



# 仿人五指灵巧手

## RH56DFTP 系列用户手册

北京因时机器人科技有限公司 | 2024 年 12 月

编号 PRJ-02-TS-U-010 | 共 29 页 | 版本 V1.0.0

# 目 录

<b>1 产品概述 .....</b>	<b>1</b>
1.1 产品特性 .....	1
1.2 性能参数 .....	1
1.3 电气连接 .....	2
1.3.1 引脚定义 .....	2
1.3.2 通信方式 .....	3
<b>2 通信协议 .....</b>	<b>4</b>
2.1 通信协议概要 .....	4
2.2 RS485 读写寄存器操作 .....	4
2.2.1 读灵巧手寄存器的操作 .....	4
2.2.2 写灵巧手寄存器的操作 .....	6
2.3 CAN 读写寄存器操作 .....	9
2.3.1 读寄存器操作 .....	10
2.3.2 写寄存器操作 .....	11
2.4 Modbus RTU 协议概要 .....	12
2.4.1 读取保持寄存器 功能码：0x03 .....	12
2.4.2 预置单寄存器 功能码：0x06 .....	13
2.4.3 预置多寄存器 功能码：0x10 .....	13
2.5 Modbus TCP 协议概要 .....	13
2.5.1 读寄存器-03（0x03）功能码 .....	15
2.5.2 写单个寄存器-06（0x06）功能码 .....	15
2.5.3 写多个寄存器-16（0x10）功能码 .....	16
2.6 寄存器说明 .....	17
2.6.1 灵巧手 ID 号 .....	19
2.6.2 波特率设置 .....	19
2.6.3 清除错误 .....	19
2.6.4 保存参数到 FLASH .....	19
2.6.5 恢复出厂设置 .....	20
2.6.6 力传感器校准 .....	20
2.6.7 目标手势序列号 .....	20
2.6.8 各自由度的上电速度设置值 .....	20
2.6.9 各自由度的上电力控阈值设置值 .....	21
2.6.10 各自由度的执行器位置设置值 .....	21

2.6.11 各自由度的角度设置值 .....	22
2.6.12 各自由度的力控阈值设置值 .....	23
2.6.13 各自由度的速度设置值 .....	24
2.6.14 各自由度的执行器位置实际值 .....	25
2.6.15 各自由度的角度实际值 .....	25
2.6.16 各手指的实际受力 .....	25
2.6.17 各自由度的执行器的电流值 .....	26
2.6.18 各执行器的故障信息 .....	26
2.6.19 各执行器的温度 .....	27
2.6.20 触觉传感器数据格式说明 .....	27



# RH56DFTP 灵巧手

## 用户手册

### 1 产品概述

#### 1.1 产品特性

RH56DFTP 系列灵巧手是一款集成小体积大扭矩直线伺服执行器微型伺服电缸而设计生产的机械灵巧手，以下简称为“灵巧手”。该灵巧手集成了 6 个微型伺服电缸（即执行器），内置灵敏的压力传感器，通过设置不同的力控阈值可实现精准的抓握力控制。手指指尖、指腹、掌心部分集成高精度触觉传感器，能够实时反馈手指抓握信息，用户接口采用网口、RS485 和 CAN2.0 通信接口，简洁高效的接口控制指令可使用户快速实现对灵巧手的操控，优质的性能使该灵巧手应用于服务机器人、教学教具、工业自动化等领域。

#### 1.2 性能参数

关节总数	12 个
自由度	6 个
力传感器数量	6 个
力传感器分辨率	0.1N
触觉传感器数量	5-17 个
指尖重复定位精度	±0.2mm
五指握力	30N
工作电压	24V
静态电流	0.2A @ 24V
空载运动平均电流	1.2A @ 24V
最大握力抓取	4.5A @ 24V
通讯接口	Modbus TCP + CAN2.0 或 Modbus TCP + RS485

## 1.3 电气连接

### 1.3.1. 引脚定义

灵巧手的手腕部分采用的航插接口型号为 **1BZ1G08CLL00000**（兼容雷莫 FGG-1 B-8P 航插接口），实物图片如下图所示，功能定义如下表。

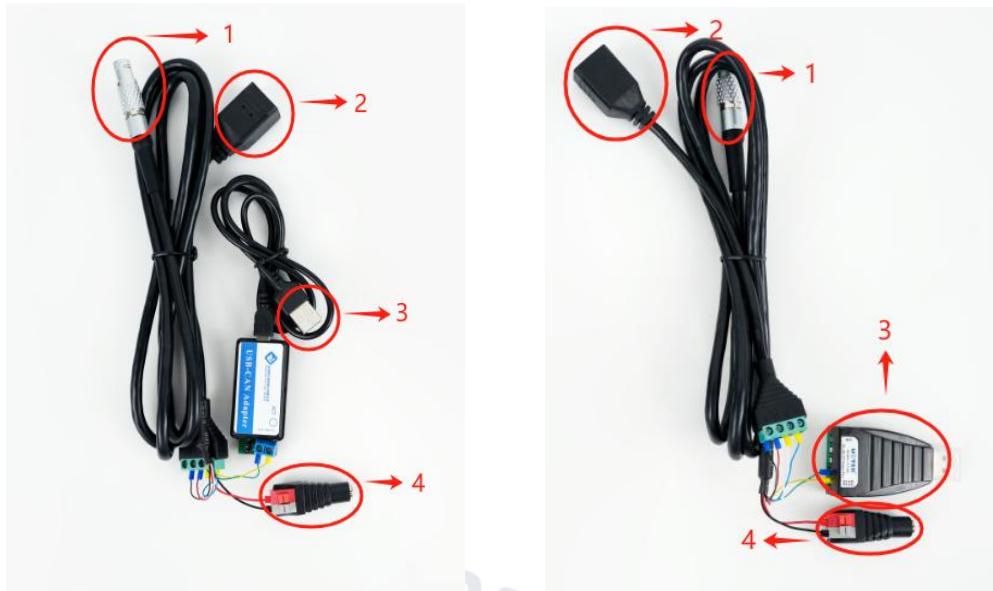
序号	定义
1	GND
2	VCC (24V)
3	485_A/CAN_H
4	485_B/CAN_L
5	TX+ (数据发送正端)
6	TX- (数据发送负端)
7	RX+ (数据接受正端)
8	RX- (数据接受负端)



