



## 关键词

- TX522D 读卡模块
- Mifare 1 S50, S70
- 扇区 密钥
- 一卡多用 一卡通
- 非接触式 智能卡
- 自动寻卡

## 摘要

TX522 系列模块是基于 13.56MHz 频率的 Mifare 卡读写模块,符合 ISO14443A 标准,可支持 Mifare1 S50、Mifare1 S70、Mifare Light、Mifare UltraLight、Mifare Pro。TX522 系列 Mifare 读写模块具有易用、高可靠、多种接口、体积小等特点,可帮助用户方便、快捷地将当今最流行的非接触式 IC 卡技术融入到系统中,提高产品的档次。

TX522 能够配置为自动寻卡模块,上电后无需向模块发送任何命令,只要有卡靠近模块就能主动通过通信接口发送卡号。

本文详细介绍了 TX522D 模块的基于 51 单片机的 C51 库函数的使用方法。

## 目 录

1. 适用范围.....	4
2. TX522D 简介 .....	5
2.1 引脚描述.....	5
2.2 典型电路.....	6
2.3 技术参数.....	6
2.4 极限参数.....	6
2.5 直流特性.....	7
2.6 封装及机械尺寸.....	7
3. TX522D 读卡模块数据传输协议.....	9
3.1 I2C 协议.....	9
3.2 协议描述.....	9
4. TX522D 的 C51 函数.....	11
4.1 函数列表.....	11
4.2 函数返回状态值列表.....	11
5. 函数描述.....	13
5.1 设备控制类命令 (CmdType = 0x01) .....	13
5.1.1 获取信息—Get Info (Cmd = ‘A’) .....	13
5.1.2 配置—Config (Cmd = ‘B’) .....	14
5.1.3 关闭—Close (Cmd = ‘C’) .....	15
5.1.4 配置—ConfigAnt.....	15
5.1.5 装载密钥—Load_Key (Cmd = ‘E’) .....	16
5.1.6 清除控制位—Clr_Control_Bit (Cmd = ‘I’) .....	17
5.1.7 置位控制位—Set_Control_Bit (Cmd = ‘J’) .....	18
5.1.8 输出蜂鸣器信号—Buzzer (Cmd = ‘K’) .....	18
5.2 ISO14443A 类命令 (CmdType = 0x02) .....	19
5.2.1 激活卡片并获取卡号—Get_CardSnr (Cmd = ‘M’) .....	19
5.2.2 证实 2—Auth2 (Cmd = ‘E’) .....	21
5.2.3 直接密码证实—Auth_Key (Cmd = ‘F’) .....	22
5.2.4 读—Read (Cmd = ‘G’) .....	23
5.2.5 写—Write (Cmd = ‘H’) .....	24
5.2.6 暂停—Halt (Cmd = ‘D’) .....	25
5.2.7 带内部自动传送的值操作—Value (Cmd = ‘J’) .....	26

5.2.8 复位—Reset (Cmd = 'L')	27
5.2.9 自动寻卡—Auto_Detect (Cmd = 'N')	28
5.2.10 读取自动检测数据—Read_ADDT	错误! 未定义书签。
5.2.11 带验证的写—Write_Auth (Cmd = 'P')	30
5.2.12 带验证的读—Read_Auth (Cmd = 'Q')	31
5.2.13 带验证的值块操作—Value_Auth (Cmd = 'R')	32
5.2.14 带验证的写值块操作—WriteValue_Auth (Cmd = 'S')	33
5.2.15 带验证的读值块操作—ReadValue_Auth (Cmd = 'T')	35
5.3 函数调用描述	36
6. 免责声明	38
7. 修订历史	39
8. 销售信息	39

## 1. 适用范围

本文对 TX522D Mifare 卡读写模块在 51 单片机下的 C51 库函数做了非常详细的讲解，使用 51 系列单片机与 TX522D 进行通讯的用户参考本文提供的各函数，可以非常容易的编写操作 Mifare 卡的应用程序。不使用 51 系列单片机，而使用其它处理器的用户，也可以参考本文所描述提供的通讯格式和函数，将本文提供的函数很容易的移植到其它处理器上。

对于 C51 函数的调用，用户可不必关心数据块格式，只要理解函数的功能，输入、输出参数即可。当用户自己编写函数（包括非 C51 下的函数）时，就需了解数据块格式，必须按照数据块的格式来编写函数。

## 2. TX522D 简介

TX522D 是一个简单的 I2C 接口读写模块，用于与常用的微处理器（如单片机、ARM）接口。

用户必须通过主机(包括单片机、ARM、DSP 等)向 TX522D 模块发送命令来对 TX522D 进行读写控制。本应用指南将重点描述 TX522D 与主机之间的串行通信协议和命令。

### 2.1 引脚描述

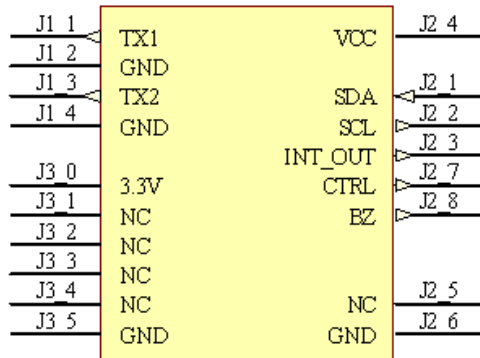


图 1 TX522 系列模块管脚图

表 1 外接天线接口 J1

接 口	管 脚	符 号	IO 类型	功能描述
J1 <sup>(1)</sup>	J1-1	TX1	输出/O	天线发送端 1
	J1-2	GND	地/Power	地
	J1-3	TX2	输出/O	天线发送端 2
	J1-4	GND	地/Power	地

(1) J1 为模块与天线的接口,对于天线一体化（带后缀 T）的模块，如果用户使用模块上的 PCB 印制天线，则可以不使用该接口；对于非天线一体化（不带后缀 T）的模块，用户要通过该接口来连接天线。

表 2 与用户 MCU 接口 J2

接 口	管脚	符 号	IO 类型	功能描述	上电状态
J2 <sup>(1)</sup>	J2-1	SDA	输入/输出	I2C 数据输入/输出,集电极开漏,内部已加 10k 上拉。	1
	J2-2	SCL	输入	I2C 时钟输入,集电极开漏,内部已加 10k 上拉。	1
	J2-3	INT_OUT	输出/O	当模块完成命令时,此管脚输出低电平。当设置为自动寻卡时,有卡时输出低电平	1
	J2-4	VCC	电源/Power	电源正极,+3.5V~+9V	
	J2-5	NC	--	空闲脚	
	J2-6	GND	地/Power	地	
	J2-7	CTRL	输出/O	控制信号输出,集电极开漏	1
	J2-8	BZ	输出/O	蜂鸣器驱动电路控制信号,集电极开漏,需要串电阻	1

(1) J2 为模块与用户控制器的接口。

表 3 用户接口 J3

接口	管脚	符号	IO 类型	功能描述
J3	J3-0	+3.3V	地/Power	3.3V 电源输出，最大提供 70mA 电流
	J3-1	NC	-	预留未来使用
	J3-2	NC	-	预留未来使用
	J3-3	NC	-	预留未来使用
	J3-4	NC	-	预留未来使用
	J3-5	GND	地/Power	地

## 2.2 典型电路

TX522DT 模块可以与任何 MCU 接口，图 2 所示为 TX522DT 与 MCS51 单片机的典型接口。

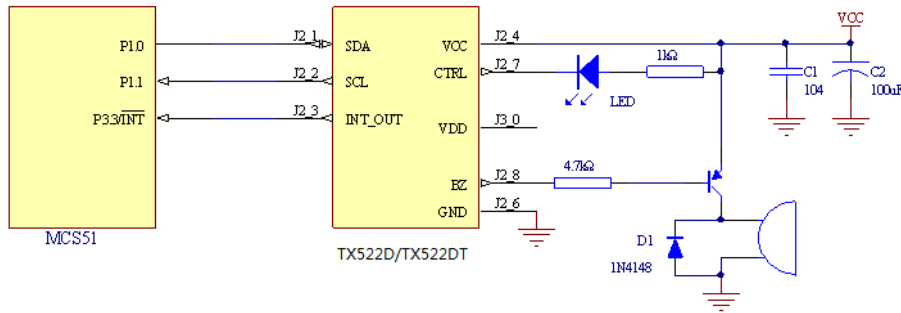


图 2 TX522DT 与 MCS51 单片机接口图

## 2.3 技术参数

表 4 TX522 模块技术参数表

产品型号	TX522 系列
功耗	30 毫安/直流 5V;
工作频率	13.56 兆赫兹
读卡距离	50~100 毫米 (mifare1 卡)
接口方式	I2C
数据传输速率	最大 200kBit/s
支持卡类型	mifare1 S50、mifare1 S70、mifare UltraLight、mifare Pro
尺寸	天线一体化 (带后缀 T): 58mm×34.5mm×2.8mm

## 2.4 极限参数

每个管脚的对地电压 .....-0.5~+5.5V  
 Vcc 对地的电压 .....-0.3~+7.0V  
 每个管脚的最大 I<sub>OL</sub> .....20mA  
 湿度 (相对湿度) .....5%~95

超出“绝对最大额定值”列出的值的条件下工作会造成器件的永久损坏。以上列出的是器件正常工作的额定值，并未涉及器件在这些条件或超出这些条件下的功能操作。器件不能长时间工作在绝对最大额定值条件下，否则会影响其可靠性。

