UP-RFID-RT 型综合教学平台

900MHz 模块串口协议

(840MHz-960MHz)

博创智联科技有限公司

2016-08-18

900MHz 模块串口通讯协议

一、 指令集合

这部分主要介绍 900MHz 模块协议下的指令汇总。按照 900MHz 模块的工作端分成两个部分介绍，ISO 18000-6C/6B 协议指令是 900MHz 模块和 Gen2 标签之间的通信协议指令，串口协议指令是 900MHz 模块与串口终端（如 PC 端、ARM 端）之间的通信协议指令。

1. ISO 18000-6C/6B 协议

ISO 18000-6C/6B 标准定义了应用于物品管理等方面的 RFID 技术标准。工作在 860MHz 到 960MHz 频段之间，适用于工业、科学、医疗等领域。ISO 18000-6B 标准特点：标准成熟、产品稳定、应用广泛；ID 好全球唯一；先读取 ID 号，后读取数据区；1024bits 或 2048bits 的大容量；98Bytes 或 216Bytes 的大用户数据区；支持多标签同时读取（最多同时读取 10 张标签）；数据读取速度为 40Kbps。ISO 18000-6C 标准特点：读取速度快，数据率可以达到 40Kbps~640Kbps；可以同时读取标签的数量相对于 6B 标准更多，理论上可以同时读取 1000 多张；读取时首先读取 EPC 号码，标签 ID 需要用读取数据的方式读取；具有多种写保护方式，安全性更强。

ISO 18000-6B/6C 协议的区别主要如下：6B 一般用在闭环的领域，如烟草等资产管理领域；6C 一般用在开环的领域，比如物流行业的资源管理。6B 和 6C 同时读取标签数量的差异，因此 6B 用户区数据较长，6C 用户区数据相对较短。在标签成本上来说 6C 相对于 6B 的标签来说更便宜。

EPC C1G2 标签采用 ISO 18000-6C 标准，基本可以说 ISO 18000-6C 是 EPC C1G2 的子集，自 2005 年 EPC C1G2 申请成为 ISO 18000-6 标准到 2006 年正式加入到 ISO 18000-6（ISO 18000-6C）标准后，ISO/EPC 18000-6 标准得到了全面的更新与完善（EPC 协议没有 AFI，有 EPC 编号）。

UP-TECH RFID 900MHz 模块采用的 ISO 18000-6C 标准，可以手动调整模块的功率、工作频率，工作频率在 840MHz 到 960MHz 之间均可调，因此可支持中国标准、欧洲标准、定频 915MHz。在实际工作过程中，对模块功率进行调整后可能会修改其工作频率，由于修改工作功率，对标签的识别距离有影响。

2. 串口协议指令

表 1.1 串口协议指令

指令标识	指令代码（Hex）	功能	响应等待时间（ms）
UHFCMD_GET_STATUS	00	询问状态	200
UHFCMD_GET_POWER	01	读取功率	200
UHFCMD_SET_POWER	02	设置功率	200
UHFCMD_GET_FRE	05	读取频率	200
UHFCMD_SET_FRE	06	设置频率	200
UHFCMD_GET_VERSION	07	读取版本信息	200
UHFCMD_INVENTORY	10	识别标签（单标签识别）	200
UHFCMD_INVENTORY_ANTI	11	识别标签（防碰撞识别）	200
UHFCMD_STOP_GET	12	停止操作	200
UHFCMD_READ_DATA	13	读取标签数据	200
UHFCMD_WRITE_DATA	14	写入标签数据	200
UHFCMD_ERASE_DATA	15	擦除标签数据	200
UHFCMD_LOCK_MEM	16	锁定标签	200
UHFCMD_KILL_TAG	17	销毁标签	200
UHFCMD_INVENTORY_SINGLE	18	识别标签（单步识别）	200
UHFCMD_WIEGAND_INVENTORY	19	韦根识别	200
UHFCMD_SINGLE_READ_DATA	20	读取标签数据（不指定 UII）	200
UHFCMD_SINGLE_WRITE_DATA	21	写入标签数据（不指定 UII）	200

二、 指令说明

1. ISO 18000-6C/6B 协议指令说明

ISO 18000-6C/6B 协议使用的是国际标准，详细说明参照 UHF-900M 部分的文档。

2. 串口协议指令说明

900MHz 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 57600。此处详细说明

900MHz 模块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，具体如表 2.1 所示。

表 2.1 命令帧数据结构

字段	值	说明
SOF	0xAA	主机（如 PC）与 900MHz 模块通信的命令帧起始字节，固定为 0xAA
Lenth	0xXX	一帧命令的长度，从 Lenth（不包含）开始到 EOF（包含）结束为止的总长度
CMD	0xXX	命令字节，如寻卡命令字为 0x18
Status	0xXX	状态字节，主要在响应帧中出现，表示请求命令是否执行成功
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
EOF	0x55	帧结束标志，固定为 0x55

