SJ-101T

数字称重变送模块

使用说明书

V1.01

一、主要参数

1.	型号：	SJ-101T
2.	准确度等级:	3 级
3.	采样速度：	10 次
4.	分度值：	1/2/5
5.	传感器灵敏度范围：	1.5～3mV / V
6.	传感器输入电压范围：	-15mV~+15mV
7.	传感器激励电压：	3.3V
8.	输入电源电压：	DC 9-24V
9.	串行通讯接口：	RS485 信号传输距离≤1000 米
10.	通信方式：	标准 MODBUS-RTU 协议
11.	使用温度	0-40℃
12.	储运温度：	-25～55℃

二、简介说明

1、尺寸

三、模块说明

1、接口定义

1、电源输入：

模块支持 9-24V 宽电压输入。（注意电源正负极性）

2、传感器接口

传感器采用四线屏蔽线接口。无长线补偿。

E(+)：传感器电源正(红)　　E(-)：传感器电源负(黑)

S(+)：传感器信号正(绿)　　S(-)：传感器信号负(白)

四、通信协议说明

1、通信配置

模块的默认配置如下：

波特率：38400　　数据位：8 位

奇偶校验：无　　停止位：1 位　　设备地址：0x01

2、MODBUS-RTU 协议

模块支持 MODBUS-RTU 从机模式。

支持 03，06，16 命令

MODBUS 采用最高频率不超过 10HZ，通信等待时间最小 50ms。

本模块采样 485 通信。

熟悉 Modbus 协议下，读下面内容对编程会有事半功倍的效果。

五、通信协议列表

40001（十进制）是 PLC 保存继存器地址，可使用 03，06，16(十进制)指令，协议地址减去 40001 即可。比如读 40002 地址内容实际发送为 01 03 00 01 00 01 xx xx。

以下灰色部分为称重功能，橙色为通讯设置，黄色部分用于校准功能，蓝色滤波设置。称重配置完成后通常就只使用灰色部分。

MODBUS-RTU 格式说明					
PLC 地址 (十进制)	**协议地址 (十进制)**	**功能**	**说明**		**操 作方式**
40001	00	第一路重量	32 位重量数据高 16 位		读
40002	01		32 位重量数据低 16 位		
……	……	……			
40019	18	第十路重量	32 位重量数据高 16 位		
40020	19		32 位重量数据低 16 位		
40021	20	第 1、2 路状态位 **单寄存器 16 位，可表示 2 路称重状态。每一位 置 1 表示不同的状态** 一共 5 个寄存器（**10 个字节,先读出的是第 2 路，后一个字节为第 1 路**）可表示 10 路状态	bit0	1：重量 1 开机不稳定	读
			Bit1	1：重量 1 当前不稳定	
			Bit2	1：重量 1 超载	
			Bit6	1：重量 1 未就绪	
			Bit7	1：重量 1 无效	
			Bit8	1：重量 2 开机不稳定	
			Bit9	1：重量 2 当前不稳定	
			Bit10	1：重量 2 超载	
			Bit14	1：重量 2 未就绪	
			Bit15	1：重量 2 无效	
……	……	……	……		
40025	24	第 9、10 路状态位	Bit15-bit0 同上		
40026	25	小数点精度如该位为 0x1220,分别为 0.0;0.00;0.00;0	Bit15-12	重量 1 的精度	读
			Bit11-8	重量 2 的精度	
			Bit7-4	重量 3 的精度	
			Bit3-0	重量 4 的精度	
40027	26	小数点精度	Bit15-12	重量 5 的精度	
			Bit11-8	重量 6 的精度	
			Bit7-4	重量 7 的精度	
			Bit3-0	重量 8 的精度	
40028	27	小数点精度	Bit15-12	重量 9 的精度	
			Bit11-8	重量 10 的精度	
40029	28	清零操作可或操作 写后自动清除该	位操作 Bit0 置 1 对秤 1 清零 Bit1 置 1 对秤 2 清零		写

