

ICS 35.240.60
L 67

DB11

北京市地方标准

DB11/T XXXX
代替:DB11/T 1164.7—2017

轨道交通联网收费系统技术要求

第7部分:终端设备

Network toll system of rail transit technical requirements—
Part 7 : Terminal equipment

(征求意见稿)

XXXX—XX—XX 发布

XXXX—XX—XX 实施

北京市市场监督管理局 发布

前 言

本文件根据 GB/T 1.1 给出的规则起草。

DB11/T 1164《轨道交通联网收费系统技术要求》分为9个部分：

- 第1部分：系统结构及功能；
- 第2部分：接口数据格式；
- 第3部分：数据传输；
- 第4部分：操作界面；
- 第5部分：车票处理单元；
- 第6部分：票卡；
- 第7部分：终端设备；
- 第8部分：检测；
- 第9部分：技术指标体系。

本文件与 DB11/T 1164.7—2017 相比，除编辑性修改外，主要技术变化如下：

- 修改了规范性引用文件（见 2, 2017 版的 2 章节）；
- 修改了术语与定义（见 3, 2017 版的 3 章节）；
- 修改了缩略语（见 4, 2017 版的 4 章节）；
- 修改了终端设备结构要求（见 5.1.2.1, 2017 版的 5.1.2.1 章节）；
- 修改了材料要求（见 5.1.3, 2017 版的 5.1.3 章节）；
- 修改了静电放电抗扰度要求（见 5.4.3, 2017 版的 5.4.3 章节）；
- 修改了噪声要求（见 5.5, 2017 版的 5.5 章节）；
- 修改了可靠性要求（见 5.11, 2017 版的 5.10 章节）；
- 删除了硬币处理模块一般要求（见 2017 版 6.2.1 章节）；
- 修改了硬币处理模块硬件配置要求（见 6.2.2.1, 2017 版的 6.2.2.1 章节）；
- 修改了硬币处理模块功能要求（见 6.2.3, 2017 版的 6.2.3 章节）；
- 删除了纸币接收模块一般要求（见 2017 版 6.3.1 章节）；
- 修改了纸币接收模块功能要求（见 6.3.3, 2017 版的 6.3.3 章节）；
- 删除了纸币找零模块一般要求（见 2017 版 6.4.1 章节）；
- 修改了纸币找零模块硬件配置要求（见 6.4.2.1, 2017 版的 6.4.2.1 章节）；
- 修改了纸币找零模块功能要求（见 6.4.3, 2017 版的 6.4.3 章节）；
- 删除了车票发售模块一般要求（见 2017 版 6.5.1 章节）；
- 修改了车票发售模块硬件配置要求（见 6.5.2.1, 2017 版的 6.5.2.1 章节）；
- 修改了车票发售模块功能要求（见 6.5.3, 2017 版的 6.5.3 章节）；
- 修改了车票发售模块性能要求（见 6.5.4, 2017 版的 6.5.4 章节）；
- 删除了车票回收模块一般要求（见 2017 版 6.6.1 章节）；
- 修改了车票回收模块硬件配置要求（见 6.6.2.1, 2017 版的 6.6.2.1 章节）；
- 修改了车票回收模块功能要求（见 6.6.3, 2017 版的 6.6.3 章节）；
- 删除了闸门及通行控制装置一般要求（见 2017 版 6.6.1 章节）；
- 修改了闸门及通行控制装置硬件配置要求（见 6.6.2.1, 2017 版的 6.6.2.1 章节）；
- 修改了闸门及通行控制装置功能要求（见 6.6.3, 2017 版的 6.6.3 章节）；

- 增加了二维码模块内容（见第 6.8）；
- 删除了自动售票机一般要求（见 2017 版 7.1 章节）；
- 修改了自动售票机硬件选配要求（见 7.2.1，2017 版的 7.3.1 章节）；
- 修改了自动售票机软件配置要求（见 7.2.3，2017 版的 7.3.3 章节）；
- 修改了自动售票机性能要求（见 7.4，2017 版的 7.5 章节）；
- 删除了半自动售票机一般要求（见 2017 版 8.1 章节）；
- 修改了半自动售票机硬件选配要求（见 8.2.1，2017 版的 8.3.1 章节）；
- 修改了半自动售票机软件配置要求（见 8.2.3，2017 版的 8.3.3 章节）；
- 删除了自动检票机一般要求（见 2017 版 9.1 章节）；
- 修改了自动检票机硬件选配要求（见 9.2.1，2017 版的 9.3.1 章节）；
- 修改了自动检票机软件配置要求（见 9.2.3，2017 版的 9.3.3 章节）；
- 修改了自动检票机性能要求（见 9.4，2017 版的 9.5 章节）；
- 修改了接口设计要求（见 10，2017 版的 10 章节）；
- 增加了参考文献。

本文件由北京市交通委员会提出并归口。

本文件由北京市交通委员会组织实施。

本文件起草单位：北京市轨道交通指挥中心。

本文件主要起草人：

本文件历次版本发布情况为：

——DB11/T 1164.7—2017。

本次为第 1 次修订。

目 次

前言	I
1 范围	1
2 规范性引用文件	1
3 术语与定义	1
4 缩略语	3
5 基本要求	3
5.1 外观与结构	3
5.2 安装要求	5
5.3 电气性能	6
5.4 电磁兼容性	7
5.5 噪声	7
5.6 防尘	8
5.7 防水	8
5.8 机械环境适应性	8
5.9 温湿环境适应性	9
5.10 光照	9
5.11 可靠性	9
5.12 可维护性	9
5.13 安全性	9
6 专用模块	10
6.1 票箱	10
6.2 硬币处理模块	10
6.3 纸币接收模块	11
6.4 纸币找零模块	13
6.5 车票发售模块	14
6.6 车票回收模块	15
6.7 闸门及通行控制装置	16
6.8 二维码模块	18
6.9 电源模块	19
6.10 主控单元	20
7 自动售票机	21
7.1 外形和结构要求	21
7.2 配置要求	21
7.3 功能要求	22
7.4 性能要求	22
8 半自动售票机	22
8.1 外形和结构要求	22
8.2 配置要求	23

8.3 功能要求	23
8.4 性能要求	23
9 自动检票机	23
9.1 外形和材料要求	23
9.2 配置要求	24
9.3 功能要求	25
9.4 性能要求	25
10 接口设计要求	25
10.1 纸币接收模块接口	25
10.2 纸币找零模块接口	32
10.3 闸门及通行控制装置模块接口	41
10.4 车票发售硬件驱动接口 (TIM)	49
10.5 车票回收模块驱动接口 (TAM)	67
10.6 硬币处理模块驱动接口 (CHA)	87
10.7 DIO 控制装置模块接口	99
10.8 紧急按钮接口	105
10.9 LED 模块接口	107
10.10 打印机模块接口	111
10.11 后台维护终端模块接口	114
附 录 A (规范性附录) 票箱设计要求	120
A.1 范围	120
A.2 设计图纸	123
A.2.1 票箱装配 1	123
A.2.2 票箱装配 2	123
A.2.3 票箱外框架焊件	125
A.2.4 票箱外框架	125
A.2.5 竖把手转轴	127
A.2.6 票箱前盖	127
A.2.7 票箱内框架焊件	129
A.2.8 票箱内框架	129
A.2.9 内框架加强条	131
A.2.10 内框架加强条 2	132
A.2.11 定位条	133
A.2.12 安全票箱盖导轨 (右)	133
A.2.13 安全票箱盖导轨 (左)	135
A.2.14 票箱加强筋	135
A.2.15 锁孔垫片	137
A.2.16 票箱锁杆限位架	137
A.2.17 票箱口封条围框	139
A.2.18 票箱装配 (票箱滑盖)	139
A.2.19 票箱滑盖前盖	141
A.2.20 票箱滑盖后盖	141
A.2.21 滑盖后盖限位架	143

A. 2. 22 滑盖前盖限位架	143
A. 2. 23 ID 卡塑胶罩.....	145
A. 2. 24 ID 卡塑胶盖.....	145
A. 2. 25 ID 卡塑胶件固定板.....	147
A. 2. 26 票箱盖限位片	147
A. 2. 27 竖把手组件	149
A. 2. 28 竖把手支架（新）	150
A. 2. 29 竖把手	151
A. 2. 30 轴套	152
A. 2. 31 托盘装配	153
A. 2. 32 前托盘焊件	153
A. 2. 33 托盘	155
A. 2. 34 小轴承固定杆	155
A. 2. 35 托盘拉簧固定杆	157
A. 2. 36 自锁钩固定架焊件	157
A. 2. 37 自锁钩固定架	159
A. 2. 38 托盘连接螺柱	159
A. 2. 39 自锁钩固定杆 2.....	161
A. 2. 40 托盘自锁钩	161
A. 2. 41 支撑板	163
A. 2. 42 托盘拉簧	163
A. 2. 43 锁杆	165
A. 2. 44 票箱锁凸轮片	165
A. 2. 45 票箱口垫块	167
A. 2. 46 票箱锁压簧	167
A. 3 锁要求	169
A. 3. 1 票箱锁品牌及型号	169
A. 3. 2 票箱锁尺寸要求	169
A. 4 车票发售/回收模块要求	169
A. 5 票箱电子标签相关要求	170
A. 5. 1 芯片具体要求	170
A. 5. 2 Mifare Ultralight C 芯片存储的数据结构.....	171
A. 5. 3 Mifare One (S50) 芯片存储的数据结构	172
A. 5. 4 终端设备关键模块处理要求	173
附 录 B （规范性附录） 底座设计要求	174
B. 1 范围	174
B. 2 详细图纸	175
B. 2. 1 200mmAG 底座.....	175
B. 2. 2 250mmAG 底座.....	176
B. 2. 3 300mmAG 底座.....	177
B. 2. 4 TVM 底座.....	177

轨道交通联网收费系统技术要求

第7部分：终端设备

1 范围

本文件规定了城市轨道交通自动售检票系统终端设备、模块及接口的技术要求。
本文件适用于轨道交通自动售检票系统的前期规划、工程建设、更新、改造和运营管理。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件。不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 191 包装储运图示标志
GB/T 2423 电工电子产品环境试验
GB 4943 信息技术设备的安全
GB/T 6107 使用串行二进制数据交换的数据终端设备和数据电路终接设备之间的接口
GB 9254 信息技术设备的无线电骚扰限值和测量方法
GB 16999 人民币鉴别仪通用技术条件
GB/T 17183 数据终端设备和数据电路终接设备用的高速25插针接口暨可替换的26插针连接器
GB/T 17618 信息技术设备抗扰度限值和测量方法
SJ/T 11363 电子信息产品中有毒有害物质的限量要求
DB11/T 1164.1 轨道交通联网收费系统技术要求 第1部分：系统结构及功能
DB11/T 1164.9 轨道交通联网收费系统技术要求 第9部分：技术指标体系

3 术语和定义

下列术语和定义适用于本文件。

3.1

付费区 paid area

车站内各自动检票机与护栏合围形成的封闭区域。

3.2

非付费区 un-paid area

付费区以外的区域。

3.3

车站终端设备 station level equipment

用于各轨道交通线路车站进行车票发售、进出站检票、充值，验票分析等读写交易处理的终端设备。

3.4

硬币处理模块 coin handling module

具备识别各种不同规格或面额的硬币和硬币找零功能的装置。

3.5

纸币接收模块 bill receiving module

具备接收和识别纸币功能的装置。

3.6

纸币找零模块 bill dispense module

具备多面额纸币自动找零功能的装置。

3.7

车票发售模块 ticket issue module

具备车票出票、清票等功能的装置。

3.8

车票回收模块 ticket accept module

具备回收车票功能的装置。

3.9

闸门及通行控制装置 gate control module

具备识别不同模式下的乘客行为并做出准确响应功能的装置。

3.10

二维码模块 QR code module

具备识别、读取二维码功能的装置。

3.11

车站计算机系统 station computer

管理车站的票务、运行、客流统计等的计算机系统。

3.12

多线路管理中心 multiple line center

用于管理多条线路自动售检票系统的计算机系统。

3.13

锁死力 lock force

当扇门关闭时，若承受的冲击力超过一定限度，扇门应能自动开启。

3.14

最大动态冲击力 the maximum dynamic impact

当传感器被屏蔽时，活动扇门对人体发生撞击时的最大冲击力。

4 缩略语

下列缩略语适用于本文件。

AFC: 自动售检票系统 (Automatic Fare Collection)
AG: 自动检票机 (Automatic Gate)
BOM: 半自动售票机 (Booking Office Machine)
MCBF: 平均无故障次数 (Mean Cycles Between Failure)
MLC: 多线路管理中心 (Multiple Line Center)
SC: 车站计算机系统 (Station Computer)
SLE: 车站终端设备 (Station Level Equipment)
TVM: 自动售票机 (Ticket Vending Machine)

5 基本要求

5.1 外观与结构

5.1.1 外观

终端设备与专用模块外观应满足如下要求:

- 表面平滑, 边角圆滑, 间隙均匀且不超过板材厚度, 无明显的焊接痕迹、尖锐棱角、机械损伤、凹痕、划伤、裂痕、变形和污染等, 无外露的螺栓头、螺帽、毛边或裂纹;
- 整体不得有锈蚀、霉变或剥落。钣金件切口可能存在的锈蚀不会向切口外蔓延;
- 零部件应紧固, 不松动;
- 表面处理工艺条纹、纹路、颜色等均匀一致;
- 说明文字、符号应清晰、端正, 粘贴牢靠, 印字清晰、正确、可辨识, 不应有颠倒的情况, 内容应符合相关国家法规规定。

5.1.2 结构

5.1.2.1 终端设备

终端设备结构应满足如下要求:

- 结构设计满足人体工程学的要求, 采用模块化集成设计, 模块布局合理, 方便维修维护, 所有模块部件固定牢固;
- 设备内部、机壳、维修门、安全门、结构和零部件等人能够接触到的边缘无毛刺、尖角, 无 90° 棱角, 用手触摸感觉圆润;
- 所有喷漆(塑)零件的表面光滑平整、色泽一致, 不应有划痕、斑疵、流挂、脱落和破损。电镀零件的表面有金属光泽, 不应有裂纹、斑点、毛刺和缺陷;
- 维修门与机壳其它部位之间缝隙均匀, 不超过板材厚度;
- 维修门开关灵活, 无异响、无擦碰, 对设备外观不能造成影响;
- 维修门门锁完好, 维修门在锁闭状态下不晃动、不松动, 维修门锁上锁、解锁顺畅不卡壳;
- 人体检测传感器的保护镜片应经过防静电处理, 与传感器表面间隙有防尘封堵措施;
- 应有防尘、防水、防静电、防鼠设计;
- 除满足以上结构要求外, 终端设备应满足表 1 的要求。

表 1 终端设备结构要求

终端设备	结构要求
自动售票机	维修门应能开启保持 120 度或以上，并有限位功能，方便维护。
	应具备前面板的维护方式，并有锁止保护装置。
	维修状态下，维修门开启不影响相邻设备的正常使用。
	显示屏及主控模块的检修能通过机器前面板维护。
	前方检修状态，不影响相邻设备正常使用。
	维修照明灯照明范围应覆盖设备各维修区域，在检修门关闭后，自动切断维修照明电源。
	现金模块只能由授权人员进行操作。
半自动售票机	电源箱包括工控机与电源模块。
	发售模块应与电源箱采用分离式结构设计。
自动检票机	维修门应能开启并保持 120 度或以上，或具备可直接摘下的功能。维修门开启后不能影响乘客使用其它通道，包括带回收模块的自动检票机。
	端盖应有限位功能。
	乘客显示面板与外壳的配合处应有防水措施。
	顶部报警灯罩与顶盖之间密封良好。

5.1.2.2 专用模块

专用模块结构应满足如下要求：

- 紧固件：紧固件应有防振措施，无松动或脱落；
- 安装和焊接：导线的排列应整齐，元器件的排列安排应合理，连接件应有防脱措施；焊接中应无虚焊、漏焊现象；
- 防腐措施：线路板及各焊点应有防腐措施；金属零部件的镀层应无脱落或锈蚀现象；
- 设计要求：现场维护时，便于票箱的拆装、钱箱的更换，以及清除卡币，清除车票等故障的排除。

5.1.3 材料

终端设备及专用模块内包含的塑料部分应采用高强度、无毒材料制作，终端设备及专用模块中有害物质限量应符合SJ/T 11363的要求。

a) 终端设备材料要求如下：

- 设备的外壳材质的强度、硬度、抗氧化、耐高温、耐腐蚀等指标应不低于 304L 型号不锈钢。在成型、焊接、表面抛光及其他所需相关材料的制作及抛光方面符合 ASTM A 480/A 480Ma 标准。在需要焊接的场合应不能产生晶间腐蚀（焊接侵蚀）。在满足设备整体刚度及强度的条件下，设备支撑部件（如槽钢、L 型钢等）强度、硬度应不低于厚度为 2.5 mm 的不锈钢；外壳上的非受力部件强度、硬度应不低于厚度为 2.0 mm 的不锈钢；
- 设备内部各模块支架采用不锈钢材质；
- 拖链式线槽应方便线缆更换，拖链内部应至少预留 30%空间；
- 模块导轨的承重规格应为负载总重量的 1.5~2 倍；
- 确保模块整体能够拉出设备，便于维护维修；
- 电源线和信号电缆上的磁环、电源端骚扰电压、辐射骚扰满足 CCC 认证 B 级要求；

- 7) 电路板应不能产生霉变、腐蚀等现象。
- b) 专用模块材料要求如下：
 - 1) 硬币处理模块的主体结构宜采用不锈钢材质；
 - 2) 纸币接收模块、纸币找零模块、硬币处理模块的钱箱可采用不锈钢或工程塑料；
 - 3) 车票发售模块、车票回收模块的传输通道结构应采用不锈钢材质；
 - 4) 闸门及通行控制装置的主体结构宜采用不锈钢材质。

5.1.4 终端设备的标志

5.1.4.1 产品标志

产品应具有相应的中文标志与提示，并在适当的位置设置产品铭牌。内容应包括：产品名称、型号、产品标准编号、制造者名称、地址、出厂日期、商标、额定电压、额定频率、额定输入功率、防水等级的IP代码等项，其标志应简明、清晰、端正和牢固。

5.1.4.2 包装标志

包装箱外应采用统一标志，标明产品名称、型号、制造者名称、出厂日期、毛重和包装箱尺寸，并印刷或贴有“易碎物品”、“怕雨”、“向上”、“禁止翻滚”等运输标志，警示安全标志应醒目，符合GB/T 191的规定。包装箱外喷刷或粘贴的标志不应因运输条件或自然条件而褪色或脱落。

5.2 安装要求

终端设备安装要求如下：

- a) 装配固定：
 - 1) 需要配置导轨的模块，应方便拖出和推入，并有限位装置；
 - 2) 打印机装配应方便日常运营维护；
 - 3) 设备内部的各部件方便拆装；
 - 4) 需要经常性维护的部件使用手拧螺丝；
 - 5) 各模块应安装牢固，螺丝螺母不松动，各活动部分顺畅无明显阻力；
 - 6) 维修门四周缝隙大小符合设计要求，且均匀，锁止后无晃动；
 - 7) 各模块应安装牢固、设计合理，安装和拆卸（卡接口）卡接顺滑，现场外接辅助设备的接口设计合理，便于在现场狭小工作环境中操作。
- b) 布线端接：
 - 1) 布线时交流线缆与直流和信号线缆分离；负载冗余合理，所有通信线采用屏蔽线；
 - 2) 设备内部的布线合理、有序、整洁，标识清晰可见；
 - 3) 所有的布线都集中在布线槽内或捆扎整齐；
 - 4) 机内布线路径与活动部分接触的地方无干涉、无摩擦；
 - 5) 设备内部外露电路板应加保护罩，保护罩应使用透明、绝缘材质；
 - 6) 设备内部二层工业以太网交换机须方便网线、电源线引出，尽量减少对本设备布线及维护的影响。
- c) 标识标牌：
 - 1) 设备应粘贴标识、标牌、模块布置及线缆连接示意图；
 - 2) 设备标识牌宜采用金属材质，内容清晰、分类明确，应粘贴牢固、平整；
 - 3) 设备内配件、线缆等应有统一风格的标识，标识符合国家法规规定；
 - 4) 每根线缆都应有线缆编号，线缆两端应有唯一的标识，标识符合国家法规规定；

- 5) 所有设备、配件、电缆等应有统一风格的标志、字母、符号。每根电缆两端应有唯一的标志；
- 6) 对涉及人身安全和设备安全的地方应有醒目的安全警示标识。

5.3 电气性能

5.3.1 抗电强度

终端设备和闸门及通行控制装置（交流电源或直流电源大于48 V）的抗电强度要求为电源输入部分到人体可接触到的金属部分之间应可以承受1500 V的交流电压1 min而不被击穿。

5.3.2 绝缘电阻

终端设备和闸门及通行控制装置（交流电源或直流电源大于48 V）的绝缘电阻要求为在进行绝缘电阻测量时，电涌抑制器可以断开。试验电压为500 V直流，或者电涌抑制器保留在位，直流电压为小于电涌抑制器动作或起弧电压的10%，测得的绝缘电阻值不应小于2 M Ω 。

5.3.3 接触电流

终端设备接触电流应满足如下要求：

- a) 测试电压为110%额定电压，测试时间60 s，接触电流不应超过3.5 mA；
- b) 如果在110%的电源电压下试验不太方便，则可以在额定的电压范围内或额定电压的容差范围内任何能获得的电压下进行试验，然后再计算出最终结果。

5.3.4 接地阻抗

终端设备接地阻抗要求为测试电流采用被测电路电流额定值的1.5倍，测试电压不应超过12 V，测试时间为60 s，接地阻抗不应大于0.1 Ω 。

5.3.5 电源适应性

5.3.5.1 电源适应性要求

终端设备及专用模块电源适应性应满足表2的要求。

表2 电源适应性要求

名称	要求
自动售票机	采用交流电源供电，应能在220 V+10%至220 V-15%，50 Hz $\pm$ 2 Hz条件下正常工作。
半自动售票机	
自动检票机	
电源模块	
主控单元	
硬币处理模块	采用直流电源供电，应能在24 V $\pm$ 10%的条件下正常工作。
纸币接收模块	
纸币找零模块	
车票发售模块	
车票回收模块	
闸门及通行控制装置	采用直流或交流电源供电，应能在标称电压的 $\pm$ 10%的条件下正常工作。

名称	要求
二维码模块	采用直流电源供电，应能在 $(5\pm5\%)$ VDC 的条件下正常工作。

5.3.5.2 用电功率

终端设备用电功率应满足表 3 的要求。

表 3 终端设备用电功率表

名称	额定功率（不包含加热模块功率）	休眠功率
自动售票机	≤ 400 W	≤ 5 W
半自动售票机	≤ 400 W	≤ 5 W
自动检票机	≤ 400 W	≤ 10 W

5.4 电磁兼容性

5.4.1 无线电骚扰

专用模块、自动检票机和半自动售票机的无线电骚扰限值应满足GB 9254 A级要求。

5.4.2 电磁敏感度

专用模块、自动检票机和半自动售票机的电磁敏感度（或抗扰度限值）应满足GB/T 17618中的要求。

5.4.3 静电放电抗扰度

专用模块、自动检票机和半自动售票机的静电放电抗扰度应满足GB/T 17618中的要求。

5.4.4 CCC 认证

自动售票机应通过的CCC认证，CCC认证标志应为CCC+S&E。

5.5 噪声

噪声标准应符合表 4 的要求。

表 4 噪声标准

名称	空闲时声功率	工作时声功率
自动售票机	≤ 55 dB (A)	≤ 65 dB (A)
半自动售票机	≤ 45 dB (A)	≤ 60 dB (A)
自动检票机	≤ 55 dB (A)	≤ 60 dB (A)
硬币处理模块	≤ 55 dB (A)	≤ 85 dB (A)
纸币接收模块	≤ 55 dB (A)	≤ 70 dB (A)
纸币找零模块	≤ 55 dB (A)	≤ 85 dB (A)
车票发售模块	≤ 55 dB (A)	≤ 65 dB (A)
车票回收模块	≤ 55 dB (A)	≤ 85 dB (A)
闸门及通行控制装置	≤ 55 dB (A)	≤ 70 dB (A)
二维码模块	≤ 55 dB (A)	≤ 70 dB (A)

5.6 防尘

终端设备的防尘要求如下（半自动售票机的防尘要求不含桌面设备）：

- a) 设备安装或使用期间可能遭受到砂尘环境，应按 GB/T 2423 的相关试验方法进行试验，试验用尘种类应采用“A类尘”试验时间为1 d；
- b) 砂尘不应对产品产生以下影响：
 - 1) 改变产品电气性能（如接触电阻和表面泄露电流改变等）；
 - 2) 阻滞和磨损导轨、电机轴承等活动部件（如机芯导轨和机芯部件等）；
 - 3) 腐蚀表面，污染润滑剂；
 - 4) 阻塞通风口、管路、过滤器、操作孔等。

5.7 防水

终端设备的防水要求如下：

- a) 自动售票机在安装或使用期间可能遭受到滴水的环境，应按 GB 2423 的相关试验方法进行试验，在表 5 所列条件下，内部不应渗水，功能及电气、机械性能正常，防水要求见表 5。

表 5 自动售票机的防水要求

项目类别	要求			
	降雨强度 mm/h	水滴降落高度 m	持续时间 min	倾斜角度
自动售票机防水性能	250	2.0±0.5	10	0°

- b) 自动检票机、半自动售票机在安装或使用期间可能遭受到溅水的环境，向外壳各方向溅水应无有害影响。应按 GB 2423 的相关试验方法进行试验，在表 6 所列条件下，内部不会有渗水，功能及电气、机械性能正常（半自动售票机的防水要求不含桌面设备），防水要求见表 6。

表 6 自动检票机及半自动售票机的防水要求

项目类别	要求			
	水流量 L/min	水流压力 kPa	持续时间 min	倾斜角度
自动检票机及半自动售票机防水性能	0.6±0.003	400	10	±90°

5.8 机械环境适应性

5.8.1 振动

采用扫频试验测试终端设备及专用模块对振动的适应能力，应符合表 7 的要求。

表 7 抗振动要求

序号	项目类别	要求	
		试验内容	数值
1	初始和最后振动响应检查	频率范围/(Hz)	10~55
2		扫频速度/(oct/min)	≤1
3		驱动振幅/(mm)	0.15
4	定频耐久试验	驱动振幅/mm	0.75 (10Hz~25Hz) 0.15 (25Hz~55Hz)
5	定频耐久试验	持续时间/(min)	30±1
6	扫频耐久试验	频率范围/(Hz)	10~55~10
7		驱动振幅/(mm)	0.15
8		循环次数/(次)	5

5.8.2 冲击

闸门及通行控制装置门部分内核应采用不锈钢钢架结构，外包表面平滑的软性材料，扇门闭合时两扇叶最接近区域200 mm×200 mm的表面受到30 kg的撞击时不应发生永久变形或断裂，对冲击的适应能力应满足表 8的要求。

表 8 抗冲击要求

序号	类别	要求		
		峰值加速度	脉冲持续时间	冲击波形
1	终端设备	300 m/s ²	11 ms	半正弦波形或锯齿波或梯形波
2	专用模块	150 m/s ²	11 ms	半正弦波形或锯齿波或梯形波

5.9 温湿环境适应性

5.9.1 工作环境温度及湿度

终端设备的工作环境温度应为-20℃～45℃，工作环境相对湿度应为20%～95%（非凝露）。

专用模块的工作环境温度应为0℃～50℃，工作环境相对湿度应为20%～90%（非凝露）。

5.9.2 存储温度及湿度

终端设备的存储温度：-20℃～60℃，存储相对湿度：5%～95%（非凝露）。

专用模块的存储温度：-20℃～60℃，存储相对湿度：10%～90%（非凝露）。

5.10 可靠性

对于终端设备及专用模块的要求及参数见表 9。

表 9 可靠性要求

名称	平均无故障次数（MCBF）	总使用次数（C）	总故障次数（F）
自动售票机	≥70000	交易次数	全部关联故障
半自动售票机	≥70000	交易次数	全部关联故障
自动检票机	≥70000	交易次数	全部关联故障
硬币找零模块	≥70000	总识别枚数	总故障数包括所有与硬币识别关联故障
	≥70000	总找零枚数	总故障数包括所有与硬币找零关联故障
纸币接收模块	≥70000	纸币接收次数	总故障数包括卡钞，符合中国人民银行《不宜流通人民币挑剔标准》的纸币不作为测试样张
纸币找零模块	≥70000	纸币找零次数	总故障数包括卡钞，符合中国人民银行《不宜流通人民币挑剔标准》的纸币不作为测试样张
车票发售模块	≥70000	发售车票次数	全部关联故障
车票回收模块	≥70000	回收车票次数	全部关联故障
闸门及通行控制装置	≥100000	闸门开关次数	全部关联故障
二维码模块	≥100000	扫码次数	全部关联故障
注：平均无故障次数（MCBF）=总使用次数（C）/总故障次数（F）。			

5.11 可维护性

整机设计应便于模块更换，单个模块平均故障修复时间不大于0.5 h。

5.12 安全性

安全性应满足如下要求：

- 终端设备（不包含自动售票机）及专用模块安全性应符合 GB 4943 中的规定；
- 自动售票机应通过 CCC 认证；

- c) 终端设备、专用模块的主板和驱动板应提供有 CNAS/CMA 资质的检验报告或 CCC 资质的认证证书；
- d) 电源模块、工控机应提供有 CNAS/CMA 资质的检验报告或 CCC 资质的认证证书。

6 专用模块

6.1 票箱

6.1.1 一般要求

票箱应能在车票发售模块和车票回收模块间循环使用，并且具有车票储存、车票保护、防止车票丢失等功能，票箱图纸见附录A。

6.1.2 外形尺寸

票箱外形尺寸应为572.5 mm×103 mm×94.3 mm（高×宽×深，不包括票箱把手和拉手）。

6.1.3 功能要求

票箱功能应满足如下要求：

- a) 框架、滑盖和前盖组成一个密闭的箱体，应保证在票箱运送、运营更换过程中，没有专用钥匙的人员不能接触到车票；
- b) 设置应有拉手（横向）和提手（纵向）。
- c) 更换票箱时应不需要调整托盘高度即可安装至模块；
- d) 托盘应根据票箱中的车票的数量而停留在合适的高度，不会在自重或车票的压力作用下而下滑。

6.1.4 性能要求

票箱的性能指标：票箱容量≥800 张。车票规格为54.0 mm×85.5 mm×0.5 mm。

6.2 硬币处理模块

6.2.1 外形尺寸

应不大于350 mm×580 mm×780 mm（宽×深×高，不包括回收箱）。

6.2.2 配置要求

6.2.2.1 硬件配置

硬币处理模块各部分的主要配置见表 10。

表 10 硬币处理模块硬件配置要求

序号	部件名称	部件描述
1	硬币识别部分	具有硬币快速识别的功能，可以配置可接收面额。
2	硬币找零部分	可以将模块中的硬币传送到模块外部或者收纳部分中，并能准确的辨识数量。
3	硬币暂存部分	暂时存放识别但未储存的硬币，并具备返还功能的部件。
4	硬币收纳部分	存放硬币的机构，包括钱箱等。
5	硬币传送部分	将模块中的硬币传输到指定位置的机构。
6	控制部分	包含主控制器、存储器及各种控制接口，如马达控制单元、传感器信号采集单元、通信及电源接口等。

6.2.2.2 软件配置

硬币处理模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的软件开发包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。接口设计要求见10.6。

6.2.2.3 接口要求

硬币处理模块应提供满足GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232（DB9）接口。

6.2.3 功能要求

硬币处理模块的功能要求见表 11。

表 11 硬币处理模块功能要求

序号	功能名称	功能说明
1	硬币识别接收功能	通过硬币接收器识别硬币真假，识别为真币的硬币进入暂存器，识别为假币的硬币直接退出。
2	原币返还功能	当取消交易时，从暂存器将硬币倒入找零口退还。
3	循环找零功能	可将暂存器的硬币倒入循环找零器，实现循环找零功能。
4	专用找零功能	当需要找币时，可以从主找零器将相应数量的硬币找给乘客。
5	硬币清空功能	将模块里的所有硬币回收到硬币回收箱里。
6	固件升级功能	支持在线和点位升级两种方式。

6.2.4 性能要求

硬币处理模块的主要性能指标见表 12。

表 12 硬币处理模块性能指标

序号	项目类别		要求
1	硬币识别器	假币拒收率	≥99.9%
2		真币接收率	≥99%
3		识别能力	至少 16 种规格或面值的硬币
4		单枚识别时间	≤0.4 s
5		寿命	≥100 万枚
6	主找零器	数量	≤2 个
7		容量	≤1000 枚 1 元硬币/个, ≤1000 枚 5 角硬币/个
8		寿命	≥100 万枚
9		找零速度	≥6~8 枚每秒（不空转，不翻转的时候）
10		清空率	≥99.9%
11	缓存找零器	数量	≤2 个
12		容量	≤100 枚 1 元硬币/个
13		寿命	≥100 万枚
14		找零速度	约 6 枚每秒（不空转，不翻转的时候）
15		清空率	≥99.9%
16	硬币暂存器	数量	≤2 个
17		容量	≤30 枚 1 元硬币/个, ≤30 枚 5 角硬币/个
18	硬币钱箱	数量	≤2 个
19		容量	≤1000 枚 1 元硬币, ≤1000 枚 5 角硬币
注：符合中国人民银行《不宜流通人民币挑剔标准》的硬币不作为测试样本。			

6.3 纸币接收模块

6.3.1 外形尺寸

应不大于175 mm×700 mm×600 mm（宽×深×高）。

6.3.2 配置要求

6.3.2.1 硬件配置

纸币接收模块主要配置见表 13。

表 13 纸币接收模块硬件配置要求

序号	部件名称	部件描述
1	纸币识别部分	具有对纸币快速识别的功能，可以在线更新纸币的识别面额，限制识别的面额等。
2	纸币传送部分	包括入钞口，退钞口，纸币在机芯内的传输通道以及通道的路径切换，以保障纸币在传输通道内传送顺畅。
3	纸币暂存部分	暂存纸币，同时具有叠钞功能。
4	纸币收纳部分	存放纸币的机构。包括钱箱及压钞机构等。
5	控制部分	包含主控制器、存储器及各种控制接口，如马达控制单元、传感器信号采集单元、通信及电源接口等。
6	通信接口	接收上位机指令，控制纸币的运行路径，同时报告模块的状态信息，并记录模块的各种运行情况，便于分析模块的运行状态。

6.3.2.2 软件配置

纸币接收模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的开发工具包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。终端设备调用接口封装要求见10.1 。

6.3.2.3 接口要求

纸币接收模块应提供满足GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232（DB9）接口。

6.3.3 功能要求

纸币接收模块的功能要求见表 14。

表 14 纸币接收模块功能要求

序号	功能名称	功能说明
1	纸币纠正功能	能把倾斜投入的纸币纠正到能正确识别和传送的位置。
2	识别功能	具有 5 种以上识别方式，正确识别纸币，能识别 13 种以上现行流通钞。
3	退币功能	能安全把暂存区所有纸币退出，对于不识别或被禁止面额的纸币，直接从退钞口退出。
4	叠钞功能	整齐将纸币堆叠，并整叠传送到相应位置。
5	暂存功能	能停放 15 张纸币，等待主控机的下一步指令动作。
6	传送压箱功能	将纸币平整送进钱箱。
7	固件升级功能	对纸币识别模块的程序及纸币数据，支持在线和点位升级两种方式。
8	钱箱互锁功能	纸币钱箱在取出及打开时，应分别具有独立的安全锁，并用不同的钥匙打开。纸币钱箱在安装到位时应能自动锁闭同时入钞口自动打开；钱箱取走时，入钞口应能自动关闭。
9	日志记录功能	记录机芯运行的各种状态，便于故障的分析。
10	状态监测功能	能通过指令或接口反馈模块当前状态，如钱箱满/正常、钱箱在位/离位、通道卡钞等。

6.3.4 性能要求

纸币接收模块性能要求见表 15。

表 15 纸币接收模块性能指标

序号	项目类别	要求	备注
1	单张识别时间	≤2.5 秒	
2	单次真钞接收率	≥95%	
3	假币拒收率	≥99.99%	
4	支持纸币种类	≥13 种（现流通的全币种）	
5	识别采样手段	≥5 种	识别手段应满足 GB 16999 要求。
6	纸币暂存张数	≥15 张	
7	纸币钱箱容量	≤1000 张	
注：选用市场上可正常流通的纸币进行测试，符合中国人民银行《不宜流通人民币挑剔标准》的纸币不作为测试样张。			

6.4 纸币找零模块

6.4.1 外形尺寸

应不大于300 mm×700 mm×520 mm（宽×深×高）。

6.4.2 配置要求

6.4.2.1 硬件配置

纸币找零模块各部分的主要配置见表 16。

表 16 纸币找零模块硬件配置要求

序号	部件名称	部件描述
1	找零存储部分	具备将纸币整叠收纳功能，可根据纸币币种调节纸币导板的长、宽。
2	纸币擦钞部分	通过滚轮和分离器把纸币逐张分离、并传送出来的子模块，擦钞轮磨损后应易于更换。
3	通道传送部分	传送从找零存储箱中擦出的纸币，具备将不合格的纸币分拣至回收部分的机构。
4	厚度检测部分	在传送过程中，检测纸币的厚度。纸币通过滚轮中间时，通过位置可动的滚轮的位置变化，检测纸币的厚度。
5	纸币回收部分	容纳不良纸币的钱箱，可以取出。宜加回收箱锁，控制回收箱的开关。
6	顶部传送部分	发放纸币的传送机构。具备对出钞纸币进行计数，确认是否出钞完毕的功能。
7	控制部分	包含主控制器、存储器及各种控制接口，如马达控制单元、传感器信号采集单元、通信及电源接口等。

6.4.2.2 软件配置

纸币找零模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的软件开发包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。接口设计要求见10.2 章节。

6.4.2.3 接口要求

纸币找零模块应提供满足GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232（DB9）接口。

6.4.3 功能要求

纸币找零模块的功能要求见表 17。

表 17 纸币找零模块功能要求

序号	功能名称	功能说明
1	拾钞功能	吸出钱箱中的纸币。

序号	功能名称	功能说明
2	分钞功能	确保每次只有 1 张纸币送出。
3	验钞功能	可鉴别重叠纸币、折叠纸币、破损纸币等。
4	回收功能	可检验纸币的厚度、长度和宽度尺寸，并对不合格纸币进行回收，回收箱为封闭式。
5	传送功能	将纸币通过通道送出。
6	少钞检测功能	检测钞箱纸币空和将空状态。
7	纸币自学习功能	自动记录纸币厚度信息。
8	日志记录功能	记录机芯运行的各种状态，便于故障的分析。
9	脱机诊断分析功能	脱机时可对故障原因诊断分析。
10	固件升级功能	支持在线和点位升级两种方式。

6.4.4 性能要求

纸币找零模块的性能要求见表 18。

表 18 纸币找零模块性能指标

序号	项目类别	要求	
		横向	纵向
1	单张出钞速度	$\geq 4 \text{ s}$	$\geq 1 \text{ s}$
2	回收率	$\leq 1\%$	$\leq 1\%$
3	卡钞率	$\leq 0.03\%$	$\leq 0.03\%$
4	纸币找零钱箱容量	$\leq 1000 \text{ 张}$	
5	纸币找零钱箱数量	$\leq 2 \text{ 个}$	

注：选用市场可正常流通的纸币进行测试，符合中国人民银行《不宜流通人民币挑剔标准》的纸币不作为测试样张。

6.5 车票发售模块

6.5.1 外形尺寸

车票发售模块外形尺寸要求如下：

- 双储票箱车票发售模块外形尺寸应不大于 220 mm×610 mm×950 mm（宽×深×高）；
- 单储票箱车票发售模块外形尺寸应不大于 155 mm×393 mm×717 mm（宽×深×高）。

6.5.2 配置要求

6.5.2.1 硬件配置

车票发售模块各部分的主要配置要求见表 19。

表 19 车票发售模块硬件配置要求

序号	部件名称	部件描述
1	车票分离部分	将车票从票箱送入传输通道，且保证每次出一张卡的装置。
2	传输通道部分	车票出票箱至出票口在内的一段通道，负责车票的数据读写和送出。
3	废票箱	用来回收不能再次循环使用或自动清理掉或不符合使用规范车票的容器。
4	储票箱	用来放置未发售车票的容器。
5	主控部分	模块的主控制板。
6	支架部分	模块的主体支架。

6.5.2.2 软件配置

车票发售模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的开发工具包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。终端设备调用接口封装要求见 10.4 章节。

6.5.2.3 接口要求

车票发售模块应提供满足GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232（DB9）接口。

6.5.3 功能要求

车票发售模块的功能要求见表 20。

表 20 车票发售模块功能要求

序号	功能名称	功能说明
1	备票功能	将指定票箱中的车票送到读写区。
2	送票功能	将读写区车票送回到出票口。
3	车票回收功能	将读写区车票送到废卡回收箱。
4	清理通道功能	将因堵塞或其他故障而残留在通道中的车票送到回收箱，主要用于排除故障，将模块恢复至正常工作状态。
5	状态报警功能	检测票箱内车票的状态，对于车票将空和已空进行报警。
6	固件升级功能	支持在线和点位升级两种方式。

6.5.4 性能要求

6.5.4.1 双储票箱车票发售模块

双储票箱车票发售模块的性能要求见表 21。

表 21 双储票箱车票发售模块性能指标

序号	项目类别	要求
1	单张车票发售处理速度（不包含读写器车票处理时间）	≤ 0.8 s
2	储票箱数量	2 个
3	单个储票箱容量	≤ 800 张
4	废票箱数量	1 个
5	废票箱容量	≤ 300 张
6	票箱保留不可发售车票数量	≤ 10 张

6.5.4.2 单储票箱车票发售模块

单储票箱车票发售模块的性能要求见表 22。

表 22 单储票箱车票发售模块性能指标

序号	项目类别	要求
1	单张车票发售处理速度（不包含读写器车票处理时间）	≤ 0.8 s
2	储票箱数量	1 个
3	单个储票箱容量	≤ 800 张
4	废票箱数量	1 个
5	废票箱容量	≤ 300 张
6	票箱保留不可发售车票数量	≤ 10 张

6.6 车票回收模块

6.6.1 外形尺寸

应不大于170 mm×680 mm×800 mm（宽×深×高，不含把手）。

6.6.2 配置要求

6.6.2.1 硬件配置

车票回收模块各部分的主要配置要求见表 23。

表 23 车票回收模块硬件配置要求

序号	部件名称	部件描述
1	插票部分	车票进口。在停止服务时或在处理车票过程中，可以阻止异物或车票插入。
2	读写部分	车票读写装置。
3	传输通道部分	车票在模块内运动的路径。
4	支架部分	模块的主体支架。
5	票箱	存放车票的容器。
6	废票箱	回收异常车票的容器。
7	主控部分	模块的主控制板。

6.6.2.2 软件配置

车票回收模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的开发工具包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。终端设备调用接口封装要求见10.5 章节。

6.6.2.3 接口要求

车票回收模块应提供满足GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于 20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232(DB9)接口。

6.6.3 功能要求

车票回收模块的功能要求见表 24。

表 24 车票回收模块功能要求

序号	功能名称	功能说明
1	吸卡功能	将从插卡口插入的车票吸入到读写区。
2	收纳功能	由主控单元控制将读卡区的车票送到指定的票箱中（包括票箱 A、票箱 B 和废票箱）。
3	退卡功能	由主控单元控制将读卡区的车票退回到插票口。
4	防止插卡功能	在停止服务时或在处理车票过程中，可以阻止异物或车票插入。
5	通道清理功能	将因堵塞或其他故障而残留在通道中的车票送到回收箱，主要用于排除故障，将模块恢复至正常工作状态。
6	废票回收功能	由主控单元控制将读写区的车票回收到废票箱。
7	状态报警功能	检测票箱内车票的状态，对于车票将满和已满进行报警。
8	固件升级功能	支持在线和点位升级两种方式。

6.6.4 性能要求

车票回收模块的性能要求见表 25。

表 25 车票回收模块的性能指标

序号	项目类别	要求
1	单张车票回收处理速度（不考虑读写处理时间）	≤0.8 s
2	储票箱数量	2 个
3	储票箱总容量	≤800 张
4	废票箱总容量	≤300 张

6.7 闸门及通行控制装置

6.7.1 外形尺寸

闸门及通行控制装置外形尺寸要求见表 26。

表 26 闸门及通行控制装置外形尺寸

序号	类型状态		尺寸要求（单位：mm）				
	机芯类型		闸门状态	长	闸门伸出时尺寸	宽	高
1	标准通道	双折门	闸门打开	≤475	255～270	≤390	≤900
			闸门关闭	≤196	N/A	≤390	≤900
2		单折门	闸门打开	≤490	≤240	≤390	≤900
			闸门关闭	≤250	N/A	≤390	≤900
3	宽通道	闸门打开	≤730	420～445	≤390	≤900	
4		闸门关闭	≤296	N/A	≤390	≤900	
注：闸门关闭时最小间隔双折门不大于 40 mm，单折门不大于 80 mm。							

6.7.2 配置要求

6.7.2.1 硬件配置

闸门及通行控制装置各部分的主要配置要求见表 27。

表 27 闸门及通行控制装置硬件配置要求

序号	部件名称	部件描述	硬件配置	
			必选	可选
1	电机部分	通过联动装置驱动闸门动作的部件。	√	
2	减速器部分	降低旋转速度，控制闸门运行平稳的部件。		√
3	闸门位置检测部分	检测闸门运动位置、状态的部件。	√	
4	控制单元	完成整个模块部件各种功能的控制管理单元。	√	
5	框架部分	固定整个闸门模块的部件。	√	
6	门部分	通道中的阻挡部件。	√	
7	联动装置部分	电机与扇门的传动机构。	√	
8	通行逻辑处理部分	用于通行逻辑判断的传感器组件（含身高检测传感器）和电路板以及程序软件。	√	

6.7.2.2 软件配置

闸门及通行控制装置应随模块提供底层驱动程序，相应的通信协议及命令接口集文档，可供第三方做二次开发的开发工具包及接口说明文档，以及具有独立的调试及诊断软件工具，日志分析工具等。终端设备调用接口封装要求见10.3 章节。

6.7.2.3 接口要求

闸门及通行控制装置应提供符合GB/T 6107（对于最大20 kb/s的数据信号速率的操作）或GB/T 17183（对于大于20 kb/s数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行RS232接口。

6.7.3 功能要求

闸门及通行控制装置的功能要求见表 28。

表 28 闸门及通行控制装置功能要求

序号	功能名称	功能说明
1	失电放行功能	在断电情况下，扇门应能自动打开

2	模式设置功能	通过上位机可从串行通信端口设置成常开模式和常闭模式
3	安全控制功能	通过扇门的承受力检测后的自动释放或打开，防止夹人 通过防夹传感器及通行逻辑检测，防止夹人
4	通行控制功能	通过与其它模块的联动，扇门接受控制板命令和电机驱动力完成开/闭动作实现通道的通行/阻挡功能 通过通行逻辑检测单元，实现不同需求的通行控制，包括：行李检测、尾随检测、身高检测及组合检测，检测完成后，根据通行规则，扇门正常开闭
5	其它功能	扇门在一定外力的作用下能够推开，而不会造成扇门驱动机构的损坏
6	固件升级功能	支持在线和点位升级两种方式。

6.7.4 性能要求

闸门及通行控制装置的性能要求见表 29。

表 29 闸门及通行控制装置的性能指标

序号	项目类别		要求
1	扇门运行时间	标准通道	$\leq 0.5 \text{ s}$
2		宽通道	$\leq 0.7 \text{ s}$
3	锁死力		$250 \text{ N} \pm 30\%$
4	最大动态冲击力		$250 \text{ N} \pm 30\%$
5	传感器数量		$\geq 16 \text{ 对}$

6.8 二维码模块

6.8.1 外形尺寸

模块识读窗口尺寸应大于等于 $75 \text{ mm} \times 50 \text{ mm}$ （长×宽）；

6.8.2 配置要求

6.8.2.1 硬件配置

二维码模块主要配置见表 30。

表 30 二维码模块硬件配置要求

序号	部件名称	部件描述
1	芯片	处理控制命令。
2	二维码扫描头	二维码扫描。
3	连接线缆	与设备上位以及内部连接。
4	主体结构	安装设备部件。

6.8.2.2 软件配置

二维码模块应随模块提供底层驱动程序、相应的通信协议及命令接口集文档、可供第三方做二次开发的开发工具包及接口说明文档以及具有独立的调试及诊断软件工具、日志分析工具等。

6.8.2.3 接口要求

二维码模块应提供满足 GB/T 6107（对于最大 20 kb/s 的数据信号速率的操作）或 GB/T 17183（对于大于 20 kb/s 数据信号速率的操作）要求的通信接口，支持同步或异步数据传输，采用串行 RS232（DB9）接口。

6.8.3 功能要求

二维码模块的功能要求见表 31。

表 31 二维码模块功能要求

序号	功能名称	功能说明
1	业务处理功能	具备二维码扫描功能，在一定范围内能够感应二维码存在并读取。
2	远程监控功能	支持状态数据监控。
3	识别功能	应能识读 QR Code 码制。能通过指令控制受控识读或感应识读。
4	照明功能	支持 LED 照明。
5	提示功能	能通过指令控制蜂鸣器开关。
6	固件升级功能	支持在线或点位升级两种方式。

6.8.4 性能要求

二维码模块性能要求见表 32。

表 32 二维码模块性能要求

序号	项目类别	要求
1	识读精度	≥20mil
2	灵敏度	倾斜±30°，旋转 360°，偏转±30°
3	视场角度	水平≥65°，垂直≥50°，对角线≥80°
4	识别高度	20 mil 二维码有效垂直扫描距离≥50 mm
5	识读速度	≤ 100 ms（采用 QR Code 300 字节）
6	环境照度	应能在 100~100,000 lx 光照条件下正常识读二维码。
注：二维码模块识读速度定义为从控制程序发出扫码指令至二维码模块开始，至二维码模块扫到码后，控制程序收到模块返回完整二维码数据的时间。		

6.9 电源模块

6.9.1 外观与结构

电源模块整体采用开关电源。电源外壳必须设计安全接地连接螺孔或螺柱，电源的外壳的所有接缝应搭接良好。外壳边沿光滑，无毛刺，不伤害人体。结构安全性符合国家CQC强制认证。直流分线板可以独立或与电源模块整合。

6.9.2 标志标识

电源模块的标志、标识要完整，清楚醒目，能够耐久，不易脱落、擦除，语言文字应符合下列要求：

- a) 输入额定值；
- b) 输出额定值；
- c) 额定功率；
- d) 电源性质符号；
- e) 制造厂商规定的机型号或型号标志；
- f) 制造厂商的名称和商标、或识别标记；
- g) 制造厂商出厂日期；
- h) 制造厂商检验合格证；
- i) 标示 CCC 认证标志；
- j) 电源输入、输出标识，指示灯标识；
- k) 接线端子要有明确的极性标志；
- l) 保护接地端子必须标示安全接地符号；
- m) 电源开关上或附近应标记通 ‘|’ 断符号 ‘0’。

6.9.3 环境要求

电源模块环境要求如下：

- a) 工作环境温度：-15℃～60℃；
- b) 工作环境湿度：5%～90 %，不结露；
- c) 存储/运输环境温度：-40℃～85℃；
- d) 存储/运输环境湿度：5%～95 %，不结露。

注：环境温度为最大功率耗散下的稳态温度。

6.9.4 噪声要求

对于安装风扇的电源，其噪声在1.5 m处测量不应大于40 dB。

6.9.5 电性能

电源模块电性能要求如下：

- a) 额定电压输出端口应至少具备 12 V、24 V 两类；
- b) 具备瞬变保护；
- c) 具有漏电保护；
- d) 保护措施为电源输出设备具备过电压、过电流保护；
- e) 电源线性调整率最大为 0.4%；
- f) 电源负载调整率最大为 0.8%；
- g) 输出纹波及干扰小于 1%或峰-峰值为 100 mV；
- h) 有效性大于 70%。

6.9.6 安全性

电源模块应符合CQC国家强制认证标准，应通过CCC认证。

6.9.7 电磁兼容性

电源模块应通过CCC认证。

6.9.8 功能要求

电源模块功能要求如下：

- a) 电源应具有开启、待机、关闭运行状态，并具有远程唤醒功能；
- b) 当环境温度高于 65℃或冷却风扇失效的情况下，能提供过热保护；当外部条件恢复正常后电源能自动恢复；
- c) 具备主动式 PFC 功能，PFC 应大于 0.93。