灵巧手的外接线缆如下图所示，线缆功能定义如下

线缆描述	功能定义
白	TX+ (数据发送正端)
蓝	TX- (数据发送负端)
红 (细)	RX+ (数据接受正端)
黑 (细)	RX- (数据接受负端)
黄	485_A/CAN_H

绿	485_B/CAN_L
红（粗）	VCC (24V)
黑（粗）	GND



线缆接口定义：

1-航插接口，连接灵巧手法兰航插

2-网络接口，连接网线

3-CAN/ RS485 接口

4-电源接口

### 1.3.2. 通信方式

灵巧手使用 **RS485** 接口时，经过一定的转换电路可以实现 **254** 只灵巧手并接在同一条总线上，灵巧手使用 **CAN** 接口时，可以实现多只灵巧手（理论最多 **16383** 只灵巧手）并接在同一条总线上。挂接在同一总线上的灵巧手，需配置为不同的 **ID** 号进行通信控制，灵巧手使用 **Modbus TCP** 控制时，通过 **IP 地址和端口号** 进行控制。（默认 **IP 地址为 192.168.11.210，端口号为 6000**）

灵巧手的通信指令和寄存器对产品用户开放，可通过以上接口与用户的上位机（控制器或通用计算机）连接。用户可以采用 **PC** 或者嵌入式控制器通过以上接口对灵巧手进行参数配置以及运动控制等。

## 2 通信协议

### 2.1 通信协议概要

主控单元通过读写灵巧手内部寄存器来实现对灵巧手的状态获取和控制。

读寄存器是指上位系统读取灵巧手内部寄存器数值（可成组读取，成组的是指若干地址相邻寄存器），上位系统向灵巧手发送读指令（包含寄存器组首地址、寄存器组的长度等内容），灵巧手收到数据并校验成功后，灵巧手将相应的寄存器数据内容返回给上位系统。

写寄存器是指上位系统向灵巧手内部寄存器写入相应的数据（可成组写入），上位系统向灵巧手发送写命令（包含寄存器组首地址、欲写入的数据内容）发送给灵巧手，灵巧手收到数据并校验成功后，返回确认信号给上位系统。

### 2.2 RS485 读写寄存器操作

RS485 通讯参数是 115200bps、8 数据位、1 停止位、无奇偶校验。

#### 2.2.1. 读灵巧手寄存器的操作

读灵巧手寄存器的指令帧格式如下，其中 **Address** 为所要读取寄存器的起始地址，**Hands\_ID** 是灵巧手的 ID 号，**Address\_L** 为 **Address** 的低 8 位，**Address\_H** 为 **Address** 的高 8 位，**Register\_Length** 为所要读寄存器的长度(单位是 byte)，**checksum** 为校验和，是除应答帧头外其余数据的累加和的低字节。

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	0x04	该帧数据部分长度
byte[4]	0x11	读寄存器
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7]	Register_Length	读取寄存器的长度

byte[8]	Checksum	校验和
---------	----------	-----

灵巧手对读寄存器指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	Register_Length+3	该帧数据部分长度
byte[4]	0x11	说明该帧是对读寄存器命令的回复
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	寄存器数值
byte[7+ Register_Length]	Checksum	校验和

举例说明，以读取灵巧手（ID 为 1）各自由度角度实际值为例。自由度角度实际值存储在寄存器组 **ANGLE\_ACT(m)** 中，起始地址为 **1546 (0x060A)**，长度为 **12byte (0x0C)**，那么这条指令帧的格式如下：

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	0x01	灵巧手 ID 号为 1
byte[3]	0x04	该帧数据部分长度
byte[4]	0x11	读寄存器
byte[5]	0x0A	寄存器起始地址低八位

byte[6]	0x06	寄存器起始地址高八位
byte[7]	0x0C	读取寄存器的长度
byte[8]	0x32	校验和

灵巧手对这条指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x0F	该帧数据部分长度 12 + 3
byte[4]	0x11	说明该帧是对读寄存器命令的回复
byte[5]	0x0A	寄存器起始地址低八位
byte[6]	0x06	寄存器起始地址高八位
byte[7] byte[8]	0x64 00	转换为整形 100(0x0064)
byte[9] byte[10]	0x64 00	转换为整形 100(0x0064)
byte[11] byte[12]	0x64 00	转换为整形 100(0x0064)
byte[13] byte[14]	0x64 00	转换为整形 100(0x0064)
byte[15] byte[16]	0xD 007	转换为整形 2000(0x07D0)
byte[17] byte[18]	0x00 00	转换为整形 0(0x0000)
byte[19]	0x98	校验和

从这条回复帧，可以得出自由度角度实际值分别为 100、100、100、100、2000 和 0.

