ER 系列工业机器人 ModbusTCP 调试手册

(**RCS2** V1.5)

修订记录

序号	日期	作者	版本	描述
1	2018.08.17	吴俣	V1.0	初次发布
2	2018.01.16	王正谦	V1.1	1、 修改了个别语言描述 2、 修改了模拟量 IO 点，即支持浮点数据
3	2018.03.05	赵朋和	V1.1.2	1、 增加了心跳检测位(对外)
4	2019.06.17	时伟青	V1.4	1、 修改了个别语言描述 2、 增加了通信操作示例
5	2020.01.13	时伟青	V1.5	1、 修改不正确的描述

目 录

前言

概述

本手册适用于控制系统 RCS2 V1.5，描述埃斯顿二代控制器 ModBusTcp 协议接口功能介绍。

读者对象

本手册仅供受过培训，熟悉各种适用国家标准的“控制、自动化和驱动工程”领域专业人员。

- 系统生产商：对自动化系统功能设计的技术人员。
- 系统集成商：指自动化设备集成的技术人员。

注意事项

- 在安装和调试这些组件时，操作人员必须严格遵循本文档的说明和解释。
- 相关负责人员必须确保所述产品的应用或使用满足所有安全要求，包括相关法律、法规、准则和标准。
- 尽管本文档经过精心编制，但由于其中所描述的产品仍处于不断更新换代中，我们可能不会在每次更新后都检查文档中所描述的产品性能数据、标准或其它特性总是与实际产品相一致。
- 本文档中难免会出现一些技术或者编辑错误，我们保留随时对文档信息做出修改之权力，恕不另行通知。对于已经变更的产品，如果本文档中的数据、图表以及文字描述没有修改，我们将不再特别加以声明。
- 任何人不得对软、硬件配置进行文本档中规定之外的修改，ESTUN 公司对因此而造成的一切后果不承担任何责任。
- 本文档中出现图示单位在没有特别标注说明时，默认单位为毫米 **mm**。

安全说明

警告	受伤的危险 不遵守本标志相关的安全说明将危及个人生命和健康安全。
注意	对环境和设备有危险 不遵守本标志相关安全说明可能明显危害环境和设备安全。
说明	说明或提示 该标志表示这些信息能够帮助您更好的理解安全说明。

第 1 章 功能简介

ModBusTcp 协议接口是指外部逻辑控制器（PLC 等）通过标准工业总线协议（ModBusTCP）与机器人通讯，读写机器人的虚拟 IO 端口的一种通讯方式。机器人可通过 ModBusTCP 的方式与外部设备进行交互，而无需额外扩展 IO 端口。

第 2 章 ModBusTcp 协议介绍

ModBus 是目前应用最广泛的现场总线之一，1999 年又推出了以太网运行的 ModBusTCP(工业以太网协议)。ModBusTCP 以一种比较简单的方式将 ModBus 帧嵌入 TCP 帧中。IANA 给 ModBus 协议赋予 TCP 端口号 502，这是其他工业以太网协议所没有的。ModBusTCP 还拥有其它工控设备的支持，如工业人机界面、变频器、软启动器、电动机控制中心、以太网 I/O、各种现场总线的网桥等。

2.1 ModBusTCP 协议报文

ModBusTCP 采用 TCP/IP 和以太网协议来传输 ModBus 信息，因此与 ModBus 串行链路数据单元类似，ModBusTCP 的应用数据单元就是将 ModBus 简单协议数据单元（PDU）按照 TCP/IP 协议进行封装。一个 TCP 帧只能传送一个 ModBusADU，建议不要在同一个 PDU 中发送多个 ModBus 请求或相应。

以下表格为 ModBusTCP 的请求、应答报文：

请求：	传输标志	协议标志	长度	从站号	功能码	寄存器起始地址	寄存器长度
长度 (Byte)	2	2	2	1	1	2	2
应答：	传输标志	协议标志	长度	从站号	功能码	长度	数据值…
长度 (Byte)	2	2	2	1	1	1	

2.2 ModBusTCP 协议功能码

ModBus 的数据结构将数据区分为以下四种：

保持线圈(Holding Coils)	可读可写的 BOOL 型变量
输入线圈(Input Coils)	只读的 BOOL 型变量
保持寄存器(Holding Registers)	可读可写的 WORD 型（16-bit）型变量
输入寄存器(Input Registers)	只读的 WORD 型（16-bit）型变量

