

Setup Your QNX Development Host

Author: Gerald Hurdle, Real-time Programming Professor at Algonquin College

Lab Objectives:

1. To create a myQNX account
2. To install the QNX Software Center for your host environment
3. To install the QNX **S**oftware **D**evelopment **P**latform (SDP)

Before You Begin:

You will need the following version of VMware Workstation (PC) or VMware Fusion (Mac) already installed

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.qnxsdp.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 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 ☺

2. Log into your myQNX account.
3. Verify that you have an academic license key.
 - a. myQNX > myQNX License Manager

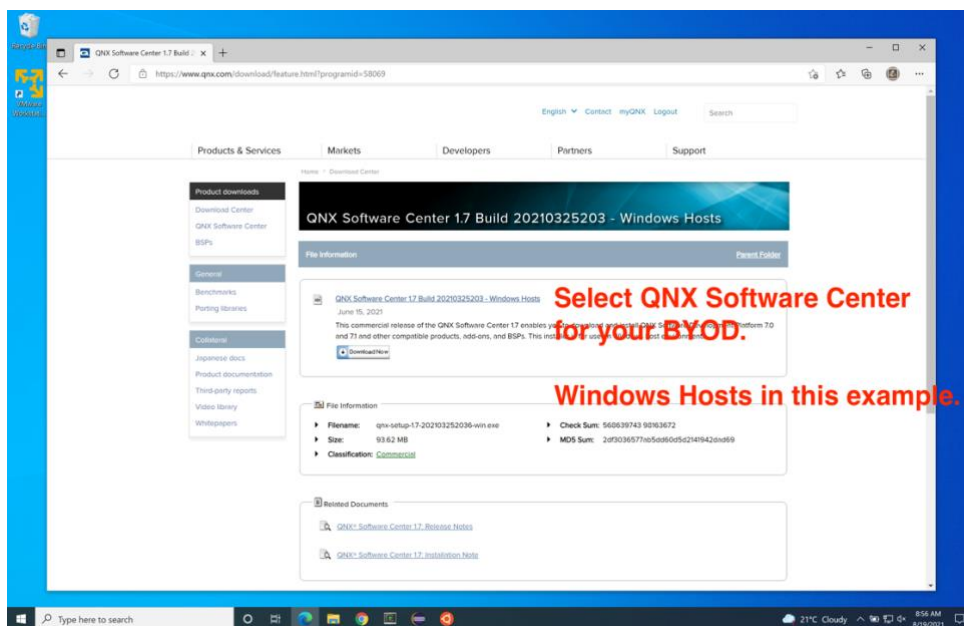
Part B: Install the QNX Software Center

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 7.0 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



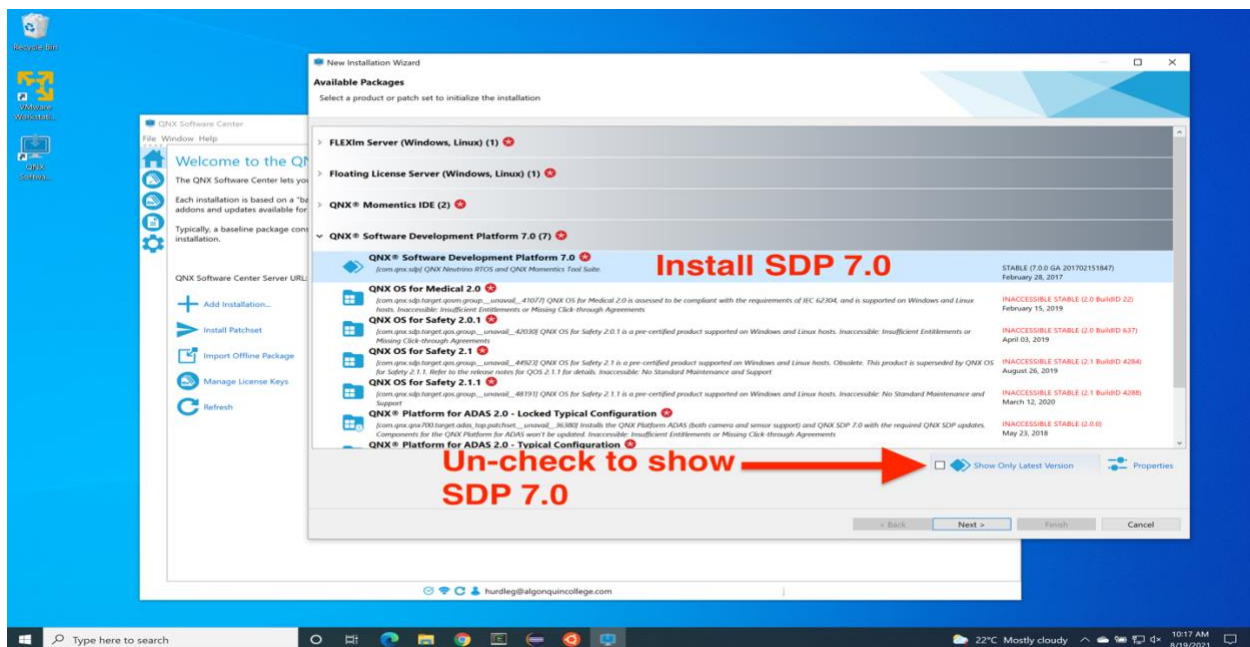
Download and launch the QNX Software Center application on your host BYOD.

