

交通标准协议



JAVA SDK 说明

目录

目录	i
1 使用说明	1
1.1 SDK 文件	1
1.2 使用前说明	1
2 功能接口	2
2.1 初始化 SDK	2
2.1.1 服务端和客户端模式使用场景对比（选择工作模式）	2
2.1.2 初始化客户端模式	2
2.1.3 初始化服务端模式	3
2.2 设置连接超时时间	5
2.3 设置读写超时时间	5
2.4 获取设备名称（终端 1.6.0.0401 及以上版本支持）	6
2.5 设置设备名称（终端 1.6.0.0401 及以上版本支持）	6
2.6 身份验证和安全通讯约定（终端 1.6.0.0401 及以上版本支持）	7
2.7 设置连接中使用 DES 加密密码	8
2.8 设置连接中使用 AES 加密密码	8
2.9 清除连接中使用的加密密码	9
2.10 发送播放列表（终端 1.6.0.0401 及以上版本支持）	9
2.11 发送局部更新（终端 1.6.0.0401 及以上版本支持）	10
2.12 移除局部更新（终端 1.6.0.0401 及以上版本支持）	11
2.13 发送文件（终端 1.6.0.0401 及以上版本支持）	12
2.14 手动亮度调节（终端 1.6.0.0401 及以上版本支持）	12
2.15 手动开关屏（终端 1.6.0.0401 及以上版本支持）	13
2.16 清理文件（终端 1.6.0.0401 及以上版本支持）	13
2.17 设置自动亮度控制参数（终端 1.6.0.0401 及以上版本支持）	14
2.18 获取自动亮度控制参数（终端 1.6.0.0401 及以上版本支持）	15
2.19 控制屏体电源开关（终端 1.6.0.0401 及以上版本支持）	15
2.20 控制本板电源开关（终端 1.6.0.0401 及以上版本支持）	16
2.21 控制多功能卡电源开关（终端 1.6.0.0401 及以上版本支持）	16
2.22 查询设备状态（终端 1.6.0.0401 及以上版本支持）	17
2.23 获取屏幕状态（终端 1.9.5.0401 及以上版本支持）	18
2.24 点检（终端 1.6.0.0401 及以上版本支持）	18

2.25 获取屏幕坏块信息（终端 1.6.0.0401 及以上版本支持）	19
2.26 获取监控卡数据（终端 1.6.0.0401 及以上版本支持）	19
2.27 设置日期时间（终端 1.6.0.0401 及以上版本支持）	22
2.28 设备重启（终端 1.6.0.0401 及以上版本支持）	22
2.29 设置高级定时任务（终端 1.9.5.0401 及以上版本支持）	23
2.30 获取高级定时任务（终端 1.9.5.0401 及以上版本支持）	24
2.31 移除高级定时任务（终端 1.9.5.0401 及以上版本支持）	24
2.32 恢复 IP 出厂设置（终端 1.6.0.0401 及以上版本支持）	25
2.33 查询版本信息（终端 1.6.0.0401 及以上版本支持）	25
2.34 设置屏体基本参数（终端 1.6.0.0401 及以上版本支持）	26
2.35 获取屏幕点阵数据（终端 1.6.0.0401 及以上版本支持）	27
2.36 设置音量大小（终端 1.6.0.0401 及以上版本支持）	27
2.37 获取音量大小（终端 1.9.5.0401 及以上版本支持）	28
2.38 获取内部存储空间大小（终端 1.9.3.0501 及以上版本支持）	28
2.39 设置时间时区（终端 1.6.0.0401 及以上版本支持）	29
2.40 获取时间时区（终端 1.6.0.0401 及以上版本支持）	30
2.41 设置 NTP 参数（终端 1.6.0.0401 及以上版本支持）	30
2.42 获取 NTP 参数（终端 1.6.0.0401 及以上版本支持）	31
2.43 设置高温关屏温度（终端 1.6.0.0401 及以上版本支持）	31
2.44 获取高温关屏温度（终端 1.6.0.0401 及以上版本支持）	32
2.45 设置网络中断后屏体播放策（虚连接）（终端 1.6.0.0401 及以上版本支持）	32
2.46 FTP 发送文件（终端 1.6.0.0401 及以上版本支持）	33
2.47 升级文件（终端 1.6.0.0401 及以上版本支持）	33
2.48 查询当前播放内容（终端 1.6.0.0401 及以上版本支持）	34
2.49 查询当前播放列表全部内容（终端 1.6.0.0401 及以上版本支持）	35
2.50 获取当前播放截图（终端 1.6.0.0401 及以上版本支持）	35
2.51 设置播放配置（终端 1.9.5.0401 及以上版本支持）	36
2.52 获取播放配置（终端 1.9.5.0401 及以上版本支持）	36
2.53 设置定时开关屏（终端 1.6.0.0401 及以上版本支持）	37
2.54 获取定时开关屏（终端 1.6.0.0401 及以上版本支持）	38
2.55 设置定时播放列表（终端 1.6.0.0401 及以上版本支持）	39
2.56 获取定时播放列表（终端 1.6.0.0401 及以上版本支持）	40
2.57 设置定时亮度（终端 1.6.0.0401 及以上版本支持）	41
2.58 获取定时亮度（终端 1.6.0.0401 及以上版本支持）	42
2.59 获取屏幕亮度信息（终端 1.9.3.0501 及以上版本支持）	43
2.60 获取所有播放列表（终端 1.9.5.0401 及以上版本支持）	43
2.61 设置定时音量控制参数（终端 1.6.0.0401 及以上版本支持）	44
2.62 获取定时音量控制参数（终端 1.6.0.0401 及以上版本支持）	45
2.63 删除指定播放列表文件（终端 1.9.3.0501 及以上版本支持）	45

2.64 下载媒体文件（终端 1.9.3.0501 及以上版本支持）	46
2.65 获取控制卡 MAC 地址（终端 1.9.5.0401 及以上版本支持）	47
2.66 获取设备 SN 号（终端 1.9.5.0401 及以上版本支持）	47
2.67 开关继电器状态（终端 1.9.3.0501 及以上版本支持）	48
2.68 主动获取设备状态（终端 1.9.5.0401 及以上版本支持）	48
2.69 更新新路网路况（终端 1.9.5.0401 及以上版本支持）	49
2.70 获取 CPU 和内存使用率（终端 1.9.3.0501 及以上版本支持）	50
2.71 获取文件列表（终端 1.9.3.0501 及以上版本支持）	51
3 播放接口	53
3.1 节目的分类	53
3.1.1 普通节目：	53
3.1.2 局部更新（高级播放）	54
3.1.3 Item 实时更新	55
3.2 普通节目	56
3.2.1 简单节目构建与播放	56
3.2.2 组合节目构建与播放	60
3.3 局部更新& item 实时更新	62
3.3.1 局部更新（高级播放）	62
3.3.2 item 实时更新	62
4 附录一：错误码和时区信息	64
4.1 错误码	64
4.2 时区信息	64
5 附录二：播放列表文件说明	68
5.1 文本媒体	71
5.2 扩展文本媒体	73
5.3 图片媒体	75
5.4 视频媒体	78
5.5 Gif 媒体	78
5.6 数字时钟媒体	79
5.7 模拟时钟媒体	82
5.8 Web 媒体	84
5.9 流媒体	84
5.10 表格	85
5.11 天气媒体	89
5.12 环境监测媒体	91
5.13 计时媒体	93
5.14 边框媒体	95
5.15 路网媒体	96

6 附录三：播放以及局部更新说明 99

1 使用说明

1.1 SDK 文件

SDK 编译环境	JDK 1.8 x86	
SDK 版本	SDK 文件	备注
V1.1 版本	nova.traffic.jar	无

1.2 使用前说明

目前此协议支持客户端和服务端两种方式连接。可创建线程多卡同时操作。

卡做客户端时，服务端可在公网搭建。

使用前请先确定 IP、端口、协议类型是否正确。

2 功能接口

2.1 初始化 SDK

SDK 初始化需要开发者先选择要使用的工作模式，不同的工作模式有不同的初始化方法及规则。

2.1.1 服务端和客户端模式使用场景对比（选择工作模式）

sdk 作为服务端和客户端的使用场景对比如下，开发者可根据实际情况选择对应的工作模式：

场景及功能	服务端模式	客户端模式
支持的网络环境	公网、局域网	局域网
连接方式	长链接	短链接
监听设备在线状态	支持	不支持
初始化参数	服务端 ip 和端口号（两个参数）	所有控制卡的 ip 和端口号
集成难易（繁琐）程度	一般	简单

2.1.2 初始化客户端模式

【功能描述】

初始化 NovaTraffic 类。建议在每次接口请求时都实例化一个新的 NovaTraffic 实例。

【接口定义】

```
public NovaTraffic(String ip, int port)
```

【输入参数】

参数	类型	说明	备注
ip	String	例如，192.168.0.220	控制卡 IP

参数	类型	说明	备注
port	int	例如，5000	控制卡端口

【返回参数】

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);
```

2.1.3 初始化服务端模式

【功能描述】

将 sdk 作为服务端的初始化操作。

【前提条件】

在进行连接前需要先将控制卡设置为客户端模式，并设置服务端 ip 和端口号。

【操作步骤】

1. 打开“JT Ark”，单击“协议/工作模式”，进入如图 2-1 所示界面。
2. 点击“控制卡工作模式”右侧的下拉箭头，选择“客户端”模式；
3. 填入服务端 ip 和端口号，最后单击“控制卡工作模式”下方的“应用”按钮

注意：单击“查询”按钮可查看当前控制卡工作模式参数。

图 2-1 交通配置工具



【初始化代码实现】

初始化并打开服务端通道，调用 setClientConnectListener 方法可设置客户端链接监听（非必须），当有设备连接上或者断开连接会回调相关函数。

```
ServerChannel serverChannel = new ServerChannel();
```



```
serverChannel.setClientConnectListener(new NovaDevice.ConnectListener() {  
    @Override  
    public void onDisconnect(NovaDevice client, String ip) {  
        //有客户端断开连接回调  
    }  
    @Override  
    public void onConnected(NovaDevice client, String ip) {  
        //有新的客户端连接成功回调  
    }  
    @Override  
    public void onReceivedHeartBeat(NovaDevice channel, String ip) {  
        //客户端心跳回调  
    }  
});  
boolean isOpen = serverChannel.open(serverIp, serverPort);  
if (isOpen) {  
    //服务端通道已开启  
} else {  
    //服务端通道开启失败  
}
```

参数	类型	说明	备注
serverIp	String	例如，192.168.0.100	服务端 IP
serverPort	int	例如，5000	服务端端口

注意：只能实例化一个 ServerChannel 实例，且只能打开一个通道（open 方法只能成功调用一次）。

【服务端接口请求】

1. 获取 NovaDevice 实例，调用 serverChannel 实例相关方法获取 NovaDevice 实例，具体有以下方法：

方法名	返回值类型	入参类型	说明
getDeviceByName	NovaDevice	String	根据设备名称获取一个 NovaDevice 实例

方法名	返回值类型	入参类型	说明
getAllDevices	List<NovaDevice>	无	获取所有 NovaDevice 实例

2. 获取 NovaTrafficServer 实例

```
NovaTrafficServer novaTraffic = novaDevice.obtainTrafficServer();
```

3. 接口请求，以“获取设备名称”为例，代码如下

```
if(novaDevice!= null && novaDevice.enable()) {  
    NovaTrafficServer novaTraffic = novaDevice.obtainTrafficServer();  
    String name = novaTraffic. getDeviceName(“设备名称”);  
    System.out.println(“设备名称: ”+name);  
}
```

【使用范例】

详见 sdkDemo;

注意：NovaTrafficServer 类和 NovaTraffic 类中的通讯接口名称、参数一致，区别在于一个用于服务端通信另一个用于客户端，以下接口示例多以 NovaTraffic 类为例。

2.2 设置连接超时时间

【功能描述】

设置连接超时时间。

【接口定义】

```
public void setConnectTimeout(int connectTimeout)
```

【输入参数】

参数	类型	说明	备注
connectTimeout	int	超时时间单位毫秒	

【返回参数】

无

2.3 设置读写超时时间

【功能描述】

设置读写超时时间。

【接口定义】

```
public void setSoTimeout(int connectTimeout)
```

【输入参数】

参数	类型	说明	备注
connectTimeout	int	超时时间单位毫秒	

【返回参数】

无

2.4 获取设备名称（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取设备的名称。

【接口定义】

```
public String getDeviceName()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
空	String	设备名称	UTF-8 编码

- null 失败

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
String name = novaTraffic.getDeviceName();
```

2.5 设置设备名称（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置设备的名称。

【接口定义】

```
public int setDeviceName(String name)
```

【输入参数】

参数	类型	说明	备注
name	String	设备名称	UTF-8 编码

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
String name = novaTraffic.setDeviceName(“Taurus50000346”);
```

2.6 身份验证和安全通讯约定(终端 1.6.0.0401 及以上版本支持)

【功能描述】

身份验证和安全通讯约定。JT100 1.7.0 及以后版本需填写 ViPlex Express 设备密码。

【接口定义】

```
public class DeviceSecretParam {  
    private String password;  
    private int enryptType;  
    private boolean isMD5Open;  
    private String newPassword;  
    private String oldPassword;  
}  
  
public int setDeviceSecretParam(DeviceSecretParam mDeviceSecretParam)
```

【输入参数】

参数	类型	说明	备注
password	String	设备密码	初始密码 123456
enryptType	int	0:不加密 1:DES 加密 2:AES 加密	
isMD5Open	boolean	发送文件接口是否采用 MD5 加密	
newPassword	String	新密钥	固定 8 位
oldPassword	String	旧密钥	固定 8 位(未加密的设备不需要旧密钥, 传空字符串)

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
DeviceSecretParam mDeviceSecretParam = new DeviceSecretParam(  
    "123456", 1, false , "12345678", "12345678");  
  
int ret = novaTraffic.setDeviceSecretParam(mDeviceSecretParam );
```

2.7 设置连接中使用 DES 加密密码

【功能描述】

NovaTraffic 中所有的消息采用 DES 加密的密码。

【接口定义】

