

UP-RFID-RT 型综合教学平台

13.56MHz 模块串口协议

(15693)

博创智联科技有限公司

2016-08-10

13. 56MHz_15639 串口通讯协议

一、 指令集合

这部分主要介绍 13.56MHz 模块支持的 15693 协议下的指令汇总。按照 13.56MHz 模块的工作端分成两个部分介绍，ISO15693 协议指令是 13.56MHz 模块和 15693 标签之间的通信协议指令，串口协议指令是 13.56MHz 模块与串口终端（如 PC 端、ARM 端）之间的通信协议指令。

1. ISO15693 协议指令

表 1.1 ISO15693 协议指令

指令标识	指令代码 (Hex)	类型
寻卡	01	强制的
静止	02	强制的
RFU	03~1F	强制的
读单一块	20	可选的
写单一块	21	可选的
锁定块	22	可选的
读多重块	23	可选的
写多重块	24	可选的
选择	25	可选的
重启准备	26	可选的
写 AFI	27	可选的
锁定 AFI	28	可选的
写 DSFID	29	可选的
锁定 DSFID	2A	可选的
获取系统信息	2B	可选的
获得多重块安全状态	2C	可选的
RFU	2D~9F	可选的
IC 生产厂家确定	A0~DF	自定义的
IC 生产厂家确定	E0~FF	专用的

2. 串口协议指令

表 1.2 串口协议指令

指令标识	指令代码 (Hex)	功能
INVENTORY16	1000	寻卡（可寻多张）
INVENTORY	1001	寻卡（不建议直接使用）
STAY_QUIET	1002	设置静默态
SELECT	1003	选择
RESET_TO_READY	1004	重置到准备态
READ_SM	1005	读单个或者多个数据块
WRITE_SM	1006	写单个数据块
LOCK_BLOCK	1007	锁定数据块
WRITE_AFI	1008	写应用族标志
LOCK_AFI	1009	锁定应用族标志
WRITE_DSfid	100A	写数据存储格式标志
LOCK_DSfid	100B	锁定数据存储格式标志
GET_SYSINFO	100C	获取系统信息
GET_MULTIBLOCK_SECURITY	100D	获取多个块的安全状态
GET_HARDMODEL	0104	获取版本号
SET_BAUDRATE	0101	设置波特率

二、指令说明

1. ISO 15693 协议指令说明

ISO 15693 协议使用的是国际标准，详细说明参照 HF13.56M/国际标准/ISO15693 部分的文档。

2. 串口协议指令说明

13.56MHz 模块作为其他主机（如 PC）的外设通过串口通信，串口默认波特率为 19200。此处详细说明 13.56MHz 模块 15693 协议下的串口指令以及各个指令帧结构。模块接收到的命令帧结构大致分为 6 部分，主要分为帧头、校验和、参数以及命令字，具体如表 2.1 所示。

表 2.1 命令帧数据结构

字段	值	说明
SOF	0xAABB	主机（如 PC）与 13.56MHz 模块通信的命令帧起始字
Lenth	0XXXXX	一帧命令的长度，从 DEV_ID（包含）开始到 FCS（包含）结束为止的总长度
Dev_ID	0x0000	设备编号，用于扩展使用
CMD	0XXXXX	命令字，如寻卡命令字为 0x1000
Status/Flag	0xXX	状态字节或/和标志字节，主要在响应帧中出现
VData	...	可变字节的负载，发送时可以携带参数，接收时为响应数据
FCS	0xXX	校验和，从 LENGTH（不包含）开始到 FCS（不包含）结束的所有字节的异或值

注：请求和响应帧结构一致，最后校验和是字节，VDATA 也是按照字节算的，其余均为字，低位在前，高位在后。

2.1 INVENTORY16 命令

INVENTORY16 命令字是 0x1000，当 13.56MHz 模块通过串口接收到 INVENTORY16 命令后执行 ISO 15693 协议下的寻卡命令。13.56MHz 模块可以寻找到处在天线辐射范围内的未休眠的标签。INVENTORY16 命令帧结构如表 2.2 所示。

