RFID EPC Class1 Gen2

符合EPC Class1 Gen2(简称G2)协议V109版的电子标签(Tag)和读写器(Reader)应该具有下述的特性

标签存储器分区

Tag memory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Bank(存储区块)

- Reserved: 存储Kill Password(灭活口令)和Access Password(访问口令)
- EPC: 存储EPC号码等
- TID: 存储标签识别号码, 每个TID号码应该是唯一的
- User: 存储用户定义的数据

此外还有各区块的Lock(锁定)状态位等用到的也是存储性质的单元

标签的状态

收到连续波(CW)照射上电(Power-up)以后, 标签可处于Ready(准备), Arbitrate(裁断), Reply(回令), Acknowledged(应答), Open(公开), Secured(保护), Killed(灭活)七种状态之一.

1. Ready状态: 未被灭活的标签上电以后, 开始所处的状态, 准备响应命令
2. Arbitrate状态: 等待响应Query等命令
3. 响应Query后, 进入Reply状态, 进一步将响应ACK命令就可以发回EPC号码
4. 发回EPC号码后, 进入Acknowledged状态, 进一步可以响应Req_RN命令
5. Access Password不为0才可以进入Open状态, 在此进行读、写操作
6. 已知Access Password才可能进入Secured状态, 进行读、写、锁定等操作
7. 进入到Killed状态的标签将保持状态不变, 永远不会产生调制信号以激活射频场, 从而永久失效。被灭活的标签在所有环境中均应保持Killed状态, 上电即进入灭活状态, 灭活操作不可逆转.

要使标签进入某一状态一般需要适当次序的一组合法命令, 反过来各命令也只能当标签在适当的状态下才能有效, 标签响应命令后也会转到其他状态.

命令分类

从命令体系架构和扩展性角度, 分为Mandatory(必备的), Optional(可选的), Proprietary(专有的)和Custom(定制的)四类
从使用功能上看, 分为标签Select(选取), Inventory(盘点)和Access(存取)命令三类, 此外还为了以后命令扩展, 预留了长短不同的编码待用.

必备的(Mandatory)命令

符合G2协议的标签和读写器, 应该支持必备的命令有十一条

1. Select(选择)
2. Query(查询)
3. QueryAdjust(调节查询)
4. QueryRep(重复查询)
5. ACK(EPC答复)
6. NAK(转向裁断)
7. Req_RN(随机数请求)
8. Read(读)
9. Write(写)
10. Kill(灭活)
11. Lock(锁定)

可选的(Optional)命令

符合G2协议的标签和读写器, 可选的命令有三条

1. Access(访问)
2. BlockWrite(块写)
3. BlockErase(块擦除)

专有的(Proprietary)命令

专有的命令一般用于制造目的, 如标签内部测试等, 标签出厂后这样的命令应该永久失效

定制的(Custom)命令

可以是制造商自己定义而开放给用户使用的命令, 如Philips公司提供有: BlockLock(块锁定), ChangeEAS(改EAS状态), EASAlarm(EAS报警)等命令(EAS是商品电子防盗窃系统Electronic Article Surveillance的缩写).

从功能角度: 选取(Select)类命令

仅有一条：Select, 是必备的. 标签有多种属性, 基于用户设定的标准和策略, 使用Select命令, 改变某些属性和标志人为选择或圈定了一个特定的标签群, 可以只对它们进行盘点识别或存取操作, 这样有利于减少冲突和重复识别, 加快识别速度.

从功能角度: 盘点(Inventory)类命令

有五条：Query, QueryAdjust, QueryRep, ACK, NAK, 都是必备的

1. 标签收到有效Query命令后, 符合设定标准被选择的每个标签产生一个随机数(类似掷骰子), 而随机数为零的每个标签, 都将产生回响(发回临时口令RN16, 一个16-bit随机数), 并转移到Reply状态; 符合另一些条件的标签会改变某些属性和标志, 从而退出上述标签群, 有利于减少重复识别.
2. 标签收到有效QueryAdjust命令后, 各标签分别新产生一个随机数(象重掷骰子), 其他同Query
3. 标签收到有效QueryRep命令后, 只对标签群中的每个标签原有的随机数减一, 其他同Query
4. 仅单一化的标签才能收到有效ACK命令(使用上述RN16, 或句柄Handle, 一个临时代表标签身份的16-bit随机数, 此为一种安全机制), 收到后发回EPC区中的内容, EPC协议最基本的功能.
5. 标签收到有效NAK命令后, 除了处于Ready、 Killed的保持原状态外, 其它情况都转到Arbitrate状态

