Oscillation Detection (HPP App) User Guide

May 2020

Neuralynx, Inc.

105 Commercial Drive Bozeman, MT 59715
406-585-4542
www.neuralynx.com
support@neuralynx.com

1 Introduction

The HPP Applications project will provide packaged applications that target the Neuralynx Hardware Processing Platform (HPP) for common real-time feedback loop and analysis processes using the Digital Lynx SX acquisition system.

The Oscillation Detection application is a closed-loop application that will be running on the HPP, which allows the detection of oscillation power within the sharp wave ripple (150 to 250 Hz) or theta band (4 to 12 Hz) and triggers a stimulus via a TTL output pulse when the oscillation has been confirmed. An oscillation is confirmed when the oscillation power is above a user defined threshold for a user set amount of time.

This "User Guide" will walk through the process of programming the oscillation detection application to the HPP over JTAG and getting started with the oscillation detection application interface.

2 Installing and Licensing Xilinx Software

For this project, we will be using the Xilinx SDK 2014.1 tool to program the oscillation detection application to the HPP processor over the JTAG connection. Refer to Section 2 of the HPP Getting Starting Guide document, for instructions on how to download, install and license the Xilinx SDK tools.

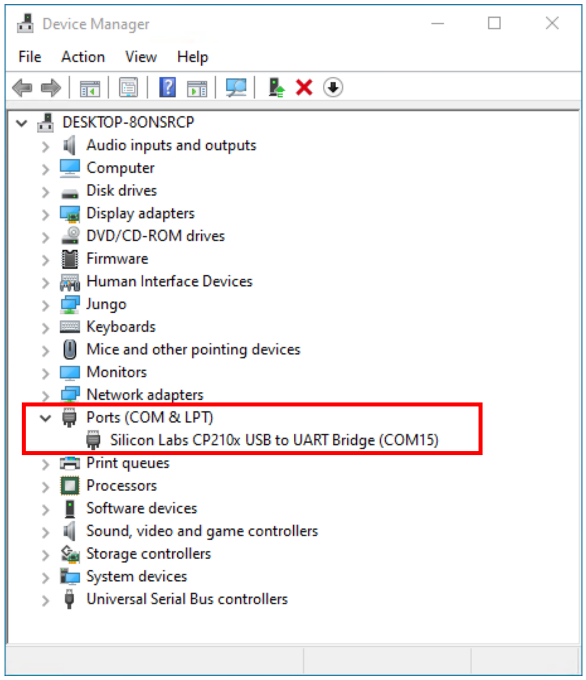
3 Setting up HPP hardware connections

The application uses a serial command-line interface (CLI) to interact with the program running on the HPP. The UART Bridge Virtual COM Port (VCP) driver must be installed on your PC.

Download and install the latest Silicon Labs CP210x Windows VCP driver from
https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Attach a USB cable from the application PC to Digital Lynx SX port "HPP Terminal"
Attach a USB cable from the application PC to Digital Lynx SX port "HPP JTAG".

In Windows Device Manager, make a note of the COM port number “Silicon Labs CP210x USB to UART Bridge” connected to your HPP USB Serial Terminal. The baud rate is 115200.



HPP connected to Silicon Labs CP210x USB to UART Bridge COM port.

4 Installing Matlab Runtime Prerequisite

The HPP Applications project will also require the MATLAB Runtime version 9.3 (R2017b) to be installed on your PC in order to be able to run the oscillation detection application control interface.

Download and install the 64-bit Windows version of the Matlab Runtime for R2017b from the following link on the Mathworks website.
https://www.mathworks.com/products/compiler/matlab-runtime.html

5 Neuralynx Cheetah Acquisition Software

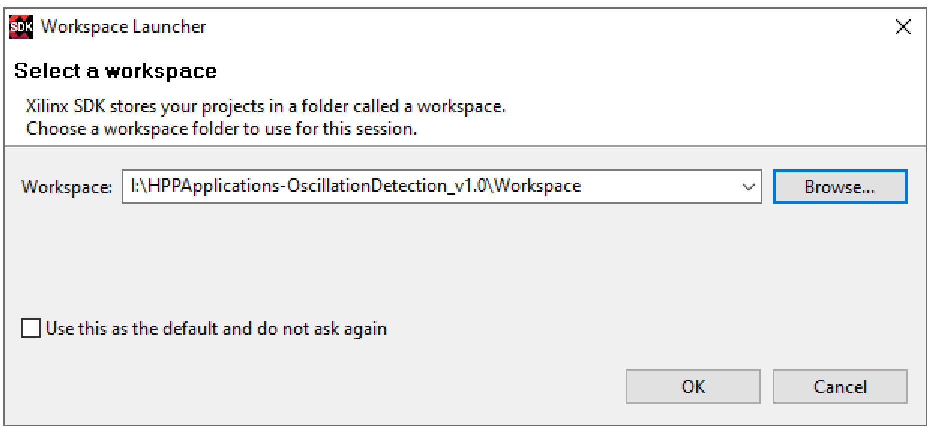
The HPP Applications project will require the Neuralynx Cheetah 6.0 or greater data acquisition software (version compatible with installed DLSX motherboard firmware on your system). Cheetah reference can be obtained from the Neuralynx website.
Start Cheetah acquisition in order to allow the HPP to be able to obtain input acquisition data from the Digital Lynx SX system.

