

# jenny/IVIANDLIBS-314: [HVAC] Peek bar is shown when fan speed is changed from CID

deepwaterooo

May 19, 2022

## Contents

<b>1 jenny-IVIANDAPPS-6293: Implementation of ADB activation from CID for EU Homologation</b>	<b>1</b>
1.1 Feature 实现要求: Description: Please implement the user interface on CID to activate or deactivate the ADB as per the Figma/UX Link as mentioned below. While implementing the feature, the following aspects needs to be considered:	1
1.2 linking 相关逻辑	3
1.3 首先找到 LucidCarSettings app 中的最相关的界面 driving/DriveSettingsFragment.java	3
1.4 DriveSettingsViewContract.java	4
1.5 DriveSettingsFragment.java 的 layout 配置: R.layout.drive_settings_frag;	4
1.6 DriveSettingsFragment.java 中的主要逻辑: 那么与会与 UI 层的 View Contract 相关? 这里的源码还没有整理/没划重点	6
1.7 LucidCarSettings 应用中 HomeActivity 中与 Controller 控制器相关/UI 回调相关/bug 相关的最重要逻辑	9
1.8 UnitPrefContoller.java 相关的 LucidCarSettings 中的控制器	11
1.9 先去把信号找到, 确定好是能够传上来的! 不要等到最后一分钟: 应该是可以传得上来的	12
1.10BcmController.java 具体三个信号的接收与监听	12
1.11AsyncResources.java 的 mBcmListener	16
1.12IMarket IRegion: lucid-carctrl-lib UnitPrefController.java	16
1.13DriveModeController.java	20
1.14adaptive 的逻辑	21
1.15LucidCarSettings 应用中 HomeActivity 中重要逻辑再稍微熟悉一下	21
<b>2 jenny/IVIANDAPPS-6093: [SCCM]{SW-issue} - Steering wheel right side buttons unoperational</b>	<b>24</b>
2.1 SteeringWheelController.java should be involved for these volume buttons (According to Soumya)	24
2.2 目前 LucidSystemUI 中处理过的几个点面: 这三个来源分别是什么意思呢?	25
2.3 logs	26
2.4 Concepts to be understood first	26
2.5 a few trials from the car	28
2.5.1 问题: 即是车的 onChangeEvent(), 为什么得到的值, 进一步传时来的参数又是 false 呢?	28
2.6 today's testing	29
2.7 Leakcary: Revanth says we are having these memeory leaks	31

<b>3 jenny-IVIANDLIBS-342: Peek bar does not show up if we change temp/fan using hard buttons: 涉及原来安卓 9 的三个 Controller 与 VHAL 的交互</b>	<b>31</b>
3.1 这个应该有几个方面的原因:	31
3.2 这里 Soumya 的解释是说, 只在 mBcmListener.init() 初始化的时候, 我们才能收到最初的状态值, 但并不是真正的回调; 但是昨天我在车里测的时候, 每次好像是都可以收到回调的, 今天需要再去车里好好测一下, 强调这一点, 确认到底是什么状态的	32
3.3 VHAL TEAM 说他们现在不传这个信号给我们了, 我们需要自己来合并!!	33
3.3.1 Current bugs are:	35
3.4 现在再理一下思路: 感觉之前我可能没有读懂代码, 为什么 CID 会显示呢?	35
3.5 S2R does NOT work for RC cars. they are moving some cars here	36
<b>4 jenny/IVIANDLIBS-314: [HVAC] Peek bar is shown when fan speed is changed from CID: 这个比较容易改</b>	<b>36</b>
<b>5 Contollers</b>	<b>36</b>
5.1 IcrHvacPeekWindow: 我觉得这里面很多代码我没有看懂, 不只要想 showWindow(), 还需要去想 hideWindow() 的逻辑	36
5.2 GearController.GearListener	37
5.3 ParkAssistController.ParkAssistListener	37
5.4 BcmController.BcmListener	37
5.5 HvacController.HvacListener	38
<b>6 jenny/IVIANDLIBS-342: Peek bar does not show up if we change temp/fan using hard buttons</b>	<b>40</b>
6.1 Log snapshots 以及可能有的问题	40
6.1.1 some kind of apk build path specification error, but does NOT matter that much	40
6.1.2 Broken Notification Channel: foreground service LucidSysUiService is NOT launched properly due to broken channel, but it does NOT matter	40
6.1.3 在 framework 层, 有 SystemUI 应用?? 它应该是负责 broken Notification Channels, 可以问一下 Soumya	41
6.1.4 LifecycleOwner 相关的: 需要自己再深入地理解一下	41
6.2 Error message 1: AndroidManifest.xml Proguard + memory leak? 应用是某个地方的路径写得不对, 但好歹它的关系不大, LoadedApk: java.lang.ClassNotFoundException: Didn't find class "com.lucid.car.sysui.lucid" on path: DexPathList[[zip file "/system/priv-app/LucidSystemUI/LucidSystemUI.apk"],nativeLibraryDirectories=[/system/priv-app/LucidSystemUI/lib/arm64, /system/priv-app/LucidSystemUI/LucidSystemUI.apk!/lib/arm64-v8a, /system/lib64, /product/lib64, /system/lib64, /product/lib64]]	42
6.3 crash at system user 0 at bery beginning, if this is the case, the log is meaningless	42
<b>7 logs from today to support all the ideas</b>	<b>42</b>
7.1 收听 BOOT_COMPLETED 的几个事件与服务与先后顺序: LucidSystemService, LucidDmsService, com.lucid.car.carsettings.BootReceiver, (InjectBaseVersionS2110: onReceive() : intent=android.intent.action.BOOT_COMPLETED), com.lucid.car.sysui.pinToDrive	
7.2 在我想要切换用户的瞬间发生的	43
7.3 ACTION_USER_XXXX related: current user (MESSAGE/ACTION_)USER_UNLOCKED, then prev user (ACTION_)USER_STOPPED	43
7.4 all onCreate() related: 从 adb reboot 开始抓的: “onCreate” 可能没有抓全	43
7.5 LucidSysUiService.java launching logics	46
7.6 LucidSystemService launches LucidSysUiService	47
7.7 UI VHALS are connecting correctly	47

7.8 IcrHvacPeekWindow initial values . . . . .	47
7.9 consitantly updating outside temp, but how do i know if BcmController is NOT break afterwards? . . . . .	47
7.10switching user process . . . . .	48
7.11哪里蹦出来的 BcmControllerLite 呢? . . . . .	48
<b>8 jenny/IVIANDLIBS-349: Merge Unit Tests for SystemUI and Settings Apps: for LucidCarSettings App</b>	<b>49</b>
8.1 Unit test ContentProvider 举例 . . . . .	50
<b>9 jenny/IVIANDLIBS-351: Merge Unit Tests for SystemUI / Connectivity: for lucid- bluetooth-lib</b>	<b>50</b>

## 1 jenny-IVIANDAPPS-6293: Implementation of ADB activation from CID for EU Homologation

- <https://lucidmotors.atlassian.net/browse/IVIANDAPPS-6293>
- 必要相关文档:<https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/2733834385/Adaptive+Driving+Beam+ADB+Adaptive+Front-lighting+System+AFS>
- CCC BCM 信号字典: <https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/1369538838/CCC+Signals>
- Figma 相关的多个图:<https://www.figma.com/file/HSdA2inAtWMU4k0GnFgXwu/Vehicle-Settings?node-id=2%3A35>

### 1.1 Feature 实现要求: Description: Please implement the user interface on CID to activate or deactivate the ADB as per the Figma/UX Link as mentioned below. While implementing the feature, the following aspects needs to be considered:

- System Requirement: <https://lucid.jamacloud.com/perspective.req#/containers/1732461?projectId=124>
- Figma / UX Link: <https://www.figma.com/file/HSdA2inAtWMU4k0GnFgXwu/Vehicle-Settings?node-id=278%3A247457>
- The ADB shall be enabled by default.
  - When the toggle button is set to OFF position, CCC shall encode {CCC\_AutHiBeamSetReq} to “0 (Disabled)”
  - When the toggle button is set to ON position, CCC shall encode {CCC\_AutHiBeamSetReq} to “1 (Enabled)”
- The CCC shall subscribe to [BCM\_AutHiBeamSetSts] to know the status of the HBA/ADB from BCM and adjust the state of the UI toggle button on CID accordingly. 应用层 UI 需要监听这些信号，以便与 BCM 硬件同步。
- The ADB settings shall be saved to the driver’s user profile. 这些个用户个性化配置需要写入用户（配置数据中?????）

- The CCC shall store last valid value of [CCC\_AutHiBeamReq] in persistent memory. BCM also stores the ADB settings persistently. In case of conflict, [BCM\_AutHiBeamSetSts] shall have priority over value loaded from CCC persistence memory. 应用层需要把这些最新数据记住（还是要写到哪里呢?????? persistent memory）；BCM 也需要记住（这个不该我管了吧?），冲突的时候以 BCM 的为准
- Implement the ADB activation user interface for the EU Region i.e. {IRegion} is encoded as “REU” .
- Note: On the CID, the menu & toggle button that is used to activate the ADB for EU occupies the same space with the menu & toggle button that is used to activate the HBA for USA.
- 按照上面这个说的，好像它还需要分左驾和右驾，得把这里的逻辑给理清楚了。。。。。。。。Left steering wheel stalk: Right steering wheel stalk: 左方向盘杆: 右方向盘杆: ???????
- Jama access ticket needs to be created, why 6 weeks, nobody even helps mention that I would need this access? maybe not as important, but still good to have access?
- for bypass TCU ticket, Paul suggests I could Pull Paul in the chat with Revanth, so that he can help me verify if I created the ticket correctly, and the ticket could be finally resolved.
- 要把 Jama Access 弄好，没有 Jama 文档的理论支持，好像难度不会小，因为它是 single source of truth，没有它，我无法确保我改对了改得符合它的要求呀- 需要确定只在 EU 版本中才会有这个 Feature 出现。IRegion = EU, AndroidManifest.xml too
- 这个 bug 也涉及 VHAL 传上来的一个信号，我也需要赶快确认，我们是否能够接收到这个信号
- 在确认我们能够接收到这个信号之前，我首先需要知道，这个应该能够传上来的信号到底是什么，是在哪些 Controller 中监听????? 这两条，需要赶快确定，可别拖到找不出来没时间什么的
- 看到那个需要通过 Power Cycle 的地方就搞不明白了，我需要立即马上就要 power cycle 吗，还是说只要车经历一次 power cycle 之后，就会自动把用户的这一个性配置同步到用户数据（IPC AIDL 什么的吗?）看到这里，终于是有点儿迷糊了。。。。。

<p>Requirements:</p> <p><b>(CCC)</b></p> <p><i>Note : always review the JAMA system, this does not replace that.</i></p>	<p><b>GIVEN</b> that the vehicle is configured to the <b>EU</b> region ("IRegion=REU")</p> <p><b>AND</b> the user is looking at the Vehicle settings page</p> <p><b>WHEN</b> the user taps on the Drive settings</p> <p><b>THEN</b> the an On/Off toggle setting named "<b>Adaptive Driving Beam</b>" should be available to the user</p> <p><b>AND</b> an information icon should be displayed next to the setting</p> <p><b>AND</b> the setting should be defaulted to On - out of factory</p> <p><b>GIVEN</b> that the vehicle is configured to the <b>EU</b> region</p> <p><b>AND</b> the user is looking at the Drive settings</p> <p><b>When</b> the user makes either an On or Off selection on the "<b>Adaptive Driving Beam</b>" setting</p> <p><b>THEN</b> the setting has to be saved over power cycles to the current loaded profile</p>	<p><b>GIVEN</b> that the vehicle is configured to the <b>EU</b> region</p> <p><b>AND</b> the user is looking at the Drive settings</p> <p><b>WHEN</b> the user taps on the information icon next to the "<b>Adaptive Driving Beam</b>" setting</p> <p><b>THEN</b> the a information modal should pop-up explaining the feature</p> <p><b>AND</b> the modal should have a dismiss icon</p> <p><b>GIVEN</b> that the vehicle is configured to the <b>EU</b> region</p> <p><b>AND</b> the user is looking at the Adaptive Driving Beam information modal</p> <p><b>WHEN</b> the user taps on the Dismiss button</p> <p><b>THEN</b> the a modal should be dismissed</p> <p>Figma - <a href="https://www.figma.com/file/Hsda2inAtWMU4k0GnFgXwu/Vehicle-Settings?node-id=14%3A85212">https://www.figma.com/file/Hsda2inAtWMU4k0GnFgXwu/Vehicle-Settings?node-id=14%3A85212</a> - Connect to preview</p> <p>W</p>
<p>Requirements:</p> <p><b>(ICC)</b></p> <p><i>Note : always review the JAMA system, this does not replace that.</i></p>	<p><b>GIVEN</b> that the vehicle is configured to the <b>EU</b> region ("IRegion=REU")</p> <p><b>AND</b> the external lights are set to Auto</p> <p><b>AND</b> the Adaptive Driving Beam fault is active ("IBCM_HBAFitSts=1")</p> <p><b>WHEN</b> the user tries to activate ADB ("IBCM_AuthiBeamSetSts = 1")</p> <p><b>THEN</b> a Level-2 visual warning "<b>Adaptive Driving Beam Unavailable</b>"</p> <p><b>AND</b> a tell-tale notifying the failure needs to be shown on the ICC.</p> <p><b>AND</b> the manual High Beams should be active</p> <p>Figma - <a href="#">SFTUXUI-1790: UI for ADB failure warning and tell tale.</a></p> <p>COMPLETED</p>	

## 1.2 linking 相关逻辑

- 当 IRegion = "EU" 时 CCC 上默认是启用，介理 BCM 上我并不知道，所以车启动，我应该是（我可以向他们再确认一下吗？）需要向下发信号，要求找开 ADB 的？我并不总是一开车就需要找开远光灯的，所以更多的应该是在 CCC 层面，以及用户配置中，这一区域的用户默认配置启用，而并非总是一开始就是向下发信号要求打开远光灯。。。。。
- lucid-carctrl-lib 里加入一个监听回调接口，用来获取用户或是车所在的 Region
- 在 AsyncResource 里的 mUnitPrefListener 里监听区域，如果 mRegion.value = "EU"，启用 ADB

## 1.3 首先找到 LucidCarSettings app 中的最相关的界面 driving/DriveSettingsFragment.java

```
package com.lucid.car.carsettings.driving;
```

```
public class DrivingListFragment extends RightPanelFrag implements View.OnClickListener {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        super.onCreateView(inflater, container, savedInstanceState);
        View view = inflater.inflate(R.layout.drive_list_frag, container, false);
        initView(view);
        return view;
    }
    private void initView(View view) {
        // view.findViewById(R.id.list_item_1).setOnClickListener(this);
    }
}
```

```

        // view.findViewById(R.id.list_item_2).setOnClickListener(this);
        view.findViewById(R.id.list_item_3).setOnClickListener(this);
    }
    @Override
    public void onClick(View view) {
        int id = view.getId();
        // if (id == R.id.list_item_1) {
        //     swapRightPanel(TripInfoFragment.newInstance(), Utils.SettingsType.VEHICLE);
        // } else if (id == R.id.list_item_2) {
        //     swapRightPanel(TirePressureFragment.newInstance(), Utils.SettingsType.VEHICLE);
        // } else
        if (id == R.id.list_item_3) { // 注意这里需要获取资源的地方, Context 相关的
            swapRightPanel(new DriveSettingsFragment(getAsyncResources(), Utils.SettingsType.VEHICLE);
        }
    }
}

```

- 注意这里 getAsyncResources() 是调用的 RightPanelFrag 的方法

```

protected AsyncResources getAsyncResources() {
    Activity activity = getActivity();
    if (activity instanceof HomeActivity)
        return ((HomeActivity) activity).getAsyncResources(); // 从 activity 再调用 getAsyncResources()
    return null;
}

```

## 1.4 DriveSettingsViewContract.java

```

public interface DriveSettingsViewContract {
    void initUi();
    void updateStoppingMode(boolean isOn);
    void updateRegenBrakingLevel(int level);
    void updateHighBeamSensitivity(boolean isOn);
    void updateHighBeamPower(boolean isOn);
    void initGear();
    void updateGearStatus(boolean isInPark);
    void initDriveMode();
    void updateDriveModeStatus(DriveMode driveMode);
    void updateLscStatus(LSCStatus lscStatus);
    void updateCreepStatus(boolean isOn);
    // 这里是不是要增加一个自己的回调接口?
}

```

## 1.5 DriveSettingsFragment.java 的 layout 配置: R.layout.drive\_settings\_frag;

- 好像还没有我需要的那个条款, 可能我抓出来的版本不对, 要去找 EU 版本的, 必要的情况下要自己加上

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- < DRIVING SETTINGS -->
    <com.lucid.resource.layout.BackButton
        android:id="@+id/layout_drive_settings_back_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="37px"
        app:title="@string/option_drive_item_2" />

    <!-- Regenerative Braking -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@style/text_title_06.default"
        android:text="@string/drive_settings_breaking"
        android:layout_marginTop="64px"
        android:layout_marginLeft="48px" />
    <com.lucid.resource.switches.Large2Toggle
        android:id="@+id/breaking_2_toggle"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:textLeft="@string/drive_settings_breaking_option_1"
        app:textRight="@string/drive_settings_breaking_option_2"
        android:layout_marginTop="24px"
        android:layout_marginLeft="48px" />
<!-- Stop Mode: 只有到止为止两个 大 的 Toggle -->
<TextView
    android:id="@+id/tv_stop_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/text_title_06.default"
    android:text="@string/drive_settings_stop"
    android:layout_marginTop="48px"
    android:layout_marginLeft="48px" />
<com.lucid.resource.switches.Large2Toggle
    android:id="@+id/stop_mode_2_toggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:textLeft="@string/drive_settings_stop_mode_1"
    app:textRight="@string/drive_settings_stop_mode_2"
    android:layout_marginTop="24px"
    android:layout_marginLeft="48px" />

<!-- Headlight Beam Direction: 感觉这里好像不对????????? -->
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40px">
    <View
        android:id="@+id/hba_hide"
        android:layout_width="315px"
        android:layout_height="80px"
        android:layout_marginLeft="48px" />
    <!-- 它说是要显示在这个以前显示 High Beam Assist 的同一个地方: app:featureName = "High Beam Assist" -->
    <!-- 那么, 不是把这个文本换一下 UI 的部分就算基本完成了吗? 剩下的就是信号传递回调连接 -->
    <com.lucid.resource.layout.FeatureToggleList
        android:id="@+id/high_beam_list"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_marginLeft="48px"
        <!-- app:featureName="@string/drive_settings_high_beam_assist" -->
        app:featureName="@string/drive_settings_adaptive_driving_beam"
        app:showInfoIcon="true" />
    <!-- string.xml 中加个字符串: 注意确保只加在 EU 版本中 -->
    <string name="drive_settings_adaptive_driving_beam">Adaptive Driving Beam</string>
    </RelativeLayout>
    <!-- 这部分我暂时把它们放在一起, 只是提醒不要制造附加问题 -->
    <!-- app:featureName = "Reduced High Beam Sensitivity" 问题来了: 这里是否与上面相关, 有没有必要的逻辑是需要跟这部分也联接好的 -->
    <com.lucid.resource.layout.FeatureToggleList
        android:id="@+id/high_beam_sensitivity_list"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_marginTop="40px"
        android:layout_marginLeft="48px"
        app:featureName="@string/drive_settings_high_beam_assist_sensitivity"
        android:visibility="gone" />
    <!-- 这里说它暂时不显示, 查一下什么地方可以从代码里控制它们显示与否 -->

    <!-- Creep -->
    <com.lucid.resource.layout.FeatureToggleList
        android:id="@+id/creep_list"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_marginTop="40px"
        android:layout_marginLeft="48px"
        app:showInfoIcon="true"
        app:description="@string/drive_settings_creep_info"
        app:featureName="@string/drive_settings_creep" />

    <!-- app:featureName = "Low-Traction Start" -->
    <com.lucid.resource.layout.FeatureToggleList
        android:id="@+id/wheelspin_list"

```

```

android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:layout_marginTop="636px"
android:layout_marginLeft="48px"
app:showDisabled="true"
android:visibility="gone"
app:description="@string/drive_settings_wheelspin_info"
app:featureName="@string/drive_settings_wheelspin" />
<!-- _info: Makes starting on loose or slippery surfaces easier -->

<!-- <!--\- app:featureName = "Reduced High Beam Sensitivity" 原本的位置：问题来了：这里是否与上面相关，有没有必要的逻辑是需
<!-- <com.lucid.resource.layout.FeatureToggleList -->
<!-- android:id="@+id/high_beam_sensitivity_list" -->
<!-- android:layout_height="wrap_content" -->
<!-- android:layout_width="wrap_content" -->
<!-- android:layout_marginTop="40px" -->
<!-- android:layout_marginLeft="48px" -->
<!-- app:featureName="@string/drive_settings_high_beam_assist_sensitivity" -->
<!-- android:visibility="gone" /> -->

<!-- Lucid Stability Control: 大的，分三截，左中右 -->
<TextView
    android:id="@+id/tv_lucid_stability_control"
    android:layout_width="wrap_content"
    android:layout_height="80px"
    android:gravity="center"
    style="@style/text_title_06.default"
    android:layout_marginTop="40px"
    android:text="@string/drive_settings_stability_control"
    android:layout_marginLeft="48px"/>
<com.lucid.resource.switches.Large3Toggle
    android:id="@+id/lsc_stability_mode_toggle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="48px"
    android:layout_marginTop="8px"
    app:textLeft="@string/drive_settings_stability_control_option_3"
    app:textMiddle="@string/drive_settings_stability_control_option_1"
    app:textRight="@string/drive_settings_stability_control_option_2" />
<!-- Recommended for most drivinbg conditions -->
<TextView
    android:id="@+id/tv_lsc_msg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@id/lsc_stability_mode_toggle"
    android:layout_alignEnd="@id/lsc_stability_mode_toggle"
    android:layout_marginTop="8px"
    android:layout_marginLeft="48px"
    android:text="@string/drive_setting_stability_msg"
    android:textAppearance="@style/text_title_07.dim" />
</LinearLayout>

```

- 这里最好是能够截几个图，方便自己想看的时候一个地方查看，免得要翻来翻去的。。。。。

## 1.6 DriveSettingsFragment.java 中的主要逻辑：那么与会与 UI 层的 View Contract 相关？这里的源码还没有整理/没划重点

```

public class DriveSettingsFragment extends RightPanelFrag
// Fragment 实现了接口，那么 fragment instance 就可以视作监听了 (Fragment 也就需要实现这个接口类中所定义的所有接口)
implements DriveSettingsViewContract, // 实现它的所有接口，就可以当回调来配置 UI
LscWarningModal.LscModalCallbacks {
    private final static String TAG = DriveSettingsFragment.class.getName();

    private AppFeature mAppFeature; // 找找这个有什么相关的 ?????
    private final AsyncResources asyncResources;

    private HbaSensitivityModal hbaSensitivityModal; // 大概是一个确认打开高光灯之类的会话框
    private boolean mIsHighBeamAssistSensitivityOn = false;
    private boolean mIsHighBeamEnabled = false; // local variable

    private final Handler handler = new Handler();
    private static final long HBA_HIDDEN_VIEW_HOLD_TIME = 30000; // 30 seconds

```



```

private final Runnable mLongPressed = () -> showHighBeamSensitivity(true);

public DriveSettingsFragment(AsyncResources asyncResources) {
    this.asyncResources = asyncResources;
}