其中 CMD 字段定义如表 2.2 所示。

表 2.2 CMD 字段的定义

位	Bit7	Bit6~Bit0
描述	0：数据包中没有 CRC-16 1：数据包中带有 CRC-16	见表 1.1

可以通过 CMD 的最高位设置该命令帧是否使用 CRC-16 进行验证，如果请求命令带有 CRC-16 验证功能，则响应帧中也带有 CRC-16 的验证功能（本协议中不使用 CRC 校验）。其中 Status 字段也有对 CRC 验证成功与否的标志，Status 字段的定义如表 2.3 所示。

表 2.3 Status 字段的定义

位	Bit7	Bit6	Bit5	Bit4	Bit3~Bit0
描述	1：命令执行失败 0：命令执行成功	1：CRC 验证失败 0：CRC 验证成功	保留	保留	0000：其他错误 0011：存储空间溢出 0100：存储空间被锁定 1011：电力不足 1111：不明错误

为了避免数据中出现 SOF、EOF 字节，实际通信过程中利用插入字节保证 SOF 和 EOF 的唯一性。当发送数据包的 SOF 和 EOF 之间出现 0xAA、0x55、0xFF 字节时，发送方应在该字节前插入一个 0XFF 字节。

接收方接收到包含插入字节的数据后应删除插入字节并提取有效数据，无论是发送方还是接受放插入字节不计入帧长度，示例如表 2.4 所示。

表 2.4 插入字节示例

需要发送的数据包（hex）	AA 04 55 00 01 55
实际发送的数据包（hex）	AA 04 FF 55 00 01 55
需要发送的数据包（hex）	AA 05 00 00 01 AA 55
实际发送的数据包（hex）	AA 05 00 00 01 FF AA 55
需要发送的数据包（hex）	AA 06 00 00 01 AA FF 55
实际发送的数据包（hex）	AA 06 00 00 01 FF AA FF FF 55

2.1 UHFCMD_GET_STATUS 命令

UHFCMD_GET_STATUS 命令字是 0x00，用来连接的时候查询 900MHz 模块的状态，主要在串口连接 900MHz 模块的时候使用。当 900MHz 模块通过串口接收到 UHFCMD_GET_STATUS 命令，请求帧结构如表 2.5 所示。

表 2.5 UHFCMD_GET_STATUS 命令请求帧结构

SOF	**Lenth**	**CMD**	**FCS**
AA	02	00	55

900MHz 模块接收到请求命令并获取状态后会将状态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 2.6 所示。

表 2.6 UHFCMD_GET_STATUS 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**VData**	**FCS**
AA	03	00	XX	XX	55

UHFCMD_GET_STATUS 命令执行成功才有响应，否则无响应。UHFCMD_GET_STATUS 命令示例如表 2.7 所示。

表 2.7 UHFCMD_GET_STATUS 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 02 00 55	请求连接模块
Recv	AA 03 00 00 55	连接成

2.2 UHFCMD_GET_POWER 命令

UHFCMD_GET_POWER 命令字是 0x01，用来读取 900MHz 模块的当前工作功率。当 900MHz 模块通过串口接收到 UHFCMD_GET_POWER 命令帧结构如表 2.8 所示。

表 2.8 UHFCMD_GET_POWER 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	01	55

900MHz 模块接收到请求命令并读取当前功率后会将功率和执行的状态结果反馈给请求方（如 PC），响应帧结构如表 2.9 所示。

表 2.9 UHFCMD_GET_POWER 命令响应帧结构

SOF	Lenth	CMD	Status	VData	FCS
AA	04	01	XX	XX	55

UHFCMD_GET_POWER 响应命令帧中 Status 用于表示成功与否，VData 字节为模块当前的功率，实际功率为 VData 字节减 0x80。UHFCMD_GET_POWER 命令示例如表 2.10 所示。

表 2.10 UHFCMD_GET_POWER 命令示例

操作	交互帧(hex)	说明
Send	AA 02 01 55	请求读取功率
Recv	AA 04 01 00 8A 55	成功，当前功率=0x8A – 0x80 = 0x0A = 10 dBm

2.3 UHFCMD_SET_POWER 命令

UHFCMD_SET_POWER 命令字是 0x02，用来设置 900MHz 模块的当前工作功率（如果没有设置则使用默认功率工作，修改功率可能修改工作频率范围并影响标签识别距离）。当 900MHz 模块通过串口接收到 UHFCMD_SET_POWER 命令帧结构如表 2.11 所示。

表 2.11 UHFCMD_SET_POWER 命令请求帧结构

SOF	Lenth	CMD	VData		FCS
AA	04	02	OPTION	POWER	55

在请求帧 VData 中为 2 个字节，第一个字节为选择字段（OPTION），第二个字节为设置的实际功率值（例如设置为 11dBm，则为 0B）。OPTIONN 字段的定义如表 2.12 所示。

