应用笔记

AN_基于FreeRTOS+LwIP的SNMP实现方案

简介

SNMP：简单网络管理协议（Simple Network Management Protocol），是由互联网工程任务组定义的一套网络管理协议。SNMP可以使网络管理员通过一台工作站完成对计算机、路由器和其他网络设备的远程管理和监视。利用SNMP协议可以更好地管理和监控网络。管理工作站可以远程管理所有支持该协议的网络设备，如监视网络状态、修改网络设备配置、接收网络事件警告等。

SNMP作为TCP/IP协议的一部分，其消息是被封装到UDP协议中进行传输的。SNMP协议主要由两大部分组成：网络管理站（也叫管理进程）和被管的网络单元（也叫被管设备），被管设备种类众多，比如：路由器、终端服务器、打印机等。在本文档中，国民技术的N32G457QEL_EVB V1.1全功能开发板就充当被管理设备。被管设备端和管理相关的软件叫代理程序（SNMP agent）或者代理进程，它是运行于设备上的代码或程序。

管理进程和代理进程之间的通信可以有两种方式：一种是管理进程向代理进程发起的，询问其具体参数值（get操作）或者为其设置某个参数值（set操作）；另一种是代理进程主动向管理进程发起的，向其报告自身的某些重要事件的发生（trap操作）。

到目前为止，SNMP共有三个版本：v1，v2和v3。V1和v2有很多共同的特征，v3在前面版本的基础上增强了安全性方面的功能。其各自的消息格式、操作命令等将在后文中详细给出。

本文档主要针对国民技术MCU N32G45x系列产品在搭载FreeRTOS + LwIP的场景下的以太网实例：SNMP的应用。

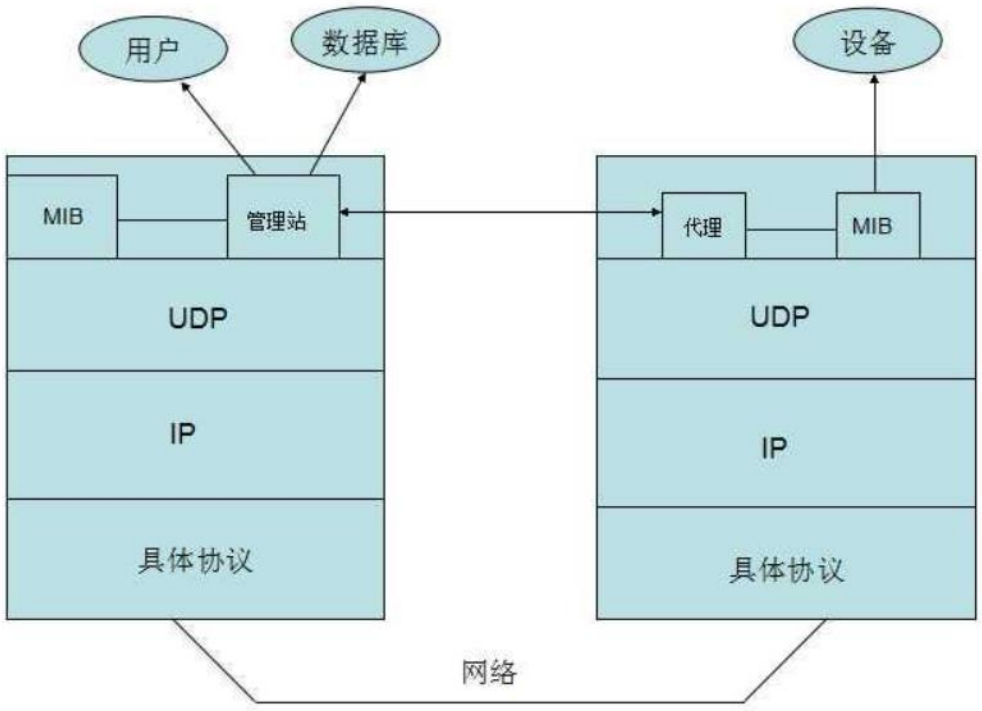
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

目录

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

1 SNMP基本概念

由于 SNMP 由管理站（NMS）和被管设备组成，在具体实现中，NMS 负责管理命令的发出、数据存储、数据分析，而运行于被管设备上的代理 agent 程序就负责与管理站的 SNMP 通信。管理站与代理 agent 端通过管理信息库（MIB）进行接口统一，MIB 定义了设备中的被管理对象。管理站和代理 agent 都实现了相应的 MIB 对象，以此来识别双方的数据，实现通信。管理站向代理 agent 请求 MIB 中定义的数据，代理 agent 识别后，将管理提供的状态或者参数转换成 MIB 定义的格式，返回给管理站，就完成了一次管理操作。

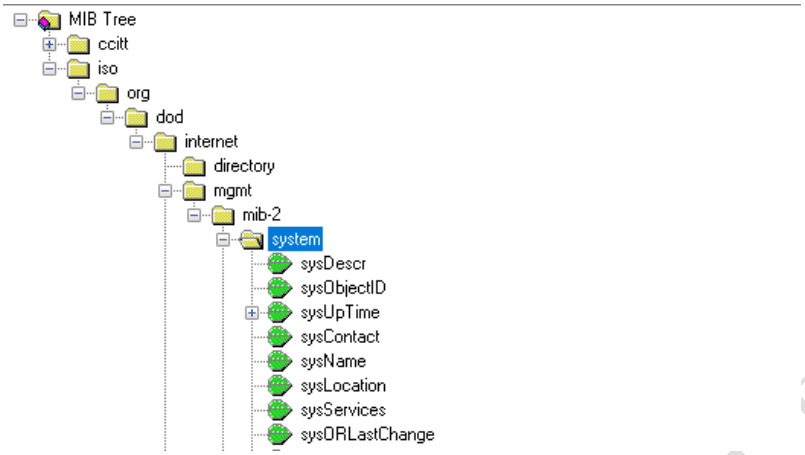


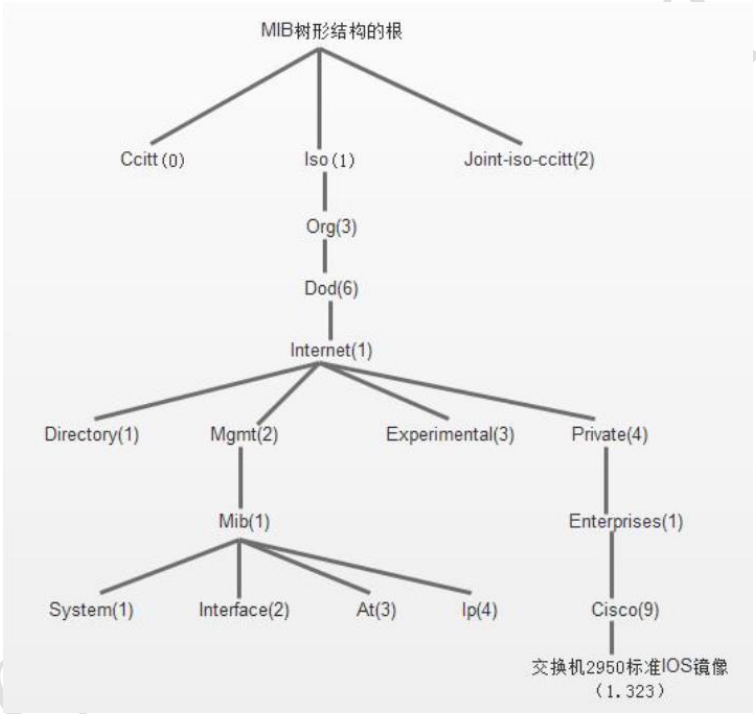
管理站与代理的通信

1.1 MIB

MIB 是（Management Information Base）管理信息库的缩写。它是 SNMP 管理与 agent 之间可以正常沟通的桥梁，是资源和对象标识符 oid（object identifier）之间唯一对应关系的数据库。任何一个被管理的资源都可以用 oid 来对应。所谓 oid，实质上是树形结构，每一个节点用一个数字对其进行标识，直至最终的资源，类似 PC 上的文件夹路径。例如下图中：Iso.Org.Dod.Internet.Mgmt.Mib.System 对应的 oid 为：.1.3.6.1.2.1.1。在实际开发中需要开发人员在 MIB 定义自己的 oid，为了便于测试，本文档中使用的是下图 System 节点下的叶子节点对应的 oid。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057





1.2 操作命令

Get-request 操作：SNMP 管理从 SNMP agent 获取一个或多各参数值；

Get-nest-request 操作：SNMP 管理从 SNMP agent 获取一个或多个参数的下一个参数值；

Get-bulk 操作：SNMP 管理从 SNMP agent 获取批量参数值，其中，non-repeaters 和 max-repetitions 两个字段必须设置，non-repeaters 是指前 n 个参数可以用 Get-nest-request 操作执行，max-repetitions 是指 n 之后的参数只能用 Get-nest-request 操作获取到最多 m 个参数值；

Set-request 操作：SNMP 管理设置 SNMP agent 的一个或者多个参数值；

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

Get-response 操作：SNMP agent 返回一个或多个参数值，是 SNMP agent 对 SNMP 管理的操作命令的响应；

Trap 操作：SNMP agent 主动发出的报文，目的是通知 SNMP 管理设备自身所发生的一些事件，执行该操作后，设备不会收到来自 SNMP 管理的回复报文；

Inform-request 操作：与 Tarp 操作类似，不同的是，SNMP agent 主动发出该报文后，如果连接畅通，将会收到来自 SNMP 管理的回复报文。

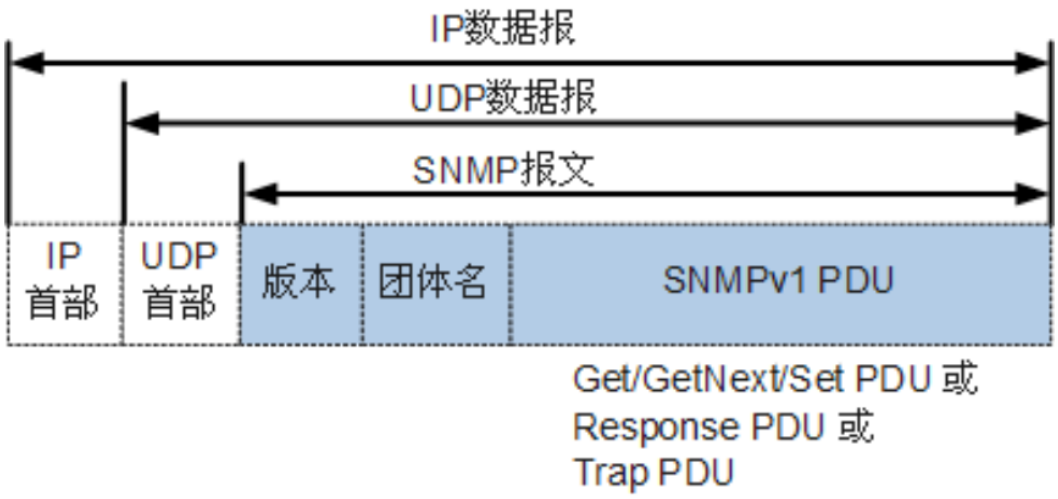
在 SNMP 三个版本中，v1 版本只包含 Get-request 操作、Get-nest-request 操作、Set-request 操作、Get-response 操作和 Trap 操作。V2 和 v3 版本都包含以上所有操作，并且 v3 在 v2 的基础上增加了安全性方面的功能。

