

Weekly Questions - 2:1

deepwaterooo

May 17, 2022

Contents

1	关于扫描人头的时候不出现，之后若不是已经倒车，就应该出现？确认一下	1
2	question: 我觉得那两部分的代码没用，没有用，可以删掉	1
3	Controller == VHAL 双向信号：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 ?????? 这个需要搞懂	2
4	怎么 book 车：	2
5	Note: S2R does NOT work for RC cars. they are moving some cars here	3
6	todo: 在 framework 层，有 SystemUI 应用？？它应该是负责 broken Notification Channels，可以问一下 Soumya 6.0.1 LifecycleOwner 相关的：需要自己再深入地理解一下	3
7	to be remembered: build.gradle for LucidCarSettings App or for any lib, for example lucid-bluetooth-lib	4
8	question: 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；	4
9	question: can we remove this part from LucidSysUiService.java ???	6

1 关于扫描人头的时候不出现，之后若不是已经倒车，就应该出现？确认一下

```
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(); // 也就是说：在用户已经在倒车时，是不能再显示的，但只要还没有倒车，若有剩余时间
}
private boolean isQNXAppClosed() {
    return mParkAssistController.getCurrentAppStatus() == ParkAssistController.CCC_QNX_ALL_APPS_CLOSED
        || mParkAssistController.getCurrentAppStatus() == -1; // LucidParkAssistApp: currently assign -1 to this app due to ...
}
private final ParkAssistController.ParkAssistListener mParkAssistListener =
    new ParkAssistController.ParkAssistListener() {
        @Override
        public void onReceiveCCCAppStatusW(Integer cccQNXAppStatus) {
            Log.d(TAG, "CCCQNXAppStatus value = " + cccQNXAppStatus);
            if (mShowWindow && cccQNXAppStatus == ParkAssistController.CCC_QNX_ALL_APPS_CLOSED) {
                Log.d(TAG, "All QNX apps closed");
                showWindow();
            }
        }
    };
}
```

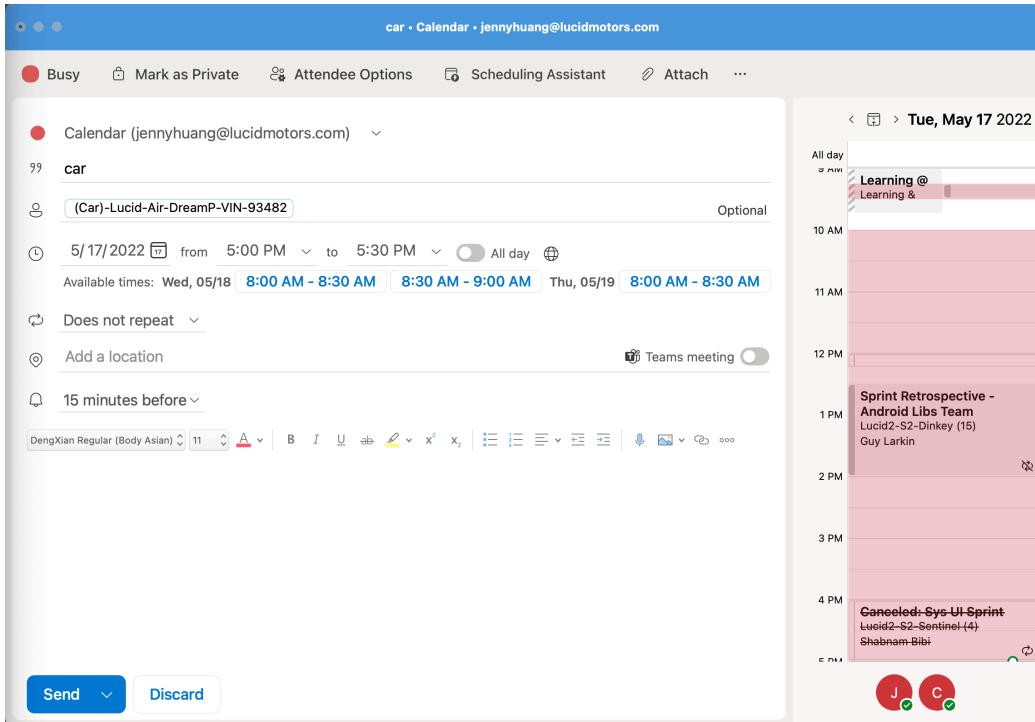
2 question: 我觉得那两部分的代码没用，没有用，可以删掉

```
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处理一下
        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是什么关系
    mBtnEngaged = true;
    mHandler.post(mHvacBtnsRunnable);
    // 我觉得这两部分的代码没有用到，用不上.....
    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);
}}
```

