

Overview

The Virtual NIC project is a simple demonstration program based on the MCUXpresso SDK. 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:
`<MCUXpresso_SDK_Install>/boards/<board>/usb_examples/usb_device_cdc_vnic/<rtos>/<toolchain>`.
For lite version, the project files are in:
`<MCUXpresso_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 the USB device port on the board to the host PC and the using a USB cable.

Note

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

Due to the speed mismatch between usb and enet, some enet frame will be discard when lots of data are received in short time, this issue is obvious, especially on Full speed usb.

The upper layer protocol will handle this issue, such as TCP/IP protocol.

Based on the test on TWR-K65, add the buffer number to receive enet frame data could reduce the discard frame, user may add the buffer number if ram is enough.

Run the example in Windows®

1. A network adapter is enumerated in Device Manager.

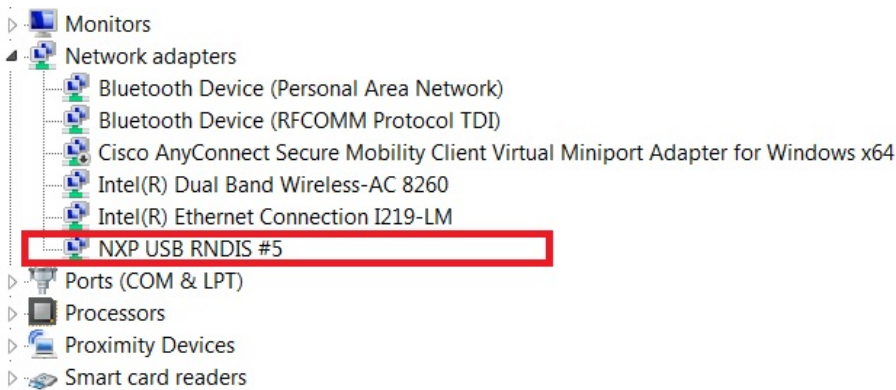


Figure 1: Virtual NIC in device manager

Note

On Windows 10, the device would be identified as an virtual com for the first time.

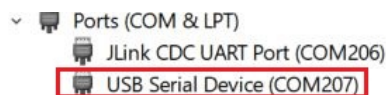


Figure 2: Virtual com in device manager on Windows 10

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



Figure 3: Virtual NIC in Network Connections

3. Connect the ethernet port to another test PC using a network cable. The IP address of this test PC can be configured as 192.168.1.102.

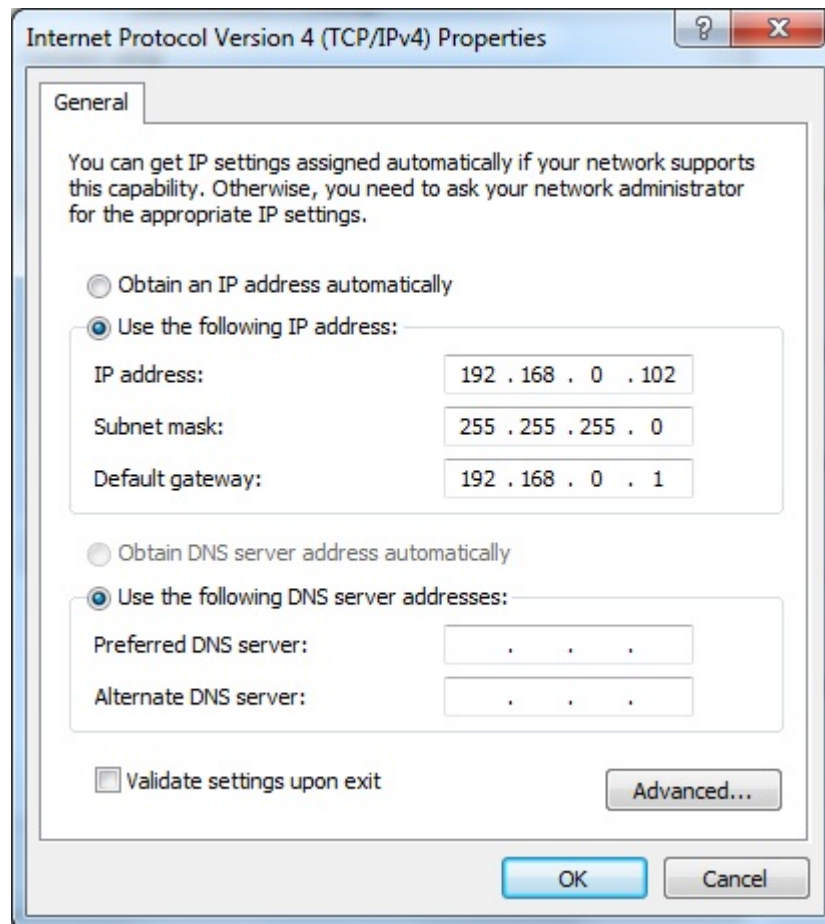


Figure 4: Test PC NIC configuration

4. Then configure the RNDIS adapter in Windows for your host PC. For this testing example, please make sure your PC host and the test PC are in the same network segment. E.g