1.3 报文格式

前面提到 SNMP 有 3 个不同版本的协议，接下来就看看它们各自的报文格式。

1.3.1SNMPv1报文格式

SNMPv1 报文主要由版本、团体名、SNMP PDU 三部分构成，如下图：



版本：表示 SNMP 的版本，版本字段的值是报文版本号减 1，如果是 SNMPv1 报文则对应字段值为 0。

团体名：用于在 agent 与 NMS 之间完成认证，字符串形式，常用的是 “public”和“private”。团体名包括“可读”和“可写”两种，执行 Get、Get-next 操作时，采用“可读团体名”进行认证；执行 Set 操作时，则采用“可写团体名”认证。

SNMPv1 PDU：包含 PDU 类型、请求标识符、变量绑定列表等信息。

其中，PDU 的格式如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

PDU Type	Request ID	Error Status	Error Index	Variable Bindings

PDU Type：协议数据单元的类型。支持 Get-request PDU、Get-next-request PDU、Set-request PDU、response PDU 或 Trap PDU 五种类型。

Request ID：请求标示字段，唯一的标示一个请求报文。

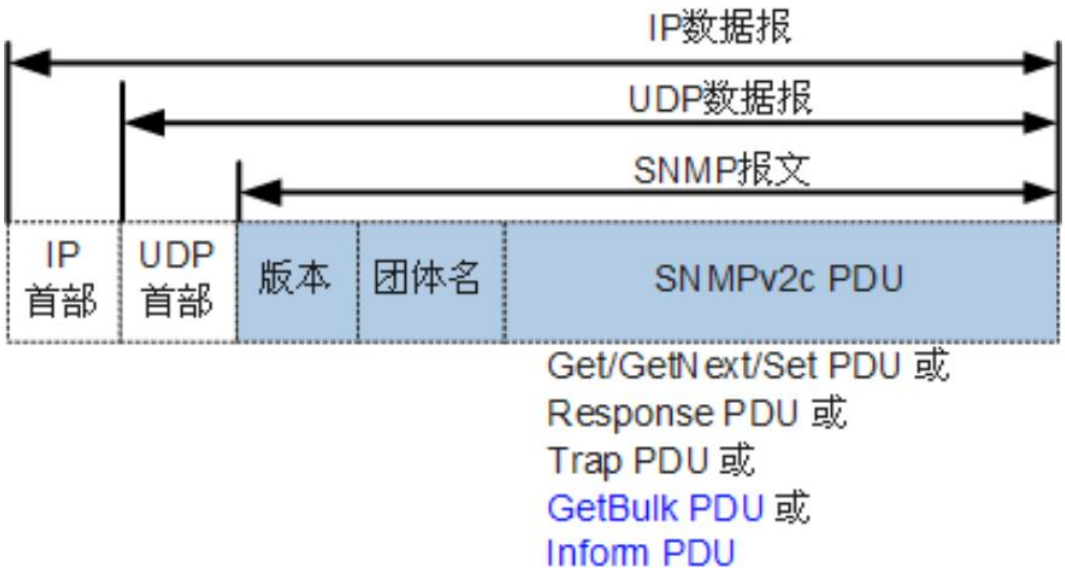
Error Status：错误状态标示字段。SNMPv1 中错误码包括：1、noSuchName：指定了一个代理不知道的对象；2、tooBig：代理不能一次把请求的结果放入到一个 PDU 中；3、badValue：进行 set 操作时候把变量修改为一个无效的值；4、genErr：除以上错误外的其他错误。

Error Index：错误索引字段。

Variable Bindings：变量绑定字段。

1.3.2 SNMPv2报文格式

与 SNMPv1 PDU 类似，SNMPv2c PDU 也包括 Get-request PDU、Get-next-request PDU、Set-request PDU、response PDU 或 Trap PDU 五种类型，并新增了 Get-bulk PDU 和 Inform PDU 两种类型，如下图：



版本：表示 SNMP 的版本，版本字段的值是报文版本号减 1，如果是 SNMPv2 报文则对应字段值为 1。

团体名：用于在 agent 与 NMS 之间完成认证，字符串形式，常用的是 “public”和“private”。团体名包括“可读”和“可写”两种，执行 Get、Get-next 操作时，采用“可读团体名”进行认证；执行 Set 操作时，则采用“可写团体名”认证。

SNMPv2c PDU：包含 PDU 类型、请求标识符、变量绑定列表等信息。

其中，PDU 格式如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

PDU Type	Request ID	Error Status	Error Index	Variable Bindings

PDU Type：协议数据单元的类型。支持 Get-request PDU、Get-next-request PDU、Set-request PDU、response PDU、Trap PDU、Get-bulk PDU 或 Inform PDU 七种类型。

Request ID：请求标示字段，唯一的标示一个请求报文。

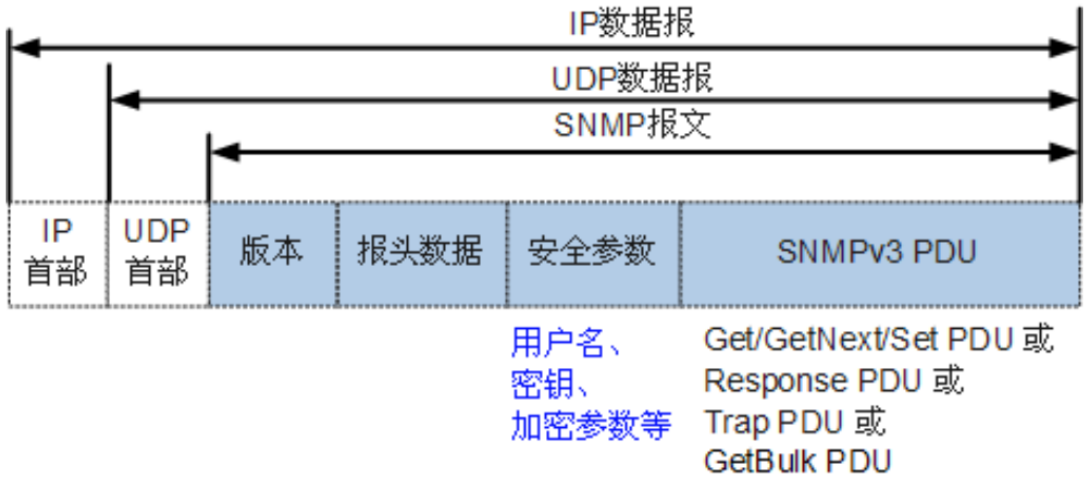
Error Status：错误状态标示字段。SNMPv2c 中错误码包括：1、wrongValue：进行 set 操作时候把变量修改为一个无效的值；2、wrongEncoding：进行编码字段的值，与其他的字段不一致；3、wrongType：进行 set 操作时候把变量修改为一个无效的类型；4、wrongLength：进行 set 操作时候把一个变量值设置成与它长度不一致的值；5、inconsistentValue：把一个变量设置为其他的情况下有效的值，当前情况下无效；6、noAccess：试图设置一个不可访问的值；7、notWritable：试图修改一个存在，但不能修改的值；8、noCreation：试图修改一个存在，但不能创建的值；9、inconsistentName：试图设置一个当前不存在且当前不能创建的变量；10、resourceUnavailable：设置过程中申请某些资源失败；11、commitFailed：set 操作失败；12、undoFailed：进行 set 操作失败，有些赋值无法回复；13、genErr：除以上错误外的其他错误。

Error Index：错误索引字段。

Variable Bindings：变量绑定字段。

1.3.3 SNMPv3报文格式

SNMPv3 报文结构从功能上来说，与 SNMPv1、SNMPv2c 的区别主要增加了报头数据和安全参数，如下图：



版本：表示 SNMP 的版本，该字段为 3.

报头数据：主要包含消息发送者所能支持的最大消息尺寸、消息是否进行加密/认证、采用的安全模式等描述内容。格式如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

Msg ID：可以使请求和应答相互关联，响应报文中的 Msg ID 和发送报文中的值相同。

Msg Max size：消息发送者支持的最大的消息尺寸。

Msg Sec Model：指明了发送方采用的安全模式。

Msg Flag：请求报文指定是否要求回应 report 消息，消息是否进行了加密和认证。

安全参数：包含用户名、密钥、加密参数等安全信息。格式如下：

Auth Engin ID：唯一的标识一个认证。

Auth Engin Boots：从配置认证引擎到现在，认证引擎重新启动的次数。

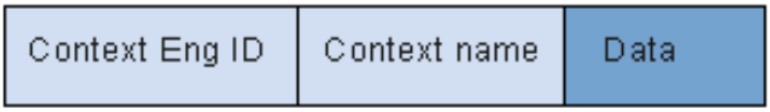
Auth Engin Time：从配置认证引擎到现在的时间。

User Name：用户名。

Auth Para：认证参数值。

Priv Para：加密后的参数值。

SNMPv3 PDU：包含 PDU 类型、请求标识符、变量绑定列表等信息，包含 Get-request PDU、Get-next-request PDU、Set-request PDU、response PDU、Trap PDU、Get-bulk PDU 或 Inform PDU 七种类型。格式如下：



Context Egine ID：SNMP 唯一标识符，和 PDU 类型一起决定应该发往那个应用程序。

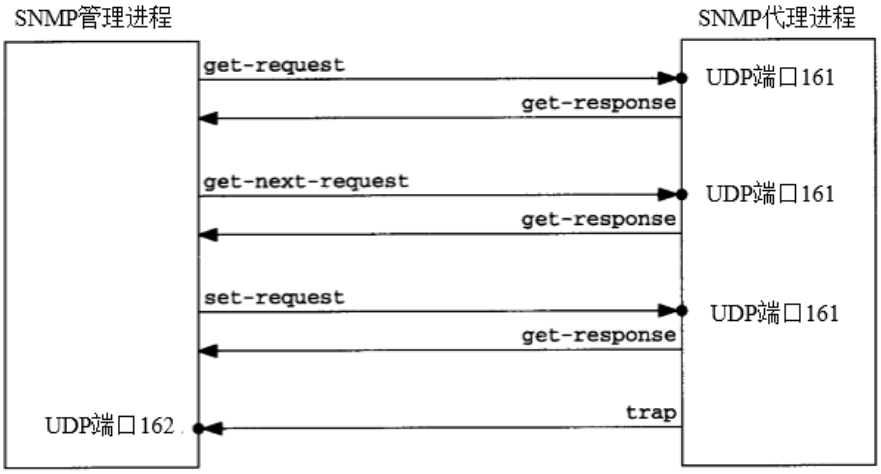
Context Name：指明上下文之间的关系，由应用程序决定。

Data：报文的数据内容。

1.4 SNMP查询上报

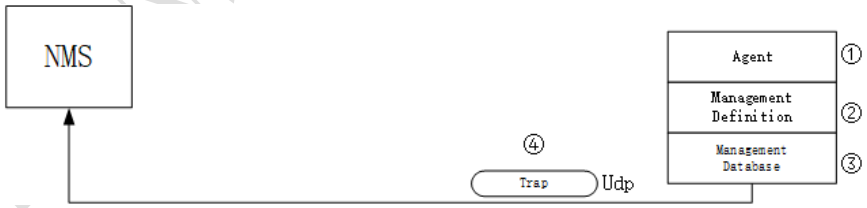
SNMP 管理与 SNMP agent 之间的查询操作采用的通信端口是 161，为了区别于查询操作，上报操作（trap）的通信端口用 162 实现：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