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 7.0 Installation
Home (icon) → + Add Installation > QNX Software Development Platform 7.0
 - Uncheck: "Show Only Latest Version"
 - accept all default values during the Install



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

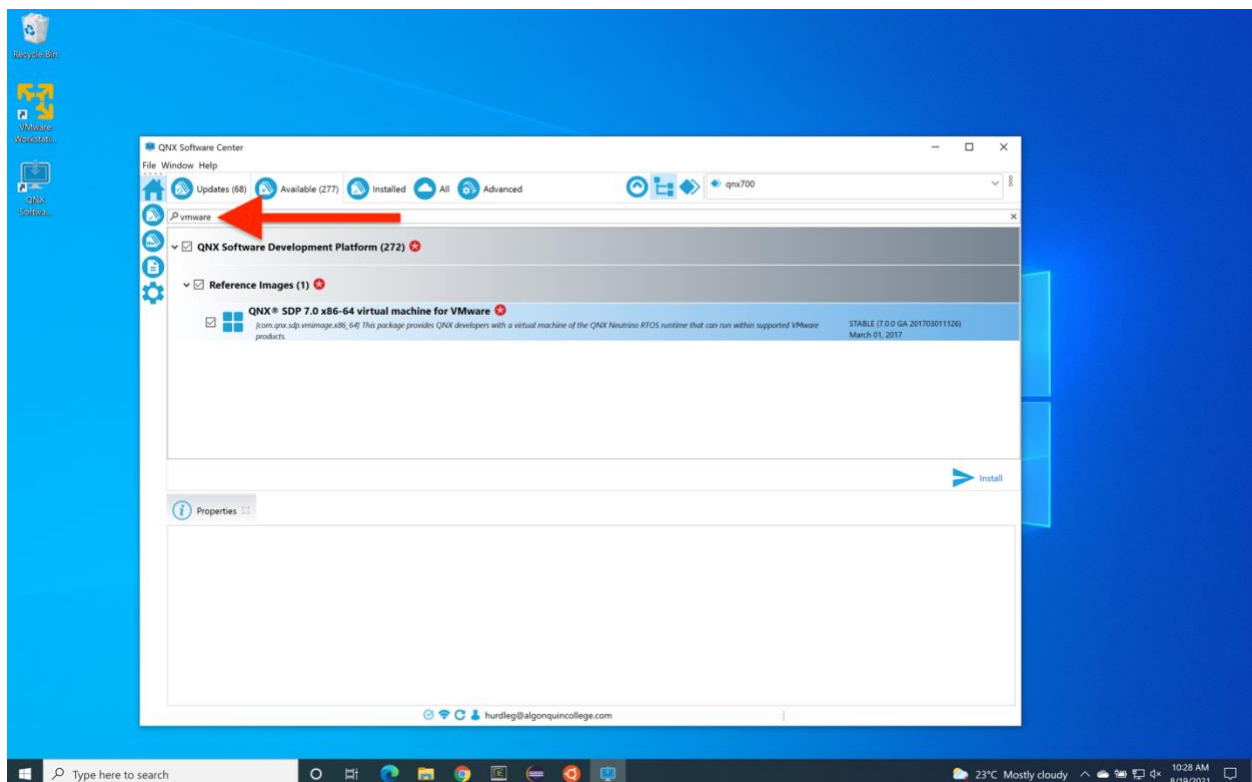
2. Install x86-64 virtual machine for VMware

Home (icon) → Manage Installation → Available (tab) → Search: vmware

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.

