

青风带你玩蓝牙 nRF52 系列教程.....	2
-----作者: 青风.....	2
作者: 青风.....	3
出品论坛: www.qfv8.com	3
淘宝店: http://qfv5.taobao.com	3
QQ 技术群: 346518370.....	3
硬件平台: 青云 QY-nRF52 开发板.....	3
2.23 蓝牙信号强度 rssi 的获取.....	3
1: nRF52 蓝牙 BLE 的 rssi 获取:	3
1.1 BLE 定时器声明.....	3
2.2 定时器开始定时.....	5
2.3 主函数编写.....	6
2 应用与调试.....	6
2.1 下载.....	6
2.2 测试.....	7

青风带你玩蓝牙 nRF52 系列教程

-----作者: 青风

出品论坛: www.qfv8.com 青风电子社区



作者: 青风

出品论坛: www.qfv8.com

淘宝店: <http://qfv5.taobao.com>

QQ 技术群: 346518370

硬件平台: 青云 QY-nRF52 开发板

2.31 蓝牙信号强度 rssi 的获取

RSSI 的全称为 Received Signal Strength Indication, 中文意思是接收的信号强度指示, 无线发送层的可选部分, 用来判定链接质量, 以及是否增大广播发送强度。通过接收到的信号强弱测定信号点与接收点的距离, 进而根据相应数据进行定位计算的一种定位技术。

RSSI (接收信号强度) Received Signal Strength Indicator:

例如: $Rss = 10 \log P$, 只需将接受到的信号功率 P 代入就是接收信号强度 (灵敏度)。

[例 1] 如果发射功率 P 为 1mw, 折算为 dBm 后为 0dBm。

[例 2] 对于 40W 的功率, 按 dBm 单位进行折算后的值应为:

$10 \lg (40W/1mw) = 10 \lg (40000) = 10 \lg 4 + 10 \lg 10 + 10 \lg 1000 = 46 \text{dBm}$ 。

首先我们需要知道的是无线信号 dbm 都是负数, 最大是 0。因此测量出来的 dbm 值肯定都是负数。因为 dbm 值只在一种情况下为 0, 那就是在理想状态下经过实验测量的结果, 一般我们认为 dbm 为 0 是其最大值, 意味着接收方把发射方发射的所有无线信号都接收到了, 即无线路由器发射多少功率, 接收的无线网卡就获得多少功率。当然这是在理想状态下测量的, 在实际中即使将无线网卡挨着无线路由器的发射天线也不会达到 dbm 为 0 的效果。所以说测量出来的 dbm 值都是负数, 不要盲目的认为负数就是信号不好。

本例在匹配的 SDK15.0 的蓝牙串口例子基础上就行编写, 目标是通过从机获取连接的主机的信号强度。

1: nRF52832 蓝牙 BLE 的 rssi 获取:

1.1 BLE 定时器声明

本例在 SDK15 下的串口蓝牙例子下进行修改, 其实定时器在协议栈下的使用, 首先设置一个定时器, 函数如下:

```
//定时器初始化
static void timers_init(void)
{
    uint32_t err_code;
    // 初始化定时器, 给一个定时器空间
    APP_TIMER_INIT(APP_TIMER_PRESCALER, APP_TIMER_OP_QUEUE_SIZE, false);
    //创建一个定时, 设置定时器模式
    err_code=app_timer_create(&m_rssi_timer_id, APP_TIMER_MODE_REPEATED,
rssi_timeout_handler);
    APP_ERROR_CHECK(err_code);
}
```

在 `app_timer_create` 函数中, `rssi_timeout_handler` 作为定时器函数创建的一个定时器超时中断处理, 需要处理的是里只需要执行 `rssi` 更新或者打印:

```
static void TIME_update(void)
{
    int8_t rssi=0;
    uint8_t p_ch_index;
    //
    sd_ble_gap_rssi_start(m_conn_handle, BLE_GAP_RSSI_THRESHOLD_INVALID, 0);
    //
    sd_ble_gap_rssi_get(m_conn_handle, &rssi, &p_ch_index);
    NRF_LOG_INFO("rssi: %d\r\n", rssi);
    NRF_LOG_INFO("Channel: %d\r\n", p_ch_index);
}

static void TIME_timeout_handler(void * p_context)
{
    UNUSED_PARAMETER(p_context);
    TIME_update();
}
```

我们要开始 `rssi` 的信号采集, 需要知道调用两个 SD 函数:

```
sd_ble_gap_rssi_start(m_conn_handle, BLE_GAP_RSSI_THRESHOLD_INVALID, 0);
sd_ble_gap_rssi_get(m_conn_handle, &rssi, &p_ch_index);
```