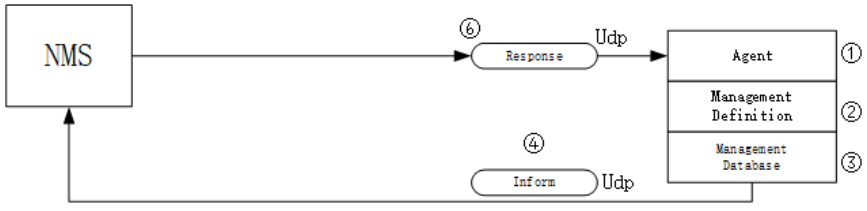


SNMP 管理的查询（get）和设置（set）流程，结合 SNMP 请求和响应，具体为：a. SNMP 管理通过资源在 MIB 上找到资源对应的 oid，组装成 UDP 报文发送给 SNMP agent。b. agent 接收并解析报文，根据报文中的 oid 去 MIB 中查找对应的资源，再根据报文中具体的指令去执行 get 或者 set 操作。c. agent get 到数据或者 set 参数成功则向 SNMP 管理作出响应，组装 UDP 响应报文返回给 SNMP 管理。如下图：

SNMP agent 上报（trap）流程，具体为：a. SNMP agent 检测到设备上有资源消息产生（设备自身发生了某些事件），根据资源到 MIB 中找到对应的 oid。b.封装成 UDP 消息发送给 SNMP 管理，注意此操作 SNMP 管理不会给 agent 发送返回数据包。如下图：



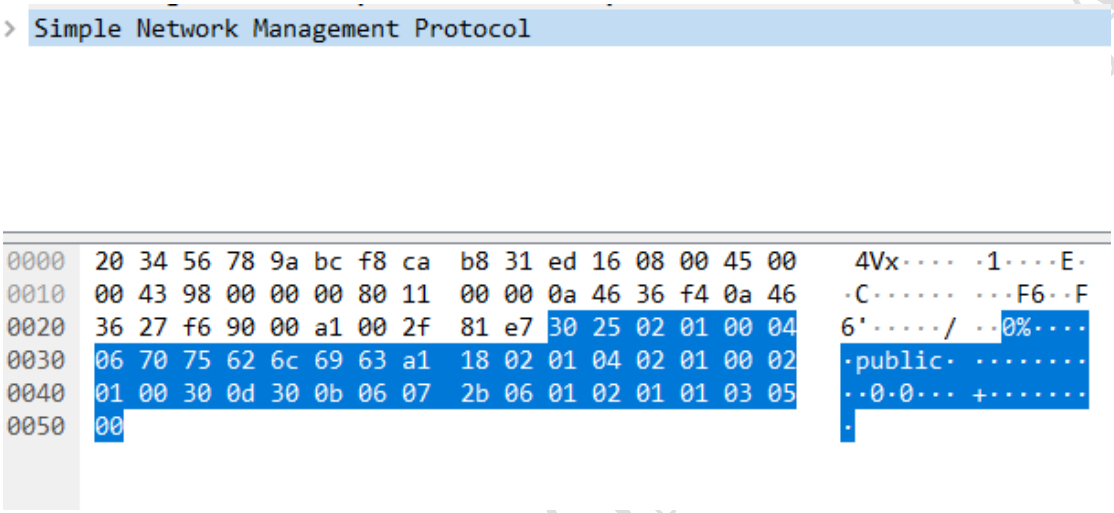
SNMP inform 流程，具体为: a. SNMP agent 检测到设备上有资源消息产生（设备自身发生了某些事件），根据资源到 MIB 中找到对应的 oid。b.封装成 UDP 消息发送给 SNMP 管理，管理接收到上报消息后查询 MIB 库，通过接收解析的 oid 找到对应的资源，最后发送响应消息给 SNMP agent 以确认收到了该消息。这也是与常规 trap 流程不同的地方，在上报消息方面比常规 trap 拥有显而易见的优势。如下图:



国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

1.5 ASN.1和BER

正式的 SNMP 规范中都采用了 ASN.1（Abstract Syntax Notation One）语法，并且报文中比特的编码采用 BER（Basic Encoding Rule）。ASN.1 语法是一种描述数据和数据特征的正式语言，为我们提供了一些基本的预定义数据类型，如：UNIVERSAL 2 表示整型，还有类似结构体类型的 SEQUENCE 等。而 BER 编码则是由三个部分组成：数据类型、数据长度和值。下面通过一段 Wireshark 抓取的 snmp 数据包来进一步了解 ASN.1 和 BER 编码的含义：



30：表示 SNMP 消息是 ASN.1 的 SEQUENCE 类型；

25：报文总长度；

02 01 00：版本号，BER 编码方式。02 表示 INTEGER 类型，01 表示数据长度为 1 个字节，00 表示值为 0，即版本号为 v1；

04 06 70 75 62 6C 69 63：团体名，BER 编码方式。04 表示 OCTETSTRING 类型，长度为 6 个字节，值对应的是“public”；

A1：表示 PDU 类型为 Get-next-request；

18：表示后面还有 0x18 个字节的数据，也是数据长度；

02 01 04：表示 request ID，BER 编码方式。值为 4；

02 01 00：表示 error statue，BER 编码方式。值为 0；

02 01 00：表示 error index，BER 编码方式。值为 0；

30 0D 30 0B：同样表示 ASN.1 的 SEQUENCE 类型，长度分别是 0x0D 和 0x0B；

06 07 2B 06 01 02 01 01 03：表示 object name，即 oid。BER 编码方式，06 表示 ASN.1 的 Object Identifier 类型，长度 7 个字节，值为：1.3.6.1.2.1.1.3；

05 00：表示 variable bindings 的值，05 表示 ASN.1 的 null 类型，长度为 0，所以后面没有其他值。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

以上就是一个 Get-next-request 请求的完整数据包组成内容，通对其分析即加深了对 ASN.1 和 BER 编码的认识也对我们自己自开发过程中编程解析数据包有帮助。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

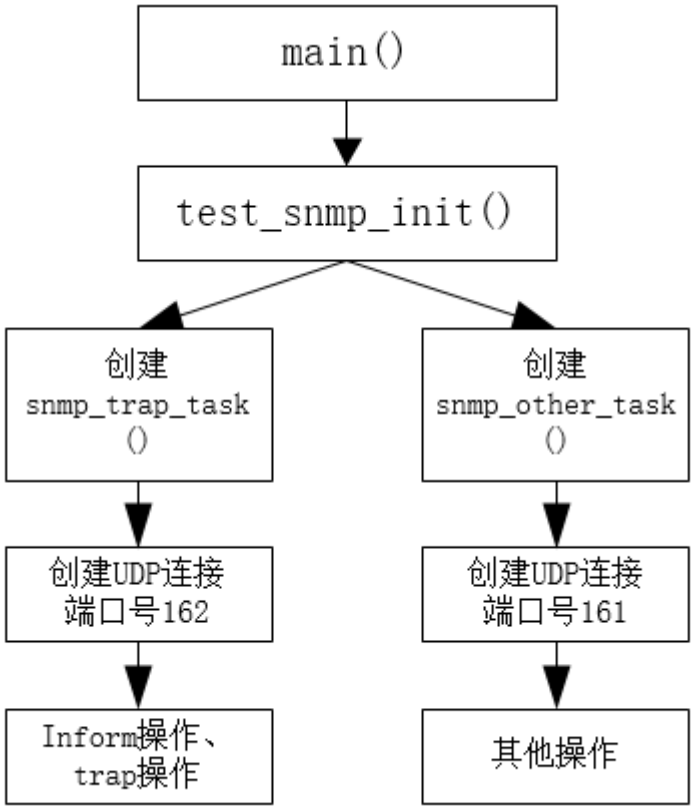
2 示例工程

硬件平台：PC（windows 系统）、N32G457QEL_EVB V1.1 全功能开发板

软件环境：keil5.30、MIB 浏览器（iReasoning MIB Browser 或 MG-SOFT MIB Browse）、FreeRTOS + LwIP

示例名称：test_snmp

示例工程中使用了操作系统和轻量型 IP 协议来实现 SNMP 协议的相关功能，主要的编程设计思路是依据 SNMP 协议的功能实现中用到两个 UDP 端口：161 和 162，故而创建两个线程，一个线程用于使用 161 端口通信的 Get-request 操作、Get-next-request 操作、Set-request 操作、Get-response 操作和 Get-bulk 操作操作，另一个线程用于使用 162 端口的 Trap 操作和 Inform 操作。如下图所示：



2.1 软件MIB库

在实现 SNMP 通信之前，需要在软件中添加自己的 MIB 库，为了便于测试，本次设计选取了官方 MIB 库中 System 节点下的叶子节点作为数据对象，软件中的 MIB 库定义如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
typedef struct{
    char oidLen;
    char oid[12];
    int tableFlag;
    char type;
    char dataLen;
    union{
        char octetstring[48];
        int intval;
    }data;
}MY_MIB;
```

其中定义了 oid 长度、oid、是否为表内对象标志位、参数值的类型、参数值的长度、整型或者字符串型参数。软件中使用的 MIB 初始化如下：

```
MY_MIB my_mib[] = {
    {7, {0x2b, 6, 1, 2, 1, 1, 1, 0},0,SNMP_DTYPE_OCTET_STRING,10,{"SNMP agent"}},
    {7, {0x2b, 6, 1, 2, 1, 1, 2, 0},0,SNMP_DTYPE_OBJ_ID,8,{0x2b,0x06,0x01,0x02,0x01,0x01,0x24,0x01}},
    {7, {0x2b, 6, 1, 2, 1, 1, 3, 0},0,SNMP_DTYPE_TIME_TICKS,1,0},
    {7, {0x2b, 6, 1, 2, 1, 1, 4, 0},0,SNMP_DTYPE_OCTET_STRING,19,{"www.nationstech.com"}},
    {7, {0x2b, 6, 1, 2, 1, 1, 5, 0},0,SNMP_DTYPE_OCTET_STRING,7,{"Nations"}},
    {7, {0x2b, 6, 1, 2, 1, 1, 6, 0},0,SNMP_DTYPE_OCTET_STRING,5,{"CHINA"}},
    {7, {0x2b, 6, 1, 2, 1, 1, 7, 0},0,SNMP_DTYPE_INTEGER,1,0},
    {7, {0x2b, 6, 1, 2, 1, 1, 8, 0},0,SNMP_DTYPE_TIME_TICKS,1,0},
    {7, {0x2b, 6, 1, 2, 1, 1, 9},0,SNMP_DTYPE_NULL_ITEM,0,0},
    {8, {0x2b, 6, 1, 2, 1, 1, 9, 1},0,SNMP_DTYPE_NULL_ITEM,0,0},
    {9, {0x2b, 6, 1, 2, 1, 1, 9, 1, 1, 0},1,SNMP_DTYPE_INTEGER,1,2},
    {9, {0x2b, 6, 1, 2, 1, 1, 9, 1, 2, 0},1,SNMP_DTYPE_OBJ_ID,8,{0x2b,0x06,0x01,0x02,0x01,0x01,0x24,0x01}},
    {9, {0x2b, 6, 1, 2, 1, 1, 9, 1, 3, 0},1,SNMP_DTYPE_OCTET_STRING,10,"sysORTable"},
    {9, {0x2b, 6, 1, 2, 1, 1, 9, 1, 4, 0},1,SNMP_DTYPE_TIME_TICKS,1,0}
};
```

2.2 解析tlv

所谓 tlv，就是指数据的类型、长度和具体的值，也就是 BER 编码格式的组成部分，由于 SNMP 协议数据包的这一编码特性，要解析整个数据包，只需依次解析出所有的 tlv，这一方法相比于按位解析有减少代码量等优势，以下代码定义了一个 tlv 结构体：

```
typedef struct{
    int tlvStartIndex;
    int tlvNextStartIndex;
    int tlvValueStartIndex;
    int dataType;
    int length;
    char value[];
}DATA_TLV;
```

其中定义了当前 tlv 开始下标、下一个 tlv 开始下标、当前 tlv 中的具体值的下标、当前 tlv 值的数据类型、当前 tlv 的值得长度和当前 tlv 的具体值。下面介绍解析 tlv 的函数实现，如下图：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
int snmp_parse_tlv(char *msg,int index,DATA_TLV *data_tlv)
{
    int pMsgStart;
    pMsgStart = *msg;
    data_tlv->tlvStartIndex = index;
    data_tlv->length = *(msg+1);//*(pMsgStart + 1);
    data_tlv->tlvValueStartIndex = index + 2;

    switch(pMsgStart)
    {
        case SNMP_DTYPE_SEQUENCE:
        case GET_REQUEST:
          if(pMsgStart == GET_REQUEST)
              data_tlv->dataType = GET_REQUEST;
        case GET_NEXT_REQUEST:
          if(pMsgStart == GET_NEXT_REQUEST)
              data_tlv->dataType = GET_NEXT_REQUEST;
        case SET_REQUEST:
          if(pMsgStart == SET_REQUEST)
              data_tlv->dataType = SET_REQUEST;
        case GET_BULK_RESPONSE:
          if(pMsgStart == GET_BULK_RESPONSE)
              data_tlv->dataType = GET_BULK_RESPONSE;
          data_tlv->tlvNextStartIndex = index + 2;
          data_tlv->length = 1;
          break;
        default:
          data_tlv->tlvNextStartIndex = index + data_tlv->length + 2;
          break;
    }
    snmp_msg_index = data_tlv->tlvNextStartIndex;

    return 1;
}
```

函数中有形参 msg、index 和 data_tlv，它们分别表示接收到的数据地址、当前数据的下标以及 DATA_TLV 类型的临时变量地址，并且传进来的 index 是一个全局变量，用于实时记录当前数据的下标。具体实现的功能是：记录当前 tlv 的开始下标、记录当前 tlv 中具体值的长度、判断并记录下一个 tlv 的开始下标以及当数据类型为 GET_REQUEST、GET_NEXT_REQUEST、SET_REQUEST、GET_BULK_RESPONSE 时分别记录当前的数据类型，这有助于在后面的解析应答中作出正确的响应操作。

2.3 解析snmpv1消息

解析 snmpv1 消息的函数是：int snmpv1_parse_message(char *pbuf)；形参 pbuf 是指发送缓存区的首地址，也就是在具体的实现过程中，在解析的同时也完成了响应数据包的打包，后面的操作也是这样，后文将不再赘述这一点。函数解析成功就返回数据包的长度，失败则返回-1。解析的过程主要是依据解析 tlv 的函数，由于篇幅原因，这里只粘贴出其中一处，如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
    /*Parse and response version number*/
    snmp_parse_tlv(&recv_buff[snmp_msg_index],snmp_msg_index,&data_tlv);
    if((recv_buff[data_tlv.tlvStartIndex] != SNMP_DTYPE_INTEGER) ||
          (recv_buff[data_tlv.tlvValueStartIndex] != SNMPV1))
    {
        return -1;
    }
    pbuf[send_data_index++] = SNMP_DTYPE_INTEGER;
    pbuf[send_data_index++] = data_tlv.length;
    pbuf[send_data_index++] = recv_buff[data_tlv.tlvValueStartIndex];
```

可以看出上面代码解析出 SNMP 版本号是否为 v1 版本，并且完成了响应报文中的版本号的打包。

前面已经定义了自己的 MIB 库，为了能够让响应报文和接收到的报文在内容上相匹配，本次设计中以两者均有的 oid 号为基准定义了函数：int find_my_mib_number(char *pData,int len)；目的是找出某个 oid 号在软件 MIB 库中对应的行号，形参 pData 是指当前接收值的地址，并且已经已知该值代表某个操作指令类型，len 是当前 oid 长度，函数成功就返回对应的行号，代码如下图：

```
int find_my_mib_number(char *pData,int len)
{
    int i,j,k = 0,numberMax = 0;
    numberMax = sizeof(my_mib) / sizeof(MY_MIB);

    for(i = 0;i < numberMax;i++,k = 0)
    {
        for(j = 0;j < len;j++)
        {
            if((*(pData+k)) != my_mib[i].oid[j])
                break;
            k++;
        }
        if(j == len)
            return i;
    }
    return -1;
}
```

成功获取了当前 oid 所在软件 MIB 中的行号后，当解析到指令类型为 GET_REQUEST 时，执行如下代码，把软件 MIB 中当前行 oid 对应的数据类型、长度、值打包到发送缓存区：

```
case GET_REQUEST:
    memcpy(&pbuf[send_data_index],&recv_buff[data_tlv.tlvStartIndex],(data_tlv.length+2));
    send_data_index += (data_tlv.length+2);

    /*Parse and response value*/
    pbuf[send_data_index++] = my_mib[number].type;
    pbuf[send_data_index++] = my_mib[number].dataLen;
    //snmp_parse_tlv(&recv_buff[snmp_msg_index],snmp_msg_index,&data_tlv);
    memcpy(&pbuf[send_data_index],&my_mib[number].data,my_mib[number].dataLen);
    send_data_index += my_mib[number].dataLen;
    break;
```

当解析到指令类型为 GET_NEXT_REQUEST 时，执行如下代码，把软件 MIB 中当前行 oid 的下一行 oid 对应的数据类型、长度、值打包到发送缓存区：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
case GET_NEXT_REQUEST:
    if(!my_mib[number].tableFlag)
    {
        if(number == ((sizeof(my_mib) / sizeof(MY_MIB)) - 1))
            number = 0;
        else
            number++;
    }

    pbuf[send_data_index++] = SNMP_DTYPE_OBJ_ID;
    pbuf[send_data_index++] = my_mib[number].oidLen + 1;
    memcpy(&pbuf[send_data_index],&my_mib[number].oid[0],my_mib[number].oidLen + 1);
    send_data_index += my_mib[number].oidLen + 1;

    /*Parse and response value*/
    pbuf[send_data_index++] = my_mib[number].type;
    pbuf[send_data_index++] = my_mib[number].dataLen;
    snmp_parse_tlv(&recv_buff[snmp_msg_index],snmp_msg_index,&data_tlv);
    memcpy(&pbuf[send_data_index],&my_mib[number].data,my_mib[number].dataLen);
    send_data_index += my_mib[number].dataLen;
    break;
```

当解析到指令类型为 SET_REQUEST 时，执行如下代码，把接收到的 oid 及其对应数据信息复制到发送缓存区，并且把该部分信息写回到软件 MIB 中当前行 oid 对应的数据类型、长度和值中，若再次发送 GET_REQUEST 指令操作，且 oid 为同一 oid 时，可以看到 my_mib 中已经成功设置了该值：

```
case SET_REQUEST:
    memcpy(&pbuf[send_data_index],&recv_buff[data_tlv.tlvStartIndex],(data_tlv.length+2));
    send_data_index += (data_tlv.length+2);

    /*Parse and response value*/
    snmp_parse_tlv(&recv_buff[snmp_msg_index],snmp_msg_index,&data_tlv);
    memcpy(&pbuf[send_data_index],&recv_buff[data_tlv.tlvStartIndex],(data_tlv.length+2));
    send_data_index += (data_tlv.length+2);

    memcpy(&my_mib[number].data,&recv_buff[data_tlv.tlvValueStartIndex],data_tlv.length);
    my_mib[number].type = recv_buff[data_tlv.tlvStartIndex];
    my_mib[number].dataLen = data_tlv.length;
    break;
```

2.4 解析snmpv2消息

解析 snmpv2 消息的函数是：int snmpv2_parse_message(char *pbuf)；形参 pbuf 是指发送缓存区的首地址，函数解析成功就返回数据包的长度，失败则返回-1。

本设计中解析 v2 消息与解析 v1 的步骤与内容大致相同，因为两者的数据包格式并无太大变化，由于篇幅原因，这里只粘贴出其中一处，其他可参考解析 snmpv1 消息的内容，代码如下：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
    /*Parse and response version number*/
    snmp_parse_tlv(&recv_buff[snmp_msg_index],snmp_msg_index,&data_tlv);
    if((recv_buff[data_tlv.tlvStartIndex] != SNMP_DTYPE_INTEGER) ||
          (recv_buff[data_tlv.tlvValueStartIndex] != SNMPV2))
    {
        return -1;
    }
    pbuf[send_data_index++] = SNMP_DTYPE_INTEGER;
    pbuf[send_data_index++] = data_tlv.length;
    pbuf[send_data_index++] = recv_buff[data_tlv.tlvValueStartIndex];
```

同样，可以看出上面代码解析出 SNMP 版本号是否为 v2 版本，并且完成了响应报文中的版本号的打包。

2.5 封装snmpv1 trap消息

前面已经介绍 v1 和 v2 报文消息的解析与封装响应包的过程，接下来介绍封装 v1 和 v2 封装 trap 消息，由于 trap 消息是 SNMP agent 主动向 SNMP 管理发出的报文消息，所以报文的每个比特数据都需要 agent 方自己填充，并且要按照相关的报文格式，否则管理方不能成功的解析出所发送的报文消息，封装 snmpv1 trap 消息的函数是：int snmpv1_trap_package(char *pbuf)；形参 pbuf 指发送缓存区地址，返回值是数据包总长度。下面贴出部分代码：