埃斯顿 ModBusTCP 通讯接口目前只支持读写保持寄存器，即以下功能码：

功能码	描述
0x03	读保持寄存器
0x06	写单个保持寄存器
0x10	写多个保持寄存器

2.3 ER 系列机器人 ModBusTCP 接口定义

ER 系列机器人的 ModBusTCP 接口配置如下：

IP：192.168.6.68（实时系统 IP 地址）

Port：502

采集周期>200ms

响应超时>150ms

说明

服务器通过检测客户端的请求来判断连接是否存在。若与客户端连接后，超过 5 秒未接收到客户端的请求，服务器将自动断开连接并重新监听 502 端口。

ER 系列机器人在 ModBusTCP 保持寄存器中封装了一层数据结构

	地址	定义
Rob Send	40002	反馈当前速度值
	40003	读写标志位应答
	40004	Rob 状态信息
	40005~40014	当前加载工程名
	40015-40018	用户用（虚拟数字输出）

	40019	Rob 执行命令状态
	40020~40051	用户用（虚拟模拟输出）
Rob Receive	40052	Rob 操作指令
	40053	全局速度设置值
	40054~40063	设置加载工程名
	40064-40067	用户用（虚拟数字输入）
	40068~40099	用户用（虚拟输入）
	40100	读写标志位请求

第 3 章 ModBusTCP 点表

Rob 作为 ModbusTCP 通信的 Server,定义了大小为 202 个字节的数据寄存器区域（即 MBDataBuffer : ARRAY [0..99] OF WORD）；

其中，MBDataBuffer[1]~MBDataBuffer[50]为 Server 的发送区域（即外部 PLC 需要从 Rob 获取相关状态信息，只读）；

其中，MBDataBuffer[51]~MBDataBuffer[99]为 Server 的接收区域（即外部 PLC 需要对 Rob 写入相关操作指令和信息）；

备注：针对以下点表中的数据操作，可结合 Multiprog 软件使用，详情可联系埃斯顿技术服务人员。

类型	本地地址	寄存器地址	定义	说明	注释
–	MBDataBuffer[0]	40001	预留	– 心跳检测位	该值由 1 到 65535 循环
–	MBDataBuffer[1]	40002	当前速度值	–	–
–	MBDataBuffer[2]	40003	写权限应答	返回服务器收到的 400100 寄存器的值	
Send	MBDataBuffer[3]	40004	Rob 状态信息	**bit0**：手动操作模式 **bit1**：自动操作模式 **bit2**：远程操作模式 **bit3**：使能状态 **bit4**：运行状态 **bit5**：错误状态 **bit6**：程序运行状态 **bit7**：机器人正在动作	–
	MBDataBuffer[4]	40005	当前加载工程名	20 个字节	例如加载的工程文件名为:estun.test,则各寄存器的值如下:
	MBDataBuffer[5]	40006			

类型	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[6]	40007			[4]0x6573, [5]0x7475, [6]0x6E2E, [7]0x6D61, [8]0x696E
	MBDataBuffer[7]	40008			
	MBDataBuffer[8]	40009			
	MBDataBuffer[9]	40010			
	MBDataBuffer[10]	40011			
	MBDataBuffer[11]	40012			
	MBDataBuffer[12]	40013			
	MBDataBuffer[13]	40014			
	MBDataBuffer[14]	40015	SimDout[1-16]	DO 1-16	–
	MBDataBuffer[15]	40016	SimDout[17-32]	DO 17-32	–
	MBDataBuffer[16]	40017	SimDout[33-48]	DO 33-48	–
	MBDataBuffer[17]	40018	SimDout[49-64]	DO 48-64	–

类型	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[18]	40019	Rob 执行命令状态	**bit0**：命令为 0 **bit1**：急停命令执行成功（目前不可用） **bit2**：启动命令执行成功 **bit3**：停止命令执行成功 **bit4**：复位命令执行成功 **bit5**：上使能命令成功（目前不可用） **bit6**：下使能命令成功（目前不可用） **bit7**：加载工程命令成功 **bit8**：注销工程命令成功 **bit9**：设置全局速度成功 **bit10**：等待控制权 **bit11**：等待命令 **bit12**：等待命令执行完成 **bit13**：命令执行错误 **bit14**：复位程序指针命令成功 **bit15**：保留	命令寄存器为 0 时，bit[0]为 1，命令寄存器由命令时，bit[0]为 0 命令执行成功后，相应成功位置 1 当重新获取控制权且命令为 0 时，之前的成功位会被清零 因此，可下发命令的状态码为 0x801
	MBDataBuffer[19]	40020	SimAout[1-16]	AO 1-16	每两个寄存器对应一个模拟量点（浮点型）
	……	–			

