

# R-600 通讯协议用户手册\_V2.38



1 通信协议结构	4
1.1 RS232 参数设置	4
1.2 数据包格式定义	5
1.2.1 上位机指令数据包格式定义	5
1.2.2 读写器返回数据包定义	6
2 指令集定义	7
2.1 系统设置指令 .....	10
2.1.1 CMD_RESET	9
2.1.2 CMD_SET_UART_BAUDRATE	11
2.1.3 CMD_GET_FIRMWARE_VERSION	13
2.1.4 CMD_SET_READER_ADDRESS	14
2.1.5 CMD_SET_WORK_ANTENNA	15
2.1.6 CMD_GET_WORK_ANTENNA	17
2.1.7 CMD_SET_OUTPUT_POWER	18
2.1.8 CMD_GET_OUTPUT_POWER	20
2.1.9 CMD_SET_FREQUENCY_REGION	21
2.1.10 CMD_GET_FREQUENCY_REGION	24
2.1.11 CMD_SET_BEEPER_MODE	25
2.1.12 CMD_GET_READER_TEMPERATURE	28

2.1.13	CMD_READ_GPIO_VALUE	30
2.1.14	CMD_WRITE_GPIO_VALUE	31
2.1.15	CMD_SET_ANT_CONNECTION_DETECTOR	32
2.1.16	CMD_GET_ANT_CONNECTION_DETECTOR	33
2.1.17	CMD_SET_TEMPORARY_OUTPUT_POWER	34
2.1.18	CMD_SET_READER_IDENTIFIER	36
2.1.19	CMD_GET_READER_IDENTIFIER	38
2.1.20	CMD_SET_RF_LINK_PROFILE	39
2.1.21	CMD_GET_RF_LINK_PROFILE	41
2.1.22	CMD_GET_RF_PORT_RETURN_LOSS	43
<b>2.2</b>	<b>18000-6C 标签操作命令</b>	<b>45</b>
2.2.1	CMD_INVENTORY	45
2.2.2	CMD_READ	47
2.2.3	CMD_WRITE	49
2.2.4	CMD_LOCK	51
2.2.5	CMD_KILL	53
2.2.6	CMD_SET_ACCESS_EPC_MATCH	55
2.2.7	CMD_GET_ACCESS_EPC_MATCH	56
2.2.8	CMD_REAL_TIME_INVENTORY	57
2.2.9	CMD_FAST_SWITCH_ANT_INVENTORY	59
2.2.10	CMD_CUSTOMIZED_SESSION_TARGET_INVENTORY	62

2.2.11	CMD_SET_IMPINJ_FAST_TID	64
2.2.12	CMD_SET_AND_SAVE_IMPINJ_FAST_TID	66
2.2.13	CMD_GET_IMPINJ_FAST_TID	66
2.3	ISO 18000-6B 标签操作命令	67
2.3.1	CMD_ISO18000_6B_INVENTORY	67
2.3.2	CMD_ISO18000_6B_READ	69
2.3.3	CMD_ISO18000_6B_WRITE	71
2.3.4	CMD_ISO18000_6B_LOCK	73
2.3.5	CMD_ISO18000_6B_QUERY_LOCK	75
2.4	缓存操作命令	77
2.4.1	CMD_GET_INVENTORY_BUFFER	77
2.4.2	CMD_GET_AND_RESET_INVENTORY_BUFFER	79
2.4.3	CMD_GET_INVENTORY_BUFFER_TAG_COUNT	79
2.4.4	CMD_RESET_INVENTORY_BUFFER	80
3	错误代码表	81
4	频率参数对应表	84
5	RSSI 参数对应表	86
6	校验和计算方法(C 语言描述)	88

# 1 通信协议结构

此通信协议是上位机通过串行通信接口操作读写器的通信规范。

命令和响应数据由连续的字节流组成数据包，长度可变，并采用校验和方法进行检错。

## 1.1 RS232 参数设置

物理接口符合 RS-232 规范要求。

1 位起始位、8 位数据位、1 位停止位、无奇偶校验。

通信波特率设计为 38400bps、115200bps 可选。默认波特率为 115200bps。

## 1.2 数据包格式定义

### 1.2.1 上位机指令数据包格式定义

Head	Len	Address	Cmd	Data	Check
0xA0	1 Byte	1 Byte	1 Byte	N Bytes	1 Byte
参数说明	Head		数据包头，每包数据均以 0xA0 开始。		
	Len		数据包从 Len 后面开始的字节数，不包含 Len 本身。		
	Address		读写器地址。供 RS-485 接口串联时使用。一般地址从 0 ~ 254(0xFE) , 255 ( 0xFF ) 为公用地址。读写器接收自身地址和公用地址的命令。		
	Cmd		命令码。		
	Data		命令参数。		
	Check		校验和，除校验和本身外所有字节的校验和。		

### 1.2.2 读写器返回数据包定义

Head	Len	Address	Data	Check
0xA0	1 Byte	1 Byte	N Bytes	1 Byte
参数说明	Head		数据包头，每包数据均以 0xA0 开始。	
	Len		数据包从 Len 后面开始的字节数，不包含 Len 本身。	
	Address		读写器自身的地址。	
	Data		读写器返回的数据。	
	Check		校验和，除校验和本身外所有字节的校验和。	

## 2 指令集定义

指令集一览表

序号	命令码	名 称	描 述
读写器操作命令			
1	0x70	cmd_reset	复位读写器
2	0x71	cmd_set_uart_baudrate	设置串口通讯波特率
3	0x72	cmd_get_firmware_version	读取读写器固件版本
4	0x73	cmd_set_reader_address	设置读写器地址
5	0x74	cmd_set_work_antenna	设置读写器工作天线
6	0x75	cmd_get_work_antenna	查询当前天线工作天线
7	0x76	cmd_set_output_power	设置读写器射频输出功率
8	0x77	cmd_get_output_power	查询读写器当前输出功率
9	0x78	cmd_set_frequency_region	设置读写器工作频率范围
10	0x79	cmd_get_frequency_region	查询读写器工作频率范围
11	0x7A	cmd_set_beeper_mode	设置蜂鸣器状态
12	0x7B	cmd_get_reader_temperature	查询当前设备的工作温度
13	0x60	cmd_read_gpio_value	读取 GPIO 电平
14	0x61	cmd_write_gpio_value	设置 GPIO 电平
15	0x62	cmd_set_ant_connection_detector	设置天线连接检测器状态
16	0x63	cmd_get_ant_connection_detector	读取天线连接检测器状态