```
160     //snmp v1 trap package start
161     pbuf[send_data_index++] = SNMP_DTYPE_SEQUENCE;
162     pbuf[send_data_index] = 0xff;
163     data_len_index1 = send_data_index++;
164     /**version**/
165     pbuf[send_data_index++] = SNMP_DTYPE_INTEGER;
166     pbuf[send_data_index++] = 0x01;
167     pbuf[send_data_index++] = SNMPV1;
168     /**communtiy**/
169     pbuf[send_data_index++] = SNMP_DTYPE_OCTET_STRING;
170     pbuf[send_data_index++] = communtiy_len;
171     memcpy(&pbuf[send_data_index],&communtiy[0],communtiy_len);
172     send_data_index += communtiy_len;
173
174     /*trap*/
175     pbuf[send_data_index++] = V1_TRAP;
176     pbuf[send_data_index] = 0xff;
177     data_len_index2 = send_data_index++;
178     /**enterprise**/
179     pbuf[send_data_index++] = SNMP_DTYPE_OBJ_ID;
180     pbuf[send_data_index++] = enterprise_len;
181     memcpy(&pbuf[send_data_index],&enterprise[0],enterprise_len);
182     send_data_index += enterprise_len;
```

数据包的封装同样按照 ASN.1 和 BER 编码方式，可以看出：161-163 完成了 trap 包第一个 SEQUENCE 结构类型的封装，165-167 完成了版本号的封装，169-172 完成了团体名的封装，175-177 完成了消息类型为 v1 版本的 trap 消息的封装，179-182 完成了该设备的 oid 的封装，接下来还包含源地址，通用 trap 类型，企业私有类型等，在本设计中方法都类似，按照报文格式完成即可。还需要提到的一个技巧是：在封装一些 SEQUENCE 结构类型或者消息类型等无法实时记录其数据长度（tlv 结构中的 l）时，在本设计中是先将该比特位做标记，如上图 162 行和 176 行处，并赋初值 0xff，在封装数据包结束后按如下图方式，对其进行填充，

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

send_data_index 是指数据包的总长度，这一技巧在前面介绍的 v1、v2 的封装响应包和接下来的封装 v2 的 trap 包和 inform 包都有用到，后文不在赘述。

```
/*Fill the SEQUENCE type data length*/
pbuf[data_len_index1] = send_data_index-(data_len_index1+1);
pbuf[data_len_index2] = send_data_index-(data_len_index2+1);
pbuf[data_len_index3] = send_data_index-(data_len_index3+1);
pbuf[data_len_index4] = send_data_index-(data_len_index4+1);
```

2.6 封装snmpv2 trap消息、inform消息

snmpv2 的 trap 消息格式与 snmpv1 的要简洁，甚至与两者的 Get、Get-next、Set 操作的数据包格式类似，inform 消息的数据包格式与 trap 的同样也并无不同之处，只是注意的是 inform 数据包发出后，如果 SNMP 管理成功收到并解析后会返回一个响应包给 agent，而 trap 操作则没有响应包，而且 inform 消息不适用于 v1 版本，同时在封装的时候也要注意两者的消息类型的不同，本次设计中封装 snmpv2 trap 消息、inform 消息的函数如下图：

```
int snmpv2_inform_package(char *pbuf)
{
    return snmpv2_trap_or_inform_package(pbuf,INFORM_REQUEST);
}

int snmpv2_trap_package(char *pbuf)
{
    return snmpv2_trap_or_inform_package(pbuf,V2_TRAP);
}
```

其中，函数：int snmpv2_trap_or_inform_package(char *pbuf,int flag)；就实现了 trap 或者 inform 消息的封装，依据 flag 值得不同封装不同类型的消息，具体实现过程与封装 v1 的 trap 数据包类似，函数返回封装数据包的总长度。

至此，数据包的解析与封装就介绍完毕，下面介绍在各任务线程中的具体流程。

2.7 snmp_trap_task（）和snmp_other_task（）

由于 SNMP 协议是内嵌在 UDP 协议内进行传输的，故 snmp_trap_task（）和 snmp_other_task（）两个任务线程都建立了 UDP 连接，不同的是 snmp_trap_task（）用的端口号是 162，snmp_other_task（）用的端口号是 161，成功建立连接后，根据不同的版本，不同的操作类型完成解包或封装数据包的操作，最终通过 sendto 函数发送给 SNMP 管理端。下面分别贴出两者的部分主要内容：

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

```
#if INFORM
        //send INFORM message and recv the reply message
        while(do_inform_flag)
        {
#endif

        //Add code to package packets to be sent to the manager
#if DO_SNMPV1
        ret = snmpv1_trap_package(send_trap_buff);
        if(ret <= 0)
        {
#endif
#if DO_SNMPV2
        ret = snmpv2_trap_package(send_trap_buff);
        if(ret <= 0)
        {
#endif

        sendto(udp_socket, send_trap_buff, ret, 0,
            (struct sockaddr *)&remote_addr,sizeof(remote_addr));

          //Add code to parse SNMP request packets and package response packets
#if DO_SNMPV1
          ret = snmpv1_parse_message(send_buff);
          if(ret < 0)
          {
#endif
#if DO_SNMPV2
          ret = snmpv2_parse_message(send_buff);
          if(ret < 0)
          {
#endif
#if DO_SNMPV3

