

# 高校图书馆 UHF-RFID 技术

## 第二部分：应用指南

(第四版)

UHF-RFID Technique for University Libraries  
Part Two: Application Guide  
(4<sup>th</sup> Edition)



上海交通大学图书馆  
高校图书馆 RFID 技术应用联盟工作小组  
2013 年 2 月

# 目 录

目录 .....	1
前言 .....	1
1 说明 .....	2
2 规范性引用文件 .....	2
3 术语和定义 .....	2
4 遵循原则 .....	3
5 存储方式 .....	3
6 版本的更新机制 .....	5
6.1 管理方式 .....	5
6.2 兼容性与升级方式 .....	5
7 EPC 数据区存储方式说明 .....	5
7.1 存储方式 .....	6
7.1.1 编码方式 1: 标签的 EPC 容量为 96bit .....	6
7.1.2 编码方式 2: 标签的 EPC 容量为 128bit .....	6
7.1.3 编码方式 3: 标签的 EPC 容量为 144bit 或以上 .....	6
7.1.4 编码方式 4: 保留 .....	7
7.1.5 推荐 EPC 存取操作方式 .....	7
7.2 检查与更新机制 .....	7
8 USER 数据区存储方式说明 .....	8
8.1 存储方式 .....	8
8.2 工作流程与更新机制 .....	9
8.2.1 借出操作流程 .....	9
8.2.2 归还操作流程 .....	10
9 数据元素取值与编码说明 .....	10
9.1 安全位、预留位、分拣信息 .....	10
9.2 编码方式及版本 .....	11
9.3 内容索引 .....	11
9.4 馆藏标识符 .....	11
9.5 所属馆标识 .....	12
9.6 卷册信息 .....	12
9.7 馆际互借借入馆 .....	13
9.8 馆际互借事务号 .....	13
9.9 馆藏类别与状态 .....	13
9.10 分馆标识 .....	13
9.11 馆藏位置 .....	14
9.12 临时馆藏位置 .....	14
9.13 主题 .....	14
9.14 国际标准书/刊号 .....	14
10 操作规范 .....	14
10.1 TID 区域读写 .....	14
10.2 锁定与口令 .....	15
11 基于 ISO28560 推荐的 ISO/IEC15962 编码与压缩规则 .....	15

附录 A: 编码与解码算法 .....	16
A1 适用情况 .....	16
A1.1 针对 EPC 为 96bit 长度 .....	16
A1.2 针对 EPC 为 128bit 长度 .....	16
A2 数据结构 .....	16
A2.1 EPC 96bit 数据结构 .....	16
A2.2 EPC 128bit 数据结构 .....	17
A3 编码程序源代码(C 语言).....	17
A3.1 EPC 96bit 编码程序源代码 (供 EPC 中编码方式为 00[方式 1]时调用) .....	17
A3.2 EPC 128bit 编码程序源代码 (供 EPC 中编码方式为 01[方式 2]时调用) .....	18
A4 解码程序源代码(C 语言).....	20
A4.1 EPC 96bit 解码程序源代码 (供 EPC 中编码方式为 00[方式 1]时调用) .....	20
A4.2 EPC 128bit 解码程序源代码 (供 EPC 中编码方式为 01[方式 2]时调用) .....	21
A5 测试程序源代码 .....	23
A5.1 EPC 96bit 测试程序源代码 .....	23
A5.2 EPC 128bit 测试程序源代码 .....	24
参考文献 .....	26

## 前言

近年来, RFID (Radio Frequency Identification, 无线射频识别) 技术获得了广泛的应用, 并取得极大成功。其中, UHF-RFID (Ultra High Frequency-RFID, 超高频 RFID) 技术在高校图书馆的应用发展十分迅速。

为了确保高校图书馆在引入 UHF-RFID 技术、开展图书馆服务应用时, 建立一套统一的标准和应用规范, 以便保障设备和系统在实际应用时的兼容性、互换性、以及可扩展性, 针对高校图书馆制订一个应用规范(含: 数据模型规范, 应用指南)具有重要的意义。为此, 受“高校图书馆 RFID 技术应用联盟<sup>[1]</sup>”(以下简称: 联盟)的委托, 在联盟工作小组第二次(2010 年 8 月)和第三次(2011 年 3 月)工作会议的基本精神的指导下, 在充分调研和听取各方面意见和建议的基础上, 上海交通大学图书馆代表联盟起草了《高校图书馆 UHF-RFID 数据模型规范》(以下简称: 规范)。

本规范的出台, 主要依据了如下国际和国家标准:

- ISO28560-1: 2011 信息和文献-图书馆无线射频识别(RFID)-实施的数据元素和总原则
- ISO28560-2: 2011 信息和文献-图书馆无线射频识别(RFID)-基于 ISO/IEC15962 规则的无线射频识别(RFID)数据元素的编码
- ISO/IEC15962: 2004 信息技术-项目管理用射频识别(RFID)-数据协议: 数据编码规则和逻辑存储功能
- WH/T 43-2012/ WH/T 44-2012: 图书馆-射频识别-数据模型(中华人民共和国文化行业标准)
- DB44/T 898.1/ DB44/T 898.2-2011: 射频识别-图书管理(广东省地方标准)

并严格遵循了上述标准的总体框架和原则。

本规范以联盟为主体提出, 其目的是, 希望为高校图书馆 UHF-RFID 技术的应用, 提供一套符合与兼容国际化标准的 UHF-RFID 数据模型规范, 以及实施应用指南, 从而满足高校图书馆使用 UHF-RFID 技术来管理图书馆的各项需求。其它图书馆、包括公共图书馆等, 亦可将此规范作为参照或扩展基点。

考虑到当前阶段 UHF-RFID 标签的用户数据区容量(参见附录 A)较小的因素, 用户数据元素需要有一定的限定。在保证满足高校图书馆应用适用性、可行性、兼容性以及可扩展性等应用需求的条件下, 本规范依据国际和国家标准作为基础, 并通过合理取舍、筛选出必备的用户数据元素。所选的用户数据元素原则上属于国际和国家标准的子集, 个别用户数据元素根据图书馆实际应用需求增加。

本规范适用于所有使用符合 ISO/IEC18000-6C (EPC Class1 Gen2) 通信协议标准的 UHF-RFID 标签与设备的(中国)高校图书馆以及公共图书馆等其它相关机构。

由于 ISO28560-4 和 ISO28560-5 国际标准(主要适用于 UHF-RFID 应用)仍在制定中, 因此, 本规范今后将会考虑与最新国际标准的接轨与兼容, 并发布相应的修订版。

## 1 说明

本部分是对《第一部分: 数据模型规范》中的数据元素的具体取值、编解码等细则的解释与说明。

## 2 规范性引用文件

本规范的制订, 引用(借鉴和参考)了若干重要标准或规范文件。特别强调, 仅所注日期的版本适用本规范, 但鼓励根据本规范达成标准的各方研究是否可使用相关修正或修订版文件; 凡是未注明日期的引用文件, 其最新版本(包括所有的修订版)适用于本规范。主要引用文件包括:

- ☑ JY/T1001-2012<sup>[2]</sup> 教育管理信息-教育管理基础代码(中华人民共和国教育部行业标准)
- ☑ WH/T43-2012<sup>[3]</sup> 图书馆-射频识别-数据模型-第 1 部分: 数据元素设置及应用规则(中华人民共和国文化行业标准)
- ☑ WH/T44-2012<sup>[4]</sup> 图书馆-射频识别-数据模型-第 2 部分: 基于 ISO/IEC15962 的数据元素编码方案(中华人民共和国文化行业标准)
- ☑ ISO/IEC646 IRV<sup>[5]</sup> 信息技术-ISO 信息交换七位编码字符集(Information technology -- ISO 7-bit coded character set for information interchange)
- ☑ ISO/IEC15961: 2004<sup>[6]</sup> 信息技术-项目管理用射频识别(RFID)-数据协议: 应用接口(Information technology -- Radio frequency identification (RFID) for item management -- Data protocol: application interface)
- ☑ ISO/IEC15962: 2004<sup>[7]</sup> 信息技术-项目管理用射频识别(RFID)-数据协议: 数据编码规则和逻辑存储功能(Information technology -- Radio frequency identification (RFID) for item management -- Data protocol: data encoding rules and logical memory functions)
- ☑ ISO/IEC18000-6: 2010<sup>[8]</sup> 信息技术-项目管理用射频识别-860 MHz 至 960 MHz 空中接口通信参数(Information technology -- Radio frequency identification for item management -- Part 6: Parameters for air interface communications at 860 MHz to 960 MHz)
- ☑ ISO28560-1: 2011<sup>[9]</sup> 信息和文献-图书馆无线射频识别(RFID)-实施的数据元素和总原则(Information and documentation -- RFID in libraries -- Part 1: Data elements and general guidelines for implementation)
- ☑ ISO28560-2: 2011<sup>[10]</sup> 信息和文献-图书馆无线射频识别(RFID)-基于 ISO/IEC15962 规则的无线射频识别(RFID)数据元素的编码(Information and documentation -- RFID in libraries -- Part 2: Encoding of RFID data elements based on rules from ISO/IEC15962)

