

CST8244 Lab 1: Setup Your QNX Development Host

Lab Objectives:

1. To create a myQNX account
2. To install the QNX Software Center for your host environment
3. To install the QNX Software Development Platform (SDP)

Before You Begin:

Question:

Do you have VMware Workstation (PC) or VMware Fusion (Mac) already installed and configured on your bring your own device (BYOD)?

YES – proceed with the lab but know that you may need to update your version of VMware

NO – check for the notification email sent to you by the IT department. That will contain all the information regarding how to get the license and install VMware. (Note: getting a license can take anywhere from a few hours to 2-3 days)

The latest versions of VMware you need:

Windows PC ☐ VMware Workstation 16.x Pro Mac

OSX ☐ VMware Fusion 12 Pro

Readings:

- This lab is based on the QNX Quickstart Guide
<http://www.qnx.com/developers/docs/7.0.0/index.html#com.qnx.doc.qnxsdg.quickstart/topic/about.html>

Part A: Create a myQNX Account

1. Create your own myQNX account at:

<https://www.qnx.com/account/login.html>

Notes:

- Use your Algonquin College email address
- Remember your myQNX password
- You'll receive an email from QNX with the Subject: **Please Activate Your myQNX Account**
 - click the link to activate your myQNX account ☐

- You'll receive *another* email from QNX with the Subject: **myQNX Account Confirmation**
 - Forward this email to me: pharasb@algonquincollege.com
 - Sometime during your scheduled lab period, I'll action the email that you forwarded to me: I'll add your Algonquin College email to my QNX academic license key. Next, I'll reply to the email that you forwarded to me.
 - Kindly wait for *my* reply email saying: Success... you've been added to my QNX License Key
 - You're not able to proceed with the rest of the lab until you receive my "success" email
- 2. Log into your myQNX account.
- 3. Verify that you have an academic license key.
 - a. myQNX > myQNX License Manager

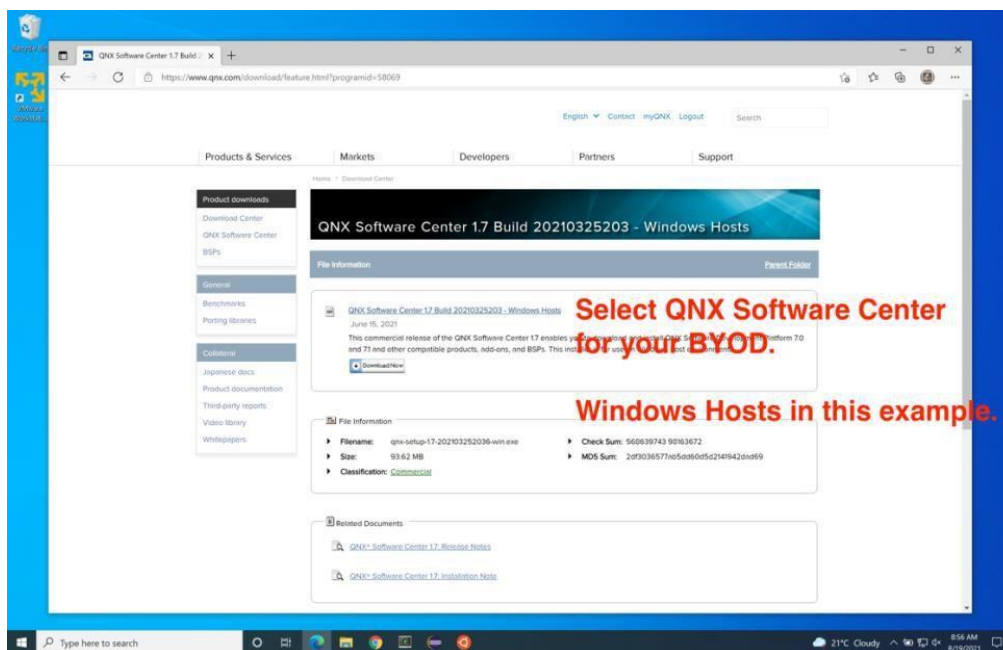
Part B: Install the QNX Software Center

Note: you require an academic license in order to download the QNX Software Center. See Part A.

The QNX Software Center enables you to download and install QNX Software Development Platform (SDP) version 7.1 and other compatible products, add-ons, and BSPs.

Download the QNX Software Center for your host BYOD device.

1. myQNX > QNX Software Center
2. Select your host BYOD: Linux, macOS, or Windows



Download and Launch the QNX Software Center on your host BYOD.

//TODO: take your screenshot for this lab's deliverable. See the Deliverable section below.

Note: there may not be enough time left in the scheduled lab to complete the remaining parts --- Parts C thru F. The biggest issue is the huge amount of time needed to download the QNX SDP v7.1 environment (i.e. high network latency).


As mentioned, this part, Part B, provides the information requested in the deliverable's screenshot. You've completed the deliverable for this lab when you submit the screenshot to Brightspace.

