底层控制雕刻机相关协议

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号： | V0.3 | 文档编号： |  |
| 文档密级： | 保密 | 归属项目： |  |
| 产品名： |  | 子系统名： | 管控单元底层协议 |
| 编写人： | 李苏旺 | 编写日期： | 2018.12.27 |

修改履历

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改人 | 日期 | 内容 |
| 0.1 | 李苏旺 | 2018.12.27 | 初版完成 |
| 0.2 | 陆明 | 2019.1.2 | 1、请求协议改为http post，同时修改部分指令名称和去除冗余参数；2、新增连接/断开雕刻机操作 |
| 0.3 | 陆明 | 2019.1.3 | 1、新增“开始雕刻”接口；2、修复json格式错误 |
| 0.4 | 陆明 | 2019.1.9 | 1、在所有响应结果外面再包一层；2、修改功能与名称不一致问题。 |

## 查询全部雕刻机信息

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

" command ":"query\_all\_machines\_info"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

All\_machines:

[

{

taskNo:任务编号,

machine\_ip:雕刻机IP，

currentStatus:working|waiting|error|maintain|\*\*\*, 用整形状态码来表示

worktime:\*\*\*,

gNo:G代码编号,

rowNo:当前执行的G代码行号,

"carveId ":"设备编号",

"errmsg": "当前设备对应的错误信息",

"errmsg\_for\_user": "用户可懂的错误信息",

"ret": 0 //0：成功；非0：错误码

}

…

],

},

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 查询一个雕刻机信息

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

"command":"query\_one\_machine\_info",

"carveId ":"设备编号"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

taskNo:任务编号,

machine\_ip:雕刻机IP，

currentStatus:working|waiting|error|maintain|\*\*\*,用整形状态码来表示

worktime:雕刻机用于雕刻的总时间，即总雕刻时间,

gNo:G代码编号,

rowNo:当前执行的G代码行号,

"carveId":"设备编号",

"errmsg": "当前设备对应的错误信息",

"errmsg\_for\_user": "用户可懂的错误信息",

"ret": 0 //0：成功；非0：错误码

},

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 通知GCode文件下载完毕

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

"command":"download\_gcode\_OK",

"filepath":"nc文件对应的绝对路径",

"carveId":"雕刻机编号",

"taskNo":"任务编号",

gNo:G代码编号

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"carveId":"设备编号", //可能为空

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg: 错误原因

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 通知一台雕刻机急停

* 请求参数

body 参数

{

"command":"emergency\_stop\_one",

"carveId":"设备编号"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"carveId":"设备编号", //可能为空

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg: 错误原因

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 通知全部雕刻机急停

* 请求参数

body 参数

{

"command":"emergency\_stop\_all"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"results":

[

{

"carveId":"设备编号",

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg:错误时的原因，未出错时为None

}

]

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 通知一台雕刻机重启

* 请求参数

body 参数

{

"command":"reboot\_one",

"carveId":"设备编号"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"carveId":"设备编号", //可能为空

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg: 错误原因

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 通知全部雕刻机重启

* 请求参数

body 参数

{

"command":"reboot\_all"

}

返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"results":

[

{

"carveId":"设备编号",

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg:错误时的原因，未出错时为None

}

]

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 调整雕刻机运行速度

* 请求参数

body 参数

{

"command":"adjust\_speed",

"carveId":"设备编号",

"speed\_percent":0.5

}

返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"carveId":"设备编号", //可能为空

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg: 错误原因

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 连接雕刻机

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

"command": "connect",

"carveInfo":

[

{

"ip":"雕刻机ip地址",

"carveType":"设备型号",

"carveId":"设备编号",

"worktime":设备之前的总雕刻时间，单位分钟,

"carveExFactory":"厂商类型，如BaoYuan"

}

]

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"results":

[

{

"carveId":"设备编号",

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg:错误时的原因，未出错时为空

}

]

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 断开雕刻机

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

"command": "disconnect",

"carveInfo":

[

{

"carveId":"设备编号"

}

]

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"results":

[

{

"carveId":"设备编号",

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg:错误时的原因，未出错时为空

}

]

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 开始雕刻

* 接口地址

http://ip:port/

* 请求方法

POST

Content-Type application/json

Content-LengthHTTP消息实体的传输长度

Accept  application/json

* 请求参数

body 参数

{

"command":"start",

"carveId":"设备编号"

}

* 返回格式：

Application\json

{

"host": "管控服务地址",

"result":{

"carveId":"设备编号", //可能为空

ret: 返回码（0:成功，非0：错误码）,

"errmsg\_for\_user": "用户可懂的错误信息",

errmsg: 错误原因

}

,

"server\_errmsg": "总体出错信息",

"server\_errno": 0 调用情况状态码 0:成功；非0：错误码 （解析子json时，请先判定相关值是否为空）

}

## 问题

1. 服务因为崩溃，重启了怎么办？是否需要增加数据存储模块？