类型	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[50]	40051			
Receive	MBDataBuffer[51]	40052	Rob 操作指令	**bit2**（0→0x4）：机器人程序启动 **bit3**（0→0x8）：机器人程序停止 **bit4**（0→0x10）：机器人错误复位 **bit7**（0→0x80）：加载工程文件 **bit8**（0→0x100）：注销当前工程文件 **bit9**（0→0x200）：设置全局速度 **bit10**（0→0x400）：指令状态机重置 **bit14**（0→0x4000）：复位程序指针	机器人必须处于远程励磁状态，所有命令均上升沿触发，配合读写标志位 40100 为 0x11（命令状态位 40019 为 0x801 时可下发指令） **特别：当指令执行失败后或状态机出错时，需利用 bit10 来重置指令执行器，才可以再下发新的指令**
	MBDataBuffer[52]	40053	全局速度值	–	设置速度值应大于 0 且小于等于 100
	MBDataBuffer[53]	40054	设置工程名	20 个字节	–
	MBDataBuffer[54]	40055			
	MBDataBuffer[55]	40056			
	MBDataBuffer[56]	40057			
	MBDataBuffer[57]	40058			

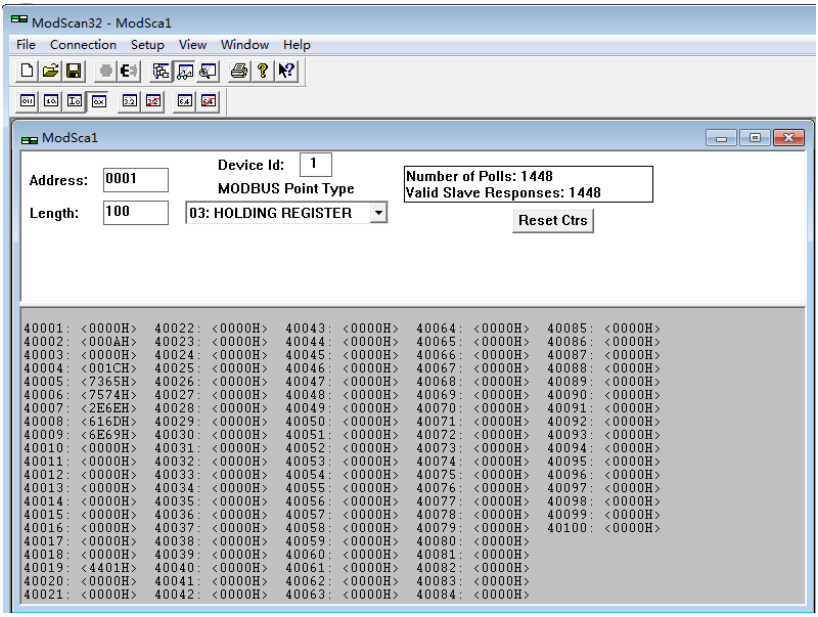
类型	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[58]	40059			
	MBDataBuffer[59]	40060			
	MBDataBuffer[60]	40061			
	MBDataBuffer[61]	40062			
	MBDataBuffer[62]	40063			
	MBDataBuffer[63]	40064	SimDI[1-16]	DI 1-16	–
	MBDataBuffer[64]	40065	SimDI[17-32]	DI 17-32	–
	MBDataBuffer[65]	40066	SimDI[33-48]	DI 33-48	（已预留可用，未在示教器上显示）
	MBDataBuffer[66]	40067	SimDI[49-64]	DI 48-64	（已预留可用，未在示教器上显示）
	MBDataBuffer[67]	40068	SimAI[1-16]	AI 1-16	每两个寄存器对应一个模拟量通道（浮点型）
	……				
	MBDataBuffer[98]	40099			
–	MBDataBuffer[99]	40100	读写标志位		0x11 打开 rob 命令下发权限

第 4 章 接口调试

该部分以 ModBus 调试助手 ModScan 为例，说明如何使用 ModBusTCP 通信接口与机器人交互，ModScan 模拟通信主站，机器人作为通信从站。