```
public boolean novaTrafficSetPassword(String password)
```

【输入参数】

参数	类型	说明	备注
password	String	DES 密码	固定 8 位

【返回参数】

无

【使用范例】

```
DeviceSecretParam mDeviceSecretParam = new DeviceSecretParam(  
    "123456", 1, false , "12345678", "12345678");  
  
boolean ret = novaTraffic.setDeviceSecretParam(mDeviceSecretParam );  
  
System.out.println( "设置 des 加密结果: " +ret);  
  
novaTraffic.NovaTrafficSetPassword("12345678");
```

2.8 设置连接中使用 AES 加密密码

【功能描述】

NovaTraffic 中所有的消息采用 AES 加密的密码。

【接口定义】

```
public boolean novaTrafficSetAesPassword(String password)
```

【输入参数】

参数	类型	说明	备注
----	----	----	----

参数	类型	说明	备注
password	String	AES 密码	固定 8 位

【返回参数】

无

【使用范例】

```
DeviceSecretParam mDeviceSecretParam = new DeviceSecretParam(  
    "123456", 2, false , "12345678","12345678");  
  
boolean ret = novaTraffic.setDeviceSecretParam(mDeviceSecretParam);  
  
System.out.println( "设置 aes 加密结果: " +ret);  
  
novaTraffic.NovaTrafficSetAesPassword("12345678");
```

2.9 清除连接中使用的加密密码

【功能描述】

清除 NovaTraffic 连接中使用的加密密码。

【接口定义】

```
public void clearPassword()
```

【输入参数】

无

【返回参数】

无

【使用范例】

```
novaTraffic.clearPassword();
```

2.10 发送播放列表（终端 1.6.0.0401 及以上版本支持）

【功能描述】

发送播放文件。如播放需要附带图片、视频等资源文件。需使用下发文件接口提前发送。

【接口定义】

```
public int sendPlayList(int id, String content)
```

【输入参数】

参数	类型	说明	备注
id	int	下发播放列表的 ID（范围 1～100）	1 代表 play001.lst。相同 ID 会覆盖之前节目
content	String	播放文件的内容, 具体参考标准协议的播放格式说明	UTF-8 编码

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
String content =
"[all]\r\nitems=1\r\n[item1]\r\nparam=100,1,1,1,0,5,1,0,1\r\nimg1=0,0,001.jpg,0,0,0\r\nimgparam1=100";

int ret = novaTraffic.sendPlayList(1, content);
```

2.11 指定播放列表（终端 1.6.0.0401 及以上版本支持）

【功能描述】

指定播放列表。和下发文件（playXXX.lst）联动使用。

【接口定义】

```
public int playListById (int id)
```

【输入参数】

参数	类型	说明	备注
id	int	下发播放列表的 ID（范围 1～100）	1 代表 play001.lst。相同 ID 会覆盖之前节目

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.playListById(1);
```

2.12 发送局部更新（终端 1.6.0.0401 及以上版本支持）

【功能描述】

发送局部更新。如播放需要附带图片、视频等资源文件。需使用下发文件接口提前发送。

【接口定义】

```
public int sendLocalUpdate(int id, String content)
```

【输入参数】

参数	类型	说明	备注
id	int	下发局部更新的区域 ID（范围 1~10）	相同 ID 会覆盖之前的播放内容
content	String	播放文件的内容，具体参考标准协议的播放格式说明	UTF-8 编码

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String content =  
"[all]\r\nitems=1\r\n[item1]\r\nparam=100,1,1,1,0,5,1,0,1\r\nnimg1=0,0,001.jp  
g,0,0,0\r\nnimgparam1=100";  
  
int ret = novaTraffic.sendLocalUpdate (1, content);
```

2.13 移除局部更新（终端 1.6.0.0401 及以上版本支持）

【功能描述】

移除某个区域的局部更新。

【接口定义】

```
public int removeLocalUpdate (int id)
```

【输入参数】

参数	类型	说明	备注
id	int	移除局部更新的区域 ID（范围 1~10）	代表局部更新

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.removeLocalUpdate (1);
```


2.14 发送文件（终端 1.6.0.0401 及以上版本支持）

【功能描述】

发送本地某个资源文件。

【接口定义】

```
public int sendFile(File file, String saveFileName)
```

【输入参数】

参数	类型	说明	备注
file	File	本地文件资源	
saveFileName	String	保存到控制卡上的文件名	UTF-8

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.sendFile(new File("d:\\2.jpg"), "1.jpg");
```

2.15 手动亮度调节（终端 1.6.0.0401 及以上版本支持）

【功能描述】

手动亮度调节。

【接口定义】

```
public int setBrightness(int brightness)
```

【输入参数】

参数	类型	说明	备注
brightness	int	亮度等级	范围 0~255

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBrightness(130, "192.168.0.220", 5000);
```

2.16 手动开关屏（终端 1.6.0.0401 及以上版本支持）

【功能描述】

手动开关屏。

【接口定义】

```
public int setPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	<ul style="list-style-type: none">• True: 开屏• False: 关屏	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
int ret = novaTraffic.setPower(true);
```

2.17 清理文件（终端 1.6.0.0401 及以上版本支持）

【功能描述】

清理文件。

【接口定义】

```
public int cleanUpFiles(int type)
```

【输入参数】

参数	类型	说明	备注
type	int	<ul style="list-style-type: none">• 0: 清理所有媒体文件• 1: 清理无效媒体文件• 2: 清理所有文件• 3: 清理升级包	参数只有 0~3 为有效值，如传入其他 int 型数值，统一按参数 1 场景处理

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
int ret = novaTraffic. cleanUpFiles (1);
```

2.18 设置自动亮度控制参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置终端自动亮度控制参数。

【接口定义】

```
public class BrightnessItem{  
    private int environment;//环境亮度  
    private int screen;//屏体亮度  
}  
  
public int setBrightnessAuto(List<BrightnessItem> brightnessItem)
```

【输入参数】

List 必须为 8 组 或者传 null 清除。

参数	类型	说明	备注
environment	int	光探头采集的亮度值	0-255
screen	int	屏体的亮度值	0-255

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
List<BrightnessItem> brightnessItem = new ArrayList<BrightnessItem>();  
for(int i = 0 ; i < 8 ; i++) {  
    BrightnessItem item = new BrightnessItem(10 * i, 10 * i);  
    brightnessItem.add(item);  
}  
  
Int ret = novaTraffic.setBrightnessAuto(brightnessItem);
```

2.19 获取自动亮度控制参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

查询终端当前自动亮度控制参数。

【接口定义】

```
List<BrightnessItem> BrightnessItemList = novaTraffic.getBrightnessAuto()
```

【输入参数】

无。

【返回参数】

参数	类型	说明	备注
environment	int	光探头采集的亮度值	0-255
screen	int	屏体的亮度值	0-255

- null：错误

【使用范例】

```
List<BrightnessItem> BrightnessItemList = novaTraffic.getBrightnessAuto();
```

2.20 控制屏体电源开关（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置为关闭时，屏体先变黑，然后关闭本板电源。开屏为开启本板电源，屏体再变亮。

【接口定义】

```
public int setBoardPowerAndScreenPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	<ul style="list-style-type: none">• true 开• false 关	

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBoardPowerAndScreenPower(true);
```

2.21 控制本板电源开关（终端 1.6.0.0401 及以上版本支持）

【功能描述】

控制控制卡本板继电器开关。

【接口定义】

```
public int setBoardPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	<ul style="list-style-type: none">• true 开• false 关	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBoardPower(true);
```

2.22 控制多功能卡电源开关（终端 1.6.0.0401 及以上版本支持）

【功能描述】

控制多功能卡电源的开关。

【接口定义】

```
public int setMultiFunctionCardPower(int cardIndex, int roadIndex, boolean power)
```

【输入参数】

参数	类型	说明	备注
cardIndex	int	多功能卡位置索引	0-255
roadIndex	int	电源路数索引	0-7
power	boolean	true 开 false 关	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setMultiFunctionCardPower(0,0,true);
```

2.23 查询设备状态（终端 1.6.0.0401 及以上版本支持）

【功能描述】

查询设备状态。

【接口定义】

```
public class DeviceType {  
    private Date date;  
    private boolean isBoxDoorOpen;  
    private boolean isScreenPowerOpen;  
    private int temperature;  
    private boolean isInputSourceInside;  
    private int environmentBrightness;  
    private int brightnessType;  
    private int screenBrightness;  
}  
  
public DeviceType getDeviceType()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
date	Date	时间	
isBoxDoorOpen	boolean	箱门是否打开	
isScreenPowerOpen	boolean	屏体电源状态	
temperature	int	温度	摄氏度
isInputSourceInside	boolean	输入源	
environmentBrightness	int	当前环境亮度值	0-255
brightnessType	int	亮度控制方式	1-自动 2-手动 3-定时

参数	类型	说明	备注
screenBrightness	int	亮度	0-255

- null: 错误

【使用范例】

```
DeviceType mDeviceType = novaTraffic.getDeviceType();
```

2.24 获取屏幕状态（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取屏幕状态。

【接口定义】

```
public int getScreenStatus()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
status	int	1: 开屏; 2 关屏	

2.25 点检（终端 1.6.0.0401 及以上版本支持）

【功能描述】

点检。

【接口定义】

```
public int spotCheckBadBlockInit()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
Status	Int	1: 成功; 非 1 失败（详见错误码）	

2.26 获取屏幕坏块信息（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取屏幕坏块信息。

【接口定义】

```
public String getSpotCheckBadBlockInfo()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
badBlockInfo	String	坏点数据	

2.27 获取监控卡数据（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取监控卡数据。

【接口定义】

```
public String getSurveil()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
screenMonitorData	object	监控数据	
receiveCardMonitorInfo	object	接收卡的监控数据	
deviceWorkState	int	工作状态，正常 OK=0，异常 Error = 1，未知 Unknown = 2	
deviceMapList	object	接受卡的位置索引信息	
deviceIndex	int	设备所在位置编号，设备串联上的第几个	
deviceType	int	设备类型	
temprature	float	接收卡本板温度	

参数	类型	说明	备注
voltage	float	接收卡本板电压	
fpgaHardwareVersionInfo	string	接收卡 FPGA 固件版本信息	
fpgaVersion	string	接收卡 FPGA 版本号	
mcuHardwareVersionInfo	string	接收卡 MCU 固件版本信息	
mcuVersion	string	接收卡 MCU 版本号	
hardwareDeviceodel	int	硬件设备型号	
remarksVersionInfo	string	接收卡备注版本信息	
monitorCardMonitorInfo	object	监控卡的监控数据，如果没有连接监控卡，则监控数据为空	
deviceWorkState	int	工作状态， 正常 OK=0，异常 Error = 1，未知 Unknown = 2	
deviceMapList	object	设备的位置索引信息	
deviceIndex	int	设备所在位置编号，设备串联上的第几个	
deviceType	int	设备类型	
temperatureUpdateInfo	object	温度状态信息	
isUpdate	boolean	是否更新	
temprature	float	温度值，-255 表示无效温度	
humidityUpdateInfo	object	湿度状态信息	
isUpdate	boolean	是否更新	
humidity	float	湿度值	
smokeUpdateInfo	object	烟雾状态信息	
isUpdate	boolean	是否更新	
smoke	boolean	是否有烟雾	
cabinDoorUpdateInfo	object	箱门状态信息	
isUpdate	boolean	是否更新	
isDoorOpen	boolean	箱门是否打开	
fansUpdateInfo	object	风扇状态信息	
isUpdate	boolean	是否更新	

参数	类型	说明	备注
fansMonitorInfoCollection	HashMap	风扇转速，key 为第几路风扇，value 为风扇转速	
powerUpdateInfo	object	电源状态信息	
isUpdate	boolean	是否更新	
powerMonitorInfoCollection	HashMap	电压监控，key 为第几路电源，value 为电源电压，0 路为监控卡的电压，其他 1 到 8 是 8 路电源电压	
socketCableUpdateInfo	object	排线状态信息	
isUpdate	boolean	是否更新	
socketCableMonitorInfoList	object	排线信息	
deviceWorkState	int	工作状态，正常 OK=0，异常 Error = 1，未知 Unknown = 2	
deviceMapList	object	设备的位置索引信息	
deviceIndex	int	设备所在位置编号，设备串联上的第几个	
deviceType	int	设备类型	
socketCableInfoMap	HashMap	key 为 int，value 为 object List<SocketCableStatus>	
isCableOK	boolean	信号状态是否正常	
cableType	int	排线数据类型	
moduleUpdateInfo	object	模组状态信息，如果没有智能模组，则该项为空	
isUpdate	boolean	是否更新	
moduleMonitorInfoCollection	HashMap	电压监控，keyint，value 为 Object，ModuleStatus	
deviceWorkStatus	int	工作状态	
busIndex	int	flash 所在总线下标	
flashIndex	int	flash 下标	
rowIndex	int	行坐标	
colIndex	int	列坐标	
temperature	float	温度	

参数	类型	说明	备注
voltage	float	电压	
workTime	float	工作时间	

2.28 设置日期时间（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置设备的日期时间。

【接口定义】

```
public int setTime(Date date)
```

【输入参数】

参数	类型	说明	备注
date	Date	日期（年月日时分秒）	此功能不能更改时区

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setTime(new Date());
```

2.29 设备重启（终端 1.6.0.0401 及以上版本支持）

【功能描述】

重启设备。

【接口定义】

```
public int restartDevice()
```

【输入参数】

无

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.restartDevice();
```

2.30 设置高级定时任务（终端 1.9.5.0401 及以上版本支持）

【功能描述】

设置高级定时任务。

【接口定义】

public int setAdvanceTiming(String policy)

【输入参数】

参数	类型	说明	备注
policy	String	策略实体	<p>Json 格式:</p> <pre>{ "dataList": [//数据列表 { "policyData": "",//规则：每天：不需要填写 按星期执行：1-7（周日到周六） 按特殊日期执行：10-1,2,3（十月 1 日 2 日 3 日） "policyType": 0, // 规则类型：0：每天 1：按星期执行 2：特殊日期 "time": "00:00:00"//任务执行时间点 }, { "policyData": "6", "policyType": 1, "time": "00:00:00" }, { "policyData": "1-18", "policyType": 2, "time": "00:00:00" }], "effectiveDate": 946656000000,//生效时间 UTC 时间戳 "expirationDate": 407088000000,//失效时间 UTC 时间戳 "type": 5//策略类型 5：定时重启 }</pre>

【返回参数】

参数	类型	说明	备注
Status	Int	1: 成功; 非 1 失败 (详见错误码)	

2.31 获取高级定时任务（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取高级定时任务。

【接口定义】

```
public TimingPolicy getAdvanceTiming()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
type	Int	策略类型	
effectiveDate	long	生效日期	
expirationDate	long	失效日期	
dataList	List	策略	

dataList 实体:

参数	类型	说明	备注
policyType	Int	0:每天, 1: 按星期; 2: 特殊日期	
time	String	cron 表达式	
policyData	String	执行任务的参数	

2.32 移除高级定时任务（终端 1.9.5.0401 及以上版本支持）

【功能描述】

移除高级定时任务。

【接口定义】

```
public int removeAdvanceTiming()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
Status	Int	1: 成功; 非 1 失败 (详见错误码)	

2.33 恢复 IP 出厂设置（终端 1.6.0.0401 及以上版本支持）

【功能描述】

恢复设备 IP 到出厂 IP: 192.168.0.220。恢复 IP 出厂设置无回复，设置后连接中断。

【接口定义】

```
public void resetDeviceIpToDefault()
```

【输入参数】

无

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.resetDeviceIpToDefault();
```

2.34 查询版本信息（终端 1.6.0.0401 及以上版本支持）

【功能描述】

查询设备的软件版本信息。

【接口定义】

```
public String getDeviceVersion()
```

【输入参数】

无

【返回参数】

- null: 失败

【使用范例】

```
String verison = novaTraffic.getDeviceVersion();
```

2.35 设置屏体基本参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置设备的屏体网络连接参数，包括控制卡 IP 参数及其连接的上位机参数。当 IP 发生变化时会断开连接。

【接口定义】

```
class DeviceNetBasicParam {  
    private String deviceIp;  
  
    private int port;  
  
    private String mark;  
  
    private String gate;  
  
    private String serviceIp;  
  
}  
  
public void setDeviceNetBasicParam(DeviceNetBasicParam mDeviceBasicParam)
```

【输入参数】

参数	类型	说明	备注
deviceIp	String	设备 IP	例” 192.168.0.220”，当只需改 port 和 serviceIp 时传 deviceIp，mark、gate 可都为 null。
port	int	端口	控制卡做服务端时，为上位机连接控制卡所需端口号；控制卡做客户端时，为控制卡连接上位机所需端口 号。port 参数设置为 0 时，保持原有端口号不变。
mark	String	设备的子网掩码	
gate	String	设备的网关	
serviceIp	String	控制卡做客户端时，对应的服务器 IP	当控制卡服务端时无效，无需改变可单独为 null

【返回参数】

- 无

【使用范例】

```
DeviceNetBasicParam mDeviceNetBasicParam = new  
DeviceNetBasicParam("192.168.0.220", 5000,  
                    "255.255.255.0", "192.168.0.1", "192.168.0.103");  
  
novaTraffic.setDeviceNetBasicParam(mDeviceNetBasicParam);
```

2.36 获取屏幕点阵数据（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取显示屏像素宽度和高度。

【接口定义】

```
public class DeviceSize {  
    private int width;  
    private int height;  
}  
  
public DeviceSize getDeviceSize()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
width	int	显示屏像素宽度	
height	int	显示屏像素高度	

- null：失败

【使用范例】

```
DeviceSize mDeviceSize = novaTraffic.getDeviceSize();
```

2.37 设置音量大小（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置设备音量大小。

【接口定义】

```
public int setDeviceVolume(int volume)
```

【输入参数】

参数	类型	说明	备注
volume	int	音量大小	取值范围：0 - 100

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setDeviceVolume(50);
```

2.38 获取音量大小（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取音量大小。

【接口定义】

```
public int getDeviceVolume()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
volume	int	音量大小	取值范围：0 - 100

2.39 获取内部存储空间大小（终端 1.9.3.0501 及以上版本支持）

【功能描述】

获取内部存储空间的总大小和剩余大小。

【接口定义】

```
public class DeviceMemorySize {  
    private int totolSize;  
    private int remainSize;  
}  
  
public DeviceMemorySize getDeviceMemorySize()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
totolSize	int	总空间大小	单位 MB

参数	类型	说明	备注
remainSize	int	剩余空间大小	单位 MB

- null: 失败

【使用范例】

```
DeviceMemorySize mDeviceMemorySize = novaTraffic.getDeviceMemorySize();
```

2.40 设置时间时区（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置设备时间时区。

【接口定义】

```
public class TimeZoneParam {  
    private long utcTime;  
    private String timeZoneId;  
    private String gmt;  
}  
  
public int setTimeAndTimeZone(TimeZoneParam timeZoneParam)
```

【输入参数】

参数	类型	说明	备注
utcTime	long	UTC 时间	
timeZoneId	String	参考 4.2 时区信息	
gmt	String	参考 4.2 时区信息	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
TimeZoneParam mTimeZoneParam = new TimeZoneParam((long) (new  
Date().getTime()), "Asia/Shanghai", "GMT+08:00");  
  
int ret = novaTraffic.setTimeAndTimeZone(mTimeZoneParam);
```

2.41 获取时间时区（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取内部存储空间的总大小和剩余大小。

【接口定义】

```
public TimeZoneParam getTimeAndTimeZone()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
utcTime	long	UTC 时间	
timeZoneId	String	参考 4.2 时区信息	
gmt	String	参考 4.2 时区信息	

- null：失败

【使用范例】

```
TimeZoneParam mTimeZoneParam = novaTraffic.getTimeAndTimeZone();
```

2.42 设置 NTP 参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置控制卡 NTP 校时服务器参数。

【接口定义】

```
public class DeviceNTPParam {  
    private boolean isOpen;  
    private String addr;  
}  
  
public int setDeviceNTPParam(DeviceNTPParam deviceNTPParam)
```

【输入参数】

参数	类型	说明	备注
isOpen	long	是否开启 ntp 对时	
addr	String	Ntp 对时 url	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
DeviceNTPParam mDeviceNTPParam = new DeviceNTPParam(false, null);  
int ret = novaTraffic.setDeviceNTPParam(mDeviceNTPParam);
```

2.43 获取 NTP 参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取控制卡当前使用的 NTP 校时服务器参数。

【接口定义】

```
public DeviceNTPParam getDeviceNTPParam()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
isOpen	long	是否开启 ntp 对时	
addr	String	Ntp 对时 url	

- null: 失败

【使用范例】

```
DeviceNTPParam mDeviceNTPParam = novaTraffic.getDeviceNTPParam();
```

2.44 设置高温关屏温度（终端 1.6.0.0401 及以上版本支持）

【功能描述】

温度到达多少度后关屏。

【接口定义】

```
public int setCloseScreenTemperature(int value)
```

【输入参数】

参数	类型	说明	备注
value	int	关屏温度	系统最小值为 70, 0 表示不进行关屏处理

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setCloseScreenTemperature(80);
```

2.45 获取高温关屏温度（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取高温关屏温度。

【接口定义】

```
public int getCloseScreenTemperature()
```

【输入参数】

无

【返回参数】

小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.getCloseScreenTemperature();
```

2.46 设置网络中断后屏体播放策（虚连接）（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置网络在虚连接检测时间阈值内未收到任何数据时（包括心跳），播放指定节目的参数。

【接口定义】

```
public int setVirtualConnection(boolean isOpen,int secondTime, int id)
```

【输入参数】

参数	类型	说明	备注
isOpen	boolean	是否开启虚连接	
secondTime	int	秒	最低 60 秒，小于 60 秒按 60 秒计算

参数	类型	说明	备注
id	int	播放列表 id	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setVirtualConnection(false, 0, 1);
```

2.47 FTP 发送文件（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设备下载指定 URL FTP 地址的内容。不支持多文件发送。下载完毕回传后在发送另一个下载指令。

【接口定义】

```
public int sendFtpDownloadaddress(String url)
```

【输入参数】

参数	类型	说明	备注
url	String	FTP 文件存储路径	Utf-8 格式，举例： ftp://user:password@192.168.0.108:21/play001.lst

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret =  
novaTraffic.sendFtpDownloadaddress("ftp://user:password@192.168.0.103:21/UP  
DATE_OS_JT-TAURUSV010803CN0601.nuzip");
```

2.48 升级文件（终端 1.6.0.0401 及以上版本支持）

【功能描述】

通过发送文件或 FTP 发送文件传输升级 NUZIP 文件后，使用此接口传入文件名，升级 APP 或系统。

【接口定义】

```
public int updateFileName(String path)
```

【输入参数】

参数	类型	说明	备注
name	String	升级文件名称	Utf-8 格式, UPDATE_OS_JT-TAURUSV010803CN0601.nuzip

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret =  
novaTraffic.updateFileName("UPDATE_OS_JT-TAURUSV010803CN0601.nuzip");
```

2.49 查询当前播放内容（终端 1.6.0.0401 及以上版本支持）

【功能描述】

查询设备当前播放的内容。切换过程中可能会回复空内容。

【接口定义】

```
public class DeviceNowPlayItem {  
    private boolean isOpen;  
    private int playId;  
    private String content;  
}  
  
public DeviceNowPlayItem getNowPlayContent()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
isOpen	boolean	是否开屏	
playId	int	当前播放列表 id	id > 0
content	String	当前播放内容	

【使用范例】

```
DeviceNowPlayItem mDeviceNowPlayItem = novaTraffic.getNowPlayContent();
```

2.50 查询当前播放列表全部内容（终端 1.6.0.0401 及以上版本支持）

【功能描述】

查询当前播放列表的全部内容。

【接口定义】

```
public class DeviceNowPlayList {  
    private int playId;  
    private String content;  
}  
  
public DeviceNowPlayList getNowPlayAllContent()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
playId	int	当前播放列表 id	id > 0
content	String	当前播放内容	

【使用范例】

```
DeviceNowPlayList mDeviceNowPlayList = novaTraffic.getNowPlayAllContent();
```

2.51 获取当前播放截图（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取当前播放截图。

【接口定义】

```
public int getDeviceScreenshot(String filePath)
```

【输入参数】

参数	类型	说明	备注
filePath	String	截图文件保存路径	UTF-8 格式，举例： "d:\\2.jpg"

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.getDeviceScreenshot(filePath);
```

2.52 设置播放配置（终端 1.9.5.0401 及以上版本支持）

【功能描述】

设置播放配置。

【接口定义】

```
public int setPlayConfig(boolean isAntialias, boolean isTxtAdaptive)
```

【输入参数】

参数	类型	说明	备注
isAntialias	boolean	是否开启抗锯齿	
isTxtAdaptive	boolean	是否开启文本自适应	

【返回参数】

参数	类型	说明	备注
Status	Int	1: 成功; 非 1 失败（详见错误码）	

2.53 获取播放配置（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取播放配置。

【接口定义】

```
public byte[] getPlayConfig()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
config	Byte[]	config[0]: 1 开启抗锯齿; 0 未开启抗锯齿 config[1]: 1 开启文本自适应; 0 未开启文本	

参数	类型	说明	备注
		自适应	

2.54 设置定时开关屏（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置终端定时开关屏的控制参数。时间段内开，时间段外关。规则覆盖操作。传 null 清除。

【规则】

比如定时开关屏策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~ H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒 ~ K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class DevicePowerByTimeParam {
    private Date start;
    private Date end;
}

public int setDevicePowerByTimeList(List<DevicePowerByTimeParam> list)
```

【输入参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
String endStr = "2020.12.31 23:59:59";

SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");
```

```
sdf.setTimeZone(TimeZone.getDefault());

Date start = null;
Date end = null;

try {

    List<DevicePowerByTimeParam> mDevicePowerByTimeParamList = new
ArrayList<DevicePowerByTimeParam>();

    start = sdf.parse(startStr);

    end = sdf.parse(endStr);

    DevicePowerByTimeParam mDevicePowerByTimeParam = new
DevicePowerByTimeParam(start, end);

    mDevicePowerByTimeParamList.add(mDevicePowerByTimeParam);

    int ret =
novaTraffic.setDevicePowerByTimeList(mDevicePowerByTimeParamList);

} catch (ParseException e) {

    e.printStackTrace();

}
```

2.55 获取定时开关屏（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取定时开关屏参数。

【接口定义】

```
public List<DevicePowerByTimeParam> getDevicePowerByTimeList()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）

- null：失败

【使用范例】

```
List<DevicePowerByTimeParam> mDevicePowerByTimeParamList =  
novaTraffic.getDevicePowerByTimeList();
```

2.56 设置定时播放列表（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置终端定时播放的控制参数。规则覆盖操作。传 null 清除。

【规则】

比如定时播放策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒~K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class PlayByTimeParam {  
  
    private Date start;  
  
    private Date end;  
  
    private int id;  
  
}  
  
public int setPlayByTimeList(List<PlayByTimeParam> list)
```

【输入参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
Id	int	播放列表 id	1-100

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
```

```
String endStr = "2020.12.31 23:59:59";
```

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");  
sdf.setTimeZone(TimeZone.getDefault());  
Date start = null;  
Date end = null;  
try {  
    List<PlayByTimeParam> mPlayByTimeParamList = new  
ArrayList<PlayByTimeParam>();  
    start = sdf.parse(startStr);  
    end = sdf.parse(endStr);  
    PlayByTimeParam mPlayByTimeParam = new PlayByTimeParam(start, end, 1);  
    mPlayByTimeParamList.add(mPlayByTimeParam);  
    int ret = novaTraffic.setPlayByTimeList(mPlayByTimeParamList);  
} catch (ParseException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

