

使用指南

N32WB03x MESH使用指南

简介

本文档将介绍N32WB03X MESH的原理和应用例程，帮助用户了解MESH流程，便于快速开发。

N32WB03X MESH功能特点：

- SIG MESH是一个适合小数据，单向传输的网络。
- SIG MESH为网状网络，适合消息需要多次跳转的场景。
- SIG MESH支持工业界的安全性，支持256bit椭圆曲线算法，和多种可选式身份验证机制。
- SIG MESH支持AES 128消息加密，确保消息安全。
- SIG MESH支持订阅与发布机制，一对多的订阅，和一对多的发布。
- SIG MESH支持中继和低功耗功能。

目录

简介	1
1 实例演示	3
1.1 LIGHT_SWITCH例程	3
1.2 入网和配置	3
1.3 订阅与发布测试	3
1.4 代理功能测试	3
1.5 RELAY节点测试	4
1.6 手动删除节点信息	4
2 FLASH内存分布	5
3 API接口讲解.....	6
3.1 GENERIC ON OFF SERVER MODEL 注册	6
3.2 GENERIC ON OFF SET 回调	6
3.3 GENERIC ON OFF GET 回调	6
3.4 GENERIC ON OFF CLIENT MODEL 注册	7
3.5 GENERIC ON OFF CLIENT SET UNACK 发送	7
3.6 MESH NVDS 数据擦除.....	7
3.7 协议栈初始化	8
3.8 协议栈使能	8
3.9 清除参数	8
3.10 MAC地址	8
4 例程讲解.....	9
4.1 LIGHT_SWITCH例程介绍	9
5 常见问题.....	11
5.1 程序烧好后，如果不能开机，先用全片擦除，清除FLASH中的数据。	11
5.2 配网过程中失败，可以使用全片擦除，再进行配网，或者APP中手动进行配置，先GET COMPOSITION，再ADD APP KEY。	错误!未定义书签。
6 版本历史.....	12
7 声明.....	13

1 实例演示

1.1 light_switch例程

- MESH灯控项目目录: n32wb03x_mesh v1.0.0\examples\light_switch\MDK-ARM\light_switch.uvprojx
- 烧录文件目录: n32wb03x_mesh v1.0.0\examples\light_switch\MDK-ARM\output\light_switch.hex
- 测试APP: 使用Nordic Mesh APP。
- 使用JLink烧录light_switch.hex, 复位芯片, 查看GPIO PB12串口输出 (波特率: 1Mbps, 8N1)

1.2 入网和配置

- 打开Nordic Mesh APP, 点击屏幕右下角ADD NODE按键, 允许提示权限。
- 打开已烧录light_switch.hex的开发板。
- APP扫描到开发板 (MESH_XXXX : XXXX为MAC地址后两个字节)
- 点击MESH_XXXX, 点击IDENTITY, 点击PROVISION, 点击OK, 等待入网结束。
- APP回到主界面, 点击MESH_XXXX设备, 点击Elements卡片右边的下角符号。
- 配置Generic On Off Server Model : 点击Generic On Off Server, 点击BIND KEY, 选择Key1, 再点击SUBSCRIBE选择existing group (如果没有选择Create a new group to subscribe), 完成订阅配置, 屏幕最下面可以点击ON/OFF测试开发板的灯是否开关正常。
- 配置Generic On Off Client Model : 点击Generic On Off Client, 点击BIND KEY, 选择Key1, 再点击SET PUBLICATION, 再点击Publish Address, 点击All Nodes右边的下标, 选择Groups, 默认和SUBSCRIBE一样的GROUP, 返回界面点击右下角的APPLY, 这样publish和subscribe属于一个group, 不同的开发板就可以相互订阅和发布。

1.3 订阅与发布测试

- 准备两个或两个以上的开发板, 每个开发板都需要完成入网和配置过程。
- Generic On Off Server的Subscription group需要和Generic On Off Client的Publish group配置一样。
- 按下开发板的button1其它开发板的灯都亮起。
- 按下开发板的button2其它开发板的灯都灭掉。

1.4 代理功能测试

- 准备两个或两个以上的开发板, 每个开发板都需要完成入网和配置过程。

- 点击APP左上角CONNECT标签，选择连接一个开发板节点。
- 回到APP Network Tab，点击选择另外一个开发板，进入Generic On Off Server，在界面右下角控制开发板灯的亮灭。