		位	…… Bit9 置 1 对秤 10 置零	
40030	29	开机置零方式	位操作 Bit0 置 1 时秤 1 开机不置零 Bit1 置 1 时秤 2 开机不置零 …… Bit9 置 1 时秤 10 开机不置零 如果十路开机都不置零，则该数据为 0x3ff	写
40031	30	波特率设置	0 9600 1 14400 2 19200 3 38400	
40032	31	校验位设置	0 无 1 奇 2 偶	
40033	32	设备地址（站号）	1-99（0 为广播地址）	
40034	33	通讯数据设置有效	发送 1 将立即生效当前设置的：波特率、校验位、设备地址 **当短接 RESET-GND 脚启动模块时，模块将以默认参数作为通讯参数：9600，n,8,1 站号：1**	
40035 …… 40040	34 …… 39	未使用		
40039	38	零点校准 用于校准时使用 不可或操作，写后将清除整个寄存器。优先处理最低位。	位操作 Bit0 置 1 校准秤 1 零点 Bit1 置 1 校准秤 2 零点	写
40040	39	校准点校准 同上	位操作 Bit0 置 1 校准秤 1 砝码点 Bit1 置 1 校准秤 2 砝码点	写
40041 40042	40 41	秤 1 满量程设置	高 16 位 低 16 位	读写
……	……	……	……	
40059 40060	58 59	秤 10 满量程设置	高 16 位 低 16 位	
40061 40062	60 61	秤 1 校准值设置	高 16 位 低 16 位	读写
……	……	……	……	
40079 40080	78 79	秤 10 校准值设置	高 16 位 低 16 位	

40081 …… 40090	80 …… 89	秤 1 校准精度设置 …… 秤 10 校准精度设置	精度设置 0-3 分别代表 0 0.0 0.00 0.0000	读写
40091 …… 40100	90 …… 99	秤 1 校准分度值设置 …… 秤 10 校准分度值设置	精度设置 0-2 分别代表 1 2 5	读写
40101 …… 40110	100 …… 109	查询秤 1 校准状态 …… 查询秤 10 校准状态	0x0040 零点校准成功 0x0080 标定点校准成功 0x0001 正在校准 0x0003 校准失败	读
40111 …… 40120	110 …… 119	秤 1 滤波设置设置 …… 查询秤 10 滤波设置	0 弱 1 中 2 强	读写

1、-

2、每一个寄存器为 16 位数据类型

2、03 为 modbus 读多个寄存器，06 为写单个寄存器，16 为写多个寄存器指令

3、当选用 485 接口时，可通过设置设备地址将多个模块进行组网。

4、发送指令速率不超过 10HZ。

5、CRC16 计算方式为 0XA001-RTU 标准。

16、有关 MODBUS 更多细节，请参照 MODBUS 标准协议。

六、协议详细说明

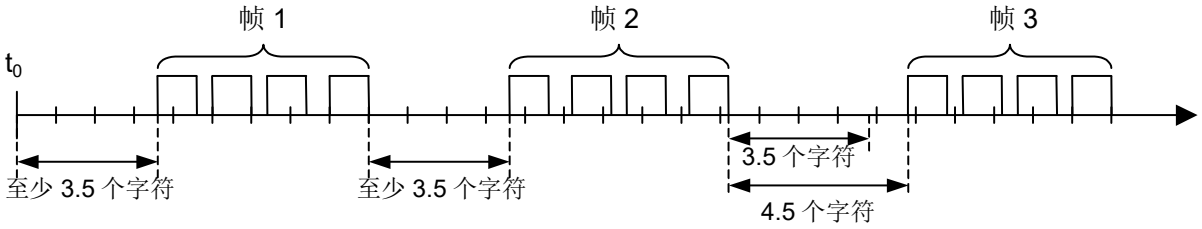
如果熟悉 modbus-rtu 协议，本章内容可以跳过。

使用 RTU 模式，一帧完整指令发送至少要以 3.5 个字符时间的停顿间隔开始。也就是 Modbus 协议并没有明显的用于判断一帧结束的帧头和帧尾，而是判断帧与帧间的间隔时间。