2.57 获取定时播放列表（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取定时播放列表。

【接口定义】

```
public List<PlayByTimeParam> getPlayByTimeList()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
Id	int	播放列表 id	1-100

- null: 失败

【使用范例】

```
List<PlayByTimeParam> mPlayByTimeParamList = novaTraffic.getPlayByTimeList();
```

2.58 设置定时亮度（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置终端定时亮度的控制参数。规则覆盖操作。传 null 清除。

【规则】

比如定时亮度调节策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~ H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒 ~ K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class BrightnessByTimeParam {  
  
    Date start;  
  
    Date end;  
  
    int brightness;  
  
}  
  
public int setBrightnessByTimeList(List<BrightnessByTimeParam> list);
```

【输入参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
brightness	int	亮度值	0-255

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
String endStr = "2020.12.31 23:59:59";
SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");
sdf.setTimeZone(TimeZone.getDefault());
Date start = null;
Date end = null;
try {
    List<BrightnessByTimeParam> mBrightnessByTimeParamList = new
ArrayList<BrightnessByTimeParam>();
    start = sdf.parse(startStr);
    end = sdf.parse(endStr);

    BrightnessByTimeParam mBrightnessByTimeParam = new
BrightnessByTimeParam(start, end, 50);
    mBrightnessByTimeParamList.add(mBrightnessByTimeParam);
    ret = novaTraffic.setBrightnessByTimeList(mBrightnessByTimeParamList);
    tempStr = "设置定时亮度列表结果:";
    outTest += tempStr + ret + "\n";
    System.out.println(name + tempStr + ret);
} catch (ParseException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

2.59 获取定时亮度（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取定时亮度。

【接口定义】

```
public List<BrightnessByTimeParam> getBrightnessByTimeList()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
brightness	int	亮度值	0-255

- null：失败

【使用范例】

```
List<BrightnessByTimeParam> mBrightnessByTimeParamList =  
novaTraffic.getBrightnessByTimeList();
```

2.60 获取屏幕亮度信息（终端 1.9.3.0501 及以上版本支持）

【功能描述】

获取屏幕亮度信息。

【接口定义】

```
public BrightnessBean getBrightnessParam()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
brightnessBean	BrightnessBean	亮度信息	

BrightnessBean 结构：

参数	类型	说明	备注
brightMode	Int	亮度模式 1:自动 2:手动 3:定时	
brightness	String	亮度百分比	

【使用范例】

```
BrightnessBean bean = novaTraffic.getBrightnessParam();
```

2.61 获取所有播放列表（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取控制卡上的所有播放列表 id。

【接口定义】

```
public List<Integer> getAllPlaylistId()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
空	List<Integer>	所有播放列表的 id	-

- null: 失败

【使用范例】

```
List< Integer > list = novaTraffic. getAllPlaylistId ();
```

2.62 设置定时音量控制参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

设置控制卡定时音量参数。

【接口定义】

```
public int setTimingVolumeParam(List<VolumeByTimeParam> list);
```

【输入参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
volume	int	音量值	0-100

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

见 demo;

2.63 获取定时音量控制参数（终端 1.6.0.0401 及以上版本支持）

【功能描述】

获取控制卡定时音量参数。

【接口定义】

```
public List<VolumeByTimeParam> getTimingVolumeParam();
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	结束时间	定时功能的结束时间点（年月日时分秒）
volume	int	音量值	0-100

- null：失败

【使用范例】

```
List<VolumeByTimeParam> volumeByTimeParams =  
novaTraffic.getTimingVolumeParam();
```

2.64 删除指定播放列表文件（终端 1.9.3.0501 及以上版本支持）

【功能描述】

根据节目 id 删除对应的播放文件。

【接口定义】

```
public int deletePlaylistById(int id);
```

【输入参数】

参数	类型	说明	备注
id	int	节目 id	不能删除正在播放的节目

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
novaTraffic.deletePlaylistById(1);
```

2.65 下载媒体文件（终端 1.9.3.0501 及以上版本支持）

【功能描述】

从控制卡上下载文件到本地。

【接口定义】

```
public boolean getMediaFile(String savePath,String fileName,long timeOut);
```

【输入参数】

参数	类型	说明	备注
savePath	String	文件下载到本地的绝对路径	路径中应该包含文件名
fileName	String	控制卡上的媒体文件名称	-
timeOut	long	下载超时时间	单位 ms

【返回参数】

参数	类型	说明	备注
	boolean	true 成功, false 失败	

【调用步骤】

1、初始化文件服务：文件服务 FileServer.class 为单例模式，只需要初始化一次，初始化两个参数分别为“本地 ip”和“文件服务端口号”；使用方法见“使用范例”；

解释：媒体文件往往较大，为了提高文件传输速度，FileServer 会新开一个 socket 服务端，控制卡不使用协议传输的 socket，而是直接通过文件服务新开一个 socket 上传文件。

2、调用 getMediaFile 接口；

【使用范例】

```
//调用文件服务时必须初始化
if(!FileServer.getInstance().isInit()) {
    FileServer.getInstance().init("192.168.0.102", 8081);
}

boolean b;
```

```
b = novaTraffic.getMediaFile("F:\\JavaSdk\\bin\\1.gif", "test.gif", 10*1000);
```

2.66 获取控制卡 MAC 地址(终端 1.9.5.0401 及以上版本支持)

【功能描述】

获取控制卡 MAC 地址。

【接口定义】

```
public byte[] getMac ();
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
	byte[]	数组长度 6 位	返回 null 时失败

【使用范例】

```
byte[] mac = novaTraffic.getMac();
```

2.67 获取设备 SN 号（终端 1.9.5.0401 及以上版本支持）

【功能描述】

获取设备 SN 号。

【接口定义】

```
public String getSN();
```

【输入参数】

无。

【返回参数】

参数	类型	说明	备注
value	String	设备 SN 号	无

【使用范例】

```
String snCode = novaTraffic.getSN();
```

2.68 开关继电器状态（终端 1.9.3.0501 及以上版本支持）

【功能描述】

设置开关继电器状态。

【接口定义】

```
public int setRelayPowers (boolean isOpen1, boolean isOpen2, boolean isOpen3,
boolean isOpen4);
```

【输入参数】

参数	类型	说明	备注
isOpen1	boolean	<ul style="list-style-type: none">• true: 开• false: 关	01 继电器开关状态
isOpen2	boolean	<ul style="list-style-type: none">• true: 开• false: 关	02 继电器开关状态
isOpen3	boolean	<ul style="list-style-type: none">• true: 开• false: 关	03 继电器开关状态
isOpen3	boolean	<ul style="list-style-type: none">• true: 开• false: 关	04 继电器开关状态

【返回参数】

参数	类型	说明	备注
value	int	<ul style="list-style-type: none">• 1: 设置成功• 其他: 失败	无

【使用范例】

```
Int code = novaTraffic.setRelayPowers(true, true, true, true);
```

2.69 主动获取设备状态（终端 1.9.5.0401 及以上版本支持）

【功能描述】

主动获取设备在线状态。（仅在服务端模式下有效）

【接口定义】

```
public boolean enable()
```

【返回参数】

参数	类型	说明	备注
----	----	----	----

参数	类型	说明	备注
	boolean	<ul style="list-style-type: none"> • true: 在线 • false 离线 	

【调用步骤】

Java 服务端创建后，每个控制卡接入服务会创建一个 NovaDevice 对象。该对象提供了 enable 方法，可以主动获取 device 的在线状态。

```
boolean isOnLine = novaDevice.enable()
```

【获取方法】

```
NovaDevice novaDevice= serverChannel.getDeviceByName("Taurus-20000852");
if (novaDevice!= null && novaDevice.enable()) {
    System.out.println("客户端连接");
} else {
    System.out.println("客户端未连接");
}
```

2.70 更新新路网路况（终端 1.9.5.0401 及以上版本支持）

【功能描述】

刷新路况

【接口定义】

```
public void updateTJRoad(List<TJRout> routes)
```

【输入参数】

参数	类型	说明	备注
routes	List<TJRout>	一个 TJRoute 表示一段路的路况信息	

TJRout 结构

参数	类型	说明	备注
id	Int32 范围	路线 id	
startPercent	Float	起始位置（路长的百分比），0-1	
endPercent	Float	结束位置（路长的百分比），0-1	
width	Int32	路宽度	
color		1-红色	

参数	类型	说明	备注
		2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 16 进制定义颜色	

【返回参数】

无

【调用示例】

```
List<TJRoute> list = new ArrayList<>();
TJRoute tjRoute1 = new TJRoute.TJRouteBuild().Id(1).color("1").
    startPercent(0.1f).endPercent(0.6f).usePercent().build();
TJRoute tjRoute2 = new TJRoute.TJRouteBuild().Id(2).color("1").
    startPercent(0.1f).endPercent(0.1f).usePercent().build();
TJRoute tjRoute3 = new TJRoute.TJRouteBuild().Id(3).color("7").
    startPercent(0.3f).endPercent(0.6f).usePercent().build();
TJRoute tjRoute5= new TJRoute.TJRouteBuild().Id(0).color("FFF000").
    startPercent(0.7f).endPercent(0.85f).usePercent().build();
list.add(tjRoute1);
list.add(tjRoute2);
list.add(tjRoute3);
list.add(tjRoute5);
novaTraffic.updateTJRoad(list);
```

【参数示例】

[{"id":1,"startPercent":0.1,"endPercent":0.6,"width":0,"color":"1"}, {"id":2,"startPercent":0.1,"endPercent":0.1,"width":0,"color":"1"}, {"id":3,"startPercent":0.3,"endPercent":0.6,"width":0,"color":"7"}, {"id":0,"startPercent":0.7,"endPercent":0.85,"width":0,"color":"FFF000"}]

2.71 获取 CPU 和内存使用率（终端 1.9.3.0501 及以上版本支持）

【功能描述】

获取设备当前的 CPU 和内存使用率。

【接口定义】

CpuAndMemoryInfo getCpuAndMemoryInfo();

【输入参数】

无

【返回参数】

参数	类型	说明	备注
code	int	错误码	1: 成功; 小于 1: 失败, 参考附录 1
cpu	int	cpu 占用率	
totalMemory	int	内存总大小 (单位 B)	
freeMemory	int	内存剩余大小 (单位 B)	

2.72 获取文件列表（终端 1.9.3.0501 及以上版本支持）

【功能描述】

获取设备存储的指定类型文件列表。

【接口定义】

public List<String> getFileListName(int type) ;

【输入参数】

参数	类型	说明	备注
type	int	文件类型	0: 播放列表 1: 局部更新 2: 媒体文件 3: 运行日志

【返回参数】

参数	类型	说明	备注
fileList	List<String>	文件名列表	

2.73 接收卡固化（终端 1.13.0 及以上版本支持）

【功能描述】

接收卡数据固化

【接口定义】


```
public int solidify(byte type);
```

【输入参数】

参数	类型	说明	备注
type	byte	固化类型	0: 接收卡所有地址数据固化 1: 接收卡参数 2: Flash 拓扑 3: Gamma 表 4: 数据组交换 RAM 和 ICN2055 功能 RAM 5: 精细灰度校正数据 6: 大表 7: 颜色管理数据 8: 除大表外其他数据 9: 灯板走线表

【返回参数】

参数	类型	说明	备注
code	int	错误码	

3 播放接口

诺瓦交通协议的节目在 sdk 和标准协议中被称为“播放列表”，其含义为多个播放页面的集合（一个 item 为一页），sdk 中封装了构建一个播放列表所必须的接口。了解以下定义有助于快速准确的构建一个播放列表。

- **item**: 节目播放时以一个 item 为一页进行轮流切换，item 中需至少包含一个媒体，一个播放列表中至少包含一个 item。
- **多个 item 如何切换**: 当一个 item 播放完成时会自动切换到下一个 item，且循环播放。

3.1 节目的分类

诺瓦交通协议的节目共分为三类：普通节目、局部更新、item 实时更新。

3.1.1 普通节目：

- 可以创建一个 item 页面，适用于播放媒体的静态展示。
- 可以创建多个 item 页面，适用于多个 item 页面进行轮流播放。

播放参考 3.2。

playBuild、ItemBuild、MediaBuild 三个对象组装媒体协议。

mediaBuild 负责构建单个媒体 build（文本、图片、视频等），构建一个简单的文本媒体，如下所示。

```
TextMediaBuilder mediaBuilder = new TextMediaBuilder();  
mediaBuilder.text(“普通文本”);
```

上面代码创建了一个简单的文本媒体文件，只需要构造一个文本参数即可，开发者可以使用 mediaBuild 组装更多媒体参数，如下所示。

```
mediaBuilder  
    .x(10) //x 坐标
```

```
.y(10) //y 坐标  
.width(200) // 宽度  
.height(200) //高度  
.textColor("4") //字符颜色  
.backGroundColor("2") //背景颜色  
.text("普通文本") //文本
```

MediaBuild 可以创建多个，组装到一个或者多个 itemBuild 中（将媒体添加到页面上）。ItemBuild 也可以创建多个，组装到 PlayBuild 中。注意需要给 playBuild 设置节目 Id。

```
//创建一个 item build（页面）  
ItemMediaBuilder itemBuild = new ItemMediaBuilder();  
//创建一个 playBuild（节目）  
PlayBuilder playBuild = new PlayBuilder();  
  
//设置节目 id，将媒体添加到 item（页面），将页面组装到播放节目中  
  
playBuild.Id(5).addBuilderItem(itemBuild.addBuilderMedia(mediaBuilder));
```

播放协议已经组装完成，PlayManager 提供了播放方法，调用方式如下所示。

```
PlayManager playManage= new PlayManager (playBuild, novaTraffic);  
//播放节目  
playManage.play()
```

注：如果创建的媒体文件存在图片、视频等媒体文件，play() 方法会先上传媒体文件，再播放内容，无需单独上传媒体文件。

3.1.2 局部更新（高级播放）

局部更新附在所有页面之上，局部更新会新建播放器，有多个局部更新 item 的时候，和播放列表一样循环播放。

在普通节目之上可以创建 10 个局部更新节目，局部更新的节目会展示在普通节目的上层。每个局部更新都有自己的切换逻辑，局部更新的内容不会随着普通节目的 item 切换而切换。

