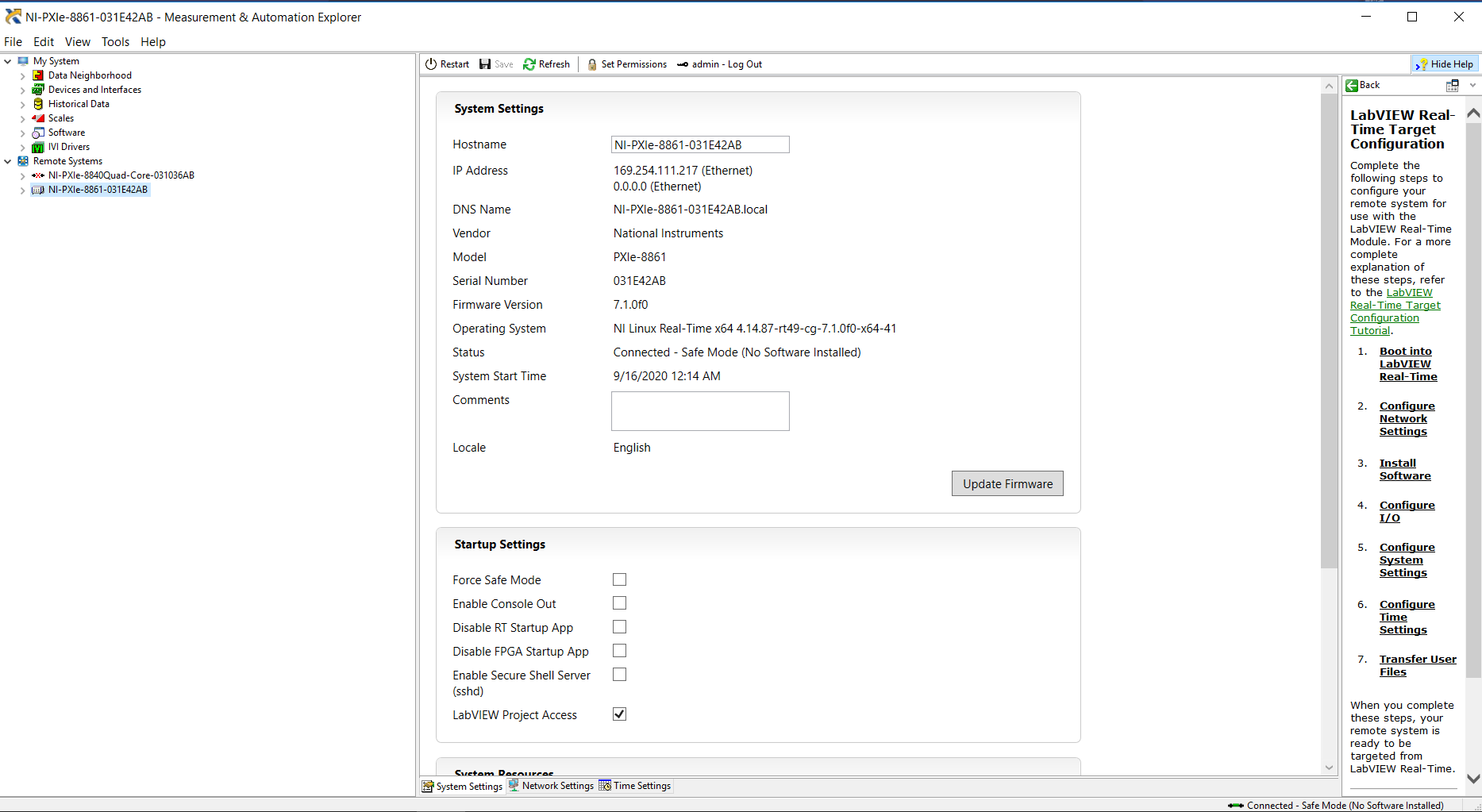
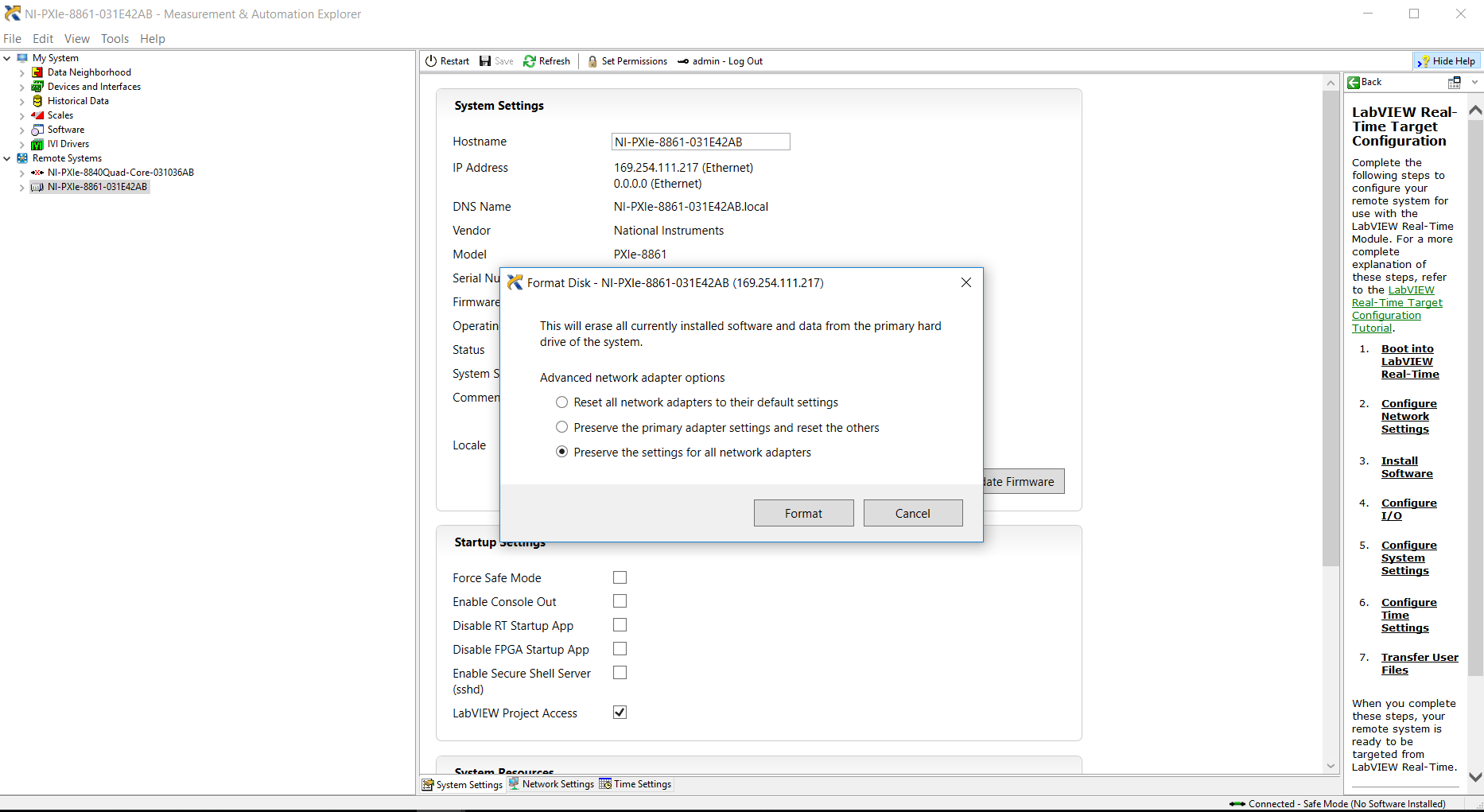
Provisioning Real Time Controller - Instructions