6.9.9 可靠性

应大于100,000 h。

6.10 主控单元

6.10.1 外观与结构

主控单元外观结构要求如下：

- a) 全密闭结构设计，密闭防尘防潮；
- b) 主控单元及其接口的设计应模块化，应满足物理上和功能上的互换性要求，部件设计应牢固，连接可靠，避免接插件松脱或接触不良；
- c) 主控单元内部需要维护或可能更换的部件应便于拆卸，不应被其它部件遮挡。

6.10.2 配置要求

主控单元配置要求如下：

- a) 主控单元应采用宽温型，低功耗、免风扇设计；
- b) 主控单元应不低于 32 位工业级微处理器；
- c) 主控单元除满足系统功能要求的软件及硬件接口外，应至少预留两个标准的 RS232 和两个标准的 USB 接口。

6.10.3 功能要求

主控单元功能要求如下：

- a) 具有良好的抗电磁干扰性能，保证整机 24 h 不间断稳定运行，并具备足够的能力提供所指定功能；
- b) 存储设计应能支持多种存储方式，以保证异常断电时交易数据、寄存器数据及系统参数不丢失；
- c) 电源故障时不会损坏已存储的数据；
- d) 内置实时时钟维持当前日期及时间，其准确性至少为 $\pm 1\text{ s/日}$ 。时钟应在电池供电下工作。时钟运行应不需人工调整润年、年尾、月尾及星期；
- e) 具有自动防故障方法检测主控单元故障，并具备在必要时复位的能力。

7 自动售票机

7.1 外形和结构要求

7.1.1 外形尺寸

应不大于 $900\text{ mm}\times 800\text{ mm}\times 1800\text{ mm}$ （宽 $\times$ 深 $\times$ 高）。

7.1.2 安装尺寸及方式

自动售票机安装底座尺寸见附录 B 中自动售票机的底座图纸。

7.1.3 面板布局

自动售票机面板布局应满足如下要求：

- a) 前面板的布局有足够及明显的标记提示，易操作，保证信息无障碍；
- b) 操作接口的设计应不会因不正常或恶意的操作而出现错误运作或损坏；
- c) 面板提供中英文标识和操作步骤说明。

7.2 配置要求

7.2.1 硬件选配

自动售票机各单元及主要模块见表 33。

表 33 自动售票机各单元包含的主要模块

序号	单元名称	包含模块	硬件配置	
			必选	可选
1	主控部分	工业级计算机	√	
2	车票处理部分	车票发售模块	√	
3		储值卡充值模块	√	
4		车票处理单元	√	
6	找零部分	纸币找零模块	√	
7		硬币处理模块	√	
8	支付部分	纸币接收模块	√	
9		银行卡处理模块		√
10		二维码模块		√
11	人机交互部分	触摸屏	√	
12		乘客显示器	√	
13		运营状态显示器	√	
14		凭条打印机	√	
15		维修模块	√	
16		招援按钮	√	

序号	单元名称	包含模块	硬件配置	
			必选	可选
17	供电部分	电源箱	√	
18		漏电保护开关	√	
19	其他辅助部分	维修灯	√	
20		维修插座	√	
21		加热模块		√
22		通风模块	√	
23		底座	√	

7.2.2 加热模块配置

自动售票机加热模块配置应满足如下要求：

- 加热模块应采用强制散热的加热器，不应影响整机内部易受高温影响的元器件。加热元件应具有安全、升温迅速、无明火、无氧耗、不易燃烧等特点；
- 加热模块应有明确的开始加热和停止加热温度指标，在工作范围内能自动启停，开始和停止加热的温度误差不超过 $\pm 3^{\circ}\text{C}$ 。在 25°C 环境下，工作时最大本体温度不超过 70°C ；
- 加热模块使用交流电源时，应确保接地可靠，电器保护等级不低于 I 级；
- 额定工作电压：220 V/50 Hz，温度控制范围： $0^{\circ}\text{C}\sim 99^{\circ}\text{C}$ ；
- 温度控制准确度： 1°C ，测量、显示温度范围： $-30^{\circ}\text{C}\sim 99^{\circ}\text{C}$ 。

7.2.3 软件配置

自动售票机软件配置要求如下：

- 操作系统应为 linux 内核 2.6.18 及以上 gcc 4.1.2；
- 应有自动售票机应用软件；
- 应有自动售票机故障诊断软件；
- 应有防病毒措施。

7.2.4 接口要求

自动售票机应提供串口、USB、以太网等通信接口，支持同步或异步数据传输。

7.3 功能要求

自动售票机应满足DB11/T 1164.1规定的功能要求。

7.4 性能要求

自动售票机性能指标要求见表 34。

表 34 自动售票机的性能指标

序号	项目类别		要求	备注
1	单张 车票 发售 时间	硬币支付无找零	≤3 s	
2		纸币支付无找零	≤4 s	
3		纸币支付硬币找零	≤7 s	
4		纸币支付混合找零	≤9 s	
5		非现金支付	≤2 s	二维码：主扫与被扫两种方式
注：单张车票发售时间定义为从完成投币后按确认或从完成投币后开始或非现金支付完成后开始，到车票和找零钱币完全送出的时间。				

8 半自动售票机

8.1 外形和结构要求

半自动售票机外形尺寸要求如下：

- a) 出票单元应不大于 300 mm×800 mm×1200 mm（宽×深×高）；
- b) 控制柜应不大于 400 mm×400 mm×500 mm（宽×深×高）。

8.2 配置要求

8.2.1 硬件配置

半自动售票机各单元及主要模块见表 35。

表 35 半自动售票机各单元包含的主要模块

序号	单元名称	主要模块	硬件配置	
			必选	可选
1	主控部分	工业级计算机	√	
2	车票处理部分	车票发售模块	√	
3		车票处理单元	√	
4	支付部分	二维码模块		√
5		银行卡处理模块		√
6	人机交互部分	操作员显示器	√	
7		乘客显示器	√	
8		凭条打印机	√	
9	供电部分	电源箱	√	
10		漏电保护开关	√	
11	其他辅助部分	维修插座	√	

8.2.2 软件配置

半自动售票机软件配置要求如下：

- a) 操作系统应为 linux 内核 2.6.18 及以上 gcc 4.1.2；
- b) 有半自动售票机应用软件；
- c) 应有半自动售票机故障诊断软件；
- d) 应有防病毒措施。

8.2.3 接口要求

半自动售票机应提供串口、USB、以太网等通信接口，支持同步或异步数据传输。

8.3 功能要求

半自动售票机应满足DB11/T 1164.1规定的功能要求。

8.4 性能要求

半自动售票机的性能指标：出票速度小于或等于 1 s/张。

9 自动检票机

9.1 外形和材料要求

9.1.1 材料

自动检票机材料要求如下：

- a) 自动检票机乘客显示器如采用玻璃材质应采用抗压强度不小于 400 MPa 的透明钢化玻璃；
- b) 自动检票机表面的刷卡标识帖等采用环保耐磨的材料；

- c) 自动检票机外壳上设置的传感器镜片应选用透光性好,且能够避免强光或其他光线干扰的材料制成,并经过防静电处理。

9.1.2 外形尺寸

自动检票机外形尺寸应满足如下要求:

- a) 常规双折门通道检票机机体应不大于 2000 mm×200 mm×1100 mm (长×宽×高);
- b) 适用于常规双折门的标准通道宽度不大于 550 mm, 闸门关闭时最小间隔不大于 40 mm;
- c) 常规单折门通道检票机机体应不大于 2000 mm×250 mm×1100 mm (长×宽×高);
- d) 适用于常规单折门的标准通道宽度不大于 550 mm, 闸门关闭时最小间隔不大于 80 mm;
- e) 双折门宽通道检票机机体应不大于 2000 mm× 300 mm×1100 mm (长×宽×高);
- f) 适用于双折门的宽通道宽度为 900 mm, 闸门关闭时最小间隔不大于 40 mm。

9.1.3 安装尺寸及方式

应设有安装底座,以便于稳定、水平调整、防水、布线等;底座应有良好的接地措施保证操作人员安全,接地电阻应小于等于 1Ω 。常规双折门通道、常规单折门通道与宽通道检票机安装尺寸见附录B。

9.2 配置要求

9.2.1 硬件配置

自动检票机各单元及主要模块见表 36。

表 36 自动检票机各单元包含的主要模块

序号	单元名称	包含模块	硬件配置	
			必选	可选
1	主控部分	工业级计算机	√	
2	车票处理部分	车票回收模块		√
3		车票处理单元	√	
4	互联网部分	二维码模块	√	
5	闸门及通行控制装置	闸门	√	
6		通行逻辑检测单元	√	
7	人机交互部分	乘客显示器	√	
8		方向指示器	√	
9		维修模块	√	
10		警示灯	√	
11		功放及扬声器	√	
12	供电部分	电源箱	√	
13		漏电保护开关		√
14	其他辅助部分	I/O 板	√	
15		维修灯	√	
16		维修插座	√	
17		到位开关	√	
18		加热模块		√
19		通风模块		√
20		底座	√	
注：一组自动检票机应有含车票回收模块的自动检票机。				

9.2.2 加热模块配置

自动检票机加热模块配置要求如下:

- a) 加热模块应采用强制散热的加热器，不应影响整机内部易受高温影响的元器件。加热元件应具有安全、升温迅速、无明火、无氧耗、不易燃烧等特点；
- b) 加热模块应有明确的开始加热和停止加热温度指标，在工作范围内能自动启停，开始和停止加热的温度误差不超过 $\pm 3^{\circ}\text{C}$ 。在 25°C 环境下，工作时最大本体温度不超过 70°C ；
- c) 加热模块使用交流电源时，应确保接地可靠，电器保护等级不低于 I 级；
- d) 额定工作电压： $220\text{ V}/50\text{ Hz}$ ，温度控制范围： $0^{\circ}\text{C}\sim 99^{\circ}\text{C}$ ；
- e) 温度控制准确度： 1°C ，测量、显示温度范围： $-30^{\circ}\text{C}\sim 99^{\circ}\text{C}$ 。

9.2.3 软件配置

自动检票机软件配置要求如下：

- a) 操作系统应为 linux 内核 2.6.18 及以上 gcc 4.1.2；
- b) 应有自动检票机应用软件；
- c) 应有自动检票机故障诊断软件；
- d) 应有防病毒措施。

9.2.4 接口要求

自动检票机应提供串口、USB、以太网等通信接口，支持同步或异步数据传输。

9.3 功能要求

自动检票机应满足DB11/T 1164.1规定的功能要求。

9.4 性能要求

具体自动检票机的性能要求见表 37。

表 37 自动检票机性能指标

序号	项目类别	指标名称	要求	备注
1	通行率	车票进站：	$\geq 50\text{ 人/min}$	
2		非回收类车票出站：	$\geq 50\text{ 人/min}$	不含二维码
3		回收类车票出站：	$\geq 40\text{ 人/min}$	
4		二维码通行率：	$\geq 45\text{ 人/min}$	
5	尾随距离	乘客尾随最小间距报警值	$\leq 300\text{ mm}$	
6		乘客尾随最小关门距离	$\geq 600\text{ mm}$	
7	读写识别性能	票卡读写距离	大天线最大读写距离不低于 60 mm ，小天线最大读写距离不低于 40 mm	
8		二维码扫描距离	$0\sim 50\text{ mm}$	二维码模块可正常识读

10 接口设计要求

10.1 纸币接收模块接口

10.1.1 结构体说明

10.1.1.1 错误与状态描述结构体

```
typedef struct
{
    //基本信息
    UINT uiProviderId; //模块厂家标识
    int iType; //错误类型：0-正常；1-警告；2-故障
    int iStdErrorCode; //统一维护的错误代码
```

```

//2. 模块的原始信息，供上位程序显示维护用
int iErrorCode; //错误代码（由不同模块厂家提供）
char acDevReturn[128]; // 硬件返回信息
char acReserve[128]; // 保留信息
} tSTDDevReturn;

```

10.1.1.2 钞票系列禁用信息结构体

```

typedef struct _SetInhibitItem
{
    int iValue; //面额，单位为分
    bool bAccepted; //是否接收
} tSetInhibitItem;
typedef struct _SetInhibitList
{
    int iNumOfItems; //现金体个数
    tSetInhibitItem *psBnaCashInfo; //现金数组基址指针
} tSetInhibitList;

```

10.1.1.3 现金信息结构体

```

typedef struct _BnaCashItem;
{
    unsigned Int iValue; //接收钞票面额，单位为分
    unsigned Int iCount; //钞票张数
} tBnaCashItem;
typedef struct _BnaCashInfo
{
    int iNumOfItems; //现金体个数
    tBnaCashItem *psBnaCashInfo;
} tBnaCashInfo;

```

10.1.1.4 纸币接收钱箱状态结构体

```

typedef struct
{
    int iModuleStatus; /* 0 正常, 1 需复位 2 异常 */
    int iBoxStatus; /* 0 安装 (正常), 1 将满 2 满 3 未安装 */
    int iInStacker; /* 0 无, 1 有钞票在暂存 */
} tBnaStatus

```

10.1.2 接口函数

说明：

- 所有函数返回值只判断传入参数是否为空；
- 逻辑或物理错误还必须判断 tSTDDevReturn 的 iType 和其他成员变量来确定；
- 每个函数内部都有线程锁，以保证标准接口函数都是串行运行：
(BNA_Start 与 BNA_Stop 除外，这两个标准接口，一个可以运行在主线程中，一个运行在新创建的子线程中，分别用以开始纸币接收与结束纸币接收)。

10.1.2.1 初始化

```
Int BNA_Init(tSTDDevReturn* p_psStatus)
```

功能说明：初始化模块，检测模块各硬件、将模块恢复到初始状态等。

参数说明见表 38。

表 38 BNA_Init() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 39。

表 39 BNA_Init() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.2 设置识别可用钞票

```
int BNA_SetPrevalue(tSetInhibitList *p_psBuyTicketSetInhibitList,
tSetInhibitList *p_psRechargeSetInhibitList,
tSTDDevReturn * p_psStatus)
```

功能说明：此命令用于设置可接收的钞票，对于禁止接收的真钞会退出。

参数说明见表 40。

表 40 BNA_SetPrevalue() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSetInhibitList *	p_psBuyTicketSetInhibitList	In	设置售票时钞票可用信息结构体数组
2	tSetInhibitList *	p_psRechargeSetInhibitList	In	设置充值时钞票可用信息结构体数组
3	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 41。

表 41 BNA_SetPrevalue() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.3 复位

```
int BNA_Reset(int iMode, int iRetryTimes, BYTE * p_byNotesRefunded,
BYTE * p_byNotesEncashed, tSTDDevReturn * p_psStatus)
```

功能说明：此命令用于清理通道中的钞票，机芯恢复正常工作。

参数说明见表 42。

表 42 BNA_Reset() 参数说明

序号	参数类型	参数名称	属性	说明
1	Int	iMode	In	如果通道中有钞票，可以选择的操作由 p_iResetMode 设定 可以设置为： 0：将传输系统和暂存区的钞票压入钱箱，如果失败则报暂停服务错误； 1：将传输系统和暂存区的钞票退出，如果失败则报暂停服务错误； 默认为 0 方式
2	Int	iRetryTimes	In	复位重试次数
3	BYTE*	p_byNotesRefunded	Out	复位中纸币退钞张数，默认为 NULL
4	BYTE*	p_byNotesEncashed	Out	复位中纸币压钞张数，默认为 NULL
5	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 43。

表 43 BNA_Reset() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.4 开始接收钞票

```
int BNA_Start(tSTDDevReturn * p_psStatus)
```

功能说明：机芯进入预备存款状态，指示灯开始闪烁提示用户投入纸币。实现验钞，退钞，钞票压入暂存区等功能。

参数说明见表 44。

表 44 BNA_Start() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 45。

表 45 BNA_Start() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

备注：可在子线程或主线程中启动，使纸币模块进入等待接收纸币的工作状态，程序将进入阻塞，除非成功接收纸币或在另外线程中调用BNA_Stop可以使该函数退出运行。

10.1.2.5 停止接收钞票

```
int BNA_Stop(tSTDDevReturn * p_psStatus);
```

功能说明：机芯停止接收钞票，指示灯熄灭。

参数说明见表 46。

表 46 BNA_Stop() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 47。

表 47 BNA_Stop() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.6 获取识别钞票信息

```
int BNA_GetCashInfo(tBnaCashInfo *p_psCashInfo, BYTE& p_byOperation,
tSTDDevReturn *p_psStatus)
```

功能说明：获取机芯接收的钞票信息。

参数说明见表 48。

表 48 BNA_GetCashInfo() 参数说明

序号	参数类型	参数名称	属性	说明
1	tBnaCashInfo *	p_psCashInfo	Out	返回钞票信息结构体，临时钱箱的信息
2	BYTE&	p_byOperation	Out	最后一张处理纸币的状态 0:没有钞票插入；

				1:无法识别的钞票退出到退钞口; 2:用户由退钞口把钞票(不能识别或者取消操作退出的)取走; 3:钞票放入又被强行拽出; 4:钞票被拉紧; 5:真钞处理完毕; 6:钞票拉紧并且超时被迫停止接收; 7:退钞口有钞票未取走超时; 8:入钞口至暂存区有钞票正在处理; 9:钱箱已经取走; 10:钱箱已满; 11:机芯进出口长时间阻塞(技术故障); 12:未知错误导致暂停服务。
3	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 49。

表 49 BNA_GetCashInfo() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

备注: 可在BNA_Start, 执行成功后, 调用该函数, 获取临时钱箱与最后一张纸币的处理信息。

- a) 对于 G&D 的纸币模块, p_byOperation 可以获取 0, 1, 3, 5;
- b) 对于 MEI 的纸币模块, p_byOperation 可以获取 0, 1, 3, 5;
- c) 对于 G&D 的纸币模块, p_byOperation 可以获取 0, 1, 5, 6, 7, 9, 10;
- d) 对于钱箱已满, 钱箱已取走, 机芯故障, 未知错误等其他错误, 可以通过 BNA_GetStatu 的硬件返回信息来判断获取。

10.1.2.7 压箱

Int BNA_Encash(tSTDDevReturn *p_psStatus);

功能说明: 将暂存器的纸币全部压入钱箱。

参数说明见表 50。

表 50 BNA_Encash() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 51。

表 51 BNA_Encash() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.8 退钞

Int BNA_Refund(tSTDDevReturn *p_psStatus);

功能说明: 将暂存器的纸币全部退出。

参数说明见表 52。

表 52 BNA_Refund() 参数说明

序号	参数类型	参数名称	属性	说明
1	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 53。

表 53 BNA_Refund() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.9 获取模块状态

Int BNA_GetStatus(tBnaStatus *p_psBnaStatus, tSTDDevReturn * p_psStatus)

功能说明：调用此接口可获取模块当前状态，详细信息查看返回结构体。

参数说明见表 54。

表 54 BNA_GetStatus() 参数说明

序号	参数类型	参数名称	属性	说明
1	tBnaStatus *	p_ psBnaStatus	Out	保存BNA模块状态
2	tSTDDevReturn *	p_ psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 55。

表 55 BNA_GetStatus() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

```
typedef struct _TAcceptorState
```

```
{
```

```
    BYTE    CurrentOCCode;
```

```
    char    CurrentCode[3];
```

```
    char    SequenceNr;
```

```
    BYTE    NrOfDocumentInintermediateStorage;
```

```
    Char    ResolutionMode;
```

```
    BYTE    ValidationMode;
```

```
    BYTE    WorkMode;
```

```
}TAcceptorState;
```

结构体返回的状态信息；因此在硬件返回信息字符串中内容格式如下：

[CurrentCode], [CurrentOCCode], [NrOfDocumentInintermediateStorage], [ResolutionMode], [SequenceNr], [ValidationMode], [WorkMode]。

10.1.2.10 获取钱箱 ID

Int BNA_GetBoxID(char * p_BoxID, tSTDDevReturn * p_psStatus)

功能说明：调用此接口可获取钱箱ID，详细信息查看返回结构体。

参数说明见表 56。

表 56 BNA_GetBoxID() 参数说明

序号	参数类型	参数名称	属性	说明
1	char*	p_BoxID	Out	获取钱箱ID
2	tSTDDevReturn *	p_ psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 57。

表 57 BNA_GetBoxID() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.11 设置模块充值或售票状态

```
enum BNA_BIZ_TYPE
{
    BNA_RECHARGE = 0,          //充值
    BNA_SELL_TICKET = 1        //售票
};
Int BNA_SetModuleBizStatus(enum BNA_BIZ_TYPE iMode, tSTDDevReturn *p_psStatus)
```

功能说明：设置业务流程处于充值或售票状态。
参数说明见表 58。

表 58 BNA_BIZ_TYPE() 参数说明

序号	参数类型	参数名称	属性	说明
1	iMode	enum BNA_BIZ_TYPE	In	0: 充值; 1: 售票
2	tSTDDevReturn *	p_psStatus	Out	保存BNA返回的逻辑错误码

返回值说明见表 59。

表 59 BNA_BIZ_TYPE() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.1.2.12 通信串口打开

```
int BNA_open(unsigned int uiCommPort, unsigned int uiBaudRate)
```

功能说明：调用此接口可打开低层串口通信。
参数说明见表 60。

表 60 BNA_open() 参数说明

序号	参数类型	参数名称	属性	说明
1	unsigned int	uiCommPort	In	串口通信号
2	unsigned int	uiBaudRate	In	波特率

返回值说明见表 61。

表 61 BNA_open() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	串口打开出错

10.1.2.13 通信串口关闭

```
int BNA_close()
```

功能说明：调用此接口可关闭低层串口通信。
返回值说明见表 62。

表 62 BNA_close() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	关闭错误

10.1.2.14 获取模块固件版本

```
int BNA_GetVersion(char* pVersion, tSTDDevReturn * p_psStatus);
```

功能说明：用于获取控制模块内的firmware版本号（如果模块有），目前只针对GRG的纸币模块有效。
参数说明见表 63。

表 63 BNA_GetVersion() 参数说明

序号	参数类型	参数名称	属性	说明
1	char*	pVersion	Out	固件版本（如果有）
2	tSTDDevReturn &	p_psStatus	Out	返回状态信息

返回值说明见表 64。

表 64 BNA_GetVersion() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	关闭错误

10.1.2.15 设置模块接口工作目录

函数名称：

void BNA_SetWorkPath(char* pchWorkPath)

功能说明：

用于设置模块接口的工作目录（模块驱动库所在的绝对路径），调用方在使用接口库其它接口之前设置。参数说明见表 65。

表 65 BNA_SetWorkPath() 参数说明

序号	参数类型	参数名称	属性	说明
1	char*	pchWorkPath	in	模块驱动库所在的绝对路径

10.2 纸币找零模块接口

10.2.1 结构体说明

10.2.1.1 返回状态结构体

```
typedef struct
{
    UINT uiType; /*错误类型：0-正常；1-警告；2-故障*/
    UINT uiErrorType; /*故障类型：0-正常；1-命令类错误；2-硬件故障；3-通道异常；4-钱箱故障*/
    int iHandle; /*处理方法：0-不处理 1-初始化 2-重发命令 3-停机休息 4-软件和硬件重新配置*/
    int iStdErrorCode; /*统一维护的错误代码*/
    UINT uiErrorCode; /*错误代码（由不同模块厂家提供）*/
    char cReverse[128]; /*预留信息*/
} tBNDDevReturn;
```

10.2.1.2 初始化张数信息

```
typedef struct
{
    UINT uiOutNumber;
    UINT uiRetractNumber;
} tBndInitNumInfo;
```

10.2.1.3 钱箱状态结构体


```
typedef struct
{
    unsigned char uzBoxStatus[MAX_BOXNUM];
    unsigned char ucNgBoxStatus;
} tBndBoxStatus;
```

10.2.1.4 设置钱箱结构体

```
typedef struct
{
    UINT uiBoxType;    //容器类型：0-废币箱 1-1元纸币箱，2-5元纸币箱，3-10
                      //元纸币箱，4-20元纸币箱，5-50元纸币箱，6-100元纸币箱*
    UINT uiBoxNo;      /**容器序号：0 ~ 6 其中： 0号废币箱 1号为5元找零箱，2号为1元找
零箱 */
    UINT uiBillValue; /**单位分：容器中所能存放纸币的面额，期中0表示不能确
                      定该容器的面额*/
    UINT uiBillNum;    /**容器中纸币的数量*/
    UINT uiBillMaxNum; /**容器中能存入纸币最大的数量*/
    UINT uiBoxStatus; /**容器的状态：0-正常；1-未安装；2-已满；3-将满；4-
                      以空；5-将空,该状态为模块硬件检测的状态，通过计数判
                      断的状态由上位机进行判断

} tCashBoxValueInfo;
```

10.2.1.5 模块容器配置信息结构体

```
typedef struct
{
    tCashBoxValueInfo pCashBoxValueInfo[MAX_BOXNUM];
    tCashBoxValueInfo stCashNGBoxValueInfo;;
} tBNDBillBoxInfo;
```

10.2.1.6 传感器状态结构体

```
typedef struct
{
    UINT uiSensorID; /**传感器ID*/
    UINT uiStausts; /**传感器状态：0-未遮挡；1-遮挡*/
} SensorStatus;
```

```
typedef struct
{
    UINT uiSensorNum; /**传感器数量*/
    SensorStatus* pSensorStatus;
} tBNDSensorStatus;
```

10.2.1.7 配钞结构体

```
typedef struct
{
    UINT uiAmount;          //总数
    UINT uiDispenseNumber[MAX_BOXNUM]; //每个钱箱需要出钱的张数
    UINT uiOutNumber[MAX_BOXNUM];      //实际每个钱箱出钱的张数（返回）
    UINT uiRejectNumber[MAX_BOXNUM];   //出钞过程中每个钱箱废钞回收的张数（返回）
} tBNDCchangeNum;
```

10.2.1.8 审计结构体

```
typedef struct
{
    UINT uiNGBoxAccInNum;        //累计进入废钞箱张数
    UINT uiBoxAccOutNum;         //累计出钞张数
}tBndAuditCashNum;
```

10.2.1.9 Rfid 信息结构体

```
typedef struct
{
    unsigned char uzCashBoxID[4] // 钱箱编号
    unsigned char uzOperateID[3] // 操作员ID
    unsigned char uzDeviceID[4]// 设备ID
    unsigned char ucCashBoxStatus// 钱箱位置状态
    unsigned char ucCashBoxOperateStatus// 操作后钱箱状态
    unsigned char ucBillType// 币种代码
    unsigned char usBillCnt// 数量
    unsigned int uiBillAmount// 纸币总金额
    unsigned char ucStationCode[2]// 车站编码
    unsigned char ucInstallPosit;// 安装位置
    unsigned int uiBlockOperateTime;// 块操作时间
    unsigned char ucOperateTime[7];// 最后操作时间
    unsigned short usCheckField;// 校验字段
}tBndRfidInfo;
```

10.2.2 接口函数

说明:

- 所有函数返回值只判断传入参数是否为空;
- 逻辑或物理错误还必须判断 tBNDDevReturn 的 iType 和其他成员变量来确定;
- 每个函数内部都有线程锁, 以保证标准接口函数都是串行运行。

10.2.2.1 通信串口打开

```
int BND_CommOpen(UINT uiCommPort, UINT uiBaudRate, tBNDDevReturn *pDevStatus)
```

功能说明: 调用此接口可打开底层串口通信。

参数说明见表 66。

表 66 BND_CommOpen() 参数说明

序号	参数类型	参数名称	属性	说明
1	unsigned int	uiCommPort	In	串口通信号
2	unsigned int	uiBaudRate	In	波特率
3	tBNDDevReturn	pDevStatus	Out	设备返回状态

返回值说明见表 67。

表 67 BND_CommOpen() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	串口打开通错

10.2.2.2 通信串口关闭

int BND_CloseComm(tBNDDevReturn *pDevStatus)

功能说明：调用此接口可关闭低层串口通信。

参数说明参见表 68。

表 68 BND_CommOpen() 参数说明

序号	参数类型	参数名称	属性	说明
1	tBNDDevReturn	pDevStatus	Out	设备返回状态

返回值说明见表 69。

表 69 BND_close() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	关闭错误

10.2.2.3 初始化

Int BND_Init(tBndInitNumInfo *pNumInfo, tBNDDevReturn *pDevStatus)

功能说明：初始化模块，检测模块各硬件、将模块恢复到初始状态等。

参数说明见表 70。

表 70 BND_Init() 参数说明

序号	参数类型	参数名称	属性	说明
1	tBndInitNumInfo *	pNumInfo	Out	退出和回收钱币信息
2	tBNDDevReturn	pDevStatus	Out	设备返回状态

返回值说明见表 71。

表 71 BND_Init() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.4 复位

int BND_Reset(tBndInitNumInfo *pNumInfo, tBNDDevReturn *pDevStatus)

功能说明：复位。

参数说明见表 72。

表 72 BND_Reset() 参数说明

序号	参数类型	参数名称	属性	说明
1	tBndInitNumInfo*	pNumInfo	Out	退出和回收钱币张数
2	tBNDDevReturn	pDevStatus	Out	设备返回状态

返回值说明见表 73。

表 73 BND_Reset() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.5 获取模块状态

函数名称：

Int BND_GetDevStatus(tBndBoxStatus *pBndBoxStatus, tBNDDevReturn *tBNDDevReturn)

功能说明：调用此接口可获取模块当前状态，详细信息查看返回结构体。
参数说明见表 74。

表 74 BND_GetDevStatus () 参数说明

序号	参数类型	参数名称	属性	说明
1	tBndBoxStatus	pBndBoxStatus	Out	返回钱箱状态
2	tBNDDevReturn *	tBNDDevReturn	Out	保存BND返回的逻辑错误码

返回值说明见表 75。

表 75 BND_GetDevStatus () 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.6 读取审计信息

函数定义：
int BND_GetAuditData(tBndAuditCashNum *pBndAuditCashNum, tBNDDevReturn *pDevStatus)
功能说明：
读取设备的审计信息。
参数说明见表 76。

表 76 BND_GetAuditData () 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tBndAuditCashNum *	pBndAuditCashNum	Out	容器配置信息
2	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 77。

表 77 BND_GetAuditData () 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.2.2.7 获取传感器状态

函数名称：
int BND_GetSensorStatus(tBNDSensorStatus *pSensorStatus, tBNDDevReturn *pDevStatus);
功能说明：获取传感器状态。
参数说明见表 78。

表 78 BND_GetSensorStatus () 参数说明

序号	参数类型	参数名称	属性	说明
1	tBNDSensorStatus *	pSensorStatus	Out	传感器状态
2	tBNDDevReturn &	pDevStatus	Out	返回状态信息

返回值说明见表 79。

表 79 BND_GetSensorStatus () 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	关闭错误

10.2.2.8 读取容器配置信息

函数定义：
int BND_GetBillBoxInfo(tBNDBillBoxInfo *pBillBoxInfo, tBNDDevReturn *pDevStatus);
功能说明：
读取钱箱结构体信息。
参数说明见表 80。

表 80 BND_GetBillBoxInfo() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tBNDBillBoxInfo *	pBillBoxInfo	Out	返回钱箱结构体信息
2	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值见表 81。

表 81 BND_GetBillBoxInfo() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.2.2.9 设置容器配置信息

函数定义：
int BND_SetBillBoxInfo(UINT uiSlotID, tBNDBillBoxInfo *pBillBoxInfo, tBNDDevReturn *pDevStatus);
功能说明：设置容器配置信息（设置纸币数量）
参数说明见表 82。

表 82 BND_SetBillBoxInfo() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UNIT	uiSlotID	In	槽位ID
2	tBNDBillBoxInfo *	pBillBoxInfo	In	容器配置信息
3	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值见表 83。

表 83 BND_SetBillBoxInfo() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.2.2.10 找零配钞

函数定义：
int BND_ChangeBill(tBNDChangeNum *pChangeNum, tBNDDevReturn *pDevStatus)
功能说明：找零配钞
参数说明见表 84。

表 84 BND_ChangeBill() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tBNDChangeNum *	pChangeNum	In	找零的纸币面额及数量信息
2	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 85。

表 85 BND_ChangeBill() 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.2.2.11 开始更换纸币找零箱

函数定义:

```
int BND_StartReplaceBox(UINT uiBoxType, UINT uiBoxNo, tBNDDevReturn *pDevStatus);
```

功能说明: 此接口用于更换纸币找零箱。

参数说明见表 86。

表 86 BND_StartReplaceBox() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UNIT	uiBoxType	In	指定纸币容器的类型
2	UINT	uiBoxNo	In	制定纸币容器的序号
3	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值见表 87。

表 87 BND_StartReplaceBox() 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1: 执行失败。

10.2.2.12 停止更换纸币找零箱

```
Int BND_StopReplaceBox(UINT uiBoxType, UINT uiBoxNo, tBNDDevReturn *pDevStatus)
```

功能说明: 停止更换纸币找零箱。

参数说明见表 88。

表 88 BND_StopReplaceBox() 参数说明

序号	参数类型	参数名称	属性	说明
1	UNIT	uiBoxType	In	指定纸币容器类型
2	UNIT	uiBoxNo	In	指定纸币容器序号
2	tBNDDevReturn *	p_psStatus	Out	保存BND返回的逻辑错误码

返回值说明见表 89。

表 89 BND_StopReplaceBox() 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.13 读取序列号

函数定义:

```
int BND_GetSerialNumber(unsigned char * pSn, tBNDDevReturn * pDevStatus);
```

功能说明: 此命令用于模块的序列号。

参数说明见表 90。

表 90 BND_GetSerialNumber() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	Unsigned char*	pSn	Out	模块的序列号, 长度32
2	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 91。

表 91 BND_GetSerialNumber () 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.2.2.14 读取固件编号

函数定义:

```
int BND_GetFWVerdion (unsigned char * pVersion, tBNDDevReturn * pDevStatus);
```

功能说明: 此命令用于模块的固件编号。

参数说明见表 92。

表 92 BND_GetFWVerdion () 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	Unsigned char*	pVersion	Out	模块的固件编号
2	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 93。

表 93 BND_GetFWVerdion () 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.2.2.15 设置模块工作模式

```
Int BND_SetWorkMode(char cMode, tBNDDevReturn *pDevStatus)
```

功能说明: 设置模块功能模式。

参数说明见表 94。

表 94 BND_SetWorkMode () 参数说明

序号	参数类型	参数名称	属性	说明
1	char	cMode	In	正常工作模式: 0x00; 维护模式: 0x01
2	tBNDDevReturn *	pDevStatus	Out	设备返回状态

返回值说明见表 95。

表 95 BND_SetWorkMode () 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.16 取消找零

```
Int BND_Cancel (tBNDDevReturn *p_psStatus);
```

功能说明: 取消找零操作。

参数说明见表 96。

表 96 BND_Cancel () 参数说明

序号	参数类型	参数名称	属性	说明
1	tBNDDevReturn *	p_psStatus	Out	保存BND返回的逻辑错误码

返回值说明见表 97。

表 97 BND_Cancel () 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.17 设置模块测试模式

Int BND_TestSetting(char cTestType, tBNDBillBoxInfo* pBillBoxInfo, tBNDSensorStatus* pSensorStatus, tBNDDevReturn *pDevStatus);

功能说明：设置模块测试模式。

参数说明见表 98。

表 98 BND_TestSetting () 参数说明

序号	参数类型	参数名称	属性	说明
1	Char	cTestType	In	测试类型
2	tBNDBillBoxInfo	pBillBoxInfo	Out	返回容器配置信息
3	tBNDSensorStatus	pSensorStatus	Out	返回传感器状态
4	tBNDDevReturn *	pDevStatus	Out	保存BND返回的逻辑错误码

返回值说明见表 99。

表 99 BND_TestSetting () 返回值说明

序号	返回值类型	返回值	说明
1	Int	0	成功
2	Int	1	传入参数为空

10.2.2.18 设置钱箱 RFID 信息

函数定义：

int BND_WriteRFIDInfo(UINT uiSlotID, tBndRfidInfo* pRFIDInfo, tBNDDevReturn *pDevStatus);

功能说明：写入钱箱RFID信息。

参数说明见表 100。

表 100 BND_WriteRFIDInfo() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UNIT	uiSlotID	In	槽位ID
2	tBndRfidInfo *	pRFIDInfo	In	设置RFID信息
3	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值见表 101。

表 101 BND_WriteRFIDInfo() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.2.2.19 读取钱箱 RFID 信息

函数定义：

int BND_ReadRFIDInfo(UINT uiSlotID, tBndRfidInfo* pRFIDInfo, tBNDDevReturn *pDevStatus);

功能说明：读取钱箱RFID信息。

参数说明见表 102。

表 102 BND_ReadRFIDInfo() 参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UNIT	uiSlotID	In	槽位ID
2	tBndRfidInfo *	pRFIDInfo	Out	设置RFID信息
3	tBNDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 103。

表 103 BND_ReadRFIDInfo() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.2.2.20 设置模块接口工作目录

函数名称：

void BND_SetWorkPath(char* pchWorkPath)

功能说明：

用于设置模块接口的工作目录（模块驱动库所在的绝对路径），调用方在使用接口库其它接口之前设置。

参数说明见表 104。

表 104 BND_SetWorkPath() 参数说明

序号	参数类型	参数名称	属性	说明
1	char*	pchWorkPath	in	模块驱动库所在的绝对路径

10.3 闸门及通行控制装置模块接口

10.3.1 接口汇总表

接口汇总表见表 105。

表 105 闸门及通行控制装置模块接口汇总

接口名称	接口说明
PCM_CommOpen	通讯建立
PCM_CommClose	通讯关闭
PCM_Init	初始化模块
PCM_Reset	模块复位
PCM_SetWorkMode	设置闸机工作模式
PCM_SetCtrlParam	设置参数
PCM_TicketValid	有效车票信号
PCM_GetDevStatus	读取模块状态
PCM_GetSensorState	读取传感器状态
PCM_ControlDoor	控制扇门
PCM_TestSetting	模块测试设置
PCM_GetVersion	获取模块固件版本

10.3.2 错误码汇总表

错误码汇总见表 106。

表 106 闸门及控制装置模块错误码汇总

错误类型（iType）	错误代码（iErrorCode）	说明
2	10000	发送串口命令失败
2	10001	扇门模式：【紧急模式】输入错误[输入值]！
2	10002	扇门模式：【维修模式】输入错误[输入值]！
2	10003	扇门模式：【状态模式】输入错误[输入值]！

2	10004	扇门模式：【工作模式】输入错误[输入值]！
2	10005	控制扇门输入错误[输入值]！
2	10006	测试类型输入错误[输入值]！
2	11000	串口不能使用！
2	11001	自动轮询失败！
2	11002	发送轮询命令失败！
2	20001	探知到机械故障
2	20002	探知到传感器故障

10.3.3 结构体说明

10.3.3.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId;    //模块厂家标识
    int iType;            //错误类型：0-正常；1-警告；2-故障
    int iErrorCode;       //错误代码
    char acReverse[4];    //预留信息
} tPcmDevReturn;
```

10.3.3.2 扇门模式结构体

```
typedef struct
{
    UCHAR ucRunMode; //工作模式：
    // 0，关闭模式；
    // 1，双向模式；
    // 2，单进模式；
    // 3，单出模式；
    // 4，进闸免检模式；
    // 5，出闸免检模式；
    // 6，进闸检票、出闸免检模式；
    // 7，出闸检票、进闸免检模式；
    // 8，进出闸都免检（自由通行）；
    UCHAR ucStandByMode; //闸门状态模式（低4位）：0x00，常闭模式；0x01，常开模式
    //儿童检测开关（高4位）：0x10，儿童检测开；0x00，儿童检测关
    UCHAR ucEmergency;   //紧急模式：0x00，正常；0x01，紧急
    UCHAR ucMaintenance; //维修模式：0x00，正常；0x01，维修
} tPcmOperationMode;
```

10.3.3.3 通行逻辑状态结构体

```
typedef struct
{
    UINT uiSide; //触发端：0，与方向无关；1，进闸端；2，出闸端；
    UINT uiPassLogicStatus; //通行逻辑状态：
    // 0x0000，正常通行逻辑
    // 0x0001，乘客进来； 0x0002，乘客离开
    // 0x0004，乘客滞留； 0x0008，乘客丢失
    // 0x0010，乘客尾随； 0x0020，无票进入
    // 0x0040，反向进闸； 0x0080，正常过闸
    // 0x0100，无乘客通过超时； 0x0200，开门受阻
    // 0x0400，强行打开扇门； 0x0800，冲撞扇门
```

```
} tPcmPassLogicStatus;
```

10.3.3.4 部件状态结构体

```
typedef struct
{
    UCHAR ucSensorStatus;    //传感器状态：0，正常；1，故障；
    UCHAR ucFlapDoorStatus; //闸门状态：0，正常；1，左闸门故障；2，右闸门故障；4，两个
    闸门都故障；
    UCHAR ucMotorStatus;     //马达状态：0，正常；1，故障；
    UCHAR ucMainStatus;      //维修门开关状态：00维修门关闭；01:维修门开（适用于维修门
    开关信号接入门机构的模块）
    UCHAR ucChildStatus;     //儿童检测开关状态：00儿童检测关闭；01:儿童检测打开
    UCHAR ucReverseStatus[7]; //预留状态
} tPcmComponentStatus;
```

10.3.3.5 模块状态结构体

```
typedef struct
{
    UCHAR ucOpModeState;      // 当前扇门模式
    UINT  unTotalEnNum;       // 进站端已授权总数量（断电清零）
    UINT  unTotalExNum;       // 出站端已授权总数量（断电清零）
    UCHAR ucStoredEnNum;      // 剩余未进闸有效乘客数量
    UCHAR ucStoredExNum;      // 剩余未出闸有效乘客数量
    tPcmPassLogicStatus tPassLogicState; // 通行状态
    tPcmComponentStatus tComponentState; // 组成部件状态
} tPcmModuStatus;
```

其中“ucOpModeState”（当前扇门模式）定义见表 107。

表 107 ucOpModeState 定义

bit	7	6	5	4	3	2	1	0
定义	闸门模式	维修模式	紧急模式		工作模式			

工作模式：定义参见“tOperationMode”。

紧急模式：0，无；1，通过命令进入；2，通过信号进入；3，通过命令和信号进入。

维修模式：0，无；1，进入。

闸门模式（Stand-by Mode）：0，常关模式；1，常开模式。

10.3.3.6 有效车票结构体

```
typedef struct
{
    UCHAR ucSide;           //刷卡端：1，进闸端；2，出闸端
    UCHAR ucQuatity;        //票数（大于0时，表示有效授权，扇门开门放行；等于0时表示无效授
    权，扇门阻挡乘客，提示非法票卡，还需点亮故障灯；当大于0时，ucLampBar1/ucLampBar2记录优惠灯
    的颜色）
    UCHAR ucLampBar1;       //优惠指示条1：0，不显示；1，显示（当ucQuatity>0时，记录票种灯
    的颜色，颜色定义参见“8.3.2 提示灯信号设置结构体”字段ucLampColor的定义）
    UCHAR ucLampBar2;       //优惠指示条2：0，不显示；1，显示
} tPcmValidTk;
```

10.3.3.7 控制参数结构体

```
typedef struct
```

```
{
    USHORT usHoldTimeout;    //单位：秒；授权超时时间，发出开门信号后无乘客进入，扇门
保持开启的时间（为0时不变）
    USHORT usLeaveTimeout;    //单位：毫秒；发出开门信号后，允许乘客在通道内滞留时间（为
0时不变）
    USHORT usClosingTimeout ; //单位：毫秒；旅客通过后门扇保持开启时段，即“两次连续进
入闸机保持门开的延迟时间”（为0时不变）
} tPcmCtrlParam;
```

10.3.3.8 版本信息结构体

