

How to Interface NetSim with LabVIEW?

Software: NetSim Standard v13.0 (64 bit), Visual Studio 2019, LabVIEW 2020.

Project Download Link:

// Need to attach link//

Follow the instructions specified in the following link to download and setup the Project in NetSim:

<https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects>

Introduction

NetSim can be interfaced with LabVIEW, which allows users to interact with LabVIEW during run-time. LabVIEW can interact and control the behavior of its Models.

LabVIEW is a system engineering software for applications that require test, measurement, and control with rapid access to hardware and data insights. NetSim now has a method of Interfacing with LabVIEW in a fashion similar to that of MATLAB interfacing. NetSim can initialize a LabVIEW Virtual Instrument (.vi) file during runtime, pass inputs to components of the file, execute the vi, read the computed parameters from its components, and terminate the vi instance at the simulation end.

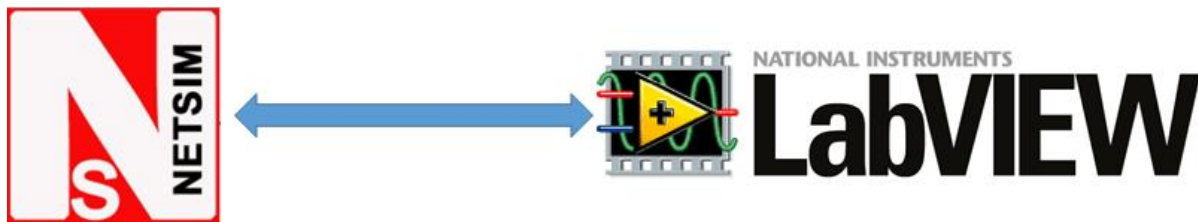


Figure 1: NetSim Interfacing with LabVIEW

LabVIEW Example

In this article, we have considered a simple LabVIEW Example which keeps generating a random number and passes periodically to NetSim during run-time.

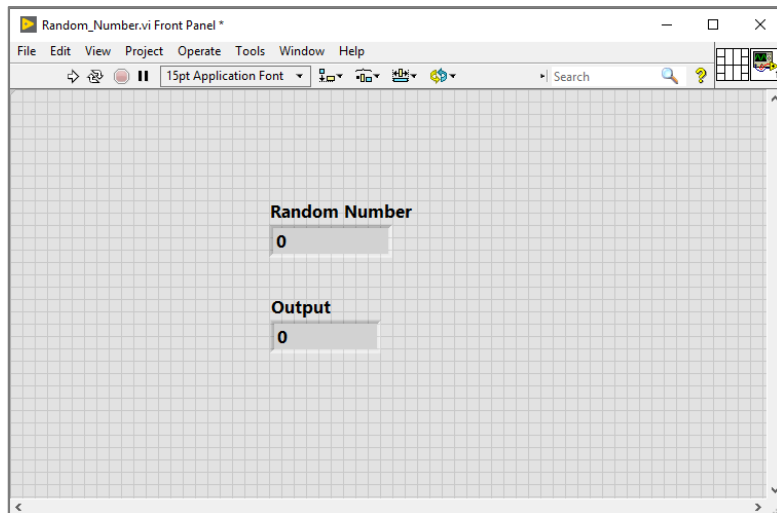


Figure 2: Random_Number.vi window

NetSim-LabVIEW interfacing

Upon interfacing NetSim with LabVIEW the following tasks are performed during simulation start:

- LabVIEW Engine process is initialized.
- LabVIEW GUI window is loaded.
- LabVIEW - Random Number generator Example is loaded.

Upon simulating a network created in NetSim the following tasks are performed periodically:

- The random number starts generating in LabVIEW.
- NetSim reads the data generated by LabVIEW.
- Prints the random number to NetSim Simulation window.

Output/Metrics specific to this example

- NetSim Event Trace - LabVIEW event is registered to periodically interact with LabVIEW.
- NetSim Run-time simulation

Modifications done to NetSim Source codes

Projects: Zigbee

Files:

- 802.15.4.c,
- 802.15.4.h,
- Zigbee.vcxproj (Project file)

Files added:

- Labview_Interface.cpp
- Labview_Interface.h

Procedure to download this project for this example

- Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio: [How to clone Netsim File Exchange project repositories](#)

from Github

- Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note:

It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim

Steps to Simulate.

