

SMC35 系列霍尔操纵杆



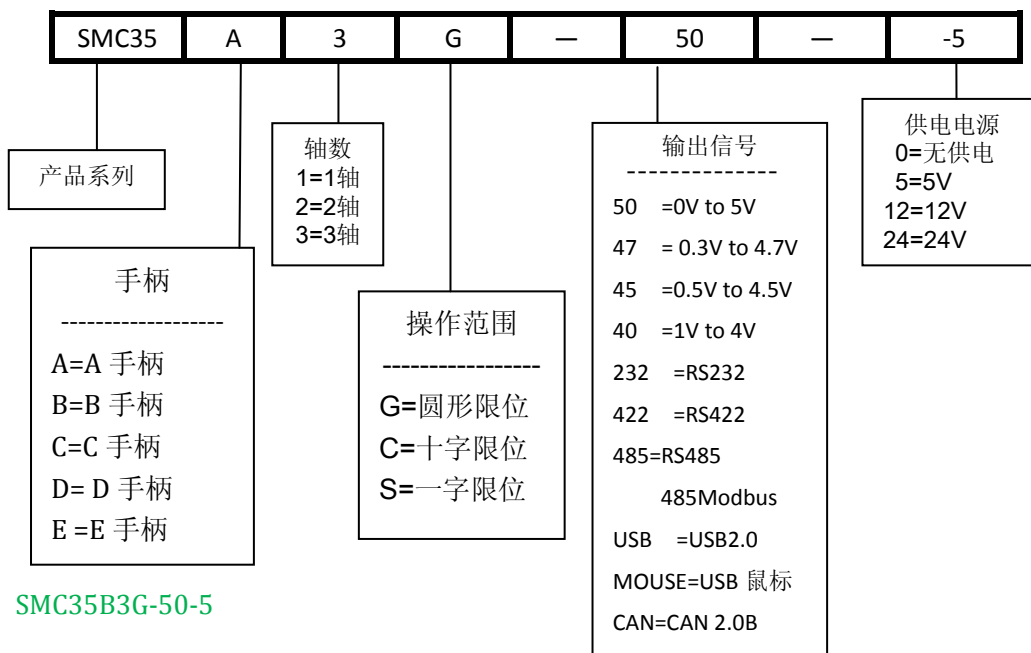
SMC35 系列霍尔操纵杆，1 轴、2 轴或 3 轴，面板安装及嵌入式，采用不锈钢及合金材料，弹簧自动回位结构，德国高精度霍尔式传感器，全温度范围线性校正，IP67 以上防护等级，较平滑的操作手感，人体工学机械设计。

适用于机器人、无人机、医疗设备、航天、船舶、广播电视设备等。

一、产品特点：

- 材 料： 不锈钢+合金+工程塑料
- 传 感 器： 霍尔传感器
- 定 位： 弹簧自动回位
- 操作角度： XY 轴： ± 20 度； Z 轴： ± 17 度
- 操作范围： 全方位(圆形)、十字、一字；
- 信号输出： 模拟电压、RS485、RS422、RS232、USB、CAN
- 供 电： DC5V、DC12-28V；
- 功 耗： 小于 9-25MA（5V 供电、与信号输出及轴数有关）
- 操作寿命： 大于 1000 万次；
- 温 度： -40 度~+70 度
- 防护等级： IP67 面板以上部分
- 底座尺寸： 44.4X44.4
- 按装尺寸： 35X35

产品型号参数选择



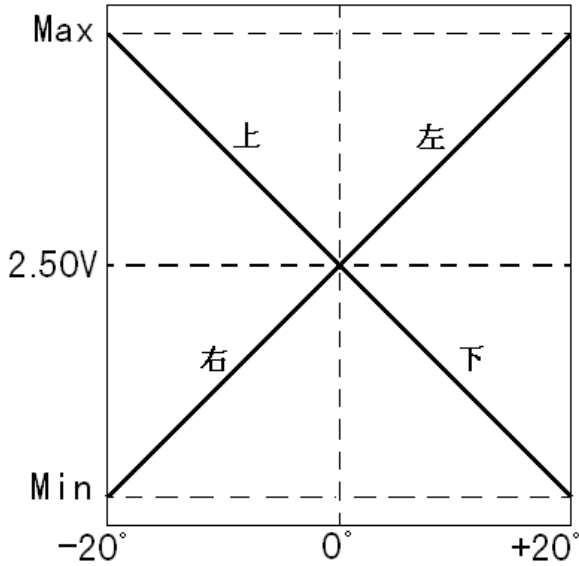
例如：SMC35B3G-50-5

- CAN 通信通信，波特率、支持标准帧/扩展帧/远程帧、协议可定制
- 弹簧力度：多种，请与工厂联系
- Z 轴是手柄旋转，可以顺时针或逆时针转动±17°，弹簧自动复位；

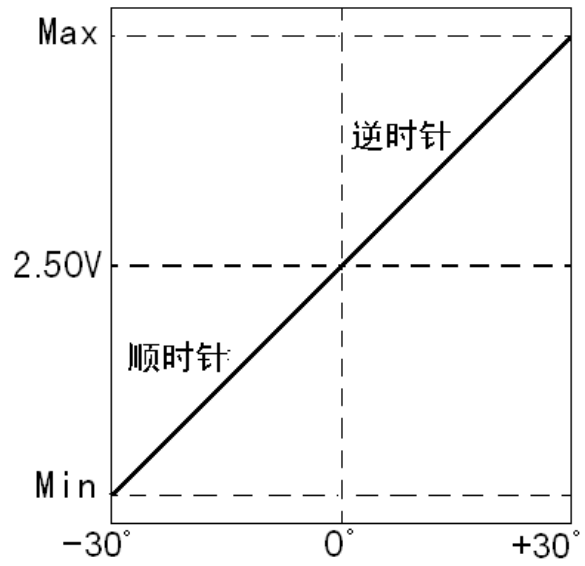
手柄选择

				
A 手柄	B 手柄	C 手柄	D 手柄	E 手柄
无按钮 轴数：1、2、3	1 按钮 轴数：1、2、3	无按钮 轴数：1、2、3 手柄可伸缩 伸缩高度：10mm 用于便携设备	1 按钮 轴数：1、2、3	无按钮 轴数：2、3 用于便携设备 杯形装饰盖 + 方形装饰盖

