

# IO 模块 Modbus-TCP 协议说明

## 使用说明书 (V2.0)

### 1、概述

此说明适用于 YX、YA、YT 等全系列 IO 模块。此文件无特别说明，都会以 YX3232 此型号作为参考样板说明。

### 2、如何在产品中切换 Modbus-TCP 与 Modbus-RTU 两种协议？

A. 只需要用 06 功能码修改 0x1FA 寄存器就可改变串口的通信协议。

B. 0x1FA 寄存器每 4 位对应一个通讯口设置，具体每个通讯口的通信协议设置，请参阅每个产品的说明书，现以 YX3232 为列，列表如下：

0x1FA 寄存器位	数据含义代码（位定义）	对应产品通讯接口序号	对应产品通信接口
Bit3:Bit0	0000: Modbus-RTU 协议(默认) 0001: Modbus-TCP 协议	第一通讯口	RS485 或网口 1
Bit7:Bit4	0000: Modbus-RTU 协议(默认) 0001: Modbus-TCP 协议	第二通讯口	RS232（非隔离时） 或网口 2
Bit11:Bit8	保留	第三通讯口	没有用到
Bit15:Bit12	保留	第四通讯口	没有用到

C. 注意：因为所有通讯口的协议格式存储在同一寄存器（0x1FA）的不同位上(16 位 2 个字节)，而我们用 06 或 16 功能码修改时，是按字节修改的，所以在修改一个通讯口的协议时，要把其它通讯口的原协议代码保留填入，否则会同步修改。

D. 举例，更改其中一个通讯口的通信协议为 Modbus-TCP：（通讯口出厂默认方式为 Modbus-RTU）

➢ 当需要把第二通讯口（RS232 或网口 2）由当前通信协议 Modbus-RTU 更改为 Modbus-TCP 协议，第一通讯口（RS485 或网 1）通信协议不变保留为 Modbus-RTU 时，则需发送命令如下：

命令：01 06 01 FA 00 10...A9 CB(返回相同指令即修改成功)，解析如下表：

设备地址	功能码	改写的寄存器		改写的的数据		CRC校验码	
		高8位	低8位	高8位 (Bit15:Bit8)	低8位 (Bit7:Bit0)	高8位	低8位
01	06	01	FA	00 ↙   ↘ 第4通讯口 第3通讯口 格式        格式	10 ↙   ↘ 第2通讯口 第1通讯口 格式        格式	A9	CB

➢ 当需要把第一通讯口（RS485 或网 1）由当前通信协议 Modbus-RTU 更改为 Modbus-TCP 协议，第二通讯口（RS232 或网口 2）通信协议不变保留为 Modbus-RTU 时，，则需发送命令如下：

命令：01 06 01 FA 00 01 69 C7(返回相同指令即修改成功)；解析如下表：

设备地址	功能码	改写的寄存器		改写的的数据				CRC校验码	
		高8位	低8位	高8位		低8位		高8位	低8位
01	06	01	FA	00 ↙ ↘ 第4通讯口 第3通讯口 格式 格式		01 ↙ ↘ 第2通讯口 第1通讯口 格式 格式		69	C7

E. 举例，由 Modbus-TCP 协议更改为 Modbus-RTU:

第一通讯口当前通讯协议为 Modbus-TCP，第二通讯口为 Modbus-RTU 协议，需要第一通讯口的协议更改为 Modbus-RTU，第二通讯口保持不变，**则如果从第一通讯口更改协议格式，则需发送命令如下**（如果要**从第二通讯口更改，因为第二通讯口当前为 Modbus-RTU，则需要用前面 D 小节的方式去修改**）：

**命令：00 00 00 00 00 06 01 06 01 FA 00 00(返回相同指令即修改成功)；解析如下表：**

事务标示符		协议标示符		数据长度		设备地址	功能码	改写的寄存器		改写的的数据			
高8位	低8位	高8位	低8位	高8位	低8位			高8位	低8位	高8位		低8位	
00	00	00	00	00	06	01	06	01	FA	00 ↙ ↘ 第4通讯口 第3通讯口 格式 格式		00 ↙ ↘ 第2通讯口 第1通讯口 格式 格式	

