

DataChannel通信协议

遥控器控制发的控制指令：

▼ 遥控器端发给车端的控制JSON数据格式：

JSON

📄 复制代码

```
1 {
2   "ts": 0,          /* 发送此消息的时间；整型数U64；格式为UNIX时间戳，是从1970年1月1
   日0点0分0秒开始所经过的毫秒数 */
3   "bn": 9120,       /* 消息流水号；整型数U64；从0开始递增 */
4   "tr": 0,          /* 油门控制量；浮点数Float32；值范围为[0, 100]，有效小数位2位
   */
5   "bk": 0,          /* 刹车控制量；浮点数Float32；值范围为[0, 100]，有效小数位2位
   */
6   "st": 0,          /* 方向盘控制量；浮点数Float32；值范围为[-100, 100]，有效小数位
   2位；负值为方向盘左转，正值为方向盘右转 */
7   "g": "N",         /* 挡位控制量；ASCII字符；值为N、D、R、P */
8   "m": 0,           /* 控制模式；整型数U8；3：车自动控制模式，5：远程控制，6：远程控制
   +车自动控制模式 */
9   "vts": 0          /* 车辆最后一次状态反馈数据包的车辆时间戳；整型数U64；格式为UNIX时
   间戳，是从1970年1月1日0点0分0秒开始所经过的毫秒数 */
10 }
```

`{"ts":1653531739951,"bn":59842,"tr":0,"bk":0,"st":100,"g":"N","m":0,"dts":0}`

车端状态反馈的数据个数：

▼ 车端将车辆状态发送改为遥控器的JSON数据格式:

JSON | 复制代码

```
1 {
2   "ts": 0,          /* 发送此消息的时间; 整型数U64; 格式为UNIX时间戳, 是从1970年1月1
   日0点0分0秒开始所经过的毫秒数 */
3   "bn": 9120,       /* 消息流水号; 整型数U64; 从0开始递增 */
4   "g": "N",         /* 挡位控制量; ASCII字符; 值为N、D、R、P、- */
5   "m": 0,           /* 控制模式; 整型数U8; 3: 车自动控制模式, 5: 远程控制, 6: 远程控制
   +车自动控制模式 */
6   "pL": 85,         /* 剩余电量百分比; 整型数S8; -1: 未知; */
7   "spd": 45,        /* 车速; 整型数S16; 单位km/h; -1: 未知; */
8   "located": true,   /* GPS是否定位; bool型; true: 定位, false: 未定位; */
9   "lng": 116.261115, /* GPS经度; 浮点数Float64; 单位度; 保留6位小数; */
10  "lat": 39.948574,  /* GPS纬度; 浮点数Float64; 单位度; 保留6位小数; */
11  "alt": 396.9,       /* GPS海拔; 浮点数Float32; 单位米; 保留1位小数; */
12  "angle": 32.2,      /* GPS航向角度; 浮点数Float32; 单位度; 保留1位小数; */
13  "satCnt": 8         /* 有效GPS卫星数; 整型数U8; */
14 }
```

附: 延时值获取方式:

车端将车的时间戳发送给遥控端;

遥控端将最后一次收到车端时间戳加到发送的指令报文中,然后发送给车端;

车端收到遥控指令后, 车的时间戳 - 遥控器端的车端时间戳, 等于约等于= 总延迟时2倍 + 遥控器的发送时间间隔 / 2;

遥控器控制发的查看文件列表指令:

▼ 遥控器端发给车端的查看文件列表指令JSON数据格式:

JSON | 复制代码

```
1 {
2   "cmd": 0x0001,     /* 消息类型, 无符号16位整型*/
3   "bn": 9120,        /* 消息流水号; 整型数U64; 从0开始递增 */
4   "type": "IMAGE",   /* 文件类型; 值为IMAGE、VIDEO、AUDIO */
5   "page": 1,         /* 当前页数; 整型数U32*/
6   "psize": 25        /* 每一页的文件数; 整型数U8; 值为25、50、100*/
7   "date": "2012-04-23" /* 特定的日期; 根据日期过滤, 值为""则不用过滤*/
8 }
```

车端状态反馈的文件列表:

▼ 车端状态反馈的文件列表JSON数据格式:

JSON | 复制代码

```

1  {
2    "cmd":0x0001,      /* 消息类型, 无符号16位整型 */
3    "res":0,          /* 回复消息类型, 0:成功/确认; -1: 失败; -2: 消息有误; -3:不支持; */
4    "bn":9120,        /* 对应服务器端请求的流水号 */
5    "type":"IMAGE",   /* 文件类型; 值为IMAGE、VIDEO、AUDIO */
6    "tf":1250,        /* 文件总数, 整型数U32 */
7    "page":1,         /* 当前页数; 整型数U32 */
8    "psize":25,       /* 每一页的文件数; 整型数U8;值为25、50、100*/
9    "date":"2012-04-23" /* 特定的日期; 根据日期过滤, 值为""则不用过滤*/
10   "fl":             /* 文件列表, JSON内包含文件名称 */
11   [
12     "2022-05-18-18:33.png" /* 文件名称*/
13     "2022-05-18-18:35.png" /* 文件名称*/
14   ]
15   /*
16   {
17     "2022-05-18-18:33.png":[image arraybuffer]
18     "2022-05-18-18:35.png":[image arraybuffer]
19   }
20   */
21 }