表 2.12　OPTION 字段的定义

位	Bit7~Bit1	Bit0
描述	保留	1：POWER 的 BIT6~0 有效

900MHz 模块接收到请求命令并设置模块功率后会将功率设置的状态结果反馈给请求方（如 PC），响应帧结构如表 2.13 所示。

表 2.13　UHFCMD_SET_POWER 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**FCS**
AA	03	02	XX	55

UHFCMD_GET_POWER 命令示例如表 2.14 所示。

表 2.14　UHFCMD_SET_POWER 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 04 02 01 0B 55	设置当前模块功率为 11dBm
Recv	AA 03 02 00 55	设置成功

2.4 UHFCMD_GET_FRE 命令

UHFCMD_GET_FRE 命令字是 0x05，用来获取 900MHz 模块的当前工作频率。当 900MHz 模块通过串口接收到 UHFCMD_GET_FRE 命令帧结构如表 2.15 所示。

表 2.15　UHFCMD_GET_FRE 命令请求帧结构

SOF	**Lenth**	**CMD**	**FCS**
AA	02	05	55

900MHz 模块接收到请求命令并读取模块频率后会将当前工作频率信息以及获取成功的状态标志反馈给请求方（如 PC），响应帧结构如表 2.16 所示。

表 2.16　UHFCMD_GET_FRE 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**VData**						**FCS**
AA	0A	05	00	FREMODE	FREBASE	BF	CN	SPC	FREHOP	55
				1bit	1bit	2bit	1bit	1bit	1bit	

在 UHFCMD_GET_FRE 命令的响应帧中，VData 包含六部分信息，FREMODE 为标准（读取频率的时候可能该字段不能正真区分），FREBASE 为频率基数，BF 为起始频率，CN 为频道数，SPC 为频道带宽，FREHOP 为跳频书序方式，具体关系和意义可参照 UHFCMD_SET_FRE 部分，这里不详述。

UHFCMD_GET_FRE 命令示例如表 2.17 所示。

表 2.17　UHFCMD_GET_FRE 命令示例

操作	交互帧(hex)	说明
Send	AA 02 05 55	请求读取频率
Recv	AA 0A 05 00 00 01 73 01 0A 04 00 55	成功，工作频率为 920.125MHz

2.5 UHFCMD_SET_FRE 命令

UHFCMD_SET_FRE 命令字是 0x06，用来设置 900MHz 模块的当前工作频率。当 900MHz 模块通过串口接收到 UHFCMD_SET_FRE 命令帧结构如表 2.18 所示。

表 2.18　UHFCMD_SET_FRE 命令请求帧结构

SOF	Lenth	CMD	VData						FCS
AA	0A	06	FREMODE	FREBASE	BF	CN	SPC	FREHOP	55
			1bit	1bit	2bit	1bit	1bit	1bit	

请求帧的 VData 部分和读取频率的响应帧类似，VData 内包含该模块与频率相关的所有信息。频率设置有六个参数：频率工作模式（FREMODE）、频率基数（FREBASE）、起始频率（BF）、频道数（CN）、频道带宽（SPC）和跳频顺序方式（FREHOP）。其中频道数是模块在跳频时支持的最大频道个数，频道带宽是每一频道的信道带宽。

900MHz 模块支持的频率范围为 840MHZ~960MHZ，可以依据应用环境需求，自己定义频率范围。目前允许使用四种频率设置模式：

- “中国标准”模式

该模式为中国的标准，频率范围为 840MHz~845MHz 或者 920MHz~925MHz。该模式下功率和频率的对应关系如表 2.19 所示。

表 2.19　中国标准下的功率、频率对应表

序号	频率范围（MHz）	功率（dBm）	功率值（W）
1	920.5~924.5	33	2
2	840.5~844.5	33	2
3	920.0~920.5,924.5~925	20	0.1
4	840.0~840.5，844.5~845	20	0.1

注：频率范围内各个频点之差为频率基数（例如频率基数是 125K，则 840.5~844.5 之间的频点为 840.625，840.75，840.875。。。）

- “ETSI 标准”模式

该模式采用欧洲标准，有效频率范围为 865MHz~868MHz，该模式下功率和频率的对应关系如表 2.20 所示。

表 2.20　ETSI 标准下的功率、频率对应表

序号	频率范围（MHz）	功率（dBm）	功率值（W）
1	865.0~865.6	20	0.1
2	867. 6~868. 0	27	0.5
3	865.6~867.6,	33	2

注：频率范围内各个频点之差为频率基数（例如频率基数是 50K，则 865~865.6 之间的频点为 865.05，865.1。。。）

- “定频 915MHz”模式

该模式用于设置 900M 模块工作频率固定在 915MHz，定频工作。

- “用户自定义”模式

用户通过设置六个参数进行设置所要的频率工作范围：频率工作模式（FREMODE）、频率基数（FREBASE）、起始频率（BF）、频道数（CN）、频道带宽（SPC）和跳频顺序方式（FREHOP）。各个参数存在如下关系：