G=圆形限位	C=十字限位	S=一字限位
		



XY 轴模拟电压信号输出图形



Z 轴模拟电压信号输出图形

按钮开关原理图:



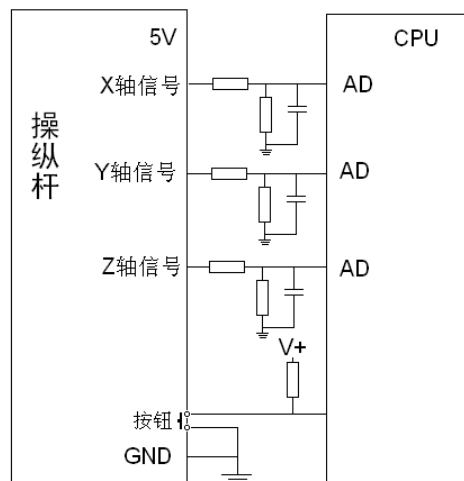
按钮开关技术参数:

- ◆ 开关方式: 常开
- ◆ 触点电流: B 手柄 1A/24V ; D 手柄 0.5A/24V
- ◆ 防护等级: B 手柄 IP67; D 手柄 IP54
- ◆ 操作寿命: 大于 100 万次

电气参数:

- 最低工作电压: 3.1V (5V 供电时)、8V (12-24V 供电时)
- 最高输入电压: 5.5V (5V 供电时)、28V (12-24V 供电时)
- 工作电流: 小于 25mA (5V 供电, 模拟电压信号输出时)
- 模拟电压信号输出负载: 大于 1KΩ
- 模拟电压信号输出中心电压: 2.50V 或 50%Vdd
- 模拟电压输出信号: 0V~5V / 0.3V ~4.7V / 0.5V ~ 4.5V / 1V~ 4V

接线图

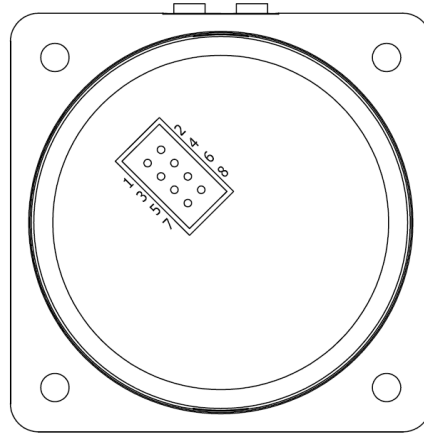


如果 CPU 的电压为 3.3V
那么分压电阻值 1.5K 和 3K
电容=1NF

如果 CPU 的电压是 5V
取消分压电阻
直接连接到 CPU 引脚
电容可有也可无;

按钮的上接电阻 10K

操纵杆端子图：



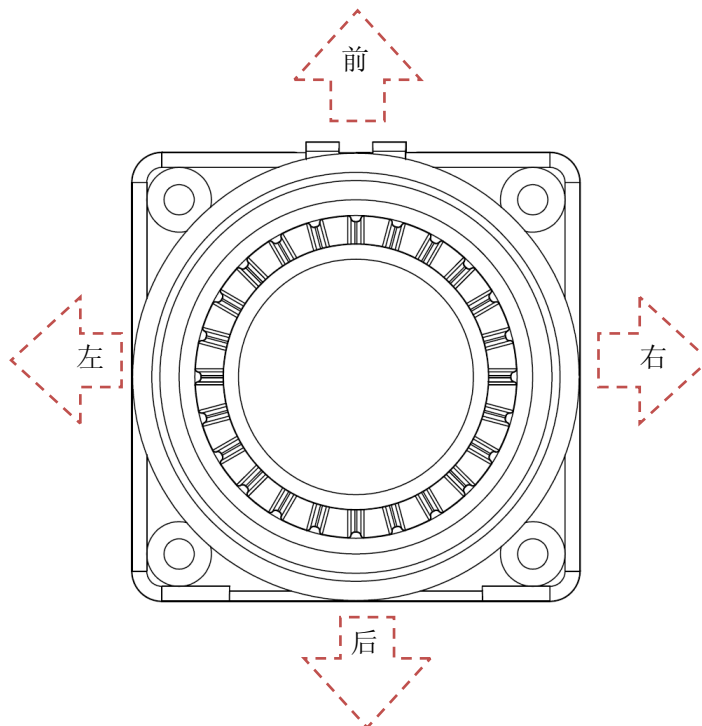
引线：

引线长度：20cm（含端子）

端子型号：PHD2.0-8P

连接器引脚定义-模拟电压信号输出：

引脚	符号	颜色	功能说明
1	V+	红色	电源+
2	Y	绿色	Y 轴信号输出
3	GND	黑色	GND 电源-
4	X	黄色	X 轴信号输出
5	B	白色	按钮（按钮有 2 脚，第 1 脚）
6	Cen	棕色	中心电位信号（可不用）
7	Z	蓝色	Z 轴信号输出
8	B	白色	按钮（按钮有 2 脚，第 2 脚）



连接器引脚定义-RS232 信号输出:

引脚	符号	颜色	功能说明
1	V+	红色	电源+
2	RX	黄色	RS232 数据接收
3	GND	黑色	GND 电源-
4	TX	绿色	RS232 数据发送
5	X		(工厂测试 X)
6	NC		空
7	Y		(工厂测试 Y)
8	B		(工厂测试 Z)

连接器引脚定义-RS422 信号输出:

引脚	符号	颜色	功能说明
1	V+	红色	电源+
2	RX+	黄色	RS422 数据接收正
3	GND	黑色	GND 电源-
4	RX-	绿色	RS422 数据接收负
5	X		(工厂测试 X)
6	TX+	蓝色	RS422 数据发送正
7	Y		(工厂测试 Y)
8	TX-	白色	RS422 数据发送负 / (工厂测试 Z)

