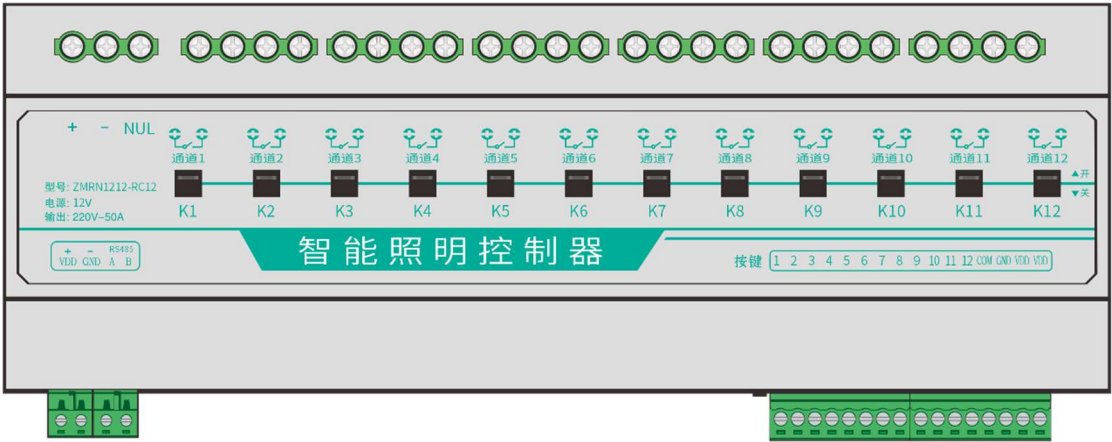


IOTZONE®

智慧照明·节能管理

12 通道网络继电器控制器

ZMRN1212-RC12



使用说明书

文件状态	项目名称	12 通道网络继电器 RC12	文件名称	使用说明书
[]草稿	文件标识	ZM-RC12-DOC	当前版本	V2.1
[√]正式发布	作者	DJB	完成时间	2022-10-8
[]正在修改	总页数	11	等级	中
说明书	页码	2—6	页数	5
通信协议	页码	7—11	页数	5

