

青风带你玩蓝牙 nRF52832 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-nRF52832 开发板.....	3
2.5 蓝牙 BLE 遥控器.....	3
1: 遥控指令设置:	3
2 应用与调试.....	4
2.1 下载.....	4
2.2 测试.....	5

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**

23 蓝牙 BLE 遥控器

很多朋友和客户希望能够把通过蓝牙发起指令控制硬件, 实现遥控器的功能, 实际上大体思路有两种方法, 方法一: 使用蓝牙串口工程, 发送指令 (类似 AT 指令) 后通过蓝牙串口 APP 实现远程控制, 本章主要介绍这种方法。第二种方法, 采用类似蓝牙按键通知这章的内容, 建立一个私有任务, 通过通知的按键进行控制, 这个前面章节已经有讲述。本下面就主要介绍方法一进行设置开发。

注意本例在上一节的蓝牙串口的基础上进行修改。

1: 遥控指令设置:

为了简单演示遥控功能, 我们采用遥控点亮 LED 灯来验证。设置 AT 指令判断执行什么操作, 代码配置如下:

```
01. static void nus_data_handler(ble_nus_evt_t * p_evt)//串口中断操作
02. {
03.     if (p_evt->type == BLE_NUS_EVT_RX_DATA)
04.     {
05.         uint32_t err_code;
06.
07.         NRF_LOG_DEBUG("Received data from BLE NUS. Writing data on UART.");
08.         NRF_LOG_HEXDUMP_DEBUG(p_evt->params.rx_data.p_data,
09.                                p_evt->params.rx_data.length);
10.         //首先判断输入的头是否为 AT+指令
11.         if((p_evt->params.rx_data.p_data[0]=='A')
12.            &&(p_evt->params.rx_data.p_data[1]=='T')
13.            &&(p_evt->params.rx_data.p_data[2]=='+'))
14.         {
15.
```

```
16.     if(((p_evt->params.rx_data.p_data[3]=='O')||(p_evt->params.rx_data.p_data[3]=='o'))
17.     &&((p_evt->params.rx_data.p_data[4]=='N')||(p_evt->params.rx_data.p_data[4]=='n')))
18.     //如果收到的数据为“ON”或者"on"
19.     {
20.         nrf_gpio_pin_clear(LED_2);    //工作指示绿灯亮
21.         nrf_gpio_pin_set(LED_3);      //工作指示红灯灭
22.         states[0]=p_evt->params.rx_data.p_data[0];          //更新要上传的状态数据
23.         states[1]=p_evt->params.rx_data.p_data[1];
24.         states[2]=0;
25.         change_state=1;          //状态发生变化，马上上传状态一次
26.     }
27.
28.     if(((p_evt->params.rx_data.p_data[3]=='O')||(p_evt->params.rx_data.p_data[3]=='o'))
29.     &&((p_evt->params.rx_data.p_data[4]=='F')||(p_evt->params.rx_data.p_data[4]=='f'))
30.     &&((p_evt->params.rx_data.p_data[5]=='F')||(p_evt->params.rx_data.p_data[5]=='f')))
31.     {
32.         //如果收到的数据为“OFF”或者"off"
33.         nrf_gpio_pin_clear(LED_3);    //工作指示红灯亮
34.         nrf_gpio_pin_set(LED_2);      //工作指示绿灯灭
35.         states[0]=p_evt->params.rx_data.p_data[0];          //更新要上传的状态数据
36.         states[1]=p_evt->params.rx_data.p_data[1];
37.         states[2]=p_evt->params.rx_data.p_data[2];
38.         change_state=1;          //状态发生变化，马上上传状态一次
39.     }
40. }
41.     if (p_evt->params.rx_data.p_data[p_evt->params.rx_data.length - 1] == '\r')
42.     {
43.         while (app_uart_put('\n') == NRF_ERROR_BUSY);
44.     }
45.
46. }
47.
48. }
```

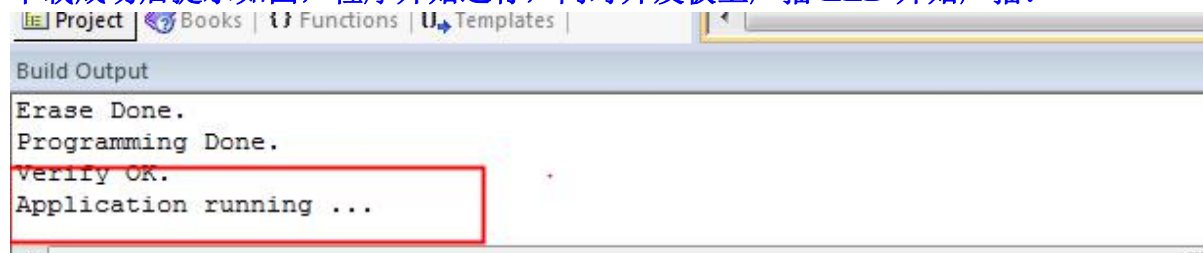
分析上面的代码，通过 APP 发给开发板数据指令，我们通过判断指令数据来执行相关的操作，比如如果发了 AT+ON，我们就执行开灯功能。如果执行的 AT+off，我们则执行了关掉灯的功能。当然还有更多的操作可以执行，比如控制继电器开关，实现智能开关的功能等等。在蓝牙接收 nus_data_handler 函数中，判断主机发过来的数据指令，来执行对应的操作，原理比较简单。

2 应用与调试

2.1 下载

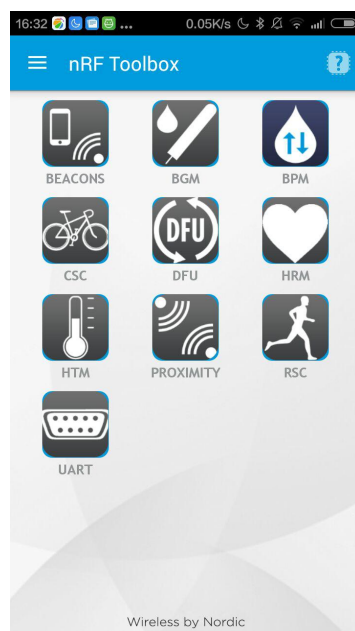
连接好开发板和仿真器后, 首先下载协议栈, 本例使用的协议栈为 S132 版本, 打开 NRFgo 进行下载, 可以参考前面样例章节介绍。首先整片擦除, 后下载协议栈:

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

打开手机 APP 软件 TOOLbox, 打开串口 uart 功能, 如下图所示:



下载完成应用代码后, 按下开发板复位按键运行程序。然后打开 app 如下图所示, 发现串口, 点击连接:



连接成功如下图所示:



发送和接收数据, 发送 AT+on 控制 led 灯开, 发送 AT+off 控制 led

灯关。如下图所示:

