

酷卓协作机器人应用手册——Modbus

1 引言

1.1 读者对象

本手册阅读对象为酷卓协作机器人 Modbus 用户。

1.2 手册内容

本手册描述了 Modbus 的基础知识、酷卓机器人作为主站和从站的应用以及工程样例。

1.3 适用系统版本

本手册版本为 1.0，适用机器人系统版本为 1.3.2 及以上。

2 Modbus 概述

2.1 MODBUS 背景

Modbus 是一种工业协议，于 1979 年由 Modicon 公司(后被施耐德(Schneider)收购)开发，旨在实现自动化设备之间的通信，这也是 Mod-Bus 名称的由来。自 1979 年以来，Modbus 一直是业界的系列事实上的标准，它继续为数以百万计的自动化设备提供通信服务。

2.2 协议类型

Modbus 是一个应用层消息传递协议，位于 OSI 模型的第 7 级，它在不同类型的总线或网络上连接的设备之间提供客户端/服务器通信。

MODBUS 是一种请求/应答协议，并提供由功能代码指定的服务。

OSI 层	OSI 模型	Modbus
7	应用层	Modbus 协议
6	表示层	-
5	会话层	-
4	传输层	-
3	网络层	-

2	数据链路层	串行链路层、TCP
1	物理层	232、485、光纤、网线等

2.3 协议传输模式

Modbus 可分为三个传输模式：基于串口的和基于以太网 TCP/IP 的。

- **基于串口的 Modbus-RTU**

数据按照标准串口协议进行编码，是使用最广泛的一种 Modbus 协议，采用 CRC-16_Modbus 校验算法。此模式比较常见。

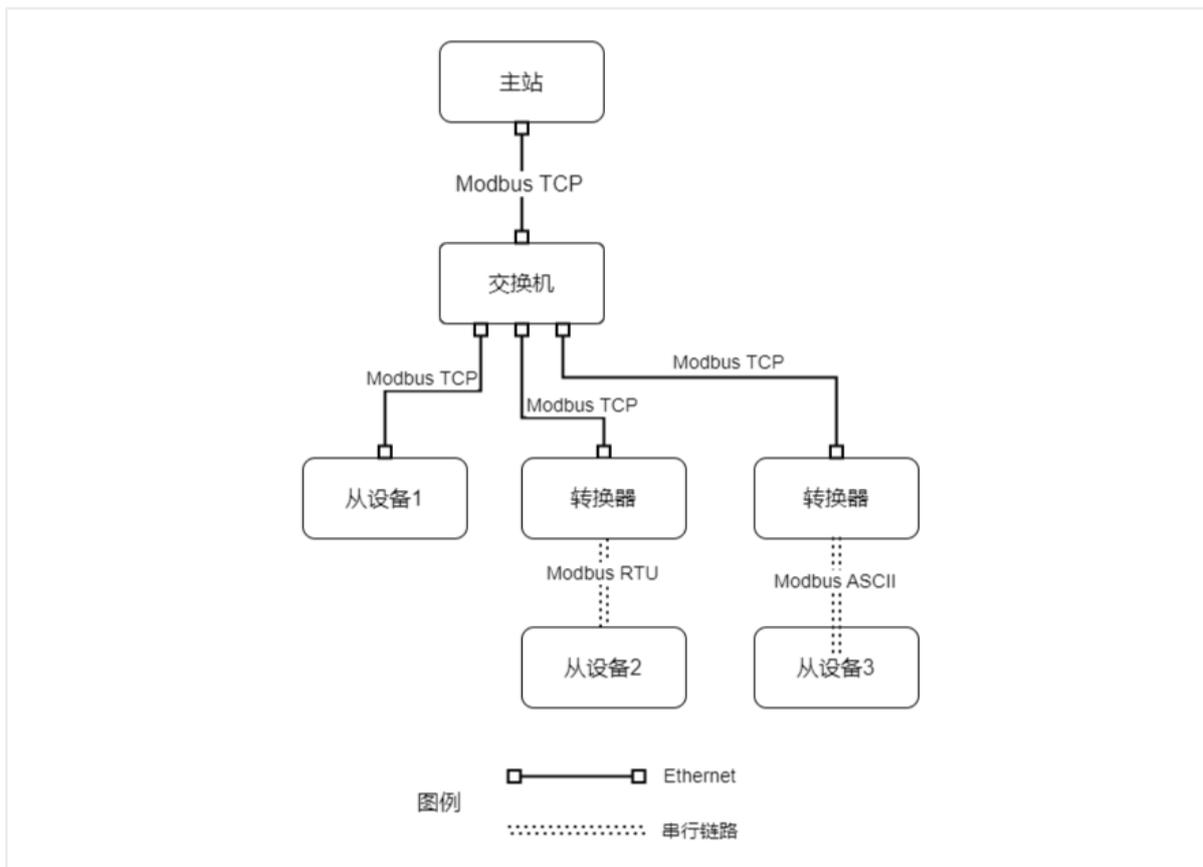
- **基于串口的 Modbus-ASCII**

所有数据都是 ASCII 格式，一个字节的原始数据需要两个字符来表示，效率低，采用 LRC 校验算法。此模式不常见，故省略详解。

- **基于网口的 Modbus-TCP**

Modbus-TCP 基于 TCP/IP 协议，占用 502 端口，数据帧主要包括两部分：MBAP(报头)+PDU(帧结构)，数据块与串口模式是一致的。此模式比较常见。

当总线上的传输模式不一致时，可以使用转换器连接。



2.4 Modbus-RTU 协议报文

Modbus-RTU 协议运行于串行链路，带有校验位和地址。Modbus-RTU 要求两个 RTU 报文帧间隔要大于 3.5 个字节时间，且每个报文帧内字节间隔小于 1.5 个字节时间，否则会认为接收不完整。

Data 数据的长度不是固定的，所以报文长度也并非固定。

		名称	长度	说明
ADU (Application Data Unit)	ADDRESS (Modbus Address)	SlaveID	1 字节	从站地址
	PDU	Function code	1 字节	功能码

	(Protocol Data Unit)	Data	N 字节	数据
	CRC	CRC 校验	2 字节	校验

2.5 Modbus-TCP 协议报文

Modbus-TCP 使 Modbus-RTU 协议运行于以太网，Modbus-TCP 使用 TCP/IP 以太网在站点间传送 Modbus 报文，Modbus-TCP 结合了以太网物理网络和网络标准 TCP/IP 以及以 Modbus 作为应用协议标准的数据表示方法。与传统的串口方式不同，Modbus-TCP 插入一个标准的 Modbus 报文到 TCP 报文中，不再带有数据校验和地址。

Data 数据的长度不是固定的，所以报文长度也并非固定。

		名称	长度	说明
ADU (Application Data Unit)	MBAP HEADER (MODBUS Application Protocol)	Transaction ID 事务处理标识	2 字节	报文的序列号，一般每次通讯加 1，用于区别不同的报文
		Protocol ID 协议标识	2 字节	00 00 代表 modbus-Tcp
		Length	2 字节	UnitID 和 PDU 的长度总和
		UnitID	1 字节	单元标识符(从站地址)
	PDU (Protocol Data Unit)	Function code	1 字节	功能码
		Data	N 字节	数据

2.6 数据类型

不论哪种模式，通常来说 Modbus 访问的数据存储四个数据库或地址范围的其中一个：线圈状态、离散量输入、保持寄存器和输入寄存器。与许多规范一样，名称可能因行业或应用而异。

内存区块名	别名	数据类型	数据长度	主站设备访问	从站设备访问
线圈状态	离散量输出	布尔	1 位	读/写	读/写
离散输入	离散输入寄存器	布尔	1 位	只读	读/写
保持寄存器	输出寄存器	无符号双字节整型	16 位	读/写	读/写
输入寄存器		无符号双字节整型	16 位	只读	读/写

2.7 Function code 功能码

Modbus 功能码，是写在主机请求数据帧中的，决定主机进行读还是写操作，是读线圈、离散量还是寄存器，是写单个寄存器还是多个寄存器等等，决定主机请求什么类型的数据。实际最常用的是公共功能码中的 4 个功能码：03(读多个保持寄存器)、04(读输入寄存器)、06(写单个保持寄存器)、16(写多个保持寄存器)。