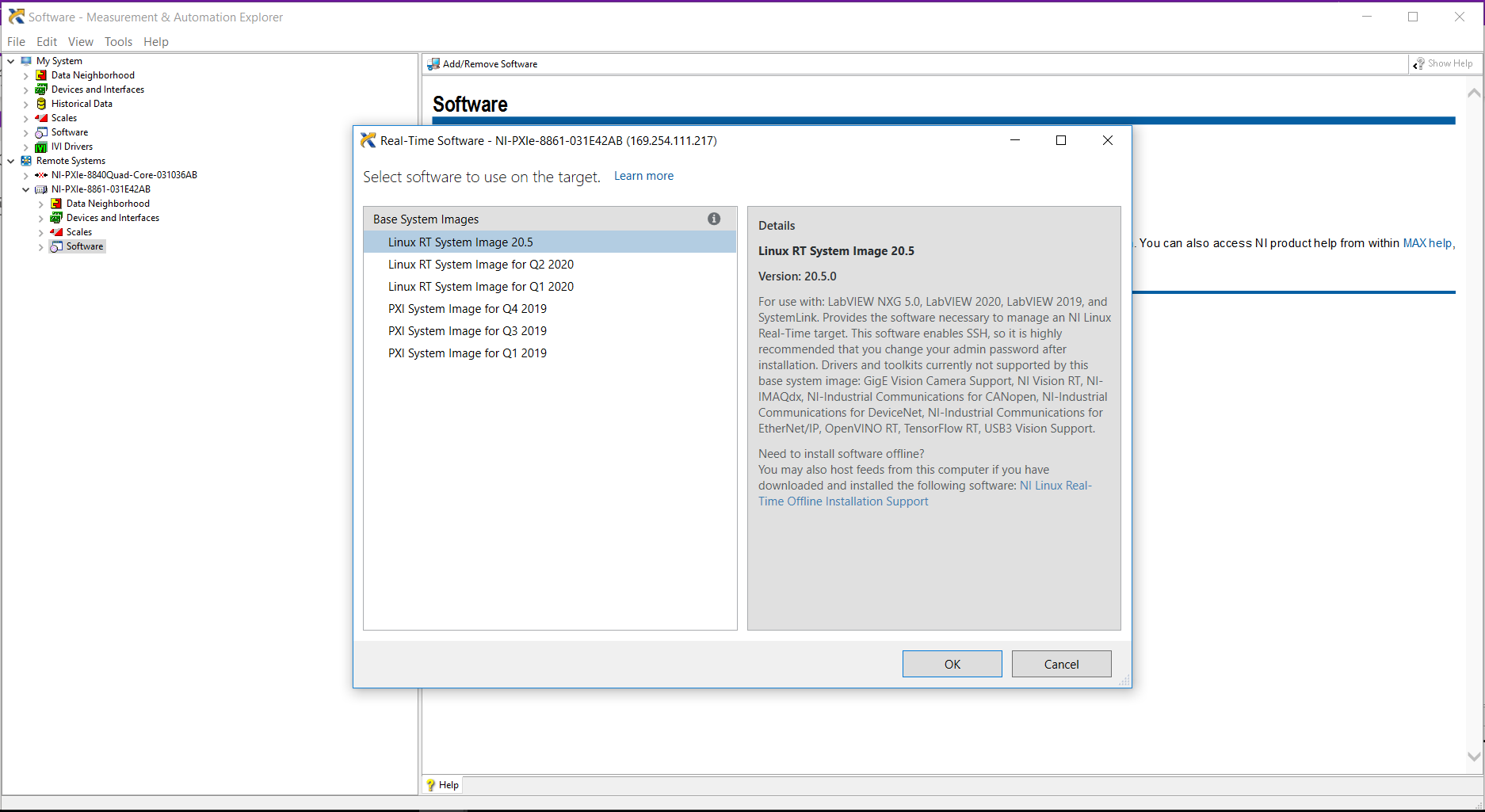
* 1. On dev machine or virtual machine, install NI Package Manager, LabVIEW 2019 and drivers (including PXI Platform Services, DC Power, DMM, and Switch)
  2. Install NI Linux RT System Image <https://www.ni.com/en-us/support/downloads/software-products/download.ni-linux-rt-system-image.html#354219> so the base image can be later deployed to the RT controller from MAX
  3. Connect to the RT controller, open MAX and find the controller under Remote Systems
     1. Dev machine or virtual machine needs to be on the same subnet as the RT controller

