

## **JIURAY UHF模块通信接口定义**

# 目录

目录.....	- 1 -
文件修改控制.....	- 4 -
版权声明.....	- 5 -
1. 简介.....	- 6 -
2. JIURAY 串口通信说明.....	- 7 -
2.1 通信数据包格式.....	- 8 -
2.1.1 数据包格式.....	- 8 -
2.1.2 数据格式.....	- 8 -
2.1.3 插入字节.....	- 9 -
2.2 上位机连续发送命令的时间间隔.....	- 10 -
2.3 JIURAY 命令列表.....	- 10 -
2.4 API 和 DEMO.....	- 13 -
3. 寄存器定义.....	- 14 -
3.1 寄存器摘要.....	- 14 -
3.2 寄存器描述.....	- 16 -
3.2.1 0x000: R_STATUS.....	- 16 -
3.2.2 0x001: R_CMD.....	- 16 -
3.2.3 0x100: R_PWR.....	- 16 -
3.2.4 0x101: R_BAUD.....	- 16 -
3.2.5 0x102: R_FRE.....	- 17 -
3.2.6 0x105: R_BLF.....	- 17 -
3.2.7 0x110: R_FREMODE.....	- 17 -
3.2.8 0x111: R_FREBASE.....	- 18 -
3.2.9 0x112-0x113: R_BF.....	- 18 -
3.2.10 0x114: R_BF_D.....	- 18 -
3.2.11 0x115: R_CN.....	- 19 -
3.2.12 0x116: R_SPC.....	- 19 -
3.2.13 0x117: R_FREHOP.....	- 19 -
3.2.14 0x120: R_BEEP.....	- 19 -
3.2.15 0x121-0x122: R_TIMER.....	- 20 -
3.2.17 0x128-0x129: R_EPC_TIMER_OUT.....	- 20 -
3.2.18 0x12A: R_BLOCK_WRITE.....	- 20 -
3.2.19 0x1F0-0x1F5: R_SERIAL.....	- 21 -
3.2.20 0x1F6-0x1F8: R_VERSION.....	- 21 -
4. 通信命令及读卡器模块响应.....	- 22 -
4.1 通用命令集.....	- 22 -
4.1.1 询问状态.....	- 22 -
4.1.2 读取功率设置.....	- 24 -
4.1.3 设置功率.....	- 25 -
4.1.4 读取频率设置.....	- 26 -

4.1.5 设置频率.....	- 27 -
4.1.6 读取寄存器.....	- 30 -

4.1.7 设置寄存器命令.....	31 -
4.1.8 恢复寄存器默认值.....	32 -
4.1.9 保存当前设置.....	33 -
4.1.10 读取 JIURAY 信息.....	34 -
4.1.11 读取 JIURAY 的 UID 信息.....	35 -
4.1.12 进入 Sleep 模式.....	36 -
4.1.13 停止操作.....	37 -
4.2 标签盘点命令集.....	38 -
4.2.1. 单标签识别_循环模式.....	38 -
4.2.2. 多标签识别_循环模式.....	39 -
4.2.3. 单标签识别_单步模式.....	40 -
4.3 标签数据存取命令集.....	41 -
4.3.1. 读取标签数据(指定 UII).....	41 -
4.3.2. 读取标签数据 ( 不指定 UII ) .....	43 -
4.3.3. 写入标签数据_单字长模式(指定 UII).....	45 -
4.3.4. 写入标签数据_单字长模式 ( 不指定 UII ) .....	46 -
4.3.5 写入标签数据_多字长模式 ( 指定 UII ) .....	47 -
4.3.6 写入标签数据_多字长模式 ( 不指定 UII ) .....	49 -
4.3.7. 擦除标签数据(指定 UII).....	51 -
4.3.8 擦除标签数据 ( 不指定 UII ) .....	52 -
4.3.9. 锁定标签(指定 UII).....	54 -
4.3.10 锁定标签 ( 不指定 UII ) .....	55 -
4.3.11. 销毁标签(指定 UII).....	56 -
4.3.12 销毁标签(不指定 UII).....	57 -
4.3.13 多标签识别并读取标签数据_循环模式.....	58 -
4.4 Select 操作命令集.....	60 -
4.4.1 添加 SELECT 记录.....	60 -
4.4.2 删除 SELECT 记录.....	62 -
4.4.3 读取 SELECT 记录.....	63 -
4.4.4 选择 SELECT 记录.....	65 -
5. 定制命令集.....	66 -
5.1 Alien Higgs-3 标签的定制命令.....	66 -
5.1.1 LoadImage(指定 UII).....	66 -
5.1.2 LoadImage(不指定 UII).....	68 -
5.1.3 FastLoadImage(指定 UII).....	69 -
5.1.4 FastLoadImage(不指定 UII).....	70 -
5.1.5 BlockReadLock(指定 UII).....	71 -
5.1.6 BlockReadLock(不指定 UII).....	73 -
5.1.7BlockPermaLock(指定 UII).....	75 -
5.1.8BlockPermaLock(不指定 UII).....	77 -
5.2 NXP 标签的定制命令.....	79 -
5.2.1 修改 Configuration Word ( 指定 UII ) .....	79 -
5.2.2 修改 Configuration Word ( 不指定 UII ) .....	81 -
5.2.3 设置读保护功能 ( 指定 UII ) .....	82 -
5.2.4 设置读保护功能 ( 不指定 UII ) .....	83 -

5.2.5 清除读保护功能（指定 UII） .....	- 84 -
-----------------------------	--------

5.2.6 清除读保护功能 ( 不指定 UII ) .....	- 85 -
5.2.7 配置 EAS 标志 ( 指定 UII ) .....	- 86 -
5.2.8 配置 EAS 标志 ( 不指定 UII ) .....	- 87 -
5.2.9 EAS 报警.....	- 88 -
附录 A : 命令摘要.....	- 89 -
附录 B : Extensible bit Vectors (EBV).....	- 91 -
附录 C : 插入字节实例.....	- 92 -
附录 D : UII 格式.....	- 93 -
附录 E : Error codes.....	- 94 -
附录 F : Lock-Command Payload.....	- 95 -
附录 G : BootLoader 固件在线升级命令说明.....	- 96 -
1、 命令 : JIURAY_START_UPDATE.....	- 96 -
-	
2、 命令 : JIURAY_SEND_INVERSE.....	- 97 -
-	
3、 命令 : JIURAY_START_TRANS.....	- 98 -
4、 命令 : JIURAY_TRAN_PACKAGE.....	- 99 -
-	
5、 命令 : JIURAY_END_UPDATE.....	- 100 -
6、 升级流程示例 : .....	- 101 -
附录 H : 参考文档.....	- 102 -

# 文件修改控制

序号	版本	*变化状态	对应的《JIURAY通信接口定义》版本	修改内容	修改日期
1	1.0	C			2012-04-05

\*变化状态：C——创建，M——修改，D——删除

## 版权声明

本文档的所有内容，包括文字、图片均为原创。对未经许可擅自使用在商业用途者，本公司保留追究其法律责任的权利。

未经授权，使用者不得擅自添加、修改、删除本文档的内容，不得以网络、光盘方式进行传播。如若违反，后果自负。





## 2.RS232 串口通信说明

玖锐读超高频读写器模块支持串行通信（UART）接口。上位机（例如：PC 或单片机）需要按照规定数据格式往模块发送命令并接收模块返回的信息。模块的 UART 接口支持多种波特率，出厂默认值为 57600，用户可通过配置相关寄存器（R\_BAUD）改变 JIURAY 的 UART 通信波特率。

表 2-1 JIURAY 支持的 UART 参数

RS232	Bit	JR2010	JR2020	JR2030
波特率	9600	√	√	√
	19200	√	√	√
	57600	√	√	√
	115200	√	√	√
数据位	8	√	√	√
奇偶校验	无			
停止位	1			

备注：划“√”表示当前产品支持该功能，划“-”表示不支持。

## 2.1 通信数据包格式

**2.1.1 数据包格式** 上位机发送到模块的数据包以下称“命令”，而模块返回到上位机的数据包以下称“响应”。以下所有数

据段的长度单位为字节。模块与上位机传递的数据包的通用格式见表2-1-1和表2-1-2。

### 2.1.2 数据格式

表 2-1-1 命令的数据包格式

数据段	SOF	LEN	CMD	PAYLOAD	*CRC16	EOF
长度	1	EBV	1	-	2	1

表 2-1-2 响应的数据包格式

数据段	SOF	LEN	CMD	STATUS	PAYLOAD	*CRC16	EOF
长度	1	EBV	1	1	-	2	1

备注：有\*为可选部分。

#### （1） SOF (Start Of Frame)

SOF 是一个字节的常数（SOF == 0xAA），表示数据帧的开始。

#### （2） LENGTH

LENGTH 部分是按字节计算的<SOF>和<EOF>之间数据（即<LENGTH>、<CMD>、<STATUS>、<PAYLOAD>、<CRC16>）的长度。其中，LEN数据段为EBV格式，详情请参考附录B。

#### （3） CMD

表 2-1-3 CMD 数据段定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
描述	CRC16控制位		模块命令					
功能	0 = 数据包中没有CRC16 1 = 数据包中带有CRC16		见命令表					

用户可利用CMD 字节的Bit 7 选择是否使用数据包的CRC16 验证功能。模块返回的响应的CRC16 设置与上位机的相应命令保持一致。

#### （4） STATUS

STATUS 是模块的响应中包含的对上位机命令的执行状态。上位机的命令中没有STATUS 部分。STATUS 中高四位是通用的标志位，而低四位是各命令中特有的状态。STATUS 通用标志位的定义见表2-1-4、低四位的定义详见各命令的状态定义表。

表 2-1-4 STATUS 的通用标志位定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit3	Bit 2	Bit 1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	见各命令的状态定义表			

备注：不带CRC16验证的情况下Bit6默认为0。

#### (5) PAYLOAD

**PAYLOAD** 是需要传递的实际数据。除了在各命令格式中已定义的**PAYLOAD** 有效字节外，在**PAYLOAD**支持的最大长度为512Bytes。

#### (6) CRC16

CRC16 部分是对<LENGTH>、<CMD>、<STATUS>(响应中)和<PAYLOAD>部分计算的CRC16 值。

用户可通过CMD 的Bit 7 选择是否使用该选项。

当上位机命令的 CRC16 验证失败时JIURAY 返回固定格式的响应，其格式如表2-1-5，其中STATUS 字节的值为0xC0。

表 2-1-5 CRC16 验证失败响应

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

#### (7) EOF (End Of Frame)

EOF 是一个字节的常数 ( EOF == 0x55 )，表示数据帧的结束。

### 2.1.3 插入字节

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。插入字节不计入 LENGTH。插入字节的实例参考附录C。

## 2.2 上位机连续发送命令的时间间隔

模块在接收到有效命令后,会在 timer\_out 时间内响应命令,时间单位为 ms,模块的 timer\_out 为 800ms (固件升级模式时为 8000ms)。当上位机连续向模块发送命令时,需注意以下规则:

(1) 若上位机没有接收到模块的命令响应,则应在上一条命令发送完毕后至少 800ms 时间间隔后,发送下一条命令。

(2) 若上位机接收到了模块的命令响应,则应在接收到该命令响应后至少 3ms 后发送下一条命令。

## 2.3 JIURAY 超高频模块命令列表

表 2-3 JIURAY 命令列表

命令	值 ( hex )	功能	命令模式	JIURAY	JIURAY	JIURAY
RLM_GET_STATUS	00	询问状态	单步模式	√	√	√
RLM_GET_POWER	01	读取功率设置	单步模式	√	√	√
RLM_SET_POWER	02	设置功率	单步模式	√	√	√
RLM_GET_FRE	05	读取频率设置	单步模式	√	√	√
RLM_SET_FRE	06	设置频率	单步模式	√	√	√
RLM_GET_VERSION	07	读取 JIURAY 信息	单步模式	√	√	√
RLM_READ_UID	0A	读取 JIURAY 的 UID 信息	单步模式	-	√	√
	0B					
-	0C-0F	Reserved				
RLM_INVENTORY	10	单标签识别_循环模式	循环模式	√	√	√
RLM_INVENTORY_ANTI	11	多标签识别_循环模式	循环模式	√	√	√
RLM_STOP_GET	12	停止操作	单步模式	√	√	√
RLM_READ_DATA	13	读取标签数据(指定 UII)	单步模式	√	√	√
RLM_WRITE_DATA	14	写入标签数据_单字长模式 (指定 UII)	单步模式	√	√	√
RLM_ERASE_DATA	15	擦除标签数据(指定 UII)	单步模式	√	√	√
RLM_LOCK_MEM	16	锁定标签(指定 UII)	单步模式	√	√	√
RLM_KILL_TAG	17	销毁标签(指定 UII)	单步模式	√	√	√
RLM_INVENTORY_SINGLE	18	单标签识别_单步模式	单步模式	√	√	√
RLM_BLOCK_WRITE_DATA	19	写入标签数据_多字长模式 (指定 UII)	单步模式	√	√	√
-	1A-1F	Reserved				
RLM_SINGLE_READ_DATA	20	读取标签数据(不指定 UII)	单步模式	√	√	√
RLM_SINGLE_WRITE_DATA	21	写入标签数据_单字长模式 (不指定 UII)	单步模式	√	√	√
RLM_SINGLE_ERASE_DATA	22	擦除标签数据(不指定 UII)	单步模式	√	√	√