#endif
          sendto(udp_socket, send_buff, ret, 0,
              (struct sockaddr *)&remote_addr,sizeof(remote_addr));
          memset(send_buff,0,sizeof(send_buff));
```

工程示例部分介绍完毕。

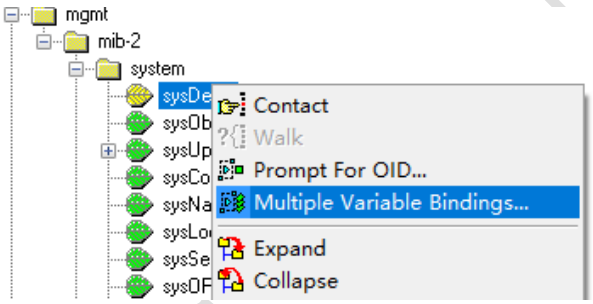
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

3 工程演示

测试 SNMP 功能需要用到官方 MIB 库，本次设计使用的是 MG-SOFT MIB Browser 专业版和 iReasoning MIB Browser MIB 浏览器。开发板作为 SNMP agent，IP 地址采用固定 IP 地址：10.70.54.39，也可以使用 DHCP 分配的动态 IP 地址。远端 PC 作为 SNMP 管理，IP 地址为：10.70.54.244，同时 PC 必须要开启相应的 SNMP 服务。为了方便查看测试和两者通信的效果，使用 Wireshark 网络抓包工具以及串口调试助手。

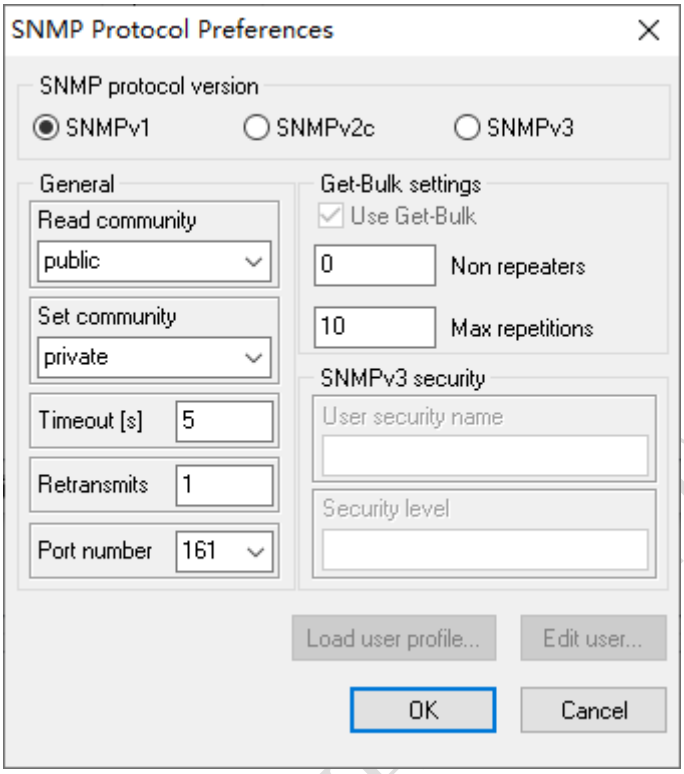
3.1 DO_SNMPV1

在 test_snmp.h 文件中设置宏 DO_SNMPV1 为 1、宏 DO_SNMPV2 为 0，即可使用 v1 版本进行测试（相反则为 v2 版，后文不再赘述），然后编译、下载程序到开发板，按复位按钮运行。提示：由于 PHY 初始化需要时间，可等待片刻，待串口调试助手成功打印“LWIP Init Success！”后进行后续操作。即，打开 MIB 浏览器在前面提到的 MIB 树的 system 子节点的叶子节点上任选一个叶子节点后，右键选择 Multiple Variable Bindings…，如下图：

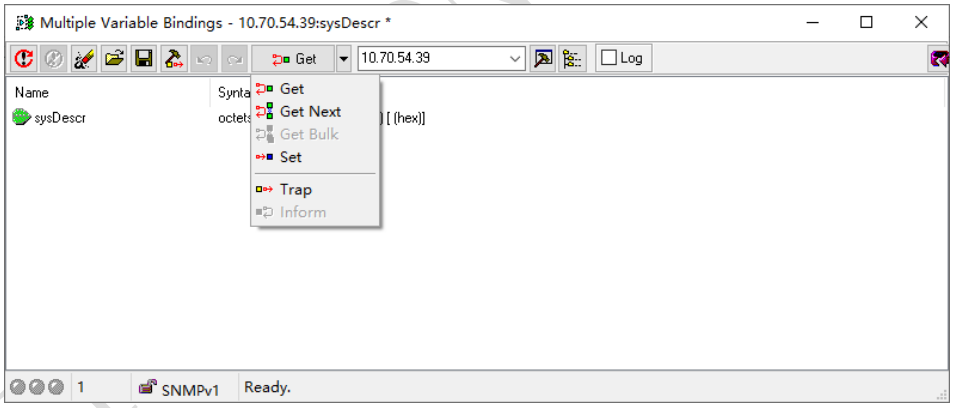


在弹出来的页面中，点击  图标，即可在新弹出的页面中选择 SNMP 版本，以及设置某些必要的参数，如下图：

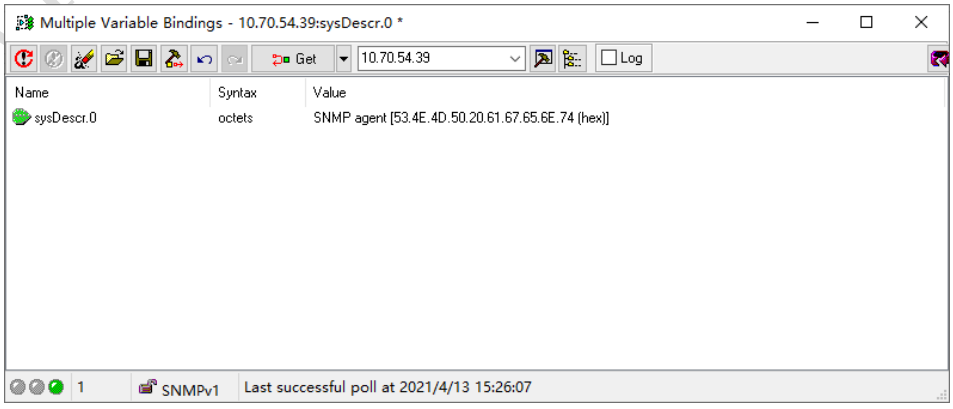
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057



版本选择 v1，其他参数可采用默认设置，点击 OK 即可。再在下图的界面中选择不同的操作指令：

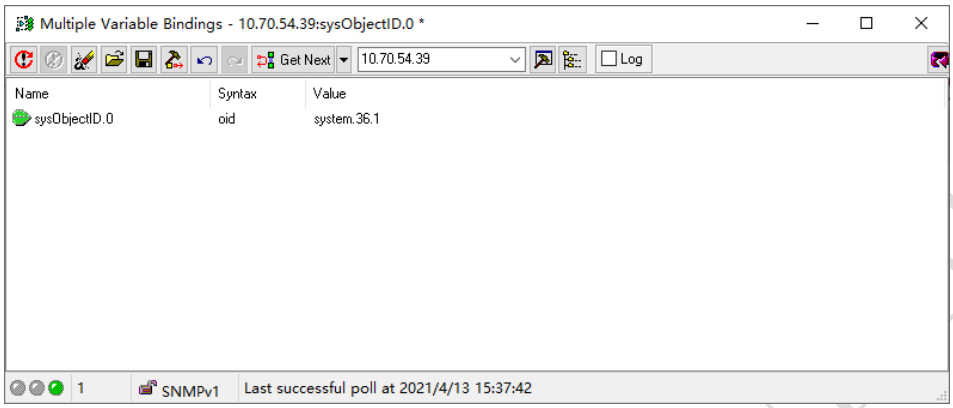


选择 Get 后点击该按钮，此时可以看到，已经获取到了当前 oid 对应的参数值，如下图所示：

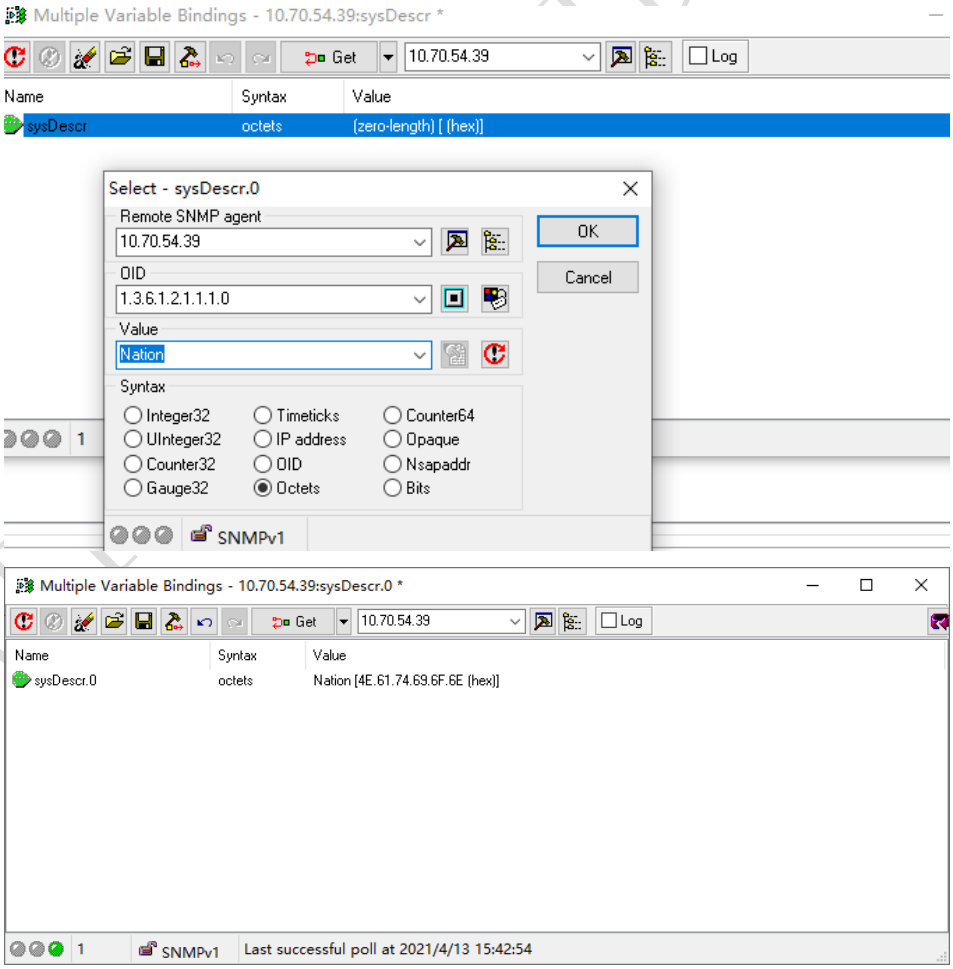


国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

选择 Get Next 后点击该按钮，此时可以看到，已经获取到了当前 oid 的下一个 oid 对应的参数值，如下图所示：



双击 sysDescr，并在如下的界面填写好 value 和选择 octets 类型后，选择 Set 操作并按该按钮，即可完成对该参数值的设置操作，此时并不能准确、直观的看出 Set 操作是否成功，我们可以按前面的操作步骤，重新进入 Multiple Variable Bindings…选项，对 sysDescr 对象重新进行 Get 操作，此时就可看到后图所示的内容：



国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

可以看出，此次获取的参数值与事先 Set 的参数值一样都是：Nation，并且类型为：octets。到目前为止，就已经实现了 v1 的 Get、Get-next、Set 操作了，再来看看 Wireshark 抓包工具对应的抓包结果：

```
458 2487.699751   10.70.54.244   10.70.54.39    SNMP   82 get-request 1.3.6.1.2.1.1.1.0
459 2487.703901   10.70.54.39    10.70.54.244   SNMP   92 get-response 1.3.6.1.2.1.1.1.0
563 3182.533231   10.70.54.244   10.70.54.39    SNMP   82 get-next-request 1.3.6.1.2.1.1.1.0
566 3182.539524   10.70.54.39    10.70.54.244   SNMP   90 get-response 1.3.6.1.2.1.1.2.0
583 3446.141575   10.70.54.244   10.70.54.39    SNMP   83 set-request 1.3.6.1.2.1.1.1.0
584 3446.145730   10.70.54.39    10.70.54.244   SNMP   83 get-response 1.3.6.1.2.1.1.1.0
590 3472.647519   10.70.54.244   10.70.54.39    SNMP   89 set-request 1.3.6.1.2.1.1.1.0
591 3472.651620   10.70.54.39    10.70.54.244   SNMP   89 get-response 1.3.6.1.2.1.1.1.0
601 3494.097218   10.70.54.244   10.70.54.39    SNMP   82 get-request 1.3.6.1.2.1.1.1.0
602 3494.101349   10.70.54.39    10.70.54.244   SNMP   88 get-response 1.3.6.1.2.1.1.1.0
```

