**M5310-A Demo 例程说明**



**重要声明**

**版权声明**

本文档中的任何内容受《中华人民共和国著作权法》的保护，版权所有 © 2017, 中移物联网有 限公司，保留所有权利，但注明引用其他方的内容除外，任何人和单位不得将该软件用于商业行为上。

**商标声明**

中移物联网有限公司和中移物联网有限公司的产品是中移物联网有限公司专有。在提及其他公司 及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。

**不作保证声明**

中移物联网有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。

**关于文档**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **修订人** |
| V1.0 | 2018.10.10 | 新增 | 杨燕鎏 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

目录

[1. M5310-A Demo 例程工程要点 5](#_Toc500151176)

[1.1 工程背景 5](#_Toc500151177)

[1.2 工程目录 5](#_Toc500151178)

[1.3 工程设置 6](#_Toc500151179)

[1.4 environment.c文件 7](#_Toc500151180)

[1.5 socket.c文件 7](#_Toc500151181)

[1.6 time.c文件的移植 8](#_Toc500151182)

[1.7 memory.c文件的移植 9](#_Toc500151183)

[1.8 模组驱动的实现 9](#_Toc500151184)

[2. M5310-A Demo例程使用要点 10](#_Toc500151185)

[2.1 网络配置 10](#_Toc500151186)

[2.2 资源配置 10](#_Toc500151187)

[2.3 终端保活 11](#_Toc500151188)

[2.4 数据上传 11](#_Toc500151189)

[2.5 下行命令处理 12](#_Toc500151190)

[2.6 处理下行ACK 12](#_Toc500151191)

[3. 常见问题 13](#_Toc500151192)

[3.1 终端无法和OneNet建立连接 13](#_Toc500151193)

[3.2 模组初始化失败 13](#_Toc500151194)

## M5310-A Demo例程工程要点

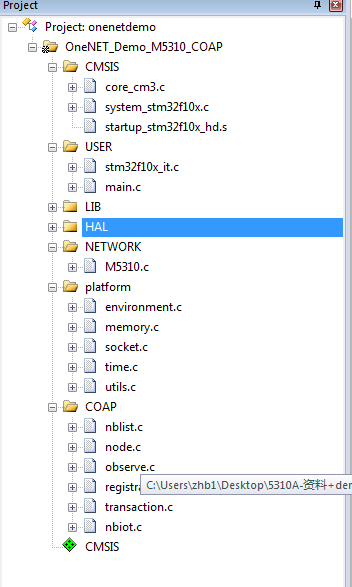
### 工程背景

开发语言：C语言

硬件平台：STM32F103RBT6

开发环境：KEIL MDK5.23

### 工程目录



* USER目录

在该目录下的main.c中实现了将温湿度传感器的值上传至OneNET平台的应用程序。

* HAL目录

在目录下实现了串口1和串口3以及各类传感器的驱动，其中串口1用于打印调试信息，串口3用于和M5310A进行通信。

* NETWORK目录

实现通信模组的驱动，这里实现了M5310A模组的驱动

* COAP目录

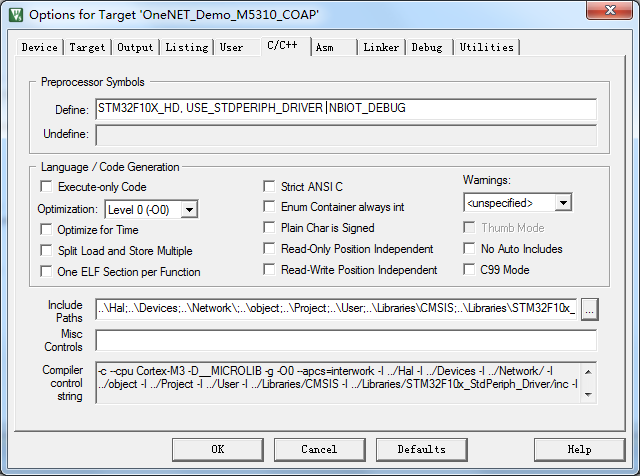
在该目录中实现了M5310A的Demo程序，该Demo程序屏蔽了M5310A的驱动细节，并向用户提供了相应的操作接口。

