机返回应答：\$REG[21]=96\r\n

(2) 修改单个寄存器

指令数据帧结构

帧头 \$GETP	固定分隔符 '='	寄存器地址	固定分隔符 '，'	寄存器值	帧尾\r\n (回车)
5 字节	1 字节	1~2 字节	1 字节	1~5 字节	2 字节

返回数据帧结构：OK\r\n

例：修改地址为 0x01 的设备的 21 寄存器的值为 1152

主机发送指令：\$SETP=21,1152\r\n

从机返回应答：OK\r\n

注意：使用字符串指令对参数进行修改后，需要使用\$SAVE 进行保存。

1.4 数据帧校验算法

无论是向设备发送指令还是接收设备返回的答应数据，均应严格进行数据校验。极少情况下，设备返回的应答数据会存在错误，通过数据帧的校验码验证可完全避免读取到错误的数据。

(1) CRC16-MODBUS 算法

```
unsigned int crc16(unsigned char *dat, unsigned int len)
{
    unsigned int crc=0xffff;
    unsigned char i;
    while(len!=0)
    {
        crc^=*dat;
        for(i=0;i<8;i++)
        {
            if((crc&0x0001)==0)
                crc=crc>>1;
            else
            {
                crc=crc>>1;
                crc^=0xa001;
            }
        }
    }
}
```

```

    }
    len-=1;
    dat++;
}
return crc;
}

```

(2) 和校验算法

```

unsigned char AddCheck(unsigned char *dat,unsigned char count)
{
    unsigned char i,Add=0;
    for (i=0;i<count;i++)
        Add+=dat[i];
    return Add;
}

```

1.5 寄存器（参数）汇总表

振弦采集仪寄存器汇总表

地址	符号	读写	名称	默认值	单位
0X00(0)	ADDR	读/写/存	模块地址（UART）	0x0001	
0X01(1)	BAUD ^①	读/写/存	通讯速率（UART）	0x0060	100bps
0X02(2)	AUX	读/写/存	辅助功能寄存器	0x0018	
0X03(3)	SYS_FUN	读/写/存	系统功能寄存器	0x0000	
0X04(4)	预留		暂未定义功能	0x0000	
0X05(5)	WKMOD	读/写/存	工作模式	0x0001	
0X06(6)	MM_INTE	读/写/存	连续测量时间间隔	0x01F4	ms
0X07(7)	ATSD_SEL	读/写/存	自动上传	0x0000	
0X08(8)	RD_INTE	读/写/存	延时采样	0x0064	ms
0X09(9)	RE_COUNT	读/写/存	期望采样数量	0x14C8	个
0X0A(10)	EX_METH	读/写/存	激励方法	0x0064	
0X0B(11)	预留		暂未定义功能	0x0000	
0X0C(12)	预留		暂未定义功能	0x0000	
0X0D(13)	HP_DUR	读/写/存	泵压时长	0x03E8	ms
0X0E(14)	HP_EXP	读/写/存	期望电压	0x8096	V
0X0F(15)	FS_FMIN	读/写/存	扫频下限	0x012C	Hz
0X10(16)	FS_FMAX	读/写/存	扫频上限	0x1388	Hz
0X11(17)	FS_STEP	读/写/存	扫频步进	0x0005	Hz
0X12(18)	FS_SCNT	读/写/存	扫频信号周期	0xC80A	个
0X13(19)	FIT_TYPE	读/写/存	软件滤波方法	0x0000	
0X14(20)	FIT_COUNT	读/写/存	滤波样本数量	0x000A	个
0X15(21)	CAL_PAR1	读/写/存	粗差剔除参数因子	0x0014	
0X16(22)	CAL_PAR2	读/写/存	优质样本数限制因子	0x0004	
0X17(23)	AMP ^①	读/写/存	信号放大倍数	0x0001	