常州贞明电子科技有限公司

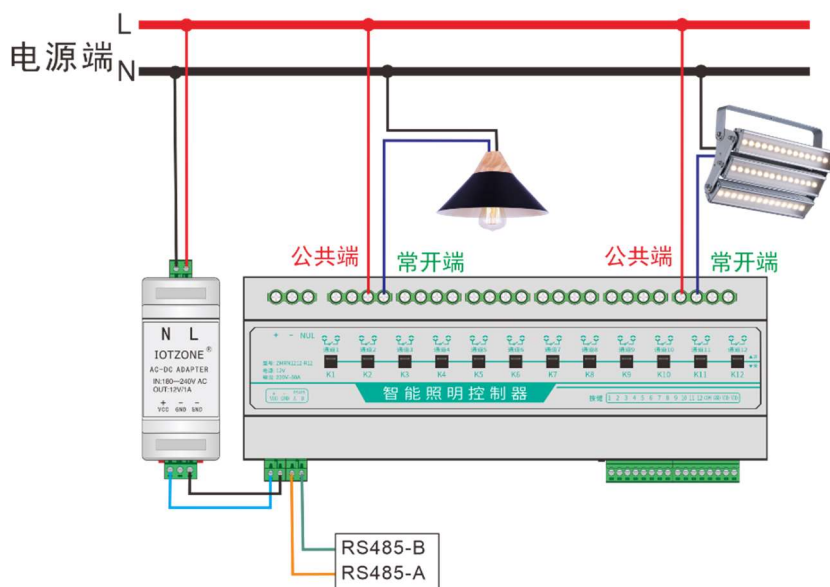
Zhenming Electronic Technology Co., Ltd

一、产品介绍

- 1、电源电压：12V，标配 12V/1A 电源；
- 2、输出端口：12 个独立 220V 无源输出；
- 3、输入端口：12 路输入，一组 RS485 输入；
- 4、继电器参数：磁保持继电器 50A 277V AC；
- 5、待机功耗：0.8W，最大功耗：3W；
- 6、输出端口允许最大电压/电流：AC 220V/50A，10 万次以上；
- 7、通讯：RS485 通讯，标准 Modbus RTU 协议；
- 8、工作温度：-45 ~ 85 摄氏度；
- 9、保存温度：-45 ~ 85 摄氏度；
- 10、保存湿度：5% ~ 95%RH；
- 11、安装：35mm 标准 DIN 导轨安装；
- 12、外壳尺寸：250×100×65mm（长*宽*高）。

二、产品接线

交流220V设备接线方法



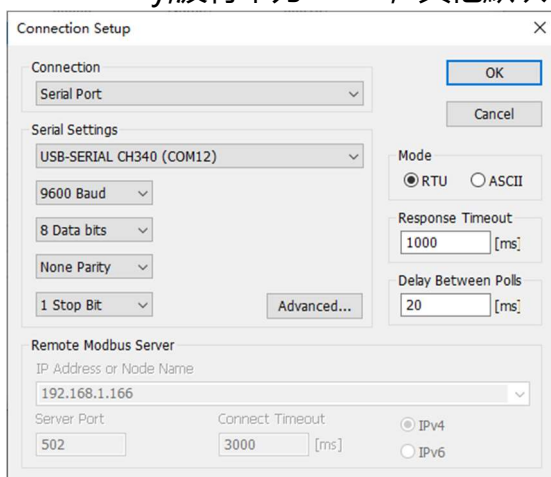
产品设备上方 12 组继电器输出接线端。设备下方从左到右依次是：外部电源输入（+、- 供电电源 12V）、RS485-A、RS485-B 端、初始化按键（用于初始化网络参数，***请勿操作）、1—12 输入端、COM、GND、VDD、VDD 端。

三、使用说明

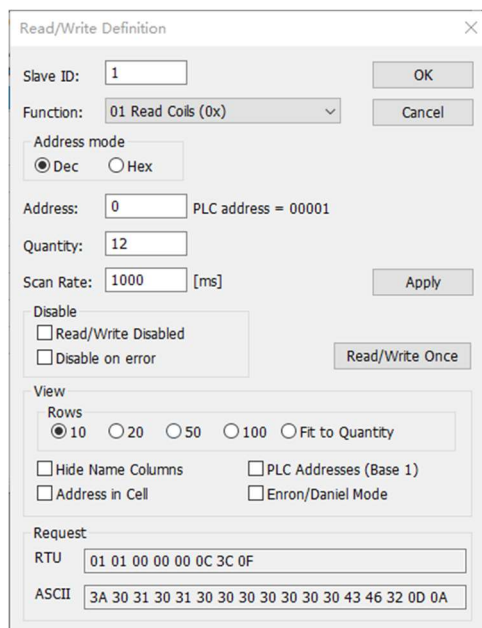
Modbus RTU 控制

设备默认地址是 1。

- (1) 设备上电。
- (2) 使用 USB 转 485 工具连接设备和电脑。
- (3) 打开 Modbus Poll 软件
- (4) 在 “Connection” 中选择 Connection Setup 进入设置页面。
“Connection” 选择 Serial Port; 在 “Serial settings” 中选择对应的 COM 端口, 选择 None Parity, 波特率为 9600, 其他默认, 设置如下图所示。