* Platform目录

该目录抽象了一些系统接口，方便用户对Demo程序进行移植，如内存管理和时间管理，下面将会详细对该目录下的文件进行说明。

### 工程设置

* 宏定义和头文件路径设置：



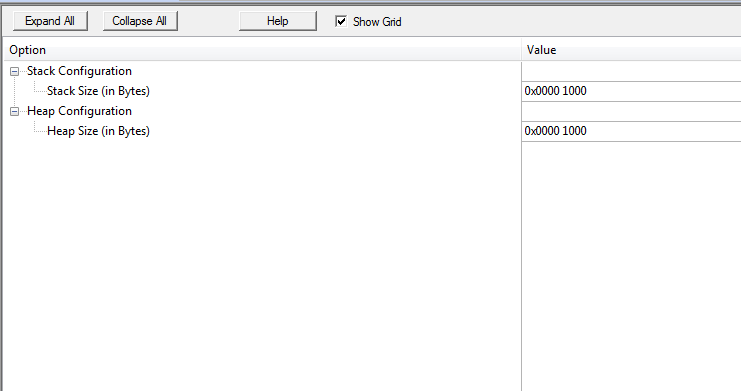
在Include Paths处添加头文件所在的路径，该路径为相对于工程所在目录的相对路径

在Preprocessor Symbols处添加条件编译选项，条件编译选项如下：

* NBIOT\_DEBUG – 输出sdk调试日志
* NOTIFY\_ACK -用户如果希望使用ACK和重传模式，可以打开该宏

注意：太多的打印会导致系统变慢，扰乱系统的时序，从而出现各种问题，因此不建议打开所有的打印。

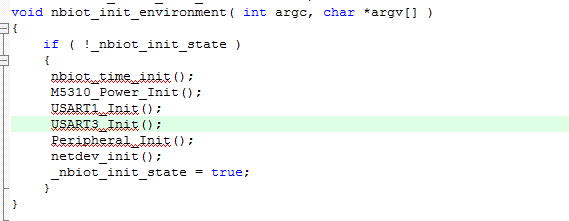
* 堆栈设置：点击startup\_stm32f10x\_hd.s启动文件，进入Configueration Wizard 配置界面，如下图所示，分别设置栈和堆的内存大小，用户可根据实际情况进行灵活设置，比如如果打开了NOTIFY\_ACK选项后，由于存在重传缓冲，则需要将堆的空间设置大些，反之设置小些；如果应用程序中建立的资源较多，也需要适当增大堆的空间，否则可能会出现内存不足或者栈溢出的现象。



### environment.c文件

* 函数
* void nbiot\_init\_environment( void )

在该函数中，初始化一切和例程运行有关的软硬件资源，包括内部时钟初始化、模组上电、打印串口初始化、模组通信串口初始化和模组工作状态初始化以及外设初始化。



* void nbiot\_clear\_environment( void )

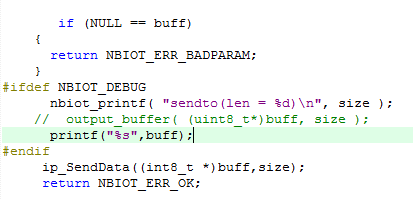
该函数中，可根据需要对当前例程的运行环境进行一些清理的工作，包括退出当前的网络，回收内存资源等动作。

### socket.c文件

* int nbiot\_udp\_send(const void \*buff,

size\_t size)