1. The cloned project will contain folders on Documentation, LabVIEW files and LabVIEW Workspace folder as shown below:

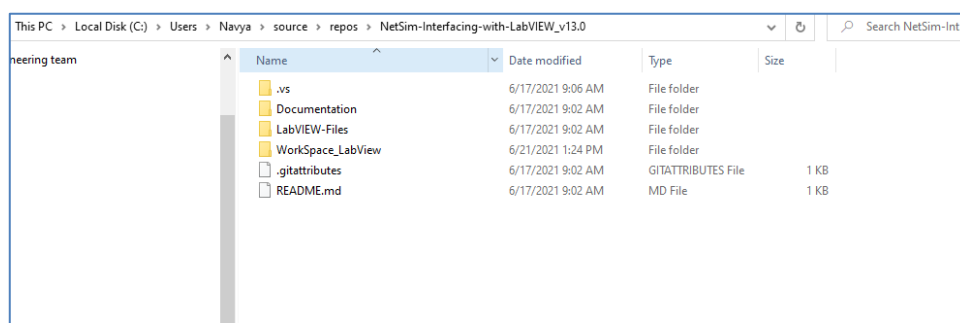


Figure 3: Cloned folder of LabVIEW Workspace

2. Import LabVIEW_Workspace by going to Open Simulation->Workspace Options->More Options in NetSim Home window. Then select Import as shown below:

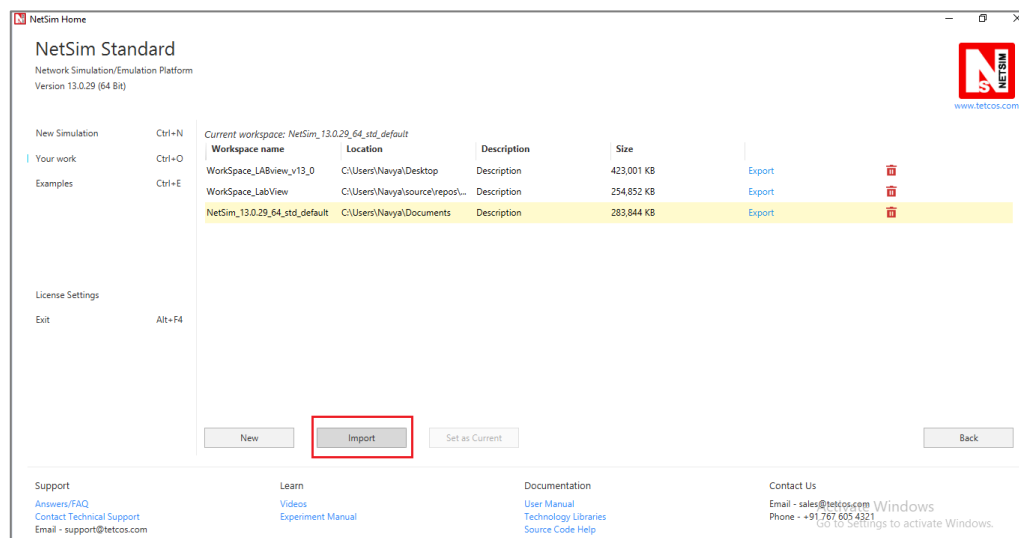


Figure 4: NetSim Home page

3. In the Import Workspace Window that appears to browse and select the *.netsim_wsp file from the cloned project directory and for the Destination path browse to select a path in your system where you want to set up the workspace folder

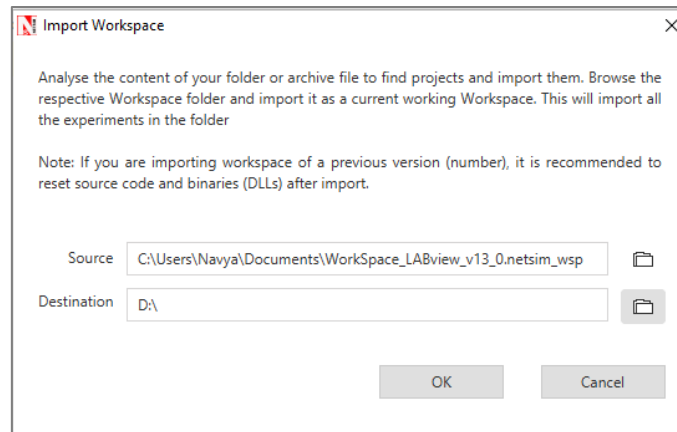


Figure 5: NetSim Import workspace window.

4. The Imported workspace will be set as the current workspace automatically. To see the imported workspace, click on Open Simulation->Workspace Options->More Options as shown below:
5. Open the Source codes in Visual Studio by going to Open Simulation-> Workspace Options and Clicking on Open code button as shown below:

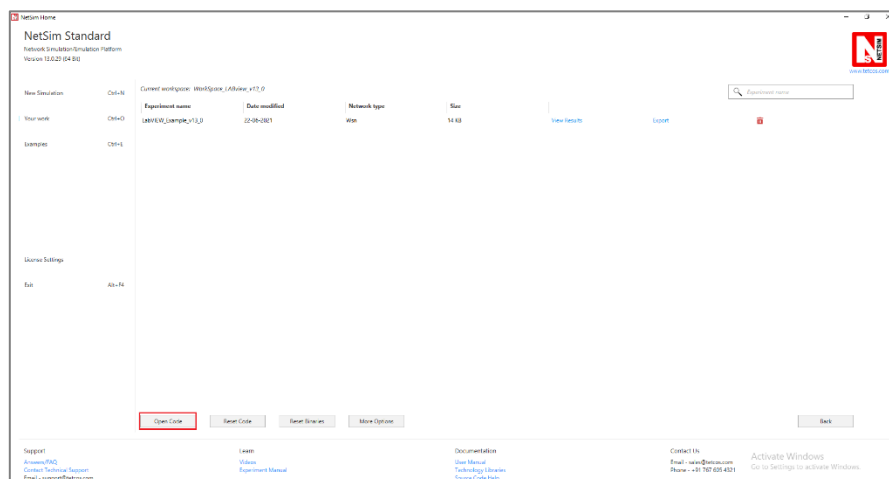


Figure 6: Open source code option in Your work window

6. In Solution explorer, you can observe, the battery model and Zigbee projects where LabVIEW_Interface.cpp and LabVIEW_Interface.h files are present inside Zigbee project which contains source code related to interactions between NetSim and LabVIEW.
7. Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit DLL files respectively as shown below:

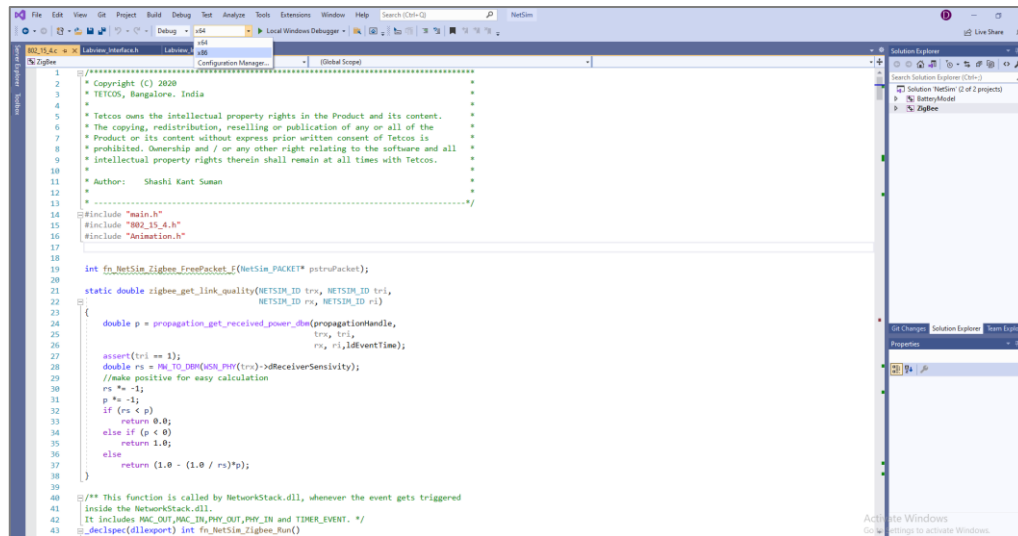


Figure 7: Based on the build of NetSim select Win32/x64 in Visual studio.

