

# Q B

## 深圳市地铁集团有限公司企业标准

QB/SZMC-20128.3-2013

代替 QB/SZMC-20128-2011 中第二册：AFC 系统接口技术规范

---

### 自动售检票系统技术标准 第三部分：接口技术规范

Technical Code of Automatic Fare Collection System—

Part 3: Interface Technical Specifications

---

2013-12-10 发布

2013-12-10 实施

深圳市地铁集团有限公司

发 布



# 深圳市地铁集团有限公司文件

深地铁〔2013〕953号

---

## 深圳市地铁集团有限公司关于印发 《自动售检票系统技术标准》（修订版）的通知

集团各部（室）、各分（子）公司：

《自动售检票系统技术标准 第一至第五部分》编号为QB/SZMC-20128.1~.5-2013，经集团公司审批，现予以颁布。本标准代替原标准《自动售检票系统企业技术标准》（QB/SZMC-20128-2011）中第一、二、三和五册（第四册将由后续颁布的其它标准替代）。

请各单位参照执行。在执行过程中，如有新的相关国家标准、行业标准、地方标准颁布或修订且其条款要求高于本企业标准，应执行较高要求的条款，同时，发现本标准需要修改和补充之处，请将意见和有关资料及时反馈至集团公司总工程师办公室。

特此通知。

附件：

1. 自动售检票系统技术标准 第一部分：业务规则  
( QB/SZMC-20128. 1-2013 )
2. 自动售检票系统技术标准 第二部分：系统和设备技术规范  
( QB/SZMC-20128. 2-2013 )
3. 自动售检票系统技术标准 第三部分：接口技术规范  
( QB/SZMC-20128. 3-2013 )
4. 自动售检票系统技术标准 第四部分：实施细则  
( QB/SZMC-20128. 4-2013 )
5. 自动售检票系统技术标准 第五部分：工程设计技术规范  
( QB/SZMC-20128. 5-2013 )

深圳市地铁集团有限公司

2013 年 12 月 9 日

## 前 言

本标准是根据深地铁重纪【2012】40号文的要求，由主编单位会同参编单位，并在国内相关各大集成商和设备厂商的参与下，对原QB/SZMC-20128-2011《自动售检票系统企业技术标准》进行全面修订而成。本次修订把原标准拆分为两个标准，即：本次颁布、五部分构成的QB/SZMC-20128. 1~5-2013《自动售检票系统技术标准》一代替原标准的第一、二、三、五册，和以后另行颁布的《自动售检票系统检测验收标准》一将代替原标准的第四册。

本标准用于规范深圳市地铁集团有限公司自动售检票系统项目的设计、招标及工程实施等项目实施环节。

本标准由深圳地铁集团有限公司批准、归口并负责解释。

本标准的修订主编单位、参编单位和主要起草人：

**主 编 单 位：**深圳市地铁集团有限公司

**参 编 单 位：**中铁二院工程集团有限责任公司

**主要起草人：**简 炼、褚技威、陈 琪、周世爽、周 聆、黄 宇、李果凤、饶庶民、梁金库、周 杰、夏劲松、白云海、彭 舸、赵 凯、王 欢、李瑗瑗、陈士润、刘 晶、谢华丽

**主要审查人：**吴 华、廖东玲、申香梅、庄运杰、陆 澜、马怀清、甘 灿、陈 宇、谷素斐、周 琪、骆海瑛、唐 斌、刘 乐、陈秀联、赵 宇、匡继承、丁 耿、周晓雄、刘光勇、孙 琰、申运华、房继尧

## 引 言

近十年来，深圳地铁得到了飞速发展，由2004年开通的深圳地铁一期（1号线罗宝线东段和4号线龙岗线南段）到2011年6月底前完成的地铁二期5条地铁线，且目前正在实施的地铁三期5条线（5年内陆续开通11、7、9、6和8号地铁线），未来，新的地铁线路还会继续扩增。这样的发展趋势，让我们意识到企业标准的制定/修订迫在眉睫。为此，启动修订《自动售检票系统企业技术标准》。

本次标准修订遵从本行业或相关行业同类产品的国家标准、行业标准及地方标准，并根据本企业的特点和要求做细化、深化和补充。修订工作是根据自动售检票系统近期工程建设及开通运营的实际经验，充分借鉴国内地铁有关成功经验和先进技术，经广泛调查和总结分析，并多次征询相关专家 and 单位的意见，反复论证和修改，最终以五部分构成的本标准 QB/SZMC-20128. 1~5-2013《自动售检票系统技术标准》呈现。新编这五部分是对原标准的系统及设备技术规范、工程设计规范、业务规则和接口规范做了深度细化和补充，并增加了实施细则规范。

自动售检票系统技术标准 第一部分：业务规则（QB/SZMC-20128. 1-2013）

自动售检票系统技术标准 第二部分：系统和设备技术规范（QB/SZMC-20128. 2-2013）

自动售检票系统技术标准 第三部分：接口技术规范（QB/SZMC-20128. 3-2013）

自动售检票系统技术标准 第四部分：实施细则（QB/SZMC-20128. 4-2013）

自动售检票系统技术标准 第五部分：工程设计技术规范（QB/SZMC-20128. 5-2013）

本标准明确了票务，维护维修等业务规则，规范了自动售检票系统的系统设计、专用设备设计、各个接口的设计以及关键性部件的设计与选型，提供了专用设备GUI、读写器和系统编码的实施细则并指导了AFC系统在现场工程的安装与布局。

本部分是标准的第三部分接口技术规范，明确了系统内部的接口规范。

## 缩 略 语

缩写词	英文解释	中文解释
ACC	AFC Clearing Center	轨道交通清分中心
ACL	Access Control List	访问控制列表
AFC	Automatic Fare Collection	自动售检票
AGM	Automatic Gate Machine	自动检票机
API	Application Programming Interface	应用编程接口
AVM	Add Value Machine	自动充值机
B/S	Browser/Server	浏览器/服务器
BIOS	Basic Input Output System	基本输入输出系统
BOM	Booking Office Machine	票务处理机/半自动售票机
CA	Certificate Authority	认证中心
CCC	China Compulsory Certification	中国强制认证
CF	Compact Flash	CF 卡
CLC	Cluster Line Center	线网中心
CSC	Contactless Smart Card	非接触式 IC 卡
DES	Data Encryption Standard	数据加密标准
ECU	Electronic Control Unit	电子控制单元
EMC	Electro Magnetic Compatibility	电磁兼容性
ES	Encoder and Sorter	编码分拣机
FAS	Fire Alarm System	火灾报警系统
FTP	File Transfer Protocol	文件传输协议
GPIO	General Purpose Input/Output	通用输入/输出
GPRS	General Packet Radio Service	通用分组无线服务技术
ISP	In System Program	在系统可编程

LAN	Local Area Network	局域网
LC	Line Center	线路中央
LCC	Line Center Computer	线路中央计算机系统
MAC	Message Authentication Code	消息认证码
MCBF	Mean Cycles Between Failure	平均无故障次数
MTBF	Mean-Time Between Failure	平均无故障时间
GMTT	Guided Mass Transit Ticket	轨道交通专用车票
MTTR	Mean Time To Repair	平均故障恢复维修时间
OCC	Operating Control Center	控制中心
P2P	Peer-to-Peer	点对点技术
PAC	Provisional Acceptance Certificate	预验收证书
PBOC	The People's Bank of China	中国人民银行
PCA	Portable Card Analyzer	便携式验票机
PIN	Personal Identification Number	个人身份标识码
QoS	Quality of Service	服务质量
RAID	Redundant Arrays of Inexpensive Disks	磁盘冗余阵列
RFID	Radio Frequency Identification	射频识别
R/W	Reader/Writer	读写器
SAM	Secure Access Module	安全存取模块
SAN	Storage Area Network	存储区域网络
SATA	Serial Advanced Technology Attachment	串行 ATA
SC	Station Computer	车站计算机
SCR	Station Control Room	车站控制室
SJT	Single Journey Ticket	单程票
SLA	Service-Level Agreement	服务等级协议
SLE	Station Level Equipment	车站终端设备
SNMP	Simple Network Management Protocol	简单网络管理协议



SSD	Solid State Disk	固态硬盘
SSH	Secure Shell	安全壳协议
SVT	Store Value Ticket	储值票
SZ	ShenZhen	深圳市
SZMC	Shenzhen Metro Group CO.,LTD.	深圳市地铁集团有限公司
TCM	Ticket Checking Machine	验票机
TCP	Transmission Control Protocol	传输控制协议
TCT	Trip Count Ticket	乘次票
TFT	Thin Film Transistor-Liquid Crystal Display	薄膜晶体管液晶显示器
TIM	Token Initialization Machine	单程票初始化编码机
TOKEN	Simplified CSC SJT	简易非接触式 IC 卡单程票
TSM	Token Sorting Machine	单程票清分机
TVM	Ticket Vending Machine	自动售票机
UPS	Uninterrupted Power Source	不间断电源
URL	Uniform Resource Locator	统一资源定位符
USB	Universal Serial BUS	通用串行总线
VGA	Video Graphics Array	视频图形阵列
VLAN	Virtual Local Area Network	虚拟局域网
VM	Virtual Machine	虚拟机
WiFi	Wireless Fidelity	基于 IEEE 802.11b 标准的无线局域网
WLAN	Wireless Local Area Network	无线局域网
MQ	Message Queue	消息队列
JMS	Java Message Service	Java 平台中关于面向消息中间件的 API
MD5	Message Digest Algorithm 5	计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护
DES	Data Encryption Standard	数据加密标准
3DES	3 Data Encryption Standard	三重数据加密标准

AES	Advanced Encryption Standard	高级加密标准
SSL	Secure Sockets Layer	安全套接层
QoS	Quality of Service	服务质量

深圳地铁 企业标准

## 专用术语及名词解释

下列定义和术语适用于本文件。

1. 便携式检验票机 portable card analyzer

可供工作人员对乘客使用的车票进行检票和验票的手持设备。

2. 车站计算机系统 station computer system

用于车站票务处理、运营管理的计算机系统。

3. 车站终端设备 station level equipment

安装于城市轨道交通线路各车站进行车票发售、进出站检票、充值、验票分析等车票交易处理的终端设备。

4. 城市轨道交通清分中心系统 guided mass transit AFC clearing center system

发行和管理城市轨道交通一票通车票，并对车票票款进行结算的系统，是实现城市轨道交通联网收费系统的核心。

5. 充值额度 recharge quota

充值额度为充值设备的最大可充值金额。

设备的充值额度为零或低于需充值金额时，该设备不执行充值。

6. 单程票编码机 token initialization machine

用于完成对系统单程票进行初始化编码的设备。

7. 单程票清分机 token sorting machine

安装于地铁各车站 AFC 的票务室，能够清点、分拣合法或非法的单程票的设备。

8. 读写器 Reader/Writer

未做特殊说明时，是指符合《票卡读写器标准》的用于处理车票的功能部件。

9. 非付费区 unpaid area

付费区以外的区域。

10. 付费区 paid area

由车站自动检票机、护栏围闭起的封闭区，包括候车站台和车厢。

11. 关键部件 critical component

在各终端设备中起关键作用的功能部件或模块。

关键部件至少包括：工控机、纸币处理模块、硬币处理模块、纸币找零模块、单程票回收机构、单程票发售机构、通行控制模块、闸门机构等。

12. 模拟大客流测试 test for virtual large passenger flow

组织人员模拟真实客流量，并以各类车票测试系统整体功能和性能。

13. 票务处理机 booking office machine

票务处理机也称为半自动售票机，是用于人工辅助发售车票，具备充值、异常处理、补票、退票、查询、更新等票务处理功能的设备。

14. 专用设备 special equipment

专用设备特指车站终端设备、以及相关的维护设备等。

15. 数据管理中心系统 data management center

是城市轨道交通线路 AFC 系统的运营商级数据管理中心，用于接受 ACC 系统参数及指令，完成与 ACC 对帐，统一完成运营商各线路的参数及报表输出的计算机系统。

16. 线路中央计算机系统 line center computer

用于管理和控制城市轨道交通线路自动售检票系统的计算机系统。

17. 线网中心系统 cluster line center

用于管理和控制城市轨道交通多条线路自动售检票系统的计算机系统集合。

18. 行政事务 administrative affairs

指城市轨道交通自动售检票（AFC）系统内与票务无关的相关事务，诸如：收取乘客行李票、自动售票机卡币退款、对乘客的违规行为进行处罚等。

#### 19. 压力测试 pressure test

选定设备/系统，组织人员、票卡及运营辅助工器具，在规定的时段内进行持续的规定操作，以检验设备/系统持续处理相关业务的能力。

#### 20. 自动充值机 adding value machine

可以对储值车票进行自助充值并具有查验交易和余额功能的设备。

#### 21. 自动检票机 automatic gate machine

可供乘客对城市轨道交通车票进行自助检票处理的设备。

#### 22. 自动检票机通道 automatic gate machine aisle

由两个自动检票机机壳安装构成的可供行人通行的区域，是自动检票机使用的基本单位，自动检票机的功能部件或模块安装在机壳表面或机壳内部。

#### 23. 自动检票机通道右机 right set

对于一个自动检票机通道，将行人由非付费区通过通道进入付费区时处于行人右手方向的机壳称为右机。

#### 24. 自动检票机通道左机 left set

对于一个自动检票机通道，将行人由非付费区通过通道进入付费区时处于行人左手方向的机壳称为左机，处于行人右手方向的机壳称为右机。

#### 25. 自动检票机通道组 row of automatic gate machine

连续排列安装的一组自动检票机通道称为一个通道组。

#### 26. 自动检票机通道组右边机 right end of one row

对于一个自动检票机通道组，将行人由非付费区通过通道进入付费区时处于行人右手方向该通道组最右边的机壳称为右边机。

27. 自动检票机通道组中间机 mid-terminal

对于连续排列安装的一组自动检票机通道，除左边机和右边机以外的机壳均称为中间机。

28. 自动检票机通道组左边机 left end of one row

对于一个自动检票机通道组，将行人由非付费区通过通道进入付费区时处于行人左手方向该通道组最左边的机壳称为左边机。

29. 自动售票机 ticket vending machine

用于自助发售车票，具备自动处理支付及找零功能的设备。

30. 自动售检票系统 automatic fare collection system

基于计算机、通信、网络、自动控制等技术，实现城市轨道交通售票、检票、计费、收费、统计、清算全过程的自动化系统。

31. 消息 message

在应用进程之间或一个应用内部交换的数据。

32. 队列 queue

消息的安全存放地，队列存储消息直到它被应用程序处理。

33. 面向消息的中间件 message oriented middleware

利用高效可靠的消息传递机制进行与平台无关的数据交流，并基于数据通信来进行分布式系统的集成。

34. 消息队列 message queue

用于应用程序之间的通信方法。

35. 队列管理器 queue manager

队列和其他对象的容器。

36. 消息通道 message channel

提供从一个队列管理器到另一个队列管理器的通信路径。

37. 死信队列 `dead letter queue`

存储无法发送到其正确目的地的消息队列。

38. 触发机制 `trigger mechanism`

一种自动启动应用程序的机制。

39. JMS

JAVA 消息服务应用程序接口，Java 平台中关于面向消息中间件（MOM）的 API。

40. 持久化 `persistence`

把数据保存到可永久保存的存储设备中。

41. SHA-2

一组由 SHA-224、SHA-256、SHA-384，和 SHA-512 组成的加密哈希函数集。

42. MD5

计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。

43. DES

一种使用密钥加密的块密码。

44. 3DES

三重数据加密算法。

45. AES

高级加密标准。

46. SSL

安全套接层。为网络通信提供安全及数据完整性的一种安全协议。

47. 无损压缩 `lossless compression`

数据经过压缩后，信息不受损失，还能完全恢复到压缩前的原样。

48. 非接触式轨道交通专用票读写器 contactless guided mass transit ticket reader/writer

用于城市轨道交通设备中，对城市轨道交通专用车票和深圳通卡、相关城市互通卡进行数据交换的终端设备，简称轨道交通票读写器。

49. 非接触式集成电路（IC）卡储值卡 contactless smart card store value ticket(csc svt)

一种 ID-1 型卡类型（如 ISO/IEC7810 中所规定），卡内嵌装集成电路，以非接触操作方式与外部集成电路进行耦合操作的卡。可反复充值使用。

50. 轨道交通专用票 guided mass transit ticket

符合本标准的城市轨道交通非接触式集成电路(IC)卡。

51. RFID-SIM

带 2.4GHZ 射频的手机 RFID-SIM 卡。

52. 防冲突 anti-confliction

多张卡同时处于非接触式 IC 卡轨道交通读写器的有效读写区域内时，读写设备可分别识别并做相应处理。

53. 密码 password

相当于通行字或口令、当一方面向另一方面提交出预先约定的密码时，递交一方的合法性才得以承认。

54. 卡内余额 SZT card balance

深圳通卡中，乘客实际可使用的金额（不包括押金部分）。

55. 最高单程票价 the one-way maxi-fare

指轨道交通线网基本票价表中的最大金额值。

56. 本站最高单程票价 the station maxi-fare

一个车站至线网其它车站的最高单程票价称为该车站的本站最高单程票价。



## 57. 票种最低票价 the card-type base fares

指乘客所使用车票种类的起步价。

## 58. 押金 deposit

深圳通公司向购买深圳通卡的乘客收取的车票抵押金。

## 59. 超时 time out

乘客在付费区内停留的时间超出规定的时间。

## 60. 超程 distance out

乘客乘坐的里程超出购买车票时预定的里程。

## 61. 异币 illegal currency

包括游戏机币、假币、外币及残币。

## 62. 收支两条线 separate management of revenue and expenditure

是指收入和支出分线管理，是企业提高内部资金管理效率的常用手段。

## 63. 账实相符 consistent accounts

是指报表台账记录与实物、款项实有数核对相符。

## 64. 坐支 cash

是指将收入的现金直接用于支出。

## 65. 正常交易 normal transaction

按 AFC 系统的统一交易数据格式生成并能正常入库的交易。

## 66. 异常交易 abnormal transaction

由于各种原因导致数据异常，无法正常入库的交易。

## 67. 灰交易 uncertain transaction

由于各种原因导致的写卡结果不确定的交易。

68. 寄存器 register

用于记录关键计数的软件或硬件实现。

69. TELNET

用于远程终端连接的标准 IP 协议。

70. 访问控制列表 access control list

是路由器和交换机接口的指令列表，用来控制端口进出的数据包。

71. 服务质量 quality of service

是网络的一种安全机制，用来解决网络延迟和阻塞等问题。

72. 服务等级协议 service level agreement

是关于网络服务供应商和客户间的一份合同，其中定义了服务类型、服务质量和客户付款等术语。

73. E1

是高速传输的另一种标准，一条 E1 可以同时有多个并发信道，每个信道都是一个独立的连接。

74. X86

X86 是由 Intel 推出的一种复杂指令集，用于控制芯片的运行程序。

75. 点对点技术 peer-to-peer

一种通信模式，每个参与者具有同等的能力，可以发起一个通信会话。

76. 即时通讯 instant messenger

是一种基于互联网的即时交流消息的业务。

77. 统一资源定位符 uniform resource locator

也被称为网页地址，是因特网上标准的资源的地址。

78. AV 特征库 anti-virus characteristic library

AV 特征库记录了设备可识别病毒的特征。

#### 79. 浏览器/服务器模式 Browser/Server

是一种网络结构模式，浏览器是客户端最主要的应用软件。

深圳地铁 企业标准

深圳地铁 企业标准

# 目 录

前 言.....	I
引 言.....	II
缩 略 语.....	III
专用术语及名词解释.....	VII
1 目的.....	1
2 范围.....	1
3 参考标准.....	1
4 系统接口技术规范.....	1
4.1 概述.....	1
4.2 接口数据组成及详细格式定义.....	2
5 通信接口技术规范.....	100
5.1 概述.....	100
5.2 通信服务程序功能摘要.....	100
5.3 通信接口.....	103
5.4 管理接口.....	106
6 设备部件接口函数规范.....	123
6.1 说明.....	123
6.2 工控机接口.....	123
6.3 单程票发售机构.....	133
6.4 单程票回收机构.....	154
6.5 通行控制模块.....	171
6.6 硬币模块.....	187
6.7 纸币接收模块.....	208
6.8 纸币找零模块.....	225
7 读写器接口技术规范.....	242

7.1 接口数据定义 .....	242
7.2 函数定义 .....	244
7.3 函数返回值 .....	261
7.4 ECU 调用流程.....	264
附录 A （资料性附录）设备部件接口业务与函数调用对应表.....	266

## 图目录

图 1 AFC 系统层次架构图.....	2
图 2 数据上传处理流程图 .....	4
图 3 数据下发处理流程图.....	5
图 4 系统控制命令执行流程图.....	6
图 5 ECU 启动读写器流程.....	264
图 6 ECU 票卡业务流程.....	265

## 表目录

表 1 参数列表.....	7
表 2 交易数据列表.....	8
表 3 审计数据列表.....	8
表 4 运营数据列表.....	9
表 5 事务数据列表.....	10
表 6 状态数据列表.....	10
表 7 票务数据列表.....	11
表 8 任务数据列表.....	11
表 9 认证数据列表.....	12

表 10 日志数据列表..... 12

表 11 SC 上传至 CLC 的数据列表..... 13

表 12 CLC 下发给 SC 的数据列表..... 14

表 13 CLC 与 SC 之间的交互数据列表..... 15

表 14 初始化机上传至 CLC 的数据列表..... 15

表 15 CLC 下发给初始化机的数据列表..... 16

表 16 CLC 与初始化机交互数据列表..... 16

表 17AGM 上传至 SC 的数据列表..... 17

表 18 SC 下发给 AGM 的数据列表..... 18

表 19 SC 与 AGM 之间的交互数据列表..... 18

表 20 TVM 上传至 SC 的数据列表..... 19

表 21 SC 下发给 TVM 的数据列表..... 19

表 22 SC 与 TVM 之间的交互数据列表..... 20

表 23 AVM 上传至 SC 的数据列表..... 20

表 24 SC 下发给 AVM 的数据列表..... 20

表 25 SC 与 AVM 之间的交互数据列表..... 21

表 26 BOM 上传至 SC 的数据列表..... 21

表 27 SC 下发给 BOM 的数据列表..... 22

表 28 SC 与 BOM 之间的交互数据列表..... 23

表 29 PCA 上传至 SC 的数据列表..... 23

表 30 SC 下发给 PCA 的数据列表..... 23

表 31 SC 与 PCA 的交互数据列表..... 23

表 32 核准机上传给 SC 数据列表..... 24

表 33 SC 下发给核准机数据列表..... 24

表 34 SC 与核准机交互数据列表..... 25

表 35 换票机上传给 SC 数据列表..... 25

表 36 SC 下发给换票机数据列表..... 26

表 37 SC 与换票机交互数据列表..... 26

表 38 数据格式说明列表..... 27

表 39 参数文件格式列表 .....	28
表 40 卡黑名单参数列表 .....	28
表 41 计价方案参数列表.....	29
表 42 参数标识字符和参数内容列表 .....	31
表 43 计费规则参数列表 .....	38
表 44 计费处理列表.....	40
表 45 计费规则范例解释列表.....	41
表 46 联程优惠规则列表 .....	43
表 47 运营点参数列表 .....	43
表 48 TVM 售票时间参数列表.....	45
表 49 TVM 售票地图参数列表.....	47
表 50 种类参数列表 .....	49
表 51 设备参数列表.....	50
表 52 交易类型列表 .....	51
表 53 售卡交易列表.....	52
表 54 现金充值交易列表 .....	53
表 55 更新交易.....	55
表 56 单程票超乘处理交易.....	56
表 57 清起始标记.....	58
表 58 消费起始交易 .....	59
表 59 普通消费交易结算.....	60
表 60 联程交易结算 .....	62
表 61 退票交易 .....	64
表 62 激活交易 .....	65
表 63 黑名单锁卡交易 .....	67
表 64 黑名单解锁交易 .....	68
表 65 行政罚款交易 .....	70
表 66 行政处理原因编码表 .....	71
表 67 行政支付交易 .....	71



表 68 AGM 寄存器 .....	72
表 69 TVM 寄存器 .....	73
表 70 AVM 寄存器 .....	74
表 71 BOM 寄存器 .....	74
表 72 初始化机寄存器.....	75
表 73 核准机寄存器.....	75
表 74 换票机寄存器.....	76
表 75 交易汇总数据.....	76
表 76 运营控制数据列表.....	77
表 77 状态数据列表.....	82
表 78 AGM 更换票箱记录 .....	82
表 79 AGM 票箱切换记录 .....	83
表 80 TVM 补票记录 .....	83
表 81 更换钱箱记录.....	84
表 82 TVM 补充找零硬币记录 .....	84
表 83 车票配发通知.....	85
表 84 车票回收通知.....	85
表 85 上交车票通知.....	86
表 86 车票调拨通知.....	86
表 87 短款补款通知.....	87
表 88 上交补款单据.....	87
表 89 上报车站售存日报.....	87
表 90 上报车站营收日报.....	88
表 91 上报售票员结算单.....	89
表 92 上报设备票款差异.....	90
表 93 生产任务单据.....	91
表 94 生产任务反馈单据.....	92
表 95 任务属性值列表.....	93
表 96 联网登录认证.....	95

表 97 密码修改认证 .....	95
表 98 获取账户分组信息.....	95
表 99 获取设备组权限 .....	96
表 100 安全认证数据 .....	96
表 101 设备认证记录.....	97
表 102 操作日志.....	99
表 103 调试日志.....	99
表 104 吞吐量要求 .....	102
表 105 API 参数接口数据定义.....	242
表 106 初始化读写器通讯口.....	244
表 107 关闭读写器通讯口.....	244
表 108 获取版本信息函数.....	245
表 109 日志开关函数.....	245
表 110 上传文件函数.....	246
表 111 更新文件函数.....	246
表 112 AFC 业务启动函数.....	247
表 113 状态查询函数.....	247
表 114 AFC 业务关闭函数.....	248
表 115 管理业务函数.....	248
表 116 输出数据说明示例 .....	248
表 117 同步时钟函数.....	249
表 118 票价计算函数.....	249
表 119 验卡函数.....	249
表 120 询卡函数.....	251
表 121 检票业务函数.....	252
表 122 发售业务函数.....	253
表 123 充值业务函数.....	254
表 124 票卡处理业务函数.....	255
表 125 换卡业务函数.....	256

表 126 个人化业务函数.....	257
表 127 初始化业务函数.....	257
表 128 预赋值业务函数.....	258
表 129 抵消业务函数.....	259
表 130 注消业务函数.....	260
表 131 票卡物理数据读卡函数.....	261
表 132 错误返回值描述.....	261
表 A. 1 单程票发售机构业务与函数调用对应表 .....	266
表 A. 2 单程票回收模块业务与函数调用对应表 .....	267
表 A. 3 机芯业务与函数调用对应表 .....	268
表 A. 4 硬币模块业务与函数调用对应表 .....	269
表 A. 5 纸币接收模块业务与函数调用对应表 .....	271
表 A. 6 纸币找零模块业务与函数调用对应表 .....	272

深圳地铁 企业标准

## 1 目的

本文档规范了深圳地铁 AFC 系统数据接口，通信中间件的功能、性能、业务接口，设备部件软件接口，终端设备和读写器之间的 API 接口。

## 2 范围

本技术规范规定了深圳地铁 AFC 系统数据接口，确定了系统数据类型和格式，给出了深圳地铁 AFC 系统数据设计标准。作为深圳地铁 AFC 系统设计、实施和项目验收的技术依据，适用于深圳地铁 AFC 系统通信中间件的设计、开发和测试指导、AFC 系统部件软件接口的设计、开发和测试和 AFC 系统终端设备和读写器的软件接口设计、开发和测试。

## 3 参考标准

《深圳市轨道交通清分中心（ACC）系统接口规范》。

《系统和设备技术规范》

## 4 系统接口技术规范

### 4.1 概述

系统与数据接口技术规范统一定义了 AFC 系统数据类型、数据格式、数据内容，在兼容现有 AFC 系统数据接口规范的基础上扩展了新的数据类型和内容。

该技术规范主要包括 AFC 系统层次架构，数据流解析、数据概览（包括：参数、交易数据、审计数据、运营控制数据、事务数据、状态数据、票务数据、任务数据、认证数据、日志等）、数据传输、数

据定义等内容。

4.2 接口数据组成及详细格式定义

4.2.1 AFC 系统数据流

4.2.1.1 AFC 系统层次结构

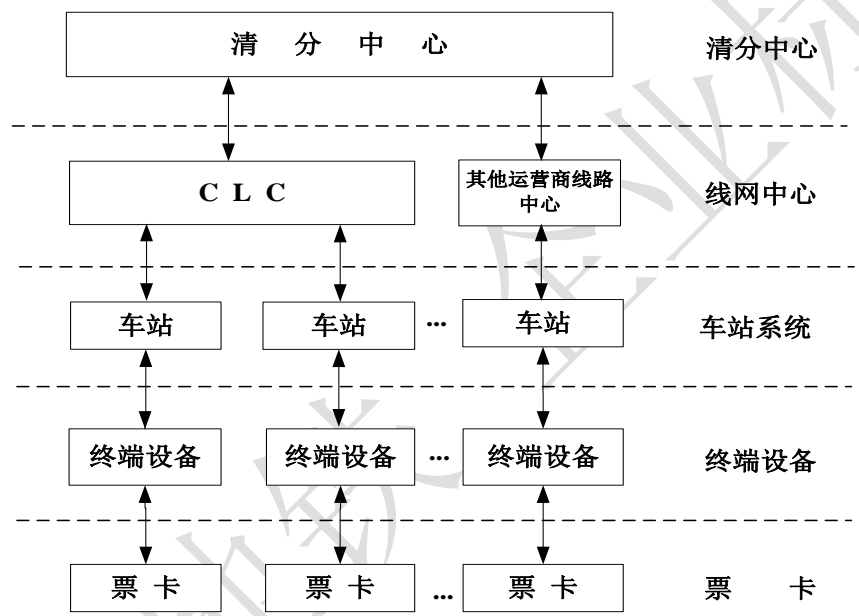


图 1 AFC 系统层次架构图

深圳地铁 AFC 系统分为 5 层：

第一层：票卡。乘客所持的车费支付媒介。

第二层：终端设备。为乘客提供售检票服务的终端设备和票务管理终端，包括：闸机、自动售票机、自动充值机、半自动票务处理机、手持检票机、单程票清分机、单程票编码器、核准机、单程票换票机等。

第三层：车站系统。主要功能是收集本站终端设备产生的交易数据、审计数据、状态数据等业务数

据，根据系统设置将数据上传至线网中心，接收线网中心下发的控制管理数据，完成存储、转发、控制等功能，实现对本站终端设备的实时监控，完成车站票务运作和管理。

第四层：CLC。CLC 是多条线路中心的集中，主要功能是完成对线网车站交易数据、审计数据、结算数据等的采集，并根据系统设置将相关数据上传至清分中心；对数据进行存储、结算，输出相应报表；统一管理线网终端设备参数和用户权限、系统运行模式等；完成 AFC 系统主要运营管理、票务管理、维护维修等工作内容。

第五层：清分中心。主要功能是接收不同运营商发送上来的原始交易，根据清分规则完成清分，并将清分结果及时反馈给运营商，管理线网化运营参数，完成 AFC 系统单程票的采购生产任务。

#### 4.2.1.2 AFC 系统数据流程图

AFC 系统完成终端设备的控制与数据采集，上级系统 CLC 和 SC 可以向设备下发各种参数控制设备的运行，也可以下发各种控制命令对设备进行控制。终端设备向上级系统上传设备交易数据、状态数据、日志数据、审计数据等。

##### 4.2.1.2.1 数据上传流程

设备生成数据后，对数据的合法性进行检查，然后按照数据的分类向上传送。车站计算机系统收到设备数据后，将数据入库并转发给线网中央计算机系统。线网中央计算机系统收到数据后将数据入库，并将交易数据转发给轨道交通清分系统。数据上传处理的流程图如下：

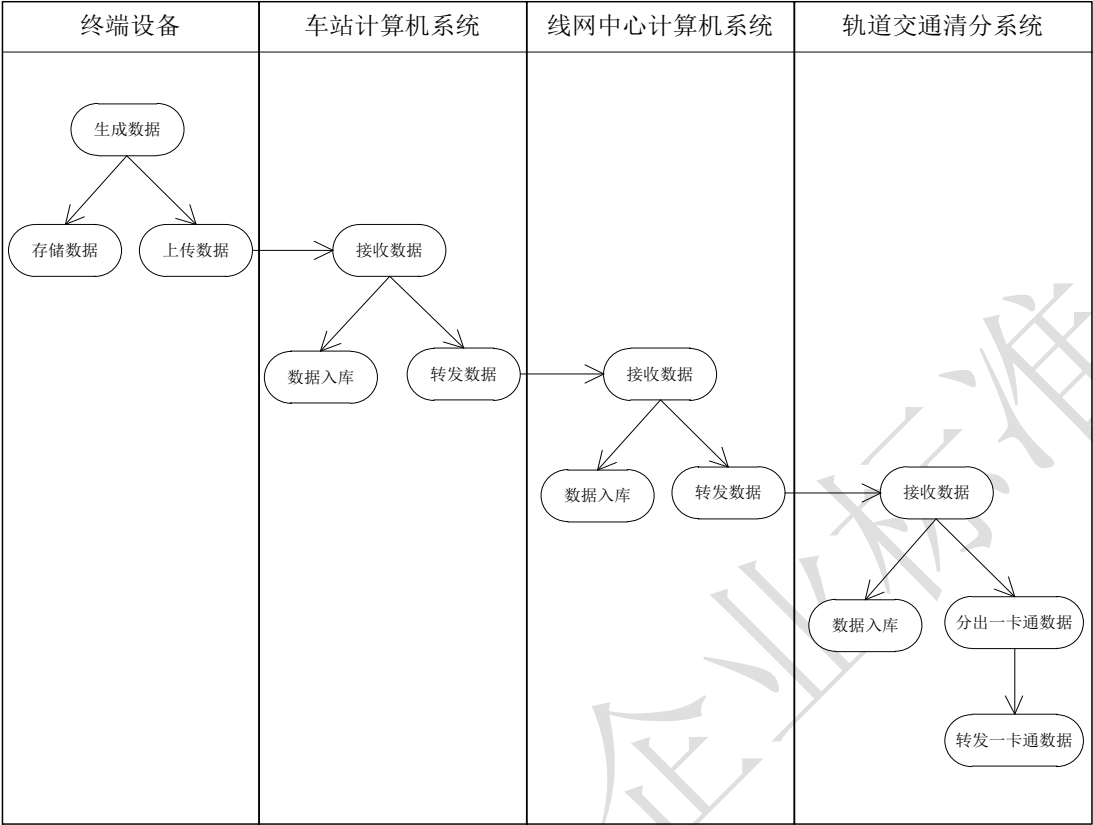


图 2 数据上传处理流程图

4.2.1.2.2 数据下发流程

线网中央计算机系统接收轨道交通清分系统下发的参数，并且定义线网内运营相关的参数，如计价方案、设备种类、黑名单、用户权限参数等。线网中央计算机系统向车站计算机系统下发相关参数，车站计算机系统转发给本站设备。

参数下发处理流程如下：



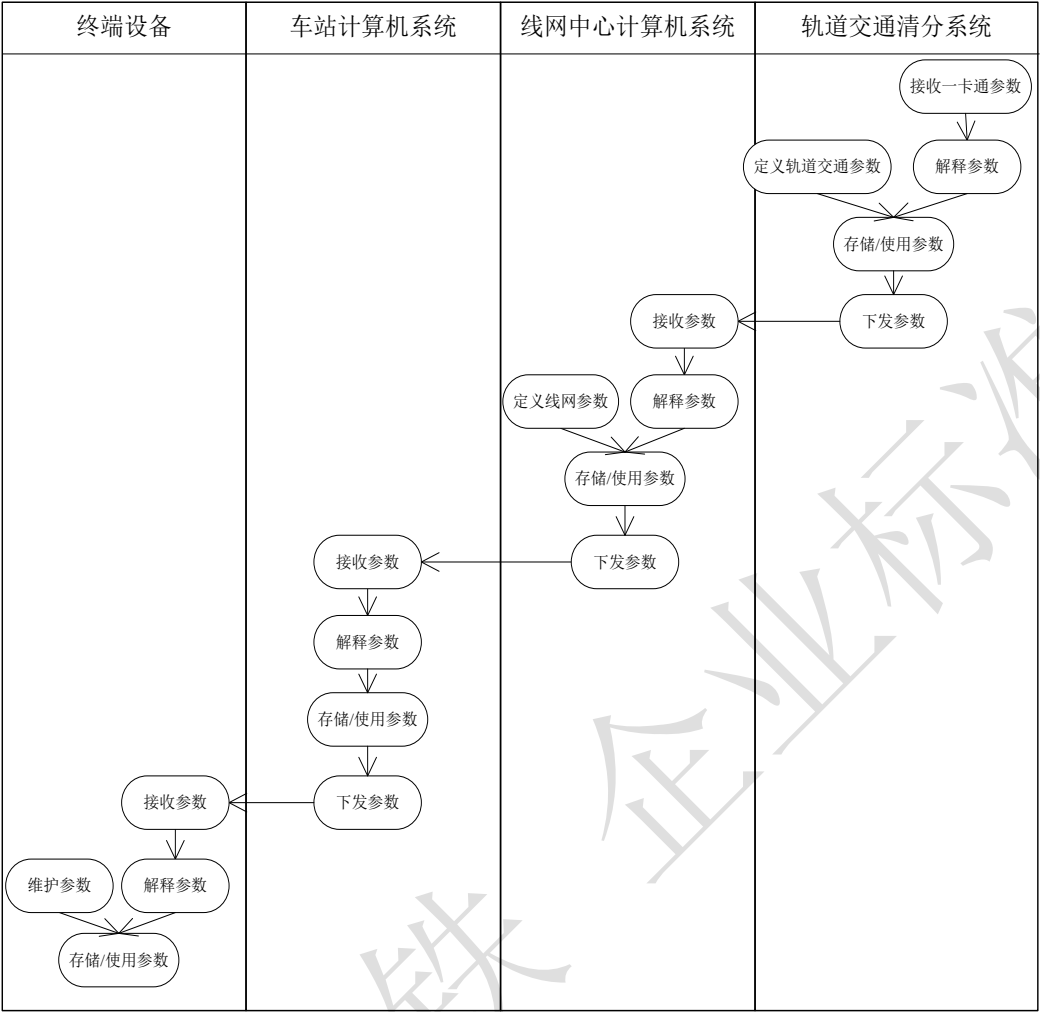


图 3 数据下发处理流程图

4.2.1.2.3 控制命令执行流程

线网中央计算机系统可以下发控制命令控制车站和设备的运营。

车站计算机系统负责转发线网中央计算机系统下发的控制命令，同时也可自行下发控制命令控制本车站设备的运营。车站终端设备负责接收控制命令并执行。

系统控制命令的执行流程如下：

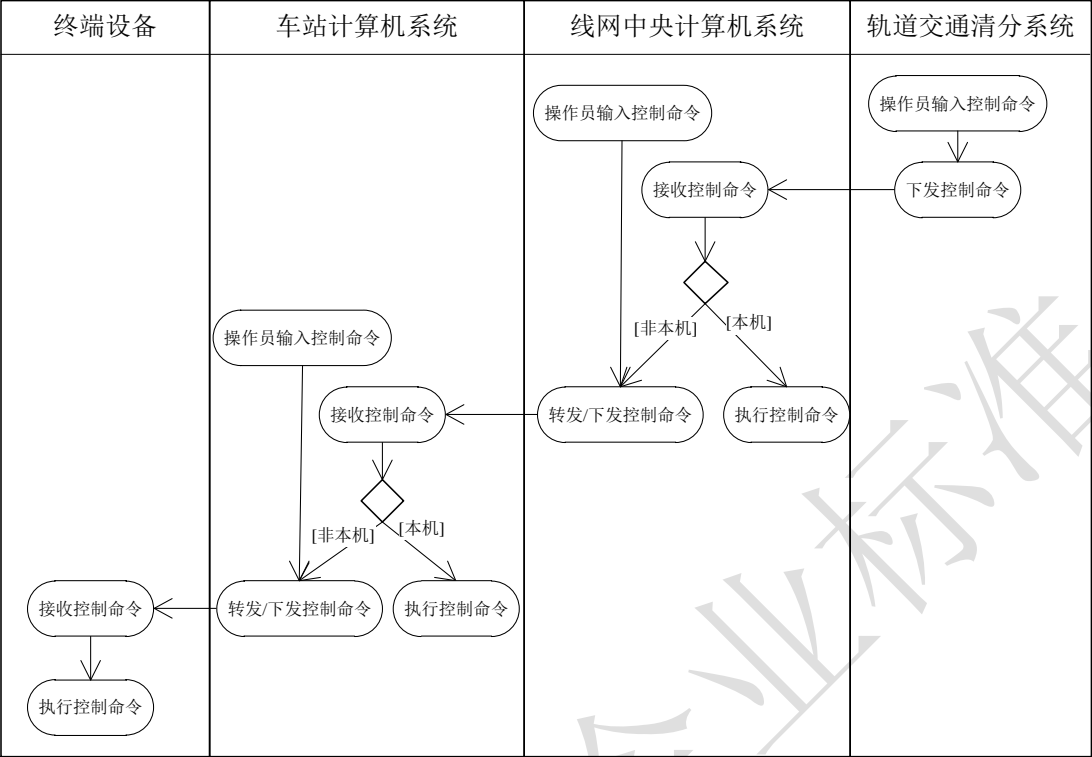


图 4 系统控制命令执行流程图

4.2.2 接口数据概览

系统接口数据包括：参数、交易数据、审计数据、运营控制数据、事务数据、状态数据、票务数据、任务数据、认证数据、日志数据。

## 4.2.2.1 参数

表 1 参数列表

数据类别	数据内容	实现功能	备注
参数	黑名单	管理深圳通下发的卡黑名单	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	计价方案	线网计价方案, 主要包括: 线网站点之间的基本票价表 各种票卡类型的应用属性 各种票卡类型不同情况下适用的计费规则以及联程规则	
	节假日表	年度节假日清单, 配合计价方案使用	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	运营点参数	主要包括运营点基本信息、线路基本信息、区段基本信息、运营点线路对应关系、运营点区段对应关系	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	TVM 售票时间参数	TVM 开始售票时间、TVM 停止售票时间	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	TVM 售票地图参数	主要包括 TVM 地图背景、TVM 运营点对应坐标、TVM 线路信息(颜色)、非当前站点颜色、当前站点颜色、选中站点颜色	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	种类参数	设备种类信息设置	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	设备参数	设备个性信息设置	参数变更时由 CLC 向 SC 发送, SC 转发给指定的车站设备
	用户权限参数	包括用户信息、用户分组信息、系统设备功能点信息	用户权限联机认证, 保留用户权限参数
	发卡方特殊参数	预留给金融卡, 或者其他发卡方, 需要下发给终端设备的特殊文件, 统一采用参数进行封装	预留

