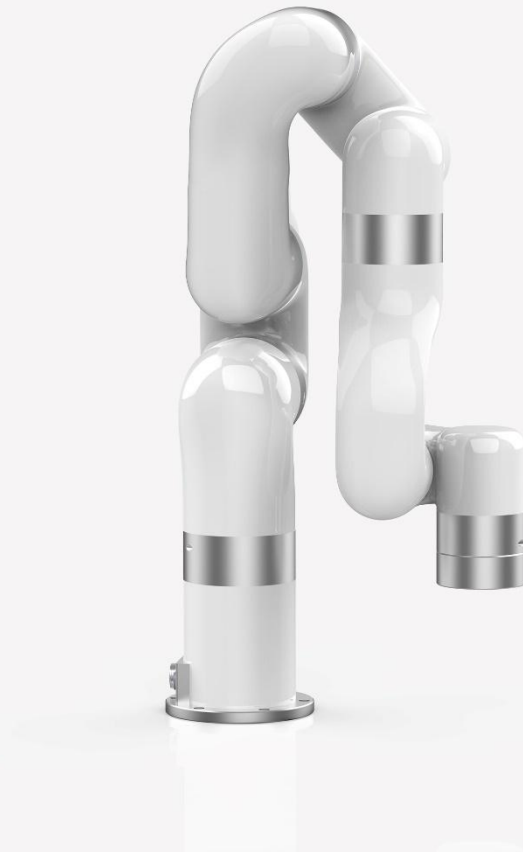




# UFACTORY XARM

开发者手册



使用前请仔细阅读本手册

V 2.0.1

# 目录

1. 前言 .....	4
1.1. 使用须知 .....	4
1.2. 手册主要内容 .....	4
1.3. 机械臂运动参数 .....	4
1.4. 单位使用说明 .....	5
1.5. 术语与定义 .....	5
1.6. 其他开发者资源 .....	5
1.7. 更多信息 .....	8
2. xArm 通信协议 .....	9
2.1. 控制器通信协议 .....	9
2.1.1. 符号意义 .....	9
2.1.2. Modbus-TCP 通信格式 .....	9
2.1.3. 寄存器（机械臂控制） .....	11
2.1.4. 寄存器（机械臂外设控制） .....	69
2.1.5. Modbus TCP 示例 .....	90
2.1.6. 自动上报数据格式 .....	107
3. 报错及处理 .....	111
3.1. 关节报警信息和常规处理方式 .....	111
3.2. 控制器报警信息及常规处理方式 .....	111
3.2.1. 控制器错误代码 .....	112
3.2.2. 控制器警告代码 .....	112
3.3. 机械爪报警信息及常规处理方式 .....	114
4. 技术规格 .....	116
4.1. xArm5/6/7 通用规格 .....	116
4.2. xArm 5 规格 .....	118

4.3. xArm 6 规格 .....	119
4.4. xArm 7 规格 .....	120

# 1. 前言

## 1.1. 使用须知

- (1) 本手册主要面向使用 xArm Modbus-TCP 通信协议进行开发的开发者。如果您使用 Python (C++ 或者 ROS) 进行应用开发, 请参考“1.6. 其他开发者资源”; 如果您使用 xArm Studio 图形化编程工具进行应用开发, 请参考“xArm 用户手册”。
- (2) 考虑到您使用 xArm Modbus-TCP 通信协议进行应用开发可能面临的风险, 我们希望在阅读本手册前, 已经仔细阅读并理解“xArm 用户手册”的所有内容, 熟悉 xArm 的风险评估, 熟悉 xArm 的运动规划, 熟练使用“xArm Studio”来设置机械臂各项参数和对机械臂进行编程。如果您尚不满足上述条件, 我们强烈建议您通过参考“xArm 用户手册”, 通过 xArm Studio 软件进行实际操作, 直至您满足上述条件后, 再开始使用 xArm Modbus-TCP 通信协议进行开发。我们认为, 这不仅能降低您使用 xArm Modbus-TCP 通信协议进行应用开发所面临的风险, 还能提高您应用开发的效率。

## 1.2. 手册主要内容

- (1) [xArm 运动特性](#)
- (2) [xArm 通讯协议](#)
- (3) [xArm 报错和处理](#)
- (4) [xArm 技术规格](#)

## 1.3. 机械臂运动参数

机械臂的运动参数见表 1.1 和表 1.2。

表 1.1 机械臂各关节工作范围

	机械臂	xArm 5	xArm 6	xArm 7
最大速度		180° /s	180° /s	180° /s
工作范围	轴 1	±360°	±360°	±360°
	轴 2	-118° ~120°	-118° ~120°	-118° ~120°
	轴 3	-225° ~11°	-225° ~11°	±360°

	轴 4	-97° ~180°	±360°	-11° ~225°
	轴 5	±360°	-97° ~180°	±360°
	轴 6	无	±360°	-97° ~180°
	轴 7	无	无	±360°

表 1.2 机械臂各运动参数范围

	TCP 运动	Joint 运动
速度 (speed)	0~1000mm/s	0~180° /s
加速度 (acc)	0~50000mm/s <sup>2</sup>	0~1145° /s <sup>2</sup>
加加速度 (jerk)	0~100000mm/s <sup>3</sup>	0~28647° /s <sup>3</sup>

注：（1）在 TCP 运动（即笛卡尔空间运动）指令（SDK 的 set\_position() 函数）中，如果同时包含位置变化和姿态变化，一般情况下姿态旋转速度由系统自动算出。此时指定的速度参数为最大位置线速度，范围为：0~1000mm/s。

（2）当期望的 TCP 运动仅限于姿态 (roll, pitch, yaw) 变化，而位置(x, y, z)保持不变时，此时指定的速度参数为姿态旋转速度，所以范围 0~1000mm/s 对应 0~180° /s。

## 1.4. 单位使用说明

该手册中 Python/Blockly 示例及通信协议中使用的单位见表 1.3。

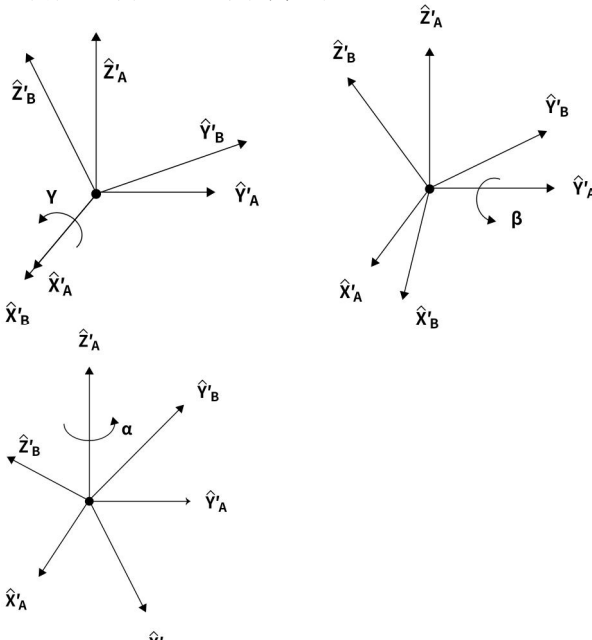
表 1.3. Python/Blockly 示例及通信协议中各参数单位

参数	Python-SDK	Blockly	通信协议
X (Y/Z)	毫米 (mm)	毫米 (mm)	毫米 (mm)
Roll (Pitch/Yaw)	度 (°)	度 (°)	弧度 (rad)
J <sub>1</sub> (J <sub>2</sub> /J <sub>3</sub> /J <sub>4</sub> /J <sub>5</sub> /J <sub>6</sub> /J <sub>7</sub> )	度 (°)	度 (°)	弧度 (rad)
TCP 速度	毫米/秒 (mm/s)	毫米/秒 (mm/s)	毫米/秒 (mm/s)
TCP 加速度	毫米/秒 <sup>2</sup> (mm/s <sup>2</sup> )	毫米/秒 <sup>2</sup> (mm/s <sup>2</sup> )	毫米/秒 <sup>2</sup> (mm/s <sup>2</sup> )
TCP 加加速度	毫米/秒 <sup>3</sup> (mm/s <sup>3</sup> )	毫米/秒 <sup>3</sup> (mm/s <sup>3</sup> )	毫米/秒 <sup>3</sup> (mm/s <sup>3</sup> )
关节速度	度/秒 (° /s)	度/秒 (° /s)	弧度/秒 (rad/s)
关节加速度	度/秒 <sup>2</sup> (° /s <sup>2</sup> )	度/秒 <sup>2</sup> (° /s <sup>2</sup> )	弧度/秒 <sup>2</sup> (rad/s <sup>2</sup> )
关节加加速度	度/秒 <sup>3</sup> (° /s <sup>3</sup> )	度/秒 <sup>3</sup> (° /s <sup>3</sup> )	弧度/秒 <sup>3</sup> (rad/s <sup>3</sup> )

## 1.5. 术语与定义

下列术语和定义适用于本手册。

术语	定义
控制器	为机械臂的核心部分，它是机械臂控制系统的集成。
末端执行器	末端执行器安装在机械臂手腕的前端，用来安装夹持器和专用工具（如机械

	爪、真空吸头等），可以直接执行工作任务。
使能机械臂	给机械臂上电，且开启机械臂电机，机械臂使能后，可正常开始运动。
TCP	工具中心点。（未设置末端执行器偏移时，为法兰盘中心）
TCP 运动	目标位置为笛卡尔空间坐标点的运动，末端在运动中遵循指定的轨迹（圆弧，直线等）。
TCP 负载 (末端负载)	负载重量是指实际的（末端执行器+托运外物）的重量，单位是 kg；X/Y/Z 轴表示 TCP 的重心相对于默认工具坐标系（位于法兰中心）的位置，单位是 mm。
TCP 偏移 (末端执行器偏移)	设置 TCP（末端执行器）坐标系与定义在法兰中心的工具坐标系之间的相对偏移量，单位是 mm。
Roll/Pitch/Yaw	<p>Roll /Pitch/Yaw 按顺序依次绕选定坐标系（基坐标系）的 X/Y/Z 旋转。 下面举例描述坐标系 {B} 姿态的一种方法： 例如首先将坐标系 {B} 和一个已知参考坐标系 {A} 重合。先将 {B} 绕 <math>\hat{X}_A</math> 旋转 <math>\gamma</math>，再绕 <math>\hat{Y}_A</math> 旋转 <math>\beta</math>，最后绕 <math>\hat{Z}_A</math> 旋转 <math>\alpha</math>。 每个旋转都是绕着固定的参考坐标系 {A} 的轴，这种方法叫 XYZ 固定角坐标系，有时把他们定义为回转角 (roll)、俯仰角 (pitch) 和偏转角 (yaw)。 上面描述的就是 XYZ 欧拉角，旋转过程如下图所示：</p>  <p>等价旋转矩阵为：  <math display="block">{}^A_B R_{XYZ}(\gamma, \beta, \alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)</math> 注： <math>\gamma</math> 对应 roll； <math>\beta</math> 对应 pitch； <math>\alpha</math> 对应 yaw。</p>
轴角	<p>Rx/Ry/Rz 与 Roll/Pitch/Yaw 一样，使用 3 个值表示姿态，但不是三个旋转角度，而是一个三维旋转向量 <math>[x, y, z]</math> 和一个旋转角度 <math>\phi</math>（标量）的乘积。 轴角表示的性质： 假设旋转轴为 <math>[x, y, z]</math>，旋转角度为 <math>\phi</math>。则轴角表示即为 <math>[R_x, R_y, R_z] = [x*\phi, y*\phi, z*\phi]</math>，其中 <math>[x, y, z]</math> 为单位向量，<math>\phi</math> 为非负值，因而 <math>[R_x, R_y, R_z]</math> 的向量长度(模)即可推算旋转角度，向量方向即为旋转方向。 如果想表示逆向旋转，则将旋转向量 <math>[x, y, z]</math> 取反，<math>\phi</math> 值不变。 使用 <math>\phi</math> 和 <math>[x, y, z]</math> 同样可以推导出单位四元数的姿态表示 <math>q = [\cos(\phi/2), \sin(\phi/2)*x, \sin(\phi/2)*y, \sin(\phi/2)*z]</math>。 举例： 当前 TCP 坐标系的姿态是基坐标系围绕某个空间向量旋转某个角度得到的。比如用基坐标系表示的旋转轴的向量为 <math>[1, 0, 0]</math>，旋转角度为 180 度 (<math>\pi</math> 弧度)，则这个姿态的轴角表示即为 <math>[\pi, 0, 0]</math>。如果旋转轴为 <math>[0.707, 0.707, 0]</math>，旋转角度为 90 度 (<math>\pi/2</math> 弧度)，则轴角姿态为</p>

	$[0.707 * (\pi/2), 0.707 * (\pi/2), 0]$ 。
基坐标系 (参考图 1)	基坐标系是以机器人安装基座为基准、用来描述机器人本体运动的笛卡尔坐标系。 任何机器人都离不开基坐标系，也是机器人 TCP 在三维空间运动所必须的基本坐标系（面对机器人前后：X 轴，左右：Y 轴，上下：Z 轴）
工具坐标系 (参考图 1)	由工具中心点 TCP 与坐标方位组成。 如果没有设置 TCP 偏移，那么默认工具坐标系位于法兰中心。 是以工具中心点作为零点，机器人的轨迹参照工具坐标系。
用户坐标系 (参考图 1)	用户坐标系可定义为机器人运动范围内的任意位置，设定任意角度的 X、Y、Z 轴，坐标系的方向根据客户需要任意定义。
手动模式	即示教模式或力矩模式，在该模式下，操作人员可直接手动控制机械臂。
示教灵敏度	示教灵敏度范围 1~5 个等级。设定的指越大，示教灵敏度等级越高，开启示教模式拖拽关节所需的力越小。
碰撞灵敏度	碰撞灵敏度范围 0~5 个等级，设置为 0 时表示不开启碰撞检测。设定的值越大，碰撞灵敏度等级越高，机械臂碰撞检测后所需的力越小。
GPIO	通用型之输入输出。 对于输入，可以通过读取某个寄存器来确定引脚电位的高低； 对于输出，可以通过写入某个寄存器来让这个引脚输出高电位或者低电位；
安全边界	该模式被激活后，可以限制机械臂笛卡尔空间的边界范围，如果工具法兰中心（TCP 偏移点）超出设置的安全边界，机械臂将停止运动。
缩减模式	该模式被激活后，机械臂的笛卡尔运动的最大运动线速度、关节运动的最大关节速度和关节范围将受到限制。

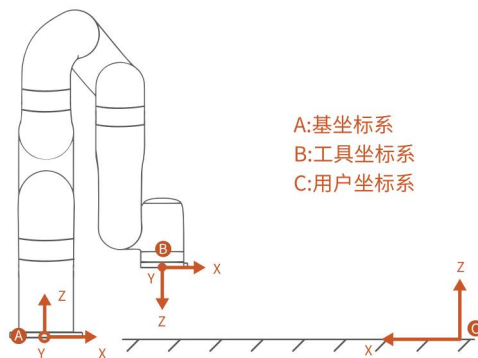


图 1

## 1.6. 其他开发者资源

最新的代码和文档说明请留意我们的 github:

ROS 库:

[https://github.com/xArm-Developer/xarm\\_ros](https://github.com/xArm-Developer/xarm_ros)

xArm-Python-SDK 库:

<https://github.com/xArm-Developer/xArm-Python-SDK>

xArm-CPLUS-SDK 库:

<https://github.com/xArm-Developer/xArm-CPLUS-SDK>

注: 对于以上三种开发者资源, 在 github 上我们有详细的安装步骤和使用说明, 请大家自行下载安装包进行进一步开发学习。

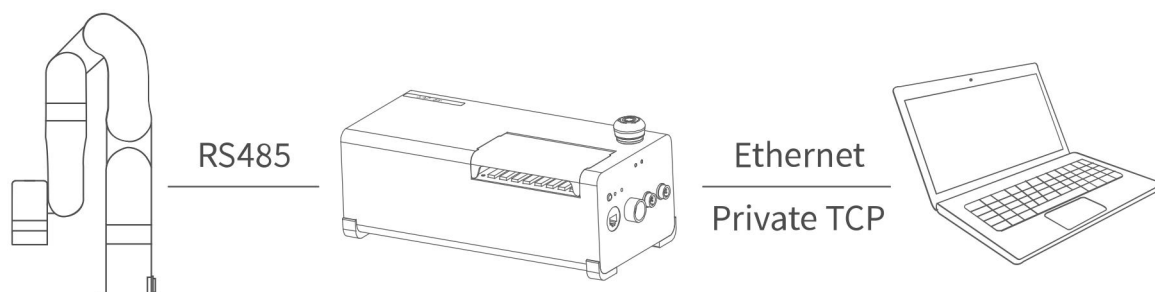
## 1.7. 更多信息

- 更多的产品介绍, xArm Studio 软件下载, 视频教程, 本手册的英文版本, 请留意我们的 UFACTORY 官网: <https://www.cn.ufactory.cc>
- 如需技术支持请邮件至: [support@ufactory.cc](mailto:support@ufactory.cc)
- 如需销售支持请邮件至: [sales@ufactory.cc](mailto:sales@ufactory.cc)



## 2. xArm 通信协议

### 2.1. 控制器通信协议



注：当前协议针对 xArm 进行了一些格式改动，在使用协议时请以本手册为主。

本章主要内容有两部分：（1）通过控制器的 Modbus TCP 协议来控制机械臂的运动。

（2）通过控制器的 Modbus TCP 协议来控制控制器和机械臂末端的 IO 设备。

### 协议格式

#### 2.1.1. 符号意义

下面对后面示例和表格中用到的一些符号进行阐述：

**【u8】**： —— 1 Byte，8 位无符号整型

**【u16】**： —— 2 Bytes，16 位无符号整型

**【fp32】**： —— 4 Bytes，浮点型

**【str】**： —— 字符串

**【系统重置】**：用户刚刚进行模式切换或者变更了某些设置（比如 TCP 偏移，灵敏度等等）后自动进入的状态，以上操作会使正在进行的运动终止并清空指令缓存，表现与 STOP 状态相同。

#### 2.1.2. Modbus-TCP 通信格式

**Modbus-TCP:**

Modbus 协议是一项应用层报文传输协议，包括 ASCII、RTU、TCP 三种报文类型。标准

Modbus 协议物理层接口有 RS232、RS422、RS485 和以太网接口，采用 master/slave 方式通信。

**Modbus TCP 通信过程：**

- (1) 建立 TCP 连接
- (2) 准备 modbus 报文
- (3) 使用 send 命令发送报文
- (4) 在同一连接下等待应答
- (5) 使用 recv 命令读取报文，完成一次数据交换
- (6) 通信任务结束时，关闭 TCP 连接

**参数：**

默认 TCP 端口：502

协议：0x00 0x02 控制(当前只有这一个)

**请求指令格式：**

格式	事务标识 (u16)	协议 (u16)	长度 (u16)	寄存器 (u8)	参数 (参照各指令说明)
长度	2 Bytes	2 Bytes	2 Bytes	1 Byte	n Bytes
示例 (使能伺服)	0x00 0x01	0x00 0x02	0x00 0x03	0x0B	0x08 0x01

**响应指令格式：**

格式	事务标识 (u16)	协议 (u16)	长度 (u16)	寄存器 (u8)	状态位 (u8)	回复参数 (参照各指令说明)
长度	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	n Bytes
示例 (使能伺服)	0x00 0x01	0x00 0x02	0x00 0x02	0x0B	0x00	none

**响应指令格式的状态位：**

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	1 存在错误 0 正常	1 存在警告 0 正常	1 不可执行运动 0 正常	保留	保留	保留	保留

注：事务标识：一般每次通信之后就要加 1 以区别不同的通信数据报文。

协议标识符：0x00 0x02 表示 ModbusTCP 协议。

长度：表示接下来的数据长度，单位为字节。

寄存器：设备地址。

**关于用户使用通信协议组织数据的大小端问题：**

Modbus-TCP 控制协议：

1. 报文头区的事务标识 (u16 )按大端序解析。
2. 报文头区的协议标识 (u16)按大端序解析。
3. 报文头区的长度 (u16) 按大端序解析。
4. 参数区 32 位数据类型 (fp32, int32) 按小端序解析。
5. 参数区涉及 GPIO 操作的 (u16 ) 按大端序解析。

自动上报数据解析：

1. 整型数据 (16/32 位 ) 按大端序解析。
2. 浮点型 (fp32) 数据按小端序解析。

示例：

假设变量 x 的类型为 int，位于地址 0x100 处，有一个十六进制的数 0x12345678（高位为 0x12, 低位为 0x78），地址范围为 0x100-0x103 的字节顺序依赖于机器的类型：

大端法：

	0x100	0x101	0x102	0x103	
...	0x12	0x34	0x56	0x78	...

小端法：

	0x100	0x101	0x102	0x103	
...	0x78	0x56	0x34	0x12	...

### 2.1.3. 寄存器（机械臂控制）

#### 2.1.3.1. 常用寄存器

下面对关节运动、轴角运动、设置参数、获取参数、特殊 IO 指令进行举例说明。

功能	<a href="#">关节运动</a>	<a href="#">设置 TCP 运动最大加速度</a>	<a href="#">获取笛卡尔位置</a>	<a href="#">轴角目标的直线运动</a>	<a href="#">控制器通用数字 IO 位置触发操作</a>
----	----------------------	--------------------------------	-------------------------	---------------------------	-----------------------------------

关节运动（P2P 运动）：

关节运动				
寄存器 23 (0x17)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29

	寄存器	1 Byte	u8	0x17
参数	关节 1 ( $J_1 = \pi/3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	关节 2 ( $J_2=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 3 ( $J_3=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 4 ( $J_4=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 5 ( $J_5=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 6 ( $J_6=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 7 ( $J_7=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8(速度= $20 * \pi / 180 \text{rad/s}$ )	4 Bytes	fp32	0xC2, 0xB8, 0xB2, 0x3E
	参数 9(加速度 $500 * \pi / 180 \text{rad/s}^2$ )	4 Bytes	fp32	0x58, 0xA0, 0x0B, 0x41
	参数 10(运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x17
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

设置 TCP 运动最大加速度:

<b>设置状态</b>				
<b>寄存器 32 (0x20)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x20
	参数 1 ( $\text{maxacc}=1000 \text{mm/s}^2$ )	4 Bytes	fp32	0x00, 0x00, 0x7A, 0x44
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x20
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

获取笛卡尔位置:

<b>获取笛卡尔位置</b>				
<b>寄存器 41 (0x29)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x29
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1A
	寄存器	1 Byte	u8	0x29
参数	状态	1 Byte	u8	0x00
	参数 1 (x=207mm)	4 Bytes	fp32	0x00, 0x00, 0x4F, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=112mm)	4 Bytes	fp32	0x00, 0x00, 0xE0, 0x42
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

**轴角姿态目标的直线运动:**

<b>轴角姿态目标的直线运动</b>				
<b>寄存器 92 (0x5C)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度 (参数长度+1)	2 Bytes	u16	0x00, 0x27
	寄存器	1 Byte	u8	0x5C
参数	参数 1 (X=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 2 (Y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (Z=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 (Rx=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (Ry=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (Rz=2 $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0xC9, 0x40
	参数 7 (速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 8 (加速度=2000mm/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 9 (运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 10 (运动坐标系=基坐标系)	1 Byte	u8	0x00
	参数 11 (给定位姿为相对位姿)	1 Byte	u8	0x01
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04

	寄存器	1 Byte	u8	0x5C
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

### 控制器通用数字 IO 位置触发操作:

控制器通用数字 IO 位置触发操作				
寄存器 145 (0x91)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x13
	寄存器	1 Byte	u8	0x91
参数	参数 1 (iomum=0)	1 Byte	u8	0x00
	参数 2 (on-off: 打开 (1))	1 Byte	u8	0x01
	参数 3 (x=300)	4 Bytes	fp32	0x00, 0x00, 0x96, 0x43
	参数 4 (y=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (z=300)	4 Bytes	fp32	0x00, 0x00, 0x96, 0x43
	参数 6 (容错半径 (tol_r)=3)	4 Bytes	fp32	0x00, 0x00, 0x40, 0x40
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x91
参数	状态	1 Byte	u8	0x00

### 2.1.3.2. 寄存器列表

#### 0~10: 公共端口段

获取版本信息 (0x01)

获取 SN 信息 (0x02)

重新加载摩擦力参数 (0x04)

读取当前关节扭矩或电机电流上报设置 (0x05)

获取目标关节相对 TCP 的旋转半径 (0x06)

读取其他开关量的设置 (0x07)

远程关闭操作系统 (0x0A)

#### 11~20: 系统状态

使能/关闭伺服系统（系统重置）（0x0B）

设置运动状态（0x0C）

获取运动状态（0x0D）

获取指令缓冲区中的指令数量（0x0E）

获取控制器错误和警告代码（0x0F）

清除控制器错误（系统重置）（0x10）

清除控制器警告（0x11）

单独设置抱闸开关（系统重置）（0x12）

设置系统运动模式（系统重置）（0x13）

### **20~30: 基础运动**

笛卡尔直线运动（0x15）

带圆弧交融的直线运动（0x16）

P2P 关节运动（0x17）

关节交融运动（0x18）

回零点运动（0x19）

停顿指令/指令延时（0x1A）

圆弧运动（0x1B）

工具坐标系直线运动（0x1C）

Servoj 运动（0x1D）

伺服笛卡尔运动(servo\_cartesian)（0x1E）

### **31~40: 系统参数设置**

设置笛卡尔空间平移的加加速度（0x1F）

设置笛卡尔空间平移最大加速度（0x20）

设置关节空间加加速度（0x21）

设置关节空间最大加速度（0x22）

设置机械臂末端工具的偏移（系统重置）（0x23）

设置机械臂末端负载（0x24）

设置碰撞检测灵敏度参数（系统重置）（0x25）

设置拖动示教的灵敏度（系统重置）（0x26）

删除当前系统配置参数（0x27）

保存当前配置参数（0x28）

#### **41~50：获取运动信息**

获取控制器当前笛卡尔位置（0x29）

获取控制器当前关节位置（0x2A）

获取逆运动学的解（0x2B）

获取正运动学的解（0x2C）

查询关节空间的限位（0x2D）

设置缩减模式最大线速度（0x2F）

设置缩减模式最大关节角速度（0x30）

读取当前缩减模式生效状态（0x31）

设置缩减模式生效状态（0x32）

#### **51~100：其他机械臂功能**

设置重力方向（0x33）

设置安全边界范围（0x34）

获取当前缩减模式的全部设置（0x35）

获取伺服当前关节力矩（0x37）

设置缩减模式关节运动范围（0x3A）

安全边界启动开关（0x3B）

设置碰撞回弹生效状态（0x3C）

设置轨迹录制（0x3D）

保存轨迹（0x3E）

加载轨迹（0x3F）

播放轨迹（0x40）

获取轨迹状态（0x41）

设置允许使用近似解回避某些奇异点附近的超速问题（0x42）

获取 DH 参数（0x43）



设置 DH 参数 (0x44)

获取当前执行的指令 (0x45)

设置关节力矩 (估算) 或电机电流上报 (0x46)

设置用户坐标系与基坐标系偏移量 (0x49)

计算两个给定点的位姿偏移量 (0x4C)

设置机械臂 (及可设置的末端工具) 的自碰撞检测功能 (0x4D)

设置自碰撞检测时加入的末端工具几何模型 (0x4E)

设置是否开启虚拟机械臂模式 (0x4F)

笛卡尔运动速度连续的全局设置 (0x50)

关节速度控制 (0x51)

末端笛卡尔速度控制 (0x52)

相对运动控制 (0x53)

获取轴角姿态表示的位姿 (0x5B)

轴角姿态目标的直线运动 (0x5C)

伺服笛卡尔运动 (轴角) (0x5D)

### 101~115: 伺服模组

获取当前机械臂伺服状态 (0x6A)

### 0~10 号 公共端口段

获取版本信息				
寄存器: 1 (0x01)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x01
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x01
参数	状态	1 Byte	u8	0x00

获取 SN 信息				
寄存器：2(0x02)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x02
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x02
参数	状态	1 Byte	u8	0x00
	参数（字符串） 手臂 SN 控制器 SN	n Byte	n * u8	XI120010191B03AC1300032100 00

重新加载摩擦力参数				
寄存器：4(0x04)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x04
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x04
参数	状态	1 Byte	u8	0x00

读取当前关节扭矩或电机电流上报设置				
寄存器：5(0x05)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x05
响应指令格式				

Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x05
参数	状态	1 Byte	u8	0x00
	参数 1 (上报关节力矩估计值) 0: 上报关节力矩估计值 1: 上报电机电流实际值	1 Byte	u8	0x00

<b>获取目标关节相对 TCP 的旋转半径</b>				
<b>寄存器: 6(0x06)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x06
	参数 1 (目标关节: 6)	1 Byte	u8	0x06
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x06
	寄存器	1 Byte	u8	0x06
参数	状态	1 Byte	u8	0x00
	参数 1 (旋转半径: mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

<b>读取其他开关量的设置</b>				
<b>寄存器: 7(0x07)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02

	长度	2 Bytes	u16	0x00,0x01
	寄存器	1 Byte	u8	0x07
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x06
	寄存器	1 Byte	u8	0x07
参数	状态	1 Byte	u8	0x00
	参数 1 (按位标识的开关量集合) bit 0: 允许使用近似解规划的开关 bit 1-31: 预留	4 Bytes	u32	0x00, 0x00, 0x00, 0x01

<b>远程关闭操作系统</b>				
<b>寄存器 10 (0x0A)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0A
参数	参数 1 (操作: 1: 远程关闭操作系统 2: 重启系统)	1 Byte	u8	0x01
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x0A

参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2	u16	0x00, 0x01

## 11~20 系统状态

使能/关闭伺服(系统重置)				
注：以上操作会使正在进行的运动终止并清空指令缓存，表现与 STOP 状态相同。				
寄存器：11 (0x0B)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x0B
参数	参数 1 (选择所有关节) 关节号 (选择所有关节) 1-7: 选择某一个关节 8: 选择所有关节	1 Byte	u8	0x08
	参数 2 (使能机械臂) 是否使能 1: 为使能 (0x01) 0: 为关闭电机 (0x00)	1 Byte	u8	0x01
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0B
参数	状态	1 Byte	u8	0x10

设置运动状态				
寄存器：12 (0x0C)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0C

参数	参数 1: (进入运动状态) 运动状态: 0: 进入运动状态 3: 暂停当前运动 4: 停止当前所有运动	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0C
参数	状态	1 Byte	u8	0x00

<b>获取运动状态</b>				
<b>寄存器: 13 (0x0D)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x0D
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x0D
参数	状态	1 Byte	u8	0x00
	参数 1 (运动中)  运动状态: 1: 运动中 2: 休眠中 3: 暂停中 4: 停止中 5: 系统重置: 用户刚刚进行模式切换或者变更了某些设置(比如 TCP 偏移, 灵敏度等等)后自动进入的状态, 以上操作会使正在进行的运动终止并清空指令缓存, 表现与 STOP 状态相同。	1 Byte	u8	0x01

获取指令缓冲区中的指令数量				
寄存器：14 (0x0E)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x0E
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x0E
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

获取控制器错误和警告代码				
寄存器：15 (0x0F)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x0F
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x0F
参数	状态	1 Byte	u8	0x00
	参数 1 (错误代码：无错误)	1 Byte	u8	0x00
	参数 2 (警告代码：无警告)	1 Byte	u8	0x00

清除控制器错误 (系统重置)				
注：以上操作会使正在进行的运动终止并清空指令缓存，表现与 STOP 状态相同。				
寄存器：16 (0x10)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x10

响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x10
参数	状态	1 Byte	u8	0x10

清除控制器警告				
寄存器: 17 (0x11)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x11
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x11
参数	状态	1 Byte	u8	0x00

单独设置抱闸开关（系统重置）				
注：以上操作会使正在进行的运动终止并清空指令缓存，表现与 STOP 状态相同。				
寄存器: 18 (0x12)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x12
参数	参数 1（选择所有关节） 选择设置的电机关节号： 1~6：单独选择一个电机关节 8：选择所有关节	1 Byte	u8	0x08
	参数 2（使能抱闸） 操作： 1：使能抱闸 0：释放抱闸	1 Byte	u8	0x01
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02