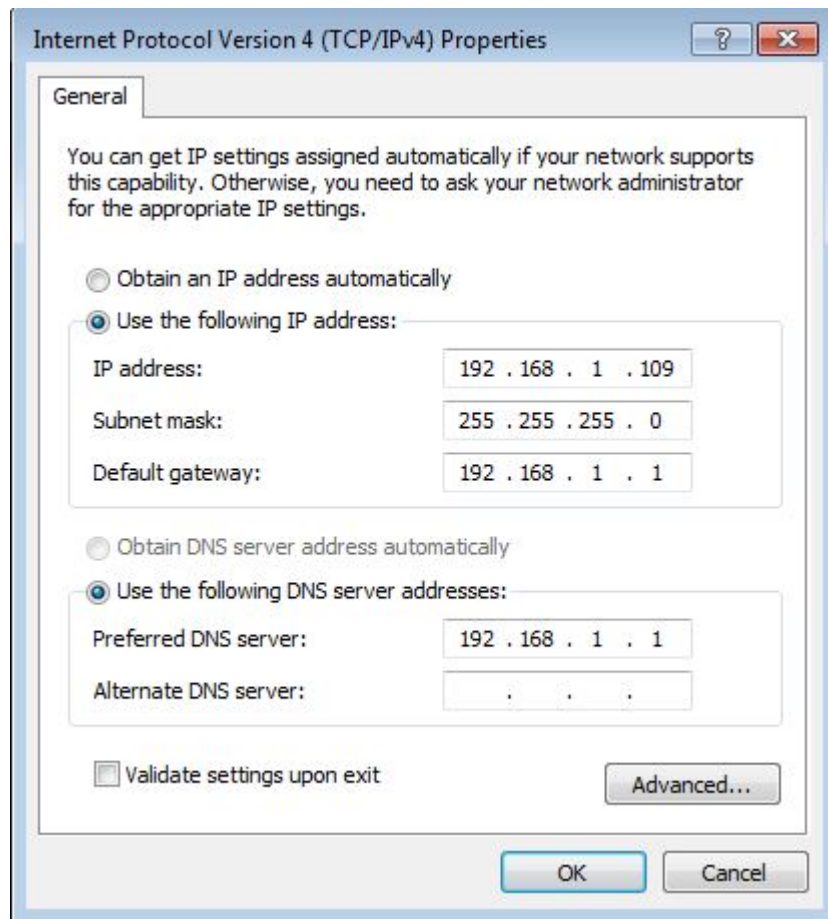


Figure 5: Host PC Virtual NIC configuration

5. Then you are supposed to access the test PC through this USB RNDIS network adapter from your host PC

```
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 6: Ping to other PC

Note

1. The VNIC CIC CLASS code can be changed to be miscellaneous device to implement that there is no need to install inf file by changing the codes for descriptor. Provide the following codes macros for descriptor to change its related codes:

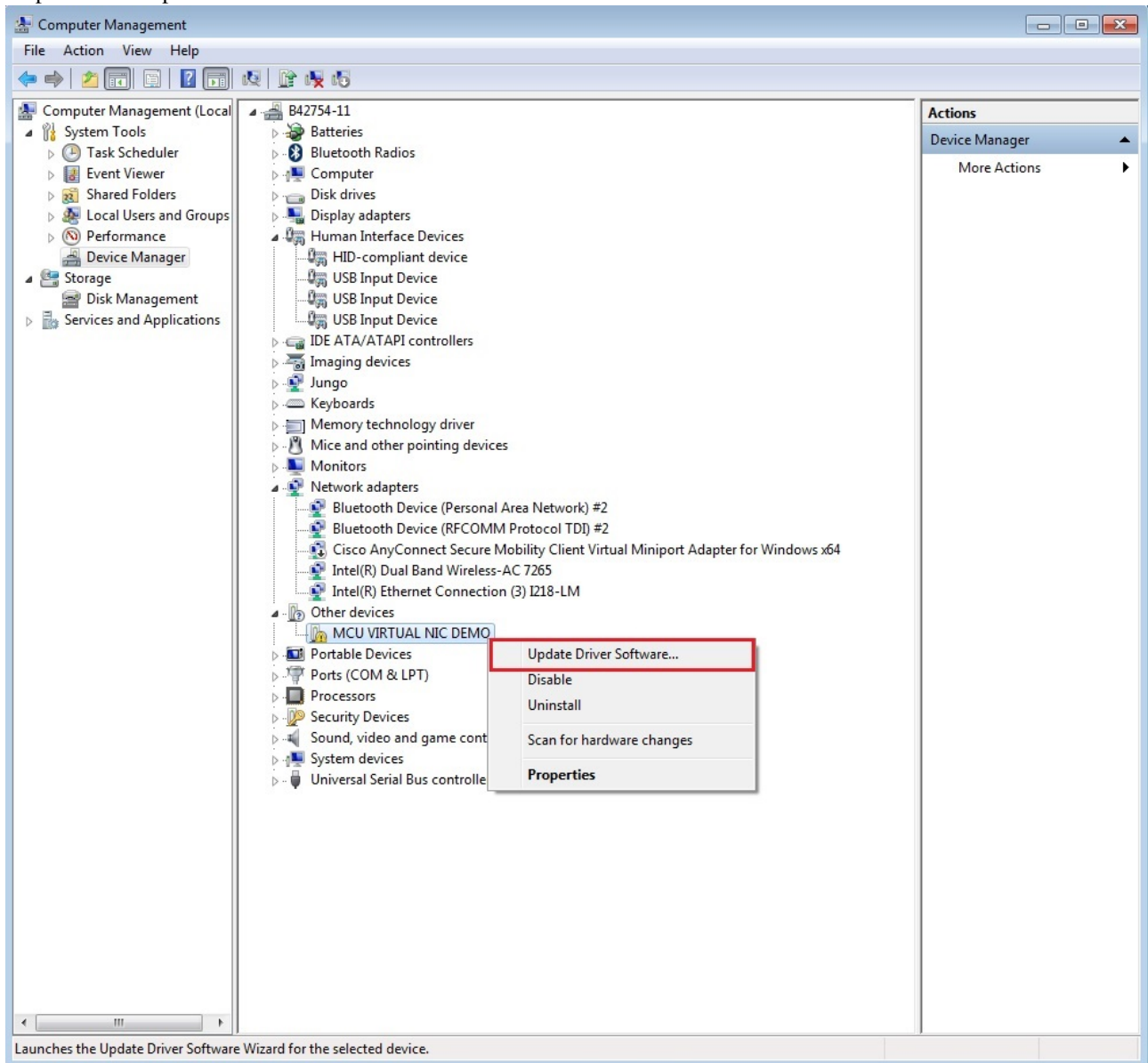
```
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_CLASS (0xEF)
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_SUBCLASS (0x04)
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_PROTOCOL (0x01)
```

Note that the above change is only applied for descriptor, please do not change the codes of class driver. This is not verified under all conditions, so please be cautious to change.

