

# UHF RFID 标签分区说明和锁定介绍

以下为通用超高频标签的说明，超高频标签实际就是一个小的存储空间，模块只是通过特殊命令来读取标签中的数据，所以可以读写多少长度数据，由标签本身决定，具体可询问标签供应商

注意:以下所有的读写操作都是以十六进制 Q 形式进行的，数据长度必须以字(word)为单位(2byte，即 4 位，如 0001)

## 1.分区

标签存储器分为哪几个区？

分为 Reserved(保留)， EPC(电子产品代码)， TID(标签识别号)和 User(用户)四个独立的存储区块(Bank)。

- RESERVER 区:8byte(4 字)大小，前 4byte(2 字)为推毁密码(用于推毁标签，一般用不到)，后 4byte(2 字)为访问密码(用于进行写数据和锁定操作)，默认值为:00000000(毁密码)0000 0000(访问密码)。

- TID 区:12byte(6 字)大小，“只可读，不可写”，出厂已经写入，为标签的唯一标识符，电子标签的产品类识别号，每个生产厂商的 TID 号都会不同。

- EPC 区为识别标签对象的电子产品码，EPC 与 TID 相比，区别在于 EPC 可以用户手动写入修改。EPC 区的数据结构是，第一个 word 是校验位，第二个 word 是长度控制，之后的才是数据存储位置，盘存的时候，是根据长度控制位所表示的长度来显示 EPC 的。实际写入 EPC 数据的时候，起始地址填 1，即从长度控制位开始写入数据，校验位会自动计算不用手动写入。长度控制位计算: EPC 数据长度\*2=i将 i 换为十六进制表示，然后在转换后的结果后面加 00，最终得到的就是长度控制位的数据。如:0001，i=4\*2-8->08->0800，长度控制位为 0800，程序中计算:

```
1 |     byte[] newEPCByte = Tools.HexString2Bytes(newEPC);
2 |     byte[] pcByte = new byte[] { 0x00, 0x00 };
3 |     pcByte[0] = (byte) (newEPCByte.length * 4);
4 |     String pc = Tools.Bytes2HexString(pcByte, 2));
```

长度根据实际需要写入的数据长度填，比如:0001，长度为 1word，4 位(需要注意的是，数据位长度必须为 4 的倍数)，加上长度控制位 0800，软件中长度填写 2;例:0001 起始地址:1，长度:2，数据:08000001

- USER 区:该存储器的长度由各个电子标签的生产厂商确定，USER 区可以写入用户自定义的数据(以 16 进制形式)

## 2. 标签锁定

关于标签锁定

锁定状态分为四种，分两种情况分别进行说明:

1)如果没有修改访问密码，为默认的 00000000 情况:

- 未锁定:使用密码 00000000 可读、可写

- 暂时锁定:使用密码 00000000 可读、可写

- 永久锁定:使用密码 00000000 可读, 不可写

- 解锁定:将暂时锁定的区域解锁, 变成未锁定状态;永久锁定的区域无法被解锁

2)如果修改了访问密码, 不为默认的 00000000 情况, 假设修改为 000000FF

- 未锁定:使用密码 00000000 或者 000000FF 可读、可写

- 暂时锁定:RESERVER 区暂时锁定后, 使用密码 000000FF 可读、可写;其他区域(TID 除外)暂时

- 锁定后, 使用密码 00000000 或 000000FF 可读, 使用密码 000000FF 可写

- 永久锁定:RESERVER 区永久锁定后, 使用密码 000000FF 不可读, 不可写其他区域(TID 除外)

永久锁定后使用密码 00000000 或 000000FF 不可读, 不可写。

- 解锁定:将暂时锁定的区域解锁, 变成未锁定状态;永久锁定的区域无法被解锁

**注意:如要进行密码区之外的区域的锁定操作, 需要先锁定两个密码区, 才能锁定其他区域**