

继电器组态软件开放协议

发布日期	版本	维护人
2020-05-23	V0.01	wky
2020-6-11	V0.02	wky

说明：

- 1、本协议旨在解决，第三方软件快速实现输出节点的控制及输入的状态的获取
- 2、组态软件作为 TCP 服务端，监听端口默认为 **6612**；
- 3、第三方软件需要作为 TCP 客户端，向服务软件发起 TCP 连接，再进行通讯；
- 4、以下，组态软件简称 **server**，第三方软件简称 **client**；
- 5、本文档所有指令均为字符串，并且必须使用 **UTF8** 编码，否则中文会出现乱码现象。
- 6、同节点数据用 “|” 符号分隔，不属同节点数据用 “,” 号分隔

目录

一、 查询系统中的所有节点名.....	3
二、 控制输出节点动作.....	3
1. 直接控制.....	3
2. 点动控制.....	3
三、 输入节点状态主动上报设置.....	4
1. 输入节点订阅.....	4
2. 取消输入节点的订阅.....	4
3. 主动发送被订阅的输入节点状态.....	5
4. 获取订阅列表.....	5
四、 查询节点状态.....	5
1. 输入输出节点均有效，读取单个节点状态如下：	5
2. 读取多个节点如下：	5
五、 自定义协议.....	5
六、 指令不存在提示.....	6
1. 例如：XXX 指令是不存在的.....	6

一、查询系统中的所有节点名

Client --> server:

AT+NodeName=?\r\n

Server --> client:

存在节点即返回:

+NodeName:<EndFlag>,name1|nodetype1,name2|nodetype2,...,namex|nodetypex\r\n (0<x<=20)

EndFlag	EndFlag 为 0 时，即为返回的最后一帧；EndFlag 非 0 时，不是最后一帧
name	节点名，可为中英文，请使用 UTF8 进行转码
nodetype	0 为输出节点类型，1 为输入节点类型

不存在节点即返回:

+NodeName:node is null\r\n

二、控制输出节点动作

1. 直接控制

client-->server:

AT+OutNodeCtl=name|state\r\n

server-->client:

存在节点并在线:

+OutNodeCtl:name|state|ok\r\n

不存在此节点名，包括不是输出节点类型:

+OutNodeCtl:name|state|null\r\n

节点离线:

+OutNodeCtl:name|state|offline\r\n

name	节点名
state	状态值，0 为断开，1 为闭合

2. 点动控制

client-->server:

AT+OutNodeCtl=name|state|interval\r\n

server-->client:

正确回复:

+OutNodeCtl:name|state|interval|ok\r\n

不存在此节点名，包括不是输出节点类型：

+OutNodeCtl:name|state|interval|null\r\n

节点离线：

+OutNodeCtl:name|state|interval|offline\r\n

name	节点名
State	状态值。为 1 时，即先闭合再断开；为 0 时，即先断开再闭合
interval	点动时间

点动开，以先闭合 3 秒再断开为例：

client-->server:

AT+OutNodeCtl=out|1|3\r\n

Server-->client:

正确回复：

+OutNodeCtl:out|1|3|ok\r\n

不存在此节点名，包括不是输出节点类型：

+OutNodeCtl:out|1|3|null\r\n

节点离线：

+OutNodeCtl:out|1|3|offline\r\n

三、输入节点状态主动上报设置

1. 输入节点订阅

client-->server:

AT+SetInNodeSub=name1,name2,...,namex\r\n (0< x <20)

server-->client:

正确订阅：

+SetInNodeSub:name1,name2,...,namex,OK\r\n

不存在的节点：

+SetInNodeSub:name1,name2,...,namex,null\r\n

2. 取消输入节点的订阅

client-->server:

AT+DelInNodeSub=name1,name2,...,namex\r\n (0<x<20)

server-->client:

+DelInNodeSub:name1,name2,...,namex\r\n (0<x<20)

3. 主动发送被订阅的输入节点状态

server-->client:
+PubInNodeState:name1|state1,name2|state2,...,namex|statex\r\n (0<x<20)

4. 获取订阅列表

client-->server:
AT+GetInNodeSub=?\r\n
server-->client:
+GetInNodeSub:name1,name2,...,namex\r\n (0 < x < 20)

name	节点名
State	输入节点状态值

四、查询节点状态

1. 输入输出节点均有效，读取单个节点状态如下：

client-->server:
AT+GetNodeState=name|?\r\n
server-->client:
正确回复：
+GetNodeState:name|state\r\n
不存在该节点名：
+GetNodeState:name|null\r\n

2. 读取多个节点如下：

client-->server:
AT+GetNodeState=name1|?,name2|?,...,namex|?\r\n (0<x<20)

name	节点名
?	表示询问状态的意思，必须为英文状态下的问号

五、自定义协议

自定义协议只适用于场景上的场景模块，例如在场景的指令模块设置了指令 abc，当场景执行到此模块时，会向客户端发送 AT+DefaultCmd=abc\r\n，其中 AT+DefaultCmd= 为固定命令头；客户端需要回应 AT+DefaultCmd=abc|OK\r\n，如果没收到客户端的回应，服务端即会每隔 5 秒重发一次，最多重发三次，最多共发四次，当客户端不在线时，即不会发送，会

等待客户上线后才发送。以 abc 指令为例，发送如下所示：

- 1、server --> client
AT+DefaultCmd=abc\r\n
- 2、Client --> server
AT+DefaultCmd=abc|OK\r\n

六、指令不存在提示

1. 例如：XXX 指令是不存在的

```
client-->server:
  AT+xxx=?\r\n
server-->client:
  +xxx:not exist!\r\n
```