

1. 公司简介

深圳市灰度科技有限公司成立于2009年，是专业从事LED显示应用技术研究的高新技术企业，是LED控制系统专业制造商。一直以来，灰度科技专注于高性能、高性价比、高智能化的LED控制系统的研发。公司掌握嵌入式控制、图形信息数字化、多媒体等核心技术，并成功应用于我们的产品中。

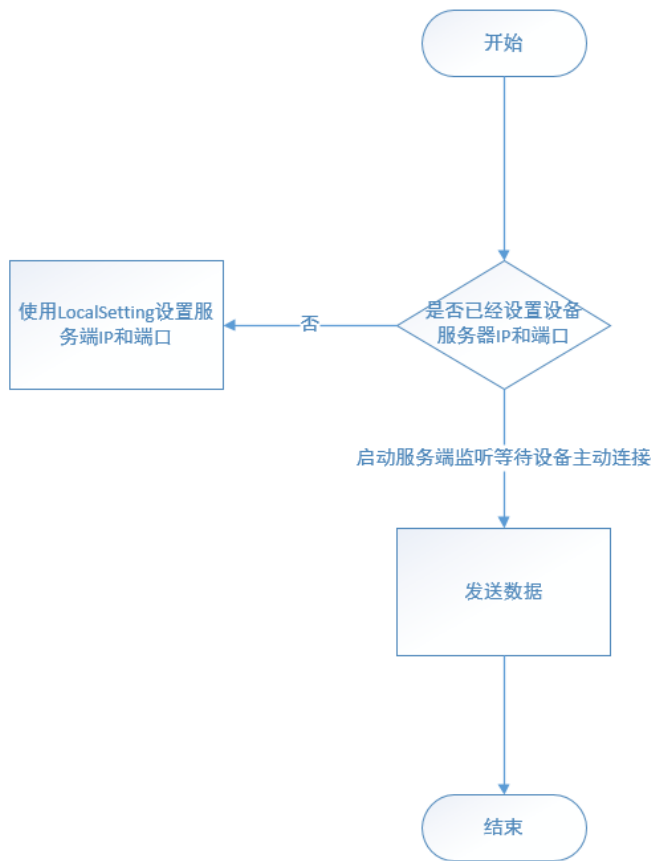
一路走来，从单双色到全彩，从小屏到大屏，灰度科技已成为领先的异步控制系统解决方案商。公司作为高科技企业，以研发为主体，以技术为本，以人才为源，实时掌控高新技术，为LED行业打造更多精彩的产品。

灰度科技正以专业的设计、精良的品质、完善的服务受到越来越多客户朋友的青睐。未来发展中，灰度科技将继续坚持专业化道路，坚持技术创新和技术服务，与国内外广大的LED显示屏制造商、系统集成商、经销商加强协作，共同发展！

2. 备注

1. 设备ID中间带D的才是工程卡, 不是工程卡不支持二次开发（比如 C16-D22-098D2）
2. 请先使用上位机软件HDPlayer调试好显示屏，使其正常显示内容。
3. SDK[通讯协议](#)是开放的，用户可根据自己需求灵活开发，或者使用我们自带的jar包进行快速开发（本文档为Java 开发包）。
4. 设备作为客户端时主动连接服务端的连接间隔时间大概时间为1-3分钟，极端的情况为11-13分钟。
5. SDK协议使用TCP协议进行数据传输
6. SDK接口数据使用UTF-8编码格式
7. 流程如下：

LocalSetting为辅助工具，自动查找局域网内的设备，可以用来设置设备IP和服务器IP 端口等信息。（必须禁用掉本地虚拟网卡，否则[LocalSetting](#)可能无法找到设备。服务器sdk 端口为SdkServer 启动时 设置的可用端口）下图为整体流程图：



