**密 级：内部公开**

**停车场ETC Controller**

**接口文档**

**广州匹伽信息科技有限公司**

|  |  |  |  |
| --- | --- | --- | --- |
| 文件状态 | 受控状态 | 文件标识： |  |
| [ √ ] 草 稿  [ ] 正式发布  [ ] 正在修改 | [ √] 受 控  [ ] 非受控 | 当前版本： | V3.0 |
| 编 制 者： | 钟杰伦 |
| 审 核 者： |  |
| 批 准 者： |  |
| 批准日期： |  |

---------------------------------------------------------------------

广州匹伽信息科技有限公司对本文件资料享受著作权及其它专属权利，未经书面许可，不得将该等文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用。

* + 1. 文件更改摘要：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本号** | **修订说明** | **修订人** | **审核人** | **批准人** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1.功能概述 3](#_Toc469922098)

[2.动态库使用说明 4](#_Toc469922099)

[2.1.所需文件清单 4](#_Toc469922100)

[2.2.部署说明 4](#_Toc469922101)

[2.3.配置文件说明 5](#_Toc469922102)

[2.4.日志格式说明 6](#_Toc469922103)

[3.数据格式说明 7](#_Toc469922104)

[4.动态库接口说明 8](#_Toc469922105)

[4.1.初始化和反初始化 8](#_Toc469922106)

[4.1.1.初始化 8](#_Toc469922107)

[4.1.2.反初始化 9](#_Toc469922108)

[4.2.天线控制 10](#_Toc469922109)

[4.2.1.读OBU接口 10](#_Toc469922110)

[4.2.2天线写出入口信息和消费接口 14](#_Toc469922111)

[4.3.读卡器控制 17](#_Toc469922112)

[4.3.1.获取卡片信息接口 17](#_Toc469922113)

[4.3.2读卡器写出入口信息和消费接口 20](#_Toc469922114)

[4.4.流水处理 23](#_Toc469922115)

[4.4.1流水上传接口 23](#_Toc469922116)

[4.4.2流水查询接口 27](#_Toc469922117)

[4.5.黑名单查询 31](#_Toc469922118)

[4.5.1黑名单查询接口 31](#_Toc469922119)

[4.6.设备管理 33](#_Toc469922120)

[4.6.1设备状态查询接口 33](#_Toc469922121)

[4.7.非现金接口 35](#_Toc469922124)

[4.7.1非现金支付请求接口 35](#_Toc469922125)

[4.7.2非现金支付结束接口 37](#_Toc469922126)

[4.7.3交易明细查询接口 38](#_Toc469922127)

[4.8.获取当前线程上一次输出 42](#_Toc469922128)

[5.调用流程说明 43](#_Toc469922129)

[5.1.天线交易流程(入口) 43](#_Toc469922130)

[5.2.读卡器交易流程(入口) 44](#_Toc469922131)

[5.3.天线交易流程(出口) 45](#_Toc469922132)

[5.4.读卡器交易流程(出口) 46](#_Toc469922133)

[5.5.车卡绑定校验流程 47](#_Toc469922134)

[5.6.非现金支付流程 48](#_Toc469922135)

[附录A 49](#_Toc469922136)

[A.1.天线相关错误码说明 49](#_Toc469922137)

[A.2.读卡器相关错误码说明 50](#_Toc469922138)

[A.3.业务进程相关错误码说明 51](#_Toc469922139)

[A.4.非现金进程相关错误码说明 51](#_Toc469922140)

1.功能概述

文件名称:EtcController.dll

功能描述:

(1)初始化和反初始化

(2)天线设备操作，包括打开天线，关闭天线，OBU搜索，获取OBU车辆信息等

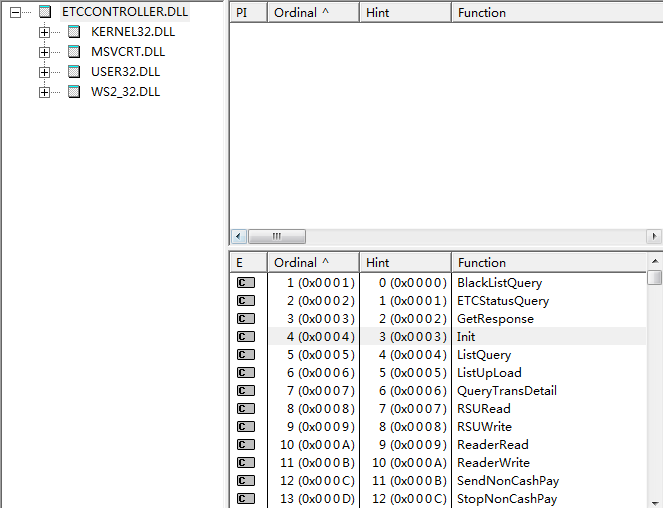
(3)读卡器设备操作，包括打天读卡器，关闭读卡器，寻卡，读卡，写卡等

(4)流水处理，包括流水上传，流水查询等

(5)黑名单查询

(6)非现金支付

动态库接口函数如下图所示:



注意事项:

(1)本文档所述的接口可能会由于网络、硬件操作等原因，造成阻塞。因此建议在新的线程中调用接口，以免对主线程和实时性要求高的线程造成影响。

(2)目前非现金支付方式只支持ETC卡支付（不写出入口信息）和微信支付。若需要使用微信支付方式，请自行登录后台系统下载微信支付的二维码，并打印粘贴于收费站。

2.动态库使用说明

## 2.1.所需文件清单

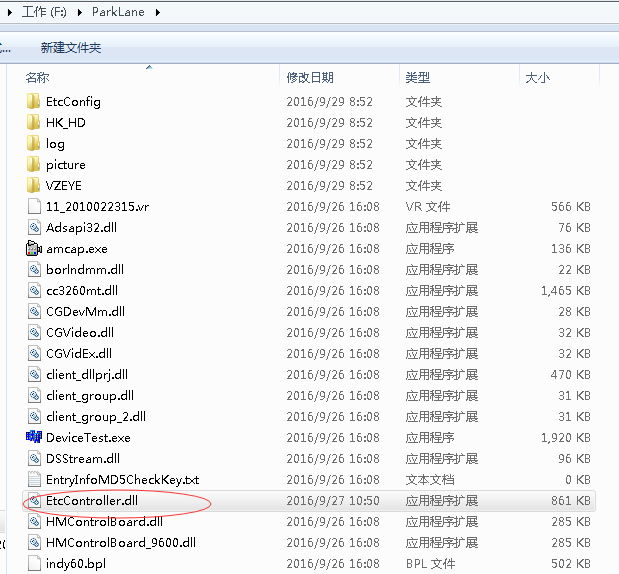
动态库文件:EtcController.dll

配置文件:EtcController.ini

## 2.2.部署说明

以华软小区车道程序为例:

第一步:只需将EtcController.dll拷贝到车道程序安装目录下,如下图所示:



第二步:创建EtcConfig目录，并将EtcController.ini拷贝到该目录下，如下图所示:



## 2.3.配置文件说明

(1)配置文件所在路径

车道程序安装目录下的EtcConfig文件夹

(2)配置文件名称

EtcController.ini

(3)配置项说明

EtcController.ini文件配置项说明如下:

[Global] #日志文件相关配置信息

LogDir=EcClientLog #日志文件存放目录

LogKeepDays=30 #日志文件保留天数(1-30，超过范围视为30)

EcLaneNum=1 #车道数量(1-2)

[EcLane0] #车道1配置信息

IpAddr=10.200.200.233 #IP地址

Port=33333 #端口(1024-65535)

TimeOut=5 #超时时间（秒，1-120）

HeartBeatTime=15 #心跳间隔（秒，1-120）

UserName=churukou3 #用户名

PassWord=123456 #密码

ProvinceNo=44 #省份编号（00-99）

CityNo=01 #城市编号（00-99）

AreaNo=3333 #小区编号（0000-9999）

GateNo=33 #大门编号（00-99）

LaneNo=55 #车道号（00-99）

RSUID=0 #天线ID(-1至1，天线ID编写规则，从零开始自增，-1表示没有)

ReaderID=-1 #读卡器ID(-1至1，同天线)

AutoLogout=0 #自动登出(0-1，0表示不做自动登出，1反之）

[EcLane1] #车道2配置信息

IpAddr=192.168.109.153

Port=33333

TimeOut=5

HeartBeatTime=15

UserName=churukou4

PassWord=123456

ProvinceNo=44

CityNo=01

AreaNo=3333

GateNo=33

LaneNo=56

RSUID=1

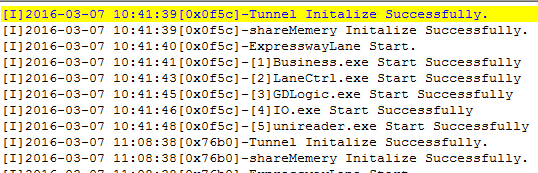
ReaderID=1

AutoLogout=0

## 2.4.日志格式说明

日志文件按天生成,命名规则:EtcController\_YYYYMMDD.log

日志文件内容如下图所示:



说明:[I] 表示该条日志等级为Info,[E]表示该条日志等级为Error

2016-03-07 10:41:39 表示该条日志生成时间

[0x0f5c] 表示对应的线程ID

Tunnel Initalize Successfully 表示该条日志内容

3.数据格式说明

类似如下函数接口：

int \_\_stdcall RSURead(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen);

参数说明：

(1)iLaneNo: 车道编号

(2)pReg：车道程序向嵌入式控制系统发送的请求包

采用json字符串格式，如下所示：

{“ID”:”12”,”name”:”zhubw”,age:”18”}

需要注意的是：所有字段值都是string类型（对应C语言中的char\*）

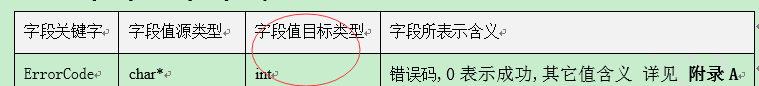
(3)pResp: 嵌入式控制系统返回给车道程序的回应包

采用json字符串格式，如下所示：

{“ID”:”12”,”name”:”zhubw”,age:”18”}

需要注意的是：所有字段值都是string类型（对应C语言中的char\*）

另外，如下图所示：



此处要求小区车道程序将ErrorCode字段值由char\*转化成int类型,但只是**建议级别**，小区车道程序可由实际情况而定;其他字段值情况类似，不与累述

(4)pLen: 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。

当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口。

4.动态库接口说明

## 4.1.初始化和反初始化

### 4.1.1.初始化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 初始化 | | | |
| 函数原型 | int \_\_stdcall Init(char\* pResult, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：初始化成功  5: pResult长度不足，请调用GetResponse接口  其他：失败 | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输出 | pResult | char\* | 返回0时为配置信息，车道数量的json数组形式的字符串。  例如:  [{“Port” : “33333”, 省略}, {“Port” : “33333”,省略}]  返回其他值时为错误信息 |
| 输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 用于初始化嵌入式控制板 | | | |
| 备注 | 在使用其他功能前,需先调用该函数（确保配置在当前目录的EtcConfig文件夹下），会根据配置文件自动连接服务器和登录。 | | | |

**参数详细说明：**

1. 参数pResult说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| IpAddr | string | string | 嵌入式设备IP地址 |
| Port | string | int | 嵌入式设备监听端口 |
| TimeOut | string | int | 超时时间(单位为秒) |
| HeartBeatTime | string | int | 心跳时间(单位为秒) |
| UserName | string | string | 用户名称 |
| PassWord | string | string | 用户密码 |
| ProvinceNo | string | string | 省份编码 |
| CityNo | string | string | 城市编码 |
| AreaNo | string | string | 区域编码 |
| GateNo | string | string | 大门编号 |
| LaneNo | string | string | 车道号 |
| RSUID | string | int | 天线ID |
| ReaderID | string | int | 读卡器ID |
| AutoLogout | string | int | 是否自动登出 |

### 4.1.2.反初始化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 反初始化 | | | |
| 函数原型 | int \_\_stdcall Uninit(void); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 反请求结果  0：反初始化成功  其他：失败 | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 功能 | 用于反初始化嵌入式控制板 | | | |
| 备注 | 退出时调用该函数，自动断开连接 | | | |