例程使用该函数，采用非阻塞方式将数据发送出去，在这里则直接调用模组的发送驱动接口ip\_SendData函数，将数据发送至模组。

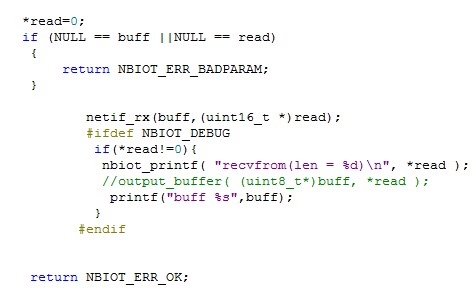


* int nbiot\_udp\_recv(void \*buff,

size\_t size,

size\_t \*read)

例程使用该函数，采用非阻塞的方式接受数据，在这里调用了netif\_rx函数，该函数负责检查接收缓存区，如果有数据则提交给例程进行处理，否则立即返回。



### time.c文件的移植

* time\_t nbiot\_time( void )

time\_t 类型为long型，nbiot\_time函数返回的是系统当前时间，单位为秒，因此使用STM32系统提供的RTC定时器作为系统滴答定时器，nbiot\_time函数读取定时器里面的值并返回。

* void nbiot\_sleep( int milliseconds )

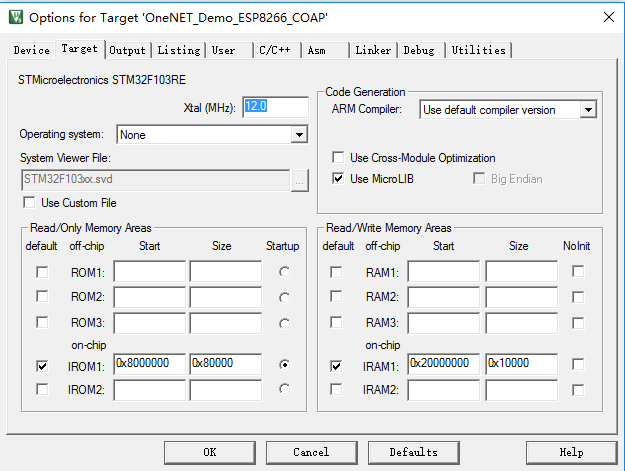
使用了RTC定时器实现了ms级别的精确延时。

### memory.c文件的移植

* void \*nbiot\_malloc( size\_t size )
* void nbiot\_free( void \*ptr )

为了管理Demo 例程运行过程的内存分配与释放，抽离出以上2个接口函数，便于不同环境下采用合适的内存管理策略，现在暂时使用KEIL 微库提供的malloc函数和free函数，需要注意以下两点：

1. 在工程配置中，需要选择Use MicroLIB这个选项



### 模组驱动的实现

在该Demo例程中已经将模组驱动封装成api接口供用户调用，用户不需要直接调用模组驱动中的函数，因此这里不详细描述驱动的实现过程。

* void netdev\_init(void)

该函数中要实现模组的上电，MCU和模组之间的串口链路的建立，模组工作状态的初始化包括无线网络的附着，ONENET服务器的连接等。

* uint32\_t ip\_SendData(int8\_t \* buf, uint32\_t len)

该函数中要根据模组提供的接口实现报文封装，并通过串口链路发送到模组中

输入参数：buf->待发送报文的内存指针，len->发送报文的长度

输出参数：无

返回值：最终发送报文的长度

* void netif\_rx(uint8\_t\*buf,uint16\_t \*read)

该函数从下行数据接收缓存区中获取模组上传的原始报文，并解封装为用户报文，上传给Demo程序进行处理。

输入参数：buf->接收到的报文的内存指针

输出参数：len->接收到的报文的长度

返回值：无

## M5310-A Demo例程使用要点

该节中将以main.c应用为例，详细描述Demo例程的使用方法，在main函数中有一个死循环，在该循环里面轮询nbiot\_device\_step函数，该函数为例程处理上下行数据的总入口。

