

上海仰邦科技股份有限公司

Java API User Manual V1.1

软件设计说明书

2014-5-9

Copyright

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by ONBON.

© 2010-2014 Onbon

版本历史:

版本号	日期	作者	描述
1.0	2016-10-26	Wangyq	第一个版本

目录

1. 开发前必读	4
1.1 你选的控制器型号正确吗?	4
1.2 什么是动态区	4
2. 关于此 SDK	5
3. 准备工作	6
3.1 相关工具	6
3.2 通讯模式	6
3.3 区域类型	6
4. 接口说明	7
4.1 SDK 初始化	7
4.2 Screen 类	7
4.3 屏幕连接	7
4.4 屏幕控制	8
4.5 参数回读	8
4.6 节目与区域	8
4.6.1 创建节目文件	9
4.6.2 创建图文区域	10
4.6.3 创建时间区	11
4.6.4 发送节目	11
4.7 动态区	12
4.7.1 创建动态区	12
4.7.2 创建显示内容	13
4.7.3 发送动态区	13
5. 联系我们	14
附录 1	15

Java API User Manual V1.1

1. 开发前必读

1.1 你选的控制器的型号正确吗？

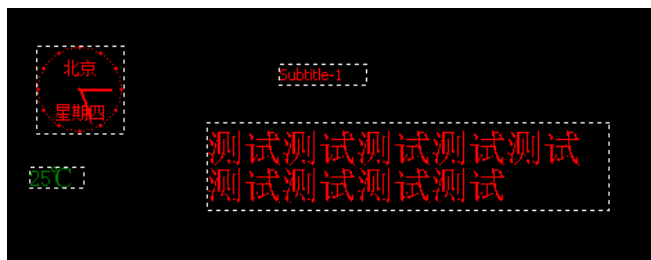
通过在 LED 行业十几年的积累和沉淀，仰邦科技拥有最全、最丰富的 LED 控制器产品线。

根据控制方式的不同可分为同步控制器和异步控制器。

	异步控制器	同步控制器
控制方式	先制作节目，然后通过网口，串口，无线，U 盘等方式将制作好的节目上传到控制器上。上传完成后，控制器可以脱机独立运行。	通过 DVI/HDMI 接口，对电脑或其它主机上显示内容进行截屏显示。无法脱离显示主机而独立对显示屏进行控制。
优点	系统架构简单，控制方式灵活，方便组网	
应用场合	工业现场，	舞台全彩屏，租赁全彩屏等
代表型号		BX-VHE, BX-VH, BX-VS, BX-V, BX-V75, BX-V75L 等

1.2 什么是动态区

根据存储介质的不同，我们将控制器上显示的内容分为两类：普通节目和动态区。普通节目通常如下图所示。它可以包括各种区域（时间，表盘，传感器，图文等）。



普通节目，会被存储在控制器上的 FLASH 存储器中，其内容掉电不丢失。但因为 FLASH 存储器擦写寿命只有 100,000 次。所以，其不能用于更新频率很高的场合。例如：停车场车位信息，外部传感器状态的实时更新，车次状态的实时更新等。

动态区是完全独立于普通节目的，其显示内容可以按区域进行单独更新。它可以与普通节目一起播放，也可以单独进行播放。它最大的特点是显示内容存储在 RAM 中，掉电不保存，但没有刷新次数的限制。因此，其通常应用于显示内容更新频率比较高的场合。






支持动态区的控制器有，BX-5K1/5K2, BX-5MK1/5MK2, BX-5E1/5E2/5E3, BX-6Q1/6Q2/6Q3, 以及 BX-YQ 全系列。

2. 关于此SDK

此 SDK 目前支持的控制器型号如下表所示：

SDK 版本	支持的型号	备注
Bx05	BX-5T, BX-5A, BX-5M, BX-5U, BX-5E, BX-5Q 全系列五代控制器	
Bx06	BX-6Q 系列控制器	

JAVA SDK 后，通常会有以下几个文件：

 bx05-0.0.2-SNAPSHOT	2016/10/13 14:25	WinZip 文件	3,217 KB
 bx05-0.0.2-SNAPSHOT_javadoc	2016/10/13 14:25	WinZip 文件	723 KB
 bx05-0.0.2-SNAPSHOT_tutorial	2016/10/13 14:25	WinZip 文件	3,229 KB
 BxClientDemo	2016/10/26 12:18	Java 源文件	17 KB
 log	2016/10/10 12:36	Properties 源文件	2 KB