## 4.2.天线控制

### 4.2.1.读OBU接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 读OBU | | | |
| 函数原型 | int \_\_stdcall RSURead(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 当天线搜素到OBU号,会将OBU信息通过嵌入式控制子系统主动发送给小区车道系统。 | | | |
| 备注 | 当调用该函数会处于阻塞状态,直到天线搜索到OBU。需继续搜索时,需重新调用该接口。 | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| TimeOut | string | 超时时间（单位：ms） |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| OBUID | string | string | OBUID,对应的16进制字符串（8位）。 |
| OBUNO | string | string | OBU序列号,对应的16进制字符串（16位）。 |
| CardNo | string | string | 卡表面号含网络编号,对应的16进制字符串（20位）。 |
| CardType | string | string | 卡片类型 |
| CardFirTime | string | string | 卡片启用时间 |
| CardExpiryTime | string | string | 卡片到期时间 |
| OBUFirTime | string | string | OBU启用日期 |
| OBUExpiryTime | string | string | OBU到期日期 |
| Balance | string | long | 卡片余额（单位：分） |
| CardPlate | string | string | 车牌号,全牌照 (汉字+字母+数字,从卡片0015文件中获取,不超过12位） |
| CardPlateColor | string | int | 车牌颜色,0：蓝色1：黄色2：黑色3：白色 其他：保留（从卡片0015文件中获取） |
| CardVehClass | string | int | 车型（从卡片0015文件中获取,不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义  11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| CardVehUserType | string | int | 车辆用户类型（从卡片0015文件中获取,不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OBUPlate | string | string | 车牌号,全牌照 (汉字+字母+数字,从OBU中获取,不超过12位） |
| OBUPlateColor | string | int | 车牌颜色,0：蓝色1：黄色2：黑色3：白色 其他：保留（从OBU中获取） |
| OBUVehClass | string | int | 车型（从OBU中获取,不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义 11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| OBUVehUserType | string | int | 车辆用户类型（从OBU中获取,不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OBUVehSize | string | string | 车辆尺寸（从OBU中获取,长[2字节可表示的整数] X 宽[1字节可表示的整数] X 高[1字节可表示的整数]）,单位：dm。三个数字用“X”分隔。 |
| OBUWheelNo | string | int | 车轮数（从OBU中获取,不超过1字节整数的表示范围） |
| OBUAxieNo | string | int | 车轴数（从OBU中获取,不超过1字节整数的表示范围） |
| OBUWheelbase | string | int | 轴距,单位：dm（从OBU中获取,不超过2字节整数的表示范围） |
| OBUWeight | string | int | 车辆载重/座位数,其中,载重的单位为：kg（从OBU中获取,不超过4字节整数的表示范围） |
| CardAreaNo | string | string | 入/出口小区编码（从卡片001A文件中获取,4位） |
| CardGateNo | string | string | 入/出口大门编码（从卡片001A文件中获取,2位） |
| CardLaneNo | string | string | 入/出口车道编码（从卡片001A文件中获取,2位） |
| PassTime | string | string | 入/出口时间（从卡片001A文件中获取,yyyymmddhhmmss） |
| VehPlate | string | string | 入/出口车牌（从卡片001A文件中获取,不超过12位） |
| VehType | string | int | 入/出口车型（从卡片001A文件中获取,不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义  11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| VehClass | string | int | 入/出口车种（从卡片001A文件中获取,不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OutFlag | string | int | 入/出口标识（从卡片001A文件中获取）  1：已做出口处理  0：未做出口处理 |
| OperatorNo | string | char\* | 入/出口收费员工号（从卡片001A文件中获取,6位） |
| LittleGateNo | string | string | 小园小门编码(国标卡不需要此项) |
| LittleLaneNo | string | string | 小园车道编码(国标卡不需要此项) |
| LittlePassTime | string | string | 小园通过日期时间(国标卡不需要此项) |
| LittleCashMoney | string | string | 小园累计金额(国标卡不需要此项) |
| LittleTime | string | string | 小园累计时间(国标卡不需要此项) |
| OfferType | string | string | 优惠类型(国标卡不需要此项) |
| OfferTime | string | string | 优惠时间(国标卡不需要此项) |
| BackUp | string | string | 备用(国标卡不需要此项) |
| CheckCode | string | string | 校验码(国标卡不需要此项) |

### 4.2.2天线写出入口信息和消费接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 天线写出入口信息和消费 | | | |
| 函数原型 | int \_\_stdcall RSUWrite(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 用于小区车道系统向嵌入式控制子系统请求写出入口信息到卡片中并进行扣款操作。对于入口车道,需写入入口信息和消费0元,对于出口车道,须写入出口信息,并进行消费。 | | | |
| 备注 |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| OBUID | string | OBUID,对应的16进制字符串（8位）。 |
| OBUNO | string | OBU序列号,对应的16进制字符串（16位）。 |
| CardNo | string | 卡表面号含网络编号,对应的16进制字符串（20位）。 |
| CashMoney | string | 消费金额（入口为0,单位：分,不超过4字节整数表示的范围） |
| CardAreaNo | string | 入/出口小区编码（从卡片001A文件中获取,4位） |
| CardGateNo | string | 入/出口大门编码（从卡片001A文件中获取,2位） |
| CardLaneNo | string | 入/出口车道编码（从卡片001A文件中获取,2位） |
| PassTime | string | 入/出口时间（yyyymmddhhmmss） |
| VehPlate | string | 入/出口车牌（不超过12位） |
| VehType | string | 入/出口车型（不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义  11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| VehClass | string | 入/出口车种（不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OutFlag | string | 入/出入标识,入口填0,出口填1  0：未做出口处理  1：已做出口处理 |
| OperatorNo | string | 入/出口收费员工号（6位） |
| LittleGateNo | string | 小园小门编码(国标卡不需要此项) |
| LittleLaneNo | string | 小园车道编码(国标卡不需要此项) |
| LittlePassTime | string | 小园通过日期时间(国标卡不需要此项) |
| LittleCashMoney | string | 小园累计金额(国标卡不需要此项) |
| LittleTime | string | 小园累计时间(国标卡不需要此项) |
| OfferType | string | 优惠类型(国标卡不需要此项) |
| OfferTime | string | 优惠时间(国标卡不需要此项) |
| BackUp | string | 备用(国标卡不需要此项) |
| CheckCode | string | 校验码(国标卡不需要此项) |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| KeyServiceNo | string | string | 密钥服务流水（30位） |
| TradeType | string | int | 消费类型  6:普通消费  9:复合消费 |
| TermTradeNo | string | string | 终端交易序列号 |
| CardTradeNo | string | string | 卡片交易序列号 |
| TermCode | string | string | PSAM终端号 |
| Tac | string | string | 校验数据（不超过30位） |

