

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
2.29 蓝牙 BLE 温湿度检测方法一.....	3
1: 温湿度采集:	3
1.1 温湿度 DHT11 采集驱动.....	3
2: 协议栈下实现流程:	6
2.1 初始化传感器.....	6
2.2 采集指令发送.....	8
3 应用与调试.....	9
3.1 下载.....	9
3.2 测试.....	10

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

2.27 蓝牙 BLE 温湿度检测方法一

很多朋友和客户希望能够把通过蓝牙检测环境温湿度,并且通过手机接收温湿度参数。为智能家居做必要的准备。本章将来讨论蓝牙温湿度采集,使用温湿度模块 DHT11.

那么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)后开始采集温湿度,温湿度通过蓝牙串口 APP 接收,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,通过通知的内容显示温湿度。这种方法将在后面的原创教程中进行讲解。

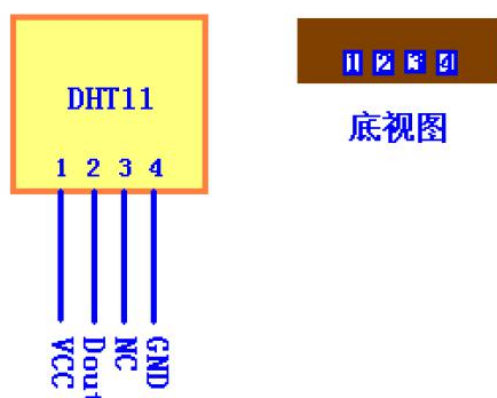
这里我们通过一个简单的例子:蓝牙 BLE 温湿度采集,来进行一个简单的思路验证。注意本例在蓝牙串口的基础上进行修改。

1: 温湿度采集:

1.1 温湿度 DHT11 采集驱动

DHT11 是一款湿温度一体化的数字传感器。该传感器包括一个电阻式测湿元件和一个 NTC 测温元件,并与一个高性能 8 位单片机相连接。通过单片机等微处理器简单的电路连接就能够实时的采集本地湿度和温度。DHT11 与单片机之间能采用简单的单总线进行通信,仅仅需要一个 I/O 口。传感器内部湿度和温度数据 40Bit 的数据一次性传给单片机,数据采用校验和方式进行校验,有效的保证数据传输的准确性。DHT11 功耗很低,5V 电源电压下,工作平均最大电流 0.5mA。

DHT11 的管脚排列如图:



虽然 DHT11 与 DS18B20 类似, 都是单总线访问, 但是 DHT11 的访问, 相对 DS18B20 来说要简单很多。下面我们先来看看 DHT11 的数据结构。

DHT11 数字湿温度传感器采用单总线数据格式。即, 单个数据引脚端口完成输入输出双向传输。其数据包由 5Byte (40Bit) 组成。数据分小数部分和整数部分, 一次完整的数据传输为 40bit, 高位先出。DHT11 的数据格式为: 8bit 湿度整数数据+8bit 湿度小数数据+8bit 温度整数数据+8bit 温度小数数据+8bit 校验和。其中校验和数据为前四个字节相加。

传感器数据输出的是未编码的二进制数据。数据(湿度、温度、整数、小数)之间应该分开处理。例如, 某次从 DHT11 读到的数据如图:

byte4	byte3	byte2	byte1	byte0
00101101	00000000	00011100	00000000	01001001
整数	小数	整数	小数	校验和
湿度		温度		校验和

由以上数据就可得到湿度和温度的值, 计算方法:

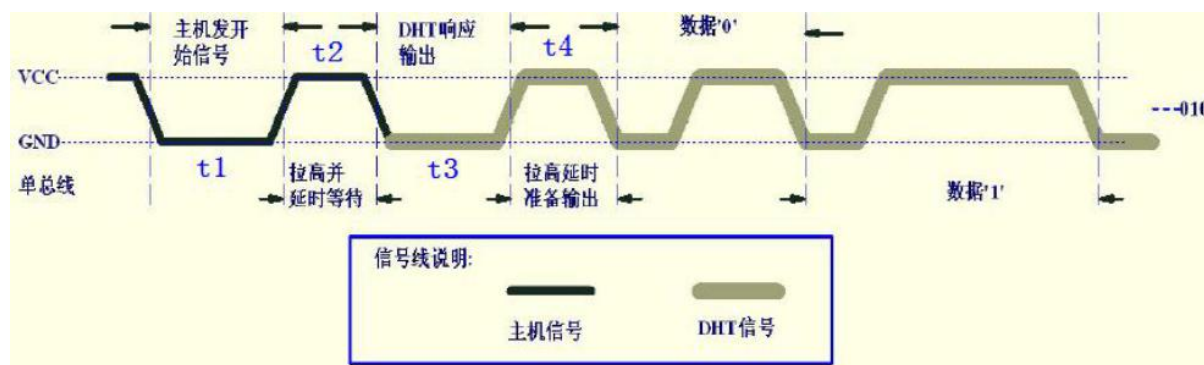
湿度 = byte4 . byte3 = 45.0 (%RH)

温度 = byte2 . byte1 = 28.0 (°C)

校验 = byte4 + byte3 + byte2 + byte1 = 73 (=湿度 + 温度)(校验正确)

可以看出, DHT11 的数据格式是十分简单的, DHT11 和 MCU 的一次通信最大为 3ms 左右, 建议主机连续读取时间间隔不要小于 100ms。

下面, 我们介绍一下 DHT11 的传输时序。DHT11 的数据发送流程如图



首先主机发送开始信号,即:拉低数据线,保持 t1(至少 18ms)时间,然后拉高数据线 t2(20~40us)时间,然后读取 DHT11 的相应,正常的话,DHT11 会拉低数据线,保持 t3(40~50us)时间,作为响应信号,然后 DHT11 拉高数据线,保持 t4(40~50us)时间后,开始输出数据。代码配置如下:

```
u8 dht_Read_Bit(void) //读 bit 位的顺序
{
    u8 time=0;
    while(dht_data_IN&&time<100)
    {
        time++;
        nrf_delay_us(1);
    }
    time=0;
    while(!dht_data_IN&&time<100)
    {
        time++;
        nrf_delay_us(1);
    }
    nrf_delay_us(40);
    if(dht_data_IN)return 1;
    else return 0;
}

u8 dht_Read_Byte(void) //读一个字节
{
    u8 i,dat;
    dat=0;
    for (i=0;i<8;i++)
    {
        dat<<=1;
        dat|=dht_Read_Bit();
    }
    return dat;
}

u8 dht_Read_Data(u8 *temp,u8 *humi) //读取整个温湿度数据
{
    u8 dat[5];
    u8 i;
    dht_Rst();
    if(dht_Check()==0)
    {
        for(i=0;i<5;i++)
        {
```

```
        dat[i]=dht_Read_Byte();
    }
    if((dat[0]+dat[1]+dat[2]+dat[3])==dat[4])
    {
        *humi=dat[0];
        *temp=dat[2];
    }
    }else return 1;
    return 0;
}
```

温湿度传感器的驱动，实际上是简单的通过 IO 端口模拟时序，读取当前测试的温湿度数据，然后把数据保存到数据指针内。

2：协议栈下实现流程：

2.1 初始化传感器

本例以蓝牙串口工程为框架，在主函数中，只需要修改一个位置，也就是加入传感器初始化。代码如下，检查是否传感器加入成功，如果初始化温湿度传感器不成功，则设备不会广播：

```
int main(void)
{
    uint32_t err_code;
    bool erase_bonds;
    uint8_t start_string[] = START_STRING;
    while(dht_Init())
    {
        nrf_delay_ms(1000);
    }
    // Initialize.
    APP_TIMER_INIT(APP_TIMER_PRESCALER, APP_TIMER_OP_QUEUE_SIZE,
false);
    uart_init();
    buttons_leds_init(&erase_bonds);
    ble_stack_init();
    gap_params_init();
    services_init();
    advertising_init();
    conn_params_init();

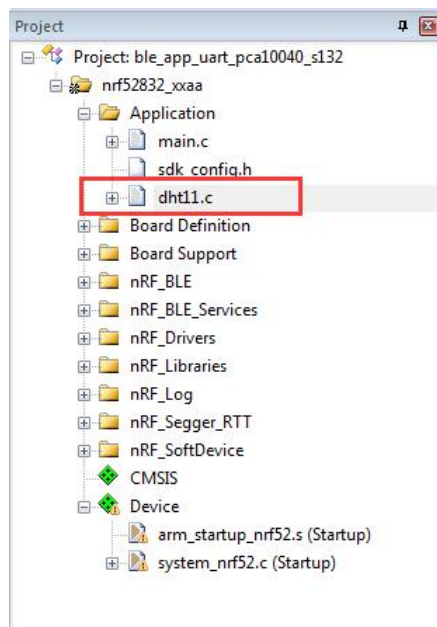
    printf("%s",start_string);
}
```