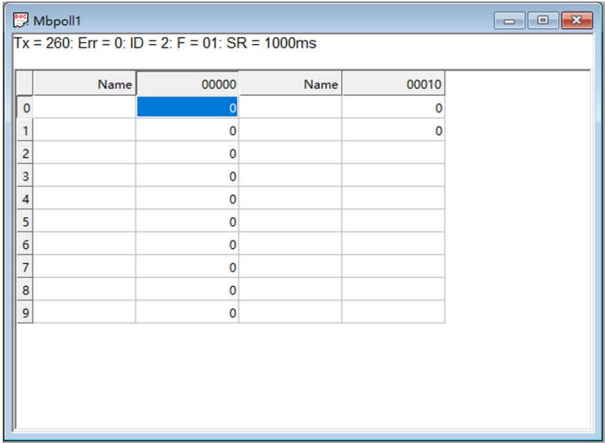


- (5) 在 “Setup” 中选择 “Read/Write Definition” 进入设置页面, “Slave ID” 是 1, 在 “Function” 中选择 01 Read Coils, “Address” 是 0, “Quantity” 根据设备继电器数量设置, 其他默认, 设置如下图所示。(例图是 12 通道继电器设备)



(6) 控制继电器

继电器的所有状态在下图显示，在图中选择就对应设备相应通道的继电器。
(例图是 12 通道继电器设备，选择的第一通道继电器)



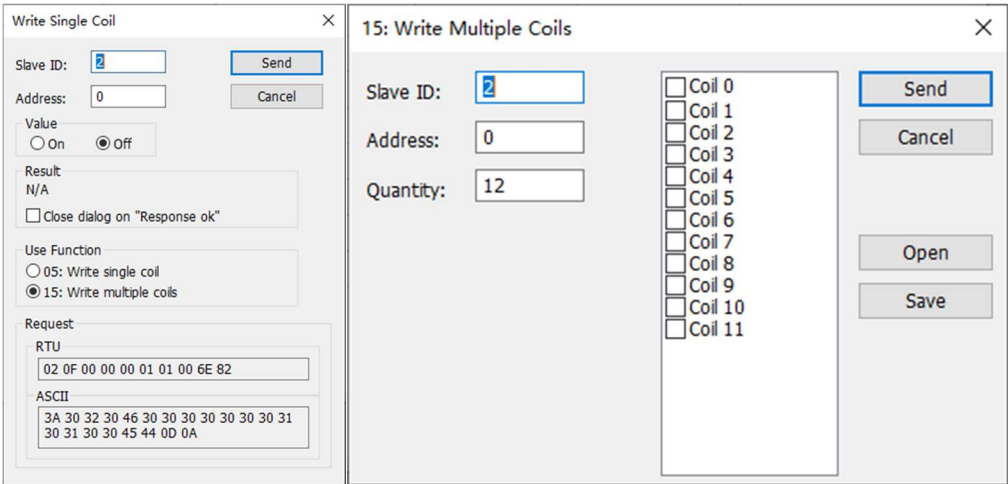
The screenshot shows a window titled 'Mbpoll1' with a status bar indicating 'Tx = 260; Err = 0; ID = 2; F = 01; SR = 1000ms'. The main area contains a table with 10 rows (0-9) and 4 columns. The first two columns are labeled 'Name' and '00000', and the next two are 'Name' and '00010'. All values in the table are '0'. The first row (0) is highlighted in blue.

	Name	00000	Name	00010
0		0		0
1		0		0
2		0		0
3		0		0
4		0		0
5		0		0
6		0		0
7		0		0
8		0		0
9		0		0

“05” 功能是控制单个继电器，“15” 是控制多个继电器打开或关闭
(在“ functions” 中选择)。

单个继电器操作示图

多个继电器操作示图



The left screenshot is titled 'Write Single Coil'. It has fields for 'Slave ID' (2), 'Address' (0), and 'Value' (radio buttons for On and Off, with Off selected). It also has a 'Send' button and a 'Cancel' button. Below these are sections for 'Result' (N/A), 'Use Function' (radio buttons for '05: Write single coil' and '15: Write multiple coils', with '15' selected), and 'Request' (RTU and ASCII hex values).

The right screenshot is titled '15: Write Multiple Coils'. It has fields for 'Slave ID' (2), 'Address' (0), and 'Quantity' (12). It features a list of coils from Coil 0 to Coil 11, each with a checkbox. There are 'Send', 'Cancel', 'Open', and 'Save' buttons on the right side.

