扫码接入说明文档

版本	描述	日期	作者
V1.0	添加扫码接入说明	2023/11/25	陈继凯
V1.1	新增 Lark 1 机型	2024/4/3	陈继凯

1.简介

该文档用于介绍如何控制钱箱控制，面向开发者
针对机型

序号	机型
1	Crane 1 16''、Crane 1 21.5''、Crane 1 27''、Crane 1 32''
2	Lark 1

2.Demo 说明

1. 在 activity 中注册广播接收者

```
public static String DEVICE_CONNECTION = "com.imin.scanner.api.DEVICE_CONNECTION";
public static String DEVICE_DISCONNECTION = "com.imin.scanner.api.DEVICE_DISCONNECTION";
public static String RESULT_ACTION = "com.imin.scanner.api.RESULT_ACTION";
public static String CONNECTION_BACK_ACTION = "com.imin.scanner.api.CONNECTION_RESULT";
public static String CONNECTION_STATUS_ACTION = "com.imin.scanner.api.DEVICE_IS_CONNECTION";
public static String LABEL_TYPE = "com.imin.scanner.api.label_type";
public static String EXTRA_DECODE_DATA = "decode_data"; //extra string
public static String EXTRA_DECODE_DATA_STR = "decode_data_str"; //extra byte
public static String CONNECTION_TYPE = "com.imin.scanner.api.status";

String mResultAction;
private void registerScannerBroadcast() {
    mResultAction = etBroadAction.getText().toString();
    if(TextUtils.isEmpty(mResultAction)){
        mResultAction = RESULT_ACTION;
    }
    IntentFilter intentFilter = new IntentFilter();
    intentFilter.addAction(DEVICE_CONNECTION); // scanner connect broadcast receiver action
    intentFilter.addAction(DEVICE_DISCONNECTION); //scanner disconnect broadcast receiver action
    intentFilter.addAction(mResultAction); //scanner content broadcast receiver action
    intentFilter.addAction(CONNECTION_BACK_ACTION); //scanner status
    //create broadcast receiver
    scannerReceiver = new ScannerReceiver();
    //register
    registerReceiver(scannerReceiver, intentFilter);
}
```

DEVICE_CONNECTION：代表扫码头连接的广播 action
DEVICE_DISCONNECTION：代表扫码头断开连接的广播 action
RESULT_ACTION：代表扫码头返回内容的 action
EXTRA_DECODE_DATA：代表返回内容携带参数 extra 的 key（value 为字符串）
EXTRA_DECODE_DATA_STR：代表返回内容携带参数 extra 的 key（value 为 byte 字节）
CONNECTION_BACK_ACTION：请求扫码头连接状态的广播 action
CONNECTION_STATUS_ACTION：扫码头连接状态返回的广播 action

2. 广播接收者处理

```
class ScannerReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();
            if (action.equals(DEVICE_CONNECTION)) { //scanner is connect
                textView.append("\nUSB Connect");
            } else if (action.equals(DEVICE_DISCONNECTION)) { //scanner is disconnect
                textView.append("\nUSB Disconnect");
            } else if (action.equals(mResultAction)) { //scanner content
                String extraByteData = etBroadByteData.getText().toString();
                String extraData = etBroadData.getText().toString();
                if(TextUtils.isEmpty(extraByteData)){
                    //extraByteData = "source_byte";
                    extraByteData = EXTRA_DECODE_DATA;
                }
                if(TextUtils.isEmpty(extraData)){
                    //extraData = "data";
                    extraData = EXTRA_DECODE_DATA_STR;
                }
                String labelType = intent.getStringExtra(LABEL_TYPE);
                byte[] decodeData = intent.getByteArrayExtra(extraByteData);
                String strData = intent.getStringExtra(extraData);

                textView.append( "\nBroadcast result：byte：" + Arrays.toString(decodeData));
                textView.append( "\nBroadcast result：string：" + strData);
                textView.append( "\nscanner count"+ mScanCount++);
                if(mScanCount % 100 == 0){
                    textView.setText("");
                }
            } else if (action.equals(CONNECTION_BACK_ACTION)) { //获取扫码设备连接状态
                int type = intent.getIntExtra(CONNECTION_TYPE, 0);
                textView.append("\ndevice isConnection is " + (type == 1));
            }

        }
    }
```