青风带你玩蓝牙 nRF52832 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-nRF52832 开发板

9 蓝牙协议栈下定时器的使用

在蓝牙下定时器会被经常使用，如果你需要按照一定时间后去执行么么任务，或者定时循环执行任务，定时器就成了必须,我们单独列出来讲一讲，因此，我们特意做了一个演示实例。

本例在匹配的 SDK15.0 的蓝牙串口样例基础上就行编写，使用的协议栈为：s132。

1：nRF51822 蓝牙 BLE 定时器设置：

1.1 BLE 定时器声明

本例在 SDK15.0 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;
    // Initialize timer module, making it use the scheduler.初始化定时器，给一个定时器空间
    ret_code_t err_code = app_timer_init();
    APP_ERROR_CHECK(err_code);

    //创建一个定时，设置定时器模式
    err_code = app_timer_create(&m_timer_id,APP_TIMER_MODE_REPEATED,
                                TIME_timeout_handler);
    APP_ERROR_CHECK(err_code);
}}
```

◎APP_TIMER_INIT 函数，深入到其内部，在函数中使用 rtc1_init 初始化调用 RTC1 作为软件定时器的时钟，也就是说软件定时器实际上采用的是实时时钟 RTC1。

◎上面 app_timer_create 函数，函数源码说明下：

1.&m_timer_id 为我们声明的定时器 ID，如果你要使用多个定时器，你就可以定义不同的 ID 就可以了，这个指向你所定义的定时器 ID。

2.APP_TIMER_MODE_REPEATED:定时器模式：有两种模式，如下说明：

```
/**@brief Timer modes. */
typedef enum
{
    APP_TIMER_MODE_SINGLE_SHOT,                /**< The timer will expire only once. */
    APP_TIMER_MODE_REPEATED                    /**< The timer will restart each time it expires. */
} app_timer_mode_t;
```

一个是定时一次，一个是重复定时。我们这里面选择重复定时。

3.TIME_timeout_handler：这个函数是创建一个定时器超时中断处理函数，需要处理的是里只需要执行电池更新函数：

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

声明完定时器后需要开始运行定时器，设置定时器运行更新的时间间隔，函数如下所示，我们简单演示，每一个时间间隔数字++一次。

```
static void TIME_update(void) //上传数据给手机，本例程是定时器自增一 ，2 秒更新一次
{
    uint32_t err_code;
    uint8_t   TIME_level;

    TIME_level=TIME;
    TIME++;
    err_code = ble_nus_string_send(&m_nus,&TIME_level, 1);
    if ((err_code != NRF_SUCCESS) &&
        (err_code != NRF_ERROR_INVALID_STATE) &&
        (err_code != BLE_ERROR_NO_TX_BUFFERS) &&
        (err_code != BLE_ERROR_GATTS_SYS_ATTR_MISSING)
    )
    {
        APP_ERROR_HANDLER(err_code);
    }
}
```

数字变化后我们如何告知手机？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。

2.2 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)
{
    uint32_t err_code;

    // Start application timers.定时时间间隔
    err_code = app_timer_start(m_timer_id, TIME_LEVEL_MEAS_INTERVAL, NULL);
    APP_ERROR_CHECK(err_code);
}
```

具体说明一下 app_timer_start 函数，这个函数第一个参数 ID 就是我们前面指定的定时器 ID，第二个参数就是时间间隔，第三个参数没有返回值就选 NULL.