而单个字符之间的空闲不得超过 1.5 个字符。

(如果双方发送单个数据的间隔不超过 1 个字符时间的话，单片机可以使用空闲帧中断来高效实现通信。常用定时判断空闲帧。)

如果不再乎速率的情况下，可以放宽帧间的时间。如主机发送一条指令后，等到下一个询问周期再发送（比如 50ms 发送一次，一秒可获取 20 组数据，需要考虑挂载个数及下位机的承载能力，这样就无需要考虑帧间隔时间）。



MODBUS 报文

起始	地址	功能代码	数据	CRC 校验	结束
≥ 3.5 字符	8 位	8 位	N x 8 位	16 位	≥ 3.5 字符

在下位机中，MCU 在内存中开辟了一组空间(N 个 16 位的数组以下称寄存器)，并且根据映射的地址赋予不一样的功能（即在单片机中每个数组成员代表着不同功能），下位机实时的更新数组内的数据，上位机可通过 03 指令读取，或者使用 06、16 指令进行修改。

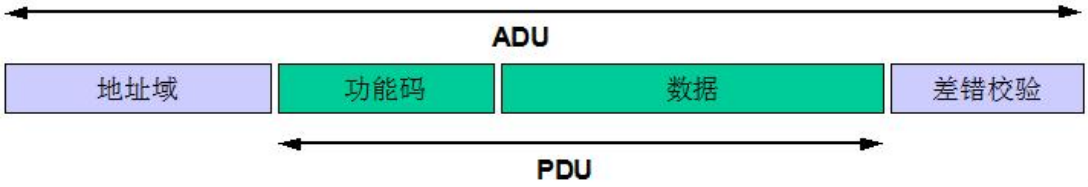
映射地址	地址内容(16 位)
0	称重数据高 16 位
1	称重数据低 16 位
2	小数点精度
3	称重状态
…	

则，上位机可以使用 03 指令读取地址 2 的数据，也可以读取地址 3 的数据，也可以一次性读取 2 3 4 的数据。

可以使用 06 指令写地址 1 的数据，也可以写 2 的数据。或者使用 16 写多个寄存器的数据。

Modbus 协议帧

无论是使用哪种指令，MODBUS 的协议帧都如下所示，由设备地址，功能码、数据域、CRC16 组成。



地址域 ：用于组网（1-30，0 为广播码，上位机发送无返回）
功能码 ：读写码（03 06 16）
数据 ：包含数据个数、数据值
差错校验：CRC16

03 指令 （PDU）读 N 寄存器

功能码	1 个字节	**0x03**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125（0x7D）

响应

功能码	1 个字节	**0x03**
字节数	1 个字节	2×**N***
寄存器值	**N***×2 个字节	

***N**＝寄存器数量

如获取设备号为 1 且寄存器地址为 0 和 1 两个数据指令如下：

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x01	0x00	0x02	xx	xx

返回（返回其它表示错误）

	设备号	功能码	返回数据个数	数据个数 4 个字节		CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x04	重量高位	重量低位	xx	xx

对于不同的设备地址，只需要改动设备号和 crc 校验

06 指令（PDU 部分）写单个寄存器

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

响应

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

如设置设备 2 的 07 地址数据为 2

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x02	0x06	0x00	0x07	0x00	0x02	xx	xx

返回和发送一置。

16(0x10)指令（PDU 部分）写 N 个寄存器

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0078
字节数	1 个字节	2×**N***
寄存器值	**N***×2 个字节	值

