

协议说明

1. 通信命令帧格式

1.1 命令格式

帧头		目的地址	数据长度	命令字	数据	校验和
0x55	0xAA	1字节	1字节	1字节	数据1~数据n	1字节

1.2 应答格式

与命令格式相同；其中“目的地址”为“机器地址”，“命令字”为返填主机发来的命令字。

1.3 继电器按钮控制命令格式

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	继电器号 (1字节)	继电器值 (1字节)	校验和 (1字节)
0x55	0xAA	addr	0x02	0x00	继电器号	继电器值	校验和

返回值(data段) 00 成功 01 失败 例如：主机发送给设备的1号继电器，控制其闭合：

55 AA 01 02 00 01 01 04 闭合一号继电器(ON)

55 AA 01 02 00 01 00 03 断开一号继电器(OFF)

注：1)继电器号：为0x01-0x08。 2)继电器值：0x01表示闭合，0x00表示断开。

1.4 获取继电器状态命令格式

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和 (1字节)
0x55	0xAA	addr	0x00	0x01	校验和

例如：读取设备的工作状态：

55 AA 01 00 01 01 获取继电器状态

返回值 (data部分) 07 (八位代表板子的继电器的号，每一位的0表示打开，1表示关闭，返回07表明前三个继电器是打开的)

1.5 改变网络信息命令格式

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	IP地址(4字节)				...
0x55	0xAA	addr	0x0E	0x02	ipAddr1	ipAddr2	ipAddr3	ipAddr4	...

...	网关地址(4字节)				子网掩码地址(4字节)				服务器端口号(2字节)	校验和	
...	gatewayAddr1	gatewayAddr2	gatewayAddr3	gatewayAddr4	netMaskAddr1	netMaskAddr2	netMaskAddr3	netMaskAddr4	端口号的高八位	端口号的低八位	校验和

例如：给 IP 地址为:192.168.0.10，端口号：2000 的设备，改为 IP：192.168.1.11，网关：192.168.1.1，子网掩码 255.255.255.0，端口号为 2000。

55 AA 01 0E 02 C0 A8 01 0A C0 A8 01 01 FF FF FF 00 07 D0 C1

即（55AA010E02C0A8010AC0A80101FFFFFF0007D0C1）

返回值 00 成功 01 失败

注：1) IP 地址：ipAddr1: ipAddr2: ipAddr3: ipAddr3。

2) 网关：gatewayAddr1: gatewayAddr2: gatewayAddr3: gatewayAddr4

3) 子网掩码：netmaskAddr1: netmaskAddr2: netmaskAddr3: netmaskAddr4

1.6 获取网络信息命令格式

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和(1字节)
0x55	0xAA	addr	0x00	0x03	校验和

返回信息命令格式（data 部分）：

IP地址 (4字节)				网关 (4字节)				...
ipAddr1	ipAddr2	ipAddr3	ipAddr4	gatewayAddr1	gatewayAddr2	gatewayAddr3	gatewayAddr4	...

...	子网掩码 (4字节)				服务器端口号(2字节)	
...	netmaskAddr1	netmaskAddr2	netmaskAddr3	netmaskAddr4	端口高字节	端口低字节

1) IP 地址：ipAddr1: ipAddr2: ipAddr3: ipAddr3

2) 网关: gatewayAddr1: gatewayAddr2: gatewayAddr3: gatewayAddr4

3)子网掩码: netmaskAddr1: netmaskAddr2: netmaskAddr3: netmaskAddr4

1.7 设置名字(名字最长 32 字节)

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	名字的ascii码(n字节 n<=32)	校验和(1字节)
0x55	0xAA	addr	名字长度	0x05	名字ascii码	校验和

返回值（data 部分） 00 成功 01 失败

如果你要设置的名字是 DFRobot，那么你的 ASCII 就是 44 46 52 6F 62 6F 74 00，别忘了最后

的 00!