宏定义一下定时器间隔，这个函数官方给出，定时 ms 级别：

```
#define TIME_LEVEL_MEAS_INTERVAL    APP_TIMER_TICKS(2000, APP_TIMER_PRESCALER)
```

2.3 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，编写代码如下：

```
int main(void)
{
    uint32_t err_code;
    bool erase_bonds;
    uint8_t  start_string[] = START_STRING;

    // Initialize.
    APP_TIMER_INIT(APP_TIMER_PRESCALER,     APP_TIMER_OP_QUEUE_SIZE,
false);
    uart_init();
    timers_init();//需要添加的部分
    buttons_leds_init(&erase_bonds);
    ble_stack_init();
    gap_params_init();
```

```
    services_init();
    advertising_init();
    conn_params_init();

    err_code = ble_advertising_start(BLE_ADV_MODE_FAST);
    APP_ERROR_CHECK(err_code);
    application_timers_start();//需要添加的部分
    // Enter main loop.
    for (;;)
    {
        power_manage();
    }
}
```

修改后编译通过，提示 OK

2 应用与调试

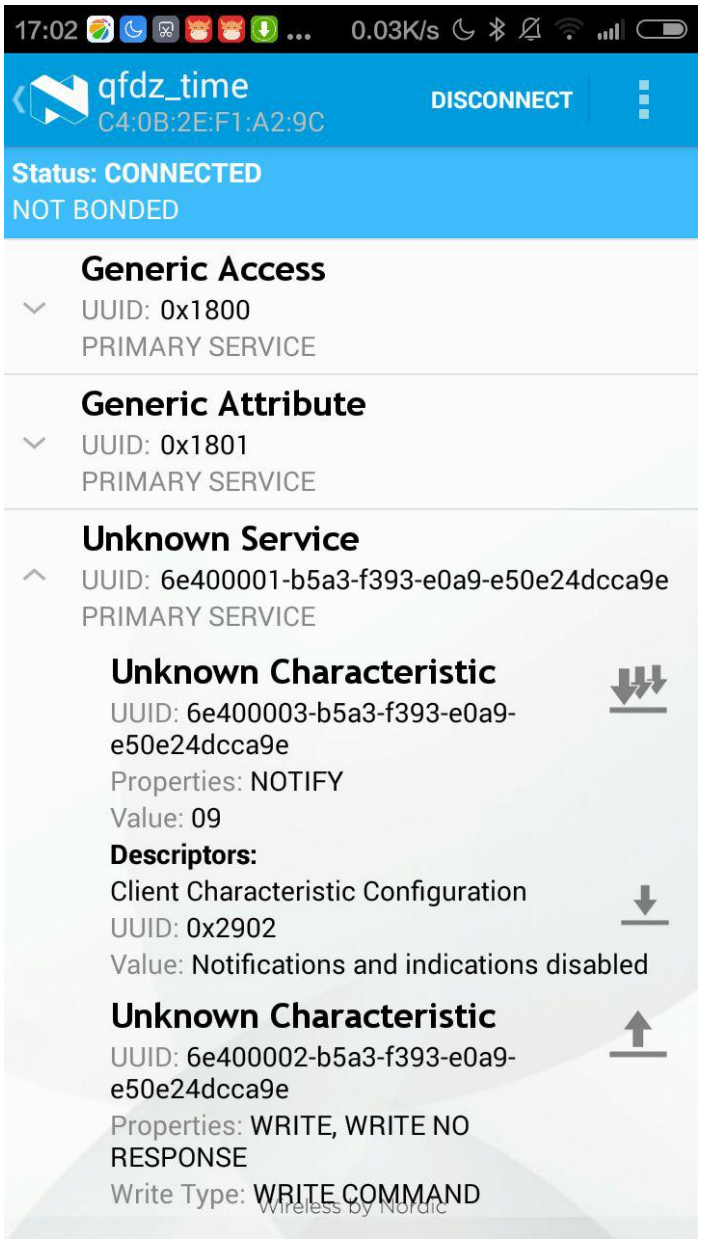
2.1 下载

打开 NRFgo 进行下载，，首先整片擦除，后下载协议栈，下载完后可以下载工程，首先把工程编译一下，通过后点击 KEIL 上的下载按键。下载成功后，程序开始运行，同时开发板上广播 LED 开始广播。

2.2 测试

本实验采用手机写 nrf connect app 软件，返现服务应用名 QFDZ_time,如下图所示：

点击打开，找到 Service，如下图所示：



特性：通知类型：点击通知使能，可以比较变化值,参数值会定时加 1：