

# MOTEC 伺服驱动器 MODBUS 使用手册

Version 2.1

MOTEC(中国)营业体系

2017-05-27

(本手册适用于 MOTEC 直流/交流伺服驱动器)

## 目 录

1. MODBUS 介绍 .....	3
1.1 物理层 .....	3
1.1.1 使用 RS232 组成网络 .....	3
1.1.2 RS485 通讯 .....	4
1.2 信号层 .....	4
1.3 通讯设定 .....	4
2. MODBUS 功能码概述 .....	5
2.1 01 号功能码<读取数字量输出状态> .....	5
2.2 02 号功能码<读取数字量输入状态> .....	6
2.3 03 号功能码<读取保持型寄存器> .....	6
2.4 05 号功能码<强制(写)单数字量输出状态> .....	7
2.5 06 号功能码<强制(写)单寄存器> .....	8
2.6 16 号功能码<强置写多个寄存器> .....	8
2.7 异常响应回复 .....	9
3. MODBUS 在 MOTEC 伺服读写驱动器状态 .....	10
3.1 设置参数 .....	10
3.1.1 使用 06 号功能码预置寄存器的值 .....	10
3.1.2 使用 16 号功能码预置寄存器的值 .....	11
3.2 读取当前寄存器内部状态 .....	12
3.3 读取(设置)当前 I/O 口的值 .....	12
3.3.1 读取数字输出口的值 .....	13
3.3.2 读取数字输入口的值 .....	13
3.3.3 设置数字输出口的值 .....	14
3.4 设置 life guard 功能 .....	16
3.4.1 检测时间设定 .....	16
3.4.2 作用模式设定 .....	16
3.5 保存所设置的参数到 FLASH 中 .....	17
4. MOTEC 驱动器 MODBUS 控制举例 .....	19
4.1 位置模式 .....	19
4.2 速度模式 .....	23
4.3 运动控制过程中运动状态的监视 .....	24
5. 连接触摸屏 .....	25
5.1 连接维纶触摸屏 .....	25
5.2 编程过程中需要注意的事项 .....	30
6. 附录: CRC 校验算法 .....	32
7. 联系方式 .....	33

## 1. MODBUS 介绍

MODBUS 协议是应用于控制领域的一种通用协议，它已经成为一通用工业标准。有了它，不同厂商生产的控制设备可以连成工业网络，进行集中监控和控制。MOTEC  $\beta$  伺服驱动器集成了 MODBUS 协议，方便用户通过各种上位机（PLC、HMI 或其它运动控制器）通讯。

MOTEC 伺服驱动器采用 RS232 或者 RS485 进行 MODBUS 通讯，使用此功能可驱动 MOTEC 伺服驱动器、变更参数以及监视 MOTEC 伺服驱动器状态等多项功能。

MODBUS 通讯协议分为 RTU 协议和 ASCII 协议，由于 RTU 模式在相同的波特率下比 ASCII 模式有更高的吞吐率，MOTEC 伺服驱动器选择支持 RTU 模式。

### 1.1 物理层

MOTEC 伺服驱动器提供 RS232 和 RS485 两种上位机连接方式来使用 MODBUS 通信协议，用户可根据自己的情况自行选择接口总线类型。

在连接 PC 机时，我们强烈建议用户使用计算机自带的串口进行通讯，而不是选用 USB 转串口的通讯方式，因为这种通讯方式通常兼容性不高，且误码率也比较高，给通讯的稳定性造成影响。如果使用者的电脑不带串口而必须使用 USB 转串口的设备，用户可以考虑我们所提供的 USB 转串口的配件，这样能保证更可靠的通讯。

#### 1.1.1 使用 RS232 组成网络

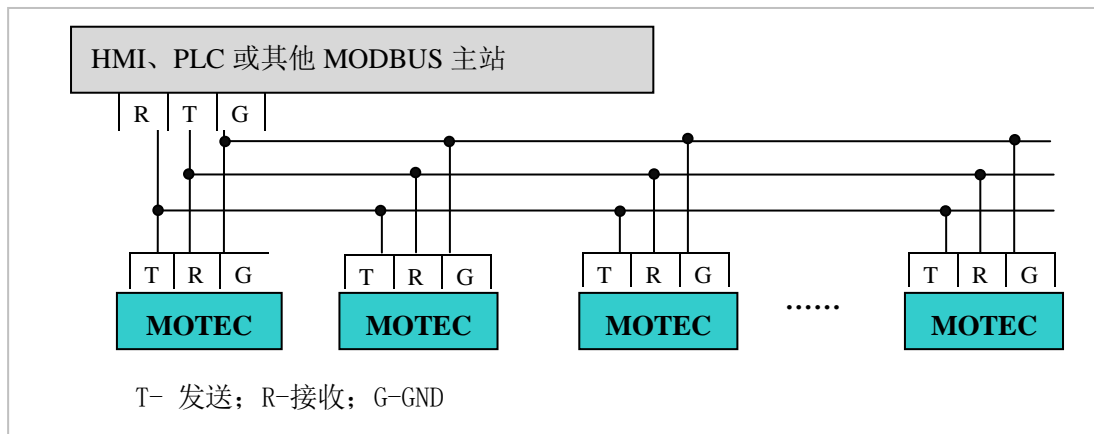


图 2 RS232 网络连接示意图

- 使用 RS232 通讯时，MOTEC 伺服驱动器支持最大通讯速率为 115200bps；
- RS232 通讯口最多可以支持连接 8 台驱动器联网；
- 支持 MOTECIAN 和 MODBUS 通讯协议；
- 支持 motionLib 函数库用于上位机是 PC 的应用；
- 实际通讯速率和通讯节点数量受使用环境与连接电缆长度等因素影响；

