高频模块实验指导书

目录

第一章　ISO/ICE 14443A 协议介绍

1.1　非接触式系统

非接触系统基本组件包括非接触式读写器和应答器。非接触式读写器主要由射频电路板与连接在之上的天线组成。应答器包括一个感应天线和接在天线尾部的集成电路。读写器和应答器结合起来，功能类似于一个变压器。当交流电通过主线圈（读写器天线）后形成了电磁场，并在次线圈（应答器天线）生成感应电流。应答器将非接触读写器传播的电磁场通过一个二极管整流器转换成直流电，为应答器的内部电路供电。两个天线的配置和调整取决于两个装置之间的耦合效率。

对于非接触卡，由非接触读写器（PCD）并由非接触卡（PICC）接收的射频能量不仅用来激励非接触卡，而且用来通过载波调制传输数据。PICC 对 PCD 传输的数据进行解码和处理，然后通过加载调制反馈给 PCD。

图 1 是读写器与作为应答器的非接触式卡片应用的一个具体例子。

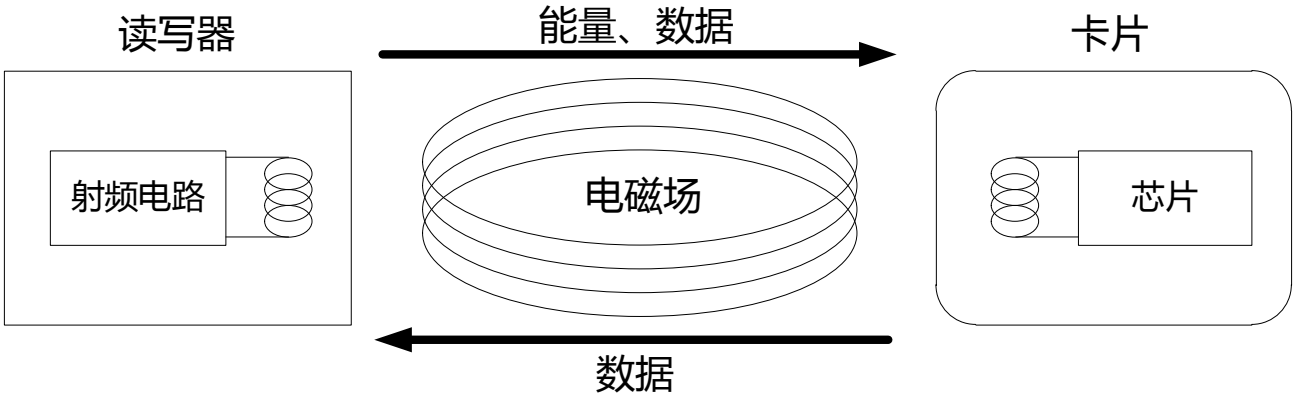

图 1 ARM7 试验箱硬件结构框图

1.2　ISO/ICE14443A 位编码

在一个数字通信系统中，数字数据要被转换成传输符号。例如，这些符号由脉冲序列组成。最简单的数据传输方式是转换发送装置的开关，开时发送 1，关时发送 0。这种编码方式被称为开/关键控（OOK）。

因为很难区分一个 0 位信号和发送装置开关确实被关，数据信号需要被编码。例如：NRZ-L 编码、曼

彻斯特编码、改进的米勒编码。

1.2.1 从 PCD 到 PICC 的通信

在 ISO14443A 中 PCD 使用 ASK100%（OOK）调制原理来产生一个低电平。ASK100%（OOK）调制方式如图 2 所示。

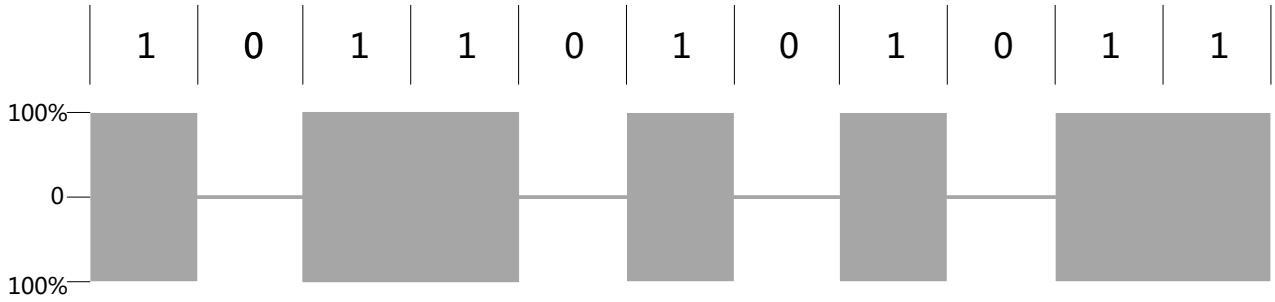

图 2 ASK100%（OOK）调制原理

ISO14443A 使用的编码方式为改进的米勒编码。改进的米勒编码定义了 3 种序列：

——在位周期的前半周期有一个“低电平”，为 X 序列，如图 3 所示。

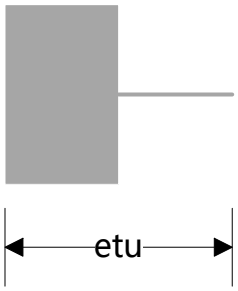

图 3 改进的米勒编码 X 序列

——在位周期的整个周期中没有“低电平”，为 Y 序列，如图 4 所示。

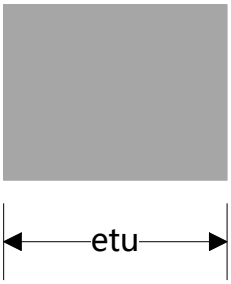

图 4 改进的米勒编码 Y 序列

——在位周期的后半周期有一个“低电平”，为 Z 序列，如图 5 所示。

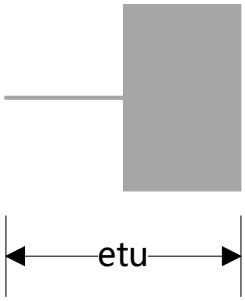

图 5 改进的米勒编码 Z 序列

这三种序列用来编码：逻辑 1，逻辑 0，通信起始，通信结束和没有信息。

——逻辑 1：序列 X

——逻辑 0：序列 Y，但是有两种情况除外，

如果有连续的两个或者更多的逻辑 0，则从第二个逻辑 0 开始用 Z 序列。

如果通信起始后第一个比特位是逻辑 0，则用 Z 序列来表示这个逻辑 0 以及紧跟其后的任何逻辑 0。

——通信起始：序列 Z

——通信结束：逻辑 0 后面紧跟序列 Y

——没有信息：至少两个序列 Y

图 6 为一个典型的改进的米勒编码的编码方式。

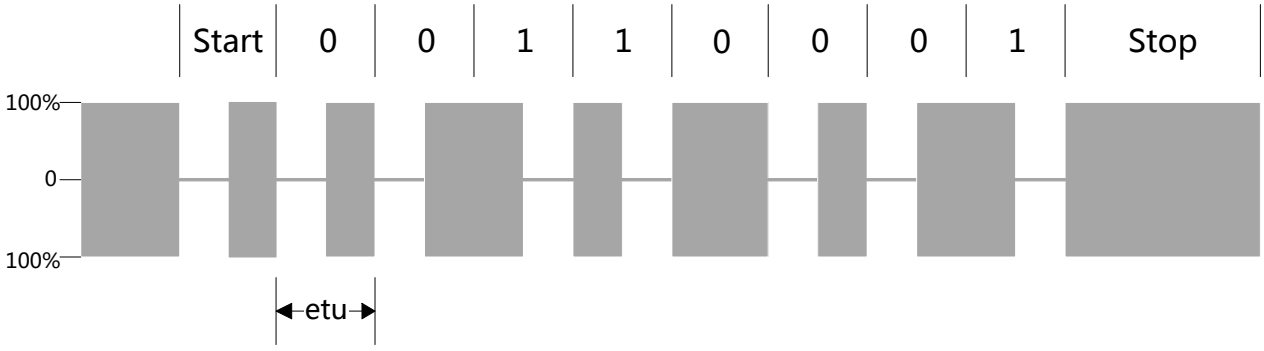

图 6 改进的米勒编码的编码方式

1.2.2 从 PICC 到 PCD 的通信

在 ISO14443A 中 PICC 使用 OOK 副载波调制的曼彻斯特编码方式进行位编码，曼彻斯特编码方式定义了三种序列：

——载波在前半个位持续时间被调制，在后半个位持续时间内不被调制，为 D 序列，如图 7 所示。

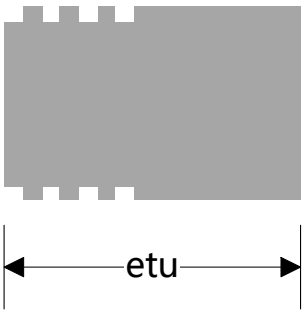

图 7 曼彻斯特编码 D 序列

——载波在前半个位持续时间内不被调制，在后半个位持续时间内被调制，为 E 序列，如图 8 所示。

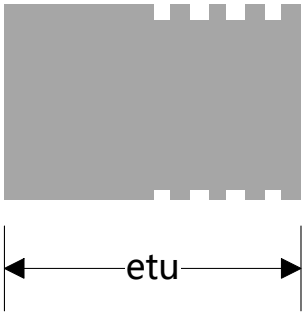

图 8 曼彻斯特编码 E 序列

——载波在整个位持续时间内都不被调制，为 F 序列，如图 9 所示。

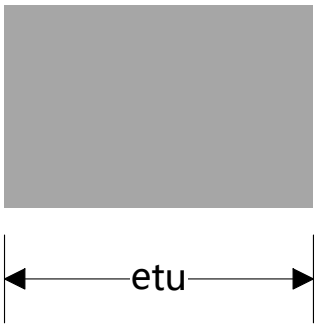

图 9 曼彻斯特编码 F 序列

这三种序列用来编码：逻辑 0，逻辑 1，通信起始，通信结束，没有信息。

——逻辑 1：序列 D

——逻辑 0：序列 E

——通信起始：序列 D

——通信结束：序列 F

——没有信息：无信息发送时，副载波不调制载波

图 10 为一个典型的曼彻斯特编码的编码方式。

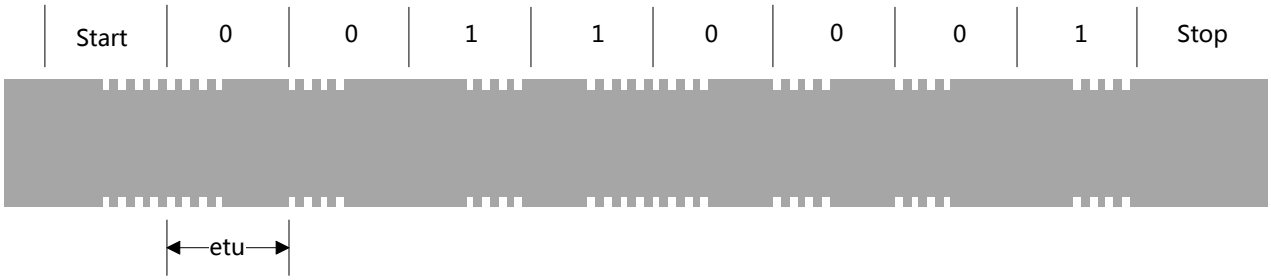

图 10 曼彻斯特编码的编码方式

1.3 ISO/ICE14443A 帧格式

ISO14443A 的帧是由所有数据位加一个通信起始（S），一个通信结束（E），并且在每 8 个数据位之后有一个奇偶校验位（P）组成，低位 LSB 先传。如图 11 所示。

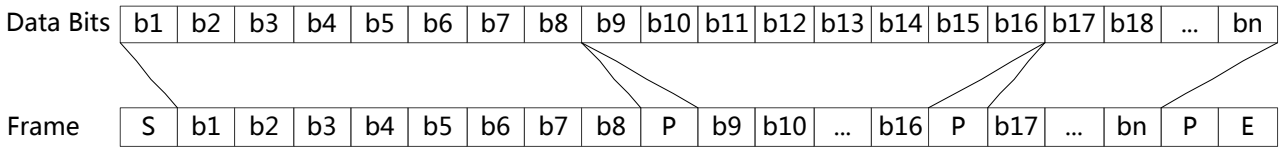

图 11 ISO14443A 帧格式

ISO14443A 使用两种帧：短帧和标准帧。短帧用于通信初始化（Wake-UP），标准帧用于数据交换。

1.3.1 短帧

短帧用于通信初始化，按一下次序组成，见图 12。

——通信开始（S）

——LSB 先传输的 7 个数据位

——通信结束（E）

注：不加奇偶校验位。

	LSB						MSB	
S	b1	b2	b3	b4	b5	b6	b7	E

图 12 ISO14443A 短帧格式

1.3.2 标准帧

标准用于数据交换，按一下次序组成，见图 13。

——通信开始（S）

——n*（8 个数据位+奇校验位），n>=1。每个字节必须接一个奇校验位。

——通信结束（E）

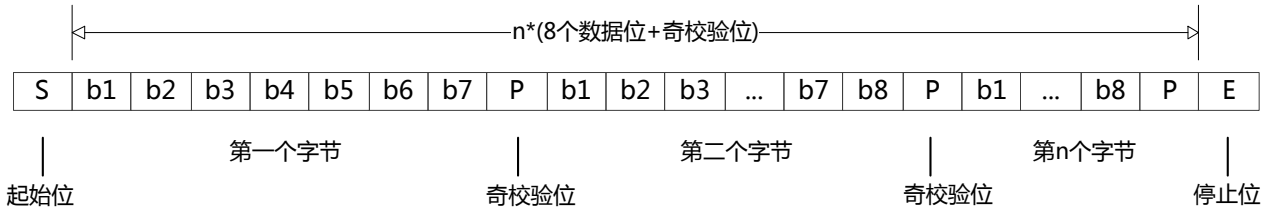

图 13 ISO14443A 标准帧格式

1.4 ISO/ICE14443A 命令与应答

1.4.1 ISO14443A 的 CRC_A

ISO14443A 的一些命令需要 CRC_A 校验字节。CRC_A 是用来对 K 个数据位的数据帧进行错误校验的，这 K 个数据位是由命令帧内除了 CRC_A 外的所有数据位组成。由于所有使用 CRC_A 的命令以字节编码，因此位数 K 位 8 的倍数。

图 14 说明了标准帧命令中 CRC_A 所处的位置。其中 CRC_A1 是最低有效字节，CRC_A2 是最高有效字节。

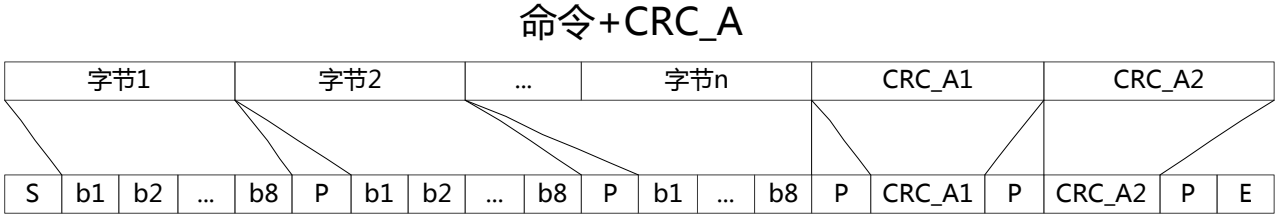

图 14 CRC_A 在标准帧命令中的位置

当标准帧命令中包含 CRC_A 时，CRC_A 应该在数据的最后一个奇校验位 P 和通信结束位 E 之间。每一个 CRC_A 字节应在末位包含一个奇校验位 P。短帧不需要 CRC_A。

1.4.2 ISO14443A 命令集

图 15 列出了的命令用于 PCD 与 PICC 之间的通信。

PCD命令	PICC应答
WUPA	ATQA
HLTA	—
ANTICOLLISION	UID
SELECT	SAK
RATS	ATS

图 15 ISO14443A 命令

1.4.3 WUPA

WUPA 命令用于 PCD 探测感应区域内的类型 A PICC。

1.4.3.1 WUPA

WUPA 命令采用短帧格式传输，命令的编码格式如图 16 所示。

b7	b6	b5	b4	b3	b2	b1	说明
1	0	1	0	0	1	0	’52‘=WUPA

图 16 WUPA 短帧编码

1.4.3.2 WUPA 应答（ATQA）

当 PCD 发出 WUPA 请求时，类型 A PICC 将根据其状态返回一个包含两字节的 ATQA。ATQA 采用标准帧格式传输但不包括 CRC_A 字节，其编码格式如图 17 和图 18 所示。

b8	b7	b6	b5	b4	b3	b2	b1	说明
0	0							UID长度：4字节
0	1							UID长度：7字节
1	0							UID长度：10字节
1	1							禁止
		0						保留
			1	0	0	0	0	比特帧防冲突
			0	1	0	0	0	比特帧防冲突
			0	0	1	0	0	比特帧防冲突
			0	0	0	1	0	比特帧防冲突
			0	0	0	0	1	比特帧防冲突

图 17 ATQA 的字节 1 编码

b8	b7	b6	b5	b4	b3	b2	b1	说明
0	0	0	0					保留
				X	X	X	X	任意值

图 18 ATQA 的字节 2 编码

1.4.4 ANTICOLLISION

ANTICOLLISION 命令用于获取一张类型 A PICC 的完整 UID，同时检测感应区内是否有多张类型 A PICC。

1.4.4.1 ANTICOLLISION 命令

ANTICOLLISION 命令采用不含 CRC_A 的标准帧格式传输，其编码格式如图 19 所示。

字节1	字节2
SEL	‘20’

图 19 ANTICOLLISION 命令的编码

SEL 编码格式如图 20 所示。

b8	b7	b6	b5	b4	b3	b2	b1	说明
1	0	0	1	0	0	1	1	防冲突串联级别1
1	0	0	1	0	1	0	1	防冲突串联级别2
1	0	0	1	0	1	1	1	防冲突串联级别3

图 20 SEL 的编码

ANTICOLLISION 命令的 SEL 字节定义了请求 UID 的哪个级别。

1.4.4.2 ANTICOLLISION 应答

当 PCD 发出 ANTICOLLISION 命令时，所有在感应区域内的 PICC 都会返回被请求级别的 UID（UID CLn，其中 n=1,2 或者 3 ）。类型 A PICC 的 UID 长度可以是 4、7 或者 10 个字节。应答信息的长度固定为 5 个字节，其编码格式取决于 SEL 的值和 UID 的长度。应答信息的数据采用标准帧格式并且不带 CRC_A 字节，其编码格式如图 21 所示。

SEL	UID长度	应答信息（UID CLn）					
‘93’	4	UID CL1	uid0	uid1	uid2	uid3	BCC
‘93’	>4	UID CL1	CT	uid0	uid1	uid2	BCC
‘95’	7	UID CL2	uid3	uid4	uid5	uid6	BCC
‘95’	>7	UID CL2	CT	uid3	uid4	uid5	BCC
‘97’	10	UID CL3	uid6	uid7	uid8	uid9	BCC

图 20 UID 的编码

说明：

——CT 是串联标记且值位 ‘88’ 。使用 CT 的目的是为了使该卡能与较短 UID 长度的卡产生一个冲突。因此，单倍长 UID 的 uid0 和双倍长 UID 的 uid3 的值不能为 ‘88’ 。

——BCC 是 UID CLn 的校验字节。BCC 为前 4 个字节的异或值。

——uidn 是 UID 的第 n 个字节，其中 uid0 为最高有效字节。

1.4.5 SELECT

SELECT 命令用于通过类型 A 的 PICC 的 UID 选择该 PICC。

1.4.5.1 SELECT 命令

SELECT 命令采用包含 CRC_A 校验字节的标准帧格式传输，其格式如图 21 所示。

字节1	字节2	字节3-7	字节8-9
SEL	‘70’	UID CLn	CRC_A

图 21 SELECT 的编码

SEL 字节的编码方式如图 20 所示。

UID CLn 的编码取决于 SEL 的值和 UID 的长度，其格式与 ANTICOLLISION 的应答信息格式相同。

1.4.5.2 SELECT 应答（ASK）

当 PICC 接收到 SELECT 命令时，如果命令中的 UID CLn 和 PICC 的 UID CLn 完全相同则 PICC 回送 SAK。SAK 的长度为 1 个字节，使用带 CRC_A 校验字节的标准帧格式传输给 PCD。SAK 的具体编码格式如图 22 所示。

b8	b7	b6	b5	b4	b3	b2	b1	说明
0	0							保留
		X						若b6=1，则PICC遵循ISO/ICE14443-4
			0	0				保留
					X			串联比特设置：若b3=1，则UID不完整
						0	0	保留