## 4.3.读卡器控制

### 4.3.1.获取卡片信息接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 获取卡片信息 | | | | |
| 函数原型 | int \_\_stdcall ReaderRead(int iLaneNo,const char\* pReg, char\* pResp, int\* pLen); | | | | |
| 返回值 | | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | | 用于小区车道系统向嵌入式控制子系统请求读取指定的卡号信息。嵌入式控制子系统控制读卡器读取卡片信息,并将信息返回给小区车道系统。 | | | |
| 备注 | |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| TimeOut | string | 超时时间（单位：ms） |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| CardNo | string | string | 卡号（不超过20位） |
| Balance | string | long | 卡片余额（单位：分） |
| CardPlate | string | string | 车牌号,全牌照 (汉字+字母+数字,从卡片0015文件中获取,不超过12位） |
| CardPlateColor | string | int | 车牌颜色,0：蓝色1：黄色2：黑色3：白色 其他：保留（从卡片0015文件中获取） |
| CardVehClass | string | int | 车型（从卡片0015文件中获取,不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义  11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| CardVehUserType | string | int | 车辆用户类型（从卡片0015文件中获取,不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| CardAreaNo | string | int | 入/出口小区编码（从卡片001A文件中获取,4位） |
| CardGateNo | string | int | 入/出口大门编码（从卡片001A文件中获取,2位） |
| CardLaneNo | string | int | 入/出口车道编码（从卡片001A文件中获取,2位） |
| PassTime | string | string | 入/出口时间（从卡片001A文件中获取,yyyymmddhhmmss） |
| VehPlate | string | string | 入/出口车牌（从卡片001A文件中获取,不超过12位） |
| VehType | string | int | 入/出口车型（从卡片001A文件中获取,不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义 11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| VehClass | string | int | 入/出口车种（从卡片001A文件中获取,不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OutFlag | string | int | 入/出口标识（从卡片001A文件中获取）  1：已做出口处理  0：未做出口处理 |
| OperatorNo | string | string | 入/出口收费员工号（从卡片001A文件中获取,6位） |
| LittleGateNo | string | int | 小园小门编码(国标卡不需要此项) |
| LittleLaneNo | string | int | 小园车道编码(国标卡不需要此项) |
| LittlePassTime | string | string | 小园通过日期时间(国标卡不需要此项) |
| LittleCashMoney | string | long | 小园累计金额(国标卡不需要此项) |
| LittleTime | string | string | 小园累计时间(国标卡不需要此项) |
| OfferType | string | string | 优惠类型(国标卡不需要此项) |
| OfferTime | string | string | 优惠时间(国标卡不需要此项) |
| CheckCode | string | string | 校验码(国标卡不需要此项) |

### 4.3.2读卡器写出入口信息和消费接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 读卡器写出入口信息和消费 | | | |
| 函数原型 | int \_\_stdcall ReaderWrite(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 用于小区车道系统向嵌入式控制子系统请求写出入口信息到卡片中并进行扣款操作。对于入口车道,需写入入口信息和消费0元,对于出口车道,须写入出口信息,并进行消费。 | | | |
| 备注 |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| CardNo | string | 卡表面号含网络编号,对应的16进制字符串（20位）。 |
| CashMoney | string | 扣款金额（入口为0,单位：分,不超过4字节整数表示的范围） |
| CardAreaNo | string | 入/出口小区编码 |
| CardGateNo | string | 入/出口大门编码 |
| CardLaneNo | string | 入/出口车道编码 |
| PassTime | string | 入/出口时间（yyyymmddhhmmss） |
| VehPlate | string | 入/出口车牌（不超过12位） |
| VehType | string | 入/出口车型（不超过1字节整数的表示范围）  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义  11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| VehClass | string | 入/出口车种（不超过1字节整数的表示范围）  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| OutFlag | string | 入/出口标识,入口填0,出口填1  0：未做出口处理  1：已做出口处理 |
| OperatorNo | string | 入/出口收费员工号（6位） |
| LittleGateNo | string | 小园小门编码(国标卡不需要此项) |
| LittleLaneNo | string | 小园车道编码(国标卡不需要此项) |
| LittlePassTime | string | 小园通过日期时间(国标卡不需要此项) |
| LittleCashMoney | string | 小园累计金额(国标卡不需要此项) |
| LittleTime | string | 小园累计时间(国标卡不需要此项) |
| OfferType | string | 优惠类型(国标卡不需要此项) |
| OfferTime | string | 优惠时间(国标卡不需要此项) |
| BackUp | string | 备用(国标卡不需要此项) |
| CheckCode | string | 校验码(国标卡不需要此项) |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| KeyServiceNo | string | string | 密钥服务流水（30位） |
| TradeType | string | int | 消费类型  6:普通消费  9：复合消费 |
| TermTradeNo | string | string | 终端交易序列号 |
| CardTradeNo | string | string | 卡片交易序列号 |
| TermCode | string | string | PSAM终端号 |
| Tac | string | string | 校验数据（不超过30位） |