***N**＝寄存器数量

响应 PDU

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 123（0x7B）

错误

差错码	1 个字节	**0x90**
异常码	1 个字节	01 或 02 或 03 或 04

如设置设备 1 的 1、2 寄存器值（2 个 16 位数据即 4 个字节）

	设备号	功能码	起始地址高位	起始地址低位	寄存器数量		字节数	数据	CRC16 低位	CRC16 高位
上位机发送	0x01	0x10	0x00	0x01	0x00	0x02	0x04	4 字节	Xx	xx

以上了解了几个功能码后，要获取组网设备的数据只需要修改设备号和 CRC 校验即可获取不同设备的重量等数据。因此可以将 Modbus 当作普通自定义协议处理，只需要处理好帧之间的间隔时间，以及考虑多设备组网下发指令单片机负荷。组网个数越多间隔时间因控制得长些。

七、指令分解说明

（一）重量读取指令

以下指令的末两都是 CRC 校验，文档将以 xx 代替，该值和设备地址，以前实际数据有关系。实际情况下需要自行在软件中计算。建议写个计算 CRC 库。

1、获取单路重量（读多路时不建议使用）

根据 modbus 协议获取寄存器重量，实际上可以获取任意一路的重量值，也可以一次性获取 10 路的重量值。如果是需要获取 10 路的重量，建议一次性读取效率最高。

1.1、获取 1 路净重值（上位机->变送器）【01 03 00 00 00 02 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x00	0x00	0x02	0xc4	0x0b

->返回 32 位净重值及相就状态位（变送器->上位机）。如 15.32 则返回 1532。

	设备号	功能码	返回数据个数	数据个数 4 个字节	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x04	32 位有符号重量数据	xx	xx

将上面发送的数据改成 01 03 00 02 00 02 xx xx。则返回第 2 路重量数据。

发送 01 03 00 04 00 02 xx xx。则返回第 3 路的重量数据。

发送 01 03 00 00 00 04 xx xx。则返回第 1 路和第 2 路的重量。

返回数据为：01 03 08 32 位重量 1 32 位重量 2 xx xx。

2、获取 10 路重量（读多路时建议使用，效率高）

1.2、获取 10 路净重值（上位机->变送器）【01 03 00 00 00 14 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x00	0x00	0x14	Xx	Xx

->返回 32 位净重值及相就状态位（变送器->上位机）。如 15.32 则返回 1532。

	设备号	功能码	返回数据个数	数据个数 4 x10 个字节	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x28	10x32 位有符号重量数据	xx	xx

注：重量为 2 个寄存器数据，占 4 个字节。读取出来的重量直接赋值给 32 位有符号的变量即可。

3、获取状态位

1、获取状态位（上位机->变送器）【01 03 00 14 00 05 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x14	0x00	0x05	xx	xx

->反回 10 个字节的状态（变送器->上位机）

	设备号	功能码	返回数据个数	10 字节数据位	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x0a	10 字节状态	xx	xx

(1)秤 1-10 状态位表示（1 个秤对应 1 个字节）

7	6	5	4	3	2	1	0
0 称重无效 1 称重有效	0 模块已就绪 1 模块未就绪	保留			0:没有超载 1:超载	0:称重不稳定 1:稳重稳定	0:开机重量稳定 1:开机重量不稳定

Bit7:称重数据有效， 可通过该位来确认读取的重量值是否有效。

Bit6:模块就绪，开机 10s 内模块先判断传感器输出是否稳定，稳定则进入就绪状态，可以进行称重操作。因此可根据此位来判断当前是否可以称重。每一路都会有判断是否就绪。

Bit0:10s 后如果模块依旧没有稳定，但也会进入称重状态，该位置 1。

字节 1：第 2 路状态

字节 2：第 1 路状态

字节 3：第 4 路状态

字节 4：第 3 路状态

注意判断高低位即可。

4、可一条指令读取重量和状态位

1、可一次性读取出重量和状态位读取 25 个寄存器，50 个字节【01 03 00 00 00 19 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x00	0x00	0x19	xx	xx