@Override public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mAppFeature = new AppFeature(this.getContext()); // 需要知道这个 AppFeature 会涉及到一些区域相关的定制吗????????
    mIsHighBeamEnabled = mAppFeature.isFeatureEnabled(DynamicFeature.AUTO_HIGH_BEAMS); // 远光灯打开了吗?
}

private void initView(View view) { // 初始化
    highBeamList = view.findViewById(R.id.high_beam_list);
    highBeamSensitivityList = view.findViewById(R.id.high_beam_sensitivity_list);
    View hbaHiddenView = view.findViewById(R.id.hba_hide); // 标在上面的隐藏着????
    // IVIANDAPPS-947
    // if (mIsHighBeamEnabled) { // 这里我打算把它改掉
    // 设置 UI event 的监听: 向下传递, 要求 BCM 同步
    highBeamList.setCheckedChangedCallback((buttonView, isChecked) -> highBeamIsToggled(isChecked));
    highBeamList.setInfoIconCallback( // 小圆圈叹号 的 UI Event 监听: 应该是再打开一个确认什么之类的小会话框
        () -> new HbaInfoModal().show(getFragmentManager(), "HbaInfoModal"));
    highBeamSensitivityList.setCheckedChangedCallback( // 这个视图的长按: 会显示某个什么东西, 只在按的过程中显示
        ((buttonView, isChecked) -> highBeamSensitivityIsToggled(isChecked)));
    hbaHiddenView.setOnTouchListener(new View.OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            switch (event.getAction()) {
                case MotionEvent.ACTION_DOWN:
                    handler.postDelayed(mLongPressed, HBA_HIDDEN_VIEW_HOLD_TIME);
                    break;
                case MotionEvent.ACTION_UP:
                    handler.removeCallbacks(mLongPressed);
                    break;
            }
            return true;
        }
    });
    // } else { // 这里我打算把它改掉
    //     highBeamList.setVisibility(View.GONE);
    //     highBeamSensitivityList.setVisibility(View.GONE);
    //     hbaHiddenView.setVisibility(View.GONE);
    // }
}

private final Runnable mLongPressed = () -> showHighBeamSensitivity(true);
private void showHighBeamSensitivity(boolean comesFromLongTouch) {
    if (mIsHighBeamEnabled) {
        if (allowToShowHbaSensitivity(comesFromLongTouch)) {
            highBeamSensitivityList.setVisibility(View.VISIBLE);
            boolean isOn = SystemPrefs.getHbaSensitivity(context) || mIsHighBeamAssistSensitivityOn;
            Log.d(TAG, "updateHighBeamSensitivityToggle() is called with status isOn=" + isOn);
            highBeamSensitivityList.setChecked(isOn);
        } else {
            highBeamSensitivityList.setVisibility(View.GONE);
        }
    }
}

private boolean allowToShowHbaSensitivity(boolean comesFromLongTouch) {
    if (mIsHighBeamEnabled) {
        boolean isInPark = asyncResources.isGearInPark();
        boolean isGuestProfile = ProfileAccountType.isGuestProfile(context);
        // if it comes from long touch, show regardless of ON status IF gear is park, 这里的逻辑有点儿说不清, 暂时不管
        // otherwise, only show if this is ON regardless of gear
        boolean allowBasedOnGear = (comesFromLongTouch && isInPark) || SystemPrefs.getHbaSensitivity(context); // 你需要
        boolean allowToShowHbaSen = highBeamList.isChecked() && !isGuestProfile && allowBasedOnGear;
        Log.d(TAG, "allow to show hba=" + allowToShowHbaSen);
        return allowToShowHbaSen;
    } else return false;
}

// 下面的两个方法来自于: prefs.SystemPref
// high beam assist sensitivity true means this is ON, false means this is off
public static boolean getHbaSensitivity(Context context) {
    SharedPreferences sharedPreferences = context.getSharedPreferences(HomeActivity.PREF_SETTINGS_APP, Context.MODE_PRIVATE);
    return sharedPreferences.getBoolean(PREF_HBA_SENSITIVITY, false);
}

// high beam assist sensitivity true means set this to ON, false means set this to off
public static void setHbaSensitivity(Context context, boolean isOn) {

```

```

Log.d(TAG, "setting HBA Sensitivity isOn=" + isOn);
SharedPreferences sharedPreferences = context.getSharedPreferences(HomeActivity.PREF_SETTINGS_APP, Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPreferences.edit();
editor.putBoolean(PREF_HBA_SENSITIVITY, isOn);
editor.apply();
}

private void highBeamIsToggled(boolean isChecked) { // CCC ==>>>> BCM
    if (mIsHighBeamEnabled) { // 如果 远光灯是打开 的前提下
        saveHighBeamPrefs(isChecked); // 这里保存会不会太早？想一下，如果 BCM 同步失败，应该也会再写一次用户数据配置
        Log.d(TAG, "updateHighBeamAssistance with status of on=" + isChecked); // 这里的 log 看得不清楚.....
        // 向下传，要求 BCM 同步配置硬件 on/off，调用控制器公共方法并启用计时器
        asyncResources.getBcmController().enableHighBeamAssistance(isChecked);
        showHighBeamSensitivity(false); // ???
    }
}

private void highBeamSensitivityIsToggled(boolean isChecked) {
    if (mIsHighBeamEnabled) { // 如果远光灯是打开的
        if (isChecked) {
            hbaSensitivityModal = new HbaSensitivityModal(asyncResources);
            hbaSensitivityModal.show(getFragmentManager(), "HbaSensitivityModal"); // 显示这个窗口
        } else {
            SystemPrefs.setHbaSensitivity(context, false); // 写入用户数据：不是数据库不是 ProfileData，是 SystemPref
            // 向下：要求硬件配置同步
            asyncResources.getBcmController().updateHighBeamSensitivity(BcmController.HIGH_BEAM_SENSITIVITY_NORMAL);
        }
    }
}

private void saveHighBeamPrefs(boolean isHighBeamAssistOn) { // BCM ==>>>> CCC ==>>>> ProfileData 保存用户的系统个性化配置
    Log.d(TAG, "PREF - saving to profile for " + ProfileKey.SETTINGS_HIGH_BEAM_ASSIST_PREF + "=" + isHighBeamAssistOn);
    // 这里应该又涉及 IPC AIDL，好像是，需要确认一下
    asyncResources.getProfileData().persistBoolean(ProfileKey.SETTINGS_HIGH_BEAM_ASSIST_PREF, isHighBeamAssistOn);
}

@Override public void onResume() {
    super.onResume();
    //this most likely be ready by the time user gets to this screen since asyncResources
    // is initialized when this app is first opened. If race condition happens, may want to show
    // a spinner or something so UI doesn't look clunky. If this screen is opened before
    // asyncResources is ready, initUi() gets called via the AdasViewContract.
    if (asyncResources.isReadyAll()) // 它说，等用户个性化配置的所有数据都 Ready 了，那么就启动 UI 吧
        initUi();
}

@Override public void initUi() {
    Log.d(TAG, "initUi was called");
    updateRegenBrakingLevel(asyncResources.getBcmController().getRegenBrakingLevel());
    // boolean isOnePedalDrivingOn = asyncResources.getBcmController().isOnePedalDrivingOn();
    // if (isOnePedalDrivingOn) {
    //     stop2Toggle.selectLeft();
    // } else {
    //     stop2Toggle.selectRight();
    // }
    if (mIsHighBeamEnabled) { // 它很特殊，它感光，它是一个 Runnable，大概不能简单地只用一个监听回调
        updateHighBeamToggle(asyncResources.getBcmController().isHighBeamPowerOn());
        // high beam sensitivity is updated within updateHighBeamToggle()
    }
    // updateCreepToggle(asyncResources.getBcmController().getCreepModeStatus() != 0);
    // initGear(); // 这两个不知道是在干什么，暂时不管
    // initDriveMode();
}

// @Override public void initGear() {
//     updateGearStatus(asyncResources.isGearInPark());
// }

@Override public void updateHighBeamSensitivity(boolean isOn) {
    if (mIsHighBeamEnabled) {
        mIsHighBeamAssistSensitivityOn = isOn;
        highBeamSensitivityList.setChecked(isOn);
        if (!isOn)
            showHighBeamSensitivity(false);
    }
}

@Override
public void updateHighBeamPower(boolean isOn) {
    if (mIsHighBeamEnabled)
        updateHighBeamToggle(isOn);
}

```

```

public void updateHighBeamToggle(boolean isOn) {
    if (mIsHighBeamEnabled) {
        Log.d(TAG, "updateHighBeamToggle() is called with status isOn=" + isOn);
        highBeamList.setChecked(isOn);
        showHighBeamSensitivity(false);
    }
}

private void showHighBeamSensitivity(boolean comesFromLongTouch) {
    if (mIsHighBeamEnabled) {
        if (allowToShowHbaSensitivity(comesFromLongTouch)) {
            highBeamSensitivityList.setVisibility(View.VISIBLE);
            boolean isOn = SystemPrefs.getHbaSensitivity(context) || mIsHighBeamAssistSensitivityOn;
            Log.d(TAG, "updateHighBeamSensitivityToggle() is called with status isOn=" + isOn);
            highBeamSensitivityList.setChecked(isOn);
        } else highBeamSensitivityList.setVisibility(View.GONE);
    }
}

private boolean allowToShowHbaSensitivity(boolean comesFromLongTouch) {
    if (mIsHighBeamEnabled) {
        boolean isInPark = asyncResources.isGearInPark();
        boolean isGuestProfile = ProfileAccountType.isGuestProfile(context);
        //if it comes from long touch, show regardless of ON status IF gear is park
        //otherwise, only show if this is ON regardless of gear
        boolean allowBasedOnGear = (comesFromLongTouch && isInPark) || SystemPrefs.getHbaSensitivity(context);
        boolean allowToShowHbaSen = highBeamList.isChecked() && !isGuestProfile && allowBasedOnGear;
        Log.d(TAG, "allow to show hba=" + allowToShowHbaSen);
        return allowToShowHbaSen;
    } else return false;
}

@Override public void disable() {
    asyncResources.getBcmController().setLSCMode(LSCRequest.OFF.getValue());
    Log.d(TAG, "Disable event triggered");
}

@Override public void onDestroy() {
    super.onDestroy();
    handler.removeCallbacks(mLongPressed);
}

@Override public void onStart() {
    super.onStart();
    Activity activity = getActivity();
    if (activity instanceof HomeActivity)
        ((HomeActivity) activity).setDriveSettingsVc(this); // 设置 view 层的监听回调
}

@Override public void onStop() {
    super.onStop();
    Activity activity = getActivity();
    if (activity instanceof HomeActivity)
        ((HomeActivity) activity).clearDriveSettingVc(); // clean up
}
}

```

## 1.7 LucidCarSettings 应用中 HomeActivity 中与 Controller 控制器相关/UI 回调相关/bug 相关的最重要逻辑

- 这里，我应该是需要往更上层地去找，是谁在设置这些个 View 层面的监听，比如：HomeActivity 是在什么时候什么情况下设置了某个它的 UI Contract 给某个特定的 Fragment？这里理解起来有点儿绕，但是就是那么个意思

```

public class HomeActivity extends AppCompatActivity {
    private static final String TAG = HomeActivity.class.getName();
    // private static final String OWNER_MANUAL_URL = "OWNER_MANUAL_URL";

    // 在 DriveSettingsFragment.java 中设置这个，在 onStart() 方法中设置：
    // ((HomeActivity) activity).setDriveSettingsVc(this);
    public void setDriveSettingsVc(DriveSettingsFragment frag) {
        driveSettingVc = frag;
    }

    public void clearDriveSettingVc() {
        driveSettingVc = null;
    }
}

```

```

// private ProfileData mProfileData;
// private AsyncResources asyncResources;
private Handler uiHandler = new Handler(Looper.getMainLooper()); // handler 机制崩出来了，如何动作的呢？
// 它有一堆 UI 相关的 Contract，同时它也有一堆的 Model，它们是怎么联系起来的呢？
// 定义些 Fragment 中的监听回调，为的就是根据传递上来的信号，来改变 Fragment UI 显示
private AdasViewContract adasVc;
private DriveSettingsViewContract driveSettingVc;
private AccessibilityViewContract accessibilityVc;
private TripInfoViewContract tripInfoVc;
private DisplaysViewContract langVc;

// private HomeViewModel mHomeViewModel;

//=====
// Controller Listeners + Async Resources connection listener
//=====
private final ProfileData.ServiceConnectionListener profileListener =
    new ProfileData.ServiceConnectionListener() {
// 它定义了一个接口在这里，可是它并没有在回调中去处理什么逻辑，那么定义在这里是为什么呢？方便别人来连它 ??????
@Override public void onConnected() { }
@Override public void onDisconnected() { }
};
AsyncResources.ConnectionsListener connectionsListener =
    new AsyncResources.ConnectionsListener(
        profileListener,
        bcmListener,
        adasListener,
        null, // 这一个是什么呢？
        gearListener,
        driveModeListener
    ) {
@Override public void onConnectedAll() {
    Log.d(TAG, "onConnectedAll");
    adasListener.onPropertiesReadyW();
    bcmListener.onPropertiesReadyW();
    gearListener.onPropertiesReadyW();
    driveModeListener.onPropertiesReadyW();
}
@Override public void onDisconnectedAny() {}
};
// 留这些 controller 在这里，就是给你留几个葫芦，你可以照着画个瓢
private final DriveModeController.DriveModeListener driveModeListener
    = new DriveModeController.DriveModeListener() {
@Override
public void onPropertiesReadyW() {
    if (uiHandler != null) {
        uiHandler.post() -> {
            // this handles the race condition if the connection has not been made
            // yet before the user navigates to the DreamDrive tab. In this case, the UI will flash.
            // May want to consider using a spinner IF this happens too frequently.
            if (driveSettingVc != null)
                driveSettingVc.initUi();
        };
    }
}
@Override public void onUpdateDriveModeW(Integer state) {
    if (uiHandler != null) {
        uiHandler.post() -> {
            if (driveSettingVc != null)
                driveSettingVc.updateDriveModeStatus(asyncResources.mapToDriveMode(state));
        };
    }
}
};
private final GearController.GearListener gearListener = new GearController.GearListener() {
public void onPropertiesReadyW() { }
public void onUpdateGearStateChangedW(Integer state) {}
};
private final BcmController.BcmListener bcmListener = new BcmController.BcmListener() {
public void onPropertiesReadyW() {}
public void onUpdateOnePedalDrivingW(boolean isOn) {}
public void onUpdateRegenBrakingLevelW(int value) {}
@Override // 这个回调是作什么用的呢？
public void onUpdateHighBeamPowerW(boolean isOn) {}
public void onUpdateHighBeamSensitivityW(int val) {}
};

```

```

    public void onCreepModeStatusChangeW(int value) {}
    public void onUpdateRcccdLockPowerW(boolean isOn) {}
    public void onUpdateLSCModeStatusW(int status) {}
    public void onUpdateTripADistanceW(float value) {}
    public void onUpdateTripAEnergyW(float value) {}
    public void onUpdateTripAEfficiencyW(float value) {}
    public void onUpdateTripBDistanceW(float value) {}
    public void onUpdateTripBEnergyW(float value) {}
    public void onUpdateTripBEfficiencyW(float value) {}
    public void onUpdateLastChargeDistanceW(float value) {}
    public void onUpdateLastChargeEnergyW(float value) {}
    public void onUpdateLastChargeEfficiencyW(float value) {}
};

private final AdasController.AdasListener adasListener = new AdasController.AdasListener() {
    public void onPropertiesReadyW() {}
    public void onUpdateHighwayAssistW(boolean isHwaOn) {}
    public void onUpdateLdpPowerW(boolean isOn) {}
    public void onUpdateLdpLevelW(int level) {}
    public void onUpdateCollisionProtectionPowerW(boolean isOn) {}
    public void onUpdateCollisionProtectionLevelW(int level) {}
    public void onUpdateCrossTrafficProtectionW(boolean isOn) {}
    public void onUpdateDriveOffAlertPowerW(boolean isOn) {}
    public void onUpdateDrowsyDriverStatusW(boolean isOn) {}
    public void onHWAStatusChanged(boolean isActive) {}
    public void onUpdateRadarStatusW(AdasController.Radar radar, Integer value){}
};
}

```

## 1.8 UnitPrefController.java 相关的 LucidCarSettings 中的控制器

- 只是看了一下，摘了一点儿，未必重要，有不懂的，改天再去把它们看懂

```

// profileAndAccess.LoginService.java
AsyncResources.ConnectionsListener mConnectionsListener =
    new AsyncResources.ConnectionsListener(
        mProfileListener,
        mBcmControllerListener,
        mAdasControllerListener,
        mUnitPrefListener, //
        null,
        null) {
    @Override
    public void onConnectedAll() {
        Log.d(TAG, "onConnectedAll()");
        // Upon connecting, get user preference
        ProfileData profileData = AsyncResources.getInstance(getBaseContext()).getProfileData();
        initUnitPrefs(profileData); // 这里可能没看懂
        initBcmPrefs(profileData);
        initAdasPrefs(profileData);
        initAudioPrefs( profileData );
    }

    @Override
    public void onDisconnectedAny() {
        // Do nothing.
    }
};

// 在 CarSetting 中，把车的起始化配置，同步写入到了这些个用户相关的 Pref 用户数据库中，也?????
// 从车的配置，向用户数据中去写 ????????.....
// 可是车配置中的初始化，来自于哪里呢？改天再来看这些吧
// CarSettings.IS_BATTERY_PERCENTAGE_UNIT_DEFAULT 等是来自 lucid-settings-lib static 定义
private void initUnitPrefs(ProfileData profileData) {
    updatePref(profileData, CarSettings.IS_BATTERY_PERCENTAGE_UNIT, CarSettings.IS_BATTERY_PERCENTAGE_UNIT_DEFAULT);
    updatePref(profileData, CarSettings.IS_PRESSURE_PSI_UNIT, CarSettings.IS_PRESSURE_PSI_UNIT_DEFAULT);
    updatePref(profileData, CarSettings.IS_TIME_24_HR, CarSettings.IS_TIME_24_HR_DEFAULT);
    updatePref(profileData, CarSettings.IS_TEMP_F_UNIT, CarSettings.IS_TEMP_F_UNIT_DEFAULT);
    notifyIccForUnitPrefs(profileData);
}

private void updatePref(ProfileData profileData, String key, boolean defaultVal) {
    boolean val = profileData.retrieveBoolean(key, defaultVal);
    Settings.Global.putString(getApplicationContext().getContentResolver(), key, String.valueOf(val));
}

// AsyncResource UnitPrefController wrapper

```

```
private final UnitPrefController.UnitPrefListener mUnitPrefControllerListener =
    new UnitPrefController.UnitPrefListener() {
        @Override
        public void onVhalConnectedW() {
            super.onVhalConnectedW();
            synchronized (mConnectionsListeners) {
                for (ConnectionsListener listener : mConnectionsListeners) {
                    if (listener.mUnitPrefListener != null)
                        listener.mUnitPrefListener.onVhalConnectedW();
                }
            }
            mUnitPrefControllerReady = true;
            checkConnectedAll();
        }
        @Override
        public void onVhalDisconnectedW() {
            super.onVhalDisconnectedW();
            mUnitPrefControllerReady = false;
            checkDisconnectedAny();
            synchronized (mConnectionsListeners) {
                for (ConnectionsListener listener : mConnectionsListeners) {
                    if (listener.mUnitPrefListener != null)
                        listener.mUnitPrefListener.onVhalDisconnectedW();
                }
            }
        }
    };
```

1.9 先去把信号找到，确定好是能够传上来的！不要等到最后一分钟：应该是可以传得上来的

- 它的意思是说，作个替换，把 US 版本的 High Beam Assist 换成 EU 版本的 Adaptive Driving Beam(ADB), 不该是换汤不换药吗？
- 信号也是用现有的，但是可能细节的要求：左驾右驾，ICC 与 CID 的要求全都需要满足

ICCC\_AuthHiBeamSetReq  
IBCM\_AuthHiBeamSetSts  
IBCM\_AuthHiBeamSts

- 需要搞清楚这三个信号分别是什么意思，怎么起作用的，见下面的截图：

HeadLight		BCM_to_HMI/IBCM_HiBeamSts BCM_to_HMI/IBCM_MainBeamSts	Off:0, On:1
Auto high beam setting (HBA high beam assist)	ICCC_AuthHiBeamReq S2R-16: CCC_to_BCM_1/ICCC_AuthHiBeamSetReq	IBCM_AuthHiBeamSts S2R-16: <b>IBCM_AuthHiBeamSetSts</b>	<u>request</u> : Idle:0, disable:1, enable:2, reserved:3 <u>status</u> : Disabled:0, Enabled:1
High Beam Assist sensitivity setting (HBA)		IHCU_L_HLASnvty	Unknown:0, Normal:1, US:2, Invalid:3

1.10 BcmController.java 具体三个信号的接收与监听

- 在 LucidCarSettings 应用中的定位是在 AsyncResources.java 的 mBcmListener 中

```

public class BcmController {
    private static final int VCU_SIGNAL_TIMEOUT = 1 * 1000; // this timeout will be used for REGEN, STOPMODE, and CREEP MOD

    public static class BcmListener {
        public void onVhalConnectedW() { }
        public void onVhalDisconnectedW() { }
        /**
         * Callback for High Beam Power
         * @param isOn = true means ON, false means OFF
         */
        public void onUpdateHighBeamPowerW(boolean isOn) { }
        /**
         * Callback for High Beam Sensitivity Level
         * @param val = HIGH_BEAM_SENSITIVITY_NORMAL or HIGH_BEAM_SENSITIVITY_US
         */
        public void onUpdateHighBeamSensitivityW(int val) { }
    }
    /**
     * High Beam Request
     * @param enable :0: Idle - 0x00
     *                (1-0): Disable - 0x01
     *                (2-0): Enable - 0x02
     *                (3-0): Reserved
     */
    public void enableHighBeamAssistance(boolean enable) { // 这个公共方法，是在 AsyncResource.java 中的 mBcmListener 中调用
        // 当你要取消定时器的時候，只需要调用 handler.removeCallbacks(runnable) 就可以了
        mSignalBroadcastHandler.removeCallbacks(mHighBeamRunnable);
        int val = enable ? 2 : 1;
        mSignalBroadcastHandler.sendMessage(mSignalBroadcastHandler
            .obtainMessage(MSG_BROADCAST_SIGNAL, ICCU_AUT_HI_BEAM_SET_REQ, val));
        // 调用 handler.post(runnable); 就能启动定时器，750 ms 内完成不了就取消操作，维持原状
        mSignalBroadcastHandler.postDelayed(mHighBeamRunnable, 750);
        // 延迟 750 ms 执行：以便 BCM 运行不力的时候，取消息上次操作，转为 Idle，即上次 request 作废，继续维持上次 request 之前的状态
    }
    final Runnable mHighBeamRunnable = new Runnable() {
        // 启用计时器：当 BCM 反应不过来，或是出错了，不能在规定的最长时间內完成我们想要的设置
        // 我们有充足的理由取消操作，向用户反馈设置失败，更新同步 UI 为硬件状态值
        @Override public void run() { // 它没事的时候，就是常年地闲逛吗？发一条消息 idle，对应 BCM 的 matching；然后再向用户反馈
            // set to idle val
            mSignalBroadcastHandler.sendMessage(
                mSignalBroadcastHandler.obtainMessage(MSG_BROADCAST_SIGNAL, ICCU_AUT_HI_BEAM_SET_REQ, 0)); // 0 0 0 0 0 idl
            // update UI for current status signal
            if (mBcmListener != null) {
                synchronized (mPropertiesLock) {
                    mBcmListener.onUpdateHighBeamPowerW(isHighBeamPowerOn);
                }
            }
        }
    };
    /**
     * High Beam Sensitivity Request
     * @param sensitivity
     *     Unknown:0, Normal:1, US:2, Invalid:3
     */
    public static int HIGH_BEAM_SENSITIVITY_NORMAL = 1;
    public static int HIGH_BEAM_SENSITIVITY_US = 2;
    // 这个方法定义出来，谁在用呢？公共方法，在应用层调用
    public void updateHighBeamSensitivity(int sensitivity) {
        if (sensitivity != HIGH_BEAM_SENSITIVITY_NORMAL && sensitivity != HIGH_BEAM_SENSITIVITY_US) {
            Log.e(TAG, "DEVELOPER ERROR - INVALID REQUEST INT SENT FOR HIGH BEAM SENSITIVITY");
            return;
        }
        mSignalBroadcastHandler.removeCallbacks(mHighBeamSensitivityRunnable);
        mSignalBroadcastHandler.sendMessage(mSignalBroadcastHandler
            .obtainMessage(MSG_BROADCAST_SIGNAL, HIGH_BEAM_ASSIST_SENSITIVITY_SEL, sensitivity));
        mSignalBroadcastHandler.postDelayed(mHighBeamSensitivityRunnable, 750);
    }
    final Runnable mHighBeamSensitivityRunnable = new Runnable() {
        @Override public void run() {
            // update UI for current status signal
            if (mBcmListener != null) {
                synchronized (mPropertiesLock) {
                    mBcmListener.onUpdateHighBeamSensitivityW(highBeamSensitivitySts);
                }
            }
        }
    }
}

```

