Indicator Light Integration Documentation

Version	Description	Date	Written by
V1.0	Added Indicator Light Control Instructions	2024/01/15	Wade Chen

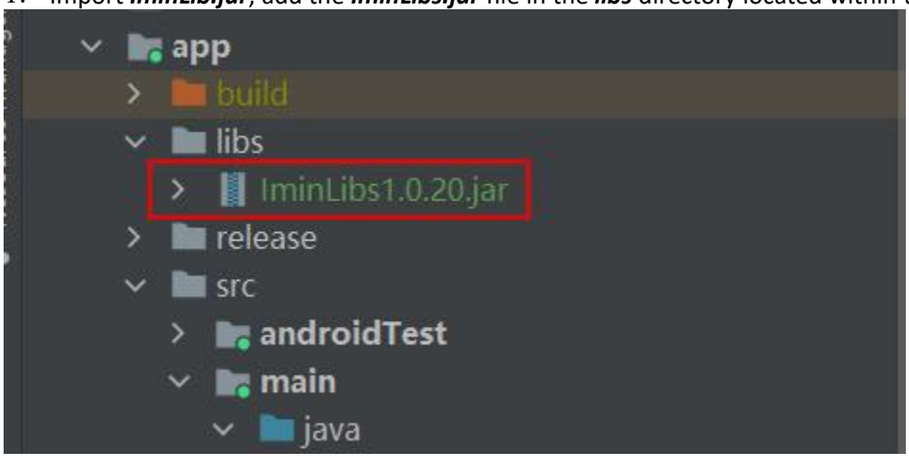
1.Overview

This documentation is designed to provide developers with instructions on controlling external indicator lights.

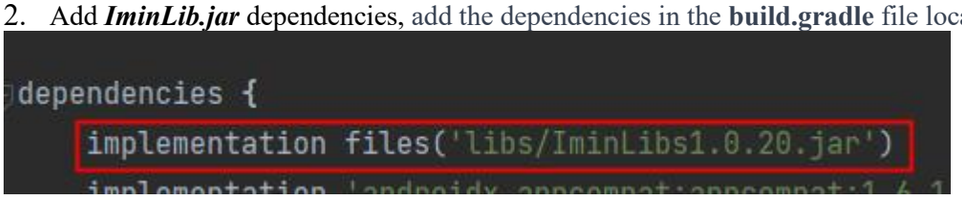
No	Model
1	Crane 1 16''、Crane 1 21.5''、Crane 1 27''、Crane 1 32''

2. Integration Steps

1. Import ***IminLib.jar***, add the ***IminLibs.jar*** file in the ***libs*** directory located within the **app** directory



2. Add ***IminLib.jar*** dependencies, add the dependencies in the **build.gradle** file located under the **app** directory



3. Declare permissions in AndroidManifest.xml

```
<uses-permission android:name="android.permission.USB_PERMISSION" />
<uses-permission android:name="android.hardware.usb.host" />
<uses-permission android:name="android.hardware.usb.accessory" />
```

```
    <uses-feature android:name="android.hardware.usb.accessory" />
    <uses-feature
        android:name="android.hardware.usb.host"
        android:required="true" />
```

1) Obtain indicator light device

```
UsbDevice usbDevice = IminSDKManager.getLightDevice(LightActivity.this);
```

2) Request to connect to indicator light

```
  @Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    ...
    mUsbManager = (UsbManager) this.getSystemService(Context.USB_SERVICE);
    ...
  }

public boolean requestPermission(UsbDevice usbDevice) {
        Log.d(TAG,"requestPermission============");
        if (usbDevice == null){
            return false;
        }

        if (mUsbManager.hasPermission(usbDevice)) {
            boolean isConnect =    IminSDKManager.connectLightDevice(LightActivity.this);
            Log.d(TAG,"usb is connect:"+isConnect);
            return true;
        } else {
            //@SuppressLint("UnspecifiedImmutableFlag") PendingIntent pendingIntent =
PendingIntent.getBroadcast(BaseApplication.getAppContext(), 0, new Intent(ACTION_USB_PERMISSION), 0);
            PendingIntent pendingIntent;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.S) {
                pendingIntent = PendingIntent.getBroadcast(this, 0, new Intent(ACTION_USB_PERMISSION),
PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);
            } else {
                pendingIntent = PendingIntent.getBroadcast(this, 0, new Intent(ACTION_USB_PERMISSION), 0);
            }

            IntentFilter intentFilter = new IntentFilter(ACTION_USB_PERMISSION);
            intentFilter.addAction(ACTION_USB_DEVICE_ATTACHED);
            intentFilter.addAction(ACTION_USB_DEVICE_DETACHED);
            this.registerReceiver(mUsbDeviceReceiver, intentFilter);
            mUsbManager.requestPermission(usbDevice, pendingIntent);
            return mUsbManager.hasPermission(usbDevice);
        }
    }


    private final BroadcastReceiver mUsbDeviceReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();
            Log.d(TAG,"UsbDeviceReceiver action = " + action);
            if (ACTION_USB_PERMISSION.equals(action)) {
                android.hardware.usb.UsbDevice device = intent.getParcelableExtra(UsbManager.EXTRA_DEVICE);
```

```
                if (intent.getBooleanExtra(UsbManager.EXTRA_PERMISSION_GRANTED, false)) {
                    if (device != null) {
                        //TODO connectAndOpen
                        boolean isConnect =    IminSDKManager.connectLightDevice(LightActivity.this);
                        Log.d(TAG,"Light Device isConnet ="+isConnect);
                    }
                }
            }else if(ACTION_USB_DEVICE_ATTACHED.equals(action)){
                //openUsbDevice();
            }else if(ACTION_USB_DEVICE_DETACHED.equals(action)){
                //TODO closeUsbDevice();
            }
        }
    };
```

3) Control indicator light

①. Turn on green light

```
IminSDKManager.turnOnGreenLight(Context context);
```

②.Turn on red light

```
IminSDKManager.turnOnRedLight(Context context);
```

③.Turn off light

```
IminSDKManager.turnOffLight(Context context);
```

4) Disconnect light

```
IminSDKManager.disconnectLightDevice(Context context);
```