比如停车诱导只需要更新车位数，可使用局部更新显示车位变化，避免其他 item 展示信息切换，导致车位显示不可见。

```
//构建节目和普通节目一致  
.....  
PlayManager playManage= new PlayManager (playBuild, novaTraffic);  
//播放节目  
playManage.regionUpdate()
```

局部更新和普通节目参数构造一致，只是发送的命令不一致，需要调用 `regionUpdate` 方法实现局部更新。

	播放 item1	播放 item2
局部更新前		
局部更新后		

3.1.3 Item 实时更新

当需要更新普通 item 中的某一块区域时，可使用实时更新。普通节目切换，实时更新的内容也会随着 item 的切换而切换。

```
//构建节目和普通节目一致
.....
实时更新需要单独设置更新的 Item 页面
playBuild.Id(2)
    .addBuilderItem(itemBuild1
        .duration(20)
        .addBuilderMedia(imageBuild1)
        .addBuilderMedia(imageBuild2)
        .itemIndex(1));
PlayManager playManage= new PlayManager (playBuild, novaTraffic);
//播放节目
playManage.liveUpdating ()
```

item 实时更新只能同时更改一条 item 数据,需要多个 item 更新内容的时候需要下发多次命令,调用的方法是 liveUpdating。

item 实时更新需要单独设置 itemIndex,标识更新的是哪个 item,例如更新普通节目 item2,则 itemIndex=2。

例如:医院叫号系统,张三到 2 号窗口,当张三进入诊断室后,节目更新成李四到 2 号窗口,姓名是会随着医生诊断结果随时更新,但是整个叫号窗口是不发生变化的,这种场景选择 item 局部更新。

3.2 普通节目

3.2.1 简单节目构建与播放

一个播放列表只包含一个 item,并且这个 item 上只存在一个媒体。使用普通节目接口播放。适用于播放媒体的静态展示,无需多 item 轮流切换的场景。

文本媒体

【媒体构建】

```
TextMedia.Builder mediaBuilder = new TextMedia.Builder();
```

可以使用基础播放,构造一个文本参数即可,如下:

```
mediaBuilder.text("普通文本");
```

【播放示例】

```
//创建一个 playBuild (节目)
PlayBuilder playBuild = new PlayBuilder();
//创建一个 item build (页面)
ItemMediaBuilder itemBuild = new ItemMediaBuilder();
//创建文本媒体 build
TextMediaBuilder mediaBuilder = new TextMediaBuilder();
//添加媒体参数,详细参数,也可直接构造一个简易参数 text 即可
mediaBuilder
    .x(10) //x 坐标
    .y(10) //y 坐标
    .width(200) // 宽度
    .height(200) //高度
    .textColor("4") //字符颜色
    .backgroundColor("2") //背景颜色
    .text("普通文本") //文本
//设置节目 id, 将媒体添加到 item (页面), 将页面组装到播放节目中
```

```
playBuild.Id(5).addBuilderItem(itemBuild.addBuilderMedia(mediaBuilder));

//创建播放管理器，该构造器需要两个参数
    //@playBuild 节目 build 对象
    //@novaTraffic 诺瓦客户端/服务器对象
PlayManager playManage= new PlayManager (playBuild, novaTraffic);

//标准播放
playManage.play();
```

扩展文本媒体

扩展文本比普通文本媒体增加了一些扩展参数，比如语音播放、走马灯效果等。

【媒体构建】

```
TextPlusBuilder mediaBuilder = new TextPlusBuilder();
```

可以使用基础播放，构造一个文本参数即可，如下：

```
mediaBuilder.text(text);
```

【播放示例】

```
//创建一个 playBuild（节目）
PlayBuilder playBuild = new PlayBuilder();

//创建一个 item build（页面）
ItemMediaBuilder itemBuild = new ItemMediaBuilder();
//创建文本媒体 build
TextPlusBuilder mediaBuilder = new TextPlusBuilder();
//完全自定义播放内容(使用中构造自己需要的参数即可)
mediaBuilder
    .x(0)
    .y(0)
    .width(0)
    .height(0)
    .textColor("4") //文字颜色
    .backgroundColor("2") //背景颜色
    .text(text) //文字显示
    .wordSpace(10) //文字间距
    .font("2") //字体类型
    .fontStyle(8) //
```

```
.textSize(30) //文字大小
.alignmentH("1")
.alignmentV("1") //对齐方式
.effect("12") //特效
.speed("4"); //特效速度
//设置节目 id, 将媒体添加到 item (页面), 将页面组装到播放节目中
playBuild.Id(5).addBuilderItem(itemBuild.addBuilderMedia(mediaBuilder));

//创建播放管理器, 该构造器需要两个参数
//@playBuild 节目 build 对象
//@novaTraffic 诺瓦客户端/服务器对象
PlayManager playManage= new PlayManager (playBuild, novaTraffic);

//标准播放
playManage.play();
```

图片媒体

【媒体构建】

```
ImageMediaBuilder imageBuild = new ImageMediaBuilder()
```

可以使用基础播放, 构造一个文本参数即可。直接替换示例代码中的红色标注部分。

```
imageBuild.filePath("C:\\Users\\Pictures\\image\\2.jpg")
```

【播放示例】

```
//创建一个 playBuild (节目)
PlayBuilder playBuild = new PlayBuilder();

//创建一个 item build (页面)
ItemMediaBuilder itemBuild = new ItemMediaBuilder();

//创建一个 image 的 build 对象
ImageMediaBuilder mediaBuilder = new ImageMediaBuilder();
//通过 build 构建 image 媒体参数
mediaBuilder
    .filePath("C:\\Users\\ Pictures\\image\\2.jpg")
    .x(10)
    .y(10)
    .width(100)
```

```
        .height(100);  
        //设置节目 id, 将媒体添加到 item (页面), 将页面组装到播放节目中  
  
playBuild.Id(5).addBuilderItem(itemBuild.addBuilderMedia(mediaBuilder));  
  
//创建播放管理器, 该构造器需要两个参数  
        //@playBuild 节目 build 对象  
        //@novaTraffic 诺瓦客户端/服务器对象  
PlayManager playManage= new PlayManager(playBuild, novaTraffic);  
  
//标准播放  
        playManage.play();
```

视频媒体

【媒体构建】

```
VideoMediaBuilder mediaBuilder = new VideoMediaBuilder();
```

可以使用基础播放, 构造一个文本参数即可, 如下:

```
mediaBuilder.filePath("C:\\Users\\Pictures\\image\\video.mp4");
```

【播放示例】

```
        //创建一个 playBuild (节目)  
        PlayBuilder playBuild = new PlayBuilder();  
  
//创建一个 item build (页面)  
        ItemMediaBuilder itemBuild = new ItemMediaBuilder();  
  
//创建一个 Video 的 build 对象  
        VideoMediaBuilder mediaBuilder = new VideoMediaBuilder();  
//通过 build 构建视频媒体参数  
mediaBuilder  
        .filePath("C:\\Users\\Pictures\\image\\video.mp4")  
        .x(10)  
        .y(150)  
        .width(100)  
        .height(100);  
        //设置节目 id, 将媒体添加到 item (页面), 将页面组装到播放节目中  
  
playBuild.Id(5).addBuilderItem(itemBuild.addBuilderMedia(mediaBuilder));
```



```
//创建播放管理器，该构造器需要两个参数
    //@playBuild 节目 build 对象
    //@novaTraffic 诺瓦客户端/服务器对象
PlayManager playManage= new PlayManager(playBuild, novaTraffic);

//标准播放
playManage.play();
```

其他媒体

GIF、数字时钟、表格媒体的构建方法类似，具体实现可关注 PlayTest.java 方法。

3.2.2 组合节目构建与播放

组合节目指的是一个播放列表有多个 item 页面，每个 item 页面上显示多个相同或不同的媒体，适合轮流播放的节目展示。

```
//创建一个 playBuild（节目）
PlayBuilder playBuild = new PlayBuilder();

//创建三个 item build（页面）
ItemMediaBuilder itemBuild1 = new ItemMediaBuilder();
ItemMediaBuilder itemBuild2 = new ItemMediaBuilder();
ItemMediaBuilder itemBuild3 = new ItemMediaBuilder();

//创建两个 image 媒体的 build 对象
ImageMediaBuilder imageBuild1 = new ImageMediaBuilder();
ImageMediaBuilder imageBuild2 = new ImageMediaBuilder();
imageBuild1
    .filePath("C:\\Users\\ Pictures\\image\\2.jpg")
    .x(10)
    .y(10)
    .width(100)
    .height(100);
imageBuild2
    .filePath("C:\\Users\\ Pictures\\image\\1.jpg")
    .x(10)
    .y(150)
    .width(100)
    .height(100);
```

```
//创建一个视频媒体的 build 对象
VideoMediaBuilder videoBuild = new VideoMediaBuilder();
videoBuild
    .filePath("C:\\Users\\ Pictures\\image\\video.mp4")
    .x(10)
    .y(150)
    .width(100)
    .height(100);

//创建一个 Gif 媒体的 build 对象
GifMedia.Builder gifBuild = new GifMedia.Builder();
gifBuild
    .filePath("C:\\Users\\ Pictures\\image\\3.gif")
    .x(10)
    .y(150)
    .width(100)
    .height(100);
playBuild.Id(10) //设置节目的 id
//添加第一个 item 页面信息
.addBuilderItem(itemBuild1
    .duration(20) //设置播放时间
    //item 添加两个媒体对象
    .addBuilderMedia(imageBuild1)
    .addBuilderMedia(imageBuild2))
//添加第二个 item 页面信息
.addBuilderItem(itemBuild2
    .addBuilderMedia(videoBuild)
    .duration(20))
//添加第三个 item 页面信息
.addBuilderItem(itemBuild3
    .addBuilderMedia(gifBuild)
    .duration(10));
PlayManager playManage= new PlayManager(playBuild, novaTraffic);
//标准播放
playManage.play();
```

3.3 局部更新& item 实时更新

3.3.1 局部更新（高级播放）

【播放示例】

```
public static int sendRegionUpdate(NovaTrafficImp novaTraffic) {  
    //创建一个 playBuild（节目）  
    PlayBuilder playBuild = new PlayBuilder();  
    //创建 item build（页面）  
    ItemMediaBuilder itemBuild1 = new ItemMediaBuilder();  
    //创建一个 image 的 build 对象  
    ImageMediaBuilder imageBuild1 = new ImageMediaBuilder();  
    //通过 build 构建 image 媒体参数  
    imageBuild1  
        .filePath("C:\\\\Users\\Nova004157\\Pictures\\image\\2.jpg")  
        .x(10)  
        .y(10)  
        .width(50)  
        .height(100);  
    PlayManager playManage = new PlayManager (playBuild, novaTraffic);  
    playBuild  
        .Id(1)  
        .addBuilderItem(itemBuild1  
            .duration(20)  
            .addBuilderMedia(imageBuild1);  
    playManage.regionUpdate();  
}
```

3.3.2 item 实时更新

```
//创建一个 playBuild（节目）  
PlayBuilder playBuild = new PlayBuilder();  
//创建一个 item build（页面）  
ItemMediaBuilder itemBuild1 = new ItemMediaBuilder();  
//创建两个 image 的 build 对象  
ImageMediaBuilder imageBuild1 = new ImageMediaBuilder();  
ImageMediaBuilder imageBuild2 = new ImageMediaBuilder();  
//通过 build 构建 image 媒体参数  
//通过 build 构建 image 媒体参数
```

```
imageBuild1
    .filePath("C:\\Users\\Nova004157\\Pictures\\image\\2.jpg")
    .x(10)
    .y(10)
    .width(50)
    .height(50);
imageBuild2
    .filePath("C:\\Users\\Nova004157\\Pictures\\image\\1.jpg")
    .x(10)
    .y(150)
    .width(50)
    .height(50);
playBuild.Id(2)
    .addItem(itemBuild1
        .duration(20)
        .addBuilderMedia(imageBuild1)
        .addBuilderMedia(imageBuild2)
        .itemIndex(1)/*实时更新需要单独设置该参数，标识更新的是那个 item*/);
PlayManage playManage = new PlayManage (playBuild, novaTraffic);
//实时更新
playManage.liveUpdating();
```

4 附录一：错误码和时区信息

4.1 错误码

参数名	说明
-1	找不到 IP 对应主机
-2	I0 错误
-3	文件名发送未收到回复
-4	文件名接收到错误
-5	文件块发送未收到回复
-6	文件块接收到错误
-7	文件块未收到结束标识
-8	文件块结束标识接受错误
-9	指定播放列表未收到回复
-10	指定播放列表接收错误
-11	未收到回复
-12	接收错误
-13	文件不存在
-14	SOCKET 错误
-15	传入参数错误

4.2 时区信息

时区 ID	时区 GMT 偏移	时区中文备注
-------	-----------	--------

时区 ID	时区 GMT 偏移	时区中文备注
Pacific/Majuro	GMT+12:00	马朱罗
Pacific/Midway	GMT-11:00	中途岛
Pacific/Honolulu	GMT-10:00	檀香山
America/Anchorage	GMT-09:00	安克雷奇
America/Los_Angeles	GMT-08:00	洛杉矶/美国太平洋
America/Tijuana	GMT-08:00	提华纳/美国太平洋
America/Phoenix	GMT-07:00	凤凰城美国山区
America/Chihuahua	GMT-07:00	奇瓦瓦
America/Denver	GMT-07:00	丹佛/美国山区
America/Costa_Rica	GMT-06:00	哥斯达黎加/美国中部
America/Chicago	GMT-06:00	芝加哥/美国中部
America/Mexico_City	GMT-06:00	墨西哥城/美国中部
America/Regina	GMT-06:00	里贾纳/美国中部
America/Bogota	GMT-05:00	波哥大/哥伦比亚
America/New_York	GMT-05:00	纽约/美国东部
America/Caracas	GMT-04:30	加拉加斯/委内瑞拉
America/Barbados	GMT-04:00	巴巴多斯/大西洋
America/Manaus	GMT-04:00	玛瑙斯/亚马逊
America/Santiago	GMT-03:00	圣地亚哥
America/St_Johns	GMT-03:30	圣约翰/纽芬兰
America/Sao_Paulo	GMT-03:00	圣保罗
America/Argentina/Buenos_Aires	GMT-03:00	布宜诺斯艾利斯
America/Godthab	GMT-03:00	戈特霍布
America/Montevideo	GMT-03:00	蒙得维的亚/乌拉圭
Atlantic/South_Georgia	GMT-02:00	南乔治亚
Atlantic/Azores	GMT-01:00	亚述尔群岛
Atlantic/Cape_Verde	GMT-01:00	佛得角
Africa/Casablanca	GMT+00:00	卡萨布兰卡
Europe/London	GMT+00:00	伦敦/格林尼治