->反回 10 个字节的状态（变送器->上位机）

	设备号	功能码	返回数据个数	40 字节数据位	10 个字节数据位	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x32	前 4x10 个为重量，	后 10x1 个为状态位	xx	xx

数据解析如上

5、获取精度

获取精度值（上位机->变送器） 【01 03 00 19 00 03 E5 CA】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x19	0x00	0x03	Xx	xx

->返回 32 位净重值及相就状态位（变送器->上位机）

	设备号	功能码	返回数据个数	2 个字节	2 个字节	1 个字节	1 字节	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x06	重量 1、2、3、4 精度	重量 5、6、7、8 精度	重量 9、10 精度	保留	xx	xx

(1) 返回值意义

第 1-2 个字节：

15-12	11-8	7-4	3-0
称重 1 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 2 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 3 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 4 的精度 0：0 1：0.0 2:0.00 3:0.000

第 3-4 个字节：

15-12	11-8	7-4	3-0
称重 5 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 6 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 7 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 8 的精度 0：0 1：0.0 2:0.00 3:0.000

第 5-6 个字节：

15-12	11-8	7-0
称重 9 的精度 0：0 1：0.0 2:0.00 3:0.000	称重 10 的精度 0：0 1：0.0 2:0.00 3:0.000	保留

如果 10 路精度是固定的，并且今后也只有这一种精度，那们在上位机中可直接约定精度，而不用再读取此值。

因读取的重量是整数，比如 12.32，读取出来是 1232。如果精度约定好是 2 位小数，则在上位机中读取到 1232 后可自行增加小数点。而如果十路中，或者通讯网络中有不同精度的称重，那么可通过读取此寄存器来区分精度。如第一路是 0.00 的秤，第 2 路是 0.0。第一路读取重量为 1232，第二路为 3245。读取精度寄存器可以计算出实际值为 12.32 和 324.5。

（二）、称重操作及配置指令

1、对模块清零操作（上位机->变送器）【01 06 00 1c 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x1c	0x00	0x01	Xx	xx

清零是消除温度引起的零点漂移。

bit4	Bit3	Bit2	Bit1	bit0
1:秤 5 置零	1:秤 4 置零	1:秤 3 置零	1:秤 2 置零	1:秤 1 置零

bit15-bit10	bit9	bit8	bit7	bit6	bit5
保留	1:秤 10 置零	1:秤 9 置零	1:秤 8 置零	1:秤 7 置零	1:秤 6 置零

该条指令可对任意一路称重进行置零操作，如数据写入 0x0001 则对秤 1 置零，写入 0x0002 则对秤 2 置零，写入 0x0003 则对秤 1 和秤 2 同时置零。

将十路一起置零，则写入 0x03ff。

->返回数据为所发送的数据（变送器->上位机）

2、开机置零操作（上位机->变送器）【01 06 00 1d 00 00 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x1d	0x00	0x00	xx	xx

操作位数据如下表

该操作置零方式类似：

清零是消除温度引起的零点漂移。

bit4	Bit3	Bit2	Bit1	bit0
0:秤 5 开机置零 1:秤 5 开机不置零	0:秤 4 开机置零 1:秤 4 开机不置零	0:秤 3 开机置零 1:秤 3 开机不置零	0:秤 2 开机置零 1:秤 2 开机不置零	0:秤 1 开机置零 1:秤 1 开机不置零

bit15-bit10	bit9	bit8	bit7	bit6	bit5
保留	0:秤 10 开机置零 1:秤 10 开机不置零	0:秤 9 开机置零 1:秤 9 开机不置零	0:秤 8 开机置零 1:秤 8 开机不置零	0:秤 7 开机置零 1:秤 7 开机不置零	0:秤 6 开机置零 1:秤 6 开机不置零