RLM_SINGLE_LOCK_MEM	23	锁定标签 ( 不指定 UII )	单步模式	√	√	√
RLM_SINGLE_KILL_TAG	24	销毁标签 ( 不指定 UII )	单步模式	√	√	√
RLM_SINGLE_BLOCK_WRITE_DATA	25	写入标签数据_多字长模式 (不指定 UII)	单步模式	√	√	√
RLM_ANTI_COLLISION_RAED_DADA	26	多标签识别并读取标签数据_循环模式	循环模式	-	√	√
-	27-2F	Reserved				
RLM_READ_REG	30	读取寄存器	单步模式	-	√	√
RLM_WRITE_REG	31	设置寄存器	单步模式	-	√	√
RLM_RESET_REG	32	恢复寄存器出厂默认值	单步模式	-	√	√
RLM_SAVE_REG	33	保存当前设置	单步模式	-	√	√
-	34-37	Reserved				
RLM_ADD_SELECT	38	添加 SELECT 记录	单步模式	-	√	√
RLM_DELETE_SELECT	39	删除 SELECT 记录	单步模式	-	√	√
RLM_GET_SELECT	3A	读取 SELECT 记录	单步模式	-	√	√
RLM_CHOOSE_SELECT	3B	选择 SELECT 记录	单步模式	-	√	√
-	3C-4F	Reserved				
RLM_ENTER_SLEEP	50	进入 Sleep 模式	单步模式	√	√	√
	51-52	Reserved				
RLM_START_UPDATE	53	开始升级操作	单步模式	-	√	√
RLM_SEND_INVERSE	54	回应下位机 RN	单步模式	-	√	√
	55	Reserved				
RLM_START_TRANS	56	开始文件传输,发送文件总长度	单步模式	-	√	√
RLM_TRAN_PACKAGE	57	发送数据包	单步模式	-	√	√
RLM_END_UPDATE	58	结束升级操作	单步模式	-	√	√
	59-5A	Reserved				
RLM_ALIEN_LOADIMAGE	0x5B	对 Alien Higgs-3 IC 标签的所有存储区写操作,( 指定 UII )	单步模式	-	√	√
RLM_ALIEN_SINGLE_LOADIMAGE	0x5C	对 Alien Higgs-3 IC 标签的所有存储区写操作( 不指定 UII )	单步模式	-	√	√
RLM_ALIEN_FASTLOADIMAGE	0x5D	对 Alien Higgs-3 IC 标签的 96-bit EPC 写操作 ( 指定 UII )	单步模式	-	√	√
RLM_ALIEN_SINGLE_FASTLOADIMAGE	0x5E	对 Alien Higgs-3 IC 标签的 96-bit EPC 写操作 ( 不指定 UII )	单步模式	-	√	√
RLM_ALIEN_BLOCK_READLOCK	0x60	对 Alien Higgs-3 IC 标签的 bank3 数据块读取锁定 ( 指定 UII )	单步模式	-	√	√
RLM_ALIEN_SINGLE_BLOCK_READLOCK	0x61	对 Alien Higgs-3 IC 标签的 bank3 数据块读取锁定 ( 不指定 UII )	单步模式	-	√	√

RLM_ALIEN_BLCOKPERMA LOCK	0x62	对 Alien Higgs-3 IC 标签的 bank3 数据永久锁定 ( 指定 UII )	单步模式	-	√	√
RLM_ALIEN_SINGLE_BLCO KPERMALOCK	0x63	对 Alien Higgs-3 IC 标签的 bank3 数据永久锁定 ( 不指 定 UII )	单步模式	-	√	√
RLM_CUSTOM_NXP_CHANG ECONFIG	64	修 改 NXP 标 签 Configuration Word	单步模式	-	√	√
RLM_CUSTOM_NXP_SINGLE _CHANGECONFIG	65	修 改 NXP 标 签 Configuration Word(不指定 UII)	单步模式	-	√	√
RLM_CUSTOM_NXP_READP RETECT	66	设置 NXP 标签读保护功能	单步模式	-	√	√
RLM_CUSTOM_NXP_SINGLE _READPRETECT	67	设置 NXP 标签读保护功能 (不指定 UII)	单步模式	-	√	√
RLM_CUSTOM_NXP_RESET_ READPRETECT	68	清除 NXP 标签读保护功能	单步模式	-	√	√
RLM_CUSTOM_NXP_SINGLE _RESET_READPRETECT	69	清除 NXP 标签读保护功能 (不指定 UII)	单步模式	-	√	√
RLM_CUSTOM_NXP_CHANG EEAS	6A	配置 NXP 标签的 EAS 标志	单步模式	-	√	√
RLM_CUSTOM_NXP_SINGLE _CHANGEEAS	6B	配置 NXP 标签的 EAS 标志 (不指定 UII)	单步模式	-	√	√
RLM_CUSTOM_NXP_EASAL ARM	6C	NXP 标签的 EAS 报警	单步模式	-	√	√
-	59-7F	Reserved				

备注：

- 1、操作命令时，请注意命令的响应等待时间；
- 2、模块首次上电，系统稳定时间最小为 150ms；
- 3、关于命令模式的详细说明，见《JIURAY 快速入门手册》4.1 小节；
- 4、划“√”表示当前产品支持该功能，划“-”表示不支持。

---

## 2.4 API 和 DEMO

JIURAY 支持如第 2 节所示的 UART 串口通信,同时,为了方便用户在 PC 和手持机上的应用开发,JIURAY 提供可在 WINDOWS 桌面和移动平台下运行的 JIURAY 的 API 函数库和 DEMO(暂时只提供 WINDOWS 下的) 程序。API 函数库用 C++语言编写并封装成 WINDOWS 的标准动态链接库 ( DLL ), DEMO 程序在.NET 平台下用 C#语言开发。

相关 API 和 DEMO 文档说明, 请参考“JIURAY\_API 说明文档.pdf”和“JIURAYDemo V1.0 使用手册.pdf”。



### 3. 寄存器定义

模块内部用 512 个 8 位寄存器存储相关参数。用户可通过相关命令对寄存器进行读、写、恢复默认值操作。寄存器表中 0x000-0x0FF 地址中的数据下电不保存，而 0x100-0x1FF 地址中的数据可以设置下电保存，对于该地址区间的寄存器，设置完成之后需要调用“保存当前设置”命令来保存相应的设置，否则下电数据丢失。

注意：0x100-0x1FF 地址区间的寄存器直接存到 FLASH 中，由于 FLASH 使用寿命有限，低于 10K 次。所以，用户不要频繁操作寄存器设置，包括“功率设置”、“频率设置”等命令操作。

#### 3.1 寄存器摘要

表 3-1-1 JIURAY 寄存器

地址	名称	默认值	操作	用途
0x000	R_STATUS	0x00	只读	模块的工作状态。
0x001	R_CMD	0x00	只读	命令状态。
0x002-0x0FF	Reserved	-	-	-
0x100	R_PWR	0x1B	读写	模块的输出功率值 (dBm)。设置 R_PWR 寄存器值应该在其有效范围之内 (10~30) dBm，不同的功率下模块会自动调整相关的频率设置与其对应
0x101	R_BAUD	0x02	读写	UART 波特率。
0x102	R_FRE	0x04	读写	模块的频率 (简单模式)。
0x103-0x104	Reserved			
0x105	R_BLF	0x19	读写	数据链路属性。暂不支持更改该寄存器值
0x106-0x10F	Reserved	-	-	-
0x110	R_FREMODE	0x01	读写	选择频率设置模式。
0x111	R_FREBASE	0x01	读写	频率设置基数。
0x112	R_BF_H	0x03	读写	起始频率整数部分，MHz 为单位。
0x113	R_BF_L	0x98	读写	
0x114	R_BF_D	0x00	读写	起始频率小数部分，R_FREBASE 的倍数。
0x115	R_CN	0x0B	读写	信道数。
0x116	R_SPC	0x04	读写	信道带宽，R_FREBASE 的倍数。
0x117	R_FREHOP	0x00	读写	跳频方式。
0x118-0x11F	Reserved	-	-	-
0x120	R_BEEP	0x01	读写	控制 BUZ 引脚开启或关闭
0x121	R_TIMER_H	0x00	读写	“单标签识别_循环模式”、“多标签识别_循环模式”的间隔时间，默认是 7ms，最大为 65535ms，设置时间精度为 1ms。
0x122	R_TIMER_L	0x07	读写	
0x123-0x127	Reserved	-	-	-
0x128	R_EPC_TIMER_OU T_H	0x02	读写	单步标签通信命令的时间间隔 T，可以通过寄存器

0x129	R_EPC_TIMER_OUT_L	0x58	读写	R_EPC_TIMER_OUT 进行设置，模块 出厂缺省值为 500ms。
0x12A	R_BLOCK_WRITE	0x00	读写	标签支持的 BLOCK_WRITE 的最大字长，当标签不支持 BLOCK_WRITE 功能，应将此寄存器清 0。详见 “写入标签数据多字长模式” 部分相关说明
0x12B-0x1EF	Reserved	-	-	-
0x1F0	R_SERIAL_5	-	只读	模块 硬件批次号。
0x1F1	R_SERIAL_4	-	只读	
0x1F2	R_SERIAL_3	-	只读	
0x1F3	R_SERIAL_2	-	只读	
0x1F4	R_SERIAL_1	-	只读	
0x1F5	R_SERIAL_0	-	只读	
0x1F6	R_VERSION_2	-	只读	模块 固件版本号。
0x1F7	R_VERSION_1	-	只读	
0x1F8	R_VERSION_0	-	只读	
0x1F9-0x1FF	Reserved	-	-	-

## 3.2 寄存器描述

### 3.2.1 0x000: R\_STATUS

位	7	6	5	4	3	2	1	0
寄存器	-	-	-	-	-	-	-	CS
初始值	0	0	0	0	0	0	0	0

◆ Bit 0 – CS：Command Status

0：模块空闲。

1：命令执行中。

### 3.2.2 0x001: R\_CMD

位	7	6	5	4	3	2	1	0
寄存器	-	CMD						
初始值	0	0	0	0	0	0	0	0

◆ Bit 6~0 – CMD：当前记录的命令状态。

### 3.2.3 0x100: R\_PWR

位	7	6	5	4	3	2	1	0
寄存器	PWR							
初始值	0	0	0	1	1	0	1	1

◆ PWR：模块的输出功率（整数，单位 dBm）。

有效值：0x0A~0x1E（10~30）默认设置：

0x1E（30）

备注：产品不同，默认输出功率也不一样 JR2010 为 20dBm, JR2020 为 27dBm, JR2030 为 30dBm。

### 3.2.4 0x101: R\_BAUD

位	7	6	5	4	3	2	1	0
寄存器	BAUD							
初始值	0	0	0	0	0	0	1	0

◆ BAUD：UART1 通信波特率。

0x00：9600

0x01：19200

0x02：57600

0x03 : 115200

Other : 57600

默认设置 : 0x02

### 3.2.5 0x102: R\_FRE

位	7	6	5	4	3	2	1	0
寄存器	FRE							
初始值	0	0	0	0	0	0	0	0

◆ FRE : 简单模式下的 JIURAY 频率设置。

0x00 : 中国标准 920-925MHz。

0x01 : 中国标准 840-845MHz。

0x02 : ETSI 标准。

0x03 : 922MHz , 定频模式。

Other : 预留。默认设置 :

0x00

### 3.2.6 0x105: R\_BLF

位	7	6	5	4	3	2	1	0
寄存器	BLF							
初始值	0	0	0	1	0	1	0	1

◆ BLF : 数据链路设置。

默认设置 : 0x19 , 数据速率 : 213 kHz , 编码方式 MILLER8。

【注】此寄存器值暂不可更改。

### 3.2.7 0x110: R\_FREMODE

位	7	6	5	4	3	2	1	0
寄存器	-	-	-	-	-	-	-	FREMODE
初始值	0	0	0	0	0	0	0	1

◆ Bit 0 – FREMODE : 频率设置模式。

0 : 简单模式。简单模式下 R\_FRE 寄存器有效 , R\_FREBASE , R\_BF\_H , R\_BF\_L , R\_BF\_D , R\_CN , R\_SPC , R\_FREHOP 寄存器无效。

1 : 高级模式。高级模式下 R\_FREBASE , R\_BF\_H , R\_BF\_L , R\_BF\_D , R\_CN , R\_SPC , R\_FREHOP

寄存器有效，R\_FRE 寄存器无效。

默认设置：0x01

### 3.2.8 0x111: R\_FREBASE

位	7	6	5	4	3	2	1	0
寄存器	-	-	-	-	-	-	-	FREBASE
初始值	0	0	0	0	0	0	0	1

◆ Bit 0 – FREBASE：频率设置基数。R\_BF\_D 和 R\_SPC 由 FREBASE 的倍数表示。

0：50kHz。

1：125kHz。默

认设置：0x01

### 3.2.9 0x112-0x113: R\_BF

	R_BF_H								R_BF_L							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
寄存器	BF															
初始值	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	0

◆ BF：高级模式下起始频率的整数部分，MHz 为单位。

取值范围：840~960MHz

默认设置：920MHz

### 3.2.10 0x114: R\_BF\_D

位	7	6	5	4	3	2	1	0
寄存器	BF_D							
初始值	0	0	0	0	0	0	0	0

◆ BF\_D：高级模式下起始频率的小数部分，由 R\_FREBASE 的倍数表示，单位为 KHz。起始频率的小数部分与 BF\_D 的关系表达式为：起始频率小数部分 = BF\_D × 50KHz ( 或 125KHz )。50KHz、125KHz 的选取由 R\_FREBASE 寄存器参数决定。注意：BF\_D × 50KHz ( 或 125KHz ) < 1000KHz。设置结果与 R\_FREBASE 乘积大于 1000KHz 时，BF\_D 值将设置为 0。

◆ 默认设置：0x00

### 3.2.11 0x115: R\_CN

位	7	6	5	4	3	2	1	0
寄存器	CN							
初始值	0	0	0	0	1	0	1	1

◆ CN：高级模式下的信道数。

备注：当最终频率大于 960MHz 时，即  $CN \times R\_SPC \times R\_FREBASE + BF$  大于 960MHz 时，CN 的值将设置为 1；

当 FREBASE = 0 时，如果  $CN \times R\_SPC \times R\_FREBASE$  大于 12MHz 时，CN 值将取 16。

当 FREBASE = 1 时，如果  $CN \times R\_SPC \times R\_FREBASE$  大于 32MHz 时，CN 值将取 16。

默认设置：0x0B

### 3.2.12 0x116: R\_SPC

位	7	6	5	4	3	2	1	0
寄存器	SPC							
初始值	0	0	0	0	0	1	0	0

◆ SPC：高级模式下的信道带宽，由 R\_FREBASE 的倍数表示。注： $R\_SPC \times R\_FREBASE$  值不能大于 1000KHz，当大于 1000KHz 时，该值将自动设置为 4。

默认设置：0x04

### 3.2.13 0x117: R\_FREHOP

位	7	6	5	4	3	2	1	0
寄存器	-	-	-	-	-	-	FREHOP	
初始值	0	0	0	0	0	0	0	0

◆ Bit 1-0 – FREHOP：高级模式下的跳频方式。

0b00：随机跳频。

0b01：从低到高顺序跳频。

0b10：从高到低顺序跳频。

Other：预留。默认设置：

0x00

### 3.2.14 0x120: R\_BEEP

位	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---

寄存器	-	-	-	-	-	-	-	BEEP
初始值	0	0	0	0	0	0	0	1

◆ Bit 0 – BEEP：Command

0：禁止 BUZ 引脚。

1：使能 BUZ 引脚。

默认设置：0x01

### 3.2.15 0x121-0x122: R\_TIMER

	R_TIMER_H								R_TIMER_L							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
寄存器	TIMER															
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

◆ Bit 15-0 – TIMER：单标签循环识别命令间隔时间

默认设置：0x07

### 3.2.17 0x128-0x129：R\_EPC\_TIMER\_OUT

	R_EPC_TIMER_OUT_H								R_EPC_TIMER_OUT_L							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
寄存器	TIMER															
初始值	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0

