RNDIS 共享 win10 网络连接

1. 配置 linux 内核，添加对 rndis 的支持，具体配置如下：

   Device Drivers ---->

   [*] USB support ---->

   <*> USB Gadget Support ---->

```
        USB Peripheral Controller  --->
<M>     USB Gadget Drivers
< >       USB functions configurable through configfs
< >       Gadget Zero (DEVELOPMENT)
< >       Audio Gadget
<M>       Ethernet Gadget (with CDC Ethernet support)
[*]         RNDIS support
```

   编译 os，在内核编译目录中能找到下列驱动：

   g_ether.ko usb_f_ecm_subset.ko libcomposite.ko u_ether.ko

   usb_f_rndis.ko usb_f_ecm.ko

   将这些驱动拷贝到设备中。

2. 在设备中加载驱动，如下图示：

```
/data/usb_modules/rndis # insmod libcomposite.ko
[  102.748578] [pax_verify_memory_region] verification skipped (level 2)
/data/usb_modules/rndis # insmod u_ether.ko
[  107.759313] [pax_verify_memory_region] verification skipped (level 2)
/data/usb_modules/rndis # insmod usb_f_ecm.ko
[  115.362108] [pax_verify_memory_region] verification skipped (level 2)
/data/usb_modules/rndis # insmod usb_f_ecm_subset.ko
[  117.679072] [pax_verify_memory_region] verification skipped (level 2)
/data/usb_modules/rndis # insmod usb_f_rndis.ko
[  127.445622] [pax_verify_memory_region] verification skipped (level 2)
/data/usb_modules/rndis # insmod g_ether.ko
[  131.503773] [pax_verify_memory_region] verification skipped (level 2)
[  131.511830] using random self ethernet address
[  131.516304] using random host ethernet address
[  131.522581] usb0: HOST MAC ae:d1:22:85:1c:84
[  131.526946] usb0: MAC 96:af:10:c4:e6:88
[  131.530910] using random self ethernet address
[  131.535399] using random host ethernet address
[  131.539978] g_ether gadget: Ethernet Gadget, version: Memorial Day 2008
[  131.546606] g_ether gadget: g_ether ready
/data/usb_modules/rndis # [  131.819927] g_ether gadget: high-speed config #2: RNDIS
```

