



关键词

- TX522B 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。

本文详细介绍了 TX522B 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. TX522HTW 简介	5
2.1 引脚描述.....	5
2.2 典型电路.....	5
2.3 技术参数.....	6
2.4 极限参数.....	6
2.5 直流特性.....	6
3. TX522B 读卡模块数据传输协议.....	7
3.1 串口协议.....	7
3.2 控制字符定义.....	7
3.3 协议描述.....	7
3.3.1 数据通信帧描述.....	7
3.3.2 主机发送命令至模块.....	8
3.3.3 从机返回数据给主机.....	8
4. TX522B 的 C51 函数.....	9
4.1 函数列表.....	9
4.2 函数返回状态值列表.....	9
4.3 函数描述.....	10
4.3.1 配置—Config.....	10
4.3.2 关闭—Close.....	11
4.3.3 获取信息—Get Info	11
4.3.4 装载密钥—Load_Key.....	12
4.3.5 激活卡片并获取卡号—Get_CardSnr.....	13
4.3.6 暂停—Halt.....	15
4.3.7 复位—Reset.....	15
4.3.8 证实 2—Auth2.....	16
4.3.9 直接密码证实—Auth_Key	17
4.3.10 写—Write.....	18
4.3.11 读—Read.....	19
4.3.12 带内部自动传送的值操作 1—Value.....	20
4.4 函数调用描述.....	21
5. 使用串口调试助手开发调试 TX522B 模块.....	23

5.1 软硬件准备.....	23
5.2 设置串口调试软件.....	23
5.3 向 TX522B 发送命令.....	23
6. 免责声明.....	25
7. 修订历史.....	26
8. 销售信息.....	26

1. 适用范围

本文对 TX522B 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 TX522B 进行通讯的 TX522B 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 TX522B 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. TX522HTW 简介

TX522HTW 是一个简单的串行读写模块，接口为 CMOS 电平，用于与常用的微处理器（如单片机、ARM）的 UART 接口；当外接 RS232 电平转换电路，能够与 PC 机的串口连接通信。

用户必须通过主机（包括单片机、ARM、DSP 以及 PC 机等）向 TX522B 模块发送命令来对 TX522B 进行读写控制。本应用指南将重点描述 TX522B 与主机之间的串行通信协议和命令。

注意：TX522HTW 采用的是 UART 通信协议(即 TX522B 的协议)，本文中也会使用 TX522B 来代替表达 TX522HTW。

2.1 引脚描述

表 1 与用户 MCU 接口

符号	IO 类型	功能描述	上电后的状态
STA	输入/I	上电如果一直悬空模块处于工作状态。上电后如果将 IDLE 置高然后再置低，模块会进入空闲模式，功耗 4uA，空闲时不读卡；置高后模块进入工作模式。不用功耗控制时上电后一直悬空即可，或者接高电平。	1
GND	地/Power	地	
TXD	输出/O	UART 发送端	1
RXD	输入/I	UART 接收端	1
VCC	电源/Power	电源正极	

2.2 典型电路

TX522BT 模块可以与任何带有 UART 口的 MCU 接口，图 1 所示为 TX522BT 与 MCS51 单片机 UART 的典型接口。

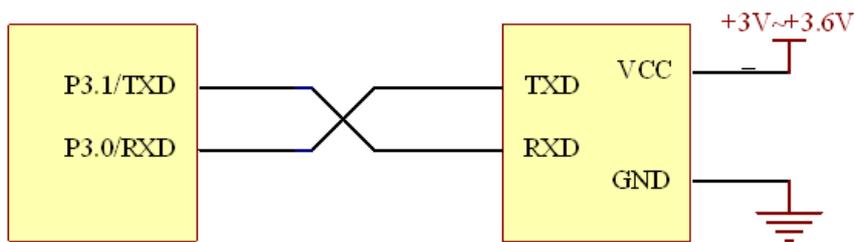


图 1 TX522HTW 与 MCS51 单片机接口图

2.3 技术参数

表 2 TX522 模块技术参数表

功耗	25 毫安/直流 3.3V;
工作频率	13.56 兆赫兹
读卡距离	30~50 毫米 (mifare1 卡)
接口方式	UART
数据传输速率	UART: 9600~230400bit/s
支持卡类型	mifare1 S50、mifare1 S70、mifare UltraLight、mifare Pro
尺寸	天线一体化: 20mm×35mm×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(3.3V)-0.3~+4.0V
 每个管脚的最大 I_{OL} (3.3V).....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

