

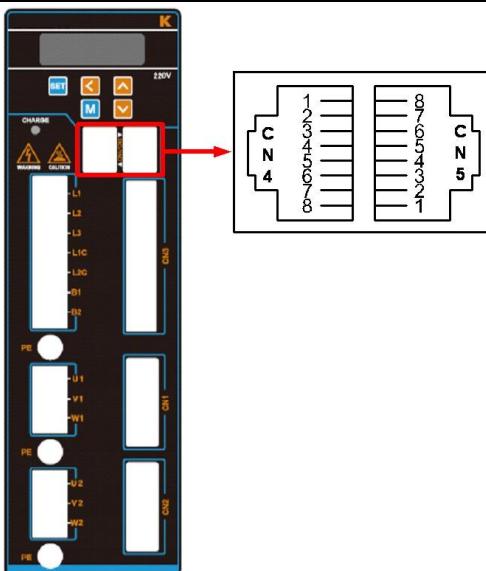
## 第六章 通讯

K 系列伺服驱动器标配 RS485 接口的 MODBUS 通讯，以及选配 CAN 接口的 CANopen(符合 DS301、DS402 标准协议)。本章节主要描述 MODBUS 通讯。

### 6.1 通讯接线

通讯连接器信号名称及其功能如下：

端子号	1	2	3	4	5	6	7	8
名 称	CN4	CANH-	CANL	GND	GND	RS485+	RS485-	保留
	CN5	CANH-	CANL	GND	GND	RS485+	RS485-	内置 120 欧电阻



伺服驱动器 CN4 总是作为通讯电缆输入端子，CN5 总是作为通讯电缆输出端子。

### 6.2 用户参数

用户参数		意义
P□600	H. □□□0	RS485 通讯波特率：4800 bps
	H. □□□1	RS485 通讯波特率：9600 bps
	H. □□□2	RS485 通讯波特率：19200 bps
	H. □□□3	RS485 通讯波特率：38460 bps
	H. □□□4	RS485 通讯波特率：57600 bps
P□600	H. □□0□	ASCII 方式，7 位数据位，无校验，2 位停止位
	H. □□1□	ASCII 方式，7 位数据位，偶校验，1 位停止位
	H. □□2□	ASCII 方式，7 位数据位，奇校验，1 位停止位
	H. □□3□	ASCII 方式，8 位数据位，无校验，2 位停止位
	H. □□4□	ASCII 方式，8 位数据位，偶校验，1 位停止位
	H. □□5□	ASCII 方式，8 位数据位，奇校验，1 位停止位
	H. □□6□	RTU 方式，8 位数据位，无校验，2 位停止位
	H. □□7□	RTU 方式，8 位数据位，偶校验，1 位停止位
	H. □□8□	RTU 方式，8 位数据位，奇校验，1 位停止位

<b>P□601</b>	RS-485通讯轴地址			速度	位置	扭矩
	设定范围		设定单位	出厂设定		电源重起
	1 ~ 127		—	1 (A轴) 2 (b轴)		需要
<b>P□602</b>	RS-485通讯超时			速度	位置	扭矩
	设定范围		设定单位	出厂设定		电源重起
	0 ~ 1000		100ms	0		不需要
<ul style="list-style-type: none"> <li>• P□602设置为零时，关闭通讯超时检测；</li> <li>• P□602设置为大于零时，表示必须在设定的时间内通讯，否则将出现通讯错误。举例，P□602设置成50时，表示必须每5秒与伺服驱动器通讯1次。</li> </ul>						

## 6.3 MODBUS 通讯协议

使用 RS-485 通讯时，每一台伺服驱动器必须预先设置参数 P□600 ~ P□601。通讯方式使用 MODBUS 协议，其中可使用下列两种模式：

ASCII 模式

RTU 模式。

以下说明 MODBUS 通讯。

### ■ 编码意义

#### ASCII 模式：

每个 8-bit 数据由两个 ASCII 字符组成。例如：一个 1-byte 数据  $64_H$  (十六进制表示法)。以 ASCII 码“64”表示，包含了‘6’的 ASCII 码 ( $36_H$ ) 和‘4’的 ASCII 码 ( $34_H$ )。数字 0 至 9、字母 A 至 F 的 ASCII 码，如下表：