Java SDK 使用流程图：

3. SDK通讯协议

1.具体通讯协议查看sdk3.0文档，本文档仅是java sdk便捷开发包

4. SDK 开发包集成

1.依赖环境：JDK8、maven3.6及以上

2.解压java sdk压缩包。包中主要包括（fullColor-xxx.jar&fullColor-xxx.pom）demo程序

3.安装SDK到本地maven repository

```
mvn install:install-file -Dfile=fullColor-xxx.jar -DgroupId=cn.huidu.sdk -DartifactId=fullColor -Dversion=<版本号xxx> -Dpackaging=jar
```

注意将上述指令中sdk版本号替换为相应版本;Windows系统下, 使用CMD控制台, 不要使用Powershell。

4.pom.xml 里添加 sdk 依赖信息

```
<dependency>
  <groupId>cn.huidu.sdk</groupId>
  <artifactId>fullColor</artifactId>
  <version>版本号xxx</version>
</dependency>
```

5. SDK API接口说明

此设备编号 仅作为举例

```
String sn = "A3-D22-A0000";
```

调用方式为 使用HDSdkRequest 调用函数

示例 : HDSdkRequest.getDevices()

注意 : 除设备列表 其他api 设备接口数据反馈属于异步方式 在事件handler监听数据; api返回值是否调用成功 指的是 api调用是否正常 不是指设备已经收到指令信息。

5.1.获取在线设备列表

```
getDevices()
```

返回值:

```
{
  "sn": "A3-D22-A0000",
  "remoteAddress": "192.168.1.94:33442",
  "heartTime": "2022-12-17 16:53:10",
  "connectTime": "2022-12-17 16:52:08"
}
```

- **sn** : String 类型, 设备编号
- **remoteAddress** : String 类型, 设备ip连接信息
- **heartTime** : String 类型, 设备心跳
- **connectTime** : String 类型, 设备连接sdk时间

5.2.获取设备信息

参数:

- **sn** : String 类型, 设备编号

```
getDeviceInfo(sn)
```

返回值:

- boolean类型, API是否调用成功

5.3.获取网络信息

参数:

- **sn**: String 类型, 设备编号

```
getNetworkInfo(sn)
```

返回值:

- boolean类型, API是否调用成功

5.4.关屏

参数:

- **sn**: String 类型, 设备编号

```
closeScreen(sn)
```

返回值:

- boolean类型, API是否调用成功

5.5.开屏

参数:

- **sn**: String 类型, 设备编号

```
openScreen(sn)
```

返回值:

- boolean类型, API是否调用成功

5.6.获取开关屏信息

参数:

- **sn**: String 类型, 设备编号

```
getSwitchTime(sn)
```

返回值:

- boolean类型, API是否调用成功

5.7.设置开关屏信息

参数:

- **sn**: String 类型, 设备编号
- **HdSwitchTimeParam**: Object 类型, 开关屏参数
 - **playEnable**: boolean类型, 是否开启时间段控制开关屏, 取值范围{"true"(开启), "false"(不开启)}
 - **HDSwitchTimeItem**: object类型, 单组开关屏信息
 - **start**: String 类型, 开屏时刻, 格式hh:mm:ss
 - **end**: String 类型, 关屏时刻, 格式hh:mm:ss

```
DSwitchTimeParam hdSwitchTimeParam = new HDSwitchTimeParam();  
// 设置参数 .....  
setSwitchTime(sn, hdSwitchTimeParam)
```

返回值:

- boolean类型, API是否调用成功

5.8.截图

参数:

- **sn**: String 类型, 设备编号

```
getScreenshot2(sn)
```

返回值:

- boolean类型, API是否调用成功

5.9.获取设备名

参数:

- **sn**: String 类型, 设备编号

```
getDeviceName(sn)
```

返回值:

- boolean类型, API是否调用成功

5.10.设置系统音量

参数:

- **sn**: String 类型, 设备编号
- **HDSysVolumeParam**: Object 类型, 设置系统音量参数
 - **defaultModel**: boolean类型, 选择模式default和ploys
 - **defaultPercent**: int类型, 默认音量 必须存在 音量值[0-100]

- **HDSysSystemVolumeItem**: object类型, 节点音量设置

- **start**: String 类型, 时间格式hh:mm:ss
- **percent**: int 类型, 音量 [0-100]

```
HDSysSystemVolumeParam hdSystemVolumeParam = new HDSysSystemVolumeParam();  
// 设置参数 .....  
setSystemVolume(sn, hdSystemVolumeParam);
```

返回值:

- boolean类型, API是否调用成功

5.11.获取系统音量

参数:

- **sn**: String 类型, 设备编号

```
getSystemVolume(sn)
```

返回值:

- boolean类型, API是否调用成功

5.12.重启设备

参数:

- **sn**: String 类型, 设备编号

```
reboot(sn)
```

返回值:

- boolean类型, API是否调用成功

5.13.获取亮度配置

参数:

- **sn**: String 类型, 设备编号

```
getLuminancePloy(sn)
```

返回值:

- boolean类型, API是否调用成功

5.14.设置亮度配置

参数:

- **sn**: String 类型, 设备编号
- **HDLuminancePloyParam**: Object 类型, 亮度设置参数
 - **mode**: object类型, "default"(默认), "ploys"(自定义模式), "sensor"(传感器模式)
 - **defaultPercent**: int类型, 默认模式时, 设置的亮度等级[1, 100]
 - **HDLuminancePloyItem**: object类型, 节点, 表示可设置在不同时间段使用不同的亮度等级
 - **start**: String 类型, 改变亮度等级的开始时间, 格式: hh:mm:ss
 - **percent**: int 类型, 亮度等级
 - sensorMin**: int类型, 传感器模式最小亮度等级
 - sensorMax**:int类型, 传感器模式最大亮度等级
 - time** :int类型, 亮度调整的间隔时间, 取值范围[5, 15] (单位秒)

```
HDLuminancePloyParam hdLuminancePloyParam = new HDLuminancePloyParam();
// 设置参数 .....
setLuminancePloy(sn, hdSystemVolumeParam);
```

返回值:

- boolean类型, API是否调用成功

5.15.设置多屏同步

参数:

- **sn**: String 类型, 设备编号
- **enable**: boolean 类型, 多屏同步使能标志, 取值范围{"true", "false"}

```
setMulScreenSync(sn, false);
```

返回值:

- boolean类型, API是否调用成功

5.16.获取多屏同步标志

参数:

- **sn**: String 类型, 设备编号

```
getMulScreenSync(sn);
```

返回值:

- boolean类型, API是否调用成功

5.17.屏幕旋转

参数:

- **sn**: String 类型, 设备编号
- **ScreenRotation**: object 类型, 屏幕旋转顺时针旋转度数

```
// 设置屏幕旋转时，会导致宽高调整和节目重播，请不要在多屏同步时旋转。  
screenRotation(sn, ScreenRotation);
```

返回值:

- boolean类型，API是否调用成功

5.18.获取时间校正数据

参数:

- **sn**: String 类型，设备编号

```
getTimeInfo(sn);
```

返回值:

- boolean类型，API是否调用成功

5.19.时间校正功能

参数:

- **sn**: String 类型，设备编号
- **HDTIMEInfo**: Object 类型，时间校正参数
 - **summer**: boolean类型，是否开启夏令时, 取值范围{"true"(开启夏令时), "false"(不开启夏令时)}
 - **timezone**: String 类型，时区字符串, 必须以(UTC+hh:mm)开始, 例如:
(UTC+08:00)Beijing,Chongqing,HongKong,Urumchi
 - **TimeInfoSync**: object类型，是否开启时间自动同步, 取值范围{"none"(不开启自动同步), "gps"(gps校时), "network"(网络校时), "auto"(自动校时)}
 - **time**: String，将设置设备的时间, 该值在TimeInfoSync="none"时生效, 其他情况忽略, 格式 yyyy-MM-dd hh:mm:ss
 - **server**: String，服务列表, 逗号分隔
 - **rfEnable**: boolean，rf同步是否使能{"false"不使能 "true"使能}
 - **master**: String，主设备{"false", "true"}
 - **channel**: String，通道

```
HDTIMEInfo hdtIMEInfo = new HDTIMEInfo();  
// 设置参数 .....  
setTimeInfo(sn, hdtIMEInfo);
```