## 3 术语和定义

下列术语和定义适用于本规范。

- (1) **UHF-RFID (超高频 RFID):** 工作频率为 860MHz 至 960MHz 之间的无线射频识别技术。
- (2) **馆际互借 (InterLibrary Loan, ILL):** 图书馆之间馆藏资源相互借阅的服务。
- (3) **馆藏 (Item):** 在图书馆系统中可被跟踪的单件物品, 本规范中指实物馆藏。
- (4) **数据元素 (Data Item):** 也称为数据项, 用来描述实体的某种属性, 是数据结构中的最小单位。数据元素可以是字母、数字或两者的组合。数据元素通过数据类型(逻辑型、数值型、字符型

等)及数据长度来描述。

- (5) **数据结构 (Data Structure)**: 是指相互之间存在一种或多种特定关系的数据元素的集合, 代表存储和组织数据的方式。
- (6) **图书馆管理系统 (Library Management System)**: 针对图书馆业务范围开发的计算机系统, 一般具备跟踪所拥有的馆藏、馆藏外借、订购、支付、读者管理、...等功能。
- (7) **分卷(册) (Part)**: 组成一套文献(包含光盘等资料)的独立物理单元。
- (8) **整套 (Set)**: 包含了多个卷的文献(包含光盘等资料), 各卷均通过同一个文献标识符来标识, 并且可作为一个整体来借还。
- (9) **空中接口协议 (Air Interface Protocol)**: RFID 读写器和相应类型 RFID 标签之间的通讯规则, 包括: 频率、调制、位编码及命令集。
- (10) **应用族标识符 (Application Family Identifier, AFI)**: RFID 应用系统中用来区分标签使用范围、方式及单品标识类别的一种管理机制。注: 实际使用时, RFID 系统可利用存储于标签中的 AFI, 从射频覆盖区中的诸多标签中选择出符合特定应用需求的射频标签, 以便实现标签数据的快速存取。
- (11) **数据格式 (Data Format)**: 数据协议中所使用的机制, 借以标识对象标识符在射频标签内的编码方式, 并且(若可能)标识出对象标识符集合的特定数据字典。
- (12) **唯一标签标识符 (TID)**: 每个标签都具有一个唯一的标签标识符, 由集成电路制造商写入。是 ISO 18000-6C 标准下定义的标签唯一标识符。
- (13) **弧 (Arc)**: 对象标识符树的特定分支, 在需要定义特定字段时增加新的分支。
- (14) **对象标识符 (Object Identifier)**: 与对象相关联的, 与其它同类型值均不相同的全局唯一值。
- (15) **相对对象标识符 (Relative-OID)**: 在根对象标识符之后用于标识余下弧(分支)相对位置与信息的特定对象标识符。
- (16) **根对象标识符 (Root-OID)**: 构成对象标识符集合的第 1 个、第 2 个和后续公共弧(分支)的特定对象标识符。根对象标识符和相对对象标识符构成完整的对象标识符。
- (17) **产品电子代码 (EPC: electronic product code)**: 国际组织推出的新一代产品标识代码, 可以对每个单品进行全球唯一的标识。目前有 96 位 EPC 编码、128 位 EPC 编码、256 位 EPC 编码等方式。
- (18) **锁定 (Lock)**: 通过软硬件指令等方式锁住 RFID 标签数据块使之不能被修改, 是一种保证数据完整性的方法。
- (19) **访问口令 (Access Password)**: 标签进行写、锁定等操作的访问密钥。
- (20) **灭活口令 (Kill Password)**: 使标签不产生调制信号以激活射频场, 从而失效的控制密钥。

## 4 遵循原则

- ◆ 提高 UHF-RFID 标签的读写速度与使用性能, 提升读者体验。
- ◆ 增加离线状态下 RFID 应用与互操作的可能性。
- ◆ 减少改写标签数据的频率, 尽量避免后续繁琐的人力操作。
- ◆ 不存放过多数据在 UHF-RFID 标签内, 确保标签的后续扩展性。

## 5 存储方式

本规范所定义的用户数据元素分别存储于标签与后台数据库, UHF-RFID 标签上的用户数据存储区域分 EPC 数据区与 USER 数据区(由于读写机制不同, EPC 数据区读写速度与优先级高于 USER 数据区)。

- (1) EPC 数据区: 放置图书馆 RFID 应用中常用的、对读写速度要求较高的用户数据元素。
- (2) USER 数据区: 放置图书馆 RFID 应用中不常用的、对读写速度要求不高或仅在馆际互借等情况下需要使用的用户数据元素。
- (3) RFID 后台数据库: 放置图书馆 RFID 应用中修改较频繁且不依赖于标签读取的用户数据元素。

表 1 给出了标签用户数据元素的存储方式; 而表 2 则列出了放置于 RFID 后台数据库的用户数据元素, 放置于标签上的用户数据元素也需要在 RFID 后台数据库中有相应的备份。原则上, RFID 后台数据库的用户数据元素不做压缩编码处理。此外, RFID 后台数据库还应包含一些统计用数据元素, 如借还次数、借还日期、出入门禁次数、盘点次数等。

RFID 后台数据库定期与图书管理系统进行单向更新操作, 用图书管理系统的数值更新 RFID 后台数据库中对应项的数据值。

表 1 UHF-RFID 标签数据元素存储方式

存储内容	数据区域
分拣、预留位、安全位	EPC
版本	
内容索引	
馆藏标识符	
所属馆标识	USER
卷册信息	
馆际互借借入馆	
备选的馆藏标识符	
扩展保留位	

表 2 RFID 后台数据库存储内容

存储内容	数据区域
标签 TID	RFID 后台数据库
馆藏类别与状态	
馆藏位置	
馆际互借事务号	
临时馆藏位置	
主题	
分馆标识	
国际标准书号/刊号	
统计字段(包含借还次数、借还日期、出入门禁次数、盘点次数等数据)	
分拣	
版本	
内容索引	RFID 后台数据库(标签数据备份)
馆藏标识符	
所属馆标识	
卷册信息	
馆际互借借入馆	
备选的馆藏标识符	

扩展保留位	
-------	--

☞	UHF-RFID 标签的 EPC+USER 总体容量是固定的, 但某些标签的 EPC 与 USER 区域大小可在出厂前进行自由调整, 例如可减小 USER 区域容量从而扩大 EPC 区域容量。如果图书馆选用的用户数据元素较多, 那么应根据实际情况选择能满足自己所需数据容量的 UHF-RFID 标签。
---	--

## 6 版本的更新机制

版本是数据模型版本的简称, 用于记载并提供数据模型与应用指南的升级与修订信息, 由联盟负责控制和统一发布。

### 6.1 管理方式

原则上, 数据模型与应用指南的版本修订、更新与发布等工作由联盟下设的工作小组 (目前设在上海交通大学图书馆) 负责具体处理。工作流程如下:

- (1) 用户或 RFID 供应厂商发现原数据模型存在缺陷、或建议修改数据模型中某些用户数据元素的定义或内涵时, 需要首先报送联盟工作小组;
- (2) 联盟工作小组也会定期收集、研讨数据模型和应用指南的修改意见和建议;
- (3) 联盟工作小组在汇总各种用户和/或厂商对数据模型与应用指南的修订意见和建议后, 定期进行分析、测评、审核;
- (4) 在广泛征求意见和充分论证、审定升级数据模型版本和/或应用指南之后, 统一对外发布升级版本 (其中, 重点是通知 RFID 供应厂商在相应设备和系统中补充更新版本);
- (5) 用户在得到版本升级信息之后, 可自行决定是否需要及时升级。

### 6.2 兼容性与升级方式

联盟的核心思路是确保所有升级的新数据模型版本可向下兼容以前的旧版本, 因此, 用户可根据自身的条件决定是否需要进行版本升级, 但是, 在涉及馆际交流等情况下, 建议能及时完成版本升级, 以便确保馆际互借等操作的充分可靠性和有效性。