图 21 SAK 的编码

1.4.6 HLTA

HLTA 命令用于使 PICC 进入 HALT 状态。

1.4.6.1 HLTA 命令

HLTA 命令包含两个字节，传输采用标准帧格式并且包含 CRC_A 校验字节，其格式如图 22 所示。

字节1	字节2	字节3-4
‘50’	‘00’	CRC_A

图 22 HTLA 的编码

1.4.6.2 HLTA 应答

PICC 对 HLTA 命令不做任何响应，PCD 总是假设 PICC 已经“确实接收”HLTA 命令。

1.5 ISO/ICE14443A 状态

1.5.1 POWER-OFF 状态

当处于 POWER-OFF 状态时，PICC 由于缺少载波能量而断电。

——如果 PICC 处于激励磁场，它应该在一段延迟后进入 IDLE 状态。

1.5.2 IDLE 状态

当处于 IDLE 状态时，PICC 上电并且监听指令。

——当处于 IDLE 状态时 ,PICC 应该在接收到一条有效的 WUPA 指令并且发送其 ATQA 后进入 READY 状态。

——PICC 应该葫忽略所有其他的指令和错误，并且保持 IDLE 状态。

1.5.3 READY 状态

当处于 READY 状态，ANTICOLLISION 指令可用于获得 PICC 全部的 UID 指令。

——当处于 READY 状态，PICC 应该一直处于 READY 状态，并且当接收到一条有效的 ANTICOLLISION CL1 命令后应该发送它的 UID CL1。

——当处于 READY 状态时，PICC 被完整的 UID 选择时，单倍长 UID 的 PICC 应该进入 ACTIVE 状态，例如：当它接收到一条和 UID CL1 相匹配的有效 SELECT CL1 指令时。PICC 应该在它的 SAK 响应中指示 UID 是完整的。READY ‘和 READY’ ’ 状态不存在与单倍长 UID 的 PICC。

——当处于 READY 状态，接收到一条和 UID CL1 相匹配的有效 SELECT CL1 指令时，双倍长或三倍长 UID 的 PICC 应该进入 READY‘状态。

——在所有其他的情况，PICC 应该返回到 IDLE 状态并且不应该发送响应至 PCD。

1.5.4 READY‘状态

READY‘状态是一个中间状态，它仅存在于双倍长和三倍长 UID 的 PICC 中。在此状态，UID 的第一层次被选择。

——当处于 READY’状态，PICC 应该一直处于 READY ‘状态，并且当接收到一条有效的 ANTICOLLISION CL2 命令后应该发送它的 UID CL2。

——当处于 READY’状态 ,PICC 被完整的 UID 选择时 ,双倍长 UID 的 PICC 应该进入 ACTIVE 状态，例如：当它接收到一条和 UID CL2 想匹配的有效 SELECT CL2 指令时。PICC 应在 SAK 响应中指出 UID 是完整的，双倍长 UID 的 PICC 不存在 READY ‘ ’。

——当处于 READY‘状态 ,接收到一条和 UID CL2 相匹配的有效 SELECT CL2 指令时 ,三倍长的 UID 的 PICC 进入 READY ‘’ 状态。

——在所有其他的情况，PICC 应该返回到 IDLE 状态并且不应该发送响应至 PCD。

1.5.5 READY ‘’ 状态

READY ‘’ 状态是一个中间状态，它仅存在与三倍长 UID 的 PICC 中。当处于此状态时，UID 的第 1 和第 2 层次被选择。

——当处于 READY ‘’ 状态 ,PICC 应一直处于 READY ‘’ ,并且当接收到一条有效的 ANTICOLLISIOIN CL3 命令时发送它的 UID CL3。

——当处于 READY ‘’ 状态，PICC 被完整的 UID 选择时，三倍长 UID 的 PICC 应该进入 ACTIVE 状态 ,例如 :当它接收到一条和 UID CL3 相匹配的有效 SELECT CL3 指令时。PICC 应在 SAK 响应中指示 UID 是完整的。

——当处于 READY‘状态，接收到一条和 UID CL2 相匹配的有效 SELECT CL2 指令时，三倍长的 UID 的 PICC 进入 READY’ ‘状态。

——在所有其他的情况，PICC 应该返回到 IDLE 状态并且不应该发送响应至 PCD。

1.5.6 ACTIVE 状态

当处于 ACTIVE 状态时，PICC 监听 RATS 命令，用以激活协议。

——当处于 ACTIVE 状态时，PICC 接收到一条有效的 RATS 指令并传送 ATS 后，PICC 应该进入 PROTOCOL 状态。

——当处于 ACTIVE 状态，PICC 接收到 HLTA 指令，PICC 应该进入 HALT 状态。

——当其他情况，PICC 应该返回至 IDLE 状态，并且不应该发送响应至 PCD。

1.5.7 PROTOCOL 状态

当处于 PROTOCOL 状态，PICC 监听所有的上层信息。

——当处于 PROTOCOL 状态时，当接收到一条有效的 S（DESELECT）请求块，PICC 应该进入 HALT 状态。

——当处于 PROTOCOL 状态，PICC 应该只对有效块做出答复。应该忽略所有其他的类型 A 指令（例如 WUPA，AC，SELECT，HLTA 和 RATS）和错误。

——当没有出现 S（DESELECT）请求块的时候，PICC 应该保持 PROTOCOL 状态直到掉电。

1.5.8 HALT 状态

当处于 HALT 状态，PICC 仅应答 WUPA 命令。

——当处于 HALT 状态，PICC 仅应答合法的 WUPA 命令，所有其他命令和传送错误应该被忽略。

——当处于 PROTOCOL 状态，PICC 应该只对有效块做出答复。应该忽略所有其他的类型 A 指令（例如 WUPA，AC，SELECT，HLTA 和 RATS）和错误。

——一旦 PICC 收到 WUPA 命令并应答 ATQA，PICC 应该从 HALT 状态转为 READY 状态。

1.5.9 PICC 状态图

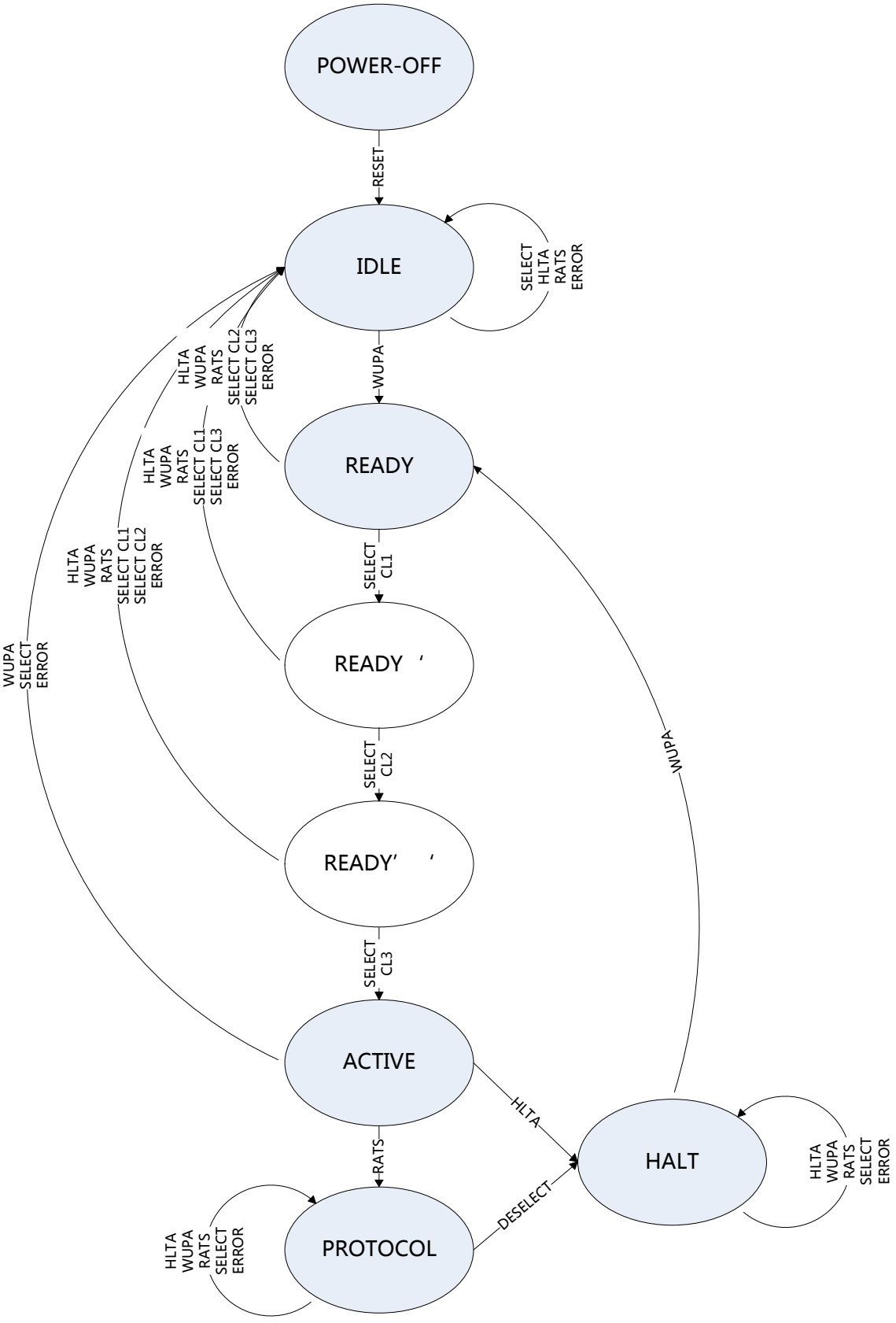

图 23 HTLA 的编码

第二章　Mifare 卡片介绍

2.1　特征

2.1.1　MIFARE RF 接口（ISO14443A）

——无线传送数据和能量不需要电池

——工作距离：最高可达 100mm

——工作频率：13.56MHz

——数据传送速度快：106kbit/s

——数据高度可靠：16 位 CRC，奇偶校验，位编码，位计数

——真正的反冲突

2.1.2　EEPROM

——1K 字节，分成 16 个区，每区有分成 4 段，每一段中有 16 个字节

——用户可以定义每一个存储器段的访问条件

——数据可以保持 10 年

——可写 100000 次

2.1.3　保密性

——需要经过 3 轮认证

——RF 信道的数据加密，有重放攻击保护

——每个区有两套独立的密钥，支持带密钥层次的多应用

——每个设备有唯一的序列号

2.2　总体描述

根据 ISO14443A 标准，NXP 开发出了无线智能卡芯片 Mifare MF1 IC S50。这个芯片的通讯层（Mifare RF 接口）遵从 ISO14443A 标准的第 2 部分和第 3 部分。保密层使用经区域验证的 CRYPTO1 流密码，使

典型 Mifare 系列芯片的数据交换得到保密。

2.2.1 无线传送数据和能量

Mifare 系统中，MF1 IC S50 连接着几匝线圈，这就形成了一张无源的无线智能卡。这种卡不需要电池。当智能卡靠近读写装置的天线时，高速 RF 通讯接口可以以 106kBit/s 的速度传送数据。

2.2.2 反冲突

智能的反冲突功能允许同一工作区域中不止一张卡同时工作。反冲突算法每次只选择一张卡，确保对被选中的卡正确执行操作而且同一区域中的其他卡不会破坏数据。

2.2.3 保密性

这个卡一个特殊的要点是保密，防止欺骗。相互询问的响应确认，数据保密和报文确认检查防止系统受到任何干扰。序列号不可修改，保证了每张卡都是唯一的。

2.2.4 多应用功能

和处理器卡的功能相比较，Mifare 系统提供了一个实时的多应用功能。每区有两个不同的密钥，这样系统可以使用密钥层次。

2.3 通信原理

通讯命令由读卡器初始化，并由 MF1 IC S50 的数字式控制单元根据相应区的有效访问条件来控制。

2.3.1 请求标准/所有

卡上电复位后，它可以给请求代码发送回应（ATQA）回复读卡器的请求命令（由读卡器发出，给所有在天线范围内的卡）。

2.3.2 反冲突循环

反冲突可以读出卡的序列号。如果读卡器的工作范围内有几张卡，读卡器通过唯一的序列号来区别它们而且每次选择其中一张卡（也叫选择卡）进行下一步操作。没有被选中的卡回到准备模式等待新的请求命令。

2.3.3 选择卡

选中了一张卡之后，读卡器指出了接着要访问的存储器位置，然后使用相应的密钥进行 3 轮确认。在成功确认之后，所有的存储器操作都是保密的。

2.3.4 3 轮确认

选中了一张卡之后，读卡器指出了接着要访问的存储器位置，然后使用相应的密钥进行 3 轮确认。在成功确认之后，所有的存储器操作都是保密的。

2.3.5 存储器操作

确认之后可以执行以下的任何操作：

——读存储器段

——写存储器段

——加值：增加存储器段的内容并将结果保存在临时的内部数据寄存器中

——减值：减少存储器段的内容并将结果保存在临时的内部数据寄存器中

——恢复：将存储器段的内容移到数据寄存器中

——传送：将临时的内部数据寄存器的内容写到值存储器段中

2.4 数据可靠（正确）性

读卡器 和卡之间的无线通讯链路使用了以下的机制确保数据可靠地传输

——每个段 16 位 CRC

——每个字节都有奇偶校验位

——位计数检查

——用位编码区别 1 0 和没有信息

——信道监控协议序列和位流分析

2.5 保密性

根据 ISO9798-2 使用 3 轮确认，保密级别很高。

3 轮确认的顺序

1、读卡器指出要访问的区并选择密钥 A 或密钥 B。

2、卡从区尾读出密钥和访问条件。然后卡发送一个随机数到读卡器（第一轮）。

3、读卡器用密钥和附加输入计算响应。然后，将响应和读卡器的随机询问一起发送到卡中（第二轮）。

4、卡用自己的询问和读卡器的响应比较确认读卡器的响应。然后卡计算询问的响应并发送出去(第三轮)。

5、读卡器用自己的询问和卡的响应相比较确认卡的响应

在发送第一个随机的询问之后，卡和读卡器之间的通讯都是保密的。

2.6 存储器结构

1024x8 位的 EEPROM 存储器被分成 16 个区，每个区有 4 个段，每段有 16 字节。

在擦除状态时，读 EEPROM 单元的值是逻辑“0“；在写状态时，读 EEPROM 单元的值是逻辑”1“。

存储器映射如图 24 所示。

		一个段内的字节																
区号	段号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	说明
15	3 (63)	KEYA						Access Bits				KEYB						区尾15
	2 (62)																	数据段
	1 (61)																	数据段
	0 (60)																	数据段
14	3 (59)	KEYA						Access Bits				KEYB						区尾14
	2 (58)																	数据段
	1 (57)																	数据段
	0 (56)																	数据段
:	:																	:
:	:																	:
:	:																	:
1	7 (7)	KEYA						Access Bits				KEYB						区尾1
	6 (6)																	数据段
	5 (5)																	数据段
	4 (4)																	数据段
0	3 (3)	KEYA						Access Bits				KEYB						区尾0
	2 (2)																	数据段
	1 (1)																	数据段
	0 (0)																	厂商段

图 24 存储器映射

2.6.1 厂商段

厂商段是存储器第一个区的第一个数据段 (段 0)。它包含了 IC 厂商的数据。基于保密性和系统的安全性，这个段在 IC 卡厂商编程之后被置为写保护，如图 25 所示。

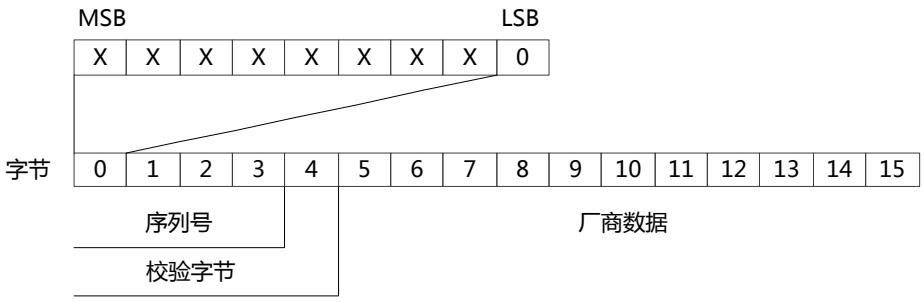

图 25 厂商段结构

2.6.2 数据段

所有的区都包含 3 个段（每段 16 字节）保存数据（区 0 只有两个数据段和一个只读的厂商段）。

数据段可以被访问控制位（access bits）配置：

——读/写段，用于譬如无线访问控制，读/写段仅支持读写操作。

——值段，用于譬如电子钱包，它需要额外的命令像直接控制保存值的增加和减少

在执行任何存储器操作前都要先执行确认命令。

2.6.3 值段

值段可以实现电子钱包的功能（有效的命令包括：读、写、加、减、恢复、传送）。

值段有一个固定的格式，可以进行错误检测和纠正并备份管理。如图 26 所示。

字节	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
说明	Value				Value				Value				Adr	$\overline{\text{Adr}}$	Adr	$\overline{\text{Adr}}$

图 26 值段的存储格式

值段只能在值段格式的写操作时产生：

——值：表示一个带符号 4 字节值。这个值的最低一个字节保存在最低的地址中。取反的字节保存在字节 4-字节 7 中。为了保证数据的正确性和保密性，值被保存了 3 次，两次不取反保存，一次取反保存。

——地址：表示一个字节的地址，当执行强大的备份管理时用于保存存储段的地址。地址字节保存了 4 次，取反和不取反各保存了 2 次。在执行加值、减值、恢复和传送等操作时，地址保持不变。它只能通过写命令改变。

2.6.4 区尾（段 3）

每个区都有一个区尾，它包括：

——密钥 A 和 B（可选），读密钥是返回逻辑 ‘0’

——访问这个区中 4 个段的条件（保存在第 6 字节-第 9 字节）。访问位（access bits）也可以指出数据段的类型（读/写或值段）。

如果不需要密钥 B，那么区尾的最后 6 个字节可以作为数据字节。

用户数据可以使用区尾的第 9 个字节，这个字节具有和字节 6、7、8 一样的访问权限。

区尾的格式如图 27 所示。

字节	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
说明	KEYA						Access Bits				KEYB					

图 27 区尾的格式

2.7 访问存储器

在执行任何存储器操作以前卡必须要被选中并经过确认

数据段可能的存储器操作要根据使用的密钥和保存在相应区尾的访问条件决定。

存储器操作类型如图 28 所示。

操作	说明	段类型
读	读一个存储器段	读/写，值，区尾
写	写一个存储器段	读/写，值，区尾
加值	增加段的内容并将结果保存在内部的数据寄存器中	值
减值	减少段的内容并将结果保存在内部的数据寄存器中	值
传送	将内部数据寄存器的值写到段中	值
恢复	将段数据读到内部数据寄存器中	值

图 28 存储器操作类型

2.7.1 访问条件

每个数据段和区尾的访问条件由 3 个位来定义，它们以取反和不取反的形式保存在指定的区尾中。

访问位控制了使用密钥 A 和 B 访问存储器的权利。当知道相关的密钥和当前的访问控制条件时，可以修改访问条件。

注：在下面的描述中，访问位是以不取反的形式显示。

MF1 IC S50 的内部逻辑确保命令只有在确认操作完毕后执行，否则不执行。

各区的访问位定义如图 29 所示。

访问位	有效命令	段	说明
$C1_3C2_3C3_3$	读、写 →	3	区尾
$C1_2C2_2C3_2$	读、写、加、减、传送、恢复 →	2	数据区
$C1_1C2_1C3_1$	读、写、加、减、传送、恢复 →	1	数据区
$C1_0C2_0C3_0$	读、写、加、减、传送、恢复 →	0	数据区