### 2.2.2. 写灵巧手寄存器的操作

写灵巧手寄存器的指令帧格式如下，Data[0]- Data[Register\_Length-1]为所要写入的数据。

	数值	说明

byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	Register_Length+3	该帧数据部分长度
byte[4]	0x12	写寄存器命令标志
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	所要写入寄存器的数据
byte[7+ Register_Length]	checksum	校验和

灵巧手对写寄存器指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	4	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	Address_L	寄存器起始地址低八位
byte[6]	Address_H	寄存器起始地址高八位
byte[7]	1	
byte[8]	checksum	校验和

当进行保存参数操作时（将寄存器 **SAVE** 设置为 1），灵巧手在返回回复帧 1 秒后，将返回保存结果信息帧，这条帧的内容如下：

	数值	说明

byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	Hands_ID	灵巧手 ID 号
byte[3]	4	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	0xED	寄存器 SAVE 起始地址低八位
byte[6]	0x03	寄存器 SAVE 起始地址高八位
byte[7]	**	0x00 保存操作成功，0xFF 保存操作失败
byte[8]	checksum	校验和

举例说明，以设置灵巧手（ID 为 1）各自由度的角度分别为 100、100、100、100、2000 和 0 为例。需要对寄存器组 ANGLE\_SET(m) 进行修改，该寄存器组的起始地址为 1486（0x05CE），长度为 12byte（0x0C），那么需要发送的指令为：

	数值	说明
byte[0]	0xEB	包头
byte[1]	0x90	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x0F	该帧数据部分长度 12 + 3
byte[4]	0x12	写寄存器命令标志
byte[5]	0xCE	寄存器起始地址低八位
byte[6]	0x05	寄存器起始地址高八位
byte[7] byte[8]	0x6400	转换为整形 100(0x0064)
byte[9] byte[10]	0x6400	转换为整形 100(0x0064)
byte[11] byte[12]	0x6400	转换为整形 100(0x0064)
byte[13] byte[14]	0x6400	转换为整形 100(0x0064)
byte[15] byte[16]	0xD007	转换为整形 2000(0x07D0)

byte[17] byte[18]	0x0000	转换为整形 0(0x0000)
byte[19]	0x5C	校验和

灵巧手对这条指令的回复帧如下：

	数值	说明
byte[0]	0x90	包头
byte[1]	0xEB	包头
byte[2]	0x01	灵巧手 ID 号
byte[3]	0x04	该帧数据部分长度
byte[4]	0x12	说明该帧是对写寄存器命令的回复
byte[5]	0xCE	寄存器起始地址低八位
byte[6]	0x05	寄存器起始地址高八位
byte[7]	0x01	
byte[8]	0xEB	校验和

## 2.3 CAN 读写寄存器操作

波特率默认为 1000K，采用扩展标示符、数据帧格式。不使用标准标示符及远程帧。其中扩展标示符共 29 位，从低位至高位定义如下：

**bit0~13:** Hand\_ID 最大支持 16383 个设备。

**bit14~25:** 要操作寄存器的起始地址。

**bit26~28:** 读写标志位，0 表示为读灵巧手寄存器操作，1 表示为写灵巧手寄存器操作，4 表示为读灵巧手手腕寄存器操作，5 表示为写灵巧手手腕寄存器操作。

**Bit29~31:** 预留标志位

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	W/R 0:R 为读灵巧手寄存器操作 1:W 为灵巧手写寄存器操作 4:R 为读灵巧手手腕寄存器操作	寄存器地址 Address	Hand_ID

		5 : W 为写灵巧手手腕寄存器操作		
--	--	--------------------	--	--

### 2.3.1. 读寄存器操作

读寄存器的标识符设置如下：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	0	Address	Hand_ID

数据长度为 **1 Byte**。

数据内容为：预读取寄存器数据的长度。

灵巧手收到以上指令并正确解析后会回复以下帧：

标识符：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	0	Address	Hand_ID

数据长度：返回寄存器数据的长度。

数据内容：寄存器数据。

例如欲读 ID 为 **1** 的灵巧手食指的当前实际角度，应向相 CAN 总线发送一下帧：

标识符：二进制为 **0000 0001 1000 0100 0000 0000 0000 0001**

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	0	ANGLE_ACT(3)的地址为 1552; 二进制为 011000010000	1

数据长度：**1**

数据内容：**2** 食指当前实际角度是一个整型数据，数据长度为 **2byte**。

灵巧手返回帧如下：

标识符：二进制为 **0000 0001 1000 0100 0000 0000 0000 0001**

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	0	ANGLE_ACT(3)的地址为 1552;	1

		二进制为 011000010000	
--	--	-------------------	--

数据长度:2

数据内容如下，食指的当前实际角度 **POS\_ACT(3)** 是整型，需要将下面数据转化为整型（低字节在前，高字节在后），高低字节交换后的 **16** 进制为 **0x01F4**，转化为 **10** 进制为 **500**，即食指的当前实际角度为 **500**。

byte0	byte1
0xF4	0x01

### 2.3.2. 写寄存器操作

写寄存器的标识符设置标识符：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	1	Address	Hand_ID

数据长度：欲写入寄存器内数据的长度。

数据内容：欲写入寄存器内的数据。

灵巧手收到以上指令并正确解析后会回复以下帧：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	1	Address	Hand_ID

数据长度：1

例如欲将 ID 为 1 的灵巧手的食指的角度设置值设置为 600，应向相 CAN 总线发送一下帧：

标识符：二进制为 0000 0101 0111 0101 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	ANGLE_SET(3)的地址为 1492; 二进制为 010111010100	1

数据长度:2

数据内容如下，食指的当前实际角度 **ANGLE\_SET(3)** 是整型，需要整型数据进行高低字节拆分，然后交换高低字节，即 600 (0x0258) 转为以下数据内容。

byte0	byte1
0x58	0x02

灵巧手返回帧如下：

标识符：二进制为 0000 0101 0111 0101 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	ANGLE_SET (3) 的地址为 1492; 二进制为 01 0111 0101 00	1

数据长度:1

## 2.4 Modbus RTU 协议概要

Modbus 协议使用的是主从请求应答通讯方式，协议帧包括了功能码、数据域、和 CRC 校验。支持读取保持寄存器（功能码 0x03）、预置单寄存器（功能码 0x06）、预置多寄存器（功能码 0x10）操作。

### 2.4.1. 读取保持寄存器 功能码：0x03

主站询问 帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	寄存器数量 (高位)	寄存器数量 (低位)	CRC
	0x01	0x03	0x6B	0x00	0x00	0x02	XXXX