### 1.1.2 RS485 通讯

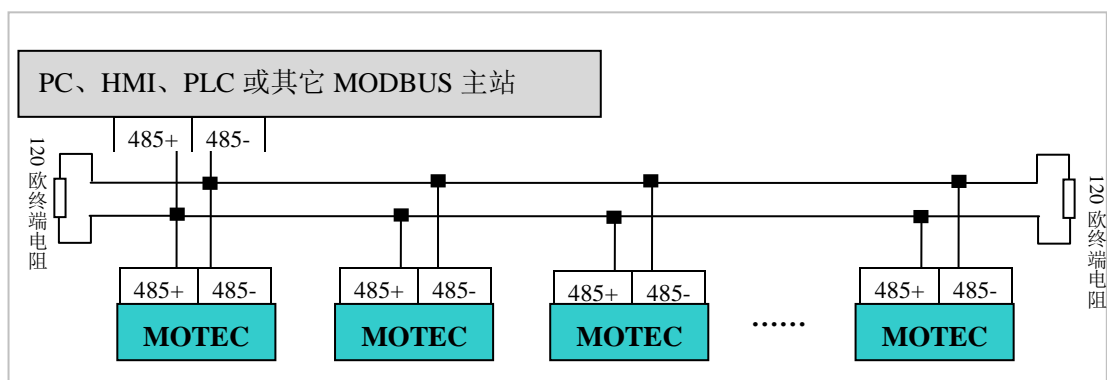


图 3 RS485 网络连接示意图

- RS-485 总线抗共模干扰能力增强，最大通信距离可达到 1000 米；
- 支持 31 台联网；
- MOTEC 伺服驱动器的 RS485 通讯支持最大传输速率为 115200bps；
- 支持 MOTECIAN 和 MODBUS 通讯协议；
- 支持 motionLib 函数库用于上位机是 PC 的应用；
- 实际通讯速率和通讯节点数量受使用环境与连接电缆长度等因素影响；

### 1.2 信号层

通讯传送分为独立的信息头，和发送的编码数据。以下的通讯传送方式定义也与 MODBUS RTU 通讯规约相兼容：

表 1. 驱动器通讯参数设置

编 码	起始位	数据位	校验位	停止位
8 位二进制	1 位	8 位	无	1 位

### 1.3 通讯设定

地址设定：驱动器的地址通过上位机软件 motionStudio 设定；而驱动器上电后默认的通讯速率为 115200bps，驱动器的通讯速率可以通过上位机利用通讯或 LED 操作面板（交流伺服）的方式进行修改。

修改通讯波特率并保存到 flash 中，重新启动驱动器后，新的波特率生效。

## 2. MODBUS 功能码概述

驱动器所支持的 MODBUS 功能码如表 2 所示，一共支持 01、02、03、05、06、和 16 共 6 种功能码，其操作方式请见以下的章节。

表 2. 驱动器支持的 MODBUS 功能码

代码	名称	描述	通讯格式
01	读取线圈(输出)状态	读取一组逻辑线圈的当前状态(ON/OFF)	RTU
02	读取输入状态	读取一组开关输入的当前状态(ON/OFF)	RTU
03	读取保持型寄存器	在一个或多个保持寄存器中读取当前二进制值	RTU
05	强制(写)单线圈(输出)状态	强制(写)一个逻辑线圈通断状态(ON/OFF)	RTU
06	强制(写)单寄存器	把二进制值写入一个保持寄存器	RTU
16	强制(写)多寄存器	把二进制值写入一串连续的保持寄存器	RTU

### 2.1 01 号功能码<读取数字量输出状态>

读取一组数字量输出 I / O 口的当前状态(ON/OFF)，不支持广播通信方式。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x01	功能代码
3	数字量输出起始地址高位	读取输出起始地址
4	数字量输出起始地址低位	
5	总位数高位	读取输出总位数
6	总位数低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x01	功能代码
3	字节数	字节数 $((N/8) + (N \bmod 8))$
N+3	字节<1>~字节<N>	N 字节数显示存储单元状态
N+4	CRC 校验码高位	CRC 校验码
N+5	CRC 校验码低位	

## 2.2 02 号功能码<读取数字量输入状态>

读取一组数字量输入 I / O 口的当前状态 (ON/OFF)，不支持广播通信方式。

### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x02	功能代码
3	数字量输入起始地址高位	读取输入起始地址
4	数字量输入起始地址低位	
5	总位数高位	读取输入总位数 N
6	总位数低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x02	功能代码
3	字节数	字节数 $((N/8) + (N \bmod 8))$
N+3	字节<1>~字节<N>	N 字节数显示存储单元状态
N+4	CRC 校验码高位	CRC 校验码
N+5	CRC 校验码低位	

## 2.3 03 号功能码<读取保持型寄存器>

在一个或多个保持寄存器中读取当前二进制值，不支持广播通信方式。

### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	寄存器起始地址高位	读取寄存器起始地址
4	寄存器起始地址低位	
5	总寄存器数 N 高位	寄存器个数 N
6	总寄存器数 N 低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码

3	字节数 (2*N)	字节数为 N 的 2 倍
4	第一个寄存器的高位	第一个寄存器的值
5	第一个寄存器的低位	
6	第二个寄存器的高位	第二个寄存器的值
7	第二个寄存器的低位	
~	~	~
2(N-1)+4	第 N 个寄存器的高位	第 N 个寄存器的值
2(N-1)+5	第 N 个寄存器的低位	
2(N-1)+6	CRC 校验码高位	CRC 校验码
2(N-1)+7	CRC 校验码低位	

#### 2.4 05 号功能码<强制(写)单数字量输出状态>

强制(写)一个数字量输出 I/O 口的通断状态(ON/OFF), 支持广播通信方式。在广播期间所有从站必须响应主站的查询, 但是不回应广播。

##### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x05	功能代码
3	数字量输出 I/O 地址高位	数字量输出 I/O 口地址
4	数字量输出 I/O 地址低位	
5	写数据高位	0x0000 表示 OFF, 0xFF00 表示 ON 其他无作用
6	写数据低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

##### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x05	功能代码
3	数字量输出 I/O 地址高位	数字量输出 I/O 口地址
4	数字量输出 I/O 地址低位	
5	写数据高位	写有效数据
6	写数据低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

## 2.5 06 号功能码<强制(写)单寄存器>

把二进制值写入一个保持寄存器,支持广播通信方式。在广播期间所有从站必须响应主站的查询,但是不回应广播。

### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	预置寄存器地址高位	预置寄存器的地址
4	预置寄存器地址低位	
5	预置寄存器值高位	需要预置寄存器的值
6	预置寄存器值低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	预置寄存器地址高位	预置寄存器的地址
4	预置寄存器地址低位	
5	预置寄存器值高位	预置寄存器的值
6	预置寄存器值低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

