Appscomm SDK for Android

version: 1.1.0

author: zenglinguo@appscomm.cn

修改记录

版本	修改日期	修改人	说明
V 1.0.0	2015-02-11	zenglinguo	1. 第一个版本，支持 L28C/S 的设备
V 1.1.0	2015-05-25	zenglinguo	1. 加入 L11/L28T/H/W 协议 2. 连接参数可选 DN 号连接 3. DEMO 程序改成 Android Studio 示例 4. 支持 Android 5.0 5. 改用本地广播方式(用此 SDK 开发多个程序不会相互冲突)。

- 系统需求： Android4.3 或以上，蓝牙需支持 4.0.
- 支持设备类型： L11 / L28T/L28W/L28H/L28S/L28C

  固件要求：L11 要求固件版本不低于 2.04，L28T 要求固件版本不低于 1.01
- SDK 文件说明：

文件	说明
apps-android-openapi-sdk.jar	Appscomm SDK 库文件
Demo 目录	DEMO 源程序(android-studio)
Doc 目录	文档目录

- apps-android-openapi-sdk.jar 内模块说明：

类名	模块说明
BluetoothLeService	蓝牙通讯模块
ProtocolParser	协议及数据格式转换模块
SportData	运动数据类型

SDK 使用说明：

1.导入 JAR 包

1.复制 apps-android-openapi-sdk.jar 到 libs 文件夹，并添加 File Dependency.

2.请加入 android-support-v4 兼容包的支持。

2.在对应工程的 AndroidManifest.xml 中加入 蓝牙访问的权限

如：

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

3.在对应工程的 AndroidManifest.xml 中加入注册蓝牙通讯服务：

```
<service android:name="com.appscomm.bleutils.BluetoothLeService " />
```

4.在 Activity 创建时绑定蓝牙服务和注册特定事件广播接收器(Demo 中的 bindLeService)，

Activity 关闭时需要取消 bind,(Demo 中的 unbindLeService) (请用本地广播器接收，如 localBroadcastManager)

5.设置对应的通讯的协议，（如 setPedometerType(Pedometer_Type.L11)，设置通讯类型为 L11 ）；

6.发送命令前，先调用 connect 方法连接（需事先知道设备的 MAC 地址，如没有，则先扫描查找） （Connnect 支持 2 个输入参数，一个是 mac 地址，一个是 DN 号，推荐使用 mac 地址，这种速度最快，而 DN 查找方式实际是在后台有个蓝牙扫描过程，最后还是通过 mac 连接）

7.当收到服务发现的通知（ACTION_GATT_SERVICES_DISCOVERED）表示设备已就绪，可以发送命令了，对蓝牙命令最好采用一问一答方式接收和发送，这样流程好控制，且不会对蓝牙设备运算造成压力。

8.此 SDK 已封装了大部分常用的协议命令，直接调用即可，如果 SDK 新增加的协议未实现，也可以调用 sendRawDataToPedometer 方法，直接发送原始数据。

9.如需要跟踪发送和接收的数据，可在 Logcat 中，新建一个 “BluetoothLeService”的 TAG 值，打印日志。

10.蓝牙通讯地址(mac)获取后应保存到本地，方便下次连接而不需要重新扫描，如果是只知道 DN 号而没有 MAC 地址提供（可在调用 connect 时,mac 值传空，DN 号传入绑定的 DN 号，蓝牙服务会在后台自动匹配出 mac 地址，并进行连接，并会发出 ACTION_GATT_MACADDR 广播（mac 值在 EXTRA_MAC_DATA 中），广播接收收到此广播需要保存 mac 在本地，方便下次连接。

11．收到蓝牙超时的消息时，可以选择重发当前命令，而不需要重新连接

BluetoothLeService 类说明

常量值（Intent 值）	说明
ACTION_GATT_DISCONNECTED	蓝牙设备断开通知
ACTION_GATT_SERVICES_DISCOVERED	BLE 服务发现过程完成通知
ACTION_GATT_SERVICES_TIMEOUT	蓝牙设备通讯超时通知
ACTION_DATA_AVAILABLE	接收到蓝牙设备数据通知
EXTRA_DATA	附加数据（实际获取的数据）

