For a first time installation, plan for approx. 2-3 hours to complete the instructions.
A. Chose a setup

In general, there are several options to distribute PHY and MAC functionality.

- The PHY can execute either one or on two FPGA targets.
- Supported FPGA targets for the PHY are USRP-RIO (bandwidth 40 MHz, 120 MHz, 160 MHz).
- The MAC can execute either on the RT Host or on a separate cloud PC.
- eNB and UE MAC can run on one or on two machines with real-time operating system.

The setups below have been tested. In theory, a mix and match of these configurations should be also possible.

1. Downlink only

This is a compact setup that is useful for tradeshows.

2. eNB and UE separate (recommended)

This is the recommended setup with physically separated nodes.
3. PHY and MAC separate

It is also possible to run the NS-3 on a Linux machine other than the PXI controller.

4. MAC and higher layers only

This setup bypasses the PHY completely. It is useful for isolated debugging of MAC and higher layers functionality.
B. Prepare the PXI controller

First, provision a PXI controller as real-time target for LabVIEW Communications. This involves the following steps:

1. Upgrade BIOS of the PXI controller (optional).
2. Install the NI Linux RT operating system.

You need:

- Generic PC with Windows operating system
- PXI controller NI 8135, with keyboard and monitor and connected to the Internet
- USB stick (2 GB or larger)
- Currently, a side-by-side installation of Windows and NI Linux RT is not supported. If you want to keep your pre-existing Windows setup on the PXI controller, it is recommended to swap the internal SSD.

5. Upgrade BIOS

A BIOS upgrade is recommended, if your controller runs with a BIOS below v1.3.0. You can enter the BIOS Setup Utility and check the BIOS version by pressing DEL while controller boots.

1. Download and unpack the files necessary for the BIOS upgrade. MAKE SURE YOU CHOSE THE VERSION THAT MATCHES WITH YOUR CONTROLLER:
2. On a Windows machine, insert the USB stick. It is recommended to use an account with admin privileges. Run CreateUSBUtilityDisk.exe and follow the instructions.
3. Power on the PXI controller.
   - While the controller boots, press DEL to enter the BIOS Setup Utility. Navigate to Advanced -- USB and ensure that Legacy USB Support is set to ENABLED. Press F10 to save and reboot.
   - While rebooting, press DEL to enter the BIOS Setup Utility again. Navigate to the Boot menu. Follow the on screen help to change the boot order so that the USB stick is the first boot device. Press F10 to save and reboot.
4. Insert the USB stick with the BIOS upgrade in the PXI controller and reboot. Once the controller boots into a DOS console, follow the onscreen instructions in order to flash the new BIOS. (Type FLASHDOS.bat and press Enter.) Once the upgrade process starts, be sure to let the program complete. Otherwise the can be damaged and become unbootable.
5. Once the upgrade is done, remove the USB stick and reboot the controller.

6. Install NI Linux RT

2. Create bootable USB stick (can be the same that you used for the BIOS upgrade before). For example, on a Windows machine, insert the USB stick and use Rufus (https://rufus.akeo.ie/). Select DD mode when asked. Press Start.

3. Connect the PXI controller to the Internet, plug in a display and a keyboard.

4. Insert the USB stick into the PXI controller on which you want to install NI Linux RT.
   a. Reboot and press F10 during startup. Select the USB stick.
   b. Confirm repartitioning of the system. This will overwrite the content of the internal disc of the PXI controller. At the end of the process, remove the USB stick and reboot the system.
   c. After boot, press enter to open command line. The default username is root and the account has no password.

5. Use ifconfig to identify the IP address of the controller. You can now access the NI Linux RT system remotely, e.g. with putty.exe, which can be obtained from here:
6. Optional: Confirm RT kernel version with `uname -r`.

```
root@v0007:~ # uname -r
4.1.13-rt15-nilrt
```
7. Run `opkg update` to refresh repository cache.


7. Build and test NS-3

After the NI Linux RT system is running and can be accessed remotely, the next step is to get the NS-3 code and compile it on the NI Linux RT controller.

1. Copy the NS-3 source code to the PXI controller
   - Install WinSCP ([https://winscp.net/eng/download.php](https://winscp.net/eng/download.php)).
   - Connect to the PXI controller either using its host name or IP address. The default username is `root` and the account has no password. Accept keys if asked.
• Upload the NS-3 source code from the Windows machine to `/home/root/ns-3` on the PXI controller.

2. Install required packages
   • Run `opkg install gcc g++ libstdc++ gcc-dev binutils python-pip python-dev boost boost-dev pkgconfig make rsync tcpdump sudo` to install required packages.
   • Run `opkg install packagegroup-core-buildessential` to set up gcc environment.
   • Type `visudo`, navigate down to the line that starts with `root All=`... and add `lvuser ALL=(ALL) NOPASSWD: ALL` below it. (Press `i` to insert text. Press `Esc`, type `:x` and press `Enter` to save the document. VI cheat sheet might be useful: [http://ryanstutorials.net/linuxtutorial/cheatsheetvi.php](http://ryanstutorials.net/linuxtutorial/cheatsheetvi.php)

3. Build NS-3
   • Run `cp /home/root/ns-3/NI_API_files/resources/conversion_fixed.hpp /usr/include/boost/date_time/gregorian/conversion.hpp` to fix conversion.hpp related compile error.
   • Run `cp /home/root/ns-3/NI_API_files/resources/pkgconfig/NI_Linux_RT/*.pc /usr/lib/pkgconfig` to copy pkgconfig related files to system directory.
   • Type `chmod -R 777 /home/root/ns-3` to change file permissions. The output of `ls -la /home/root/ns-3` should now be `-rwxrwxrwx` or `drwxrwxrwx`.
   • Run `ulimit -s unlimited` to adjust the maximum number of open files.
- Switch to NS-3 main directory with `cd /home/root/ns-3`.
- Run `./waf configure`. The expected output is:
Run **make**. During the compilation, the output should start with `[xxxx/1860] cxx:

...Successful output:

```
```

4. Test NS-3

- Make sure you are still in the NS-3 root folder (`cd /home/root/ns-3`).
- Edit configuration file with `vi configs/config_loopback`. Make sure the four IP address match with the current output of `ifconfig`. Use `i` to insert text. Press `Esc`, type `:x` and press `Enter` to save and close the file. An example config file for downlink and uplink loopback operation should look like this:

```plaintext
ns3_base_dir="/home/root/ns-3/
simTime="300"
msg_print_on="false"
enableLogging="0"
AppPacketInterval="1"
AppPacketSizeByte="50"
Phy2PhyDelayMeasEnabled="false"
enbMacToChannelDelay="7"
OneWayLatency="6"
enableTapBridge="false"
enb_mac_ip_address="10.0.36.41"
enb_phy_ip_address="10.0.36.41"
enb_mac2phy_tx_port="10201"
enb_mac2phy_rx_port="10700"
enb_phy2mac_tx_port="10301"
enb_phy2mac_rx_port="10600"
ue_mac_ip_address="10.0.36.41"
ue_phy_ip_address="10.0.36.41"
ue_phy2mac_tx_port="10401"
ue_phy2mac_rx_port="10700"
ue_mac2phy_tx_port="10501"
ue_mac2phy_rx_port="10600"
dl_loopback="true"
ul_loopback="true"
```
• First, start UE with `/home/root/ns-3/start_ns3_ni.sh UE config_loopback`
• Then, in a second terminal, start eNB with `/home/root/ns-3/start_ns3_ni.sh eNB config_loopback`

Successful output:

5. Troubleshooting
• NS-3 compilation fails with segmentation error: .... Most likely `ulimit -s unlimited` is not set.
• NS-3 compilation fails at Checking for g++: .... Run `opkg install packagegroup-core-buildessential` to set up gcc environment.
• Copy the pkg-config related files to the system directory with `cp /home/root/ns-3/NI_API_files/resources/pkgconfig/NI_Linux_RT/* .pc /usr/lib/pkgconfig`. 

![Initial Attach](image.png)
![Payload Exchange](image.png)
C. Prepare the development PC

The next step is to install the LabVIEW Communications Design Suite 2.0 and the LTE Application Framework 2.0.1 on a Windows PC.

You will need:

- PXI controller with NI Linux RT installed and NS-3 working.
- Windows 7 based host system (PC or laptop).
  Recommended specifications: Intel i7 processor, 16+ GB RAM, SSD drive
- Administrator privileges

1. Instructions

1. Install LabVIEW Communications Design Suite 2.0 and LTE Application Framework 2.0.1 on the Windows PC.
2. Open LabVIEW Communications 2.0 and register the NI Linux RT target.
   - Go to Network Systems (top right corner).
   - All RT systems in the network are automatically detected. The PXI controller should be visible as an unregistered system.
   - Click on the target and press register. If successful, the target will appear in the list of registered systems.
2. Test the configuration

1. Create a new LTE Application Framework project, chose USRP-RIO.

2. Save the newly created LTE Application Framework project.
3. Go into System Designer and adjust the system according to your components.
   - Delete everything from the diagram.
   - In the palette, go to Discovered Hardware --> Local Devices. You should see PC on which you are running LabVIEW Communication. Drop it into the diagram.
   - In the palette, go to Discovered Hardware --> Network Systems. You should see the controller that you registered earlier. Drop it into the diagram.
   - Optionally, configure the controller host name in the right rail.
   - Connect the Network Adapter of the Windows PC and the NI Linux RT controller.
   - Drop LTE Host DL.gvi into software field of the controller.

- Make sure the RF is wired for downlink only operation.

- Run LTE Host DL.gvi, wait until the FPGA Ready signal in the top right corner becomes true and turn on eNB Transmitter and UE Receiver.
If everything goes well, you should see data in the power spectral density, constellation and throughput plots. In that case the system is configured properly.