含义：读 1 号 (0x01) 号从站保持寄存器，起始地址=0x006B；寄存器个数=0x0002，结束地址=0x006B+2-1=0x006C，即读 17 号从站保持寄存器 0x006B-0x006C，共两个寄存器。

从站应答 帧格式	从站地址	功能码	字节计数	0x006B 寄存器 (高位)	0x006B 寄存器 (低位)	0x006C 寄存器 (高位)	0x006C 寄存器 (低位)	CRC
	0x01	0x03	0x04	0x00	0x01	0x00	0x02	XXXX

含义：返回 1 (0x01) 号从站保持寄存器 0x006B-0x006C，共两个寄存器，0x006B 寄存器数值是 0x0001，0x0062 寄存器数值是 0x0002。

### 2.4.2. 预置单寄存器 功能码：0x06

主站询问 帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	数据内容 (高位)	数据内容量 (低位)	CRC
	0x01	0x06	0x00	0x6B	0x10	0x00	XXXX

含义：设置 1 号（0x01）号从站保持寄存器，寄存器地址 0x006B，数据内容是 0x1000。

从站应答 帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	数据内容 (高位)	数据内容量 (低位)	CRC
	0x01	0x06	0x00	0x00	0x00	0x00	XXXX

### 2.4.3. 预置多寄存器 功能码：0x10

主站 询问 帧格 式	从站 地址	功能码	起始 寄存器 (高位)	起始 寄存器 (低位)	寄存器 数量 (高位)	寄存器 数量 (低位)	字节 计数	数据 (高位)	数据 (低位)	数据 (高位)	数据 (低位)	CRC
		0x10	0x00	0x01	0x00	0x02	0x04	0x00	0x0A	0x01	0x02	XXXX

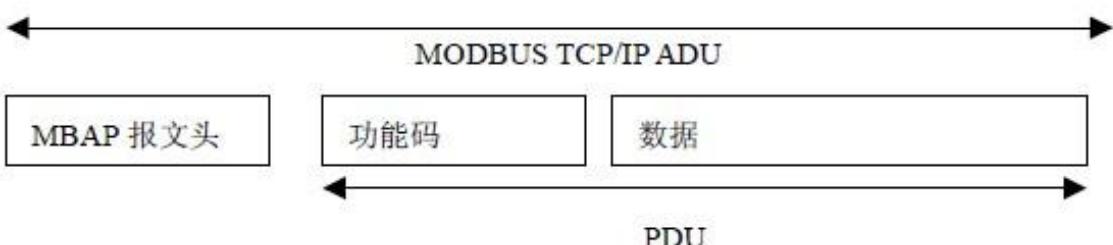
含义：设置 1 号（0x01）号从站保持寄存器，寄存器起始地址 0x0001，寄存器个数为 0x0002，数据内容字节计数是 0x04，数据内容分别是 0x000A、0x0102。

从站应答 帧格式	从站地址	功能码	起始寄存器 (高位)	起始寄存器 (低位)	寄存器数量 (高位)	寄存器数量 (低位)	CRC
	0x01	0x10	0x00	0x01	0x00	0x02	XXXX

## 2.5 Modbus TCP 协议概要

网口转换模块采用 ModbusTCP 协议，此协议是施耐德公司推出的基于以太网 TCP/IP 的 Modbus 协议（是一项应用层报文传输协议，采用 master/slave 方式通讯）。

ModbusTCP 的数据帧可分为两部分：MBAP+PDU。



### MBAP 报文头：

域	长度	描述	客户机	服务器
事务元标识符	2byte	MODBUS 请求/响应事务处理的识别码	客户机启动	服务器从接收的请求中重新复制
协议标识符	2byte	0=MODBUS 协议	客户机启动	服务器从接收的请求中重新复制
长度	2byte	以下字节的数量	客户机启动 (请求)	服务器 (响应) 启动
单元标识符	2byte	串行链路或其它总线上连接的远程从站的识别码	客户机启动	服务器从接收的请求中重新复制

其中长度：包括单位标识符、功能码、数据的字节数；

单元表示符：0xFF,对于 TCP 来说，单元标识符无用 (TCP 通过 IP 来寻址)；

### 帧结构 PDU：

PDU 由功能码+数据组成。功能码为 1 个字节，数据长度不定，由具体功能决定。

Modbus 的功能码有下表所示的 8 种，我们只使用 03、06 和 16 三种功能码。

代码	中文名称	英文名称	位操作/字操作	操作数量
01	读线圈状态	READ COIL STATUS	位操作	单个或多个
02	读离散输入状态	READ INPUT STATUS	位操作	单个或多个
03	读保持寄存器	READ HOLDING REGISTER	字操作	单个或多个
04	读输入寄存器	READ INPUT REGISTER	字操作	单个或多个
05	写线圈状态	WRITE SINGLE COIL	位操作	单个
06	写单个保持寄存器	WRITE SINGLE REGISTER	字操作	单个
15	写多个线圈	WRITE MULTIPLE COIL	位操作	多个
16	写多个保持寄存器	WRITE MULTIPLE REGISTER	字操作	多个

### 2.5.1. 读寄存器-03 (0x03) 功能码

#### 请求

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
寄存器数量	N*2个字节	

例子：这是一个请求读保持寄存器 108-110 的实例。将寄存器 108 的内容表示为两个十六进制字节值 02 2B，或十进制 555。将寄存器 109-110 的内容分别表示为十六进制 00 00 和 00 64，或十进制 0 和 100。

请求		响应	
域名	十六进制	域名	十六进制
功能	03	功能	03
高起始地址	00	字节数	06
低起始地址	6B	寄存器值 Hi (108)	02
高寄存器编号	00	寄存器值 Lo (108)	2B
低寄存器编号	03	寄存器值 Hi (109)	00
		寄存器值 Lo (109)	00
		寄存器值 Hi (110)	00
		寄存器值 Lo (110)	64

