

RFID EPC Gen2 技术问答(本文由 jellycici 搜集整理)

符合 EPC Class1 Gen2 (简称 G2) 协议 V109 版的电子标签 (Tag, 简称标签) 和 Reader (读写器), 应该具有下述的特性。下文中的斜体字如 Query 为具体的命令名称。

Q: 标签存储器分为哪几个区?

A: Tag memory (标签内存) 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)。

Reserved 区: 存储 Kill Password (灭活口令) 和 Access Password (访问口令)。

EPC 区: 存储 EPC 号码等。

TID 区: 存储标签识别号码, 每个 TID 号码应该是唯一的。

User 区: 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。

Q: 标签有哪几种状态?

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

Ready 状态是未被灭活的标签上电以后, 开始所处的状态, 准备响应命令。

在 Arbitrate 状态, 主要为等待响应 Query 等命令。

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。

发回 EPC 号码后, 进入 Acknowledged 状态, 进一步可以响应 Req RN 命令。

Access Password 不为 0 才可以进入 Open 状态, 在此进行读、写操作。

已知 Access Password 才可能进入 Secured 状态, 进行读、写、锁定等操作。

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。

Q: 命令分为哪几类?

A: 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。

Q: 必备的 (Mandatory) 命令有哪些?

A: 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: Select (选择), Query (查询), QueryAdjust (调节查询), QueryRep (重复查询), ACK (EPC 答复), NAK (转

向裁断), Req_RN (随机数请求), Read (读), Write (写), Kill (灭活), Lock (锁定)。

Q: 可选的 (Optional) 命令有哪些?

A: 符合 G2 协议的标签和读写器, 支持也可以不支持可选的命令有三条: Access (访问), BlockWrite (块写), BlockErase (块擦除)。

Q: 定制的 (Custom) 命令会有哪些?

A: 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)。

Q: 选取 (Select) 类命令有哪些?

A: 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就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。

Q: 存取 (Access) 类命令有哪些?

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

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。

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。

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。

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。

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。

标签收到有效 Access (with Access Password, RN16 & Handle) 命令后, 发回句柄。

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。

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。

Q: G2 中访问 (Access) 等命令是可选的, 若标签或读写器不支持可选的命令怎么办?

A: 若不支持 BlockWrite 或 BlockErase 命令, 完全可以由 Write 命令 (一次写 16-bit) 多使用几次代替, 因为擦除可以认为是写 0, 前者块写、块擦除的块是几倍的 16-bit, 其他使用条件类似。

若不支持 Access 命令, 只有 Access Password 为 0, 才可进入 Secured 状态, 才能使用 Lock 命令。

在Open或Secured状态里都可以改变Access Password，之后再使用Lock命令锁定或永久锁定Access Password的话（pwd-read/write位为1，permalock位为0或1，参考附表），则标签再也进不了Secured状态了，也再不能使用Lock命令去改变任何锁定状态了。若支持Access命令，才可能使用相应的命令自由进入全部各种状态，除了标签被永久锁定或永久不锁而拒绝执行某些命令和处于Killed状态以外，也多能有效执行各个命令。G2协议规定的Access命令属于Optional可选的，但日后若能让Access命令成为必备的或者厂商生产对G2标签和读写器都支持Access命令的话，则控制和使用起来将比较灵活和全面。

Q: G2 协议中的灭活（Kill）命令效果怎么样？能否重新使用已灭活的标签？

A: 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。果然如此的话，人们应该可以彻底放心了。

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。

Q: 所谓冲突（collisions）是怎么回事，怎样抗冲突？G2 用什么机制抗冲突的？

A: 按上述 Q9 解答中提到的，当有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：随机数为零的标签才能发回 RN16，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。

Q: 概率/分槽防冲突算法

A: 在 RFID 多电子标签识别环境中,标签间冲突是影响 RFID 系统标签阅读速度的一个重要因素。Gen-2 标准采用了基于概率/分槽防冲突算法。该防冲突算法的实现与标签 ID 内容无关。

在阅读器开始进行一轮阅读操作时,其阅读标签命令里有一个参数 Q(Q 取值范围为 1~15),

该参数控制标签往各自的分槽计算器内载入一个随机数(取值范围 $0 \sim 2Q-1$)。当标签接收到阅读器相关命令时,分槽计算器值减 1,仅当标签内分槽计数器值为 0 时,标签才对阅读器进行应答;当分槽计数器值不为 0 时,标签不对阅读器进行应答,而是根据阅读器的不同命令,执行分槽计数器值继续减 1 操作,或者根据新的 Q 参数值来再次载入另一随机数(该随机数取值范围必须同样在 $0 \sim 2Q-1$)。已经阅读成功的标签,退出这轮标签阅读。当有二个或者多个标签的分槽计算器值同时为 0 时,这些标签会同时对阅读器进行应答,从而造成冲突。阅读器检测到冲突发生后,发出相关命令,让冲突标签的分槽计数器值从 0 变到 $0xFFFF$ (16 位二进制最大值),继续留在这轮阅读周期内,以后阅读器再通过设置新的 Q 参数来散列发生冲突的标签。这个阅读过程一直继续下去,直到完成这轮阅读周期。在阅读器命令参数 Q 的选择上,Gen-2 推荐了图 6 的算法。图中 Q_{fp} 是参数 Q 的浮点表示,阅读器对 Q_{fp} 取整得到 Q ,标签用 Q 作参数,在 $(0 \sim 2Q-1)$ 取值范围内,随机散列分槽计数器值,以实现标签的高效率读取。图中 $0.1 < C < 0.5$,且 Q 较大时, C 取较小值;而 Q 较小时, C 取较大值。

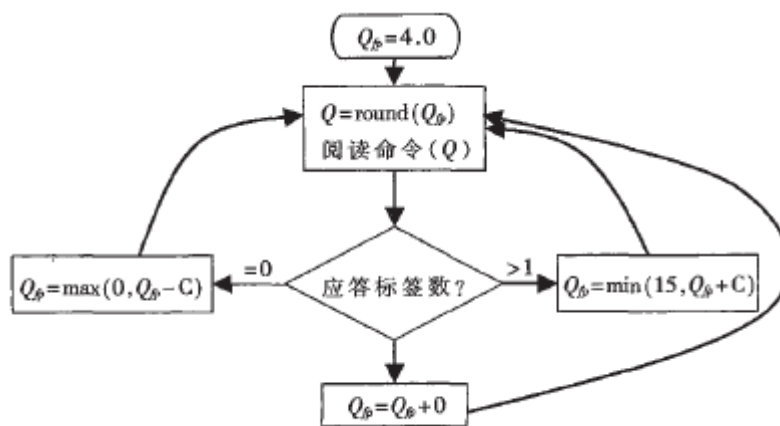


图 6 参数 Q 选择算法