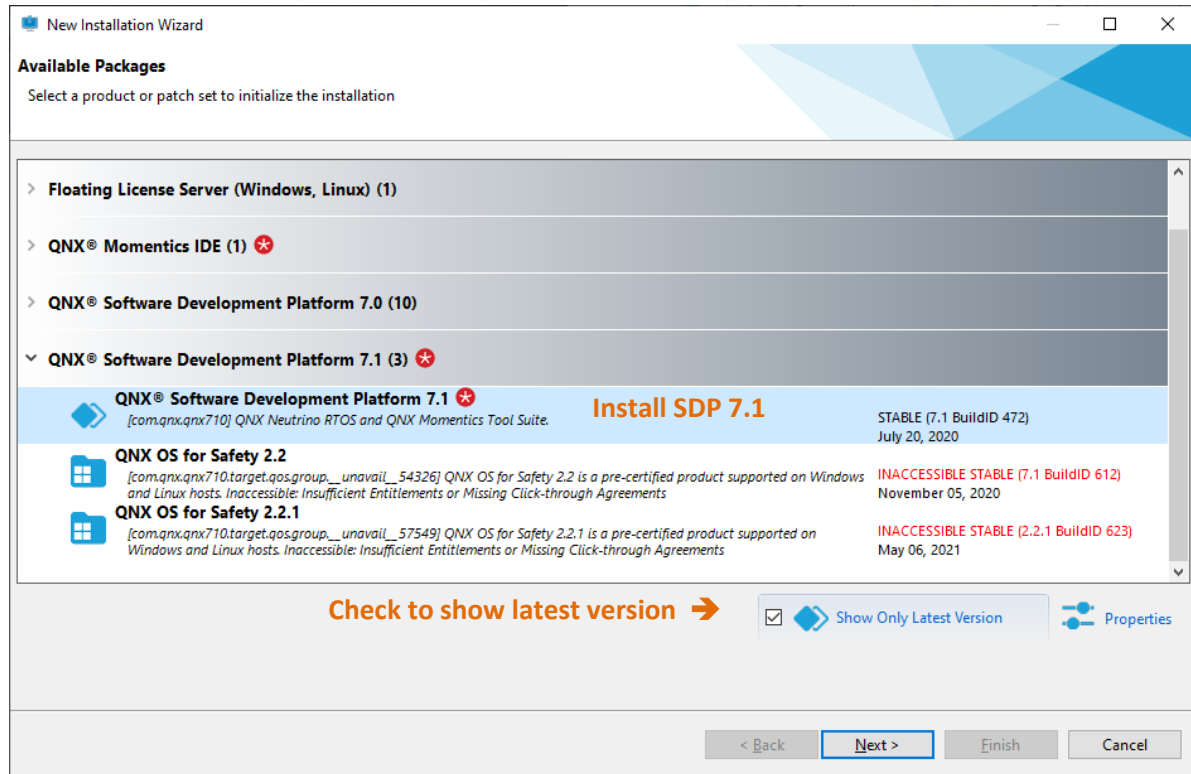
The remaining parts of the lab --- Parts C thru F --- are to be completed before your next scheduled lab period in Week 2 (i.e. by next week's lab).

Part C: Install the QNX Software Development Platform

The QNX Software Development Platform (SDP) includes the QNX Neutrino and the QNX Momentics Tool Suite. This software provides a complete and comprehensive development environment for QNX Neutrino-based devices and systems.

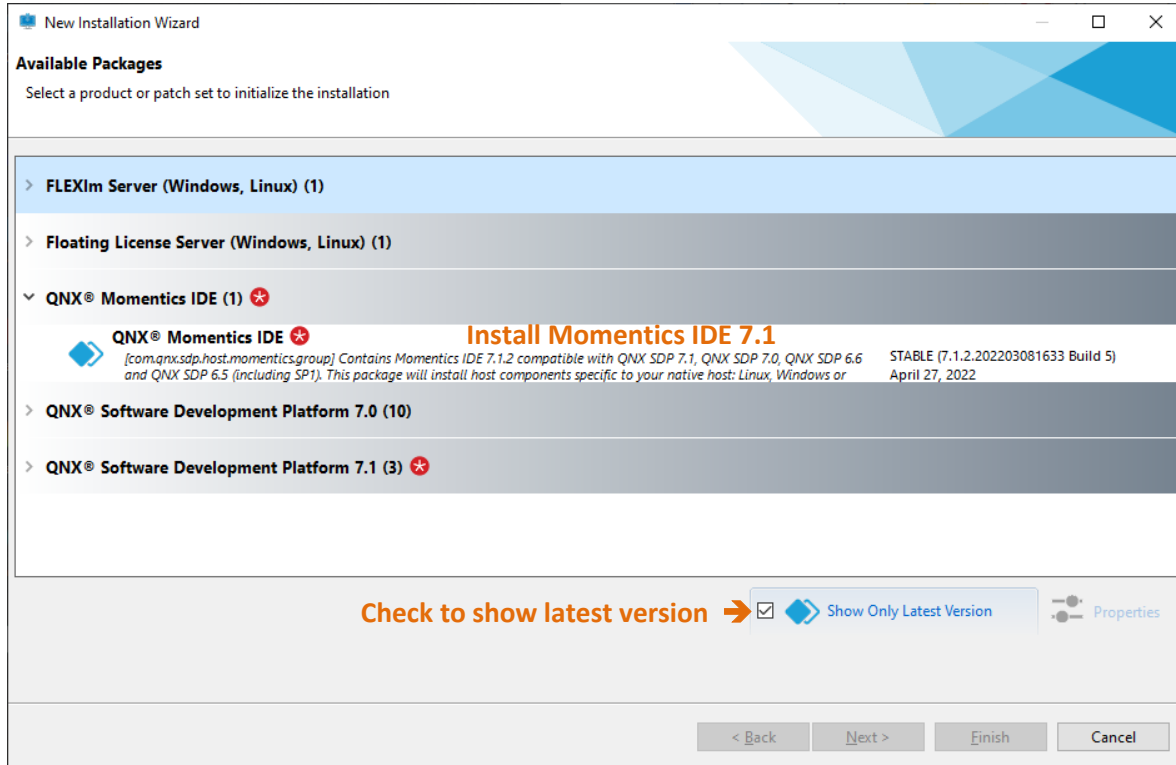
Launch the QNX Software Center and create a new installation of QNX SDP.