### 2.5.2. 写单个寄存器-06 (0x06) 功能码

#### 请求

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	N*2个字节	0x0000 至 0xFFFF

例子：这是一个请求将十六进制 00 03 写入寄存器 2 的实例

请求		响应	
域名	十六进制	域名	十六进制
功能	06	功能	06
寄存器地址 Hi	00	输出地址 Hi	00
寄存器地址 Lo	01	输出地址 Lo	01
寄存器值 Hi	00	输出值 Hi	00
寄存器值 Lo	03	输出值 Lo	03

### 2.5.3. 写多个寄存器-16 (0x10) 功能码

#### 请求

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	

#### 响应

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)

例子：这是一个请求将十六进制 00 0A 和 01 02 写入以 2 开始的两个寄存器的实例

请求		响应	
域名	十六进制	域名	十六进制
功能	10	功能	10
高起始地址	00	高起始地址	00
低起始地址	01	低起始地址	01
寄存器数量 Hi	00	寄存器数量 Hi	00
寄存器数量 Lo	02	寄存器数量 Lo	02
字节数	04	寄存器值 00 0A 01 02	
寄存器值 Hi	00		
寄存器值 Lo	0A		
寄存器值 Hi	01		
寄存器值 Lo	02		

## 2.6 寄存器说明

灵巧手对用户开放的寄存器参数如下：

地址	含义	缩写	长度	读写权限
1000	灵巧手 ID	HAND_ID	1byte	W/R
1002	波特率设置	REDU_RATIO	1byte	W/R

1004	清除错误	CLEAR_ERROR	1byte	W/R
1005	保存数据至 Flash	SAVE	1byte	W/R
1006	恢复出厂设置	RESET_PARA	1byte	W/R
1009	受力传感器校准	GESTURE_FORCE_CLB	1byte	W/R
1032	各自由度的上电速度设置值	DEFAULT_SPEED_SET(m)	6short(12byte)	W/R
1044	各自由度的上电力控阈值设置值	DEFAULT_FORCE_SET(m)	6short(12byte)	W/R
1474	各自由度的执行器位置设置值	POS_SET(m)	6short(12byte)	W/R
1486	各自由度的角度设置值	ANGLE_SET(m)	6short(12byte)	W/R
1498	各自由度的力控阈值设置值	FORCE_SET(m)	6short(12byte)	W/R
1522	各自由度的速度设置值	SPEED_SET(m)	6short(12byte)	W/R
1534	各自由度的执行器位置实际值	POS_ACT(m)	6short(12byte)	R
1546	各自由度的角度实际值	ANGLE_ACT(m)	6short(12byte)	R
1582	各手指的实际受力	FORCE_ACT(m)	6short(12byte)	R
1594	各自由度的执行器的电流值	CURRENT(m)	6short(12byte)	R
1606	各自由度的执行器的故障信息	ERROR(m)	6byte	R
1612	各自由度的状态信息	STATUS(m)	6byte	R
1618	各自由度的执行器的温度	TEMP(m)	6byte	R
1700	IP 地址第 1 字段， 默认 192, 范围 0-255， 重新上电生效	IP_PART1	1byte	W/R
1701	IP 地址第 2 字段， 默认 168, 范围 0-255， 重新上电生效	IP_PART2	1byte	W/R
1702	IP 地址第 3 字段， 默认 11, 范围 0-255， 重新上电生效	IP_PART3	1byte	W/R
1703	IP 地址第 4 字段， 默认 210, 范围 0-255， 重新上电生效	IP_PART4	1byte	W/R

3000	小拇指触觉传感器	FINGERONE_TOUCH	370byte	R
3370	无名指触觉传感器	FINGERTWO_TOUCH	370byte	R
3740	中指触觉传感器	FINGERTHE_TOUCH	370byte	R
4110	食指触觉传感器	FINGERFOR_TOUCH	370byte	R
4480	大拇指触觉传感器	FINGERFIV_TOUCH	420byte	R
4900	手掌触觉传感器	FINGERPALM_TOUCH	224byte	R

### 2.6.1. 灵巧手 ID 号

默认值 1，范围：1-254,可保存。

当一条总线上连接多只灵巧手时，需要给每一只灵巧手分配不同 ID 号。

### 2.6.2. 波特率设置

RS485 接口：默认值 0 范围 0-3 可保存

0:波特率 115200

1: 波特率 57600

2: 波特率 19200

3: 波特率 921600

CAN 接口：默认值 0 范围 0-1 可保存

0: 波特率 1000K

1: 波特率 500K

### 2.6.3. 清除错误

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手将清除可清除的故障(执行器的堵转故障、过流故障、异常故障以及通讯故障)。

注意：执行器的过温故障是不可清除故障，当温度回落后，故障会自动清除。

### 2.6.4. 保存参数到 FLASH

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手将当前参数写入 **flash**，断电后参数不丢失。

#### 2.6.5. 恢复出厂设置

默认值 0 范围 0-1 不可保存

写入 1 后，灵巧手的参数将恢复为出厂设置参数。

#### 2.6.6. 力传感器校准

当用户设置参数为 1 时，灵巧手启动力传感器校准过程。

**注意：校准过程必须要保证灵巧手处于手掌张开状态，并且手指不能接触任何物体。**

#### 2.6.7. 目标手势序列号

默认值 0xFF 范围 1-40 不可保存。

当用户修改此寄存器数值时，灵巧手会做出与序列号相应手势

1-13 号手势是固定手势（产品出厂前已保存），分别是张开，二指捏，握拳等；

14-40 号手势是用户自定手势（数据存储在 **USER\_DEF\_ANGLE (K, m)** 寄存器中），可以被用户修改并保存。

#### 2.6.8. 各自由度的上电速度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的上电速度设置值，具体说明见下表。这些参数可断电保存。

地址	名称	说明	数据类型	范围
1032-1033	DEFAULT_SPEED_SET(0)	小拇指上电初始速度	short	0-1000
1034-1035	DEFAULT_SPEED_SET(1)	无名指上电初始速度	short	0-1000
1036-1037	DEFAULT_SPEED_SET(2)	中指上电初始速度	short	0-1000
1038-1039	DEFAULT_SPEED_SET(3)	食指上电初始速度	short	0-1000
1040-1041	DEFAULT_SPEED_SET(4)	大拇指弯曲上电初始速度	short	0-1000
1042-1043	DEFAULT_SPEED_SET(5)	大拇指旋转上电初始速度	short	0-1000