VDD=+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 3 TX522HTW 模块的直流特性

符号	参数	测试条件	最小值	典型值 ⁽¹⁾	最大值	单位
VDD	工作电压		+2.7	+3.0	+3.6	V
T _{OK}	上电后稳定工作时间		5			ms
I _{CC1} (3V)	电流消耗	读卡芯片配置成功		25		mA
I _{CC2} (3V)	电流消耗	关闭读卡芯片		5		mA
V _{IL}	输入低电平		0.22VDD	0.4VDD	-	V
V _{IH}	输入高电平		-	0.6VDD	0.7VDD	V
V _{OL}	输出低电平	I _{OL} =20mA		0.6	1.0	V
		I _{OL} =3.2mA		0.2	0.3	V
V _{OH}	输出高电平	I _{OH} =-20uA	VDD-0.3	VDD-0.2		V
I _{IL}	逻辑低电平输入电流	V _{pin} =0.4V			-80	μA
I _{TL}	逻辑1到0变化电流	V _{pin} =2V	-30		-450	μA
I _{OL}	低电平时的灌电流				-20	mA
I _{OH}	高电平时的拉电流				20	μA
C _{IO}	管脚输入电容				15	pF
T _{OP}	工作温度(I)		-30		+85	℃
T _{STR}	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。

3. TX522B 读卡模块数据传输协议

3.1 串口协议

主机与 TX522B 串行接口通信过程中一帧的数据格式采用 1 个起始位，8 个数据位、无奇偶校验位、1 个停止位。。

3.2 控制字符定义

下表列出了 TX522B 与主机串行通信过程中用到的控制字符定义。

表 4 TX522B 串行通信控制字符表

描述	定义	值
开始符	STX	0x20
终止符	ETX	0x03
应答	ACK	0x06
无应答	NAK	0x15

3.3 协议描述

3.3.1 数据通信帧描述

数据通信以一帧为单位进行，格式如下：

表 5 数据通信帧结构

起始符 STX	包号 SEQNR	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 6 所示：

表 6 数据帧各字段说明

字段	长度	说明	补充
STX	1	STX=0x20，数据帧的起始符，每一帧数据都是以 STX 开始	
SEQNR	1	该数据帧包序号，从 0 到 255 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。	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。从包号（SEQNR）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	

为了下文的描述，把 SEQ+CMD/Status+LEN+DATA+BCC 同称为数据块 DATA Block。

表 7 数据块 DATA BLOCK

DATA BLOCK	包号 SEQNR	命令/状态 CMD/Status	数据长度 Length	数据 DATA	校验和 BCC
	1byte	1byte	1byte	N bytes	1byte

数据帧接收规则：

- 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 7 字节，最大不能超过 64 字节，且帧长必须等于数据长度加 6；
- BCC 计算必须正确。

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。

3.3.2 主机发送命令至模块

通信必须先由主机发送命令和数据给 TX522B，TX522B 执行命令完毕后，再将执行命令后的状态和响应数据发回给主机。

主机发送格式如表 8 所示。

表 8 主机发送给 TX522B 遵循的格式

主机	数据传送方向	TX522B	说明
STX+DATA BLOCK+ETX	→		一次将一帧数据全部发送完。

用户在给 TX522B 模块发送命令时，连续的发送 STX (0x20)+数据块+0x03(结束符)。通过判断 TX522B 返回数据的正确性来判断 TX522B 是否正确执行了本条命令。如果 TX522B 系列模块在接收到按照上面格式发送的命令后，会在接收到数据帧第一字节 STX 后给出应答 0x06，用户可以不用理会。