6 Programming the Application to the HPP Hardware

Download the HPPApplications-OscillationDetection_v1.0.zip project from the Neuralynx website.

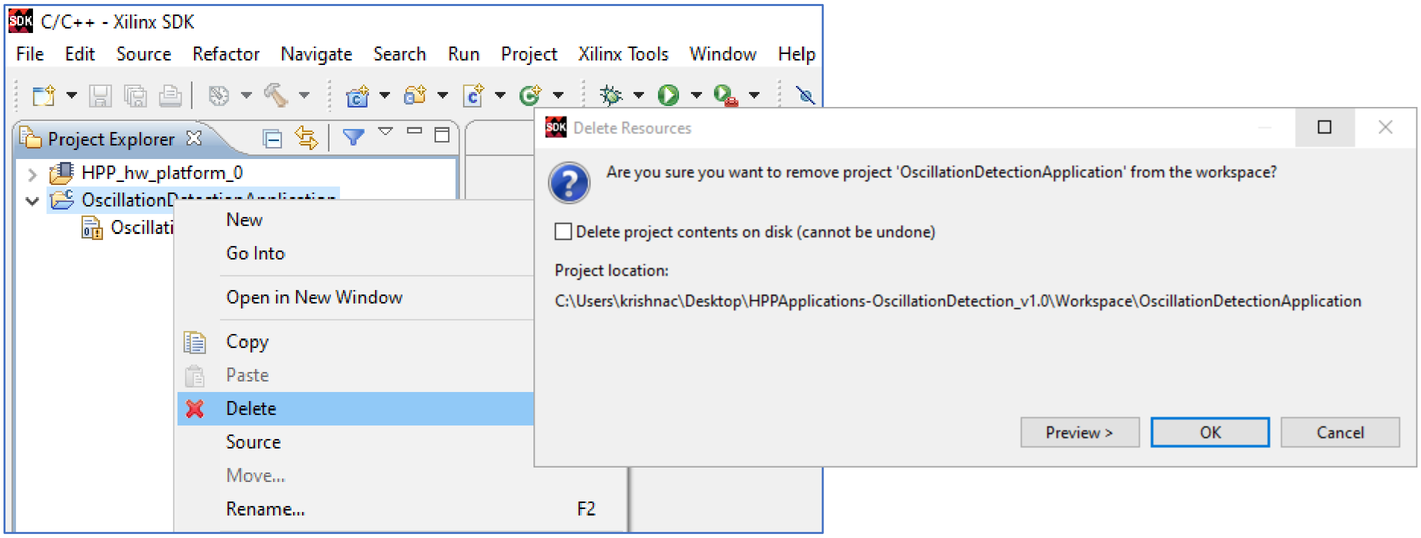
Extract the compressed project to a folder on your applications PC.

Open the Xilinx SDK 2014.1 software.



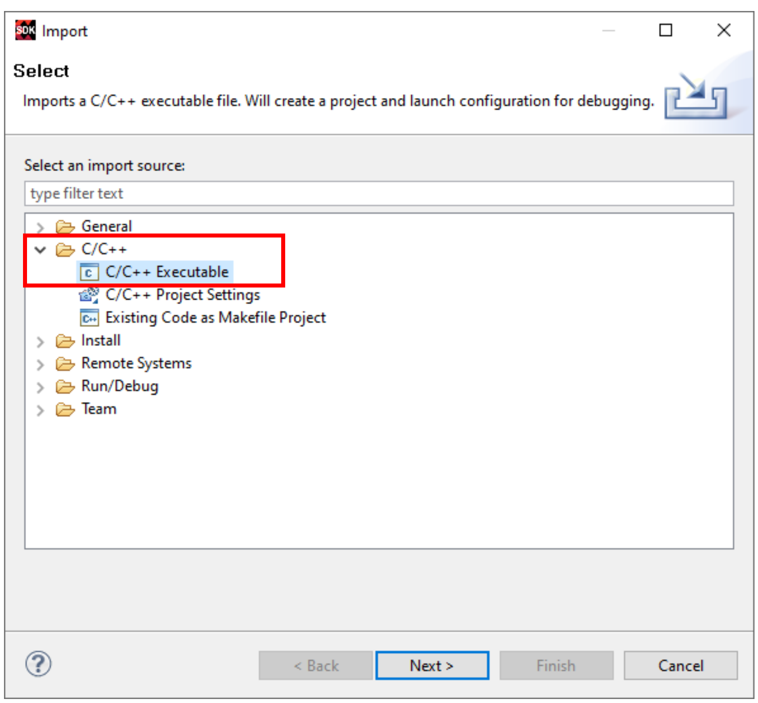
Browse to the workspace directory of the extracted project located at <your_proj_dir>\HPPApplications-OscillationDetection_v1.0\Workspace\ and click “OK” to continue.