0X18(24)	FSG_TH	读/写/存	反馈渐进频率上下限	0x1414	Hz
0X19(25)	DAO_TH	读/写/存	模拟输出频率上下限	0x2100	百 Hz
0X1A(26)	TEMP_PAR1 ^①	读/写/存	温度计算参数 1	0x0F6E	
0X1B(27)	TEMP_PAR2 ^①	读/写/存	温度计算参数 2	0x0064	0.01
0X1C(28)	TEMP_EX ^①	读/写/存	温度传感器设置	0x0202	
0X1D(29)	EXS_TH	读/写/存	信号质量限值	0x0046	%
0X1E(30)	SIG_TH	读/写/存	信号幅值上下限	0x6400	% %
0X1F(31)	CRC	只读	参数 CRC 校验		
0X20(32)	SYS_STA	读/写/存	系统状态寄存器	0	
0X21(33)	SFV	只读	当前扫频频率		Hz
0X22(34)	SMP_QUA	只读	优质样本质量评定值		%
0X23(35)	S_FRQ	只读	传感器频率值		0.1Hz
0X24(36)	F_REQM	只读	频率模数高 16 位		100Hz2
0X25(37)		只读	频率模数低 16 位		
0X26(38)	预留	只读	暂未定义功能		
0X27(39)	S_RES	只读	线圈电阻		Ω
0X28(40)	V_SEN	只读	实时激励电压		0.01V
0X29(41)	TEMP	只读	温度值		0.1℃
0X2A(42)	SMP_STD	只读	样本标准差		Hz
0X2B(43)	BQ_COUNT	只读	“优质”样本数量		个
0X2C(44)	SIG_VAL	只读	信号幅值		% % % %
0X2D(45)		只读			
0X2E(46)	V_POW	只读	VDD 电压值		Mv
0X2F(47)	NOISE_FRO	只读	底噪频率		0.1Hz
0X30(48)	NOISE_AMP	只读	噪声强度		dBm
0X31(49)	CH_STA	只读	多通道测量模块专用通道状态和测量完成的频率值（温度值） ^③		
0X32(50)	CH_NUM	只读			
0X33(51)	CH01	只读			
0X34(52)	CH02	只读			
0X35(53)	CH03	只读			
0X36(54)	CH04	只读			
0X37(55)	CH05	只读			
0X38(56)	CH06	只读			
0X39(57)	CH07	只读			
0X3A(58)	CH08	只读			

读：可读取；写：可修改；存：掉电不遗失；复：上电后复位为默认值不要修改预留寄存器（位）的上电初始值。①：这些寄存器（参数）在下次启动时生效 ②：实际值为 0x8082，即：启用期望电压功能，期望电压为 120V ③：寄存器 CH_STA. [7:0]1~8 通道线圈接入状态，CH_STA. [15:8]表示每通道的频率值是否来自 SFC 估值。 ④：高 8 位字节为“优质样本占全部样本的百分比值”，低 8 位字节为“优质样本评估质量”。

1.6 模块地址操作（UART）

将地址为 0x01 的模块地址修改为 0x02

1.6.1 修改已知设备地址的地址

模块地址 0x01

MODBUS 指令： 0x01 0x06 0x00 0x00 0x00 0x02 **0x08 0x0B**

AABB 指令： 0xAA 0xBB 0x01 0x80 0x00 0x02 **0xE8**

1.6.2 读取未知设备地址的地址

未知地址的模块仅可使用 AABB 协议的通用地址 0xFF

AABB 指令： 0xAA 0xBB 0xFF 0x00 **0x64**

1.6.3 修改未知设备地址的地址

AABB 指令： 0xAA 0xBB 0xFF 0x80 0x00 0x02 **0xE6**

修改模块地址后，新的地址立即生效，收到的应答信息中的模块地址码部分是新的地址值。后续的指令应使用新的地址对模块进行操作。

1.7 通讯速率和软件握手（UART）

1.7.1 通讯速率

VMXXX 的 UART 接口支持 9600~460800bps 通讯速率，通过设置寄存器 BAUD. [13:0] 来改变通讯速率，BAUD. [13:0] 的单位为“每秒百位”或“百 bps”。寄存器取值与对应的通讯速率如下表。

UART 通讯速率寄存器 BAUD（0x01）

位	符号	值	描述			默认值
bit15			是否启用软件握手协议功能			0
bit14			是否忽略“测量正忙”标志而立即响应指令			0
bit13:0		值	速率	值	速率	96
		96	9600bps（默认）	768	76800bps（NR）	
		128	12800bps	1152	115200bps	
		144	14400bps	1280	128000bps	
		192	19200bps	1536	153600bps	
		288	28800bps（NR）	2304	230400bps	
		384	38400bps（NR）	2560	256000bps	
		560	56000bps（NR）	4608	460800bps	
		576	57600bps（NR）			
通讯速率在下次启动时生效，非法的通讯速率值会引起参数恢复出厂值。NR 为不推荐						

在条件允许情况下，尽量使用较高的通讯速率，缩短数据传输时间。

注：关于 UART 通讯的其它参数（校验位、数据位、停止位），详见 3.22 辅助功能寄存器说明。

1.7.2 软件握手

模块开始一次测量时，从 UART 接口主动发送 XOFF 信号（0x13），表示模块开始忙于测量数据，当测量完成时主动发送 XON 信号（0x11），表示模块本次测量完成，正处于空闲状态。

在开启模块的软件握手功能后，若需要向模块发送指令，建议 UART 的通讯流程为：首先等待模块返回 XON 信号（0x11），当收到 XON 信号或等待超时后立即向模块发送指令。