	寄存器	1 Byte	u8	0x12
参数	状态	1 Byte	u8	0x10

设置系统运动模式(系统重置)				
注：以上操作会使正在进行的运动终止并清空指令缓存，表现与 STOP 状态相同。				
寄存器：19 (0x13)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x03
	寄存器	1 Byte	u8	0x13
参数	参数 1 (位置控制模式) 运动模式： 0：位置控制模式 1：servo 运动模式 2：关节示教模式 3：笛卡尔示教模式（暂无） 4：关节速度模式 5：笛卡尔速度模式 6：关节在线规划模式 7：笛卡尔在线规划模式	1 Byte	u8	0x00
	参数 2 (示教模式负载检测) 0：开启 1：关闭	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01

	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x13
参数	状态	1 Byte	u8	0x00

## 21~30 基础运动

笛卡尔直线运动				
寄存器 21 (0x15)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x25
	寄存器	1 Byte	u8	0x15
参数	参数 1 (x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8 (速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 9 (加速度 $2000\text{mm/s}^2$ ) = $500 * \pi / 180\text{rad/s}^2$ )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 10 (运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x15
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

带圆弧交融的直线运动				
寄存器: 22 (0x16)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29
	寄存器	1 Byte	u8	0x16
参数	参数 1 (x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (运动速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 8 (加速度=2000mm/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 9 (运动时间 (0))	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 10 (圆弧交融半径=50mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x42
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x16
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

P2P 关节运动				
寄存器: 23 (0x17)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29
	寄存器	1 Byte	u8	0x17
参数	参数 1 ( $J_1 = \pi/3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	参数 2 ( $J_2 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $J_3 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 ( $J_4 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $J_6 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

	参数 7 ( $J_7=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8(速度= $20*\pi/180\text{rad/s}$ )	4 Bytes	fp32	0xC2, 0xB8, 0xB2, 0x3E
	参数 9(加速度 $500*\pi/180\text{rad/s}^2$ )	4 Bytes	fp32	0x58, 0xA0, 0x0B, 0x41
	参数 10(运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x17
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

<b>关节交融运动 (指定交融半径)</b>				
<b>寄存器: 24 (0x18)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29
	寄存器	1 Byte	u8	0x18
参数	参数 1 ( $J_1= \pi/3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	参数 2 ( $J_2=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $J_3=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 ( $J_4=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $J_6=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 ( $J_7=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8(速度= $20*\pi/180\text{rad/s}$ )	4 Bytes	fp32	0xC2, 0xB8, 0xB2, 0x3E
	参数 9(加速度 $500*\pi/180\text{rad/s}^2$ )	4 Bytes	fp32	0x58, 0xA0, 0x0B, 0x41
	参数 10(交融半径=10mm)	4 Bytes	fp32	0x00, 0x00, 0x20, 0x41
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x18
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

回零点运动				
寄存器: 25 (0x19)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x0D
	寄存器	1 Byte	u8	0x19
参数	参数 1 (运动速度=50rad/s)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 2 (加速度=600rad/s <sup>2</sup> )	4 Bytes	fp32	0xF3, 0x66, 0xDF, 0x40
	参数 3 (运动时间为 0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x19
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

停顿指令, 指令延时				
寄存器: 26 (0x1A)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x1A
参数	参数 1 (停顿时间=3s)	4 Bytes	fp32	0x00, 0x00, 0x40, 0x40
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x1A
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

圆弧运动: 运动根据三点坐标计算出空间圆的轨迹, 三点坐标分别为 (当前起点、位置 1、位置 2)				
寄存器: 27 (0x1B)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x41
	寄存器	1 Byte	u8	0x1B
参数	参数 1(x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2(y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3(z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4(roll=π)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5(pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6(yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7(x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 8(y=100mm)	4 Bytes	fp32	0x00, 0x00, 0xC8,
	参数 9(z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 10(roll=π)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 11(pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 12(yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 13(运动速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 14 (加速度=2000mm/s <sup>2</sup> ) =500*π/180rad/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 15(运动时间为0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 16 (运动的弧长与圆周长的百分比=50%)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x42
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x1B
参数	状态	1 Byte	u8	0x00
	参数 1(指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

<b>工具坐标系直线运动：基于当前的工具坐标系，做笛卡尔直线相对运动。</b>				
<b>寄存器：28 (0x1C)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x25

	寄存器	1 Byte	u8	0x1C
参数	参数 1 (x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (运动速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 8 (加速度=2000mm/s <sup>2</sup> ) =500* $\pi$	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 9 (运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x1C
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

<b>Servoj 运动</b>				
<b>寄存器: 29 (0x1D)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29
	寄存器	1 Byte	u8	0x1D
参数	参数 1 ( $J_1 = \pi/3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	参数 2 ( $J_2 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $J_3 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 ( $J_4 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $J_6 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 ( $J_7 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8 (速度, 无意义, 为 0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 9 (加速度, 无意义, 为 0) =500* $\pi/180\text{rad/s}^2$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 10 (运动时间, 无意义, 为 0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02

	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x1D
参数	状态	1 Byte	u8	0x00

伺服笛卡尔运动(servo_cartesian)				
寄存器: 30 (0x1E)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x25
	寄存器	1 Byte	u8	0x1E
参数	参数 1 (x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (运动速度无意义)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8 (加速度无意义) = $500 * \pi / 180 \text{rad/s}^2$	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 9 (基坐标系) 运动坐标系: 0 : 基坐标系 1 : 工具坐标系	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x1E
参数	状态	1 Byte	u8	0x00

### 31~40 运动参数设置

设置笛卡尔空间平移的加加速度				
寄存器: 31 (0x1F)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x1F
参数	参数 1 (加加速度 = $2000 \text{mm/s}^3$ )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44



响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x1F
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

设置笛卡尔空间平移最大加速度				
寄存器: 32 (0x20)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x20
参数	参数 1 (最大加速度= 6000mm/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x80, 0xBB, 0x45
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x20
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

设置关节空间加加速度				
寄存器: 33 (0x21)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x21
参数	参数 1 (加加速度= 10000rad/s <sup>3</sup> )	4 Bytes	fp32	0x00, 0x40, 0x1C, 0x46
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x21
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

设置关节空间最大加速度				
寄存器: 34 (0x22)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x22
参数	参数 1 (最大加速度=400 rad/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x22
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

设置机械臂末端工具的偏移(系统重置)				
注: 以上操作会使正在进行的运动终止并清空指令缓存, 表现与 STOP 状态相同。				
寄存器: 35 (0x23)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x19
	寄存器	1 Byte	u8	0x23
参数	参数 1 (x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (roll=π)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x23
参数	状态	1 Byte	u8	0x10

设置机械臂末端负载				
寄存器: 36 (0x24)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x11
	寄存器	1 Byte	u8	0x24
参数	参数 1 (负载质量 =1kg)	4 Bytes	fp32	0x00, 0x00, 0x80, 0x3F
	参数 2 (负载重心 X=400 mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 3 (负载重心 Y= 0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 (负载重心 Z=200 mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x24
参数	状态	1 Byte	u8	0x00

设置碰撞检测灵敏度参数(系统重置)				
注: 以上操作会使正在进行的运动终止并清空指令缓存, 表现与 STOP 状态相同。				
寄存器: 37 (0x25)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x25
参数	参数 1 (检测灵敏度=4)	1 Byte	u8	0x04
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x25
参数	状态	1 Byte	u8	0x10

设置拖动示教的灵敏度(系统重置)				
注: 以上操作会使正在进行的运动终止并清空指令缓存, 表现与 STOP 状态相同。				
寄存器: 38 (0x26)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x26
参数	参数 1 (示教的灵敏度=4)	1 Byte	u8	0x04
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x26
参数	状态	1 Byte	u8	0x10

<b>删除当前系统配置参数</b>				
<b>寄存器: 39 (0x27)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x27
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x27
参数	状态	1 Byte	u8	0x00

<b>保存当前配置参数</b>				
<b>寄存器: 40 (0x28)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x28
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x28
参数	状态	1 Byte	u8	0x00

### 41~50 获取运动信息

获取机械臂当前笛卡尔位置				
寄存器 41 (0x29)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x29
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x0, 0x1A
	寄存器	1 Byte	u8	0x29
参数	状态	1 Byte	u8	0x00
	参数 1 (x=207mm)	4 Bytes	fp32	0x00, 0x00, 0x4F, 0x43
	参数 2 (y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (z=112mm)	4 Bytes	fp32	0x00, 0x00, 0xE0, 0x42
	参数 4 (roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

获取机械臂当前关节位置				
寄存器: 42 (0x2A)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x2A
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1E
	寄存器	1 Byte	u8	0x2A
参数	状态	1 Byte	u8	0x00
	关节 1 ( $J_1 = \pi/3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	关节 2 ( $J_2 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

	关节 3 ( $J_3=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 4 ( $J_4=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 5 ( $J_5=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 6 ( $J_6=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	关节 7 ( $J_7=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

获取逆运动学的解				
寄存器: 43 (0x2B)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x19
	寄存器	1 Byte	u8	0x2B
参数	参数 1 ( $x=400\text{mm}$ )	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 ( $y=0\text{mm}$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $z=200\text{mm}$ )	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 ( $\text{roll}=\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 ( $\text{pitch}=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $\text{yaw}=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1E
	寄存器	1 Byte	u8	0x2B
参数	状态	1 Byte	u8	0x00
	参数 1 ( $J_1=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 2 ( $J_2=0.081803$ )	4 Bytes	fp32	0x38, 0x88, 0xA7, 0x3D
	参数 3 ( $J_3=-0.641152$ )	4 Bytes	fp32	0x88, 0x22, 0x24, 0xBF
	参数 4 ( $J_4=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5=0.559349$ )	4 Bytes	fp32	0x81, 0x31, 0x0F, 0x3F
	参数 6 ( $J_6=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 ( $J_7=0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

获取正运动学的解				
寄存器: 44 (0x2C)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1D

	寄存器	1 Byte	u8	0x2C
参数	参数 1 ( $J_1 = \pi / 3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	参数 2 ( $J_2 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $J_3 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 ( $J_4 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $J_6 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 ( $J_7 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1A
	寄存器	1 Byte	u8	0x2C
参数	状态	1 Byte	u8	0x00
	参数 1 ( $x = 103.5\text{mm}$ )	4 Bytes	fp32	0x18, 0x00, 0xCF, 0x42
	参数 2 ( $y = 179.27\text{mm}$ )	4 Bytes	fp32	0x80, 0x44, 0x33, 0x43
	参数 3 ( $z = 112\text{mm}$ )	4 Bytes	fp32	0x08, 0x01, 0xA0, 0x42
	参数 4 ( $\text{roll} = -\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0xC0
	参数 5 ( $\text{pitch} = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x80
	参数 6 ( $\text{yaw} = -\pi / 3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F

<b>查询关节空间的限位</b>				
<b>寄存器: 45 (0x2D)</b>				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1D
	寄存器	1 Byte	u8	0x2D
参数	参数 1 ( $J_1 = \pi / 3$ )	4 Bytes	fp32	0x92, 0x0A, 0x86, 0x3F
	参数 2 ( $J_2 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 ( $J_3 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 ( $J_4 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 ( $J_5 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 ( $J_6 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 ( $J_7 = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03

	寄存器	1 Byte	u8	0x2D
参数	状态	1 Byte	u8	0x00
	参数 1(不发生碰撞) 1 : 发生碰撞 0 : 不发生碰撞	1 Byte	u8	0x00

设置缩减模式最大线速度				
寄存器: 47 (0x2F)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x2F
参数	参数 1 (最大线速度=400mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x2F
参数	状态	1 Byte	u8	0x00

设置缩减模式最大关节角速度				
寄存器: 48 (0x30)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x05
	寄存器	1 Byte	u8	0x30
参数	参数 1 (最大关节速度=1.0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x80, 0x3F
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x30
参数	状态	1 Byte	u8	0x00



读取当前缩减模式生效状态				
寄存器: 49 (0x31)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x31
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x31
参数	状态	1 Byte	u8	0x00
	参数 1 (缩减模式生效状态) 0: 未生效 1: 生效中	1 Byte	u8	0x00

设置缩减模式生效状态				
寄存器: 50 (0x32)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x32
参数	参数 1 (缩减模式生效状态) 0-不生效; 1-设置生效	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x32
参数	状态	1 Byte	u8	0x00

51~100 其他机械臂功能

设置重力方向				
设置重力方向，用于正确的力矩补偿和碰撞检测，改动后需要调用 <code>save_conf()</code> 函数或参考 操作保存设置，否则下次重启会失效。				
寄存器：51 (0x33)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x0D
	寄存器	1 Byte	u8	0x33
参数	参数 1 (机械臂基坐标系下的重力方向向量 X=0) 例：水平安装 (默认) : [0, 0, -1]	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 2 (机械臂基坐标系下的重力方向向量 Y=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (机械臂基坐标系下的重力方向向量 Z=-1)	4 Bytes	fp32	0x00, 0x00, 0x80, 0xBF
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x33
参数	状态	1 Byte	u8	0x00

设置安全边界范围				
设置三维空间的安全围栏边界范围，如果机械臂 TCP 超出此边界会触发控制器错误 C35				
寄存器：52 (0x34)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x19
	寄存器	1 Byte	u8	0x34
参数	参数 1 (笛卡尔边界值 x+ =600mm)	4 Bytes	int32	0x58, 0x02, 0x00, 0x00
	参数 2 (笛卡尔边界值 x- =200mm)	4 Bytes	int32	0xC8, 0x00, 0x00, 0x00

	参数 3 (笛卡尔边界值 $y^+ = 500\text{mm}$ )	4 Bytes	int32	0xF4, 0x01, 0x00, 0x00
	参数 4 (笛卡尔边界值 $y^- = 100\text{mm}$ )	4 Bytes	int32	0x64, 0x00, 0x00, 0x00
	参数 5 (笛卡尔边界值 $z^+ = 600\text{mm}$ )	4 Bytes	int32	0x58, 0x02, 0x00, 0x00
	参数 6 (笛卡尔边界值 $z^- = 200\text{mm}$ )	4 Bytes	int32	0xC8, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x34
参数	状态	1 Byte	u8	0x00

<b>获取当前缩减模式的全部设置</b>				
<b>寄存器: 53 (0x35)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x35
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x51
	寄存器	1 Byte	u8	0x35
参数	状态	1 Byte	u8	0x00
	参数 1 (缩减模式是否生效: 0 未生效)	1 Byte	u8	0x00
	参数 2~7 (安全边界: $[x_{\max}, x_{\min}, y_{\max}, y_{\min}, z_{\max}, z_{\min}]$ , 单位 mm)	2 Bytes*6	int 16	

参数 8 (最大线速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
参数 9 (最大关节速度=3.14 rad/s)	4 Bytes	fp32	0xC2, 0xF5, 0x48, 0x40
参数 10~23 (关节范围: [J1_min, J2_max, ..., J7_min, J7_max])	4 Bytes*14	fp32	
参数 24 (安全边界是否生效: 0-未生效)	1 Byte	u8	0x00
参数 25 (碰撞回弹是否生效: 0-未生效)	1 Byte	u8	0x00

获取伺服当前关节力矩 基于电流和理论模型估算的关节力矩, 仅供参考				
寄存器: 55 (0x37)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x37
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1E
	寄存器	1 Byte	u8	0x37
参数	状态	1 Byte	u8	0x00
	参数 1 (Joint1 理论力矩位=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 2 (Joint2 理论力矩位=-13.7N.m)	4 Bytes	fp32	0x2A, 0xC5, 0x5B, 0xC1
	参数 3 (Joint3 理论力矩位=-6.17N.m)	4 Bytes	fp32	0x79, 0xA4, 0xC5, 0xC0

	参数 4 (Joint4 理论力矩位=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (Joint5 理论力矩位=-1.83N.m)	4 Bytes	fp32	0x87, 0xA3, 0xE9, 0xBF
	参数 6 (Joint6 理论力矩位=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (Joint7 理论力矩位=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

