

# 课程实验

课程名称 : RFID 原理与应用  
实验名称 : ISO/IEC 14443A 防碰撞协议  
学号 : 21281280  
姓名 : 柯劲帆  
班级 : 物联网2101班  
指导老师 : 赵帅锋  
日期 : 2024年6月9日

## 1. 实验环境准备

### 2. 实验原理

- 2.1. ISO 14443A 防碰撞算法
- 2.2. ISO 14443A UID
- 2.3. ISO 14443A 命令
  - 2.3.1. REQA 命令
  - 2.3.2. WUPA 命令
  - 2.3.3. ATQA 应答
  - 2.3.4. ANTICOLLISION 命令
  - 2.3.5. SELECT 命令
  - 2.3.6. SAK应答
  - 2.3.7. HALT命令

### 3. 实验过程

- 3.1. PICC ID 为 4 字节
- 3.2. PICC ID 为 7 字节
- 3.3. PICC ID 为 10 字节

# 1. 实验环境准备

- 实验 PC : Windows 11 笔记本
- 实验软件: pcdSim、piccSim、Microsoft Visual Studio 2022

# 2. 实验原理

## 2.1. ISO 14443A 防碰撞算法

算法工作流程如下：

- 卡片呼叫**: 当两张以上卡片进入射频场时，读写器发出卡呼叫命令，所有卡片同时回答“有卡片”。
- 卡号回送**: 读写器发送防冲突命令，要求所有卡片回送卡号。
- 检测冲突**: 读写器接收卡号。若卡号前几位相同，检测到某一位（如第五位）有冲突（部分卡片为“0”，部分为“1”）。
- 消除冲突**: 读写器指示卡号前四位（e.g. 为“1010”）、第五位（e.g. 为“1”）的卡片继续发言，其他卡片停止发言。
- 重复过程**:
  - 若与指定第五位相符的卡片不止一张，继续检测冲突。

- 重复步骤4，让冲突位为指定位的卡片继续发言，其他卡片停止发言。
6. **卡片选定**: 经过多次防冲突循环，当只剩一张卡片时，无冲突发生。该卡片完整回送卡号，读写器发出卡选择命令，这张卡片被选中。
7. **优先级设置**: 在防冲突过程中，读写器可选择冲突位为“1”或“0”的卡片胜出。

这个算法将 PCD 射频域内的最高  $x$  位、 $n$  张卡 ID 视为深度为  $x$ 、有  $n$  个叶子节点的二叉树，采用递归（类似先序遍历）的方式将叶子节点（即卡 ID）按顺序访问。

这个过程确保了在多卡环境下，逐步消除冲突，最终唯一选定一张卡片进行后续通信。

## 2.2. ISO 14443A UID

UID 的标准长度为 4 字节、7 字节或 10 字节，分别对应于单级联、双级联和三级联的 UID 结构。

以下是各个长度的 UID 存储结构：

### 4 字节 UID (单级联)

行为方	发送内容				
PCD	SEL = '93'				
PICC	uid0	uid1	uid2	uid3	BCC

### 7 字节 UID (双级联)

行为方	发送内容				
PCD	SEL = '93'				
PICC	CT = '88'	uid0	uid1	uid2	BCC
PCD	SEL = '95'				
PICC	uid3	uid4	uid5	uid6	BCC

### 10 字节 UID (三级联)

行为方	发送内容				
PCD	SEL = '93'				
PICC	CT = '88'	uid0	uid1	uid2	BCC
PCD	SEL = '95'				
PICC	CT = '88'	uid3	uid4	uid5	BCC
PCD	SEL = '97'				
PICC	uid6	uid7	uid8	uid9	BCC

在 ANTICOLLISION 和 SELECT 命令中，第一个字节就是在指定 SEL，PICC会根据 SEL 回复指定部分的 UID。

## 2.3. ISO 14443A 命令

### 2.3.1. REQA 命令

REQA (Request for Answer to Request Type A) 命令用于激活在读写器射频场中的卡片，使其进入通信状态。作为防冲突过程的起始命令，确保所有符合 ISO 14443A 协议的卡片都能被读写器检测到。

REQA 命令发送后，在射频场中的所有符合 ISO 14443A 协议的卡片都会以 ATQA (Answer to Request Type A) 响应。读写器收到 ATQA 响应后，知道有卡片存在，并可以继续进行防冲突和通信过程。

REQA 命令的帧结构为：

SOF	0x26	EOF
	0110010	

交互流程：

- 发送 REQA 命令**：读写器发送 REQA 命令；
- 卡片响应**：卡片收到 REQA 命令并返回 ATQA 响应；
- 防冲突和选卡**：读写器根据 ATQA 响应的信息，决定下一步的防冲突处理或直接进行数据通信。