3.3.3 从机返回数据给主机

TX522B 在接收到主机发送的数据后，根据数据块的内容执行相应命令，并将执行命令后的状态或者数据（以下将状态和数据都统称为数据）返回给主机。

TX522B 发送格式如表 9 所示：

表 9 TX522B 发送格式

TX522B	数据传送方向	主机	说明
STX	→		若 TX522B 在 20ms 内未收到主机回应 ACK，直接发送 DATA BLOCK+ETX
	←	ACK	
DATA BLOCK+ETX	→		

TX522B 返回数据给主机时，首先发送起始符 STX 给主机，并等待主机返回应答信号 ACK。TX522B 在接收到主机返回的 ACK 时，会立即将需要返回的数据块发出。如果主机在 20ms 内一直不应答 ACK，那么 20ms 后，TX522B 也会将数据块发回给主机。

在设计主机程序时，如果对时间要求比较严格，给 TX522B 发送数据或命令结束后，在等待接收 TX522B 返回数据时，先检测 TX522B 发出的 STX 信号并立即给出应答 ACK，然后接收 TX522B 发回的数据块。这种方式可以最快的接收 TX522B 的数据，提高实时性。

由于主机不给 ACK 信号，在 TX522B 接收命令后发出返回数据启动信号 STX 20ms 后也会将数据块发出，因此主机也可以在向 TX522B 发送命令后，不做任何检测和应答而直接等待 20ms 后的应答。这对于使用串口调试助手来调试 TX522B 特别有用。

4. TX522B 的 C51 函数

TX522B 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 10 TX522B 提供的串行通讯库函数列表

函数名称	命令值	输入参数(发送)	输出参数(接收)	功能描述
TX_Config	0x52	--	--	复位且配置模块
TX_Close	0x3F	--	--	关闭模块
TX_Get_Info	0x4F	--	*Info	读取模块信息
TX_Load_Key	0x4C	KeyAB,Sector,*Key	--	改变存贮在模块内密匙区中的密匙
TX_Get_CardSnr	0x10	ReqCode,*TagType,*Sak,*SnrLen,*Snr	--	激活卡片并获取卡号
TX_Halt	0x45	--	--	将卡置于挂起模式
TX_Reset	0x4E	Msec	--	关闭天线输出数 ms，使卡复位
TX_Request	0x41	ReqCode	*TagType	请求卡，检查在有效范围内是否有卡
TX_Casc_Anticoll	0x74	Bcnt, Select_Code	*SNR	可实现三层防碰撞协议
TX_Casc_Select	0x75	Select_Code,*SNR	*Sak	可实现三层选择
TX_Auth2	0x72	KeyAB,Sector,Key_Sector	--	使用模块内部密匙区中的密码对指定的卡的扇区 Sector 进行验证
TX_Auth_Key	0x73	KeyAB,Sector,*Key	--	直接密码验证
TX_Write	0x47	Block,*Data	--	向卡中指定块写入一 16 字节的数据块
TX_Read	0x46	Block	*Data	从卡中指定块中读出一个 16 字节的块
TX_Value	0x70	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

表 10 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。

除表 10 提供的通过串口与 TX522B 通讯的库函数，表 11 列出了 5 个其它函数。

表 11 TX522B 提供的其它库函数列表

函数名称	输入参数(发送)	输出参数(接收)	功能描述
Delay_50us	num	--	延时 50us 的 num 倍
Serial_Init	Baud_Num	--	串口按指定波特率初始化

4.2 函数返回状态值列表

表 12 TX522 库函数的用到的状态值列表

名称	值	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	10	卡没有验证
BIT_COUNT_ERR	11	从卡中接收到了错误数量的位
BYTE_COUNT_ERR	12	从卡中接收到了错误数量的字节
TRANS_ERR	14	调用 Transfer 函数出错
WRITE_ERR	15	调用 Write 函数出错
INCR_ERR	16	调用 Increment 函数出错
DECR_ERR	17	调用 Decrment 函数出错
READ_ERR	18	调用 Read 函数出错
COLL_ERR	24	冲突错
ACCESS_TIMEOUT	27	访问超时
QUIT	30	上一次了送命令时被打断
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	255	串行通信错误
MI_WRONG_VALUE	123	值块格式错误