ACTION_GATT_MACADDR	附加数据（返回蓝牙 MAC 地址，值在 EXTRA_MAC_DATA 中）
EXTRA_MAC_DATA	实际的蓝牙 Mac 地址值
EXTRA_DN_DATA	对应蓝牙地址的 DN 号码（20 位）

调用函数：

注：以下调用 **BluetoothLeService 的**函数全部为异步实现过程，一调用马上返回，要获取返回数据，请从对应广播的事件获取。

获取数据或结果： 数据返回 通过 BluetoothLeService 类中的 *ACTION_DATA_AVAILABLE 和 EXTRA_DATA 2 个 Intent 通知返回。具体看参考 DEMO 程序。*

蓝牙返回命令结果分为 2 种：

1. 响应消息返回：只用来表示命令设置是否成功，无需再从设备获取额外数据，

比如： 设置时间， 设置目标，

数据固定长度为 6 位，如：0x6E, 0x01, 0x01,命令码 , 结果, 0x8F

其中变化的只有命令码 和 结果,

命令码：请参考通讯协议查看。

结果码	说明
0x00	成功
0x01	失败
0x02	非法命令

2 .特定数据的返回： 用来获取蓝牙设备上的数据。

比如：获取电量，运动数据等。

BluetoothLeService 类调用说明

public void connect(String addr, String dn_NO)	/** 连接蓝牙设备 * @param addr 设备的MAC地址 如 00:00:00:00:00:01 * @param dn_NO: 对应的设备DN号码。一般为空 1.当mac不为空时，蓝牙服务会直接连接此mac，如果mac为空，指定DN号时，蓝牙服务会在后台搜索与DN号匹配的蓝牙设备，并广播出匹配的mac地址 * */
public void setSynTime()	/** * 同步手机当前时间到手环 */
public void getDeviceInfo(int getType)	/** * 获取设备硬件信息 * * @param getType * 0x00:获取产品类型， 返回 0x0A 表示 L28 0x01 获取设备唯一 id 号 0x02 获取固件版本信息 * 0x03 获取软件版本信息 */
public void getSportDataDetail()	/** * 获取运动数据详细 */
public void synPersonData(int sex, int year, int month, int day, int height, int weight)	/** * 设置个人信息到手环 * * @param sex * 0 男 1 女 * @param year * /month/day 出生日期 * @param height * 身高 cm * @param weight * 体重 kg * @ */

public void setGoalSteps(int steps)	/** * 设置目标步数 * * @param steps * 步数 */
public void setGoalCal(int Cal)	设置目标卡路里
public void setGoalDis(int Dis)	设置目标距离（单位 米）
public void getBatteryLevel()	/** * 获取电量 * */
public void getSportDataCount(int type)	/** * 获取运动或者睡眠记录条数 * * @param type * 0：运动记录 1：睡眠记录 * */
public void reSetFactoryMode()	/** * 重置设备为出厂模式 */
public void getPersonData()	/** * 返回设备上存储的个人信息 */
public void getSportDataTotal()	/** * 获取当日运动汇总 */
public void cleanAllReminder()	/** * 清除设备上所有的提醒 */
public void deleteAReminder(int hour, int min)	/** * 删除指定时间的提醒 * * @param hour * 小时 * @param min