## 2.5 直流特性

VCC=+5.0V（订货可选+3.3V），器件都工作在建议的温度范围-30~85℃条件下，除非特别说明。

表 5 TX522 模块的直流特性

符号	参数	测试条件	最小值	典型值 <sup>(1)</sup>	最大值	单位
VCC	工作电压		+3.5	+5.0	+24	V
VDD	输出电压			3.3V		V
I <sub>VDD max</sub>	VDD最大输出电流				70	mA
T <sub>OK</sub>	上电后稳定工作时间		5			ms
I <sub>CC</sub>	电流消耗	读卡芯片配置成功		25		mA
V <sub>IL</sub>	输入低电平		0.7	1.3	-	V
V <sub>IH</sub>	输入高电平		-	2	2.3	V
V <sub>OL</sub>	输出低电平	I <sub>OL</sub> =20mA		0.6	1.0	V
		I <sub>OL</sub> =3.2mA		0.2	0.3	V
V <sub>OH</sub>	输出高电平	I <sub>OH</sub> =-20uA	3	3.1		V
I <sub>IL</sub>	逻辑低电平输入电流	V <sub>pin</sub> =0.4V			-80	μA
I <sub>TL</sub>	逻辑1到0变化电流	V <sub>pin</sub> =2V	-30		-450	μA
I <sub>OL</sub>	低电平时的灌电流				-20	mA
I <sub>OH</sub>	高电平时的拉电流				20	μA
C <sub>IO</sub>	管脚输入电容				15	pF
T <sub>OP</sub>	工作温度(I)		-30		+85	℃
T <sub>STR</sub>	存储温度		-55		+125	℃

(1) 典型值是难以保证的，这个值是在常温条件下测试得到。

(2) 模块上电后，必须等待 5ms 以上时间才能稳定工作。

## 2.6 封装及机械尺寸

TX522 模块根据采用天线一体化封装型号：TX522T。封装尺寸如图 3 所示。为保证用户目标板和 TX522 系列模块接口尺寸一致，用户可以在 <http://www.txrfid.com> 上获取 TX522 系列模块元件库和 PCB 封装。





### 3. TX522D 读卡模块数据传输协议

#### 3.1 I2C 协议

TX522DT 通信速率最大为 200kHz，从机地址为 0xB2。只要从机地址正确，速率适当，模块就可以有效接收主机发送的命令。

主机有两种方式来判断模块是否执行完命令。第一种，通过判断 J23(INT\_OUT)是否变成低电平，当模块执行完成命令后 J23(INT\_OUT)变成低电平，主机读取数据时 INT\_OUT 变回高电平。第二种方式，主机发送完命令后，向模块发送从机读地址，如果没有 ACK，继续重复发送。如果模块在没有执行完成命令前，主机发送从机地址时，模块不会 ACK。

#### 3.2 协议描述

数据通信以一帧为单位进行，格式如下：

表 6 数据通信帧结构

帧长 FrameLen	包号/命令类型 SEQNR/CmdType	命令/状态 CMD/Status	数据长度 Length	数据 DATA	校验和 BCC	帧结束符 ETX
1byte	1byte	1byte	1byte	N bytes	1byte	1byte

数据帧中各字段说明如表 7 所示：

表 7 数据帧各字段说明

字段	长度	说明	补充
FrameLen	1	数据帧的长度，包括其本身，包括帧结束符 ETX	
SEQNR/ CmdType	1	Bit7-4: 该数据帧包序号，从 0 到 15 循环。可以用来作为通信间的错误检查，从机（模块）接收到主机发来的信息，在应答信息中发出一个同样的 SEQ 信息，主机可以通过此信息检查是否发生的“包丢失”的错误。第一个包的 SEQ 可为任意值。 Bit3-0: 命令类型。 0x00: 未定义 0x01: 设备控制类命令，如读产品序号，控制蜂鸣器等 0x02: ISO14443-A 命令 其它值保留。注意：从机返回相同的 CmdType	该字段主机发送和接收的应该相同
Cmd /Status	1	主机——从机：命令 Command 从机——主机：状态 Status	
Length	1	该帧所带数据信息长度 若模块返回状态不为 0 (OK)，则 Length=0。	
DATA	Length	数据信息，长度等于 Length	
BCC	1	校验和。从 FrameLen 开始到数据 (DATA) 的最后一字节异或取反。	
ETX	1	ETX=0x03，是一个帧的结束标志	

数据帧接收规则：

- 无论何时，若接收方在 20ms 内没有接收到一个字节，表示下一接收的字节为一帧数据的开始，即帧长数据。即每发送一条命令，字节间的最大间隔不能超过 20ms。
- 一帧的结束一定是 ETX，但接收到 0x03 则不一定是帧结束。
- 帧长必须不小于 6 字节，最大不能超过 31 字节，且帧长必须等于数据长度加 6；
- BCC 计算必须正确。
- 如果采用模拟 I<sup>2</sup>C，如果不判断 SCL 是否冲突，建议向模块发送命令时，每发送一个字节延时 20us 以上再发送下一字节。

- 如果向模块发送命令没有应答或者出错，应该重复发送。

主机发送数据必须符合以上规则，否则从机不会执行任何命令，也不会有任何错误响应。同样主机接收从机的数据也必须符合以上规则，如果不符合，主机必须丢弃这帧数据。

## 4. TX522D 的 C51 函数

TX522D 提供了 C51 函数库，使用 C51 单片机的客户可直接调用这些函数库中的函数，其它系列的微控制器可参考 C51 函数库来进行移植，移植过程只要进行少量的修改。

### 4.1 函数列表

表 8 TX522D 提供的串行通讯库函数列表

命令类型 CmdType	命令	函数名	输入参数(发送)	输出参数 (接收)	功能描述
设备控制类 0x01	'A'	Get_Info	--	*Info	读取模块信息
	'B'	Config	--	--	复位且配置模块
	'C'	Close	--	--	关闭模块
	'D'	ConfigAnt	AntMode	--	配置天线驱动模式
	'E'	Load_Key	KeyAB,Sector,*Key	--	改变存贮在模块内密钥区中的密钥
	'I'	Clr_Control_Bit	--	--	将控制位置为低电平
	'J'	Set_Control_Bit	--	--	将控制位置为高电平
	'K'	Buzzer	Freguence,Opentm, Closetm,Repcnt	--	输出驱动蜂鸣器信号，能控制动作时间、间隙时间和重复次数
TYPE A 0x02	'A'	Request	ReqCode	*TagType	请求卡，检查在有效范围内是否有卡
	'B'	Casc_Anticoll	Select_Code ,Bcnt	*SNR	可实现三层防碰撞协议
	'C'	Casc_Select	Select_Code, *SNR	*Sak	可实现三层选择
	'D'	Halt	--	--	将卡置于挂起模式
	'E'	Auth2	KeyAB,*SNR, Key_Sector,Block	--	使用模块内部密钥区中的密码对指定的卡的扇区 Sector 进行验证
	'F'	Auth_Key	KeyAB,*SNR,*Key,Block	--	直接密码验证
	'G'	Read	Block	*Data	从卡中指定块中读出一个 16 字节的块
	'H'	Write	Block,*Data	--	向卡中指定块写入一 16 字节的数据块
	'I'	Ul_Write	Block,*Data	--	向 UltraLight 卡中相应块写入 4 字节数据
	'J'	Value	ValueMode,Block,*Value, Trans_Block	--	包含加、减、恢复函数，并可进行不同块之间的自动传送
	'L'	Reset	Msec	--	关闭天线输出数 ms，使卡复位
	'M'	Get_CardSnr	ReqCode	*TagType, *Sak,*SnrLen, *Snr	激活卡片并获取卡号
	'N'	Auto_Detect	ADMode,TxMode, ReqCode,AuthMode, KEYAB,*Key, uchar Block		配置 TX522D 进入自动寻卡模式
	'P'	Write_Auth	KeyAB, Key_Sector , Block, idata *Data	--	带验证的写操作
	'Q'	Read_Auth	KeyAB, Key_Sector , Block	*Data	带验证的读操作
	'R'	Value_Auth	KeyAB,Key_Sector, ValueMode,Block, *Value, Trans_Block	--	带验证的值块操作
	'S'	WriteValue_Auth	KeyAB,Key_Sector, Block, *Value	--	带验证的写值块操作
'T'	ReadValue_Auth	KeyAB,Key_Sector, Block,	*Value	带验证的读值块操作	

表 8 所有函数的返回值都是该函数执行后的状态结果，具体返回值请查看各函数说明。如果各函数有返回数据，则都是以指针的形式返回。

### 4.2 函数返回状态值列表

表 9 TX522 库函数的用到的状态值列表

名称	状态值（16 进制）	描述
OK, COMM_OK	0	函数调用成功
NO_TAG_ERR	1	在有效区域内没有卡
CRC_ERR	2	从卡中接收到了错误的 CRC 校验和
EMPTY	3	值溢出
AUTH_ERR	4	不能验证
PARITY_ERR	5	从卡中接收到了错误的校验位
CODE_ERR	6	通信错误
SENDER_ERR	8	在防冲突时读到了错误的串行码
KEY_ERR	9	证实密码错
NOT_AUTH_ERR	0x0A	卡没有验证
BIT_COUNT_ERR	0x0B	从卡中接收到了错误数量的位
BYTE_COUNT_ERR	0x0C	从卡中接收了错误数量的字节
TRANS_ERR	0x0E	调用 Transfer 函数出错
WRITE_ERR	0x0F	调用 Write 函数出错
INCR_ERR	0x10	调用 Increment 函数出错
DECR_ERR	0x11	调用 Decrment 函数出错
READ_ERR	0x12	调用 Read 函数出错
COLL_ERR	0x18	冲突错
ACCESS_TIMEOUT	0x1B	访问超时
QUIT	0x1E	上一次发送命令时被打断
CHK_WR_OK	0	Check Write 正确
CHK_WR_FAILED	1	Check Write 出错
CHK_WR_COMP_ERR	2	Check Write:写出错（比较出错）
COMM_ERR	0xFF	串行通信错误
MI_WRONG_VALUE	0x7B	值块格式错误