连接器引脚定义-RS485 信号输出:

引脚	符号	颜色	功能说明
1	V+	红色	电源+
2			
3	GND	黑色	GND 电源-
4			
5	X		(工厂测试 X)
6	RS485A+	蓝色	RS422 数据发送正
7	Y		(工厂测试 Y)
8	RS485A-	白色	RS422 数据发送负 / (工厂测试 Z)

连接器引脚定义-CAN 信号输出:

引脚	符号	颜色	功能说明
1	V+	红色	电源+
2	TX	绿色	RS232 数据发送(RS232 用于参数设置)
3	GND	黑色	GND 电源-
4	RX	黄色	RS232 数据接收(RS232 用于参数设置)
5	CAN-H	蓝色	CAN 信号高
6	Y		(工厂测试 Y)
7	CAN-L	白色	CAN 信号低 / (工厂测试 Z)
8	X		(工厂测试 Y)

只引出红、黑、蓝、白 4 条线，RS232 用于 CAN 通信参数设置

连接器引脚定义-USB 信号输出:

引脚	符号	颜色	功能说明
1	Z		测试-Z (工厂测试用)
2	V+	红色	电源+
3	X		测试-X (工厂测试用)
4	D-	白色	USB 信号输出 D-绿色
5	Y		测试-Y (工厂测试用)
6	D+	蓝色	USB 信号输出 D+
7	GND	屏蔽	USB 线的屏蔽层
8	GND	黑色	GND 电源-

USB 接口的相关连接线及通信协议

USB 通信协议: USB 2.0 HID 人机介面协议标准

支持微软操作系统, 免驱动; 支持 directX 库
相关例程网上查 “joystick directx input”

一、USB 键盘发送的数据格式 (7 个字节 HEX):

USB 键盘发送键盘的 3 轴操纵杆的角度参数和按键盘的状态值

byte1	byte2	byte3	byte4	byte5	byte6	byte7
XXL	XXH	YYL	YYH	ZZL	ZZH	BB1

XXXX: X 轴数据,0000-03FF,(BYTE2 数据高位,BYTE1 数据低位)

0X0000-0X01FE 左
0X0200 停止
0X0200-0X03FF 右

YYYY: Y 轴数据,0000-03FF,(BYTE4 数据高位,BYTE3 数据低位)

0X0000-0X01FE 下
0X0200 停止
0X0200-0X03FF 上

ZZZZ: Z 轴数据,0000-03FF,(BYTE6 数据高位,BYTE5 数据低位)

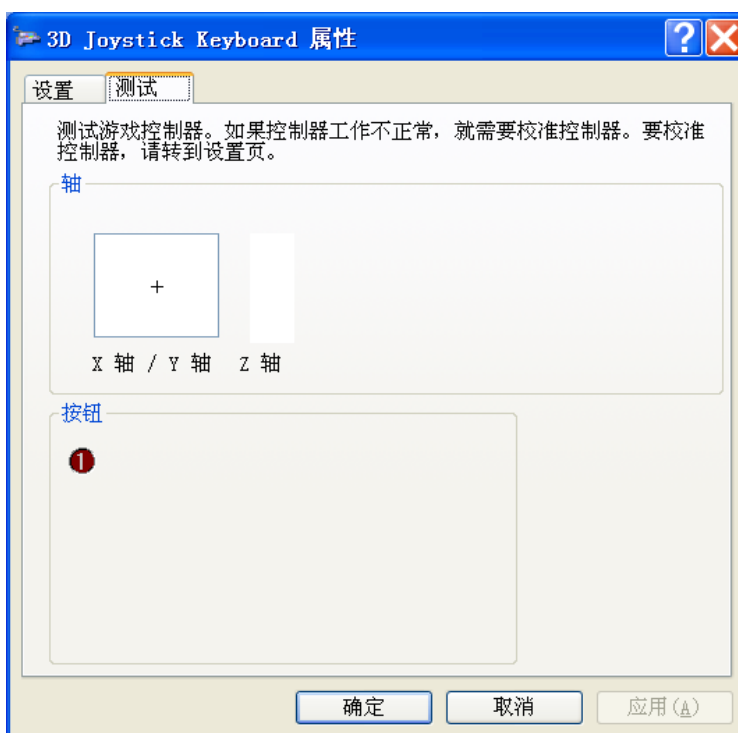
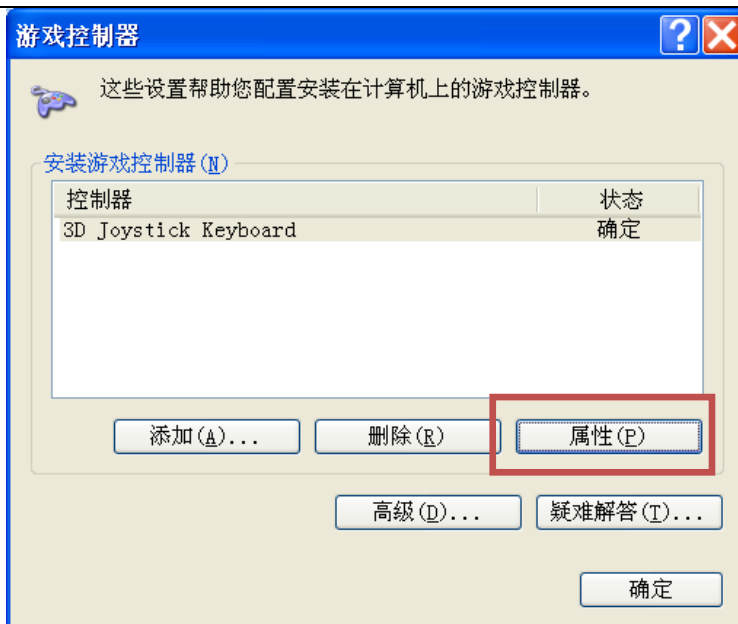
0X0000-0X01FE 逆时针(wide)
0X0200 停止
0X0200-0X03FF 顺时针(tele)

BB1:按钮第 1 组

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
按钮 8	按钮 7	按钮 6	按钮 5	按钮 4	按钮 3	按钮 2	按钮 1