	* 分钟 */
public void addAReminder(int id, int hour, int min, String repeat)	/** * 手动加提醒 * * @param id * 提醒类型 0，吃饭 1：吃药；2：运动；3：睡觉；4：清醒；5:自定义；； * @param hour * @param min * @param repeat * 8 位字符串 分别表示星期日到星期一，1 为提醒，0 不提醒，比如星期 1,3 提醒,则为 00000101(最左边一直为 0） */
public void getSleepDataDetail()	/** * 获取睡眠数据详细 * */
public void setAutoSleepRange(boolean isEnAutoSleep,int startHours,int startMins,int endHours,int endMins)	/** * @param isEnAutoSleep 是否开启自动睡眠 * @param startHours 自动进入睡眠的小时 eg: 22 * @param startMins 自动进入睡眠的分钟 eg: 00 * @param endHours 自动退出睡眠的小时 eg: 07 * @param endMins 自动退出睡眠的分钟 eg: 00 */
public void setManualMode(int mode)	/** * 设置数据同步的手动模式 * * @param mode * 0x01:删除运 动数据 0x02：删除睡 眠数据

	0x03：设置成 自动删除运 动数据和睡眠数据命令 0x04：设置成 * 手动删除运 动数据和睡眠数据命令 */
public void setTimeType(boolean is24H,boolean isKm,boolean isShowDate, boolean isShowBattery, int dateType)	* 设置 时间和英里 单位 * * @param is24H 是否 24 小时显示 * @param isKm 是否千米显示（不支持的固件，此参数无用） * @param isShowDate 是否显示日期（不支持的固件，此参数无用） * @param isShowBattery 是否显示电量（不支持的固件，此参数无用） * @param dateType 日期显示方式（不支持的固件，此参数无用） * 0：DD/MM/YY * 1：MM/DD/YY * 2：YY/DD/MM
public void setSleepStatus(int status)	
public void getWatchId()	/** * 获取设备序列号 20 位 */
public void sendRawDataToPedometer(byte abyte0[])	/** 直接发送命令给手环,请按协议格式发送 * @param abyte0 要发送的数据 */
public void setAntiActivtiy (boolean isAntiSwON,String repeatWeeks,int interval,int startHour,int startMin, int endHour,int endMin,int stepLimit)	设置静坐提醒命令： 1 isAntiSwOn: 是否打开(Boolean) 2 repeatWeeks: 7 位长度的字符串，分别表示星期 1 到星期天是否设置，1 表示开，0 表示关，（eg: 星期 2，星期 3 打开 值为“0110000” 2. interval : 检测的间隔（分钟） 3. startHour,startMin: 开始的小时，分钟 4. endHour,endMin: 结束的小时，分钟

	5. stepLimt 检测的步数，如 100 步
public void setANCS_SW (boolean isCallON,boolean isMisCallON,boolean isSMSON,boolean isEmailON,boolean isSocialON, boolean iscalendarON,boolean isAntiLostON)	设置 ANCS 消息推送开关： isCallON 来电 isMisCallON 未接来电 isSMSON 短信 isEmailON 邮件 isSocialON 社交媒体 iscalendarON 日历 isAntiLostON 防丢失
public void setPushUSocNum(int mNum)	@param mNum 社交消息的数量
public void setPushUCalNum(int mNum)	@param mNum 日程提醒的数量
public void setPushUEmailNum(int mNum)	* @param mNum 未读邮件的数量
public void setPushSMSNum(int mNum)	@param mNum 未读短信的数量
public void setPushUCallNum(int mNum)	@param mNum 推送未接来电的数量
public void setPushCallName(String name)	param name 显示的来点号码，或者来电姓名（仅支持英文名字）

ProtocolParser类常量说明：

常量名	说明
VERSION	协议版本
PROTOCOLNAME	协议名称

ProtocolParser类 函数说明

public static SportsData parseSportDetailData(byte[] bytes)	解析运动详细数据， 输入： 字符数组 输出：返回SportData类型 （步数，卡路里）
public static SportsData parseSportTotalData(byte[] bytes) {	解析运动汇总数据， 输入： 字符数组

	输出：返回SportData类型 （步数，卡路里）
public static byte[] byteArrayReverse(byte[] bs)	byte[] 高低位转换
public static int byteReverseToInt(byte[] b)	2进制的byte[]高低位置换数组转int类型
public static String bytes2HexString(byte[] b)	打印 byte[]