1. Add a QNX SDP 7.1 Installation
Home (icon)  Add Installation > QNX Software Development Platform 7.1
 - Check: "Show Only Latest Version"
 - accept all default values during the Install



Please be patient for the software packages to be downloaded and installed.

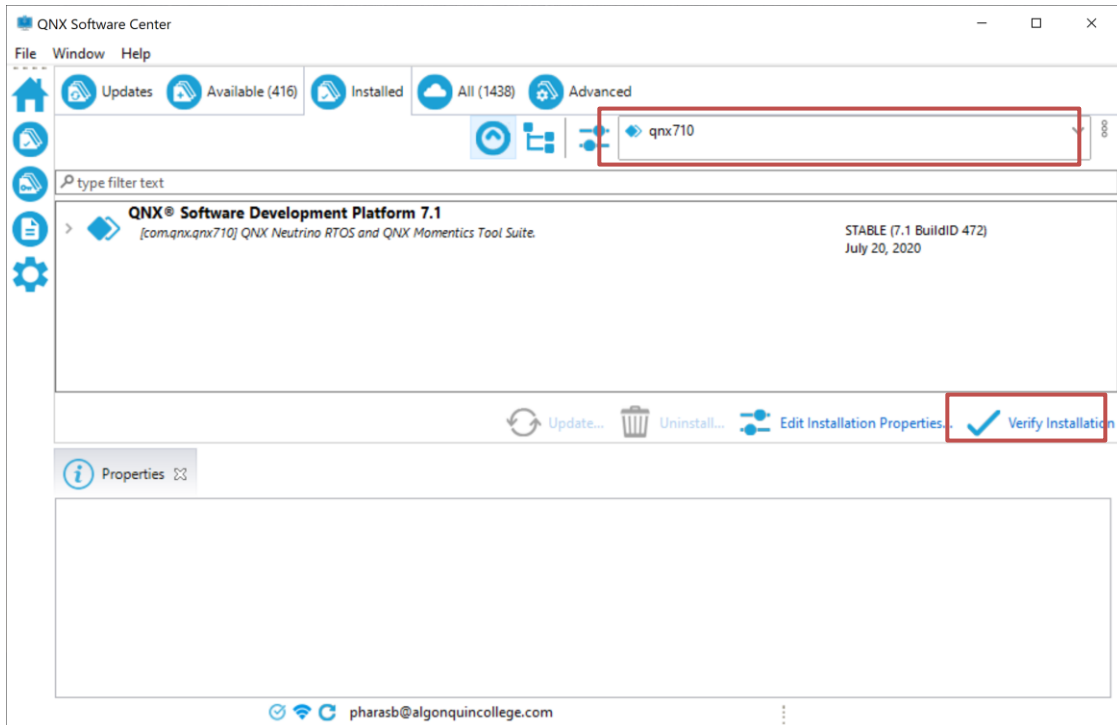
2. Install Momentics IDE 7.1
Home (icon) + Add Installation QNX Momentics IDE



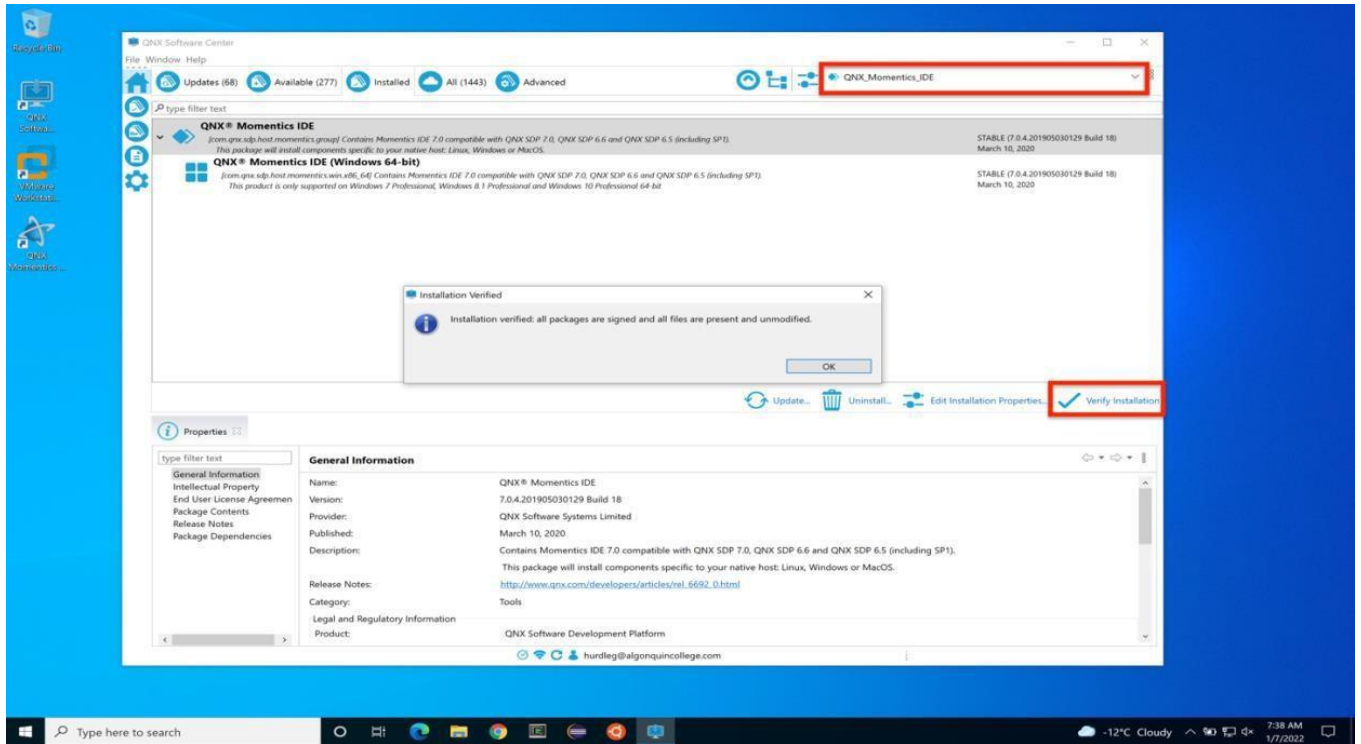
Once installed you will get a popup dialog message “Operation completed Successfully”