其中：

bx05-0.0.2-SNAPSHOT – JAR 库文件，解压后作为库文件导入工程

bx05-0.0.2-SNAPSHOT_javadoc – SDK 各个接口的说明文档

bx05-0.0.2-SNAPSHOT_tutorial – 各个功能的 demo 用例

BxClientDemo – 一个功能较全，注释详细的 DEMO 用例

log.property – log4j 配置文件

3. 准备工作

3.1 相关工具

在进行开发之前，请先确保您已经具备以下资源。如果没有，请先根据以下链接，到相应的地址进行下载。

编号	名称	说明	备注
1	Ledshow TW 2015	五/六代单双色与全彩门楣控制器的配套 PC 软件。此软件可以用于对控制器硬件进行调试。	
2	JAVA SDK	Java 版本的二次开包	

3.2 通讯模式

控制器及 SDK 可以支持以下几种通讯模式，请根据实际需求进行选择。

编号	模式	说明	备注
1	串口	此模式用于使用 RS232/RS485 接口的应用场合	
2	网络 (Client)	此模式用于局域网内，控制器 IP 固定的应用场合。此时 PC 端为 TCP Client。	
3	网络 (Server)	此模式用于广域网上，控制器外网 IP 不固定，PC 无法通过控制器 IP 直接访问控制器。此时，PC 应具备固定 IP，且为 TCP Server，控制器设置好后，会主动连接 Server。	
4	GPRS	当使用 GPRS DTU 作为通讯信道时，使用此种模式。此时，PC 作为 TCP Server，控制器/DTU 作为 Client 主动连接 Server。	

3.3 区域类型

4. 接口说明

4.1 SDK初始化

五代控制器

```
//
// SDK 初始化
//Bx5GEnv.initial("log.properties");
Bx5GEnv.initial("log.properties", 30000);

//
// 5Q 系列控制器
// Bx5GScreenClient screen = new Bx5GScreenClient("MyScreen", new Bx5Q());
```

六代控制器

```
//
// SDK 初始化
//Bx6GEnv.initial("log.properties");
Bx6GEnv.initial("log.properties", 30000);
```

4.2 Screen类

与控制器的所有命令交互都需要通过 Bx5GScreen 类，或其子类来进行。其子类包括：Bx5GScreenClient, Bx5GScreenRS, Bx5GScreenServer。

其中：

Bx5GScreenClient – 客户端模式使用

Bx5GScreenRS – 串口模式使用

Bx5GScreenServer – 服务器模式使用

创建 Screen 对象的通常代码如下：

```
//
// 5Q 系列控制器
// Bx6GScreenClient screen = new Bx6GScreenClient("MyScreen", new Bx5Q());

//
// 其它控制器
// 创建 screen 对象，用于对控制器进行访问
Bx6GScreenClient screen = new Bx6GScreenClient("MyScreen");
```

4.3 屏幕连接

在对控制器交互之前，需先与控制器建立连接，其代码如下：

```
//
// 连接控制器
// 其中，192.168.88.199 为控制器的实际 IP，请根据实际情况填写。
// 如你不知道控制器的 IP 是多少，请先使用 LEDSHOW TW 软件对控制器进行 IP 设置
// 端口号默认为 5005
if (!screen.connect("192.168.88.198", 5005)) {
    System.out.println("connect failed");
}
```