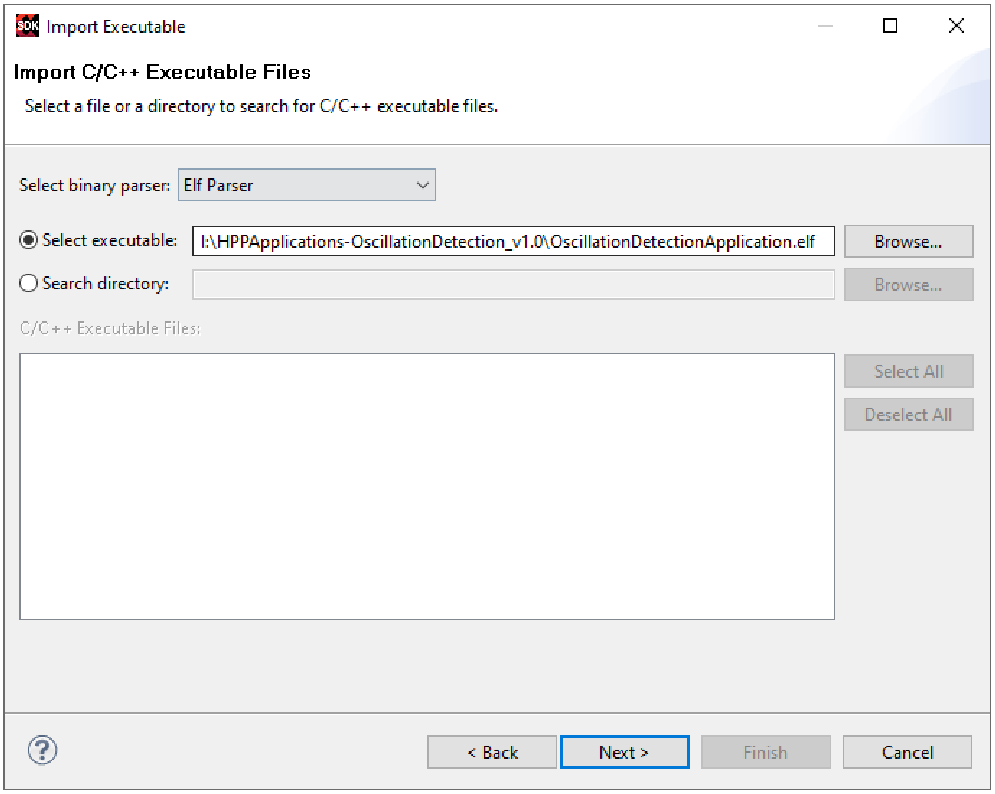
In the SDK Project Explorer on the left, right-click on “OscillationDectionApplication” folder and select “Delete” from the menu.



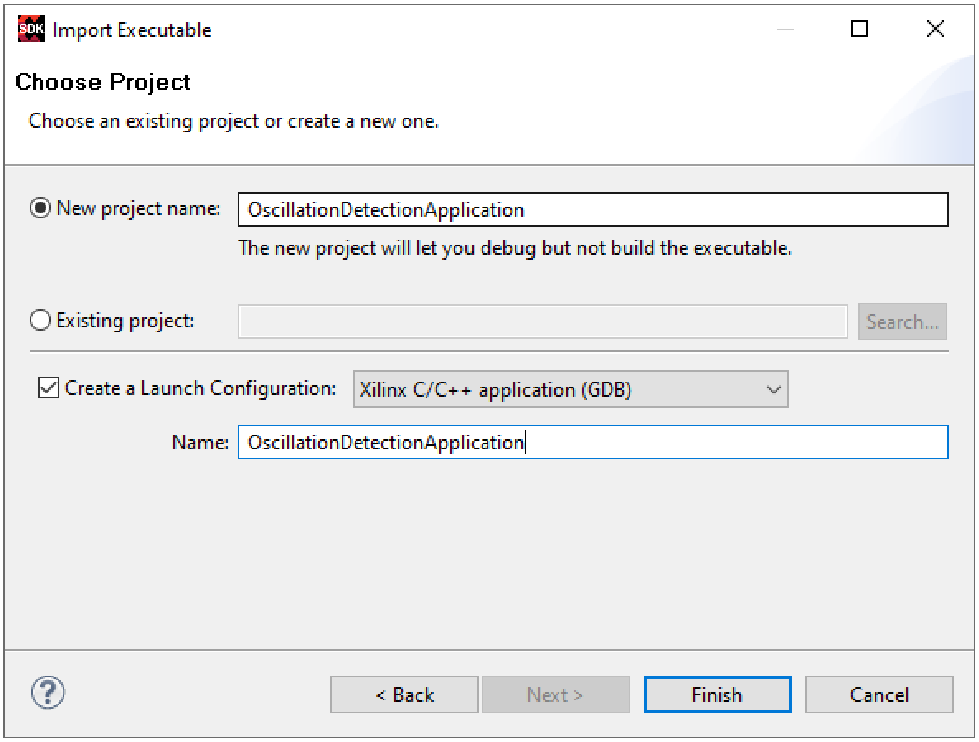
Click “OK” to confirm the deletion.

