

Overview

The Virtual NIC project is a simple demonstration program based on the KSDK. It is enumerated as a network adapter. Users can access the network by properly configuring this network adapter. The purpose of this demo is to show how to build a device of USB CDC class to implement RNDIS protocol and to provide a simple project for further development.

System Requirement

Hardware requirements

- Mini/micro USB cable
- USB A to micro AB cable
- Hardware (Tower module/base board, and so on) for a specific device
- Personal Computer

Software requirements

- The project files are in:
<SDK_Install>/boards/<board>/usb_examples/usb_device_cdc_vnic/<rtos>/<toolchain>.
For lite version, the project files are in:
<SDK_Install>/boards/<board>/usb_examples/usb_device_cdc_vnic_lite/<rtos>/<toolchain>.

Note

The <rtos> is Bare Metal or FreeRTOS OS.

Getting Started

Hardware Settings

Note

Set the hardware jumpers (Tower system/base module) to default settings.

Prepare the example

1. Download the program to the target board.
2. Either press the reset button on your board or launch the debugger in your IDE to begin running the demo.
3. Plug in the network cable before running this example. (Or you may have to disable and then enable the RNDIS network adapter after you plug in the network cable.)
4. Connect a USB cable between the PC host and the USB device port on the board.

Note

For detailed instructions, see the appropriate board User's Guide.

Run the example in Windows[®]