4.3 函数描述

下面是 C51 函数声明，包含在头文件“TX_B.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。

4.3.1 配置—Config

函数原型：uchar TX_Config(void);

输入参数：无

输出参数：无

函数返回：**TX522MB** 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，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）：

SEQNR: 0（可自定义）

COMMAND: 0x52

LENGTH: 0

DATA[...]: 无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x52	0x00	none	0xAD	0x03

TX522→主机（响应模式）：

SEQNR: 0
STATUS: OK, QUIT, COMM_ERR 中的某一个
LENGTH: 0
DATA[...]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.2 关闭—Close

函数原型：uchar TX_Close(void);

输入参数：无

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, COMM_ERR。

功能描述：此函数将关闭 TX522B 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 TX522B 重新进行配置。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0（可自定义）
COMMAND: 0x3F
LENGTH: 0
DATA[...]: 无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x3F	0x00	none	0xC0	0x03

TX522→主机（响应模式）：

SEQNR: 0
STATUS: OK, COMM_ERR 中的某一个
LENGTH: 0
DATA[...]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.3 获取信息—Get Info

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’，‘B’，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：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：获取 TX522B 模块的信息。

数据块格式描述：

主机→TX522 命令模式）：

SEQNR: 0（可自定义）

COMMAND: 0x4F

LENGTH: 0

DATA[...]: 无

例如：数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4F	0x00	none	0xB0	0x03

TX522→主机（响应模式）：

SEQNR: 0

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, ‘H’

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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x0a	9 字节产品型号及版本	0xXX	0x03

4.3.4 装载密钥—Load_Key

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 0~7

*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：TX523B 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 E2PROM 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 8 个密码区（区号 0~7），称它为密钥区号 Key_Sector。每个区分密钥 A 和密钥 B 两个，总共 16 个密码。装载成功后，可用该密钥对 Mifare1 卡进行验证。

模块的密码区和 mifare 卡的扇区并不要求有对应关系，即可以用模块 0 区的密钥去验证卡片的所有扇区。如果卡片的所有扇区密钥相同，可以统一装载到 0 区。

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）：

SEQNR: 0（可自定义）

COMMAND: 0x4C

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

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4c	0x08	0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff	0xbb	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK, QUIT, AUTH_ERR, COMM_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.5 激活卡片并获取卡号—Get_CardSnr

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 取值为 1 或 0

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

ReqCode=1 (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。

*TagType 的含义如**错误! 未找到引用源**。所示:

表 13 *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					

各种常用卡返回的数据如下**错误! 未找到引用源**。所示:

表 14 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		

(2) *Sak: 最后一级选择应答的应答, 各种常用卡返回的*Sak 数据见下表。

表 15 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

(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: 卡片读写操作前, 必须先激活。该命令激活卡片并获取卡片的序列号等信息。

数据块格式描述:

主机→TX522 命令模式):

SEQNR: 0 (可自定义)

COMMAND: 0x10

LENGTH: 1

DATA[0]: ReqCode

例如: 以 IDLE 方式激活卡的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x10	0x01	0x00	0xee	0x03

TX522→主机 (响应模式):

SEQNR: 0

STATUS: OK, QUIT, COMM_ERR 中的一个

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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x0b	0x44 0x00 0x00 0x07 0x42 0x0A 0x7E 0x00 0x00 0x00 0x00	0xXX	0x03

4.3.6 暂停—Halt

函数原型: uchar TX_Halt(void)

输入参数: 无

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: 将天线区所选择卡置为挂起状态。如果要进行重新选择, 则应用 ALL 模式调用 TX_Request 命令。如果要进行重新选择, 也可以将卡离开天线操作区再进入, 或执行复位函数 TX_Reset()。