3. Verify QNX SDP 7.1 Installation
Home (icon) -> Manage Installation

You should see what I see:



4. Verify Momentics IDE installation
You should see what I see:



5. Creating the virtual machine image of a QNX Neutrino runtime system

You have several choices for the target system that will run the QNX Neutrino RTOS:

- Embedded hardware: You run the QNX Neutrino RTOS on a reference platform, a reference design made by a CPU vendor. You'll need a QNX Board Support Package (BSP) for your platform. Each BSP comes with documentation that explains how to build a QNX Neutrino image and install it on that target system.
- Virtual machine (VM): You can run the QNX Neutrino RTOS as a virtual machine in a VMware session.

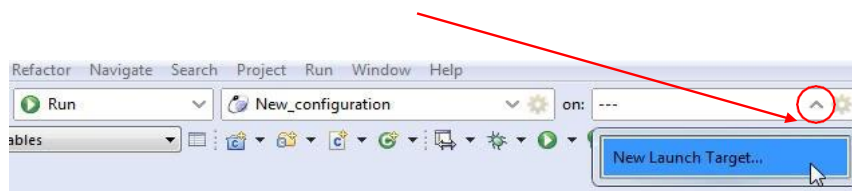
Prior to creating your virtual machine (VM) image using the IDE, you must install the 3rd party application required to host the VM you have chosen above; VMware Workstation (PC) or VMware Fusion (Mac) [refer to page 1]

To create your image, do the following:

- a. Launch the Momentics IDE and create a new Workspace to hold the VM target.

Workspace folder. for example: "C:\Users\john\qnx710\vmimages"

- b. To access your target system from the IDE and run programs on it, you have to create a *target connection*. Click in the **on:** box in the launch bar at the top of the IDE, and select **New Launch Target**.



- c. Select **QNX Virtual Machine Target** in the list of target types. This tells the IDE to set up a target system by generating a VM that runs QNX Neutrino. Then, click Next.
- d. In the New QNX Virtual Machine Target window, configure the settings as needed and click Finish.

New QNX Virtual Machine Target

QNX Virtual Machine Target
Edit the properties of the QNX Virtual Machine

Target Name: Local_VM71

VM Platform: vmware

CPU Architecture: x86_64

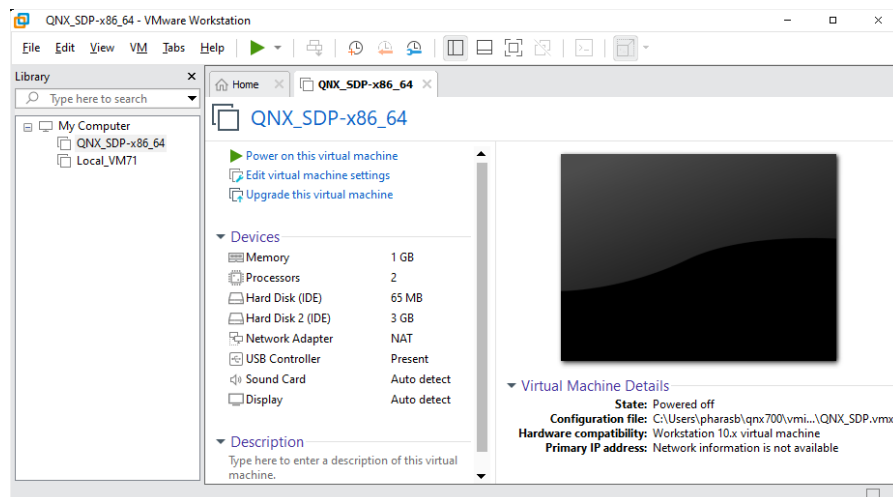
IP Address: <leave blank for automatic>

Extra Options: <leave blank for default options>

Preview:
QNX_TARGET=C:/Users/pharash/qnx710/target/qnx7 C:/Users/pharash/qnx710/host/win64/x86_64/usr/bin/bash C:/Users/pharash/qnx710/host/common/bin/mkqnximage --noprompt --hostname=Local_VM71 --type=vmware --arch=x86_64

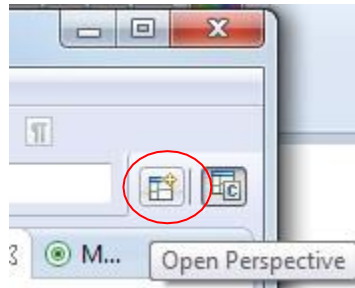
? < Back Next > Finish Cancel

- e. The created virtual machine will automatically be launched and started using VMware as shown below. *[Note: Some options will automatically launch the 3rd party virtualization application and run the newly created VM image for you and connect the target and IDE. Others may require that you launch the supporting application and open the newlycreated VM machine yourself. Refer to Part D]*



f. Testing to make sure that the virtual machine works

- 1.) Once the virtual machine has started,
- 2.) Using the “Open Perspective” icon near the top right of your Momentics IDEwindow, ...