设置缩减模式关节运动范围				
寄存器: 58 (0x3A)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x39
	寄存器	1 Byte	u8	0x3A
参数	参数 1, 2 (J1 最小值=-3.14rad; J1 最大值=3.14rad)	4 Bytes *2	fp32	0xC2, 0xF5, 0x48, 0xC0; 0xC2, 0xF5, 0x48, 0x40
	参数 3, 4 (J2 最小值=-2.05 rad; J2 最大值=2.09rad)	4 Bytes *2	fp32	0x33, 0x33, 0x03, 0xC0; 0x8F, 0xC2, 0x05, 0x40
	参数 5, 6 (J3 最小值=-3.14rad; J3 最大值=3.14rad)	4 Bytes *2	fp32	0xC2, 0xF5, 0x48, 0xC0; 0xC2, 0xF5, 0x48, 0x40
	参数 7, 8 (J4 最小值=-0.19rad; J4 最大值=3.92rad)	4 Bytes *2	fp32	0x5C, 0x8F, 0x42, 0xBE; 0x47, 0xE1, 0x7A, 0x40
	参数 9, 10 (J5 最小值=-3.14rad; J5 最大值=3.14rad)	4 Bytes *2	fp32	0xC2, 0xF5, 0x48, 0xC0; 0xC2, 0xF5, 0x48, 0x40

	参数 11, 12 (J6 最小值=-6.19rad; J6 最大值=3.14rad)	4 Bytes *2	fp32	0xEB, 0x51, 0xD8, 0xBF; 0xC8, 0x00, 0x00, 0x00
	参数 13, 14 (J7 最小值=-3.14rad; J7 最大值=3.14rad)	4 Bytes *2	fp32	0xC2, 0xF5, 0x48, 0xC0; 0xC2, 0xF5, 0x48, 0x40
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3A
参数	状态	1 Byte	u8	0x00

<b>安全边界启动开关</b> 设置三维空间的安全围栏边界生效开关，生效后如果机械臂 TCP 超出此边界会触发控制器错误 C35				
<b>寄存器：59 (0x3B)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3B
参数	参数 1 (关闭安全边界检测) 生效开关： 0：关闭安全边界检测 1：启动安全边界检测	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3B
参数	状态	1 Byte	u8	0x00

设置碰撞回弹生效状态				
寄存器: 60 (0x3C)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3C
参数	参数 1 (碰撞回弹生效状态) 0-不生效; 1-设置生效	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3C
参数	状态	1 Byte	u8	0x00

设置轨迹录制				
寄存器: 61 (0x3D)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3D
参数	操作: 0-停止录制; 1-开始录 制	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3D
参数	状态	1 Byte	u8	0x00

保存轨迹				
寄存器：62 (0x3E)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x0A
	寄存器	1 Byte	u8	0x3E
参数	轨迹名称（最长 80 个字节） 例：test. traj	n Byte	u8	0x74, 0x65, 0x73, 0x74, 0x2E, 0x74, 0x72, 0x61, 0x6A
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3E
参数	状态	1 Byte	u8	0x00

加载轨迹				
寄存器：63 (0x3F)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x0A
	寄存器	1 Byte	u8	0x3F
参数	轨迹名称（最长 80 个字节） 例：test. traj	n Byte	u8	0x74, 0x65, 0x73, 0x74, 0x2E, 0x74, 0x72, 0x61, 0x6A
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x3F
参数	状态	1 Byte	u8	0x00



播放轨迹				
寄存器: 64 (0x40)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x09
	寄存器	1 Byte	u8	0x40
参数	参数 1: 循环次数	4 Bytes	u32	0x00, 0x00, 0x00, 0x01
	参数 2: 倍数 1-1 倍数; 2-2 倍数; 4-4 倍数	4 Bytes	u32	0x00, 0x00, 0x00, 0x01
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x40
参数	状态	1 Byte	u8	0x00

获取轨迹状态				
寄存器: 65 (0x41)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x41
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x41
参数	状态	1 Byte	u8	0x00
	参数 1: 0-空闲; 1-正在加载; 2-加载成功; 3-加载失败; 4-正在保存; 5-保存成功; 6-保存失败	1 Byte	u8	0x00

设置允许使用近似解回避某些奇异点附近的超速问题				
寄存器: 66 (0x42)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x42
参数	参数 1 (是否允许) 0 为不允许 1 为允许	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x42
参数	状态	1 Byte	u8	0x00

获取 DH 参数				
寄存器: 67 (0x43)				
请求指令格式				
Modbus TCP 包	事务标识	2 Bytes	u16	0x00,0x01

头	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x01
	寄存器	1 Byte	u8	0x43
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x43
参数	状态	1 Byte	u8	0x00
	DH 参数 (每个关节 4 个浮点参数) (共 112 字节, 28 个浮点数据, 5 轴只有前 20 个浮点数据为有效数据)	4 Bytes * 28	fp32	

<b>设置 DH 参数</b>				
<b>寄存器: 68 (0x44)</b>				
<b>请求指令格式</b>				
Modbus TCP 包	事务标识	2 Bytes	u16	0x00,0x01

头	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x01
	寄存器	1 Byte	u8	0x44
	要设置的 DH 参数 (每个关节 4 个浮点参数)  (共 112 字节, 28 个浮点数据, 5 轴只有前 20 个浮点数据为有效数据, 后面 8 个浮点数据用 0 代替)	4 Bytes * 28	fp32	
设置标志  0:使用设置的 DH 参数, 但不保存到配置文件  1: 使用设置的 DH 参数, 并保存到配置文件  2: 使用设置的 DH 参数, 并删除配置文件  3: 使用默认的 DH 参数, 但不删除配置文件  4: 使用默认的 DH 参数, 并删除配置文件	1 Byte	u8	0x00	
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02

	寄存器	1 Byte	u8	0x44
参数	状态	1 Byte	u8	0x00

<b>获取当前执行的指令</b>				
<b>寄存器: 69 (0x45)</b>				
<b>请求指令格式</b>				
Modbus TCP 包 头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x01
	寄存器	1 Byte	u8	0x45
<b>响应指令格式</b>				
Modbus TCP 包 头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x45
参数	状态	1 Byte	u8	0x00
	当前执行运动指令寄存器:	1 Byte	u8	0x15

	<p>0x15: 笛卡尔直线运动</p> <p>0x16: 带圆弧交融的直线运动</p> <p>0x17: P2P 关节运动</p> <p>0x18: 关节交融运动</p> <p>0x19: 回零点运动</p> <p>0x1B: 圆弧运动</p> <p>0x1C: 工具坐标系直线运动</p> <p>0x5C: 轴角姿态目标的直线运动</p>			
	<p>当前执行运动指令内容</p> <p>(根据运动指令寄存器进行解析)</p>	n Byte	u8	

设置关节力矩（估算）或电机电流上报 (对应各上报端口的 60~87Bytes 内容)				
寄存器: 70 (0x46)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x46
参数	<p>参数 1 (上报关节力矩估计值)</p> <p>0 为上报关节力矩估计值, 单位: Nm</p> <p>1 为上报电机读取的电流值, 单位: A</p>	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x46
参数	状态	1 Byte	u8	0x00

设置用户坐标系与基坐标系偏移量 设置用户坐标系与基坐标系偏移量，具体指机械臂基坐标系在用户定义坐标系下描述的偏移量					
寄存器：73 (0x49)					
请求指令格式					
Modbus TCP 包头	事务标识		2 Bytes	u16	0x00, 0x01
	协议		2 Bytes	u16	0x00, 0x02
	长度		2 Bytes	u16	0x00, 0x19
	寄存器		1 Byte	u8	0x49
参数	参数 1 (笛卡尔偏移量 X=400 mm)		4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2 (笛卡尔偏移量 Y=0 mm)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (笛卡尔偏移量 Z=200 mm)		4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4 (笛卡尔偏移量 Roll= $\pi$ rad)		4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (笛卡尔偏移量 Pitch=0 rad)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (笛卡尔偏移量 Yaw=0 rad)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
响应指令格式					
Modbus TCP 包头	事务标识		2 Bytes	u16	0x00, 0x01
	协议		2 Bytes	u16	0x00, 0x02
	长度		2 Bytes	u16	0x00, 0x02
	寄存器		1 Byte	u8	0x49
参数	状态		1 Byte	u8	0x00

计算两个给定点的位姿偏移量 给定机械臂的两个坐标点，可以计算得出他们之间的偏移量坐标					
寄存器：76 (0x4C)					
请求指令格式					
Modbus TCP 包头	事务标识		2 Bytes	u16	0x00, 0x01
	协议		2 Bytes	u16	0x00, 0x02
	长度		2 Bytes	u16	0x00, 0x33
	寄存器		1 Byte	u8	0x4C
参数	Point1	参数 1 (X=400mm)	4 Bytes*6	fp32*6	0x00, 0x00, 0xC8, 0x43
		参数 2 (Y=0)			0x00, 0x00, 0x00, 0x00
		参数 3 (Z=200mm)			0x00, 0x00, 0x48, 0x43

		参数 4 (Roll= $\pi$ rad)			0xDB, 0x0F, 0x49, 0x40	
		参数 5 (Pitch=0)			0x00, 0x00, 0x00, 0x00	
		参数 6 (Yaw=0)			0x00, 0x00, 0x00, 0x00	
	Point2	4 Bytes*6	fp32*6	参数 7 (X=400mm)	0x00, 0x00, 0xC8, 0x43	
				参数 8 (Y=0)	0x00, 0x00, 0x00, 0x00	
				参数 9 (Z=100mm)	0x00, 0x00, 0xC8, 0x42	
				参数 10 (Roll= $\pi$ rad)	0xDB, 0x0F, 0x49, 0x40	
				参数 11 (Pitch=0)	0x00, 0x00, 0x00, 0x00	
				参数 12 (Yaw=0)	0x00, 0x00, 0x00, 0x00	
	参数 13 ( ) 输入姿态的表示法: 0 : RPY (Roll, Pitch, Yaw) 1 : 轴角 (Rx, Ry, Rz)		1 Byte	u8	0x00	
	参数 14 ( ) 输出姿态的表示法: 0 : RPY (Roll, Pitch, Yaw) 1 : 轴角 (Rx, Ry, Rz)		1 Byte	u8	0x00	
	<b>响应指令格式</b>					
	Modbus TCP 包头	事务标识		2 Bytes	u16	0x00, 0x01
协议		2 Bytes	u16	0x00, 0x02		
长度		2 Bytes	u16	0x00, 0x1A		
寄存器		1 Byte	u8	0x4C		
参数	状态		1 Byte	u8	0x00	
	参数 1 (偏移量笛卡尔坐标 X=0)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00	
	参数 1 (偏移量笛卡尔坐标 Y=0)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00	
	参数 1 (偏移量笛卡尔坐标 Z=-100)		4 Bytes	fp32	0x00, 0x00, 0xC8, 0xC2	
	参数 1 (偏移量笛卡尔坐标 Roll=-0)		4 Bytes	fp32	0x00, 0x00, 0x80, 0x99	
	参数 1 (偏移量笛卡尔坐标 Pitch=-0)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x80	
	参数 1 (偏移量笛卡尔坐标 Yaw=0)		4 Bytes	fp32	0x00, 0x00, 0x00, 0x00	

设置机械臂（及可设置的末端工具）的自碰撞检测功能					
寄存器：77 (0x4D)					
请求指令格式					
Modbus TCP 包头	事务标识		2 Bytes	u16	0x00, 0x01
	协议		2 Bytes	u16	0x00, 0x02



	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x4D
参数	参数 1 (开启自碰撞检测) 0 为关闭自碰撞检测 1 为开启自碰撞检测	1 Byte	u8	0x01
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x4D
参数	状态	1 Byte	u8	0x00

设置自碰撞检测时加入的末端工具几何模型				
寄存器: 78 (0x4E)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x0E <b>(2+x*4)</b>
	寄存器	1 Byte	u8	0x4E
参数	参数 1 (末端工具为长方体) x=20mm, y=30mm, z=50mm 额外定义参数区: x 最大为 6, 实际长度根据工具类型定义需要的参数数目而定, 无参数则此处无数据。  <b>末端工具类型定义:</b> 1) 自定义检测模型 (需要额外定义参数): 圆柱体: 额外定义参数为: 半径 radius (mm), 高度 height (mm) 长方体: 额外定义参数为: 与默认 TCP 坐标系方向一致的长宽 x, y(mm), 高度 z(mm)  2) 已支持的检测模型 (不需额外定义参数): 无末端工具, xArm 机械爪, xArm 吸头, xArm BIO 机械爪, Robotiq 2F-85 机械爪, Robotiq 2F-140 机械爪	12Bytes (x*4 Byte)	3*fp32 (x*fp32)	0x00, 0x00, 0xA0, 0x41  0x00, 0x00, 0xF0, 0x41  0x00, 0x00, 0x48, 0x42

	参数 2 (末端工具类型编号=22) <b>末端工具类型编号:</b> 1) 自定义检测模型 (需要额外定义参数): 圆柱体: 21 长方体: 22  2) 已支持的检测模型 (不需额外定义参数): 无末端工具: 0 xArm 机械爪: 1 xArm 吸头: 2 xArm BIO 机械爪: 3 Robotiq 2F-85 机械爪: 4 Robotiq 2F-140 机械爪: 5	1 Byte	u8	0x16
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x4E
参数	状态	1 Byte	u8	0x00

<b>设置是否开启虚拟机械臂模式</b>				
如果进入仿真模式, 真实机械臂不会动作, 但上报的机械臂位置会随指令变化, 带动虚拟机械臂动作。				
<b>寄存器: 79 (0x4F)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x4F
参数	参数 1 (设为虚拟机械臂模式) 0 为真实机械臂模式 1 为虚拟机械臂模式	1 Byte	u8	0x01
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x4F
参数	状态	1 Byte	u8	0x00

**笛卡尔运动速度连续的全局设置**

寄存器: 80 (0x50)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x50
参数	参数 1 (是否允许) 0 为速度不连续, 默认 1 为速度连续	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x50
参数	状态	1 Byte	u8	0x00

关节速度控制				
设置关节目标速度, 用于关节速度模式 (4)				
寄存器: 81 (0x51)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x22
	寄存器	1 Byte	u8	0x51
参数	参数 1 (关节 1 目标速度 $\pi/6$ rad/s)	4 Bytes	fp32	0x91, 0x0A, 0x06, 0x3F
	参数 2 (关节 2 目标速度 -0.1 rad/s)	4 Bytes	fp32	0xCC, 0xCC, 0xCC, 0xBD
	参数 3 (关节 3 目标速度 0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 (关节 4 目标速度 0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (关节 5 目标速度 0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (关节 6 目标速度 0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (关节 7 目标速度 0 rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8 (所有关节同步加减速: 1: 是)	1 Byte	u8	0x01
	参数 9 (超时时间: 0.2s)	4 Bytes	fp32	0xCC, 0xCC, 0x4C, 0x3E
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x51
参数	状态	1 Byte	u8	0x00

<b>末端笛卡尔速度控制</b>				
设置目标笛卡尔线速度、角速度，用于笛卡尔速度模式 (5)				
<b>寄存器: 82 (0x52)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1E
	寄存器	1 Byte	u8	0x52
参数	参数 1 (笛卡尔线速度 $V_x = 30$ mm/s)	4 Bytes	fp32	0x00, 0x00, 0xF0, 0x41
	参数 2 (笛卡尔线速度 $V_y = 0$ mm/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (笛卡尔线速度 $V_z = 20$ mm/s)	4 Bytes	fp32	0x00, 0x00, 0xA0, 0x41
	参数 4 (笛卡尔角速度 $\omega_x = \pi/6$ rad/s)	4 Bytes	fp32	0x91, 0x0A, 0x06, 0x3F
	参数 5 (笛卡尔角速度 $\omega_y = 0$ rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (笛卡尔角速度 $\omega_z = 0$ rad/s)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (目标坐标系 0: 基坐标系)	1 Byte	u8	0x00
	参数 8 (超时时间: 0.2s)	4 Bytes	fp32	0xCC, 0xCC, 0x4C, 0x3E
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x52
参数	状态	1 Byte	u8	0x00

