

# 网络功能使用手册

2207



目录

网络功能使用手册 .....	3
> TCP 外部通讯 .....	3
TCP 通讯 .....	3
> 网络通讯类指令 .....	4
SENDMSG-发送数据 .....	4
PARSEMSG-解析数据 .....	4
READCOMM-读取 .....	5
OPENMSG-打开数据 .....	6
CLOSEMSG-关闭数据 .....	6
PRINT-输出信息 .....	7
MSG_CONN_ST-获取信息连接状态 .....	7
> 数据上传 .....	8
基本设置 .....	8
> 数据格式 .....	9
生成 csv 文件示例 .....	10
> 外部传输点 .....	11
> 通讯方式 .....	12
> 点位存放的数据 .....	13
> 指令 .....	13
MOVCOMM-外部点 .....	13

# 网络功能使用手册

## > TCP 外部通讯

### TCP 通讯

与外部设备进行通讯时，可以选择 TCP 通讯。

参数设置

设置 TCP 通讯需要进入“设置-TCP 通讯设置”界面。

参数	值	注释
IP	192.168.1.111	当前服务器端IP
端口	9000	通讯端口
帧头	@	数据帧头，留空为没有
分隔符	,	数据分隔符
结束符	!	数据帧尾，留空为没有
进制	十进制	接收到的数据按该进制解析

工艺号：支持 9 个工艺号。

连接开关：方式为客户端时白色代表断开，绿色代表连接；方式为服务器时白色代表关闭，绿色代表打开。（连接上该界面会提示通讯连接成功）

方式：将控制器作为服务器或客户端。（控制器为客户端或服务器时都可使用不同的工艺号与多台外部设备进行通讯和数据收发；注：控制器为客户端多个工艺号与多台服务器设备连接时工艺号之间的 IP 和端口不能一致，控制器为服务器多个工艺号与多台客户端设备连接时端口不能一致）

IP：当控制器作为服务器（方式选择为服务器），此处为控制器 IP 无需修改。当控制器作为客户端，此处需设置为外部通讯设备的 IP。

端口：方式为服务器时，为本机监听端口，供客户端连接；方式为客户端时，为连接服务器端口。

帧头：数据通讯时，控制器接收外部设备消息时的帧头，可修改。

分隔符：数据通讯时，控制器接收外部设备消息时的分隔符，可修改。

结束符：数据通讯时，控制器接收外部设备消息时的结束符，可修改。

进制：将要接收的十进制或十六进制数据选择对应的进制再以 10 进制解析后输出。

注:TCP 通讯进行连接时,首先将控制器 IP 与外部设备 IP 设置为同一网段,如 192.168.1.xxx,若设置控制器为客户端,则外部设备为服务端,再将网络设置中的 IP 和端口与外部设备网络调试软件中的 IP 和端口设置一致,连接开关打开,提示连接成功

## > 网络通讯类指令

### SENDMSG-发送数据

该指令用于向已连接的外部设备发送数据,通过选择对应的工艺号可以发送字符串和变量。字符串与变量可以混合发送。向外部设备发送信息不使用在“设置-TCP 通讯设置”界面设置的帧头、分隔符、结束符、进制。

如果要发送变量,则在变量前加入\$。

SENDMSG		
参数	值	注释
ID	1	工艺号(1-9)
发送字符	\$GD001=123	
若要发送变量,则在变量前加入\$ 例如,需要发送D001的值,则填入SENDMSG ID = 1 # \$D001#		

例:

前提: GD001=123, I001=10

需要向网络设置工艺号为 3 的上位机发送数据“The value of GD001 is 123, and the value of I001 is 10”

插入指令 SENDMSG 为:

ID=1

发送字符: The value of GD001 is \$GD001, and the value of I001 is \$I001

### PARSEMSG-解析数据

该指令用来解析外部设备传来的一组数据。

该指令会将外部设备传来的数据分别存于多个全局变量中,本指令要设置的为第一个变量。

解析后不清除缓存区:外部设备发送的数据会暂时存在控制器的缓存区; a、不清除缓存区在第一次解析后下一组数据还没有发送之前数据仍存在缓存区; b、清除缓存区在第一次解析完成后会对缓存区的数据进行清除。

PARSEMSG		
参数	值	注释
ID	1	工艺号(1-9)
数据存放的首个变量	S001 <span>更多</span>	查询的数据依次存放的首位置
解析后清除缓存区	否	
数据存放数	<span>更多</span>	记录提取数据的数量
示例: PARSEMSG ID = 1 GI001 CLEARCACHE = 0 I001		
<p>当有TCP接收到多位数值时, 会将数值分别存入多个变量中, 所使用的变量分别为第一位变量、第一位变量往下顺延。</p> <p>即, 若发来3位数值, A、B、C, 设置的第一位变量名为GI006, 则将A存入GI006、B存入GI007、C存入GI008。</p>		

例:

帧头: @

分隔符: ,

结束符: !