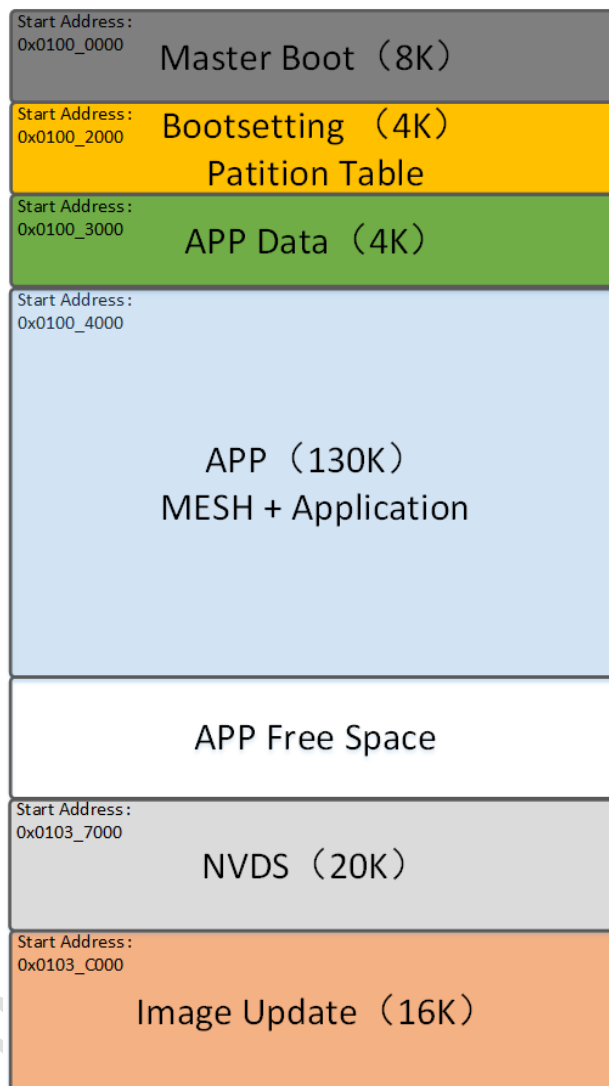
1.5 Relay节点测试

- 准备三个或三个以上的开发板，每个开发板都需要完成入网和配置过程。
- 三个开发板之间相隔5米到10米，通过开发板的button1控制最远端开发板灯亮，再通过开发板button2控制最远端开发板灯灭，完成relay节点测试。

1.6 手动删除节点信息

- 长按开发板button2，芯片会恢复成未入网状态，APP端删除设备后，重新开始入网和配置流程。

2 FLASH内存分布



- Light_switch例程: Program Size: Code=120260 RO-data=5664 RW-data=1492 ZI-data=20088
- Mesh协议栈占用24K Ram (4K RAM用于加密算法), 用户可以使用8K Ram。
- Mesh None Volatile Data Storage 占用5个Flash Sectors (5*4K), 分别存放网络Key和应用层配置
- Master Boot 用于程序跳转和串口升级。
- Bootsetting 用于保存分区表信息。
- APP DATA 用于保存应用数据。
- APP 应用程序。
- APP Free Space 应用程序剩余空间。
- NVDS Mesh 数据保存区域。
- Image Update 用于蓝牙OTA升级。

3 API接口讲解

3.1 Generic On Off Server Model 注册

- APP_ONOFF_SERVER_DEF(m_onoff_server_0,
false,
MESH_TRANSMIC_SIZE_SMALL,
app_onoff_server_set_cb, //状态改变回调函数
app_onoff_server_get_cb) //读取状态回调函数
- 使用宏定义配置灯状态改变回调函数，和读取状态回调函数。
- app_onoff_init(&m_onoff_server_0, 0); //第二个参数是Element Index，因为例程只有一个Element，所以配置为0
- 使用接口将注册宏定义Model至第0个Element。

3.2 Generic On Off Set 回调

- static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff)
{
if(onoff){
bsp_led_on(LED2_GPIO_PORT, LED_GPIO2_PIN);
}else{
bsp_led_off(LED2_GPIO_PORT, LED_GPIO2_PIN);
}
}
- 收到需要改变灯状态的消息时，进入该回调函数，函数中判断onoff状态，根据状态设置灯的亮灭。

3.3 Generic On Off Get 回调

- static void app_onoff_server_get_cb(const app_onoff_server_t * p_server, bool * p_present_onoff)
{
*p_present_onoff = bsp_led_read(LED2_GPIO_PORT, LED_GPIO2_PIN);
}
- 收到需要读取灯状态的消息时，进入该回调函数，将灯状态赋值给p_present_onoff变量。