```

    }
};

//=====
// Init / Cleanup
//=====
private Context mContext;
private BcmListener mBcmListener;
public BcmController() {
    // Signal broadcasting worker thread.
    mHandlerThread = new HandlerThread( TAG );
    mHandlerThread.start();
    mSignalBroadcastHandler = new SignalBroadcastHandler( mHandlerThread.getLooper() );
}
public void init(Context context, BcmListener bcmListener) {
    mContext = context;
    mBcmListener = bcmListener;
    mCarApiClient = Car.createCar( mContext, mCarConnectionCallback );
    mCarApiClient.connect();
}
public void cleanup() {
    if ( mCarApiClient != null ) {
        mCarApiClient.disconnect();
    }
    mBcmListener = null;
    mContext = null;
}

//=====
// Connection to VHal
//=====
private Car mCarApiClient;
private CarPropertyManager mPropertyManager;
private final ServiceConnection mCarConnectionCallback = new ServiceConnection() {
    @SuppressWarnings("StaticFieldLeak")
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        if ( mBcmListener != null ) {
            if (IS_DEBUG_BUILD) Log.d( TAG, "mBcmListener.onVhalConnectedW()" );
            mBcmListener.onVhalConnectedW();
        }
        try {
            mPropertyManager = (CarPropertyManager) mCarApiClient
                .getCarManager( android.car.Car.PROPERTY_SERVICE );
            for ( int propId : S_NOTIFY_PROP_IDS )
                mPropertyManager.registerCallback(mPropertiesListener, propId, 100 );
            // Fetch initial values.
            new AsyncTask<Void, Void, Void>() {
                @Override
                protected Void doInBackground(Void... nulls) {
                    fetchProperties();
                    return null;
                }
            }.execute();
            @Override
            protected void onPostExecute(Void aVoid) {
                // Notify
                if ( mBcmListener != null ) {
                    if (IS_DEBUG_BUILD) Log.d( TAG, "mBcmListener.onPropertiesReadyW()" );
                    mBcmListener.onPropertiesReadyW();
                }
            }
        } catch ( CarNotConnectedException e ) {
            Log.e(TAG, "onConnected() : Car not connected.");
        } catch ( IllegalStateException e ) {
            Log.e(TAG, "Illegal State Exception... " + e.getMessage());
        }
    }
}
@Override
public void onServiceDisconnected(ComponentName name) {
    for( int propId : S_NOTIFY_PROP_IDS ) {
        mPropertyManager.unregisterCallback(mPropertiesListener, propId );
    }
    if ( mBcmListener != null ) {
        if (IS_DEBUG_BUILD) Log.d( TAG, "mBcmListener.onVhalDisconnectedW()" );
    }
}

```



```

        mBcmListener.onVhalDisconnectedW();
    }
}
};

private void fetchProperties() { // 初始化监听的两个属性本地变量值
    Log.d( TAG, "fetchProperties() : BEGIN" );
    int value;
    float fValue;
    synchronized (mPropertiesLock) {
        try {
            value = mPropertyManager.getIntProperty(IBCMAUT_HI_BEAM_SET_STS, 0);
            isHighBeamPowerOn = value == 1;
            value = mPropertyManager.getIntProperty(HCU_HLAS_SENSITIVITY_STS, 0);
            highBeamSensitivitySts = value;
        } catch ( android.car.CarNotConnectedException e ) {
            Log.e(TAG, "onConnected() : Car not connected.");
        }
    }
    Log.d( TAG, "fetchProperties() : END" );
}

//=====
// Incoming BCM Signals Handling
//=====
private CarPropertyManager.CarPropertyEventCallback mPropertiesListener =
    new CarPropertyManager.CarPropertyEventCallback() {
        @Override
        public void onErrorEvent(int i, int i1) {
            Log.e( TAG, "onErrorEvent() : i=" + i + ", i1=" + i1 );
        }
        @Override
        public void onChangeEvent(CarPropertyValue carPropertyValue) { // 更新监听的两个属性本地变量的值
            if (IS_DEBUG_BUILD) Log.d(TAG, " onChangeEvent " + carPropertyValue.getPropertyId()
                + " (0x" + Integer.toHexString(carPropertyValue.getPropertyId()) + ")");
            switch ( carPropertyValue.getPropertyId() ) {
                case IBCMAUT_HI_BEAM_SET_STS: // BCM_to_HMI/IBCM_HiBeamSts: on/off
                    handleHighBeamPowerW((Integer) carPropertyValue.getValue());
                    break;
                case HCU_HLAS_SENSITIVITY_STS:
                    handleHighBeamSensitivityPowerW((Integer) carPropertyValue.getValue());
                    break;
            }
        }
    };

// boolean parseCarPropertyValue(CarPropertyValue carPropertyValue) {
//     return (Integer) carPropertyValue.getValue() == 1 ? true : false;
// }
void handleHighBeamPowerW(Integer value) { // 只有有效值的时候，才会收到回调，idle 与 matching 不会收到回调
    if (value != 0 && value != 1) {
        Log.e(TAG, "VHAL ERROR - INVALID VAL IS SENT FOR HIGH BEAM POWER");
        return;
    }
    synchronized (mPropertiesLock) {
        isHighBeamPowerOn = value == 1;
    }
    if (mBcmListener != null) { // 不管是 ON，还是 OFF，都会收到回调
        boolean isOn = value == 1;
        mBcmListener.onUpdateHighBeamPowerW(isOn);
    }
}

void handleHighBeamSensitivityPowerW(Integer value) {
    synchronized (mPropertiesLock) {
        highBeamSensitivitySts = value;
    }
    if (mBcmListener != null)
        mBcmListener.onUpdateHighBeamSensitivityW(value);
}

//=====
// Car Property IDs
//=====
// 好像是可以找到要求的三个中的两个信号
private static final int ICCCAUT_HI_BEAM_SET_REQ = 0x2140114A; // high beam request signal - ARXML ICCCAuthiBeamSetRe
private static final int IBCMAUT_HI_BEAM_SET_STS = 0x2140114B; // high beam status signal - ARXML IBCMAuthiBeamSetSt
private static final int HIGH_BEAM_ASSIST_SENSITIVITY_SEL = 0x21401149; // high beam sensitivity request signal - ARXML

```

```

//STATUS Signal to ICC
private static final int TRIP_INF_DISP_SETTING_REQ = 0x2140170E;
private static final int IVCU_LSC_MODE_STAT = 0x21401610; //FULL = 0, SEMI = 1, OFF = 2
private static final int ICCU_LSC_MODE_REQ = 0x21401611; //Idle:0, Mode0:1, Mode1:2, Mode2:3.
public static final int ICCU_CRP_MOD_REQ = 0x21401620;
public static final int IVCU_CRP_MOD_ST = 0x21401621;
private final static int[] S_NOTIFY_PROP_IDS = new int [] {
    IBCM_AUT_HI_BEAM_SET_STS,
};

private CidState mCidState;
private int mGloveBoxStatus;
private int mSteerColTele = 0;
private int mSteerColTilt = 0;
private float mTargetSOCStatus;
private float realTimePowerInfoStatus;
private int onePedalDrivingStatus; // 它依赖一只脚吗???????
// private int regenBreakingLevel = REGEN_BRAKING_LVL_MED;
private boolean isHighBeamPowerOn; // 打开远光灯了吗? 初始化的时候 fetch 初始值; 然后收到回调的时候变更
private int highBeamSensitivitySts;
private int lscModeStatus;
private int screenCleaningMode;
public static final int HIGH_BEAM_SENS_STS_ON = 2;
// private static boolean isRcccdLockOn;
private final Object mPropertiesLock = new Object();

//=====
// Handling Broadcasting of Signals (from CCC): 这个部分比较熟悉, 可以暂时不用看了
//=====
private static final int MSG_BROADCAST_SIGNAL = 100;
private HandlerThread mHandlerThread;
private SignalBroadcastHandler mSignalBroadcastHandler;
class SignalBroadcastHandler extends Handler {
    SignalBroadcastHandler(Looper looper) {
        super(looper);
    }
    @Override
    public void handleMessage(Message msg) {
        if ( msg.what == MSG_BROADCAST_SIGNAL ) {
            broadcastCccSignalW( msg.arg1, msg.arg2 );
        }
    }
    void broadcastCccSignalW(int propId, int value) {
        try {
            mPropertyManager.setIntProperty( propId, 0, value );
        } catch ( android.car.CarNotConnectedException e ) {
            Log.e( TAG, "broadcastCccSignalW() : VHAL not connected." );
        }
    }
}
}
}

```

## 1.11 AsyncResources。java 的 mBcmListener

## 1.12 IMarket IRegion: lucid-carctrl-lib UnitPrefController.java

- 我找不到：上次 UnitsFragment 都没有找到，今天早上赶快找，把它们找不出来，找不出来就生成不了 EU 版本特别是的这个 Featurture
- 所有重要的相关逻辑都在这里了。如果需要添加监听，应该还是不是在这里，还需要再去找，这里只是硬件软件配制初始化的源头?????.....
- 注意为了只在 EU 中，为了能在我的 Pixel3 上简单测试，我可以直接改一下 lucid-settings-lib 就可以了；可是 IRegion 不在这，我改了没有用.....

```

public class UnitPrefController {
    public static class UnitPrefListener {
        public void onVhalConnectedW() {}
        public void onVhalDisconnectedW() {}
    }
}

```

```
// public void onStart() { // 我不需要这个回调，定义自己的 IMarket IRegion 回调 }  
// hack: temp hack IRegion in LucidCarSettings app even in US to be hard coded to be  
// mRegion.value = "EU", so that I can test locally in Pixel3 and bench in US  
// They could help build/configure the bench to simulate EU too, as team suggested  
public Region onRegionReady() { }  
}  
//=====br/>// Init / Cleanup  
//=====br/>private Context mContext;  
private UnitPrefController.UnitPrefListener mUnitPrefListener;  
private final static float SIGNAL_REFRESH_RATE = 0;  
private static final int VEHICLE_GLOBAL_AREA = 0;  
  
private Market mMarket = Market.UNKNOWN;  
private Region mRegion = Region.UNKNOWN;  
private boolean mIsDebuggable = false;  
private final static String MARKET_CANADA = "CAN";  
private final static String MARKET_SAUDI_ARABIA = "KSA";  
private final static String MARKET_CHINA = "CHI";  
private final static String MARKET_US = "USA";  
private final static String MARKET_UK = "GBR";  
private final static String REGION_NORTH_AMERICA = "RNA";  
private final static String REGION_EUROPE = "REU"; // <<<<<<<<<<<=====br/>private final static String REGION_ASIA = "RAP";  
private final static String REGION_MIDDLE_EAST = "RME";  
public static final String IS_DISTANCE_IMPERIAL_MI_UNIT = "IS_DISTANCE_IMPERIAL_MI_UNIT";  
  
public UnitPrefController() {  
    // Signal broadcasting worker thread.  
    mHandlerThread = new HandlerThread(TAG);  
    mHandlerThread.start();  
    mSignalBroadcastHandler = new UnitPrefController.SignalBroadcastHandler(mHandlerThread.getLooper());  
}  
public void init(Context context, UnitPrefController.UnitPrefListener unitPrefListener) { // 这个方法总是会被调用，因为要  
    mContext = context;  
    TAG = mContext.getPackageName() + " | " + TAG;  
    mUnitPrefListener = unitPrefListener; //  
    mCarApiClient = Car.createCar(mContext, mCarConnectionCallback); // 准备一个车  
    mCarApiClient.connect(); // 连上这辆车  
    mIsDebuggable = isDebuggable(); //  
}  
public void cleanup() {  
    if (mCarApiClient != null)  
        mCarApiClient.disconnect(); // 断开与车的连接：因为是为了在一个更高的层面上 clean up，免得 Application 不能回收，浪费资源  
    mUnitPrefListener = null;  
    mContext = null;  
}  
//=====br/>// Connection to VHal  
//=====br/>private Car mCarApiClient;  
private CarPropertyManager mPropertyManager;  
private final ServiceConnection mCarConnectionCallback = new ServiceConnection() {  
    @Override public void onServiceConnected(ComponentName name, IBinder service) {  
        try {  
            Log.d(TAG, "onConnected..");  
            mPropertyManager = (CarPropertyManager) mCarApiClient  
.getCarManager(android.car.Car.PROPERTY_SERVICE);  
            loadDefaultValues(); // 这里应试是车的初始化的起始配置：这就涉及到了车/用户所在的版块分区语言与车的硬件 FW 配置等  
        } catch (android.car.CarNotConnectedException e) {  
            Log.e(TAG, "onConnected() : Car not connected.");  
        }  
        if (mUnitPrefListener != null) {  
            Log.d(TAG, "mUnitPrefListener.onVhalConnectedW()");  
            mUnitPrefListener.onVhalConnectedW();  
        }  
    }  
};  
@Override public void onServiceDisconnected(ComponentName name) {  
    if (mUnitPrefListener != null) {  
        Log.d(TAG, "mUnitPrefListener.onVhalDisconnectedW()");  
        mUnitPrefListener.onVhalDisconnectedW();  
    }  
}
```

```

};
private CarPropertyManager.CarPropertyEventCallback mPropertiesListener =
    new CarPropertyManager.CarPropertyEventCallback() {
        @Override public void onErrorEvent(int i, int il) {
            Log.e(TAG, "onErrorEvent() : i=" + i + ", il=" + il);
        }
        @Override public void onChangeEvent(CarPropertyValue carPropertyValue) {
            int propId = carPropertyValue.getPropertyId();
            Log.d(TAG, "onChangeEvent..." + propId);
            switch (propId) {
                case BCM_TO_CCC_IMARKET:
                case BCM_TO_CCC_IREGION:
                    String value = parseStringSafely(carPropertyValue);
                    if (BCM_TO_CCC_IMARKET == propId) {
                        log("Received Market changed, value:" + value);
                        if (!TextUtils.isEmpty(value)) {
                            Market tempMarket = Market.valueOfMarket(value);
                            Log.d(TAG, " tempMarket: " + tempMarket + " mMarket: " + mMarket);
                            if (tempMarket != mMarket)
                                mMarket = tempMarket;
                        }
                    } else { // BCM_TO_CCC_IREGION == propId
                        log("Received Region changed, value:" + value);
                        if (!TextUtils.isEmpty(value)) {
                            Region tempRegion = Region.valueOfRegion(value);
                            Log.d(TAG, " tempRegion: " + tempRegion + " mRegion: " + mRegion);
                            if (tempRegion != mRegion)
                                mRegion = tempRegion;
                        }
                    }
                    Log.d(TAG, " mUnitPrefListener: " + mUnitPrefListener + " mMarket: " + mMarket + " mRegion: " + mRegion);
                    if (mUnitPrefListener != null && (mMarket != Market.UNKNOWN || mRegion != Region.UNKNOWN))
                        mUnitPrefListener.onDistanceUnitReady(isDistanceUnitInMiles());
                    break;
            }
        }
    };
private void loadDefaultValues() throws CarNotConnectedException { //
    CarPropertyValue<String> marketValue = mPropertyManager.getProperty(BCM_TO_CCC_IMARKET, VEHICLE_GLOBAL_AREA); // IM
    String strMarket = parseStringSafely(marketValue);
    log("Get Market, value:" + strMarket);
    if (!TextUtils.isEmpty(strMarket))
        mMarket = Market.valueOfMarket(strMarket);
    CarPropertyValue<String> regionValue = mPropertyManager.getProperty(BCM_TO_CCC_IREGION, VEHICLE_GLOBAL_AREA); // IR
    String strRegion = parseStringSafely(regionValue);
    log("Get Region, value:" + strRegion);
    if (!TextUtils.isEmpty(strRegion))
        mRegion = Region.valueOfRegion(strRegion);
    if (mUnitPrefListener != null && (mMarket != Market.UNKNOWN || mRegion != Region.UNKNOWN))
        mUnitPrefListener.onDistanceUnitReady(isDistanceUnitInMiles());
    mPropertyManager.registerCallback(mPropertiesListener, BCM_TO_CCC_IMARKET, SIGNAL_REFRESH_RATE); // 注册回调
    mPropertyManager.registerCallback(mPropertiesListener, BCM_TO_CCC_IREGION, SIGNAL_REFRESH_RATE);
}

//=====
// Handling Broadcasting of Signals (from CCC)
//=====
private static final int MSG_BROADCAST_SIGNAL = 100;
private HandlerThread mHandlerThread;
private UnitPrefController.SignalBroadcastHandler mSignalBroadcastHandler;
class SignalBroadcastHandler extends Handler {
    SignalBroadcastHandler(Looper looper) {
        super(looper);
    }
    @Override
    public void handleMessage(Message msg) {
        if (msg.what == MSG_BROADCAST_SIGNAL)
            broadcastCccSignalW(msg.arg1, msg.arg2);
    }
    void broadcastCccSignalW(int propId, int value) {
        try {
            mPropertyManager.setIntProperty(propId, 0, value);
        } catch (android.car.CarNotConnectedException e) {
            Log.e(TAG, "broadcastCccSignalW() : VHAL not connected.");
        }
    }
}

```

```

    }
}

//=====
// Request signals
//=====
public boolean isUKMarket() {
    return mMarket == Market.UNITED_KINGDOM;
}
public boolean hasValidMarket() {
    return mMarket != Market.UNKNOWN;
}
public boolean hasValidRegion() {
    return mRegion != Region.UNKNOWN;
}
public Market getDefaultMarket() {
    return mMarket;
}
public Region getDefaultRegion() { // 硬件配置的区域
    return mRegion;
}
private String parseStringSafely(CarPropertyValue<String> carPropertyValue) {
    String value = "";
    if (carPropertyValue == null)
        return value;
    return carPropertyValue.getValue();
}
private void log(String msg) {
    Log.d(TAG, msg);
}
private boolean isDebuggable() {
    boolean debuggable = false;
    PackageManager pm = mContext.getPackageManager();
    try {
        ApplicationInfo appInfo = pm.getApplicationInfo(mContext.getPackageName(), 0);
        debuggable = (0 != (appInfo.flags & ApplicationInfo.FLAG_DEBUGGABLE));
    } catch (Exception e) {
    }
    return debuggable;
}
private static final int BCM_TO_CCC_IMARKET = 0x2110116B;
private static final int BCM_TO_CCC_IREGION = 0x2110116C;
public boolean isEuRegion() { // 这个公共接口可以一用
    return mRegion == Region.EU;
}
// PM hasn't defined all the default values for all the countries. we will use this for now. may change it in the future
public enum Market {
    UNKNOWN("", DistanceUnit.MILES, TimeFormat.TWELVE_HOUR, TemperatureFormat.FAHRENHEIT, TirePressureFormat.PSI),
    CANADA(MARKET_CANADA, DistanceUnit.KILOMETERS, TimeFormat.TWENTY_FOUR_HOUR, TemperatureFormat.CELSIUS, TirePressureFormat.PSI),
    SAUDI_ARABIA(MARKET_SAUDI_ARABIA, DistanceUnit.KILOMETERS, TimeFormat.TWELVE_HOUR, TemperatureFormat.FAHRENHEIT, TirePressureFormat.PSI),
    CHINA(MARKET_CHINA, DistanceUnit.KILOMETERS, TimeFormat.TWELVE_HOUR, TemperatureFormat.CELSIUS, TirePressureFormat.PSI),
    UNITED_STATES(MARKET_US, DistanceUnit.MILES, TimeFormat.TWELVE_HOUR, TemperatureFormat.FAHRENHEIT, TirePressureFormat.PSI),
    UNITED_KINGDOM(MARKET_UK, DistanceUnit.MILES, TimeFormat.TWELVE_HOUR, TemperatureFormat.FAHRENHEIT, TirePressureFormat.PSI),
    public String value;
    public DistanceUnit unit;
    public TemperatureFormat temperatureFormat;
    public TimeFormat timeFormat;
    public TirePressureFormat tirePressureFormat;
    Market(String val, DistanceUnit unit, TimeFormat timeFormat, TemperatureFormat temperatureFormat, TirePressureFormat tirePressureFormat) {
        this.value = val;
        this.unit = unit;
        this.temperatureFormat = temperatureFormat;
        this.timeFormat = timeFormat;
        this.tirePressureFormat = tirePressureFormat;
    }
    public static Market valueOfMarket(String value) {
        for (Market market : Market.values()) {
            if (market.value.equals(value))
                return market;
        }
        return Market.UNKNOWN;
    }
}
public enum Region {
    UNKNOWN("", DistanceUnit.MILES),

```

```

        ASIA(REGION_ASIA, DistanceUnit.KILOMETERS),
        EU(REGION_EUROPE, DistanceUnit.KILOMETERS),
        MIDDLE_EAST(REGION_MIDDLE_EAST, DistanceUnit.KILOMETERS),
        NORTH_AMERICA(REGION_NORTH_AMERICA, DistanceUnit.MILES);
        public String value;
        public DistanceUnit unit;
        Region(String val, DistanceUnit unit) {
            this.value = val;
            this.unit = unit;
        }
        public static Region valueOfRegion(String value) {
            for (Region region : Region.values()) {
                if (region.value.equals(value))
                    return region;
            }
            return Region.UNKNOWN;
        }
    }
}

```

## 1.13 DriveModeController.java

- 这个应该还是不相关的

```

public class DriveModeController {
    private static final int SIGNAL_TIMEOUT_MILLISEC = 3000;

    public static class DriveModeListener {
        public void onVhalConnectedW() { }
        public void onVhalDisconnectedW() { }
        public void onPropertiesReadyW() { }

        public void onUpdateDriveModeW(Integer state) { }
        public void onUpdateVehicleSpeedW(Float speed) { }
    }

    //=====
    // Connection to VHal
    //=====

    private Car mCarApiClient;
    private CarPropertyManager mPropertyManager;
    private ValuesDumper mDumper;

    private final ServiceConnection mCarConnectionCallback = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            if (mDriveModeListener != null) {
                Log.d(TAG, "mDriveModeListener.onVhalConnectedW()");
                mDriveModeListener.onVhalConnectedW();
            }
            try {
                mPropertyManager = (CarPropertyManager) mCarApiClient
                    .getCarManager(android.car.Car.PROPERTY_SERVICE);
                for (int propId : S_NOTIFY_PROP_IDS)
                    mPropertyManager.registerCallback(mPropertiesListener, propId, 100);
                // Fetch initial values. 它只监听了两个信号，并且都不相关
                int driveMode = mPropertyManager.getIntProperty( CCC_DRIVE_MODE_STATUS, 0 );
                float speed = mPropertyManager.getFloatProperty( PERF_VEHICLE_SPEED, 0 );
                synchronized ( mPropertiesLock ) {
                    mDriveMode = driveMode;
                    mLastSpeed = speed;
                }
            } catch (android.car.CarNotConnectedException e) {
                Log.e(TAG, "onConnected() : Car not connected.");
            }

            mPropertiesReady = true;