## 2.6 16 号功能码<强置写多个寄存器>

把二进制值写入一串连续的保持寄存器,支持广播通信方式。在广播期间所有从站必须响应主站的查询,但是不回应广播。

### ■ <命令信息>

字节	数据	说明
1	01~31	驱动器地址
2	0x10	功能代码
3	预置寄存器起始地址高位	预置寄存器起始地址
4	预置寄存器起始地址低位	
5	预置寄存器总数 N 高位	预置寄存器总数
6	预置寄存器总数 N 低位	
7	字节数<2*N>	预置寄存器字节数
8	预置第一个寄存器值高位	预置第一个寄存器值
9	预置第一个寄存器值低位	



10	预置第二个寄存器值高位	预置第二个寄存器值
11	预置第二个寄存器值低位	
.....	.....	.....
2(N-1)+8	预置第 N 个寄存器值高位	预置第 N 个寄存器值
2(N-1)+9	预置第 N 个寄存器值低位	
2(N-1)+10	CRC 校验码高位	CRC 校验
2(N-1)+11	CRC 校验码低位	

■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	10	功能代码
3	预置寄存器起始地址高位	预置寄存器的地址
4	预置寄存器起始地址低位	
5	预置寄存器个数 N 高位	预置寄存器的值
6	预置寄存器个数 N 低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

## 2.7 异常响应回复

MOTEC 伺服驱动器集成的 MODBUS 协议采用常用的 3 个错误代码来回应错误的命令信息，当下表任何情况发生时，从站必须发送一个异常响应。

代码	名称	说明
01	功能码错误	对于驱动器来说，询问中接收到的功能码是不可允许的操作。由于 MOTEC 伺服驱动器集成的 MODBUS 协议并不是使用所有的功能码，所以对于 01 02 03 05 06 16 号功能码除外的命令，驱动器将产生异常码 01。
02	寄存器错误，数据地址非法	对于驱动器来说，询问中接收到的数据地址是不可允许的地址。特别是参考号和传输长度的组合是无效的。对于带有 100 个寄存器的控制器来说，带有偏移量 96 和长度 4 的请求会成功，带有偏移量 96 和长度 5 的请求将产生异常码 02。
03	非法数据值	对于驱动器来说，询问中包括的值是不可允许的值。因为 MODBUS 协议不知道任何特殊寄存器的任何特殊值的重要意义，寄存器中被提交存储的数据项有一个应用程序期望之外的值。

■ <回应信息>

字节	数据	说明
1	1~31	驱动器地址
2	0x81	功能代码
3	异常响应代码	0x01~0x03
4	CRC 校验码高位	CRC 校验码
5	CRC 校验码低位	

### 3. MODBUS 读写驱动器状态

通过MODBUS协议上位机能方便地对MOTEC伺服驱动器进行参数调整、控制和在线监测等功能。本节将介绍MODBUS协议可以进行调整的参数和在线监测可以读取的参数，在第4节将专门介绍MODBUS协议如何对MOTEC伺服驱动器进行控制。

#### 3.1 设置参数

MOTEC伺服驱动器内置了丰富的参数用来供用户自定义和修改。在预置单个寄存器的值的时候，用户可以使用06号功能码或者16号功能码。

##### 3.1.1 使用06号功能码预置寄存器的值

使用06号功能码可以预置单个寄存器的值，注意，只有在该寄存器是可以写入的情况下才能使用该功能码。

例如使用06号功能码设置电机的模拟模式最大速度，电机模拟模式最大速度参数存放的地址是0x006D，向此地址写入模拟模式最大速度的设置值。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	模拟模式最大速度参数存放的寄存器地址
4	0x6D	
5	模拟模式最大速度高位	模拟模式最大速度值
6	模拟模式最大速度低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	模拟模式最大速度参数存放的寄存器地址
4	0x6D	
5	模拟模式最大速度高位	模拟模式最大速度参数存放的寄存器地址
6	模拟模式最大速度低位	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

### 3.1.2 使用16号功能码预置寄存器的值

使用16号功能码可以预置单个寄存器的值，需要注意，只有在该寄存器是可以写入的情况下才能使用该功能码，并且写入的所有寄存器的值必须是在可以选择的范围之内的。

例如使用16号功能码设置电机的运动距离。由于在驱动器内部地址存储的数据长度是16位的，而驱动器位置是32位的数据，所以需要同时设置运动距离的高位和低位。

#### ■ <命令信息>

字节	数据	说明
1	01~31	驱动器地址
2	0x10	功能代码
3	0x00	预置寄存器起始地址
4	0x26	
5	0x00	预置寄存器总数
6	0x02	
7	0x04	预置寄存器字节数
8	预置 0026h 寄存器值高位	预置 0026h 寄存器值
9	预置 0026h 寄存器值低位	
10	预置 0027h 寄存器值高位	预置 0027h 寄存器值
11	预置 0027h 寄存器值低位	
12	CRC 校验码高位	CRC 校验
13	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x10	功能代码
3	0x00	预置运动距离起始地址
4	0x26	
5	0x00	预置运动距离寄存器总数
6	0x02	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

注意：在MOTEC伺服驱动器内部地址和数据，高字节在前，低字节在后，例如对于驱动器设定的运动距离：0x12345678；在0x26h中存储的数据是0x1234，在0x27中存储的是0x5678，需要用户注意。

3.2 读取当前寄存器内部状态

在MOTEC 伺服驱动器内部，所有可以用户自定义和修改的寄存器参数都可以进行读取，另外还提供丰富的表征当前驱动器运动状态的值，但是这些值是只读的，不允许用户进行更改。

用户可以使用03号功能码读取这些参数。例如，使用03号功能码读取电机编码器当前位置。

■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x00	读取电机位置寄存器起始地址
4	0xAC	
5	0x00	寄存器个数 2
6	0x02	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x04	字节数为寄存器个数的 2 倍
4	0x00AC 寄存器的高位	0x00AC 寄存器的值
5	0x00AC 寄存器的低位	
6	0x00AD 寄存器的高位	0x00AD 寄存器的值
7	0x00AD 寄存器的低位	
8	CRC 校验码高位	CRC 校验码
9	CRC 校验码低位	

注意：在MOTEC 伺服驱动器内部地址和数据，高字节在前，低字节在后，例如假设电机的当前位置为0x12345678；在0x00AC中存储的数据是0x1234，在0x00AD中存储的是0x5678，需要用户注意。