1.8 查询名字

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和(1字节)
0x55	0xAA	addr	0x00	0x06	校验和

假如名字是 DFRobot 则返回值 (data 部分) 为 44 46 52 6F 62 6F 74 00, 别忘了 00

1.9 查询版本号

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和(1字节)
0x55	0xAA	addr	0x00	0x07	校验和

假如版本是 V1.0 则返回值 (data 部分) 为 56 31 2E 30 00

1.10 更改波特率

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	波特率(4字节)	校验和(1字节)
0x55	0xAA	addr	0x04	0x08	波特率 (大端字节序)	校验和

例如: 设置波特率为 115200 则应该发送

55 AA 01 04 08 00 01 C2 00 CF 即 (55AA0104080001C200CF)

返回值(data 部分) 00 成功 01 失败

当前支持的波特率为 256000, 128000, 115200, 57600, 38400, 28800, 19200, 14400, 9600, 4800, 2400, 1200

1.11 更改 485 地址

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	485地址(1字节)	校验和(1字节)
0x55	0xAA	addr	0x01	0x09	0-255 0xAB不可用	校验和

假设设置原来地址为 01 的 485 设备新地址为 02, 那么发送的命令应该是:

55 AA 01 01 09 02 0C

返回值 (data 部分) 00 设置成功 01 设置失败

1.12 设置 DHCP(打开或关闭)

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	打开/关闭	校验和(1字节)
0x55	0xAA	addr	1	0x0A	1 / 0	校验和

返回值 (data 部分) 00 成功 01 失败

1.13 查询 DHCP

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和(1字节)
0x55	0xAA	addr	0x00	0x0B	校验和

返回值 (data 部分) 00 关闭 DHCP 01 打开 DHCP

1.14 重启

帧头 (2字节)		目的地址 (1字节)	数据长度 (1字节)	命令 (1字节)	校验和(1字节)
0x55	0xAA	addr	0x00	0x04	校验和

返回值 00 成功 01 失败

2.JSON Communication Format

(json 格式只支持网络端口解析, USB 口和 485 接口不支持此格式):

*设置继电器状态(on 表示打开, off 表示关闭)

示例

*发送正确命令:

```
{"relay1":"on","relay2":"off","relay3":"off","relay4":"on","relay5":"on","relay6":"on",  
"relay7":"off","relay8":"off"}
```

返回: {"resp":"ok"}

*发送错误命令:

```
{"relay1":"off","relay2":"off","relay3":"off","relay4":"off","relay5":"off","relay6":"off",  
"relay7":"off","relay9":"off"}
```

返回: {"resp":"error"}

*获取继电器状态

示例

发送: {"get":"relayStatus"}

返回:

```
{"relay1":"off","relay2":"off","relay3":"off","relay4":"off","relay5":"off","relay6":"off",  
"relay7":"off","relay8":"off"}
```

*设置设备名称

示例

发送: {"name":"DFRobot"}

返回: {"resp":"ok"}

*获取设备名称

示例

发送: {"get": "name"}

返回: {"name": "DFRobot"}

*设置网络配置信息

示例

发送:

```
{"ipaddr": "192.168.1.10", "gateway": "192.168.1.1", "netmask": "255.255.255.0", "port": "2000"}
```

返回: {"resp": "ok"}

*获取网络配置信息

示例

发送: {"get": "netconfig"}

返回:

```
{"ipaddr": "192.168.1.10", "gateway": "192.168.1.1", "netmask": "255.255.255.0", "port": "2000"}
```

*获取软件版本

示例

发送: {"get": "version"}

返回: {"version": "V1.0"}

*设置 485 地址

示例

发送: {"485addr": "22"} 设置 485 地址为 22, 立即生效

返回: : {"resp": "ok"} 成功 {"resp": "error"} 失败

*获取 485 地址

示例

发送: {"get": "485addr"}

返回: 如果 485 地址是 25, 则返回{"485addr": "25"}

*设置波特率

示例

发送: {"baudrate": "9600"} 设置通信波特率为 9600, 重启生效

返回: : {"resp": "ok"} 成功 {"resp": "error"} 失败

*获取波特率

示例

发送: {"get": "baudrate"}

返回: 如果波特率是 115200, 则返回{"baudrate": "115200"}

*设置 DHCP 状态

示例

发送: {"dhcp": "on"} 打开 dhcp {"dhcp": "off"} 关闭 dhcp

返回： : {"resp":"ok"} 成功 {"resp":"error"} 失败

*获取 DHCP 状态

示例

发送： {"get":"dhcp"}

返回 {"dhcp":"on"} DHCP 已开启 {"dhcp":"off"} DHCP 已关闭

*重启

{"reboot":"1"}

*错误命令

示例

发送： {"cmd":"test"}

返回： {"resp":"unsupported"}