3.4 Generic On Off Client Model 注册

- static generic_onoff_client_t m_clients = {
 .settings.p_callbacks = NULL, //回调函数，UnAck消息不需要注册回调
 .settings.timeout = 0,
 .settings.force_segmented = false,
 .settings.transmic_size = MESH_TRANSMIC_SIZE_SMALL,
 };
- generic_onoff_client_init(&m_clients, 0); //第二个参数是Element Index，因为例程只有一个Element，所以配置为0

3.5 Generic On Off Client Set UnAck 发送

- static void generic_onoff_set(bool onoff)
 {
 static uint8_t tid = 0; //Transaction ID 每次发送加一
 generic_onoff_set_params_t set_params;
 set_params.on_off = onoff; //设置ONOFF状态
 set_params.tid = tid++;
 generic_onoff_client_set_unack(&m_clients, &set_params, NULL, 0); //发送不用响应的消息
 }

3.6 Mesh NVDS 数据擦除

- static void mesh_nvds_erase(void)
 {
 Qflash_Erase_Sector(0x103B000); //擦除0x103B000地址后4K数据
 Qflash_Erase_Sector(0x103C000);
 Qflash_Erase_Sector(0x103D000);
 Qflash_Erase_Sector(0x103E000);
 Qflash_Erase_Sector(0x103F000);
 NVIC_SystemReset(); //软复位
 }

3.7 协议栈初始化

```
/**  
 * @brief Initialize the mesh stack.  
 * @param model_init function used to initialize the application models.  
 */  
void ns_mesh_stack_init(void (*model_init)(void))
```

3.8 协议栈使能

```
/**  
 * @brief Mesh stack enables.  
 */  
void ns_mesh_stack_enable(void)
```

3.9 清除参数

```
/**  
 * @brief Reset the flash sectors.  
 */  
void ns_mesh_nvds_clear(void)
```

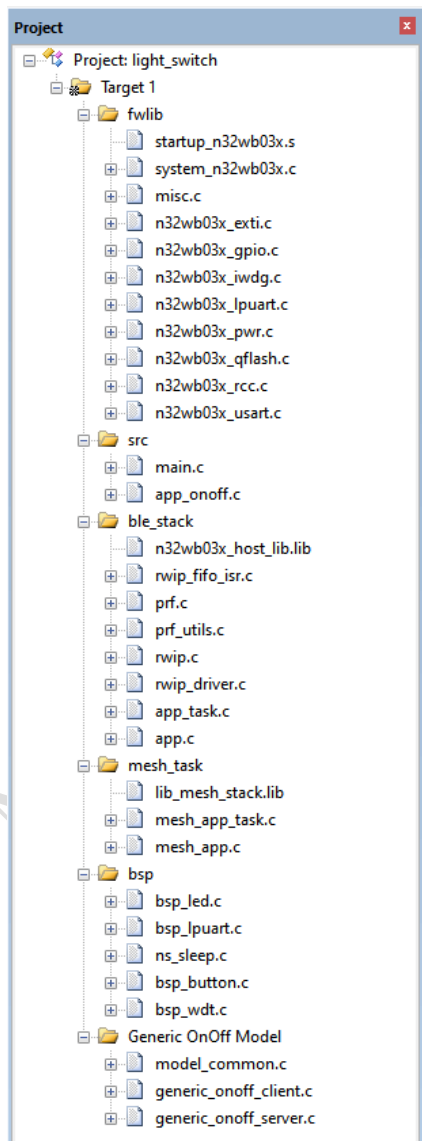
3.10 Mac地址

```
/**  
 * @brief A method to get the mac address.  
 * @note This function should only be called at initialization, since the address will be used later.  
 */  
void make_ble_mac(void)
```