## 5. 函数描述

下面是 C51 函数声明，包含在头文件“TX\_D.h”中，写应用程序时，将其包含在应用函数中即可。对于 C51 函数的调用，用户可不必关心数据块格式，只要理解函数的功能，输入、输出参数即可。当用户自己编写函数（包括非 C51 下的函数）时，就需了解数据块格式，必须按照数据块的格式来编写函数。

对于只需要读取卡片某个固定块数据的应用，用户只要了解和使用 TX\_Load\_Key、TX\_Auto\_Detect 和 TX\_Read\_ADDT 即可。

对应大多数需要改写卡片数据的应用，用户一般只需要了解和使用 TX\_Config、TX\_Close、TX\_Load\_Key、TX\_Get\_CardSnr、TX\_Write\_Auth、TX\_UI\_Write、TX\_Read\_Auth、TX\_WriteValue\_Auth、TX\_Value\_Auth、TX\_ReadValue\_Auth、TX\_Halt、TX\_Reset 即可。

### 5.1 设备控制类命令 (CmdType = 0x01)

#### 5.1.1 获取信息—Get Info (Cmd = 'A')

函数原型: uchar TX\_Get\_Info(uchar idata \*Info);

输入参数: 无

输出参数: 模块的信息\*Info, Info 为保存信息空间的首地址。Info[0]~Info[4]为模块类型标识, 依次为 '5', '2', '2', 'D', 0, Info[5]~Info[8]为模块的唯一序列号, Info[9]为固件版本号, 高四位为版本号的整数, 取值从 1 到 15, 低四位为版本号的小数, 取值从 0 到 9。

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR。

功能描述: 获取 TX522D 模块的信息。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x06

CTYPE: 1

COMMAND: 0x41

LENGTH: 0

DATA[...]: 无

例如: 数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x06	0x01	0x41	0x00	none	0xB9	0x03

TX522→主机 (响应模式):

FrameLen: 0x0F

CTYPE: 1

STATUS: OK, QUIT, COMM\_ERR 中的某一个

LENGTH: 10

DATA[0]: 产品类型标识 0, '5'

DATA[1]: 产品类型标识 1, '2'

DATA[2]: 产品类型标识 2, '2'

DATA[3]: 产品类型标识 3, 'D'

DATA[4]: 产品类型标识 4, 0  
 DATA[5]: 读卡芯片序列号 0  
 DATA[6]: 读卡芯片序列号 1  
 DATA[7]: 读卡芯片序列号 2  
 DATA[8]: 读卡芯片序列号 3  
 DATA[9]: 产品软件版本号

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x0F	0x01	0x00	0x0a	9 字节产品型号及版本	0xXX	0x03

### 5.1.2 配置—Config (Cmd = 'B')

函数原型: uchar TX\_Config(void);

输入参数: 无

输出参数: 无

函数返回: TX522D 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR。

功能描述: 对模块进行初始化, 初始化成功后, 模块上的指示灯将点亮, 此时天线发射载波信号, 任何进入天线感应区的卡可得电进入 IDLE 状态, 可使用任一函数对卡进行操作。此时读卡芯片完全被激活, 所消耗的电流最大。

**建议用户每次读卡操作前都调用该函数。**

注意: 对模块的配置需要大约 2.5ms 的时间, 因此调用该函数后需要等待 2.5ms 以上才能进行其他操作。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 6  
 CTYPE: 1  
 COMMAND: 0x42  
 LENGTH: 0  
 DATA[...]: 无

例如: 数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x06	0x01	0x42	0x00	none	0xBA	0x03

TX522→主机 (响应模式):

FrameLen: 6  
 CTYPE: 1  
 STATUS: OK, QUIT, COMM\_ERR 中的某一个  
 LENGTH: 0  
 DATA[...]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x01	0x00	0x00	none	0xF8	0x03

### 5.1.3 关闭—Close (Cmd = 'C')

函数原型: `uchar TX_Close(void);`

输入参数: 无

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: `OK`, `COMM_ERR`。

功能描述: 此函数将关闭 TX522D 模块, 指示灯熄灭, 天线不发送载波信号, 模块消耗的电流最小, 在此状态在, 模块不能使用。若要重新使用模块, 需要调用 `TX_Config()` 函数对 TX522D 重新进行配置。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 6  
CTYPE: 1  
COMMAND: 0x43  
LENGTH: 0  
DATA[···]: 无

例如: 数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x06	0x01	0x43	0x00	none	0xBB	0x03

TX522→主机 (响应模式):

FrameLen: 6  
CTYPE: 1  
STATUS: `OK`, `COMM_ERR` 中的某一个  
LENGTH: 0  
DATA[···]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x01	0x00	0x00	none	0xF8	0x03

### 5.1.4 配置—ConfigAnt

函数原型: `uchar TX_ConfigAnt(TXMode);`

输入参数: `TxMode`: 天线驱动模式, 0x00= TX1 和 TX2 关闭; 0x01=TX1 驱动; 0x02 = TX2 驱动; 0x03 = TX1 和 TX2 同时驱动

输出参数: 无

函数返回: TX522B 执行命令后的状态, 可能的状态值如下: `OK`, `QUIT`, `COMM_ERR`。

功能描述: 对模块天线驱动模式进行配置。模块默认为 TX1 和 TX2 同时驱动。只要要进行双天线操作时, 才会使用到该函数。分体式模块不支持该函数。

数据块格式描述:

主机→TX522 命令模式):

FrameLen: 6  
CTYPE: 1

COMMAND: 0x44  
LENGTH: 1  
DATA[0]: TxMode

例如：配置为 TX1 驱动的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x06	0x01	0x44	0x01	0x01	0xBD	0x03

TX522→主机（响应模式）：

FrameLen: 6  
CTYPE: 1  
STATUS: OK, COMM\_ERR 中的某一个  
LENGTH: 0  
DATA[···]: 无

例如：数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x01	0x00	0x00	none	0xF8	0x03

### 5.1.5 装载密钥—Load\_Key (Cmd = 'E')

函数原型：uchar TX\_Load\_Key(uchar KeyAB,uchar Key\_Sector,uchar idata \*Key)

输入参数：KeyAB：密钥类型（1 字节）。可取值为 KeyAB=0x00（KEYA）—密钥 A，或 KeyAB=0x04（KEYB）—密钥 B。

Key\_Sector：模块内的密钥区号（1 字节）：取值范围 0~15

\*Key：需要装载到模块内密钥区的密钥（6 字节）

输出参数：无

函数返回：执行命令后的状态，可能的状态值如下：OK, QUIT, AUTH\_ERR, COMM\_ERR。

功能描述：此函数的作用是将指定的密码（\*Key）装载到模块内指定的密钥区（Key\_Sector），并非改变 Mifare1 卡内扇区的密码。本函数只对模块进行操作，模块与卡之间没有数据传输。

装置的密钥掉电不丢失，因此只用装载一次就可以。考虑到系统安全性，装载密钥过程可单独进行，用户程序中可不出现该命令。

模块内有 16 个密码区（区号 0——15），称它为密钥区号 Key\_Sector。每个区分密钥 A（0x60）和密钥 B（0x61）两个，总共 32 个密码。装载成功后，可用该密钥对 Mifare1 卡进行验证。

在 M1 卡中也有 16 个存储区，称它为扇区号 Sector。若要改变 Mifare1 卡内的密钥，可在用原密码验证通过后，直接用写块数据 TX\_Write()函数，将密码块改写。Mifare 卡出厂后的初始密钥为 6 个 FFH，A 和 B 密钥都一样。

数据块格式描述：

主机→TX522 命令模式：

FrameLen: 0x0E



CTYPE: 1  
 COMMAND: 0x45  
 LENGTH: 8  
 DATA[0]: KeyAB  
 DATA[1]: Key\_Sector  
 DATA[2]: Key[0]  
 ...  
 DATA[7]: Key[5]

例如：往密钥 0 区装载密钥 A: 0xff 0xff 0xff 0xff 0xff 0xff 的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x0E	0x01	0x45	0x08	0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff	0xBD	0x03

TX522→主机（响应模式）：

FrameLen: 6  
 CTYPE: 1  
 STATUS: OK, QUIT, AUTH\_ERR, COMM\_ERR 中的某一个  
 LENGTH: 0  
 DATA[0]: 无

例如：数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x01	0x00	0x00	none	0xF8	0x03

### 5.1.6 清除控制位—Clr\_Control\_Bit (Cmd = 'I')

函数原型：TX\_Clr\_Control\_Bit(void);

