1. **单次读取卡号：**

Send：BB 00 22 00 00 22 7E

16进制数，一共7个字节；

如果读到卡，模块回复：

BB 02 22 00 11 D5 30 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 EB 98 0C 7E

BB 02 22 :是包识别符，长度3个字节；

00 11 ：是包长度，16进制，0x11表示 17个字节，长度2个字节；

D5 30 00 ：可以不处理；长度3个字节，具体内容可能会变化；

E2 ~ B4 02：卡号，一共12个字节；

EB 98：卡的CRC；

0C: Checksum；具体计数方法见下方写入发送的包的字段解析；

7E；结束符；

如果读不到卡，或者无卡，模块回复：

BB 01 FF 00 01 15 16 7E

一共8个字节；

1. **群读卡号指令：**

Send：BB 00 27 00 03 22 FF FF 4A 7E

BB 00 27 ：帧标志，3个字节；

00 03：数据长度，2个字节；0003表示3个字节；

22：保留字节；

FF FF ： 读取次数，连续读取65535次；如果连续读取100次，填入00 64；

4A：Checksum，00 27 00 03 22 FF FF 每个字节都累加起来，得到0x024A；支取低8位 4A;

7E：结束符

Recv： BB 01 FF 00 01 15 16 7E

BB 01 FF 00 01 15 16 7E

.

.

.

BB 02 22 00 11 C8 34 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 16 3D D3 7E

BB 01 FF 00 01 15 16 7E

BB 02 22 00 11 C9 34 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 16 3D D4 7E

BB 02 22 00 11 C0 34 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 16 3D CB 7E

BB 01 FF 00 01 15 16 7E

发出连读读取帧后，会联续的接收到收到两种类型的包。

BB 01 FF 00 01 15 16 7E

这是其中一种，表示读取失败；

BB 02 22 00 11 C0 34 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 16 3D CB 7E

这是另外一种，表示读到卡号：

具体格式与单次读取卡号的回复包相同；

BB 02 22：帧标志，3个字节；

00 11：数据长度，11 表示16进制，实际为17个字节；

C0：信号强度；RSSI,一个字节；

34 00：PC，2个字节；

E2 00 10 71 00 00 52 9B 09 40 B4 02：卡号，12个字节；

16 3D ：CRC2个字节；

CB：Checksum，02 ~ 16 3D 累加，取低8位；

7E：结束符；

1. **结束群读：**

Send： BB 00 28 00 00 28 7E

由于群读次数多时，操作时间会很长，客户可以发送该指令结束群读指令；

Recv： BB 01 28 00 01 00 2A 7E

模块执行结束群读指令的回复。

1. **Select Set：**

Send：BB 00 0C 00 07 23 00 00 00 00 60 00 96 7E

无掩码模式；

BB 00 0C 00 13 23 00 00 00 00 60 00 E2 00 00 16 55 11 02 06 03 90 EA AF 34 7E

其中 E2 00 00 16 55 11 02 06 03 90 EA AF 是卡的EPC(卡号)掩码，加了以后，就可以不受影响的读写 指定的卡。

Rcve：BB 01 0C 00 01 00 0E 7E

设置成功。

1. **Write写入：**

Send：BB 00 0C 00 07 23 00 00 00 00 60 00 96 7E

BB 00 49 00 11 00 00 00 00 03 00 00 00 04 01 02 03 04 05 06 07 08 85 7E

其实是发送了两包数据，第一包是Select Set；第二包是写入包。Select Set的详细信息请参考上面Select Set的帧解析；下面介绍写入包：

BB 00 49 :是包识别符，长度3个字节；

00 11 ：是包长度，16进制，0x11表示 17个字节，长度2个字节；

00 00 00 00：是访问密码（默认是00 00 00 00），长度4个字节；

03 : 表示选择用户存储区；

00 00 ：表示写入的存储区的地址偏移量，00 00 指从0地址开始写入；

00 04 ：表示写入的数据长度，00 04 表示写入4个字（8个字节）；

01 02 03 04 05 06 07 08 ：是写入的数据；

85：Checksum，计算公式是，Checksum 字节前面的所有字节，除了第一个字节BB外，每个字节的累加，结果只取低8位；

比如：00 49 00 11 00 00 00 00 03 00 00 00 04 01 02 03 04 05 06 07 08 累加的结果是：0x85，所以 Checksum就是85;