酷卓协作机器人主站支持功能码 01(读单个线圈寄存器)，02(读离散输入寄存器)，03(读单个保持寄存器)，04(读输入寄存器)，05(写单个线圈寄存器)，06(写单个保持寄存器)。

类型

读单个线圈寄存器	功能码
读单个线圈寄存器	01
读离散输入寄存器	02
读单个保持寄存器	03
读输入寄存器	04
写单个线圈寄存器	05
写单个保持寄存器	06

				Function Codes		(hex)	Section
				code	Sub code		
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access	Read File record		20		14	6.14
		Write File record		21		15	6.15

3 酷卓协作机器人作为 Modbus 主站

本章节描述酷卓协作机器人作为 Modbus 主站时的使用。

3.1 项目需求

酷卓协作机器人作为 Modbus 主站，使用 modbus 拓展机器人模拟量 io 数量。

3.2 需求分析

模拟量 io 为保持寄存器，即读单个保持寄存器与写单个保持寄存器。

3.3 机器人配置

Modbus 的连接设置在设置选项卡-通讯中，开机自动加载该设置。

设置如下图所示，IP 地址为从站设备 IP 地址。为简化演示，此处仅新建 2 个寄存器，分别是读单个保持寄存器与写单个保持寄存器。为方便在编程时查找该变量，可以为其重命名名称。

信号的频率可选择 0 或其他频率，当频率设置为 0 时，将在程序种使用指令按需启动 MODBUS 请求，当频率非 0 时，Modbus 将在后台以设定频率自动实时刷新数值，此时不需要使用指令读取或设置对应变量。为演示，此处设置读取寄存器频率为 0，写入寄存器频率为 10。

The screenshot displays the 'MODBUS 客户端设置' (MODBUS Client Settings) interface. At the top, there's a 'MODBUS' tab. Below it, a 'MODBUS 客户端设置' header is followed by a '添加 MODBUS 设备' (Add MODBUS Device) button. The main configuration area includes fields for '设备名称' (Device Name: MODBUS_lyBqybn), 'IP 地址' (IP Address: 192.168.71.1), and '端口' (Port: 502). There are also buttons for '添加新信号' (Add New Signal) and '删除设备' (Delete Device). Below this, a summary shows '重新连接计数: 0', 'ModBus 数据包错误: 0', and '连接状态: 已连接' (Connected). The main table lists two registers:

类型	地址	名称	值
读单个保持寄存器	0	read_register	0
写单个保持寄存器	1	MODBUS_lyBqy0b	33

Each register entry includes a '频率[Hz]' (Frequency) dropdown, a 'MODBUS 从设备地址' (MODBUS Slave Address) field, and a '删除' (Delete) button. The first register has a frequency of 0 and a slave address of 1. The second register has a frequency of 10 and a slave address of 1. Below the table, there are fields for '响应时间[ms]' (Response Time), '超时' (Timeout), '请求失败' (Request Failure), and '实际频率' (Actual Frequency). The first register has a response time of --, timeout of --, request failure of --, and actual frequency of -- Hz. The second register has a response time of 35, timeout of 0, request failure of 0, and actual frequency of 9 Hz. At the bottom, there is a '保存' (Save) button.

3.4 工程样例

The image displays a PLC ladder logic editor interface. On the left, a vertical timeline shows 10 steps:

- 1. Start
- 2. MovJ P1 (highlighted)
- 3. ReadSingleHoldReg
- 4. If read.value > 10000
- 5. MovL P2
- 6. ...=... write_register.value=4564
- 7. Else
- 8. MovL P3
- 9. GoTo Start 1
- 10. End

On the right, the configuration panel for the selected '2. MovJ' step is shown:

2. MovJ

复制 注释 删除

目标位置 : P1

+APOS +CPOS [Location Icon] [Close Icon]