4.1 ModScan 调试助手

① 该调试软件界面如下：

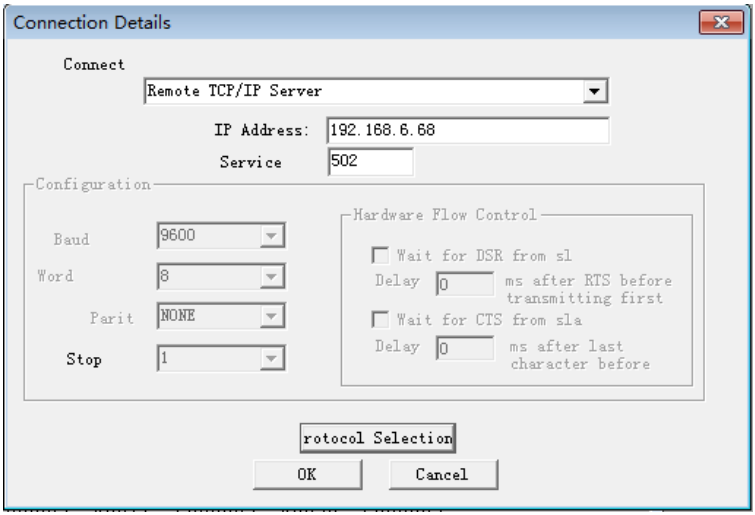


② 连接调试步骤

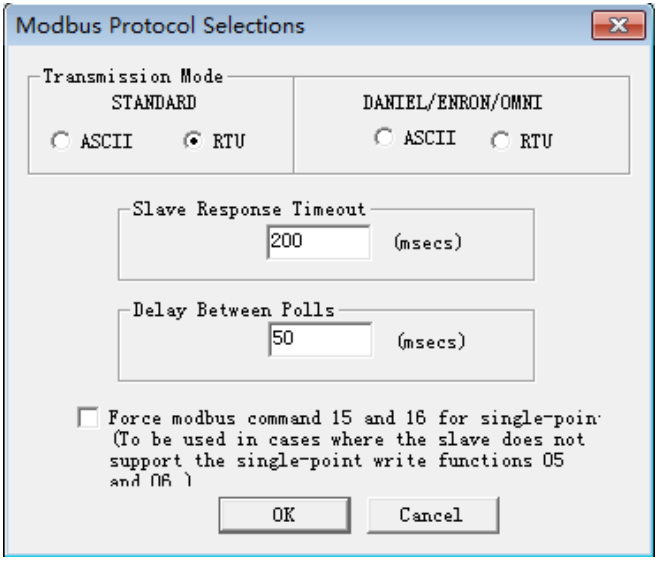
1、 将本地电脑连通机器人控制系统。(可 ping 通实时系统 IP192.168.6.68)

2、 配置 ModScan 连接参数

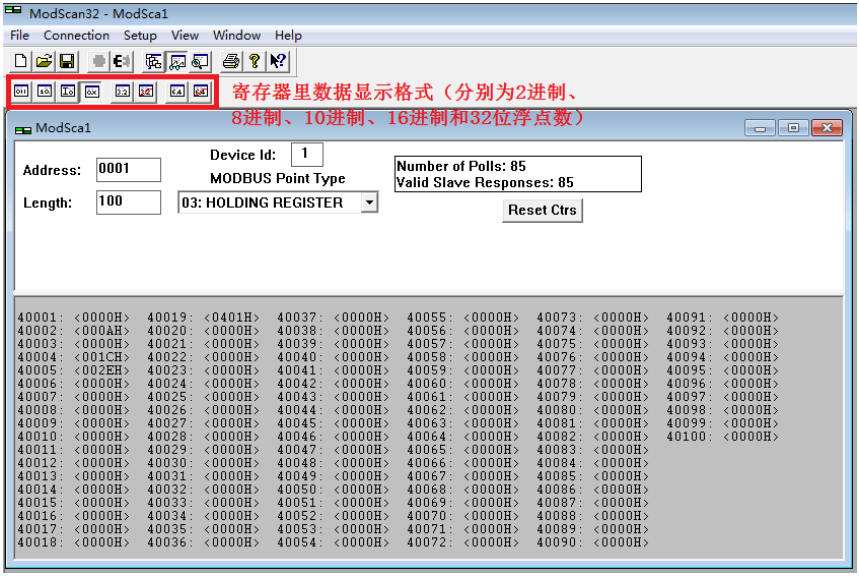
点击 Connection→Connect 进入如下界面，配置机器人 IP 和端口号。