◆ Bit 15-0 – TIMER：DATA\_EXCHANGE\_TIMER 操作时间溢出控制，时间单位以 1ms 为间隔。最小值为 500ms。

默认设置：0x0258

### 3.2.18 0x12A：R\_BLOCK\_WRITE

位	7	6	5	4	3	2	2	0
寄存器	Block_Write							
初值	0	0	0	0	0	0	0	0

◆ Block\_Write：标签支持的 BLOCK\_WRITE 的数据长度。

多字节写入操作时，配合该寄存器使用，可提高多字节写入效率。更详细信息，参考本文档“多字节写入标签数据（指定/不指定 UII）”部分的相关内容 默认设置：0x00

**3.2.19 0x1F0-0x1F5: R\_SERIAL**

	R_SERIAL_5	R_SERIAL_4	R_SERIAL_3	R_SERIAL_2	R_SERIAL_1	R_SERIAL_0
寄存器	SERIAL					
初始值	-	-	-	-	-	-

◆ SERIAL : JIURAY 序列号。



4. 通信命令及读卡器模块响应

此部分内容，详细说明上位机向读卡器模块发送的命令，以及该命令发送完毕后，读卡器模块的响应信息。

4.1 通用命令集

此部分内容，详细说明上位机向读卡器发送的通用命令，通用命令指的是，上位机设置或读取读卡器模块的相关设置，如功率、频率设置等。

4.1.1 询问状态

4.1.1.1 功能简介

该命令询问模块 的状态，用户可利用该命令查询 模块 是否连接，如果有响应则说明模块已经连接，如果在指定时 间内没有响应则说明模块没有连接成功。

在命令格式设置正确，硬件通信接口设置正确，上位机波特率设置正确的前提下，该命令总是能得到正确的响应。若该命令发出后

①□ 模块无响应，则用户仔细检查硬件通信接口是否正确连接，上位机波特率是否设置正确。当确认硬件接口正确连接 且波特率设置正确的前提下，则仔细检查命令格式是否符合表 4-1-1-1 要求。

② 模块 返回失败 则 模块 收到了数据包 ,但是数据包的 CRC16 校验出错。则用户可以仔细检查是否采用的是带 CRC16 校验的命令包，但命令包中的 CRC16 值不正确。

4.1.1.2 数据格式

表 4-1-1-1 询问状态命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

表 4-1-1-2 询问状态响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

4.1.1.3 命令状态定义

表 4-1-1-3 询问状态 STATUS

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
功能	通用位	保留	0 = 连接成功

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

4.1.1.4 相关 API 函数

表 4-1-1-4 询问状态相关 API 函数

函数名	说明
UhfReaderDisconnect();	打开串口并与模块 建立链接

#### 4.1.1.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 00 55	成功：aa 03 00 00 55
	失败：无返回

## 4.1.2 读取功率设置

### 4.1.2.1 功能简介

该命令读取模块的功率设置。用户使用模块对标签进行操作前可用该命令读取 模块的功率设置。该命令有两种响 应格式，即操作成功（表 4-1-2-2）和失败（表 4-1-2-3）。

### 4.1.2.2 数据格式

表 4-1-2-1 读取功率设置命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

表 4-1-2-2 读取功率设置响应格式（成功）

数据段	SOF	LEN	CMD	STATUS	POWER	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 4-1-2-3 读取功率设置响应格式（失败）

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-2-4 POWER 数据段格式

POWER	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
描述	保留	输出功率（dbm）						

### 4.1.2.3 命令状态定义

该命令只支持通用状态位。

### 4.1.2.4 相关 API 函数

表 4-1-2-5 读取功率设置相关 API 函数

函数名	说明
UhfGetPower();	读取模块 的功率设置

### 4.1.2.5 命令示例

发送命令格式（hex）	返回数据格式（hex）
aa 02 01 55	成功：aa 04 01 00 1a 55
	失败：无返回

### 4.1.3 设置功率

#### 4.1.3.1 功能简介

该命令设置 模块 的输出功率。用户使用 模块 对标签进行操作前需要用该命令设置 模块 的输出功率。若用户没有设置 模块 的功率，模块 工作时将使用默认设置。另外，可以通过写 R\_PWR 寄存器来设置功率，注意，模块 能够自动保存功率设置，下电不丢失数据，所以，一次设置成功之后下次上电时无须重复设置。

备注：在 模块 的频率设置模式为简单模式的条件下，改变输出功率设置将有可能改变工作频率范围。

#### 4.1.3.2 数据格式

表 4-1-3-1 设置功率命令格式

数据段	SOF	LEN	CMD	OPTION	POWER	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 4-1-3-2 设置功率响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-3-3 OPTION 数据段格式

OPTION	Bit 7 ~ 0
描述	必须置为 0x01

备注：POWER 数据段的定义见表 4-1-2-4。

#### 4.1.3.3 命令状态定义 该命令只

支持通用状态位。

#### 4.1.3.4 相关 API 函数

表 4-1-3-4 设置功率相关 API 函数

函数名	说明
UhfSetPower ();	设置模块 的功率

#### 4.1.3.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 04 02 01 1a 55	成功：aa 03 02 00 55
	失败：无返回

## 4.1.4 读取频率设置

### 4.1.4.1 功能简介

该命令读取模块的频率设置。频率设置相关数据段的说明见 4.1.5 节。

### 4.1.4.2 数据格式

表 4-1-4-1 读取频率设置命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

表 4-1-4-2 读取频率设置响应格式

数据段	SOF	LEN	CMD	STATUS	FRE MODE	FRE BASE	BF	CN	SPC	FREHO P	*CRC1 6	EOF
长度	1	EBV	1	1	1	1	2	1	1	1	2	1

### 4.1.4.3 命令状态定义 该命令只

支持通用状态位。

### 4.1.4.4 相关 API 函数

表 4-1-4-3 读取频率设置相关 API 函数

函数名	说明
UhfGetFrequency ();	读取模块 频率设置

### 4.1.4.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 05 55	成功 : aa 0a 05 00 00 01 73 05 10 02 00 55
	失败 : 无返回

## 4.1.5 设置频率

### 4.1.5.1 功能简介

该命令设置模块的频率。模块的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）、信道数（CN）、信道带宽（SPC）和跳频顺序方式（FREHOP）。其中信道数是模块在跳频时支持的最大信道个数，信道带宽是每一信道的信道带宽。

模块支持的基准频率范围为 840MHz~960MHz，用户可以依据应用环境需求，自己定义频率范围。目前模块允许使用四种频率设置模式：

(1) “中国标准”模式

(2) “ETSI 标准”模式，该模式采用欧洲标准，有效频率范围为 865-868MHz。  
频率设置应用笔记.pdf 文档。

(3) “定频模式”，该模式将频率设置为 922MHz，并且定频工作。

(4) “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）、信道数（CN）、信道带宽（SPC）和跳频顺序方式（FREHOP）。注意：当用户选择“中国标准”、“ETSI 标准”、“定频模式”时，FREMODE 字段有效，其他字段无效，忽略用户所设置的参数值。

依据表 4-1-5-2、表 4-1-5-3、表 4-1-5-4、表 4-1-5-5、表 4-1-5-6，起始频率（BF）、信道数（CN）、信道带宽（SPC）、最终频率和带宽存在以下关系：

(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) 信道带宽（SPC）= 【信道带宽积数】 × 【频率基数】（注 1）

如：信道带宽（SPC）= 2 × 125KHz = 250KHz

(3) 最终频率 = 起始频率（BF）+ （信道数（CN）- 1）× 信道带宽（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）（注 2）

如：带宽 = 844.375MHz - 840.625MHz = 3.75MHz

【注 1】【信道带宽】不能超过 1000KHz；

【注 2】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。

#### 4.1.5.2 数据格式

表 4-1-5-1 设置频率命令格式

数据段	SOF	LEN	CMD	FRE MODE	FRE BASE	BF	CN	SPC	FREHOP	*CRC16	EOF
长度	1	EBV	1	1	1	2	1	1	1	2	1

表 4-1-5-2 FREMODE 字段定义

位	Bit7~Bit4	Bit3~Bit0
功能	保留	频率工作模式
		0000：中国标准（920-925MHz） 0001：中国标准（840-845MHz） 0010：ETSI 标准 0011：定频模式（922MHz） 0100：用户自定义 其他：预留

表 4-1-5-3 FREBASE 字段定义

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

表 4-1-5-4 BF 字段定义

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

表 4-1-5-5 CN 字段定义

位	Bit7 ~ Bit0
功能	信道数

备注：CN 字段不能为 0。

表 4-1-5-6 SPC 字段定义

位	Bit7 ~ Bit4	Bit3~Bit0
功能	保留	信道带宽积数

表 4-1-5-7 FREHOP 字段定义

位	Bit7 ~ Bit2	Bit1~Bit0
功能	保留	跳频顺序方式
		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

表 4-1-5-8 设置频率响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

#### 4.1.5.3 命令状态定义 该命令只

支持通用状态位。

#### 4.1.5.4 相关 API 函数

表 4-1-5-3 设置频率相关 API 函数

函数名	说明
UhfSetFrequency();	设置 模块的频率

#### 4.1.5.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 09 06 00 01 73 05 10 02 00 55	成功 : aa 03 06 00 55
	失败 : 无返回



## 4.1.6 读取寄存器

### 4.1.6.1 功能简介

该命令读取模块的寄存器表。

### 4.1.6.2 命令帧格式

表 4-1-6-1 读取寄存器命令帧格式

数据段	SOF	LEN	CMD	RADD	RLEN	*CRC16	EOF
长度	1	EBV	1	2	2	2	1

数据段说明：

RADD：寄存器段的起始地址（0x000~0x1FF）

RLEN：读取的寄存器段的长度，以字节为单位

### 4.1.6.3 响应帧格式

表 4-1-6-2 读取寄存器成功帧响应格式

数据段	SOF	LEN	CMD	STATUS	REG	*CRC16	EOF
长度	1	EBV	1	1	RLEN	2	1

数据段说明：

REG：读出的寄存器值

RLEN：读出的寄存器字段长度，以字节为单位

表 4-1-6-3 读取寄存器响应格式（失败）

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-6-4 读取寄存器 STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：RADD + RLEN >= 512 0：其他未知错误

### 4.1.6.4 相关 API 函数

函数名	说明
UhfGetRegister();	读取寄存器命令

### 4.1.6.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 06 30 01 00 00 01 55	成功：aa 04 30 00 1a 55
	失败：aa 03 30 81 55

## 4.1.7 设置寄存器命令

### 4.1.7.1 功能简介

该命令设置模块寄存器值。

### 4.1.7.2 命令帧格式

表 4-1-7-1 设置寄存器命令格式

数据段	SOF	LEN	CMD	RADD	RLEN	REG	*CRC16	EOF
长度	1	EBV	1	2	2	RLEN	2	1

数据段说明：

RADD： 寄存器段的起始地址 ( 0x000~0x1FF )

RLEN： 欲设置的寄存器段的长度，以字节为单位

REG： 欲设置的寄存器段的值

### 4.1.7.3 响应帧格式

表 4-1-7-2 设置寄存器响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-7-3 设置寄存器 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~2	Bit 1~0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	00：其他错误 01：RADD + RLEN 超出范围 10：地址中包含只读寄存器 11：REG数据段长度错误

### 4.1.7.4 相关 API 函数

函数名	说明
UhfSetRegister();	设置寄存器命令

### 4.1.7.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 07 31 01 00 00 01 1b 55	成功： aa 03 31 00 55
	失败： aa 03 30 81 55, aa 03 31 82 55, aa 03 31 83 55

## 4.1.8 恢复寄存器默认值

### 4.1.8.1 功能简介

该命令把 模块 的所有寄存器恢复到默认值。

### 4.1.8.2 命令帧格式

表 4-1-8-1 恢复寄存器默认值命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

### 4.1.8.3 响应帧格式

表 4-1-8-2 恢复寄存器默认值响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-8-3 恢复寄存器 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	0

### 4.1.8.4 相关 API 函数

函数名	说明
UhfResetRegister();	恢复寄存器命令

### 4.1.8.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 32 55	成功 : aa 03 32 00 55
	失败 : aa 03 32 80 55

## 4.1.9 保存当前设置

### 4.1.9.1 功能简介

该命令将寄存器（地址：0x100~0x1FF）的值保存到内部 FLASH 中,下电数据不丢失。该命令用在“设置寄存器”命令之后。

### 4.1.9.2 命令帧格式

表 4-1-9-1 保存当前设置命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

### 4.1.9.3 响应帧格式

表 4-1-9-2 保存当前设置命令响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

### 4.1.9.4 相关 API 函数

函数名	说明
UhfSaveRegister();	保存当前设置命令

### 4.1.9.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 33 55	成功： aa 03 33 00 55
	失败： aa 03 33 80 55

## 4.1.12 进入 Sleep 模式

### 4.1.12.1 功能简介

模块提供省电模式(sleep)，该模式下的功耗比正常模式低得多，但是相对于 Standby 模式稍高一点

### 4.1.14.2 数据格式

表 4-1-12-1 进入省电模式命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

表 4-1-12-2 进入省电模式响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

### 4.1.12.3 命令状态定义

表 4-1-12-3 设置寄存器 STATUS

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit0
功能	通用位	保留	0：成功 1：失败

注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

### 4.1.12.4 相关 API 函数

函数名	说明
UhfEnterSleepMode();	进入省电模式

### 4.1.12.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 50 55	成功：aa 03 50 00 55
	失败：aa 03 50 81 55

### 4.1.13 停止操作

#### 4.1.13.1 功能简介

该命令用于停止模块当前所进行的任何操作。模块接收到该命令之后退出当前操作状态，进入正常待机模式

#### 4.1.13.2 命令帧格式

表 4-1-13-1 停止操作命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

#### 4.1.13.3 响应帧格式

表 4-1-13-2 停止操作响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-1-13-3 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	0

该命令只支持通用状态位。

#### 4.1.13.4 相关 API 函数

函数名	说明
UhfStopOperation();	停止操作命令

#### 4.1.13.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 12 55	成功：aa 03 12 00 55
	失败：无返回

## 4.2 标签盘点命令集

### 4.2.1. 单标签识别\_循环模式

#### 4.2.1.1 功能简介

该命令启动标签识别循环，对单张标签进行识别时使用该命令。该命令有两种响应格式：

(1) 模块接收“单标签识别\_循环模式”命令后返回该命令确认响应（表 4-2-1-2），通知上位机“单标签识别\_循环模式”启动成功，然后模块进入步骤 2；

(2) 模块不断盘点通讯场内的标签，并按照（表 4-2-1-3）帧格式实时返回标签的 UII 信息直到接收停止操作命令。

#### 4.2.1.2 命令帧格式

表 4-2-1-1 单标签识别\_循环模式命令格式（单标签识别）

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

#### 4.2.1.3 响应帧格式

表 4-2-1-2 单标签识别\_循环模式命令确认响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-2-1-3 单标签识别\_循环模式命令的标签 UII 信息响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