联盟发布新的升级版本之后, RFID 供应厂商必须及时根据说明文件更新、修订相应的设备、软件数据库以及系统等相关内容, 并及时告知所有关联客户。

## 7 EPC数据区存储方式说明

由于各标签的 EPC 数据区容量并不相同, 而且所能容纳的用户数据元素长度也很有限, 因此, 本应用指南将针对不同的 EPC 容量, 定义对应的用户数据元素的编码与存储方式。

建议在可能的条件下, 尽量采用紧凑的编码方式。例如:

- 当条码最多前 3 位可能出现大写字母时, 可采用编码方式 1, 即, 采用 96bit 的 EPC 区域;
- 当条码各位均有可能出现大写字母时, 可采用编码方式 2, 即, 采用 128bit 的 EPC 区域;
- 当条码各位均有可能出现 ISO/IEC646 IRV 所定义的任意字符时, 只能采用编码方式 3, 即, 采用 144bit 或以上的 EPC 区域。

## 7.1 存储方式

### 7.1.1 编码方式 1: 标签的EPC容量为 96bit

适用情况: 馆藏标识符总长度为 1-14 位字符。

- 馆藏标识符长度为 1-7: 可选字符集为 ISO/IEC646 IRV 中的任意字符。
- 馆藏标识符长度为 8-14: 第 1 至 3 位字符的可选字符集为 A-Z 字母或 0-9 数字(如有小写字母需转换为大写字母再传入), 第 4 至倒数第 2 位的可选字符集为 0-9 数字, 最后 1 位的可选字符集为 ISO/IEC646 IRV 中的任意字符。

EPC 数据区存储方式如表 3 所示。

表 3 EPC 数据区存储方式

字节顺序	第 1 字节			第 2 字节		第 3-4 字节	第 5-12 字节
长度(Byte)	1			1		2	8
bit 顺序	高	←	低	高	←	低	高 ← 低
地址(bit)	7	6-5	4-0	15-14	13-8	31-16	95-32
字段顺序	安全位	预留位	分拣	编码方式	版本	内容索引	馆藏标识符

### 7.1.2 编码方式 2: 标签的EPC容量为 128bit

适用情况: 馆藏标识符总长度为 1-14 位字符。

- 馆藏标识符长度为 1-11: 可选字符集为 ISO/IEC646 IRV 中的任意字符。
- 馆藏标识符长度为 12-14: 第 1 至倒数第 2 位的可选字符集为 A-Z 字母或 0-9 数字(如有小写字母需转换为大写字母再传入), 最后 1 位的可选字符集为 ISO/IEC646 IRV 中的任意字符。

EPC 数据区存储方式如表 4 所示。

表 4 EPC 数据区存储方式

字节顺序	第 1 字节(高位)			第 2 字节		第 3-4 字节	第 5-16 字节(低位)
长度(Byte)	1			1		2	12
bit 顺序	高	←	低	高	←	低	高 ← 低
地址(bit)	7	6-5	4-0	15-14	13-8	31-16	127-32
字段顺序	安全位	预留位	分拣	编码方式	版本	内容索引	馆藏标识符

### 7.1.3 编码方式 3: 标签的EPC容量为 144bit或以上

适用情况: 馆藏标识符总长度不大于 14 个字符, 可选字符集为 ISO/IEC646 IRV 中的任意字符。(例如, 14 位字符中任意位出现小写字母或特殊字符的情况)。

EPC 数据区存储方式如表 5 所示。

表 5 EPC 数据区存储方式

字节顺序	第 1 字节			第 2 字节		第 3-4 字节	第 5-18 字节
长度(Byte)	1			1		2	14
bit 顺序	高	←	低	高	←	低	高 ← 低
地址(bit) (起始位是 0)	7	6-5	4-0	15-14	13-8	31-16	143-32
字段顺序	安全位	预留位	分拣	编码方式	版本	内容索引	馆藏标识符

### 7.1.4 编码方式 4: 保留

(暂无定义)。

### 7.1.5 推荐EPC存取操作方式

在实际操作 EPC 存取数据时, 可以采用如下数据结构完成快捷处理。

#### ☑ 定义

```

union
{
    unsigned char ucStr[18];
    struct
    {
        unsigned char ucSorting;           // 分拣
        unsigned char ucVersion;          // 版本
        unsigned char ucContentIndex[2];  // 内容索引
        unsigned char ucItemIdentifier[14]; // 馆藏标识符
    } sPackDat;
} unEPCDat;

```

#### ☑ 读取或写入例

```

读入:
for( i=0; i<18; ++i )
{
    unEPCDat.ucStr[i] = SourceBuf[i];
}
写出:
for( i=0; i<18; ++i )
{
    DestinationBuf[i] = unEPCDat.ucStr[i];
}

```

#### ☑ 操作或处理例

```

获得安全位信息: iStatus = (unsigned int)UnEPCDat.ucStr.ucSorting & 0x80;
获取编码方式:   iCodingType = ((unsigned int)UnEPCDat.ucStr.ucVersion & 0xC0) >> 6;

```

## 7.2 检查与更新机制

实际应用过程中, 需要配置 RFID 后台数据库, 用于存储/备份 RFID 标签 EPC 数据区中记录的数据, 也可以用于批量预设置选项值, 以便批量修改 RFID 标签 EPC 数据区中的某些数据项。因此, 保持 EPC 数据区中的数据项与 RFID 后台数据库的同步十分重要。

这里所给出的检查与更新机制, 其目的是要提供一种在应用过程中及时检查与更新 RFID 标签 EPC 数据区中的用户数据元素内容、确保 EPC 数据区内容与 RFID 后台数据库始终保持对应一致的操作方法。一般原则是, 首先更改 RFID 后台数据库中 with RFID 标签所对应的数据项。例如:

- (1) 当图书馆藏位置信息发生变化时, 修改 RFID 后台数据库中的分拣信息;
- (2) 当版本升级或改变时, 根据联盟发布的说明文件修改 RFID 后台数据库中的对应内容;
- (3) 当用户数据元素状态发生变化时, 修改 RFID 后台数据库中的内容索引值。

通常, 当 RFID 读写器进行 RFID 标签写入操作 (如: 借、还书过程) 时, 需要执行检查与更新操作。检查与更新的具体操作流程 (如图 1 所示) 为:

- (1) 首先读取 RFID 后台数据库的相关内容并放入一个专用数据 Buffer 中;
- (2) 同时, 读取 RFID 标签的 EPC 数据区内容;
- (3) 然后, 进行比对。若一致, 则直接写入 EPC 数据区; 否则, 如发现不一致, 则将该信息记入系统日志, 并且, 用数据库 (已在 Buffer 中) 的值改写 EPC 数据区。

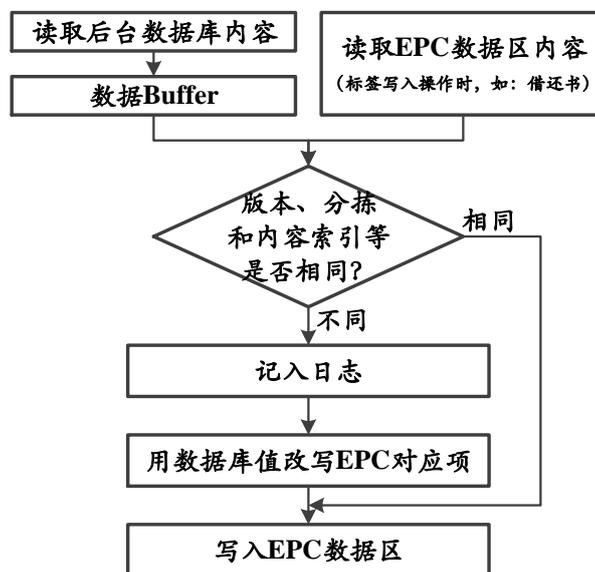


图 1 EPC 数据区检查与更新流程

## 8 USER数据区存储方式说明

### 8.1 存储方式

USER 数据区的用户数据元素存放方式为变长数据块形式, 每一个用户数据元素由“相对-OID 号”、“用户数据元素长度”与“用户数据元素内容”组, 数据结构如表 7 所示。

- ☑ 相对-OID 号: 占用 6bit, 与规范中用户数据元素的序号对应, 最多可表示 64 个不同的用户数据元素。
- ☑ 用户数据元素长度: 占用 10bit, 最大可表示 128Byte 的数据元素长度。
- ☑ 用户数据元素内容: 不定长, 根据具体的用户数据元素取值而定, 最小单位为 2Byte, 按字 (Word, 双字节) 对齐, 不足部分补 0。