可以配合使用 TX_Request()和 TX_Halt()函数, 进行一次性扣费, 如卡进入感应区后只扣一次 (一元钱), 离开后, 下次进入再扣一次, 若卡在感应区内停留时间较长, 也不会扣多一次。

数据块格式描述:

主机→TX522 命令模式):

SEQNR: 0 (可自定义)

COMMAND: 0x45

LENGTH: 0

DATA[0]: 无

例如: 数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x45	0x00	none	0xba	0x03

TX522→主机 (响应模式):

SEQNR: 0

STATUS: OK, QUIT, COMM_ERR 中的某一个

LENGTH: 0

DATA[0]: 无

例如: 数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.7 复位—Reset

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: TX522B 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):

SEQNR: 0 (可自定义)

COMMAND: 0x4E

LENGTH: 1

DATA[0]: Msec

例如: 将天线信号关闭 1ms 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x4e	0x01	0x01	0xb0	0x03

TX522→主机 (响应模式):

SEQNR: 0

STATUS: `OK`, `QUIT`, `COMM_ERR` 中的某一个

LENGTH: 0

DATA[0]: 无

例如: 数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.8 证实 2—Auth2

函数原型: `uchar TX_Auth2(uchar KeyAB, uchar Sector, uchar Key_Sector);`

输入参数: **KeyAB**: 密钥类型 (1 字节)。可取值为 `KeyAB=0x00` (`KEYA`), 利用密钥 A 进行验证, 或 `KeyAB=0x04` (`KEYB`), 利用密钥 B 进行验证。

Sector: 所要验证的卡扇区号 (也即将要访问的卡的扇区号), 取值范围 0~39, 能用于 S70 卡。

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

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`, `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扇区 `Sector` 进行

验证，若卡 Sector 区中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。

数据块格式描述：

主机→TX522 命令模式)：

SEQNR: 0 (可自定义)

COMMAND: 0x72

LENGTH: 3

DATA[0]: KeyAB

DATA[1]: Sector

DATA[2]: Key_Sector

例如：用密钥 0 区的密钥 A 证实卡的扇区 0 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x72	0x03	0x00 0x00 0x00	0x8e	0x03

TX522→主机 (响应模式)：

SEQNR: 0

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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.9 直接密码证实—Auth_Key

函数原型：uchar TX_Auth_Key (uchar KeyAB, uchar Sector, uchar idata *Key);

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

Sector: 所要验证的卡扇区号 (也即将要访问的卡的扇区号)，取值范围 0~39，能用于 S70 卡。

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

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, 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）：

SEQNR: 0（可自定义）

COMMAND: 0x73

LENGTH: 8

DATA[0]: KeyAB

DATA[1]: Sector

DATA[2]: Key[0]

...

DATA[7]: Key[5]

例如：用密码 0xff 0xff 0xff 0xff 0xff 0xff 证实扇区 0 的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x73	0x08	0x00 0x00 0xff 0xff 0xff 0xff 0xff 0xff	0x84	0x03

TX522→主机（响应模式）：

SEQNR: 0

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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.3.10 写—Write

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

*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：TX522B 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)：
 SEQNR: 0 (可自定义)
 COMMAND: 0x47
 LENGTH: 17
 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

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x47	0x11	0x02 16 字节数据	0xXX	0x03

TX522→主机 (响应模式)：
 SEQNR: 0
 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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	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。

4.3.11 读—Read

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

输出参数：*Data: Data 为读回 16 字节数据的首地址。

函数返回：TX522B 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）：

SEQNR: 0（可自定义）

COMMAND: 0x46

LENGTH: 1

DATA[0]: Block

例如：读取块 2 的数据的数据帧

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x46	0x01	0x02	0xba	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK, QUIT, NO_TAG_ERR, CRC_ERR, NOT_AUTH_ERR, PARITY_ERR, BIT_COUNT_ERR, COMM_ERR 中的某一个

LENGTH: 16

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

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x10	16 字节数据	0xXX	0x03

4.3.12 带内部自动传送的值操作 1—Value

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

*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：TX522B 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