2. 键盘操纵杆中心点设置指令 (接收来自主机的指令), 出厂时已经设置好, 用户可不用这个指令;

设置操纵杆中心点 F5 00 00 00 00 01 55 56 (操纵杆接收来自主机的指令)



开发相关技术支持资料如下：

1. USB 测试软件
2. USB 编写例程
3. USB 键盘通信协议
4. 本键盘驱动免驱动, 支持 Windows 的 DirectX, 请在百度上搜索"direct input joystick", 网上有各种语言的相关设计.

SMC35 系列霍尔操纵杆 RS232/RS422 通信协议

通信方式:

- 数据有变就发送, 停在一个位置不动就不发送重复的数据了;
- 定时发送, 可在 10-50MS 定时发送, 这个时间可按客户要求修改;
<定货时注明通信方式>

一、通用协议

(无地址位, 适用于 RS232, RS422, RS485) 出厂默认为此模式);

1、操纵杆发送数据 (9 字节) (操纵杆-PC):

功能: 发送操纵杆的各轴的位置参数)

波特率 9600. 8. 1. N

操纵杆发送数据格式: (16 进制 HEX 数据)

FF	YYH	YYL	XXH	XXL	ZZH	ZZL	Button	CH
头	Y 高位	Y 低位	X 高位	X 低位	Z 高位	Z 低位	按钮	校验和

YYYY Y 轴角度

XXXX X 轴角度

ZZZZ Z 轴角度

BB joystick button 控制杆上的按钮

CH =XXH+XXL+YYH+YYL+ZZH+ZZL+Button (00-FF)

X 轴参数

MAX	左	MIN	停止	MIN	右	MAX
0X0030-	--	0X01ff	0200	0X0201-	--	0X03D0

Y 轴参数

MAX	下	MIN	停止	MIN	上	MAX
0X0030-	--	0X01ff	0200	0X0201-	--	0X03D0

Z 轴参数

MAX	WIDE	MIN	停止	MIN	TELE	MAX
0X0030-	--	0X01ff	0200	0X0201-	--	0X03D0

Button 按钮参数

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	Joystick button	0	0000			

Joystick button =1 有按键按下, 0 无按键按下

例如: FF 02 00 02 00 03 FF 00 06

2、设置操纵杆的中心点 (用于校正中心点位置) (PC->操纵杆)

数据格式: AA 55 AF 00

向操纵杆发送些数据, 重新设置操纵杆的停止位置 (中心点)

3、查巡操纵杆位置 (PC->操纵杆)

数据格式: AA 55 AF 01

操纵杆收到这个数据就回送当前的位置, 并不在主动发送数据.

二、带地址的总线通信协议

（设置地址后，此协议有效）

该方式通信适用于 RS422 通信方式，多个摇杆，同一总线连接，查询方式工作；

只有在设置了操纵杆的地址后，此通信方式才有效，当取消操纵杆地址时，此协议无效，并恢复通用协议。

以下数据为 HEX 格式，波特率波特率 9600. 8. 1. N

1. 修改操纵杆波特率(PC->操纵杆)

A5 55 01 xx FF

Xx: 00=9600

01=19200

02=115200

操纵杆收到此指令，波特率有效，然后回复 ACK

出厂默认 9600

9600 波特率: A5 55 01 00 FF

19200 波特率: A5 55 01 01 FF

115200 波特率: A5 55 01 02 FF

例如：当前 9600，修改为 19200

1. (9600) PC->操纵杆 A5 55 01 01 FF

2. 波特率更新

3. (19200) 操纵杆->PC A5 55 AF

2. 设置操纵杆地址(PC->操纵杆)

A5 55 03 Add FF

Add=0x01~0x40 地址 1-64

操纵杆收到此指令，执行后回复 ACK

例如：设置 1 号地址 A5 55 03 01 FF

3. 取消操纵杆地址(PC->操纵杆):

A5 55 05 00 FF

操纵杆收到此指令，执行后回复 ACK

恢复通用协议（通用协议-9 字节操纵杆发送数据通信方式）

4. ACK 确认（操纵杆-PC）

A5 55 AF

表明操纵杆成功接收到地址设置指令，并执行完成。

5. 带地址位的查巡（PC->操纵杆）

A5 55 09 Add FF

Add=0x01-0x40 地址 1-64

例如：查询 2 号地址 A5 55 09 02 FF

6. 带地址位的操纵杆发送数据（10 字节）（操纵杆-PC）

FF	Add	YYH	YYL	XXH	XXL	ZZH	ZZL	Button	CH
头	地址	Y 高位	Y 低位	X 高位	X 低位	Z 高位	Z 低位	按钮	校验和

Add 地址 0X01-0X40 与查巡的地址相同
其它与通用协议相同

CH =Add+XXH+XXL+YYH+YYL+ZZH+ZZL+Button (00-FF)

例如 FF 01 02 00 02 00 02 00 00 07

Modbus 通信协议 Modbus (RTU 模式)

操纵杆主站模式:

1. 波特率: 9600
2. 数据位: 1 个起始位, 8 个数据位, 1 个停止位, 无效验位
3. 通信接口: RS485 和 RS232 只能选其一, 不能同时用
4. 数据格式: Modbus
5. 工作模式: 主站 (主站向从站 1 发送数据)
6. 工作模式: 操纵杆->从站
 - ◆ 定时发送数据, 帧间隔 17ms, 约 20HZ/帧;
 - ◆ 从站不做应答;

