

# HPRT蓝牙扫描头M300 SDK\_v1.07.02文档

---

## 1.扫描头说明

该蓝牙扫描头是通过HID和BLE的透传来实现通讯，HID来获取扫描头扫描到的数据，透传来实现扫描头和android设备的交互

## 2.SDK使用

添加SDK jar包，注册蓝牙服务

```
<service android:name="com.example.bluetooth.prt.BluetoothLeService"
android:enable="true"/>
```

## 3.接口说明

### 3.1连接扫描枪

```
void buleconnect(Context context, HPRTHelper.onConnect onConnect)
```

例：

```
HPRTHelper mHelper = HPRTHelper.getHPRTHelper(context);
//注意：连接之前传入蓝牙设备
mHelper.setDevice(bluetoothDevice);
mHelper.buleconnect(context, new onConnect(){
    @Override
    public void succeed(){
        //todo
    }
    @Override
    public void failure(){

    }
});
```

### 3.2断开连接

```
void disconnect(onDisconnect ondisconnect)
```

例：

```
mHelper.disconnect(new onDisconnect(){
    @Override
    public void succeed(){

    }
});
```

### 3.3配置设备

```
boolean setSetting(byte[] data)
```

data[]	value
0	0x02
1	0x10
2	长滴时长，取值范围3-30，单位0.1秒
3	短滴时长，取值范围1-5，单位0.1秒
4	短滴间隔时长，取值范围1-5，单位0.1秒
5	短滴次数，取值范围1-10
6	震动时长，取值范围1-30，单位0.1秒
7	结尾回车： 0x00 扫描数据结尾不添加回车 0x01扫描数据的结尾添加回车
8	大小写字母转换： 0x00 所有字母保持原样 0x01 所有字母转换为小写 0x02所有字母转换为大写
9	短滴和长滴的duty，取值范围1-255
10	震动的duty,取值范围1-255（暂未启用）
11	无按键关闭扫描头时间，取值范围3-60秒
12	无按键自动关机时间，取值范围3-60分钟
13	自动连续扫描模式 0x00 关闭自动连续扫描模式 0x01 开启自动连续扫描模式
14	声音 0x00 关闭声音 0x01 开启声音
15	震动 0x00 关闭震动 0x01 开启震动
16	连续扫描间隔时间高位，十六进制编码，共两字节
17	连续扫描间隔时间低位，十六进制编码，共两字节 data[16]&data[17]取值范围100-5000,单位ms
18	校验码，以上所有字节的累加和

### 3.3读取设备配置

```
void getSettingData(onReadData readdata)
```

返回值说明

data[]	value
0	0x81
1	0x10
2	长滴时长，取值范围 3-30，单位 0.1 秒（默认值=5）
3	短滴时长，取值范围 1-5，单位 0.1 秒（默认值=1）
4	短滴间隔时长，取值范围 1-5，单位 0.1 秒（默认值=2）
5	短滴次数，取值范围 1-10（默认值=1）
6	震动时长，取值范围 3-30，单位 0.1 秒（默认值=5）

| 7 | 结尾回车  
0x00 扫描数据的结尾不添加回车 0x01 扫描数据的结尾添加回车（默认值=0x01） || 8 | 大小写字母转换  
0x00 所有字母保持原样（默认值=0x00） 0x01 所有字母转换为小写 0x02 所有字母转换为大写 || 9 | 短滴和长滴的 duty，取值范围 1-255（默认值=50） || 10 | 震动的 duty，取值范围 1-255（默认值=50）（暂未启用） || 11 | 无按键关闭扫描头时间，取值范围3-60秒（默认值=10秒） || 12 | 按键关闭 || 13 | 自动连续扫描模式  
0x00 关闭自动连续扫描模式  
0x01 开启自动连续扫描模式 || 14 | 声音  
0x00 关闭声音  
0x01 开启声音（默认值=0x01） || 15 | 震动  
0x00 关闭震动  
0x01 开启震动 || 16 | 连续扫描间隔时间高位，十六进制编码，共两字节 || 17 | 连续扫描间隔时间低位，十六进制编码，共两字节  
data[16]&data[17]取值范围100-5000，单位毫秒 || 18 | 校验码，以上所有字节累加和 |

### 3.4控制灯光

```
boolean printLight(int lightstype)
```

参数说明：

lightstype  
=1 常亮  
=2 闪烁  
=3 常灭

### 3.5控制震动时间

```
boolean printShock(int time)
```

参数说明：

time：震动时间，取值范围1-30，单位0.1秒

### 3.6获取版本号

```
void setReaddeviceversion(onReaddeviceversion onread)
```

返回值说明：

data：最后两位代表版本号

例：

```
mHelper.setReaddeviceversion(new onReaddeviceversion() {  
    @Override  
    public void succeed(byte[] data) {  
        tv_version.setText("版本号："+data[data.length-1]+data[data.length-2]);  
    }  
  
    @Override  
    public void failure() {  
        tv_version.setText("版本号：点击获取");  
    }  
});
```

### 3.7获取电量

```
void getElectricity(onElectricity onE)
```

返回值说明:

data:

data[2] 电池电压高位

data[3] 电池电压低位

data[4] 电量百分比

例：

```

mHelper.getElectricity(new onElectricity() {
    @Override
    public void succeed(byte[] data) {
        tv_electricity.setText("电量："+data[4]+"%");
        String parseHex= HPRTHelper.bytetohex(data[2])+HPRTHelper.bytetohex(data[3]);
        tv_voltage.setText("电压："+Integer.parseInt(parseHex.trim(),16)+"mV");
    }

    @Override
    public void failure() {

    }

});

