**泰凌微TLSR8258开发笔记**

1. TLSR8258基本特性

1.32 Bit RISC-V 处理器，最高主频48MHz，内置32.768KHz RC时钟

2.512 KBytes 片上可编程Flash，64 KBytes 片上RAM

3.内置硬件AES加密单元

4.丰富的外设：17路GPIO，一路SPI、IIC、UART、六路PWM，支持红外信号生成

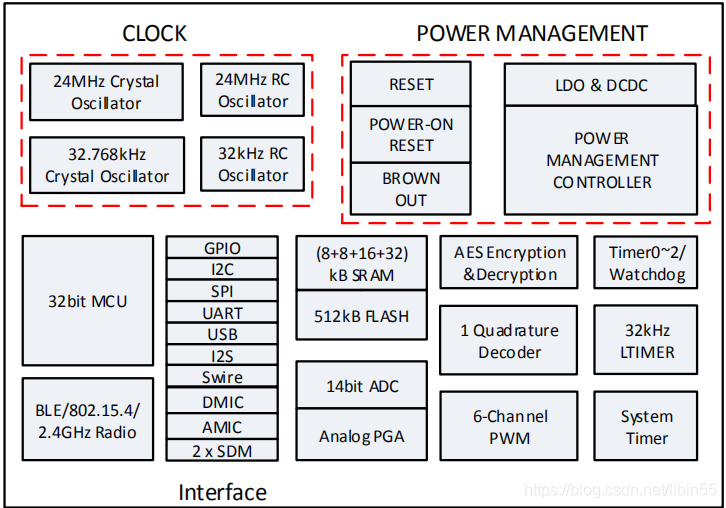
5.6路14Bit高精度ADC，支持电池电量采集、4通道差分输入可编程增益PGA

6.高达10dBm的发射功率，-96dBm接受灵敏度

7.优秀的功耗管理，深度睡眠电流0.4uA，保持RAM数据睡眠电流1.4uA，接收模式5.3mA，发送模式4.8mA(0dbm)