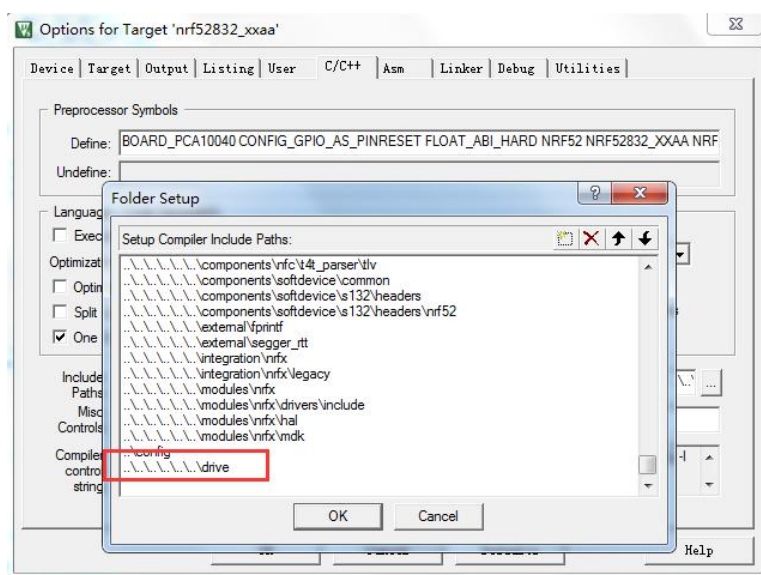
```
err_code = ble_advertising_start(BLE_ADV_MODE_FAST);
APP_ERROR_CHECK(err_code);

// Enter main loop.
for (;;)
{
    power_manage();
}
}
```

同时在工程中，加入我们写的温湿度传感器的驱动函数，如下图所示：



同时需要添加工程文件路径：



2.2 采集指令发送

蓝牙串口发送采集指令, 通过命令判断是否开始采集, 类似于 AT 指令, 然后再通过 **BLE 串口透传通道** 温湿度模块采集到的数据上传给你的主机。代码编写如下:

```
void ble_send(uint8_t * string,uint16_t length)
{
    uint32_t err_code;

    err_code = ble_nus_data_send(&m_nus, string,&length,m_conn_handle);//把采集
    的数据通过 BLE 串口透传通道上传给主机

    if (err_code != NRF_ERROR_INVALID_STATE)
    {
        APP_ERROR_CHECK(err_code);
    }
}
```