17	0x66	cmd_set_temporary_output_power	设置读写器临时射频输出功率
18	0x67	cmd_set_reader_identifier	设置读写器识别码
19	0x68	cmd_get_reader_identifier	读取读写器识别码
20	0x69	cmd_set_rf_link_profile	设置射频链路的通讯速率
21	0x6A	cmd_get_rf_link_profile	读取射频链路的通讯速率
22	0x7E	cmd_get_rf_port_return_loss	测量天线端口的回波损耗
18000-6C 命令			
23	0x80	cmd_inventory	盘存标签
24	0x81	cmd_read	读标签
25	0x82	cmd_write	写标签
26	0x83	cmd_lock	锁定标签
27	0x84	cmd_kill	灭活标签
28	0x85	cmd_set_access_epc_match	匹配 ACCESS 操作的 EPC 号
29	0x86	cmd_get_access_epc_match	查询匹配的 EPC 状态
30	0x89	cmd_real_time_inventory	盘存标签(实时上传标签数据)
31	0x8A	cmd_fast_switch_ant_inventory	快速轮询多个天线盘存标签
32	0x8B	cmd_customized_session_target_inventory	自定义 session 和 target 盘存
33	0x8C	cmd_set_impinj_fast_tid	设置 Monza 标签快速读 TID (设置不被保存至内部 FLASH)
34	0x8D	cmd_set_and_save_impinj_fast_tid	设置 Monza 标签快速读 TID (设置被保存至内部 FLASH)

35	0x8E	cmd_get_impinj_fast_tid	查询当前的快速 TID 设置
ISO18000-6B 命令			
36	0xB0	cmd_iso18000_6b_inventory	盘存 18000-6B 标签
37	0xB1	cmd_iso18000_6b_read	读 18000-6B 标签
38	0xB2	cmd_iso18000_6b_write	写 18000-6B 标签
39	0xB3	cmd_iso18000_6b_lock	锁定 18000-6B 标签
40	0xB4	cmd_iso18000_6b_query_lock	查询 18000-6B 标签
缓存操作命令			
41	0x90	cmd_get_inventory_buffer	提取标签数据保留缓存备份
42	0x91	cmd_get_and_reset_inventory_buffer	提取标签数据并删除缓存
43	0x92	cmd_get_inventory_buffer_tag_count	查询缓存中已读标签个数
44	0x93	cmd_reset_inventory_buffer	清空标签数据缓存

## 2.1 系统设置指令

### 2.1.1 cmd\_reset

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x70	

◆**操作成功**：无数据返回，读写器重启，蜂鸣器响一声。

◆**操作失败**：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x70		
参数说明	ErrorCode	错误代码			

## 2.1.2 cmd\_set\_uart\_baudrate

上位机指令数据包：

Head	Len	Address	Cmd	BaudRate	Check
0xA0	0x04		0x71		
参数说明	BaudRate	0x03	38400 bps		
		0x04	115200 bps		

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x71	CommandSucces ss	

读写器成功收到此命令帧后，用先前波特率返回应答数据包，然后重新启动读写器。

新的波特率保存在内部 FLASH 中，断电不丢失。

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x71		
参数说明	ErrorCode	错误代码			

## 2.1.3 cmd\_get\_firmware\_version

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x72	

读写器返回数据包：

Head	Len	Address	Cmd	Major	Minor	Check
0xA0	0x05		0x72			
参数说明	Major	固件主版本号。				
	Minor	固件次版本号。				

## 2.1.4 cmd\_set\_reader\_address

上位机指令数据包：

Head	Len	Address	Cmd	Address	Check
0xA0	0x04		0x73		
参数说明	Address	读写器地址，取值范围 0 – 254。			

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x73	CommandSucces ss	

新的读写器地址立即生效，并被写入 FLASH 保存，断电不丢失。

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x73		
参数说明	ErrorCode	错误代码			

## 2.1.5 cmd\_set\_work\_antenna

上位机指令数据包：

Head	Len	Address	Cmd	AntennaID	Check
0xA0	0x04		0x74		
参数说明	AntennaID	天线号	0x00	天线 1	
			0x01	天线 2	
			0x02	天线 3	
			0x03	天线 4	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x74	CommandSucces ss	

◆操作失败：



读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x74		
参数说明	ErrorCode	错误代码			

## 2.1.6 cmd\_get\_work\_antenna

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x75	

读写器返回数据包：

Head	Len	Address	Cmd	AntennaID	Check
0xA0	0x04		0x75		
参数说明	AntennaID (天线号)		0x00	天线 1	
			0x01	天线 2	
			0x02	天线 3	
			0x03	天线 4	

## 2.1.7 cmd\_set\_output\_power

上位机指令数据包：

Head	Len	Address	Cmd	RfPower	Check
0xA0	0x04		0x76		
参数说明	RfPower	RF 输出功率,取值范围 0-33(0x00 – 0x21), 单位 dBm。			

或者：

Head	Len	Address	Cmd	Power1	Power2	Power3	Power4	Check
0xA0	0x07		0x76					
参数说明	Power1	天线 1 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power2	天线 2 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power3	天线 3 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power4	天线 4 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
------	-----	---------	-----	-----------	-------

0xA0	0x04		0x76	CommandSucces ss	
------	------	--	------	---------------------	--

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x76		
参数说明	ErrorCode	错误代码			

操作成功后输出功率值将被保存在内部的 Flash 中，断电后不丢失。

注意：

★此命令耗时将超过 100mS。

★如果需要动态改变射频输出功率，请使用 cmd\_set\_temporary\_output\_power

命令，否则将会影响 Flash 的使用寿命。

## 2.1.8 cmd\_get\_output\_power

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x77	

如果所有天线的功率设置相同，读写器返回数据包：

Head	Len	Address	Cmd	OutputPower	Check
0xA0	0x04		0x77		
参数说明	OutputPower	读写器当前的射频输出功率。			

否则返回：

Head	Len	Address	Cmd	Power1	Power2	Power3	Power4	Check
0xA0	0x07	s	0x77					
参数说明	Power1	天线 1 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power2	天线 2 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power3	天线 3 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						
	Power4	天线 4 输出功率，取值范围 0-33(0x00 – 0x21), 单位 dBm。						