功能	数据	参数范围
设备地址	0x01	Modbus 站号
功能码	0x10	
第 1 个寄存器地址-高位	0x60	寄存器地址
第 1 个寄存器地址-低位	0x00	
寄存器数量-高位	0x00	
寄存器数量-低位	0x04	
数据长度	0x08	
按钮高位 Bit15-Bit8	0x00	1-16 个按钮 (Bit0=按钮 1) 1=ON, 0=OFF
按钮低位 Bit7-Bit0	0x00	
操纵杆 1 X 轴高位	0x02	0x0030-----0x01ff, 0x0200, 0x0201-----0x03d0 (max 右 min) 停止 (min 左 max)
操纵杆 1 X 轴低位	0x00	
操纵杆 1 Y 轴高位	0x02	0x0030-----0x01ff, 0x0200, 0x0201-----0x03d0 (max 下 min) 停止 (min 上 max)
操纵杆 1 Y 轴低位	0x00	
操纵杆 1 Z 轴高位	0x02	0x0030-----0x01ff, 0x0200, 0x0201-----0x03d0 (max 逆 min) 停止 (min 顺 max)
操纵杆 1 Z 轴低位	0x00	
CRC 高位	36	
CRC 低位	E0	

定时 45ms 发送一帧

操纵杆从站模式：

（主从方式）

1. 波特率： 9600/115200
2. 数据位： 1 个起始位，8 个数据位，1 个停止位，无效验位
3. 通信接口： RS485 和 RS232 只能选其一，不能同时用
4. 数据格式： Modbus
5. 工作模式： 从站
6. 工作模式： 主从方式（收到读指令，回位 1 帧数据）

寄存器地址： 60000

Modbus 主机读取数据及从机应答格式（功能码 03）（PC→操纵杆）

字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7	字节 8
地址	读命令	开始地址		寄存器数		CRC	
0x01	0x03	高	低	高	低	高	低

实例： 01 03 60 00 00 04 5A 09

当操纵杆收到此指令，操纵杆由主模式改为从模式。（停止主动发送数据，只有收到这个读的数据，操纵杆才响应如下的数据）；

操纵杆的响应（操纵杆→PC）

功能	数据	参数范围
设备地址	0x01	设备地址
功能码	0x03	
数据长度	0x08	
按钮高位 Bit15-Bit8	0x00	1-16 个按钮（Bit0=按钮 1） 1=ON, 0=OFF
按钮低位 Bit7-Bit0	0x00	
操纵杆 1 X 轴高位	0x02	0x0030----0x01ff, 0x0200, 0x0201----0x03d0 (max 右 min) 停止 (min 左 max)
操纵杆 1 X 轴低位	0x00	
操纵杆 1 Y 轴高位	0x02	0x0030----0x01ff, 0x0200, 0x0201----0x03d0 (max 下 min) 停止 (min 上 max)
操纵杆 1 Y 轴低位	0x00	
操纵杆 1 Z 轴高位	0x02	0x0030----0x01ff, 0x0200, 0x0201----0x03d0 (max 逆 min) 停止 (min 顺 max)
操纵杆 1 Z 轴低位	0x00	
CRC 高位	94	
CRC 低位	ED	

例如：

主机→操纵杆： 设备地址=1： 01 03 60 00 00 04 5A 09

操纵杆→主机（响应）： 01 03 08 00 00 02 00 02 00 02 00 94 ED

（主机→操纵杆： 设备地址=2： 02 03 60 00 00 04 5A 3A）

Modbus 通信协议-操纵杆的参数设置 (RS232/RS422):**1. 修改操纵杆波特率 (PC->操纵杆)**

A5 55 01 xx FF

Xx: 00=9600

01=19200

02=115200

操纵杆收到此指令，波特率有效，然后回复 ACK

出厂默认 9600

例如：当前 9600，修改为 19200

4. (9600) PC->操纵杆 A5 55 01 01 FF

5. 波特率更新

6. (19200) 操纵杆->PC A5 55 AF

2. 设置操纵杆地址 (PC->操纵杆)

A5 55 03 Add FF

Add=0x01~0x40 地址 1-64

操纵杆收到此指令，执行后回复 ACK

例如：设置 1 号地址 A5 55 03 01 FF

3. 操纵杆中心校正 (PC->操纵杆):

A5 55 04 00 FF

操纵杆收到此指令，校正中心点，回复 ACK

4. 操纵杆工作模式 (PC->操纵杆):

A5 55 08 Mode FF

Mode=00 主站模式

Mode=01 从站模式

操纵杆收到此指令，修改模式，回复 ACK

当操纵杆接收到正确的“主机读取数据”指令，工作模式也自动改为从站模式 (01 03 60 00 00 04 5A 09) 但是不永久存贮，开机又恢复主站模式。

5. ACK 确认 (操纵杆-PC)

A5 55 AF

表明操纵杆成功接收到地址设置指令，并执行完成。

设置参数如下，通过串口进行设置

CAN 总线通信方式

CAN2.0B, 可定制协议

支持扩展帧 ID, 标准帧 ID, 远程帧, 用户可修改 ID

用户可修改波特率

发送方式: 间隔 45ms 连续发送

- 扩展数据帧: 默认 ID= 扩展码 0X00F0F101 (可通过 RS232 修改此结点标识码)
- 波特率: 默认=250K (可通过 RS232 修改波特率)
- 发送方式: 间隔 45ms 连续发送

操纵杆发送数据格式: (16 进制 HEX 数据)

BYTE0	XXL	X 轴低位	X 轴数据 0X0060~0X0800~0X0FA0
BYTE1	XXH	X 轴高位	
BYTE2	YYL	Y 轴低位	Y 轴数据 0X0060~0X0800~0X0FA0
BYTE3	YYH	Y 轴高位	
BYTE4	ZZL	Z 轴低位	Z 轴数据 0X0060~0X0800~0X0FA0
BYTE5	ZZH	Z 轴高位	
BYTE6	Button	按钮	按钮
BYTE7	0XA5		尾标志

XXL	XXH	YYL	YYH	0X00	0X00	Button	A5
X 低位	X 高位	Y 低位	Y 高位	0X00	0X00	按钮	尾标志

YYYY Y 轴角度

XXXX X 轴角度

ZZZZ Z 轴角度

Button 按钮

例如: 00 08 00 08 00 08 00 A5

X 轴参数

MAX	左	MIN	停止	MIN	右	MAX
0X0060-	--	0X07ff	0800	0X0801-	--	0X0FA0

Y 轴参数

MAX	下	MIN	停止	MIN	上	MAX
0X0060-	--	0X07ff	0800	0X0801-	--	0X0FA0

Z 轴参数

MAX	WIDE	MIN	停止	MIN	TELE	MAX
0X0060-	--	0X07ff	0800	0X0801-	--	0X0FA0

Button 按钮参数

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	按钮	0	0000			

1 有按键按下, 0 无按键按下

CAN 通信操纵杆参数设置 版本: Ver: 改 14. 04. 28

CAN 通信, 用户可能要对操纵杆进行结点标识进行设置, 通过操纵杆的 RS232 通信接口与 PC 相连接, 可对 CAN 通信的相关的技术参数进行设置;