表 7 USER 区数据结构

长度(Byte)	2		变长(2Byte 的倍数)
长度(bit)	6	10	变长
组成结构	相对-OID 号	用户数据元素长度	用户数据元素内容

图书馆可根据需求选择 USER 数据区存放的用户数据元素, 数据的存储方式如表 8 所示。

表 8 USER 数据区存储方式

相对-OID号1	用户数据元素1长度	用户数据元素1内容
相对-OID号2	用户数据元素2长度	用户数据元素2内容
相对-OID号3	用户数据元素3长度	用户数据元素3内容
.....		读取长度(最大使用长度)
USER区总长度		

USER 数据区的读写机制与 EPC 数据区不同, 需设定起始位置与数据长度。如图书馆使用的 USER 数据区最大长度固定, 那么在 RFID 系统软件配置表中记录该数值, 每次读写时进行调用, 即从 USER 数据区的起始位开始, 读取该标签 USER 区现使用最大长度的所有数据。

### 8.2 工作流程与更新机制

USER 数据区目前存放的用户数据元素为三项: 所属馆标识、卷册信息、馆际互借借入馆。由于所属馆标识与卷册信息为固定项, 因此将其置于 USER 数据区的第一、二项; 而馆际互借借入馆为动态变化项, 因此将其置于 USER 数据区的第三项, 方便数据的快速读写。如表 9 所示:

表 9 USER 数据区存储顺序

相对-OID号: 3	所属馆标识长度	所属馆标识
相对-OID号: 4	卷册信息长度	卷册信息
相对-OID号: 11	馆际互借借入馆长度	馆际互借借入馆

在 USER 数据区中, 只有馆际互借借入馆是动态变化项, 因此在馆际互借的工作流程中实现 USER 数据区的更新机制。操作的前提是图书馆管理系统支持馆际互借标准协议 ISO10160 和 ISO10161。

#### 8.2.1 借出操作流程

通常, 当收到馆际互借请求后, 需要执行馆际互借借出操作与标签数据的更新操作。具体操作流程 (如图 2 所示) 为:

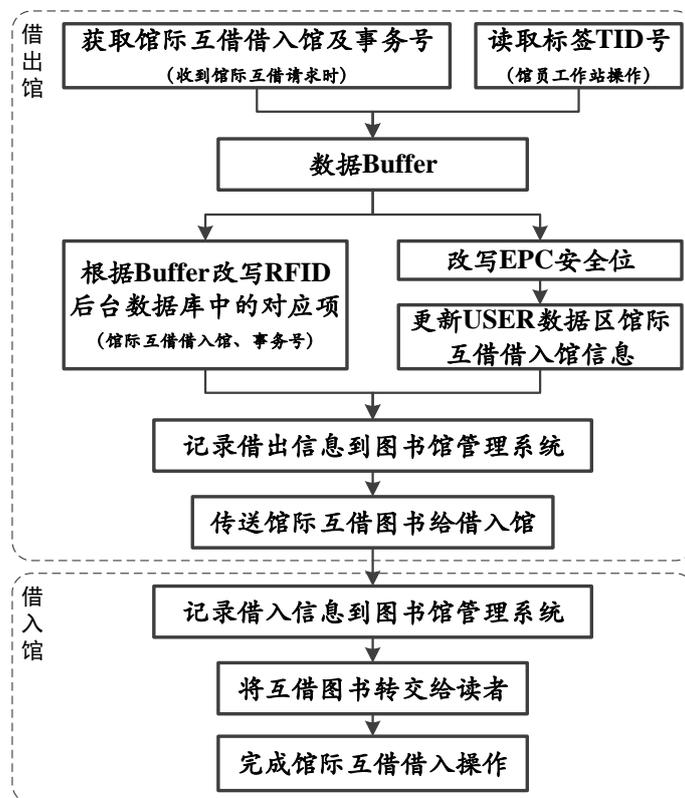


图 2 借出操作流程

- (1) 借出馆在收到馆际互借请求之后, 首先通过图书馆管理系统获得馆际互借借入馆代码与馆际互借事务号; 同时, 找到相应图书, 并通过馆员工作站读取标签的 TID 号;
- (2) 将上述馆际互借借入馆、馆际互借事务号和标签 TID 号一起放入专用数据 Buffer 中;

- (3) 按数据 Buffer 的内容修改 RFID 后台数据库中的馆际互借借入馆、馆际互借事务号信息; 同时修改 EPC 数据区的安全位, 并更新 USER 数据区的馆际互借借入馆信息;
- (4) 将相应图书的借出信息登记到图书馆管理系统中;
- (5) 发送相关馆际互借图书给借入馆
- (6) 借入馆在收到馆际互借图书之后, 首先记录相关馆际互借借入信息到图书馆管理系统中;
- (7) 然后将通过馆际互借所取得的图书转交给读者。

### 8.2.2 归还操作流程

通常, 在归还馆际互借的图书时, 需要执行馆际互借归还操作与标签数据的更新操作(只有借出馆能修改 RFID 标签内容)。具体操作流程(如图 3 所示)为:

- (1) 读者在借入馆归还馆际互借图书
- (2) 借入馆将相关归还信息登记到图书馆管理系统中(注意: 借入馆不能修改 RFID 标签的内容);
- (3) 借入馆将图书归还给借出馆;
- (4) 借出馆在收到图书之后, 通过馆员工作站修改标签 EPC 数据区的安全位(标记为归还)。注意: 为了提高操作效率, 原则上可以不用更新 USER 数据区内容, 任其保留至下次再次出借时重新写入新的馆际互借借入馆代码;
- (5) 之后, 在图书馆管理系统中做相应的归还信息登记, 标识图书已归还至本馆。

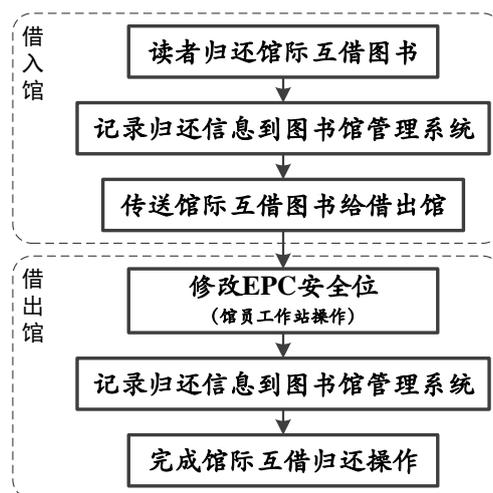


图 3 归还操作流程

## 9 数据元素取值与编码说明

### 9.1 安全位、预留位、分拣信息

- (1) 存储位置: EPC 的第 1 字节
- (2) 说明: 该用户数据元素包含三项内容, 安全位、预留位、分拣信息, 是需要首先读取的重要内容。
- (3) 取值方式: 1 字节定长字段。
- (4) 取值示例:
  - 第 7bit (最高 bit) 为安全位, 有 0 和 1 两个取值, 0 代表未出借状态 (馆内存放), 1 代表出借状态。
  - 第 5-6bit 为预留位, 例如可用于扩展分拣信息(初始化值为 00)。
  - 第 0-4bit 为分拣信息, 可用于标识馆藏分拣时所属的分拣箱号, 目前共定义了 32 种不同的

取值。

- (5) 编码方式: 不压缩。

### 9.2 编码方式及版本

- (1) 存储位置: EPC 的第 2 字节
- (2) 说明: 该用户数据元素包含编码方式和数据模型版本两项内容。
- (3) 取值方式: 1 字节定长字段, 高 2bit 用于说明压缩方式, 低 6bit 用于指定数据模型版本。
- (4) 取值示例:

- 第 6-7bit (最高 2bit) 为编码方式, 其中:
  - ✧ 00 编码方式 1, EPC 容量为 96bit (12Byte)
  - ✧ 01 编码方式 2, EPC 容量为 128bit (16Byte)
  - ✧ 10 编码方式 3, EPC 容量为 144bit 或以上 (≥18Byte)
  - ✧ 11 编码方式 4, 保留 (暂无定义)
- 第 0-5bit (低 6bit) 为版本说明, 可用于标识 64 种不同的版本信息(0-63)。

<b>版本</b>	编码方式 1	数据模型版本 3
<b>取值</b>	00 <sub>2</sub>	000010 <sub>2</sub>

- (5) 编码方式: 不压缩。

### 9.3 内容索引

- (1) 存储位置: EPC 的第 3-4 字节
- (2) 说明: 该用户数据元素用于定义或描述用户数据元素的选用状态, 以便索引、并确定进一步所需进行的操作。
- (3) 编码方式: 不压缩。
- (4) 取值方式: 二进制位序列。
- (5) 取值示例:
  - 如果采用了用户数据元素的相对-OID 值为 3、12、15 的项, 则内容索引编码为: 00A1<sub>HEX</sub> 或 00000000<sub>2</sub> 10100001<sub>2</sub>, 对应关系参见表 10。

表 10 OID 索引映射图实例

Bit 顺序	高 ← 低															
相对-OID	31	30	29	28	27	26	24	16	15	14	12	11	6	5	4	3
编码	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1

### 9.4 馆藏标识符

- (1) 存储位置: EPC 的第 5-12/16/18 字节
- (2) 说明: 该用户数据元素是 RFID 应用及日常管理必不可少的常用数据项之一, 并且是与图书馆管理系统直接关联的字段。
- (3) 取值方式: 馆藏条形码号。
- (4) 取值示例:

<b>馆藏标识符</b>	条码号
<b>取值</b>	Z20120001%

- (5) 编码方式:
  - 编码方式 1: 针对 EPC 容量为 96bit

适用情况: 馆藏标识符总长度为 1-14 位的由字母和数字组成的字符串。当:

- ◇ 馆藏标识符长度为 1-7: 可选字符集为 ISO/IEC646 IRV 中的任意字符。
- ◇ 馆藏标识符长度为 8-14: 第 1 至 3 位字符的可选字符集为 A-Z 字母或 0-9 数字(如有小写字母需转换为大写字母再传入), 第 4 至倒数第 2 位的可选字符集为 0-9 数字, 最后 1 位的可选字符集为 ISO/IEC646 IRV 中的任意字符。

处理方式: 参照附录 A 所给定的 96bit EPC 编码与解码算法进行处理。

➤ 编码方式 2: 针对 EPC 容量为 128bit

适用情况: 馆藏标识符总长度为 1-14 位字符。当:

- ◇ 馆藏标识符长度为 1-11: 可选字符集为 ISO/IEC646 IRV 中的任意字符。
- ◇ 馆藏标识符长度为 12-14: 第 1 至倒数第 2 位的可选字符集为 A-Z 字母或 0-9 数字(如有小写字母需转换为大写字母再传入), 最后 1 位的可选字符集为 ISO/IEC646 IRV 中的任意字符。

处理方式: 参照附录 A 所给定的 128bit EPC 编码与解码算法进行处理。

➤ 编码方式 3: 针对 EPC 容量为 144bit 或以上

适用情况: 馆藏标识符总长度不大于 14 个字符, 可选字符集为 ISO/IEC646 IRV 中的任意字符。

处理方式: 直接存取 (不做任何编码、解码处理)。

➤ 编码方式 4: 保留 (暂无定义)

### 9.5 所属馆标识

- (1) 存储位置: USER
- (2) 说明: 该用户数据元素是用于区分馆藏所属馆的标识。本馆服务时, 属于不常用字段。
- (3) 取值方式: 2 字节整型数。参照中华人民共和国教育部行业标准 JY/T1001-2012, 《教育管理信息-教育管理基础代码》中的《中国高等院校代码表》取值, 以 5 位数字所代表的高等院校代码来标识所属馆。其中, 针对首位数字为 9 (军事院校, 例如: 90001 代表国防大学) 的情况, 在存放时, 需要将 9 变成为 6, 然后以 2 字节整型数存储, 在读取后, 需要将 6 变成为 9, 恢复成原始代码。
- (4) 取值示例:

所属馆标识	上海交通大学图书馆
取值	10248

- (5) 编码方式: 2 字节整型数。

### 9.6 卷册信息

- (1) 存储位置: USER
- (2) 说明: 该用户数据元素用于标识一卷具有多册数的馆藏的卷册总数和卷册序号, 在 RFID 应用中属于不常用字段。可在离线状态下完成判断。注: 假如图书馆的馆藏政策是每个条码都对应多卷册的一个单册, 则不需要使用该用户数据元素。
- (3) 取值方式: 总 4 字节定长字段, 其中: 高 2 字节存放卷册总数 (最多 65536 卷册), 低 2 字节存放卷册序号 (1 - 65536)。
- (4) 取值示例:

卷册信息	某总数为 100 册的多卷册馆藏的第 5 册
取值	100 / 5

- (5) 编码方式: 两个 2 字节整型数。

## 9.7 馆际互借借入馆

- (1) 存储位置: USER
- (2) 说明: 该用户数据元素用于标识馆藏的借入馆, 即馆际互借中的请求馆信息。通过该用户数据元素可方便地从标签上直接获取到借入馆信息。注: 该用户数据元素的修改控制方为出借馆藏的所属馆, 而非借入请求馆。
- (3) 取值方式: 2 字节整型数。参照中华人民共和国教育部行业标准 JY/T1001-2012, 《教育管理信息-教育管理基础代码》中的《中国高等院校代码表》取值, 以 5 位数字所代表的高等院校代码来标识所属馆。其中, 针对首位数字为 9 (军事院校, 例如: 90001 代表国防大学) 的情况, 在存放时, 需要将 9 变成为 6, 然后以 2 字节整型数存储, 在读取后, 需要将 6 变成为 9, 恢复成原始代码。
- (4) 取值示例:

馆际互借借入馆	上海交通大学图书馆
取值	10248

- (5) 编码方式: 2 字节整型数。

## 9.8 馆际互借事务号

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素是表示馆际互借事务的唯一编码, 与馆际互借系统中的编号一致对应。
- (3) 取值方式: 由图书馆管理系统或馆际互借系统确定。
- (4) 取值示例: CALIS2012210000536。
- (5) 编码方式: 无。

## 9.9 馆藏类别与状态

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素可用于区分馆藏的所属种类及状态。属于非常用字段, 且变更较为频繁, 因此, 不考虑依赖标签存取。
- (3) 取值方式: 总长 2 字节定长字段, 其中: 高字节存放馆藏类别(主限定标识), 低字节存放馆藏状态(次限定标识)。类别值及状态值的定义, 参照《第一部分: 数据模型规范》附录 B。
- (4) 取值示例:

应用类别	某不可外借光盘
取值	1 / 1

## 9.10 分馆标识

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素用于进一步定义馆藏所属的下级分馆或所在地。属于非常用字段, 且变更较为频繁, 因此, 不考虑依赖标签存取。
- (3) 取值方式: 通常由图书馆管理系统决定, 应参照相应图书馆管理系统的相关字段取值。
- (4) 取值示例:

分馆标识	上海交通大学图书馆的主馆
取值	LTSKJ

### 9.11 馆藏位置

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素用于定义馆藏的具体位置, 以便为整架、自动分拣或快速定位等提供方便。属于非常用字段, 且变更较为频繁, 因此, 不考虑依赖标签存取。
- (3) 取值方式: 通常由图书馆管理系统决定, 应参照相应图书馆管理系统的相关字段取值。
- (4) 取值示例:

<b>馆藏位置</b>	上海交通大学图书馆的主馆 A 区 3 楼理科阅览室
<b>取值</b>	LA3ZJ

### 9.12 临时馆藏位置

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素用于定义馆藏的临时存放位置。属于非常用字段, 且变更较为频繁, 因此, 不考虑依赖标签存取。
- (3) 取值方式: 通常由图书馆管理系统决定, 应参照相应图书馆管理系统的相关字段取值。
- (4) 取值示例:

<b>临时馆藏位置</b>	上海交通大学图书馆的主馆 A 区 3 楼理科阅览室
<b>取值</b>	LA3ZJ

### 9.13 主题

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素用于定义馆藏所属的分类主题信息。属于非常用字段, 可不依赖标签存取。
- (3) 取值方式: 中国图书馆分类法<sup>[11]</sup>。
- (4) 取值示例:

<b>主题</b>	通信理论
<b>取值</b>	TN911

### 9.14 国际标准书/刊号

- (1) 存储位置: RFID 后台数据库
- (2) 说明: 该用户数据元素用于标识馆藏所属的 ISBN/ISSN 号信息, 可用于在图书管理系统中快速获取馆藏信息, 如题名等, 方便判断对应馆藏是否为本馆馆藏。属于非常用字段, 可不依赖标签存取。
- (3) 取值方式: 通常由图书馆管理系统决定, 应参照相应图书馆管理系统的相关字段取值。
- (4) 取值示例: ISBN 编号通常为 10 位或 13 位的数字字符串, ISSN 编号则为 8 位的数字字符串。

