

标识: AEX-MJ-DEVEL-GUIDE

版本: V1.0

锁相开源门禁机 开发指南



深圳市安卓工控设备有限公司

2016 年 6 月

1. 引言

1.1. 标识

项目名称: 锁相开源门禁机(Lockaxial Open Source Door Access Equipment, 简称 LosDAE)

文档名称: 锁相开源门禁机

文档简称: 智能门禁机方案

文档标识号: AEX-LosDAE-V10

1.2. 概述

锁相开源门禁机(LosDAE)是互联网+安防门禁的应用实例。它是使用安卓系统的控制板结合腾讯 QQ 物联和微信设备实现安防门禁功能。

使用场景:

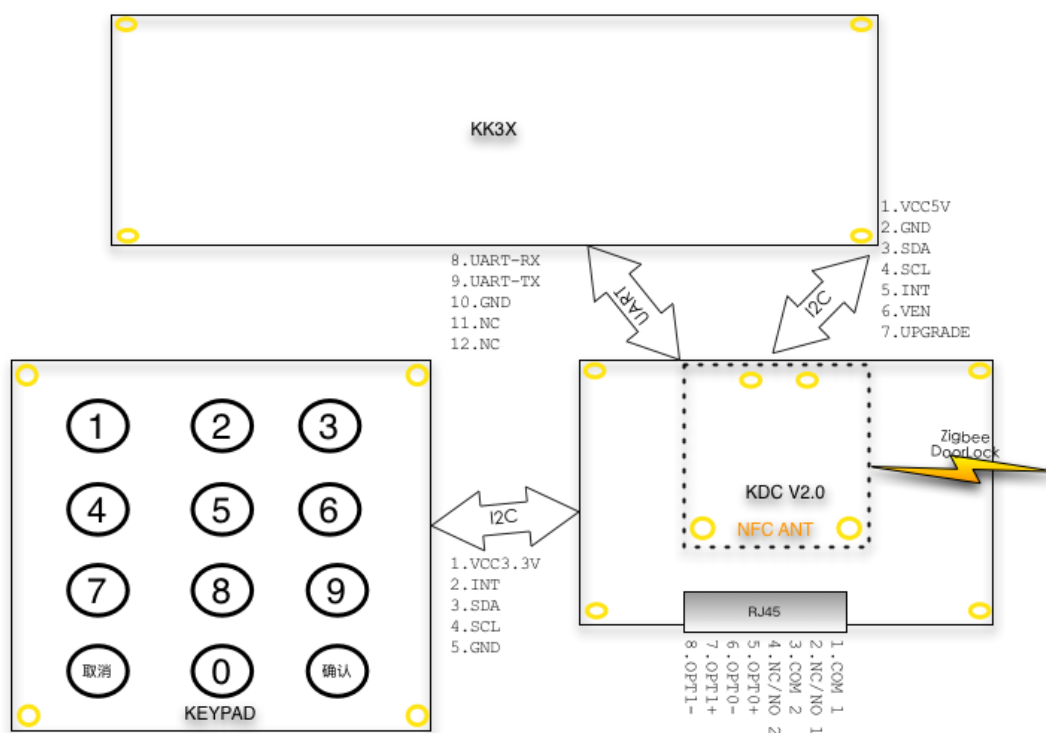
- 手机 QQ 远程开门;
 - a) 用户到门禁附近打开手机 QQ 点击开锁功能开门;
 - b) 访客在门禁机上按房号呼叫用户手机 QQ, 可视对讲开门。
- 近距离蓝牙连接开门;
 - c) 用户到门禁附近通过手机 QQ 的连接门禁蓝牙模块开门;
 - d) 蓝牙智能穿戴设备开门。
- 带 NFC 手机刷手机开门;
 - e) NFC 手机模拟门禁卡开门。
- 各种 NFC 卡开门
 - f) 小区物业卡、停车卡;
 - g) 公交卡;
 - h) 银联闪付卡;
 - i) 电卡、燃气卡等。