操纵杆的 RS232 通信接口, 出厂参数: 波特率 9600. 8. 1. N

1、操纵杆发送数据的结点标识 (29 位扩展码) 设置: (PC->操纵杆)

PC 与操纵杆的 RS232 端口相连接, 波特率 9600

0xaf 0x01 D1 D2 D3 D4 0xf5
头 命令 数据 1 数据 2 数据 3 数据 4 尾

D1. 7=0 扩展帧 29 位

D1. 7=1 标准帧 11 位

- 29 位扩展帧: 数据范围 0X0-0X0FFFFFFF, 数据 D1-D4 对应“结点标识码”

例如: 设置发送结点标识码-扩展帧“0X00F0F101”

af 01 00 f0 f1 01 f5 (HEX)

- 11 位标准帧: 数据范围 0X000-0X3FF, 数据 D3-D4 对应“结点标识码”

例如: 设置发送结点标识码-标准帧“0X1E1”

af 01 80 00 01 E1 f5 (HEX)

2、操纵杆接收数据的结点标识 (29 位扩展码) 设置: (PC->操纵杆)

PC 与操纵杆的 RS232 端口相连接, 波特率 9600

0xaf 0x02 D1 D2 D3 D4 0xf5
头 命令 数据 1 数据 2 数据 3 数据 4 尾

D1. 7=0 扩展帧 29 位

D1. 7=1 标准帧 11 位

- 29 位扩展帧: 数据范围 0X0-0X0FFFFFFF, 数据 D1-D4 对应“结点标识码”

例如: 设置接收结点标识码-扩展帧“0X00F0F101”

af 02 00 f0 f1 01 f5 (HEX)

- 11 位标准帧: 数据范围 0X000-0X3FF, 数据 D3-D4 对应“结点标识码”

例如: 设置接收结点标识码-标准帧“0X1E1”

af 02 80 00 01 E1 f5 (HEX)

3、操纵杆屏蔽的结点标识符 (29 位扩展码) 设置: (PC->操纵杆)

PC 与操纵杆的 RS232 端口相连接, 波特率 9600

0xaf 0x03 D1 D2 D3 D4 0xf5
头 命令 数据 1 数据 2 数据 3 数据 4 尾

D1. 7=0 扩展帧 29 位

D1. 7=1 标准帧 11 位

- 29 位扩展帧: 数据范围 0X0-0X0FFFFFFF, 数据 D1-D4 对应“结点标识码”

例如: 设置屏蔽结点标识码-扩展帧“0X00002201”

af 03 00 00 22 01 f5 (HEX)

- 11 位标准帧: 数据范围 0X000-0X3FF, 数据 D3-D4 对应“结点标识码”

例如: 设置屏蔽结点标识码-标准帧“0X122”

af 03 80 00 01 22 f5 (HEX)

4、操纵杆通信端口设置：（PC→操纵杆）

PC 与操纵杆的 RS232 端口相连接, 波特率 9600
 0xaf 0x05 XX 00 00 00 0xf5
 头 命令 数据 1 数据 2 数据 3 数据 4 尾
 XX=00 CAN 通信;
 XX=01 RS232 通信
 XX=02 RS422 通信
 例如: af 05 01 00 00 00 f5 (HEX) RS232 通信
 af 05 02 00 00 00 f5 (HEX) RS422 通信
 af 05 00 00 00 00 f5 (HEX) CAN 通信

5、设 CAN 端口波特率：（PC→操纵杆）

PC 与操纵杆的 RS232 端口相连接, 波特率 9600
 0xaf 0x06 XX 00 00 00 0xf5
 头 命令 数据 1 数据 2 数据 3 数据 4 尾
 XX=00 125K
 XX=01 250K (默认)
 XX=02 500K
 XX=03 1000K
 例如: af 06 00 00 00 00 f5 (HEX) CAN 波特率=125K
 af 06 01 00 00 00 f5 (HEX) CAN 波特率=250K (默认)
 af 06 02 00 00 00 f5 (HEX) CAN 波特率=500K
 af 06 03 00 00 00 f5 (HEX) CAN 波特率=1000K

6、查巡操纵杆位置(PC→操纵杆)，仅限于 RS232 通信

0xaf 0x07 00 00 00 Addr 0xf5
 头 命令 数据 1 数据 2 数据 3 数据 4 尾
 ◆ Addr=0x01-0x40 当地址正确时，回送
 ◆ 当无地址模式时，收到任意地址都回送数据
 ◆ 操纵杆收到这个数据就回送当前的位置, 并不在主动发送数据.
 ◆ 只有 PC 向操纵杆发送复位指令后，操纵杆重新启动后，才能恢复自动发送模式（参见“复位操纵杆”指令）

例如 RS232：以当前的通信模式回送

(PC→操纵杆) af 07 00 00 00 01 f5 (HEX)

(操纵杆→PC) FF 01 08 00 08 00 08 00 00 19

操纵杆收到这个数据就回送当前的位置, 并不在主动发送数据.