```
typedef struct
{
    UCHAR pFirmwareVerInfo[32]; //固件版本信息，32 字节，不够后补0x00
    UCHAR pDriverVerInfo[32];    //驱动版本信息，32 字节，不够后补0x00
    UCHAR pHardwareVerInfo[32];  //硬件版本信息，32 字节，不够后补0x00
} tPcmVerInfo;
```

10.3.4 通讯建立

函数名称：int PCM_CommOpen(UINT uiCommPort, UINT uiBaudRate)
功能说明：建立串口通讯并进行设备的轮询以保证串口通讯的正常。
参数说明见表 108。

表 108 PCM_CommOpen() 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	uiCommPort	In	串口号
2	UINT	uiCommPort	In	波特率

返回值说明见表 109。

表 109 PCM_CommOpen() 返回值说明

序号	类型	说明
1	Int	0：执行成功；1，执行失败。

10.3.5 通讯关闭

函数名称：int PCM_CommClose()
功能说明：关闭串口通讯。
参数说明：无。
返回值说明见表 110。

表 110 PCM_CommClose() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.3.6 模块复位

函数名称：int PCM_Reset(tPcmDevReturn& tReturn)
功能说明：重新启动扇门模块的控制程序。
参数说明见表 111。

表 111 PCM_Reset() 参数说明

序号	参数类型	参数名称	属性	说明
----	------	------	----	----

1	tPcmDevReturn&	tReturn	in	返回状态信息
---	----------------	---------	----	--------

返回值说明见表 112。

表 112 PCM_Reset() 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.3.7 初始化

函数名称: int PCM_Init(tPcmDevReturn& tReturn)

功能说明: 初始化模块, 检查模块是否可用, 包括对扇门状态检测、传感器状态检测等。

参数说明见表 113。

表 113 PCM_Init() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmDevReturn&	tReturn	in	返回状态信息

返回值说明见表 114。

表 114 PCM_init() 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.3.8 设置闸机工作模式

函数名称: int PCM_SetWorkMode(tPcmOperationMode* pMode, tPcmDevReturn& tReturn)

功能说明: 设置闸机的各种工作模式。

参数说明见表 115。

表 115 PCM_SetWorkMode() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmOperationMode*	pMode	in	工作模式
2	tPcmDevReturn&	tReturn	out	返回状态信息

注: 对于方向指示器、顶棚向导标识控制的约定: 如果硬件中DIO模块不包含方向指示器和顶棚向导的控制接口, 则应根据本接口设置的工作模式, 自行控制方向指示器和顶棚向导(例如: 单进模式时, 进站方向通行指示器和顶棚显示绿色箭头, 出站方向显示红色叉)。

10.3.9 设置控制参数

函数名称: int PCM_SetCtrlParam(tPcmCtrlParam* pParam, tPcmDevReturn& tReturn)

功能说明: 设置闸机的各种工作参数。

参数说明见表 116。

表 116 PCM_SetCtrlParam() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmCtrlParam*	pParam	in	控制参数
2	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 117。

表 117 PCM_SetCtrlParam() 返回值说明

序号	类型	说明
1	int	0: 执行成功; 1, 执行失败。

10.3.10 有效车票投入信号

函数名称：int PCM_TicketValid(tPcmValidTk* pValidTk, tPcmDevReturn& tReturn)

功能说明：设置闸机的有效车票信息。

参数说明见表 118。

表 118 PCM_TicketValid() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmValidTk*	pValidTk	in	有效车票信息
2	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 119。

表 119 PCM_TicketValid() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.3.11 取模块状态

函数名称：int PCM_GetDevStatus(tPcmModuStatus& tModuStatus, tPcmDevReturn& tReturn)

功能说明：获取通行逻辑控制模块的实时状态。

参数说明见表 120。

表 120 PCM_GetDevStatus() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmModuStatus&	tModuStatus	in	模块状态
2	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 121。

表 121 PCM_GetDevStatus() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

10.3.12 取传感器状态

函数名称：int PCM_GetSensorStatus(UINT* pSensorNum, BYTE* pSensorStatus, UINT* pHeightSensorNum, BYTE* pHeightSensorStatus, BYTE* pSensorFaultState, BYTE* pHeightSensorFaultState, tPcmDevReturn& tReturn)

功能说明：获取所有通行传感器(通行传感器和身高传感器)的遮挡状态、故障状态。

参数说明见表 122。

表 122 PCM_GetSensorStatus() 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT*	pSensorNum	out	通行传感器数量
2	BYTE*	pSensorStatus	out	通行传感器遮挡状态，长度为pSensorNum个字节，每一个传感器状态占用1字节（0：未遮挡；1：遮挡）
3	UINT*	pHeightSensorNum	out	身高传感器数量
4	BYTE*	pHeightSensorStatus	out	身高传感器遮挡状态，长度为pSensorNum个字节，每一个传感器状态占用1字节（0：未遮挡；1：遮挡）
5	BYTE*	pSensorFaultState	out	通行传感器故障状态，长度为pSensorNum个字节，每一个传感器状态占用1字节（0：正常；1：故障）

6	BYTE*	pHeightSensorFaultState	out	身高传感器故障状态，长度为pHeightSensorNum个字节，每一个传感器状态占用1字节（0：正常；1：故障）
7	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 123。

表 123 PCM_GetSensorStatus() 返回值说明

序号	参数名称	类型	说明
1	tReturn	int	0：执行成功；1，执行失败。

10.3.13 控制扇门

函数名称：
int PCM_ControlDoor(BYTE ucCtrlType, tPcmDevReturn& tReturn)
功能说明：
用于强制控制闸门的打开与关闭，而不考虑当前的运行模式的逻辑控制。
参数说明见表 124。

表 124 PCM_ControlDoor() 参数说明

序号	参数类型	参数名称	属性	说明
1	BYTE	ucCtrlType	in	控制类型：0， 恢复到按工作模式的逻辑控制 1， 开扇门 2， 关扇门
2	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 125。

表 125 PCM_ControlDoor() 返回值说明

序号	类型	说明
1	int	0：执行成功；1，执行失败。

说明：
MCU 设置了强制打开、强制关闭后，一定要记得发送恢复命令。

10.3.14 模块测试设置

函数名称：
int PCM_TestSetting(BYTE ucTestType, tPcmDevReturn& tReturn)
功能说明：
用于测试扇门模块的部件，包括传感器测试、闸门测试、方向指示器测试。
参数说明见表 126。

表 126 PCM_TestSetting() 参数说明

序号	参数类型	参数名称	属性	说明
1	BYTE	ucTestType	in	测试类型：0， 停止当前测试 1， 闸门动作测试 2， 方向指示器测试 3， 传感器测试 4， 蜂鸣器测试
2	tPcmDevReturn&	tReturn	out	返回状态信息

返回值说明见表 127。

表 127 PCM_TestSetting() 返回值说明

序号	类型	说明
----	----	----

1	int	0：执行成功；1，执行失败。
---	-----	----------------

说明：

- a) 此命令只能在“维修模式下”使用，否则“无效命令”；
- b) 部件测试之间可以不发送“停止当前测试”而直接切换；
- c) 退出“维修模式”，自动停止测试。

10.3.15 获取模块固件版本

函数名称：
int PCM_GetVersion(tPcmVerInfo& tVersion, tPcmDevReturn& tReturn)
功能说明：
用于获取控制模块内的 firmware 版本号。
参数说明见表 128。

表 128 PCM_GetVersion() 参数说明

序号	参数类型	参数名称	属性	说明
1	tPcmVerInfo&	tVersion	out	模块固件版本信息

返回值说明见表 129。

表 129 PCM_GetVersion() 返回值说明

序号	参数名称	类型	说明
1	tReturn	int	0：执行成功；1，执行失败。

10.3.16 设置模块接口工作目录

函数名称：
void PCM_SetWorkPath(char* pchWorkPath)
功能说明：
用于设置模块接口的工作目录（模块驱动库所在的绝对路径），调用方在使用接口库其它接口之前设置。
参数说明见表 130。

表 130 PCM_SetWorkPath() 参数说明

序号	参数类型	参数名称	属性	说明
1	char*	pchWorkPath	in	模块驱动库所在的绝对路径

操作流程图见图 1。

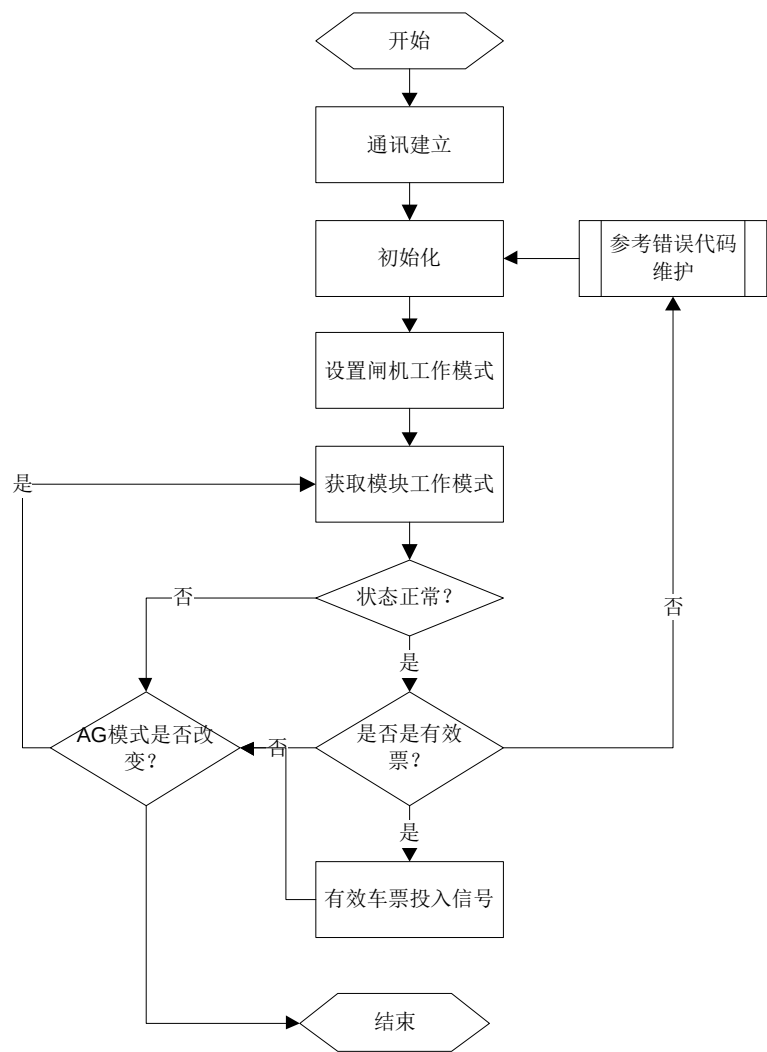


图 1 操作流程

10.4 车票发售硬件驱动接口（TIM）

10.4.1 车票发售控制模块接口

10.4.1.1 接口汇总表

接口汇总表见表 131。

表 131 接口汇总表

接口名称	接口说明
TIM_CommOpen()	通讯建立
TIM_Init()	模块初始化
TIM_Reset()	模块复位
TIM_GetVersion()	读取设备驱动版本信息
TIM_GetDevStatus()	读取设备状态
TIM_CardOut()	获取票卡
TIM_SendCard()	送出票卡
TIM_RetractCard()	回收票卡
TIM_GetAuditData()	读取审计累计值

接口名称	接口说明
TIM_GetTicketBoxRFIDInfo	读取票箱 RFID 信息
TIM_GetTicketBoxCountInfo	读取票箱数量信息
TIM_SetTicketBoxRFIDInfo	设置票箱 RFID 信息
TIM_SetTicketBoxCountInfo	设置票箱数量信息
TIM_ReplaceTBox()	更换票箱
TIM_GetSN()	读取序列号
TIM_CommClose()	关闭通讯
TIM_SetWorkMode()	设置模块工作模式
TIM_TestSetting()	模块测试设置

10.4.1.2 设备返回码的定义及其说明

设备返回码的定义及说明见表 132。

表 132 设备返回码定义及说明

故障类型	错误类型	错误代码	说明
命令类错误 1	2	41441	无效命令(没有调用串口打开时也返回该错误代码)
	2	41442	无效参数
	2	41484	设备忙
	2	41490	未知错
	2	41493	串口配置错
	2	41494	收到 NAK, 发送数据有错
	2	41495	接收到的数据有误
	2	41496	发送 ENQ 出错
命令类错误 1	2	41497	发送命令错
	2	41498	等待 ACK 超时
	2	41499	等待响应超时
	2	41420	警告码基址
	2	41440	错误码基址
	1	31000	命令错误
	2	31001	系统忙
	2	31002	继续处理可能异常
硬件故障 2	2	31003	参数错误
	2	41443	电子 ID1 故障
	2	41444	电子 ID2 故障
	2	41445	电子 ID3 故障
	2	41446	读卡箱 ID 失败
	2	41447	写卡箱 ID 失败
	2	41448	读设备型号失败
	2	41449	写设备型号失败
	2	41450	读 Log 失败
	2	41451	写 Log 失败
	2	41452	EEPROM 故障
	2	41474	读 flash

故障类型	错误类型	错误代码	说明
	2	41475	擦除 flash 芯片错
	2	41476	无该日期 Log
	2	41478	通道传感器故障
	2	41479	主马达堵转
	2	41480	换向器故障
	2	41485	无法检测到票箱的 RFID 卡
	2	41486	票箱 RFID 卡认证失败
	2	41487	读写票箱 RFID 卡参数错
	2	41488	读写票箱 SN 号错
	2	31004	硬件故障
	2	31006	RFID 故障
	2	31007	刮票器 1 没有压下
硬件故障 2	2	31008	刮票器 2 没有压下
	2	31009	刮票 1 失败
	2	31010	刮票 2 失败
	2	31011	票箱 1 传输失败
	2	31012	票箱 2 传输失败
	2	31015	废票箱舌档未打开
	2	31016	出票电机卡票
通道异常故障 3	2	31017	回收电机故障
	2	41453	有卡在出口位置堵塞错
	2	41454	有卡在读卡位置堵塞错
	2	41457	卡堵在通道中
	2	41458	摩卡 A 入口通道堵塞
	2	41459	摩卡 B 入口通道堵塞
	2	41472	回收箱满/有卡在回收位置堵塞错
	2	41477	感应区无卡
	2	41421	读卡位置无卡存在
票箱故障 4	2	31034	通道上有残留车票
	2	41455	A 卡箱摩卡失败
	2	41460	A 卡箱把手未升起
	2	41456	B 卡箱摩卡失败
	2	41461	B 卡箱把手未升起
	2	41462	两卡箱把手均未打开
	2	41463	发卡箱 A 盖未打开
	2	41464	发卡箱 B 盖未打开
	2	41465	两卡箱盖均未打开
	1	41466	A 卡箱空
	1	41467	B 卡箱空
	1	41468	两卡箱均空
	2	41469	A 卡箱不存在
	2	41470	B 卡箱不存在

故障类型	错误类型	错误代码	说明
票箱故障 4	2	41471	两卡箱均不存在
	2	41473	回收箱不存在
	1	41481	A 卡箱将空
	1	41482	B 卡箱将空
	1	41483	两卡箱都将空
	2	31013	票箱 1 舌档未打开
	2	31014	票箱 2 舌档未打开
	1	31022	票箱均未安装
	1	31023	废票箱安装检测
	1	31024	票箱均空
	1	31025	废票箱满检测-
	1	31026	票箱 1 安装检测
	1	31027	票箱 2 安装检测
	1	31028	票箱 1 车票有无检测
	1	31029	票箱 2 车票有无检测
	1	31030	票箱均将空
	1	31031	票箱一将空
	1	31032	票箱二将空
	1	31033	废票箱将满检测

10.4.1.3 传感器定义

传感器的定义及说明见表 133。

表 133 传感器定义

ID	传感器	说明
1	int	0: 执行成功; 1, 执行失败。
41001	电子 ID2	0:检测不到电子ID2 1:检测到电子ID2 在对电子ID2操作后此状态才更新。
41002	电子 ID3	0:检测不到电子ID3 1:检测到电子ID3 在对电子ID3操作后此状态才更新。
41003	EEPROM 状态传感器	0: EEPROM正常 1:EEPROM故障 在对EEPROM操作后此状态才更新
41004	BCSA	卡箱A进入通道传感器 0:正常1:故障
41005	BCSB	卡箱B进入通道传感器 0:正常1:故障
41006	RWPS	卡读写位置传感器 0:正常1:故障
41007	ECS	出口计数传感器 0:正常1:故障
41008	BAAES	票箱A将空检测传感器 0: 正常 1: 将空
41009	BAPS	票箱A存在传感器 0: 正常 1: 不存在
41010	BAES	票箱A空检测传感器 0: 正常 1: 票箱空
41011		票箱A编码器传感器 0: 正常 1: 故障

ID	传感器	说明
41012		票箱A进入通道传感器 0：正常 1：故障
41013		票箱A黑块传感器 0：正常 1：故障
41014	BBAES	票箱B将空检测传感器 0：正常 1：将空
41015	BBPS	票箱B存在传感器 0：正常 1：不存在
41016	BBES	票箱B空检测传感器 0：正常 1：票箱空
41017		票箱B编码器传感器 0：正常 1：故障
41018		票箱B进入通道传感器 0：正常 1：故障
41019		票箱B黑块传感器 0：正常 1：故障
41020	TRPS	票卡在读写区定位传感器 0：正常 1：故障
41021		出口计数传感器 0：正常 1：故障
41022	TSCS	检测票箱C换向器动作传感器 0：正常 1：故障
41023		通道编码器传感器 0：正常 1：故障
41024	BCPS	回收箱箱存在传感器 0：正常 1：不存在
41025	BCTPS	回收箱C内票卡满传感器 0：正常 1：票箱满
41026		对进入回收箱C的票卡进行计数传感器 0：正常 1：故障
41027	BACPS	检测票箱A上盖开闭状态 0：正常 1：未上盖
41028	BBCPS	检测票箱B上盖开闭状态 0：正常 1：未上盖
41029	BCCPS	检测票箱C上盖开闭状态 0：正常 1：未上盖
41030	TIS	检测票卡是否成功发售
41031	TAPS	检测票卡从票箱A挖出位置 0：正常 1：故障
41032	TABPS	检测票卡在票箱A、B之间的位置 0：正常 1：故障
41033	TBPS	检测票卡从票箱B挖出位置 0：正常 1：故障
41034	BAHPS	检测票箱A手柄位置 0：正常 1：故障
41035	BBHPS	检测票箱B手柄位置 0：正常 1：故障
41036	WAP1S	检测挖卡轮A位置状态 0：正常 1：故障
41037	WAP2S	检测挖卡轮A位置状态 0：正常 1：故障
41038	WBP1S	检测挖卡轮B位置状态 0：正常 1：故障
41039	WBP2S	检测挖卡轮B位置状态 0：正常 1：故障

ID	传感器	说明
		备用1传感器 0: 正常 1: 故障
		备用2传感器 0: 正常 1: 故障

传感器ID定义见表 134。

表 134 传感器 ID 定义

传感器 ID	说明
31000	票箱 2 舌档关闭检测
31001	废票箱将满检测-
31002	废票箱舌档关闭
31003	票箱一将空
31004	票箱二将空
31005	票箱 1 车票排出定位
31006	票箱 1 安装检测
31007	票箱 2 安装检测
31008	票箱 2 车票排出定位
31009	废票箱安装检测
31010	票箱 1 提供位置检测
31011	票箱 1 下限传感器
31012	票箱 2 下限传感器
31013	票箱 2 提供位置检测
31014	票箱 1 锁定爪传感器
31015	票箱 2 锁定爪传感器
31016	票箱 2 接近下限
31017	废票箱锁定爪传感器
31018	取出口抽取检测传感
31019	排出结束检测传感器
31020	废票回收结束检测
31021	IC 卡 R/W 停止位置
31022	票箱 1 提取完成
31023	票箱 2 提取完成
31024	票箱 1 提供部位置
31025	预留
31026	票箱 2 提供部位置
31027	预留
31028	清分舌档位置检测
31029	预留
31030	票箱 1 上端检测
31031	票箱 2 上端检测-
31032	票箱 1 车票有无检测
31033	票箱 2 车票有无检测
31034	废票箱满检测-
31035	票箱 1 舌档打开检测

传感器 ID	说明
31036	票箱 2 舌档打开
31037	票箱 1 接近下限
31038	废票箱舌档打开
31039	票箱 1 舌档关闭
31040	BOM 到位开关
31041	TVM 到位开关
31042	预留
31043	预留
31044	预留
31045	预留
31046	预留
31047	预留

10.4.1.4 结构体说明

10.4.1.4.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId; //模块厂家标识三星0x02、广电0x04、中软0x06、欧姆龙0x05;
    UINT uiModuleType; //模块类型标识 0x06车票发售模块
    UINT uiType; //错误类型: 0-正常; 1-警告; 2-故障
    UINT uiErrorType; //故障类型 0-正常; 1-命令类错误; 2-硬件故障; 3-通道异常故障; 4-票箱故障; 5-回收模块卡票
    UINT uiErrorCode; //错误代码 (由不同模块厂家提供)
    BYTE cReverse[128]; //预留信息
} tTIMDevReturn;
```

10.4.1.4.2 票箱数据信息结构体

```
typedef struct
{
    UINT uiCassetteACardNum; // 票箱A 卡片数量
    UINT uiCassetteBCardNum; // 票箱B 卡片数量
    UINT uiRejectCassetteNum; // 废票箱卡片数量
} tTIMCardNum;
```

10.4.1.4.3 模块状态结构体

```
typedef struct
{
    bool bCardInReadArea; // 是否有卡在读卡区, TRUE: 有卡, FALSE: 无卡
    char cCassetteAStatus; // A 票箱状态, 0:正常, 1:将空, 2:全空, 3:未安装, 4:安装不到位, 5:票箱故障
    char cCassetteBStatus; // B 票箱状态, 0:正常, 1: 将空, 2:全空, 3:未安装, 4:安装不到位, 5:票箱故障
    char cRejectCassetteStatus; // 废票箱状态, 0:正常1:将满, 2:满, 3:未安装, 4:安装不到位, 5:票箱故障
    UINT uiSensorCount; //传感器数量
} Struct
```

```
{
    UINT uiSensotID; //传感器ID
    BYTE btSensorStatus; // 传感器状态(0:正常 1:故障 3:未遮挡 4: 遮挡 0xff: 不存在
该传感器)
    } tSensorType[256]; // 实际使用范围从tSensorType[0]--- tSensorType[uiSensorCount-1]
    tTIMCardNum tCardNum; // 票箱票卡数量
    char cReserve[256]; // 保留信息
} tTIMModuleStatus;
```

10.4.1.4.4 审计信息结构体

```
typedef struct
{
    UINT wCassetteAAuditCardNum; // 票箱A累计发卡数
    UINT wCassetteBAuditCardNum; // 票箱B累计发卡数
    UINT wRejectCassetteAuditNum; // 废票箱卡片数量
} tTIMAuditCardNum;
```

10.4.1.4.5 电子标签信息结构体

```
typedef struct
{
    unsigned char ucBoxID[4]; // 票箱编号
    unsigned char ucOperateID[3]; // 操作员ID
    unsigned char ucDeviceID[4]; // 设备ID
    unsigned char ucBoxStatus; // 票箱位置状态
    unsigned char ucBoxOperateStatus; // 操作后票箱状态
    unsigned char ucCardProviderID[4]; // 票卡发行商ID
    unsigned char ucCardType; // 票卡物理类型
    unsigned char ucTicketClass[2]; // 车票产品种类
    unsigned char ucPreValueType; // 预赋值属性
    unsigned char ucDerivedID; // 衍生产品ID
    unsigned char ucCardNum[2]; // 票卡数量
    unsigned char ucStationCode[2]; // 车站编码
    unsigned char ucInstallPosit; // 安装位置
    unsigned char ucOperateTime[7]; // 最后操作时间
    unsigned char ucReserve[12]; // 预留字段
} tTIMRFIDInfo;
```

10.4.1.5 通讯建立

函数定义：
int TIM_CommOpen(UINT uiCommPort, UINT uiBaudRate);
功能说明：
建立串口通信，如参数设置错误或串口被占用，返回错误码。
参数说明见表 135。

表 135 通讯建立参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UINT	uiCommPort	In	端口号
2	UINT	uiBaudRate	In	波特率

返回值说明见表 136。

表 136 通讯建立返回值说明

类型	说明
int	0: 执行成功; 1, 执行失败。

注意事项:

执行其他操作前, 必须先设置串口。

如果该接口返回失败, 请检查串口ID或者设备逻辑名是否正确以及串口设备是否正确等。

10.4.1.6 模块初始化

函数定义:

```
int TIM_Init(char cClearMode, UINT* uiRetractNum, tTIMModuleStatus* pModuleStatus,
tTIMDevReturn* pDevStatus);
```

功能说明:

此接口用于设备的初始化, 检查通道、托盘等硬件是否正常, 并在初始化过程中发卡通道中的卡(清理通道), 如果正常则处于待机状态, 允许进一步操作。如果不正常则报错, 需要排除故障后才能工作。

注: 当设备不支持从退票口退票功能时, 将自动清理到废票箱。

参数说明见表 137。

表 137 模块初始化参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cClearMode	In	车票清理方式: 1、从退票口退票 2、清理到废票箱
2	UINT*	uiRetractNum	Out	本次操作处理的卡片数
3	tTIMModuleStatus*	pModuleStatus	Out	返回模块状态信息
4	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 138。

表 138 模块初始化返回值说明

类型	说明
int	0: 执行成功; 1: 执行失败。

可能出现的错误见表 139。

表 139 模块初始化可能出现的错误说明

可能出现的错误	错误类型	推荐处理措施
废票箱不存在错误	错误严重	停机维护
废票箱满/有卡在回收位置堵塞错	严重	停机维护
主马达堵转	严重	重试
出口位置有卡堵塞	严重	调用“清理通道”接口
卡读写位置有卡堵塞	严重	调用“清理通道”接口
通道内有卡堵塞	严重	调用“清理通道”接口
换向器故障	严重	调用“清理通道”接口
有卡在通道入口位置堵塞错	严重	调用“清理通道”接口
通道传感器故障	严重	停机维护

注意事项:

通道传感器包括: A 通道入口、B 通道入口、出口计数、读写卡位置传感器。

设备开机或复位后需要调用本函数。

10.4.1.7 模块复位

函数定义:

```
int TIM_Reset(char cClearMode, UINT* uiRetractNum, tTIMModuleStatus* pModuleStatus,
tTIMDevReturn* pDevStatus);
```

功能说明:

重新启动发售模块的控制程序。
参数说明见表 140。

表 140 模块复位参数说明

序号	类型	说明	序号	类型
1	char	cClearMode	In	车票清理方式： 1、从退票口退出 2、清理到废票箱
2	UINT*	uiRetractNum	Out	本次操作回收的卡片数
3	tTIMModuleStatus*	pModuleStatus	Out	返回模块状态信息
4	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值见表 141。

表 141 模块复位返回值说明

类型	说明
int	0：执行成功；1，执行失败。

注意事项：
在设备故障或从维修模式退出时（对于支持维修模式的情况），执行本函数。

10.4.1.8 读取设备驱动版本信息

函数定义：
int TIM_GetVersion(char cFirmWareVersion[32], char cDriverVersion[32], tTIMDevReturn *pDevStatus);
功能说明：
获取设备驱动版本号、固件版本号(如发售模块的ID等)、状态信息。
参数说明见表 142。

表 142 读取设备驱动版本信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cFirmWareVersion[32]	Out	返回固件版本号
2	char	cDriverVersion[32]	Out	返回驱动版本号
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 143。

表 143 读取设备驱动版本信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.4.1.9 读取设备状态

函数定义：
int TIM_GetDevStatus(tTIMModuleStatus* pModuleStatus, tTIMDevReturn *pDevStatus);
功能说明：
此接口用于获取读写区是否有卡，票箱状态、票箱数量以及各传感器实时状态。
参数说明见表 144。

表 144 读取设备状态参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTIMModuleStatus *	pModulesStatus	Out	模块的状态信息
2	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 145。

表 145 读取设备返回值

类型	说明
int	0: 执行成功; 1, 执行失败。

可能出现的错误:
见设备返回码的定义及其说明。
注意事项: 无。

10.4.1.10 获取车票

函数定义:

```
int TIM_CardOut(char cBoxNo, tTIMModuleStatus * pModuleStatus, UINT* uiTransportNum,
tTIMDevReturn * pDevStatus);
```

功能说明:

此命令用于将卡从票箱中提出送到读卡器处停止。

检测模块各硬件、参数是否正确, 根据参数控制发卡, 主要内容包括:

- 检测发卡参数的合法性;
- 将车票从指定票箱中传输到读卡器位置停止下来。

参数说明见表 146。

表 146 获取车票参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cBoxNo	In	票箱号: 1: A 票箱; 2: B 票箱。
2	tTIMModuleStatus *	pModuleStatus	Out	返回模块状态信息
3	UINT*	uiTransportNum	Out	获取的车票数量
4	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 147。

表 147 获取车票参数返回值说明

类型	说明
int	0: 执行成功; 1, 执行失败。

可能出现的错误说明见表 148。

表 148 获取车票参数可能出现的错误

可能出现的错误	错误类型	推荐处理措施
废票箱不存在	错误严重	停机维护
废票箱满/有卡在回收位置堵塞错	严重	停机维护
主马达堵转	严重	重试
出口位置有卡堵塞	严重	调用“清理通道”接口
卡读写位置有卡堵塞	严重	调用“清理通道”接口
通道内有卡堵塞	严重	调用“清理通道”接口
换向器故障	严重	调用“清理通道”接口
有卡在通道入口位置堵塞错	严重	调用“清理通道”接口
A/B 票箱盖没有打开	严重	停机维护
A/B 把手没有升起--票箱黑块传感器	严重	停机维护
票箱 A/B 空	严重	停机维护
发票箱 A/B 摩不出卡错	严重	重试
通道传感器故障	严重	停机维护

注意事项:

通道传感器故障: 可能坏的传感器有A 通道入口、B 通道入口、出口计数、读写卡位置传感器。

10.4.1.11 送出车票

函数定义：

```
int TIM_SendCard(tTIMModuleStatus* pModuleStatus, UINT* uiTransportNum, tTIMDevReturn* pDevStatus);
```

功能说明：

此命令用于将车票处理单元天线板位置的车票送到出口。

参数说明见表 149。

表 149 送出车票参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTIMModuleStatus*	pModuleStatus	Out	返回模块状态信息
2	UINT*	uiTransportNum	Out	本次操作发送的卡片数
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 150。

表 150 送出车票返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误见表 151。

表 151 送出车票参数可能出现的错误

可能出现的错误	错误类型	推荐处理措施
读写卡位置无卡警告	一般警告	无需处理
主马达堵转	严重	重试
卡读写位置有卡堵塞	严重	调用“清理通道”接口

10.4.1.12 回收车票

函数定义：

```
int TIM_RetractCard(tTIMModuleStatus* pModuleStatus, UINT* uiRetractNum, tTIMDevReturn* pDevStatus);
```

功能说明：

此命令用于将车票处理单元天线板位置的车票回收到废票箱。

参数说明见表 152。

表 152 回收车票参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTIMModuleStatus *	pModuleStatus	Out	返回模块状态信息
2	UINT	uiRetractNum	Out	回收车票数量
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 153。

表 153 回收车票返回值说明

类型	参数名称	说明
int	tReturn	0：执行成功；1，执行失败。

可能出现的错误见表 154。

表 154 回收车票可能出现的错误

可能出现的错误	错误类型	推荐处理措施
读写卡位置无卡警告	一般警告	无需处理
废票箱不存在	错误严重	停机维护
废票箱满/有卡在回收位置堵塞错	严重	停机维护
主马达堵转	严重	重试
出口位置有卡堵塞	严重	调用“清理通道”接口

卡读写位置有卡堵塞	严重	调用“清理通道”接口
通道内有卡堵塞	严重	调用“清理通道”接口
换向器故障	严重	调用“清理通道”接口

注意事项：

无。

pDevStatuspDevStatus返回状态信息

10.4.1.13 读取审计信息

函数定义：

```
int TIM_GetAuditData(tTIMAuditCardNum* pAuditCardNum, tTIMDevReturn * pDevStatus);
```

功能说明：

读取设备的审计信息。

参数说明见表 155。

表 155 读取审计信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTIMAuditCardNum*	pAuditCardNum	Out	读出的审计信息，包括：A、B 票箱累计发卡数和废票箱累计回收卡数
2	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 156。

表 156 读取审计信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：

无。

注意事项：

A、B 票箱累计发卡数：指的是从票箱发出、并且送卡成功的次数。即总装入的卡数是A、B 累计发卡数和废票箱回收卡数的和。

10.4.1.14 读取票箱电子标签信息

函数定义：

```
int TIM_GetTicketBoxRFIDInfo(char cTicketBoxNo, tTIMRFIDInfo* pRFIDInfo, tTIMDevReturn *pDevStatus);
```

功能说明：

读取票箱电子标签信息。

电子标签信息中的密钥不需要处理。

参数说明见表 157。

表 157 读取票箱电子标签信息

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1：票箱 A 2：票箱 B 3：废票箱
2	tTIMRFIDInfo*	pRFIDInfo	Out	返回电子标签信息
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值见表 158。

表 158 读取票箱电子标签信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.15 读取票箱数量信息

函数定义：
int TIM_GetTicketBoxCountInfo(char cTicketBoxNo, tTIMCardNum * pCardNum, tTIMDevReturn *pDevStatus);
功能说明：
读取票箱数量信息。
参数说明见表 159。

表 159 读取票箱信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1：票箱 A 2：票箱 B 3：废票箱
2	tTIMCardNum *	pCardNum	Out	返回票箱数量信息
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值见表 160。

表 160 读取票箱信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.16 设置票箱电子标签信息

函数定义：
int TIM_SetTicketBoxRFIDInfo(char cTicketBoxNo, tTIMRFIDInfo* pRFIDInfo, tTIMDevReturn *pDevStatus);
功能说明：
写入票箱电子标签信息。
参数说明见表 161。

表 161 设置票箱电子标签信息

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1：票箱 A 2：票箱 B 3：废票箱
2	tTIMRFIDInfo*	pRFIDInfo	In	设置电子标签信息

3	tTIMDevReturn*	pDevStatus	Out	返回状态信息
---	----------------	------------	-----	--------

返回值见表 162。

表 162 读取票箱电子标签信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.17 设置票箱数量信息

函数定义：
int TIM_SetTicketBoxCountInfo(char cTicketBoxNo, tTIMCardNum* pCardNum, tTIMDevReturn *pDevStatus);
功能说明：
写入票箱信息。
参数说明见表 163。

表 163 设置票箱数量信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱： 1：票箱 A； 2：票箱 B； 3：废票箱。
2	tTIMCardNum*	pCardNum	In	设置票箱数量信息
3	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 164。

表 164 设置票箱数量信息返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.18 更换票箱

函数定义：
int TIM_ReplaceTBox(char cBoxNo, char cBoxMode, tTIMDevReturn* pDevStatus);
功能说明：
此接口用于更换票箱。
在取出票箱前，应使用“取票箱”方式先调用本命令；在放入票箱后，应使用“装票箱”方式调用本命令。
“取票箱”操作，是为了将票箱当前的车票数量写入票箱的电子标签 里保存（是为了设定要更换的票箱，再调用设置票箱信息命令）。“装票箱”操作，是为了将票箱电子标签 里的车票数量读取出来（是为了设定要更换的票箱，再调用读取票箱信息命令）。
参数说明见表 165。

表 165 更换票箱参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cBoxNo	In	票箱号： 0：更换所有票箱 1：更换 A 票箱； 2：更换 B 票箱； 3：更换废票箱。
2	char	btBoxMode	In	1：取票箱； 2：装票箱。
3	tTIMDevReturn *	pDevStatus	Out	返回状态信息

返回值见表 166。

表 166 更换票箱返回值说明

类型	说明
int	0：执行成功；1：执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.19 读取序列号

函数定义：
Int TIM_GetSN(char* cModuleSN, tTIMDevReturn *pDevStatus);
功能说明：
此命令用于读出设备的序列号。
参数说明见表 167。

表 167 读取序列号参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char*	cModuleSN	Out	模块的序列号，长度 20
2	tTIMDevReturn*	pDevStatus	Out	返回状态信息

返回值说明见表 168。

表 168 读取序列号返回值说明

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.4.1.20 关闭通讯

函数定义：
int TIM_CommClose();
功能说明：
关闭通讯。
参数说明：
无。
返回值说明见表 169。

表 169 关闭通讯返回值说明

类型	说明
----	----

int	0：执行成功；1：执行失败。
-----	----------------

可能出现的错误：
无。
注意事项：
无。

10.4.1.21 设置模块功能模式

函数定义：
int TIM_SetWorkMode(char cMode, tTIMDevReturn* pDevStatus);
功能说明：
设置模块为正常工作或维护模式。
参数说明见表 170。

表 170 设置模块功能模式参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cMode	In	正常工作模式：0x00 维护模式：0x01
2	tTIMDevReturn&	pDevStatus	Out	返回状态信息

返回值说明见表 171。

表 171 设置模块功能模式返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.4.1.22 设置模块测试

函数定义：
int TIM_TestSetting(char cTestType, tTIMModuleStatus* pModuleStatus, tTIMDevReturn* pDevStatus);
功能说明：
用于测试模块的部件，包括传感器测试、马达测试、翻板测试等。
参数说明见表 172。

表 172 设置模块测试参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTestType	In	测试类型： 0. 停止当前测试 1. 废票箱与正常出票传输切换翻板测试（如果含有） 2. 2 个票箱间出票传输切换翻板测试（如果含有） 3. 传感器测试（含各类到位触点和光电管） 4. A 票箱刮票马达测试 5. B 票箱刮票马达测试 6. A 票箱出票马达测试（如果含有） 7. B 票箱出票马达测试（如果含有） 8. 出票马达测试 9. A 票箱连续出票测试 10. B 票箱连续出票测试 11. 2 个票箱连续出票测试 12. A 票箱连续出卡测试 13. B 票箱连续出卡测试 14. A 票箱升降装置上升测试 15. A 票箱升降装置下降测试 16. B 票箱升降装置上升测试 17. B 票箱升降装置下降测试

				18. 出票口 LED 指示 1（如果含有） 19. 出票口 LED 指示 2（如果含有） 20. 出票口 LED 指示 3（如果含有） 21. 模块控制电路板上故障指示器测试（如果含有） 22. 模块控制电路板上蜂鸣器测试（如果含有） 23. A 票箱电子标签读测试 24. B 票箱电子标签读测试 25. 废票箱电子标签读测试（如果含有） 26. A 票箱电子标签写测试 27. B 票箱电子标签写测试 28. 废票箱 电子标签写测试（如果含有） 29-100 保留 101-255 各模块厂商自定义
2	tTIModuleSta tus*	pModuleStat us	Out	返回模块状态信息
3	tTIMDevReturn &	pDevStatus	Out	返回状态信息

补充说明：
此命令只能在“维修模式下”使用，否则“无效命令”；
部件测试之间可以不发送“停止当前测试”而直接切换；
退出“维修模式”，自动停止测试；
返回值说明见表 173。

表 173 设置模块测试返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.4.2 推荐操作流程

密钥采用明文方式在本动态库中写成固定值，与上位不传送密钥信息。
推荐的操作流程见图 2。

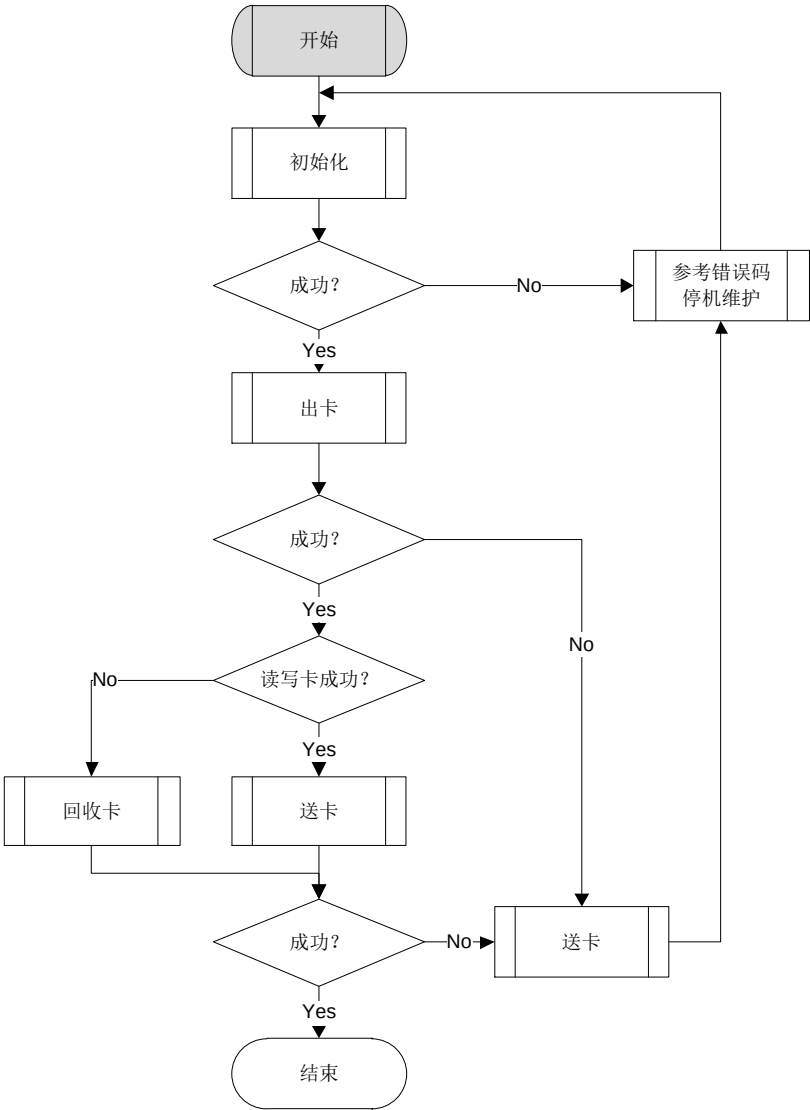


图 2 推荐操作流程

10.5 车票回收模块驱动接口（TAM）

10.5.1 车票回收模块接口

10.5.1.1 接口汇总表

接口汇总表见表 174。

表 174 接口汇总

接口名称	接口说明
TAM_CommOpen()	通讯建立
TAM_Init()	模块初始化
TAM_Reset()	模块复位
TAM_GetVersion()	读取设备驱动版本信息
TAM_GetDevStatus()	读取设备状态
TAM_SetServiceMode()	设置接收模式
TAM_AcceptCard()	回收车票
TAM_ReturnCard()	退出车票

TAM_GetAuditData	读取审计累计值
TAM_GetTicketBoxInfo ()	读取票箱信息
TAM_SetTicketBoxInfo ()	设置票箱信息
TAM_ReplaceTBox ()	更换票箱
TAM_GetSN ()	读取序列号
TAM_CommClose()	关闭通讯
TAM_SetWorkMode()	设置模块工作模式
TAM_TestSetting()	模块测试设置
TAM_GetCardInReadArea()	获取读卡区是否有车票

10.5.1.2 设备返回码的定义及其说明（包括状态码和错误码）

设备返回的定义及其说明见表 175。

表 175 设备返回码的定义及其说明

故障类型	错误类型	错误代码	说明
命令类错误 1	2	41521	警告错无效的输入参数
	2	41578	未知错误
	2	41590	无 ACK 应答
	2	41591	ACK 应答错
	2	41592	包格式错
	2	41593	串口配置错
	2	41594	初始化 COM 组件失败
	2	41595	通信超时
	1	31000	命令错误
	2	31001	系统忙
	2	31002	继续处理可能异常
	2	31003	参数错误
硬件故障 2	2	31005	附加数据异常
	2	41541	换向器故障或换向器传感器故障
	2	41542	SDS 传感器故障
	2	41543	CIS 传感器故障
	2	41544	RPS 传感器故障
	2	41545	CAES 传感器故障
	2	41546	CCAS 传感器故障
	2	41547	DAUS 传感器故障
	2	41548	CBES 传感器故障
	2	41549	CCBS 传感器故障
	2	41550	TAPS 传感器故障
	2	41551	TBPS 传感器故障
	2	41552	COAS 传感器故障
	2	41553	COBS 传感器故障
	2	41554	TAPS 传感器故障
	2	41555	TBPS 传感器故障
	2	41556	CANS 传感器故障
	2	41557	CAFS 传感器故障
	2	41558	CBNS 传感器故障
	2	41559	CBFS 传感器故障

故障类型	错误类型	错误代码	说明
	2	41560	TBAS 传感器故障
	2	41561	CEAS 传感器故障
	2	41562	TBBS 传感器故障
	2	41563	CEBS 传感器故障
	2	41576	读卡箱电子标签错
	2	41577	写卡箱电子标签错
	2	31004	硬件故障
	2	31006	电子标签通讯故障
	2	31009	传输电机运行不到位
通道异常故障 3	2	31015	分配 1 通过前传感器检测失败
	2	31016	回收箱不到位
	2	31027	票箱 A 回收结束传感器遮光
	2	31028	票箱 B 回收结束传感器遮光
	2	31029	废票箱回收结束传感器遮光
	2	41522	读卡位置无卡存在
	2	41540	车票入口闸门故障
	2	41564	CIS 传感器位置卡票
	2	41565	CIS 到 RPS 之间卡票
	2	41566	通道卡票
	2	31007	未传输到读写卡停止位置
	2	31008	未传输到读卡位置
	2	31012	车票从读写卡到回收箱传输失败

表 168 设备返回码的定义及其说明（续）

故障类型	错误类型	错误代码	说明
	2	31013	废卡回退到读卡器停止位置失败
	2	31014	废卡从读卡器停止位置到废票箱失败
票箱故障 4	2	41523	票箱 A 不存在
	2	41524	票箱 B 不存在
	1	41525	票箱 A 将满
	1	41526	票箱 A 满
	1	41527	票箱 B 将满
	1	41528	票箱 B 满
	2	41567	卡箱 A 盖没打开
	2	41568	卡箱 B 盖没打开
	2	41569	托盘 A 不能上升到位
	2	41570	托盘 A 不能下降到位
	2	41571	托盘 B 不能上升到位

	2	41572	托盘 B 不能下降到位
	2	41573	回收卡错, 卡箱 A 满
	2	41574	回收卡错, 卡箱 B 满
	1	31010	票箱 A 未安装
	1	31011	票箱 B 未安装
	2	31017	票箱 A 提升超时, 没有提升到位
	2	31018	票箱 A 下降超时, 没有下降到位
	2	31019	票箱 B 提升超时, 没有提升到位
	2	31020	票箱 B 下降超时, 没有下降到位
	2	31021	票箱 A 解锁前未关舌档
	2	31022	票箱 B 解锁前未关闭舌档
	2	31023	废票箱解锁前未关闭舌档
	2	31024	票箱 A 回收前未锁定
	2	31025	票箱 B 回收前未锁定
	2	31026	废票箱回收前未锁定
	2	31030	废票箱未安装
	2	31031	票箱 A 盖子未打开
	2	31032	票箱 B 盖子未打开
	2	31033	废票箱盖子未打开
	1	31039	票箱均未安装
	1	31040	票箱均满
	1	31041	废票箱满检测-

表 168 设备返回码的定义及其说明 (续)

故障类型	错误类型	错误代码	说明
	1	31042	票箱 A 已满
	1	31043	票箱 B 已满
	1	31044	票箱均将满
	1	31045	票箱一将满
	1	31046	票箱二将满
	1	31047	废票箱将满检测

10.5.1.3 传感器定义

传感器定义如表 176。

表 176 传感器定义

传感器 ID	传感器	说明
41000	TGS	检测票口有车票插入传感器 0: 无卡 1: 有卡
41001	TRPS	检测车票是否在读卡区传感器 0: 无卡 1: 有卡
41002	BATP	检测车票将进入票箱 A 传感器

传感器 ID	传感器	说明
		0: 没档 1: 挡住
41003	BBTP	检测车票将进入票箱 B 传感器 0: 没档 1: 挡住
41004	BACS	进入票箱 A 计数传感器 0: 没档 1: 挡住
41005	BBCS	进入票箱 B 计数传感器 0: 没档 1: 挡住
41006	TSAS	检测票箱 A 换向器动作传感器 0: down 1: up
41007	TSBS	检测票箱 B 换向器动作传感器 0: down 1: up
41008	TGPS	检测入口闸门动作传感器 0: 关闭 1: 打开
41009	BAFS	检测票箱 A 满传感器 0: 正常 1: 满
41010	BBFS	检测票箱 B 满传感器 0: 正常 1: 满
41011	BCCS	检测票箱 C 满传感器 0: 正常 1: 满
41012	UCCS	废票箱计数传感器 0: 正常 1: 故障

表 169 传感器定义（续）

传感器 ID	传感器	说明
41013	RES	检测废票箱存在传感器 0: 正常 1: 不存在
41014	TAPS	托盘 A 定位传感器
41015	TBPS	托盘 B 定位传感器
41016	BAAFS	检测票箱 A 将满传感器 0: 正常 1: 将满
41017	BBAFS	检测票箱 B 将满传感器 0: 正常 1: 将满
41018	BAPS	检测票箱 A 存在传感器 0: 正常 1: 不存在
41019	BBPS	检测票箱 B 存在传感器 0: 正常 1: 不存在
41020	BACPS	检测票箱 A 上盖开闭状态传感器 0: 正常 1: 未上盖
41021	BBCPS	检测票箱 B 上盖开闭状态传感器 0: 正常 1: 未上盖
41022	TBAS	检测底部托块 A 存在
41023	TBBS	检测底部托块 B 存在
41024	BCCPS	检测票箱 C 上盖开闭状态传感器 0: 正常 1: 未上盖
41025	BCIR	对票箱 C 的 ID 卡进行读写操作

传感器ID定义如表 177。

表 177 传感器 ID 定义

传感器 ID	传感器名称	传感器说明
31000	PH01	票箱 B 舌档关闭检测
31001	PH02	预留

传感器 ID	传感器名称	传感器说明
31002	PH03	票箱 A 舌档关闭检测
31003	PH04	票箱 B 下限传感器
31004	PH05	票箱 B 将满检测传感器
31005	PH06	预留
31006	PH07	票箱 A 安装检测
31007	PH08	票箱 B 安装检测
31008	PH09	预留
31009	PH10	废票箱安装检测
31010	PH11	预留
31011	PH12	票箱 A 将满检测传感器
31012	PH13	票箱 A 下限传感器

表 170 传感器 ID 定义（续）

传感器 ID	传感器名称	传感器说明
31013	PH14	预留
31014	PH15	票箱 A 锁定爪传感器
31015	PH16	票箱 B 锁定爪传感器
31016	PH17	预留
31017	PH18	废票箱锁定爪传感器
31018	PH19	投入检测（右）传感器
31019	PH20	投入检测（左）传感器
31020	PH21	进入检测传感器
31021	PH22	分配 1 通过前检测
31022	PH23	IC 卡 R/W 停止位置
31023	PH24	票箱 A 回收结束传感器
31024	PH25	票箱 B 回收结束传感器
31025	PH26	废票箱回收结束传感器
31026	PH27	预留
31027	PH28	预留
31028	PH29	预留
31029	PH30	预留
31030	PH31	废票箱将满检测传感器
31031	PH32	票箱 A 上限检测传感器
31032	PH33	票箱 B 上限检测传感器
31033	PH34	预留
31034	PH35	废票箱已满检测
31035	PH36	票箱 A 舌档打开检测
31036	PH37	票箱 B 舌档打开检测
31037	PH38	预留
31038	PH39	废票箱舌挡打开检测
31039	PH40	废票箱舌挡关闭检测
31040	PH41	回收单元到位开关
31041	PH42	预留

10.5.1.4 结构体说明

10.5.1.4.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId;           //模块厂家标识三星0x02、广电0x04、中软0x06;
    UINT uiModuleType           //模块类型标识 0x43车票回收模块
    UINT uiType;                //错误类型：0-正常；1-警告；2-故障
```



```

        UINT uiErrorType           //故障类型 0-正常；1-命令类错误；2-硬件故障；3-通道异常
                                   故障；4-票箱故障；
        UINT uiErrorCode;           //错误代码（由不同模块厂家提供）
        char cReverse[128];         //预留信息
    } tTAMDevReturn;

```

10.5.1.4.2 票箱数据信息结构体

```

typedef struct
{
    UINT uiCassetteACardNum;        // 回收箱A卡片数量
    UINT uiCassetteBCardNum;        // 回收箱B卡片数量
    UINT uuiRejectCassetteNum;      // 废票箱卡片数量
} tTAMCardNum;

```

10.5.1.4.3 模块状态结构体

```

typedef struct
{
    bool bCardInTicketGate;         //是否有卡投票口, TRUE:有卡, FALSE:无卡
    bool bCardInReadArea;           //是否有卡在读卡区, TRUE:有卡, FALSE:无卡
    UINT bCassetteAStatus;           //回收箱A状态, 0正常, 1:半满, 2:全满, 3:未安装, 4:安装
    不到位, 5:票箱故障
    UINT bCassetteBStatus;           //回收箱B状态, 0正常, 1:半满, 2:全满, 3:未安装, 4:安装
    不到位, 5:票箱故障
    UINT bRejectCassetteStatus;      //废票箱状态, 0正常, 1:半满, 2:全满, 3:未安装, 4:安装
    不到位, 5:票箱故障
    UINT uiSensorCount;              //传感器数量
    Struct
    {
        UINT uiSensotID;             //传感器ID
        BYTE btSensorStatus;          //传感器状态(0:正常 1:故障 3:未遮挡 4:遮挡 0xff:
                                      不存在该传感器)
    } tSensorType[256];              // 实际使用范围从
                                      tSensorType[0]--- tSensorType[uiSensorCount-1]
    tTAMCardNum tCarNum;              // 票箱车票数量
    char cReserve[256];              // 保留信息
} tTAMModuleStatus;

```

10.5.1.4.4 审计累计值结构体

```

typedef struct
{
    UINT wCassetteAAuditCardNum;     // 票箱A累计发卡数
    UINT wCassetteBAuditCardNum;     // 票箱B累计发卡数
    UINT wRejectCassetteAuditNum;     // 废票箱卡片数量
} tTAMAuditCardNum;

```

10.5.1.4.5 电子标签信息结构体

```

typedef struct
{
    unsigned char ucBoxID[4];         // 票箱编号
    unsigned char ucOperateID[3];     // 操作员ID

```

```
    unsigned char ucDeviceID[4];           // 设备ID
    unsigned char ucBoxStatus;             // 票箱位置状态
    unsigned char ucBoxOperateStatus;      // 操作后票箱状态
    unsigned char ucCardProviderID[4];     // 车票发行商ID
    unsigned char ucCardType;              // 车票物理类型
    unsigned char ucTicketClass[2];        // 车票产品种类
    unsigned char ucPreValueType;          // 预赋值属性
    unsigned char ucDerivedID;             // 衍生产品ID
    unsigned char ucCardNum[2];            // 车票数量
    unsigned char ucStationCode[2];        // 车站编码
    unsigned char ucInstallPosit;          // 安装位置
    unsigned char ucOperateTime[7];        // 最后操作时间
    unsigned char ucReserve[12];           // 预留字段
} tTAM电子标签Info;
```

10.5.1.5 通讯建立

函数定义：
int TAM_CommOpen(UINT uiCommPort, UINT uiBaudRate);
功能说明：
建立串口通信，如参数设置错误或串口被占用，返回错误码。
参数说明：
通讯建立的参数说明如表 178。

表 178 通讯建立参数说明

序号	参数类型	参数名称	输入/输出	说明
1	UINT	uiCommPort	In	端口号
2	UINT	uiBaudRate	In	波特率

返回值：
返回值如表 179。

表 179 通讯建立返回值

类型	说明
int	0: 执行成功; 1, 执行失败。

注意事项：
执行其他操作前，必须先设置串口。
如果该接口返回失败，请检查串口ID或者设备逻辑名是否正确以及串口设备是否正确等。

10.5.1.6 模块初始化

函数定义：
int TAM_Init(char cClearMode, UINT* uiRetractNum, tTAMModuleStatus* pModuleStatus, tTAMDevReturn* pDevStatus);
功能说明：
此接口用于设备的初始化，检查通道、托盘等硬件是否正常，并在初始化过程中清理通道中的车票，如果部件等正常则处于待机状态，允许进一步操作。如果不正常则报错，需要排除故障后才能工作。
参数说明：
模块初始化参数说明如表 180。

表 180 模块初始化参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cClearMode	In	车票清理方式:

				1、从退票口退票 2、清理到废票箱
2	UINT*	uiRetractNum	Out	本次操作处理的卡片数
3	tTAMModuleStatus*	pModuleStatus	Out	返回模块状态信息
4	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：
模块初始化返回值如表 181。

表 181 模块初始化返回值

类型	说明
int	0：执行成功；1，执行失败。

注意事项：
设备开机或复位后需调用本函数。

10.5.1.7 模块复位

函数定义：
int TAM_Reset(char cClearMode, UINT* uiRetractNum, tTAMModuleStatus* pModuleStatus, tTAMDevReturn* pDevStatus);

功能说明：
重新启动回收模块的控制程序。

参数说明：
模块复位参数说明如表 182。

表 182 模块复位参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cClearMode	In	车票清理方式： 1、从退票口退出 2、清理到废票箱
2	UINT*	uiRetractNum	Out	本次操作回收的卡片数
3	tTAMModuleStatus*	pModuleStatus	Out	返回模块状态信息
4	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：
模块复位返回值如表 183。

表 183 模块复位返回值

类型	说明
int	0：执行成功；1，执行失败。

注意事项：
在设备故障或从维修模式退出时（对于支持维修模式的情况），执行本函数。

10.5.1.8 读取设备驱动版本信息

函数定义：
int TAM_GetVersion(char cFirmWareVersion[32], char cDriverVersion[32], tTAMDevReturn* pDevStatus);

功能说明：
获取设备驱动版本号、固件版本号(如回收模块的ID等)、状态信息。

参数说明：
读取设备驱动版本信息参数说明如表 184。

表 184 读取设备驱动版本信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cFirmWareVersion[32]	Out	返回固件版本号
2	char	cDriverVersion[32]	Out	返回驱动版本号
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：

读取设备驱动版本信息返回值如表 185。

表 185 读取设备驱动版本信息返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.1.9 读取设备状态

函数定义：

```
int TAM_GetDevStatus(tTAMModuleStatus* pModuleStatus, tTAMDevReturn* pDevStatus);
```

功能说明：

此接口用于获取读写区是否有卡，票箱状态、票箱数量以及各传感器实时状态。

参数说明：

读取设备状态参数说明如表 186。

表 186 读取设备状态参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTAMModuleStatus*	pModuleStatus	Out	设备状态
2	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：

读取设备状态返回值如表 187。

表 187 读取设备状态返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.1.10 设置接收模式

函数定义：

```
int TAM_SetServiceMode(char cOption, tTAMDevReturn* pDevStatus);
```

功能说明：

此接口用于设置是否允许回收模块回收车票，即是否打开回收口。

如果允许接收车票，则监控读写区是否有卡并获取票箱状态、票箱数量以及各传感器实时状态。

如果不允许接收车票，则插卡口闸门处于关闭状态，模块不检测插卡口是否有卡插入，不会自动将车票吸入到车票处理单元的天线板位置。

一般应在停止服务时，调用本功能，以便关闭插卡口闸门。

参数说明：

设置接收模式参数说明如表 188。

表 188 设置接收模式参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cOption	In	接收模式： 0，允许接收车票(打开回收口) 1，拒绝接收车票(关闭回收口)
2	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：

设置接收模式返回值如表 189。

表 189 设置接收模式返回值

类型	说明
int	0: 执行成功; 1, 执行失败。

10.5.1.11 回收车票

函数定义:

```
int TAM_AcceptCard(char cTicketBoxNo, tTAMCardNum *pCassetteCardNum, tTAMDevReturn* pDevStatus);
```

功能说明:

此接口用于将处于车票处理单元天线板位置的车票回收至票箱里。

参数说明:

回收车票参数说明如表 190。

表 190 回收车票参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	选择 Box 1: 回收箱 A 2: 回收箱 B 3: 废票箱
2	tTAMCardNum*	pCassetteCardNum	Out	票箱数量结构体
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值:

回收车票返回值如表 191。

表 191 回收车票返回值

类型	说明
int	0: 执行成功; 1, 执行失败。

可能出现的错误:

回收车票可能出现的错误如表 192。

表 192 回收车票可能出现的错误

序号	类型	说明
读写卡位置无卡警告	一般警告	无需处理
主马达堵转	严重	重试
卡读写位置有卡堵塞	严重	调用“模块复位”接口

注意事项:

无。

10.5.1.12 退出车票

函数定义:

```
int TAM_ReturnCard(char cReturnMode, tTAMDevReturn* pDevStatus);
```

功能说明:

退出无效车票。

参数说明:

退出车票参数说明如表 193。

表 193 退出车票参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cReturnMode	In	退卡方式: 1、车票从插卡口退出并丢弃;

				2、车票退出到插卡口并夹住
2	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：

退出车票返回值如表 194。

表 194 退出车票返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：

退出车票可能出现的错误如表 195。

表 195 退出车票可能出现的错误

可能出现的错误	错误类型	推荐处理措施
读写卡位置无卡警告	一般警告	无需处理
废票箱不存在	错误严重	停机维护
废票箱满/有卡在回收位置堵塞错	严重	停机维护
主马达堵转	严重	重试
出口位置有卡堵塞	严重	调用“模块复位”接口
卡读写位置有卡堵塞	严重	调用“模块复位”接口
通道内有卡堵塞	严重	调用“模块复位”接口
换向器故障	严重	调用“模块复位”接口

注意事项：

无。

10.5.1.13 读取审计累计值

函数定义：

```
int TAM_GetAuditData(tTAMAuditCardNum* pAuditCardNum, tTAMDevReturn *pDevStatus);
```

功能说明：

读取设备的审计累计值。

参数说明：

读取审计累计值参数说明如表 196。

表 196 读取审计累计值参数说明

序号	参数类型	参数名称	输入/输出	说明
1	tTAMAuditCardNum*	pAuditCardNum	Out	读出的审计累计值，包括：A、B 票箱累计回收卡数和废票箱累计回收卡数
2	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：

读取审计累计值返回值如表 197。

表 197 读取审计累计值返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：

无。

注意事项：

A、B 票箱累计回收卡数：指的是票箱回收成功的次数。即总装入的卡数是A、B 累计回收卡数和废票箱回收卡数的和。

10.5.1.14 读取票箱电子标签信息

函数定义:

```
int TAM_GetTicketBoxRFIDInfo(char cTicketBoxNo, tTAMRFIDInfo* pRFIDInfo, tTAMDevReturn *pDevStatus);
```

功能说明:

读取票箱电子标签信息。

电子标签信息中的密钥不需要处理。

参数说明:

读取票箱电子标签信息参数说明如表 198。

表 198 读取票箱电子标签信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1: 回收箱 A 2: 回收箱 B 3: 废票箱
2	tTAMRFIDInfo*	pRFIDInfo	Out	返回 RFID 信息
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值:

读取票箱电子标签信息返回值如表 199。

表 199 读取票箱电子标签信息返回值

类型	说明
int	0: 执行成功; 1, 执行失败。

可能出现的错误:

见设备返回码的定义及其说明。

注意事项:

无。

10.5.1.15 读取票箱数量信息

函数定义:

```
int TAM_GetTicketBoxCountInfo(char cTicketBoxNo, tTAMCardNum * pCardNum, tTAMDevReturn *pDevStatus);
```

功能说明:

读取票箱数量信息。

参数说明:

读取票箱数量信息参数说明如表 200。

表 200 读取票箱数量信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1: 回收箱 A 2: 回收箱 B 3: 废票箱
2	tTAMCardNum *	pCardNum	Out	返回票箱数量信息
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值:

读取票箱数量信息返回值如表 201。

表 201 读取票箱数量信息返回值

类型	说明
----	----

int	0：执行成功；1，执行失败。
-----	----------------

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.5.1.16 设置票箱电子标签信息

函数定义：
int TAM_SetTicketBoxRFIDInfo(char cTicketBoxNo, tTAMRFIDInfo* pRFIDInfo, tTAMDevReturn *pDevStatus);
功能说明：
写入票箱电子标签信息。
参数说明：
设置票箱电子标签信息参数说明如表 202。

表 202 设置票箱电子标签信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1：回收箱 A 2：回收箱 B 3：废票箱
2	tTAMRFIDInfo*	pRFIDInfo	In	设置 RFID 信息
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：
设置票箱电子标签信息返回值如表 203。

表 203 设置票箱电子标签信息返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.5.1.17 设置票箱数量信息

函数定义：
int TAM_SetTicketBoxCountInfo(char cTicketBoxNo, tTAMCardNum* pCardNum, tTAMDevReturn *pDevStatus);
功能说明：
设置票箱数量信息。
参数说明：
设置票箱数量信息参数说明如表 204。

表 204 设置票箱数量信息参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cTicketBoxNo	In	设置的票箱 1：回收箱 A 2：回收箱 B 3：废票箱
2	tTAMCardNum*	pCardNum	In	设置票箱数量信息
3	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：
设置票箱数量信息返回值如表 205。

表 205 设置票箱数量信息返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.5.1.18 更换票箱

函数定义：
int TAM_ReplaceTBox(char cBoxNo, char cBoxMode, tTAMDevReturn* pDevStatus);
功能说明：
此接口用于更换票箱。
在取出票箱前，应使用“取票箱”方式先调用本命令；在放入票箱后，应使用“装票箱”方式调用本命令。
“取票箱”操作，是为了将票箱当前的车票数量写入票箱的电子标签 里保存。“装票箱”操作，是为了将票箱电子标签 里的车票数量读取出来。
参数说明：
更换票箱参数说明如表 206。

表 206 更换票箱参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char	cBoxNo	In	票箱号： 0：更换所有票箱 1：更换回收箱 A； 2：更换回收箱 B； 3：更换废票箱；
2	char	cBoxMode	In	1：取票箱； 2：装票箱
3	tTAMDevReturn *	pDevStatus	Out	返回状态信息

返回值：
更换票箱返回值如表 207。

表 207 更换票箱返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.5.1.19 读取序列号

函数定义：
Int TAM_GetSN(char* cModuleSN, tTAMDevReturn *pDevStatus);
功能说明：
此命令用于读出设备的序列号。
参数说明：
读取序列号参数说明如表 208。

表 208 读取序列号参数说明

序号	参数类型	参数名称	输入/输出	说明
1	char*	cModuleSN	Out	模块的序列号，长度 20
2	tTAMDevReturn*	pDevStatus	Out	返回状态信息

返回值：
读取序列号返回值如表 209。

表 209 读取序列号返回值

类型	说明
int	0：执行成功；1，执行失败。

可能出现的错误：
见设备返回码的定义及其说明。
注意事项：
无。

10.5.1.20 关闭通讯

函数定义：
int TAM_CommClose();
功能说明：
关闭串口通讯。
参数说明：
无。
返回值：
关闭通讯返回值如表 210。

表 210 关闭通讯返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.1.21 设置模块工作模式

函数定义：
int TAM_SetWorkMode(char cMode, tTAMDevReturn* pDevStatus);
功能说明：
设置模块为正常工作或维护模式。
参数说明：
设置模块工作模式参数说明如表 211。

表 211 设置模块工作模式参数说明

序号	参数类型	参数名称	属性	说明
1	char	cMode	In	正常工作模式：0x00 维护模式：0x01
2	tTAMDevReturn&	pDevStatus	Out	返回状态信息

返回值：
设置模块工作模式返回值如表 212。

表 212 设置模块工作模式返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.1.22 设置模块测试

函数定义：

```
int TAM_TestSetting(char cTestType, tTAMModuleStatus* pModuleStatus, tTAMDevReturn* pDevStatus);
```

功能说明：

用于测试模块的部件，包括传感器测试、马达测试、翻板测试等。

参数说明：设置模块测试参数说明如表 213。

表 213 设置模块测试参数说明

序号	参数类型	参数名称	属性	说明
1	char	cTestType	In	测试类型： 0. 停止当前测试 1. 废票箱与正常回收传输切换翻板测试（如果模块包含本设备） 2. 2 个票箱间回收传输切换翻板测试（如果模块包含本设备） 3. 传感器测试（含各类到位触点和光电管） 4. 入票口传输马达测试（如果模块包含本设备） 5. 回收传输马达测试 6. 回收口快门测试 7. A 票箱连续回收测试 8. B 票箱连续回收测试 9. 废票箱连续回收测试 10. 2 个票箱连续回收测试 11. A 票箱升降装置上升测试 12. A 票箱升降装置下降测试 13. B 票箱升降装置上升测试 14. B 票箱升降装置下降测试 15. 废票箱升降装置上升测试（如果模块包含本设备） 16. 废票箱升降装置下降测试（如果模块包含本设备） 17. 回收口 LED 指示 1（如果模块包含本设备） 18. 回收口 LED 指示 2（如果模块包含本设备） 19. 回收口 LED 指示 3（如果模块包含本设备） 20. 模块控制电路板上故障指示器测试（如果模块包含本设备） 21. 模块控制电路板上蜂鸣器测试（如果模块包含本设备） 22. A 票箱电子标签读测试 23. B 票箱电子标签读测试 24. 废票箱电子标签读测试（如果模块包含本设备） 25. A 票箱电子标签写测试 26. B 票箱电子标签写测试 27. 废票箱 电子标签写测试（如果模块包含本设备） 28-100 保留 101-255 各模块厂商自定义
2	tTAMModuleStatus*	pModuleStatus	Out	返回模块状态信息
3	tTAMDevReturn&	pDevStatus	Out	返回状态信息

补充说明：

此命令只能在“维修模式下”使用，否则“无效命令”；

部件测试之间可以不发送“停止当前测试”而直接切换；

退出“维修模式”，自动停止测试；

返回值：

设置模块测试返回值如表 214。

表 214 设置模块测试返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.1.23 获取读卡区是否存在车票

函数定义：

```
int TAM_GetCardInReadArea(bool* bCardInReadArea);
```

功能说明：

获取车票处理单元天线位置是否存在车票。

参数说明：

获取读卡区是否存在车票参数说明如表 215。

表 215 获取读卡区是否存在车票参数说明

参数类型	参数名称	属性	说明
Bool*	bCardInReadArea	out	有卡：true 无卡：false

返回值：

设置模块工作模式返回值如表 216。

表 216 获取读卡区是否存在车票返回值

类型	说明
int	0：执行成功；1，执行失败。

10.5.2 推荐操作流程

10.5.2.1 上电初始化流程

上电初始化流程如图 3。

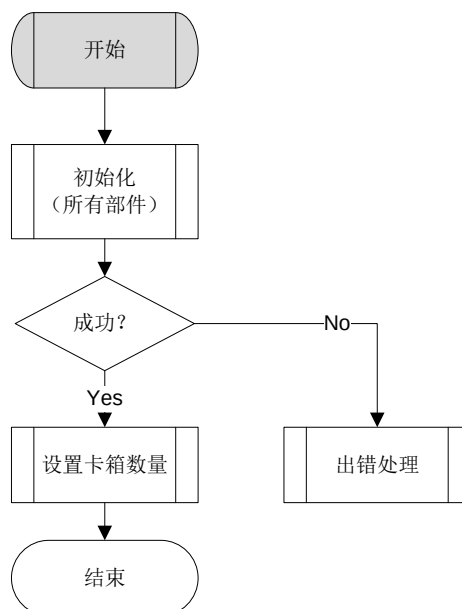


图 3 上电初始化流程

10.5.2.2 接收车票流程

接收车票流程如图 4。

取票箱流程如图 5。

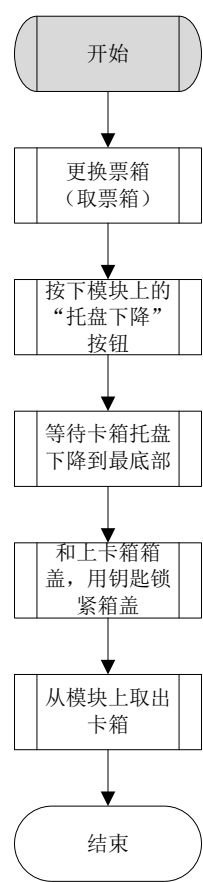


图 5 取票箱流程

10.5.2.4 装票箱流程

装票箱流程如图 6。

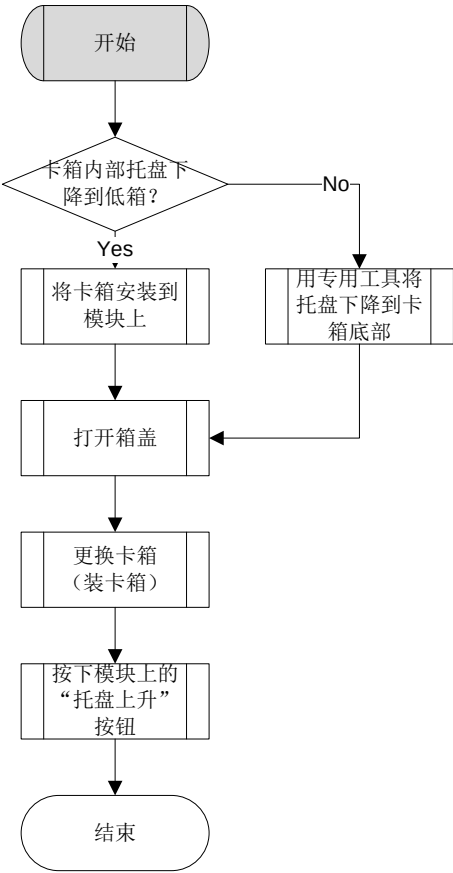


图 6 装票箱流程

10.6 硬币处理模块驱动接口（CHA）

10.6.1 硬币处理模块接口

10.6.1.1 接口汇总表

接口汇总表见表 217。

表 217 接口汇总

功能：	通讯建立
函数定义：	int CHA_CommOpen(UINT uiCommPort, UINT uiBaudRate, tCHADevReturn * pDevStatus);
功能：	初始化
函数定义：	int CHA_Init(tCHADevReturn * pDevStatus);
功能：	获取设备状态
函数定义：	int CHA_GetDevStatus(tCHADevReturn * pDevStatus);
功能：	获取传感器状态
函数定义：	int CHA_GetSensorStatus(tCHASensorStatus* pSensorStatus, tCHADevReturn * pDevStatus);
功能：	获取容器配置信息
函数定义：	int CHA_GetCoinBoxInfo(tCoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);
功能：	设置硬币数量
函数定义：	int CHA_SetCoinBoxInfo(tCoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);
功能：	开始接收硬币
函数定义：	int CHA_StartReceiveCoin(tCHADevReturn * pDevStatus);
功能：	结束接收硬币
函数定义：	int CHA_StopReceiveCoin(tCHADevReturn * pDevStatus);
功能：	获取接收硬币信息
函数定义：	int CHA_GetReceiveNumber(tCHACoinValueInfo *pCoinValueInfo, tCHADevReturn * pDevStatus);

功能:	存入硬币
函数定义:	int CHA_AcceptCoin(tCHADevReturn * pDevStatus);
功能:	退出硬币
函数定义:	int CHA_EjectCoin(UINT * uiReturnCoinNum ,tCHADevReturn * pDevStatus);
功能:	找零
函数定义:	int CHA_ChangeCoin(tCHACHangeNum * pChangeNum, tCHADevReturn * pDevStatus);
功能:	更换容器
函数定义:	int CHA_StartReplaceBOX (UINT uiBoxType, UINT uiBoxNo, tCHADevReturn * pDevStatus);
功能:	停止更换容器
函数定义:	int CHA_StopReplaceBOX (UINT uiBoxType, UINT uiBoxNo, tCHADevReturn * pDevStatus);
功能:	清空硬币
函数定义:	int CHA_ClearCoin(UINT uiBoxTypeOutput, UINT uiBoxNoOutput, UINT uiBoxTypeInput, UINT uiBoxNoInput, tCoinBoxInfo *pCoinBoxInfo, UINT uiCoinNum, tCHADevReturn * pDevStatus);
功能:	读容器电子标签电子标签信息
函数定义:	int CHA_ReadRFIDInfo(UINT uiBoxType, UINT uiBoxNo, unsigned char RFIDInfo[36], tCHADevReturn * pDevStatus);
功能:	写容器电子标签信息
函数定义:	int CHA_WriteRFIDInfo(UINT uiBoxType, UINT uiBoxNo, unsigned char RFIDInfo[36], tCHADevReturn * pDevStatus);
功能:	获取模块介质版本
函数定义:	int CHA_GetVersion(char cVersion[32], tCHADevReturn * pDevStatus);
功能:	获取模块序号
函数定义:	int CHA_GetSerialNumber(char cVersion[32], tCHADevReturn * pDevStatus);
功能:	获取审计数据
函数定义:	int CHA_GetAuditData (tCoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);
功能:	关闭通讯
函数定义:	int CHA_CloseComm(tCHADevReturn * pDevStatus);

10.6.1.2 结构体说明

10.6.1.2.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId;
    UINT uiModuleType;
    UINT uiType;
    UINT uiErrorType[16];
    UINT uiErrorCode[16];
    char cReverse[8];
} tCHADevReturn;
结构体中变量说明见表 218。
```

表 218 返回状态结构体说明

名称	说明
uiProviderId	模块厂家标识: 亿雅捷 0x01、三星 0x02、怡力 0x03、广电 0x04、中软 0x06、欧姆龙 0x05; 普天 0x07。
uiModuleType	模块类型标识: 0x04 硬币处理模块。
uiType	状态类型: 0-正常; 1-警告; 2-故障; 3-只接收模式; 4-只找零模式。
uiErrorType	标准状态代码—详见标准状态代码定义: 模块有可能返回多个状态代码。
uiErrorCode	原始状态代码—为不同模块厂家提供的原始状态代码, 同时每个原始状态代码都能够对应到唯一的标准状态代码中; 原始代码对应标准状态代码可以为多对一。上位机通过该值, 可以显示出详细的错误原因及故障点定位。
cReverse	预留信息

10.6.1.2.2 模块容器配置信息结构体


```
typedef struct
{
    UINT uiBoxType;
    UINT uiBoxNo;
    UINT uiCoinValue;
    UINT uiCoinNum;
    UINT uiCoinMaxNum;
    UINT uiBoxStatus;
} tCoinBoxValueInfo;
typedef struct
{
    UINT uiCoinBoxNum;
    tCoinBoxValueInfo* pCoinBoxValueInfo;
} tCHACoinBoxInfo;
模块容器配置信息结构体中变量说明见表 219。
```

表 219 模块容器配置信息结构体说明

名称	说明
uiCoinBoxNum	硬币容器数量
uiBoxType	容器类型：1：加币箱；2：找零箱；3：暂存器；4：缓存找零器；5：回收箱。
uiBoxNo	容器序号：初始值为 0；同一类型容器序号单独排号，例如加币箱配置 4 个，序号分别为 1—4。
uiCoinValue	单位分；容器中所能存放硬币的面额，该字段值只能为 100、50、0； 其中 0 表示不能确定该容器的面额。
uiCoinNum	容器中硬币的数量
uiCoinMaxNum	容器中能存入硬币最大的数量
uiBoxStatus	容器状态： 0-正常；1-未安装；2-已满；3-将满；4-已空；5-将空。该状态为模块硬件检测的状态，通过计数判断的状态由上位机进行判断

10.6.1.2.3 接受硬币信息结构体

```
typedef struct
{
    UINT uiBoxType;
    UINT uiBoxNo;
    UINT uiCoinValue;
    UINT uiCoinNum;
} tCoinBoxValueInfo;
typedef struct
{
    tCoinBoxValueInfo stCoinBoxValueInfo[2];
    UINT uiEjectCoinNum;
} tCHACoinValueInfo;
接受硬币信息结构体中变量说明见表 220。
```

表 220 接受硬币信息结构体说明

名称	说明
uiBoxType	1：加币箱；2：找零箱；3：暂存器；4：缓存找零器；5：回收箱。
uiBoxNo	与 3.2 节结构体定义保持一致
uiCoinValue	单位分；容器中所能存放硬币的面额，该字段值只能为 100、50。
uiCoinNum	容器中硬币的数量 （从开始接收至结束接收阶段的累加值）
tCoinBoxValueInfo	最多只能接收两种面额接收

uiEjectCoinNum	本次交易退出未识别硬币数量（从开始接收—结束接收阶段的累加值）
----------------	---------------------------------

10.6.1.2.4 传感器状态结构体

```
typedef struct
{
    UINT uiSensorID;
    UINT uiStatus;
} SensorStatus;
typedef struct
{
    UINT uiSensorNum;
    SensorStatus* pSensorStatus;
} tCHASensorStatus;
传感器状态结构体中变量说明见表 221。
```

表 221 传感器状态结构体说明

名称	说明
uiSensorNum	传感器数量
uiSensorID	传感器 ID
uiStatus	0：未遮挡；1：遮挡。

10.6.1.2.5 找零硬币信息结构体

```
typedef struct
{
    UINT uiCoinValue;
    UINT uiCoinNum;
} tCoinChangeInfo;
typedef struct
{
    UINT uiCoinTypeNum;
    tCoinChangeInfo* pCoinChangeInfo;
} tCHACHangeNum;
找零硬币信息结构体中变量说明见表 222。
```

表 222 找零硬币信息结构体说明

名称	说明
uiCoinValue	单位分；找零硬币的面额，该字段值只能为 100、50。
uiCoinNum	对应面额找零硬币的数量
uiCoinTypeNum	需找零硬币种类数量（按面额区分）

10.6.1.3 通讯建立

通讯建立函数说明见表 223。

表 223 通讯建立函数说明

功能:	通讯建立
函数定义:	int CHA_CommOpen(UINT uiCommPort, UINT uiBaudRate, tCHADevReturn * pDevStatus);
功能说明:	建立串口通信，如参数设置错误或串口被占用，返回状态代码。如模块需要与上位机建立额外的连接，需在该函数中实现。
参数说明:	uiCommPort: 串口端口号（输入）
	uiBaudRate: 波特率（输入）
	pDevStatus: 设备返回状态（输出）

10.6.1.4 初始化

初始化函数说明见表 224。

表 224 初始化函数说明

功能:	初始化
函数定义:	int CHA_Init(tCHADevReturn * pDevStatus);
功能说明:	对硬件进行初始化，用于检测各硬件模块。初始化后清除接收硬币的信息。
参数说明:	pDevStatus: 设备返回状态（输出）

10.6.1.5 获取模块状态

获取模块状态函数说明见表 225。

表 225 获取模块状态函数说明

功能:	获取设备状态
函数定义:	int CHA_GetDevStatus(tCHADevReturn * pDevStatus);
功能说明:	获取当前设备的信息，判断设备是否可用。
参数说明:	pDevStatus: 设备返回状态（输出）

10.6.1.6 获取传感器状态

获取传感器状态函数说明见表 226。

表 226 获取传感器状态函数说明

功能:	获取传感器状态
函数定义:	int CHA_GetSensorStatus(tCHASensorStatus* pSensorStatus, tCHADevReturn * pDevStatus);
功能说明:	获取传感器状态
参数说明:	pSensorStatus: 传感器状态（输出）
	pDevStatus: 设备返回状态（输出）

10.6.1.7 获取容器配置信息

获取容器配置信息函数说明见表 227。

表 227 获取容器配置信息函数说明

功能:	获取容器配置信息
函数定义:	int CHA_GetCoinBoxInfo(tCHACoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);
功能说明:	获取当前设备容器的信息。
参数说明:	pCoinBoxInfo: 容器配置信息（输出）
	pDevStatus: 设备返回状态（输出）

10.6.1.8 设置硬币数量

设置硬币数量函数说明见表 228。

表 228 设置硬币数量函数说明

功能:	设置硬币数量
函数定义:	<code>int CHA_SetCoinBoxInfo(tCHACoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);</code>
功能说明:	设置硬币数量。只能对“获取容器配置信息”所返回的信息进行硬币数量更改，其它字段不允许更改。
参数说明:	pCoinBoxInfo: 容器配置信息（输入）
	pDevStatus: 设备返回状态（输出）

10.6.1.9 开始接收硬币

开始接收硬币函数说明见表 229。

表 229 开始接收硬币函数说明

功能:	开始接收硬币
函数定义:	<code>int CHA_StartReceiveCoin(tCHADevReturn * pDevStatus);</code>
功能说明:	开始接收硬币。开始接收后需循环调用获取接收硬币信息来判断是否结束接收。调用该接口后清除接收硬币的信息。
参数说明:	pDevStatus: 设备返回状态（输出）

10.6.1.10 结束接收硬币

结束接收硬币函数说明见表 230。

表 230 结束接收硬币函数说明

功能:	结束接收硬币
函数定义:	<code>int CHA_StopReceiveCoin(tCHADevReturn * pDevStatus);</code>
功能说明:	停止接收硬币，关闭硬币接收口。
参数说明:	pDevStatus: 设备返回状态（输出）

10.6.1.11 获取接收硬币信息

获取接收硬币信息函数说明见表 231。

表 231 获取接收硬币信息函数说明

功能:	获取接收硬币信息
函数定义:	<code>int CHA_GetReceiveNumber(tCHACoinValueInfo *pCoinValueInfo, tCHADevReturn * pDevStatus);</code>
功能说明:	获取从本次开始接收硬币到结束接收硬币之间的累计值。
参数说明:	pCoinValueInfo: 获取本次接收硬币信息（输出）
	pDevStatus: 设备返回状态（输出）

10.6.1.12 存入硬币

存入硬币函数说明见表 232。

表 232 存入硬币函数说明

功能:	存入硬币
函数定义:	<code>int CHA_AcceptCoin(tCHADevReturn * pDevStatus);</code>
功能说明:	将本次接收的硬币进行保存，硬币的存入位置由模块确定。调用该接口后清除接收硬币的信息。

参数说明:	pDevStatus: 设备返回状态 (输出)
-------	-------------------------

10.6.1.13 退出硬币

退出硬币函数说明见表 233。

表 233 退出硬币函数说明

功能:	退出硬币
函数定义:	int CHA_EjectCoin(UINT * uiReturnCoinNum, tCHADevReturn * pDevStatus);
功能说明:	将本次接收到暂存区的硬币退出到退币口。调用该接口后清除接收硬币的信息。
参数说明:	uiReturnCoinNum: 本次操作退出硬币的数量, 不包含未识别硬币而由模块退出的数量 (输出)
	pDevStatus: 设备返回状态 (输出)

10.6.1.14 找零

找零函数说明见表 234。

表 234 找零函数说明

功能:	找零
函数定义:	int CHA_ChangeCoin(tCHACoinNum * pChangeNum, tCHADevReturn * pDevStatus);
功能说明:	按照硬币的面额及数量信息进行找零操作。
参数说明:	pChangeNum: 找零的硬币面额及数量信息 (输入)
	pDevStatus: 设备返回状态 (输出)

10.6.1.15 更换容器

更换容器函数说明见表 235。

表 235 更换容器函数说明

功能:	更换容器
函数定义:	int CHA_StartReplaceBOX (UINT uiBoxType, UINT uiBoxNo, tCHADevReturn * pDevStatus);
功能说明:	对指定的容器开始进行更换操作。
参数说明:	uiBoxType: 指定硬币容器的类型, 参见 tCHACoinBoxInfo->uiBoxType 定义 (输入)
	uiBoxNo: 指定硬币容器的序号, 参见 tCHACoinBoxInfo->uiBoxNo 定义 (输入)
	pDevStatus: 设备返回状态 (输出)

10.6.1.16 停止更换容器

停止更换容器函数说明见表 236。

表 236 停止更换容器函数说明

功能:	停止更换容器
函数定义:	int CHA_StopReplaceBOX (UINT uiBoxType, UINT uiBoxNo, tCHADevReturn * pDevStatus);
功能说明:	对指定的容器结束更换操作。
参数说明:	uiBoxType: 指定硬币容器的类型, 参见 tCHACoinBoxInfo->uiBoxType 定义 (输入)
	uiBoxNo: 指定硬币容器的序号, 参见 tCHACoinBoxInfo->uiBoxNo 定义 (输入)
	pDevStatus: 设备返回状态 (输出)

10.6.1.17 清空硬币

清空硬币函数说明见表 237。

表 237 清空硬币函数说明

功能:	清空硬币
函数定义:	int CHA_ClearCoin(UINT uiBoxTypeOutput, UINT uiBoxNoOutput, UINT uiBoxTypeInput, UINT uiBoxNoInput, tCHACoinBoxInfo *pCoinBoxInfo,UINT uiCoinNum,tCHADevReturn * pDevStatus);
功能说明:	将指定容器的硬币清空。清空硬币后函数返回清币信息。
参数说明:	uiBoxTypeOutput:指定清币容器的类型,参见 tCHACoinBoxInfo->uiBoxType 定义(输入),当该值输入 0 时,表示将所有容器的硬币均清空。此时清币数量将不指定,由模块将所有容器内硬币清空,参数 uiCoinNum 将无效。
	uiBoxNoOutput:指定清币容器的序号,参见 tCHACoinBoxInfo->uiBoxNo 定义(输入)
	uiBoxTypeInput:指定回收容器的类型,参见 tCHACoinBoxInfo->uiBoxType 定义(输入),当该值输入 0 时,表示不指定回收容器,由模块自行指定。
	uiBoxNoInput:指定回收容器的序号,参见 tCHACoinBoxInfo->uiBoxNo 定义(输入)
	pCoinBoxInfo:容器配置信息(输出),其中 uiCoinNum 表示清出硬币的数量
	uiCoinNum: 该值为 0 时,表示将指定容器清空;非 0 时,表示指定清除指定数量的硬币(输入)
	pDevStatus: 设备返回状态(输出)

10.6.1.18 读容器电子标签信息

读容器电子标签信息函数说明见表 238。

表 238 读容器电子标签信息函数说明

功能:	读容器电子标签信息
函数定义:	int CHA_ReadRFIDInfo(UINT uiBoxType, UINT uiBoxNo,unsigned char RFIDInfo[36],tCHADevReturn * pDevStatus);
功能说明:	读取指定容器的电子标签信息。
参数说明:	uiBoxType:指定容器的类型,参见 tCHACoinBoxInfo->uiBoxType 定义(输入)
	uiBoxNo:指定容器的序号,参见 tCHACoinBoxInfo->uiBoxNo 定义(输入)
	RFIDInfo[36]:数据结构参见导则定义,前 4 字节为容器编号,块 A、块 B 由标准化接口封装层进行控制(输出)
	pDevStatus: 设备返回状态(输出)

10.6.1.19 写容器电子标签信息

写容器RFID信息函数说明见表 239。

表 239 写容器电子标签信息函数说明

功能:	写容器电子标签信息
函数定义:	int CHA_WriteRFIDInfo(UINT uiBoxType, UINT uiBoxNo,unsigned char RFIDInfo[36],tCHADevReturn * pDevStatus);
功能说明:	写指定容器的电子标签信息。
参数说明:	uiBoxType: 容器的类型,参见 tCHACoinBoxInfo->uiBoxType 定义(输入)
	uiBoxNo: 指定容器的序号,参见 tCHACoinBoxInfo->uiBoxNo 定义(输入)
	RFIDInfo[36]:数据结构参见导则定义,前 4 字节为容器编号需分配该空间由标准化接口忽略该值,块 A、块 B 由标准化接口封装层进行控制(输入)
	pDevStatus: 设备返回状态(输出)

10.6.1.20 获取模块介质版本

获取模块介质版本函数说明见表 240。

表 240 获取模块介质版本函数说明

功能:	获取模块介质版本
函数定义:	int CHA_GetVersion(char cVersion[32],tCHADevReturn * pDevStatus);
功能说明:	获取模块介质版本。

参数说明:	cVersion:版本号, ASCII 编码 (输出)
	pDevStatus: 设备返回状态 (输出)

10.6.1.21 获取模块序号

获取模块序号函数说明见表 241。

表 241 获取模块序号函数说明

功能:	获取模块序号
函数定义:	int CHA_GetSerialNumber(char cVersion[32], tCHADevReturn * pDevStatus);
功能说明:	获取模块序号。
参数说明:	cVersion:序号, ASCII 编码 (输出)
	pDevStatus: 设备返回状态 (输出)

10.6.1.22 获取审计数据

获取审计数据函数说明见表 242。

表 242 获取审计数据函数说明

功能:	获取审计数据
函数定义:	int CHA_GetAuditData (tCHACoinBoxInfo *pCoinBoxInfo, tCHADevReturn * pDevStatus);
功能说明:	获取审计数据, 该值为模块开始使用后的累加值。
参数说明:	pCoinBoxInfo:容器配置信息 (输出)
	pDevStatus: 设备返回状态 (输出)

10.6.1.23 关闭通讯

关闭通讯函数说明见表 243。

表 243 关闭通讯函数说明

功能:	关闭通讯
函数定义:	int CHA_CloseComm(tCHADevReturn * pDevStatus);
功能说明:	关闭通讯。
参数说明:	pDevStatus: 设备返回状态 (输出)

10.6.2 推荐操作流程

10.6.2.1 初始设置及主要工作

初始设置及主要工作流程如图 7。

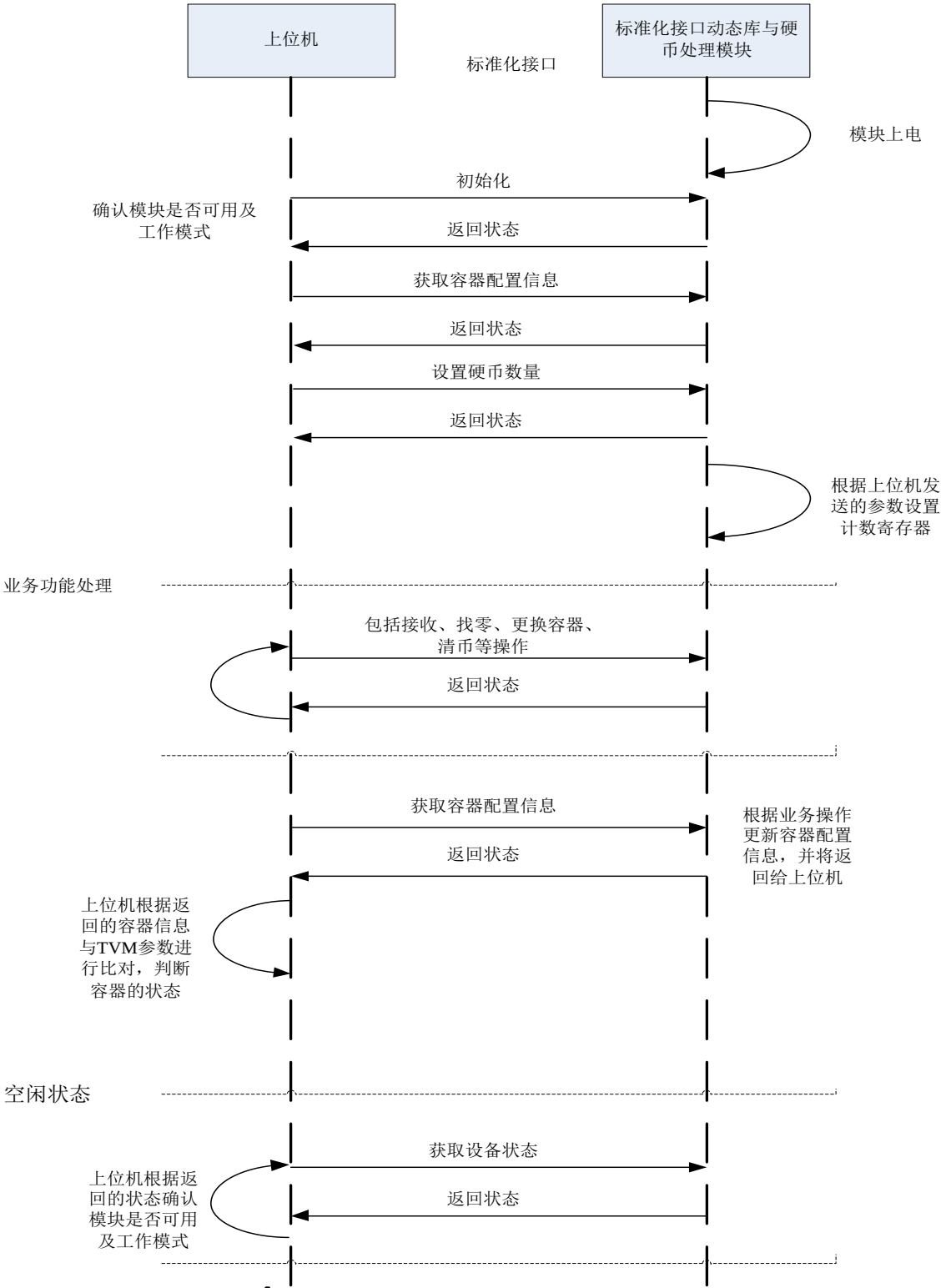


图 7 初始设置及主要工作流程

10.6.2.2 找零流程

找零流程如图 8。

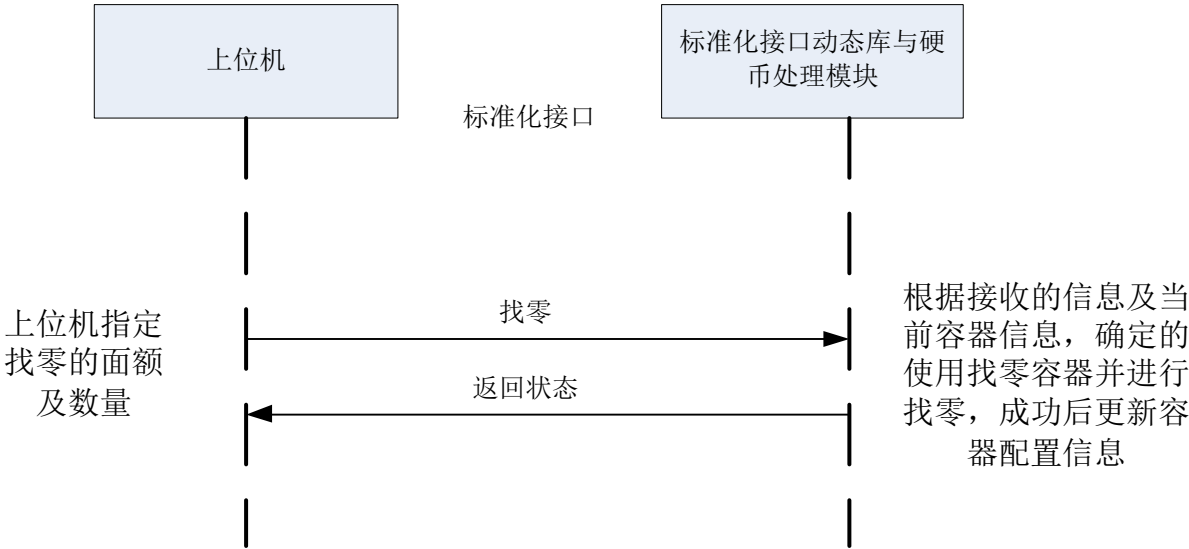


图 8 找零流程

10.6.2.3 接收硬币流程

接收硬币流程如图 9。

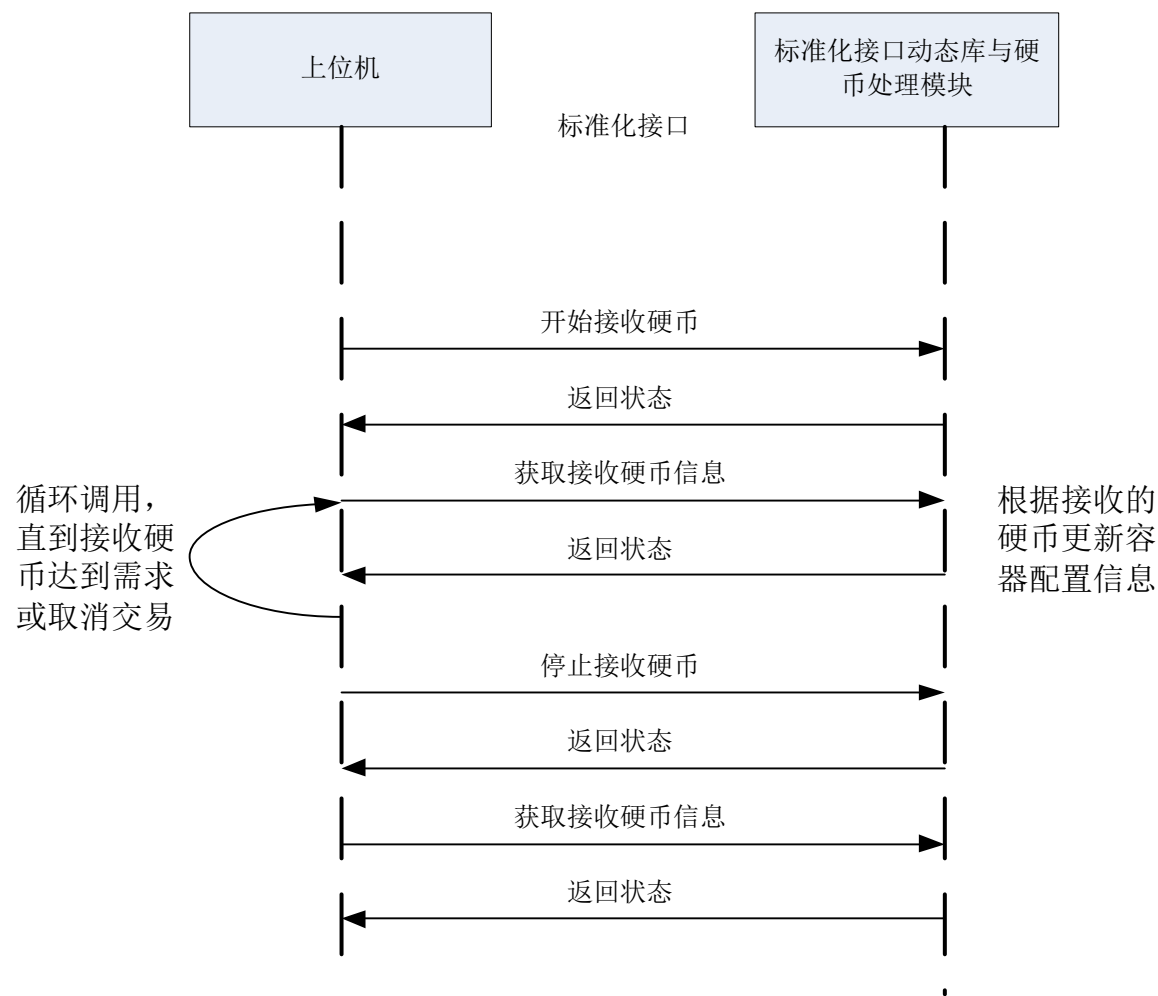


图 9 接收硬币流程

10.6.2.4 清理硬币流程

清理硬币流程如图 10。

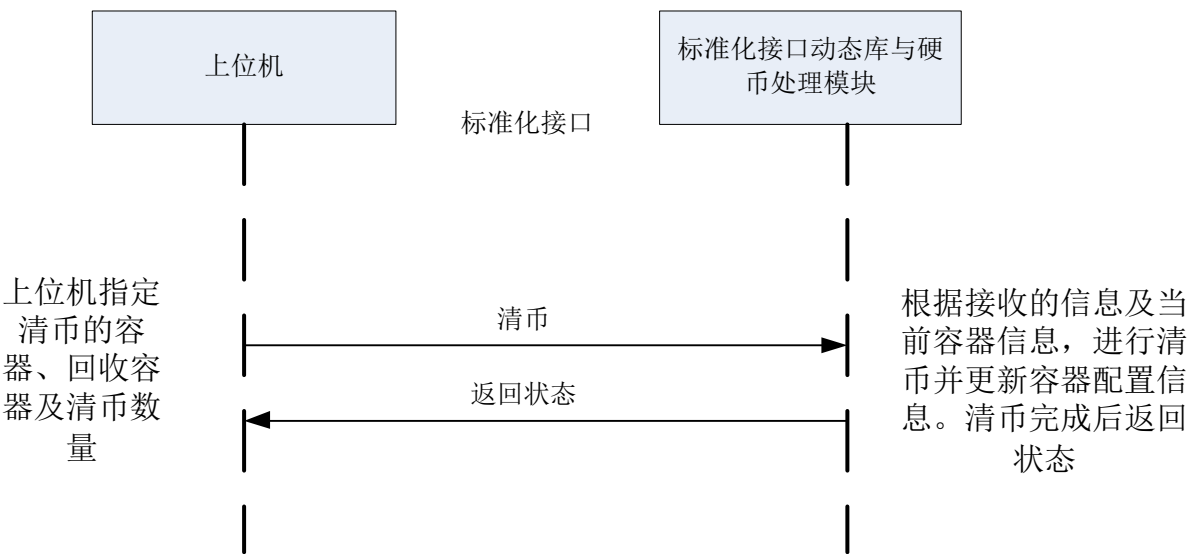


图 10 清理硬币流程

10.6.2.5 更换容器流程

更换容器流程如图 11。

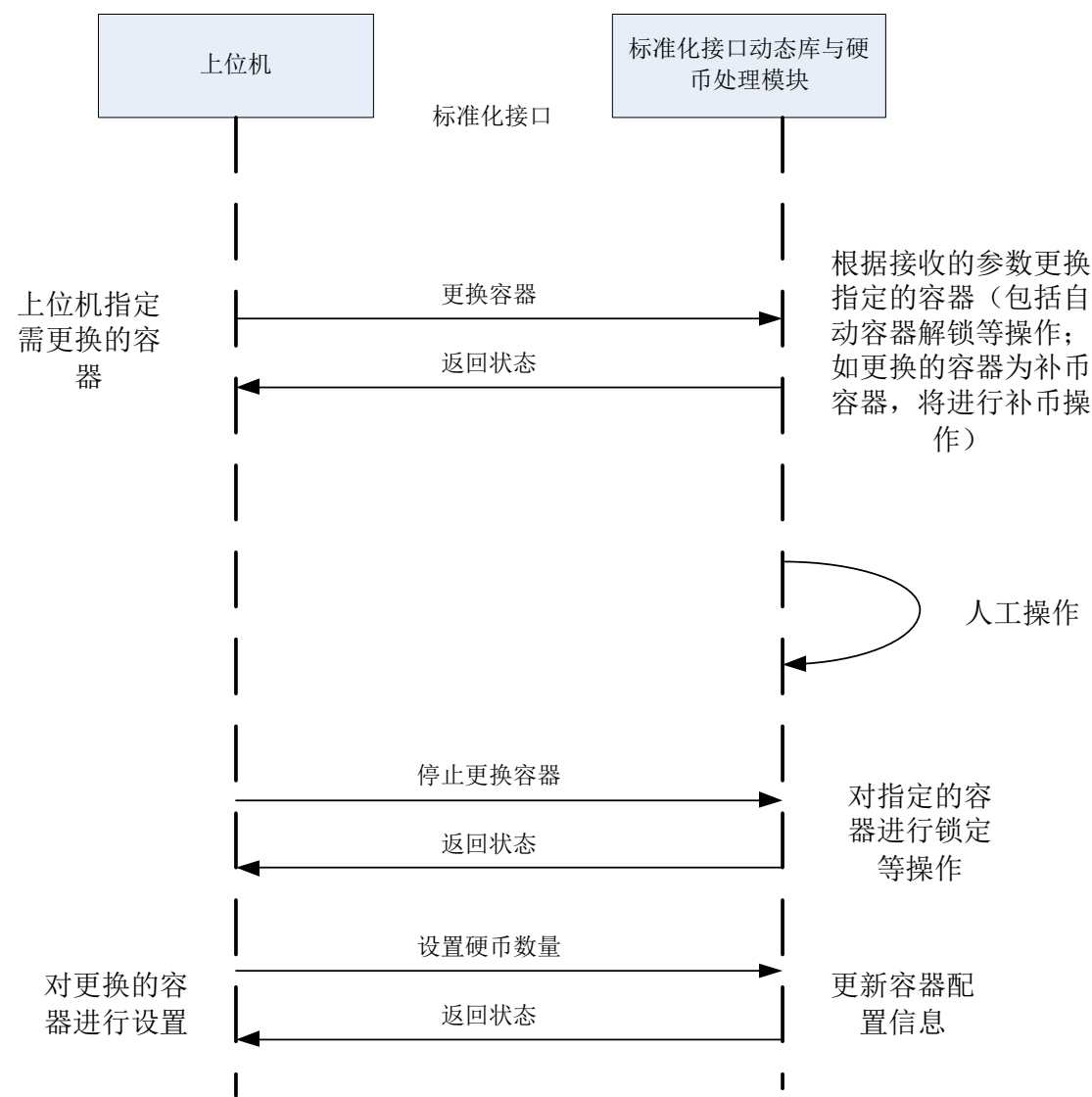


图 11 更换容器流程

10.7 DIO 控制装置模块接口

10.7.1 接口汇总表

接口汇总表见表 244。

表 244 控制装置模块接口汇总

接口名称	接口说明
DIO_Open	通讯建立
DIO_Close	通讯关闭
DIO_Init	初始化模块
DIO_SetLampSignal	设置灯信号
DIO_SetSwitchSignal	设置开关信号
DIO_GetSwitchSignal	获取开关信号数据
DIO_GetVersion	获取模块固件版本

10.7.2 错误码汇总表

错误码汇总见表 245。

表 245 控制装置模块错误码汇总

错误类型 (iType)	错误代码 (iErrorCode)	说明
2	10000	打开 DIO 模块错误!
2	10001	初始化 DIO 模块错误!
2	10002	复位 DIO 模块错误!
2	10003	提示灯信号编号输入错误!
2	10004	设置灯信号发生错误!
2	10005	开关信号编号输入错误!
2	10006	设置开关信号发生错误!
2	10007	获取开关信号发生错误!
1	20000	硬件不支持该功能!