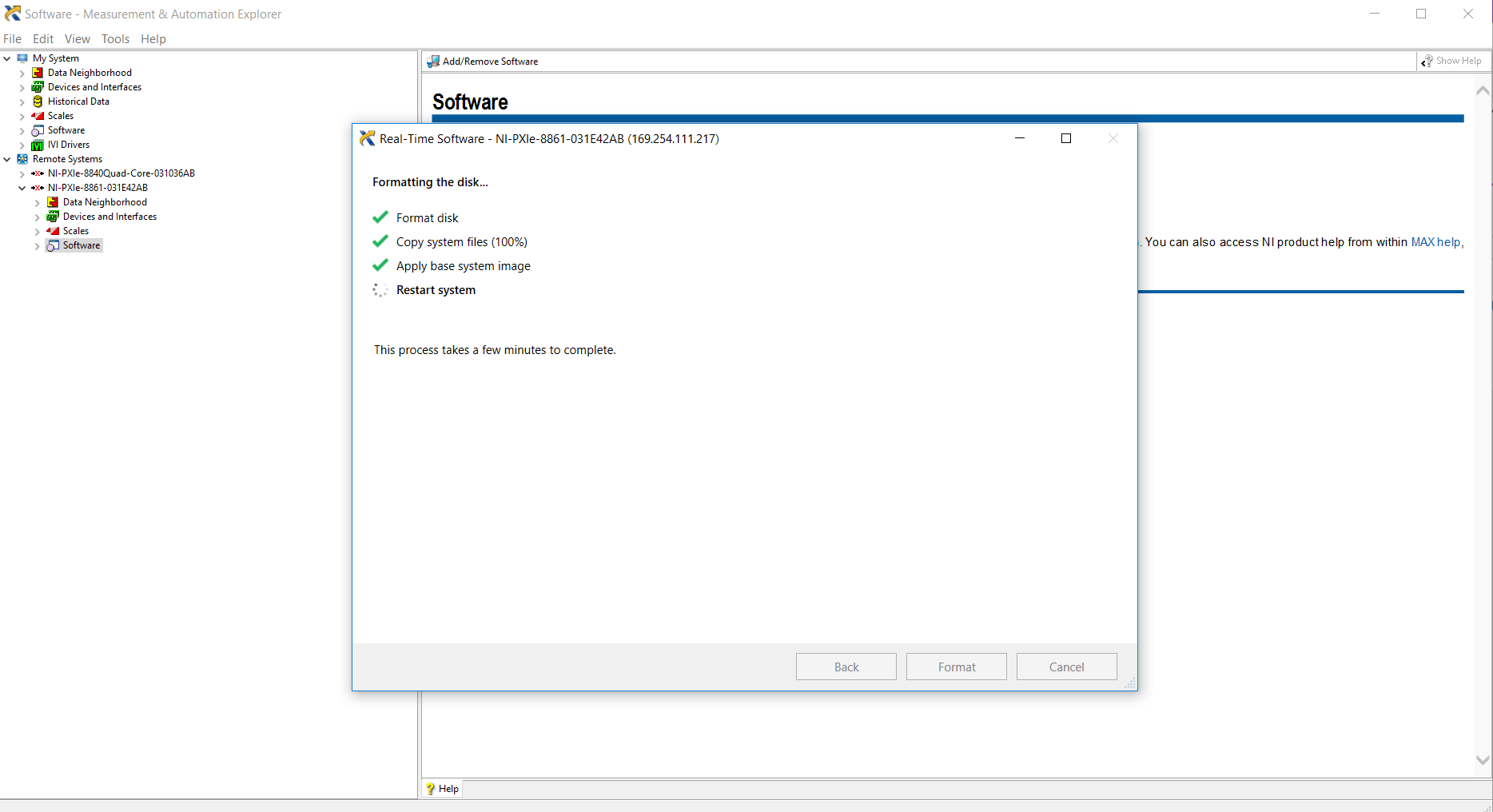


* 1. Format disk
     1. Select to Preserve Network Settings

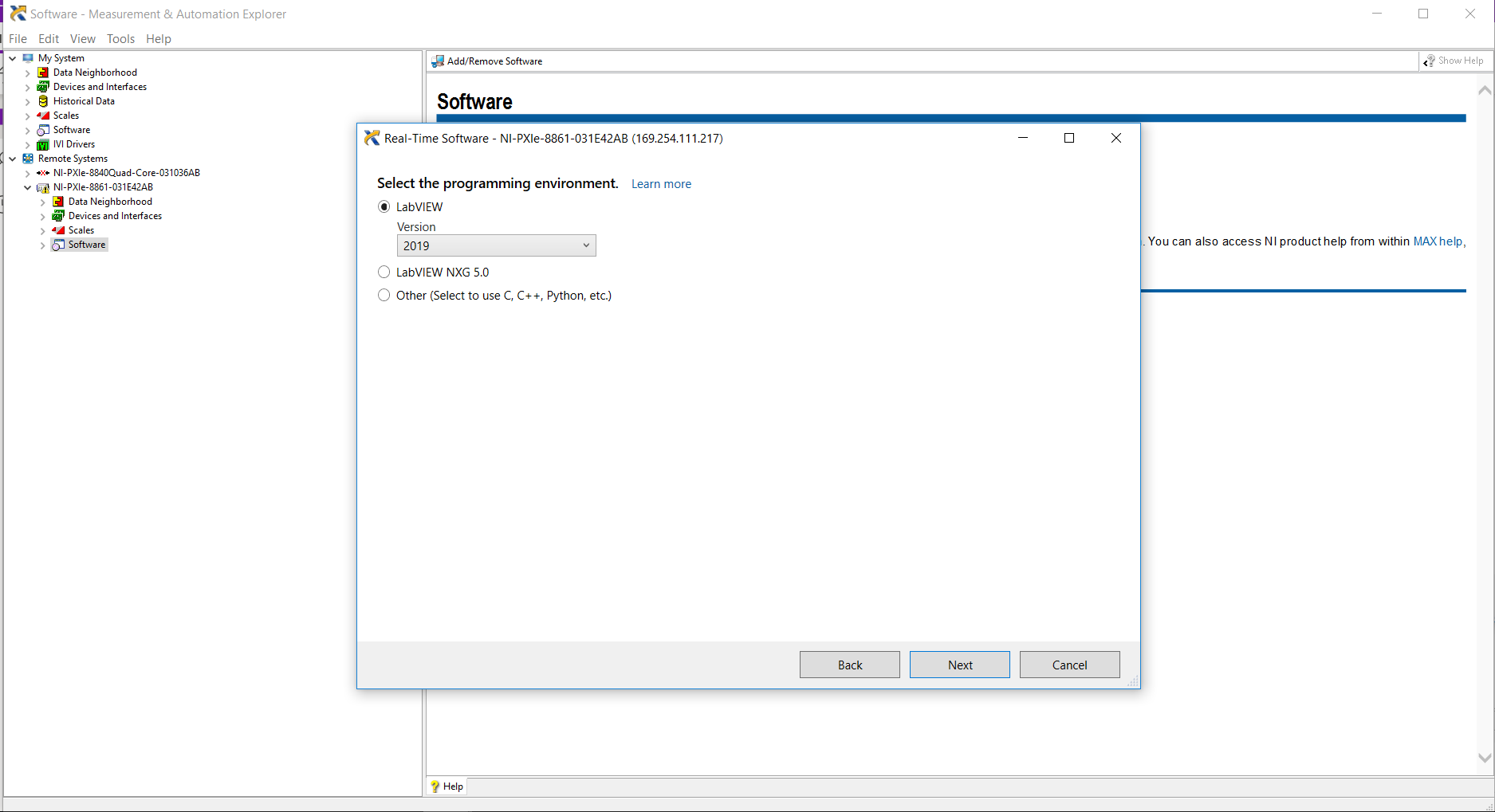


* 1. Right-click on Software and select Add/Remove Software.
  2. Select the Linux RT System Image 20.5