(1)、起始频率（BF）=【起始频率（整数部分）】＋【频率基数】×【起 始频率尾数积数】

如：起始频率＝840MHZ ＋125KHZ ×5 ＝840.625MHZ

(2)、频道带宽（SPC）＝【频道带宽积数】×【频率基数】

如：频道带宽（SPC）＝2 ×125KHZ ＝250KHZ

(3)、最终频率＝起始频率（BF）＋（频道数（CN）－1）×频道带宽（SPC）

如：最终频率＝840.625MHZ ＋（16－1）×250KHZ ＝844.375MHZ

(4)、带宽＝最终频率－起始频率（BF）

如：带宽＝844.375MHZ －840.625MHZ ＝3.75MHZ

注：【频率基数】×【频道带宽积数】不能超过 1000KHZ；当【频率基数】为 50KHZ 时，【带宽】不能大于 12MHZ，当【频率基数】为 125KHZ 时，【带宽】不能大于 32MHZ。

请求帧的 VData 部分各个字段定义如表 2.21 到表 2.25 所示，FREMODE 字段的定义如表 2.21 所示。

表 2.21 FREMODE 字段的定义

位	Bit7~Bit4	Bit3~Bit0
功能	保留	0000：中国标准（920-925MHz） 0001：中国标准（840-845MHz） 0010：ETSI 标准 0011：定频模式（915MHz） 0100：用户自定义 其他：中国标准（920-925MHz）

FREBASE 字段的定义如表 2.22 所示。

表 2.22 FREBASE 字段的定义

位	Bit7~Bit1	Bit0
描述	保留	频率基数
功能		0：50 KHz 1：125 KHz

BF 字段的定义如表 2.23 所示。

表 2.23 BF 字段的定义

位	Bit15	Bit14~Bit5	Bit4~Bit0
功能	保留	起始频率（整数部分）	起始频率尾数部分积数

CN、SPC 字段定义如表 2.24 所示。

表 2.24 CN、SPC 字段的定义

字段	CN	SPC	
位	Bit7 ~Bit0	Bit7 ~Bit4	Bit3~Bit0
功能	频道数	保留	频道带宽积数

FREHOP 字段的定义如表 2.25 所示。

表 2.25 FREHOP 字段的定义

位	Bit7 ~Bit2	Bit7 ~Bit4	Bit1~Bit0
功能	频道数	保留	00：随机跳频 01：从高往低顺序跳频 10：从低往高顺序跳频 其他：随机跳频

900MHz 模块接收到请求命令并设置模块频率后会将执行状态标志反馈给请求方（如 PC），响应帧结构如表 2.26 所示。

表 2.26　UHFCMD_SET_FRE 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**FCS**
AA	03	06	00	55

UHFCMD_GET_FRE 命令示例如表 2.27 所示。

表 2.27　UHFCMD_ SET_FRE 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 09 06 00 01 73 01 0A 04 00 55	请求设置频率为 920.125MHz，频道数 10
Recv	AA 03 06 00 55	设置成功

表 2.27 所示为设置频率实例，在设置频率的请求帧中，VData 为 00 01 73 01 0A 04 00，根据表 2.21 到表 2.25 可以得到：FREMODE= 0x00，中国标准，920MHz~925MHz；FREBASE=0x01，频率基数为 125K；BF=0x7301，转化为二进制：0111 0011 0000 0001，根据表 2.23，红色部分为整数部分，重新写一下为:11 1001 1000，也就是 0x398=920，尾数的积数为 0x01，也就是 1，而频率基数为 125K，因此这里设置的工作频率为 920.125MHz；频道数 CN=0x0A=10；频道带宽 SPC =0x04=4；跳频方式为随机跳。

2.6 UHFCMD_GET_VERSION 命令

UHFCMD_GET_VERSION 命令字是 0x07，用来获取 900M 模块的版本号和序列号。当 900MHz 模块通过串口接收到 UHFCMD_GET_VERSION 命令帧结构如表 2.28 所示。

表 2.28　UHFCMD_GET_VERSION 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	07	55

900MHz 模块接收到请求命令并获取模块版本信息后会将版本信息反馈给请求方（如 PC），响应帧结构如表 2.29 所示。

表 2.29　UHFCMD_GET_VERSION 命令响应帧结构

SOF	Lenth	CMD	Status	VData		FCS
AA	0A	07	XX	Serial（6 字节）	Version（1 字节）	55

900M 模块响应帧 VData 分为两个部分，前面表示序列号，后面表示版本号。UHFCMD_GET_VERSION 命令示例如表 2.30 所示。

表 2.30　UHFCMD_GET_VERSION 命令示例

操作	交互帧(hex)	说明
Send	AA 02 07 55	请求获取版本号信息
Recv	AA 0A 07 00 00 00 00 00 00 00 58 55	序列号为 000000000000，版本号为 5.8

2.7 UHFCMD_INVENTORY 命令

UHFCMD_INVENTORY 命令字是 0x10，用于单步循环识别标签 ID，对单张卡识别一般用该命令。当 900MHz 模块通过串口接收到 UHFCMD_INVENTORY 命令帧结构如表 2.31 所示。