4.2.2.2 交易数据

表 2 交易数据列表

数据类别	数据内容	实现功能	备注
交易数据	AFC 系统交易数据	格式化 AFC 系统的交易数据	设备向 SC 发送，SC 向 CLC 发送

4.2.2.3 审计数据

表 3 审计数据列表

数据类别	数据内容	实现功能	备注
审计数据	寄存器数据	设备按照统计维度记录寄存器或流水数据，按照设置条件上传至 SC，SC 转发至 CLC	设备向 SC 发送,SC 向 CLC 发送
	交易汇总数据	运营结束时，设备和 SC 统计当前运营日的所有交易汇总数据。	设备向 SC 发送,SC 向 CLC 发送

## 4.2.2.4 运营控制数据

表 4 运营数据列表

数据类别	数据内容	实现功能	备注
运营控制数据	运营模式通知	当有车站的运营模式发生变化, CLC 发送该线路车站模式变化的通知, 使其他车站根据规则做相应的模式调整	车站发生模式事件后, 向 CLC 发送通知。然后 CLC 向其它车站以及 ACC 转发此模式通知。
	运营模式控制	用于控制本车站进入某种运营模式	SC 发送命令至本站的设备, 并同时向 CLC 发送通知
	设备暂停	用于控制设备暂停服务	SC 发送命令至指定设备
	设备开启	用于控制设备重新开启服务	SC 发送命令至指定设备
	设备重启	用于控制设备重新开机	SC 发送命令至指定设备
	设备关机	用于控制设备关机	SC 发送命令至指定设备
	设备工作方式控制	用于控制指定设备的工作方式	SC 发送命令至指定设备
	设备运营模式控制	用于控制指定设备的运营模式	SC 发送命令至指定设备
	停运车站	目的车站为停运车站, 整站停止运营; 目的车站为非停运车站, TVM 不能发售至停运车站的单程票	CLC 向线网内所有车站发送通知, SC 接收后发给指定的 TVM, 并同时通知 ACC
	参数版本同步通知	通知指定设备与 CLC 同步参数版本	SC 发送命令至指定设备
	参数版本查询	向指定设备查询目前设备使用的参数版本信息	SC 发送命令至指定设备
	软件版本查询	向指定设备查询目前设备使用的软件版本信息	SC 发送命令至指定设备
	时钟查询	CLC 向 SC 以及设备查询时钟状况, SC 可向设备查询时钟状况	CLC 向 SC 发时钟查询请求, SC 应答 CLC, 上传时钟状况
	时钟同步	每隔设定的时间设备向 CLC 同步时钟, 时钟误差在规定范围内则自动同步; 误差超过规定范围则人工处理	ACC 提供两种同步方式可选: 开放标准的时钟同步协议和端口 (SNTP 协议, UDP 123 端口) 提供专用程序模块, 在 LC 端计算机安装后实现时钟同步

4.2.2.5 事务数据

表 5 事务数据列表

数据类别	数据内容	实现功能	备注
事务数据	钱箱操作	设备更换钱箱的信息	设备向 SC 定时发送，SC 向 CLC 定时发送
	票箱操作	设备更换票箱的信息	设备向 SC 定时发送，SC 向 CLC 定时发送

4.2.2.6 状态数据

表 6 状态数据列表

数据类别	数据内容	实现功能	备注
状态数据	设备状态信息	上传设备运行的实时状态信息	设备向 SC 定时发送，SC 向 CLC 定时发送
	事件报警信息	上传设备、系统的告警信息	设备、系统出现告警即时向上级发送

## 4.2.2.7 票务数据

表 7 票务数据列表

数据类别	数据内容	实现功能	备注
车票数据	车票配发通知	CLC 向车站配发车票	需要时 CLC 提供服务供 SC 调用
	车票调拨通知	不同车站之间调拨车票	
	车票回收通知	CLC 向车站下达回收车票通知	
	接收车票处理	车站接收 CLC 配发过来的车票入库处理	
	上交车票处理	车站上交车票出库处理	
	车站上报车票库存	上传车站车票的库存数据	每日运营结束后 SC 向 CLC 发送当站车票库存
	上报设备车票差异		车站每个运营日结算时上报
收益数据	短款补款通知		CLC 收益核对完成之后, 给车站下达补款通知
	上缴补款处理		车站补款时确认处理, 处理结果需要传送至 CLC
	上报车站售存日报		车站每个运营日结算时上报
	上报车站营收日报		车站每个运营日结算时上报
	上报售票员结算		车站每个运营日结算时上报
	上报设备票款差异		车站每个运营日结算时上报

## 4.2.2.8 任务数据

表 8 任务数据列表

数据类别	数据内容	实现功能	备注
任务数据	生产任务文件	CLC 票务系统给编码器下发的任务文件	CLC 发送至编码器
	生产任务反馈文件	编码器根据任务完成情况汇总成文件反馈给 CLC	编码器发送至 CLC

## 4.2.2.9 认证数据

表 9 认证数据列表

数据类别	数据内容	实现功能	备注
权限联机数据	联网登录	在设备联网下连接CLC登录设备	直接连接 CLC 服务
	修改密码	在设备联网下登录成功的操作员修改自己的密码	直接连接 CLC 服务
	获取权限信息	手动刷新以及开机时连接CLC服务, 获取本设备的权限信息并保存至本地	直接连接 CLC 服务
	SAM 卡认证	设备初次接入系统时认证SAM卡	直接连接 CLC 服务

## 4.2.2.10 日志

表 10 日志数据列表

数据类别	数据内容	实现功能	备注
日志	调试日志	设备调试日志文件根据需要采集至 SC 并保存, 如果 CLC 有需从 SC 拷贝至 CLC 本地。	设备定时上传至 SC
	操作日志	主要包括: 用户登录设备信息, 操作设备信息, 维护维修设备操作记录, 系统控制设备记录, 以及业务流程记录及结果, 由设备上传至 SC, SC 转发至 CLC	设备定时上传至 SC, SC 定时上传至 CLC 并入库



## 4.2.3 接口数据传输

## 4.2.3.1 LC 与 SC 间数据传输

## 4.2.3.1.1 SC 上传至 CLC 的数据列表

表 11 SC 上传至 CLC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时/定时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	车票配送接口	车票配发/回收申请数据	需要时
8		车票配发反馈数据	需要时
9		车票上交数据	需要时
10		车票调拨数据	需要时
11		车票库存数据	需要时
12	日报接口	车站营收日报	定时/需要时
13		车站车票售存日报	定时/需要时
14		设备票款差异日报	定时/需要时
15		设备车票差异日报	定时/需要时
16	通信接口	包统计数据	运营结束前
17	软件版本控制接口	软件版本检查请求	运营开始前/需要时
18		软件版本更新请求	运营开始前/需要时
19	收益管理接口	收益寄存器数据	运营结束时/定时
20		收益审计数据	运营结束时
21		交易查询请求数据	需要时
22		交易申请数据	需要时
23	维护维修接口	用户登录设备日志	及时
24		设备维修数据	及时
25	时钟同步接口	时钟同步请求	运营开始前/定时

## 4.2.3.1.2 CLC 下发给 SC 的数据列表

表 12 CLC 下发给 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/变更时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		卡黑名单参数	运营开始前/变更时
6		设备黑名单参数	运营开始前/变更时
7		运作时间参数	运营开始前/变更时
8		运营点参数	运营开始前/变更时
9		TVM 售票地图参数	运营开始前/变更时
10		系统其他参数	运营开始前/变更时
11	车票配送接口	流通车票类型数据	需要时
12		车票配送数据	需要时
13		车票回收通知数据	需要时
14		车票调拨通知数据	需要时
15		车票申请反馈数据	需要时
16		车票回收反馈数据	需要时
17	控制命令接口	设备暂停命令	需要时
18		设备暂停后重新开启命令	需要时
19		设备参数版本查询命令	需要时
20		设备软件版本查询命令	需要时
21		模式转换命令	需要时
22		数据重传命令	需要时
23		寄存器查询命令	需要时
24	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
25		软件更新数据	运营开始前/变更时
26	时钟同步接口	时钟同步数据	运营开始前/需要时
27	收益管理接口	交易查询反馈数据	需要时
28		交易申请反馈数据	需要时

## 4.2.3.1.3 CLC 与 SC 之间的交互数据列表

表 13 CLC 与 SC 之间的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时/变更时
3		一卡通交易认证数据	及时

## 4.2.3.2 CLC 与初始化机间数据传输

## 4.2.3.2.1 初始化机上传至 CLC 的数据列表

表 14 初始化机上传至 CLC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时//需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/定时
8		收益审计数据	运营结束时
9	软件版本控制接口	软件版本检查请求	运营结束时/需要时
10		软件版本更新请求	运营结束时/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	需要时
13		设备维修数据	需要时

4.2.3.2.2 CLC 下发给初始化机的数据列表

表 15 CLC 下发给初始化机的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4	控制命令接口	设备暂停命令	需要时
5		设备暂停后重新开启命令	需要时
6		设备参数版本查询命令	需要时
7		设备软件版本查询命令	需要时
8		模式转换命令	需要时
9		数据重传命令	需要时
10		寄存器查询命令	需要时
11	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
12		软件更新数据	运营开始前/需要时
13	时钟同步接口	时钟同步数据	运营开始前/需要时

4.2.3.2.3 CLC 与初始化机交互数据列表

表 16 CLC 与初始化机交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时

## 4.2.3.3 SC 与 AGM 间数据传输

## 4.2.3.3.1 AGM 上传至 SC 的数据列表

表 17 AGM 上传至 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时//需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/定时
8		收益审计数据	运营结束时
9	软件版本控制接口	软件版本检查请求	运营结束时/需要时
10		软件版本更新请求	运营结束时/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	需要时
13		设备维修数据	需要时

## 4.2.3.3.2 SC 下发给 AGM 的数据列表

表 18 SC 下发给 AGM 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		卡黑名单参数	运营开始前/变更时
6		设备黑名单参数	运营开始前/变更时
7		运作时间参数	运营开始前/变更时
8		运营点参数	运营开始前/变更时
9	控制命令接口	设备暂停命令	需要时
10		设备暂停后重新开启命令	需要时
11		设备参数版本查询命令	需要时
12		设备软件版本查询命令	需要时
13		模式转换命令	需要时
14		数据重传命令	需要时
15		寄存器查询命令	需要时
16	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
17		软件更新数据	运营开始前/需要时
18	时钟同步接口	时钟同步数据	运营开始前/需要时

## 4.2.3.3.3 SC 与 AGM 之间的交互数据列表

表 19 SC 与 AGM 之间的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时

## 4.2.3.4 SC 与 TVM 间数据传输

## 4.2.3.4.1 TVM 上传至 SC 的数据列表

表 20 TVM 上传至 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时/需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/需要时
8		收益审计数据	运营结束时/需要时
9	软件版本控制接口	软件版本检查请求	运营开始前/需要时
10		软件版本更新请求	运营开始前/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	及时
13		设备维修数据	及时

## 4.2.3.4.2 SC 下发给 TVM 的数据列表

表 21 SC 下发给 TVM 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		运作时间参数	运营开始前/变更时
6		运营点参数	运营开始前/变更时
7		TVM 售票地图参数	运营开始前/变更时
8	控制命令接口	设备暂停命令	需要时
9		设备暂停后重新开启命令	需要时
10		设备参数版本查询命令	需要时
11		设备软件版本查询命令	需要时
12		模式转换命令	需要时
13		数据重传命令	需要时
14		寄存器查询命令	需要时
15	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
16		软件更新数据	运营开始前/需要时
17	时钟同步接口	时钟同步数据	运营开始前/需要时

## 4.2.3.4.3 SC 与 TVM 之间的交互数据列表

表 22 SC 与 TVM 之间的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时

## 4.2.3.5 SC 与 AVM 间数据传输

## 4.2.3.5.1 AVM 上传至 SC 的数据列表

表 23 AVM 上传至 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时/需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/需要时
8		收益审计数据	运营结束时/需要时
9	软件版本控制接口	软件版本检查请求	运营开始前/需要时
10		软件版本更新请求	运营开始前/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	及时
13		设备维修数据	及时

## 4.2.3.5.2 SC 下发给 AVM 的数据列表

表 24 SC 下发给 AVM 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时



表 24 SC 下发给 AVM 的数据列表（续）

序号	接口类型	数据类型	交换时间
5		运作时间参数	运营开始前/变更时
6		运营点参数	运营开始前/变更时
7	控制命令接口	设备暂停命令	需要时
8		设备暂停后重新开启命令	需要时
9		设备参数版本查询命令	需要时
10		设备软件版本查询命令	需要时
11		模式转换命令	需要时
12		数据重传命令	需要时
13		寄存器查询命令	需要时
14	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
15		软件更新数据	运营开始前/需要时
16	时钟同步接口	时钟同步数据	运营开始前/需要时

## 4.2.3.5.3 SC 与 AVM 之间的交互数据列表

表 25 SC 与 AVM 之间的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时
3		一卡通交易认证数据	及时/变更时

## 4.2.3.6 SC 与 BOM 间数据传输

## 4.2.3.6.1 BOM 上传至 SC 的数据列表

表 26 BOM 上传至 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时/需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时

表 26 BOM 上传至 SC 的数据列表 (续)

序号	接口类型	数据类型	交换时间
7	软件版本控制接口	软件版本检查请求	运营开始前/需要时
8		软件版本更新请求	运营开始前/需要时
9	收益管理接口	收益寄存器数据	运营结束时/定时
10		收益审计数据	运营结束时
11		交易查询请求数据	需要时
12		交易申请数据	需要时
13	维护维修接口	用户登录设备日志	需要时
14		设备维修数据	需要时
15	时钟同步接口	时钟同步申请	运营开始前/定时

## 4.2.3.6.2 SC 下发给 BOM 的数据列表

表 27 SC 下发给 BOM 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/变更时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		卡黑名单参数	运营开始前/变更时
6		设备黑名单参数	运营开始前/变更时
7		运作时间参数	运营开始前/变更时
8		运营点参数	运营开始前/变更时
9	控制命令接口	设备暂停命令	需要时
10		设备暂停后重新开启命令	需要时
11		设备参数版本查询命令	需要时
12		设备软件版本查询命令	需要时
13		模式转换命令	需要时
14		数据重传命令	需要时
15		寄存器查询命令	需要时
16	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
17		软件更新数据	运营开始前/变更时
18	时钟同步接口	时钟同步数据	运营开始前/需要时
19	收益管理接口	交易查询反馈数据	需要时
20		交易申请反馈数据	需要时

## 4.2.3.6.3 SC 与 BOM 之间的交互数据列表

表 28 SC 与 BOM 之间的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	及时
2		密钥数据	及时
3		一卡通交易认证数据	及时/变更时

## 4.2.3.7 SC 与手持检验票机间数据传输

## 4.2.3.7.1 PCA 上传至 SC 的数据列表

表 29 PCA 上传至 SC 的数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时/需要时

## 4.2.3.7.2 SC 下发给 PCA 的数据列表

表 30 SC 下发给 PCA 的数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3	软件版本控制接口	软件更新数据	运营开始前/需要时
4	时钟同步接口	时钟同步数据	运营开始前/需要时

## 4.2.3.7.3 SC 与 PCA 的交互数据列表

表 31 SC 与 PCA 的交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	连接时
2		密钥数据	连接时
3		一卡通交易认证数据	连接时

4.2.3.8 SC 与核准机间数据传输

4.2.3.8.1 核准机上传给 SC 数据列表

表 32 核准机上传给 SC 数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时//需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/定时
8		收益审计数据	运营结束时
9	软件版本控制接口	软件版本检查请求	运营结束时/需要时
10		软件版本更新请求	运营结束时/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	需要时
13		设备维修数据	需要时

4.2.3.8.2 SC 下发给核准机数据列表

表 33 SC 下发给核准机数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		运作时间参数	运营开始前/变更时
6		运营点参数	运营开始前/变更时
7	控制命令接口	设备暂停命令	需要时
8		设备暂停后重新开启命令	需要时
9		设备参数版本查询命令	需要时
10		设备软件版本查询命令	需要时
11		模式转换命令	需要时
12		数据重传命令	需要时
13		寄存器查询命令	需要时

表 33 SC 下发给核准机数据列表（续）

序号	接口类型	数据类型	交换时间
14	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
15		软件更新数据	运营开始前/需要时
16	时钟同步接口	时钟同步数据	运营开始前/需要时

## 4.2.3.8.3 SC 与核准机交互数据列表

表 34 SC 与核准机交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	连接时
2		密钥数据	连接时
3		一卡通交易认证数据	连接时

## 4.2.3.9 SC 与换票机间数据传输

## 4.2.3.9.1 换票机上传给 SC 数据列表

表 35 换票机上传给 SC 数据列表

序号	接口类型	数据类型	交换时间
1	数据采集接口	交易数据	及时
2		状态数据	及时
3		日志数据	及时
4		寄存器数据	运营结束时//需要时
5	参数管理接口	参数版本检查请求	运营开始前/需要时
6		软件版本更新请求	运营开始前/需要时
7	收益管理接口	收益寄存器数据	运营结束时/定时
8		收益审计数据	运营结束时
9	软件版本控制接口	软件版本检查请求	运营结束时/需要时
10		软件版本更新请求	运营结束时/需要时
11	时钟同步接口	时钟同步请求	运营开始前/定时
12	维护维修接口	用户登录设备日志	需要时
13		设备维修数据	需要时

4.2.3.9.2 SC 下发给换票机数据列表

表 36 SC 下发给换票机数据列表

序号	接口类型	数据类型	交换时间
1	参数管理接口	种类参数	运营开始前/变更时
2		设备参数	运营开始前/需要时
3		操作员权限参数	运营开始前/变更时
4		计价方案参数	运营开始前/变更时
5		运作时间参数	运营开始前/变更时
6		运营点参数	运营开始前/变更时
7	控制命令接口	设备暂停命令	需要时
8		设备暂停后重新开启命令	需要时
9		设备参数版本查询命令	需要时
10		设备软件版本查询命令	需要时
11		模式转换命令	需要时
12		数据重传命令	需要时
13		寄存器查询命令	需要时
14	软件版本控制接口	软件版本反馈数据	运营开始前/需要时
15		软件更新数据	运营开始前/需要时
16	时钟同步接口	时钟同步数据	运营开始前/需要时

4.2.3.9.3 SC 与换票机交互数据列表

表 37 SC 与换票机交互数据列表

序号	接口类型	数据类型	交换时间
1	安全管理接口	设备认证数据	连接时
2		密钥数据	连接时
3		一卡通交易认证数据	连接时

4.2.4 数据格式定义

4.2.4.1 格式说明

表 38 数据格式说明列表

格式	说明
Char	ASCII 字符
BIN	二进制字节
BCD	二-十进制代码
HEX	由二进制字节数据扩展成的 16 进制 ASCII 码字符串

4.2.4.1.1 电子签名

ACC 与 CLC 的接口数据中，部分重要数据需要电子签名来保证数据的安全，通过电子签名可以确认数据真实且不可篡改，并确保数据发布不可抵赖。

算法：

D1 = MD5（传输数据）。

电子签名 = RSA（D1）。

在文件格式中，电子签名在文件尾部，其中，传输数据时从文件开头到电子签名前的所有文件数据

4.2.4.1.2 补充说明

- a) 字符串补齐说明：本接口文档中需要定长的字符串中，对于数值类的字段一律使用左补0的方式补足定长的位数,对于非数值类字符串一律使用左补空格的方式补足定长的位数。
- b) 时间：14 Bytes，格式：“YYYYMMDDHHMMSS”
- c) 日期：ASCII，8 Byte 格式：“YYYYMMDD”
- d) 数值：以字符形式体现的数值，左对齐，不足补空
- e) 金额：ASCII，8 Bytes，精确到分，如“00001000”表示10 元，如果为负值，则最高位为“-”，即负10 元表示为“-0001000”
- f) 汇率：8 Bytes ，“06.13471”表示1 单位外币/1 单位本币： 6.13471。
- g) X：表示内容不确定。
- h) TAB 键：键值为 asc(09)。

4.2.4.2 参数数据

#### 4.2.4.2.1 参数文件格式

参数数据采用 XML 格式组织，每组参数包括格式相同的参数版本信息如下表：

表 39 参数文件格式列表

标记	属性	说明
<Version		
	VersionId	参数版本号（字符串，非空，小于等于 30 位）
	VersionDate	参数版本日期（时间型，格式：YYYYMMDDHHMMSS）
	VersionType	版本类型（分为历史版本，当前版本，将来版本） 0：历史版本 1：当前版本 2：将来版本
	VersionValidate	版本生效时间（时间型，格式：YYYYMMDDHHMMSS）
	VersionSystem	参数发布方系统编号（数字串，小于等于 3 位）
	VersionOper	操作员（数字串，6 位）
	FormatVersion	格式版本（数字串，2 位）
/>		

#### 4.2.4.2.2 卡黑名单参数

文件名： BlackList\_khmd +版本号（yyyymmdd+4 位流水号）.xml

表 40 卡黑名单参数列表

标记	标记出现次数	属性	说明
<BlackList >	1, 1		
<Version/>	1, 1		版本
<ReaderParam	1, 1		读写器相关参数
		SrcAddr	源地址
		CreateDate	生成时间
< ReaderParamList	0, ∞		
		FileName	读写器参数文件名全称
		VersionId	读写器参数版本
		ParamType	读写器参数种类
		ContentLength	文件长度
		VersionContent	版本内容, 按照读写器参数内容直接体现在此字段中



表 40 卡黑名单参数列表（续）

标记	标记出现次数	属性	说明
/>			
</ReaderParam>			
</BlackList>			

## 4.2.4.2.3 计价方案参数

文件名： Price\_jjfa+版本号（yyyymmdd+4 位流水号）.xml

表 41 计价方案参数列表

标记	标记出现次数	属性	说明
<Price>	1, 1		计价方案参数
<Version/>	1, 1		版本信息
<CardType>	1, 1		卡类型参数
<CardTypeInfo	1, ∞		卡类型参数
		CardType	卡类型编号（数字串，4 位）
		CardTypeName	卡类型名称（字符串，非空，小于等于 40 位）
		CardTypeName_E	卡类型名称（英文）（字符串，非空，小于等于 40 位）
		Description	卡类型描述（说明）（字符串，小于等于 100 位）
		ForeGift	押金；单位：分；（整型，[0, 99999999]）
		Charge	售卡费用；单位：分；（整型，[0, 99999999]）
		DateAvailable	启用后的有效期天数，0 表示人工设定，（整型，[0, 99999999]）
		MediaType	票卡采用的介质类型（数字串，2 位），详见下表
		CardAttrib	卡属性参数（字符串，小于等于 2000 位）
		IsSelled	是否可售
		BasicPlanNo	关联的基本价格方案编号（数字串，4 位）
/>			
</CardType>			
<StationInfo	1, ∞		运营点矩阵

表 41 计价方案参数列表（续）

标记	标记出现次数	属性	说明
		Length	站点编号长度（整型，[0, 99]）
		content	站点 1 编号+站点 2 编号+…… 站点 n 编号
/>			
<BasicFare>	1, 1		基本价格
<BasicPlanInfo	1, ∞		每个标记下存储一个基本价格信息
		BasicPlanNo	基本价格方案编号（数字串，4 位）
		BasicPlanName	基本价格方案名称（字符串，非空，小于等于 40 位）
		BasicRule	计费类型（数字串，1 位），详细定义见下表
		ValueType	价格类型（数字串，1 位）， 1：金额 2：次数 3：有效期
		Divide	站间票价列表，对应运营点矩阵的每个站间组合；每个票价长 6 数字字符
/>			
</BasicFare>			
<ReaderParam	1, 1		读写器相关参数
		SrcAddr	源地址
		CreateDate	生成时间
<ReaderParamList	0, ∞		
		FileName	参数文件名全称
		VersionId	参数版本
		ParamType	参数种类
		ContentLength	文件长度
		VersionContent	版本内容，详见读写器相关参数规范
/>			
</ReaderParam >			
</Price>			

## 4.2.4.2.3.1 卡属性参数

卡属性参数是用来指导设备在处理某一类卡时应该进行的操作和处理，由一系列标志字符组成，以下为属性参数信息表示方法：

- a) 卡属性参数为一不定长的 ASCII 字符组成；
- b) 参数可根据需要进行扩展；
- c) 不同的标志参数段通过“\”来分隔；
- d) 参数的第一个字母为大写，表示参数的类型，其后为参数内容。
- e) 当参数内有表示金额的数字时，金额的单位为分
- f) 当参数内有表示日期的数字时，日期的格式为 4 位年+2 位月份+2 位日期，如：20001001 表示 2000 年 10 月 1 日

- g) 参数段内由“参数标识字符” + “参数内容”构成

以下为“参数标识字符”和“参数内容”的说明：

表 42 参数标识字符和参数内容列表

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
单程标志	S	1 位数字	0: 非单程 1: 单程	0	
记名标志	N	1 位数字	0: 不记名 1: 普通记名 2: 个性化记名	0	
退卡余额标志	R	1 位数字	0: 不可退卡余额 1: 可退卡余额	1	
退押金标志	D	1 位数字 + 4 位数字	0: 不可退押金 1: 可退押金	00000	4 位数字表示可退押金的间隔天数
值类型	C	1 位数字 + 8 位数字	0: 无值类型 1: 金额 2: 次数 3: 有效期	100100000	8 位数字表示钱包上限(单位: 分)
充值标志	A	1 位数字 + 12 位数字 (后 12 位数字循环)	0: 不可充值 1: 充金额: 6 位金额值 + 6 位金额 2: 充次数: 4 位次数 + 8 位金额 3: 充有效期: 4 位天数 + 8 位金额	00000000000000	与“值类型”对应, (金额的单位: 分);

表 42 参数标识字符和参数内容列表（续 1）

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
扣费标志	F	1 位数字	0: 不扣费 1: 扣费	1	具体扣费方式按“值类型”进行处理
进出站次序检查	B	1 位数字	0: 不检查 1: 检查	1	
最大预售天数	M	1 位数字 + 3 位数字	0: 不允许预售 1: 允许预售(单位: 天)	0000	3 位数字为可预售的天数
有效起始时间处理	E	1 位数字 + 8 位数字 (日期格式)	0: 检查卡内有效起始时间 1: 检查固定有效起始时间 (当前日期未达到该日期则禁用该卡)	00000000 00	保留未使用
有效终止时间处理	T	4 位数字型字符	第 1 位字符表示检查卡有效时间的方式: 0: 默认检查票卡内有效期开始和结束时间 1: 直接检查卡内有效终止时间 2: 检查卡内有效终止时间+适当延长分钟数 3: 出站检查卡有效截止日期+适当的延长分钟数 第 2-4 位: 适当延长的分钟数	1000	方式 3 的用法解释: 目前主要用于日期票的出站判断时增加延长分钟数判断, 进站的时候仍然是已票卡上的有效期进行检查
积分标志	V	1 位数字	0: 不处理 1: 处理	0	
回收标志	X	1 位数字	0: 不回收 1: 回收	0	
信用标志	Y	1 位数字	0: 不处理 1: 处理	0	

表 42 参数标识字符和参数内容列表 (续 2)

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
提示类型标志	P	1 位数字	0: 无声 1: 鸣叫一声 15: 鸣叫一声读员工卡 16: 鸣叫一声读学生卡 17: 鸣叫一声读老人卡 18: 鸣叫一声读优惠卡 19: 鸣叫一声读特殊卡 2: 鸣叫二声 25: 鸣叫二声读员工卡 26: 鸣叫二声读学生卡 27: 鸣叫二声读老人卡 28: 鸣叫二声读优惠卡 29: 鸣叫二声读特殊卡 3: 鸣叫三声 45: 读员工卡 46: 读学生卡 47: 读老人卡 48: 读优惠卡 49: 读特殊卡	1	设备读卡时对卡种类发出的提示音以便进行识别
透支标志	0	1 位数字 + 8 位数字	0: 不能透支 1: 可透支	00000000	8 位数字为透支上限值 (单位: 分)
定期激活	H	1 位数字+3 位数字+5 位数字	0: 不需定期激活 1: 需要定期激活	00000000	3 位数字表示激活周期 (单位: 天) +5 位数字表示激活费用 (单位: 分)
灯光提示类型标志	L	1 位数字	0: 优惠灯不亮 1: 第一种优惠灯亮 2: 第二种优惠灯亮 3: 第三种优惠灯亮	0	
OTP 有效标志	W	1 位数字	0: OTP 有效取消 1: OTP 有效	0	
手续费收取方式	G	1 位数字 + 10 位数字 (后 10 位数字循环)	0: 不收手续费 1: 收手续费: 3 位起始月数+3 位截至月数 + 4 位金额	00000000 0000	金额单位为分
租金收取方式	I	1 位数字 + 3 位数字+10 位数字 (后 10 位数字循环)	0: 不收租借 1: 收租金: 3 位收租月数+3 位起始月数+3 位截至月数 + 4 位金额	00000000 00000000	金额单位为分

表 42 参数标识字符和参数内容列表（续 3）

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
售卡方式	J	2 位数字	00: 出租 01: 销售 02: 纪念	01	
黑名单检查标志	K	1 位数字	0: 不检查 1: 检查	0	
卡类型显示	Q	不定长	卡类型汉字名称	空	用于带显示功能的深圳通设备显示简要的卡类型名称，通常为 5 个汉字
卡类型激活状态标志	U	1 位数字 + 2 位数字	0: 不检查激活状态 1: 检查激活状态	000	此标志仅用于不能存储全部黑名单的设备，如：车载机、手持机等；具有大容量存储器且可存储全部黑名单的设备不适用，如闸机、充值机等。 2 位数字表示激活月份。 当此标志为 1 时，需对卡进行如下检查： 如果“卡最近使用日期”（不含时分秒）+“激活月份”所得的日期大于或等于设备当前日期时，此卡可在此设备上使用，否则提示乘客需进行激活。
最低可进站余额检查	h	7 位数字	检查标记（1 位）+最低余额（次）（6 位数字）	1000200	检查标记： 0: 不检查；1: 检查 最低余额（次）：以分（次）为单位的卡最低余额

表 42 参数标识字符和参数内容列表（续 4）

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
本站进出滞留时间	n	1 位字符+3 位数字+6 位数字	本站滞留时间检查标识（1 字符）：0：不检查；1：检查； 滞留时间（3 位数字）：本站进出最大滞留时间（以分钟为单位）； 滞留超时费用（6 位数字，以分为单位）（后 9 位数字循环）	0000000 000	本站进出采用特殊的滞留时间检查，超过此时间按指定票价扣费。
激活属性	c	19 位字符	第 1 位字符表示激活级别(1 位)： 0：不需要激活； 1：第一次使用前在自动设备或以上的自动激活 2：第一次使用前需在人工设备级别上激活 第 2 位字符表示判断允许激活时间段方式标志： 0：不需要判断允许激活时间段； 1：根据读取票卡有效开始时间与有效结束时间进行有效期判断（如果激活当前时间不在范围内不可进行激活）； 2：根据票卡有效开始时间开始计算进行判断； （如果激活当前时间不在票卡有效开始时间至票卡有效开始时间+允许激活的最长天数的范围内， 第 3~5 位字符表示允许激活的最长时间天数限制(3 位)：	0	

表 42 参数标识字符和参数内容列表（续 5）

参数说明	参数标识字符	参数内容	参数内容定义	默认值	备注
			<p>第 6 位字符表示激活改有效期方式(1 位):</p> <p>0: 不修改有效期</p> <p>1: 用当前时间修改起始有效期, 结束有效期修改为: 当前时间+有效天数 (pw_fare_cardtype 中 dateavailable)。</p> <p>2: 用当前时间修改起始有效期, 结束有效期修改为: 当前时间+激活后的票卡可用有效期的数值 (按照单位计算)。</p> <p>第 7~12 位表示激活应收取的费用, 单位: 分 (6 位)</p> <p>第 13 位表示收取的费用方式: (1 位)</p> <p>0: 仅能使用现金</p> <p>1: 仅能使用票卡</p> <p>2: 优先票卡, 票卡余额不足, 可采用现金;</p> <p>第 14 位字符表示 OTP 激活标识:</p> <p>0: 不采用 OTP 激活;</p> <p>1: 采用 OTP 激活</p> <p>第 15 位表示激活后的票卡可用有效期的单位:</p> <p>0: 分钟</p> <p>1: 小时</p> <p>2: 天数</p> <p>第 16~19 位表示激活后的票卡可用有效期的数值, 例如: 如果单位为分钟, 1450 则表示此票激活后可在 1450 分钟之内使用。</p>		



#### 4.2.4.2.3.2 票价计算方法

- a) 通过计价方案检索当前的票卡方案、基本价格表、联程规则方案
- b) 根据票卡方案，由卡类型检索对应的计费规则。
- c) 由 基本价格 + 计费规则 计算出票价。
- d) 如果符合联程规则，则在已计算出的票价基础上按规则计算出最终票价。

#### 4.2.4.2.3.3 基本价格表

- a) 基本价格表是一个枚举任意两站之间的票价的票价表。
- b) 运营点矩阵：从运营点的站点排列顺序站点 1 编号+站点 2 编号+……+站点 n 编号，可以获得以下的站点矩阵组合：[站点 1，站点 1]；[站点 1，站点 2]；[站点 1，站点 3]；……；[站点 1，站点 n]；[站点 2，站点 1]；[站点 2，站点 2]；[站点 2，站点 3]；……；[站点 2，站点 n]；……；[站点 n，站点 1]；[站点 n，站点 2]；……；[站点 n，站点 n]。
- c) 基本价格表生成：基本价格中的站间票价列表 (Divide) 是一个连续的票价序列，按顺序对应以上运营点矩阵的每个站间组合，表示两站间的基本票价。两者结合形成基本价格表。

#### 4.2.4.2.3.4 计费规则

计费规则是指一系列对基本价格进行处理或约束的组合。规则的表示采用参数化的表达方式：通过带分隔符号和参数标识的字符串表示计费规则，参数标识可根据需要进行相应的扩充。将多种规则及其处理进行相应的组合，产生一定的计算价格的运算法则，终端设备获得基本价格并按卡类型找到相应的计费规则后，对计费规则进行解释运算得到最终票价。

以下为计费规则表示方法：

- a) 计费规则为一有序的字符串，处理顺序为由左至右，一个完整的计费规则由一段或多段具体的规则构成。
- b) 规则段的构成：分隔符号 + [参数] + [处理]，“参数”和“处理”为可选项，但不可即无“参数”也无“处理”。
  - 1) 分隔符号：“\”，用于分隔两条规则；
  - 2) 参数：参数的第一个字母为大写，表示参数的类型；
  - 3) 处理：当符合参数的定义的条件时，对价格进行运算处理。如果无参数，则表示无条件对价格进行处理。
- c) 规则内含义说明：

- 1) 时间表示法：年（带世纪）+月+日+时+分：yyyymmddhhmm，例：200001011259 表示 2000 年 1 月 1 日 12 点 59 分；每日的时间为 24 小时制，从 00 点 00 分到 23 点 59 分；时间段包含段的边界时间
- 2) 数值表示法：数值为一整数，当处理为“加”或“减”时最后两位表示实际数值的小数部分，当处理为“乘”时数值表示百分数。例：“加”或“减”运算时 1000 表示 10.00，1 表示 0.01，“乘”运算时 10 表示 10%，110 表示 110%

d) 参数表：

表 43 计费规则参数列表

参数类型	参数格式	参数含义	备注
基本价格规则	P<op>	从基本价格方案定义的基本价格表查找站间价格	规则的第一项必为此参数
	X<op>	由外部输入价格（手工输入、计价器输入等）	
	0<op>	单一价格，即基本价格方案中定义的单位价格	
年龄约束规则	A+[年龄段]+[年龄段]+...	<p>年龄段定义：aff&lt;op&gt;</p> <p>a 为年龄段分隔符，如果持卡人年龄（从卡上信息获取）在 f 岁至 f 岁之间，则对“基本价格规则”中的“P”、“X”或“0”进行年龄规则中定义的&lt;op&gt;处理，并不执行“基本价格规则”中的&lt;op&gt;处理，如果年龄不在所有年龄段中则按“基本价格规则”中的&lt;op&gt;产生基本价格</p> <p>f 为三位定长数字，不足三位的在前面补 0</p>	如果有此规则，则此参数必为第二项对于非记名卡此规则无效
滞留时间规则	T+[滞留时段]+[滞留时段]+...	<p>滞留时段定义：tf-f:&lt;op&gt;</p> <p>t 为滞留时段分隔符，如果滞留时间在 f 分钟至 f 分钟内（减去合法乘坐时间的 N 倍（参数），当获取的乘坐时间为 0 或空时，该规则失效），就执行&lt;op&gt;定义的操作</p> <p>f 为不定长数字，f 最大值：99999，单位：分钟</p>	