1. A network adapter is enumerated in Device Manager.

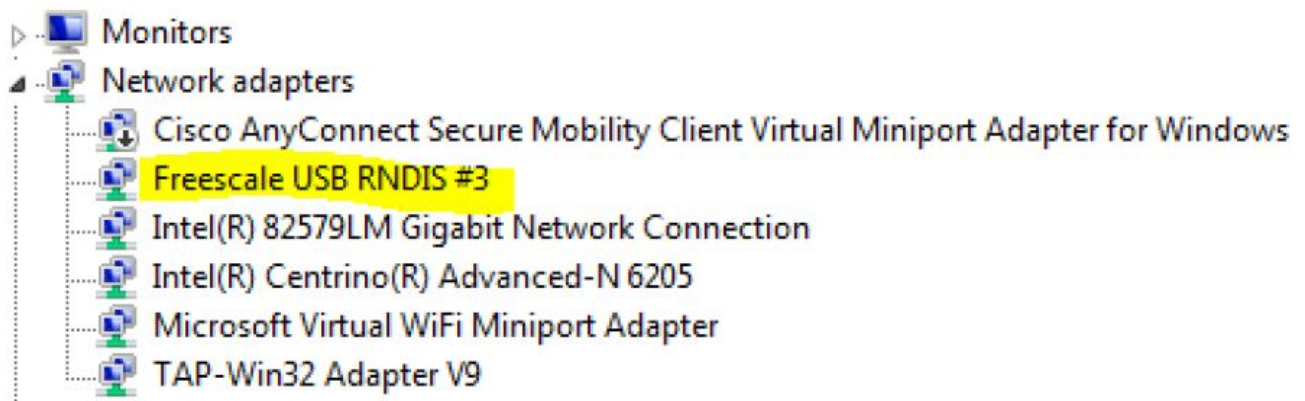


Figure 1: Virtual NIC in device manager

2. In *Control Panel\Network and Internet\Network Connections*, the RNDIS network adapter is listed as below.

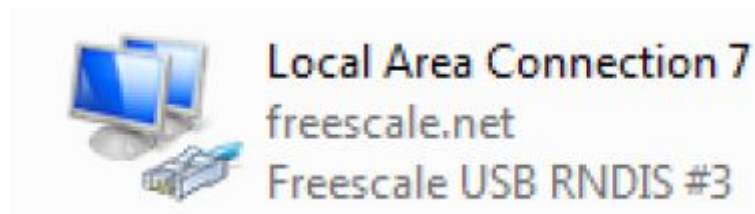


Figure 2: Virtual NIC in Network Connections

3. Configure the RNDIS adapter in Windows. E.g

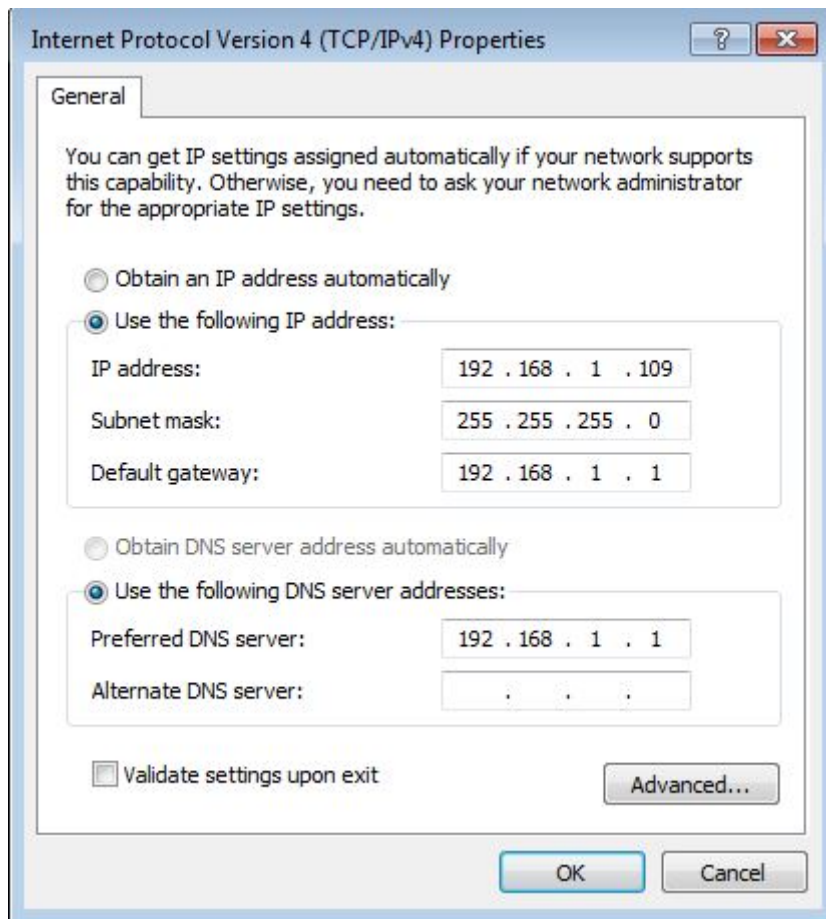


Figure 3: Virtual NIC configuration

Then you are supposed to access the network through this USB RNDIS network adapter.

```
C:\Users\>ping -t -S 192.168.1.109 192.168.1.102

Pinging 192.168.1.102 from 192.168.1.109 with 32 bytes of data:
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
Request timed out.
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
Reply from 192.168.1.102: bytes=32 time=1ms TTL=64
```

Figure 4: Ping to other PC

Note

1. The VNIC CIC CLASS code can be changed to wireless controller class code (0xE0), like:

```
#define USB_CDC_VNIC_CIC_CLASS (0xE0)
```

```
#define USB_CDC_VNIC_CIC_SUBCLASS (0x01)
```