8. Change the Labview.tlb path as per your installation in Labview_interface.cpp file.

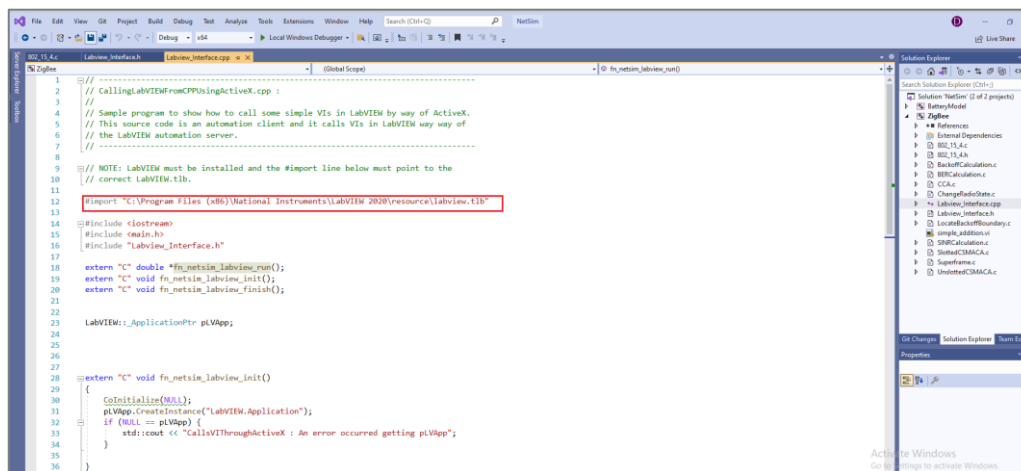


Figure 8: Adding LabVIEW.tlb path in LabVIEW_interface.cpp.

9. Right click on Zigbee project and click on Rebuild.
10. Copy the Random_Number.vi LabVIEW file which is present in LabVIEW file folder to "C:\Program Files (x86) \National Instruments\LabVIEW 2020\examples

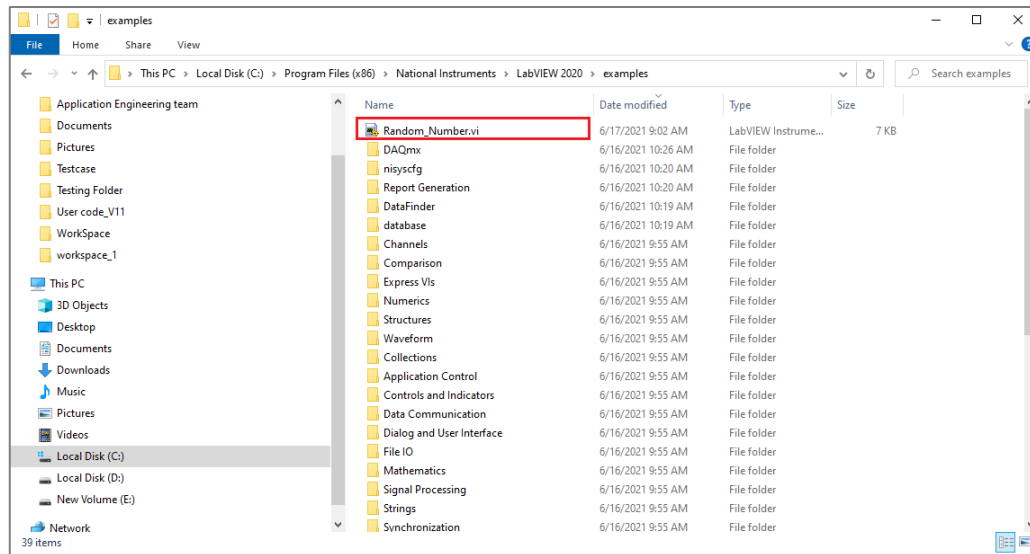


Figure 9: Copy Random_Number.vi in National Instruments\LabVIEW 2020\examples

11. To open this LabVIEW example, go to Open Simulation and click on the Example that is present under the list of experiments as shown below.

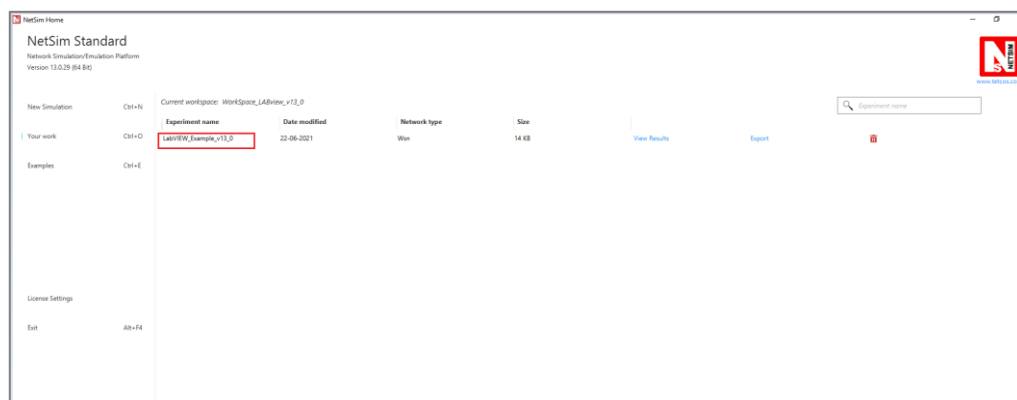


Figure 10: LabVIEW saved example in your work.