表 43 计费规则参数列表（续）

参数类型	参数格式	参数含义	备注
特殊时段规则	E + [日期或星期规则] + [日期或星期规则] + ...	规则内的日期规则或星期规则由左至右优先级依次降低，当处理时间满足其中一个规则，则结束本规则段处理。 日期规则： dyyyyymmddyyyyymmddhhmmhhmm<op> d 指按日期定义，如果当前时间在 yyyy 年 mm 月 dd 日至 yyyy 年 mm 月 dd 日中任意一天的 hh 点 mm 分到 hh 点 mm 分内（不包含最后一日内的时段），则执行本规则定义的处理并结束本规则段的处理；否则处理下一规则。<op>为处理。 星期规则： wffhmmhhmm<op> w 指按星期定义，如果当前时间在星期 f 至星期 f 中任意一天的 hh 点 mm 分到 hh 点 mm 分内，则执行本规则定义的处理并结束本规则段的处理；否则处理下一规则。<op>为处理 f 为一位数字，表示星期中的一天，星期日为一星期中的第一天，星期日至星期六依次用数字 0 - 6 表示	当定义的第二时间点小于第一个时间点时，第二时间点为下一天的时间点
			为便于设备进行时间处理，日期规则中不包含最后一日内的时段
最高限价规则	Hf	如果计算后的价格大于最高限价定义的价格 f，则将价格设置为最高限价定义的价格 f，f 为不定长数字	用于设置长途优惠
最低限价规则	Lf	如果计算后的价格小于最低限价定义的价格 f，则将价格设置为最低限价定义的价格 f，f 为不定长数字	
尾程优惠规则	F	如果卡内余额小于计算后应扣除的价格，则将价格设置为卡内余额	
余额全收规则	R	如果卡内余额大于计算后的价格，则将价格设置为卡内余额	用于普通单程票

注：

- 1) <op>为处理；
- 2) 基本价格规则、年龄约束规则、滞留时间规则、最高限价、最低限价、尾程优惠和余额全收规则在计费规则中最多只能出现一次；
- 3) 尾程优惠规则和余额全收规则不能同时存在，只能选择一种；
- 4) 最高限价、最低限价、尾程优惠（或余额全收）为固定排列在计费规则字符串的最后，次序不能改变；
- 5) 年龄规则中的年龄从卡内证件号码相应部分的出生日期计算所得。

- 6) 如果计费规则中包含滞留时间规则, 则对于滞留时间大于最后一个时段定义的最大时间的交易, 按最后一个时段定义的规则进行处理;
- 7) 特殊时段规则的设置原则:
- 8) 对于通常每日都使用的特殊时段 (如每天 22:00 至第二日 05:59), 用星期规则表示为 w0622000559<op>, 即星期天至星期六每天 22:00 至第二日 05:59。
- 9) 对于相互间有完全包含关系的时段, 如果只能执行其中的一种规则, 则将两规则放置在同一规则段内并按优先次序进行排列; 如果要叠加执行, 则将两规则放置在不同的规则段内。如对于规则 w0622000559<op>和规则 d200110012001100700002359<op>, 如果当前时间为 2001 年 10 月 6 日星期六 09:00, 当前只能执行第二个规则, 则规则段的表示方法为: \Ed200110012001100700002359<op>w0622000559<op>, 如果两个规则要叠加执行, 则规则段的表示方法为:
- \Ed200110012001100700002359<op>\Ew0622000559<op>。

## e) 处理表

表 44 计费处理列表

处理说明	格式	处理的含义	备注
算术运算	+f	$\text{fare} = \text{fare} + f$	整数, 以“分”为单位
	-f	$\text{fare} = \text{fare} - f$	整数, 以“分”为单位
	*f	$\text{fare} = \text{fare} * f$	整数形式的百分数 (如' 90' 表示' 0.9')
	=f	$\text{fare} = f$	整数, 以“分”为单位, 表示将价格设置为值 f
禁止使用	s	禁止使用此卡	对计费规则处理结束, 跳出处理
	a+f	禁止使用此卡并进行人工罚款 罚款金额 = f	计算完此规则后, 继续进行下一规则的处理
	a*f	禁止使用此卡并进行人工罚款 罚款金额 = fare*f	计算完此规则后, 继续进行下一规则的处理
对此计费方案基本价格处理	pf	$\text{fare} = \text{fare} + (p * f)$ , p (小写字母) 为此计费方案的基本价格 (对基本价格进行处理后的价格)	在此计费规则方案基本价格基础上进行加价 (f 为正数) 或打折 (f 为负数)

注:

- 1) fare 为计算后的价格;
- 2) f 为一数值。

## f) 计费规则范例

\P\*90\Tt61-90:\*105t91-120:+200t121-480:p10

\Ew0207002359\*105d199901011999010322000559sd6000002359=200

\Ew1521300530p10\+200\\*110\p10\p-10\H1000\L200 \F

含义如下:

表 45 计费规则范例解释列表

字符串	含义	计算公式
\P*90	从基本价格表查找两站间基本价格, 此计费规则基本价格(p)为两站间基本价格乘以 0.90	$p=P*0.90$
\Tt61-90:*105t 91-120:+200t12 1-480:p10	滞留 61-90 分钟在前面处理后价格的基础上加收 5%	$fare=fare*1.05$
	滞留 91-120 分钟加收 2 元	$fare=fare+2.00$
	滞留 121-480 分钟或 480 分钟以上, 在此计费规则基本价格(p)的基础上加收 10%	$fare=fare+p*0.10$
\Ew0207002359* 105d1999010119 99010322000559 s d6000002359=20 0	如果当前时间在星期日至星期二每天的 07:00 至 23:59 内, 在前面处理后价格的基础上加收 5%, 并结束本特殊时段规则处理	$fare=fare*1.05$
	如果当前时间在 1999 年 1 月 1 日 22:00 至 1999 年 1 月 2 日 05:59 和 1999 年 1 月 2 日 22:00 至 1999 年 1 月 3 日 05:59 内时, 此种卡类型禁止使用, 并结束本特殊时段规则处理	禁止使用
	星期六和星期日全天将当前计算后的价格设置为 2 元	$fare=2.00$
\Ew1521300530p 10	星期一至星期五每天 21:30 至每二天早上 05:30, 在此计费规则基本价格(p)的基础上加收 10%	$fare=fare+p*0.10$
\+200	在前面处理后价格的基础上价格加 2 元	$fare=fare+2.00$
\*110	在前面处理后价格的基础上加收 10%	$fare=fare*1.10$
\p10	在此计费规则基本价格(p)的基础上加收 10%	$fare=fare+p*0.10$
\p-10	在此计费规则基本价格(p)的基础上减收 10%	$fare=fare+p*(-0.10)$
\H1000	最高限价: 10.00 元	如果 $fare > 10$ 则 $fare=10$
\L200	最低限价: 2.00 元	如果 $fare < 2$ 则 $fare = 2$
\F	尾程优惠	如果卡内余额 < $fare$ 则 $fare = \text{卡内余额}$

## 4.2.4.2.3.5 联程规则处理流程

- a) 持卡人在运营公司的运营线路上，进行一次地铁进站操作，在卡的交易起始记录区写上交易起始地点、交易计价方案（线路）、交易时间。进行一次公共交通的刷卡消费（地铁的出站，公交的消费），设备在卡的交易应用信息区写入本设备当前企业编号、计价方案编号（线路编号）、最近使用时间、最近使用地点。
- b) 持卡人进行第二次消费；
- c) 设备根据卡上记录的前次消费的运营公司和采用的计价方案编号（线路号）、前次交易地点，与本设备内保存的联程方案中允许的前次消费交易的运营公司、计价方案编号（线路号）、交易地点进行比较；
- d) 如果卡上信息与联程方案一致，则继续以下流程，否则结束本流程并按正常计费方式进行计费；
- e) 对于地铁的交易，读取卡上交易起始记录区的计价方案编号（线路号）、交易地点和联程方案中允许的交易地点进行比较；（注：本次进站线路不参与比较，因为闸机进站时暂时没有往卡上写线路）
- f) 如果卡上信息与联程方案一致，则继续以下流程，否则结束本流程并按正常计费方式进行计费；
- g) 判断两次消费间的时间间隔是否在联程规则定义的最小和最长时间间隔内以及是否满足最低金额限制；
- h) 设备根据当前使用的卡类型，在规则中查找适用于本卡类型的规则，如果有则继续以下流程，如果没有则说明此卡类型无联程处理，对此卡类型按正常的计费方式进行计费；
- i) 按计价方案进行相应的计费规则计算，计算出本次消费的正常收费价格；
- j) 设备根据联程规则定义，对本次消费的基本价格按规则进行相应处理，得出将优惠的金额（根据规则定义也可能是要加价的金额）；

注：本处的基本价格指本次使用的卡种的基本价格，如 A 站到 B 站的运营站点价格表上的价格为 3 元，假设成人卡的计费规则（前一步骤中使用的计费规则）为 9 折，则本处使用的基本价格为 2.7 元，学生卡的计费规则为 5 折，则本处使用的基本价格为 1.5 元。以下举例说明：

表 46 联程优惠规则列表

联程优惠规则	优惠金额的计算方法		实际应优惠金额		应收票价	
	成人卡	学生卡	成人卡	学生卡	成人卡	学生卡
减收 1 元规则	-1	-1	1 元	1 元	1.7 元	0.5 元
减收 20%规则（打八折）	$3*0.9*(-0.2)$	$3*0.5*(-0.2)$	0.54 元	0.3 元	2.16 元	1.2 元

将正常收费价格加上或减去计算出的金额，得本次实际应从卡上扣除的消费金额。如果出现优惠金额大于正常收费价格时，车载机将按正常收费价格从卡上扣费，即不减去联程优惠金额

#### 4.2.4.2.4 运营点参数

文件名：Station\_yyd+版本号（yyyymmdd+4 位流水号）.xml

表 47 运营点参数列表

标记	标记出现次数	属性	说明
<Station>	1, 1		
<Version/>	1, 1		版本信息
<StationBaseInfo>	1, 1		车站信息
<BasicInfo	1, ∞		
		Id	车站编号；（数字串，10 位）
		Name	车站名称（中文）（字符串，非空，小于等于 80 位）
		StationName_En	车站名称（英文）（字符串，非空，小于等于 80 位）
		StationNo_Ex	车站扩展编号；（数字串，10 位，此处与车站编号一致）
		Default_StoreId	默认票卡位置代码（按运营点售票或 TVM 上选站售票时可以试用，表示存储在票卡上的代码）（数字串，4 位）
/>			
</StationBaseInfo>			
<StationInfo>	1, 1		车站信息
<BasicInfo>	1, ∞		
		StationId	车站编号（数字串，4 位）
		StationName	车站名称（字符串，非空，小于等于 80 位）

表 47 运营点参数列表（续）

标记	标记出现次数	属性	说明
		StoreId	票卡位置代码（存放在卡上的代码）（数字串，4 位）
		StationBaseId	归属运营点（计算票价用）
/>			
</StationInfo>			
<LineBaseInfo>	1, 1		线路信息
<BasicInfo>	1, ∞		
		Id	线路编号（数字串，3 位）
		Name	线路名称（字符串，非空，小于等于 40 位）
<StationList			
		StationBaseId	此线路所包含的站点编号（数字串，10 位）
		SortId	此线路所包含的站点的方向排序序号（整型，[0, 999] 位）
/>			
</BasicInfo>			
</LineBaseInfo>			
<ReaderParam	1, 1		读写器相关参数
		SrcAddr	源地址
		CreateDate	生成时间
< ReaderParamList	0, ∞		
		FileName	参数文件名全称（0201, 0204）
		VersionId	参数版本
		ParamType	参数种类
		ContentLength	文件长度
		VersionContent	
/>			
</ReaderParam >			
</Station>			

## 4.2.4.2.5 TVM 售票时间参数

文件名：TVMTIME\_spsj+版本号（yyyymmdd+4 位流水号）.xml



表 48 TVM 售票时间参数列表

标记	标记出现次数	属性	说明
<TVMTIME >	1, 1		
<Version/>	1, 1		版本信息
<StationInfo	1, 1		运营点矩阵
		Length	规约的站点编号位数（整型，[0, 99]）
		Content	站点 1 编号+站点 2 编号+.....+ 站点 n 编号；（数字串，n*10 位）
/>			
<SaleTime>	1, 1		TVM 售票时间
<StartPlan>	1, 1		开始售票时间
<PlanInfo	1, ∞		
		PlanId	开始售票时间方案编号（数字 串，4 位）
		PlanName	开始售票时间方案名称（字符 串，非空，小于等于 40 位）
		PlanContent	开始售票时间序列；多个 hhmmss 连续排列（数字串，n*6 位）
/>			
</StartPlan>			
<StopPlan>	1, 1		停止售票时间
<PlanInfo	1, ∞		
		PlanId	停止售票时间方案编号（数字 串，4 位）
		PlanName	停止售票时间方案名称（字符 串，非空，小于等于 40 位）
		PlanContent	停止售票时间序列；多个 hhmmss 连续排列（数字串，n*6 位）
/>			
</StopPlan>			
<DatePlan>	1, 1		
<DatePlanInfo	1, ∞		

表 48 TVM 售票时间参数列表（续）

标记	标记出现次数	属性	说明
		DateType	1: 默认情况下的 TVM 售票时间 (DateValue 忽略); 2: 星期日到星期六 (DateValue = 0-6) 中某一天的 TVM 售票时间。 3: 指定具体日期 (DateValue = YYYYMMDD) 的 TVM 售票时间。 优先级为 3/2/1
		DateValue	依照 DateType 所决定的类型设置
		StopPlan	使用的停止售票时间方案编号 (数字串, 4 位)
		StartPlan	使用的开始售票时间方案编号 (数字串, 4 位)
/>			
</DatePlan>			
</SaleTime>			
<Signature>	1, 1		电子签名
		SignatureType	电子签名类型 1: 签名 (MD5) 2: 签名 (MD5) + 加密
		SignatureData	电子签名数据
/>			
</TVMTime>			

4.2.4.2.5.1 运营点矩阵格式说明

从运营点的站点排列顺序站点 1 编号+站点 2 编号+.....+站点 n 编号, 可以获得以下的站点矩阵组合: [站点 1, 站点 1]; [站点 1, 站点 2]; [站点 1, 站点 3]; .....; [站点 1, 站点 n]; [站点 2, 站点 1]; [站点 2, 站点 2]; [站点 2, 站点 3]; .....; [站点 2, 站点 n]; .....; [站点 n, 站点 1]; [站点 n, 站点 2]; .....; [站点 n, 站点 n]。

4.2.4.2.5.2 开始售票时间格式说明

是一系列开始售票时间的排列, 每个时间按顺序对应运营点矩阵中形成的站点矩阵组合, 规定该组合两个站点间从前者到后者的售票开始时间。

4.2.4.2.5.3 停止售票时间格式说明

是一系列停止售票时间的排列，每个时间按顺序对应运营点矩阵中形成的站点矩阵组合，规定该组合两个站点间从前者到后者的售票停止时间。

4.2.4.2.5.4 TVM 运营时间（方案）格式说明

DatePlan 指定了具体某日或每周某天的 TVM 开始售票时间和停止售票时间，不在以上指定日期的则采用默认的 TVM 运营时间。

4.2.4.2.6 TVM 售票地图参数

文件名：StationMap\_TVM +版本号（yyyymmdd+4 位流水号）.xml

表 49 TVM 售票地图参数列表

标记	标记出现次数	属性	
< StationMap >	1, 1		
<Version/>	1, 1		
<ResolutionInfo>	1 , ∞		分辨率信息
		ResolutionId	分辨率编号
		Content	分辨率内容，如 1024*768
< GlobalInfo	1, 1		
		ImageName	图片名称
		StationCount	站点数
/>			
<StationInfo	1, ∞		
		StationId	站点编号
		IsUse	是否正在运营
		X	X 坐标
		Y	Y 坐标
		W	宽
		H	高
		Rx	圆角矩形 X
		Ry	圆角矩形 y
/>			
</ GlobalInfo >			
<LineInfo >	1, 1		
<MapInfo	1, ∞		
		LineId	线路号
		ImageName	图片名称
		IsUse	该线路是否正在运营
		StationCount	站点总数

表 49 TVM 售票地图参数列表 (续 1)

标记	标记出现次数	属性	
/>			
<StationInfo	1, ∞		
		StationId	站点编号
		IsUse	是否正在运营
		X	X 坐标
		Y	Y 坐标
		W	宽
		H	高
		Rx	圆角矩形 X
		Ry	圆角矩形 y
/>			
<MapInfo/>			
</ LineInfo >			
< /ResolutionInfo>			
</ StationMap >			
< StationMap >	1, 1		
<Version/>	1, 1		
<ResolutionInfo>	1, ∞		分辨率信息
		ResolutionId	分辨率编号
		Content	分辨率内容, 如 1024*768
< GlobalInfo	1, 1		
		ImageName	图片名称
		StationCount	站点数
/>			
<StationInfo	1, ∞		
		StationId	站点编号
		IsUse	是否正在运营
		X	X 坐标
		Y	Y 坐标
		W	宽
		H	高
		Rx	圆角矩形 X
		Ry	圆角矩形 y
/>			
</ GlobalInfo >			
<LineInfo >	1, 1		
<MapInfo	1, ∞		

表 49 TVM 售票地图参数列表（续 2）

		LineId	线路号
		ImageName	图片名称
		IsUse	该线路是否正在运营
		StationCount	站点总数
/ >			
<StationInfo	1, ∞		
		StationId	站点编号
		IsUse	是否正在运营
		X	X 坐标
		Y	Y 坐标
		W	宽
		H	高
		Rx	圆角矩形 X
		Ry	圆角矩形 y
/ >			
<MapInfo/>			
</ LineInfo >			
< /ResolutionInfo>			
</ StationMap >			

## 4.2.4.2.7 种类参数

文件名：SpeciesParam\_zl+版本号（yyyymmdd+4 位流水号）.xml

表 50 种类参数列表

标记	标记出现次数	属性	说明
<SpeciesParam>	1, 1		
<Version/>	1, 1		版本
<Species	0, ∞		种类信息
		SpeciesId	种类编号
< ControlParm	1, 1		基本参数信息
		根据参数信息动态生成	
/ >			
< SpeciesCode	1, 1		设备字典数据
/ >			
/Species >			
/ >			

4.2.4.2.8 设备参数

文件名： EquParam\_ sb+版本号（yyyymmdd+4 位流水号）.xml

表 51 设备参数列表

标记	标记出现次数	属性	说明
< EquParam >	1, 1		
<Version/>	1, 1		版本
< Equipment	0, ∞		设备信息
		EquipmentId	设备编号
<BasicInfo	1, 1		
		SystemId	系统编号
		SpeciesId	种类编号
		NetId	网络地址
		IfAspect	
		AuthenticType	认证方式
		DefaultMode	默认模式
		GroupId	组编号
		IfDynamicIp	是否动态 IP
/>			
< ControlParm	1, 1		基本参数信息
		根据参数信息动态生成	根据不同的设备定义的参数信息动态生成，采用（TimeLimit=" 30" ）key-value 方式定义
/>			
< EquComponentParam	1, 1		部件参数
/>			
</ EquParam >			

## 4.2.4.3 交易数据

## 4.2.4.3.1 交易类型说明

表 52 交易类型列表

交易类型与交易格式代号对照表			
交易名称	交易类型	交易格式代号	备注
售卡	01	232	
售出站票	03	232	
售行李票	05	301	
现金充值	11	241	
超程处理	15	241	
地铁消费起始	21	211	
地铁消费(结算)	22	221	
地铁消费(联程结算)	22	223	
储值卡购票	41	221	
好卡退卡	71	391	
非及时退卡	72	251	
挂失	74	252	
退余额	75	391	
黑名单卡锁定	77	221	
卡锁定解除	78	252	
滞留超时处理	83	262	
清交易起始标识	84		
钱包 2 清交易起始标志	85		
地铁无入站处理	89	262	
行政罚款	91	301	
行政支付	92		
激活	96	401	
售儿童票	97	301	

## 4.2.4.3.2 售卡交易

包括售卡(01)、发售出站票 (03)

表 53 售卡交易列表

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC (读写器返回 10 位, 暂定取前 8 位)
卡号	CHAR	20	转换为 20 位的数字字符 (不足 20 位, 后补空格)
交易类型	CHAR	2	参见 2.3.2
交易地点	CHAR	10	计费运营点 (换乘站只有唯一的一个计费运营点, 不足 10 位前补 0)
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	默认值 000000
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	设备本地累计交易序号
交易金额	CHAR	8	钱包增加的金额, 单位: 分
交易值	CHAR	8	钱包增加值, 单位: 分/次
押金	CHAR	8	储值卡中押金金额
费用	CHAR	8	手续费, 默认值 0
卡交易序号	CHAR	5	票卡扣款计数
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负, 直接用绝对值累加。
有效起始时间	CHAR	14	YYYYMMDD HH24MISS
有效终止时间	CHAR	14	YYYYMMDD HH24MISS
目的站点	CHAR	10	单程票站点售票时目的站点编码
前次交易终端代号	CHAR	9	填写默认值全 0
前次交易类型	CHAR	2	填写默认值全 0
前次交易时间	CHAR	14	填写默认值 20000101000000, YYYYMMDDHHMMSS
前次交易金额	CHAR	8	填写默认值全 0
前次交易操作员号	CHAR	6	填写默认值全 0
SAM 卡号	CHAR	20	SAM 卡 ID 号 (不足 20 位, 后补空格)
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	读写器返回卡类型中前 2 位
交易小类	CHAR	2	默认填 “00”
票卡物理卡号	CHAR	16	票卡物理 ID, 读写器返回 20 位中取前 16 位
测试标记	CHAR	1	0-正常交易; 1-测试交易
灰交易标记	CHAR	1	0-确定交易; 1-灰交易 (不确定交易)
物理类型	CHAR	2	00:单程票, 01: 非单程票



表 53 售卡交易列表（续）

卡版本号	CHAR	2	默认值全 0
卡子类型	CHAR	2	读写器返回卡类型中后 2 位
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.3 现金充值交易

表 54 现金充值交易列表

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC（读写器返回 10 位，暂定取前 8 位）
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	参见 2.3.2
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	默认值 000000
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	设备本地累计交易序号
交易金额	CHAR	8	钱包增加对应的金额，单位：分
交易值	CHAR	8	钱包增加值，单位：分/次
卡余额	CHAR	8	卡内当前值
卡交易序号	CHAR	5	票卡充值计数
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
前次交易终端代号	CHAR	9	读卡分析中前次线路站点+前次设备类型+前次设备编号
前次交易类型	CHAR	2	默认值 00
前次交易时间	CHAR	14	读卡分析中前次交易时间
前次交易金额	CHAR	8	默认值全 0

表 54 现金充值交易列表（续）

名称	类型	长度(Byte)	注释
前次交易操作员号	CHAR	6	默认值全 0
SAM 卡号	CHAR	20	SAM 卡 ID 号（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	读写器返回卡类型中前 2 位
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	票卡物理 ID，读写器返回 20 位中取前 16 位
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
前次充值全局设备 ID	CHAR	9	转换为 9 位的数字字符
前次充值交易类型	CHAR	2	默认值全 0
前次充值交易时间	CHAR	14	默认值全 0
前次充值交易金额	CHAR	8	默认值全 0
前次充值操作员号	CHAR	6	默认值全 0
前次充值交易序号	CHAR	5	默认值全 0
物理类型	CHAR	2	00:单程票, 01: 非单程票
卡版本号	CHAR	2	默认值全 0
卡子类型	CHAR	2	读写器返回卡类型中后 2 位
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.4 更新交易

包括出站超时更新(83)、无入站更新(89)

表 55 更新交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
更新时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
更新金额	CHAR	8	与交易值保持一样
交易值	CHAR	8	钱包减少值，单位：分/次
费用	CHAR	8	
卡余额	CHAR	8	卡当前值
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
卡当前进出站标志	CHAR	3	
前次进出站时间	CHAR	14	YYYYMMDDHHMMSS
前次进出站终端号	CHAR	9	
前次进出站站点	CHAR	10	
前次交易全局设备 ID	CHAR	9	转换为 9 位的数字字符
前次交易类型	CHAR	2	
前次交易时间	CHAR	14	
前次交易金额	CHAR	8	
前次交易操作员号	CHAR	6	
SAM 卡号	CHAR	20	SAM 卡 ID 号（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPUA 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	正常交易：1-测试交易

表 55 更新交易（续）

名称	类型	长度(Byte)	注释
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.5 单程票超乘处理

表 56 单程票超乘处理交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	与交易值一样
交易值	CHAR	8	钱包增加值，单位：分
卡余额	CHAR	8	
卡交易序号	CHAR	5	

表 56 单程票超乘处理交易（续）

名称	类型	长度(Byte)	注释
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
前次交易终端代号	CHAR	9	
前次交易类型	CHAR	2	
前次交易时间	CHAR	14	
前次交易金额	CHAR	8	
前次交易操作员号	CHAR	6	
SAM 卡号	CHAR	20	SAM 卡 ID 号（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
前次充值全局设备 ID	CHAR	9	转换为 9 位的数字字符
前次充值交易类型	CHAR	2	
前次充值交易时间	CHAR	14	
前次充值交易金额	CHAR	8	
前次充值操作员号	CHAR	6	
前次充值交易序号	CHAR	5	
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.6 清起始标记

包括清起始标记（84），钱包 2 清交易起始标志（85）

表 57 清起始标记

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
更新时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
更新金额	CHAR	8	与交易值保持一致
交易值	CHAR	8	钱包减少值，单位：分/次
费用	CHAR	8	
卡余额	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
卡当前进出站标志	CHAR	3	
前次进出站时间	CHAR	14	YYYYMMDDHHMMSS
前次进出站终端号	CHAR	9	
前次进出站站点	CHAR	10	
前次交易全局设备 ID	CHAR	9	转换为 9 位的数字字符 单程票记录售票信息
前次交易类型	CHAR	2	单程票记录售票信息
前次交易时间	CHAR	14	单程票记录售票信息，YYYYMMDDHHMMSS
前次交易金额	CHAR	8	单程票记录售票信息
前次交易操作员号	CHAR	6	单程票记录售票信息
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”

表 57 单程票超乘处理交易（续）

名称	类型	长度(Byte)	注释
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.7 消费交易-起始

表 58 消费起始交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	

表 58 消费起始交易（续）

名称	类型	长度(Byte)	注释
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.8 消费交易-普通结算

表 59 普通消费交易结算

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	



表 59 普通消费交易结算（续 1）

名称	类型	长度(Byte)	注释
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	钱包实际减少值（折扣后的票价，或者次票的时候为次数），单位：分/次
交易值	CHAR	8	基本价格（两种之间的票价）
卡余额	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
进站时间	CHAR	14	YYYYMMDD HH24MISS
进站全局设备 ID	CHAR	9	转换为 9 位的数字字符
进站站点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
前次交易全局设备 ID	CHAR	9	转换为 9 位的数字字符
前次交易类型	CHAR	2	
前次交易时间	CHAR	14	
前次交易金额	CHAR	8	
前次交易操作员号	CHAR	6	
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	

表 59 普通消费交易结算（续 2）

名称	类型	长度(Byte)	注释
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.9 消费交易-联程结算

表 60 联程交易结算

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	钱包实际减少值（折扣后的票价-换乘优惠价格），单位：分
交易值	CHAR	8	基本价格（两种之间的票价）
卡余额	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
进站时间	CHAR	14	YYYYMMDD HH24MISS
进站全局设备 ID	CHAR	9	转换为 9 位的数字字符

表 60 联程交易结算（续）

名称	类型	长度(Byte)	注释
进站站点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
前次交易全局设备 ID	CHAR	9	转换为 9 位的数字字符
前次交易类型	CHAR	2	
前次交易时间	CHAR	14	
前次交易金额	CHAR	8	
前次交易操作员号	CHAR	6	
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
换乘企业编号	CHAR	3	
换乘线路编号	CHAR	4	
换乘最近使用时间	CHAR	14	
换乘最近使用地点	CHAR	10	
联程优惠金额	CHAR	8	
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC 0001: 一卡通； 0002: 银联； 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.10 退卡交易

表 61 退票交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	钱包实际减少值，单位为分/次
交易值	CHAR	8	实际的退卡的金额（钱包对应的退款+押金-费用）
卡余额	CHAR	8	
押金	CHAR	8	
费用	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
姓名	CHAR	16	后补空格
性别	CHAR	1	
证件号码	CHAR	20	后补空格
证件类型	CHAR	2	
客户 PIN 密文	CHAR	16	
有效起始时间	CHAR	14	
有效终止时间	CHAR	14	
信用标志	CHAR	2	
积分	CHAR	8	
前次交易全局设备 ID	CHAR	9	转换为 9 位的数字字符，单程票记录前次出站信息
前次交易类型	CHAR	2	单程票记录前次出站信息
前次交易时间	CHAR	14	单程票记录前次出站信息，YYYYMMDDHHMMSS
前次交易金额	CHAR	8	单程票记录前次出站信息
前次交易操作员号	CHAR	6	单程票记录前次出站信息
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
透支金额	CHAR	8	
卡类型	CHAR	2	

表 61 退票交易（续）

交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.11 激活

表 62 激活交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
激活时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	钱包改变值，单位：分/次

表 62 激活交易（续）

交易值	CHAR	8	钱包改变值，单位：分/次
费用	CHAR	8	
卡当前值	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
原有效开始时间	CHAR	14	
原有效结束时间	CHAR	14	
新有效开始时间	CHAR	14	
新有效结束时间	CHAR	14	
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充分散因子 当票卡为 M1 卡时此数据字段填充 CARD SN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.12 黑名单锁卡

表 63 黑名单锁卡交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC（读写器返回 10 位，暂定取前 8 位）
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	参见 2.3.2
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	默认值 000000
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	设备本地累计交易序号
交易金额	CHAR	8	钱包减少值，单位：分/次
交易值	CHAR	8	钱包减少值，单位：分/次
卡余额	CHAR	8	
卡交易序号	CHAR	5	票卡扣款计数
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
进站时间	CHAR	14	YYYYMMDD HH24MISS
进站全局设备 ID	CHAR	9	转换为 9 位的数字字符
进站站点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
前次交易全局设备 ID	CHAR	9	读卡分析中前次线路站点+前次设备类型+前次设备编号
前次交易类型	CHAR	2	默认值 00
前次交易时间	CHAR	14	读卡分析中前次交易时间
前次交易金额	CHAR	8	默认值全 0
前次操作员 ID	CHAR	6	默认值全 0
SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	读写器返回卡类型中前 2 位
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	票卡物理 ID，读写器返回 20 位中取前 16 位
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:单程票，01：非单程票
卡版本号	CHAR	2	默认值全 0
卡子类型	CHAR	2	读写器返回卡类型中后 2 位
城市代码	CHAR	4	默认填“0000”

表 63 黑名单锁卡交易（续）

名称	类型	长度(Byte)	注释
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000：ACC； 0001：一卡通； 0002：银联； 其他：预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.13 黑名单解锁

包括卡锁定解除（78），挂失（74）

表 64 黑名单解锁交易

名称	类型	长度(Byte)	注释
交易校验码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	钱包增加值，单位：分/次
交易值	CHAR	8	钱包增加值，单位：分/次
卡余额	CHAR	8	
押金	CHAR	8	
费用	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	投入使用以来的总交易额。交易金额不管正负，直接用绝对值累加。
姓名	CHAR	16	
性别	CHAR	1	
证件号码	CHAR	20	
证件类型	CHAR	2	
客户 PIN 密文	CHAR	16	



表 64 黑名单解锁交易（续）

SAM 卡号	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
透支金额	CHAR	8	
卡类型	CHAR	2	
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPU 卡时此数据字段填充充分散因子 当票卡为 M1 卡时此数据字段填充 CARDSN 其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
物理类型	CHAR	2	00:Mifare Ultralight 卡 01:Mifare One 1K 卡 02:Mifare One 4K 卡 04:CPU 卡
卡版本号	CHAR	2	
卡子类型	CHAR	2	
城市代码	CHAR	4	默认填“0000”
行业代码	CHAR	4	默认填“0000”
SAM 卡设备号	CHAR	12	数字字符，不足 12 位，后补空格
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000：ACC； 0001：一卡通； 0002：银联； 其他：预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.3.14 行政罚款

表 65 行政罚款交易

名称	类型	长度(Byte)	注释
交易验证码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	收取的罚金
交易值	CHAR	8	收取的罚金
费用	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	
行政处理原因	CHAR	2	见行政处理原因编码表
行政处理描述	CHAR	30	后补空格
SAM 卡 ID	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同
卡类型	CHAR	2	默认填“00”
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPUA 卡时此数据字段填充分散因子，其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000：ACC； 0001：一卡通； 0002：银联； 其他：预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

表 66 行政处理原因编码表

行政罚款原因	01	票务违章
	02	违规使用车票
	03	冒用车票证件
	99	其它
乘客事务退款原因	01	TVM 卡币
	02	TVM 卡票
	03	TVM 少找币
	04	TVM 少出票
	05	设备发售无效票
	06	设备充值失败
	07	储值卡已扣值但发售未完成
	99	其它

## 4.2.4.3.15 行政支付

表 67 行政支付交易

名称	类型	长度(Byte)	注释
交易验证码	CHAR	8	TAC
卡号	CHAR	20	转换为 20 位的数字字符（不足 20 位，后补空格）
交易类型	CHAR	2	
交易地点	CHAR	10	计费运营点（换乘站只有唯一的一个计费运营点，不足 10 位前补 0）
全局设备 ID	CHAR	9	转换为 9 位的数字字符
操作员 ID	CHAR	6	
交易时间	CHAR	14	YYYYMMDD HH24MISS
终端交易序号	CHAR	10	
交易金额	CHAR	8	支付的金额
交易值	CHAR	8	支付的金额
费用	CHAR	8	
卡交易序号	CHAR	5	
设备交易总金额	CHAR	10	
行政处理原因	CHAR	2	见行政处理原因编码表
行政处理描述	CHAR	30	后补空格
SAM 卡 ID	CHAR	20	SAM 卡 ID 号；（不足 20 位，后补空格）
SAM 卡交易序号	CHAR	10	当票卡为 CPU 卡时此数据为 SAM 卡交易序号 其他票卡时与终端交易序号相同

表 67 行政支付交易（续）

名称	类型	长度(Byte)	注释
卡类型	CHAR	2	默认填“00”
交易小类	CHAR	2	默认填“00”
票卡物理卡号	CHAR	16	当票卡为 CPUA 卡时此数据字段填充分散因子，其他情况为票卡物理号
测试标记	CHAR	1	0-正常交易；1-测试交易
灰交易标记	CHAR	1	0-确定交易；1-灰交易（不确定交易）
特定行业分隔	CHAR	1	固定为 TAB 键
票卡发行方标识	CHAR	4	0000: ACC; 0001: 一卡通; 0002: 银联; 其他: 预留
特定行业数据	CHAR	可变	根据票卡发行方标识，按照发行方交易格式组织交易，长度可变
CRC16 校验	CHAR	5	

## 4.2.4.4 审计数据

## 4.2.4.4.1 AGM 寄存器数据

表 68 AGM 寄存器

序号	寄存器 ID	名称	值标识	说明
1		进闸人数		成功进闸刷卡后该票卡类型人数和总人数累加
2		出闸人数		成功出闸刷卡后该票卡类型人数和总人数累加
3		付费出站票人数		付费出站票成功刷卡后累加
4		免费出站票人数		免费出站票成功刷卡后累加
5		拒绝人数		刷卡报无效票时累加
6		拒绝进闸人数		进闸刷卡报无效票时累加
7		拒绝出闸人数		出闸刷卡报无效票时累加
8		车票回收数量		回收一张车票时累加
9		出闸扣费金额		成功出闸刷卡后该票卡类型扣费金额和总扣费金额累加
10		付费出站票扣款金额		付费出站票成功刷卡后累加
11		进闸允许通过人数		闸机收到读写器发出的允许入闸通行信号，维护模式除外

表 68 AGM 寄存器 (续)

序号	寄存器 ID	名称	值标识	说明
12		出闸允许通过人数		闸机收到读写器发出的允许出闸通行信号, 维护模式除外
13		正常通过人数		正常验票后正常通过人数 (通行逻辑控制器返回)
14		非法通过人数		尾随通过及其他能检测到的非法通过
15		超时关门次数		正常验票后无人通过, 超时关门次数

## 4.2.4.4.2 TVM 寄存器数据

表 69 TVM 寄存器

序号	寄存器 ID	名称	值标识	说明
1		车票出售数量		成功出售一张单程票后该面额数量和总数量累加
2		充值次数		成功充值一张票卡后该票卡充值次数和总充值次数累加
3		车票发售的票值		成功出售一张单程票后该面额发售额累加
4		TVM 累计接收硬币的总数		成功接收一枚硬币后该面额数量和总数量累加
5		TVM 累计接收纸币的总数		成功接收一张纸币后该面额数量和总数量累加
6		找零口找出硬币的总数		成功找出硬币后累加
7		进入废票箱的车票数量		车票进入废票箱后累加, 维护模式除外
8		累计找零纸币总数		成功找零一张纸币后该面额数量和总数量累加
9		纸币找零进废币箱累计数		纸币找零进入废币箱一张纸币后该面额数量和总数量累加

## 4.2.4.4.3 AVM 寄存器数据

表 70 AVM 寄存器

序号	寄存器 ID	名称	值标识	说明
1		充值次数		成功充值一张票卡后该票卡充值次数和总充值次数累加
2		充值金额		成功充值一张票卡后该票卡充值和总充值金额累加
3		AVM 累计接收纸币的总数		成功接收一张纸币后该面额数量和总数量累加

## 4.2.4.4.4 BOM 寄存器数据

表 71 BOM 寄存器

序号	寄存器 ID	名称	值标识	说明
1		单程票发售数量		成功出售一张单程票后该面额数量和总数量累加
2		充值次数		成功充值一张票卡后该票卡充值次数和总充值次数累加
3		非单程票发售数量		成功出售一张非单程票后该票卡类型数量累加
4		超乘更新数量		成功超乘更新一张车票后该票卡类型数量累加
5		超时更新数量		成功超时更新一张车票后该票卡类型数量累加
6		无进站更新数量		成功无进站更新一张车票后该票卡类型数量累加
7		无出站更新数量		成功无出站更新一张车票后该票卡类型数量累加
8		车票退款数量		成功退款一张车票后该票卡类型数量累加
9		发售付费出站票数量		成功发售付费出站票后数量累加
10		发售免费出站票数量		成功发售免费出站票后数量累加
11		老人免费票激活数量		成功激活一张老人免费票后数量累加
12		单程票发售金额		成功出售一张单程票后该面额金额和总金额累加

表 71 BOM 寄存器 (续)

序号	寄存器 ID	名称	值标识	说明
13		充值金额		成功充值一张票卡后该票卡充值和总充值金额累加
14		非单程票发售金额		成功出售一张非单程票后该票卡类型金额累加
15		超乘更新金额		成功超乘更新一张车票后该票卡类型金额累加
16		超时更新金额		成功超时更新一张车票后该票卡类型金额累加
17		无进站码更新金额		成功无进站更新一张车票后该票卡类型金额累加
18		无出站更新金额		成功无出站更新一张车票后该票卡类型金额累加
19		车票退款金额		成功退款一张车票后该票卡类型金额累加
20		发售付费出站票金额		成功发售付费出站票后金额累加