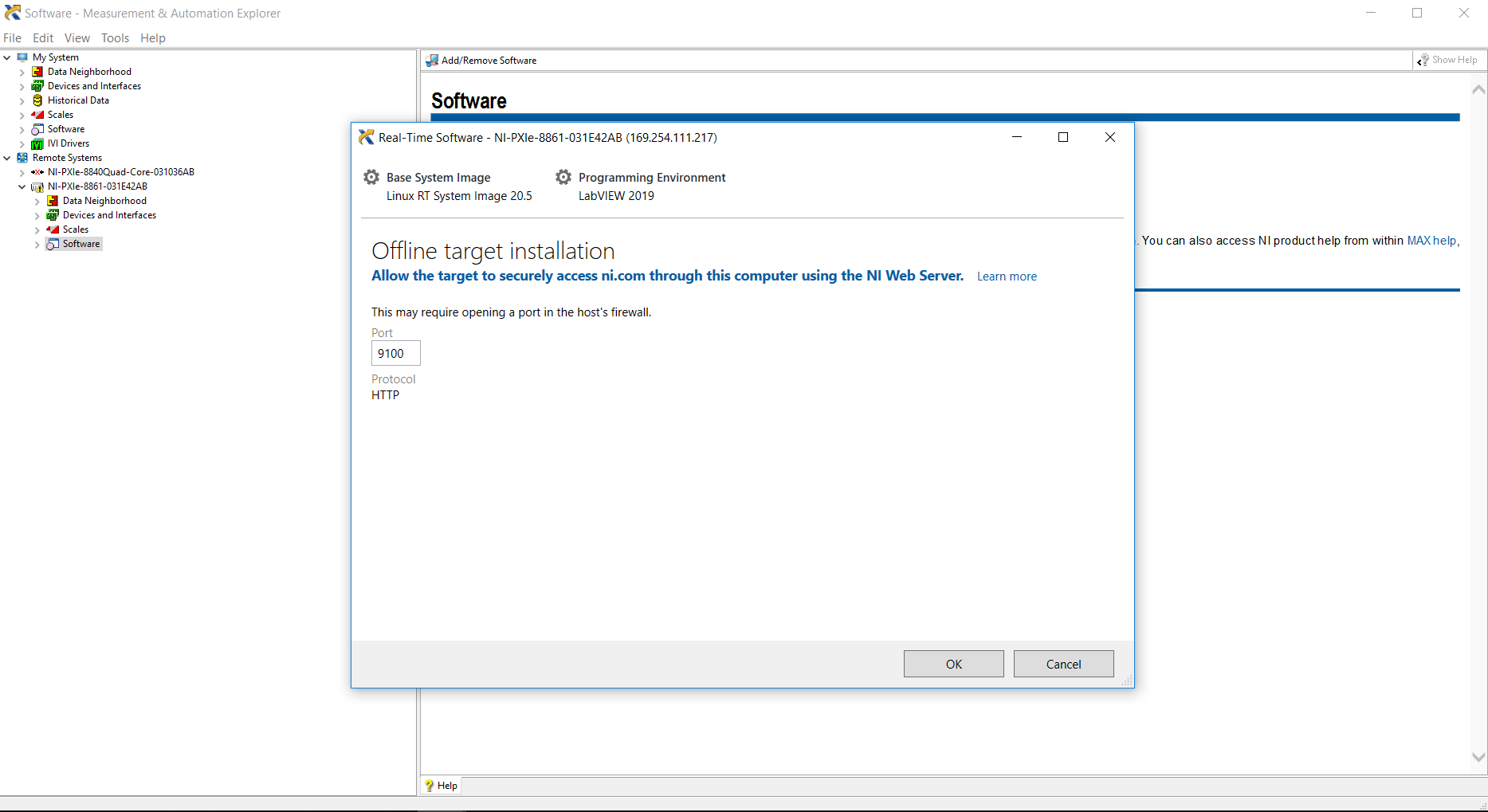




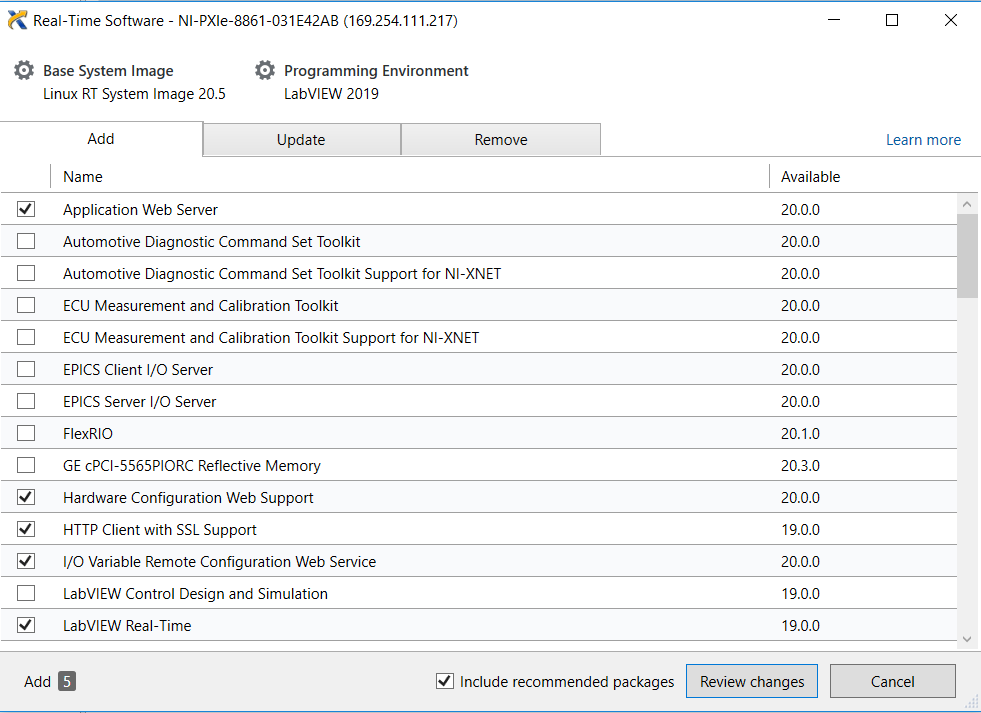
* 1. Select LabVIEW 2019

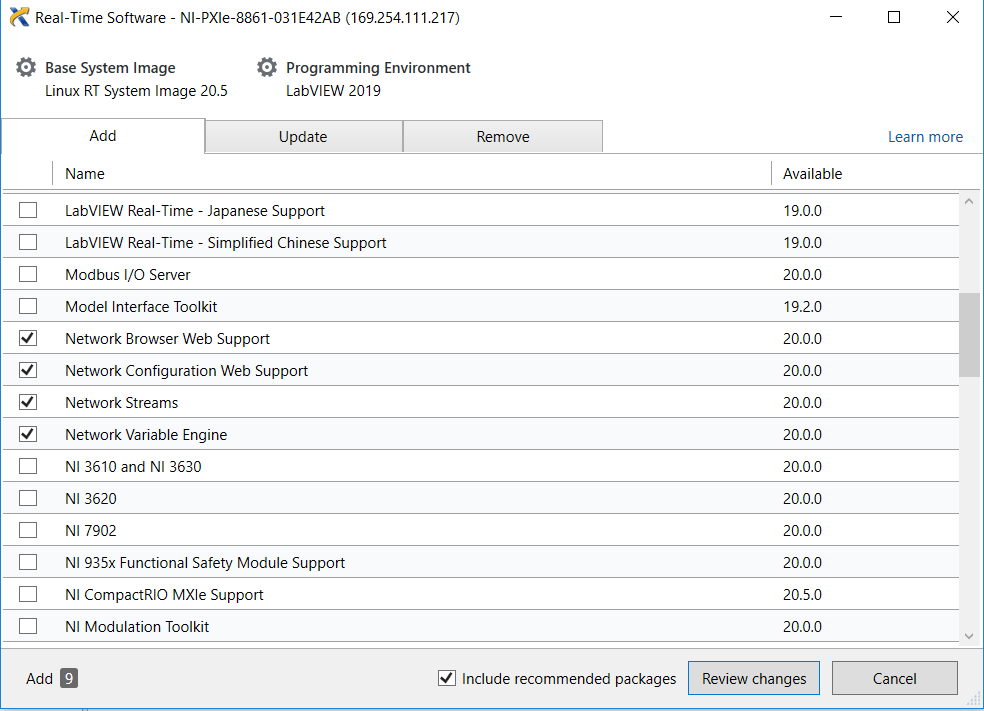


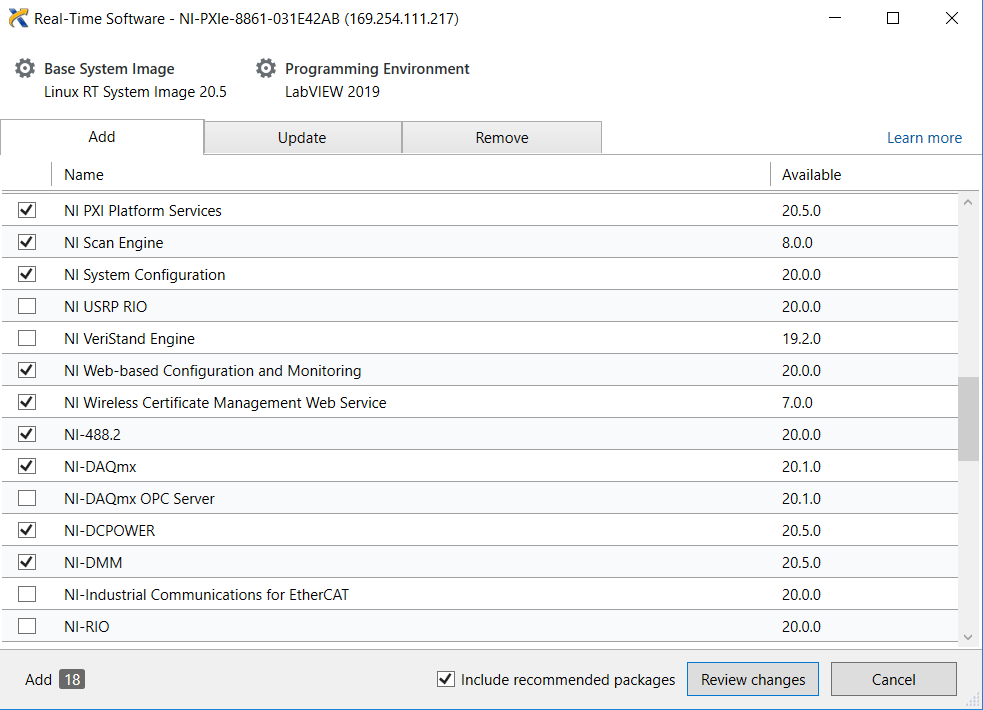
* 1. Decide on whether you want the controller to access ni.com using NI Web Server
     1. If so, then select a port