锁相开源门禁的特点:

- 开门的方式: 多样、便捷、安全;

- 实时反馈：下面的信息均会实时反馈到系统服务器和用户的手机上
 - 谁、什么时间、通过什么方式开门？
 - 开锁后门是否开启？门是否关好？
- 使用 QQ 物联及微信设备做云通讯平台：安全、可靠、运维成本低；
- 开放：客户可以定制硬件、软件、平台，打造自己的门禁机品牌。
 - 门禁机终端硬件开源：客户可以定制设备的外观、添加和裁剪功能；
 - 标准化配件：外壳、摄像头、按键板、门禁控制板、主控板等均提供标准化配件，降低客户定制成本；
 - 门禁机终端软件开源：具备基础门禁功能的锁相门禁机终端软件向所有客户开源，客户可以根据自己的实际需要定制。

2. 功能框图



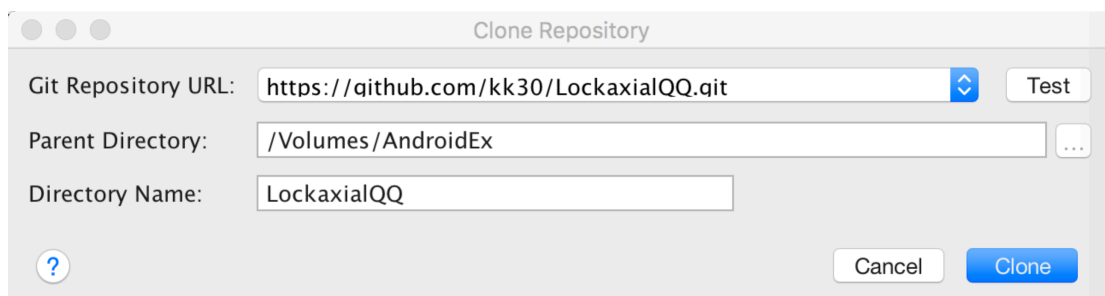
锁相开源门禁机分为三部分组成：安卓控制板、按键板、门禁控制器。

- ✧ 安卓控制板：采用 KK34V2.1，负责液晶显示、网络通讯、音视频采集 &编解码&传输、以及与门禁控制板通讯完成按键接收、开门控制、状态事件报告等功能；
- ✧ 门禁控制板：采用 KDC V2.0，负责转发按键信息，转发 NFC 读卡信息，控制门锁继电器，接收门磁输入状态；
- ✧ 按键板：采用 KEYPAD V1.0，实现 3X4 的按键，并通过 I2C 与门禁控制板连接。

3. SDK 下载

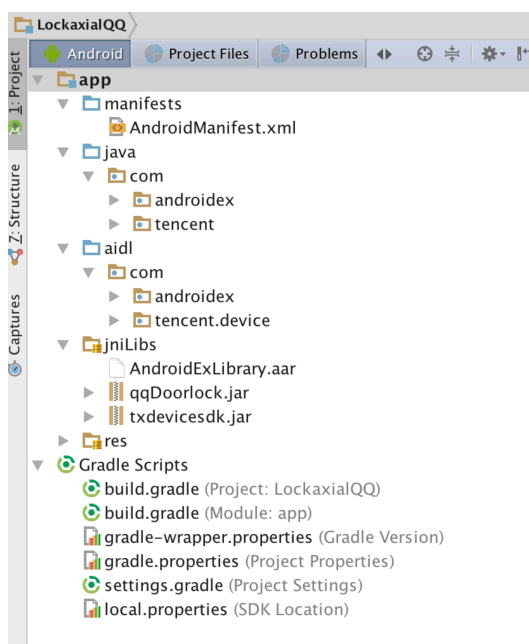
SDK 下载地址: <https://github.com/Lockaxial/LockaxialQQ.git>