   加载顺序如下：

   （1） insmod libcomposite.ko

   （2） insmod u_ether.ko

   （3） insmod usb_f_ecm.ko

   （4） insmod usb_f_ecm_subset.ko

   （5） insmod usb_f_rndis.ko

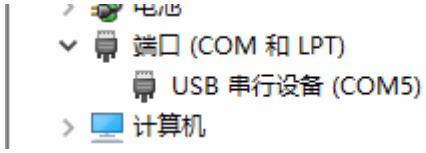
   （6） insmod g_ether.ko

   驱动加载完成后，设备中就会多出一个 usb0 的虚拟网卡，如下图示：

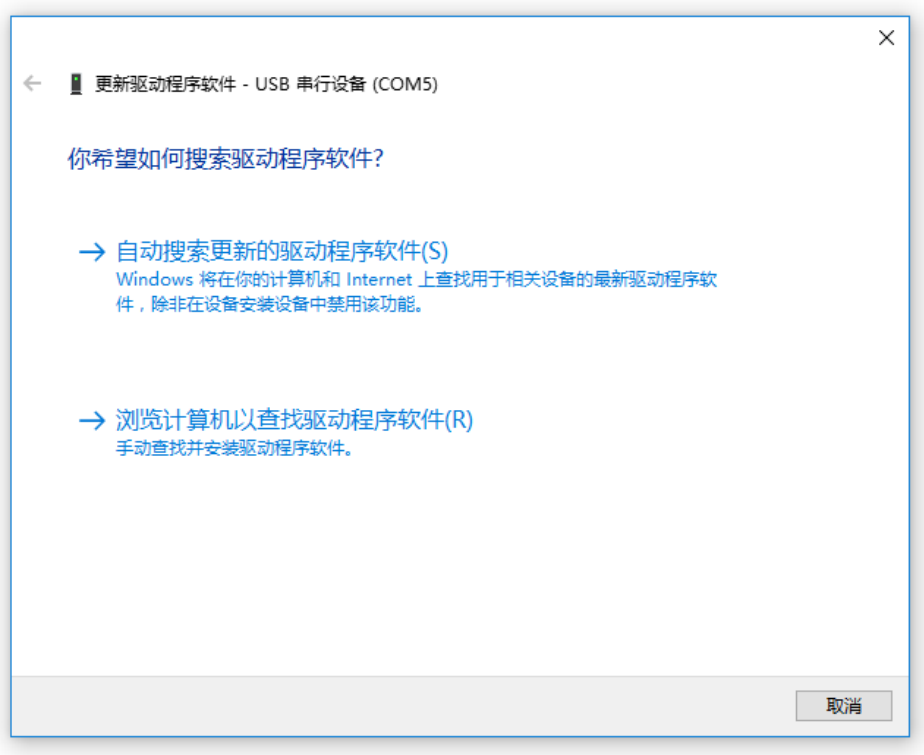
   图中的 usb0 即是 usb 在设备端虚拟出来的网卡。

```
/data/usb_modules/rndis # ifconfig -a
eth0      Link encap:Ethernet  HWaddr [redacted]
          inet addr:192.168.1.42  Bcast:192.168.1.255  Mask:255.255.255.0
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22160 (21.6 KiB)  TX bytes:22160 (21.6 KiB)

sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0      Link encap:Ethernet  HWaddr [redacted]
          inet addr:172.16.144.5  Bcast:172.16.147.255  Mask:255.255.252.0
          inet6 addr: fe80::94af:10ff:fec4:e688/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4578 errors:0 dropped:38 overruns:0 frame:0
          TX packets:279 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:447433 (436.9 KiB)  TX bytes:26678 (26.0 KiB)
```

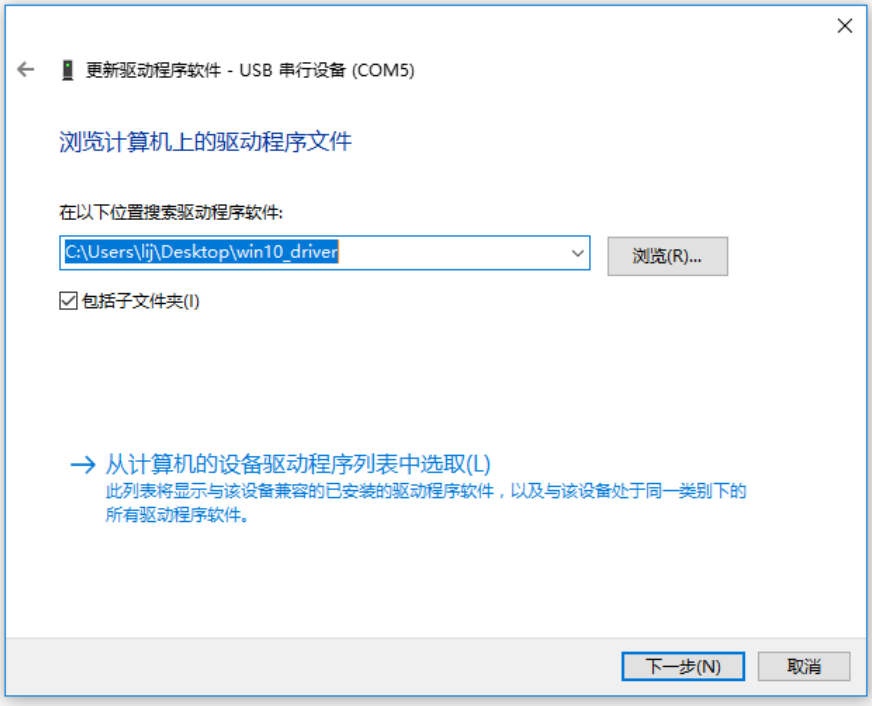
3. 在设备端启动网卡，
   （1） ifconfig –a 查看虚拟网卡 usb0
   （2） ifconfig usb0 up 启动虚拟网卡
4. 在 windows 中，设置外网网卡 A 的网络参数，确保 PC 能通过网卡 A 连接网络。
5. 将设备接入 PC，在设备管理器中找到新介入的设备，更新设备驱动程序，安装 windows 的 rndis 驱动。完成后，设备将被识别为虚拟网卡 B。
   由于 windows 系统自身的问题，在插入设备后，windows 可能自动安装错误的驱动程序，导致在设备管理器中，被识别成非虚拟网卡的设备。如在我的 win10 系统中，当设备插入时，被识别成了通用串行设备。如下图：