3.3 读取（设置）当前I/O口的值

MOTEC 伺服驱动器提供了多路数字输入口以及多路数字输出口，另外还提供模拟量输入口。对于输入输出口的操作如下：

- 1、使用01功能码读取3路数字输出口的当前状态；
- 2、使用02功能码读取8路数字输入口的当前状态；
- 3、使用05功能码设置当前数字输出口的状态；

4、使用03功能码读取当前模拟量输入口的值；

### 3.3.1 读取数字输出口的值

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	01~31 驱动器地址
2	0x01	功能代码
3	0x00	读取输出起始地址
4	0x00	
5	0x00	读取输出总位数
6	0x09	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x01	功能代码
3	0x02	字节数
4	0x07	显示数字量单元状态
5	0x00	
6	CRC 校验码高位	CRC 校验码
7	CRC 校验码低位	

注意：读取数字输出口的状态在驱动器内部是以16位数据存储的，低位有效，高位用“0”补齐。在传输时，高八位在后，低八位在前。

另外在寄存器地址0x00C6中同样存储的是数字输出口的值，低位有效。可以使用03号功能码读取此参数。

### 3.3.2 读取数字输入口的值

数字输入的值在MOTEC 伺服驱动器内部的编号为INPUT1、INPUT2、……只能使用02号功能码来读取，例如读取5个输入I/O口当前的状态，1为高，0为低。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x02	功能代码
3	0x00	读取输入起始地址
4	0x00	
5	0x00	读取输入总位数
6	0x09	

7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x02	功能代码
3	0x02	字节数
4	0xC0	显示数字量单元状态
5	0x00	
6	CRC 校验码高位	CRC 校验码
7	CRC 校验码低位	

注意：读取数字输入口的状态在驱动器内部是以16位数据存储的，低位有效，高位用“0”补齐。在传输时，高八位在后，低八位在前。

另外在寄存器地址0x00C5中同样存储的是数字输入口的值，低位有效，高位用“0”补齐，可以使用03号功能码来读取此参数。

#### 3.3.3 设置数字输出口的值

数字输出口的值在MOTEC 伺服驱动器内部的编号为OUTPUT1, OUTPUT2, OUTPUT3, 使用05号功能码可以将数字输出口设置成1“导通”或者0“关闭”，但是一次只能设置一个输出口的状态。例如，设置OUTPUT2为高。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x05	功能代码
3	0x00	数字量输出 I/O 口地址
4	0x02	
5	0xFF	0000h 表示 OFF, FF00h 表示 ON 其他无作用
6	0x00	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x05	功能代码
3	0x00	数字量输出 I/O 口地址
4	0x02	
5	0xFF	0000h 表示 OFF, FF00h 表示 ON 其他无作用
6	0x00	

7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

注意：设置输出I/O口状态的时候，只能设置成0000h “关闭”或者FF00h “导通”，其余任何数据都是非法数据。

另外在寄存器地址0x0079中同样存储的是数字输出口的值，低位有效。可以使用06号功能码在0x00C6地址内的后6位写“0”或者写“1”来置高或者置低改输出口。

### 3.3.4 读取模拟量输入口的值

模拟输入口的值在MOTEC 伺服驱动器内部的地址为0x0109，只能使用03号功能码来读取。并且为只读模式。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x01	读取模拟量输入的值
4	0x09	
5	0x00	寄存器个数 1
6	0x01	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x02	字节数为寄存器个数的 2 倍
4	0x0109 寄存器的高位	0x0109 寄存器的值
5	0x0109 寄存器的低位	
6	CRC 校验码高位	CRC 校验码
7	CRC 校验码低位	

3.4 设置 lifeguard 功能

MOTEC 伺服驱动器在网络模式下，控制模式为电流模式、速度模式和位置模式具备 lifeguard 功能，使用 06 功能码设置。

参数表中Pr.274是lifeguard检测时间，参数Pr.276是作用模式。当Pr.274为非零值，参数Pr.276为9999时，启动lifeguard功能，在Pr.274时间内，上位机没有指令信号给驱动器，则认为上位机断线，驱动器会报警并disable电机。

3.4.1 检测时间设定

■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x01	Life guard 检测时间地址
4	0x12	
5	0x00	检测时间值，单位为 ms
6	0x00	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x01	Life guard 检测时间地址
4	0x12	
5	0x00	检测时间值，单位为 ms
6	0x00	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

3.4.2 作用模式设定

■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x01	Life guard 作用模式
4	0x14	
5	0x27	启用 life guard 功能
6	0x0F	
7	CRC 校验码高位	CRC 校验码



8	CRC 校验码低位	
---	-----------	--

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x01	Life guard 作用模式
4	0x14	
5	0x27	启用 life guard 功能
6	0x0F	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

### 3.5 保存所设置的参数到 FLASH 中

使用设置参数的功能更改 MOTEC 伺服驱动器内部参数的时候，并不是直接将参数设置到 FLASH 中，这样可以用来保证用户在误设置参数后能重启驱动器还原为上电状态的参数。待用户确认无误，可以永久保存更改的参数时，需要使用保存参数命令，将之前的更改保存到 FLASH 中，以便下次驱动器上电以后能保持该参数。

使用保存参数功能，用户需要使用写寄存器值的功能在寄存器 0x0035 中写“1”，如下表所示使用 06 号功能码触发该功能。

#### ■ <命令信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	预置寄存器的地址为 0x0035
4	0x35	
5	0x00	需要预置寄存器的值为 0x0001
6	0x01	
7	CRC 校验码高位	CRC 校验码
8	CRC 校验码低位	

#### ■ <回应信息>

字节	数据内容	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	预置寄存器的地址为 0x0035
4	0x35	
5	0x00	需要预置寄存器的值为 0x0001
6	0x01	
7	CRC 校验码高位	CRC 校验码

8	CRC 校验码低位	
---	-----------	--

注意：只有在向地址 0x0035 的寄存器写“1”该命令才能被触发，命令执行完毕之后，此寄存器的值会自动归零。

#### 4. MOTEC 驱动器 MODBUS 控制举例

通过 MODBUS 协议，可以将 MOTEC 伺服驱动器的控制模式设置为位置模式或者速度模式。下面将分别说明如何使用 MODBUS 协议来设置 MOTEC 伺服驱动器使电机运动在位置模式和速度模式。

