



展锐基线系统休眠唤醒通知STR状态方案

一、背景

- 模块在休眠前，相关进程需要保存数据，因此系统需要提供休眠唤醒通知上层app和native/hal层STR状态
- 和展锐沟通确认，目前基线休眠未使用VHAL和CarPowerManager，如果需要知道休眠唤醒状态，需要监听亮灭屏广播

二、系统休眠唤醒通知STR状态方案

基于(一)背景，需要依据亮灭屏来通知STR状态，针对上层app和native/hal层给出通知STR状态方案。

!! 核心总结：动态注册广播监听屏幕事件、获取 WakeLock 防止保存数据时 CPU 休眠，以及正确处理权限和组件生命周期

2.1 通知APP STR状态

Android 系统会在屏幕状态改变时发送全局广播，可以通过动态注册 BroadcastReceiver 来监听屏幕亮灭事件。

1 监听 Intent.ACTION_SCREEN_ON、Intent.ACTION_SCREEN_OFF，此广播必须在代码中动态注册（建议在Service 中）

代码注册广播示例，非正式代码，仅作参考

```
1  BroadcastReceiver screenReceiver = new BroadcastReceiver() {
2      @Override
3      public void onReceive(Context context, Intent intent) {
4          if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
5              // 亮屏
6          } else if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
7              // 灭屏
8          }
9      }
10 };
11
12 IntentFilter filter = new IntentFilter();
13 filter.addAction(Intent.ACTION_SCREEN_ON);
14 filter.addAction(Intent.ACTION_SCREEN_OFF);
```

```
15 registerReceiver(screenReceiver, filter);
```

2 添加唤醒锁、处理数据、释放唤醒锁

添加唤醒锁、释放唤醒锁示例，非正式代码，仅作参考

```
1  <uses-permission android:name="android.permission.WAKE_LOCK" /> //权限，如果还需要其他权限自行添加
2
3  private PowerManager.WakeLock wakeLock;
4  private PowerManager powerManager = (PowerManager) getSystemService(Context.POWER_SERVICE);
5
6  private void acquireWakeLock() {
7      if (wakeLock == null) {
8          // 使用 PARTIAL_WAKE_LOCK 保证 CPU 运行即可，不影响屏幕状态
9          wakeLock = powerManager.newWakeLock(
10              PowerManager.PARTIAL_WAKE_LOCK, //关于锁的类型，请自行查找代码按需使用
11              "XXXX");
12      }
13      //如果需要，可以给锁指定超时时间
14      if (wakeLock != null && !wakeLock.isHeld()) {
15          wakeLock.acquire(10 * 60 * 1000L);
16      }
17  }
18
19  private void releaseWakeLock() {
20      if (wakeLock != null && wakeLock.isHeld()) {
21          wakeLock.release();
22      }
23      wakeLock = null;
24  }
```

2.2 通知native/hal STR状态

待补充