Select menu option “File -> Import...” or right-click in Project Explorer and select “Import ...”, then choose “C/C++ Executable” under “C\C++” and click “Next” to continue.



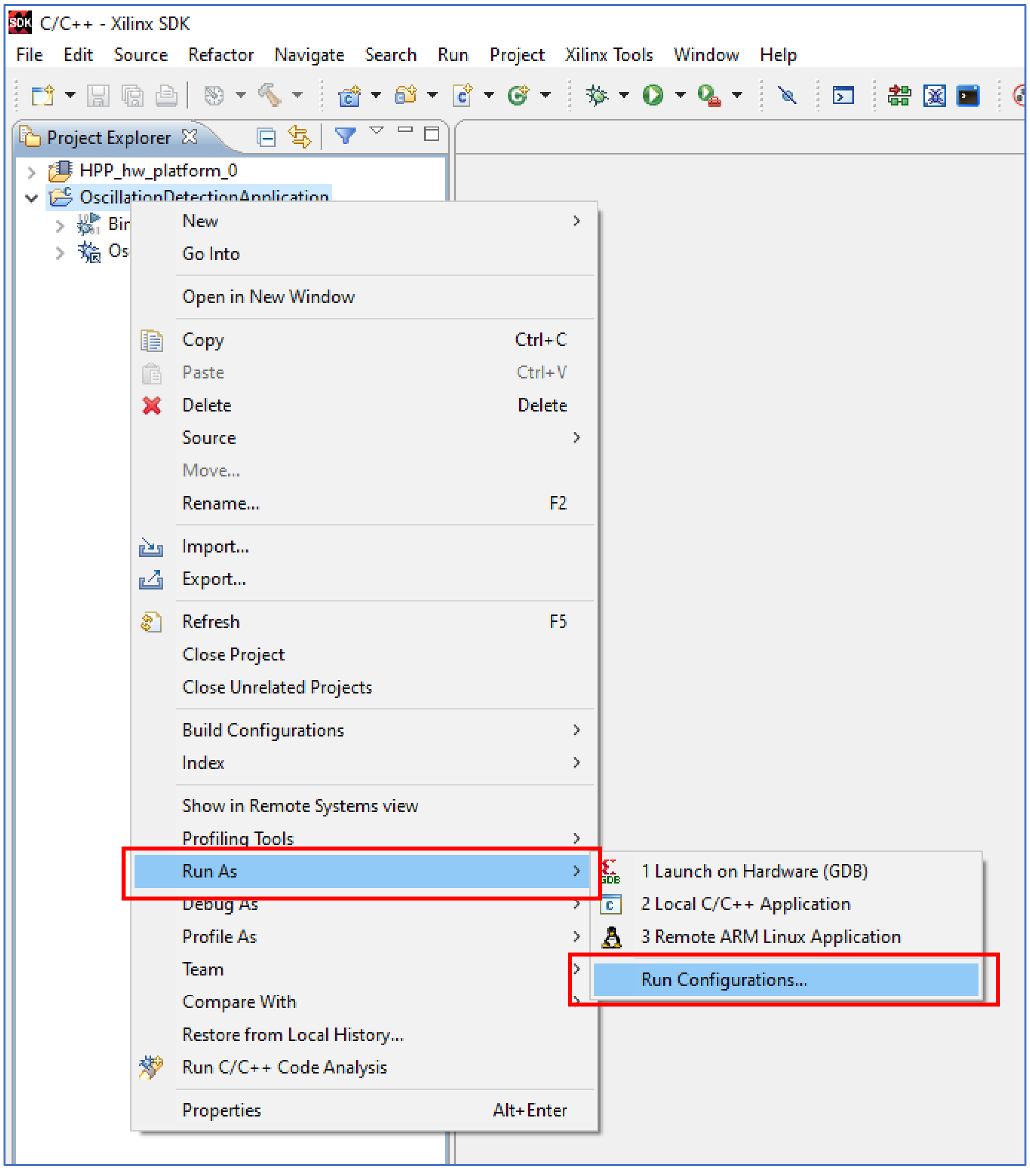


Ensure that “Select binary parser” is set to the “Elf Parser” option.
Click the “Browse” button and select the provided ELF file
<your_proj_dir>\HPPApplications-OscillationDetection_v1.0\OscillationDetectionApplication.elf
and click “Next” to continue.