表 2.2 INVENTORY16 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	FCS
AA BB	05 00	00 00	00 10	10

13.56MHz 模块接收到请求命令并寻卡后会将寻到的卡号和执行的结果状态结果反馈给请求方(如 PC)，响应帧结构如表 2.3 所示。

表 2.3 INVENTORY16 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	VData	FCS
AA BB	0F 00	00 00	00 10	00 15	UID	fcs

表 2.3 中，Lenth 字段不一定是 0x000F，如果有多张卡，Lenth 的长度是 7+8*标签的数量，Status/Flag 第一个字节为状态字，用于表明寻卡成功与否，第二个字节为数据存储格式标志，例如 DSFID 为 15，则响应帧的 Status/Flag 字段第二个字节就是 15，VData 部分为卡号，8 字节，如果有多张卡，VData 会以 8 字节为单位向后扩展。后续的命令大部分对特定标签进行操作，因此首先需要寻卡，然后才能进一步操作。

示例：

操作	交互帧
Send	AA BB 05 00 00 00 00 10 10
Recv	AA BB 0F 00 00 00 00 10 00 45 76 9D 97 29 00 01 04 E0 E5

2.2 STAY_QUIET 命令

STAY_QUIET 命令字是 0x1002, 当 13.56MHz 模块通过串口接收到 STAY_QUIET 命令后执行 ISO 15693 协议下的静止命令，处于静默态的标签不处理任何寻卡请求命令。当模块断电重启和接收到 SELECT、RESET_TO_READY 命令时退出静默状态进入到对应的选择态或者准备态。13.56MHz 模块接收到 STAY_QUIET 命令帧结构如表 2.4 所示。

表 2.4 STAY_QUIET 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData	FCS
AA BB	0D 00	00 00	02 10	UID	fcs

13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.5 所示。

表 2.5 STAY_QUIET 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status	FCS
AA BB	06 00	00 00	02 10	XX	fcs

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0）
Send	AA BB 0D 00 00 00 02 10 76 9D 97 29 00 01 04 E0 A2
Recv	AA BB 06 00 00 00 02 10 00 12

2.3 SELECT

SELECT 命令字是 0x1003, 当 13.56MHz 模块通过串口接收到 SELECT 命令后执行 ISO 15693 协议下的选择命令，处于静默态或准备态的标签会进入到选择态。处于选择态的标签可以执行该标签支持的任何指令。13.56MHz 模块接收到 SELECT 命令帧结构如表 2.6 所示。

表 2.6 SELECT 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData	FCS
AA BB	0D 00	00 00	03 10	UID	fcs

13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.7 所示。

表 2.7 SELECT 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status	FCS
AA BB	06 00	00 00	03 10	XX	fcs

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0）
Send	AA BB 0D 00 00 00 03 10 76 9D 97 29 00 01 04 E0 A3
Recv	AA BB 06 00 00 00 03 10 00 13

2.4 RESET_TO_READY 命令

RESET_TO_READY 命令字是 0x1004，当 13.56MHz 模块通过串口接收到 RESET_TO_READY 命令后执行 ISO 15693 协议下的重启到准备态命令。处于选择态或静默态的标签通过该命令可以进入到准备态。13.56MHz 模块接收到 RESET_TO_READY 命令帧结构如表 2.8 所示。

表 2.8 RESET_TO_READY 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData	FCS
AA BB	0E 00	00 00	04 10	02 UID	fcs

在表 2.8 中，VData 中 UID 前面有一个标志字节 02，用于表明是地址请求标志还是选择请求标志，这里固定为 02（地址请求标志）。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.9 所示。

表 2.9 RESET_TO_READY 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status	FCS
AA BB	06 00	00 00	04 10	XX	fcs

示例：

TEL: 010-82110740 82110741 82110742 82110743
FAX: 总机转 828

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0）
Send	AA BB 0E 00 00 00 04 10 02 76 9D 97 29 00 01 04 E0 A6
Recv	AA BB 06 00 00 00 04 10 00 14