### 3、YX3232 寄存器说明

#### 3.1 开关量输入状态寄存器

寄存器地址：0000H 至 001FH--存 32 路开关量输入状态，只能用 02 功能码或 04 功能码读。

#### 3.2 开关量输出状态寄存器

寄存器地址：0000H 至 001FH--存 32 路继电器输出状态，只能用 01 功能码读或 05 与 15 功能码写。

#### 3.3 配置字寄存器

此类寄存器只能用 03 功能码读或 06 与 16 功能码写，见表如下：

表 (2)

寄存器地址(Hex)	保持寄存器内容	寄存器个数	寄存器状态	数据范围
0000H--001FH	继电器输出方式	32	读/写	0000--继电器常闭常开输出 0001--继电器 0.5HZ 闪动 0002--继电器输出 1 秒脉冲 0003--互锁功能：每次有输入信号，对应输出反向 0004--锁存功能：开关量有输入，对应输出就一直闭合，只有发命令才能复位继电器 0005--继电器闭合后，定时器按 10ms 计时，时间到后继电器断开； 0006--继电器闭合后，定时器按 1 分钟计时，时间到后继电器断

				开。
0050H	地址	1	读/写	地址(0-254)(默认 01) 如果板端拨码开关第 6 位为 ON (1) 状态, 则产品用此寄存器地址; 如果为 0 状态, 则由拨码开关第 5 至 1 位(对应二进制 bit4 至 bit0 位) 决定地址。
0051H	第 1 串口波特率	1	读/写	0000 设置波特率-115200bps 0001 设置波特率-9600bps(默认) 0002 设置波特率-19200bps 0003 设置波特率-38000bps 0004 设置波特率-2400bps 0005 设置波特率-4800bps 0006 设置波特率-9600bps 0007 设置波特率-19200bps 0008 设置波特率-38400bps 0009 设置波特率-57600bps 000A 设置波特率-115200bps
0052H	第 1 串口寄偶校验	1	读/写	0000 无校验, 1 个停止位(默认) 0001 奇校验, 1 个停止位 0002 偶校验, 1 个停止位 0003 无校验, 2 个停止位 0004 奇校验, 2 个停止位 0005 偶校验, 2 个停止位
0055H	模块名称--高	1	读/写	默认:5958H (XY 的 ASCII 码)
0056H	模块名称--中	1	读/写	默认:3332H (32 的 ASCII 码)
0057H	模块名称--低	1	读/写	默认:3332H (32 的 ASCII 码)
0058H	软件版本	1	读	3032: 02 的 ASCII 码
0059H	软件子版本	1	读	3031: 01 的 ASCII 码
005AH	第 2 串口波特率			同 0051H
005BH	第 2 串口寄偶校验			同 0052H
0060H--007FH	继电器定时器值	1	读/写	对应 1~32 路继电器, 当继电器工作方式在 05 方式时, 按 10ms 倒计时; 当继电器工作方式在 06 方式时, 按 1 分钟倒计时。 比如: 如果要第 3 路继电器闭合 10 分钟后断开, 则可设 0002H 寄存器为 06, 设 0062H 寄存器为 10; 则在继电器闭合 10 分钟

				后断开。
0x1FAH	通讯协议定义	1	读/写	详见第 2 小节

## 4、Modbus-TCP 通讯协议

如下所有命令都是以硬件地址为 01 来举例说明；

### 4.1 读继电器开关量输出状态命令（01 功能码，按位读）

#### 主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	01	功能码	1
6	00 03	起始开关量序号，高 8 位在前，低 8 位在后 参照 3.2	2
7	00 20	读取开关量个数，高 8 位在前，低 8 位在后 （此列读取 32 个开关量数据）	2