## 4.4.流水处理

### 4.4.1流水上传接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 流水上传 | | | | |
| 函数原型 | int \_\_stdcall ListUpLoad(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen); | | | | |
| 返回值 | | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | | 用于小区车道系统将出入口流水发送给嵌入式控制子系统,嵌入式控制子系统接收流水,并将流水上传至后台服务子系统。 | | | |
| 备注 | | 请自行对上传失败的流水做持久化处理 | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| ListType | string | 流水类型  0：入口流水  1：出口流水  2：咪表消费流水  3：非现金支付流水（使用非现金支付的出入口流水、月报、物业等费用都归到此处）。 |
| ListNo | string | 流水号（28位）：省份编号（2位）+城市编号（2位）+小区编号（4位）+大门编号（2位）+车道编号（2位）+时间（yyyymmddhhmmss,14位）+顺序号（2位）。 |
| KeyServiceNo | string | 密钥服务流水,进行消费后返回的密钥服务流水。（30位）  当ListType=3时，为对应的非现金支付流水号。 |
| TradeType | string | 消费类型  6:普通消费  9：复合消费  当ListType=3时，该项为空。 |
| TermCode | string | PSAM终端编码  当ListType=3时，该项为空。 |
| TermTradeNo | string | 终端交易序列号  当ListType=3时，该项为空。 |
| CardTradeNo | string | 卡片交易序列号  当ListType=3时，该项为空。 |
| Tac | string | TAC码  当ListType=3时，该项为空。 |
| OBUID | string | OBU号  当ListType=3时，该项为空。 |
| OBUNO | string | OBU应用序列号  当ListType=3时，该项为空。 |
| CardNo | string | CPU卡表面号  当ListType=3时，该项为空。 |
| CashMoney | string | 消费金额（入口为0,单位：分,不超过4字节整数表示的范围） |
| Balance | string | 卡片余额（单位：分，消费后的余额）  当ListType=3时，该项为空。 |
| TradeDevice | string | 交易设备  0:使用RSU交易  1:使用读卡器交易  2：使用咪表终端交易  3：ETC卡支付  4：银联支付  5微信支付  6：支付宝支付 |
| VehPicture | string | 车辆图片(Base64编码),无车辆图片为NULL。 |
| VehPictureLen | string | 图片长度 |
| SquadDate | string | 工班日期（yyyymmdd） |
| ShiftID | string | 收费员班次 |
| ExTime | string | 出口时间（yyyymmddhhmmss）,入口为NULL。 |
| ExAreaNo | string | 出口小区编码（4位） |
| ExGateNo | string | 出口大门编码(2位)。 |
| ExLaneNo | string | 出口车道编码(2位)。 |
| ExOperatorNo | string | 出口收费员工号,入口为NULL。（6位） |
| ExVehPlate | string | 出口车牌,入口为NULL。（不超过12位） |
| ExVehType | string | 出口车型,入口为255。(不超过1字节整数的表示范围)  1：一型车2：二型车3：三型车4：四型车:5：五型车6：六型车7-10自定义  11-20：用于计重收费货车车型分类:11：一型车12：二型车:13：三型车:14：四型车15：五型车:16：六型车:17-20：自定义 |
| ExVehClass | string | 出口车种,入口为255。（不超过1字节整数的表示范围)  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| EnTime | string | 入口时间（yyyymmddhhmmss）,出口时从卡片读取。 |
| EnOperatorNo | string | 入口收费员工号,出口时从卡片读取。（6位） |
| EnAreaNo | string | 入口小区编码（4位） |
| EnGateNo | string | 入口大门编码 (2位) |
| EnLaneNo | string | 入口车道编码 (2位)。 |
| EnVehPlate | string | 入口车牌,出口时从卡片入口信息文件读取。（不超过12位） |
| EnVehType | string | 入口车型,出口时从卡片入口信息文件读取。（不超过1字节整数的表示范围)  1：一型车2：二型车3：三型车4：四型车:5：五型车6：六型车7-10自定义  11-20：用于计重收费货车车型分类:11：一型车12：二型车:13：三型车:14：四型车15：五型车:16：六型车:17-20：自定义 |
| EnVehClass | string | 入口车种,出口时从卡片入口信息文件读取。（不超过1字节整数的表示范围)  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |

### 4.4.2流水查询接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 流水查询 | | | | |
| 函数原型 | int \_\_stdcall ListQuery(int iLaneNo,const char\* pReg,char\* pResp, int\* pLen); | | | | |
| 返回值 | | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | | 用于小区车道系统将流水号发送给嵌入式控制子系统,嵌入式控制子系根据流水号到后台检索对应的流水信息,并将流水返回给小区车道系统。 | | | |
| 备注 | |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| ListNo | string | 流水号 |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| ListType | string | int | 流水类型  0：入口流水  1：出口流水  2：咪表消费流水 |
| KeyServiceNo | string | string | 密钥服务流水,进行消费后返回的密钥服务流水。（30位） |
| TradeType | string | int | 消费类型  6:普通消费  9：复合消费 |
| TerminalNo | string | string | 设备终端编码,对于停车场应用为嵌入式终端设备号,对于咪表应用为咪表终端号。（32位）,缺省时填NULL。 |
| TermCode | string | string | PSAM终端编码 |
| TermTradeNo | string | string | 终端交易序列号 |
| CardTradeNo | string | string | 卡片交易序列号 |
| Tac | string | string | TAC校验数据（不超过30位） |
| OBUID | string | string | OBU号 |
| OBUNO | string | string | OBU应用序列号 |
| CardNo | string | string | CPU卡表面号 |
| CashMoney | string | long | 消费金额（入口为0,单位：分,不超过4字节整数表示的范围） |
| Balance | string | long | 卡片余额（单位：分，消费后的余额） |
| TradeDevice | string | long | 交易设备  0:使用RSU交易  1:使用读卡器交易  2：使用咪表终端交易 |
| VehPicture | string | string | 车辆图片,无图片时为NULL。 |
| VehPictureLen | string | long | 图片长度 |
| SquadDate | string | string | 工班日期（yyyymmdd） |
| ShiftID | string | string | 收费员班次 |
| ExTime | string | string | 出口时间（yyyymmddhhmmss）,入口为NULL。 |
| ExAreaNo | string | int | 出口小区编码（4位） |
| ExGateNo | string | int | 出口大门编码(2位)。 |
| ExLaneNo | string | int | 出口车道编码(2位)。 |
| ExOperatorNo | string | string | 出口收费员工号,入口为NULL。（6位） |
| ExVehPlate | string | string | 出口车牌,入口为NULL。（不超过12位） |
| ExVehType | string | int | 出口车型,入口为255。(不超过1字节整数的表示范围)  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义 11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| ExVehClass | string | int | 出口车种,入口为255。（不超过1字节整数的表示范围)  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |
| EnTime | string | string | 入口时间（yyyymmddhhmmss）,出口时从卡片读取。 |
| EnOperatorNo | string | string | 入口收费员工号,出口时从卡片读取。（6位） |
| EnAreaNo | string | int | 入口小区编码（4位） |
| EnGateNo | string | int | 入口大门编码 (2位) |
| EnLaneNo | string | int | 入口车道编码 (2位)。 |
| EnVehPlate | string | string | 入口车牌,出口时从卡片入口信息文件读取。（不超过12位） |
| EnVehType | string | int | 入口车型,出口时从卡片入口信息文件读取。（不超过1字节整数的表示范围)  1：一型车2：二型车3：三型车 4：四型车:5：五型车 6：六型车 7-10自定义 11-20：用于计重收费货车车型分类:11：一型车 12：二型车:13：三型车:14：四型车 15：五型车:16：六型车:17-20：自定义 |
| EnVehClass | string | int | 入口车种,出口时从卡片入口信息文件读取。（不超过1字节整数的表示范围)  0：普通车:6：公务车8：军警车10紧急车12免费车14车队0-20内其他：自定义；21-255：保留 |

## 4.5.设备管理

### 4.5.1设备状态查询接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 设备状态查询 | | | |
| 函数原型 | int \_\_stdcall ETCStatusQuery(int iLanNo, const char\* pReg, char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 用于小区车道系统查询嵌入式控制子系统的设备状态信息,包括天线设备状态信息、读卡器设备状态信息和嵌入式控制设备状态信息。 | | | |
| 备注 |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| Connection | string | int | 连接状态，0表示正常 |
| HeartBeat | string | int | 心跳状态，0表示正常，其他参考附录A |
| CardReader | object/null | - | 读卡器状态，null表示没有可用信息，类型为object时见下文 |
| RSU | object/null | - | 天线状态，null表示没有可用信息，类型为object时见下文 |



(3) CardReader/RSU说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| Status | string | int | 读卡器/天线当前状态  0：正常  -1：异常 |
| LastTime | string | string | 读卡器/天线状态更新时间，格式如下：  2016-12-13 14:43:22.811 |
| ErrorMsg | string | string | 错误信息 |

## 4.6.非现金接口

### 4.6.1非现金支付请求接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 发送非现金支付请求 | | | |
| 函数原型 | int \_\_stdcall SendNonCashPay(int iLaneNo, const char\* pReq, char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 若调用该接口时已进入非现金支付状态，则会清除前次的非现金支付请求信息。 | | | |
| 备注 |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| ScannerID | string | 扫码枪1:0  扫码枪2:1 |
| UserName | string | 用户名（不超过32位） |
| ProvinceNo | string | 省份编码,按照GB/T 2260（2位） |
| CityNo | string | 城市编码,按照GB/T 2260（2位） |
| AreaNo | string | 小区编码（4位） |
| GateNo | string | 大门编码（2位） |
| LaneNo | string | 车道编码（2位） |
| NonCashTsSn | string | 非现金支付流水号，整个交易过程中保持一致。流水号（28位）：省份编号（2位）+城市编号（2位）+小区编号（4位）+大门编号（2位）+车道编号（2位）+收费时间（yyyymmddhhmmss,14位）+顺序号（2位）. |
| CashMoney | string | 支付金额,单位：分,不超过4字节整数表示的范围 |
| CardNo | string | 卡号，可以为ETC卡卡号、临时卡、月保卡卡号，20位，不足20位的，高位补0. |
| RequestTime | string | 支付请求时间（yyyymmddhhmmss），若使用ETC卡支付，该时间作为密钥计算的参数。 |
| VehPlate | string | 车牌，不超过12位 |
| ChargeType | string | 收费类型：  1：停车费 |
| ChargeDetail | string | 收费项目特殊说明，可为空。 |
| Timeout | string | 超时时间（单位：ms） |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |
| CashMoney | string | long | 实收金额,单位：分,不超过4字节整数表示的范围 |
| PayType | string | int | 支付方式  3：ETC卡支付  其他待补充 |
| PayTime | string | string | 支付时间 |

### 4.6.2非现金支付结束接口

消息方向：小区车道系统→动态库

说明：该接口用于小区车道系统向嵌入式控制系统请求结束非现金支付流程，嵌入式控制系统返回操作结果。若正在进行非现金支付操作，则返回结束失败。

参数内容包括：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 非现金支付结束 | | | |
| 函数原型 | int \_\_stdcall StopNonCashPay(int iLaneNo,const char\* pReq, char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 | 若调用该接口时已进入非现金支付状态，则会清除前次的非现金支付请求信息。 | | | |
| 备注 |  | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| NonCashTsSn | string | 非现金支付流水号，整个交易过程中保持一致。流水号（28位）：省份编号（2位）+城市编号（2位）+小区编号（4位）+大门编号（2位）+车道编号（2位）+收费时间（yyyymmddhhmmss,14位）+顺序号（2位）. |

(2)参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| ErrorCode | string | int | 错误码,0表示成功,其它值含义 详见 **附录A** |

### 4.6.3交易明细查询接口

消息方向：小区车道系统→动态库

说明：该接口用于小区车道系统向嵌入式控制系统请求交易明细查询，嵌入式控制系统返回查询结果。

参数内容包括：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 交易明细查询 | | | |
| 函数原型 | int \_\_stdcall QueryTransDetail(int iLaneNo,const char\* pReq, char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  1：设备正忙  2：操作超时  3：非法的请求参数  4：非法的回应  5：pResp长度不足  6：发送数据失败  1000：车道不存在  其他：参见附录A | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输入 | iLaneNo | int | 车道编号 |
| 输入 | pReq | const char\* | 发送给嵌入式控制系统的请求包 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 |  | | | |
| 备注 | 如果page=1，pagesize=50。则表示取最近50条。page=2，pagesize=30则表示取最近的31条至60条。 | | | |

**参数详细说明：**

(1)参数pReq说明(pReq为json字符串, 数据格式如下所述)

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| Page | string | 要查询的交易明细页 |
| PageSize | string \* | 一页交易明细页，包含的明细数目，最多为50条。 |