目标速度 : V100

加速度 : ACC100

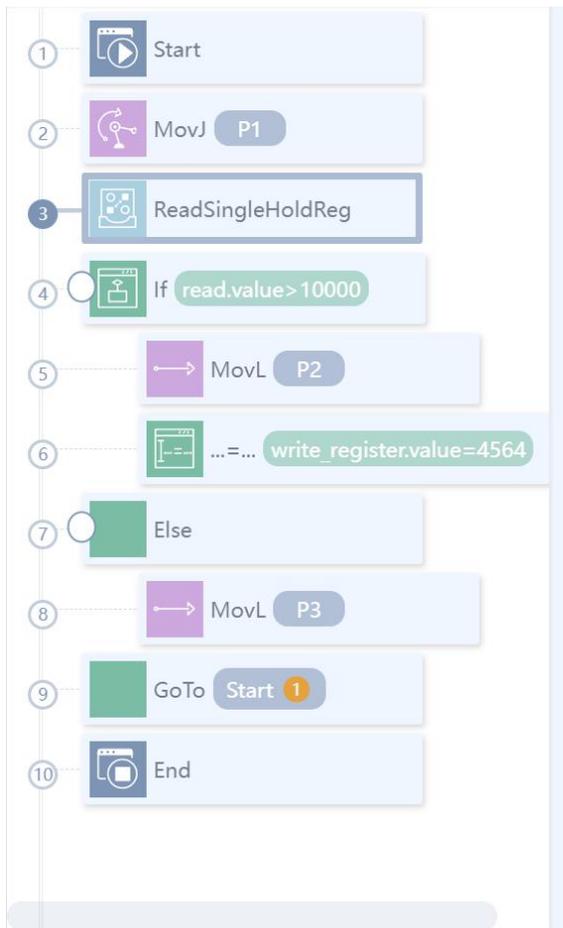
过渡类型 : FINE

过渡值 : ZONE0

工具参数 : DEFAULT

坐标系 : DEFAULT

工件负载 : DEFAULT



3. ReadSingleHoldReg

复制
 注释
 删除

设备名称 : MODBUS_jobox

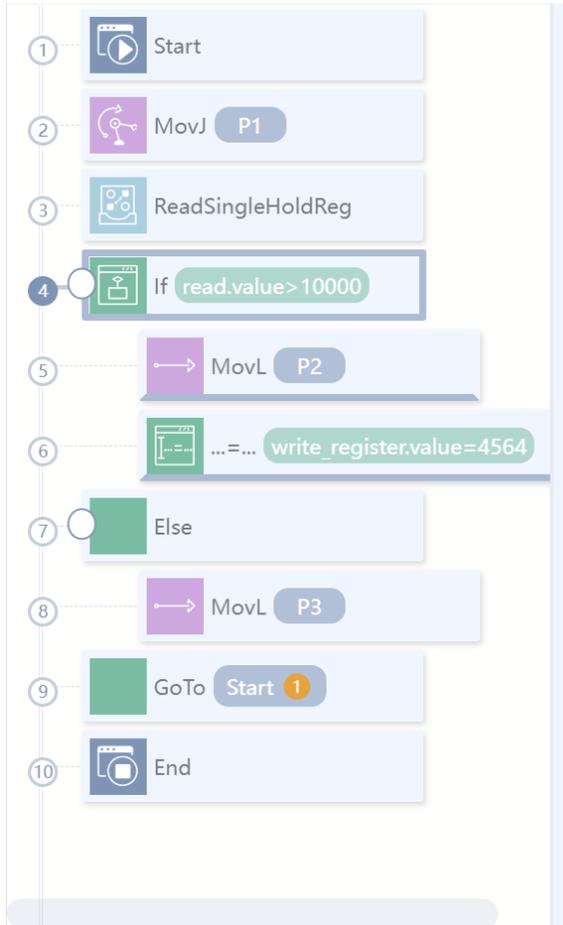
地址 : 0

目标寄存器值 : read_register

从设备地址 : 1

超时时间 : 3000 ms

操作返回值 : sucess



4. If

复制