表 2.31　UHFCMD_INVENTORY 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	10	55

900MHz 模块接收到请求命令后进行循环识别标签，该命令需与 UHFCMD_STOP_GET 命令结合使用，用于停止循环识别标签。模块识别到标签后将标签的 ID 获取并反馈给请求方（如 PC），响应帧结构如表 2.32 所示。

表 2.32　UHFCMD_INVENTORY 命令响应帧结构

SOF	Lenth	CMD	Status	VData	FCS
AA	XX	10	XX	UID	55

900M 模块响应帧 VData 分为两个部分，PCBits 和 UII。PCBits 为协议控制字节，后面紧接着是 UII，PCBits 的值决定了 UII 的长度，UHFCMD_INVENTORY 命令示例如表 2.33 所示。

表 2.33　UHFCMD_INVENTORY 命令示例

操作	交互帧(hex)	说明
Send	AA 02 10 55	请求单步循环识别标签
Recv	AA 05 10 00 04 00 55	成功，卡号为 00 04

表 2.33 所示为 UHFCMD_INVENTORY 的一次交互示例，当不使用 UHFCMD_STOP_GET 命令，将一直上报 Recv 帧。UHFCMD_INVENTORY 和后面的 UHFCMD_INVENTORY_SINGLE 命令类似，UHFCMD_INVENTORY_SINGLE 命令每次只识别一次标签。响应帧的 VData 部分为 UII（卡号），卡号由 PCBits + UII 构成，在对指定 UII 操作的时候也需要携带 PCBits，因此在识别卡号处理卡号的时候默认卡号包含 PCBits 和 UII。PCBits 详细格式如表 2.34 所示。

表 2.34　PCBits 数据格式的定义

位	Bits0~Bit4	Bit5~Bit6	Bit7~Bit15
描述	以 word（两个字节）为单位的 PC 和 UII 的总体长度	未定义	NSI(未使用)

其中 UII 从低位开始传输，前五位表示 PCBits 和 UII 的总长度，卡号长度为协议控制位的值乘以二，PCBits 和卡号的长度关系如表 2.35 所示。

表 2.35　PCBits 和卡号长度对应关系

PCBit0~PCBit4	PC+UII 长度（字节）
00000	2
00001	4
00010	6
….	….

2.8 UHFCMD_INVENTORY_ANTI 命令

UHFCMD_INVENTORY_ANT 命令字是 0x11，用于防碰撞识别标签 ID，对多张卡识别采用该命令。当 900MHz 模块通过串口接收到 UHFCMD_INVENTORY_ANT 命令帧结构如表 2.36 所示。

表 2.36　UHFCMD_INVENTORY_ANT 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	11	55

900MHz 模块接收到请求命令后进行循环识别标签，该命令需与 UHFCMD_STOP_GET 命令结合使用，用于停止循环识别标签。模块识别到标签后将标签的 ID 获取并反馈给请求方（如 PC），响应帧结构如表 2.37 所示。

表 2.37　UHFCMD_INVENTORY_ANT 命令响应帧结构

SOF	Lenth	CMD	Status	VData	FCS
AA	XX	11	XX	UID	55

900M 模块响应帧的 Length 随卡号的长度变化而变化，VData 为卡号，格式和 UHFCMD_INVENTORY 命令部分一致。

UHFCMD_INVENTORY_ANT 命令示例如表 2.38 所示。

表 2.38　UHFCMD_INVENTORY_ANT 命令示例

操作	交互帧(hex)	说明
Send	AA 02 11 55	请求防碰撞识别标签
Recv	AA 05 11 00 04 00 55	成功，卡号为 00 04

示例命令帧中各个字段的含义和 UHFCMD_INVENTORY 命令的示例含义一致。

2.9 UHFCMD_STOP_GET 命令

UHFCMD_STOP_GET 命令字是 0x12，用于停止 900M 模块当前所进行的任何操作，终结当前任何操作后模块处于空闲状态。当 900MHz 模块通过串口接收到 UHFCMD_STOP_GET 命令帧结构如表 2.39 所示。

表 2.39　UHFCMD_STOP_GET 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	12	55

900MHz 模块接收到请求命令后停止当前的任何操作，然后将处理结果反馈给请求方（如 PC），响应帧结构如表 2.40 所示。

表 2.40　UHFCMD_STOP_GET 命令响应帧结构

SOF	Lenth	CMD	Status	FCS
AA	03	12	XX	55

UHFCMD_STOP_GET 命令示例如表 2.41 所示。

表 2.41　UHFCMD_STOP_GET 命令示例

操作	交互帧(hex)	说明
Send	AA 02 12 55	请求停止当前所有操作
Recv	AA 03 12 00 55	停止操作成功

2.10 UHFCMD_READ_DATA（指定 UII）命令

UHFCMD_READ_DATA 命令字是 0x13，必须由用户指定特定的 UII 信息才能读取该标签的内部数据。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_READ_DATA 命令帧结构如表 2.42 所示。

表 2.42　UHFCMD_READ_DATA 命令请求帧结构

SOF	Lenth	CMD	VData					FCS
AA	xx	13	APWD	BANK	ADDR	CNT	UII	55
			4bit	1bit	1bit	1bit	n bit	

注：APWD 是读写标签等操作的存取密码；BANK 是您选择的区编号，如 UII 为 01；ADDR 是该区的具体起始地址；CNT 是读取等操作的长度（字）；UII 为指定的标签 ID。

900MHz 模块接收到请求命令后处理请求并将结果反馈给请求方（如 PC），响应帧结构如表 2.43 所示。

表 2.43　UHFCMD_READ_DATA 命令响应帧结构

SOF	Lenth	CMD	Status	VData	FCS
AA	XX	13	XX	UII	55

UHFCMD_READ_DATA 命令示例如表 2.44 所示。

表 2.44　UHFCMD_READ_DATA 命令示例

操作	交互帧(hex)	说明
Send	AA 0D 13 00 00 00 00 01 01 01 0C 00 12 34 55	请求读取 UII 的地址 1 处一个字的数据
Recv	AA 05 13 00 0C 00 55	读取成功，数据 0C 00

2.11 UHFCMD_WRITE_DATA（指定 UII）命令

UHFCMD_WRITE_DATA 命令字是 0x14，必须由用户指定特定的 UII 信息才能将数据写入该标签。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_WRITE_DATA 命令帧结构如表 2.45 所示。

表 2.45　UHFCMD_WRITE_DATA 命令请求帧结构

SOF	Lenth	CMD	VData						FCS
AA	xx	14	APWD	BANK	ADDR	CNT	DATA	UII	55
			4bit	1bit	1bit	1bit	2bit	n bit	

注：APWD 是读写标签等操作的存取密码；BANK 是您选择的区编号，如 UII 为 01；ADDR 是该区的具体起始地址；CNT 是读取等操作的长度（字）；DATA 为需写入的数据，目前支持写一个字；UII 为指定的标签 ID。

900MHz 模块接收到请求命令后验证密码并写入数据到标签，但如果写入的 UII 区，很可能修改 PCBits，如果修改了 PCBits 则后续操作如果需要指定 UII，则需要重新识别标签获取 UII 才能操作。UHFCMD_WRITE_DATA 命令与读取标签内容命令类似，处理完求命令后将结果反馈给请求方（如 PC），响应帧结构如表 2.46 所示。

表 2.46　UHFCMD_WRITE_DATA 命令响应帧结构

SOF	Lenth	CMD	Status	FCS
AA	XX	14	XX	55

UHFCMD_WRITE_DATA 命令示例如表 2.47 所示。

表 2.47　UHFCMD_WRITE_DATA 命令示例

操作	交互帧(hex)	说明
Send	AA 0F 14 00 00 00 00 01 01 01 0B 00 0C 00 12 34 55	请求在 UII 的地址 1 处写入 0x0B00
Recv	AA 03 14 00 55	写入成功

2.12 UHFCMD_ERASE_DATA（指定 UII）命令

UHFCMD_ERASE_DATA 命令字是 0x15，必须由用户指定特定的 UII 信息才能将该标签指定的数据段擦除。该命令只对支持 BlockErase 命令的标签有效。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_ ERASE _DATA 命令帧结构如表 2.48 所示。

表 2.48　UHFCMD_ERASE_DATA 命令请求帧结构

SOF	**Lenth**	**CMD**	**VData**					**FCS**
AA	xx	15	APWD	BANK	ADDR	CNT	UII	55
			4bit	1bit	1bit	1bit	n bit	

注：APWD 是擦除标签等操作的存取密码；BANK 是您选择的区编号，如 UII 为 01；ADDR 是该区的具体起始地址；CNT 是读取等操作的长度（字）； UII 为指定的标签 ID。

900MHz 模块接收到擦除命令后验证密码并擦除数据，处理完求命令后将结果反馈给请求方（如 PC），响应帧结构如表 2.49 所示。

表 2.49　UHFCMD_ERASE _DATA 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**FCS**
AA	03	15	XX	55

UHFCMD_ERASE_DATA 命令示例如表 2.50 所示。

表 2.50　UHFCMD_ERASE_DATA 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 0B 15 00 00 00 00 01 01 01 04 00 55	请求在 UII 的地址 1 处做擦除操作
Recv	AA 03 15 00 55	擦除成功

2.13 UHFCMD_LOCK_MEM 命令

UHFCMD_LOCK_MEM 命令字是 0x16，必须由用户指定特定的 UII 信息才能对该标签执行 lock 操作。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_LOCK_DATA 命令帧结构如表 2.51 所示。

表 2.51　UHFCMD_LOCK_DATA 命令请求帧结构

SOF	Lenth	CMD	VData			FCS
AA	xx	16	APWD	LOCKDATA	UII	55
			4bit	3bit	n bit	

注：APWD 是锁定标签等操作的存取密码；LOCKDATA 数据段的高四位保留，低二十位是 Lock-CommandPayload； UII 为指定的标签 ID。