PARSEMSG 指令第一位变量类型 GDOUBLE, 第一位变量名 GD003.

外部设备发送数据: @,12,6,47,102,77.88,!

则 EXPLAIN 指令将这 5 个数值分别存于 GD003、GD004、GD005、GD006、GD007 中。

GD003=12、GD004=6、GD005=47、GD006=102、GD007=77.88

GD003=12

GD004=6

GD005=47

GD006=102

GD007=77.88

解析后清除缓存区: 是或否

## READCOMM-读取

读取以太网或 Modbus 发送的点位存到位置变量中、个数存到数值变量中。

注: 使用方法同《外部点功能》, 该指令目前仅支持 Modbus

READCOMM			
参数	值		注释
工艺号	1		1-9
通讯方式	ETHERNET		以太网或Modbus
位置变量类型	GP0001	更多	已存点位: 0
变量名	I001	更多	I001,GI001
示例:READCOMM ID = 1 EHTERNET TO P0001 I001			

工艺号：要打开通讯的网络通讯的工艺号。

通讯方式：使用以太网通讯或者 Modbus 通讯。

位置变量类型：可选全局位置变量、局部位置变量。

变量名：存接收到点位的数量

## OPENMSG-打开数据

打开对应工艺号的网络通讯。运行 OPENMSG 指令后即打开通讯。（连接通讯）

OPENMSG		
参数	值	注释
ID	1	工艺号(1-9)
示例: OPENMSG ID = 1		

工艺号：要连接网络通讯的工艺号。

## CLOSEMSG-关闭数据

关闭对应工艺号的网络通讯。运行 CLOSEMSG 指令后即关闭通讯。（断开通讯）

CLOSEMSG		
参数	值	注释
ID	1	工艺号(1-9)
示例: CLOSEMSG ID = 1		

工艺号：要断开网络通讯的工艺号。

## PRINT-输出信息

屏幕输出指令，以三种形式在示教器上显示内容。可输出自定义字符或变量的数据

工程预览/程序指令/指令插入/参数设定			
PRINTMSG			
参数	值		注释
类型	消息	警告	错误
输出字符	\$GD001		
示例: PRINTMSG #输入内容#			

输出信息现分成三个类型：消息、警告、错误。

输出字符：输出字符。可输入任意字符（支持转义符），同时可以输出变量，如 GD001 变量，GD001=10;在输出信息消息、警告、错误指令里分别输入：\$GD001

此时运行或单步这条指令时，示教盒右下角弹出的分别是：

白色消息内容为：10;

黄色警告内容为：10;

红色错误内容为：10;

## MSG\_CONN\_ST-获取信息连接状态

MSG_CONN_ST		
参数	值	注释
工艺号	1	网络设置的工艺号
状态存入变量名	B001	更多 变量名
示例:MSG_CONN_ST 1 B001		

工艺号：判断网络通讯连接状态的工艺号。

存入变量类型：将通讯状态存入到局部的 BOOL 变量或者全局的 GBOOL 变量。

存入变量名：将通讯状态存入的变量的变量名。

读取当前工艺号网络通讯状态到对应的全局布尔或局部布尔变量中。通讯正常则存入的值为 1，通讯失败则存入的值为 0。

## > 数据上传

### 基本设置

数据上传功能可以定时自动采集并上传当前机器人运行状态、参数，并将数据整合成 csv、txt 文件上传到指定服务器。

在设置-数据上传中点击【修改】按钮以设置相关连接 ftp 服务器所需要的参数。



设置/数据上传

数据传输开关:

上传方式: FTP

文件格式: csv

服务器IP: 192.168.1.233

端口: 5050

用户名: inxbot

密码: password

路径: /robot/

数据采集周期: 1 s

数据上传周期: 20 s

是否发送说明文件:

返回 保存 数据格式

数据传输开关：打开后则开始连接 ftp 服务器并上传数据。在所有参数填写好之后再打开该开关，本开关打开后，开机将自动开始采集并上传数据。

上传方式：当前仅支持 ftp 协议。所以在使用本功能之前请先拥有一个 ftp 服务器。

文件格式：当前支持 csv 与 txt 格式。其文件内容相同，文件格式不同。csv 格式更方便进行数据的统计。

服务器 IP：ftp 服务器的 ip 地址，请保证本控制器与 ftp 服务器在同一个网络内，保证其网关相同（控制器网关在设置-系统设置-IP 设置内进行查看和修改）。

端口：ftp 服务器的 ftp 协议所使用的端口。一般的 ftp 协议使用的默认端口为 21。

用户名：登录 ftp 服务器所使用的用户名。需先在 ftp 服务器处创建好一个用户。

密码：登录 ftp 服务器所使用的密码。

路径：文件上传到 ftp 服务器的路径。本路径是相对于 ftp 根目录的路径。

数据采集周期：根据设定的时间，每隔一定时间，控制器采集一次当前数据并存入要发送的文件中。

数据上传周期：根据设定的时间，每隔一定时间，控制器将已采集好数据的文件发送到 ftp 服务器指定的目录下。

是否发送说明文件：说明文件在开机或打开【数据传输开关】后第一次发送数据文件前发送。内容可自定义，一般用来说明当前机器人的序号等信息。若本开关关闭，则不发送说明文件。

## > 数据格式

配置好 ftp 的连接相关参数后则需要配置发送的数据文件中的数据格式。在设定数据格式时使用特殊字符串代表所需要发送的参数。例如要发送当前的日期，格式如下“2019-03-07”，则需在数据格式中填写如下：“\$Y\$%- \$m\$%- \$d\$%”（不包括引号）。

生成的文件若要 csv 格式，每一项之间要用英文逗号分割。

特殊字符串代表参数如下：

IP地址	\$IP%	
MAC地址	\$MAC%	
日期		
年	\$Y%	
月	\$m%	
日	\$d%	
时间		
时	\$H%	
分	\$M%	
秒	\$S%	
状态代码	\$StatusCode%停止0, 暂停1, 运行2	
错误代码	\$ErrorCode%	
J1-J6转速	\$RPM_j1%	\$RPM_j2% ...
J1-J6扭矩	\$Torsion_j1%	\$Torsion_j2% .
..		
J1-J6负载	\$Load_j1%	\$Load_j2% ...
变量		
全局整型变量	\$GI001%	\$GI002% ...
全局浮点变量	\$GD001%	\$GD002% ...
全局布尔变量	\$GB001%	\$GB002% ...

## 生成 csv 文件示例

希望得到的结果如下

说明文档文件名： Robot-R1\_年-月-日\_时:分:秒\_INFO

说明文档内容： Robot-R1,年-月-日,时:分:秒,本机 IP,本机 MAC,技术部,加工零件,1 轴电机转速,2 轴电机转速,3 轴电机转速,4 轴电机转速,5 轴电机转速,6 轴电机转速,1 轴电机扭矩,2 轴电机扭矩,3 轴电机扭矩,4 轴电机扭矩,5 轴电机扭矩,6 轴电机扭矩,1 轴电机负载,2 轴电机负载,3 轴电机负载,4 轴电机负载,5 轴电机负载,6 轴电机负载,当前控制器状态,当前错误代码

数据文档文件名： Robot-R1\_年-月-日\_时:分:秒\_DATA

数据内容： Robot-R1,年-月-日,时:分:秒,本机 IP,本机 MAC,1 轴电机转速,2 轴电机转速,3 轴电机转速,4 轴电机转速,5 轴电机转速,6 轴电机转速,1 轴电机扭矩,2 轴电机扭矩,3 轴电机扭矩,4 轴电机扭矩,5 轴电机扭矩,6 轴电机扭矩,1 轴电机负载,2 轴电机负载,3 轴电机负载,4 轴电机负载,5 轴电机负载,6 轴电机负载,当前控制器状态,当前错误代码

所编写的格式如下：

说明文档文件名： Robot-R1\_ \$Y%-\$m%-\$d%\_ \$H%:\$M%:\$S%\_INFO

说明内容： Robot-R1,\$Y%-\$m%-\$d%,\$H%:\$M%:\$S%, \$IP%,\$MAC%,技术部,加工零件,\$RPM\_j1%,\$RPM\_j2%,\$RPM\_j3%,\$RPM\_j4%,\$RPM\_j5%,\$RPM\_j6%,\$Torsion\_j1%,\$Torsion\_j2%,\$Torsion\_j3%,\$Torsion\_j4%,\$Torsion\_j5%,\$Torsion\_j6%,\$Load\_j1%,\$Load\_j2%,\$Load\_j3%,\$Load\_j4%,\$Load\_j5%,\$Load\_j6%,\$StatusCode%,\$ErrorCode%