2.5 READ_SM 命令

READ_SM 命令字是 0x1005，当 13.56MHz 模块通过串口接收到 READ_SM 命令后执行 ISO 15693 协议下的读取单个或多个块数据命令。读取单个或者多个数据块通过 VData 中的参数来表明是读取单个块还是多个块，13.56MHz 模块接收到 READ_SM 命令帧结构如表 2.10 所示。

表 2.10 READ_SM 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData				FCS
AA BB	0E 00	00 00	05 10	Flag	UID	Addr	Nblock	fcs
				02	Uid	BlockAddr	n	

在表 2.10 请求帧中，VData 分为四个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行读取操作；第三部分是要读取的块起始地址（Address），表明是从哪块开始读取；第四部分是读取块的数量，当 $n = 1$ 时读单块，当 $n > 1$ 是读多个块，从 Addr 给定的地址开始连续读取 n 块的数据。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.11 所示。

表 2.11 READ_SM 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status	VData	FCS
AA BB	0A 00	00 00	05 10	XX	Data	fcs

读取单个或者多个数据块响应帧如表 2.11 所示，Status 为读取单个或多个块的成功与否的状态，VData 为该块开始、请求读取的 n 块数据。每个块 4 个字节，因此此处 Data 的长度是 4 的整数倍，数据从左向右按照块地址从低到高排列。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，读取块 3 的数据，响应帧数据内容 12 23 56 78）
Send	AA BB 10 00 00 00 05 10 02 76 9D 97 29 00 01 04 E0 03 01 A5
Recv	AA BB 0A 00 00 00 05 10 00 12 34 56 78 1D

2.6 WRITE_SM 命令

WRITE_SM 命令字是 0x1006，当 13.56MHz 模块通过串口接收到 WRITE_SM 命令后执行 ISO 15693 协议下的写入单个块数据命令。13.56MHz 模块接收到 WRITE_SM 命令帧结构如表 2.12 所示。

表 2.12 WRITE_SM 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData				FCS
AA BB	13 00	00 00	06 10	Flag	UID	Addr	Data	fcs
				02	Uid	BlockAddr	Data	

在表 2.12 请求帧中，VData 分为四个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行写入操作；第三部分是要写入的地址（Address）；第四部分是写入的数据（Data），每个块长度为四个字节，因此 Data 也为四个字节。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.13 所示。

表 2.13 WRITE_SM 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	06 10	XX	fcs

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，写入块 8 的数据，数据内容 12 23 56 78）
Send	AA BB 13 00 00 00 06 10 02 76 9D 97 29 00 01 04 E0 08 12 34 56 78 A4
Recv	AA BB 06 00 00 00 06 10 00 16

2.7 LOCK_BLOCK 命令

LOCK_BLOCK 命令字是 0x1007，当 13.56MHz 模块通过串口接收到 LOCK_BLOCK 命令后执行 ISO 15693 协议下的锁定数据块命令。锁定数据块需谨慎操作，数据块锁定之后不能解锁，而一旦锁定后该数据块不能再次写入。锁定命令适合使用在发放标签时写入一些只读的特殊信息。13.56MHz 模块接收到 LOCK_BLOCK 命令帧结构如表 2.14 所示。

表 2.14 LOCK_BLOCK 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData			FCS
AA BB	0F 00	00 00	07 10	Flag	UID	Addr	fcs

02	Uid	BlockAddr
----	-----	-----------

在表 2.14 请求帧中，VData 分为三个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行锁定操作；第三部分是要锁定的地址（Address）。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.15 所示。

表 2.15 LOCK_BLOCK 命令响应帧数据

SOE	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	07 10	XX	fcs

表 2.15 中 Status/Flag 为状态标志，如果成功返回 00，失败返回 18，为写入失败。因为锁定数据块实际是写该地址的锁存位，因此当失败时错误还是写卡失败。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，锁定 07 地址）
Send	AA BB 0F 00 00 00 07 10 02 76 9D 97 29 00 01 04 E0 07 A2
Recv	AA BB 06 00 00 00 07 10 00 17

2.8 WRITE_AFI 命令

WRITE_AFI 命令字是 0x1008，当 13.56MHz 模块通过串口接收到 WRITE_AFI 命令后执行 ISO 15693 协议下的写入 AFI 命令。写入 AFI 主要用于给标签设置类型标识。13.56MHz 模块接收到 WRITE_AFI 命令帧结构如表 2.16 所示。