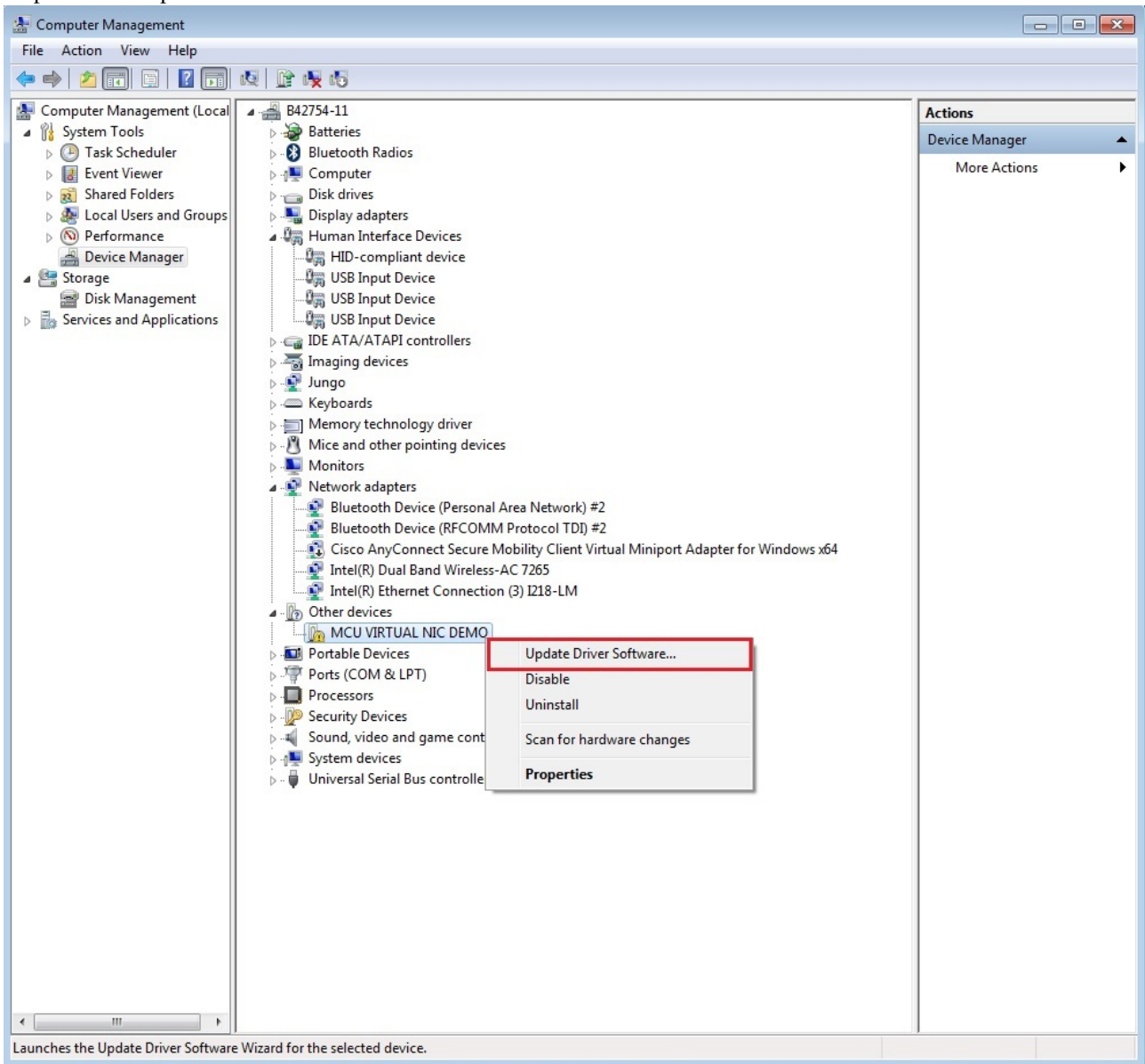
```
#define USB_CDC_VNIC_CIC_PROTOCOL (0x03)
```

Then there is no need to install inf file and the USB protocol suite can recognize the RNDIS and OID messages directly, but this is not verified under all conditions.