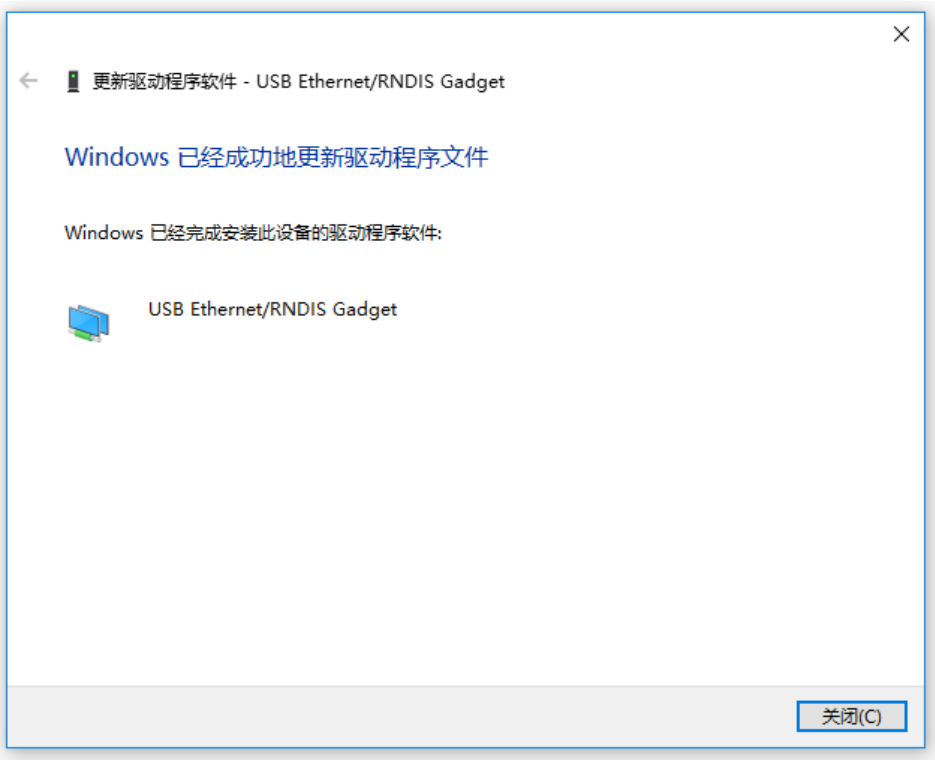
当出现这样的情况时，需要在设备管理其中仔细查找到新插入的设备究竟被识别成了什么，然后卸载掉该设备的驱动程序，自己手动安装正确的驱动程序。