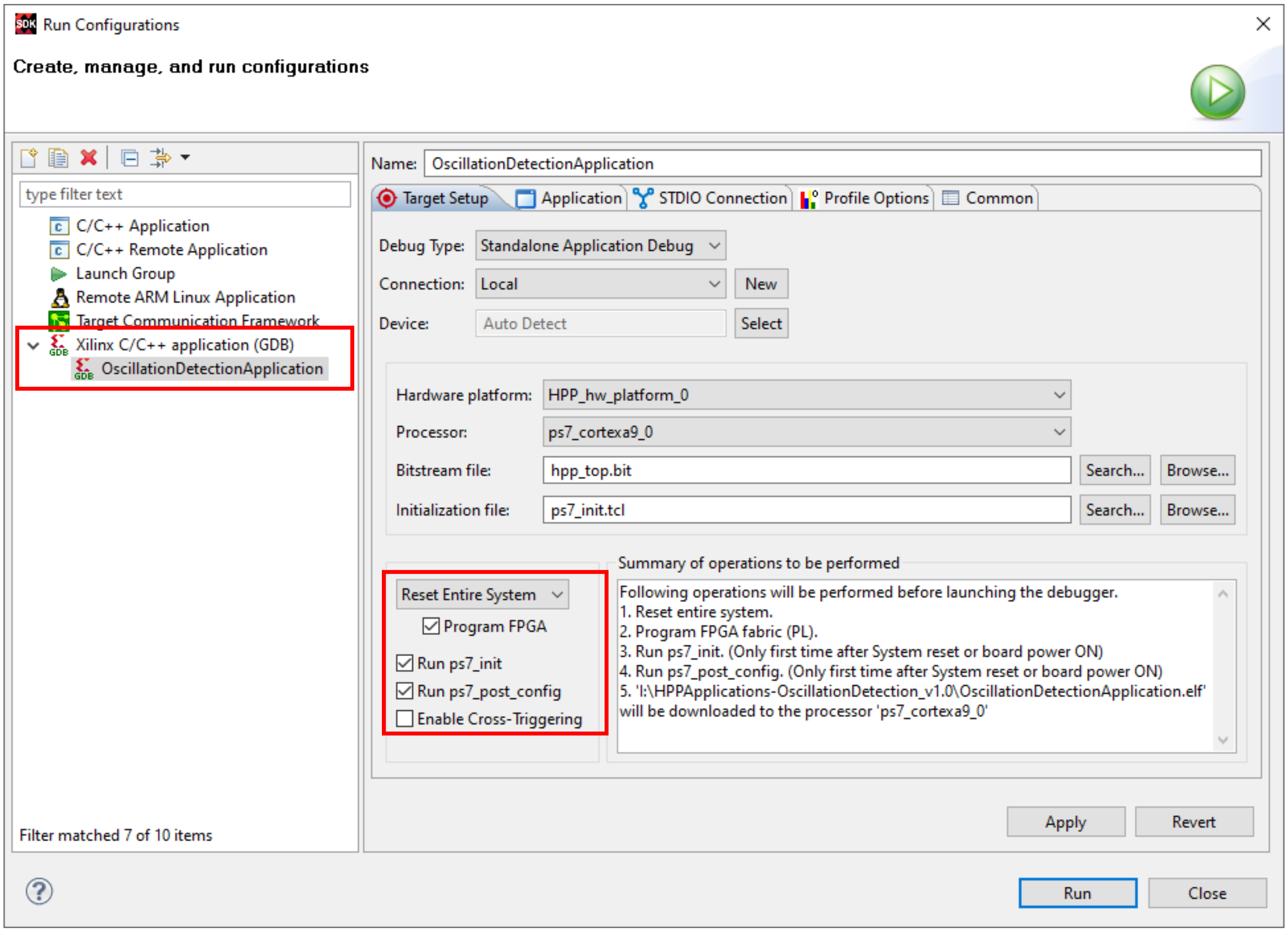


In the “New project name” field, enter “OscillationDetectionApplication” as the new project name.

Ensure “Create a Launch Configuration” is checked, and select “Xilinx C/C++ application (GDB)” option. Enter “OscillationDetectionApplication” in the Launch Configuration “Name” field.



In the SDK Project Explorer on the left, right-click on “OscillationDectionApplication” folder and select “Run As -> Run Configurations…” from the menu.



In the Run Configurations window select "OscillationDectionApplication" under Xilinx C/C++ application (GDB).

Next to the "Bitstream file" field click on "Search…" and select *hpp_top.bit* and click "OK" to continue. This is the bitstream file that will be programmed to the Xilinx FPGA.

Next to the "Initialization file" field click on "Search…" and select *ps7_init.tcl* and click "OK" to continue. This is the initialization file for setting up the Xilinx processor.

Select "Reset Entire System" and check the "Program FPGA", "Run ps7_init" and "Run ps7_post_config" options as shown in the figure above.

Click the "Apply" button to save the Run Configuration settings and "Run" to download and program HPP Xilinx processor and FPGA hardware.