时区 ID	时区 GMT 偏移	时区中文备注
Europe/Amsterdam	GMT+01:00	阿姆斯特丹/中欧
Europe/Belgrade	GMT+01:00	贝尔格莱德/中欧
Europe/Brussels	GMT+01:00	布鲁塞尔/中欧
Europe/Sarajevo	GMT+01:00	萨拉热窝/中欧
Africa/Windhoek	GMT+01:00	温得和克
Africa/Brazzaville	GMT+01:00	布拉扎维/西部非洲
Asia/Amman	GMT+02:00	安曼/东欧
Europe/Athens	GMT+02:00	雅典/东欧
Asia/Beirut	GMT+02:00	贝鲁特/东欧
Africa/Cairo	GMT+02:00	开罗/东欧
Europe/Helsinki	GMT+02:00	赫尔辛基/东欧
Asia/Jerusalem	GMT+02:00	耶路撒冷/以色列
Europe/Minsk	GMT+03:00	明斯克
Africa/Harare	GMT+02:00	哈拉雷/中部非洲
Asia/Baghdad	GMT+03:00	巴格达
Europe/Moscow	GMT+03:00	莫斯科
Asia/Kuwait	GMT+03:00	科威特
Africa/Nairobi	GMT+03:00	内罗毕/东部非洲
Asia/Tehran	GMT+03:30	德黑兰/伊朗
Asia/Baku	GMT+04:00	巴库
Asia/Tbilisi	GMT+04:00	第比利斯
Asia/Yerevan	GMT+04:00	埃里温
Asia/Dubai	GMT+04:00	迪拜
Asia/Kabul	GMT+04:30	喀布尔/阿富汗
Asia/Karachi	GMT+05:00	卡拉奇
Asia/Oral	GMT+05:00	乌拉尔
Asia/Yekaterinburg	GMT+05:00	叶卡捷林堡
Asia/Calcutta	GMT+05:30	加尔各答
Asia/Colombo	GMT+05:30	科伦坡
Asia/Katmandu	GMT+05:45	加德满都/尼泊尔

时区 ID	时区 GMT 偏移	时区中文备注
Asia/Almaty	GMT+06:00	阿拉木图
Asia/Rangoon	GMT+06:30	仰光/缅甸
Asia/Krasnoyarsk	GMT+07:00	克拉斯诺亚尔斯克
Asia/Bangkok	GMT+07:00	曼谷
Asia/Shanghai	GMT+08:00	北京/中国
Asia/Hong_Kong	GMT+08:00	香港/中国
Asia/Irkutsk	GMT+08:00	伊尔库茨克
Asia/Kuala_Lumpur	GMT+08:00	吉隆坡
Australia/Perth	GMT+08:00	佩思
Asia/Taipei	GMT+08:00	台北时间
Asia/Seoul	GMT+09:00	首尔
Asia/Tokyo	GMT+09:00	东京/日本
Asia/Yakutsk	GMT+09:00	雅库茨克
Australia/Adelaide	GMT+09:30	阿德莱德
Australia/Darwin	GMT+09:30	达尔文
Australia/Brisbane	GMT+10:00	布里斯班
Australia/Hobart	GMT+10:00	霍巴特
Australia/Sydney	GMT+10:00	悉尼
Asia/Vladivostok	GMT+10:00	符拉迪沃斯托克/海参崴
Pacific/Guam	GMT+10:00	关岛
Asia/Magadan	GMT+10:00	马加丹
Pacific/Auckland	GMT+12:00	奥克兰
Pacific/Fiji	GMT+12:00	斐济
Pacific/Tongatapu	GMT+13:00	东加塔布

5

附录二：播放列表文件说明

设备共支持播放列表1-100。文件名为:play001.lst-play100.lst。保存编码格式为UTF8, 每个播放文件内最多有24个显示条目。格式如下:

[all]

items=总条目数

[item1]

param=停留时间, 入屏方式, 出屏方式, 入屏速度, 闪烁速度, 闪烁次数, 播放次数

period=item有效期起始时间[yyyy-MM-dd], item有效期结束时间[yyyy-MM-dd], item有效星期[1234567, 1为星期天, 7为星期六], 有效期每天的时间段[起点-结束点/起点-结束点 “注: 时间格式为HH:mm:ss”];

举例: period=2020-9-17, 2020-10-18,
123, 10:00:00-10:10:00/10:30:00-11:00:00;

表示: 在2020年9月17日到2020年10月18日之间如果这一天是周日、周一或者周二, 则每天的10点整到10点10分以及10点30分到11点整此item都是有效的;

注意: 一天之内支持多个有效时间段, 时间段之间用“/”隔开; 支持多条有效期, 每条数据之间用“;”隔开。

说明: item有效期, 针对的是item切换时间而定义。当item切换时, 如果当前时间在item有效期内, 则item可播放。有效期截止时如果当前item未播放结束, 此item仍可以完成本次播放。

img1=x 坐标, y 坐标, 文件名称, 闪烁, 显示区域宽度, 显示区域高度

imgparam1=停留时间

.....

imgN=x 坐标, y 坐标, 文件名称, 闪烁, 显示区域宽度, 显示区域高度

imgparamN=停留时间, 占位符 0, 动画类型, 播放次数, 动画时长(单位 100ms)

txt1=x 坐标, y 坐标, 字体, 字号, 前景颜色, 背景颜色, 闪烁, 文字内容, 显示区域宽度, 显示区域高度, 字体风格

txtparam1=字符间距, 排列方式

.....

txtN=x 坐标, y 坐标, 字体, 字号, 前景颜色, 背景颜色, 闪烁, 文字内容, 显示区域宽度, 显示区域高度, 字体风格

txtparamN=字符间距, 排列方式

video1=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放次数

.....

videoN=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放次数

webview1=x 坐标, y 坐标, URL, 保留位默认填入 0, 显示区域宽度, 显示区域高度

.....

webviewN=x 坐标, y 坐标, URL, 保留位默认填入 0, 显示区域宽度, 显示区域高度

stream1=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, URL, 播放次数

.....

streamN= x 坐标, y 坐标, 显示区域宽度, 显示区域高度, URL, 播放次数

gif1=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放时长, 播放次数, 背景颜色

.....

gifN= x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放时长, 播放次数, 背景颜色

time1=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 播放时长, 前景颜色, 背景颜色, 字号, 字体, 字体风格, 是否显示四位年, 日期显示格式, 是否显示年, 月, 日, 时, 分, 秒, 星期, 是否单行显示, 播放次数

.....

timeN=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 播放时长, 前景颜色, 背景颜色, 字号, 字体, 字体风格, 是否显示四位年, 日期显示格式, 是否显示年, 月, 日, 时, 分, 秒, 星期, 是否单行显示, 播放次数

txttext1=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 字体, 字号, 字体风格, 水平对齐, 垂直对齐, 行距, 字距, 前景颜色, 背景颜色, 播放特效, 特效速度, 播放时长, 播放次数, 文本内容, 是否播放文字, 是否同步播放, 设置发声音人, 音量, 语速, 语调

.....

txttextN=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, 字体, 字号, 字体风格, 水平对齐, 垂直对齐, 行距, 字距, 前景颜色, 背景颜色, 播放特效, 特效速度, 播放时长, 播放次数, 文本内容, 是否播放文字, 是否同步播放, 设置发声音人, 音量, 语速, 语调

form1=JSON 格式参数列表

.....

formN=JSON 格式参数列表

注：param 参数支持以下两种：

1. param=停留时间, 入屏方式, 出屏方式, 入屏速度, 闪烁速度, 闪烁次数, 播放次数
2. param=停留时间, 入屏方式, 出屏方式, 入屏速度, 闪烁速度, 闪烁次数, 播放次数, 特效类型, 特效速度（目前特效类型和特效速度为预留字段）

说明：

1. 所有数字使用 ASCII 表示；
2. 值的内容中小项使用半角逗号“,” 隔开, 如果内容为空则采用默认值, 但必须有分隔符“,”；
3. N 表示有多个内容, 如有多个条目, 每个条目有多个图片、多个文字, 从 1 开始编号, 按次序编号, 不能颠倒, 不能缺失；
4. itemN 表示播放列表中的第 N 个条目；
5. imgN 表示条目中的第 N 个图片（可无）；

- 6. txtN 表示条目中的第 N 个文本（可无）；
- 7. 条目中的媒体（图片，文本等）按先后顺序放置，如果坐标有冲突，后面的将覆盖前面的内容；
- 8. 除文字内容外不能包含空格；
- 9. x 坐标，y 坐标，显示区域宽度和显示区域高度一起定义了内容在屏上的显示位置和显示区域的大小；
- 10. video、gif、img 标签中的文件名称格式为“文件名.后缀”（例如：filename.bmp、filename.jpg 等）；
- 11. 字体使用整数 1-5（1：黑体，2：楷体，3：宋体，4：仿宋，5：隶书），也可以通过字体名称自定义其它字体（如：Arial Narrow、Courier New 等）
- 12. 自定义字体功能可通过 ViPlex Express 软件将字体文件发送至控制卡。
- 13. 字号定义字符的大小，定义格式为 WWHH, WW 表示宽度, HH 表示高度, 单位为像素（如 3232 表示 32 像素宽*32 像素高的字体，0909 表示 09 像素宽*09 像素高的字体，实际显示以高度为准）
- 14. 前景色背景色使用整数 1-8（1：红色，2：绿色，3：蓝色，4：黄色，5：紫色，6：青色，7：白色，8：黑色），也可以通过 RGB 自定义要使用的颜色，定义格式为 32 位 16 进制的整数（如红色定义为 FF0000）

5.1 文本媒体

文本媒体定义字符显示需要的参数。一个文本媒体定义包含两行。

第一行：txtN=英文逗号间隔的参数列表。

第二行：txtparamN=英文逗号间隔的参数列表。txt 和 txtparam 成对出现，一起定义一个文本的显示，N 表示 item 中的第几个文本媒体，是从 1 开始的整数。

表 5-1 参数说明

格式	txtN=x 坐标,y 坐标, 字体, 字号, 前景颜色, 背景颜色, 闪烁, 文字内容, 显示区域宽度, 显示区域高度, 字体风格 txtparamN=字符间距, 排列方式		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
字体	字体类型	1-黑体 2-楷体 3-宋体 4-仿宋 5-隶书 可以使用字体名称，如	1

		arial 等	
字号	字体大小，单位像素。由宽度和高度组成，宽高各占两个字符。如 0808、3232，999999，对应的字体大小是 8、32、999		1616
前景颜色	字符显示颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	1
背景颜色	字符显示区域背景颜色。	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	8
闪烁	目前无用		0
文字内容	显示的字符	如果显示字符中出现了英文逗号、等号和‘\’，应该在前面添加转义符‘\’	空
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
字体风格	字体风格	0-常规 1-加粗 2-倾斜 4-下划线	0

		8-中划线 字体风格可以组合，例如 3 表示加粗且倾斜显示	
字符间距	字符间隔大小，单位像素	0-1000	0
排列方式	字符排列方向	横向排列 纵向排列	0

5.2 扩展文本媒体

扩展文本是为了方便用户控制纵向和横向排列，实现走马灯显示，增加的一种文本媒体。一个扩展文本定义一行。

txtextN=英文逗号间隔的参数列表。N 表示 item 中的第几个扩展文本，是从 1 开始的整数。

表 5-2 参数说明

格式	txtextN=x 坐标,y 坐标,显示区域宽度,显示区域高度,字体,字号,字体风格,水平对齐,垂直对齐,行距,字距,前景颜色,背景颜色,播放特效,特效速度,播放时长,播放次数,文本内容,是否播放文字,是否同步播放,设置发声音人,音量,语速,语调		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
字体	字体类型	1-黑体 2-楷体 3-宋体 4-仿宋 5-隶书 可以使用字体名称,如 arial 等	1
字号	字体大小，单位像素。由宽度和高度组成，宽高各占两个字符		1616

	如 0808、3232，999999，对应的字体大小是 8、32、999		
字体风格	字体风格	0-常规 1-加粗 2-倾斜 4-下划线 8-中划线 字体风格可以组合，例如 3 表示加粗且倾斜显示	0
水平对齐	文本内容水平对齐方式	0-左对齐 1-右对齐 2-居中	0
垂直对齐	文本内容垂直对齐方式	0-上对齐 1-下对齐 2-居中	0
行距	行间距，单位像素	0-100	1
字距	字符间距，单位像素	0-100	0
前景颜色	字符显示颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	1
背景颜色	字符显示区域背景颜色	同上	8
播放特效	播放特效类型	0-无特效 1-立即显示 2-上展 3-下展 4-左展 5-右展 6-上移 7-下移 8-左移	0

		9-右移 10-淡入淡出 255-随机 12-从右往左滚动 13-从左往右滚动 14-从下往上滚动 15-从上往下滚动	
特效速度	播放特效速度	0-最慢 1-较慢 2-常速 3-较快 4-最快	2
播放时长	播放时长，单位 0.1 秒	1-2147483647	100
播放次数	播放次数	1-255	1
文本内容	显示的字符	如果显示字符中出现了英文逗号、等号和‘\’，应该在前面添加转义符‘\’	空
是否播放文字	是否语音播报文字内容	0-不播报 1-播报	0
是否同步播放	语音播报与文字显示是否同步	0-不同步 1-同步	0
设置发声音人	语音播报发音人	0-普通女声 1-普通男声 2-特别男声 3-情感男声 4-情感儿童声	0
音量	文字合成语音的音量	0-9	5
语速	文字合成语音的速度	0-9	5
语调	文字合成语音的音调	0-9	5