0 开机置零：开机无论传感器加载多少重量，都当作称重零点。

1 置零零点：设备将保存每一次的置零操作后的零点作为开机零点。

开机置零即为：开机前，秤台上无论放多少东西，秤开机自检后都显示 0 开始秤重。

开机不置零即为：关机后在秤上放了 10kg 的物体，开机后秤将显示物体重量。如果开机后的重量并非 10kg。这时候需要清除掉物体，此时会有余数，发送清零指令将示值清零，下次开机后计算重量将以该零点作为参考零点。该点实际应用中会因为环境、温度、传感器蠕变等因素而产生漂移。

同上，该条指令可对任意一路称重进行设置开机置零方式，如数据写入 0x0001 则秤 1

开机不自动置零，写入 0x0002 则秤 2 开机不自动置零，写入 0x0003 则秤 1 和秤 2 开机不自动置零

如果需要设置十路开机都不自动置零，则写入 0x03ff。

->返回数据为所发送的数据（变送器->上位机）

3、滤波强度设置（上位机->变送器）【01 06 00 6e 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x6e	0x00	低位	xx	xx

数据操作位

0x00	0x01	0x02
弱	中	强

滤波强度为了适应不同的环境称重的稳定性，越稳定则反应灵敏度变低。默认为中。

->返回数据为所发送的数据（变送器->上位机）

该条指令只对秤 1 进行设置，如需要对秤 2 进行配置，只需要将地址相应增加即可。如对第 2 路进行配置，则发送 01 06 00 6f 00 01 xx xx（蓝色为地址位，红色为数据位）。相应第三路地址则为 0x0070，第四路为 0x0071，第 10 路为 0x0077。

注：滤波一般出厂前会设置出一个合理的值，正常工作环境及示值稳定的情况下可不用考虑设置该位。

（三）、通讯操作指令(短接 RESET-GND 脚 3 秒恢复默认设置)

1、波特率设置（上位机->变送器）【01 06 00 1e 00 00 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x1e	0x00	低位	Xx	xx

数据操作位

0x00	0x01	0x02	0x03
9600	14400	19200	38400

设置后只有发送生效指令后才生效。

->返回数据为所发送的数据（变送器->上位机）

2、校验位设置（上位机->变送器）【01 06 00 1f 00 00 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x1f	0x00	低位	Xx	xx

数据操作位

0x00	0x01	0x02
无	奇	偶

设置后只有发送生效指令后才生效。

->返回数据为所发送的数据（变送器->上位机）

3、设备站号设置（上位机->变送器）【01 06 00 20 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x20	0x00	低位	Xx	xx

数据操作位：1-30

设置后只有发送生效指令后才生效。

->返回数据为所发送的数据（变送器->上位机）

4、生效通讯参数设置（上位机->变送器）【01 06 00 21 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x21	0x00	0x01	Xx	xx

发送 1 后将生效当前的波特率、校验位、站号设置。

->返回数据为所发送的数据（变送器->上位机）

注：如果不需要组网，且 9600 的波特率能满足要求的情况下可以不用考虑配置该参数。当然 10 路秤重中获取 10 路重量，在波特率为 38400 的情况下，获取数据能力是 9600 的 4 倍。在显示重量情况下比 9600 更加快速。

对于奇偶校验，个人建议选择无。因为 modbus-rtu 本身自带了 crc 校验，而加一个奇偶校验反而增加了通讯负荷。

（四）、标定操作指令

1、称重满量程设置（上位机->变送器）

	设备号	功能码	起始地址高位	起始地址低位	寄存器数量		字节数	数据	CRC16 低位	CRC16 高位
上位机发送	0x01	0x10	0x00	0x28	0x00	0x02	0x04	4 字节	xx	xx

4 字节数据位

即所要设置的满量程的值，该值为无符号整型。该值设置后需要发送标定数据同步指令后才有效（如称重满量程为 10000g 精度为 0.1g。则需要发送 100000）。