 注释
 删除

条件表达式

ⓘ 点击节点, 编辑下面表单提交。

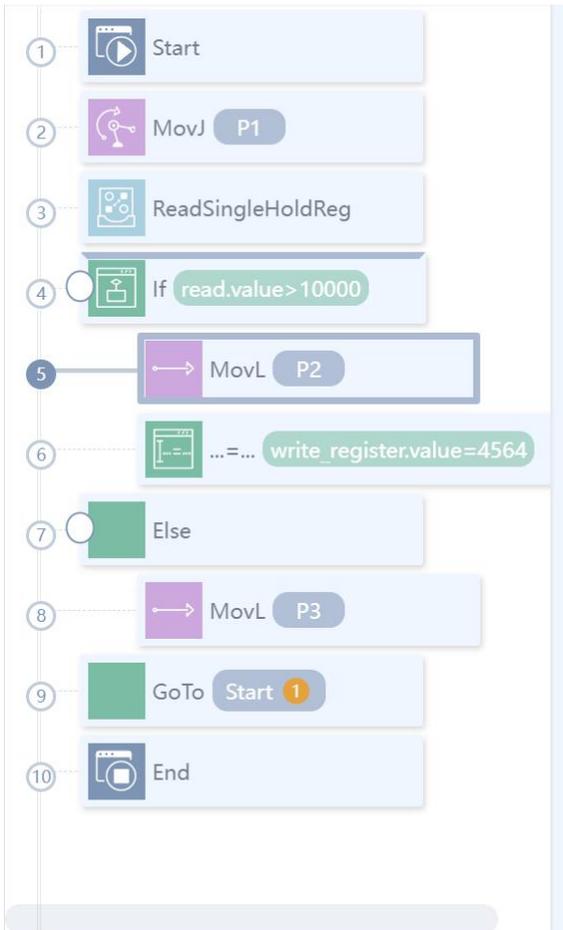
$read.value > 10000$

编辑表达式

类型 : 值

数据类型 : 变量

选择变量 : read / value



5. MovL

复制
 注释
 删除

目标位置： P2

+APOS +CPOS

目标速度： V100

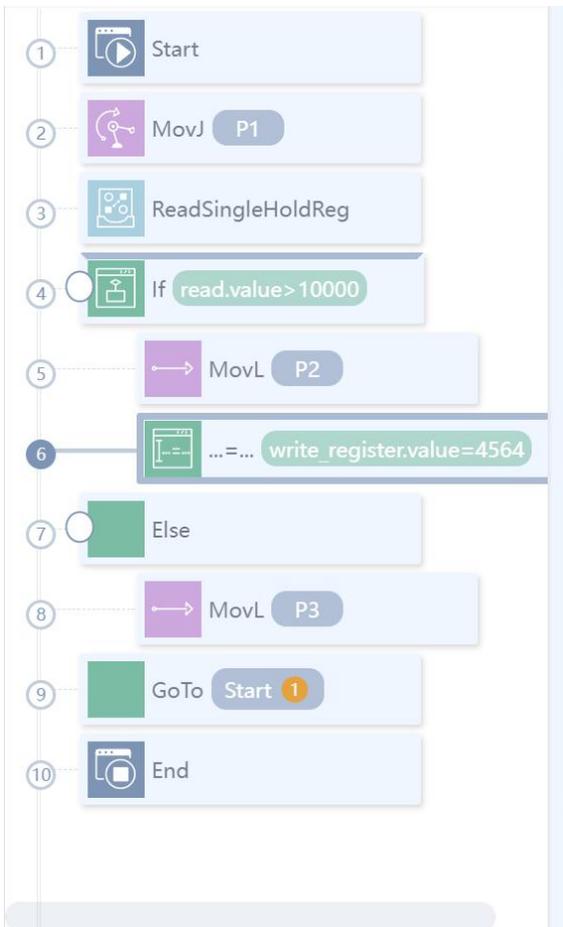
加速度： ACC100

过渡类型： FINE

工具参数： DEFAULT

坐标系： DEFAULT

工件负载： DEFAULT



6. ...=...

复制
 注释
 删除

条件表达式

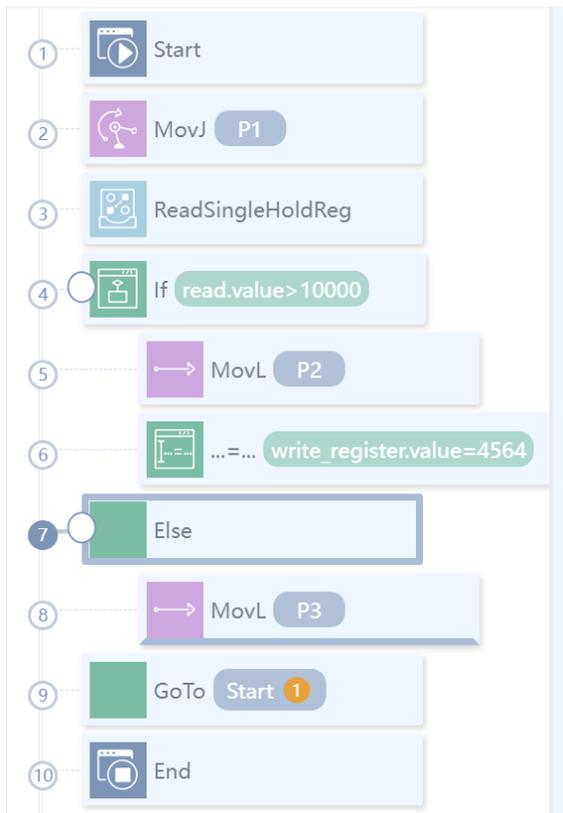
i 点击节点，编辑下面表单提交。 ×

write_register.value = 4564 +

编辑表达式

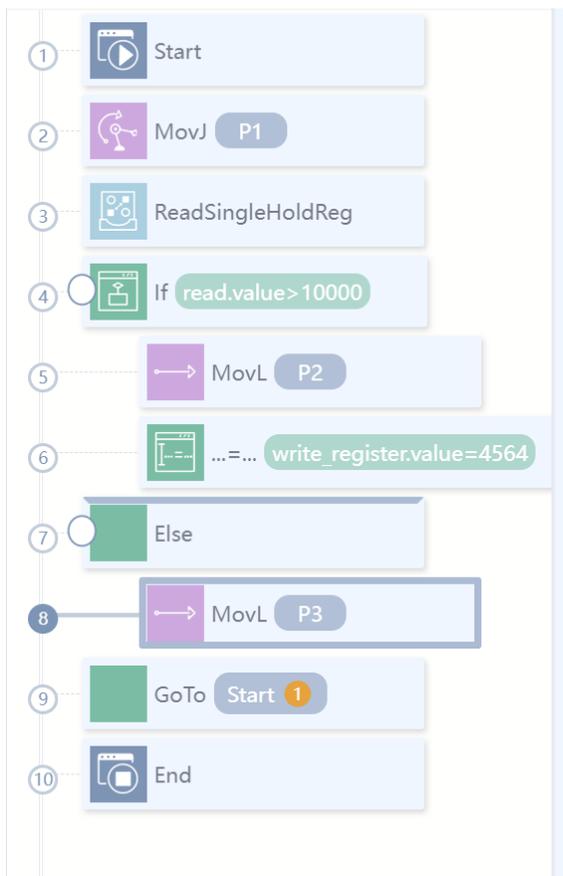
数据类型： 变量

选择变量： write_register / value



7. Else

复制 注释 删除



8. MovL

复制 注释 删除

目标位置 : P3

+APOS +CPOS

目标速度 : V100

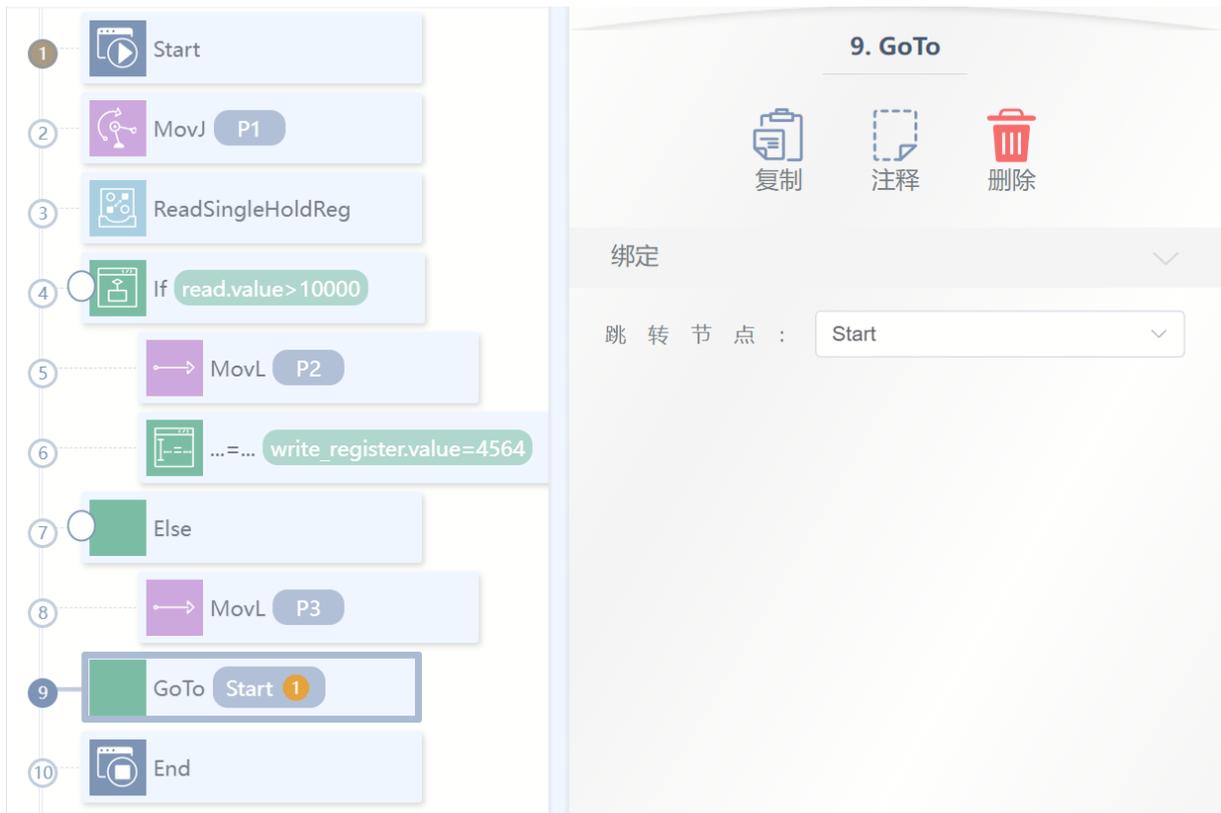
加速度 : ACC100

过渡类型 : FINE

工具参数 : DEFAULT

坐标系 : DEFAULT

工件负载 : DEFAULT



该程序是工程下主程序调用的一共子程序，其中每一行的含义如下：

行号	指令	说明
2	MovJ P1	关节移动到 P1 点。
3	ReadSingleHoldReg	使用指令从寄存器读取数值，因为在设置中频率为 0，必须使用指令手动刷新。
4	If read.value>10000	判断如果该模拟量数值大于 10000.
5	MovL P2	直线移动到目标点位 P2。