10.7.3 结构体说明

10.7.3.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId;    //模块厂家标识
    int iType;            //错误类型: 0-正常; 1-警告; 2-故障
    int iStdErrorCode;    //标准错误代码
    int iErrorCode; //厂商错误代码
    char chErrorCode[128]; // 硬件返回信息
    char acReverse[128];  //预留信息
} tDioDevReturn
```

10.7.3.2 提示灯信号设置结构体

```
typedef struct
{
    USHORT usLampType; //提示灯类型:
    UCHAR ucLampColor; //提示灯颜色, 单色灯不使用该参数;
                        // 1, 红灯(警告/错误灯)
                        // 2, 绿灯
                        // 3, 蓝灯
                        // 4, 黄灯
                        // 5, 白灯
    UCHAR ucOperType; //操作类型: 0x00, 关闭灯; 0x01, 打开灯
    UCHAR ucBlinkTimes; //闪烁次数 (操作类型为0x01时有效):
                        // 0x00, 不闪烁;
                        // ≥ 0x00, 闪烁次数
} tDioLampConfig;
```

10.7.3.3 开关信号配置结构体

```
typedef struct
{
    USHORT usSignalType; //开关信号类型
    UCHAR ucOperType;    //操作类型:
                        // 0x00, 关闭信号
                        // 0x01, 打开信号
}
```

```
} tDioSignalConfig;
```

表 246 usSignalType 定义

序号	信号类型编码	名称	工控机输入/输出	说明
公共信号				
1	0x0001	低功率信号	0	用于设备休眠，进入节能状态；
2	0x0002	紧急信号	I	1、2、八通线通过此信号判断紧急模式和来源。标准化软件通过导则数据解决此问题；
3	0x0003	TPU_A 复位信号	0	用于 TPU 软件复位
4	0x0004	TPU_B 复位信号	0	用于 TPU 软件复位
5	0x0005	回收模块复位信号	0	
6	0x0006	扇门模块复位信号	0	
7	0x0007	发票模块复位信号	0	
8	0x0008	纸币接收模块复位信号	0	
9	0x0009	纸币找零模块复位信号	0	
10	0x000A	硬币处理模块复位信号	0	
11	0x000B	维修门 1 开关信号	I	维修门编号按位置按序排列 注：个别线路 AG 的维修门连接 PCM 板，该状态通过闸门模块获得，此情况与目前要求的实现方式有差别。参见模块标准化接口中闸门模块状态中的维修门状态；
12	0x000C	维修门 2 开关信号	I	维修门编号按位置按序排列
13	0x000D	维修门 3 开关信号	I	维修门编号按位置按序排列
14	0x000E	维修门 4 开关信号	I	维修门编号按位置按序排列
15	0x000F	维修门 5 开关信号	I	维修门编号按位置按序排列
16	0x0010	维修门 6 开关信号	I	维修门编号按位置按序排列
17	0x0011	维修门 7 开关信号	I	维修门编号按位置按序排列
18	0x0012	维修门 8 开关信号	I	维修门编号按位置按序排列
19	0x0013	蜂鸣器	0	

表 239 usSignalType 定义（续）

序号	信号类型编码	名称	工控机输入/输出	说明
AG 相关信号				
1	0x1001	进站顶棚导向	0	
2	0x1002	出站顶棚导向	0	
3	0x1003	进站通行逻辑授权信号	0	
4	0x1004	出站通行逻辑授权信号	0	
5	0x1005	回收模块投票口信号灯	0	
6	0x1006	进站通行指示灯	0	
7	0x1007	出站通行指示灯	0	
8	0x1008	进站授权反馈信息	I	
9	0x1009	出站授权反馈	I	
10	0x100A	进站刷卡提示灯	0	
11	0x100B	出站刷卡提示灯	0	
12	0x100C	进站顶灯（票种提	0	

		示灯)		
13	0x100D	出站顶灯 (票种提示灯)	0	
14	0x100E	闸机进闸刷卡蜂鸣器	0	
15	0x100F	闸机出闸刷卡蜂鸣器	0	
16	0x1010	本地控制开关	I	
BOM 相关信号				
1	0x2001	发票模块到位信号	I	
2	0x2002	现金抽屉信号	0	
TVM 相关信号				
1	0x3001	人体传感器 1	I	乘客接近
2	0x3002	人体传感器 2	I	
3	0x3003	人体传感器 3	I	
4	0x3004	充值卡打开	0	点击充值按钮后打开
5	0x3005	充值卡弹卡/解锁	0	解锁+弹卡
6	0x3006	充值卡到位	I	卡到位信号
7	0x3007	硬币投币口	0	打开时间不能太长, 电磁铁
8	0x3008	风扇开关	0	人工控制与业务无关
9	0x3009	招援按钮	I	

表 239 usSignalType 定义 (续)

序号	信号类型编码	名称	工控机输入/输出	说明
10	0x3010	安全门开关	0	
11	0x3011	安全门到位	I	
12	0x3012	纸币接收安全门	I	
13	0x3013	纸币找零安全门	I	
14	0x3014	硬币模块安全门	I	
15	0x3015	纸币接收到位	I	
16	0x3016	纸币找零到位	I	
17	0x3017	硬币模块到位	I	
18	0x3018	发售模块到位	I	
19	0x3019	乘客打印机到位	I	
20	0x3020	硬币操作提示灯	0	
21	0x3021	纸币接 3 收提示灯	0	
22	0x3022	储值卡座提示灯	0	
23	0x3023	打印机提示灯	0	
24	0x3024	取票找零提示灯	0	
25	0x3025	预留银行卡提示灯	0	编号, 增加 AG 的两个灯

10.7.3.4 版本信息结构体

```
typedef struct
{
    UCHAR  pDriverVerInfo[32];    //软件版本信息, 32 字节, 不够后补0x00
    UCHAR  pHardwareVerInfo[32];  //硬件版本信息, 32 字节, 不够后补0x00
} tDioVerInfo;
```

10.7.4 通讯建立

```
typedef struct {
    int chSerialPort;    // 串口号
```

```
int nIoSpeed;      // 波特率
int nStopbits;     // 停止位
int nDatabits;     // 数据位
int nParity;       // 校验位
} tSerialConfigInfo;
函数名称: int DIO_Open( int nNum, tSerialConfigInfo * pSerialConfigInfo);
功能说明: 与DIO模块建立通讯并进行设备的轮询以保证模块通讯的正常。
参数说明见表 247。
```

表 247 DIO_Open() 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	nNum	In	串口个数
2	tSerialConfigInfo	pSerialConfigInfo	In	串口信息

注：如果无串口信息，参数nNum填0，pSerialConfigInfo为NULL；如果DIO存在多个串口，应明确串口号的顺序及含义，如：TVM有两个DIO:安全门,乘客传感器，则需说明为：安全门 com1 115200...乘客传感器 com2 115200...。

标准软件通过配置文件与各家集成商设备的环境进行匹配。

动态库名称为: StdDiod.dll

配置文件为: StdDio.conf

[Serial]

count=2

串口号, 波特率, 停止位, 数据位, 校验位

com1=4, 115200, 1, 8, 0

com2=5, 115200, 1, 8, 0

返回值说明见表 248。

表 248 DIO_Open() 返回值说明

类型	说明
Int	0: 执行成功; 1, 执行失败。

10.7.5 通讯关闭

函数名称: int DIO_Close()

功能说明: 关闭模块内所有通讯。

参数说明: 无。

返回值说明见表 249。

表 249 DIO_Close() 返回值说明

类型	说明
int	0: 执行成功; 1, 执行失败。

10.7.6 初始化

函数名称: int DIO_init(tDioDevReturn& tReturn)

功能说明: 初始化模块，检查模块是否可用。

参数说明见表 250。

表 250 DIO_init() 参数说明

参数类型	参数名称	属性	说明
------	------	----	----

tDioDevReturn&	tReturn	in	返回状态信息
----------------	---------	----	--------

返回值说明见表 251。

表 251 DIO_init() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.7.7 设置提示灯信号

函数名称：int DIO_SetLampSignal(tDioLampConfig* pLampConfig, UCHAR ucLampCnt, tDioDevReturn& tReturn)
功能说明：设置灯状态。参数说明见表 252。

表 252 DIO_SetLampSignal() 参数说明

序号	参数类型	参数名称	属性	说明
1	tDioLampConfig*	pLampConfig	in	需要设置的灯信息
2	UINT	ucLampCnt	in	需要设置提示灯的数量（tDioLampConfig 的个数）
3	tDioDevReturn&	tReturn	out	返回状态信息

返回值说明见表 253。

表 253 DIO_SetLampSignal() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.7.8 设置开关信号

函数名称：int DIO_SetSwitchSignal(tDioSignalConfig* pSignalConfig, UCHAR ucSignalCnt, tDioDevReturn& tReturn)
功能说明：设置各类模块开关信号。
参数说明见表 254。

表 254 DIO_SetSwitchSignal() 参数说明

序号	参数类型	参数名称	属性	说明
1	tDioSignalConfig*	pSignalConfig	in	需要设置的开关信号
2	UCHAR	ucSignalCnt	in	需要设置的开关信号的数量（tDioSignalConfig 的个数）
3	tDioDevReturn&	tReturn	out	返回状态信息

返回值说明见表 255。

表 255 DIO_SetSwitchSignal() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.7.9 获取开关信号

函数名称：int DIO_GetSwitchSignal(tDioSignalConfig* pSignalConfig, UCHAR& ucSignalCnt, tDioDevReturn& tReturn)
功能说明：获取开关输入信号。
参数说明见表 256。

表 256 DIO_GetSwitchSignal() 参数说明

序号	参数类型	参数名称	属性	说明
1	tDioSignalConfig*	pSignalConfig	In/out	需要返回的开关信号配置结构（需要由调用方开辟内存空间）

2	UCHAR&	ucSignalCnt	in/out	返回开关信号的数量（返回 tDioSignalConfig 的个数，个数不超过传入的数量；当传入值为 0 时，返回所有的，调用方应保障调用前为 pSignalConfig 申请足够的空间）
3	tDioDevReturn&	tReturn	out	返回状态信息

返回值说明见表 257。

表 257 DIO_GetSwitchSignal() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.7.10 获取模块固件版本

函数名称：
int DIO_GetVersion(tDioVerInfo& tVersion, tDioDevReturn& tReturn)
功能说明：
用于获取控制模块内的 firmware 版本号。
参数说明见表 258。

表 258 DIO_GetVersion() 参数说明

参数类型	参数名称	属性	说明
tDioVerInfo&	tVersion	out	模块固件版本信息

返回值说明见表 259。

表 259 DIO_GetVersion() 返回值说明

参数名称	类型	说明
tReturn	int	0：执行成功；1，执行失败。

10.8 紧急按钮接口

10.8.1 接口汇总表

接口汇总表见表 260。

表 260 紧急按钮模块错误码汇总

接口名称	接口说明
Urgbtn_Open	通讯建立
Urgbtn_Close	通讯关闭
Urgbtn_Init	初始化模块
Urgbtn_GetStatus	获取状态
Urgbtn_GetVersion	获取模块固件版本

10.8.2 错误码汇总表

错误码汇总见表 261。

表 261 紧急按钮模块错误码汇总

错误类型 (iType)	错误代码 (iErrorCode)	说明
2	10000	打开模块错误！
2	10001	初始化模块错误！
2	10003	获取状态错误！

10.8.3 结构体说明

10.8.3.1 返回状态结构体

```
typedef struct
{
    UINT uiProviderId;    //模块厂家标识
    int iType;            //错误类型：0-正常；1-警告；2-故障
    int iStdErrorCode;    //标准错误代码
    int iErrorCode; //厂商错误代码
    char chErrorCode[128]; // 硬件返回信息
    char acReverse[128];  //预留信息
} tUrgbtnDevReturn;
```

10.8.4 通讯建立

函数名称：int Urgbtn_Open(UINT uiCommPort, UINT uiBaudRate)
功能说明：与紧急按钮模块建立通讯并进行设备的轮询以保证模块通讯的正常。
参数说明见表 262。

表 262 Urgbtn_Open() 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	uiCommPort	In	串口号
2	UINT	uiBaudRate	In	波特率

返回值说明见表 263。

表 263 Urgbtn_Open() 返回值说明

类型	说明
Int	0：执行成功；1，执行失败。

10.8.5 初始化

函数名称：int Urgbtn_Init(tUrgbtnDevReturn& tReturn)
功能说明：初始化模块，检查模块是否可用。
参数说明见表 264。

表 264 Urgbtn_Init() 参数说明

参数类型	参数名称	属性	说明
tUrgbtnDevReturn&	tReturn	in	返回状态信息

返回值说明见表258。

表 265 Urgbtn_Init() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.8.6 获取状态

函数名称：int Urgbtn_GetStatus(int& nStatus, tUrgbtnDevReturn& tReturn)
功能说明：设置各类模块开关信号。
参数说明见表 266。

表 266 Urgbtn_Getstatus() 参数说明

序号	参数类型	参数名称	属性	说明
1	Int	nStatus	out	返回紧急按钮的状态 0 紧急，1 非紧急
2	tUrgbtnDevReturn&	tReturn	out	返回状态信息

返回值说明见表 267。

表 267 Urgbtn_Getstatus() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.8.7 获取固件版本

函数名称：int Urgbtn_GetVersion(tUrgbtnDevReturn& tReturn)
功能说明：获取模块内固件版本号。
参数说明见表 268。

表 268 Urgbtn_GetVersion() 参数说明

参数类型	参数名称	属性	说明
tUrgbtnDevReturn&	tReturn	out	返回状态信息

返回值说明见表 269。

表 269 Urgbtn_GetVersion() 返回值说明

类型	说明
int	0：执行成功；1，执行失败。

10.9 LED 模块接口

10.9.1 接口汇总表

表 270 LED 模块接口汇总

接口名称	接口说明
LED_CommOpen()	通讯建立
LED_Init()	初始化
LED_Reset()	模块复位
LED_Download()	下载显示信息
LED_Display()	显示信息
LED_GetDevStatus()	读取模块状态
LED_GetVersion()	读取模块驱动版本信息
LED_CommClose()	通讯关闭

10.9.2 设备返回码的定义及其说明

表 271 LED 模块错误代码汇总

错误类型(uiType)	错误代码(uiStdErrorCode)	说明
0	1800	状态正常
2	1851	无效命令(没有调用串口打开时也返回该错误代码)
2	1852	故障
2	1853	串口配置错
2	1854	通信超时
2	1855	传入参数错误

10.9.3 结构体说明

10.9.3.1 错误与状态描述结构体

typedef struct

```
{
    //1. 基本信息
    UINT uiProviderId; //模块厂家标识。三星 0x02、广电 0x04、中软 0x06、欧姆龙 0x05;
    UINT uiModuleType; //模块类型标识 0x09 LED 模块
    UINT uiType; //错误类型：0-正常；1-警告；2-故障
    UINT uiStdErrorCode[16]; //统一维护的错误代码
    UINT uiErrorCode[16]; //错误代码 （由不同模块厂家提供）
    char acDevReturn[128]; // 硬件返回信息
    char acReserve[128]; // 保留信息
} tLEDDevReturn;
```

10.9.3.2 模块状态结构体

```
typedef struct
{
    UINT uiModuleStatus; //模块状态：0-正常；1-故障
} tLEDModuleStatus;
```

10.9.3.3 下载内容结构体

```
typedef struct
{
    UINT uiTextNum; //需要显示的记录条数
    tLED_Download_Text_Para* p LED_Download_Text_Para;
} tLED_Download_Para;
```

10.9.3.4 显示内容结构体

```
typedef struct
{
    UINT nSerialNO; //序号
    char cNText[256]; // 需要显示的中文
    char cENText[256]; // 需要显示的英文
    UINT iFontColor; // 字体颜色。0 ~6 分别对应 红、绿、黄、蓝、紫、青、白等颜色。
    UINT iStayTime; // 停留时间。停留时间，0 ~ 9, 0-最短，9-最长，默认值为 5。
    如果中文/英文一屏可以显示时，此参数有效，同时居中显示。其他情况可采用滚动播放。如果中文/英文可一屏显示，中英文切换时采用立即消失。滚动播放时采用从由右向左方向滚动播放。
} tLED_Download_Text_Para;
```

10.9.4 接口函数

10.9.4.1 函数说明

- 10.9.4.1.1 所有函数返回值只判断传入参数是否为空；
- 10.9.4.1.2 逻辑或物理错误还必须判断 tLEDDevReturn 的 iType 和其他成员变量来确定；
- 10.9.4.1.3 每个函数内部都有线程锁，以保证标准接口函数都是串行运行。

10.9.4.2 通信建立

函数名称: int LED_CommOpen(UINT uiCommPort, UINT uiBaudRate)
功能说明: 建立串口通信, 如参数设置错误或串口被占用, 返回错误码。
参数说明见表 272。

表 272 LED_CommOpen() 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	uiCommPort	In	串口通信号
1	UINT	uiBaudRate	In	波特率

返回值说明见表 273。

表 273 LED_CommOpen() 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:
执行其他操作前, 必须先设置串口。
如果该接口返回失败, 请检查串口ID或者设备逻辑名是否正确以及串口设备是否正确等。

10.9.4.3 初始化

函数名称: int LED_Init(tLEDDevReturn* pDevStatus)
功能说明: 初始化模块, 将模块恢复到初始状态等。
参数说明见表 274。

表 274 LED_Init () 参数说明

参数类型	参数名称	属性	说明
tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 275。

表 275 LED_ Init () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:
设备开机后需要调用本函数。

10.9.4.4 模块复位

函数名称: int LED_Reset(tLEDDevReturn* pDevStatus)
功能说明: 初始化模块, 将模块恢复到初始状态等。
参数说明见表 276。

表 276 LED_ Reset () 参数说明

参数类型	参数名称	属性	说明
tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 277。

表 277 LED_ Reset () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:
在设备故障或从维修模式退出时 (对于支持维修模式的情况), 执行本函数。

10.9.4.5 下载显示信息

函数名称: int LED_Download(tLED_Download_Para* pLED_Download_Para, tLEDDevReturn *pDevStatus)

功能说明: 此命令用于将需要显示的信息下载到LED控制板上或保存在内存变量中。

参数说明见表 278。

表 278 LED_Download () 参数说明

序号	参数类型	参数名称	属性	说明
1	tLED_Download_Para*	pLED_Download_Para	In	需下载的数据
2	tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 279。

表 279 LED_Download () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:

每次初始化完成后需重新调用此接口进行下载, 再调用显示信息接口进行内容的显示。

10.9.4.6 显示信息

函数名称: int LED_Display(UINT nSerialNO, tLEDDevReturn * pDevStatus)

功能说明: 此命令用于根据选择的序号将保存在LED控制板上或内存变量中的字符串内容进行显示。

要求显示内容切换时, 需立即显示。

参数说明见表 280。

表 280 LED_Display () 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	nSerialNO	In	需显示的数据编号。与 tLED_Download_Text_Para::nSerialNO 对应。
2	tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 281。

表 281 LED_Display () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.9.4.7 获取模块状态

函数名称: int LED_GetDevStatus(tLEDModuleStatus *pModuleStatus, tLEDDevReturn *pDevStatus)

功能说明: 调用此接口可获取模块当前状态。

参数说明见表 282。

表 282 LED_GetDevStatus () 参数说明

序号	参数类型	参数名称	属性	说明
1	tLEDModuleStatus *	pModuleStatus	Out	返回模块状态信息
2	tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 283。

表 283 LED_GetDevStatus () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.9.4.8 获取模块驱动版本信息

函数名称: int LED_GetVersion(char cFirmWareVersion[32], char cDriverVersion[32], tLEDDevReturn *pDevStatus);
功能说明: 获取设备驱动版本号、固件版本号、状态信息。
参数说明见表 284。

表 284 LED_GetVersion () 参数说明

序号	参数类型	参数名称	属性	说明
1	char	cFirmWareVersion[32]	out	返回固件版本号
2	char	cDriverVersion[32]	out	返回驱动版本号
3	tLEDDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 285。

表 285 LED_GetVersion () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.9.4.9 通信关闭

函数名称: int LED_CommClose()
功能说明: 调用此接口可关闭底层串口通信。
参数说明见下表: 无
返回值说明见表 286。

表 286 LED_CommClose () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.10 打印机模块接口

10.10.1 接口汇总表

表 287 打印机模块接口汇总表

接口名称	接口说明
PRT_Init()	初始化
PRT_Reset()	模块复位
PRT_Print()	打印
PRT_GetDevStatus()	读取模块状态
PRT_GetVersion()	读取模块驱动版本信息
PRT_CommOpen()	通讯建立
PRT_CommClose()	通讯关闭

10.10.2 设备返回码的定义及其说明

表 288 打印机模块错误代码汇总表

错误类型(uiType)	错误代码(uiStdErrorCode)	说明
1	1611	打印机少纸
1	1612	警告错误
2	1651	无效命令(没有调用串口打开时也返回该错误代码)
2	1652	打印机无纸
2	1653	打印切纸错
2	1654	打印机故障
2	1655	卡纸

2	1656	串口配置错
2	1657	通信超时
2	1658	传入参数错误

10.10.3 结构体说明

10.10.3.1 错误与状态描述结构体

```
typedef struct
{
    //1. 基本信息
    UINT uiProviderId; //模块厂家标识。三星 0x02、广电 0x04、中软 0x06、欧姆龙 0x05;
    UINT uiModuleType; //模块类型标识 0x07 打印机模块
    UINT uiType; //错误类型：0-正常；1-警告；2-故障
    UINT uiStdErrorCode[16]; //统一维护的错误代码
    UINT uiErrorCode[16]; //错误代码 （由不同模块厂家提供）
    char acDevReturn[128]; // 硬件返回信息
    char acReserve[128]; // 保留信息
} tPRTDevReturn;
```

10.10.3.2 模块状态结构体

```
typedef struct
{
    UINT uiPaperStatus; //打印纸状态：0-正常；1-少纸；2-无纸
    UINT uiModuleStatus; //模块状态：0-正常；1-故障
} tPRTModuleStatus;
```

10.10.4 接口函数

10.10.4.1 函数说明

- 10.10.4.1.1 所有函数返回值只判断传入参数是否为空；
- 10.10.4.1.2 逻辑或物理错误还必须判断 tPRTDevReturn 的 iType 和其他成员变量来确定；
- 10.10.4.1.3 每个函数内部都有线程锁，以保证标准接口函数都是串行运行。

10.10.4.2 初始化

函数名称：Int PRT_Init(tPRTDevReturn* pDevStatus)
功能说明：初始化模块，检测模块各硬件、将模块恢复到初始状态等。
参数说明见表 289。

表 289 PRT_Init() 参数说明

参数类型	参数名称	属性	说明
tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 290。

表 290 PRT_Init() 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

注意事项：
设备开机后需要调用本函数。

10.10.4.3 模块复位

函数名称：Int PRT_Reset(tPRTDevReturn* pDevStatus)
功能说明：初始化模块，检测模块各硬件、将模块恢复到初始状态等。

参数说明见表 291。

表 291 PRT_Reset () 参数说明

参数类型	参数名称	属性	说明
tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 292。

表 292 PRT_Reset () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

注意事项：
在设备故障或从维修模式退出时（对于支持维修模式的情况），执行本函数。

10.10.4.4 打印

函数名称：int PRT_Print(char* cPrintData, bool bCutPaper, tPRTDevReturn *pDevStatus)
功能说明：此命令用于将输入的字符串进行打印。
参数说明见表 293。

表 293 PRT_Print 参数说明

序号	参数类型	参数名称	属性	说明
1	Char*	cPrintData	In	需打印的数据。必须/0 结尾。
2	bool	bCutPaper	In	是否切纸。true：切纸；false：不切纸。
3	tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 294。

表 294 PRT_Print () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

10.10.4.5 获取模块状态

函数名称：int PRT_GetDevStatus(tPRTModuleStatus *pModuleStatus, tPRTDevReturn *pDevStatus)
功能说明：调用此接口可获取模块当前状态。
参数说明见表 295。

表 295 PRT_GetDevStatus () 参数说明

序号	参数类型	参数名称	属性	说明
1	tPRTModuleStatus *	pModuleStatus	Out	返回模块状态信息
2	tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 296。

表 296 PRT_GetDevStatus () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

10.10.4.6 获取模块驱动版本信息

函数名称：int PRT_GetVersion(char cFirmWareVersion[32], char cDriverVersion[32], tPRTDevReturn *pDevStatus);
功能说明：获取设备驱动版本号、固件版本号、状态信息。
参数说明见表 297。

表 297 PRT_GetVersion () 参数说明

序号	参数类型	参数名称	属性	说明
1	char	cFirmWareVersion[32]	out	返回固件版本号
2	char	cDriverVersion[32]	out	返回驱动版本号
3	tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 298。

表 298 PRT_GetVersion () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.10.4.7 通信建立

函数名称: int PRT_CommOpen(UINT uiCommPort, UINT uiBaudRate)
功能说明: 建立串口通信, 如参数设置错误或串口被占用, 返回错误码。
参数说明见表 299。

表 299 PRT_CommOpen () 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	uiCommPort	In	串口通信号
2	UINT	uiBaudRate	In	波特率

返回值说明见表 300。

表 300 PRT_CommOpen () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:
执行其他操作前, 必须先设置串口。
如果该接口返回失败, 请检查串口ID或者设备逻辑名是否正确以及串口设备是否正确等。

10.10.4.8 通信关闭

函数名称: int PRT_CommClose()
功能说明: 调用此接口可关闭底层串口通信。
参数说明: 无
返回值说明见表 301。

表 301 PRT_CommClose () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11 后台维护终端模块接口

10.11.1 接口汇总表

表 302 后台维护终端模块接口汇总表

接口名称	接口说明
Backer_CommOpen()	通讯建立
Backer_Init()	初始化
Backer_Reset()	模块复位
Backer_Light()	点亮/关闭显示屏
Backer_ShowString()	显示字符串
Backer_GetStringInput()	获取输入字符串
Backer_GetKeyInput()	获取输入按键

Backer_Clear()	清空屏幕
Backer_GetDevStatus()	读取模块状态
Backer_GetVersion()	读取模块驱动版本信息
Backer_CommClose()	通讯关闭

10.11.2 结构体说明

10.11.2.1 错误与状态描述结构体

```
typedef struct
{
    //1. 基本信息
    UINT uiProviderId; //模块厂家标识。三星 0x02、广电 0x04、中软 0x06、欧姆龙 0x05;
    UINT uiModuleType; //模块类型标识 0x09 后台维护终端模块
    UINT uiType; //错误类型：0-正常；1-警告；2-故障
    UINT uiStdErrorCode[16]; //统一维护的错误代码
    UINT uiErrorCode[16]; //错误代码 （由不同模块厂家提供）
    char acDevReturn[128]; // 硬件返回信息
    char acReserve[128]; // 保留信息
} tBackerDevReturn;
```

10.11.2.2 模块状态结构体