图 29 各区访问位定义

访问位在区尾的存储形式如图 30 所示。

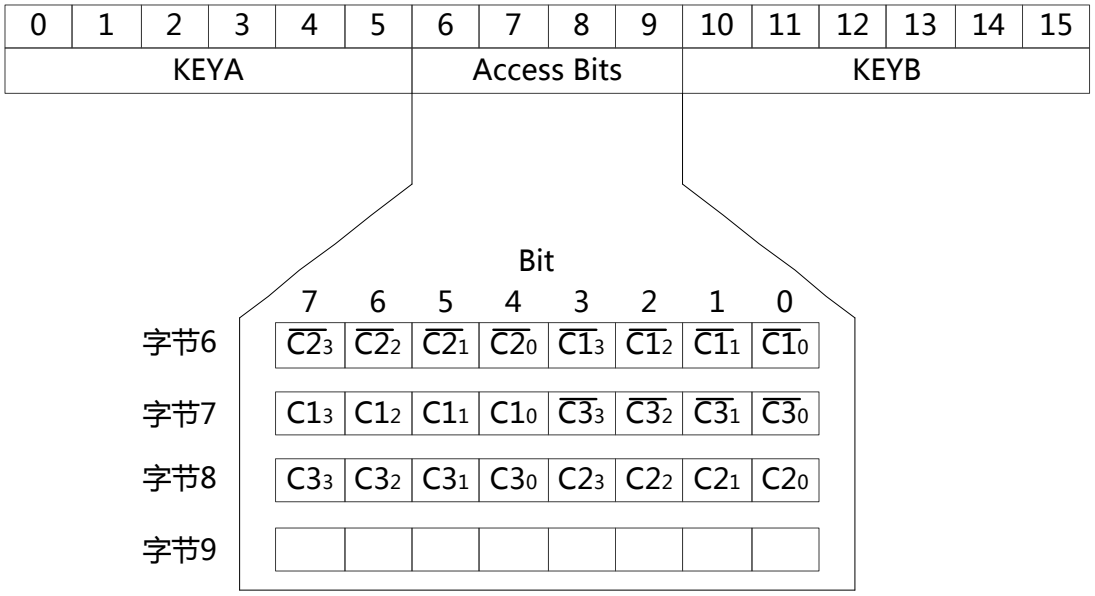

图 3 0 访问位存储格式

2.7.2 区尾的访问条件

区尾（段 3）的访问位，密钥和访问位的读写访问可分为“从不”、“密钥 A”、“密钥 B”或“密钥 A|B”（密钥 A 或密钥 B）。

卡片的区尾和密钥 A 在出厂时，被初始化。

由于访问位可以被阻塞，在个人化卡的时候要特别注意。

区尾的访问条件如图 31 所示。

访问位			访问条件						说明
			KEYA		Access Bits		KEYB		
C1	C2	C3	读	写	读	写	读	写	
0	0	0	never	KEYA	KEYA	never	KEYA	KEYA	密钥B可被读
0	1	0	never	never	KEYA	never	KEYA	never	密钥B可被读
1	0	0	never	KEYB	KEYA|B	never	never	KEYB	
1	1	0	never	never	KEYA|B	never	never	never	
0	0	1	never	KEYA	KEYA	KEYA	KEYA	KEYA	密钥B可被读 传送配置
0	1	1	never	KEYB	KEYA|B	KEYB	never	KEYB	
1	0	1	never	never	KEYA|B	KEYB	never	never	
1	1	1	never	never	KEYA|B	never	never	never	

图 31 区尾的访问条件

注：用灰色标明的行是密钥 B 可被读的访问条件，此时密钥 B 可以存放数据。

例如：当段 3 的访问条件 $C1_3C2_3C3_3$=100 时，表示：密钥 A 不可读（隐藏），验证密钥 B 正确后，可写（或更改）；访问控制位在验证密钥 A 或密钥 B 正确后，可读不可写（写保护）；密钥 B 不可读，在验证密钥 B 正确后可写。

又如：当段 3 的访问条件 $C1_3C2_3C3_3$=110 或者 111 时，除访问控制位需要在验证密钥 A 或密钥 B 正确后仅仅可读外，其他如访问控制位的改写，密钥 A，密钥 B 的读写权限均被锁死而无法访问。

2.7.3 数据段的访问条件

数据段（段 0-2）的访问位，读/写访问可分为“从不”、“密钥 A”、“密钥 B”或“密钥 A|B”（密钥 A 或密钥 B）。相关访问位的设置定义了应用以及相应的应用命令。

——读/写段：可以进行读操作和写操作。

——值段：可以进行加、减、传送和恢复的值操作。其中一种情况中（“001”）只能对不可再充电的卡进行读操作和减操作。另一种情况中（“110”）使用密钥 B 可以再充电。

——厂商段：无论设置任何的访问位，这个段都只是只读的。

——密钥管理：在传输配置中，密钥 A 必须用于确认。

数据段的访问条件如图 32 所示。

访问位			访问条件				应用
C1	C2	C3	读	写	加值	减值、传送、恢复	
0	0	0	KEYA|B	KEYA|B	KEYA|B	KEYA|B	传送配置
0	1	0	KEYA|B	never	Never	never	读/写段
1	0	0	KEYA|B	KEYB	never	never	读/写段
1	1	0	KEYA|B	KEYB	KEYB	KEYA|B	值段
0	0	1	KEYA|B	never	never	KEYA|B	值段
0	1	1	KEYB	KEYB	never	never	读/写段
1	0	1	KEYB	never	never	never	读/写段
1	1	1	never	never	never	never	读/写段

图 32 数据段的访问条件

注：如果密钥 B 可以在相应的区尾被读出，它就不能用于确认（在前面所有表中的灰色行）。如果读卡器要用这些（带灰色标记的）访问条件的密钥 B 确认任何段，卡会在确认后拒绝任何存储器访问操作。

2.8 Mifare S50 初始化值

MF1 卡分为 16 个扇区，每区有 4 段(段 0～段 3)，共 64 段，按段号编址为 0～63。第 0 扇区的段 0(即绝对地址块 0)用于存放芯片商，卡商相关代码，已经固化不可更改。其他各扇区的段 0，段 1，段 2 为数据块，用于存贮用户数据；块 3 为各扇区控制段，用于存放密码 A，存取控制条件设置，密码 B。各区控

制段结构相同，如图 33 所示：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
KEYA						Access Bits				KEYB					
FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF

图 33 段 3 的初始值

Mifare S50 的出厂时，访问控制字节（字节 6-字节 9）被初始化为“FF 07 80 69”，所以初始访问控制位默认值为 :$C1_0C2_0C3_0$=000 ;$C1_1C2_1C3_1$=000 ;$C1_2C2_2C3_2$=000 ;$C1_3C2_3C3_3$=001。KEYA 和 KEYB 的默认值为 FF FF FF FF FF FF。

则区尾的访问控制字节按如图 34 格式存储。

Bit	7	6	5	4	3	2	1	0
字节6	1	1	1	1	1	1	1	1
字节7	0	0	0	0	0	1	1	1
字节8	1	0	0	0	0	0	0	0

图 34 区尾的访问条件

如果用户要读段 1 的内容，按照之前的说明，当访问控制 $C1_1C2_1C3_1$=000 时，必须正确校验 KEYA 或 KEYB 后才可允许读取块 1 的内容，否则，读卡器会因校验某区密码出错而无法读取和传送数据。以此类推，用户要进行其他操作时，可根据访问控制条件，按照之前的说明来决定其操作权限。

2.9 Mifare S50 卡修改各区段访问条件的值和数据举例

2.9.1 以常用设置“08 77 8F 69”访问条件为例，首先搞清楚它具有的访问权限。

1、对“08 77 8F 69”值进行计算，该值定位于各区段 3 的 6,7,8,9 四个字节内，字节 6=“08”，字节 7=“77”，字节 8=“8F”，字节 0=“69”（默认值，不予计算）。

2、例如：字节 6=“08”，对应其二进制值=00001000，则对 6,7,8 这三个字节进行二进制转换结果见图 35。

字节6=00001000	字节7=01110111	字节8=10001111

图 35 字节 6,7,8 的二进制转换结果

3、将字节 6 的全部二进制值取反，字节 7 的低四位二进制值取反，字节 8 不变，得到的结果如图 36 所示。

字节号	对应二进制值	位置	高4位	低4位	低4位
字节6	00001000	C2Y	1111	C1Y	0111
字节7	01110111	C1Y	0111	C3Y	1000
字节8	10001111	C3Y	1000	C2Y	1111

图 36 取反后的结果

4、对以上的字节 6，7，8 的值已取反值，得到的转换后的各段访问条件如图 37 所示。

段3	字节7，字节6，字节8 = C13，C23，C33 = C1Y，C2Y，C3Y = 0 1 1
段2	字节7，字节6，字节8 = C12，C22，C32 = C1Y，C2Y，C3Y = 1 1 0
段1	字节7，字节6，字节8 = C11，C21，C31 = C1Y，C2Y，C3Y = 1 1 0
段0	字节7，字节6，字节8 = C10，C20，C30 = C1Y，C2Y，C3Y = 1 1 0

图 37 转换后的结果

注意：高 4 位的各段值=低 4 位的各段值时，其值可用。高 4 位值≠低 4 位值，其值不可用。

5、查看访问权限，该例“08 77 8F 69”的访问权限为：

——段 3=011：权限为：KEYA，KEYB 均不可读，验证 KEYB 正确后可改写 KEYA 和 KEYB，验证 KEYA 或 KEYB 正确后可读“访问控制位”。在此可见密钥 KEYB 的重要性，KEYB 不正确是无法看到段 3 的访问控制位，更无法修改密钥。

——段 2=段 1=段 0=110：权限为：验证 KEYA 或 KEYB 后可读该段数据，减值以及初始化值，只有验证 KEYB 正确后才可改写该段数据，在此可以看出密钥 KEYB 对改写数据块也起着关键性作用。

2.9.2 “08 77 8F 69”访问控制条件设置步骤

由上可知 KEYB 设置后为不可读，并且改写数据和改写控制位都需要正确验证它，故 KEYB 设置后程序

操作员必须妥善保管 KEYB 值 ,否则以后改写数据和控制位时 ,不正确的 KEYB 值将无法实现卡的任何操作!!!

1、 修改段 3 控制位的值：最初的各区段 3 内的 KEYA，KEYB 都是厂商 12 个"F"默认值(KEYA 在任何条件下均为不可读 ,大部分读写机程序表现 KEYA 为未知的 12 个"0"）,在修改控制值时，先不要修改默认密码 KEYA 和 KEYB，在控制位修改成功后，再去更改新密码值。即先对段 3 的控制位进行修改(默认值 FF 07 80 69 改为新值 08 77 8F 69)并执行写操作。控制位写成功后，KEYB 亦为 12 个"0"不可读了，但仍是隐藏的 12 个"F"默认值。

2、 修改段 3 的 KEYA 和 KEYB 值 控制位 08 77 8F 69 值写成功后 验证 KEYB 正确后方可改 KEYA 和 KEYB 新密码。在密码操作模式键入要改写区块之先前密码 B(先前密码为默认值时，则不需改动和加载)，加载后反回数据操作模式，再进行读值，KEYA 和 KEYB 值的改写。

3、 修改段 0～段 2 中数据 :由新的控制条件“08 77 8F 69“可知 ,要修改数据 ,必须先验证 KEYB，故先设置密码操作为 KEYB 认证方式，加载后再返回数据操作模式，对要修改的数据段进行值的改写操作。

4、 上例中分析了"08 77 8F 69"的访问条件及其改写步骤 ,对用户的其它控制条件亦可参照应用。

第三章　实验一 14443A 协议实验

3.1　实验目的

1、　了解 ISO14443A 协议的相关内容

2、　深刻理解 ISO14443A 的 PICC 的各个状态

3、　熟悉并能熟练运用 ISO14443A 所涉及的各个命令

3.2　实验设备

1、　基于 ISO14443A 协议的 13.56MHz 高频读卡器

2、　NXP Mifare One S50 卡片

3.3　实验内容

1、　通过控制读卡器向卡片发出不同的命令来使卡片进入各个不同的状态

3.4　实验步骤

1、　将读卡器通过 USB 连接线连接到 PC 机的 USB 口。

2、　打开 PC 机上所安装的调试软件，如图 38 所示。

图 37 调试软件

3、 当卡片远离读卡器时，由于卡片没有处于电磁场中，它没有被激励，所以它处于 POWER-OFF 状态。

4、 将卡片接近读卡器，此时卡片处于电磁场中，它被激励上电，进而进入 IDLE 状态。

5、 通过调试软件向读卡器发送 WUPA 指令（十六进制格式：AA 01 07 4E），若指令执行成功，读卡器返回成功指令（十六进制格式：AA 02 04 00 50，其中 04 00 为 WUPA 应答，即 ATQA），卡片进入 READY 状态；若指令执行不成功，读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A），卡片仍然停留在 IDLE 状态。如图 38 所示。

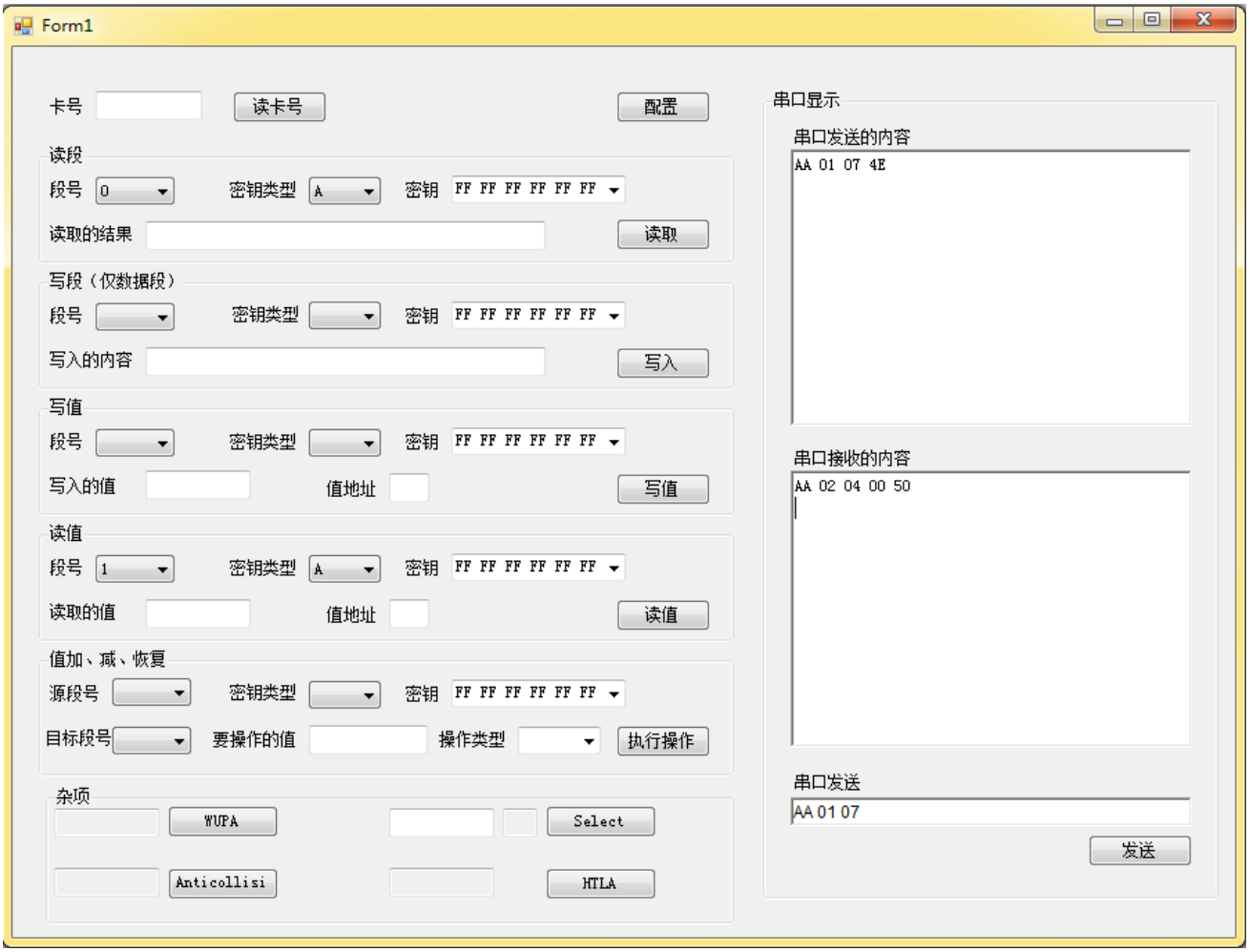

图 38 发送 WUPA 指令

6、 通过调试软件向读卡器发送 ANTICOLLISION 指令（十六进制格式： AA 01 08 4D），若指令执行成功，读卡器返回成功指令（十六进制格式： AA 04 XX XX XX XX 校验字节，其中 XX XX XX XX 为所读取的 4 字节卡号），卡片进入 ACTIVE 状态；若指令执行不成功，读卡器返回错误指令（十六进制格式： AA 03 45 52 52 6A），卡片返回 IDLE 状态。如图 39 所示。

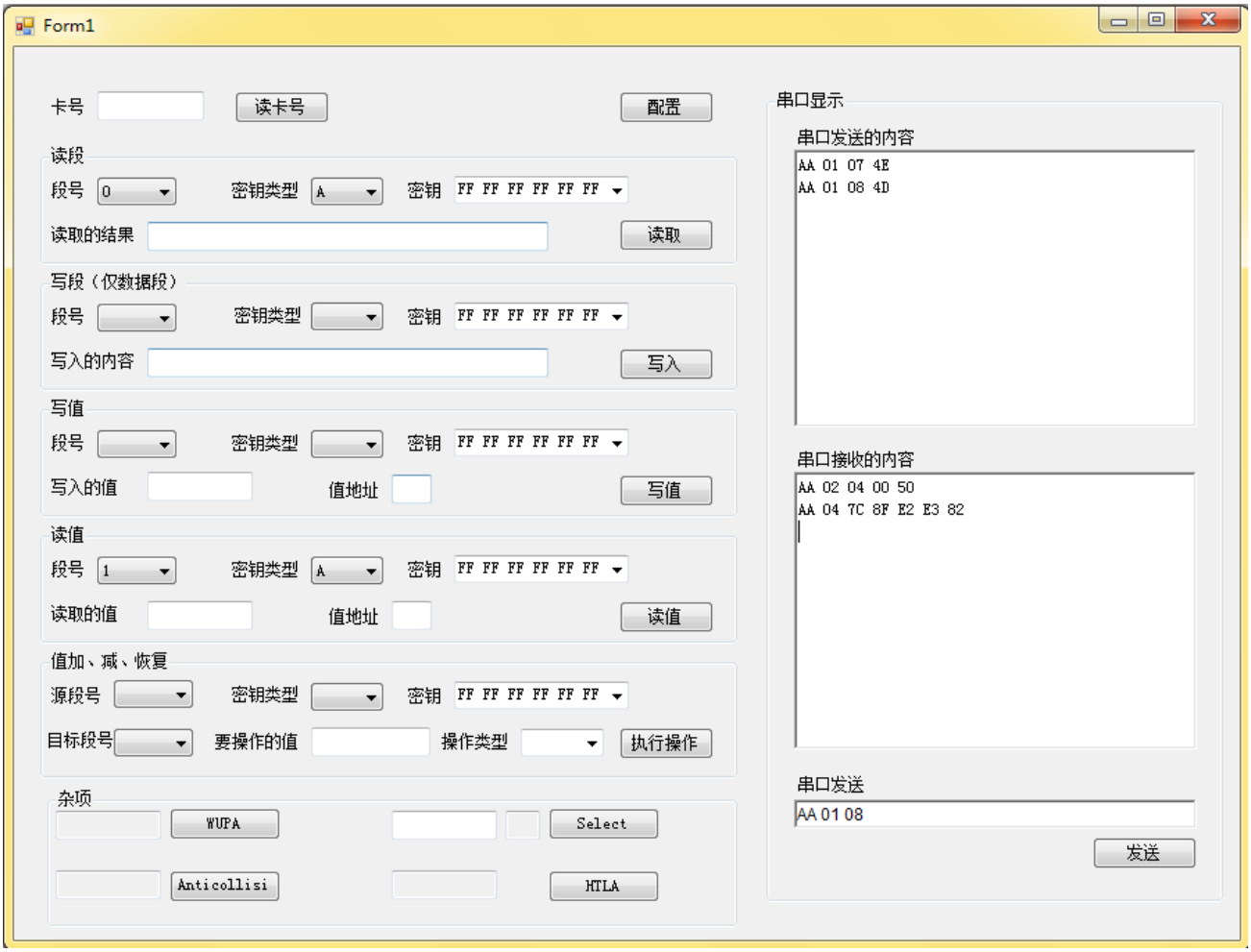

图 39 发送 ANTICOLLISION 指令

7、 通过调试软件向读卡器发送 SELECT 指令(十六进制格式 :AA 05 09 XX XX XX XX 校验字节)，若指令执行成功，读卡器返回成功指令（十六进制格式：AA 01 XX 校验字节，其中 XX 是 SELECT 应答，即 SAK)，卡片进入 ACTIVE 状态；若指令执行不成功，读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A)，卡片返回 IDLE 状态。卡片进入 ACTIVE 状态后，可以通过读卡器发送其他操作卡片的具体指令，如读卡、写卡、加值、减值等，这些操作我们将在以后涉及到。如图 40 所示。