## 4.2.4.4.5 初始化机寄存器数据

表 72 初始化机寄存器

序号	寄存器 ID	名称	值标识	说明
1		初始化不同面额票卡次数		成功初始化一张票卡后该值累加
2		SAM 卡额度消耗		成功充值一张票卡后 SAM 卡额度减少相应金额
3		进入废票箱的车票数量		车票初始化时进入废票箱后累加, 维护模式除外

## 4.2.4.4.6 核准机寄存器数据

表 73 核准机寄存器

序号	寄存器 ID	名称	值标识	说明
1		核准机读卡器读写次数		成功核准一张票卡后该值累加

4.2.4.4.7 换票机寄存器数据

表 74 换票机寄存器

序号	寄存器 ID	名称	值标识	说明
1		车票出售数量		成功出售一张票后该值数量累加
2		接收车票数量		成功接收一张票卡后该次数累加
3		车票发售的票值		成功出售一张票后该面额发售额累加
4		累计接收硬币的总数		成功接收一枚硬币后该面额数量和总数量累加
5		累计接收纸币的总数		成功接收一张纸币后该面额数量和总数量累加
6		找零口找出硬币的总数		成功找出硬币后累加
7		进入废票箱的车票数量		车票进入废票箱后累加，维护模式除外
8		累计找零纸币总数		成功找零一张纸币后该面额数量和总数量累加
9		纸币找零进废币箱累计数		纸币找零进入废币箱一张纸币后该面额数量和总数量累加

4.2.4.4.8 交易汇总数据

表 75 交易汇总数据

序号	字段	描述
1	票卡类型	
2	票卡子类型	
3	交易类型	
4	交易子类型	
5	交易合计次数	
6	交易合计金额	



## 4.2.4.5 运营控制数据

表 76 运营控制数据列表

命令类别	命令名称	命令代码	内容格式	说明	备注
模式控制	运营模式控制	017	模式编号：字符 模式内容：不定长	控制系统处于某种模式： 00 正常模式 03 紧急模式 05 离线模式 06 站厅关闭服务模式 07 维护模式 08 关闭服务 09 故障模式 10 列车故障模式 11 进、出站次序免检模式 12 乘车时间免检 13 车票日期免检 14 车费免检	
	运营模式控制反馈	018			
控制命令	暂停	009	无		
	暂停反馈	010	执行结果	0：成功 9：失败	
	开启（暂停后恢复正常）	011	无		
	开启（暂停后恢复正常）反馈	012	执行结果	0：成功 9：失败	
	重启	015	无		
	重启反馈	016	应答结果	0：执行	由设备决定当前状态是否允许重启，如果允许，则在做完应答后重启
	关机	049	无		

表 76 运营控制数据列表（续 1）

命令类别	命令名称	命令代码	内容格式	说明	备注
	关机反馈	050	应答结果	0: 执行	由设备决定当前状态是否允许关机, 如果允许, 则在做完应答后关机。
	停运车站	019	停运类型: 解除所有售票限制, 此时车站编码填全 0; 01: 允许售票; 02: 拒绝售票。 车站编号: 此处为计费的站点编号		
	停运车站反馈	020	执行结果	0: 执行成功 9: 执行失败	
	时钟同步请求	021	无		下级向上级请求时钟
	时钟同步请求反馈	022	时钟信息		
	时钟同步	023	时钟信息		上级强制同步下级时钟
	时钟同步反馈	024	执行结果	0: 同步成功 9: 同步失败	
	版本同步	027	参数种类 版本类型 版本编号	参数版本的按如下次序格式编码 0: 历史版本 1: 当前版本 2: 将来版本 版本编号不足 30 位前补空格	上级下达同步命令
	版本同步反馈	028	执行结果	0: 同步成功 9: 同步失败	

表 76 运营控制数据列表（续 2）

命令类别	命令名称	命令代码	内容格式	说明	备注
	设备工作方式控制	041	工作方式	根据设备种类不同而不同 闸机工作方式：0X TVM 工作方式：1X AVM 工作方式：2X BOM 工作方式：3X 闸机运作方式： 00：正常（默认） 01：只处理储值票 02：处理所有车票 TVM 工作方式： 10：正常（默认） 11：只收纸币 12：只收硬币 13：纸币硬币全收 BOM 工作方式： 20：正常（默认） 21：售票 22：补票 23：售补兼顾 AVM 工作方式： 30：正常（默认） 31：只收纸币 32：只收银行卡 33：纸币银行卡全收	
	设备工作方式控制反馈	042	执行结果	0：成功 9：失败	
	设备部件控制	043	部件类型 控制方式		例如机芯 开关控制
	设备部件控制反馈	044	执行结果	0：成功 9：失败	

表 76 运营控制数据列表（续 3）

命令类别	命令名称	命令代码	内容格式	说明	备注
	设备运营方式控制	045		根据设备种类不同而不同 闸机运营方式：0X TVM 运营方式：1X AVM 运营方式：2X BOM 运营方式：3X 闸机业务方式： 00：进站 01：出站 02：双向 TVM 运营方式： 10：正常（既可售票也可充值） 11：只售票 12：只充值	
	设备运营方式控制反馈	046	执行结果	0：成功 9：失败	
	闸机常开/闭控制	047		00：常开 01：常闭	
	闸机常开/闭控制反馈	048	执行结果	0：成功 9：失败	
查询命令	参数版本查询	051	参数种类 查询模式	查询模式（版本类型）： 0：历史版本 1：当前版本 2：将来版本 9：全部版本	

表 76 运营控制数据列表（续 4）

命令类别	命令名称	命令代码	内容格式	说明	备注
	参数版本查询反馈	052	参数种类 版本类型 版本编号	格式采用定长 参数版本的类型按如下次序排列 0: 历史版本 1: 当前版本 2: 将来版本 版本编号不足 30 位前补空格	如果回复中只有参数种类, 则表明种类参数不存在或查询失败。如果回复中只有参数种类+ 版本类型, 则表明种类参数的版本类型不存在或查询失败。
	软件版本查询	053	应用编号	无应用编号表示查询本设备所有应用版本	
	软件版本查询反馈	054	应用编号 反馈信息	反馈信息格式: “/”+应用编号（不定长） + “:” + 【版本编号】	如果回复只有应用编号则表明应用类型不存在或查询失败
	寄存器查询	055	寄存器		
	寄存器查询反馈	056	寄存器 ID 寄存器值		

4.2.4.6 状态数据

表 77 状态数据列表

序号	名称	数据类型	长度	备注
1	终端编号	CHAR	9	一般为 9 位设备编码
2	终端类型	CHAR	4	一般为 4 位设备类型编码
3	当前时间	DATE		当前时间 YYYYMMDD HH24MISS
4	终端状态	CHAR	2	00 正常 01 告警运行 02 操作间休（BOM 专用） 09 故障暂停 10 维护 90 暂停（关闭服务/手工暂停） 98 离线
5	模式状态	CHAR	2	00 正常模式 XX 其它运营模式编号
6	时钟误差类型	CHAR	1	+: 当前时钟快于时钟服务器 -: 当前时钟慢于时钟服务器
7	当前时钟误差	CHAR	5	时钟误差数（秒）；大于 99999 填写 99999
8	事件类型	CHAR	2	按照需要上报的事件通知进行分类
9	事件内容	CHAR	4000	包含设备基本故障信息

4.2.4.7 事务数据

4.2.4.7.1 AGM 更换票箱记录

表 78 AGM 更换票箱记录

序号	数据项	类型	长度	描述
1	设备编号	CHAR	9	
2	部件电子 ID	CHAR	16	
3	部件类型	CHAR	1	
4	操作类型	CHAR	2	
5	时间	DATE	14	
6	操作员	CHAR	6	
7	事务内容	CHAR	50	
8	票卡类型	CHAR	4	

表 78 AGM 更换票箱记录 (续)

序号	数据项	类型	长度	描述
9	票卡有效状态	CHAR	1	1: 有效票 0: 废票
10	票卡数量	CHAR	5	不足左补 0

## 4.2.4.7.2 AGM 票箱切换记录

表 79 AGM 票箱切换记录

序号	数据项	类型	长度	描述
1	设备编号	CHAR	9	
2	部件电子 ID	CHAR	16	
3	部件类型	CHAR	1	
4	操作类型	CHAR	2	
5	时间	DATE	14	
6	操作员	CHAR	6	
7	事务内容	CHAR	50	
8	票卡类型	CHAR	4	
9	票卡有效状态	CHAR	1	1: 有效票 0: 废票
10	票卡数量	CHAR	5	不足左补 0

## 4.2.4.7.3 TVM 补票记录

表 80 TVM 补票记录

序号	数据项	类型	长度	描述
1	设备编号	CHAR	9	
2	部件电子 ID	CHAR	16	
3	部件类型	CHAR	1	
4	操作类型	CHAR	2	
5	时间	DATE	14	
6	操作员	CHAR	6	
7	事务内容	CHAR	50	
8	票卡类型	CHAR	4	
9	票卡有效状态	CHAR	1	1: 有效票 0: 废票
10	票卡数量	CHAR	5	不足左补 0

4.2.4.7.4 更换钱箱记录

表 81 更换钱箱记录

序号	数据项	类型	长度	描述
1	设备编号	CHAR	9	
2	部件电子 ID	CHAR	16	
3	部件类型	CHAR	1	
4	操作类型	CHAR	2	
5	时间	DATE	14	
6	操作员	CHAR	6	
7	事务类型	CHAR	2	
8	现金种类	CHAR	3	
9	现金数量	CHAR	5	
10	现金金额	CHAR	5	

4.2.4.7.5 TVM 补充找零硬币记录

表 82 TVM 补充找零硬币记录

序号	数据项	类型	长度	描述
1	设备编号	CHAR	9	
2	部件电子 ID	CHAR	16	
3	部件类型	CHAR	1	
4	操作类型	CHAR	2	
5	时间	DATE	14	
6	操作员	CHAR	6	
7	事务类型	CHAR	2	
8	现金种类	CHAR	3	
9	现金数量	CHAR	5	
10	现金金额	CHAR	5	

4.2.4.8 票务数据

4.2.4.8.1 车票管理

4.2.4.8.1.1 车票配发通知



表 83 车票配发通知

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	20	
2	车站编号	CHAR	6	
3	配发时间	DATE	14	
4	预计配发时间	DATE	14	
5	操作员号	CHAR	6	
6	运营商配发申请单据编号	CHAR	8	
7	车票种类数	CHAR	3	
8	车票类型	CHAR	2	
9	子卡类型	CHAR	2	
10	有效终止日期	CHAR	8	
11	车票状态	CHAR	1	
12	车票面额	CHAR	8	
13	车票属性	CHAR	2	
14	车票数量	CHAR	8	
15	备注	CHAR	50	

## 4.2.4.8.1.2 车票回收通知

表 84 车票回收通知

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	20	
2	车站编号	CHAR	6	
3	通知日期时间	DATE	14	
4	要求上交时间	DATE	14	
5	操作员号	CHAR	6	
7	车票种类数	CHAR	3	
8	车票类型	CHAR	2	
9	子卡类型	CHAR	2	
10	有效终止日期	CHAR	8	
11	车票状态	CHAR	1	
12	车票面额	CHAR	8	
13	车票属性	CHAR	2	
14	车票数量	CHAR	8	
15	备注	CHAR	50	

## 4.2.4.8.1.3 上交车票处理

表 85 上交车票通知

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	20	
2	车站编号	CHAR	6	
3	日期时间	DATE	14	
5	操作员号	CHAR	6	
6	通知单据编号	CHAR	20	
7	车票种类数	CHAR	3	
8	车票类型	CHAR	2	
9	子卡类型	CHAR	2	
10	有效终止日期	CHAR	8	
11	车票状态	CHAR	1	
12	车票面额	CHAR	8	
13	车票属性	CHAR	2	
14	车票数量	CHAR	8	
15	备注	CHAR	50	

## 4.2.4.8.1.4 车票调拨通知

表 86 车票调拨通知

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	20	生成规则: ZPDB+yyyy+5 位流水 例: ZJDB201100002
2	调出车站编号	CHAR	6	
3	调入车站编号	CHAR	6	
4	日期时间	DATE	14	YYYYMMDDHHMMSS 默认值: 当前系统时间
5	操作员号	CHAR	6	
6	通知单据编号	CHAR	20	
7	车票种类数	CHAR	3	车票的种类数 N
8	车票类型	CHAR	2	
9	子卡类型	CHAR	2	
10	有效终止日期	CHAR	8	YYYYMMDD 无限制填 19000101
11	车票状态	CHAR	1	0: 正常卡
12	车票面额	CHAR	8	以分、次、天为单位;
13	车票属性	CHAR	2	颜色、版面等
14	车票数量	CHAR	8	
15	备注	CHAR	50	

## 4.2.4.8.2 收益数据

## 4.2.4.8.2.1 短款补款通知

表 87 短款补款通知

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	14	
2	核对员工号	CHAR	6	
3	差错员工	CHAR	6	
4	补款截止日期	CHAR	8	
5	短款金额	CHAR	8	
6	差错时间	DATE	14	
7	差错原因	CHAR	50	
8	差错说明	CHAR	50	
9	备注	CHAR	50	

## 4.2.4.8.2.2 上交补款处理

表 88 上交补款单据

序号	名称	数据类型	长度	备注
1	单据编号	CHAR	14	
2	核对员工号	CHAR	6	
3	差错员工	CHAR	6	
4	补款截止日期	CHAR	8	
5	短款金额	CHAR	8	
6	差错时间	DATE	14	
7	差错原因	CHAR	50	
8	差错说明	CHAR	50	
9	备注	CHAR	50	

## 4.2.4.8.2.3 上报车站售存日报

表 89 上报车站售存日报

序号	名称	数据类型	长度	备注
1	车站编号	CHAR	10	
2	报表编号	CHAR	20	
3	报表日期	DATE	14	
4	起始日期	DATE	14	
5	终止日期	DATE	14	

表 89 上报车站售存日表（续）

序号	名称	数据类型	长度	备注
6	车票类型	CHAR	10	
7	车票状态	CHAR	1	
8	昨日库存	NUMBER(10)	10	
9	本日中心配票数	NUMBER(10)	10	
10	本日回收	NUMBER(10)	10	
11	本日交回数据	NUMBER(10)	10	
12	其它收入 1	NUMBER(10)	10	
13	其它收入 2	NUMBER(10)	10	
14	本日上缴金额	NUMBER(10)	10	
15	售票员配票数	NUMBER(10)	10	
16	TVM 配票数	NUMBER(10)	10	
17	其它支出 1	NUMBER(10)	10	
18	其它支出 2	NUMBER(10)	10	
19	本日短款	NUMBER(10)	10	
20	本日库存	NUMBER(10)	10	
21	客值	CHAR	20	
22	调入数量	NUMBER(10)	10	
23	调出数量	NUMBER(10)	10	
24	处理标志	CHAR	1	

## 4.2.4.8.2.4 上报车站营收日报

表 90 上报车站营收日报

序号	名称	数据类型	长度	备注
1	车站编号	CHAR	10	
2	报表日期	DATE	14	
3	报表编号	CHAR	20	
4	开始时间	DATE	14	
5	结束时间	DATE	14	
6	客值	CHAR	20	
7	昨日剩余金额	NUMBER(18, 2)	18	
8	昨日解行	NUMBER(18, 2)	18	
9	今日剩余金额	NUMBER(18, 2)	18	
10	今日解行	NUMBER(18, 2)	18	
11	昨日实际解行	NUMBER(18, 2)	18	
12	今日实际解行	NUMBER(18, 2)	18	
13	TVM 收入	NUMBER(18, 2)	18	

表 90 上报车站营收日报（续）

序号	名称	数据类型	长度	备注
14	AVM 收入	NUMBER (18, 2)	18	
15	BOM 收入	NUMBER (18, 2)	18	
16	异常金额	NUMBER (18, 2)	18	
17	短款补款	NUMBER (18, 2)	18	
18	卡押金	NUMBER (18, 2)	18	
19	预制票	NUMBER (18, 2)	18	
20		NUMBER (18, 2)	18	
21		NUMBER (18, 2)	18	
22	应收金额	NUMBER (18, 2)	18	
23	实收金额	NUMBER (18, 2)	18	
24	差异金额	NUMBER (18, 2)	18	
25	核对人编号	CHAR	20	
26	处理标志	CHAR	1	

## 4.2.4.8.2.5 上报售票员结算单

表 91 上报售票员结算单

序号	名称	数据类型	长度	备注
1	车站编号	CHAR	10	
2	售票员编号	CHAR	20	
3	结算单编号	CHAR	20	
4	结算日期	DATE	14	
5	起始时间	DATE	14	
6	终止时间	DATE	14	
7	备用金配备总额	NUMBER (18, 2)	18	
8	备用金剩余总额	NUMBER (18, 2)	18	
9	退款总金额	NUMBER (18, 2)	18	
10	充值金额	NUMBER (18, 2)	18	
11	超程超时补值	NUMBER (18, 2)	18	
12	进出更新现金	NUMBER (18, 2)	18	
13	进出更新卡扣	NUMBER (18, 2)	18	
14	行政费用	NUMBER (18, 2)	18	
15	退卡	NUMBER (18, 2)	18	
16	记名卡申请	NUMBER (18, 2)	18	
17	预收款	NUMBER (18, 2)	18	
18	预收款人	CHAR	20	
19	预收款人姓名	CHAR	20	
20	应收金额	NUMBER (18, 2)	18	

表 91 上报售票员结算单（续）

序号	名称	数据类型	长度	备注
21	实收金额	NUMBER (18, 2)	18	
22	上交金额	NUMBER (18, 2)	18	

4.2.4.8.2.6 上报设备票款差异

表 92 上报设备票款差异

序号	数据名称	数据类型	长度	备注
1	车站编号	CHAR	10	
2	报表编号	CHAR	20	
3	报表日期	DATE	14	
4	起始日期	DATE	14	
5	终止日期	DATE	14	
6	TVM 审核标志	CHAR	1	
7	TVM 审核人	CHAR	10	
8	TVM 审核日期	DATE	14	
9	AVM 审核标志	CHAR	1	
10	AVM 审核人	CHAR	10	
11	AVM 审核日期	DATE	14	
12	GATE 审核标志	CHAR	1	
13	GATE 审核人	CHAR	10	
14	GATE 审核日期	DATE	14	
15	TVM 复核人	CHAR	10	

## 4.2.4.9 任务数据

## 4.2.4.9.1 生产任务文件

表 93 生产任务单据

序号	数据名称	数据类型	长度	备注
生产任务信息				
1	任务单号	CHAR	10	任务单号
2	任务类型	CHAR	2	任务类型 00: 票卡初始化任务 01: 预赋值任务 02: 抵消任务 03: 逻辑销毁 05: 退卡注销任务
3	数量	INT	10	任务计划制票数量, 单位: 张
4	票卡类型	CHAR	2	计划制票的票卡类型
5	生成人员	CHAR	6	任务计划生产人员编号
6	终端编号	CHAR	9	任务计划终端编号
7	制单人员	CHAR	6	任务制定人员编号
8	格式版本	CHAR	2	格式版本
9	任务描述	CHAR	>=256	任务描述信息
10	卡批次号	CHAR	9	卡应用号批次号
11	制卡数量	INT	10	与上述任务关联卡应用号数量
12	关联方式	CHAR	1	与上述任务关联方式 0: 部分关联 1: 完全关联
13	卡应用号	CHAR	9	
14	卡物理号	CHAR	16	
乘客批次信息				
15	乘客批次号	CHAR	9	乘客批次号
16	乘客信息数量	INT	10	与上述任务关联的乘客信息数量
17	关联方式	CHAR	1	与上述任务关联方式 0: 部分关联 1: 完全关联
乘客明细信息				
18	乘客姓名	CHAR	>=16	乘客姓名 (最大 8 位中文)
19	乘客性别	CHAR	1	乘客性别, 0 表示男, 1 表示女
20	证件类型编号	CHAR	2	乘客证件类型编号, 范围: 00~99
21	证件号码	CHAR	<=20	乘客证件号码

表 93 生产任务单据（续）

序号	数据名称	数据类型	长度	备注
22	乘客地址	CHAR	<=30	乘客的地址
23	卡批次号	CHAR	9	卡应用号批次号
24	制卡数量	INT	10	与上述任务关联卡应用号数量
25	关联方式	CHAR	1	与上述任务关联方式 0：部分关联 1：完全关联

## 4.2.4.9.2 生产任务反馈文件

表 94 生产任务反馈单据

序号	数据名称	数据类型	长度	备注
任务汇总信息				
1	任务单号	CHAR	10	任务单号
2	任务类型	CHAR	2	00：票卡初始化任务 01：预赋值任务 02：抵消任务 03：注销任务
3	数量	INT	10	任务总数 单位：张
4	好票数量	INT	10	任务已完成的好票累计数量, 单位：张
5	坏票数量	INT	10	任务已完成坏票数量, 单位：张
6	任务状态	CHAR	2	任务状态 2：安排 3：执行 4：结束
7	生成人员	CHAR	6	任务反馈的生产人员编号
8	终端编号	CHAR	9	任务反馈的终端编号
9	格式版本	CHAR	2	格式版本
票卡完成信息				
10	制卡状态	CHAR	1	制卡状态 0：制卡成功 1：制卡失败 2：未制
11	任务批次号	CHAR	9	所属的此任务批次号
12	卡应用号	CHAR	9	卡应用号, 可选



表 94 生产任务反馈单据（续）

序号	数据名称	数据类型	长度	备注
13	卡物理号	CHAR	16	卡物理号, 可选
14	通信命令号	CHAR	3	交易数据的通信命令号
15	制卡交易数据	CHAR	不定长, <512	制卡交易数据, 按照下述命令号进行组织, 可选
乘客明细信息				
16	制卡状态	CHAR	1	制卡状态 0: 制卡成功 1: 制卡失败 2: 未制
17	记名批次	CHAR	9	所属乘客记名批次
18	乘客姓名	CHAR	>=16	乘客姓名 (最大 8 位中文), 可选
19	乘客性别	CHAR	1	乘客性别, 0 表示男, 1 表示女, 可选
20	证件类型	CHAR	2	乘客证件类型编号, 范围: 00~99, 可选
21	证件号码	CHAR	>=20	乘客证件号码, 可选
22	卡应用号	CHAR	9	对应的卡应用号, 可选

## 4.2.4.9.3 任务属性值

表 95 任务属性值列表

序号	名称	数据类型	长度	备注
1	激活状态	CHAR	1	
2	卡应用版本	CHAR	2	
3	行业代码	CHAR	4	
4	卡应用标识	CHAR	1	
5	发卡方式	CHAR	1	0: 空白卡发卡 1: 回收卡发卡
6	卡密钥版本	CHAR	2	
7	票卡主类型	CHAR	2	00: 一票通单程票 01: 一票通计值票 02: 一票通计期票 03: 一票通计次票 99: 城市一卡通
8	卡版本	CHAR	2	
9	费用	INT	8	
10	建设部标识	CHAR	4	

表 95 任务属性值列表（续）

序号	名称	数据类型	长度	备注
11	国家标识	CHAR	4	
12	押金	INT	8	
13	卡有效结束日期	DATE	14	yyyyMMddmmhhss
14	抵消是否判断票卡有效期	CHAR	1	0:否 1:是
15	发行批次	CHAR	9	
16	发行日期	DATE	14	yyyyMMddmmhhss
17	发行商代码	CHAR	4	
18	城市代码	CHAR	4	
19	发行区锁定标志	CHAR	1	
20	钱包标识	CHAR	1	
21	是否需要激活标志	CHAR	1	0: 不需要激活 1: 需要激活
22	是否读写个人信息标志			0:否; 1:是
23	卡有效开始日期	DATE	14	yyyyMMddmmhhss
24	可使用站点 1	CHAR	4	
25	可使用站点 2	CHAR	4	
26	可用区段	CHAR	4	
27	卡状态	CHAR	1	0: 未售 3: 已售 7: 回收 8: 已挂失 9: 黑名单
28	终止日期	DATE	14	yyyyMMddmmhhss
29	卡子类型	CHAR	2	
30	售票金额	INT	8	
31	抵消允许结束日期	DATE	14	yyyyMMddmmhhss
32	抵消开始结束日期	DATE	14	yyyyMMddmmhhss
33	卡金额	INT	8	
34	启用标识	CHAR	1	

## 4.2.4.10 认证数据

## 4.2.4.10.1 权限数据

## 4.2.4.10.1.1 联网登录

表 96 联网登录认证

序号	名称	数据类型	长度	备注
1	用户编号	CHAR	6	
2	密码	CHAR	6	
3	终端编号	CHAR	9	一般为 9 位设备编码
4	登录时间	CHAR	14	当前时间 YYYYMMDD HH24MISS
5	功能点编号	CHAR	4	一般为 4 位设备类型编码
6	功能点名称	CHAR	20	00 正常
7	功能点类型	CHAR	2	
8	功能点父节点	CHAR	2	
9	功能点序列号	CHAR	2	

## 4.2.4.10.1.2 修改密码

表 97 密码修改认证

序号	名称	数据类型	长度	备注
1	用户编号	CHAR	6	
2	旧密码	CHAR	6	(加密后的密文)
2	新密码	CHAR	6	(加密后的密文)
3	终端编号	CHAR	9	一般为 9 位设备编码
4	操作时间	CHAR	14	当前时间 YYYYMMDD HH24MISS

## 4.2.4.10.1.3 获取账户分组信息

表 98 获取账户分组信息

序号	名称	数据类型	长度	备注
1	终端编号	CHAR	9	一般为 9 位设备编码
2	操作时间	CHAR		当前时间 YYYYMMDD HH24MISS
3	用户编号	CHAR	6	
4	用户组	CHAR	6	
5	用户姓名	CHAR	20	
6	密码	CHAR	6	(加密后的密文)
7	用户卡号	CHAR	9	
8	用户卡物理号	CHAR	16	

表 98 获取账户分组信息（续）

序号	名称	数据类型	长度	备注
9	用户卡有效截止日期（预留）	CHAR	14	用户卡有效截止日期（预留）
10	用户属性（预留）	CHAR	2	用户属性（预留）
11	组 ID	CHAR	2	
12	组名称	CHAR	20	

4.2.4.10.1.4 获取设备组权限

表 99 获取设备组权限

序号	名称	数据类型	长度	备注
1	终端编号	CHAR	9	一般为 9 位设备编码
2	操作时间	CHAR		当前时间 YYYYMMDD HH24MISS
3	种类功能点	CHAR	6	
4	用户组功能点	CHAR	6	
5	功能点 ID	CHAR	20	
6	功能点名称	CHAR	6	
7	功能点类型	CHAR	9	
9	用户组 ID	CHAR	14	
10	功能点权限	CHAR		功能点权限 0：无权限 1：有权限

4.2.4.10.2 安全认证数据

表 100 安全认证数据

序号	名称	数据类型	长度	备注
1	系统编号	CHAR	9	
2	设备编号	CHAR	9	
3	SAM 卡编号	CHAR	20	
4	SAM 物理编号	CHAR	16	
5	SAM 卡版本	NUMBER(2)	2	
7	应用版本	NUMBER(2)	2	
8	SAM 卡类型	CHAR	3	
9	最大值	NUMBER(10)	10	

表 100 安全认证数据（续）

序号	名称	数据类型	长度	备注
10	最小值	NUMBER(10)	10	
11	适用值	NUMBER(10)	10	
12	有效起始日期	DATE	14	
13	有效终止日期	DATE	14	
14	SAM 卡状态	CHAR DEFAULT '0'	1	
15	在线状态	CHAR(1) DEFAULT '0'	1	
16	认证时间	DATE	14	
17	操作员	CHAR	6	
18	键索引	NUMBER	1	
19	有效天数	NUMBER	10	
20	更新时间	DATE	14	
21	最近申请时间	DATE	14	
22	是否同步	CHAR	1	
23	SAM 卡主类型	CHAR	1	
24	密钥卡 ID	CHAR	16	
25	密钥卡类型	CHAR	2	
26	SAM 卡序号	NUMBER(10)	10	
27	最后一次申请时间	DATE	14	
28	申请序列号	CHAR	20	
29	当前值	NUMBER(10)	10	
30	查询时间	DATE	14	
31	提前申请天数	NUMBER(10)	10	
32	当前时间	DATE	14	
33	控制 SAM 卡 ID	CHAR	16	

表 101 设备认证记录

序号	名称	数据类型	长度	备注
1	SAM 卡编号	CHAR	20	
2	交易时间	DATE	14	
3	接收时间	DATE	14	
4	剩余金额	NUMBER(10)	10	
5	申请金额	NUMBER(10)	10	
7	申请状态	CHAR DEFAULT '0'	1	

表 101 设备认证记录（续）

序号	名称	数据类型	长度	备注
8	原因	CHAR	10	
9	接收金额	NUMBER(10)	10	
10	有效日期	DATE	14	
11	更新状态	CHAR DEFAULT '9'	1	
12	申请前额度	NUMBER(10)	10	
13	申请后额度	NUMBER(10)	10	
14	旧文件名	CHAR	50	
15	文件名	CHAR	100	
16	申请类型	CHAR DEFAULT '0'	1	
17	设备编号	CHAR	9	
18	服务类型	CHAR	2	
19	加密机	CHAR	20	
20	旧授权序列号	CHAR	20	
21	授权序列号	CHAR	20	

4.2.4.11 日志数据

4.2.4.11.1 日志级别

LOG\_LEVEL\_DEBUG 调试信息

LOG\_LEVEL\_INFO 普通日志信息

LOG\_LEVEL\_WARNING 告警信息

LOG\_LEVEL\_ERROR 错误信息

4.2.4.11.2 日志格式

日志文件的命名格式为：

Transaction\_设备编号\_yyyymmddhh24+序号.log

日志文件的容量以及记录日志文件的数量可通过系统下发设备参数进行控制。

日志文件统一采用 ASCII 码记录。不同列用 {TAB} 分隔符来分隔。

## 4.2.4.11.3 操作日志

表 102 操作日志

序号	数据名称	数据类型	数据描述	备注
1	操作时间	Date		精确到毫秒
2	操作对象	CHAR		设备编号/系统编号
3	操作员代号	CHAR		
4	操作员名称	CHAR		
5	操作类型	CHAR		
6	操作内容	CHAR		不同设备、系统采用不同的内容详细格式
7	操作结果	CHAR		根据不同的设备、系统反馈不同的结果信息，如：一条AVM充值事务，反馈充值是否成功。
8	日志等级	CHAR		日志等级，包括：调试信息、普通日志、告警信息、错误信息等不同级别。

## 4.2.4.11.4 调试日志

表 103 调试日志

序号	数据名称	数据类型	数据描述	备注
1	操作时间	DATE		精确到毫秒
2	操作对象	CHAR		设备编号/系统编号
3	线程编号	CHAR		
4	进程编号	CHAR		
5	进程名称	CHAR		
6	方法路径	CHAR		
7	操作员代号	CHAR		
8	操作员名称	CHAR		
9	操作类型	CHAR		
10	操作内容	CHAR		不同设备、系统采用不同的内容详细格式
11	日志等级	CHAR		日志等级，包括：调试信息、普通日志、告警信息、错误信息等不同级别。

## 5 通信接口技术规范

### 5.1 概述

通信服务程序可以分布在 AFC 系统的各级计算机上, 包括: 中心 (CLC)、车站计算机 (SC)、车站终端设备 (SLE)。CLC、SC、SLE 之间的数据传输均通过服务器进行。

本部分定义 AFC 系统内各层级服务器 (CLC、SC、SLE) 之间的在线数据 (buffer/file) 传输交换及通信控制规定。依据通信数据格式标准和交互协议完成数据的传输和处理。作为通信节点的通信服务程序向其他节点上的通信服务程序发送数据, 同时接收其他节点传来的数据。

每个节点上的通信服务程序主动向配置的直接上级节点通信服务程序请求建立连接, 同时被动接受下级节点的通信服务程序连接请求。

### 5.2 通信服务程序功能摘要

#### 5.2.1 平台要求

- a) 通信服务程序须支持大型机、小型机、各大服务器厂商的各系统服务器、各类刀片机、工控机、PC 机等目前主流的硬件平台。
- b) 通信服务程序须支持 IBM AIX、Windows NT、Windows 2000、Windows XP、Windows2003、Windows Vista、Sun Solaris systems、HP-UX systems、Linux platform 等操作系统平台。
- c) 通信服务程序须支持标准 TCP/IP 协议, 进行消息和文件的可靠传输。

#### 5.2.2 基本功能要求

- a) 通信服务程序须提供可靠的数据传输能力, 保证数据的完整性、一致性, 保证传输的数据不丢失、不重复。
- b) 通信服务程序须支持异常时的消息重发和系统重启恢复的消息重发。
- c) 通信服务程序须支持集群, 并可进行动态扩展。
- d) 通信服务程序须支持路由转发功能。
- e) 通信服务程序须有死信队列。



- f) 通信服务程序须有完备的日志和集中配置、监控界面。
- g) 通信服务程序须支持应用负载均衡。
- h) 通信服务程序须支持触发机制。
- i) 通信服务程序须提供符合地铁要求的 API 接口，至少支持 C，C++，C#，Java，VB 等程序设计语言，并支持基于 JMS 平台的开发。
- j) 通信服务程序须支持点对点和订阅发布两种网络传输模式。
- k) 通信服务程序之间的网络连接支持长连接和短连接等。
- l) 通信服务程序须支持事务机制。

### 5.2.3 消息传输要求

- a) 须至少支持文本数据，多媒体数据，二进制数据等类型的消息传输。
- b) 须支持的消息长度最少为 4M。
- c) 须支持持久化存储消息，可根据需要设定是否进行持久化存储。
- d) 须支持分包传输消息功能、支持断点续传。
- e) 消息须支持优先级，分为 10 个优先级：9 为独占，0 为最低，1~8 为普通优先级，通信服务程序在处理不同优先级的消息时，按照优先级进行权重。
- f) 须可设置消息的生命周期。
- g) 须支持消息的加密，至少支持 SHA-2，MD5，DES，3DES，AES 等加密算法。支持 SSL 在传输层对网络连接进行加密。
- h) 须支持消息的无损压缩。
- i) 须支持消息的删除。

### 5.2.4 文件传输要求

- a) 须至少支持文本数据，多媒体数据，二进制数据等类型的文件传输。
- b) 须支持的文件大小最少为 1000M。
- c) 须支持文件的断点续传。
- d) 须可设置文件消息的生命周期。
- e) 须支持文件的加密，至少支持 SHA-2，MD5，DES，3DES，AES 等加密算法。支持 SSL 在传输层对网络连接进行加密。

- f) 须支持文件的无损压缩。
- g) 须支持文件消息的删除，同时删除文件。

### 5.2.5 性能要求

- a) 通信服务程序服务端须支持最少 1000 个客户端的同时连接。
- b) 通信服务程序的吞吐量须满足表 5-2- 1 的要求。

表 104 吞吐量要求

系统类型	收发类型	数据类型	消息种类	带宽 (Mbps)	吞吐量 (笔/秒)	备注
设备	本地	消息	非持久	--	$\geq 100$	消息大小为 2KB
			持久	--	$\geq 60$	消息大小为 2KB
	点对点	消息	非持久	100	$\geq 100$	消息大小为 2KB
			持久	100	$\geq 60$	消息大小为 2KB
			非持久	2	$\geq 50$	消息大小为 2KB
			持久	2	$\geq 30$	消息大小为 2KB
		文件	--	100	$\geq 2$	文件大小为 2MB
			--	100	$\geq 0.05$	文件大小为 100MB
系统	本地	消息	非持久	--	$\geq 5000$	消息大小为 2KB
			持久	--	$\geq 3000$	消息大小为 2KB
	点对点	消息	非持久	100	$\geq 4000$	消息大小为 2KB
			持久	100	$\geq 2500$	消息大小为 2KB
			非持久	2	$\geq 80$	消息大小为 2KB
			持久	2	$\geq 50$	消息大小为 2KB

表 104 吞吐量要求（续）

系统类型	收发类型	数据类型	消息种类	带宽（Mbps）	吞吐量（笔/秒）	备注
		文件	--	100	>=5	文件大小为2MB
			--	100	>=0. 1	文件大小为100MB
			--	2	>=0. 1	文件大小为2MB

注 1：设备配置参见《系统和设备技术规范》

注 2：车站系统服务器和中心通信服务器配置参见《系统和设备技术规范》通信接口

5.3 通信接口

5.3.1 建立连接

函数原型：int SZMC\_TX\_Conn(char\* szMgrName, SZMCHandle pHandle, SZMC\_ERR\_INFO& errInfo);

功能描述：应用实例和服务节点建立连接、申请节点资源。

参数说明：

szMgrName：队列管理器名称。如果没有，则设置为 NULL。

pHandle：通信句柄。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

5.3.2 断开连接

函数原型：int SZMC\_TX\_Disconn(SZMCHandle pHandle, SZMC\_ERR\_INFO& errInfo);

功能描述：应用实例和服务节点断开连接、释放节点资源。

参数说明：

pHandle：通信句柄。

errInfo：错误描述。

返回值:

0: 成功

非 0: 失败

### 5.3.3 注册

函数原型: `int SZMC_TX_Login(SZMCHandle pHandle, char* szQueueName, int iOptions, SZMC_ERR_INFO& errInfo);`

功能描述: 应用实例注册、申请队列资源。

参数说明:

**pHandle:** 通信句柄。

**szQueueName:** 队列名称。

**iOptions:** 打开队列的参数。

**errInfo:** 错误描述。

返回值:

0: 成功

非 0: 失败

### 5.3.4 注销

函数原型: `int SZMC_TX_Logout(SZMCHandle pHandle, SZMC_ERR_INFO& errInfo);`

功能描述: 应用实例注销、释放队列资源。

参数说明:

**pHandle:** 通信句柄。

**errInfo:** 错误描述。

返回值:

0: 成功

非 0: 失败

### 5.3.5 线路检测

函数原型: `int SZMC_TX_Testline(SZMCHandle pHandle, char *szDest, SZMC_ERR_INFO& errInfo);`

功能描述: 检测指定的线路是否正常。

参数说明:

pHandle: 通信句柄。

szDest: 线路名称。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

### 5.3.6 获取线路名称

函数原型: int SZMC\_TX\_GetLineName(SZMCHandle pHandle, char \*szLineName, SZMC\_ERR\_INFO& errInfo);

功能描述: 获取本线路的名称。

参数说明:

pHandle: 通信句柄。

szLineName: 本线路名称。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

### 5.3.7 线路活动性检测

函数原型: int SZMC\_TX\_ChkLine(SZMCHandle pHandle, SZMC\_ERR\_INFO& errInfo);

功能说明: 检测线路活动性。

参数说明:

pHandle: 通信句柄。

errInfo: 错误描述。

返回值:

0: 活动

非 0: 不活动

### 5.3.8 消息发送

函数原型：int SZMC\_TX\_PutMsg(SZMCHandle pHandle, SZMCMsg\_Info \*pUsrMsg, char \*szMsgContent,int iMsgLen, SZMC\_ERR\_INFO& errInfo);

功能说明：向指定节点的指定队列发送消息（普通消息或文件消息）。

参数说明：

pHandle：通信句柄。

pUsrMsg：消息参数。

szMsgContent：消息体（对于文件消息，则为文件名）。

iMsgLen：消息长度。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

### 5.3.9 消息接收

函数原型：int SZMC\_TX\_GetMsg(SZMCHandle pHandle, SZMCMsg\_Info \*pUsrMsg, char \*szMsgContent,int& iMsgLen, SZMC\_ERR\_INFO& errInfo);