先点击 protocol selection 配置连接参数，再点击 OK 完成配置并连接机器人



3、 Address（起始地址）设为 1，Length（长度）设为 100，功能码下拉选择 03: HOLDING REGISTER(保持寄存器)。

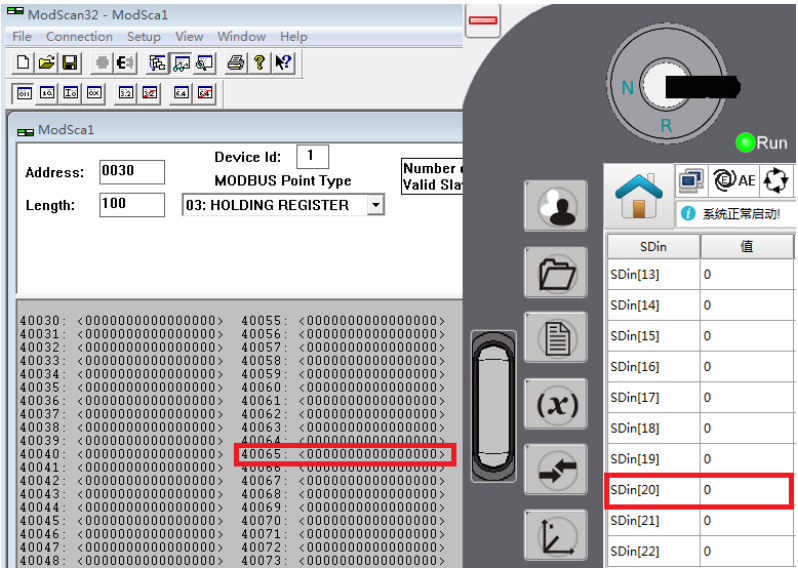


4.2 虚拟数字量交互示例

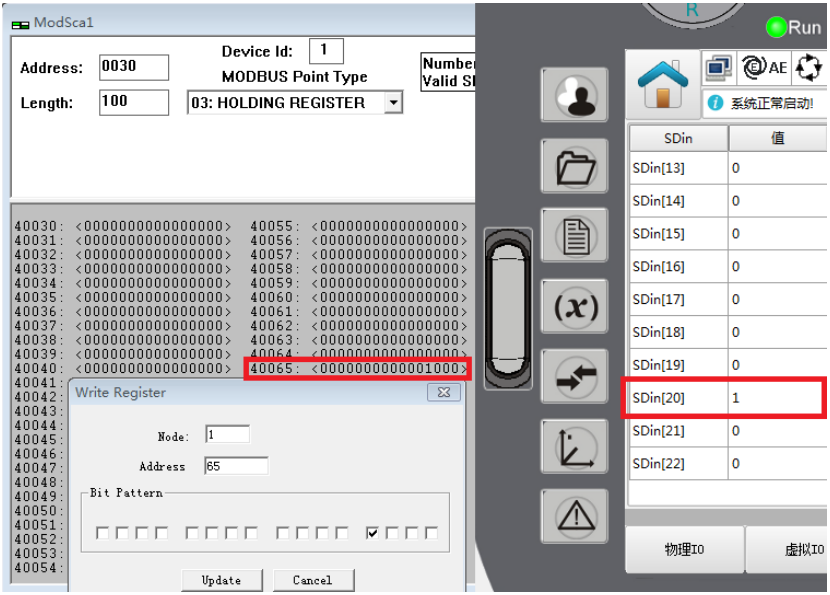
- ModScan（PLC）给机器人 SimDI20 端口发送高电平；

通过查看本文第 3 章 ModBus Tcp 点表可知，SimDI20 对应机器人寄存器 40065 的 bit4。

通信交互前：



通信交互后：双击 ModScan 对应的 40065 寄存器弹出写值窗口，勾选 bit4 点击 Update 后，示教器端的 SDin[20]变为 1，通信成功。



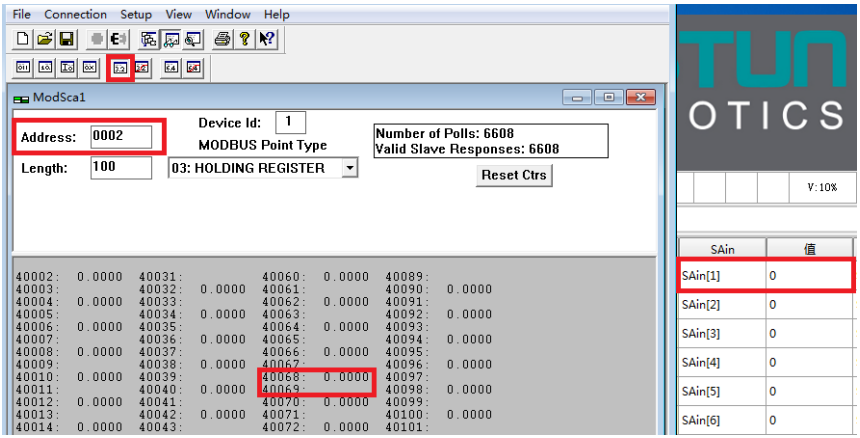
4.3 虚拟模拟量交互示例

- ModScan（PLC）给机器人 SimAI1 端口发送数值 25；

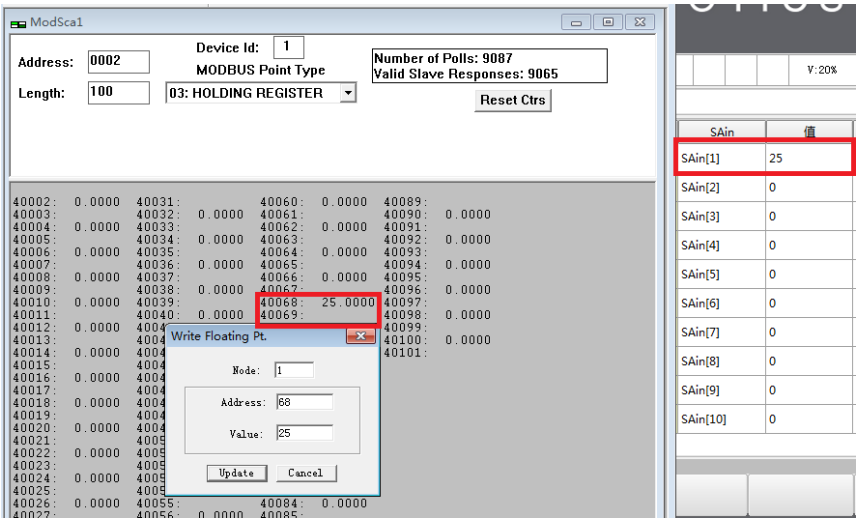
通过查看本文第 3 章 ModBus Tcp 点表可知，SimAI1 对应机器人寄存器 40068 和 40069，模拟通道传输的是浮点型数据，一个模拟通道对应两个寄存器地址。