            // Notify
            if (mDriveModeListener != null) {
                Log.d(TAG, "mDriveModeListener.onPropertiesReadyW()");
                mDriveModeListener.onPropertiesReadyW();
            }
        }
    };
}

```

```

    }
    if( IS_DEBUG_BUILD ) {
        mDumper = new ValuesDumper(TAG, S_NOTIFY_PROP_IDS, mSignalBroadcastHandler, mPropertyManager);
    }
}
}}

```

## 1.14 adaptive 的逻辑

- 之前没有想到，这里有一点问题是：是 adaptive 的，那么 supposedly，它应该秀聪明，能够自动感受来自对面车道的光。而如果对面有光足够强，那么它会自动关掉 high beam；而等对面的车过去，光强消失，那么它会再自动打开
- 原则上来讲，这些逻辑至少应该是 BCM-VHAL team 把这个 Feature 包装好，提供给我们上层应用层来直接使用。
- 因为越是接近底层接近硬件层，它的逻辑与控制实现起来，性能方面应该表现都是最好的；而越是靠近应用层，如果需要在应用层包装，就太消耗性能了
- supposedly，vHal team 把这个 feature 包装好之后，提供一个信号给我们应用层使用
- 可是如果不是这样，我要如何实现这个 feature 呢，逻辑控制?????

## 1.15 LucidCarSettings 应用中 HomeActivity 中重要逻辑再稍微熟悉一下

```

public class HomeActivity extends AppCompatActivity {
    private static final String TAG = HomeActivity.class.getName();

    private AsyncResources asyncResources;
    public AsyncResources getAsyncResources() {
        return asyncResources;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        asyncResources = AsyncResources.getInstance(this.getApplicationContext()); // 全局唯一 Singleton 吗，还是静态的呢？
        asyncResources.registerConnection(connectionsListener); // 注册监听者

        mProfileData = new ProfileData(getApplicationContext()); // IPC AIDL 的服务联接交互
        mProfileData.bindService(new ProfileData.ServiceConnectionListener() {
            @Override public void onConnected() {
                mIsBindServiceCalled = true;
                UserProfileData.setProfileDataInstance(mProfileData);
            }
            @Override public void onDisconnected() {
                UserProfileData.setProfileDataInstance(null);
            }
        });
        boolean showProfileTab = getIntent().getBooleanExtra("SHOW_PROFILE_TAB", false);
        //boolean shouldShowDialog = getIntent().getBooleanExtra("SHOW_DIALOG", false);
        // 用户帐户相关的部分
        IntentFilter intentFilter = new IntentFilter();
        intentFilter.addAction(Utils.ACTION_OPEN_PROFILE_SETTINGS);
        intentFilter.addAction(Utils.ACTION_UPDATE_LUCID_ID_CONNECTION_STATUS);
        registerReceiver(mProfileSettingsUIReceiver, intentFilter);
        IntentFilter filter = new IntentFilter();
        filter.addAction(ACTION_CREATE_NEW_USER_PROFILE);
        filter.addAction(Utils.ACTION_DISPLAY_HOME_ACTIVITY_VIEW);
        getApplicationContext().registerReceiver(mProfileReceiver, filter);

        setHomeActivityView(showProfileTab); // 默认显示用户帐户

        registerMediaApplicationBroadcast();
        registerDeepLinkBroadcast();
        setUpViewModel();
        setUpObserver(); // 这里可能也需要去看一下
    }
}

```

```

    // // This is required so that whenever the car settings are opened,
    // // application can essentially "invoke" the bluetooth service to initialize.
    // BluetoothUtils.getLocalBtManager(getApplicationContext()); //暂时不管
    // AtmosReceiver.connectReceiver(getApplicationContext()); // 不知道这个是在干什么
}
// @Override public void onPause() {
//     super.onPause();
// }

@Override
protected void onStart() {
    // we need to initialize controllers before calling super.onStart()
    // as some of the fragments may be dependent on those
    Context context = getApplicationContext(); // 获得应用层面的 Context
    if (context == null) {
        Log.e(TAG, "ERROR - CONTEXT IS NULL, SO CONTROLLERS WILL CRASH");
        return;
    }
    initControllers(context); //
    // context.registerReceiver(mThemeBroadcastReceiver, CarTheme.getThemeIntentFilter()); //我觉得目前这个不相关
    super.onStart();
}
@Override
protected void onStop() {
    super.onStop();
    // getApplicationContext().unregisterReceiver(mThemeBroadcastReceiver); //这个东西目前对我不重要
    cleanUpControllers(); //
}
@Override
protected void onDestroy() { // "HomeActivity onDestroy called";
    getViewModelStore().clear(); // ?????? 这是在干什么呢?
    if (mIsBindServiceCalled) {
        mIsBindServiceCalled = false; //如果建立了远程服务联接, 现在断开
        mProfileData.unbindService(); //断开联接
    }
    // getApplicationContext().unregisterReceiver(mProfileReceiver); // 前面不是注册了这个监听吗?
    // unregisterReceiver(mProfileSettingsUIReceiver); // 它也只干两件事: 负责显示用户帐户相关 Fragment, 以及
    // unregisterMediaApplicationBroadcast();
    // unregisterDeepLinkBroadcast(); // 它只干一件事
    asyncResources.unregisterConnections(connectionsListener);
    AtmosReceiver.disconnectReceiver(getApplicationContext());
    super.onDestroy();
}
public void initControllers(Context context) { // 三个 !!!
    Log.d(TAG, "init controllers");
    bcmPresenter = new BcmPresenter(context);
    gearPresenter = new GearPresenter(context);
    alertPresenter = new AlertPresenter(context);
}
private void cleanUpControllers() {
    Log.d("TAG", "cleanUpControllers");
    if (bcmPresenter != null) {
        bcmPresenter.cleanup();
        bcmPresenter = null;
    }
    if (gearPresenter != null) {
        gearPresenter.cleanup();
        gearPresenter = null;
    }
    if (alertPresenter != null) {
        alertPresenter.cleanup();
        alertPresenter = null;
    }
}
// 它只干一件事: LucidSystemUI 应用中: IcrSettingsWindow 中几个 Tabs 中有每个 Tab 中那个键点击后, 会广播要求 CID LucidCarSet
private void registerDeepLinkBroadcast(){
    IntentFilter filter = new IntentFilter();
    filter.addAction(CarSettingsConstants.ACTION_DEEP_LINK_TRIGGERED);
    registerReceiver(mDeepLinkReceiver, filter);
}
private void unregisterDeepLinkBroadcast(){
    unregisterReceiver(mDeepLinkReceiver);
}
private BroadcastReceiver mDeepLinkReceiver = new BroadcastReceiver() {

```



```

@Override
public void onReceive(Context context, Intent intent) { // 它只干一件事: 要打开哪个就打开哪个
    String action = intent.getAction();
    if(CarSettingsConstants.ACTION_DEEP_LINK_TRIGGERED.equals(action)){
        Log.d(TAG, "Deep Link Triggered");
        loadStartScreenFromDeepLink();
    }
}

};

public static boolean getPrimaryLucidIdStatus(Context context) { // 来自 Utils.java: IPC 进程间通信, 用户帐户连起来了吗?
    String isLucidIdConnected = Settings.Global.getString(context.getContentResolver(),
        IS_LUCID_ID_CONNECTED);

    Log.d(TAG, "getPrimaryLucidIdStatus() " + isLucidIdConnected + " boolean value: " + Boolean.parseBoolean(isLucidIdConnected));
    return Boolean.parseBoolean(isLucidIdConnected);
}

private void refreshUserProfileLucidIdStatus(){
    UserInfo userInfo = mCarUserManagerHelper.getCurrentForegroundUserInfo(); // 前台用户
    Bitmap bitmap = mCarUserManagerHelper.getUserIcon(userInfo); // 用户圆形头像
    if (bitmap != null) {
        Bitmap circleBitmap = Utils.getCircularBitmap(bitmap);
        mProfilePicture.setImageDrawable(new BitmapDrawable(getResources(), circleBitmap));
    }
    mProfileName.setText(userInfo.name); // 用户名
    if (ProfileAccountType.ProfileType.PRIMARY_USER == Utils.getProfileType()) { // PRIMARY DRIVER
        mOwnerText.setVisibility(View.VISIBLE);
        mProfilePicture.setEnabled(true);
        mLucidIdConnectionStatus = Utils.getPrimaryLucidIdStatus(getApplicationContext()); // 定义在上面
        if (mLucidIdConnectionStatus) // 自动导出 PRIMARY DRIVE 的邮箱
            mProfileEmailId.setTitle(getApplicationContext().getString(R.string.alex_sign_out_button).toUpperCase());
        else // 邮箱缺省为: "CONNECT YOUR LUCID ID"
            mProfileEmailId.setTitle(getString(R.string.connect_lucid_id_text));
    }
    // 不知道这些监听回调现在在干什么鬼事, 现在不用管它们
    mProfileEmailId.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (!mLucidIdConnectionStatus){
                mLucidIdConnectionModal = new LucidIdConnectionModal(getApplicationContext(), mMessenger);
                if (getSupportFragmentManager() != null)
                    mLucidIdConnectionModal.launch(getSupportFragmentManager());
            }else{
                UserInfo userInfo = mCarUserManagerHelper.getCurrentForegroundUserInfo();
                Bitmap bitmap = mCarUserManagerHelper.getUserIcon(userInfo);
                LucidIdSignOutRequestModal lucidIdSignOutRequestModal = new LucidIdSignOutRequestModal(getApplicationContext(), mMessenger);
                if (getSupportFragmentManager() != null)
                    lucidIdSignOutRequestModal.show(getSupportFragmentManager(), "LucidIdSignOutRequestModal");
            }
        }
    });
    mProfilePicture.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (ProfileAccountType.ProfileType.PRIMARY_USER == Utils.getProfileType()){
                if (!mLucidIdConnectionStatus){
                    EditProfileNameAvatarModal editProfileNameAvatarModal = new EditProfileNameAvatarModal(getApplicationContext(), mMessenger);
                    if (getSupportFragmentManager() != null)
                        editProfileNameAvatarModal.show(getSupportFragmentManager(), "EditProfileNameAvatarModal");
                }else{
                    UserInfo userInfo = mCarUserManagerHelper.getCurrentForegroundUserInfo();
                    Bitmap bitmap = mCarUserManagerHelper.getUserIcon(userInfo);
                    MobileAppModal mobileAppModal = new MobileAppModal(getApplicationContext(), userInfo.name, bitmap);
                    if (getSupportFragmentManager() != null)
                        mobileAppModal.show(getSupportFragmentManager(), "MobileAppModal");
                }
            }else if (ProfileAccountType.ProfileType.SECONDARY_USER == Utils.getProfileType()){
                EditProfileNameAvatarModal editProfileNameAvatarModal = new EditProfileNameAvatarModal(getApplicationContext(), mMessenger);
                if (getSupportFragmentManager() != null)
                    editProfileNameAvatarModal.show(getSupportFragmentManager(), "EditProfileNameAvatarModal");
            }
        }
    });
}

BroadcastReceiver mProfileSettingsUIReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {

```

```

Log.d(TAG, "onReceive() with action: " + intent.getAction());
if (Utils.ACTION_OPEN_PROFILE_SETTINGS.equals(intent.getAction())) { // 如果收到要求显示用户帐户设置, 那绝隐藏当前界面
    mLayoutApplication.setVisibility(View.INVISIBLE);
    mCustomViewApplication.setVisibility(View.INVISIBLE);
    mCustomView.setVisibility(View.VISIBLE);
    if (currentlySelected != null && currentlySelected != accessProfileListItem) {
        currentlySelected.unhighlight();
        currentlySelected = accessProfileListItem;
        accessProfileListItem.highlight();
    }
    FragmentManager fragmentManager = getSupportFragmentManager(); // 应该是到用户帐户界面相关的什么东西
    Fragment profileFragment = fragmentManager.findFragmentByTag("AccessAndProfileFragment");
    if (fragmentManager.getFragments().size() == 0) {
        Log.d(TAG, "No fragments found");
        FragmentTransaction ft = fragmentManager.beginTransaction();
        ft.add(R.id.home_activity_secondary_view, new AccessAndProfileFragment(), "AccessAndProfileFragment");
        ft.commitNow();
    } else if (profileFragment == null || !profileFragment.isVisible()) {
        Log.d(TAG, "There is no profile UI visible to user");
        FragmentTransaction ft = fragmentManager.beginTransaction();
        ft.replace(R.id.home_activity_secondary_view, new AccessAndProfileFragment(), "AccessAndProfileFragment");
        ft.commitNow();
    }
    mProfileRecentlyOpened = true;
    new Handler().postDelayed(() -> mProfileRecentlyOpened = false, 1000);
} else if (Utils.ACTION_UPDATE_LUCID_ID_CONNECTION_STATUS.equals(intent.getAction())) {
    Log.d(TAG, "Broadcast received ACTION_UPDATE_LUCID_ID_CONNECTION_STATUS");
    refreshUserProfileLucidIdStatus();
    getApplicationContext().sendBroadcast(new Intent(Utils.ACTION_UPDATE_RESET_PIN_ACCESS)); // 为什么这里崩出来
}
}
};
BroadcastReceiver mProfileReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) { // 它只做两件事
        Log.d(TAG, "onReceive() with action: " + intent.getAction());
        if (ACTION_CREATE_NEW_USER_PROFILE.equals(intent.getAction())) {
            int imageId = intent.getIntExtra(EXTRA_AVATAR_INFO, 0);
            int[] images = Utils.getImages();
            String profileName = intent.getStringExtra(EXTRA_USER_PROFILE_NAME);
            mProfilePicture.setImageResource(images[imageId]);
            mProfileName.setText(profileName);
        } else if (Utils.ACTION_DISPLAY_HOME_ACTIVITY_VIEW.equals(intent.getAction())) {
            setHomeActivityView(false); // 大致是一大堆跟显示这个活动以及用户帐户相关的东西
        }
    }
};

```

## 2 jenny/IVIANDAPPS-6093: [SCCM]{SW-issue} - Steering wheel right side buttons unoperational

- <https://lucidmotors.atlassian.net/browse/IVIANDAPPS-6093>
- for Audio Volume related, I think this page has to be understood: <https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/2666988689/Spec+Android+Audio+1.9>
- Peek Window use cases and corner cases: <https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/2213478978/Peek>
  - 注意这里: HvacPeek 与 VolumePeek 这里有个内循环

### 2.1 SteeringWheelController.java should be involved for these volume buttons (According to Soumya)

```

//public class SteeringWheelController {
public static class SteeringWheelListener {

```

```

public void onVhalConnectedW() { }
public void onVhalDisconnectedW() { }
public void onPropertiesReadyW() { }
// Callback for on Voice Button status.
// NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
// FailureDetected:6, Invalid:7
public void onUpdateVoiceButtonW(int value) { }
// Callback for skip previous button.
// NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
// FailureDetected:6, Invalid:7
public void onUpdateSkipPreviousW(int value){ }
// Callback for skip next button.
// NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
// FailureDetected:6, Invalid:7
public void onUpdateSkipNextW(int value){ }

// Callback for Rocket Right Button.
// NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
// FailureDetected:6, Invalid:7
public void onUpdateRockerRightButtonW(int value){
    Log.d(TAG, "onUpdateRockerRightButtonW() "); // this log will OFTEN @Override by subclass, so won't NECESSARILY see this
}
// Callback for Toggle Right Button.
// NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
// FailureDetected:6, Invalid:7
public void onUpdateToggleRightW(int value) {
    Log.d(TAG, "onUpdateToggleRightW() "); // this log will OFTEN @Override by subclass, so won't NECESSARILY see this
}
}
}

```

## 2.2 目前 LucidSystemUI 中处理过的几个点面：这三个来源分别是什么意思呢？

```

public static String LUCID_MEDIA_MUTE = "LUCID_MEDIA_MUTE";
public static String LUCID_NAVIGATION_MUTE = "LUCID_NAVIGATION_MUTE";
public static String LUCID_PHONE_MUTE = "LUCID_PHONE_MUTE";

public static final int MEDIA_VOL_GROUP_ID = 0;
public static final int NAV_VOL_GROUP_ID = 1;
public static final int PHONE_VOL_GROUP_ID = 2;

private final ServiceConnection mServiceConnection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        try {
            mCarAudioManager = (CarAudioManager) mCar.getCarManager(Car.AUDIO_SERVICE);
            mCarAudioManager.registerCarVolumeCallback( mVolumeChangeCallback );
            // Restore volume levels: 是包括在哪个版块里面呢？
            restoreVolumeLevel( MEDIA_VOL_GROUP_ID );
            restoreVolumeLevel( NAV_VOL_GROUP_ID );
            restoreVolumeLevel( PHONE_VOL_GROUP_ID );
            // Update UI
            int vol = mCarAudioManager.getGroupVolume( MEDIA_VOL_GROUP_ID );
            mVolumeProgress.setProgress( (vol*3), true );
        } catch ( CarNotConnectedException e) {
            Log.e(TAG, "Car is not connected!");
        }
    }
    @Override
    public void onServiceDisconnected(ComponentName name) {
        try {
            mCarAudioManager.unregisterCarVolumeCallback( mVolumeChangeCallback );
        } catch (CarNotConnectedException e) {
            Log.e(TAG, "Car is not connected!");
        }
        mCarAudioManager = null;
    }
};

```

- CarAudioManager.CarVolumeCallback mVolumeChangeCallback 回调

```

// Multi-thread access. Need to be locked.
private Integer mVolGroupId = 0;

```

```
// 需要理解 这个 Property 包括几个和哪几个按钮的信号: ???
private final CarAudioManager.CarVolumeCallback mVolumeChangeCallback = new CarAudioManager.CarVolumeCallback() {
    @Override
    public void onGroupVolumeChanged(int zoneId, int groupId, int flags) { // flags parameter 什么意思?
        Log.d(TAG, "onGroupVolumeChanged() ");
        super.onGroupVolumeChanged(zoneId, groupId, flags);
        // Only show volume peek window if flag is set.
        Log.d(TAG, "( flags & AudioManager.FLAG_SHOW_UI ): " + ( flags & AudioManager.FLAG_SHOW_UI )); // 会 == 0 吗
        if ( ( flags & AudioManager.FLAG_SHOW_UI ) != 0 ) { // flags: 是什么 意思 呢?
            // 0- music, 1-nav, 2- call, 3-other
            synchronized( mVolGroupId ) {
                mVolGroupId = groupId;
            }
            Log.d(TAG, "mVolGroupId: " + mVolGroupId);
            mHandler.post( mUpdateVolumeUI );
        }
        // Persist volume level as soon as it's changed.
        persistVolumeLevel( groupId );
    }
}
@Override
public void onMasterMuteChanged(int zoneId, int flags) {
    Log.d(TAG, "onMasterMuteChanged() ");
    super.onMasterMuteChanged(zoneId, flags);
}
};
```

## 2.3 logs

- when steering wheel buttons are pushed

```
D IcrVolumeWindow: SteeringWheel Controller VHAL connected
D IcrVolumeWindow: restoreVolumeLevel(): groupId=0, vol=11 // 这个函数调用的目的是什么呢?
D IcrVolumeWindow: onGroupVolumeChanged()
D IcrVolumeWindow: ( flags & AudioManager.FLAG_SHOW_UI ): 0
D IcrVolumeWindow: muteUnmute() groupId: 0 volume: 11
D IcrVolumeWindow: LUCID_MEDIA_MUTE=0
D IcrVolumeWindow: persistVolumeLevel(): groupId=0, vol=11
D IcrVolumeWindow: restoreVolumeLevel(): groupId=1, vol=17
D IcrVolumeWindow: onGroupVolumeChanged()
D IcrVolumeWindow: ( flags & AudioManager.FLAG_SHOW_UI ): 0
D IcrVolumeWindow: muteUnmute() groupId: 1 volume: 17
D IcrVolumeWindow: LUCID_NAVIGATION_MUTE=0
D IcrVolumeWindow: persistVolumeLevel(): groupId=1, vol=17
D IcrVolumeWindow: restoreVolumeLevel(): groupId=2, vol=17
D IcrVolumeWindow: onGroupVolumeChanged()
D IcrVolumeWindow: ( flags & AudioManager.FLAG_SHOW_UI ): 0
D IcrVolumeWindow: persistVolumeLevel(): groupId=2, vol=17
D IcrVolumeWindow: muteUnmute() groupId: 2 volume: 17
D IcrVolumeWindow: LUCID_PHONE_MUTE=0
```

## 2.4 Concepts to be understood first

- Steering wheel buttons



- 好像是第 4 第 5 个按钮

- pages are: <https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/1738905435/Steering+Wheel+Buttons+Signals+-+Infotainment>
- <https://lucidmotors.atlassian.net/wiki/spaces/TET/pages/2690319222/AIR+Steering+Wheel+2.0>

#4 Right - Toggle/Rocker Center Button	ISWS_RockerRightButtonSts	0 == NOT PRESSED 1 == SHORT PRESS 2 == LONG PRESS 6 == FAILURE DETECTED 7 == INVALID
#5 Right - Toggle Rocker	ISWS_ToggleRightSts	0 == NOT PRESSED 1 == TOGGLE UP 2 == HOLD UP 3 == TOGGLE DOWN 4 == HOLD DOWN 6 == FAILURE DETECTED 7 == INVALID

```
// public class SteeringWheelController {
public static class SteeringWheelListener {

    // Callback when Vhal is connected.
    public void onVhalConnectedW() {
    }
    // Callback when Vhal is disconnected.
    public void onVhalDisconnectedW() {
    }
    // Callback after Vhal is connected and properties have been fetched.
    public void onPropertiesReadyW() {
    }
    // Callback for on Voice Button status.
    // NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
    // FailureDetected:6, Invalid:7
    public void onUpdateVoiceButtonW(int value) {
    }
    // Callback for skip previous button.
    // NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
    // FailureDetected:6, Invalid:7
    public void onUpdateSkipPreviousW(int value){
    }
    // Callback for skip next button.
    // NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
    // FailureDetected:6, Invalid:7
    public void onUpdateSkipNextW(int value){
    }

    // Callback for Rocket Right Button.
    // NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
    // FailureDetected:6, Invalid:7
    public void onUpdateRockerRightButtonW(int value){
    }
    // Callback for Toggle Right Button.
    // NotPressed:0, ShortPress:1, LongPress:2, TBD1:3, TBD2:4, TBD3:5,
```

```
// FailureDetected:6, Invalid:7
public void onUpdateToggleRightW(int value) {
}
}
```

- ACC status: ACC + LC Controllers: 分别是什么? Adaptive cruise control, lane Centering, Lane Keep Assist, 这里是需要分驾驶习惯, 是左驾还是右驾还决定是在哪一边
  - 另一边会是 Media Controller

ADAS	Media
ACC Activation	Skip Track Forward
Lane Centering Activation	Skip Track Backward
ACC Speed Limit	Voice
Gap setting	Volume up/down
	Mute

2.5 a few trials from the car

- why when I clicked sync from CID, hvacController is checking is steering heating?