输入块内的数据被覆盖，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）：

SEQNR: 0（可自定义）

COMMAND: 0x70

LENGTH: 7

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

STX	SEQNR	CMD	Length	DATA	BCC	ETX
0x20	0x00	0x70	0x07	0xc0 0x01 0x03 0x00 0x00 0x00 0x02	0x48	0x03

TX522→主机（响应模式）：

SEQNR: 0

STATUS: OK, QUIT, NO_TAG_ERR, BIT_COUNT_ERR, TRANS_ERR, CODE_ERR, COMM_RERR 中的某一个

LENGTH: 0

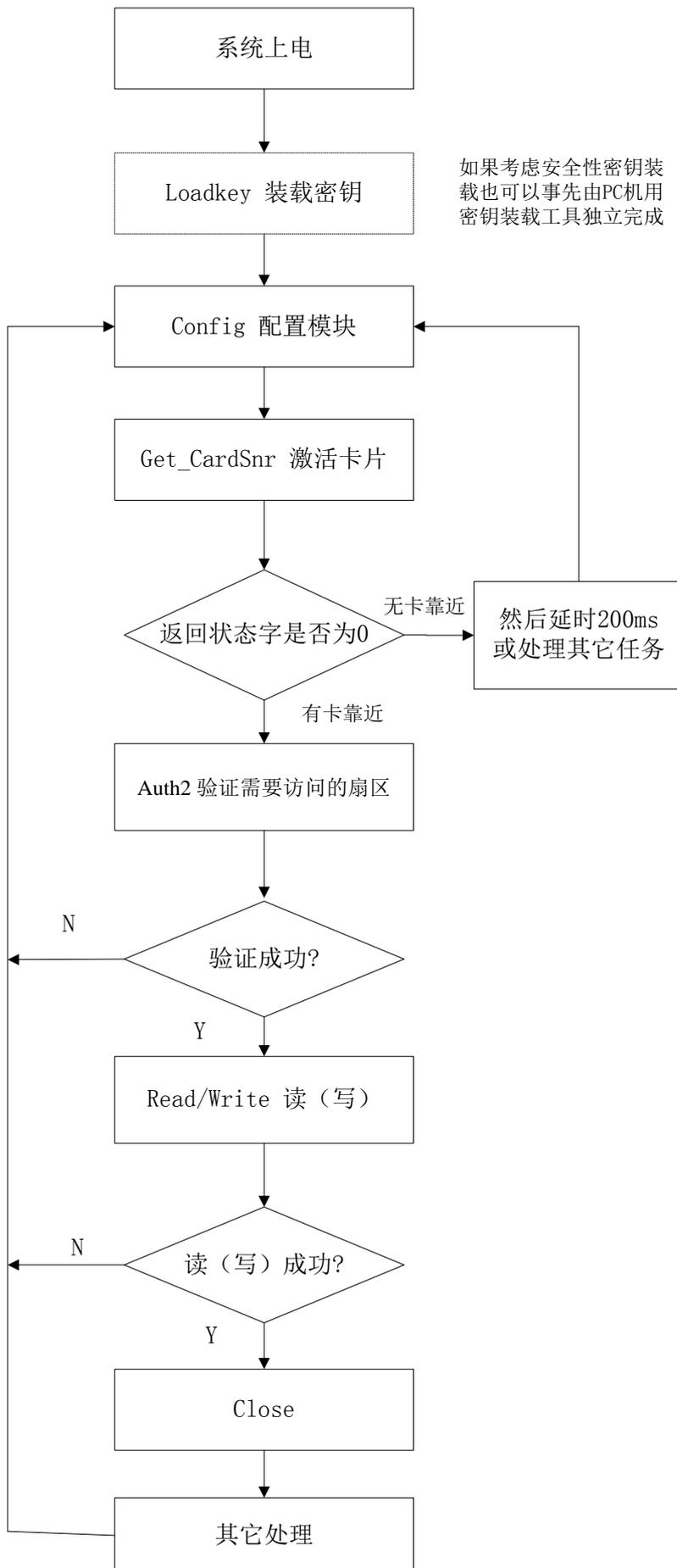
DATA[0]: 无

例如：数据帧

STX	SEQNR	Status	Length	DATA	BCC	ETX
0x20	0x00	0x00	0x00	none	0xFF	0x03

4.4 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。



5. 使用串口调试助手开发调试 TX522B 模块

5.1 软硬件准备

1. TX522HTW 模块一块，外接 RS232 转换
2. 串口电缆
3. 3.3V 电源
4. 5 根连线
5. Mifare 卡一张
6. 串口调试软件