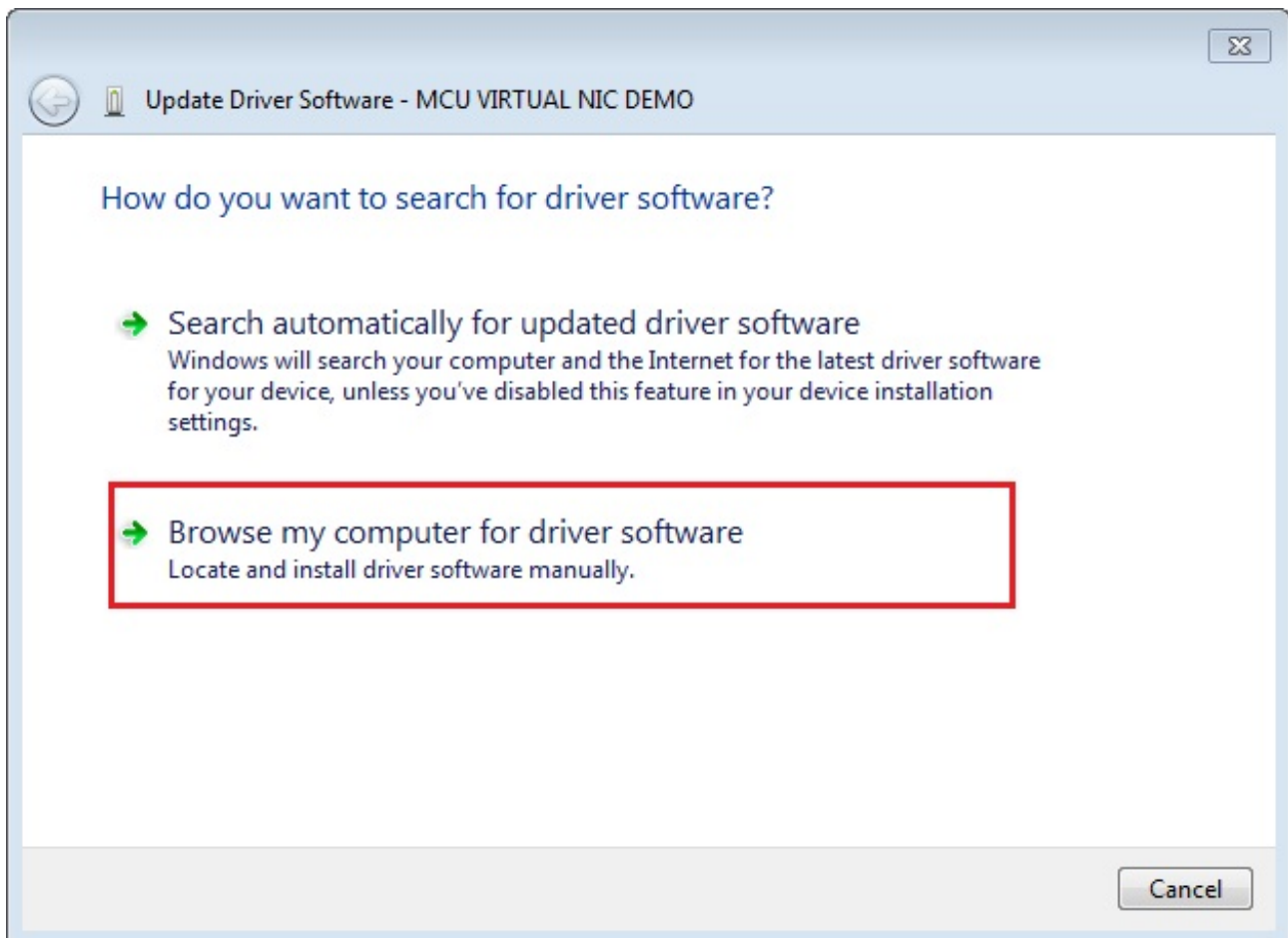
Installing the RNDIS driver for virtual_nic example

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

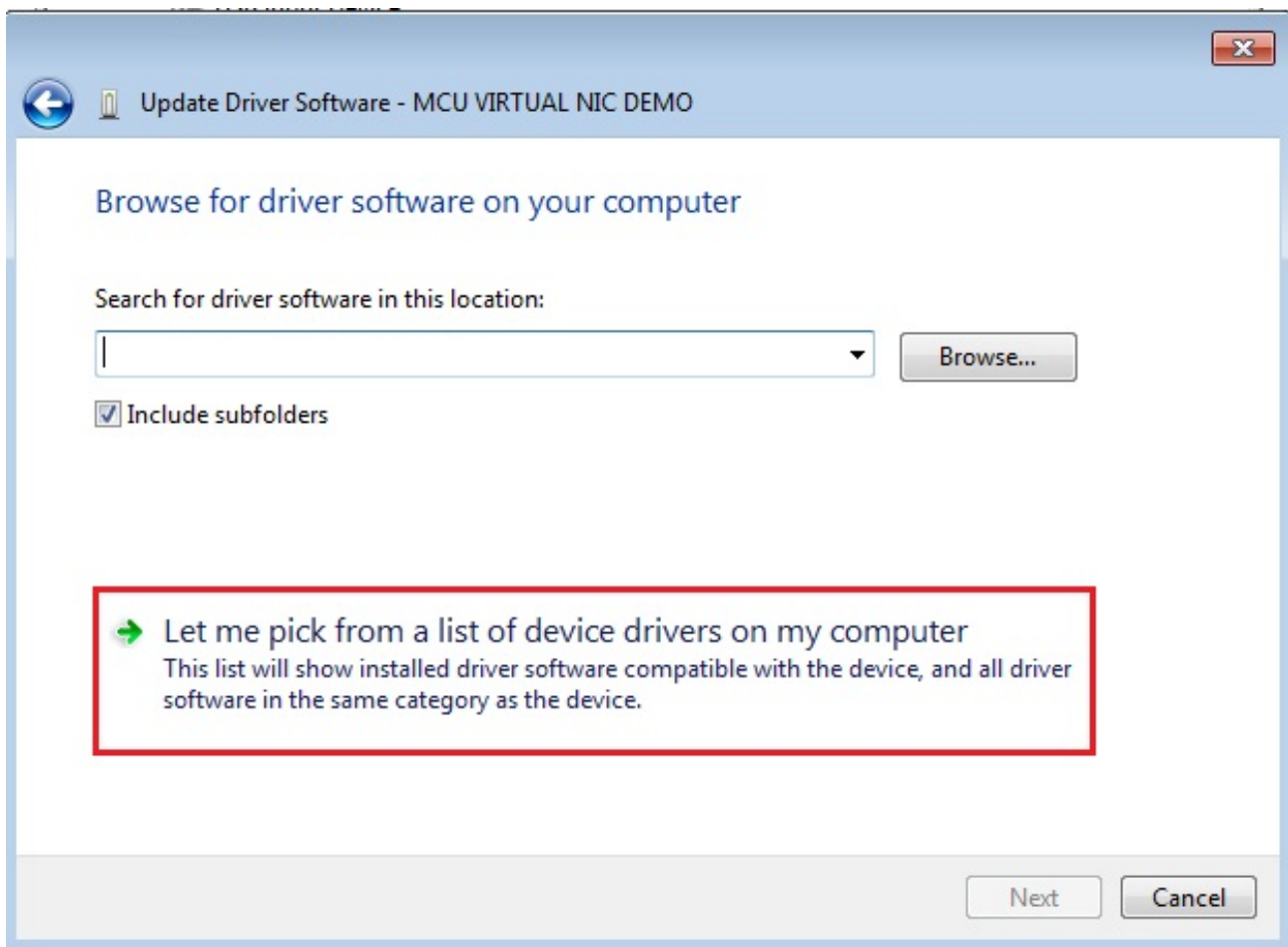
Step 1. Click "Update Driver Software..."



Step 2. Choose "Browse..."



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

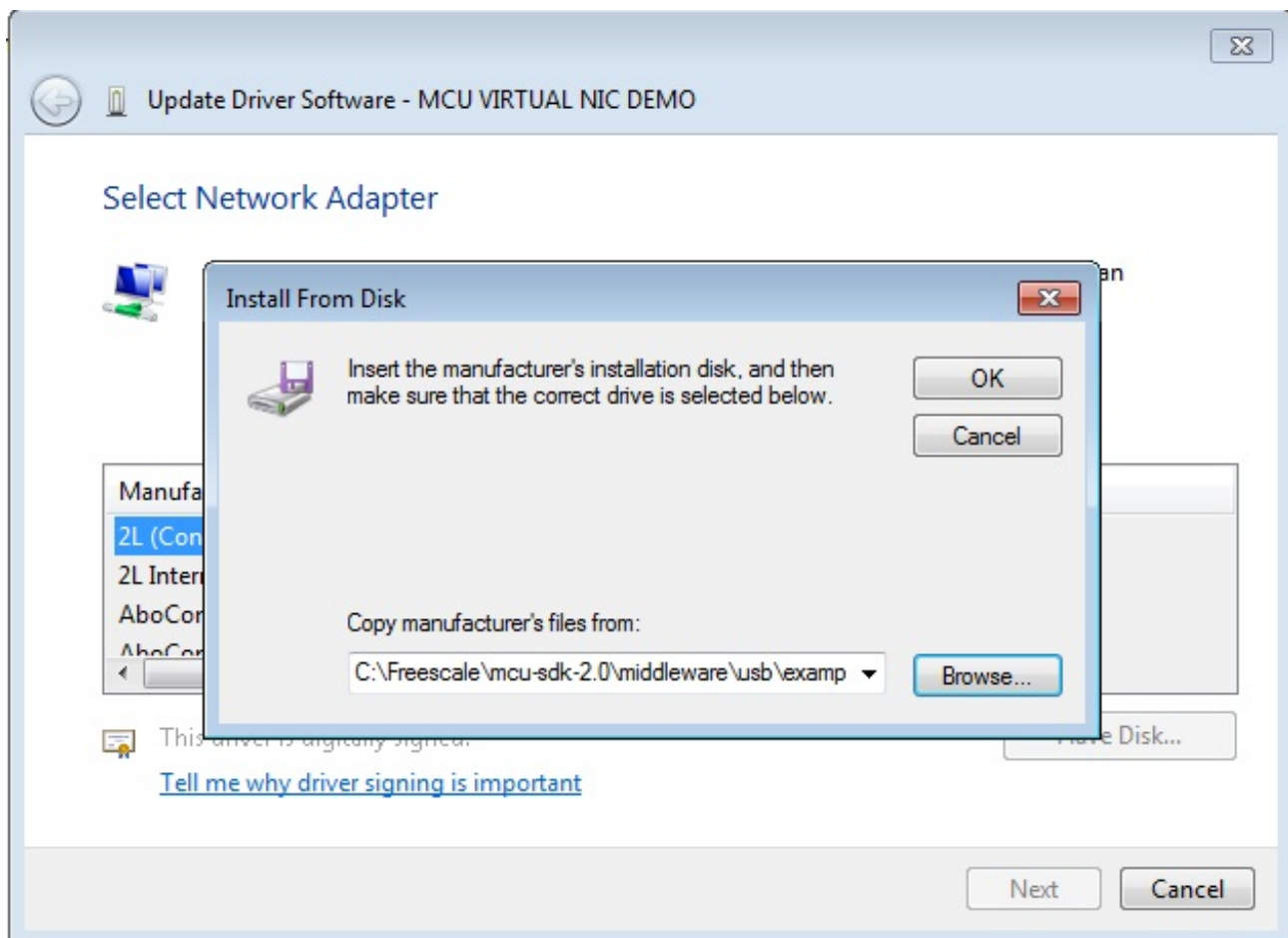


Step 4. Navigate to your RNDIS driver location.

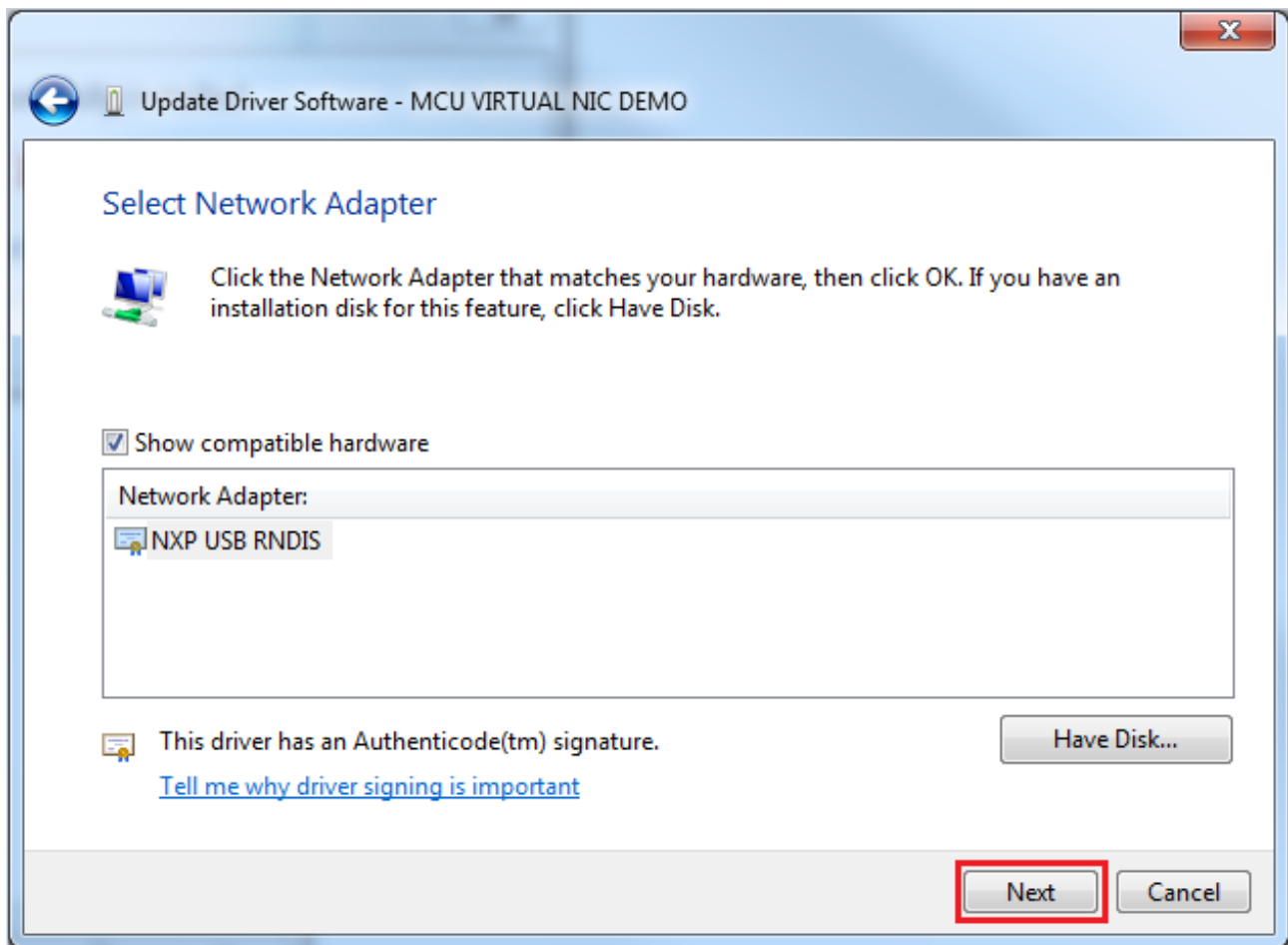
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic\inf

or

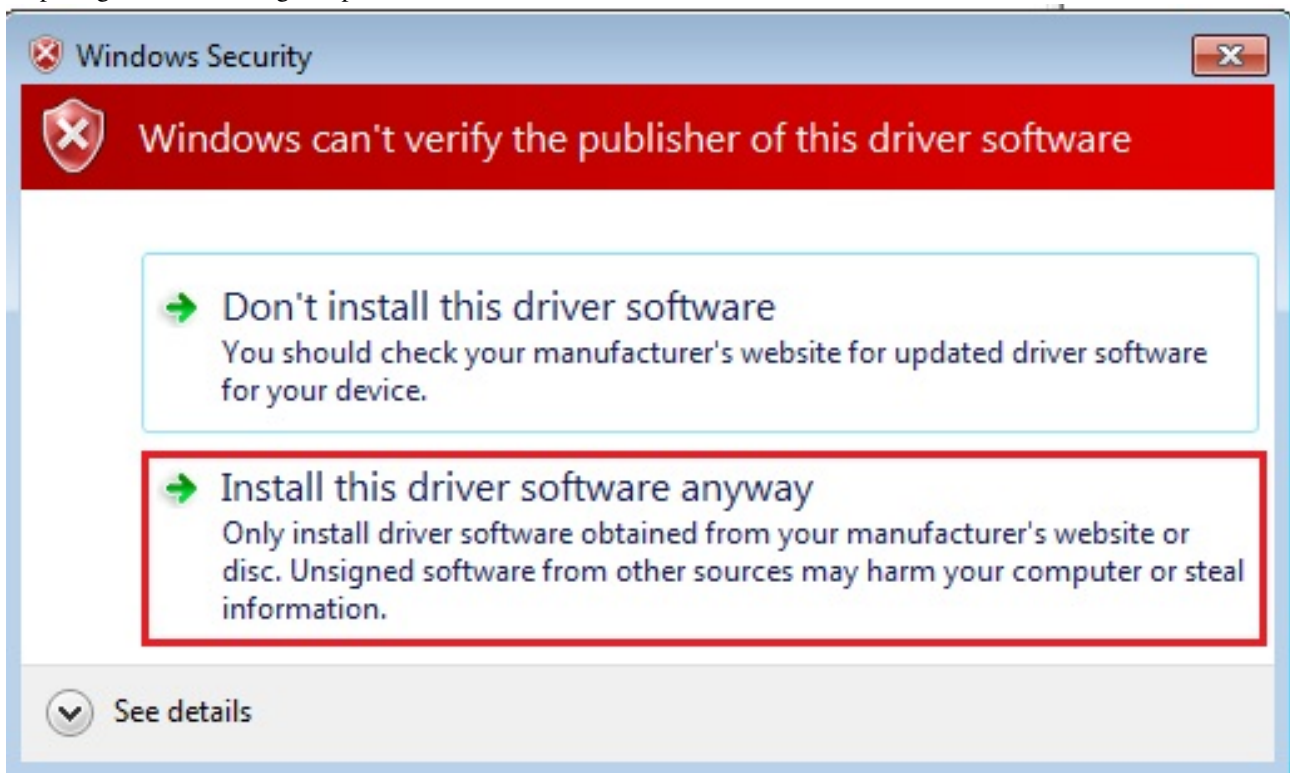
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic_lite\inf



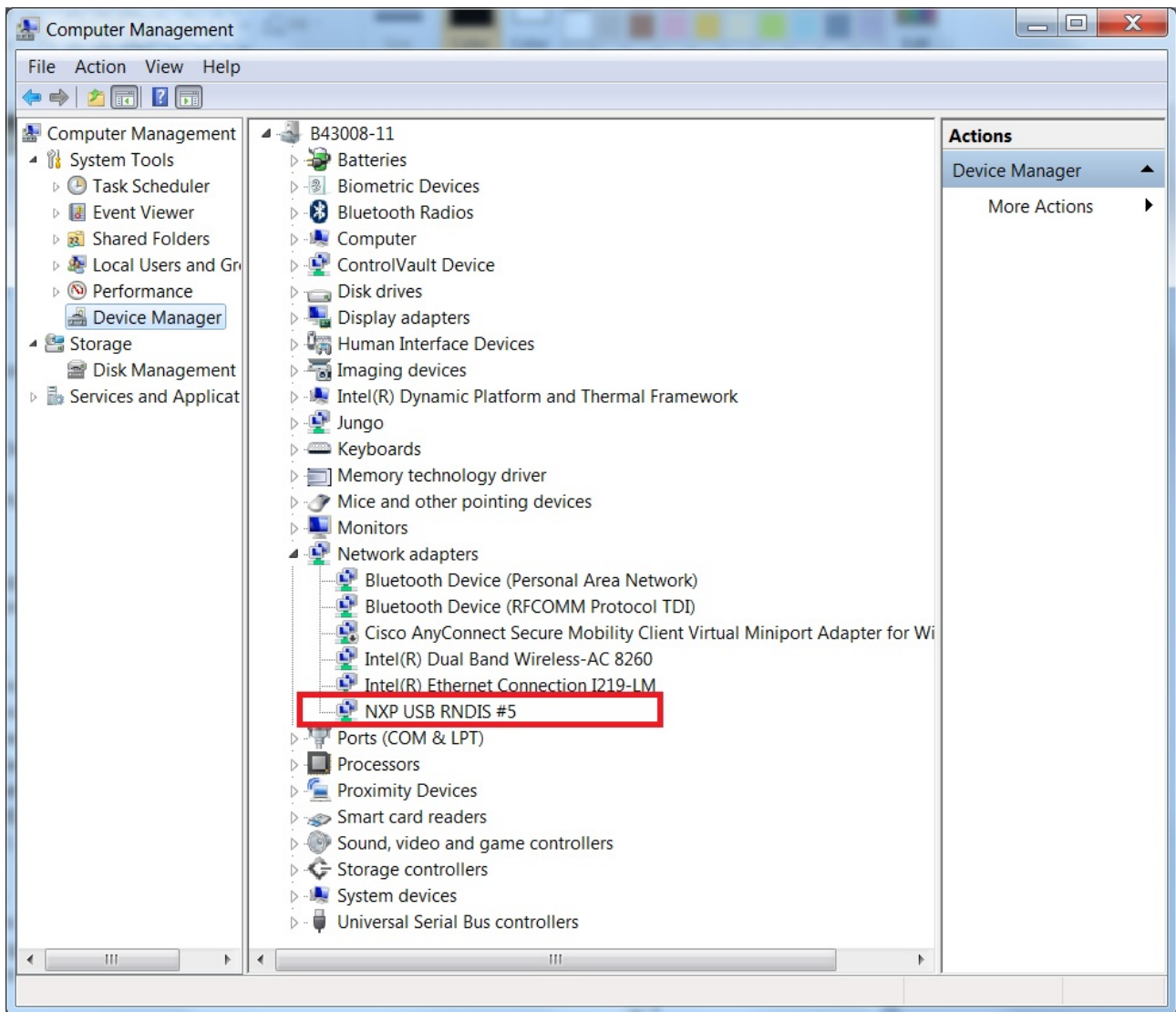
Step 5. Press "Next".



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

This example application demonstrates the USB CDC Virtual NIC example.

Overview

The Virtual NIC project is a simple demonstration program based on the MCUXpresso SDK. 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:
`<MCUXpresso_SDK_Install>/boards/<board>/usb_examples/usb_device_cdc_vnic/<rtos>/<toolchain>.`
 For lite version, the project files are in:
`<MCUXpresso_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

- Connect JP13 2-3. Populate R274 to sync reference clock.

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 the USB device port on the board to the host PC and the using a USB cable.

Note

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

Due to the speed mismatch between usb and enet, some enet frame will be discard when lots of data are received in short time, this issue is obvious, especially on Full speed usb.

The upper layer protocol will handle this issue, such as TCP/IP protocol.

Based on the test on TWR-K65, add the buffer number to receive enet frame data could reduce the discard frame, user may add the buffer number if ram is enough.

Run the example in Windows[®]

1. A network adapter is enumerated in Device Manager.

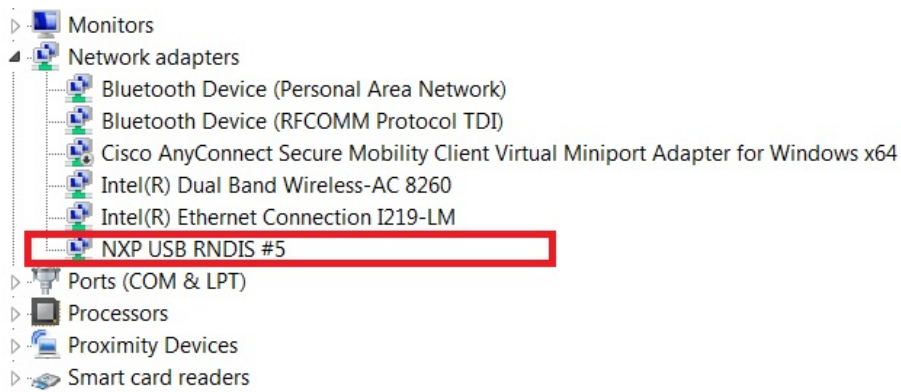


Figure 7: Virtual NIC in device manager

Note

On Windows 10, the device would be identified as an virtual com for the first time.

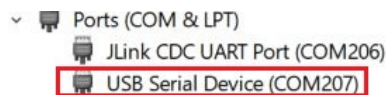


Figure 8: Virtual com in device manager on Windows 10

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



Figure 9: Virtual NIC in Network Connections

3. Connect the ethernet port to another test PC using a network cable. The IP address of this test PC can be configured as 192.168.1.102.

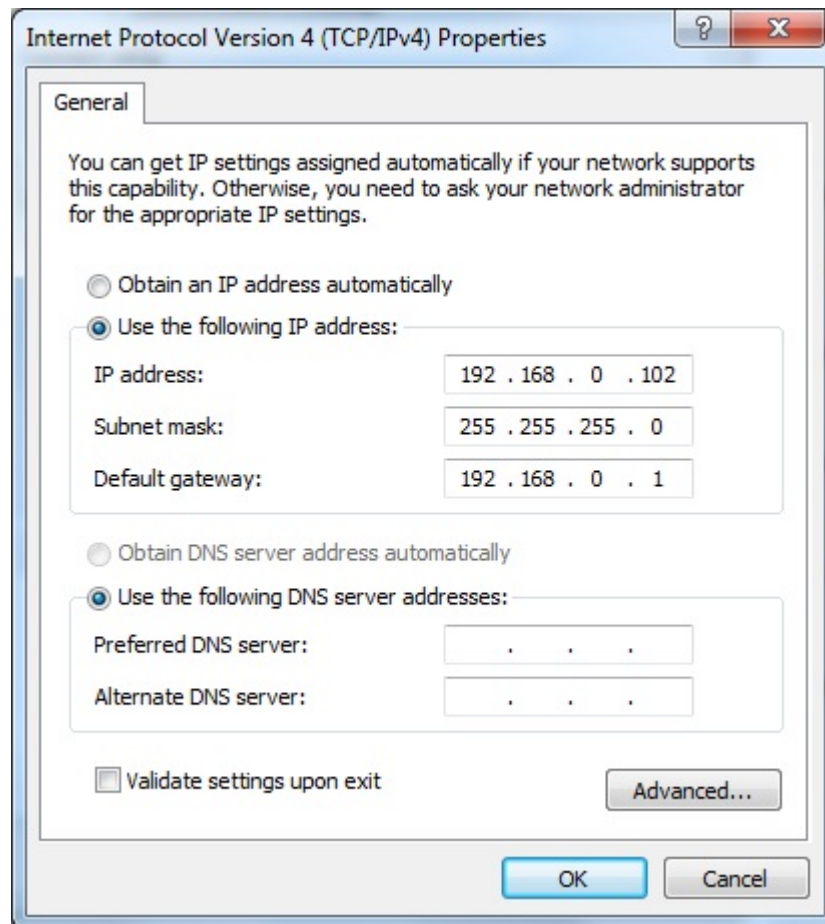


Figure 10: Test PC NIC configuration

4. Then configure the RNDIS adapter in Windows for your host PC. For this testing example, please make sure your PC host and the test PC are in the same network segment. E.g

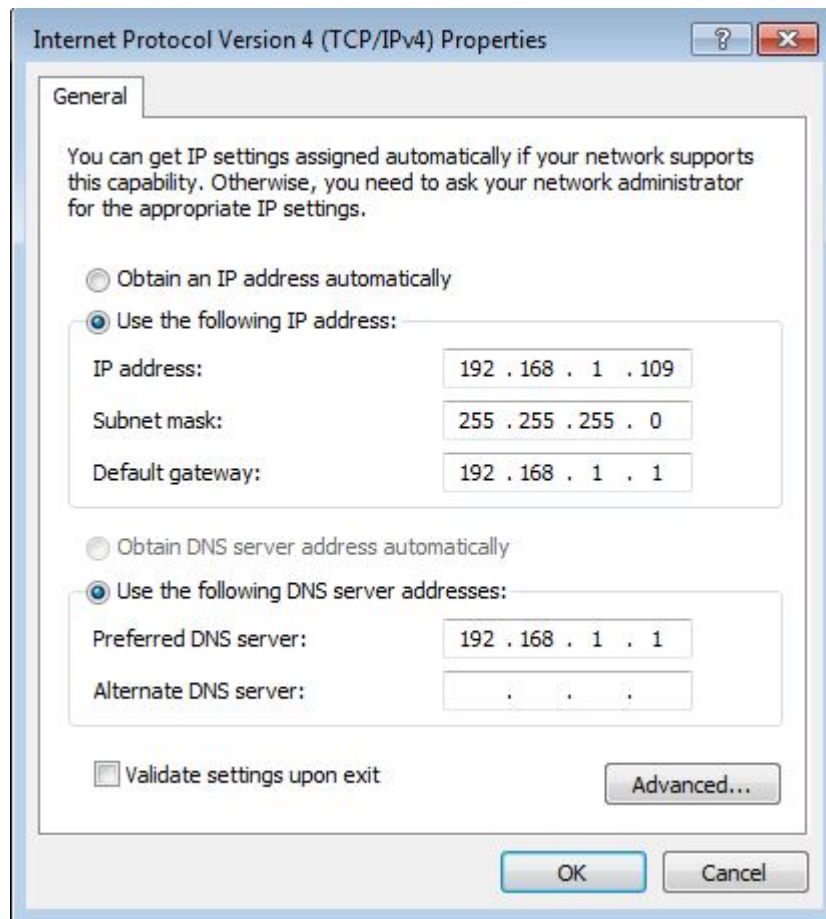


Figure 11: Host PC Virtual NIC configuration

5. Then you are supposed to access the test PC through this USB RNDIS network adapter from your host PC