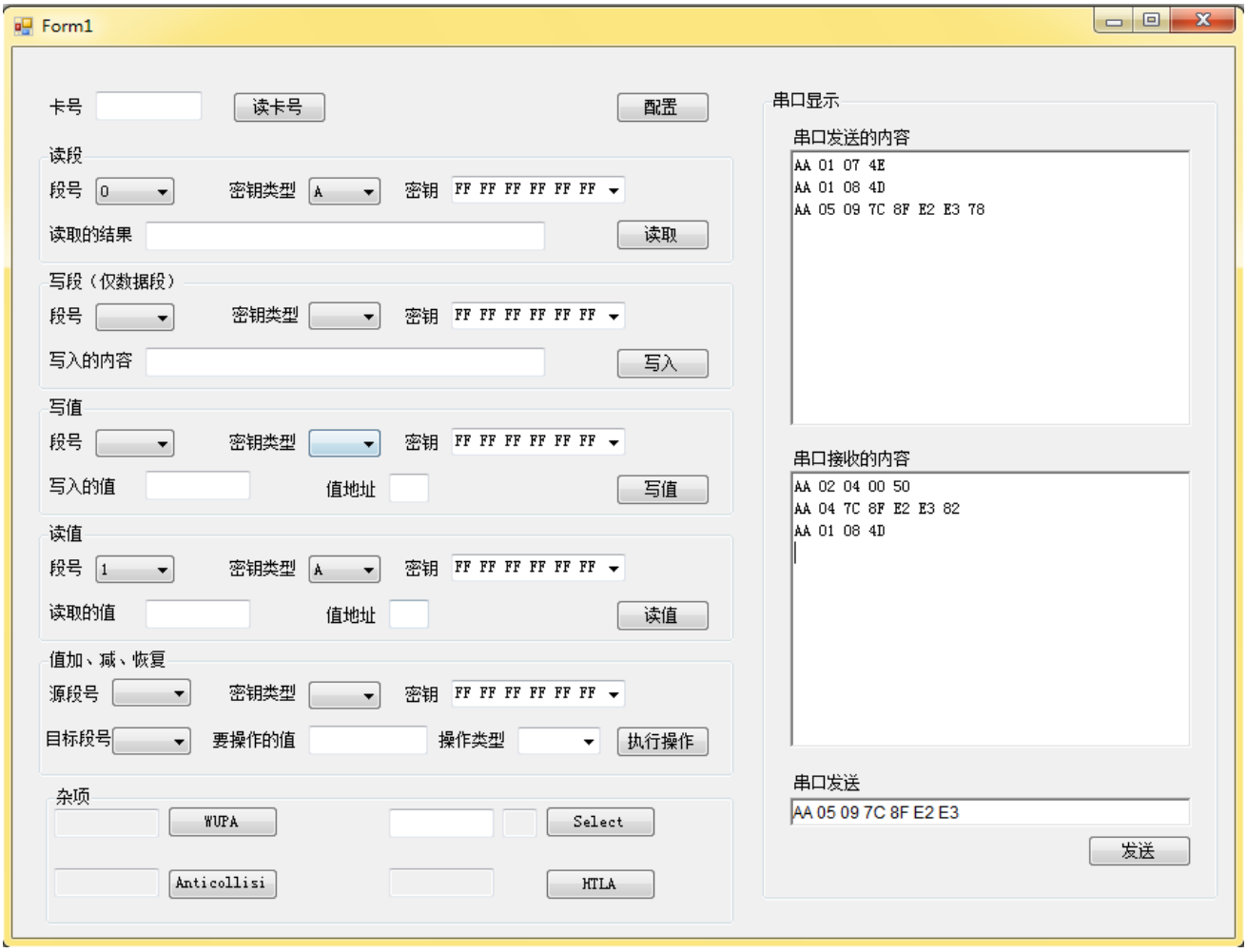

图 40 发送 SELECT 指令

8、 通过调试软件向读卡器发送 HLTA 指令（十六进制格式：AA 01 0A 4B），若指令执行成功，读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C），卡片进入 HALT 状态；若指令执行不成功，读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A），卡片返回 IDLE 状态。如图 41 所示。

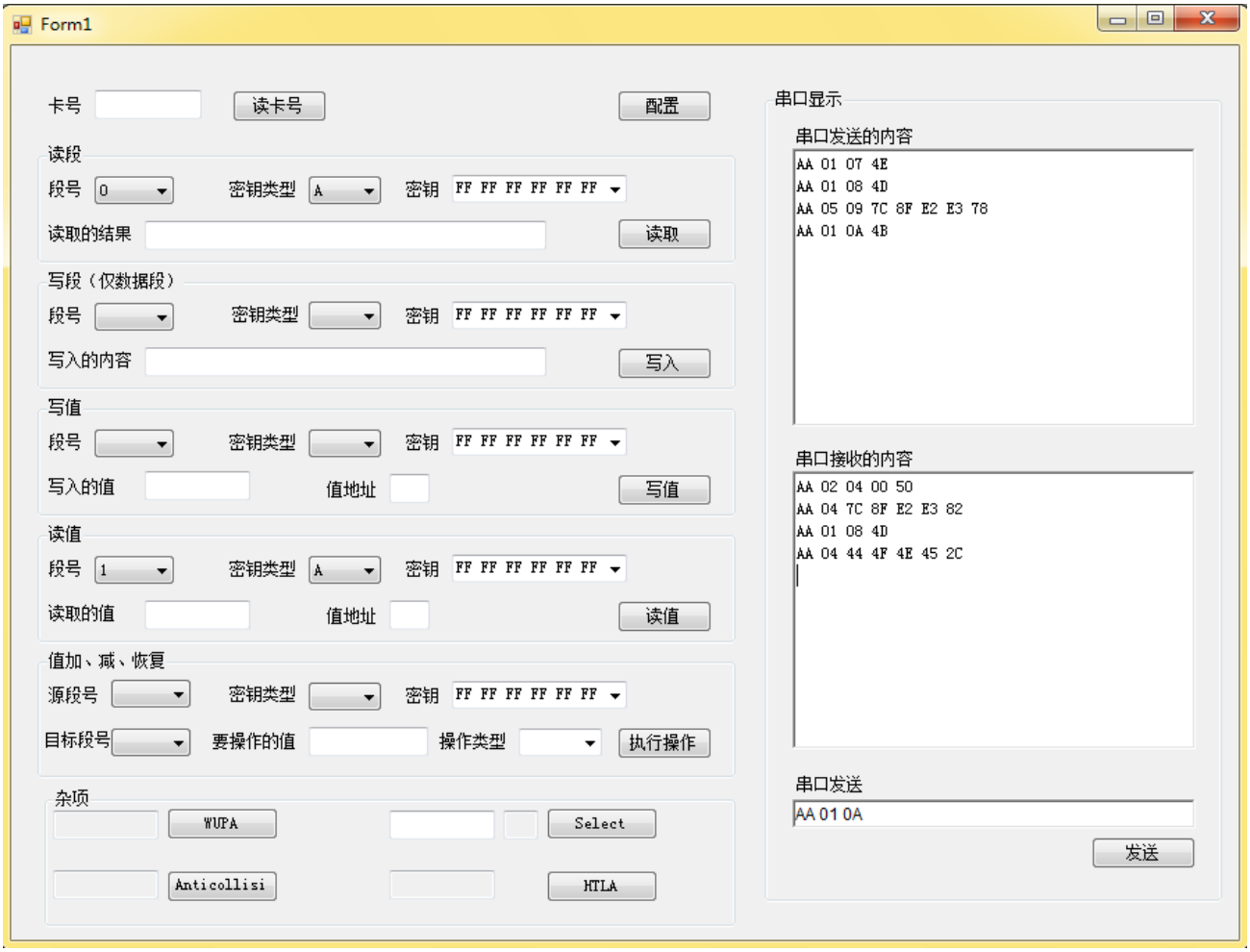

图 41 发送 HLTA 指令

9、 根据状态图发送其他相应的命令使卡片进入其他的状态来进一步熟悉卡片的相关操作。

第四章　实验二 Mifare S50 卡片存储器读写实验

4.1　实验目的

1、　了解 Mifare S50 卡片的通信原理

2、　了解 Mifare S50 卡片的存储器结构

3、　对 Mifare S50 卡片的数据区进行读写操作

4.2　实验设备

1、　基于 ISO14443A 协议的 13.56MHz 高频读卡器

2、　NXP Mifare One S50 卡片

4.3　实验内容

1、　通过调试软件控制读卡器向卡片发送读存储器命令来进行读卡操作

2、　通过调试软件控制读卡器向卡片发送写存储器命令来进行写卡操作

4.4　实验步骤

1、　将读卡器通过 USB 连接线连接到 PC 机的 USB 口。

2、　打开 PC 机上所安装的调试软件。如图 42 所示。

图 42 调试软件

3、 将卡片接近读卡器，此时卡片处于电磁场中，它被激励上电，进而进入 IDLE 状态。

4、 通过调试软件向读卡器发送读卡 ID 指令（十六进制：AA 01 01 54），若指令执行成功，则返回卡号（十六进制：AA 04 XX XX XX XX 校验字节，XX XX XX XX 为卡号）；若指令执行失败，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 43 所示。

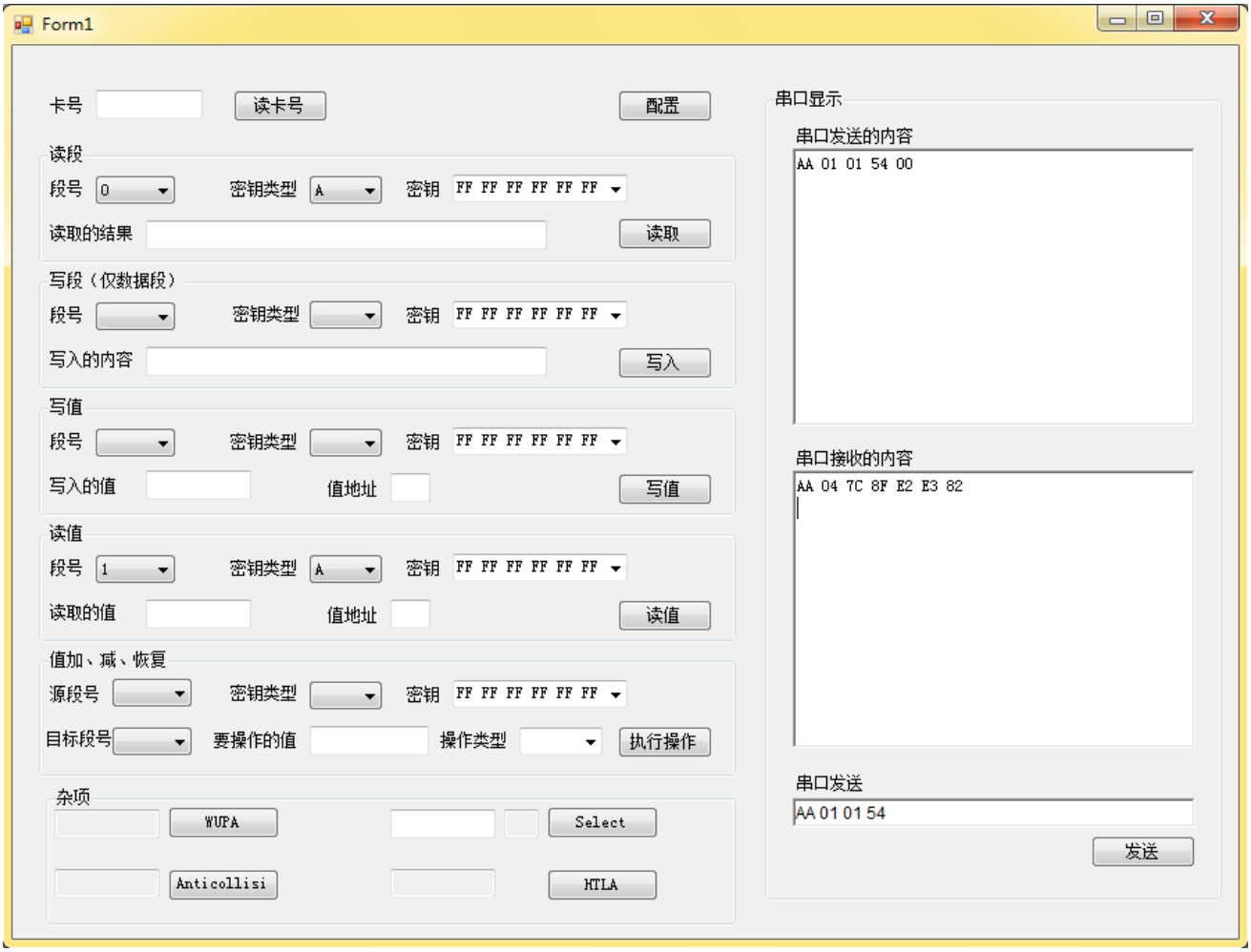

图 43 发送读卡 ID 指令

5、 通过调试软件向读卡器发送读卡指令(十六进制格式 :AA 09 02 KEY BLOCK XX XX XX XX XX XX 校验字节，其中 KEY 的值若为 60 则使用密钥 A，若为 61 则使用密钥 B；BlOCK 为要读取的块号，取值范围为 0-63；XX XX XX XX XX XX 为 6 字节的密钥值)，若指令执行成功则读卡器返回读取的数据 (十六进制格式：AA 10 XX XX XX XX XX XX XX XX XX XX XX XX XX XX 校验字节，其中 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX 为读取的 16 个字节的数据);若指令执行失败 ,则读卡器返回错误指令(十六进制格式 :AA 03 45 52 52 6A)。例如，使用密钥 A 读取第 0 区的第 0 段，即厂商段的值，通过调试软件发送 (十六进制格式：AA 09 02 60 00 FF FF FF FF FF FF F1)，若指令执行成功，则读卡器返回读取的数据(十六进制格式 :AA 10 XX XX XX XX 41 08 04 00 62 63 64 65 66 67 68 69 校验字节，其中 XX XX XX XX 为序列号，41 为校验字节，其后的 11 个字节为厂商数据); 若指令执行不

成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 44 所示。

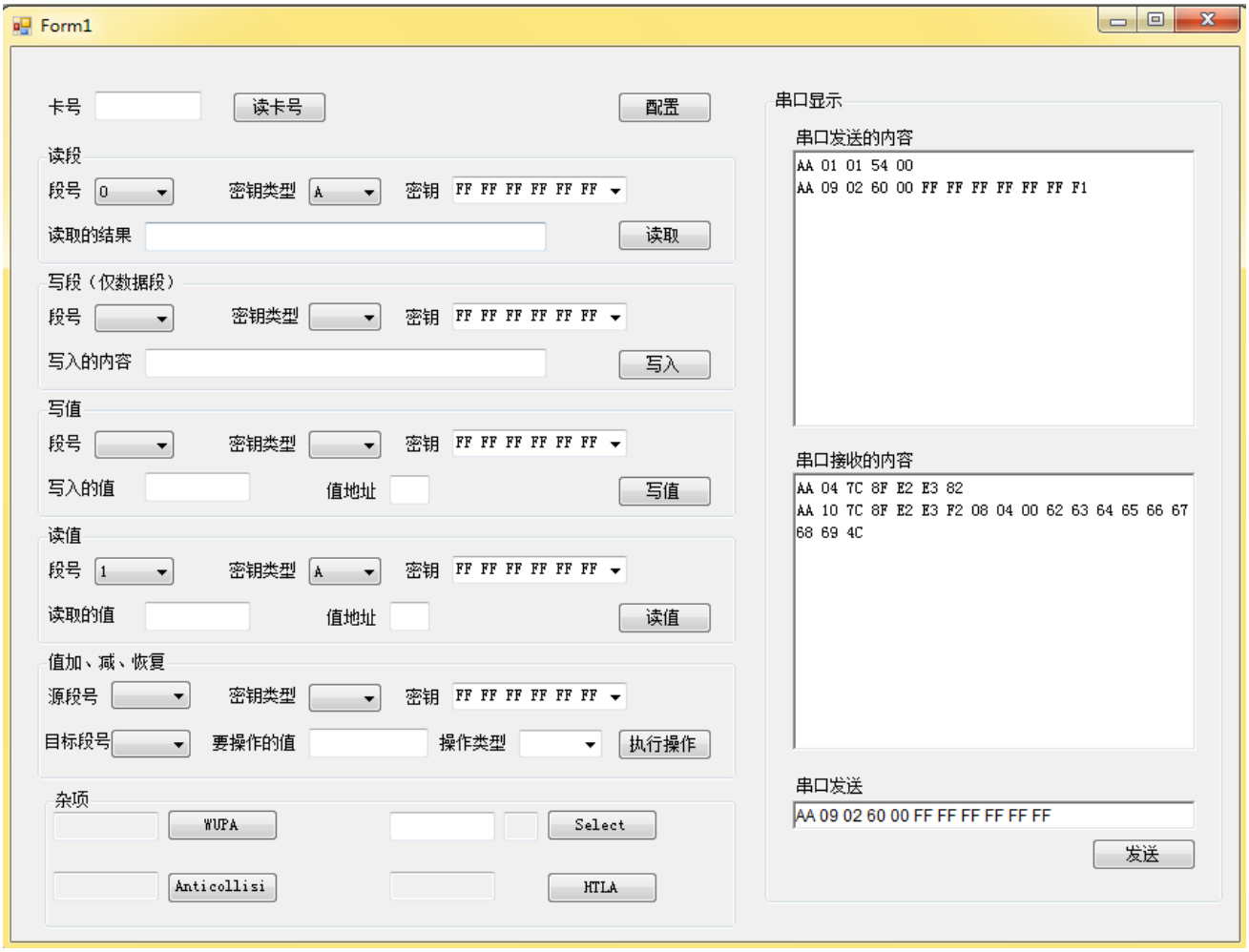

图 44 发送读卡指令

6、 通过调试软件向读卡器发送写卡指令（十六进制格式：AA 19 03 KEY BLOCK FF FF FF FF FF FF XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX 校验字节，其中 KEY 的值若为 60 则使用密钥 A，若为 61 则使用密钥 B；BlOCK 为要读取的块号，取值范围为 0-63；FF FF FF FF FF FF 为 6 字节的密钥值；XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX 为要写入的 16 个字节的数据），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。例如，使用密钥 A（FF FF FF FF FF FF）向第 0 个区的第 1 个段写入数据 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F，则通过软件发送（十六进制格式：AA 19 03 60 01 FF FF FF FF FF FF 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 67），若指令执行成

功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 45 所示。

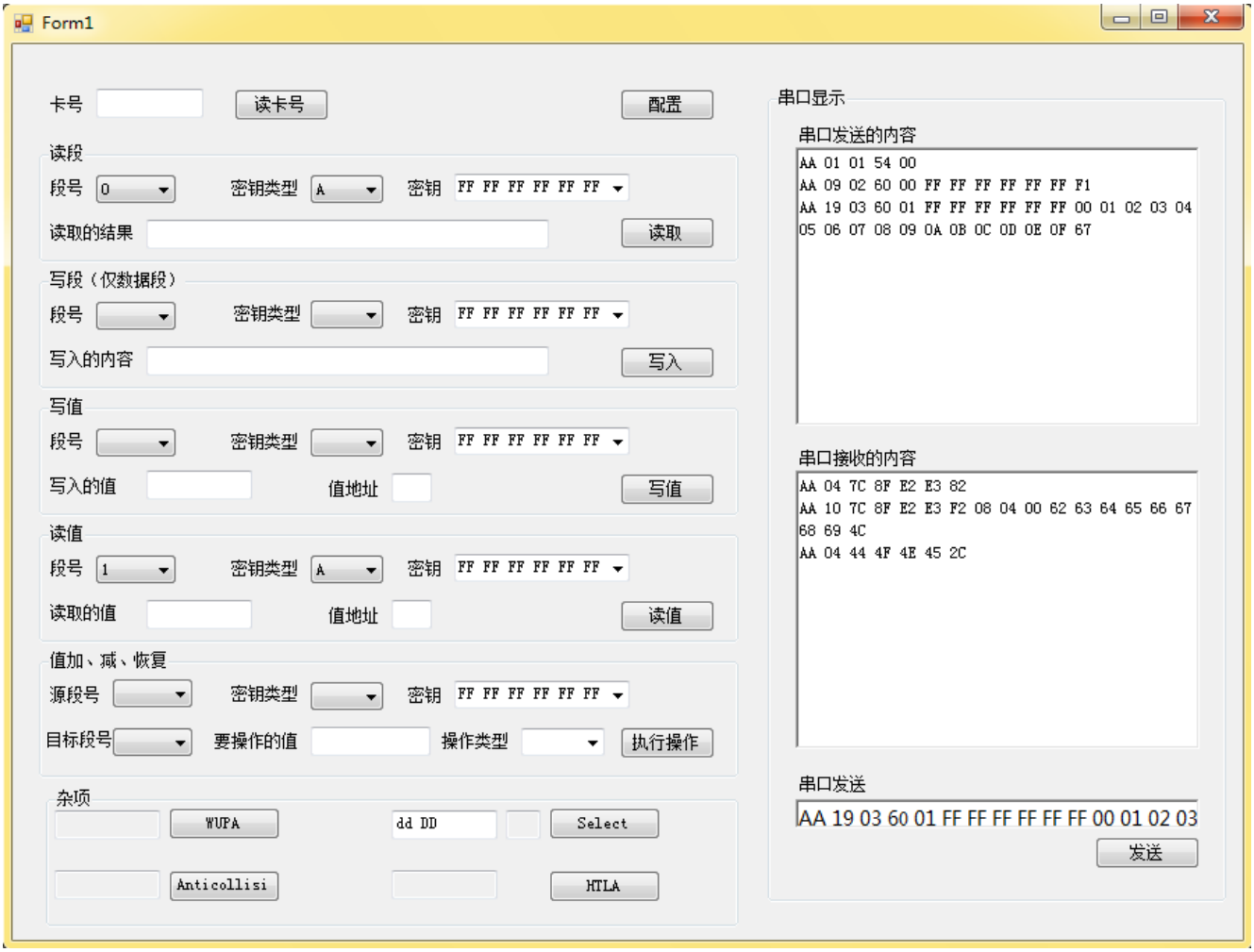

图 45 发送写卡指令

要注意的是，在此实验中不要向每个区的尾段写数据，因为这样或许将改变密钥 A、密钥 B 和控制位的值，这样卡片以后的操作会受到影响，对于区尾的操作将在接下来的实验中涉及。

