

ble_peripheral.md

1. Introduction

1.1 Demo Function

1. 设备做为外设，包含多个服务 profile:

- `battery server`,
- `ota server`,
- `custom server`.

此工程实现了设备作为外设的功能，外设备备透传功能，OTA 升级功能。透传功能在 `usr_server` 中实现，OTA 升级功能在 `ota_server` 中实现。此文档重点介绍透传功能，OTA 升级功能的详细说明见 OTA 文档。

程序主要流程为：

- 设备初始化
- 设备发送广播，等待主设备连接. 调用函数 `app_start_advertising` 实现。
- 被主设备服务发现完成.
- 主设备配置属性描述符，在回调函数 `custom_svc_cb_att_val_set` 中处理，并发送 `notify` 数据，调用函数 `slave_send_data`。
- 与主设备断连后重新发起广播.

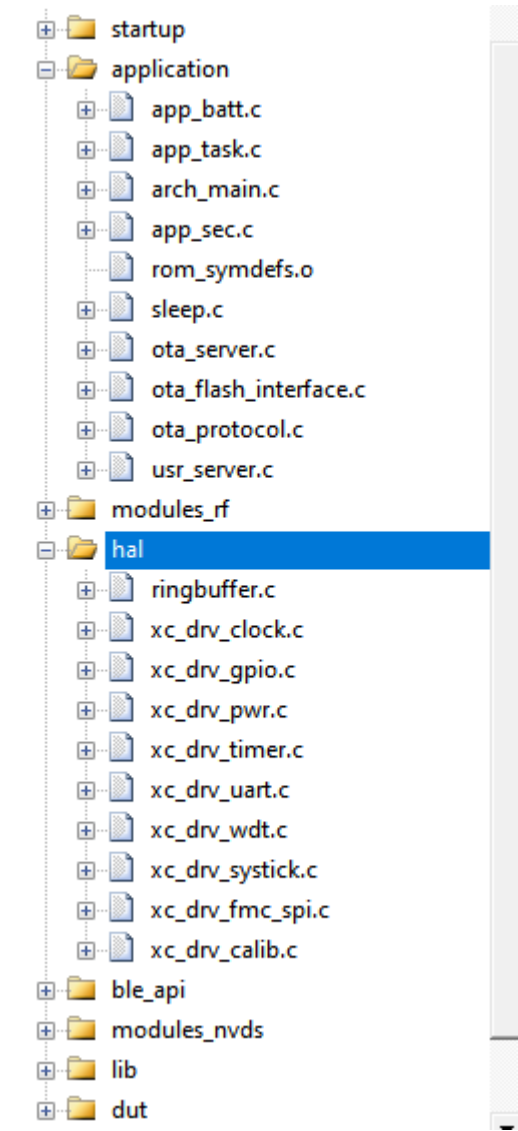
从设备可以像主设备发起安全配对请求，支持传统配对与安全配对.

1.2 Demo file structure

该工程由5个目录组成:

- `start_up`: 包含工程的启动文件
- `dut`: 包含设备rf 测试用的文件，与cmw500 的uart 通讯，该功能默认不使能，用户可以不用关心。
- `application`: 包含设备的所有应用逻辑
- `modules_rf`: 包含设备的rf 驱动配置，用户可以不关心。
- `hal`: 包含设备的所有驱动，比如 `adc`, `spi`, `gpio` 等

目录结构如下图所示：



其中 application 目录下每个文件功能如下表：

File Name	Description
app_batt.c	实现 battery server profile 的逻辑
app_task.c	服务添加，发送广播，建立连接等逻辑
arch_main.c	主程序 main 函数入口，负责事件调度，休眠逻辑等
app_sec.c	处理设备配对相关逻辑
sleep.c	实现蓝牙低功耗的逻辑
ota_server.c	实现了OTA profileOTA APP的交互逻辑
ota_protocol.c	实现了 OTA APP的交互逻辑
ota_flash_interface.c	实现了OTA 过程中对flash 的操作逻辑，每缓存 4k 数据写入一下 flash
usr_server.c	实现了透传的交互逻辑

usr_server.c 文件实现的自定义服务的UUID, 以及对应的配置如下图所示：

Unknown Service

UUID: 0xFFFF0

PRIMARY SERVICE

Unknown Characteristic

UUID: 0xFFFF3

Properties: READ, WRITE, WRITE NO RESPONSE

Unknown Characteristic

UUID: 0xFFFF4

Properties: NOTIFY, READ

Descriptors:

Client Characteristic Configuration

UUID: 0x2902

↓↑

↓⚡

↓↑

> Note: > > 每一个 profile 应该是一个单独的文件。 > > arch_main.c 中用户可以添加应用逻辑，但程序执行时间不能太长，否则会影响到协议栈的调度。