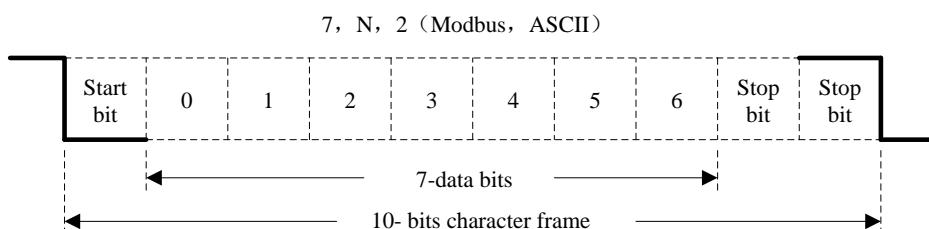
字符符号	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’
对应 ASCII 码	$30_H$	$31_H$	$32_H$	$33_H$	$34_H$	$35_H$	$36_H$	$37_H$
字符符号	‘8’	‘9’	‘A’	‘B’	‘C’	‘D’	‘E’	‘F’
对应 ASCII 码	$38_H$	$39_H$	$41_H$	$42_H$	$43_H$	$44_H$	$45_H$	$46_H$

#### RTU 模式：

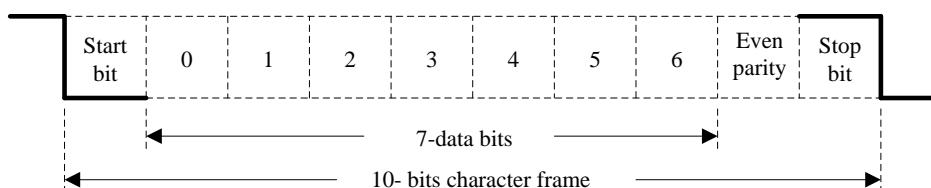
每个 8-bit 数据由两个 4-bit 的十六进制数据组成。例如：十进制 100 用 1-byte 的 RTU 数据表示为  $64_H$ 。

### ■ 字符结构

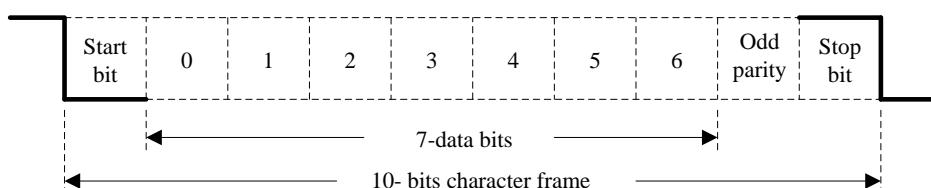
#### 10bit 字符格式（用于 7-bit 数据）



7, E, 1 (Modbus, ASCII)

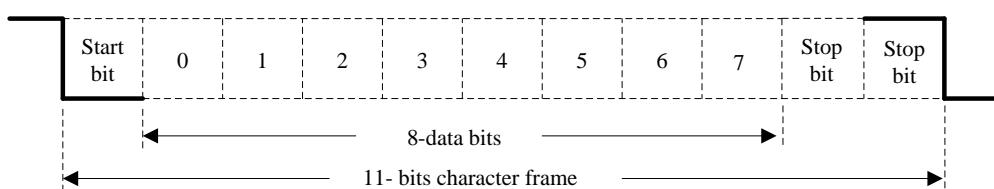


7, O, 1 (Modbus, ASCII)

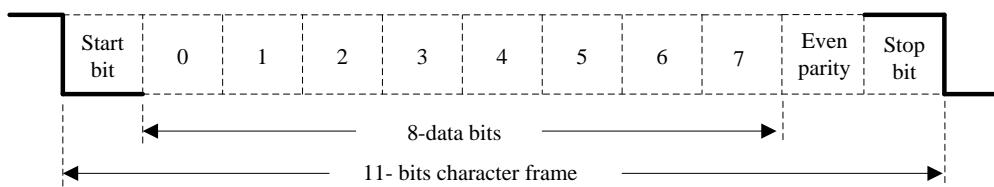


### 11bit 字符格式 (用于 8-bit 数据)

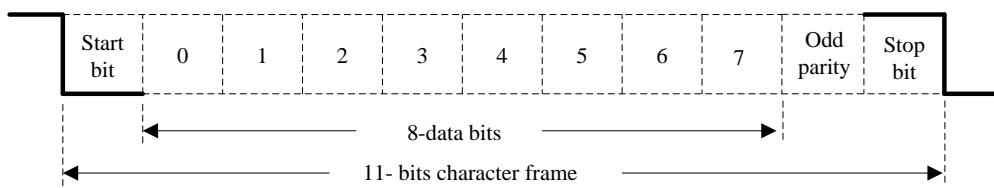
8, N, 2 (Modbus, ASCII / RTU)



8, E, 1 (Modbus, ASCII / RTU)



8, O, 1 (Modbus, ASCII / RTU)



## ■ 通讯数据结构

通讯数据结构：

**ASCII 模式：**

STX	起始字符 ‘:’ => (3A <sub>H</sub> )
ADR	通讯地址=>1-byte 包含了 2 个 ASCII 码
CMD	指令码=>1-byte 包含了 2 个 ASCII 码
DATA(n-1)	数据内容=>n-word=2n-byte 包含了 4n 个 ASCII 码， n 不大于 12
.....	
DATA(0)	
LRC	校验码=>1-byte 包含了 2 个 ASCII 码

End 1	结束码 1=> (0D <sub>H</sub> ) (CR)
End 0	结束码 0=> (0A <sub>H</sub> ) (LF)

**RTU 模式:**

STX	至少 4 个字节传输时间的静止时段
ADR	通讯地址=>1-byte
CMD	指令码=>1-byte
DATA(n-1)	数据内容=>n-word=2n-byte, n 不大于 12
.....	
DATA(0)	数据内容=>n-word=2n-byte, n 不大于 12
CRC	
End 1	至少 4 个字节传输时间的静止时段

通讯协议的数据格式说明如下：

**STX (通讯起始)**

ASCII 模式：‘:’ 字符。

RTU 模式：超过 4 个字节的通讯时间（根据通讯速度不同而自动改变）的静止时间。

**ADR (通讯地址)**

合法的通讯地址范围为 1 到 254 之间。

例如对地址为 32 (十六进制为 20) 的伺服进行通讯：

ASCII 模式：ADR= ‘2’， ‘0’ => ‘2’ =32<sub>H</sub>， ‘0’ =30<sub>H</sub>

RTU 模式：ADR=20<sub>H</sub>

**CMD (命令指令) 及 DATA (数据)**

数据的格式根据命令码而定。常用的命令码如下：

命令码：03<sub>H</sub>，读取 N 个字 (word)，N 最大为 20。

例如：从地址为 01<sub>H</sub> 的伺服读取从起始地址 0200<sub>H</sub> 开始的 2 个字。

**ASCII 模式:**

指令信息：

STX	‘:’
ADR	‘0’
	‘1’
CMD	‘0’
	‘3’
起始数据位置	‘0’
	‘2’
	‘0’
	‘0’
数据数目	‘0’
	‘0’
	‘0’
LRC Check	‘F’
	‘8’
End 1	(0DH)(CR)
End 0	(0AH)(LF)

回应信息：

STX	‘:’
ADR	‘0’
	‘1’
CMD	‘0’
	‘3’
数据数 (以 byte 计算)	‘0’
	‘4’
起始数据地址 0200H 的内容	‘0’
	‘0’
	‘B’
	‘1’
第二笔数据地址 0201H 的内容	‘1’
	‘F’
	‘4’
LRC Check	‘0’
	‘E’
	‘8’
End 1	(0DH)(CR)
End 0	(0AH)(LF)

**RTU 模式:**

指令信息:

ADR	01H
CMD	03H
起始数据位置	02H (高字节) 00H (低字节)
数据数 (以 word 计算)	00H 02H
CRC Check Low	C5H (低字节)
CRC Check High	B3H (高字节)

回应信息:

ADR	01H
CMD	03H
数据数 (以 byte 计算)	04H
起始数据地址	00H (高字节) 0200H 的内容
第二笔数据地址	1FH (高字节) 0201H 的内容
CRC Check Low	40H (低字节)
CRC Check High	A3H (低字节)
CRC Check High	D4H (高字节)

指令码: 06H, 写入 1 个字 (word)

例如: 将 100 (0064H) 写入到局号 01H 伺服的地址 0200H。

**ASCII 模式:**

指令信息:

STX	:
ADR	'0'
	'1'
CMD	'0'
	'6'
	'0'
起始数据地址	'2'
	'0'
	'0'
	'0'
	'0'
	'0'
数据内容	'0'
	'0'
	'6'
	'4'
LRC Check	'9'
	'3'
End 1	(0DH)(CR)
End 0	(0AH)(LF)

回应信息:

STX	:
ADR	'0'
	'1'
CMD	'0'
	'6'
	'0'
起始数据地址	'2'
	'0'
	'0'
	'0'
	'0'
数据内容	'0'
	'0'
	'6'
	'4'
LRC Check	'9'
	'3'
End 1	(0DH)(CR)
End 0	(0AH)(LF)

**RTU 模式:**

指令信息:

ADR	01H
CMD	06H
起始数据地址	02H (高字节) 00H (低字节)
数据内容	00H (高字节) 64H (低字节)
CRC Check Low	89H (低字节)
CRC Check High	99H (高字节)

回应信息:

ADR	01H
CMD	06H
起始数据地址	02H (高字节) 00H (低字节)
数据内容	00H (高字节) 64H (低字节)
CRC Check Low	89H (低字节)
CRC Check High	99H (高字节)

LRC (ASCII 模式) 和 CRC (RTU 模式) 值误值的计算:

**ASCII 模式的 LRC 计算:**

ASCII 模式采用 LRC (Longitudinal Redundancy Check) 值误值。LRC 值误值是从 ADR 至最后一笔资料内容之和, 得到之结果以 256 为单位, 去除超出的部分 (例如加总后得到的结果为十六进制的 128H, 则只取 28H), 然后计算其补数, 最后得到的结果即为 LRC 值误值。

例如: 从局号 01H 伺服驱动器的 0201 地址读取 1 个字 (word)。

STX	'.'
ADR	'0'
	'1'
CMD	'0'
	'3'
起始数据地址	'0'
	'2'
	'0'
	'1'
数据数	'0'
	'0'
	'0'
	'1'
LRC Check	'F'
	'8'
End 1	(0DH)(CR)
End 0	(0AH)(LF)

从 ADR 的数据加至最后一笔数据:

$01_H + 03_H + 02_H + 01_H + 00_H + 01_H = 08_H$ , 对  $08_H$  取 2 的补数为  $F8_H$ , 所以 LRC 为 'F', '8'。

#### RTU 模式的 CRC 计算:

RTU 模式采用 CRC (Cyclical Redundancy Check) 值误值。

CRC 值误值计算步骤如下:

- 步骤一: 载入一个内容为  $FFFF_H$  的 16-bit 寄存器, 称之为 “CRC” 寄存器。
- 步骤二: 将指令讯息的第一个位 (bit0) 与 16-bit CRC 寄存器的低位位 (LSB) 进行 XOR 运算, 并将结果存回 CRC 寄存器;
- 步骤三: 检查 CRC 寄存器的最低位 (LSB), 若此位为 0, 则 CRC 寄存器值右移一位; 若此位为 1, 则 CRC 寄存器值右移一位后, 再与  $A001_H$  进行 XOR 运算;
- 步骤四: 回到步骤三, 直到步骤三已被执行过 8 次, 才进行到步骤五;
- 步骤五: 对指令讯息的下一位重复步骤二到步骤四, 直到所有位都被这样处理过, 此时 CRC 寄存器的内容即是 CRC 值误值。