功能说明：从指定节点的指定队列接收消息。

参数说明：

pHandle：通信句柄。

pUsrMsg：消息参数。

lpszMsgContent：消息体（对于文件消息，则为文件名）。

iMsgLen：消息长度。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

## 5.4 管理接口

#### 5.4.1 启动队列管理器

函数原型： `int SZMC_TX_StartQueueManager(LPSTR lpszManagerName, SZMC_ERR_INFO& errInfo);`

功能说明：启动指定名称的队列管理器。

参数说明：

`lpszManagerName`：队列管理器名称。

`errInfo`：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.2 停止队列管理器

函数原型： `int SZMC_TX_StopQueueManager(LPSTR lpszManagerName, SZMC_ERR_INFO& errInfo);`

功能说明：停止指定名称的队列管理器。

参数说明：

`lpszManagerName`：队列管理器名称。

`errInfo`：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.3 新增队列管理器

函数原型： `int SZMC_TX_AddQueueManager(LPSTR lpszManagerName, SZMC_Manager_Config* pConfig, SZMC_ERR_INFO& errInfo);`

功能说明：新增队列管理器。

参数说明：

`lpszManagerName`：队列管理器名称。

`pConfig`：队列管理器配置参数。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.4 删除队列管理器

函数原型: `int SZMC_TX_DeleteQueueManager(LPSTR lpszManagerName, SZMC_ERR_INFO& errInfo);`

功能说明: 删除队列管理器。

参数说明:

lpszManagerName: 队列管理器名称。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.5 查询队列管理器状态

函数原型: `int SZMC_TX_QueryQueueManager(LPSTR lpszManagerName, SZMC_ERR_INFO& errInfo);`

功能说明: 查询队列管理器的当前状态。

参数说明:

lpszManagerName: 队列管理器名称。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.6 新增发送队列

函数原型: `int SZMC_TX_AddSendQueue(LPSTR lpszSendQueueName, SZMC_Send_Queue_Config* pConfig, SZMC_ERR_INFO& errInfo);`



功能说明：新增发送队列。

参数说明：

lpszSendQueueName：发送队列名。

pConfig：发送队列配置参数。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.7 删除发送队列

函数原型：int SZMC\_TX\_DeleteSendQueue(LPSTR lpszSendQueueName, SZMC\_ERR\_INFO& errInfo);

功能说明：删除发送队列。

参数说明：

lpszSendQueueName：发送队列名。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.8 获取发送队列信息

函数原型：int SZMC\_TX\_GetSendQueue(LPSTR lpszSendQueueName, SZMC\_Send\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取发送队列信息。

参数说明：

lpszSendQueueName：发送队列名。

pInfo：发送队列信息。

errInfo：错误描述。

返回值：

0: 成功

非 0: 失败

#### 5.4.9 清空发送队列消息

函 数 原 型 : int SZMC\_TX\_ClearSendQueueMessage(LPSTR lpszSendQueueName, SZMC\_ERR\_INFO& errInfo);

功能说明: 清空发送队列消息。

参数说明:

lpszSendQueueName: 发送队列名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.10 删除发送队列指定消息

函数原型: int SZMC\_TX\_DeleteSendQueueMessage(LPSTR lpszSendQueueName, SZMC\_Send\_Queue\_Message\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 删除发送队列指定消息。

参数说明:

lpszSendQueueName: 发送队列名。

pInfo: 发送队列消息信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.11 获取发送队列指定消息

函数原型: int SZMC\_TX\_GetSendQueueMessage((LPSTR lpszSendQueueName, SZMC\_Send\_Queue\_Get\_Msg\_Opt\* pInfo, SZMC\_Send\_Queue\_Msg\_Info\* pMsgInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取发送队列指定消息。

参数说明：

lpzSendQueueName：发送队列名。

pInfo：获取消息参数。

pMsgInfo：获取的消息内容。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.12 新增远程队列

函数原型：int SZMC\_TX\_AddRemoteQueue(LPSTR lpzRemoteQueueName, SZMC\_Remote\_Queue\_Config\* pConfig, SZMC\_ERR\_INFO& errInfo);

功能说明：新增远程队列。

参数说明：

lpzRemoteQueueName：远程队列名。

pConfig：远程队列配置参数。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.13 删除远程队列

函数原型：int SZMC\_TX\_DeleteRemoteQueue(LPSTR lpzRemoteQueueName, SZMC\_ERR\_INFO& errInfo);

功能说明：删除远程队列。

参数说明：

lpzRemoteQueueName：远程队列名。

errInfo：错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.14 获取远程队列信息

函数原型: int SZMC\_TX\_GetRemoteQueue(LPSTR lpszRemoteQueueName,  
SZMC\_Remote\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 获取远程队列信息。

参数说明:

lpszRemoteQueueName: 远程队列名。

pInfo: 获取的远程队列信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.15 新增本地队列

函数原型: int SZMC\_TX\_AddLocalQueue(LPSTR lpszLocalQueueName,  
SZMC\_Remote\_Queue\_Config\* pConfig, SZMC\_ERR\_INFO& errInfo);

功能说明: 新增本地队列。

参数说明:

lpszLocalQueueName: 本地队列名

pConfig: 本地队列配置参数。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.16 删除本地队列

函数原型: int SZMC\_TX\_DeleteLocalQueue(LPSTR lpszLocalQueueName, SZMC\_ERR\_INFO&

errInfo);

功能说明：删除本地队列。

参数说明：

lpzLocalQueueName：本地队列名。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.17 获取本地队列信息

函数原型：int SZMC\_TX\_GetLocalQueue(LPSTR lpzLocalQueueName, SZMC\_Local\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取本地队列信息。

参数说明：

lpzLocalQueueName：本地队列名。

pInfo：本地队列信息。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.18 修改本地队列信息

函数原型：int SZMC\_TX\_SetLocalQueue(LPSTR lpzLocalQueueName, SZMC\_Local\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：修改本地队列信息。

参数说明：

lpzLocalQueueName：本地队列名。

pInfo：本地队列信息。

errInfo：错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.19 清空本地队列消息

函数原型: int SZMC\_TX\_ClearLocalQueueMessage(LPSTR lpszLocalQueueName,  
SZMC\_ERR\_INFO& errInfo);

功能说明: 清空指定队列消息。

参数说明:

lpszLocalQueueName: 本地队列名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.20 删除本地队列指定消息

函数原型: int SZMC\_TX\_DeleteLocalQueueMessage(LPSTR lpszLocalQueueName,  
SZMC\_Local\_Queue\_Message\_Info\* pMsgInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 删除本地队列指定消息。

参数说明:

lpszLocalQueueName: 本地队列名。

pMsgInfo: 本地队列消息信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.21 获取本地队列指定消息信息

函数原型: int SZMC\_TX\_GetLocalQueueMessage(LPSTR lpszLocalQueueName,  
SZMC\_Local\_Queue\_Message\_Opt\* pInfo, SZMC\_Local\_Queue\_Message\_Info\*

pMsgInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取本地队列指定消息信息。

参数说明：

lpzLocalQueueName:

pInfo: 本地队列消息配置信息。

pMsgInfo: 本地队列消息信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.22 新增集群队列

函数原型: int SZMC\_TX\_AddClusterQueue(LPSTR lpzClusterQueueName,  
SZMC\_Cluster\_Queue\_Config\* pConfig, SZMC\_ERR\_INFO& errInfo);

功能说明：新增集群队列。

参数说明：

lpzClusterQueueName: 集群队列名。

pConfig: 集群队列配置参数。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.23 删除集群队列

函数原型: int SZMC\_TX\_DeleteClusterQueue(LPSTR lpzClusterQueueName, SZMC\_ERR\_INFO&  
errInfo);

功能说明：删除集群队列。

参数说明：

lpzClusterQueueName: 集群队列名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.24 获取集群队列信息

函数原型: int SZMC\_TX\_GetClusterQueue(LPSTR lpszClusterQueueName,  
SZMC\_Cluster\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 获取集群队列信息。

参数说明:

lpszClusterQueueName: 集群队列名。

pInfo: 集群队列信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.25 修改集群队列

函数原型: int SZMC\_TX\_SetClusterQueue(LPSTR lpszClusterQueueName,  
SZMC\_Cluster\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 修改集群队列。

参数说明:

lpszClusterQueueName: 集群队列名。

pInfo: 集群队列信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.26 获取集群队列状态



函数原型: `int SZMC_TX_GetClusterQueueStatus(LPSTR lpszClusterQueueName, SZMC_Cluster_Queue_Status* pStatus, SZMC_ERR_INFO& errInfo);`

功能说明: 获取指定的集群队列状态。

参数说明:

`lpszClusterQueueName`: 集群队列名。

`pStatus`: 集群队列状态。

`errInfo`: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.27 新增集群目的队列

函数原型: `int SZMC_TX_AddClusterDestQueue(LPSTR lpszClusterDestQueueName, SZMC_Cluster_Dest_Queue_Config* pConfig, SZMC_ERR_INFO& errInfo);`

功能说明: 新增集群目的队列。

参数说明:

`lpszClusterDestQueueName`: 集群目的队列名。

`pConfig`: 集群目的队列配置参数。

`errInfo`: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.28 删除集群目的队列

函数原型: `int SZMC_TX_DeleteClusterDestQueue(LPSTR lpszClusterDestQueueName, SZMC_ERR_INFO& errInfo);`

功能说明: 删除集群目的队列。

参数说明:

`lpszClusterDestQueueName`: 集群目的队列名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.29 获取集群目的队列信息

函数原型: int SZMC\_TX\_GetClusterDestQueue(LPSTR lpszClusterDestQueueName,  
SZMC\_Cluster\_Dest\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 获取集群目的队列信息。

参数说明:

lpszClusterDestQueueName: 集群目的队列名。

pInfo: 集群目的队列信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.30 修改集群目的队列

函数原型: int SZMC\_TX\_SetClusterDestQueue(LPSTR lpszClusterDestQueueName,  
SZMC\_Cluster\_Dest\_Queue\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 修改集群目的队列信息。

参数说明:

lpszClusterDestQueueName: 集群目的队列名。

pInfo: 集群目的队列信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.31 新增发布订阅

函数原型: int SZMC\_TX\_AddPubSub(LPSTR lpszPubSubName, SZMC\_Pub\_Sub\_Config\* pConfig, SZMC\_ERR\_INFO& errInfo);

功能说明: 新增发布订阅。

参数说明:

lpszPubSubName: 发布订阅名。

pConfig: 发布订阅配置参数。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.32 删除发布订阅

函数原型: int SZMC\_TX\_DeletePubSub(LPSTR lpszPubSubName, SZMC\_ERR\_INFO& errInfo);

功能说明: 删除发布订阅。

参数说明:

lpszPubSubName: 发布订阅名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.33 获取发布订阅信息

函数原型: int SZMC\_TX\_GetPubSub(LPSTR lpszPubSubName, SZMC\_Pub\_Sub\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 获取发布订阅信息。

参数说明:

lpszPubSubName: 发布订阅名。

pInfo: 发布订阅信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.34 修改发布订阅

函数原型: int SZMC\_TX\_SetPubSub(LPSTR lpszPubSubName, SZMC\_Pub\_Sub\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明: 修改发布订阅。

参数说明:

lpszPubSubName: 发布订阅名。

pInfo: 发布订阅信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.35 获取指定的发布订阅状态

函数原型: int SZMC\_TX\_GetPubSubStatus(LPSTR lpszPubSubName, SZMC\_Pub\_Sub\_Status\* pStatus, SZMC\_ERR\_INFO& errInfo);

功能说明: 获取指定的发布订阅状态。

参数说明:

lpszPubSubName: 发布订阅名。

pStatus: 发布订阅状态。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.36 获取指定的主题订阅信息

函数原型: int SZMC\_TX\_GetTopicAndSubscriber(LPSTR lpszPubSubName, SZMC\_Pub\_Sub\_Info\*

pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取指定的主题订阅信息。

参数说明：

lpszPubSubName：发布订阅名。

pInfo：主题订阅信息。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.37 根据主题名获取指定的订阅者信息

函数原型：int SZMC\_TX\_GetSubscriberbyTopic(LPSTR lpszTopicName, SZMC\_Sub\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：根据主题名获取指定的订阅者信息。

参数说明：

lpszTopicName：主题名

pInfo：订阅者信息。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.38 根据订阅者获取指定的主题名信息

函数原型：int SZMC\_TX\_GetTopicbySubscriber (LPSTR lpszSubName, SZMC\_Topic\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：根据订阅者获取指定的主题名信息。

参数说明：

lpszSubName：订阅者名。

pInfo：主题名信息。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.39 新增主题

函数原型: int SZMC\_TX\_AddTopic(LPSTR lpszTopicName, SZMC\_Topic\_Config\* pConfig, SZMC\_ERR\_INFO& errInfo);

功能说明: 新增主题。

参数说明:

lpszTopicName: 主题名。

pConfig: 主题配置参数。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.40 删除主题

函数原型: int SZMC\_TX\_DeleteTopic(LPSTR lpszTopicName, SZMC\_ERR\_INFO& errInfo);

功能说明: 删除主题。

参数说明:

lpszTopicName: 主题名。

errInfo: 错误描述。

返回值:

0: 成功

非 0: 失败

#### 5.4.41 获取主题信息

函数原型: int SZMC\_TX\_GetTopic(LPSTR lpszTopicName, SZMC\_Topic\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：获取主题信息。

参数说明：

lpzTopicName：主题名。

pInfo：主题信息。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

#### 5.4.42 修改主题信息

函数原型：int SZMC\_TX\_SetTopic(LPSTR lpzTopicName, SZMC\_Topic\_Info\* pInfo, SZMC\_ERR\_INFO& errInfo);

功能说明：修改主题信息。

参数说明：

lpzTopicName：主题名。

pInfo：主题信息。

errInfo：错误描述。

返回值：

0：成功

非 0：失败

## 6 设备部件接口函数规范

### 6.1 说明

本文档统一以 C 语言描述 API 函数。

### 6.2 工控机接口

6.2.1 基本函数接口

6.2.1.1 打开看门狗

6.2.1.1.1 函数原型

INT32 Szmc\_IPC\_Open\_Dog(SZMCHANDLE& hCom, UINT32 uiResetTime)

6.2.1.1.2 函数功能描述

建立与看门狗的通讯及初始化看门狗。

6.2.1.1.3 输入参数说明

hCom: 看门狗句柄。

uiResetTime: 看门狗复位时间。

6.2.1.1.4 输出参数说明

无。

6.2.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.2 关闭看门狗

6.2.1.2.1 函数原型

INT32 Szmc\_IPC\_Close\_Dog(SZMCHANDLE hCom)

6.2.1.2.2 函数功能描述

断开与看门狗的通讯。

6.2.1.2.3 输入参数说明

hCom: 看门狗句柄。



6.2.1.2.4 输出参数说明

无。

6.2.1.2.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.3 获取操作错误信息

6.2.1.3.1 函数原型

VOID Szmc\_IPC\_GetErrorInfo(SZMC\_IPC\_ERR\_INFO& info)

6.2.1.3.2 函数功能描述

获取错误信息的具体内容。

6.2.1.3.3 输入参数说明

无。

6.2.1.3.4 输出参数说明

info:

```
typedef _szmc_ipc_err_info
{
    INT32 iErrorNo;
    LPSTR lpszPhyErrorNo;
    LPSTR lpszErrInfo;
} SZMC_IPC_ERR_INFO;
```

iErrorNo: 逻辑错误码。

lpszPhyErrorNo: 物理错误码。

lpszErrInfo: 错误描述。

6.2.1.3.5 函数返回值说明

无。

6.2.1.4 复位看门狗

6.2.1.4.1 函数原型

INT32 Szmc\_IPC\_Reset\_Dog(SZMCHANDLE hCom)

6.2.1.4.2 函数功能描述

复位看门狗。

6.2.1.4.3 输入参数说明

hCom: 看门狗句柄。

6.2.1.4.4 输出参数说明

无。

6.2.1.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.5 开启 IO 控制端口

6.2.1.5.1 函数原型

INT32 Szmc\_IPC\_Open\_IO(SZMCHANDLE& hCom,UINT32 uiInput,UINT32 uiOutput)

6.2.1.5.2 函数功能描述

开启工控机 IO 端口, 并进行设置。

6.2.1.5.3 输入参数说明

hCom: IO 句柄。

**uiInput:** IO 输入方向设置。**uiInput** 每一位对应一位输入，如果某一位设置为 1，则该位对应的输入改为输出，否则该位为输入。

**uiOutput:** IO 输出方向设置。**uiOutput** 每一位对应一位输出，如果某一位设置为 1，则该位对应的输入改为输入，否则该位为输出。

6.2.1.5.4 输出参数说明

无。

6.2.1.5.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.6 关闭 IO 控制端口

6.2.1.6.1 函数原型

INT32 Szmc\_IPC\_Close\_IO(SZMCHANDLE hCom)

6.2.1.6.2 函数功能描述

关闭工控机 IO。

6.2.1.6.3 输入参数说明

**hCom:** IO 句柄。

6.2.1.6.4 输出参数说明

无。

6.2.1.6.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.7 获取工控机 ID

6.2.1.7.1 函数原型

INT32 Szmc\_IPC\_GETID(LPSTR lpszID)

6.2.1.7.2 函数功能描述

获取工控机 ID。

6.2.1.7.3 输入参数说明

无。

6.2.1.7.4 输出参数说明

lpszID: 工控机 ID。

6.2.1.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.8 设置寄存器

6.2.1.8.1 函数原型

INT32 Szmc\_IPC\_SetReg(UINT32 uiRegID,LPSTR lpszRegValue)

6.2.1.8.2 函数功能描述

设置寄存器值。

6.2.1.8.3 输入参数说明

uiRegID: 寄存器 ID。

lpszRegValue: 寄存器值。

6.2.1.8.4 输出参数说明

无。

6.2.1.8.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.9 获取寄存器值

6.2.1.9.1 函数原型

INT32 Szmc\_IPC\_GetReg(UINT32 uiRegID,LPSTR lpszRegValue)

6.2.1.9.2 函数功能描述

获取寄存器值。

6.2.1.9.3 输入参数说明

uiRegID: 寄存器 ID。

6.2.1.9.4 输出参数说明

lpszRegValue: 寄存器值。

6.2.1.9.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.10 获取 CPU 信息

6.2.1.10.1 函数原型

INT32 Szmc\_IPC\_GetCpuInfo(SZMC\_IPC\_CPU\_INFO& info)

6.2.1.10.2 函数功能描述

获取工控机 cpu 信息。

6.2.1.10.3 输入参数说明

无。

6.2.1.10.4 输出参数说明

SZMC\_IPC\_CPU\_INFO& info:

typedef struct \_szmc\_ipc\_cpu\_info

{

int iFrequency;

int iUsage;

} SZMC\_IPC\_CPU\_INFO,\*LPSZMC\_IPC\_CPU\_INFO;

iFrequency: CPU 主频, 单位 MHz。

iUsage: CPU 使用率, 单位 1%。

6.2.1.10.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.11 获取内存信息

6.2.1.11.1 函数原型

INT32 Szmc\_IPC\_GetMemInfo(SZMC\_IPC\_MEM\_INFO& info)

6.2.1.11.2 函数功能描述

获取工控机内存信息。

6.2.1.11.3 输入参数说明

无。

6.2.1.11.4 输出参数说明

SZMC\_IPC\_MEM\_INFO& info:

```
typedef struct _szmc_ipc_mem_info
{
    UINT32 uiSize;

    UINT32 uiUsage;
} SZMC_IPC_MEM_INFO,*LPSZMC_IPC_MEM_INFO;
```

uiSize: 内存大小, 单位为 MB。

uiUsage: 物理内存使用率, 单位 1%。

6.2.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.1.12 获取硬盘信息

6.2.1.12.1 函数原型

```
INT32 Szmc_IPC_GetDiskInfo(SZMC_IPC_DISK_INFO& info)
```

6.2.1.12.2 函数功能描述

获取工控机硬盘信息。

6.2.1.12.3 输入参数说明

无。

6.2.1.12.4 输出参数说明

SZMC\_IPC\_DISK\_INFO& info:

```
typedef struct _szmc_ipc_disk_info
{
    UINT32 uiSize;
```

```
    UINT32 uiUsage;

    UINT32 uiPartitionCount;

    LPSZMC_IPC_PARTITION_INFO lpList;

} SZMC_IPC_DISK_INFO,*LPSZMC_IPC_DISK_INFO;
```

iSize: 物理硬盘大小, 单位为 MB。

iUsage: 物理硬盘使用率, 单位 1%。

uiPartitionCount: 分区数。

LPSZMC\_IPC\_PARTITION\_INFO lpList:

```
typedef struct _szmc_ipc_partition_info
{
    UINT32 uiSize;

    UINT32 uiUsage;

    LPSTR lpszName;

}SZMC_IPC_PARTITION_INFO,*LPSZMC_IPC_PARTITION_INFO;
```

uiSize: 分区容量, 单位 MB。

uiUsage: 分区使用率, 单位 1%。

lpszName: 分区名称。

#### 6.2.1.12.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

#### 6.2.1.13 获取其他信息

##### 6.2.1.13.1 函数原型

```
INT32 Szmc_IPC_GetOtherInfo(UINT32 uiInfoID,SZMC_IPC_OTHER_INFO& info)
```

##### 6.2.1.13.2 函数功能描述

获取工控机除上述函数接口中未定义的其他信息。



6.2.1.13.3 输入参数说明

uiInfoID: 其他信息 ID 号。

6.2.1.13.4 输出参数说明

SZMC\_IPC\_OTHER\_INFO& info:  
typedef struct \_szmc\_ipc\_other\_info  
{  
LPSTR lpszInfo;  
} SZMC\_IPC\_OTHER\_INFO,\*LPSZMC\_IPC\_OTHER\_INFO;  
lpszInfo: 获取的信息内容。

6.2.1.13.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.2.2 其他函数接口

无

6.3 单程票发售机构

6.3.1 基本函数接口

6.3.1.1 打开模块

6.3.1.1.1 函数原型

INT32 Szmc\_STM\_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)

6.3.1.1.2 函数功能描述

建立与模块之间的通信并进行初始化。

6.3.1.1.3 输入参数说明

hCom: 模块句柄。

lpzLogicalName: 部件预定义逻辑名称。

6.3.1.1.4 输出参数说明

无。

6.3.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.2 关闭模块

6.3.1.2.1 函数原型

INT32 Szmc\_STM\_Close (SZMCHANDLE hCom)

6.3.1.2.2 函数功能描述

断开与模块的通讯。

6.3.1.2.3 输入参数说明

hCom: 模块句柄。

6.3.1.2.4 输出参数说明

无。

6.3.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_ERR_STM_PARA	存在非法参数
SZMC_ERR_STM_EXECUTE	执行命令失败

## 6.3.1.3 获取操作错误信息

## 6.3.1.3.1 函数原型

```
VOID Szmc_STM_GetErrorInfo(SZMCHANDLE hCom, SZMC_STM_ERR_INFO& info)
```

## 6.3.1.3.2 函数功能描述

获取错误信息的具体内容。

## 6.3.1.3.3 输入参数说明

hCom: 模块句柄。

## 6.3.1.3.4 输出参数说明

info:

```
typedef _szmc_stm_err_info
```

```
{
```

```
    INT32 iErrorNo;
```

```
    LPSTR lpszPhyErrorNo;
```

```
    LPSTR lpszErrInfo;
```

```
} SZMC_STM_ERR_INFO;
```

iErrorNo: 逻辑错误码。

lpszPhyErrorNo: 物理错误码。

lpszErrInfo: 错误描述。

## 6.3.1.3.5 函数返回值说明

无。

## 6.3.1.4 获取模块信息

## 6.3.1.4.1 函数原型

```
INT32 Szmc_STM_GetInfo(SZMCHANDLE hCom, SZMC_STM_INFO& info)
```

6.3.1.4.2 函数功能描述

获取模块相关信息。

6.3.1.4.3 输入参数说明

hCom: 模块句柄。

6.3.1.4.4 输出参数说明

SZMC\_STM\_INFO info:

typedef struct \_szmc\_stm\_info

{

LPSTR lpszSTMVer;

LPSTR lpszDriverVer;

LPSTR lpszManufacturerVer;

LPSTR lpszSTMID;

UINT32 uiDeviceType;

LPSTR lpszDeviceName;

UINT32 uiStatus;

} SZMC\_STM\_INFO,\*LPSZMC\_STM\_INFO;

lpszSTMVer: 模块版本。

lpszDriverVer: 模块驱动版本。

lpszManufacturerVer: 制造商版本。

lpszSTMID: 模块 ID。

iDeviceType: 模块类型。

lpszDeviceName: 模块类型名。

uiStatus: 模块状态。uiStatus 为下列值之一:

值	含义
SZMC_STATUS_STM_IDLE	模块空闲中
SZMC_STATUS_STM_BUSY	模块工作中
SZMC_STATUS_STM_OFFLINE	模块未连接
SZMC_STATUS_STM_COMERR	模块无法连接
SZMC_STATUS_STM_HWERR	模块硬件故障

SZMC_STATUS_STM_NOCOMP	无此模块或者未配置该模块
------------------------	--------------

6.3.1.4.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.5 获取模块状态

6.3.1.5.1 函数原型

INT32 Szmc\_STM\_GetStatus(SZMCHANDLE hCom,SZMC\_STM\_STATUS& status)

6.3.1.5.2 函数功能描述

获取模块状态。

6.3.1.5.3 输入参数说明

hCom：模块句柄。

SZMC\_STM\_STATUS status:

```
typedef _szmc_stm_status
{
    UINT32 uiDevice;
    UINT32 uiDevPosition;
    LPSTR lpszExtra;
} SZMC_STM_STATUS,*LPSZMC_STM_STATUS;
```

uiDevice：部件状态。uiDevice 为下列值之一：

值	含义
SZMC_STATUS_STM_IDLE	模块空闲中
SZMC_STATUS_STM_BUSY	模块工作中
SZMC_STATUS_STM_OFFLINE	模块未连接
SZMC_STATUS_STM_COMERR	模块无法连接
SZMC_STATUS_STM_HWERR	模块硬件故障
SZMC_STATUS_STM_NOCOMP	无此模块或者未配置该模块

uiDevPosition：设备的位置。iDevPosition 为下列值之一：

值	含义
SZMC_STM_INPOSITION	部件在其正常的运行位置
SZMC_STM_NOTINPOSITION	部件已经从其正常的运行位置移走
SZMC_STM_POSUNKNOWN	部件的位置无法确定
SZMC_STM_POSNOTSUPPLY	该部件不具备探测位置的能力

lpszExtra: 指向厂商特有或任何其它扩展信息的一个列表。返回信息为一系列“键=值”字符串以便 SP 进行扩展。每一个字符串都是以空字符 NULL 结束的，最后一个字符串以两个空字符 NULL 结束。一个空列表可通过一个空指针或两个连续的空字符 NULL 来表示。

#### 6.3.1.5.4 输出参数说明

无。

#### 6.3.1.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

#### 6.3.1.6 重启模块

##### 6.3.1.6.1 函数原型

INT32 SzmC\_STM\_Reset(SZMCHANDLE hCom)

##### 6.3.1.6.2 函数功能描述

模块软复位。

##### 6.3.1.6.3 输入参数说明

hCom: 模块句柄。

##### 6.3.1.6.4 输出参数说明

无。

##### 6.3.1.6.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.7 设置参数

6.3.1.7.1 函数原型

```
INT32 Szmc_STM_SetParam(SZMCHANDLE hCom, SZMC_STM_PARAM_INFO info)
```

6.3.1.7.2 函数功能描述

设置模块参数。

6.3.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_STM\_PARAM\_INFO info:

```
typedef struct _szmc_stm_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_STM_PARAM_INFO,*LPSZMC_STM_PARAM_INFO;
```

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.3.1.7.4 输出参数说明

无。

6.3.1.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

#### 6.3.1.7.6 其他说明

应包括部件日志的参数设置。

#### 6.3.1.8 获取参数

##### 6.3.1.8.1 函数原型

```
INT32 Szmc_STM_GetParam(SZMCHANDLE hCom, SZMC_STM_PARAM_INFO& info)
```

##### 6.3.1.8.2 函数功能描述

获取模块参数。

##### 6.3.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_STM\_PARAM\_INFO info:

```
typedef struct _szmc_stm_param_info
```

```
{
```

```
    LPSTR lpszParamName;
```

```
    LPSTR lpszParamValue;
```

```
} SZMC_STM_PARAM_INFO,*LPSZMC_STM_PARAM_INFO;
```

lpszParamName: 参数名。

##### 6.3.1.8.4 输出参数说明

SZMC\_STM\_PARAM\_INFO info:

```
typedef struct _szmc_stm_param_info
```

```
{
```

```
    LPSTR lpszParamName;
```

```
    LPSTR lpszParamValue;
```

```
} SZMC_STM_PARAM_INFO,*LPSZMC_STM_PARAM_INFO;
```

lpszParamValue: 参数值。



6.3.1.8.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.9 设置模式

6.3.1.9.1 函数原型

INT32 Szmc\_STM\_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)

6.3.1.9.2 函数功能描述

设置各种模式。

6.3.1.9.3 输入参数说明

hCom：模块句柄。

uiModeID：模式 ID。

6.3.1.9.4 输出参数说明

无。

6.3.1.9.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.10 获取模式

6.3.1.10.1 函数原型

INT32 Szmc\_STM\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.3.1.10.2 函数功能描述

获取当前模式。

6.3.1.10.3 输入参数说明

hCom: 模块句柄。

6.3.1.10.4 输出参数说明

uiModeID: 模式 ID

6.3.1.10.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.11 设置寄存器

6.3.1.11.1 函数原型

INT32 SzmC\_STM\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.3.1.11.2 函数功能描述

设置各种寄存器。

6.3.1.11.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

lpszRegValue: 寄存器值。

6.3.1.11.4 输出参数说明

无。

6.3.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.12 获取寄存器值

6.3.1.12.1 函数原型

```
INT32 SzmC_STM_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpszRegValue)
```

6.3.1.12.2 函数功能描述

获取各种寄存器值。

6.3.1.12.3 输入参数说明

- hCom: 模块句柄。
- uiRegID: 寄存器 ID。

6.3.1.12.4 输出参数说明

lpszRegValue: 寄存器值。

6.3.1.12.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.13 自检控制命令

6.3.1.13.1 函数原型

```
INT32 SzmC_STM_SelfTest(SZMCHANDLE hCom,UINT32 uiSelfTestType)
```

6.3.1.13.2 函数功能描述

自检。

6.3.1.13.3 输入参数说明

- hCom: 模块句柄。
- uiSelfTestType: 自检类型。uiSelfTestType 为下列值之一:

值	含义
---	----

SZMC_STM_SIMPLE_SELFTEST	简单自检
SZMC_STM_HIGH_SELFTEST	高级自检

6.3.1.13.4 输出参数说明

无。

6.3.1.13.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.1.14 软件更新

6.3.1.14.1 函数原型

INT32 Szmc\_STM\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.3.1.14.2 函数功能描述

软件更新。

6.3.1.14.3 输入参数说明

hCom：模块句柄。

lpszPara：软件更新时使用的参数，比如更新路径及文件名等。

6.3.1.14.4 输出参数说明

无。

6.3.1.14.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2 其他函数接口

## 6.3.2.1 获取票箱信息

## 6.3.2.1.1 函数原型

```
INT32 SzmC_STM_GetBoxInfo(SZMCHANDLE hCom, UINT32 uiBoxNo ,SZMC_STM_BOXINFO&
stBoxInfo)
```

## 6.3.2.1.2 函数功能描述

获取票箱信息。

## 6.3.2.1.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 票箱号。当 uiBoxNo 为 0 时，则查询所有票箱信息。

## 6.3.2.1.4 输出参数说明

SZMC\_STM\_BOXINFO stBoxInfo:

```
typedef struct _szmc_stm_box_info
```

```
{
```

```
    UINT32 uiCount;
```

```
    LPSZMC_STM_BOXUNIT lpList;
```

```
} SZMC_STM_BOXINFO,*LPSZMC_STM_BOXINFO;
```

uiCount: 要查询的票箱的返回数。

LPSZMC\_STM\_BOXUNIT lpList:

```
typedef struct _szmc_stm_box_unit
```

```
{
```

```
    UINT32 uiBoxNo;
```

```
    UINT32 uiTicketType;
```

```
    LPSTR lpszBoxID;
```

```
    UINT32 uiBoxFullNum;
```

```
    UINT32 uiBoxWillFullNum;
```

```
    UINT32 uiBoxEmptyNum;
```

```
    UINT32 uiBoxWillEmptyNum;  
  
    UINT32 uiTicketNum;  
  
    UINT32 uiStatus;  
  
} SZMC_STM_BOXUNIT,*LPSZMC_STM_BOXUNIT;
```

uiBoxNo: 票箱号。

uiTicketType: 可售单程票类型。iTicketType 为下列值之一:

值	含义
SZMC_TICKET_COMMON	普通单程票
SZMC_TICKET_CHILDREN	儿童单程票
SZMC_TICKET_VIP	VIP 单程票
SZMC_TICKET_OTHER	其他单程票

lpszBoxID: 票箱 ID 号。

uiBoxFullNum: 票箱满数量。

uiBoxWillFullNum: 票箱将满数量。

uiBoxEmptyNum: 票箱空数量。

uiBoxWillEmptyNum: 票箱将空数量。

uiTicketNum: 现有数量。

uiStatus: 票箱当前状态。iStatus 为下列值之一:

值	含义
SZMC_STM_BOX_STAT_OK	票箱正常
SZMC_STM_BOX_STAT_FULL	票箱满
SZMC_STM_BOX_STAT_WILLFULL	票箱将满
SZMC_STM_BOX_STAT_EMPTY	票箱空
SZMC_STM_BOX_STAT_WILL_EMPT	票箱将空
Y	
SZMC_STM_BOX_STAT_NO_EXIST	该票箱不存在
SZMC_STM_BOX_STAT_MISSING	该票箱不在合法的位置上
SZMC_STM_BOX_STAT_UNABLE	该票箱设定为不可用

#### 6.3.2.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.2 设置票箱数量

6.3.2.2.1 函数原型

```
INT32 SzmC_STM_SetBoxNum(SZMCHANDLE hCom,UINT32 uiBoxNo,UINT32 uiBoxNum)
```

6.3.2.3 函数功能描述

设置票箱数量。

6.3.2.3.1 输入参数说明

- hCom: 模块句柄。
- uiBoxNo: 票箱编号。
- uiBoxNum: 票箱数量。

6.3.2.3.2 输出参数说明

无。

6.3.2.3.3 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.4 出票到读写区

6.3.2.4.1 函数原型

```
INT32 SzmC_STM_TicketToReader(SZMCHANDLE hCom)
```

6.3.2.4.2 函数功能描述

出票到读写区。

6.3.2.4.3 输入参数说明

- hCom: 模块句柄。

6.3.2.4.4 输出参数说明

无。

6.3.2.4.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.5 出票到出票口

6.3.2.5.1 函数原型

INT32 SzmC\_STM\_TicketOut(SZMCHANDLE hCom)

6.3.2.5.2 函数功能描述

出票到出票口。

6.3.2.5.3 输入参数说明

hCom：模块句柄。

6.3.2.5.4 输出参数说明

无。

6.3.2.5.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.6 回收票到回收箱

6.3.2.6.1 函数原型

INT32 SzmC\_STM\_Collect(SZMCHANDLE hCom)



6.3.2.6.2 函数功能描述

回收票。

6.3.2.6.3 输入参数说明

hCom: 模块句柄。

6.3.2.6.4 输出参数说明

无。

6.3.2.6.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.7 回收票到废票箱

6.3.2.7.1 函数原型

INT32 Szmc\_STM\_TicketToWastebin(SZMCHANDLE hCom)

6.3.2.7.2 函数功能描述

回收票到废票箱。

6.3.2.7.3 输入参数说明

hCom: 模块句柄。

6.3.2.7.4 输出参数说明

无。

6.3.2.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.8 盘点

6.3.2.8.1 函数原型

INT32 Szmc\_STM\_StartInventory(SZMCHANDLE hCom, SZMC\_STM\_INVENTORY\_INFO & info)

6.3.2.8.2 函数功能描述

开始盘点。

6.3.2.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_STM\_INVENTORY\_INFO info:

typedef struct \_szmc\_stm\_inventory\_info

{

SZMC\_STM\_BOX\_INVENTORY\_INFO boxInventoryInfo[MAX\_TICKET\_BOX\_NUM];

}SZMC\_STM\_INVENTORY\_INFO,\*LPSZMC\_STM\_INVENTORY\_INFO;

typedef struct \_szmc\_stm\_box\_inventory\_info

{

UINT32 uiBoxNo;

BOOL bInventory;

UINT32 uiInventoryNum;

}SZMC\_STM\_BOX\_INVENTORY\_INFO,\*LPSZMC\_STM\_BOX\_INVENTORY\_INFO;

uiBoxNo: 票箱号。

bInventory: 是否盘点。bInventory 为下列值之一:

值	含义
SZMC_STM_INVENTORY	盘点
SZMC_STM_NOINVENTORY	不盘点

6.3.2.8.4 输出参数说明

```
SZMC_STM_INVENTORY_INFO info:

typedef struct _szmc_stm_inventory_info
{
    SZMC_STM_BOX_INVENTORY_INFO boxInventoryInfo[MAX_TICKET_BOX_NUM];
}SZMC_STM_INVENTORY_INFO,*LPSZMC_STM_INVENTORY_INFO;

typedef struct _szmc_stm_box_inventory_info
{
    UINT32 uiBoxNo;
    BOOL bInventory;
    UINT32 uiInventoryNum;
}SZMC_STM_BOX_INVENTORY_INFO, *LPSZMC_STM_BOX_INVENTORY_INFO;

uiInventoryNum: 实际盘点数量。
```

6.3.2.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.9 停止盘点

6.3.2.9.1 函数原型

```
INT32 SzmC_STM_StopInventory(SZMCHANDLE hCom, SZMC_STM_INVENTORY_INFO & info)
```

6.3.2.9.2 函数功能描述

停止盘点。

6.3.2.9.3 输入参数说明

hCom: 模块句柄。

```
SZMC_STM_INVENTORY_INFO info:

typedef struct _szmc_stm_inventory_info
{
    SZMC_STM_BOX_INVENTORY_INFO boxInventoryInfo[MAX_TICKET_BOX_NUM];
}SZMC_STM_INVENTORY_INFO,*LPSZMC_STM_INVENTORY_INFO;
```

```
typedef struct _szmc_stm_box_inventory_info
{
    UINT32 uiBoxNo;

    BOOL bInventory;

    UINT32 uiInventoryNum;
}SZMC_STM_BOX_INVENTORY_INFO, *LPSZMC_STM_BOX_INVENTORY_INFO;
```

uiBoxNo: 票箱号。

bStartInventory: 是否盘点。bStartInventory 为下列值之一:

值	含义
SZMC_STM_INVENTORY	不停止盘点
SZMC_STM_NOINVENTORY	停止盘点

6.3.2.9.4 输出参数说明

```
SZMC_STM_INVENTORY_INFO info:

typedef struct _szmc_stm_inventory_info
{
    SZMC_STM_BOX_INVENTORY_INFO boxInventoryInfo[MAX_TICKET_BOX_NUM];
}SZMC_STM_INVENTORY_INFO,*LPSZMC_STM_INVENTORY_INFO;
```

```
typedef struct _szmc_stm_box_inventory_info
{
    UINT32 uiBoxNo;

    BOOL bInventory;

    UINT32 uiInventoryNum;
```

```
} SZMC_STM_BOX_INVENTORY_INFO,*LPSZMC_STM_BOX_INVENTORY_INFO;
```

uiInventoryNum: 实际盘点数量。

6.3.2.9.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.3.2.10 通用命令接口

6.3.2.10.1 函数原型

```
INT32 Szmc_STM_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo,LPVOID lpIn,LPVOID lpOut)
```

6.3.2.10.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.3.2.10.3 输入参数说明

hCom: 模块句柄。

uiCmdNo: 命令号。

lpIn: 输入参数字符串。

6.3.2.10.4 输出参数说明

lpOut: 输出参数字符串。

6.3.2.10.5 函数返回值说明

返回值至少包含下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	执行命令失败

6.4 单程票回收机构

6.4.1 基本函数接口

6.4.1.1 打开模块

6.4.1.1.1 函数原型

INT32 SzmC\_RTM\_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)

6.4.1.1.2 函数功能描述

建立与模块之间的通信。

6.4.1.1.3 输入参数说明

iCom: 模块句柄。

6.4.1.1.4 输出参数说明

无。

6.4.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.2 关闭模块

6.4.1.2.1 函数原型

INT32 SzmC\_RTM\_Close (SZMCHANDLE hCom)

6.4.1.2.2 函数功能描述

断开与模块的通讯。

6.4.1.2.3 输入参数说明

hCom: 模块句柄。

6.4.1.2.4 输出参数说明

无。

6.4.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.3 获取操作错误信息

6.4.1.3.1 函数原型

VOID Szmc\_RTM\_GetLastError(SZMCHANDLE hCom, SZMC\_RTM\_ERR\_INFO& info)

6.4.1.3.2 函数功能描述

获取最后一次对模块操作的的错误信息。

6.4.1.3.3 输入参数说明

hCom: 模块句柄。

6.4.1.3.4 输出参数说明

info:  
typedef \_szmc\_rtm\_err\_info  
{

INT32 iErrorNo;  
  
LPSTR lpszPhyErrorNo;  
  
LPSTR lpszErrInfo;  
  
} SZMC\_RTM\_ERR\_INFO;

iErrorNo: 逻辑错误码。

lpzPhyErrorNo: 物理错误码。

lpzErrInfo: 错误描述。

#### 6.4.1.3.5 函数返回值说明

无。

#### 6.4.1.4 获取模块信息

##### 6.4.1.4.1 函数原型

INT32 SzmC\_RTM\_GetInfo(SZMCHANDLE hCom,SZMC\_RTM\_INFO& info)

##### 6.4.1.4.2 函数功能描述

获取模块相关信息。

##### 6.4.1.4.3 输入参数说明

hCom: 模块句柄。

##### 6.4.1.4.4 输出参数说明

info:

```
typedef struct _szmc_rtm_info
```

```
{
```

```
    LPSTR lpzRTMVer;
```

```
    LPSTR lpzDriverVer;
```

```
    LPSTR lpzManufacturerVer;
```

```
    LPSTR lpzRTMID;
```

```
    UINT32 uiDeviceType;
```

```
    LPSTR lpzDeviceName;
```

```
    UINT32 uiStatus;
```

```
} SZMC_RTM_INFO,*LPSZMC_RTM_INFO;
```

lpzRTMVer: 模块版本。

lpzDriverVer: 模块驱动版本。



lpszManufacturerVer: 制造商版本。

lpszRTMID: 模块 ID。

uiDeviceType: 模块类型。

lpszDeviceName: 模块类型名。

uiStatus: 模块状态。uiStatus 为下列值之一:

值	含义
SZMC_RTM_NORMAL	模块正常
SZMC_RTM_JAM	模块卡票
SZMC_RTM_BOX_PAUSE	因无票箱可用导致模块无法使用
SZMC_RTM_SENSOR_PAUSE	因传感器故障导致模块无法使用
SZMC_RTM_OTHER_PAUSE	因其他原因导致模块无法使用

6.4.1.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.5 获取模块状态

6.4.1.5.1 函数原型

INT32 Szmc\_RTM\_GetStatus(SZMCHANDLE hCom,SZMC\_RTM\_STATUS& status)

6.4.1.5.2 函数功能描述

获取模块状态。

6.4.1.5.3 输入参数说明

hCom: 模块句柄。

SZMC\_RTM\_STATUS status:

typedef \_szmc\_rtm\_status

{

    UINT32 uiDevice;

    UINT32 uiDevPosition;

    LPSTR lpszExtra;

} SZMC\_RTM\_STATUS,\*LPSZMC\_RTM\_STATUS;

uiDevice: 部件状态。uiDevice 为下列值之一:

值	含义
SZMC_STATUS_RTM_IDLE	模块空闲中
SZMC_STATUS_RTM_BUSY	模块工作中
SZMC_STATUS_RTM_OFFLINE	模块未连接
SZMC_STATUS_RTM_COMERR	模块无法连接
SZMC_STATUS_RTM_HWERR	模块硬件故障
SZMC_STATUS_RTM_NOCOMP	无此模块或者未配置该模块

uiDevPosition: 设备的位置。iDevPosition 为下列值之一:

值	含义
SZMC_RTM_INPOSITION	部件在其正常的运行位置
SZMC_RTM_NOTINPOSITION	部件已经从其正常的运行位置移走
SZMC_RTM_POSUNKNOWN	部件的位置无法确定
SZMC_RTM_POSNOTSUPPLY	该部件不具备探测位置的能力

lpszExtra: 指向厂商特有或任何其它扩展信息的一个列表。返回信息为一系列“键=值”字符串以便 SP 进行扩展。每一个字符串都是以空字符 NULL 结束的, 最后一个字符串以两个空字符 NULL 结束。一个空列表可通过一个空指针或两个连续的空字符 NULL 来表示。

#### 6.4.1.5.4 输出参数说明

无。

#### 6.4.1.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

#### 6.4.1.6 重启模块

##### 6.4.1.6.1 函数原型

INT32 SzmC\_RTM\_Reset(SZMCHANDLE hCom)

##### 6.4.1.6.2 函数功能描述

模块软复位。

6.4.1.6.3 输入参数说明

hCom: 模块句柄。

6.4.1.6.4 输出参数说明

无。

6.4.1.6.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.7 设置参数

6.4.1.7.1 函数原型

INT32 SzmC\_RTM\_SetParam(SZMCHANDLE hCom, SZMC\_RTM\_PARAM\_INFO info)

6.4.1.7.2 函数功能描述

设置模块参数。

6.4.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_RTM\_PARAM\_INFO info:

typedef struct \_szmc\_rtm\_param\_info

{

LPSTR lpszParamName;

LPSTR lpszParamValue;

} SZMC\_RTM\_PARAM\_INFO,\*LPSZMC\_RTM\_PARAM\_INFO;

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.4.1.7.4 输出参数说明

无。

6.4.1.7.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.7.6 其他说明

应包括部件日志的参数设置。

6.4.1.8 获取参数

6.4.1.8.1 函数原型

INT32 Szmc\_RTM\_GetParam(SZMCHANDLE hCom, SZMC\_RTM\_PARAM\_INFO& info)

6.4.1.8.2 函数功能描述

获取模块参数。

6.4.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_RTM\_PARAM\_INFO info:

typedef struct \_szmc\_rtm\_param\_info

{

LPSTR lpszParamName;

LPSTR lpszParamValue;

} SZMC\_RTM\_PARAM\_INFO,\*LPSZMC\_RTM\_PARAM\_INFO;

lpszParamName: 参数名。

6.4.1.8.4 输出参数说明

SZMC\_RTM\_PARAM\_INFO info:

```
typedef struct _szmc_rtm_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_RTM_PARAM_INFO,*LPSZMC_RTM_PARAM_INFO;
```

lpszParamValue: 参数值。

6.4.1.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.9 设置模式

6.4.1.9.1 函数原型

```
INT32 Szmc_RTM_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)
```

6.4.1.9.2 函数功能描述

设置各种模式。

6.4.1.9.3 输入参数说明

hCom: 模块句柄。  
uiModeID: 模式 ID。

6.4.1.9.4 输出参数说明

无。

6.4.1.9.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_ERR_RTM_EXECUTE	命令执行失败

6.4.1.10 获取模式

6.4.1.10.1 函数原型

INT32 Szmc\_RTM\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.4.1.10.2 函数功能描述

获取当前模式。

6.4.1.10.3 输入参数说明

hCom: 模块句柄。

6.4.1.10.4 输出参数说明

uiModeID: 模式 ID。

6.4.1.10.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.11 设置寄存器

6.4.1.11.1 函数原型

INT32 Szmc\_RTM\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.4.1.11.2 函数功能描述

设置各种寄存器。

6.4.1.11.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

lpszRegValue: 寄存器值。

6.4.1.11.4 输出参数说明

无。

6.4.1.11.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.12 获取寄存器值

6.4.1.12.1 函数原型

INT32 SzmC\_RTM\_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpszRegValue)

6.4.1.12.2 函数功能描述

获取各种寄存器值。

6.4.1.12.3 输入参数说明

hCom：模块句柄。

uiRegID：寄存器 ID。

6.4.1.12.4 输出参数说明

lpszRegValue：寄存器值。

6.4.1.12.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.13 自检

6.4.1.13.1 函数原型

INT32 SzmC\_RTM\_SelfTest(SZMCHANDLE hCom,UINT32 uiSelfTestType)

6.4.1.13.2 函数功能描述

模块自检。

6.4.1.13.3 输入参数说明

hCom: 模块句柄。

uiSelfTestType: 自检类型。iSelfTestType 为下列值之一:

值	含义
SZMC_RTM_SIMPLE_SELFTEST	简单自检
SZMC_RTM_HIGH_SELFTEST	高级自检

6.4.1.13.4 输出参数说明

无。

6.4.1.13.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.1.14 软件更新

6.4.1.14.1 函数原型

INT32 SzmC\_RTM\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.4.1.14.2 函数功能描述

软件更新。

6.4.1.14.3 输入参数说明

hCom: 模块句柄。

lpszPara: 软件更新时使用的参数，比如更新路径及文件名等。

6.4.1.14.4 输出参数说明

无



6.4.1.14.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2 其他函数接口

6.4.2.1 获取票箱信息

6.4.2.1.1 函数原型

INT32 SzmC\_RTM\_GetBoxInfo(SZMCHANDLE hCom, UINT32  
uiBoxNo ,LPSZMC\_RTM\_BOXINFO& lpBoxInfo)

6.4.2.1.2 函数功能描述

获取票箱信息。

6.4.2.1.3 输入参数说明

hCom: 模块句柄。  
uiBoxNo: 票箱号。当 uiBoxNo 为 0 时，则查询所有票箱信息。

6.4.2.1.4 输出参数说明

LPSZMC\_RTM\_BOXINFO lpBoxInfo:  
typedef struct \_szmc\_RTM\_box\_info  
{  
    UINT32 uiCount;  
    SZMC\_RTM\_BOXUNIT boxUnits[MAX\_RTM\_BOX\_NUM];  
} SZMC\_RTM\_BOXINFO,\*LPSZMC\_RTM\_BOXINFO;  
uiCount: 要查询的票箱的返回数。  
SZMC\_RTM\_BOXUNIT boxUnits:  
typedef struct \_szmc\_rtm\_box\_unit  
{

```
UINT32 uiBoxNo;

UINT32 uiTicketType;

LPSTR lpszBoxID;

UINT32 uiBoxEmptyNum;

UINT32 uiBoxWillEmptyNum;

UINT32 uiTicketNum;

UINT32 uiStatus;

} SZMC_RTM_BOXUNIT,* LPSZMC_RTM_BOXUNIT;
```

uiBoxNo: 票箱号。

uiTicketType: 回收单程票类型。iTicketType 为下列值之一:

值	含义
SZMC_TICKET_COMMON	普通单程票
SZMC_TICKET_CHILDREN	儿童单程票
SZMC_TICKET_VIP	VIP 单程票
SZMC_TICKET_OTHER	其他单程票

lpszBoxID: 票箱 ID 号。

uiBoxEmptyNum: 票箱空数量。

uiBoxWillEmptyNum: 票箱将空数量。

uiTicketNum: 现有数量。

uiStatus: 票箱当前状态。iStatus 为下列值之一:

值	含义
SZMC_RTM_BOX_STAT_OK	票箱正常
SZMC_RTM_BOX_STAT_EMPTY	票箱空
SZMC_RTM_BOX_STAT_WILL_EMPT	票箱将空
Y	
SZMC_RTM_BOX_STAT_NO_EXIST	该票箱不存在
SZMC_RTM_BOX_STAT_MISSING	该票箱不在合法的位置上
SZMC_RTM_BOX_STAT_UNABLE	该票箱设定为不可用

#### 6.4.2.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2.2 获取模块传感器状态

6.4.2.2.1 函数原型

```
INT32 SzmC_RTM_GetSensorInfo(SZMCHANDLE hCom,SZMC_RTM_SENSOR_INFO& info)
```

6.4.2.2.2 函数功能描述

获取所有模块传感器状态。

6.4.2.2.3 输入参数说明

hCom: 模块句柄。

6.4.2.2.4 输出参数说明

```
SZMC_RTM_SENSOR_INFO& info:
typedef struct _szmc_rtm_sensor_info
{
    UINT32 uiSensorCount;
    UINT32 uiSensorStatus[MAX_RTM_SENSOR_NUM];
} SZMC_RTM_SENSOR_INFO,*LPSZMC_RTM_SENSOR_INFO;
```

uiSensorCount: 实际的传感器数。最大数不能超过 MAX\_RTM\_SENSOR\_NUM。

uiSensorStatus: 传感器状态。uiSensorStatus 为下列值之一:

值	含义
SZMC_RTM_SENSOR_BLOCKED	传感器完全挡住了
SZMC_RTM_SENSOR_PENETRATE_LITTLE	传感器穿透了一部分
SZMC_RTM_SENSOR_PENETRATE_HALF	传感器穿透了一半
SZMC_RTM_SENSOR_PENETRATE_LOT	传感器穿透了大部分
SZMC_RTM_SENSOR_UNBLOCKED	传感器完全没有被挡住
SZMC_RTM_SENSOR_OTHER	其他状态

6.4.2.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2.3 回收

6.4.2.3.1 函数原型

INT32 Szmc\_RTM\_Reclaim(SZMCHANDLE hCom,SZMC\_RECLAIM\_INFO info)

6.4.2.3.2 函数功能描述

回收票卡到票箱。

6.4.2.3.3 输入参数说明

hCom: 模块句柄。

SZMC\_RECLAIM\_INFO info:

typedef struct \_szmc\_reclaim\_info

```
{
    UINT32 uiReclaimType;
    UINT32 uiReclaimBox;
} SZMC_RECLAIM_INFO,*LPSZMC_RECLAIM_INFO;
```

uiReclaimType: 回收票卡类型。

uiReclaimBox: 票箱号。当同一回收票卡类型的票箱数大于 1 时有效。票箱号从 1 开始编码，所有回收票卡类型的票箱统一编码，下同。

6.4.2.3.4 输出参数说明

6.4.2.3.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2.4 退票

6.4.2.4.1 函数原型

INT32 Szmc\_RTM\_ExitTicket(SZMCHANDLE hCom)

6.4.2.4.2 函数功能描述

退出票卡到退票口。

6.4.2.4.3 输入参数说明

hCom: 模块句柄。

6.4.2.4.4 输出参数说明

无。

6.4.2.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2.5 切换票箱

6.4.2.5.1 函数原型

INT32 SzmC\_RTM\_SwitchBox(SZMCHANDLE hCom,UINT32 uiBoxNo)

6.4.2.5.2 函数功能描述

切换到指定票箱。

6.4.2.5.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 要切换到的票箱号。

6.4.2.5.4 输出参数说明

无。

6.4.2.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.4.2.6 通用命令接口

6.4.2.6.1 函数原型

INT32 SzmC\_RTM\_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo,LPVOID lpIn,LPVOID lpOut)

6.4.2.6.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.4.2.6.3 输入参数说明

hCom: 模块句柄。

uiCmdNo: 命令号。

lpIn: 输入参数字符串。

6.4.2.6.4 输出参数说明

lpOut: 输出参数字符串。

6.4.2.6.5 函数返回值说明

返回值至少包含下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5 通行控制模块

6.5.1 基本控制命令

6.5.1.1 打开模块

6.5.1.1.1 函数原型

```
INT32 Szmc_PCM_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)
```

6.5.1.1.2 函数功能描述

建立与模块之间的通信。

6.5.1.1.3 输入参数说明

- hCom: 模块句柄。
- lpszLogicalName: 部件预定义逻辑名称。

6.5.1.1.4 输出参数说明

无。

6.5.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.2 关闭模块

6.5.1.2.1 函数原型

```
INT32 Szmc_PCM_Close (SZMCHANDLE hCom)
```

6.5.1.2.2 函数功能描述

断开与模块的通讯。

6.5.1.2.3 输入参数说明

hCom: 句柄。

6.5.1.2.4 输出参数说明

无。

6.5.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.3 获取操作错误信息

6.5.1.3.1 函数原型

VOID Szmc\_PCM\_GetLastError(SZMCHANDLE hCom, SZMC\_PCM\_ERR\_INFO& info)

6.5.1.3.2 函数功能描述

获取最后一次对模块操作的错误信息。

6.5.1.3.3 输入参数说明

hCom: 模块句柄。

6.5.1.3.4 输出参数说明

SZMC\_PCM\_ERR\_INFO& info:

typedef \_szmc\_pcm\_err\_info

```
{
    INT32 iErrorNo;
    LPSTR lpszPhyErrorNo;
    LPSTR lpszErrInfo;
} SZMC_PCM_ERR_INFO;
```

iErrorNo: 逻辑错误码。



lpzPhyErrorNo: 物理错误码。

lpzErrInfo: 错误描述。

#### 6.5.1.3.5 函数返回值说明

无。

#### 6.5.1.4 获取模块信息

##### 6.5.1.4.1 函数原型

INT32 Szmc\_PCM\_GetInfo(SZMCHANDLE hCom,SZMC\_PCM\_INFO& info)

##### 6.5.1.4.2 函数功能描述

获取模块相关信息。

##### 6.5.1.4.3 输入参数说明

hCom: 模块句柄。

##### 6.5.1.4.4 输出参数说明

SZMC\_PCM\_INFO& info:

typedef struct \_szmc\_pcm\_info

{

LPSTR lpzPCMVer;

LPSTR lpzDriverVer;

LPSTR lpzManufacturerVer;

LPSTR lpzPCMid;

UNIT32 uiDeviceType;

UNIT32 uiStatus;

} SZMC\_PCM\_INFO, \*LPSZMC\_PCM\_INFO;

szPCMVer: 模块版本。

szDriverVer: 模块驱动版本。

szManufacturerVer: 制造商版本。

szPCMid: 模块 ID。

uiDeviceType: 模块类型。

uiStatus: 模块状态。uiStatus 为下列值之一:

值	含义
SZMC_PCM_NORMAL	模块正常
SZMC_PCM_ERROR	模块故障

6.5.1.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.5 获取模块状态

6.5.1.5.1 函数原型

INT32 SzmC\_PCM\_GetStatus(SZMCHANDLE hCom,SZMC\_PCM\_STATUS\_INFO& info)

6.5.1.5.2 函数功能描述

查询模块状态。

6.5.1.5.3 输入参数说明

hCom: 模块句柄。

6.5.1.5.4 输出参数说明

SZMC\_PCM\_STATUS\_INFO& info:

typedef \_szmc\_pcm\_status\_info

```
{  
    BOOL bEmergency;  
    BOOL bMaintain;  
    BOOL bAlarm;  
    UNIT32 uiUnPassedNum[2];  
    UNIT32 uiReserved;
```

} SZMC\_PCM\_STATUS\_INFO, \*LPSZMC\_PCM\_STATUS\_INFO;

bEmergency: 紧急标志。bEmergency 为下列值之一:

值	含义
SZMC_PCM_STATUS_EMERGENCY	紧急
SZMC_PCM_STATUS_NOTEMERGEN	非紧急
CY	

bMaintain: 维护标志。bMaintain 为下列值之一:

值	含义
SZMC_PCM_STATUS_INMAITAIN	维护中
SZMC_PCM_STATUS_OUTMAINTAIN	非维护

bAlarm: 告警标志。bAlarm 为下列值之一:

值	含义
SZMC_PCM_STATUS_ALARM	报警中
SZMC_PCM_STATUS_NOTALARM	非报警

uiUnPassedNum: 未通讯数量。其中 uiUnPassedNum[0]为进站方向未通行人数, uiUnPassedNum[1]为出站方向未通行人数。

uiReserved: 预留标志。

6.5.1.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.6 重启模块

6.5.1.6.1 函数原型

INT32 Szmc\_PCM\_Reset(SZMCHANDLE hCom)

6.5.1.6.2 函数功能描述

模块软复位。

6.5.1.6.3 输入参数说明

hCom: 模块句柄。

6.5.1.6.4 输出参数说明

无。

6.5.1.6.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.7 设置参数

6.5.1.7.1 函数原型

INT32 Szmc\_PCM\_SetParam(SZMCHANDLE hCom,SZMC\_PCM\_PARAM\_INFO info)

6.5.1.7.2 函数功能描述

设置模块参数。

6.5.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_PCM\_PARAM\_INFO info:

typedef struct \_szmc\_pcm\_param\_info

{

LPSTR lpszParamName;

LPSTR lpszParamValue;

} SZMC\_PCM\_PARAM\_INFO,\*LPSZMC\_PCM\_PARAM\_INFO;

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.5.1.7.4 输出参数说明

6.5.1.7.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.8 获取参数

6.5.1.8.1 函数原型

```
INT32 Szmc_PCM_GetParam(SZMCHANDLE hCom, SZMC_PCM_PARAM_INFO & info)
```

6.5.1.8.2 函数功能描述

获取模块参数。

6.5.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_PCM\_PARAM\_INFO info:

```
typedef struct _szmc_pcm_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_PCM_PARAM_INFO,*LPSZMC_PCM_PARAM_INFO;
```

lpszParamName: 参数名。

6.5.1.8.4 输出参数说明

SZMC\_PCM\_PARAM\_INFO info:

```
typedef struct _szmc_pcm_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_PCM_PARAM_INFO,*LPSZMC_PCM_PARAM_INFO;
```

lpszParamValue: 参数值。

6.5.1.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_ERR_PCM_EXECUTE	命令执行失败

6.5.1.9 设置模式

6.5.1.9.1 函数原型

INT32 Szmc\_PCM\_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)

6.5.1.9.2 函数功能描述

设置各种模式。

6.5.1.9.3 输入参数说明

hCom: 模块句柄。

uiModeID: 模式 ID。

6.5.1.9.4 输出参数说明

无。

6.5.1.9.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.10 获取模式

6.5.1.10.1 函数原型

INT32 Szmc\_PCM\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.5.1.10.2 函数功能描述

获取当前模式。

6.5.1.10.3 输入参数说明

hCom: 模块句柄。

6.5.1.10.4 输出参数说明

uiModeID: 模式 ID。

6.5.1.10.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.11 设置寄存器

6.5.1.11.1 函数原型

INT32 Szm\_PCM\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.5.1.11.2 函数功能描述

设置各种寄存器。

6.5.1.11.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

lpszRegValue: 寄存器值。

6.5.1.11.4 输出参数说明

无。

6.5.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.12 获取寄存器值

6.5.1.12.1 函数原型

INT32 Szmc\_PCM\_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpszRegValue)

6.5.1.12.2 函数功能描述

获取各种寄存器值。

6.5.1.12.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

6.5.1.12.4 输出参数说明

lpszRegValue: 寄存器值。

6.5.1.12.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.13 自检控制命令

6.5.1.13.1 函数原型

INT32 Szmc\_PCM\_TestControl(SZMCHANDLE hCom, uiSelfTestType)

6.5.1.13.2 函数功能描述

自检。

6.5.1.13.3 输入参数说明

hCom: 模块句柄。

uiSelfTestType: 自检类型。uiSelfTestType 为下列值之一:



值	含义
SZMC_STM_SIMPLE_SELFTEST	简单自检
SZMC_STM_HIGH_SELFTEST	高级自检

6.5.1.13.4 输出参数说明

无。

6.5.1.13.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.1.14 软件更新

6.5.1.14.1 函数原型

INT32 SzmC\_PCM\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.5.1.14.2 函数功能描述

软件更新。

6.5.1.14.3 输入参数说明

hCom: 模块句柄。

lpszPara: 软件更新时使用的参数，比如更新路径及文件名等。

6.5.1.14.4 输出参数说明

无。

6.5.1.14.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2 其他命令

6.5.2.1 获取模块传感器状态

6.5.2.1.1 函数原型

INT32 Szmc\_PCM\_GetSensorInfo(SZMCHANDLE hCom,SZMC\_PCM\_SENSOR\_INFO& info)

6.5.2.1.2 函数功能描述

获取所有模块传感器状态。

6.5.2.1.3 输入参数说明

hCom: 模块句柄。

6.5.2.1.4 输出参数说明

SZMC\_PCM\_SENSOR\_INFO& info:

typedef struct \_szmc\_pcm\_sensor\_info

{

UNIT32 uiSensorCount;

UNIT32 uiSensorStatus[MAX\_PCM\_SENSOR\_NUM];

} SZMC\_PCM\_SENSOR\_INFO, \*LPSZMC\_PCM\_SENSOR\_INFO;

uiSensorCount: 实际的传感器数。最大数不能超过 MAX\_PCM\_SENSOR\_NUM。

uiSensorStatus: 传感器状态。uiSensorStatus 为下列值之一:

值	含义
SZMC_PCM_SENSOR_BLOCKED	传感器完全挡住了
SZMC_PCM_SENSOR_PENETRATE_LITTLE	传感器穿透了一部分
SZMC_PCM_SENSOR_PENETRATE_HALF	传感器穿透了一半
SZMC_PCM_SENSOR_PENETRATE_LOT	传感器穿透了大部分
SZMC_PCM_SENSOR_UNBLOCKED	传感器完全没有被挡住
SZMC_PCM_SENSOR_OTHER	其他状态

6.5.2.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2.2 开门

6.5.2.2.1 函数原型

```
INT32 Szmcm_PCM_Open(SZMCHANDLE hCom,UNIT32 uiDirection)
```

6.5.2.2.2 函数功能描述

模块开门。

6.5.2.2.3 输入参数说明

hCom: 模块句柄。  
uiDirection: 开门方向。

6.5.2.2.4 输出参数说明

无。

6.5.2.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2.3 紧急控制命令

6.5.2.3.1 函数原型

```
INT32 Szmcm_PCM_EmergeControl(SZMCHANDLE hCom,UNIT32 uiControlType)
```

6.5.2.3.2 函数功能描述

进入紧急及退出紧急命令。

6.5.2.3.3 输入参数说明

hCom: 模块句柄。  
uiControlType: 控制方式。uiControlType 为下列值之一:

值	含义
SZMC_PCM_IN_EMERGENCY	进入紧急
SZMC_PCM_OUT_EMERGENCY	退出紧急

6.5.2.3.4 输出参数说明

无。

6.5.2.3.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2.4 维护控制命令

6.5.2.4.1 函数原型

INT32 Szmc\_PCM\_MaintainControl(SZMCHANDLE hCom,UNIT32 uiControlType)

6.5.2.4.2 函数功能描述

进入维护及退出维护命令。

6.5.2.4.3 输入参数说明

hCom：模块句柄。

uiControlType：控制方式。uiControlType 为下列值之一：

值	含义
SZMC_PCM_IN_MAINTAIN	进入维护
SZMC_PCM_OUT_MAINTAIN	退出维护

6.5.2.4.4 输出参数说明

无。

6.5.2.4.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2.5 测试控制命令

6.5.2.5.1 函数原型

```
INT32 Szmc_PCM_TestControl(SZMCHANDLE hCom,UNIT32 uiControlType)
```

6.5.2.5.2 函数功能描述

进入测试及退出测试命令。

6.5.2.5.3 输入参数说明

hCom: 模块句柄。

uiControlType: 控制方式。uiControlType 为下列值之一:

值	含义
SZMC_PCM_IN_TEST	进入测试
SZMC_PCM_OUT_TEST	退出测试

6.5.2.5.4 输出参数说明

无。

6.5.2.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.5.2.6 通用命令接口

6.5.2.6.1 函数原型

```
INT32 Szmc_PCM_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo,LPVOID lpIn,LPVOID lpOut)
```

6.5.2.6.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.5.2.6.3 输入参数说明

hCom: 模块句柄。

uiCmdNo: 命令号。

lpIn: 输入参数字符串。

6.5.2.6.4 输出参数说明

lpOut: 输出参数字符串。

6.5.2.6.5 函数返回值说明

返回值至少包含下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6 硬币模块

6.6.1 基本控制命令

6.6.1.1 打开模块

6.6.1.1.1 函数原型

```
INT32 Szmc_CCM_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)
```

6.6.1.1.2 函数功能描述

建立与模块之间的通信。

6.6.1.1.3 输入参数说明

hCom: 模块句柄。  
lpszLogicalName: 部件预定义逻辑名称。

6.6.1.1.4 输出参数说明

无。

6.6.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.2 关闭模块

6.6.1.2.1 函数原型

```
INT32 Szmc_CCM_Close (SZMCHANDLE hCom)
```

6.6.1.2.2 函数功能描述

断开与模块的通讯。

6.6.1.2.3 输入参数说明

hCom: 模块句柄。

6.6.1.2.4 输出参数说明

无。

6.6.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.3 获取操作错误信息

6.6.1.3.1 函数原型

VOID Szmc\_CCM\_GetLastError(SZMCHANDLE hCom, SZMC\_CCM\_ERR\_INFO& info)

6.6.1.3.2 函数功能描述

获取最后一次对模块操作的错误信息。

6.6.1.3.3 输入参数说明

hCom: 模块句柄。

6.6.1.3.4 输出参数说明

SZMC\_CCM\_ERR\_INFO& info:

typedef \_szmc\_ccm\_err\_info

{

INT32 iErrorNo;

LPSTR lpszPhyErrorNo;

LPSTR lpszErrInfo;

} SZMC\_CCM\_ERR\_INFO;

iErrorNo: 逻辑错误码。



lpszPhyErrorNo: 物理错误码。

lpszErrInfo: 错误描述。

#### 6.6.1.3.5 函数返回值说明

无。

#### 6.6.1.4 获取模块信息

##### 6.6.1.4.1 函数原型

```
INT32 Szmc_CCM_GetInfo(SZMCHANDLE hCom,SZMC_CCM_INFO& info)
```

##### 6.6.1.4.2 函数功能描述

获取模块相关信息。

##### 6.6.1.4.3 输入参数说明

hCom: 模块句柄。

##### 6.6.1.4.4 输出参数说明

SZMC\_CCM\_INFO& info:

```
typedef struct _szmc_ccm_info
```

```
{
```

```
    LPSTR lpszCCMVer;
```

```
    LPSTR lpszDriverVer;
```

```
    LPSTR lpszManufacturerVer;
```

```
    LPSTR lpszCCMID;
```

```
    UINT32 uiDeviceType;
```

```
    UINT32 uiStatus;
```

```
} SZMC_CCM_INFO,*LPSZMC_CCM_INFO;
```

lpszCCMVer: 模块版本。

lpszDriverVer: 模块驱动版本。

lpszManufacturerVer: 制造商版本。

lpszCCMID: 模块 ID。

uiDeviceType: 模块类型。

uiStatus: 模块状态。uiStatus 为下列值之一:

值	含义
SZMC_STATUS_CCM_IDLE	模块空闲中
SZMC_STATUS_CCM_BUSY	模块工作中
SZMC_STATUS_CCM_UNINITIAL	模块未初始化
SZMC_STATUS_CCM_COM_ERR	模块通讯故障
SZMC_STATUS_CCM_ERROR	模块故障

6.6.1.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.5 获取模块状态

6.6.1.5.1 函数原型

INT32 Szmc\_CCM\_GetStatus(SZMCHANDLE hCom,UINT32& iStatus)

6.6.1.5.2 函数功能描述

获取模块状态。

6.6.1.5.3 输入参数说明

hCom: 模块句柄

iStatus: 传感器状态。iStatus 为下列值之一:

值	含义
SZMC_STATUS_CCM_IDLE	模块空闲中
SZMC_STATUS_CCM_BUSY	模块工作中
SZMC_STATUS_CCM_UNINITIAL	模块未初始化
SZMC_STATUS_CCM_COM_ERR	模块通讯故障
SZMC_STATUS_CCM_ERROR	模块故障

6.6.1.5.4 输出参数说明

无。

6.6.1.5.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.6 重启模块

6.6.1.6.1 函数原型

INT32 Szmc\_CCM\_Reset(SZMCHANDLE hCom)

6.6.1.6.2 函数功能描述

模块软复位。

6.6.1.6.3 输入参数说明

hCom：模块句柄。

6.6.1.6.4 输出参数说明

无。

6.6.1.6.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.7 设置参数

6.6.1.7.1 函数原型

INT32 Szmc\_CCM\_SetParam(SZMCHANDLE hCom, SZMC\_CCM\_PARAM\_INFO info)

6.6.1.7.2 函数功能描述

设置模块参数。

6.6.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_CCM\_PARAM\_INFO info:

typedef struct \_szmc\_ccm\_param\_info

{

LPSTR lpszParamName;

LPSTR lpszParamValue;

} SZMC\_CCM\_PARAM\_INFO,\*LPSZMC\_CCM\_PARAM\_INFO;

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.6.1.7.4 输出参数说明

无。

6.6.1.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.8 获取参数

6.6.1.8.1 函数原型

INT32 Szmc\_CCM\_GetParam(SZMCHANDLE hCom, SZMC\_CCM\_PARAM\_INFO& info)

6.6.1.8.2 函数功能描述

获取模块参数。

6.6.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_CCM\_PARAM\_INFO& info:

typedef struct \_szmc\_ccm\_param\_info

```
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_CCM_PARAM_INFO,*LPSZMC_CCM_PARAM_INFO;
lpszParamName: 参数名。
```

6.6.1.8.4 输出参数说明

```
SZMC_CCM_PARAM_INFO& info:
typedef struct _szmc_ccm_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_CCM_PARAM_INFO,*LPSZMC_CCM_PARAM_INFO;
lpszParamValue: 参数值。
```

6.6.1.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.9 设置模式

6.6.1.9.1 函数原型

```
INT32 Szmc_CCM_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)
```

6.6.1.9.2 函数功能描述

设置各种模式。

6.6.1.9.3 输入参数说明

hCom: 模块句柄。  
uiModeID: 模式 ID

6.6.1.9.4 输出参数说明

无。

6.6.1.9.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.10 获取模式

6.6.1.10.1 函数原型

INT32 Szmcc\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.6.1.10.2 函数功能描述

获取当前模式。

6.6.1.10.3 输入参数说明

hCom：模块句柄。

6.6.1.10.4 输出参数说明

uiModeID：模式 ID。

6.6.1.10.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.11 设置寄存器

6.6.1.11.1 函数原型

INT32 Szmcc\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.6.1.11.2 函数功能描述

设置各种寄存器。

6.6.1.11.3 输入参数说明

- hCom: 模块句柄。
- uiRegID: 寄存器 ID。
- lpzRegValue: 寄存器值。

6.6.1.11.4 输出参数说明

无。

6.6.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.12 获取寄存器值

6.6.1.12.1 函数原型

INT32 Szmc\_CCM\_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpzRegValue)

6.6.1.12.2 函数功能描述

获取各种寄存器值。

6.6.1.12.3 输入参数说明

- hCom: 模块句柄。
- uiRegID: 寄存器 ID。

6.6.1.12.4 输出参数说明

lpzRegValue: 寄存器值。

6.6.1.12.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.13 自检控制命令

6.6.1.13.1 函数原型

INT32 Szmc\_CCM\_SelfTest(SZMCHANDLE hCom)

6.6.1.13.2 函数功能描述

自检。

6.6.1.13.3 输入参数说明

hCom: 模块句柄。

6.6.1.13.4 输出参数说明

无。

6.6.1.13.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.1.14 软件更新

6.6.1.14.1 函数原型

INT32 Szmc\_CCM\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.6.1.14.2 函数功能描述

软件更新。



6.6.1.14.3 输入参数说明

hCom: 模块句柄。

lpszPara: 软件更新时使用的参数，比如更新路径及文件名等。

6.6.1.14.4 输出参数说明

无

6.6.1.14.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2 其他命令

6.6.2.1 获取钱箱信息

6.6.2.1.1 函数原型

INT32 Szmc\_CCM\_GetBoxInfo(SZMCHANDLE hCom,UINT32 uiBoxNo, SZMC\_CCM\_BOXINFO& stBoxInfo)

6.6.2.1.2 函数功能描述

获取钱箱信息。

6.6.2.1.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 钱箱号。当 uiBoxNo 为 0 时，则查询所有钱箱信息。

6.6.2.1.4 输出参数说明

SZMC\_CCM\_BOXINFO& stBoxInfo:

```
typedef struct _szmc_ccm_boxinfo
{
    UINT32 uiCount;
```

```
LPSZMC_CCM_BOXUNIT lpList;  
}  
SZMC_CCM_BOXINFO,*LPSZMC_CCM_BOXINFO;  
  
uiCount: 要查询的钱箱的返回数。  
  
LPSZMC_CCM_BOXUNIT lpList:  
  
typedef struct _szmc_ccm_boxunit  
{  
  
    UINT32 uiBoxNo;  
  
    BOOL bExist;  
  
    UINT32 uiCoinType;  
  
    LPSTR lpszBoxID;  
  
    UINT32 uiBoxNum;  
  
    UINT32 uiBoxWillNum;  
  
    UINT32 uiCoinNum;  
  
    UINT32 uiStatus;  
  
}  
SZMC_CCM_BOXUNIT,*LPSZMC_CCM_BOXUNIT;
```

bExist: 钱箱是否到位。只有钱箱到位，以下字段有效。bExist 为下列值之一：

值	含义
SZMC_CCM_BOX_IN_PLACE	钱箱到位
SZMC_CCM_BOX_DISLOCATION	钱箱离位

uiCoinType: 硬币类型。iCoinType 为下列值之一：

值	含义
SZMC_CCM_HALF_YUAN	5 角
SZMC_CCM_ONE_YUAN	1 元

lpszBoxID: 钱箱 ID 号。

uiBoxNum: 钱箱满/空数量。

uiBoxWillNum: 钱箱将满/将空数量。

uiCoinNum: 硬币数量。

uiStatus: 钱箱当前状态。uiStatus 为下列值之一：

值	含义
SZMC_CCM_BOX_COMMON	钱箱正常
SZMC_CCM_BOX_WILL_FULL	钱箱将满
SZMC_CCM_BOX_FULL	钱箱满

SZMC_CCM_BOX_WILL_EMPTY	钱箱将空
SZMC_CCM_BOX_EMPTY	钱箱空

6.6.2.1.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.2 开始接收硬币

6.6.2.2.1 函数原型

INT32 Szmc\_CCM\_StartAccept(SZMCHANDLE hCom)

6.6.2.2.2 函数功能描述

模块开始接收硬币。

6.6.2.2.3 输入参数说明

hCom：模块句柄。

6.6.2.2.4 输出参数说明

无。

6.6.2.2.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.3 停止接收硬币

6.6.2.3.1 函数原型

INT32 Szmc\_CCM\_StopAccept(SZMCHANDLE hCom)

6.6.2.3.2 函数功能描述

停止接收硬币。

6.6.2.3.3 输入参数说明

hCom: 模块句柄。

6.6.2.3.4 输出参数说明

无。

6.6.2.3.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.4 获取接收到的硬币信息

6.6.2.4.1 函数原型

INT32 Szmc\_CCM\_GetCoinInfo(SZMCHANDLE hCom, SZMC\_CCM\_COIN\_INFO& info)

6.6.2.4.2 函数功能描述

获取接收到的硬币信息。

6.6.2.4.3 输入参数说明

hCom: 模块句柄。

6.6.2.4.4 输出参数说明

SZMC\_CCM\_COIN\_INFO& info:

typedef struct \_szmc\_ccm\_coin\_info

{

    UINT32 uiAmount;

    UINT32 uiCount;

```
UINT32 uiNew;  
  
LPSTR lpszCashInfo;  
  
} SZMC_CCM_COIN_INFO;
```

uiAmount: 硬币总金额。单位为分。

uiCount: 硬币总数量。

uiNew: 是否有新硬币。

lpszCashInfo: 所有的硬币面值信息。各个硬币面值中间用逗号隔开，按入币顺序排列。示例：“50,100,100” 则代表第一次入币 5 角，第二次入币 1 元，第三次入币 1 元。

6.6.2.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.5 找零

6.6.2.5.1 函数原型

```
INT32 Szmc_CCM_Change(SZMCHANDLE hCom)
```

6.6.2.5.2 函数功能描述

找零。

6.6.2.5.3 输入参数说明

hCom: 模块句柄。

6.6.2.5.4 输出参数说明

无。

6.6.2.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.6 退币

6.6.2.6.1 函数原型

INT32 Szmc\_CCM\_Refund(SZMCHANDLE hCom)

6.6.2.6.2 函数功能描述

退币。

6.6.2.6.3 输入参数说明

hCom: 模块句柄。

6.6.2.6.4 输出参数说明

无。

6.6.2.6.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.7 盘点

6.6.2.7.1 函数原型

INT32 Szmc\_CCM\_StartInventory(SZMCHANDLE hCom,SZMC\_CCM\_INVENTORY\_INFO& info)

6.6.2.7.2 函数功能描述

开始盘点。

6.6.2.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_CCM\_INVENTORY\_INFO& info:

```
typedef struct _szmc_ccm_inventory_info
{
    SZMC_BOX_INVENTORY_INFO boxInventInfo[MAX_CCM_BOX_NUM];
    }SZMC_CCM_INVENTORY_INFO, *LPSZMC_CCM_INVENTORY_INFO;
```

boxInventInfo:单个钱箱盘点信息。

```
typedef struct _szmc_box_inventory_info
{
    UINT32 uiBoxNo;
    BOOL bInventory;
    UINT32 uiInventoryNum;
} SZMC_BOX_INVENTORY_INFO, *LPSZMC_BOX_INVENTORY_INFO;
```

uiBoxNo: 钱箱号。

bInventory: 是否盘点。bInventory 为下列值之一:

值	含义
SZMC_CCM_INVENTORY	盘点
SZMC_CCM_NOINVENTORY	不盘点

6.6.2.7.4 输出参数说明

SZMC\_CCM\_INVENTORY\_INFO& info:

```
typedef struct _szmc_ccm_inventory_info
{
    SZMC_BOX_INVENTORY_INFO boxInventInfo[MAX_CCM_BOX_NUM];
    }SZMC_CCM_INVENTORY_INFO, *LPSZMC_CCM_INVENTORY_INFO;
```

boxInventInfo:单个钱箱盘点信息。

```
typedef struct _szmc_box_inventory_info
{
    UINT32 uiBoxNo;
    BOOL bInventory;
    UINT32 uiInventoryNum;
} SZMC_BOX_INVENTORY_INFO,*LPSZMC_BOX_INVENTORY_INFO;
```

uiInventoryNum: 实际盘点数量。

6.6.2.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.8 停止盘点

6.6.2.8.1 函数原型

INT32 Szmc\_CCM\_StopInventory(SZMCHANDLE hCom, SZMC\_CCM\_INVENTORY\_INFO& info)

6.6.2.8.2 函数功能描述

停止盘点。

6.6.2.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_CCM\_INVENTORY\_INFO& info:

typedef struct \_szmc\_ccm\_inventory\_info

{

    SZMC\_BOX\_INVENTORY\_INFO boxInventInfo[MAX\_CCM\_BOX\_NUM];

} SZMC\_CCM\_INVENTORY\_INFO,\*LPSZMC\_CCM\_INVENTORY\_INFO;

boxInventInfo:单个钱箱盘点信息。

typedef struct \_szmc\_box\_inventory\_info

{

    UINT32 uiBoxNo;

    BOOL bInventory;

    UINT32 uiInventoryNum;

} SZMC\_BOX\_INVENTORY\_INFO,\*LPSZMC\_BOX\_INVENTORY\_INFO;

uiBoxNo: 钱箱号。

bStartInventory: 是否盘点。bStartInventory 为下列值之一:



值	含义
SZMC_CCM_INVENTORY	不停止盘点
SZMC_CCM_NOINVENTORY	停止盘点

6.6.2.8.4 输出参数说明

```
info:
SZMC_CCM_INVENTORY_INFO& info:
typedef struct _szmc_ccm_inventory_info
{
    SZMC_BOX_INVENTORY_INFO boxInventInfo[MAX_CCM_BOX_NUM];
} SZMC_CCM_INVENTORY_INFO,*LPSZMC_CCM_INVENTORY_INFO;
boxInventInfo:单个钱箱盘点信息。
typedef struct _szmc_box_inventory_info
{
    UINT32 uiBoxNo;
    BOOL bInventory;
    UINT32 uiInventoryNum;
} SZMC_BOX_INVENTORY_INFO,*LPSZMC_BOX_INVENTORY_INFO;
uiInventoryNum: 实际盘点数量。
```

6.6.2.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.9 回收

6.6.2.9.1 函数原型

```
INT32 Szmccm_Collect(SZMCHANDLE hCom, SZMC_COLLECT_INFO& info)
```

6.6.2.9.2 函数功能描述

回收。

6.6.2.9.3 输入参数说明

hCom: 模块句柄。

SZMC\_COLLECT\_INFO& info:

typedef struct \_szmc\_collect\_info

```
{
    UINT32 uiBoxNo;
    UINT32 uiCollectNum;
    UINT32 uiFactNum;
} SZMC_COLLECT_INFO,*LPSZMC_COLLECT_INFO;
```

uiBoxNo: 钱箱号。

uiCollectNum: 要回收的硬币数。

6.6.2.9.4 输出参数说明

SZMC\_COLLECT\_INFO& info:

typedef struct \_szmc\_collect\_info

```
{
    UINT32 uiBoxNo;
    UINT32 uiCollectNum;
    UINT32 uiFactNum;
} SZMC_COLLECT_INFO,*LPSZMC_COLLECT_INFO;
```

uiFactNum: 实际回收的硬币数。

6.6.2.9.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.6.2.10 通用命令接口

6.6.2.10.1 函数原型

INT32 Szmcc\_CCM\_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo,LPVOID lpIn,LPVOID lpOut)

6.6.2.10.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.6.2.10.3 输入参数说明

- hCom: 模块句柄。
- uiCmdNo: 命令号。
- lpIn: 输入参数字符串。

6.6.2.10.4 输出参数说明

lpOut: 输出参数字符串。

6.6.2.10.5 函数返回值说明

返回值至少包含下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7 纸币接收模块

6.7.1 基本控制命令

6.7.1.1 打开模块

6.7.1.1.1 函数原型

INT32 SzmC\_BNA\_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)

6.7.1.1.2 函数功能描述

建立与模块之间的通信。

6.7.1.1.3 输入参数说明

hCom: 模块句柄。

lpszLogicalName: 部件预定义逻辑名称。

6.7.1.1.4 输出参数说明

无。

6.7.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.2 关闭模块

6.7.1.2.1 函数原型

INT32 SzmC\_BNA\_Close (SZMCHANDLE hCom)

6.7.1.2.2 函数功能描述

断开与模块的通讯。

6.7.1.2.3 输入参数说明

hCom: 模块句柄。

6.7.1.2.4 输出参数说明

无。

6.7.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.3 获取操作错误信息

6.7.1.3.1 函数原型

VOID Szmc\_BNA\_GetLastError(SZMCHANDLE hCom, SZMC\_BNA\_ERR\_INFO& info)

6.7.1.3.2 函数功能描述

获取错误信息的具体内容。

6.7.1.3.3 输入参数说明

hCom: 模块句柄。

6.7.1.3.4 输出参数说明

SZMC\_BNA\_ERR\_INFO& info:

```
typedef _szmc_bna_err_info
{
```

```
    INT32 iErrorNo;

    LPSTR lpszPhyErrorNo;

    LPSTR lpszErrInfo;

} SZMC_BNA_ERR_INFO;
```

iErrorNo: 逻辑错误码。

lpszPhyErrorNo: 物理错误码。

lpszErrInfo: 错误描述。

#### 6.7.1.3.5 函数返回值说明

无。

#### 6.7.1.4 获取模块信息

##### 6.7.1.4.1 函数原型

INT32 SzmC\_BNA\_GetInfo(SZMCHANDLE hCom,SZMC\_BNA\_INFO& info)

##### 6.7.1.4.2 函数功能描述

获取模块相关信息。

##### 6.7.1.4.3 输入参数说明

hCom: 模块句柄。

##### 6.7.1.4.4 输出参数说明

SZMC\_BNA\_INFO& info:

typedef struct \_szmc\_bna\_info

{

LPSTR lpszBNAVer;

LPSTR lpszDriverVer;

LPSTR lpszManufacturerVer;

LPSTR lpszBNAID;

UINT32 uiDeviceType;

UINT32 uiStatus;

} SZMC\_BNA\_INFO,\*LPSZMC\_BNA\_INFO;

lpszBNAVer: 模块版本。

lpszDriverVer: 模块驱动版本。

lpszManufacturerVer: 制造商版本。

lpszBNAID: 模块 ID。

uiDeviceType: 模块类型。

uiStatus: 模块状态。uiStatus 为下列值之一：

值	含义
SZMC_STATUS_BNA_IDLE	模块空闲中
SZMC_STATUS_BNA_BUSY	模块工作中
SZMC_STATUS_BNA_UNINITIAL	模块未初始化
SZMC_STATUS_BNA_COM_ERR	模块通讯故障
SZMC_STATUS_BNA_ERROR	模块故障

6.7.1.4.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.5 获取模块状态

6.7.1.5.1 函数原型

INT32 Szmc\_BNA\_GetStatus(SZMCHANDLE hCom,UINT32& uiStatus)

6.7.1.5.2 函数功能描述

获取模块状态。

6.7.1.5.3 输入参数说明

hCom: 模块句柄。

uiStatus: 传感器状态。uiStatus 为下列值之一：

值	含义
SZMC_STATUS_BNA_IDLE	模块空闲中
SZMC_STATUS_BNA_BUSY	模块工作中
SZMC_STATUS_BNA_UNINITIAL	模块未初始化
SZMC_STATUS_BNA_COM_ERR	模块通讯故障
SZMC_STATUS_BNA_ERROR	模块故障

6.7.1.5.4 输出参数说明

无。

6.7.1.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.6 重启模块

6.7.1.6.1 函数原型

INT32 SzmC\_BNA\_Reset(SZMCHANDLE hCom)

6.7.1.6.2 函数功能描述

模块软复位。

6.7.1.6.3 输入参数说明

hCom: 模块句柄。

6.7.1.6.4 输出参数说明

无。

6.7.1.6.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.7 设置参数

6.7.1.7.1 函数原型

INT32 SzmC\_BNA\_SetParam(SZMCHANDLE hCom, SZMC\_BNA\_PARAM\_INFO info)

6.7.1.7.2 函数功能描述

设置模块参数。



6.7.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_BNA\_PARAM\_INFO info:

```
typedef struct _szmc_bna_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_BNA_PARAM_INFO,*LPSZMC_BNA_PARAM_INFO;
```

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.7.1.7.4 输出参数说明

无。

6.7.1.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.8 获取参数

6.7.1.8.1 函数原型

```
INT32 Szmc_BNA_GetParam(SZMCHANDLE hCom, SZMC_BNA_PARAM_INFO& info)
```

6.7.1.8.2 函数功能描述

获取模块参数。

6.7.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_BNA\_PARAM\_INFO info:

```
typedef struct _szmc_bna_param_info
```

```
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_BNA_PARAM_INFO,*LPSZMC_BNA_PARAM_INFO;
lpszParamName: 参数名。
```

6.7.1.8.4 输出参数说明

SZMC\_BNA\_PARAM\_INFO info:

```
typedef struct _szmc_bna_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_BNA_PARAM_INFO,*LPSZMC_BNA_PARAM_INFO;
lpszParamValue: 参数值。
```

6.7.1.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.9 设置模式

6.7.1.9.1 函数原型

```
INT32 Szmc_BNA_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)
```

6.7.1.9.2 函数功能描述

设置各种模式。

6.7.1.9.3 输入参数说明

hCom: 模块句柄。

uiModeID: 模式 ID。

6.7.1.9.4 输出参数说明

无。

6.7.1.9.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.10 获取模式

6.7.1.10.1 函数原型

INT32 SzmC\_BNA\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.7.1.10.2 函数功能描述

获取当前模式。

6.7.1.10.3 输入参数说明

hCom：模块句柄。

6.7.1.10.4 输出参数说明

uiModeID：模式 ID。

6.7.1.10.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.11 设置寄存器

6.7.1.11.1 函数原型

INT32 SzmC\_BNA\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.7.1.11.2 函数功能描述

设置各种寄存器。

6.7.1.11.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

lpzRegValue: 寄存器值。

6.7.1.11.4 输出参数说明

无。

6.7.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.12 获取寄存器值

6.7.1.12.1 函数原型

INT32 SzmC\_BNA\_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpzRegValue)

6.7.1.12.2 函数功能描述

获取各种寄存器值。

6.7.1.12.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

6.7.1.12.4 输出参数说明

lpzRegValue: 寄存器值。

6.7.1.12.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.13 自检控制命令

6.7.1.13.1 函数原型

INT32 Szmc\_BNA\_SelfTest(SZMCHANDLE hCom)

6.7.1.13.2 函数功能描述

自检。

6.7.1.13.3 输入参数说明

hCom：模块句柄。

6.7.1.13.4 输出参数说明

无。

6.7.1.13.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.1.14 软件更新

6.7.1.14.1 函数原型

INT32 Szmc\_BNA\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.7.1.14.2 函数功能描述

软件更新。

6.7.1.14.3 输入参数说明

hCom: 模块句柄。

lpszPara: 软件更新时使用的参数，比如更新路径及文件名等。

6.7.1.14.4 输出参数说明

无。

6.7.1.14.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2 其他命令

6.7.2.1 获取钱箱信息

6.7.2.1.1 函数原型

INT32 Szmc\_BNA\_GetBoxInfo(SZMCHANDLE hCom, UINT32 uiBoxNo, SZMC\_BNA\_BOX\_INFO& info)

6.7.2.1.2 函数功能描述

获取模块钱箱在位、可用状态，钱箱信息。

6.7.2.1.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 钱箱号。当 uiBoxNo 为 0 时，则查询所有钱箱信息。

6.7.2.1.4 输出参数说明

SZMC\_BNA\_BOX\_INFO& info:

```
typedef struct _szmc_bna_box_info
{
    UINT32 uiCount;
```

```
LPSZMC_BNA_BOXUNIT lpList;
} SZMC_BNA_BOXINFO,*LPSZMC_BNA_BOXINFO;

uiCount: 要查询的钱箱的返回数。

LPSZMC_BNA_BOXUNIT lpList:

typedef struct _szmc_bna_box_unit
{
    UINT32 uiBoxNo;

    BOOL bExist;

    UINT32 uiNoteType;

    LPSTR lpszBoxID;

    UINT32 uiBoxNum;

    UINT32 uiBoxWillNum;

    UINT32 uiNoteNum;

    UINT32 uiStatus;
} SZMC_BNA_BOX_UNIT,*LPSZMC_BNA_BOX_UNIT;
```

bExist: 钱箱是否到位。只有钱箱到位，以下字段有效。bExist 为下列值之一：

值	含义
SZMC_BNA_BOX_IN_PLACE	钱箱到位
SZMC_BNA_BOX_DISLOCATION	钱箱离位

uiNoteDeno: 纸币类型。uiNoteDeno 为下列值之一：

值	含义
SZMC_BNA_ONE_YUAN	1 元
SZMC_BNA_FIVE_YUAN	5 元
SZMC_BNA_TEN_YUAN	10 元
SZMC_BNA_TWENTY_YUAN	20 元
SZMC_BNA_FIFTY_YUAN	50 元
SZMC_BNA_ONE_HUNDRED_YUAN	100 元

lpszBoxID: 钱箱 ID 号。

uiBoxNum: 钱箱满数量。

uiBoxWillNum: 钱箱将满数量。

uiNoteNum: 纸币数量。

uiStatus: 钱箱当前状态。uiStatus 为下列值之一：

值	含义
SZMC_BNA_BOX_COMMON	钱箱正常
SZMC_BNA_BOX_WILL_FULL	钱箱将满
SZMC_BNA_BOX_FULL	钱箱满

6.7.2.1.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.2 开始接收纸币

6.7.2.2.1 函数原型

INT32 SzmC\_BNA\_StartAccept(SZMCHANDLE hCom)

6.7.2.2.2 函数功能描述

模块开始接收纸币。

6.7.2.2.3 输入参数说明

hCom：模块句柄。

6.7.2.2.4 输出参数说明

无。

6.7.2.2.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.3 停止接收纸币

6.7.2.3.1 函数原型

INT32 SzmC\_BNA\_StopAccept(SZMCHANDLE hCom)



6.7.2.3.2 函数功能描述

停止接收纸币。

6.7.2.3.3 输入参数说明

hCom: 模块句柄。

6.7.2.3.4 输出参数说明

无。

6.7.2.3.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.4 获取接收到的纸币信息

6.7.2.4.1 函数原型

INT32 Szmc\_BNA\_GetCashInfo(SZMCHANDLE hCom, SZMC\_BNA\_CASH\_INFO& info)

6.7.2.4.2 函数功能描述

获取接收到的纸币信息。

6.7.2.4.3 输入参数说明

hCom: 模块句柄。

6.7.2.4.4 输出参数说明

SZMC\_BNA\_CASH\_INFO& info:

```
typedef struct _szmc_bna_cash_info
{
    UINT32 uiAmount;
    UINT32 uiCount;
```

```
UINT32 uiNew;

LPSTR lpszCashInfo;

UNIT32 uiEncashed;

LPSTR lpszEncashedCashInfo;

} SZMC_BNA_CASH_INFO;
```

uiAmount: 纸币总金额。单位为分。

uiCount: 纸币总数量。

uiNew: 是否有新纸币。

lpszCashInfo: 所有的纸币面值信息。各个纸币面值中间用逗号隔开，按入钞顺序排列。示例：“500,500,1000”则代表第一次入币 5 元，第二次入币 5 元，第三次入币 10 元。

uiEncashed: 是否有纸币压钱箱。

lpszEncashedCashInfo: 所有已压钱箱的纸币面值信息。各个纸币面值中间用逗号隔开，按入钞顺序排列。示例：“500,1000”则代表第一次压钞 5 元，第二次压钞 10 元。

6.7.2.4.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.5 压钱箱

6.7.2.5.1 函数原型

```
INT32 SzmC_BNA_Encash(SZMCHANDLE hCom)
```

6.7.2.5.2 函数功能描述

压钱箱。

6.7.2.5.3 输入参数说明

hCom: 模块句柄。

6.7.2.5.4 输出参数说明

无。

6.7.2.5.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.6 退币

6.7.2.6.1 函数原型

INT32 Szmc\_BNA\_Refund(SZMCHANDLE hCom)

6.7.2.6.2 函数功能描述

退币。

6.7.2.6.3 输入参数说明

hCom：模块句柄。

6.7.2.6.4 输出参数说明

无。

6.7.2.6.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.7 更换钱箱

6.7.2.7.1 函数原型

INT32 Szmc\_BNA\_ChangeNoteBox(SZMCHANDLE hCom)

6.7.2.7.2 函数功能描述

更换钱箱。

6.7.2.7.3 输入参数说明

hCom: 模块句柄。

6.7.2.7.4 输出参数说明

无。

6.7.2.7.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.7.2.8 通用命令接口

6.7.2.8.1 函数原型

INT32 Szmc\_BNA\_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo,LPVOID lpIn,LPVOID lpOut)

6.7.2.8.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.7.2.8.3 输入参数说明

hCom: 模块句柄。

uiCmdNo: 命令号。

lpIn: 输入参数字符串。

6.7.2.8.4 输出参数说明

lpOut: 输出参数字符串。

6.7.2.8.5 函数返回值说明

返回值至少包含下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8 纸币找零模块

6.8.1 基本控制命令

6.8.1.1 打开模块

6.8.1.1.1 函数原型

INT32 SzmC\_BND\_Open (SZMCHANDLE& hCom, LPSTR lpszLogicalName)

6.8.1.1.2 函数功能描述

建立与模块之间的通信。

6.8.1.1.3 输入参数说明

hCom: 模块句柄。

lpszLogicalName: 部件预定义逻辑名称。

6.8.1.1.4 输出参数说明

无。

6.8.1.1.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.2 关闭模块

6.8.1.2.1 函数原型

INT32 SzmC\_BND\_Close(SZMCHANDLE hCom)

6.8.1.2.2 函数功能描述

断开与模块的通讯。

6.8.1.2.3 输入参数说明

hCom: 模块句柄。

6.8.1.2.4 输出参数说明

无。

6.8.1.2.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.3 获取操作错误信息

6.8.1.3.1 函数原型

VOID SzmC\_BND\_GetLastError(SZMCHANDLE hCom, SZMC\_BND\_ERR\_INFO& info)

6.8.1.3.2 函数功能描述

获取最后一次对模块操作的错误信息。

6.8.1.3.3 输入参数说明

hCom: 模块句柄。

6.8.1.3.4 输出参数说明

info:

typedef \_szmc\_bnd\_err\_info

{

INT32 iErrorNo;

LPSTR lpszPhyErrorNo;

LPSTR lpszErrInfo;  
 } SZMC\_BND\_ERR\_INFO;  
 iErrorNo: 逻辑错误码。  
 lpszPhyErrorNo: 物理错误码。  
 lpszErrInfo: 错误描述。

#### 6.8.1.3.5 函数返回值说明

无。

#### 6.8.1.4 获取发售模块信息

##### 6.8.1.4.1 函数原型

INT32 SzmC\_BND\_GetInfo(SZMCHANDLE hCom, SZMC\_BND\_INFO& info)

##### 6.8.1.4.2 函数功能描述

获取模块相关信息。

##### 6.8.1.4.3 输入参数说明

hCom: 模块句柄。

##### 6.8.1.4.4 输出参数说明

SZMC\_BND\_INFO info:

typedef struct \_szmc\_bnd\_info

{

LPSTR lpszBNDVer;

LPSTR lpszDriverVer;

LPSTR lpszManufacturerVer;

LPSTR lpszBNDID;

UINT32 uiDeviceType;

LPSTR lpszDeviceName;

UINT32 uiStatus;

```
UINT32 uiRecycleDispensedNote;  
} SZMC_BND_INFO,*LPSZMC_BND_INFO;
```

lpszBNDVer: 模块版本。

lpszDriverVer: 模块驱动版本。

lpszManufacturerVer: 制造商版本。

lpszBNDID: 模块 ID。

iDeviceType: 模块类型。

lpszDeviceName: 模块类型名。

uiStatus: 模块状态。uiStatus 为下列值之一:

值	含义
SZMC_STATUS_BND_IDLE	模块空闲中
SZMC_STATUS_BND_BUSY	模块工作中
SZMC_STATUS_BND_OFFLINE	模块未连接
SZMC_STATUS_BND_COMERR	模块无法连接
SZMC_STATUS_BND_HWERR	模块硬件故障
SZMC_STATUS_BND_NOCOMP	无此模块或者未配置该模块

uiRecycleDispensedNote: 是否能够回收出钞口的纸币。uiRecycleDispensedNote 为下列值之一:

值	含义
SZMC_BND_CAN_RECYCLE_DISPENSED_NOTE	可回收出钞口的纸币
SZMC_BND_CAN_NOT_RECYCLE_DISPENSED_NOTE	无法回收出钞口的纸币

#### 6.8.1.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

#### 6.8.1.5 获取模块状态

##### 6.8.1.5.1 函数原型

```
INT32 SzmC_Note_GetStatus(SZMCHANDLE hCom,UINT32& uiStatus)
```



6.8.1.5.2 函数功能描述

获取模块状态。

6.8.1.5.3 输入参数说明

hCom: 模块句柄。

uiStatus: 传感器状态。uiStatus 为下列值之一:

值	含义
SZMC_STATUS_BND_IDLE	模块空闲中
SZMC_STATUS_BND_BUSY	模块工作中
SZMC_STATUS_BND_UNINITIAL	模块未初始化
SZMC_STATUS_BND_COM_ERR	模块通讯故障
SZMC_STATUS_BND_ERROR	模块故障

6.8.1.5.4 输出参数说明

无。

6.8.1.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.6 重启模块

6.8.1.6.1 函数原型

INT32 Szmc\_BND\_Reset(SZMCHANDLE hCom)

6.8.1.6.2 函数功能描述

模块软复位。

6.8.1.6.3 输入参数说明

hCom: 模块句柄。

6.8.1.6.4 输出参数说明

无。

6.8.1.6.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.7 设置参数

6.8.1.7.1 函数原型

INT32 Szmc\_BND\_SetParam(SZMCHANDLE hCom, SZMC\_BND\_PARAM\_INFO info)

6.8.1.7.2 函数功能描述

设置模块参数。

6.8.1.7.3 输入参数说明

hCom: 模块句柄。

SZMC\_BND\_PARAM\_INFO info:

typedef struct \_szmc\_bnd\_param\_info

{

LPSTR lpszParamName;

LPSTR lpszParamValue;

} SZMC\_BND\_PARAM\_INFO,\*LPSZMC\_BND\_PARAM\_INFO;

lpszParamName: 参数名。

lpszParamValue: 参数值。

6.8.1.7.4 输出参数说明

6.8.1.7.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.8 获取参数

6.8.1.8.1 函数原型

```
INT32 Szmc_BND_GetParam(SZMCHANDLE hCom, SZMC_BND_PARAM_INFO& info)
```

6.8.1.8.2 函数功能描述

获取模块参数。

6.8.1.8.3 输入参数说明

hCom: 模块句柄。

SZMC\_BND\_PARAM\_INFO info:

```
typedef struct _szmc_bnd_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_BND_PARAM_INFO,*LPSZMC_BND_PARAM_INFO;
```

lpszParamName: 参数名。

6.8.1.8.4 输出参数说明

SZMC\_BND\_PARAM\_INFO info:

```
typedef struct _szmc_bnd_param_info
{
    LPSTR lpszParamName;
    LPSTR lpszParamValue;
} SZMC_BND_PARAM_INFO,*LPSZMC_BND_PARAM_INFO;
```

lpszParamValue: 参数值。

6.8.1.8.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.9 设置模式

6.8.1.9.1 函数原型

INT32 SzmC\_BND\_SetMode(SZMCHANDLE hCom,UINT32 uiModeID)

6.8.1.9.2 函数功能描述

设置各种模式。

6.8.1.9.3 输入参数说明

hCom: 模块句柄。

uiModeID: 模式 ID。

6.8.1.9.4 输出参数说明

无。

6.8.1.9.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.10 获取模式

6.8.1.10.1 函数原型

INT32 SzmC\_BND\_GetMode(SZMCHANDLE hCom,UINT32& uiModeID)

6.8.1.10.2 函数功能描述

获取当前模式。

6.8.1.10.3 输入参数说明

hCom: 模块句柄。

6.8.1.10.4 输出参数说明

uiModeID: 模式 ID。

6.8.1.10.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.11 设置寄存器

6.8.1.11.1 函数原型

INT32 Szmcbnd\_SetReg(SZMCHANDLE hCom,UINT32 uiRegID,LPSTR lpszRegValue)

6.8.1.11.2 函数功能描述

设置各种寄存器。

6.8.1.11.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

lpszRegValue: 寄存器值。

6.8.1.11.4 输出参数说明

无。

6.8.1.11.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.12 获取寄存器值

6.8.1.12.1 函数原型

INT32 Szmc\_BND\_GetReg(SZMCHANDLE hCom,UINT32 uiRegID, LPSTR lpszRegValue)

6.8.1.12.2 函数功能描述

获取各种寄存器值。

6.8.1.12.3 输入参数说明

hCom: 模块句柄。

uiRegID: 寄存器 ID。

6.8.1.12.4 输出参数说明

lpszRegValue: 寄存器值。

6.8.1.12.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.13 自检

6.8.1.13.1 函数原型

INT32 Szmc\_BND\_SelfTest(SZMCHANDLE hCom)

6.8.1.13.2 函数功能描述

自检。

6.8.1.13.3 输入参数说明

hCom: 模块句柄。

6.8.1.13.4 输出参数说明

无。

6.8.1.13.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.1.14 软件更新

6.8.1.14.1 函数原型

INT32 SzmC\_BND\_SoftUpdate(SZMCHANDLE hCom,LPSTR lpszPara)

6.8.1.14.2 函数功能描述

软件更新。

6.8.1.14.3 输入参数说明

hCom：模块句柄。

lpszPara：软件更新时使用的参数，比如更新路径及文件名等。

6.8.1.14.4 输出参数说明

无。

6.8.1.14.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2 其他命令

6.8.2.1 获取钱箱信息

#### 6.8.2.1.1 函数原型

```
INT32 SzmC_BND_GetBoxInfo(SZMCHANDLE hCom, UINT32 uiBoxNo,  
SZMC_BND_BOX_INFO& info)
```

#### 6.8.2.1.2 函数功能描述

获取模块钱箱在位、可用状态，钱箱信息。

#### 6.8.2.1.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 钱箱号。当 uiBoxNo 为 0 时，则查询所有钱箱信息。

#### 6.8.2.1.4 输出参数说明

SZMC\_BND\_BOX\_INFO& info:

```
typedef struct _szmc_bnd_box_info
```

```
{
```

```
    UINT32 uiCount;
```

```
    LPSZMC_BND_BOXUNIT lpList;
```

```
} SZMC_BND_BOXINFO,*LPSZMC_BND_BOXINFO;
```

uiCount: 要查询的钱箱的返回数

LPSZMC\_BND\_BOXUNIT lpList:

```
typedef struct _szmc_bnd_box_unit
```

```
{
```

```
    UINT32 uiBoxNo;
```

```
    BOOL bExist;
```

```
    UINT32 uiNoteType;
```

```
    LPSTR lpzBoxID;
```

```
    UINT32 uiBoxNum;
```

```
    UINT32 uiBoxWillNum;
```

```
    UINT32 uiNoteNum;
```



UINT32 uiStatus;  
} SZMC\_BND\_BOX\_UNIT,\*LPSZMC\_BND\_BOX\_UNIT;  
bExist: 钱箱是否到位。只有到位，以下字段有效。bExist 为下列值之一：

值	含义
SZMC_BND_BOX_IN_PLACE	钱箱到位
SZMC_BND_BOX_DISLOCATION	钱箱离位

uiNoteDeno: 纸币类型。uiNoteDeno 为下列值之一：

值	含义
SZMC_BND_ONE_YUAN	1 元
SZMC_BND_FIVE_YUAN	5 元
SZMC_BND_TEN_YUAN	10 元
SZMC_BND_TWENTY_YUAN	20 元
SZMC_BND_FIFTY_YUAN	50 元
SZMC_BND_ONE_HUNDRED_YUAN	100 元

lpszBoxID: 钱箱 ID 号。

uiBoxNum: 钱箱空数量。

uiBoxWillNum: 钱箱将空数量。

uiNoteNum: 纸币数量。

uiStatus: 钱箱当前状态。uiStatus 为下列值之一：

值	含义
SZMC_BND_BOX_COMMON	钱箱正常
SZMC_BND_BOX_WILL_EMPTY	钱箱将空
SZMC_BND_BOX_EMPTY	钱箱空

6.8.2.1.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2.2 出钞

6.8.2.2.1 函数原型

INT32 SzmC\_BND\_Dispende(SZMCHANDLE hCom, SZMC\_DISPENSE\_INFO info,

SZMC\_DISPENSE\_RESULT& result)

6.8.2.2.2 函数功能描述

出钞。

6.8.2.2.3 输入参数说明

hCom: 模块句柄。

SZMC\_DISPENSE\_INFO info:

typedef struct \_szmc\_dispense\_info

```
{  
    UINT32 uiAmount;  
    UINT32 uiDispPlan;  
    LPSTR lpszPlanPara;  
} SZMC_DISPENSE_INFO,*LPSZMC_DISPENSE_INFO;
```

uiAmount: 出钞金额。

uiDispPlan: 出钞方案。uiDispPlan 为下列值之一:

值	含义
SZMC_BND_DISP_AUTO	自动出钞
SZMC_BND_MANUAL	人工出钞

lpszPlanPara: 出钞参数。

6.8.2.2.4 输出参数说明

SZMC\_DISPENSE\_RESULT& result:

typedef struct \_szmc\_dispense\_result

```
{  
    UINT32 uiAmount;  
    UINT32 uiBoxNum[MAX_BND_BOX_NUM];  
} SZMC_DISPENSE_RESULT,*LPSZMC_DISPENSE_RESULT;
```

uiAmount: 实际出钞金额。

uiBoxNum: 钱箱实际出钞数量。

6.8.2.2.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2.3 回收出钞口钞票

6.8.2.3.1 函数原型

INT32 Szmc\_BND\_Retract(SZMCHANDLE hCom)

6.8.2.3.2 函数功能描述

检测出钞嘴的钞票是否被取走，如果取走，则关闭出钞闸门，如果超时未被取走，则回收钞票到回收箱并关闭出钞闸门。

6.8.2.3.3 输入参数说明

hCom：模块句柄。

6.8.2.3.4 输出参数说明

无。

6.8.2.3.5 函数返回值说明

返回值可能为下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2.4 取走钱箱

6.8.2.4.1 函数原型

INT32 Szmc\_Note\_TakeBox(SZMCHANDLE hCom,UINT32 uiBoxNo)

6.8.2.4.2 函数功能描述

取走钱箱。

6.8.2.4.3 输入参数说明

hCom: 模块句柄。

uiBoxNo: 钱箱号。

6.8.2.4.4 输出参数说明

无。

6.8.2.4.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2.5 装回钱箱

6.8.2.5.1 函数原型

INT32 Szmc\_Note\_InstallBox(SZMCHANDLE hCom, SZMC\_INSTALL\_BOX\_INFO info, SZMC\_INSTALL\_BOX\_RESULT& result)

6.8.2.5.2 函数功能描述

装回钱箱。

6.8.2.5.3 输入参数说明

hCom: 模块句柄。

SZMC\_INSTALL\_BOX\_INFO info:

typedef struct \_szmc\_install\_box\_info

{

    UINT32 uiBoxNo;

    UINT32 uiCount;

} SZMC\_INSTALL\_BOX\_INFO, \*LPSZMC\_INSTALL\_BOX\_INFO;

uiBoxNo: 装回钱箱号。

uiCount: 添加数量。

6.8.2.5.4 输出参数说明

```
SZMC_INSTALL_BOX_RESULT& result:

typedef struct _szmc_install_box_result
{
    UINT32 uiBoxNum;

    LPSTR lpszBoxID;
} SZMC_INSTALL_BOX_RESULT,*LPSZMC_INSTALL_BOX_RESULT;

uiBoxNum: 钱箱数量。

lpszBoxID: 钱箱 ID。
```

6.8.2.5.5 函数返回值说明

返回值可能为下列值:

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

6.8.2.6 通用命令接口

6.8.2.6.1 函数原型

```
INT32 SzmC_BND_Execute(SZMCHANDLE hCom,UINT32 uiCmdNo, LPVOID lpIn,LPVOID lpOut)
```

6.8.2.6.2 函数功能描述

基于扩展性的要求，增加该命令接口。

6.8.2.6.3 输入参数说明

- hCom: 模块句柄。
- uiCmdNo: 命令号。
- lpIn: 输入参数字符串。

6.8.2.6.4 输出参数说明

- lpOut: 输出参数字符串。

6.8.2.6.5 函数返回值说明

返回值至少包含下列值：

值	含义
SZMC_OK	成功
SZMC_FAILED	命令执行失败

7 读写器接口技术规范

7.1 接口数据定义

API 参数中相关的接口数据定义如下表。

表 105 API 参数接口数据定义

接口数据项	数值	宏定义	含义
发卡方标识	01H	ISSUER_ACC	ACC
	02H	ISSUER_SZT	深圳通
	03H	ISSUER_LNT	岭南通
	04H	ISSUER_BDT	八达通
票卡物理类型	00H	CARD_ULTRALIGHT	Ultralight
	01H	CARD_S50	S50
	02H	CARD_S70	S70
	03H	CARD_DESFIRE	Desfire
	04H	CARD_ACPU	ACPU
	05H	CARD_SONonyCard	SonyCard
	06H	CARD_Ultralight C	Ultralight C
	07H	CARD_2.4G	2.4G
	0FH	CARD_UNKNOW	其他未知类型
文件类型	01H	FILE_LOG	日志文件
	02H	FILE_TRADE	交易文件
	03H	FILE_PARAMETRE	参数文件
	04H	FILE_SOFT	软件
	05H	FILE_BLACK	黑名单文件
卡状态	00H	CARDSTATUS_UN SOLD	未售：待出售卡
	03H	CARDSTATUS_SOLD	已售：正在使用中的卡

表 105 API 参数接口数据定义 (续 1)

	07H	CARDSTATUS_REBACK	回收: 已回收卡
	09H	CARDSTATUS_BLACK	黑名单锁定: 禁止交易
设备应用类型	01H	SLE_TIM	编码分拣机
	02H	SLE_TSM	分拣机
	03H	SLE_BOM	半自动售票机
	04H	SLE_AVM	自动增值机
	05H	SLE_TVM	自动售票机
	06H	SLE_TCM	验票机
	07H	SLE_INAGM	进站闸机
	08H	SLE_OUTAGM	出站闸机
	09H	SLE_IOAGM	双向闸机
	0AH	SLE_PCA	手持检票机
天线用途	00H	ENTENNA_SINGLE	设备业务唯一天线
	01H	ENTENNA_INGATE	进站票卡天线
	02H	ENTENNA_OUTGATE_NON_RECYCLE	出站非回收票卡天线
	03H	ENTENNA_OUTGATE_RECYCLE	出站回收票卡天线
钱包单位	01H	MONEY_UNIT_FEN	分
	02H	MONEY_UNIT_JIAO	角
	03H	MONEY_UNIT_YUAN	元
	04H	MONEY_UNIT_COUNT	次
	05H	MONEY_UNIT_HOUR	小时
	06H	MONEY_UNIT_DAY	天
	07H	MONEY_UNIT_MONTH	月
	08H	MONEY_UNIT_YEAR	年
AFC 业务类型	01H	CARD_TREAT_INIT	票卡初始化业务
	02H	CARD_TREAT_PRESELL	票卡预赋值业务
	03H	CARD_TREAT_CANCEL	票卡抵消业务
	04H	CARD_TREAT_SELL	票卡发售业务
	05H	CARD_TREAT_ADDVALUE	票卡充值业务
	06H	CARD_TREAT_ADDCANCEL	票卡冲正业务
	07H	CARD_TREAT_ACTIVE	票卡激活业务
	08H	CARD_TREAT_INGATE	票卡进站业务
	09H	CARD_TREAT_OUTGATE	票卡出站业务
	0AH	CARD_TREAT_PERSONAL	票卡个人化业务
	0BH	CARD_TREAT_CHECK	票卡验卡业务
	0CH	CARD_TREAT_NOENTER	票卡无进站处理业务
	0DH	CARD_TREAT_NOEXIT	票卡无出站处理业务
	0EH	CARD_TREAT_OVERTIME	票卡滞留超时处理业务
	0FH	CARD_TREAT_OVERRUN	票卡超程补卡业务
	10H	CARD_TREAT_RETURN	票卡退卡业务