### 资源配置

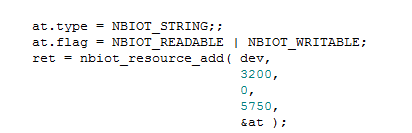
在Demo例程中，设备上所有与onenet进行交互的数据变量都需要表示为资源，比如温湿度传感器的温度和湿度可以分别表示为2个值类型为float的资源，不过在main.c函数中为了方便同时传输温湿度的值，就把这两个值转换为字符串的形式，用一个string类型的资源就可以表示了。

配置好资源属性后，设备在向服务器注册的过程中会携带相应的资源列表 , 服务器端会主动向设备订阅（Observe）资源列表中所有资源。资源的属性如下表格所示:



**注意：objid和resid是不能够随意定义的，必须遵守**《IPSO-Smart-Objects-Expansion-Pack》文档中的规定

创建资源的流程如下图所示，首先声明一个string型资源变量at，然后初始化该资源的属性，最后调用nbiot\_resource\_add接口将该资源注册进SDK中



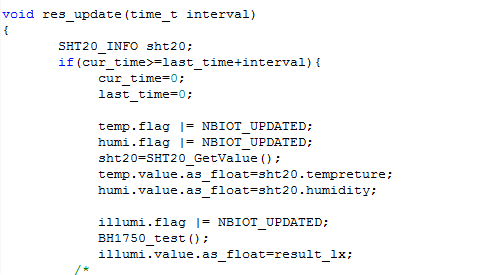
### 终端保活

用户需要在Main函数中设置lifetime的值，例程在登陆云平台的时候会将该值告知云平台，如果超时了模组将会向上提示MIPLCLOSE的信息，例程收到该信息后会重新登陆，整个过程不需要用户参与。

需要注意得是lifetime值要根据具体的应用场景设置，太短不利于模组休眠，太长则不利于平台监控设备掉线的情况，默认为300s。

### 数据上传

在某个时刻如果用户需要上传某个资源的值，只需要设置该资源的flag属性为NBIOT\_UPDATED，用来通知Demo例程该资源的值发生了变化，例程会自动将该资源的新值上传至onenet中。例如，在main.c应用程序中设置每30秒上传一次温湿度传感器的值，进入res\_update函数中，首先将temp资源的状态设置为NBIOT\_UPDATED，然后读取温湿度传感器的值.



温湿度传感值在OneNET平台上的数据呈现如下图所示:



### 下行命令处理

命令由第三方应用发送至onenet，并由云平台转发至终端设备，终端只需要将下行命令缓存起来等待Demo例程处理即可。Demo 例程从下行缓存区里面获取下行数据并正确解析后会自动调用该命令对应的资源中的write钩子函数或者execute钩子函数，用户只需要编写该2个函数即可以完成下行命令的处理。

### 处理下行ACK

用户上传数据后，往往会希望获知上传的结果，即平台是否已经正确收到了上传的数据并返回了相应的ACK，这里有一个宏定义开关NOTIFY\_ACK，如果不打开这个开关，则例程不会进行重传；如果打开这个开关，例程会自动处理下行ACK， 超时则会触发重传，在处理完下行ACK后会触发notify\_ack\_callback回调，这样用户可以在该回调函数中完成收到ACK后的后续处理。

## 常见问题

### 终端无法和OneNet建立连接

* 首先确定终端是否正常附着到NB网络中，通过“AT+CEREG?”命令查看模组的当前状态，如果返回结果包含“2:1”则说明模组已经和基站建立了通信链路，如果返回的结果错误，则需要确定物联卡是否欠费，甚至是物联卡是否已经正常开通。
* 如果确定了终端已经成功附着到NB网络中，但是终端发送的登陆信息无法传递至云平台，则很有可能是NB核心网对OneNET平台有限制，需要联系当地的运营商。

### 模组初始化失败

Demo例程在启动的时候会首先对当前模组的工作模式进行配置，使其能够附着进NB网络中，配置的方式是通过一系列AT指令，需要注意的是模组处理每条AT指令都需要一定的时间，因此建议AT指令间可以延迟几秒，否则会造成初始化过程失败。