3 Controller == VHAL 双向信号：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 ?????? 这个需要搞懂

4 怎么 book 车:

- Revanth 发了一个链接: <https://lucidmotors.atlassian.net/wiki/spaces/INFO/pages/2159839721/Infotainment+Software+test+cars>
 - 两个 A10 的车: Vin52373 和 Vin93482
 - 我笔记本的 bypass TCU 还没有过, 今天还不能用: 上个周五有弄个 ticket, 好像还没有过呀



5 Note: S2R does NOT work for RC cars. they are moving some cars here

6 todo: 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/

6.0.1 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
- 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
- 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
- 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);
        }
    }
}

```

7 to be remembered: build.gradle for LucidCarSettings App or for any lib, for example lucid-bluetooth-lib

- 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 question: 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；

- 后来去车上测，确实是这样，只是在初始化的时候才收到两条回调，那么现在的问题是：对于 IcrHvacPeekWindow 里 BcmController 的新增四个回调，就没有初始化，就一定会返回 true 吗？我不太理解，也不信呀。。。。。
- Soumya 说了，这个没有关系，因为即使它显示也只是在不该显示的时候显示 1 秒，那么有必要的时候，我们随时可以再把它改掉。现在没有关系

```

private final BcmController.BcmListener mBcmListener =
    new BcmController.BcmListener() {
        @Override public void onVhalConnected() {
            Log.d(TAG, "Bcm Controller VHAL connected");
        }
    }

```

```

@Override public void onVhalDisconnectedW() {
    Log.d(TAG, "Inputs Controller VHAL disconnected");
}
@Override public void onUpdateVolumeRollerW(int adjustValue) {
    if (mShowWindow)
        hideWindow();
}
@Override public void onUpdateHvacBtnDriverTemperature(int state) {
    mBtnEngaged = true; // 为什么 Soumya 就可以这么确信地认为他们被按，初始化的时候收到回调吗？是被按了吗？
    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; // 为什么 Soumya 就可以这么确信地认为他们被按，初始化的时候收到回调吗？是被按了吗？
    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; // 为什么 Soumya 就可以这么确信地认为他们被按，初始化的时候收到回调吗？是被按了吗？
    mHandler.post(mHvacBtnsRunnable);
}
@Override public void onUpdateHvacBtnDPassengerFanSpeed(int state) {
    mBtnEngaged = true; // 为什么 Soumya 就可以这么确信地认为他们被按，初始化的时候收到回调吗？是被按了吗？
    mHandler.post(mHvacBtnsRunnable);
}
};

• 是在 BcmController 中的
=====

// 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();
}
// Connection to VHal
=====
private Car mCarApiClient;
private CarPropertyManager mPropertyManager;
private final ServiceConnection mCarConnectionCallback = new ServiceConnection() {
    @SuppressLint("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 )
            // 这里注册了回调，当初始化的时候，它们怎么可能返回 true 呢？
            mPropertyManager.registerCallback(mPropertiesListener, propId, 100 );
        // Fetch initial values.
        new AsyncTask<Void, Void, Void>() {

```