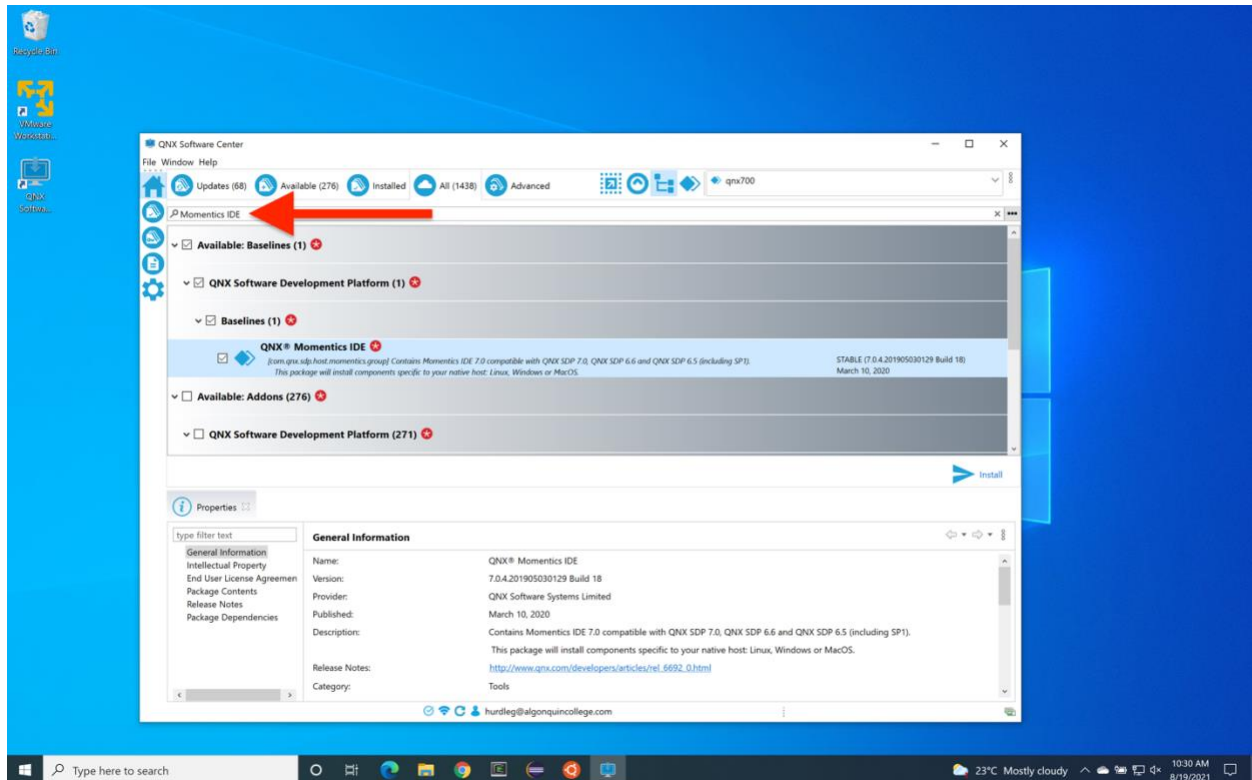
QNX provides a VMware image that's compatible with VMware Workstation Pro 12.0 or later, VMware Workstation Player 12.0 or later, and VMware Fusion Pro 8.0 or later. This image is a minimal QNX Neutrino system.



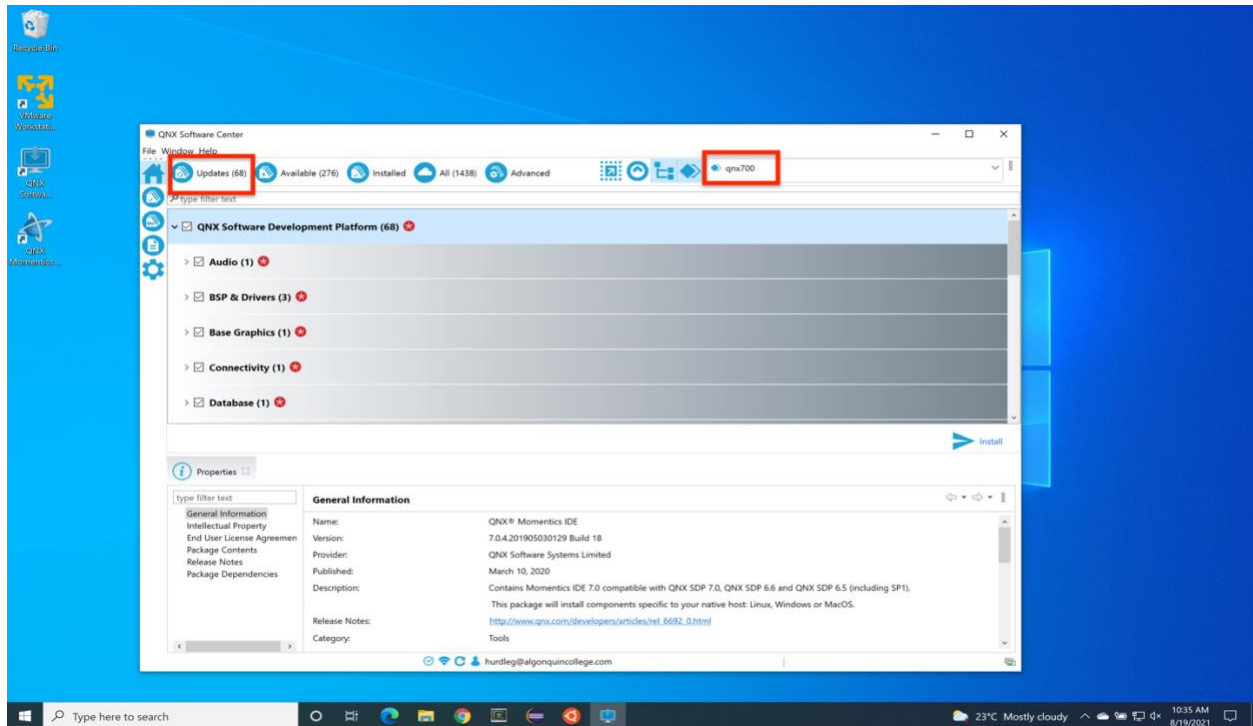
3. Install Momentics IDE 7.0

Home (icon) → Manage Installation → All (tab) → Search: Momentics IDE

Note: once installed, a new installation will appear: QNX_Momentics_IDE (Momentics.app on Mac)

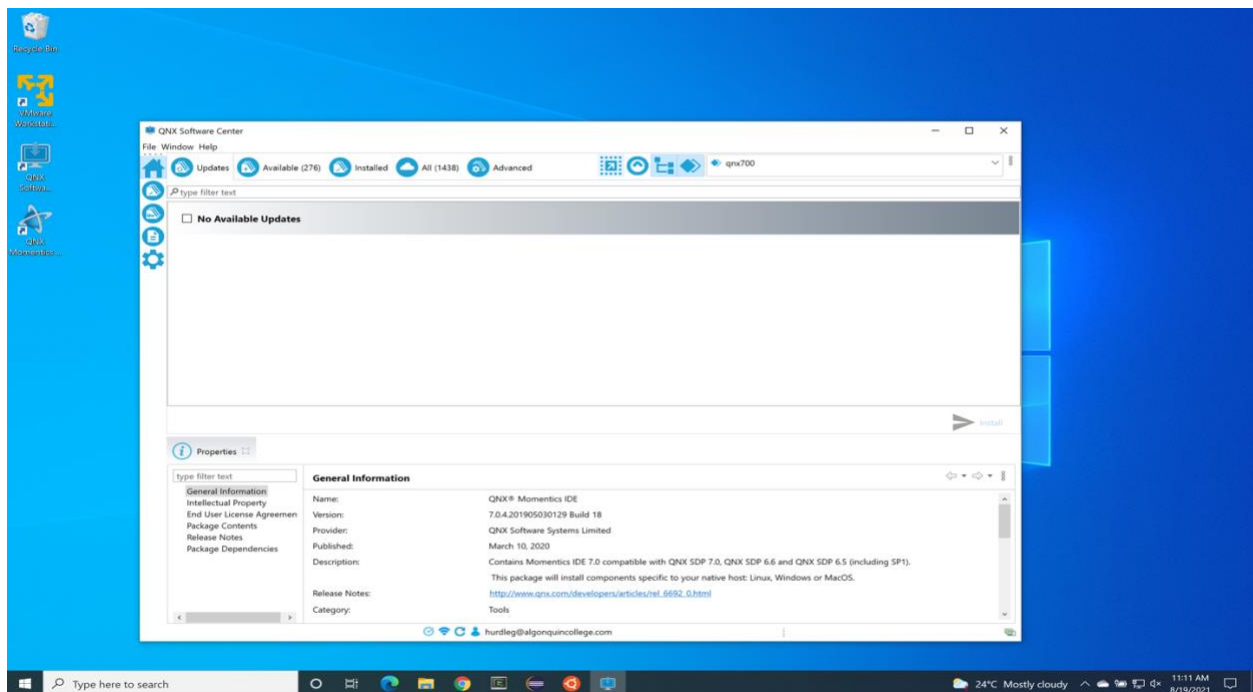


4. Update the software packages: SDP 7.0
Home (icon) → Check for Package Updates > Select All to Install

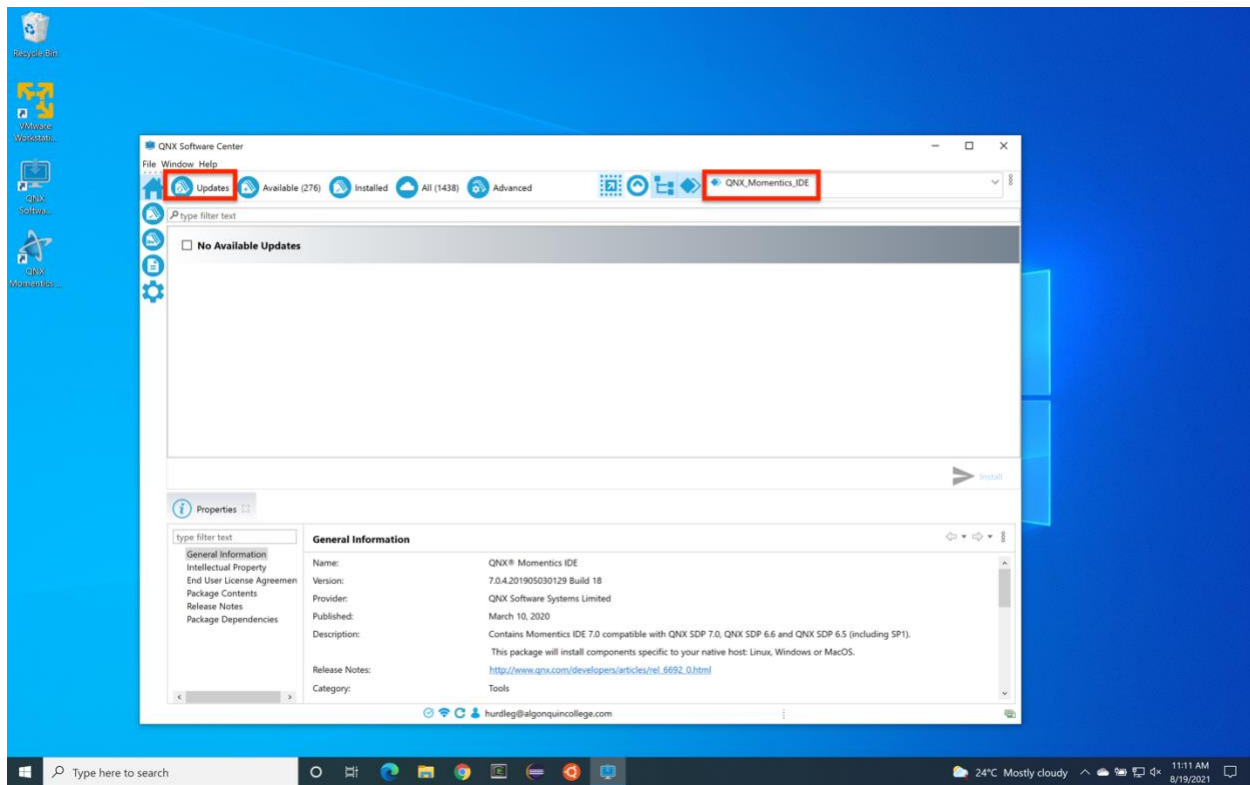


Please be patient for the software packages to be downloaded and installed.

Repeat the update process until there are no more updates.