```
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 12: Ping to other PC

Note

1. The VNIC CIC CLASS code can be changed to be miscellaneous device to implement that there is no need to install inf file by changing the codes for descriptor. Provide the following codes macros for descriptor to change its related codes:

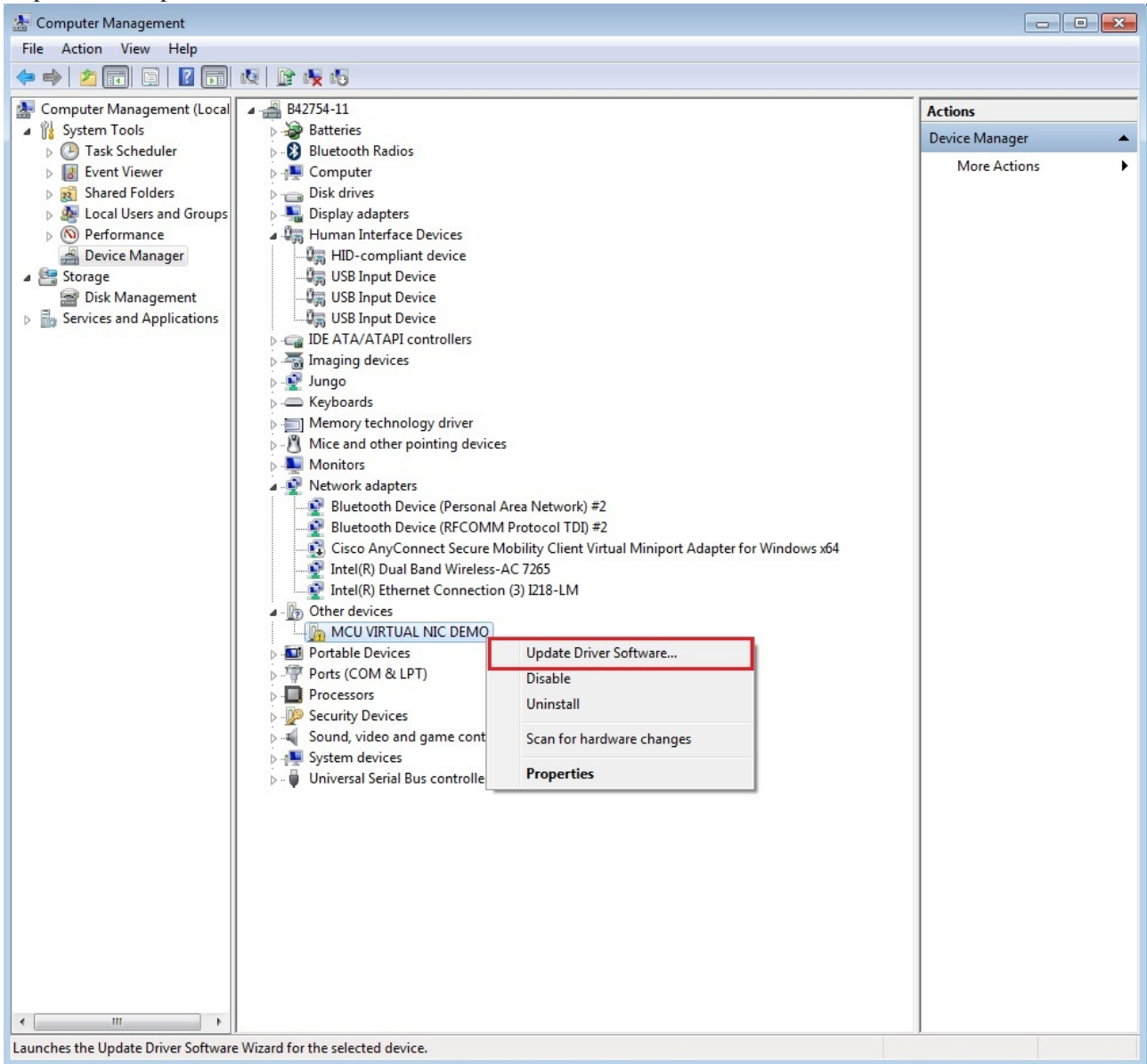
```
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_CLASS (0xEF)
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_SUBCLASS (0x04)
#define USB_CDC_IFACE_DESCRIPTOR_VNIC_CIC_PROTOCOL (0x01)
```

Note that the above change is only applied for descriptor, please do not change the codes of class driver. This is not verified under all conditions, so please be cautious to change.

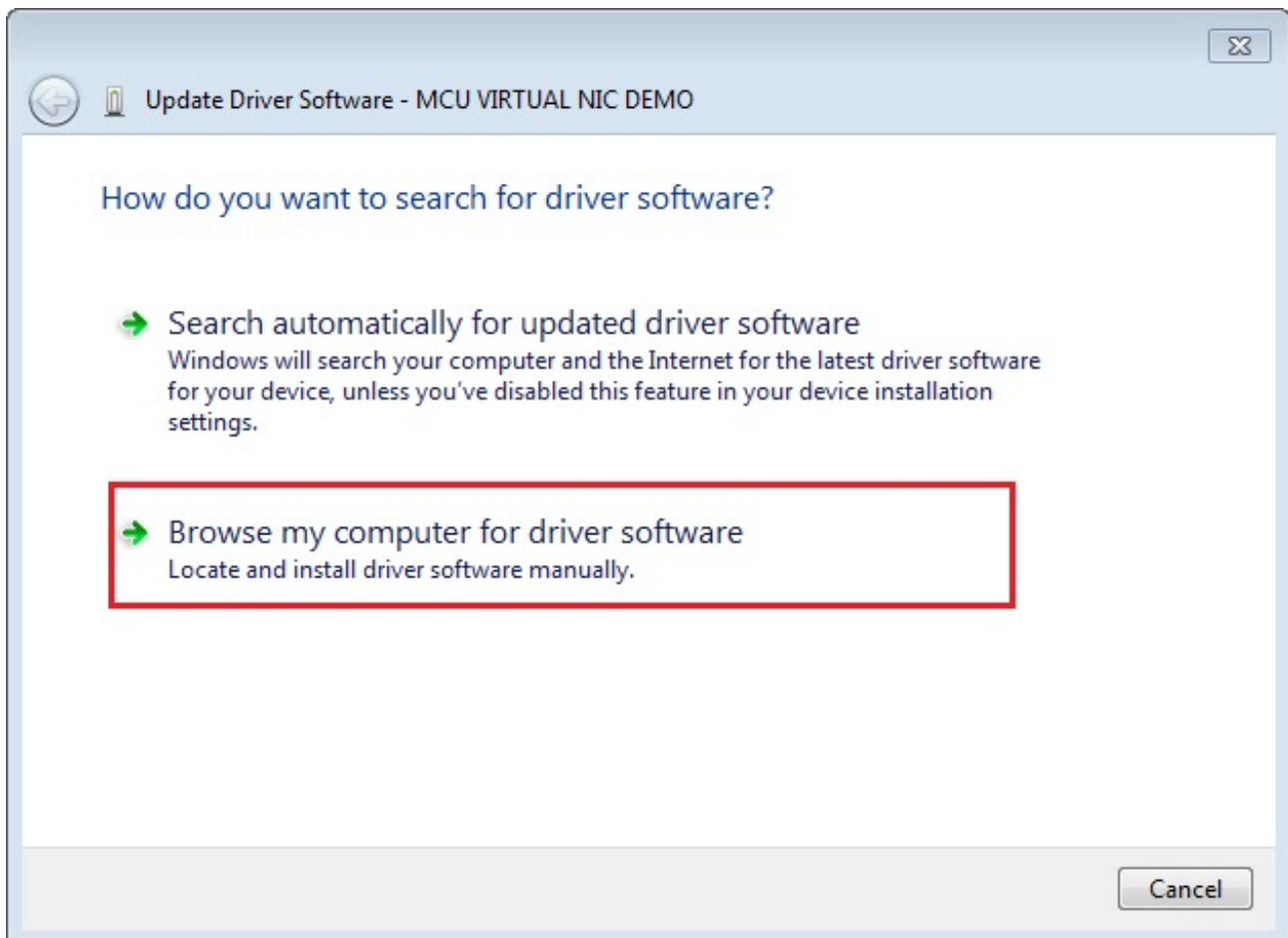
Installing the RNDIS driver for virtual_nic example