四、蓝牙工具使用

蓝牙工具可远程控制其他设备的蓝牙功能，比如远程打开/关闭设备、发送数据、修改设备的 485 地址、修改 IP、还可以设置发送命令。

操作步骤如下：

1. 关注物联地带公众号，进入公众号，右下角点击“更多”选择蓝牙工具
2. 首页点击正确的设备号连接（注意：不可两台手机同时连接设备蓝牙，只可一对一连接）
3. 点击读取配置
4. 在对应页面进行操作（远程控制、修改 IP、修改 485 地址等）



扫一扫，关注公众号



五、应用场景

网络继电器控制器作为一种结合网络通信技术与继电器控制功能的设备，广泛应用于各类远程控制场景，为工业自动化、智能家居、农业、安防等领域提供了高效、便捷的解决方案。

1、工业自动化控制

设备远程启停：如电机、泵、阀门等。自动化生产线监控与联动：实时采集设备状态（如温度、压力、电流）根据预设条件触发联动控制。分布式设备管理：支持多台设备联网，通过云平台或本地服务器集中监控与管理，降低人工巡检成本。

2、智能家居与楼宇自动化

家电远程控制：通过手机/电脑远程控制空调、灯光、窗帘等设备，实现定时开关、场景模式切换。能源管理：结合电表数据，实现用电设备的智能调度降低能耗。安防联动：与门窗传感器、摄像头联动，当检测到异常时自动触发报警并关闭指定设备（如切断电源）。

3、农业物联网应用

灌溉系统控制：根据土壤湿度传感器数据，自动控制水泵的启停实现精准灌溉。温室环境调控：结合温湿度传感器，自动调节通风设备、遮阳帘，优化作物生长环境。畜牧业管理：远程控制饲料投放设备、通风系统，提升养殖效率。水务系统：远程控制水泵、阀门，实现水资源的智能调度与管理。

4. 能源与公共设施管理

电力设备监控：远程控制配电柜、断路器，实时监测电流、电压，预防过载或短路。路灯管理：根据光照强度或时间计划，自动控制路灯的开关，降低能耗。共享充电桩：用户通过手机/电脑远程启动充电桩，控制电源通断实现充电过程的自动化管理。智能储物柜：结合二维码或 RFID 技术，用户扫码后控制柜门开关，提升共享设备的便捷性。

5、安防与监控系统

门禁与报警：网络继电器可控制电磁锁、报警器。视频监控联动：与摄像头联动，自动控制灯光、警报器或通知安保人员。周界防护：结合红外对射传感器，在检测到入侵时自动启动防护设备（如高压电网、声光报警）。

6. 特殊环境应用

高温、高湿环境：网络继电器采用工业级设计，适应恶劣环境，如钢铁厂、化工厂的设备控制。环境模拟系统：控制温湿度试验箱、振动台等设备，模拟不同环境条件进行产品测试。自动化测试平台：在实验室中，可远程控制测试设备的电源、信号输入实现自动化测试流程。远程无人值守站点：在偏远地区（如基站、风力发电站），通过网络继电器实现设备的远程监控与维护。

通信协议

1. 查询设备状态

发送 state=?

返回

```
{  
  "cmd": "state",  
  "output": "000000000000",  
  "input": "000000000000",  
  "timer_en": 1,  
  "ts": 0,  
  "runtime": 2703,  
  "sn": " RC12xxxxxxxx638"  
}
```

2. 继电器控制

发送 setr=111111111111

1 表示打开继电器

0 表示关闭继电器

2 表示点触

3 表示翻转

X 表示不动作

4 表示互锁

返回

```
{  
  "cmd": "setr",  
  "output": "000000000000",  
  "input": "000000000000",  
  "timer_en": 1,  
  "ts": 0,  
  "runtime": 2944,  
  "sn": " RC12xxxxxxxx638"  
}
```

3. 设置 RS485 地址

发送 addr=2 表示地址为 2

设备地址可以 1~255

返回

```
{  
  "cmd": "device",  
  "sn": " RC12xxxxxxxx638",  
  "ts": 0,
```

```
"runtime": 3234,  
"addr": 2,  
"baudrate": 9600,  
"rsmode": 1,  
"crc": 0,  
"rtc": 1,  
"eeprom": 1,  
"powerflag": 0,  
}
```