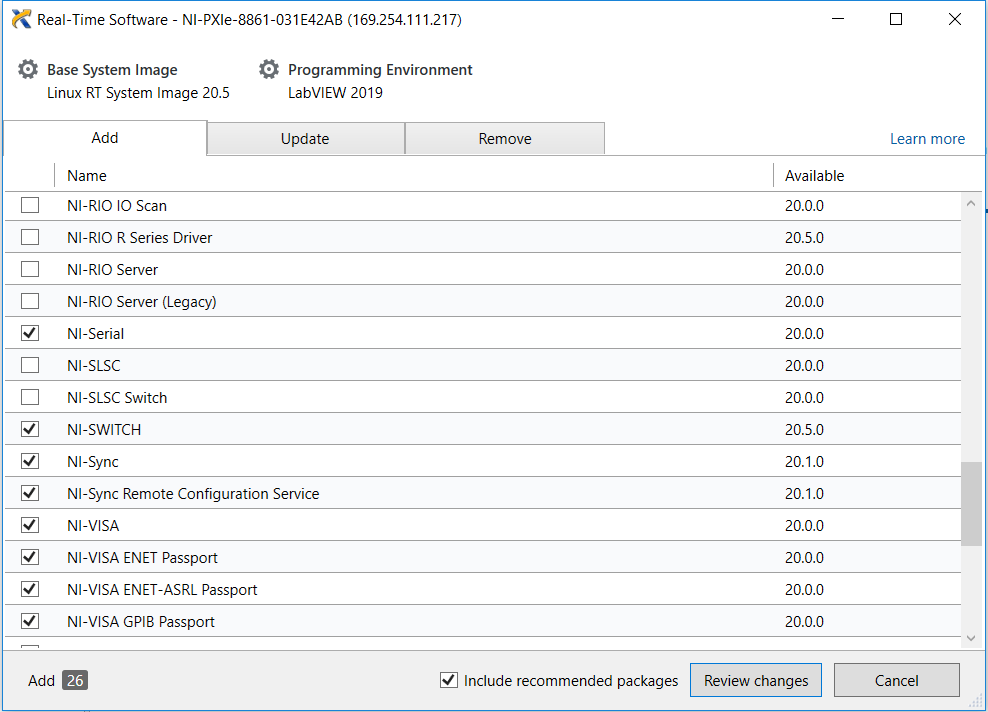


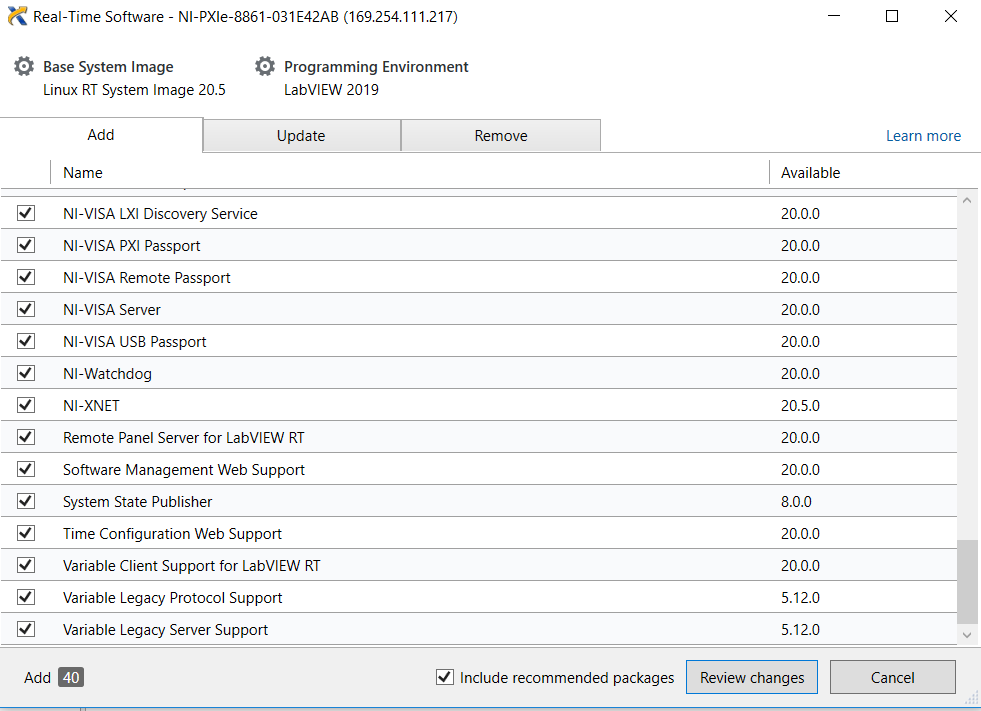
* 1. Select the software to Add



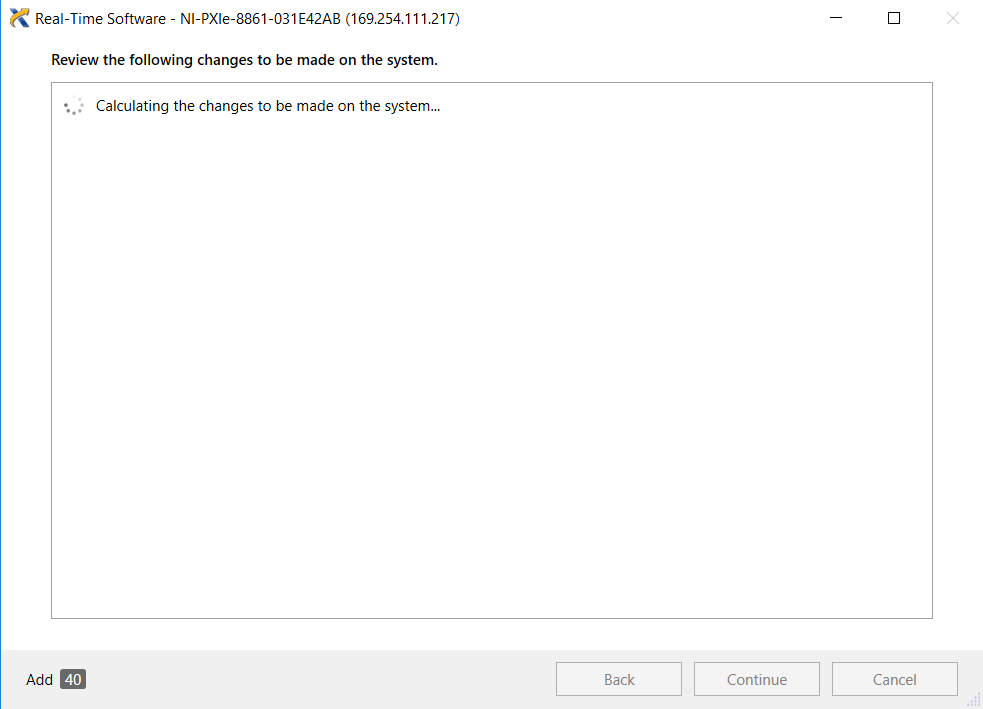


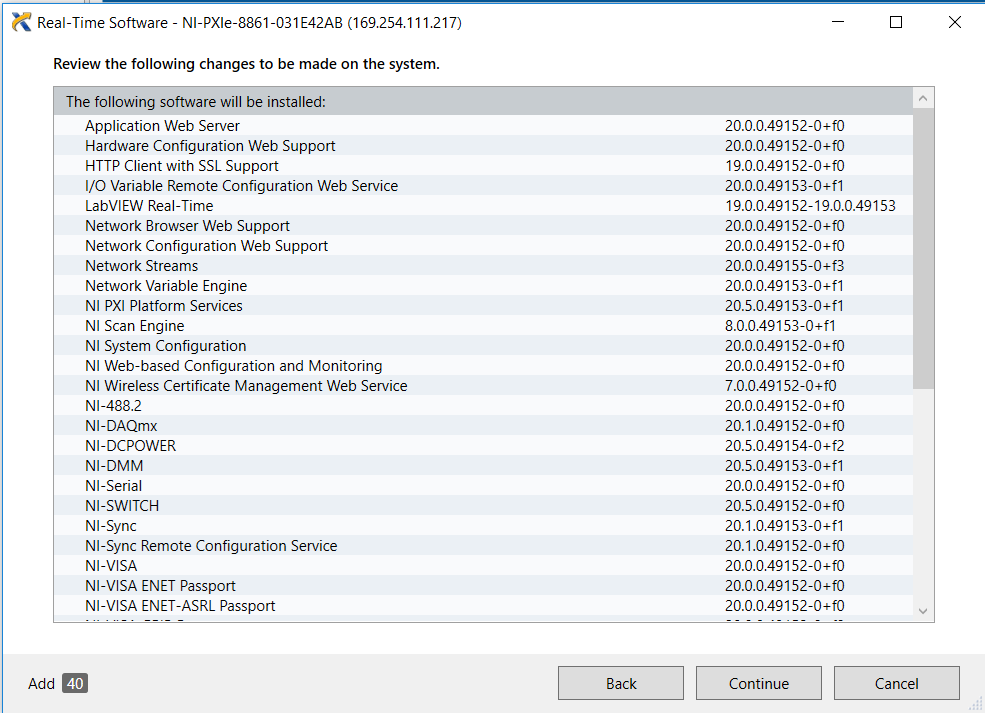


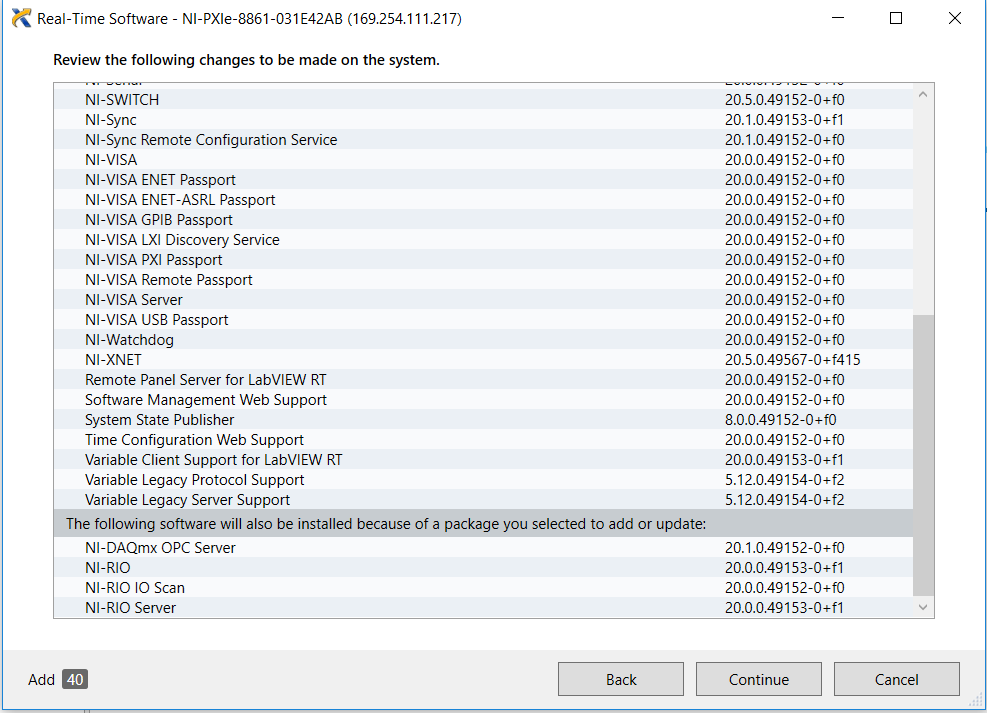




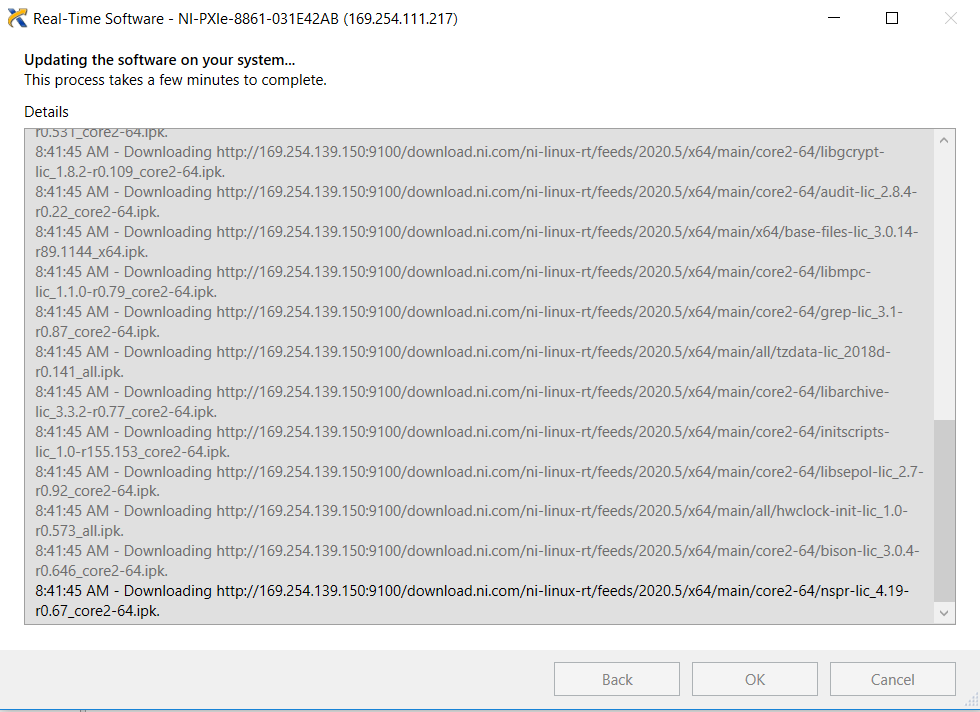
* 1. Press Review Changes to proceed with the install of software to the RT controller