```

附：需测试是否能将缩略图加入文件列表中（数据过大可能会传送失败）

遥控器控制发的获取文件指令：

▼ 遥控器控制发的获取指令JSON数据格式:

JSON | 复制代码

```

1  {
2    "cmd":0x0002,      /* 消息类型, 无符号16位整型 */
3    "bn":9120,        /* 消息流水号; 整型数U64; 从0开始递增 */
4    "type":"IMAGE",   /* 文件类型; 值为IMAGE、VIDEO、AUDIO */
5    "fn":"2022-05-18-18:33.png" /* 文件名称*/
6  }

```

车端状态反馈的获取文件指令结果：

▼ 车端状态反馈的获取文件指令结果JSON数据格式:

JSON | 复制代码

```
1 {
2   "cmd":0x0002,      /* 消息类型, 无符号16位整型 */
3   "res":0,           /* 回复消息类型, 0:成功/确认; -1: 失败; -2: 消息有误; -3:不
   支持; */
4   "bn":9120,         /* 对应服务器端请求的流水号 */
5   "fs":297503        /* 文件大小, 整数型U64; 将要发送文件的大小单位bytes */
6 }
```

附: 当车端获得文件名称后, 在同一个peer connection中打开一个新的data channel, 用文件名命名, 将图片以arraybuffer的形式传送给遥控器端 (文件较大的话 需要切成几块发送
<https://levelup.gitconnected.com/send-files-over-a-data-channel-video-call-with-webrtc-step-6-d38f1ca5a351>)

遥控器控制发的获取配置信息指令:

▼ 遥控器控制发的获取配置信息JSON数据格式:

JSON | 复制代码

```
1 {
2   "cmd":0x0003,      /* 消息类型, 无符号16位整型 */
3   "bn":9120          /* 消息流水号; 整数型U64; 从0开始递增 */
4 }
```

车端状态反馈的配置信息:

▼ 车端状态反馈的配置信息JSON数据格式:

JSON | 复制代码

```
1 {
2   "cmd":0x0003,      /* 消息类型, 无符号16位整型 */
3   "res":0,           /* 回复消息类型, 0:成功/确认; -1: 失败; -2: 消息有误; -3:不
   支持; */
4   "bn":9120,         /* 对应服务器端请求的流水号 */
5   "autoR":true,      /* 自动录像; B00L; 开/关*/
6   "autoL":true,      /* 开机自启动; B00L; 开/关*/
7   "audio":true,      /* 音频; B00L; 是/否*/
8   "duration":30,     /* 录像时长; 整数型U8; 30分钟*/
9   "quality":1,       /* 录像质量; 整数型U8; 1:低; 2:中; 3:高*/
10  "memoryL":100      /* 最大存储内存; 整数型U8; 100GB*/
11 }
```

遥控器控制发的更改配置信息指令：

▼ 遥控器控制发的更改配置信息JSON数据格式：

[JSON](#)[复制代码](#)

```
1 {
2   "cmd":0x0004,      /* 消息类型，无符号16位整型 */
3   "bn":9120,         /* 对应服务器端请求的流水号 */
4   "autoR":true,      /* 自动录像；B00L；开/关*/
5   "autoL":true,      /* 开机自启动；B00L；开/关*/
6   "audio":true,      /* 音频；B00L；是/否*/
7   "duration":30,     /* 录像时长；整型数U8；30分钟*/
8   "quality":1,       /* 录像质量；整型数U8；1:低；2:中；3:高*/
9   "memoryL":100      /* 最大存储内存；整型数U8；100GB*/
10 }
```

车端状态反馈的更改配置结果：

▼ 车端状态反馈的配置信息JSON数据格式：

[JSON](#)[复制代码](#)

```
1 {
2   "cmd":0x0004,      /* 消息类型，无符号16位整型 */
3   "res":0,           /* 回复消息类型，0:成功/确认；-1:失败；-2:消息有误；-3:不支持；*/
4   "bn":9120          /* 对应服务器端请求的流水号 */
5 }
```

遥控器控制发的附加功能指令：

▼ 遥控器控制发的附加功能JSON数据格式：

[JSON](#)[复制代码](#)

```
1 {
2   "cmd":0x0005,      /* 消息类型，无符号16位整型 */
3   "ts": 0,           /* 发送此消息的时间；整型数U64；格式为UNIX时间戳，是从1970年1月1日0点0分0秒开始所经过的毫秒数 */
4   "bn":9120,         /* 对应服务器端请求的流水号 */
5   "action": 1        /* 对应操作功能，1:清扫；2:扫盘上升；3:扫盘下降；0:无操作 */
6 }
```

车端状态反馈的附加功能结果：

▼ 车端状态反馈的配置信息JSON数据格式:

JSON | 复制代码

```
1 {  
2   "cmd":0x0005,      /* 消息类型，无符号16位整型 */  
3   "ts": 0,           /* 发送此消息的时间；整型数U64；格式为UNIX时间戳，是从1970年  
1月1日0点0分0秒开始所经过的毫秒数 */  
4   "bn":9120,         /* 对应服务器端请求的流水号 */  
5   "action": 1,       /* 对应操作功能，1:清扫；2:扫盘上升；3:扫盘下降；0:无操作 */  
6   "res":0            /* 回复消息类型，0:成功/确认；-1:失败；-2:消息有误；-3:不  
支持； */  
7 }
```