注：在一主多从的总线应用中，严禁开启模块的软件握手功能。

1.7.3 修改已知设备地址的通讯速率

模块地址 0x01，将波特率修改为 115200bps

MODBUS 指令：0x01 0x06 0x00 0x01 0x04 0x80 **0xDB 0x6A**

AABB 指令：0xAA 0xBB 0x01 0x81 0x04 0x80 **0x6B**

1.7.4 读取未知设备地址的通讯速率

未知地址的模块仅可使用 AABB 协议的通用地址 0xFF

AABB 指令：0xAA 0xBB 0xFF 0x01 **0x65**

模块返回：0xAA 0xBB 0x01 0x01 0x00 0x60 **0xC7**

1.7.5 修改未知设备地址的通讯速率

AABB 指令：0xAA 0xBB 0xFF 0x81 0x04 **0x80 0x69**

修改 UART 通讯速率后，新的值会在下次启动时生效，未重启之前模块会继续使用修改前的通讯速率。重启后应使用新的通讯速率与模块进行通讯。

除上述模块地址、通讯速率寄存器外，其它所有寄存器的访问方法均相同，不再一一举例。

1.8 测量模式

模块有连续测量和单次测量两种测量模式，通过向测量模式寄存器 WKMOD.[0]写入 1 使模块工作于连续测量工作模式，写入 0 使模块工作于单次测量工作模式。WKMOD.[15]用来设置是、否在模块“忙”时禁用数字接口，当数字接口被禁用期间，模块不会收到任何经由数字接口传、输的数据或指令，当数字接口不被禁用时，模块内部维持传感器测量优先的逻辑，收到的指令、会在模块完成当次测量后得到响应。

工作模式寄存器 WKMOD（0x05）

位	符号	值	描述	默认值
bit15		数字接口工作策略		0
		0	从不关闭数字接口	
		1	忙时关闭数字接口	
bit14 ^①		0	修改参数时自动保存	0
		1	修改参数时不保存（下次上电恢复）	
bit13		0	是否按照通道顺序依次回应实时数据	0
bit12		0	是否测量电子标签	0
bit11:4		0	预留	0
bit3:1		频模寄存器显示内容设置		0
		0	显示频率模数	
		1	显示高分辨率频率值，分辨率：0.01Hz	
		2~7	暂未定义功能	
bit0		工作模式设置		1
		0	将模块切换至单次测量模式	
		1	将模块切换至连续测量模式	

注 1：当需要频繁的修改模块参数时，建议此位设置为 1，以减少写操作对 EEPROM 寿命的影响。

1.8.1 连续测量模式

在连续测量模式下，模块每间隔一定的时间自动进行一次传感器激励和数据读取操作，相邻两次测量的时间间隔由寄存器 MM_INTE 进行设置，单位为毫秒。连续测量时间间隔寄存器 MM_INTE（0x06）

位	符号	值	描述	默认值
bit15:0		5~60000	连续测量时间间隔，单位：mS	500
		60001~65535	时间间隔=(此值-60000)*60000mS	

无论是连续测量模式还是单次测量模式，在每次向传感器发送激励信号之前均会等待 MM_INTE 毫秒，所以，连续测量时间间隔也称为“激励前等待时长”。

1.8.2 单次测量模式

单次测量是指模块一直处于“空闲”状态，当收到单次测量指令后，立即启动单次测量过程的工作流程。在单次测量模式，每隔 3 秒在 RTS 管脚输出一短促高脉冲（10ms）表示当前处于单次工作模式。有四类指令可以触发一次单次测量。

（1）使用专用\$指令集指令

通过 UART 接口向模块发送单次测量指令 \$MSFR=x 或者 \$MSFT=x。

例：在单次测量模式下向模块发送单次测量指令：\$MSFR=3\r\n 模块开始完成 3 次测量，当完成后输出频率测量结果数据：\$FR=1343.3Hz\r\n 向模块发送单次测量指令：\$MSFT=3\r\n 模块开始完成 3 次测量，当完成后输出频率和温度测量结果数据：\$FR=1343.3Hz\t\$TE=30.2'C\r\n

(2) 使用专用单次测量指令 0xAAAA 或 0xAAAB

通过 UART 接口向模块发送特定的单次测量指令，模块测量完成后主动上传频率和温度测量结果，指令格式为：

指令数据帧结构

指令头 0xAA 0xA?	地址码	功能码	和校验
2 字节	1 字节	1 字节	1 字节

返回数据帧结构

指令头 0xAA 0xA?	地址码	功能码	频率值/温度值	和校验
2 字节	1 字节	1 字节	2/4 字节	1 字节

指令头：固定为 16 进制 AAAA 或者 AAAB 两个字节，AAAA 指令仅返回频率值，AAAB 指令返回频率和温度值。

功能码：0x1x、0x3x 或 0x7x，末尾“x”表示进行几次激励和读数操作。0x1x 功能码表示直接进行 x 次测量读数，0x3x 表示测量前清除历史数据，0x7x 表示测量过程中检测到传感器信号质量合格或者达到 x 次立即停止测量过程。