**说明:** 计算出 CRC 值误值之后, 在指令讯息中, 须先填上 CRC 的低位, 再填上 CRC 的高位, 请参考以下例子。

例如: 从局号为  $01_H$  伺服的  $0101_H$  地址读取 2 个字 (word)。从 ADR 至资料数的最后一位所计算出的 CRC 寄存器的最后内容为  $3794_H$ , 则其指令讯息如下所示, 须注意的是  $94_H$  在  $37_H$  的前面传送。

ADR	$01_H$
CMD	$03_H$
起始资料地址	$01_H$ (地址高位)
	$01_H$ (地址低位)
资料数 (以 word 计算)	$00_H$ (高位)
	$02_H$ (低位)
CRC 校验低位	$94_H$ (校验低位)
CRC 校验高位	$37_H$ (校验高位)

**End1、End0 (通讯侦完成)**

**ASCII 模式:**

以 (0D<sub>H</sub>) 即字符 ‘\r’『carriage return』及 (0A<sub>H</sub>) 即字符为 ‘\n’『new line』，代表通讯结束。

**RTU 模式:**

超过当前通讯速率下的 4 个字节通讯时间的静止时段表示通讯结束。

**范例:**

下面以 C 语言产生 CRC 值。此函数需要两个参数：

```
unsigned char * data;
unsigned char length;
此函数将回传 unsigned integer 型态的 CRC 值。
unsigned int crc_chk(unsigned char * data,unsigned char length){
    int i,j;
    unsigned int crc_reg = 0xFFFF;
    while(length- -){
        crc_reg ^=*data++;
        for(j=0;j<8;j++){
            if(crc_reg & 0x01){
                crc_reg=( crc_reg >>1)^0xA001;
            }
            Else
            {
                crc_reg=crc_reg >>1;
            }
        }
    }
    return crc_reg;
}
```

**■ 通讯出错**

在通讯过程中，有可能会发生错误，常见错误源如下：

- 读写参数时，数据地址不对；
- 写参数时，数据超过此参数的最大值或者小于此参数的最小值；
- 通讯受到干扰，数据传输错误或者校验码错误。

当出现上述前两种通讯错误时，伺服驱动器运行不受影响，同时伺服驱动器会反馈一错误帧。当出现第三种错误时，传输数据将会被认为无效丢弃，不返回帧。

错误帧格式如下：

**上位机数据帧：**

start	从站地址	命令	数据地址、资料等	校验
		命令		

**伺服驱动器反馈错误帧：**

start	从站地址	响应代码	错误代码	校验
		命令 + 80 <sub>H</sub>		

- 其中，错误帧响应代码=命令+80<sub>H</sub>；  
 错误代码=00<sub>H</sub>：通讯正常；  
 =01<sub>H</sub>：伺服驱动器不能识别所请求的功能；  
 =02<sub>H</sub>：请求中给出的数据地址在伺服驱动器中不存在；  
 =03<sub>H</sub>：请求中给出的数据在伺服驱动器中不允许（超过参数的最大或最小值）；  
 =04<sub>H</sub>：伺服驱动器已经开始执行请求，但不能完成该请求；

例如：伺服驱动器轴号为03<sub>H</sub>，对参数Pn100写入数据06<sub>H</sub>，由于参数Pn100参数范围0~6，所以写入数据将不被允许，伺服驱动器将返回一个错误帧，错误代码为03<sub>H</sub>（超过参数的最大或最小值），结构如下：

**上位机数据帧：**

start	从站地址	命令	数据地址、资料等	校验
	03 <sub>H</sub>	06 <sub>H</sub>	0002 <sub>H</sub> 0006 <sub>H</sub>	