注意：只有驱动器的操作模式是网络模式的时候才可以运用此功能。

##### 4.1 位置模式

MOTEC 伺服驱动器的 MODBUS 通讯支持 RS232 和 RS485，当驱动器运行于位置控制时，支持点到点的运动。位置控制的轨迹规划功能由 MOTEC 伺服驱动器自动实现，采用的是 S 曲线轨迹规划功能，上位机只需发送位置命令，由 MOTEC 伺服驱动器完成所有的其他功能。

使用 MODBUS 通信协议进行位置控制操作步骤：

1、设置操作模式：将操作模式（0x0024）设置成 “网络操作模式”

命令信息			回应信息	
字节	数据	说明	数据	说明
1	01~31	驱动器地址	01~31	驱动器地址
2	0x06	功能代码	0x06	功能代码
3	0x00	操作模式地址	0x00	操作模式地址
4	0x24		0x24	
5	0x00	设置成网络模式	0x00	设置成网络模式
6	0x00		0x00	
7	CRC 校验码高位	CRC 校验	CRC 校验码高位	CRC 校验
8	CRC 校验码低位		CRC 校验码低位	

2、设置控制模式：将控制模式（0x0025）设置成位置控制模式。

命令信息			回应信息	
字节	数据	说明	数据	说明
1	01~31	驱动器地址	01~31	驱动器地址
2	0x06	功能代码	0x06	功能代码
3	0x00	控制模式地址	0x00	控制模式地址
4	0x25		0x25	
5	0x00	设置成位置模式	0x00	设置成位置模式
6	0x02		0x02	
7	CRC 校验码高位	CRC 校验	CRC 校验码高位	CRC 校验
8	CRC 校验码低位		CRC 校验码低位	

3、设置运动模式：将运动模式（0x002D）设置成相对运动或者绝对运动（0-相对运行模式；1-绝对运行模式）。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	运动模式地址
4	0x2D	
5	0x00	设置运动模式
6	0x00 或者 0x01	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	运动模式地址
0x2D	
0x00	设置运动模式
0x00 或者 0x01	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

4、使能电机（0x0028）

向地址 0x0028 中写入 1，电机使能，如写入 0 则释放电机使能。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	使能电机寄存器地址
4	0x28	
5	0x00	使能电机
6	0x01	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	使能电机寄存器地址
0x28	
0x00	使能电机
0x01	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	使能电机寄存器地址
4	0x28	
5	0x00	释放电机使能
6	0x00	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	使能电机寄存器地址
0x28	
0x00	释放电机使能
0x01	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

### 5、设置运动距离（0x0026）

利用功能码 16 向地址 0x0026 写入一个 32 位有符号数，设定电机的运动距离。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x10	功能代码
3	0x00	预置寄存器起始地址
4	0x26	
5	0x00	预置寄存器总数
6	0x02	
7	0x04	预置寄存器字节数
8	0x0026 高位	0x26 高位
9	0x0026 低位	0x26 低位
10	0x0027 高位	0x27 高位
11	0x0027 低位	0x27 低位
12	CRC 校验码高位	CRC 校验
13	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x10	功能代码
0x00	预置寄存器起始地址
0x26	
0x00	预置寄存器总数
0x02	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

### 6、开始运动（0x0029）

利用功能码 06 向地址 0x0029 写入“1”开始运动。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	运动/停止寄存器地址
4	0x29	
5	0x00	开始运动
6	0x01	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	运动/停止寄存器地址
0x29	
0x00	开始运动
0x01	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

### 7、读取电机位置设定值（0x00AA）

利用 03 功能码从地址 0x00AA 读取当前电机的位置设定值。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x03	功能代码

回应信息	
数据	说明
01~31	驱动器地址
0x03	功能代码

3	0x00	读取寄存器起始地址
4	0xAA	
5	0x00	寄存器个数 N
6	0x02	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

0x04	读取的字节数
0x00AA 地址中的高位内容	
0x00AA 地址中的低位内容	
0x00AB 地址中的高位内容	
0x00AB 地址中的低位内容	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

8、读取电机位置实际值（0x00AC）

利用 03 功能码从地址 0x00AC 读取当前电机的位置实际值。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x00	读取寄存器起始地址
4	0xAC	
5	0x00	寄存器个数 N
6	0x02	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	
9		

回应信息	
数据	说明
01~31	驱动器地址
0x03	功能代码
0x04	读取的字节数
0x00AC 地址中的高位内容	
0x00AC 地址中的低位内容	
0x00AD 地址中的高位内容	
0x00AD 地址中的低位内容	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

4.2 速度模式

当 MOTEC 伺服驱动器运行于速度控制模式，上位机发来的速度设定值发生变化时，MOTEC 伺服驱动器会根据设定的加速度值进行加减速控制，从而不会导致电机速度突变对机械系统的运行造成影响。此时所采用的是 T 曲线轨迹规划功能。

- 使用 MODBUS 通信协议进行速度控制操作步骤：
- 1、将操作模式设置成（0x0024）“网络操作模式”  
具体内容详见 4.1 位置模式步骤 1
  - 2、将控制模式设置成（0x0025）“速度控制模式”  
向 0x0025 中写入 01，可以将控制模式设置成速度控制模式。

命令信息			回应信息	
字节	数据	说明	数据	说明
1	01~31	驱动器地址	01~31	驱动器地址
2	06	功能代码	06	功能代码
3	0x00	控制模式地址	0x00	控制模式地址
4	0x25		0x25	
5	0x00	设置成速度模式	0x00	设置成速度模式
6	0x01		0x01	
7	CRC 校验码高位	CRC 校验	CRC 校验码高位	CRC 校验
8	CRC 校验码低位		CRC 校验码低位	

- 3、设置速度（0x00AE）  
向 0x00AE 中写入速度值，即可改变速度设定值，此速度设定值单位为 rmp “转/分钟”，如果是正值则是正转，反之则电机反转。

命令信息			回应信息	
字节	数据	说明	数据	说明
1	01~31	驱动器地址	01~31	驱动器地址
2	0x06	功能代码	0x06	功能代码
3	0x00	电机速度设置寄存器地址	0x00	电机速度设置寄存器地址
4	0xAE		0xAE	
5	速度设置值高位	设置最高速度	速度设置值高位	设置最高速度
6	速度设置值低位		速度设置值低位	
7	CRC 校验码高位	CRC 校验	CRC 校验码高位	CRC 校验
8	CRC 校验码低位		CRC 校验码低位	