## 2.1.9 cmd\_set\_frequency\_region

上位机指令数据包：

射频频谱的定义有两种方法。

方法一：使用系统默认的频点（参见频率参数对应表），上位机发送：

Head	Len	Address	Cmd	Region	StartFreq	EndFreq	Check
0xA0	0x06		0x78				
参数说明	Region	射频规范	0x01	FCC			
			0x02	ETSI			
			0x03	CHN			
	StartFreq	频率起始点		可以在射频规范的频率范围内再设置跳频的范围。参数所对应的频率请参见频率参数对应表。参数的设置规则为：1，起始频率与结束频率不能超过射频规范的范围。2，起始频率必须低于结束频率。3，起始频率等于结束频率则定频发射。			
	EndFreq	频率结束点					

方法二：用户自定义频谱，上位机发送以下命令：

Head	Len	Address	Cmd	Region	FreqSpace	RreqQuantity	StartFreq	Check
0xA0	0x09		0x78	0x04			3bytes	
参数说明	Region	射频规范	此值固定为 0x04。					
	FreqSpace	频点间隔	频点间隔 = FreqSpace x 10KHz。					
	FreqQuantity	频点数量	包含起始频率的频点数量,1 为以起始频率定频发射。此参数必须大于 0。					
	StartFreq	起始频率	单位为 KHz。16 进制数高位在前。例如 915000KHz 则发送 0D F6 38。					

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x78	CommandSuccessful	

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x78		
参数说明	ErrorCode	错误代码			



## 2.1.10 cmd\_get\_frequency\_region

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x79	

如果使用的是系统默认频点，则读写器返回以下数据包：

Head	Len	Address	Cmd	Region	StartFreq	EndFreq	Check
0xA0	0x06		0x79				
参数说明	Region	射频规范	0x01	FCC			
			0x02	ETSI			
			0x03	CHN			
	StartFreq	频率起始点		跳频频率范围的低点。			
	EndFreq	频率结束点		跳频频率范围的高点。			

如果使用的是自定义频点，则读写器返回以下数据包：

Head	Len	Address	Cmd	Region	FreqSpace	RreqQuantity	StartFreq	Check
0xA0	0x09		0x79	0x04			3bytes	

参数说明	Region	射频规范	此值固定为 0x04。
	FreqSpace	频点间隔	频点间隔 = FreqSpace x 10KHz。
	FreqQuantity	频点数量	包含起始频率的频点数量, 1 为以起始频率定频发射。此参数必须大于 0。
	StartFreq	起始频率	单位为 KHz。16 进制数高位在前。例如 915000KHz 则返回 0D F6 38。

## 2.1.11 cmd\_set\_beeper\_mode

上位机指令数据包：

Head	Len	Address	Cmd	Mode	Check
0xA0	0x04		0x7A		
参数说明	Mode (操作标签时蜂鸣器状态)		0x00	安静	
			0x01	每次盘存后鸣响	
			0x02	每读到一张标签鸣响	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x7A	CommandSucces ss	

操作成功后此配置将保存至内部 FLASH, 断电后不丢失。

**注意：**

★读到一张标签后蜂鸣器鸣响，会占用大量处理器时间，若此选项打开，将会明显影响到读多标签（防冲突算法）的性能，此选项应作为测试功能选用。

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x7A		
参数说明	ErrorCode	错误代码			

## 2.1.12 cmd\_get\_reader\_temperature

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x7B	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	PlusMinus	Temp	Check
0xA0	0x05		0x7B			
参数说明	PlusMinus		0x00	零下		
			0x01	零上		
	Temp		摄氏度			

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x7B		
参数说明	ErrorCode	错误代码			

## 2.1.13 cmd\_read\_gpio\_value

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x60	

读写器返回数据包：

Head	Len	Address	Cmd	Gpio1	Gpio2	Check
0xA0	0x05		0x60			
参数说明		Gpio1	0x00	Gpio1 的电平为低		
			0x01	Gpio1 的电平为高		
		Gpio2	0x00	Gpio2 的电平为低		
			0x01	Gpio2 的电平为高		

## 2.1.14 cmd\_write\_gpio\_value

上位机指令数据包：

Head	Len	Address	Cmd	ChooseGpio	GpioValue	Check
0xA0	0x05		0x61			
参数说明		ChooseGpio	0x03	设置 GPIO 3		
			0x04	设置 GPIO 4		
		GpioValue	0x00	设置为低电平		
			0x01	设置为高电平		

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x61		
参数说明	ErrorCode	错误代码			



## 2.1.15 cmd\_set\_ant\_connection\_detector

上位机指令数据包：

Head	Len	Address	Cmd	DetectorSensitivity	Check
0xA0	0x04		0x62		
参数说明	DetectorSensitivity		0x00	关闭天线连接检测。	
				天线连接检测的灵敏度(端口回波损耗值)，单位 dB。值越大，对端口的阻抗匹配要求越高。	

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x62		
参数说明	ErrorCode	错误代码			

## 2.1.16 cmd\_get\_ant\_connection\_detector

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x63	

读写器返回数据包：

Head	Len	Address	Cmd	DetectorSensitivity	Check
0xA0	0x04		0x63		
参数说明	DetectorSensitivity		0x00	天线连接检测已关闭。	
				天线连接检测的灵敏度(端口回波损耗值)。	

## 2.1.17 cmd\_set\_temporary\_output\_power

上位机指令数据包：

Head	Len	Address	Cmd	RfPower	Check
0xA0	0x04		0x66		
参数说明	RfPower	RF 输出功率，取值范围 20-33(0x14 – 0x21), 单位 dBm。			

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x66	CommandSuccess	

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x66		
参数说明	ErrorCode	错误代码			

操作成功后输出功率值将不会被保存在内部的 Flash 中，重新启动或断电后输出功率将恢复至内部 Flash 中保存的输出功率值。此命令的操作速度非常快，并且不写 Flash，从而不影响 Flash 的使用寿命，适合需要反复切换射频输出功率的应用。

## 2.1.18 cmd\_set\_reader\_identifier

上位机指令数据包：

Head	Len	Address	Cmd	Identifier	Check
0xA0	0x0F		0x67	12 Bytes	
参数说明	Identifier	12 字节的读写器识别字符。			