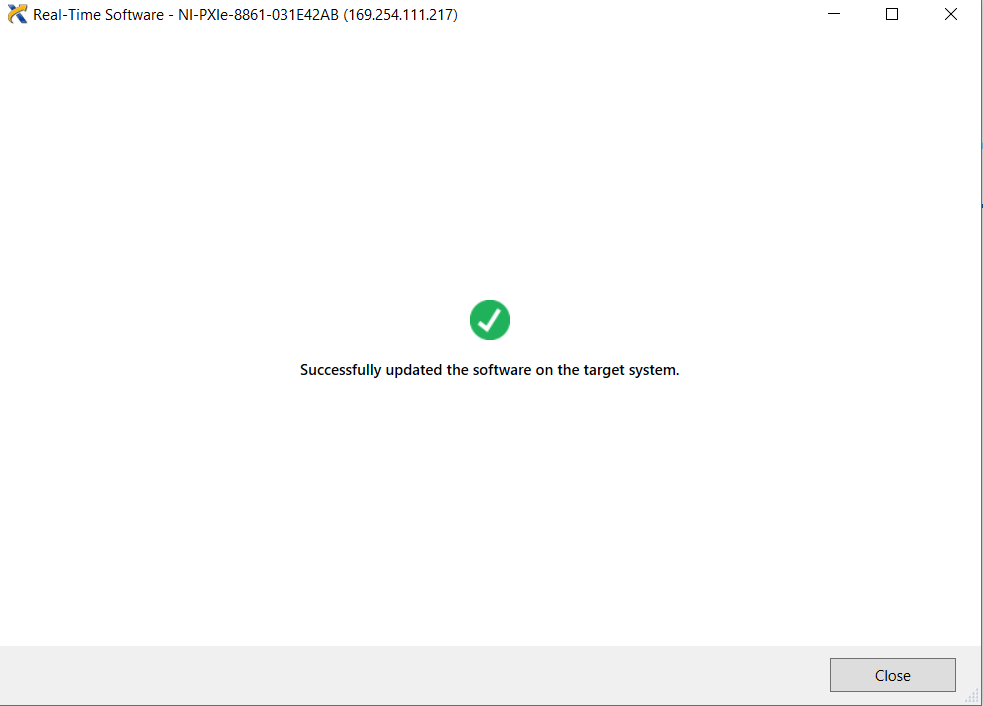




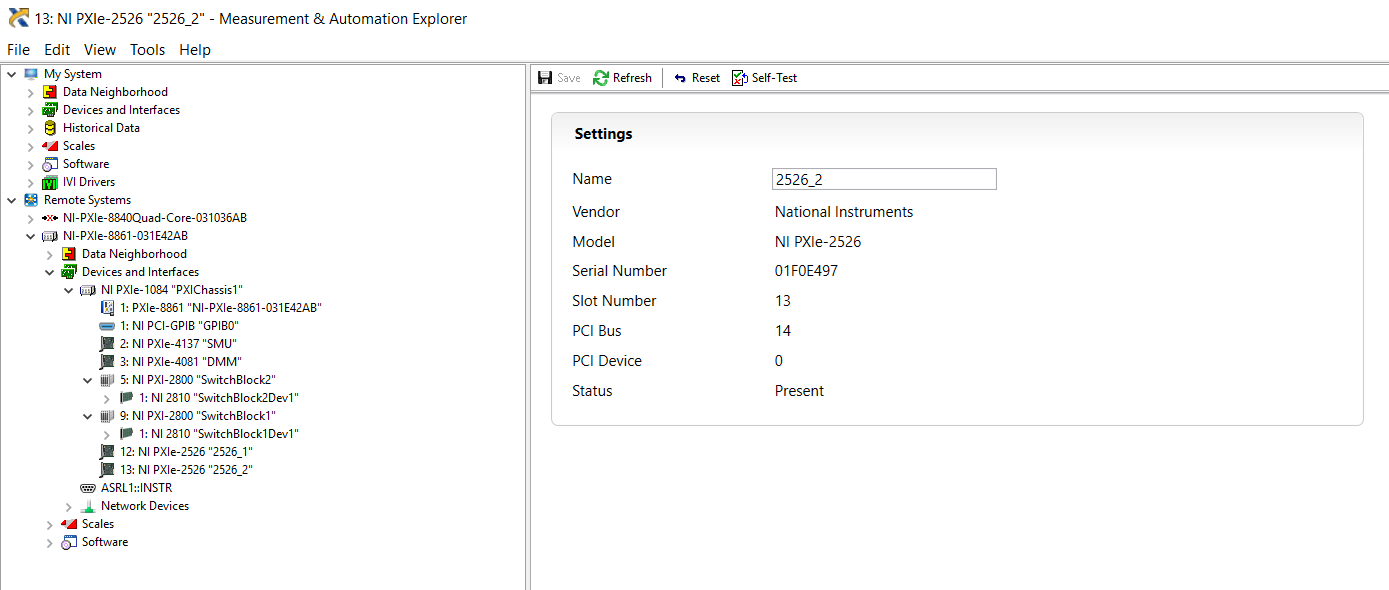


* 1. Press Continue





* 1. Click Set Permissions to set a new password for the RT controller
     1. Default password is blank
  2. Rename the device descriptors for the instruments in MAX
     1. Combine switchblock modules



* 1. Connect controller to internet to obtain gRPC source code from Github

On NI Linux RT…

NI-PXIe-8861-031E42AB login: admin

Password: password (or whatever you choose to be your password in step 12)

Should then show the following…

admin@NI-PXIe-8861-031E42AB:~#

As admin, Install required packages not installed by default

> opkg update  
> opkg install git  
> opkg install git-perltools  
> opkg install cmake  
> opkg install g++  
> opkg install g++-symlinks

*From <*[*https://github.com/ccifra/Labview-grpc-measurement-server#building-on-linux-rt*](https://github.com/ccifra/Labview-grpc-measurement-server#building-on-linux-rt)*>*

Download the repo and update submodules. This will pull the gRPC components and all dependencies.

* + git clone <https://github.com/ccifra/Labview-grpc-measurement-server.git> labview-grpc-query-server
  + cd labview-grpc-query-server
  + git submodule update --init --recursive

Update the proto file if necessary (measurement\_service.proto)

Update any source code if necessary (measurement\_server.h, measurement\_server.cc, server\_interop.cc, event\_data.cc)

Build

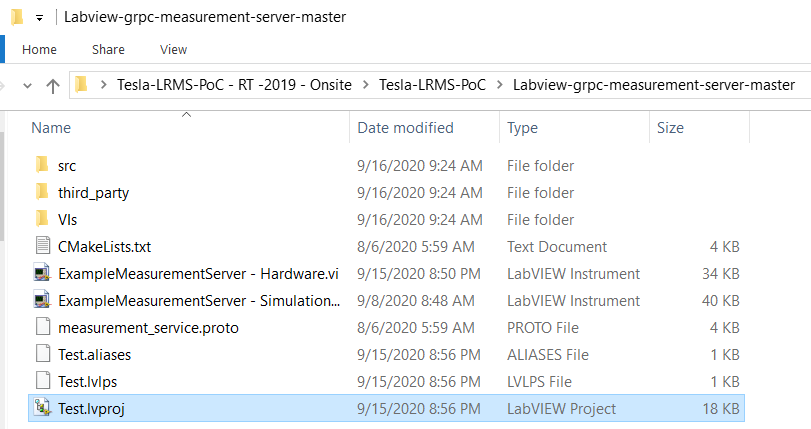
* + cmake .
  + make

The result of this build is a resultant .so file (i.e. liblabview\_measurement\_server.so). This file will be used in call library function nodes in your LabVIEW gRPC VIs.