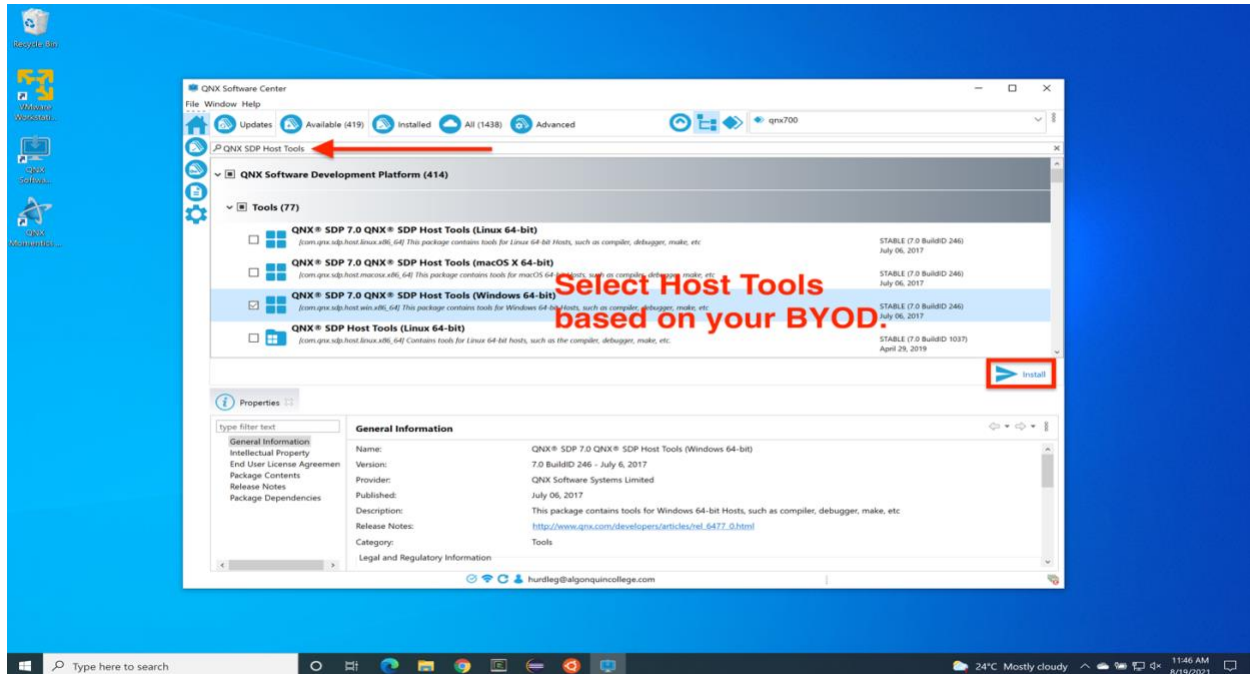


5. Update the software packages: Momentics IDE
Home (icon) → Check for Package Updates > Select All to Install

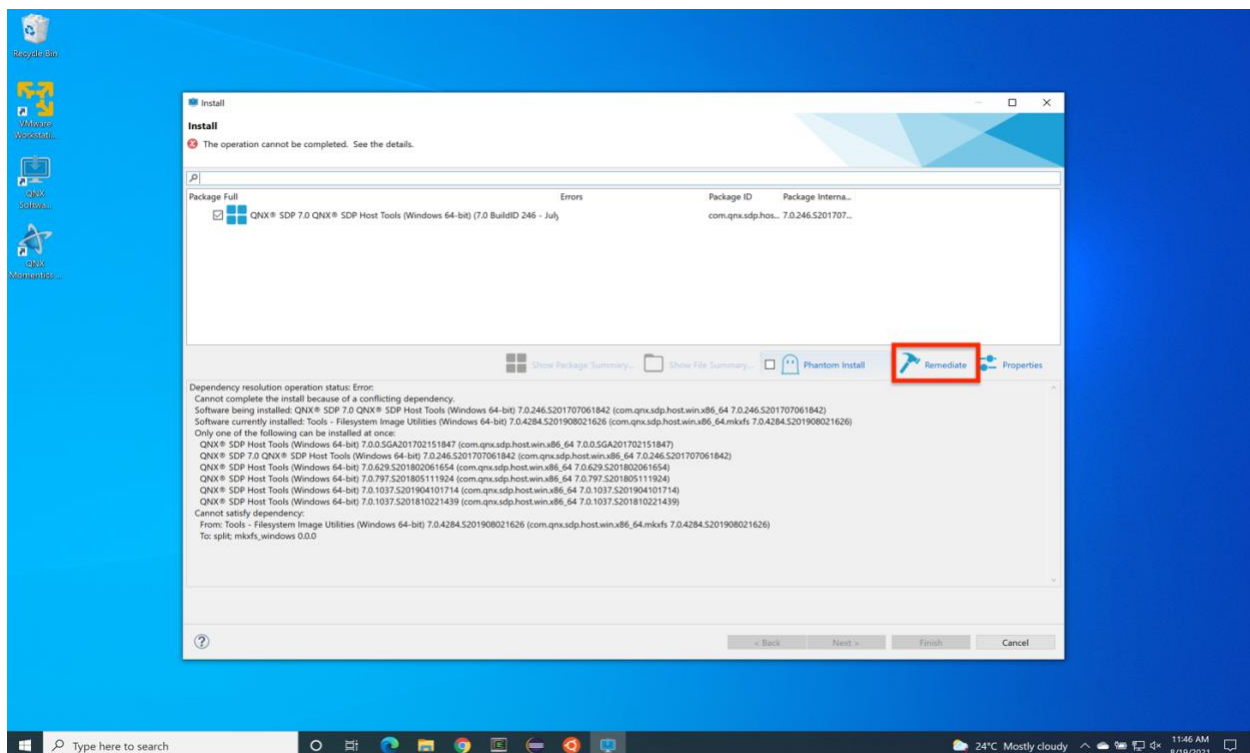


- Install: QNX SDP Host Tools for your BYOD
Home (icon) → Manage Installation → Available (tab) → Search: QNX SDP Host Tools