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x67	CommandSuccess	

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x67		
参数说明	ErrorCode	错误代码			

操作成功后 12 字节的读写器识别字符串将会保存在内部的 Flash 中，断电后不丢失。

## 2.1.19 cmd\_get\_reader\_identifier

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x68	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	Identifier	Check
0xA0	0x0F		0x68	12 Bytes	
参数说明	Identifier	12 字节的读写器识别字符。			

## 2.1.20 cmd\_set\_rf\_link\_profile

上位机指令数据包：

Head	Len	Address	Cmd	ProfileID	Check
0xA0	0x04		0x69		
参数说明	ProfileID	0xD0	Profile 0 : Tari 25uS , FM0 40KHz。		
		0xD1	Profile 1 : Tari 25uS , Miller 4 250KHz。 此设置为推荐设置，并为系统默认设置。		
		0xD2	Profile 2 : Tari 25uS , Miller 4 300KHz。		
		0xD3	Profile 3 : Tari 6.25uS , FM0 400KHz。		

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x69	CommandSucces ss	



**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x69		
参数说明	ErrorCode	错误代码			

操作成功后读写器会重新启动，配置保存在内部的 Flash 中，断电后不丢失。

## 2.1.21 cmd\_get\_rf\_link\_profile

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x6A	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ProfileID	Check
0xA0	0x04		0x6A		
参数说明	ProfileID	0xD0	Profile 0 : Tari 25uS , FM0 40KHz。		
		0xD1	Profile 1 : Tari 25uS , Miller 4 250KHz。 此设置为推荐设置 , 并为系统默认设置。		
		0xD2	Profile 2 : Tari 25uS , Miller 4 300KHz。		
		0xD3	Profile 3 : Tari 6.25uS , FM0 400KHz。		

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x69		
参数说明	ErrorCode	错误代码			

## 2.1.22 cmd\_get\_rf\_port\_return\_loss

上位机指令数据包：

Head	Len	Address	Cmd	FreqParameter	Check
0xA0	0x04		0x7E		
参数说明	FreqParameter		频率参数参见频率参数对应表。 系统将获取此频点当前工作天线端口的回波损耗值。		

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ReturnLoss	Check
0xA0	0x04		0x7E		
参数说明	ReturnLoss	回波损耗值，单位是 dB。VSWR = $(10^{RL/20} + 1) / (10^{RL/20} - 1)$ 。			

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x7E		
参数说明	ErrorCode	错误代码			

## 2.2 18000-6C 标签操作命令

### 2.2.1 cmd\_inventory

上位机指令数据包：

Head	Len	Address	Cmd	Repeat	Check
0xA0	0x04		0x80		
参数说明	Repeat	盘存过程重复的次数。  Repeat = 0xFF 则此轮盘存时间为最短时间。如果射频区域内只有一张标签，则此轮的盘存约耗时为 30-50mS。一般在四通道机器上快速轮询多个天线时使用此参数值。			

读写器收到此命令后，进行多标签识别操作。标签数据存入读写器缓存区。

**注意：**

★将参数设置成 255 ( 0xFF ) 时，将启动专为读少量标签设计的算法。对于少量标的应用来说，效率更高，反应更灵敏，但此参数不合同时读取大量标签的应用。

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	TagCount	ReadRate	TotalRead	Check
0xA0	0x0C		0x80		2 Bytes	2Bytes	4Bytes	
参数说明	AntID		此次盘存使用的天线号。					
	TagCount		识别标签的总数量，根据 EPC 号来区分标签，相同 EPC 号的标签将被视为同一张标签。若未清空缓存，标签数量为多次盘存操作的数量累加。					
	ReadRate		此次执行命令的标签识别速度(成功读取标签的次数/秒)。不区分是否多次读取同一张标签。					
	TotalRead		此次执行命令的标签的总读取标签次数，不区分是否多次读取同一张标签。					

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x80		
参数说明	ErrorCode	错误代码			

## 2.2.2 cmd\_read

上位机指令数据包：

Head	Len	Address	Cmd	MemBank	WordAdd	WordCnt	PassWord	Check
0xA0	0x0A		0x81				4 Bytes	
参数说明	MemBank  (标签存储区域)				0x00	RESERVED		
					0x01	EPC		
					0X02	TID		
					0X03	USER		
	WordAdd (读取数据首地址)				取值范围请参考标签规格。			
	WordCnt  (读取数据长度)				字长，WORD(16 bits)长度。  取值范围请参考标签规格书。			
	PassWord				标签访问密码，4 字节。			

◆操作成功：

读写器返回数据包：此数据可能返回多条。数量等于读取的标签数量(无重复数据)。

Head	Len	Address	Cmd	TagCount	DataLen	Data	ReadLen	AntID	ReadCount	Check
0xA0			0x81	2 Bytes		N Bytes				



参数说明	TagCount	成功操作的标签总数。16 bits。
	DataLen	所操作标签的有效数据长度。( PC+CRC+EPC+读取的标签数据 )。 单位是字节。
	Data	所操作标签的有效数据。 PC (2 字节) + EPC (根据标签规格) + CRC (2 字节) + 读取的数据。 (PC(2 字节) + EPC + CRC(2 字节) 即 EPC 存储区域中的全部内容。)
	ReadLen	Read 操作的数据长度。单位是字节。
	AntID	高 6 位是第一次读取的频点参数，低 2 位是天线号。
	ReadCount	该标签被成功操作的次数。

**注意：**

★相同 EPC 的标签，若读取的数据不相同，则被视为不同的标签。

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x81		
参数说明	ErrorCode	错误代码			

## 2.2.3 cmd\_write

上位机指令数据包：

Head	Len	Address	Cm d	PassWor d	MemBan k	WordAd d	WordCnt	Data	Check
0xA0			0x82	4 Bytes				WordCnt  *2	
参数说明	PassWord					标签访问密码，4 字节。			
	MenBank  (标签存储区域)					0x00		RESERVED	
						0x01		EPC	
						0x02		TID	
						0x03		USER	
	WordAdd  (数据首地址)					WORD(16 bits)地址。  写入 EPC 存储区域一般从 02 开始，该区域前四个字节存放 PC+CRC。			
	WordCnt (写入的字长度)					WORD(16 bits)长度，数值请参考标签规格。			

◆操作成功：

读写器返回数据包：此数据可能返回多条。数量等于写入的标签数量(无重复数据)。

Head	Len	Address	Cmd	TagCount	DataLen	Data	ErrCode	AntID	WriteCount	Check
0xA0			0x82	2 Bytes		N Bytes				