输入参数：无

输出参数：无

函数返回：执行命令后状态，可能状态值如下：OK, QUIT, COMM\_ERR 中的某一个。

功能描述：此函数设置 TX522D 模块的控制位 Ctrl (J2\_1 脚) 为低电平。

数据块格式描述：

主机→TX522 命令模式：

FrameLen: 6  
 CTYPE: 1  
 COMMAND: 0x49  
 LENGTH: 0  
 DATA[0]: 无

TX522→主机（响应模式）：

FrameLen: 6  
 CTYPE: 1  
 STATUS: OK, QUIT, COMM\_ERR 中的某一个  
 LENGTH: 0  
 DATA[0]: 无

### 5.1.7 置位控制位—Set\_Control\_Bit (Cmd = 'J')

函数原型: uchar TX\_Set\_Control\_Bit(void);

输入参数: 无

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR 中的一个。

功能描述: 此函数设置 TX522D 模块的控制位 Ctrl (J2\_1 脚) 为高电平。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 6  
CTYPE: 1  
COMMAND: 0x4A  
LENGTH: 0  
DATA[0]: 无

TX522→主机 (响应模式):

FrameLen: 6  
CTYPE: 1  
STATUS: OK, QUIT, COMM\_ERR 中的一个  
LENGTH: 0  
DATA[0]: 无

### 5.1.8 输出蜂鸣器信号—Buzzer (Cmd = 'K')

函数原型: TX\_Buzzer(uchar Frquence, uchar Opentm, uchar Closetm, uchar Repcnt);

输入参数: Frquence: 输出方波频率, 取值 (0~255), 对应频率 (0.73~4K), 0 为直流, 198 对应 2K。

Opentm: 方波输出持续时间, 取值 (0~255), 10ms 的分辨率

Closetm: 间隙时间, 取值 (0~255), 10ms 的分辨率

Repcnt: 重复次数

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, COMM\_ERR 中的一个。

功能描述: 此函数输出一方波用于驱动交流蜂鸣器或低电平驱动直流蜂鸣器, 驱动的频率、持续时间、间隙时间和重复次数可设定。

向模块发送蜂鸣器控制命令后, 在命令设定的时间内模块不会再接收其它命令。因此必须等蜂鸣器鸣叫执行完成后, 才能向模块发送其它命令。例如 Opentm = 50, Closetm = 20, Repcnt = 2, 那蜂鸣器鸣叫的时间为  $50*2*10ms + 20*(2-1)*10ms = 1.2s$ 。

下面是频率对照表。

表 10 设定值与输出频率对照表

Frquence	输出频率 (KHz)	Frquence	输出频率 (KHz)
0	输出低电平 (直流)	140	1.333
1	0.735	160	1.515

20	0.787	180	1.74
40	0.847	198	2.00
60	0.913	200	2.04
80	0.990	220	2.50
100	1.081	240	3.17
120	1.198	255	4.00

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x0A  
CTYPE: 1  
COMMAND: 0x4B  
LENGTH: 4  
DATA[0]: Frequence  
DATA[1]: Opentm  
DATA[2]: Closetm  
DATA[3]: Repcnt

TX522→主机 (响应模式):

FrameLen: 6  
CTYPE: 1  
STATUS: OK, COMM\_ERR 中的某一个  
LENGTH: 0  
DATA[0]: 无

## 5.2 ISO14443A 类命令 (CmdType = 0x02)

### 5.2.1 激活卡片并获取卡号—Get\_CardSnr (Cmd = 'M')

函数原型: `uchar TX_Get_CardSnr(uchar ReqCode, uchar idata *TagType, uchar idata *Sak, uchar idata *SnrLen, uchar idata *Snr);`

输入参数: ReqCode: 请求模式

ReqCode=0x26 (IDLE), 请求天线范围内 IDLE 状态的卡 (HALT 状态的除外)

ReqCode=0x52 (ALL), 请求天线范围内的所有卡。

输出参数:

(1) \*TagType: 请求应答: 2 个字节的卡片类型, 其意义见后。

(2) \*Sak: 最后一级选择应答的应答, 其意义见后。

(3) \*SnrLen: 返回卡片序列号的长度。

(4) \*Snr: 返回卡片的序列号。

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR 中的一个。

功能描述: 每张卡片必须先激活才能进行后面的验证和读写操作。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x07  
 CTYPE: 2  
 COMMAND: 0x4d  
 LENGTH: 1  
 DATA[0]: ReqCode

例如：以 IDLE 方式激活卡的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x07	0x02	0x4d	0x01	0x26	0x90	0x03

TX522→主机（响应模式）：

FrameLen: 0x0e  
 CTYPE: 2  
 STATUS: OK, QUIT, COMM\_ERR 中的某一个  
 LENGTH: 4 字节+序列号的长度, Mifare1 S50、S70、Light 卡: 8 字节,  
 Mifare0 UltraLight 和 Mifare3 Desfire 卡: 11 字节  
 DATA[0..1]: \*TagType: 请求应答, 2 个字节的卡片类型  
 DATA[2]: \*Sak: 最后一级选择应答的应答  
 DATA[3]: \*SnrLen: 返回卡片序列号的长度  
 DATA[4..4+ SnrLen]: \*Snr: 返回卡片的序列号

例如：一张序列号为 0x007e0a42 的 Mifare1 S50 卡返回的数据

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x0e	0x02	0x00	0x08	0x04 0x00 0x08 0x04 0x42 0x0A 0x7E 0x00	0xXX	0x03

例如：一张序列号为 0x000000007e0a42 的 Mifare UltraLight 卡返回的数据

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x11	0x02	0x00	0x0b	0x44 0x00 0x00 0x07 0x42 0x0A 0x7E 0x00 0x00 0x00 0x00	0xXX	0x03

表 11 \*TagType 的含义

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
保留								序列号大小 00:4bytes 01:7bytes 10:10bytes	保留	任何位为 1 时, 则为比特帧防碰撞方式					

表 12 常用卡的\*TagType 值

卡类型	*TagType	卡类型	*TagType
Mifare1 S50	0x0004	SHC1101	0x0004
Mifare1 S70	0x0002	SHC1102	0x3300
Mifare Light	0x0010	11RF32	0x0004
Mifare UltraLight	0x0044		

表 13 常用卡的\*Sak 的值

卡类型	*Sak	卡类型	*Sak
Mifare1 S50	0x08	Mifare0 UltraLight	0x04
Mifare1 S70	0x18	SHC1101	0x22
Mifare1 Light	0x01	11RF32	0x08

### 5.2.2 证实 2—Auth2 (Cmd = 'E')

函数原型: uchar TX\_Auth2(uchar KeyAB, uchar idata \*SNR, uchar Key\_Sector, uchar Block);

输入参数: KeyAB: 密钥类型 (1 字节)。可取值为 KeyAB=0x60 (KEYA), 利用密钥 A 进行验证, 或 KeyAB=0x61 (KEYB), 利用密钥 B 进行验证。

\*SNR: 前一次防碰撞返回的卡的序号, 或已知的卡的序列号。无符号 4 字节, 低字节放在低地址处。

Key\_Sector: 模块内的密钥区号 (1 字节): 取值范围 0~15。

Block: 所要验证的卡块号 (1 字节): S50: 0~63; S70: 0~255。

Block=当前扇区\*4+当前扇区的块号

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, NO\_TAG\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR。

功能描述: 使用模块内部密钥区中 Key\_Sector 中的密码对指定的卡的块号 Block 进行验证, 若卡 Block 的密码与存储在模块内 Key\_Sector 中的密码相同, 则验证成功, 返回 OK。

该函数也依赖于 TX\_Load\_Key 函数曾经成功执行过, 因为模块内部密码区(Key\_Sector) 中的密码要由 TX\_Load\_Key 函数事先装载。该函数适用于对于所有卡来说密码相同的应用, 密钥的装载可以在一个安全的场合一次性装入。

卡片扇区中块的内容必须经过验证后才能进行读写操作或者值块操作。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x0d  
 CTYPE: 2  
 COMMAND: 0x45  
 LENGTH: 0x07  
 DATA[0]: KeyAB  
 DATA[1]: 卡的序列号第 1 字节 (最低字节)  
 DATA[2]: 卡的序列号第 2 字节  
 DATA[3]: 卡的序列号第 3 字节  
 DATA[4]: 卡的序列号第 4 字节  
 DATA[5]: Key\_Sector  
 DATA[6]: Block

例如: 用密钥 0 区的密钥 A 证实卡的扇区 0 的第 2 块数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x0d	0x02	0x45	0x07	0x60 0x42 0x0A 0x7E 0x00 0x00 0x02	0xxx	0x03

TX522→主机 (响应模式):

FrameLen: 0x06  
 CTYPE: 2  
 STATUS: OK, QUIT, NO\_TAG\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个  
 LENGTH: 0

DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

### 5.2.3 直接密码证实—Auth\_Key (Cmd = 'F')

函数原型: `uchar TX_Auth_Key (uchar KeyAB, uchar idata *SNR, uchar idata *Key, uchar Block);`

输入参数: **KeyAB**: 密钥类型 (1 字节)。可取值为 `KeyAB=0x60 (KEYA)`, 利用密钥 A 进行验证, 或 `KeyAB=0x61 (KEYB)`, 利用密钥 B 进行验证。