Select the appropriate *Host Tools* package for your BYOD and click **Install**

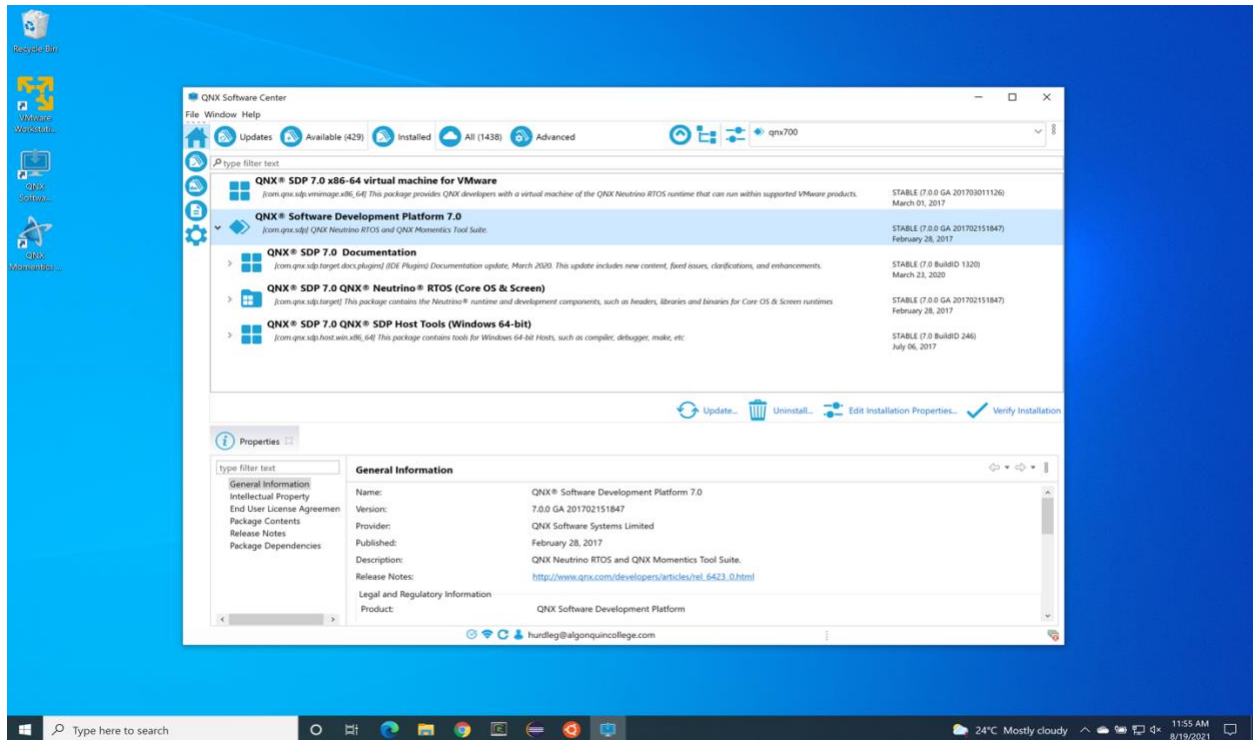


Click **Remediate** to complete the installation, and click Next, Next, etc. until Finish.



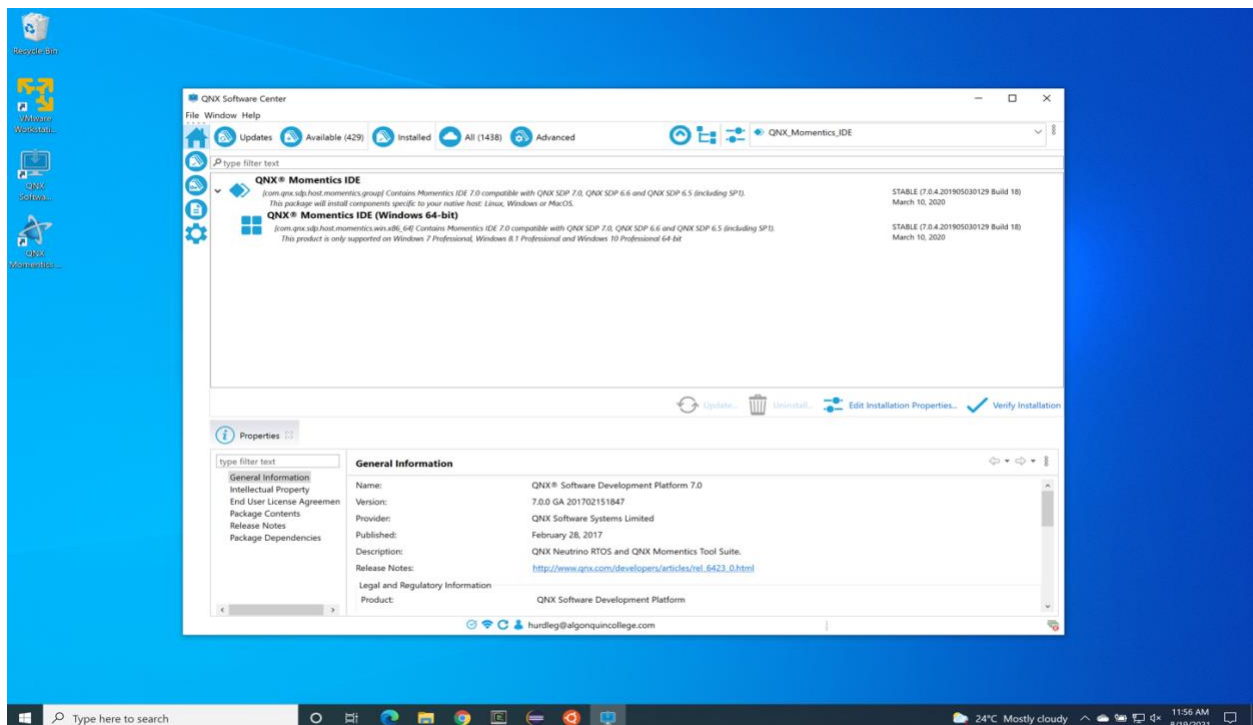
7. Verify QNX SDP 7.0 Installation

You should see what I see:



8. Verify Momentics IDE Installation

You should see what I see:



Part D: Launch the QNX Neutrino RTOS VM

When the installation of the VM image is complete, start VMware, choose **File > Open...** and navigate to `vm_base_directory/vmimages/qnx700.architecture.build_ID/architecture/QNX_SDP.vmx`, where `vm_base_directory` is where you installed the addon.

