1.编码总接口rrc\_encode()定义于rrc\_encode.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int RRC\_encode(RRC\_Sdecode\_Result\_4\_L2 \*pinpdu,

RRC\_Encode\_Result\_t \*result)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 参数1：pinpdu RRC解码结果结构体指针，包含信道信息及需要编码的PDU的结构体

/\* 参数2：result RRC编码结果结构体指针，包含一个指向编码数据的指针及编码数据的

/\* 字节数

/\* 返回值：RRC编码成功标志，0：编成功，-1：失败

/\* 功能说明：该函数是RRC编码的总的调用接口，应用程序调用该函数对PDU消息进行编

/\* 码，返回编码成功标志。

/\* 使用方法及注意事项：调用该接口前，需要定义RRC\_Encode\_Result\_t 类型变量及申请一

/\* 块编码数据存储空间并初始化，该空间大小需指定；需要提供RRC\_Sdecode\_Result\_4\_L2

/\* 解码结构体变量填充需要编码的PDU消息，示例如下：

/\* ......

int rv = -1; //编码返回值变量

RRC\_Encode\_Result\_t enresult; //编码结果结构体

uint8\_t \*enbuf; //编码数据存储buffer

enbuf = malloc(ENCODE\_MAX\_SIZE); // 申请空间，ENCODE\_MAX\_SIZE需要指定

if(enbuf == NULL)

{

//malloc error handle;

}

memset(enbuf,0, ENCODE\_MAX\_SIZE); //初始化

memset(&enresult,0,sizeof(RRC\_Encode\_Result\_t)); //初始化编码结果结构体变量

enresult.data = enbuf; //指向编码buffer

rv = rrc\_encode(pderesult, &enresult); //解码接口调用，pderesult为解码结果结构体指针

if(rv < 0 )

{

//encode error handle;

}

/\* ......

2.编码接口调用函数：RRC\_sencode()定义于：asn1c/rrc\_sencode4L2.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int CYDD\_EXPORT RRC\_sencode(RRC\_Sdecode\_Result\_4\_L2 \*input,

void \*output,

size\_t \*size)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 参数1：RRC解码结果结构体指针，包含信道信息及需要编码的PDU的结构体

/\* 参数2：编码数据存储buffer指针

/\* 参数3：编码数据字节数变量指针

/\* 返回值：0：成功，-1：失败，-2：输入消息类型错误

/\* 功能说明：该函数实现对不同PDU消息进行编码，返回编码成功标志，其内部调用ASN1C

/\* 工具生成的编码接口实现编码

/\* 使用说明：调用该函数时，编码数据存储buffer指针不能为空，需提前申请足够的空间

3.ASN1C工具生成编码接口：uper\_encode\_to\_new\_buffer()定义于：asn1c/per\_encoder.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ssize\_t

uper\_encode\_to\_new\_buffer(asn\_TYPE\_descirptor\_t \*td,

asn\_per\_constraints\_t \*constraints,

void \*sptr,

void \*\*buffer\_r)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 参数1：PDU消息描述变量指针，定义与各自的消息文件中

/\* 参数2：设为0

/\* 参数3：需要编码的PDU结构体指针

/\* 参数4：编码数据存放的二级指针

/\* 返回值：编码数据的字节数，小于0：编码失败

/\* 功能说明：该函数实现对指定PDU类型的消息进行编码，返回编码数据的字节数

/\* 调用说明：调用该函前，需指定PDU消息的类型描述变量并作为参数传入，该变量定义

/\* 于各自的消息文件中；该函数会申请动态内存，调用完该函数且无错误后应及时取走数

/\* 据并调用free()函数释放掉该部分内存，防止内存泄漏。