**\*SNR**: 前一次防碰撞返回的卡的序号, 或已知的卡的序列号。无符号 4 字节, 低字节放在低地址处。

**\*Key**: 用于证实的密码首地址, 应在外部定义一个共 6 个字节的数组用于存放密码。

**Block**: 所要验证的卡块号 (1 字节): `S50: 0~63; S70: 0~255。`

**Block=当前扇区\*4+当前扇区的块号**

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: `OK, QUIT, NO_TAG_ERR, AUTH_ERR, PARITY_ERR, BIT_COUNT_ERR, COMM_ERR。`

功能描述: `TX_Auth2` 函数都必须依赖于 `TX_Load_Key()` 函数的曾经的成功执行, 在进行证实之前, 一定要确认正确的密钥已经存于模块的密钥区内。而 `TX_Auth_Key` 函数则将密钥存放于 `Key` 指针所指向的 6 个字节存储区内, `TX_Auth_Key` 函数执行时不对密钥区进行操作, 因此也称为直接密码证实。若卡中的密钥与所传输的密码相匹配。则证实成功, 函数将返回 `OK`。

直接密码证实一般用于对每一张卡来说密钥都不同的应用, 如在使用安全模块 (PSAM 卡) 的消费应用中, 消费机首先将卡的序列号读出, 然后与 PSAM 卡中消费主密钥一起生成导出密钥, 然后直接用导出密钥与卡的一个应用扇区相互证实。

数据块格式描述:

主机→TX522 命令模式:

```

FrameLen:    6
CTYPE:      2
COMMAND:    0x46
LENGTH:     0x0C
DATA[0]:    KeyAB
DATA[1]:    卡的序列号第 1 字节 (最低字节)
DATA[2]:    卡的序列号第 2 字节
DATA[3]:    卡的序列号第 3 字节
DATA[4]:    卡的序列号第 4 字节
DATA[5]:    Key[0]
...
DATA[10]:   Key[5]
DATA[11]:   Block
    
```

例如: 用密码 A `0xff 0xff 0xff 0xff 0xff 0xff` 证实扇区 0 的第 2 块数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x12	0x02	0x46	0x0c	0x60 0x42 0x0A 0x7E 0x00 0xff 0xff 0xff 0xff 0xff 0x02	0xXX	0x03

TX522→主机（响应模式）：

FrameLen: 0x06

CTYPE: 2

STATUS: OK, QUIT, NO\_TAG\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

#### 5.2.4 读—Read (Cmd = 'G')

函数原型: uchar TX\_Read(uchar Block,uchar idata \*Data)

输入参数: Block: 卡块号 (1 字节): S50: 0~63; S70: 0~255

Block=当前扇区\*4+当前扇区的块号

输出参数: \*Data: Data 为读回 16 字节数据的首地址。

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个。

功能描述: 在验证成功后, 使用该函数读 Mifare 卡中相应块的数据。Mifare 卡中一个块的数据是 16 字节, 因此读写一次均是 16 个字节。

所读块号必须与之前所验证的块号在同一个扇区内, mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。密码数据不能被读取。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x07

CTYPE: 2

COMMAND: 0x47

LENGTH: 1

DATA[0]: Block

例如: 读取块 2 的数据的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x07	0x02	0x47	0x01	0x02	0xbe	0x03

TX522→主机（响应模式）：

FrameLen: 0x16

CTYPE: 2

STATUS: OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR,

PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个

LENGTH: 0x10  
DATA[0]: 所访问块的第一个字节  
:  
DATA[15]: 所访问块的最后一个字节  
例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x16	0x02	0x00	0x10	16 字节数据	0xXX	0x03

### 5.2.5 写—Write (Cmd = 'H')

函数原型: uchar TX\_Write(uchar Block,uchar idata \*Data)

输入参数: Block: 卡块号 (1 字节): S50: 1~63; S70: 1~255

Block=当前扇区\*4+当前扇区的块号

\*Data: 16 字节数据指针, Data 为写入的 16 字节数据的首地址。

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个。

功能描述: 对卡内某一块进行验证成功后, 即可对同一扇区的各个块进行写操作 (只要访问条件允许), 其中包括位于扇区尾的密码块, 这是更改密码的唯一方法。

Mifare 卡中一个块的数据是 16 字节, 因此读写一次均是 16 个字节。所读块号必须与之前所验证的块号在同一个扇区内, mifare1 卡从块号 0 开始按顺序每 4 个块 1 个扇区。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x17  
CTYPE: 2  
COMMAND: 0x48  
LENGTH: 0x11  
DATA[0]: Block  
DATA[1]: 所要写的第一个字节  
:  
DATA[16]: 所要写的最后一个字节  
例如: 往块 2 写入数据的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x17	0x02	0x48	0x11	0x02 16 字节数据	0xXX	0x03

TX522→主机 (响应模式):

FrameLen: 0x06  
CTYPE: 2  
STATUS: OK, QUIT, NO\_TAG\_ERR, NOT\_AUTH\_ERR, WRITE\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个



LENGTH: 0  
DATA[0]: 无  
例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

补充说明:

注意: 在将数据写到卡片上的某一扇区时, 一定要小心。因为有些 block 中存储了密码数据以及存储允许使能数据。特别是每一个扇区的 Block3 中存放了该扇区的存取条件, 包含有 KEYA, KEYB 及该扇区的控制字。Mifare 1 卡片出厂时的 Block3 有缺省值, 为: “a0a1a2a3a4a5ff078069b0b1b2b3b4b5”, 共 16 个 Bytes。

涉及 Mifare 1 卡片的存储结构等信息, 请参考 Mifare 1 卡片数据手册。

程序员在使用 Mifare 1 卡片做应用时, 一定要清清楚楚记住每一个扇区的 Block3 的数据, 这样也就记住了扇区的密码和存取控制字。否则, 扇区的存储空间将不执行 Read/Write 等操作而失效。

任何人试图用任何方式来读写不知密码的卡片或某一扇区都是徒劳无益的。

卡片应放在安全的地方, 即不要放在离模块天线较近的地方。因为当模块对其它卡片执行某些指令时, 有可能无意间对这一卡片进行了读/写等操作, 从而操作卡片的失效。

### 5.2.6 暂停—Halt (Cmd = 'D')

函数原型: uchar TX\_Halt(void)

输入参数: 无

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR。

功能描述: 将天线区所选择卡置为挂起状态。如果要进行重新激活, 则应用 ALL 模式调用激活命令。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x06

CTYPE: 2

COMMAND: 0x44

LENGTH: 0

DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x06	0x02	0x44	0x00	none	0xbf	0x03

TX522→主机 (响应模式):

FrameLen: 6

CTYPE: 2

STATUS: OK, QUIT, COMM\_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

### 5.2.7 带内部自动传送的值操作—Value (Cmd = 'J')

函数原型：uchar TX\_Value(uchar ValueMode, uchar Block, uchar idata \*Value, uchar Trans\_Block);

输入参数：ValueMode: 0xC0—减； 0xC1—加； 0xC2—恢复

Block: 卡内块地址，对该块进行值操作，取值范围：S50: 1~63; S70: 1~255

Block=当前扇区\*4+当前扇区的块号

\*Value: 4 字节数据指针，用来存储减少值或增加值，当进行恢复操作时，该值为空值。

Value 是减少值或增加值存放的首地址，存放时，低地址存放高字节。

Trans\_Block: 传输块地址，取值范围：S50: 1~63; S70: 1~255。

输出参数：无

函数返回：执行命令后的状态，可能的状态值如下：OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个。

功能描述：此函数对卡内的某一块进行加、减或数据备份，该块必须为值块格式，并支持自动传送。该函数其实是 TX\_Increment、TX\_Decrement 或 TX\_Restore 函数与 TX\_Transfer 函数的组合。因此可以用该函数替换上述函数。若卡块号与传输块号相同，则将操作后的结果写入原来的块内；若卡块号与传输块号不相同，则将操作后的结果写入传输块内，结果传输块内的数据被覆盖，原块内的值不变。当模式为“恢复”时，“值”无意义。

数据块格式描述：

主机→TX522 命令模式：

FrameLen: 0x0D  
 CTYPE: 2  
 COMMAND: 0x4A  
 LENGTH: 0x07  
 DATA[0]: ValueMode  
 DATA[1]: Block  
 DATA[2]: Value(LL)  
 DATA[3]: Value(LH)  
 DATA[4]: Value(HL)  
 DATA[5]: Value(HH)  
 DATA[6]: Trans\_Block

例如：将块 1 的值减 3，然后传送到块 2 的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x0d	0x02	0x4a	0x07	0xc0 0x01 0x03 0x00 0x00 0x00 0x02	0xXX	0x03

TX522→主机（响应模式）：

FrameLen: 6

CTYPE: 2  
STATUS: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个

LENGTH: 0  
DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

### 5.2.8 复位—Reset (Cmd = 'L')

函数原型: uchar TX\_Reset(uchar Msec);

输入参数: Msec: 取值 0~255, 模块上射频电路关闭时间 (以 ms 毫秒为单位), Msec=0 时, 一直关闭。

输出参数: 无

函数返回: 执行命令后的状态, 可能的状态值如下: OK, QUIT, COMM\_ERR 中的某一个。

功能描述: 该函数使模块上的射频电路关闭, 关闭的时间由参数 Msec 指定, 若 Msec=0, 射频电路将一直处于关闭状态, 一直到下一个 TX\_Request 命令到来。关闭射频电路能使天线内的所有卡复位。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x07

CTYPE: 2

COMMAND: 0x4C

LENGTH: 1

DATA[0]: Msec

例如: 将天线信号关闭 1ms 的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x07	0x02	0x4C	0x01	0x01	0xb6	0x03

TX522→主机 (响应模式):

FrameLen: 0x06

CTYPE: 2

STATUS: OK, QUIT, COMM\_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

### 5.2.9 自动寻卡—Auto\_Detect (Cmd = 'N')

函数原型：uchar TX\_Auto\_Detect(uchar ADMMode, uchar TxMode, uchar ReqCode,  
uchar AuthMode, uchar KEYAB,  
uchar idata \*Key, uchar Block)

输入参数：

(1) ADMMode--自动检测模式 (1 字节)

B7	B6	B5	B4	B3	B2	B1	B0
RFU 00000				1: 最后执行 Halt() 命令; 0: 无动作。	数据输出后 1: 继续检测; 0: 不继续检测。	RFU	RFU

(2) TxMode--天线驱动方式 (1 字节) 暂时保留不使用。

(3) ReqCode--请求代码 (1 字节):  
0x26——IDLE  
0x52——ALL

(4) AuthMode--验证命令 (1 字节):  
'E' (0x45)——用 E2 密码验证  
'F' (0x46)——用直接密码验证  
0——不验证

(5) KEYAB--密钥 AB (1 字节):  
0x60——密钥 A  
0x61——密钥 B

(6) Key--密钥:  
若验证命令='E', 则为密钥区号 (1 字节)  
若验证命令='F', 则为密钥 (6 字节)

(7) Block--卡块号 (1 字节):  
S50: 0~63  
S70: 0~255

**Block=当前扇区\*4+当前扇区的块号**

输出参数：无。

返回值：执行命令后的状态，可能的状态值如下：COMM\_ERR、COMM\_OK 等。

功能描述：用于配置 TX522D 进入自动寻卡模式。配置好后模块不断去读取卡片，检测是否有卡片进入天线区域。如果有卡片进入，则自动读取卡号，并根据配置验证卡片读取数据，并产生中断。在主机读取数据后，根据配置继续进行自动寻卡，或者退出自动寻卡模式。请注意下面几点：

1. 自动寻卡命令会保存在内部 E2ROM，掉电保存。下一次上电或者复位后继续有效。
2. 寻到卡后 INT\_OUT 变由高电平变为低电平，以通知外部控制器读取数据。数据读走或者 450ms 后 INT\_OUT 变回高电平。
3. 寻到卡后，外部控制器必须在 450ms 内将数据读走，否则数据会丢失或覆盖。
4. 寻到卡后，外部控制器直接向模块读取数据，类似于读取 E2PROM 的方法。

自动寻卡模式的退出方式：

1. 配置自动寻卡时就配置为寻到卡后取消自动寻卡；
2. 发送一个数据长度小于 2（帧长小于 8）的命令，建议使用 GetInfo 命令，因为该命令不会影响当前卡片的状态。

建议采用发送上面的第 2 种方法来结束自动寻卡。因为用户可以先读取自动寻卡数据，确定正确读取到所需要的数据，然后再发送长度小于 8 的命令来结束寻卡。

取消自动寻卡后，一定要延时 5ms 才能再次向模块发送命令，否则模块不会响应。

自动寻卡期间，模块将对数据长度大于 2（帧长大于 8）的命令不响应。

数据块格式描述:

1. 主机→TX522 (命令模式):

FrameLen: 0xxx

CTYPE: 2

COMMAND: 0x4e

LENGTH: 若验证命令='E', 则为 7  
若验证命令='F', 则为 12  
若验证命令=NULL, 则为 4

DATA:

验证命令	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]~[10]	D[11]
NULL					-	-	-	-	-
'E'	ADMode	TxMode	ReqCode	AuthMode	KEYAB	密钥扇区号	块号	-	-
'F'					KEYAB	密钥			块号

例如: 当检测到有卡时产生中断, 以 IDLE 方式激活卡, 用直接密码验证密钥 A (密码为 0xff 0xff 0xff 0xff 0xff 0xff), 读出第 1 块数据, 数据输出后停止自动寻卡。则数据帧如下:

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x12	0x02	0x4e	0x0C	0x00 0x00 0x26 0x46 0x60 0xff 0xff 0xff 0xff 0xff 0xff 0x01	0xXX	0x03

2. TX522→主机 (响应模式):

FrameLen: 0x06

CTYPE: 2

STATUS: OK 等

LENGTH: 0

DATA[0]: 无

例如, 数据帧:

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

当在自动寻卡模块下, 刷卡后, INT\_OUT 变由高电平变为低电平, 以通知外部控制器读取数据。外部控制器类似于读取 E2PROM 的方法直接向模块读取数据。数据读走或者 450ms 后 INT\_OUT 变回高电平。读取到的数据格式如下:

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
帧长	0x02			卡号信息+数据块信息	0xXX	0x03

FRAMELEN: 第一个字节为帧长, 用于判断需要后续接收的数据长度。帧长

CTYPE: 2

STATUS: OK—成功

LENGTH: 数据长度, 根据是否读取扇区内容数据长度不同

DATA: 数据, 数据定义如下表

验证命令	D[0]	D[1]~[2]	D[3]	D[4]	D[5]~[8]	D[9]~[24]
NULL	TxDrv	ATQ	SAK	序列号	序列号	序列号（根据长度而定）
'E'、'F'	(天线驱动)	(请求应答)	(选择应答)	长度	序列号	扇区数据

输出参数：  
 TxDrv – 天线驱动方式（1 字节）；  
 ATQ – 请求应答（2 字节）；  
 SAK – 选择应答（1 字节）；  
 SnrLen- 序列号长度(根据卡片类型不同而不同，通常为 4 或者 7)  
 PiccSnr – 卡号（4 或者 7 字节）；  
 DTBfr –数据（16 字节，如果自动寻卡命令的验证命令不为 NULL）；

例如：检测到序列号为 0x007e0a42 的 S50 卡，从机发送的数据为：

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x0f	0x02	0x00	0x09	0x03 0x04 0x00 0x08 0x04 0x42 0x0a 0x7e 0x00	0xXX	0x03

### 5.2.10 带验证的写—Write\_Auth (Cmd = 'P')

函数原型：uchar TX\_Write\_Auth(uchar KeyAB, uchar Key\_Sector, uchar Block, uchar idata \*Data)

输入参数：

- KEYAB--密钥 AB（1 字节）：  
 0x60——密钥 A  
 0x61——密钥 B
- Key\_Sector: 模块内的密钥区号（1 字节）：取值范围 0~15。
- Block--卡块号（1 字节）：  
 S50: 1~63  
 S70: 1~255

**Block=当前扇区\*4+当前扇区的块号**

- \*Data: 16 字节数据指针，Data 为写入的 16 字节数据的首地址。

输出参数： 无

函数返回：执行后可能返回：OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR, CHK\_WR\_FAILED, CHK\_WR\_COMP\_ERR 中的某一个。

功能描述：该函数在 Get\_CardSnr 后执行，先对卡内某一块进行验证，成功后对指定块进行写操作（只要访问条件允许），其中包括位于扇区尾的密码块，这是更改密码的唯一方法。该函数在写入数据后会立即进行读操作并进行数据写入正确性判断。

此命令相当于 Auth2 和 Write 的组合。

数据块格式描述：

主机→TX522 命令模式：

FrameLen: 0x19  
 CTYPE: 2  
 COMMAND: 0x50  
 LENGTH: 0x13  
 DATA[0]: KEYAB  
 DATA[1]: Key\_Sector  
 DATA[2]: Block  
 DATA[3]: 所要写的第一个字节

:

DATA[18]: 所要写的最后一个字节

例如: 使用模块内部密匙 0 区的密匙 A 进行验证, 往块 2 写入数据的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x19	0x02	0x50	0x13	0x60 0x00 0x02 16 字节数据	0xXX	0x03

TX522→主机(响应模式):

FrameLen: 0x06

CTYPE: 2

STATUS: OK, QUIT, NO\_TAG\_ERR, NOT\_AUTH\_ERR, WRITE\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

补充说明:

注意: 在将数据写到卡片上的某一扇区时, 一定要小心。因为有些 block 中存储了密码数据以及存储允许使能数据。特别是每一个扇区的 Block3 中存放了该扇区的存取条件, 包含有 KEYA, KEYB 及该扇区的控制字。Mifare 1 卡片出厂时的 Block3 有缺省值, 为: “a0a1a2a3a4a5ff078069b0b1b2b3b4b5”, 共 16 个 Bytes。

涉及 Mifare 1 卡片的存储结构等信息, 请参考 Mifare 1 卡片数据手册。

程序员在使用 Mifare 1 卡片做应用时, 一定要清清楚楚记住每一个扇区的 Block3 的数据, 这样也就记住了扇区的密码和存取控制字。否则, 扇区的存储空间将不执行 Read/Write 等操作而失效。

任何试图用任何方式来读写不知密码的卡片或某一扇区都是徒劳无益的。

卡片应放在安全的地方, 即不要放在离模块天线较近的地方。因为当模块对其它卡片执行某些指令时, 有可能无意间对这一卡片进行了读/写等操作, 从而操作卡片的失效。

### 5.2.11 带验证的读—Read\_Auth (Cmd = 'Q')

函数原型: uchar TX\_Read\_Auth(uchar KeyAB, uchar Key\_Sector, uchar Block, uchar idata \*Data)

输入参数:

- (1) KEYAB--密钥 AB (1 字节):  
0x60——密钥 A  
0x61——密钥 B
- (2) Key\_Sector: 模块内的密钥区号 (1 字节): 取值范围 0~15。
- (3) Block--卡块号 (1 字节):  
S50: 1~63  
S70: 1~255

Block=当前扇区\*4+当前扇区的块号

输出参数: \*Data: Data 为读回 16 字节数据的首地址。

函数返回: 执行后可能返回: OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个。

功能描述: 该函数在 Get\_CardSnr 后执行, 先对卡内某一块进行验证, 成功后读 Mifare 卡中相应块的数据。密码数据不能被读取。

此命令相当于 Auth2 和 Read 的组合。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x09

CTYPE: 2

COMMAND: 0x51

LENGTH: 3

DATA[0]: KEYAB

DATA[1]: Key\_Sector

DATA[2]: Block

例如: 使用模块内部密钥 0 区的密钥 A 进行验证, 读取块 2 的数据的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x09	0x02	0x51	0x03	0x60, 0x00, 0x02	0xA4	0x03

TX522→主机 (响应模式):

FrameLen: 0x16

CTYPE: 2

STATUS: OK, QUIT, NO\_TAG\_ERR, CRC\_ERR, NOT\_AUTH\_ERR, PARITY\_ERR, BIT\_COUNT\_ERR, COMM\_ERR 中的某一个

LENGTH: 0x10

DATA[0]: 所访问块的第一个字节

:

DATA[15]: 所访问块的最后一个字节

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x16	0x02	0x00	0x10	16 字节数据	0XX	0x03

### 5.2.12 带验证的值块操作—Value\_Auth (Cmd = 'R')

函数原型: uchar TX\_Value\_Auth (uchar KeyAB, uchar Key\_Sector, uchar ValueMode, uchar Block, uchar idata \*Value, uchar Trans\_Block);

输入参数:

- (1) KEYAB--密钥 AB (1 字节): 0x60——密钥 A  
0x61——密钥 B
- (2) Key\_Sector: 模块内的密钥区号 (1 字节): 取值范围 0~15。
- (3) ValueMode: 0xC0—减; 0xC1—加; 0xC2—恢复
- (4) Block--卡块号 (1 字节): S50: 1~63  
S70: 1~255

**Block=当前扇区\*4+当前扇区的块号**

- (5) \*Value: 4 字节数据指针, 用来存储减少值或增加值, 当进行恢复操作时, 该值为空值。Value 是减少值或增加值存放的首地址, 存放时, 低地址存放高字节。



(6) Trans\_Block: 传输块地址, 取值范围: S50: 1~63; S70: 1~255。

输出参数: 无

函数返回: 执行后可能返回: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个。

功能描述: 该函数在 Get\_CardSnr 后执行, 先对卡内某一块进行验证, 成功后对卡内的某一块进行加、减或数据备份, 该块必须为值块格式, 并支持自动传送。若卡块号与传输块号相同, 则将操作后的结果写入原来的块内; 若卡块号与传输块号不相同, 则将操作后的结果写入传输块内, 结果传输块内的数据被覆盖, 原块内的值不变。当模式为“恢复”时, “值”无意义。

此命令相当于 Auth2 和 Value 的组合。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x0F  
 CTYPE: 2  
 COMMAND: 0x52  
 LENGTH: 0x09  
 DATA[0]: KEYAB  
 DATA[1]: Key\_Sector  
 DATA[2]: ValueMode  
 DATA[3]: Block  
 DATA[4]: Value(LL)  
 DATA[5]: Value(LH)  
 DATA[6]: Value(HL)  
 DATA[7]: Value(HH)  
 DATA[8]: Trans\_Block

例如: 使用模块内部密钥 0 区的密钥 A 进行验证, 将块 1 的值减 3, 然后传送到块 1 的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x0F	0x02	0x52	0x09	0x60 0x00 0xc0 0x01 0x03 0x00 0x00 0x00 0x01	0Xxx	0x03

TX522→主机 (响应模式):

FrameLen: 0x06  
 CTYPE: 2  
 STATUS: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个

LENGTH: 0  
 DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0Xfb	0x03

### 5.2.13 带验证的写值块操作—WriteValue\_Auth (Cmd = 'S')

函数原型: uchar TX\_WriteValue\_Auth(uchar KeyAB, uchar Key\_Sector , uchar Block, uchar idata \*Value);

输入参数:

- (1) KEYAB--密钥 AB (1 字节):           0x60——密钥 A  
  0x61——密钥 B
- (2) Key\_Sector: 模块内的密钥区号 (1 字节): 取值范围 0~15。
- (3) Block--卡块号 (1 字节):            S50: 1~63  
  S70: 1~255

**Block=当前扇区\*4+当前扇区的块号**

- (4) \*Value: 4 字节数据指针, 用来存储减少值或增加值, 当进行恢复操作时, 该值为空值。Value 是减少值或增加值存放的首地址, 存放时, 低地址存放高字节。

输出参数: 无

函数返回: 执行后可能返回: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个。

功能描述: 该函数在 Get\_CardSnr 后执行, 先对卡内某一块进行验证, 成功后往 Mifare 卡中相应块写入值块格式的数据 (只要访问条件允许)。该函数在写入数据后会立即进行读操作并进行数据写入正确性判断。

数据块格式描述:

主机→TX522 命令模式:

- FrameLen: 0x0d
- CTYPE: 2
- COMMAND: 0x53
- LENGTH: 0x07
- DATA[0]: KEYAB
- DATA[1]: Key\_Sector
- DATA[2]: Block
- DATA[3]: Value(LL)
- DATA[4]: Value(LH)
- DATA[5]: Value(HL)
- DATA[6]: Value(HH)

例如: 使用模块内部密匙 0 区的密匙 A 进行验证, 将块 1 写入值块 0x00 0x00 0x55 0x11(低位), 相当于数值 21777 的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x0d	0x02	0x53	0x07	0x60 0x00 0x01 0x11 0x55 0x00 0x00	0Xxx	0x03

TX522→主机 (响应模式):

- FrameLen: 0x06
- CTYPE: 2
- STATUS: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个
- LENGTH: 0
- DATA[0]: 无

例如: 数据帧

FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x06	0x02	0x00	0x00	none	0xFB	0x03

### 5.2.14 带验证的读值块操作—ReadValue\_Auth (Cmd = 'T')

函数原型: uchar TX\_ReadValue\_Auth(uchar KeyAB, uchar Key\_Sector, uchar Block, uchar idata \*Value);

输入参数:

- (1) KEYAB--密钥 AB (1 字节):           0x60——密钥 A  
  0x61——密钥 B
- (2) Key\_Sector: 模块内的密钥区号 (1 字节): 取值范围 0~15。
- (3) Block--卡块号 (1 字节):            S50: 1~63  
  S70: 1~255

**Block=当前扇区\*4+当前扇区的块号**

输出参数: \*Value: Value 为读回 4 字节数据的首地址。

函数返回: 执行后可能返回: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个。

功能描述: 该函数在 Get\_CardSnr 后执行, 先对卡内某一块进行验证, 然后读出值块。注意读出值块时, 会对值块的正确性进行判断, 如果值块不合法, 返回错误, 不会返回值块数据。

数据块格式描述:

主机→TX522 命令模式:

FrameLen: 0x09  
CTYPE: 2  
COMMAND: 0x54  
LENGTH: 3  
DATA[0]: KEYAB  
DATA[1]: Key\_Sector  
DATA[2]: Block

例如: 使用模块内部密钥 0 区的密钥 A 进行验证, 读块 1 的值块的数据帧

FRAMELEN	CTYPE	CMD	Length	DATA	BCC	ETX
0x09	0x02	0x54	0x03	0x00 0x00 0x01	0Xxx	0x03

TX522→主机 (响应模式):

FrameLen: 0x0a  
CTYPE: 2  
STATUS: OK, QUIT, NO\_TAG\_ERR, BIT\_COUNT\_ERR, TRANS\_ERR, CODE\_ERR, COMM\_RERR 中的某一个

LENGTH: 4  
DATA[0]: Value(LL)  
DATA[1]: Value(LH)  
DATA[2]: Value(HL)  
DATA[3]: Value(HH)

例如：数据帧

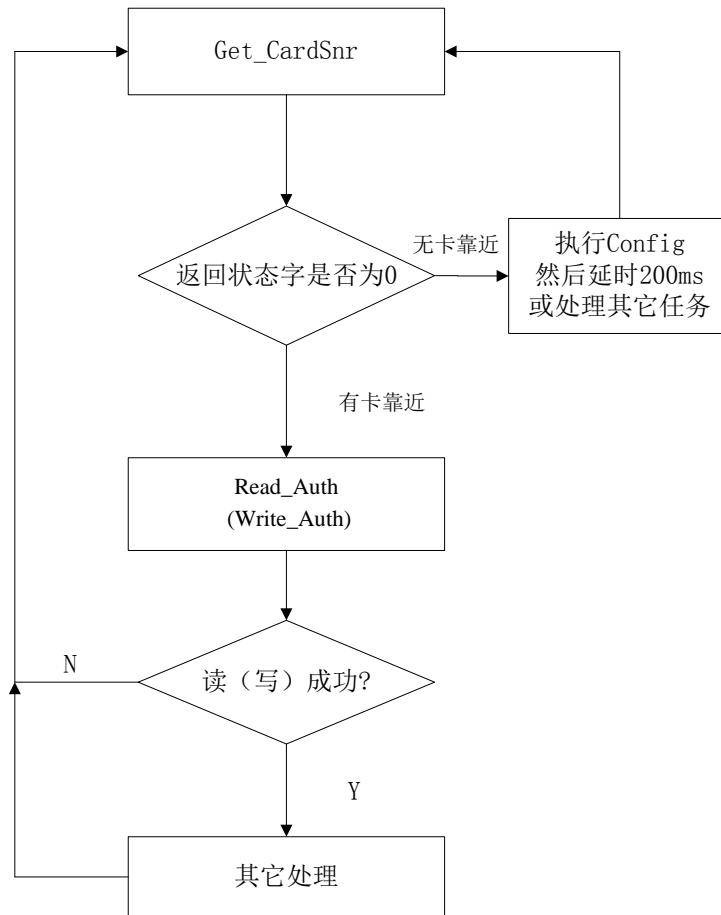
FRAMELEN	CTYPE	Status	Length	DATA	BCC	ETX
0x0a	0x02	0x00	0x04	0x0E 0x55 0x00 0x00	0xXX	0x03