备注：本文档中的 UII 包括 PC bits，即 PC+UII。UII 的格式见附录 D。

表 4-2-1-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：该响应为“单标签识别_循环模式命令确认响应” 0：该响应为“单标签识别_循环模式命令确认响应”

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

#### 4.2.1.4 相关 API 函数

函数名	说明
UhfStartInventory();	单标签识别_循环模式命令

#### 4.2.1.5 通讯示例

发送命令格式（hex）	返回响应数据格式（hex）
aa 02 10 55	成功：
	先返回确认命令：aa 03 10 01 55（收到识别标签命令）
	再返回标签数据：aa 07 10 00 08 00 00 01 55（不断返回标签数据）
	失败：仅返回确认命令：aa 03 10 01 55（没有识别到标签）

## 4.2.2. 多标签识别\_循环模式

### 4.2.2.1 功能简介

该命令启动标签识别循环，对多张标签进行识别时使用该命令。发送命令时需指定防碰撞识别的初始 Q 值。Q 值范围 0-15，模块使用默认 Q 值为 3。该命令的响应方式与单标签识别命令一致。

### 4.2.2.2 命令帧格式

表 4-2-2-1 识别标签命令格式（防碰撞识别）

数据段	SOF	LEN	CMD	Q	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-2-2-2 Q 数据段格式

Q	Bit 7 ~ Bit 4	Bit 3 ~ Bit 0
描述	保留	Q Bit 3 ~ 0

### 4.2.2.3 响应帧格式

表 4-2-2-3 多标签识别\_循环模式命令确认响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-2-2-4 单标签识别\_循环模式命令的标签 UII 信息响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 4-2-2-5 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：该响应为“单标签识别_循环模式命令确认响应” 0：该响应为“单标签识别_循环模式命令确认响应”

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

### 4.2.2.4 相关 API 函数

函数名	说明
UhfStartInventory();	多标签识别_循环模式命令

### 4.2.2.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 03 11 03 55	成功：
	先返回确认命令：aa 03 11 01 55 ( 收到识别标签命令 )
	再返回标签数据：aa 07 11 00 08 00 00 01 55 ( 不断返回标签数据 )
	失败：仅返回确认命令：aa 03 11 01 55 ( 没有识别到标签 )



### 4.2.3. 单标签识别\_单步模式

#### 4.2.3.1 功能简介

该命令识别单张标签。与单标签识别和防碰撞识别命令不同的是：该命令不启动识别循环。每次上位机发送该命令时，模块识别标签，如果识别到标签则返回标签号，若没有识别到标签则返回失败。

#### 4.2.3.2 命令帧格式

表 4-2-3-1 单标签识别\_单步模式命令格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

#### 4.2.3.3 响应帧格式

表 4-2-3-2 单标签识别\_单步模式响应格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 4-2-3-3 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1 : 盘点标签失败 0 : 盘点标签成功

#### 4.2.3.4 相关 API 函数

函数名	说明
UhfInventorySingleTag();	单标签识别_单步模式命令

#### 4.2.3.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 02 18 55	盘点成功：aa 07 18 00 08 00 00 01 55
	盘点失败：aa 03 18 01 55

## 4.3 标签数据存取命令集

### 4.3.1. 读取标签数据(指定 UII)

#### 4.3.1.1 功能简介

该命令从指定了 UII 的目标标签中读取数据。

#### 4.3.1.2 命令帧格式

表 4-3-1-1 读取标签数据 (指定 UII) 命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	UII	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1		2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

BANK：标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR：标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 D

CNT：数据长度，以 WORD ( 2 字节 ) 为单位，支持 CNT=0。

CNT !=0 时的响应帧格式见表 4-3-1-2

CNT=0 时的响应帧格式见表 4-3-1-3

【注】CNT=0，目前模块仅支持存储区大小≤233 个字长，当存储区大小>233 字长，模块返回 STATUS=0x80 的失败响应。

#### 4.3.1.3 响应帧格式

表 4-3-1-2 读取标签数据 (指定 UII) 命令成功响应帧格式 (CNT!=0)

数据段	SOF	LEN	CMD	STATUS	DATA	*CRC16	EOF
长度	1	EBV	1	1	CNT*2	2	1

表 4-3-1-3 读取标签数据 (指定 UII) 命令成功响应帧格式 (CNT=0)

数据段	SOF	LEN	CMD	STATUS	DATA_LEN	DATA	*CRC16	EOF
长度	1	EBV	1	1	2	CNT*2	2	1

备注：DATA\_LEN 为 DATA 的数据长度。

表 4-3-1-4 读取标签数据 (指定 UII) 命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE ( Error Code ) 数据段的定义见本文的附录 E。

表 4-3-1-5 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1 : 响应中含ECODE数据段 0 : 响应中不含ECODE数据段

#### 4.3.1.4 相关 API 函数

函数名	说明
UhfReadDataByEPC(); UhfReadMaxDataByEPC();	读取标签数据 ( 指定 UII )

#### 4.3.1.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 0d 13 00 00 00 00 01 01 01 08 00 00 01 55	成功 : aa 05 13 00 08 00 55
	失败 : aa 04 13 81 04 55

### 4.3.2. 读取标签数据（不指定 UII）

#### 4.3.2.1 功能简介

该命令指示 模块 从通讯场内的 1 个标签读取数据。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则读取该标签的目标存储区数据，然后立刻将电子标签的 UII 信息+数据存储区数据上传。

#### 4.3.2.2 命令帧格式

表 4-3-2-1 读取标签数据（不指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	2	1

数据段说明：

APWD： 标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 D

CNT： 数据长度，以 WORD（2 字节）为单位，支持 CNT=0。

CNT!=0 时的响应帧格式见表 4-3-2-2

CNT=0 时的响应帧格式见表 4-3-2-3

【注】CNT=0，当存储区大小>233 字长，JIURAY 返回 STATUS=0x80 的失败响应

#### 4.3.2.3 响应帧格式

表 4-3-2-2 读取标签数据（不指定 UII）命令成功响应帧格式（CNT!=0）

数据段	SOF	LEN	CMD	STATUS	DATA	UII	*CRC16	EOF
长度	1	EBV	1	1	CNT*2		2	1

数据段说明：

UII：目标标签的 UII 信息（即该 UII 对应的标签的数据被读取）

表 4-3-2-3 读取标签数据（不指定 UII）命令成功响应帧格式（CNT=0）

数据段	SOF	LEN	CMD	STATUS	DATA_LEN	DATA	UII	*CRC16	EOF
长度	1	EBV	1	1	2	CNT*2		2	1

数据段说明：

UII：目标标签的 UII 信息（即该 UII 对应的标签的数据被读取）。

DATA\_LEN： 目标标签的相应存储区的长度，以字节为单位。

【注】CNT=0 时，读写器读出目标存储区的所有数据，目前模块支持最大读取的长度为 233 个字长，当存储区长度>233 个字，则读写器发出 CNT=0 的读取指令，则 模块 不返回任何响应

表 4-3-2-4 读取标签数据 ( 不指定 UII ) 响应格式 ( 失败 )

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE ( Error Code ) 数据段的定义见本文的附录 E。

表 4-3-1-5 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含ECODE数据段 0：响应中不含ECODE数据段

#### 4.3.2.4 相关 API 函数

函数名	说明
UhfReadDataFromSingleTag(); UhfReadMaxDataFromSingleTag();	读取标签数据 ( 不指定 UII )

#### 4.3.2.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 09 20 00 00 00 00 01 01 01 55	成功：aa 09 20 00 08 00 08 00 00 01 55 失败：aa 04 20 81 04 55

### 4.3.3. 写入标签数据\_单字长模式(指定 UII)

#### 4.3.3.1 功能简介

该命令向指定了 UII 的目标标签中写入 1 个字长 ( 2 字节 ) 的数据。

#### 4.3.3.2 命令帧格式

表 4-3-3-1 写入标签数据\_单字长模式 ( 指定 UII ) 命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	DATA	UII	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	CNT*2		2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，以WORD ( 2 字节 ) 为单位，支持CNT=0。

DATA： 欲写入的数据，长度为 2\*CNT 个字节

UII： 目标标签的 UII 数据

#### 4.3.3.3 响应帧格式

表 4-3-3-2 写入标签数据\_单字长模式 ( 指定 UII ) 响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE ( Error Code ) 数据段的定义见本文的附录 E。

表 4-3-3-3 STATUS 数据段定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含ECODE数据段 0：响应中不含ECODE数据段

#### 4.3.3.4 相关 API 函数

函数名	说明
UhfWriteDataByEPC();	写入标签数据_单字长模式 ( 指定 UII ) 命令

#### 4.3.3.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 0f 14 00 00 00 00 01 01 01 10 00 08 00 00 01 55	成功：aa 03 14 00 55
	失败：aa 04 14 81 04 55

#### 4.3.4. 写入标签数据\_单字长模式（不指定 UII）

##### 4.3.4.1 功能简介

该命令指示模块向通讯场内的 1 个标签写入 1 个字长（2 字节）数据。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则向该标签的目标存储区写入数据，然后向上位机返回响应，指示写入操作是否成功。

##### 4.3.4.2 命令帧格式

表 4-3-4-1 写入标签数据\_单字长模式（不指定 UII）据命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	DATA	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	CNT*2	2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，CNT必须为0x010x010x010x01

##### 4.3.4.3 响应帧格式

表 4-3-4-2 写入标签数据\_单字长模式（不指定 UII）响应帧格式

数据段	SOF	LEN	CMD	STATUS	*UII	*ECODE	*CRC16	EOF
长度	1	EBV	1	1		1	2	1

数据段说明：

UII： 目标标签的 UII 信息（即该 UII 对应的标签被写入了数据）

ECODE： 写入操作中遇到的错误类型，详细见本文附录 E

表 4-3-4-3 写入标签数据\_单字长模式（不指定 UII）STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含ECODE数据段 0：响应中不含ECODE数据段

#### 5.3.4 相关 API 函数

函数名	说明
UhfWriteDataToSingleTag();	写入标签数据_单字长模式（不指定 UII）命令

##### 4.3.4.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 0b 21 00 00 00 00 01 01 01 10 00 55	成功：aa 07 21 00 08 00 00 01 55
	失败：aa 04 21 81 04 55

### 4.3.5 写入标签数据\_多字长模式（指定 UII）

#### 4.3.5.1 功能简介

该命令向指定了 UII 的目标标签中写入 N 个字长（2\*N 字节）的数据。

该命令配合 R\_BLOCK\_WRITE 寄存器设置使用。根据电子标签是否支持 BLOCK\_WRITE 功能，按以下规则进行设置。

- （1） 电子标签不支持 BLOCK\_WRITE 功能

R\_BLOCK\_WRITE=0。

JIURAY 内部启用 Write 功能，将 N 个字长的数据，分 N 次循环写入目标存储区。

- （2） 电子标签支持 BLOCK\_WRITE 功能的最大字长为 M 个字长。

R\_BLOCK\_WRITE=M。

模块 内部启用 BLOCK\_WRITE 功能，将 N 个字长的数据，快速写入目标存储区。

#### 4.3.5.2 命令帧格式

表 4-3-5-1 写入标签数据\_多字长模式（指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	DATA	UII	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	CNT*2		2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，CNT不能为0000，最大支持220220220220

DATA： 欲写入的数据，长度为 2\*CNT 个字节

UII： 目标标签的 UII 数据

#### 4.3.5.3 响应帧格式

表 4-3-5-2 写入标签数据\_多字长模式（指定 UII）命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-3-5-3 多字节写入标签数据（指定 UII）响应格式（失败）

数据段	SOF	LEN	CMD	STATUS	*ECODE	WRITED-LEN	*CRC16	EOF
长度	1	EBV	1	1	1	1	2	1

备注：WRITED-LEN 表示标签未能成功写入 N 个字长的数据，但已将 WRITED-LEN（WRITED-LEN<N）个字长的数据写入标签中



表 4-3-5-4 多字节写入标签数据 ( 指定 UII ) STATUS

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~2	Bit 1~0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	00 : 响应中不含 ECODE 数据段 01 : 响应中含 ECODE 数据段 , 数据未写完成 02 : 响应中不含 ECODE 数据段 , 数据未写完成

备注：该命令的 STATUS Bit1~0 只在 Bit 7 为 1 时有效。

#### 4.3.5.4 相关 API 函数

函数名	说明
UhfBlockWriteDataByEPC();	写入标签数据_多字长模式 ( 指定 UII ) 命令

#### 4.3.5.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 17 19 00 00 00 00 03 00 05 11 22 33 44 66 77 88 99 10 77 08 00 00 01 55	成功 : aa 03 19 00 55
	失败 : aa 03 19 80 55 aa 05 19 81 0b 00 55 或 aa 04 19 82 04 55

### 4.3.6 写入标签数据\_多字长模式（不指定 UII）

#### 4.3.6.1 功能简介

该命令指示模块向通讯场内的 1 个标签写入 N 个字长（2\*N 字节）的数据。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则向该标签的目标存储区写入数据，然后向上位机返回响应，指示写入操作是否成功。

该命令配合 R\_BLOCK\_WRITE 寄存器设置使用。根据电子标签是否支持 BLOCK\_WRITE 功能，按以下规则进行设置。

（1） 电子标签不支持 BLOCK\_WRITE 功能

R\_BLOCK\_WRITE=0。

JIURAY 内部启用 Write 功能，将 N 个字长的数据，分 N 次循环写入目标存储区。

（2） 电子标签支持 BLOCK\_WRITE 功能的最大字长为 M 个字长。

R\_BLOCK\_WRITE=M。

JIURAY 内部启用 BLOCK\_WRITE 功能，将 N 个字长的数据，快速写入目标存储区。

#### 4.3.6.2 命令帧格式

表 4-3-6-1 写入标签数据\_多字长模式（不指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	DATA	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	CNT*2	2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，CNT 不能为 0000，最大支持 220220220220

DATA： 欲写入的数据，长度为 2\*CNT 个字节

#### 4.3.6.3 响应帧格式

表 4-3-6-2 写入标签数据\_多字长模式（不指定 UII）成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

数据段说明：

UII： 目标标签的 UII 信息（即该 UII 对应的标签被写入了数据

表 4-3-6-3 写入标签数据\_多字长模式（不指定 UII）失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	WRITED-LEN	UII	*CRC16	EOF
长度	1	EBV	1	1	1	1		2	1

数据段说明：

UII：目标标签的 UII 信息（即该 UII 对应的标签被写入了数据

WRITED-LEN：表示标签未能成功写入 N 个字长的数据，但已将 WRITED-LEN ( WRITED-LEN<N ) 个字长的数据写入标签中

表 4-3-5-4 写入标签数据\_多字长模式（不指定 UII）STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~2	Bit 1~0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	00：响应中不含 ECODE 数据段 01：响应中含 ECODE 数据段，数据未写完成 02：响应中不含 ECODE 数据段，数据未写完成

备注：该命令的 STATUS Bit1~0 只在 Bit 7 为 1 时有效。

#### 4.3.6.4 相关 API 函数

函数名	说明
UhfBlockWriteDataToSingleTag();	写入标签数据_多字长模式（不指定 UII）命令

#### 4.3.6.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 13 25 00 00 00 00 03 00 05 12 34 56 78 9a 10 11 12 13 14 55	成功：aa 07 25 00 08 00 00 01 55 失败：aa 03 25 80 55 aa 09 25 81 0b 00 08 00 00 01 55 aa 08 25 82 04 08 00 00 01 55

### 4.3.7. 擦除标签数据(指定 UII)

#### 4.3.7.1 功能简介

该命令将指定了 UII 的目标标签中的目标存储区内的数据擦除。用户使用该操作时，请详细阅读目标标签的相关资料，确定标签是否支持 BLOCK\_ERASE 功能。若不支持，则该操作无效。

#### 4.3.7.2 命令帧格式

表 4-3-7-1 擦除标签数据命令格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	UII	*CRC16	EOF
长度	1	1	1	4	1	EBV	1		2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 D

CNT： 数据长度，以 WORD（2 字节）为单位，CNT 不能为 0。

CNT： 标签存储区欲擦除数据段的长度，以字（2 字节）为单位。

UII： 目标标签的 UII

#### 4.3.7.3 响应帧格式

表 4-3-7-2 擦除标签数据（指定 UII）响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 4-3-7-3 擦除标签数据（指定）STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

#### 4.3.7.4 相关 API 函数

函数名	说明
UhfEraseDataByEPC();	擦除标签数据（指定 UII）命令

#### 4.3.7.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 0d 15 00 00 00 00 11 01 01 08 00 00 01 55	成功：aa 03 15 00 55
	失败：aa 04 15 81 04 55

4.3.8 擦除标签数据（不指定 UII）

4.3.8.1 功能简介

该命令指示模块 将通讯场内的 1 个标签的相应存储区的数据擦除。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则擦除该标签的目标存储区数据，然后根据操作执行情况返回成功/失败响应。

用户使用该操作时，请详细阅读目标标签的相关资料，确定标签是否支持 BLOCK\_ERASE 功能。若不支持，则该操作无效。

4.3.8.2 命令帧格式

表 4-3-8-1 擦除标签数据(不指定 UII)命令格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	2	1

数据段说明：

APWD： 标签的数据访问密码。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，以 WORD（2 字节）为单位，CNT 不能为 0。

CNT： 标签存储区欲擦除数据段的长度，以字（2 字节）为单位。

4.3.8.3 响应帧格式

表 4-3-8-2 擦除标签数据（不指定 UII）成功响应格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

UII： 目标标签的 UII 信息（即该 UII 对应的标签数据被擦除）

表 4-3-8-3 擦除标签数据(不指定 UII) 失败响应格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

ECODE： 擦除操作中遇到的错误类型，详见本文附录 E。

表 4-3-8-4 擦除标签数据 STATUS

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含ECODE数据段 0：响应中不含ECODE数据段

4.3.8.4 相关 API 函数

函数名	说明
UhfEraseDataFromSingleTag();	擦除标签数据（不指定 UII）命令

#### 4.3.8.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 09 22 12 34 56 78 03 10 02 55	成功： aa 07 22 00 08 00 00 01 55
	失败： aa 03 22 80 55

4.3.9. 锁定标签(指定 UII)

4.3.9.1 功能简介

该命令将指定了 UII 的目标标签中的目标存储区内的数据锁定/解锁。目标存储区锁定后，该存储区的数据可读不可写。

4.3.9.2 命令帧格式

表 4-3-9-1 锁定标签（指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	LOCKDATA	UII	*CRC16	EOF
长度	1	EBV	1	4	3		2	1

数据段说明：

APWD： 标签的数据访问密码。

LOCKDATA： 数据段的高四位为保留位，低二十位是 Lock-Command Payload，详见附录 F。

4.3.9.3 响应帧格式

表 4-3-9-2 锁定标签（指定 UII）响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 4-3-9-3 锁定标签（指定 UII）STATUS

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含ECODE数据段 0：响应中不含ECODE数据段

4.3.9.4 相关 API 函数

函数名	说明
UhfLockMemByEPC();	擦除标签数据（指定 UII）命令

4.3.9.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 0d 16 00 00 00 00 00 10 04 08 00 00 01 55	成功：aa 03 16 00 55
	失败：aa 04 16 81 04 55

### 4.3.10 锁定标签（不指定 UII）

#### 4.3.10.1 功能简介

该命令指示模块将通讯场内的 1 个标签的相应存储区的数据擦除。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则锁定/解锁该标签的目标存储区，然后根据操作执行情况返回成功/失败响应。目标存储区锁定后，该存储区的数据可读不可写。

#### 4.3.10.2 命令帧格式

表 4-3-10-1 锁定标签(不指定 UII)命令格式

数据段	SOF	LEN	CMD	APWD	LOCKDATA	*CRC16	EOF
长度	1	EBV	1	4	3	2	1

数据段说明：

APWD： 标签的数据访问密码。

LOCKDATA： 数据段的高四位为保留位，低二十位是 Lock-Command Payload，详见附录 F。

#### 4.3.10.3 响应帧格式

表 4-3-10-2 锁定标签（不指定 UII）成功响应格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

UII：目标标签的 UII 信息（即该 UII 对应的标签存储区被锁定）

表 4-3-10-3 锁定标签（不指定 UII）失败响应格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

ECODE： 锁定操作中遇到的错误类型，详见见本文附录 E

表 4-3-10-4 锁定标签 STATUS

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

备注：该命令的 STATUS Bit0 只在 Bit 7 为 1 时有效。

#### 4.3.10.4 相关 API 函数

函数名	说明
UhflLockMemFromSingleTag();	锁定标签数据（不指定 UII）

#### 4.3.10.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 09 23 12 34 56 78 0a 98 06 55	成功： aa 11 23 00 30 00 80 03 45 67 89 01 23 45 00 14 19 83 55
	失败： aa 03 23 80 55 aa 04 23 81 04 55



### 4.3.11. 销毁标签(指定 UII)

#### 4.3.11.1 功能简介

该命令将指定了 UII 的目标标签杀死。标签被杀死后，将不支持读写器的任何操作。当目标标签的 Kill Pass 为 0，标签不可被杀死。

#### 4.3.11.2 命令帧格式

表 4-3-11-1 销毁标签（指定 UII）命令帧格式

数据段	SOF	LEN	CMD	KILLPWD	UII	*CRC16	EOF
长度	1	EBV	1	4		2	1

数据段说明：

KILLPWD：标签的杀死密码。当 KILLPWD=0 时，标签不可被杀死。

#### 4.3.11.3 响应帧格式

表 4-3-11-2 销毁标签（指定 UII）响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 4-3-11-3 销毁标签（指定 UII）STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

#### 4.3.11.4 相关 API 函数

函数名	说明
UhfKillTagByEPC();	销毁标签（指定 UII）

#### 4.3.11.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 0a 17 00 00 00 00 08 00 00 01 55	成功：aa 03 17 00 55
	失败：aa 04 17 81 04 55

### 4.3.12 销毁标签(不指定 UII)

#### 4.3.12.1 功能简介

该命令指示模块将通讯场内的 1 个标签的相应存储区的数据擦除。该操作启动后，读写器盘点通讯场内标签，当成功盘点出 1 张标签后，则杀死该标签，然后根据操作执行情况返回成功/失败响应。

标签被杀死后，将不支持读写器的任何操作。当目标标签的 Kill Pass 为 0，标签不可被杀死。

#### 4.3.12.2 命令帧格式

表 4-3-12-1 销毁标签(不指定 UII)命令定义

数据段	SOF	LEN	CMD	KILLPWD	*CRC16	EOF
长度	1	EBV	1	4	2	1

数据段说明：

KILLPWD：标签的杀死密码。当 KILLPWD=0 时，标签不可被杀死。

#### 4.3.12.3 响应帧格式

表 4-3-12-2 销毁标签(不指定 UII) 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

UII：目标标签的 UII 信息（即该 UII 对应的标签被杀死）

表 4-3-12-3 销毁标签(不指定 UII) 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

ECODE：杀死操作中遇到的错误类型，详见见本文附录 E

表 4-3-12-4 销毁标签 STATUS

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

#### 4.3.12.4 相关 API 函数

函数名	说明
UhfKillSingleTag();	销毁标签数据（不指定 UII）

#### 4.3.12.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
aa 06 24 12 34 56 78 55	成功：aa 11 24 00 30 00 80 03 45 67 89 01 23 45 00 14 19 83 55
	失败：aa 03 24 80 55

4.3.13 多标签识别并读取标签数据\_循环模式

4.3.13.1 功能简介

为方便用户在盘点标签的同时，能够获取标签的存储数据，模块还提供了“多标签识别并读取标签数据\_循环模式”操作。该操作具有以下特点：

- (1) 启动防碰撞机制，盘点通讯场内的标签；
- (2) 盘点标签的同时，获取标签的目标存储区数据，最大支持 2 个不同数据存储区数据读取，数据存储区数据地址、数据长度可自由配置；
- (3) 循环模式命令。

用户启动“多标签识别并读取标签数据\_循环模式”操作后，模块 首先向上位机返回“多标签识别并读取标签数据\_循环模式”命令的 ACK 响应，然后进行如（1）所描述的操作

(1) 模块 启动防碰撞机制，不断盘点其通信场内的标签、读取标签目标存储区数据，并实时上传 模块 识别到的电子标签的“UII 信息+目标存储器数据信息”，直至模块判定场内的标签盘点结束，然后进入步骤（2）

(2) 模块 转入正常等待工作模式，并等待 R\_TIMER，然后进入步骤（1）

以上 2 个步骤，不断往复，直至模块接收到一个有效的“停止操作”。模块接收到有效停止操作后，进入正常等待工作模式。

4.3.13.2 命令帧格式

表 4-3-13-1 多标签识别并读取标签数据\_循环模式命令帧格式

数据段	SOF	LEN	CMD	APWD	BANK	PTR	CNT	OPTION	*PAYLOAD	*CRC16	EOF
长度	1	EBV	1	4	1	EBV	1	1		2	1

数据段说明：

APWD： 标签的数据访问密码。APWD 必须置 0。

BANK： 标签的存储分区。

Reserved 存储区，BANK=0x00

UII 存储区，BANK=0x01

TID 存储区，BANK=0x02

User 存储区，BANK=0x03

本命令不支持 Reserved 存储区读取

PTR： 标签存储区的起始地址，EBV 格式。关于 EBV 的详细说明，参见本文附录 B

CNT： 数据长度，CNT不能为0000，最大支持220220220220

DATA： 欲写入的数据，长度为 2\*CNT 个字节

UII： 目标标签的 UII 数据

OPTION = 0x00，PAYLOAD 数据段格式

数据段	Q	CMD_SECOND
长度	1	1

2、OPTION = 0x01,PAYLOAD 数据段格式

数据段	BANK1	PTR1	CNT1	Q	CMD_SECOND
长度	1	EBV	1	1	1

备注：

- 1、OPTION = 0x00，为读取 1 级数据，在防冲突识别的时候能读出其他任意一个数据块的数据内容。
- 2、OPTION = 0x01，为读取 2 级数据内容，在防冲突识别的时候能读出其他任意两个数据块的数据内容，BANK,PTR,CNT 是指定第一级数据块的参数，BANK1,PTR1,CNT1 指定第二级数据块的参数。
- 3、CMD 为防冲突读取数据命令，CMD\_SECOND 为“读取标签数据（不指定 UII 模式）”命令，代号 0x20。
- 4、CNT、CNT1 数据段是以 WORD（2 字节）为单位的读出数据的长度，CNT、CNT1 长度支持 0 的情况。
- 5、CNT 与 CNT1 之和必须小于 222，且 CNT、CNT1 均不为 0 时。

表 4-3-13-2 防碰撞识别读取标签数据响应格式（成功）

数据段	SOF	LEN	CM D	STATUS	DATA_ LEN1	DATA1	*DATA_ LEN2	*DATA2	UII	*CRC16	EOF
长度	1	EBV	1	1	2	CNT*2 ( DATA1 )	2	CNT1*2 ( DATA2 )		2	1

备注：DATA\_LEN2 和 DATA2 字段，在成功读取到第二级数据时有效。

### 4.3.13.3 命令状态定义

表 4-3-13-3 防碰撞识别读取标签数据 STATUS

位	Bit 7 ~ 4	Bit 3 ~ 2	Bit 1~0
功能	通用位	保留	00 : 成功读取到 1 级数据 01 : 成功读取到 2 级数据

注：该命令的 STATUS Bit1~0 只在 Bit 7 为 0 时有效。

备注：status 定义

- 1、0x00, 成功读取到标签数据，读取 1 级数据；
- 2、0x01，成功读取到标签数据，读取 2 级数据；

### 4.3.13.4 相关 API 函数

函数名	说明
UhfStartReadDataFromMultiTag ();	防碰撞识别读取标签数据命令

### 4.3.13.5 命令示例

一级数据读取方式：

发送命令格式（hex）	返回数据格式（hex）
aa 0c 26 00 00 00 00 02 00 00 00 03 20 55	先返回确认命令：AA 03 26 01 55 成功： aa 2b 26 00 00 18 e2 00 34 12 01 2c 10 00 09 5e 0d 58 06 12 00 7f 00 04 5f bb ff ff ff ff dc 50 34 00 e2 00 34 12 dc 03 01 19 25 14 13 36 55 失败：无数据响应

两级数据读取方式：

发送命令格式（hex）	返回数据格式（hex）
aa 0f 26 00 00 00 00 03 00 04 01 02 00 00 03 20 55	先返回确认命令：aa 03 26 01 55 成功： aa 27 26 01 00 08 e2 00 34 12 01 2c 10 00 00 0a 11 11 11 11 11 11 11 11 11 34 00 e2 00 34 12 dc 03 01 19 25 14 13 36 55 失败：无数据响应

## 4.4 Select 操作命令集

为方便用户自定义的选取目标标签群，读写器模块提供了丰富的 Select 操作，开放符合 EPC C1 GEN2/ISO18000-6C 协议的 Select 命令相关参数供用户进行标签的灵活选择。

### 4.4.1 添加 SELECT 记录

#### 4.4.1.1 功能简介

该命令添加 SELECT 记录到 模块。如果相应 SINDEX 的 SRECORD 已存在，则覆盖原来的 SRECORD。SRECORD 数字段是复合数据段，其帧格式见表 4-4-1-2，

#### 4.4.1.2 命令帧格式

表 4-4-1-1 添加 SELECT 记录命令格式

数据段	SOF	LEN	CMD	SRECORD	*CRC16	EOF
长度	1	EBV	1	-	2	1

表 4-4-1-2 SRECORD 数据段格式

数据段	SINDEX	SLEN	TARGET	ACTION	BANK	PTR	LEN	MASK	TRUNCATE
长度	1	1	1	1	1	EBV	1		1

数据段说明：

SINDEX：SRECORD 记录的索引。添加记录命令的 SINDEX 大于 0，小于等于 15

SLEN：SRECORD 记录的长度

其他字段：参考 ISO18000-6C/EPC C1GEN2 协议中 Select 命令的相应字段定义

#### 4.4.1.3 响应帧格式

表 4-4-1-3 添加 SELECT 记录响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-4-1-4 添加 SELECT 记录 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~2	Bit 1~0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	00：其他错误 01：SINDEX 超出范围 10：SINDEX=0 是只读 SRECORD 11：未定义

#### 4.4.1.4 相关 API 函数

函数名	说明
UhfAddFilter ();	添加 SELECT 记录

#### 4.4.1.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 0e 38 01 0c 04 00 01 10 20 30 00	成功： aa 03 38 00 55
80 03 00 55	失败： aa 03 33 80 55 , aa 03 33 81 55 , aa 03 33 82 55

## 4.4.2 删除 SELECT 记录

### 4.4.2.1 功能简介

该命令删除 模块的 SELECT 记录。

### 4.4.2.2 命令帧格式

表 4-4-2-1 删除 SELECT 记录命令帧格式

数据段	SOF	LEN	CMD	SINDEX	*CRC16	EOF
长度	1	EBV	1	1	2	1

### 4.4.2.3 响应帧格式

表 4-4-2-2 删除 SELECT 记录响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 4-4-2-3 删除 SELECT 记录 STATUS

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

### 4.4.2.4 相关 API 函数

函数名	说明
UhfDeleteFilterByIndex ();	删除 SELECT 记录

### 4.4.2.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 03 39 01 55	成功： aa 03 39 00 55
	失败： aa 03 39 81 55 , aa 03 39 82 55

### 4.4.3 读取 SELECT 记录

#### 4.4.3.1 功能简介

该命令读取 模块的 SELECT 记录列表信息，SINDEX 是需要读取的 SRECORD 的起始序号，SNUM 是需要读取的 SRECORD 的个数。模块接收到有效的“读取 SELECT 记录”命令后，首先返回对该命令的 ACK 响应，其帧定义见表 4-4-3-2，然后返回 SNUM 个 SRECORD 帧响应，其帧格式见表 4-4-3-3。如果 SNUM 为 0，则返回所有的 SRECORD。

#### 4.4.3.2 命令帧格式

表 4-4-3-1 读取 SELECT 记录命令格式

数据段	SOF	LEN	CMD	SINDEX	SNUM	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

#### 4.4.3.3 响应帧格式

表 4-4-3-2 读取 SELECT 记录命令响应格式（成功）

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

备注：成功：STATUS = 0x01；失败：无返回。

表 4-4-3-3 获取 SELECT 记录响应格式（成功）

数据段	SOF	LEN	CMD	STATUS	SRECORD	*CRC16	EOF
长度	1	EBV	1	1	-	2	1

表 4-4-3-4 获取 SELECT 记录响应格式（失败）

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

#### 4.4.3.3 命令状态定义

表 4-4-3-4 获取 SELECT 记录响应 STATUS 字段定义

位	Bit 7 ~ 4	Bit 3 ~ 2	Bit1 ~ 0
功能	通用位	保留	00：有后续响应 01：没有后续响应，完成 SNUM 个 SRECORD 传输 10：没有后续响应，该序号 Select 信息不存在 11：此记录信息错误

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

表 4-4-3-5 获取 SELECT 记录 STATUS（失败）

位	Bit 7 ~ 4	Bit 3 ~ 2	Bit1 ~ 0
功能	通用位	保留	00：其他错误 01：SINDEX + SNUM 超出范围

备注：该命令的 STATUS Bit0 只在 Bit 7 为 1 时有效。

#### 4.4.3.4 相关 API 函数



函数名	说明
UhfReadFilterByIndex (); UhfStartGetFilterByIndex();	读取 Select 记录

#### 4.4.3.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 04 3a 00 01 55	成功： aa 03 3a 01 55 ( 确认收到读取 Select 命令的响应 ) aa 0b 3a 01 00 08 00 04 01 00 00 00 55 ( 返回 Select 数据包 )
	失败： aa 03 3a 80 55 , aa 03 3a 81 55

## 4.4.4 选择 SELECT 记录

### 4.4.4.1 功能简介

该命令用于选取用户添加的“SELECT 记录”，被选取的记录将在识别标签、读取数据、写入数据、擦除数据、锁定、销毁操作时自动发送用户自定义的 SELECT 命令。该命令对高级识别操作无效。

### 4.4.4.2 命令帧格式

表 4-4-4-1 选择 SELECT 记录命令格式

数据段	SOF	LEN	CMD	SINDEX	SNUM	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 4-4-4-2 选择 SELECT 记录命令响应格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

### 4.4.4.3 命令状态定义

选择 SELECT 记录 STATUS (失败)

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
功能	通用位	保留	0 : SINDEX + SNUM 超出范围

备注：该命令的 STATUS Bit0 只在 Bit 7 为 1 时有效。

### 4.4.4.4 相关 API 函数

函数名	说明
UhfSelectFilterByIndex ();	选择 SELECT 记录命令

### 4.4.4.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
aa 04 3b 01 01 55	成功： aa 03 3b 00 55
	失败： aa 03 3b 80 55

## 5. 定制命令集

### 5.1 Alien Higgs-3 标签的定制命令

以下第 5.1 节的命令为 Alien Higgs-3 标签的私有命令，只适用于 Alien Higgs-3 标签，对其他厂商的标签无效。

#### 5.1.1 LoadImage(指定 UII)

##### 5.1.1.1 功能简介

使用这个命令可以对 Alien Higgs-3 IC 标签的整个存储区一次性编程，存储区包括 Reserve Bank, EPC Bank and User Bank。其中 TID 字段必须与当前标签一致。使用该命令需要指定标签的 UII 信息。

##### 5.1.1.2 命令帧格式

表 5-1-1-1 loadimage 命令帧格式

数据段	SOF	LEN	CMD	APWD	TID	ACCESS-PWD	KILL-PWD	EPC	USER	UII	*CRC16	EOF
长度	1	EBV	1	4	4	4	4	14	64		2	1

APWD：标签的数据访问密码。

TID：对于标签，该数据不可以修改，最好的方式是读出当前标签的该区域的值来写入。TID 区 offset 10 and offset 9.

ACCESS-PWD：为 Access Password Lo 和 Access Password Hi。

KILL-PWD：为 Kill Password Lo 和 Kill Password Hi。

##### 5.1.1.3 响应帧格式

表 5-1-1-2 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-1-1-3 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-1-4 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

##### 5.1.1.4 相关 API 函数

函数名	说明
UhfAlienLoadImageByEPC();	对 Alien Higgs-3 IC 标签的整个存储区一次性编程(指定 UII)

##### 5.1.1.5 命令示例

发送命令格式 (hex)	返回数据格式 (hex)
AA 6E 5B 00 00 00 00 FF FF FF	成功：AA 03 5B 00 55

FF 5F FB 00 00 00 00 00 00 00	失败 : AA 03 5B 80 55
00 30 00 00 00 00 00 00 00 00	
00 00 00 00 00 CD EF 89 AB 45	
67 01 23 CD EF 89 AB 45 67 01	
23 CD EF 89 AB 45 67 01 23 CD	
EF 89 AB 45 67 01 23 CD EF 89	
AB 45 67 01 23 CD EF 89 AB 45	
67 01 23 CD EF 89 AB 45 67 01	
23 CD EF 89 AB 45 67 01 23 34	
00 00 1D 90 61 53 19 01 32 19	
10 4D A9 55	

## 5.1.2 LoadImage(不指定 UII)

### 5.1.2.1 功能简介

使用这个命令可以对 Alien Higgs-3 IC 标签的整个存储区一次性编程，存储区包括 Reserve Bank, EPC Bank and User Bank。其中 TID 字段必须与当前标签一致。使用该命令不需要指定标签的 UII 信息。

### 5.1.2.2 命令帧格式

表 5-1-2-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	TID	ACCESS-P WD	KILL-P WD	EPC	USER	*CRC1 6	EOF
长度	1	EBV	1	4	4	4	4	14	64	2	1

APWD：标签的数据访问密码。

TID：对于标签，该数据不可以修改，最好的方式是读出当前标签的该区域的值来写入。TID 区 offset 10 and offset 9.

ACCESS-PWD：为 Access Password Lo 和 Access Password Hi.

KILL-PWD：为 Kill Password Lo 和 Kill Password Hi.

### 5.1.2.3 响应帧格式

表 5-1-2-2 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 5-1-2-3 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-2-4 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.1.2.4 相关 API 函数

函数名	说明
UhfAlienLoadImageFromSingleTag();	对 Alien Higgs-3 IC 标签的整个存储区一次性编程(不指定 UII)

### 5.1.2.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 60 5C 00 00 00 00 FF FF FF FF 5F FB 00 00 00 00 00 00 00 00 30 00 00 00 00 00 00 00 00 00 00 00 00 00 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 CD EF 89 AB 45 67 01 23 55	成功：AA 11 5C 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00 55 失败：AA 03 5C 80 55

### 5.1.3 FastLoadImage(指定 UII)

#### 5.1.3.1 功能简介

使用FastLoadImage命令写整个标签存储区，该命令不同于LoadImage命令，使用该命令时，H3标签的bank3区将自动擦除。只有Reserved Memory 和 96 bit EPC Memory 被编程。使用该命令必须指定标签的UII。

#### 5.1.3.2 命令帧格式

表 5-1-3-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	TID	ACCESS-P WD	KILL-P WD	EPC	UII	*CRC16	EOF
长度	1	EBV	1	4	4	4	4	14		2	1

APWD：标签的数据访问密码。

TID：对于标签，该数据不可以修改，最好的方式是读出当前标签的该区域的值来写入。TID 区 offset 10 and offset 9.

ACCESS-PWD：为 Access Password Lo 和Access Password Hi.

KILL-PWD：为 Kill Password Lo 和 Kill Password Hi.

#### 5.1.3.3 响应帧格式

表 5-1-3-2 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-1-3-3 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-3-4 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

#### 5.1.3.4 相关 API 函数

函数名	说明
UhfAlienFastLoadImageByEPC();	对 Alien Higgs-3 IC 标签的 bank0 和 bank1 编程，同时擦出 bank3(指定 UII)

#### 5.1.3.5 Command example

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 2E 5D 00 00 00 00 FF FF FF FF 5F FB 00 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00 55	成功：AA 03 5D 00 55 失败：AA 03 5D 80 55

## 5.1.4 FastLoadImage(不指定 UII)

### 5.1.4.1 功能简介

使用FastLoadImage命令写整个标签存储区，该命令不同于LoadImage命令，使用该命令时，H3标签的bank3将自动擦除。只有Reserved Memory 和 96 bit EPC Memory 被编程。

### 5.1.4.2 命令帧格式

表 5-1-4-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	TID	ACCESS-P WD	KILL-P WD	EPC	*CRC16	EOF
长度	1	EBV	1	4	4	4	4	14	2	1

APWD：标签的数据访问密码。

TID：对于标签，该数据不可以修改，最好的方式是读出当前标签的该区域的值来写入。TID 区 offset 10 and offset 9.

ACCESS-PWD：为 Access Password Lo 和Access Password Hi.

KILL-PWD：为 Kill Password Lo 和 Kill Password Hi.

### 5.1.4.3 响应帧格式

表 5-1-4-2 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 5-1-4-3 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-4-4 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.1.4.4 相关 API 函数

函数名	说明
UhfAlienFastLoadImageFromSingleTag();	对 Alien Higgs-3 IC 标签的 bank0 和 bank1 编程，同时擦出 bank3(不指定 UII)

### 5.1.4.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 20 5E 00 00 00 00 FF FF FF FF 5F FB 00 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00 55	成功：AA 11 5E 00 30 00 00 00 00 00 00 00 00 00 00 00 00 00 55 失败：AA 03 5E 80 55

## 5.1.5 BlockReadLock(指定 UII)

### 5.1.5.1 功能简介

在开放状态，该命令能阻止读取标签的 User 存储区 ( bank3 )，因此，bank3 区提供了一个安全的数据存储，通过该命令开启和关闭 bank3 数据块读取锁定的功能。使用该命令必须指定标签的 UII。

### 5.1.5.2 命令帧格式

表 5-1-5-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	Read-LockBits	UII	*CRC16	EOF
长度	1	EBV	1	4	1		2	1

Read-LockBits : 8 bits的读-锁数据位。

表 5-1-5-2 Read-LockBits域:

Read-LockBits的发送顺序，逻辑1对应的数据块被读锁定。

1st	2nd	3rd	4th	5th	6th	7th	8 <sup>th</sup> &Last
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7

表 5-1-5-3 Block与User区的对应图

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
User Word 0	User Word 4	User Word 8	User Word 12	User Word 16	User Word 20	User Word 24	User Word 28
User Word 1	User Word 5	User Word 9	User Word 13	User Word 17	User Word 21	User Word 25	User Word 29
User Word 2	User Word 6	User Word 10	User Word 14	User Word 18	User Word 22	User Word 26	User Word 30
User Word 3	User Word 7	User Word 11	User Word 15	User Word 19	User Word 23	User Word 27	User Word 31

因为EPC/User存储区的划分，根据下表所示，User块的数量依靠EPC长度来决定表5-1-5-4。

EPC Length 0 thru 6	EPC Length 7 thru 10	EPC Length 11 thru 14	EPC Length 15 thru 18	EPC Length 19 thru 22	EPC Length 23 thru 26	EPC Length 27 thru 30	EPC Length 31
Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0
Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	N.A.
Block 2	Block 2	Block 2	Block 2	Block 2	Block 2	N.A.	N.A.
Block 3	Block 3	Block 3	Block 3	Block 3	N.A.	N.A.	N.A.
Block 4	Block 4	Block 4	Block 4	N.A.	N.A.	N.A.	N.A.
Block 5	Block 5	Block 5	N.A.	N.A.	N.A.	N.A.	N.A.
Block 6	Block 6	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Block 7	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

N.A. = Not available

### 5.1.5.3 响应帧格式

表 5-1-5-5 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1



表 5-1-5-6 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-5-7 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1 : 响应中含 ECODE 数据段 0 : 响应中不含 ECODE 数据段

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

#### 5.1.5.4 相关 API 函数

函数名	说明
UhfAlienBlockReadLockByEPC();	Alien Higgs-3 IC 标签的 bank3 读锁定(指定 UII)

#### 5.1.5.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 15 60 00 00 00 00 80 34 00	成功 : AA 03 60 00 55
00 1D 90 61 53 19 01 32 19 10 4D A9 55	失败 : AA 03 60 80 55

## 5.1.6 BlockReadLock(不指定 UII)

### 5.1.6.1 功能简介

在开放状态，该命令能阻止读取标签的 User 存储区 ( bank3 )，因此，bank3 区提供了一个安全的数据存储，通过该命令开启和关闭 bank3 数据块读取锁定的功能。

### 5.1.6.2 命令帧格式

表 5-1-6-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	Read-LockBits	*CRC16	EOF
长度	1	EBV	1	4	1	2	1

Read-LockBits : 8 bits的读-锁数据位。

表 5-1-6-2 Read-LockBits域:

Read-LockBits的发送顺序，逻辑1对应的数据块被读锁定。

1st	2nd	3rd	4th	5th	6th	7th	8 <sup>th</sup> &Last
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7

表 5-1-6-3 Block与User区的对应图

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
User Word 0	User Word 4	User Word 8	User Word 12	User Word 16	User Word 20	User Word 24	User Word 28
User Word 1	User Word 5	User Word 9	User Word 13	User Word 17	User Word 21	User Word 25	User Word 29
User Word 2	User Word 6	User Word 10	User Word 14	User Word 18	User Word 22	User Word 26	User Word 30
User Word 3	User Word 7	User Word 11	User Word 15	User Word 19	User Word 23	User Word 27	User Word 31

因为EPC/User存储区的划分，根据下表所示，User块的数量依靠EPC长度来决定表5-1-6-4.