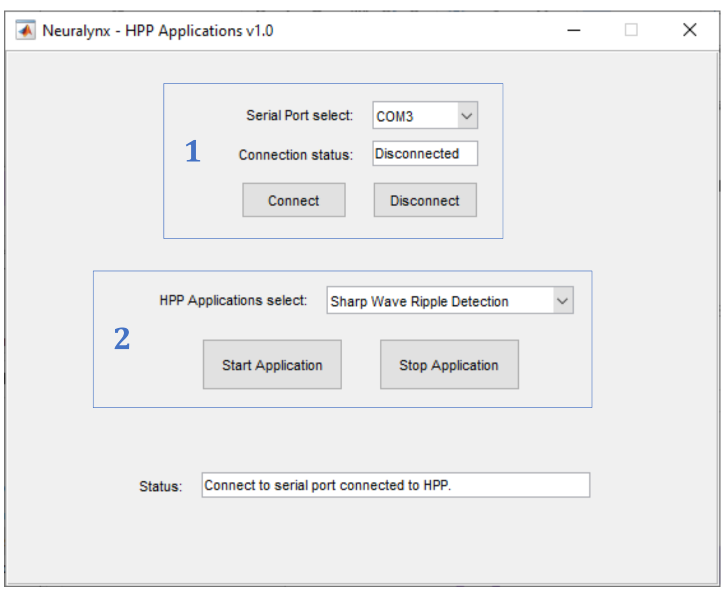
7 Running the Application Interface from the PC.

The oscillation detection application interacts with the oscillation detection algorithm programmed onto the HPP through a serial interface. The application provides the user access to four independent modules to be able to detect oscillations from the input acquisition channels of the Digital Lynx SX. Individual parameters can be set to adjust the oscillation detection sensitivity for each module. Each detection module triggers a TTL bit output response on the Digital Lynx SX when an oscillation has been confirmed.

The diagram below outlines the flow implemented for the oscillation detection algorithm.

A/D channel in → bandpass filter → calculate oscillation power → detect oscillation → output TTL

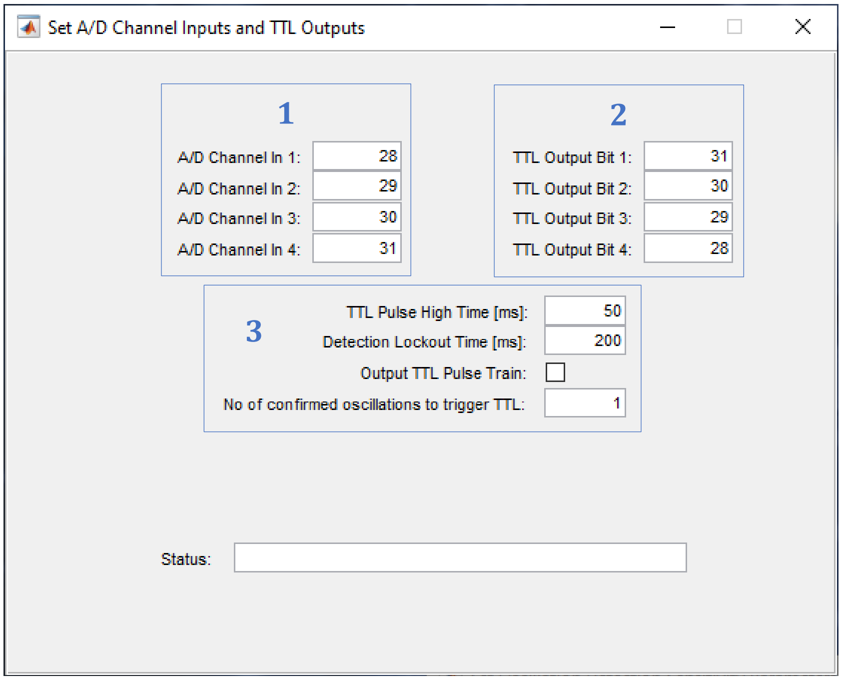
Double-click on <your_proj_dir>\HPPApplications-OscillationDetection_v1.0**HPPApplications-OscillationDetection.exe** to run the executable that starts the oscillation application interface program. Start Acquisition in the Cheetah Data Acquisition software running on your PC to allow the HPP to be able to obtain input acquisition data from the Digital Lynx SX system.



1. Select the COM Port connected to the HPP from the "Serial Port select" drop-down menu, and click the "Connect" button to open to the serial port connection. Refer to Section 3 of this User Guide to determine which COM port is connected to your HPP through Windows Device Manager.

2. Select either "Sharp Wave Ripple Detection" or "Theta Oscillation Detection" from the "HPP Applications select" drop-down, and click the "Start Application" button to begin running one of the provided oscillation detection applications. The "Stop Application" button stops the application running on the hardware.

After waiting a few seconds for initialization of parameters between the application and the hardware, the following interface windows will appear to allow the user to control and interact with the oscillation detection application.



Set A/D Channel Inputs and TTL Outputs interface window

1. Select A/D Channel Inputs

The “A/D Channel In {1:4}” fields allow the user to choose the Digital Lynx SX acquisition channel inputs (0 through 63) to each of the four (4) oscillation detection modules {1:4}. The input channel data is downsampled to 2kHz, and routed through the selected detection band filter to the oscillation detection module.