速度为 1000：表示指手指在空载时从最大角度运动到最小角度用时 600ms，如果负载较大时，这个实际速度会有一定程度的降低。

### 2.6.9. 各自由度的上电力控阈值设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的上电力控阈值设置值，具体说明见下表。这些参数可断电保存。

地址	名称	说明	数据类型	范围
1044-1045	DEFAULT_FORCE_SET(0)	小拇指上电初始力控	short	0-3000
1046-1047	DEFAULT_FORCE_SET(1)	无名指上电初始力控	short	0-3000
1048-1049	DEFAULT_FORCE_SET(2)	中指上电初始力控	short	0-3000
1050-1051	DEFAULT_FORCE_SET(3)	食指上电初始力控	short	0-3000
1052-1053	DEFAULT_FORCE_SET(4)	大拇指弯曲上电初始力控	short	0-3000
1054-1055	DEFAULT_FORCE_SET(5)	大拇指旋转上电初始力控	short	0-3000

寄存器数值含义是指相应手指指尖提供的握力，例如 **DEFAULT\_FORCE\_SET(1)** 设置为 **800** 时，表示允许无名指手指指尖提供 **800g** 的握力。（如果手指接触物体的部分不是指尖，那么可提供的握力则有所不同，具体大小与力臂长度有关，力臂越短，提供的握力越大）。

### 2.6.10. 各自由度的执行器位置设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度执行器的设置位置，具体说明见下表。这些参数不能保存。

地址	名称	说明	数据类型	范围
1474-1475	POS_SET(0)	小拇指执行器位置设置	short	0-2000
1476-1477	POS_SET(1)	无名指执行器位置设置	short	0-2000
1478-1479	POS_SET(2)	中指执行器位置设置	short	0-2000
1480-1481	POS_SET(3)	食指执行器位置设置	short	0-2000
1482-1483	POS_SET(4)	大拇指弯曲执行器位置设置	short	0-2000
1484-1485	POS_SET(5)	大拇指旋转执行器位置设置	short	0-2000

**0** 表示：执行器最短行程，对应手指角度的最大值（即手指张开）。

**2000** 表示：执行器最大行程，对应手指角度的最小值（即手指弯曲）。

-1 表示：执行器不执行任何动作。

**不建议用户通过设置这组寄存器来设置给手指的位置角度。**

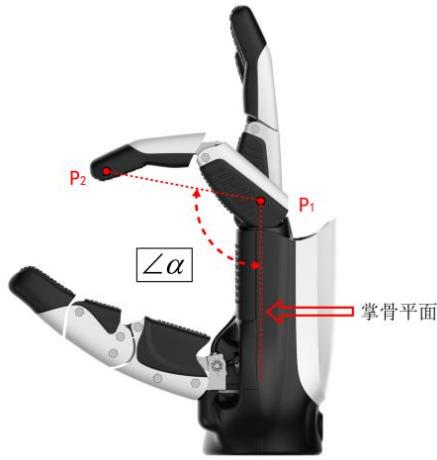
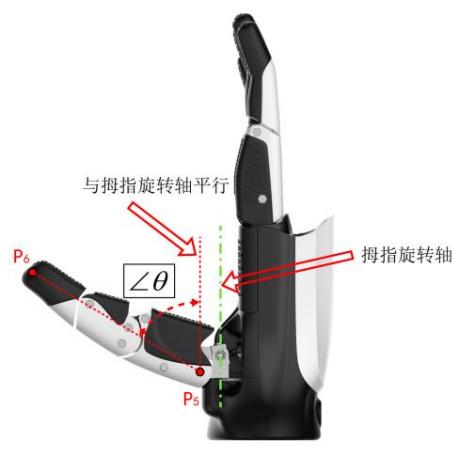
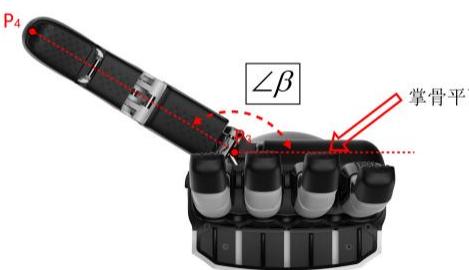
### 2.6.11. 各自由度的角度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的角度设置值，具体说明见下表。当某一自由度的角度设置值被设置为 0-1000 范围内的数据时，相应的手指会立即动作，如果角度设置值被设置为 -1 时，相应的手指无反应，例如将 ANGLE\_SET(0)-ANGLE\_SET(5)6 个寄存器设置为 500,500,-1,0,500,500,后小拇指、无名指、大拇指弯曲和大拇指旋转 4 个自由度会运动到 500 的角度，食指运动到 0 的角度，而中指将不会动作（即保持当前实际位置不动作）。

地址	名称	说明	数据类型	范围
1486-1487	ANGLE_SET(0)	小拇指上电初始角度	short	-1, 0-1000
1488-1489	ANGLE_SET(1)	无名指上电初始角度	short	-1, 0-1000
1490-1491	ANGLE_SET(2)	中指上电初始角度	short	-1, 0-1000
1492-1493	ANGLE_SET(3)	食指上电初始角度	short	-1, 0-1000
1494-1495	ANGLE_SET(4)	大拇指弯曲上电初始角度	short	-1, 0-1000
1496-1497	ANGLE_SET(5)	大拇指旋转上电初始角度	short	-1, 0-1000