LOCKDATA 的数据格式如表 2.52 所示。

表 2.52　LOCKDATA 的数据格式定义

	Kill password		Accesspassword		UIImemory		TIDmemory		Usermemory	
23~20	19	18	17	16	15	14	13	12	11	10
保留	S/W	S/W	S/W	S/W	S/W	S/W	S/W	S/W	S/W	S/W
	9	8	7	6	5	4	3	2	1	0
	PRW	PL	PRW	PL	PW	PL	PW	PL	PW	PL

注：S/W 是 Skip/Write 的缩写，PRW 为 Pwdread/write 的缩写，PL 为 Permalock 的缩写，PW 为 Pwdwrite 的缩写。位 10~19 为 Mask，位 0~9 为 Action，Action 字段决定响应的数据段在 OPEN 或 SECURED 状态下是否可读写、锁定等操作。

LOCKDATA 具体控制标签的哪个区和控制效果如表 2.52 所示，LOCKDATA 主要控制五个部分，分别是 Killpassword、Accesspassword、UIImemory、TIDmemory、Usermemory，每个区有四个控制位控制，高位是掩码，低位为具体操作。如 Usermemory，由 LOCKDATA 的第 0、1、10、11 位控制，10、11 为掩码，0、1 为允许对 Usermemory 操作的具体动作（Lock Action）。Lock Action 定义如表 2.53 所示。

表 2.53　Lock　Action 的定义（Pwd-write/Permalock）

Pwd-write	Permalock	描述
0	0	相应数据段在 OPEN 或 SECURED 状态下可写入
0	1	相应数据段在 OPEN 或 SECURED 状态下永久可写入，不可锁定
1	0	相应数据段在 SECURED 状态下可写入，OPEN 状态下不可写入
1	1	相应数据段在任何状态下不可写入

表 2.53 Lock Action 的定义（Pwd-read/write / Permalock）

Pwd-read/write	Permalock	描述
0	0	相应数据段在 OPEN 或 SECURED 状态下可读取和写入
0	1	相应数据段在 OPEN 或 SECURED 状态下永久可读取和写入，不可锁定
1	0	相应数据段在 SECURED 状态下可读写，OPEN 状态下不可读写
1	1	相应数据段在任何状态下不可读取和写入

900MHz 模块接收到请求命令，处理完命令后将结果反馈给请求方（如 PC），响应帧结构如表 2.54 所示。

表 2.54 UHFCMD_LOCK_MEM 命令响应帧结构

SOF	Lenth	CMD	Status	FCS
AA	XX	16	XX	55

UHFCMD_LOCK_MEM 命令示例如表 2.55 所示。

表 2.55 UHFCMD_LOCK_MEM 命令示例

操作	交互帧(hex)	说明
Send	AA 0D 16 00 00 00 00 00 10 04 08 00 00 01 55	请求锁定标签（08 00 00 01）
Recv	AA 03 16 00 55	锁定成功

2.14 UHFCMD_KILL_TAG 命令

UHFCMD_KILL_TAG 命令字是 0x17，必须由用户指定特定的 UII 信息才能对该标签执行销毁操作。当 900MHz 模块通过串口接收到 UHFCMD_KILL_TAG 命令帧结构如表 2.56 所示。

表 2.56 UHFCMD_KILL_TAG 命令请求帧结构

SOF	Lenth	CMD	VData		FCS
AA	xx	17	KILLPWD	UII	55
			4bit	n bit	

注：KILLPWD 是销毁标签等操作的密码。

900MHz 模块接收到请求命令，处理完命令后将结果反馈给请求方（如 PC），响应帧结构如表 2.57 所示。

表 2.57　UHFCMD_KILL_TAG 命令响应帧结构

SOF	Lenth	CMD	Status	FCS
AA	XX	17	XX	55

UHFCMD_KILL_TAG 命令示例如表 2.58 所示。

表 2.58　UHFCMD_KILL_TAG 命令示例

操作	交互帧(hex)	说明
Send	AA 0A 17 00 00 00 00 08 00 00 01 55	请求销毁标签（08 00 00 01）
Recv	AA 03 17 00 55	销毁成功

2.15 UHFCMD_INVENTORY_SINGLE 命令

UHFCMD_INVENTORY_SINGLE 命令字是 0x18，用于单步识别标签 ID，对单张卡非循环识别一般用该命令。当 900MHz 模块通过串口接收到 UHFCMD_INVENTORY_SINGLE 命令帧结构如表 2.59 所示。

表 2.59　UHFCMD_INVENTORY_SINGLE 命令请求帧结构

SOF	Lenth	CMD	FCS
AA	02	18	55

模块接收到请求命令后进行单步识别标签并将标签的 ID 反馈给请求方，响应帧结构如表 2.60 所示。

表 2.60　UHFCMD_INVENTORY_SINGLE 命令响应帧结构

SOF	Lenth	CMD	Status	VData	FCS
AA	XX	18	XX	UID	55

900M 模块响应帧 VData 分为两个部分，PCBits 和 UII。PCBits 为协议控制字节，后面紧接着是 UII，PCBits 的值决定了 UII 的长度，UHFCMD_INVENTORY_SIGLE 命令示例如表 2.61 所示。