```
▾ get-response
    request-id: 11
    error-status: noError (0)
    error-index: 0
  ▾ variable-bindings: 1 item
    ▾ 1.3.6.1.2.1.1.1.0: 4e6174696f6e
        Object Name: 1.3.6.1.2.1.1.1.0 (iso.3.6.1.2.1.1.1.0)
      › Value (OctetString): 4e6174696f6e
```

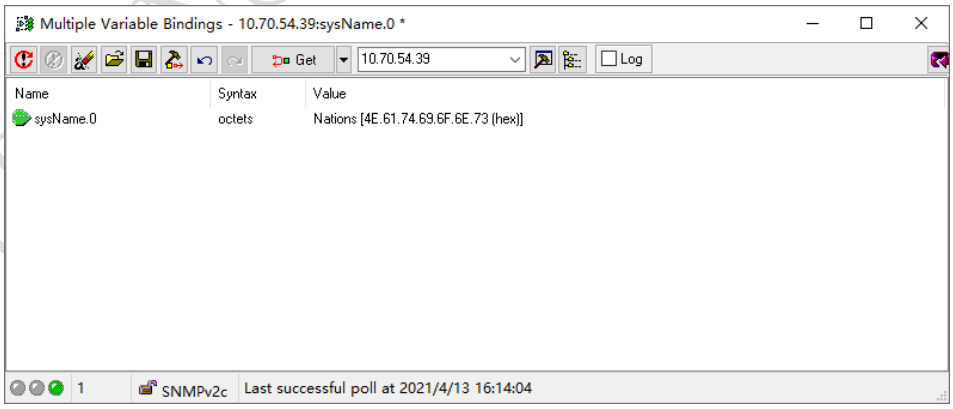
```
0000  f8 ca b8 31 ed 16 20 34  56 78 9a bc 08 00 45 00   ···1·· 4 Vx····E·
0010  00 4a 00 04 00 00 80 11  b8 f8 0a 46 36 27 0a 46   ·J····· ···F6'·F
0020  36 f4 00 a1 e0 a3 00 36  31 00 30 2c 02 01 00 04   6·····6 1·0,····
0030  06 70 75 62 6c 69 63 a2  1f 02 01 0b 02 01 00 02   ·public· ········
0040  01 00 30 14 30 12 06 08  2b 06 01 02 01 01 01 00   ··0·0··· +·······
0050  04 06 4e 61 74 69 6f 6e                           ··Nation
```

可以看出，每一次请求操作都有一次响应，并且对应的参数值均与软件 MIB 库里的值相匹配。

3.2 DO_SNMPV2

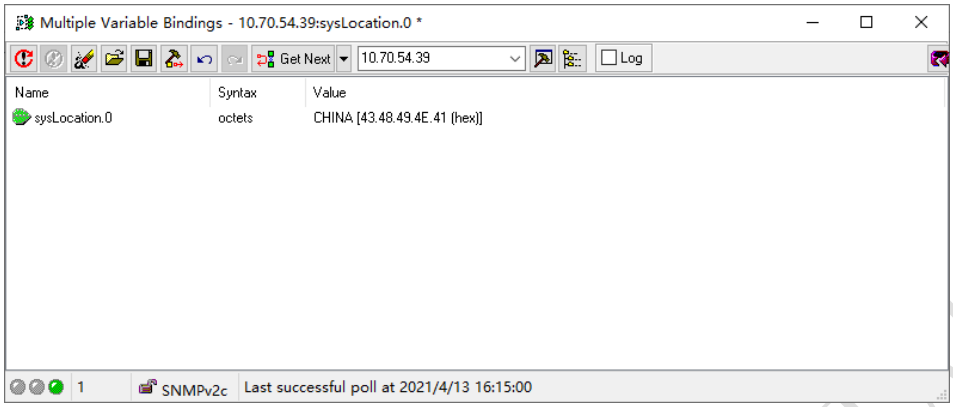
测试 v2 版本的操作与一样，只需要注意前面介绍的选择版本号的界面的操作，一定要正确选择版本号，其他参数都可以保持默认配置，下面就直接看各种操作指令的结果。

Get 操作：

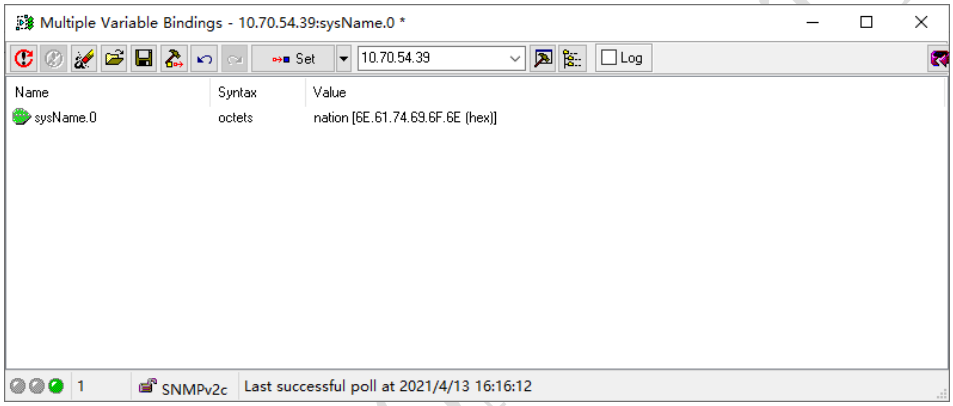


Get-next 操作：

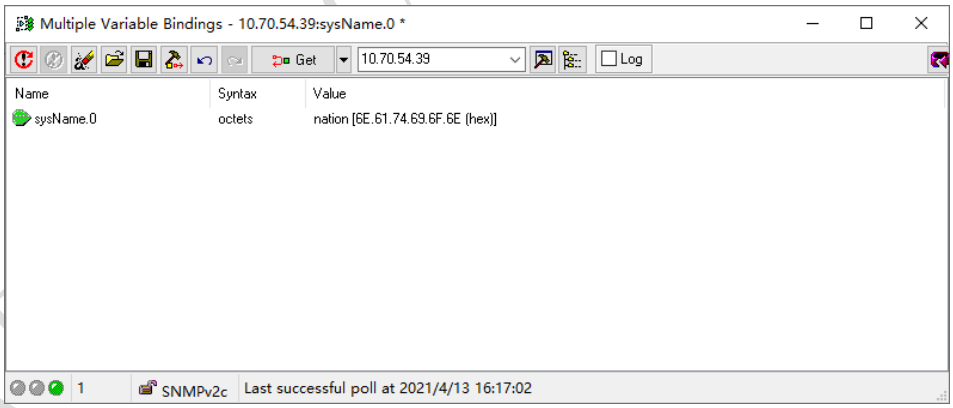
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057



Set 操作：

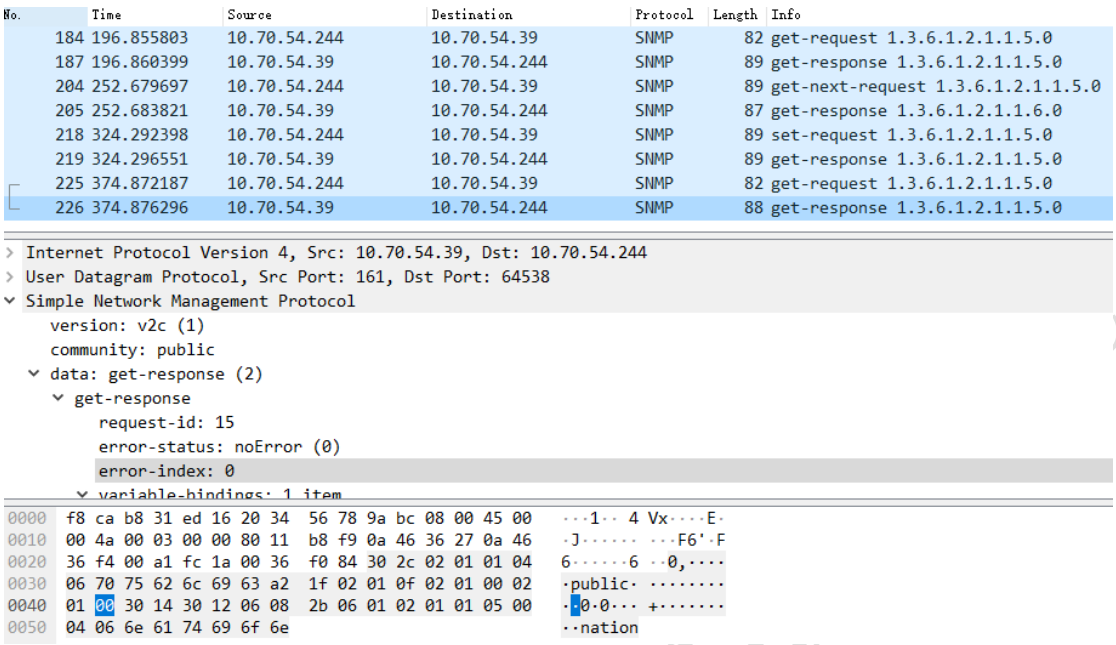


Set 完再 Get：



Wireshark 抓包结果：

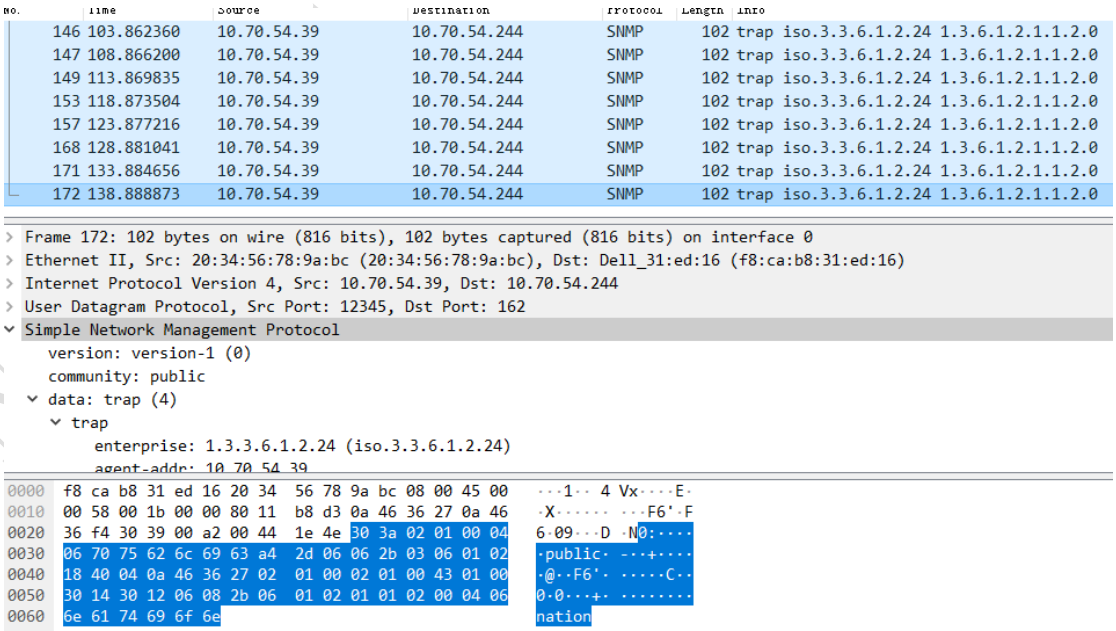
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057



从各种操作的结果反馈上可以看出各参数值与软件 MIB 库里的值相匹配，Wireshark 抓包依然呈现出一次请求对应着一次响应，这就是 SNMPv2 的各种操作功能。

3.3 SNMPv1 trap

trap 操作是 SNMP agent 主动发出的数据包，本设计中为了方便看到结果，设置为每 5s 发出一个 trap 包，Wireshark 抓取数据如下图：



3.4 SNMPv2 trap和inform

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

对于 v2 的 trap，同样定义为 5s 发送一个数据包，Wireshark 抓取数据如下图：

```
   260 30.805215    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   280 35.808920    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   282 40.812608    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   283 45.816310    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   286 50.820102    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   288 55.823754    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   289 60.827509    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
   293 65.831214    10.70.54.39     10.70.54.244    SNMP    88 snmpV2-trap 1.3.6.1.2.1.1.2.0
```

```
> Frame 293: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface 0
> Ethernet II, Src: 20:34:56:78:9a:bc (20:34:56:78:9a:bc), Dst: Dell_31:ed:16 (f8:ca:b8:31:ed:16)
> Internet Protocol Version 4, Src: 10.70.54.39, Dst: 10.70.54.244
> User Datagram Protocol, Src Port: 12345, Dst Port: 162
v Simple Network Management Protocol
    version: v2c (1)
    community: public
  v data: snmpV2-trap (7)
    v snmpV2-trap
        request-id: 10
        error-status: noError (0)
```

```
0000  f8 ca b8 31 ed 16 20 34  56 78 9a bc 08 00 45 00   ···1·· 4 Vx····E·
0010  00 4a 00 0a 00 00 80 11  b8 f2 0a 46 36 27 0a 46   ·J····· ···F6'·F
0020  36 f4 30 39 00 a2 00 36  bf 65 30 2c 02 01 01 04   6·09···6 ·e0,····
0030  06 70 75 62 6c 69 63 a7  1f 02 01 0a 02 01 00 02   ·public· ········
0040  01 00 30 14 30 12 06 08  2b 06 01 02 01 01 02 00   ··0·0··· +·******
0050  04 06 6e 61 74 69 6f 6e                           ··nation
```

Wireshark 抓取 inform 发送数据包如下图：

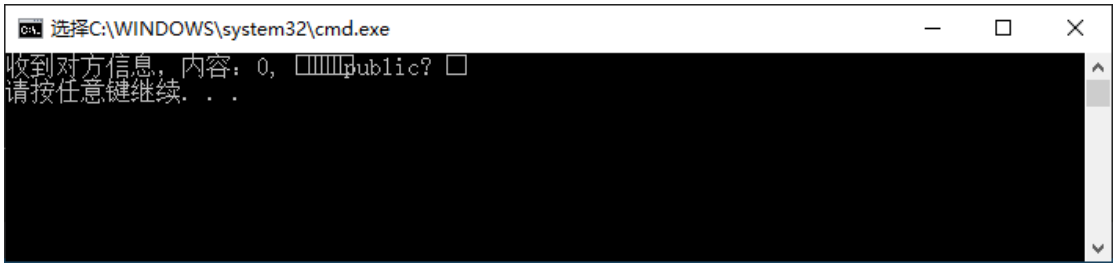
```
   414 120.322971   10.70.54.39     10.70.54.244    SNMP    88 informRequest 1.3.6.1.2.1.1.2.0
```

