更新纪录

1）201700214： 初版。

2）201700301： 增加demo演示程序文件传输。

3）20170309：增加状态监控软件的帧协议

增加协议栈数据流转记录备忘

4）20170314：修改端口，调整状态监控软件中密码握手包的功能。

5) 20170405：修改原功率参数为RF参数集，增加网关表的设置

# 概述

宽带无线模块主控和基带采用SmartFusion2M150T平台。

## 对外接口



## 内部资源

* 内部CortexM3主频100MHz；
* 内部存储器：64KB+16KB；
* 内部FLASH：512KB
* 外部存储器：64MB LPDDR；
* 以太网接口：100Mbps；

## 无线通信特性

* 通信制式QPSK；
* 传输速率为4Mbps；
* 多频点选择；
* 传输包长可变；（472Bytes的倍数，最大472\*16=7552）
* 发送时长：前导码360us+（发射时间）720us\*N+（射频切换时间）100us；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帧头 | 倍数 | 包长（字节） | 发送时长 |  |
| 0 | 1 | 472 | 1180 | 推荐使用 |
| 1 | 2 | 944 | 1900 | 推荐使用 |
| 2 | 3 | 1416 | 2620 | 推荐使用 |
| 3 | 4 | 1888 | 3340 | 推荐使用 |
| 4 | 5 | 2360 | 4060 |  |
| 5 | 6 | 2832 | 4780 |  |
| 6 | 7 | 3304 | 5500 |  |
| 7 | 8 | 3776 | 6220 |  |
| 8 | 9 | 4248 | 6940 |  |
| 9 | 10 | 4720 | 7660 |  |
| 10 | 11 | 5192 | 8380 |  |
| 11 | 12 | 5664 | 9100 |  |
| 12 | 13 | 6136 | 9820 |  |
| 13 | 14 | 6608 | 10540 |  |
| 14 | 15 | 7080 | 11260 |  |
| 15 | 16 | 7552 | 11980 |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 协议功能

* 组网容量最大32台（同频点）；
* 组内处于同网段内IP节点互访；
* 最大支持15跳；
* 支持IPv4透明传输（支持TCP，UDP，ICMP，IGMP，ARP等）；
* 支持IP，Port，协议类型的选择传输配置；

## 协议软件框架设计



### 任务0

完成设备的初始化工作，并且通过信号量事件，来等待进入测试和测试状态。

#### 测试程序命令格式

[cmd][-m=x];[-l=xxx];[-n=xxx];[-t=xxx];[-src=xxxx];[-dst=xxxx];[-v=xxxx];

cmd：tx，rx，cca，exit

m：0：信号源，1：设备，2：接收显示（默认0）

m：0：发送数据，1：载波测试，2：快速发送

p：接收显示长度（默认36字节）

l：表示帧长度（1888）

n：表示帧数量（1500）

t：发送间隔（5）

src：源地址（0xffff）

dst：目的地址（0xffff）

v：内容字节（0xa5）

ac：CSMA功能（0）

dma：dma接收发射（关闭）

例子：

设置寄存器方式

[0x31000000]=0x1234;可以连续写入。

### 任务1

完成无线通信的MAC层和PHY层协议，采用CSMA-CA方式。

### 任务2

完成无线的NWK路由层和对以太网收发的操作，通过信号量完成。

### 任务3（隐藏）

内嵌简易TCP/IP协议栈，直接调用POSIX标准接口。

### 任务4

应用层相关处理。

### 任务5

空闲任务，负责运行显示，按键和语音轮训处理的过程。

# 帧协议格式

帧总格式：**以小段字节序表示，以方便使用C语言结构体。**

## 

## 备注：checksum校验范围扩展到payload的前16字节，因为这样保证了以太网数据报头的MAC地址和类型正确。

## 上层协议封装



## 物理帧格式

物理帧采用组合方式，第一字节为帧长的472的整数倍。

## Mesh帧格式

### Frame Control（帧控制）



#### Protocol位

默认为0。

#### Type位

|  |  |  |  |
| --- | --- | --- | --- |
| 2-3 | 4 | 5 | 6-7 |
| Ctrl | Probe | 保留 | Data |
| 00：None  01：RTS  10：CTS  11：ACK | 0：None  1：Probe | 0：None  1：Reserve | 0：None  3：QoS高  2：QoS中  1：QoS低 |
|
|
|