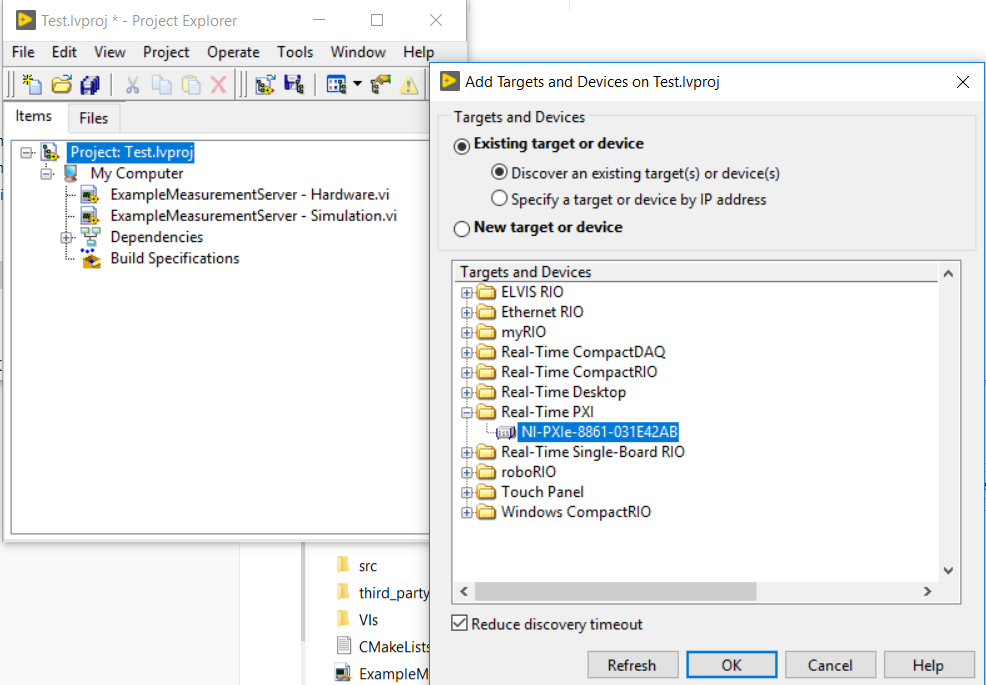
* + Transfer jsontestfile.json from dev machine to RT controller
    1. Temporarily this will be done via USB key (can also be performed by LabVIEW package deployment)
    2. Place in labview-grpc-query-server directory
       1. Copy from USB key
       2. Change permissions so all users can access this JSON config file

* + cd /media/sda1
  + cp jsontestfile.json /home/admin/labview-grpc-query-server/
  + cd /home/admin/labview-grpc-query-server
  + chmod a+rwx jsontestfile.json

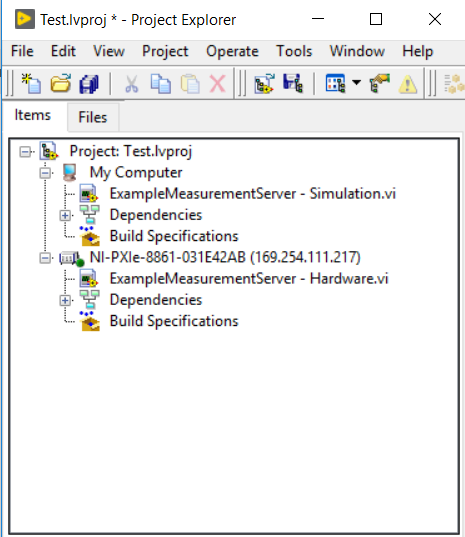
* + Deploy code from dev computer to RT controller
  + Open Test.lvproj



* 1. Verify the controller is configured correctly or add the target so that it is correct
     1. Add target by right-clicking on Project, hover over New >> Targets and Devices

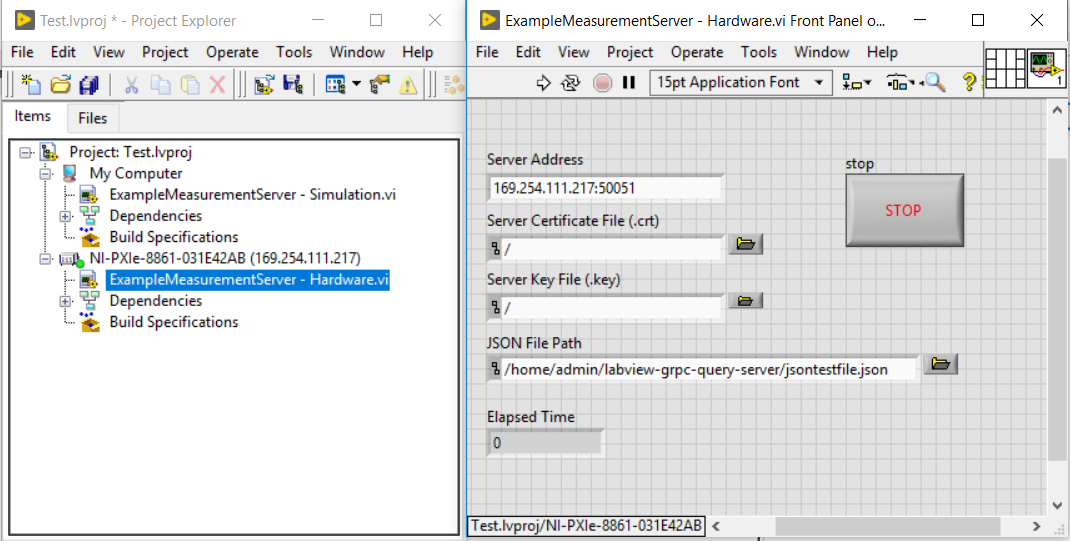


* 1. In the project, move ExampleMeasurementServer - Hardware.vi to the Real Time Target (this VI may also simply be named ExampleMeasurementServer.vi)

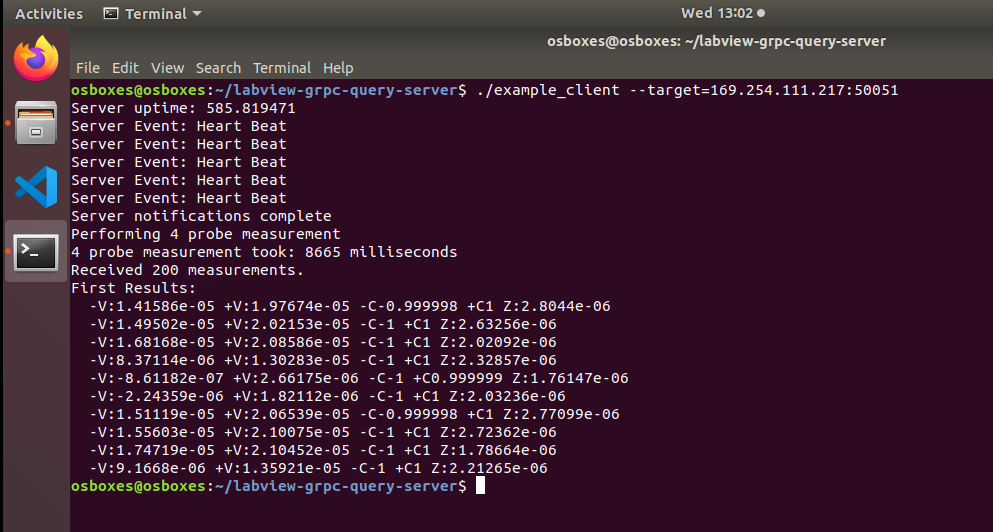


* 1. Connect to the RT Target by right-clicking on the target and selecting Connect. The LED should turn from a dark green to a lit up, bright green if successful.

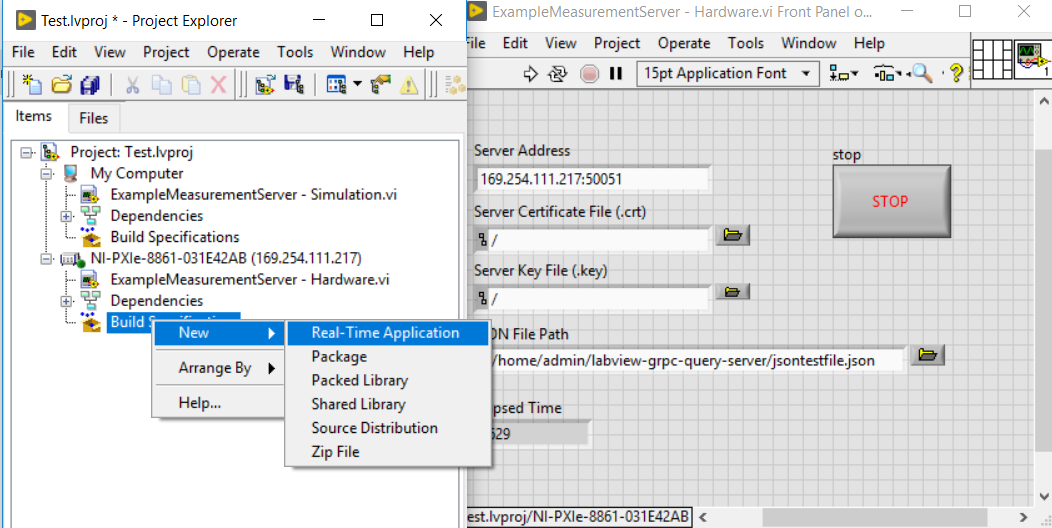
* 1. Verify IP address and jsontestfile.json file location is correct. Run the VI to deploy it to the controller