ModScan 端需要按照下图所示的设置，显示格式改为 32 位浮点数，起始改为 2，这样才能使 40068 和 40069 两个寄存器显示一个浮点数（与外部设备实际应用中不需要更改任何设置）。

通信交互前：



通信交互后：双击 ModScan 对应的 40068 寄存器弹出写值窗口，Value 写入 25 点击 Update，示教器端 SAin[1]收到数值 25，通信成功。

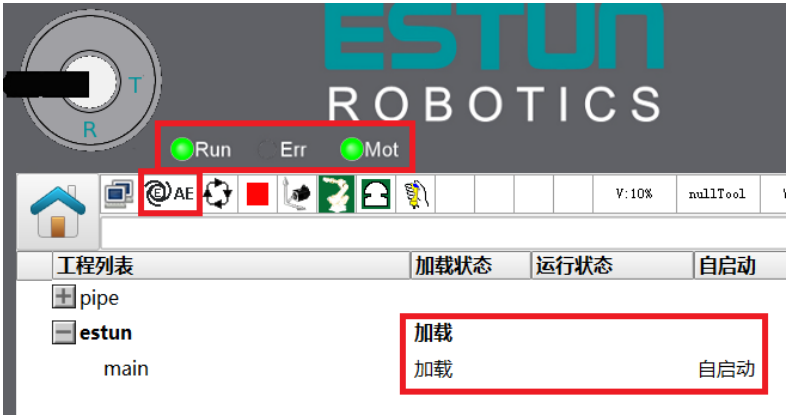


4.4 远程启动机器人程序

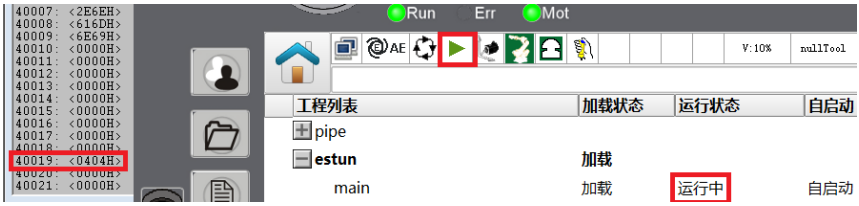
- ModScan（PLC）给机器人发送启动运行程序命令（停止复位操作与此类似）；

通过查看本文第 3 章 ModBus Tcp 点表可知，所有给机器人发送命令的操作，前提有二：①机器人处于远程励磁状态；②所以命令操作皆配合读写标志寄存器 40100 且上升沿触发有效，远程启动对应的命令寄存器是 40052，值为 0x4。

① 将机器人示教器钥匙开关拨至远程状态，机器人将会自动上励磁和加载已经设为自启动的程序，如下图。



② 确保 40100 和 40052 两个寄存器里值为 0，首先往 40100 寄存器里写入 0x11，此时 40019 寄存器状态变为 0x801（该值表示当前系统处于等待命令状态），再往 40052 寄存器里写入 0x4，此时机器人程序启动，40019 寄存器状态变为 0x404（该值表示启动命令执行成功并继续等待新的控制权），如果后续接着给机器人发送其它命令，应先把 40100 和 40052 两个寄存器内容清 0。

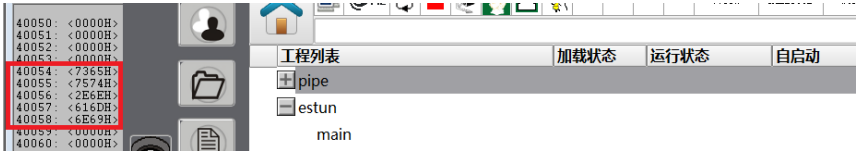


4.5 远程加载机器人程序

- ModScan（PLC）给机器人发送加载程序命令，程序名为 estun.main，然后示教器加载该程序；

通过查看本文第 3 章 ModBus Tcp 点表可知，所有给机器人发送命令的操作，前提有二：①机器人处于远程励磁状态；②所以命令操作皆配合读写标志寄存器 40100 且上升沿触发有效，远程加载对应的命令寄存器是 40052，值为 0x80，要加载的程序名存储在以 40054 寄存器开头的后面 10 个寄存器内。