参数说明	TagCount	成功操作的标签总数。16 bits。
	DataLen	所操作标签的有效数据长度。( PC+CRC+EPC )。 单位是字节。
	Data	所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格) + CRC (2 字节) ( PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。)
	ErrCode	所操作标签的操作结果，即错误代码。
	AntID	高 6 位是第一次读取的频点参数，低 2 位是天线号。
	WriteCount	该标签被操作的次数。

### ◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x82		
参数说明	ErrorCode	错误代码			

## 2.2.4 cmd\_lock

上位机指令数据包：

Head	Len	Address	Cmd	PassWord	Menbank	LockType	Check
0xA0	0x09		0x83	4 Bytes			
参数说明	PassWord			标签访问密码，4 字节。			
	Menbank (操作的数据区域)			0x01	User Memory		
				0x02	TID Memory		
				0x03	EPC Memory		
				0x04	Access Password		
				0x05	Kill Password		
	LockType (锁操作类型)			0x00	开放		
				0x01	锁定		
				0x02	永久开放		
				0x03	永久锁定		

◆操作成功：

读写器返回数据包：此数据可能返回多条。数量等于锁定的标签数量(无重复数据)。

Head	Len	Address	Cmd	TagCount	DataLen	Data	ErrCode	AntID	LockCount	Check
0xA0			0x83	2 Bytes		N Bytes				

参数说明	TagCount	成功操作的标签总数。16 bits。
	DataLen	所操作标签的有效数据长度。( PC+CRC+EPC )。 单位是字节。
	Data	所操作标签有效数据。 PC(2 字节) + EPC (根据标签规格 + CRC (2 字节)) ( PC (2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。)
	ErrCode	所操作标签的操作结果，即错误代码。
	AntID	高 6 位是第一次读取的频点参数，低 2 位是天线号。
	LockCount	该标签被操作的次数。

### ◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x83		
参数说明	ErrorCode	错误代码			

## 2.2.5 cmd\_kill

上位机指令数据包：

Head	Len	Address	Cmd	PassWord	Check
0xA0	0x07		0x84	4 Bytes	
参数说明	PassWord		标签销毁密码		

◆操作成功：

读写器返回数据包：

此数据可能返回多条。数量等于销毁的标签数量(无重复数据)。

Head	Len	Address	Cmd	TagCount	DataLen	Data	ErrCode	AntID	KillCount	Check
0xA0			0x84	2 Bytes		N Bytes				
参数说明	TagCount		成功操作的标签总数。16 bits。							
	DataLen		所操作标签的有效数据长度。( PC+CRC+EPC )。 单位是字节。							

	Data	所操作标签有效数据。  PC(2 字节) + EPC (根据标签规格 + CRC (2 字节))  ( PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。)
	ErrCode	所操作标签的操作结果，即错误代码。
	AntID	高 6 位是第一次读取的频点参数，低 2 位是天线号。
	KillCount	销毁标签操作只能为 1。

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x84		
参数说明	ErrorCode	错误代码			

## 2.2.6 cmd\_set\_access\_epc\_match

上位机指令数据包：

Head	Len	Address	Cmd	Mode	EpcLen	Epc	Check
0xA0			0x85			EpcLen 个字节	
参数说明	Mode		0x00	EPC 匹配一直有效，直到下一次刷新。			
			0x01	清除 EPC 匹配。			
	EpcLen		EPC 长度。				
	Epc		EPC 号，由 EpcLen 个字节组成。				

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x85		
参数说明	ErrorCod e	错误代码			



## 2.2.7 cmd\_get\_access\_epc\_match

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0			0x86	

读写器返回数据包：

Head	Len	Address	Cmd	Status	EpcLen	EPC	Check
0xA0			0x86				
参数说明	Status	0x00			有匹配		
		0x01			无匹配		
	EpcLen	匹配的 EPC 号长度，无匹配时不返回此数据。					
	EPC	匹配的 EPC 号，无匹配时不返回此数据。					

## 2.2.8 cmd\_real\_time\_inventory

上位机指令数据包：

Head	Len	Address	Cmd	Repeat	Check
0xA0	0x04		0x89		
参数说明	Repeat	盘存过程重复的次数。  Repeat = 0xFF 则此轮盘存时间为最短时间。如果 射频区域内只有一张标签，则此轮的盘存约耗时为 30-50mS。一般在四通道机器上快速轮询多个天线 时使用此参数值。			

读写器收到此命令后，进行 EPC GEN2 多标签识别操作。标签数据不存入读写器缓存区，而是实时上传标签的数据。

**注意：**

★由于硬件为双 CPU 架构，主 CPU 负责轮询标签，副 CPU 负责数据管理。轮询标签和发送数据并行，互不占用对方的时间，因此串口的数据传输不影响读写器工作的效率。

如有标签应答，返回如下数据包(多条)：

Head	Len	Address	Cmd	FreqAnt	PC	EPC	RSSI	Check
0xA0			0x89		2 bytes	N bytes		

参数说明	FreqAnt	此字节高 6 位是读取标签的频点参数，低 2 位是天线号。
	PC	标签的 PC，固定两个字节
	EPC	标签的 EPC 号，长度可变化。
	RSSI	标签的实时 RSSI。

命令完成，读写器返回如下数据包：

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	ReadRate	TotalRead	Check
0xA0	0x0A		0x89		2bytes	4 bytes	
参数说明	AntID	此次盘存使用的天线号。					
	ReadRate	此轮命令标签识别速率。					
	TotalRead	标签应答的总记录数。					

◆操作失败：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x89		
参数说明	ErrorCode	错误代码			

## 2.2.9 cmd\_fast\_switch\_ant\_inventory

上位机指令数据包：

Head	Len	Address	Cmd	A	Stay	B	Stay	C	Stay	D	Stay	Interval	Repeat	Check
0xA0	0x0D		0x8A											
参数说明	A		首先轮询的天线 (00 – 03) ，天线号大于三则表示不轮询。											
	Stay		天线重复轮询的次数。每个天线可单独配置。											
	B		第二个轮询的天线 (00 – 03) ，天线号大于三则表示不轮询。											
	C		第三个轮询的天线 (00 – 03) ，天线号大于三则表示不轮询。											
	D		第四个轮询的天线 (00 – 03) ，天线号大于三则表示不轮询。											
	Interval		天线间的休息时间。单位是 mS。休息时无射频输出，可降低功耗。											
	Repeat		重复以上天线切换顺序次数。											

读写器收到此命令后，进行多标签识别操作。标签数据不存入读写器缓存区，而是实时上传标签的数据。同时，会依次按照 A->D 的顺序自动切换天线。如果在射频区域内没有标签，或者只有一两张标签在射频区域内，则每个天线平均耗时 30mS 左右。如果标签数量比较多，则耗时时间会相应增加。此命令非常适合需要高速切换多个天线识别标签的应用。

### 注意：

- ★由于硬件为双 CPU 架构，主 CPU 负责轮询标签，副 CPU 负责数据管理。轮询标签和发送数据并行，互不占用对方的时间，因此串口的数据传输不影响读写器工作的效率。
- ★此命令读取大量标签时，效率没有 cmd\_real\_time\_inventory 命令高。

如有标签应答，返回如下数据包(多条)：

Head	Len	Address	Cmd	FreqAnt	PC	EPC	RSSI	Check
0xA0			0x8A		2 bytes	N bytes		
参数说明		FreqAnt		此字节高 6 位是读取标签的频点参数，低 2 位是天线号。				
		PC		标签的 PC，固定两个字节。				
		EPC		标签的 EPC 号，长度可变化。				
		RSSI		标签的实时 RSSI。				

如果打开了天线连接检测，并检测到端口未连接天线则返回：

Head	Len	Address	Cmd	AntID	ErrorCode	Check
0xA0	0x05		0x8A		0x22	

参数说明	AntID	未连接的天线号(00 – 03)。
	ErrorCode	0x22，天线未连接错误。

命令完成，读写器返回如下数据包：

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	TotalRead	CommandDuration	Check
0xA0	0x0A		0x8A	3 bytes	4 bytes	
参数说明	TotalRead			总共上传的标签数据记录数，3 字节，高位在前。		
	CommandDuration			命令总共消耗的时间，单位是毫秒，4 字节，高位在前。		

◆操作失败：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x8A		
参数说明	ErrorCode	错误代码			

## 2.2.10 cmd\_customized\_session\_target\_inventory

上位机指令数据包：

Head	Len	Address	Cmd	Session	Target	Repeat	Check
0xA0	0x06		0x8B				
参数说明	Session	指定盘存的 session。					
	Target	指定盘存的 Inventoried Flag , 00 为 A , 01 为 B。					
	Repeat	盘存过程重复的次数。					

读写器收到此命令后，按照指定的 session 和 inventoried flag 进行 EPC GEN2 多标签识别操作。标签数据不存入读写器缓存区，而是实时上传标签的数据。

**注意：**

★由于硬件为双 CPU 架构，主 CPU 负责轮询标签，副 CPU 负责数据管理。轮询标签和发送数据并行，互不占用对方的时间，因此串口的数据传输不影响读写器工作的效率。

如有标签应答，返回如下数据包(多条)：

Head	Len	Address	Cmd	FreqAnt	PC	EPC	RSSI	Check
		s						
0xA0			0x8B		2 bytes	N bytes		

参数说明	FreqAnt	此字节高 6 位是读取标签的频点参数，低 2 位是天线号。
	PC	标签的 PC，固定两个字节
	EPC	标签的 EPC 号，长度可变化。
	RSSI	标签的实时 RSSI。

命令完成，读写器返回如下数据包：

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	ReadRate	TotalRead	Check
0xA0	0x0A		0x8B		2bytes	4 bytes	
参数说明	AntID	此次盘存使用的天线号。					
	ReadRate	此轮命令标签识别速率。					
	TotalRead	标签应答的总记录数。					

◆操作失败：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x8B		
参数说明	ErrorCode	错误代码			



## 2.2.11 cmd\_set\_impinj\_fast\_tid

上位机指令数据包：

Head	Len	Address	Cmd	FastTID	Check
0xA0	0x04		0x8C		
参数说明	FastTID	除 0x8D 外的其他值		关闭 FastTID	
		0x8D		打开 FastTID	

注意：

- ★此功能仅对 Impinj Monza 标签的部分型号有效。
- ★此功能在识别 EPC 的同时识别 TID，因此大大提高了读 TID 的效率。
- ★打开此功能后，特定型号的标签会在盘存的过程中将 TID 打包到 EPC 中。因此，标签的 PC 会被修改，原来的 PC+EPC 变为：修改后的 PC + EPC + (EPC 的 CRC) + TID。
- ★如果在识别 TID 的过程中出现错误，则上传原来的 PC+EPC。
- ★如不需要此功能请将其关闭，避免不必要的时间消耗。
- ★此命令不保存至内部的 Flash 中，重启后将回复至 Flash 保存的值。

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x8C	CommandSucces ss	

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x8C		
参数说明	ErrorCode	错误代码			

## 2.2.12 cmd\_set\_and\_save\_impinj\_fast\_tid

请参考 cmd\_set\_impinj\_fast\_tid 命令。

此命令将配置保存至内部的 Flash 中，断电不丢失。

## 2.2.13 cmd\_get\_impinj\_fast\_tid

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x8E	

读写器返回数据包：

Head	Len	Address	Cmd	FastTID	Check
0xA0	0x04		0x8E		
参数说明	FastTID	0x8D	FastTID 打开		
		0x00	FastTID 关闭		

## 2.3 ISO 18000-6B 标签操作命令

### 2.3.1 cmd\_iso18000\_6b\_inventory

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0xB0	

读写器收到此命令后，进行 ISO 18000 -6B 多标签识别操作。标签数据不存入读写器缓存区。

如有标签应答，返回如下数据包(多条)：

Head	Len	Address	Cmd	AntID	UID	Check
0xA0	0x0C		0xB0		8 bytes	
参数说明	AntID		工作天线号。			
	UID		ISO 18000-6B 标签的 8 字节 UID。			

命令完成，读写器返回如下数据包：

## ◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	TagFound	Check
0xA0	0x05		0xB0			
参数说明	AntID	此次盘存使用的天线号。				
	TagFound	盘存到的标签数量。				

## ◆操作失败：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0xB0		
参数说明	ErrorCode	错误代码			

## 2.3.2 cmd\_iso18000\_6b\_read

上位机指令数据包：

Head	Len	Address	Cmd	UID	StartAddress	Length	Check
0xA0	0x0D		0xB1	8 bytes			
参数说明	UID		被操作标签的 UID。				
	StartAddress		要读取的数据首地址。				
	Length		要读取的数据长度。				

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	Data	Check
0xA0			0xB1		N bytes	
参数说明	AntID		此次读操作使用的天线号。			
	Data		读出的数据。			

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0xB1		
参数说明	ErrorCode	错误代码			

### 2.3.3 cmd\_iso18000\_6b\_write

上位机指令数据包：

Head	Len	Address	Cmd	UID	StartAddress	Length	Data	Check
0xA0			0xB2				N bytes	
参数说明		UID		被操作标签的 UID。				
		StartAddress		写入数据的首地址。				
		Length		写入数据的长度。				
		Data		写入的数据。				

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	WrittenCount	Check
0xA0	0x05		0xB2			
参数说明	AntID		此次写操作使用的天线号。			
	WrittenCount		成功写入的字节数。			



注意：

★可以一次性写入多个字节。一旦写入某个字节出现错误，此命令不会继续写入后面的数据。同时命令返回已经成功写入的字节数。

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0xB2		
参数说明	ErrorCode	错误代码			

## 2.3.4 cmd\_iso18000\_6b\_lock

上位机指令数据包：

Head	Len	Address	Cmd	UID	LockAddress	Check
0xA0	0x0C		0xB3	8 bytes		
参数说明	UID		被操作标签的 UID。			
	LockAddress		被锁定的地址。			

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	Status	Check
0xA0	0x05		0xB3			
参数说明	AntID		此次盘存使用的天线号。			
	Status	0x00			该字节成功锁定。	
		0xFE			该字节已是锁定状态。	
		0xFF			该字节无法锁定。	

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0xB3		
参数说明	ErrorCode	错误代码			

## 2.3.5 cmd\_iso18000\_6b\_query\_lock

上位机指令数据包：

Head	Len	Address	Cmd	UID	QueryAddress	Check
0xA0	0x0C		0xB4	8 bytes		
参数说明	UID		被操作标签的 UID。			
	QueryAddress		要查询的地址。			

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	AntID	Status	Check
0xA0	0x05		0xB4			
参数说明	AntID		此次盘存使用的天线号。			
	Status	0x00			该字节未锁定。	
		0xFE			该字节已是锁定状态。	

◆操作失败：

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0xB4		
参数说明	ErrorCode	错误代码			

## 2.4 缓存操作命令

### 2.4.1 cmd\_get\_inventory\_buffer

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x90	

◆操作成功：

读写器返回数据包：

此数据可能返回多条。数量等于缓存中的标签数量(无重复数据)。

Head	Len	Address	Cmd	TagCount	DataLen	Data	RSSI	FREQ	FreqAnt	InvCount	Check
0xA0			0x90	2 Bytes		N Bytes					
参数说明	TagCount		成功操作的标签总数。16 bits。								
	DataLen		所操作标签的有效数据长度。( PC+CRC+EPC )。 单位是字节。								

	Data	所操作标签有效数据。  PC(2 字节) + EPC (根据标签规格 + CRC (2 字节))  ( PC(2 字节) + EPC + CRC (2 字节) 即 EPC 存储区域中的全部内容。)
	RSSI	第一次读到该标签时的信号强度。此参数对应的 dBm 值请参见 RSSI 参数对应表。
	FreqAnt	高 6 位是第一次读取的频点参数，低 2 位是天线号。
	InvCount	该标签成功读取的次数，如果该值为 0xFF，则说明成功读取次数 >= 255 次。

**注意：**

★命令完成后，缓存中的数据并不丢失，可以多次提取。

★若再次运行 cmd\_inventory 命令，则盘存到的标签将累计存入缓存。

★若再次运行其他的 18000-6C 命令，缓存中的数据将被清空。

**◆操作失败：**

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x90		
参数说明	ErrorCode	错误代码			

## 2.4.2 cmd\_get\_and\_reset\_inventory\_buffer

数据格式请参考 cmd\_get\_inventory\_buffer 命令。

命令成功完成后，缓存中的数据将被全部清空。

## 2.4.3 cmd\_get\_inventory\_buffer\_tag\_count

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x92	

◆操作成功：

读写器返回数据包：

Head	Len	Address	Cmd	TagCount	Check
0xA0	0x05		0x92	2 Bytes	
参数说明	TagCount	缓存中标签数据数量。无重复数据。			
	t				



## 2.4.4 cmd\_reset\_inventory\_buffer

上位机指令数据包：

Head	Len	Address	Cmd	Check
0xA0	0x03		0x93	

读写器返回数据包：

Head	Len	Address	Cmd	ErrorCode	Check
0xA0	0x04		0x93	CommandSucces ss	

### 3 错误代码表

序号	值	名 称	描 叙
1	0x10	command_success	命令成功完成
2	0x11	command_fail	命令执行失败
3	0x20	mcu_reset_error	CPU 复位错误
4	0x21	cw_on_error	打开 CW 错误
5	0x22	antenna_missing_error	天线未连接
6	0x23	write_flash_error	写 Flash 错误
7	0x24	read_flash_error	读 Flash 错误
8	0x25	set_output_power_error	设置发射功率错误
9	0x31	tag_inventory_error	盘存标签错误
10	0x32	tag_read_error	读标签错误
11	0x33	tag_write_error	写标签错误
12	0x34	tag_lock_error	锁定标签错误
13	0x35	tag_kill_error	灭活标签错误
14	0x36	no_tag_error	无可操作标签错误
15	0x37	inventory_ok_but_access_fail	成功盘存但访问失败
16	0x38	buffer_is_empty_error	缓存为空
17	0x40	access_or_password_error	访问标签错误或访问密码错误