**伺服驱动器反馈错误帧：**

start	从站地址	响应代码	错误代码	校验
	03 <sub>H</sub>	86 <sub>H</sub>	03 <sub>H</sub>	

另外，如果上位机发送的数据帧中的从站地址为00<sub>H</sub>，表示此帧数据是广播数据，伺服驱动器将不返回帧。

## 6.4 MODBUS 通讯地址

通讯数据地址 十六进制	含义	说明	操作
0000 <sub>H</sub> ~ 03FF <sub>H</sub>	参数区	对应参数表中的参数	可读可写
0400 <sub>H</sub> ~ 0409 <sub>H</sub>	报警信息存储区	10个历史报警	只读
0410 <sub>H</sub>	速度指令零点偏移量		只读
0411 <sub>H</sub>	转矩指令零点偏移量		只读
0412 <sub>H</sub>	Iu 零点偏移量		只读
0413 <sub>H</sub>	Ipv 零点偏移量		只读
0420 <sub>H</sub> ~ 0437 <sub>H</sub>	监控数据		只读
0420 <sub>H</sub>	电机转速	单位：1r/min	只读
0422 <sub>H</sub>	旋转角（电气角）	单位：1deg	只读
0424 <sub>H</sub>	输入指令脉冲速度	单位：1kHz	只读
0426 <sub>H</sub>	母线电压	单位：1V	只读
0428 <sub>H</sub>	模拟输入速度指令值	单位：1 r/min	只读
042A <sub>H</sub>	模拟输入转矩指令百分比	单位：1%	只读
042C <sub>H</sub>	内部转矩指令百分比	单位：1%或0.1A	只读
042E <sub>H</sub>	输入信号监视	——	只读
0430 <sub>H</sub>	输出信号监视	——	只读

通讯数据地址 十六进制	含义	说明	操作
<b>0432<sub>h</sub></b>	编码器信号监视	——	只读
<b>0434<sub>h</sub></b>	输入指令脉冲计数器	单位： 1 指令脉冲	只读
<b>0436<sub>h</sub></b>	反馈脉冲计数器	单位： 1 指令脉冲	只读
<b>0438<sub>h</sub></b>	位置偏移量计数器	单位： 1 指令脉冲	只读
<b>043A<sub>h</sub></b>	累计负载	单位： 1%	只读
<b>043C<sub>h</sub></b>	转动惯量百分比	单位： 1%	只读
<b>043E<sub>h</sub></b>	编码器实际角度	单位： 1 指令脉冲	只读
<b>0440<sub>h</sub></b>	编码器多圈位置	单位： 1 圈	只读
<hr/>			
<b>044A<sub>h</sub></b>	当前报警		只读
<b>0451<sub>h</sub></b>	通讯 IO 信号 *1	掉电不保存	可读可写
<b>0452<sub>h</sub></b>	通讯输出口取反	掉电不保存	可读可写
<b>0457<sub>h</sub></b>	伺服操作状态 *2		只读
<b>045E<sub>h</sub></b>	软件版本号		只读
<b>045F<sub>h</sub></b>	FPGA 版本号		只读
<hr/>			
<b>0520<sub>h</sub></b>	清除历史报警	1： 清除历史报警	可读可写
<b>0521<sub>h</sub></b>	清除当前报警	1： 清除当前报警	可读可写
<b>0522<sub>h</sub></b>	清除总线编码器报警	1： 清除总线编码器报警	可读可写
<b>0523<sub>h</sub></b>	清除总线编码器多圈数据	1： 清除总线编码器多圈数据	可读可写
<hr/>			
<b>0528<sub>h</sub></b>	速度 JOG (转速为 P□304 设置)	BIT15:1 JOG 伺服使能 BIT01:1 JOG- (JOG 正转) BIT00:1 JOG+ (JOG 反转)	可读可写
<b>0529<sub>h</sub></b>	位置 JOG (转速为 P□304 设置)	BIT15:1 进入位置点动模式 BIT01:1 JOG- BIT00:1 JOG+	可读可写
<hr/>			
<b>0540<sub>h</sub></b>	恢复出厂值	1： 恢复出厂值	可写
<b>0541<sub>h</sub></b>	复位	1： 复位	可写
<hr/>			
<b>05F0<sub>h</sub></b>	当前正在运行数据组号		只读
<b>05F1<sub>h</sub></b>	即将运行数据组号		只读
<b>05F2<sub>h</sub></b>	实际位置低 16 位	位置接点电子齿轮后的位置	只读
<b>05F3<sub>h</sub></b>	实际位置高 16 位		只读
<b>05F4<sub>h</sub></b>	位置节点方式	0: 任务 1: 外部	只读
<b>05F5<sub>h</sub></b>	加速度	10rpm/s/s	可读可写
<b>05F6<sub>h</sub></b>	减速度	10rpm/s/s	可读可写
<b>05F7<sub>h</sub></b>	紧急减速速度	10rpm/s/s	可读可写
<b>05F8<sub>h</sub></b>	位置接点电子齿轮分子		可读可写
<b>05F9<sub>h</sub></b>	位置接点电子齿轮分母		可读可写
<b>05FA<sub>h</sub></b>	找参考点方式		可读可写