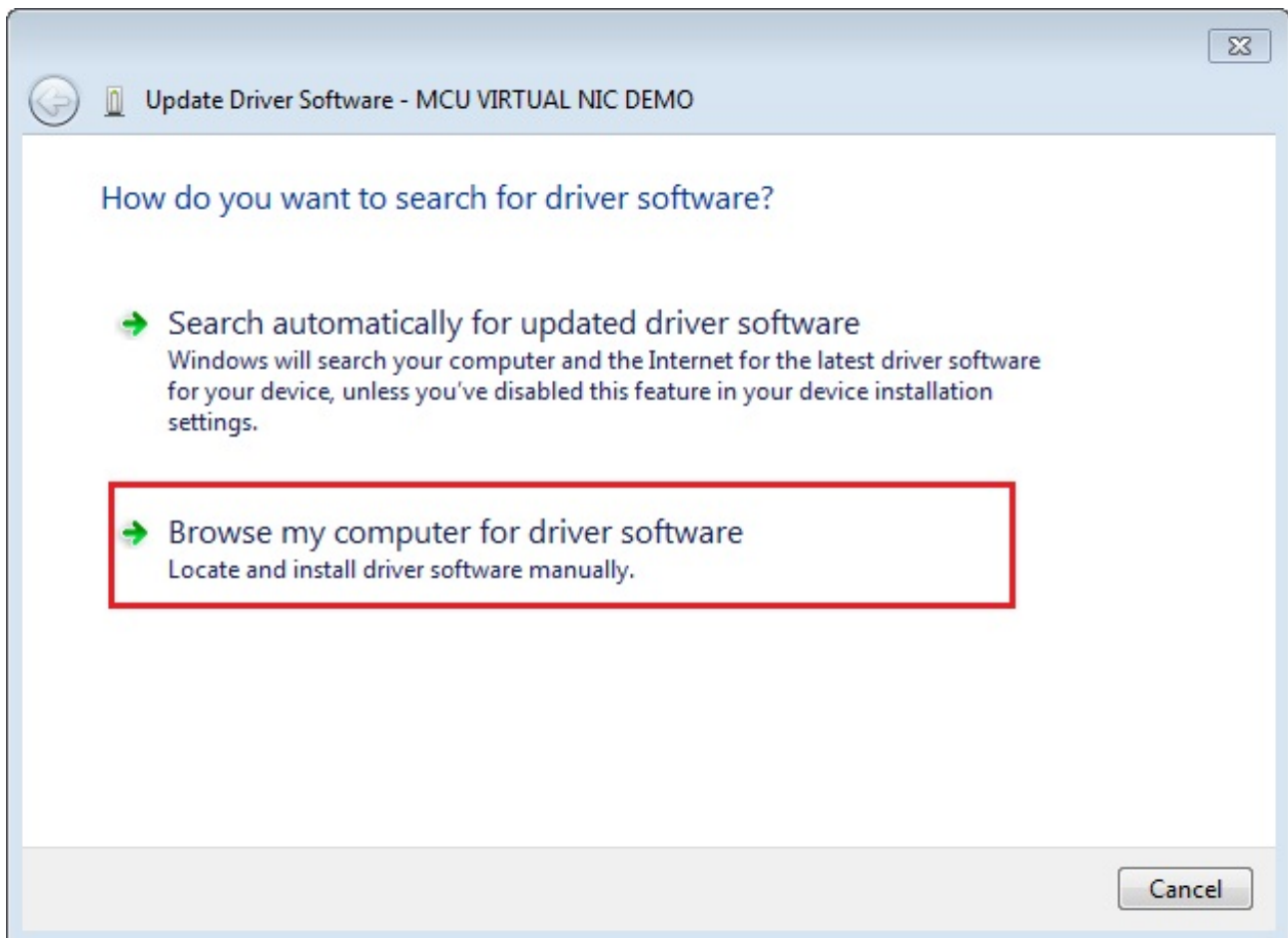
Installing the RNDIS driver for virtual_nic example

Below are the steps to install the RNDIS driver