#### 从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	01	功能码	1
6	04	数据区返回的字节个数(读取寄存器个数/8)	1
7	80 00 00 02	转换成二进制数为“10000000 00000000 00000000 00000010”， 从左至右分别对应 32 路继电器 Do08-Do01, Do16-Do09, Do24-Do17, Do32-Do25 的状态 （此列表示 Do08 与 Do26 闭合，其它为断开状态）	按序列 6 表示的长度

## 4.2 读开关量输入命令 (02 功能码, 按位读)

## 主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符, 固定	2
3	00 06	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随有 6 个字节的数据)	2
4	01	从设备地址, 可变 (1-255) (此列为 01 设备地址)	1
5	02	功能码	1
6	00 03	起始开关量序号, 高 8 位在前, 低 8 位在后 参照 3.1	2
7	00 20	读取开关量个数, 高 8 位在前, 低 8 位在后 (此列读取 32 个开关量数据)	2

## 从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符, 与主设备发送报文保持一致	2
3	00 07	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随有 7 个字节的数据)	2
4	01	从设备地址, 与主设备发送报文保持一致	1
5	02	功能码	1
6	04	数据区返回的字节个数(读取寄存器个数/8)	1
7	40 00 00 01	转换成二进制数为“01000000 00000000 00000000 00000001”, 从左至右分别对应 32 路继电器 Di08-Di01, Di16-Di09, Di24-Di17, Di32-Di25 的状态 (此列表示 Di07 与 Di25 开关闭合, 其它为断开状态)	按序列 6 表示的长度

#### 4.3 读保持寄存器命令（03 功能码）

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	03	功能码	1
6	00 53	数据起始寄存器地址，高 8 位在前，低 8 位在后 参照产品寄存器表（2）	2
7	00 02	读取寄存器个数，高 8 位在前，低 8 位在后 （此列读取 2 个寄存器数据）	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	03	功能码	1
6	04	数据区返回的字节个数(读取寄存器个数 x2)	1
7	40 02 03 01	每 2 个字节表示一个寄存器值，数据高位在前，低位在后 （此列表示 0053 寄存器数据为 4002H，0054 寄存器数据为 0301H）	按序列 6 表示的长度

**4.4 读开关量输入命令 (04 功能码, 字节读)**
**主设备发送报文**

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符, 固定	2
3	00 06	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随有 6 个字节的数据)	2
4	01	从设备地址, 可变 (1-255) (此列为 01 设备地址)	1
5	04	功能码	1
6	00 00	起始开关量序号, 高 8 位在前, 低 8 位在后 参照 3.1	2
7	00 03	读取开关量个数, 高 8 位在前, 低 8 位在后 (此列读取 3 个开关量数据)	2

**从设备返回正确报文**

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符, 与主设备发送报文保持一致	2
3	00 07	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随有 7 个字节的数据)	2
4	01	从设备地址, 与主设备发送报文保持一致	1
5	04	功能码	1
6	06	数据区返回的字节个数(读取开关量个数 x2)	1
7	00 01 00 00 00 00	每 2 个字节表示一个寄存器值, 数据高位在前, 低位在后 (此列表示 Di01 开关闭合, Di02 与 Di03 断开)	按序列 6 表示的长度

#### 4.5 继电器输出控制命令:

##### A. 多个继电器控制命令（15 功能码 多路同步控制继电器吸合）:

##### 主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	0F	功能码	1
6	00 03	起始开关量序号，高 8 位在前，低 8 位在后 参照 3.2	2
7	00 20	写入继电器长度，高 8 位在前，低 8 位在后 （此列写入 32 个继电器状态）	2
8	04	写入字节长度（写入继电器长度/8）	1
9	01 00 10 00	写入的数据，转换成 2 进制数为“00000001 00000000 00010000 00000000”，字节从左至右分别对应 Do08-Do01, Do16-Do09, Do24-Do17, Do32-Do25 路数字；即 Do01、Do21 闭合，其他通道断开	按序列 8 表示的字节数