表 2.16 WRITE_AFI 命令请求帧数据

SOE	Lenth	Dev_ID	CMD	VData			FCS
AA BB	0F 00	00 00	08 10	Flag	UID	AFI	fcs
				02	Uid	Afi	

在表 2.16 请求帧中，VData 分为三个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行写入操作；第三部分是要写入的标志（AFI）。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.17 所示。

表 2.17 WRITE_AFI 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	08 10	XX	fcs

表 2.17 中 Status/Flag 为状态标志，如果成功返回 00，失败返回 18，错误码含义为写入失败。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，AFI 为 12）
Send	AA BB 0F 00 00 00 08 10 02 76 9D 97 29 00 01 04 E0 12 B8
Recv	AA BB 06 00 00 00 08 10 00 18

2.9 LOCK_AFI 命令

LOCK_AFI 命令字是 0x1009，当 13.56MHz 模块通过串口接收到 LOCK_AFI 命令后执行 ISO 15693 协议下的锁定 AFI 命令。锁定 AFI 需谨慎操作，AFI 锁定之后不能解锁，而一旦锁定后该卡的 AFI 不能再次写入。锁定 AFI 命令适合使用在发放标签时为特殊的人群或者物品设置标志。13.56MHz 模块接收到 LOCK_AFI 命令帧结构如表 2.18 所示。

表 2.18 LOCK_AFI 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData		FCS
AA BB	0E 00	00 00	09 10	Flag	UID	fcs
				02	Uid	

在表 2.18 请求帧中，VData 分为两个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行锁定操作。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.19 所示。

表 2.19 LOCK_AFI 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	09 10	XX	fcs

表 2.19 中 Status/Flag 为状态标志，如果成功返回 00，失败返回 18，为写入失败。因为锁定 AFI 实际是写 AFI 的锁存位，因此当失败时错误还是写卡失败。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0）
Send	AA BB 0E 00 00 00 09 10 02 76 9D 97 29 00 01 04 E0 AB
Recv	AA BB 06 00 00 00 09 10 00 18

2.10 WRITE_DSFD 命令

WRITE_DSFD 命令字是 0x100A，当 13.56MHz 模块通过串口接收到 WRITE_DSFD 命令后执行 ISO 15693 协议下的写入数据存储格式标志（DSFD）命令。13.56MHz 模块接收到 WRITE_DSFD 命令帧结构如表 2.20 所示。在表 2.20 请求帧中，VData 分为三个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行写入操作；第三部分是要写入的标志（DSFD）。DSFD 位没有锁定时可以反复写入，已经锁定则不能再次写入。

表 2.20 WRITE_DSFD 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData			FCS
AA BB	0F 00	00 00	0A 10	Flag	UID	DSFD	fcs
				02	Uid	Flag	

13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.21 所示。

表 2.21 WRITE_DSFD 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	0A 10	XX	fcs

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，DSFD 为 45）
Send	AA BB 0F 00 00 00 0A 10 02 76 9D 97 29 00 01 04 E0 45 ED
Recv	AA BB 06 00 00 00 0A 10 00 1A

2.11 LOCK_DSFD 命令

LOCK_DSFD 命令字是 0x100B，当 13.56MHz 模块通过串口接收到 LOCK_DSFD 命令后执行 ISO 15693 协议下的锁定数据存储格式标志命令。DSFD 没有锁定时可以随便写入，但锁定后将不能再次写入，

因此锁定操作需谨慎。DSFID 主要用于标识卡内数据的存储格式或者编码等（如 DSFID = 21 时采用大端模式，DSFID = 12 时采用小端模式）。13.56MHz 模块接收到 LOCK_DSFIID 命令帧结构如表 2.22 所示。

表 2.22 LOCK_DSFIID 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData		FCS
AA BB	0E 00	00 00	0B 10	Flag	UID	fcs
				02	Uid	

在表 2.22 请求帧中，VData 分为两个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是对哪张卡进行锁定操作。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.23 所示。