各自由度的角度定义和运动范围说明如下。

角度	图例说明	范围
小拇指 无名指 中指 食指		20°~176°

		
大拇指弯曲角度		-13°~70°
大拇指旋转角度		90°~165°

### 2.6.12. 各自由度的力控阈值设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度力控阈值设置值，具体说明见下表。用户可设置这组寄存器来实现对手指握力的控制，例如将 **FORCE\_SET(3)**

设置为 300，当前食指实际角度 **ANGLE\_ACT(3)** 为 1000（即完全张开状态），当用户将食指设置角度 **ANGLE\_SET(3)** 设置为 0 后，食指将在向手掌弯曲运动，在弯曲过程中如果检测到食指实际受力（即 **FORCE\_ACT(3)**）到达 300 后，食指停止运动，如果食指实际受力没有到达 300，那么食指将运动到设置角度。

地址	名称	说明	数据类型	范围
1498-1499	<b>FORCE_SET(0)</b>	小拇指力控设置值	short	0-3000
1500-1501	<b>FORCE_SET(1)</b>	无名指力控设置值	short	0-3000
1502-1503	<b>FORCE_SET(2)</b>	中指力控设置值	short	0-3000
1504-1505	<b>FORCE_SET(3)</b>	食指力控设置值	short	0-3000
1506-1507	<b>FORCE_SET(4)</b>	大拇指弯曲力控设置值	short	0-3000
1508-1509	<b>FORCE_SET(5)</b>	大拇指旋转力控设置值	short	0-3000

寄存器数值含义是指相应手指指尖提供的握力，例如 **DEFAULT\_FORCE\_SET(1)** 设置为 800 时，表示允许无名指手指指尖提供 800g 的握力。（如果手指接触物体的部分不是指尖，那么可提供的握力则有所不同，具体大小与力臂长度有关，力臂越短，提供的握力越大）。

### 2.6.13. 各自由度的速度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的速度设置值，具体说明见下表。这些参数可断电保存。

地址	名称	说明	数据类型	范围
1522-1523	<b>SPEED_SET(0)</b>	小拇指速度设置值	short	0-1000
1524-1525	<b>SPEED_SET(1)</b>	无名指速度设置值	short	0-1000
1526-1527	<b>SPEED_SET(2)</b>	中指速度设置值	short	0-1000
1528-1529	<b>SPEED_SET(3)</b>	食指速度设置值	short	0-1000
1530-1531	<b>SPEED_SET(4)</b>	大拇指弯曲速度设置值	short	0-1000
1532-1533	<b>SPEED_SET(5)</b>	大拇指旋转速度设置值	short	0-1000

速度为 1000：表示指手指在空载时从大角度运动到最小角度用时 600ms，如果负载较大时，这个实际速度会有一定程度的降低。

### 2.6.14. 各自由度的执行器位置实际值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度执行器的当前实际位置，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型	范围
1534-1535	POS_ACT(0)	小拇指执行器位置实际值	short	0-2000
1536-1537	POS_ACT(1)	无名指执行器位置实际值	short	0-2000
1538-1539	POS_ACT(2)	中指执行器位置实际值	short	0-2000
1540-1541	POS_ACT(3)	食指执行器位置实际值	short	0-2000
1542-1543	POS_ACT(4)	大拇指弯曲执行器位置实际值	short	0-2000
1544-1545	POS_ACT(5)	大拇指旋转执行器位置实际值	short	0-2000

0 表示：执行器最短行程，对应手指角度的最大值。

2000 表示：执行器最大行程，对应手指角度的最小值。

### 2.6.15. 各自由度的角度实际值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的角度实际值，具体说明见下表。各自由度的角度定义可以参考 **1.7 DEFAULT\_ANGLE\_SET(m)** 寄存器组的说明。这些参数为只读参数。

地址	名称	说明	数据类型	范围
1546-1547	ANGLE_ACT(0)	小拇指角度实际值	short	0-1000
1548-1549	ANGLE_ACT(1)	无名指角度实际值	short	0-1000
1550-1551	ANGLE_ACT(2)	中指角度实际值	short	0-1000
1552-1553	ANGLE_ACT(3)	食指角度实际值	short	0-1000
1554-1555	ANGLE_ACT(4)	大拇指弯曲角度实际值	short	0-1000
1556-1557	ANGLE_ACT(5)	大拇指旋转角度实际值	short	0-1000

### 2.6.16. 各手指的实际受力

此寄存器组由六个寄存器组成，分别对应灵巧手六个手指的实际受力值，单位：**g**，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型	范围
1582-1583	FORCE_ACT(0)	小拇指实际受力值	short	-4000-4000
1584-1585	FORCE_ACT(1)	无名指实际受力值	short	-4000-4000
1586-1587	FORCE_ACT(2)	中指实际受力值	short	-4000-4000
1588-1589	FORCE_ACT(3)	食指实际受力值	short	-4000-4000
1590-1591	FORCE_ACT(4)	大拇指弯实际受力值	short	-4000-4000
1592-1593	FORCE_ACT(5)	大拇指旋转实际受力值	short	-4000-4000

### 2.6.17. 各自由度的执行器的电流值

此寄存器组由六个寄存器组成，分别对应灵巧手六个执行器的电流值，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型	范围	单位
1594-1595	CURRENT(0)	小拇指执行器电流值	short	0-2000	mA
1596-1597	CURRENT(1)	无名指执行器电流值	short	0-2000	mA
1598-1599	CURRENT(2)	中指执行器电流值	short	0-2000	mA
1600-1601	CURRENT(3)	食指执行器电流值	short	0-2000	mA
1602-1603	CURRENT(4)	大拇指弯曲执行器电流值	short	0-2000	mA
1604-1605	CURRENT(5)	大拇指旋转执行器电流值	short	0-2000	mA