->返回数据为

	设备号	功能码	起始地址高位	起始地址低位	寄存器数量		CRC16 低位	CRC16 高位
上位机发送	0x01	0x10	0x00	0x10	0x00	0x02	Xx	xx

称重重量 2 满量程设置将起始地址偏移 4。如下:

【0x01 0x10 0x00 0x2a 0x00 0x02 0x04　4 字节重量　xx xx】　蓝色为地址=0x0028+2

称重重量 3 满量程设置将起始地址偏移 8。如下：

【0x01 0x10 0x00 0x2c 0x00 0x02 0x04　4 字节重量　xx xx】　蓝色为地址=0x0028+4

以此类推其它几路设置

2、称重标定砝码值设置（上位机->变送器）

	设备号	功能码	起始地址高位	起始地址低位	寄存器数量		字节数	数据	CRC16 低位	CRC16 高位
上位机发送	0x01	0x10	0x00	0x3c	0x00	0x02	0x04	4 字节	Xx	xx

4 字节砝码重量值

即所要设置的砝码的值，该值为无符号整型。该值设置后需要发送标定数据同步指令后才有效（如砝码为 2000g 精度为 0.1g。则需要发送 20000）。

->返回数据为

	设备号	功能码	起始地址高位	起始地址低位	寄存器数量		CRC16 低位	CRC16 高位
上位机发送	0x01	0x10	0x00	0x3C	0x00	0x02	Xx	xx

称重 2 标定砝码设置将起始地址偏移 10。如

【0x01 0x10 0x00 0x3e 0x00 0x02 0x04　4 字节重量　xx xx】　蓝色为地址=0x003c+2

称重 3 标定砝码设置将起始地址偏移 8。如

【0x01 0x10 0x00 0x40 0x00 0x02 0x04　4 字节重量　xx xx】　蓝色为地址=0x003c+4

以此类推其它几路设置

3、称重精度设置（上位机->变送器）

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x50	高位	低位	Xx	xx

设置值如下，该值设置后需要发送标定数据同步指令后才有效

0x0000	0x0001	0x0002	0x0003
0	0.0	0.00	0.000

->返回数据为所发送的数据（变送器->上位机）

称重 2 精度设置将起始地址偏移 1。即 0x0051

称重 3 精度设置将起始地址偏移 2。即 0x0052

4、称重分度值设置（上位机->变送器）

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x5a	高位	低位	Xx	xx

设置值如下，该值设置后需要发送标定数据同步指令后才有效

0x0000	0x0001	0x0002	其它
1	2	5	保留

分度值，常用为 1 默认为 1。

称重 2 分度值设置将起始地址偏移 1。即 0x005b

称重 3 分度值设置将起始地址偏移 2。即 0x005c

5、零点标定（上位机->变送器）【01 06 00 26 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x26	0x00	0x01	xx	xx

设置值如下，该值设置后需要发送标定数据同步指令后才有效

需要注意的是同一时间只能校准一路零点。如果多位置 1，只识别第 1 路。

bit4	Bit3	Bit2	Bit1	bit0
1:秤 5 零点校准	1:秤 4 零点校准	1:秤 3 零点校准	1:秤 2 零点校准	1:秤 1 零点校准

bit15-bit10	bit9	bit8	bit7	bit6	bit5
保留	1:秤 10 零点校准	1:秤 9 零点校准	1:秤 8 零点校准	1:秤 7 零点校准	1:秤 6 零点校准

该条指令可对任意一路称重进行零点操作，如数据写入 0x0001 则对秤 1 零点校准，写入 0x0002 则对秤 2 零点校准，写入 0x0004 则对秤 3 零点校验。

第十路零点校准，则写入 0x0200。

->返回数据为所发送的数据（变送器->上位机）

发送标定零点后，需要查询标定状态，零点标定成功后可进行下一步操作。零点标定失败的原因之一即是当前 AD 值一直不能稳定，检查连线或者使用环境是否会造成称重不稳定。