使用 android studio 的 File->New->Project from version control->github



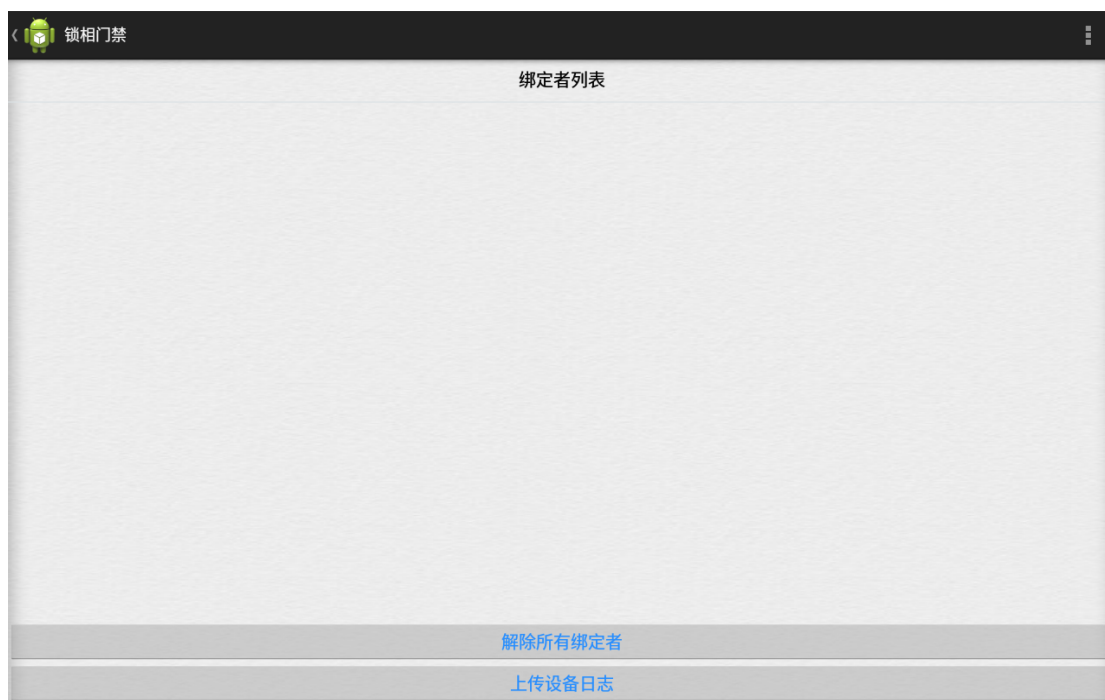
4. 编译 APK

使用 Android Studio 打开 LockaxialQQ 项目，选择 Build->Make Project

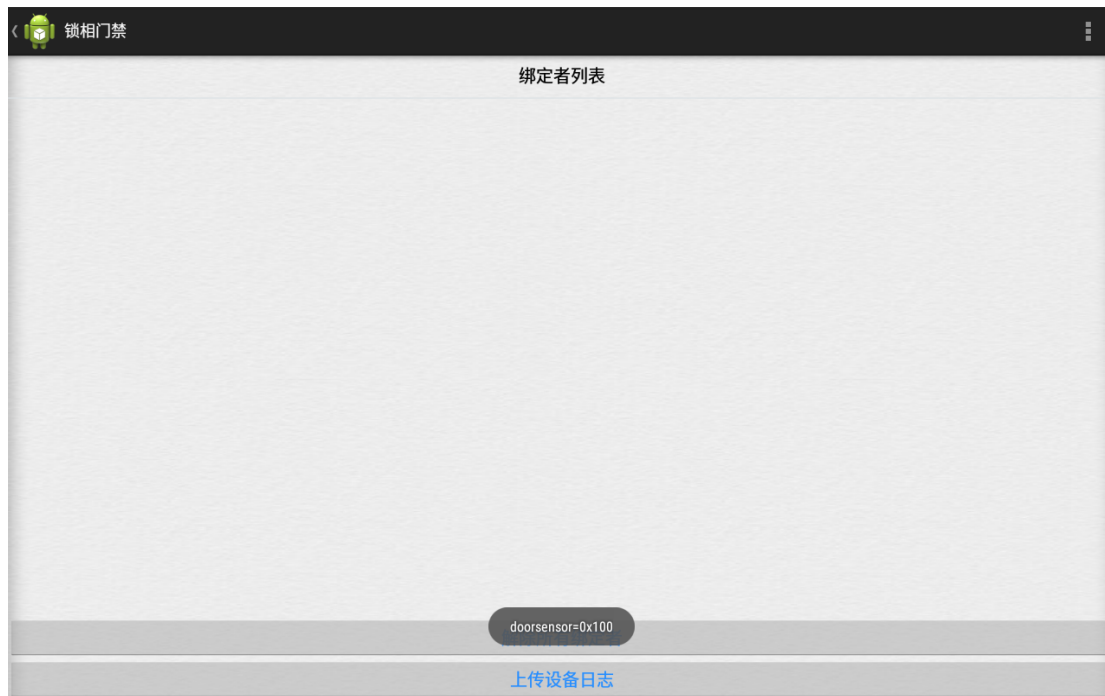


5. 运行测试

启动画面如下，正常时程序启动后会打开和关闭一下副门锁。



下面是开启副门后，程序收到的开门事件。



SDK 说明参见工程目录下的 Docs/index.html。



6. 开发要点

6.1. 板子硬件参数

板子硬件参数是指板子上固化区域参数的参数，一般包括：UUID，序列号，Mac 地址以及用户自定义的数据。即使格式化重新烧录固件这些数据仍然存在。

类：`com.androidex.plugins.kkaexparams`

方法概要

返回类型	方法和说明	说明
static void	runShellCommand (String cmd)	运行 shell 命令
String	getAndroidExParam (String address)	
int	setAndroidExParam (String address, String hexValue)	
boolean	has_pass ()	判断是否已经设置了密码
boolean	has_pass (String pass)	判断提供的参数是否是设置过密码的
void	clear_pass ()	清除密码
String	getPass ()	获得 HASH 密码
int	setPass (String hexPass)	设置 HASH 密码
boolean	verify_pass (String pass) 校验密码与存储的是否一致	校验密码
String	getUUID ()	获取 UUID