设置了电机速度以后，电机就会加速到设置速度并恒速运行，并且电机运动速度用户可以实时修改，但是速度不会马上达到设定值，而是根据预先设定的加速度值加速到设定速度。

- 4、使能电机（0x0028）  
向地址 0x0028 中写入 1，电机开始运动，如写入 0 则释放电机，电机停止运动。

电机启动运动：

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	使能电机寄存器地址
4	0x28	
5	0x00	使能电机
6	0x01	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	使能电机寄存器地址
0x28	
0x00	使能电机
0x01	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

电机停止运动：

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x06	功能代码
3	0x00	使能电机寄存器地址
4	0x28	
5	0x00	释放电机使能
6	0x00	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x06	功能代码
0x00	使能电机寄存器地址
0x28	
0x00	释放电机使能
0x00	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

5、读取电机实际运动速度（0x00AF）

利用 03 功能码从地址 0x00AF 读取当前电机的速度实际值，单位为 rmp “转/分钟”。

命令信息		
字节	数据	说明
1	01~31	驱动器地址
2	0x03	功能代码
3	0x00	读取寄存器起始地址
4	0xAF	
5	0x00	寄存器个数 N
6	0x01	
7	CRC 校验码高位	CRC 校验
8	CRC 校验码低位	

回应信息	
数据	说明
01~31	驱动器地址
0x03	功能代码
0x02	读取的字节数
0x00AF 地址中的高位内容	
0x00AF 地址中的低位内容	
CRC 校验码高位	CRC 校验
CRC 校验码低位	

4.3 运动控制过程中运动状态的监视

MOTEC 伺服驱动器可以实时更新当前的运动状态，例如故障代码、完成标识位，运动状态标识位，电机当前位置等状态。例如可以读取电机当前的故障代码（0x00C8）。



命令信息			回应信息	
字节	数据	说明	数据	说明
1	01~31	驱动器地址	01~31	驱动器地址
2	0x03	功能代码	0x03	功能代码
3	0x00	故障代码地址	0x04	0—无故障； 1—有故障
4	0xC8		故障代码 0x00C8 高位	
5	0x00	读取寄存器个数	故障代码 0x00C8 低位	
6	0x02		故障代码 0x00C9 高位	
7	CRC 校验码高位	CRC 校验	故障代码 0x00C9 低位	CRC 校验
8	CRC 校验码低位		CRC 校验码高位	
			CRC 校验码低位	

5. 连接触摸屏

MOTEC 伺服驱动器可以连接集成 MODBUS 协议的触摸屏，通过 MODBUS 进行通信，实现只用触摸屏对驱动器进行操作，控制电机实现各种运动。

下面将以这两种触摸屏为例，实现触摸屏对驱动器的简单控制功能。

5.1 连接维纶触摸屏

- 驱动器：MOTEC 伺服驱动器
- 操作模式：网络指令模式；
- 控制模式：位置模式；
- 实现功能：走一段位置模式点到点运动。
- 触摸屏：维纶触摸屏；（本例程中只说明触摸屏与驱动器通信相关的设置部分，至于对于如何使用触摸屏，请详细阅读相关厂家的说明书。）

维纶触摸屏所用的编程软件为 Easybuilder8000，在编写程序之前，通过分析得知，在操作模式为网络指令模式、控制模式为位置模式下，要想实现任意一段的距离，需要三个步骤：（1）电机使能；（2）设置所要走的距离；（3）启动电机运动。

1) 创建工程，并设置通讯属性

新建一个文件，需要进行如图 4 的系统设置：

如图 4 所示，设备属性需要进行如下设置，其中 PLC 类型必须要设置为 MODBUS RTU，接口类型有 RS232、RS485、以太网和 USB 可选，COM 端口和通讯速率等可根据实际使用的情况自己设定，本例程中使用的是 RS232 通讯接口，使用的端口为 COM1，通讯速度为 115200，无校验，数据位 8，停止位 1。PLC 预设站号为 1。通讯接口的设置如图 5 所示。