表 2.23 LOCK_DSFIID 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	0B 10	XX	fcs

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0）
Send	AA BB 0E 00 00 00 0B 10 02 76 9D 97 29 00 01 04 E0 A9
Recv	AA BB 06 00 00 00 0B 10 00 1B

2.12 GET_SYSINFO 命令

GET_SYSINFO 命令字是 0x100C，当 13.56MHz 模块通过串口接收到 GET_SYSINFO 命令后执行 ISO 15693 协议下的获取系统信息命令。13.56MHz 模块接收到 GET_SYSINFO 命令帧结构如表 2.24 所示。

表 2.24 GET_SYSINFO 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData		FCS
AA BB	0E 00	00 00	0C 10	Flag	UID	fcs
				02	Uid	

在表 2.24 请求帧中，VData 分为两个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是后去哪张卡的系统信息。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.25 所示。

表 2.25 GET_SYSINFO 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status	VData						FCS
AA BB	14 00	00 00	0C 10	XX	Flag	UID	DSFID	AFI	VICC	IC	fcs

信息标志

卡号

数据存储格式标志

应用族标志

内存容量信息

IC 参考值

响应帧如表 2.25 所示，VData 部分为获取系统信息后的响应负载，分为六个部分。信息标志表明 DSFID、AFI、VICC 等是否支持，数据存储格式标志、应用族标志分别代表标签类型和内部数据的存储格式，内存容量表明该标签有多少个块，每个块可以存储多少个字节，最后 IC 参考值为标签厂商给定。下面分别介绍信息标志和内存容量部分的含义。信息标志定义如表 2.26 所示。

表 2.26 信息标志定义

Bit	标志名字	值	描述
B0	DSFID	0	不支持 DSFID。DSFID 域不出现
		1	支持 DSFID。DSFID 域出现
B1	AFI	0	不支持 AFI。AFI 域不出现
		1	支持 AFI。AFI 域出现
B2	VICC 内存容量	0	不支持信息的 VICC 内存容量。内存容量域不出现
		1	支持信息的 VICC 内存容量。内存容量域出现
B3	IC 参考	0	不支持信息的 IC 参考。IC 参考域不出现
		1	支持信息的 IC 参考。IC 参考域出现
B4~B7	RFU	0	保留

内存容量信息如表 2.27 所示。

表 2.27 内存容量信息

B15~B13	B12~B8	B8~B0
RFU	块容量	块数

注：块容量信息和块数均比实际的小 1，例如获取到内存容量为 031B，则实际上每块的容量是 4，块的数目是 1C，原因是块容量和块数均从 0 开始，例如块 00 01 02 03，实际是块是 4 块。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，VData: 0F 76 9D 97 29 00 01 04 E0 45 45 1B 03 01）
Send	AA BB 0E 00 00 00 0C 10 02 76 9D 97 29 00 01 04 E0 AE
Recv	AA BB 14 00 00 00 0C 10 00 0F 76 9D 97 29 00 01 04 E0 45 45 1B 03 01 BA

2.13 GET_MULTIBLOCK_SECURITY 命令

GET_MULTIBLOCK_SECURITY 命令字是 0x100D，当 13.56MHz 模块通过串口接收到 GET_MULTIBLOCK_SECURITY 命令后执行 ISO 15693 协议下的获取多个块的安全状态命令。13.56MHz 模块接收到 GET_MULTIBLOCK_SECURITY 命令帧结构如表 2.28 所示。

表 2.28 GET_MULTIBLOCK_SECURITY 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData				FCS
AA BB	10 00	00 00	0D 10	Flag	UID	StartAddr	EndAddr	fcs
				02	Uid	Start	End	

在表 2.28 请求帧中，VData 分为四个部分，第一部分为标志（Flag），固定为 02；第二部分为 UID，表明是需要获取哪张卡的相关信息；第三部分为获取块的起始地址；第四部分为获取块的结束地址。13.56MHz 模块接收到请求命令并执行后会将执行结果状态反馈给请求方（如 PC），响应帧结构如表 2.29 所示。