与控制器交互完成后，需断开与控制器之前的连接，其代码如下：

```
//
// 断开与控制器之间的链接
screen.disconnect();
```


4.4 屏幕控制

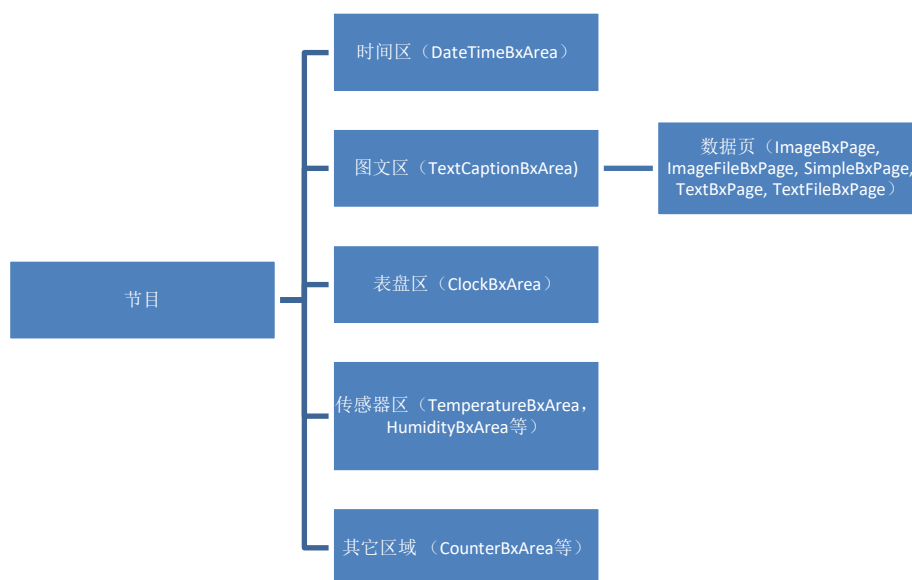
```
//  
//  
// 以下为一些简单控制命令的使用方法  
// 更多命令可以参考 Javadoc 中的 Bx6GScreen 类  
  
//  
// 开关机  
screen.turnOff();      // 关机  
screen.turnOn();       // 开机  
  
// 校时  
screen.syncTime();  
  
// 调整亮度  
// 亮度值 为 1- 16, 16 级为最高亮度  
screen.manualBrightness((byte)8); // 将屏幕亮度调整至 8  
screen.manualBrightness((byte)16); // 将屏幕亮度调整到 16
```

4.5 参数回读

```
//  
// 可以通过以下接口, 回读控制器状态  
Bx6GScreen.Result <ReturnControllerStatus> result1 = screen.checkControllerStatus();  
if(result1.isOK()) {  
    ReturnControllerStatus stauts = result1.reply;  
    stauts.getBrightness();  
    stauts.getRtcDay();  
    stauts.getScreenOnOff();  
    //  
    // status 还有很多其它接口, 可以根据实际需求在此调用以获取相应状态  
}  
else {  
    ErrorType error = result1.getError();  
}
```

4.6 节目与区域

节目主要用于组合屏上显示的内容, 它由多个区域组成。控制器同一时间只能播放一个节目, 它是控制器显示内容可以单独更新的最小单位。



以下，我们将按步骤创建一个节目（将包括一个图文区和一个时间区），并将其发送到控制器进行显示。其步骤如下：

- ✧ 创建节目文件
- ✧ 创建图文区，并将其添加到节目中
- ✧ 创建时间区，并将其添加到节目中
- ✧ 将节目发送至控制器显示

4.6.1 创建节目文件

下面代码创建了一个节目文件，并设置了其边框属性。创建节目时需要传入两个参数。

其中，第一个参数表示节目名称。其长度为4个字符，第一个字符必需为P，后三位为数字，例如P042。节目名称用于对节目进行唯一标识。

第二个参数为显示屏属性，具体可以参考Bx5GscreenProfile类。

```

//
// 下面开始创建第一个节目，P000
// 此节目包括只包括一个图文区，图文区中包括两个数据页：一页文本，一页图片
// 显示节目边框
// 区域显示边框

// 创建节目文件
ProgramBxFile p0 = new ProgramBxFile("P000", profile);
// 是否显示节目边框
p0.setFrameShow(true);
// 节目边框的移动速度
p0.setFrameSpeed(20);
// 使用第几个内置边框
p0.loadFrameImage(13);
  
```

关于节目类的其它接口可参考ProgramBxFile类。

4.6.2 创建图文区域

控制器支持的区域有很多种，例如：图文区、时间区、传感器区等。其中，最常用的图文区（TextCaptionBxArea）。图文区可以用于显示文本和图片，文字或图片可以按数据页依次添加到区域中，每页数据均可设置特技方式，停留时间等属性。