7、 在向数据区中成功写入数据后，可以通过调试软件发送读卡指令来验证写入的数据是否正确。例如，使用密钥 A 读取第 0 区的第 1 个段的数据，则发送指令（AA 09 02 60 01 FF FF FF FF FF FF F0）若返回的数据为(AA 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F CE），则说明上一步写入的数据是正确的。如图 46 所示。

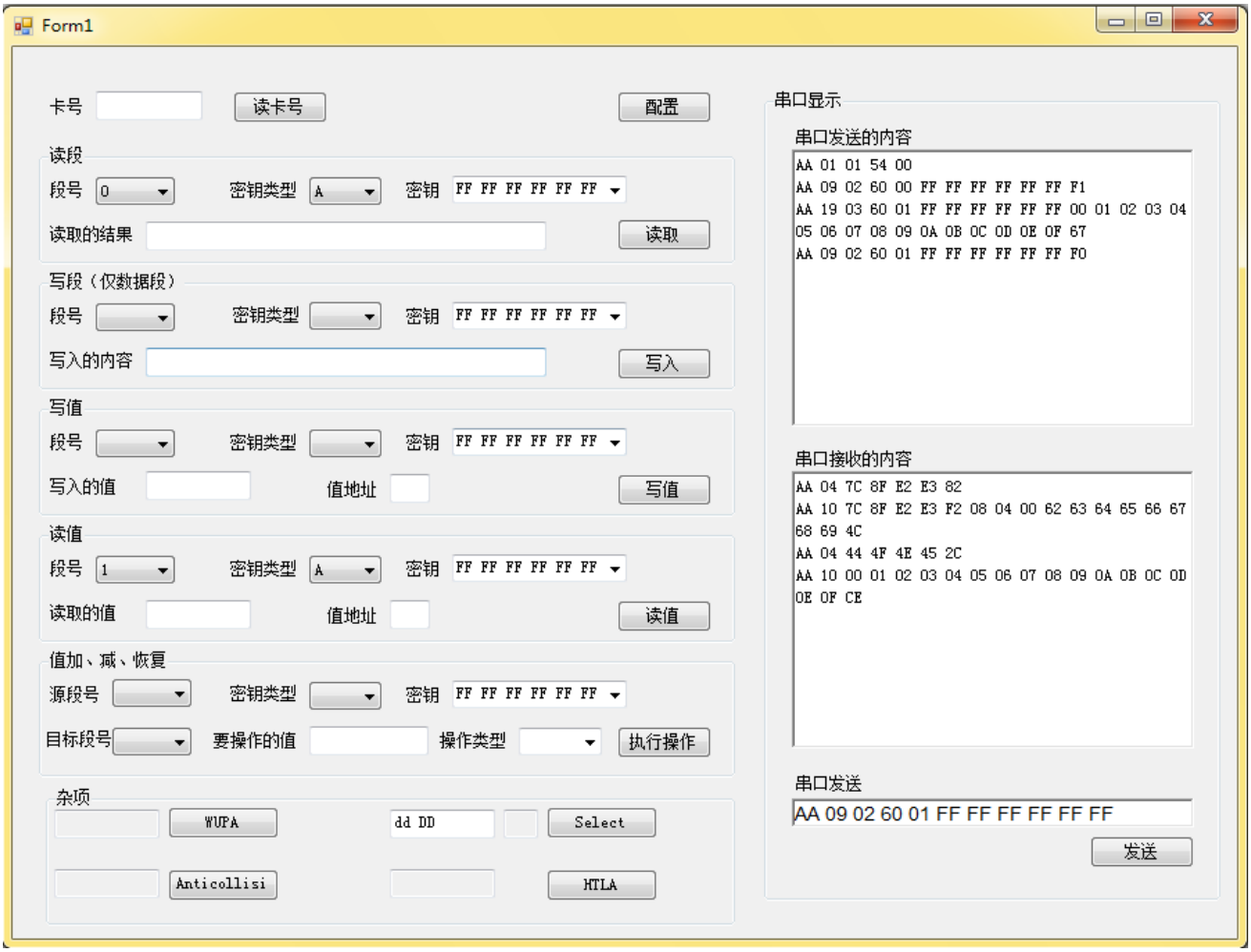

图 46 发送读卡指令

8、 使用密钥 B 对重复上述实验。

第五章　实验三 Mifare S50 卡片加密及访问控制位修改实验

5.1　实验目的

1、　理解 Mifare S50 卡片的区尾结构

2、　理解 Mifare S50 卡片的密钥系统并进行修改

3、　理解 Mifare S50 卡片的访问控制条件并进行修改

5.2　实验设备

1、　基于 ISO14443A 协议的 13.56MHz 高频读卡器

2、　NXP Mifare S50 卡片

5.3　实验内容

1、　通过调试软件控制读卡器向卡片发送修改密钥的指令来修改密钥

2、　通过调试软件控制读卡器向卡片发送修改访问控制条件的指令来修改访问控制条件

5.4　实验步骤

1、　将读卡器通过 USB 连接线连接到 PC 机的 USB 口。

2、　打开 PC 机上所安装的调试软件。如图 47 所示。

图 47 调试软件

3、 将卡片接近读卡器，此时卡片处于电磁场中，它被激励上电，进而进入 IDLE 状态。

4、 要修改卡片的访问密钥及控制条件首先要清楚卡片现在的密钥及访问控制条件。一张全新的卡片它的密钥 A 和密钥 B 都为“FF FF FF FF FF FF”，而访问控制条件为“FF 07 80 69”，按照此前所述，$C1_0C2_0C3_0$=000；$C1_1C2_1C3_1$=000；$C1_2C2_2C3_2$=000；$C1_3C2_3C3_3$=001。根据此前所说的区尾的访问控制权限可知，如果要修改区尾的值必须要先认证密钥 A。

5、 假设要将第 0 区的密钥 A 的值修改为“00 11 22 33 44 55”，则通过调试软件向读卡器发送写存储器段指令(十六进制格式 :AA 19 03 60 03 FF FF FF FF FF FF 00 11 22 33 44 55 FF 07 80 69 FF FF FF FF FF FF F5 ,其中 03 为第 0 区的区尾的段号 ,60 表示使用密钥 A 进行认证，全新的卡片的密钥 A 的值为 FF FF FF FF FF FF，00 11 22 33 44 55 的值为要修改的密钥 A 的值，访问控制条件为 FF 07 80 69 保持不变，KEYB 的值 FF FF FF FF FF FF 保持不变，F5

为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 48 所示。

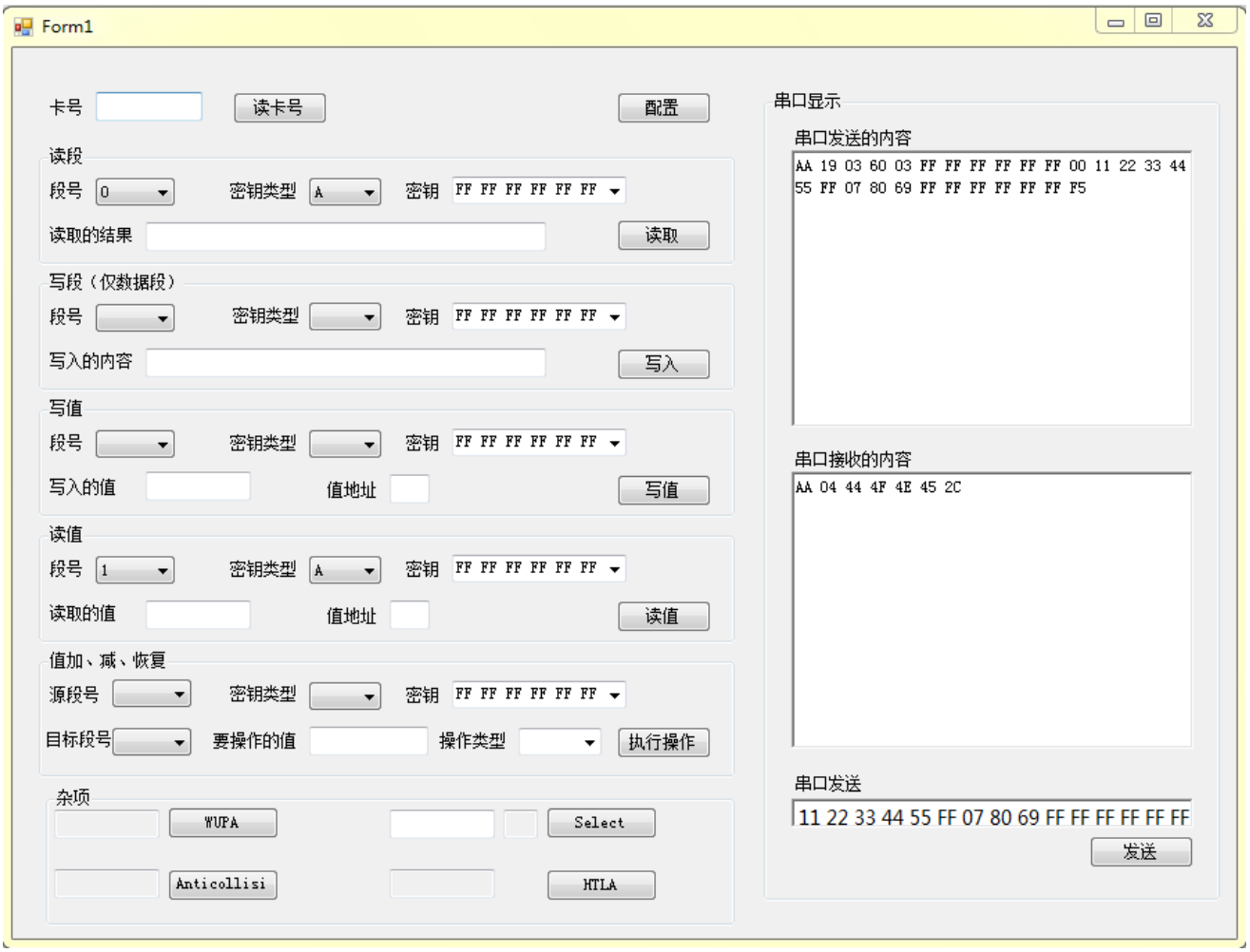

图 48 发送写卡指令

6、 使用新修改的密钥 A”00 11 22 33 44 55”来读取第 0 区的区尾的值，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 60 03 00 11 22 33 44 55 E9，其中 03 为第 0 区的区尾的段号，60 表示使用密钥 A 进行认证，这里应该使用修改后的第 0 区的密钥 A 的值 00 11 22 33 44 55，E9 为校验字节），若指令执行成功，则读卡器返回第 0 区区尾数据（十六进制格式：AA 10 00 11 22 33 44 55 FF 07 80 69 FF FF FF FF FF FF 5E），说明密钥 A 已经修改成功；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 49 所示。

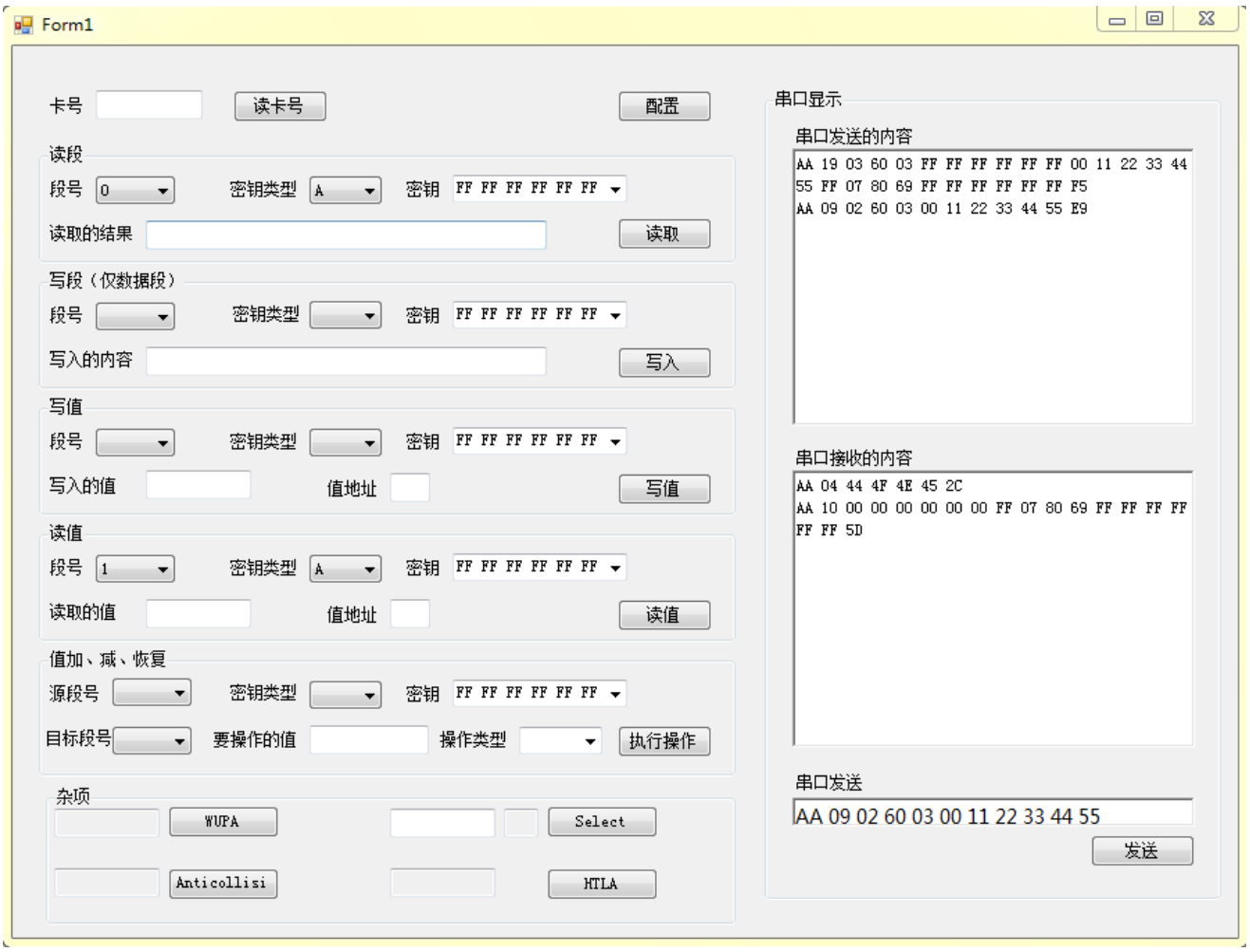

图 49 发送读卡指令

7、 接下来使用新修改的密钥 A 的值“00 11 22 33 44 55”来修改第 0 区的密钥 B 的值为“00 01 02 03 04 05”，通过调试软件向读卡器发送写存储器段指令（十六进制格式：AA 19 03 60 03 00 11 22 33 44 55 00 11 22 33 44 55 FF 07 80 69 00 01 02 03 04 05 DB，其中 03 为第 0 区的区尾的段号，60 表示使用密钥 A 进行认证，这里应该使用修改后的第 0 区的密钥 A 的值 00 11 22 33 44 55，00 11 22 33 44 55 表示密钥 A 的值，访问控制条件为 FF 07 80 69 保持不变，密钥 B 的值修改为 00 01 02 03 04 05，DA 为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 50 所示。

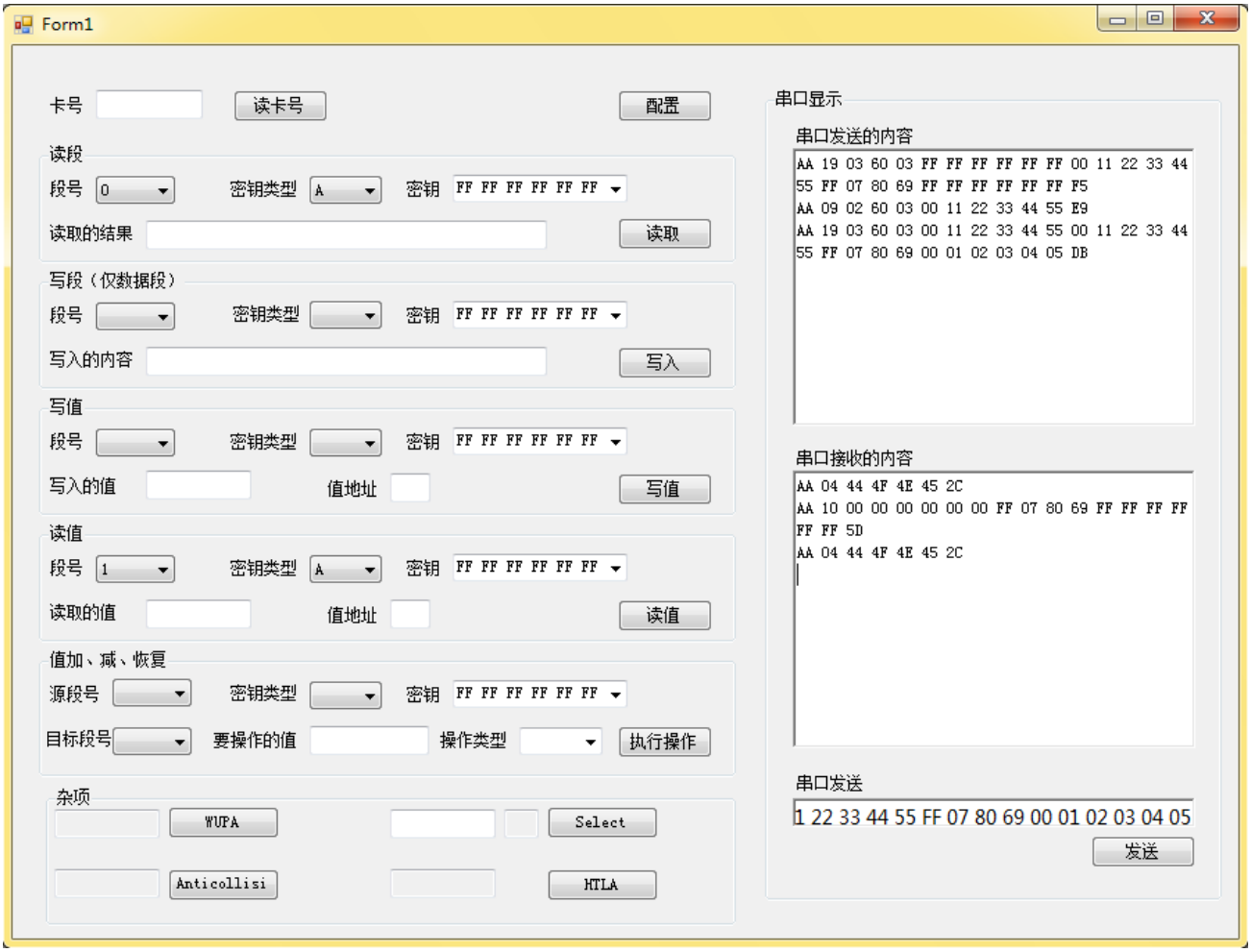

图 50 发送写卡指令

8、 使用新修改的密钥 A”00 11 22 33 44 55”来读取第 0 区的区尾的值，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 60 03 00 11 22 33 44 55 E9，其中 03 为第 0 区的区尾的段号，60 表示使用密钥 A 进行认证，这里应该使用修改后的第 0 区的密钥 A 的值 00 11 22 33 44 55，E9 为校验字节），若指令执行成功，则读卡器返回第 0 区区尾数据（十六进制格式：AA 10 00 11 22 33 44 55 FF 07 80 69 00 01 02 03 04 05 49）,说明密钥 B 已经修改成功；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 51 所示。

