

## 盈客通维码器 socket 协议文档 v1.5

### 测试服务：

地址 (Ip: 端口) 类型: TCP:

47.94.208.29:7701 (协议 1.1 该服务不支持)

备注: 最好自行搭建服务 (较容易调试),

后台发送指令接口 (供测试客户端接收指令)

客户端相关: url: <http://47.94.208.29:5543/api/client/>

查看所有客户端: 方式: get

参数: 无

授权客户端: 方式: Put

参数 (x-www-form-urlencoded) : id(客户端列表中客户端的 ID)

取消授权客户端: 方式: Delete

参数 (x-www-form-urlencoded) : id(客户端列表中客户端的 ID)

出币: url: <http://47.94.208.29:5543/api/coin> 方式: Post

参数 (x-www-form-urlencoded) : CoinCount (出币数量)、clientid (客户端 ID, 识别号)、chance (概率)、voltage (电压)

### 数据头和尾

名称	数据	数据类型
头	+cmd+	string
尾	-end-	string

### 指令编码:

心跳	1000
接受出币	2000
出币完成	2001
执行失败	4000
上报故障	7001

---

指令直接转为 byte[]

---

## 特定数据规定

### 数据类型最大长度

数据类型	长度
String	100
json	

## 消息关键词解释

数据名称	解释	数据类型	长度		
ClientID	客户端 ID	String			
MessageID	消息 ID	String	24(max)		
CoinCount	出币数量	Int	1-99		
Chance	概率	Int	1-99		
Voltage	电压	Int	1-99		

## 概率解释 (Chance)

取值范围为 0 到 100

0 为永不中奖, 100 为每次都中奖

### 1. 心跳

过程解释:

当完成后, 马上开启心跳线程 (注意: 服务端也会向客户端发起心跳, 客户端必须处理服务端的心跳) 心跳速度为 10s/次

指令发送:

传递[心跳]指令和设备 ID 数据为:

```
{
  "ClientID": "3234"
}
```

将数据转为 byte[] 后追加到指令后面

指令接收:

接收服务端[心跳]指令(服务器只发送指令)

## 2. 接收出币

过程解释:

服务器会随时向客户端[接受出币]指令, 客户端需要按照该设备做出出币的动作

注意: 这里的概率是指当前投币后生效的概率, 当前投币结束后再回到默认的概率

指令接收:

[接受出币]+携带数据

从 byte[] 转 string 为:

```
{  
  "MessageID": "jfooejewfjiojfoeiie",  
  "ClientID": "3234",  
  "CoinCount": 32,  
  
  "Voltage": 32,  
  "Chance": 80  
}
```

注意: 接收到该数据后要核实该 ClientID 是否与本客户端 ID 一致, 如果不一致则发送失败信息到服务端

CoinCount、Voltage、Chance 最值都为 1-99

## 3. 出币完成

过程解释:

当出币完成后, 马上发送[出币完成]指令到服务端

指令发送:

传递[出币完成]指令+携带数据

```
{  
  "MessageID": "jfooejewfjiojfoeiie",  
  "ClientID": "3234"  
}
```

将数据转为 byte[] 后追加到指令后面)

## 4. 执行失败

过程解释:

当执行失败后, 马上发送[执行失败]指令到服务端

指令发送:

传递[执行失败]指令+携带数据

```
{  
  "MessageID": "jfooejewfjiojfoeiie",  
  "ClientID": "3234"  
}
```

将数据转为 byte[]后追加到指令后面)

## 5. 上报故障

过程解释:

当客户端有特殊故障信息后, 马上发送[上报故障]指令到服务端

指令发送:

传递[上报故障]指令+携带数据

```
{  
  "ClientID": "3234",  
  "BugInfo": {  
    "Content": "日志详情内容",  
  }  
}
```

将数据转为 byte[]后追加到指令后面)

## 6. 断开自动连接

过程解释:

当连接中断时, 尝试再次连接, 3 次后 3 分钟后再执行该任务