### 2.3.2. WUPA 命令

WUPA (Wake-Up Type A) 命令用于唤醒在读写器射频场中可能处于低功耗或休眠状态的卡片，使其进入通信状态。WUPA 命令类似于 REQA 命令，但主要用于唤醒已经进入低功耗模式的卡片。

WUPA 命令发送后，在射频场中的所有符合 ISO 14443A 协议的卡片都会以 ATQA (Answer to Request Type A) 响应。读写器收到 ATQA 响应后，知道有卡片存在，并可以继续进行防冲突和通信过程。

WUPA 命令的帧结构为：

SOF	0x52	EOF
	01010010	

交互流程：

- 发送 WUPA 命令**：读写器向射频场发送 WUPA 命令，以唤醒可能处于低功耗状态的卡片。
- 卡片响应**：所有符合 ISO 14443A 协议并在射频场内的卡片返回 ATQA 响应，表明它们已被唤醒并准备进入通信状态。
- 防冲突和选卡**：读写器根据收到的 ATQA 响应，继续进行防冲突处理和选卡操作，以唯一识别和通信。

### 2.3.3. ATQA 应答

ATQA (Answer to Request Type A) 是卡片对 REQA 或 WUPA 命令的响应，用于指示其存在并提供一些基本的配置信息。通过 ATQA 应答，读写器可以确认在射频场中的卡片数量和特性，从而决定接下来的防冲突和选卡步骤。

#### ATQA 应答的帧结构

ATQA 应答由两个字节（16位）组成，其结构如下：

b16-b13	b12-b9	b8-b7	b6	b5-b1
RFU	经营者代码/RFU	UID 长度	RFU	比特帧防冲突方式

位字段说明：

- **b16-b13**: RFU，通常设置为 0000。
- **b12-b9**: 经营者代码或保留位 (RFU)，可以用于特定运营者的编码或保留使用。
- **b8-b7**: UID 长度 (UID Size)，表示 UID 的长度级别。
  - 00 : 表示 UID 长度为 4 字节 (32 位)
  - 01 : 表示 UID 长度为 7 字节 (56 位)
  - 10 : 表示 UID 长度为 10 字节 (80 位)
  - 11 : RFU
- **b6**: RFU，通常设置为 0。
- **b5-b1**: 比特帧防冲突方式，任何一位为 1 表示采用比特帧防冲突方式。

#### ATQA 应答流程

1. **发送 REQA 或 WUPA 命令**：读写器向射频场中的卡片发送 REQA 或 WUPA 命令。
2. **卡片响应**：在射频场中的所有符合 ISO 14443A 协议的卡片返回 ATQA 应答，提供基本配置信息。
3. **读写器处理**：读写器接收并解析 ATQA 应答，确认卡片存在并了解其基本特性，以决定接下来的防冲突和选卡步骤。

#### 2.3.4. ANTICOLLISION 命令

ANTICOLLISION 命令用于解决射频场中多张卡片同时响应的问题，通过逐步发送卡片的 UID 部分，读写器可以唯一识别每张卡片。该命令在不同的级联级别 (Cascade Level) 使用不同的标识符 '93'，'95'，'97'，分别表示第一级、第二级和第三级的防冲突级别。

ANTICOLLISION 命令的帧结构如下：

SOF	SEL	NVB	EOF
	0x93 (Cascade Level 1) 或 0x95 (Cascade Level 2) 或 0x97 (Cascade Level 3)	0x20	

交互流程：

1. **发送 ANTICOLLISION 命令**：读写器向射频场中的卡片发送 ANTICOLLISION 命令，要求卡片发送其 UID 部分。
2. **卡片响应**：卡片根据级联级别，返回部分 UID 和 BCC。
3. **检测冲突**：读写器检测是否存在冲突。如果存在冲突，读写器通过选择特定位进行防冲突处理，直到只剩下一个唯一的 UID。

如果有冲突：

SOF	SEL (8 bits)	NVB (8 bits)	指定的 UID 比特	EOF
	0x93 或 0x95 或 0x97	PCD 指定的比特数		

### 交互流程：

- 发送 SELECT 命令**：读写器向射频场中的卡片发送 ANTI-COLLISION 命令，包含卡片的部分 UID。
- 卡片确认**：卡片收到命令后，若指定 UID 与自身相符，返回自身 UID 余下部分。
- 进一步通信**：选中的卡片现在可以接收读写器发送的其他命令，如读写操作命令。

### 2.3.5. SELECT 命令

SELECT 命令用于选择和激活在防冲突过程中唯一识别的卡片，使其进入选中状态，准备接受进一步的命令。SELECT 命令也分为不同的级联级别，使用与 ANTI-COLLISION 命令相同的标识符。