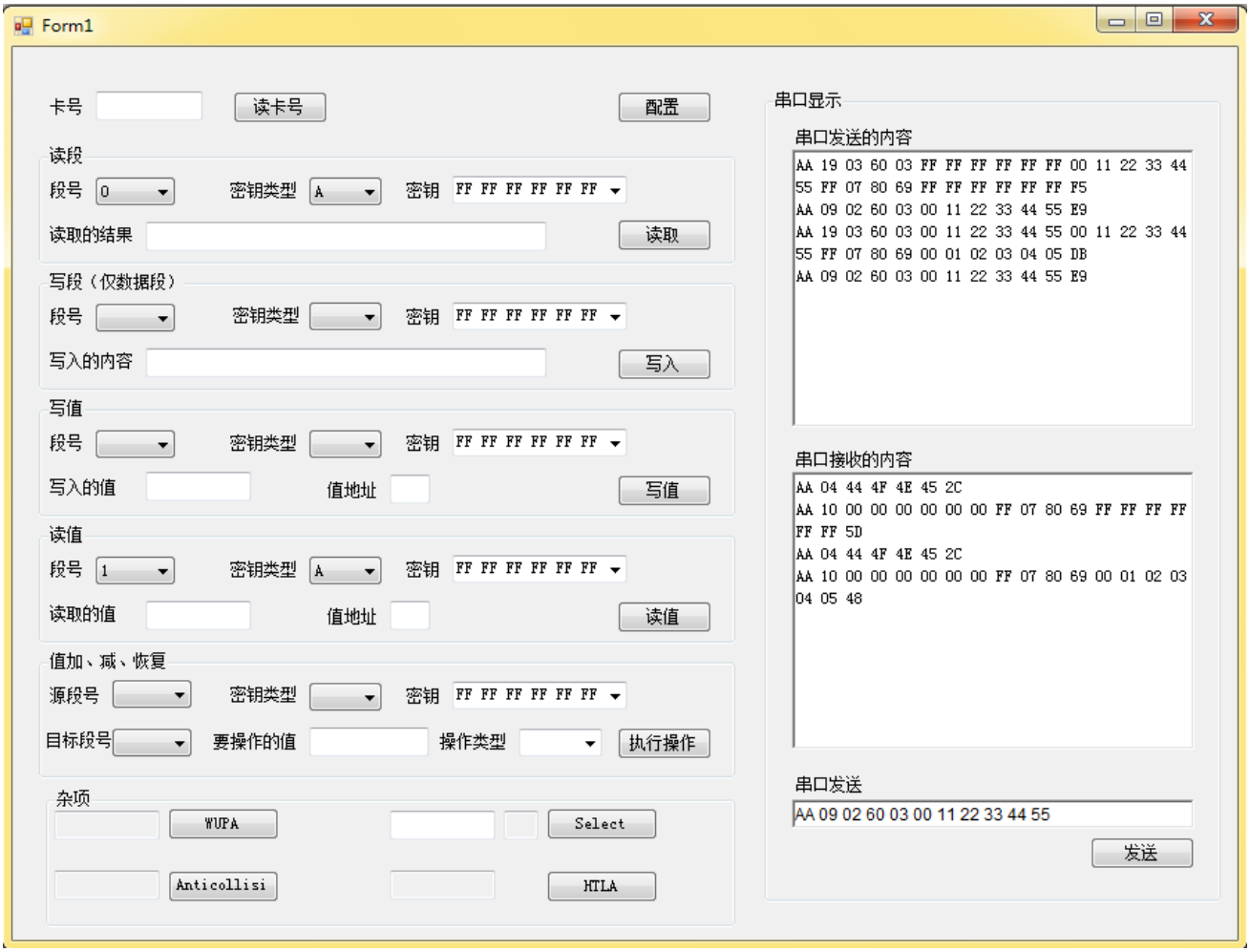

图 51 发送读卡指令

9、 接下来将第 0 区的区尾的访问控制位修改为“08 77 8F 69”，通过调试软件向读卡器发送写存储器段指令（十六进制格式：AA 19 03 60 03 00 11 22 33 44 55 00 11 22 33 44 55 08 77 8F 69 00 01 02 03 04 05 53，其中 03 为第 0 区的区尾的段号，60 表示使用密钥 A 进行认证，这里应该使用修改后的第 0 区的密钥 A 的值 00 11 22 33 44 55，00 11 22 33 44 55 表示密钥 A 的值 ,访问控制条件修改为 08 77 8F 69 ,密钥 B 的值 00 01 02 03 04 05 保持不变，53 为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。根据之前所述，访问控制位“08 77 8F 69”，表示尾区即段 3,$C1_3C2_3C3_3$=011；段 2,$C1_2C2_2C3_2$=110；段 1,$C1_1C2_1C3_1$=110；段 0,$C1_0C2_0C3_0$=110。如图 52 所示。

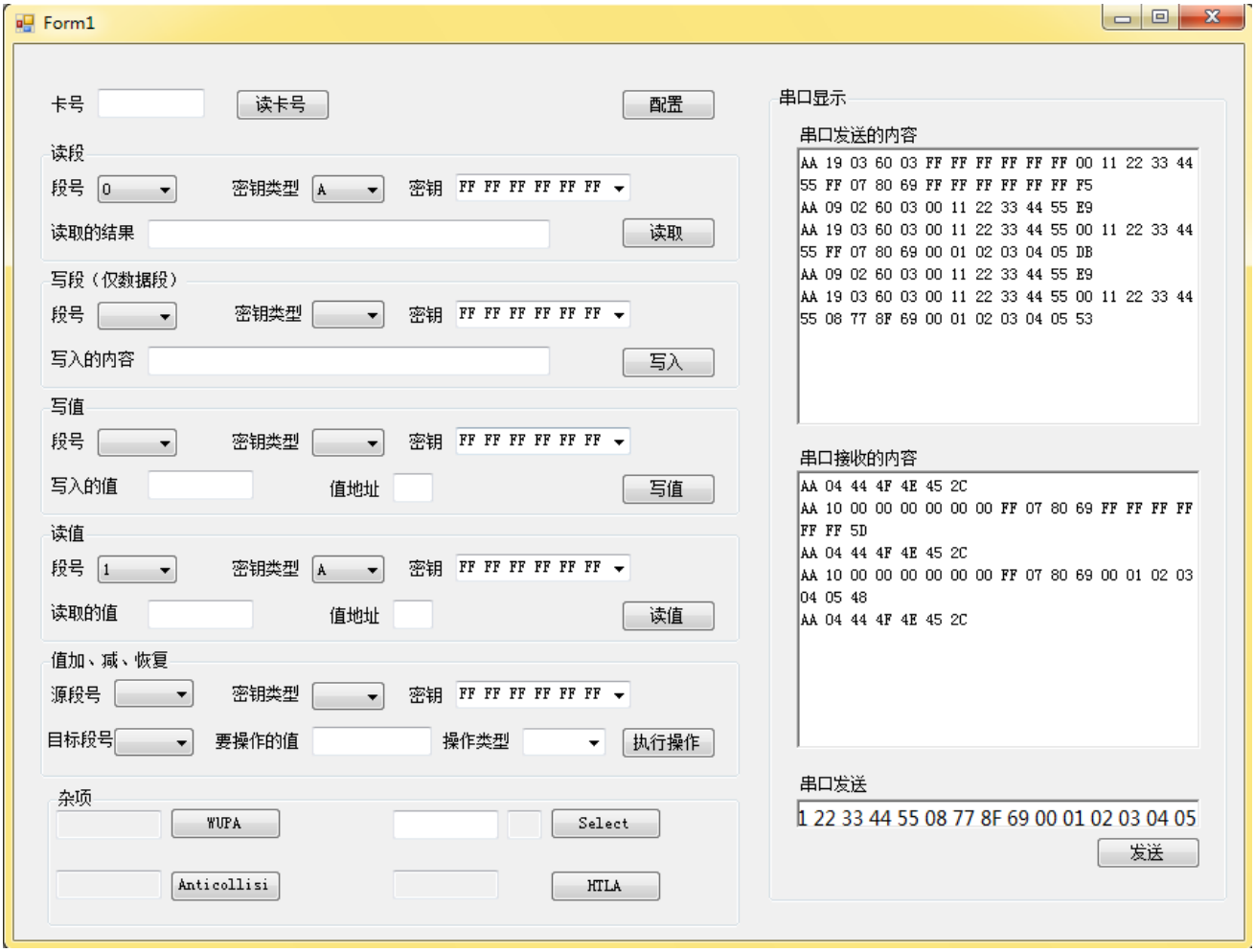

图 52 发送写卡指令

10、 由于访问控制位修改后 $C1_3C2_3C3_3$=011，所以如果要读第 0 区的密钥 A 或者密钥 B 都是不可能的，使用密钥 A”00 11 22 33 44 55”或者密钥 B 来读取第 0 区的区尾的访问控制位的值；使用密钥 B 可以改写访问控制位的值，接下来使用密钥 B 来读访问控制位的值，则通过调试软件向读卡器发送读存储器段指令(十六进制格式 :AA 09 02 61 03 00 01 02 03 04 05 D8，其中 03 为第 0 区的区尾的段号，61 表示使用密钥 B 进行认证，这里应该使用修改后的第 0 区的密钥 B 的值 00 01 02 03 04 05，D8 为校验字节)，若指令执行成功，则读卡器返回第 0 区区尾数据（十六进制格式：AA 10 00 00 00 00 00 00 08 77 8F 69 00 00 00 00 00 00 CF ）,密钥 A 和密钥 B 的值都为 00 00 00 00 00 00 是因为按照访问控制位的规定密钥 A 和密钥 B 的值是不能被读出的，访问控制位的值可以被读出为 08 77 8F 69；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A ）。如图 53 所示。

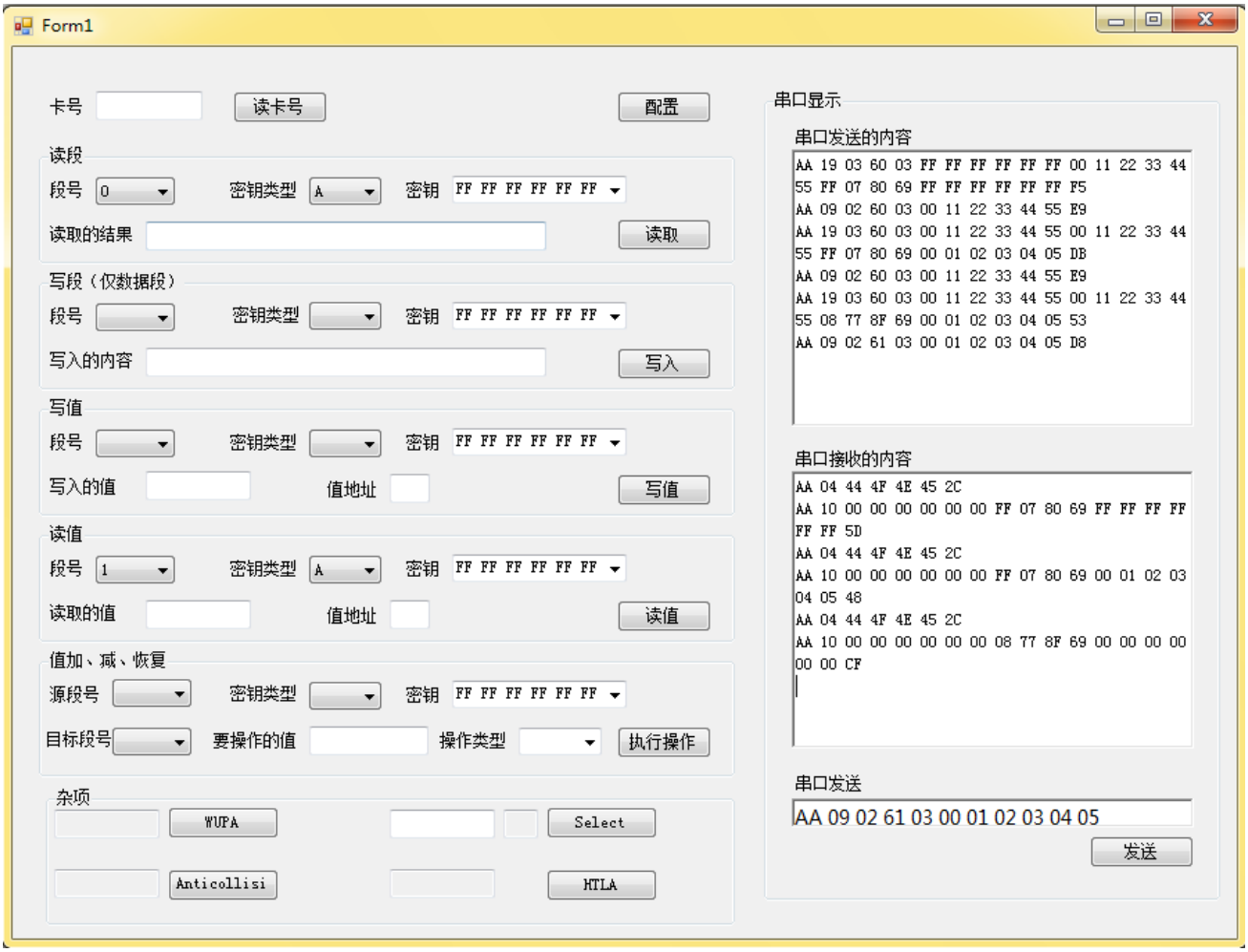

图 53 发送读卡指令

11、 由于访问控制位修改后 $C1_1C2_1C3_1=110$，所以如果要向第 0 区的段 1 写数据，需要认证密钥 B，通过调试软件向读卡器发送写存储器段指令（十六进制格式：AA 19 03 61 01 00 01 02 03 04 05 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF D1，其中 61 表示使用密钥 B 进行认证，这里应该使用修改后的第 0 区的密钥 A 的值 00 01 02 03 04 55，01 为第 0 区段 1，00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 表示要写的数据，D1 为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 54 所示。

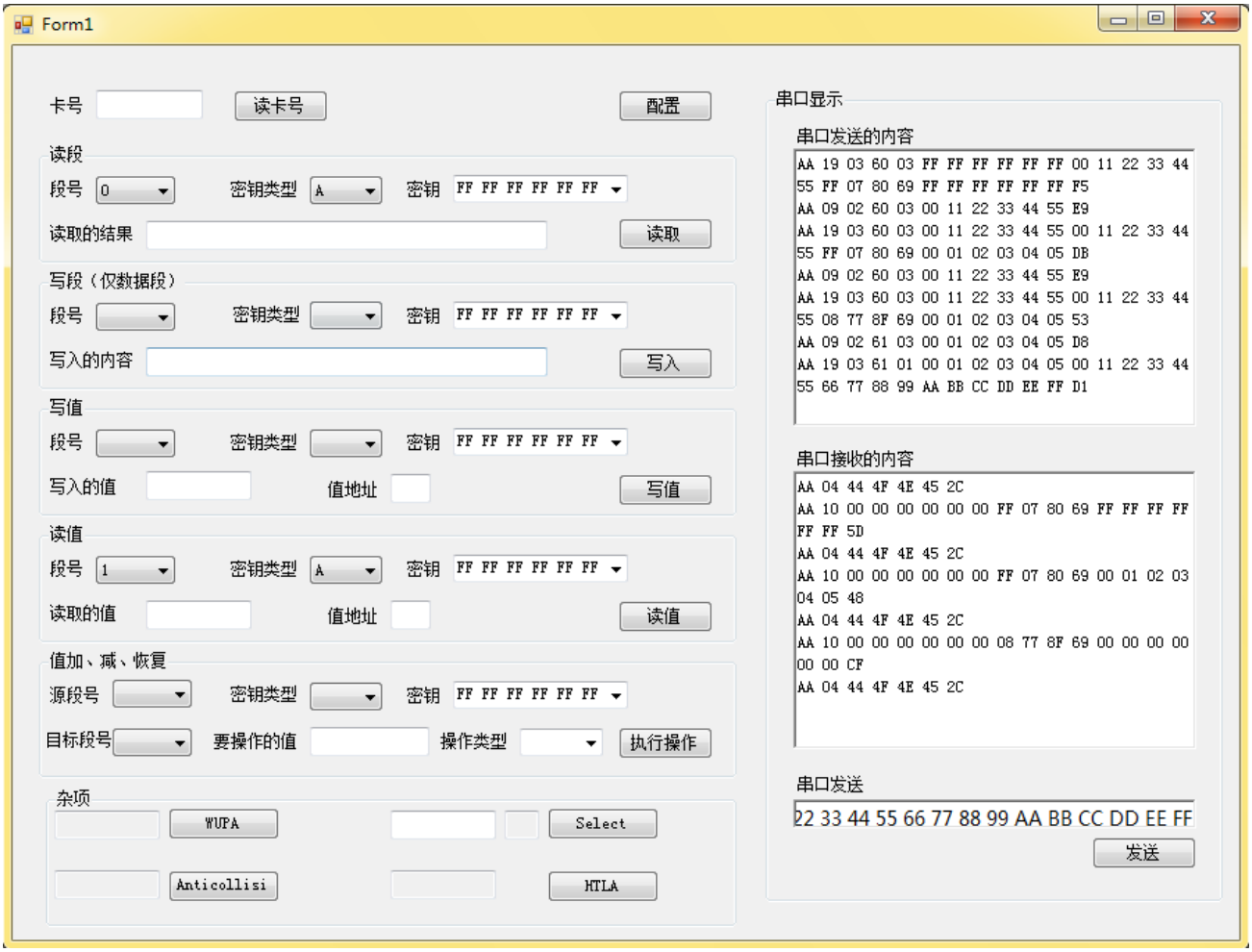

图 54 发送写卡指令

12、 然后使用密钥 B 读取第 0 区的段 1 的数据，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 01 00 01 02 03 04 05 DA，其中 01 表示第 0 区段 1，61 表示使用密钥 B 进行认证，密钥 B 的值为 00 01 02 03 04 05，DA 为校验字节），若指令执行成功，则读卡器返回第 0 区区尾数据（十六进制格式：AA 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 4E）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 55 所示。

图 55 发送读卡指令

13、 尝试将第 1 区的尾区的密钥 A 修改为“FF EE DD CC BB AA”,密钥 B 修改为“55 44 33 22 11 00”，段 2，段 1，段 0 的访问控制位修改为 011，区尾的访问控制位修改为 011，并向第 1 区的段 0 中写入”00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F”的数据并读出。

第六章　实验四 Mifare S50 卡片值段操作实验

6.1　实验目的

1、　理解 Mifare S50 卡片的值段结构

2、　将数据段配置成值段，并实现加值、减值、恢复和传送等值段相关操作

6.2　实验设备

1、　基于 ISO14443A 协议的 13.56MHz 高频读卡器

2、　NXP Mifare S50 卡片

6.3　实验内容

1、　通过调试软件向读卡器发送指令将数据段配置为值段

2、　通过调试软件向读卡器发送指令来对值段进行初始化

3、　通过调试软件向读卡器发送指令实现对值段的加值、减值、恢复等操作

6.4　实验步骤

1、　将读卡器通过 USB 连接线连接到 PC 机的 USB 口。

2、　打开 PC 机上所安装的调试软件。如图 56 所示。

图 56 调试软件

3、 将卡片接近读卡器，此时卡片处于电磁场中，它被激励上电，进而进入 IDLE 状态。

4、 假设要将第 2 区的段 0，段 1，段 2 配置为值段，通过调试软件向读卡器发送写存储器段指令（十六进制格式：AA 19 03 60 0B FF FF FF FF FF FF 00 11 22 33 44 55 08 77 8F 69 00 01 02 03 04 05 50，其中 0B 为第 2 区的区尾的段号，60 表示使用密钥 A 进行认证，全新的卡片的密钥 A 的值为 FF FF FF FF FF FF，00 11 22 33 44 55 的值为要修改的密钥 A 的值，访问控制条件修改为 08 77 8F 69 表示表示尾区即段 3,$C1_3C2_3C3_3$=011 ;段 2,$C1_2C2_2C3_2$=110；段 1,$C1_1C2_1C3_1$=110；段 0,$C1_0C2_0C3_0$=110，KEYB 的值修改为 00 01 02 03 04 05，50 为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 57 所示。

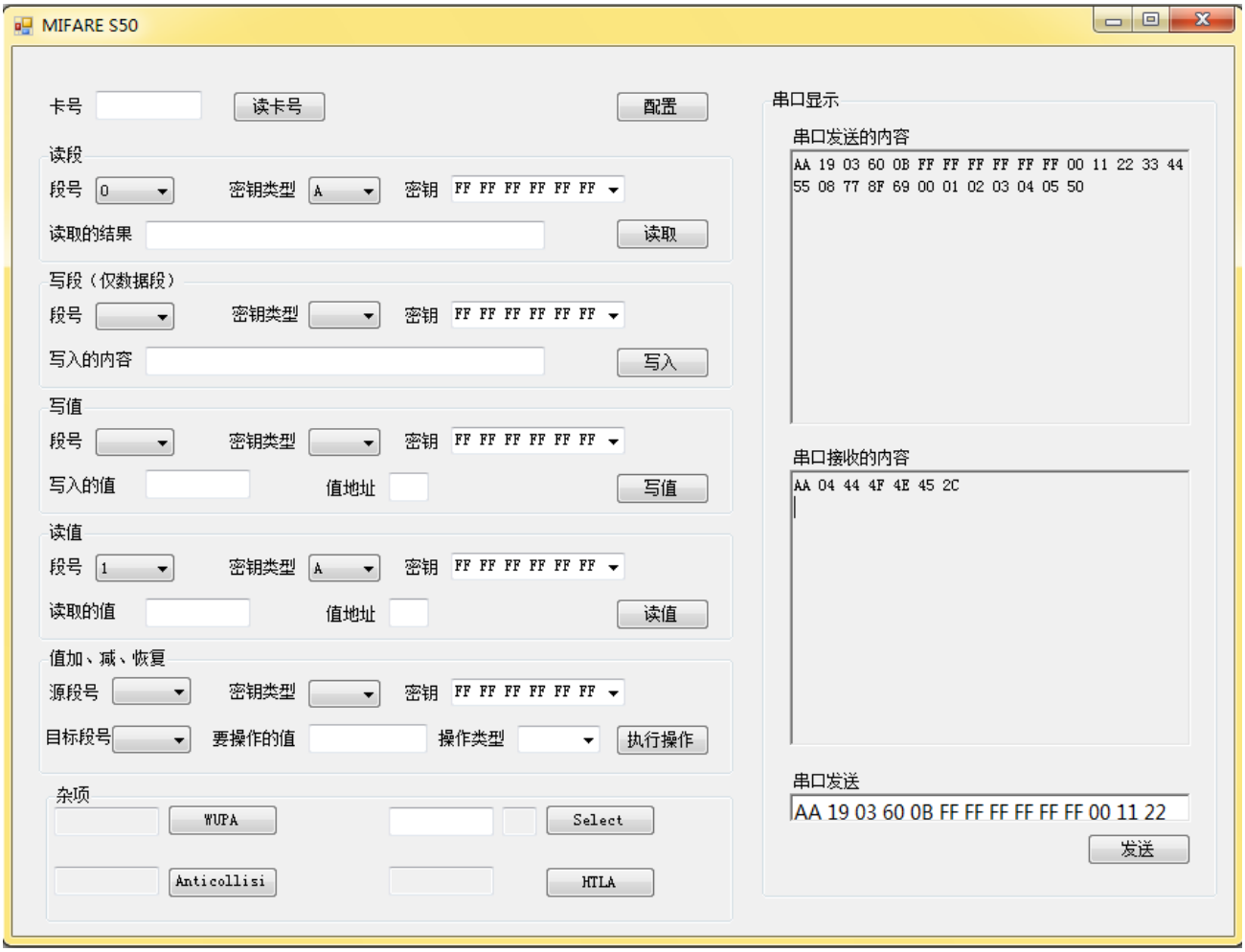

图 57 发送写卡指令

5、 由于访问控制位修改后 $C1_3C2_3C3_3$=011，所以如果要读第 2 区的密钥 A 或者密钥 B 都是不可能的，使用密钥 A”00 11 22 33 44 55”或者密钥 B 来读取第 0 区的区尾的访问控制位的值；使用密钥 B 可以改写访问控制位的值，接下使用密钥 B 来读访问控制位的值，则通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 0B 00 01 02 03 04 05 D0，其中 0B 为第 2 区的区尾的段号，61 表示使用密钥 B 进行认证，这里应该使用修改后的第 2 区的密钥 B 的值 00 01 02 03 04 05，D0 为校验字节），若指令执行成功，则读卡器返回第 2 区区尾数据（十六进制格式：AA 10 00 00 00 00 00 00 08 77 8F 69 00 00 00 00 00 00 CF）,密钥 A 和密钥 B 的值都为 00 00 00 00 00 00 是因为按照访问控制位的规定密钥 A 和密钥 B 的值是不能被读出的，访问控制位的值可以被读出为 08 77 8F 69；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 58 所示。

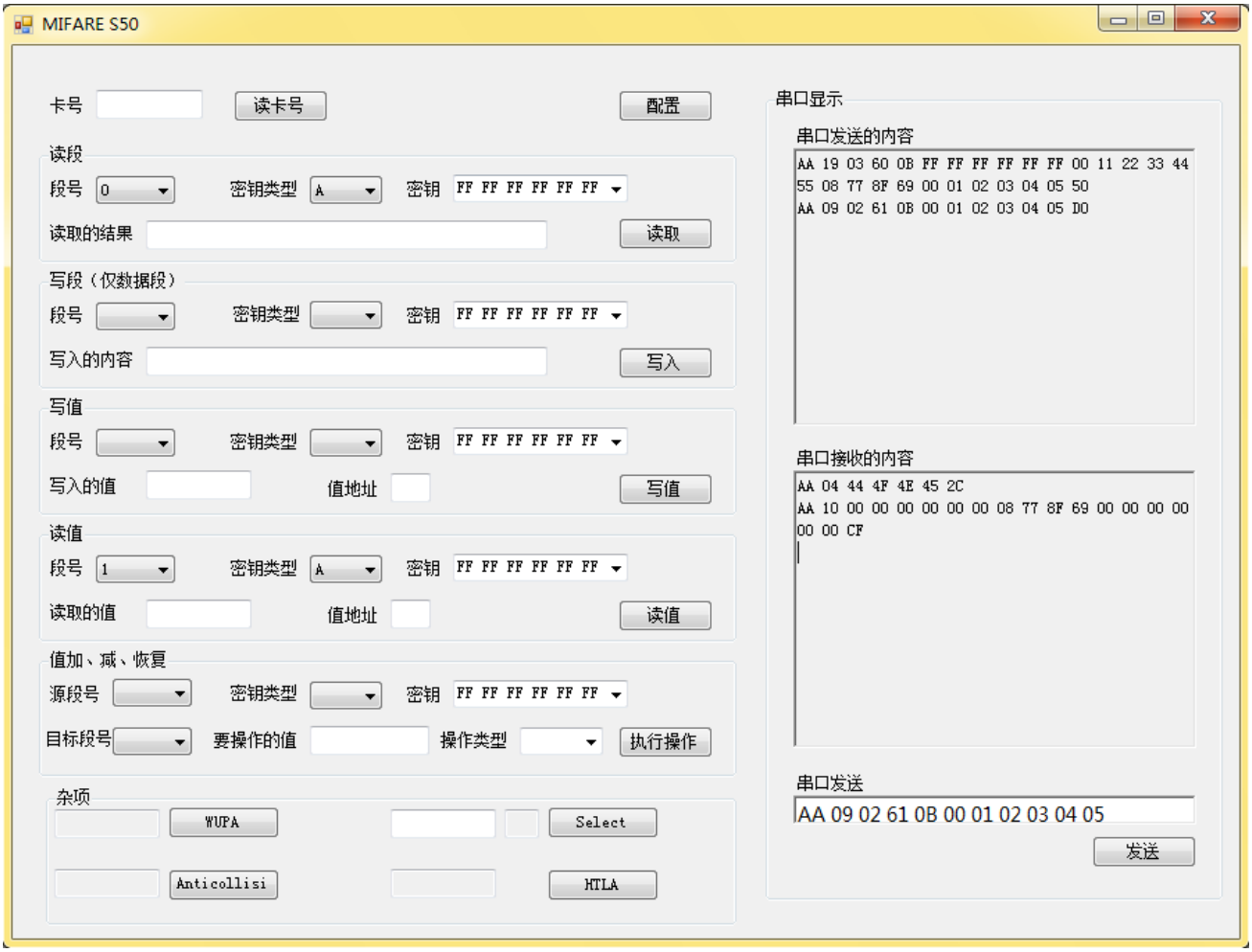

图 58 发送读卡指令

6、 接下来初始化第 2 区的段 0，将由于访问控制位修改后 $C1_1C2_1C3_1=110$，所以如果要向第 0 区的段 1 写数据，需要认证密钥 B，通过调试软件向读卡器发送写存储器段指令（十六进制格式：AA 19 03 61 08 00 01 02 03 04 05 03 00 00 00 FC FF FF FF 03 00 00 00 08 F7 08 F7 C5 ,其中 61 表示使用密钥 B 进行认证 ,这里应该使用修改后的第 2 区的密钥 B 的值 00 01 02 03 04 55，08 为第 2 区段 0，03 00 00 00 表示要写入的值其中低位字节存在最低的地址，FC FF FF FF 表示值取反后的值， 08 为地址，F7 为取反的地址，C5 为校验字节），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 59 所示。

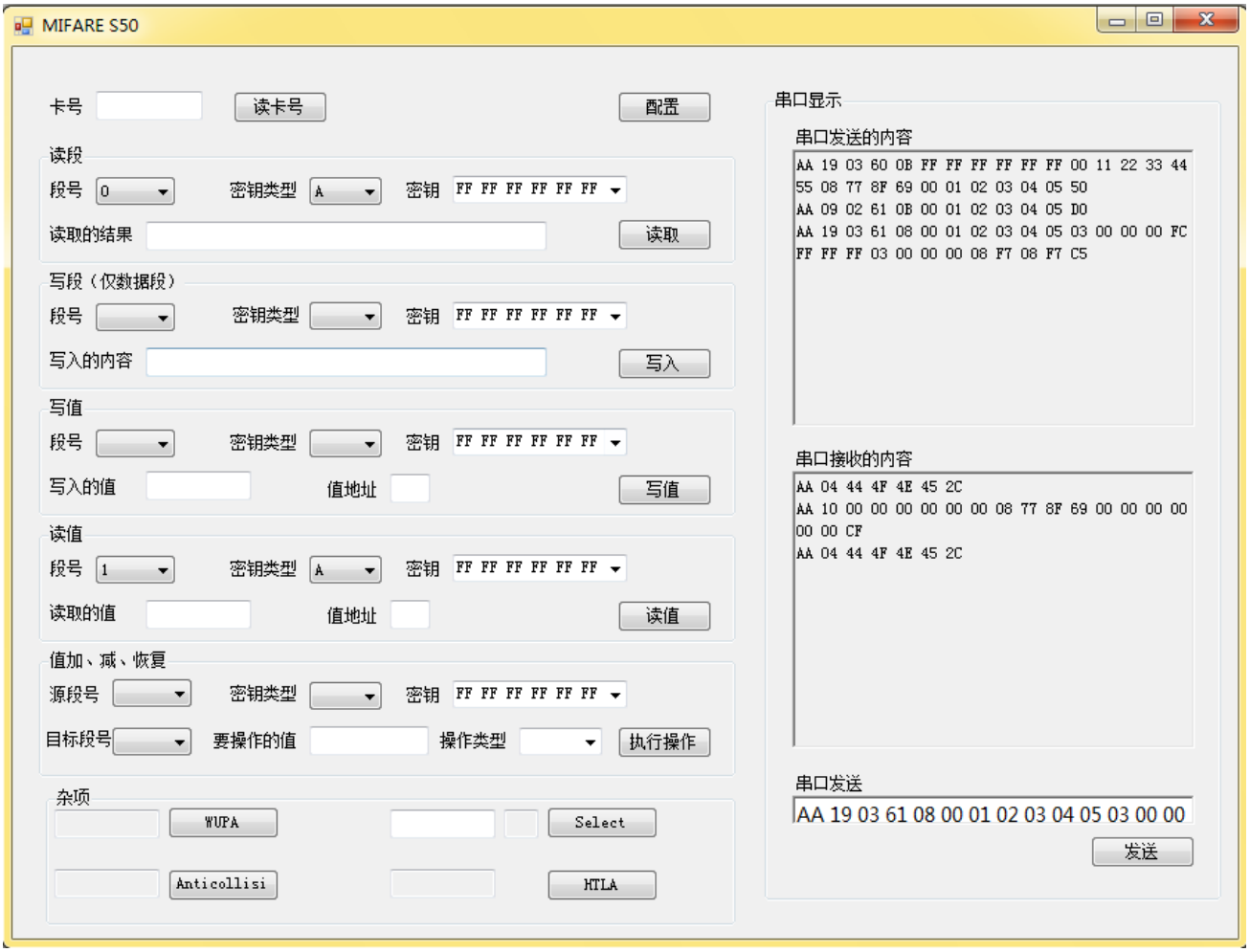

图 59 发送写卡指令

7、 然后使用密钥 B 读取第 2 区的段 0 的数据，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 08 00 01 02 03 04 05 D3，其中 08 表示第 2 区段 0，61 表示使用密钥 B 进行认证，密钥 B 的值为 00 01 02 03 04 05，D3 为校验字节），若指令执行成功，则读卡器返回第 2 区段 0 数据（十六进制格式：AA 10 03 00 00 00 FC FF FF FF 03 00 00 00 08 F7 08 F7 49）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 60 所示。

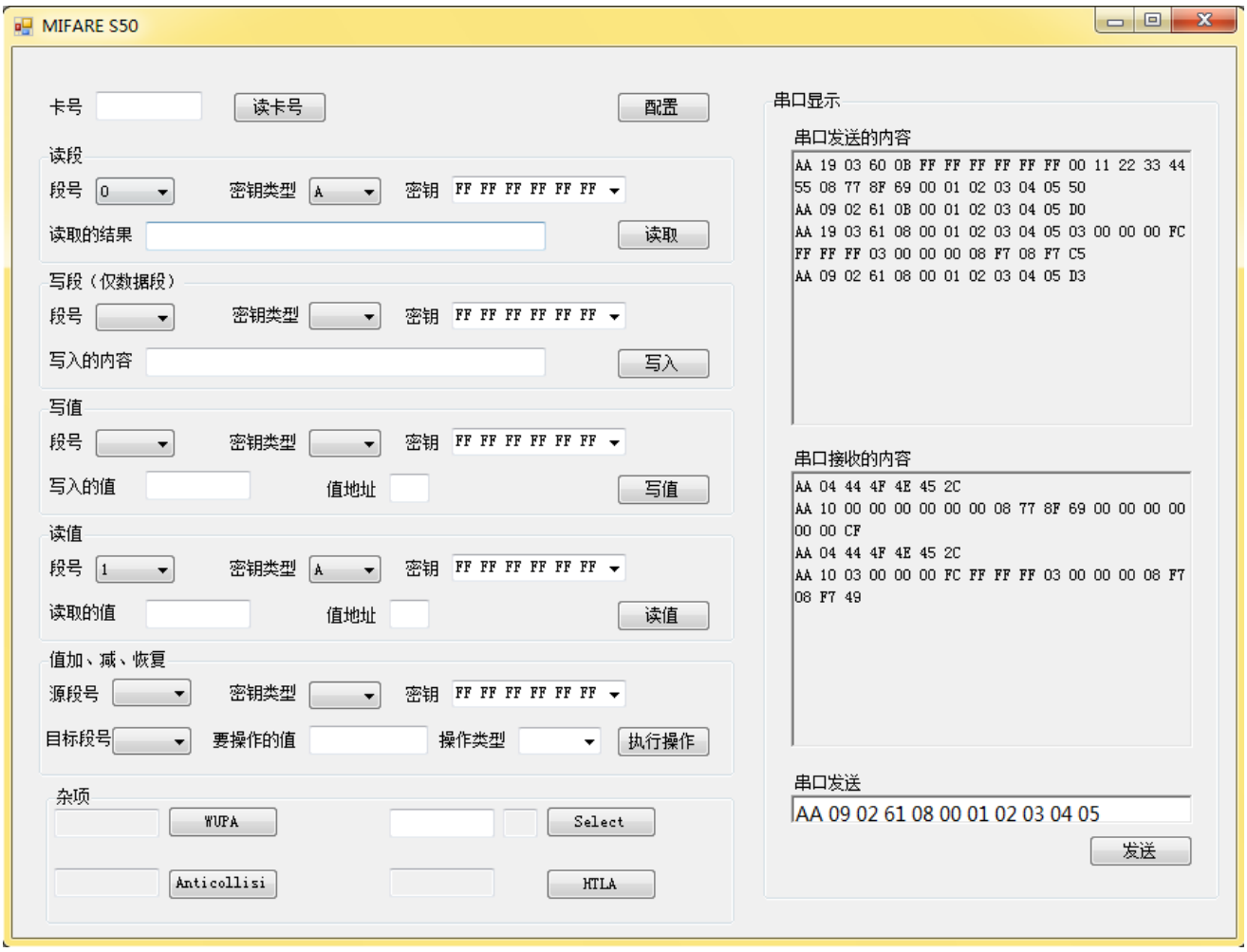

图 60 发送读卡指令

8、 接下来对第 2 区的段 0 进行加值操作，只能使用密钥 B 进行认证，通过调试软件向读卡器发送加值指令（十六进制格式：AA 0E 04 61 08 00 01 02 03 04 05 03 00 00 00 08 C1，其中 61 表示使用密钥 B，其值为 00 01 02 03 04 05；第一个 08 表示要将第 2 区段 0 的值进行相加；03 00 00 00 表示要加的值，第二个 08 表示将加后的结果存入第 2 区段 0 中，即传送指令），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 61 所示。

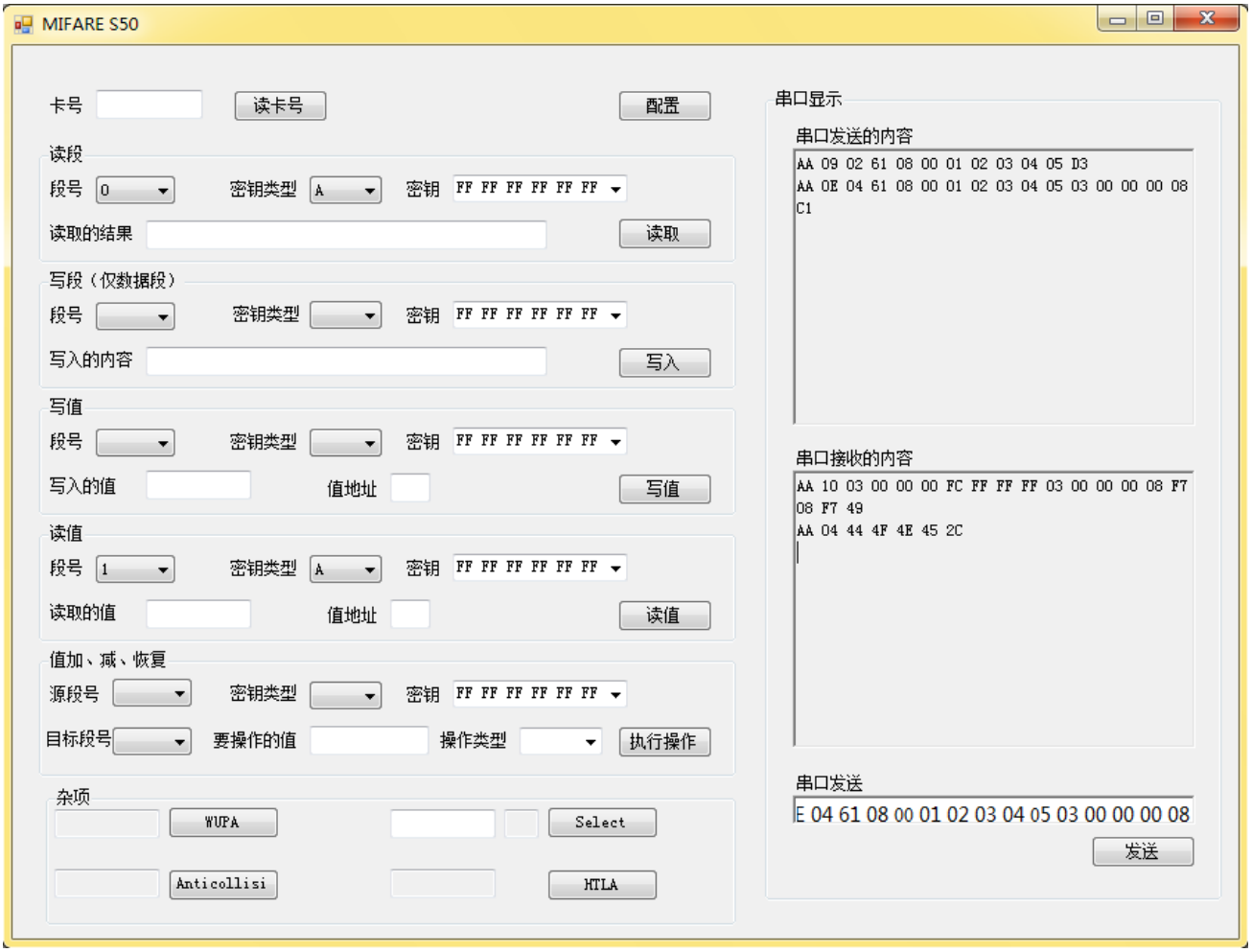

图 61 发送加值指令

9、 然后使用密钥 B 读取第 2 区的段 0 的数据，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 08 00 01 02 03 04 05 D3，其中 08 表示第 2 区段 0，61 表示使用密钥 B 进行认证，密钥 B 的值为 00 01 02 03 04 05，D3 为校验字节），若指令执行成功，则读卡器返回第 2 区段 0 数据（十六进制格式：AA 10 06 00 00 00 F9 FF FF FF 06 00 00 00 08 F7 08 F7 46），说明加值操作执行成功；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 62 所示。

