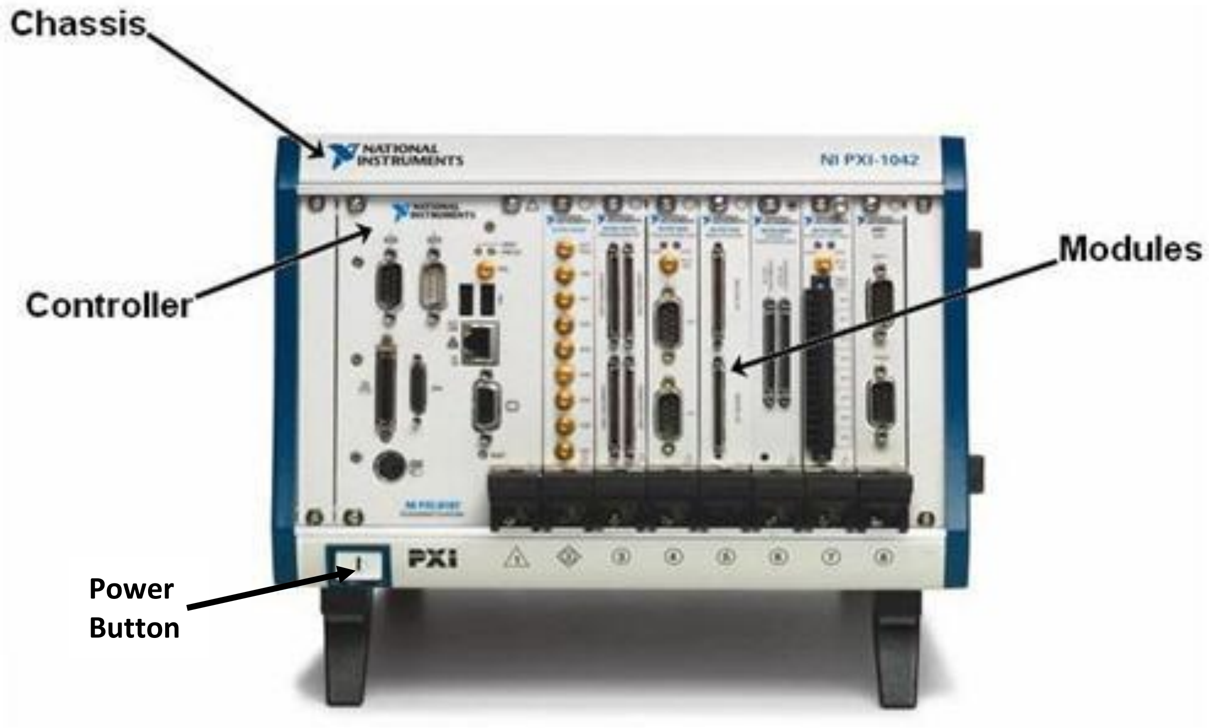
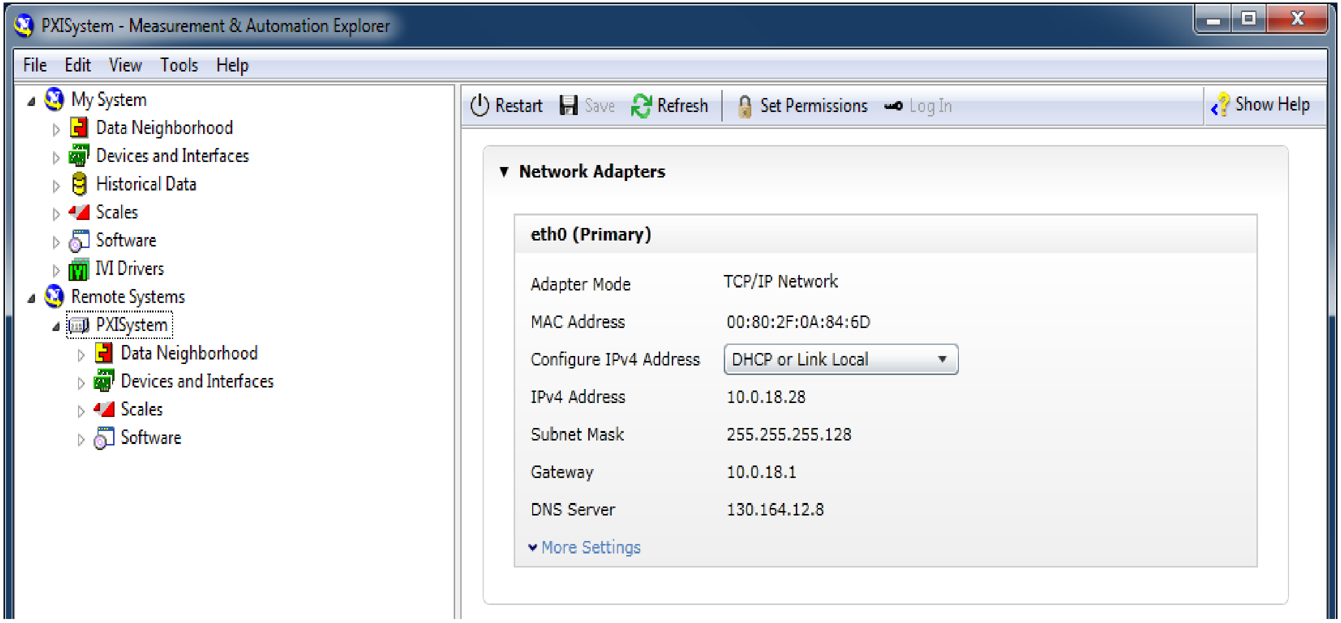
Working with NI PXI:



1. Turn on the chassis (NI PXIe-1062Q), by pressing the power button.
2. In most PXI systems in 1-004, there are 3 modules:
    a. NI PXIe-8133 Embedded Controller – Slot-1
    b. NI PXI-7852R FPGA card – Slot-2
    c. NI PXIe-6363 X-series multifunction DAQ – Slot-3
3. For analog input and output connections, we will use the NI BNC-2110 or the NI BNC-2120 connector blocks.

4. The PXI system is connected to each lab station computer via. Ethernet.
5. The NI PXIe-8133 Embedded controller runs its own real-time operating system and takes a few minutes to boot up. Once it is booted up, open up the Measurement and Automation Explorer (MAX) on the computer and verify that you see the PXI chassis under Remote Systems.



Network Adapters

eth0 (Primary)

Adapter Mode TCP/IP Network

MAC Address 00:80:2F:0A:84:6D

Configure IPv4 Address DHCP or Link Local

IPv4 Address 10.0.18.28

Subnet Mask 255.255.255.128

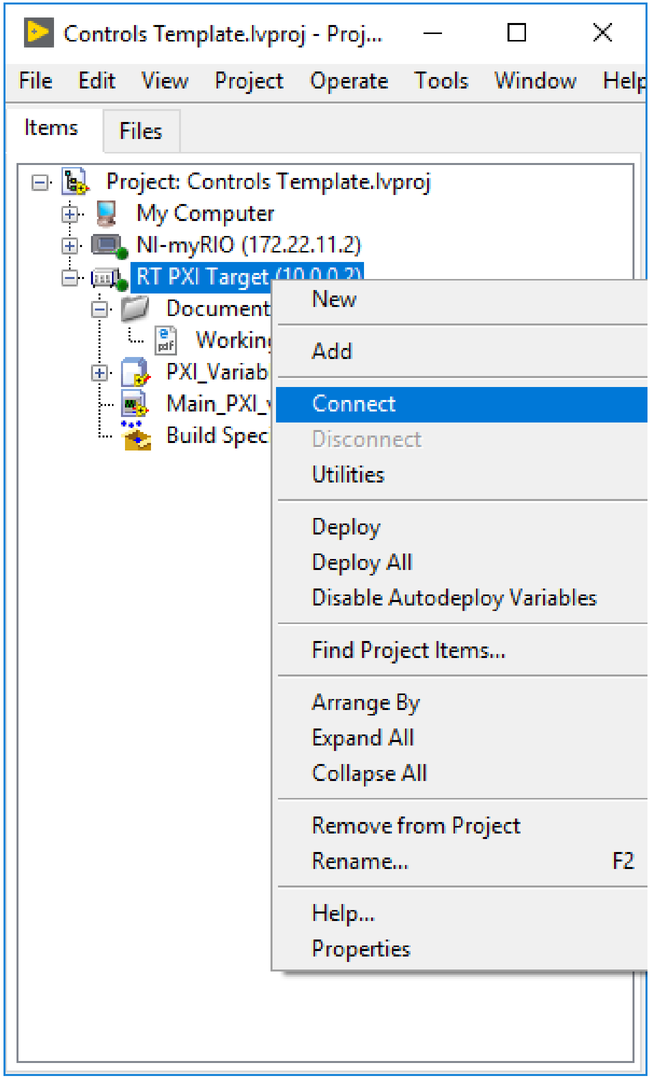
Gateway 10.0.18.1

DNS Server 130.164.12.8

More Settings

6. Remember to shut down the PXI chassis by pressing the power button at the end of your work on the PXI system.

7. Now, open the LabVIEW template project file and edit/run your code.
   a. Connect to the PXI system: right-click -> Connect. Then open the “Main_PXI” vi.



   b. Analog inputs and Outputs correspond to the Analog inputs/outputs of the BNC-2110 or BNC-2120 connectors.
   c. Make sure you run your code as fast as you can. Test that you are not finishing late by checking the #Finished Late counter on the front panel.
   d. Make changes to your controller as you see fit inside the Timed Loop. CD&Sim loop does not support hardware timing and therefore cannot be used in PXI systems. You will need to implement your controller transfer functions in discrete time. Use MATLAB to convert the controller into discrete time using the c2d function. If you are converting from a continuous time controller, use the ‘tustin’ method. *Reference* - https://www.youtube.com/watch?v=yetLPW9sQaI&list=PLUMWjy5jgHK0MLv6Ksf-NHi7Ur8NRNU4Z&index=3
   e. You can modify the location where reference signal adds onto the control loop. It can be either at the location of the reference or just before the output by changing the

variables. You should be familiar working with RT FIFO variables from what you did in Lab-3 and Lab-4.

f. It is important to measure the dynamic response of the plant, and the return ratio after your design the controller for this lab. Please use Dynamic Signal Analyzer code either embedded in the template or as a standalone code posted on stellar. You could also measure the frequency response by manually sweeping sine-waves over multiple frequencies and capturing the magnitude and phase.
g. **LabVIEW implementation:** Ideally, for real-time applications, there should be no user-interface (UI) elements in the code running on the Real-Time (RT) controller or the FPGA. All user-interface code should run on the Host-PC with data transmission between the Host-PC and the RT-target using Network Streams or STM Messaging. See their example code for working templates.
h. **Useful references**
    i. http://www.ni.com/product-documentation/4322/en/
    ii. http://www.ni.com/product-documentation/5489/en/#nidaqmxhelp (Tells how to access DAQmx help)
i. Please post any questions on Piazza.