##### 从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	0F	功能码	1
6	00 03	起始开关量序号，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2
7	00 20	写入继电器长度，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2

## B、单个继电器控制命令（05 功能码）：

**主设备发送报文**

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	05	功能码	1
6	00 03	开关量序号，高 8 位在前，低 8 位在后 参照 3.2	2
7	00 FF	写入数据 FF00H 时代表断路器吸合，写入 0000 数据，代表继电器继开	2

**从设备返回正确报文**

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	05	功能码	1
6	00 03	开关量序号，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2
7	00 FF	写入数据，与主设备发送的报文相同	2

#### 4.6 配置寄存器（表 2）修改命令：

##### 4.6.1 单个寄存器修改命令（06 功能码 每次只能修改一个寄存器）

##### 主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	06	功能码	1
6	00 03	寄存器地址，高 8 位在前，低 8 位在后，参照产品寄存器表（2）	2
7	00 01	寄存器数据，参照产品寄存器表（2）	2

##### 从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	06	功能码	1
6	00 03	寄存器地址，高 8 位在前，低 8 位在后，与主设备发送的报文相同	2
7	00 01	寄存器数据，与主设备发送的报文相同	2

## 4.6.2 连续修改多个寄存器命令（16 功能码）

## 主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文	2
2	00 01	表示协议标识符，固定	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随着 6 个字节的数据）	2
4	01	从设备地址，可变（1-255） （此列为 01 设备地址）	1
5	10	功能码	1
6	00 03	起始寄存器，高 8 位在前，低 8 位在后 参照产品寄存器表（2）	2
7	00 04	写入寄存器长度，高 8 位在前，低 8 位在后 （此列写入 4 个寄存器）	2
8	08	写入字节长度（写入寄存器长度 x2）	1
9	00 00 00 01 00 03 00 06	写入的数据，每 2 个字节表示一个寄存器数据，高位在前，低位在后；此列表示把 0003H 寄存器写入数据 0000H，0004H 寄存器写入数据 0001H，0005H 寄存器写入数据 0003H，0006H 寄存器写入数据 0006H	按序列 8 表示的字节数

## 从设备返回正确报文

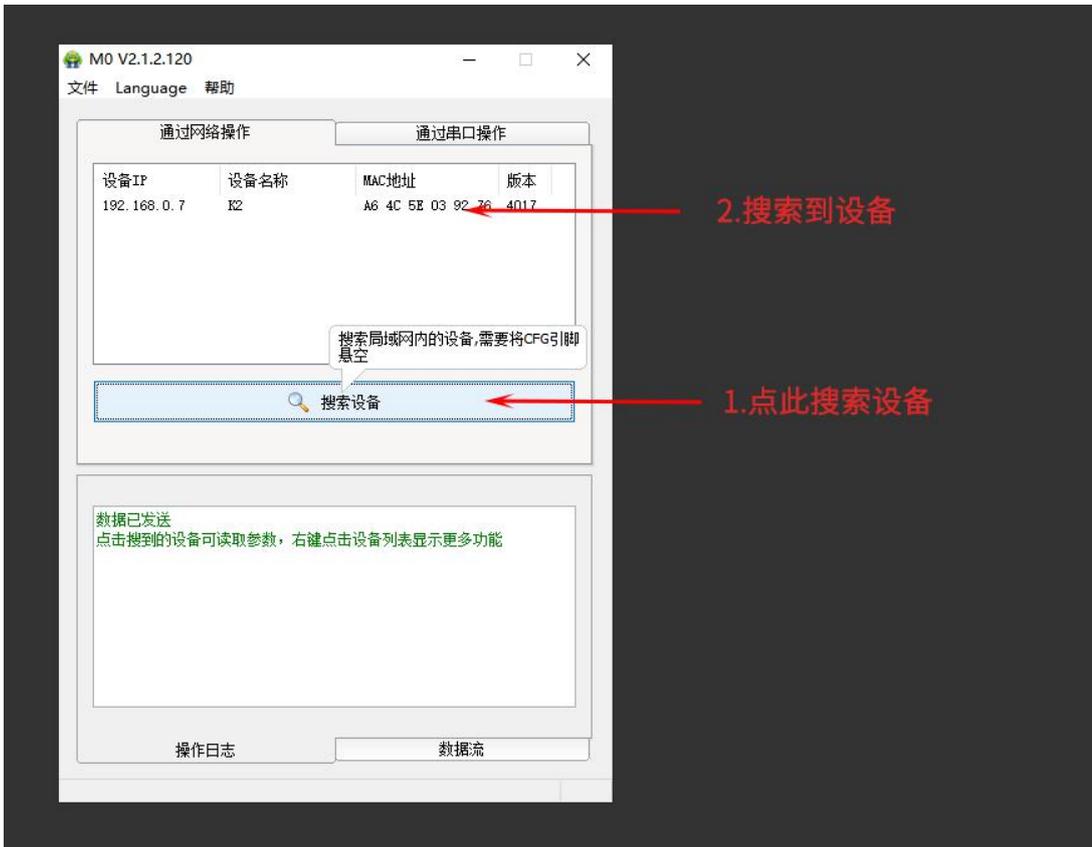
序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随着 7 个字节的数据）	2
4	01	从设备地址，与主设备发送报文保持一致	1
5	10	功能码	1
6	00 03	起始寄存器，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2
7	00 04	写入寄存器长度，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2