```
typedef struct
{
    UINT uiModuleStatus; //模块状态：0-正常；1-故障
} tBackerModuleStatus;
```

10.11.3 接口函数

10.11.3.1 函数说明

- 10.11.3.1.1 所有函数返回值只判断传入参数是否为空；
- 10.11.3.1.2 逻辑或物理错误还必须判断 tBackerDevReturn 的 iType 和其他成员变量来确定；
- 10.11.3.1.3 每个函数内部都有线程锁，以保证标准接口函数都是串行运行。

10.11.3.2 通信建立

函数名称：int Backer_CommOpen(UINT uiCommPort, UINT uiBaudRate)
功能说明：建立串口通信，如参数设置错误或串口被占用，返回错误码。
参数说明见表 303。

表 303 Backer_CommOpen () 参数说明

序号	参数类型	参数名称	属性	说明
1	UINT	uiCommPort	In	串口通信号
2	UINT	uiBaudRate	In	波特率

返回值说明见表 304。

表 304 Backer_CommOpen () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

注意事项：
执行其他操作前，必须先设置串口。
如果该接口返回失败，请检查串口ID或者设备逻辑名是否正确以及串口设备是否正确等。

10.11.3.3 初始化

函数名称: Int Backer_Init(int &nLineNum, int &nColumnNum, tPRTDevReturn* pDevStatus)

功能说明: 初始化模块, 检测模块各硬件、将模块恢复到初始状态等。同时返回当前显示屏的行数和列数。

参数说明见表 305。

表 305 Backer_Init () 参数说明

序号	参数类型	参数名称	属性	说明
1	int&	nLineNum	Out	返回屏幕行数
2	int&	nColumnNum	Out	返回屏幕列数
3	tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 306。

表 306 Backer_Init () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:

设备开机后需要调用本函数。

10.11.3.4 模块复位

函数名称: Int Backer_Reset(tPRTDevReturn* pDevStatus)

功能说明: 初始化模块, 检测模块各硬件、将模块恢复到初始状态等。

参数说明见表 307。

表 307 Backer_Reset () 参数说明

参数类型	参数名称	属性	说明
tPRTDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 308。

表 308 Backer_Reset () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

注意事项:

在设备故障或从维修模式退出时 (对于支持维修模式的情况), 执行本函数设备开机后需要调用本函数。

10.11.3.5 点亮/关闭显示屏

函数名称: int Backer_Light(bool bLightOpen, tBackerDevReturn* pDevStatus)

功能说明: 此命令用于将显示屏幕点亮或关闭。

参数说明见表 309。

表 309 Backer_Light () 参数说明

序号	参数类型	参数名称	属性	说明
1	bool	bLightOpen	In	是否点亮显示屏。true: 点亮; false: 关闭。
2	tBackerDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 310。

表 310 Backer_Light () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11.3.6 显示字符串

函数名称: int Backer_ShowString (int nLine, int nColumn, char* cShowString, tBackerDevReturn* pDevStatus)

功能说明: 此命令用于将所需的字符串再屏幕上特定坐标显示, 不支持自动换行, 超出屏幕信息将截断。

参数说明见表 311。

表 311 Backer_ShowString () 参数说明

序号	参数类型	参数名称	属性	说明
1	Int	nLine	In	开始显示的行号
2	Int	nColumn	In	开始显示的列号
3	Char*	cShowString	In	需要显示的字符串
4	tBackerDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 312。

表 312 Backer_ShowString () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11.3.7 获取输入字符串

函数名称: int Backer_GetStringInput (int nLine, int nColumn, bool bIsPassword, char* cInput, int &nLength, int &nType, tBackerDevReturn* pDevStatus, int nTimeOut= INFINITE)

功能说明: 此命令用于获取输入的字符串

注意事项:

1、字符串将在指定nLine和nColumn位置开始输入, 支持退格键 (输入过程中删除前1字符), 输入完成后按Enter确认或按ESC取消输入

2、未输入任何数据直接按Enter, 输出为"Enter"

3、按ESC键后返回成功, 输出为"Esc"

参数说明见表 313。

表 313 Backer_GetStringInput () 参数说明

序号	参数类型	参数名称	属性	说明
1	Int	nLine	In	开始显示的行号
2	Int	nColumn	In	开始显示的列号
3	Bool	bIsPassword	In	是否是获取密码输入
4	Char*	cInput	Out	已输入的字符串
5	int&	nLength	in/out	返回输入数据长度
6	int&	nType	Out	输入的类型 1-字符串 2-仅 Enter 键 3-取消 4-超时
7	tBackerDevReturn *	pDevStatus	Out	返回状态信息
8	int	nTimeOut	in	超时时间, 单位微妙

返回值说明见表 314。

表 314 Backer_GetStringInput () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11.3.8 获取输入按键

函数名称: int Backer_GetKeyInput (iunsigned cha&r cInput, tBackerDevReturn* pDevStatus,int nTimeOut= INFINITE)

功能说明: 此命令用于获取输入的按键。

注意事项: cInput为按键的ASCII码, 获取按键失败或超时该值为'\0'

参数说明见表 315。

表 315 Backer_GetKeyInput () 参数说明

序号	参数类型	参数名称	属性	说明
1	unsigned Char&	cInput	Out	按键的 ASCII
2	tBackerDevReturn *	pDevStatus	Out	返回状态信息
3	int	nTimeOut	In	输入超时的时间, 单位微妙

返回值说明见表 316。

表 316 Backer_GetKeyInput () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11.3.9 清空屏幕

函数名称: int Backer_Clear(int nLine,int nColumn,int nWidth,int nHeight,tBackerDevReturn* pDevStatus)

功能说明: 此命令用于清空屏幕的特定区域。

参数说明见表 317。

表 317 Backer_Clear () 参数说明

序号	参数类型	参数名称	属性	说明
1	Int	nLine	In	清空区域的开始行号
2	Int	nColumn	In	清空区域的开始列号
3	Int	nWidth	In	清空区域的宽度 (列数)
4	Int	nHeight	In	清空区域的高度 (行数)
5	tBackerDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 318。

表 318 Backer_Clear () 返回值说明

返回值类型	说明
Int	0: 执行成功; 1, 执行失败。

10.11.3.10 获取模块状态

函数名称: int Backer_GetDevStatus(tBackerModuleStatus *pModuleStatus, tBackerDevReturn *pDevStatus)

功能说明: 调用此接口可获取模块当前状态。

参数说明见表 319。

表 319 Backer_GetDevStatus () 参数说明

序号	参数类型	参数名称	属性	说明
1	tBackerModuleStatus *	pModuleStatus	Out	返回模块状态信息
2	tBackerDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 320。

表 320 Backer_GetDevStatus () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

10.11.3.11 获取模块驱动版本信息

函数名称：int Backer_GetVersion(char cFirmWareVersion[32], char cDriverVersion[32], tBackerDevReturn *pDevStatus);
功能说明：获取设备驱动版本号、固件版本号、状态信息。
参数说明见表 321。

表 321 Backer_GetVersion () 参数说明

序号	参数类型	参数名称	属性	说明
1	char	cFirmWareVersion[32]	out	返回固件版本号
2	char	cDriverVersion[32]	out	返回驱动版本号
3	tBackerDevReturn *	pDevStatus	Out	返回状态信息

返回值说明见表 322。

表 322 Backer_GetVersion () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

10.11.3.12 通信关闭

函数名称：int Backer_CommClose()
功能说明：调用此接口可关闭底层串口通信。
参数说明：无
返回值说明见表 323。

表 323 Backer_CommClose () 返回值说明

返回值类型	说明
Int	0：执行成功；1，执行失败。

附 录 A
(规范性附录)
票箱设计要求

A.1 范围

本附录内容为标准票箱的设计图纸、标准票箱锁要求、车票发售/回收模块要求和标准票箱电子标签相关要求，设计图纸含装配图及零件图共计46张，图纸涵盖组成票箱的全部57类零件，每张图纸的简要说明详见表A.1（票箱机械明细表），以此附录作为票箱的加工及检测依据。

表 A.1 票箱机械明细表

序号	幅面	图纸序号	名称	数量	型号/规格	类别	材质	厚度	重量（g）	备注/说明
1	A3	1, 2	票箱装配	1	N/A	Assembly	组件		5276.3	票箱总装图
1.1	A3	3	票箱外框架焊件	1	N/A	Assembly	组件		2477.1	外观及保护内框、车票
1.1.1	A3	4	票箱外框架	1	N/A	钣金件	SUS 304	1.5	1886.5	外围框
1.1.2	A4	5	竖把手转轴	2	N/A	机加工件	SUS 304		3.1	转轴
1.1.3	A3	6	票箱前盖	1	N/A	钣金件	SUS 304	1	563.7	前门
1.1.4			1.5 寸铰链	2	N/A	Assembly	组件		10.419	