<b>相对运动控制</b>				
<b>寄存器: 83 (0x53)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x1E
	寄存器	1 Byte	u8	0x53
参数	参数 1 如果是 TCP 控制, 此参数为 x (mm) 如果是关节控制, 此参数为 J1 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 2 如果是 TCP 控制, 此参数为 y (mm) 如果是关节控制, 此参数为 J2 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 3 如果是 TCP 控制, 此参数为 z (mm) 如果是关节控制, 此参数为 J3 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 4 如果是 TCP 控制, 此参数为 roll (rad) 如果是关节控制, 此参数为 J4 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 5 如果是 TCP 控制, 此参数为 pitch (rad) 如果是关节控制, 此参数为 J5 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 6 如果是 TCP 控制, 此参数为 yaw (rad) 如果是关节控制, 此参数为 J6 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 7 如果是 TCP 控制, 此参数为无意义 如果是关节控制, 此参数为 J7 (rad)	4 Bytes	fp32	0x00,0x00,0x00,0x00

	参数 8 速度(mm/s, rad/s)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 9 加速度(mm/s <sup>2</sup> , rad/s <sup>2</sup> )	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 10 运动时间 (无用, 为 0 即可)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 11 交融半径(mm)	4 Bytes	fp32	0x00,0x00,0x00,0x00
	参数 12 是否是关节运动 0: TCP 运动 1: 关节运动	1 Byte	u8	0x00
	参数 13 是否是轴角控制, 仅在 TCP 运动有效 (即参数 12 为 0 时有用) 0: RPY 控制 1: 轴角控制	1 Bytes	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x02
	寄存器	1 Byte	u8	0x53
参数	状态	1 Byte	u8	0x00

获取轴角姿态表示的位姿				
获取当前的 TCP 位姿，姿态使用轴角表示法而不是 Roll/Pitch/Yaw 表示法				
寄存器：91 (0x5B)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x5B
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1A
	寄存器	1 Byte	u8	0x5B
参数	状态	1 Byte	u8	0x00
	参数 1 (当前 TCP 笛卡尔坐标 X=300mm)	4 Bytes	fp32	0x00, 0x00, 0x96, 0x43
	参数 2 (当前 TCP 笛卡尔坐标 Y=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (当前 TCP 笛卡尔坐标 Z=150mm)	4 Bytes	fp32	0x00, 0x00, 0x16, 0x43
	参数 4 (当前 TCP 笛卡尔坐标 $R_x = \pi$ rad)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (当前 TCP 笛卡尔坐标 $R_y = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (当前 TCP 笛卡尔坐标 $R_z = 0$ )	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

轴角姿态目标的直线运动				
规划直线运动，目标位姿使用轴角表示法，支持绝对目标位姿/相对目标位姿，以及基坐标系/工具坐标系运动选项				
寄存器：92 (0x5C)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x27
	寄存器	1 Byte	u8	0x5C



参数	参数 1 (X=300mm)	4 Bytes	fp32	0x00, 0x00, 0x96, 0x43
	参数 2 (Y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (Z=150mm)	4 Bytes	fp32	0x00, 0x00, 0x16, 0x43
	参数 4 (Rx= $\pi$ rad)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (Ry=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (Rz=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (运动速度=200mm/s)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 8 (加速度=2000mm/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 9 (运动时间为 0 即可)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 10 (基坐标系运动) 运动坐标系: 0 代表基坐标系运动 1 代表工具坐标系运动	1 Byte	u8	0x00
	参数 11 (目标为绝对位姿) 如果运动坐标系为基坐标系 0 代表给定目标为绝对位姿 1 代表给定位姿为相对位姿 注: 所给的参数 1-6 坐标是基于当前位置的一个偏移量	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x5C
参数	状态	1 Bytes	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

伺服笛卡尔运动(轴角) 接收高频率连续笛卡尔轨迹运动的接口, 姿态使用轴角表示法				
寄存器: 93 (0x5D)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x26
	寄存器	1 Byte	u8	0x5D
参数	参数 1 (X=300mm)	4 Bytes	fp32	0x00, 0x00, 0x96, 0x43
	参数 2 (Y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3 (Z=150mm)	4 Bytes	fp32	0x00, 0x00, 0x16, 0x43
	参数 4 (Rx= $\pi$ rad)	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5 (Ry=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

	参数 6 (Rz=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (运动速度=200mm/s)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 8 (加速度=2000mm/s <sup>2</sup> )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 10 (基坐标系运动) 运动坐标系: 0 代表基坐标系运动 1 代表工具坐标系运动	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 11 (目标为绝对位姿) 如果运动坐标系为基坐标系 0 代表给定目标为绝对位姿 1 代表给定目标为相对位姿 注: 所给的参数 1-6 坐标是基于当前位置的一个偏移量	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x5D
参数	状态	1 Bytes	u8	0x00

### 101~115 伺服模组

<b>获取当前机械臂伺服的状态</b>				
<b>寄存器: 106 (0x6A)</b>				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x6A
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x13
	寄存器	1 Byte	u8	0x6A
参数	状态	1 Byte	u8	0x00
	参数 1 (正常) 指令执行状态 0: 正常 1: 伺服有错误信息 3: 通信失败	1 Byte	u8	0x00
	参数 2 (第 1 关节伺服状态=正常)	1 Byte	u8	0x00
	参数 3 (第 1 关节伺服错误代码=正常)	1 Byte	u8	0x00
	参数 4 (第 2 关节伺服状态=正常)	1 Byte	u8	0x00
	参数 5 (第 2 关节伺服错误代码=正常)	1 Byte	u8	0x00

参数 6 (第 3 关节伺服状态=正常)	1 Byte	u8	0x00
参数 7 (第 3 关节伺服错误代码=正常)	1 Byte	u8	0x00
参数 8 (第 4 关节伺服状态=正常)	1 Byte	u8	0x00
参数 9 (第 4 关节伺服错误代码=正常)	1 Byte	u8	0x00
参数 10 (第 5 关节伺服状态=正常)	1 Byte	u8	0x00
参数 11 (第 5 关节伺服错误代码=正常)	1 Byte	u8	0x00
参数 12 (第 6 关节伺服状态=正常)	1 Byte	u8	0x00
参数 13 (第 6 关节伺服错误代码=正常)	1 Byte	u8	0x00
参数 14 (第 7 关节伺服状态=正常)	1 Byte	u8	0x00
参数 15 (第 7 关节伺服错误代码=正常)	1 Byte	u8	0x00
参数 16 (机械爪伺服状态=正常)	1 Byte	u8	0x00
参数 17 (机械爪伺服错误代码=正常)	1 Byte	u8	0x00

## 115 关节摩擦力辨识

启动关节摩擦力辨识流程 (推荐使用 Studio 引导进行)				
寄存器: 115 (0x73)				
请求指令格式				
Mod bus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x0F
	寄存器	1 Byte	u8	0x73

参数	参数 1: 待辨识机械臂的序列号 SN (例: XI120307201L1B) ) 参考 ASCII 码	14 Bytes	u8	0x58,0x49,0x31,0x32,0x30,0x33,0x30,0x37,0x32,0x30,0x31,0x4C,0x31,0x42
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x06
	寄存器	1 Byte	u8	0x73
参数	状态	1 Byte	u8	0x00
	参数 1: 辨识状态 0.0: 辨识成功 1.0: 辨识失败	4 Bytes	fp32	0x00,0x00,0x00,0x00

## 2.1.4. 寄存器（机械臂外设控制）

### 124: 机械爪模组

使能/关闭机械爪 (0x7C)

设置机械爪模式 (0x7C)

设置机械爪速度 (0x7C)

设置机械爪位置 (0x7C)

获取机械爪位置 (0x7C)

获取机械爪错误 (0x7C)

清除机械爪错误 (0x7C)

### 124~127: 末端 Modbus RTU 控制

设置末端 RS485 波特率 (0x7F)

### 127~128: 末端 IO 模组

设置末端数字量输出 (0x7F)

获取末端数字量输入 (0x80)

获取末端模拟量输入 (0x80)

### 130~141: 控制器 GPIO 模组

获取可配置数字 gpio 输入 (0x83)

获取模拟输入 AI1 (0x84)

获取模拟输入 AI2 (0x85)

设置可配置数字 gpio 输出 (0x86)

设置模拟输出 A01 (0x87)

设置模拟输出 A02 (0x88)

配置数字输入 IO 功能 (0x89)

配置数字输出 IO 功能 (0x8A)

获取 GPIO 状态 (0x8B)

获取 GPIO 版本 (0x8C)

### 142~146: 特殊 IO 指令

控制器通用数字 IO 延时输出操作 (0x8E)  
 末端通用数字 IO 延时输出操作 (0x8F)  
 控制器通用数字 IO 位置触发操作 (0x90)  
 末端通用数字 IO 位置触发操作 (0x91)  
 控制器、末端 IO 在停止 (STOP) 状态下是否自动清零的设置 (0x92)  
 控制器通用模拟 IO 位置触发操作 (0x93)

## 200~212: 六维力矩传感器

获取六维力矩传感器数据 (外力检测值) (0xC8)  
 使能/关闭六维力矩传感器 (0xC9)  
 设置六维力矩传感器控制模式 (0xCA)  
 获取六维力矩传感器控制模式 (0xCB)  
 执行末端负载辨识 (0xCC)  
 设置六维力矩传感器的负载和偏移参数 (0xCD)  
 设置当前状态为六维力矩传感器零点 (0xCE)  
 设置六维力矩传感器的导纳参数 (控制参数+MKB) (0xCF)  
 设置六维力矩传感器的 PID 参数 (0xD0)  
 设置六维力矩传感器的力控制参数 (0xD1)  
 单独设置六维力矩传感器导纳控制的 MKB 参数 (0xD2)  
 单独设置六维力矩传感器的导纳控制参数 (0xD3)  
 获取六维力矩传感器的全部反馈参数 (0xD4)

## 124 机械爪模组

xArm 机械爪固定参数解释:

参数	主机 ID	机械爪 ID	功能码
长度	1Byte	1Byte	1Byte
固定值	0x09	0x08	0x10/0x03

注: 若为第三方机械爪, 则机械爪 ID 和功能码与上面固定值不同。

使能/关闭机械爪			
寄存器: 124 (0x7C)			
请求指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01,0x00
	寄存器数量	2 Bytes	0x00,0x01
	字节数	1 Byte	0x02
	寄存器 (使能机械爪)	2 Bytes	0x00,0x01
响应指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10

	寄存器起始地址	2 Bytes	0x01,0x00
	寄存器数量	2 Bytes	0x00,0x01

设置机械爪模式			
寄存器：124 (0x7C)			
请求指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01,0x01
	寄存器数量	2 Bytes	0x00,0x01
	字节数	1 Byte	0x02
	寄存器 (位置模式) 0: 位置模式 1: 速度模式	2 Bytes	0x00,0x00
响应指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C



参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01,0x00
	寄存器数量	2 Bytes	0x00,0x01

设置机械爪速度			
寄存器: 124 (0x7C)			
请求指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x03,0x03
	寄存器数量	2 Bytes	0x00,0x01
	字节数	1 Byte	0x02
	寄存器 (设置速度为 1500r/min)	2 Bytes	0x05,0xDC
响应指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01

	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x03,0x03
	寄存器数量	2 Bytes	0x00,0x01

<b>设置机械爪位置</b>			
<b>寄存器: 124 (0x7C)</b>			
<b>请求指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0D
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07,0x00
	寄存器数量	2 Bytes	0x00,0x02
	字节数	1 Byte	0x04
	寄存器 (机械爪位置: 400)	4 Bytes	0x00,0x00,0x01,0x90

响应指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07,0x00
	寄存器数量	2 Bytes	0x00,0x02
<b>获取机械爪位置</b>			
<b>寄存器: 124 (0x7C)</b>			
请求指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x08
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07,0x02
	寄存器数量	2 Bytes	0x00,0x02
<b>响应指令格式</b>			

Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x07,0x02
	寄存器数量	2 Bytes	0x00,0x02

<b>获取机械爪错误</b>			
<b>寄存器：124 (0x7C)</b>			
<b>请求指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x08
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	0x00,0x0F
	寄存器数量	2 Bytes	0x00,0x01
<b>响应指令格式</b>			

Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x08
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	字节数	1 Byte	0x02
	寄存器值 (无错误发生)	2 Bytes	0x00,0x00

<b>清除机械爪错误</b>			
<b>寄存器: 124 (0x7C)</b>			
<b>请求指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0x7C
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01,0x09
	寄存器数量	2 Bytes	0x00,0x01
	字节数	1 Byte	0x02

	寄存器	2 Bytes	0x00,0x01
<b>响应指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x09
	寄存器	1 Byte	0x7C
参数	状态	1 Byte	0x00
内部使用	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	0x01,0x09
	寄存器数量	2 Bytes	0x00,0x01

### 124~127 末端 Modbus RTU 控

<b>设置末端 RS485 波特率</b>				
<b>寄存器: 127 (0x7F)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x08
	寄存器	1 Byte	u8	0x7F
参数	主机 ID	1 Byte	u8	0x09
	地址	2 Bytes	u16	0x1A, 0x0B

	参数 1 (2000000bps) 0:4800 bps; 1:9600bps; 2:19200bps; 3:38400bps; 4:57600bps; 5:115200bps; 6:230400bps; 7: 460800bps; 8:921600bps; 9: 1000000bps; 10:1500000bps; 11:2000000bps; 12:2500000bps;	4 Bytes	fp32	0x00, 0x00, 0x30, 0x41
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x7F

### 127~128 末端 IO 模组

<b>设置末端数字量输出</b>				
<b>寄存器: 127 (0x7F)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x08
	寄存器	1 Byte	u8	0x7F
参数	主机 ID	1 Byte	u8	0x09
	寄存器起始地址	2 Bytes	u16	0x0A, 0x15
	参数 1 (打开 0) 数据 256.0: 关闭 0; 257.0: 打开 0; 512.0: 关闭 1; 514: 打开 1	4 Bytes	fp32	0x00, 0x80, 0x80, 0x43
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x7F
参数	状态	1 Byte	u8	0x00

获取末端数字量输入				
寄存器：128 (0x80)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x80
参数	主机 ID	1 Byte	u8	0x09
	寄存器起始地址	2 Bytes	u16	0x0A, 0x14
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x06
	寄存器	1 Byte	u8	0x80
参数	状态	1 Byte	u8	0x00
	参数 1 (0) 最末端字节，表示输入状态。 第 0 位对应输入 0，第 1 位对应输入 1。	4 Bytes	u8*4	0x00, 0x00, 0x00, 0x00

获取末端模拟量输入				
寄存器：128 (0x80)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x80
参数	主机 ID	1 Byte	u8	0x09
	寄存器起始地址 (输入 0) 地址：0x0A, 0x16: 输入 0 0x0A, 0x17: 输入 1	2 Bytes	u16	0x0A, 0x16
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x06
	寄存器	1 Byte	u8	0x80
参数	状态	1 Byte	u8	0x00



	参数 1 (输入 1) *模拟输入, 范围 0~4096, 对应 0~3.3V	4 Bytes	u32	0x00, 0x00, 0x07, 0x0D
--	---	---------	-----	------------------------

### 131~140 控制器 GPIO 模组

获取可配置数字 gpio 输入				
寄存器: 131 (0x83)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x83
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x83
参数	状态	1 Byte	u8	0x00
	参数 1 (gpio1 的信号为低) gpio 信号, bit0 ~ bit15 分别对 应 gpio0~gpio15 的信号	2 Bytes	u16	0xFF, 0xFD

获取模拟输入 AI1				
寄存器: 132 (0x84)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x84
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x84
参数	状态	1 Byte	u8	0x00
	参数 1 (模拟输入 0) 模拟输入 0, 范围 0~4095, 对应 0~10V	2 Bytes	u16	0x00, 0x12

获取模拟输入 AI2				
寄存器: 133 (0x85)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x85
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x85
参数	状态	1 Byte	u8	0x00
	参数 1 (模拟输入 1) 模拟输入 1, 范围 0~4095, 对应 0~10V	2 Bytes	u16	0x00, 0x15

设置可配置数字 gpio 输出				
寄存器: 134 (0x86)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x86
参数	参数 1 (设置 gpio7 为 0) gpio 信号, 高 8 位是使能位, 低 8 位是设置位	2 Bytes	u16	0x80, 0x00
	参数 2 (设置 gpio15 为 0) gpio 信号, 高 8 位是使能位, 低 8 位是设置位	2 Bytes	u16	0x80, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x86
参数	状态	1 Byte	u8	0x00