7E ：结束字符；

Recv：写入成功会收到：

BB 01 0C 00 01 00 0E 7E

BB 01 49 00 10 0E 34 00 E2 00 00 16 55 11 02 06 03 90 EA AF 00 2E 7E

写入成功会接收到其实是2包，第一包是Select Set 的响应包；第二包是写入的响应包：

BB 01 49 :是包识别符,表示写入成功，长度3个字节；

00 10 ：是包长度，16进制，0x10表示 16个字节，长度2个字节；

0E ：PC+卡号的长度，16进制，0x0E表示 14个字节，长度1个字节；

34 00： 是PC值，这里不作解析，可以不处理；

E2 00 ~ EA AF ：是成功写入的卡号，一共12个字节；

00 ： 表示操作成功；

2E : Checksum；

7E：结束符；

写入失败会接收到：

BB 01 0C 00 01 00 0E 7E

BB 01 FF 00 10 10 0E 34 00 E2 00 00 16 55 11 02 06 03 90 EA AF F4 7E

写入失败接收到其实是2包，第一包是Select Set 的响应包；第二包是写入的响应包：

BB 01 FF :是包识别符,表示出错，长度3个字节；

00 10 ：是包长度，16进制，0x10表示 16个字节，长度2个字节；

10： 错误码： 0x10表示没找到卡；

0x16表示 访问密码错误；

0xB3表示超出读写范围；

其他字段可以不作考虑，这里就不一一解析了；

1. **Read读卡内容 ：**

Send ：BB 00 0C 00 07 23 00 00 00 00 60 00 96 7E

BB 00 39 00 09 00 00 00 00 03 00 00 00 04 49 7E

其实是发送了两包数据，第一包是Select Set；第二包是写入包。Select Set的详细信息请参考上面Select Set的帧解析；下面介绍读取包：

BB 00 39 :是包识别符，长度3个字节；

00 09 ：是包长度，16进制，0x09表示 9个字节，长度2个字节；

00 00 00 00：是访问密码（默认是00 00 00 00），长度4个字节；

03 : 表示选择用户存储区；

00 00 ：表示读取的存储区的地址偏移量，00 00 指从0地址开始写入；

00 04 ：表示去读的数据长度，00 04 表示写入4个字（8个字节）；

49：Checksum，计算公式是，Checksum字节前面的所有字节，除了第一个字节BB外，每个字节的累加，结果只取低8位；

比如：00 39 00 09 00 00 00 00 03 00 00 00 04 累加的结果是：0x49，所以 Checksum就是0x49;

7E ：结束字符；

Recv：BB 01 0C 00 01 00 0E 7E

BB 01 39 00 17 0E 34 00 E2 00 00 16 55 11 02 06 03 90 EA AF 01 02 03 04 05 06 07 08 49 7E

读取成功会接收到其实是2包，第一包是Select Set 的响应包；第二包是读取成功的响应包：

BB 01 39 :是包识别符,表示读取成功，长度3个字节；

00 17 ：是包长度，16进制，0x17表示 23个字节，长度2个字节；

0E ：PC+卡号的长度，16进制，0x0E表示 14个字节，长度1个字节；

34 00： 是PC值，这里不作解析，可以不处理；

E2 00 ~ EA AF ：是成功写入的卡号，一共12个字节；

01 02 03 04 05 06 07 08 ： 是读取的具体数据，一共8个字节。

49: Checksum；

7E：结束符；

读取失败会收到：

BB 01 0C 00 01 00 0E 7E

BB 01 FF 00 10 09 0E 34 00 E2 00 10 71 00 00 52 9B 09 40 B4 02 AA 7E

读取失败接收到其实是2包，第一包是Select Set 的响应包；第二包是读取失败的响应包：

BB 01 FF :是包识别符,表示出错，长度3个字节；

00 10 ：是包长度，16进制，0x10表示 16个字节，长度2个字节；

09： 错误码： 0x09表示没找到卡；

0x16表示 访问密码错误；

0xA3表示超出读写范围；

其他字段可以不作考虑，这里就不一一解析了；

1. **设置发射功率：**

Send：

BB 00 B6 00 02 03 E8 A3 7E 10 dBm

BB 00 B6 00 02 04 4C 08 7E 11 dBm

BB 00 B6 00 02 04 B0 6C 7E 12 dBm

BB 00 B6 00 02 05 14 D1 7E 13 dBm

BB 00 B6 00 02 05 78 35 7E 14 dBm

BB 00 B6 00 02 05 DC 99 7E 15 dBm

BB 00 B6 00 02 06 40 FE 7E 16 dBm

BB 00 B6 00 02 06 A4 62 7E 17 dBm

BB 00 B6 00 02 07 08 C7 7E 18 dBm

BB 00 B6 00 02 07 6C 2B 7E 19 dBm

BB 00 B6 00 02 07 D0 8F 7E 20 dBm

BB 00 B6 00 02 08 34 F4 7E 21 dBm

BB 00 B6 00 02 08 98 58 7E 22 dBm

BB 00 B6 00 02 08 FC BC 7E 23 dBm

BB 00 B6 00 02 09 60 21 7E 24 dBm

BB 00 B6 00 02 09 C4 85 7E 25 dBm

BB 00 B6 00 02 0A 28 EA 7E 26 dBm

以上的数据包，选择发送一条；

Recv：

BB 01 B6 00 01 00 B8 7E

发送设置任何发射功率，设置成功后，均回复该数据包。

1. **设置工作区：**

Send：

BB 00 07 00 01 01 09 7E ；设置China2区 （920~925MHz）；

BB 00 07 00 01 04 0C 7E ；设置China1区 （840~845MHz）；

BB 00 07 00 01 02 0A 7E ；设置US区 （902.25~927.75MHz）；

BB 00 07 00 01 03 0B 7E ；设置Europe区 （865~868MHz）；

BB 00 07 00 01 06 0E 7E ；设置Korea区 （917~923MHz）；

以上的数据包，选择发送一条；

Recv：

BB 01 07 00 01 00 09 7E

模块回复，设置工作区成功。

**9. 设置RF信道**

Send:

BB 00 AB 00 01 00 AC 7E

BB 00 AB :是包识别符,表示出错，长度3个字节；

00 10 ：是包长度，16进制，0x10表示 16个字节，长度2个字节；

00: 信道号，00表示0号信道，长度为1个字节；

China2区 （920.125~924.875MHz）信道参数(0~0x13)，共20个信道，间隔0.25M。

China1区 （840.125~844.875MHz）信道参数(0~0x13)，共20个信道，间隔0.25M。

US区（902.25~927.75MHz）信道参数(0~0x33)，共54个信道，间隔0.5M。

Europe区（865.1~867.9MHz）信道参数(0~0x0E)，共15个信道，间隔0.2M。

Korea区（917.1~923.3MHz）信道参数(0~0x1F)，共32个信道，间隔0.2M。

AC：Checksum，具体计算公式请看上文；

7E：结束符；

Recv：

BB 01 AB 00 01 00 AD 7E

模块回复，设置信道成功。