...open the System Information Perspective

- 3.) In the Target Navigator view (top left-hand corner) you should see a connection to the target, and if you click the expansion icon, you'll see a list of the running processes. If something is wrong, you'll see a red 'X' beside the target.

Local_VM71 - Last Updated: Thu, May 12 15:43:06 EDT 2022

System Specifications

Hostname: Local_VM71
Board: x86pc
OS Version: 7.1.0 (2021/05/07-14:19:56EDT)
Boot Date: Thu May 12 15:42:17 EDT 2022

CPU Details

x86_64 @ 1804Mhz
x86_64 @ 1820Mhz

System Memory

Used: 112M Free: 910M Total: 1023M

Total Processes: 22

Process Name	Code	Data	Stack	Data Usage Delta	CPU Usage	CPU Usage Delta	Start Time
procnto-smp-instr...	810K	153K	0	0	1m 35s 656ms	9s 985ms	Thu May 12 15:42:17 E
slogger2 (16387)	68K	196K	12K	0	5ms	0	Thu May 12 15:42:17 E
pci-server (28676)	144K	228K	28K	0	50ms	0	Thu May 12 15:42:17 E
devb-eide (40965)	128K	3292K	144K	0	272ms	52us 254ns	Thu May 12 15:42:17 E
devc-con (114694)	92K	148K	8192	0	2ms	0	Thu May 12 15:42:18 E
syslogd (118791)	28K	140K	40K	0	2ms	26us 163ns	Thu May 12 15:42:18 E
random (122888)	36K	396K	32K	0	4ms	19us 852ns	Thu May 12 15:42:18 E
pipe (135177)	28K	148K	56K	0	3ms	0	Thu May 12 15:42:18 E
devc-pty (147466)	68K	176K	8192	0	888us 954ns	0	Thu May 12 15:42:18 E
dumper (151563)	132K	180K	8192	0	1ms	0	Thu May 12 15:42:18 E
io-pkt-v6-hc (196...	1588K	752K	136K	0	476ms	6ms	Thu May 12 15:42:18 E
ksh (159757)	296K	160K	8192	0	1ms	0	Thu May 12 15:42:18 E
ksh (167950)	296K	160K	8192	0	2ms	0	Thu May 12 15:42:18 E
ksh (176143)	296K	160K	8192	0	2ms	0	Thu May 12 15:42:18 E
ksh (184336)	296K	160K	8192	0	2ms	0	Thu May 12 15:42:18 F

[disconnected] 78M of 303M

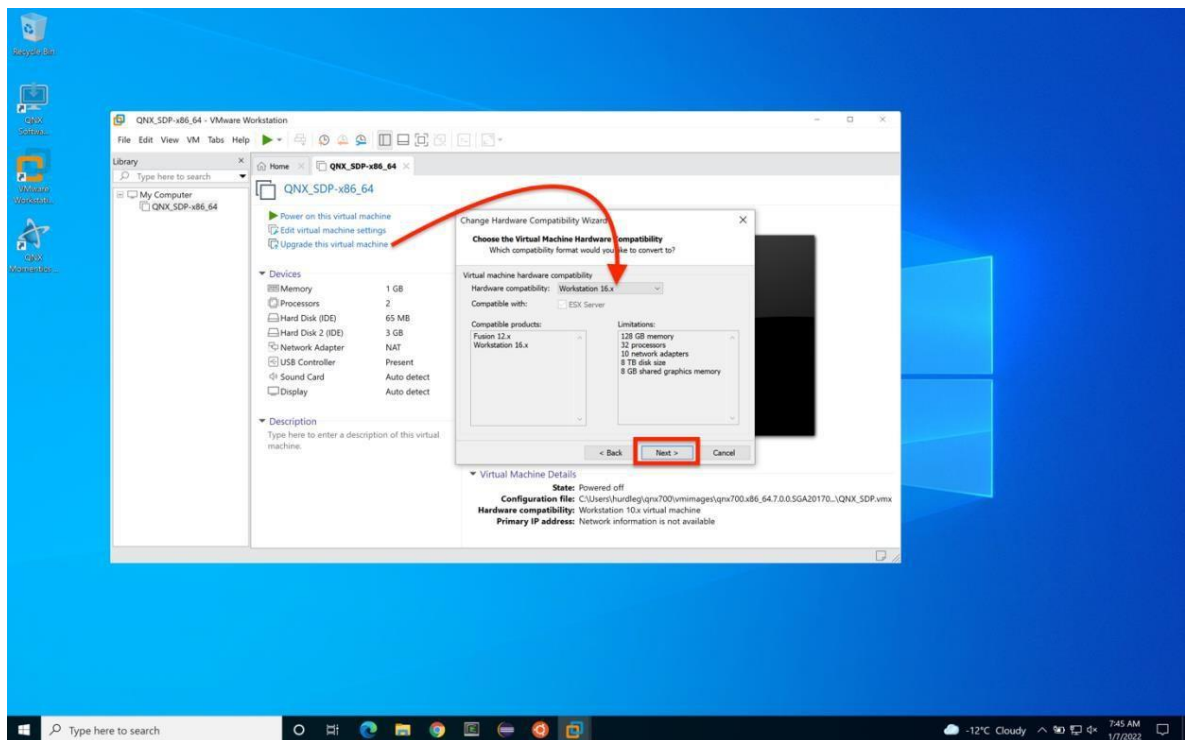
Part D: Go to Part E, if your virtual machine is automatically started otherwise follow the steps below;