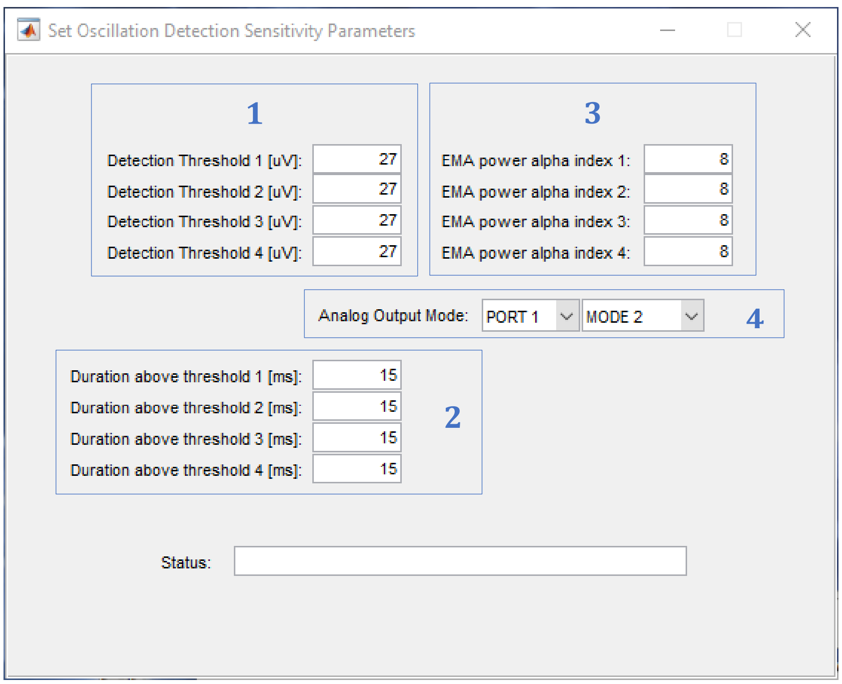
2. Select TTL Output Bits

The “TTL Output Bit {1:4}” fields allow the user to set the TTL output response bit (0 through 31) that is triggered on the Digital Lynx SX when an oscillation has been confirmed for each of the four (4) oscillation detection modules {1:4}.

3. Set TTL Output Response Parameters

The “TTL Pulse High Time [ms]” field allows the user to set the time in milliseconds that the TTL pulse output remains high when triggered.
The “Detection Lockout Time [ms]” field allows the user to set the time in milliseconds that the TTL pulse output remains low. Oscillation detection is blocked during this time.

The “Output TTL Pulse Train” checkbox allows the user to be able to enable a repeating TTL pulse output while an oscillation is being detected (oscillation power above threshold). The TTL output pulse high and low times are determined by the “TTL Pulse High Time [ms]” and “Detection Lockout Time [ms]” (pulse low time) values set in the previous fields. When disabled (checkbox not checked), only a single pulse is output and the TTL remains low until the next confirmed oscillation has been triggered.
The “No of confirmed oscillations to trigger TTL” field allows the user to set the number of oscillations that needs to be confirmed before triggering the TTL output. A value of “1” triggers the TTL output for every confirmed oscillation, while a value of “2” triggers the TTL output every other confirmed oscillation. A value of “0” disables any TTL output triggering.



Set Oscillation Detection Sensitivity Parameters interface window

1. Set Oscillation Detection Thresholds

The “Detection Threshold {1:4} [uV]” fields allow the user to set the oscillation detection threshold in microvolts that the oscillation power must be above, to confirm an oscillation has been detected for each of the four (4) oscillation detection modules {1:4}.

2. Set Oscillation Duration above Threshold Time

The “Duration above threshold {1:4}” fields allow the user to set the duration of time in milliseconds that the oscillation power must be above the threshold, to confirm an oscillation has been detected for each of the four (4) detection modules {1:4}.

3. Set Oscillation Power Sensitivity Parameter

The oscillation power is calculated using an integrated exponential moving averaging (EMA) method applied the bandpass filtered signal.

$$curr\_power = prev\_power + \alpha lpha \times [abs(curr\_sample) - prev\_power)],$$
$$\text{where } alpha = 1/(2\hat{}N), \quad 0 >= N >= 31$$

The index *N* is referred to as the "EMA power alpha index" in this application.