1. 参数pResp说明(pResp为json字符串,数据格式如下所述)

|  |  |  |  |
| --- | --- | --- | --- |
| 字段关键字 | 字段值源类型 | 字段值目标类型 | 字段所表示含义 |
| Result | json数组 | - | 查询结果，json数组形式，各项为各条流水，流水的字段见下表。 |
| ErrorCode | string | 错误码,0表示成功,其它值含义 详见 **附录A** |  |

1. 流水说明

|  |  |  |
| --- | --- | --- |
| 字段关键字 | 字段值类型 | 字段所表示含义 |
| NonCashTsSn | string | 非现金支付流水号，整个交易过程中保持一致。流水号（28位）：省份编号（2位）+城市编号（2位）+小区编号（4位）+大门编号（2位）+车道编号（2位）+收费时间（yyyymmddhhmmss,14位）+顺序号（2位）. |
| UserName | string | 用户名（不超过32位） |
| ProvinceNo | string | 省份编码,按照GB/T 2260（2位） |
| CityNo | string | 城市编码,按照GB/T 2260（2位） |
| AreaNo | string | 小区编码（4位） |
| GateNo | string | 大门编码（2位） |
| LaneNo | string | 车道编码（2位） |
| KeyServiceNo | string | 对于ETC卡交易为密钥服务流水号、对于微信支付为微信订单号。 |
| CashMoney | string | 支付金额,单位：分,不超过4字节整数表示的范围 |
| CardNo | string | 卡号，可以为ETC卡卡号、临时卡、月保卡卡号，20位，不足20位的，高位补0. |
| RequestTime | string | 支付请求时间 |
| VehPlate | string | 车牌，不超过12位 |
| ChargeType | string | 收费类型：  1：停车费  2：物管费  3：月保费  4：其他 |

|  |  |  |
| --- | --- | --- |
| ChargeDetail | string | 收费项目特殊说明，可为空。 |
| PayType | string | 支付方式  3：ETC卡支付  4：银联支付  5微信支付  6：支付宝支付 |
| PayTime | string | 支付时间 |
| PayResult | string | 支付结果  0：成功  -1：余额不足  -2：ETC卡黑名单  其他：对应的错误码 |
| CreateFlag | string | 生成标志 |
| ListStatus | string | 流水状态 |
| ETCCardNo | string | ETC卡卡号，使用ETC卡支付时，该项不能为空。 |
| Balance | string | 卡片余额（单位：分，消费后的余额），使用ETC卡支付时，该项不能为空。 |
| TermCode | string | PSAM终端编码，使用ETC卡支付时，该项不能为空。 |
| TermTradeNo | string | 终端交易序列号，使用ETC卡支付时，该项不能为空。 |
| CardTradeNo | string | 卡片交易序列号，使用ETC卡支付时，该项不能为空。 |
| Tac | string | 校验码，使用ETC卡支付时，该项不能为空。 |

## 4.7.获取当前线程上一次输出

当前面所述的接口返回5时，表示缓冲区大小不足，需要调用此接口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | 获取当前线程上一次输出 | | | |
| 函数原型 | int \_\_stdcall GetResponse(int\* pErrorCode, char\* pResp, int\* pLen); | | | |
| 返回值 | 返回值类型 | | 返回值说明 | |
| int | | 请求结果  0：操作成功  3：无信息  5：pResp长度不足 | |
| 参数 | 出/入 | 参数名称 | 类型 | 含义 |
| 输出 | pErrorCode | int\* | 上一次的错误码 |
| 输出 | pResp | char\* | 嵌入式控制系统返回的回应包 |
| 输入/输出 | pLen | int\* | 传入时，指针指向的值应为pResult缓冲区的可容纳的字节数，以防止越界。  当接口返回5时，\*pLen会被改写为实际所需大小，用户可根据此大小调用GetResponse接口 |
| 功能 |  | | | |
| 备注 | 基于可重入性和防止数据丢失的考虑，该接口以线程为key保存输出信息，因此需要在动态库接口返回5时在同一线程立即调用此接口。 | | | |