图 4. 维纶触摸屏系统参数设定



图 5. 维纶触摸屏通讯接口定义

## 2) 添加电机使能/释放按钮

如图 6 所示，设置一个位状态切换开关，用来控制电机的使能/释放，需要设置的开关属性如下（如图 7 所示）：

PLC 名称：MODBUS RTU；

设备类型：6x\_bit

地址：4100（地址格式：DDDDdd [范围：100~~6553515，dd 位地址：0~15]）

开关类型：切换开关；

状态 0/1 标签：0-enable，1-disable；

其中：PLC 名称是在建立新程序时，设定的名称；地址 4100，第 41 号参数是电机使能/释放参数，0-处于释放状态，1-处于使能状态；

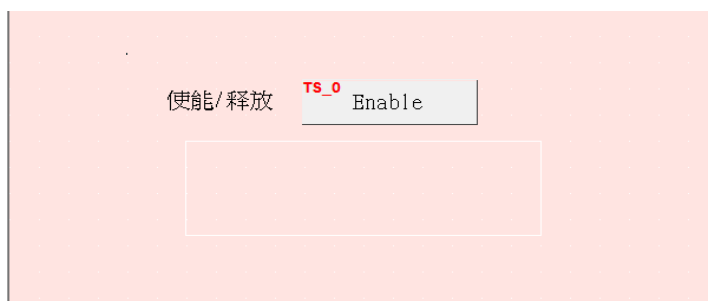


图 6. 添加电机使能/释放按钮

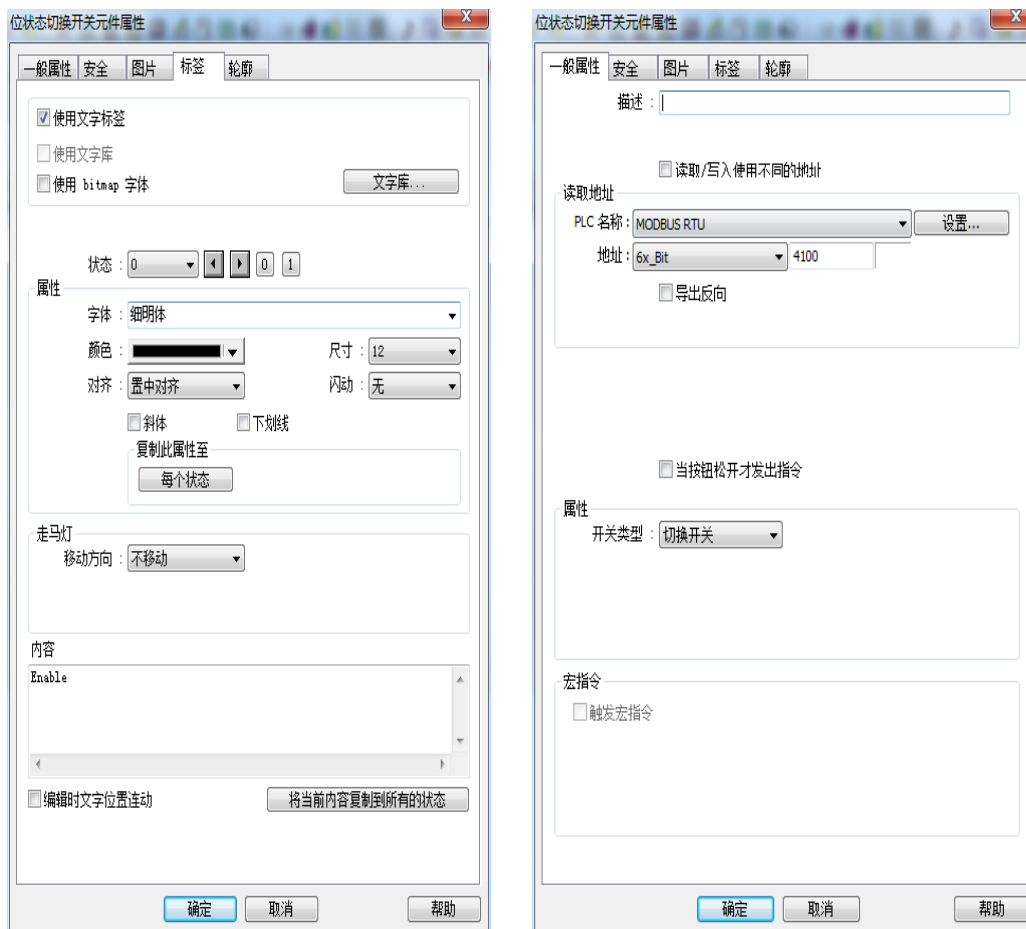


图 7. 使能/释放按钮属性设置

## 3) 添加运动距离数值元件

设置一个数值元件，用来设置电机运动点到点的距离，元件属性设置如下（如图 8 和图 9 所示）：

启动输入功能：因为要输入运动距离，所以要启用输入功能；

PLC 名称：MODBUS RTU

设备类型：5x

地址： 39 （地址格式：DDDD [范围：1~65535]）

资料格式：32-bit Signed

限制：输入常数，-2147483647~-2147483647

数字位数：小数点前 11 位，小数点后 0 位；

其中： 39 表示第 39、40 号参数组成的 32 位有符号数的高八位，组成的 32 位数表示所设置的运动距离；

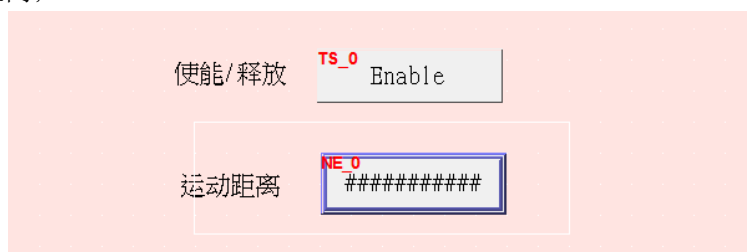


图 8. 添加运动距离编辑框

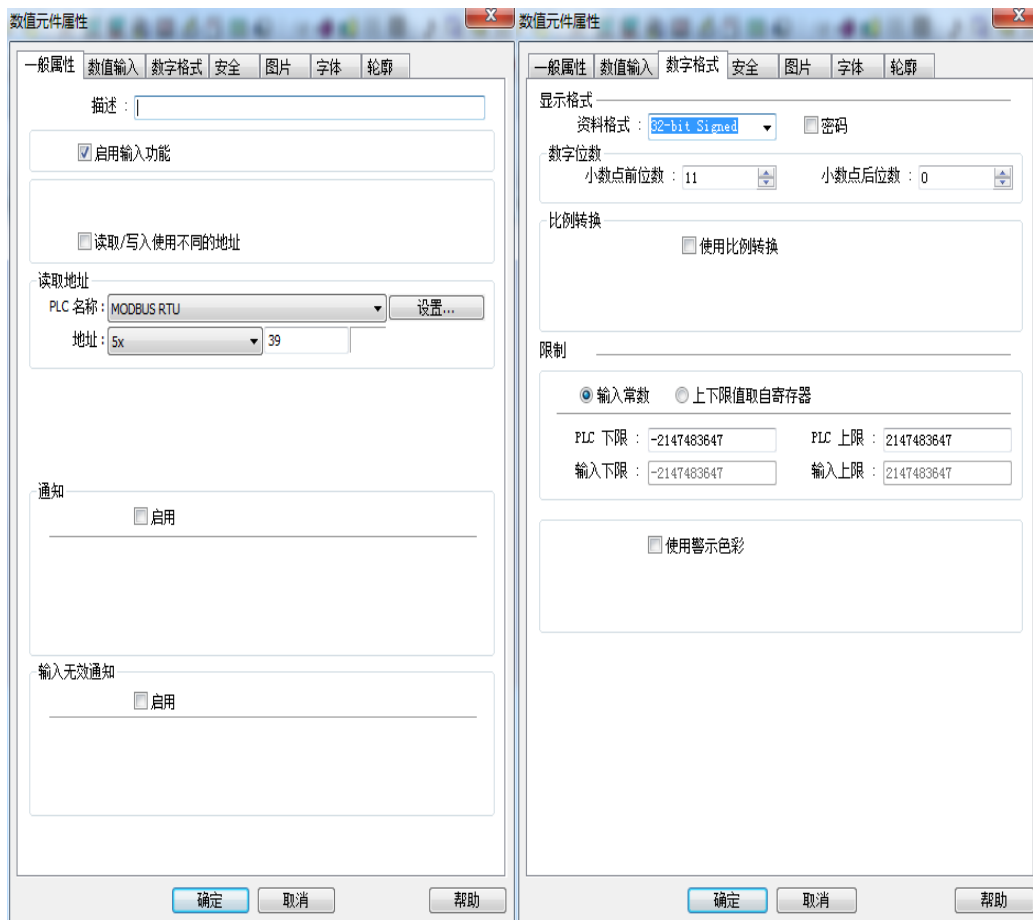


图 9. 运动距离对话框属性设置

## 4) 添加开始运动按钮

设置一个位状态切换开关，用作电机的启动按钮，开关元件属性设置如下(如图 10 和图 11 所示):