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

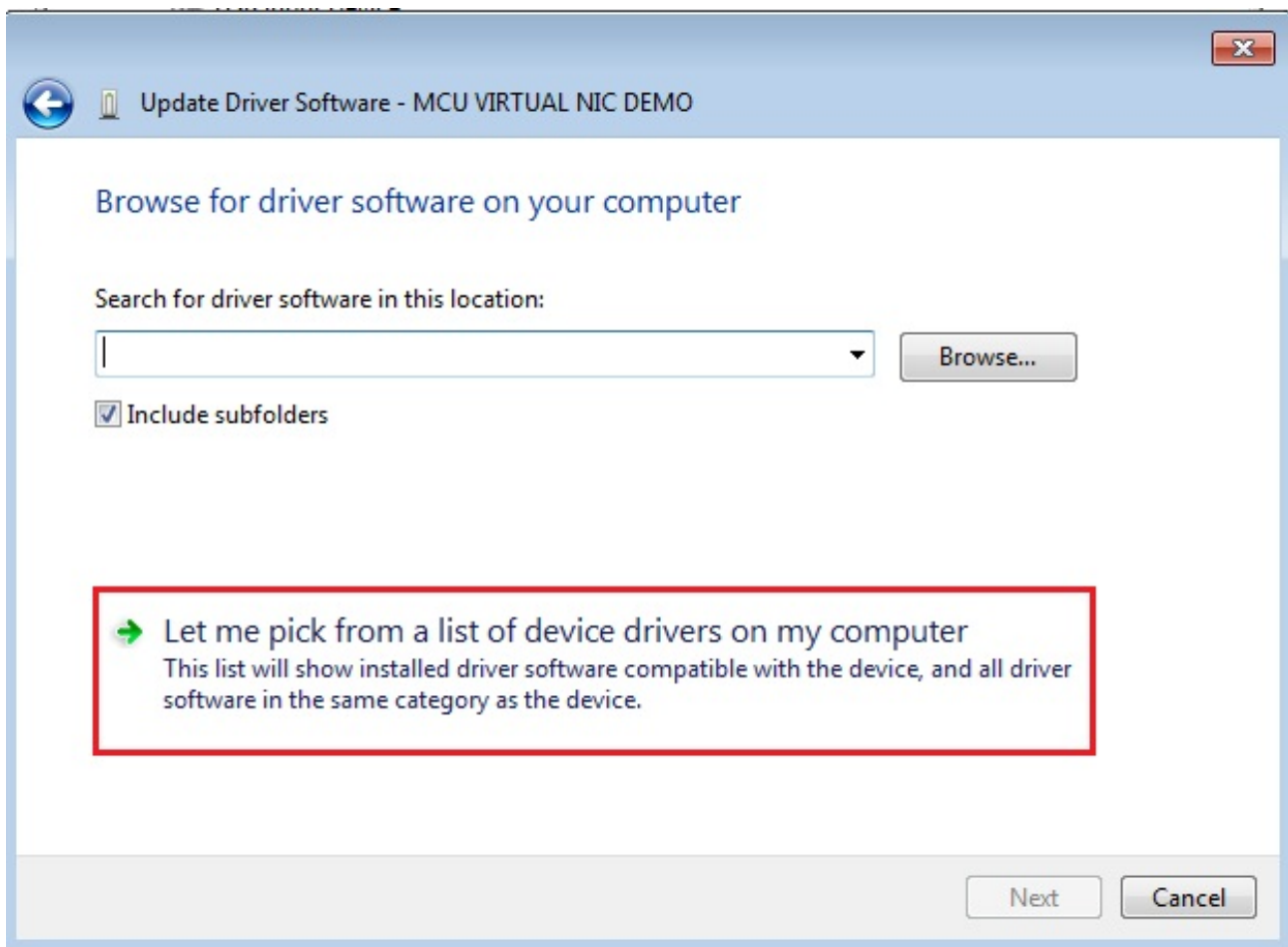
Step 1. Click "Update Driver Software..."



Step 2. Choose "Browse..."



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

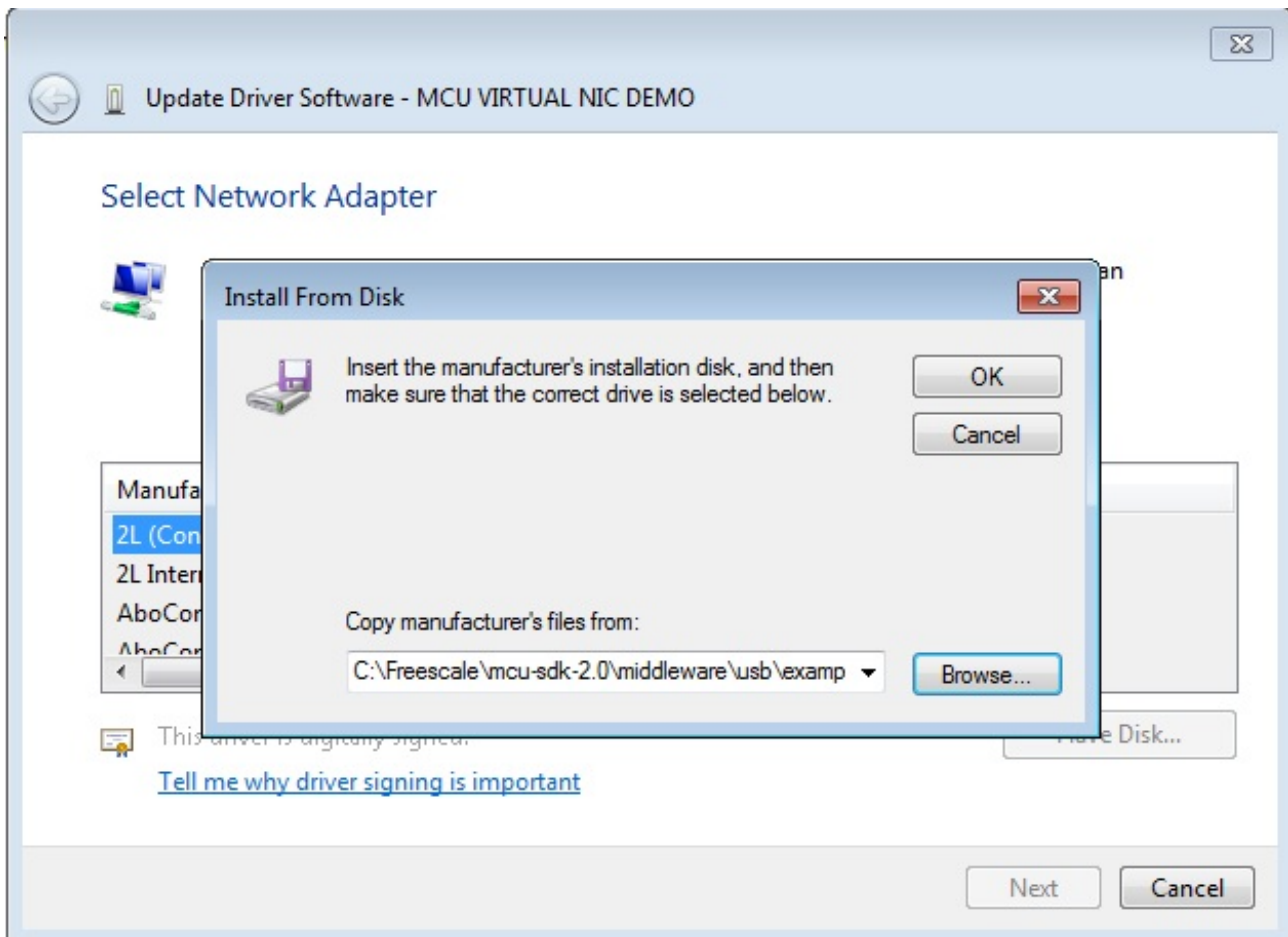


Step 4. Navigate to your RNDIS driver location.

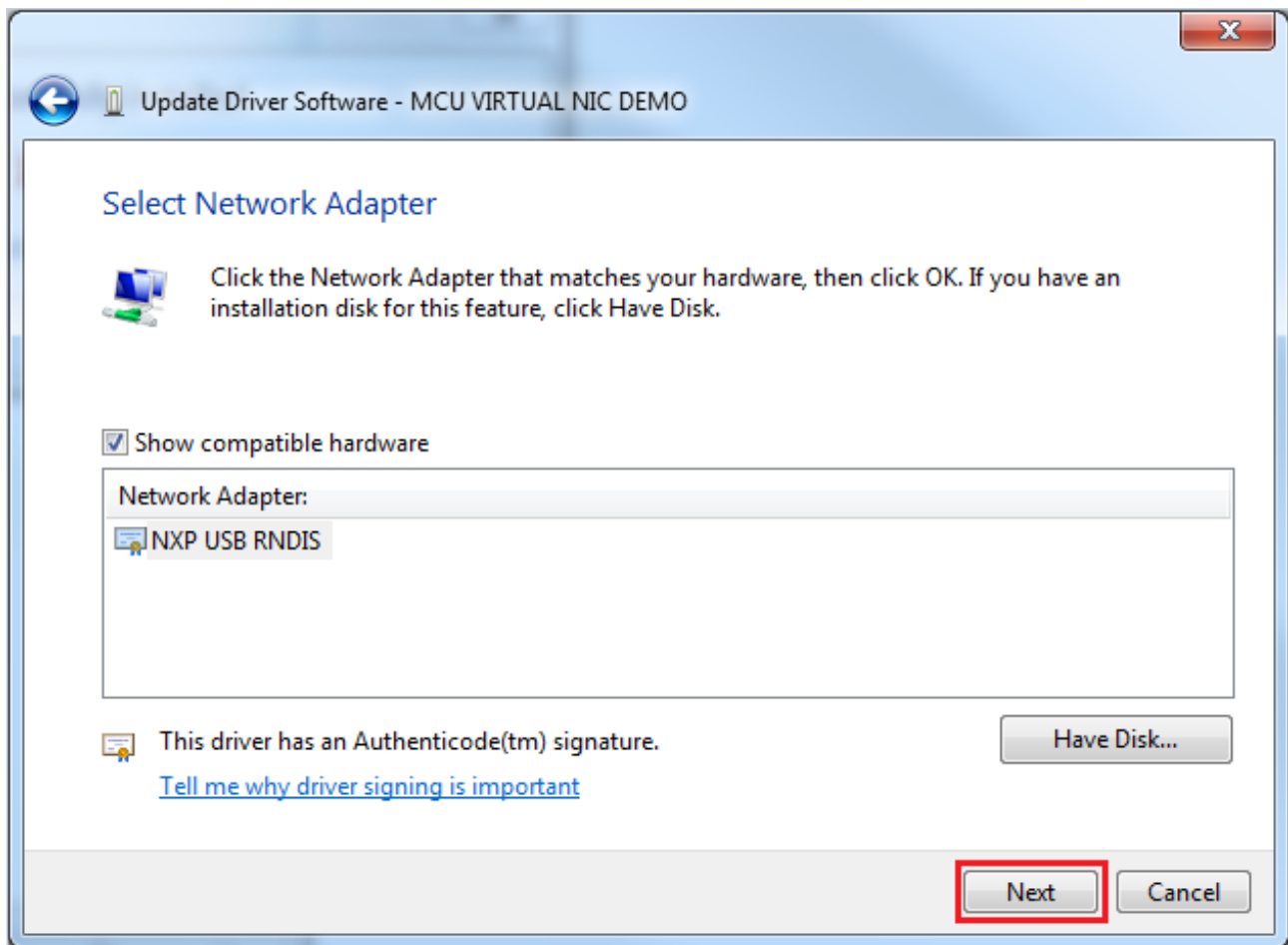
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic\inf