<b>书名</b>	RFID 物联网世界最新应用
<b>国际标准书号(ISBN)</b>	9787564062507

## 10 操作规范

### 10.1 TID 区域读写

TID 区在读写时应确保数据一次性全部读出, 然后再对数据长度做判断, 以适应各类不同 TID 容量标签的情况。读写器应按照此标准进行读写操作。

## 10.2 锁定与口令

### (1) 锁定

为确保标签数据的安全性, 标签用户数据元素需进行锁定, 进行标签数据写操作时先执行解锁操作。使用一般锁定指令, 不使用永久锁定指令。锁密钥由各图书馆自行定义。

### (2) 口令

标签设置统一的访问口令, 口令密钥由各图书馆自行定义。

### (3) 安全性

锁与口令相结合可确保标签数据的安全性, 防止数据被未授权的读写器改写。

## 11 基于ISO28560 推荐的ISO/IEC15962 编码与压缩规则

附录 A 所使用的编码与解码算法基于 ISO/IEC15962 编码与压缩规则, 实际处理时, 采用了多种压缩方式的组合应用。



更详细的压缩编码规则请参考 ISO28560-2 标准。

## 附录 A: 编码与解码算法 (资料性附录)

### A1 适用情况

本编码与解码算法适用于总长度为 1-14 位字符条形码 (馆藏标识符) 的情况。

☞	截止 2012 年 11 月 22 日, 本应用指南出台时, 中国高等院校图书馆均采用条形码作为馆藏标识符, 其长度范围在 5-14 位字符 (字母+数字组成) 范围之内。
---	--

#### A1.1 针对 EPC 为 96bit 长度

当条码长度为 1-7: 可选字符集为 ISO/IEC646 IRV 中的任意字符。

当条码长度为 8-14: 第 1 至 3 位字符的可选字符集为 A-Z (大写) 字母或 0-9 数字 (如有小写字母需转换为大写再传入), 第 4 至倒数第 2 位的可选字符集为 0-9 数字, 最后 1 位 (常被用作校验位) 的可选字符集为 ISO/IEC646 IRV 中的任意字符。

☞	编码后数据的统一长度为 8 字节。当条码总长度小于 7 位字符时, 编码后数据长度为不定值, 不足部分的高位字节可用任意数值补齐 8 字节。
---	--

#### A1.2 针对 EPC 为 128bit 长度

当条码长度为 1-11: 可选字符集为 ISO/IEC646 IRV 中的任意字符。

当条码长度为 12-14: 第 1 至倒数第 2 位的可选字符集为 A-Z 字母或 0-9 数字 (如有小写字母需转换为大写再传入), 最后 1 位的可选字符集为 ISO/IEC646 IRV 中的任意字符。

☞	编码后数据的统一长度为 12 字节。当条码总长度小于 11 位字符时, 编码后数据长度为不定值, 不足部分的高位字节可用任意数值补齐 12 字节。
---	---

### A2 数据结构

#### A2.1 EPC 96bit 数据结构

当条码长度为 8-14 位字符时, 编码后的数据结构如表 A-1 所示。

当条码长度为 1-7 位字符时, 编码后的数据结构如表 A-2 所示。

表 A-1 条码长度为 8-14

bit 顺序	高 ←			低		
长度(Byte)	4			4		
长度(bit)	30	2	2	18	8	4
数据结构	条码第 5 位到倒数第 2 位编码	条码第 4 位编码的高 2bit	条码第 4 位编码的低 2bit	条码前 3 位编码	条码最后 1 位(校验位)	条码长度

表 A-2 条码长度为 1-7

bit 顺序	高 ←				低	
长度(Byte)	1-7 字节不定长, 高位				1	
长度(bit)	8	……	8	8	4	4
数据结构	条码第 7 位	……	条码第 2 位	条码第 1 位	空位	条码长度

### A2.2 EPC 128bit 数据结构

条码长度为 12-14, 数据结构如表 A-3 所示。

条码长度为 1-11, 数据结构如表 A-4 所示。

表 A-3 条码长度为 12-14

bit 顺序	高		←	低	
长度(Byte)	4	4		4	
长度(bit)	30	30	18	8	4
数据结构	条码第 1 位到第 5 位	条码第 6 位到第 10 位	条码第 11 位到倒数第 2 位	条码最后 1 位(校验位)	条码长度

表 A-4 条码长度为 1-11

bit 顺序	高				←	低	
长度(Byte)	1-11 字节不定长, 高位					1	
长度(bit)	8		8	8	4	4	
数据结构	条码第 11 位	……	条码第 2 位	条码第 1 位	空位	条码长度	

## A3 编码程序源代码(C语言)

### A3.1 EPC 96bit 编码程序源代码 (供 EPC 中编码方式为 00[方式 1]时调用)

```

/*-----+*/
/* iRFID_Encode96bit( ) */
/*
/* Input: ucBarCode, 输入时, 是一个1-14Byte的无符号字符串 */
/* 由于结束符'\0' 占用一个空间, 最大需要开辟15Byte空间 */
/* ucBarCode字符数为1-7: */
/* 可选字符集为ISO/IEC646 IRV中的任意字符 */
/* ucBarCode字符数为8-14: */
/* 第1至3位字符的可选字符集为A-Z字母或0-9数字(小写字母需转换为大写再传入) */
/* 第4至倒数第2位的可选字符集为0-9数字 */
/* 最后1位的可选字符集为ISO/IEC646 IRV中的任意字符 */
/* Output: 输出时, ucBarCode是一个2-8Byte的无符号字符串 */
/* 返回值等于BarCode长度, -1代表输入条码长度不合法(输入条码长度介于1-14之间) */
/*
/* 说明: 用于压缩1-14位条码至2-8Byte */
/* ucBarCode字符数为1-7: */
/* 可选字符集为ISO/IEC646 IRV中的任意字符 */
/* ucBarCode字符数为8-14: */
/* 第1至3位字符的可选字符集为A-Z字母或0-9数字(小写字母需转换为大写再传入) */
/* 第4至倒数第2位的可选字符集为0-9数字 */
/* 最后1位的可选字符集为ISO/IEC646 IRV中的任意字符 */
/*
/* Version 2.0 */
/* 完成日期: 2012-11-16 */
/*-----+*/
int iRFID_Encode96bit( unsigned char *ucBarCode )
{
    int i, iBarCodeLen;
    unsigned char ucCompressedBarCode[4], ucTmpChar;
    unsigned char *pucTmpDat;
    union
    {
        unsigned char ucStr[4];
        unsigned int uiDat;
    } unWorkDat;

    iBarCodeLen = strlen( (char *)ucBarCode );
    if( iBarCodeLen < 1 || iBarCodeLen > 14 ) return ( -1 );
}
    
```

```

if( iBarCodeLen < 8 )                // 处理1-7字符长度的BarCode
{
    for( i=iBarCodeLen; i>0; --i )
    {
        ucBarCode[i] = ucBarCode[i-1];
    }
    ucBarCode[0] = (unsigned char)iBarCodeLen;
    // 存放原条码长度到低4bit
    return ( iBarCodeLen );
}

ucTmpChar = ucBarCode[iBarCodeLen - 1]; // 临时存放BarCode校验位
ucBarCode[iBarCodeLen - 1] = 0;        // 最后一位(即校验位)清零

pucTmpDat = &ucBarCode[4];            // 处理压缩后的高4字节
unWorkDat.uiDat = atoi( (char *)pucTmpDat ); // 压缩BarCode从第5位起除最后1位的2-9位数字
unWorkDat.uiDat <<= 2;
unWorkDat.uiDat |= ( ( (unsigned int)ucBarCode[3] - 0x30 ) & 0x0C )
                >> 2;                // 将Barcode的第4位数字压缩后的高2bit存入

for( i=0; i<4; ++i )                // 临时保存压缩后的高4字节
{
    ucCompressedBarCode[i] = unWorkDat.ucStr[i];
}
// 处理压缩后的低4字节
unWorkDat.uiDat = ( (unsigned int)ucBarCode[3] - 0x30 )
                & 0x03 ;            // 压缩第4位数字压缩后的低2bit存入

unWorkDat.uiDat = (unWorkDat.uiDat << 6)
                | ((unsigned int)ucBarCode[0] - 0x30); // 压缩最前面3位字母或数字
unWorkDat.uiDat = (unWorkDat.uiDat << 6) | ((unsigned int)ucBarCode[1] - 0x30);
unWorkDat.uiDat = (unWorkDat.uiDat << 6) | ((unsigned int)ucBarCode[2] - 0x30);

unWorkDat.uiDat = (unWorkDat.uiDat << 8) | ((unsigned int)ucTmpChar );
// 压缩BarCode最后1位字符, 通常为校验码

unWorkDat.uiDat = (unWorkDat.uiDat << 4) | (unsigned int)iBarCodeLen;
// 记录原BarCode编码长度

for( i=0; i <4; ++i )                // 构成输出压缩后的BarCode
{
    ucBarCode[i] = unWorkDat.ucStr[i];
    ucBarCode[i + 4] = ucCompressedBarCode[i];
}

return ( iBarCodeLen );
}

