neaSNOM Microscope

SDK

see the nanoworld
neaspec

Overview

1. **General design SDK / DLL**

2. **Setup access to automatically updated dll of neaSNOM**

3. **LabVIEW**
    1. Connect to neaSNOM SDK with LabVIEW
    2. LabVIEW example: PsHet scan routine

1. **Python**
    1. Installation Python and pythonnet
    2. Python example: AFM scan routine

2. **Documentation**

1 General design neaSNOM SDK / DLL

The general design is depicted beneath, how to access any functionality of the **neaSNOM:**

1. **Connect to SDK**
2. **Create Microscope with the function ‚Connect()' of the SDK**
   => You have now access to all functionallities of the AFM e.g. AutoApproach(), GotoTipPosition()
3. **You can now also connect to a Stage, Channel, or Scan:**
   - **Stage: Allows you to move any stage in the Micsoscope, e.g. Parabolic mirror, sample**
   - **Channel: Allows you read out the current Value of any channel with CurrentValue**
   - **Scan: Allows you to create any Scan already implemented in the neaSNOM and you can set the parameter for the scan, e.g. AfmScan, WhitelightScan**
4. **The function Start():**
   1. **Starts the scan**
   2. **Creates one or all images of the channels available in this scan.**

1.

Nea.Client.Hardware.SDK.Connection:		
Properties	Methods A	Methods B
Logfile	Disconnect()	Connect()
etc.	Discopes()	
	etc.	

2.

Microscope		
Properties	Methods A	Methods B
ClientVersion	ActivateLeftDetectorForMidInfrared()	ActivateBottomParabolicMirrorStage() ActivateSampleStage() etc.
IsInContact	AutoApproach()	
TipPositionX	GotoInterferometerPosition()	
etc.	GotoTipPosition()	GetChannel() (e.g M1a, O0A, Z
	RegulatorOff()	
	etc.	PrepareAfmScan() PrepareWhitelightScan() etc.

3.

Stage	
Properties	Methods A
HasPositionSensors	FindAbsoluteMotorPosition()
PositionX	MoveX()
etc.	GotoRelativePosition()
	etc.

Channel	
Properties	Methods A
CurrentValue	

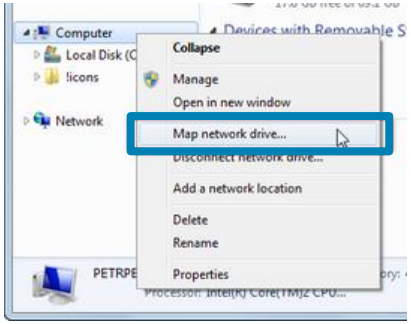
Nea.Client.Hardware.SDK.Scan (AFM,Fourier,PsHet,ApproachCurve)		
Properties	Methods A	Methods B
IsStarted	SetSamplingTime()	Start()
IsSuspended	SetScanArea()	
etc.	SetCenter()	
	etc.	

4.

Nea.Client.Hardware.SDK.Images (AFM,Fourier,PsHetS,ApproachCurve)	
Properties	Methods
	GetChannel()

All SDK functionalities are documented under the following location: \\nea-server\updates\SDK\doc\index.html

2 Setup access to automatically updated dll of neaSNOM



Map Network Drive:

1) Click with the right mouse button on Computer in the Windows Explorer

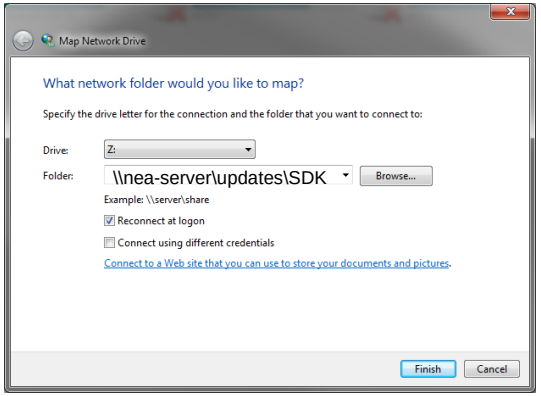
2) Select Map network drive…

3) Name the Drive N: (any letter possible)

4) Connect to the following folder:

\\nea-server\updates\SDK

5) The SDK (DLL) in this loaction will be automatically updated with each **neaSCAN** update



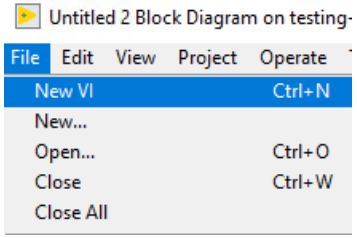
3.1 Connect to neaSNOM with LabVIEW

neaspec LabView SDK guide:

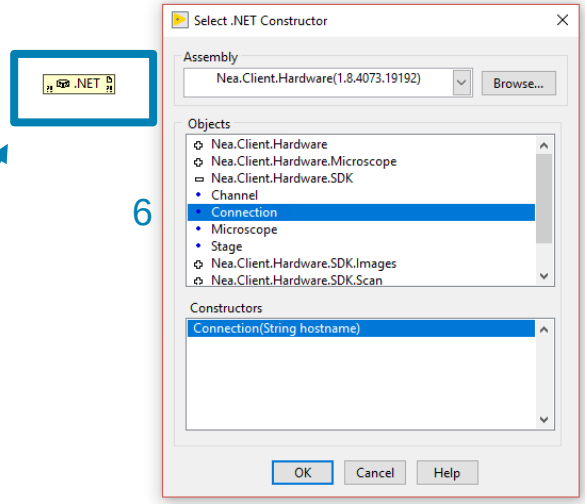
1) Start your LabView Application
2) Open a new project or VI
3) Open the *Block Diagram View* by pressing 'Ctrl + E'
4) Right click on the white background and select:

Connectivity ⟹ *.NET* ⟹ *Constructor Node*

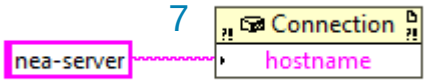
5) Place the constructor node in the block diagram
6) Select *Nea.Client.SDK* ⟹ *Connection*
7) To establish a connection type in the hostname via a *String Constant* and call it *nea-server*
8) Now open the .NET element *Invoke Node*
9) Wire it with the *Connection Node* and select *Connect()* as Method
10) The VI should look like this for now and the connection to the **neaSNOM** is established