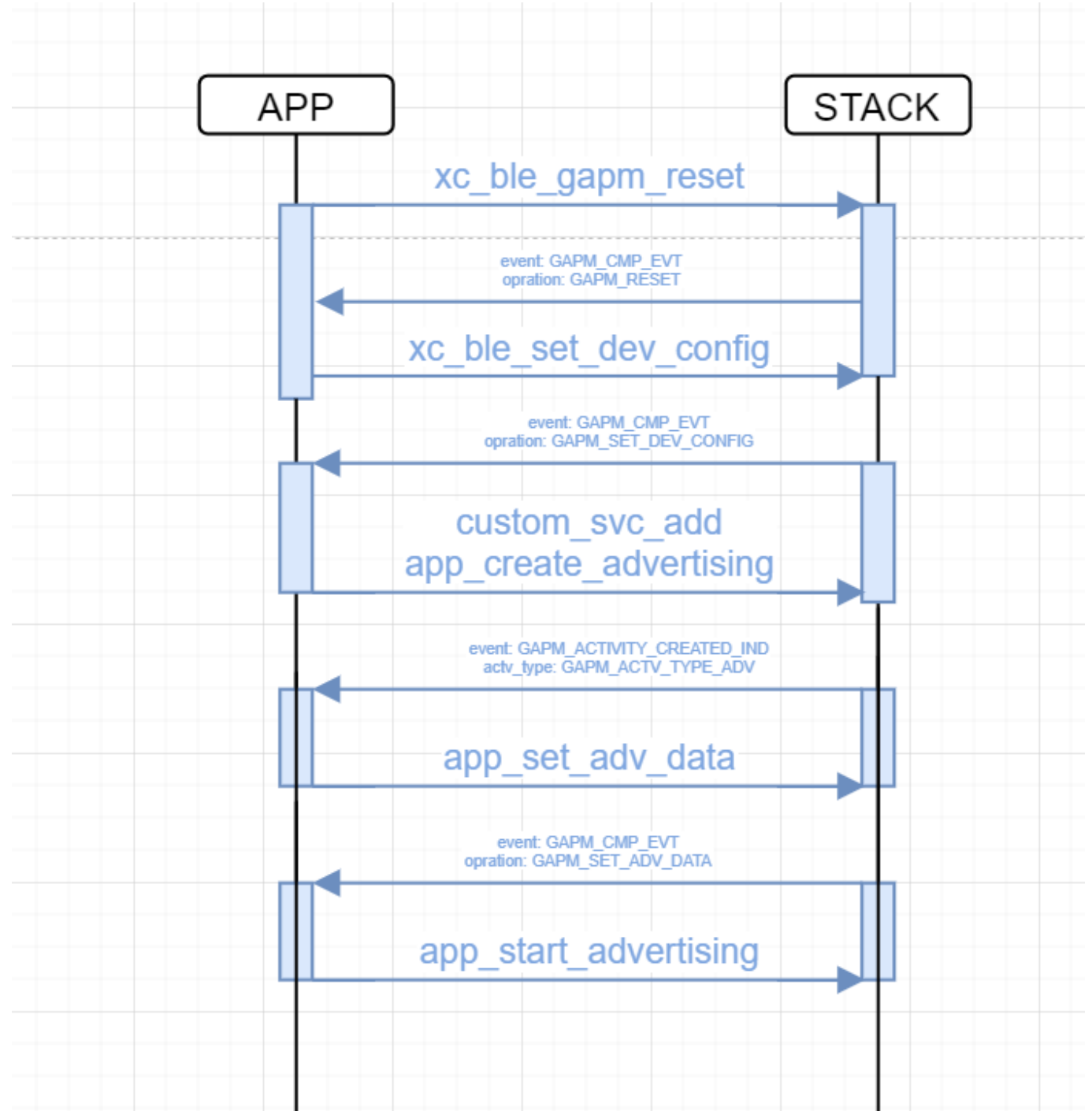
1.3 环境准备

- 设备烧写并运行 ble_peripheral 工程.
- 手机端可以使用 nrf connect app, 如果需要使用 OTA 升级功能，则需要使用 BLEOTA app.