从功能角度: 存取(Access)类命令

有五条必备的: Req_RN, Read, Write, Kill, Lock, 和三条可选的: Access, BlockWrite, BlockErase

1. 标签收到有效Req_RN(with RN16 or Handle)命令后, 发回句柄, 或新的RN16, 视状态而不同。
2. 标签收到有效Read(with Handle)命令后, 发回出错类型代码,或所要求区块的内容和句柄。
3. 标签收到有效Write(with RN16 & Handle)命令后, 发回出错类型代码,或写成功就发回句柄。
4. 标签收到有效Kill(with Kill Password, RN16 & Handle)命令后, 发回出错类型代码,或灭活成功就发回句柄。
5. 标签收到有效Lock(with Handle)命令后, 发回出错类型代码,或锁定成功就发回句柄。
6. 标签收到有效Access(with Access Password, RN16 & Handle)命令后, 发回句柄。
7. 标签收到有效BlockWrite(with Handle)命令后, 发回出错类型代码,或块写成功就发回句柄。
8. 标签收到有效BlockErase(with Handle)命令后, 发回出错类型代码,或块擦除成功就发回句柄。

G2用什么机制避免冲突

上述解答中提到, 当不止一个随机数为零的标签各发回不同的RN16时, 它们在接收天线上会出现不同RN16的波形迭加, 也即所谓冲突(collisions), 从而不能正确解码。有多种抗冲突机制可以避免波形迭加变形, 例如设法(时分)使某时刻只有一个标签“发言”, 接着再单一化处理, 就能识别读写多张标签中的每一张标签.
上述三条Q字头的命令体现了G2的抗冲突机制:

1. 随机数为零的标签才能发回RN16
2. 若同时有多个标签随机数为零, 而不能正确解码, 就策略性地重发Q字头的命令或组合给被选择的标签群, 直到能正确解码

标签识别号(TID)唯一性如何达成

标签识别号TID(Tag identifier)是标签之间身份区别的标志(可以类比于钞票的编号). 从安全和防伪角度考虑, 任何两张G2标签不应该完全相同, 标签应该具有唯一性. 标签四个存储区块各有用处, 出厂后有的还能随时改写, 只有

TID应该也可以担当此任, 所以标签的TID应该具有唯一性.

出厂前G2芯片的生产厂家应使用Lock命令或其他手段作用于TID, 使之永久锁定, 并且生产厂家或有关组织应该保证每个G2芯片适当长度的TID是唯一的, 任何情况下不会有第二个同样的TID, 即使某G2标签处于Killed状态不会被激活再使用, 它的TID(仍在此标签中)也不会出现在另一张G2标签中.

这样由于TID是唯一的, 虽然标签上的EPC码等可以被复制到另一张标签上去, 也能通过标签上的TID加以区分, 从而正本清源. 此种架构和方法简单可行, 但要注意保证唯一性的逻辑链.

V109版的G2协议对TID的规定, 必须的仅有32-bit(包括8-bit allocation class identifier, 12-bit tag mask-designer identifier, 12-bit tag model number), 对更多位的bit, 如SNR(serial number序列号)是 `Tags may contain` 而非 `should` . 但由于EPC号码被设计成会用到区分单件商品上, 32-bit大概是不够用的, 应该具有SNR.

G2协议中的灭活(Kill)命令

G2协议设置了Kill命令, 并且用32-bit的密码来控制, 有效使用Kill命令后标签永远不会产生调制信号以激活射频场, 从而永久失效. 但原来的数据可能还在标签中, 若想读取它们并非完全不可能, 可以考虑改善Kill命令的含义--附带擦除这些数据.

此外在一定时期内, 由于G2标签使用的成本或其他原因, 会考虑到兼顾标签能回收重复使用的情况(如用户要周转使用带标签的托盘、箱子, 内容物更换后相应的EPC号码、User区内容要改写; 更换或重新贴装标签所费不菲、不方便等等), 需要即使被永久锁定了的标签内容也能被改写的命令, 因为不同锁定状态的影响, 仅用Write或BlockWrite, BlockErase命令, 不一定能改写EPC号码、User内容或者Password(如标签的EPC号码被锁定从而不能被改写, 或未被锁定但忘了这个标签的Access Password而不能去改写EPC号码). 这样就产生了一个需求, 需要一个简单明了的Erase命令--除了TID区及其Lock状态位(标签出厂后TID不能被改写), 其他EPC号码、Reserved区、User区的内容和其它的Lock状态位, 即使是永久锁定了的, 也将全部被擦除以备重写.

比较起来, 改善的Kill命令和增加的Erase命令功能基本相同(包括应该都使用Kill Password), 区别仅在于前者Kill命令使不产生调制信号, 这样也可以统一归到由Kill命令所带参数RFU的不同值来考虑.