当使用 AAAA 指令进行单次频率测量时，在指定的 x 次测量完成后会主动输出频率值（上述“返回数据帧”），若当前设置了滤波功能则输出频率值为 x 次的滤波值，若未设置滤波功能，则输出频率值为最后一次测量的实时频率值。频率值用 2 个字节表示，高字节在前，单位为 0.1Hz。当使用 AAAB 指令进行单次频率测量时，在指定的 x 次测量完成后会主动输出频率值和温度值，若当前设置了滤波功能则输出值为 x 次的滤波值，若未设置滤波功能，则输出频率和温度值为最后一次测量的实时值。频率值用 2 个字节表示、温度值用 2 个字节表示。

单次测量时，测量次数尽量>=3 次，且推荐使用高压激励方法、开启历史数据滤波功能。若采用扫频法，可能首次测量失败，影响最终结果计算（若使用了历史数据滤波功能）。

例：在单次测量模式下

向模块发送单次测量指令：AA AA 01 13 68

模块开始完成 3 次测量，当完成后输出频率测量结果数据：

单通道模块返回：AA AA 01 13 34 3A D6，频率值 $(0x34*256+0x3A)/10=1337.0\text{Hz}$ 。

4 通道模块返回：

8 通道模块返回：

向模块发送单次测量指令：AA AB 01 13 69

模块开始完成 3 次测量，当完成后输出频率和温度测量结果数据：

单通道模块返回：AA AB 01 13 34 3A 00 F5 CC，频率值 1337.0Hz，温度值 24.5℃。

4 通道模块返回：

8 通道模块返回：

(3) 向系统功能寄存器写入单次测量指令码

使用数字接口，任意一种通讯协议，向系统功能寄存器 SYS_FUN 写入 0x1x、0x3x

或者 0x7x 指令码，触发一次单次测量。使用这种方法时，模块遵循主从通讯机制，在完成测量后不会主动上传数据，可以通过读取系统状态寄存器 SYS_STA. [4]来判断当前是否已经完成了本次单次测量，并在检测到完成时读取频率寄存器 S_FRQ 获得本次测量的频率结果。

例：在单次测量模式下使用 MODBUS 协议

向模块发送单次测量指令： 01 06 00 03 00 13 38 07

模块返回 MODBUS 应答数据： 01 06 00 03 00 13 38 07

模块开始完成 3 次测量，当完成后不输出任何信息，需要读取 SYS_STA. [4]来判断测量是否完成。

（4）直接读取频率寄存器 S_FRQ

在单次测量模式下使用 MODBUS 或 AABB 通讯协议读取频率寄存器 S_FRQ， VM 模块执行 0x73 指令码，并根据使用的通讯协议返回寄存器值。

单次测量指令举例（模块地址为 0x01）

单次测量指令	模块应答	备注说明
AA AA 01 13 68	AA AA 01 13 <i>35 B3</i> 50	进行 3 次测量并返回频率值
AA AA 01 33 88	AA AA 01 33 <i>35 B4</i> 71	清空历史频率值后进行 3 次测量并
		返回频率值。
AA AA 01 73 C8	AA AA 01 73 <i>35 B4</i> B1	检测到合格的返回信号后立即停止测量并返回频率值（最多测量 3 次）。
01 06 00 03 00 13 38 07	01 06 00 03 00 13 38 07 ^①	进行 3 次测量并将频率值保存于频率寄存器。
01 06 00 03 00 33 39 DF	01 06 00 03 00 33 39 DF ^①	清空历史频率值后进行 3 次测量并将频率值保存于频率寄存器。
01 06 00 03 00 73 38 2F	01 06 00 03 00 73 38 2F ^①	检测到合格的返回信号后立即停止测量并将频率值保存于频率寄存器（最多测量 3 次）。
AA BB 01 83 00 13 FC	AA BB 01 03 00 13 7C ^①	
AA BB 01 83 00 33 1C	AA BB 01 03 00 33 9C ^①	
AA BB 01 83 00 73 5C	AA BB 01 03 00 73 DC ^①	
01 03 00 23 00 01 75 C0	01 03 02 <i>35 B0</i> AE A0	检测到合格的返回信号后立即停止测量并返回频率值（最多测量 3 次）。
AA BB 01 23 89	AA BB 01 23 <i>35 B0</i> 6E	
注 1：此条数据是模块对修改系统功能寄存器指令的标准应答（非频率值）。 斜体字为模块返回的频率值。		

注：其它具体参数解释说明可参照“振弦传感器读数模块用户”



深圳安锐科技有限公司

通讯地址：广东省深圳市龙华区泰宇丰工业园D栋3楼

联系电话：0755-23776237

官方网址：www.aiterich.com

邮箱：kerich@aiterich.com