数据文档文件名： Robot-R1\_ \$Y%-\$m%-\$d%\_ \$H%:\$M%:\$S%\_DATA

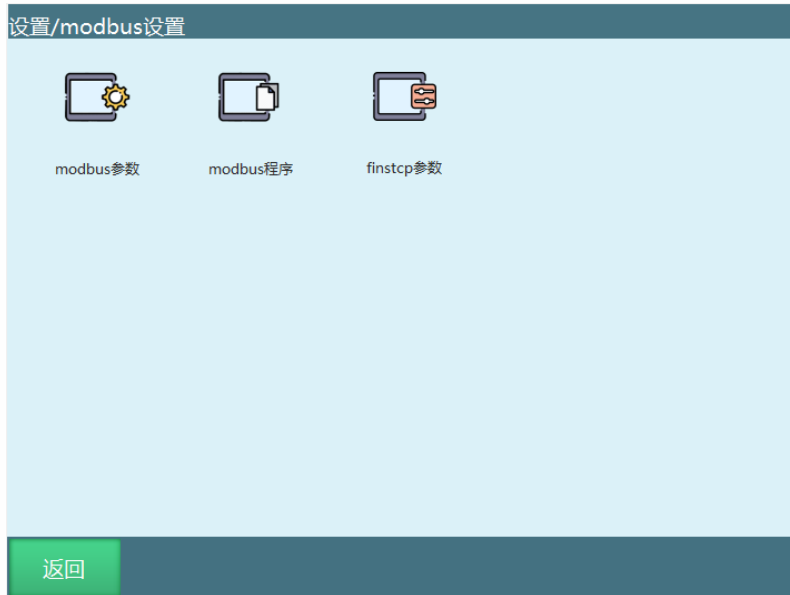
数据内容： Robot-R1,\$Y%-\$m%-\$d%,\$H%:\$M%:\$S%, \$IP%,\$MAC%,\$RPM\_j1%,\$RPM\_j2%,\$RPM\_j3%,\$RPM\_j4%,\$RPM\_j5%,\$RPM\_j6%,\$Torsion\_j1%,\$Torsion\_j2%,\$Torsion\_j3%,\$Torsion\_j4%,\$Torsion\_j5%,\$Torsion\_j6%,\$Load\_j1%,\$Load\_j2%,\$Load\_j3%,\$Load\_j4%,\$Load\_j5%,\$Load\_j6%,\$StatusCode%,\$ErrorCode%

\*涉及轴的参数需要手动输入哪个轴，如 1 轴转速：\$RPM\_j%需要在 j 后面写 1

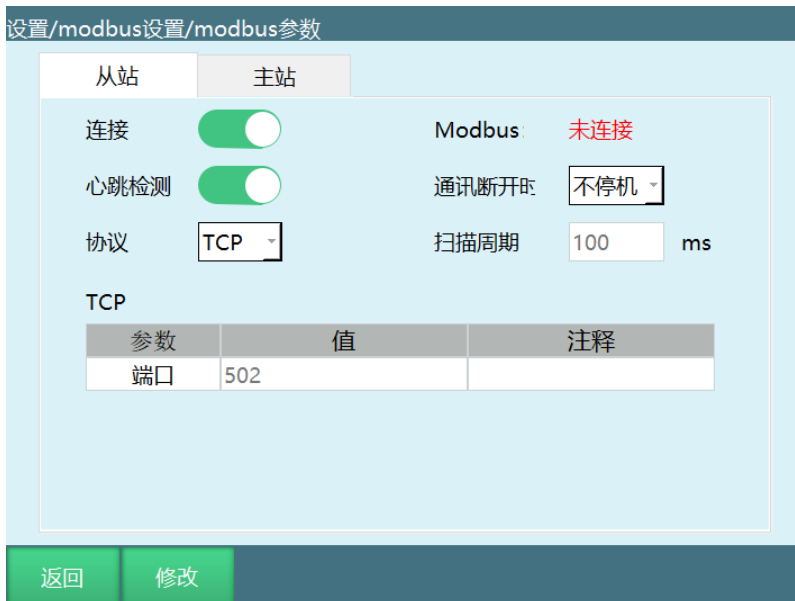
## > 外部传输点

### 参数设置

外部通讯可使用 modbus，设置参数需要进入“设置-modbus 设置-modbus 参数”界面。（也可查看 modbus 相关手册）



开关的检测与状态显示的参数



连接：modbus 的开关，打开后检测 modbus 信号。

心跳检测：打开后用于检测 modbus 与控制器之间的收发频率，断开 modbus 连接后心跳检测显示数据收发关闭。

Modbus: 显示 modbus 与控制器之间的连接状态。

通讯断开时: 不停机: 当 Modbus 从站断开、通讯断开时不会停止运行或下电;