通讯数据地址 十六进制	含义	说明	操作
<b>05FB<sub>h</sub></b>	找参考点开关速度	0~6000 rpm	可读可写
<b>05FC<sub>h</sub></b>	离开参考点开关速度	0~6000 rpm	可读可写
<b>05FD<sub>h</sub></b>	示教位置低位		可读可写
<b>05FE<sub>h</sub></b>	示教位置高位		可读可写
<hr/>			
数据组 0 参数：			
<b>0600<sub>h</sub></b>	目标位置低位		可读可写
<b>0601<sub>h</sub></b>	目标位置高位		可读可写
<b>0602<sub>h</sub></b>	目标速度	rpm	可读可写
<b>0603<sub>h</sub></b>	换步属性 * <sup>3</sup>		可读可写
<b>0604<sub>h</sub></b>	换步条件 1 数值		可读可写
<b>0605<sub>h</sub></b>	换步条件 2 数值		可读可写
<b>0606<sub>h</sub></b>	后续数据组号		可读可写
<b>0607<sub>h</sub></b>	数据组类型	0: NULL; 1: 绝对; 2: 相对	可读可写
<hr/>			
数据组 1 参数：			
<b>0608<sub>h</sub></b>	目标位置低位		可读可写
<b>0609<sub>h</sub></b>	目标位置高位		可读可写
<b>060A<sub>h</sub></b>	目标速度	rpm	可读可写
<b>060B<sub>h</sub></b>	换步条件属性		可读可写
<b>060C<sub>h</sub></b>	换步条件 1 数值		可读可写
<b>060D<sub>h</sub></b>	换步条件 2 数值		可读可写
<b>060E<sub>h</sub></b>	后续数据组号		可读可写
<b>060F<sub>h</sub></b>	数据组类型	0: NULL; 1: 绝对; 2: 相对	可读可写
<hr/>			
数据组 2 参数：			
<b>0610<sub>h</sub></b>	目标位置低位		可读可写
<b>0611<sub>h</sub></b>	目标位置高位		可读可写
<b>0612<sub>h</sub></b>	目标速度	rpm	可读可写
<b>0613<sub>h</sub></b>	换步条件属性		可读可写
<b>0614<sub>h</sub></b>	换步条件 1 数值		可读可写
<b>0615<sub>h</sub></b>	换步条件 2 数值		可读可写
<b>0616<sub>h</sub></b>	后续数据组号		可读可写
<b>0617<sub>h</sub></b>	数据组类型	0: NULL; 1: 绝对; 2: 相对	可读可写
<hr/>			
数据组 3 参数：			
<b>0618<sub>h</sub></b>	目标位置低位		可读可写
<b>0619<sub>h</sub></b>	目标位置高位		可读可写
<b>061A<sub>h</sub></b>	目标速度	rpm	可读可写
<b>061B<sub>h</sub></b>	换步条件属性		可读可写
<b>061C<sub>h</sub></b>	换步条件 1 数值		可读可写
<b>061D<sub>h</sub></b>	换步条件 2 数值		可读可写