```

### 3.8设置工作模式

```
boolean setWorkModel(int stype)
```

参数说明：

stype：1.GATT模式

2.HID模式

例：

```

if(mHelper.setWorkModel(GATT)){
    Util.ToastMessage(mContext, "设置成功");
}else{
    Util.ToastMessage(mContext, "设置失败");
}

```

### 3.9获取工作模式

```
void getWorkModel(onWorkModel onw)
```

返回值说明：

data:

data[2]: 1 GATT模式

2 HID模式

例：

```

mHelper.getWorkModel(new onWorkModel() {
    @Override
    public void succeed(byte[] data) {
        if (data[2]==1) {
            tv_work_model.setText("模式：GATT");
        }else if(data[2]==2){

```

```

        tv_work_model.setText("模式：HID");
    }else{

    }
}
@Override
public void failure() {

}
});

```

### 3.10关机

```
boolean setPowerOff()
```

例：

```

if(mHelper.setPowerOff()){
    Util.ToastMessage(mContext,"发送关机成功");
}

```

### 3.11恢复出厂设置

```
boolean setRestorefactory()
```

例：

```

if(mHelper.setRestorefactory()){
    Util.ToastMessage(mContext,"恢复出厂设置成功");
}

```

### 3.12GATT模式下获取扫描数据

```
void getGattData(onGattdata onG)
```

返回值说明

data 扫描数据，ASCII编码

例：

```

mHelper.getGattData(new onGattdata() {
    @Override
    public void getdata(byte[] data) {
        ed_scandata.setText(Util.byteASCIIstr(data));
    }
});

```

### 3.13设置震动开关

```
boolean setShock(int shock)
```

参数：

shock: 1 开启

2 关闭

例：

```
if(mHelper.setShock(shock)){
    Util.ToastMessage(mContext,"修改震动成功");
}else{
    Util.ToastMessage(mContext,"修改震动成功");
}
```

### 3.14获取震动开关

```
void getShock(onShock ons)
```

返回值说明：

data：

data[2]: 0 关闭

1开启

例：

```
mHelper.getShock(new onShock(){
    @Override
    public void succeed(byte[] data) {
        if (data[2]==0) {
            Util.ToastMessage(mContext,"震动：关闭");
        }else if(data[2]==1){
            Util.ToastMessage(mContext,"震动：开启");
        }
    }
    @Override
    public void failure() {

    }
});
```

### 3.15修改设备名称

```
boolean setDeviceName(String deviceName)
```

例：

```

if(mHelper.setDeviceName("HPRT SCAN")){
    Util.ToastMessage(mContext,"修改成功");
}else{
    Util.ToastMessage(mContext,"修改失败");
}

```

### 3.16获取设备名称

```

void getDeviceName(onDeviceName onD)

```

例：

```

mHelper.getDeviceName(new onDeviceName(){
    @Override
    public void succeed(byte[] data) {
        tv_device_name.setText("设备名称："+Util.byteASCIIstr(data));
    }

    @Override
    public void failure() {

    }
});

```

### 3.17设置暂存开关

```

boolean setTemporarilySave(int TSave)

```

参数说明：

TSave：0 关闭

1 开启

例：

```

if(mHelper.setTemporarilySave(0)){
    Util.ToastMessage(mContext,"关闭成功");
}else{
    Util.ToastMessage(mContext,"关闭失败");
}

```

### 3.18清除暂存数据

```

boolean cleanTemporarilySaveData()

```

例：

```

if(mHelper.cleanTemporarilySaveData()){
    Util.ToastMessage(mContext,"清除成功");
}else{
    Util.ToastMessage(mContext,"清除失败");
}

```

### 3.19读取暂存统计数据信息

```

void getTemporarilySaveCount(onTemporarilySaveCount onT)

```

返回值说明：

isSvaeOpen: true 开启 false 关闭

savedCount：已存条数

leftCount: 剩余可存条数

例：

```

mHelper.getTemporarilySaveCount(new onTemporarilySaveCount() {
    @Override
    public void succeed(boolean isSvaeOpen, int savedCount, int leftCount) {
        if(isSvaeOpen){
            Util.ToastMessage(mContext,"暂存功能开启");
        }else{
            Util.ToastMessage(mContext,"暂存功能关闭");
        }
        tv_save_count.setText("已存条数："+ savedCount+ "    剩余条数："+leftCount);
    }

    @Override
    public void failure() {

    }

});

```

### 3.20暂存数据上传

```

void upLodeTemporarilyData(onUpLodeTemporarilyData onUpLode)

```

返回值说明：

data 暂存数据内容

例：

```
mHelper.upLodeTemporarilyData(new onUpLodeTemporarilyData() {  
    @Override  
    public void succeed(byte[] data,int tap) {  
        Util.ToastMessage(mContext,Util.byteASCIIstr(data));  
    }  
  
    @Override  
    public void failure() {  
        // TODO Auto-generated method stub  
    }  
});
```

### 3.21修改扫描时间间隔

```
boolean setIntervalTime(int time)
```

参数说明：

time：间隔时间，范围100-5000，单位ms

例：

```
if(mHelper.setIntervalTime(time)){  
    Util.ToastMessage(mContext, "修改扫描时间间隔成功");  
}else{  
    Util.ToastMessage(mContext, "修改扫描时间间隔失败");  
}
```