18	0x41	parameter_invalid	无效的参数
19	0x42	parameter_invalid_wordCnt_too_long	wordCnt 参数超过规定长度
20	0x43	parameter_invalid_membank_out_of_range	MemBank 参数超出范围
21	0x44	parameter_invalid_lock_region_out_of_range	Lock 数据区参数超出范围
22	0x45	parameter_invalid_lock_action_out_of_range	LockType 参数超出范围
23	0x46	parameter_reader_address_invalid	读写器地址无效
24	0x47	parameter_invalid_antenna_id_out_of_range	Antenna_id 超出范围
25	0x48	parameter_invalid_output_power_out_of_range	输出功率参数超出范围
26	0x49	parameter_invalid_frequency_region_out_of_range	射频规范区域参数超出范围
27	0x4A	parameter_invalid_baudrate_out_of_range	波特率参数超出范围
28	0x4B	parameter_beeper_mode_out_of_range	蜂鸣器设置参数超出范围
29	0x4C	parameter_epc_match_len_too_long	EPC 匹配长度越界
30	0x4D	parameter_epc_match_len_error	EPC 匹配长度错误
31	0x4E	parameter_invalid_epc_match_mode	EPC 匹配参数超出范围
32	0x4F	parameter_invalid_frequency_range	频率范围设置参数错误
33	0x50	fail_to_get_RN16_from_tag	无法接收标签的 RN16
34	0x51	parameter_invalid_drm_mode	DRM 设置参数错误
35	0x52	pll_lock_fail	PLL 不能锁定
36	0x53	rf_chip_fail_to_response	射频芯片无响应

37	0x54	fail_to_achieve_desired_output_power	输出达不到指定的输出功率
38	0x55	copyright_authentication_fail	版权认证未通过
39	0x56	spectrum_regulation_error	频谱规范设置错误
40	0x57	output_power_too_low	输出功率过低

#### 4 频率参数对应表

频率参数	对应频点	频率参数	对应频点
0(0x00)	865.00 MHz	30(0x1E)	913.50 MHz
1(0x01)	865.50 MHz	31(0x1F)	914.00 MHz
2(0x02)	866.00 MHz	32(0x20)	914.50 MHz
3(0x03)	866.50 MHz	33(0x21)	915.00 MHz
4(0x04)	867.00 MHz	34(0x22)	915.50 MHz
5(0x05)	867.50 MHz	35(0x23)	916.00 MHz
6(0x06)	868.00 MHz	36(0x24)	916.50 MHz
7(0x07)	902.00 MHz	37(0x25)	917.00 MHz
8(0x08)	902.50 MHz	38(0x26)	917.50 MHz
9(0x09)	903.00 MHz	39(0x27)	918.00 MHz
10(0x0A)	903.50 MHz	40(0x28)	918.50 MHz
11(0x0B)	904.00 MHz	41(0x29)	919.00 MHz
12(0x0C)	904.50 MHz	42(0x2A)	919.50 MHz
13(0x0D)	905.00 MHz	43(0x2B)	920.00 MHz
14(0x0E)	905.50 MHz	44(0x2C)	920.50 MHz
15(0x0F)	906.00 MHz	45(0x2D)	921.00 MHz
16(0x10)	906.50 MHz	46(0x2E)	921.50 MHz

17(0x11)	907.00 MHz	47(0x2F)	922.00 MHz
18(0x12)	907.50 MHz	48(0x30)	922.50 MHz
19(0x13)	908.00 MHz	49(0x31)	923.00 MHz
20(0x14)	908.50 MHz	50(0x32)	923.50 MHz
21(0x15)	909.00 MHz	51(0x33)	924.00 MHz
22(0x16)	909.50 MHz	52(0x34)	924.50 MHz
23(0x17)	910.00 MHz	53(0x35)	925.00 MHz
24(0x18)	910.50 MHz	54(0x36)	925.50 MHz
25(0x19)	911.00 MHz	55(0x37)	926.00 MHz
26(0x1A)	911.50 MHz	56(0x38)	926.50 MHz
27(0x1B)	912.00 MHz	57(0x39)	927.00 MHz
28(0x1C)	912.50 MHz	58(0x3A)	927.50 MHz
29(0x1D)	913.00 MHz	59(0x3B)	928.00 MHz

## 5 RSSI 参数对应表

RSSI 参数	对应信号强度 ( 对数 )	RSSI 参数	对应信号强度 ( 对数 )
98(0x62)	-31dBm	64(0x40)	-66dBm
97(0x61)	-32dBm	63(0x3F)	-67dBm
96(0x60)	-33dBm	62(0x3E)	-68dBm
95(0x5F)	-34dBm	61(0x3D)	-69dBm
94(0x5E)	-35dBm	60(0x3C)	-70dBm
93(0x5D)	-36dBm	59(0x3B)	-71dBm
92(0x5C)	-37dBm	58(0x3A)	-72dBm
91(0x5B)	-38dBm	57(0x39)	-73dBm
90(0x5A)	-39dBm	56(0x38)	-74dBm
89(0x59)	-41dBm	55(0x37)	-75dBm
88(0x58)	-42dBm	54(0x36)	-76dBm
87(0x57)	-43dBm	53(0x35)	-77dBm
86(0x56)	-44dBm	52(0x34)	-78dBm
85(0x55)	-45dBm	51(0x33)	-79dBm
84(0x54)	-46dBm	50(0x32)	-80dBm
83(0x53)	-47dBm	49(0x31)	-81dBm
82(0x52)	-48dBm	48(0x30)	-82dBm

81(0x51)	-49dBm	47(0x2F)	-83dBm
80(0x50)	-50dBm	46(0x2E)	-84dBm
79(0x4F)	-51dBm	45(0x2D)	-85dBm
78(0x4E)	-52dBm	44(0x2C)	-86dBm
77(0x4D)	-53dBm	43(0x2B)	-87dBm
76(0x4C)	-54dBm	42(0x2A)	-88dBm
75(0x4B)	-55dBm	41(0x29)	-89dBm
74(0x4A)	-56dBm	40(0x28)	-90dBm
73(0x49)	-57dBm	39(0x27)	-91dBm
72(0x48)	-58dBm	38(0x26)	-92dBm
71(0x47)	-59dBm	37(0x25)	-93dBm
70(0x46)	-60dBm	36(0x24)	-94dBm
69(0x45)	-61dBm	35(0x23)	-95dBm
68(0x44)	-62dBm	34(0x22)	-96dBm
67(0x43)	-63dBm	33(0x21)	-97dBm
66(0x42)	-64dBm	32(0x20)	-98dBm
65(0x41)	-55dBm	31(0x1F)	-99dBm



## 6 校验和计算方法(C 语言描述)

```
unsigned char  CheckSum(unsigned char *uBuff, unsigned char uBuffLen)
{
    unsigned char i,uSum=0;
    for(i=0;i<uBuffLen;i++)
    {
        uSum = uSum + uBuff[i];
    }
    uSum = (~uSum) + 1;
    return uSum;
}
```