

# 开源硬件(8)

陈天翼



# 接口协议

界面端和树莓派基于Open Sound Control(OSC)消息协议完成数据通信

界面端定义为客户端（监听端口9091），只能连接一个服务端

树莓派端定义为服务端（监听端口9090），只响应一个已连接的客户端，忽略其它客户端请求

统一使用OSC消息地址 “/jsonmsg”

数据通信分为3个步骤

- 1) 连接请求
- 2) 传送指令和接收传感器数据
- 3) 退出连接

所有传送的数据遵循JSON格式

# 客户端指令（1）连接请求

由客户端发起连接请求，格式如下：

```
{  
  "command": "conn",  
  "id": "XXXX",    //自己起的名称ID  
  "server": "XXXX", //服务器端的名称ID  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```

服务端返回响应，格式如下：

```
{  
  "error": XXXX, //如果成功，返回0。否则，返回自定义错误码(>0)。  
  "message": "XXXXXX", //返回成功或错误信息。  
  "clientid": "XXXX" //客户端起的名称ID  
  "timestamp": XXXXXX //客户提交的时间戳  
  "token": "XXXXXX" //服务端生成一个令牌，用于后面客户端发的消息  
}
```

## 客户端指令 (2) 断开连接

格式如下：

```
{  
  "token": "XXX", //握手时服务端返回的令牌  
  "command": "quit",  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```

## 客户端指令 (3) 发送预定义节点指令

客户端发送预定义模式运行指令，格式如下：

```
{  
  "token": "XXX", //握手时服务端返回的令牌  
  "command": "run", //按预定义模式运行  
  "mode": "1", //预定义模式1或2  
  "settings": [], //  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```

# 客户端指令（4） 发送自定义节点指令

客户端发送自定义模式运行指令，格式如下：

```
{
  "token": "XXX", //握手时服务端返回的令牌
  "command": "run",
  "mode": "custom", //按自定义模式运行
  "settings": [    //设置
    {
      "node": 1, //节点号
      "power": 0, //运行状态，可为0、1、2
      "order": 0, //运行顺序，第几个运行，可多个同时运行，值范围 0 - 15
      "duration": X, //运行时长，可为0 - 循环不停，1 - 1秒，2, 3
      "delay": 0 //完成后等待秒数
    },
    ...
    {
      "node": 16, //节点号
      .....
    }
  ],
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性
}
```

## 客户端指令 (5) 发送停止节点指令

客户端发送预定义模式运行指令，格式如下：

```
{  
  "token": "XXX", //握手时服务端返回的令牌  
  "command": "stop", //停止预定义模式运行  
  "mode": "custom", //预定义模式1、2或custom  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```

## 客户端指令 (5) 发送传感器启用指令

客户端发送传感器指令，格式如下：

```
{  
  "token": "XXX", //握手时服务端返回的令牌  
  "command": "sensor", //按预定义模式运行  
  "enable": 1, //启用传感器1、2或3  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```



## 客户端指令（6） 发送传感器禁用指令

客户端发送传感器指令，格式如下：

```
{  
  "token": "XXX", //握手时服务端返回的令牌  
  "command": "sensor", //按预定义模式运行  
  "disable": 1, //禁用传感器1、2或3  
  "timestamp": XXXXXX //时间戳对每条消息具有唯一性  
}
```

# 服务端响应 (1)

对于客户端指令 (2 – 5) , 服务端返回响应, 格式如下:

```
{  
  "error": XXXX,           //如果成功, 返回0。否则, 返回自定义错误码(>0)。  
  "message": "XXXXXX",    //返回成功或错误信息。  
  "timestamp": XXXXXXXX   //客户提交的时间戳  
}
```

## 服务器响应 (2)

服务端向客户端发送传感器数据，格式如下：

```
{  
  "sensor": 1,    //传感器编号。  
  "value": [      //传感器数据（数值或字符串）数组，可只回传一个数据  
    XX,  
    XX,  
    ...  
  ],  
  "token": "XXXXXX" //客户端令牌，用于身份确认  
  "timestamp": XXXXXX //服务端时间戳  
}
```

# 注意事项

1) 有疑惑就问

2) 注意中英文区别，不要拼写错!!!

谢谢