① 机器人程序名 estun.main 对应的 16 进制数值分别为 0x7365、0x7574、0x2E6E、0x616D 和 0x6E69，将上述数值分别存入以 40054 开头的 5 个寄存器内。

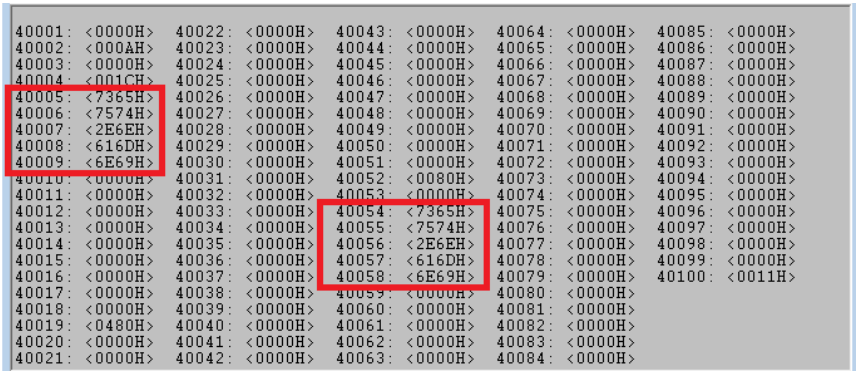


② 确保 40100 和 40052 两个寄存器里值为 0，首先往 40100 寄存器里写入 0x11，此时 40019 寄存器状态变为 0x801（该值表示当前系统处于等待命令状态），再往 40052 寄存器里写入 0x80，此时示教器上加载 estun.main，40019 寄存器状态变为 0x480(该值表示加载工程命令成功并继续等待新的控制权)，如果

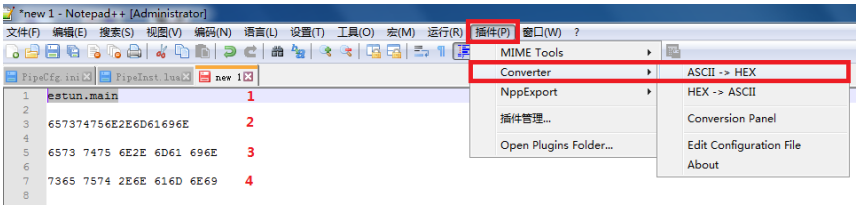
后续接着给机器人发送其它命令，应先把 40100 和 40052 两个寄存器内容清 0。

注：机器人程序名为 estun.main，有以下两种方法知道其对应的 16 进制数值：

① 在示教器端手动加载 estun.main 程序，将在以 40005 为首的 10 个寄存器内显示当前加载的程序名。



② 安装 Notepad++软件，很好用的一个文本软件，新建一页，在里面输入 estun.main，选中 estun.main 后，点击菜单栏 插件 -> Converter -> ASCII->HEX，将其转为 16 进制数，然后每隔四位分开，将每个字的高低字节调换顺序后就是我们需要的结果，如下图：