因此，创建图文区的步骤大致如下：

- ✧ 创建 TextCaptionBxArea 对象
- ✧ 创建 TextBxPage 或 ImageFileBxPage 对象
- ✧ 将创建好的 Page 对象添加到 TextCaptionBxArea 中

如下代码，创建了一个文本区，并向这个区域中，添加一个文本页，一个图片页。

```
// 创建一个数据页，并希望显示“P000”这几个文字
TextBxPage page = new TextBxPage("P000 - This is the first program!");
// 对文本的处理是否自动换行
page.setLineBreak(true);
// 设置文本水平对齐方式
page.setHorizontalAlignment(TextBinary.Alignment.NEAR);
// 设置文本垂直居中方式
page.setVerticalAlignment(TextBinary.Alignment.CENTER);
// 设置文本字体
// !!! 注意：有些字体是不支持中文的，比如：consolas 字体，因此，如果要显示中文请将字体改成宋体等中文字体
//page.setFont(new Font("宋体", Font.PLAIN, 12));           // 字体
page.setFont(new Font("consolas", Font.PLAIN, 12));         // 字体
// 设置文本颜色
page.setForeground(Color.red);
// 设置区域背景色，默认为黑色
page.setBackground(Color.darkGray);
// 调整特技方式
page.setDisplayStyle(styles[0]);

// 调整特技速度
page.setSpeed(1);
// 调整停留时间，单位 10ms
page.setStayTime(100);

// 将前面创建的 page 添加到 area 中
tArea.addPage(page);

// 再创建一个数据页，用于显示图片
ImageFileBxPage iPage = new ImageFileBxPage("d:/1.png");
// 调整特技方式
iPage.setDisplayStyle(styles[1]);
// 调整特技速度
iPage.setSpeed(1);
// 调整停留时间，单位 10ms
iPage.setStayTime(100);

// 将前面创建的 iPage 添加到 area 中
tArea.addPage(iPage);

// 将 area 添加到节目中
p0.addArea(tArea);
```

关于 TextCaptionBxArea, TextBxPage, ImageFileBxPage 的更多接口, 请参考 JAVADOC 中相关类的说明。

4.6.3 创建时间区

时间区的创建过程大致如下:

- ✧ 创建 DateTimeBxArea 对象
- ✧ 设置各时间单元显示方式
- ✧ 将 DateTimeBxArea 添加到节目中

```
//  
// 下面代码创建了一个时间区  
// 注意: 只需要输入时间区的起始坐标, 区域的宽度和高度 SDK 会根据字体和显示方式自动计算  
DateTimeBxArea dtArea = new DateTimeBxArea(0, 0, screen.getProfile());  
// 设置字体  
dtArea.setFont(new Font("Arial", Font.PLAIN, 12));  
// 设置颜色  
dtArea.setForeground(Color.YELLOW);  
// 多行显示还是单行显示  
dtArea.setMultiline(true);  
//  
// 年月日的显示方式  
// 如果不显示, 则设置为 null  
dtArea.setDateStyle(DateStyle.YYYY_MM_DD_1);  
dtArea.setTimeStyle(TimeStyle.HH_MM_SS_1);  
dtArea.setWeekStyle(null);  
  
// 将时间区添加到节目中  
p0.addArea(dtArea);
```

关于 DateTimeBxArea 类的更多接口, , 请参考 JAVADOC 中相关类的说明。

4.6.4 发送节目

针对节目发送, 我们提供了多组接口, 其可分为两类。一类为同步接口, 一类为异步接口。同步接口发送时, 会阻塞主线程。异步接口会新建一个线程来发送节目, 不会阻塞主线程。

同步接口包括 writeProgram(), writePrograms()

异步接口包括 writeProgramAsync(), writeProgramsAsync()

参考代码如下:

```
//  
// 将节目文件发送到控制器  
// 发送节目有三种方式  
// 可以根据自己的需求进行选择  
  
//  
// 1. writeProgramsAsync - 异步方式, 即 SDK 会自己起线程来发送  
// 此时需传入 BxFileWriterListener  
// 可在相应的接口对相应的事件进行处理  
//screen.writeProgramsAsync(plist, new WriteProgramTextCaptionWithStyle());  
  
//  
// 2. writePrograms - 同步方式, 即 SDK 会 BLOCK 住一直等到节目发送完毕  
screen.writePrograms(plist);
```

```
//
// 此方法通常不用
// 3. 同步方式将节目写入控制器，本方法不做任何检查，从而提高发送效率
//screen.writeProgramQuickly(pf);
```