String	getSerial ()	序列号为 UUID 的后 32 位
String	getUserinfo ()	获取用户存储区数据
int	setUserinfo (String hexValue)	设置用户存储区数据

参数可以按照文件的方式存取，参数存取的目录为：

```
/sys/class/androidex_parameters/androidex
```

主要参数有：

```
bt_mac    flag1    notbefore  uuid
```

certsn	lan_mac	p7cert	serial	uevent	wlan_mac
flag0	notafter	password	state	userinfo	

读取参数的方式:

```
cat /sys/class/androidex_parameters/androidex/uuid
```

写入参数的方式:

```
echo "0x34" > /sys/class/androidex_parameters/androidex/flag0
```

如果写入的字符串中有引号的要使用单引号。如'{"A"}'。

手工设置参数示例:

```
echo "0x34">/sys/class/androidex_parameters/androidex/flag0
```

```
echo "d2bd1b63-f186-4b01-8325-e0f014fbf411" > /sys/class/androidex_parameters/androidex/uuid
```

```
echo "" > /sys/class/androidex_parameters/androidex/userinfo //这一句很重要，否则腾讯的 QQ 物联会出错
```

查看门禁机软件通过 ANDROID_ID 读取到的序列号的方法:

```
cat /sys/class/androidex_parameters/androidex/serial; echo "\n"
```