EPC Length 0 thru 6	EPC Length 7 thru 10	EPC Length 11 thru 14	EPC Length 15 thru 18	EPC Length 19 thru 22	EPC Length 23 thru 26	EPC Length 27 thru 30	EPC Length 31
Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0
Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	N.A.
Block 2	Block 2	Block 2	Block 2	Block 2	Block 2	N.A.	N.A.
Block 3	Block 3	Block 3	Block 3	Block 3	N.A.	N.A.	N.A.
Block 4	Block 4	Block 4	Block 4	N.A.	N.A.	N.A.	N.A.
Block 5	Block 5	Block 5	N.A.	N.A.	N.A.	N.A.	N.A.
Block 6	Block 6	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Block 7	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

N.A. = Not available

### 5.1.6.3 响应帧格式

表 5-1-6-5 成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 5-1-6-6 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-6-7 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1 : 响应中含 ECODE 数据段 0 : 响应中不含 ECODE 数据段

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

#### 5.1.6.4 相关 API 函数

函数名	说明
UhfAlienBlockReadLockFromSingleTag();	Alien Higgs-3 IC 标签的 bank3 读锁定(不指定 UII)

#### 5.1.6.5 命令示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 07 61 00 00 00 00 80 55	成功 : AA 11 61 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55
	失败 : AA 03 61 80 55

## 5.1.7 BlockPermaLock(指定 UII)

### 5.1.7.1 功能简介

使用 BlockPermaLock 命令，将阻止对标签 User 区（bank3）写操作，一旦使用了该命令，标签数据区将永久锁定，不可解锁，使用该命令必须指定 UII。

### 5.1.7.2 命令帧格式

表 5-1-7-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	Read/Lock	BANK	PTR	RANGE	MASK	UII	*CRC16	EOF
长度	1	EBV	1	4	1	1	1	1	2		2	1

APWD：为 ACCESS PASSWORD.

Read/Lock：1 bit. 如果要读锁定位，改位置0，如果要永久锁定，改为置1.

BANK: 必须为bank3

MASK：16bit对应Block Perma Locks.

表 5-1-7-2 发送Perma lock bits的顺序:

发送永久锁定位的顺序，逻辑1为对应的数据块被锁定。

1st	2nd	3rd	4th	5th	6th	7th	8th	9 <sup>th</sup> thru16th
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	0x00

表 5-1-7-3 Block与User区的对应图

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
User Word 0	User Word 4	User Word 8	User Word 12	User Word 16	User Word 20	User Word 24	User Word 28
User Word 1	User Word 5	User Word 9	User Word 13	User Word 17	User Word 21	User Word 25	User Word 29
User Word 2	User Word 6	User Word 10	User Word 14	User Word 18	User Word 22	User Word 26	User Word 30
User Word 3	User Word 7	User Word 11	User Word 15	User Word 19	User Word 23	User Word 27	User Word 31

因为EPC/User存储区的划分，根据下表所示，User块的数量依靠EPC长度来决定表 5-1-7-4.

EPC Length 0 thru 6	EPC Length 7 thru 10	EPC Length 11 thru 14	EPC Length 15 thru 18	EPC Length 19 thru 22	EPC Length 23 thru 26	EPC Length 27 thru 30	EPC Length 31
Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0
Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	N.A.
Block 2	Block 2	Block 2	Block 2	Block 2	Block 2	N.A.	N.A.
Block 3	Block 3	Block 3	Block 3	Block 3	N.A.	N.A.	N.A.
Block 4	Block 4	Block 4	Block 4	N.A.	N.A.	N.A.	N.A.
Block 5	Block 5	Block 5	N.A.	N.A.	N.A.	N.A.	N.A.
Block 6	Block 6	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Block 7	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

N.A. = Not available

### 5.1.7.3 响应帧格式

表 5-1-7-5 成功响应 Read/Lock = 0

数据段	SOF	LEN	CMD	STATUS	Permalock bits	*CRC16	EOF
长度	1	EBV	1	1	2	2	1

表 5-1-7-6 成功响应 Read/Lock = 1

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-1-7-6 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-7-7 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1 : 响应中含 ECODE 数据段 0 : 响应中不含 ECODE 数据段

#### 5.1.7.4 相关 API 函数

函数名	说明
UhfAlienBlockPermaLockByEPC();	Alien Higgs-3 IC 标签的 bank3 永久锁定(指定 UII)

#### 5.1.7.5 命令示例

成功响应 Read/Lock = 1

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 1A 62 00 00 00 01 03 00 01 80 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55	Success : AA 03 62 00 55
	Fail : AA 03 62 80 55

成功响应 Read/Lock = 0

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 1A 62 00 00 00 00 03 00 01 80 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55	成功 : AA 05 62 01 80 00 55
	失败 : AA 03 62 80 55

## 5.1.8 BlockPermaLock(不指定 UII)

### 5.1.8.1 功能简介

使用 BlockPermaLock 命令，将阻止对标签 User 区（bank3）写操作，一旦使用了该命令，标签数据区将永久锁定，不可解锁。

### 5.1.8.2 命令帧格式

表 5-1-8-1 命令帧格式

数据段	SOF	LEN	CMD	APWD	Read/Lock	BANK	PTR	RANGE	MASK	*CRC16	EOF
长度	1	EBV	1	4	1	1	1	1	2	2	1

APWD：为 ACCESS PASSWORD.

Read/Lock：1 bit. 如果要读锁定位，改位置0，如果要永久锁定，改为置1.

BANK: 必须为bank3

MASK：16bit对应Block Perma Locks.

表 5-1-8-2 发送Perma lock bits的顺序:

发送永久锁定位的顺序，逻辑1为对应的数据块被锁定。

1st	2nd	3rd	4th	5th	6th	7th	8th	9 <sup>th</sup> thru16th
Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	0x00

表 5-1-8-3 Block与User区的对应图

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
User Word 0	User Word 4	User Word 8	User Word 12	User Word 16	User Word 20	User Word 24	User Word 28
User Word 1	User Word 5	User Word 9	User Word 13	User Word 17	User Word 21	User Word 25	User Word 29
User Word 2	User Word 6	User Word 10	User Word 14	User Word 18	User Word 22	User Word 26	User Word 30
User Word 3	User Word 7	User Word 11	User Word 15	User Word 19	User Word 23	User Word 27	User Word 31

因为EPC/User存储区的划分，根据下表所示，User块的数量依靠EPC长度来决定表5-1-8-4.

EPC Length 0 thru 6	EPC Length 7 thru 10	EPC Length 11 thru 14	EPC Length 15 thru 18	EPC Length 19 thru 22	EPC Length 23 thru 26	EPC Length 27 thru 30	EPC Length 31
Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0	Block 0
Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	Block 1	N.A.
Block 2	Block 2	Block 2	Block 2	Block 2	Block 2	N.A.	N.A.
Block 3	Block 3	Block 3	Block 3	Block 3	N.A.	N.A.	N.A.
Block 4	Block 4	Block 4	Block 4	N.A.	N.A.	N.A.	N.A.
Block 5	Block 5	Block 5	N.A.	N.A.	N.A.	N.A.	N.A.
Block 6	Block 6	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Block 7	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

N.A. = Not available

### 5.1.8.3 响应帧格式

表 5-1-8-5 成功响应 Read/Lock = 0

数据段	SOF	LEN	CMD	STATUS	Permalock bits	UII	*CRC16	EOF
长度	1	EBV	1	1	2		2	1

表 5-1-8-6 成功响应 Read/Lock = 1

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 5-1-8-7 失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

表 5-1-8-8 STATUS 定义

位	Bit 7 ~ 4	Bit 3 ~ 1	Bit 0
描述	通用位	保留	1 : 响应中含 ECODE 数据段 0 : 响应中不含 ECODE 数据段

备注：该命令的 STATUS Bit0 只在 Bit 7 为 0 时有效。

### 5.1.8.4 相关 API 函数

函数名	说明
UhfAlienBlockPermaLockFromSingleTag();	Alien Higgs-3 IC 标签的 bank3 永久锁定(不指定 UII)

### 5.1.8.5 命令示例

成功响应 Read/Lock = 1

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 1A 62 00 00 00 00 01 03 00 01 80 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55	成功 : AA 11 63 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55 失败 : AA 03 63 80 55

成功响应 Read/Lock = 0

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 0C 63 00 00 00 00 00 03 00 01 80 00 55	成功 : AA 13 63 01 80 00 34 00 00 1D 90 61 53 19 01 32 19 10 4D A9 55 失败 : AA 03 63 80 55

## 5.2 NXP 标签的定制命令

以下第 5.2 小节的命令为 NXP 标签的私有命令，只适用于 NXP 标签，对其他厂商的标签无效。

### 5.2.1 修改 Configuration Word (指定 UII)

#### 5.2.1.1 功能简介

该命令用于指定 NXP 标签的 UII 后，修改标签的 Configuration Word，Configuration Word 为 NXP 标签独有，位于标签 UII 区的地址 200h 处，长度为 1 个字。

#### 5.2.1.2 命令帧格式

表 5-2-1-1 修改 Configuration Word (指定 UII) 命令帧格式

数据段	SOF	LEN	CMD	APWD	Config_Word	UII	*CRC16	EOF
长度	1	EBV	1	4	2		2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

Config\_Word：标签的功能配置字。

#### 5.2.1.3 响应帧格式

表 5-2-1-2 修改 Configuration Word (指定 UII) 命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	Config_Word	*CRC16	EOF
长度	1	EBV	1	1	2	2	1

表 5-2-1-4 修改 Configuration Word (指定 UII) 命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE (Error Code) 数据段的定义见本文的附录 E。

表 5-2-1-5 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

#### 5.2.1.4 相关 API 函数

函数名	说明
UhfNxpChangeConfigByEPC();	设置 NXP 标签的 configuration Word(指定 UII)



**5.2.1.5 通讯示例**

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 16 64 12 34 56 78 00 41 30 00 E2 00 68 05 00 00 00 00 00 00 00 00 55	成功 : AA 05 64 00 00 41 55
	失败 : AA 03 64 80 55

## 5.2.2 修改 Configuration Word ( 不指定 UII )

### 5.2.2.1 功能简介

该命令不用指定 NXP 标签的 UII，修改目标标签的 Configuration Word。Configuration Word 为 NXP 标签独有，位于标签 UII 区的地址 200h 处，长度为 1 个字。

### 5.2.2.2 命令帧格式

表 5-6-2-1 修改 Configuration Word ( 不指定 UII ) 命令帧格式

数据段	SOF	LEN	CMD	APWD	Config_Word	*CRC16	EOF
长度	1	EBV	1	4	2	2	1

数据段说明：

APWD： 标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

Config\_Word： 标签的功能配置字分区。

### 5.2.2.3 响应帧格式

表 5-2-2-2 修改 Configuration Word ( 不指定 UII ) 命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	Config_Word	UII	*CRC16	EOF
长度	1	EBV	1	1	2		2	1

表 5-2-2-3 修改 Configuration Word ( 不指定 UII ) 命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE ( Error Code ) 数据段的定义见本文的附录 E。

表 5-2-1-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.2.4 相关 API 函数

函数名	说明
UhfNxpChangeConfigFromSingleTag();	设置 NXP 标签的 configuration Word(不指定 UII)

### 5.2.2.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 08 65 12 34 56 78 00 41 55	成功：AA 13 65 00 00 41 30 00 E2 00 68 05 00 00 00 00 00 00 00 00 55
	失败：AA 03 65 80 55

## 5.2.3 设置读保护功能（指定 UII）

### 5.2.3.1 功能简介

该命令用于指定 NXP 标签的 UII，设置标签的读保护功能，将标签的 EPC 和 TID 区数据屏蔽，对外显示为 0，不影响标签正常识别。

### 5.2.3.2 命令帧格式

表 5-2-3-1 设置读保护（指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	UII	*CRC16	EOF
长度	1	EBV	1	4		2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

### 5.2.3.3 响应帧格式

表 5-2-3-2 设置读保护（指定 UII）命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-2-3-3 设置读保护（指定 UII）命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 5-2-3-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.3.4 相关 API 函数

函数名	说明
UhfNxpReadProtectByEPC();	使能 NXP 标签的 EPC 和 TID 数据区的读保护（指定 UII）

### 5.2.3.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
AA 14 66 12 34 56 78 30 00 E2	成功：AA 03 66 00 55
00 68 05 00 00 00 00 00 00 00	失败：AA 03 66 80 55
00 55	

## 5.2.4 设置读保护功能（不指定 UII）

### 5.2.4.1 功能简介

该命令不用指定 NXP 标签的 UII，设置标签的读保护功能，将标签的 EPC 和 TID 区数据屏蔽，对外显示为 0，不影响标签正常识别。

### 5.2.4.2 命令帧格式

表 5-6-4-1 设置读保护（不指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	*CRC16	EOF
长度	1	EBV	1	4	2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

### 5.2.4.3 响应帧格式

表 5-2-4-2 设置读保护（不指定 UII）命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	UII	*CRC16	EOF
长度	1	EBV	1	1		2	1

表 5-2-4-3 设置读保护（不指定 UII）命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 5-2-4-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.4.4 相关 API 函数

