山河行 MQTT 通讯协议 V1.0

1. 格式说明

{
"id":"",
"type":"格式",
"msg":
}

id	说明
id	发送的次序，每次都不一样，

type	格式说明
modbus	modbus 相关命令
modbus_reply	modbus 相关命令响应
hex	透传 16 进制的数据（预留）
ascii	透传 ascii 数据（预留）
heart	心跳主动上传

msg	说明
msg	发送的消息内容

2. 协议说明

2.1. modbus 类型命令

命令码总表：

命令码	说明
getdo	查询 DO 的状态
open	打开继电器
close	关闭继电器
opentime	打开继电器一段时间后再自动关闭
closetime	关闭继电器一段时间后再自动打开
setdo	控制多路继电器
opendo	打开多路继电器
closedo	关闭多路继电器
inversedo	反相操作多路继电器
getdi	查询 DI 的状态
getao	查询保持寄存器
setao	写保持寄存器

2.1.1. getdo

举例：查询 4 路 DO 状态

发送信息：

{"id":"12345","type":"modbus","msg":{"cmd":"getdo","addr":255,"start":0,"num":4}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	getdo 查询 DO 状态
addr	设备地址，广播地址为 255
start	起始位置，默认为 0
num	读取数量

回传信息：

{"id":"12345","type":"modbus_reply","msg":{"cmd":"getdo","data":"0101"}}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
cmd	getdo 查询 DO 状态
data	DO 状态数值，从左到右 “0101” 代表 1,3 路继电器断开，2,4 路继电器导通

2.1.2. open

举例：打开第一路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"open","addr":255,"channel":1}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	open 打开继电器
addr	设备地址，广播地址为 255
channel	1 代表第一路

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.3. close

举例：关闭第一路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"close","addr":255,"channel":1}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	close 关闭继电器
addr	设备地址，广播地址为 255
channel	1 代表第一路

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.4. opentime

举例：打开第一路继电器，2 秒后自动断开
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"opentime","addr":255,"channel":1,"time":20}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	opentime 打开继电器一段时候后自动断开
addr	设备地址，广播地址为 255
channel	1 代表第 1 路
time	操作的时间，单位是 0.1 秒

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.5. closetime

举例：关闭第一路继电器，2 秒后自动打开
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"closetime","addr":255,"channel":1,"time":20}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	closetime 打开继电器一段时候后自动断开
addr	设备地址，广播地址为 255
channel	1 代表第一路
time	操作的时间，单位是 0.1 秒

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.6. setdo

举例：打开 2,4,6,7,8 路继电器，关闭 1,3,5 路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"setdo","addr":255,"start":1,"val":"01010111"}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	setdo 操作继电器
addr	设备地址，广播地址为 255
start	1 代表从第一路开始
val	从左到右 0 代表断开，1 代表导通

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.7. opendo

举例：打开 2,4,6,7,8 路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"opendo","addr":255,"start":1,"val":"01010111"}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	opendo 打开继电器
addr	设备地址，广播地址为 255
start	1 代表从第一路开始
val	从左到右 0 代表忽略，1 代表导通

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.8. closedo

举例：关闭 2,4,6,7,8 路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"closedo","addr":255,"start":1,"val":"01010111"}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	closedo 关闭继电器
addr	设备地址，广播地址为 255
start	1 代表从第一路开始
val	从左到右 0 代表忽略，1 代表关闭

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.9. inversedo

举例：反向操作 2,4,6,7,8 路继电器
发送信息内容：
{"id":"12345","type":"modbus","msg":{"cmd":"inversedo","addr":255,"start":1,"val":"01010111"}
}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	inversedo 反向操作继电器
addr	设备地址，广播地址为 255
start	1 代表从第一路开始
val	从左到右 0 代表忽略，1 代表反向操作

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.1.10. getdi

举例：查询 4 路 DI 状态

发送信息：

{"id":"12345","type":"modbus","msg":{"cmd":"getdi","addr":255,"start":0,"num":4}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	getdi 查询 DI 状态
addr	设备地址，广播地址为 255
start	起始位置，默认为 0
num	读取数量

回传信息：

{"id":"12345","type":"modbus_reply","msg":{"cmd":"getdi","data":"0101"}}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
cmd	getdi 查询 DI 状态
data	DI 状态数值，从左到右 “0101” 代表 1,3 路 DI 导通 ， 2,4 路 DI 断开

2.1.11. getao

举例：查询 4 路保持寄存器
发送信息：
{"id":"12345","type":"modbus","msg":{"cmd":"getao","addr":255,"start":0,"num":4}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	getao 查询保持寄存器
addr	设备地址，广播地址为 255
start	起始位置，默认为 0
num	读取数量

回传信息：
{"id":"12345","type":"modbus_reply","msg":{"cmd":"getao","data":"006403E827104E20"}}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
cmd	getao 查询保持寄存器
data	16 进制的数据 0064 寄存器 1 数据 十进制为 100 03E8 寄存器 2 数据 十进制为 1000 2710 寄存器 3 数据 十进制为 10000 4E20 寄存器 4 数据 十进制为 20000

2.1.12. setao

举例：设置 4 路保持寄存器
发送信息：
{"id":"12345","type":"modbus","msg":{"cmd":"setao","addr":255,"start":800,"num":4,"val":"0064003E827104E20"}}

参数	说明
id	发送次序 id
type	modbus 类型命令
cmd	setao 设置保持寄存器
addr	设备地址，广播地址为 255
start	起始位置，默认为 800
num	读取数量
val	设置的数据 16 进制， 0064 寄存器 1 数据 十进制为 100 03E8 寄存器 2 数据 十进制为 1000 2710 寄存器 3 数据 十进制为 10000 4E20 寄存器 4 数据 十进制为 20000

回传信息：
{"id":"12345","type":"modbus_reply","msg":"success"}

参数	说明
id	发送次序 id
type	modbus_reply 响应命令
msg	成功 success

2.2. 透传 16 进制数据

发送信息：

{"id":"12345","type":"hex","msg":"01040000000131CA"}

2.3. 透传 ascii 数据

发送信息：

{"id":"12345","type":"ascii","msg":"abcd1234"}

2.4. 主动上传数据

设备 DO 主动上传：

{"id":"12345","type":"heart","msg":{"cmd":"getdo","data":"0101"}}

参数	说明
id	发送次序 id
type	heart 心跳上传命令
cmd	getdo　DO 状态
data	DO 状态数值，从左到右 “0101”代表 1,3 路继电器断开，2,4 路继电器导通

设备 DI 主动上传：

{"id":"12345","type":"heart","msg":{"cmd":"getdi","data":"0101"}}

参数	说明
id	发送次序 id
type	heart 心跳上传命令
cmd	getdi　DI 状态
data	DI 状态数值，从左到右 “0101”代表 1,3 路 DI 导通 ，2,4 路 DI 断开

设备 AI 主动上传：

{"id":"12345","type":"heart","msg":{"cmd":"getao","data":"006403E827104E20"}}

参数	说明
id	发送次序 id
type	heart 心跳上传命令
cmd	getao　AI 保持寄存器
data	16 进制的数据 0064 寄存器 1 数据 十进制为 100 03E8 寄存器 2 数据 十进制为 1000 2710 寄存器 3 数据 十进制为 10000 4E20 寄存器 4 数据 十进制为 20000