### 2.6.18. 各执行器的故障信息

此寄存器组由六个寄存器组成，分别对应灵巧手六个执行器的故障码，具体说明见下表。这些参数为只读参数。

地址	名称	说明	数据类型
1606	ERROR(0)	小拇指执行器故障码	byte
1607	ERROR (1)	无名指执行器故障码	byte
1608	ERROR (2)	中指执行器故障码	byte
1609	ERROR (3)	食指执行器故障码	byte

1610	<b>ERROR (4)</b>	大拇指弯曲执行器故障码	byte
1611	<b>ERROR (5)</b>	大拇指旋转执行器故障码	byte

故障码含义如下表所示，例如 **ERROR (3)** 为 **0x06** (转换为二进制为 **00000110**) 时，说明食指执行器出现过温故障和过流故障。

	说明
Bit0	堵转故障
Bit1	过温故障
Bit2	过流故障
Bit3	电机异常
Bit4	通讯故障

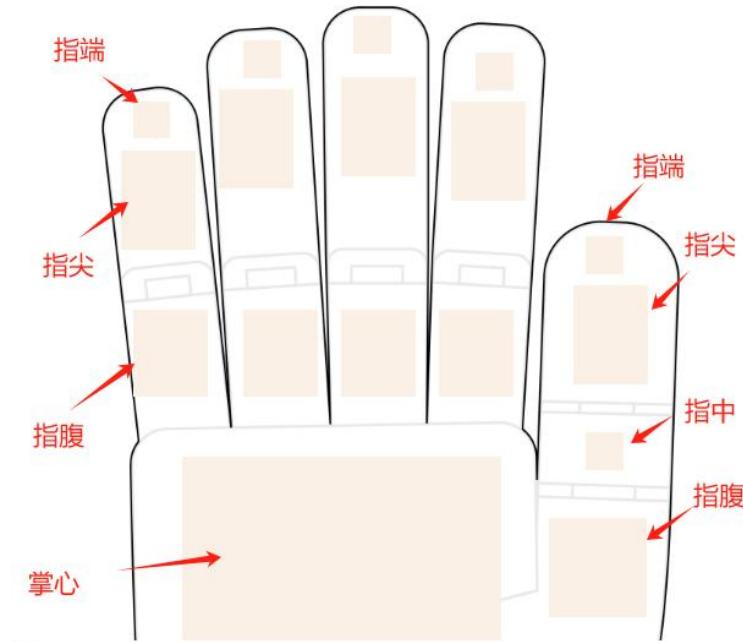
### 2.6.19. 各执行器的温度

地址	名称	说明	数据类型	范围	单位
1618	TEMP(0)	小拇指执行器温度值	byte	0-100	°C
1619	TEMP(1)	无名指执行器温度值	byte	0-100	°C
1620	TEMP(2)	中指执行器温度值	byte	0-100	°C
1621	TEMP(3)	食指执行器温度值	byte	0-100	°C
1622	TEMP(4)	大拇指弯曲执行器温度值	byte	0-100	°C
1623	TEMP(5)	大拇指旋转执行器温度值	byte	0-100	°C

此寄存器组由六个寄存器组成，分别对应灵巧手六个执行器温度值，具体说明见下表。这些参数为只读参数。

### 2.6.20. 触觉传感器数据格式说明

灵巧手触觉分布如下图所示（淡黄色部分为触觉传感器所在位置）。



地址	含义	行列数	长度	读写权限
3000-3017	小拇指指端触觉数据	3*3	18byte	R
3018-3209	小拇指指尖触觉数据	12*8	192byte	R
3210-3369	小拇指指腹触觉数据	10*8	160byte	R
3370-3387	无名指指端触觉数据	3*3	18byte	R
3388-3579	无名指指尖触觉数据	12*8	192byte	R
3580-3739	无名指指腹触觉数据	10*8	160byte	R
3740-3757	中指端触觉数据	3*3	18byte	R
3758-3949	中指指尖触觉数据	12*8	192byte	R
3950-4109	中指指腹触觉数据	10*8	160byte	R
4110-4127	食指指端触觉数据	3*3	18byte	R
4128-4319	食指指尖触觉数据	12*8	192byte	R
4320-4479	食指指腹触觉数据	10*8	160byte	R
4480-4497	大拇指指端触觉数据	3*3	18byte	R
4498-4689	大拇指指尖触觉数据	12*8	192byte	R

4690-4707	大拇指指中触觉数据	3*3	18byte	R
4708-4899	大拇指指腹触觉数据	12*8	192byte	R
4900-5123	掌心触觉数据	8*14	224byte	R

以上数据中每一个触觉点的数值为 **16** 位整型数据（两个 **byte** 数据组成，采用小端模式，低位在前，高位在后），数值范围为 **0-4096**。

五指部分数据点对应实际位置关系为：数据点 **1** 对应第一行，第一列；数据点 **2** 对应第一行第二列；数据点 **3** 对应第一行，第三列；数据点 **4** 对应第一行，第三列；数据点 **5** 对应第一行，第三列；数据点 **6** 对应第一行，第三列；数据点 **7** 对应第一行，第三列；数据点 **8** 对应第一行，第三列；数据点 **9** 对应第二行，第一列；数据点 **10** 对应第二行，第一列；以此类推。

手掌部分数据点对应实际位置关系为：数据点 **1** 对应第八行，第一列，数据点 **2** 对应第七行，第一列；数据点 **3** 对应第六行，第一列；数据点 **4** 对应第五行，第一列；数据点 **5** 对应第四行，第一列；数据点 **6** 对应第三行，第一列；数据点 **7** 对应第二行，第一列；数据点 **8** 对应第一行，第一列；数据点 **9** 对应第八行，第二列；数据点 **10** 对应第七行，第二列；以此类推。