on Windows 7. Similar steps apply for Windows XP.

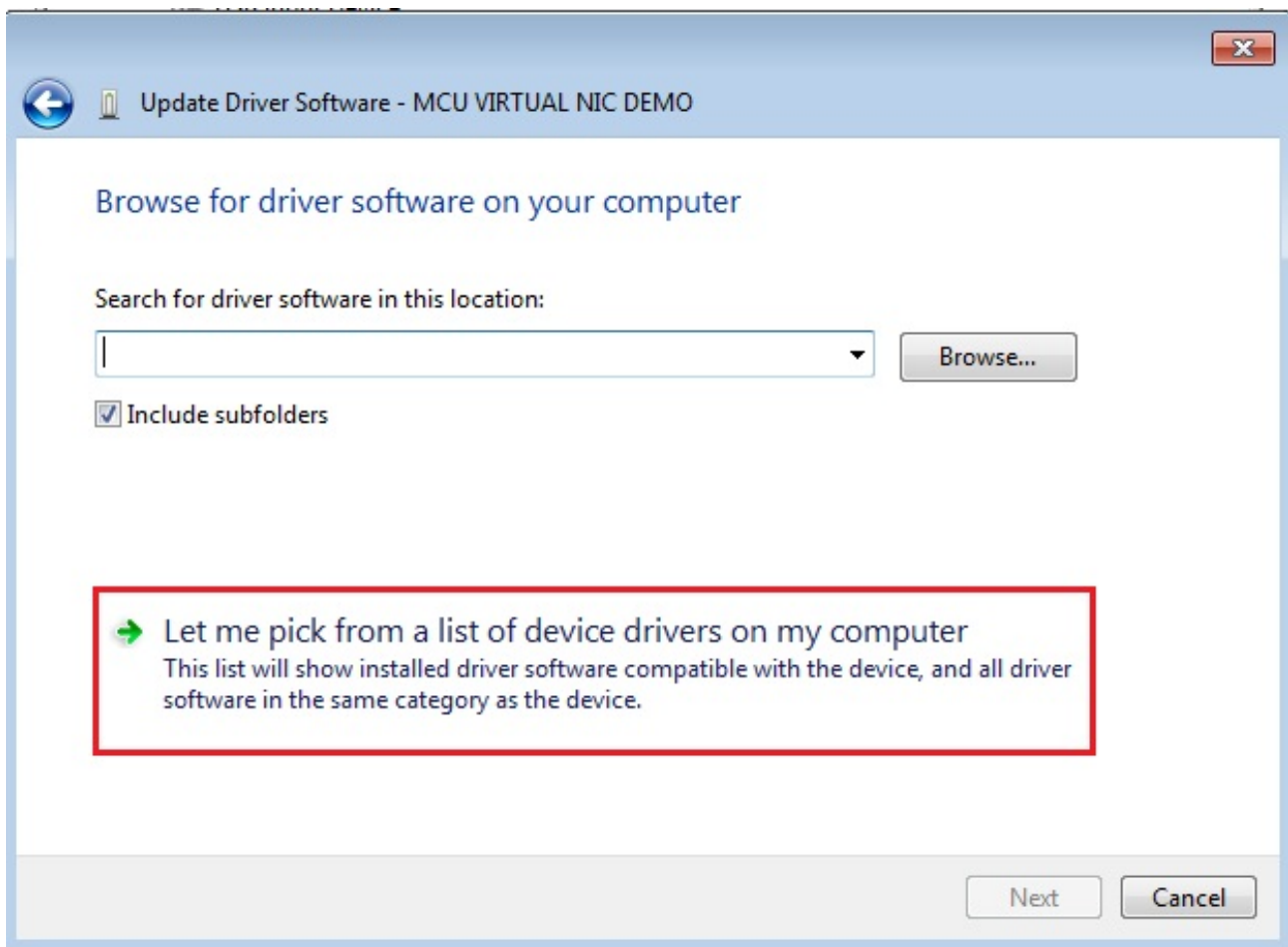
Step 1. Click "Update Driver Software..."



Step 2. Choose "Browse..."



Step 3. Select "Let me pick..."

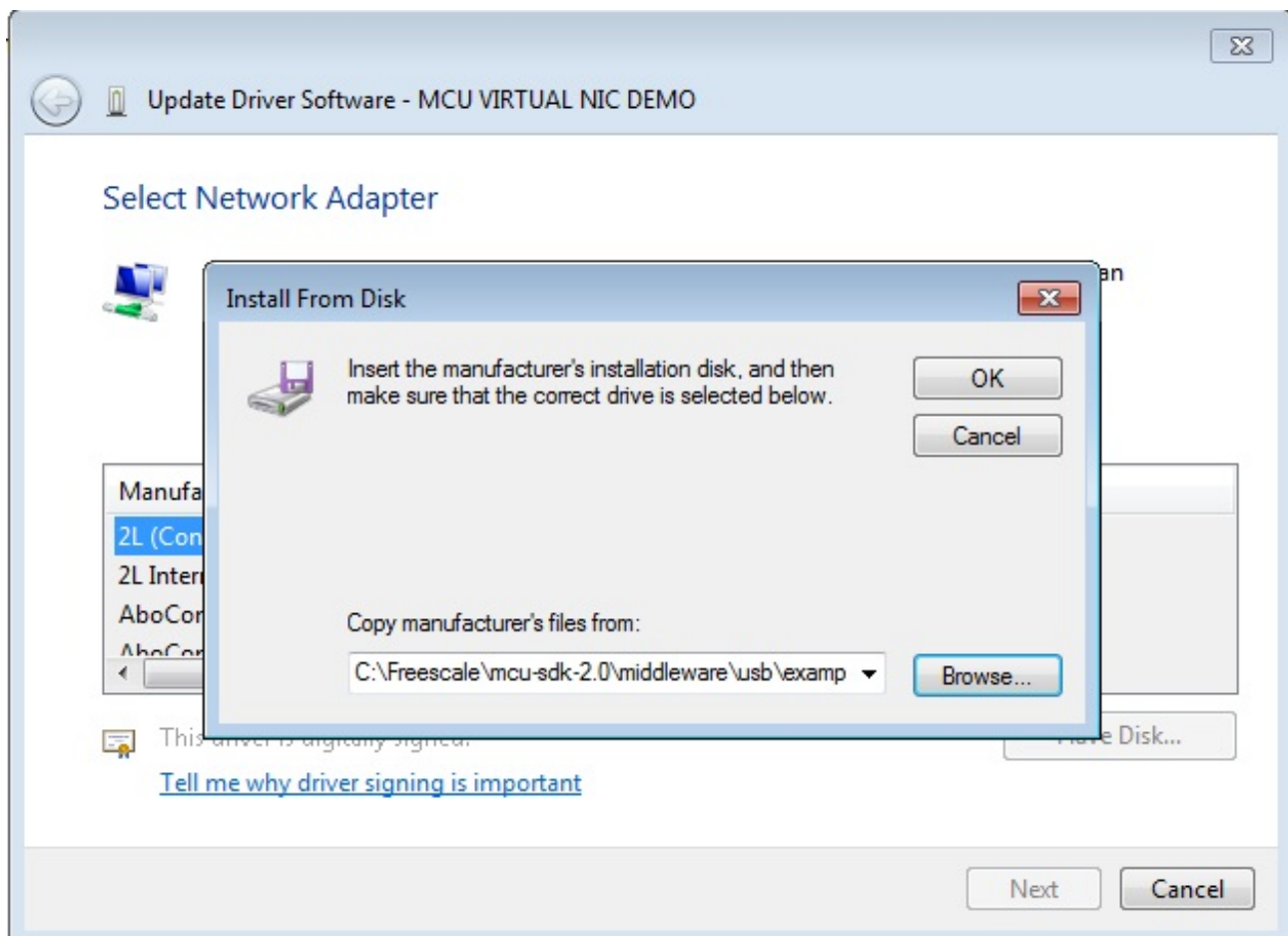


Step 4. Navigate to your RNDIS driver location.

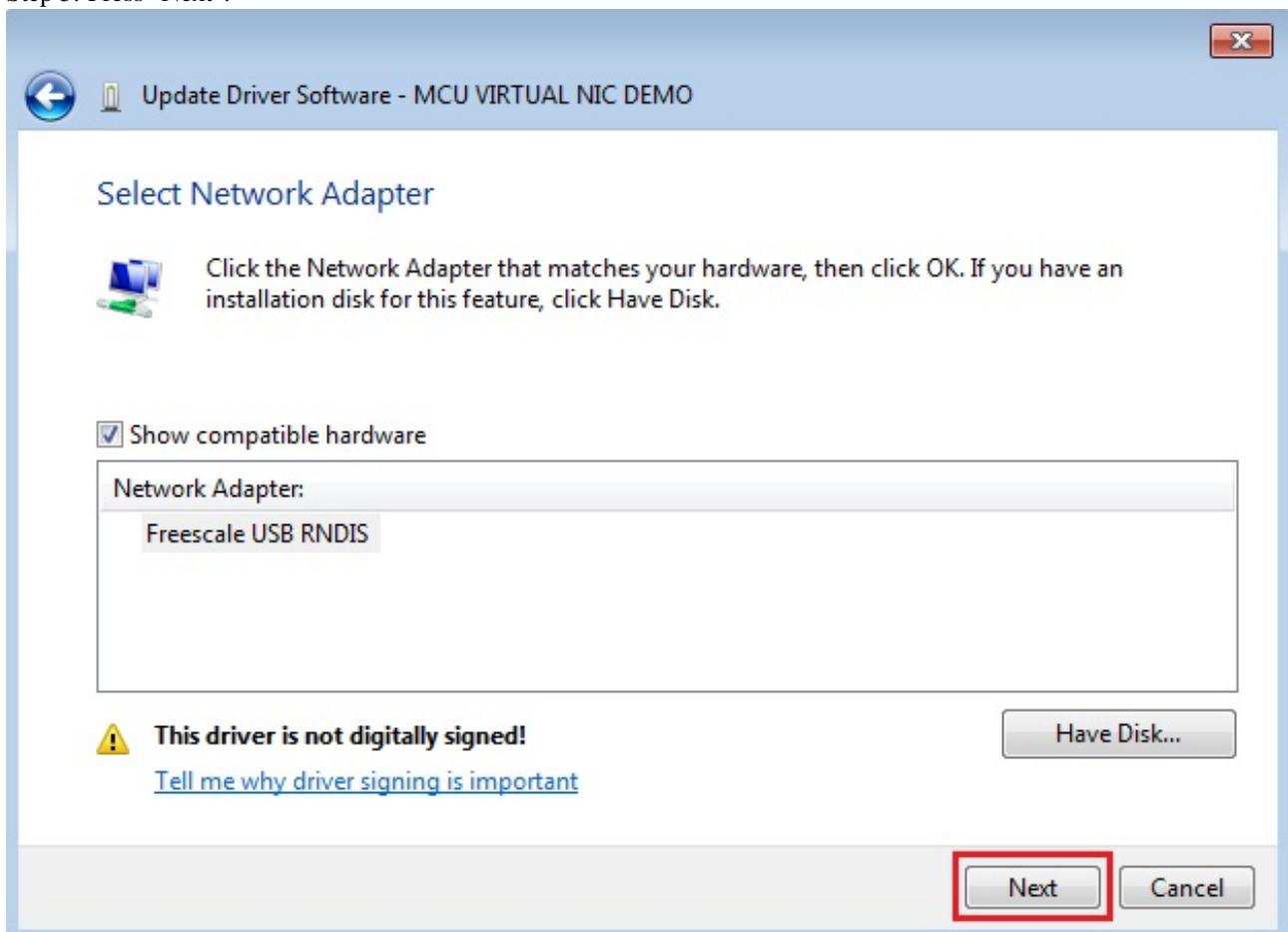
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic\inf

or

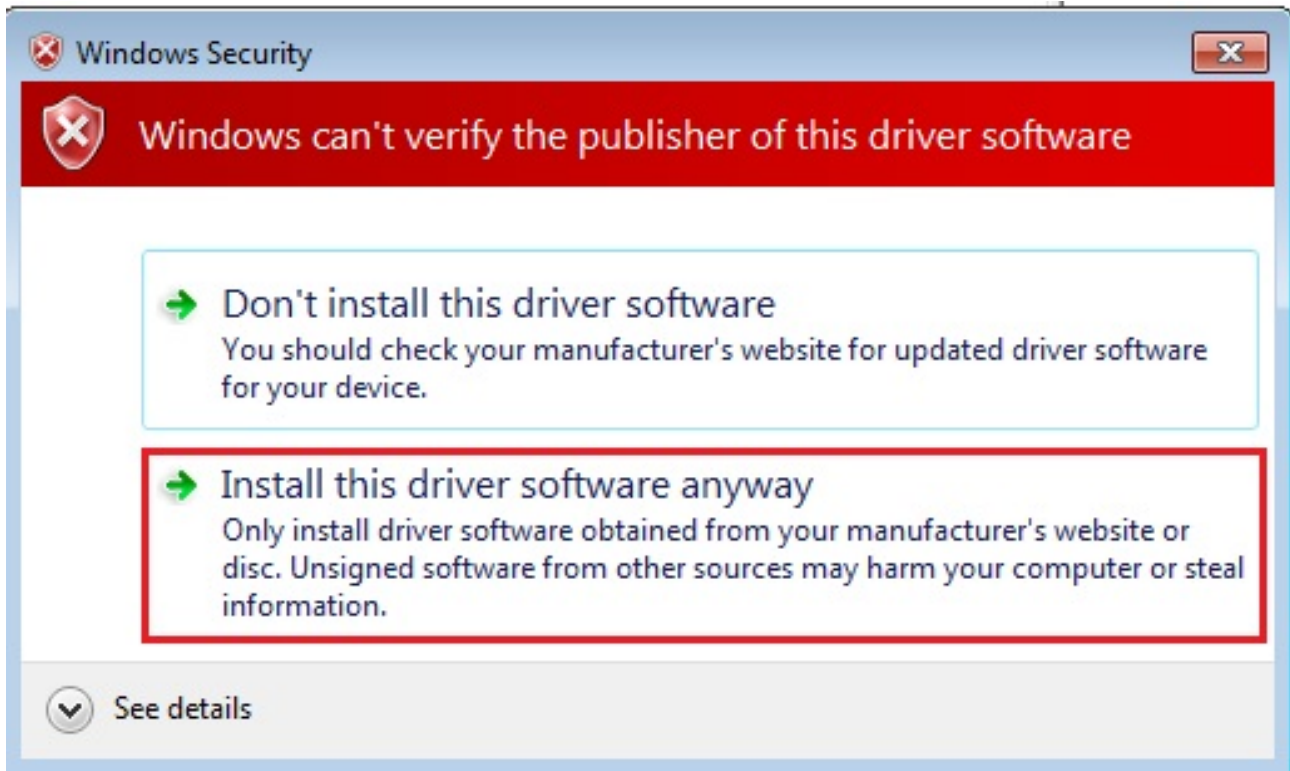
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic_lite\inf



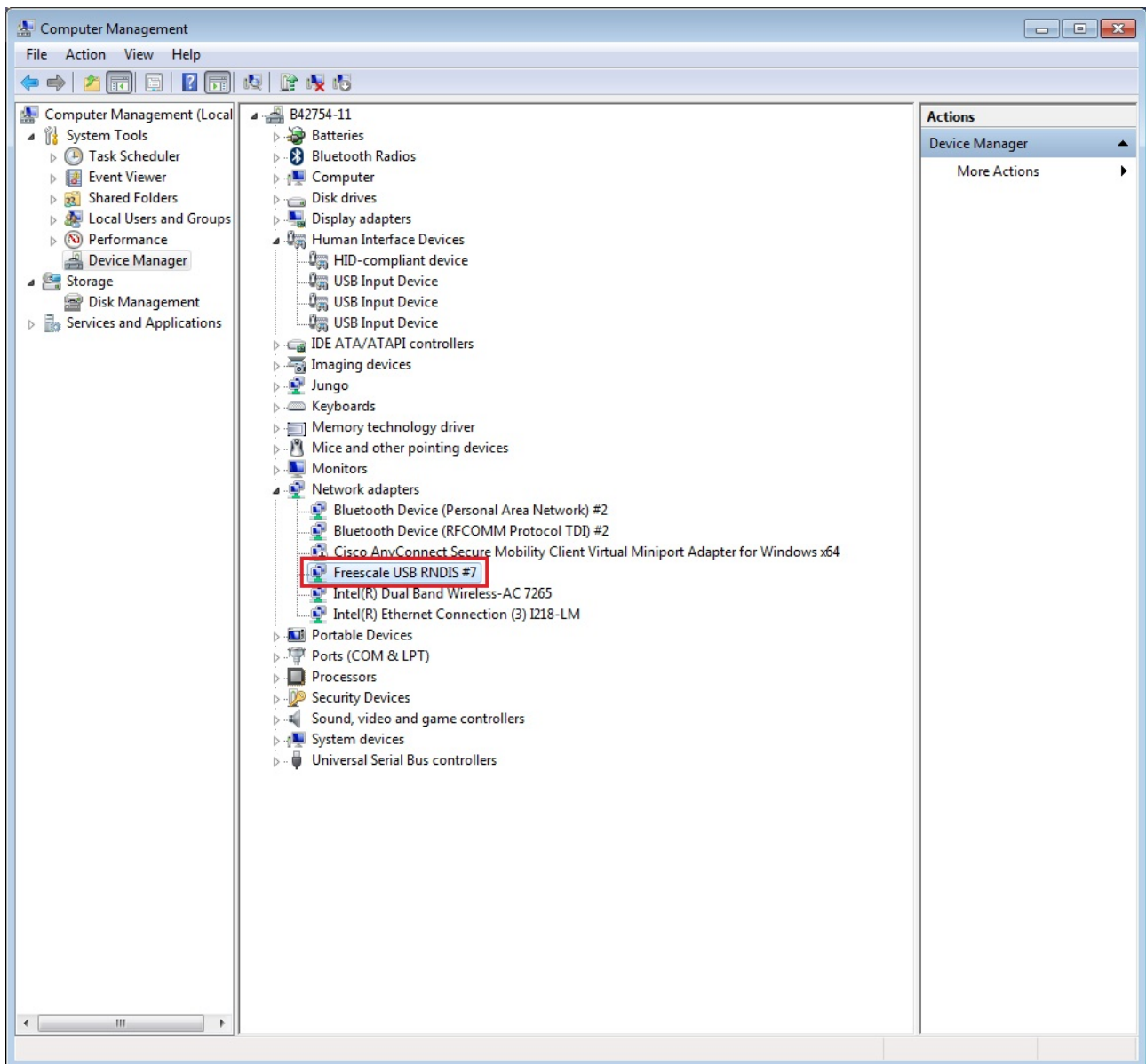
Step 5. Press "Next".



Step 6. Ignore the warning and press "Yes".



Step 7. Now the RNDIS driver should be installed successfully.



- If a driver signature issue occurs on Windows 8 OS, see the link,
Disabling Driver Signature on Windows 8
- To enable driver signing on Windows OS, see the link,
 - Driver Signing
 - Practical Windows Code and Driver Signing