表 105 API 参数接口数据定义（续 2）

	11H	CARD_TREAT _REPLACE	票卡换卡业务
	12H	CARD_TREAT_LOGOFF	票卡注销业务
	13H	CARD_TREAT_LOCK	票卡锁卡业务
	14H	CARD_TREAT_UNLOCK	票卡解锁业务
支付方式	00H	PAY_CASH	现金
	01H	PAY_BANKCARD	银行卡
	02H	PAY_SZTCARD	深圳通卡

## 7.2 函数定义

函数定义应包括函数声明、函数说明、函数参数说明、函数返回及函数调用举例。以下为读写器 API 接口函数范围及参数数据项说明。

### 7.2.1 初始化读写器通讯口

表 106 初始化读写器通讯口

函数名	初始化读写器通讯口		
函数功能			
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
端口号	In	int	取值为 1~32 时，表示串口 1~32；为 100 时，表示 USB 口通讯，此时波特率无效。
波特率	In	int	通讯波特率，例如 9600，115200，38400。
设备通讯句柄	Out	int	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

### 7.2.2 关闭读写器通讯口

表 107 关闭读写器通讯口

函数名	关闭读写器通讯口		
函数功能			
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
设备通讯句柄	In	int	在 WIN32 环境下设备通讯句柄，必须释放后才可以再次连接。
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

### 7.2.3 获取版本信息函数



表 108 获取版本信息函数

函数名	获取读写器文件版本信息		
函数功能	获得该读写器的软件和参数文件的版本信息（文件名含释义），用于 ECU 比对是否为最新版本。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
文件版本号长度	Out	int	
文件版本号	Out	char	包含读写器动态库版本号、读写器软件版本号、发卡方 1 版本号、发卡方 2 版本号…，各版本号之间以 0x0D0x0A 分隔。
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.4 日志开关函数

表 109 日志开关函数

函数名	设置日志		
函数功能	设置 API、读写器主控模块和各发卡方模块是否写日志及日志详细程度级别。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
API 日志	In	int	0: 关闭 1: 级别 1（最简单） 2: 级别 2（次简单） ... 9: 级别 9（最详细）
读写器日志	In	int	0: 关闭 1: 级别 1（最简单） 2: 级别 2（次简单） ... 9: 级别 9（最详细）
发卡方日志	In	char	发卡方 1-8 1: 级别 1（最简单） 2: 级别 2（次简单） ... 9: 级别 9（最详细）
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.5 上传文件函数

表 110 上传文件函数

函数名	上传读写器文件		
函数功能	实现读取读写器中的文件（比如日志文件、交易文件等）		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
文件来源标识	In	int	0: 读写器 1-8:发卡方 1-8 .....
文件类型	In	int	见接口数据定义
文件小类	In	int	用于区分同一来源中同一文件类型有不同的文件小类。 0: 无小类 其他: 根据各文件类型需要定义
文件压缩标识	In	int	0: 不压缩 1: ZIP 格式压缩 .....
存储文件路径	In	char	
文件名	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.6 更新文件函数

表 111 更新文件函数

函数名	更新读写器文件		
函数功能	根据文件名，更新读写器软件或参数等各类文件。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
文件目的地	In	int	0: 读写器 1-8:发卡方 1-8 .....
文件类型	In	int	见接口数据定义
文件压缩标识	In	int	0: 不压缩 1: ZIP 格式压缩 .....
文件名	In	char	
是否重启	In	int	更新文件后是否自动重启。 0: 否 1: 是
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.7 AFC 业务启动函数

表 112 AFC 业务启动函数

函数名	AFC 业务启动		
函数功能	读写器进行自检、启动 SAM 卡检查是否配套（调用发卡方启动函数）、检查参数文件数据项是否完整并载入业务应用所需参数，启动寄存器。如果读写器工作模式为主动方式，读写器开始工作。启动成功后才可以进行该业务读写器允许的票卡业务交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
终端机编号	In	char	表示该读写器服务的应用设备的终端机编号，也用于比较 SAM 中绑定的终端机编号是否一致（按发卡方）。
运营点编号	In	char	
操作员	In	char	
设备应用类型	In	int	设置读写器应用于该设备应用类型（也可放在读写器配置参数文件中则去除该项）。 1: INAGM 2: OUTAGM 3: IOAGM 4:TIM ...BOM AVM TVM TSM TCM （设备应用类型与使用发卡方票卡对照表，全）
寻卡工作模式	In	int	0: 被动 1: 主动
天线配置	In	char	8 个字符表示。 例如“11000000”表示有天线 1 天线 2。
天线用途配置	In	char	8 个字符表示。 见接口数据定义
日志文件容量	In	int	单个文件容量
读写器电子 ID	Out	char	返回读写器唯一的电子编码。
SAM 卡在位信息	Out	char	卡槽号+发卡方标识+ SAM 信息长度+SAM 信息+间隔符+卡槽号+发卡方标识+ SAM 信息长度+SAM 信息+间隔符…… SAM 信息一般包含：SAMID+类型+终端机编号+额度（可选），由各发卡方定义。 该数据项为空表示没有任何 SAM 在位。
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.8 状态查询函数

表 113 状态查询函数

函数名	状态查询		
函数功能	返回读写器当前状态。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
读写器状态	Out	int	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

7.2.9 AFC 业务关闭函数

表 114 AFC 业务关闭函数

函数名	AFC 业务关闭
函数功能	关闭读写器 AFC 业务。
函数参数说明（无）	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。

7.2.10 管理业务函数

表 115 管理业务函数

函数名	管理业务函数		
函数功能	实现各发卡方的管理业务功能，比如查询或重载额度。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
业务类型	In	int	1: 查询或重载额度 .....
发卡方标识	In	int	表示管理的对象。 见接口数据定义。
输入数据长度	In	int	
输入数据	In	char	
输出数据长度	Out	int	
输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

表 116 输出数据说明示例

标签 (Tag)	长度 (Length)	数据值(Value)			
		数据项	数据类型	数据长度	数据说明
1 (查询和重载 额度)	53	SAMID	char	20	
		重载结果	int	1	0: 重载成功 1: 不需重载 2: 重载失败
		当前额度	char	10	
		额度有效期	char	14	
		授权版本号	char	8	

7.2.11 同步时钟函数

表 117 同步时钟函数

函数名	同步读写器时钟		
函数功能	实现读写器时间同步。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
当前时间	In	char	按当前时间同步读写器的时间。
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.12 票价计算函数

表 118 票价计算函数

函数名	票价计算		
函数功能	用于自动售票机和半自动售票机进行售票时计算两个运营点之间的票价。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
卡类型	In	int	
子类型	In	int	
交易日期时间	In	char	用于优惠时段票价计算
计价起始运营点	In	char	
计价终止运营点	In	char	
票价	Out	int	两个运营点之间的计算出的票价。
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.13 验卡函数

表 119 验卡函数

函数名	AFC 业务验卡		
函数功能	获取各业务需要的验卡数据及结果。并检查黑名单，如发现是黑名单根据参数设置情况进行锁卡。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
读票卡历史记录数量	In	int	0：不读。>0 表示需要读的记录数，返回数量以票卡记录数量为准。
是否读个人信息	In	int	1：读，0：不读
是否读末笔充值交易	In	int	1：读，0：不读
票卡分析类型	In	int	0：票卡一般分析 1：含付费区分析 2：含非付费区分析

表 119 验卡函数（续 1）

函数名	AFC 业务验卡		
交易日期时间	In	char	黑名单锁卡交易用
设备交易序号	In	int	黑名单锁卡交易用
物理介质类型	Out	int	见接口数据定义
物理卡号	Out	char	
发卡方标识	Out	int	见接口数据定义
票卡分析结果	Out	int	0: 正常 1: 非本系统卡 2: 票卡校验码错 3: 付费区超时 4: 付费区超程 5: 付费区超时超程 6: 付费区无入站标识 7: 非付费区已入站标识 .....
卡号	Out	char	
卡类型	Out	int	
卡子类	Out	int	卡类型结合卡子类用于显示票种名称。
卡状态	Out	int	见接口数据定义
激活情况	Out	int	结合卡属性得到 0: 不需激活 1: 未激活使用（针对首次使用需激活的票卡） 2: 已激活使用（针对首次使用需激活的票卡） 3: 需定期激活（针对有定期激活属性设置且符合定期激活条件的票卡）
发卡方代码	Out	int	卡上数据
城市代码	Out	char	
卡应用标识	Out	int	0: 正常票, 1: 测试票, 其他预留
发行日期	Out	char	
发行批次	Out	char	
有效起始日期时间	Out	char	
有效结束日期时间	Out	char	
卡余额	Out	int	
押金	Out	int	
售票站点	Out	char	
售票金额	Out	int	
入站标志	Out	int	
交易日期时间	Out	char	
交易站点	Out	char	

表 119 验卡函数（续 2）

函数名	AFC 业务验卡		
可使用站点	Out	char	单个站点或区段编号
最近使用日期时间	Out	char	
最近使用地点	Out	char	
姓名	Out	char	
性别	Out	int	
证件号码	Out	char	
证件类型	Out	int	
末笔充值记录	Out	char	
票卡历史记录数量	Out	int	
票卡历史记录	Out	char	
交易记录数量	Out	byte	
交易类型 +交易记录长度 +交易记录数据	Out	char	
后续操作标识	In/Out	int	0: 无后续操作 1: 交易确认
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.14 询卡函数

表 120 询卡函数

函数名	询卡		
函数功能	票卡检测，并判断是否本系统适用票卡。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	Out	int	见接口数据定义
物理卡号	Out	char	
发卡方标识	Out	int	见接口数据定义
天线标识	Out	char	8 个字符表示。 例如“10000000”表示天线 1 寻到卡。
函数返回值	0 表示成功寻卡。其他值表示应用错误码，见错误码说明。		

## 7.2.15 检票业务函数

表 121 检票业务函数

函数名	检票业务		
函数功能	完成进出站检票交易。由天线用途配置决定是进站检票还是出站检票。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
交易日期时间	In	char	
设备交易序号	In	int	
卡类型	Out	int	用于 ECU 判断是否为优惠票种并进行声光提示及是否回收等操作。
子类型	Out	int	
钱包余额	Out	int	用于 ECU 显示钱包余额。
钱包单位	Out	int	用于 ECU 显示钱包余额的单位。 见接口数据定义
是否需要回收	Out	int	指导 ECU 是否进行票卡回收 0：不回收 1：回收
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	进站交易 + 进闸时自动激活的交易记录或进闸时补扣上次的出闸交易
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	



## 7.2.16 发售业务函数

表 122 发售业务函数

函数名	售票交易		
函数功能	实现票卡售票交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
交易日期时间	In	char	
设备交易序号	In	int	
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	
交易小类	In	int	
有效起始日期时间	In	char	
有效结束日期时间	In	char	
售票起始站	In	char	
售票目的站	In	char	
实收金额	In	int	表示实际收到的金额（如有优惠是优惠后金额）。
应收金额	In	int	表示应该收到的金额（优惠前金额）。
交易值	In	int	钱包类型为货币时，为货币应收金额； 钱包类型为次数时，为应充值次数；
支付方式	In	int	见接口数据定义
支付票卡卡号	In	char	
支付票卡消费序号	In	int	
钱包余额	Out	int	用于判断是否大于允许出站的最小余额。
钱包单位	Out	int	见接口数据定义
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.17 充值业务函数

表 123 充值业务函数

函数名	充值交易		
函数功能	实现票卡充值交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	
交易小类	In	int	
有效起始日期时间	In	char	
有效结束日期时间	In	char	
实收金额	In	int	表示实际收到的金额（如有优惠是优惠后金额）。
应收金额	In	int	表示应该收到的金额（优惠前金额）。
交易值	In	int	钱包类型为货币时，为货币应收金额； 钱包类型为次数时，为应充值次数；
售票起始站	In	char	
售票目的站	In	char	
支付方式	In	int	见接口数据定义
支付票卡卡号	In	char	
支付票卡消费序号	In	int	
交易日期时间	In	char	
设备交易序号	In	int	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.18 票卡处理业务函数

表 124 票卡处理业务函数

函数名	票卡处理		
函数功能	完成半自动售票机进行票卡处理，票务处理费用（费用参数由 BOM 读取）根据票务规定收取。该函数能处理的业务类型包括：票卡无入站处理业务、票卡无出站处理业务、票卡滞留超时处理业务、票卡超程补卡（原卡充值）业务、票卡退卡业务、票卡冲正业务、票卡激活业务、票卡锁卡业务、票卡解锁业务等。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
业务类型	In	int	见接口数据定义
交易小类	In	int	退卡等有交易小类
实收/退金额	In	int	
应收/退金额	In	int	
交易值	In	int	根据票卡类型确定为金额或次数；
进站站点	In	char	用于无入站处理
有效起始日期时间	In	char	用于激活
有效结束日期时间	In	char	用于激活
交易日期时间	In	char	
设备交易序号	In	int	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.19 换卡业务函数

表 125 换卡业务函数

函数名	换卡		
函数功能	根据原卡信息进行卡片更换。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	新卡物理卡号
旧卡卡号	In		应写在交易中用于后台审核
旧卡卡类型	In	int	旧卡类型
旧卡子类型	In	int	旧卡子类
旧卡有效起始日期时间	In	char	旧卡有效期
旧卡有效结束日期时间	In	char	旧卡有效期
旧卡卡交易序号	In	int	旧卡最后的卡交易序号，应写在交易中便于后台审核更换数据
旧卡卡余额/次	In	int	旧卡余额/次，新卡照此赋值或减少次数
旧卡激活状态	In	int	旧卡激活状态为已激活的，根据此项确定是否激活新卡。
旧卡本单位职工标识	In	int	
旧卡持卡人姓名	In	char	
旧卡持卡人证件号码	In	char	
旧卡持卡人证件类型	In	int	
旧卡持卡人性别	In	int	
个人密码(PIN)	In	char	
实收金额	In	int	
应收金额	In	int	
交易日期时间	In	char	
设备交易序号	In	int	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.20 个人化业务函数

表 126 个人化业务函数

函数名	票卡个人化		
函数功能	用于修改票卡个人信息。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	
本单位职工标识	In	int	
持卡人姓名	In	char	
持卡人证件号码	In	char	
持卡人证件类型	In	int	
持卡人性别	In	int	
个人密码(PIN)	In	char	
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.21 初始化业务函数

表 127 初始化业务函数

函数名	初始化		
函数功能	用于编码器进行票卡初始化交易交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
是否旧卡初始化	In	int	0: 1: 是
批次号	In	char	
发行商代码	In	int	
卡应用标识	In	int	
密钥版本	In	int	
卡应用版本	In	int	
卡号	In	int	

表 127 初始化业务函数（续）

函数名	初始化		
卡类型	In	int	
子类型		int	
发行批次	In	char	
钱包		char	
钱包单位	In	int	
有效起始日期时间	In	char	
有效结束日期时间	In	char	
可使用站点 1	In	char	
可使用站点 2	In	char	
激活标识	In	int	
交易日期时间	In	char	
设备交易序号	In	char	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0: 否 1: 是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.22 预赋值业务函数

表 128 预赋值业务函数

函数名	预赋值交易		
函数功能	用于编码机按批次进行预赋值交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
交易小类	In	int	
批次号	In	int	供读写器交易前比对，非同一批次返回错误
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	

表 128 预赋值业务函数（续）

函数名	预赋值交易		
函数功能	用于编码器按批次进行预赋值交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
有效起始日期时间	In	char	
有效结束日期时间	In	char	
售票起始站	In	char	可使用站点 1
售票目的站	In	char	可使用站点 2
交易值	In	int	票卡钱包预赋值
激活标志	In	int	
交易日期时间	In	char	
设备交易序号	In	int	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0: 否 1: 是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.23 抵消业务函数

表 129 抵消业务函数

函数名	预赋值抵消交易		
函数功能	用于编码器按批次进行预赋值和抵消交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
批次号	In	int	供读写器交易前比对，非同批次返回错误
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	
交易值	In	int	需抵消的钱包值
交易日期时间	In	char	

表 129 抵消业务函数（续）

函数名	预赋值抵消交易		
函数功能	用于编码器按批次进行预赋值和抵消交易。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
设备交易序号	In	int	
交易记录数量	Out	int	
交易类型 +交易记录长度 +交易记录数据	Out	char	
是否交易确认	In	int	0: 否 1: 是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.24 注消业务函数

表 130 注消业务函数

函数名	注消交易		
函数功能	用于编码器将票卡注销，回到未售状态。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	char	
卡类型	In	int	确认业务操作的卡类型
子类型	In	int	
是否交易确认	In	int	0：否 1：是
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

## 7.2.25 票卡物理数据读卡函数



表 131 票卡物理数据读卡函数

函数名	票卡物理数据读卡函数		
函数功能	用于票卡分析及测试。		
函数参数说明			
数据项	I/O 类型	数据类型	数据说明
物理介质类型	In	int	见接口数据定义
物理卡号	In	int	
读取范围	In	int	0：整张卡 1：发行信息区 2：售票信息区 3：交易应用区 4：轨道应用区 5：交易记录区 6：钱包区（CPU 卡） 7：个人信息区 8：充值信息区 .....
输出数据长度	Out	int	
输出票卡数据	Out	HEX	
预留输入数据长度	In	int	
预留输入数据	In	char	
预留输出数据长度	Out	Int	
预留输出数据	Out	char	
函数返回值	0 表示成功，其他值表示应用错误码，见错误码说明。		

7.3 函数返回值

描述各 API 函数的返回值, 如 0 代表函数执行成功, 其他数字代表错误返回值, 归类各错误返回值的描述。

表 132 错误返回值描述

结果代码 (BCD 字符)	结果代码描述
0x00	命令执行成功
0x01	天线有效区域无卡 (未检测到卡或寻卡超时)
0x02	天线有效区域有多张卡
0x03	SAM 卡命令执行返回错误的 SW

表 132 错误返回值描述 (续 1)

结果代码 (BCD 字符)	结果代码描述
0x04	通讯错误: 发送数据不成功, 请检查通讯端口是否正常
0x05	通讯错误: 无应答数据
0x06	数据错误
0x07	串口打开失败
0x08	当前操作的物理卡号不相符
0x09	卡操作流程不符合
0x0A	坏卡
0x0B	读写器返回的其他错误
0x0C	SAM 卡启动失败
0x0D	多张 Sam 卡终端机编号不一致
0x0E	票卡发行区已锁定
0x0F	运行所需要的 SAM 卡检查不通过
0x10	读写器主动寻卡方式下无数据
0x11	卡交易信息读回后比较失败
0x12	读卡失败
0x13	无效的 SAM 功能调用
0x14	SAM 卡类型不匹配
0x16	卡认证失败
0x17	写卡失败, 卡状况未知, 需要验卡重试保证卡完整性
0x18	卡状态不是黑名单, 不能解锁
0x19	块数据校验码错误
0x20	返回的数据范围有误
0x21	提供的写卡数据有误
0x22	充值额度不足
0x23	充值授权有效期已过
0x24	充值限额文件未找到
0x25	充值授权序列号错误
0x26	安全报文数据项不正确
0x27	应用已被锁定
0x28	金额不足
0x29	相互认证的 SAM 卡信息不符
0x2A	缺少相互认证的 SAM 卡
0x2B	SAM 卡认证不通过
0x33	没有找到文件
0x39	超过 SAM 卡有效期

表 132 错误返回值描述 (续 2)

结果代码 (BCD 字符)	结果代码描述
0x3A	SAM 状态不对
0x46	SAM 卡未激活, 不能执行此命令
0x50	在线下载读写器软件时发生的错误
0x63	单程票 MAC 验证错误
0x64	单程票发行 MAC 错误
0x80	网络通讯失败
0x81	网络通讯返回数据有误
0x85	TAC 校验错
0x99	不支持该功能

7.4 ECU 调用流程

7.4.1 ECU 启动读写器流程

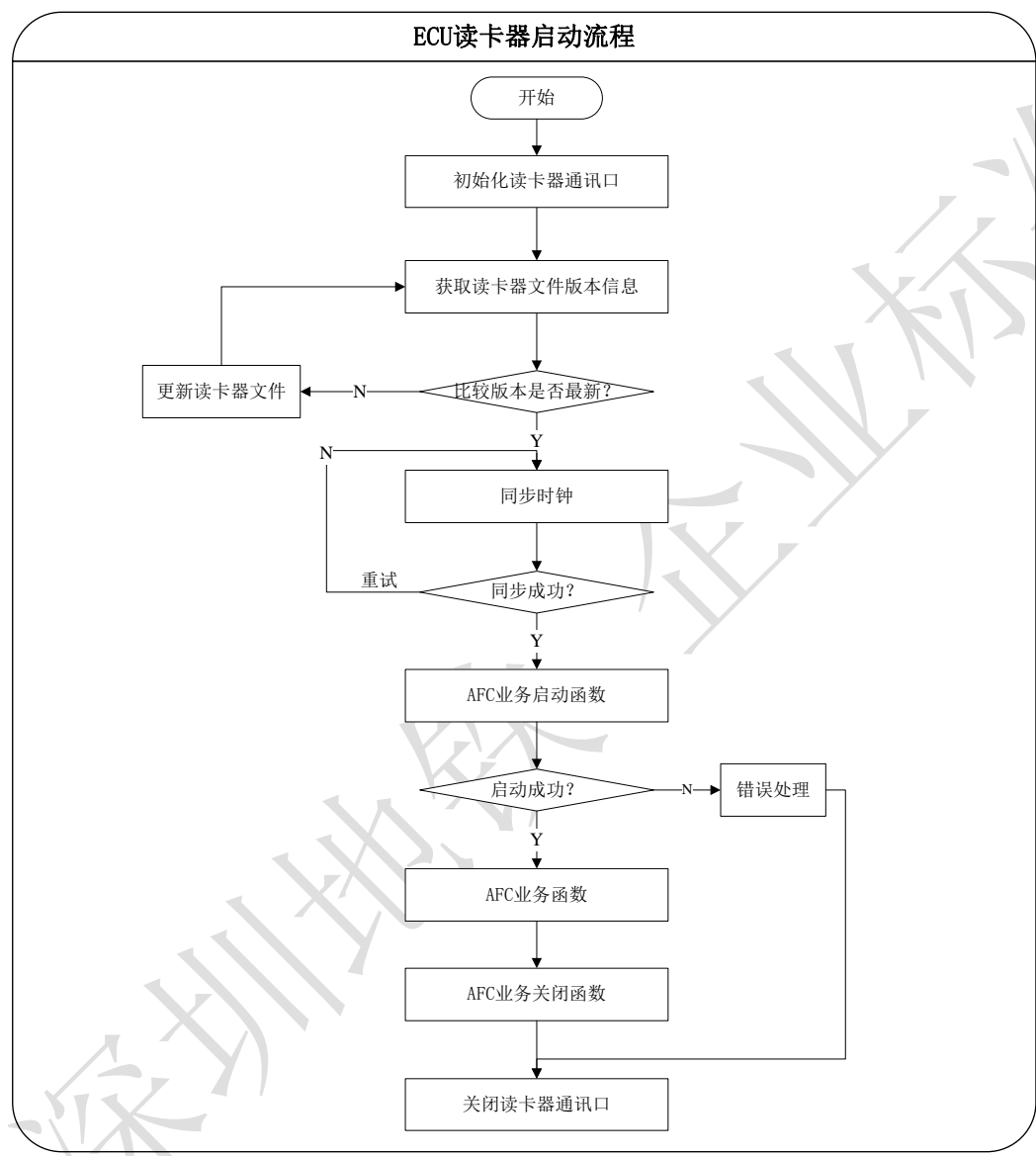


图 5 ECU 启动读写器流程

7.4.2 ECU 票卡业务流程

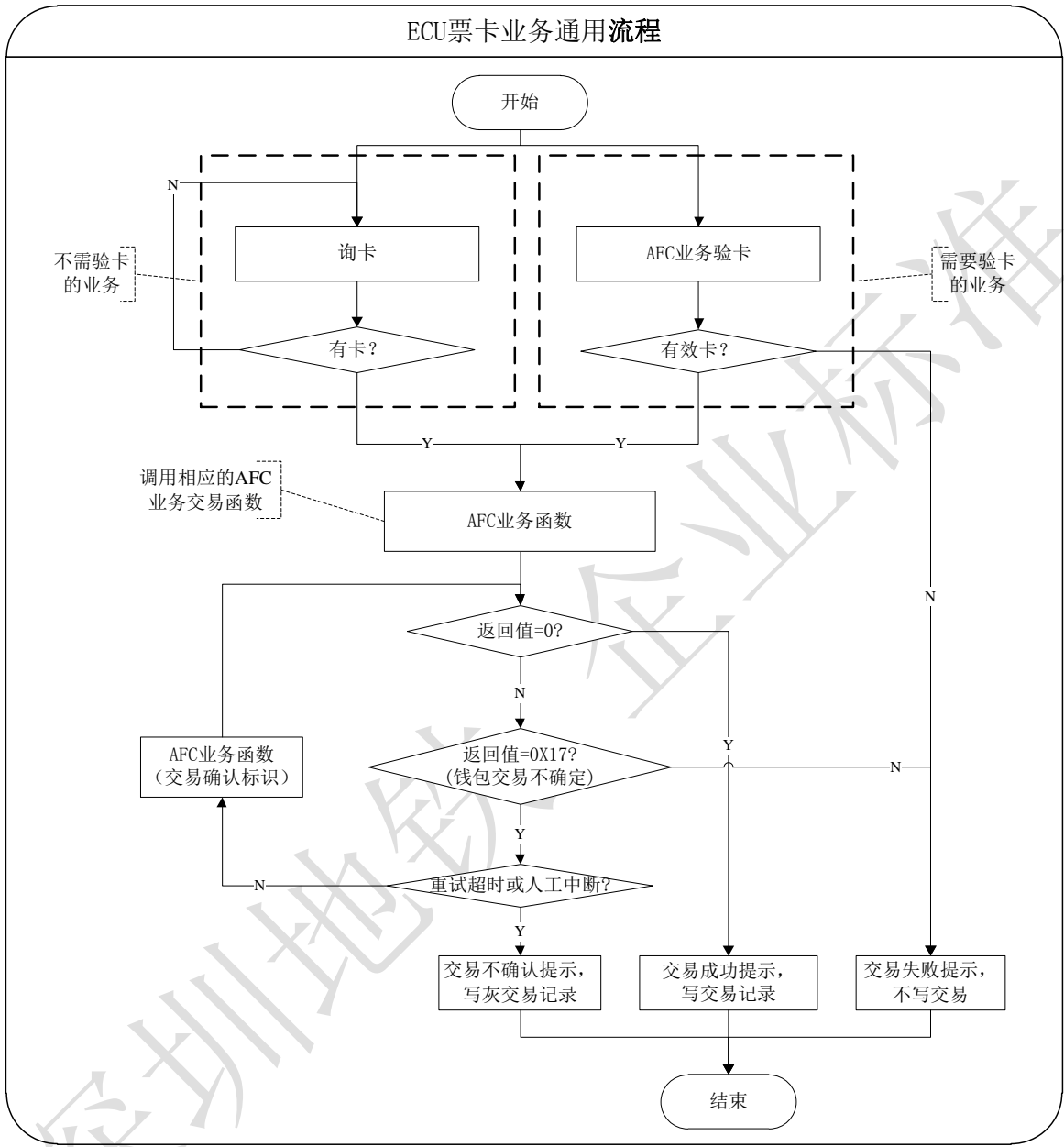


图 6 ECU 票卡业务流程

附录 A  
(资料性附录)  
设备部件接口业务与函数调用对应表

表 A.1 单程票发售机构业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开硬币模块	
2 退出服务	断开与模块的通信	关闭硬币模块	当业务软件退出, 或者要求关机等原因时, 调用该接口
3 记录日志开关	打开日志及关闭日志设置	设置参数	参数名: Log 参数值: 0: 关闭日志 1: 打开日志
	设置日志级别	设置参数	参数名: LogLevel 参数值: 1~5。 1: 日志级别最低 5: 日志级别最高
4 票箱将空数量	单程票数量低于票箱将空的数量时, 发送将空报警信息	设置参数	参数名: WillEmptyNum 参数值: 大于 0。
5 票箱空数量	硬币数量低于空数量时, 发送空报警信息。	设置参数	参数名: EmptyNum 参数值: 大于等于 0。 将空数量必须大于等于空数量。如果将空数量小于空数量, 则将空数量参数无效, 将空数量设置为空数量。
6 票箱单程票种类设置	设置所有票箱的单程票种类	设置参数	参数名: BoxTokenType 参数值: 由设置的单程票种类组成。各个种类间用逗号隔开。 示例: “0,1”, 票箱 1 设置为种类 “0”, 票箱 2 设置为种类 “1”。
7 自检	自检操作。	自检	
8 底层更新	更新底层软件版本	软件更新	
9 获取当前钱箱信息	包括钱箱 ID, 钱箱数量, 纸币面值等信息。	获取钱箱信息	
10 票箱数量设置	票箱数量设置	设置票箱数量	
11 出票到读写区	出票到读写区	出票到读写区	
12 出票到出票口	出票到出票口	出票到出票口	

表 A.1 单程票发售机构业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
13 回收票到回收箱	回收票到回收箱	回收票到回收箱	
14 回收票到废票箱	回收票到废票箱	回收票到废票箱	
15 盘点	盘点	盘点	
16 停止盘点	停止盘点	停止盘点	
17 上传审计数据	定时上传审计数据	获取寄存器值	

表 A.2 单程票回收模块业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开回收模块	
2 退出服务	断开与模块的通信	关闭回收模块	当业务软件退出，或者要求关机等原因时，调用该接口
3 记录日志开关	打开日志及关闭日志设置	设置参数	参数名：Log 参数值： 0：关闭日志 1：打开日志
	设置日志级别	设置参数	参数名：LogLevel 参数值：1~5 1：日志级别最低 5：日志级别最高
4 票箱将满数量	单程票数量大于票箱将满的数量时，发送将满报警信息	设置参数	参数名：WillFullNum 参数值：大于等于 0
5 票箱满数量	单程票数量大于满数量时，发送满报警信息	设置参数	参数名：FullNum 参数值：大于 0 将满数量必须小于等于满数量。如果将满数量大于满数量，则将满数量参数无效，将满数量设置为满数量
6 自检	自检操作	自检	
7 底层更新	更新底层软件版本	软件更新	
8 获取当前票箱信息	包括票箱 ID, 票箱数量, 单程票种类等信息	获取钱箱信息	
9 获取传感器状态	获取所有传感器状态	获取回收模块传感器状态	
10 回收	回收单程票	回收	

表 A.2 单程票回收模块业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
11 退票	退出单程票	退票	
12 切换票箱	在票箱间进行切换	切换票箱	
13 上传审计数据	定时上传审计数据	获取寄存器值	

表 A.3 机芯业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开模块	
2 退出服务	断开与模块的通信	关闭模块	当业务软件退出，或者要求关机等原因时，调用该接口
3 记录日志开关	打开日志及关闭日志设置	设置参数	参数名：Log 参数值： 0：关闭日志 1：打开日志
	设置日志级别	设置参数	参数名：LogLevel 参数值：1~5 1：日志级别最低 5：日志级别最高
4 通行模式	进站方向	设置参数	参数名：ModeIn 参数值： 0：禁止通行 1：自由通行 2：受控通行（缺省值）
	出站方向	设置参数	参数名：ModeOut 参数值： 正整数
5 最大通行人数	进站方向	设置参数	参数名：PMaxMemo 参数值： 正整数
6 工作模式	工作模式	设置参数	参数名：WorkMode 参数值： 0：常开 1：常闭
7 通过通道的超时时间	超过该时间不通过通道，则设备会自动关闭	设置参数	参数名：T0Impegno 参数值： 正整数



表 A.3 机芯业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
8 是否允许通道内刷卡	设置是否可以在通道内刷卡	设置参数	参数名: T0Impegno 参数值: 0: 不允许 1: 允许
9 通过后延时关门的时间	通过后关闭闸门的时间	设置参数	参数名: TRitChiudi 参数值: 正整数
10 自检	自检操作	自检	
11 底层更新	更新底层软件版本	软件更新	
12 显示传感器的状态	显示所有传感器的状态	获取模块传感器状态	
13 开门	包括进站方向和出站方向开门	开门	
14 进入紧急	进入紧急	紧急控制命令	
15 退出紧急	退出紧急	紧急控制命令	
16 进入维护	进入维护	维护控制命令	
17 退出维护	退出维护	维护控制命令	
18 测试控制	包括开始测试和停止测试	测试控制命令	

表 A.4 硬币模块业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开模块	
2 退出服务	断开与模块的通信	关闭模块	当业务软件退出, 或者要求关机等原因时, 调用该接口
3 记录日志开关	打开日志及关闭日志设置	设置参数	参数名: Log 参数值: 0: 关闭日志 1: 打开日志
	设置日志级别	设置参数	参数名: LogLevel 参数值: 1~5。 1: 日志级别最低 5: 日志级别最高
4 最多一次找零数量	一次最多能找硬币的数量	设置参数	参数名: MaxOnceChangeNum 参数值: 大于等于 0。等于 0 时则不找零。

表 A.4 硬币模块业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
5 钱箱将空数量	硬币数量低于钱箱将空的数量时，发送将空报警信息	设置参数	参数名：WillEmptyNum 参数值：大于 0。
6 钱箱空数量	硬币数量低于空数量时，发送空报警信息。	设置参数	参数名：EmptyNum 参数值：大于等于 0。 将空数量必须大于等于空数量。如果将空数量小于空数量，则将空数量参数无效，将空数量设置为空数量。
7 可接收的硬币面值	设置硬币模块可以接收的硬币面额	设置参数	参数名：AvailCoins 参数值：由可接收的硬币面额组成，各个面额间用逗号分开。未列出的面额则不接收。硬币面额的单位为分。 示例：“50,100”，表示硬币模块只接收 5 角，1 元。
8 自检	自检操作。	自检	
9 底层更新	更新底层软件版本	软件更新	
10 获取当前钱箱信息	包括钱箱 ID, 钱箱数量, 硬币面值等信息。	获取钱箱信息	
11 开始接收硬币	开始接收硬币	开始接收硬币	
12 停止接收硬币	停止接收硬币	停止接收硬币	
13 获取接收到的硬币信息	获取接收到的硬币的相关信息	获取接收到的硬币信息	
14 硬币找零	找硬币给乘客	找零	
15 退币	退硬币给乘客	退币	
16 盘点	硬币盘点	盘点	
17 停止盘点	停止硬币盘点	停止盘点	
18 回收硬币到回收箱	回收硬币到回收箱	回收	
19 上传审计数据	定时上传审计数据	获取寄存器值	

表 A. 5 纸币接收模块业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开模块	
2 退出服务	断开与模块的通信	关闭模块	当业务软件退出，或者要求关机等原因时，调用该接口
3 记录日志开关	打开日志及关闭日志设置	设置参数	参数名: Log 参数值: 0: 关闭日志 1: 打开日志
	设置日志级别	设置参数	参数名: LogLevel 参数值: 1~5。 1: 日志级别最低 5: 日志级别最高
4 钱箱将满数量	纸币数量大于钱箱将满的数量时，发送将满报警信息	设置参数	参数名: WillFullNum 参数值: 大于等于 0。
5 钱箱满数量	纸币数量大于满数量时，发送满报警信息。	设置参数	参数名: FullNum 参数值: 大于 0。 将满数量必须小于等于满数量。如果将满数量大于满数量，则将满数量参数无效，将满数量设置为满数量。
6 可接收的纸币面值	设置纸币模块可以接收的纸币面额	设置参数	参数名: AvailNotesDeno 参数值: 由可接收的纸币面额组成，各个面额间用逗号分开。未列出的面额则不接收。纸币面额的单位为分。 示例: “500, 1000”，表示纸币模块只接收 5 元，10 元。
7 缓存满时的处理	当纸币超过缓存最大张数时该如何处理	设置参数	参数名: BufFullOpMode 参数值: 0: 停止入币等待人工操作 1: 纸币入钱箱
8 纸币模块缓存最大张数	设置纸币模块一次最多能接收的纸币数量	设置参数	参数名: MaxBufNum 参数值: 大于等于 0。 当缓存中的纸币超过该参数值时，按照“缓存满时的处理”进行处理。
9 自检	自检操作。	自检	
10 底层更新	更新底层软件版本	软件更新	

表 A.5 纸币接收模块业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
11 获取当前钱箱信息	包括钱箱 ID, 钱箱数量, 纸币面值等信息。	获取钱箱信息	
12 开始接收纸币	开始接收纸币, 纸币指示灯亮	开始接收纸币	
13 获取入币信息	获取乘客的入币信息	获取接收到的纸币信息。	
14 停止接收纸币	停止接收纸币, 纸币指示灯灭	停止接收纸币	
15 压钱箱	纸币入钱箱	压钱箱	
16 退币给乘客	退币给乘客	退币	
17 更换钱箱	更换钱箱	更换钱箱	
18 上传审计数据	定时上传审计数据	获取寄存器值	

表 A.6 纸币找零模块业务与函数调用对应表

业务名称	业务内容	函数调用	说明
1 初始化	初始化模块	打开模块	
2 退出服务	断开与模块的通信	关闭模块	当业务软件退出, 或者要求关机等原因时, 调用该接口
3 纸币找零	找零给乘客	出钞	
4 记录日志开关	打开日志及关闭日志设置	设置参数	参数名: Log 参数值: 0: 关闭日志 1: 打开日志
	设置日志级别	设置参数	参数名: LogLevel 参数值: 1~5 1: 日志级别最低 5: 日志级别最高
5 钱箱将空数量	纸币数量低于钱箱将空的数量的时, 发送将空报警信息	设置参数	参数名: WillEmptyNum 参数值: 大于 0
6 钱箱空数量	纸币数量低于空数量时, 发送空报警信息	设置参数	参数名: EmptyNum 参数值: 大于等于 0 将空数量必须大于等于空数量。如果将空数量小于空数量, 则将空数量参数无效, 将空数量设置为空数量

表 A.6 纸币找零模块业务与函数调用对应表（续）

业务名称	业务内容	函数调用	说明
7 纸币找零失败最大重试次数	一次纸币找零过程中，最大的纸币找零重试次数。当超过此参数时，则发送告警信息，并停止此次找零	设置参数	参数名：MaxRetryTimes 参数值：大于等于 1。当等于 1 时，则不重试
8 一次最大回收数量	一次纸币找零过程中，最大的纸币回收数量。当超过此参数时，则发送告警信息，并停止此次找零	设置参数	参数名：MaxRecycledNum 参数值：大于等于 1
9 钱箱面额设置	设置每个钱箱的面额	设置参数	参数名：BoxNotesDeno 参数值：由设置的纸币面额组成，各个面额间用逗号分开。如果某个钱箱无需设置则为空。纸币面额的单位为分 示例：“500,1000”，假设一共有 4 个钱箱，则设置钱箱 1 为 5 元，钱箱 2 不设置，设置钱箱 3 为 10 元，钱箱 4 不设置
10 自检	自检操作	自检	
11 底层更新	更新底层软件版本	软件更新	
12 获取当前钱箱信息	包括钱箱 ID，钱箱数量，纸币面值等信息	获取钱箱信息	
13 回收出钞口的纸币	回收出钞口未被取走的钞票。需要模块支持	回收出钞口钞票	
14 更换钱箱	先取走钱箱，让后装回钱箱	取走钱箱→装回钱箱	
15 上传审计数据	定时上传审计数据	获取寄存器值	