4. 设置时序间隔时间

发送 jgtime=10 时序间隔时间为 1 秒

返回

```
{  
  "sn": " RC12xxxxxxxx638",  
  "intype": "111111111111",  
  "iorelation": "111111111111",  
  "ptime": [  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
    10,  
  ],  
  "save": 0,  
  "jgtime": 10,  
  "cmd": "relay"  
}
```

5. 输入类型设置

发送 intype=111111111111

1 表示边沿输入

0 表示电平输入

返回

```
{  
  "sn": " RC12xxxxxxxx638",
```



```
"intype": "111111111111",
"iorelation": "111111111111",
"ptime": [
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    ],
"save": 0,
"jgtime": 10,
"cmd": "relay"
}
```

6. 查询定时参数

逐个定时参数查询

发送 timer1=?

返回

```
{
    "sn": " RC12xxxxxxxx638",
    "num": 1,                //定时器序号
    "hour": 15,              //时
    "min": 27,               //分
    "sec": 51,               //秒
    "week": "0000000", //从左到右分别是星期日、星期一、星期二、星期三、星期四、星期五、星期六
    "output": " 000000000000",
    "ts": 1651908559,
    "runtime": 23067,
    "cmd": "timer1"
}
```

7. 定时参数设置

发送 timer1=150630/0000001/111111111111

timer1 定时器 1

150630 15(时)06(分)30(秒)

0000001 从左到右星期日、星期一、星期二、星期三、星期四、星期五、星期六

111111111111 设置继电器 1-12 打开

返回

```
{
  "sn": " RC12xxxxxxxx638",
  "num": 1,
  "hour": 15,
  "min": 6,
  "sec": 30,
  "week": "0000001",
  "output": "111111111111",
  "ts": 0,
  "runtime": 3982,
  "cmd": "timer1"
}
```

8. 场景参数查询

发送 scene=?

返回

```
{
  "sn": " RC12xxxxxxxx638",
  "son1": "3xxxxxxxxxx",
  "soff1": "3xxxxxxxxxx",
  "son2": "x3xxxxxxxxxx",
  "soff2": "x3xxxxxxxxxx",
  "son3": "xx3xxxxxxxxxx",
  "soff3": "xx3xxxxxxxxxx",
  "son4": "xxx3xxxxxxxxxx",
  "soff4": "xxx3xxxxxxxxxx",
  "son5": "xxxx3xxxxxxxxxx",
  "soff5": "xxxx3xxxxxxxxxx",
  "son6": "xxxxx3xxxxxxxxxx",
  "soff6": "xxxxx3xxxxxxxxxx",
  "son7": "xxxxxx3xxxxxxxxxx",
  "soff7": "xxxxxx3xxxxxxxxxx",
  "son8": "xxxxxxx3xxxxxxxxxx",
  "soff8": "xxxxxxx3xxxxxxxxxx",
  "son9": "xxxxxxx3xxxxxx",
  "soff9": "xxxxxxx3xxxxxx",
  "son10": "xxxxxxx3xx",
  "soff10": "xxxxxxx3xx",
}
```

```
"son11": "xxxxxxxxxx3x",  
"soff11": "xxxxxxxxxx3x",  
"son12": "xxxxxxxxxx3",  
"soff12": "xxxxxxxxxx3",  
"cmd": "scene"  
}
```

9. 传感器模式

发送 rs485mode=0 传感器模式

rs485mode=1 从机模式

```
{  
  "cmd": "device",  
  "sn": " RC12xxxxxxxx638",  
  "ts": 0,  
  "runtime": 82,  
  "addr": 1,  
  "baudrate": 9600,  
  "rsmode": 1,  
  "crc": 0,  
  "rtc": 1,  
  "eeprom": 1,  
  "powerflag": 0,  
}
```

10. 重启设备

发送 restart 重启设备