停机: 当 Modbus 从站断开、通讯断开时会停止运行或下电。

扫描周期: 执行一次扫描操作所需要的时间。

在这个界面可以设置 modbus 是否进行连接、modbus 连接所使用的协议、本控制器为 modbus 主站/从站以及当连接时的各个参数。

本机为主站

TCP (端口): 从站的连接端口

RTU (端口): 从站 modbus 连接的端口

RTU (ID): 从站的 ID

RTU (波特率): modbus 的波特率, 需要设置

本机为从站

TCP (端口): 本机用来连接的端口, 需要设置

RTU (ID): 本机用来连接的 ID, 需要设置

RTU (端口): 本机用来连接的端口, 需要设置

RTU (波特率): 本机用来连接的波特率, 需要设置

## > 通讯方式

因地址码限制, 对于过多的点位需要进行分次发送, 每次最多发送 30 个点。

只要控制器与 PLC 连接上便可发送点位, 控制器会自动存储。

用途	地址码	过程
全部点位发送标志	1001	PLC 需发送点位时将其置 1, 发送结束后将其置 2, 控制器接收完毕将其置 0。
发送一次的发送标志	1002	PLC 需发送点位时将其置 1, 控制器接收完将其置 0, PLC 再次将其置 1 来进行下一次的发送过程
发送一次的点位数量	1003	PLC 一次发送时的点位数量, 最多 30 个
点位存放的数据	根据个数	下面详解
每帧数据的帧编号	1004	每一次发点都要更改编号数字, 不可与上次相同

清空控制器点位队列标志	1005	若要抛弃已发送给控制器的点位队列，PLC 将其置 1，控制器清除之后将其置 0。
-------------	------	--

## > 点位存放的数据

一个点位数据包含 1 个坐标系和 6 个轴的值（若为四轴机器人，则包含 1 个坐标系和四个轴的值）。

第 i 个点位	地址码	注释
坐标系	$1010+20*(i-1)$	$1 \leq i \leq 32$
是否使用	$1011+20*(i-1)$	$1 \leq i \leq 32$ ; 发 0 使用，发 1 不使用
第 j 个轴的值	$1010+2+20*(i-1)+2*(j-1)$	$1 \leq i \leq 32, 1 \leq j \leq 9$ , 轴的值使用 float 类型，所以占用 2 个地址

示例

需要发送 88 个点位。由于每次只能发 32 个，所以需要分为 3 次发送，发送的个数分别为 32、32、24。

过程如下：

PLC 设置 1003 为 32，设置点位存放数据所用各个地址码的值，设置 1001 为 1，设置 1002 为 1；

控制器检测到 1002 为 1,1001 为 1，则根据 1003 的值取出点位存放地址码的数据，然后设置 1002 为 0；

PLC 检测到 1002 位 0，设置 1003 为 32，设置点位存放地址码的数据，然后设置 1002 为 1；

控制器检测到 1002 的值为 1，1001 为 1，根据 1003 的值取出点位存放地址码的数据，然后设置 1002 为 0；

PLC 检测到 1002 的值为 0，设置 1003 为 24，设置点位存放地址码的数据，设置 1001 为 2，设置 1002 为 1；

控制器判断到 1002 为 1,1001 为 2，根据 1003 的值取出点位存放地址码的数据，然后设置 1002 的值为 0，再将 1001 设置为 0。

## > 指令

### MOVCOMM-外部点

该指令用于将控制器中存放的点位按照设置的插补方式进行运动。

MOVCOMM			
参数名	参数		注释
插补方式	关节		
VJ	10	更多	速度范围1-100
PL	0	更多	平滑过渡 (0-5)
ACC	20	更多	运动加速度
DEC	20	更多	运动减速度
TIME	0	更多	提前执行,自然数(ms)
示例: MOVCOMM MOVL VJ = 10% PL = 0 ACC = 10 DEC = 10			

插补方式：

- a. 关节
- b. 直线
- c. 曲线

运动时插补方式，所有点位均用该插补方式运动

速度：插补方式为关节，1-100

其它插补方式，2-1000

运动时的最大速度。

平滑（PL）：0-5，插补方式为曲线时可填0，平滑过渡。

加速度：运行时的最大加速度。

减速度：运行时的最大减速度。

时间：提前执行时间可提前执行下一条可提前执行指令。

参数来源：可自定义或绑定变量。