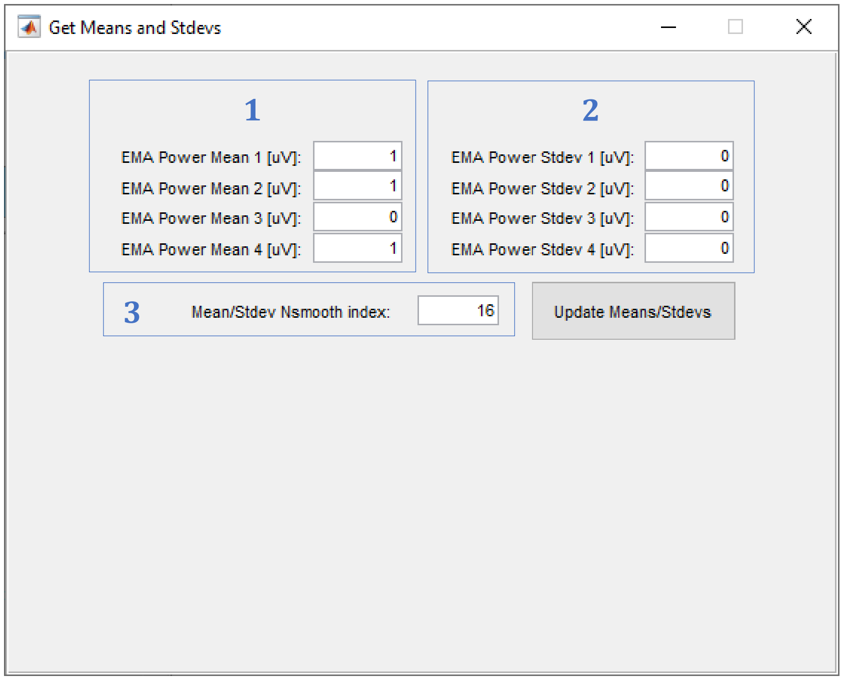
The "EMA power alpha index {1:4}" fields allow the user to set the value of N, the EMA power alpha index for each of the four (4) detection modules {1:4}.
The larger *N* is made, the smaller alpha becomes and the smoother the power signal over time. As *N* is made smaller, the *alpha* value approaches 1, and the calculated power approaches the oscillation signal. Making *N* too large, reduces the overall oscillation power signal. Typical values for the EMA power alpha index range from 8 to 16.
A combination of setting the oscillation power smoothing index, and adjusting the threshold level provides the user with the ability to adjust the sensitivity of the oscillation detection algorithm.

4. Select Analog Output Port and Modes

In order to be able to display waveform signals in real-time while the application is running, the analog outputs on the Digital Lynx SX front panel have been integrated into the application. A single stereo audio to dual BNC cable and a 2-channel oscilloscope, or two stereo audio to dual BNC cables and a 4-channel oscilloscope can be used to view waveforms.
The "Analog Output Mode" Port {1:4} drop-down, allows the user to select which of the four detection modules {1:4} to output the corresponding waveform signals.
The output waveforms have been grouped into logical pairs (MODES) to be output through each of the two stereo analog output ports.
The table below shows the defined modes, ranges and the corresponding waveform outputs.

	Input Range (Scale)	AN1	AN2	AN3	AN4
MODE 0:	2048 uV (x64)	Power Mean	EMA Power	Power Std Dev	EMA Power
MODE 1:	2048 uV (x64)	Filtered Output	A/D Ch Input	Filtered Output	EMA Power
MODE 2:	2048 uV (x64)	Filtered Output	EMA Power	Threshold	EMA Power
MODE 3:	132 mV (x1)	Filtered Output	A/D Ch Input	Filtered Output	EMA Power
MODE 4:	132 mV (x1)	Filtered Output	EMA Power	Threshold	EMA Power
MODE 5:	132 mV (x1)	Power Mean	EMA Power	Power Std Dev	EMA Power
MODE 6:	2048 uV (x64)	Filtered Output	A/D Ch Input	Threshold	EMA Power

AN1 and AN2 refers to the left and right stereo outputs of the first output jack, and AN3 and AN4 refers to the left and right stereo outputs of the second output jack on Digital Lynx front panel.
At full scale (x1), the full 131,072uV input range of the acquisition system is mapped to +/-1V analog output. In (x64) scale, the input acquisition range +/- 2048uV is mapped to the +/-1V analog output range.



Get Mean and Standard Deviations interface window

1. Get Oscillation Power Mean

The "EMA Power Mean {1:4} [uV]" fields return the running mean of the oscillation EMA power to the user in microvolts for each detection module {1:4} when the "Update Means/Stdevs" button is pressed".

2. Get Oscillation Power Mean

The "EMA Power Mean {1:4} [uV]" fields return the running standard deviation of the oscillation EMA power to the user in microvolts for each detection module {1:4} when the "Update Means/Stdevs" button is pressed".

3. Set Number of Samples over which to Calculate Mean and Standard Deviation.

The "Mean/Stdev Nsmooth index" value allows the user to be able to set the number of samples over which the running mean and standard deviation is calculated for the oscillation power.

No of samples = 2^*Nsmooth*

For a downsampled rate of 2kHz, a *Nsmooth* value of 16, calculates the mean and stdev over ~30s worth of oscillation sample values.

The mean and standard deviation values can be useful for determining what value to set the threshold for example: threshold = mean + *k**stdev, where k is user determined constant.

8 Testing the Oscillation Detection Application.

Two sample audio wavefiles, “test_swr_detection.wav” and “test_theta_detection.wav”, that have been converted from CSC acquisition recordings, have been provided to allow the user to be to test the oscillation detection application.

The wavefiles can be played from a PC or media device with audio output. Output at 100% volume with no filters applied from the audio software will result in an output signal range of +/-1000uV. The audio signal is fed through a stereo to BNC cable to an input signal mouse connected to the Digital Lynx acquisition system.

For more information or questions, please contact:

HPP_Support@neuralynx.com / 406-585-4542