5.3 图片媒体

图片媒体节点定义图片显示需要的参数。一个图片媒体定义包含两行。

第一行：imgN=英文逗号间隔的参数列表。

第二行：imgparamN=停留时间。img 和 imgparam 成对出现，一起定义一个图片媒体的显示。N 表示 item 中第几个图片媒体，是从 1 开始的整数。

表 5-3 参数说明

格式	imgN=x 坐标, y 坐标, 文件名称, 闪烁, 显示区域宽度, 显示区域高度 imgparamN=停留时间, 占位符 0, 动画类型, 播放次数, 动画时长(单位 100ms)		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
文件名称	图片文件名		空
闪烁	目前无用		0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
停留时间	停留时间，单位 0.1 秒	与 param 的停留时间保持一致	100
占位符	无特殊意义	默填 0	
动画类型	图片动画的类型	按“表 3-4”取值	
播放次数	在整个“停留时间”内动画播放的次数	1- 65535	
动画时长	动画播放的时长，单位 0.1 秒	不能超过“停留时间”除以“播放次数”的值	

表 5-4 图片动画及其对应参数值

动画名称	参数值
无特效	00
随机	01
向右扩展	10
向下扩展	11
向左扩展	12
向上扩展	13

动画名称	参数值
中间向外扩展	14
左右扩展	15
上下扩展	16
向右平移	20
向下平移	21
向左平移	22
向上平移	23
向右压缩	30
向下压缩	31
向左压缩	32
向上压缩	33
上下压缩	34
左右压缩	35
向下展开卷轴	40
向上展开卷轴	41
水平百叶窗	50
垂直百叶窗	51
变焦全屏	70
轮子	90
上下齿轮	A0
淡入	B0
向右堆积	C0
向下堆积	C1
向左堆积	C2
向上堆积	C3
左镭射	D0
上镭射	D1
右镭射	D2
下镭射	D3
向下展开	F0

动画名称	参数值
向上展开	F1
上下展开	F2
上下合并	F3

5.4 视频媒体

视频媒体节点定义视频媒体显示需要的参数。一个视频媒体定义一行。
videoN=英文逗号分隔的参数列表。N 表示 item 中第几个视频媒体，是从 1 开始的整数。

表 5-5 参数说明

格式	videoN=x 坐标,y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放次数		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度, 单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度, 单位像素	0-2147483647 0-表示全屏高度	0
文件名称	视频文件名		空
播放次数	视频媒体播放次数	1-255	1

5.5 Gif 媒体

Gif 媒体节点定义 gif 媒体显示需要的参数。一个 gif 媒体定义一行。
gifN=英文逗号间隔的参数列表。N 表示 item 中第几个 gif 媒体，是从 1 开始的整数。

表 5-6 参数说明

格式	gifN=x 坐标,y 坐标, 显示区域宽度, 显示区域高度, 文件名称, 播放时长, 播放次数, 背景颜色		
分隔符	英文等号、逗号		
参数	含义	范围	默认值

x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
文件名称	gif 文件名		空
播放时长	gif 播放时长，单位 0.1 秒	1-4294967295	100
播放次数	gif 媒体播放次数	1-255	1
背景颜色	显示区域背景颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	8

5.6 数字时钟媒体

数字时钟媒体节点定义数字时钟媒体显示需要的参数。一个数字时钟媒体定义一行。

timeN=英文逗号和|分隔的参数列表。N 表示 item 中第几个数字时钟媒体，是从 1 开始的整数。

表 5-7 参数说明

格式	timeN=x 坐标,y 坐标,显示区域宽度,显示区域高度,播放时长,前景颜色,背景颜色,字号,字体,字体风格,是否显示四位年,日期显示格式,是否显示年,月,日,时,分,秒,星期,是否单行显示,播放次数,时区,时区 GMT 偏移,周日 周一 周二 周三 周四 周五 周六,显示格式,上午 下午,月份字符 注意：非必填参数可以不填，但还需使用“,”对各个参数做分割		
分隔符	英文等号、逗号、		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0

y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
播放时长	播放时长，单位 0.1 秒	1-2147483647	100
前景颜色	字符显示颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	1
背景颜色	显示区域背景色	同上	8
字号	字体大小，单位像素。由宽度和高度组成，宽高各占两个字符。如 0808、3232，999999，对应的字体大小是 8、32、999		1616
字体	字体类型	1-黑体 2-楷体 3-宋体 4-仿宋 5-隶书 可以使用字体名称，如 arial 等	1
字体风格	字体风格	0-常规 1-加粗 2-倾斜 4-下划线 8-中划线 字体风格可以组合，例如 3 表示加粗且倾斜显示。	0
是否显示四位年	年显示样式	0-显示两位年 1-显示四位年	1

日期显示格式	日期年、月、日显示顺序	0-年月日顺序 1-日月年顺序 2-月日年顺序	0
是否显示年	是否显示年	0-不显示 1-显示	1
是否显示月	是否显示月	0-不显示 1-显示	1
是否显示日	是否显示日	0-不显示 1-显示	1
是否显示时	是否显示时	0-不显示 1-显示	1
是否显示分	是否显示分	0-不显示 1-显示	1
是否显示秒	是否显示秒	0-不显示 1-显示	1
是否显示星期	是否显示星期	0-不显示 1-显示	1
是否单行显示	数字时钟显示样式	0-多行显示 1-单行显示	1
播放次数	播放次数	1-255	1
时区	数字时钟显示的时区 ID		Asia/Shanghai
时区 GMT 偏移	数字时钟显示的时区的 GMT 偏移		GMT+08:00
一周字符	一周显示字符，使用 隔开	例：周日 周一 周二 周三 周四 周五 周六	不填此项 默认中文

显示格式	日期、时间、星期的显示格式	<div>\$yyyy 年份</div> <div>\$MM 月份</div> <div>\$dd 天</div> <div>\$HH 小时</div> <div>\$mm 分</div> <div>\$ss 秒</div> <div>\$E 星期</div> <div>\$N 上午/下午</div> <div>_n_ 换行</div> <div>例如：“\$dd-\$MM-\$yyyy \$HH:\$mm:\$ss \$E” 的显示效果为 “24-09-2020 12:10:10 星期四”</div>	非必填值，可以为空
上午/下午	上午、下午显示字符，用“ ”隔开，共需设置 2 个	<div>当“显示格式”中有“\$N”时，12 点以前会显示上午对应的字符，12 点以后显示下午对应的字符；</div> <div>例如：可以设置此参数为“上半天 下半天”</div>	非必填参数，程序默认显示“上午 下午”
月份字符	月份显示的字符，用“ ”隔开，共需设置 12 个	<div>月份的显示格式默认为数字，设置此参数后月份的字符文字会替换数字</div> <div>例如：可以设置此参数为“一月 二月 三月 ... 十二月”</div>	非必填参数，程序默认显示数字格式的月份

5.7 模拟时钟媒体

模拟时钟媒体节点定义模拟时钟显示需要的参数。一个模拟时钟媒体定义一行。

clockN=英文逗号分隔的参数列表。N 表示 item 中第几个模拟时钟媒体，是从 1 开始的整数。

表 5-8 参数说明

格式	timeN=x 坐标,y 坐标,显示区域宽度,显示区域高度,播放时长,前景颜色,背景颜色,字号,字体,字体风格,是否显示四位年,日期显示格式,是否显示年,月,日,时,分,秒,星期,是否单行显示,播放次数,时区,时区 GMT 偏移,周日 周一 周二 周三 周四 周五 周六		
分隔符	英文等号、逗号、		
参数	含义	范围	默认值

x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
背景颜色	字符显示颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	8
播放时长	播放时长，单位 0.1 秒	1-2147483647	100
时区	模拟时钟显示的时区 ID		Asia/Shanghai
时区 GMT 偏移	模拟时钟显示的时区的 GMT 偏移		GMT+08:00
时标样式	小时刻度样式	0-矩形 1-圆形 2-数字	0
时标大小	小时刻度大小，单位像素	1-256	5
时标颜色	小时刻度颜色	同背景颜色	2
分标样式	分钟刻度样式	0-矩形 1-圆形	1
分标大小	分钟刻度大小，单位像素	1-256	5
分标颜色	分钟刻度颜色	同背景颜色	3
时针颜色	时针颜色	同背景颜色	4
分针颜色	分针颜色	同背景颜色	2
秒针颜色	秒针颜色	同背景颜色	1
是否显示秒针	是否显示秒针	0-不显示 1-显示	0

5.8 Web 媒体

Web 媒体节点定义 web 媒体显示需要的参数。一个 web 媒体定义一行。

webviewN=英文逗号间隔的参数列表。N 表示 item 中第几个 web 媒体，是从 1 开始的整数。

表 5-9 参数说明

格式	webviewN=x 坐标, y 坐标, URL, 刷新闻隔时间, 显示区域宽度, 显示区域高度		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
URL	Web 的 URL 地址		空
刷新闻隔时间	页面刷新闻隔时间	Int32 范围; 单位 100ms, 为 0 时不刷新, 设置为 0-100 时自动修正为 100	0
显示区域宽度	显示区域宽度, 单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度, 单位像素	0-2147483647 0-表示全屏高度	0

备注:

Web 媒体的播放时长使用 param 中定义的停留时间。

5.9 流媒体

流媒体节点定义流媒体播放需要的参数。一个流媒体定义一行。

streamN=英文逗号间隔的参数列表。N 表示 item 中第几个流媒体，是从 1 开始的整数。

表 5-10 参数说明

格式	streamN=x 坐标, y 坐标, 显示区域宽度, 显示区域高度, URL, 播放次数		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0

y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
URL	流媒体 URL 地址		空
播放次数	播放次数	1-255	1

5.10 表格

表格媒体节点定义表格媒体需要的参数。一个表格媒体定义一行。
formN=JSON 格式的参数列表。N 表示 item 中第几个表格媒体，是从 1 开始的整数。

表 5-11 参数说明

格式		formN=JSON 参数		
分隔符		英文等号		
参数		含义	范围	默认值
x		显示区域左上角的 x 坐标	int	0
y		显示区域左上角的 y 坐标	int	0
width		显示区域宽度，单位像素	int	0
height		显示区域高度，单位像素	int	0
FormBaseInfo	backgroundColor	表格背景颜色	String 1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色	8

			8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	
	isShowYSequence	是否显示行序列	bool true-显示 false-不显示	false
	ySequenceDateArray	行序列	String[]	null
	ySequenceFontName	行序列字体	String 1-黑体 2-楷体 3-宋体 4-仿宋 5-隶书 可以使用字体名称，如 arial 等	1
	ySequenceFontSize	行序列字体大小	int	0
	ySequenceFontColor	行序列字体颜色	同表格背景颜色	1
	ySequenceFontStyle	行序列字体风格	int 0-常规 1-加粗 2-倾斜 4-下划线 8-中划线 字体风格可以组合，例如 3 表示加粗且倾斜显示	0
	isShowXSequence	是否显示列序列	同是否显示行序列	false
	xSequenceDateArray	列序列	String[]	null
	xSequenceFontName	列序列字体	同行序列字体	1
	xSequenceFontSize	列序列字体大小	int	0
	xSequenceFontColor	列序列字体	同表格背景颜色	1

		颜色	色	
	xSequenceFontStyle	列序列字体风格	同行序列字体风格	0
	sequenceGridWidth	序列线粗	int	0
	sequenceGridColor	序列线颜色	同表格背景颜色	2
	horizontalPadding	单元格水平内边距	int	0
	verticalPadding	单元格垂直内边距	int	0
	animationType	动画类型	0-无特效 1-滚动 2-翻动	0
	animationValue	动画参数	特效 1 时,表示每次滚动像素; 特效 2 时,表示每次翻动的高度	0
	animationWaitValue	动画时长	特效 1 时,最上方和下方停留时间,单位毫秒; 特效 2 时,表示每页停留时间,单位毫秒	0
	backgroudColorX	按行单元格背景色	String[],值同表格背景颜色	null
	colorY	按行单元格前景色		null
	backgroudColorY	按列单元格背景色		null
	colorX	按列单元格前景色		null
	yTableHightArray	行高	int[]	null
	xTableWidthArray	列宽	int[]	null
	contentFontName	内容部分字体	同行序列字体	1
	contentFontSize	内容部分字体大小	Int	0
	contentFontColor	内容部分字	同表格背景颜	1

			体颜色	色	
	contentFontStyle		内容部分字体风格	同行序列字体风格	0
	contentGridWidth		内容部分线粗	int	0
	contentGridColor		内容部分线条颜色	同表格背景颜色	2
	yTableAlignArray		按行单元格水平布局	int[] 0-居左 1-居右 2-居中	null
	xTableAlignArray		按列单元格水平布局		null
	YXDate2DArray		自定义数据内容	String[][]	null
	userCellRange		合并单元格，单元格起始和结束行列下标，从0开始	Int[][] 每个合并区域为一个长度为4的数组，格式为：[[起始行，结束行，起始列，结束列], [], []]	例如：合并第一行第一列至第一行第三列的所有单元格，数据格式应该为：[[0, 0, 0, 2]]
	excelName		显示 Excel 名字	String	null
	excelSheetIndex		显示 Excel 第几个 sheet	int	
	DBInfo	DBType	数据库类型	String 取值 sql、mysql、oracle	
		IP	数据库服务器地址	String	
		Port	数据库服务器端口	int	
		DBTable	数据库名称	String	
		User	用户名	String	
		Password	密码	String	
		CycleTime	查询周期	int	
	DBSql		查询 SQL	String	

