

## 盈客通维码器 socket 协议文档 v1.0

### 测试服务：

地址 (Ip: 端口) :

47.94.208.29:7701

备注：最好自行搭建服务（较容易调试）

服务器出币：

<https://xcxh.360yingketong.com/LSSDeviceServer/server/sendoutcoin?id=11&count=10>

参数解释：id：客户端 id，count：出币数量

注意：连接后的 client 需要进行授权，授权后才可以进行出币

授权页面：

<https://xcxh.360yingketong.com/LSSDeviceServer>

### 指令编码：

心跳	1000
接受出币	2000
出币完成	2001
执行失败	4000
设置默认概率	9000
设置默认概率成功	9001

---

指令要写在 byte[] 数据的前 4 位，其中每一位的转换顺序为  
int->string->char->byte

---

### 指令编码数据示例：

```
var type = 1000;
var arr = type.ToString().ToArray();
var head=new byte[] { (byte)(int)arr[0], (byte)(int)arr[1], (byte)(int)arr[2],
(byte)(int)arr[3] };
```

## 消息关键词解释

数据名称	解释	数据类型
ClientID	客户端 ID	String
MessageID	消息 ID	String
CoinCount	出币数量	Int
Chance	概率	Int

### 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:

服务器会随时的向客户端发送此命令, 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:

指令+携带数据

从 byte[] 转 string 为:

```
{
  "MessageID": "jfooejewfjiojfoeie",
  "ClientID": "3234",
  "CoinCount": 32,
  "Chance": 100
}
```

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

### 3. 出币完成

过程解释：

当出币完成后，马上发出币完成指令到服务端

指令发送：

传递指令+携带数据

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

将数据转为 byte[] 后追加到指令后面)

### 4. 设置默认概率

过程解释：

服务器向客户端发送此命令，客户端需要按照该设备设置默认概率

指令接收：

指令+携带数据

从 byte[] 转 string 为：

```
{
  "MessageID": "jfooejewfjiojfoeie",
  "ClientID": "3234",
  "Chance": 100
}
```

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

### 5. 设置默认概率成功

过程解释：

当概率设置完成后，马上发送完成指令到服务端

指令发送:

传递指令+携带数据

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

将数据转为 byte[] 后追加到指令后面)

## 6. 执行失败

过程解释:

当概率设置完成后, 马上发送完成指令到服务端

指令发送:

传递指令+携带数据

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

将数据转为 byte[] 后追加到指令后面)

## 7. 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