注：发送节目时，如果控制器上之前存储有相同文件名的节目，则老节目会被自动覆盖。因此，通常无需手动调用删除节目命令。

4.7 动态区

动态区是一种比较特殊的区域，其有以下几个主要特点：

- ✧ 更新次数没有限制
- ✧ 内容掉电不保存
- ✧ 独立于节目进行编辑
- ✧ 可以支持多个区域，且每个区域可以进行单独更新
- ✧ 可以和单个或多个节目绑定显示，即作为节目的一个区域进行显示
- ✧ 可以作为单独的一个节目进行独立播放
- ✧ 灵活的控制方式：超时时间控制、是否立即显示等

其创建过程通常如下：

- ✧ 创建 `DynamicBxAreaRule` 对象
- ✧ 创建 `TextCaptionBxArea` 对象，并向此对象中添加相应的数据页
- ✧ 发送动态区

4.7.1 创建动态区

`DynamicBxArea` 类的构造函数，如下所示：

```
DynamicBxAreaRule(int id, byte runMode, byte immediatePlay, int timeout)
```

参数说明：

Id – 动态区的 ID 号，用于对多个动态区进行区分

RunMode – 表示动态区的运行模式，其具体定义如下：

0x00：多个数据页循环显示

0x01：显示完成后静止显示最后一页数据。

0x02：循环显示，超过设定时间后数据仍未更新时不再显示。

0x03：循环显示，超过设定时间后数据仍未更新时显示 Logo 信息。

0x04：循环显示，显示完最后一页后就不再显示。

immediatePlay – 表示内容更新后动态区的播放方式，其具体定义如下：

0x00：与异步节目一起播放，即可以作为某一个或多个节目的一个区域进行显示

0x01：异步节目停止播放，仅播放动态区域，可以理解成此时，动态区是作为单独的一个节目来播放。且更新后会立即播放。

0x02：当播放完节目编号最高的异步节目后播放该动态区域。此时动态区也是作为单独的节目播放，但会等普通节目播放完成再播放。

Timeout – 如果需要超时处理，可设置此超时时间，其对应于 `runMode` 中的 0x01, 0x02 模式。

注：

如果 `immediatePlay` 选用了模式 0x00，可以使用 `addProgram()` 接口与节目进行绑定。

以下代码创建了一个动态区，ID 为 0，`runmode` 为 0，`immediatePlay` 为 0x00。且此动态区与节目 0 和节目 1 进行了绑定。其具体代码如下：

```
//  
// 定义一个动态区  
// 可以通过 ID 来更新不同的动态区内容，此处 ID 为 0  
  
DynamicBxAreaRule dynRule = new DynamicBxAreaRule(0, (byte)0, (byte)0, 0);  
  
dArea.addProgram("P000");  
dArea.addProgram("P001");
```

4.7.2 创建显示内容

动态区的显示内容与普通图文区一致，如下所示：

```
TextCaptionBxArea dAreaContent = new TextCaptionBxArea(16, 16, 64, 16,  
screen.getProfile());  
page = new TextBxPage("Dynamic");  
dAreaContent.addPage(page);
```

4.7.3 发送动态区

```
// 发送动态区之前，如果需要删除之前的动态区，可以调用以下接口  
// 通常如果动态区的位置或大小没有发生改变，不用删除  
screen.deleteAllDynamic();  
// 更新动态区  
screen.writeDynamic(dynRule, dAreaContent);
```

5. 联系我们

上海仰邦科技股份有限公司（总部）

地址：上海市徐汇区钦州北路 1199 号 88 幢 7 楼

电话：021-64554198 021-64554199

传真：021-64955166

邮编：200233

网址：<http://www.onbonbx.com>

仰邦（江苏）光电实业有限公司（昆山光电产业基地）

地址：江苏省昆山市开发区富春江路 1299 号

电话：0512-36912677 0512-36912688 0512-36912699

传真：0512-36912670

邮编：215300

二次开发支持

邮箱：dev@onbonbx.com

附录1