1.解码总调用接口rrc\_decode 定义于：rrc\_decode.c：

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

signed int rrc\_decode(const void\* proinfoP,

const void\* sdataP,

const int32\_t sdlen,

void\* const resultP)

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

/\* 参数1：proinfoP数据的信道及载波方向等信息

/\* 参数2：sdataP PDU原始数据指针

/\* 参数3：sdlen PDU原始数据长度

/\* 参数4：resultP解码结果指针

/\* 返回值：解码成功标志，0：成功，-1：解码失败，-2：无对应的传输信道信或上下行方，

/\* -3：无对应逻辑信道

/\* 功能说明：该函为RRC解码调用的总接口，实现对RRC PDU数据的解码，并返回解码成

/\* 功标志

/\* 调用说明：调用该函数，需要提供原始数据的指针及长度、信道信息及上下行方，示例

/\* 如下：

/\* ......

ProcH\_t \*proinfoP = \*bbdP->bbdheader.ProcH; //信道及载波方向等信息变量

void \*sdataP = bbdP->macpdu; // 需要解码的RRC数据

int32\_t sdlen = bbdP->datalen; // 需要解码的数据长度

RRC\_Sdecode\_Result\_4\_L2 result; //解码结果结构体变量

int rv = -1 ; //解码成功标志变量

rv = rrc\_decode(proinfoP, sdataP, sdlen, &result); //调用解码接口

if(rv != 0)

{

//decode error handle;

}

/\* ......

1.解码接口调用函数：RRC\_sdecode ()定义于：asn1c/rrc\_sdecode4L2.c：

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

int CYDD\_EXPORT

RRC\_sdecode(ProcH\_t\* proch\_infoP,

void \*data,

size\_t size,

RRC\_Sdecode\_Result\_4\_L2 \*output)

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

/\* 参数1：proch\_infoP 信道及载波方向等信息指针

/\* 参数2：data PDU原始数据指针

/\* 参数3：size PDU数据的长度

/\* 参数4：output 解码结果指针

/\* 返回值：解码成功标志，0：成功，-1：解码失败，-2：无对应的传输信道信或上下行方，

-3：无对应逻辑信道

/\* 功能说明：该函数为解码子接口，其根据信道类型和上下行方向，通过调用ASN1C工具

/\* 生成的解码接口完成对不同信息的解码

/\* 调用说明：该函数由rrc\_decode()函数调用

2.ASN1C生成解码接口uper\_decode\_complete()定义与：asn1c/per\_decode.c

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

asn\_dec\_rval\_t

uper\_decode\_complete(asn\_codec\_ctx\_t \*opt\_codec\_ctx,

asn\_TYPE\_descriptor\_t \*td,

void \*\*sptr,

const void \*buffer,

size\_t size)

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

/\* 参数1：opt\_codec\_ctx 此值设为0 禁用堆栈边界检测

/\* 参数2：td RRCPDU类型描述变量，定义与对应消息类型文件中

/\* 参数3：sptr 解码数据存放指针

/\* 参数4：buffer RRCPDU原始数据指针

/\* 参数5：size RRCPDU原始数据大小

/\* 返回值：asn\_dec\_rval\_t 解码返回结构体变量，包含了解码成功标志，0：成功，2：失败

/\* 详见代码

/\* 功能说明：实现对一种RRC PDU类型数据解析

/\* 调用说明：调用该函数前，需要指定是那种类型的消息，并传入对应消息的描述信息；

/\* 若传入的解码结果指针为空，则该解码接口会申请动态内存，使用后需要free()

/\* 申请的内存，防止内存泄漏