**一、Wifi**

**1、初始化**

WifiController.instance().init(Context context);

**2、判断Wifi是否打开**

boolean isWifiOpen = WifiController.instance().isWifiOpen();

**3、获取连接Wifi信息**

WifiInfo wifiInfo = WifiController.instance().getConnectionWifiInfo();

**4、打开Wifi**

boolean success = WifiController.instance().openWifi();

**5、关闭Wifi**

Boolean success = WifiController.instance().closeWifi();

**6、连接指定Wifi**

WifiController.instance().connectToWifi(WifiConfiguration wifiConfiguration, WifiConnectCallback callback);

**7、断开wifi**

boolean success = WifiController.instance().disconnectWifi();

**8、获取Wifi列表**

List<WifiDetail> wifiDetails = WifiController.instance().getWifiDetails();

**9、设置飞行模式**

WifiController.instance().setAirplaneMode(Context context, boolean ebable);

**10、是否开启飞行模式**

boolean isOn = WifiController.instance().isAirplaneModeOn(Context context);

**11、设置移动网络开关**

WifiController.instance().setMobileNetMode(Context context, boolean ebable);

**12、是否开启移动网络**

boolean isOn = WifiController.instance().isMobileNetOpen(Context context);

**二、热点**

**1、判断热点是否打开**

boolean isWifiApOpen = WifiController.instance().isWifiApOpen();

**2、打开热点**

boolean success = WifiController.instance().openWifiAp();

**3、关闭热点**

boolean success = WifiController.instance().closeWifiAp();

**4、获取热点连接设备**

*(1) Actvity.onResume()注册热点监听器*

WifiController.instance().registerWifiApReceiver(Context context, IWifiApListener listener);

(*2)Activity.onPause()注销热点监听器，注册和注销必须成对存在*

WifiController.instance().unregisterWifiApReceiver(Context context, IWifiApListener listener);

*(3).扫描热点连接设备*

WifiController.instance().scanWifiApClients();

**5、配置热点**

boolean success = WifiController.instance().createWifiApConfiguration(WifiApServer wifiApServer);

**6、获取热点数据**

WifiApServer wifiApServer = WifiController.instance().getWifiApConfiguration();

**三、蓝牙**

**1、初始化**

BluetoothController.instance().init();

**2、判断蓝牙是否打开**

boolean isOpen = BluetoothController.instance().isBluetoothOpen();

**3、打开蓝牙**

boolean success = BluetoothController.instance().openBluetooth();

**4、关闭蓝牙**

boolean success = BluetoothController.instance().closeBluetooth();

**5、允许蓝牙被检查**

BluetoothController.instance().enableDiscover(Activity activity, int duration, int request);

**6、取消蓝牙被检查**

BluetoothController.instance().disableDiscover(Context context);

**7、注册扫描器Activity.onResume()**

BluetoothController.instance().registerScanner(Context context)

**8、注销扫描器Activity.onPause()**

BluetoothController.instance().unregisterScanner(Context context)

**9、扫描蓝牙设备**

BluetoothController.instance().startScan(IBluetoothListener listener);

**10、取消扫描蓝牙设备**

BluetoothController.instance().stopScan();

**11、获取本机蓝牙名**

String ownBluetoothName = BluetoothController.instance().getOwnBluetoothName();

**12、设置本机蓝牙包名**

boolean success = BluetoothController.instance().setOwnBluetoothName(String name);

**四、设置**

**1、获取屏幕亮度：亮度值 0~255**

int screenBrightness = SettingsController.instance().getScreenBrightness(Context context);

**2、设置屏幕亮度：亮度值 0~255**

boolean success = SettingsController.instance().setScreenBrightness(Context context, int brightness);

**3、获取手机运营商**

String operator = SettingsController.instance().getOperator(Context context);

**4、获取音量大小：**

*a.按键音量AudioManager.STREAM\_VOICE\_SYSYEM*

*b.媒体音量AudioManager.STREAM\_MUSIC*

int volume = SettingsController.instance().getVolume(Context, context, int streamType);

**5、设置音量大小**

SettingsController.instance().setStreamVolume(Context context, int streamType, int volume, Boolean playSound);

**6、获取系统语言**

Locale[] languages = SettingsController.instance().getSystemLanguages();

**7、获取手机运行内存**

long size = SettingsController.instance().getTotalMemorySize(Context context);

**8、获取手机内部剩余存储空间**

long size = SettingsController.instance().getAvailableInternalMemorySize();

**9、获取手机内部总存储空间**

long size = SettingsController.instance().getTotalInternalMemorySize();

**10、监听电池电量变化**

SettingsController.instance().queryBatteryCapacity(Context context, OnBatteryListener listener);

**11、监听信号强度变化**

SettingsController.instance().querySignalStrength(Context context, OnSignalStrengthListener);

**12、设置系统时间**

SettingsController.instance().setSystemTimes(Context, long);

**13、设置12小时展示样式**

SettingsController.instance().set12Hour(Context);

**14、设置24小时展示样式**

SettingsController.instance().set24Hour(Context);

**15、设置自动确定时间**

SettingsController.instance().setAutoModifyTime(Context context, boolean auto);

**16、设置自动确定时区**

SettingsController.instance().setAutoModifyTimeZone(Context ntext, boolean auto);

**17、获取网络状态（2/3/4G）**

int networkType = NetworkUtils.getNetworkType(Context context);

**重要：**

**设置时间和设置飞行模式、移动网络开关等需要系统权限。**

**在AndroidManifest.xml头加入 android:shareUserId=”android.uid.system”,并且需要打系统签名，确保uid和系统一直，否则apk一直安装不了。**

**另外定位服务和读取手机状态权限必须打开，否则wifi、蓝牙、热点、亮度、声音等功能无法正常使用。**