6.2. 开门

开门的实现是向/dev/rkey 写入指定格式的数据即可，参考代码如下:

```
String rkeyDev = "/dev/rkey";
int ident = 0;

/**
 * 开门指令
 * @param index    门的序号,主门=0,副门=1
 * @param delay    延迟关门的时间,0 表示不启用延迟关门,大于 0 表示延迟时间,延迟时间为 delay*150ms
 * @return        大于 0 表示成功,实际上等于 9 表示真正的成功,因为返回值表示写入的数据,开门指令长度为 9.
 */
public int openDoor(int index, int delay){
    kkfile rkey = new kkfile();

    if(index < 0 || index > 0xFE) index = 0;
    if(ident < 0 || ident > 0xFE) ident = 0;
    if(delay < 0 || delay > 0xFE) delay = 0;
    String cmd = String.format("FB%02X2503%02X01%02X00FE",ident,index,delay);
    int r = rkey.writeHex(rkeyDev,cmd);
    return r > 0?1:0;
}

public int closeDoor(int index){
    kkfile rkey = new kkfile();

    if(index < 0 || index > 0xFE) index = 0;
```

```
if(ident < 0 || ident > 0xFE) ident = 0;
String cmd = String.format("FB%02X2503%02X000000FE",ident,index);
int r = rkey.writeHex(rkeyDev,cmd);
return r > 0 ? 1:0;
}
```

请记住使用下面代码设置读写权限，一般把下面的代码放到程序的启动代码里：

```
kkaexparams.runShellCommand("chmod 0666 /dev/rkey");
```

6.3. 获取门状态

获取开门状态是通过监听广播来实现的，启动代码里添加

```
mReceiver = new NotifyReceiver();
IntentFilter filter = new IntentFilter();
filter.addAction("com.android.action.doorsensor");
.....
registerReceiver(mReceiver, filter);
```

接收代码如下：

```
if (intent.getAction().equals(mDoorSensorAction)){
    String doorsensor = intent.getStringExtra("doorsensor"); //获取广播消息内容
    UEventMap mds = new UEventMap(doorsensor);

    Intent ds_intent = new Intent();
    ds_intent.setAction(DoorLock.DoorLockStatusChange);
    ds_intent.putExtra("doorsensor",mds.get("doorsensor"));
    /**
     * doorsensor 的格式为 0xXXYY
     * XX : 表示门的序号, 00=主门, 01=副门
     * YY : 表示门的状态, 默认设置 00 表示打开, 01 表示关闭, 实际情况要
     根据门磁传感器接常开还是常闭来决定。
     */
    sendBroadcast(ds_intent);
}
```

全部示例代码如下：

```
/**
 *
 * 锁相开源门禁机软件的主要服务类,DoorLock 主要提供开门,关门指令以及上报门开
 * 和关闭的事件.
 */
```



```
public class DoorLock extends Service implements OnBackCall {

    public static final String TAG = "DoorLock";
    public static final String mDoorSensorAction = "com.android.action.doorsensor";

    private DoorLockServiceBinder mDoorLock;

    /**
     * 当门的状态改变时的事件定义
     */
    public static final String DoorLockStatusChange = "DoorLockStatusChange";
    /**
     * DoorLock 通过 DoorLockOpenDoor 广播获得开门指令并发送给门禁控制器
     */
    public static final String DoorLockOpenDoor = "DoorLockOpenDoor";
    private NotifyReceiver mReceiver;
    private static DoorLock mServiceInstance = null;

    public static DoorLock getInstance()
    {
        return mServiceInstance;
    }

    @Override
    public void onCreate() {
        super.onCreate();
        mDoorLock = new DoorLockServiceBinder();
        mReceiver = new NotifyReceiver();
        IntentFilter filter = new IntentFilter();
        filter.addAction(mDoorSensorAction);
        filter.addAction(DoorLockOpenDoor);
        registerReceiver(mReceiver, filter);
        int r = mDoorLock.openDoor(1,16);
        if(r == 9)
            Toast.makeText(DoorLock.this, String.format("Open %d,delay %ds
close.",1,16*150/1000), Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(DoorLock.this, String.format("Open door 1 fail
return %d.",r), Toast.LENGTH_SHORT).show();

        Log.d(TAG,String.format("open door %d",r));
    }

    @Override
```

```

    public void onDestroy() {
        unregisterReceiver(mReceiver);
        super.onDestroy();
        mServiceInstance = null;
        mDoorLock = null;
    }

    @SuppressWarnings("deprecation")
    @Override
    public void onStart(Intent intent, int startId) {

        super.onStart(intent, startId);
        mServiceInstance = this;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        return super.onStartCommand(intent, flags, startId);
    }

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return mDoorLock;
    }

    @Override
    public void onBackCallEvent(int code, String args) {
        Log.v("onBackCallEvent", args);
    }

    public class DoorLockServiceBinder extends IDoorLockInterface.Stub{
        String rkeyDev = "/dev/rkey";
        int ident = 0;

        /**
         * 开门指令
         * @param index    门的序号,主门=0,副门=1
         * @param delay    延迟关门的时间,0 表示不启用延迟关门,大于 0 表示延迟时间,延迟时间为 delay*150ms
         * @return        大于 0 表示成功,实际上等于 9 表示真正的成功,因为返回值表示写入的数据,开门指令长度为 9.
         */
        public int openDoor(int index, int delay){

```

```

        kkfile rkey = new kkfile();

        if(index < 0 || index > 0xFE) index = 0;
        if(ident < 0 || ident > 0xFE) ident = 0;
        if(delay < 0 || delay > 0xFE) delay = 0;
        String cmd =
String.format("FB%02X2503%02X01%02X00FE",ident,index,delay);
        int r = rkey.writeHex(rkeyDev,cmd);
        return r > 0 ? 1:0;
    }
    public int closeDoor(int index){
        kkfile rkey = new kkfile();

        if(index < 0 || index > 0xFE) index = 0;
        if(ident < 0 || ident > 0xFE) ident = 0;
        String cmd = String.format("FB%02X2503%02X000000FE",ident,index);
        int r = rkey.writeHex(rkeyDev,cmd);
        return r > 0 ? 1:0;
    }
}

public class NotifyReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(mDoorSensorAction)){
            String doorsensor = intent.getStringExtra("doorsensor");
            UEventMap mds = new UEventMap(doorsensor);

            Log.d(TAG, String.format("%s\t Door
sensor=%s\n",mds.get("doorsensor"), mds.toString()));

            Intent ds_intent = new Intent();
            ds_intent.setAction(DoorLock.DoorLockStatusChange);
            ds_intent.putExtra("doorsensor",mds.get("doorsensor"));
            sendBroadcast(ds_intent);
        } else if(intent.getAction().equals(DoorLock.OpenDoor)) {
            int index = intent.getIntExtra("index", 0);
            int status = intent.getIntExtra("status", 0);

            if (status != 0){
                mDoorLock.openDoor(index, 0x20);
            } else {
                mDoorLock.closeDoor(index);
            }
        }
    }
}

```

```

    }
}

}

public static final class UEventMap {
    // collection of key=value pairs parsed from the uevent message
    private final HashMap<String,String> mMap = new HashMap<String,String>();

    public UEventMap(String message) {
        int offset = 0;
        int length = message.length();

        if(length == 0)return;
        if(message.substring(0,1).equals("{")){
            message = message.substring(1);
        }
        if(message.substring(message.length() - 1,message.length()).equals("}")){
            message = message.substring(0,message.length() - 1);
        }
        length = message.length();
        while (offset < length) {
            int equals = message.indexOf('=', offset);
            int at = message.indexOf(',', offset);
            if (at < 0) break;

            if (equals > offset && equals < at) {
                // key is before the equals sign, and value is after
                mMap.put(message.substring(offset, equals).trim(),
                    message.substring(equals + 1, at).trim());
            }

            offset = at + 1;
        }
    }

    public String get(String key) {
        return mMap.get(key);
    }

    public String get(String key, String defaultValue) {
        String result = mMap.get(key);
        return (result == null ? defaultValue : result);
    }
}

```

```
public String toString() {  
    return mMap.toString();  
}  
}
```



深圳市安卓工控设备有限公司

ShenZhen AndroidEx Co., Ltd.

地址：深圳市龙岗布吉龙景工业园 E 栋二楼东

主页：<http://www.androidex.cn>

电话：+86-755-28435491