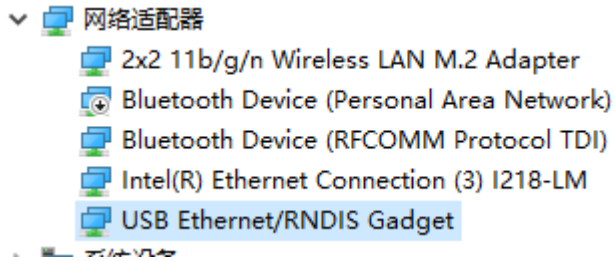
选择”浏览计算机以查找驱动程序软件”,选择存放正确驱动的路径,



点击下一步，安装驱动。安装完成后，弹出如下界面:



此时，在设备管理器中，就能找到如下的设备信息显示：



上图中的”USB Etnernet/RNDIS Gadget”即为设备在 windows 中虚拟出来的网络适配器。

此时，进入网络设置界面，就能看到虚拟出来的网卡，如下图：

上图中的”以太网 2”就是刚才设备虚拟出来的网卡。

6. 在 windows 中点选网卡 A 和网卡 B，建立网桥。等待网络桥接完成。

如图，我这里需要桥接无线网卡和设备的虚拟网卡，如下图示：

点击桥接后，windows 会建立网桥，实现两个网卡的桥接。如下图:

桥接完成后，网桥和虚拟网卡的状态都会发生改变，如下图示：

此时，虚拟网卡已经经由网桥和无线网卡桥接在了一起。

7. 在设备端，ifconfig usb0 up，启动设备中的虚拟网卡。设置网卡 usb0 的参数。
如下图示：

```
/data/usb_modules/rndis # udhcpc -i usb0
udhcpc (v1.22.1) started
Setting IP address 0.0.0.0 on usb0
Sending discover...
Sending discover...
Sending discover...
Sending select for 172.16.144.5...
Lease of 172.16.144.5 obtained, lease time 259200
Setting IP address 172.16.144.5 on usb0
Deleting routers
route: SIOCDELRT: No such process
Adding router 172.16.144.1
Recreating /etc/resolv.conf
 Adding DNS server 192.168.0.111
 Adding DNS server 192.168.0.126
/data/usb_modules/rndis #
```

完成后，在端就能连接网络了，结果如下：

```
/data/usb_modules/rndis # ping -I usb0 www.baidu.com
PING www.baidu.com (119.75.218.70): 56 data bytes
64 bytes from 119.75.218.70: seq=0 ttl=52 time=43.055 ms
64 bytes from 119.75.218.70: seq=1 ttl=52 time=87.769 ms
64 bytes from 119.75.218.70: seq=2 ttl=52 time=53.794 ms
64 bytes from 119.75.218.70: seq=3 ttl=52 time=45.096 ms
64 bytes from 119.75.218.70: seq=4 ttl=52 time=51.447 ms
64 bytes from 119.75.218.70: seq=5 ttl=52 time=48.630 ms
64 bytes from 119.75.218.70: seq=6 ttl=52 time=57.938 ms
64 bytes from 119.75.218.70: seq=7 ttl=52 time=42.089 ms
64 bytes from 119.75.218.70: seq=8 ttl=52 time=44.390 ms
64 bytes from 119.75.218.70: seq=9 ttl=52 time=41.867 ms
64 bytes from 119.75.218.70: seq=10 ttl=52 time=53.434 ms
```

注意：

1. 上述方法不能共享 win7 的网络连接，因为 windows 自带的 rndis 驱动存在问题，当在 win7 中尝试安装该驱动时，win7 会弹出警告界面，告知用户由于设备兼容性问题，可能导致系统异常。实际测试中，当在 win7 64 位系统中安装该驱动，并插入设备进行测试，win7 有很高的概率出现蓝屏，系统崩溃。
2. 在 win10 设备中插入设备时，win10 可能安装错误的驱动程序，造成不能实现网络共享。具体解决办法请参考 https://forum.moddevices.com/t/rndis-driver-for-windows-10/299，所需要安装的驱动见 mod-duo-rndis。

稳定性：建立 TCP 长连接，做小数据通信，持续 6 个多小时，网络通信正常，系统未出现异常。