

ble_central.md

1. Introduction

1.1 Demo Function

此工程实现了设备作为中心 (central) 的功能,

程序主要流程为：

- 设备初始化
- 设备开启扫描, 调用函数 `app_create_scan`, `app_start_scan` 实现。
- 收到 `TARGET_DEVICE_ADDR` 地址的广播包后，调用函数 `app_start_init` 与外设建立连接。
- 连接建立完成后，注册 `client` 服务, 调用函数 `app_cli_register`
- 发现`ble_central` 工程的透传服务 (`CUSTOM_SVC_UUID`)，调用函数 `xc_ble_gatt_cli_discover_svc`
- 获取透传服务 (`CUSTOM_SVC_UUID`) 的handle, 并保存在 `cli_tx_hdl`, `cli_chg_ccc_hdl` 变量中。
- 在服务发现完成后进行写操作，调用函数 `send_data`。
- 在服务发现完成后 使能从机的 `notify`，调用 `xc_ble_gatt_cli_write`。

工程中的配置项说明：

- `BOND_TEST`: 用于对从机发起配对，并进行绑定。
- `MULTI_ROLE_TEST`: 用于实现主从一体的功能。

注意：此工程也支持主从一体的功能

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

其中 `application` 目录下每个文件功能如下表：

File Name	Description
<code>app_task.c</code>	服务添加，发送广播，建立连接等逻辑
<code>usr_client.c</code>	用于与 <code>ble_peripheral</code> 的透传服务 (<code>usr_server</code>) 进行读写操作等
<code>sleep.c</code>	实现蓝牙低功耗的逻辑
<code>arch_main.c</code>	主程序 <code>main</code> 函数入口，负责事件调度，休眠逻辑等
<code>app_sec.c</code>	处理设备配对相关逻辑

File Name	Description
usr_server.c	实现了透传的交互逻辑

1.3 环境准备

- 设备烧写并运行 `ble_central` 工程作为中心设备.
- 设备烧写并运行 `ble_peripheral` 工程作为从设备.

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, 打开扫描 流程如下：

step 2：主机扫描到从机的广播后，匹配广播地址，匹配成功后会发起连接，发现透传服务的 `handle` 值，并使能外设的 `notify`.

step 3：接收从机的 `notify` 数据，对外设服务的 `CUSTOM_SVC_RX_CHAR_UUID` 特征值进行写操作。