```
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 2
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.2 temp 25.2
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D HvacPresenter: [HvacPresenter.handleInitialPropertySignal():131] IS steering heat state : false
D HvacPresenter: [HvacPresenter.manuallyGetUpdatedValues():280] Check Steering heat state = false
D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.2 temp 25.2
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.2 temp 25.2
D HvacPresenter: [HvacPresenter.handleInitialPropertySignal():131] IS steering heat state : false
D HvacPresenter: [HvacPresenter.manuallyGetUpdatedValues():280] Check Steering heat state = false
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.2 temp 25.2
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 2
D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUiService.ACTION_WIFI_SIGNAL
D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.2 temp 25.2
I_SIGNAL why hvacController keeps calling this?
05-12 17:17:18.461 11372 11372 D HvacPresenter: [HvacPresenter.handleInitialPropertySignal():131] IS steering heat state :
05-12 17:17:18.461 11372 11372 D HvacPresenter: [HvacPresenter.manuallyGetUpdatedValues():280] Check Steering heat state =
05-12 17:17:18.731 11139 11139 D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 1
05-12 17:17:18.737 11139 11139 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-12 17:17:19.499 12344 12732 I AHE-HybridChannelController: Session updated (Switched):currentSessionId=12,nextSessionId=
05-12 17:17:19.731 11139 11139 D LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.1 temp 25.1
05-12 17:17:19.731 11139 11139 D LucidSysUiService: onWifiSignalUpdate Wifi_Strength:..... 2
```

2.5.1 问题: 即是车的 onChangeEvent(), 为什么得到的值, 进一步传时来的参数又是 false 呢?

- I want to understand one part of logic: after 1 of the 4 hvac physical buttons is touch, within maximumly how long in milliseconds, we won't receive any callback from vhal that INPUTS\_HVAC\_BTNS\_ENGAGED property has changed again to be untouched?

```
public class BcmController {
    private static final String TAG = "BcmController";
```

```

    private static final int VCU_SIGNAL_TIMEOUT = 1 * 1000; // this timeout will be used for REGEN, STOPMODE, and CREEP MOD
//=====
// Incoming BCM Signals Handling
//=====
private CarPropertyManager.CarPropertyEventCallback mPropertiesListener =
    new CarPropertyManager.CarPropertyEventCallback() {
        @Override
        public void onErrorEvent(int i, int i1) {
            Log.e( TAG, "onErrorEvent() : i=" + i + ", i1=" + i1 );
        }
        @Override
        public void onChangeEvent(CarPropertyValue carPropertyValue) { // 如果时间足够短, 也不会传到 onChangeEvent(CarPropertyVal
            if (IS_DEBUG_BUILD)
                Log.d(TAG, " onChangeEvent " + carPropertyValue.getPropertyId()
                    + " (0x" + Integer.toHexString(carPropertyValue.getPropertyId()) + ")");
            if (carPropertyValue.getPropertyId() == INPUTS_HVAC_BTNS_ENGAGED) {
                Log.d(TAG, "(carPropertyValue.getPropertyId() == INPUTS_HVAC_BTNS_ENGAGED): " + (carPropertyValue.getProper
                    Log.d(TAG, "(Boolean)carPropertyValue.getValue(): " + (Boolean)carPropertyValue.getValue());
            }
            switch ( carPropertyValue.getPropertyId() ) {
                case CID_POSITION_STATUS :
                    handleUpdateCidStateW( (Integer) carPropertyValue.getValue() );
                    break;
// 4/15 log all getting true, 1-peek from yesterdsay getting all false, so I simply handle false situations to show IcrHvac
                case INPUTS_HVAC_BTNS_ENGAGED : // 说明这些东西的属性已经变化了, 可是为什么取得的值总是 false 呢? 触摸之后显示是 true 的
                    handleUpdateHvacBtnsEngagedW( (Boolean) carPropertyValue.getValue() ); // this method is called, but passed in
                    break;
            }
            //...
        }
    };
void handleUpdateHvacBtnsEngagedW(Boolean value) {
    if ( mBcmListener!= null ) {w
        if (IS_DEBUG_BUILD) Log.d( TAG, "mBcmListener.onEngagedHvacBtnsW() : value=" + value );
        mBcmListener.onEngagedHvacBtnsW( value );
    }
}
private final static int[] S_NOTIFY_PROP_IDS = new int [] {
    CID_POSITION_STATUS,
    INPUTS_HVAC_BTNS_ENGAGED, // WHY it was writen in 3 times?
    INPUTS_HVAC_BTN_DRIVER_TEMPERATURE,
    INPUTS_HVAC_BTN_DRIVER_FANSPEED,
    INPUTS_HVAC_BTN_PASSENGER_FANSPEED,
    INPUTS_HVAC_BTN_PASSENGER_TEMPERATURE,
    // INPUTS_HVAC_BTNS_ENGAGED, // WHY it was writen in 3 times?
    // INPUTS_HVAC_BTNS_ENGAGED, // WHY it was writen in 3 times?
}

```

## 2.6 today's testing

```

05-13 11:02:23.928 7855 7855 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 2
05-13 11:02:23.934 7855 7855 D LucidSysUIService: onWifiConnectionStatusChangedW wifiStatus: 2
05-13 11:02:23.940 7855 7855 D LucidSysUIService: onLteConnectionStatusChangedW Lte_Status: 2 Wifi_Status: 2
05-13 11:02:23.943 7855 7855 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-13 11:02:23.946 7855 7855 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-13 11:02:23.952 7855 7855 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 1
05-13 11:02:23.954 7855 7855 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-13 11:02:23.957 7855 7855 D BcmController: (carPropertyValue.getPropertyId() == INPUTS_HVAC_BTNS_ENGAGED): true
05-13 11:02:23.957 7855 7855 D BcmController: (Boolean)carPropertyValue.getValue(): false
05-13 11:02:23.957 7855 7855 E IcrVolumeWindow: onhvacbtn clickedfalse
05-13 11:02:24.024 7855 7855 D BcmController: handleOnUpdateOnePedalDrivingW 0
05-13 11:02:24.024 7855 7855 E BcmController: reported regen braking level is invalid
05-13 11:02:24.035 7855 7855 D BcmController: IVCU_LSC_MODE_STAT value 0
05-13 11:02:24.036 7855 7855 D BcmController: LSC Mode update value 0
05-13 11:02:24.036 7855 7855 D BcmController: creep mode 0
05-13 11:02:24.036 7855 7855 D BcmController: creep mode status = 0
05-13 11:02:24.082 7855 7855 D LucidSysUIService: onUpdateSkipPreviousW: 7
05-13 11:02:24.083 7855 7855 D LucidSysUIService: onUpdateSkipNextW: 7
05-13 11:02:24.083 7855 7855 D LucidSysUIService: onUpdateRockerRightButton()
05-13 11:02:24.083 7855 7855 D LucidSysUIService: onUpdateToggleRightW()
05-13 11:02:24.083 7855 7855 D LucidSysUIService: onUpdateToggleRightW: 7
05-13 11:02:24.084 7855 7855 D LucidSysUIService: Update is available0
05-13 11:02:24.089 7855 7855 D LucidSysUIService: ota update 0
05-13 11:02:24.096 7855 7855 D LucidSysUIService: onUpdateGearStateChangedW4
05-13 11:02:24.098 7855 7855 D LucidSysUIService: onUpdateGearStateChangedW4
05-13 11:02:24.099 7855 7855 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 2

```

```

05-13 11:02:24.104 7855 7855 D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 0.0 temp 25.0
05-13 11:02:24.104 7855 7855 D LucidSysUIService: onUpdateOutsideTemperatureW25.0

05-13 11:02:22.810 7855 7855 E LucidSysUIService: TCU vHal connected
05-13 11:02:22.816 7855 7928 D TcuController: fetchWifiSignalState strength: 1 status: 2 mCtrlsListener: com.lucid.car.sys
05-13 11:02:22.816 7855 7928 D LucidSysUIService: onWifiConnectionStatusChangedW wifiStatus: 2
05-13 11:02:22.818 7855 7928 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 1
05-13 11:02:22.827 7855 7928 D TcuController: fetchLteSignalState strength: 3 status: 2 mCtrlsListener: com.lucid.car.sys
05-13 11:02:22.827 7855 7928 D LucidSysUIService: onLteConnectionStatusChangedW Lte_Status: 2 Wifi_Status: 2
05-13 11:02:23.044 7855 7928 D BcmController: fetchProperties() : BEGIN
05-13 11:02:23.058 7855 7928 D BcmController: ONE_PEDAL_DRIVING_STS > ( 0 )
05-13 11:02:23.082 7855 7928 D BcmController: fetchProperties() : END
05-13 11:02:23.401 7855 7855 D LucidSysUIService: Steering Wheel VHAl Connected
05-13 11:02:23.522 7855 7855 D LucidSysUIService: Steering Wheel property ready
05-13 11:02:23.857 7855 7855 D IcrHvacPeekWindow: CCCQNXAppStatus value = -1
05-13 11:02:23.859 7855 7855 D IcrHvacPeekWindow: Gear value = 4
05-13 11:02:23.860 7855 7855 D BcmController: (carPropertyValue.getPropertyId() == INPUTS_HVAC_BTNS_ENGAGED): true
05-13 11:02:23.860 7855 7855 D BcmController: (Boolean)carPropertyValue.getValue(): false
05-13 11:02:23.860 7855 7855 D IcrHvacPeekWindow: onhvacbtn clickedfalse
05-13 11:02:23.865 7855 7855 D BcmController: handleOnUpdateOnePedalDrivingW 0
05-13 11:02:23.865 7855 7855 E BcmController: reported regen braking level is invalid
05-13 11:02:23.869 7855 7855 D BcmController: IVCU_LSC_MODE_STAT value 0
05-13 11:02:23.869 7855 7855 D BcmController: LSC Mode update value 0
05-13 11:02:23.869 7855 7855 D BcmController: creep mode 0
05-13 11:02:23.869 7855 7855 D BcmController: creep mode status = 0
05-13 11:02:23.890 7855 7855 D IcrHvacPeekWindow: fan speed changed2317
05-13 11:02:23.893 7855 7855 D IcrHvacPeekWindow: fan speed changed1018
05-13 11:02:23.895 7855 7855 D IcrHvacPeekWindow: fan speed changed033
05-13 11:02:23.897 7855 7855 D IcrHvacPeekWindow: fan speed changed034
05-13 11:02:23.899 7855 7855 D IcrHvacPeekWindow: temp changed16717
05-13 11:02:23.902 7855 7855 D IcrHvacPeekWindow: temp changed15018
05-13 11:02:23.904 7855 7855 D IcrHvacPeekWindow: temp changed15033
05-13 11:02:23.906 7855 7855 D IcrHvacPeekWindow: temp changed15034
05-13 11:02:23.928 7855 7855 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 2
05-13 11:02:23.934 7855 7855 D LucidSysUIService: onWifiConnectionStatusChangedW wifiStatus: 2
05-13 11:02:23.940 7855 7855 D LucidSysUIService: onLteConnectionStatusChangedW Lte_Status: 2 Wifi_Status: 2

```

- try to test in real time: it is NOT happending during real time !!!!!
- commands to see how many profiles are in background ????????
- all the system users ??????

u10_a10	9821	3003	5039824	120456	SyS_epoll+	0 S android.ext.services
u10_system	9834	3003	4970172	104976	SyS_epoll+	0 S com.android.car
root	9889	2	0	0 0		0 R [kworker/u16:1]
u10_a17	9891	3003	5152224	175224	SyS_epoll+	0 S com.android.systemui
u10_a6	9960	3003	4962148	104072	SyS_epoll+	0 S com.android.car.messenger
u10_a38	9990	3003	5515028	164716	SyS_epoll+	0 S com.clearchannel.iheartradio.controller
u10_a9	10012	3003	5000572	118400	SyS_epoll+	0 S android.process.media
u10_system	10079	3003	5142132	161060	SyS_epoll+	0 S com.lucid.car.carlauncher
u10_system	10095	3003	5373492	161796	SyS_epoll+	0 S com.lucid.car.sysui
root	10118	2	0	0 worker_th+		0 I [kworker/4:1]
u10_system	10132	3003	4974756	95900	SyS_epoll+	0 S com.android.keychain
u0_a17	10154	3003	5556600	193652	SyS_epoll+	0 S com.android.systemui
u10_a40	10186	3003	5881656	208992	SyS_epoll+	0 S tunein.player
u10_system	10209	3003	5251548	140344	SyS_epoll+	0 S com.lucid.car.media
u10_a39	10221	3003	5641996	157996	SyS_epoll+	0 S com.aspiro.tidal
u10_system	10267	3003	5261972	179288	SyS_epoll+	0 S com.lucid.car.carsettings
u10_system	10328	3003	6029872	221432	SyS_epoll+	0 S com.lucid.car.carcontrols
u10_system	10372	3003	5216400	167500	SyS_epoll+	0 S com.lucid.car.hvac
u10_system	10610	3003	4982600	99200	SyS_epoll+	0 S com.lucid.userprofile
u10_a30	10724	3003	4948688	112984	SyS_epoll+	0 S com.android.inputmethod.latin
u10_a45	10762	3003	4963724	100424	SyS_epoll+	0 S com.android.contacts
u10_system	10800	3003	4922468	93228	SyS_epoll+	0 S com.android.dynsystem
u10_a3	10825	3003	4954280	115032	SyS_epoll+	0 S android.process.acore
u10_a18	10865	3003	4927232	90184	SyS_epoll+	0 S com.android.managedprovisioning
u10_a42	10892	3003	4922284	87336	SyS_epoll+	0 S com.android.onetimeinitializer
root	10909	2	0	0 worker_th+		0 I [kworker/u16:3]
u10_a13	10915	3003	4941296	86660	SyS_epoll+	0 S com.android.packageinstaller
u10_a21	10936	3003	4945160	94392	SyS_epoll+	0 S com.android.permissioncontroller
u10_system	10975	3003	4942572	100524	SyS_epoll+	0 S com.google.android.car.kitchensink



```
u10_system 11000 3003 6719856 227724 binder_io+ 0 S com.here.hnod.feynman
root 11021 2 0 0 worker_th+ 0 I [kworker/7:0]
u10_system 11128 3003 6033132 219708 Sys_epoll+ 0 S com.lucid.assistant
u10_system 11161 3003 4976048 102040 Sys_epoll+ 0 S com.lucid.car.dialer
u10_system 11211 3003 5093844 118280 Sys_epoll+ 0 S com.lucid.car.service
u10_system 11235 3003 4960524 99312 Sys_epoll+ 0 S com.amazon.alexalve
u10_system 11261 3003 6463448 730976 Sys_epoll+ 0 S com.amazon.alexalve:ahe
u10_system 11291 3003 4992320 110652 Sys_epoll+ 0 S com.amazon.alexalve:sc_car_control
u10_system 11314 3003 4989168 110244 Sys_epoll+ 0 S com.amazon.alexalve:sc_comms
u10_system 11337 3003 5004468 110084 Sys_epoll+ 0 S com.amazon.alexalve:sc_media
u10_system 11356 3003 5042728 109208 Sys_epoll+ 0 S com.amazon.alexalve:sc_nav
u10_system 11377 3003 5075344 107420 Sys_epoll+ 0 S com.amazon.alexalve:contacts
u10_system 11399 3003 4988108 105880 Sys_epoll+ 0 S com.amazon.alexalve:leds
root 11427 2 0 0 worker_th+ 0 I [kworker/3:0]
u10_a22 11450 3003 4976880 97508 Sys_epoll+ 0 S com.lucid.ota_updater
u10_system 11506 3003 4923452 95504 Sys_epoll+ 0 S com.lucid.otaupdaterservice
u10_system 11529 3003 4985896 99884 Sys_epoll+ 0 S com.lucid.profilessetup
u10_a29 11638 3003 5831316 183584 Sys_epoll+ 0 S com.spotify.music
u10_system 11775 3003 5170996 145196 Sys_epoll+ 0 S com.android.car.radio
u10_bluetoo+ 11793 3003 4946804 103872 Sys_epoll+ 0 S com.android.bluetooth
u10_a34 11816 3003 5376672 136004 Sys_epoll+ 0 S au.com.shiftyjelly.pocketcasts
root 12008 2 0 0 worker_th+ 0 I [kworker/1:2]
root 12037 8599 34220 8680 __skb_wai+ 0 S logcat
root 12635 8599 33496 8356 Sys_rt_si+ 0 S sh
root 12637 2 0 0 worker_th+ 0 I [kworker/2:3]
root 12649 12635 35980 9304 0 0 R ps
```

2.7 Leakcary: Revanth says we are having these memeory leaks

```
05-13 11:51:33.390 16693 16693 D LucidSysUIService: onWifiSignalUpdate Wifi_Strength:..... 2
05-13 11:51:33.393 16693 16693 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-13 11:51:33.646 16693 19316 D LeakCanary: Leaking: NO (LucidSysUIService is not leaking)
05-13 11:51:33.646 16693 19316 D LeakCanary: Leaking: NO (LucidSysUIService is not leaking)
05-13 11:51:33.646 16693 19316 D LeakCanary: mContext instance of com.lucid.car.sysui.LucidSysUIService
05-13 11:51:33.646 16693 19316 D LeakCanary: LucidSysUIService.mIcrVolumeWindow <-----
05-13 11:51:33.646 16693 19316 D LeakCanary: com.lucid.car.sysui.hvacvolpeak.IcrVolumeWindow instance
05-13 11:51:33.646 16693 19316 D LeakCanary: IcrVolumeWindow.mIcrPeakWindow <-----
05-13 11:51:34.237 16693 16693 D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.8 temp 25.8
```

- create ticket for the website: by pass website , by pass for only RC cars

vehicle commanding web page vin numbers take whole vin number connect to adb bypass the TCU so that I can connect to the car using adb keybox

3 jenny-IVIANDLIBS-342: Peek bar does not show up if we change temp/fan using hard buttons: 涉及原来安卓 9 的三个 Controller 与 VHAL 的交互

- <https://lucidmotors.atlassian.net/browse/IVIANDLIBS-342>
- 信号传不上来。Soumya is communicating with VHAL team to look into this signal.

3.1 这个应该有几个方面的原因:

- 为什么每次收到回调，它传回来的值都是 false 呢？它不是应该是能够收到回调，便是有物理键的涉入，应该永远传的是 true 才对吗？怎么会既能收到回调，又传回来的是 false 呢？

```
05-12 17:19:56.901 13877 13877 D IcrHvacPeekWindow: CCCQNXAppStatus value = -1
05-12 17:19:56.901 13877 13877 D LucidStatusBarView: onReceive of mLucidSysUiReceiver: com.lucid.car.settings.LucidSysUi
05-12 17:19:56.918 13877 13877 D IcrHvacPeekWindow: Gear value = 4
05-12 17:19:56.918 13877 13877 D IcrHvacPeekWindow: Gear value = 4
```

Soumya 的解释是：这只是在初始化的时候才会发生，可是我记得昨天傍晚我测的时候是都有的，今天需要再去车

```

05-12 17:19:56.919 13877 13877 D IcrHvacPeekWindow: onhvacbtn clickedfalse <=====
05-12 17:19:56.933 13877 13877 D IcrSettingsWindow: mRefreshProfilesUI() : gear=4, isUserStopping=true, mDisableLogin=true
05-12 17:19:56.967 13877 13877 D IcrHvacPeekWindow: fan speed changed017
05-12 17:19:56.969 13877 13877 D IcrHvacPeekWindow: fan speed changed018
05-12 17:19:56.971 13877 13877 D IcrHvacPeekWindow: fan speed changed033
05-12 17:19:56.972 13877 13877 D IcrHvacPeekWindow: fan speed changed034
05-12 17:19:56.974 13877 13877 D IcrHvacPeekWindow: temp changed15017
05-12 17:19:56.975 13877 13877 D IcrHvacPeekWindow: temp changed15018
05-12 17:19:56.976 13877 13877 D IcrHvacPeekWindow: temp changed15033
05-12 17:19:56.977 13877 13877 D IcrHvacPeekWindow: temp changed15034
05-12 17:19:56.981 13877 13877 D IcrSettingsWindow: vin number: 50EA1DAA5NA093482

```

- fixed: fix LucidSystemUI IcrHvacPeekWindow 中所有有物理键涉入，但是窗口没正常显示的情况，即在必要的情况下调用 showWindow() 或者 mHandler.post(mHvacBtnsRunnable)，检查四个物理键按键的回调，确认是否调用。
- 这里有个逻辑是：当用户刚进入车里，ICC 还在进行面部识别或界面都是摄像头扫描状态时，IcrHvacPeekWindow 是不会显示的（suppose 正常的显示时间是 5 秒，而扫描占去 2 秒，那么当 GearMode=drive 的时候，IcrHvacPeekWindow 会再显示 3 秒）。所以那两行的代码我也是可以不用检查 mHvacBtnEngaged 的，因为当需要显示 3 秒界面的时候，mHvacBtnEngaged = false，但是界面仍然需要正常显示（我之前并不了解这个逻辑，所以检查了，反而改错了。。。。。）
  - bug 更本质的原因是在 Adnroid 10 不同账户登录登出后，LucidSystemUI 是在每一个非系统用户下都可以正常启动，每个用户的账户下都会以 SingleInstance 的形式存在（即是一个 Instance）。Switich Profile 之后，LucidSysUiService.java 服务是可以正常启动的，具备它自己作为服务的生命周期。而且只有一个 foreground instance 在 run
  - 在安卓 10 多用户系统下，Deamon/Client 交互对多用户的影响，三个不曾重构的 Controller 在切换用户后，也仍然以安卓 9 每个应用都直接与 VHAL 进行交互，可以正常与 VHAL 建立联接。它们同样具备 VHAL 在 A10 多用户系统下继续交互的环境（系统会自己会它们创建 Context()），HvacController series works cause it sets the environment under com.lucid.car.blahblah 在一个更高的层面上已经建立好了联接，但其它 A9 Controlles 没有重构，在当前用户已经启动的上下文环境下以安卓 9 原始的方式与 VHAL 建立联接。
    - \* HvacController 重构后的源码，自己可能仍然只是部分理解了，很多部分需要再确认一下：把笔记找出来看一下!!!!
    - \* 可是其它重构前的源码，也还需要再好好地理解一下（必须），好像还没有能够很好地联接起来

### 3.2 这里 Soumya 的解释是说，只在 mBcmListener.init() 初始化的时候，我们才能收到最初的状态值，但并不是真正的回调；但是昨天我在车里测的时候，每次好像是都可以收到回调的，今天需要再去车里好好测一下，强调这一点，确认到底是什么状态的

- 314 4/15 那时总是返回 true 时的 log: 因为你现在没有 log，如果你现在能够根据这份 log 再去分析为什么 CID 会显示，可能能够帮助你理解代码，今天晚上回去搜一下!!!!