设置模拟输出 A01				
寄存器: 135 (0x87)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x87
参数	参数 1 (模拟输出 0 为 0) 模拟输出 0, 范围 0~4095, 对应 0~10V	2 Bytes	u16	0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x87
参数	状态	1 Byte	u8	0x00

设置模拟输出 A02				
寄存器: 136 (0x88)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x88
参数	参数 1 (模拟输出 1 为 0) 模拟输出 1, 范围 0~4095, 对应 0~10V	2 Bytes	u16	0x00, 0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x88
参数	状态	1 Byte	u8	0x00

**配置数字输入 I0 功能**

寄存器：137 (0x89)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x89
参数	参数 1 (gpio15) gpio 编号, 0~15 对应 gpio0 ~ gpio15	1 Byte	u8	0x0F
	参数 2 0: 通用输入 1: 外部急停 2: 防护重置 11: 离线任务 12: 示教模式 13: 缩减模式 14: 使能机械臂	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x89
参数	状态	1 Byte	u8	0x00

配置数字输出 IO 功能				
寄存器：138 (0x8A)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x8A
参数	参数 1 (gpio15) gpio 编号, 0~15 对应 gpio0 ~ gpio15	1 Byte	u8	0x0F

	参数 2 (系统处在 STOP 状态) 功能号 0: 通用输出 1: 机械臂处于停止状态 2: xArm 运动中 11: 有错误发生 12: 有警告发生 13: 发生碰撞 14: 拖动示教模式生效 15: 正在运行离线任务 16: 缩减模式生效 17: 机械臂已使能 18: 控制器急停按下	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x8A
参数	状态	1 Byte	u8	0x00

<b>获取 GPIO 状态</b>				
<b>寄存器: 139 (0x8B)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0x8B
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x24
	寄存器	1 Byte	u8	0x8B
参数	状态	1 Byte	u8	0x00
	GPIO 模块状态 0 : 正常 3 : 爪子 有错误信息 6 : 通信失败	1 Byte	u8	0x00
	GPIO 模块的错误代码 0: 正常 非 0: 错误代码	1 Byte	u8	0x00
	辅助输入 I0 状态	2 Bytes	u16	0x01, 0x00
	数字输入 I0 状态	2 Bytes	u16	0xFF, 0xFD

	数字输出功能 I0 状态	2 Bytes	u16	0x00, 0x00
	数字输出配置 I00-I015 状态	2 Bytes	u16	0xFF, 0x00
	模拟输入 1	2 Bytes	u16	0x00, 0x11
	模拟输入 2	2 Bytes	u16	0x00, 0x15
	模拟输出 1	2 Bytes	u16	0x00, 0x00
	模拟输出 2	2 Bytes	u16	0x00, 0x00
	数字输入 I00-I07 配置信息	1 Byte*8	u8*8	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	数字输出 I00-I07 配置信息	1 Byte*8	u8*8	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	数字输入 I08-I015 配置信息	1 Byte*8	u8*8	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	数字输出 I08-I015 配置信息	1 Byte*8	u8*8	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

<b>获取 GPIO 版本</b>				
<b>寄存器: 140 (0x8C)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x01
	寄存器	1 Byte	u8	0x8C
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00,0x01
	协议	2 Bytes	u16	0x00,0x02
	长度	2 Bytes	u16	0x00,0x08
	寄存器	1 Byte	u8	0x8C
参数	状态	1 Byte	u8	0x00
	电源板版本第一位	1 Byte	u8	0x01
	电源板版本第二位	1 Byte	u8	0x01
	电源板版本第三位	1 Byte	u8	0x02

	安全板版本第一位	1 Byte	u8	0x04
	安全板版本第二位	1 Byte	u8	0x06
	安全板版本第三位	1 Byte	u8	0x01

## 142~146 特殊 IO 指令

控制器通用数字 IO 延时输出操作				
由指令时刻开始算起，一段时间后触发某一控制器数字输出开关。				
寄存器 142 (0x8E)				
请求指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x07
	寄存器	1 Byte	u8	0x8E
	参数 1 (控制器数字 IO 端口号: 0~7)	1 Byte	u8	0x00
	参数 2 (打开) 开关量: 0 为关; 1 为开。	1 Byte	u8	0x01
	参数 3 (从当前开始延时生效时间=3s)	4	fp32	0x00, 0x00, 0x40, 0x40
响应指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x8E
参数	状态	1 Byte	u8	0x00

末端通用数字 IO 延时输出操作				
由指令时刻开始算起，一段时间后触发某一末端数字输出开关。				
寄存器 143 (0x8F)				
请求指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x07
	寄存器	1 Byte	u8	0x8F
	参数 1 (0) (末端 IO 端口号: 0~7)	1 Byte	u8	0x00

	参数 2 (打开) 开关量: 0 为关; 1 为开。	1 Byte	u8	0x01
	参数 3 (从当前开始延时生效时间=3s)	4	fp32	0x00, 0x00, 0x40, 0x40
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x8F
参数	状态	1 Byte	u8	0x00

<b>控制器通用数字 IO 位置触发操作</b>				
由指令时刻开始, TCP 到达指定位置区域后触发某一控制器数字输出开关, 单次有效。				
<b>寄存器 144 (0x90)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x13
	寄存器	1 Byte	u8	0x90
	参数 1 (0) (末端 IO 端口号: 0~7)	1 Byte	u8	0x00
	参数 2 (打开) 开关量: 0 为关; 1 为开。	1 Byte	u8	0x01
	参数 3 (X=400mm)	4	fp32	0x00, 0x00, 0xC8, 0x43
	参数 4 (Y=0mm)	4	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (Z=200mm)	4	fp32	0x00, 0x00, 0x48, 0x43
	参数 6 (容错半径=50) 容错半径(mm), 当机械臂到达指定位置时 (以触发位置点 (x, y, z) 为中心指定的 球面体区域 (球面体半径为容错半 径)), 触发 IO。如果不设置容错半径, 则机械臂以非 0 的速度通过指定点时可能 因无法准确检测到而导致漏触发。	4 Bytes	fp32	0x00, 0x00, 0x48, 0x42
	<b>响应指令格式</b>			
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x90
参数	状态	1 Byte	u8	0x00



末端通用数字 IO 位置触发操作				
寄存器 145 (0x91)				
请求指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x13
	寄存器	1 Byte	u8	0x91
	参数 1 (0) (末端 IO 端口号: 0/1)	1 Byte	u8	0x00
	参数 2 (打开) 开关量: 0 为关; 1 为开。	1 Byte	u8	0x01
	参数 3 (X=400mm)	4	fp32	0x00, 0x00, 0xC8, 0x43
	参数 4 (Y=0mm)	4	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (Z=200mm)	4	fp32	0x00, 0x00, 0x48, 0x43
	参数 6 (容错半径=50mm) 容错半径(mm), 当机械臂到达指定位置时 (以触发位置点 (x, y, z) 为中心指定的 球面体区域 (球面体半径为容错半 径)), 触发 IO。如果不设置容错半径, 则机械臂以非 0 的速度通过指定点时可能 因无法准确检测到而导致漏触发。	4 Bytes	fp32	0x00, 0x00, 0x48, 0x42
响应指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x91
参数	状态	1 Byte	u8	0x00

控制器、末端 IO 在停止(STOP)状态下是否自动清零的设置				
寄存器 146 (0x92)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x92
	参数 1 (控制器 IO) IO 类型: 0 : 控制器 IO, 1 : 末端 IO	1 Byte	u8	0x00
	参数 2 (打开) 开关量: 0 为关, 停止 (STOP) 状态不清零。 1 为开, 停止 (STOP) 状态清零。	1 Byte	u8	0x01
响应指令格式				

Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x92
参数	状态	1 Byte	u8	0x00
	参数 1 (指令缓冲区中的指令数量)	2 Bytes	u16	0x00, 0x01

控制器通用模拟 IO 位置触发操作				
由指令时刻开始, TCP 到达指定位置区域后触发某一控制器模拟输出开关, 单次有效。				
寄存器 147 (0x93)				
请求指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x14
	寄存器	1 Byte	u8	0x93
	参数 1 (0) (控制器 IO 端口号: 0/1)	1 Byte	u8	0x00
	参数 2 (模拟输出 0 为 0) 模拟输出 0, 范围 0~4095, 对应 0~10V。	2 Bytes	u16	0x00, 0x00
	参数 3 (X=400mm)	4	fp32	0x00, 0x00, 0xC8, 0x43
	参数 4 (Y=0mm)	4	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (Z=200mm)	4	fp32	0x00, 0x00, 0x48, 0x43
	参数 6 (容错半径=50) 容错半径(mm), 当机械臂到达指定位置时 (以触发位置点 (x, y, z) 为中心指定的 球面体区域 (球面体半径为容错半 径)), 触发 IO。如果不设置容错半径, 则机械臂以非 0 的速度通过指定点时可能 因无法准确检测到而导致漏触发。	4 Bytes	fp32	0x00, 0x00, 0x48, 0x42
响应指令格式				
Modbus TCP 包头	事务标识	2	u16	0x00, 0x01
	协议	2	u16	0x00, 0x02
	长度	2	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x93
参数	状态	1 Byte	u8	0x00

获取六维力矩传感器数据				
经过滤波、负载和偏置补偿的末端六维力/力矩传感器的外力检测值				
寄存器 200 (0xC8)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0xC8
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x1A
	寄存器	1 Byte	u8	0xC8
参数	状态	1 Byte	u8	0x00
	参数 1 (传感器数据)	24 Bytes	fp32*6	0x00, 0x00

使能/关闭六维力矩传感器				
寄存器 201 (0xC9)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xC9
参数	参数 1 (使能/关闭) 0-关闭; 1-使能	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xC9
	状态	1 Byte	u8	0x00

设置六维力矩传感器控制模式				
寄存器 202 (0xCA)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCA
参数	参数 1 (控制模式) 0-关闭; 1-导纳控制; 2-力控模式	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCA
	状态	1 Byte	u8	0x00

获取六维力矩传感器控制模式				
寄存器 203 (0xCB)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0xCB
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0xCB
参数	状态	1 Byte	u8	0x00
	参数 1 (力矩控制模式) 0-关闭; 1-导纳控制; 2-力控模式	1 Byte	u8	0x00

执行末端负载辨识				
寄存器 204 (0xCC)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCC
参数	参数 1 (辨识类型) 0-力矩传感器辨识; 1-电流辨识	1 Byte	u8	0x00
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	辨识类型 0: 0x00, 0x2A 辨识类型 1: 0x00, 0x12
	寄存器	1 Byte	u8	0xCC
参数	状态	1 Byte	u8	0x00
	参数 1 (辨识结果) 当辨识类型为 0 时, N=10。 [负载质量 M(Kg), 质心偏移 Cx, Cy, Cz (mm), 6 维力矩传感器读数偏移量 Fx0, Fy0, Fz0 (N), Tx0, Ty0, Tz0 (Nm)] 当辨识类型为 1 时, N=4。[负载质量 M(Kg), 质心偏移 Cx, Cy, Cz (mm)]	4 * N Bytes	fp32	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 ... 0x00, 0x00, 0x00, 0x00

设置六维力矩传感器的负载和偏移参数				
寄存器 205 (0xCD)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x29
	寄存器	1 Byte	u8	0xCD
参数	参数 1 (质量, kg)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 2 (质心 x, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

	参数 3 (质心 y, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 4 (质心 z, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 5 (偏移 Fx, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6 (偏移 Fy, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (偏移 Fz, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 8 (偏移 Tx, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 9 (偏移 Ty, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 10 (偏移 Tz, mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCD
参数	状态	1 Byte	u8	0x00

<b>设置当前状态为六维力矩传感器零点</b>				
<b>寄存器 206 (0xCE)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0xCE
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCE
参数	状态	1 Byte	u8	0x00

<b>设置六维力矩传感器的导纳参数 (控制参数+MKB)</b>				
<b>寄存器 207 (0xCF)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x50
	寄存器	1 Byte	u8	0xCF

参数	参数 1 (坐标系) 0-基坐标系; 1-工具坐标系	1 Byte	u8	0x00
	参数 2 (阻抗控制向量) 对应方向的值为 1 表示该方向将产生阻抗	6 Bytes	u8*6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	参数 3 (等效质量和转动惯量 M) M => [Mx, My, Mz, Mr, Mp, My'] 等效质量 (xyz 方向): 0.02 ~ 1.0 (kg) 转动惯量 (rpy 方向): 0.0001 ~ 0.01 (kg*m <sup>2</sup> )	24 Bytes	fp32*6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	参数 4 (刚度向量 K) K => [Kx, Ky, Kz, Kr, Kp, Ky'] 刚度 (xyz 方向): 0 ~ 2000 (N/m) 刚度 (rpy 方向): 0 ~ 20 (Nm/rad)	24 Byte	fp32 * 6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	参数 5 (阻尼系数向量 B) B[i]为非负数	24 Byte	fp32*6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xCF
参数	状态	1 Byte	u8	0x00

设置六维力矩传感器的 PID 参数				
寄存器 208 (0xD0)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x61
	寄存器	1 Byte	u8	0xD0
参数	参数 1 (比例增益向量 KP) KP[i]: 0~0.05	24 Bytes	fp32*6	0x00, 0x00
	参数 2 (积分增益向量 KI) KI[i]: 0~0.0005	24 Bytes	fp32*6	0x00, 0x00
	参数 3 (微分增益向量 KD) KI[i]: 0~0.05	24 Bytes	fp32*6	0x00, 0x00
	参数 4 (最大 TCP 速度矢量 VMAX) 对应方向的值表示沿/围绕对应轴的最 大允许 TCP 速度 VMAX[i]: 0~200 (毫米/秒)	24 Byte	fp32 * 6	0x00, 0x00



响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xD0
参数	状态	1 Byte	u8	0x00

设置六维力矩传感器的力控制参数				
寄存器 209 (0xD1)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x38
	寄存器	1 Byte	u8	0xD1
参数	参数 1 (坐标系) 0-基坐标系; 1-工具坐标系	1 Byte	u8	0x00
	参数 2 (力控向量) 对应方向的值为 1 表示该方向可以力控	6 Bytes	u8*6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00
	参数 3 (力向量 F) $F \Rightarrow [F_x, F_y, F_z, F_r, F_p, F_{y'}]$ F[i]: 机械臂沿/围绕对应轴调整其位置, 以达到指定的力/扭矩 F <sub>x</sub> : -150 ~ 150 (N) F <sub>y</sub> : -150 ~ 150 (N) F <sub>z</sub> : -200 ~ 200 (N) F <sub>r</sub> : -4 ~ 4 (Nm) F <sub>p</sub> : -4 ~ 4 (Nm) F <sub>y'</sub> : -4 ~ 4 (Nm)	24 Bytes	fp32*6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

	<p>参数 4 (最大 TCP 速度矢量 VMAX)</p> <p>对应方向的值表示沿/围绕对应轴的最大允许 TCP 速度</p> <p>VMAX[i]: 0~200 (毫米/秒)</p>	24 Byte	fp32 * 6	<p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00</p>
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xD1
参数	状态	1 Byte	u8	0x00

<b>单独设置六维力矩传感器导纳控制的 MKB 参数</b>				
<b>寄存器 210 (0xD2)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x49
	寄存器	1 Byte	u8	0xD2
参数	<p>参数 1 (等效质量和转动惯量 M)</p> <p>M =&gt; [Mx, My, Mz, Mr, Mp, My']</p> <p>等效质量 (xyz 方向): 0.02 ~ 1.0 (kg)</p> <p>转动惯量 (rpy 方向): 0.0001 ~ 0.01 (kg*m<sup>2</sup>)</p>	24 Bytes	fp32 * 6	<p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00</p>

	<p>参数 2 (刚度向量 K)</p> <p><math>K \Rightarrow [K_x, K_y, K_z, K_r, K_p, K_y']</math></p> <p>刚度 (xyz 方向): 0 ~ 2000 (N/m)</p> <p>刚度 (rpy 方向): 0 ~ 20 (Nm/rad)</p>	24 Bytes	fp32 * 6	<p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00</p>
	<p>参数 3 (阻尼系数向量 B)</p> <p>B[i]为非负数</p>	24 Bytes	fp32 * 6	<p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00,</p> <p>0x00, 0x00, 0x00, 0x00</p>
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xD2
参数	状态	1 Byte	u8	0x00