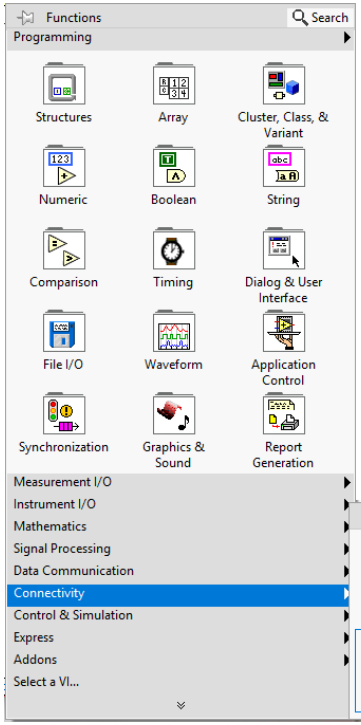


2

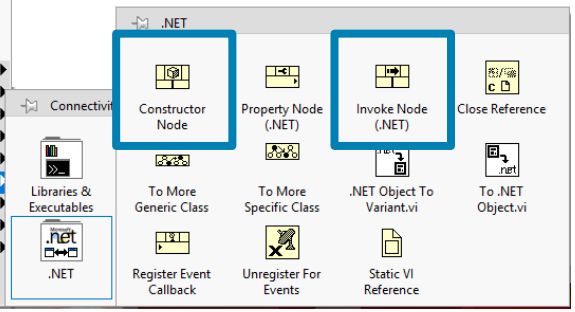


6

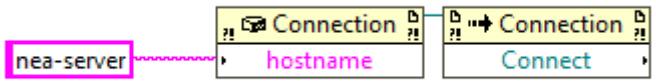




3



8

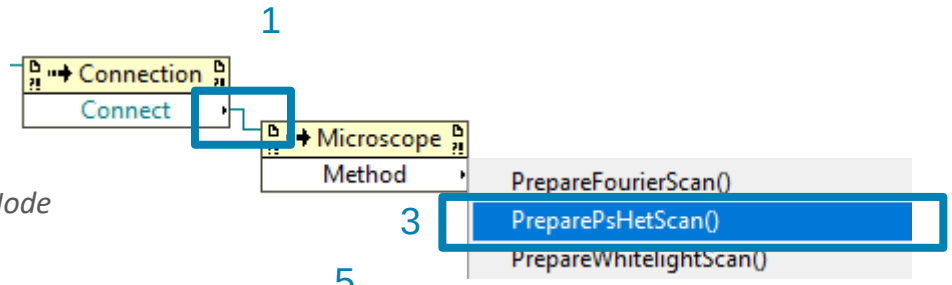


9

3.2 LabVIEW Example: PsHet Scan Routine

After setting up the connection to the microscope all acquired Methods for standard Scan Routines and DAQ of the **neaSNOM** are accessible

Example PsHet Scan Routine:



1) Use another *Invoke Node* and wire it directly to *Connect* in the *Connection Node*

2) Choose the desired *Method* in the *Microscope* Node

3) In this example *PreparePsHetScan()* is selected as Method

4) Wire an Invoke Node to *PreparePsHetScan()*

5) This gives access to various scan parameters:

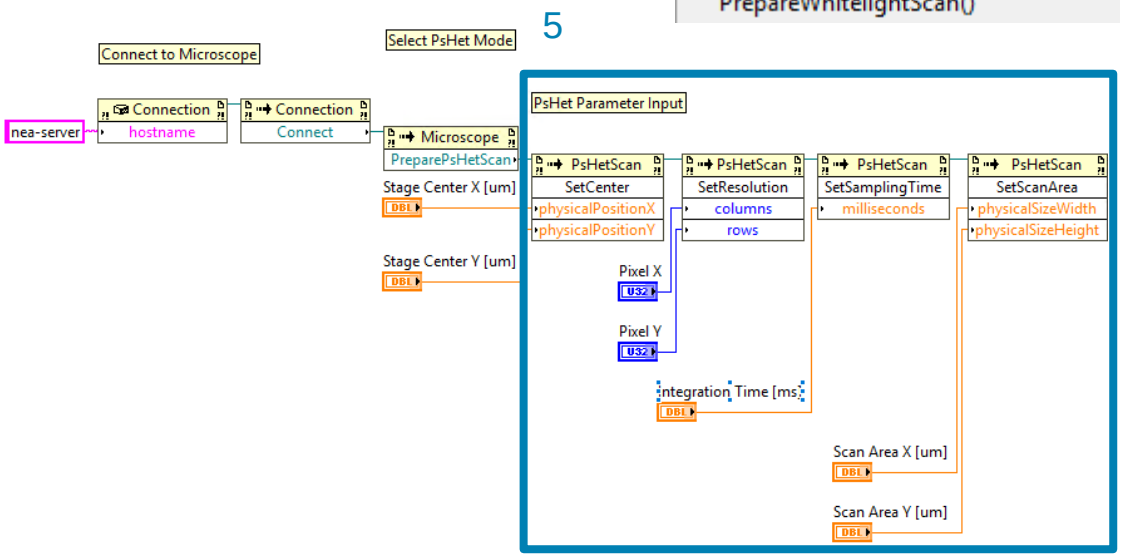
   Set Center

   Set Resolution

   Set Sampling Time

   Set Scan Area

   … (further Settings in Documentation)



6) Connect all the parameters necessary for the scan in series

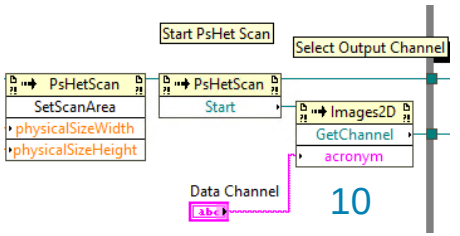
7) Wire the last *PsHetScan* Node with another Invoke Node and select the *Start* method to initiate the scan