```

D Property.service: registerListener: propId=0x21201301 rate=0.0
04-14 18:53:45.970 3556 3875 D PropertyHalService: subscribeProperty propId=0x21201301, rate=0.0

13:45:00.557 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:00.557 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:00.593 3330 4330 W ActivityManager: Receiver with filter android.content.IntentFilter@2f66aeb already r

13:45:01.555 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:01.556 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:01.569 3330 5020 W ActivityManager: Receiver with filter android.content.IntentFilter@b3e0dd5 already r

```

```

13:45:03.353 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:03.353 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:03.366 3330 4222 W ActivityManager: Receiver with filter android.content.IntentFilter@75eb251 already r

13:45:03.552 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:03.552 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:03.566 3330 4222 W ActivityManager: Receiver with filter android.content.IntentFilter@c945a89 already r

13:45:04.351 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:04.351 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:04.364 3330 4222 W ActivityManager: Receiver with filter android.content.IntentFilter@456c8fd already r

13:45:04.950 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:04.950 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:04.964 3330 5020 W ActivityManager: Receiver with filter android.content.IntentFilter@acblab5 already r
04-15 13:45:05.001 3510 3510 D KeyguardClockSwitch: Updating clock: 145

13:45:05.749 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:05.749 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:05.763 3330 4222 W ActivityManager: Receiver with filter android.content.IntentFilter@6d0eb6d already r

13:45:07.047 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:07.047 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:07.061 3330 5020 W ActivityManager: Receiver with filter android.content.IntentFilter@b391b25 already r

13:45:07.546 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:07.547 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:07.559 4797 4797 I chatty : uid=1001000(com.lucid.car.lucidota) com.lucid.car.sysui expire 6 lines
04-15 13:45:07.560 3330 4222 W ActivityManager: Receiver with filter android.content.IntentFilter@81089dd already r

13:45:08.146 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:08.146 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:08.146 2799 7338 E ResolverController: No valid NAT64 prefix (100, <unspecified>/0)
04-15 13:45:08.162 3330 5020 W ActivityManager: Receiver with filter android.content.IntentFilter@6201795 already r

13:45:10.143 3278 3317 D LucidVehicleHalImpl: get_hvac_hw_switch_prop: HVAC h/w button engaged
04-15 13:45:10.143 3278 3317 D LucidVehicleHalImpl: doHalEvent prop_val.prop = 0x21201301 for signal InfoT_Body/HVA
04-15 13:45:10.156 3330 5020 W ActivityManager: Receiver with filter android.content.IntentFilter@26ca44d already r

```

- 信号数字：我应试是听这四个信号才对呀

```

private static final int INPUTS_HVAC_BTNS_ENGAGED = 0x21201301;
private static final int INPUTS_VOLUME_ROLLER = 0x21401302;
private static final int INPUTS_HVAC_BTN_DRIVER_TEMPERATURE = 0x21401303;
private static final int INPUTS_HVAC_BTN_DRIVER_FANSPEED = 0x21401304;
private static final int INPUTS_HVAC_BTN_PASSENGER_FANSPEED = 0x21401305;
private static final int INPUTS_HVAC_BTN_PASSENGER_TEMPERATURE = 0x21401306;

```

### 3.3 VHAL TEAM 说他们现在不传这个信号给我们了，我们需要自己来合并!!

- 下面是 BcmController 四个物理键的回调：下面我贴的不一定正确，真正的来源是在<https://lucidmotors.atlassian.net/wiki/spaces/INF0/pages/1369538838/CCC+Signals>

Feature	Request Signal	Status Signal	Values
Fan speed physical button (driver & passenger)		HVAC_Status/IHVAC_Key2Sts	Not_Pressed:0, Up_Single_Press:1, Up_Press_Hold:2, Down_Single_Press:3
		HVAC_Status/IHVAC_Key3Sts	Down_Press_Hold:4, TBD:5, Failure_Detected:6, Invalid:7
Temperature physical button (driver & passenger)		HVAC_Status/IHVAC_Key1Sts	Not_Pressed:0, Up_Single_Press:1, Up_Press_Hold:2, Down_Single_Press:3
		HVAC_Status/IHVAC_Key4Sts	Down_Press_Hold:4, TBD:5, Failure_Detected:6, Invalid:7

```

const string kSigHvacKey1Sts = "IHVAC_Key1Sts";
const string kSigHvacKey2Sts = "IHVAC_Key2Sts";
const string kSigHvacKey3Sts = "IHVAC_Key3Sts";
const string kSigHvacKey4Sts = "IHVAC_Key4Sts";

```

- 找点日志出来看一下这些是什么呢?
- VHAL 层面 HvacPropertyHelper.cpp 的实现逻辑

```

bool get_hvac_hw_switch_prop(SignalPathTuple signal_path, double value,
                             VehiclePropValuePool* value_pool,
                             std::vector<VehicleHal::VehiclePropValuePtr>& prop_vals) {

    bool handled = false;
    std::string sig_name = std::get<2>(signal_path);

    auto val_prop = value_pool->obtain(VehiclePropertyType::INT32);
    val_prop->timestamp = elapsedRealtimeNano();
    val_prop->areaId = 0;

    // Both of the following is for the volume scroller
    if (sig_name.find(kHvacScrollPrefix) != std::string::npos) {
        if (sig_name == kSigHvacScrollDirection) {
            mRollerState[RollerStateIndex::SCROLL_DIR] = (int) value; // 0, 1, or 2
        }
        if (sig_name == kSigHvacScrollsNumber) {
            mRollerState[RollerStateIndex::SCROLL_VAL] = (int) value;
            // invalidate non-zero value so vhal can be notified again even if
            // value hasn't changed
            if (0 != value) {
                invalidate_signal(signal_path);
            }
        }
    }

    bool roll_dir_valid = (mRollerState[RollerStateIndex::SCROLL_DIR] != 0 &&
                           mRollerState[RollerStateIndex::SCROLL_DIR] != 3);
    if (roll_dir_valid && mRollerState[RollerStateIndex::SCROLL_VAL] != 0) {
        handled = true;
        int dir = (mRollerState[RollerStateIndex::SCROLL_DIR] == RollerDirection::UP) ? 1 : -1;
        int vol = dir * mRollerState[RollerStateIndex::SCROLL_VAL];

        if (is_debuggable) {
            ALOGD("%s: Roller engaged: %d", __FUNCTION__, vol);
        }

        val_prop->prop = LucidVehicleProperty::INPUTS_VOLUME_ROLLER;
        val_prop->value.int32Values[0] = vol;
    }
    else if (sig_name.find(kHvacKeyPrefix) != std::string::npos) {
        // if any hvac hard buttons are interacted with
        if (sig_name == kSigHvacKey2Sts) {
            mHvacBtnState[PhysicalHwButton::DRV_FAN] = (int) value;
        }
        if (sig_name == kSigHvacKey1Sts) {
            mHvacBtnState[PhysicalHwButton::DRV_TEMP] = (int) value;
        }
        if (sig_name == kSigHvacKey3Sts) {
            mHvacBtnState[PhysicalHwButton::PASS_FAN] = (int) value;
        }
        if (sig_name == kSigHvacKey4Sts) {
            mHvacBtnState[PhysicalHwButton::PASS_TEMP] = (int) value;
        }
    }
    int btn_state = 0;
    for (int state : mHvacBtnState) {
        if (state > 0 && state < 5) {
            btn_state = 1;
            if (is_debuggable) {
                ALOGD("%s: HVAC h/w button engaged", __FUNCTION__);
            }
            handled = true;
            break;
        }
    }
    val_prop->prop = LucidVehicleProperty::INPUTS_HVAC_BTNS_ENGAGED;
    val_prop->value.int32Values[0] = btn_state;
}

```

```

    }
    if (handled) {
        prop_vals.push_back(std::move(val_prop));
    }
    return handled;
}

```

```

INPUTS_HVAC_BTN_DRIVER_TEMPERATURE,
INPUTS_HVAC_BTN_DRIVER_FANSPEED,
INPUTS_HVAC_BTN_PASSENGER_FANSPEED,
INPUTS_HVAC_BTN_PASSENGER_TEMPERATURE,

```

### 3.3.1 Current bugs are:

- auto time may not working, or not updated properly, which result in that the IcrHvacPeekWindow is NOT closing;
- when CID two front seats fan buttons are pressed, IcrHvacPeekWindow are showing, which should NOT.
- I may need to check some corner cases for example VolumeWindow/HvacPeekWindow switches etc.

### 3.4 现在再理一下思路：感觉之前我可能没有读懂代码，为什么 CID 会显示呢？

- 那么我如何才能知道这个回调是有物理键的涉入呢？对代码还是不够熟悉吧???
- 我们可以简单地收到温度和风扇改变的信号，但是是由于物理键才贴到回调的呢，还是因为 CID 呢？
- 那么可以检查这四个回调，当其中的某一个返回真的时候，就需要显示窗口；
- 但是这里的问题是 BcmController 没有监听这些回调，应该去 HvacController 或者是 FrontSeatHvacController 里去找这些东西，或者是需要在 BcmController 里面去实现和监听这四个信号
- to be able to identify the 4 signals for physical buttons, we will need to check FrontSeatHvacController if they are listened, or implemented by us own from BcmController the 4 signals of buttons:
- FROM CID the 2 fan runnable clicks will SHOW THE HvacPeekWindow, which is the bug to be fixed
- the other thing is the the peek window is not closing ..... in time, never close
- interfere with CarReverse some thing, the logic here I an NOT understanding, nor getting it, need to check source code or ask Soumay
- l 178 line of : 可能这就是几个 corner case：就是在扫描人人头的时候，不能出现吧。。。。。

```

private boolean shouldDelayTimerToShowWindow() {
    // Determine if car is in reverse AND parking assist apps are not open
    Log.d(TAG, "Car Reverse? " + carIsInReverse() + " QNXAppClosed? " + isQNXAppClosed());
    return !carIsInReverse() && isQNXAppClosed();
}
// Log.d(TAG, "Car Reverse? " + carIsInReverse() + " QNXAppClosed? " + isQNXAppClosed());

```

### 3.5 S2R does NOT work for RC cars. they are moving some cars here

## 4 jenny/IVIANDLIBS-314: [HVAC] Peek bar is shown when fan speed is changed from CID: 这个比较容易改

- <https://lucidmotors.atlassian.net/browse/IVIANDLIBS-314>
- 无论最初是发自 UI 的按键，还是来自物理的按键，来自两个方向的，UI 都会收到来自 HvacController 的回调
- 但是只有，最初来自车上物理键的四个键的按键，IcrHvacPeekWindow 才会出现，其它不应该出来
- 因为 IcrHvacPeekWindow 目前显示不出来，暂时无法测试，暂且 blocked 处理

## 5 Contollers

### 5.1 IcrHvacPeekWindow: 我觉得这里面很多代码我没有看懂，不只要想 showWindow(), 还需要去想 hideWindow() 的逻辑

```
public void showWindow() {
    Log.d(TAG, " Show ICRhvacVolumeWIndow");
    initTheme();
    if (mShowWindow) { // 如果已经显示了
        initSystemPrefs();
        updateAutoCloseTimer(); // 则：更新 剩余还将显示时间
        return;
    }
    mShowWindow = true;
    mHandler.post(mUpdateHvacUI); // 定义在下面 =====>>>>>>>>
    // Attach window.
    WindowManager.LayoutParams lp = new WindowManager.LayoutParams(
        ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.MATCH_PARENT,
        WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY, // WindowManager.LayoutParams.TYPE_DISPLAY_OVERLAY,
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE
        | WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS
        & ~WindowManager.LayoutParams.FLAG_WATCH_OUTSIDE_TOUCH,
        PixelFormat.TRANSLUCENT);
    lp.gravity = Gravity.BOTTOM;
    lp.x = 0;
    lp.y = 0;
    lp.windowAnimations = R.style.hvac_pop_up;
    mWindowManager.addView(mIcrPeakWindow, lp);
    initSystemPrefs(); // 这里就是 LucidCarSettings ==> Display ==> Units ==> Temperature: in F or C, 有个广播需要监听一下
    updateAutoCloseTimer(); // 更新 剩余还将显示时间
}

private final Runnable mUpdateHvacUI = new Runnable() {
    @Override
    public void run() {
        mHandler.removeCallbacks(mUpdateHvacUI);
        synchronized (mHvacController) {
            int dTempCelsiusValue = mHvacController.getTemperature(HvacController.ZONE_FRONT_LEFT);
            int pTempCelsiusValue = mHvacController.getTemperature(HvacController.ZONE_FRONT_RIGHT);
            int dFanSpeed = mapFanSpeed(0, 100, 0, 10,
                mHvacController.getFanSpeed(HvacController.ZONE_FRONT_LEFT));
            int pFanSpeed = mapFanSpeed(0, 100, 0, 10,
                mHvacController.getFanSpeed(HvacController.ZONE_FRONT_RIGHT));
            updateTempText(mDriverTemp, dTempCelsiusValue);
            updateTempText(mPassengerTemp, pTempCelsiusValue);
            if (dFanSpeed == 0) {
                mDriverFanSpeed.setText(mContext.getString(R.string.general_off));
                mDriverFanImg.setVisibility(View.GONE);
                mDriverTemp.setVisibility(View.GONE);
            } else {
                mDriverTemp.setVisibility(View.VISIBLE);
            }
        }
    }
}
```

```

        mDriverFanImg.setVisibility(View.VISIBLE);
        mDriverFanSpeed.setText(String.valueOf(dFanSpeed));
    }
    if (pFanSpeed == 0) {
        mPassengerFanImg.setVisibility(View.GONE);
        mPassengerFanSpeed.setText(mContext.getResources().getString(R.string.general_off));
        mPassengerTemp.setVisibility(View.GONE);
    } else {
        mPassengerTemp.setVisibility(View.VISIBLE);
        mPassengerFanImg.setVisibility(View.VISIBLE);
        mPassengerFanSpeed.setText(String.valueOf(pFanSpeed));
    }
}
showWindow(); //
}
};

```

## 5.2 GearController.GearListener

```

private final GearController.GearListener mGearListener =
    new GearController.GearListener() {
        @Override
        public void onVhalConnectedW() { }
        @Override
        public void onVhalDisconnectedW() { }
        @Override
        public void onUpdateGearStateChangedW(Integer state) {
            // 判断标准：是在 Drive or Reverse 吗，车发动了吗？可以显示那两种窗口了吗？
            if (mShowWindow && state != GearController.GEAR_REVERSE
                && state != GearController.GEAR_IS_NOT_VALID) {
                showWindow();
            }
        }
    };

```

## 5.3 ParkAssistController.ParkAssistListener

```

private final ParkAssistController.ParkAssistListener mParkAssistListener =
    new ParkAssistController.ParkAssistListener() {
        @Override
        public void onVhalConnectedW() { }
        @Override
        public void onVhalDisconnectedW() { }
        @Override
        public void onReceiveCCCAppStatusW(Integer cccQNXAppStatus) {
            Log.d(TAG, "CCCQNXAppStatus value = " + cccQNXAppStatus);
            // 这里应该是用来判断：
            // 是否位于刚扫描到有用户坐入，或正在进行摄像头扫描，因为这种情况下，IcrHvacPeekWindow & IcrVolumWindow 会在 Drive 时有可能才再显
            if (mShowWindow && cccQNXAppStatus == ParkAssistController.CCC_QNX_ALL_APPS_CLOSED) {
                Log.d(TAG, "All QNX apps closed");
                showWindow(); // 显示 3 秒 = 5 秒 - 2 秒
            }
        }
    };

```

## 5.4 BcmController.BcmListener

- 看起来现在 BcmController 并不能收到 vhal 的信号，把这所有的回调（四个物理键的回调）都检查一遍
- 这里的问题可能是：BcmController 和其它两个不曾重构的 Controllers 并不能像已经重构过的 HvacController 那样能够从 Vhal 接收到信号？可以的!!!

```

private final BcmController.BcmListener mBcmListener = new BcmController.BcmListener() {
    @Override
    public void onVhalConnectedW() { }
    @Override
    public void onVhalDisconnectedW() { }
    // @Override // 现在不再能够收到这个信号了

```

```

// public void onEngagedHvacBtnsW(boolean bEngaged) { // Soumya: 车上四个物理键被按的回调
//     Log.d(TAG, "onhvacbtn clicked" + bEngaged); // 这个从来不曾打印, 也就是 vhal 的信号没能够传上来!!!!!!!!!!!!!!
//     mBtnEngaged = bEngaged;
//     if (mShowWindow) {
//         updateAutoCloseTimer();
//     }
//     mHandler.post(mHvacBtnsRunnable); // showWindow()
// }

@Override public void onUpdateVolumeRollerW(int adjustValue) { // 一个特定键被按的回调
    if (mShowWindow) // 这里也就处理了两种 PeekWindow 交互的一个方向: 当正在显示, 按了音量, 就隐藏, 可能小细节还需要再另外
        hideWindow();
}

@Override public void onUpdateHvacBtnDriverTemperature(int state) { // 感觉这里的逻辑还是有些混乱呀, 风扇与温度之间到底
    mBtnEngaged = true;
    mHandler.post(mHvacBtnsRunnable);
    if ( state >= 1 && state <= 4 && mHvacController != null &&
        mHvacController.getFanSpeed( HvacController.ZONE_FRONT_LEFT ) == 0 ) {
        int mappedFanSpeed = mapFanSpeed( 0, 10, 0, 100, 1 );
        mHvacController.setFanSpeed( HvacController.ZONE_FRONT_LEFT, mappedFanSpeed );
    }
}

@Override public void onUpdateHvacBtnPassengerTemperature(int state) {
    mBtnEngaged = true;
    mHandler.post(mHvacBtnsRunnable);
    if ( state >= 1 && state <= 4 && mHvacController != null &&
        mHvacController.getFanSpeed( HvacController.ZONE_FRONT_RIGHT ) == 0 ) {
        int mappedFanSpeed = mapFanSpeed( 0, 10, 0, 100, 1 );
        mHvacController.setFanSpeed( HvacController.ZONE_FRONT_RIGHT, mappedFanSpeed );
    }
}

@Override public void onUpdateHvacBtnDriverFanSpeed(int state) {
    mBtnEngaged = true;
    mHandler.post(mHvacBtnsRunnable);
}

@Override public void onUpdateHvacBtnPassengerFanSpeed(int state) {
    mBtnEngaged = true;
    mHandler.post(mHvacBtnsRunnable);
}
}

```

## 5.5 HvacController.HvacListener

```

private final HvacController.HvacListener mHvacListener = new HvacController.HvacListener() {
    @Override
    public void onVhalConnectedW() {
        Log.d(TAG, "Hvac Vhal connected");
        super.onVhalConnectedW();
    }
    @Override
    public void onVhalDisconnectedW() {
        Log.d(TAG, "Hvac Vhal disconnected");
        super.onVhalDisconnectedW();
    }
    @Override // 2 sources of reasons: UI or VHALL
    public void onUpdateFanSpeedW(int zone, int speed) { // 这是从另一个信号来的风扇实际值, 从下往上传来的, 但是未必来自物理键的
        Log.d(TAG, "fan speed changed" + speed + zone);
        if(zone == HvacController.ZONE_FRONT_LEFT) {
            mDriverSpeed = mapFanSpeed(0,100,0,10,speed);
        } else if(zone == HvacController.ZONE_FRONT_RIGHT) {
            mPassengerSpeed = mapFanSpeed(0,100,0,10,speed);
        }
        mFanZone = zone;
        if (mShowWindow) {
            updateAutoCloseTimer();
        }
        mHandler.post(mFanUIRunnable); // 没有 call showWindow()
    }
    // 这下面一行需要 (根据 BcmController.BcmListener 的四个物理键回调) 作判断: 有物理键吗, 有才显示窗口
    // 还是觉得这一行有问题
    mHandler.post(mHvacBtnsRunnable); // 延迟: 需要想一下, 如果再延迟, 那么结果会不会 不一样????
    // showWindow(), 2 sources of reasons, but only VHALL signaled physical buttons should showWindow()
}
@Override // Soumya said this callback does NOT call showWindow() is a bug to be fixed
public void onUpdateTemperatureW(int zone, int temp) { // 2 sources of reasons: UI or VHALL
    Log.d(TAG, "temp changed" + temp + zone);
    mTempZone = zone;
}

```



```

    if (mShowWindow) {
        Log.d(TAG, "update time in temp");
        updateAutoCloseTimer();
    }
    mHandler.post(new Runnable() {
        @Override
        public void run() {
            if (zone == HvacController.ZONE_FRONT_LEFT && tempInCelsius != temp) {
                mDriverTemp.setTextColor(mContext.getColor(R.color.primaryWhite));
                mPassengerFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerTemp.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerFanImg.setImageResource(R.drawable.fan_img);
                mDriverFanImg.setImageResource(R.drawable.fan_img);
                mPassFanOff.setTextColor(mContext.getColor(R.color.textDim));
            } else if (zone == HvacController.ZONE_FRONT_RIGHT && tempInCelsius != temp) {
                mPassengerTemp.setTextColor(mContext.getColor(R.color.primaryWhite));
                mPassengerFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverTemp.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerFanImg.setImageResource(R.drawable.fan_img);
                mDriverFanImg.setImageResource(R.drawable.fan_img);
            }
        }
    });
    tempInCelsius = temp;
    mHandler.post(mTempUIRunnable);
    // mHandler.post(mHvacBtnsRunnable); // showWindow() 这里被自己加错了.....
}
@Override
public void onUpdateSyncZonesW(boolean isSyncOn) {
    mIsSyncOn = isSyncOn;
}
};
private final Runnable mHvacBtnsRunnable = new Runnable() {
    @Override public void run() {
        mHandler.removeCallbacks(mHvacBtnsRunnable);
        synchronized (mBtnEngaged) {
            // 判断标准：四个物理键之一有涉入 mBtnEngaged = true, 需要 BcmController 传四个回调回来
            if (mBtnEngaged) {
                Log.d(TAG, "onhvacbtn clicked" + mBtnEngaged);
                showWindow();
            }
        }
    }
};
private final Runnable mTempUIRunnable = new Runnable() {
    @Override
    public void run() {
        mHandler.removeCallbacks(mTempUIRunnable);
        Log.d(TAG, "Entered temp runnable");
        synchronized (mTempZone) {
            if (mTempZone == HvacController.ZONE_FRONT_LEFT && tempFromCallback != mCelsiusTemp) { // F
                mDriverTemp.setTextColor(mContext.getColor(R.color.primaryWhite));
                mPassengerFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerTemp.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerFanImg.setImageResource(R.drawable.fan_img);
                mDriverFanImg.setImageResource(R.drawable.fan_img);
                mPassFanOff.setTextColor(mContext.getColor(R.color.textDim));
            } else if (mTempZone == HvacController.ZONE_FRONT_RIGHT && tempFromCallback != mCelsiusTemp) { // F
                mPassengerTemp.setTextColor(mContext.getColor(R.color.primaryWhite));
                mPassengerFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverFanSpeed.setTextColor(mContext.getColor(R.color.textDim));
                mDriverTemp.setTextColor(mContext.getColor(R.color.textDim));
                mPassengerFanImg.setImageResource(R.drawable.fan_img);
                mDriverFanImg.setImageResource(R.drawable.fan_img);
            }
        }
        Log.d(TAG, "temp fan synchronized");
        switch (mTempZone) { // C
            case HvacController.ZONE_FRONT_LEFT:
                Log.d(TAG, "driver temp in Celsius" + tempInCelsius);
                updateTempText(mDriverTemp, tempInCelsius);
                break;
            case HvacController.ZONE_FRONT_RIGHT:

```

```
Log.d(TAG, "passenger temp in Celsius" + tempInCelsius);  
updateTempText(mPassengerTemp, tempInCelsius);  
break;  
default:  
    Log.d(TAG, "Unknown zone id: " + mTempZone);  
    if(mIsSyncOn) { // 把两个座位的温度同步成一样的  
        updateTempText(mPassengerTemp, tempInCelsius);  
        updateTempText(mDriverTemp, tempInCelsius);  
    }  
}  
}  
}  
};
```

## 6 jenny/IVIANDLIBS-342: Peek bar does not show up if we change temp/fan using hard buttons

- <https://lucidmotors.atlassian.net/browse/IVIANDLIBS-342>
- 这个应该有几个方面的原因：这里是考到了逻辑的最死角了吗???????? VHAL team 所设置的物理按钮的响应时间极短??????
  - fix LucidSystemUI IcrHvacPeekWindow 中所有有物理☑️涉入，但是窗口没有显示的情况，即在必要的情况下调用 showWindow() 或者 mHandler.post(mHvacBtnsRunnable); (这种情况出现的可能过时了。。。)

## 6.1 Log snapshots 以及可能有的问题

### 6.1.1 some kind of apk build path specification error, but does NOT matter that much

