ADDR：设备的站地址，默认是1，可以使用remoteIO软件修改，一字节

CRC： CRC16校验码，两字节

一、读取DO状态

发送：ADDR 01 00 10 00 08 CRC

应答：ADDR 01 01 status CRC

其中，status是一个字节，将其展开成二进制，由低至高表示8路DO的状态，1表示闭合，0表示断开。例如，status为0F，那么表示DO1-4处于闭合状态；DO5-8处于断开状态。

二、控制DO闭合/断开

发送：ADDR 05 register status CRC

应答：ADDR 05 register status CRC

其中，register为希望控制的DO对应的寄存器地址，两个字节，见下

DO1 → 00 10

DO2 → 00 11

DO3 → 00 12

DO4 → 00 13

DO5 → 00 14

DO6 → 00 15

DO7 → 00 16

DO8 → 00 17

而status为希望控制这个DO达到哪个状态，两个字节，闭合DO使用FF 00，断开DO使用00 00。

三、读取DI状态

发送：ADDR 02 00 00 00 08 CRC

应答：ADDR 02 01 status CRC

其中，status是一个字节，将其展开成二进制，由低至高表示8路DI的状态，1表示导通，0表示开路。例如，status为F0，那么表示DI1-4处于开路状态；DI5-8处于导通状态。

四、DI主动上报

当使用remoteIO软件启用了DI主动上报功能后，DI每次发生变化（由开路变为导通或者由导通变为开路）都会主动上报一个特定报文，这样就不用上位机再去查询了。

上报：UPADDR 05 register status CRC

注意这里是UPADDR，也就是设置DI主动上报时的上报地址，不是设备自身的地址。

而register表示发生状态变化的是第几路DI，两个字节，见下

DI1 → 00 10

DI2 → 00 11

DI3 → 00 12

DI4 → 00 13

DI5 → 00 14

DI6 → 00 15

DI7 → 00 16

DI8 → 00 17

还有status表示这个DI最新的状态，两个字节，FF 00表示它处于导通状态了，00 00表示它处于开路状态了。

五、读取AI数值

发送：ADDR 04 start quantity CRC

应答：ADDR 04 length data CRC

其中，start表示从第几路AI开始读取，两个字节，对应关系见下

AI1 → 00 00

AI2 → 00 01

AI3 → 00 02

AI4 → 00 03

AI5 → 00 04

AI6 → 00 05

AI7 → 00 06

AI8 → 00 07

quantity表示需要从start开始往后读多少路AI，两个字节，例：start=0，quantity=8表示读取全部8路AI的值；start=4，quantity=4表示读取后4路（AI5-8）的值。

length表示后面data的字节数，而data的字节数是不确定的，取决于读取了多少路AI，每一路有两个字节，所以length=quantity×2，注意后面CRC还有两个字节，不计算在length里。

data依次表示所读取的每一路AI的值，每一路两个字节，将这两个字节转为十进制，是一个处于[0,1024]区间的数，线性对应这一路AI的量程，目前有三种量程，计算公式见下

（value为处于[0,1024]区间的裸数值，计算结果为接入这一路AI的实际电流/电压值）

0-5V：value÷1024×5，单位V

0-10V：value÷1024×10，单位V

0-25mA：value÷1024×25，单位mA

注意工业说法一般是4-20mA，但是计算的时候要按照0-25mA来计算。

附：Modbus TCP报文

前文所述均为Modbus RTU报文，Modbus TCP报文要做一些修改。

Modbus TCP的指令结构如下

|  |  |
| --- | --- |
| Modbus TCP前缀 | Modbus RTU（去掉CRC） |

Modbus TCP前缀 为6个字节：00 00 00 00 00 LEN，最后一个字节LEN表示后续的Modbus RTU报文（去掉CRC）的字节数。

举例，闭合DO1 的Modbus TCP报文为00 00 00 00 00 06 01 05 00 10 ff 00，其他报文也是一样的转换方法。