#### To DS位和From DS位

略，暂填0。

#### More fragments位

略，暂填0。

#### Retry位

可能需要重传帧。任何重传帧会设置为1，以协助接收端剔除重复帧。

#### Power management位

略，暂填0。

#### More data位

略，暂填0。

#### Protected Frame位

略，暂填0。1表示加密使用。

#### Order位

帧可依序传送，此位设置为1。0则表示不按照依序传输。

### Duration（帧控制）



用来被设置为NAV。此数值代表目前所进行的传输预计使用信道多少微秒。节点必须监听所收到的任何帧头，并据以更新NAV，阻止其他工作站访问介质。

### Mesh ID（网络标识码）

每个网络均具有自己的Mesh ID，所有的无线数据，均检测此ID，若非本机的Mesh ID，除Duration位需要设置，防止碰撞外，其余不需要上传处理。

### Device ID（设备地址）

Device ID1设备地址作为发送方，也就是源地址。Device ID2设备地址作为接收方，也就是目的地址。其中0xFF表示广播地址。其余暂时均作为单体地址。可能后期增加组播地址。Send\_id作为原始发送方地址。Target\_id作为实际需要达到的地址。

### Seq-ctrl（顺序控制）



当上层交付MAC传送师，会赋予一个sequence number（顺序编号）。此位的作用，相当于已传帧的计数器取4096的模（modulo）。此计数器由0起，MAC每处理一个上层封包就会累加1。如果上层封包被切割处理，所有帧片段都会具有相同的顺序编号。如果重传帧，则顺序编号不会有任何改变。

帧片段之间的差异在于fragment number（片段编号）。第一个片段的编号为0,其后每个片段依序累加1。重传的片段会保有原来的sequence number协助重组。

### Frame Length（负载长度）

负载长度以小段字节序表示。

### CRC32（Mesh帧头校验）

校验从PHY到Frame Length，固定校验长度为13个字节。备注数据内容Payload校验，通过Payload的不同类型来进行校验，或者叫由TCP/IP协议，或者上层应用协议。

# 无线路由机制

# 协议栈数据流转记录备忘

## 一，数据包的来源为设备的lwip协议栈，即本节点网卡输出的数据，其目的出口为以下；

## 一），以太网广播包，出口为mesh和eth；

## 二），以太网普通包，查询地址表中此以太网目的mac地址的挂载节点ID：

## 1），地址表中无此条目，丢弃；

## 2），地址表中查询得知挂载节点ID是本节点，出口为eth；

## 3），地址表中查询得知挂载节点ID非本节点，出口为mesh；

## 二，数据包的来源为eth，即内网发来的本节点网卡收到的数据，其目的出口为以下：

## 一），若设置了绑定设备的MAC，则判断以太网源源mac地址，若非此绑定设备mac，丢弃；

## 二），无论何种包，若以太网源目的mac为本节点，出口为ip；

## 三），以太网广播包，分多种情况；

## 1），ARP请求包；

## a），请求的目的IP为本节点IP，出口为ip；

## b），请求的目的IP非本节点IP，出口为mesh；

## 2），IP包，先做部分过滤性操作，如封某些端口IP等，然后再判断出口；

## a），IP广播包，出口为ip和mesh；

## b），目的IP为本节点IP（会有这种包？），出口为ip；

## c），其它，丢弃；

## 四），以太网普通包，查询地址表中此以太网目的mac地址的挂载节点ID：

## 1），地址表中无此条目，丢弃；

## 2），地址表中查询得知挂载节点ID为本节点，不存在这种可能，直接走第二条；

## 3），地址表中查询得知挂载节点ID非本节点，出口为mesh；

## 三，数据包的来源为mesh，即无线自组网收到的数据，其目的出口为以下：

## 一），判断帧类型，若是管理帧，返回，无线自组网协议栈处理；

## 二），无论何种包，若以太网源目的mac为本节点，出口为ip；

## 三），以太网广播包，分多种情况；

## 1），ARP请求包；

## a），请求的目的IP为本节点IP，出口为ip；

## b），请求的目的IP非本节点IP，出口为mesh和eth；

## 2），IP包；

## a），IP广播包，出口为ip和mesh和eth；

## b），目的IP为本节点IP（会有这种包？），出口为ip；

## c），其它，丢弃；

## 四），以太网普通包，确定是转发内网还是转发无线：

## 1），如果目标ID是本节点，出口为eth；

## 2），如果目标ID非本节点，出口为mesh；

# 状态监控软件（配套）

## 监控软件与节点设备采用UDP方式通信，目标端口为60001，本地收发端口默认也为60001，如果此端口不可用，可以使用其它端口。被其帧头信息如下

#pragma pack(1)

typedef struct

{

uint16\_t head;

uint16\_t length;

uint16\_t type;

uint16\_t stype;

}app\_manager\_frm\_head\_t;

#pragma pack()

|  |  |  |  |
| --- | --- | --- | --- |
| 识别码 | 帧长  （不包含帧头） | 帧类型 | 帧内容 |
| 帧头 | | |
| 2字节 | 2字节 | 4字节 |  |
| 0xD5C8 | 0~65535 | 见下表 | Variable |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帧类型 | | 介绍 | | |
| 2字节（主类型） | 2字节（子类型） | | 类型名 | 方向 |
| 0x00（监听） | 0x00 | | 监听帧 | 节点至PC |
| 0x01（状态） | 0x00 | | 状态推送帧 | 节点至PC |
| 0x02（设置） | 0x00 | | 密码握手 | PC至节点 |
| 0x01 | | 密码握手ACK | 节点至PC |
| 0x02 | | ID设置 | PC至节点 |
| 0x03 | | ID设置ACK | 节点至PC |
| 0x04 | | IP设置 | PC至节点 |
| 0x05 | | IP设置ACK | 节点至PC |
| 0x06 | | RF参数集设置 | PC至节点 |
| 0x07 | | RF参数集设置ACK | 节点至PC |
| 0x08 | | 挂载设备查询 | PC至节点 |
| 0x09 | | 挂载设备查询ACK | 节点至PC |
| 0x0a | | 网关表设置 | PC至节点 |
| 0x0b | | 网关表设置ACK | 节点至PC |
| 0x0c | | 网关表清除 | PC至节点 |
| 0x0d | | 网关表清除ACK | 节点至PC |

## 状态推送

状态推送包由各节点设备自行以UDP局域网广播的形式进行定时推送，长度为500Byte，加上帧头共508字节。

软件界面以ID作为行标分行显示，显示参数有以下部分组成：

1. ID（包括device\_id, mesh\_id, mode, type），4Byte
2. IP地址（包括IP地址，子网掩码，网关），12Byte
3. MAC地址，6Byte
4. 定位信息，10Byte
5. 路由路径和链路质量，2\*N Byte（N为最大节点数，暂定32）
6. 电量参数，1Byte
7. RF参数集，179Byte
8. 网关表，50Byte
9. 剩余待添加

附相关数据结构：

1)定位信息10字节

typedef struct \_\_s\_dev\_pos

{

uint16\_t long\_int;

uint16\_t long\_dec;

bool\_t we;

uint16\_t lat\_int;

uint16\_t lat\_dec;

bool\_t sn;

} dev\_pos\_t;

2)RF参数集179字节

typedef struct

{

uint8\_t pll;

uint8\_t set1;

uint8\_t set2;

uint8\_t set3;

uint8\_t set4;

fp64\_t lo;

uint8\_t ref;

}hal\_rf\_freq\_t;

typedef struct

{

hal\_rf\_freq\_t freq\_cal;

uint8\_t use\_level;

uint32\_t pa\_power[10];//0@1300 1@1400

uint32\_t ofdm\_lms\_power[10];//0@1300 1@1400

uint32\_t ofdm\_scl\_power[10];//0@1300 1@1400

int32\_t ofdm\_rssi\_offset[10];//0@1300 1@1400

fp32\_t ofdm\_rssi\_thred;//

}hal\_rf\_param\_t;

3)网关表50字节

typedef struct \_gateway\_table\_item\_t {

uint8\_t addr[6];

uint32\_t net\_segment;

} gateway\_table\_item\_t;

typedef struct \_gateway\_table\_t {

gateway\_table\_item\_t item[5];

} gateway\_table\_t;

## 参数设置与查询

参数设置查询由本软件主动发起，节点响应后进行操作并给予反馈。首先必需进行密码握手，成功后才能进行其它设置。

### 密码握手包

软件发送给节点，包括6Byte的密码字符。

### 密码握手包ACK

节点发送给软件。内容包括一字节的信息，为0表示成功，为1表示密码错误。

### ID设置包

软件发送给节点，包括4Byte的ID。

### ID设置包ACK

节点发送给软件，内容包括一字节的信息，为0表示成功，为1表示设置不成功。

### IP设置包

软件发送给节点，包括4Byte的IP，4Byte的掩码，4Byte的网关，共12字节。

### IP设置包ACK

节点发送给软件，内容包括一字节的信息，为0表示成功，为1表示设置不成功。

### RF参数集设置包

软件发送给节点，包括179Byte的RF参数集设置。

### RF参数集设置包ACK

节点发送给软件，内容包括一字节的信息，为0表示成功，为1表示设置不成功。

### 挂载设备查询包

软件发送给节点，内容待定。

### 挂载设备查询包ACK

节点发送给软件，内容包括若干字节的信息，第1字节为设备数量N，其后是N个设备的信息，每个设备包括4字节的ip地址，6字节的mac地址。

### 网关表设置包

软件发送给节点，包括50Byte网关表设置。

### 网关表设置包ACK

节点发送给软件，内容包括一字节的信息，为0表示成功，为1表示设置不成功。

### 网关表清除包

软件发送给节点，无内容。

### 网关表清除包ACK

节点发送给软件，内容包括一字节的信息，为0表示成功，为1表示设置不成功。

# 监听诊断软件（配套）

进入监听模式，则将节点接收到的有线和无线数据，全部显示出来，作为数据流转的诊断分析工具。参考Wireshark软件。

## 监听诊断协议

采用UDP方式推送至监听软件，设备端口任意，监听软件端口默认为60000。

#pragma pack(1)

typedef struct

{

uint16\_t head;

uint16\_t length;

uint32\_t type;

}app\_sniffer\_frm\_head\_t;

#pragma pack()

|  |  |  |  |
| --- | --- | --- | --- |
| 识别码 | 帧长  （不包含帧头） | 帧类型 | 帧内容 |
| 帧头 | | |
| 2字节 | 2字节 | 4字节 | Mesh帧 |
| 0xD5C8 | 0~65535 | 0为监听数据类型 | Variable |

## 结构体表征

#define NONE 0

#define PROBE 1

#define RTS 1

#define CTS 2

#define ACK 3

#define QOS\_H 3

#define QOS\_M 2

#pragma pack(1)

typedef struct

{

uint16\_t protocol: 2,

ctl\_mgmt: 2,

probe\_flag: 1,

reserve: 1,

qos\_level: 2,

to\_ds: 1,

from\_ds: 1,

more\_frag: 1,

retry: 1,

pwr\_mgmt: 1,

more\_data: 1,

protected: 1,

order: 1;

}mac\_frm\_ctrl\_t;

typedef struct

{

uint16\_t nav: 15,

reserve: 1;

}mac\_frm\_duration\_t;

typedef struct

{

uint16\_t frag\_num: 4,

seq\_num: 12;

}mac\_frm\_seq\_t;

typedef struct

{

uint8\_t phy;

mac\_frm\_ctrl\_t frm\_ctrl;

uint16\_t duration;

uint8\_t mesh\_id;

uint8\_t src\_dev\_id;

uint8\_t dest\_dev\_id;

uint8\_t sender\_id;

uint8\_t target\_id;

mac\_frm\_seq\_t seq\_ctrl;

uint16\_t frm\_len;

uint16\_t chksum;

} mac\_frm\_head\_t;

#pragma pack()

#define NODE\_MAX\_NUM 32

#pragma pack(1)

typedef struct \_neighbor\_field\_item\_t

{

uint8\_t snr;

} neighbor\_field\_item\_t;

typedef struct \_neighbor\_field\_t

{

neighbor\_field\_item\_t item[NODE\_MAX\_NUM];

} neighbor\_field\_t;

typedef struct \_route\_field\_item\_t

{

uint8\_t next\_id;

uint8\_t hop;

uint32\_t seq;

} route\_field\_item\_t;

typedef struct \_route\_field\_t

{

route\_field\_item\_t item[NODE\_MAX\_NUM];

} route\_field\_t;

#pragma pack()

typedef struct \_probe\_data\_t

{

neighbor\_field\_t nf;

route\_field\_t rf;

//affiliate\_field\_t af;

uint16\_t chksum;

} probe\_data\_t;

# 演示软件（配套）

采用多协议传输任意大小的文件。

传输方式可选：TCP/UDP(目前只需完成UDP方式)。

单次传输Block Size可配置，发送间隔可配置。

接收文件保存路径可选择。

发送文件路径可选择。

UDP方式：传输需要有确认机制。