6、加载标定——需要获取到当前状态稳定后发送（上位机->变送器）

	设备号	功能码	起始地址高位	起始地址低位	2 字节操作数据位		CRC16 低位	CRC16 高位
上位机发送	0x01	0x06	0x00	0x27	0x00	0x01	xx	Xx

设置值如下，该值设置后需要发送标定数据同步指令后才有效

标定完零点后，且零点标定有效，则将设置好的标定值的砝码放到秤台上等其稳定后发送该条指令。

需要注意的是同一时间只能校准一路零点。如果多位置 1，只识别第 1 路。

bit4	Bit3	Bit2	Bit1	bit0
1:秤 5 砝码校准	1:秤 4 砝码校准	1:秤 3 砝码校准	1:秤 2 砝码校准	1:秤 1 砝码校准

bit15-bit10	bit9	bit8	bit7	bit6	bit5
保留	1:秤 10 砝码校准	1:秤 9 砝码校准	1:秤 8 砝码校准	1:秤 7 砝码校准	1:秤 6 砝码校准

该条指令可对任意一路称重进行砝码校准操作，如数据写入 0x0001 则对秤 1 砝码加载校准，写入 0x0002 则对秤 2 砝码加载校准，写入 0x0004 则对秤 3 砝码加载校验。

第十路砝码加载校准，则写入 0x0200。

->返回数据为所发送的数据（变送器->上位机）

零点标定成功后，将设置好的标定值对应的砝码放到秤量台上后发送该指令。随后查询标定状态。直到加载点标定成功后，再时行以下操作。如果标定失败，则需要检查连线或者使用环境是否会造成称重不稳定。

7、标定状态查询（上位机->变送器）【01 03 00 64 00 01 xx xx】

	设备号	功能码	起始地址高位	起始地址低位	寄存器数高位	寄存器数低位	CRC16 低位	CRC16 高位
上位机发送	0x01	0x03	0x00	0x64	0x00	0x01	xx	xx

->返回 8 位状态位（变送器->上位机）

	设备号	功能码	数据位数高位	数据个数 1 个字节	CRC16 低位	CRC16 高位
上位机接收	0x01	0x03	0x01	1 字节状态	xx	xx

(1)状态位表示

7-4	3	2	1	0
保留	0:加载点无效 1:加载点有效	0:零点无效 1:零点有效	0:称重状态 2:校准成功	1:正在校准 3:校准失败

称重 2 标定状态将起始地址偏移 1。即 0x0065

称重 3 标定状态将起始地址偏移 2。即 0x0066

注 对十路标定，建议一路标定成功后再标定下一路。

八、操作与说明

1、置零

置零是用于清除称重随温度或其它因素引起的零点漂移。清零有一定的范围要求，为保护传感器用户可根据置零范围进行设置（国标置零范围为+-2%）。清零不改变称量范围。

九、标定说明

零点标定，即变送器记录所接传感器空载时的输出值。标定零点时，变送器会判断当前传感器输出值是否处于稳定状态，如果 10s 内一直不稳定状态，则零点标定无效。上位机可查询当前的标定状态

标定点标定，即变送器记录所接传感器放上对应砝码的输出值。标定该值时，变送器会判断当前传感器输出值是处于稳定状态同时会判断零点是否标定成功，如果 10s 内不稳定状态或者零点标定未成功，则标定点标定无效。

当零点标定及标定点标定都成功后，此时发送数据标定数据生效指令更新当前校准数据。

标定步骤

如：满量程 2000g 精度 0.01g、分度值为 1、砝码重量为 1000g：

1、设置满量程值：200000

2、设置砝码值，即标定点 100000

3、设置精度值 2

4、设置分度值 0

5、清空秤台，发送标定零点指令

6、查询标定状态，直到显示 0x04，放上标定砝码

7、发送标定点指令

8、查询标定状态，直到显示 0x0c 再发送更新标定数据指令