```
> Frame 414: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface 0
> Ethernet II, Src: 20:34:56:78:9a:bc (20:34:56:78:9a:bc), Dst: Dell_31:ed:16 (f8:ca:b8:31:ed:16)
> Internet Protocol Version 4, Src: 10.70.54.39, Dst: 10.70.54.244
> User Datagram Protocol, Src Port: 12345, Dst Port: 162
v Simple Network Management Protocol
    version: v2c (1)
    community: public
  v data: informRequest (6)
    v informRequest
        request-id: 0
        error-status: noError (0)
        error-index: 0
      v variable-bindings: 1 item
        v 1.3.6.1.2.1.1.2.0: 6e6174696f6e
            Object Name: 1.3.6.1.2.1.1.2.0 (iso.3.6.1.2.1.1.2.0)
            Value (OctetString): 6e6174696f6e
```

```
0000  f8 ca b8 31 ed 16 20 34  56 78 9a bc 08 00 45 00   ···1·· 4 Vx····E·
0010  00 4a 00 00 00 00 80 11  b8 fc 0a 46 36 27 0a 46   ·J····· ···F6'·F
0020  36 f4 30 39 00 a2 00 36  bf 70 30 2c 02 01 01 04   6·09···6 ·p0,····
0030  06 70 75 62 6c 69 63 a6  1f 02 01 00 02 01 00 02   ·public· ········
0040  01 00 30 14 30 12 06 08  2b 06 01 02 01 01 02 00   ··0·0··· +·******
0050  04 06 6e 61 74 69 6f 6e                           ··nation
```

到目前为止，有一点需要我们注意，不管是 v1 还是 v2 的 trap 操作和 v2 的 inform 操作，都用的是 162 端口，但是我们的 PC 端开启了相应的 SNMP 服务，所以在使用的过程中会出现 162 号端口被占用的情况，主要体现在 inform 操作时程序会一直停在接收响应包的函数处，在测试的时候关闭相关服务即可正常完成 trap 或者 inform 操作。在本设计的测试过程中，修改了 162 号端口号，并在 Visual 开发工具上创建了一个简单的 UDP 服务端，用于接收 inform 数据包，成功后返回一个响应包，开发板成功收到响应则进行后面的 trap 操作，PC 端的输出结果如下图：

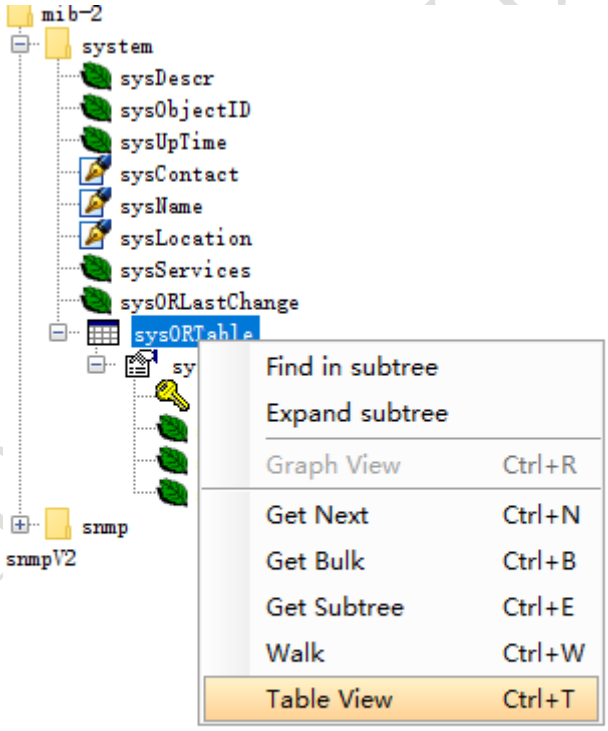
国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057



由于 PC 端的简易程序没有正规解析收到的数据包，但是从打印的内容中的部分字符不难看出数据确实来自我们的 SNMP agent 开发板。

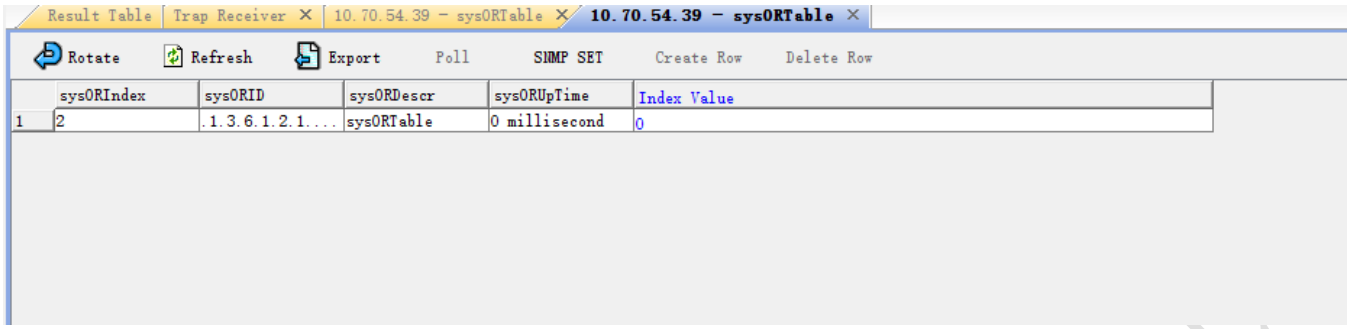
3.5 TableView功能

本设计中实现了 v1 和 v2 的 TableView 功能，任然利用 system 节点下的 sysORTable 表节点作简单功能测试，具体操作如下（说明：本功能用 “iReasoning MIB Browser”MIB 浏览器测试的，当然前面介绍的也可以用该 MIB 浏览器完成相关功能）：

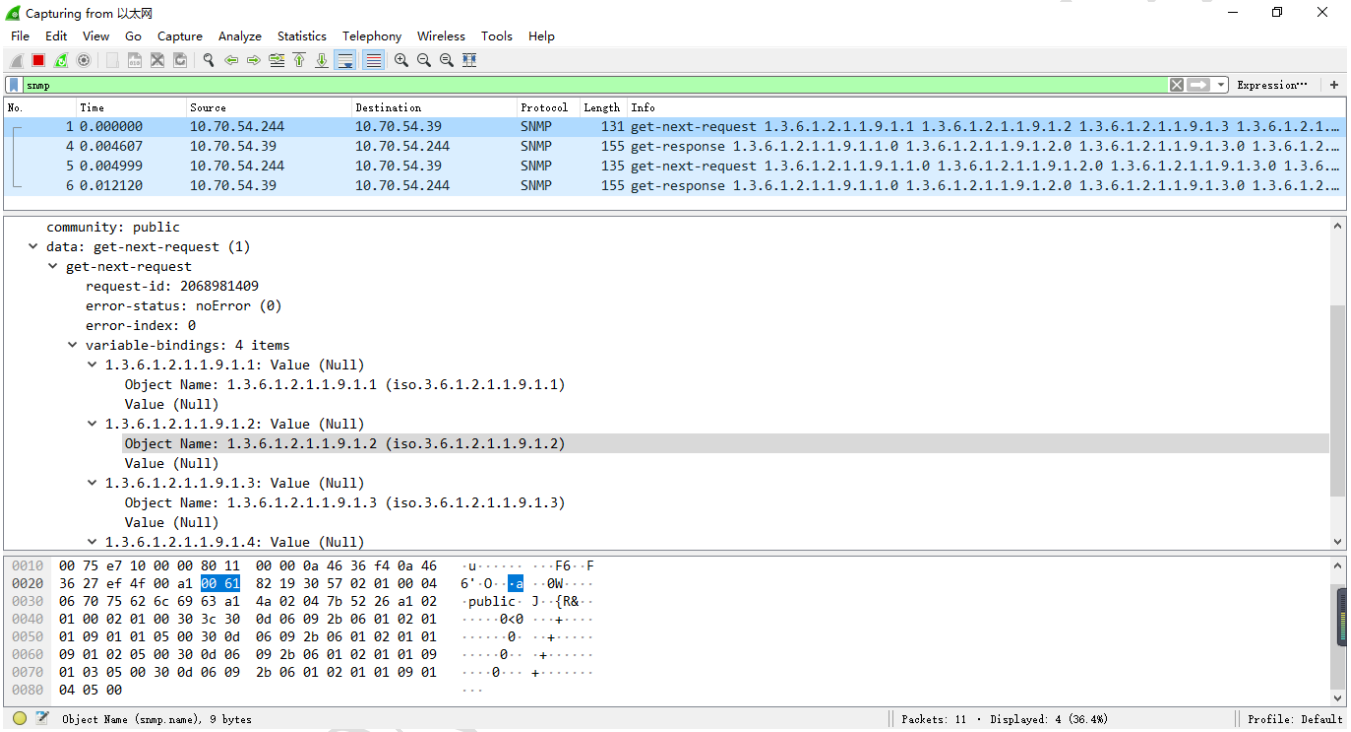


右键 sysORTable，选择 Table View 即可在界面右侧收到如下图所示的表格，并且已经成功返回了表格内各参数对象和其实际值，该值与软件 MIB 中相匹配。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057



通过抓包工具可看到如下图所示的数据包：



国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

4 总结

本示例实现了SNMPv1和SNMPv2两个版本的相关通信功能，其中两者均包含：Get-request 、Get-next-request 、Set-request 、Get-response 和Trap操作，除此之外，SNMPv2还包含Inform操作和两个版本均实现了Table View功能，用户可以根据需要灵活切换。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

5 历史版本

版本	日期	备注
V0.1	2021.04.13	新建文档

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057

6 声明

国民技术股份有限公司（以下简称国民技术）保有在不事先通知而修改这份文档的权利。国民技术认为提供的信息是准确可信的。尽管这样，国民技术对文档中可能出现的错误不承担任何责任。在购买前请联系国民技术获取该器件说明的最新版本。对于使用该器件引起的专利纠纷及第三方侵权国民技术不承担任何责任。另外，国民技术的产品不建议应用于生命相关的设备和系统，在使用该器件中因为设备或系统运转失灵而导致的损失国民技术不承担任何责任。国民技术对本文档拥有版权等知识产权，受法律保护。未经国民技术许可，任何单位及个人不得以任何方式或理由对本文档进行使用、复制、修改、抄录、传播等。

国民技术股份有限公司 Nations Technologies Inc.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
邮箱：info@nationz.com.cn 邮编：518057