Launch the QNX Neutrino RTOS VM created in step 5

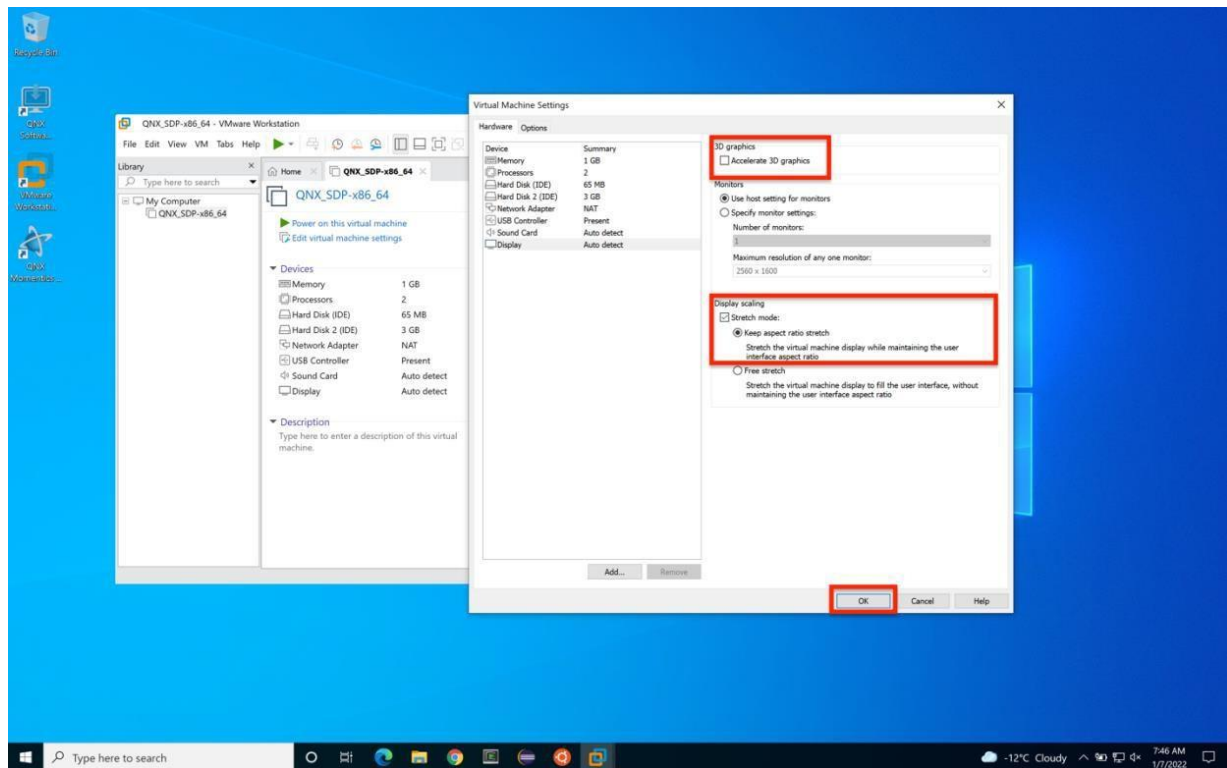
When the installation of the VM image is complete, start VMware, choose **File > Open...** and navigate to *the folder* where you created the virtual machine

Workspace folder. for example: "C:\Users\john\qnx710\ ... \local\vmware_files\vmware.vmx"

Upgrade the virtual machine to **Workstation 16.x**
Apply the changes to the existing VM.



Next, edit the virtual machine settings -> Display
(optional: remove the Sound Card)



After you start the virtual machine, you're automatically logged in as root. To see a list of the processes that currently exist in your system, type the following on the QNX Neutrino target's console:

pidin | less

```

non UEFI or UEFI+CSM boot
No low memory for AP startup; using bootloader
apstart_init(00000000000000600,0000000000008174c3) [size=0x000000b6]
MMFLAGS=1

Welcome to QNX Neutrino 7.0.0 on VMware on x86_64!

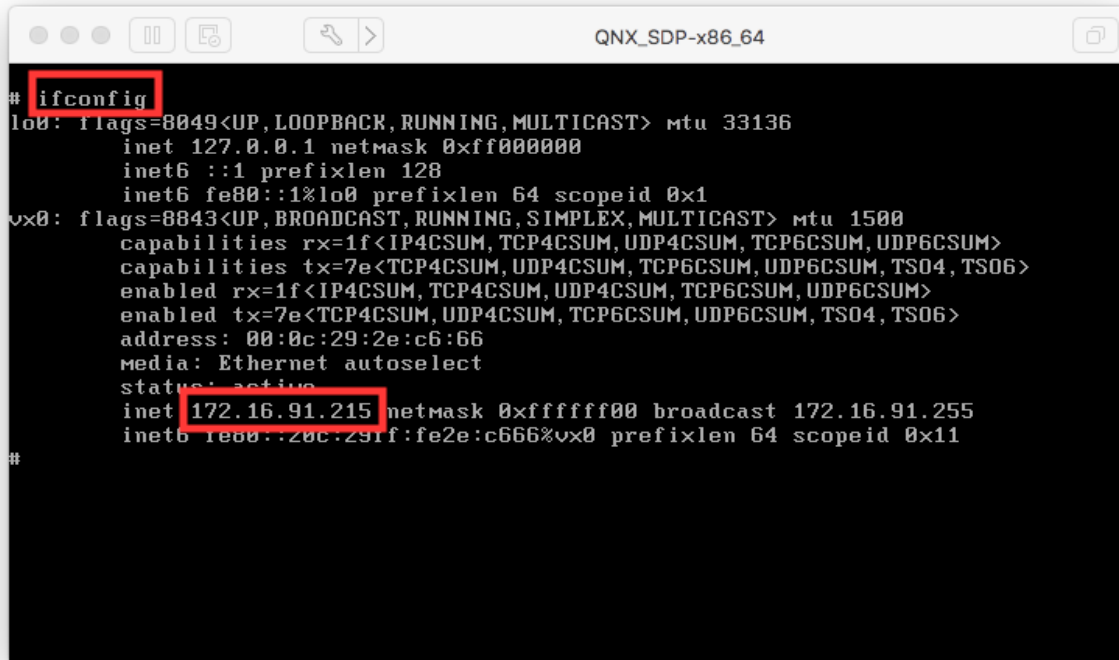
---> Starting PCI Services
---> Starting EIDE Driver
P---> Mounting Filesystems
ath=0 - Intel 82371AB
target=0 lun=0      Direct-Access(0) -      VMware Virtual I Rev: 0000
target=1 lun=0      Direct-Access(0) -      VMware Virtual I Rev: 0000
---> Starting HID Server
---> Starting console
---> Running startup script
---> Starting serial
---> Starting USB
---> Mounting USB HID driver
---> Starting Networking
---> Starting Debug Services
---> Starting audio
---> To start graphics (screen) run /etc/graphics-startup.sh
---> Starting shell
# pidin | less_
  
```

Each process is optional (except **procnto-smp-instr**, which is the microkernel), which means that later in your design, you can remove processes to save resources --- or you can add other processes to increase the system's functionality. This also applies to graphics, networking, or audio; each QNX Neutrino component is a single process that you can load dynamically.

One of the running programs you should see in the list is **qconn** (**pidin | grep qconn**). It is used by the IDE (i.e. QNX Momentics) to push binaries to the QNX target. Type **q** to exit the **less** command.

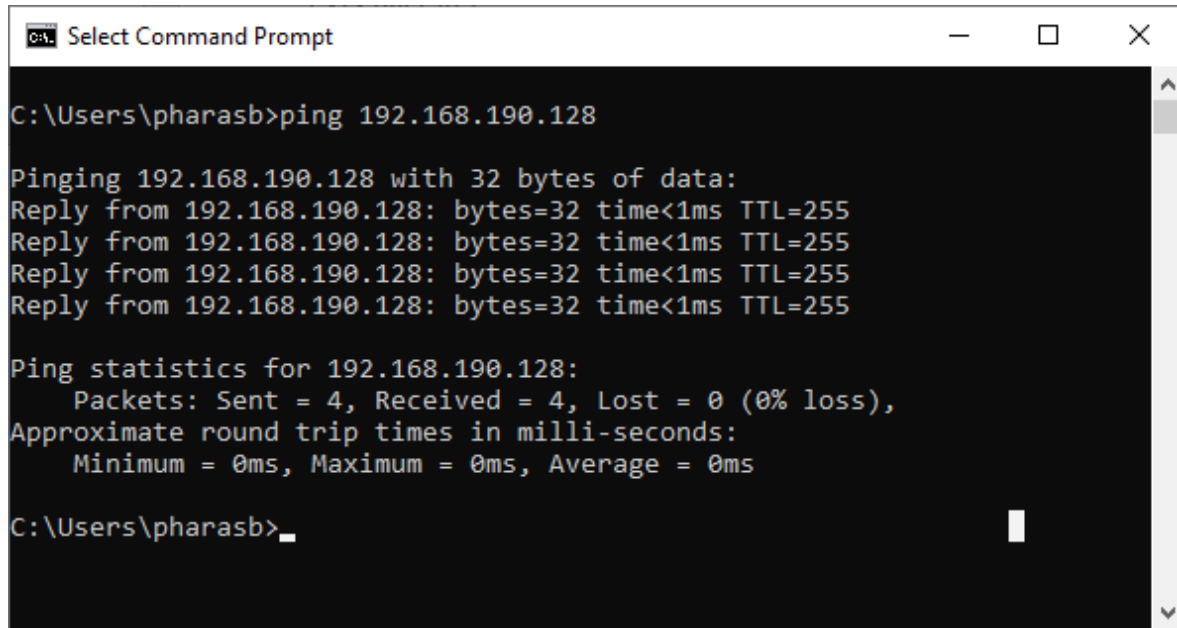
Part E: Networking with the QNX Neutrino RTOS