12. You can be able to observe the sample with WSN sensors and WSN sink connected with Adhoc link.

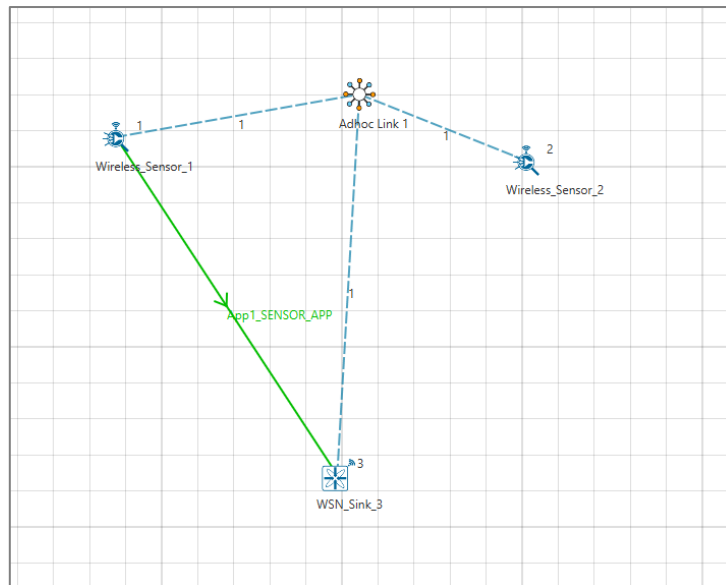


Figure 11 : WSN Scenario

- Click on Run button and simulate the scenario. In Netsim Console it will ask the user to initialize the LabVIEW, click on Enter.

```

C:\Users\Navya\Desktop\WorkSpace_LABview_v13_0\bin\bin_x64\NetSimCore.exe
Config file reading complete
License re-validation complete
Protocol binaries loaded
Executing command --- DEL "C:\Users\Navya\AppData\Local\Temp\NetSim\std13.0.29_x64\*.pcap"
Could Not Find C:\Users\Navya\AppData\Local\Temp\NetSim\std13.0.29_x64\*.pcap
Emulation is disabled
Stack variables initialized
Could Not Find C:\Users\Navya\AppData\Local\Temp\NetSim\std13.0.29_x64\Plot_*
Metrics variables initialized

initializing Labview
Press any key to continue...
  
```

Figure 12: Initializing LabVIEW to Interface with NetSim.

- You can be able to observe that Random number generated in LabVIEW is captured in NetSim while Run time interaction.

```

C:\Windows\System32\cmd.exe - NetSimCore.exe -apppath "C:\Users\Navya\Desktop\WorkSpace_LABview_v13_0\bin\bin_x64" -logpath "C:\Users\Nav...
Value returned from LABVIEW: 9.280742
Value returned from LABVIEW: 4.028747
76 % is completed... Simulation Time=380000.000 ms Event Id=7475
Value returned from LABVIEW: 0.687332
Value returned from LABVIEW: 3.693726
80 % is completed... Simulation Time=400000.000 ms Event Id=7877
Value returned from LABVIEW: 7.017087
Value returned from LABVIEW: 8.731716
Value returned from LABVIEW: 3.303725
86 % is completed... Simulation Time=430000.000 ms Event Id=8480
Value returned from LABVIEW: 7.522320
Value returned from LABVIEW: 0.150312
90 % is completed... Simulation Time=450000.000 ms Event Id=8882
Value returned from LABVIEW: 2.071169
Value returned from LABVIEW: 6.556341
Value returned from LABVIEW: 4.570911
96 % is completed... Simulation Time=480000.000 ms Event Id=9485
Value returned from LABVIEW: 9.363423
Value returned from LABVIEW: 1.119209
100 % is completed... Simulation Time = 500000.000 ms Event Id=0
Total time taken (wall clock) = 3906 ms
Total events processed = 10084
  
```

Figure 13 : Interaction of NetSim with LabVIEW