表 2.29 GET_MULTIBLOCK_SECURITY 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	VData	FCS
AA BB	XX XX	00 00	0D 10	XX	XX ...	fcs

在响应帧中，各个块的安全状态由一个字节表示，如果锁定该字节为 01，未锁定为 00，VData 的长度与获取的多个模块安全状态的起始和结束地址相关。

示例：

操作	交互帧（卡号为 76 9D 97 29 00 01 04 E0，获取 00 到 08 地址的安全状态信息）
Send	AA BB 10 00 00 00 0D 10 02 76 9D 97 29 00 01 04 E0 00 09 A6
Recv	AA BB 0F 00 00 00 0D 10 00 01 00 00 01 01 00 00 01 00 1D

2.14 GET_HARDMODEL 命令

GET_HARDMODEL 命令字是 0x0104，该命令是对模块操作，因此不涉及到 ISO 15693 相关的协议，

有无标签在天线识别范围内对该命令无影响，当 13.56MHz 模块通过串口接收到 GET_HARDMODEL 命令帧结构如表 2.30 所示。

表 2.30 GET_HARDMODEL 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	FCS
AA BB	05 00	00 00	04 01	fcs

13.56MHz 模块获取固件版本信息后将版本信息作为负载反馈给请求端，响应帧结构如表 2.31 所示。

表 2.31 GET_HARDMODEL 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	VData	FCS
AA BB	12 00	00 00	04 01	XX	...	fcs

VData 部分为版本信息的字节数组，无需转换，直接显示即可。操作示例如下：

操作	交互帧
Send	AA BB 05 00 00 00 04 01 05
Recv	AA BB 12 00 00 00 04 01 00 53 4C 36 30 31 46 2D 30 35 31 32 00 40

2.15 SET_BAUDRATE 命令

SET_BAUDRATE 命令字是 0x0101，该命令是对模块操作，因此不涉及到 ISO 15693 相关的协议，有无标签在天线识别范围内对该命令无影响，当 13.56MHz 模块通过串口接收到 SET_BAUDRATE 命令帧结构如表 2.32 所示。

表 2.32 SET_BAUDRATE 命令请求帧数据

SOF	Lenth	Dev_ID	CMD	VData	FCS
AA BB	06 00	00 00	01 01	XX	fcs

在表 2.32 请求帧中，在 VData 部分用一个字节代表一个特定的波特率，13.56MHz 模块支持的波特率如表 2.33 所示。

表 2.33 13.56MHz 模块支持的波特率

编码	0	1	2	3	4	5	6	7
波特率	4800	9600	14400	19200	28800	38400	57600	115200

13.56MHz 模块接收到请求帧，VData 字节为表 2.33 中某一个编码，根据编码将模块波特率设置成对应的波特率，在不断电的情况下一只工作在改波特率下。同时用修改之前的波特率将设置结果响应给请求端，响应帧格式如表 2.34 所示。

表 2.34 SET_BAUDRATE 命令响应帧数据

SOF	Lenth	Dev_ID	CMD	Status/Flag	FCS
AA BB	06 00	00 00	01 10	XX	fcs

注：波特率的设置同时会影响 15693 协议，在不断电的情况下 15693 和 14443A 必须始终采用一致的波特率进行工作。

示例：

操作	交互帧（设置波特率为 115200）
Send	AA BB 06 00 00 00 01 01 07 07
Recv	AA BB 06 00 00 00 01 01 00 00

附录

FCS（Frame check sequence）：

在 15693 协议下，每个命令帧最后会追加一个字节作为该帧的校验和，从 LENGTH（不包含）开始到 FCS（不包含）结束的所有字节的异或值。计算方法（C 语言实现）：

```
//校验值计算
uint8 RC632_UartCalcFCS( uint8 *msg_ptr, uint8 len )
{
    uint8 x;
    uint8 xorResult;
    xorResult = 0;
    for ( x = 0; x < len; x++, msg_ptr++ )
        xorResult = xorResult ^ *msg_ptr;
    return ( xorResult );
}
```

说明：第一个参数是要计算的字节数组的起始地址，第二个参数为要计算的长度。