```

05-03 17:02:37.476 4404 4404 E LoadedApk: java.lang.ClassNotFoundException: Didn't find class "com.lucid.car.sysui.lucid"
dexPathList[[zip file "/system/priv-app/LucidSystemUI/LucidSystemUI.apk"],
nativeLibraryDirectories=[/system/priv-app/LucidSystemUI/lib/arm64, /system/priv-app/LucidSystemUI/LucidSystemUI.apk!/lib/a
05-03 17:02:37.476 4404 4404 E LoadedApk: ^Iat dalvik.system.BaseDexClassLoader.findClass(BaseDexClassLoader.java:196)
05-03 17:02:37.476 4404 4404 E LoadedApk: ^Iat java.lang.ClassLoader.loadClass(ClassLoader.java:379)
05-03 17:02:37.476 4404 4404 E LoadedApk: ^Iat java.lang.ClassLoader.loadClass(ClassLoader.java:312)
05-03 17:02:37.476 4404 4404 E LoadedApk: ^Iat android.app.LoadedApk.createAppFactory(LoadedApk.java:256)
05-03 17:02:37.476 4404 4404 E LoadedApk: ^Iat android.app.LoadedApk.createOrUpdateClassLoaderLocked(LoadedApk.java:855)

```

### 6.1.2 Broken Notification Channel: foreground service LucidSysUIService is NOT launched properly due to broken channel, but it does NOT matter

```

05-03 17:02:37.474 3324 3341 W ActivityManager: Unable to start service Intent { cmp=com.lucidmotors.lucidcalendarsync/.s
05-03 17:02:37.476 4404 E LoadedApk: Unable to instantiate appComponentFactory
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.LoadedApk.getClassLoader(LoadedApk.java:950)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.LoadedApk.getResources(LoadedApk.java:1188)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ContextImpl.createAppContext(ContextImpl.java:2462)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ContextImpl.createAppContext(ContextImpl.java:2454)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ActivityThread.handleBindApplication(ActivityThread.java:6343)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ActivityThread.access$1300(ActivityThread.java:219)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ActivityThread$H.handleMessage(ActivityThread.java:1859)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.os.Handler.dispatchMessage(Handler.java:107)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.os.Looper.loop(Looper.java:214)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat android.app.ActivityThread.main(ActivityThread.java:7356)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat java.lang.reflect.Method.invoke(Native Method)
05-03 17:02:37.476 4404 E LoadedApk: ^Iat com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.j
05-03 17:02:37.476 4404 E LoadedApk: ^Iat com.android.internal.os.ZygoteInit.main(ZygoteInit.java:930)
05-03 17:02:37.531 3324 3355 V ActivityManager: Attempted to start a foreground service (com.lucid.car.sysui/.LucidSysUiS
with a broken notification (no icon: Notification(channel=LucidSysUIService
pri=0 contentView=null vibrate=null sound=null defaults=0x0 flags=0x40 color=0x00000000 vis=PRIVATE))
05-03 17:02:37.558 3324 3692 D DynamicSystemService: GsiService is not ready, wait for 64ms

```

### 6.1.3 在 framework 层, 有 SystemUI 应用 ?? 它应该是负责 broken Notification Channels, 可以问一下 Soumya

- [http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid\\_dash5dv\\_10r40/frameworks/base/packages/SystemUI/tests/src/com/android/systemui/](http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid_dash5dv_10r40/frameworks/base/packages/SystemUI/tests/src/com/android/systemui/)

### 6.1.4 LifecycleOwner 相关的: 需要自己再深入地理解一下

- 来自 framework 的 WifiDetailPreferenceController [http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid\\_dash5dv\\_10r40/packages/apps/Settings/src/com/android/settings/wifi/details/WifiDetailPreferenceController.java?r=40e7c05e](http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid_dash5dv_10r40/packages/apps/Settings/src/com/android/settings/wifi/details/WifiDetailPreferenceController.java?r=40e7c05e)
- 来自 CarRadioApp 的系统服务 com.android.car.radio.service.RadioAppService [http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid\\_dash5dv\\_10r40/packages/apps/Car/Radio/src/com/android/car/radio/service/RadioAppService.java](http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid_dash5dv_10r40/packages/apps/Car/Radio/src/com/android/car/radio/service/RadioAppService.java)
- SystemUI lifecycle [http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid\\_dash5dv\\_10r40/frameworks/base/packages/SystemUI/src/com/android/systemui/util/SysuiLifecycle.java?r=8e4e4cb9](http://hqdevbld01.corp.lucid.lcl:8888/source/xref/lucid_dash5dv_10r40/frameworks/base/packages/SystemUI/src/com/android/systemui/util/SysuiLifecycle.java?r=8e4e4cb9)
- 参考下面的源码, 我可以做点儿什么呢?

```
package com.android.systemui.util;

import static androidx.lifecycle.Lifecycle.State.DESTROYED;
import static androidx.lifecycle.Lifecycle.State.RESUMED;
import android.view.View;
import android.view.View.OnAttachStateChangeListener;
import androidx.annotation.NonNull;
import androidx.lifecycle.Lifecycle;
import androidx.lifecycle.LifecycleOwner;
import androidx.lifecycle.LifecycleRegistry;

// * Tools for generating lifecycle from sysui objects.
public class SysuiLifecycle {
    private SysuiLifecycle() { }
    /**
     * Get a lifecycle that will be put into the resumed state when the view is attached
     * and goes to the destroyed state when the view is detached.
     */
    public static LifecycleOwner viewAttachLifecycle(View v) {
        return new ViewLifecycle(v);
    }
    private static class ViewLifecycle implements LifecycleOwner, OnAttachStateChangeListener {
        private final LifecycleRegistry mLifecycle = new LifecycleRegistry(this);
        ViewLifecycle(View v) {
            v.addOnAttachStateChangeListener(this);
        }
        @NonNull @Override
        public Lifecycle getLifecycle() {
            return mLifecycle;
        }
        @Override public void onViewAttachedToWindow(View v) {
            mLifecycle.markState(RESUMED);
        }
        @Override
        public void onViewDetachedFromWindow(View v) {
            mLifecycle.markState(DESTROYED);
        }
    }
}
```

**6.2 Error message 1: AndroidManifest.xml Proguard + memory leak ?** 应用是某个地方的路径写得不对，但好歹它的关系不大，LoadedApk: java.lang.ClassNotFoundException: Didn't find class "com.lucid.car.sysui" on path: DexPathList[[zip file "/system/priv-app/LucidSystemUI/LucidSystemUI.apk", /system/priv-app/LucidSystemUI/LucidSystemUI.apk, /system/priv-app/LucidSystemUI/LucidSystemUI.apk, /system/lib64, /product/lib64, /system/lib64, /product/lib64]]

**6.3 crash at system user 0 at very beginning, if this is the case, the log is meaningless**

```
05-03 14:51:26.684 2780 2780 D ZygoteInitTiming_lazy: BeginPreload took to complete: 11ms
05-03 14:51:26.735 3324 3324 I SettingsState: No settings state /data/system/users/0/settings_config.xml
05-03 14:51:26.737 3324 3324 I SettingsState: directory info for directory/file /data/system/users/0/settings_config.xml
05-03 14:51:26.737 3324 3324 I SettingsState: java.lang.Exception
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsState.logSettingsDirectoryInfo
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsState.readStateSyncLocked(Setting
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsState.<init>(SettingsState.java:
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider$SettingsRegistry.ensu
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider$SettingsRegistry.peak
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider$SettingsRegistry.getS
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider$SettingsRegistry.sync
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider$SettingsRegistry.<ini
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsProvider.onCreate(SettingsProv
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat android.content.ContentProvider.attachInfo(ContentProvider.java:2092)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat android.content.ContentProvider.attachInfo(ContentProvider.java:2066)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat android.app.ActivityThread.installProvider(ActivityThread.java:6983)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat android.app.ActivityThread.installContentProviders(ActivityThread.java:
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat android.app.ActivityThread.installSystemProviders(ActivityThread.java:
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.server.am.ActivityManagerService.installSystemProviders(A
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.server.SystemServer.startOtherServices(SystemServer.java:
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.server.SystemServer.run(SystemServer.java:513)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.server.SystemServer.main(SystemServer.java:350)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat java.lang.reflect.Method.invoke(Native Method)
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeIn
05-03 14:51:26.737 3324 3324 I SettingsState: ^Iat com.android.internal.os.ZygoteInit.main(ZygoteInit.java:908)
05-03 14:51:26.737 3324 3324 I SettingsState: ancestor directory /data/system/users/0 exists
05-03 14:51:26.737 3324 3324 I SettingsState: ancestor directory /data/system/users/0 permissions: r: true w: true x: tru
05-03 14:51:26.738 3324 3324 I SettingsState: ancestor's parent directory /data/system/users permissions: r: true w: true
05-03 14:51:26.749 3324 3324 I SettingsState: No settings state /data/system/users/0/settings_ssaid.xml
05-03 14:51:26.750 3324 3324 I SettingsState: directory info for directory/file /data/system/users/0/settings_ssaid.xml w
05-03 14:51:26.750 3324 3324 I SettingsState: java.lang.Exception
05-03 14:51:26.750 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsState.logSettingsDirectoryInfo
05-03 14:51:26.750 3324 3324 I SettingsState: ^Iat com.android.providers.settings.SettingsState.readStateSyncLocked(Setting
```

## 7 logs from today to support all the ideas

- 大致明白了这些方法，改天需要或者有什么不懂的地方的时候就先从自己的手机上抓一抓，实在不行的就再上 bench 上去抓一抓，再不行就上车上上去抓一抓

**7.1 收听 BOOT\_COMPLETED 的几个事件与服务与先后顺序: LucidSystemService, LucidDmsService, com.lucid.car.carsettings.BootReceiver, (InjectBaseVersionS2110: onReceive() : intent=android.intent.action.BOOT\_COMPLETED), com.lucid.car.sysui.pinToDrive.PinToDriveService**

```
I AlarmClock: AlarmInitReceiver android.intent.action.LOCKED_BOOT_COMPLETED
I ActivityManager: Posting BOOT_COMPLETED user #10
D LucidSystemService: onReceive() ACTION_USER_UNLOCKED
D LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
D LucidSystemService: (intent.hasExtra(EXTRA_USER_HANDLE)): true
D LucidSystemService: START LucidCarLauncher
D LucidSystemService: START LucidSysUiService.java
D LucidSystemService: onStartCommand :isFteRequiredfalse
```

```

D ONSAutoboot: Received android.intent.action.LOCKED_BOOT_COMPLETED
D LucidDmsService: onReceive(): android.intent.action.BOOT_COMPLETED
D AlertReceiver: onReceive: a=android.intent.action.BOOT_COMPLETED Intent { act=android.intent.action.BOOT_COMPLETED flg=0x
I AlarmClock: AlarmInitReceiver android.intent.action.BOOT_COMPLETED
D MapApplication: received the boot complete event
E com.lucid.car.carsettings.BootReceiver: BOOT COMPLETED CALLED FROM CAR SETTINGS
D InjectBaseVersionS2110: onReceive() : intent=android.intent.action.BOOT_COMPLETED
D com.lucid.car.sysui.pinToDrive.PinToDriveService: onBoot completed...
D LucidSysUIService: Boot up completed android.intent.action.BOOT_COMPLETED
D LucidSysUIService: creating alarm manager service
I ActivityManager: Finished processing BOOT_COMPLETED for u10
D LucidSystemService: onReceive() ACTION_USER_UNLOCKED
D LucidSystemService: onReceive: intent android.intent.action.USER_STOPPED

```

## 7.2 在我想要切换用户的瞬间发生的

```

05-12 08:11:42.354 7902 7902 D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
05-12 08:11:43.259 7902 7902 D LucidSysUIService: onStartCommand
05-12 08:11:43.259 7902 7902 D LucidSysUIService: (ACTION_LAUNCH_CID.equals(action)): false
05-12 08:11:43.259 7902 7902 D LucidSysUIService: tapOnStatusForHomeLink ACTION_LUCID_ICR_SETTINGS_BARtrue
05-12 08:11:43.378 7902 7902 E LucidSysUIService: OTA_REMIND_LATER_TIME
05-12 08:11:43.379 7902 7902 W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.showOTANotifications(LucidSysUISer
05-12 08:11:43.379 7902 W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.onStartCommand(LucidSysUIService.j
05-12 08:11:43.477 7902 7902 D LucidSysUIService: mWifiEnableReceiver :com.lucid.carlib.ctrls.ACTION_WIFI_ENABLE
05-12 08:11:43.477 7902 7902 D LucidSysUIService: wifiEnable: false
05-12 08:11:43.478 7902 7902 D LucidSysUIService: onReceive: com.lucid.carlib.ctrls.ACTION_WIFI_ENABLE wifiEnable: false
05-12 08:11:44.354 7902 7902 D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0

```

## 7.3 ACTION\_USER\_XXXX related: current user (MESSAGE/ACTION\_)USER\_U then prev user (ACTION\_)USER\_STOPPED

```

D LucidSystemService: onReceive() ACTION_USER_UNLOCKED
D LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
D LucidSystemService: (intent.hasExtra(EXTRA_USER_HANDLE)): true
D LucidSystemService: START LucidCarLauncher
D LucidDmsService: On received HMI action: android.intent.action.USER_UNLOCKED
D LucidSystemService: START LucidSysUIService.java
D LucidSystemService: onStartCommand :isFteRequiredfalse
D BluetoothManagerService: MESSAGE_USER_UNLOCKED
D LucidDmsService: On received HMI action: android.intent.action.USER_STOPPED
D LucidSystemService: onReceive() ACTION_USER_UNLOCKED
D LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
D LucidSystemService: (intent.hasExtra(EXTRA_USER_HANDLE)): true
D LucidSystemService: START LucidCarLauncher
D LucidDmsService: On received HMI action: android.intent.action.USER_UNLOCKED
D LucidSystemService: START LucidSysUIService.java
D LucidSystemService: onStartCommand :isFteRequiredfalse
E ActivityManager: Sending non-protected broadcast com.lucid.home.ProfileSwitcher.ACTION_USER_LOGGED_IN from system 12202:c
E ActivityManager: Sending non-protected broadcast com.lucid.home.ProfileSwitcher.ACTION_USER_LOGGED_IN from system 12202:c
D BluetoothManagerService: MESSAGE_USER_UNLOCKED 可能是它的最后一次广播吧
D LucidDmsService: On received HMI action: android.intent.action.USER_STOPPED
D LucidSystemService: onReceive() ACTION_USER_UNLOCKED
D LucidSystemService: onReceive: intent android.intent.action.USER_STOPPED

```

## 7.4 all onCreate() related: 从 adb reboot 开始抓的: “onCreate” 可能没有抓全

```

D MainActivity: onCreate Home app
D MediaFragment: onCreateView
D BluetoothAdapterService: onCreate()
D HereNavigationNotificationService: onCreate()
D MainActivity: onCreate
D MediaFragment: onCreateView
D MediaFragment: onCreateView
D LucidCarServiceApp: onCreate...
D LucidCarServiceApp: startLucidSystemServices()
D CarLatinIME: onCreate
D LucidCarServiceApp: starting PowerManagementService...
D LucidCarServiceApp: starting LucidTimeZoneService...

```

```

D CarLatinIME: onCreate
D LucidTimeZoneService: onCreate timeZone: America/Los_Angeles
D HfpClientConnService: onCreate
D MainActivity: onCreate
D MediaFragment: onCreateView
D HereNavigationNotificationService: onCreate()
D CarLatinIME: onCreate
D LucidCarServiceApp: OnBootReceiver.onReceive()
D LucidCarServiceApp: getEcuVersions - exec udscat
D LucidCarServiceApp: getEcuVersions - java.io.IOException: Cannot run program "udscat": error=13, Permission denied
D LucidCarServiceApp: getEcuVersions - exec udscat done
D MainActivity: onCreate
D MediaFragment: onCreateView
D HereNavigationNotificationService: onCreate()
D CarLatinIME: onCreate
D MainActivity: onCreate
D Cluster.Service: onCreate
D MediaFragment: onCreateView
D MediaFragment: onCreateView
D CarLatinIME: onCreate
D LucidSystemService: START LucidSysUiService.java
I ActivityManager: Start proc 4675:com.lucid.car.sysui/u11s1000 for service {com.lucid.car.sysui/com.lucid.car.sysui.LucidS
D CM.MessengerService: onCreate
D LucidLauncherActivity: onCreate
D QuickControlsActivity: onCreate
D FrontSeatHvacController: [FrontViewFragment.onCreateView():240] Daemon Instance is set to true
D LucidSysUiService: onCreate Called
D LucidSysUiService: creating alarm manager service
D CarNotificationListener: onNotificationPosted: StatusBarNotification(pkg=com.lucid.car.sysui user=UserHandle{11} id=1001
D com.lucid.car.carsettings.CarSettingsApplication: onCreate
D IcrSettingsWindow: cursor dataandroid.content.ContentResolver$CursorWrapperInner@d188b45
D IcrSettingsWindow: homeLinkList Size0
D com.lucid.car.carsettings.home.HomeActivity: onCreate
D IcrSettingsWindow: init bluetooth controller in bluetooth tab
D IcrSettingsWindow: WifiTab: setVisibility false
D LucidSysUiService: settings system property done
D LucidSysUiService: onStartCommand
D LucidSysUiService: (ACTION_LAUNCH_CID.equals(action)): true
D LucidSysUiService: (!FteUtil.isFteRequired(context)): true
D LucidSysUiService: send stop animation to profiles
E LucidSysUiService: OTA_REMIND_LATER_TIME
D IcrHvacPeekWindow: Park Assist Controller connected
D IcrHvacPeekWindow: Gear Controller VHAL connected
D IcrHvacPeekWindow: Bcm Controller VHAL connected
D IcrHvacPeekWindow: Hvac Vhal connected

```

## • 系统层面整个系统的启动顺序

```

05-13 18:06:44.202 1377 1377 I SettingsState: ^^Iat com.android.providers.settings.SettingsProvider.onCreate(Setting
05-13 18:06:44.209 1377 1377 I SettingsState: ^^Iat com.android.providers.settings.SettingsProvider.onCreate(Setting
05-13 18:06:44.220 1377 1377 I SettingsState: ^^Iat com.android.providers.settings.SettingsProvider.onCreate(Setting
05-13 18:06:44.682 1377 1814 D WifiActiveModeWarden: Switching from WifiDisabledState to WifiDisabled
05-13 18:06:44.867 1377 1377 D VoiceInteractionManagerService: switchImplementation(0) took to complete: 0ms
05-13 18:06:44.876 1377 1377 I CarServiceHelper: Switching to user 10 on boot
05-13 18:06:45.059 1961 1961 I CAR.SERVICE: Service onCreate
05-13 18:06:45.140 1961 2092 I CAR.POWER: User switch disallowed while booting
05-13 18:06:45.152 1961 2092 D CAR.POWER: getTargetUserId(): current=10, target=UserInfo[id=10, name=Primary Driver
05-13 18:06:45.153 1961 2092 D CAR.POWER: Not switching to 10 because it's not allowed
05-13 18:06:45.357 2146 2146 I AlarmClock: AlarmInitReceiver android.intent.action.LOCKED_BOOT_COMPLETED
05-13 18:06:45.376 2185 2185 D CarUserService: onCreate()
05-13 18:06:45.388 1377 1433 D VoiceInteractionManagerService: switchImplementation(0) took to complete: 4ms
05-13 18:06:45.435 1377 1433 I ActivityManager: Posting BOOT_COMPLETED user #0
05-13 18:06:45.543 2266 2266 D CarrierProvider: onCreate
05-13 18:06:45.552 2266 2266 D CarrierIdProvider: onCreate
05-13 18:06:45.556 2359 2359 W ContextImpl: Calling a method in the system process without a qualified user: andro
05-13 18:06:45.558 2359 2359 D ProfileDataService: onStartCommand(): starting service with flags= 0, startId= 1
05-13 18:06:45.558 2359 2359 D ProfileDataService: onStartCommand(): initializing repository...
05-13 18:06:45.560 2359 2359 D ProfileDataService: onStartCommand(): starting random number generator for IM ALIVE
05-13 18:06:45.560 2359 2359 D ProfileDataService: onStartCommand(): service has been started
05-13 18:06:45.619 2266 2266 D ONSAutoboot: Received android.intent.action.LOCKED_BOOT_COMPLETED

```

```
05-13 18:06:45.722 2488 2488 D LucidDmsService: onReceive(): android.intent.action.BOOT_COMPLETED
05-13 18:06:45.997 1377 1433 I SystemServiceManager: Calling switchUser u10
```

```
05-13 18:06:46.112 1377 1433 D BluetoothManagerService: Bluetooth boot completed
05-13 18:06:46.163 1988 1988 D KeyguardClockSwitch: Updating clock: 606
```

```

05-13 18:06:46.176 2185 2185 D CarUserService: onCreate()
05-13 18:06:46.177 1377 1415 D VoiceInteractionManagerService: switchImplementation(10) took to complete: 1ms
05-13 18:06:46.225 2699 2699 D LucidCarServiceApp: onCreate...
05-13 18:06:46.225 2699 2699 D LucidCarServiceApp: onCreate...
05-13 18:06:46.225 2699 2699 D LucidCarServiceApp: startLucidSystemService()
05-13 18:06:46.232 2699 2699 W ContextImpl: Calling a method in the system process without a qualified user: android.os
05-13 18:06:46.241 2764 2764 I SecureElementService: main onCreate
05-13 18:06:46.255 2699 2699 W ContextImpl: Calling a method in the system process without a qualified user: android
05-13 18:06:46.278 2784 2784 D MainActivity: onCreate
05-13 18:06:46.278 2699 2699 D LucidSystemService: LucidSystemService is created.
05-13 18:06:46.335 2784 2784 D MediaFragment: onCreateView
05-13 18:06:46.336 2699 2699 D LucidTimeZoneService: onCreate timeZone: America/Los_Angeles
05-13 18:06:46.342 2804 2804 D BluetoothAdapterService: onCreate()
05-13 18:06:46.375 1377 1377 I Telecom : MissedCallNotifierImpl: reloadFromDatabase: Boot not yet complete -- call
05-13 18:06:46.404 2784 2784 D HereNavigationNotificationService: onCreate()
05-13 18:06:46.464 1377 1377 W FingerprintService: User switched while performing cleanup
05-13 18:06:46.615 2784 2784 D MainActivity: onCreate
05-13 18:06:46.643 2784 2784 D MediaFragment: onCreateView
05-13 18:06:46.657 2784 2784 D MediaFragment: onCreateView
05-13 18:06:46.695 3053 3053 D CarLatinIME: onCreate
05-13 18:06:46.783 3053 3053 D CarLatinIME: onCreate
05-13 18:06:46.831 1377 2296 W ActivityManager: User switch slowed down by observer #3 KeyguardUpdateMonitor: result
05-13 18:06:46.831 1377 1433 D ActivityManager: Continue user switch oldUser #0, newUser #10
05-13 18:06:46.977 1377 1433 W BroadcastQueue: Skipping deliver [background] BroadcastRecord{898e2e2 u0 android.int
05-13 18:06:46.994 2338 2338 E AndroidRuntime: ^Iat com.android.service.ims.presence.EABService.onStartCommand(EAB
05-13 18:06:47.146 3237 3237 E AndroidRuntime: ^Iat com.android.service.ims.presence.EABService.onStartCommand(EAB
05-13 18:06:47.226 3293 3293 E AndroidRuntime: ^Iat com.android.service.ims.presence.EABService.onStartCommand(EAB
05-13 18:06:47.656 3214 3214 D KeyguardClockSwitch: Updating clock: 606
05-13 18:06:47.952 3420 3420 D AlertReceiver: onReceive: a=android.intent.action.BOOT_COMPLETED Intent { act=andro
05-13 18:06:48.037 2146 2146 I AlarmClock: AlarmInitReceiver android.intent.action.BOOT_COMPLETED
05-13 18:06:48.359 2804 2804 D HfpClientConnService: onCreate
05-13 18:06:48.365 2804 2804 D HfpClientConnService: onStartCommand Intent { cmp=com.android.bluetooth/.hfpclient.o
05-13 18:06:48.469 3649 3649 D MainActivity: onCreate
05-13 18:06:48.507 3649 3649 D MediaFragment: onCreateView
05-13 18:06:48.552 3649 3649 D HereNavigationNotificationService: onCreate()
05-13 18:06:48.571 2266 2266 D PhoneSwitcherNetworkRequestListener: Registering NetworkFactory
05-13 18:06:48.571 1377 1825 D ConnectivityService: Got NetworkFactory Messenger for PhoneSwitcherNetworkRequestList
05-13 18:06:48.601 2266 2266 I VvmSimStateTracker: android.intent.action.BOOT_COMPLETED
05-13 18:06:48.674 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=15
05-13 18:06:48.674 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=12
05-13 18:06:48.674 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ BACKGROUND_REQUEST
05-13 18:06:48.674 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ REQUEST id=7, [ Cap
05-13 18:06:48.674 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ REQUEST id=1, [ Cap
05-13 18:06:48.675 3053 3053 D CarLatinIME: onCreate
05-13 18:06:48.706 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=18
05-13 18:06:48.726 2266 2266 D PhoneGlobals: Radio technology switched. Now CDMA is active.
05-13 18:06:48.945 1377 2112 I ActivityManager: Finished processing BOOT_COMPLETED for u0
05-13 18:06:49.498 2266 2266 D PhoneGlobals: Radio technology switched. Now GSM is active.
05-13 18:06:50.619 4002 4002 D MainActivity: onCreate
05-13 18:06:50.648 4002 4002 D MediaFragment: onCreateView
05-13 18:06:50.685 4002 4002 D HereNavigationNotificationService: onCreate()
05-13 18:06:50.804 3053 3053 D CarLatinIME: onCreate
05-13 18:06:51.837 1377 1419 D WifiService: Handle boot completed
05-13 18:06:52.022 4078 4078 D ClusterService: onCreate
05-13 18:06:52.039 4154 4154 I bootstat: Service started: /system/bin/bootstat --record_boot_complete --record_boot
05-13 18:06:52.077 4002 4002 D MainActivity: onCreate
05-13 18:06:52.116 4002 4002 D MediaFragment: onCreateView
05-13 18:06:52.121 4002 4002 D MediaFragment: onCreateView
05-13 18:06:52.180 4172 4172 I AlarmClock: AlarmInitReceiver android.intent.action.LOCKED_BOOT_COMPLETED
05-13 18:06:52.236 3053 3053 D CarLatinIME: onCreate
05-13 18:06:52.266 1377 1433 D VoiceInteractionManagerService: switchImplementation(10) took to complete: 0ms

```

```

05-13 18:06:52.269 1377 1433 I ActivityManager: Posting BOOT_COMPLETED user #10
05-13 18:06:52.299 2699 2699 D LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
05-13 18:06:52.300 2699 2699 D LucidSystemService: START LucidCarLauncher
05-13 18:06:52.344 4258 4258 D ONSAutoboot: Received android.intent.action.LOCKED_BOOT_COMPLETED
05-13 18:06:52.347 2699 2699 D LucidSystemService: onStartCommand :isFteRequiredfalse
05-13 18:06:52.429 4291 4291 D CM.MessengerService: onCreate
05-13 18:06:52.458 4361 4361 D LucidLauncherActivity: onCreate
05-13 18:06:52.460 4344 4344 D LucidSysUIService: onCreate Called

```

- 如果只是启动了 LucidSystemUI, 那么我再点击进入 LucidCarSettings 呢?