<b>单独设置六维力矩传感器导纳控制参数</b>				
<b>寄存器 211 (0xD3)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x08
	寄存器	1 Byte	u8	0xD3
参数	<p>参数 1 (坐标系)</p> <p>0-基坐标系; 1-工具坐标系</p>	1 Byte	u8	0x00

	参数 2 (阻抗控制向量) 对应方向的值为 1 表示该方向将产生 抗阻	6 Bytes	u8* 6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0xD3
参数	状态	1 Byte	u8	0x00

<b>获取六维力矩传感器的全部反馈参数</b>				
<b>寄存器 212 (0xD4)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x01
	寄存器	1 Byte	u8	0xD4
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x01, 0x1A
	寄存器	1 Byte	u8	0xD4
参数	状态	1 Byte	u8	0x00
	参数 1 (力矩控制模式)	1 Byte	u8	0x00
	参数 2 (力矩使能状态)	1 Byte	u8	0x00
	参数 3 (力矩传感器类型)	1 Byte	u8	0x00
	参数 4 (力矩传感器 ID)	1 Byte	u8	0x08
	参数 5 (力矩传感器频率)	2 Bytes	u16	0x03, 0xE8
	参数 6 (质量)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 7 (保留)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
参数 8 (质心)	12 Bytes	fp32 * 3	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00	

参数 9 (偏移)	24 Bytes	fp32 * 6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 10 (导纳控制坐标系)	1 Byte	u8	0x00
参数 11 (阻抗控制向量)	6 Bytes	u8 * 6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00
参数 12 (等效质量和转动惯量 M)	24 Bytes	fp32 * 6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 13 (刚度 K)	24 Bytes	fp32 * 6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 14 (阻尼 B)	24 Bytes	fp32 * 6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 15 (力控坐标系)	1 Byte	u8	0x00
参数 16 (力控向量)	6 Bytes	u8 * 6	0x00, 0x00, 0x00, 0x00, 0x00, 0x00

参数 17 (力向量)	24 Bytes	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 18 (保留)	24 Bytes	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 19 (比例增益 KP)	24 Bytes	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 20 (积分增益 KI)	24 Bytes	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
参数 21 (微分增益 KD)	24 Bytes	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00

	参数 22 (最大 TCP 速度向量)	24 Byte	fp32 *6	0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00 0x00, 0x00, 0x00, 0x00
--	---------------------	---------	------------	--

### 240~241 485 外设透传

<b>外设 485 透传超时</b>			
<b>寄存器: 240 (0xF0)</b>			
<b>请求指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0xF0
参数	超时时间 (s)	1 Byte	0x01
<b>响应指令格式</b>			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x08
	寄存器	1 Byte	0xF0
参数	状态	1 Byte	0x00

外设 485 透传通信			
寄存器: 241 (0xF1)			
请求指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x0B
	寄存器	1 Byte	0xF1
内部使用	主机 ID: 末端 485: 0x09 控制器 485: 0x0a	1 Byte	0x09
485 数据	用户数据	N Byte	0x00
响应指令格式			
Modbus TCP 包头	事务标识	2 Bytes	0x00,0x01
	协议	2 Bytes	0x00,0x02
	长度	2 Bytes	0x00,0x08
	寄存器	1 Byte	0xF1
参数	状态	1 Byte	0x00
内部使用	主机 ID: 末端 485: 0x09 控制器 485: 0x0a	1 Byte	0x09
485 数据	用户数据	N Byte	0x00



## 2.1.5. Modbus TCP 示例

如果想让机械臂执行一个基本运动，请按以下步骤发送指令：

- (1) 使能机械臂
- (2) 设置机械臂运动模式
- (3) 设置机械臂运动状态
- (4) 发送运动指令

下面将按以上步骤进行举例说明：

功能	<a href="#">使能机械臂</a>	<a href="#">设置模式</a>	<a href="#">设置状态</a>	<a href="#">笛卡尔直线运动</a>
----	-----------------------	----------------------	----------------------	-------------------------

注：（1）下面事例中的请求和响应指令参数的格式请参照 P27-P28。

（2）下面用到的符号的解释：

u8（1 Byte，8 位无符号整型）      u16（2 Bytes，16 位无符号整型，大端解析）

fp32（4 Bytes，浮点型，小端解析）、str（字符串）

### 使能机械臂：

使能机械臂				
寄存器 11 (0x0B)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x03
	寄存器	1 Byte	u8	0x0B
	参数 1 (servo_id)	1 Byte	u8	0x08
	参数 2 (enable)	1 Byte	u8	0x01
响应指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0B
参数	状态	1 Byte	u8	0x00

### 设置模式：

设置模式				
寄存器 19 (0x13)				
请求指令格式				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01

	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x13
	参数 1(位置控制模式)	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x13
参数	状态	1 Byte	u8	0x00

### 设置状态:

<b>设置状态</b>				
<b>寄存器 12 (0x0C)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0C
	参数 1(运动状态)	1 Byte	u8	0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x02
	寄存器	1 Byte	u8	0x0C
参数	状态	1 Byte	u8	0x00

### 笛卡尔直线运动:

<b>笛卡尔直线运动</b>				
<b>寄存器 21 (0x15)</b>				
<b>请求指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x25
	寄存器	1 Byte	u8	0x15
参数	参数 1(x=400mm)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x43
	参数 2(y=0mm)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 3(z=200mm)	4 Bytes	fp32	0x00, 0x00, 0x48, 0x43
	参数 4(roll= $\pi$ )	4 Bytes	fp32	0xDB, 0x0F, 0x49, 0x40
	参数 5(pitch=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
	参数 6(yaw=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00

	参数 8(速度=100mm/s)	4 Bytes	fp32	0x00, 0x00, 0xC8, 0x42
	参数 9(加速度 2000mm/s <sup>2</sup> ) =500* ( $\frac{2000}{1000} \times 1000$ )	4 Bytes	fp32	0x00, 0x00, 0xFA, 0x44
	参数 10(运动时间=0)	4 Bytes	fp32	0x00, 0x00, 0x00, 0x00
<b>响应指令格式</b>				
Modbus TCP 包头	事务标识	2 Bytes	u16	0x00, 0x01
	协议	2 Bytes	u16	0x00, 0x02
	长度	2 Bytes	u16	0x00, 0x04
	寄存器	1 Byte	u8	0x15
参数	状态	1 Byte	u8	0x00
	参数 1	2 Bytes	u16	0x00, 0x01

## 2.1.6. 自动上报数据格式

### REPORT\_TCP\_DEVELOP:

REPORT_TCP_DEVELOP			
默认端口	30003		
频率	100Hz		
字节顺序内容	1~4 Bytes		字节数
	5 Byte	u8	Bit0~Bit3 表示运动状态, Bit4~Bit7 表示运动模
	6~7 Bytes	u16	指令缓存数
	8~35 Bytes	fp32	机械臂当前各个关节的角度
	36~59	fp32	机械臂当前的位置和姿态
	60~87	fp32	机械臂关节力矩
	88~111	fp32	经过滤波、负载和偏置补偿的末端六维力/力矩传感器的外力检测值
	112~135	fp32	末端六维力/力矩传感器的直接读数, 未经任何处理
示例			
假如获取 36~59 Bytes 的数据	0x18, 0x00, 0x4F, 0x43, 0x24, 0xFC, 0x8A, 0x28, 0x08, 0x01, 0xE0, 0x42 0xDB, 0x0F, 0x49, 0xC0, 0x00, 0x00, 0x00, 0x24, 0x00, 0x00, 0x00, 0x00,		
解析结果	0x18, 0x00, 0x4F, 0x43	207.0003662109375	
	0x24, 0xFC, 0x8A, 0x28	1.54304263051859e-14	
	0x08, 0x01, 0xE0, 0x42	112.00201416015625	
	0xDB, 0x0F, 0x49, 0xC0	3.1415927410125732	
	0x00, 0x00, 0x00, 0x24	2.7755575615628914e-17	
	0x00, 0x00, 0x00, 0x00	0.0	

### REPORT\_TCP\_NORMAL:

REPORT_TCP_NORMAL
-------------------

默认端口	30001		
频率	5Hz		
字节顺序内容	1~87 Bytes		同【REPORT_TCP_DEVELOP 的自动上报格式】
	88 Bytes	u8	伺服抱闸状态 Bit0 ~ Bit7 : 分别对应 1~6 关节 置 0 未使能, 置 1 使能
	89 Bytes	u8	伺服抱闸状态 Bit0 ~ Bit7 : 分别对应 1~6 关节 置 0 未使能, 置 1 使能
	90 Bytes	u8	错误代码
	91 Bytes	u8	警告代码
	92~115 Bytes	fp32 *6	TCP 偏移量
	116~131 Bytes	fp32 *4	末端负载参数
	132 Bytes	u8	碰撞检测灵敏度
	133 Bytes	u8	拖动示教灵敏度
134~145 Bytes	fp32 *3	为表示重力方向的单位向量 ( x, y, z) , 相对于基坐标系。	
示例			
同【REPORT_TCP_DEVELOP】			

### REPORT\_TCP\_RICH:

REPORT_TCP_RICH			
默认端口	30002		
频率	5Hz		
字节顺序内容	1~145 Bytes		同【REPORT_TCP_DEVELOP 的自动上报格式】
	146 Bytes	u8	机械臂类型编号 (5/6/7)
	147 Bytes	u8	机械臂关节编号 (5/6/7)
	148 Bytes	u8	通信 MASTER ID (0xAA 固定)
	149 Bytes	u8	通信 SLAVE ID (0x55 固定)
	150 Bytes	0	预留
	151 Bytes	0	预留
	152~181 Bytes		固件版本号字符串 (30 字节)
	182~201Bytes	fp32 *5	当前设定的笛卡尔加加速度( $\text{mm/s}^3$ ), (可设置的) 笛卡尔最小加速度( $\text{mm/s}^2$ ), (可设置的) 笛卡尔最大加速度( $\text{mm/s}^2$ ), (可设置的) 笛卡尔最小速度( $\text{mm/s}$ ), (可设置的) 笛卡尔最大速度( $\text{mm/s}$ )
202~221 Bytes	fp32 *5	当前设定的关节加加速度( $\text{radian/s}^3$ ), (可设置的) 关节最小加速度( $\text{radian/s}^2$ ), (可设置的) 关节最大加速度( $\text{radian/s}^2$ ), (可设置的) 关节最小速度( $\text{radian/s}$ ), (可设置的)	

			关节最大速度 (radian/s)
222~229 Bytes	fp32 * 2		[姿态旋转加速度 (radian/s <sup>3</sup> ), 姿态旋转加速度 (radian/s <sup>2</sup> )] 注: 用户不可自己设置以上两个参数值。
230~243 Bytes	u8		关节伺服错误类型, 关节伺服错误代码
244~245 Bytes	u8		末端 I0 错误类型, 末端 I0 错误代码
246~252 Bytes	int8		关节摄氏温度
253~256 Bytes	fp32		控制器规划的笛卡尔运动指令的 TCP 速度 (mm/s)
257~284 Bytes	fp32 * 7		控制器规划的关节运动指令的角速度 (rad/s) 注意: servoj 不经过控制器运动规划, 如果用 servoj 控制, 用该方法无法获得速度值。
285~288 Bytes	u32		当前指令计数器的值
289~312 Bytes	fp32 * 6		用户坐标系偏移量 [x (mm), y (mm), z (mm), roll (radian), pitch (radian), yaw (radian)]
313 Bytes	u8		控制器 I0 停止状态清零开关量
314 Bytes	u8		末端 I0 停止状态清零开关量
315 Bytes	u8		虚拟控制开关量
316 Bytes	u8		自碰撞检测开关量
317 Bytes	u8		自碰撞检测末端工具类型编号
318~341 Bytes	fp32 * 6		自碰撞检测末端工具模型参数 单位: 毫米 (mm), 小端字节序
342~355 Bytes	u16*7		机械臂关节电压 (数值已做 X100 处理)
356~383 Bytes	fp32 * 7		关节电流, 单位: 安培 (A)
384 Bytes	u8		GPIO 模块状态 (参照 139 接口) 0: 正常 3: 爪子有错误信息 6: 通信失败
385 Bytes	u8		GPIO 模块的错误代码 (参照 139 接口) 0: 正常 非 0: 错误代码
386~387 Bytes	u16		数字输入功能 I0 状态 (参照寄存器: 139)
388~389 Bytes	u16		数字输入配置 I0 状态 (参照寄存器: 139)
390~391 Bytes	u16		数字输出功能 I0 状态 (参照寄存器: 139)
392~393 Bytes	u16		数字输出配置 I0 状态 (参照寄存器: 139)
394~395 Bytes	u16		模拟输入 1 (参照寄存器: 139)
396~397 Bytes	u16		模拟输入 2 (参照寄存器: 139)
398~399 Bytes	u16		模拟输出 1 (参照寄存器: 139)
400~401 Bytes	u16		模拟输出 2 (参照寄存器: 139)

	402~409 Bytes	u8*8	数字输入 I00~I07 配置信息 (参照寄存器: 139)
	410~417 Bytes	u8*8	数字输出 I00~I07 配置信息 (参照寄存器: 139)
	418~425 Bytes	u8*8	数字输入 I08~I015 配置信息 (参照寄存器: 139)
	426~433 Bytes	u8*8	数字输出 I08~I015 配置信息 (参照寄存器: 139)
	434~457 Bytes	fp32*6	经过滤波、负载和偏置补偿的末端六维力/力矩传感器的外力检测值。单位 (N, N, N, Nm, Nm, Nm)
	458~481 Bytes	fp32*6	末端六维力/力矩传感器的直接读数, 未经任何处理。单位 (N, N, N, Nm, Nm, Nm)
	482 Byte	u8	自动辨识程序完成进度 (百分比)
	483~494 Bytes	fp32*3	当前末端姿态 (轴角表示法)
示例			
同【REPORT_TCP_DEVELOP】			

## 3. 报错及处理

### 3.1. 关节报警信息和常规处理方式

报警处理方式可采用重新上电，步骤如下（重新上电需要走完以下所有步骤）：

1. 通过控制器上的紧急停止按钮重新对机械臂上电
2. 使能机械臂

xArm Studio 使能方式：点击报错弹窗的引导按钮或者首页的使能按钮。

xArm-Python-SDK 使能方式：[参照报警处理方式](#)。

xarm\_ros 库：查看相关文档 [https://github.com/xArm-Developer/xarm\\_ros](https://github.com/xArm-Developer/xarm_ros)

若多次上电无效后请寻找 UFACTORY 团队支持，至邮件：[support@ufactory.cc](mailto:support@ufactory.cc)。

软件报错代码	报警代码	报警处理
S0	0x00	关节通信失败 请通过控制器上的紧急停止按钮重启机械臂。如多次重启无效，请联系技术支持。
S10	0x0A	电流检测异常 请通过控制器上的紧急停止按钮重启机械臂
S11	0x0B	关节电流过大 请通过控制器上的紧急停止按钮重启机械臂
S12	0x0C	关节速度过大 请通过控制器上的紧急停止按钮重启机械臂
S14	0x0E	位置指令过大 请通过控制器上的紧急停止按钮重启机械臂
S15	0x0F	关节过热 如果机械臂长时间运行温度过高，请停机冷却后重启机械臂
S16	0x10	编码器初始化异常 请确保机械臂通电时，无外力推动机械臂运动。请通过控制器上的紧急停止按钮重启机械臂
S17	0x11	单圈编码器错误 请通过控制器上的紧急停止按钮重启机械臂，如多次重启无效，请联系技术支持。
S18	0x12	多圈编码器错误 请进入“设置-高级设置-高级工具-关节工具-关节调参”，点击“清除多圈错误”，随后将控制器电源开关推至 OFF 档，等待 5 秒后重新上电。如多次尝试无效，请联系技术支持。
S19	0x13	电池电压过低
S20	0x14	驱动 IC 硬件异常 请重新使能机械臂。
S21	0x15	驱动 IC 初始化异常 请通过控制器上的紧急停止按钮重启机械臂
S22	0x16	编码器配置错误

