```
/**
 *
 * The Turntable Controller for Android Project
 * 深圳康信技术有限公司 版权所有
 * Shenzhen ComXim Technologies Co., Ltd.  ComXim® Copyright
 *  Copyright (C) 2018
 *
 **/


private List<String> StrListBleSend   = new ArrayList<String>();// 定义一个 List 用来暂存截取之后的命令队列
private Handler mSendHandler = new Handler(); // 定义一个Handler, 用Handler.postDelayed 来实现定时发送命令
private int miBlePkgLength  = 20 ; // 定义一个常量控制截取长度，协议有变化则修改此处截取长度，也可以顶一个宏。
private int miBleSendInterval  = 25 ; //定义一个常量控制发送的时间间隔，25mm 是经过验证的，太短可能处理不过来，建议不要大于 50ms，虽然转盘最长可以等待 200ms，但是蓝牙空口的传输可能授信号干扰有延时建议预留足够的裕量

    /**
     *  BLE 设备发送函数
     // 发送处理大于 20 字节的数据
     // 背景：蓝牙 4.0 以后 BLE 支持低功耗传输，传输距离更远，同时功耗更低，代价是限制了每次传输字节数，默认情况下限制了 MTU 最大 20 字节留给用户
     // 即用户单次发送的字节数不大于 20 字节。因此传送超过 20 字节时候有两种方式：1 . 和对端设备启动协议 MTU 自协商机制，但是此方法依赖 两点 ：a.双方
     // 的蓝牙版本都支持自协商，2. 协商失败后的处理，因此，逻辑和代码过于复杂。2. 发送端自己将超过 20 字节的传输报文截取 20 字节一段，分多次发送。
     // 该方法一步变应万变，适应所有蓝牙版本。
     */

    //
    // CT_CMD_str 是要传送的完整的 CT 命令，将你要发送的命令交给此函数
    private void DoSendStrList(String CT_CMD_str)
    {
        //
        StrListBleSend.clear();// 每次发动前清空暂存队列
        // 向列添加新的命令并截取，添加时候调用字符串截取函数 getStrListByLen 按照 miBlePkgLength 函数截取
        StrListBleSend.addAll(getStrListByLen(CT_CMD_str.trim(),miBlePkgLength));
```

```
        mSendHandler.post(SendRunnable); // 交给 mSendHandler
    }
```

```
/**
 * 蓝牙发送线程
 */
Runnable SendRunnable = new Runnable()
{
    @Override
    public void run()
    {
        if (StrListBleSend.size()>0)  // 交给检查是否发送队列为空
        {
            String substr = StrListBleSend.remove(0); // 取走队列里的第一个截取的字符，并移除第一个
            //设置要发送数据内容
            MyAttCharacteristic.setValue(substr);
            //往蓝牙 mBLE 模块写入数据
            mBLE.writeCharacteristic(MyAttCharacteristic);
            //检查是否需要继续发送
            if(StrListBleSend.size()>0) // 如果队列不为空，那就间隔 miBleSendInterval 毫秒再次调本段代码继续发送
            {
                mSendHandler.postDelayed(this, miBleSendInterval);
            }
        }
    }

};
//
```

以下为截取处理字符串的函数。

```
/**
 * 把原始字符串分割成指定长度的字符串列表
 *
 * @param inputString
 *            原始字符串
 * @param length
 *            指定长度
 * @return
 */
public static List<String> getStrListByLen(String inputString, int length) {
    int size = inputString.length() / length;
```

```
    if (inputString.length() % length != 0) {
        size += 1;
    }
    return getStrList(inputString, length, size);
}

/**
 * 把原始字符串分割成指定长度的字符串列表
 *
 * @param inputString
 *            原始字符串
 * @param length
 *            指定长度
 * @param size
 *            指定列表大小
 * @return
 */
public static List<String> getStrList(String inputString, int length,int size)
{
    List<String> list = new ArrayList<String>();
    for (int index = 0; index < size; index++) {
        String childStr = substring(inputString, index * length,
                (index + 1) * length);
        list.add(childStr);
    }
    return list;
}

/**
 * 分割字符串，如果开始位置大于字符串长度，返回空
 *
 * @param str
 *            原始字符串
 * @param f
 *            开始位置
 * @param t
 *            结束位置
 * @return
 */
public static String substring(String str, int f, int t) {
    if (f > str.length())
        return null;
    if (t > str.length()) {
        return str.substring(f, str.length());
```

```
    } else {
        return str.substring(f, t);
    }
}
```