5.11 天气媒体

天气媒体节点定义天气媒体需要的参数。一个天气媒体定义一行。

weatherN=JSON 格式的参数字列表。N 表示 item 中第几个天气媒体，是从 1 开始的整数。

area 相关内容使用上位机工具查询

表 5-12 参数说明

格式		formN=JSON 参数		
分隔符		英文等号		
参数		含义	范围	默认值
x		显示区域左上角的 x 坐标	int	0
y		显示区域左上角的 y 坐标	int	0
width		显示区域宽度，单位像素	int	0
height		显示区域高度，单位像素	int	0
duration		播放时长	long（单位 0.1 秒，例如 10 代表 1 秒）	0
metaDataWeather	language	表示天气数据的显示语言	String 如 zh-cn, en, jp 等	
	cityNewCode	表示城市 code 的新字段	String citycode 例 CN101110101	
	windType	风力显示类型	String 风力:wind_sc 风速 wind_spd;默认为风力	
	windSpeedUnit	风速计量单位	风速计量单位：公制:m 英制:i;默认为公制单位	
	isShowUnit	是否显示温度单位	beanlean	
	temperatureUnit	温度单位	String（摄氏温度:" Celsius", 华氏温度:" Fahrenheit"	
	regular	显示规则	String（例如："天气:\$w\n 温度:\$t\n 风力:\$b\n 当前温度:\$c\n 当前湿度:\$h"） 注意:\$w\n+\$t\n+\$b\n+\$c\n+\$h 结构不能变，可以删减，例如：如果不想显示湿度数据应该是这样的"天气:\$w\n 温度:\$t\n 风力:\$b	

			级\n 当前温度:\$c\n”		
	refreshPeriod	数据更细频率	int，以毫秒为单位		5000， 最小值 5000
	isShowInOneLine	是否单行显示	boolean		
	textAttributes	textColor	文字颜色		String（例如： #000000）
			size	字体大小	int
			family	字体名称	String[] （例如： [“Arial”]）， 目前数组长度只支持为1
			style	字体样式	String（可选值： 正常： NORMAL； 加粗：BOLD； 斜体： ITALIC； 斜体加粗：BOLD_ITALIC）
			isUnderline	是否显示下划线	boolean FALSE
	effects	speed	滚动速度		float； 当 speedByPixelEnable 参数为 true 时 speed 的值的单位为“像素 / 秒”； speedByPixelEnable 为 false 时，speed 表示

				速度档位， 取值应该是 1 到 10 之 间的整数， 数值越大速 度越快	
		speedByPixelEnable	是否按像素滚动	boolean	FALSE
		headTailSpacing	收尾相接的间距	String	"-1"
		isHeadTail	是否收尾相接	boolean	
		animation	动画滚动效果	String 向左滚动： MARQUEE_LEFT; 向右滚动： MARQUEE_RIGHT; 静止：NONE; 滚动动画仅在 isShowInOnlineLine 参数 为 true 时生效	

5.12 环境监测媒体

环境监测媒体节点定义环境监测媒体需要的参数。一个环境监测媒体定义一行。

environmentN=JSON 格式的参数列表。N 表示 item 中第几个环境监测媒体，是从 1 开始的整数。

表 5-13 参数说明

格式	formN=JSON 参数		
分隔符	英文等号		
参数	含义	范围	默认值
x	显示区域左上角的 x 坐标	int	0
y	显示区域左上角的 y 坐标	int	0
width	显示区域宽度，	int	0

			单位像素		
height			显示区域高度，单位像素	int	0
duration			播放时长	long（单位 0.1 秒，例如 10 代表 1 秒）	0
temperatureUnit			温度单位	String	"℃"
temperatureType			温度类型	int 0: 摄氏温度 1: 华氏温度	0
temperatureScale			温度精度	int (0-2) 0: 只显示整数 1: 保留一位小数 2: 保留两位小数	1
temperatureCompensation			温度补偿值	float	0
airPressureUnit			气压单位	String	"bar"
windSpeedUnit			风速单位	String	"级"
regular			显示规则	String（例如："噪音:\$n\n温度:\$t\n气湿:\$AH\n气压:\$AP\n风速:\$WS\n风向:\$WD\nPM2.5:\$PM2.5\nPM10:\$PM10\nCO2:\$CO2\n"） 注意："\\n" 用来分割显示参数，显示参数可以选择性的设置；"\$" 之前的参数标签可修改，例如可以将 "CO2:\$CO2" 修改为"二氧化碳:\$CO2";"\$" 符号后面的字符不可修改	
textAttributes	backgroundColor		背景色	String（例如：#000000）	
	textColor		文字颜色	String（例如：#000000）	
	font	size	字体大小	int	
		family	字体名称	String[]（例如：["Arial"]），目前数组长度只支持为 1	
		style	字体样式	String（可选值： 正常：NORMAL； 加粗：BOLD； 斜体：ITALIC； 斜体加粗：BOLD_ITALIC）	

		isUnderline	是否显示下划线	boolean	FALSE
--	--	-------------	---------	---------	-------

测试数据：

```
[all]
items=1
[item1]
param=50,255,1,1,0,5,6,0,1
environment1={"airPressureUnit":"kPa","duration":20,"regular":"噪音:$n\n 温度:$t\n 气湿:$AH\n 气压:$AP\n 风速:$WS\n 风向:$WD\nPM2.5:$PM2.5\nPM10:$PM10\nCO2:$CO2\n","temperatureCompensation":0.0,"temperatureUnit":"度",
"temperatureType":0,"temperatureScale":0,"textAttributes":{"font":{"family":["幼圆"],
"isUnderline":false,"size":14,"style":"NORMAL"},"textColor":"#fff000","background-color":"#0000ff00"},"windSpeedUnit":"级",
"height":160,"width":180,"x":0,"y":0}
```

5.13 计时媒体

计时媒体节点定义计时媒体需要的参数。一个计时媒体定义一行。

counttimerN=英文逗号间隔的参数列表。N 表示 item 中的第几个计时媒体，是从 1 开始的整数。

表 5-14 参数说明

格式	counttimerN =x 坐标,y 坐标,显示区域宽度,显示区域高度,播放次数,字体,文字大小,前景色,背景色,固定文本,字体风格,计时方式(倒计时或者正计时),计时长(单位为秒),时间单位(时 分 秒),是否显示时,是否显示分,是否显示秒,是否单行显示		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度,单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度,单位像素	0-2147483647 0-表示全屏高度	0
播放次数	计时媒体的播放次数	目前只支持 1	1
字体	字体类型	1-黑体	1

		2-楷体 3-宋体 4-仿宋 5-隶书 可以使用字体名称，如 arial 等。	
文字大小	字体大小，单位像素。由宽度和高度组成，宽高各占两个字符。如 0808、3232、999999，对应的字体大小是 8、32、999		1616
前景颜色	字符显示颜色	1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 RGB 定义，如 FF0000 表示红色	1
背景颜色	字符显示区域背景颜色	同上	8
固定文本	固定显示在文本时间前面或者上面的文字	前边要显示的字符内容 字符中如果需要输入'、'、'='、'\、'、'\" 应该为 \"、'\"、'\"、'\"\"	
字体风格	字体风格	0-常规 1-加粗 2-倾斜 4-下划线 8-中划线 字体风格可以组合，例如 3 表示加粗且倾斜显示。	0
计时方式	倒计时或者正计时	0 倒计时 1 正计时	0
计时时长	正计时起始时间点或者倒计时总结束时间点	Unix 时间戳，例如 1595911541 代表 2020/7/28 12:45:41	0
时间单位	时间的显示单位	格式为“天 时 分 秒”，使用“ ”隔开，文字都可	

		以编辑	
是否显示时	是否显示时	0 否 1 是	0
是否显示分	是否显示分	0 否 1 是	0
是否显示秒	是否显示秒	0 否 1 是	0
是否单行显示	是否单行显示	0 否 1 是	0

测试数据：

```
[all]
items=1
[item1]
param=100,1,1,1,0,5,1,0,1
counttimer1=0,0,400,400,1,1,2020,1,8,倒计时,0,0,1595911541,日|时|分|
秒,1,1,1,1,0
```

5.14 边框媒体

边框媒体节点定义边框媒体播放需要的参数。一个边框媒体定义一行。

边框媒体需要指定一张图片，整个边框是由这一张图片循环拼接围成的一个矩形。边框的厚度等于图片的高度。

frameN=英文逗号间隔的参数列表。N 表示 item 中第几个边框媒体，是从 1 开始的整数。

表 5-15 参数说明

格式	frameN=x 坐标,y 坐标,显示区域宽度,显示区域高度,图片文件名称,边框动画,动画帧刷新时间等级,媒体播放		
分隔符	英文等号、逗号		
参数	含义	范围	默认值
x 坐标	显示区域左上角的 x 坐标	Int32 范围	0
y 坐标	显示区域左上角的 y 坐标	Int32 范围	0
显示区域宽度	显示区域宽度，单位像素	0-2147483647 0-表示全屏宽度	0
显示区域高度	显示区域高度，单位像素	0-2147483647 0-表示全屏高度	0
图片文件名称	边框图片的名称		空
边框动画	边框动画特效	0 无动画 1 闪烁	0

		2 顺时针旋转 3 逆时针旋转	
动画帧刷新时间等级	动画每两帧之间的时间间隔,每隔 16 毫秒为一个等级。 (无动画时此参数无效) 举例: 0 代表 16 毫秒刷新一帧; 1 代表 32 毫秒刷新一帧; n 代表 16* (n+1) 毫秒刷新一帧;	Int32 范围	0
播放时长	媒体播放时长	int32 (单位 0.1 秒, 例如 10 代表 1 秒) 特殊值: -1 代表播放时间无限长	

测试数据:

```
frame1=0, 0, 200, 100, frame. jpg, 2, 0, 50
```

5.15 路网媒体

路网媒体节点定义路网媒体播放需要的参数。一个路网媒体定义一行。

roadN=英文逗号分隔的参数列表。N 表示 item 中第几个路网媒体, 是从 1 开始的整数, 一般场景, 建议一个 item 里面包含一个路网媒体。

表 5-16 参数说明

格式	roadN=json		
参数	含义	范围	备注
x	显示区域左上角的 x 坐标	Int32 范围	
y	显示区域左上角的 y 坐标	Int32 范围	
width	显示区域宽度, 单位像素	0-2147483647 0-表示全屏宽度	
height	显示区域高度, 单位像素	0-2147483647 0-表示全屏高度	
backgroundColor	路网媒体的背景颜色	1-红色 2-绿色	

		3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 16 进制定义颜色	
roadInfo	路信息	List	

roadInfo 结构

参数	含义	范围	备注
color	路的颜色	十六进制颜色如 (49c96a)	
id	路线 id	Int32 范围	
length	路线长度（在路网媒体上长度）	Int32 范围	
name	路的名称	String	
width	路的宽度	Int32 范围	
points	组成路的关键点信息	List	

Point 结构

参数	含义	范围	备注
x	关键点 x 坐标	Int32 范围	
y	关键点 y 坐标	Int32 范围	

示例数据数据如下：

```
road1={"backgroundColor":"00000000","height":134,"roadInfo":[{"color":"49c96a","id":1,"length":373,"name":"路线1","points":[{"x":13,"y":33}, {"x":83,"y":33}, {"x":163,"y":33}, {"x":306,"y":33}, {"x":386,"y":33}], "width":2}, {"color":"49c96a","id":2,"length":345,"name":"路线2","points":[{"x":17,"y":89}, {"x":136,"y":89}, {"x":260,"y":89}, {"x":362,"y":89}], "width":2}], "width":400,"x":0,"y":0}
```

更新路网：

发送 item_road_0.load 文件，文件内容为 json 数据：

参数如下：

参数	类型	说明	备注
routes	List<TJRoute>	一个 TJRoute 表示一段路的路况信息	

TJRoute 结构

参数	类型	说明	备注
id	Int32 范围	路线 id	
startPercent	Float	起始位置（路长的百分比），0-1	
endPercent	Float	结束位置（路长的百分比），0-1	
width	Int32	路宽度	
color		1-红色 2-绿色 3-蓝色 4-黄色 5-紫色 6-青色 7-白色 8-黑色 可以使用 16 进制定义颜色	

示例数据如下：

```
[{"id":1,"startPercent":0.1,"endPercent":0.6,"width":0,"color":"1"}, {"id":2,"startPercent":0.1,"endPercent":0.1,"width":0,"color":"1"}, {"id":3,"startPercent":0.3,"endPercent":0.6,"width":0,"color":"7"}, {"id":0,"startPercent":0.7,"endPercent":0.85,"width":0,"color":"FF0000"}]
```

6

附录三：播放以及局部更新说明

1. 指定播放：可以页面切换，每个页面上可以附着多个媒体，每个媒体可以在不同空间（item 即为页面）。
2. 局部更新：
 - a. 只有一个 item，item 编号不为-1。这种情况会跟随普通播放 item 编号出现或者隐藏。
 - b. 只有一个 item，item 编号为-1。这种情况会在所有播放最上层一直显示。
 - c. 多个 item。这种情况会独立切换播放，内容和普通播放一样。层级按序号，并且在普通播放之上。
3. 指定播放的 item 的 param 的时间只对下属的 txt 和 img 文件生效。
4. 每个页面的切换时间为 下属所有媒体的最大时间。
 - a. 播放接口调用顺序。指定播放
下发播放文件 和 播放资源文件。
指定播放列表。
 - b. 局部更新
下发资源文件。
使用局部更新接口。

版权所有 ©2022 西安诺瓦星云科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

NOVA STAR 是诺瓦星云的注册商标。

声明

欢迎您选用西安诺瓦星云科技股份有限公司的产品，如果本文档为您了解和使用产品带来帮助和便利，我们深感欣慰。我们在编写文档时力求精确可靠，随时可能对内容进行修改或变更，恕不另行通知。如果您在使用中遇到任何问题，或者有好的建议，请按照文档提供的联系方式联系我们。对您在使用中遇到的问题，我们会尽力给予支持，对您提出的建议，我们衷心感谢并会尽快评估采纳。

24小时免费服务热线

400-696-0755

<http://www.novastar-led.cn>

西安总部

地址：西安市高新区科技二路72号西安软件园零壹广场DEF101

电话：029-68216000



诺瓦科技官方微信号