SELECT 命令的帧结构如下：

SOF	SEL (8 bits)	NVB (8 bits)	UID	BCC	EOF
	0x93 或 0x95 或 0x97	UID 比特长度 + 16 比特长度			

### 交互流程：

- 发送 SELECT 命令**：读写器向射频场中的卡片发送 SELECT 命令，包含完整 UID 和 BCC。
- 卡片确认**：卡片收到命令后，若指定 UID 与自身相符，将会回复 SAK。
- 进一步通信**：选中的卡片现在可以接收读写器发送的其他命令，如读写操作命令。

### 2.3.6. SAK 应答

SAK (Select Acknowledge) 应答是卡片 (PICC) 在接收到 SELECT 命令后返回给读写器 (PCD) 的应答码，用于确认卡片是否已被选中，并提供一些附加信息。

SAK 应答的帧结构如下：

SOF	b6	b5 - b4	b3	b2 - b0	CRC_A (16 bits)	EOF
	1：支持 ISO/IEC 14443-4；0：不支持。	RFU	1：表示 UID 不完整，继续防冲突；0：表示 UID 完整。	RFU		

### 交互流程：

- 发送 SELECT 命令**：读写器向射频场中的卡片发送 SELECT 命令，包含卡片的部分 UID 和 BCC。
- 卡片确认**：卡片收到命令后，返回 SAK 码，表示其已被选中并进入选中状态。
- 进一步通信**：选中的卡片现在可以接收读写器发送的其他命令，如读写操作命令。

### 2.3.7. HALT 命令

HALT (休眠) 命令用于使卡片 (PICC) 进入休眠状态，停止与读写器 (PCD) 的通信。这个命令用于结束当前卡片的会话，直到卡片被重新唤醒 (通过 WUPA 命令)。

HALT 命令的帧结构如下：

SOF	命令码第一字节	命令码第二字节	CRC_A (16 bits)	EOF
	0x50	0x00		

交互流程：

1. **发送 HALT 命令**：PCD 向射频场中的目标卡片发送 HALT 命令。
2. **卡片进入休眠状态**：接收到 HALT 命令的 PICC 将停止响应读写器的进一步命令，进入休眠状态。
3. **重新唤醒**：要重新与该卡片通信，PCD 需要发送 WUPA 命令将其唤醒。

## 3. 实验过程

IP 和端口保持默认不变。设置 PICC ID。

### 3.1. PICC ID 为 4 字节

PCDSim

PCD/PICC之间的通信用UDP仿真，PCD->PICC用UDP广播仿真；PICC->PCD用UDP单播仿真

PCD rx udp port  PCD rx udp port  退出

防冲突中，新增加比特先取值  0  1

PCD运行：单步 或者 连续运行 单步运行 连续运行 下一步 暂停

PCD 状态  POWER\_OFF  START  ANTI  SELECTTING  SELECTED

运行过程/防碰撞过程：

序号	方向	数据(16进制)	命令	解析	动作
0	tx	26	REQA	26, REQA	wait ATRQ
1	rx	00,01	ATQA	bit8:7=00, CL1	send ANTI
2	tx	9320	ANTI_CL=93	UID:void	wait ANTI
3	rx	801228219b	ANTI	UID=80122821, BCC=9b	send ANTI
4	tx	933280.....00	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=80_00	wait ANTI
5	tx	933280.....10	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=80_10	wait ANTI
6	rx	000100...28219b	ANTI	UID=000100...2821, BCC=9b	send ANTI
7	tx	93448012.....0000	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8012_0000	wait ANTI
8	tx	93448012.....1000	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8012_1000	wait ANTI
9	rx	0010....219b	ANTI	UID=0010....21, BCC=9b	send ANTI
10	tx	9357801228_0100001	ANTI_CL=93	93:ANTI_CL1, 57:NVB,Byte num = 5, bit num = 7, UID=801228_0100001	wait ANTI
11	rx	0.....9b	ANTI	UID=0.....9b	send ANTI
12	tx	9370801228219b	SELECT_CL=93	93:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=80122821, BCC=9b	wait SAK
13	rx	00	SAK	00000000, b3=0, UID OK	uid=21.28.12.80.
14	tx	5000	HALT	5000, HALT	
15	tx	26	REQA	26, REQA	wait ATRQ

PICCSim

PCD/PICC之间的通信用UDP仿真，PCD->PICC用UDP广播仿真；PICC->PCD用UDP单播仿真

PICC rx port  PCD Ip addr  PCD rx port  结束

请以十六进制格式输入4/7/10字节的PICC ID：

按压[开始]，表示PICC进入PCD射频场，耦合电源成功，开始工作； 开始

按压[停止]，表示PICC离开PCD射频场，停止工作； 停止

PICC状态  POWER\_OFF  IDLE  READY  ACTIVE  HALT

通信过程/防碰撞过程：

序号	方向	数据(16进制)	命令	解析	动作
0	rx	26	REQA	26, REQA	Rx REQA. Tx ATQA.
1	tx	00,01	ATQA	bit8:7=00, CL1	Rx ANTI, Send ANTI.
2	rx	9320	ANTI_CL=93	UID:void	Rx ANTI, not my UID. do nothing.
3	tx	801228219b	ANTI	UID=80122821, BCC=9b	Rx ANTI, not my UID. do nothing.
4	rx	933280.....00	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=80_00	Rx ANTI, not my UID. do nothing.
5	rx	933280.....10	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=80_10	Rx ANTI, not my UID. do nothing.
6	tx	000100...28219b	ANTI	UID=000100...2821, BCC=9b	Rx ANTI, not my UID. do nothing.
7	rx	93448012.....0000	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8012_0000	Rx ANTI, not my UID. do nothing.
8	rx	93448012.....1000	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8012_1000	Rx ANTI, not my UID. do nothing.
9	tx	0010....219b	ANTI	UID=0010....21, BCC=9b	Rx ANTI, not my UID. do nothing.
10	rx	9357801228_0100001	ANTI_CL=93	93:ANTI_CL1, 57:NVB,Byte num = 5, bit num = 7, UID=801228_0100001	Rx ANTI, not my UID. do nothing.
11	tx	0.....9b	ANTI	UID=0.....9b	Rx ANTI, not my UID. do nothing.
12	rx	9370801228219b567c	SELECT_CL=93	93:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=80122821, BCC=9b	Rx SELECT, Tx SAK
13	tx	00	SAK	00000000, b3=0, UID OK	
14	rx	5000	HALT	5000, HALT	Rx HALT. Tx nothing.

过程分析：

1. **PCD**：发送 REQA，激活在读写器射频场中的卡片；

2. **PICC**：回复 ATQA，bit 8 到 bit 7 为 00，说明 UID 长度为 4 字节；bit 5 到 bit 1 含有 1，说明采用比特帧防冲突方式；
3. **PCD**：发送 ANTICOLLISION，指定查询第 1 级 UID；
4. **PICC**：回复 ANTICOLLISION，回复自己的 UID = 0x80122821 和 BCC = 0x9b；
5. **PCD**：发现收到的 UID 中，第 2 字节的第 2 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 32，即会指定 UID 的前 1 字节和第 2 字节的前 2 比特；指定 UID 前 1 字节为 80，第 2 字节的前 2 比特为 00；
  2. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 1，与收到的 ANTICOLLISION 命令不匹配，不回复；
  3. **PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 32，即会指定 UID 的前 1 字节和第 2 字节的前 2 比特；指定 UID 前 1 字节为 80，第 2 字节的前 2 比特为 10；
  4. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 1，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 000100，剩下字节为 2821；
6. **PCD**：发现收到的 UID 中，第 3 字节的第 4 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 44，即会指定 UID 的前 2 字节和第 3 字节的前 4 比特；指定 UID 前 2 字节为 8012，第 3 字节的前 4 比特为 0000；
  2. **PICC**：发现自身 UID 第 3 字节的第 4 比特为 1，与收到的 ANTICOLLISION 命令不匹配，不回复；
  3. **PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 32，即会指定 UID 的前 2 字节和第 3 字节的前 4 比特；指定 UID 前 3 字节为 8012，第 3 字节的前 4 比特为 1000；
  4. **PICC**：发现自身 UID 第 3 字节的第 4 比特为 1，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 3 字节的剩下比特为 0010，剩下字节为 21；
7. **PCD**：发现收到的 UID 中，第 4 字节的第 7 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 57，即会指定 UID 的前 3 字节和第 4 字节的前 7 比特；指定 UID 前 3 字节为 801228，第 4 字节的前 7 比特为 0100001；
  2. **PICC**：发现自身 UID 第 4 字节的第 7 比特为 0，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 4 字节的剩下比特为 0，无剩下字节；
  8. **PCD**：发送 SELECT 命令，指定 0x93 第 1 级 UID；指定 NVB = 70；指定 UID = 80122821，BCC = 9b；
  9. **PICC**：回复 SAK = 00，表示不支持 ISO/IEC 14443-4，UID 已完整；
  10. **PCD**：发送 HALT 命令，使得被选中的卡片进入睡眠状态。

## 3.2. PICC ID 为 7 字节

PCDSim

PCD/PICC之间的通信用UDP仿真，PCD->PICC用UDP广播仿真；PICC->PCD用UDP单播仿真

PCC rx udp port	10001	PCD rx udp port	10000
-----------------	-------	-----------------	-------

退出

防冲突中，新增加比特先取值  0  1

PCD运行：单步 或者 连续运行  单步运行  连续运行  下一步  暂停

运行过程/防碰撞过程：

序号	方向	数据(16进制)	命令	解析	动作
0	tx	26	REQA	26, REQA	wait ATRQ
1	rx	00,41	ATQA	bit8:7=01, CL2	send ANTI
2	tx	9320	ANTI_CL=93	UID=void	wait ANTI
3	rx	8880122832	ANTI	UID=88801228, BCC=32	send ANTI
4	tx	933288.....00	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=88_00	wait ANTI
5	rx	100000...122832	ANTI	UID=100000...1228, BCC=32	send ANTI
6	tx	93448880....0010	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8880_0010	wait ANTI
7	rx	0001....2832	ANTI	UID=0001....28, BCC=32	send ANTI
8	tx	9357888012_0101000	ANTI_CL=93	93:ANTI_CL1, 57:NVB,Byte num = 5, bit num = 7, UID=888012_0101000	wait ANTI
9	rx	0.....32	ANTI	UID=0.....32, BCC=32	send ANTI
10	tx	93708880122832	SELECT_CL=93	93:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=88801228, BCC=32	wait SAK
11	rx	04	SAK	00000100, b3=1, incompleted UID, continue anti	continue anti
12	tx	9520	ANTI_CL=95	UID=void	wait ANTI
13	rx	2144102055	ANTI	UID=21441020, BCC=55	send ANTI
14	tx	9525....00001	ANTI_CL=95	95:ANTI_CL2, 25:NVB,Byte num = 2, bit num = 5, UID=_00001	wait ANTI
15	rx	001....44102055	ANTI	UID=001....441020, BCC=55	send ANTI
16	tx	95702144102055	SELECT_CL=95	95:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=21441020, BCC=55	wait SAK
17	rx	00	SAK	00000000, b3=0, UID OK	uid=20.10.44.21.28.12.80.
18	tx	5000	HALT	5000, HALT	
19	tx	26	REQA	26, REQA	wait ATRQ

PICCSim

PCD/PICC之间的通信用UDP仿真，PCD->PICC用UDP广播仿真；PICC->PCD用UDP单播仿真

PCC rx port	10001	PCD Ip addr	127 . 0 . 0 . 1	PCD rx port	10000
-------------	-------	-------------	-----------------	-------------	-------

结束

请以十六进制格式输入4/7/10字节的PICC ID：

按下[开始]，表示PICC进入PCD射频场，耦合电源成功，开始工作；

按下[停止]，表示PICC离开PCD射频场，停止工作；

PICC状态  POWER\_OFF  IDLE  READY  ACTIVE  HALT

通信过程/防碰撞过程：

序号	方向	数据(16进制)	命令	解析	动作
0	rx	26	REQA	26, REQA	Rx REQA. Tx ATQA.
1	tx	00,41	ATQA	bit8:7=01, CL2	Rx ANTI, Send ANTI.
2	rx	9320	ANTI_CL=93	UID=void	
3	tx	8880122832	ANTI	UID=88801228, BCC=32	
4	rx	933288.....00	ANTI_CL=93	93:ANTI_CL1, 32:NVB,Byte num = 3, bit num = 2, UID=88_00	Rx ANTI, Send ANTI.
5	tx	100000...122832	ANTI	UID=100000...1228, BCC=32	
6	rx	93448880....0010	ANTI_CL=93	93:ANTI_CL1, 44:NVB,Byte num = 4, bit num = 4, UID=8880_0010	Rx ANTI, Send ANTI.
7	tx	0001....2832	ANTI	UID=0001....28, BCC=32	
8	rx	9357888012_0101000	ANTI_CL=93	93:ANTI_CL1, 57:NVB,Byte num = 5, bit num = 7, UID=888012_0101000	Rx ANTI, Send ANTI.
9	rx	0.....32	ANTI	UID=0.....32, BCC=32	
10	rx	93708880122832c8eb	SELECT_CL=93	93:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=88801228, BCC=32	Rx SELECT, Tx SAK
11	tx	04	SAK	00000100, b3=1, incompleted UID, continue anti	
12	rx	9520	ANTI_CL=95	UID=void	Rx ANTI, Send ANTI.
13	tx	2144102055	ANTI	UID=21441020, BCC=55	
14	rx	9525....00001	ANTI_CL=95	95:ANTI_CL2, 25:NVB,Byte num = 2, bit num = 5, UID=_00001	Rx ANTI, Send ANTI.
15	tx	001....44102055	ANTI	UID=001....441020, BCC=55	
16	rx	95702144102055512d	SELECT_CL=95	95:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=21441020, BCC=55	Rx SELECT, Tx SAK
17	tx	00	SAK	00000000, b3=0, UID OK	
18	rx	5000	HALT	5000, HALT	Rx HALT. Tx nothing.

### 过程分析：

- PCD**：发送 REQA，激活在读写器射频场中的卡片；
- PICC**：回复 ATQA，bit 8 到 bit 7 为 01，说明 UID 长度为 7 字节；bit 5 到 bit 1 含有 1，说明采用比特帧防冲突方式；
- PCD**：发送 ANTICOLLISION，指定查询第 1 级 UID；
- PICC**：回复 ANTICOLLISION，回复第 1 级 UID = 0x88801228 和 BCC = 0x32；
- PCD**：发现收到的 UID 中，第 2 字节的第 2 比特发生了碰撞。处理碰撞：
  - PCD**：发送 ANTICOLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 32，即会指定 UID 的前 1 字节和第 2 字节的前 2 比特；指定 UID 前 1 字节为 88，第 2 字节的前 2 比特为 00；
  - PICC**：发现自身 UID 第 2 字节的第 2 比特为 0，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 100000，剩下字节为 1228；

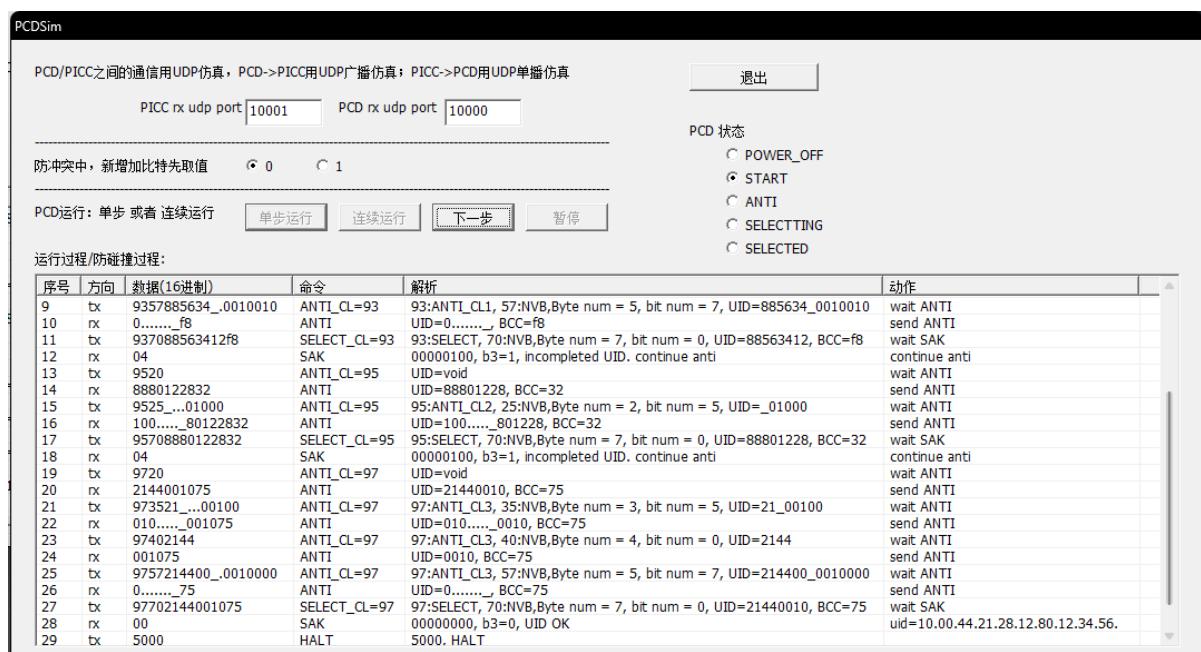
6. **PCD**：发现收到的 UID 中，第 3 字节的第 4 比特发生了碰撞。处理碰撞：

1. **PCD**：发送 ANTI COLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 44，即会指定 UID 的前 2 字节和第 3 字节的前 4 比特；指定 UID 前 2 字节为 8880，第 3 字节的前 4 比特为 0010；
2. **PICC**：发现自身 UID 第 3 字节的第 4 比特为 0，与收到的 ANTI COLLISION 命令匹配，回复剩下的 UID：第 3 字节的剩下比特为 0001，剩下字节为 28；

7. **PCD**：发现收到的 UID 中，第 4 字节的第 7 比特发生了碰撞。处理碰撞：

1. **PCD**：发送 ANTI COLLISION 命令，指定 0x93 第 1 级 UID；指定 NVB = 57，即会指定 UID 的前 3 字节和第 4 字节的前 7 比特；指定 UID 前 3 字节为 888012，第 4 字节的前 7 比特为 0101000；
2. **PICC**：发现自身 UID 第 4 字节的第 7 比特为 0，与收到的 ANTI COLLISION 命令匹配，回复剩下的 UID：第 4 字节的剩下比特为 0，无剩下字节；
8. **PCD**：发送 SELECT 命令，指定 0x93 第 1 级 UID；指定 NVB = 70；指定 UID = 88801228，BCC = 32；
9. **PICC**：回复 SAK = 04，表示不支持 ISO/IEC 14443-4，UID 未完整；
10. **PCD**：发送 ANTI COLLISION，指定查询第 2 级 UID；
11. **PICC**：回复 ANTI COLLISION，回复第 2 级 UID = 0x21441020 和 BCC = 0x55；
12. **PCD**：发现收到的 UID 中，第 1 字节的第 5 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTI COLLISION 命令，指定 0x95 第 2 级 UID；指定 NVB = 25，即会指定 UID 的第 1 字节的前 5 比特为 00001；
  2. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 0，与收到的 ANTI COLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 001，剩下字节为 441020；
13. **PCD**：发送 SELECT 命令，指定 0x95 第 2 级 UID；指定 NVB = 70；指定 UID = 21441020，BCC = 55；
14. **PICC**：回复 SAK = 00，表示不支持 ISO/IEC 14443-4，UID 已完整；
15. **PCD**：发送 HALT 命令，使得被选中的卡片进入睡眠状态。

### 3.3. PICC ID 为 10 字节



PCCSim

PCD/PICC之间的通信用UDP仿真，PCD->PICC用UDP广播仿真；PICC->PCD用UDP单播仿真

PICC rx port [10001] PCD Ip addr [127.0.0.1] PCD rx port [10000]

请以十六进制格式输入4/7/10字节的PICC ID: [10004421281280123456]

按下[开始]，表示PICC进入PCD射频场，耦合电源成功，开始工作；

按下[停止]，表示PICC离开PCD射频场，停止工作；

PICC状态  
 POWER\_OFF  
 IDLE  
 READY  
 ACTIVE  
 HALT

通信过程/防碰撞过程

序号	方向	数据(16进制)	命令	解析	动作
12	tx	04	SAK	00000100, b3=1, incompleted UID. continue anti	
13	rx	9520	ANTI_CL=95	UID=void	Rx ANTI, Send ANTI.
14	tx	8880122832	ANTI	UID=88801228, BCC=32	
15	rx	9525...01000	ANTI_CL=95	95:ANTI_CL2, 25:NVB,Byte num = 2, bit num = 5, UID=_01000	Rx ANTI, Send ANTI.
16	tx	100...._80122832	ANTI	UID=100...._801228, BCC=32	
17	rx	9570888012283205b3	SELECT_CL=95	95:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=88801228, BCC=32	Rx SELECT, Tx SAK
18	tx	04	SAK	00000100, b3=1, incompleted UID. continue anti	
19	rx	9720	ANTI_CL=97	UID=void	Rx ANTI, Send ANTI.
20	tx	2144001075	ANTI	UID=21440010, BCC=75	
21	rx	973521...00100	ANTI_CL=97	97:ANTI_CL3, 35:NVB,Byte num = 3, bit num = 5, UID=21_00100	Rx ANTI, Send ANTI.
22	tx	010...._001075	ANTI	UID=010...._0010, BCC=75	
23	rx	97402144	ANTI_CL=97	97:ANTI_CL3, 40:NVB,Byte num = 4, bit num = 0, UID=2144	Rx ANTI, Send ANTI.
24	tx	001075	ANTI	UID=0010, BCC=75	
25	rx	9757214400_0010000	ANTI_CL=97	97:ANTI_CL3, 57:NVB,Byte num = 5, bit num = 7, UID=214400_0010000	Rx ANTI, Send ANTI.
26	tx	0....._75	ANTI	UID=0....._75, BCC=75	
27	rx	97702144001075df08	SELECT_CL=97	97:SELECT, 70:NVB,Byte num = 7, bit num = 0, UID=21440010, BCC=75	Rx SELECT, Tx SAK
28	tx	00	SAK	00000000, b3=0, UID OK	
29	rx	5000	HALT	5000, HALT	Rx HALT. Tx nothing.

### 过程分析：

1. **PCD**：发送 REQA，激活在读写器射频场中的卡片；
2. **PICC**：回复 ATQA，bit 8 到 bit 7 为 **10**，说明 UID 长度为 10 字节；bit 5 到 bit 1 含有 **1**，说明采用比特帧防冲突方式；
3. **PCD**：发送 ANTICOLLISION，指定查询第 1 级 UID；
4. **PICC**：回复 ANTICOLLISION，回复第 1 级 UID = **0x88563412** 和 BCC = **0xf8**；
5. **PCD**：发现收到的 UID 中，第 2 字节的第 2 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 **0x93** 第 1 级 UID；指定 NVB = **32**，即会指定 UID 的前 1 字节和第 2 字节的前 2 比特；指定 UID 前 1 字节为 **88**，第 2 字节的前 2 比特为 **00**；
  2. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 **1**，与收到的 ANTICOLLISION 命令不匹配，不回复；
  3. **PCD**：发送 ANTICOLLISION 命令，指定 **0x93** 第 1 级 UID；指定 NVB = **32**，即会指定 UID 的前 1 字节和第 2 字节的前 2 比特；指定 UID 前 1 字节为 **88**，第 2 字节的前 2 比特为 **10**；
  4. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 **0**，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 **0101001**，剩下字节为 **3412**；
6. **PCD**：发现收到的 UID 中，第 3 字节的第 4 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 **0x93** 第 1 级 UID；指定 NVB = **44**，即会指定 UID 的前 2 字节和第 3 字节的前 4 比特；指定 UID 前 2 字节为 **8856**，第 3 字节的前 4 比特为 **0100**；
  2. **PICC**：发现自身 UID 第 3 字节的第 4 比特为 **0**，与收到的 ANTICOLLISION 命令匹配，回复剩下的 UID：第 3 字节的剩下比特为 **0011**，剩下字节为 **12**；
7. **PCD**：发现收到的 UID 中，第 4 字节的第 7 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTICOLLISION 命令，指定 **0x93** 第 1 级 UID；指定 NVB = **57**，即会指定 UID 的前 3 字节和第 4 字节的前 7 比特；指定 UID 前 3 字节为 **885634**，第 4 字节的前 7 比特为 **0010010**；

2. **PICC**：发现自身 UID 第 4 字节的第 7 比特为 0，与收到的 ANTCOLLISION 命令匹配，回复剩下的 UID：第 4 字节的剩下比特为 0，无剩下字节；
8. **PCD**：发送 SELECT 命令，指定 0x93 第 1 级 UID；指定 NVB = 70；指定 UID = 88563412，BCC = f8；
9. **PICC**：回复 SAK = 04，表示不支持 ISO/IEC 14443-4，UID 未完整；
10. **PCD**：发送 ANTCOLLISION，指定查询第 2 级 UID；
11. **PICC**：回复 ANTCOLLISION，回复第 2 级 UID = 0x88801228 和 BCC = 0x32；
12. **PCD**：发现收到的 UID 中，第 2 字节的第 5 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTCOLLISION 命令，指定 0x95 第 2 级 UID；指定 NVB = 25，即会指定 UID 的第 1 字节的前 5 比特为 01000；
  2. **PICC**：发现自身 UID 第 2 字节的第 2 比特为 0，与收到的 ANTCOLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 100，剩下字节为 801228；
13. **PCD**：发送 SELECT 命令，指定 0x95 第 2 级 UID；指定 NVB = 70；指定 UID = 88801228，BCC = 32；
14. **PICC**：回复 SAK = 04，表示不支持 ISO/IEC 14443-4，UID 未完整；
15. **PCD**：发送 ANTCOLLISION，指定查询第 3 级 UID；
16. **PICC**：回复 ANTCOLLISION，回复第 3 级 UID = 0x21440010 和 BCC = 0x75；
17. **PCD**：发现收到的 UID 中，第 2 字节的第 5 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTCOLLISION 命令，指定 0x97 第 3 级 UID；指定 NVB = 35，即会指定 UID 的前 1 字节和第 2 字节的前 5 比特；指定 UID 前 1 字节为 21，第 2 字节的前 5 比特为 00100；
  2. **PICC**：发现自身 UID 第 2 字节的第 5 比特为 0，与收到的 ANTCOLLISION 命令匹配，回复剩下的 UID：第 2 字节的剩下比特为 010，剩下字节为 0010；
18. **PCD**：发现收到的 UID 中，第 2 字节的第 8 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTCOLLISION 命令，指定 0x97 第 3 级 UID；指定 NVB = 40，即会指定 UID 的前 3 字节为 2144；
  2. **PICC**：发现自身 UID 第 2 字节的第 8 比特为 0，与收到的 ANTCOLLISION 命令匹配，回复 UID 剩下字节为 0010；
19. **PCD**：发现收到的 UID 中，第 4 字节的第 7 比特发生了碰撞。处理碰撞：
  1. **PCD**：发送 ANTCOLLISION 命令，指定 0x97 第 3 级 UID；指定 NVB = 57，即会指定 UID 的前 3 字节和第 4 字节的前 7 比特；指定 UID 前 3 字节为 214400，第 4 字节的前 7 比特为 0010000；
  2. **PICC**：发现自身 UID 第 4 字节的第 7 比特为 0，与收到的 ANTCOLLISION 命令匹配，回复剩下的 UID：第 4 字节的剩下比特为 0，无剩下字节；
20. **PCD**：发送 SELECT 命令，指定 0x97 第 3 级 UID；指定 NVB = 70；指定 UID = 21440010，BCC = 75；
21. **PICC**：回复 SAK = 00，表示不支持 ISO/IEC 14443-4，UID 已完整；
22. **PCD**：发送 HALT 命令，使得被选中的卡片进入睡眠状态。