## 5、网口配置说明

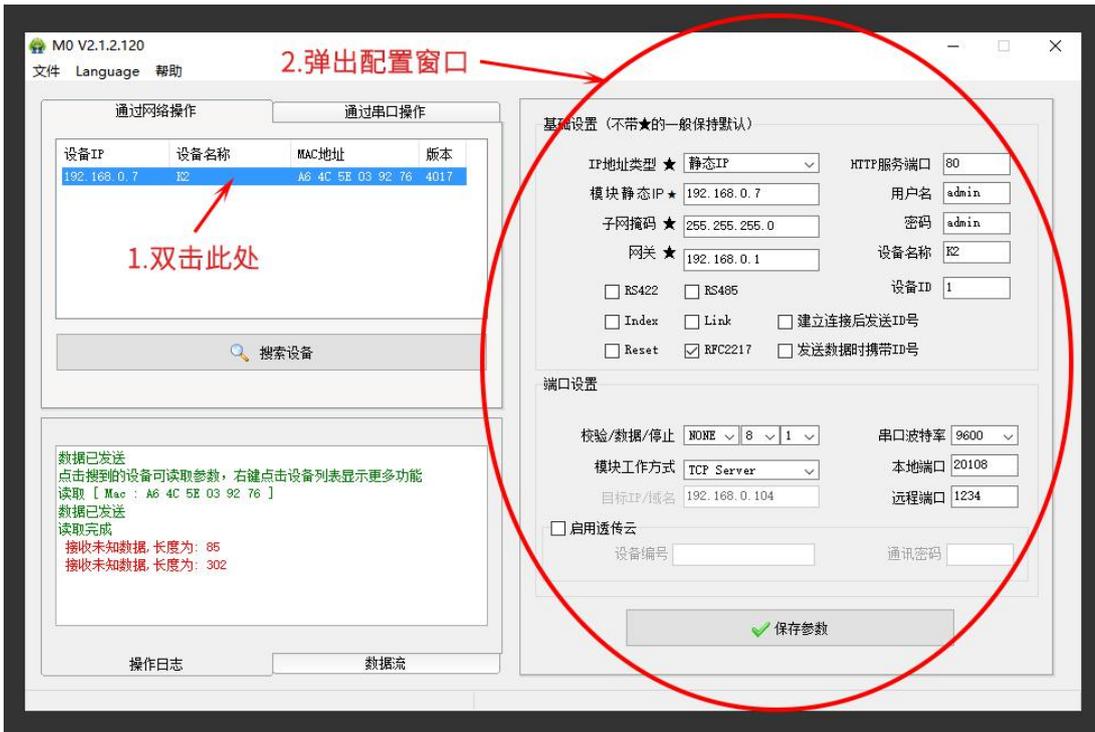
5.1 网口配置需要用到此软件



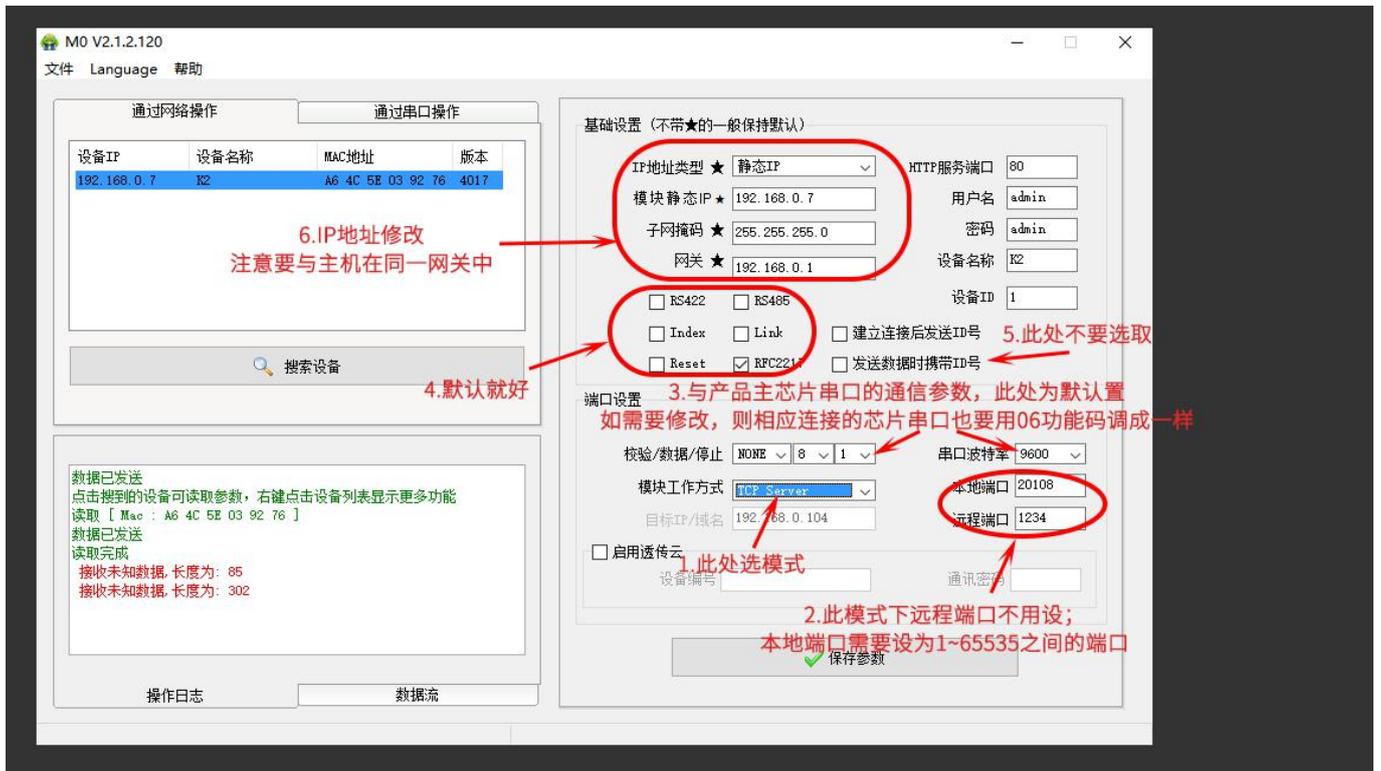