1.4 程序完整运行逻辑

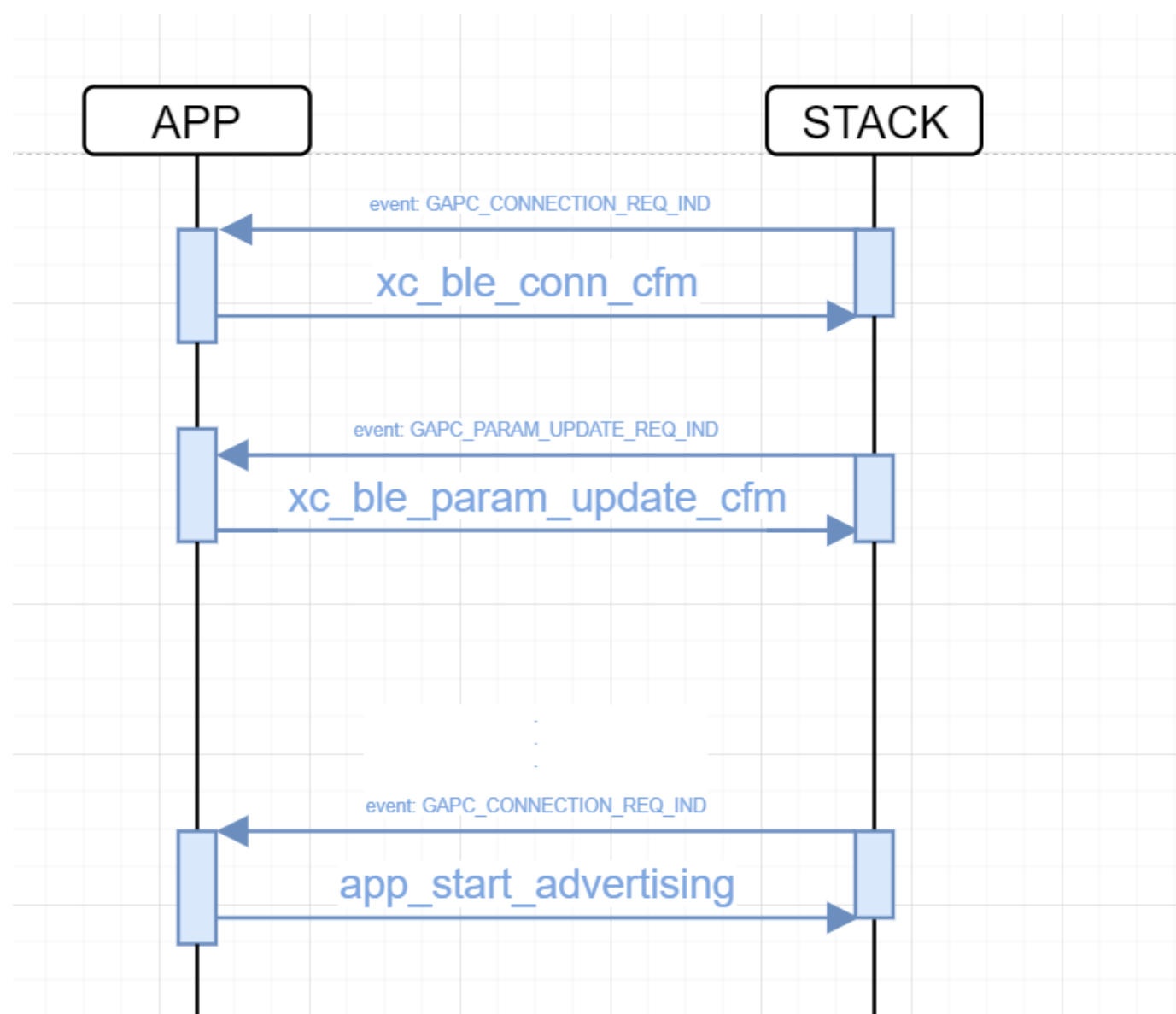
```
int main(void)
{
    board_init();
    bluetooth_init();
    while(1)
    {
        // schedule all pending events
        rwip_schedule();
        xc_ota_schedule();
        #if (SLEEP_ENABLE)
        sleep_schedule();
        #endif // (SLEEP_ENABLE)
    }
}
```

step 1：ble初始化时调用函数xc_ble_gapm_reset复位协议栈，应用层gapm_cmp_evt函数中会收到协议栈复位完成的消息，之后配置蓝牙设备信息，添加服务,开启广播等操作,流程如下：



如上设备发送广播，等待主设备连接

step 2：主机扫描到从机的广播后，会发起连接，发现服务，更新连接参数等，断开连接后从机会发起广播等待主机重连，流程如下：



step 3：主机和从机建立连接后，进行数据交互。主机可以向从机读写数据，从机可以向追加发送数据。主机向从机写数据时，从机从custom_svc_cb_att_val_set回调函数中接收到主机写的的数据。主机向从机读数据时，从机从custom_svc_cb_att_read_get回调函数中将数据发送给主机。从机调用函数slave_send_data向主机发送数据。流程图如下：