8.支持低功耗蓝牙5.0 (不支持经典蓝牙)、支持SigMesh，可对接天猫精灵

9.电压范围1.8V~3.6V



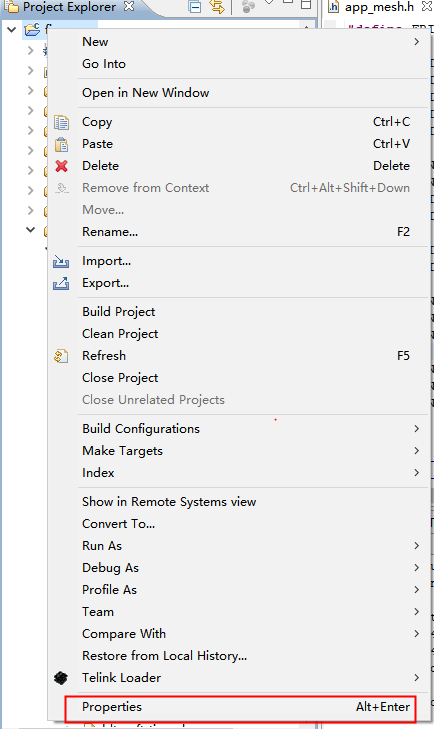
1. IDE使用

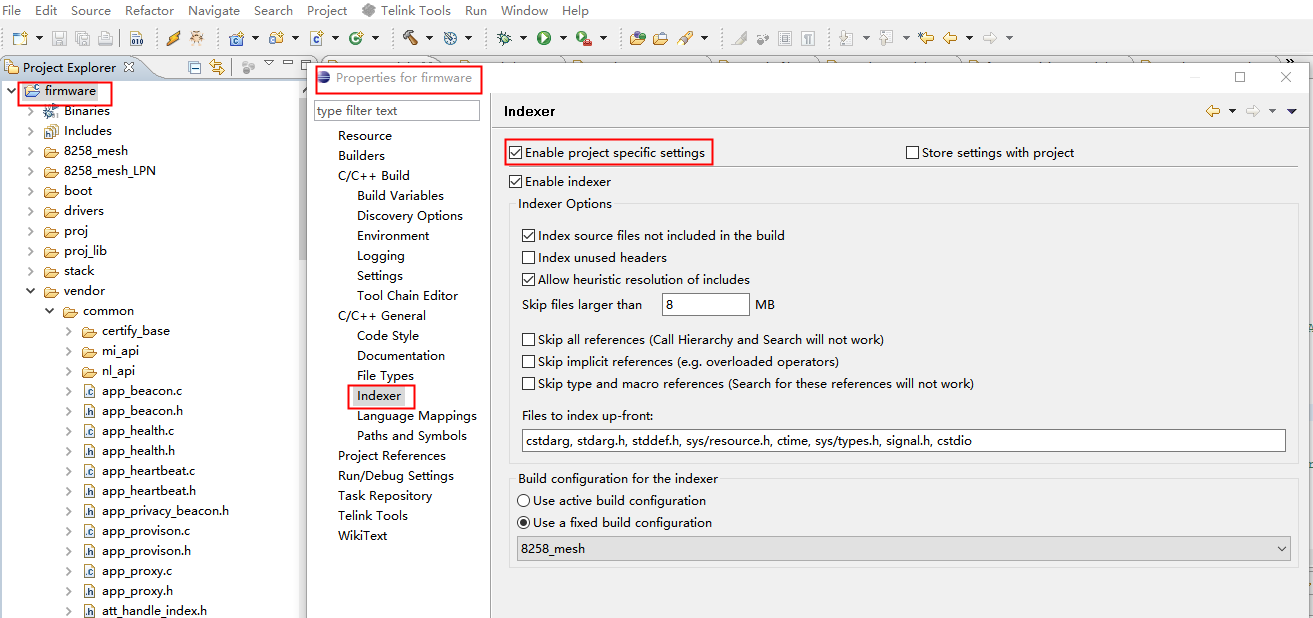
问题一：修改头文件内容，编译没有效果

解决方法：项目有涉及头文件修改的，要先clean项目！！！

问题二：选择了编译项目，项目内的文件显示未选中状态

解决方法：点击项目，右键选择properties，然后设置成下图所示，即可





1. Mesh SDK工程开发

1.Log打印串口使用：8258默认使用串口引脚PB2，根据手册1.6.2 Log 打印调试 配置即可

1. 增加vendor指令，mesh\_cmd\_vd\_func[]内的回调函数需以cb\_开头，否则指令接收报错
2. 结构体使用“.”报错，应换成“->”

typedef struct{

u16 nk\_idx; //netkey index

u16 ak\_idx; //app\_key index

u8 retry\_cnt; // 应用层需要retry的次数，适用于需要回复的命令，为0表示使用“retry”控件的值。

u8 rsp\_max; // 表示需要回复节点的个数，0和1是效果相同，用来判断是否需要retry

u16 adr\_dst; // 目标地址

u8 op; //操作码，取值范围0xC0—0xFF

u16 cid; //company id，即vendor id

u8 op\_rsp; //伪参数，内部使用，不会发送出去，详见下文说明

u8 tid\_pos; //伪参数，内部使用，不会发送出去，详见下文说明

u8 par[MESH\_CMD\_ACCESS\_LEN\_MAX]; // parameter after opcode

};

1. mesh自组网自定义协议

低功耗节点只能接收单播信息

接收到节点信息时，判断目标节点是否为朋友节点判断，为朋友节点时，向低功耗节点发送相应的数据

协议内容：目标节点地址（2byte）+ 源节点地址（2byte）+ data

**int** is\_cmd2lpn(u16 adr\_dst) //查找是否为与自己建立朋友关系的低功耗节点

1. Mesh基本功能、基本特性

网关：增加删除节点，分配节点地址，对相应的节点进行控制、发送数据

普通节点：具有朋友、中继功能，最多支持200个实时控制节点，朋友节点最多支持10个低功耗节点（手册建议不要超过8个）

低功耗节点：只能通过朋友节点接收和发送数据

消息最多支持127次中继；sdk默认10次 TTL\_DEFAULT

Mesh网络中以UUID来识别node，UUID可作为唯一识别码

增加一个恢复出厂设置按键：长按3s，清除网络信息，进入配网状态

测试内容：

通信响应时间

最大的通信节点数量

低功耗节点功耗

1. mesh地址

在蓝牙Mesh规范中，将地址分为四类：

单播地址（Unicast Address）用于唯一标识网络中节点的元素地址。

组播地址（Group Address）用于表示网络中的组地址，组地址可以被一个或多个节点所共有。

虚拟地址（Virtual Address）作用类似与组地址，不同点在于其地址空间为2^16。

未分配地址 （Unassigned address）表示此地址不能被使用。

单播地址

单播地址被用于唯一的标识蓝牙Mesh网络中节点的元素（Element)，单播地址从0x0001开始到0x7FFF结束，该地址是在蓝牙Mesh设备配网阶段由配置者分配给设备的，单播地址在各节点之间不能重复，而在节点内必须递增。例如：网络由三个节点组成，节点A有1个元素，节点B有2个元素，节点C由2个元素，节点A的地址为0x0001，节点B的第一个元素如果为0x0002的话，第二个元素必须为0x0003，节点C同理。单播地址在网络中用于唯一的标识网络的节点的元素，因此在网络数据包中源地址（SRC）字段必须设置为该单播地址，而目的地址（DST）则无此要求。

组播地址

组播地址一般被用于元素模型的订阅（Subscribe）和发布（Publish）地址，组播地址从0xC000开始，到0xFEEE结束可以被任意使用，而0xFF00到0xFFFB为保留地址，不能使用。0xFFFC到0xFFFF用于特殊使用：

0xFFFC 代指所有的代理节点（Proxy Node）

0xFFFD 代指所有的朋友节点（Friend Node）

0xFFFE 代指所有的转发节点（Relay Node）

0xFFFF 代指所有的节点（All Nodes）

蓝牙Mesh网络的通信模型是取决于目的地址（DST）字段，如果目的地址为单播地址，那么该消息只能被单播地址为该目的地址的元素所处理，如果目的地址为组播地址的话，该消息可以被订阅该地址的模型所处理。

由此推出通信模型：客户端/服务端模型 订阅/发布模型

1. uart串口

通过此语句可获得串口接收数据

uart\_data\_t\* q = (uart\_data\_t \*)my\_fifo\_get(&hci\_rx\_fifo);

**if** (uart\_Send(q->data, q->len))

{

my\_fifo\_pop (&hci\_tx\_fifo);

}

发送数据后hci\_tx\_fifo队列指针跟着变化

现在出现接收数据后，需要两到三帧数据才能正确接收，中间接收数据为上一次的数据，未及时更新，问题待排查

1. ADC采样

共10个采样引脚

GPIO\_PinTypeDef ADC\_GPIO\_tab[10] = {

*GPIO\_PB0*,*GPIO\_PB1*,

*GPIO\_PB2*,*GPIO\_PB3*,

*GPIO\_PB4*,*GPIO\_PB5*,

*GPIO\_PB6*,*GPIO\_PB7*,

*GPIO\_PC4*,*GPIO\_PC5*

};

泰凌微开发板默认自带电源电压检测，需用ADC采样时，需关闭电源电压检测

1. 配网流程

配网流程分为5个阶段：

* 信标(Beconing)阶段

搜索周边未配网设备，mesh beacon type：Unprovisioned Device（0x00）

* 邀请(Invitation)阶段

配网器与未配网设备建立连接

* 交换公钥(Exchange Public Keys)阶段

明文公钥或者OOB公钥

OOB公钥指通过二维码、NFC等间接方式获取的公钥

* 身份认证(Authentication)阶段

通过使用ECDH秘钥以及OOB秘钥消息对随机数进行加密计算得到认证值，双方 通过对比换算后的认证值来完成认证

包括4种认证方式：输出式OOB认证、输入式OOB认证、静态OOB认证、无 OOB认证

* 分发配网数据(Distribution Of Provisioning Data)阶段

分配配网数据，包括网络秘钥、秘钥索引、单播地址等

配置数据格式：

|  |  |  |
| --- | --- | --- |
| Field | Size(byte) | Note |
| Network Key | 16 | 网络密钥 |
| Key Index | 2 | 网络密钥索引 |
| Flags | 1 | 标识 |
| IV Index | 4 | 当前IV索引 |
| Unicast Address | 2 | 主元素单播地址 |

具体可参考：https://blog.csdn.net/libin55/article/details/112933779

1. mesh\_node.c

节点基本信息定义在mesh\_composition\_data\_local\_t中完成

typedef struct{

u16 loc;

u8 nums;

u8 numv;

u16 md\_sig[ARRAY\_SIZE(md\_id\_sig\_primary)];

u32 md\_vendor[ARRAY\_SIZE(md\_id\_vendor\_primary)];

}mesh\_element\_primary\_t;

typedef struct{

u16 loc;

u8 nums;

u8 numv;

u16 md\_sig[ARRAY\_SIZE(md\_id\_sig\_second)];

#if (LIGHT\_CNT > 1)

#if MD\_SERVER\_EN

u32 md\_vendor[ARRAY\_SIZE(md\_id\_vendor\_second)];

#endif

#endif

}mesh\_element\_second\_t;

typedef struct{

mesh\_page0\_head\_t head;

mesh\_element\_primary\_t ele\_primary;

#if (ELE\_CNT > 1)

mesh\_element\_second\_t ele\_second[ELE\_CNT-1];

#endif

}page0\_local\_t;

typedef struct{

page0\_local\_t page0; // differrent page with differrent struct, so can not use array.

}mesh\_composition\_data\_local\_t;

typedef struct{

u16 relay :1;

u16 proxy :1;

u16 frid :1;

u16 low\_power :1;

u16 rfu :12;

}mesh\_page\_feature\_t;

typedef struct{

u16 cid; //company id

u16 pid; //product id

u16 vid; //version id

u16 crpl; //设备中重放保护列表项的最小数量

mesh\_page\_feature\_t feature;

}mesh\_page0\_head\_t;

其中pid决定了节点的具体功能，目前支持以下几种功能

#define LIGHT\_TYPE\_NONE 0

#define LIGHT\_TYPE\_CT 1

#define LIGHT\_TYPE\_HSL 2

#define LIGHT\_TYPE\_XYL 3

#define LIGHT\_TYPE\_POWER 4

#define LIGHT\_TYPE\_CT\_HSL 5

#define LIGHT\_TYPE\_DIM 6 // only single PWM

#define LIGHT\_TYPE\_PANEL 7 // only ONOFF model

#define LIGHT\_TYPE\_LPN\_ONOFF\_LEVEL 8 // only ONOFF , LEVEL model

#define TYPE\_TOOTH\_BRUSH 9

1. mesh工程下的app.c

user\_init();

1. mesh工程下的mesh\_config.h