5.2 产品上电，并连接好网线，然后就可以搜索产品了：



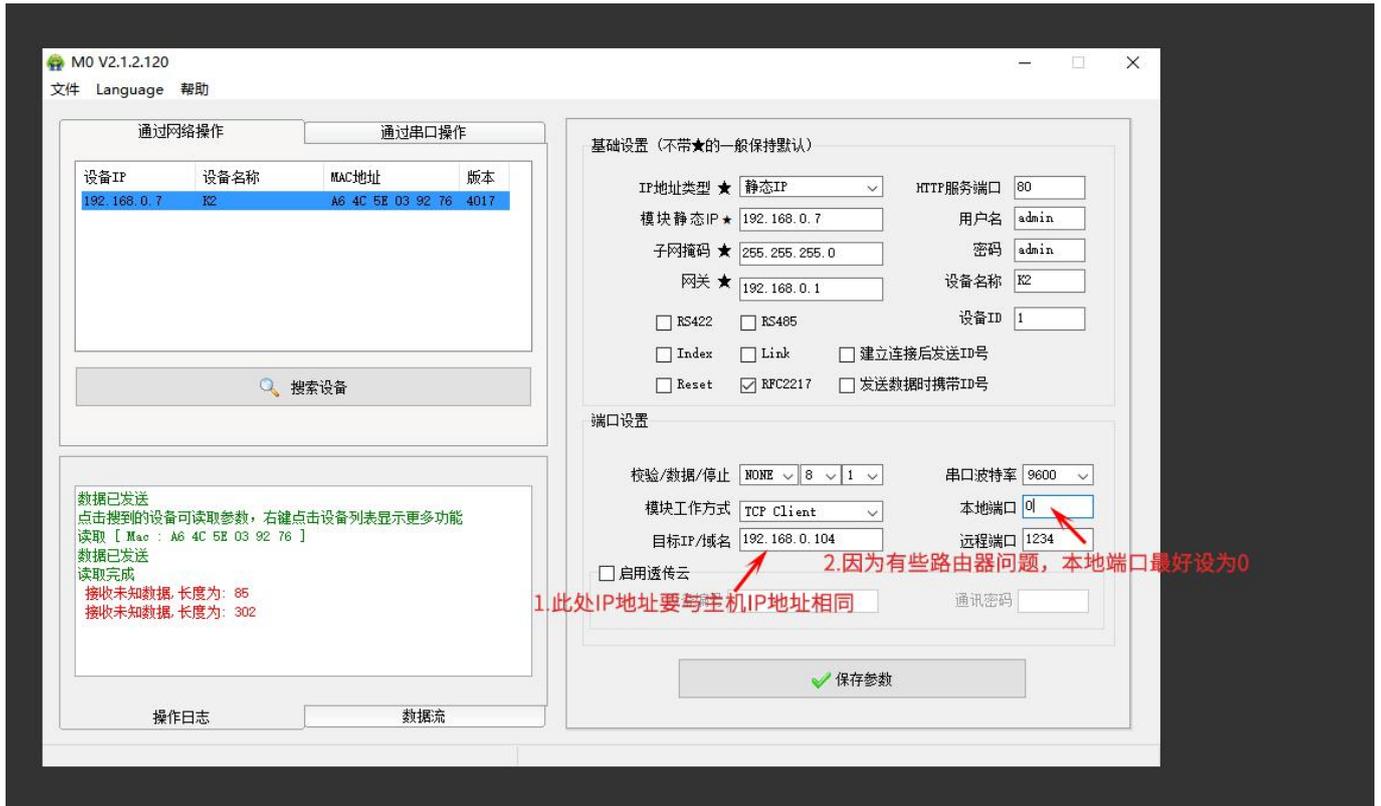
5.3 双击搜索到的产品，然后弹出配置窗口：



#### 5.4 配置成 TCP Server 模式:



#### 5.5 配置成 TCP Client 模式:



版本: V2.0 2021.10.08 更新