### 5.3 函数调用描述

要确保 Load\_Key 函数正确执行。考虑到安全性，要保证密钥在产品使用过程中不在通信过程中传输，可以将 Load\_Key 在用户程序以外提前进行，可以使用我们公司提供的密钥装载工具在 PC 机上完成。Mifare 卡默认密钥为 0xffffffff，默认密钥也要执行装载操作。

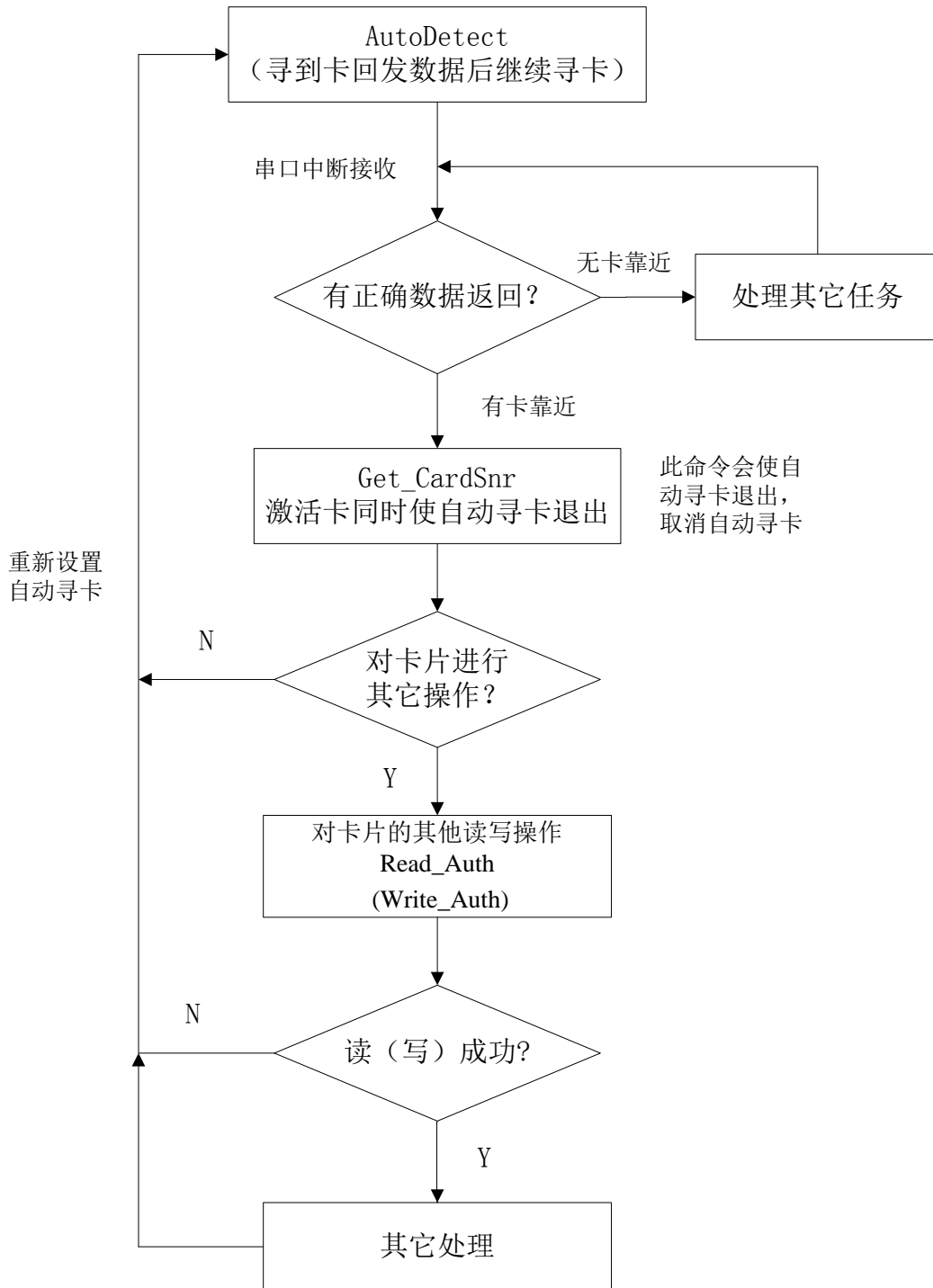
关于对卡片操作，有两种方法。

方法一，用户主程序定时（比方间隔 250ms）循环发送 GetCardSnr，依据返回值判断是否有卡片靠近。如果有卡靠近再做相应处理。



方法二，用户通过 `Auto_Detect` 命令将模块配置为自动寻卡状态，由模块主动寻卡。如果卡片靠近，模块会返回数据给用户的主控器，为了防止返回数据的遗漏，请尽量采用串口中断接收，并将串口接收中断优先级设为最高。建议用户配置为自动寻卡后继续寻卡，然后通过发命令来停止寻卡。如果配置为寻卡后继续寻卡，则可以通过发送 `Get_CardSnr` 命令取消自动寻卡同时激活卡片，然后再进行对卡片的读写操作。在对卡片完成所有操作后，再重新将模块配置为自动寻卡。

注意自动寻卡只能读取一个块的内容，如果要读取多块内容，需要在取消自动寻卡后再调用读命令读取其它块。



## 6. 免责声明

- **开发预备知识**

TX®系列产品将提供尽可能全面的开发模板、驱动程序及其应用说明文档以方便用户使用，但也需要用户熟悉自己设计产品所采用的硬件平台及相关 C 语言的知识。

- **EMI 与 EMC**

TX®系列模块机械结构决定了其 EMI 性能必然与一体化电路设计有所差异。TX®系列模块的 EMI 能满足绝大部分应用场合，用户如有特殊要求，必须事先与我们协商。

TX®系列模块的 EMC 性能与用户底板的设计密切相关，尤其是电源电路、I/O 隔离、复位电路，用户在设计底板时必须充分考虑以上因素。我们将努力完善 TX®系列模块的电磁兼容特性，但不对用户最终应用产品 EMC 性能提供任何保证。

- **修改文档的权利**

东莞同欣智能保留任何时候在不事先声明的情况下对 TX®系列产品相关文档的修改权力。

- **ESD 静电放电保护**

TX®系列产品部分元器件内置 ESD 保护电路，但在使用环境恶劣的场合，依然建议用户在设计底板时提供 ESD 保护措施，特别是电源与 I/O 设计，以保证产品的稳定运行。安装 TX®系列产品，为确保安全请先将积累在身体上的静电释放，例如佩戴可靠接地的静电环，触摸接入大地的自来水管等。



## 7. 修订历史

版本	日期	原因
V1.00	2009/04/01	创建文档。
V1.01	2009/06/01	1. 修改典型电路原理图 2. 增加免责声明部分 3.
V1.02	2009/07/11	1. 去掉了一下与 TX500 兼容，但无用的函数 2. 固定波特率。
V1.03	2009/08/11	1. 修改了图 2，蜂鸣器和 LED 供电改为 5V，并增加续流二极管。
V1.04	2010/07/05	增加双天线配置命令
V1.05	2010/02/25	1. 增加了取消自动寻卡说明 2. 增加了 Leadkey 只用装载一次的说明 3. 去掉了一下被组合命令替换的命令说明。
V1.06	2012/06/07	1. 模块返回状态字改为 16 进制 2. 电源电压改为 3.5-9V 3. TX_ConfigAnt(oid)更正为 TX_ConfigAnt(TXMode); 4. Block=当前扇区*4+当前扇区的块号 5. 带验证的例句: KeyA=0x00 改为 KeyA=0x60 6. 增加了函数调用说明中操作流程图中。
V1.07	2013.01.24	1. 取消了 Read_ADDT 命令 2.

## 8. 销售信息

东莞市同欣智能科技有限公司

地 址：广东省东莞市石碣镇沙腰管理区林屋洲

邮 编：523292

销售电话：0769-86019851-168; 13652608930 QQ:872089468

技术支持：0769-86019851-138; 0769-86019853; 0769-86019851-258;  
18666865339 ; 18664019683 QQ: 14754020; 185792657

传 真：0769-86019852

网 址：[http:// www.TXRFID.com](http://www.TXRFID.com)

E-mail: [sales@TXRFID.com](mailto:sales@TXRFID.com) [support@TXRFID.com](mailto:support@TXRFID.com)