## 地址说明：

### \*1、通讯 IO 输入（**0451<sub>h</sub>**）

输入信号可通过 MODBUS 通讯的通讯 IO 输入（**0451<sub>h</sub>**）寄存器给定，该寄存器定义如下：

bit15 /START-HOME	bit14 /POS-STEP	bit13 /POS-START	bit12 /POS-REF	bit11 /POS2	bit10 /POS1	bit9 /POS0	bit8 /G-SEL
----------------------	--------------------	---------------------	-------------------	----------------	----------------	---------------	----------------

bit7 /N-CL	bit6 /P-CL	bit5 /CLR	bit4 /ALM-RST	bit3 N-OT	bit2 P-OT	bit1 /P-CON	bit0 /SON
---------------	---------------	--------------	------------------	--------------	--------------	----------------	--------------

仅在该信号不从 CN3 输入（信号分配参数设置成“无效”）时，该寄存器中信号输入才有效。

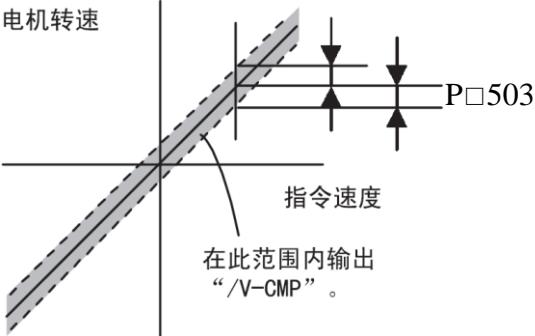
例：通过通讯 IO 输入寄存器输入/POS-START，首先需要设置 P□512.1=0，修改通讯 IO 输入（**0451<sub>h</sub>**）寄存器的 bit13 位才有效。

### \*2、伺服操作状态（**0457<sub>h</sub>**）

bit15 ALM	bit14 REF-PASS	bit13 保留	bit12-bit8 输入刀号	bit7 S-RDY	bit6 S-ST	bit5 COIN	bit4 AC-IN	bit3 PS-IN	bit2 TGON	bit1 N-OT	bit0 P-OT
伺服报警标志：‘1’ 表示有报警产生											位置超程：‘1’ P-OT有效
寻找参考点：‘1’ 表示已找到参考点											位置超程：‘1’ N-OT有效
保留											旋转检测：‘1’ 电机转速高于规定值
刀号对应：bit12-bit8对应tools4-tools0											位置控制：‘1’ 表示脉冲输入中速度控制：‘1’ 表示速度给定高于规定值
伺服准备好标志：‘1’ 表示准备好											
伺服使能状态：‘1’ 表示伺服使能，‘0’ 表示没有使能											
位置控制时‘1’ 表示定位完成；速度控制时‘1’ 表示电机速度到达给定速度											电源输入标志：‘1’ 表示驱动器的R、T端子有电源输入

### \*3、换步条件属性



P□503	同速检测信号宽度			
	速度			
	设定范围	设定单位	出厂设定	电源重起
	0 ~ 100	1r/min	10	不需要
如果电机转速与指令速度之差低于 P□503 的设定值，则输出 “/V-CMP” 信号。 <b>■例</b> P□503=100、指令速度为 2000r/min 时，如果电机转速处在 1900 ~ 2100r/min 之间，则将 “/V-CMP” 置为 ON。		 电机转速 指令速度 在此范围内输出 “/V-CMP”。		
<b>■补充</b> “/VCMP” 信号是速度控制时的输出信号。如果是位置控制，则功能自动地变为 “/COIN”，如果是扭矩控制，则自动地变为 “OFF(H 电平)”。				