具体说明如下:

函数: `sd_ble_gap_rssi_start(m_conn_handle, BLE_GAP_RSSI_THRESHOLD_INVALID, 0);`

功能: 开始向应用程序报告接收到的信号强度。当 `RSSI` 值发生变化时, 将报告一个新的事件直到函数 `sd_ble_gap_rssi_stop` 被调用。

产生事件 `BLE_GAP_EVT_RSSI_CHANGED`, 新的 `RSSI` 数据就可以用了。事件的生成频率取决于 `<code>threshold_dbm</code>` and `<code>skip_count</code>` 输入参数。}

参数: `conn_handle` 连接句柄。

参数: [in] threshold_dbm 在触发 BLE_GAP_EVT_RSSI_CHANGED 事件之前最小的变化 dBm 值。如果 threshold_dbm 等于参数 BLE_GAP_RSSI_THRESHOLD_INVALID, 事件将会被关闭。

参数: [in] skip_count 在发送一个新的参数 BLE_GAP_EVT_RSSI_CHANGED 事件之前, 更改了一个或多个 threshold_dbm 的 RSSI 样本的数量值。

返回值:NRF_SUCCESS 成功激活 RSSI 报告。

返回值:NRF_ERROR_INVALID_STATE RSSI 报告已经在进行中。

返回值:BLE_ERROR_INVALID_CONN_HANDLE 提供的连接句柄无效。

函数: sd_ble_gap_rssi_get(m_conn_handle, &rssi,&p_ch_index);

功能: 获取最后一个连接事件的接收信号强度。在使用这个函数之前, 必须调用 sd_ble_gap_rssi_start 来开始报告 RSSI。直到在调用函数 sd_ble_gap_rssi_start 之后第一次采样 RSSI, 将返回 NRF_ERROR_NOT_FOUND。

参数: conn_handle 连接句柄。

参数: [out] p_rssi 指向应该存储 RSSI 测量值的位置的指针。

参数: [out] p_ch_index 指向 RSSI 测量通道索引的存储位置的指针。

返回值:NRF_SUCCESS 成功读取 RSSI。

返回值:NRF_ERROR_NOT_FOUND 没有例子被发现。

返回值:NRF_ERROR_INVALID_ADDR 提供无效的指针

返回值:BLE_ERROR_INVALID_CONN_HANDLE 提供的连接句柄无效。

返回值:NRF_ERROR_INVALID_STATE RSSI reporting is not ongoing.

当主机设备和我们设备连接后, 我们直接通过串口 LOG 打印主机 rssi 信号强度的值:

```
NRF_LOG_INFO("rssi: %d\r\n",rssi);
```

同时打印 RSSI 测量通道 NRF_LOG_INFO("Channel: %d\r\n",p_ch_index);

或者我们如何告知手机, 这里面直接用串口上传函数 **ble_nus_string_send**, 因为 rx 被定义了通知类型, 所以数据会被上传到手机通知中, 当手机通知使能就可以观察到变化数字。

1.2 定时器开始定时

定时器开始定时, 开始定时后设置时间间隔, 规定对应时间内执行超时中断操作, 具体代码如下:

```
//开始定时开始定时
static void application_timers_start(void)
{
    // Start application timers.定时时间间隔
    uint32_t err_code;
    err_code = app_timer_create(&m_timer_rssi_id,
                                APP_TIMER_MODE_REPEATED,
                                TIME_timeout_handler);
    APP_ERROR_CHECK(err_code);
}
```

```
}
```

1.3 主函数编写

主函数写一个测试函数，主要是定时器初始化和开始定时更新获得的 rssi 信号强度，编写代码如下：

```
int main(void)
{
    bool erase_bonds;

    // Initialize.
    log_init();
    timers_init();
    buttons_leds_init(&erase_bonds);
    power_management_init();
    ble_stack_init();
    gap_params_init();
    gatt_init();
    advertising_init();
    services_init();
    conn_params_init();
    peer_manager_init();

    // Start execution.
    NRF_LOG_INFO("Template example started.");
    application_timers_start();

    advertising_start(erase_bonds);
    // Enter main loop.
    for (;;)
    {
        idle_state_handle();
    }
}
```

修改后编译通过，提示 OK

2 应用与调试

2.1 下载

协议和应用程序下载成功后提示如图，程序开始运行，同时开发板上广播 LED 开始广播：



2.2 测试

本实验采用手机 nrf connect app 软件连接开发板设备, 同时打开串口助手, 串口助手设置如下图所示, 波特率为 115200, 会观察到输出的 RSSI 信号强度和测试频道:

