Scanner Integration Documentation

Version	Description	Date	Written by
V1.0	Added scanner integration documentation	2023/11/25	Wade Chen
V1.1	Add Target model: Lark 1	2024/4/3	Wade Chen

1.Introduction

This documentation is designed to provide developers with instructions on controlling the scanner.

Target model

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

2.Demo Guideline

1. Register a broadcast receiver in activity

```
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：Indicate the action of scanner broadcast connection
DEVICE_DISCONNECTION：Indicate the action of scanner broadcast disconnection
RESULT_ACTION：Indicate the action of scanner return result
EXTRA_DECODE_DATA：Indicate that the return result carrying key with extra parameter (The value is a string）
EXTRA_DECODE_DATA_STR：Indicate that the return result carrying key with extra parameter（The value in byte format）
CONNECTION_BACK_ACTION：Indicate the broadcast action of requesting scanner connection status
CONNECTION_STATUS_ACTION: The return broadcast action of the scanner connection status

Broadcast receiver processing

```
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)) { //Obtain scanner device connection status
                int type = intent.getIntExtra(CONNECTION_TYPE, 0);
                textView.append("\ndevice isConnection is " + (type == 1));
            }

        }
    }
```