QNX has set up the virtual machine to use **Network Address Translation (NAT)** so that it's on the same IP network as your development host. To determine the target system's IP address, you can use the **ifconfig** command on the target's console:



```
# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33136
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
vx0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    capabilities rx=1f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM>
    capabilities tx=7e<TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM,TS04,TS06>
    enabled rx=1f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM>
    enabled tx=7e<TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM,TS04,TS06>
    address: 00:0c:29:2e:c6:66
    media: Ethernet autoselect
    status: active
    inet 172.16.91.215 netmask 0xfffff00 broadcast 172.16.91.255
    inet6 fe80::20c:291f:fe2e:c666%vx0 prefixlen 64 scopeid 0x11
#
```

On your development host, use **ping IP_address** to check that it can reach your QNX Neutrino target on the network:



```
C:\Users\pharasb>ping 192.168.190.128

Pinging 192.168.190.128 with 32 bytes of data:
Reply from 192.168.190.128: bytes=32 time<1ms TTL=255
Reply from 192.168.190.128: bytes=32 time<1ms TTL=255
Reply from 192.168.190.128: bytes=32 time<1ms TTL=255
Reply from 192.168.190.128: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.190.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\pharasb>
```

Troubleshooting:

Can't ping your running instance of QNX Neutrino? Configure VMware Workstation (Fusion on MacOS) to use NAT, instead of 'bridge' mode.

Problem during the QNX software center installation? Regarding your operating system, is there a **space** in your username? For example C:/Users/Bhuwan Pharasi

Then do the following steps:

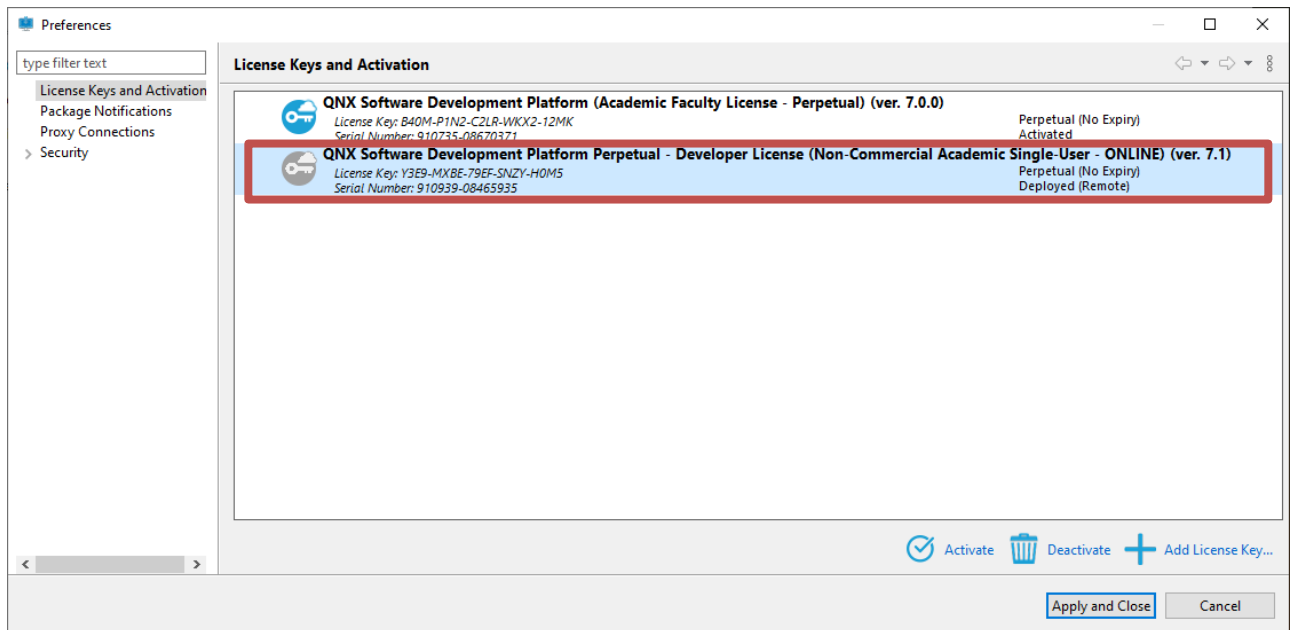
1. Edit: \$HOME\QNX Software Center\configuration\config.ini
2. Change **osgi.instance.area.default** to a new path with no spaces, such as:
C:/QNX/install/.qnx/swupdate
3. Copy the old /swupdate/ folder contents to the new path
4. Repeat steps 2-3 with \$HOME\QNX Momentics\configuration\config.ini
Suggestion for the new path: C:/QNX/install/ide-7.0-workspace

Deliverable:

Upload a screenshot of your Manage License Keys tab from within the QNX Software Center application running on your BYOD device.

1. Launch: **QNX Software Center**
2. From the Home tab, click: **Preferences** (gear wheel) icon
3. Take a screenshot of your BYOD's Desktop.

Upload and submit your screenshot to Brightspace before the due date. Reference Screenshot: [Note: you may have only one entry]



Post-lab:

You rarely need to reboot a Neutrino system. If a driver or other system process crashes, you can usually restart that one process.

To shut down or reboot the system in text mode, use the **shutdown** command. You can do this only if you're logged in as root. This utility has several options that let you:

- **-n *nodename*** :: name the node to shutdown (default is the current node)
- **-f** :: shutdown quickly
- **-v** :: list the actions taken while shutting down (i.e. be verbose)

By default, the shutdown command will reboot. Use the **-b** option to shut down without rebooting:

shutdown -b

Next, shut down VMware:

Virtual Machine > Shut Down

Finally, quit VMware.