PLC 名称: MODBUS RTU

设备类型: 6x\_bit

地址: 4200 (地址格式: DDDDDdd [范围: 100~~6553515, dd 位地址: 0~15])

开关类型: 切换开关

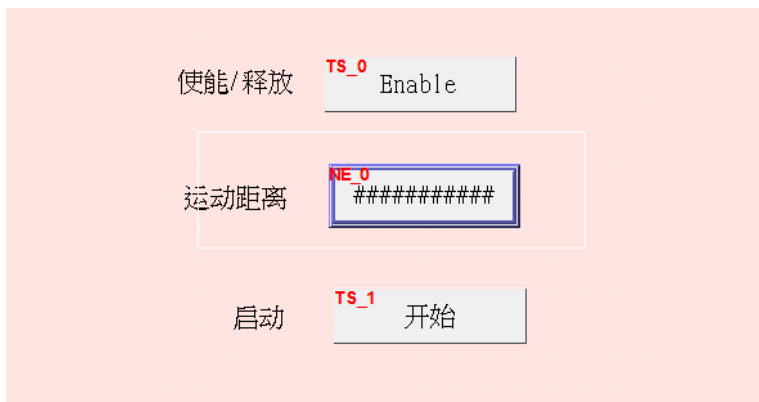


图 10. 添加开始运动按钮



图 11. 开始运动按钮属性设置

5) 程序编写完成后, 通过 USB 下载线可以把程序下载到触摸屏中去, 然后依次操作三个功能键便可实现电机的任意点到点的运动。

关于维纶触摸屏中 MODBUS 各地址的说明

软件中 MODBUS 协议的设备类型为 0x、1x、3x、4x、5x、6x, 还有 3x\_bit, 4x\_bit, 6x\_bit, 0x\_multi\_coils 等, 下面分别说明这些设备类型在 MODBUS 协议中支持哪些功能码。

0x: 是一个可读可写的设备类型, 相当于操作 PLC 的输出点。该设备类型读取位状态的时候, 发出的功能码 01H, 写位状态的时候发出的功能码是 05H。写多个寄存器时出的功能码 0FH。

1x: 是一个只读的设备类型, 相当于读取 PLC 的输入点。读取位状态的时候发出的功能码为 02H。

3x: 是一个只的设备类型, 相当于读取 PLC 的模拟量。读取数据的时候, 发出的功能码是 04H。

4x: 是一个可读可写的设备类型, 相当于操作 PLC 的数据寄存器。当读取数据的时候, 发出的功能码是 03H, 当写数据的时候发出的功能码是 10H, 可写多个寄存器的数据。

5x: 该设备类型与 4x 的设备类型是一样的。即发出读写的功能码完全一样, 不同之处在于: 当为双字时, 例如 32\_bit unsigned 格式的数据, 使用 5x 和 4x 两种设备类型分别读取数据时, 高字和低字的位置是颠倒的。例如, 使用 4x 设备类型读到的数据是 0x12345678, 那么使用 5x 设备类型读到的数据是 0x56781234。

6x: 是一个可读可写的设备类型, 读取数据的时候, 发出的功能码也是 03H, 与 4x 的不同之处在于写数据的时候发出的功能码是 06H, 即写单个寄存器的数据。

3x\_bit: 该设备类型发出的功能码与 3x 设备类型完全一致, 不同之处是, 3x 是读数据而 3x\_bit 是读数据中的某一位的状态。

4x\_bit: 该设备类型发出的功能码与 4x 设备类型完全一致, 不同之处是, 4x 是读数据而 4x\_bit 是读数据中的某一位的状态。

6x\_bit: 该设备类型发出的功能码与 6x 设备类型完全一致, 不同之处是, 6x 是读数据, 而 6x\_bit 是读数据中的某一位的状态。

0x\_multi\_coils: 该设备类型发出的功能码与 0x 设备类型完全一致, 不同之处是, 0x 是读 16 整数倍数位数的 bit, 而 0x\_multi\_coils 可读任意位数的 bit。

## 5.2 编程过程中需要注意的事项

触摸屏 MODBUS 地址: 根据实际对相应参数的读/写操作, 选择合适的命令, 如 5x, 6x, 6x\_bit, 3x 等。

驱动器参数地址: 根据驱动器的参数表的参数地址, 对应的参数号+1 后填入到参数地址位置, 例如电机使能的参数为 Pr.40, 则在参数地址位置填写 4100 (因所用的命令地址为 6x\_bit, 所以 41 要写成 4100); 电机速度参数号为 174, 则在参数地址位置填写 175.; 如果是 32 位数据, 则在参数地址处填写高八位所对应的参数号+1, 例如: 电机设定的距

离参数 Pr.38 和 Pr.39 组成 32 位的数据，则在参数地址处填入 39。

其他：其他的设置根据实际情况进行修改。例如：开关类型的选择，标签的修改等等（触摸屏具体操作，请参考厂家说明书）。

## 6. 附录：CRC 校验算法

以下的函数中 ptr 为指向需要校验的数据的指针，len 为数据长度。函数返回值为所得的 16 位 CRC 校验结果。

```
unsigned short getCRC16(unsigned char *ptr,unsigned char len)
{
    unsigned char i;
    unsigned short crc = 0xFFFF;
    if(len==0)
    {
        len = 1;
    }
    while(len--)
    {
        crc ^= *ptr;
        for(i=0; i<8; i++)

            if(crc&1)
            {
                crc >>= 1;
                crc ^= 0xA001;
            }
            else

                crc >>= 1;

        }
        ptr++;
    }
    return(crc);
}
```



## 7. 联系方式

Website: <http://www.motec365.com.cn>;

地址: 北京市通州区环科中路 17 号 11B;

服务热线: 010-56298855-666;

Email: [motecSupport@sina.com](mailto:motecSupport@sina.com);