函数名	说明
UhfNxpReadProtectFromSingleTag();	使能 NXP 标签的 EPC 和 TID 数据区的读保护（不指定 UII）

### 5.2.4.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
AA 06 67 12 34 56 78 55	成功：AA 11 67 00 30 00 E2 00 68 05 00 00 00 00 00 00 00 00 55 失败：AA 03 67 80 55

## 5.2.5 清除读保护功能（指定 UII）

### 5.2.5.1 功能简介

该命令用于指定 NXP 标签的 UII，清除标签的读保护功能，将标签的 EPC 和 TID 区数据正常显示。

### 5.2.5.2 命令帧格式

表 5-2-5-1 清除读保护（指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	UII	*CRC16	EOF
长度	1	EBV	1	4		2	1

数据段说明：

APWD： 标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

### 5.2.5.3 响应帧格式

表 5-2-5-2 清除读保护（指定 UII）命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-2-5-3 清除读保护（指定 UII）命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 5-2-5-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.5.4 相关 API 函数

函数名	说明
UhfNxpResetReadProtectByEPC();	清除 NXP 标签的对 EPC 和 TID 数据的读保护功能（指定 UII）

### 5.2.5.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
AA 14 68 12 34 56 78 30 00 E2	成功：AA 03 68 00 55
00 68 05 00 00 00 00 00 00 00	失败：AA 03 68 80 55
00 55	

## 5.2.6 清除读保护功能（不指定 UII）

### 5.2.6.1 功能简介

该命令不用指定 NXP 标签的 UII，清除标签的读保护功能，将标签的 EPC 和 TID 区数据正常显示。

### 5.2.6.2 命令帧格式

表 5-2-6-1 清除读保护（不指定 UII）命令帧格式

数据段	SOF	LEN	CMD	APWD	*CRC16	EOF
长度	1	EBV	1	4	2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

### 5.2.6.3 响应帧格式

表 5-2-6-2 清除读保护（不指定 UII）命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-2-6-3 清除读保护（不指定 UII）命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE（Error Code）数据段的定义见本文的附录 E。

表 5-2-6-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.6.4 相关 API 函数

函数名	说明
UhfNxpResetReadProtectFromSingleTag();	清除 NXP 标签的对 EPC 和 TID 数据的读保护功能（不指定 UII）

### 5.2.6.5 通讯示例

发送命令格式（hex）	返回数据格式（hex）
AA 06 69 12 34 56 78 55	成功：AA 11 69 00 30 00 E2 00 68 05 00 00 00 00 00 00 55
	失败：AA 03 69 80 55

## 5.2.7 配置 EAS 标志 (指定 UII)

### 5.2.7.1 功能简介

该命令指定 NXP 标签的 UII，配置 EAS 功能，用于设定 PSF 位，只有改为置 1，EAS\_Alarm 命令才有效。

### 5.2.7.2 命令帧格式

表 5-2-7-1 配置 EAS (指定 UII) 命令帧格式

数据段	SOF	LEN	CMD	APWD	ChangeEAS	UII	*CRC16	EOF
长度	1	EBV	1	4	1		2	1

数据段说明：

APWD：标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

ChangeEAS: 1 为设置 PSF 位，0 为清除 PSF 位。

### 5.2.7.3 响应帧格式

表 5-2-7-2 配置 EAS (指定 UII) 命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-2-7-3 配置 EAS (指定 UII) 命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE (Error Code) 数据段的定义见本文的附录 E。

表 5-2-7-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.7.4 相关 API 函数

函数名	说明
UhfNxpChangeEasByEPC();	设置 NXP 标签的 EAS 位 (指定 UII)

### 5.2.7.5 通讯示例

发送命令格式 (hex)	返回数据格式 (hex)
AA 15 6A 12 34 56 78 01 30 00 E2 00	成功：AA 03 6A 00 55
68 05 00 00 00 00 00 00 00 00 55	失败：AA 03 6A 80 55

## 5.2.8 配置 EAS 标志 ( 不指定 UII )

### 5.2.8.1 功能简介

该命令不用指定 NXP 标签的 UII，配置 EAS 功能，用于设定 PSF 位。

### 5.2.8.2 命令帧格式

表 5-2-8-1 配置 EAS ( 不指定 UII ) 命令帧格式

数据段	SOF	LEN	CMD	APWD	ChangeEAS	*CRC16	EOF
长度	1	EBV	1	4	1	2	1

数据段说明：

APWD： 标签的数据访问密码。

当用户欲读取的数据存储区为非 Reserved 存储区时，将 APWD 置为 0x00000000。

当用户欲读取的数据存储区为 Reserved 存储区时，APWD=标签的 ACCESS 密钥。

ChangeEAS: 1 为设置 PSF 位，0 为清除 PSF 位。

### 5.2.8.3 响应帧格式

表 5-2-8-2 配置 EAS ( 不指定 UII ) 命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	*CRC16	EOF
长度	1	EBV	1	1	2	1

表 5-2-8-3 配置 EAS ( 不指定 UII ) 命令失败响应帧格式

数据段	SOF	LEN	CMD	STATUS	*ECODE	*CRC16	EOF
长度	1	EBV	1	1	1	2	1

备注：ECODE ( Error Code ) 数据段的定义见本文的附录 E。

表 5-2-8-4 数据 STATUS 定义

位	Bit 7	Bit 6	Bit5	Bit 4	Bit 3~1	Bit 0
描述	1 = 执行命令失败 0 = 执行命令成功	1 = CRC16 验证失败 0 = CRC16 验证成功	保留	保留	保留	1：响应中含 ECODE 数据段 0：响应中不含 ECODE 数据段

### 5.2.8.4 相关 API 函数

函数名	说明
UhfNxpChangeEasFromSingleTag();	设置 NXP 标签的 EAS 位 ( 不指定 UII )

### 5.2.8.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 07 6B 12 34 56 78 00 55	成功：AA 11 6B 00 30 00 E2 00 68 05 00 00 00 00 00 00 00 00 55
	失败：AA 03 6B 80 55



## 5.2.9 EAS 报警

### 5.2.9.1 功能简介

该命令用于 NXP 标签返回 64 位的 EAS Code,在使用该命令时 ,如果配置字中 PSF 位为 1 ,标签返回 64 位的 EAS Code,否则 , 不返回。

### 5.2.9.2 命令帧格式

表 5-2-9-1 配置 EAS ( 指定 UII ) 命令帧格式

数据段	SOF	LEN	CMD	*CRC16	EOF
长度	1	EBV	1	2	1

### 5.2.9.3 响应帧格式

表 5-2-9-2 配置 EAS ( 指定 UII ) 命令成功响应帧格式

数据段	SOF	LEN	CMD	STATUS	EAS CODE	*CRC16	EOF
长度	1	EBV	1	1	8	2	1

表 5-2-9-3 配置 EAS ( 指定 UII ) 命令失败响应帧格式

备注：通用格式

### 5.2.9.4 相关 API 函数

函数名	说明
UhfNxpEasAlarm();	获得 NXP 标签的 EAS Code

### 5.2.9.5 通讯示例

发送命令格式 ( hex )	返回数据格式 ( hex )
AA 02 6C 55	成功 : AA 0B 6C 00 69 0A EC 7C D2 15 D8 F8 55
	失败 : AA 03 6C 80 55

# 附录 A：命令摘要

表 A-1 JIURAY 超高频模块命令摘要

Command	Parameter1	Parameter2	Parameter 3	Parameter4	Parameter5	Parameter6
RLM_GET_STATUS						
RLM_GET_POWER						
RLM_SET_POWER	OPTION	POWER				
RLM_GET_FRE						
RLM_SET_FRE	FREMODE	FREBASE	BF	CN	SPC	FREHOP
RLM_GET_VERSION						
RLM_READ_UID						
RLM_INVENTORY						
RLM_INVENTORY_ANTI	Q					
RLM_STOP_GET						
RLM_READ_DATA	APWD	BANK	PTR	CNT	UII	
RLM_WRITE_DATA	APWD	BANK	PTR	CNT	DATA	UII
RLM_ERASE_DATA	APWD	BANK	PTR	CNT	UII	
RLM_LOCK_MEM	APWD	LOCKDATA	UII			
RLM_KILL_TAG	KILLPWD	UII				
RLM_INVENTORY_SINGLE						
RLM_BLOCK_WRITE_DATA	APWD	BANK	PTR	CNT	DATA	UII
RLM_SINGLE_READ_DATA	APWD	BANK	PTR	CNT		
RLM_SINGLE_WRITE_DATA	APWD	BANK	PTR	CNT	DATA	
RLM_SINGLE_ERASE_DATA	APWD	BANK	PTR	CNT		
RLM_SINGLE_LOCK_MEM	APWD	LOCKDATA				
RLM_SINGLE_KILL_TAG	KILLPWD					
RLM_SINGLE_BLOCK_WRITE_DATA	APWD	BANK	PTR	CNT		
RLM_ANTI_COLLISION_READ_DATA						

RLM_GET_REG	RADD	RLEN				
RLM_SET_REG	RADD	RLEN	REG			
RLM_DEFAULT_REG						
RLM_SAVE_REG						
RLM_ADD_SELECT	SINDEX	RECORD				
RLM_DELETE_SELECT	SINDEX					
RLM_GET_SELECT	SINDEX	SNUM				
RLM_CHOOSE_SELECT	SINDEX	SNUM				
T						
RLM_ENTER_SLEEP						
RLM_START_UPDATE						
RLM_SEND_INVERSE	RN					
RLM_START_TRANS	FILESIZE					
RLM_TRAN_PACKAGE	PACKNUM	LASTPACK	DATALEN	TRANDATA		
RLM_END_UPDATE						

## 附录 B：Extensible bit Vectors (EBV)

EBV ( Extensible Bit Vector ) 是一种能表示可延伸数据的数据结构。本文档中提到的 EBV 是以字节为单位的数组，数组中每个字节的最高位是延伸位。如果延伸位为 0，则表示该字节是最后一个字节；如果延伸位为 1，则表示后续还有有效字节。EBV 格式数据串表示的有效数据是从左到右忽略延伸位的比特流。

JIURAY模块 只支持一个字节和两个字节的 EBV 数据，其格式如下：

0	X X X X X X X	
1	X X X X X X X	0 X X X X X X X

其中每个字节的最高位是延伸位。当 EBV 需要表示的数小于等于 127 时可用一个字节，而当 EBV 需要表示的数大于 127 小于 16384 时需用两个字节。例如，

12：00001100，

130：1000000100000010。

## 附录 C：插入字节实例

在实际通信过程中，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。插入字节不计入 LENGTH。例如：

- ◆ 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

## 附录 D：UII 格式

本文档中所谓的 UII 包含 PC bits。UII 的前两个字节是 PC (Protocol-control) 位，其格式见表 D-1。

表 D-1 PC bits 格式

Bits 0 ~ 4	Bits 5 ~ 6	Bits 7 ~ 15
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)

注：UII 从低位开始传输。

PC 的前五位表示 PC 和 UII 的总体长度。例如，

PC bits 0~4 (bin)                  PC+UII 长度 (字节)

00000                                  2

00001                                  4

00010                                  6

...                                      ...

整段 UII 的数据信息是由 PC 加上 EPC 构成。所以，用户可以通过 PC 的前 5 位计算出整个 UII 的长度。

公式： $\text{LengthUII} = (((\text{UII}[0] \gg 3) \& 0x1F) + 1) * 2$

单位：字节。

例如：

1、一张标签的 UII(hex) = 30 00 12 34 56 78 53 40 00 00 12 34 85 1A

UII[0] = 0x30;

根据公式： $\text{LengthUII} = (((\text{UII}[0] \gg 3) \& 0x1F) + 1) * 2$

计算结果： $\text{LengthUII} = 14$

所以整段卡号的长度就为 14 个字节。

2、如果您想写一张标签的 UII 总长度为 12 个字节 根据公

式： $\text{LengthUII} = (((\text{UII}[0] \gg 3) \& 0x1F) + 1) * 2 = 12$  计算结

果： $\text{UII}[0] = 0x28$

所以将 UII[0] = 0x28 写入 UII 的第一个字节之后的卡号为：28 00 12 34 56 78 53 40 00 00 12 34

综上所述，实际计算长度为 UII[0] 的前 5 个比特，具体细节请参考 ISO18000-6-C 协议。

## 附录 E：Error codes

对标签进行数据操作（读取标签数据、写入标签数据、擦除标签数据、锁定标签、销毁标签）时，如果标签遇到错误则会返回错误码（Error Code）。

表 E-1 标签错误码

Error Code 支持	Error Code 值 ( bin )	Error Code 名	Error 描述
Error-specific	00000000	其他错误	其他错误码未定义的错误
	00000011	存储空间溢出或未支持的 PC 值	指定的存储空间不存在或标签不支持指定的 PC 值
	00000100	存储空间被锁定	指定存储空间被锁定，不能进行读/写操作
	00001011	电量不足	因电量不足不能进行写入操作
Non-specific	00001111	不明错误	标签不支持 Error-specific 码

## 附录 F：Lock-Command Payload

Lock-Command Payload 是二十位的数据，高十位是 Mask，低十位是 Action。其格式见表

F-1。当 Mask

置为 1 时对应的 Action 位有效。Action 位的含义见表 F-2。

表 F-1 Lock-Command Payload 数据格式

Kill password		Access password		UII memory		TID memory		User memory	
19	18	17	16	15	14	13	12	11	10
Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write	Skip/ Write
9	8	7	6	5	4	3	2	1	0
Pwd read/ write	Perma lock	Pwd read/ write	Perma lock	Pwd write	Perma lock	Pwd write	Perma lock	Pwd write	Perma lock

表 F-2 Lock Action 位

Pwd-write	Permalock	描述
0	0	未锁定状态：相应数据存储区（备注）支持在 OPEN 或 SECURED 状态下的写入操作
0	1	永久不可锁定状态：相应数据存储区（备注）支持在 OPEN 或 SECURED 状态下的写入操作，且该数据存储区不可被锁定
1	0	锁定状态：相应数据存储区仅（备注）支持在 SECURED 状态下的写入操作，OPEN 状态下不可写入
1	1	永久锁定状态：相应数据存储区（备注）不支持任何状态下的写入操作，且不可被解锁
Pwd-read/write	Permalock	描述
0	0	未锁定状态：相应密钥字段（备注）支持在 OPEN 或 SECURED 状态下的写入操作
0	1	永久不可锁定状态：相应密钥字段（备注）支持在 OPEN 或 SECURED 状态下的写入操作，且该数据存储区不可被锁定
1	0	锁定状态：相应密钥字段（备注）仅支持在 SECURED 状态下的写入操作，OPEN 状态下不可写入
1	1	永久锁定状态：相应密钥字段（备注）不支持任何状态下的写入操作，且不可被解锁

【备注】相应存储区是指：UII 存储区、TID 存储区、

User 存储区 响应密钥字段是指：Kill password 字

段，Access password 字段