or

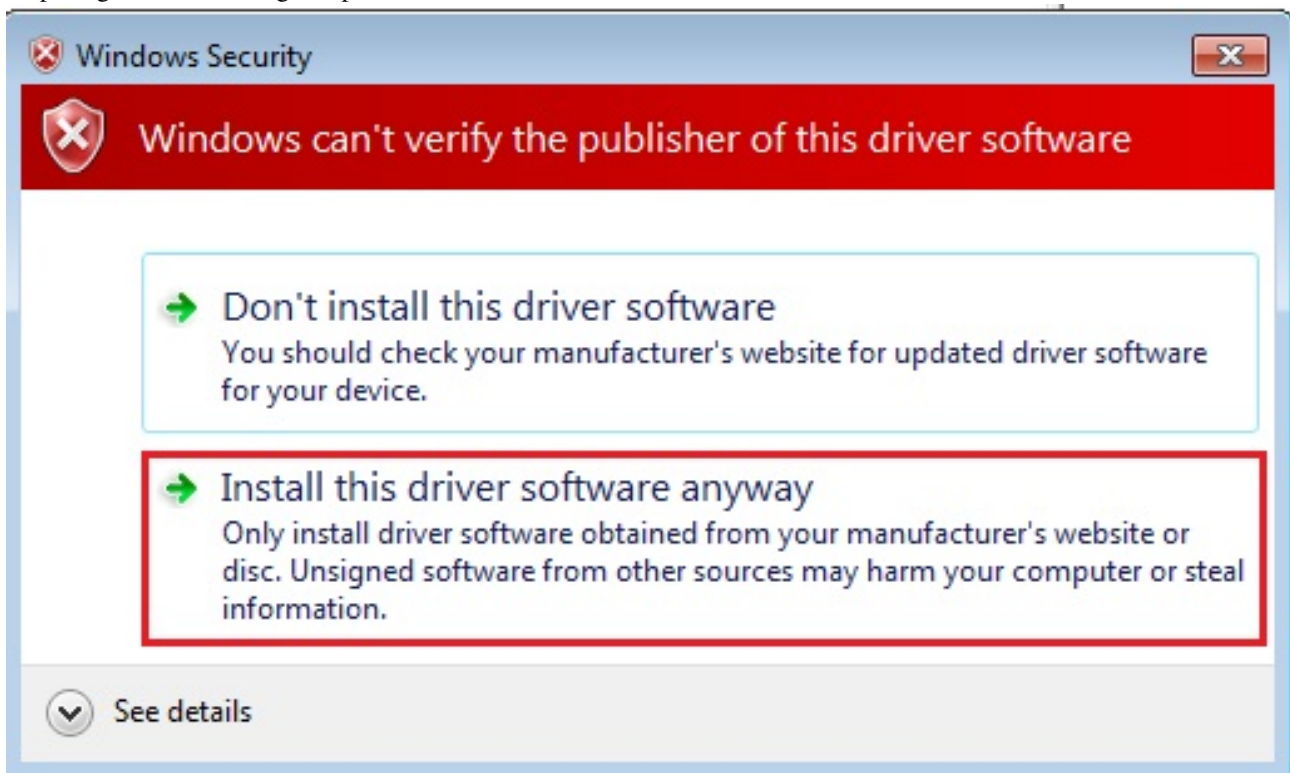
<install_dir>\boards\<board>\usb_examples\usb_device_cdc_vnic_lite\inf



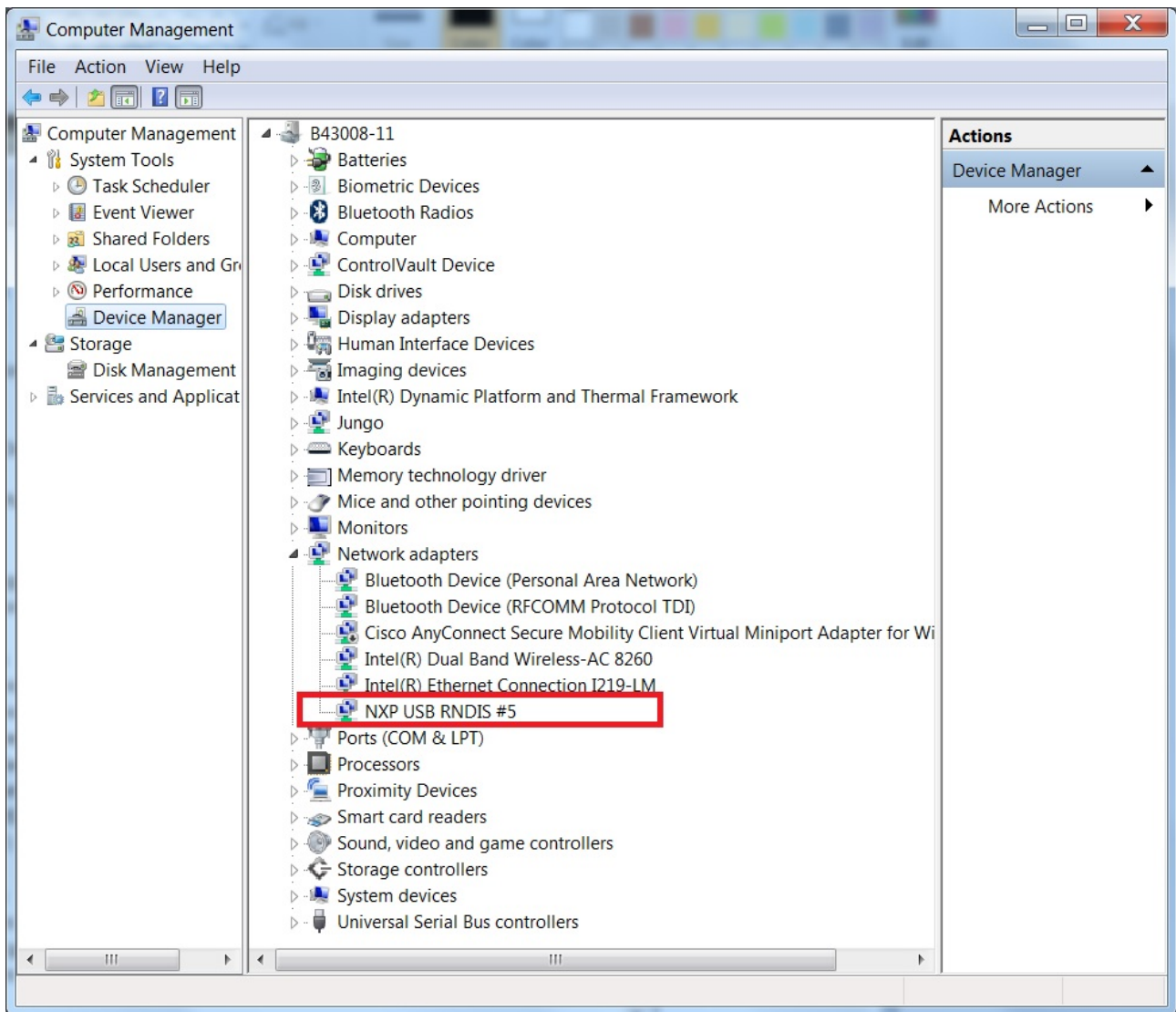
Step 5. Press "Next".



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](#)