```

### A3.2 EPC 128bit编码程序源代码 (供EPC中编码方式为 01[方式 2]时调用)

```

/*-----*/
/*| iRFID_Encode128bit( ) |*/
/*| |*/
/*| Input: ucBarCode, 输入时, 是一个1-14Byte的无符号字符串 |*/
/*| 由于结束符'\0' 占用一个空间, 最大需要开辟15Byte空间 |*/
/*| ucBarCode字符数为1-11: |*/
/*| 可选字符集为ISO/IEC646 IRV中的任意字符 |*/
/*| ucBarCode字符数为12-14: |*/
/*| 第1至倒数第2位可选字符集为A-Z字母或0-9数字(小写字母需转换为大写再传入) |*/
/*| 最后1位的可选字符集为ISO/IEC646 IRV中的任意字符 |*/
/*| Output: 输出时, ucBarCode是一个2-12Byte的无符号字符串 |*/

```

```

/*      返回值等于BarCode长度, -1代表输入条码长度不合法(输入条码长度介于1-14之间) */
/*      */
/*      说明: 用于压缩1-14位条码至2-12Byte */
/*      ucBarCode字符数为1-11: */
/*      可选字符集为ISO/IEC646 IRV中的任意字符 */
/*      ucBarCode字符数为12-14: */
/*      第1至倒数第2位可选字符集为A-Z字母或0-9数字(小写字母需转换为大写再传入) */
/*      最后1位的可选字符集为ISO/IEC646 IRV中的任意字符 */
/*      */
/*      Version 2.0 */
/*      完成日期: 2012-11-16 */
/*-----+*/
int iRFID_Encode128bit( unsigned char *ucBarCode )
{
    int          i, iBarCodeLen;
    unsigned char ucCompressedBarCode[13] = "000000000000";
    unsigned char *pucTmpDat;
    union
    {
        unsigned char ucStr[4];
        unsigned int  uiDat;
    } unWorkDat;

    iBarCodeLen = strlen( (char *)ucBarCode );
    if( iBarCodeLen < 1 || iBarCodeLen > 14 ) return ( -1 );

    if( iBarCodeLen < 12 )                // 处理1-11字符长度的BarCode
    {
        for( i=iBarCodeLen; i>0; --i )
        {
            ucBarCode[i] = ucBarCode[i-1];
        }
        ucBarCode[0] = (unsigned char)iBarCodeLen;// 存放原条码长度到低4bit

        return ( iBarCodeLen );
    }

    pucTmpDat = ucBarCode;                // 处理BarCode的第1-5个字符
    unWorkDat.uiDat = 0;
    for( i=0; i<5; ++i )
    {
        unWorkDat.uiDat = (unWorkDat.uiDat << 6) | ((unsigned int)pucTmpDat[i] - 0x30);
    }
    for( i=0; i<4; ++i )
    {
        ucCompressedBarCode[i + 8] = unWorkDat.ucStr[i];
    }

    pucTmpDat = &ucBarCode[5];           // 处理BarCode的第6-10个字符
    unWorkDat.uiDat = 0;
    for( i=0; i<5; ++i )
    {
        unWorkDat.uiDat = (unWorkDat.uiDat << 6) | ((unsigned int)pucTmpDat[i] - 0x30);
    }
    for( i=0; i<4; ++i )
    {
        ucCompressedBarCode[i + 4] = unWorkDat.ucStr[i];
    }

    pucTmpDat = &ucBarCode[10];          // 处理BarCode的第11-倒数第二位字符
    unWorkDat.uiDat = 0;
    for( i=0; i<iBarCodeLen-11; ++i )

```

```

{
    unWorkDat.uiDat = (unWorkDat.uiDat << 6) | ((unsigned int)pucTmpDat[i] - 0x30);
}
unWorkDat.uiDat = (unWorkDat.uiDat << 8) | ( (unsigned int)( ucBarCode[iBarCodeLen - 1] ) );
    // 校验位
unWorkDat.uiDat = (unWorkDat.uiDat << 4) | (unsigned int)iBarCodeLen;
    // 记录原BarCode编码长度到低4bit

for( i=0; i<4; ++i )
{
    ucCompressedBarCode[i] = unWorkDat.ucStr[i];
}

for( i=0; i <12; ++i )                // 构成编码后的12字节BarCode数据
{
    ucBarCode[i] = ucCompressedBarCode[i];
}

return ( iBarCodeLen );
}

```

## A4 解码程序源代码(C语言)

### A4.1 EPC 96bit解码程序源代码（供EPC中编码方式为 00[方式 1]时调用）

```

/*-----*/
/*| iRFID_Decode96bit( ) |*/
/*| |*/
/*| Input: ucBarCode,输入时,应当是开辟15字节空间的变量存储一个2-8Byte的无符号字符串 |*/
/*| Output: 输出时, ucBarCode是一个1-14Byte(字符数)的字符 |*/
/*| 返回值等于BarCode长度, -1代表原始BarCode与解码后BarCode长度不一致 |*/
/*| |*/
/*| 说明: 用于解压缩条码, 将压缩为2-8Byte的条码解压缩为压缩前的1-14Byte条码 |*/
/*| |*/
/*| Version 2.0 |*/
/*| 完成日期: 2012-11-16 |*/
/*-----*/
int iRFID_Decode96bit( unsigned char *ucBarCode )
{
    int i, iDatLen, iBarCodeLen;
    unsigned char ucTmpChar;
    unsigned char ucTmpStr[9];
    unsigned char ucWorkStr[15] = "00000000000000";
    // 字符串长度实际为14位字符+结束符'\0'

    union
    {
        unsigned char ucStr[4];
        unsigned int uiDat;
    } unWorkDat;

    iBarCodeLen = (unsigned int)( ucBarCode[0] & 0x0F );
    // 获取压缩前BarCode编码长度
    if( iBarCodeLen < 1 || iBarCodeLen > 14 ) return ( -1 );

    if( iBarCodeLen < 8 )                // 处理1-7字符长度的BarCode
    {
        for( i=0; i<iBarCodeLen; ++i )
        {
            ucBarCode[i] = ucBarCode[i+1];
        }
        ucBarCode[iBarCodeLen] = 0;
    }
}

```

```

        return ( iBarcodeLen );
    }

    for( i=0; i<4; ++i )          // 将压缩码的前面(低)4字节取出
    {
        unWorkDat.ucStr[i] = ucBarcode[i];
    }

    unWorkDat.uiDat >>= 4;
    ucWorkStr[iBarcodeLen - 1] = (unsigned char)( unWorkDat.uiDat & 0xFF );
        // 还原最后1位字符, 即校验位

    unWorkDat.uiDat >>= 8;
    for( i=2; i>=0; --i )        // 解码最前面3位字母或数字
    {
        ucWorkStr[i] = (unsigned char)( ( unWorkDat.uiDat & 0x3F ) + 0x30 );
        unWorkDat.uiDat >>= 6;
    }

    ucWorkStr[3] = (unsigned char)( unWorkDat.uiDat & 0x03 );
        // 还原第四位数字的低2bit
    for( i=0; i<4; ++i )        // 将压缩码的前面(高)4字节取出
    {
        unWorkDat.ucStr[i] = ucBarcode[i+4];
    }

    ucWorkStr[3] = ( ucWorkStr[3] | ( (unsigned char)( unWorkDat.uiDat & 0x03 ) << 2 ) ) + 0x30;
// 取出第4位数字的低2bit, 还原第4位数字
    unWorkDat.uiDat >>= 2;
    itoa( unWorkDat.uiDat, (char *)ucTmpStr, 10 );
        // 还原Barcode从第5位起除最后1位的2-9位数字
    iDatLen = strlen( (char *)ucTmpStr );
    for( i=0; i<iDatLen; ++i )
    {
        ucWorkStr[i + iBarcodeLen - iDatLen - 1] = ucTmpStr[i];
    }

    for( i=0; i<iBarcodeLen; ++i ) // 将解码后的Barcode通过参数返回
    {
        ucBarcode[i] = ucWorkStr[i];
    }
    ucBarcode[iBarcodeLen] = 0;

    return ( iBarcodeLen );
}

```

**A4.2 EPC 128bit解码程序源代码 (供EPC中编码方式为 01[方式 2]时调用)**

```

/*-----*/
/* iRFID_Decode128bit( )                                     */
/*-----*/
/* Input:  ucBarcode, 输入时, 应当是开辟15字节空间的变量存储一个2-12Byte的无符号字符串 */
/* Output: 输出时, ucBarcode是一个1-14Byte (字符数) 的字符串 */
/*         返回值等于Barcode长度, -1代表原始Barcode与解码后Barcode长度不一致 */
/*-----*/
/* 说明: 用于解压缩条码, 将压缩为2-12Byte的条码解压缩为压缩前的1-14Byte条码 */
/*-----*/
/*                                     Version 2.0 */
/*                                     完成日期: 2012-11-16 */
/*-----*/
int iRFID_Decode128bit( unsigned char *ucBarcode )
{
    int          i, j, iDatLen, iBarcodeLen;
}

```

```

unsigned char   ucTmpChar;
unsigned char   ucTmpStr[9];
unsigned char   ucWorkStr[15] = "00000000000000";
                // 字符串长度实际为14位字符+结束符'\0'

union
{
    unsigned char ucStr[4];
    unsigned int  uiDat;
} unWorkDat;

iBarcodeLen = (unsigned int)( ucBarcode[0] & 0x0F );
                // 获取编码前Barcode编码长度
if( iBarcodeLen < 1 || iBarcodeLen > 14 ) return ( -1 );

if( iBarcodeLen < 12 )    // 处理1-12字符长度的Barcode
{
    for( i=0; i<iBarcodeLen; ++i )
    {
        ucBarcode[i] = ucBarcode[i+1];
    }
    ucBarcode[iBarcodeLen] = 0;

    return ( iBarcodeLen );
}

for( i=0; i<4; ++i )    // 将压缩码的低4字节取出
{
    unWorkDat.ucStr[i] = ucBarcode[i];
}

unWorkDat.uiDat >>= 4;
ucWorkStr[iBarcodeLen - 1] = (unsigned char)( unWorkDat.uiDat & 0xFF );
                // 还原最后1位字符, 即校验位
unWorkDat.uiDat >>= 8;
for( i=iBarcodeLen-2; i>=10; --i )    // 解码第11至倒数第2个字符
{
    ucWorkStr[i] = (unsigned char)( ( unWorkDat.uiDat & 0x3F ) + 0x30 );
    unWorkDat.uiDat >>= 6;
}

for( i=0; i<4; ++i )    // 将压缩码的中间4字节取出
{
    unWorkDat.ucStr[i] = ucBarcode[i+4];
}
for( i=9; i>=5; --i )    // 解码第6-10个字符
{
    ucWorkStr[i] = (unsigned char)( ( unWorkDat.uiDat & 0x3F ) + 0x30 );
    unWorkDat.uiDat >>= 6;
}

for( i=0; i<4; ++i )    // 将压缩码的高4字节取出
{
    unWorkDat.ucStr[i] = ucBarcode[i+8];
}
for( i=4; i>=0; --i )    // 解码第1-5个字符
{
    ucWorkStr[i] = (unsigned char)( ( unWorkDat.uiDat & 0x3F ) + 0x30 );
    unWorkDat.uiDat >>= 6;
}

for( i=0; i<iBarcodeLen; ++i )    // 将解码后的Barcode通过参数返回
{

```



```
        {
            printf( "\n解码成功, 解码后BarCode长度为%d, BarCode为%s",
                    iLen, ucCompressedBarCode[i] );
        }
    }
    printf( "\n%s", "" );
}

system("pause");
}
```

## A5.2 EPC 128bit测试程序源代码

```
/*-----*/
/* vTest128bit()                                     */
/*-----*/
/* Input:  无                                       */
/* Output: 无                                       */
/*-----*/
/* 说明: 测试方法, 通过给定各种典型参数用来测试iRFID_Encode与iRFID_Decode函数 */
/*-----*/
/*                                           Version 2.0 */
/*                                           完成日期: 2012-11-16 */
/*-----*/
void vTest128bit()
{
    int            i, iLen;
    unsigned char  ucCompressedBarCode[16][15]; // 最后1位需要存储结束符\0

    strcpy( (char *)ucCompressedBarCode[0], "A"); // 给定各种典型情况的输入参数
    strcpy( (char *)ucCompressedBarCode[1], "ABC012");
    strcpy( (char *)ucCompressedBarCode[2], "ABC0123*");
    strcpy( (char *)ucCompressedBarCode[3], "ABC01234=");
    strcpy( (char *)ucCompressedBarCode[4], "ABC012345~");
    strcpy( (char *)ucCompressedBarCode[5], "ABC0123456a");
    strcpy( (char *)ucCompressedBarCode[6], "ABC01234567z");
    strcpy( (char *)ucCompressedBarCode[7], "ABC012345678*");
    strcpy( (char *)ucCompressedBarCode[8], "ABC0123456789?");
    strcpy( (char *)ucCompressedBarCode[9], "ABC0123456789_");
    strcpy( (char *)ucCompressedBarCode[10], "A00000@");
    strcpy( (char *)ucCompressedBarCode[11], "AA0000@");
    strcpy( (char *)ucCompressedBarCode[12], "AAA90000000000");
    strcpy( (char *)ucCompressedBarCode[13], "ZZZ80007000000");
    strcpy( (char *)ucCompressedBarCode[14], "99999999999999");
    strcpy( (char *)ucCompressedBarCode[15], "ZZZZZZZZZZZZZZ");

    for( i=0; i<16; ++i )
    {
        printf( "\n原始BarCode为:%s", ucCompressedBarCode[i] );
        iLen = iRFID_Encode128bit( ucCompressedBarCode[i] );
        if(iLen == -1)
        {
            printf( "\n%s", "编码失败, 原始BarCode长度错误:BarCode应为7-14位的字符串" );
        }
        else
        {
            printf( "\n编码成功, 原始BarCode长度为%d", iLen );
            iLen = iRFID_Decode128bit( ucCompressedBarCode[i] );
            if(iLen == -1)
            {
                printf( "\n%s", "解码失败, 原始BarCode与解码后BarCode长度不一致" );
            }
        }
    }
}
```

```
        else
        {
            printf( "\n解码BarCode为:%s", ucCompressedBarCode[i]);
            printf( "\n解码成功, 解码后BarCode长度为%d", iLen);
        }
    }
    printf( "\n%s", "" );
}

system("pause");
}
```

## 参考文献

- [1] UHF RFID 技术应用的发展与最佳实践.内地与香港图书馆及 UHF RFID 业界交流会暨高校图书馆 UHF RFID 应用工作小组第三次会议网站.  
[http://www.cityu.edu.hk/lib/about/event/rfid\\_conf2011/index.html](http://www.cityu.edu.hk/lib/about/event/rfid_conf2011/index.html)
- [2] JY/T1001-2012. 教育管理信息-教育管理基础代码
- [3] WH/T 43-2012. 图书馆-射频识别-数据模型-第 1 部分: 数据元素设置及应用规则
- [4] WH/T 44-2012. 图书馆-射频识别-数据模型-第 2 部分: 基于 ISO/IEC15962 的数据元素编码方案
- [5] ISO/IEC646 IRV. 信息技术: ISO 信息交换七位编码字符集
- [6] ISO/IEC15961: 2004. 信息技术-项目管理用射频识别(RFID)-数据协议: 应用接口
- [7] ISO/IEC15962: 2004. 信息技术-项目管理用射频识别(RFID)-数据协议: 数据编码规则和逻辑存储功能
- [8] ISO/IEC18000-6. 信息技术-项目管理用射频识别-860MHz 至 960MHz 空中接口通信参数
- [9] ISO28560-1: 2011. 信息和文献-图书馆无线射频识别(RFID)-实施的数据元素和总原则
- [10] ISO28560-2: 2011. 信息和文献-图书馆无线射频识别(RFID)-基于 ISO/IEC15962 规则的无线射频识别(RFID)数据元素的编码
- [11] 国家图书馆. 中国图书馆分类法. 第 5 版. 北京: 国家图书馆出版社, 2010
- [12] 陈进等. 图书馆 RFID 技术及应用. 上海: 上海交通大学出版社, 2013