7、设置操纵杆的中心点(用于校正中心点位置)（PC→操纵杆）

出厂时，已经设置好, 可忽略此指令
 PC 与操纵杆的 RS232 端口相连接, 波特率 9600
 0xaf 0x09 00 00 00 00 0xf5
 头 命令 数据 1 数据 2 数据 3 数据 4 尾
 向操纵杆发送些数据, 重新设置操纵杆的停止位置(中心点)
 例如: af 09 00 00 00 00 f5 (HEX)

8、设置 RS232 波特率(PC->操纵杆), 仅限于 RS232 通信

RS232 波特率一样, 设置同时有效

0xaf	0x0b	00	00	00	Baud	0xf5
头	命令	数据 1	数据 2	数据 3	数据 4	尾

Baud=0X00	波特率=9600
Baud=0X01	波特率=19200
Baud=0X02	波特率=57600
Baud=0X03	波特率=115200

例如: 设置 19200

af 0b 00 00 00 01 f5 (HEX)

操纵杆收到此指令, 执行后, 回复 ACK

9、设置操纵杆地址(PC->操纵杆), 仅限于 RS232 通信

0xaf	0x0d	00	00	00	Add	0xf5
头	命令	数据 1	数据 2	数据 3	数据 4	尾

Add=0x01~0x40 地址 1-64

Add=0x00 无地址位

例如: 设置 1 号地址

af 0d 00 00 00 01 f5 (HEX)

操纵杆收到此指令, 执行后, 回复 ACK

例如: 设置原来 1 号地址的摇杆为无地址方式

af 0d 00 00 00 00 f5 (HEX)

操纵杆收到此指令, 执行后, 回复 ACK

10、复位操纵杆 (PC->操纵杆), 仅限于 RS232 通信

0xaf	0x15	00	00	00	Add	0xf5
头	命令	数据 1	数据 2	数据 3	数据 4	尾

Add=0x01~0x40 地址 1-64, 地址要与操纵杆的地址一样才能复位

Add=0x00 复位所有地址的操纵杆, 任何地址 (包括无址) 都被复位

Add 范围不在 0-0x40 无效

例如: 复位 1 号地址操纵杆

af 15 00 00 00 01 f5 (HEX)

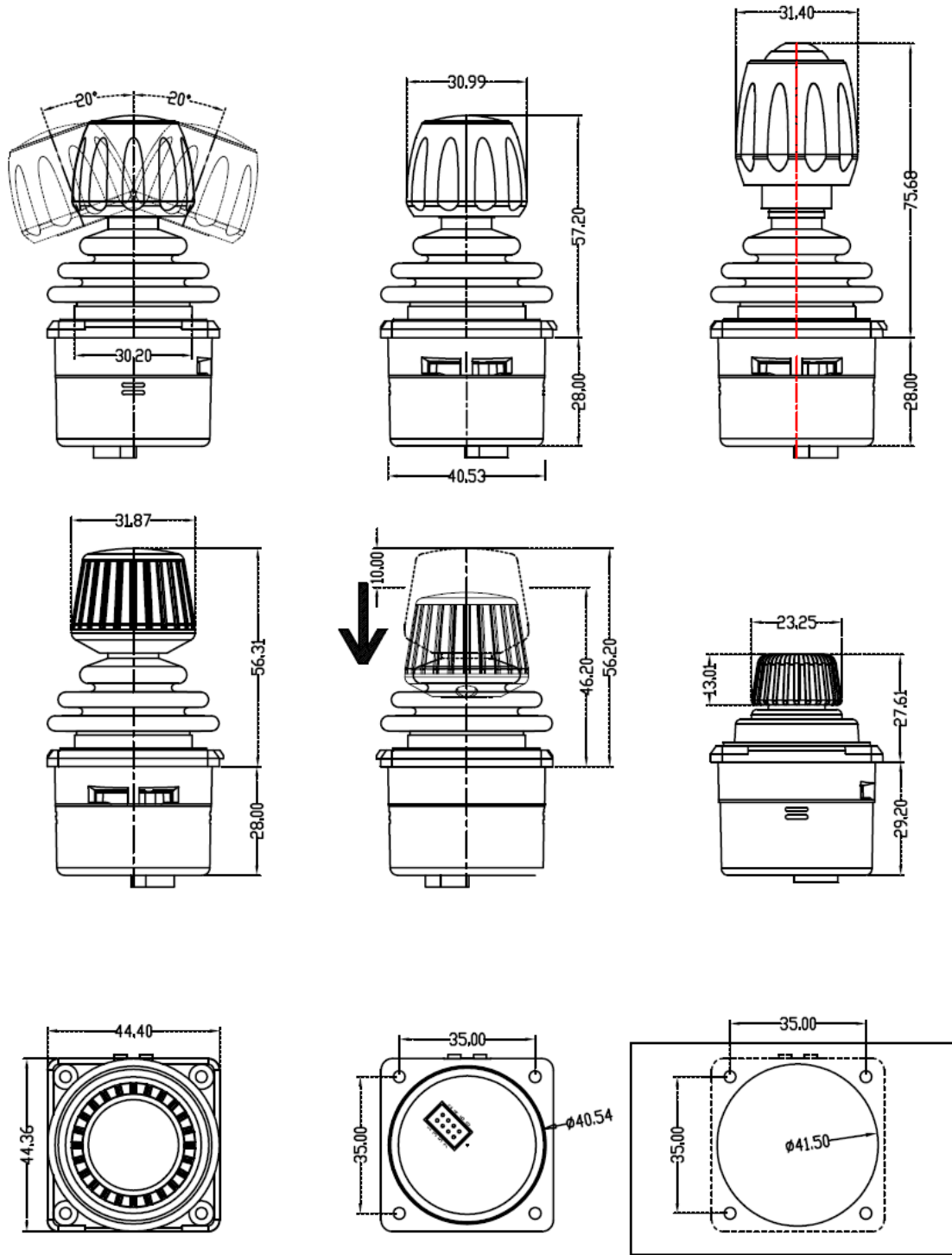
操纵杆收到此指令→回复 ACK→复位操纵杆

11、ACK 确认 (操纵杆-PC)