```

05-13 18:06:48.675 3053 3053 D CarLatinIME: onCreate
05-13 18:06:48.706 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=18, [ Ca
05-13 18:06:48.726 2266 2266 D PhoneGlobals: Radio technology switched. Now CDMA is active.
05-13 18:06:48.945 1377 2112 I ActivityManager: Finished processing BOOT_COMPLETED for u0
05-13 18:06:49.498 2266 2266 D PhoneGlobals: Radio technology switched. Now GSM is active.
05-13 18:06:50.804 3053 3053 D CarLatinIME: onCreate
05-13 18:06:52.022 4078 4078 D Cluster.Service: onCreate
05-13 18:06:52.039 4154 4154 I bootstat: Service started: /system/bin/bootstat --record_boot_complete --record_boot_reaso
05-13 18:06:52.180 4172 4172 I AlarmClock: AlarmInitReceiver android.intent.action.LOCKED_BOOT_COMPLETED
05-13 18:06:52.236 3053 3053 D CarLatinIME: onCreate
05-13 18:06:52.266 1377 1433 D VoiceInteractionManagerService: switchImplementation(10) took to complete: 0ms
05-13 18:06:52.269 1377 1433 I ActivityManager: Posting BOOT_COMPLETED user #10
05-13 18:06:52.344 4258 4258 D ONSAutoboot: Received android.intent.action.LOCKED_BOOT_COMPLETED
05-13 18:06:52.347 2699 2699 D LucidSystemService: onStartCommand :isFteRequiredfalse
05-13 18:06:52.429 4291 4291 D CM.MessengerService: onCreate
05-13 18:06:52.656 1377 3021 I ActivityTaskManager: START u10 {flg=0x10000000 cmp=com.lucid.car.carsettings/.home.HomeAct
05-13 18:06:53.275 3053 3053 D CarLatinIME: onCreate
05-13 18:06:53.535 4793 4793 E AndroidRuntime: ^Iat com.android.car.radio.RadioService.onCreate(RadioService.java:216)
05-13 18:06:54.049 1377 1434 I ActivityManager: Start proc 4883:com.here.hnod.feynman/u10s1000 for broadcast {com.here.hn
05-13 18:06:54.646 4922 4922 E AndroidRuntime: ^Iat com.android.car.radio.RadioService.onCreate(RadioService.java:216)
05-13 18:06:54.866 1377 1434 I ActivityManager: Start proc 5001:com.lucid.assistant/u10s1000 for broadcast {com.lucid.ass
05-13 18:06:55.065 1377 1434 I ActivityManager: Start proc 5047:com.lucid.car.service/u10s1000 for broadcast {com.lucid.c
05-13 18:06:55.158 1377 1434 I ActivityManager: Start proc 5080:com.lucid.profilessetup/u10s1000 for broadcast {com.lucid.
05-13 18:06:55.292 1377 1434 I ActivityManager: Start proc 5155:com.lucid.userprofile/u10s1000 for broadcast {com.lucid.u
05-13 18:06:55.697 5349 5349 D AlertReceiver: onReceive: a=android.intent.action.BOOT_COMPLETED Intent { act=android.inte
05-13 18:06:55.826 4172 4172 I AlarmClock: AlarmInitReceiver android.intent.action.BOOT_COMPLETED
05-13 18:06:55.982 5537 5537 I ContactsPerf: VoicemailContentProvider.onCreate start
05-13 18:06:55.991 5537 5537 I ContactsPerf: VoicemailContentProvider.onCreate finish
05-13 18:06:56.289 2488 2488 D com.lucid.userprofile.utils.KeyFobUtil: Properties are ready for switchUser
05-13 18:06:56.290 2488 2488 D com.lucid.userprofile.utils.KeyFobUtil: switchUserForKeyFobDetection()
05-13 18:06:56.291 2488 2488 D com.lucid.userprofile.utils.KeyFobUtil: switching to default primary profile
05-13 18:06:56.292 2488 2488 D UserProfileUtil: Switching to userId: 10
05-13 18:06:56.293 1377 3028 I ActivityManager: switching to user 10
05-13 18:06:56.302 2488 2488 D com.lucid.userprofile.services.TriggerScanService: onCreate()
05-13 18:06:56.308 2488 2488 D com.lucid.userprofile.services.TriggerScanService: onStartCommand()
05-13 18:07:00.014 3214 3214 D KeyguardClockSwitch: Updating clock: 607
05-13 18:07:01.054 2266 2266 D PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=24, [ Ca
05-13 18:07:01.152 1377 4001 I ActivityManager: Finished processing BOOT_COMPLETED for u10

```

## 7.5 LucidSysUIService.java launching logics

```

D LucidSysUIService: onCreate Called
D LucidSysUIService: creating alarm manager service
D CarNotificationListener: onNotificationPosted: StatusBarNotification(pkg=com.lucid.car.sysui user=UserHandle{11} id=1001
Notification(channel=LucidSysUIService pri=0 contentView=null vibrate=null sound=null defaults=0x0 flags=0x62 color=0x000
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
D LucidSysUIService: settings system property done
D LucidSysUIService: onStartCommand
D LucidSysUIService: (ACTION_LAUNCH_CID.equals(action)): true
D LucidSysUIService: (!FteUtil.isFteRequired(context)): true
D LucidSysUIService: send stop animation to profiles
E LucidSysUIService: OTA_REMIND_LATER_TIME
W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.showOTANotifications(LucidSysUIService.java:887)
W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.onStartCommand(LucidSysUIService.java:397)
E LucidSysUIService: TCU vHal connected
D TcuController: fetchWifiSignalState strength: 0 status: 0 mCtrlsListener: com.lucid.car.sysui.LucidSysUIService$5@f6e298e
D LucidSysUIService: onWifiConnectionStatusChangedW wifiStatus: 0
D TcuController: fetchLteSignalState strength: 0 status: 0 mCtrlsListener: com.lucid.car.sysui.LucidSysUIService$5@f6e298e
D LucidSysUIService: onLteConnectionStatusChangedW Lte_Status: 0 Wifi_Status: 0
D LucidSysUIService: Steering Wheel VHAL Connected
D LucidSysUIService: Steering Wheel property ready
W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService$1.run(LucidSysUIService.java:184)

```



```

D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 0.0 temp 25.0
D LucidSysUIService: onUpdateOutsideTemperatureW25.0
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
E ActivityManager: Sending non-protected broadcast com.lucid.home.ProfileSwitcher.ACTION_USER_LOGGED_IN from system 12202:c
E ActivityManager: Sending non-protected broadcast com.lucid.home.ProfileSwitcher.ACTION_USER_LOGGED_IN from system 12202:c
D LucidSysUIService: mWifiEnableReceiver :com.lucid.carlib.ctrls.ACTION_WIFI_ENABLE
D LucidSysUIService: wifiEnable: false
D LucidSysUIService: onReceive: com.lucid.carlib.ctrls.ACTION_WIFI_ENABLE wifiEnable: false Wifi_Strength: 0 sWifiConnected
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0

```

## 7.6 LucidSystemService launches LucidSysUIService

```

D LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
D LucidSystemService: START LucidCarLauncher
D LucidSystemService: onStartCommand :isFteRequired false
D LucidSystemService: onReceive: intent android.intent.action.USER_STOPPED
I ActivityManager: Start proc 21885:com.lucid.car.sysui/u10s1000 for service {com.lucid.car.sysui/com.lucid.car.sysui.LucidSysUIService}
D LucidSysUIService: onCreate Called
D LucidSysUIService: creating alarm manager service
D hvacFrontViewFragment: [FrontViewFragment.access$100():55] On properties ready in seat hvac profile switcher is called
D LucidSysUIService: settings system property done
D LucidSysUIService: onStartCommand
D LucidSysUIService: (ACTION_LAUNCH_CID.equals(action)): true
D LucidSysUIService: (!FteUtil.isFteRequired(context)): true
D LucidSysUIService: send stop animation to profiles

```

## 7.7 UI VHALS are connecting correctly

```

D LucidSysUIService: send stop animation to profiles
W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.showOTANotifications(LucidSysUIService.java:892) // 这些是 OTA 的
W System.err: ^Iat com.lucid.car.sysui.LucidSysUIService.onStartCommand(LucidSysUIService.java:402)
D IcrHvacPeekWindow: Park Assist Controller connected
D comfort_sport_%d: Switching from  to comfort_sport_%d
D IcrHvacPeekWindow: Gear Controller VHAL connected
D com.lucid.userprofile.ui.ProfileSwitchingCIDAnimation: removeAnimation()
D IcrHvacPeekWindow: Bcm Controller VHAL connected <===== BcmController VHAL connect
D bt_btif : bta_av_do_disc_a2dp: peer_addr: a0:4e:cf:f3:5c:e5 use_rc: 1 switch_res:0, oc:0
D bt_l2cap: l2cu_create_conn_after_switch :0 num_acl:0 no_hi: 0 is_bonding:0
D com.lucid.userprofile.ui.ProfileSwitchingICRAAnimation: removeAnimation()
D IcrHvacPeekWindow: Hvac Vhal connected
I bt_l2cap: L2CA_SetDesireRole() new:x1, disallow_switch:0
E LucidSysUIService: TCU vHal connected

```

## 7.8 IcrHvacPeekWindow initial values

```

D LucidSysUIService: Steering Wheel property ready
D IcrHvacPeekWindow: CCCQNXAppStatus value = -1
D IcrHvacPeekWindow: Gear value = 4
D IcrHvacPeekWindow: Gear value = 4
D IcrHvacPeekWindow: onhvacbtn clickedfalse
D IcrHvacPeekWindow: fan speed changed017
D IcrHvacPeekWindow: fan speed changed018
D IcrHvacPeekWindow: fan speed changed033
D IcrHvacPeekWindow: fan speed changed034
D IcrHvacPeekWindow: temp changed017
D IcrHvacPeekWindow: temp changed018
D IcrHvacPeekWindow: temp changed033
D IcrHvacPeekWindow: temp changed034

```

## 7.9 consitantly updating outside temp, but how do i know if BcmController is NOT break afterwards?

```

D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
E MapRegionSwitcher: [ERROR] MapRegionSwitcher - Rescheduling authorization request for map switch as current failed with:
E MapRegionSwitcher: [ERROR] MapRegionSwitcher - Rescheduling authorization request for map switch as current failed with:
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
D LucidSysUIService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0

```

## 7.10 switching user process

```
LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
com.lucid.userprofile.ui.ProfileSwitchingICRAnimation: showProfileSwitchingAnimation()
ActivityManager: switching to user 11
com.lucid.userprofile.ui.ProfileSwitchingICRAnimation: loadUserProfileInfo()
com.lucid.userprofile.ui.ProfileSwitchingCIDAnimation: showProfileSwitchingAnimation()
SystemServiceManager: Calling switchUser u11
SystemServiceManager: onSwitchUser-11 com.android.server.pm.Installer took to complete: 0ms
SystemServiceManager: onSwitchUser-11 com.android.server.os.DeviceIdentifiersPolicyService took to complete: 0ms
ActivityManager: Continue user switch oldUser #10, newUser #11
IcrHvacPeekWindow: Hvac Vhal disconnected
IcrSettingsWindow: onReceive(): android.intent.action.USER_SWITCHED <<<<<<<<<===== 为什么它是
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
IcrSettingsWindow: onReceive(): android.intent.action.USER_SWITCHED
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
GattService: User switched to userId 11
VoiceInteractionManagerService: switchImplementation(11) took to complete: 1ms
LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
LucidSystemService: START LucidCarLauncher
LucidSystemService: onStartCommand :isFteRequiredfalse
LucidSystemService: onReceive: intent android.intent.action.USER_STOPPED
ActivityManager: Start proc 24807:com.lucid.car.sysui/u11s1000 for service {com.lucid.car.sysui/com.lucid.car.sysui.LucidSy
BcmControllerLite: init was called
LucidSysUiService: onCreate Called
LucidSysUiService: creating alarm manager service
BcmControllerLite: Car service was connected successfully.
BcmController: fetchProperties() : BEGIN
BcmController: ONE_PEDAL_DRIVING_STS > ( 1 )
PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=57, [ Capabilities: INTERNET&NOT_RESTRICT
BcmController: fetchProperties() : END
BcmController: fetchProperties() : BEGIN
```

## 7.11 哪里蹦出来的 BcmControllerLite 呢？

```
IcrHvacPeekWindow: Hvac Vhal disconnected
IcrSettingsWindow: onReceive(): android.intent.action.USER_SWITCHED <<<<<<<<<===== 为什么它是
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
IcrSettingsWindow: onReceive(): android.intent.action.USER_SWITCHED
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
LucidCarProvider: action: android.intent.action.USER_SWITCHED, userId: 10
LucidSysUiService: onUpdateOutsideTemperatureW prevTemp 25.0 temp 25.0
GattService: User switched to userId 11
VoiceInteractionManagerService: switchImplementation(11) took to complete: 1ms
LucidSystemService: onReceive: intent android.intent.action.USER_UNLOCKED
LucidSystemService: START LucidCarLauncher
LucidSystemService: onStartCommand :isFteRequiredfalse
LucidSystemService: onReceive: intent android.intent.action.USER_STOPPED
ActivityManager: Start proc 24807:com.lucid.car.sysui/u11s1000 for service {com.lucid.car.sysui/com.lucid.car.sysui.LucidSy
BcmControllerLite: init was called
LucidSysUiService: onCreate Called
LucidSysUiService: creating alarm manager service
BcmControllerLite: Car service was connected successfully.
BcmController: fetchProperties() : BEGIN
BcmController: ONE_PEDAL_DRIVING_STS > ( 1 )
PhoneSwitcherNetworkRequestListener: got request NetworkRequest [ TRACK_DEFAULT id=57, [ Capabilities: INTERNET&NOT_RESTRICT
BcmController: fetchProperties() : END
BcmController: fetchProperties() : BEGIN
BcmController: ONE_PEDAL_DRIVING_STS > ( 1 )
hvacFrontViewFragment: [FrontViewFragment.access$100():55] On properties ready in seat hvac profile switcher is called
BcmController: fetchProperties() : END
BcmController: handleOnUpdateOnePedalDrivingW 1
BcmController: reported regen braking level is invalid
BcmController: VHAL ERROR - INVALID VAL IS SENT FOR RCCD LOCK
BcmController: VHAL ERROR - INVALID VAL IS SENT FOR HIGH BEAM POWER
BcmController: IVCU_LSC_MODE_STAT value 3
BcmController: LSC Mode update value 3
BcmController: creep mode 0
BcmController: creep mode status = 0
LucidSysUiService: settings system property done
LucidSysUiService: onStartCommand
LucidSysUiService: (ACTION_LAUNCH_CID.equals(action)): true
```

```

public void showWindow() {
    Log.d(TAG, " Show ICRhvacVolumeWindow");
    initTheme();
    if (mShowWindow) { // false
        initSystemPrefs();
        updateAutoCloseTimer();
        return;
    }
    mShowWindow = true;
    // 当 mHandler.post() 之后是异步进行的, 当前线程会继续往下执行, 而 post 之后的任务会异步执行, 两者同步发生着
    mHandler.post(mUpdateHvacUI); // ==> showWindow():
    // Attach window.
    WindowManager.LayoutParams lp = new WindowManager.LayoutParams(
        ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.MATCH_PARENT,
        WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY, // WindowManager.LayoutParams.TYPE_DISPLAY_OVERLAY,
        WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE
            | WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS
            & ~WindowManager.LayoutParams.FLAG_WATCH_OUTSIDE_TOUCH,
        PixelFormat.TRANSLUCENT);
    // .....
    mWindowManager.addView(mIcrPeakWindow, lp);
    initSystemPrefs();
    updateAutoCloseTimer();
}
private final Runnable mUpdateHvacUI = new Runnable() {
    @Override public void run() {
        mHandler.removeCallbacks(mUpdateHvacUI); // 移除之前的
        synchronized (mHvacController) {
            int dTempCelsiusValue = mHvacController.getTemperature(HvacController.ZONE_FRONT_LEFT);
            int pTempCelsiusValue = mHvacController.getTemperature(HvacController.ZONE_FRONT_RIGHT);
            int dFanSpeed = mapFanSpeed(0, 100, 0, 10,
                mHvacController.getFanSpeed(HvacController.ZONE_FRONT_LEFT));
            int pFanSpeed = mapFanSpeed(0, 100, 0, 10,
                mHvacController.getFanSpeed(HvacController.ZONE_FRONT_RIGHT));
            updateTempText(mDriverTemp, dTempCelsiusValue); // 温度总是有的
            updateTempText(mPassengerTemp, pTempCelsiusValue);
            if (dFanSpeed == 0) {
                mDriverFanSpeed.setText(mContext.getResources().getString(R.string.general_off)); // OFF, 这个需要设置显示吗
                mDriverFanImg.setVisibility(View.GONE);
                mDriverTemp.setVisibility(View.GONE);
            } else {
                mDriverTemp.setVisibility(View.VISIBLE);
                mDriverFanImg.setVisibility(View.VISIBLE);
                mDriverFanSpeed.setText(String.valueOf(dFanSpeed));
            }
            if (pFanSpeed == 0) {
                mPassengerFanImg.setVisibility(View.GONE);
                mPassengerFanSpeed.setText(mContext.getResources().getString(R.string.general_off));
                mPassengerTemp.setVisibility(View.GONE);
            } else {
                mPassengerTemp.setVisibility(View.VISIBLE);
                mPassengerFanImg.setVisibility(View.VISIBLE);
                mPassengerFanSpeed.setText(String.valueOf(pFanSpeed));
            }
        }
        showWindow(); // 不是又调用了一遍, 而是 handler 异步执行, 这里要再好好想一想
    }
};

```

## 8 jenny/IVIANDLIBS-349: Merge Unit Tests for SystemUI and Settings Apps: for LucidCarSettings App

- 以后记住, 只要是动过 build.gradle, 都需要真机上测试一下, 尤其是像这次也动过 lucid-bluetooth-lib, 也需要真机上测试一下这个模块没有被我弄坏
- 应该也是需要从 server jenkins 上来 build 一下 LucidCarSettings app, 来确保自己做对了的。这应该也是 Arun 还在 Jenkins 上测试的原因。

## 8.1 Unit test ContentProvider 举例

- <https://www.796t.com/post/M2lmNW8=.html>

```
import android.content.ContentResolver;
import android.database.Cursor;
import android.net.Uri;
import org.joda.time.LocalDate;
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.robolectric.Robolectric;
import org.robolectric.RobolectricTestRunner;
import org.robolectric.annotation.Config;
import org.robolectric.shadows.ShadowContentResolver;
import co.tomup.app.db.DbSchema;
import co.tomup.app.db.TomupContentProvider;
import co.tomup.app.db.model.CalendarDay;
import co.tomup.app.db.tables.CalendarDayTable;
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;
// @Config(emulateSdk = 18) // 这个应试可以不用了
@RunWith(RobolectricTestRunner.class)
public class CalendarDayProviderTest {
    private ContentResolver mContentResolver;
    private ShadowContentResolver mShadowContentResolver;
    private TomupContentProvider mProvider;
    @Before
    public void setup() {
        mProvider = new TomupContentProvider();
        mContentResolver = Robolectric.application.getContentResolver();
        mShadowContentResolver = Robolectric.shadowOf(mContentResolver);
        mProvider.onCreate();
        ShadowContentResolver.registerProvider(DbSchema.AUTHORITY, mProvider);
    }
    @Test
    public void testInsertAndDelete() {
        // insert
        CalendarDay calendarDay = new CalendarDay();
        calendarDay.setId(1L);
        calendarDay.setDay(new LocalDate());
        calendarDay.setMoonPhase("new");
        calendarDay.setSunrise(1);
        calendarDay.setSunset(100);
        Uri insertionId = mContentResolver.insert(CalendarDayTable.CONTENT_URI,
                                                    calendarDay.toSQLiteContentValues());
        Cursor cursorCheck = mShadowContentResolver.query(CalendarDayTable.CONTENT_URI,
                                                            null, null, null, null);

        while (cursorCheck.moveToNext()) {
            CalendarDay calendarDayCheck = CalendarDay.fromSQLiteCursor(cursorCheck);
            assertEquals(calendarDay, calendarDayCheck);
        }
        assertTrue(cursorCheck.getCount() > 0);
        // delete
        mShadowContentResolver.delete(insertionId, null, null);
        cursorCheck = mShadowContentResolver.query(CalendarDayTable.CONTENT_URI, null, null, null, null);
        assertTrue(cursorCheck.getCount() == 0);
    }
}
```

## 9 jenny/IVIANDLIBS-351: Merge Unit Tests for SystemUI / Connectivity: for lucid-bluetooth-lib