S23	0x17	电机位置偏差过大 请检查机械臂运动是否受阻，末端负载是否超过机械臂额定负载，机械臂加速度值是否设置过大
S26	0x1A	第 N 关节正向超限 请检测 N 关节角度值是否设置过大
S27	0x1B	第 N 关节负向超限 请检测第 N 关节角度值是否设置过大，如果是，请点击清除报错后，手动解锁该关节并转动该关节至其运动范围内
S28	0x1C	关节指令错误 机械臂未使能，请点击“使能机械臂”
S33	0x21	驱动器过载， 请确保机械臂负载处于额定负载内
S34	0x22	电机过载 请确保机械臂负载处于额定负载内
S35	0x23	电机类型错误 请通过控制器上的紧急停止按钮重启机械臂
S36	0x24	驱动器类型错误 请通过控制器上的紧急停止按钮重启机械臂
S39	0x27	关节过压 请在运动设置中减少加速度值
S40	0x28	关节欠压 请在运动设置中减少加速度值。请检查控制器紧急停止开关是否松开
S49	0x31	EEPROM 读写错误 请通过控制器上的紧急停止按钮重启机械臂
S52	0x34	电机角度初始化失败 请通过控制器上的紧急停止按钮重启机械臂
上表中未出现的报警代码：重新上电。如多次上电无效，请联系技术支持。		

## 3.2. 控制器报警信息及常规处理方式

### 3.2.1. 控制器错误代码

机器人硬件出现错误、控制软件出现错误、下发指令错误，会发出错误或警告，这个错误/警告信号会在用户发送任意指令时反馈回去，也就是这个反馈是被动的，并非主动上报。

注意：

出现以下错误后，机器人会立即停止工作，并且丢弃控制器缓存指令。需手动清除错误后，机器人方可继续正常工作。根据上报的错误信息，重新调整机器人的运动规划。

软件报错代码	错误代码	报警处理
C1	0x01	控制器上的紧急停止按钮被按下 请释放紧急停止按钮，然后点击“使能机械臂”。
C2	0x02	控制器上的紧急停止 IO 被触发



		请将控制器的 2 组 EI 接地，然后点击“使能机械臂”。
C3	0x03	三态开关的紧急停止按钮被按下 请释放三态开关的紧急停止按钮，然后点击“使能机械臂”。
C11-C17	0x0B-0x11	请通过控制器上的紧急停止按钮重启机械臂。
C18	0x12	力矩传感器通信失败 前检查力矩传感器是否安装
C19	0x13	末端通信失败 请检测机械爪是否安装，波特率设置是否正确。
C21	0x15	运动学错误 请重新规划路径。
C22	0x16	自碰撞错误 机械臂即将发生自碰撞，请重新规划路径。如果机械臂持续报自碰撞错误，请到“实时控制”界面开启“手动模式”，将机械臂拖回正常位置。
C23	0x17	关节角度超出限制 请点击“零点”按钮回到关节零点。
C24	0x18	速度超出限制 请检查机械臂是否超出运动范围，或减小运动速度和加速度。
C25	0x19	规划错误 请重新规划路径或者减小运动速度。
C26	0x1A	Linux RT 错误
C27	0x1B	回复指令错误 请重试，或通过控制器上的紧急停止按钮重启机械臂。
C29	0x1D	其他错误
C30	0x1E	反馈速度超出限制
C31	0x1F	机械臂电流异常 1. 检查机械臂是否发生碰撞； 2. 检查“设置”-“TCP 设置”-“TCP 负载”处设置的质量、质心是否与实际负载匹配； 3. 检查“设置”-“安装方式”处设置的安装方向是否与实际匹配； 4. 检查程序中设置的 TCP 负载参数是否与实际匹配； 5. 降低机械臂运动速度； 6. 前往“设置”-“运动”-“灵敏度设置”，调低碰撞检测灵敏度。
C32	0x20	三点圆弧指令计算出错 请重新设置圆弧指令
C33	0x21	控制器 GPIO 模块异常
C34	0x22	轨迹录制超时 轨迹录制时间超过最大限制 5 分钟，建议重新录制
C35	0x23	机械臂到达安全边界 机械臂到达安全边界，请让机械臂在安全边界内运行
C36	0x24	延时指令数量超限 1. 请检查位置检测或 IO 延时指令是否过多。 2. 增加位置检测指令的容错半径。
C37	0x25	手动模式运动异常（负载设置或安装方式设置错误） 请检查机械臂的 TCP 负载设置和机械臂安装方式设置是否与实际匹配。
C38	0x26	关节角度异常 请通过控制器上的紧急停止按钮停止机械臂，并联系技术支持。
C39	0x27	电源板主从 IC 通信异常

		请联系技术支持。
C50	0x32	六维力矩传感器错误 请检查传感器错误码，定位问题并重新上电。如果无法解决，请联系技术支持。
C51	0x33	六维力矩传感器模式设置错误 请确保机械臂没有处于手动模式，检查本指令给定值是否为0/1/2
C52	0x34	六维力矩传感器零点设置错误 请检查传感器通信接线以及通电是否正常。
C53	0x35	六维力矩传感器过载 请减小负载或施加的外力。
C110	0x6E	底座板通信异常
C111	0x6F	控制器外挂 485 设备通信异常
上表中未出现的报警代码：重新使能机械臂和机械爪。如频繁出现，请联系技术支持。		

### 3.2.2. 控制器警告代码

警告是不影响机器人正常工作，但有可能会影响用户的程序运行，一旦发生警告，机器人会设置警告标志位，并在指令回复中一起返回，除此之外不会进行其他操作，机械臂依然会正常运行。

警告代码	描述	报警处理
11 (0x0B)	缓存溢出	控制指令缓存
12 (0x0C)	指令参数异常	检查发送指令
13 (0x0D)	指令不存在	检查发送指令
14 (0x0E)	指令无解	检查发送指令

### 3.3. 机械爪报警信息及常规处理方式

报警处理方式可采用重新上电，步骤如下（重新上电需要走完以下所有步骤）：

1. 通过控制器上的紧急停止按钮重新对机械臂上电
2. 使能机械臂

xArm Studio 使能方式：点击报错弹窗的引导按钮或者首页的使能按钮。

xArm-Python-SDK 使能方式：参照报警处理方式。

xarm\_ros 库：查看相关文档 [https://github.com/xArm-Developer/xarm\\_ros](https://github.com/xArm-Developer/xarm_ros)

3. 新使能机械爪：若多次上电无效后请联系 UFACTORY 团队。

软件报错代码	报警代码	报警处理
G9	0x09	机械爪电流检测异常 请通过控制器上的紧急停止按钮重启机械臂
G11	0x0B	机械爪电流过大 请点击“确认”重新使能机械爪
G12	0x0C	机械爪速度过大 请点击“确认”重新使能机械爪
G14	0x0E	机械爪位置指令过大 请点击“确认”重新使能机械爪
G15	0x0F	机械爪 EEPROM 读写错误 请点击“确认”重新使能机械爪
G20	0x14	机械爪驱动 IC 硬件异常 请点击“确认”重新使能机械爪
G21	0x15	机械爪驱动 IC 初始化异常 请点击“确认”重新使能机械爪
G23	0x17	机械爪电机位置偏差过大 请检查机械爪运动是否受阻，如机械爪运动未受阻，请点击“确认”重新使能机械爪
G25	0x19	机械爪指令超软件限位 请检测机械爪指令是否设置超出软件限制
G26	0x1A	机械爪反馈位置超限软件限位
G33	0x21	机械爪驱动器过载
G34	0x22	机械爪电机过载
G36	0x24	机械爪驱动器类型错误 请点击“确认”重新使能机械爪
上表中未出现的报警代码：重新使能机械臂和机械爪。如频繁出现，请联系技术支持。		

xArm-Python-SDK 报警处理方式：

在用 Python 库设计机器人运动规划时，如果机器人出现故障，需要手动清除错误。清除错误后，仍需重新给机器人使能，设置运动模式，方可使机器人正常运动。此时根据上报的错误信息，应重新调整机器人的路径规划。

Python 库清除错误步骤：（如下接口，详细说明请查看 GitHub）

1. 清除错误：clean\_error()
2. 重新使能机械臂：motion\_enable(true)
3. 设置运动状态：set\_state(0)

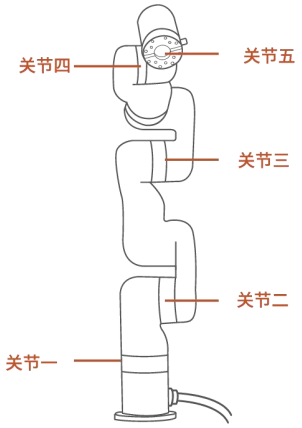
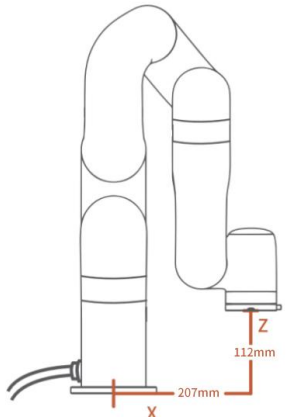
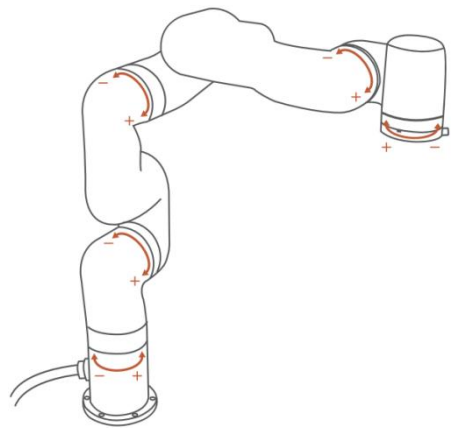
## 4. 技术规格

### 4.1. xArm5/6/7 通用规格

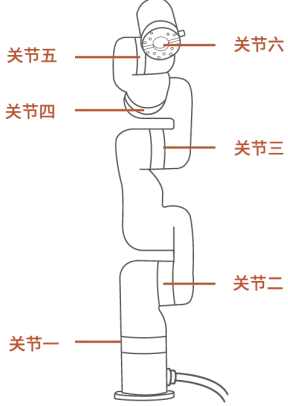
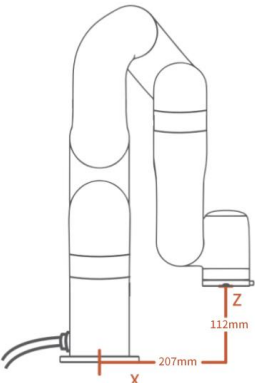
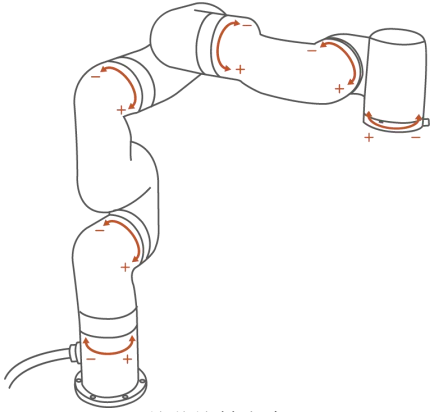
xArm		
笛卡尔范围	X	±700mm
	Y	±700mm
	Z	-400mm~951.5mm
	Roll/Yaw/Pitch	± 180°
最大关节速度	180° /s	
有效工作半径	700mm	
重复定位精度	±0.1mm	
最大末端速度	1m/s	
*环境温度	0-50 ° C*	
功耗	最低 8.4 W, 典型 200 W, 最高 500W	
输入电源	24 V DC, 20.8A	
ISO 洁净室等级	5	
机械臂安装	任意角度	
编程方式	xArm Studio 图形界面 Python/C++/ROS 底层接口	
机械臂通信协议	Modbus TCP	
末端工具 IO 接口	2 个数字输入, 2 个数字输出, 2 个模拟输入	
末端工具 485 通信协议	Modbus RTU	
占地面积(mm)	Ø 126 mm	
材料	铝、碳纤维	
末端工具法兰	DIN ISO 9409-1-A50/63 (M5*6)	
控制器		
	交流控制器	直流控制器
输入	100-240V AC 50/60Hz	24-72V DC
输出	24V DC 20.8A	24V DC 672Wmax
控制器通信协议	Modbus TCP	
控制器通信方式	Ethernet (以太网)	
控制器 IO 接口	8*CI+8*DI (数字输入) 8*CO+8*DO (数字输出) 2*AI (模拟输入) 2*AO (模拟输出) 1*RS-485 主	8*CI+8*DI (数字输入) 8*CO+8*DO (数字输出) 2*AI (模拟输入) 2*AO (模拟输出) 1*RS-485 主)
质量	3.9kg	2.6kg
尺寸(L*W*H)	285*135*101mm	262*160*76mm
机械爪		

额定电源电压	24V DC
绝对最大电源电压	28V DC
静态功耗（最低功耗）	1.5W
峰值电流	1.5A
工作范围	86mm
最大夹持力度	30N
质量	822g
通信方式	RS-485
通信协议	Modbus RTU
可编程参数	位置、速度
反馈	位置
真空吸头	
额定电源电压	24V DC
最大电源电压	28V DC
静态电流	30mA
峰值电流	400mA
真空度	78%
真空流速(L/min)	>5.6L/min
质量 (g)	610g
尺寸 (L*W*H)	122.5*91.6*75mm
有效载荷 (kg)	≤5kg
噪音水平(30cm 内)	<60dB
通信方式	数字 IO
状态指示灯	电源，工作状态
反馈信号	气压（低或常规）
备注： *1. 机械臂的工作温度是 0-50℃，关节连续高速操作时，降低环境温度。	

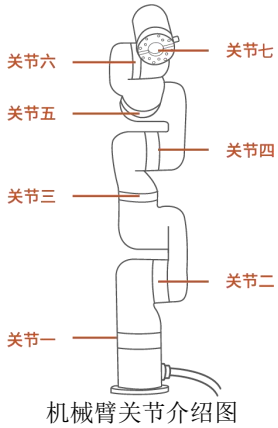
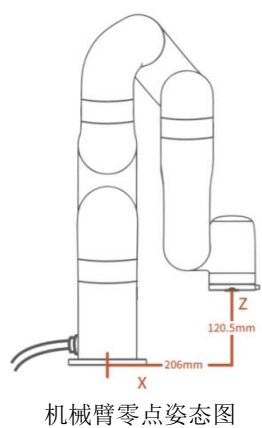
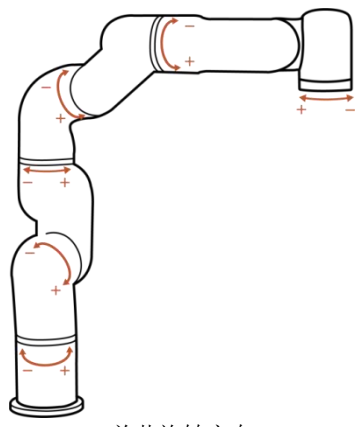
## 4.2. xArm 5 规格

关节范围	1, 5	$\pm 360^\circ$
	2	$-118^\circ \sim 120^\circ$
	3	$-225^\circ \sim 11^\circ$
	4	$-97^\circ \sim 180^\circ$
有效负载		3kg
自由度		5
机械臂重量 (不含控制器)		11.2 kg
 <p>机械臂关节介绍图</p>		 <p>机械臂零点姿态图</p>
 <p>关节旋转方向</p>		

### 4.3. xArm 6 规格

关节范围	1, 4, 6	$\pm 360^\circ$
	2	$-118^\circ \sim 120^\circ$
	3	$-225^\circ \sim 11^\circ$
	5	$-97^\circ \sim 180^\circ$
有效负载		5 kg
自由度		6
机械臂重量 (不含控制器)		12.2 kg
 <p>机械臂关节介绍图</p>		 <p>机械臂零点姿态</p>
 <p>关节旋转方向</p>		

## 4.4. xArm 7 规格

关节范围	1, 3, 5, 7	$\pm 360^\circ$
	2	$-118^\circ \sim 120^\circ$
	4	$-11^\circ \sim 225^\circ$
	6	$-97^\circ \sim 180^\circ$
有效负载		3.5kg
自由度		7
机械臂重量 (不含控制器)		13.7kg
 <p>机械臂关节介绍图</p>		 <p>机械臂零点姿态图</p>
 <p>关节旋转方向</p>		