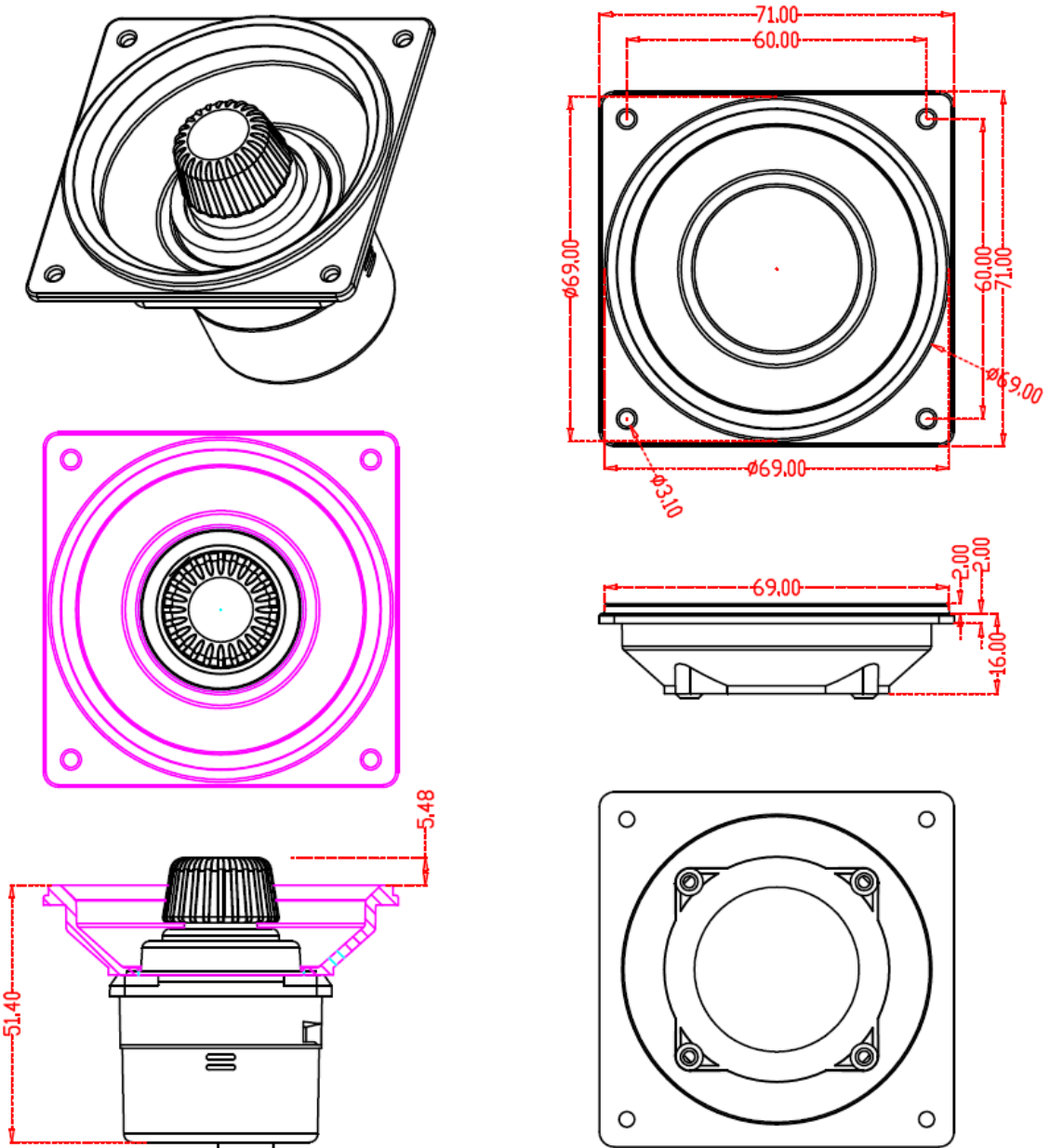
AA 55 AF

表明操纵杆成功接收到地址设置指令, 并执行完成。

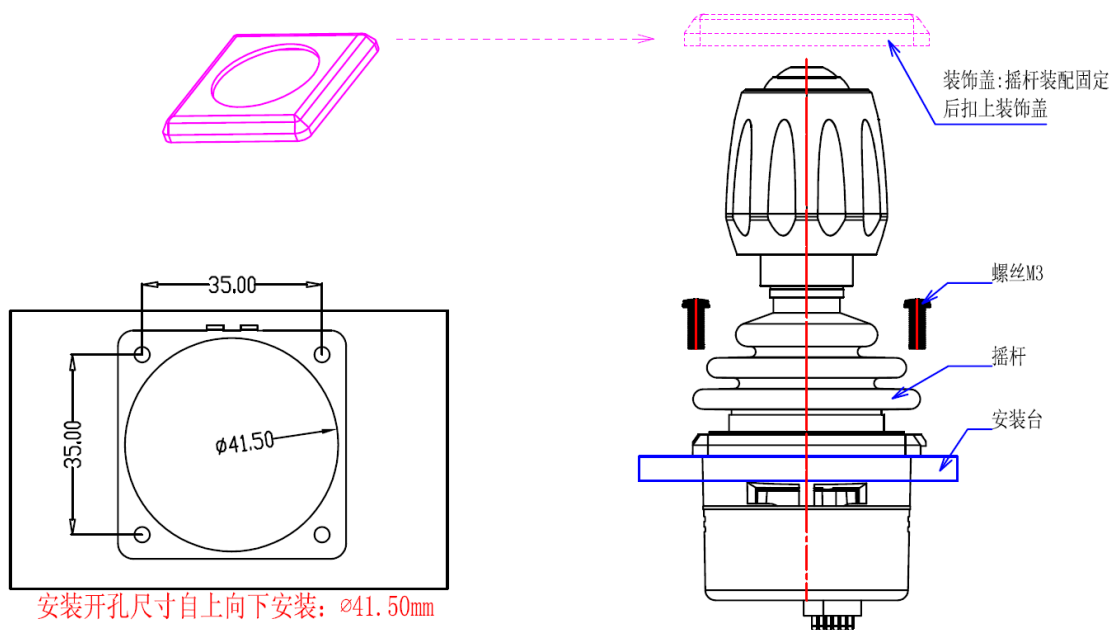
外形尺寸图:



安装开孔尺寸自上向下安装: $\phi 41.50\text{mm}$



安装方式:



深圳市小龙电器有限公司

www.longcctv.com

电话: 0755-29671606

传真: 0755-29671575

EMAIL: XL@LONGCCTV.COM

技术支持: QQ: 4358032