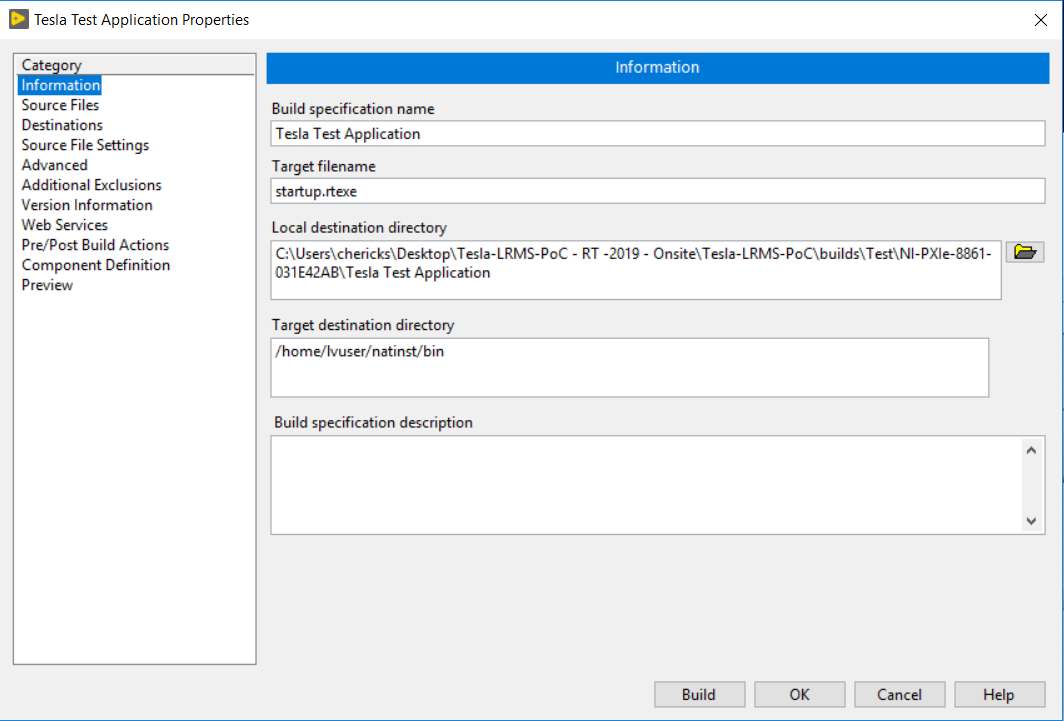


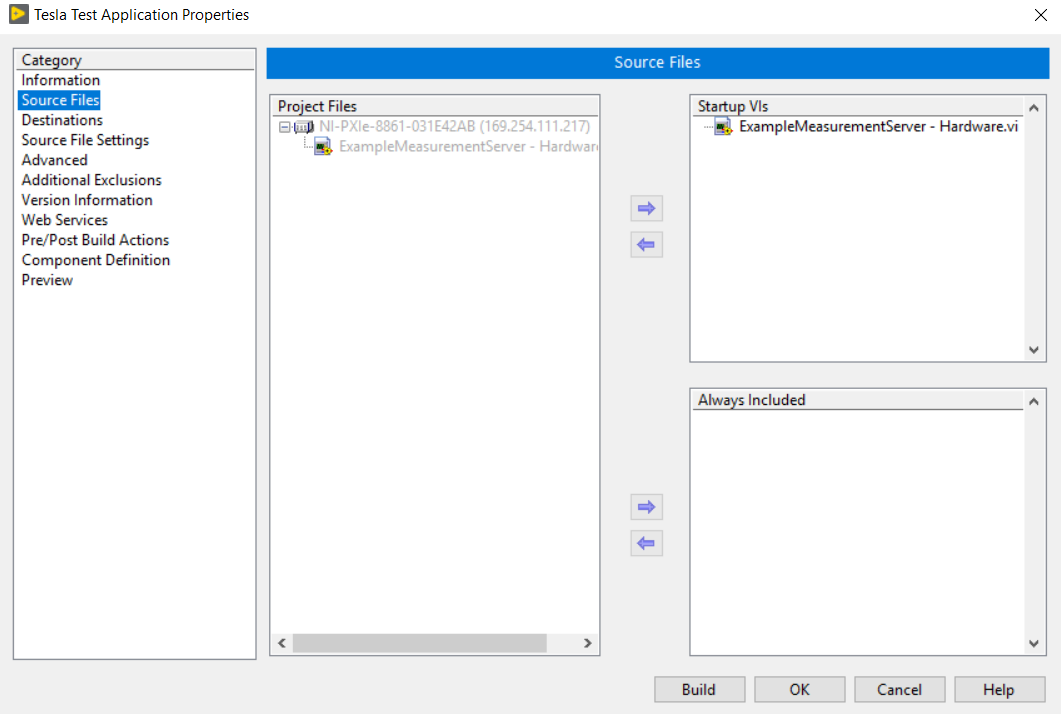
* 1. On the NUC, execute the client to request a 4 probe measurement



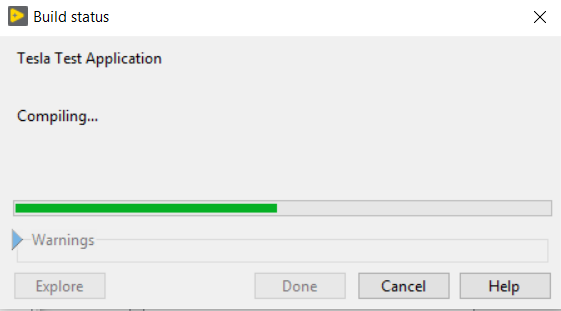
* 1. Once the server code is validated to work as expected, create a build specification to create a RT startup executable. The RT (real time) startup exe will enable the server VI to run by default whenever the PXI RT controller is powered on, or rebooted.

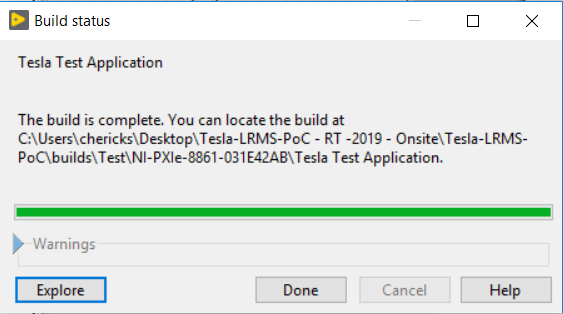




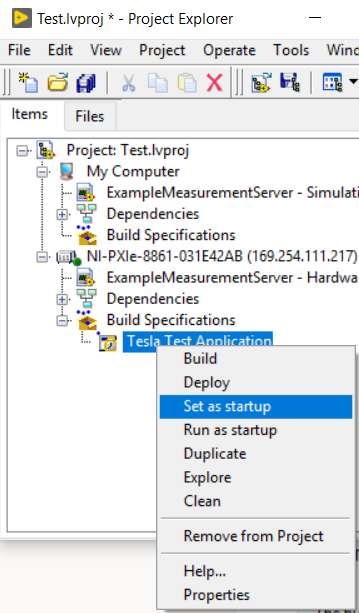


Press Build

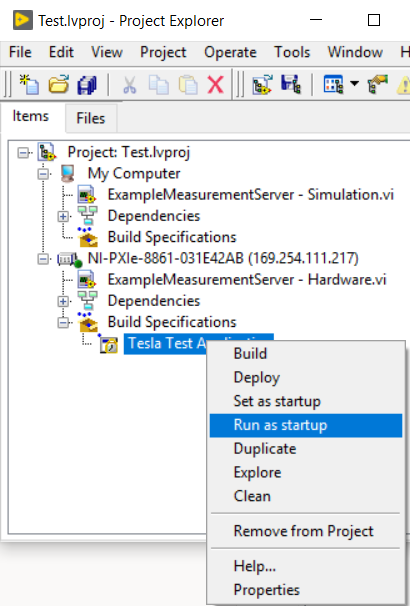




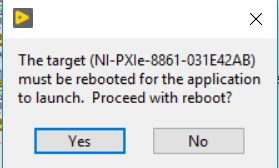
Set as startup



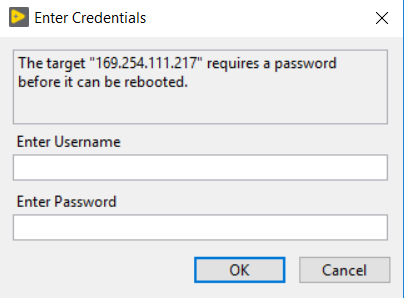
Then, Run as startup



Reboot when prompted.



You may need to enter user name and password credentials, i.e. admin password



Congratulations, your exe is now deployed!

You can now create a Build Specification for a Package deployment. You can deploy the jsontestfile.json and liblabview\_measurement\_server.so files via Package. Configure the package to place these files in the /home/lvuser/ directory. Configure the package to run the .exe as a startup executable.

Once built, transfer the package to the real time target and run it using the following terminal command

* + opkg install [package name]

Example:

* + opkg install lrms\_1.0.0.0\_x64.ipk