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:

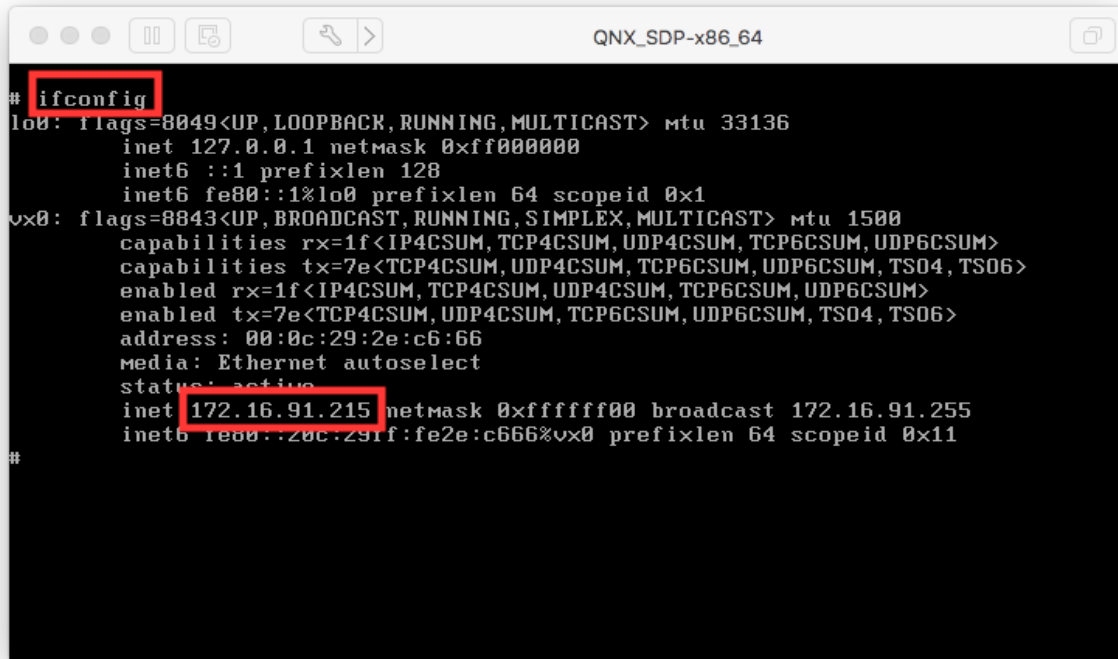
```
pidin | less
```

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 for graphics, networking or audio; each QNX Neutrino component is a single process that you can load dynamically.

One 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.

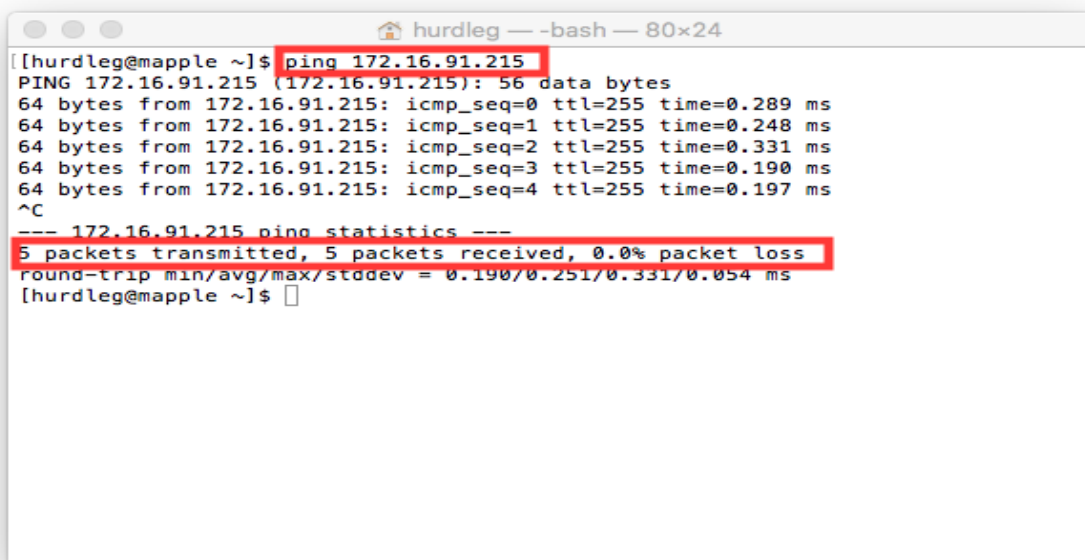
Part E: Networking with the QNX Neutrino RTOS

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:29ff: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:



```
hurdleg - -bash- 80x24
[hurdleg@mapple ~]$ ping 172.16.91.215
PING 172.16.91.215 (172.16.91.215): 56 data bytes
64 bytes from 172.16.91.215: icmp_seq=0 ttl=255 time=0.289 ms
64 bytes from 172.16.91.215: icmp_seq=1 ttl=255 time=0.248 ms
64 bytes from 172.16.91.215: icmp_seq=2 ttl=255 time=0.331 ms
64 bytes from 172.16.91.215: icmp_seq=3 ttl=255 time=0.190 ms
64 bytes from 172.16.91.215: icmp_seq=4 ttl=255 time=0.197 ms
^C
--- 172.16.91.215 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.190/0.251/0.331/0.054 ms
[hurdleg@mapple ~]$
```

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/Gerald C Hurdle

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

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.