4. Troubleshooting

- It is recommended to start on a machine with a fresh Windows installation
- If using USRP-RIO devices connected via MXI to PXI don’t connect it in system designer as then the device mapping is lost and e.g. the RIO identifier cannot be changed.
- For troubleshooting, sometimes it can be necessary to get the system configuration and version numbers of the system components. Log into the PXI controller and type `lsni -v`

Sometimes the “Registered Hardware” tab states that the network manager initialization in has failed. Reboot the Windows PC and PXI controller.
D. Run the NS-3/LTE project

After completing sections A, B and C, the final step is to run the NI NS3/LTE Network Prototyping Application Example.

1. Prepare the LTE PHY
   1. Make sure that the setup is wired correctly (section A).
   2. Make sure that the RT target is registered (section C.1).

4. Go to the folder LTE v2.0.1\NS3_LTE. Depending on your hardware, open the project
   - NS3_LTE Design USRP RIO v2.0.1.lvproject
5. Add new PXI system in the system diagram:
6. From the left rail in LabVIEW Communications, open the *.lvproject file and make sure the displayed setup matches your real-world setup. (Picture shows the Downlink only configuration from section A.1.)
7. Depending on your hardware configuration, open the main VI(s)
   • NS3_LTE Host DL.gvi and L1L2 DL.gvi when working with setup A.1.
   • NS3_LTE Host UE.gvi and NS3_LTE Host eNB.gvi and L1L2 UE.gvi and L1L2 eNB.gvi when working with setups A.2 or A.3.

8. In the red GUI, make sure the correct PXI target, RIO Device and USRP Bandwidth are selected.
9. In the blue GUI, make sure that in tab **UDP read**, the control **L9 message source** is set to **test**.

![GUI screenshot showing L9 priority control set to 100, L9 pid set to -2, L9 period set to 1000, and L9 offset set to 0. Message source is set to test.]

10. Run the VI with the green button ➤. In the top part of the user interface, start the eNB Transmitter and the UE Receiver.

![LTE Application Framework diagram showing eNB Transmitter and UE Receiver active and connected.]

11. In the bottom part of the interface, enable **PHY plots**.

12. If everything goes well,
   - The indicator **Sync found** should be **True**.
   - The power spectrum plots in the red GUI should match the **Resource Block Allocation** and **MCS** displayed in the **PHY config** tab of the blue GUI.
13. Changing `msg_index` in the **UDP read** tab in blue GUI should change the PSD plots.
2. Prepare and run the NS-3 application

1. Make sure the PHY in the red GUI is not running.

2. Open `configure_setup.gvi` and make sure it is executed on the Windows PC.

3. Set-up the **IPs and ports** tab. Configure the four IP address fields and the two loopback flags. Generally, the TX and RX ports can be left with the default values.
4. Make sure that the folder `C:\temp` exists and that `putty.exe` is placed inside. Alternatively, modify the paths in the `paths and commands` tab.

5. Press the green run button 🔄. This will distribute the API configuration files to PXI target(s). Note that a putty confirmation window might show up the first time this VI is execute. Confirm and proceed.

6. Switch to the red GUI (`NS3_LTE Host XXX.gvi`). Set PHY plots is False.

7. Switch to the blue GUI (`L1L2 XXX.gvi`). In tab UDP read, make sure that the data L9 message source is set to UDP.
8. Start the LTE PHY (as described in section D.1), however this time keep `PHY plots` set to `False`.
9. In the bottom part of the GUI, go to the **XXX NS-3 controls** tab.
   - First press `run UE`
   - Then press `run eNB`
This will send Linux commands to the PXI controller, which will start the NS-3 binaries. The terminal output will be collected in the LabVIEW Communications GUI.

10. To confirm that NI API packets are being exchanged, you can log into the Linux machine using putty and run the command `tcpdump -i any_host 10.0.36.41 and udp -n`. The ports and IPs should match to what is configured in `configure_setup.gvi`. 
11. Note that NS-3 execution is timed. After a certain amount of time, the binaries will terminate themselves.

12. To terminate NS-3 execution manually,
   - First press the button **kill**. This will terminate the NS-3 binaries on the Linux system.
   - Then press the button **stop** to terminate the PHY.

3. Troubleshooting
   - In the **NS-3 controls** tab, if UE or eNB cannot started in LabVIEW, it is worth checking the **sudo** permissions (cf. section B.6).
   - If **L9 message source** is set to **test**, the NS3 application will throw an error.
   - There is the option to use one node (e.g. UE) with an USRP target and another node (e.g. eNB) with a FlexRIO target. In this case, it is necessary to open and configure the USRP-RIO and FlexRIO projects separately. This can be done in parallel instances of LabVIEW Communications on the same system. Due to the code structure, it is not possible to run a USRP-RIO and a FlexRIO target in a single project. Also, two PXI targets are necessary, because two instances of LabVIEW Communications cannot access the same controller at the same time.
   - In case something goes wrong:
     - Press **kill**, press **stop** and start over.
     - Close LabVIEW Communications and start over
     - Reboot the Windows PC and the PXI controller and start over.