#源地址 = 本机簇ID+本机簇内编号

#目的地址 = 目的簇ID+目的簇内编号

Beacon 8bytes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 低功耗 | 目的簇ID | 目的簇内编号 | 源簇ID | 源簇内编号 | 空闲负载数量 | 校验 |
| 8bit | 6 | 1 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |

typedef struct {

uint8 length; // 包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 power:1; //是否开启低功耗

uint8 des\_cluster\_id; //簇ID

uint8 des\_cluster\_innernum; //簇内编号

uint8 cluster\_id; //目的簇ID

uint8 cluster\_innernum; //目的簇内编号

uint8 free\_num:8; //空闲负载数

uint8 check; //校验

}BeaconPacketStruct;

JoinRequest 7bytes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 保留 | 目的簇ID | 目的簇内编号 | 物理地址 | 校验 |
| 8bit | 6 | 1 | 1 | 8 | 8 | 16 | 8 |

typedef struct{

uint8 length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 :1; //保留

uint8 des\_cluster\_id; //目的簇ID

uint8 des\_cluster\_innernum; //目的簇内编号

uint16 phy\_address; //物理地址

uint8 check; //校验

}JoinRequestPacketStruct;

JoinRequest-ACK 9bytes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 接受入网 | 目的地址 | 源地址 | 分配的LA | 校验 |
| 8bit | 6 | 1 | 1 | 16 | 16 | 16 | 8 |

typedef struct{

uint8 length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 accept:1; //是否接受入网

uint8 src\_cluster\_id; //源簇ID

uint8 src\_cluster\_innernum; //源簇内编号

uint16 des\_phy\_address; //目的地址

uint8 cluster\_id; //分配簇ID

uint8 cluster\_innernum; //分配簇内编号

uint8 check; //校验

}JoinRequestACKPacketStruct

JoinRequest-ACK-OK 6bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 保留 | 目的地址 | 源地址 |
| 8bit | 6 | 1 | 1 | 16 | 16 |

typedef struct{

uint8 length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 :1; //保留

uint8 src\_cluster\_id; //源簇ID

uint8 src\_cluster\_innernum; //源簇内编号

uint8 des\_cluster\_id; //目的簇ID

uint8 des\_cluster\_innernum; //目的簇内编号

}JoinRequestACKOKPacketStruct;

LeaveRequest 7bytes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 保留资源 | 目的地址 | 源地址 | 离开原因 |
| 8bit | 6 | 1 | 1 | 16 | 16 | 8 |

LeaveRequest—ACK 6bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 接受保留资源 | 目的地址 | 源地址 |
| 8bit | 6 | 1 | 1 | 16 | 16 |

Data 12bytes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 保留 | 目的地址 | 源地址 | 绝对时隙号 | 数据 | 校验 | ACK丢失数量(QOS Test) |
| 8bit | 6 | 1 | 1 | 16 | 16 | 16 | 8 | 8 | 16 |

typedef struct{

uint8 pack\_length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 :1; //保留

uint8 des\_cluster\_id; //目的簇ID

uint8 des\_cluster\_innernum; //目的簇内编号

uint8 src\_cluster\_id; //源簇ID

uint8 src\_cluster\_innernum; //源簇内编号

uint16 ab\_slot\_num; //绝对时隙号

uint8 data; //数据

uint8 check; //校验

uint16 acklost\_count //ACK 丢失数量

}DataPacketStruct;

Data-ACK 10bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 重新入网 | 目的地址 | 源地址 | 时间戳 | 控制命令 | 校验 |
| 8bit | 6bit | 1 | 1 | 16 | 16 | 16 | 8 | 8 |

typedef struct{

uint8 pack\_length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 rejoin:1; //重新入网

uint8 des\_cluster\_id; //目的簇ID

uint8 des\_cluster\_innernum; //目的簇内编号

uint8 src\_cluster\_id; //源簇ID

uint8 src\_cluster\_innernum; //源簇内编号

uint16 time\_stamp; //时间戳

uint8 cmd; //控制命令

uint8 check; //校验

}DataACKPacketStruct;

ReJoin 9bytes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 包长度 | 包类型 | 需要应答 | 保留 | 目的逻辑地址 | 源地址 | 目的物理地址 | 校验 |
| 8bit | 6bit | 1 | 1 | 16 | 16 | 16 | 8 |

typedef struct{

uint8 length; //包长度

uint8 pack\_type:6; //包类型

uint8 ack\_en:1; //是否需要应答

uint8 :1; //保留

uint8 des\_cluster\_id; //目的簇ID

uint8 des\_cluster\_innernum; //目的簇内编号

uint8 src\_cluster\_id; //源簇ID

uint8 src\_cluster\_innernum; //源簇内编号

uint16 des\_phy\_address; //目的物理地址

uint8 check; //校验

}ReJoinPacketStruct;

//packet type define

#define BEACON\_TYPE 1

#define JOINREQUEST\_TYPE 2

#define JOINREQUESTACK\_TYPE 3

#define JOINREQUESTACKOK\_TYPE 4

#define DATA\_TYPE 5

#define DATAACK\_TYPE 6

#define REJOIN\_TYPE 7

//event handler define

#define EVENT\_SCAN\_CHANNEL 3

#define EVENT\_BEACON\_SEND 4

#define EVENT\_BEACON\_HANDLER 5

#define EVENT\_JOINREQUEST\_SEND 6

#define EVENT\_JOINREQUEST\_HANDLER 7

#define EVENT\_JOINREQUESTACK\_HANDLER 8

#define EVENT\_JOINREQUESTACKOK\_HANDLER 9

#define EVENT\_DATA\_SEND 10

#define EVENT\_DATA\_HANDLER 11

#define EVENT\_DATAACK\_HANDLER 12

#define EVENT\_WAKE\_A7139 13

#define EVENT\_CSMA\_RESEND 14

#define EVENT\_COLLECT\_DATA 15

#define EVENT\_IDENTIFY\_CAR 16

#define EVENT\_UPLOAD\_DATA 17

#define EVENT\_REJOIN\_SEND 18

#define EVENT\_REJOIN\_HANDLER 19