# 5.调用流程说明

## 5.1.天线交易流程(入口)

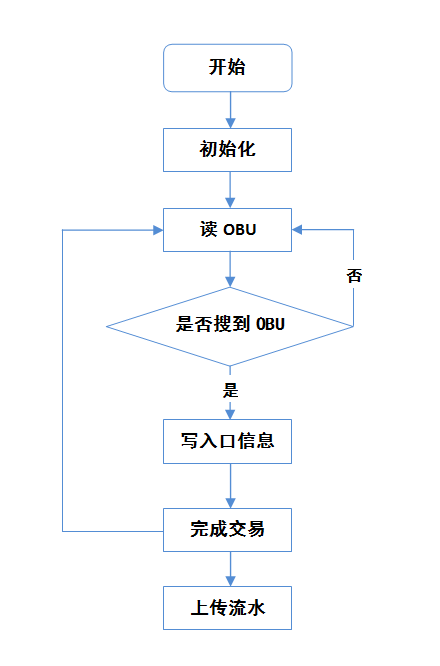


图 1天线交易流程（入口）

## 5.2.读卡器交易流程(入口)

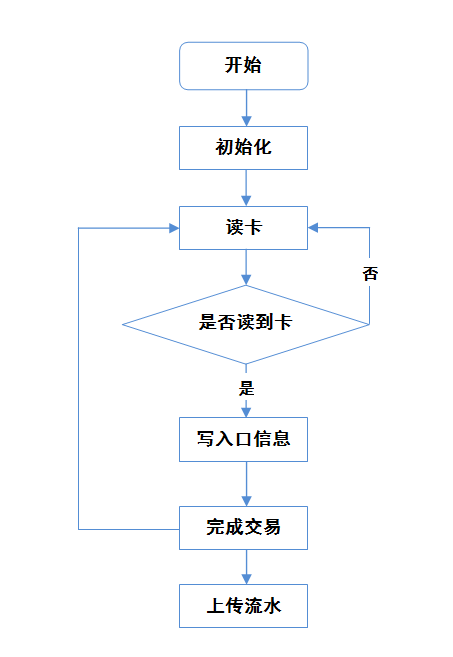


图 2读卡器交易流程（入口）

## 5.3.天线交易流程(出口)

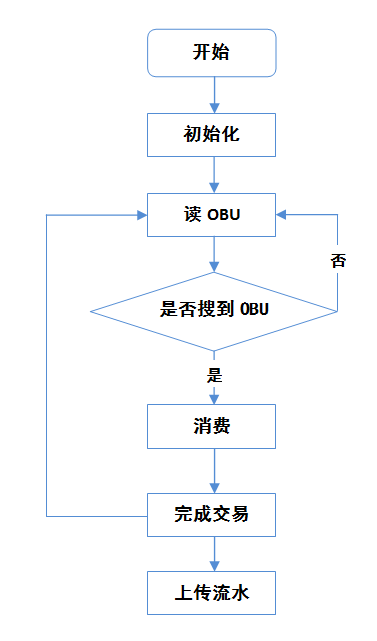


图 3天线交易流程（出口）

## 5.4.读卡器交易流程(出口)

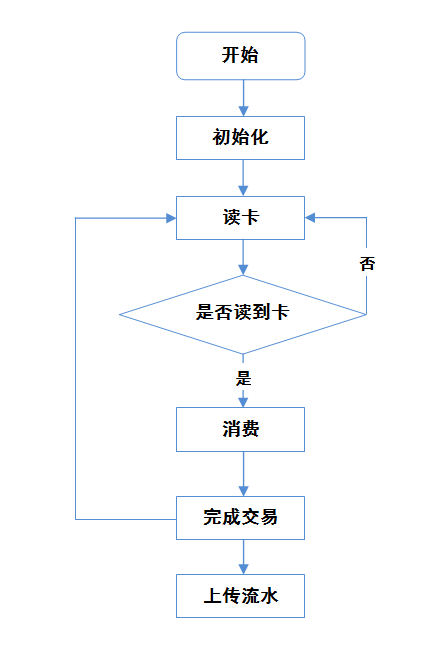


图 4读卡器交易流程（出口）

## 5.5.车卡绑定校验流程

若车道启用了车卡绑定限制，当进行读卡器消费时，需按以下流程执行：



## 5.6.非现金支付流程



# 附录A

## A.1.天线相关错误码说明

|  |  |
| --- | --- |
| 错误码值 | 错误码所表示含义 |
| 0 | 表示操作成功 |
| -1100 | JSON数据类型匹配错误 |
| -1101 | 无效的JSON值 |
| -1102 | JSON解析错误 |
| -1300 | 天线未打开 |
| -1301 | 天线打开串口失败 |
| -1302 | 天线调整功率失败 |
| -1303 | 天线关闭失败 |
| -1304 | 搜索OBU失败 |
| -1305 | 读0015文件失败 |
| -1306 | 之前读取到的卡号与车道给的卡号不一致 |
| -1307 | 查询余额失败 |
| -1308 | 读001a失败 |
| -1309 | 001a文件长度不匹配（参数不正确） |
| -1310 | 消费失败 |
| -1311 | ESAM选择目录失败 |
| -1312 | ESAM都EF1D文件失败 |
| -1313 | CPU卡选择目录失败 |
| -1314 | CPU卡写卡失败 |
| -1315 | 广东消费失败 |
| -1316 | 读取的卡片或者标签与 |
| -1317 | 不是记账卡或者储值卡 |
| -1318 | 卡片或者OBU过期 |
| -1319 | 卡签不一致 |
| -1320 | 获取设备MAC失败 |
| -1321 | 后台链接错误 |
| -1322 | web返回错误 |
| -1323 | 黑名单校验不通过 |
| -1324 | 消费初始化失败 |
| -1325 | 获取TAC失败 |
| -1326 | 时间格式错误 |
| -1327 | 不是十六进制字符串 |
| -1328 | 设备超时错误 |
| -1329 | 获取OBU密文失败 |
| -1330 | 国标消费失败 |
| -1331 | 非法拆卸 |
| -1332 | 消费模式不存在 |
| -1333 | 数据库操作错误 |

## A.2.读卡器相关错误码说明

|  |  |
| --- | --- |
| 错误码值 | 错误码所表示含义 |
| 0 | 表示操作成功 |
| -2100 | JSON数据类型匹配错误 |
| -2101 | 无效JSON值 |
| -2100 | JSON解析错误 |
| -2300 | 读卡器未开启 |
| -2301 | 读卡器初始化失败 |
| -2302 | 读卡器关闭失败 |
| -2303 | 打开卡片失败 |
| -2304 | 寻卡失败 |
| -2305 | 读0015失败 |
| -2306 | 卡号不匹配 |
| -2307 | 查询余额失败 |
| -2308 | 未知卡片类型 |
| -2309 | 读001A失败 |
| -2310 | 001A长度不匹配 |
| -2311 | 保留 |
| -2312 | 卡类型错误 |
| -2313 | 卡过期 |
| -2314 | 写001A失败 |
| -2315 | GSOAP调用失败 |
| -2316 | 卡无效 |
| -2317 | 卡在黑名单内 |
| -2319 | 时间格式错误 |
| -2320 | 半条流水校验失败 |
| -2321 | 正在进行非现金交易 |
| -2400 | 消费失败 |
| -2401 | 消费失败（可能存在半条流水） |

## A.3.业务进程相关错误码说明

|  |  |
| --- | --- |
| 错误码值 | 错误码所表示含义 |
| 0 | 表示操作成功 |
| -3001 | 设备终端与车道不存在绑定关系 |
| -3002 | 设备终端号不存在或者设备处于禁用状态 |
| -3003 | 车道用户的登录密码错误 |
| -3004 | 车道用户不存在或车道用户处于禁用状态 |
| -3005 | 车道号不存在或车道号处于禁用状态 |
| -3006 | 车道用户不存在，车道用户退出失败 |
| -3007 | 未查询到相应流水 |
| -3008 | 流水类型错误 |
| -3009 | TokenId失效（心跳超时或者已经退出登录） |
| -3010 | 黑名单或者过期 |
| -3012 | 车道用户没有消费权限 |
| -3333 | 访问后台超时 |
| -3334 | JSON解析错误 |
| -3335 | 操作码不存在 |

## A.4.非现金进程相关错误码说明

|  |  |
| --- | --- |
| 错误码值 | 错误码所表示含义 |
| 0 | 表示操作成功 |
| -4001 | 嵌入端程序响应超时 |
| -4002 | 后台响应超时 |
| -4003 | 后台与嵌入段都超时 |