1.2	A3	7	票箱内框架焊件	1	N/A	Assembly	组件		1103	容纳车票
1.2.1	A3	8	票箱内框架	1	N/A	钣金件	SUS 304	1	940.9	容纳车票
1.2.2	A4	9	内框架加强条	1	N/A	钣金件	SUS 304	1	7.6	上端加强条
1.2.3	A4	10	内框架加强条 2	1	N/A	钣金件	SUS 304	1	4.1	下端加强条
1.2.4	A4	11	定位条	1	N/A	钣金件	SUS 304	2	128.8	托盘定位
1.2.5	A4	12	安全票箱盖导轨（右）	1	N/A	钣金件	SUS 304	1	10.778	上盖右导轨
1.2.6	A4	13	安全票箱盖导轨（左）	1	N/A	钣金件	SUS 304	1	10.804	上盖左导轨
1.3	A3	14	票箱加强筋	1	N/A	钣金件	SUS 304	1	623.4	增加票箱前门强度
1.4			把手	1	LS506-100	外购件	SUS 304		115.77	拉手
1.5			票箱锁 MS824-2-180 （永恒）	1	N/A	外购件	锌合金		48.13	
1.6			MS824 锁紧螺母	1	N/A	外购件	锌合金		5.075	
1.7	A4	15	锁孔垫片	2	N/A	钣金件	SUS 304	3	8.455	调整锁体高度
1.8	A4	16	票箱锁杆限位架	1	N/A	钣金件	SUS 304	1.5	23.74	锁杆限位

表 A.1 票箱机械明细表（续）

1.9	A4	17	票箱口封条围框	1	N/A	钣金件	SUS 304	1.5	58.66	票箱口车票限位
1.10	A4	18	票箱装配（票箱滑盖）	1	N/A	Assembly	组件		157.955	上盖
1.10.1	A4	19	票箱滑盖前盖	1	N/A	钣金件	SUS 304	1	78.7	上盖前盖
1.10.2			1.5 寸铰链	1	N/A	Assembly	组件		10.419	
1.10.3	A4	20	票箱滑盖后盖	1	N/A	钣金件	SUS 304	1	36.82	上盖后盖
1.10.4	A4	21	滑盖后盖限位架	1	N/A	钣金件	SUS 304	1	11.983	限制上盖打开角度
1.10.5	A4	22	滑盖前盖限位架	1	N/A	钣金件	SUS 304	1	20.026	限制上盖打开角度
1.11	A4	23	ID 卡塑胶罩	1	N/A	机加工件	PC		14.929	固定和保护 ID 卡
1.12	A4	24	ID 卡塑胶盖	1	N/A	机加工件	PC		4.9	固定和保护 ID 卡
1.13	A4	25	ID 卡塑胶件固定板	1	N/A	钣金件	SUS 304	1	23.3	固定 ID 卡塑胶件
1.14			ID 卡	1	N/A	外购件	ABS		2.2	电子标签
1.15	A4	26	票箱盖限位片	1	N/A	钣金件	SUS 304	3	38.3	限制票箱门打开角度
1.16	A4	27	竖把手组件	1	N/A	Assembly	组件		207.3	提手
1.16.1	A4	28	竖把手支架（新）	2	N/A	钣金件	SUS 304	3	68.5	提手连杆
1.16.2	A4	29	竖把手	1	N/A	机加工件	SUS 304		70.288	提手手柄
1.17	A4	30	轴套	2	N/A	机加工件	SUS 304		1.5	提手定位
1.18	A3	31	托盘装配	1	N/A	Assembly	组件		276.8	车票托块
1.18.1	A3	32	前托盘焊件	1	N/A	Assembly	组件		132.5	车票托块
1.18.1.1	A3	33	托盘	1	N/A	钣金件	SUS 304	1.5	126.5	前托盘钣金件
1.18.1.2	A4	34	小轴承固定杆	4	N/A	机加工件	SUS 304		0.9	安装轴承
1.18.1.3	A4	35	托盘拉簧固定杆	1	N/A	机加工件	SUS 304		2.3	固定拉簧
1.18.2	A4	36	自锁钩固定架焊件	1	N/A	Assembly	组件		52.4	固定托盘自锁钩
1.18.2.1	A4	37	自锁钩固定架	1	N/A	钣金件	SUS 304	1.5	35.9	自锁钩固定架钣金件
1.18.2.2	A4	33	小轴承固定杆	4	N/A	机加工件	SUS 304		0.9	安装轴承
1.18.2.3	A4	38	托盘连接螺柱	2	N/A	机加工件	SUS 304		5.2	与前托盘连接、固定自锁钩
1.18.2.4	A4	39	自锁钩固定杆-2	1	N/A	机加工件	SUS 304		2.4	固定自锁钩
1.18.3	A4	40	托盘自锁钩	1	N/A	钣金件	SUS 304	2	22.6	锁定托块
1.18.4	A4	41	支撑板	1	N/A	机加工件	尼龙 101		51.1	放置车票
1.18.5	A4	42	托盘拉簧	1	N/A	外购件	65Mn		0.564	使自锁钩处于锁定状态
1.18.6			尼龙垫片 Φ8	6	Φ8X16X1.5	外购件	尼龙 6/10		0.3	
1.18.7			十字槽盘头组合螺钉	4	GB9074.4 M3X10	紧固件	SUS304		1.07	
1.18.8			十字槽盘头组合螺钉	2	GB9074.4 M4X10	紧固件	SUS 304		2.2	

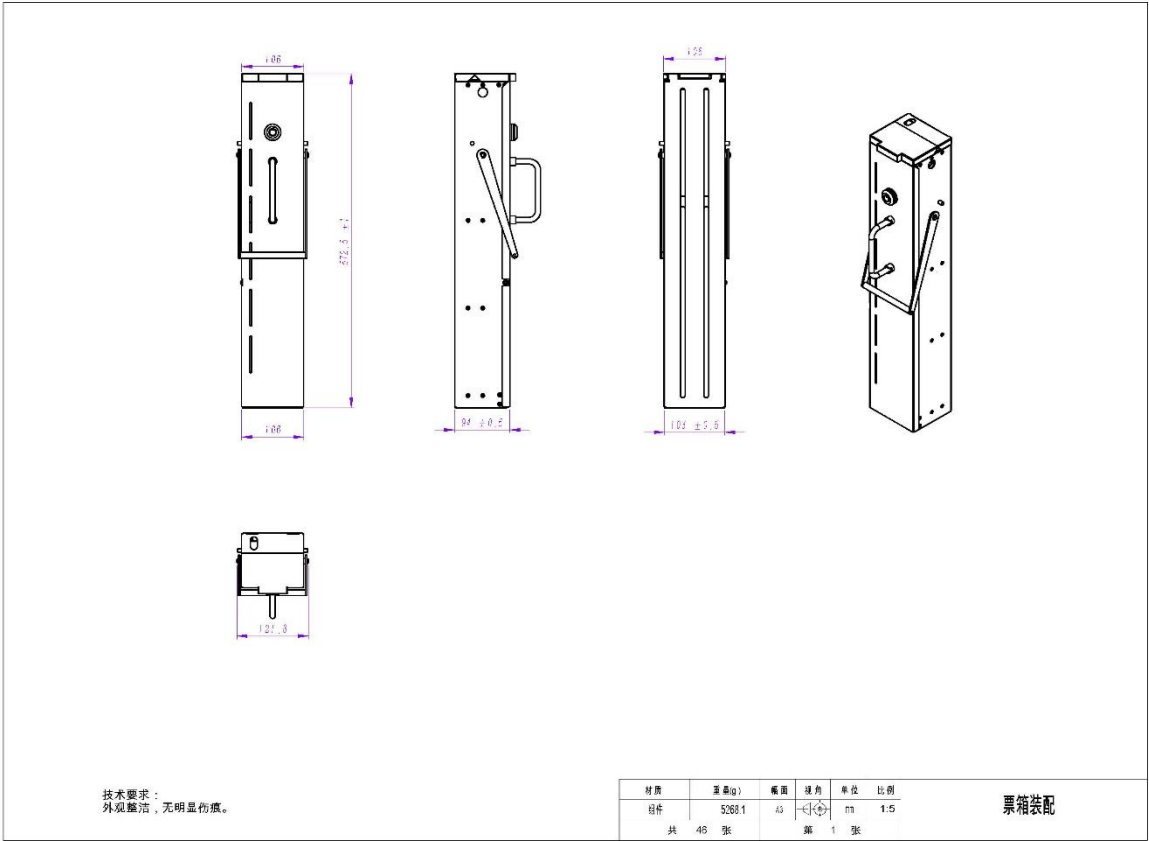
表 A.1 票箱机械明细表（续）

1. 18. 9			开口挡圈	3	GB896 6	紧固件	SUS 304		0. 42	
1. 18. 10			开口挡圈	8	GB896 3	紧固件	SUS 304		0. 1	
1. 18. 11			深沟球轴承 MR84zz	8	N/A	外购件	SUS 304		0. 65	
1. 19	A4	43	锁杆	1	N/A	机加工件	SUS 304		9. 7	
1. 20	A4	44	票箱锁凸轮片	1	N/A	钣金件	SUS 304	3	29. 2	
1. 21	A4	45	票箱口垫块	2	N/A	钣金件	SUS 304	5	7. 9	
1. 22	A4	46	票箱锁压簧	1	N/A	外购件	65Mn		0. 6	
1. 23			开口挡圈	1	GB896 4	紧固件	SUS 304		0. 21	
1. 24			开口挡圈	2	GB896 5	紧固件	SUS 304		0. 24	
1. 25			开口挡圈	2	GB896 3	紧固件	SUS 304		0. 1	
1. 26			十字槽沉头螺钉	16	GB819 M3X6	紧固件	SUS 304		0. 44	
1. 27			内六角圆柱头螺钉	4	GB70. 1 M3X8	紧固件	SUS 304		0. 95	
1. 28			平垫圈	4	GB95 4	紧固件	SUS 304		0. 31	
1. 29			平垫圈	8	GB95 3	紧固件	SUS 304		0. 12	
1. 30			弹簧垫圈	8	GB93 3	紧固件	SUS 304		0. 07	
1. 31			六角螺母	8	GB6170 M3	紧固件	SUS 304		0. 39	
1. 32			十字槽沉头螺钉	4	GB819 M3X12	紧固件	SUS 304		0. 78	

A.2 设计图纸

A.2.1 票箱装配1

图A.1规定了票箱装配1设计图纸。

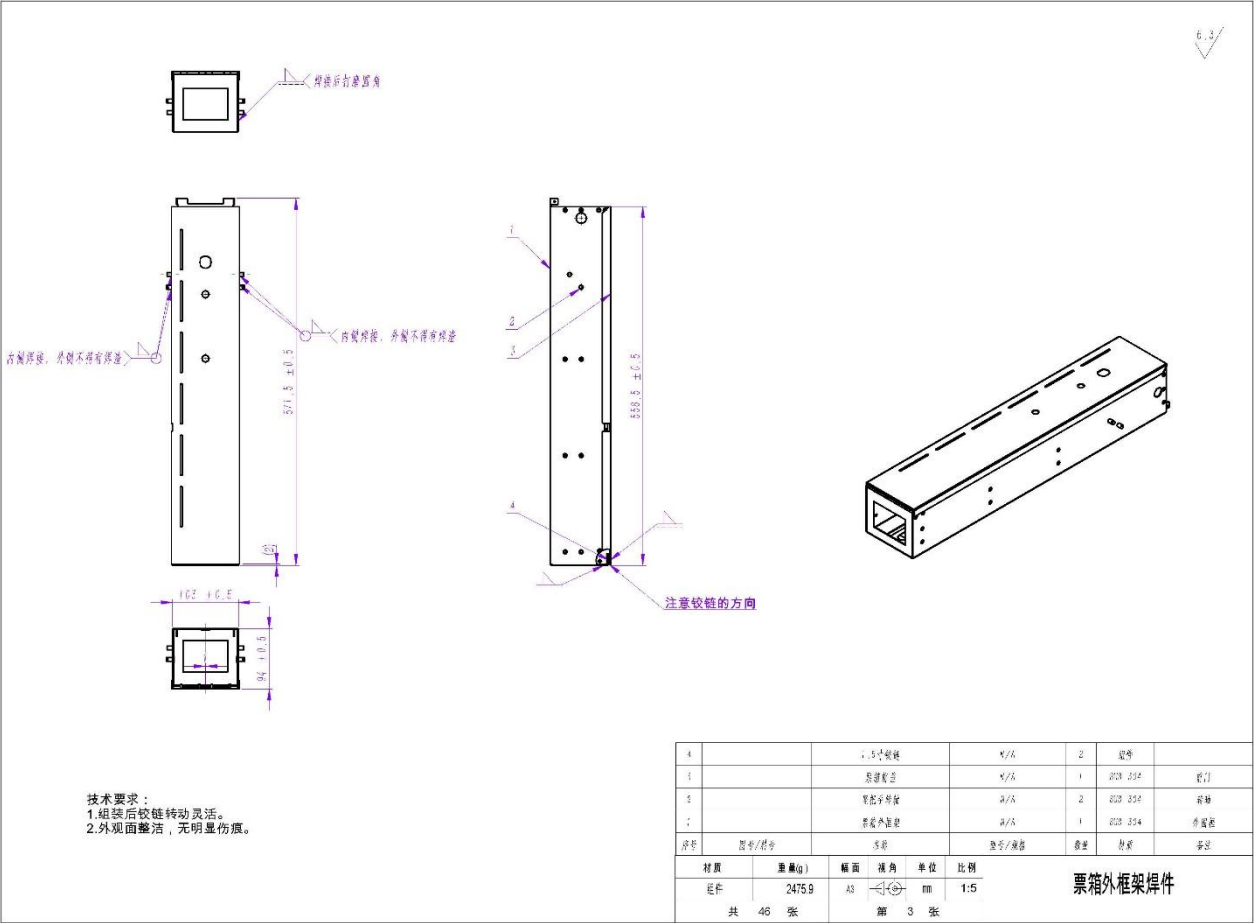


图A.1 票箱装配图 1

A.2.2 票箱装配2

A. 2. 3 票箱外框架焊件

图A. 3规定了票箱外框架焊件设计图纸。

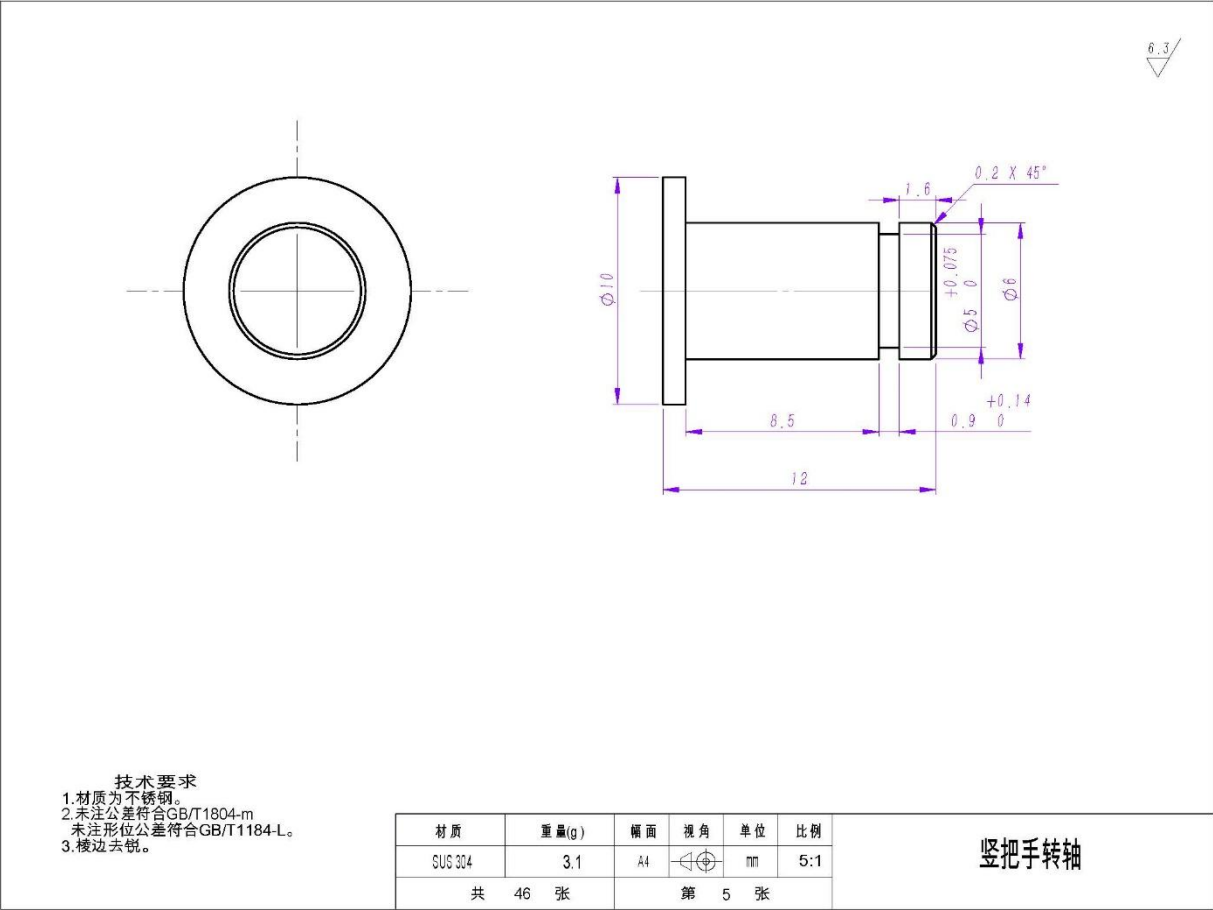


图A. 3 票箱外框架焊件

A. 2. 4 票箱外框架

A. 2. 5 竖把手转轴

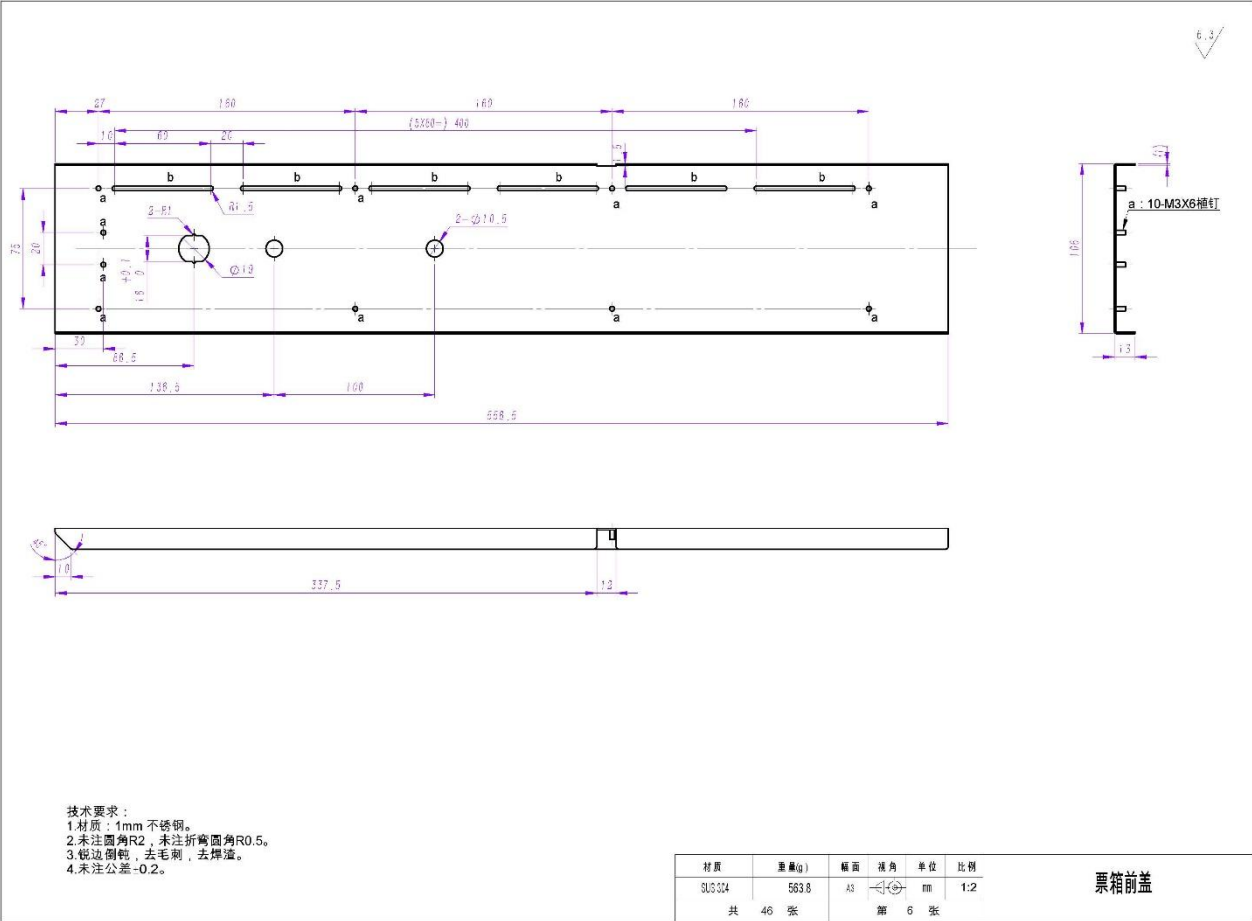
图A. 5规定了竖把手转轴设计图纸。



图A. 5 竖把手转轴

A. 2. 6 票箱前盖

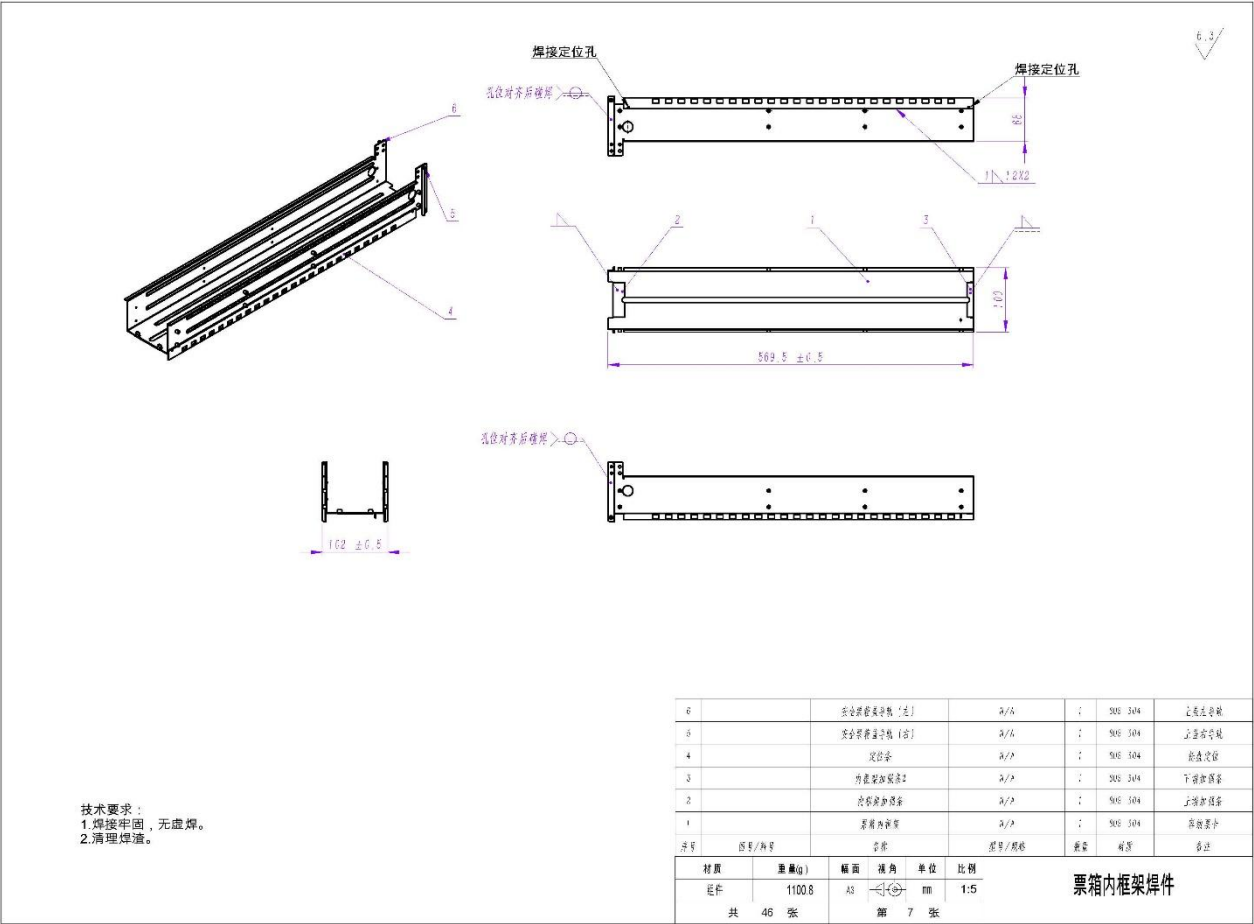
图A.6规定了票箱前盖设计图纸。



图A.6 票箱前盖

A. 2. 7 票箱内框架焊件

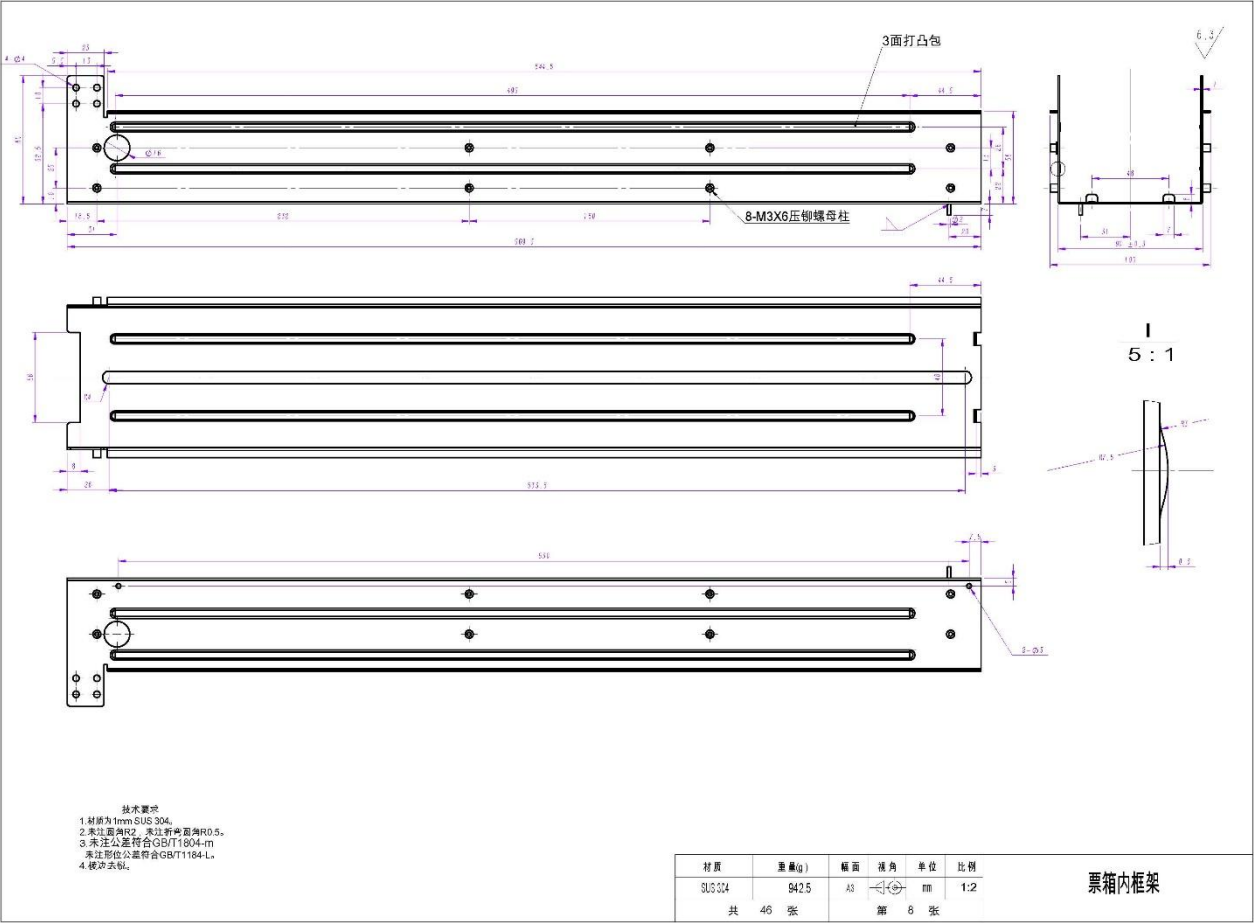
图A. 7规定了票箱内框架焊件设计图纸。



图A. 7 票箱内框架焊件

A. 2. 8 票箱内框架

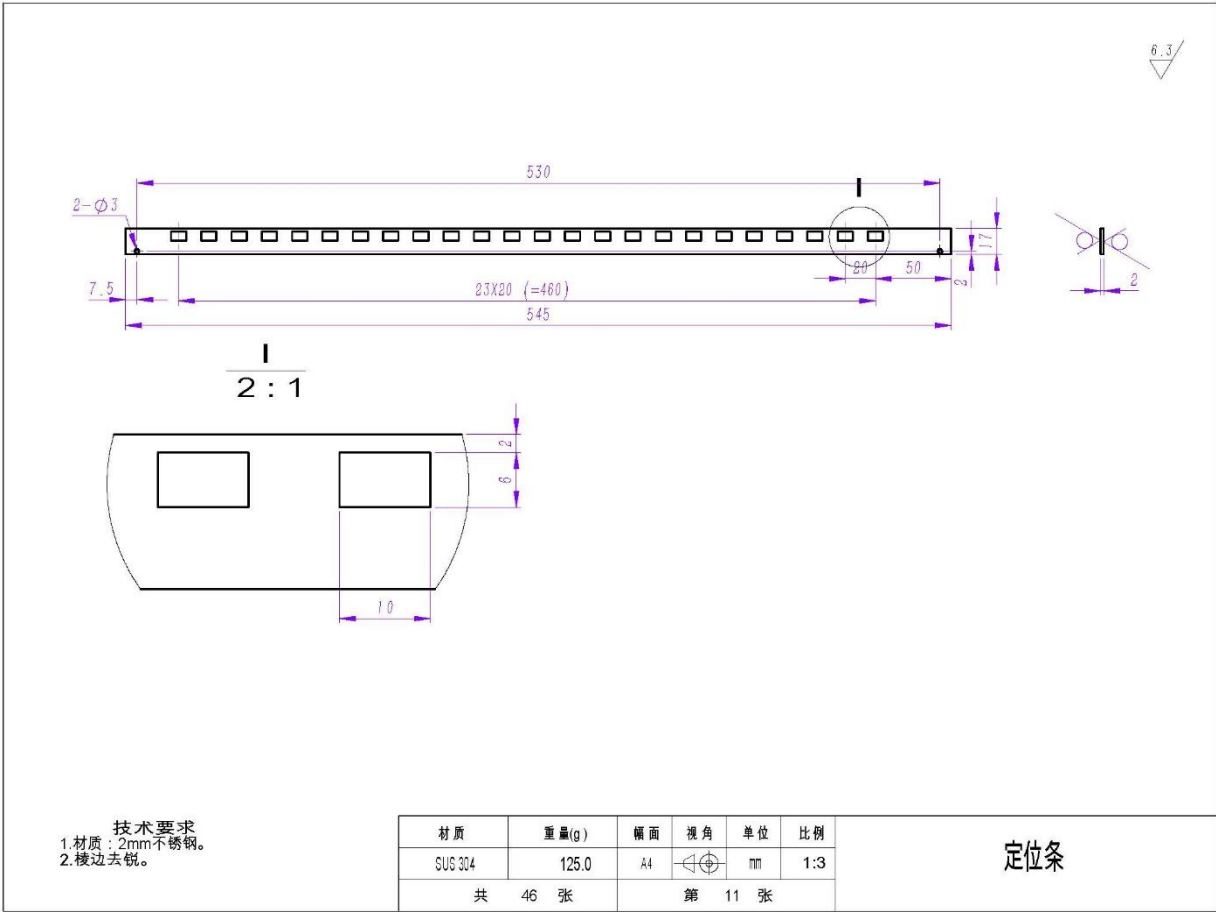
图A.8规定了票箱内框架设计图纸。



图A.8 票箱内框架

A. 2. 11 定位条

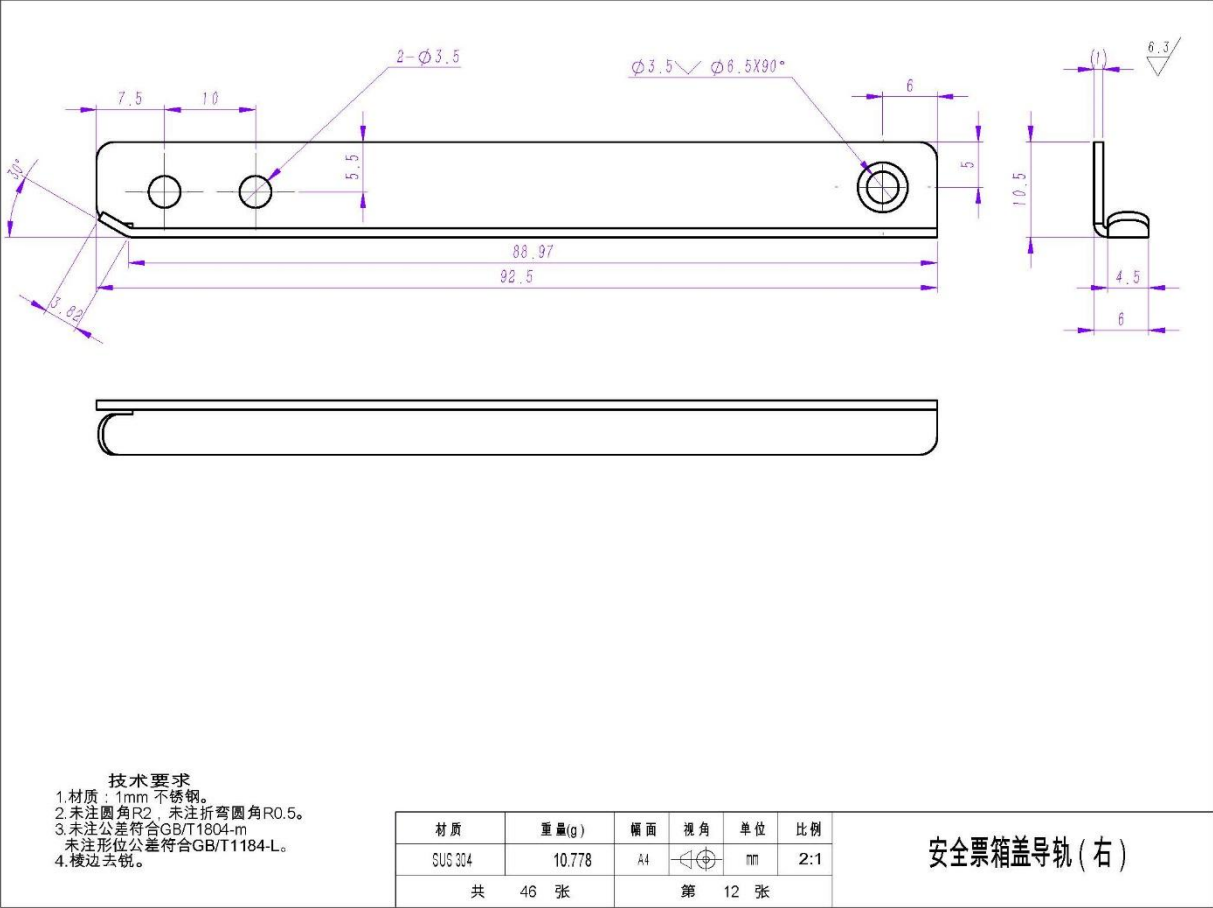
图A. 11规定了定位条设计图纸。



图A. 11 定位条

A. 2. 12 安全票箱盖导轨（右）

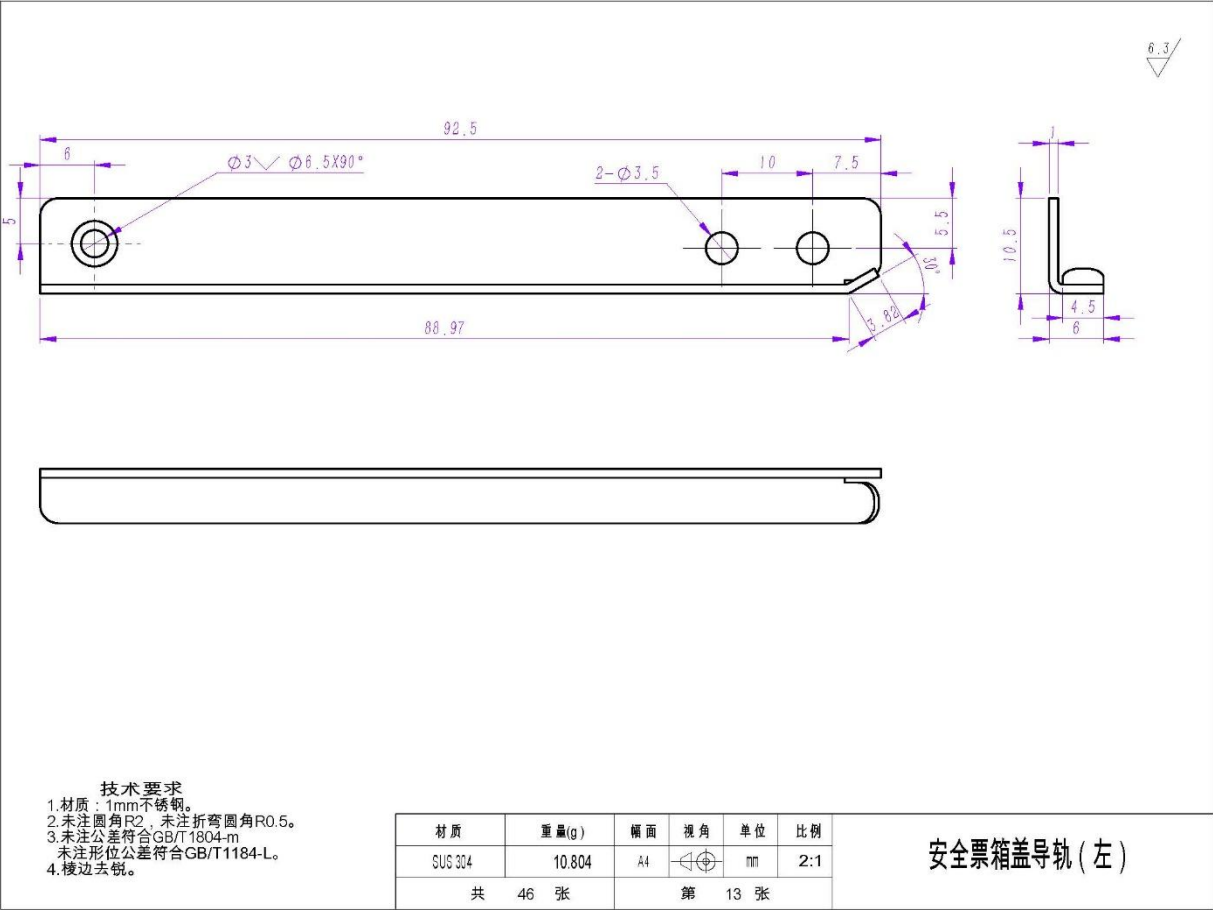
图A. 12规定了安全票箱盖导轨（右）设计图纸。



图A. 12 安全票箱盖导轨（右）

A. 2. 13 安全票箱盖导轨（左）

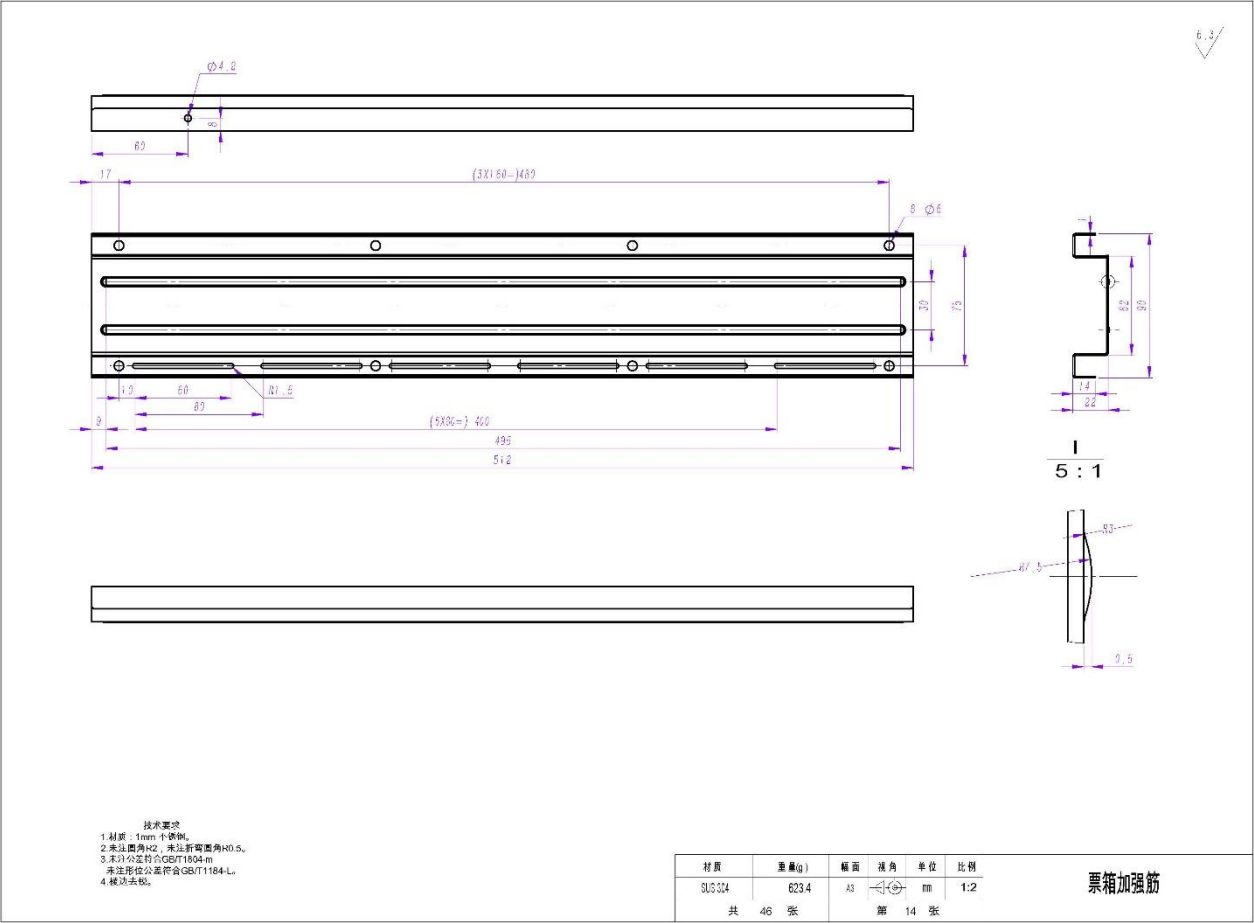
图A. 13规定了安全票箱盖导轨（左）设计图纸。



图A. 13 安全票箱盖导轨（左）

A. 2. 14 票箱加强筋

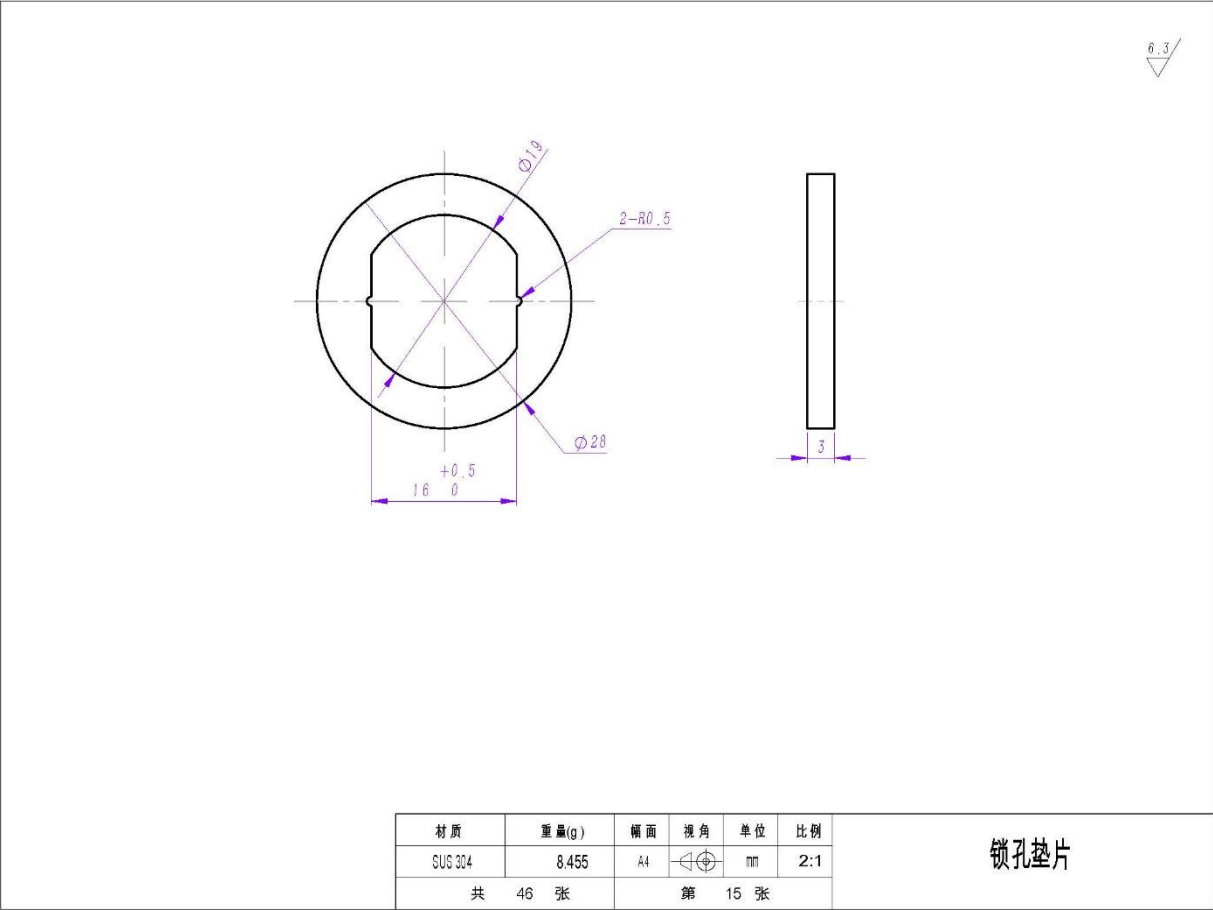
图A. 14规定了票箱加强筋设计图纸。



图A. 14 票箱加强筋

A. 2. 15 锁孔垫片

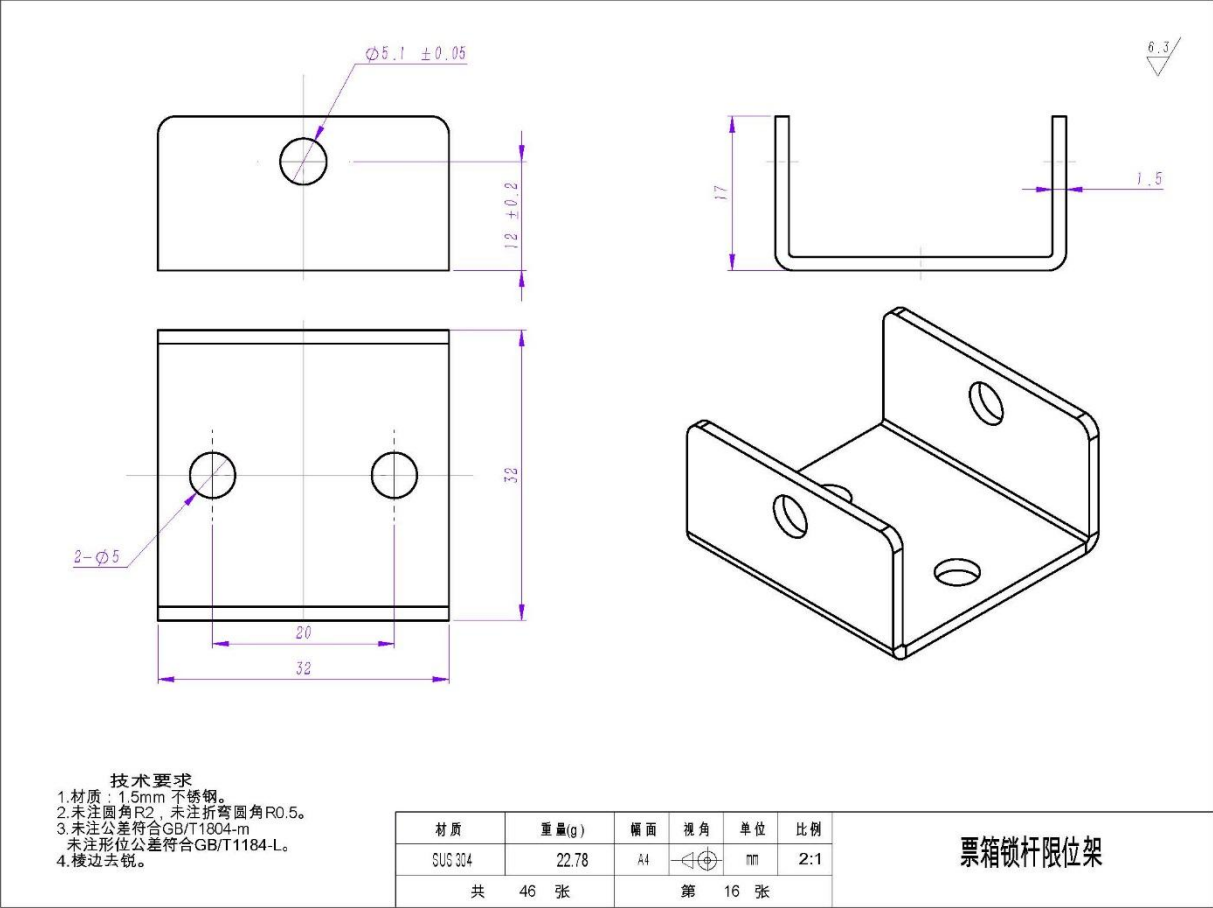
图A. 15规定了锁孔垫片设计图纸。



图A. 15 锁孔垫片

A. 2. 16 票箱锁杆限位架

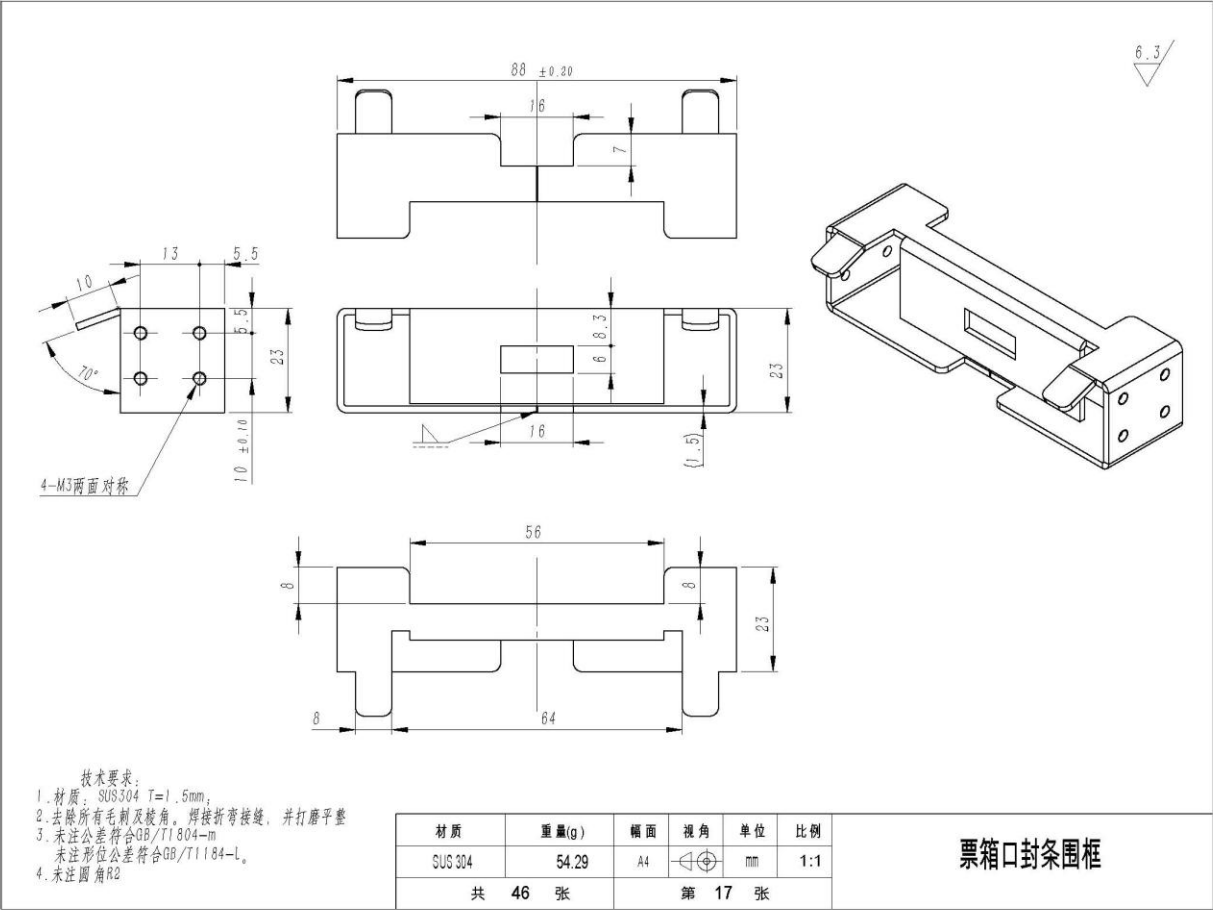
图A. 16规定了票箱锁杆限位架设计图纸。



图A. 16 票箱锁杆限位架

A. 2. 17 票箱口封条围框

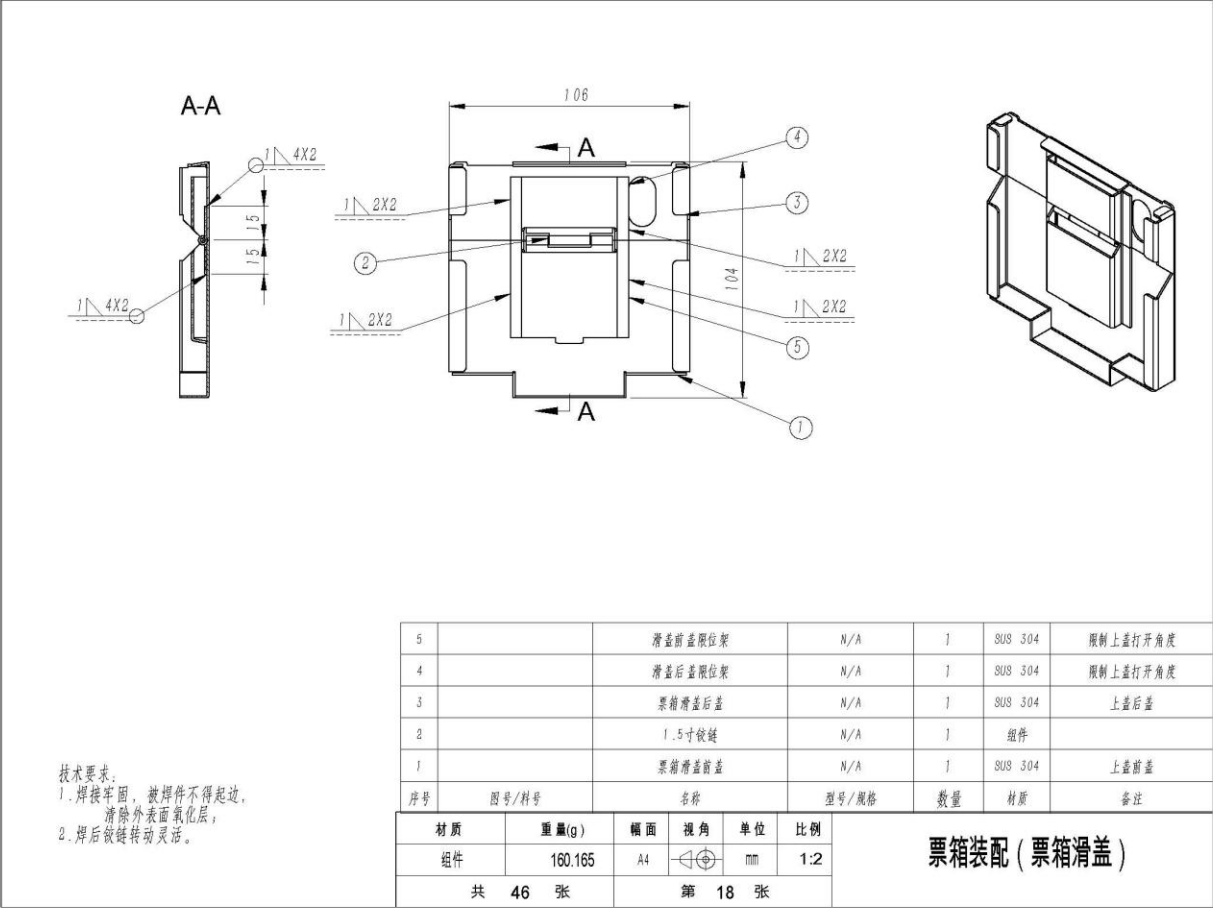
图A. 17规定了票箱口封条围框设计图纸。



图A. 17 票箱口封条围框

A. 2. 18 票箱装配（票箱滑盖）

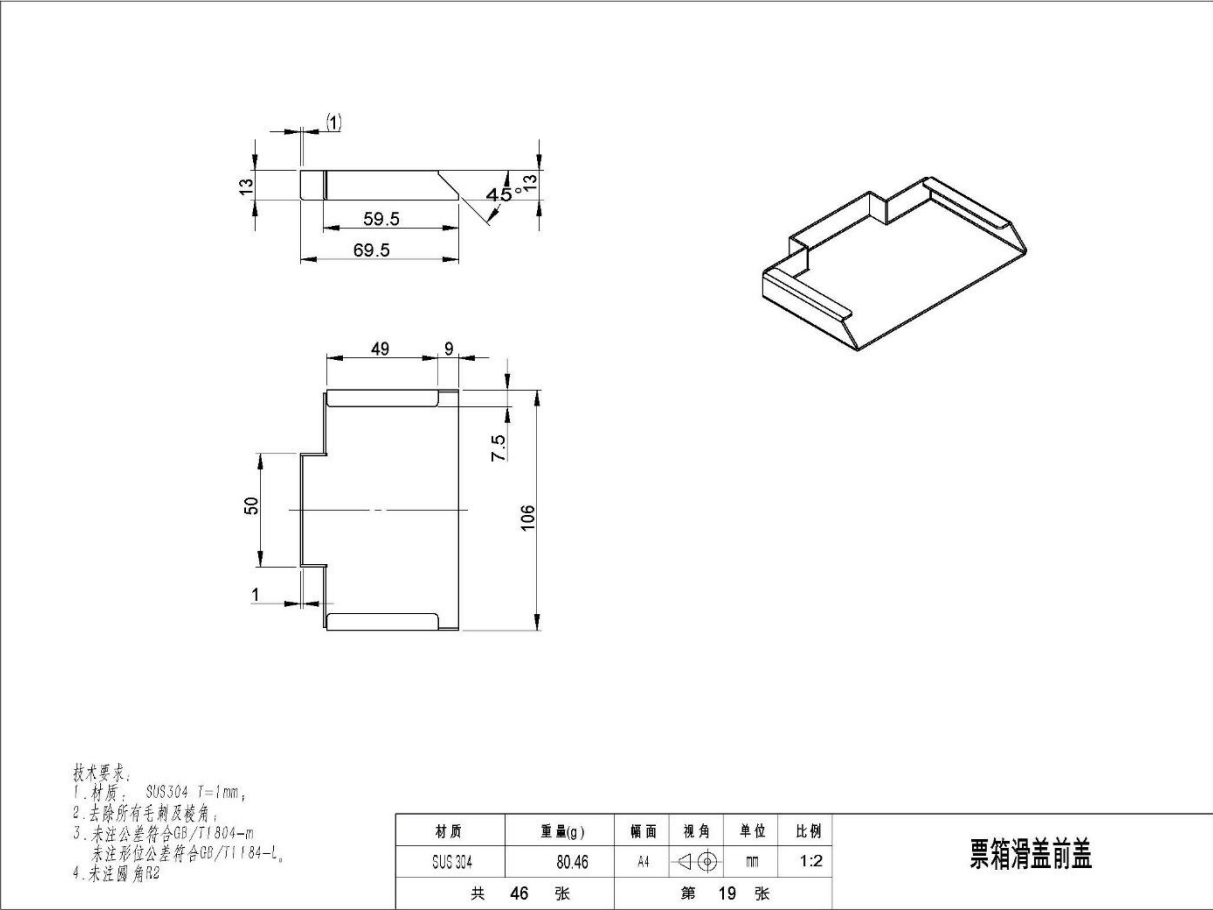
图A. 18规定了票箱装配（票箱滑盖）设计图纸。



图A. 18 票箱装配（票箱滑盖）

A. 2. 19 票箱滑盖前盖

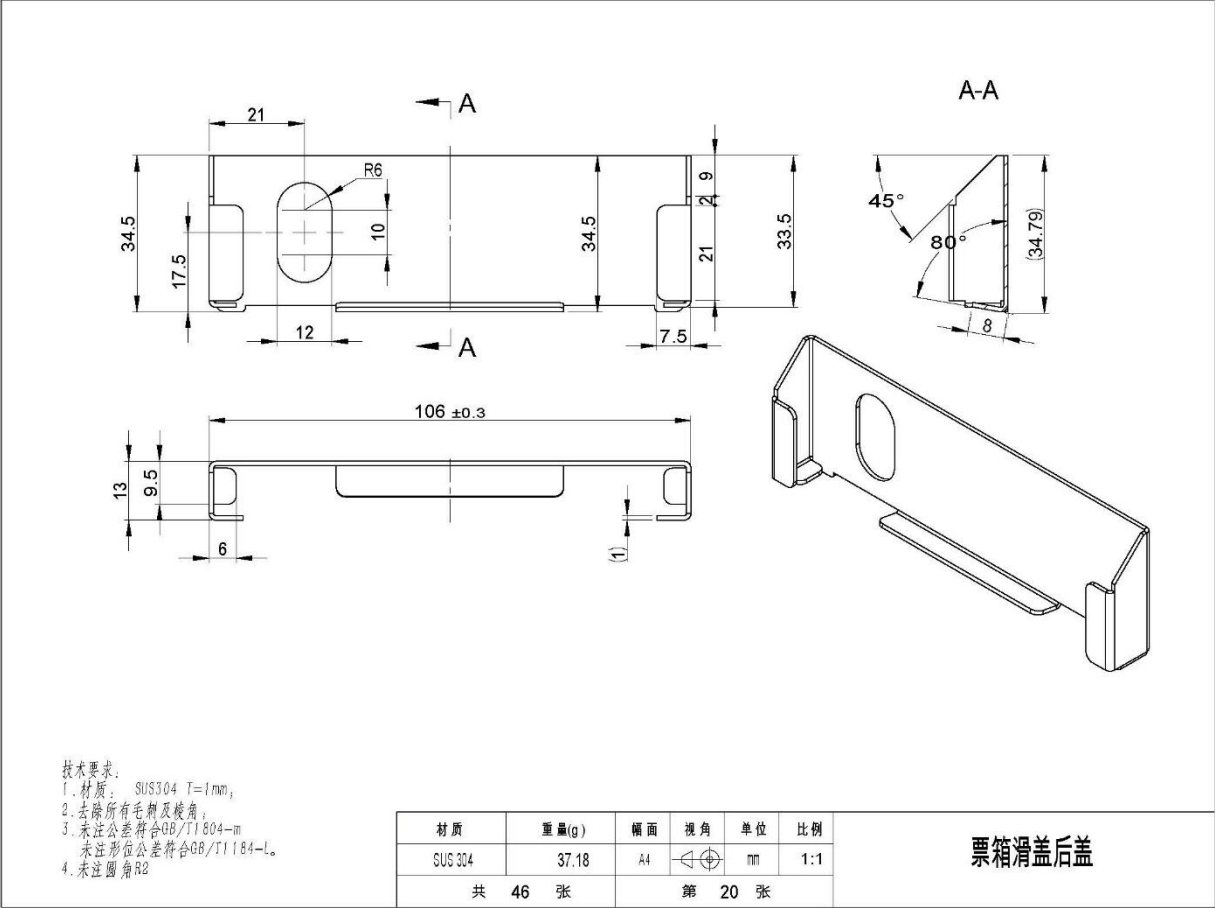
图A. 19规定了票箱滑盖前盖设计图纸。



图A. 19 票箱滑盖前盖

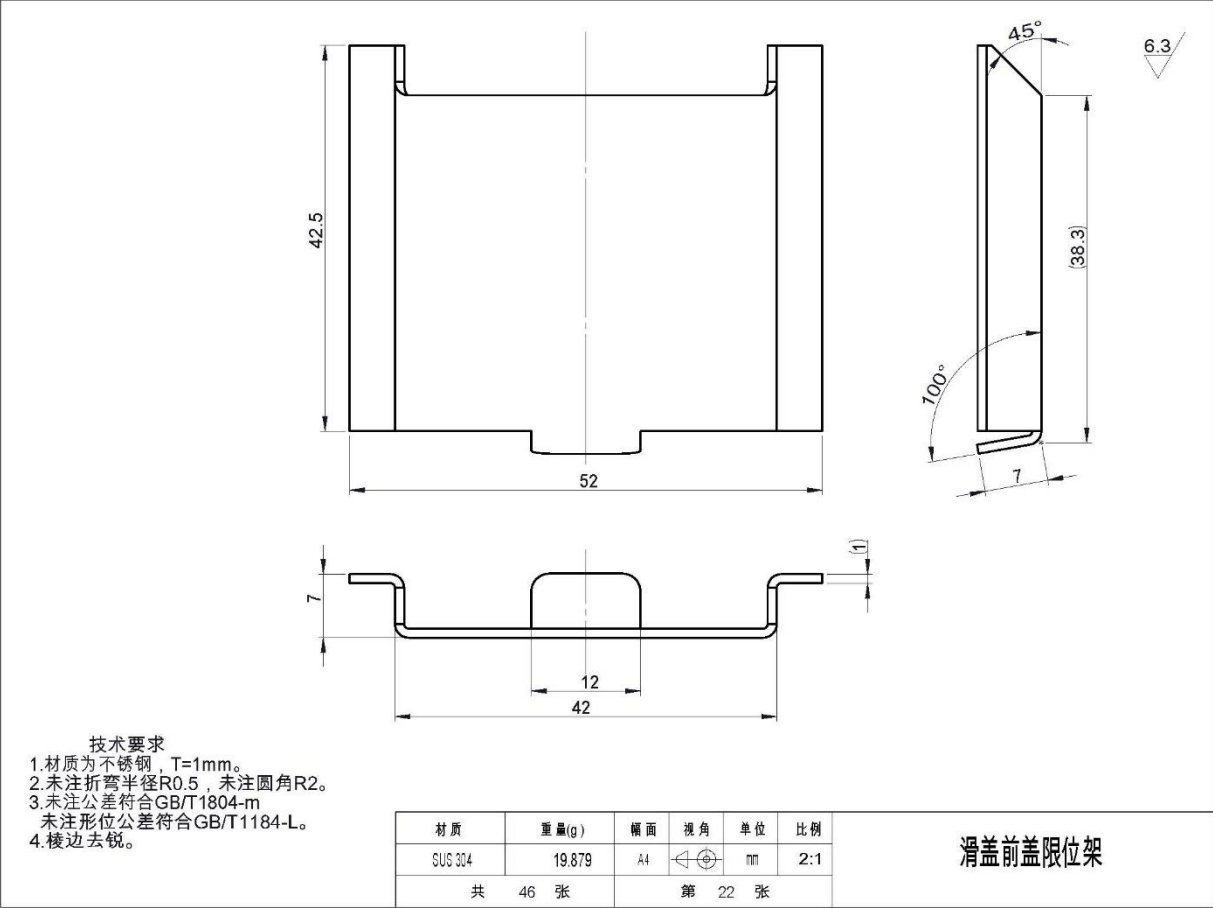
A. 2. 20 票箱滑盖后盖

图A. 20规定了票箱滑盖后盖设计图纸。



图A. 20 票箱滑盖后盖

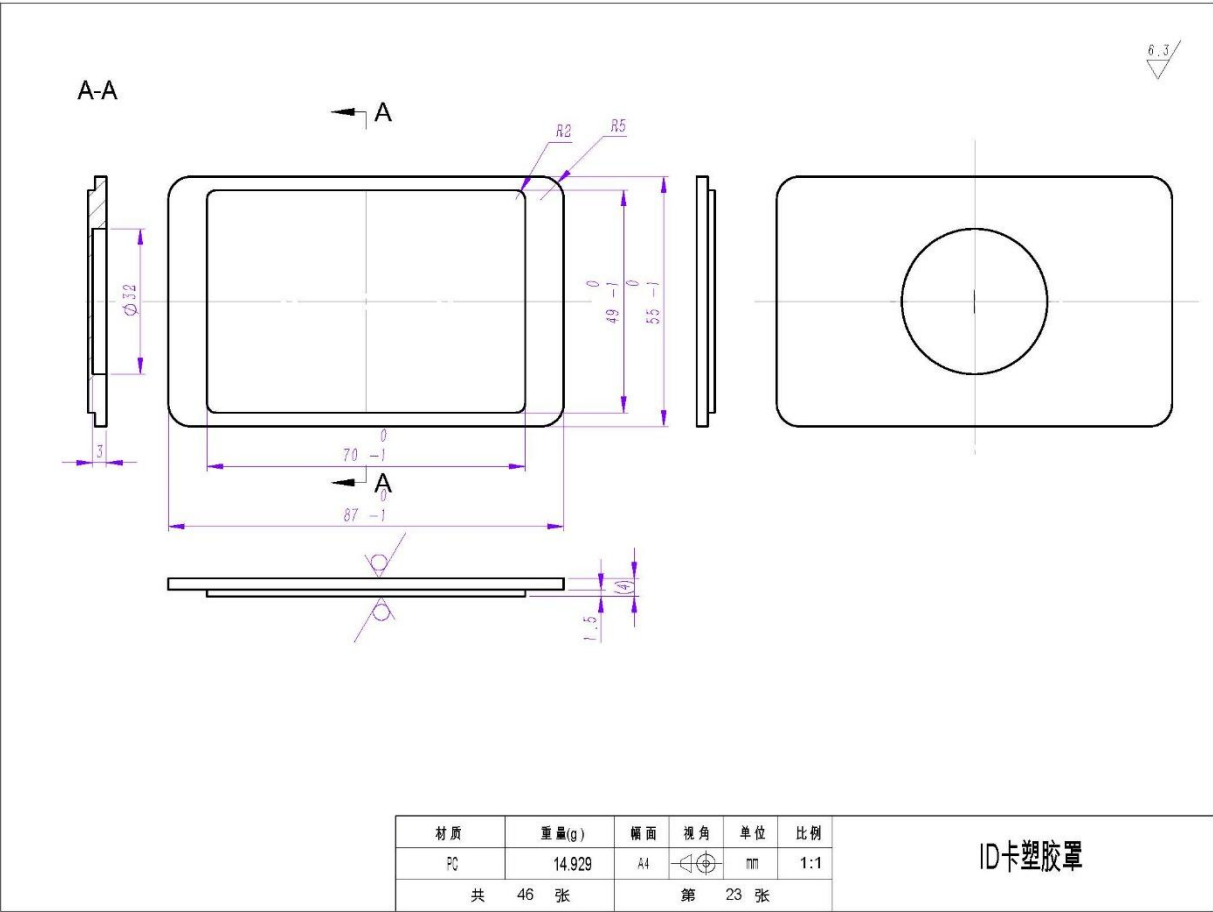
图A. 22规定了滑盖前盖限位架设计图纸。



图A. 22 滑盖前盖限位架

A. 2. 23 ID卡塑胶罩

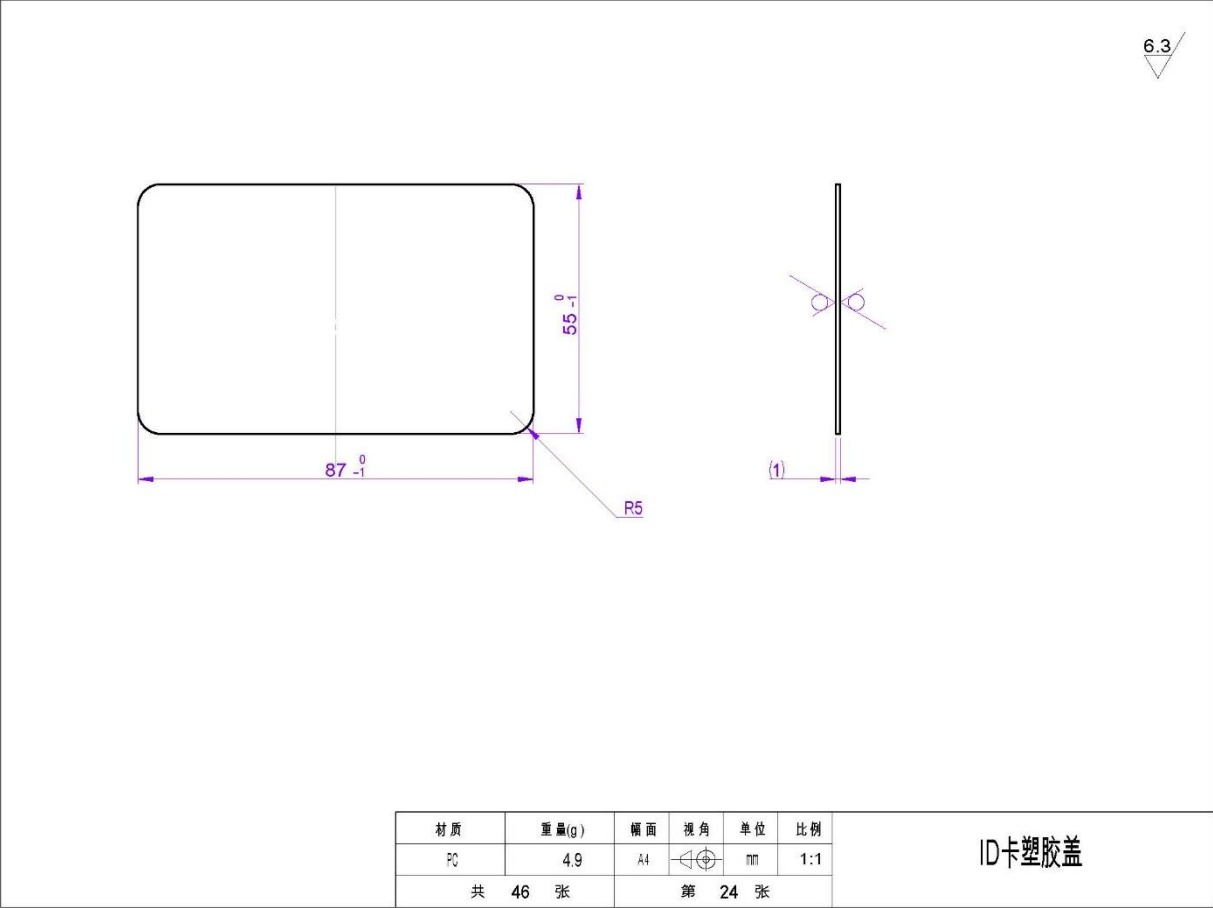
图A. 23规定了ID卡塑胶罩设计图纸。



图A. 23 ID卡塑胶罩

A. 2. 24 ID卡塑胶盖

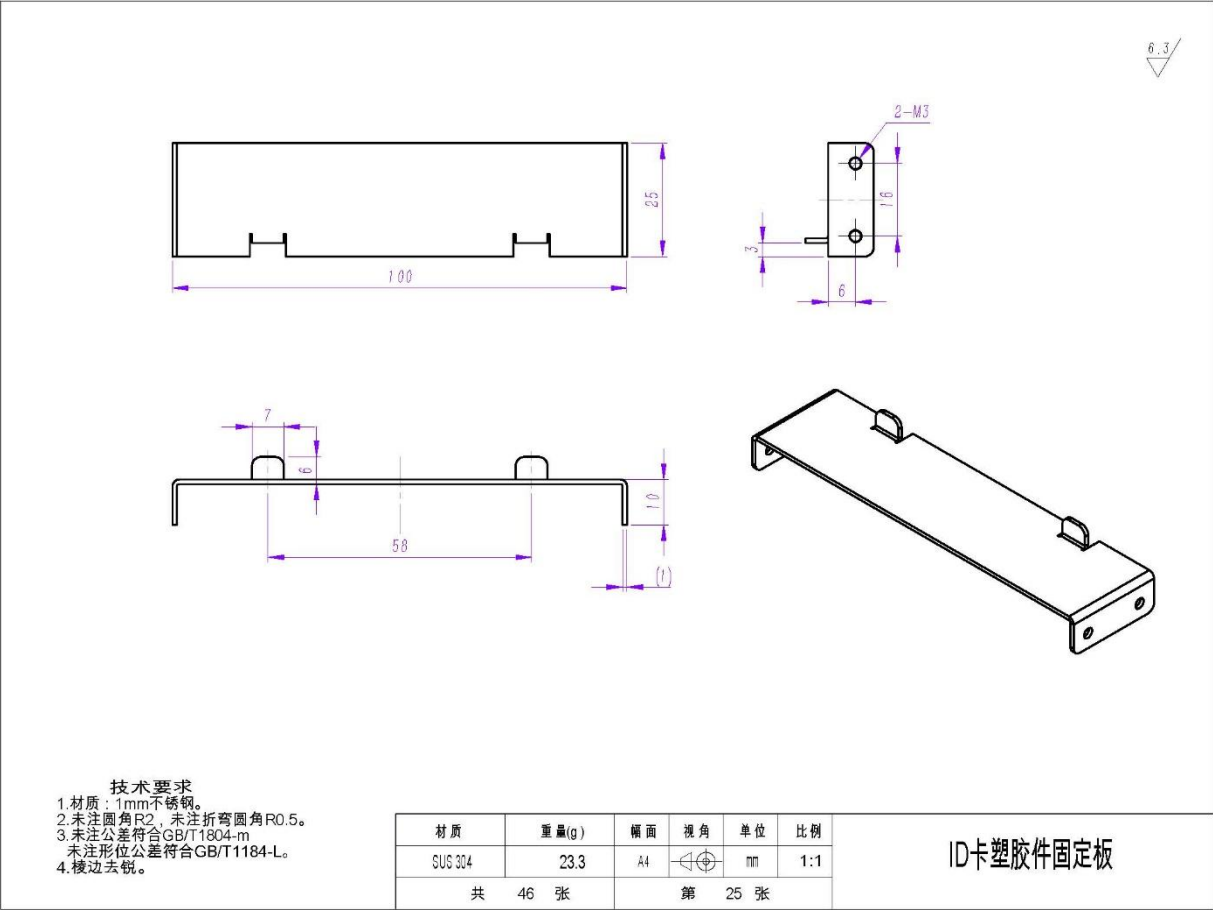
图A. 24规定了ID卡塑胶盖设计图纸。



图A. 24 ID 卡塑胶盖

A. 2. 25 ID卡塑胶件固定板

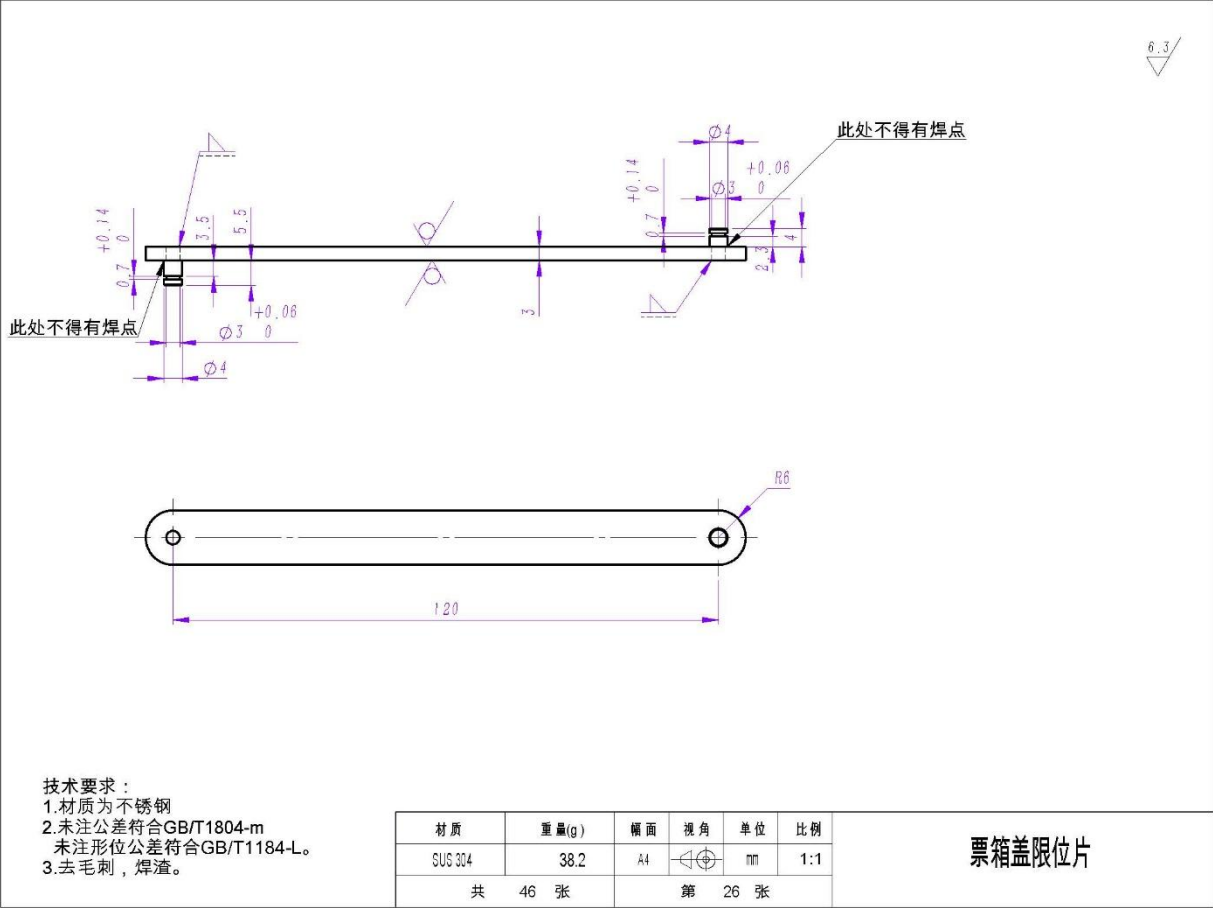
图A. 25规定了ID卡塑胶件固定板设计图纸。



图A. 25 ID 卡塑胶件固定板

A. 2. 26 票箱盖限位片

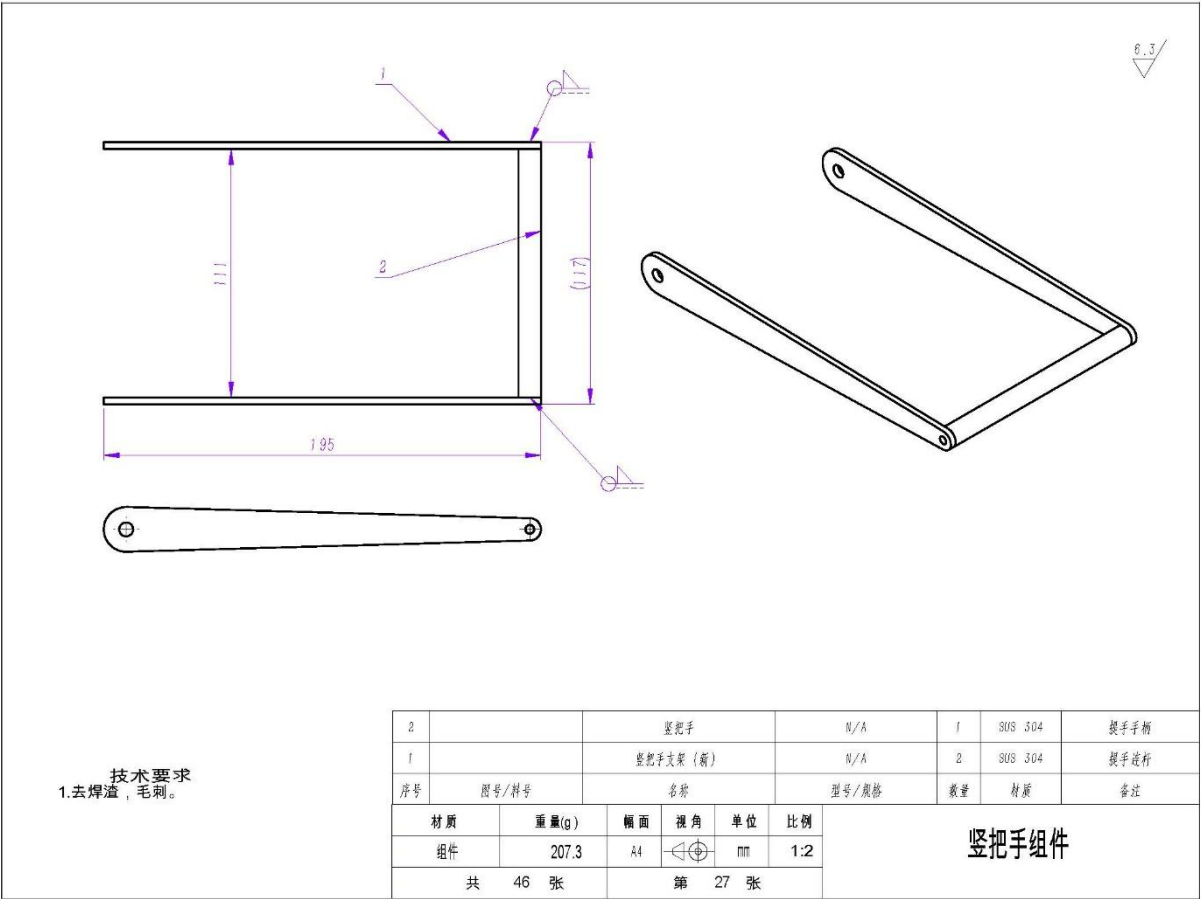
图A. 26规定了票箱盖限位片设计图纸。



图A. 26 票箱盖限位片

A. 2. 27 竖把手组件

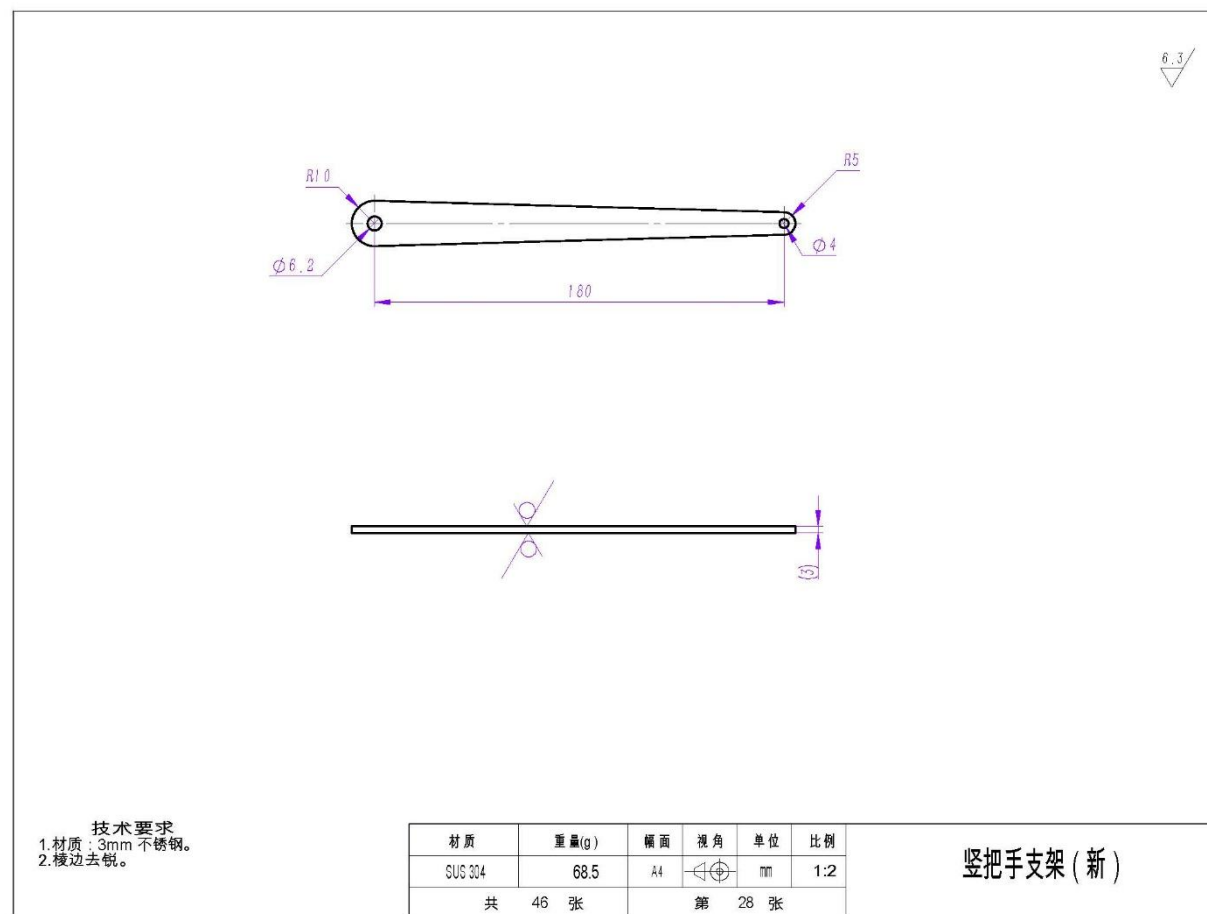
图A. 27规定了竖把手组件设计图纸。



图A. 27 竖把手组件

A. 2. 28 竖把手支架 (新)

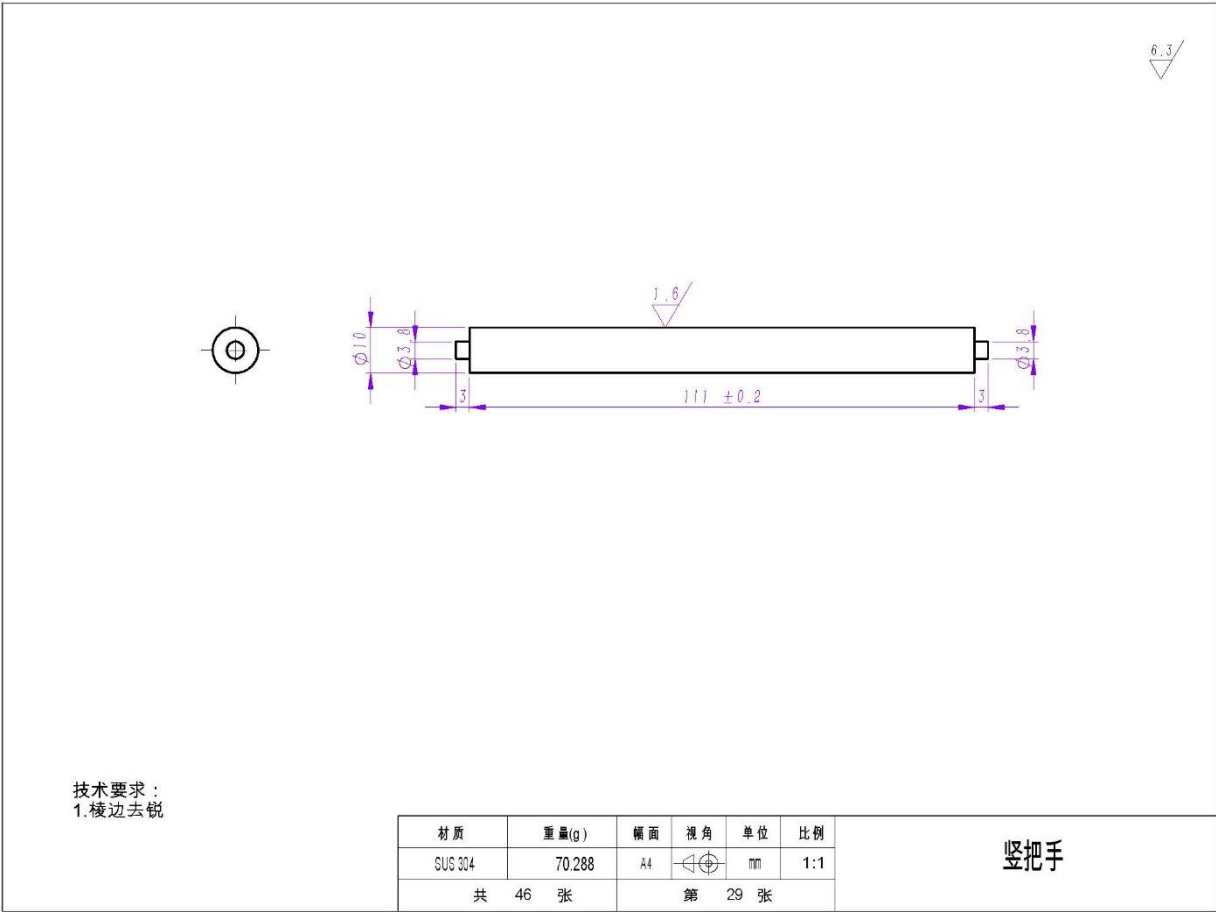
图A.28规定了竖把手支架（新）设计图纸。



图A. 28 竖把手支架（新）

A. 2. 29 竖把手

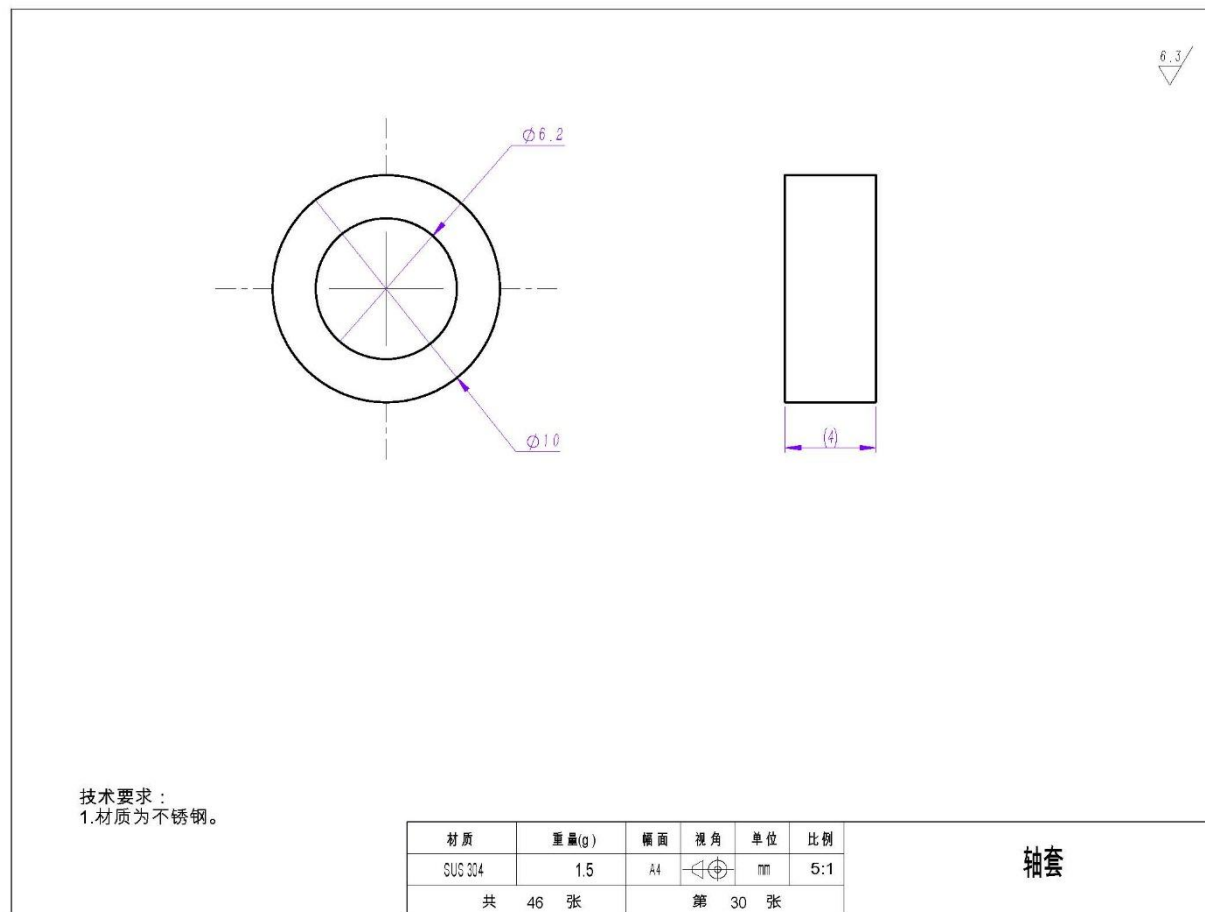