6	...=...	将数值 4564 写入到 write_register 寄存器，后面无需调用 Modbus 写的指令，因为设置项中被设置为 10Hz 自动刷新。
7	Else	不满足 If 的条件则运行下列指令。
8	MovL P3	直线移动到 P3。
9	GoTo	跳转到程序开始点，循环这个程序

再次强调，信号的频率可选择 0 或其他频率，当频率设置为 0 时，将在程序种使用指令按需启动 MODBUS 请求，当频率非 0 时，Modbus 将在后台以设定频率自动实时刷新数值，此时不需要使用指令读取或设置对应变量。

4 酷卓协作机器人作为 Modbus 从站

酷卓协作机器人作为 Modbus 从站时，既有出厂时预先定义好的寄存器，同时还具有用户可自定义的寄存器。

4.1 从站协议

酷卓协作机器人作为 Modbus 从站时，其保持寄存器具有已经预先定义好的信息，如机器人基本信息、关节信息、末端信息、IO 信息、基本控制，此外还包含整数类型变量、实数类型变量以及布尔类型变量。具体的定义请查看《酷卓协作机器人 Modbus 从站协议》

4.2 机器人配置

对于整数类型变量、实数类型变量以及布尔类型变量，在设置选项卡中可以对各个变量重命名别名，这对于程序调试是很大的便捷。

5 参考资料

酷卓协作机器人用户手册

酷卓协作机器人 Modbus 从站协议

Modbus 国标手册