返回值:

- boolean类型，API是否调用成功

5.20.固件升级

参数:

- **sn**: String 类型，设备编号
- **HDFileInfo**: Object 类型，文件信息参数

- **name**: String, 为固件名 资源名或者http的路径
- **md5**: String 类型, 文件md5值
- **size**: long类型, 文件的大小

```
HDFFileInfo fileInfo = new HDFFileInfo();
// 设置参数 .....
httpUpgrade(sn, fileInfo);
```

返回值:

- boolean类型, API是否调用成功

5.21.设置设备名

参数:

- **sn**: String 类型, 设备编号
- **deviceName**:String 类型, 设备名 大小小于64长度

```
setDeviceName(sn, "newName");
```

返回值:

- boolean类型, API是否调用成功

5.22.获取当前播放节目的GUID

参数:

- **sn**: String 类型, 设备编号

```
getCurrentPlayProgramGUID(sn);
```

返回值:

- boolean类型, API是否调用成功

5.23.发布节目

参数:

- **sn**: String 类型, 设备编号
- **xml**: String 类型, 节目xml 按照文档构建

```
String xml = "节目xml..."
sendProgram(sn, xml);
```

返回值:

- boolean类型, API是否调用成功

5.24.自定义内容

参数:

- **sn**: String 类型, 设备编号
- **xml**: String 类型, xml内容 按照文档构建

```
String xml = "xml..."
sendCustomData(sn, xml);
```

返回值:

- boolean类型, API是否调用成功

6. SDK demo

参照demo程序进行开发 (demo 使用springboot作为演示) , demo程序在demo目录下。

```
// HD全彩 sdk 服务启动 demo

@SpringBootApplication
public class HDDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(HDDemoApplication.class, args);

        // 添加处理Event的handler
        SdkMessageHandlerConfig.initSdkMessage(new HDEventMessageListener());

        // 启动 sdk服务
        SdkServer server = new SdkServer();

        // port: 监听的tcp端口 keepLive: 超时时间 单位秒
        server.startServer(1884, 120);

        // 服务停止
        Runtime.getRuntime().addShutdownHook(new Thread() {
            @Override
            public void run() {
                //sdk 停止
                server.stopServer();
            }
        });
    }
}
```

```
// 事件回调消息监听 demo

public class HDEventMessageListener implements EventHandler<SdkPublishMessage> {

    @Override
    public void onEvent(SdkPublishMessage sdkPublishMessage, long sequence,
        boolean endOfBatch) {
        // 接收消息
    }
}
```

```
    try {  
        // 业务代码.....  
        System.err.println("sdk message : " + sdkPublishMessage.toString());  
    } catch (Exception e) {  
        // 注意 必须抓异常 否则影响后面消息监听!  
    }  
}  
}
```

• SdkPublishMessage

- **sn** : String 类型, 设备编号
- **message** : String 类型, 回调消息内容 内容格式下面说明
- **remoteAddress** : String 类型, 客户端ip信息
- **type**: int类型, 消息类型 0 : xml接口反馈 ; 1 : action 设备上下线 注: sdk服务停止后或者杀死进程 设备离线消息会部分丢失 需自己处理此情况

type	描述
0	xml格式 具体内容查看sdk3.0文档 接口反馈
1	json格式 action : online 上线 offline 离线 sn : 设备编号