8) Invoke the *Images 2D* Node to get access to individual channels in the *GetChannel* method

9) The *String Input* wired to *acronym* gives access to an user set Data Channel

10) For more than one channel, wire *GetChannel* methods in parallel

11) The path splits in two branches, data handling and scan status



3.2 LabVIEW Example: PsHet Scan Routine

Data Handling:

1) Within the *While Loop* the data flow for each pixel is saved to a customizable file location

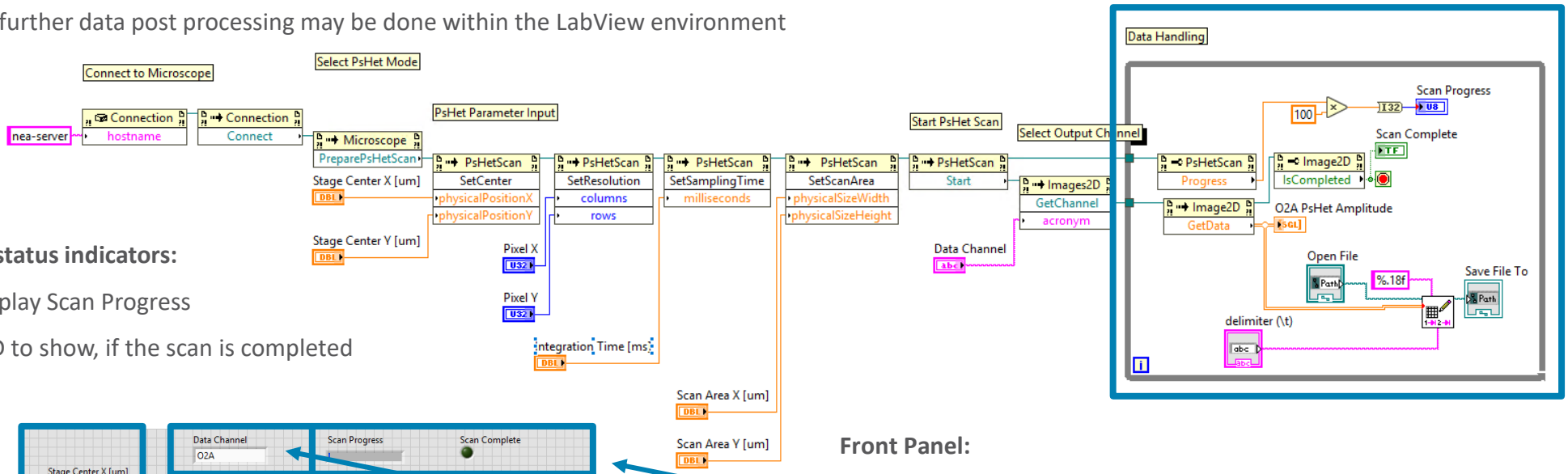
2) In parallel the Image (*Image2D)* of the selected *Data Channel* is displayed in the *Front Panel*

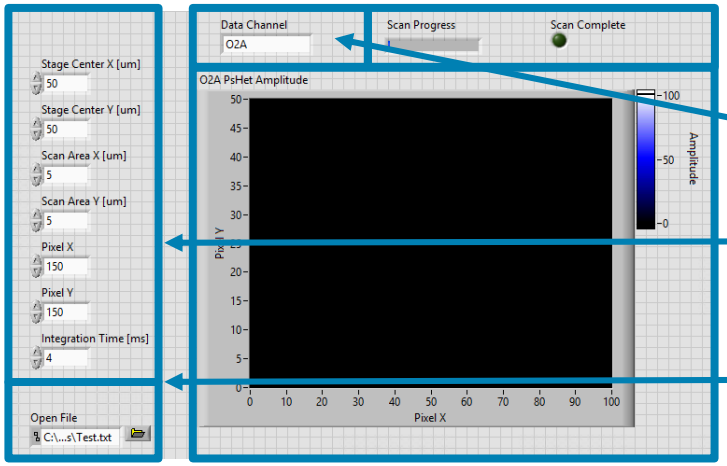
3) All further data post processing may be done within the LabView environment