发送 ON 指令控制温湿度模块开始采集, 采集后的数据发送给手机, 蓝牙部分的功能设置请具体参考《蓝牙串口详解》, 代码如下:

```
static void nus_data_handler(ble_nus_t * p_nus, uint8_t * p_data, uint16_t length)
{
    uint8_t data[4],w,s;
    if (p_evt->type == BLE_NUS_EVT_RX_DATA)
    {
        NRF_LOG_DEBUG("Received data from BLE NUS. Writing data on UART.");
        NRF_LOG_HEXDUMP_DEBUG(p_evt->params.rx_data.p_data,
        p_evt->params.rx_data.length);

        if(((p_evt->params.rx_data.p_data[0]=='O')||(p_evt->params.rx_data.p_data[0]=='o'))&&((p_e
        vt->params.rx_data.p_data[1]=='N')||(p_evt->params.rx_data.p_data[1]=='n')))
            //如果收到的数据为“ON”或者"on"
            {

                data[0]=0x33; //首先给一个头
                if(dht_Read_Data(&w,&s)) //如果没有读取到数据则发送 0
                {
                    data[1]=0x00;
                    data[2]=0x00;
                    ble_send(data,3);
                }
            }
    }
}
```



```
        else
        {
            data[1]=w;
            data[2]=s;//读取到了数据后，把读取的数据上传主机
            ble_send(data,3);
        }

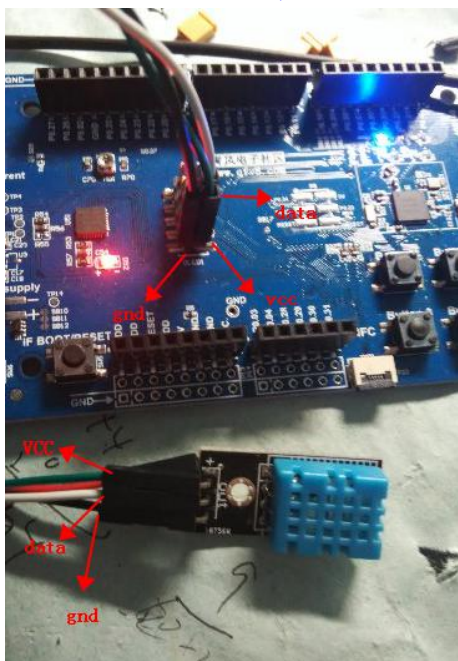
    }

    if (p_evt->params.rx_data.p_data[p_evt->params.rx_data.length - 1] == '\r')
    {
        while (app_uart_put('\n') == NRF_ERROR_BUSY);
    }
}
}
```

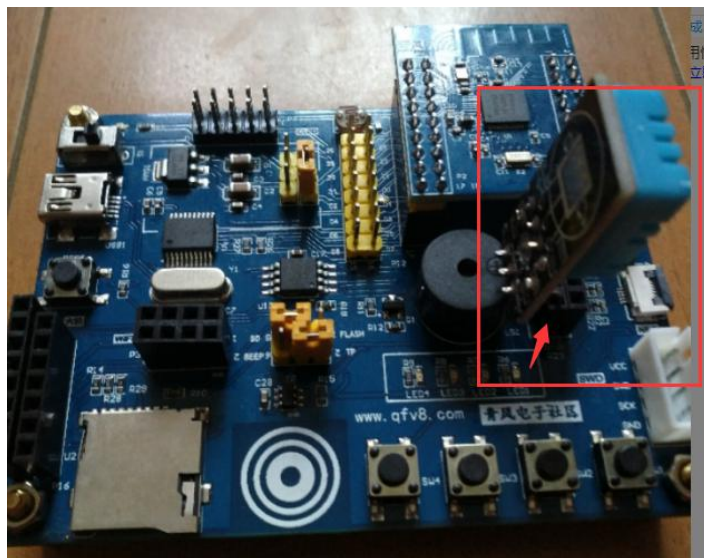
3 应用与调试

3.1 下载

开发板接上传感器，如下图连接，连接传感器的三个接口，DK 开发板如下所示接：



EK 开发板如下所示接：

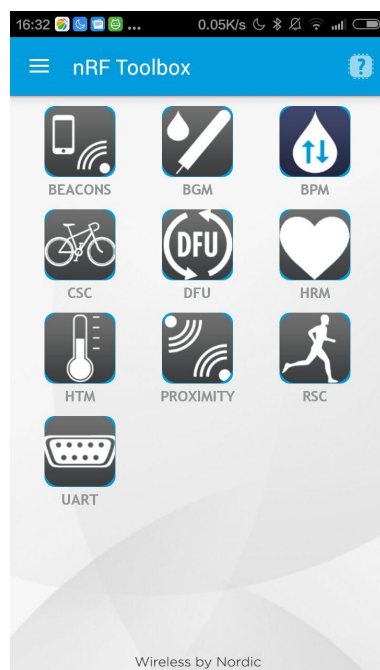


打开 NRFgo 进行下载, 首先整片擦除, 后下载协议栈: 下载完后可以下载工程, 首先把工程编译一下, 通过后点击 KEIL 上的下载按钮。下载成功后提示如图, 程序开始运行, 同时开发板上广播 LED 开始广播:

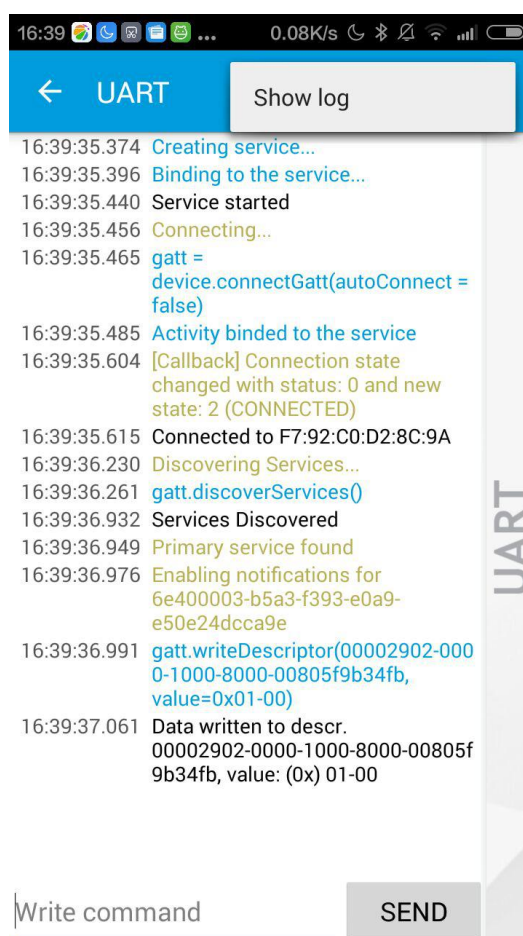


3.2 测试

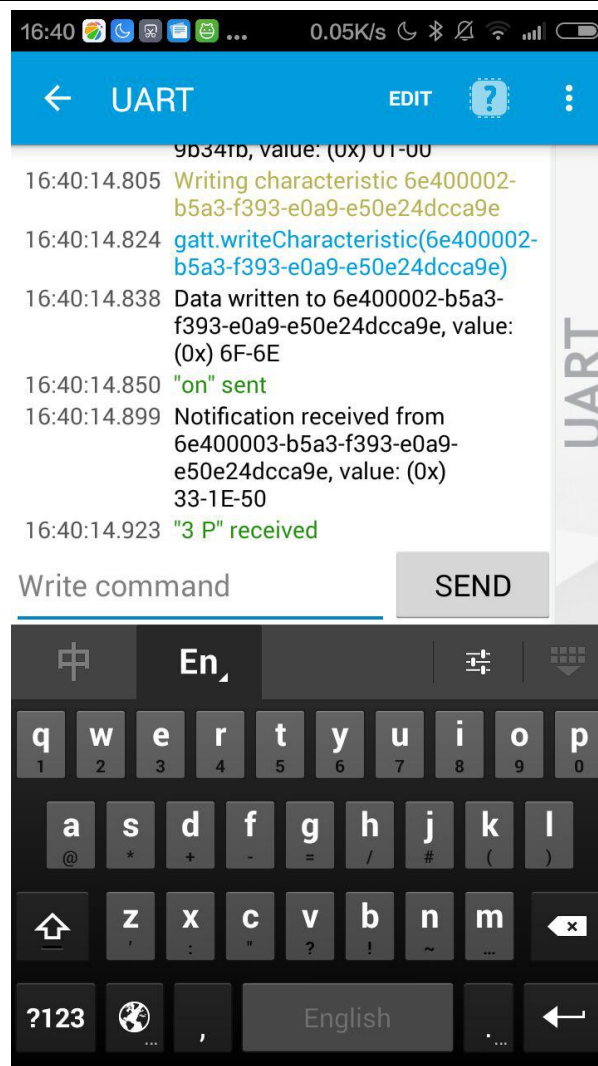
打开手机 APP 软件 TOOLbox, 打开串口 uart 功能, 如下图所示:



点击连接后如下图所示，点击修改设备 show log, 如下图所示：



填入指令 on 开始采样文档和湿度：



如上图所示，采集的 VALE 值显示如图：33-1E-50，其中 `data[0]=0x33`，`data[1]=w`；`data[2]=s`；