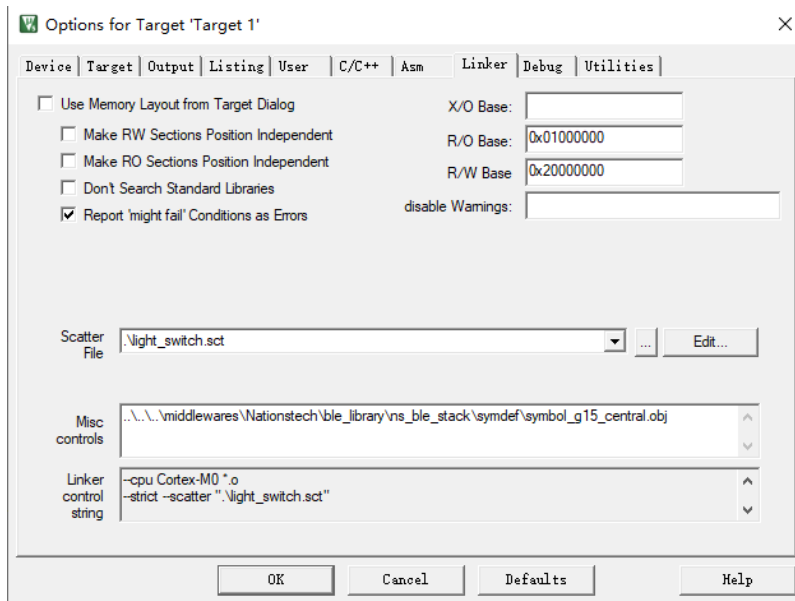
4 例程讲解

4.1 Light_switch例程介绍

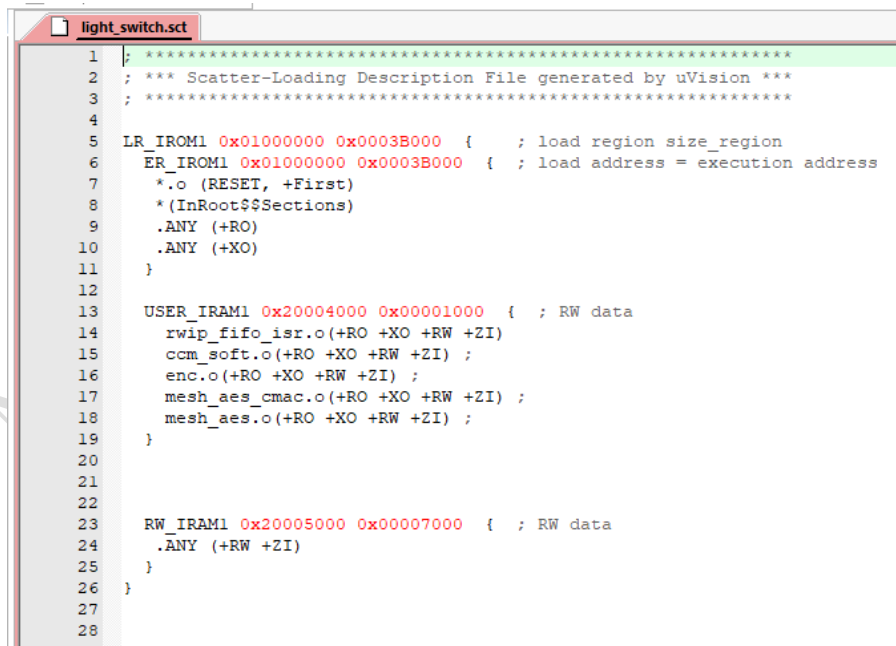
- 拥有一个Element，三个Model， Configuration Server Model， Generic On Off Server Model， 和Generic On Off Client Model。
- 系统时钟设置64MHz。
- 按键驱动支持短按键发送Generic OnOff Client Set Messages， 和长按键删除入网和配置信息并复位功能。
- 节点支持Proxy代理功能， 和Relay中继功能。
- ns32wb03x_host_lib.lib蓝牙host library。
- Lib_mesh_stack.lib SIG MESH library。



- fwlib 目录存放硬件相关驱动库
- src 目录存放上层应用相关文件
- ble_stack 目录存放蓝牙相关文件
- mesh_task 目录存放Mesh相关文件
- bsp 目录存放开发板灯，按键，以及设备相关文件
- Generic_OnOff_Model 目录存放Generic OnOff模型相关文件



- 加入自定义light_switch.sct文件。



- USER_IRAM1将加密算法放入RAM执行，加快算法速度。

5 常见问题

- 5.1 用户程序烧写后，如果不能正常运行MESH功能或者配网过程中失败，可能跟flash data区默认有数据有关，需要执行芯片全擦后，再烧写用户程序，并配网。

NATIONS CONFIDENTIAL

6 版本历史

日期	版本	修改
2021.12.01	V1.0	初始版本

7 声明

国民技术股份有限公司（以下简称国民技术）保有不事先通知而修改的权利。国民技术认为提供的信息准确可信，尽管这样，国民技术对准确性和可靠性不承担任何责任。购买前请获取器件说明的最新版本。在法律允许的最大范围内，任何明示、暗示或保证，包括但不限于适销性、特定用途适用性和第三方知识产权侵权责任，国民技术概不承担不承认。在任何情况下，国民技术均不对因使用本产品而产生的任何直接、间接、偶然、特殊、惩戒性或后果性损害负责，即使已告知可能发生此类损害。不建议应用于与生命相关的设备和系统。国民技术对本手册拥有专属产权。未经明确许可，任何人不得以任何理由对本手册的全部或部分进行使用、复制、修改、抄录、传播。