图A. 29规定了竖把手设计图纸。



图A. 29 竖把手

A. 2. 30 轴套

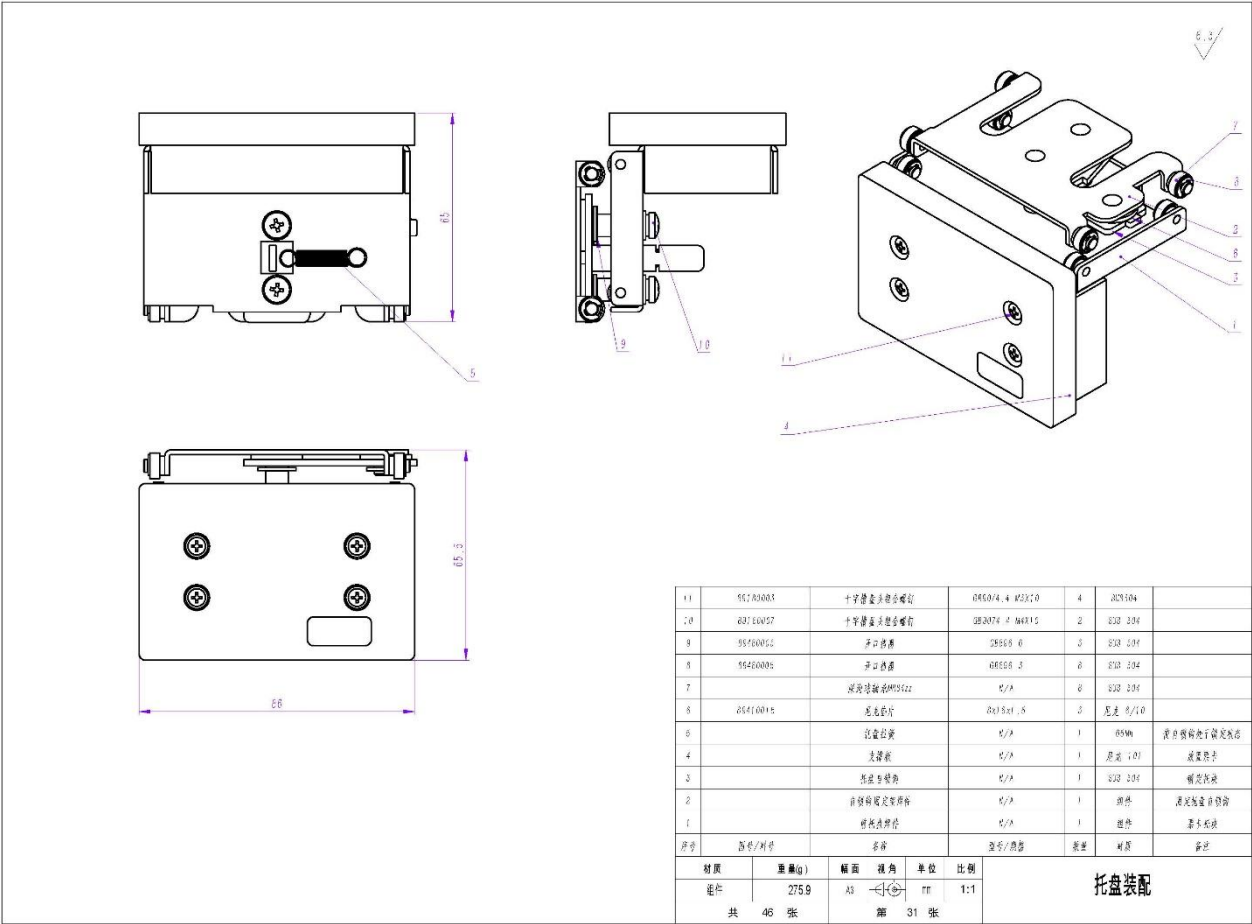
图A.30规定了轴套设计图纸。



图A. 30 轴套

A. 2. 31 托盘装配

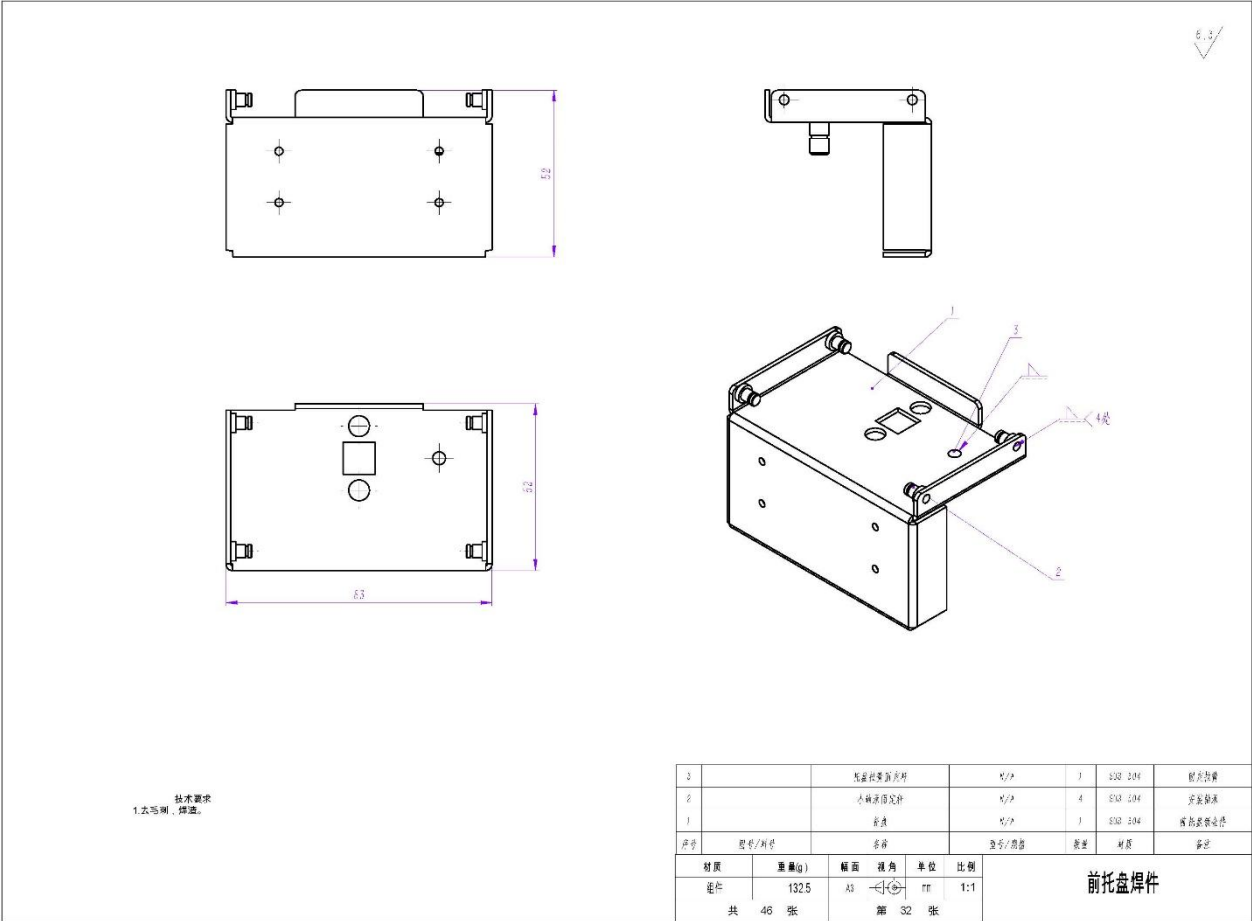
图A. 31规定了托盘装配设计图纸。



图A. 31 托盘装配

A. 2. 32 前托盘焊件

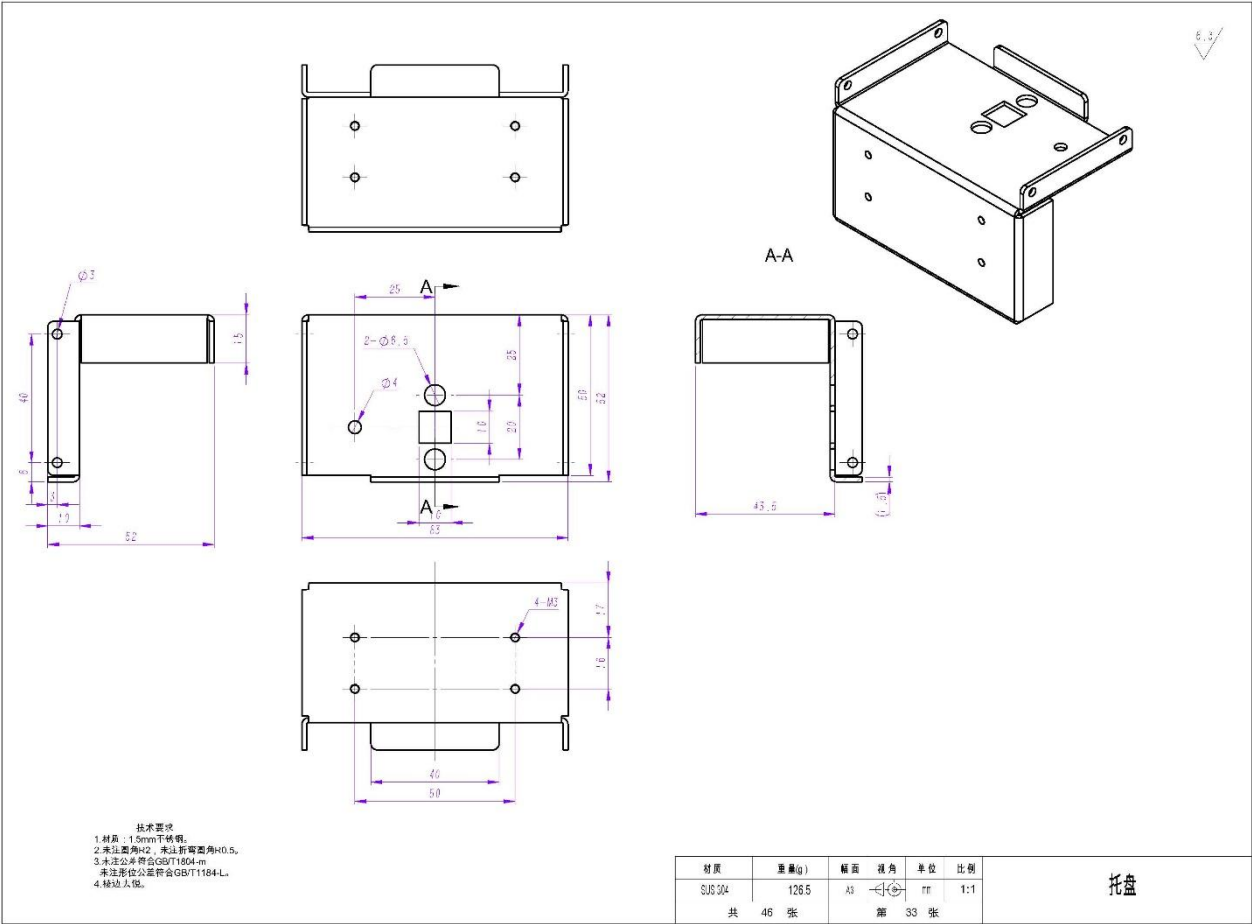
图A. 32规定了前托盘焊件设计图纸。



图A. 32 前托盘焊件

A. 2. 33 托盘

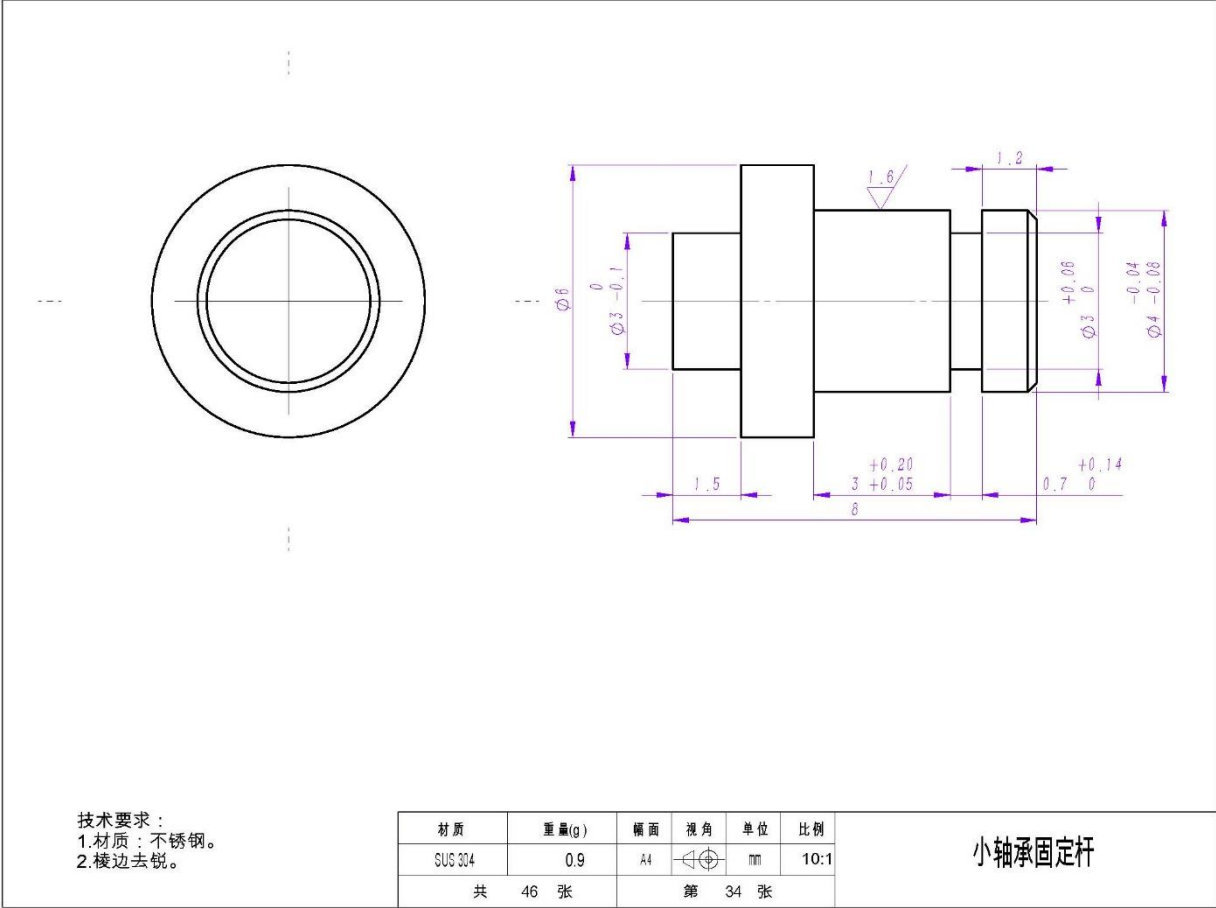
图A. 33规定了托盘设计图纸。



图A. 33 托盘

A. 2. 34 小轴承固定杆

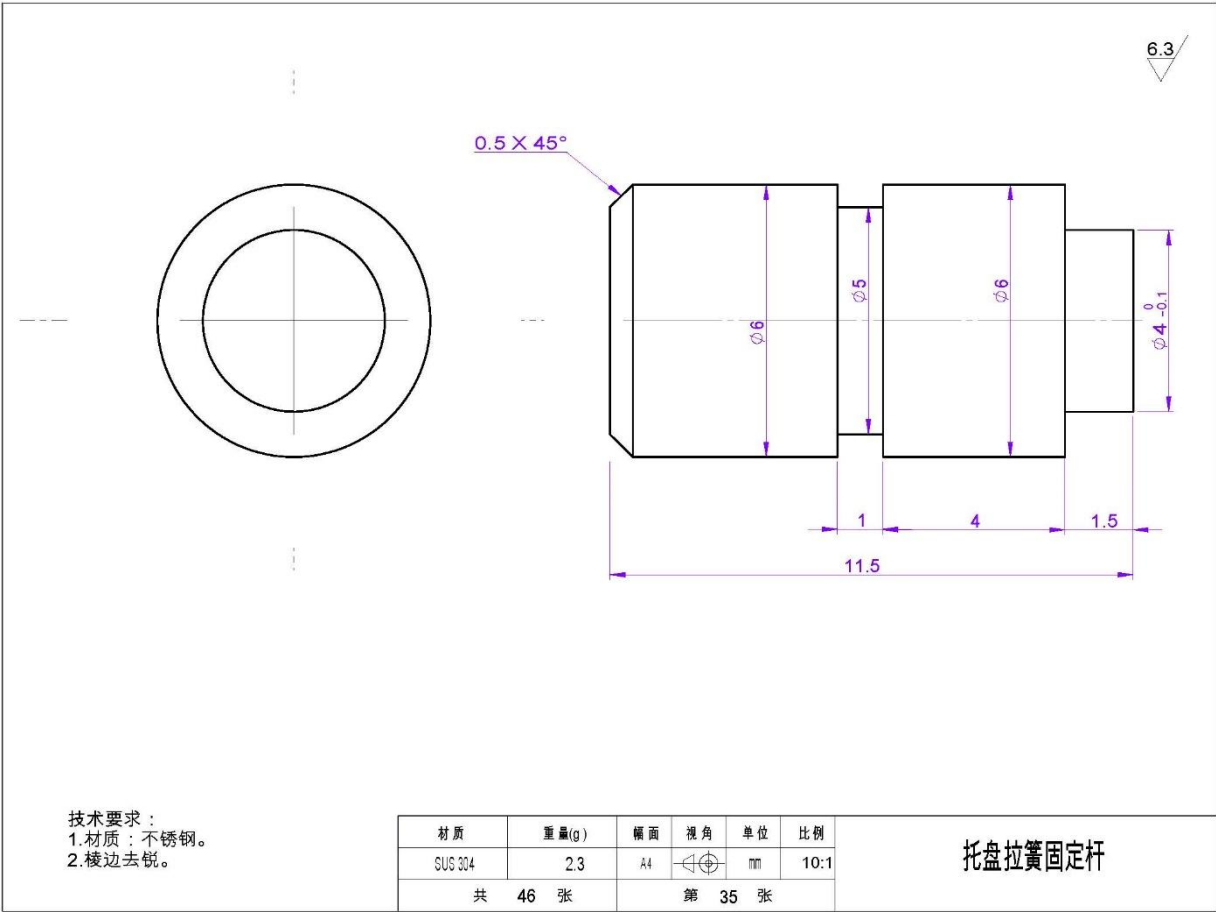
图A. 34规定了小轴承固定杆设计图纸。



图A. 34 小轴承固定杆

A. 2. 35 托盘拉簧固定杆

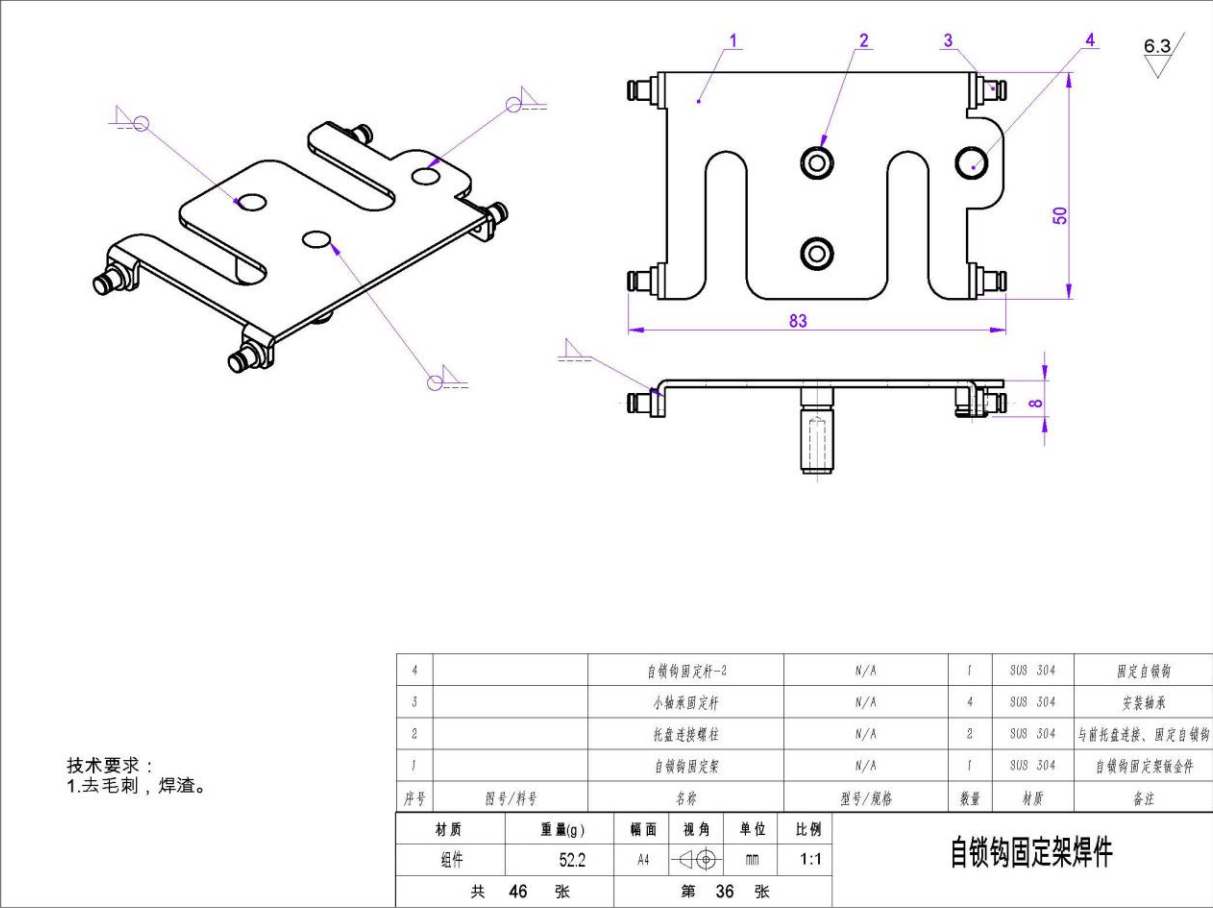
图A. 35规定了托盘拉簧固定杆设计图纸。



图A. 35 托盘拉簧固定杆

A. 2. 36 自锁钩固定架焊件

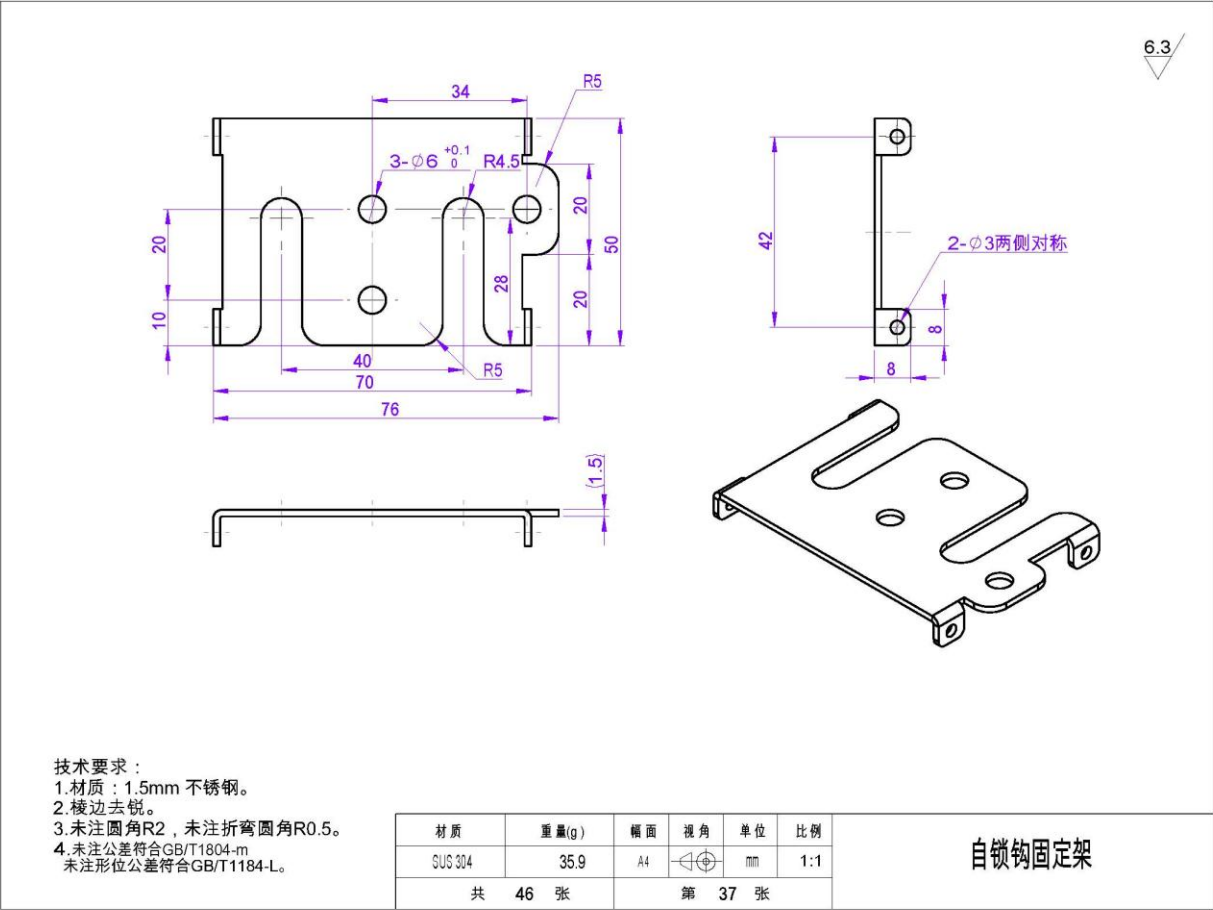
图A. 36规定了自锁钩固定架焊件设计图纸。



图A. 36 自锁钩固定架焊件

A. 2. 37 自锁钩固定架

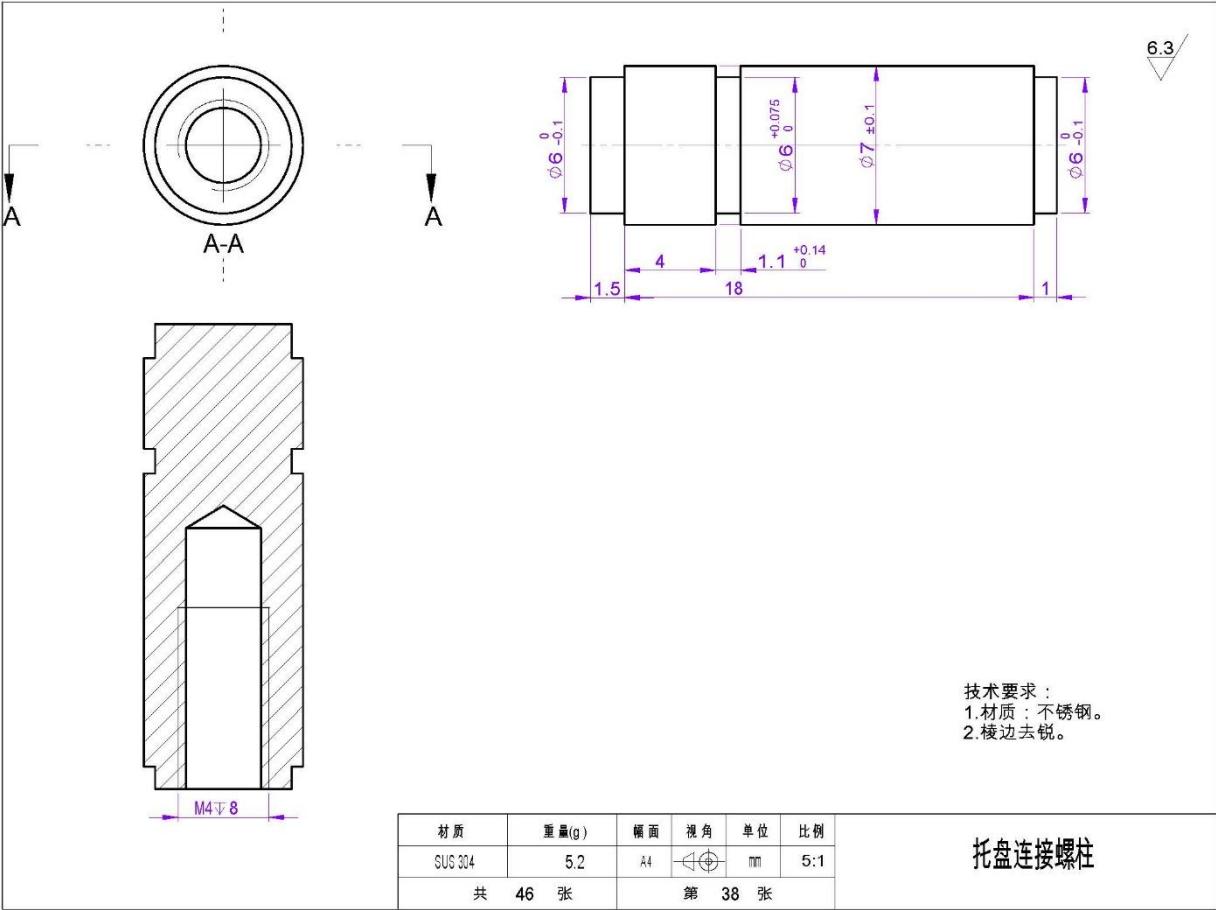
图A. 37规定了自锁钩固定架设计图纸。



图A. 37 自锁钩固定架

A. 2. 38 托盘连接螺柱

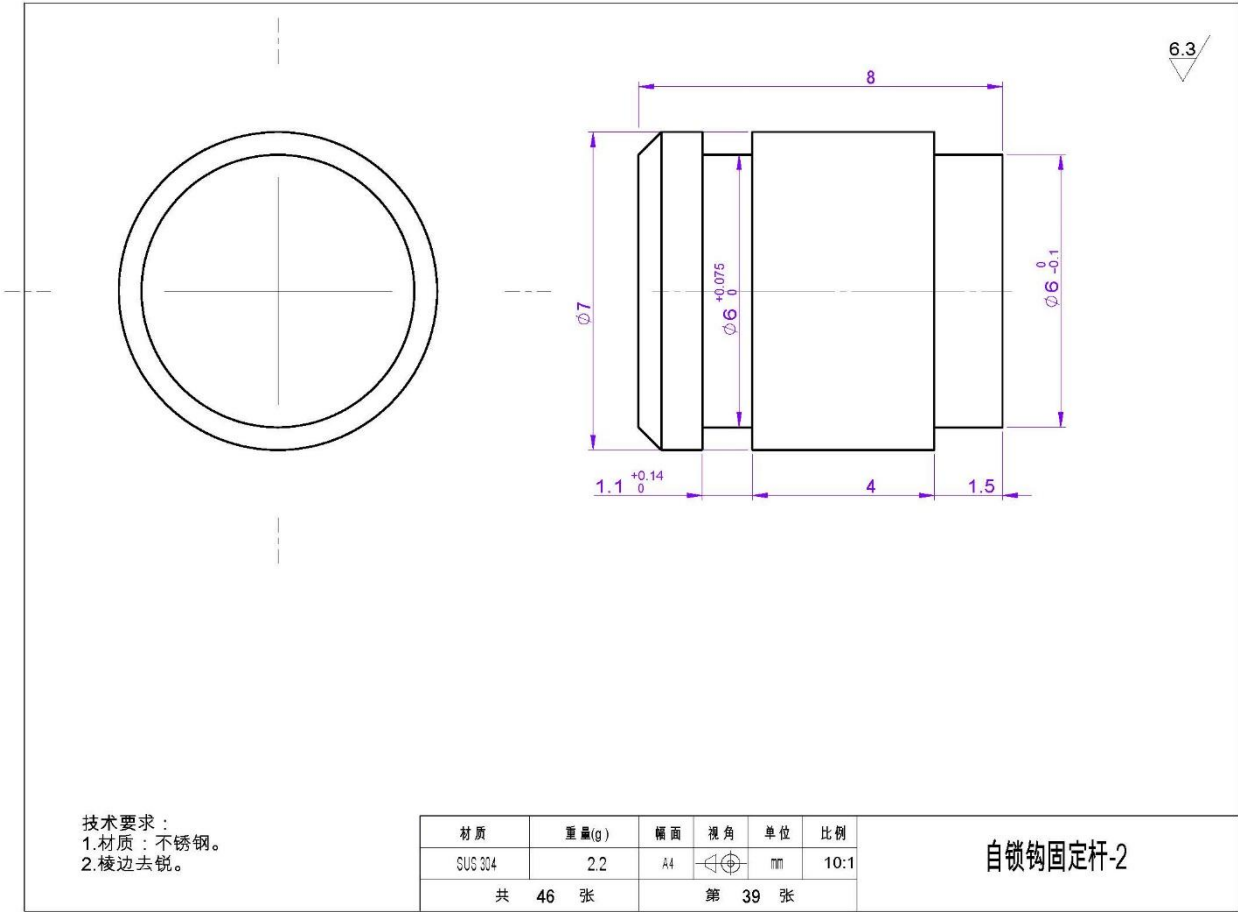
图A. 38规定了托盘连接螺柱设计图纸。



图A. 38 托盘连接螺柱

A. 2. 39 自锁钩固定杆2

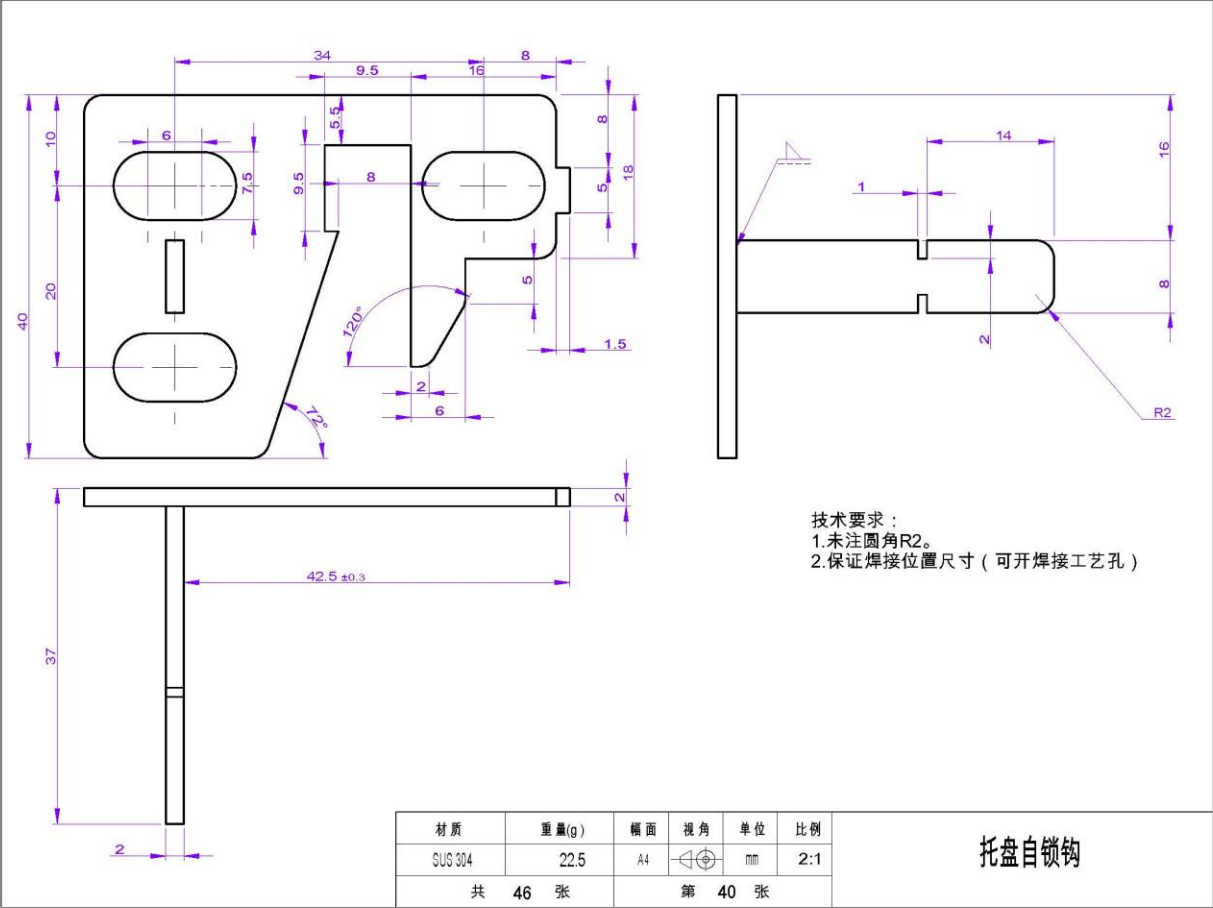
图A. 39规定了自锁钩固定杆2设计图纸。



图A. 39 自锁钩固定杆-2

A. 2. 40 托盘自锁钩

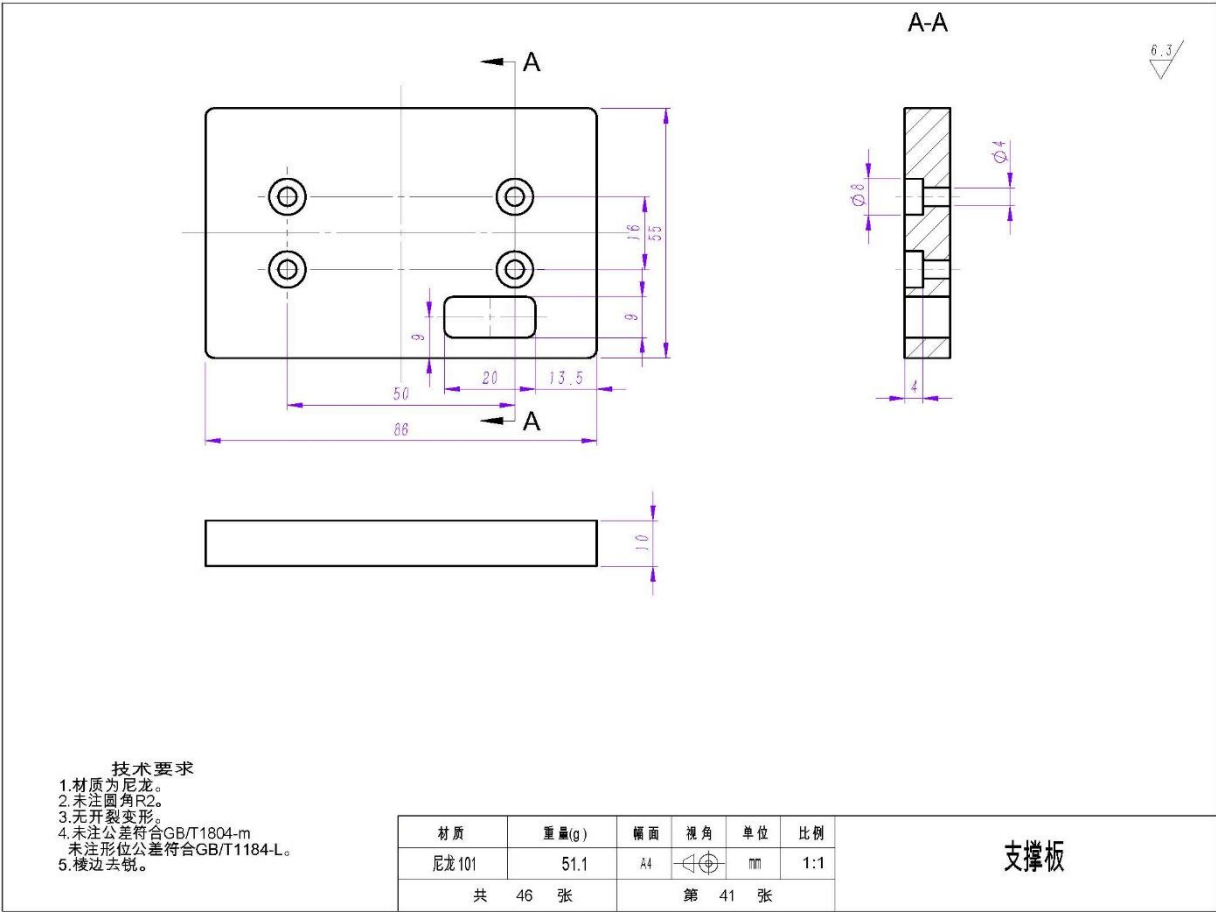
图A. 40规定了托盘自锁钩设计图纸。



图A. 40 托盘自锁钩

A. 2. 41 支撑板

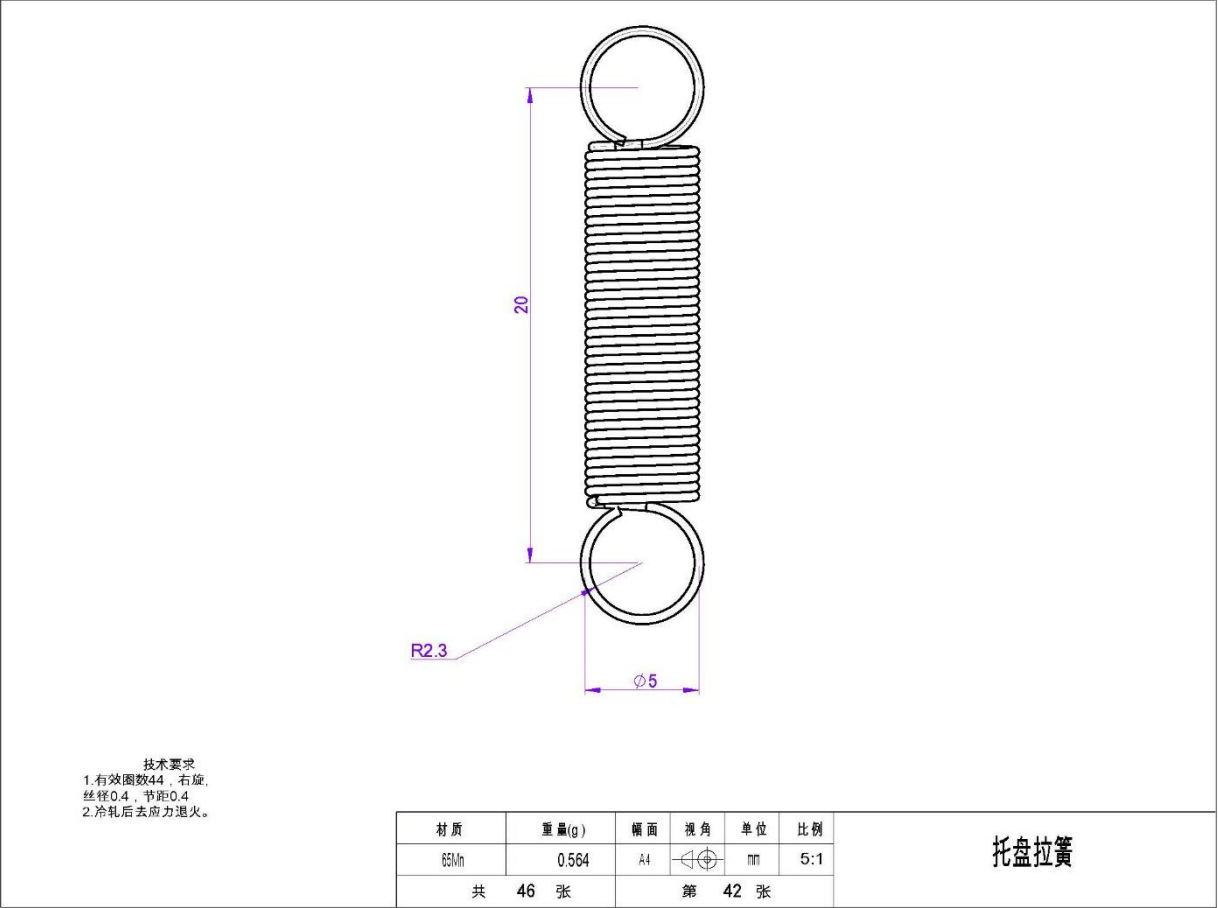
图A. 41规定了支撑板设计图纸。



图A. 41 支撑板

A. 2. 42 托盘拉簧

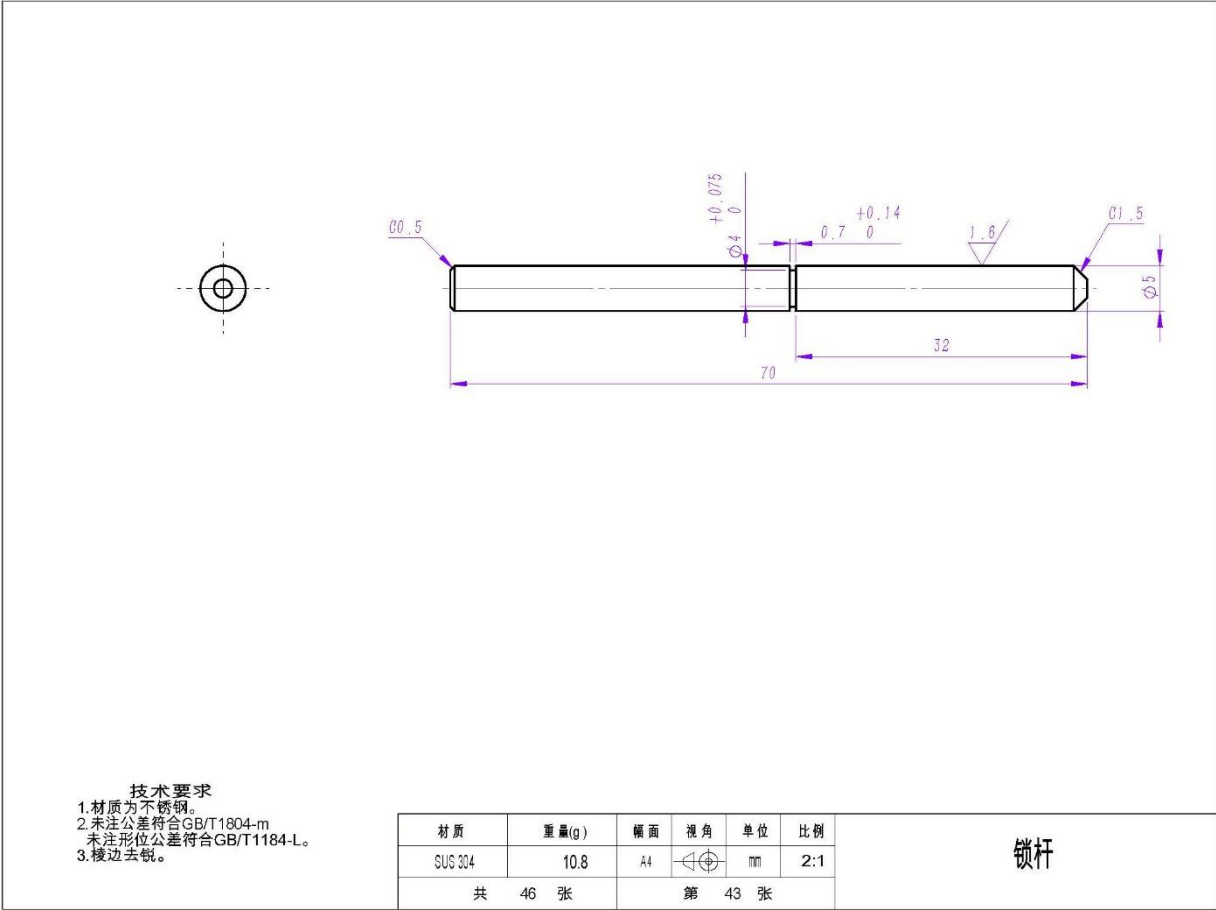
图A. 42规定了托盘拉簧设计图纸。



图A. 42 托盘拉簧

A. 2. 43 锁杆

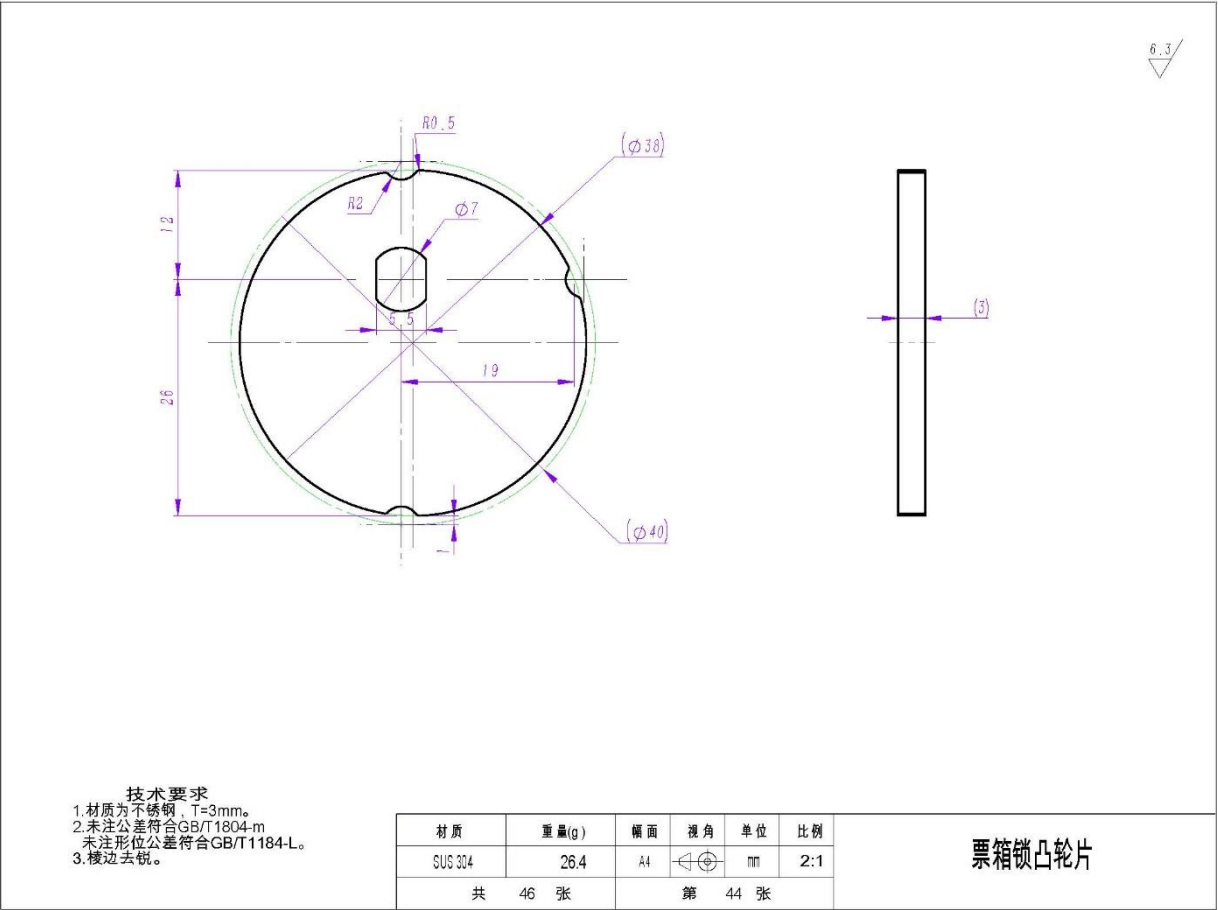
图A. 43规定了锁杆设计图纸。



图A. 43 锁杆

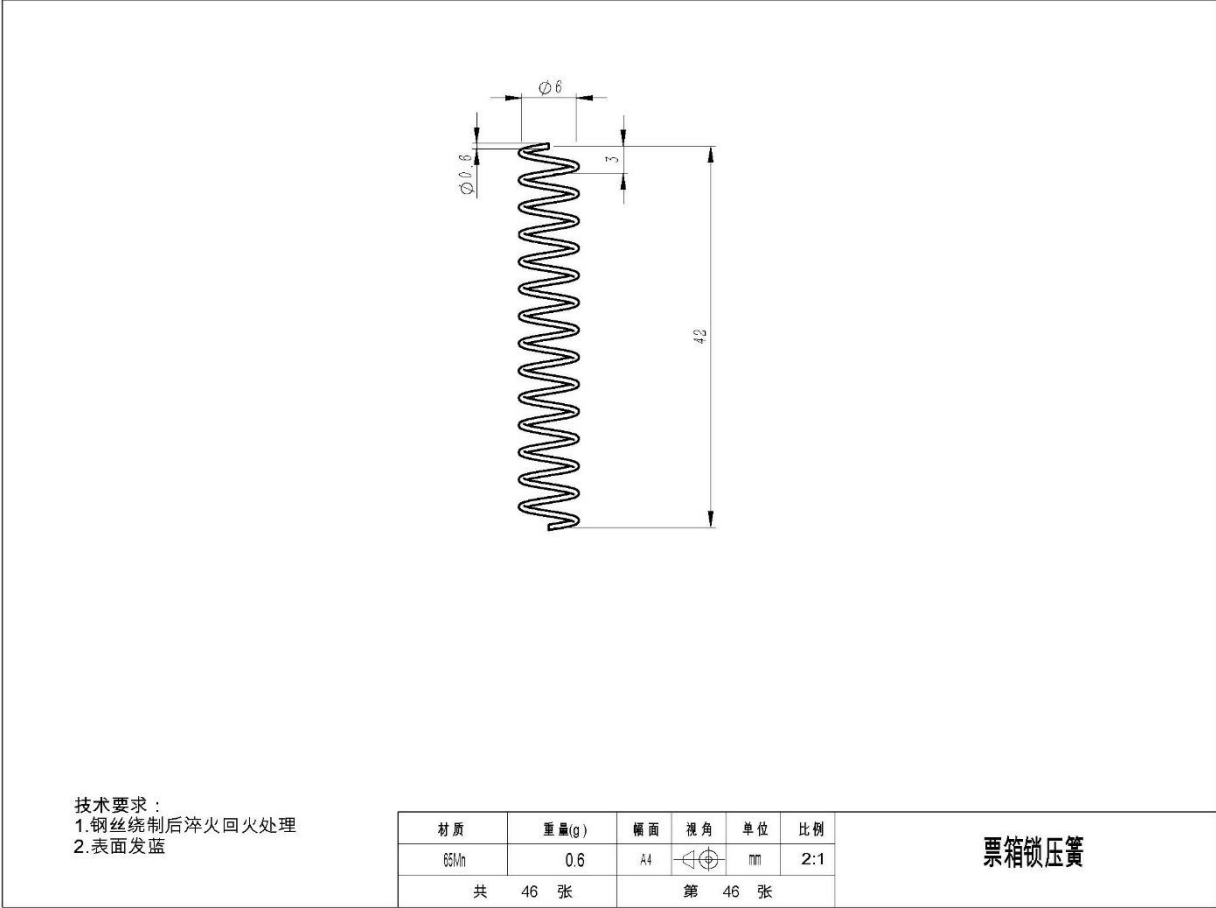
A. 2. 44 票箱锁凸轮片

图A. 44规定了票箱锁凸轮片设计图纸。



图A. 44 票箱凸轮片

图A. 46规定了票箱锁压簧设计图纸。



图A. 46 票箱锁压簧

A.3 锁要求

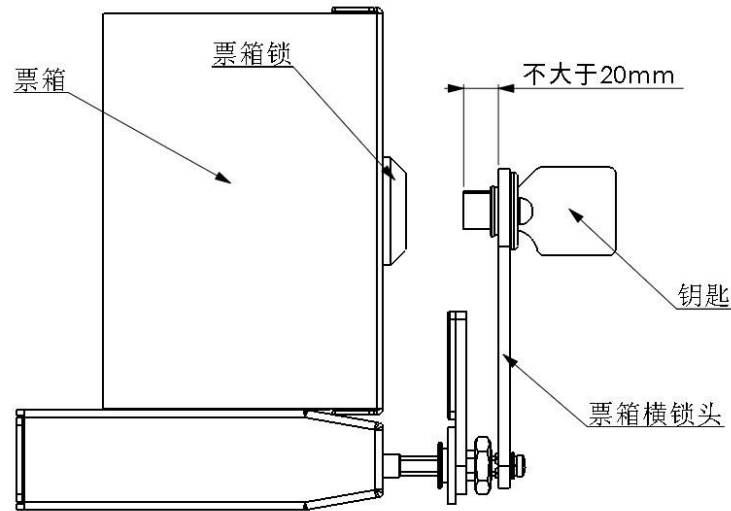
A.3.1 票箱锁品牌及型号

票箱锁在满足票箱功能的基础上，不做品牌和型号要求。

A.3.2 票箱锁尺寸要求

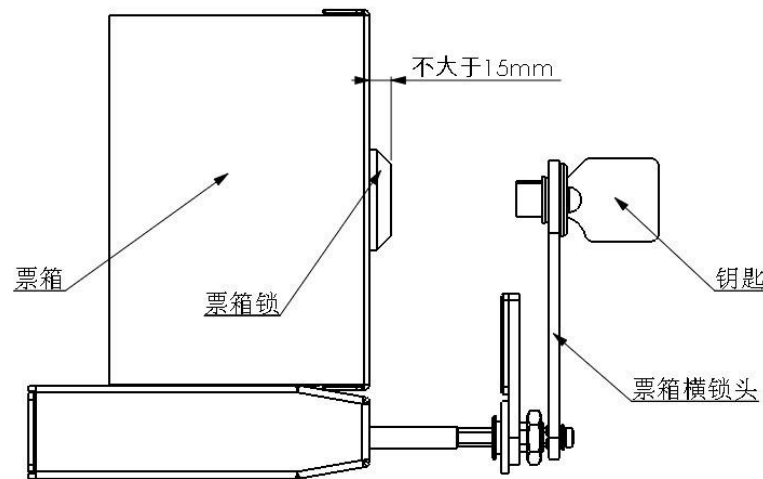
根据运营企业对于票箱整体调配的需求，配合车票发售模块和车票回收模块的互换功能，现对票箱锁提出要求如下：

- a) 车票发售/回收模块中固定票箱锁钥匙部件票箱一侧钥匙长度应不大于 20mm，如图 A.47 所示：



图A.47 票箱锁钥匙要求

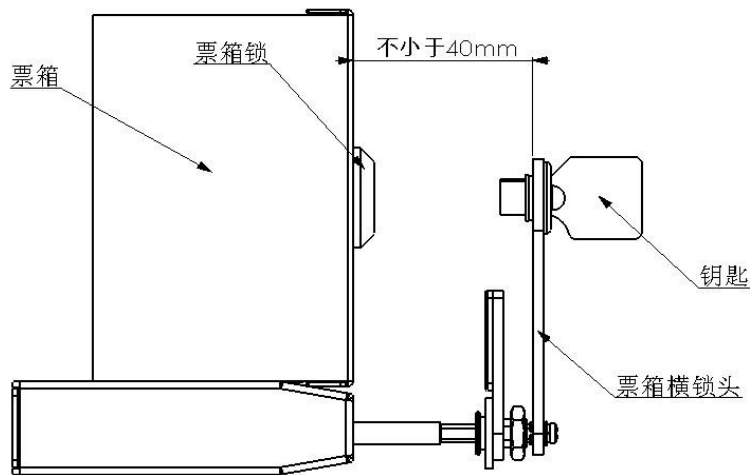
- b) 票箱锁安装完毕后凸出票箱的高度应不大于 15mm，如图 A.48 所示；



图A.48 票箱锁凸出高度要求

A.4 车票发售/回收模块要求

车票发售/回收模块中固定票箱锁钥匙部件，其拉出票箱平面的距离应不小于40mm，如图A.49所示；



图A. 49 部件拉出票箱平面距离

A. 5 票箱电子标签相关要求

A. 5. 1 芯片具体要求

根据运营企业对于票箱整体调配的需求，票箱内使用的电子标签芯片型号为：Mifare Ultralight C 和Mifare One（S50）一两种电子标签均为无电源，被动式射频全功能加密的可读写芯片，内含加密控制逻辑和通讯逻辑电路，并具有唯一序列号（32位），具有防冲突机制，支持多卡读取操作。

- a) 协议：ISO 14443 A；
- b) 频率：13.56 Mhz；
- c) 调制：ASK；
- d) 存储空间：
 - 1) Mifare Ultralight C 芯片：1536-Bits(192Byte) EEPROM，应具备 16 个扇区，每个扇区 4 个字节，具有独立的一组密码及访问控制；
 - 2) Mifare One（S50）芯片：1K 字节。
- e) 读写距离：0-100mm；
- f) 通信速率：106KBPS；
- g) 数据保存期为 10 年，可改写 10 万次，读无限次；
- h) 储存环境：温度：-25 ℃到 50 ℃；
- i) 湿度：20%到 90%RH；
- j) 工作温度：-40 ℃到 65 ℃尺寸要求：直径不大于 32mm 厚度不大于 3mm；
- k) 电子标签内存储的数据结构见表 337 和表 338 所示。

A.5.2 Mifare Ultralight C芯片存储的数据结构

Mifare Ultralight C数据结构见表A.2。

表 A.2 Mifare Ultralight C 数据结构示意表

电子标签数据结构定义				MIFARE Ultralight C 数据结构定义				
块	起始位置	字段	长度(BYTE)	页地址	0	1	2	3
静态区域	0	票箱编号	4	4				
块 A 48 byte	0	操作员 ID	3	5				
	3	设备 ID	4	5				
				6				
	7	票箱位置状态	1	6				
	8	操作后票箱状态	1	7				
	9	车票发行商 ID	4	7				
				8				
	13	车票物理类型	1	8				
	14	车票产品种类	2	8				
	16	预赋值属性	1	9				
	17	衍生产品 ID	1	9				
	18	车票数量	2	9				
	20	车站编码	2	10				
	22	安装位置	1	10				
	23	块操作标记	4	10				
				11				
	27	最后操作时间	7	11				
				12				
				13				
块 B 48 byte	34	校验字段	2	13				
	36	预留字段	12	14				
				15				
				16				
		结构同块 A	48	17—28	结构同上			

A.5.3 Mifare One (S50) 芯片存储的数据结构

Mifare One (S50) 数据结构见表A.3。

表 A.3 Mifare One (S50) 数据结构示意图表

电子标签数据结构定义				基于 M1 卡存储位置定义			
块	起始位置	字段	长度 (BYTE)	扇区号	相对块号	存储数据内容	长度 (BYTE)
				0	0	CSN	4
静态区域	0	票箱编号	4	2	0	同左侧电子标签定义	16
					1	无	16
					2	无	16
					3	KeyA(2) Access KeyB(2)	16
块 A 48 byte	0	操作员 ID	3	3	0	同左侧电子标签定义	16
	3	设备 ID	4			同左侧电子标签定义	
	7	票箱位置状态	1			同左侧电子标签定义	
	8	操作后票箱状态	1			同左侧电子标签定义	
	9	车票发行商 ID	4			同左侧电子标签定义	
	13	车票物理类型	1			同左侧电子标签定义	
	14	车票产品种类	2			同左侧电子标签定义	
	16	预赋值属性	1			同左侧电子标签定义	
	17	衍生产品 ID	1		1	同左侧电子标签定义	16
	18	车票数量	2			同左侧电子标签定义	
	20	车站编码	2			同左侧电子标签定义	
	22	安装位置	1			同左侧电子标签定义	
	23	块操作标记	4			同左侧电子标签定义	
	27	最后操作时间	7			同左侧电子标签定义	
	34	校验字段	2		2	同左侧电子标签定义	16
	36	预留字段	12			同左侧电子标签定义	
					3	KeyA(3) Access KeyB(3)	16
块 B 48 byte		结构同块 A	48	4	0、1、2	同块 A 数据定义	3*16
					3	KeyA(4) Access KeyB(4)	16

Mifare One (S50) 各扇区读写密钥见表A.4。

表 A.4 Mifare One (S50) 密钥设计方案表

电子标签数据结构	扇区	KeyA	Access (读写权限控制位)	KeyB
静态区	2	A	验证 KeyA 后具有读数据权限 验证 KeyB 后具有读写数据权限	B
块 A	3	A	验证 KeyA 后具有读写数据权限	B
块 B	4	A	验证 KeyA 后具有读写数据权限	B

其中，A、B代表实际定义的6字节密钥数值；静态区存储票箱编号，在车票处理单元中只需读该编号，但不会更改该编号，该编号应在票箱初始化或领用时更改，车票处理单元不使用KeyB密钥；扇区中KeyA密钥相同并使用明文；KeyA值为A0 A1 A2 A3 A4 A5，KeyB值为 CSN0 CSN1 CSN2 CSN3 CSN0 CSN1。

A.5.4 终端设备关键模块处理要求

终端设备关键模块在处理票箱电子标签时，兼容Mifare Ultralight C和Mifare One (S50) 芯片，根据不同的芯片物理类型进行读写。

附录 B (规范性附录) 底座设计要求

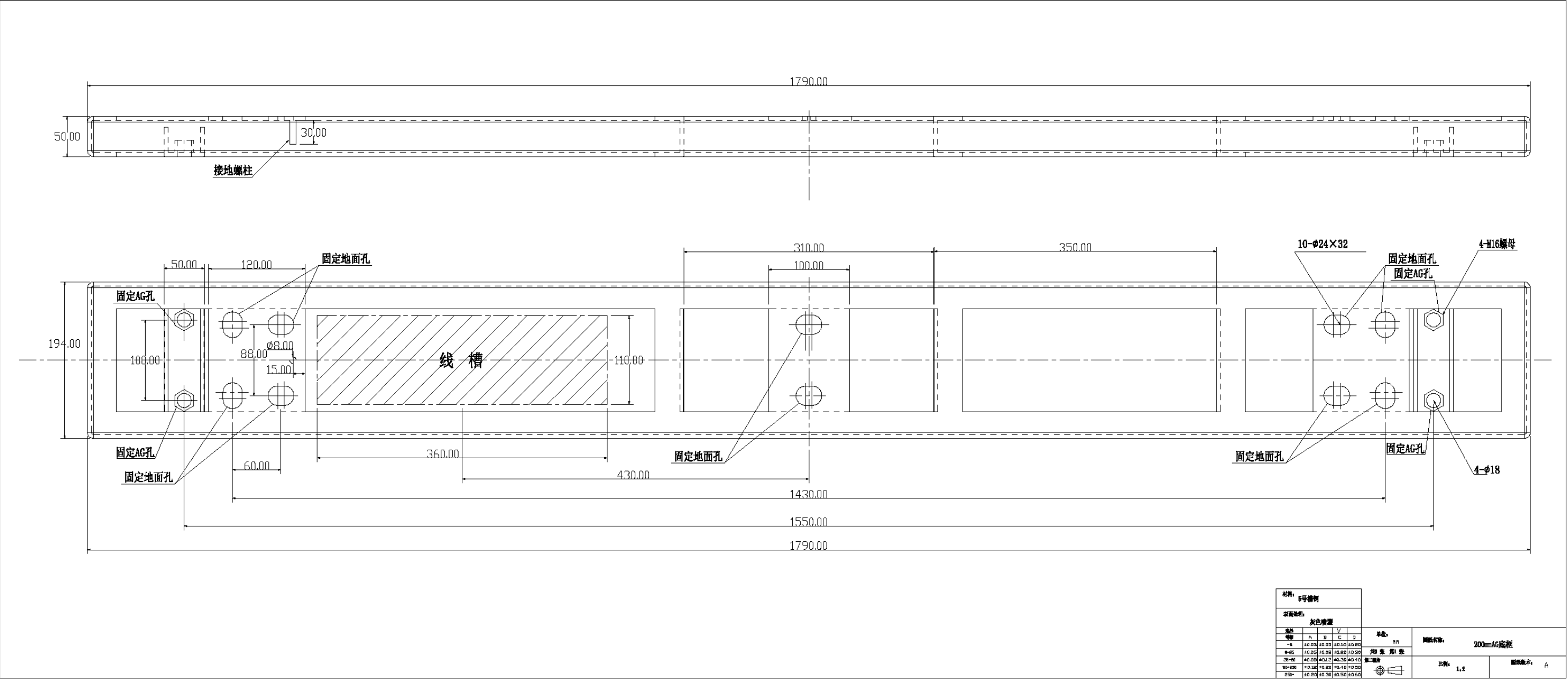
B.1 范围

本附录内容为轨道交通终端设备标准底座的设计图纸，图纸中包括常规双折门自动检票机、常规单折门自动检票机、宽通道自动检票机及自动售票机四类设备的标准安装底座图纸。以此附录作为设备安装底座的加工及检测依据。

B.2 详细图纸

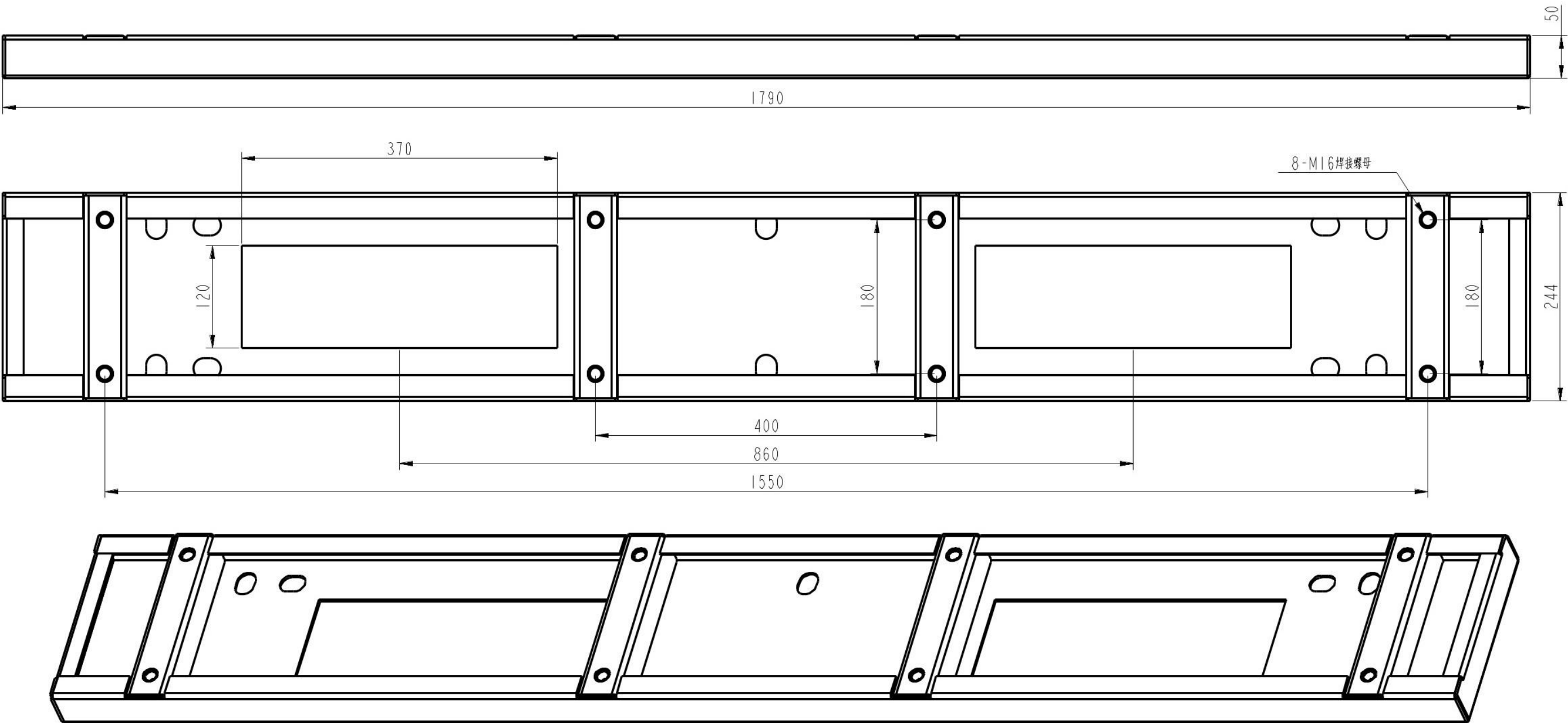
B.2.1 200mmAG底座

图B.1规定了200mmAG底座详细图纸。



B. 2. 2 250mmAG底座

图B. 2规定了250mmAG底座详细图纸。

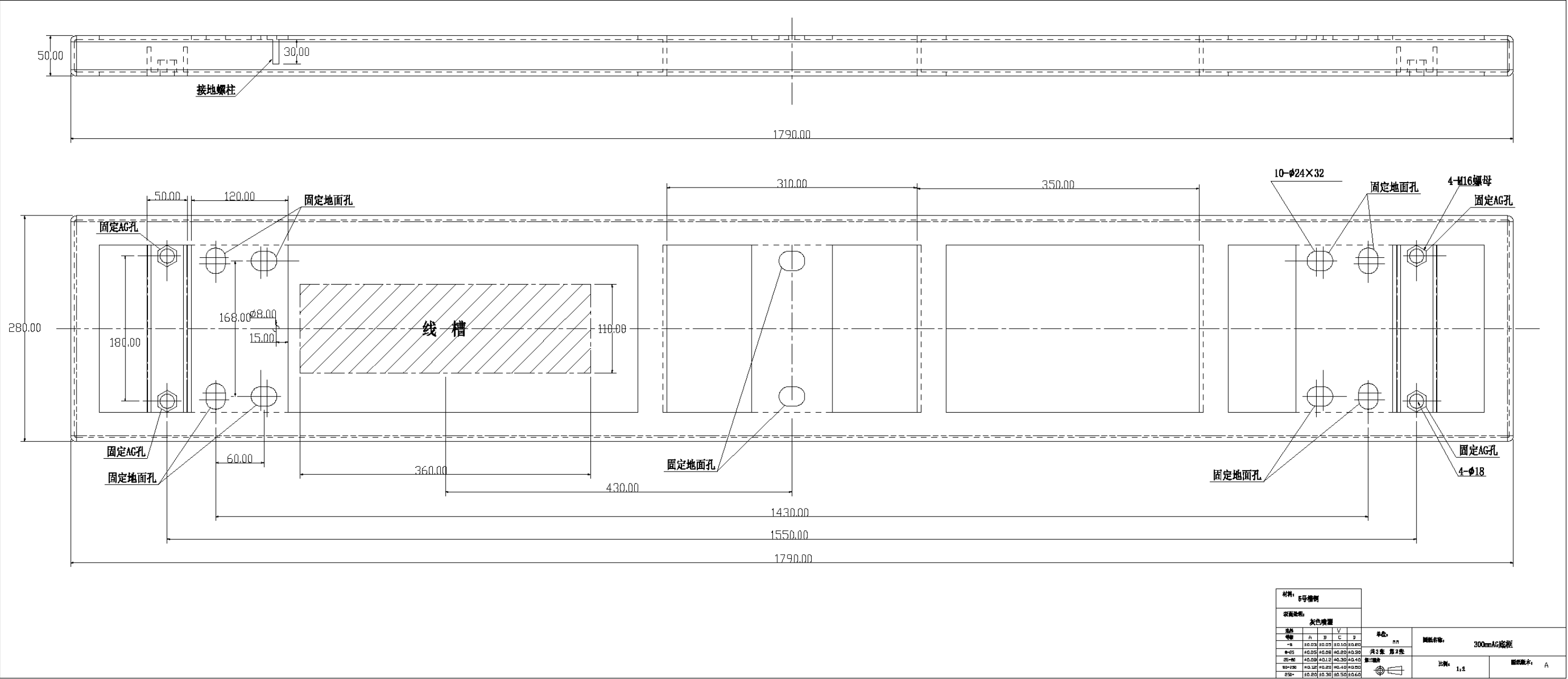


2	AGCS2004-MB3W3-A10-002	预埋底座固定板	4	Φ235B(冷) T=3.0
1	AGCS2004-MB3W3-A10-001	预埋底座板	1	Φ235B(冷) T=3.0

图 B. 2 250mmAG 底座

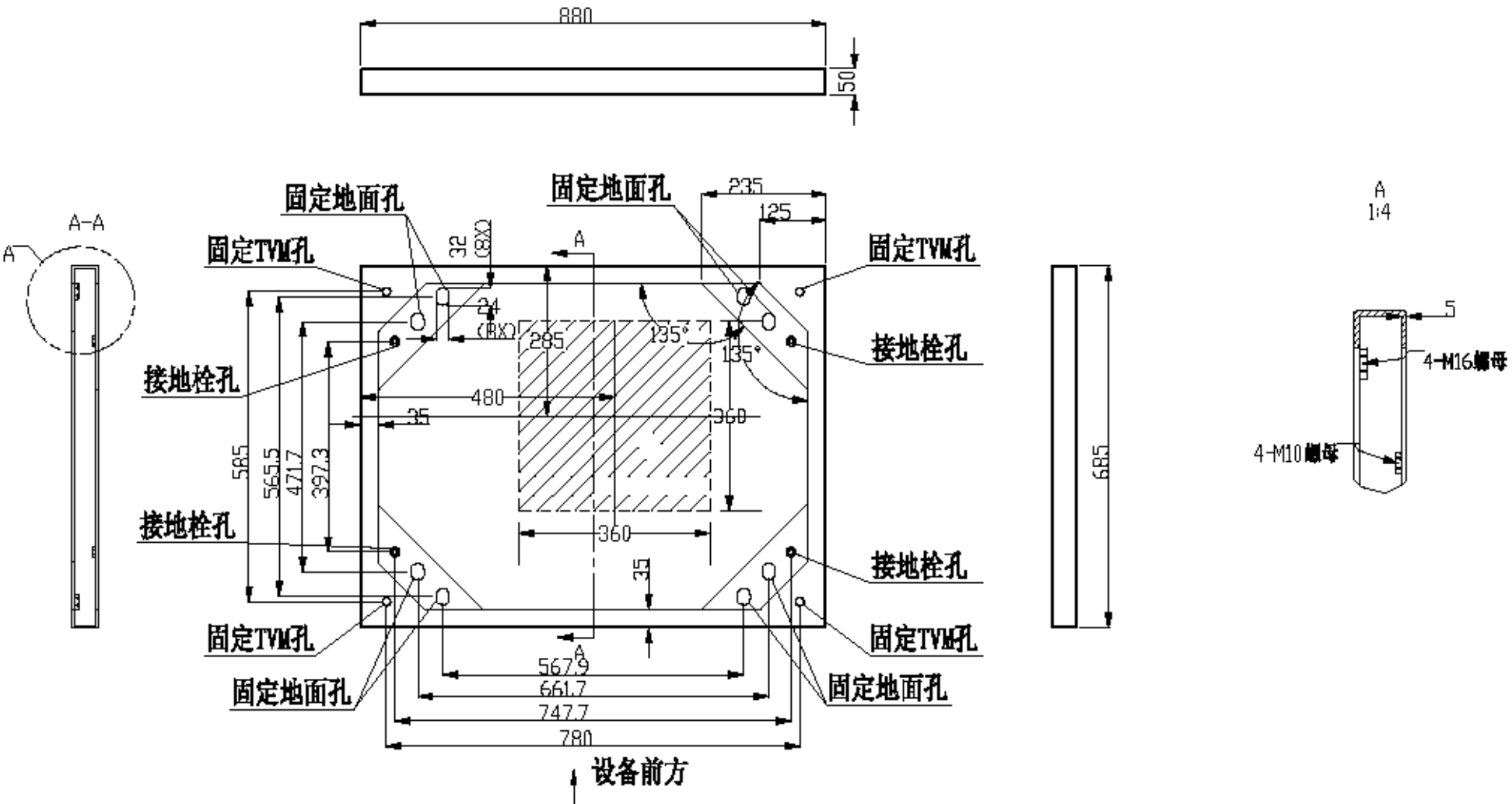
B. 2. 3 300mmAG底座

图B. 3规定了300mmAG底座详细图纸。



B. 2. 4 TVM底座

图B. 4规定了TVM底座详细图纸。



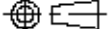
材料: 5号精制					单位: mm	图样名称: TVM底座	
表面处理: 灰色喷塑							
规格	A	B	C	D	共3张 第3张	比例: 1:2	图样版本: A
~8	±0.03	±0.05	±0.10	±0.20			
8~25	±0.05	±0.08	±0.20	±0.30	第三张角 		
25~60	±0.08	±0.12	±0.30	±0.40			
60~250	±0.12	±0.20	±0.40	±0.50			
250~	±0.20	±0.30	±0.50	±0.60			

图 B.4 TVM 底座

参考文献

[1]ASTM A 480/A 480Ma 标准规范的平轧不锈钢和一般要求耐热钢板，薄板和带材
[2]ISO 14443 非接触式IC卡标准协议