7. 串口监视软件（选）
8. PC 机

5.2 设置串口调试软件

按照图 2 所示设置串口调试软件。

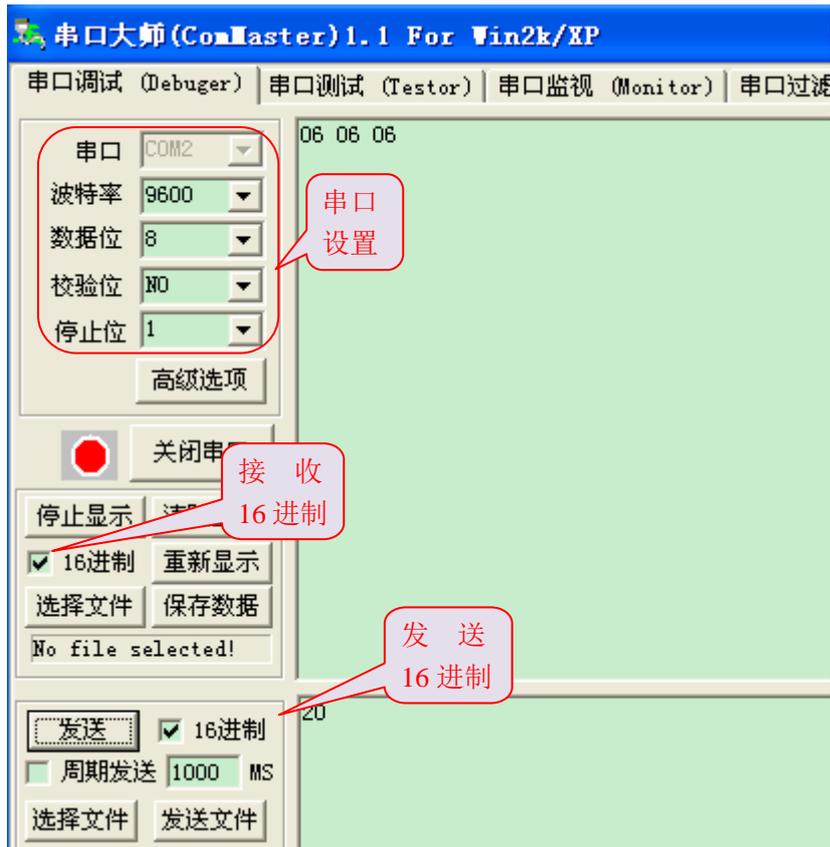


图 2 串口调试软件的设置

5.3 向 TX522B 发送命令

发送命令的格式为：起始符(STX: 0x20) + 包序号(一般设 00 即可) + 命令字 + 数据长度 + 数据 + BCC 校验 + 结束符(EXT: 03)。详细说明见前文。

《TX522B 命令集_V1.00.txt》列出了一些常用的命令，一些可以直接拷贝到串口调试软件的发送窗口中直接进行发送。有一些需要用户进行一定的修改，例如使用的卡片不同，发送选择命令（select）时应填入相应的卡号。注意如果数据有些不同，则 BCC 校验也不同。

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	2009/08/11	1. 去掉了一下与 TX500 兼容，但无用的函数 2. 固定波特率。
V1.03	2011/09/13	1. 增加了功耗控制管脚 STA

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; 18666865339 QQ: 14754020

传 真：0769-86019852

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

E-mail: sales@TXRFID.com support@TXRFID.com