**Nb-IOT SDK使用说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **修订人** |
| V1.0 | 2017.03.14 | 新增 | XiaoBo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 主要流程

## Step1 初始化SDK环境

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

用于初始化sdk的相关环境，比如socket、memory pool等。

## Step2 创建设备（SDK）

**int nbiot\_device\_create( nbiot\_device\_t \*\*dev,**

**uint16\_t local\_port )**

创建OneNET接入设备实例。

**参数**：

* dev - 指向设备指针的内存，用于返回设备实例指针
* local\_port - 本地绑定的udp端口号

**返回值：**

创建NB设备实例成功返回NBIOT\_ERR\_OK,否则返回对应的错误码。

## Step3 连接OneNET服务

**int nbiot\_device\_connect( nbiot\_device\_t \*dev,**

**const char \*server\_uri,**

**time\_t life\_time )**

连接OneNET服务；

此过程并非真正连接OneNET服务，而是配置接入OneNET服务的相关信息。

**参数：**

* dev - 指向OneNET接入设备实例的内存，由nbiot\_device\_create创建。
* server\_uri - OneNet接入服务的coap链接地址（例如：coap://183.230.40.40:5683）。
* life\_time – 设备的保活时长（单位：秒）。

**返回值：**

如果成功设置返回NBIOT\_ERR\_OK，否则返回对应的错误码。

## Step4 配置设备

**int nbiot\_device\_configure( nbiot\_device\_t \*dev,**

**const char \*endpoint\_name,**

**const char \*auth\_code,**

**nbiot\_resource\_t \*res\_array[],**

**size\_t res\_num )**

配置设备的资源、名称以及鉴权码等信息。

**参数：**

* dev - 指向OneNET接入设备实例的内存，由nbiot\_device\_create创建。
* endpoint\_name – 终端名称，与设备注册OneNET的设置保持一致。
* auth\_code – 设备鉴权码，与设备注册OneNET的设置保存一致。
* res\_array – 资源数组
* res\_name – 资源总数

**返回值：**

配置成功返回NBIOT\_ERR\_OK，否则返回对应的错误码。

## Step5 数据驱动以及设备保活

**int nbiot\_device\_step( nbiot\_device\_t \*dev,**

**time\_t timeout )**

驱动sdk运行以及设备保活。

**参数：**

* dev - 指向OneNET接入设备实例的内存，由nbiot\_device\_create创建。
* timeout – 执行超时时间（单位：秒）。

**返回值：**

成功返回NBIOT\_ERR\_OK，否则返回对应的错误码。

## Step6 主动上报资源数据

**int nbiot\_device\_notify( nbiot\_device\_t \*dev,**

**uint16\_t objid,**

**uint16\_t instid,**

**uint16\_t resid )**

主动上报设置资源数据。

参数：

* dev - 指向OneNET接入设备实例的内存，由nbiot\_device\_create创建。
* objid – object id（lwm2m资源模型 – 对象）
* instid – instance id（lwm2m资源模型 – 对象实例）
* resid – resource id（lwm2m资源模型 – 属性）

返回值：

成功返回NBIOT\_ERR\_OK，否则返回对应的错误码。

## Step7 销毁SDK

int nbiot\_device\_close( nbiot\_device\_t \*dev )

关闭与OneNET服务的连接。

void nbiot\_device\_destroy( nbiot\_device\_t \*dev )

销毁OneNET设备实例。

void nbiot\_clear\_environment( void )

清理sdk运行环境等。

**参数：**

* dev -指向OneNET接入设备实例的内存，由nbiot\_device\_create创建。

**返回值：**

成功返回NBIOT\_ERR\_OK，否则返回对应的错误码。

# 实例

## 主动上报资源数据

每天主动上报一次资源数据，具体代码如下：

int main( int argc, char \*argv[] )

{

/\* 初始化sdk运行环境 \*/

nbiot\_init\_environment();

do

{

int i;

int ret;

nbiot\_device\_t \*dev;

/\* 资源声明 \*/

nbiot\_resource\_t dic;

dic.objid = 3200;

dic.instid = 0;

dic.resid = 5501;

dic.type = NBIOT\_VALUE\_INTEGER;

dic.value.as\_int = 100;

dic.flag = NBIOT\_RESOURCE\_READABLE;

dic.write = NULL;

dic.execute = NULL;

/\* 创建OneNET接入设备实例 \*/

ret = nbiot\_device\_create( &dev, /\*本地udp绑定端口\*/ );

if ( NBIOT\_ERR\_OK != ret )

{

break;

}

/\* 连接OneNET服务 \*/

ret = nbiot\_device\_connect( dev,

“coap://183.230.40.40:5683”,

/\*保活时长\*/ );

if ( NBIOT\_ERR\_OK != ret )

{

nbiot\_device\_destroy( dev );

break;

}

/\* 配置设置 \*/

ret = nbiot\_device\_configure( dev,

/\*终端名称\*/,

/\*设备鉴权码\*/,

&(&dic),

1 );

if ( NBIOT\_ERR\_OK != ret )

{

nbiot\_device\_close( dev );

nbiot\_device\_destroy( dev );

break;

}

/\* 等待设备与OneNET服务连接就绪 \*/

i = 0;

do

{

/\* 数据驱动以及设备保活 \*/

ret = nbiot\_device\_step( dev, 1 );

if ( NBIOT\_ERR\_OK != ret )

{

break;

}

else

{

nbiot\_sleep( 10 );

}

} while ( !nbiot\_device\_ready() );

/\* 主动上报资源数据 \*/

if ( nbiot\_device\_ready() )

{

dic.value.as\_int = 123;

nbiot\_device\_notify( dev,

dic.objid,

dic.instid,

dic.resid );

/\* 确保上报资源数据成功 \*/

i = 100;

while ( i > 0 )

{

nbiot\_device\_step( dev, 1 );

nbiot\_sleep( 10 );

}

}

/\* 关闭连接，销毁设备实例 \*/

nbiot\_device\_close( dev );

nbiot\_device\_destroy( dev );

} while(0);

/\* 清理sdk运行环境 \*/

nbiot\_clear\_environment();

return 0;

}

**附录A CMake编译选项**

* HAVE\_DTLS - sdk支持dtls加密传输
* BIG\_ENDIAN – 内存模型为大字节序
* WITH\_LOGS – 输出日志
* BOOTSTRAP – 使用bootstrap(目前OneNET服务暂不提供bootstrap支持)

**附录B 编译时宏定义**

* NBIOT\_DEBUG – 输出sdk调试日志
* LWM2M\_WITH\_LOGS – 输出lwm2m相关的日志
* COAP\_WITH\_LOGS – 输出coap相关的日志
* NBIOT\_WIN – 目标系统为windows
* NBIOT\_POSIX – 目标系统支持posix
* BIG\_ENDIAN – 内存模型为大字节序
* LITTLE\_ENDIAN – 内存模型为小字节序
* HAVE\_DTLS – 带dtls加密传输
* DTLS\_WITH\_LOGS – 输出dtls相关的日志