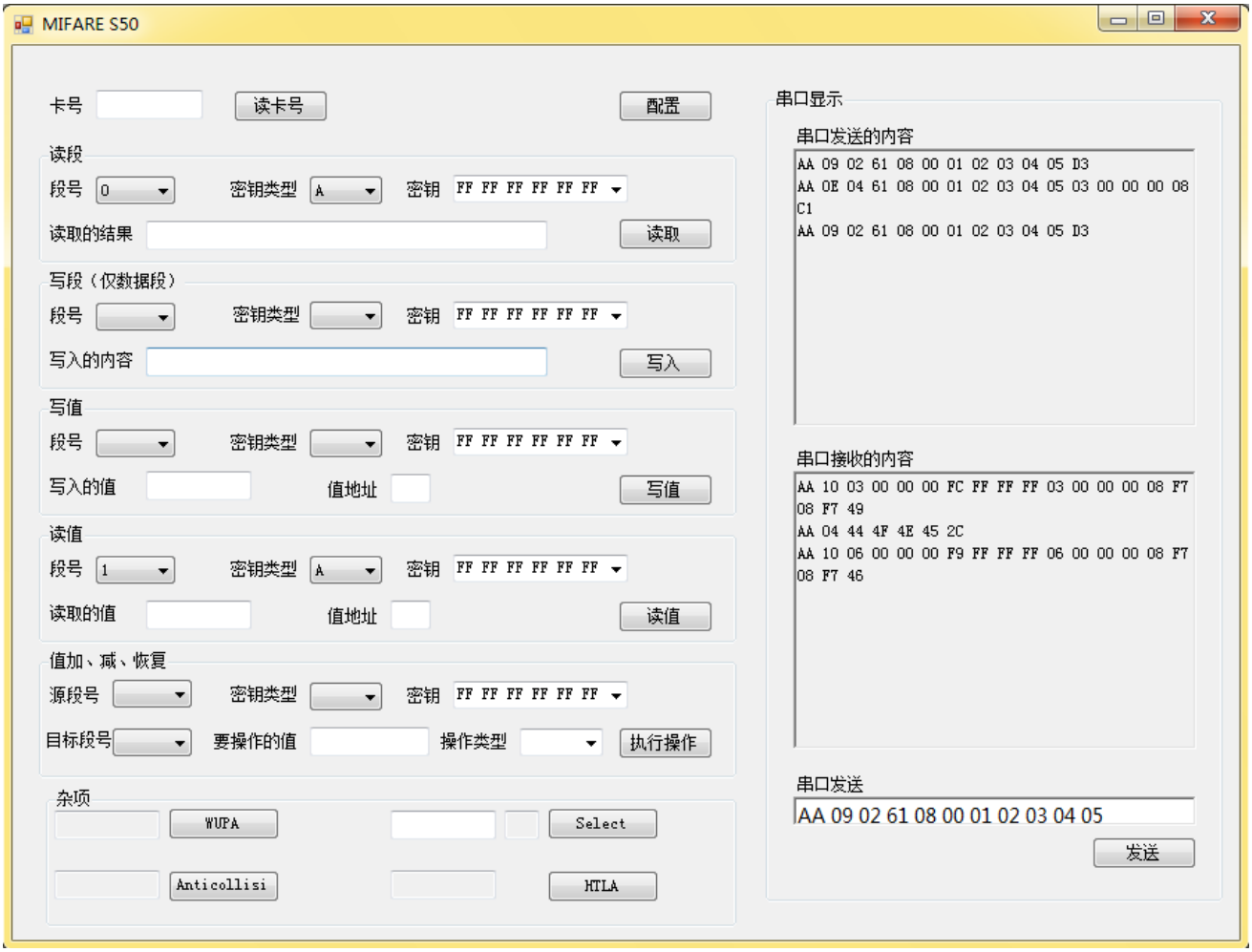

图 62 发送读卡指令

10、 接下来对第 2 区的段 0 进行减值操作，使用密钥 B 进行认证，通过调试软件向读卡器发送加值指令（十六进制格式：AA 0E 05 61 08 00 01 02 03 04 05 03 00 00 00 08 C0，其中 61 表示使用密钥 B ,其值为 00 01 02 03 04 05 ;第一个 08 表示要讲第 2 区段 0 的值进行相减；03 00 00 00 表示要减的值，第二个 08 表示将减后的结果存入第 2 区段 0 中，即传送指令），若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C）；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 63 所示。

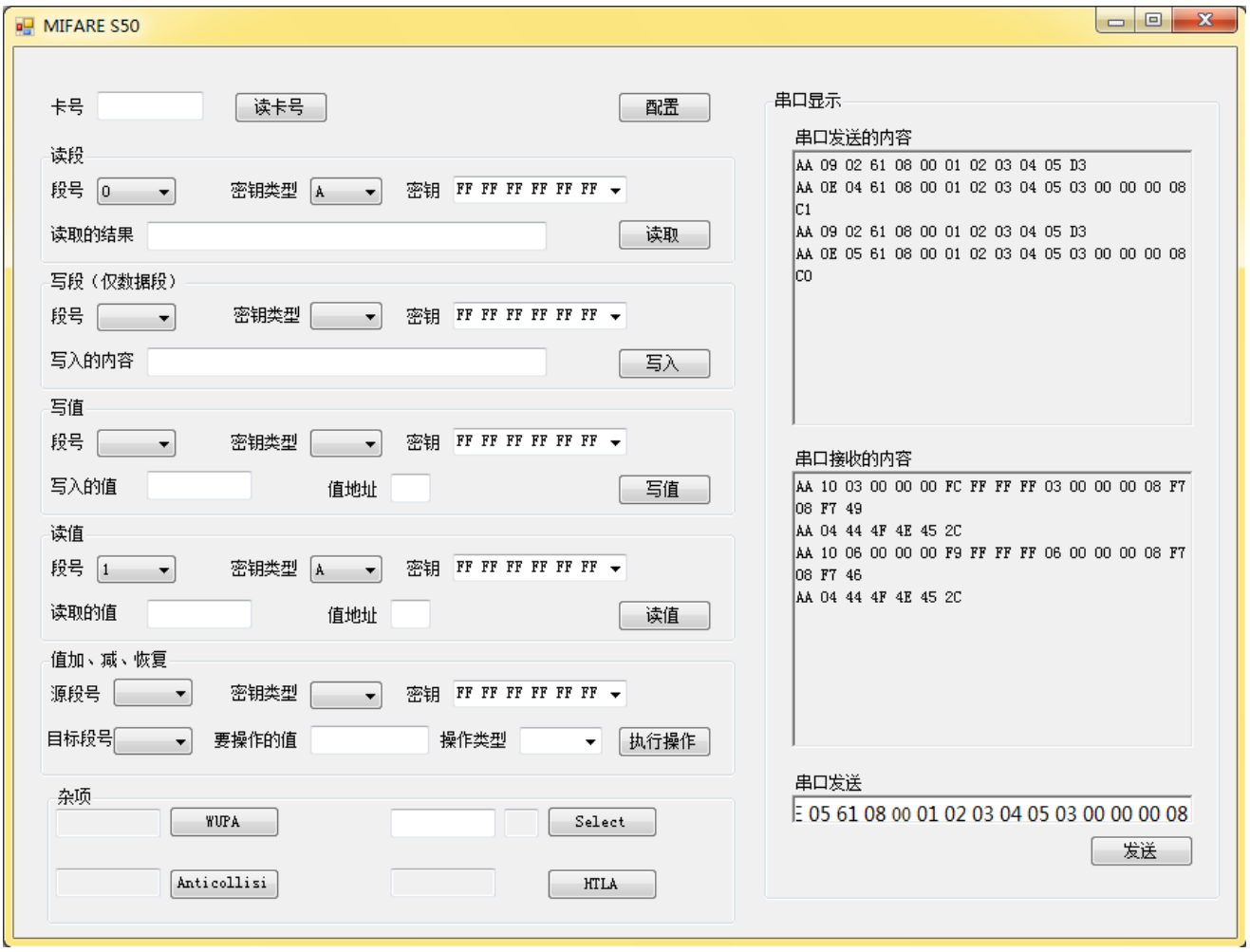

图 63 发送减值指令

11、 然后使用密钥 B 读取第 2 区的段 0 的数据，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 08 00 01 02 03 04 05 D3，其中 08 表示第 2 区段 0，61 表示使用密钥 B 进行认证，密钥 B 的值为 00 01 02 03 04 05，D3 为校验字节），若指令执行成功，则读卡器返回第 2 区段 0 数据（十六进制格式：AA 10 03 00 00 00 FC FF FF FF 03 00 00 00 08 F7 08 F7 49），说明减值操作执行成功；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 64 所示。

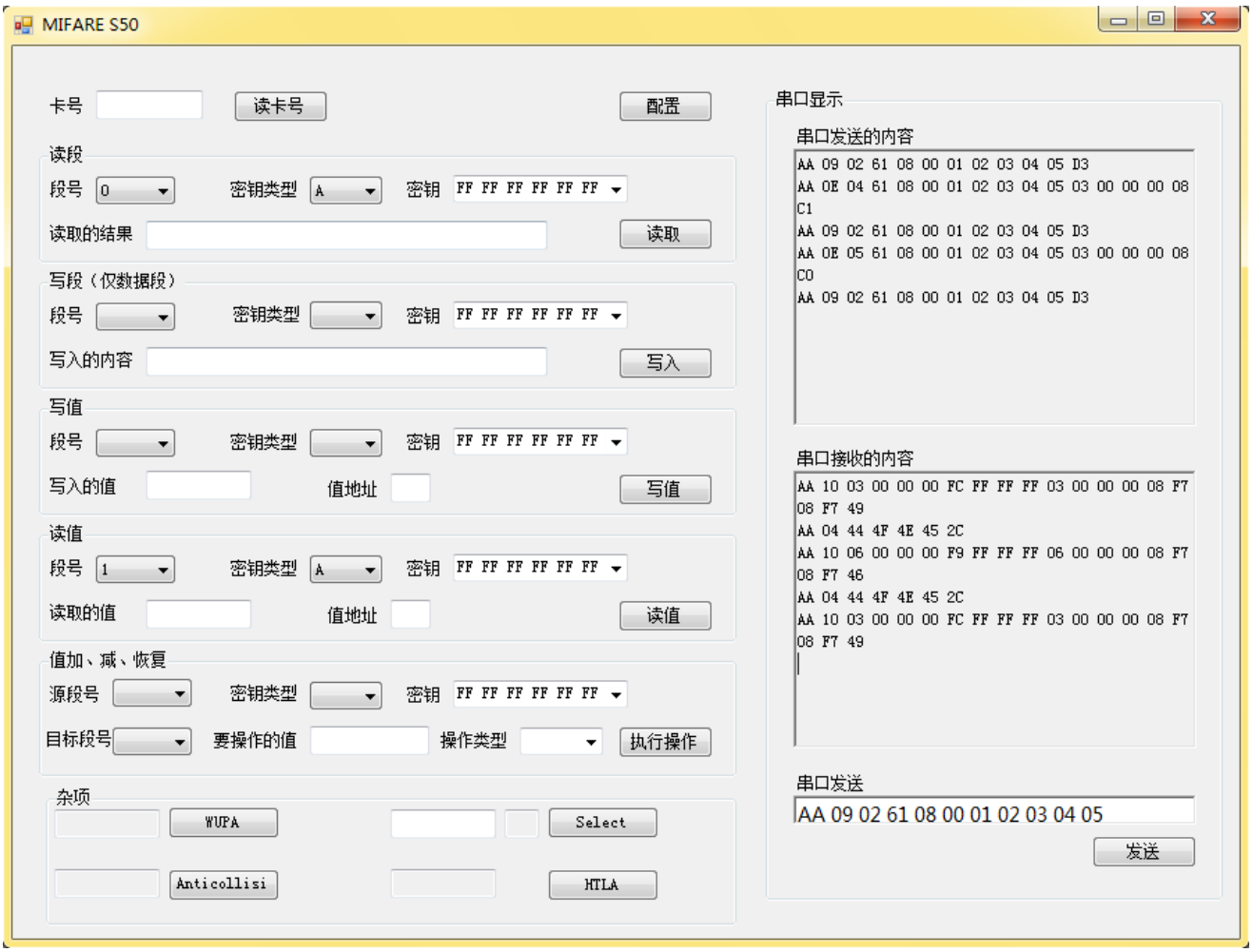

图 64 发送读卡指令

12、 接下来对第 2 区的段 0 进行恢复操作，并将值存入段 1 中，使用密钥 B 进行认证，通过调试软件向读卡器发送加值指令(十六进制格式 :AA 0E 06 61 08 00 01 02 03 04 05 00 00 00 00 09 C1，其中 61 表示使用密钥 B，其值为 00 01 02 03 04 05；08 表示要讲第 2 区段 0 的值进行恢复操作；00 00 00 00 表示恢复操作的值不应改变，09 表示将从第 2 区段 0 中恢复出来的值的存入第 2 区段 1 中，即传送指令)，若指令执行成功，则读卡器返回成功指令（十六进制格式：AA 04 44 4F 4E 45 2C)；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A)。如图 65 所示。

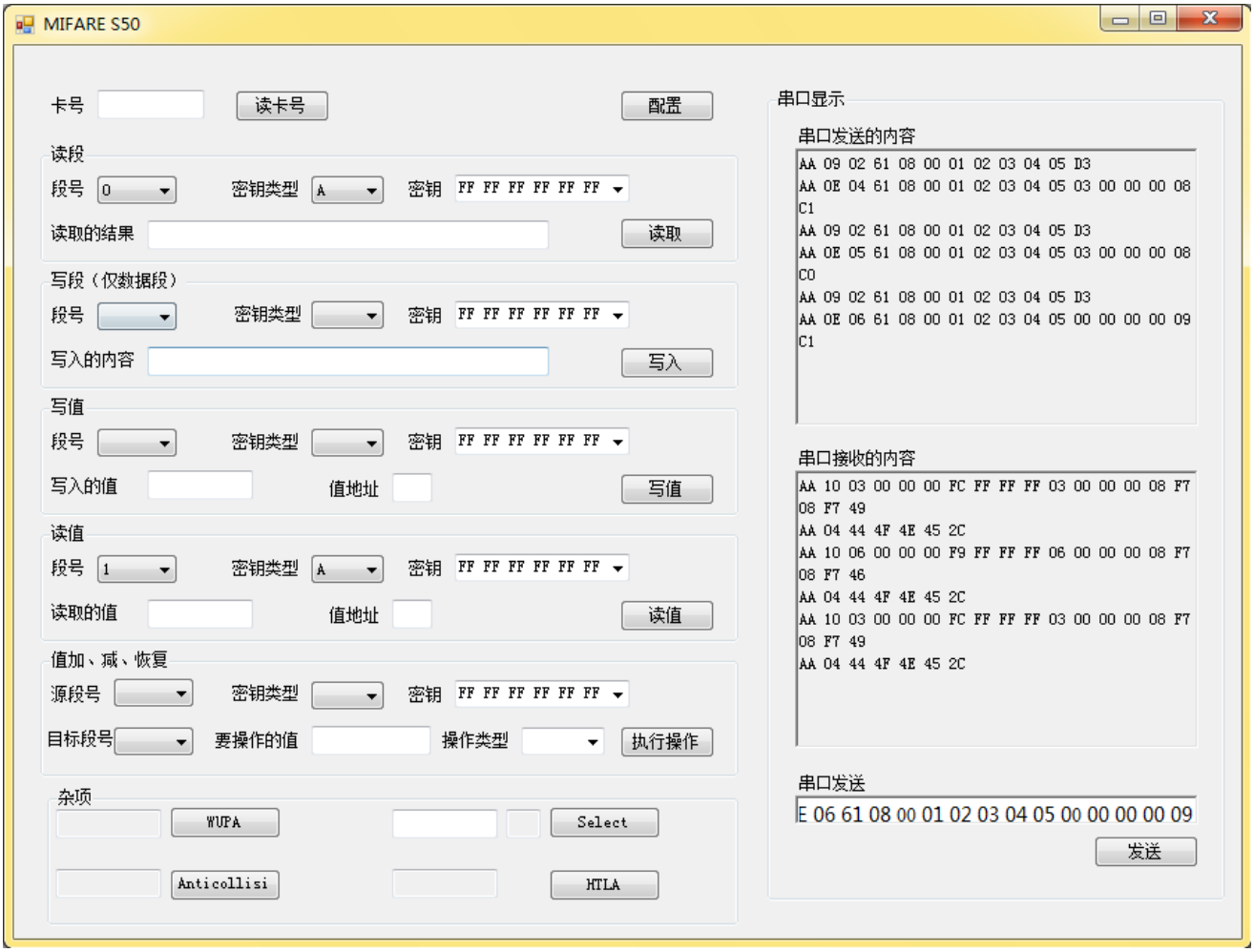

图 65 发送恢复指令

13、 然后使用密钥 B 读取第 2 区的段 1 的数据，通过调试软件向读卡器发送读存储器段指令（十六进制格式：AA 09 02 61 09 00 01 02 03 04 05 D2，其中 09 表示第 2 区段 1，61 表示使用密钥 B 进行认证，密钥 B 的值为 00 01 02 03 04 05，D3 为校验字节），若指令执行成功，则读卡器返回第 2 区段 0 数据（十六进制格式：AA 10 03 00 00 00 FC FF FF FF 03 00 00 00 08 F7 08 F7 49），说明恢复操作执行成功；若指令执行不成功，则读卡器返回错误指令（十六进制格式：AA 03 45 52 52 6A）。如图 66 所示。

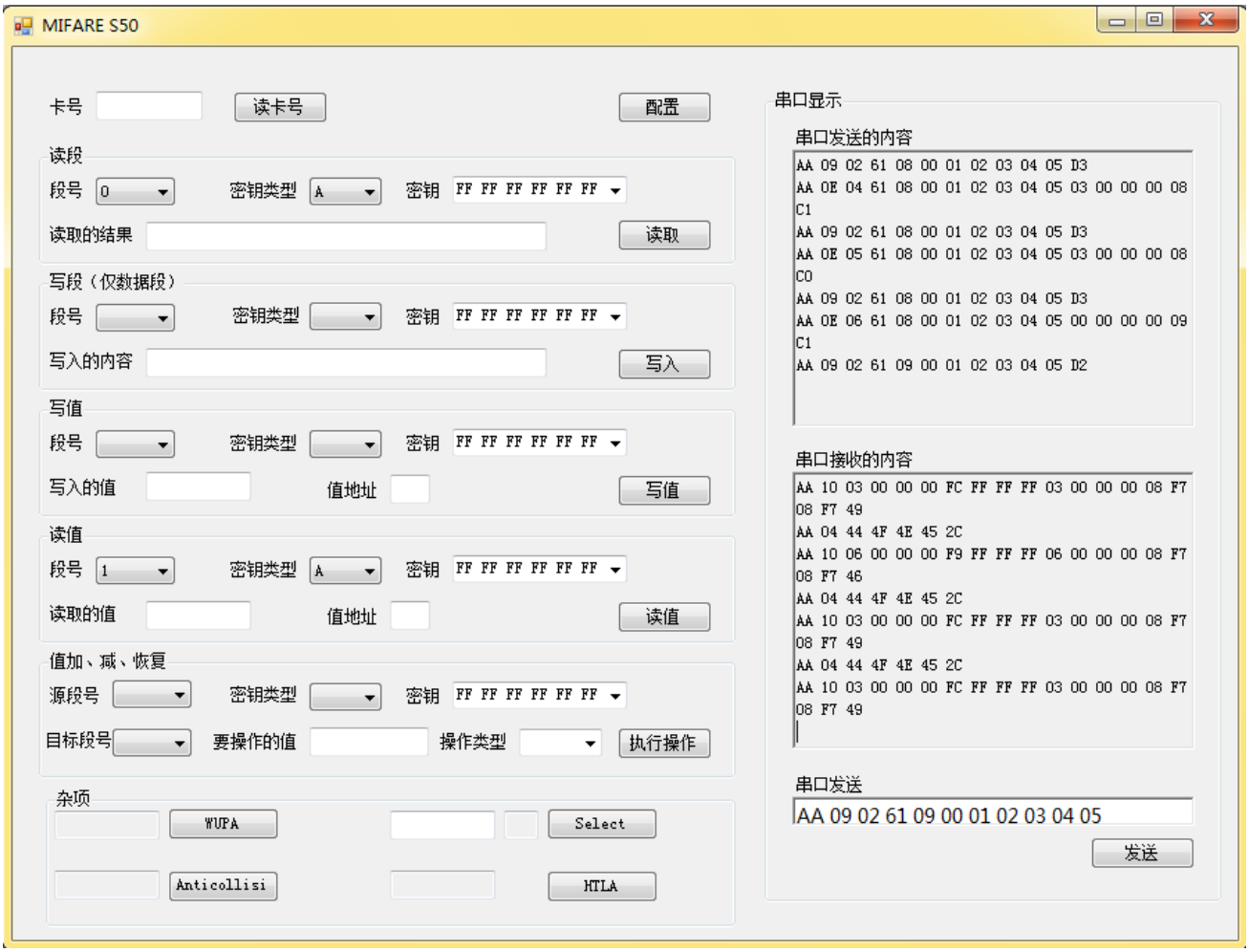

图 66 发送读卡指令

14、 把第 3 区段 0，段 1，段 2 配置成值段，并写入初始值并进行加值，减值和恢复操作，由于值是有符号的，观察当被减数小于减数时，得到的值是否正确，当加值产生溢出时的值会有什么改变。