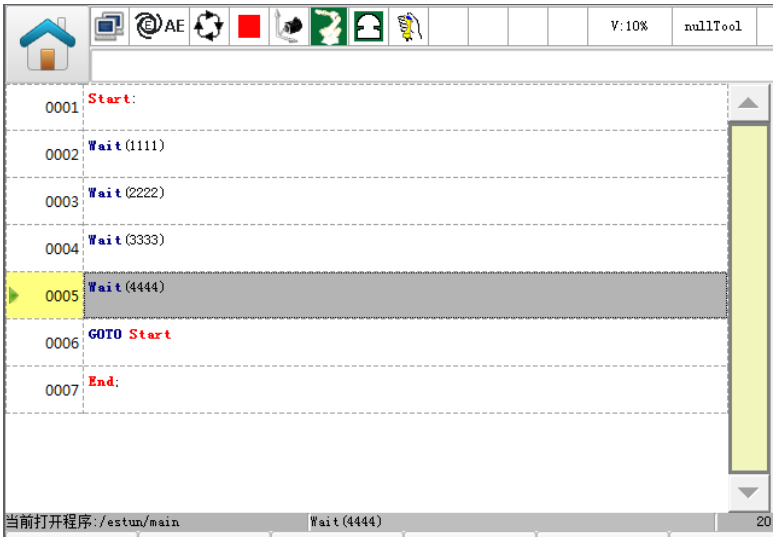


4.6 远程复位程序指针

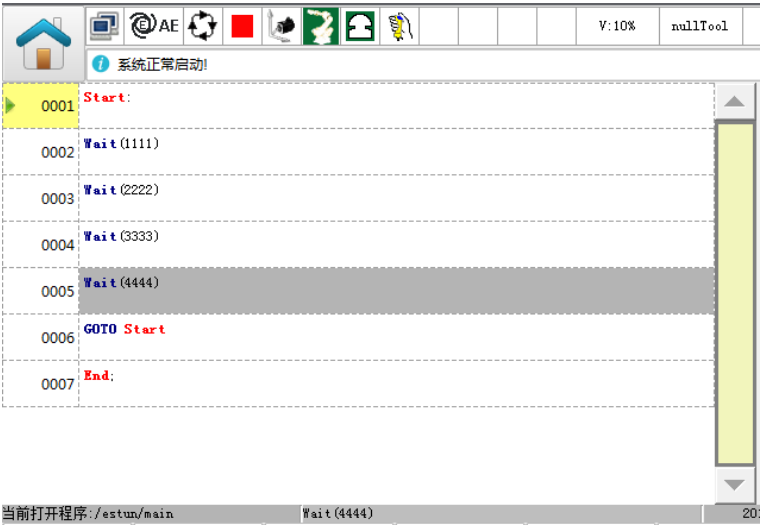
- ModScan（PLC）给机器人发送复位程序指针命令，机器人重新加载自启动程序，使 PC 指针回到第一行（如果机器人当前处于运动状态，同样响应该命令，先停止当前运动再重新加载自启动程序）。

通过查看本文第 3 章 ModBus Tcp 点表可知，所有给机器人发送命令的操作，前提有二：①机器人处于远程励磁状态；②所以命令操作皆配合读写标志寄存器 40100 且上升沿触发有效，远程复位程序指针对应的命令寄存器是 40052，值为 0x4000。

① 将 estun.main 设为启动程序，当前 PC 指针停留在第 5 行。



② 确保 40100 和 40052 两个寄存器里值为 0，首先往 40100 寄存器里写入 0x11，此时 40019 寄存器状态变为 0x801（该值表示当前系统处于等待命令状态），再往 40052 寄存器里写入 0x4000，此时机器人重新加载 estun.main 程序，PC 指针回到第一行，40019 寄存器状态变为 0x4400(该值表示复位程序指针命令成功并继续等待新的控制权)，如果后续接着给机器人发送其它命令，应先把 40100 和 40052 两个寄存器内容清 0。



4.7 调试注意事项

- 本文介绍默认机器人作为 Modbus Tcp 通信从站,外部设备作为 Modbus Tcp 通信主站，机器人只接受一个主站的连接，不可以多个客户端同时连接机器人（如有该需求请联系埃斯顿技术人员）；如果外部设备主站通信协议部分是由个人编制，可以先用 ModSim 测试功能可用后再去连接机器人。

- IO 通信部分机器人无论处于手动、自动或远程模式，系统都会响应处理，而命令操作部分，机器人必须处于远程且励磁状态，触发方式皆上升沿触发有效，按顺序先打开控制权限（40100）再下发对应的命令（40052）。

- 通信交互时首先应确定两端设备通信寄存器地址是正确的，否则肯定不会成功。当外部设备给机器人发送数据时，可以先确定下 SimDI 对应的寄存器，因为从示教器端看 SimDI 信号接收情况最直观，当 SimDI 对应的外部设备通信地址确定下来时，可以根据机器人通信点表顺序偏移来确定其它的通信寄存器地址，当外部设备接收机器人数据时，同样可以通过 SimDO 来确定地址。

- 当与外部设备交互时，外部设备可以通过机器人返回的状态信息判断命令是否下发成功，在 40019 寄存里每个命令都有对应的成功标志位，此外，在通

信过程中如果 40019 寄存器里错误位为 TRUE，则需要用下发重置状态机命令去清除错误，否则后续的命令系统皆不响应。

- 当机器人与西门子 PLC 进行 ModBus Tcp 通信交互模拟量数据时，由于西门子数据存储是根据低地址字节作为双字的高字节，高地址字节作为双字的低字节，所以在数据交互时需要特别注意对应关系。

```
52  //Read Robot Data to PLC
53  "Tag_1" := "NET_BUFF_1".MODBUS_READ[20]; //Robot 40021 -> PLC MW0
54  "Tag_2" := "NET_BUFF_1".MODBUS_READ[19]; //Robot 40020 -> PLC MW2
55  #x := DWORD_TO_REAL("Tag_3"); //MD0
56
57  //Write PLC Data to Robot
58  #y:=REAL_TO_DWORD("Tag_6");//MD4
59  "NET_BUFF_1".MODBUS_WRITE[16] := "Tag_4";//Write MW4 to 40069
60  "NET_BUFF_1".MODBUS_WRITE[15] := "Tag_5";//Write MW6 to 40068
```