表 2.61　UHFCMD_INVENTORY_SINGLE 命令示例

操作	交互帧(hex)	说明
Send	AA 02 18 55	请求单步识别标签
Recv	AA 05 18 00 04 00 55	成功，卡号为 00 04

PCBits 等信息和前面所述 UHFCMD_INVENTORY 命令一致，可参考 UHFCMD_INVENTORY 命令。

2.16 UHFCMD_SINGLE_READ_DATA（不指定 UII）命令

UHFCMD_SINGLE_READ_DATA 命令字是 0x20，不必由用户指定特定的 UII 信息来读取该标签的内部数据。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_SINGLE_READ_DATA 命令帧结构如表 2.62 所示。

表 2.62　UHFCMD_SINGLE_READ_DATA 命令请求帧结构

SOF	**Lenth**	**CMD**	**VData**				**FCS**
AA	xx	20	APWD	BANK	ADDR	CNT	55
			4bit	1bit	1bit	1bit	

注：APWD 是读写标签等操作的存取密码；BANK 是您选择的区编号，如 UII 为 01；ADDR 是该区的具体起始地址；CNT 是读取等操作的长度（字）。

UHFCMD_SINGLE_READ_DATA 命令和前面 UHFCMD_READ_DATA 命令类似，各个字段的含义也一致，可以参照 UHFCMD_READ_DATA 命令。900MHz 模块接收到请求命令后处理请求并将结果反馈给请求方（如 PC），响应帧结构如表 2.63 所示。

表 2.63　UHFCMD_SINGLE_READ_DATA 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**VData**		**FCS**
AA	XX	20	XX	Data	UII	55

UHFCMD_SINGLE_READ_DATA 命令示例如表 2.64 所示。

表 2.64　UHFCMD_SINGLE_READ_DATA 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 09 20 00 00 00 00 03 01 01 55	请求读取 UII 的地址 1 处一个字的数据
Recv	AA 07 20 00 00 00 04 00 55	读取成功，数据 00 00 ，卡号 04 00

在响应帧中和 UHFCMD_READ_DATA 命令略有不同，该命令的响应帧中包含了 UII 信息。可以理解为没有指定 UII 则需要知道读取的是哪张卡的数据，因此附带了卡号信息，而指定了 UII 进行读取，也就是提前知道该数据是哪张卡的数据，因此没必要在响应帧中携带是哪张标签的信息。

2.17 UHFCMD_SINGLE_WRITE_DATA（不指定 UII）命令

UHFCMD_SINGLE_WRITE_DATA 命令字是 0x21，不必由用户指定特定的 UII 信息，直接将数据写入到标签。读取标签、写入标签、擦除标签、锁定标签操作的命令帧中需含有存取密码，当存取密码不全为 0 时，模块利用 ACCESS 命令确保标签处在 SECURED 状态后进行相应操作。当 900MHz 模块通过串口接收到 UHFCMD_SINGLE_WRITE_DATA 命令帧结构如表 2.65 所示。

表 2.65　UHFCMD_SINGLE_WRITE_DATA 命令请求帧结构

SOF	**Lenth**	**CMD**	**VData**					**FCS**
AA	xx	21	APWD	BANK	ADDR	CNT	DATA	55
			4bit	1bit	1bit	1bit	Data	

注：APWD 是读写标签等操作的存取密码；BANK 是您选择的区编号，如 UII 为 01；ADDR 是该区的具体起始地址；CNT 是读取等操作的长度（字），目前只支持 1 个字；DATA 为要写入的数据（长度为 2 字节）。

UHFCMD_SINGLE_WRITE_DATA 命令和前面 UHFCMD_WRITE_DATA 命令类似，各个字段的含义也一致，可以参照 UHFCMD_WRITE_DATA 命令。900MHz 模块接收到请求命令后处理请求并将结果反馈给请求方（如 PC），响应帧结构如表 2.66 所示。

表 2.66　UHFCMD_SINGLE_ WRITE _DATA 命令响应帧结构

SOF	**Lenth**	**CMD**	**Status**	**VData**	**FCS**
AA	XX	21	XX	UII	55

UHFCMD_SINGLE_WRITE_DATA 命令和 UHFCMD_SINGLE_READ_DATA 命令类似，反馈信息中包含 UII 信息，可以参照 UHFCMD_SINGLE_READ_DATA 命令，示例如表 2.67 所示。

表 2.67　UHFCMD_SINGLE_WRITE_DATA 命令示例

操作	**交互帧(hex)**	**说明**
Send	AA 0B 21 00 00 00 00 01 01 01 10 00 55	请求请求写入数据（10 00）
Recv	AA 05 21 00 04 00 55	写入成功，卡号 04 00