Scan status indicators:

4) Display Scan Progress

5) LED to show, if the scan is completed





Front Panel:

Scan status

Data Channel

Scan Parameters

Display for Scanned Data

Save File

4.1 Installation Python and pythonnet

1. It is possible to use Python 2.7 and 3.7, but we recommend 3.7

2. Recommended download locations:
   - Anaconda includes already most important packages: https://www.anaconda.com/distribution/#download-section
   - Basic python (only basic packages installed): https://www.python.org/downloads/

3. After installations of python, install the packacke pythonnet:
   "**pip install pythonnet"**

   pythonnet allows python to handle the C# dll of the neaSNOM

4. Recommended packages to install (used in examples):
   - numpy
   - matplolib
   - scipy
   - statistics

4.2 Python example: AFM scan routine

```
import sys                                                  # sys class for importing folder
import time                                                 # time class for short delay
import clr                                                  # pythonnet for reading C# SDK

#Basic steps to connect to Microscope
assembly_folder = '//nea-server/updates/SDK'        #Import all DLLs in the folder or 'N:/updates/SDK' to mapped drive
sys.path.append(assembly_folder)                            # Import all DLLs in the folder
clr.AddReference('Nea.Client.Hardware')                     # Load the main DLL
import Nea.Client.Hardware.SDK as neaSDK                    # Import the DLL as element neaSDK
neaClient = neaSDK.Connection('nea-server')                 # Open up connection to microscope called neaClient
neaMic = neaClient.Connect()                                # Define the Microscope neaMIC
time.sleep(0.1)                                             # Short delay makes things working fine

#Basic infos of Client and Server and print them to the console of python
ClVersion = neaMic.ClientVersion                            # get Client Version
print("Client Version: " + ClVersion)
SeVersion = neaMic.ServerVersion                            # get Server Version
print("Server Version: " + SeVersion)

#example to start auto approach after being connected
neaMic.CancelCurrentProcedure()                             # Cancel any running procedure (not obligatory)
neaMic.RegulatorOff()                                       # Retract sample (not obligatory)
if not neaMic.IsInContact:                                  # Check if system in contact (not obligatory)
    neaMic.AutoApproach(setpoint)                           # Start auto approach with chosen setpoint

#go out of contact and disconnect from the microscope
neaMic.RegulatorOff()                                       # Retract sample
neaClient.Disconnect()                                      # Disconnect from SDK
plt.show()
```

This file neaSNOM_AFMscan.py is an example for a python script how to perform, show and save an AFM Scan with the SDK containing the example above!

5 Documentation

Overview of the whole SDK functionality is documented under the following location:

\\nea-server\updates\SDK\doc\index.html

Nea.Client.Hardware

Test Bla und so.

Nea.Client.Hardware.SDK Namespace

Type	Description
Connection	Provides a connection to remote neaSNOM controller.
Microscope	Represents a remote micrscope.
Stage	Represents a stage with motors on three axes.

Nea.Client.Hardware.SDK.Images Namespace

Type	Description
Image2D	Represents a 2D image.
Image3D	Represents a 3D image.
Images<TScan,TImage>	Represents a collection of channels with scanned data.
Images2D	Represents a collection of 2D images.
Images3D	Represents a collection of 3D images.
Interferogram	Represents Interferogramms.
Interferograms	Represents a collection of interferograms.
RawData	Represents a channel with scanned data.

Nea.Client.Hardware.SDK.Scan Namespace

.

Type	Description
AfmScan	Represents a scan with AFM parameters.
BasicScan	Represents a scan with basic parameters.
CfmScan	Represents a scan with CFM parameters.
FourierScan	Represents a scan with Fourier parameters.
PsHetScan	Represents a scan with PsHet parameters.
Scan2D	Represents a scan with 2D parameters.
Scan3D	Represents a scan with 3D parameters.
SerpentableScan2D	Represents a scan with basic parameters and optional serpent mode.
WhitelightScan	Represents a scan with Whitelight parameters.

Neaspec GmbH