```

        @Override protected Void doInBackground(Void... nulls) {
            fetchProperties();
            return null;
        }
        @Override protected void onPostExecute(Void aVoid) {
            // Notify
            if (mBcmListener != null) {
                if (IS_DEBUG_BUILD) Log.d(TAG, "mBcmListener.onPropertiesReadyW()");
                mBcmListener.onPropertiesReadyW();
            }
        }
    }.execute();
} 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) { //...
}

};

private void fetchProperties() {
    Log.d(TAG, "fetchProperties() : BEGIN" );
    int value;
    float fValue;
    synchronized (mPropertiesLock) {
        try {
            value = mPropertyManager.getIntProperty(CID_POSITION_STATUS, 0);
            CidState cidState = (value < 0 || value > CidState.values().length ?
                CidState.CIDFAULT : CidState.values()[value]);
            mCidState = cidState;
            value = mPropertyManager.getIntProperty(MPB_EVSE_TYPE, 0);
            chargingCurrentStatus = value;
            //..... 还有很多.....
        }
    }
}

//=====
// 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) {
            if (IS_DEBUG_BUILD) Log.d(TAG, " onChangeEvent " + carPropertyValue.getPropertyId()
                + " (0x" + Integer.toHexString(carPropertyValue.getPropertyId()) + ")");
            switch (carPropertyValue.getPropertyId() ) {
                case CID_POSITION_STATUS :
                    handleUpdateCidStateW( (Integer) carPropertyValue.getValue() );
                    break;
                case INPUTS_VOLUME_ROLLER :
                    handleUpdateVolumeRollerW( (Integer) carPropertyValue.getValue() );
                    break;
                // case INPUTS_HVAC_BTNS_ENGAGED :
                //     handleUpdateHvacBtnsEngagedW( (Boolean) carPropertyValue.getValue() );
                //     break;
                case INPUTS_HVAC_BTN_DRIVER_TEMPERATURE:
                    handleUpdateHvacBtnDriverTemperature( (Integer) carPropertyValue.getValue() );
                    break;
                case INPUTS_HVAC_BTN_PASSENGER_TEMPERATURE:
                    handleUpdateHvacBtnPassengerTemperature( (Integer) carPropertyValue.getValue() );
                    break;
                case INPUTS_HVAC_BTN_DRIVER_FANSPEED:
                    handleUpdateHvacBtnDriverFanSpeed( (Integer) carPropertyValue.getValue() );
                    break;
                case INPUTS_HVAC_BTN_PASSENGER_FANSPEED:
                    handleUpdateHvacBtnPassengerFanSpeed( (Integer) carPropertyValue.getValue() );
                    break;
            }
        }
    };
}

```

}

9 question: can we remove this part from LucidSysUiService.java ???

- 横看成岭侧成峰，远近高低各不同；每看一次，总是忍不住会去想，曾经，曾几何时，它们是如何工作的呢？我翻遍了 lucid-carctrls-lib, lucid-hvac-lib, hal repositroy, framework 也再找不到任何的蛛丝马迹。。。。。
- 要不，我们还是把它留在这里，让它继续去迷惑曾经像我一样好奇害死猫的人。。。。。

```
public static final String ACTION_SHOW_ICR_PEAK_VIEW = // IVI 现在还有
    "com.lucid.car.sysui.ACTION_SHOW_ICR_PEAK_VIEW"; // IVI 现在还有
@Override public int onStartCommand(Intent intent, int flags, int startId) { // ISSW exists only
    if (null == intent || null == intent.getAction()) {
        mCidNavBarWindow.showWindow();
        Log.d(TAG, "Intent is null");
    } else {
        // 我想了多少次：是谁在发这个信号呢??? 踏尽铁鞋无觅处，洒尽了青春与热泪。。。。。。
        // Not useful, perhaps only for debugging.
        if (ACTION_SHOW_ICR_PEAK_VIEW.equals(intent.getAction())) {
            mIcrHvacPeekWindow.showWindow();
        }
        //....
    }
}
}}}
```